발간등록번호
경북교육 2022-13

따뜻한 경북교육

2022년

데이터 기반 인공지능 교육 자료

경상북도교육청
Gyeongsangbuk-do Office of Education

목 차

2022년 데이터 기반 인공지능 교육 자료

Contents

목 차

Intro.
데이터 기반 인공지능

경산과학고등학교 교사 임 진 숙

01 데이터와 인공지능

1. 데이터는 인공지능과 어떤 관계가 있을까?

데이터 기반 인공지능을 흔히 기계학습(Machine Learning)이라고 부른다. 1959년, 아서 사무엘은 기계 학습을 "기계가 일일이 코드로 명시하지 않은 동작을 데이터로부터 학습하여 실행할 수 있도록 하는 알고리즘을 개발하는 연구 분야"라고 정의하였다.

기계학습은 데이터를 통해 학습하고 새로운 데이터를 예측할 수 있는 모델을 만드는 것이다. 따라서 기계학습을 하기 위해서는 많은 데이터가 필요하다. 많은 양의 데이터(big data)는 데이터 기반 인공지능 모델을 만드는데 중요한 역할을 한다.

2. 기계학습의 종류

기계학습은 크게 지도학습, 비지도 학습, 강화학습의 세 가지로 구분할 수 있다. 이 교재에서는 기계학습의 지도학습을 적용하여 해결할 수 있는 문제만 다루었다.

가. 지도 학습

지도학습(Supervised Learning)은 정답(레이블)이 있는 데이터를 학습하여 데이터가 가지고 있는 특징을 스스로 찾아내는 것으로, 회귀와 분류가 있다. 주어진 데이터와 레이블을 이용해 새로운 데이터의 레이블을 예측해야 할 때 사용한다. 정답 레이블이 있는 데이터를 학습하고 모델을 만들어 얼마나 정답을 잘 예측하였는지 성능을 평가할 수 있다.

나. 비지도 학습

비지도 학습(Unsupervised Learning)은 레이블(정답)이 없는 데이터를 학습하는 방법이다. 정답이 없는 데이터로부터 특징을 발견하고 숨은 패턴을 찾아내는 방식으로 군집화와 차원의 축소가 대표적인 방법이다.

다. 강화 학습

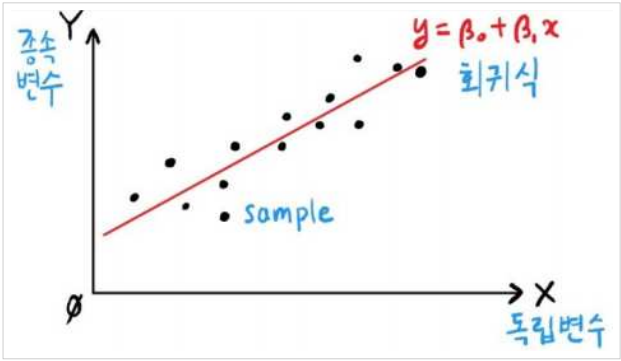
강화학습(Reinforcement Learning)은 보상 및 처벌의 형태로 학습 데이터가 주어지고 에이전트가 최고의 보상을 얻는 쪽으로 행동하도록 학습하는 방식이다.

3. 지도 학습의 회귀와 분류

정답이 있는 데이터를 이용하여 학습하는 지도학습은 회귀와 분류로 구분된다.

가. 회귀

기계학습의 회귀(Regression) 분석은 비교적 적은 데이터로 독립변수나 종속 변수의 관계를 수식으로 표현하는 예측기법이다.



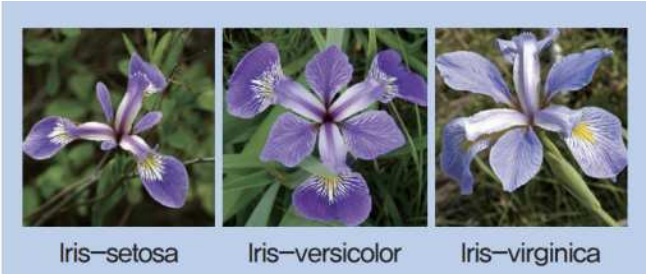
[그림 1] 회귀 모델
(그림 출처: 야사와 만화로 배우는 인공지능)

아파트값이라는 종속변수와 그것을 결정하는 여러 독립변수(평수, 교통, 학군, 편의시설 등)의 관계는 회귀의 방법으로 예측할 수 있다. 기온에 따른 전기 사용량, 도 회귀로 예측할 수 있다.

나. 분류

기계학습의 분류(Classification)는 어떤 패턴을 찾아서 공통되는 특징으로 묶어서 분류하는 것이고, 회귀는 데이터를 통해 학습 후 새로운 데이터의 값을 예측하는 것이다.

붓꽃(iris)의 종류를 구분하는 문제는 대표적인 기계학습의 분류 문제이다. 잘 알려진 영국의 통계학자 로널드 피셔가 만든 붓꽃 데이터를 이용하면 붓꽃의 품종을 정확하게 분류할 수 있다.

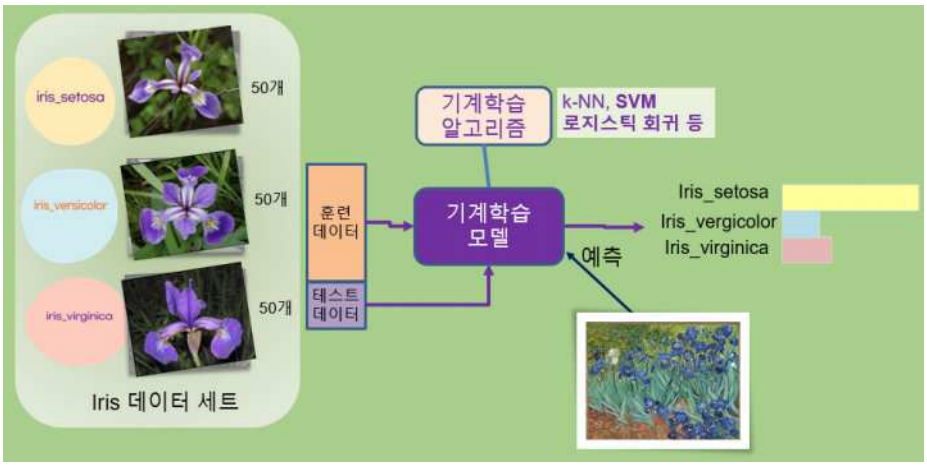


[그림 2] 붓꽃의 세가지 품종
(그림 출처: 인공지능 기초 교과서(씨마스))

4. 훈련 데이터와 테스트 데이터

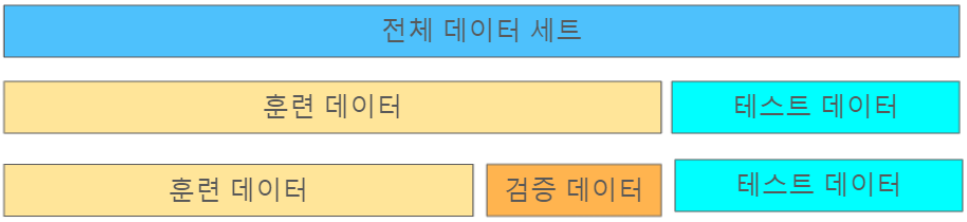
기계학습에서 모델을 만들기 위해 많은 데이터를 사용한다. 많은 데이터를 모아서 만든 데이터 세트를 훈련 데이터와 테스트 데이터로 분리한다. 여기서 훈련 데이터는 모델의 학습에 사용하는 데이터이고, 테스트 데이터는 학습을 마친 모델의 성능을 평가하기 위해 사용하는 데이터이다. 보통 훈련 데이터와 테스트 데이터의 비율을 7:3 또는 8:2정도로 설정한다.

꽃의 품종을 구분하는 기계모델을 구현하는 과정을 살펴보자. 학습 모델을 만들기 위해서 서로 다른 품종의 꽃 데이터를 많이 수집하고, 기계학습 알고리즘을 적용하여 꽃의 품종을 잘 분류할 수 있는 모델을 만든다, 이 모델에 새로운 붓꽃 데이터를 입력하면 구 꽃이 어떤 품종인지 예측할 수 있다. 기계학습의 분류에 사용되는 알고리즘에 k-NN, SVM, 로지스틱 회귀 등이 있다.



[그림 3] 붓꽃의 품종을 분류하는 기계학습 모델

훈련 데이터로 모델 학습을 할 때, 학습을 잘하고 있는지 평가하기 위해 검증 데이터를 사용한다. 훈련 데이터의 일부를 여러 번 떼어 내어 모델을 검증하는데, 이를 k-폴드 교차 검증이라고도 한다. 검증 데이터는 모델 학습시 성능을 검증하기 위해 사용하므로, 검증 데이터의 위치를 바꾸어 가면서 학습과 검증을 반복한다.



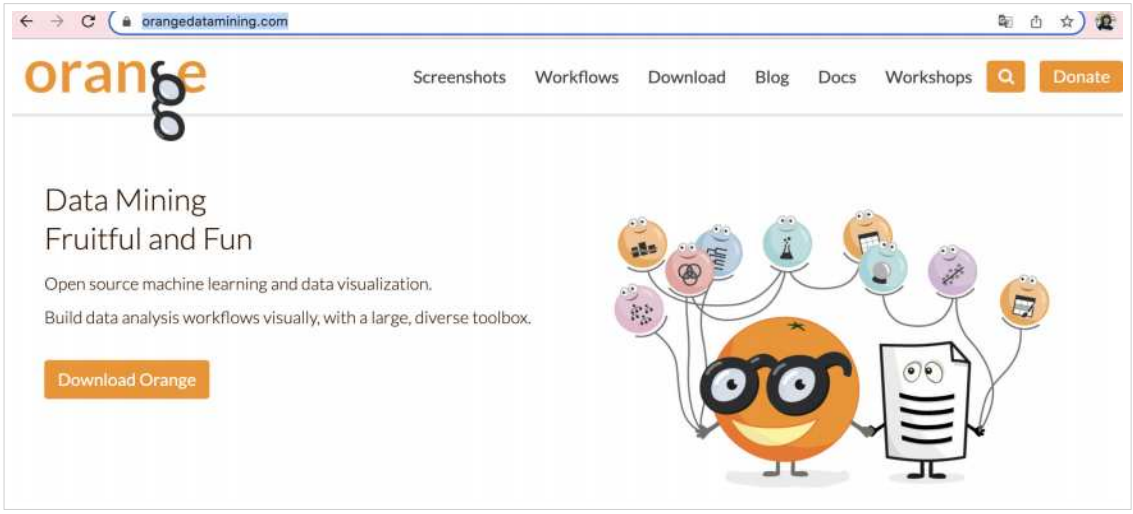
[그림 4] 훈련 데이터, 검증 데이터, 테스트 데이터

02 오렌지를 이용하여 기계학습 구현하기

이 교재는 데이터 기반 인공지능으로 문제를 해결하기 위해 오렌지(Orange)라는 도구를 이용하였다.

1. 오렌지는 무엇일까?

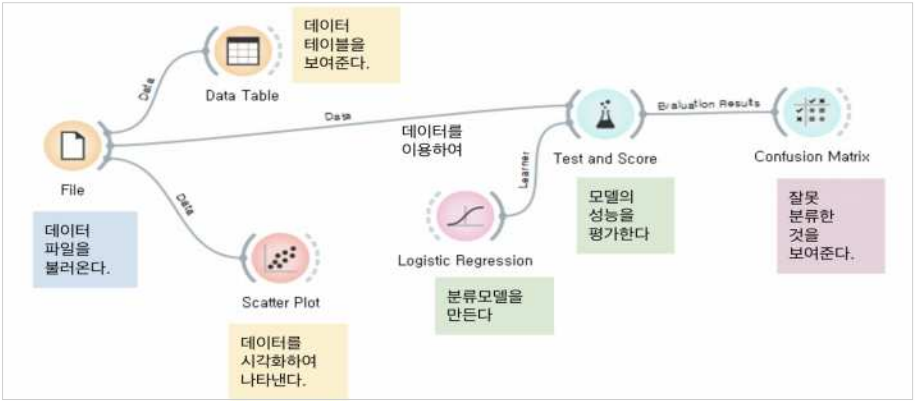
오렌지(Orange)는 오픈 소스 기계학습 및 데이터 시각화를 할 수 있는 도구로서, 크고 다양한 도구 상자를 사용하여 데이터 분석 워크 플로를 시각적으로 구축할 수 있다. 초보자뿐 아니라 전문 데이터 과학자에게도 훌륭한 데이터 마이닝[1] 도구이다. 프로그램은 오렌지(https://orangedatamining.com/) 사이트에서 다운로드할 수 있다.



[그림 5] 오렌지데이터마이닝 사이트

오렌지는 그림과 같이 구성 요소를 워크플로(workflow)에 쌓아 데이터 분석을 수행할 수 있다. 위젯이라고 하는 각 구성 요소는 데이터 가져오기, 전처리, 시각화, 모델링 또는 평가 작업을 포함한다. 워크플로에서 여러 위젯을 연결하면 이동하면서 포괄적인 데이터 분석 스키마를 구축할 수 있다.

1) 데이터 마이닝: 데이터의 미처 몰랐던 속성을 발견하는 것에 집중한 것이다.



[그림 6] 오렌지의 위젯과 워크플로

오렌지의 데이터 시각화로 숨겨진 데이터 패턴을 발견하고, 데이터 분석 절차에 대한 직관을 제공하거나 데이터 과학자와 도메인 전문가 간에 소통하는데 도움이 된다. 시각화 위젯에는 산점도, 상자 플롯 및 히스토그램, 실루엣 플롯 및 트리 시각화와 같은 모델별 시각화가 포함된다. 애드온(Ads-on)을 설치하면 네트워크, 워드 클라우드, 지도 등을 시각화할 수 있다.

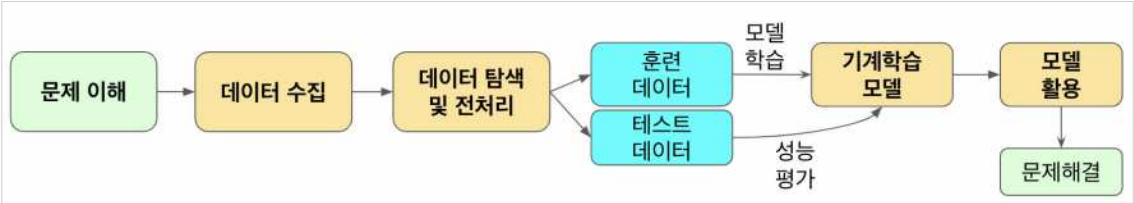
[그림 0-7] 오렌지 3의 시각화 도구

2. 기계학습을 통한 문제해결 과정

기계학습으로 문제를 해결하는 과정을 살펴보자.

인공지능으로 문제를 해결하는 하기 위해 먼저 문제를 이해하고 문제해결에 필요한 데이터 수집한 후, 데이터를 탐색하여 기계학습에 필요한 형태로 전처리한다. 전처리를 마친 데이터 세트를 훈련 데이터와 테스트 데이터로 분할한다. 경우에 따라서는 데이터 분할후 데이터를 전처리하기도 한다. 훈련 데이터는 모델 학습에 사용하고 그 결과 기계학습 모델이 만

들어진다. 그리고 테스트 데이터로 만들어진 모델의 성능을 평가한다. 테스트 데이터로 모델의 성능이 우수하다면 모델을 활용하여 문제를 해결할 수 있다. 기계학습으로 문제를 해결하는 과정을 그림으로 나타내면 다음과 같다.



[그림 8] 데이터 기반 인공지능으로 문제해결 과정

[참고문헌]

권건우(2020). 야사와 만화로 배우는 인공지능. 루나파인북스.

오렌지. 오렌지의 특징. https://orange.biolab.si/#Orange-Features. 2021.12.10. 검색

위키피디아. 기계학습. https://ko.wikipedia.org/wiki/기계_학습. 2022. 3.22. 검색

All-in-One(2020). 인공지능언플러그드. http://ai4edu.kr

이영준외 6인. 인공지능 기초 교과서. 씨마스.

이 교재는 13개의 주제를 통해 데이터 기반 인공지능에 대해 교육할 수 있는 자료입니다.

각 장별 내용을 요약하면 다음과 같습니다.

1장 '무는 원숭이를 찾아라'는 이미지 데이터를 이용하여 기계학습의 분류를 배울 수 있습니다. 원숭이 얼굴 특징을 보고 무는 원숭이인지 물지 않는 원숭이인지 분류할 수 있는 기계학습 모델을 만들어 봅니다. 훈련 데이터로 학습하고, 테스트 데이터를 이용하여 모델의 성능을 평가해 볼 수 있습니다. 이 과정에서 이미지 데이터를 전처리하는 방법, 모델의 성능평가 지표 등을 배울 수 있습니다

이 자료는 인공지능 언플러그드 활동으로 배운 것을 오렌지를 이용해 실제로 기계학습 모델을 구현해 볼 수 있다는 점이 좋습니다. 수업에서는 이미지 데이터를 활용하여 기계학습의 분류를 지도할 때 활용할 수 있습니다.

2장 개인에 관한 신상정보 데이터를 이용하여 기계학습의 회귀에 관하여 배울 수 있습니다. 기존 통계자료에 의하면 개인의 소득에 학력이 많은 영향을 미친다고 예상합니다. 학력만이 아닌 성별, 가족 구성원 수, 태어난 해 등 개인에 관한 여러 가지 정보를 캐글 사이트에서 수집하여 Linear Regression 등의 기계학습 알고리즘을 통해 학습하고 테스트해본 후 모델의 성능을 평가해볼 수 있습니다.

3장 청소년이 느끼는 행복감을 기계학습을 알아보기 위해 세계행복보고서를 분석하여 설문 내용을 작성하여 실제 우리 학교 학생을 대상으로 관련 데이터를 수집하였습니다. 데이터 시각화를 통해 행복에 영향을 미치는 요소를 찾아보고, SVM 등의 회귀 알고리즘을 사용하여 데이터를 학습하고 성능을 평가해볼 수 있습니다. 청소년들이 삶에서 중요하다고 느끼고 있는 것은 무엇인지, 어떤 요소가 자신의 삶에 행복감을 줄 수 있는지 알아보고 나아가 우리가 행복의 기준을 무엇에 두고 살아가는지 어떻게 하면 행복한 삶을 살아갈 수 있는지 생각해볼 수 있습니다.

4장 밀알 낟알의 길이, 너비 등의 다양한 수치 데이터를 이용하여 기계학습의 분류에 관하여 배울 수 있습니다. 오렌지3에서 제공하는 다양한 데이터 시각화 도구를 이용해 훈련에 사용된 데이터의 특성을 표현할 수 있으며 시각화 된 데이터의 특성을 이용하여 밀알의 품종 분류에 가장 많은 영향을 미치는 속성이 무엇인지 예측해보고 이를 확인해 볼 수 있습니다. 또한 하나의 데이터 파일에서 훈련 데이터와 테스트 데이터를 분리하여 학습할 수 있도록 스프레드시트를 이용해 데이터를 처리하는 방법에 대해서도 알 수 있습니다.

5장 코로나19 환자의 흉부 X-ray 사진을 이용하여 코로나 감염 여부를 분류해 보며 비정형 데이터인 이미지를 이용한 기계학습의 분류에 관하여 배울 수 있습니다. 훈련 데이터를 이용해 학습을 진행하고 테스트 데이터를 이용해 분류 성능을 평가하는 과정에서 제시되는 다양한 분류 성능 평가 척도에 대해 학습하고 이를 이용해 분류 모델의 성능을 비교해 보며 학습 데이터를 가장 잘 학습한 모델을 찾을 수도 있습니다.

6장 태아의 심박수, 움직임 등의 수치 데이터를 이용하여 기계학습의 분류에 관하여 배울 수 있습니다. 데이터 시각화를 통해 건강 상태에 미치는 요소를 찾아보고, Random Forest 등의 회귀 알고리즘을 사용하여 데이터를 학습하고 성능을 평가해볼 수 있습니다. 훈련 데이터와 테스트 데이터로 분리하여 모델학습과 모델의 성능을 평가해볼 수 있습니다.

7장 교통사고의 개별 정보를 이용하여 기계학습의 분류를 학습할 수 있습니다. 사고유형, 도로상태, 기상상태 등의 속성들이 교통사고 발생시 피해자의 신체상해정도에 영향을 미치는지를 파악하는 활동을 할 수 있습니다. 또 훈련 데이터와 테스트 데이터를 나누어 훈련 데이터로 교통사고발생을 분류하는 기계학습 모델을 만든 후 테스트 데이터를 통해 기계학습 모델의 성능을 평가해볼 수 있습니다.

8장 영화 평점 리뷰에서 많이 등장하는 단어는? 에서 웹크롤링을 통해 영화 평점과 리뷰를 수집하고 워드 클라우드를 이용하여 시각화 및 데이터에 나타난 감정을 분석해볼 수 있습니다. 기본적인 정보교과시간이나 동아리활동에서 파이썬을 학습한 학생들에게 나아가 데이터를 크롤링해볼 수 있는 기회를 제공하고 나아가 다른 리뷰를 크롤링하여 워드클라우드와 감정분석을 해볼 수 있습니다.

9장 생선 눈 이미지로 생선의 신선도를 구분하는 활동은 이미지 데이터를 이용하여 기계학습의 분류를 배울 수 있습니다. 신선한 생선 눈과 신선하지 않은 생선 눈 이미지로 신선도를 분류할 수 있는 기계학습 모델을 만들어 평가해볼 수 있습니다.

10장 심부전증 데이터세트를 가지고 심부전증 발생에 대한 기계학습의 분류를 학습할 수 있습니다. 나이, 성별, 가슴 통증 유형, 혈압의 수치, 콜레스테롤 수치 등의 속성들이 심부전증 발생에 영향을 미치는지 시각화를 통해 파악하는 활동을 할 수 있습니다. 또 훈련 데이터와 테스트 데이터를 나누어 훈련 데이터로 심부전증 발생을 분류하는 기계학습 모델을 만든 후 테스트 데이터를 통해 기계학습 모델의 성능을 평가해볼 수 있습니다.

11장 바다 동물 중 바다표범, 고래, 상어 이미지 데이터를 활용하여 이미지를 분류하는 기

계학습 모델을 만들어봅니다. 비슷한 생김새의 고래와 상어, 조금 다른 생김새의 바다표범까지 기계학습 모델이 이 셋의 바다 동물을 잘 분류할 수 있을까요? 이미지 데이터를 기계학습에 이용하기 위해 이미지 임베딩을 수행하는 과정을 학습할 수 있습니다. 훈련 데이터와 테스트 데이터를 분류하고 훈련 데이터로 바다 동물을 분류하는 기계학습 모델을 만들 수 있어요. 또 테스트 데이터를 통해 만든 기계학습 모델의 성능을 평가해볼 수 있습니다.

12장은 피마 인디언 당뇨병 데이터를 이용하여 당뇨인지 아닌지 분류하는 기계하는 모델을 만들어 당뇨병은 예측할 수 있습니다. 피마 인디언 당뇨병 데이터에는 이상치가 많아서 이러한 데이터를 전처리하는 방법, 훈련 데이터와 테스트 데이터로 분리하여 모델 학습과 모델의 성능평가에 사용하는 방법을 배울 수 있습니다.

13장은 코로나 19 확진자 수 데이터를 분석하여 이를 보기 쉬운 형태로 시각화 해보고 국가별 HDI 데이터와 연관지어 어떤 요소가 코로나19 확진자 수와 가장 관련이 있는지 분석해봅니다. 코로나19 확진자 데이터의 전처리 방법과 다른 두 개의 데이터를 병합하는 방법, 데이터 시각화와 Rank로 주요 속성 추출하는 방법을 배울 수 있습니다.

MEMO

01. 무는 원숭이를 찾아라

경산과학고등학교 교사 임 진 숙

학습 진행 과정

1단계	데이터 준비	- 사용 데이터: Monkeydataset - 수집: https://aiunplugged.org - 데이터 편집: 훈련 데이터와 테스트 데이터 분리
2단계	데이터 불러오기	- 학습 데이터 불러오기 - 데이터의 속성별 Role(역할) 설정하기
3단계	데이터 시각화	- 시각화 도구을 이용한 데이터 탐색 - 사용한 시각화 도구: Radviz
4단계	데이터 전처리	- 이미지 임베딩
5단계	모델 학습	- 분류를 이용한 모델 학습 - 사용된 알고리즘: SVM, Neural Network, Logistic Regression, k-NN
6단계	성능 평가	- test and score, cross validation을 이용한 성능 평가 - 혼동 행렬을 이용한 성능 평가
7단계	예측	- Prediction 위젯으로 테스트 데이터로 예측하기

학습 내용 요약

데이터 종류	문제 유형	기계학습 알고리즘	성능 평가 도구
비정형 데이터(이미지)	분류	SVM, Neural Network, Logistic Regression, k-NN	CA 혼동행렬

문제 상황

많은 사람들이 찾는 동물원에서 많은 동물들을 만날 수 있다. 동물은 대체로 사람을 물지 않지만 사나운 동물들도 있다. 동물원에서 만나는 동물들이 사람을 문다면 여러 가지 문제가 생길 수 있다. 동물의 얼굴을 보고 무는지 물지 않는지 검사 후 출입한다면 사람들이 안심하고 동물 가까이에서 만날 수 있을 것이다. 우리는 원숭이 동물원의 사육사이다. 원숭이 동물원의 입구에 카메라를 설치하여 무는 원숭이인지 물지 않는 원숭이인지 알려주는 인공지능 장치를 설치하려고 한다.

01 데이터를 준비하자!

1 Monkey Dataset

원숭이 데이터 세트(Monkey Dataset)는 독일의 Stefan Seegerer과 Annabel Lindner가 개발한 AI 언플러그드(https://aiunplugged.org)의 원숭이 이미지를 이용하였다. 무는 원숭이와 물지 않는 원숭이 이미지는 20개의 심플 버전과 40개의 고급 버전이 있는데, [그림 1-1]과 같이 2가지 색으로 구성된 40장의 고급 버전 이미지를 사용하였다. 이미지를 잘라서 [그림 1-1]과 같이 파일 이름을 붙이고 전체 이미지 데이터를 Monkey Dataset이라고 명명하였다.

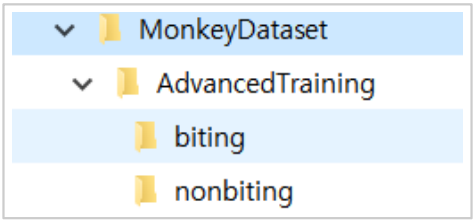
[그림 1-1] MonkeyDataset

2 기계학습을 위한 데이터 준비

지도 학습을 위해서는 데이터에 정답 레이블을 붙여야 한다. 오렌지 프로그램으로 기계 학습을 하기 위해 Monkey Dataset 폴더를 만든다.

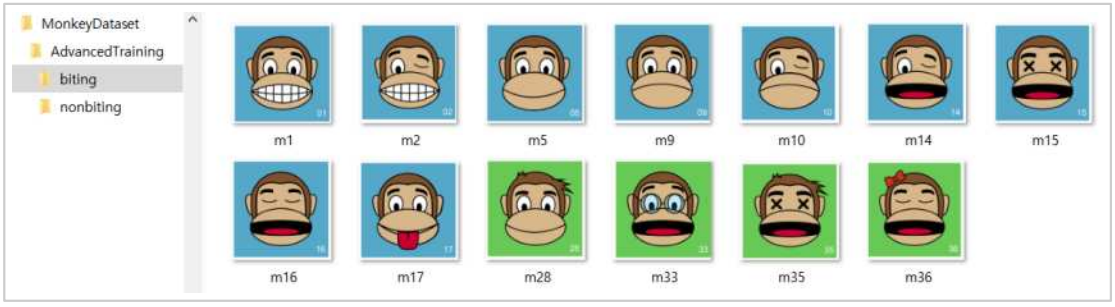
① 훈련 데이터

[그림 1-2]와 같이 다시 하위 폴더를 만들고 각각 무는 원숭이와 물지 않는 원숭이의 이미지 파일을 추가하면, 이것이 곧 이미지 데이터에 레이블을 붙이는 것과 같다. [그림 1-2]와 같이 Training 이름의 폴더를 만들고, 하위 폴더에 biting과 nonbiting의 클래스 폴더를 만들었다. 각각의 폴더에 훈련 데이터 이미지를 넣는다.



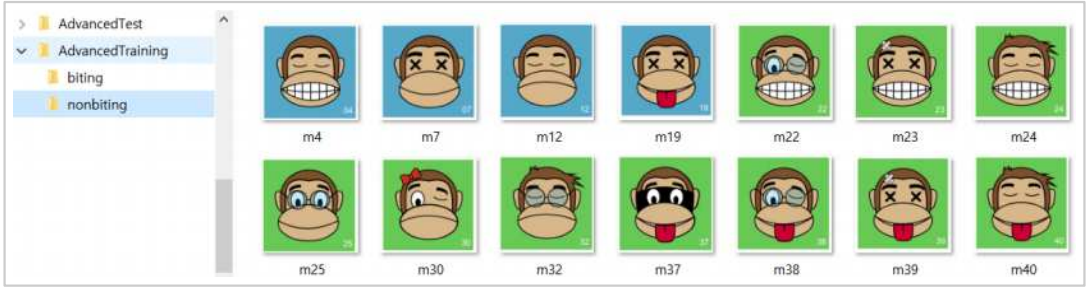
[그림 1-2] 훈련 데이터 폴더 구성

훈련 데이터의 biting 폴더에는 [그림 1-3]과 같이 무는 원숭이 15장의 이미지가 들어가도록 한다.



[그림 1-3] 무는 원숭이(biting) 훈련 데이터

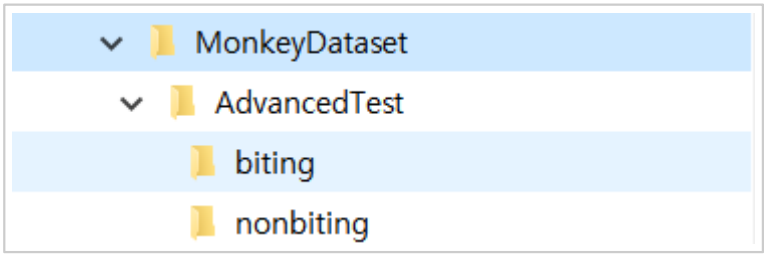
훈련 데이터의 nonbiting 폴더에도 [그림 1-4]와 같이 16장의 이미지가 들어가도록 한다.



[그림 1-4] 물지 않는 원숭이(nonbiting) 훈련 데이터

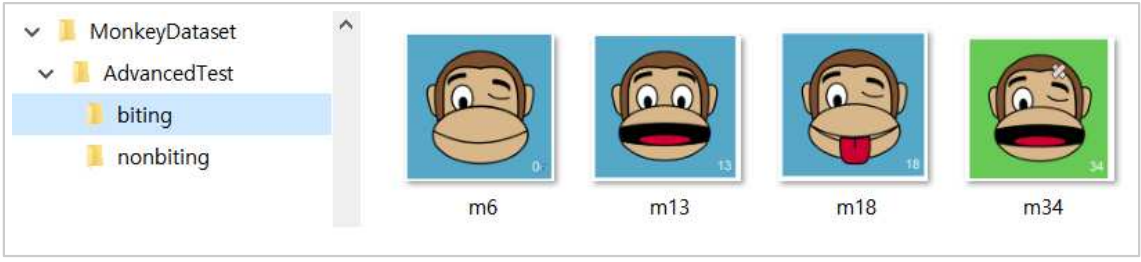
② 테스트 데이터

테스트 데이터도 마찬가지로 [그림 1-5]와 같이 Test 폴더를 만들고 하위폴더에 biting과 nonbiting을 폴더를 만들어 이미지를 넣는다.



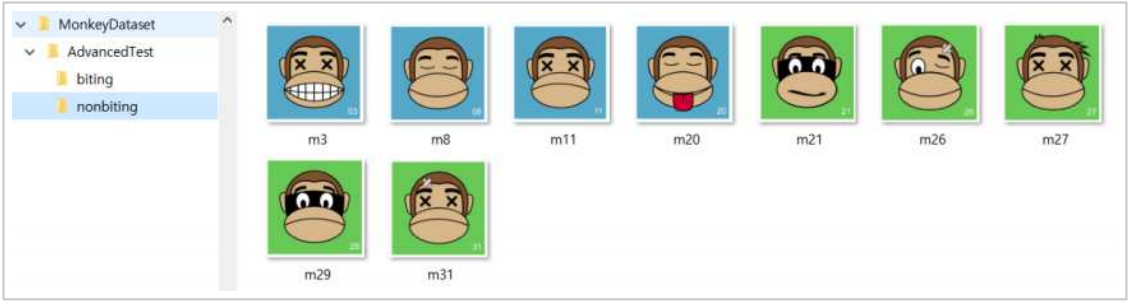
[그림 1-5] 테스트 데이터 폴더

테스트 데이터의 biting 폴더에는 [그림 1-6]과 같이 무는 원숭이 4장의 이미지가 들어가도록 한다.



[그림 1-6] 무는 원숭이(biting) 테스트 데이터

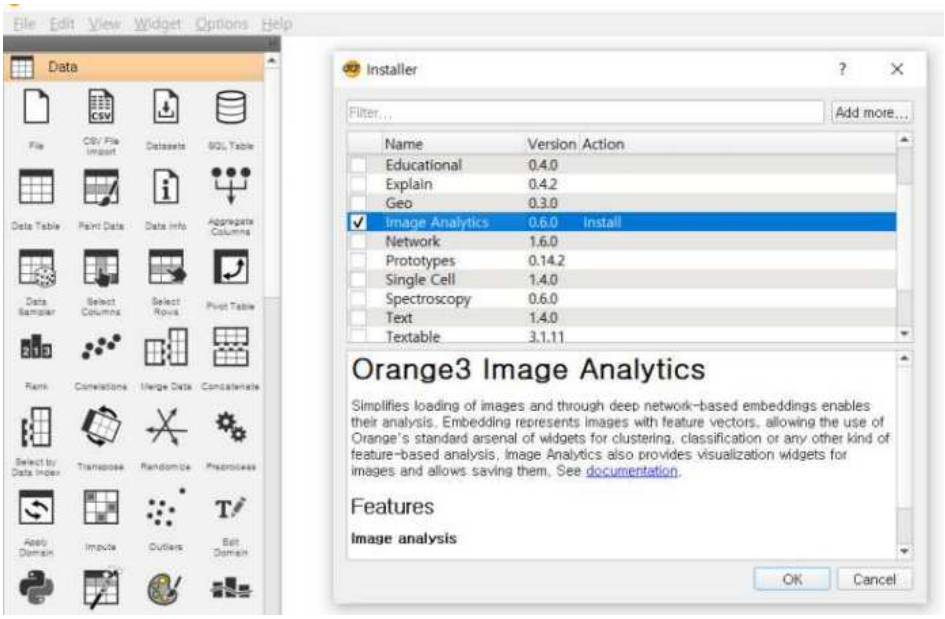
테스트 데이터의 nonbiting 폴더에도 [그림 1-7]과 같이 10장의 이미지가 들어가도록 한다.



[그림 1-7] 물지 않는 원숭이(nonbiting) 테스트 데이터

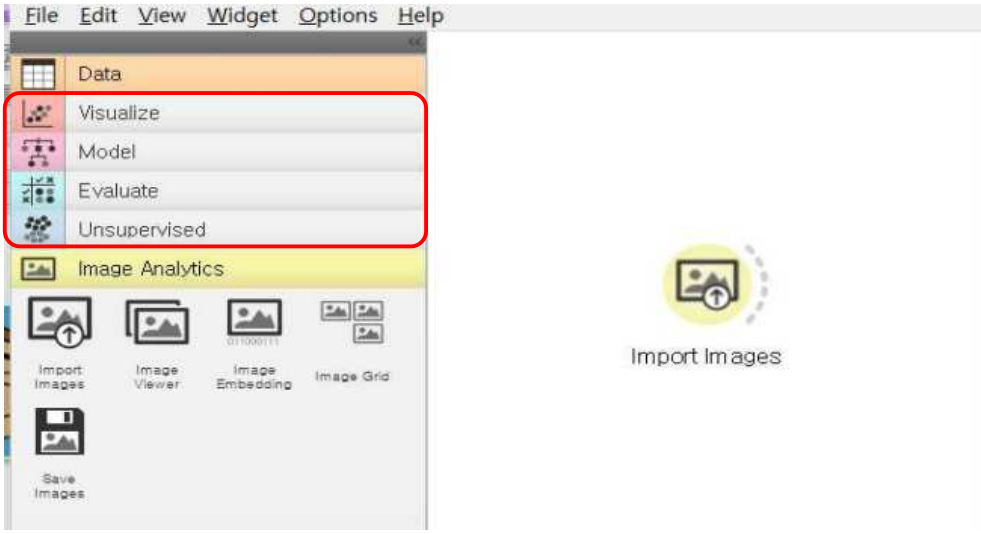
3 오렌지에 학습 데이터 불러오기

먼저 이미지 분석을 위하여 [Options] 메뉴에서 [Add-ons]을 클릭하면 다음과 같은 창이 나타난다. 추가 기능 중에서 ImageAnaytics앞에 체크(☑)를 눌러 추가 설치한다.



[그림 1-8] Image Anaytics 설치

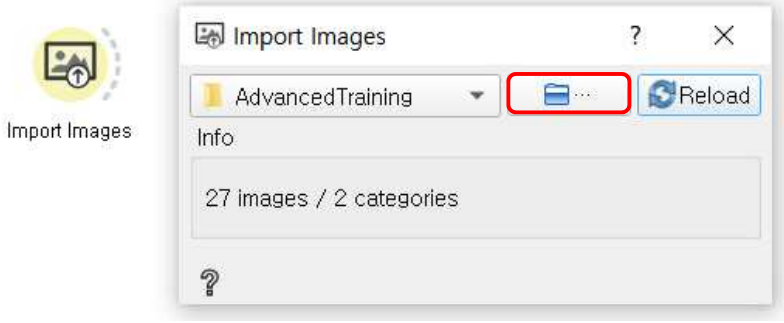
설치 후 오렌지를 종료하고 다시 실행해야 추가 설치된 기능을 사용할 수 있다. [그림 1-9]와 같이 오렌지 실행시 왼쪽 위젯 도구상자에 Image Analytics가 추가되었다.



[그림 1-9] Image Analytics 도구

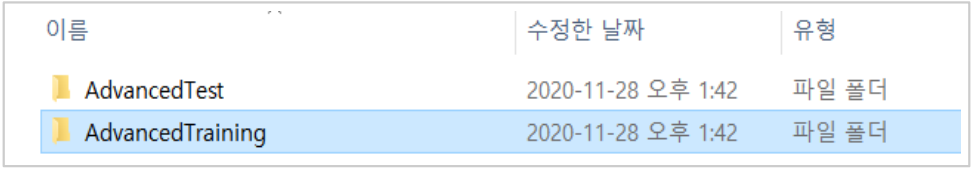
① 학습 데이터 불러오기

이미지 학습 데이터를 불러오기 위해 Import Images 위젯을 선택하여 창에 놓는다. Import Images을 더블 클릭하면 [그림 1-10]과 같이 이미지를 업로드하기 위해 폴더를 선택할 수 있다.



[그림 1-10] 훈련 데이터 입력

MonkeyDataset에서 Training 폴더를 선택한다.



[그림 1-11] 훈련 데이터 폴더 선택하기

② 이미지 데이터 보기

이미지를 확인하기 위해 [Image Analytics]위젯에서 Image Viewer 위젯을 끌어내어 Import Images 위젯과 연결한다.

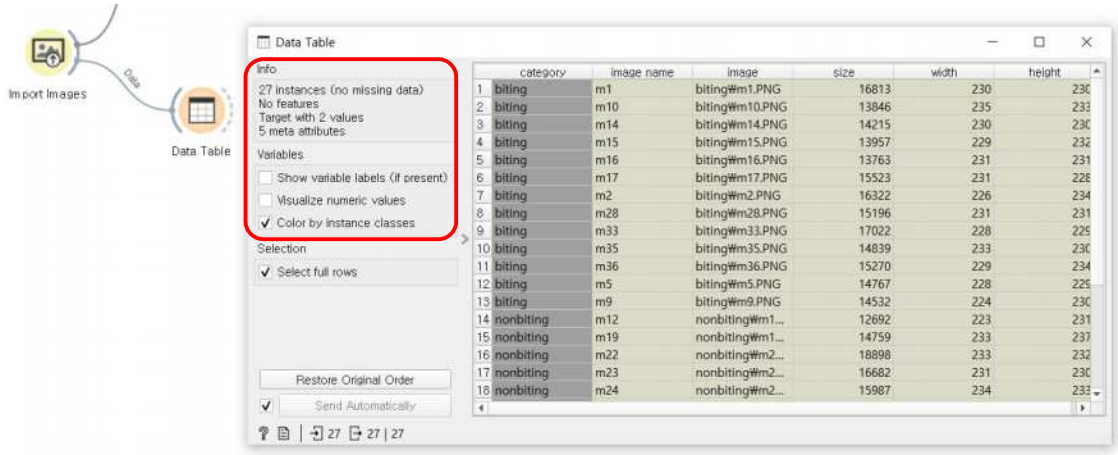
[그림 1-12] Image Viewer 연결

Image Veiwer 위젯을 더블 클릭하면 폴더에 있는 이미지를 이름과 함께 볼 수 있다.

[그림 1-13] Image Viewer로 이미지 보기

③ 데이터 테이블 보기

가져온 이미지에 데이터 테이블을 연결하면 [그림 1-14]와 같이 이미지 이름과 크기, 가로 세로 길이 등을 테이블 형태로 보여준다. 정보(Info)를 살펴보면 2개의 값을 지닌 독립변수(Target)가 있으며, 기계학습을 위한 독립변수(features)는 없고, 5개의 meta 속성으로만 구성되어 있다.

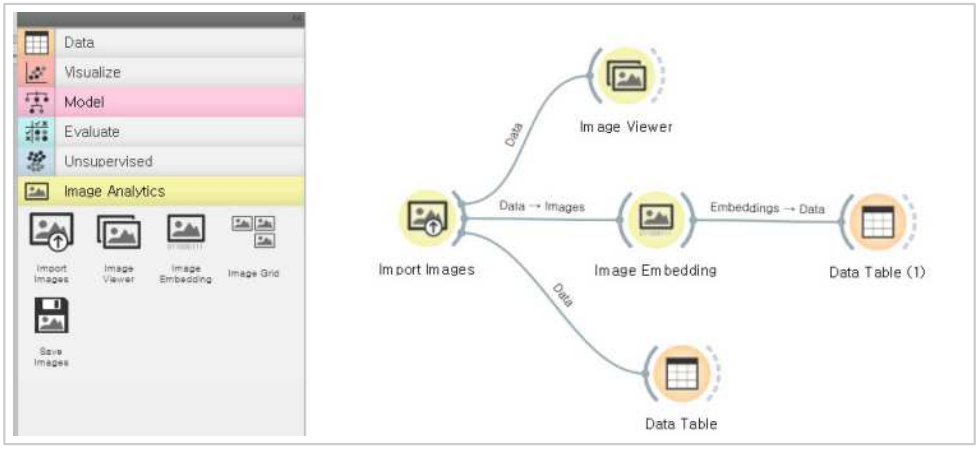


[그림 1-14] 훈련 데이터 테이블

02 데이터를 탐색하고 전처리하자

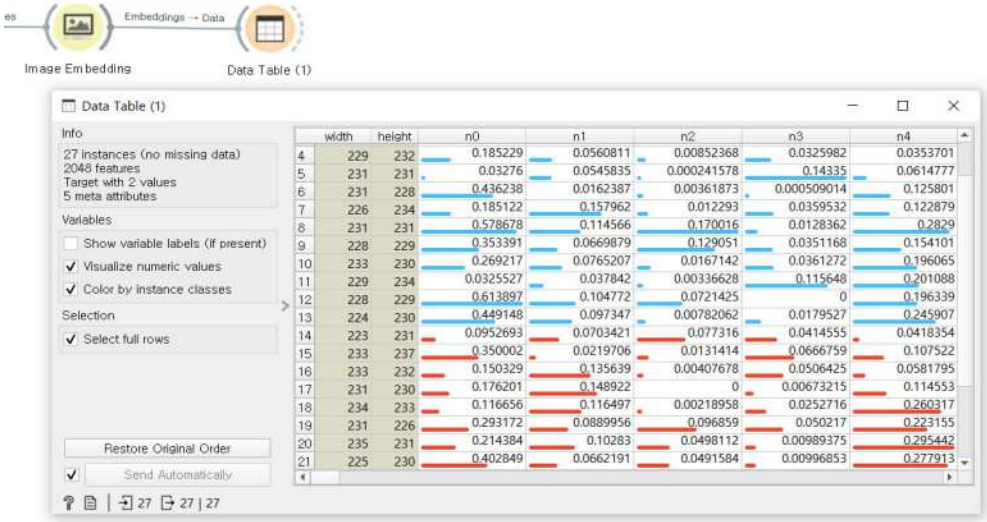
1 훈련 데이터 이미지 임베딩

이미지를 기계학습에 이용하려면 이미지 임베딩(Image Embedding)을 수행해야 한다. 이미지 임베딩은 딥러닝을 이용하여 각 이미지의 특징 벡터를 추출해 낸다. 이미지 임베딩을 수행하고 데이터 테이블을 살펴보면 특징 벡터 속성이 추가되어 있다.



[그림 1-15] 이미지 임베딩으로 기계학습을 위한 처리

임베딩한 후 데이터 테이블을 열어보면 [그림 1-16]과 같이 2048개의 특징(features)들이 추가된 것을 확인할 수 있다. 추가된 특성은 이미지 데이터의 내용에서 특징을 추출하여 수치화하여 나타낸 것이다. 인공지능은 이 속성을 이용하여 모델 학습을 한다.



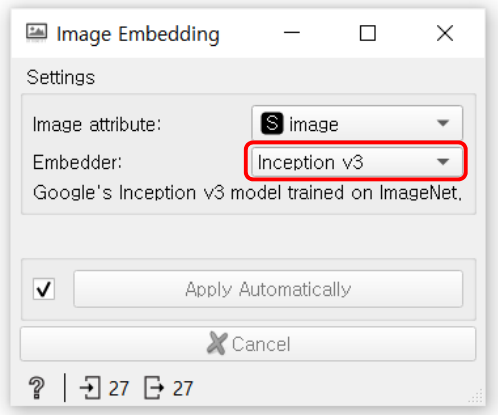
[그림 1-16] 이미지 임베딩한 결과 데이터 테이블

※ 이미지 임베딩(Image Embedding)이란 무엇일까?

이미지 데이터는 기계학습에 바로 사용할 수 없기 때문에 숫자 형태의 벡터로 변환하는 전처리 과정이 필요하다.

Image Embedding 위젯은 수치화된 벡터로 변환하기 위해 사전 훈련된 임베더(Embedder)를 사용한다. 오렌지에서 제공하는 임베더는 사전 훈련된 심층 신경망(Deep Neural Network)을 사용한다. 임베더 옵션을 클릭해보면 Inception V3, VGG-16, VGG-19 등이 있다.

Image Embedding 위젯의 기본 옵션인 Inception V3는 구글이 'GoogLeNet'이란 이름으로 발표한 합성곱 신경망으로 V3는 세 번째 버전을 의미한다.



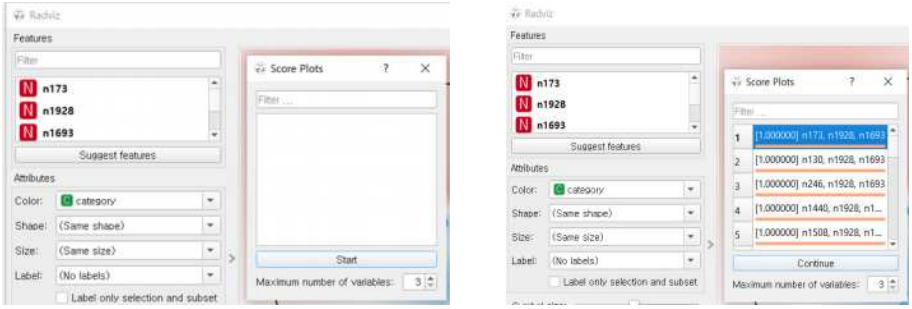
[그림 1-17] 이미지 임베딩에서 Inception v3 임베더

* 여기서 주의해요 : Inception V3, VGG-16, VGG-19와 같은 임베더를 사용하기 위해서는 컴퓨터가 인터넷에 연결되어 있어야 한다. 만약, 컴퓨터가 인터넷에 연결되지 않으면 두 번째 옵션인 SqueezeNet(local)를 사용하기 때문에 1000개의 특징만 추출해낸다.

2 이미지 특성 시각화

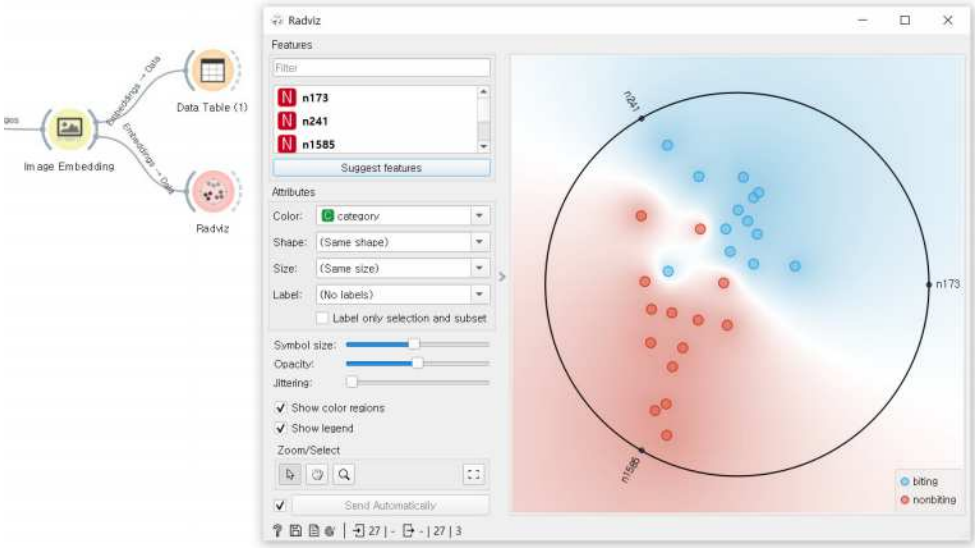
이미지 임베딩한 결과 데이터 테이블을 살펴보면 2048개의 특성이 추가되고 수치화된 값으로 나타난다. 이 특성으로 분류가 가능한지 알아보기 위해 Radviz 위젯으로 시각화하여 나타내었다. Radviz 위젯은 3개 이상의 변수의 데이터를 2차원에 투영하여 투영하여 시각화해 준다. 데이터 인스턴스는 원 내부의 점으로 나타나며, 다차원의 데이터 속성으로 분류가 가능한지 나타내어 준다.

Radviz 위젯에서 'Suggest feature'기능을 클릭하고, 'start'를 누르면 수 많은 속성 중에 어떤 속성의 조합으로 분류가 가능한지 찾아서 추천해준다.



[그림 1-18] Raviz 시각화를 위한 특징 추출

아래 그림은 2048개의 속성 중 n173, n241, n1585의 세가지 속성으로 조합할 때, 무는 원숭이와 물지 않는 원숭이를 분류할 수 있다는 것을 보여준다.

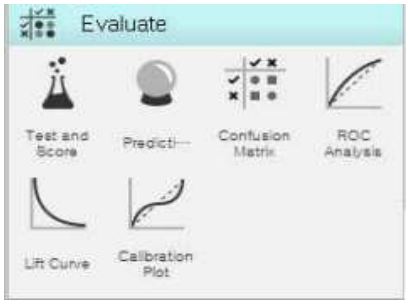


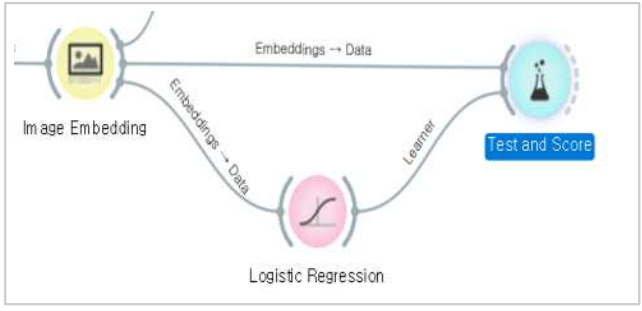
오렌지의 Radviz 위젯은 기계학습에 영향을 미치는 핵심 속성을 스스로 찾아서 추천해 준다. 이를 통해 이미지 임베딩 위젯으로 수치화된 특성 값은 이미지 분류에 유용하게 사용할 수 있다는 것을 알 수 있다.

03 모델 학습과 성능 평가

1 학습 모델 선택하고 학습시키기

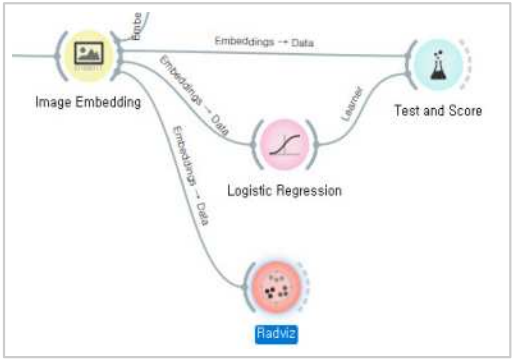
이미지 임베딩으로 처리한 이미지 데이터와 기계학습 알고리즘을 연결하면 모델을 만들 수 있다. 모델 학습에 필요한 것은 임베딩한 이미지 데이터와 기계학습 알고리즘이다. 모델 학습의 결과, 얼마나 정확히 분류하는지 평가하기 위해 Test and Score 위젯을 연결한다. Test and Score는 Evaluate 도구에서 찾아볼 수 있다.

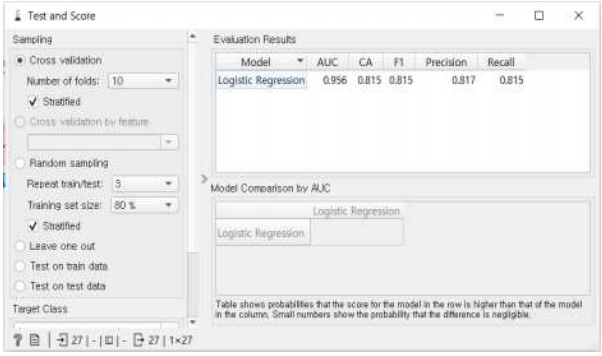




2 성능 평가하기

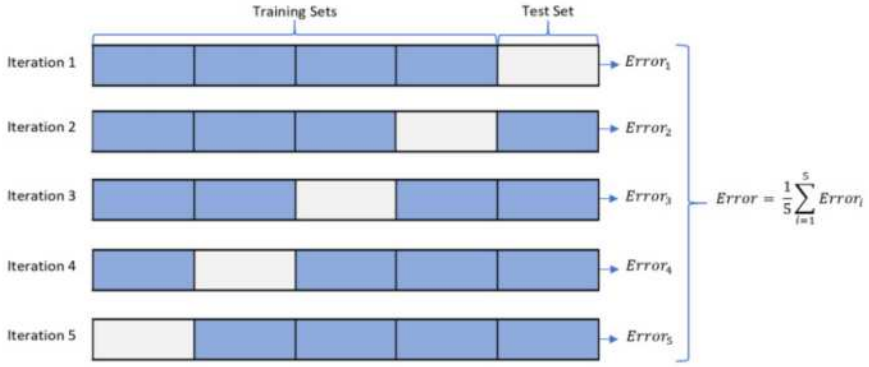
Test and Score 위젯을 통해 다양한 성능평가 방법을 설정할 수 있다. 여기서는 별도의 Test 데이터가 있기 때문에 27개의 훈련 데이터를 이용하여 교차 검증(Cross validation)을 선택하고 folds의 수를 10으로 설정하여 성능 평가하여, 분류 정확도가 0.815으로 나타났다. 여기서 Cross validation을 선택했기 때문에 테스트 데이터를 사용하지 않고 훈련 데이터로 검증한 결과이다.





※ 교차 검증(Cross Validation)이란 무엇인가?

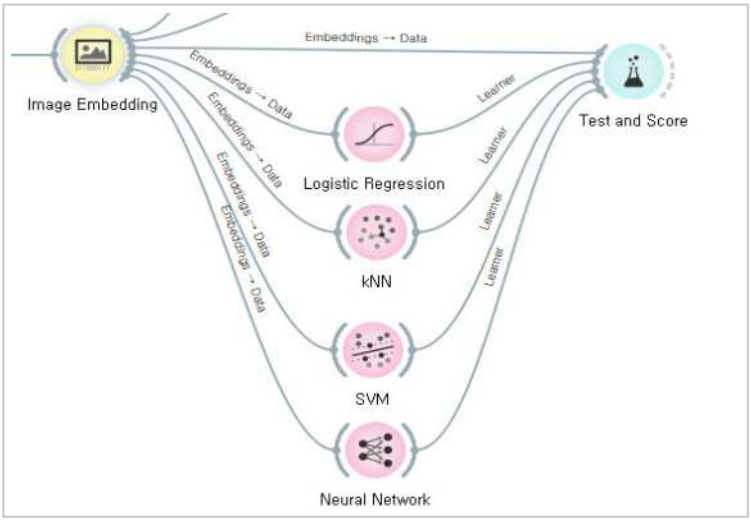
교차 검증이란 훈련 데이터의 일부를 여러 번 떼어 내어 모델을 검증하는데, k-폴드 교차 검증이라고도 한다. 이 때 사용된 데이터를 검증 데이터라고 한다. 예를 들여, k가 5일 때 교차 검증은 훈련데이터를 5등분하여 1/5을 검증 데이터로 사용하고 나머지는 학습 데이터로 사용한다. 검증 데이터는 모델 학습시 성능을 검증하기 위해 사용하므로, 검증 데이터의 위치를 바꾸어 가면서 총 5번의 학습과 검증을 반복한다.



K 폴드 CV, K=5

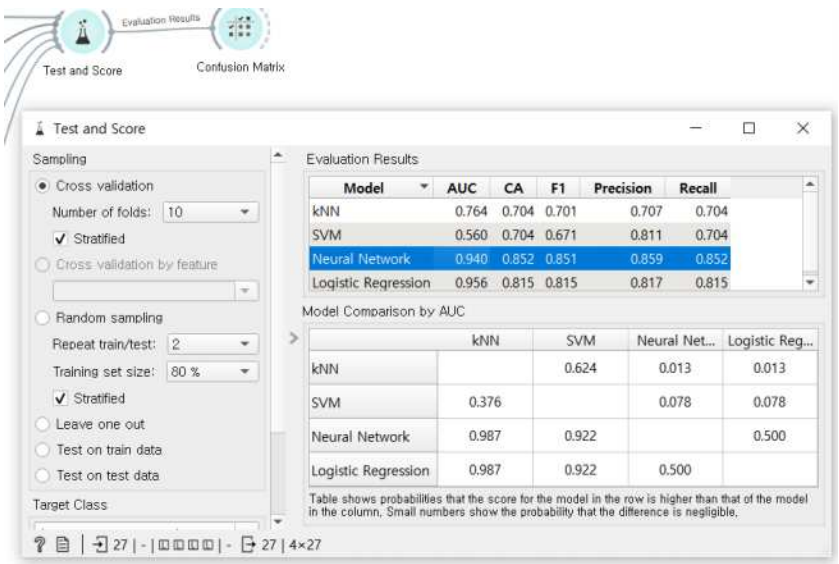
※ 그림 출처: https://towardsdatascience.com/cross-validation-k-fold-vs-monte-carlo-e54df2fc179b

오렌지에서는 여러 모델을 동시에 연결하여 어느 것이 성능이 좋은 지 비교해볼 수 있는 장점이 있다. 여기서는 Logistic Regression과 kNN, SVM, Neural Network 모델을 동시에 연결하였다.



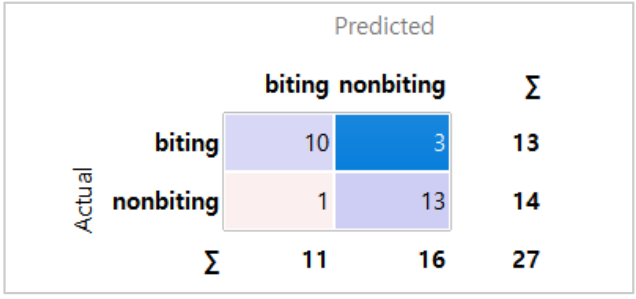
[그림 1-19] 모델 학습

27개의 훈련 데이터를 사용하여 통해 10번 교차 검증한 결과를 비교해 보면 Neural Network 모델의 성능(CA)이 가장 우수한 것으로 나타났다. 따라서 무는 원숭이와 물지 않는 원숭이를 분류하는 문제는 Neural Network을 이용하여 모델을 만드는 것이 적합하다고 말할 수 있다.



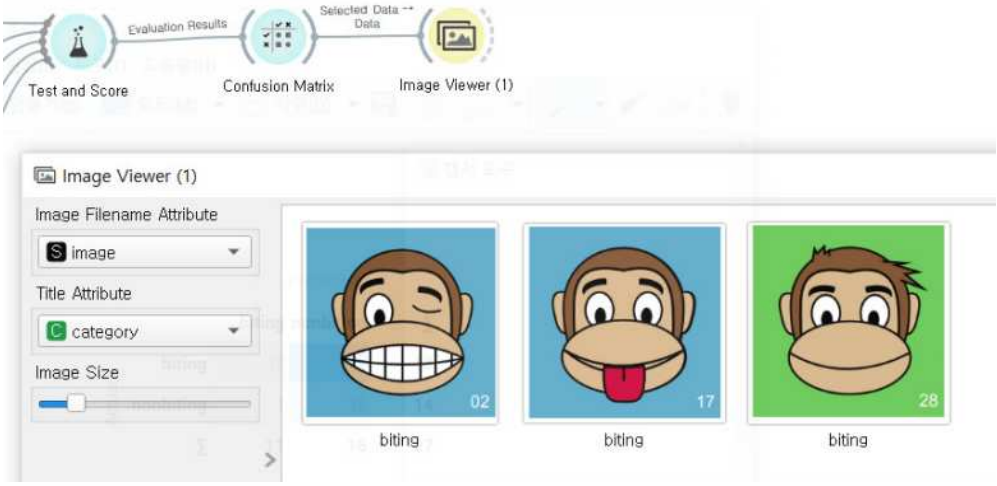
[그림 1-20] 모델 학습과 검증 결과

검증 데이터로 테스트한 결과를 혼동 행렬과 연결하여 실제 데이터를 어떻게 예측하였는지 살펴보면 [그림 1-21]과 같이 무는 원숭이 13 개중 10개를 무는 원숭이로 예측하였고, 물지 않는 원숭이는 14개중 13개를 물지 않는 것으로 예측하였다. 따라서 분류 정확도(CA)는 $\frac{10+13}{27}=\frac{23}{27}$=0.852로 나타났다.



[그림 1-21] 모델 학습 결과의 혼동 행렬

무는 원숭이를 물지 않는 원숭이로 분류한 3가지 이미지를 보고자 할 때 아래와 같이 Image viewer를 더블 클릭하면 이미지를 확인할 수 있다.



[그림 1-22] 모델 학습에서 잘 못 분류한 이미지

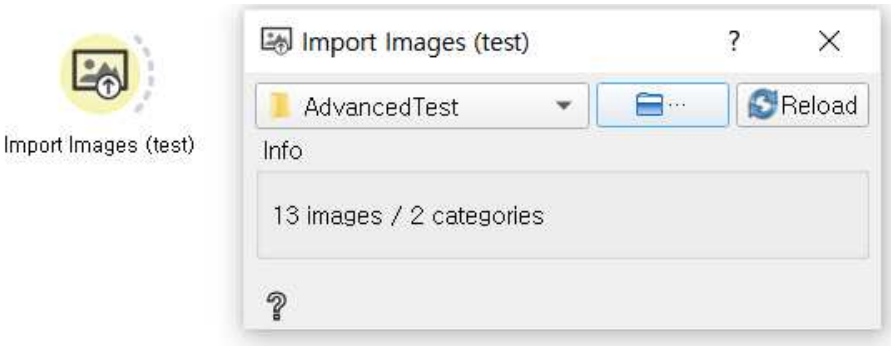
3 테스트 데이터로 예측하기

이제 테스트 데이터로 얼마나 무는 원숭이를 얼마나 잘 예측할 수 있는지 살펴보자.

① 테스트 데이터 불러오기

테스트 데이터도 같은 방법으로 Import Images 위젯으로 [Test] 폴더에 저장된 테스

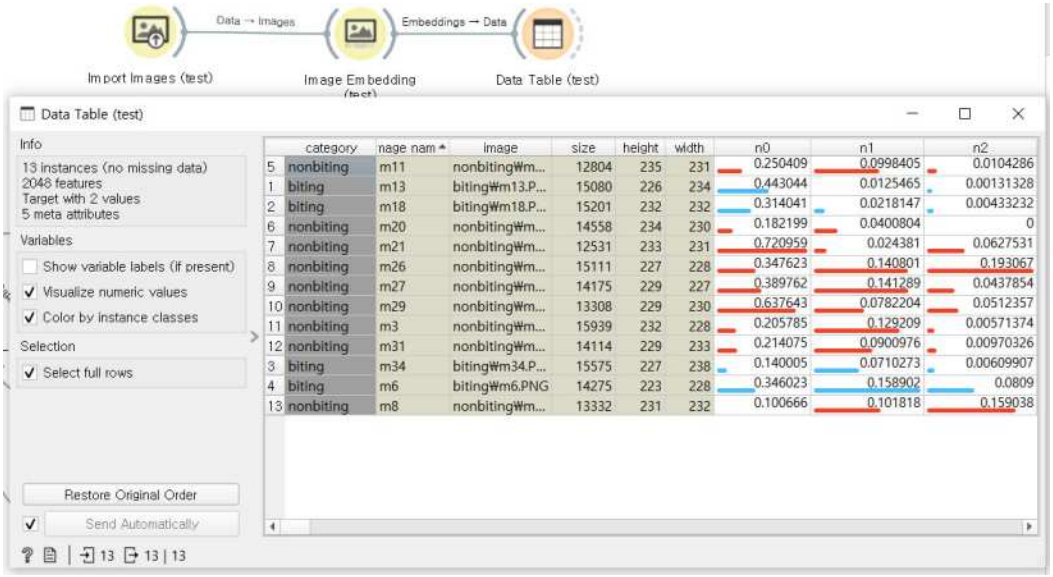
트 데이터 이미지를 불러온다.



[그림 1-23] 테스트 데이터 로드하기

② 테스트 데이터 이미지 임베딩

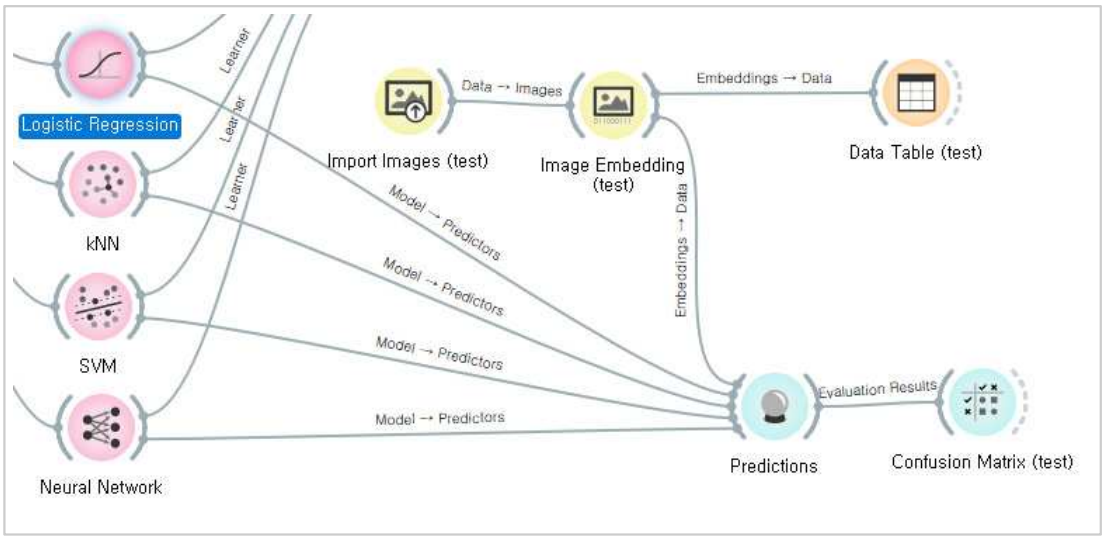
테스트 데이터도 같은 방법으로 이미지 임베딩을 수행하고 데이터 테이블을 보면 수치화된 데이터 테이블을 확인할 수 있다. 테스트 데이터도 폴더로 구분했기 때문에 정답 레이블이 있는 데이터이다.



[그림 1-24] 테스트 데이터 이미지 임베딩

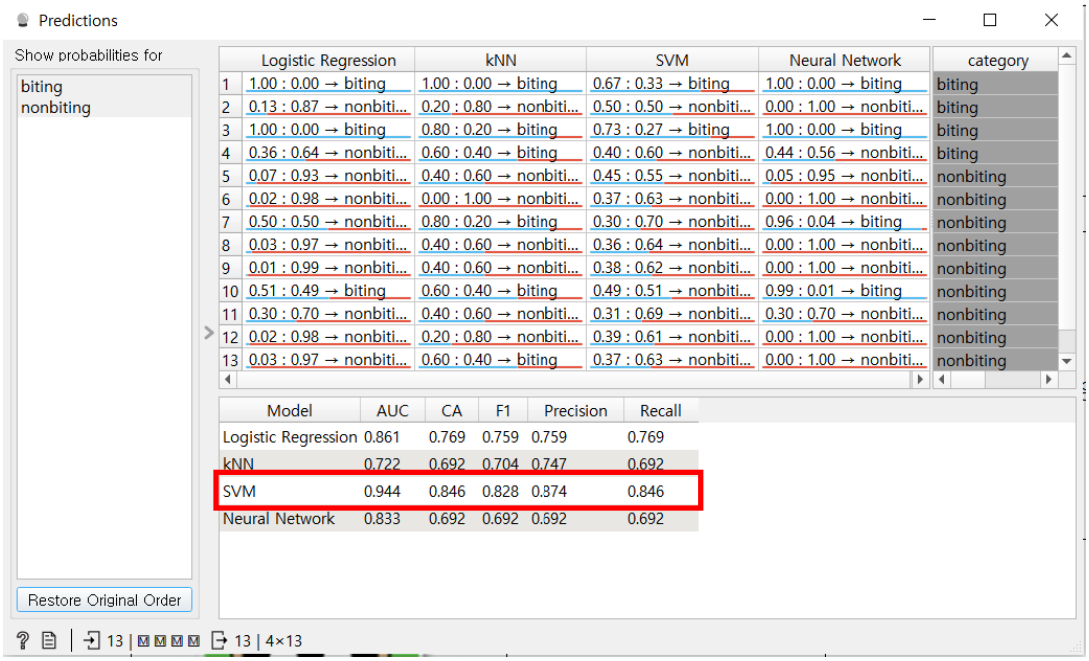
③ 테스트 데이터로 성능 평가하고 예측하기

앞서 모델링에서 상대적으로 성능이 좋은 Logistic Regression을 비롯한 4가지 모델과 테스트 데이터를 Predictions 위젯에 연결한다. 훈련에 사용하지 않는 데이터를 얼마나 잘 예측하였는지 Predictions의 예측 결과를 통해 확인할 수 있다.



[그림 1-25] 테스트 데이터로 성능 평가하고 예측하기

Prediction 위젯으로 성능 평가하고 예측한 결과를 나타내면 다음과 같다. 테스트 데이터로 성능을 평가한 결과 Neural Network보다 SVM의 성능이 높게 나왔다. 기계학습 모델은 훈련에 사용하지 않은 데이터에 얼마나 잘 예측할 수 있는지 성능평가를 하여 일반화하는 것이 중요하다. 따라서 테스트 데이터로 평가하여 좋은 성능을 보여준 SVM 모델을 선택하는 것이 바람직하다.



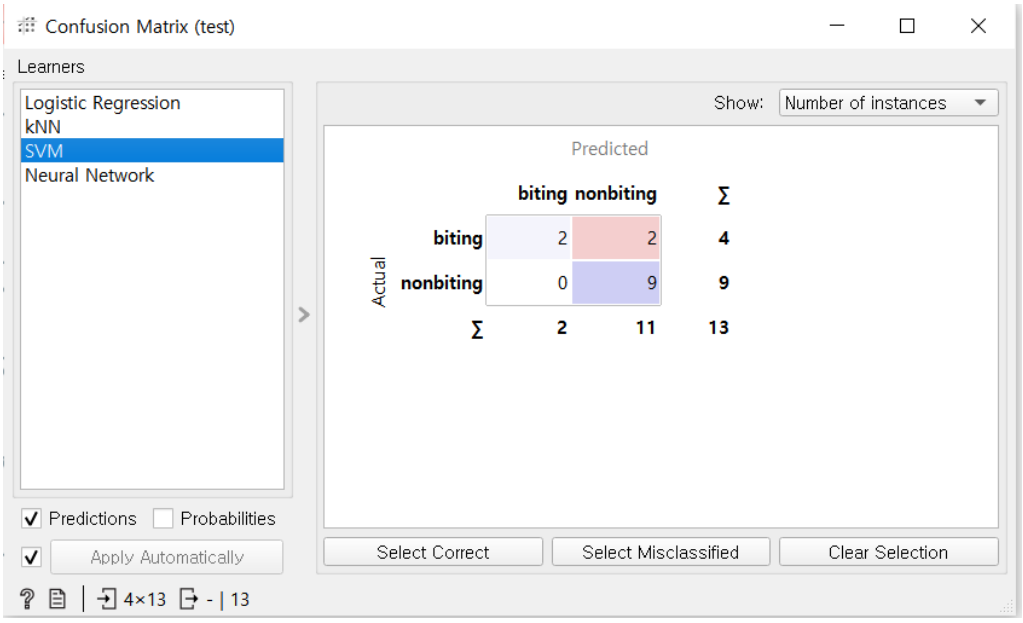
	Logistic Regression	kNN	SVM	Neural Network	category
1	1.00 : 0.00 → biting	1.00 : 0.00 → biting	0.67 : 0.33 → biting	1.00 : 0.00 → biting	biting
2	0.13 : 0.87 → nonbiti...	0.20 : 0.80 → nonbiti...	0.50 : 0.50 → nonbiti...	0.00 : 1.00 → nonbiti...	biting
3	1.00 : 0.00 → biting	0.80 : 0.20 → biting	0.73 : 0.27 → biting	1.00 : 0.00 → biting	biting
4	0.36 : 0.64 → nonbiti...	0.60 : 0.40 → biting	0.40 : 0.60 → nonbiti...	0.44 : 0.56 → nonbiti...	biting
5	0.07 : 0.93 → nonbiti...	0.40 : 0.60 → nonbiti...	0.45 : 0.55 → nonbiti...	0.05 : 0.95 → nonbiti...	nonbiting
6	0.02 : 0.98 → nonbiti...	0.00 : 1.00 → nonbiti...	0.37 : 0.63 → nonbiti...	0.00 : 1.00 → nonbiti...	nonbiting
7	0.50 : 0.50 → nonbiti...	0.80 : 0.20 → biting	0.30 : 0.70 → nonbiti...	0.96 : 0.04 → biting	nonbiting
8	0.03 : 0.97 → nonbiti...	0.40 : 0.60 → nonbiti...	0.36 : 0.64 → nonbiti...	0.00 : 1.00 → nonbiti...	nonbiting
9	0.01 : 0.99 → nonbiti...	0.40 : 0.60 → nonbiti...	0.38 : 0.62 → nonbiti...	0.00 : 1.00 → nonbiti...	nonbiting
10	0.51 : 0.49 → biting	0.60 : 0.40 → biting	0.49 : 0.51 → nonbiti...	0.99 : 0.01 → biting	nonbiting
11	0.30 : 0.70 → nonbiti...	0.40 : 0.60 → nonbiti...	0.31 : 0.69 → nonbiti...	0.30 : 0.70 → nonbiti...	nonbiting
12	0.02 : 0.98 → nonbiti...	0.20 : 0.80 → nonbiti...	0.39 : 0.61 → nonbiti...	0.00 : 1.00 → nonbiti...	nonbiting
13	0.03 : 0.97 → nonbiti...	0.60 : 0.40 → biting	0.37 : 0.63 → nonbiti...	0.00 : 1.00 → nonbiti...	nonbiting

Model	AUC	CA	F1	Precision	Recall
Logistic Regression	0.861	0.769	0.759	0.759	0.769
kNN	0.722	0.692	0.704	0.747	0.692
SVM	0.944	0.846	0.828	0.874	0.846
Neural Network	0.833	0.692	0.692	0.692	0.692

[그림 1-26] 테스트 데이터로 성능 평가와 모델 예측

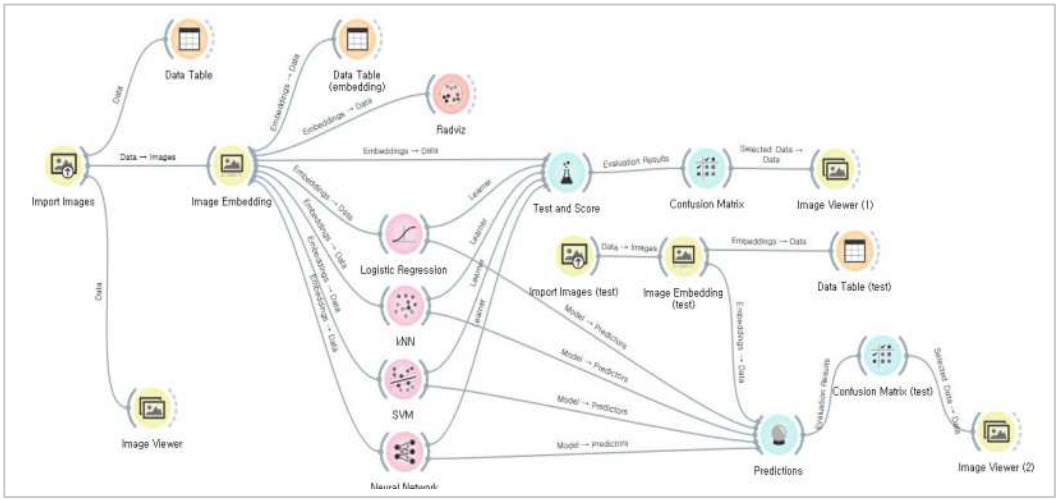
SVM 모델은 13개의 테스트 데이터 중에 2개가 잘 못 분류하였다.

이는 biting 데이터의 수가 nonbiting에 비해서 적기 때문에, 테스트 정확도가 떨어지는 것으로 해석할 수 있다. 데이터 수를 늘리고 두가지 카테고리 데이터의 수를 비슷하게 한다면 성능을 개선할 수 있을 것이다.



[그림 1-27] SVM 모델의 혼동 행렬

원숭이 데이터를 분류하는 기계학습 모델을 구현하는 과정은 다음 그림과 같다.

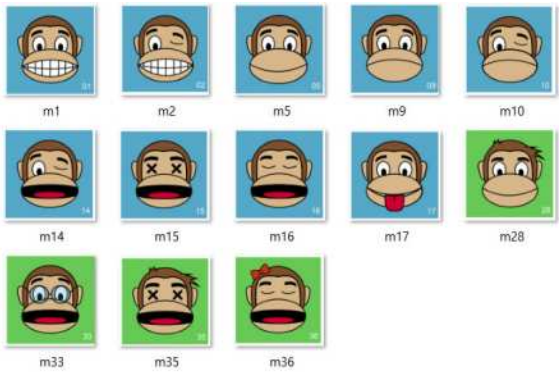
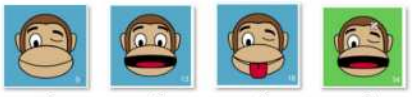


SVM 모델은 훈련할 때 성능이 상대적으로 좋지 않았지만 새로운 데이터를 입력했을 때 좋은 성능을 보이는 일반화된 모델을 선택하는 것이 바람직하다.

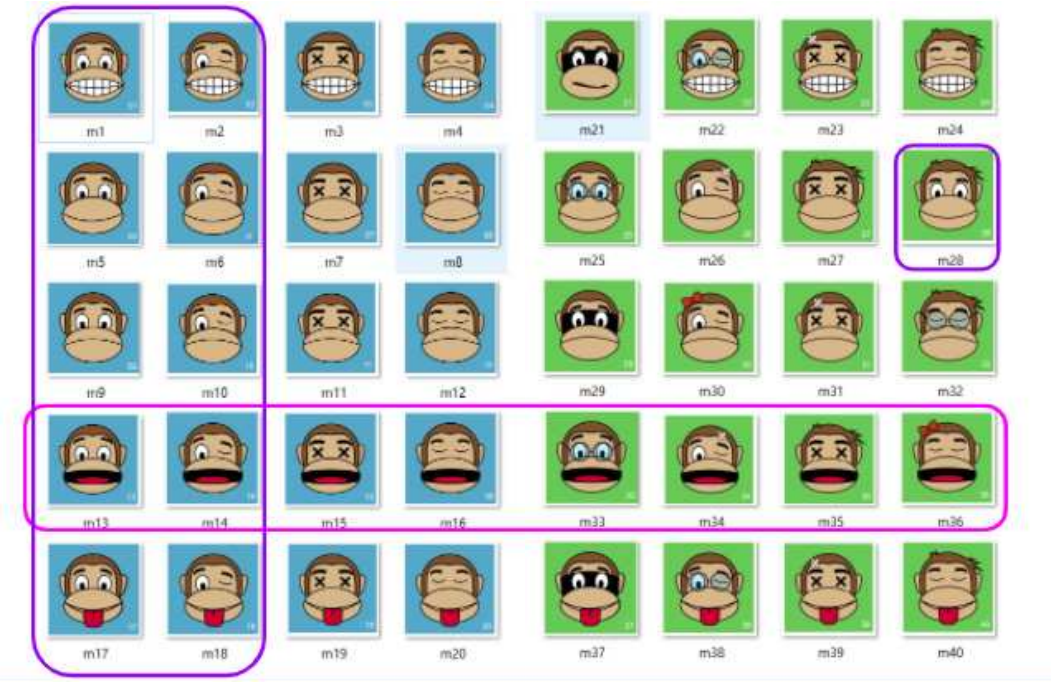
정리하기

동물원에 있는 원숭이가 무는지 물지 않는지 분류하는 인공지능을 만들기 위하여, 원숭이 데이터를 훈련 데이터와 테스트 데이터로 분리하고, 무는 원숭이와 물지 않는 원숭이를 분류할 수 있는 기계학습 모델을 만들었다. 모델 학습의 결과 SVM 모델의 성능이 가장 우수하였다. 이 모델을 이용하면 새로운 원숭이 이미지를 입력했을 때 무는 원숭이 인지 물지 않는 원숭이인지 예측할 수 있을 것이다.

원숭이 데이터 세트를 한눈에 살펴보면 그림과 같다.

	훈련 데이터	테스트 데이터
biting		
nonbiting		

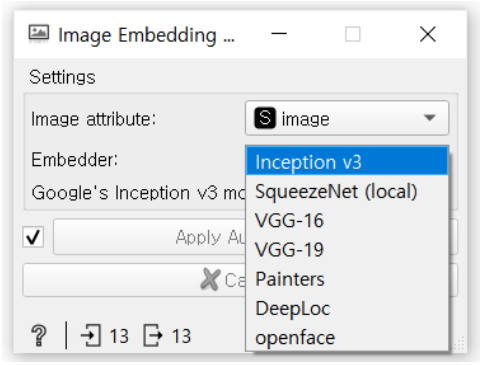
데이터에 숨어 있는 비밀 특성을 살펴보면 입을 벌리고 있거나 액세서리를 착용하지 않고 두 눈을 뜨고 있거나, 한쪽 눈을 감고 있는 특징이 있다.



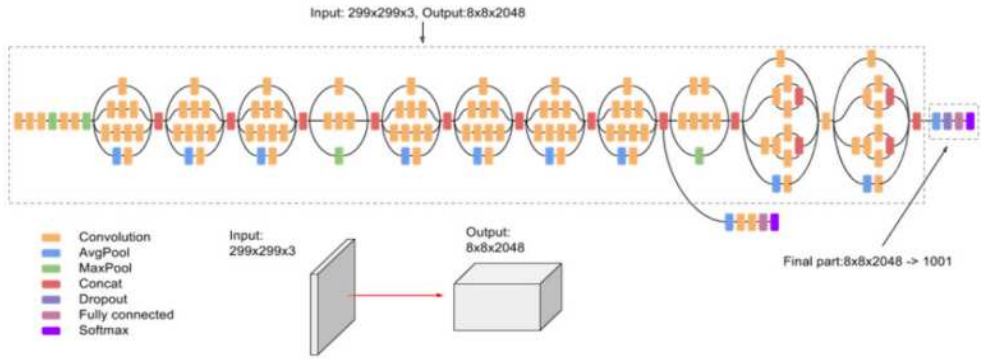
[그림 1-28] 무는 원숭이와 물이 않는 원숭이의 특징

AI 더 알아보기

◆ 사전 훈련된(pre-trained) 모델



이미지 임베딩은 이미지 데이터를 벡터화된 수치로 변환하기 위해 이미지 임베더를 사용한다. 이미지 임베더는 사전 훈련된 모델들이다. 학습 대상 이미지 데이터가 한정적일 때, 반복 학습하면 과적합 발생하여 정확도가 높은 모델을 만들기 어렵다. 거대한 인공신경망의 학습을 진행할 경우 컴퓨터 자원과 많은 시간이 필요하다. 이러한 문제를 해결하기 위한 방법으로 사전 훈련된 신경망을 사용할 수 있다. 훈련된 신경망은 대량의 데이터 세트에서 미리 학습된 신경망으로, 이미지의 일반적인 특성을 추출한다.



[그림 1-29] Inception V3 모델의 구조
(출처:https://cloud.google.com/tpu/docs/inception-v3-advanced?hl=ko)

Inception V3 모델은 구글에서 만든 모델인데 2014년 IRSVRC 대회에서 1등을 차지한 모델 GoogLeNet을 응용한 버전이다. Inception V3의 오류율은 3.6%정도로 사람의 오류율 5%보다 낮다.

[참고 문헌]

Stefan Seegerer, Annabel Lindner(2019), Classification with Decision Tree. https://www.aiunplugged.org

교차 검증.
https://towardsdatascience.com/cross-validation-k-fold-vs-monte-carlo-e54df2fc179b

Inception V3의 구조. 2021.9.30. 검색
https://cloud.google.com/tpu/docs/inception-v3-advanced?hl=ko. 2021.10.10.검색

MEMO

02. 개인에 관한 신상정보로 소득 정도를 예측할 수 있을까?

구미산동고등학교 교사 황 은 아

학습 진행 과정

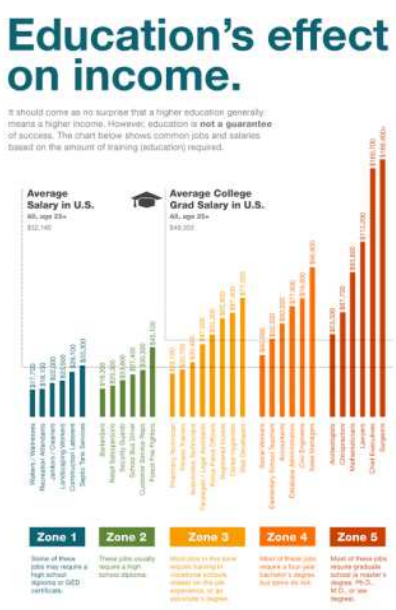
1단계	데이터 준비	- 수집: 캐글「Korea Income and Welfare」 - 데이터 편집: 데이터 속성명 추가
2단계	데이터 불러오기	- 학습 데이터 불러오기 - 데이터의 속성별 Role(역할) 설정하기
3단계	데이터 시각화	- 시각화 도구을 이용한 데이터 탐색 - 사용한 시각화 도구: Rank, Correlations, Scatter Plot
4단계	속성 추출	- 데이터 시각화 결과 또는 Rank로 주요 속성 추출하기
5단계	모델 학습	- 데이터 나누기 - 회귀를 이용한 모델 학습 - 사용된 알고리즘: Linear Regression, kNN, Random Forest
6단계	성능 평가	- test and score를 이용한 성능 평가
7단계	예측	- Prediction을 이용한 테스트 데이터로 예측하기

학습 내용 요약

데이터 종류	문제 유형	사용된 학습 알고리즘	성능 평가 도구
정형 데이터(수치형)	예측(회귀)	Linear Regression kNN Random Forest	test and score

문제 상황

사람은 누구나 경제적으로 풍족한 삶을 꿈꾼다. 행복한 삶을 살기 위해서 가장 중요한 조건이 무엇이냐고 묻는다면 대부분 돈을 먼저 말하게 될 것이다. 이처럼 돈은 우리의 행복한 삶을 위한 물질적 조건의 하나로서 누구나 많은 돈을 벌기를 바라고, 풍족한 삶을 영위하기 위해 어려움 없이 소비하기를 원한다.



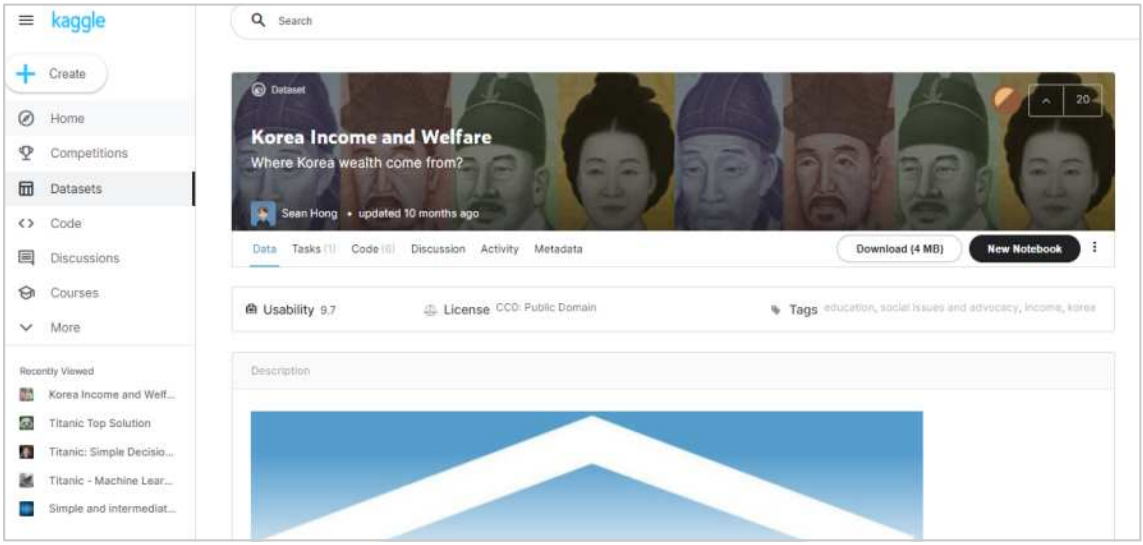
이러한 돈, 즉 개인의 소득은 어떤 요인을 통해서 결정되는 것일까? 우리 사회는 오래 전부터 교육을 통해서 획득할 수 있는 다양한 가치들에 대해 인식하고 있었다. 특히 급속한 경제 성장과 사회 발전을 이루는 과정 속에서 부족한 물적 자원보다는 인적자원을 통해 소득을 창조할 수 있다고 믿어 왔다. 또한 고학력과 학벌이 개인의 소득에 결정적인 영향을 미친다고 인식하고 있어 자녀에 대한 교육에 관심과 투자를 아끼지 않았다.

위의 그래프와 같이 미국에서 조사한 통계자료에 의하면 대학을 졸업한 사람의 평균 소득이 모든 근로자의 평균 소득보다 높다는 의미에서 학력이 소득에 영향을 미치기는 하지만 학력만이 소득을 결정하는 요소라고 보기에는 어렵다. 성별, 가족 구성원 수, 태어난 해, 결혼 유무 등 개인에 관한 여러 가지 정보가 소득에 직간접적으로 영향을 미칠 수 있기 때문이다. 개인에 관한 정보 중 어떤 것이 개인의 소득에 영향을 미치는지 그 상관관계를 알아보고, 개인에 관한 정보를 통해 개인의 소득에 관한 정도를 예측해보도록 하자.

01 데이터 준비하기

1 개인 소득 데이터 세트

캐글 사이트(https://kaggle.com)에 접속하면 기계학습을 위한 다양한 데이터 세트를 다운로드할 수 있다, 캐글(Kaggle)은 2010년 설립된 예측모델 및 분석 대회 플랫폼을 말한다. 기업 및 단체에서 데이터와 해결과제를 등록하면, 데이터 과학자들이 이를 해결하는 모델을 개발하고 경쟁하게 되는데, 기계학습과 관련하여 필요한 데이터를 손쉽게 얻을 수 있다. 이 곳에서 제공하는 한국인의 소득과 복지에 관한 데이터 세트를 이용하면 데이터의 속성으로 한국인의 소득 수준을 예측할 수 있는지 알아보자.



[그림 2-1] 캐글사이트에서 제공하는 「Korea Income and Welfare」 데이터 세트

[data]를 클릭하면 아래와 같이 데이터 파일을 다운로드 할 수 있다.

[그림 2-2] 캐글에서 제공하는 데이터 파일

제시된 2개의 파일 중 'Korea Income ans Welfare.csv 파일을 다운로드 받는다. 이 데이터는 2005년부터 2018년 사이 총 14년 동안 한국인의 소득, 지역, 종교 등에 관한

데이터이다. 총 92,877개의 데이터로 구성되어 있고 14개의 속성으로 구성되어 있다. 데이터를 다운로드 받아 열어보면 일부 속성에서 지나치게 결측값이 많은 것을 확인할 수 있다. 오렌지3를 사용하여 전처리하기 전에 기계학습에 사용하지 않을 속성 3가지 occupation, company_size, reasonnoneworker을 삭제한 후 파일을 수정하여 사용하려고 한다.

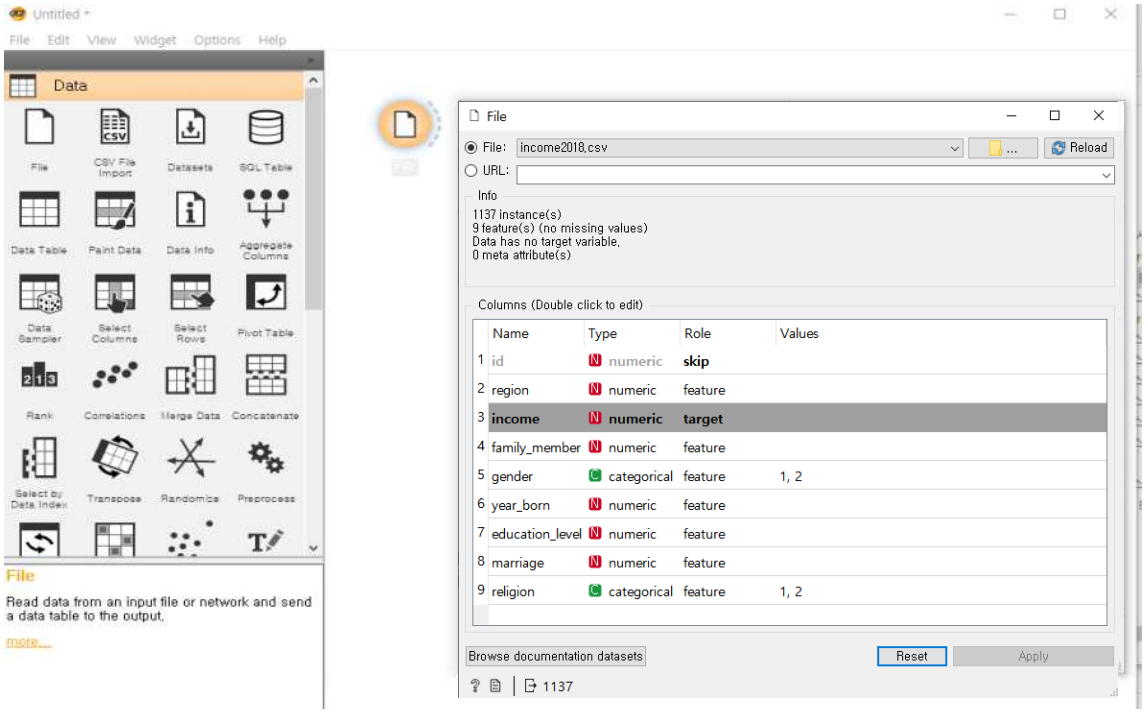
캐글에서 제공하는 데이터 정보는 다음과 같다.

연번	속성	이름	데이터	의미
1	id	순번	10101~98000701	연구대상자 아이디
2	year	연도	2015~2018	연구를 수행한 해
3	wave	차수	1~14	연구를 수행한 차수 2005년 1차부터 2018년 14차까지 시행
4	region	지역	1~7	1) 서울 2) 경기 3) 경남 4) 경북 5) 충남 6) 강원, 충북 7) 전라, 제주
5	income	소득	0 이상의 숫자값	연간 소득 M KRW (백만 원. 1100 KRW = 1 USD)
6	family_member	가족 구성원의 수	1 이상의 숫자값	가족 구성원의 수
7	gender	성별	1,2	1) 남성 2) 여성
8	year_born	태어난 해	1900이상의 숫자값	태어난 해
9	education_level	학력	1~9	1) 교육 없음(7세 미만) 2) 교육 없음(7세 이상) 3) 초등학교 4) 중학교 5) 고등학교 6) 전문대 7) 대학 학위 8) 석사 9) 박사
10	marriage	결혼상태	1~6	1) 해당 사항 없음(18세 미만) 2) 기혼 3) 사별 4) 별거 5) 미혼 6) 기타
11	religion	종교유무	1,2	1) 종교가 있습니다 2) 종교가 없습니다
12	occupation	직업의 종류	1 이상의 숫자값	149가지의 직업 종류 별도의 직업 종류 코드에 관한 파일을 제공함
13	company_size	회사규모	1~11	회사의 규모가 클수록 높은 값을 가짐
14	reasonnoneworker	기타사유	1~11	1) 능력 없음 2) 병역 중 3) 학교에서 공부 4) 학교 준비 5) 취업 준비 6) 가사도우미 7) 집에서 아이 돌보기 8) 간호 9) 경제 활동 포기 10) 일할 의사가 없음 11) 기타

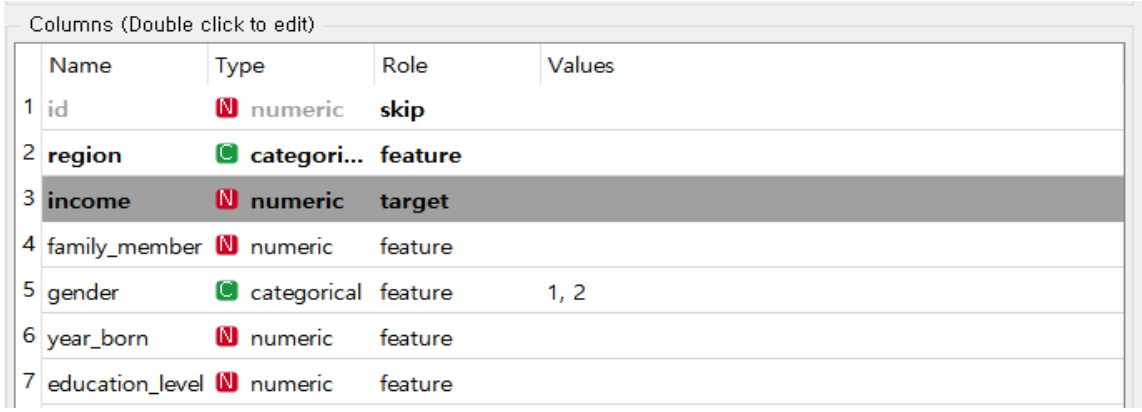
데이터를 살펴보면 2015년부터 연구대상자의 소득, 지역, 가족의 수 등의 데이터를 14차까지 지속적으로 수집한 데이터이다. 이번 학습에서는 개인의 소득을 예측해보는 과정으로 제시된 모든 데이터를 사용할 필요는 없다. 가장 최근의 데이터인 2018년 데이터만을 사용하여 학습해보도록 하자.

2 데이터 불러오기

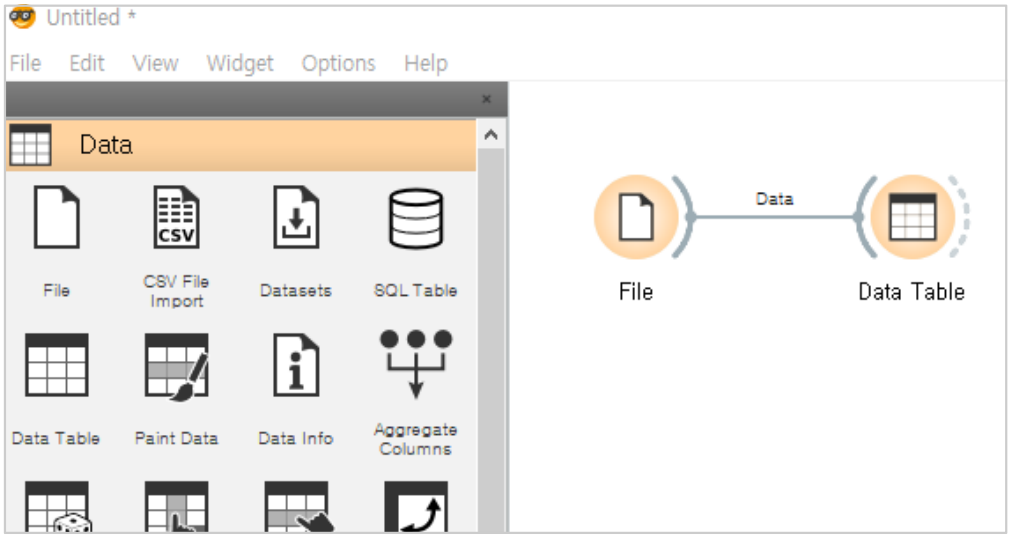
① Orange3을 실행한 후 왼쪽 위젯 팔레트에서 파일 위젯을 캔버스에 갖다 놓는다. 캔버스에 갖다 놓은 파일 위젯을 더블 클릭하여 개인 소득 데이터 세트(2018년) 파일(income2018.csv)을 불러온다.



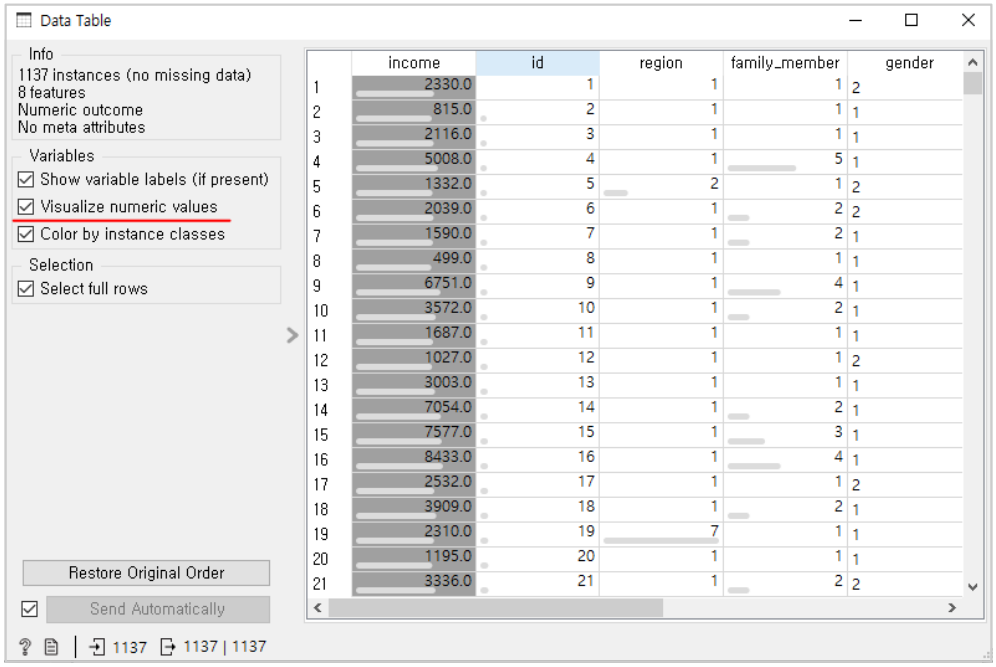
② columns에서 데이터 형식과 데이터의 Role(역할)을 확인한다. 이 중에서 income(소득)이 결과값이므로 target으로 지정하고 id를 제외한 데이터는 원인이므로 feature로 지정한다.



③ 데이터를 잘 가져왔는지 확인하기 위해 위젯 팔레트에서 Data Table 위젯을 캔버스에 드래그한다. File의 오른쪽 괄호를 드래그하고 팝업에서 Data Table을 선택해서 데이터 테이블을 추구한다.



④ Data Table을 더블 클릭해 살펴보면 income이 결과이고, 그 옆의 다른 열은 원인이다. 이처럼 숫자 데이터가 있을 때는 Visualize numeric values 옵션을 체크하면 값의 크기를 시각적으로 파악하기 수월하다.



	income	id	region	family_member	gender
1	2330.0	1	1	1	2
2	815.0	2	1	1	1
3	2116.0	3	1	1	1
4	5008.0	4	1	5	1
5	1332.0	5	2	1	2
6	2039.0	6	1	2	2
7	1590.0	7	1	2	1
8	499.0	8	1	1	1
9	6751.0	9	1	4	1
10	3572.0	10	1	2	1
11	1687.0	11	1	1	1
12	1027.0	12	1	1	2
13	3003.0	13	1	1	1
14	7054.0	14	1	2	1
15	7577.0	15	1	3	1
16	8433.0	16	1	4	1
17	2532.0	17	1	1	2
18	3909.0	18	1	2	1
19	2310.0	19	7	1	1
20	1195.0	20	1	1	1
21	3336.0	21	1	2	2

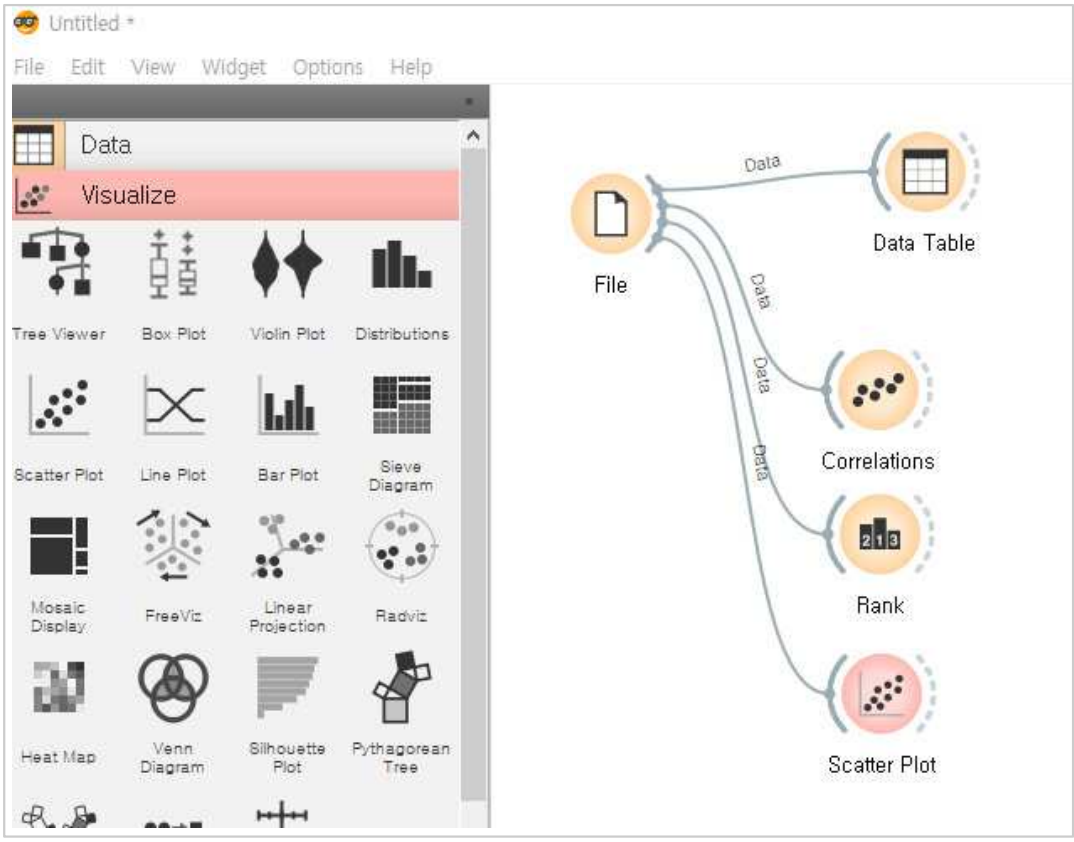
02 데이터 탐색하자

개인의 소득 수준에 결정적으로 영향을 미치는 속성은 무엇일까? 어떤 속성이 소득의 정도를 예측하는 기계학습에 많은 영향을 미치는지 알아보기 위해 데이터를 시각화해보자.

이러한 과정을 통해 기계학습에 영향을 주는 핵심 데이터 속성을 추출할 수 있는데 이러한 과정을 데이터 탐색이라고 한다. 데이터 탐색은 분석 대상 데이터를 다양한 관점으로 살펴보고 그 특성을 이해하는 과정으로 좋은 분석 모델을 만들기 위해 반드시 필요한 과정이다. 다양한 관점으로 깊이 있게 데이터를 파악하기 위해 주로 히스토그램과 산점도 등 데이터를 시각화하여 분석하게 된다.

1 데이터 시각화 하기

- 데이터를 시각화하기 위해서 속성간의 상관관계를 파악할 수 있는 Rank, Correlations, Scatter Plot를 사용하려고 한다.
- 각각 Data와 Visualize 위젯 팔레트에서 해당 위젯을 찾아 캔버스로 드래그한 후 파일의 오른쪽 곡선을 드래그앤 드롭하여 연결한다.

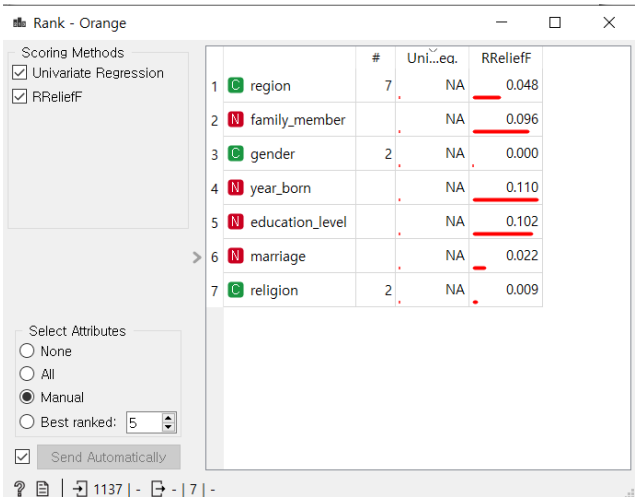


2 Rank로 나타내기

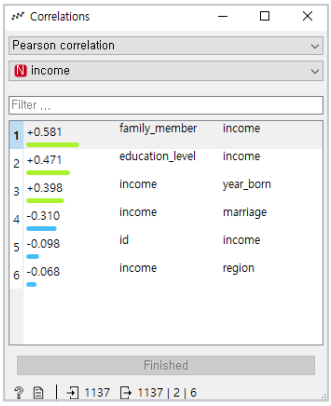
- Rank는 회귀 또는 분류시 데이터 속성을 순위로 분석해준다.
- 회귀에서는 Univar.reg.(일변량 회귀)와 RReliefF값을 제공한다.

> • 일변량 회귀 : 단일 변수에 대한 선형 회귀
> • RReliefF : 두 인스턴스의 예측(클래스) 값 사이의 상대 거리.

- ReliefF는 유사한 데이터 속성의 클래스(타켓)을 구별하는 속성의 기능을 말한다.
- 데이터 위젯의 Rank를 사용하여 속성을 분석해보면 year_born이 가장 높은 값으로 예측에 큰 영향을 미치며 education_level, family_member순으로 결과에 영향을 미친다는 것을 확인할 수 있다.
- 그 외 속성은 개인 소득에 미치는 영향이 미미하다는 것을 확인할 수 있다.



3 Correlations로 나타내기



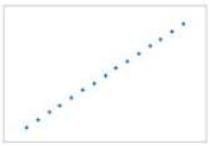
- 모든 속성의 쌍별로 상관관계를 계산할 수 있다.
- 데이터를 입력받아 처리하면 속성 간의 상관 점수를 데이터테이블로 제공한다.
- 상관관계는 데이터 세트의 모든 기능 쌍에 대하여 Pearson 또는 Spearman 상관 점수를 계산하여 결과를 알려주는데 원하는 상관계수는 창의 상단에서 직접 선택할 수 있다.
- 상관 관계를 분석해보면 앞에서 분석한 결과와 마찬가지로 개인의 소득(income)에는 education_level, family_member이 상관계수의 절대값이 0.4 이상으로 영향을 미친다는 것을 알 수 있다.
- year_born의 상관계수도 -0.0398로 반올림을 할 경우 절대값이 0.4보다 높으므로 개인의 소득(iocome)에 영향을 미친다는 것을 알 수 있다.

AI의 기초 알아보기

◆ 상관의 정의

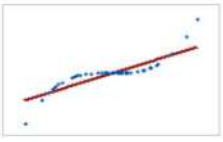
- 상관 계수는 두 변수가 함께 변화하는 경향이 있는 범위를 측정한다. 이 계수는 상관 관계의 정도와 방향을 해준다.
- Pearson 곱적률 상관
  - Pearson 상관은 두 계량형 변수 사이의 선형 관계를 평가한다. 한 변수의 변화가 다른 변수의 변화에 비례적으로 연관되어 있는 경우 선형 관계가 있다.
  - 예를 들어, 생산 설비의 온도 증가가 초콜릿 코팅의 두께 변화와 연관성이 있는지 여부를 평가하기 위해 Pearson 상관을 사용할 수 있다.
- Spearman 순위 상관
  - Spearman 상관은 두 계량형 변수 또는 순서형 변수 사이의 단순 관계를 평가한다. 단순 관계에서 두 변수는 함께 변화하는 경향이 있지만 반드시 일정한 비율로 변화하는 것은 아니며, Spearman 상관 계수는 원시 데이터가 아니라 각 변수에 대해 순위를 매긴 값을 기반으로 한다.
  - Spearman 상관은 종종 순서형 변수가 포함된 관계를 평가하기 위해 사용되는데 예를 들어, 직원들이 테스트 연습을 완료하는 순서가 고용된 개월 수와 관련이 있는지 여부를 평가하기 위해 Spearman 상관을 사용할 수 있다.

◆ Pearson 및 Spearman 계수의 비교



Pearson = +1, Spearman = +1

Pearson 및 Spearman 상관 계수는 -1에서 +1 범위의 값입니다. Pearson 상관 계수가 +1이 되도록 하기 위해 한 변수가 증가하면 다른 변수가 일정한 양만큼 증가합니다. 이 관계는 완전한 선을 형성합니다. 이 경우 Spearman 상관 계수도 +1입니다.



Pearson = +0.851, Spearman = +1

한 변수가 증가하면 다른 변수가 증가하지만 양이 일정하지 않은 관계인 경우, Pearson 상관 계수는 양수이지만 +1보다 작습니다. 이 경우 Spearman 계수는 여전히 +1입니다.

Pearson = −0.093, Spearman = −0.093

관계가 랜덤이거나 존재하지 않을 경우 두 상관 계수 모두 0에 가깝습니다.

Pearson = −1, Spearman = −1

관계가 감소하는 관계에 대한 완전한 선인 경우 두 상관 계수 모두 -1입니다.

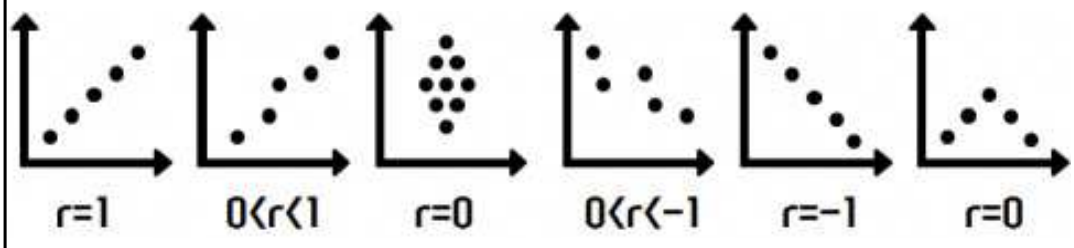
Pearson = −0.799, Spearman = −1

한 변수가 감소하면 다른 변수가 증가하지만 양이 일정하지 않은 관계인 경우, Pearson 상관 계수는 음수이지만 -1보다 큽니다. 이 경우 Spearman 계수는 여전히 -1입니다.

- 상관 관계를 분석해보면 앞에서 분석한 결과와 마찬가지로 개인의 소득(income)에는 education_level, family_member이 상관계수의 절대값이 0.4 이상으로 영향을 미친다는 것을 알 수 있다.
- year_born의 상관계수도 -0.0398로 반올림을 할 경우 절대값이 0.4보다 높으므로 개인의 소득(iocome)에 영향을 미친다는 것을 알 수 있다.

◆ 상관관계

- 두 변수 사이의 상관관계의 정도를 나타내는 수치(계수)를 뜻한다.
- -1과 1 사이의 값을 가지며, 절대값이 1에 가까울수록 두 변수 간의 상관관계의 정도가 높은 것으로 볼 수 있다.
- 상관계수로 두 변수의 인과관계는 알 수 없지만 선형 관계를 파악할 수 있다.
- 가장 높은 상관관계의 상관계수는 1이고, 두 변수 간에 상관 관계가 전혀 없으면 상관계수는 0이다.

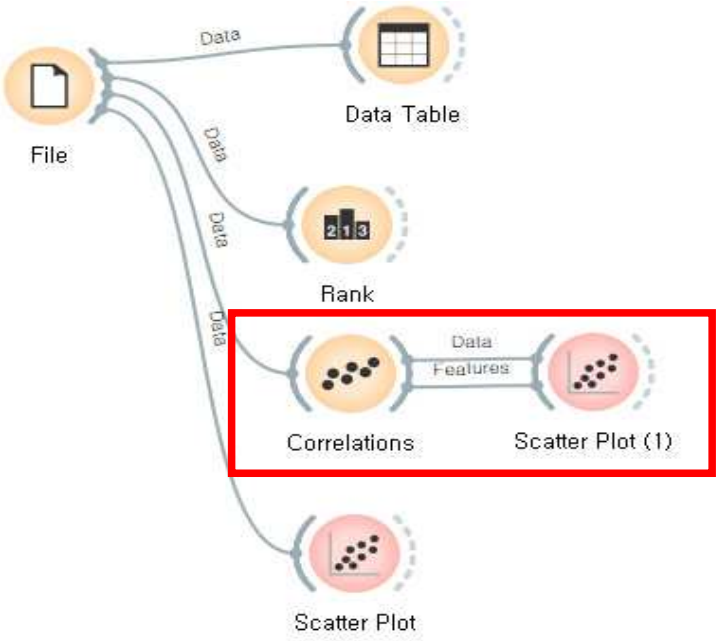


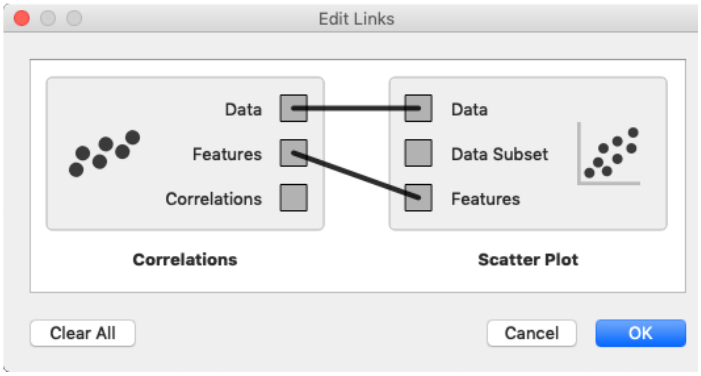
양의 상관관계가 강하다.	양의 상관관계가 있다.	상관관계가 없다.	음의 상관관계가 있다.	음의 상관관계가 강하다.	상관관계가 없다.

- 상관계수(r)의 범위에 따라 다음과 같이 해석하는데 상관계수 앞에 -가 붙으면 음의 상관관계를 가지게 된다.

0.0~0.2 : 상관관계가 거의 없다. 0.4~0.6 : 상관관계가 있다. 0.8~1.0 : 상관관계가 매우 높다.	0.2~0.4 : 상관관계가 낮다. 0.6~0.8 : 상관관계가 높다.

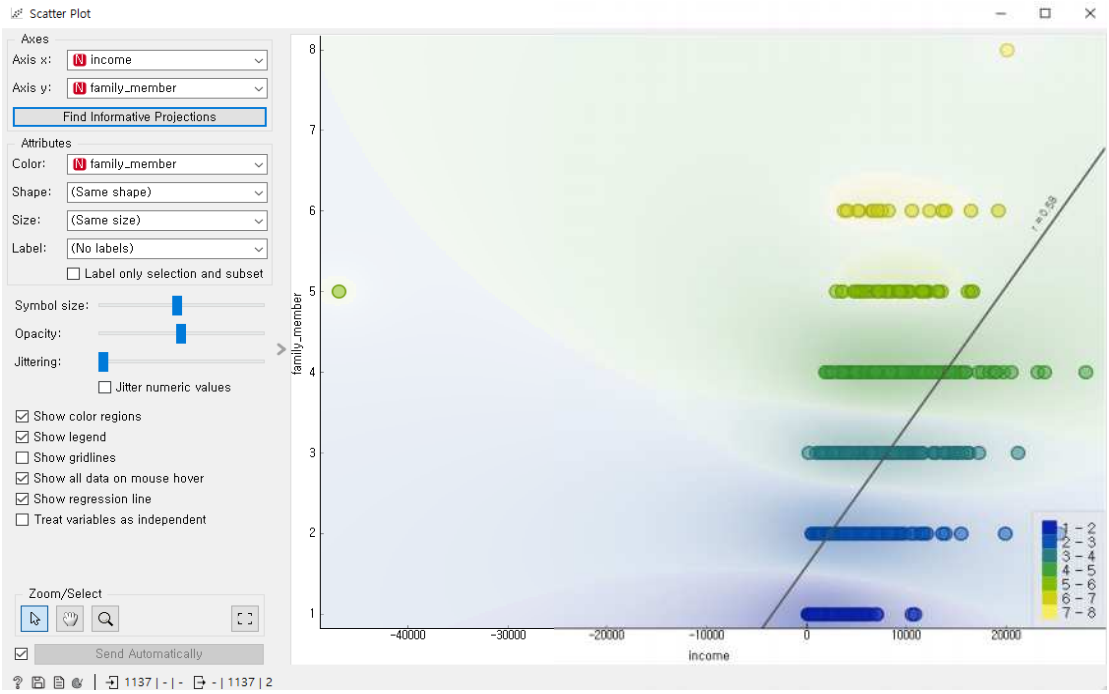
- Correlations에 Scatter Plot를 연결하여 특징간의 상관관계를 산점도로 확인할 수 있다.





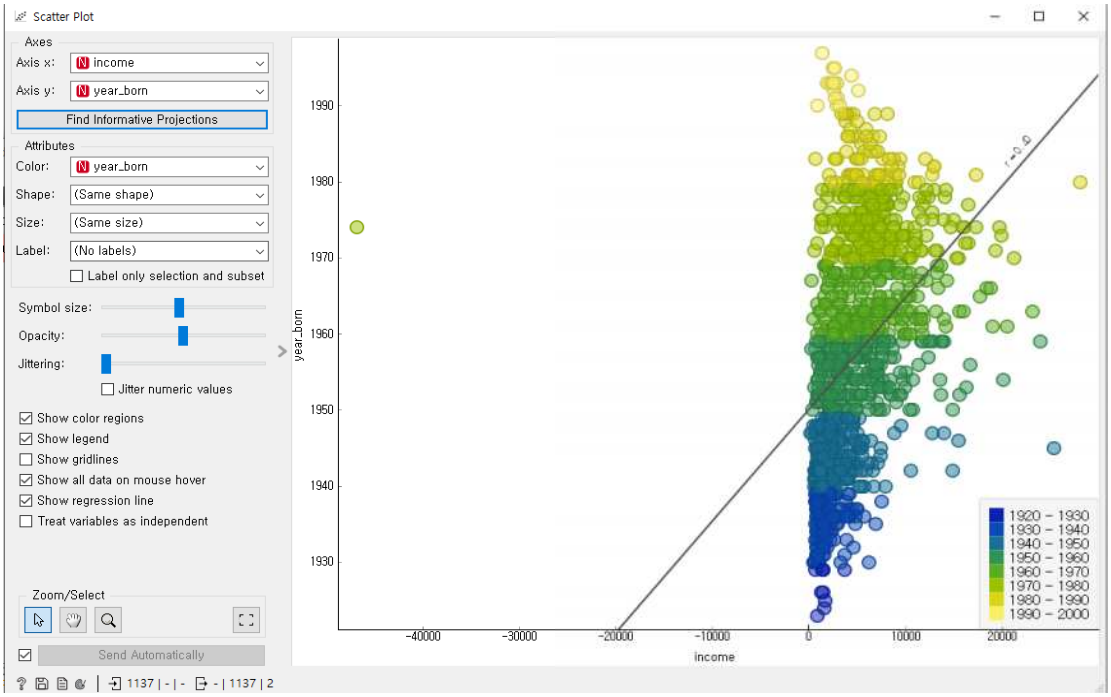
4 Scatter Plot로 나타내기

- Scatter Plot를 사용하여 2차원 산점도 시각화 자료를 얻을 수 있다.
- 데이터는 각각 가로축의 위치를 결정하는 x축 속성의 값과 세로축의 위치를 결정하는 y축 속성의 값을 갖는 점의 모음으로 표시된다.
- 위젯의 왼쪽에서 색상, 포인트 크기 및 모양, 축 제목, 최대 포인트 크기 및 지터링과 같은 그래프의 다양한 속성을 조정할 수 있다.
- Axis x를 개인의 소득(income)으로 설정하고 Axis y의 값을 다른 특징들로 변경해보면 두 변수 간의 상관계수를 그래프와 r계수의 값으로 확인해볼 수 있다.



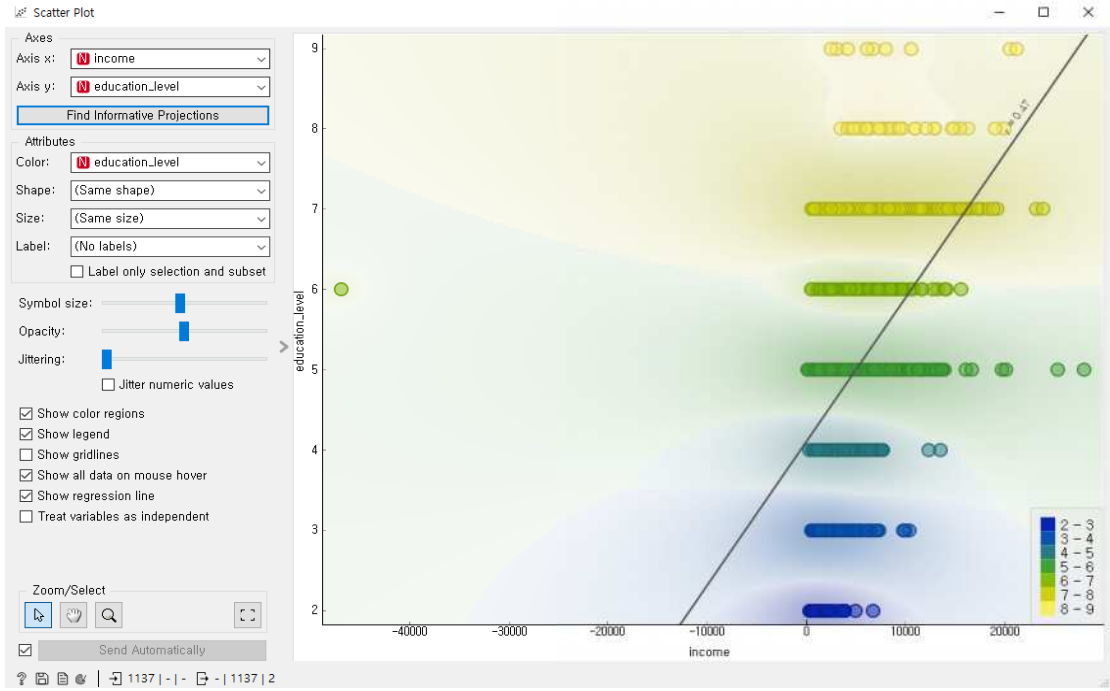
[그림 2-3] family_member의 분포 (R=0.58)

family_member는 상관계수 값이 0.58로 개인의 소득(income)과의 상관 관계가 있다.



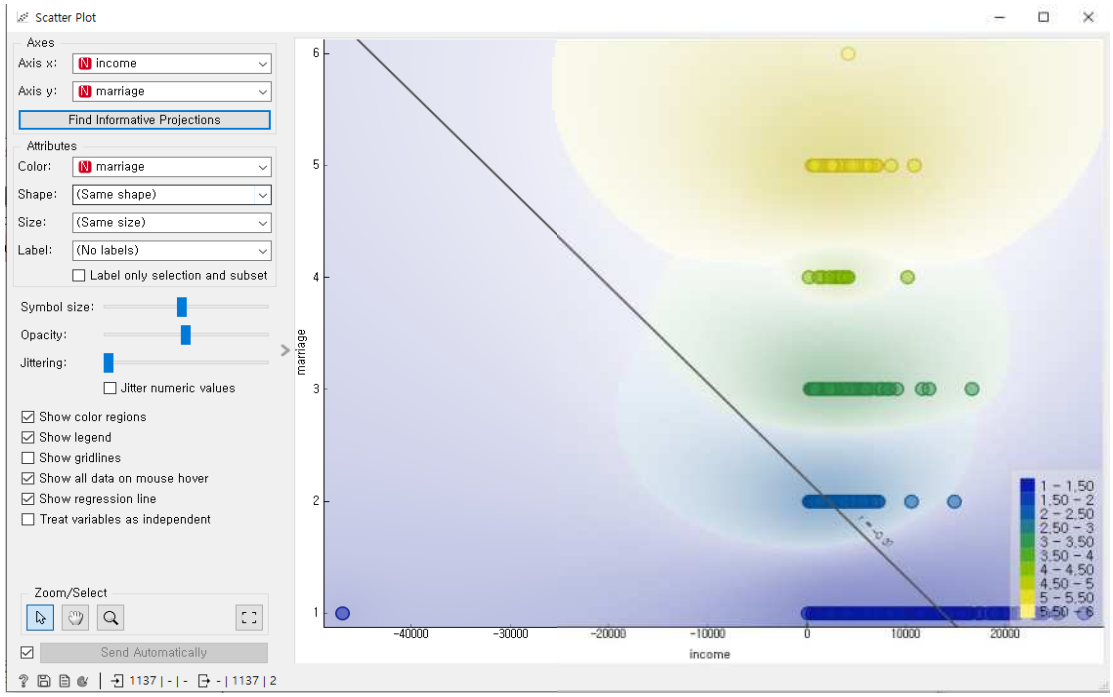
[그림 2-4] year_born의 분포 (R=-0.40)

year_born은 상관계수 값이 -0.40으로 개인의 소득(income)과의 상관 관계가 있다.



[그림 2-5] education_level의 분포 (R=0.47)

education_level은 상관계수값이 0.47로 개인의 소득(income)과의 상관 관계가 있다.



[그림 2-6] marridge의 분포 (R=-0.31)

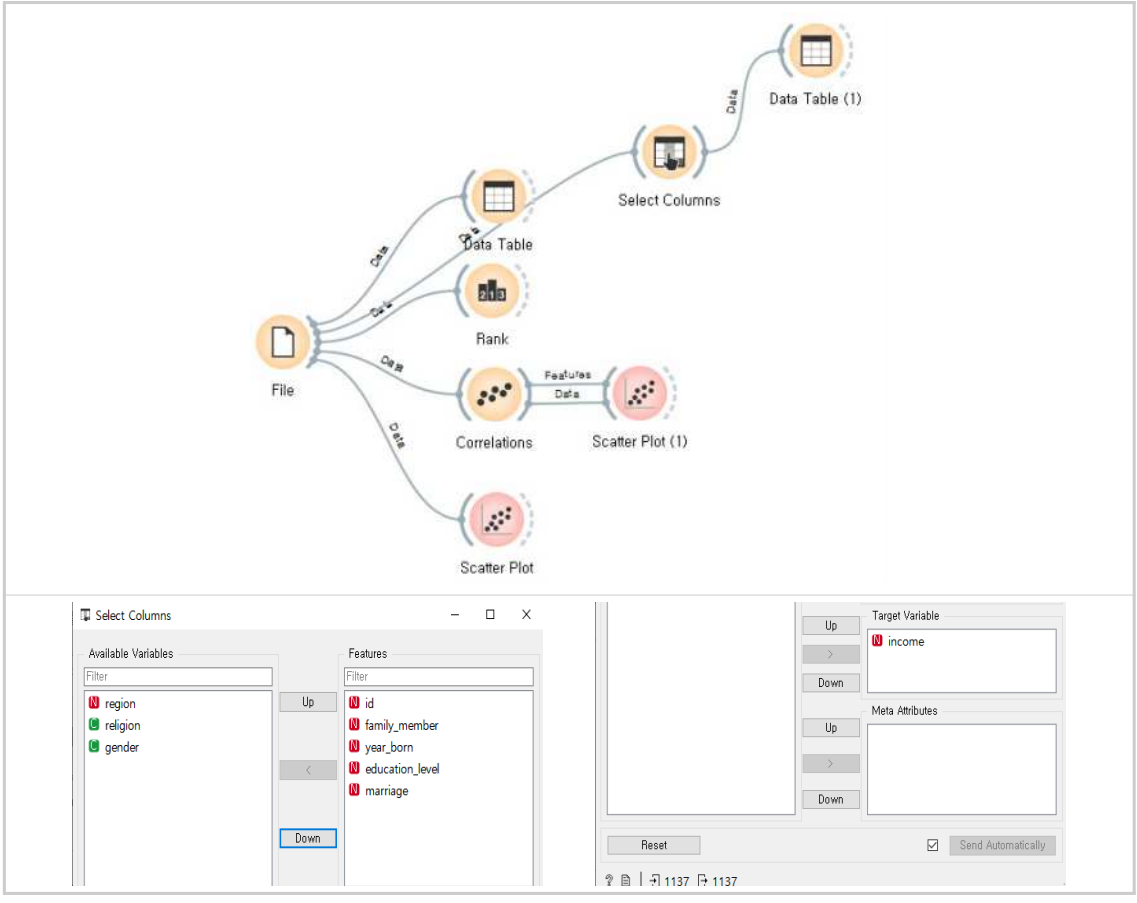
marridge는 상관계수값이 -0.31로 개인의 소득(income)과의 상관 관계가 매우 미미하다.

03 모델 학습하고 성능 평가하자

추출한 데이터 속성을 바탕으로, 기계학습 알고리즘과 데이터를 연결하여 모델 학습한다. 오렌지에서는 다양한 기계학습 알고리즘을 한꺼번에 연결하여 모델을 만들 수 있다. 여기서는 회귀에 자주 사용하는 Linear Regression, kNN, Random Forest를 이용하여 모델을 구성하였다.

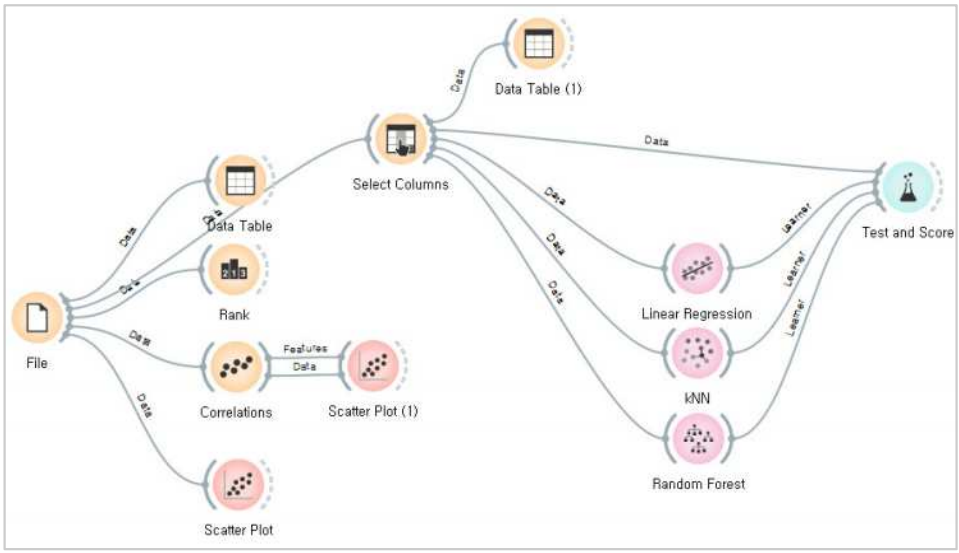
1 데이터 특성 선택하기

- 데이터 탐색을 통해 분석해 본 결과 데이터 속성 중 상관관계를 가지고 있는 것은 family_member, year_born, marridge, education_level이다. 반대로 개인 소득(income)과 상관관계를 가지고 있는 속성은 region, gender, religion이다.
- 입력 데이터 중 데이터 탐색의 결과를 토대로 타겟과 상관관계를 가지고 있는 속성만으로 학습을 하려고 한다.
- select columns을 통해 상관관계를 가지고 있는 속성만을 학습에 반영한다. 실제 데이터에서 필요한 컬럼만 추출이 되었는지 data table을 통해 확인할 수 있다.

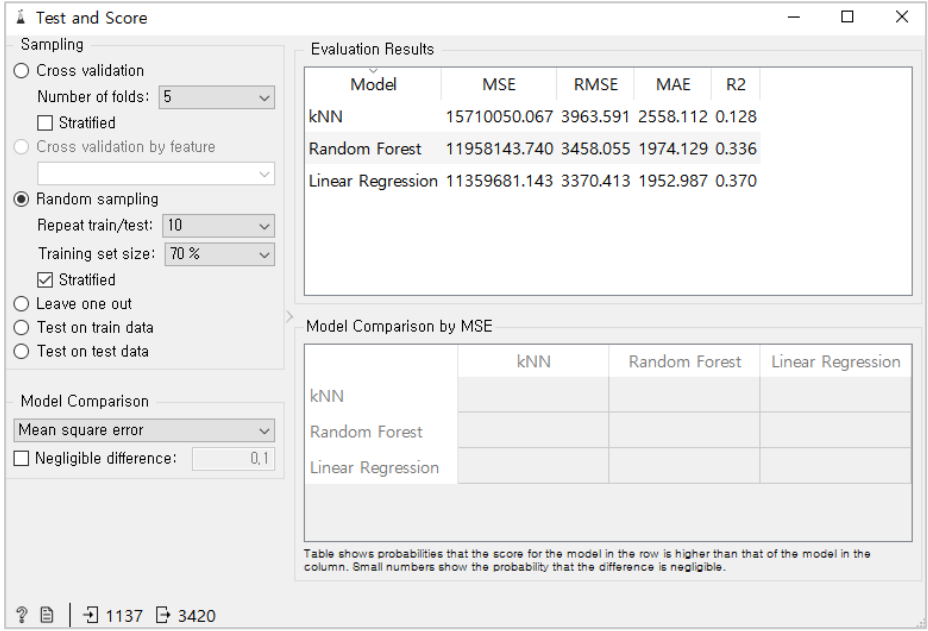


2 모델 성능 평가하기

- Test and Score를 이용하여 모델의 성능을 확인해보자.
- 사용할 모델은 Linear Regression, kNN, Random Forest이며 select Columns의 출력을 입력데이터로 사용한다.



- 샘플링 방식 중 Random Sampling은 전체 데이터를 섞어서 무작위로 훈련 데이터와 테스트 데이터를 분리한다. 또한 훈련과 테스트의 반복횟수를 설정할 수 있다. 테스트 데이터를 이용한 계산을 쉽게하기 위해 반복(repeat train/test)를 10회로 설정하였다.



- R2는 R-squared 즉 R의 제곱을 말하는데 우리는 이것을 회귀 제곱합 또는 총 제곱합이라고 한다. 다른 말로는 설명력, 결정계수라고 부른다.
- 독립변수가 종속변수에 얼마나 설명력을 가지는지 보여주는 수치인데, 1에 가까울수록 예측 정확도가 높다는 것을 의미한다.

r=예측 Variance / 실제값 Variance

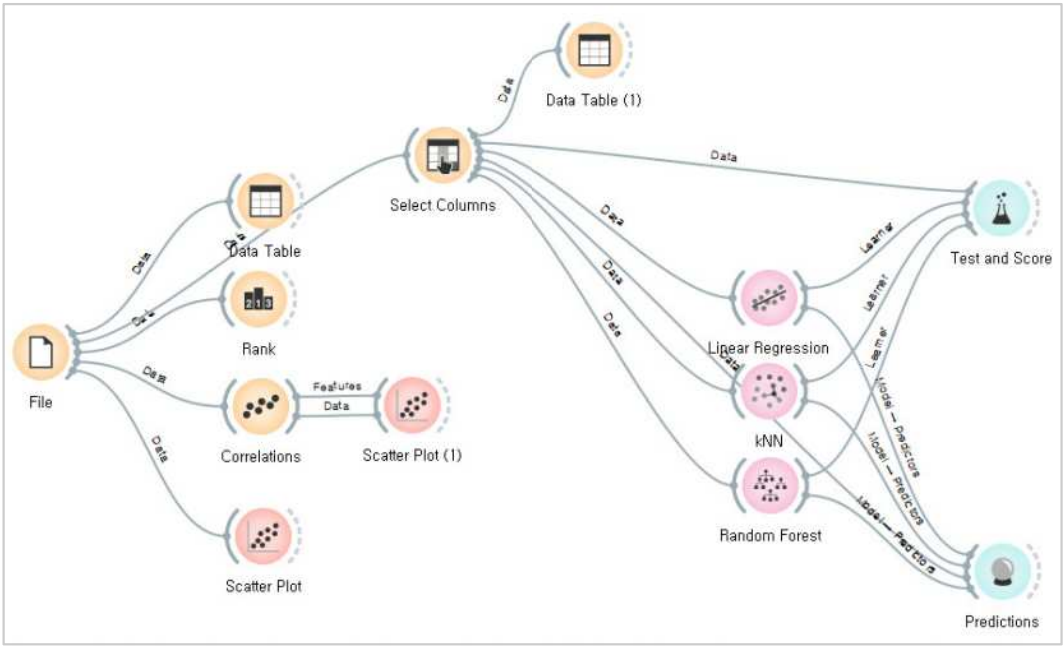
- 예를 들어 R2의 값이 0.5이면 50%의 설명력을 가졌다는 의미로 해석할 수 있다.
- MSE와 RMSE, MAE는 다음과 같이 오차에 관한 성능 지표이다. 실제값과 예측값 차이를 기준으로 오차에 대한 값으로 숫자가 낮을수록 성능이 좋다.

MSE(Mean Squared Error): 오차들의 제곱값 평균
RMSE(Root Mean Squared Error): 오차들의 제곱값 평균에 루트를 씌운 값
MAE(Mean Absolute Error): 오차들의 절대값 평균

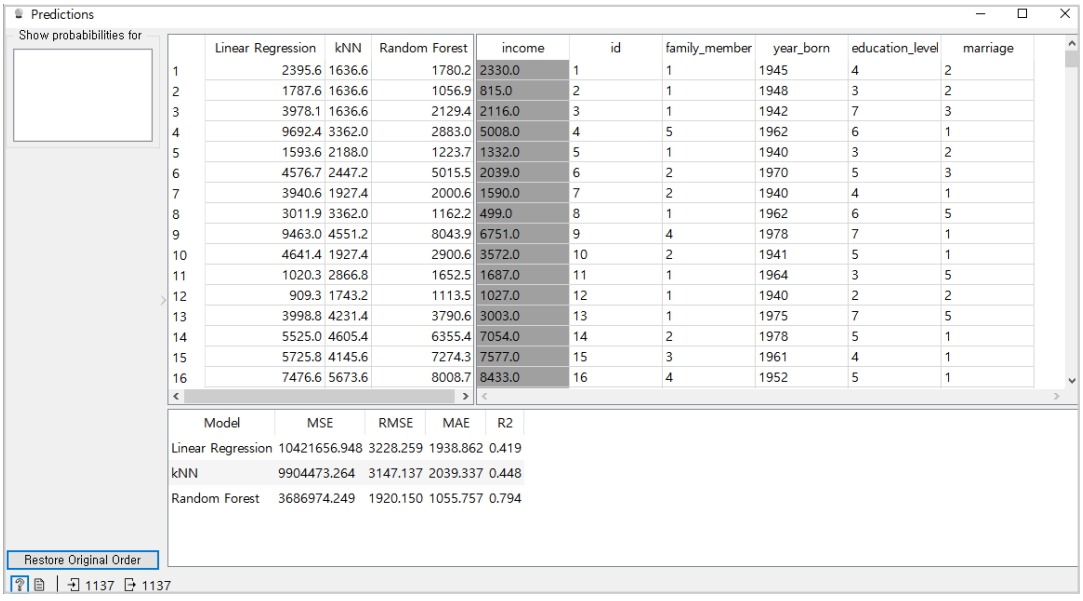
- 사용한 3가지 모델 Linear Regression, kNN, Random Forest 중 MSE, RMSE, MAE 수치가 낮고, R2 수치가 높은 모델이 예측 모델로서 가장 좋은 성능을 보인 것이다. 모든 평가 지표면에서 Linear Regression이 가장 좋은 성능을 보이고, 반면 k-NN이 가장 열악한 성능을 보이고 있다.
- 다만 원 데이터가 전체 데이터 중 2018년 소득에 대한 데이터만을 사용하여 성능에서 아쉬움이 남는다. 캐글에서 전체 데이터를 다운 받은 후 전처리를 통해 결측치를 제거하고 전체 데이터를 이용하여 학습한다면 더 나은 성능을 보일 수 있을 것이라 예상한다.

3 데이터 예측하기

- 이제 학습시킨 모델들을 이용하여 데이터를 예측해보자.
- Evaluate에서 Predictions를 가져오고 데이터를 입력해준다. 데이터는 다음과 같이 select columns의 출력을 입력데이터로 연결해준다.



- 앞에서 학습시킨 모델을 Predictions에 이어주면 학습한 모델에 기반하여 값을 다음과 같이 예측할 수 있다.



Predictions

Show probabilities for

	Linear Regression	kNN	Random Forest	income	id	family_member	year_born	education_level	marriage
1	2395.6	1636.6	1780.2	2330.0	1	1	1945	4	2
2	1787.6	1636.6	1056.9	815.0	2	1	1948	3	2
3	3978.1	1636.6	2129.4	2116.0	3	1	1942	7	3
4	9692.4	3362.0	2883.0	5008.0	4	5	1962	6	1
5	1593.6	2188.0	1223.7	1332.0	5	1	1940	3	2
6	4576.7	2447.2	5015.5	2039.0	6	2	1970	5	3
7	3940.6	1927.4	2000.6	1590.0	7	2	1940	4	1
8	3011.9	3362.0	1162.2	499.0	8	1	1962	6	5
9	9463.0	4551.2	8043.9	6751.0	9	4	1978	7	1
10	4641.4	1927.4	2900.6	3572.0	10	2	1941	5	1
11	1020.3	2866.8	1652.5	1687.0	11	1	1964	3	5
12	909.3	1743.2	1113.5	1027.0	12	1	1940	2	2
13	3998.8	4231.4	3790.6	3003.0	13	1	1975	7	5
14	5525.0	4605.4	6355.4	7054.0	14	2	1978	5	1
15	5725.8	4145.6	7274.3	7577.0	15	3	1961	4	1
16	7476.6	5673.6	8008.7	8433.0	16	4	1952	5	1

Model	MSE	RMSE	MAE	R2
Linear Regression	10421656.948	3228.259	1938.862	0.419
kNN	9904473.264	3147.137	2039.337	0.448
Random Forest	3686974.249	1920.150	1055.757	0.794

Restore Original Order

1137 1137

- Predictions 창을 더블클릭해서 열어보면 위와 같이 income을 어떻게 예측했는지 확인해볼 수 있다.
- 앞서 성능평가 결과를 확인했을 때 사용한 모델 중 Linear Regression이 가장 우수했던 것과 달리 Random Forest가 0.794로 실제 income의 데이터와 학습

한 모델로 예측한 값을 비교해보았을 때 오차가 가장 적은 것을 확인할 수 있다.

- 실제 테스트 데이터로 예측한 결과를 확인해보면 성능평가와 같이 3가지 모델 중 Random Forest이 R2의 값이 0.794로 가장 높은 값을 가지고 나머지 오차에 대한 지표인 MSE, RMSE, MAE가 가장 낮은 값을 가지는 것을 확인할 수 있다.

정리하기

개인의 소득에 영향을 미치는 요소에 대해 데이터를 수집하고 학습해보았다. 문제 상황에서 제기하였듯이 단순히 개인의 소득에 학력만이 영향을 끼치는지 확인하기 위해 사는 지역, 태어난 해, 결혼유무 등의 데이터를 사용하여 학습해보았고, 그 결과 Linear Regression이 가장 우수한 학습 결과를 보여주었다. 학습 결과를 분석해본 결과 개인의 소득에는 학력 뿐만 아니라 가족 구성원의 수(family_member), 태어난 해(year_born), 결혼 유무(marriage), 교육 정도(education_level)와 같이 다른 요소가 영향을 미친다는 것을 알 수 있었다. 다시 말해 학력이 소득에 영향을 미치기는 하지만 학력만이 소득을 결정하는 요소라고 보기에는 어렵고, 성별, 가족 구성원 수, 태어난 해, 결혼 유무 등 개인에 관한 여러 가지 정보가 소득에 직간접적으로 영향을 미친다는 것을 알 수 있다.

[참고 문헌]

서울과학종합대학원 디지털 혁신처. 2021. 오렌지. 국제경제경영.
손원성외 3인. 2021. 오렌지3로 알아가는 머신러닝 데이터 분석. 홍릉.
이고잉외 2인. 2021. 생활코딩 머신러닝. 위키북스.
Pearson 및 Spearman상관 방법의 비교, 2021. 11. 25. 3시 접속.
https://support.minitab.com/ko-kr/minitab/18/help-and-how-to/statistics/basic-statistics/supporting-topics/correlation-and-covariance/a-comparison-of-the-pearson-and-spearman-correlation-methods/
Orange Visual Programming. 2021. 11. 29. 4시 접속.
https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/data/correlations.html

03. 청소년 여러분, 행복하십니까?

구미산동고등학교 교사 황 은 아

학습 진행 과정

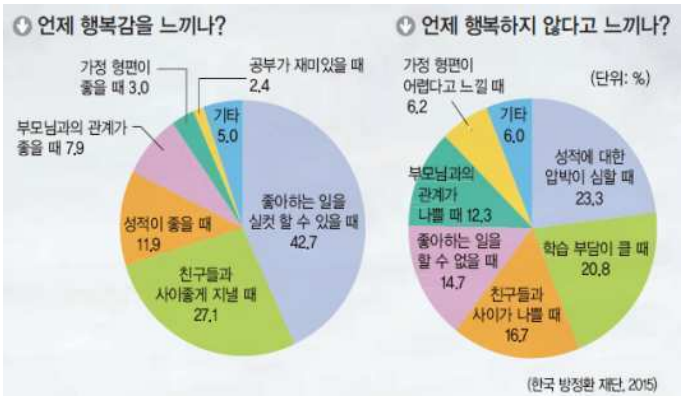
1단계	데이터 준비	- 사용 데이터: happy - 수집: 학생대상 설문조사 - 데이터 편집: 데이터 속성명 추가
2단계	데이터 불러오기	- 학습 데이터 불러오기 - 데이터의 속성별 Role(역할) 설정하기
3단계	데이터 시각화	- 시각화 도구을 이용한 데이터 탐색 - 사용한 시각화 도구: Distribution, Scattor Plot, Feature Statistics
4단계	속성 추출	- 데이터 시각화 결과 또는 Rank로 주요 속성 추출하기
5단계	모델 학습	- 분류를 이용한 모델 학습 - 사용된 알고리즘: SVM, Tree, Neural Network, Random Forest, k-NN
6단계	성능 평가	- test and score를 이용한 성능 평가 - 혼동 행렬을 이용한 성능 평가
7단계	예측	- Prediction을 이용한 테스트 데이터로 예측하기

학습 내용 요약

데이터 종류	문제 유형	사용된 학습 알고리즘	성능 평가 도구
정형 데이터(수치형)	분류	SVM, Tree, Neural Network, Random Forest, k-NN	혼동 행렬

문제 상황

2015년, '한국 방정환 재단'의 조사 결과 우리나라 청소년이 느끼는 주관적 행복도는 경제 협력 개발 기구(OECD) 국가들의 평균 행복 지수를 100으로 보았을 때, 74점 정도였다. 청소년이 행복감을 느끼는 때와 그렇지 않은 때를 조사한 결과는 오른쪽 그래프와 같다. 요즘 우리 청소년들의 경제적 여건이나 환경은 이전 세대에 비해 좋아졌지만, 자료를 보면 성적이나 학습에 대한 부담으로 인해 마음으로 느끼는 행복감은 크지 않다는 것을 알 수 있다.



이처럼 낮은 행복감은 자살, 우울증 등의 사회적 문제로 이어져 사회와 가정 내에서의 여러 가지 노력을 통해 극복하고 있지만 OECD 국 중에서는 여전히 최하위권에 머물고 있는 것으로 나타났다. 보건복지부가 한국보건사회연구원에 의뢰해 조사한 '2018년 아동 종합실태조사' 결과에 따르면 우리나라 9세~17세 아동 청소년 삶의 만족도는 10점 만점에 6.57점으로 2013년 조사 때보다 소폭 상승한 것을 보더라도 우리 사회 청소년의 삶에 대한 만족도가 낮다는 것을 알 수 있다. 청소년들이 삶에서 중요하다고 느끼고 있는 것은 무엇인지, 어떤 요소가 자신의 삶에 행복감을 줄 수 있는지 알아보고 나아가 우리가 행복의 기준을 무엇에 두고 살아가는지 어떻게 하면 행복한 삶을 살아갈 수 있는지 생각해보자.

01 데이터 준비하기

1 데이터 준비하기

청소년이 느끼는 행복감을 머신러닝으로 알아보기 위해서는 데이터 세트가 필요하다. 하지만 캐글(Kaggle), UCI(머신러닝 저장소) 등에서는 적합한 데이터를 구할 수 없다. 필요한 데이터를 구하기 위해서는 수집할 데이터 항목을 정하고 크롤링을 통해 웹사이트의 데이터를 수집하거나 설문조사를 통해 필요한 자료를 직접 수집하는 방법을 사용해야 한다.

실제 행복과 관련하여 수집할 데이터 항목을 고르기 위해서는 우선 행복감에 영향을 끼칠 수 있는 요소에 어떤 것이 있는지 알아보아야 한다. 이를 위해 매년 유엔 산하 자문기구인 '지속가능 발전 해법 네트워크'에서는 세계행복보고서(World Happiness Report)를 발간하는데, 각 나라의 경제적인 부분, 자연 환경 등 광범위하게 조사를 진행하여 매년 3월 22일 세계에서 가장 행복한 나라 순위를 제공한다. 행복을 어떻게 숫자로 계산하고 표현할 수 있

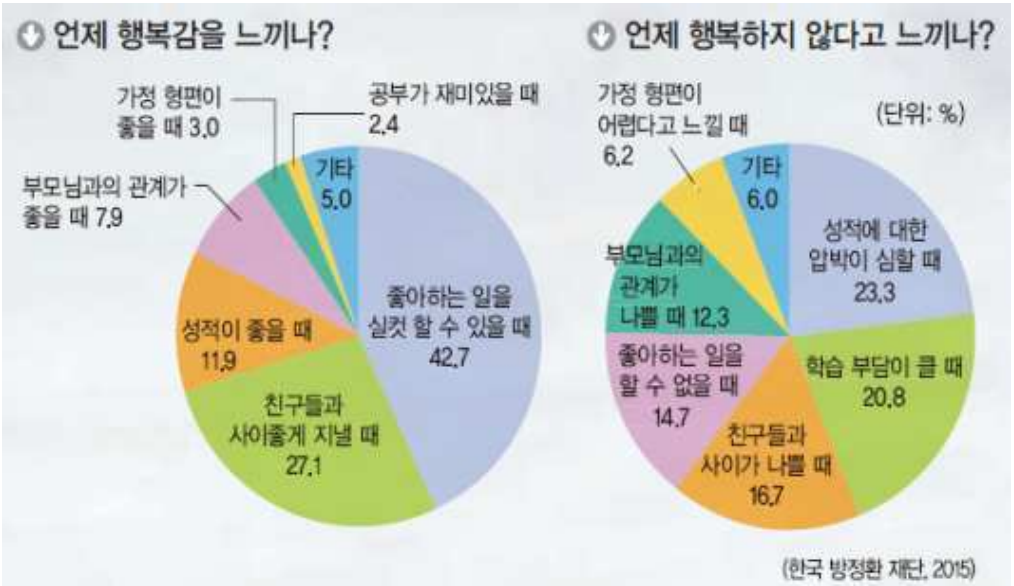
을까 싶지만 추상적으로 미뤄 생각하는 것 보다 객관적인 지표를 가지고 숫자로 알아보는 것이 더 정확하기도 하다. 이 보고서에서는 소득 수준, 건강 기대수명, 사회적 복지, 선택의 자유, 국가 부정부패에 대한 인식, 사회의 관용 등의 항목을 바탕으로 삶의 만족도, 즉 행복한 정도를 산출한다.

[그림 3-1] 2021 세계 행복 보고서

위 보고서에 따르면 한국은 전체 설문대상 95개국 중 50위에 선정되었고, 현재 국내 총생산(GDP)규모가 세계 10위임을 감안하면 경제력과 행복도가 비례하지 않는다는 것을 알 수 있다. 해당 보고서를 근거로 기관의 연구진이 개인의 삶을 행복하게 만드는 요인을 조사한 결과 다음과 같은 특성을 제시하였다.

목표달성 유무 명확한 목적 의식 감사하는 마음	소속감 시간 및 위치의 유연성 포용적이고 존중하는 환경	성장하는 느낌 공정한 보수 동료 간의 신뢰

우리나라에서도 청소년을 대상으로 행복감에 대하 설문을 자료가 있다. 2014년, '한국 방정환 재단'의 조사 결과 우리나라 청소년이 느끼는 주관적 행복도는 경제 협력 개발 기구(OECD) 국가들의 평균 행복 지수를 100으로 보았을 때, 74점 정도였다. 청소년이 행복감을 느끼는 때와 그렇지 않은 때를 조사한 결과는 오른쪽 그래프와 같다. 요즘 우리 청소년들의 경제적 여건이나 환경은 이전 세대에 비해 좋아졌지만, 자료를 보면 성적이나 학습에 대한 부담으로 인해 마음으로 느끼는 행복감은 크지 않다는 것을 알 수 있다.



이 조사에서는 청소년의 삶을 행복하게 만드는 요인을 다음과 같이 제시하였다.

타인(부모님, 친구)과의 관계
가정의 금전적인 환경
공부에 대한 흥미와 압박감
좋아하는 일을 선택할 수 있는 자유

위에서 제시한 자료들을 바탕으로 청소년이 자신의 삶을 행복하게 느끼고 있는지 여부를 알아보기 위해서 다음과 같이 설문조사를 실시하였다. 설문의 항목은 위의 자료를 근거로 하여 다음과 같이 설정하였다.

1. 나는 우리집의 가정형편이 좋다고 생각한다.
2. 나와 우리 가족 구성원은 모두 건강하다.
3. 나는 내가 좋아하는 일을 마음대로 자유롭게 할 수 있다.
4. 나는 가족, 부모님의 사랑과 인정을 받고 있다.
5. 나는 친구들의 사랑과 인정을 받고 있다.
6. 나는 성적이 좋은 편이다.
7. 나는 성적에 대한 압박이 크다.
8. 나는 공부가 재미있다.
9. 나는 미래의 나에 대한 진로와 꿈을 가지고 있다.
10. 나는 지금 행복하다.

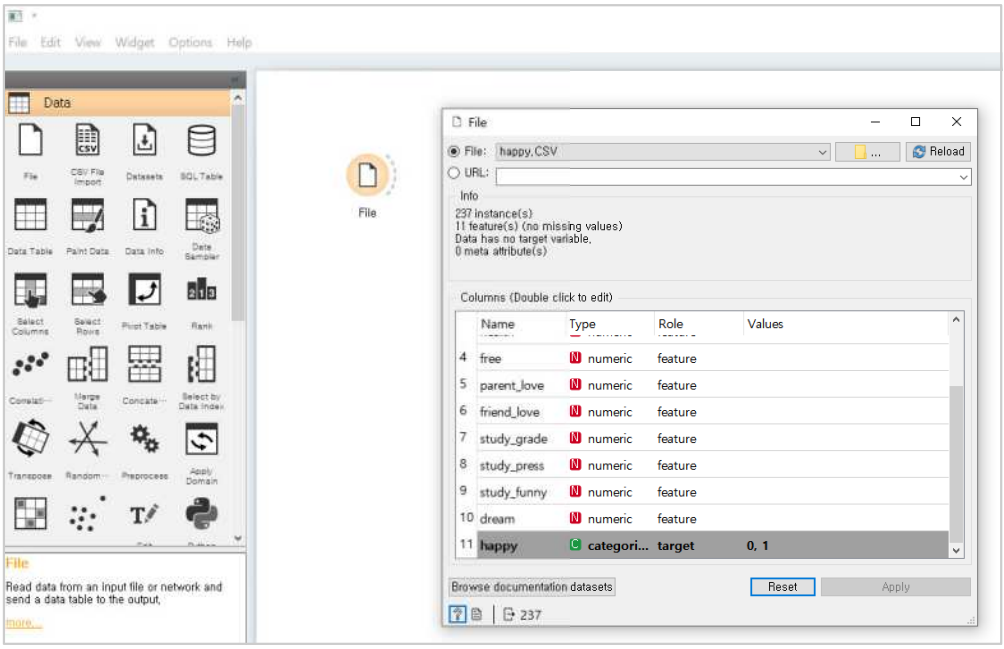
위의 총 10개 항목에 석적고등학교 고1~3학생 400여명이 응답하였고, 항목별로 1점(전혀 그렇지 않다)에서 10점(매우 그렇다)로 응답하였다. 위 결과를 이용하여 청소년이 느끼는 행복한 삶과 행복하지 않은 삶을 분류해보고자 한다. 단, 10번 문항은 1~10점으로 응답한 결과를 중간점수(5점)을 기준으로 1~5점일 경우 행복하지 않다, 6~10점은 행복하다로 변경하여 이진분류를 통해 청소년의 행복한 삶을 분류해보고자 한다.

설문조사의 결과를 바탕으로 모델 학습에 사용될 데이터를 결정하였고 그 정보는 다음과 같다.

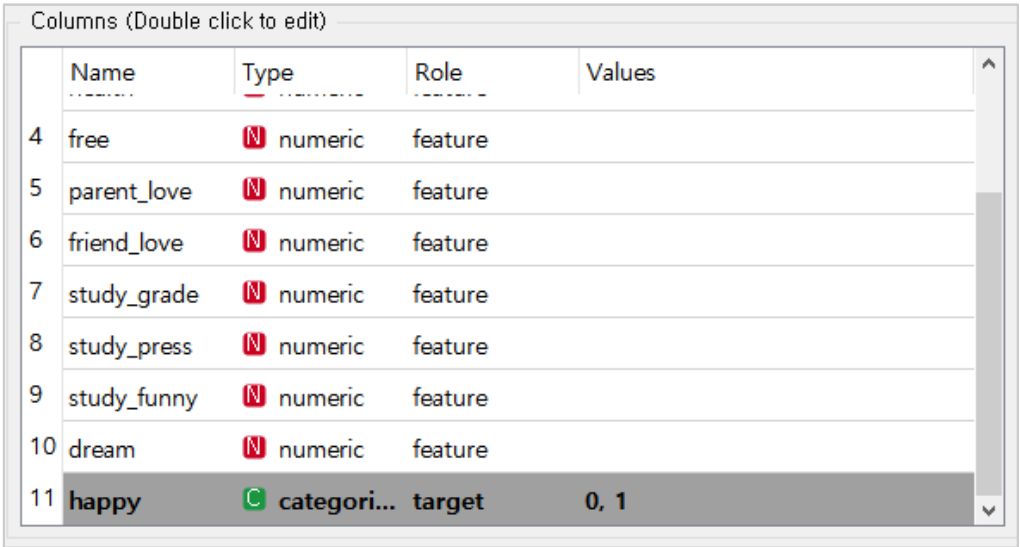
	속성	이름	데이터	의미
1	id	순번	-	-
2	money	나는 우리집의 가정형편이 좋다고 생각한다.	1~10	1) 전혀 그렇지 않다. 10) 매우 그렇다.
3	health	나와 우리 가족 구성원은 모두 건강하다.	1~10	1) 전혀 그렇지 않다. 10) 매우 그렇다.
4	free	나는 내가 좋아하는 일을 마음대로 자유롭게 할 수 있다.	1~10	1) 전혀 그렇지 않다. 10) 매우 그렇다.
5	parent_love	나는 가족, 부모님의 사랑과 인정을 받고 있다.	1~10	1) 전혀 그렇지 않다. 10) 매우 그렇다.
6	friend_love	나는 친구들의 사랑과 인정을 받고 있다.	1~10	1) 전혀 그렇지 않다. 10) 매우 그렇다.
7	study_grade	나는 성적이 좋은 편이다.	1~10	1) 전혀 그렇지 않다. 10) 매우 그렇다.
8	study_press	나는 성적에 대한 압박이 크다.	1~10	1) 전혀 그렇지 않다. 10) 매우 그렇다.
9	study_funny	나는 공부가 재미있다.	1~10	1) 전혀 그렇지 않다. 10) 매우 그렇다.
10	dream	나는 미래의 나에 대한 진로와 꿈을 가지고 있다.	1~10	1) 전혀 그렇지 않다. 10) 매우 그렇다.
11	happy	나는 지금 행복하다.	0,1	0) 행복하지 않다. 1) 행복하다.

2 데이터 불러오기

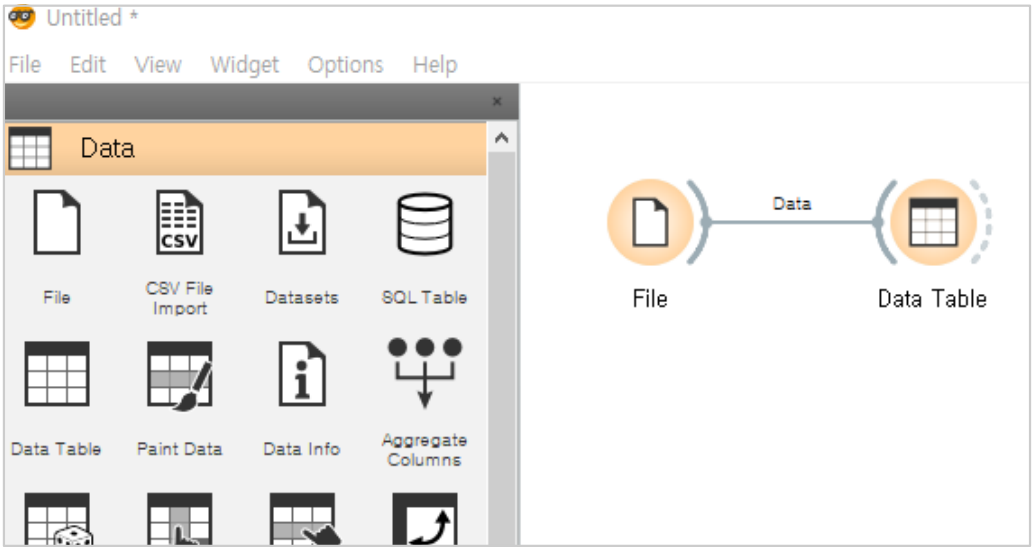
① Orange3을 실행한 후 왼쪽 위젯 팔레트에서 파일 위젯을 캔버스에 갖다 놓는다. 캔버스에 갖다 놓은 파일 위젯을 더블 클릭하여 청소년 행복 데이터 세트 파일(happy.csv)을 불러온다.



② columns에서 데이터 형식과 데이터의 Role(역할)을 확인한다. 이 중에서 happy(행복)이 결과값이므로 target으로 지정하고 id를 제외한 데이터는 원인이므로 feature로 지정한다.



③ 데이터를 잘 가져왔는지 확인하기 위해 위젯 팔레트에서 Data Table 위젯을 캔버스에 드래그한다. File의 오른쪽 괄호를 드래그하고 팝업에서 Data Table을 선택해서 데이터 테이블을 추구한다.



④ Data Table을 더블 클릭해 살펴보면 happy가 결과이고, 그 옆의 다른 열은 원인이다. 이처럼 숫자 데이터가 있을 때는 Visualize numeric values 옵션을 체크하면 값의 크기를 시각적으로 파악하기 수월하다.

	happy	money	health	free	parent_love
1	1	5	9	7	7
2	1	8	8	10	9
3	1	9	10	10	7
4	1	3	5	7	8
5	1	8	10	10	10
6	1	10	10	10	10
7	1	7	8	9	8
8	1	8	10	10	10
9	1	7	8	10	10
10	1	10	7	10	10
11	1	6	6	6	8
12	1	5	5	5	5
13	0	8	10	6	8
14	0	4	7	3	4
15	0	6	9	7	6

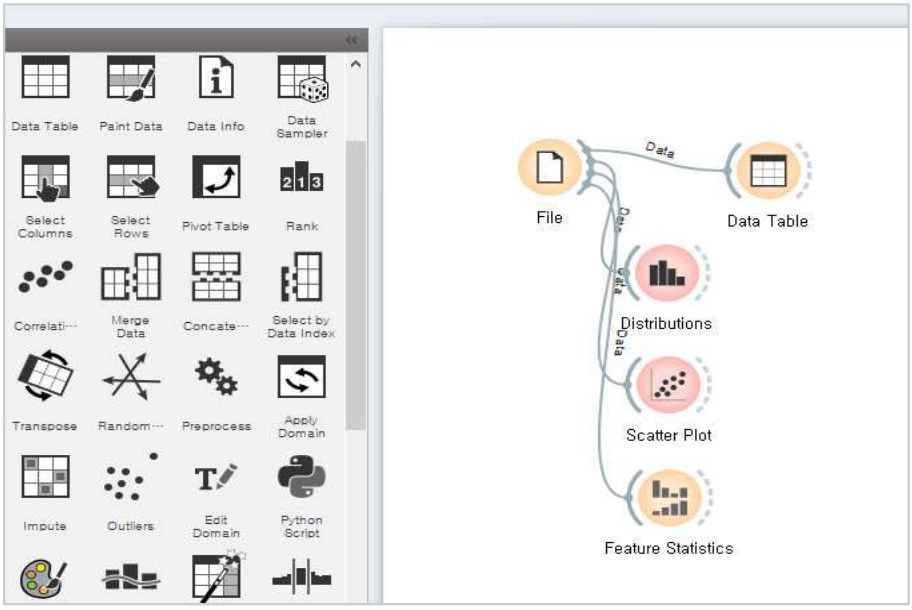
02 데이터 탐색

청소년의 행복감에 결정적으로 영향을 미치는 속성은 무엇일까? 어떤 속성이 청소년이 자신의 삶을 행복하다고 느끼게 하는지를 알아보고, 이 속성이 기계학습의 분류에 많은 영향을 미치는지 알아보기 위해 데이터를 시각화해보자.

이러한 과정을 통해 기계학습에 영향을 주는 핵심 데이터 속성을 추출할 수 있는데 이러한 과정을 데이터 탐색이라고 한다. 데이터 탐색은 분석 대상 데이터를 다양한 관점으로 살펴보고 그 특성을 이해하는 과정으로 좋은 분석 모델을 만들기 위해 반드시 필요한 과정이다. 다양한 관점으로 깊이 있게 데이터를 파악하기 위해 주로 히스토그램과 산점도 등 데이터를 시각화하여 분석하게 된다.

1 데이터 시각화 하기

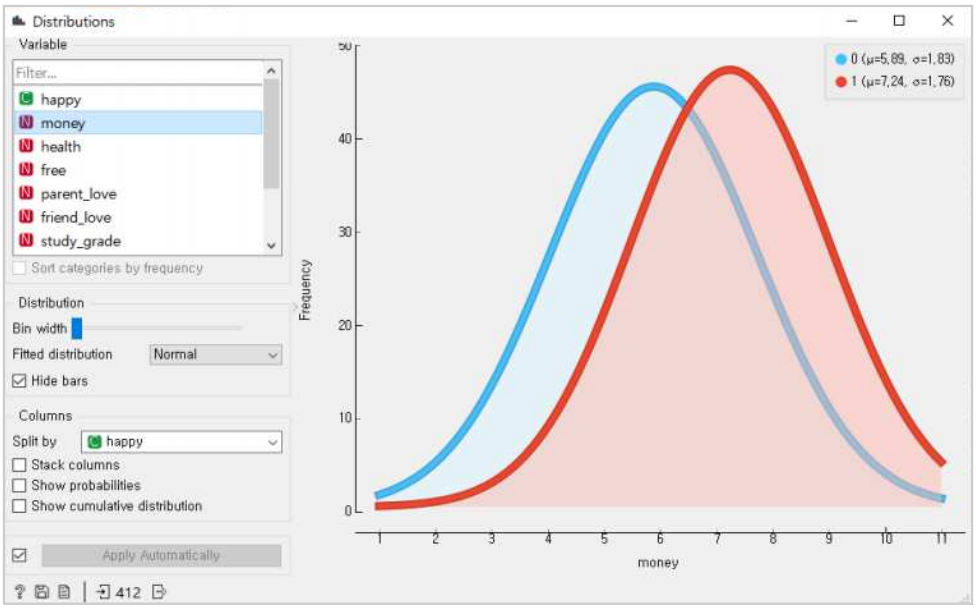
- 데이터를 시각화하기 위해서 속성간의 상관관계를 파악할 수 있는 Distributions, Scatter Plot, Feature Statistics를 사용하려고 한다.
- 각각 Data와 Visualize 위젯 팔레트에서 해당 위젯을 찾아 캔버스로 드래그한 후 파일의 오른쪽 곡선을 드래그앤 드롭하여 연결한다.



2 Distributions로 나타내기

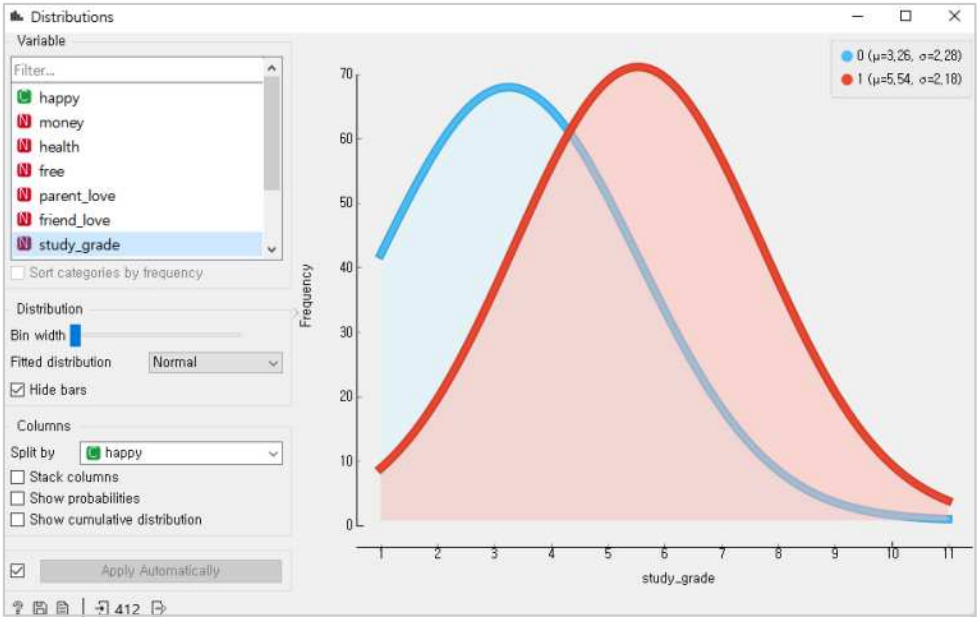
- 단일 속성에 대한 값 분포를 표시해준다.
- 선택된 데이터에 따라 히스토그램 데이터(히스토그램의 빈도수)를 그래프로 출력해준다.
- 그래프는 각 속성 값이 데이터에 나타나는 횟수(예: 인스턴스 수)를 보여준다.

- 데이터에 클래스 변수가 포함된 경우 각 속성 값에 대한 클래스 분포가 표시됩니다.
- 표시할 변수 목록에서 원하는 변수를 선택하면 표시된 값을 빈도별로 정렬한다.
- Coloums는 분할 기준인 happy(행복)을 선택한다.



[그림 3-2] money의 분포

money는 행복하지 않다(0)와 행복하다(1)의 빈도가 비슷하고 값의 1.5정도 차이를 보이므로 두 클래스를 분류하는데 영향을 미치는 속성이다.



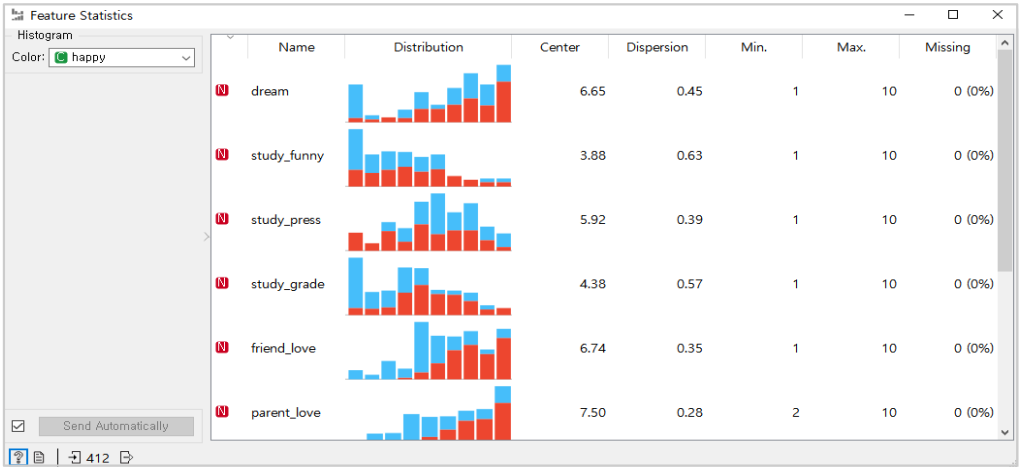
[그림 3-3] study_grade의 분포

study_grade는 행복하지 않다(0)와 행복하다(1)의 빈도가 비슷하고 값의 2.3정도 차이를 보이므로 두 클래스를 분류하는데 영향을 미치는 속성이다.

- 데이터 위젯의 Distributions를 사용하여 속성을 분석해보면 money와 study_grade 속성이 학습에 큰 영향을 미치며 id를 제외한 나머지 속성도 두 클래스간에 다른 결과를 보이므로 결과에 영향을 미친다는 것을 확인할 수 있다.

3 Feature Statistics 나타내기

- 데이터 속성에 대한 기본 통계를 제공한다.
- 지정된 데이터 세트의 속성을 신속하게 검사하게 제공하는데 왼쪽의 histogram에서 선택한 속성에 따라 통계값을 제공한다.
- 각 속성값의 통계정보와 도수분포를 동시에 볼 수 있는 그래프이다. 특성별 도수분포와 최솟값, 최댓값, 중앙값 등을 확인할 수 있다. 이를 통해 상대적으로 작은 값과 큰 값을 가지는 속성이 있는지, 전처리가 필요한 지도 확인할 수 있다.
- 속성을 특성 통계표로 살펴보았을 때 행복하다(1)는 빨강색, 행복하지 않다(0)은 파랑색으로 표시된 것을 확인할 수 있다. 모든 속성에 결측치가 없고 속성값이 상대적으로 크거나 작은 값이 없으므로 전처리가 필요하지 않음을 알 수 있다.

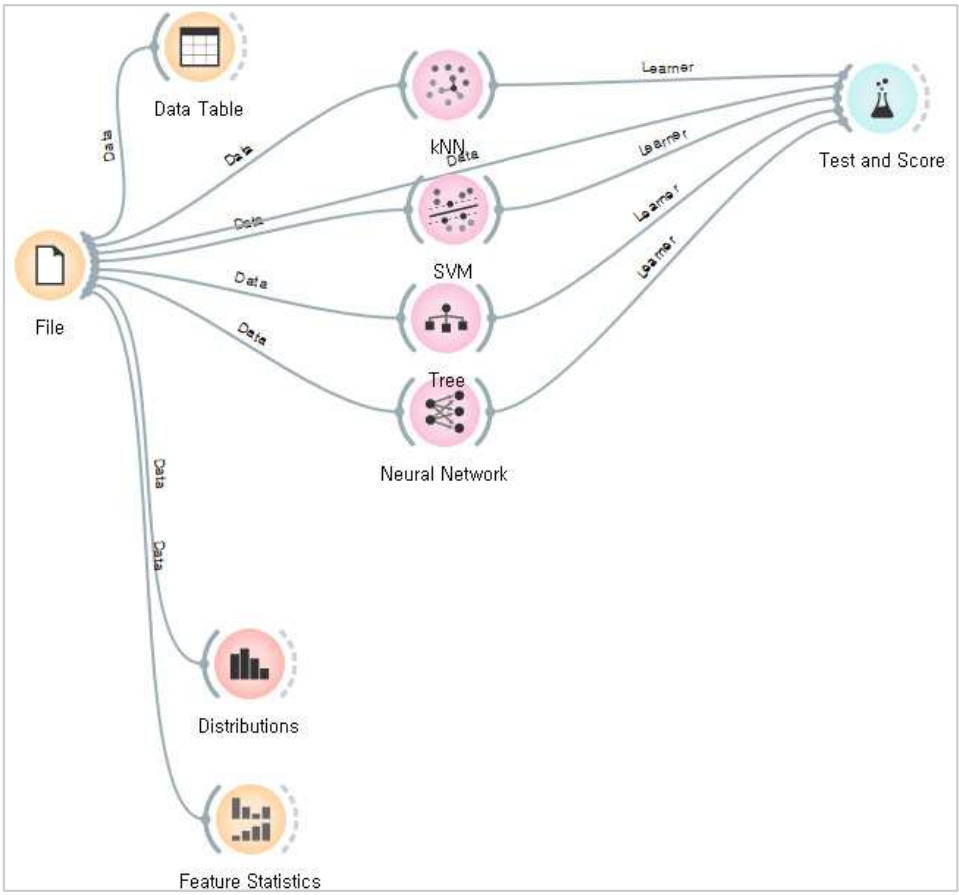


03 모델 학습과 성능 평가하자

추출한 데이터 속성을 바탕으로, 기계학습 알고리즘과 데이터를 연결하여 모델 학습한다. 오렌지에서는 다양한 기계학습 알고리즘을 한꺼번에 연결하여 모델을 만들 수 있다. 여기서는 분류에 자주 사용하는 kNN, SVM, Tree, Neural Network를 이용하여 모델을 구성하였다.

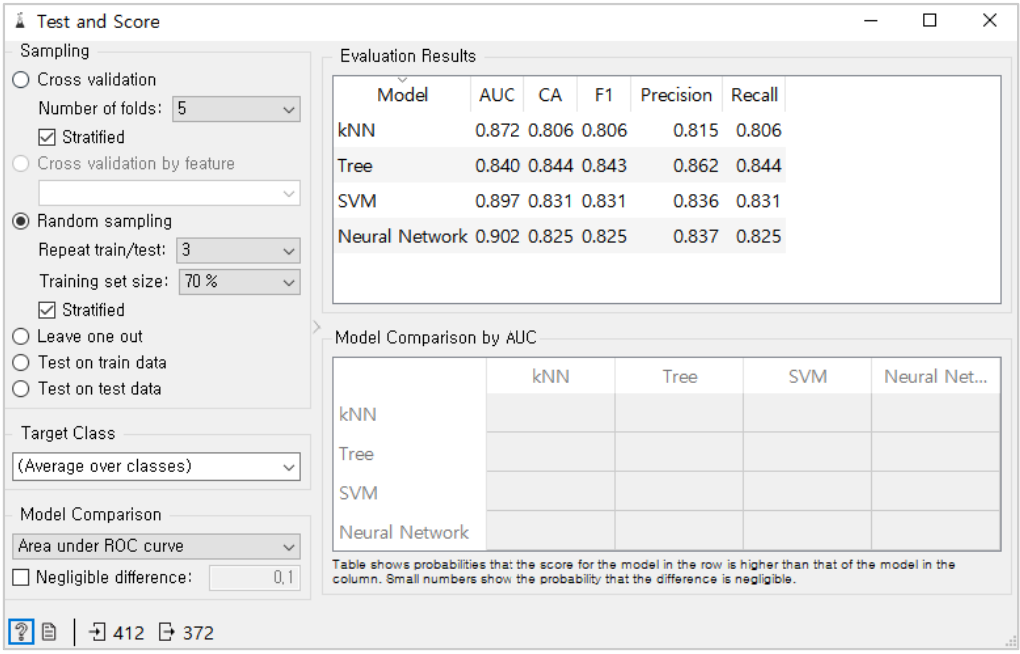
1 모델 성능 평가하기

- Test and Score를 이용하여 모델의 성능을 확인해보자.
- 좌측 모델 항목에서 kNN, SVM, Tree, Neural Network, Random Forest을 선택해 왼쪽 마우스로 오른쪽 화면으로 끌고 온다.
- file의 오른쪽에 있는 둥근 점선 부분에 마우스 커서를 놓고 각 모델로 드래그한다.



- 샘플링 방식 중 Random Sampling은 전체 데이터를 섞어서 무작위로 훈련 데이터와 테스트 데이터를 분리한다. 또한 훈련과 테스트의 반복횟수를 설정할 수 있

다. 테스트 데이터를 이용한 계산을 쉽게하기 위해 반복(repeat train/test)를 10회로 설정하였다.



- test and score를 더블클릭해서 모델의 테스트 결과 즉 성능을 확인할 수 있다.
- Evaluation Results(성능평가 결과)를 보면 좌측에 각 모델이 나타나있고 각 모델별 성능지표(AUC, CA, F1, Precision, Recall 등)을 확인할 수 있다.
- 모델의 성능을 종합적으로 평가할 수 있는 F1을 클릭해보면 Tree가 0.843으로 가장 높은 것을 알 수 있다.
- F1은 Precision과 Recall을 섞어서 한번에 보여주는 지표이다. 보통은 Precision과 Recall의 조화평균을 내서 구하는데 1의 값에 가까울수록 모델의 성능을 좋다고 판단한다.

F1=2 * (Precision * Recall / Precision + Recall)

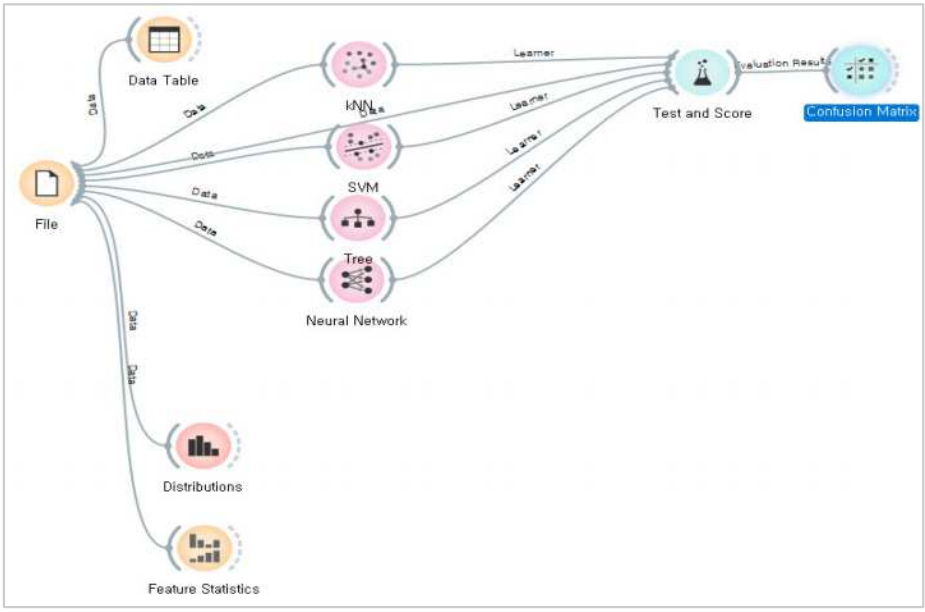
- Precision과 Recall은 다음과 같은 값으로 모델의 성능을 평가하는 다른 지표 중 하나이다.

  - Precision: 데이터 중에서 실제로 양성인 값의 비율을 계산한 수치
  - Recall: 전체 양성 중에서 진단으로 양성을 찾아낸 비율

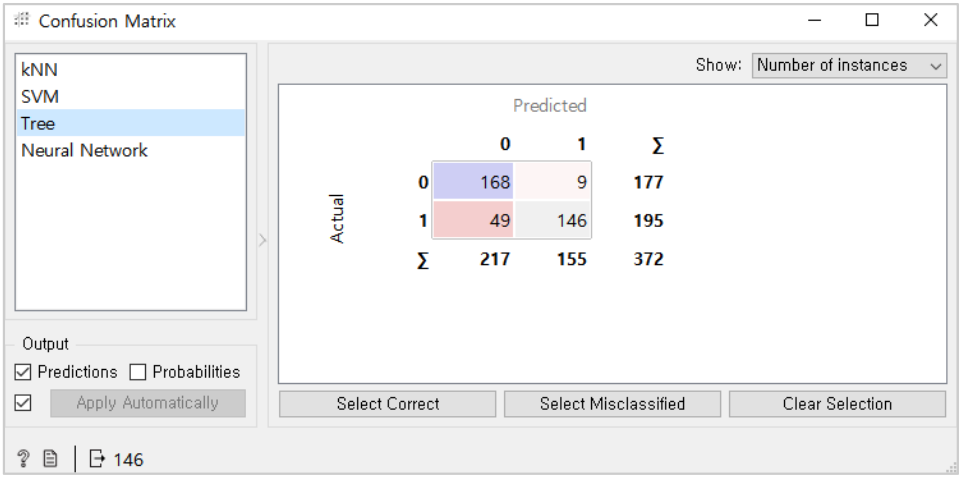
- 그 외 CA(Classification Accuracy)는 정확도를 의미한다.
- 위 성능평가를 분석해보면 Tree의 F1이 가장 높아 종합적으로 보았을 때 가장 우수한 모델이고 다음으로 SVM, Neural Network, kNN 순으로 성능이 평가된다는 것을 확인할 수 있다.

2 Confusion Matrix 이용하기

- Confusion Matrix를 이용하면 각 모델별 성능을 훨씬 구체적으로 확인할 수 있다.
- 좌측 항목 Evaluate에서 Confusion Matrix를 클릭하여 오른쪽 화면에 끌어 놓은 후 Test and Score와 연결한다.
- Confusion Matrix를 더블클릭하면 각 모델별 정확도를 넘어 구체적으로 어떤 모델이 어떤 경우에는 맞혔고 어떤 경우에는 틀렸는지 확인할 수 있다.

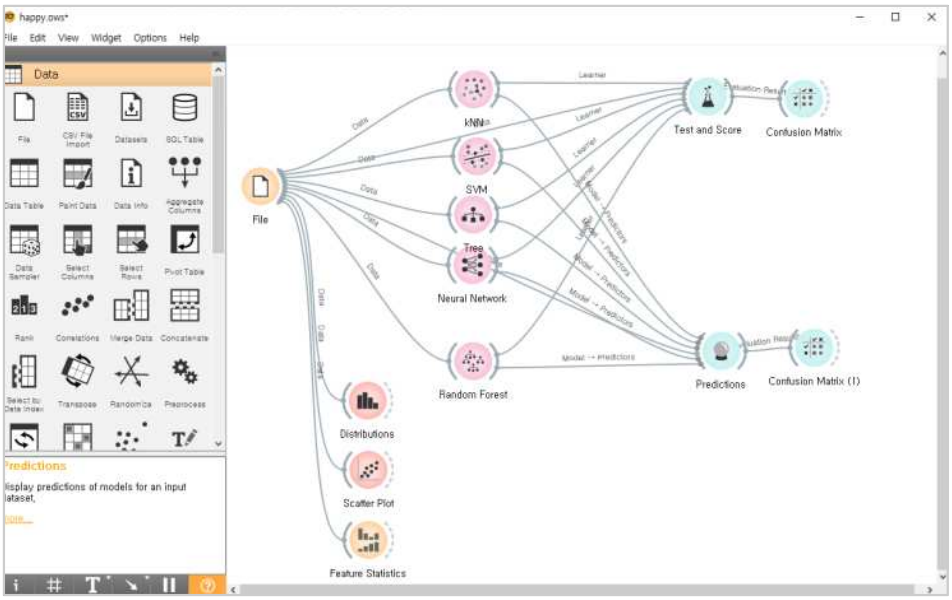


- 가장 성능이 우수했던 tree를 클릭하여 맞춘 개수를 확인해보면
- 실제 행복하지 않다(0)라고 데이터를 정확하게 분류한 결과가 168, 잘못 분류한 결과가 9이다.
- 실제 행복하다(1)라고 데이터를 정확하게 분류한 결과가 146, 잘못 분류한 결과가 49이다.

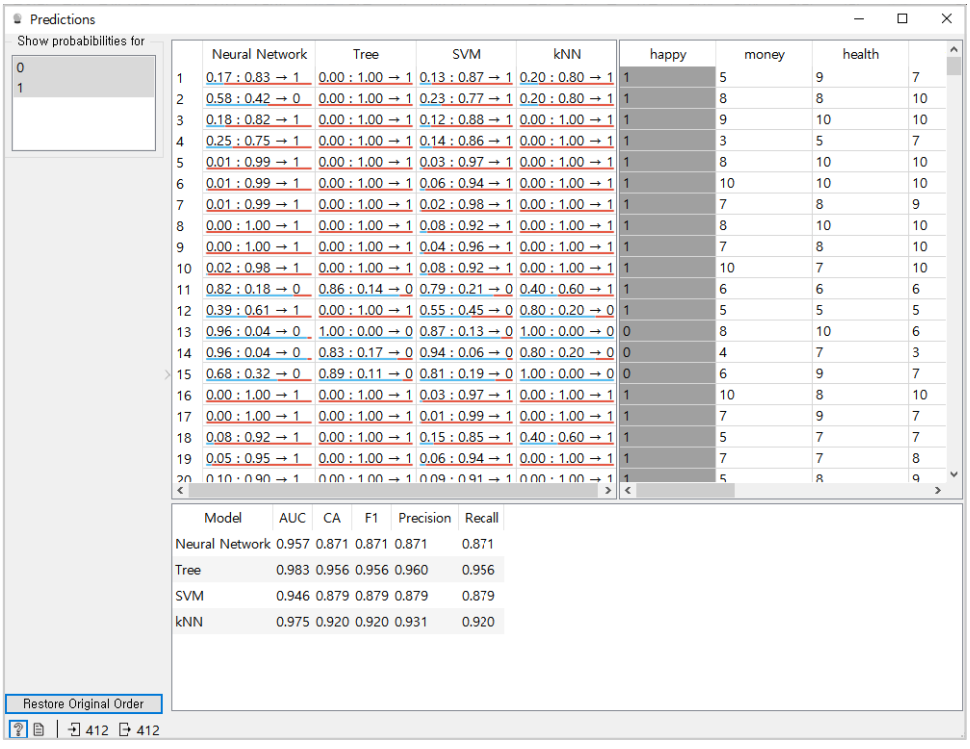


3 데이터 예측하기

- 이제 학습시킨 모델들을 이용하여 데이터를 예측해보자.
- Evaluate에서 Predictions를 가져오고 데이터를 입력해준다.



- 앞에서 학습시킨 모델을 Predictions에 이어주면 학습한 모델에 기반하여 값을 다음과 같이 예측할 수 있다.



	Neural Network	Tree	SVM	kNN	happy	money	health	
1	0.17 : 0.83 → 1	0.00 : 1.00 → 1	0.13 : 0.87 → 1	0.20 : 0.80 → 1	1	5	9	7
2	0.58 : 0.42 → 0	0.00 : 1.00 → 1	0.23 : 0.77 → 1	0.20 : 0.80 → 1	1	8	8	10
3	0.18 : 0.82 → 1	0.00 : 1.00 → 1	0.12 : 0.88 → 1	0.00 : 1.00 → 1	1	9	10	10
4	0.25 : 0.75 → 1	0.00 : 1.00 → 1	0.14 : 0.86 → 1	0.00 : 1.00 → 1	1	3	5	7
5	0.01 : 0.99 → 1	0.00 : 1.00 → 1	0.03 : 0.97 → 1	0.00 : 1.00 → 1	1	8	10	10
6	0.01 : 0.99 → 1	0.00 : 1.00 → 1	0.06 : 0.94 → 1	0.00 : 1.00 → 1	1	10	10	10
7	0.01 : 0.99 → 1	0.00 : 1.00 → 1	0.02 : 0.98 → 1	0.00 : 1.00 → 1	1	7	8	9
8	0.00 : 1.00 → 1	0.00 : 1.00 → 1	0.08 : 0.92 → 1	0.00 : 1.00 → 1	1	8	10	10
9	0.00 : 1.00 → 1	0.00 : 1.00 → 1	0.04 : 0.96 → 1	0.00 : 1.00 → 1	1	7	8	10
10	0.02 : 0.98 → 1	0.00 : 1.00 → 1	0.08 : 0.92 → 1	0.00 : 1.00 → 1	1	10	7	10
11	0.82 : 0.18 → 0	0.86 : 0.14 → 0	0.79 : 0.21 → 0	0.40 : 0.60 → 1	1	6	6	6
12	0.39 : 0.61 → 1	0.00 : 1.00 → 1	0.55 : 0.45 → 0	0.80 : 0.20 → 0	1	5	5	5
13	0.96 : 0.04 → 0	1.00 : 0.00 → 0	0.87 : 0.13 → 0	1.00 : 0.00 → 0	0	8	10	6
14	0.96 : 0.04 → 0	0.83 : 0.17 → 0	0.94 : 0.06 → 0	0.80 : 0.20 → 0	0	4	7	3
15	0.68 : 0.32 → 0	0.89 : 0.11 → 0	0.81 : 0.19 → 0	1.00 : 0.00 → 0	0	6	9	7
16	0.00 : 1.00 → 1	0.00 : 1.00 → 1	0.03 : 0.97 → 1	0.00 : 1.00 → 1	1	10	8	10
17	0.00 : 1.00 → 1	0.00 : 1.00 → 1	0.01 : 0.99 → 1	0.00 : 1.00 → 1	1	7	9	7
18	0.08 : 0.92 → 1	0.00 : 1.00 → 1	0.15 : 0.85 → 1	0.40 : 0.60 → 1	1	5	7	7
19	0.05 : 0.95 → 1	0.00 : 1.00 → 1	0.06 : 0.94 → 1	0.00 : 1.00 → 1	1	7	7	8
20	0.10 : 0.90 → 1	0.00 : 1.00 → 1	0.09 : 0.91 → 1	0.00 : 1.00 → 1	1	5	8	9

Model	AUC	CA	F1	Precision	Recall
Neural Network	0.957	0.871	0.871	0.871	0.871
Tree	0.983	0.956	0.956	0.960	0.956
SVM	0.946	0.879	0.879	0.879	0.879
kNN	0.975	0.920	0.920	0.931	0.920

- Predictions 창을 더블클릭해서 열어보면 위와 같이 happy를 어떻게 예측했는지 확인해볼 수 있다.
- 앞서 성능평가 결과를 확인했을 때 사용한 모델 중 Tree가 가장 우수했던 것과 같이 실제 happy의 데이터와 학습한 모델로 예측한 값을 비교해보았을 때 성능이 0.956으로 우수한 것을 확인할 수 있다.

03. 청소년 여러분, 행복하십니까?

정리하기

청소년들이 삶에서 중요하다고 느끼고 있는 것은 무엇인지, 어떤 요소가 자신의 삶에 행복감을 줄 수 있는지 알아보기 위해 데이터를 수집하고 학습해보았다. 청소년의 행복에 영향을 미치는 요소에는 타인(부모님, 친구)과의 관계, 가정의 금전적인 환경, 공부에 대한 흥미와 압박감, 좋아하는 일을 선택할 수 있는 자유 등의 데이터를 사용하여 학습해보았고, 그 결과 Tree가 가장 우수한 학습결과를 보여주었다. 학습 결과를 분석해보면 청소년의 행복에는 가정의 금전적인 환경(money)와 자신의 성적(grade)가 행복을 결정하는 중요한 요소라는 것을 알 수 있다. 문제상황에서 제시한 2015년 한국 방정환 재단의 조사결과와 같이 성적이나 학습에 대한 부담감이 청소년의 행복에 영향을 미치고 있었다. 반면 청소년들의 경제적 여건이나 환경은 이전 세대에 비해 좋아졌다는 결과와 달리 여전히 가정의 금전적인 환경도 행복에 영향을 미친다는 결과가 나타났다. 본 데이터가 일부 지역의 학생을 대상으로 한 데이터로, 해당 지역이 공장이 인접한 도농복합 지역으로 가정의 경제적 격차가 큰 지역이기 때문에 해당 요소가 영향을 미치는 요소로 나타났을 것이라 예상한다.

[참고 문헌]

서울과학종합대학원 디지털 혁신처(2021). 오렌지. 국제경제경영.
손원성 외 3인(2021). 오렌지3로 알아가는 머신러닝 데이터 분석. 홍릉.
이고잉 외 2인(2021). 생활코딩 머신러닝. 위키북스.
Pearson 및 Spearman상관 방법의 비교. 2021. 11. 25. 3시 접속.
https://support.minitab.com/ko-kr/minitab/18/help-and-how-to/statistics/basic-statistics/supporting-topics/correlation-and-covariance/a-comparison-of-the-pearson-and-spearman-correlation-methods/
Orange Visual Programming. 2021. 11. 29. 4시 접속.
https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/data/correlations.html

MEMO

04. 밀알의 크기로 밀알의 종류를 구분할 수 있을까?

금오고등학교 교사 박 윤 희

학습 진행 과정

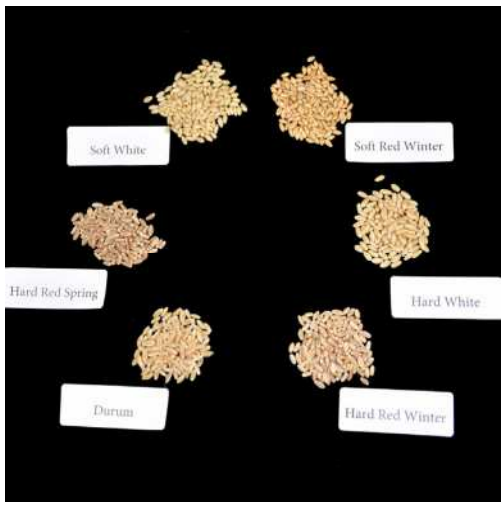
1단계	데이터 준비	- 사용 데이터: seeds dataset - 수집: 캐글(Kaggle)
2단계	데이터 불러오기	- 학습 데이터 불러오기 - 데이터의 속성별 Role(역할) 설정하기
3단계	데이터 시각화	- 시각화 도구을 이용한 데이터 탐색 - 사용한 시각화 도구: Distribution, Scatter Plot, Feature Statistics
4단계	모델 학습	- 분류를 이용한 모델 학습 - 사용된 알고리즘: SVM, Neural Network, Random Forest, kNN
5단계	성능 평가	- Test and Score를 이용한 성능 평가 - 혼동 행렬을 이용한 성능 평가
6단계	예측	- Predictions를 이용한 테스트 데이터로 예측하기

학습 내용 요약

데이터 종류	문제 유형	사용된 학습 알고리즘	성능 평가 도구
정형 데이터(수치형)	분류	SVM, Neural Network, Random Forest, kNN	혼동 행렬

문제 상황

밀은 전 세계적으로 재배되는 작물로 세계 곡물 생산량에서 옥수수에 이어 2위를 차지하는 작물이다. 또한 유럽, 아메리카 등지의 여러 나라에서는 주식용으로 쓰인다. 밀은 낟알을 곱게 빻아 밀가루로 만들어 빵, 과자, 면 등을 만들기도 한다. 다양한 음식을 만들 때 사용되는 밀가루는 단백질 함량과 글루텐의 세기에 따라 강력분, 중력분, 박력분으로 구분되고 있다. 이러한 밀가루의 단백질 함량과 글루텐의 세기는 밀의 종류에 따라 다르다.



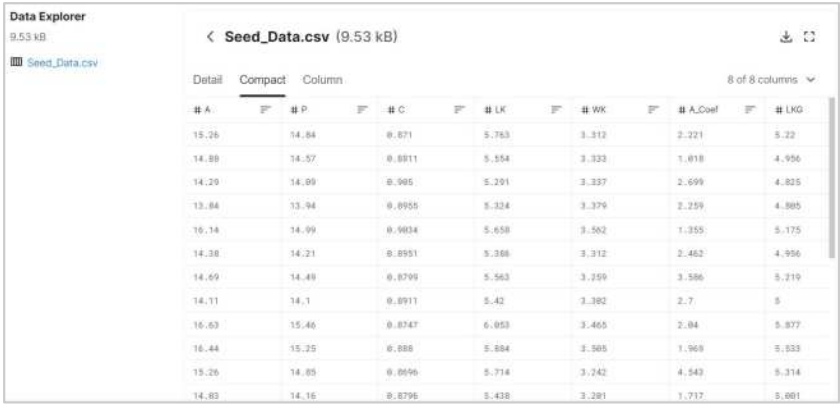
위의 사진에서 보다시피 밀의 품종에 따라 밀 낟알의 모양이 다르다. 밀 낟알에 대한 수치 정보만을 가지고 이 낟알이 어떤 품종인지 판단하고 분류할 수 있을까?

01 데이터 준비하기

1 seed 데이터 세트

밀알에 대한 수치 정보를 이용해 품종을 분류하기 위해서는 밀알 품종 분류와 관련된 데이터 세트가 필요하다. 기계학습에 필요한 데이터 세트는 캐글(Kaggle)에 접속하여 다운로드 받을 수 있다. 캐글은 2010년에 설립된 예측 모델 및 대회 플랫폼으로 기계학습에 사용할 수 있는 다양한 데이터 세트를 무료로 다운로드 받을 수 있는 사이트이다. 밀알 데이터 세트는 다음의 과정에 따라 다운로드 받을 수 있다.

다운로드 아래의 Description 영역의 하단에 있는 [∨]을 클릭하여 확장하면 해당 데이터 세트에 대한 상세 내용을 확인할 수 있다. 3가지 서로 다른 종(Kama, Rosa, Canadian)의 밀알의 수치 정보를 수집한 데이터 세트로 총 7개의 속성(Feature)과 1개의 타겟(Target)값으로 구성되어 있으며, 총 210개의 데이터로 구성되어 있다. 다운로드 받은 데이터 세트를 열어보거나 Data Explorer의 Compact 탭을 선택하면 상세 데이터를 확인할 수 있다.



[그림 4-1] seed 데이터 세트 내용

데이터 세트를 확인해 보니 속성 이름이 아주 간단하게 기록되어 있다. 이 데이터 세트의 원본을 제공하는 UCI Machine Learning Repository에서는 데이터의 속성 정보를 제공하고 있어 이를 이용해 데이터 세트의 속성 이름을 다음과 같이 확인할 수 있다.

Attribute Information:

To construct the data, seven geometric parameters of wheat kernels were measured:
1. area A,
2. perimeter P,
3. compactness C = 4*pi*A/P^2,
4. length of kernel,
5. width of kernel,
6. asymmetry coefficient
7. length of kernel groove.

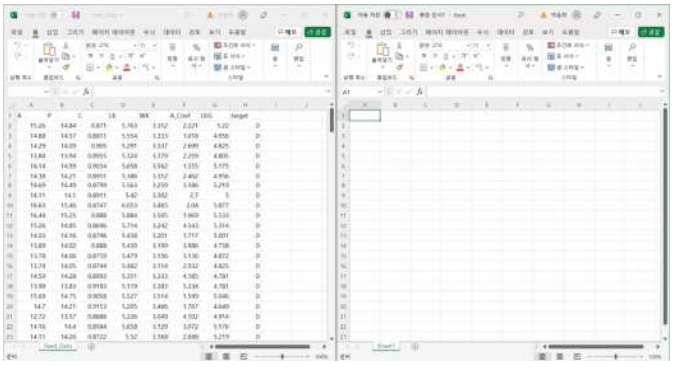
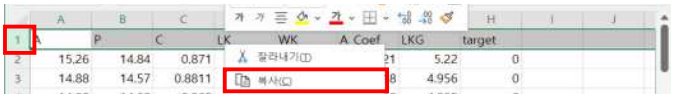
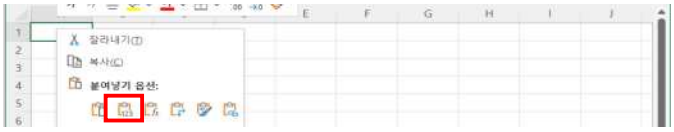
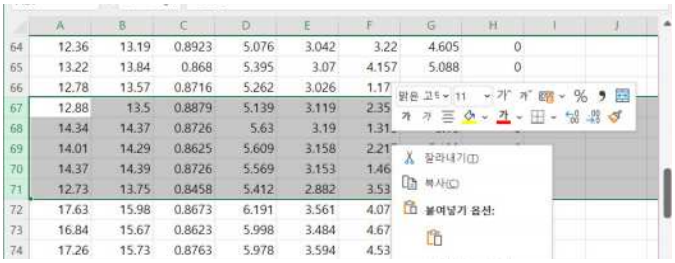
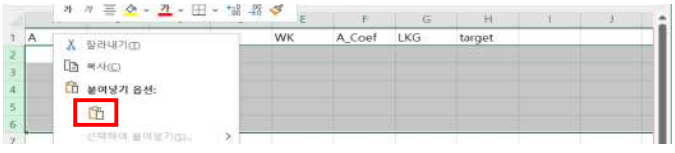
All of these parameters were real-valued continuous.

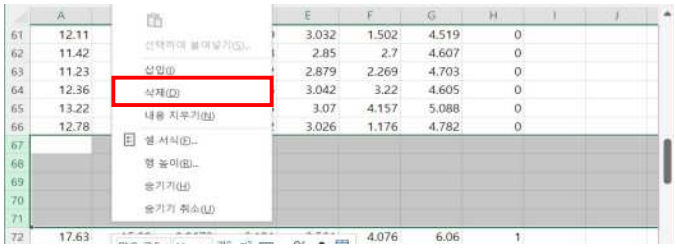
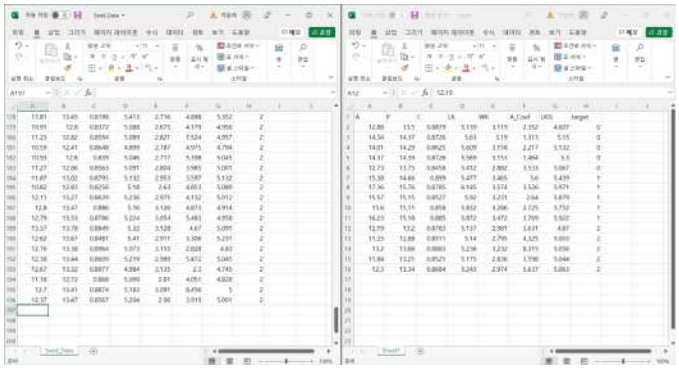
[그림 4-2] 밀알 데이터 세트의 속성에 대한 설명(출처: UCI Machine Learning Repository)

속성	설명	비고
A(area)	면적	
P(perimeter)	둘레	
C(compactness)	조밀성	C = 4*pi*A/P^2
LK(length of kernel)	낟알의 길이	
WK(width of kernel)	낟알의 너비	
A_Coef(asymmety coefficient)	비대칭 계수	
LKG(length of kernel groove)	낟알 홈의 길이	
target(species)	밀알 품종	0: Kama 1: Rosa 2: Canadian

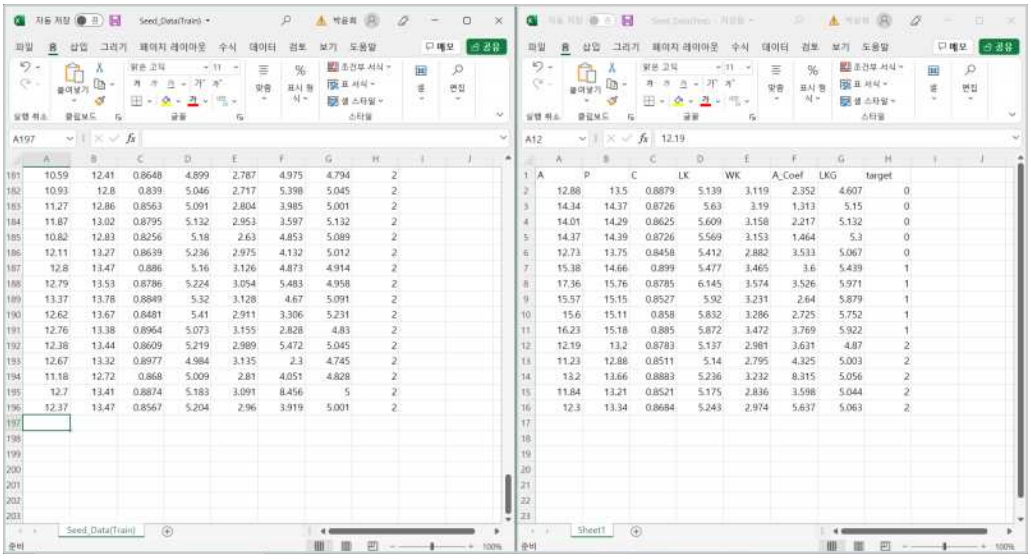
데이터 세트의 8번째 속성은 target으로 작성되어 있으나 속성 정보에는 나와 있지 않다. 기계학습에서 target은 기계학습 모델을 이용하여 분류하거나 예측할 속성을 가리킨다. 이 데이터 세트는 밀알에 대한 수치 정보를 이용해 밀알의 품종을 분류하기 위해 사용하거나 유사한 품종끼리 군집화(Clustring)하는데 사용할 수 있으므로 target은 밀알 품종에 대한 정보를 나타내는 것이라 볼 수 있다.

해당 데이터 세트를 이용하여 훈련(Train)을 진행하고 훈련에 사용되지 않은 새로운 데이터를 이용해 테스트(Test)해 보기 위해 다운로드 받은 데이터 중에서 각 품종별 5개의 데이터를 추출하여 15개의 데이터를 테스트 데이터로 사용할 것이다. 훈련 데이터와 테스트 데이터를 준비하는 과정은 다음과 같다.

단계	설명
	seed 데이터 세트 파일을 열고, 테스트 데이터를 저장할 새로운 스프레드 시트를 준비한다.
	기계학습에 사용할 속성 이름이 있는 [1행]을 선택하여 [복사]한다.
	**새로운 스프레드 시트**의 [A1셀]을 선택한 후 [붙여넣기] 한다. 붙여넣기 옵션은 [값]을 선택한다.
	target(타겟)값이 0인 데이터 중 마지막 5개의 데이터를 [행 번호 67~71]을 이용해 선택한 후 [마우스 우클릭] - [잘라내기]를 선택한다.
	새로운 스프레드 시트의 [A2셀]을 선택한 후 [붙여넣기] 한다. 붙여 넣기 옵션은 [붙여넣기]를 선택한다.

단계	설명
	기존 파일에서 데이터를 잘라낸 [행67~71]을 선택하여 마우스 우클릭 - [삭제]한다.
	위의 과정과 동일하게 target값이 1인 데이터(행 132~136)와 2인 데이터(행 197~202)를 5개씩 잘라내서 붙여넣는다.
	메뉴 - [다른 이름으로 저장]을 눌러 **기존 파일(훈련 데이터)**을 "Seed_Data(Train).csv"로 저장한다.
	**새로운 파일(테스트 데이터)**은 "Seed_Data(Test).csv"로 저장한다.

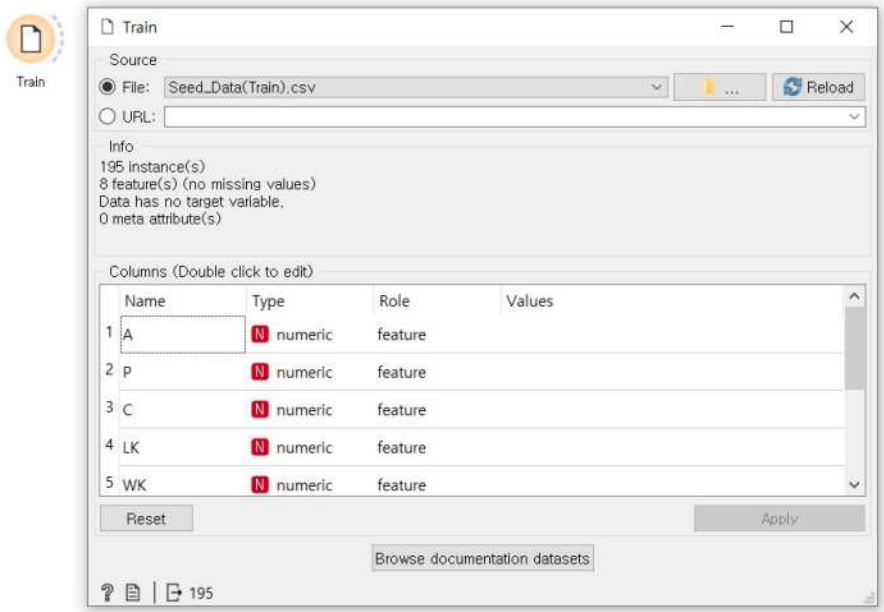
위의 단계를 모두 완료하면 다음과 같은 상태가 된다.



[그림 4-3] 훈련 데이터와 테스트 데이터 분리

2 데이터 불러오기

① Orange3을 실행하여 다운로드 받은 Seed_Data(train) 파일을 불러온다.
Data - File을 선택하여 위젯을 화면에 배치시킨 후, 위젯의 이름을 Train으로 변경한 후 데이터 파일을 로드한다.

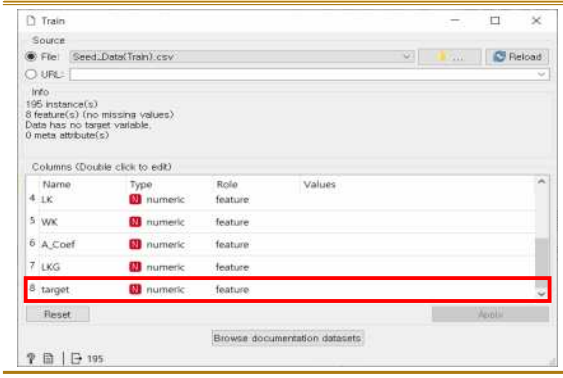
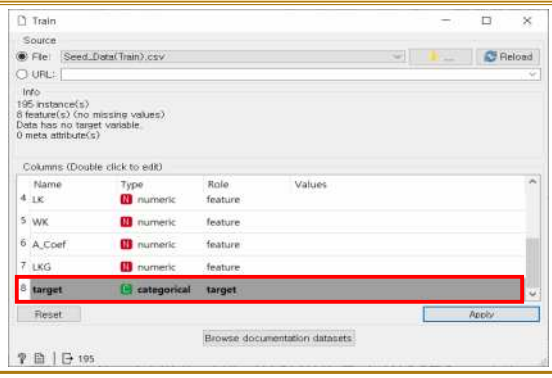


[그림 4-4] 훈련 데이터 불러오기

② 데이터 속성을 확인하고 type과 role을 변경한다.

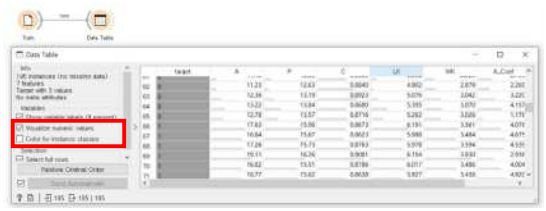
기계학습을 진행하기 위해 속성의 type과 role를 변경한다. 7개의 속성을 이용하여 품종을 분류하는 것이 목적이므로 target의 type을 categorical로 변경하고 role을 target으로 변경한다.

type	설명	Role	설명
categorical	범주형	feature	목표에 영향을 주는 값
numeric	수치형	target	목표값
text	문자형	meta	목표에 영향을 주지 않으나 참고할만한 값
datetime	날짜	skip	무시해야 하는 값

변경 전	변경 후
	

③ Data Table 위젯을 이용하여 데이터 속성과 값을 확인한다.

Data Table 위젯을 이용하여 데이터와 데이터 속성을 확인한다. Visualize numeric values를 선택하면 숫자형 데이터를 막대로 시각화하여 볼 수 있고, color by instance classes를 선택하면 Target의 속성별로 색상을 다르게 표현할 수 있다.

color by instance classes 적용 전	color by instance classes 적용 후
	

02 데이터 시각화하기

밀알의 품종을 구분하는데 가장 큰 영향을 미치는 속성은 무엇일까? 데이터 시각화를 통해 기계 학습을 통해 밀알의 품종을 결정하는데 영향을 미치는 핵심 데이터 속성을 추출할 수 있다.

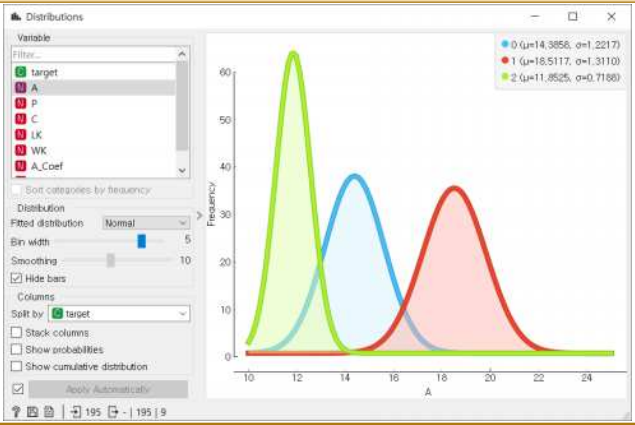
1 특성 통계표로 나타내기

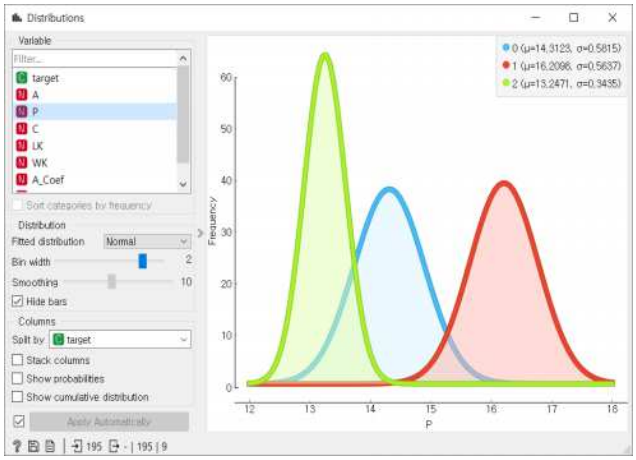
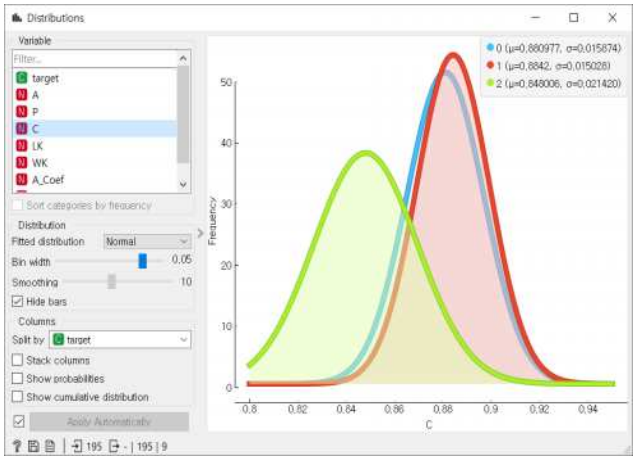
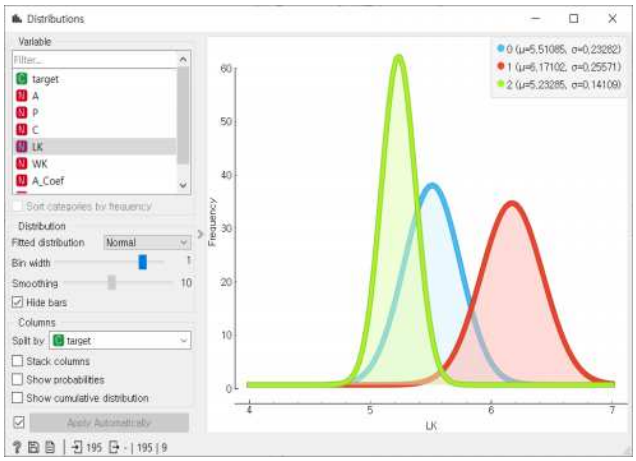
특성 통계표는 데이터 테이블을 구성하는 각 속성값의 통계정보와 도수분포를 동시에 볼 수 있는 그래프이다. 특성별 도수분포와 최솟값, 최댓값, 중앙값 등을 확인할 수 있다. 이를 통해 상대적으로 작은 값과 큰 값을 가지는 속성이 있는지, 전처리가 필요한 지도 확인할 수 있다. Data - Feature Statics 위젯을 클릭한 후 File 위젯과 연결한다. 속성 통계표의 아래쪽에 있는 color에서 속성을 선택하면 밀의 품종별로 통계치를 확인할 수 있다. 아래는 8개의 속성값을 통계분포도로 나타낸 것이다. 모든 데이터에서 결측치가 없고 값의 범위가 크게 차이나는 속성이 없으므로 데이터 전처리 과정은 거치지 않고 기계학습을 진행했다.

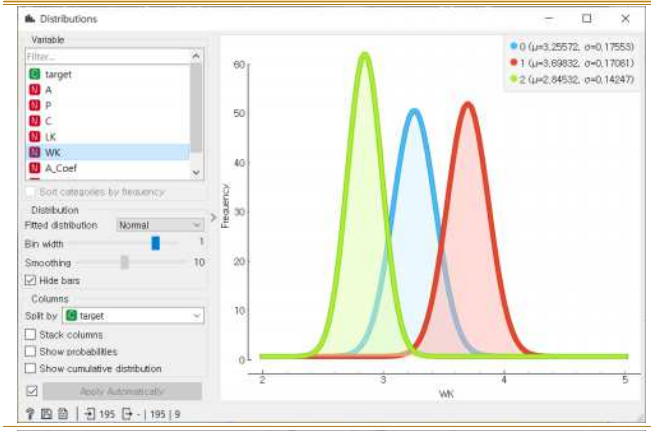
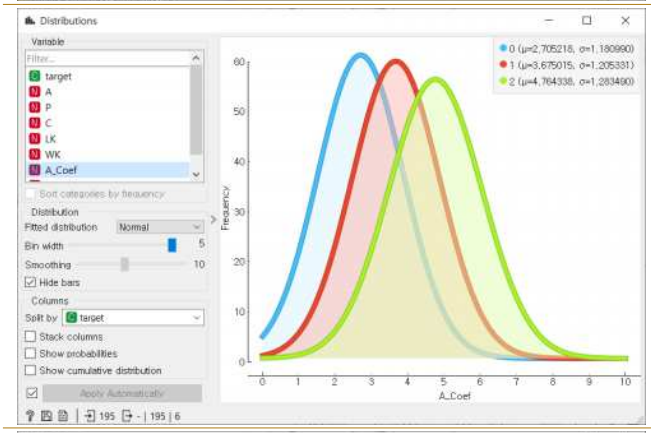
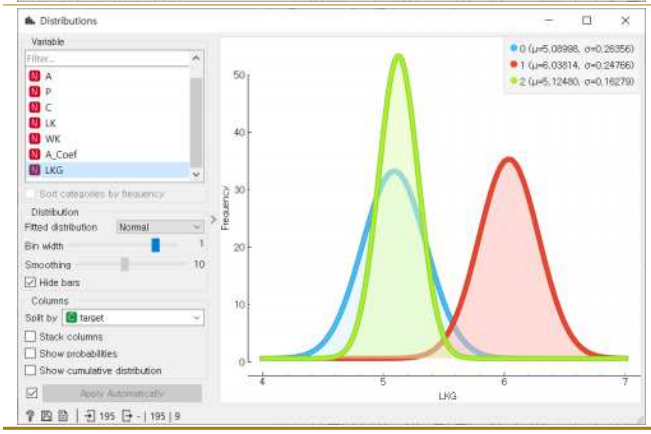
[그림 4-5] 훈련 데이터의 특성 통계표

2 Distributions로 데이터 탐색하기

Visualize - Distributions를 이용하여 데이터 속성별 분포를 시각적으로 파악할 수 있다. 분포도는 Distributions 위젯을 클릭하여 캔버스에 배치하고 File 위젯과 연결시킨다.

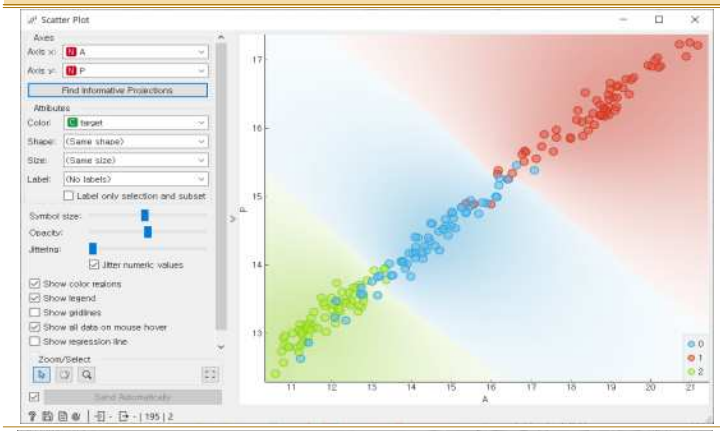
속성별 분포도	데이터 해석
	면적(A, area)를 기준으로 분포도를 살펴보면 세 가지 품종의 값의 평균값이 명확하게 구분되고 분포도 상의 겹치는 부분도 넓지 않아 분류의 속성으로 사용 가능하다.

속성별 분포도	데이터 해석
	둘레(P, perimeter)를 기준으로 분포도를 살펴보면 세 가지 품종의 평균값이 명확하게 구분되고 겹치는 부분의 면적도 적으므로 분류의 속성으로 사용 가능하다.
	조밀성(C, Compactness)을 기준으로 분포도를 살펴보면 0번과 1번의 평균값이 거의 차이가 없고 값의 분포도 거의 유사하여 단일 속성만으로는 2번 품종은 분류할 수 있으나 0번 품종과 1번 품종을 분류하기는 어렵다.
	낟알의 길이(LK, Length of Kernel)을 기준으로 분포도를 살펴보면 0번 품종과 2번 품종의 평균값의 차이가 크지 않고 겹치는 값이 많아 단일 속성만으로는 1번 품종을 분류할 수 있으나 0번 품종과 2번 품종을 분류하기 어렵다.

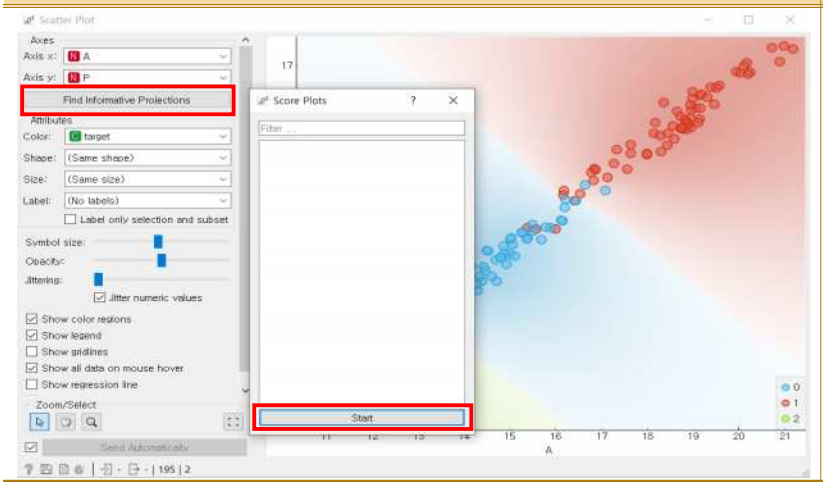
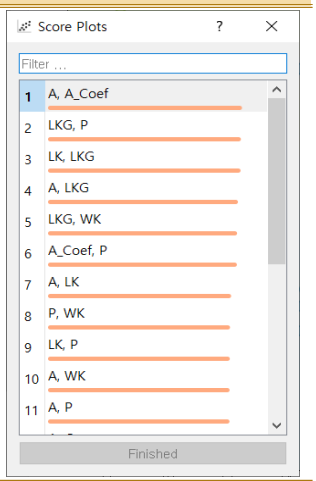
속성별 분포도	데이터 해석
	낟알의 길이(WK, Width of Kernel)을 기준으로 분포도를 살펴보면 세 가지 품종의 평균값의 차이는 크지 않으나 분산이 낮아 겹치는 면적이 적으므로 분류의 속성으로 사용 가능하다.
	비대칭 계수(A_Coef, asymmetry coefficient)를 기준으로 분포도를 살펴보면 세 가지 품종의 평균값의 차이가 크지 않고, 분산이 커서 겹치는 면적이 넓어 단일 속성만으로 3가지 품종을 분류하기에는 어렵다.
	낟알 홈의 길이(LKG, Length of Kernel Groove)을 기준으로 분포도를 살펴보면 0번과 2번의 품종의 평균값과 값의 분포가 유사하므로 분류 속성으로 사용하기는 어렵다.

3 산점도로 나타내기

산점도(Scatter Plot)는 x축과 y축으로 이루어진 2차원 평면 상에 데이터를 점으로 표현하여 속성 사이의 관계를 파악할 수 있다. Axis x, Axis y를 이용하여 직접 속성을 설정할 수도 있고, [Find Informative Projections]를 이용하여 유용한 속성 쌍을 선택할 수도 있다.

2가지 속성을 이용한 산점도	데이터 해석
	Distributions를 통해 탐색한 속성 중 품종을 잘 분류하는 속성으로 생각되는 면적(A, area), 둘레(P, Perimeter)를 이용해 산점도로 표현했다.
	Distributions를 통해 탐색한 속성 중 품종을 잘 분류하지 못하는 속성으로 생각되는 조밀성(C, compactness)과 비대칭 계수(A_Coef, asymmety coefficient)를 이용해 산점도로 표현했다.

우리가 사용한 데이터 세트는 속성 값이 총 7개밖에 되지 않아 일일이 데이터의 평균값과 분산 정도를 확인할 수 있지만, 데이터 세트의 속성의 종류가 많다면 일일이 시각화 결과를 확인하며 분류 속성을 찾는 것은 매우 어려울 것이다. 이 때는 [Axes]에 있는 [Find informative Projections]을 클릭하면 학습에 사용할 수 있는 핵심 속성을 추천받을 수 있다.

속성 추천	추천 결과
	

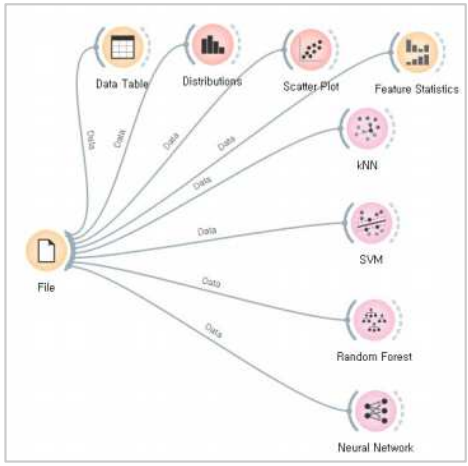
오렌지에서 추천한 속성을 이용해 분류한 결과를 산점도로 표시하면 다음과 같다.

1순위	2순위
	

03 모델 학습하고 성능 평가하자

1 모델학습

기계학습 알고리즘과 데이터를 연결하여 모델 학습한다. 오렌지3에서는 다양한 기계학습 알고리즘을 한꺼번에 연결하여 모델을 만들 수 있다. 여기서는 분류에 자주 사용하는 kNN, SVM, random Forest, Neural Network을 이용하여 모델을 구성하였다. Model 위젯 카테고리에서 해당하는 학습 알고리즘 위젯을 선택하여 File 위젯과 연결한다. 학습에 사용되는 속성의 수가 많지 않아 속성을 추출하는 작업 없이 학습을 진행하였다.



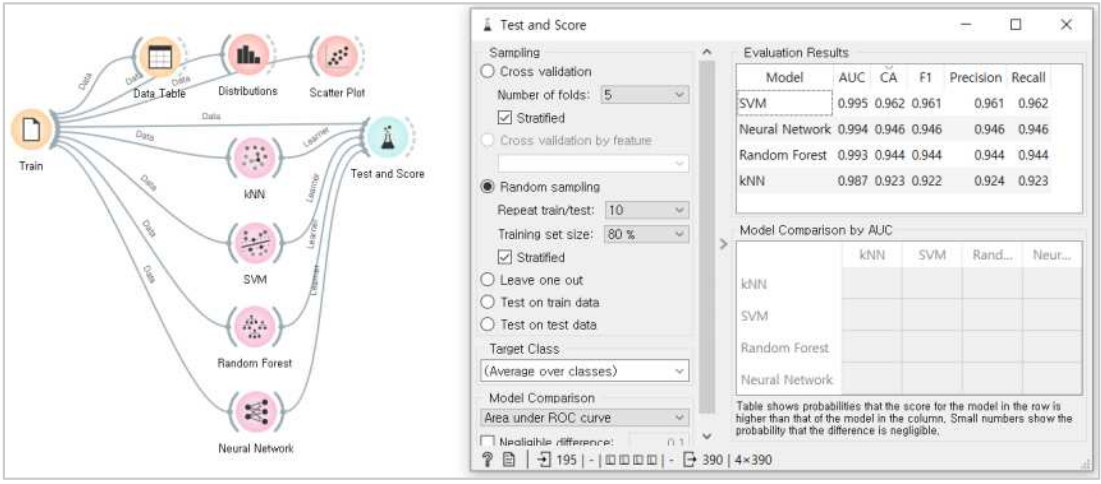
[그림 4-6] 분류 모델 만들기

2 성능 평가

Test and Score를 이용하여 모델의 성능을 평가해보자. Evaluate - Test and Score 위젯을 선택하고 File과 각 학습 알고리즘을 연결한다.

Test and Score 위젯에서 모델 학습과 테스트 데이터의 비율을 설정할 수 있다. 씨앗 데이터는 각 품종별 데이터가 70개이지만 그 중 65개를 훈련 데이터로 사용하고 나머지는 테스트 데이터로 사용하기 위해 데이터를 추출해 둔 상태이다. 그러므로 좀 더 많은 데이터를 이용해 학습시키기 위해 훈련 데이터셋의 크기를 80%로 설정했다.

샘플링 방식 중 Random Sampling은 전체 데이터를 섞어서 무작위로 훈련 데이터와 테스트 데이터를 분리한다. 또한 훈련과 테스트의 반복 횟수를 설정할 수 있다. 테스트 데이터를 이용한 계산을 쉽게 하기 위해 반복(repeat train/test)를 10회로 설정하였다. Test and Score 위젯을 더블클릭하면 학습 결과를 평가해 준다. 분류 유형에서는 AUC, CA, F1, Precision, Recall, LogLoss와 같은 성능 평가 지표가 있다.

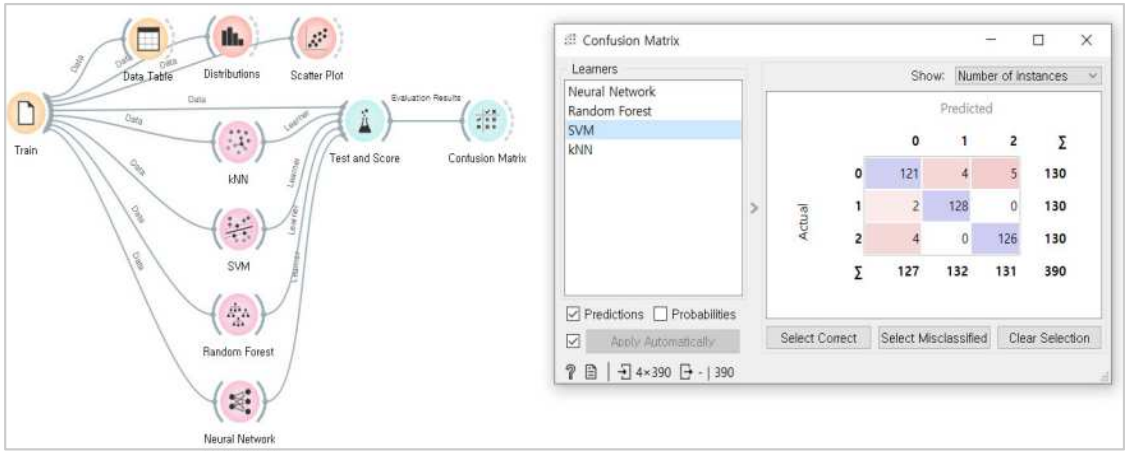


[그림 4-7] 훈련 데이터의 성능 평가 결과

분류 성능 평가 척도	상세 설명
AUC(Area under ROC)	가능한 모든 분류 임계값에 대한 종합적인 성능 측정값
CA(Classification accuracy)	올바르게 분류된 예(TN, TP)의 비율
Precision(정밀도)	Positive(양성)로 분류된 인스턴스 중 참 양성(True Positive)의 비율
Recall(재현율)	데이터의 모든 Positive(양성) 사례 중 참 양성(True Positive)의 비율
F1	Precision(정밀도)와 Recall(재현율)의 가중 조화 평균
LogLoss	모델 예측과 목표 값 간의 교차 엔트로피. 이 범위는 0에서 무한대까지이며 값이 낮을수록 모델의 품질이 더 높음을 나타냄

SVM의 분류의 정확도를 확인하기 위해 혼동 행렬(Confusion Matrix)를 사용해 보자.

Evaluate - Confusion Matrix 위젯을 선택하고 Test and Score 위젯과 연결시킨다. 위젯을 더블클릭하여 왼쪽에서 알고리즘을 선택하고 혼동 행렬을 확인한다. 여기서 테스트 데이터의 수가 390인 이유는 195개의 데이터 중 20%에 해당하는 39개의 데이터를 10번 반복하여 테스트했기 때문이다.



[그림 4-8] SVM모델의 혼동 행렬

SVM 알고리즘으로 만든 모델의 혼동 행렬을 살펴보면, 실제 1번 품종을 1번으로 예측한 경우가 121개, 2번 품종을 2번으로 예측한 경우가 128번, 3번 품종을 3번으로 예측한 경우가 126번이다. 데이터를 정확하게 분류하는 것은 성능 평가 지표 중 CA(Classification Accuracy)이다. 이 경우는

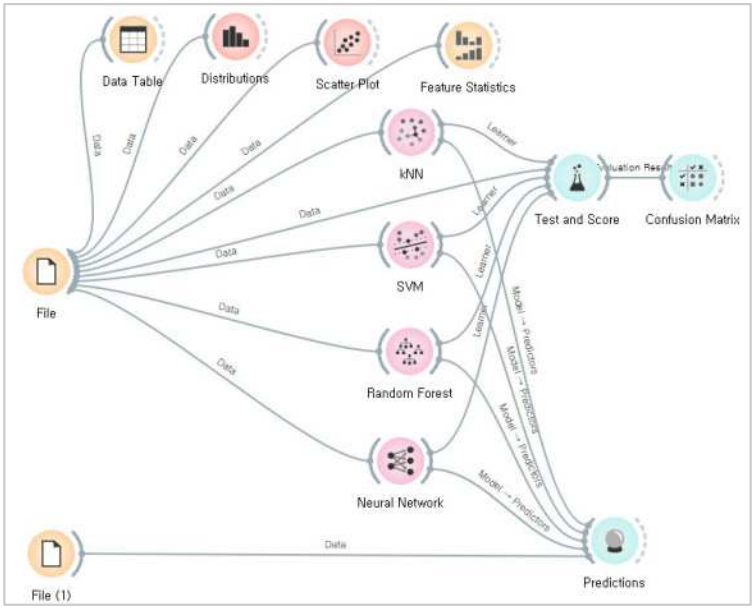
$$CA = \frac{121+128+126}{130+130+130} = \frac{375}{390} = 0.962$$

로 계산되는 것을 알 수 있다.

3 예측하기

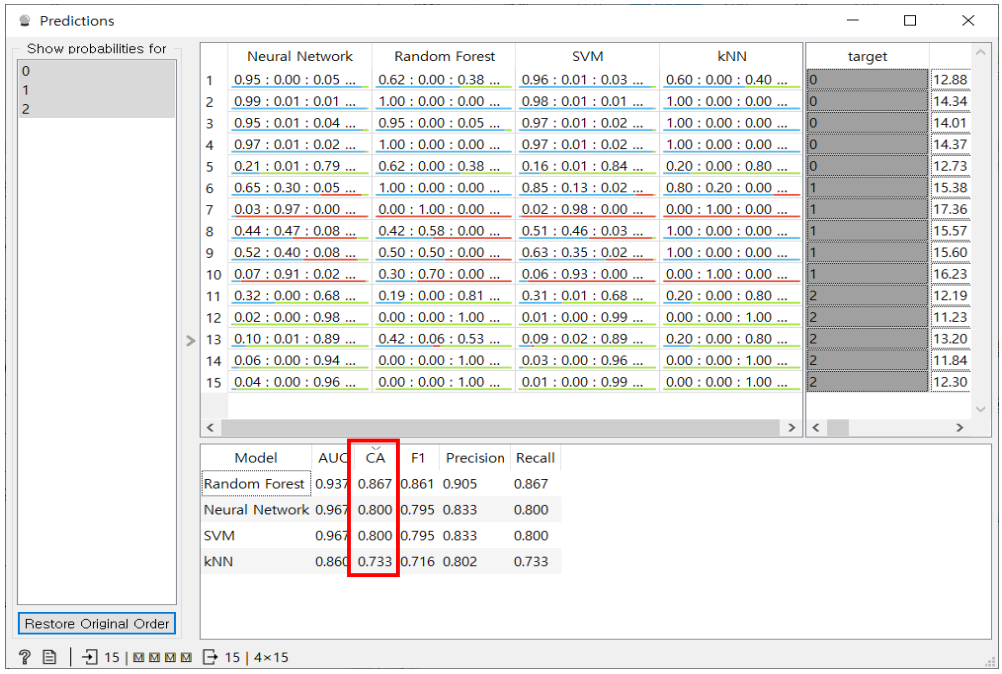
훈련 데이터를 이용해 학습한 모델을 이용하여 실제 테스트 데이터를 이용해 확인해 보자.

1) Data - File 위젯을 선택하여 캔버스에 배치한 후 File에 15개의 테스트 데이터 파일을 업로드한다.
2) Evaluate - Predictions 위젯을 선택하여 캔버스에 배치한다.
3) File과 기존 모델 생성에 사용했던 네 가지 학습 알고리즘을 모두 Predictions에 연결한다.
4) Predictions를 클릭하여 결과를 확인한다.



[그림 4-9] 테스트 데이터 예측 모델 만들기

Predictions 위젯을 눌러 결과를 확인해 보자. 왼쪽에는 품종의 종류가 제시되어 있고, 가운데는 각 알고리즘별로 값을 예측한 결과가 나와있다. 그리고 가장 오른쪽에는 테스트 데이터 값이 제시되어 있다. 아래쪽에는 테스트 데이터를 이용한 모델별 성능 평가 척도의 값이 제시되어 있다.



	Neural Network	Random Forest	SVM	kNN	target	
1	0.95 : 0.00 : 0.05 ...	0.62 : 0.00 : 0.38 ...	0.96 : 0.01 : 0.03 ...	0.60 : 0.00 : 0.40 ...	0	12.88
2	0.99 : 0.01 : 0.01 ...	1.00 : 0.00 : 0.00 ...	0.98 : 0.01 : 0.01 ...	1.00 : 0.00 : 0.00 ...	0	14.34
3	0.95 : 0.01 : 0.04 ...	0.95 : 0.00 : 0.05 ...	0.97 : 0.01 : 0.02 ...	1.00 : 0.00 : 0.00 ...	0	14.01
4	0.97 : 0.01 : 0.02 ...	1.00 : 0.00 : 0.00 ...	0.97 : 0.01 : 0.02 ...	1.00 : 0.00 : 0.00 ...	0	14.37
5	0.21 : 0.01 : 0.79 ...	0.62 : 0.00 : 0.38 ...	0.16 : 0.01 : 0.84 ...	0.20 : 0.00 : 0.80 ...	0	12.73
6	0.65 : 0.30 : 0.05 ...	1.00 : 0.00 : 0.00 ...	0.85 : 0.13 : 0.02 ...	0.80 : 0.20 : 0.00 ...	1	15.38
7	0.03 : 0.97 : 0.00 ...	0.00 : 1.00 : 0.00 ...	0.02 : 0.98 : 0.00 ...	0.00 : 1.00 : 0.00 ...	1	17.36
8	0.44 : 0.47 : 0.08 ...	0.42 : 0.58 : 0.00 ...	0.51 : 0.46 : 0.03 ...	1.00 : 0.00 : 0.00 ...	1	15.57
9	0.52 : 0.40 : 0.08 ...	0.50 : 0.50 : 0.00 ...	0.63 : 0.35 : 0.02 ...	1.00 : 0.00 : 0.00 ...	1	15.60
10	0.07 : 0.91 : 0.02 ...	0.30 : 0.70 : 0.00 ...	0.06 : 0.93 : 0.00 ...	0.00 : 1.00 : 0.00 ...	1	16.23
11	0.32 : 0.00 : 0.68 ...	0.19 : 0.00 : 0.81 ...	0.31 : 0.01 : 0.68 ...	0.20 : 0.00 : 0.80 ...	2	12.19
12	0.02 : 0.00 : 0.98 ...	0.00 : 0.00 : 1.00 ...	0.01 : 0.00 : 0.99 ...	0.00 : 0.00 : 1.00 ...	2	11.23
13	0.10 : 0.01 : 0.89 ...	0.42 : 0.06 : 0.53 ...	0.09 : 0.02 : 0.89 ...	0.20 : 0.00 : 0.80 ...	2	13.20
14	0.06 : 0.00 : 0.94 ...	0.00 : 0.00 : 1.00 ...	0.03 : 0.00 : 0.96 ...	0.00 : 0.00 : 1.00 ...	2	11.84
15	0.04 : 0.00 : 0.96 ...	0.00 : 0.00 : 1.00 ...	0.01 : 0.00 : 0.99 ...	0.00 : 0.00 : 1.00 ...	2	12.30

Model	AUC	CA	F1	Precision	Recall
Random Forest	0.937	0.867	0.861	0.905	0.867
Neural Network	0.967	0.800	0.795	0.833	0.800
SVM	0.967	0.800	0.795	0.833	0.800
kNN	0.860	0.733	0.716	0.802	0.733

[그림 4-10] 테스트 데이터의 예측 결과

예측 결과를 살펴보면 CA값이 가장 높은 알고리즘은 Random Forest이다. 학습에서 CA값이 가장 높았던 알고리즘이 SVM이었으나 학습하지 않은 새로운 데이터를 예측하는 것은 Random Forest보다 성능이 조금 떨어진다. 다양한 인공지능 모델에서 학습에서는 좋은 성능을 나타내지만 새로운 값을 이용해 예측할 때는 성능이 떨어지는 경우를 종종 볼 수 있다. 이러한 현상은 훈련 데이터를 너무 잘 학습하여 최적화된 상태라 새로운 값을 제대로 예측하지 못하는 것이다. 이러한 현상을 과적합(overfit)이라고 한다. 그림을 통해 과적합이 어떤 상태인지 이해할 수 있을 것이다. 이러한 경우에 학습 모델을 학습 시키기 위해 더 많은 데이터를 추가하여 새롭게 학습하거나 반복 횟수를 늘려서 학습을 시켜볼 필요가 있다.

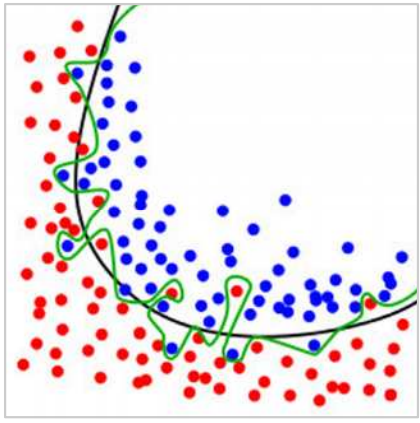


그림 출처: https://ko.wikipedia.org/wiki/과적합

이와 같이 훈련 데이터의 성능과 테스트 데이터의 성능이 우수한 모델이 다르게 나타난다면 테스트 데이터를 더 잘 분류한 Random Forest모델을 이용하는 것이 더 나을 것이다.

04. 밀알의 크기로 밀알의 종류를 구분할 수 있을까?

정리하기

밀알에 대한 수치 정보를 이용해 품종을 분류하는 인공지능을 만들기 위해서, 밀알의 수치 정보를 이용해 훈련 데이터와 테스트 데이터로 분류하고 이를 이용해 Kama, Rosa, Canadian과 같은 밀알의 품종을 분류할 수 있는 기계학습 모델을 만들었다. 모델의 학습 결과 SVM모델의 성능이 가장 우수하였으나 테스트 결과는 Random Forest모델이 가장 우수한 것으로 나타났다. 이처럼 훈련 데이터의 성능과 테스트 데이터의 결과 성능이 우수한 모델이 다르게 나타난다면 학습 결과를 이용해 새로운 값을 더 잘 분류하는 모델을 선택하는 것이 효율적일 것이다. 이 경우에는 SVM 모델 보다는 Random Forest 모델을 선택하는 것이 좋을 것이다.

기계 학습 알고리즘

◆ SVM(support vector machine) 알고리즘이란?

서포트 벡터 머신은 서로 다른 분류에 속한 데이터 간에 간격이 최대가 되는 선(또는 평면)을 찾아 이를 기준으로 데이터를 분류하는 모델이다. 우측의 그림에서 흰색 원과 검은색 원은 서로 다른 분류를 나타낸다. 서포트 벡터 머신은 각 분류에 속하는 데이터로부터 같은 간격으로, 그리고 최대로 멀리 떨어진 선 또는 평면을 찾는다. 이러한 선 또는 평면을 최대 여백 초평면이라고 하고 이 평면이 분류를 나누는 기준이 된다. 그림에서는 실선으로 그어진 선이 최대여백 초평면에 해당된다. 그리고 이 실선과 가장 가까운 각 분류에 속하는 점들을 서포트 벡터라고 한다.

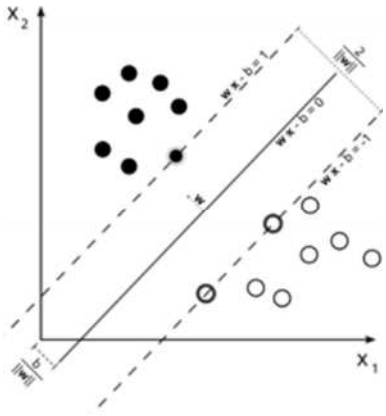


그림 출처:
https://ko.wikipedia.org/wiki/서포트_벡터_머신

[참고 문헌]

1. 손원성 외 3명(2021). 오렌지3로 알아가는 머신러닝 데이터 분석. 홍릉
2. 서울과학종합대학원 디지털혁신처(2021). 3시간 만에 배우는 인공지능 데이터분석. 오렌지. 서울경제경영
3. 조태호(2020). 모두의 딥러닝 개정 2판. 길벗
4. 서민구(2014). R을 이용한 데이터 처리&분석 실무. 길벗
5. 밀-위키백과. https://ko.wikipedia.org/wiki/%EB%B0%80
6. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/seeds
7. wheat kernels. https://wheatkernels.weebly.com/
8. 오렌지. https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/data/rank.html
9. 밀 낟알 사진. https://agclassroomstore.com/wheat-kernel-samples/
10. 과적합 상태. https://ko.wikipedia.org/wiki/과적합
11. 서포트 벡터 머신. https://ko.wikipedia.org/wiki/서포트_벡터_머신

MEMO

05. 흉부 영상으로 코로나19를 판단할 수 있을까?

금오고등학교 교사 박 윤 희

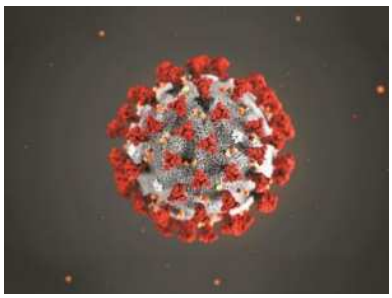
학습 진행 과정

1단계	데이터 준비	- 사용 데이터: COVID-19 Patients Lungs X Ray Images 10000 - 수집: 캐글 - 데이터 편집: 데이터 폴더(카테고리) 수정
2단계	데이터 불러오기	- 학습 데이터 불러오기 - 이미지 데이터 임베딩(Embedding)하기
3단계	속성 추출	- Rank로 주요 속성 추출하기
4단계	데이터 시각화	- 시각화 도구를 이용한 데이터 탐색 - 사용한 시각화 도구: Radviz
5단계	모델 학습	- 분류를 이용한 모델 학습 - 사용된 알고리즘: Logistic Regression, SVM, Neural network
6단계	성능 평가	- Test and score를 이용한 성능 평가 - Confusion Matrix를 이용한 성능 평가
7단계	예측	- Predictions을 이용한 테스트 데이터로 예측하기

학습 내용 요약

데이터 종류	문제 유형	사용된 학습 알고리즘	성능 평가 도구
비정형 데이터(이미지)	분류	Logistic Regression, SVM, Neural Network	혼동 행렬

문제 상황



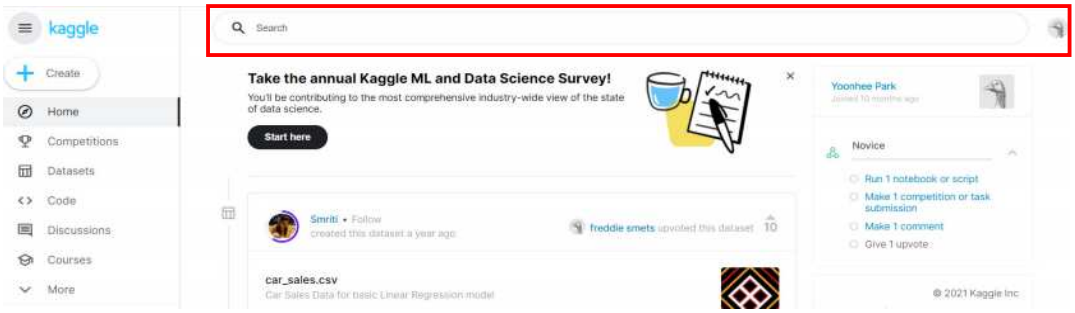
코로나 바이러스는 포유류와 새를 포함한 동물에게도 감염을 일으키지만 사람에게도 흔하게 감염을 일으키는 바이러스의 한 종류로 감기를 포함하여 인후염, 비염 등의 상부 호흡기 감염을 일으키는 호흡기 바이러스 중 하나이다. 이 중 2019년 12월 중국 우한에서 집단으로 발생한 폐렴의 원인을 조사하는 과정에서 밝혀진 2019년 신종 코로나 바이러스로 인하여 세계 곳곳에서 코로나 바이러스에 감염된 환자들이 발생하고 사망자가 발생하고 있다.

2019 신종코로나 바이러스는 호흡기 바이러스이기 때문에 기본적으로 일반 감기나 독감, 기관지염, 폐렴의 증상과 동일하며, 급성호흡기감염 때 발생할 수 있는 발열, 오한, 근육통이 발생하고, 호흡기 증상으로 인후통, 콧물, 기침, 객담이 발생할 수 있다. 하지만 숨이 가쁜 느낌이나 호흡 곤란을 느끼게 되면 폐렴이 발생하였을 가능성이 높고 중증폐렴으로 진행될 위험성이 있고 제 때 치료를 받지 못하면 호흡곤란으로 인해 사망에 이를 수도 있다. 코로나 바이러스를 정확하게 진단하기 위해 유전자 증폭 검사(RT-PCR)를 시행하고 있다. 하지만 이 방식으로 다른 코로나 바이러스와 2019 신종코로나 바이러스를 매우 정확하게 구분할 수 있으나 검사 결과를 확인하기 위해서는 6시간 정도 소요가 되며, 대부분은 1~2일 이내에 결과를 확인하게 된다. 하지만 코로나 바이러스가 호흡기 질환을 유발하므로 RT-CPR 검사의 결과를 기다리는 동안 또는 RT-CPR의 결과가 음성이고 코로나 바이러스 증상이 있는 감염자를 진단하기 위해 흉부 영상을 사용하여 감염자를 판단할 수 있을 것이다. 감염 의심자의 흉부 영상(X-ray)을 이용하여 코로나 감염 여부를 인공지능으로 구분할 수 있을까?

01 데이터 준비하기

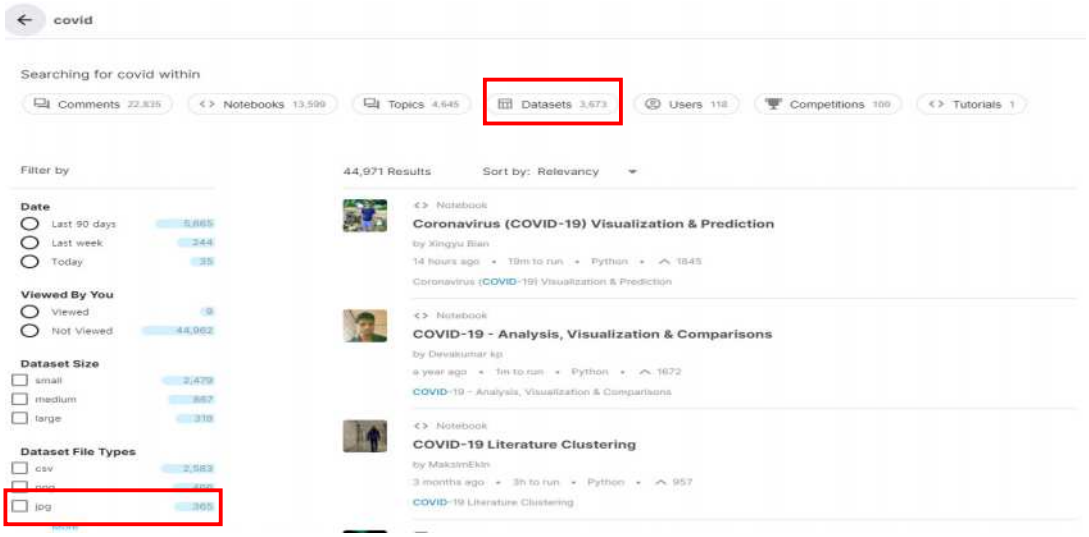
1 corona virus 데이터 세트

코로나 환자의 폐 X-ray 사진을 이용해 코로나 환자와 정상을 분류하는 모델을 생성하기 위해 캐글에서 제공하는 데이터 세트를 다운로드 받는다.

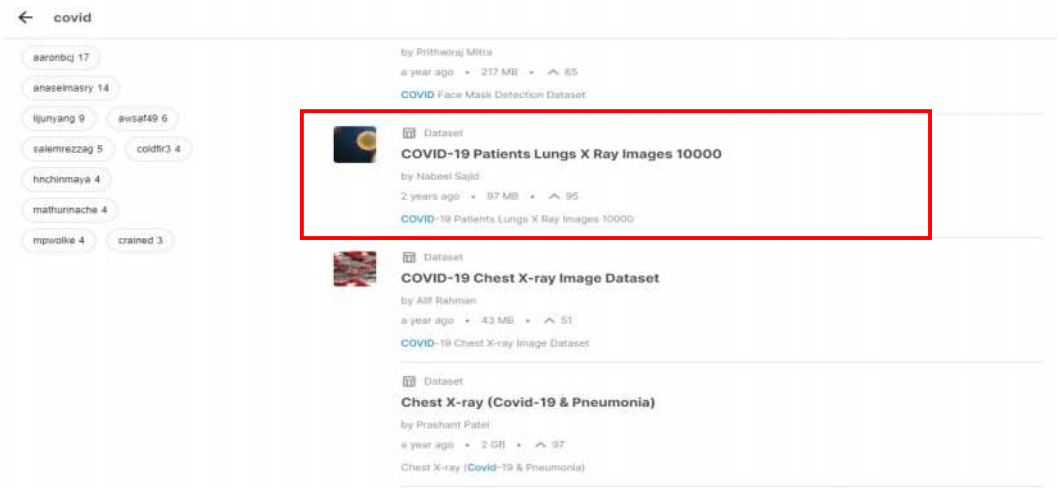


[그림 5-1] 캐글에 접속하여 데이터 검색하기

검색창에 covid를 입력하면 다양한 내용이 검색된다. 검색 내용 중 인공지능 학습에 필요한 데이터가 필요하므로 유형은 [dataset]을 선택하고, 데이터 세트의 파일 유형을 이미지 형태인 [jpg]로 설정한다. 다음과 같이 조건을 설정하여 이미지로 된 데이터 세트를 검색한 후 'COVID-19 Patients Lungs X Ray Images 10000'를 클릭하여 모델 생성에 필요한 데이터 세트를 다운로드 받는다.

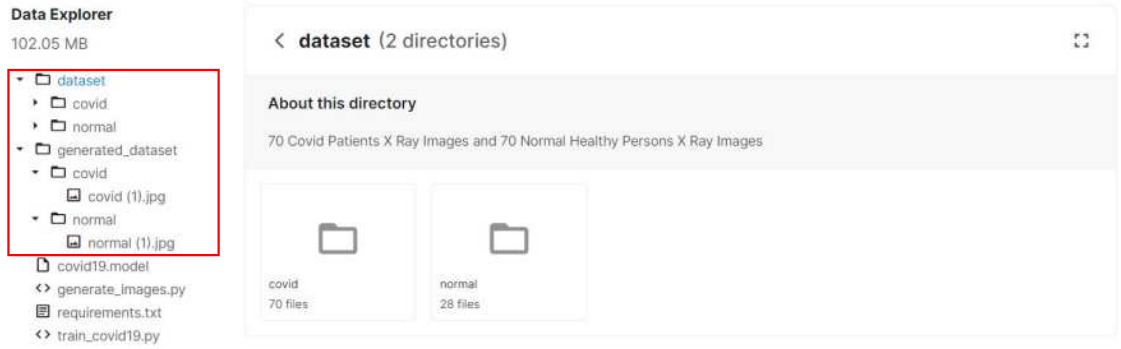


[그림 5-2] 데이터 검색을 위한 설정



[그림 5-3] 조건에 맞게 검색된 데이터 목록

Data Explorer를 이용하여 데이터셋의 구성을 확인하면 dataset과 generated_dataset으로 구성되어 있다. dataset에는 70개의 코로나 환자의 폐 X-ray 사진과 28개의 정상 폐 X-ray 사진이 있으며, generated_dataset에는 1개의 코로나 환자의 폐 X-ray 사진과 1개의 정상 폐 X-ray 사진이 있다.



[그림 5-4] 데이터 세트의 데이터 구성

기계 학습을 위해 dataset을 훈련 데이터로 사용하고, generated_dataset을 테스트 데이터로 사용하며, 모델을 생성할 때 혼란이 없도록 dataset은 dataset(train)으로 generated_dataset은 dataset(test)로 폴더명을 변경한다.

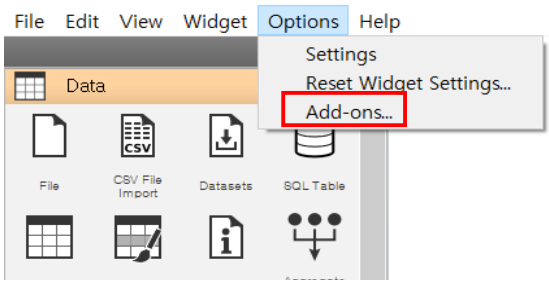
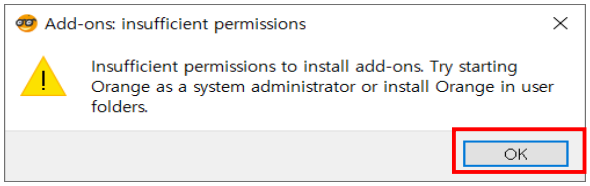
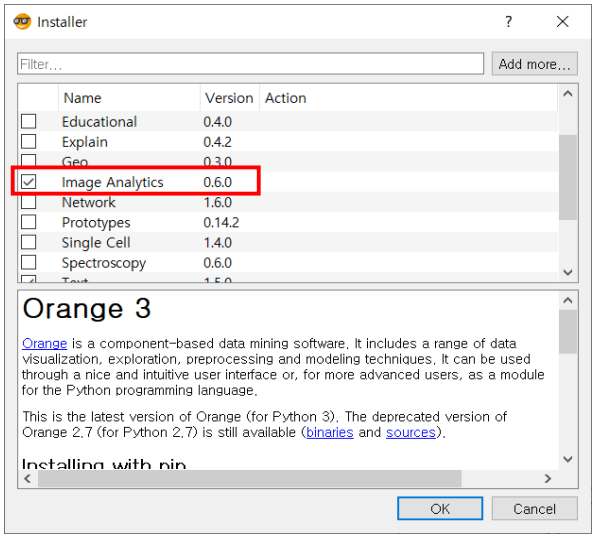
변경 전	변경 후
이름 dataset generated_dataset covid19.model generate_images requirements train_covid19	이름 dataset(test) dataset(train) covid19.model generate_images requirements train_covid19

이미지 데이터를 분류하기 위해서는 Target값으로 사용할 카테고리 값이 있어야 한다. 보통은 폴더명을 카테고리 값으로 사용하는데 이 데이터 세트에는 covid와 normal로 폴더가 구성되어 있어 따로 폴더명을 수정할 필요는 없다.

2 데이터 불러오기

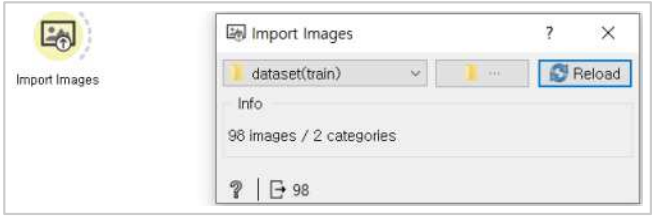
① 이미지 분석 위젯 추가하기

이미지 데이터를 불러오기 위해서는 오렌지3의 이미지 분석 위젯을 추가해야 한다. 이 때 사용하는 컴퓨터가 인터넷에 연결된 상태일 때 위젯을 다운로드 받아 설치할 수 있다. 이미지 분석 위젯을 추가하는 방법은 아래와 같다.

단계	설명
	[메뉴] - [Options]- [Add-ons..]를 선택한다.
	설치 허가와 관련된 내용을 [OK]를 눌러 확인하고 설치를 진행한다.
	설치 가능한 부가기능 목록에서 [Image Analytics]를 선택하여 설치를 진행한다.

② 이미지 데이터 불러오기

이미지 데이터를 불러오기 위해 Image Analytics - Import Image를 선택하여 캔버스(작업 공간)에 배치한 후 위젯을 더블클릭하여 학습에 필요한 데이터를 로드한다.



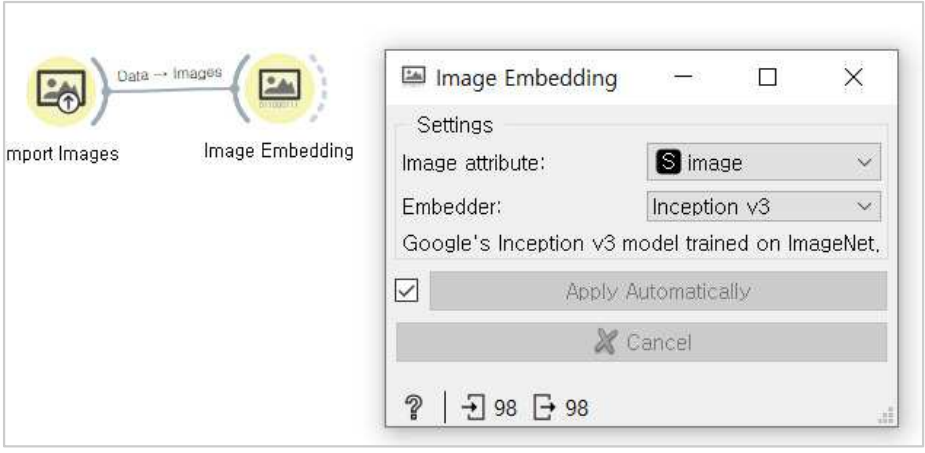
[그림 5-5] 이미지 불러오기

02 데이터 전처리하고 탐색하자

1 이미지 데이터 임베딩 하기

사람은 이미지 그 자체를 인식하지만 컴퓨터는 이미지를 숫자의 배열로 인식하기 때문에 이를 변환하는 과정이 필요하고 이 과정을 이미지 임베딩(Image Embedding)이라고 한다. 이미지를 임베딩하기 위해서는 Image Analytics - Image Embedding을 선택하여 Import Image 위젯과 연결한다. 해당 위젯을 더블클릭하면 설정을 확인할 수 있다. 여기서 Embedder를 이용해 이미 훈련된 이미지 인식 모델을 선택하여 사용할 수 있다.

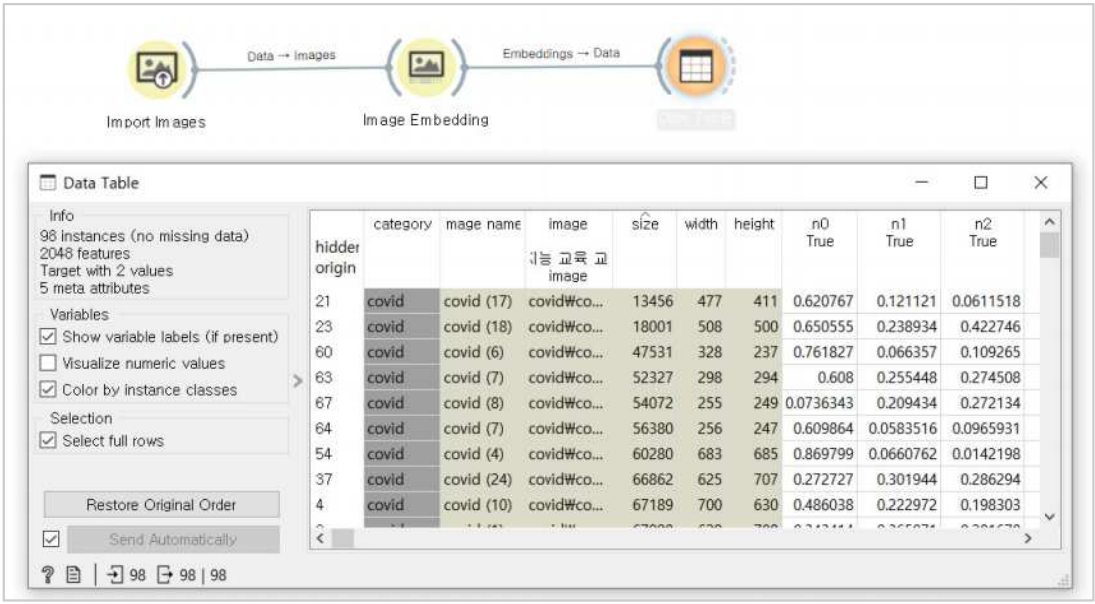
임베더	설명
SqueezeNet	ImageNet에서 훈련된 이미지 인식을 위한 작고 빠른 모델 (Local)
Inception v3	ImageNet에서 훈련된 Google의 Inception v3 모델 (기본값)
VGG-16	ImageNet에서 훈련된 16계층 이미지 인식 모델
VGG-19	ImageNet에서 훈련된 19계층 이미지 인식 모델
Painters	예술 작품 이미지에서 화가를 예측 하도록 훈련된 모델
DeepLoc	효모 세포 이미지를 분석하도록 훈련된 모델



[그림 5-6] 이미지 임베딩하기

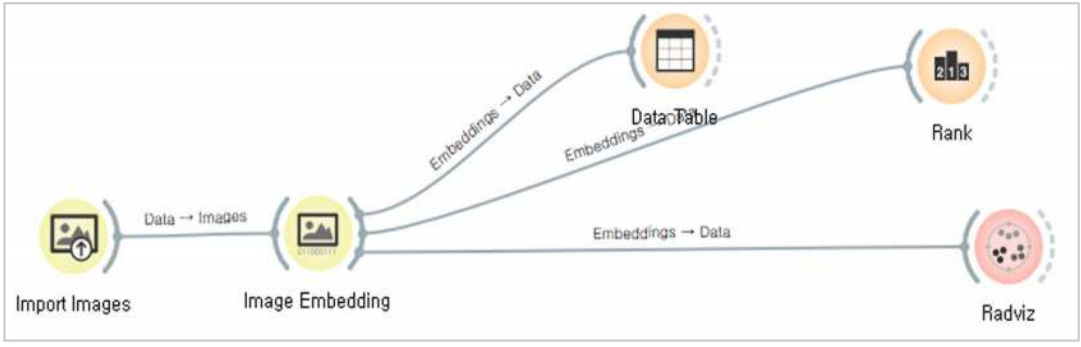
2 이미지 속성 확인하기

Data - Data Table 위젯을 선택하여 Image Embedding에 연결한 후 로드된 이미지 정보를 확인한다. 제시된 데이터 속성 중 category는 target 값으로 사용되는 속성이다. 훈련 데이터 폴더 하위에 분류 항목을 폴더로 구성해야 카테고리로 인식되어 학습에 사용할 수 있다. 이미지 임베딩이 과정을 거쳤기 때문에 이미지가 숫자값으로 변환된 것을 확인할 수 있다.



[그림 5-7] 임베딩한 이미지 데이터 확인하기

3 이미지 데이터 시각화하기

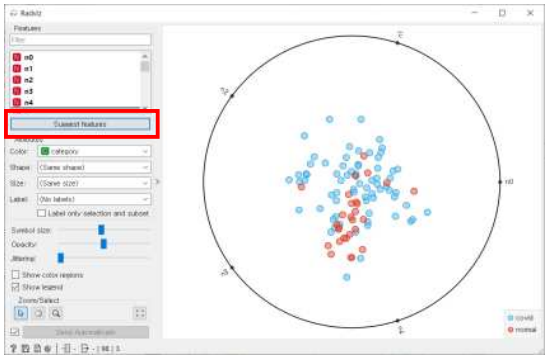
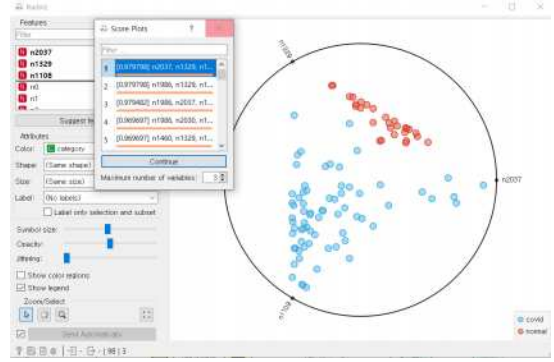
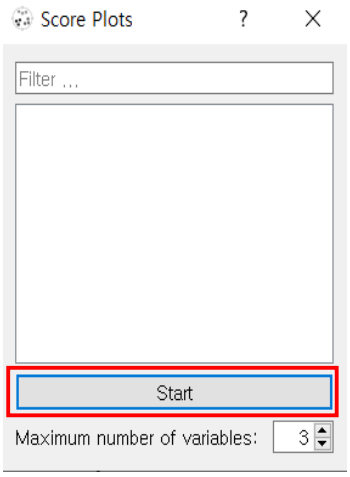
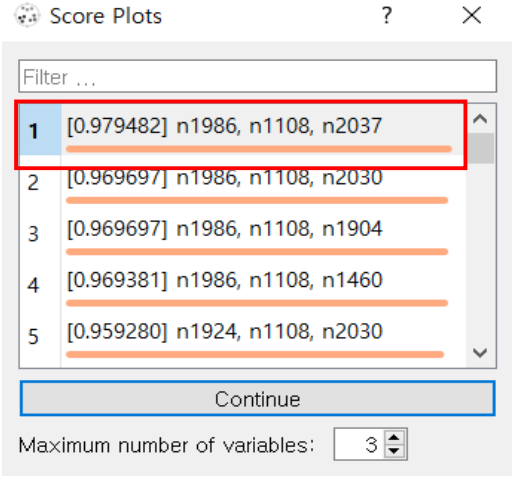


[그림 5-8] Radviz 위젯을 이용한 데이터 시각화

Visualize - Radviz 위젯을 캔버스에 배치하고 Image Embedding 위젯과 연결한 후 Radviz 위젯을 더블클릭하면 결과를 확인할 수 있다. Radviz는 3개 이상의 변수에 의해 정의된 데이터를 2차원 투영으로 표시할 수 있는 비선형 다차원 시각화 기법이다. 시각화된 변수는 단위 원의 둘레에 동일한 간격의 점으로 표시된다. 기본적으로 3개의 속성을 사용하여 분류한 결과가 나타나게 되는데 굵은 줄 아래에 있는 속성을 [Add]하면 해당 속성이 Feature에 추가된다. 임베딩한 이미지는 2048개의 속성으로 이루어져 있는데 그 중 두 개의 클래스를 분류할 수 있는 주요 속성의 조합을 찾아내는 것은 매우 어려운 일이다. 이 때 [Suggest Features]를 이용하면 손쉽게 주요 속성의 조합을 찾아낼 수 있다. [Suggest

Features]의 [Start]를 눌러 속성 조합을 찾고, 적당한 시점에 [Stop]을 눌러 속성 조합을 찾는 것을 중지한다. 2048개의 속성 중 3개의 속성 조합을 모두 찾는 것은 매우 시간이 많이 걸리기 때문이다.

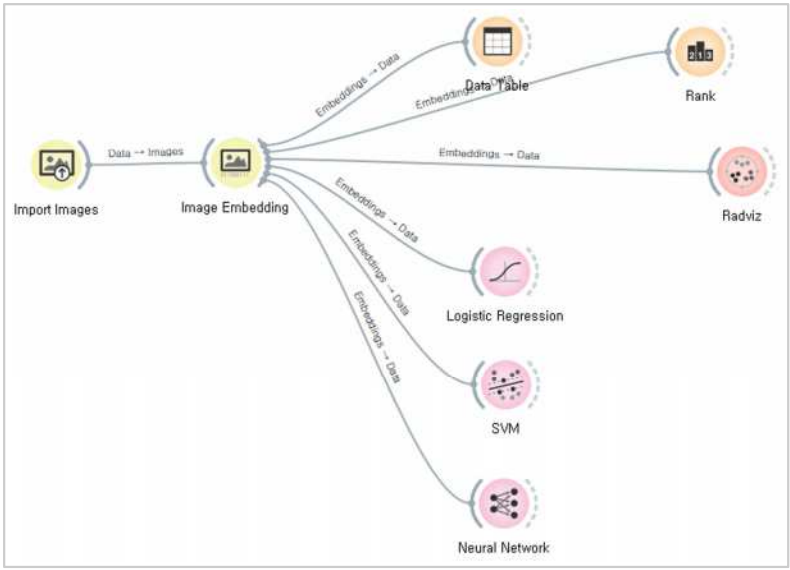
[표 5-1] Image Embedder의 종류 및 설명

Radviz 실행 초기 화면	Suggest Features를 사용한 시각화 결과
	
Suggest Features 시작하기	**Suggest Features 실행 결과**
	

03 모델 학습하고 성능 평가하자

1 모델 학습하기

지도 학습 모델 중 분류 문제를 해결하기 위해서 사용할 수 있는 다양한 학습 알고리즘이 있는데 이 중 Logistic Regression, SVM, Neural Network를 이용해 보자. Model – Logistic Regression, SVM, Neural Network 위젯을 선택하여 캔버스에 배치하고 Image Embedding과 연결한다.



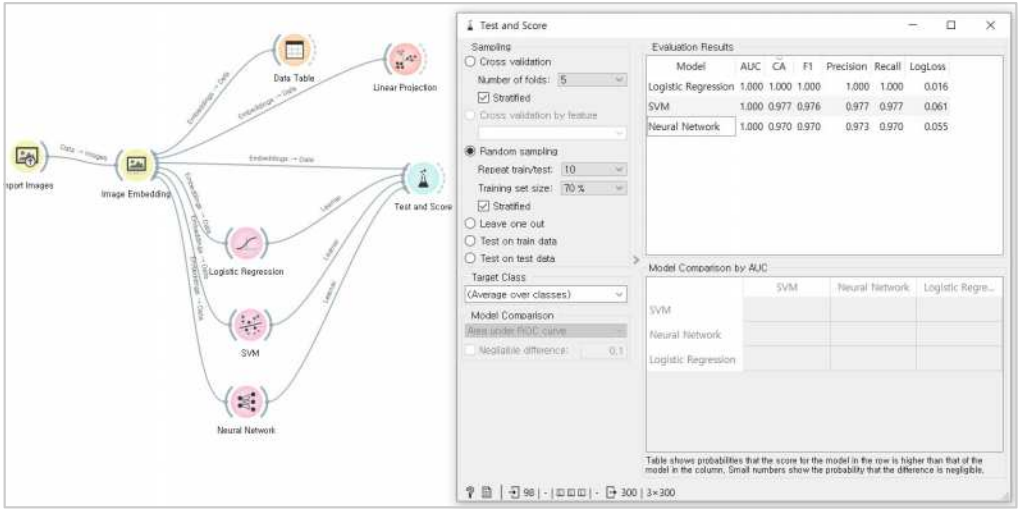
[그림 5-9] 분류 모델 만들기

2 성능 평가

기계학습 모델과 이미지 데이터를 연결하여 모델을 학습한 후 성능을 평가할 수 있다.

Evaluate - Test and Score 위젯을 선택하고 File과 각 학습 알고리즘을 연결시키면 모델별 성능 평가 척도가 수치로 제시된다.

성능을 평가하기 위해서 학습에 사용한 데이터 중 일부 데이터를 임의로 추출하여 성능을 평가하는데 사용할 수 있다. Test and Score 위젯을 더블클릭하면 테스트 데이터의 비율을 변경할 수 있다. 이 모델에서는 학습에 사용한 데이터 중 30%를 무작위로 추출하여 테스트하였고 이러한 테스트를 총 10회 반복하도록 설정했다.



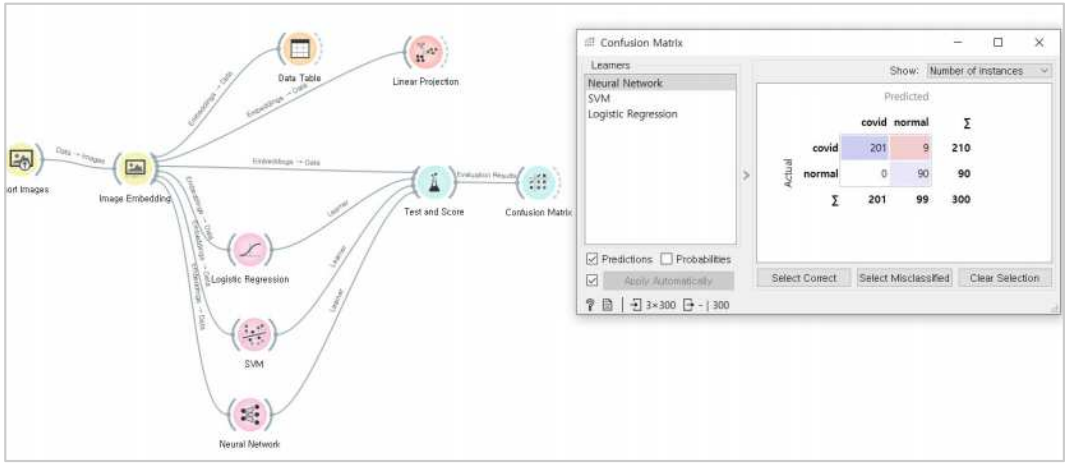
[그림 5-10] 분류 모델 학습 후 평가하기

3개의 학습 모델의 성능을 평가한 결과 정확도(CA) 순으로 모델을 나열하면 Logistic Regression, SVM, Neural Network 순으로 나타났다. Logistic Regression의 경우 정확도가 1.000으로 나왔는데 이는 코로나 환자와 정상의 경우를 모두 정확하게 분류했다는 뜻이다. 성능을 평가하는 척도는 정확도 이외에 다양한 지표가 존재하는데 각 지표에 대한 설명은 다음과 같다.

[표 5-2] 분류 성능 평가 척도와 설명

분류 성능 평가 척도	설명
AUC(Area under ROC)	가능한 모든 분류 임계값에 대한 종합적인 성능 측정값
CA(Classification accuracy)	올바르게 분류된 예(TN, TP)의 비율
Precision(정밀도)	Positive(양성)로 분류된 인스턴스 중 참 양성(True Positive)의 비율
Recall(재현율)	데이터의 모든 Positive(양성) 사례 중 참 양성(True Positive)의 비율
F1	Precision(정밀도)와 Recall(재현율)의 가중 조화 평균
LogLoss	모델 예측과 목표 값 간의 교차 엔트로피. 이 범위는 0에서 무한대까지이며 값이 낮을수록 모델의 품질이 더 높음을 나타냄

혼동 행렬을 이용하여 성능 평가 척도를 계산하는 방법을 이해해 보기 위해 Neural Network의 혼동 행렬(confusion matrix)을 확인해 보자. Evaluate - confusion Matrix 위젯을 선택하고 Test and Score와 연결시킨다. 위젯을 더블클릭하여 왼쪽에서 알고리즘을 선택하고 혼동 행렬을 확인한다. 여기서 테스트 데이터의 수가 300인 이유는 98개의 데이터 중 30%에 해당하는 30개의 데이터를 10번 반복하여 테스트했기 때문이다.



[그림 5-11] Neural Network의 혼동 행렬

Neural Network 모델의 혼동 행렬을 살펴보면 [그림 5-11]과 같다. 이 중 분류 정확도(CA)를 계산해 보자. CA는 올바르게 분류된 것의 예이며 전체 데이터 중 코로나 환자를 코

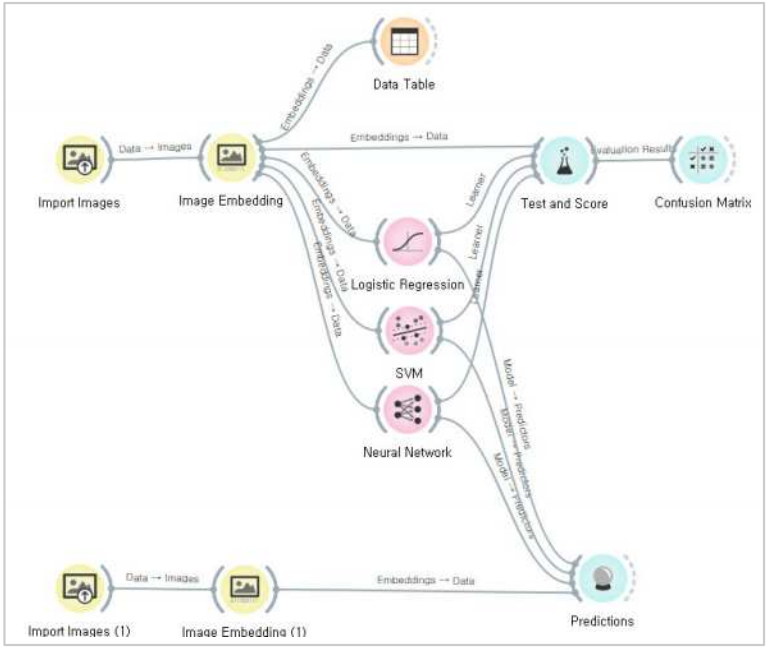
로나 환자로, 정상을 정상으로 분류한 비율에 해당한다. 혼동 행렬에서 실제 코로나 환자를 코로나 환자로 예측한 경우를 TP(True Positive), 실제 코로나 환자를 정상으로 예측한 경우 FN(False Negative), 실제 정상을 코로나 환자로 예측한 경우 FP(False Positive), 실제 정상을 정상으로 예측한 경우 TN(True Negative)라고 한다.

예측 데이터 / 실제 데이터	코로나 환자(Positive)	정상(Negative)
코로나 환자(True)	TP	FN
정상(False)	FP	TN

$$CA = \frac{TP + TN}{TP + FN + FP + TN} = \frac{201 + 90}{201 + 9 + 0 + 90} = \frac{291}{300} = 0.97$$

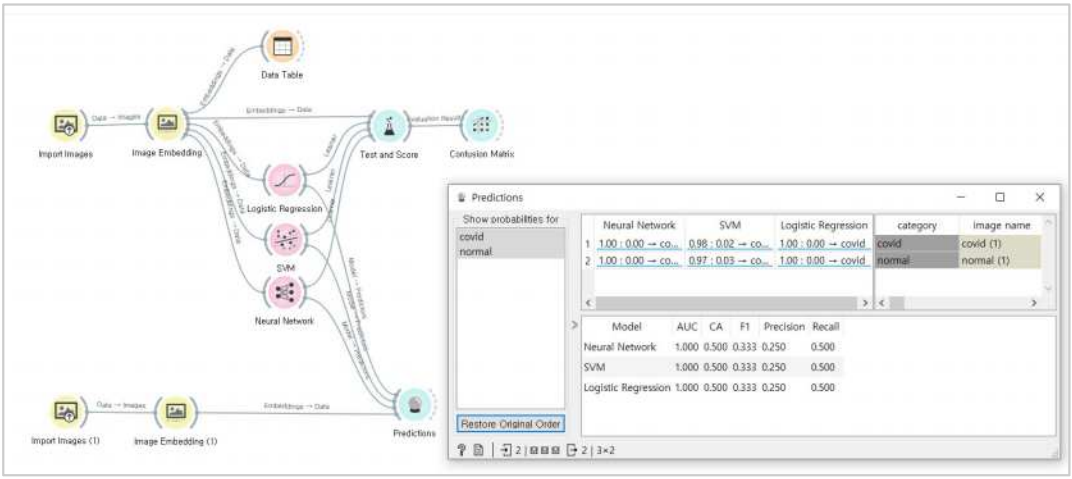
그렇다면 실제 테스트 데이터를 이용해 확인해 보자.

1) Image Analytics - Import Image 위젯을 선택하여 캔버스에 배치한 후 dataset(test)를 로드한다.
2) Image Analytics - Image Embedding 위젯을 선택하여 Import Image 위젯에 연결한다.
3) Evaluate - Predictions 위젯을 선택해 캔버스에 배치한다.
4) Image Embedding과 기존 모델 생성에 사용했던 3가지 학습 모델을 모두 Predictions에 연결한다.
5) Predictions를 클릭하여 결과를 확인한다.



[그림 5-12] 테스트 데이터 예측 모델 만들기

Predictions 위젯을 눌러 결과를 확인해 보자. 카테고리는 실제 값이고 각 모델별로 예측 값이 표시되어 있다. 두 개의 데이터를 이용해 테스트를 해 본 결과 두 개의 사진을 모두 코로나 환자의 사진이라고 판단하였고 정확도는 0.5가 나왔다. 이 모델의 성능을 평가할 때는 세 가지 모델 모두 97%가 넘는 정확도를 보인 것에 비하면 실제 테스트 결과는 좋지 않음을 알 수 있다. 이는 테스트에 사용된 데이터의 양이 너무 적어 하나만 잘못 예측해도 정확도가 절반으로 줄어들기 때문이다.



[그림 5-13] 예측 결과 확인

그렇다면 테스트 데이터를 이용한 분류 성능을 향상시키기 위해서는 어떻게 해야 할까? 모델을 생성하기 위해 사용한 훈련 데이터를 늘려주면 더 정확하게 분류할 수 있다. 해당 모델을 생성하기 위해 사용한 훈련 데이터는 총 98개 정도이다. 이보다 더 많은 훈련 데이터를 사용하여 학습한다면 더 나은 테스트 결과를 얻을 수 있을 것이다.

05. 흉부 영상으로 코로나19를 판단할 수 있을까?

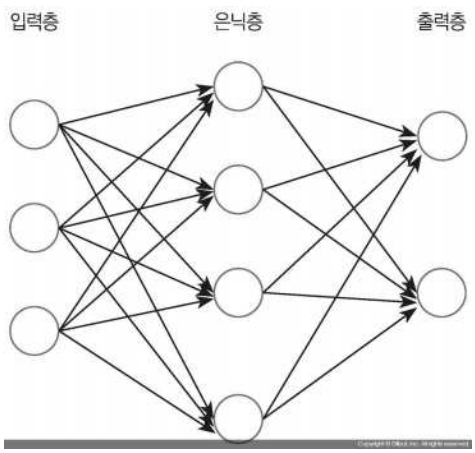
정리하기

흉부 영상을 이용해 코로나-19 감염 여부를 분류하는 인공지능을 만들기 위하여, 정상과 코로나 환자의 흉부 영상을 훈련 데이터와 테스트 데이터로 분리하고 정상과 코로나 환자를 분류하는 기계학습 모델을 만들었다. 모델의 학습 결과 로지스틱 회귀 모델의 성능이 가장 우수하였다. 하지만 테스트 데이터의 경우 정확도가 50% 정도로 측정되었으나 더 많은 훈련 데이터를 사용하여 학습시킨다면 더 나은 결과를 얻을 수 있을 것이다.

AI 더 알아보기

◆ Neural Network(신경망)

신경망은 인간의 뇌를 본 따서 만든 모델이다. 인간의 뇌는 뉴런으로 구성되어 있는데 이를 수학적으로 모델링한 것이 바로 퍼셉트론이다. 우측의 그림에서 원은 퍼셉트론을 나타내며 노드(Node)라고도 한다. 이러한 노드들이 다른 노드들과 서로 연결된 모습을 층(또는 Layer)라고 한다. 입력층은 데이터를 입력받는 층이며, 인공지능이 예측한 값은 출력층으로 나온다. 그리고 입력층과 출력층 사이에 있는 층이 은닉층으로 입력층에서 들어온 데이터가 여러 신호로 바뀌어 출력층까지 전달되는 것이다. 이러한 은닉층이 깊은 층으로 구성된 인공 신경망을 심층 신경망이라고 부르고, 이 심층 신경망이 학습하는 과정을 딥러닝(Deep Learning)이라고 한다.



[참고 문헌]

1. 손원성 외 3인(2021). 오렌지3로 알아가는 머신러닝 데이터 분석. 홍릉.
2. 서울과학종합대학원 디지털혁신처(2021). 3시간 만에 배우는 인공지능 데이터분석. 오렌지. 서울경제경영.
3. 서민구(2014). R을 이용한 데이터 처리&분석 실무. 길벗.
4. 대한감염학회. https://www.ksid.or.kr/rang_board/list.html?code=ncov_faq
5. cochrane.org. https://www.cochrane.org/ko/CD013639/INFECTN_covid-19-jindaneulwihan-hyungbu-yeongsangeun-eolmana-jeonghwaghabnigga
6. 오렌지. https://orangedatamining.com/widget-catalog/
7. 데이터 수집(kaggle). https://www.kaggle.com/nabeelsajid917/covid-19-x-ray-10000-images
8. 이미지 학습 방법 설명. https://orangedatamining.com/widget-catalog/image-analytics/imageembedding/
9. 코로나-19 이미지. https://www.joongang.co.kr/article/25010253#home

MEMO

06. 태아의 건강 상태를 미리 알 수 있을까?

상모중학교 교사 황 상 연

학습 진행 과정

1단계	데이터 준비	- 사용 데이터: fetal_health - 수집: 캐글 - 데이터 편집: 데이터 속성명 추가
2단계	데이터 불러오기	- 학습 데이터 불러오기 - 데이터의 속성별 Role(역할) 설정하기
3단계	데이터 시각화	- 시각화 도구을 이용한 데이터 탐색 - 사용한 시각화 도구: Distribution, Scattor Plot
4단계	속성 추출	- 데이터 시각화 결과 또는 Rank로 주요 속성 추출하기
5단계	모델 학습	- 분류를 이용한 모델 학습 - 사용된 알고리즘: SVM, Neural Network, Random Forest, k-NN, Logistic Regression
6단계	성능 평가	- test and score를 이용한 성능 평가
7단계	예측	- Prediction을 이용한 테스트 데이터로 예측하기

학습 내용 요약

데이터 종류	문제 유형	사용된 학습 알고리즘	성능 평가 도구
정형 데이터(수치형)	분류	SVM, Neural Network, Random Forest, kNN, Logistic Regression	test and score

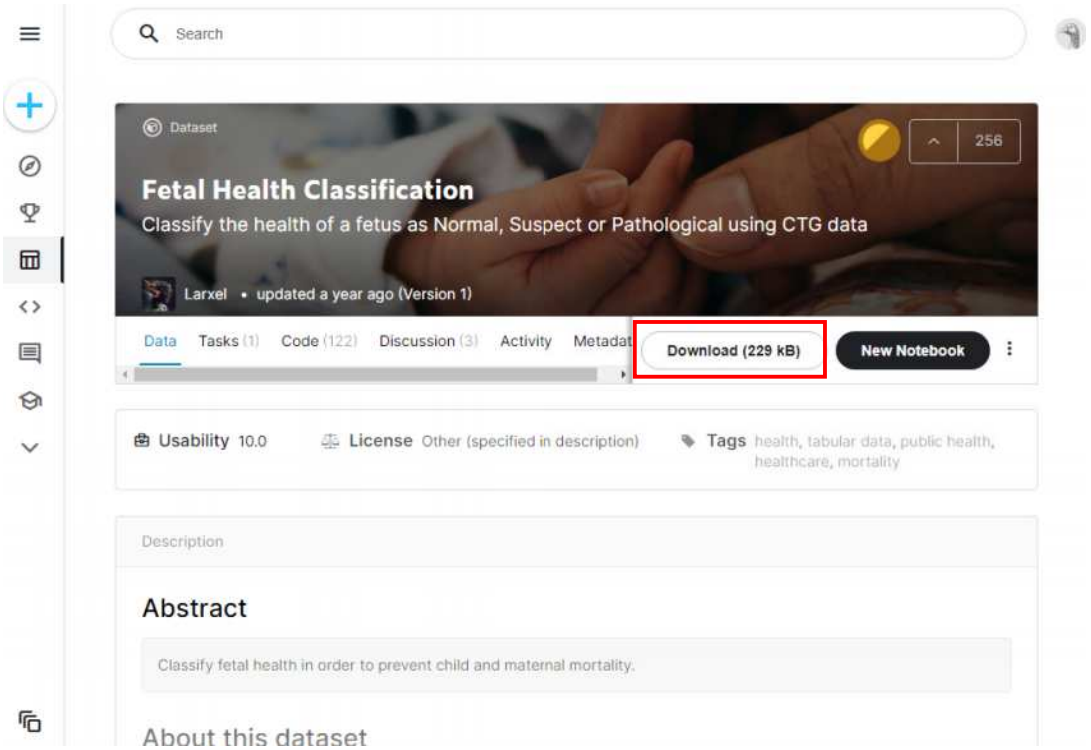
문제 상황

우리나라는 저출산으로 인해 많은 사회적 어려움이 예상되는 상태이다. 저출산의 문제를 해결하는 건 우리 사회의 중요한 숙제 중 하나이다. 이러한 문제를 해결하기에 앞서 저출산이 지속하고 있는 현재 한 명의 소중한 생명 또한 우리에겐 값진 선물과도 다름없다. 이러한 선물을 지키기 위해서는 태아의 건강을 파악하는 것이 중요하다. 태아의 다양한 데이터를 활용해 건강 상태를 예측해보자.

01 데이터 준비하기

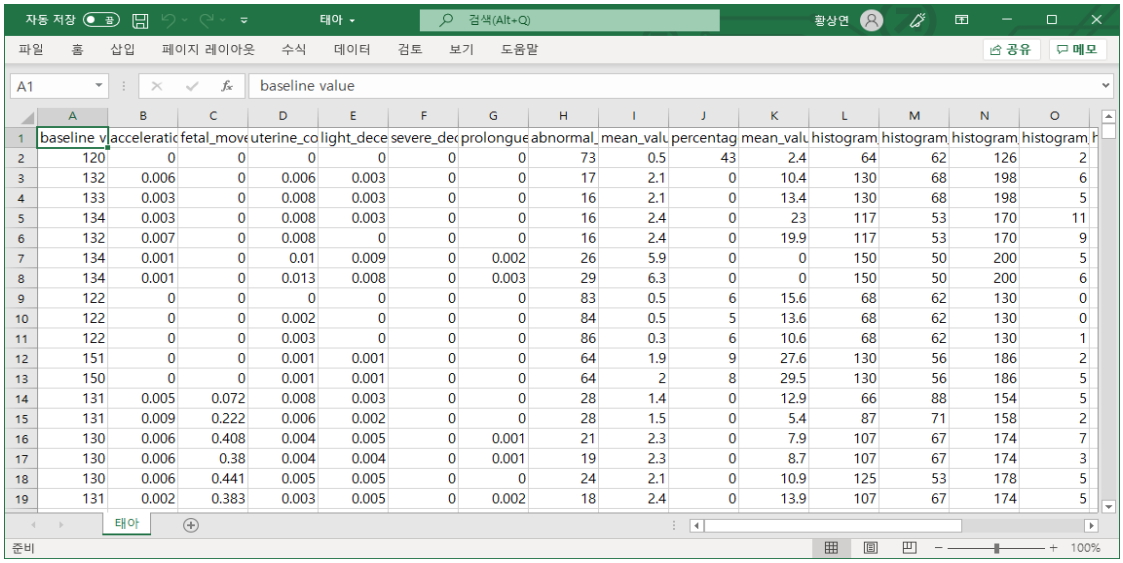
1 Fetal-Health 데이터 세트

Kaggle은 데이터 과학 목표를 달성할 수 있도록 제공하는 세계 최대 규모의 데이터 과학 커뮤니티이다. 기계학습을 위한 다양한 데이터 세트를 다운로드 할 수 있다. 이곳에서 태아 건강에 관한 데이터를 다운받아 태아의 건강 상태를 예측해보자.



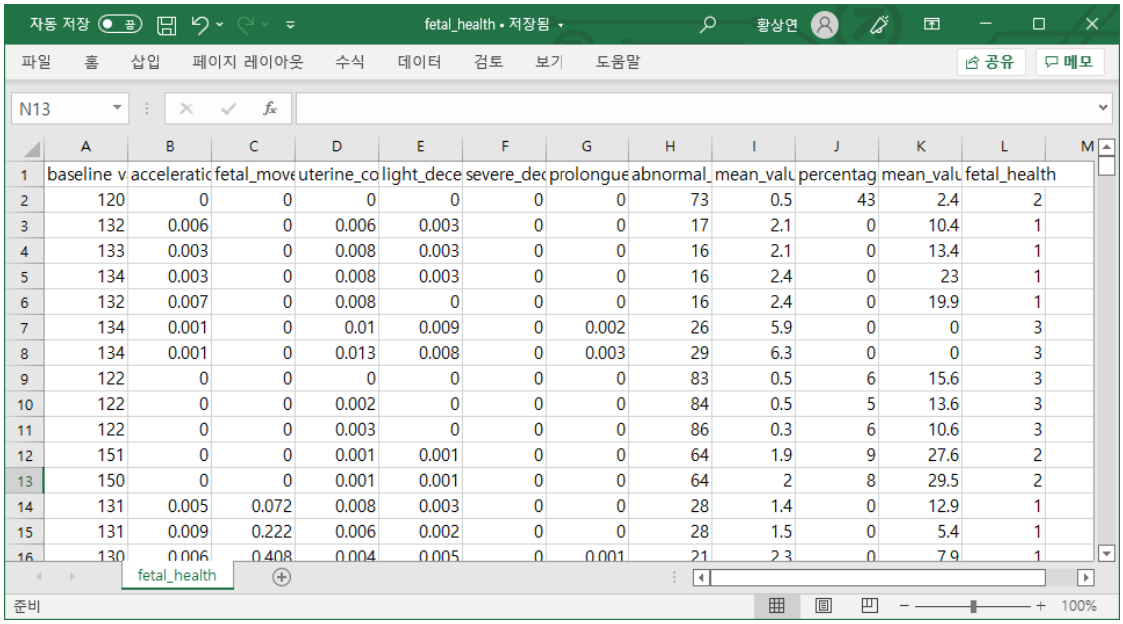
[그림 6-1] Kaggle에서 feal health 데이터 수집

[그림 6-1]과 같은 화면에서 Download를 클릭하여 데이터 세트 파일을 다운로드 받는다. 이 파일의 데이터 속성은 22가지를 가지고 있다. 속성 중 histogram이라는 이름이 들어간 속성은 다른 속성의 통계치이므로 삭제하고 사용한다.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	baseline v	acceleratic	fetal_move	uterine_co	light_dece	severe_dec	prolongue	abnormal_	mean_valu	percentag	mean_valu	histogram	histogram	histogram	histogram
2	120	0	0	0	0	0	0	73	0.5	43	2.4	64	62	126	2
3	132	0.006	0	0.006	0.003	0	0	17	2.1	0	10.4	130	68	198	6
4	133	0.003	0	0.008	0.003	0	0	16	2.1	0	13.4	130	68	198	5
5	134	0.003	0	0.008	0.003	0	0	16	2.4	0	23	117	53	170	11
6	132	0.007	0	0.008	0	0	0	16	2.4	0	19.9	117	53	170	9
7	134	0.001	0	0.01	0.009	0	0.002	26	5.9	0	0	150	50	200	5
8	134	0.001	0	0.013	0.008	0	0.003	29	6.3	0	0	150	50	200	6
9	122	0	0	0	0	0	0	83	0.5	6	15.6	68	62	130	0
10	122	0	0	0.002	0	0	0	84	0.5	5	13.6	68	62	130	0
11	122	0	0	0.003	0	0	0	86	0.3	6	10.6	68	62	130	1
12	151	0	0	0.001	0.001	0	0	64	1.9	9	27.6	130	56	186	2
13	150	0	0	0.001	0.001	0	0	64	2	8	29.5	130	56	186	5
14	131	0.005	0.072	0.008	0.003	0	0	28	1.4	0	12.9	66	88	154	5
15	131	0.009	0.222	0.006	0.002	0	0	28	1.5	0	5.4	87	71	158	2
16	130	0.006	0.408	0.004	0.005	0	0.001	21	2.3	0	7.9	107	67	174	7
17	130	0.006	0.38	0.004	0.004	0	0.001	19	2.3	0	8.7	107	67	174	3
18	130	0.006	0.441	0.005	0.005	0	0	24	2.1	0	10.9	125	53	178	5
19	131	0.002	0.383	0.003	0.005	0	0.002	18	2.4	0	13.9	107	67	174	5

[그림 6-2] feal health 데이터 세트 수정 전 파일



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	baseline v	acceleratic	fetal_move	uterine_co	light_dece	severe_dec	prolongue	abnormal_	mean_valu	percentag	mean_valu	fetal_health	
2	120	0	0	0	0	0	0	73	0.5	43	2.4	2	
3	132	0.006	0	0.006	0.003	0	0	17	2.1	0	10.4	1	
4	133	0.003	0	0.008	0.003	0	0	16	2.1	0	13.4	1	
5	134	0.003	0	0.008	0.003	0	0	16	2.4	0	23	1	
6	132	0.007	0	0.008	0	0	0	16	2.4	0	19.9	1	
7	134	0.001	0	0.01	0.009	0	0.002	26	5.9	0	0	3	
8	134	0.001	0	0.013	0.008	0	0.003	29	6.3	0	0	3	
9	122	0	0	0	0	0	0	83	0.5	6	15.6	3	
10	122	0	0	0.002	0	0	0	84	0.5	5	13.6	3	
11	122	0	0	0.003	0	0	0	86	0.3	6	10.6	3	
12	151	0	0	0.001	0.001	0	0	64	1.9	9	27.6	2	
13	150	0	0	0.001	0.001	0	0	64	2	8	29.5	2	
14	131	0.005	0.072	0.008	0.003	0	0	28	1.4	0	12.9	1	
15	131	0.009	0.222	0.006	0.002	0	0	28	1.5	0	5.4	1	
16	130	0.006	0.408	0.004	0.005	0	0.001	21	2.3	0	7.9	1	

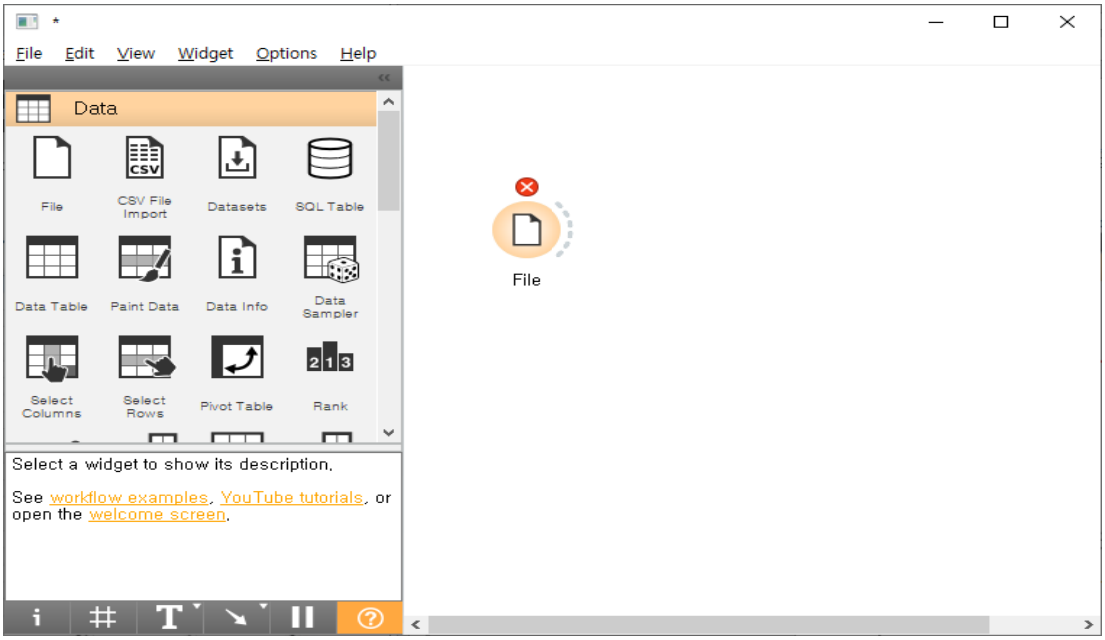
[그림 6-3] feal health 데이터 세트 수정 후 파일

[표 6-1] 태아 데이터의 속성

	속성	이름	비고
1	baseline value	태아 심박 수	
2	accelerations	초당 가속 횟수	
3	fetal_movement	초당 태아 움직임의 수	
4	uterine_contractions	초당 자궁 수축 횟수	
5	light_decelerations	빛 반복	
6	severe_decelerations	심각한 느린 맥박	
7	prolongued_decelerations	지속되는 느린 맥박	
8	abnormal_short_term_variability	비정상적 단기 변동성이 있는 시간의 백분율	
9	mean_value_of_short_term_variability	단기 변동성의 평균값	
10	percentage_of_time_with_abnormal_long_term_variability	비정상적인 자이 변동성이 있는 시간의 백분율	
11	mean_value_of_long_term_variability	장기 변동성의 평균값	
12	fetal_health	태아 건강 상태	1-정상, 2-주의, 3-나쁨

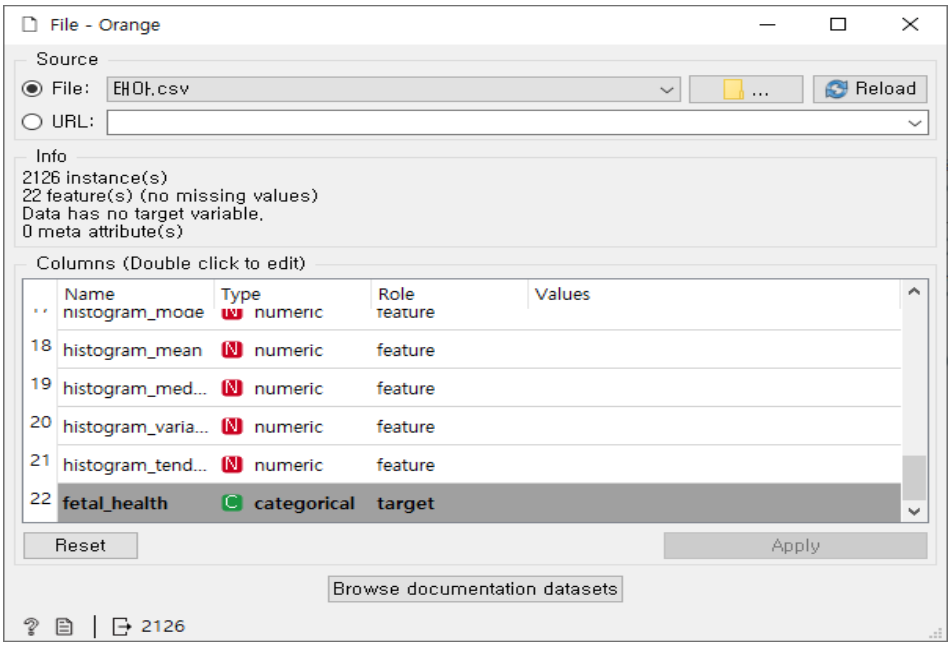
2 데이터 불러오기

① 오렌지3을 실행하여 File을 드래그하여 오른쪽 캔버스에 놓는다.



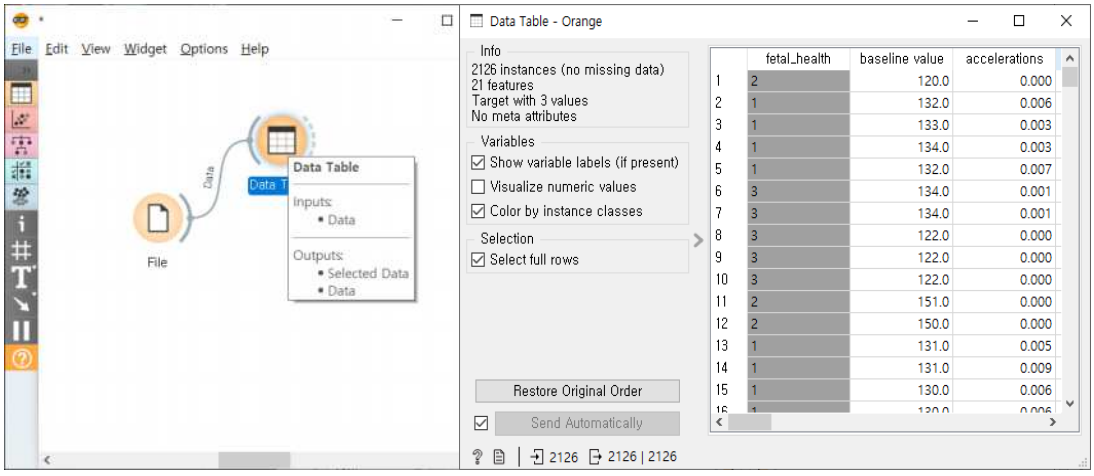
[그림 6-4] 캔버스에 파일 추가하기

② File 위젯을 더블클릭하여 다운로드 받은 CSV 파일을 불러오기 한다. fetal_health (태아 건강 상태) 속성을 target으로 설정한다.



[그림 6-5] 속성 역할 바꾸기

③ 파일에서 Data Table을 만들면 CSV 파일의 데이터를 확인할 수 있다.



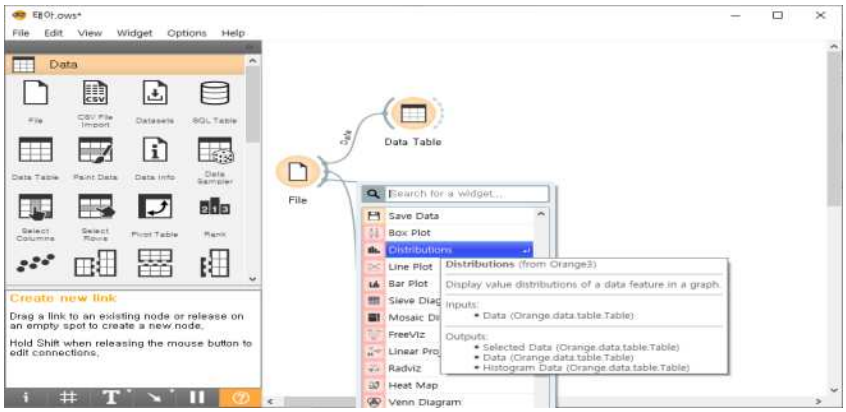
[그림 6-6] 데이터 세트 확인하기

02 데이터 탐색하자

1 데이터 시각화하기

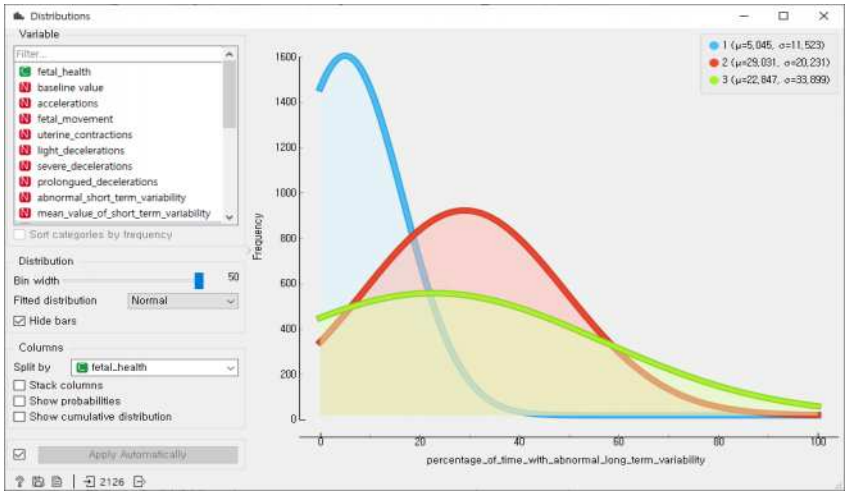
태아의 산모와 함께 있어서 의사가 직접적으로 진찰을 하기에 어려움이 있다. 그래서 여러 가지 데이터를 종합하여 태아의 상태가 건강한지 건강하지 않은지 확인해 볼 수 있다. 우리는 각 속성의 관계를 데이터 시각화를 통해 알아보도록 하자.

① File 위젯에서 오른쪽 괄호를 드래그하고 팝업에서 Distributions 위젯을 만들어 주도록 한다.



[그림 6-7] 시각화 추가하기

Distributions 위젯은 이산형, 연속형 속성의 값 분포를 표시한다. 이산형 속성의 경우 위젯에 의해 표시되는 그래프는 각 속성값이 데이터에 나타나는 횟수를 보여준다. 데이터에 클래스 변수가 포함된 경우 각 속성값에 대한 클래스 분포도 함께 표시된다.



[그림 6-8] Distributions 확인하기

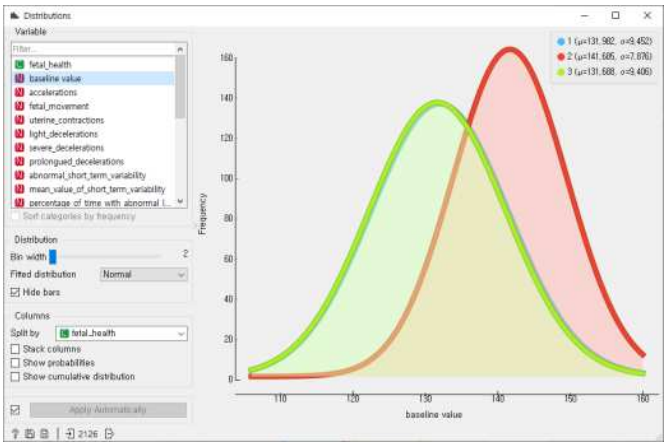
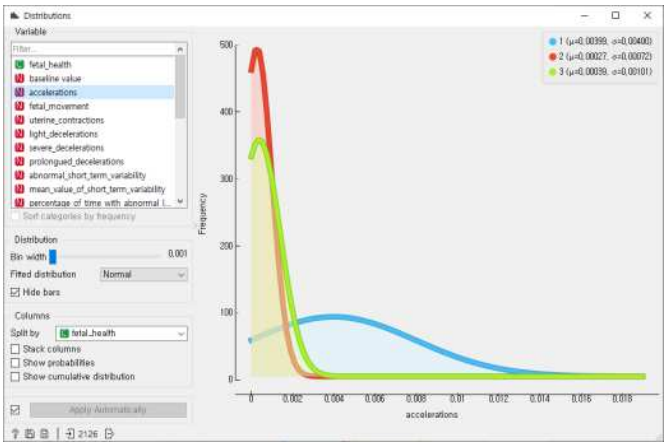
Distribution 위젯 창 세부 설명

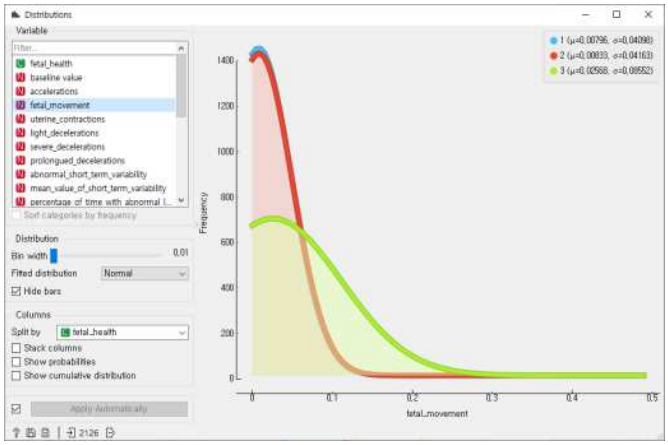
Variable	분포 표시에 대한 변수 리스트를 보여준다.
Group by	상대 빈도도 표시는 데이터 세트의 백분율로 데이터를 확장한다.
Probabilities	확률을 표시할 수 있다.

Columns를 fetal_health로 설정을 하면 각 데이터가 나타내는 값의 태아 상태를 색상 구분을 통해 확인할 수 있다.

1, 2, 3 값에 따라서 색을 다르게 표현 할 수 있다. 3가지의 색이 뚜렷하게 구분이 되는 속성이 있다면 그 속성은 건강 상태를 확인하기 위한 중요한 변인이 될 것이다.

① Distributions로 데이터 탐색하기

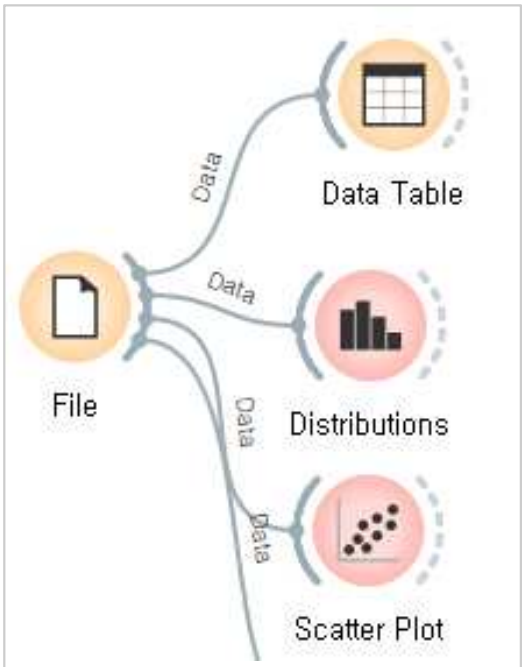
속성별 분포도	데이터 해석
	속성: baseline value(태아 심박수) 1번정상과 3번나쁨의 평균값이 거의 차이가 없고 겹치는 값이 많아 단일 속성만으로 분류하기에는 어려움이 있음
	속성: accelerations(초당 가속 횟수) 2번주의와 3번나쁨의 평균값이 거의 차이가 없고 겹치는 값이 많아 단일 속성만으로 분류하기에는 어려움이 있음

속성별 분포도	데이터 해석
	속성: fetal_movement (초당 태아 움직임의 수) 1번정상과 2번주의의 평균값이 거의 차이가 없고 겹치는 값이 많아 단일 속성만으로 분류하기에는 어려움이 있음

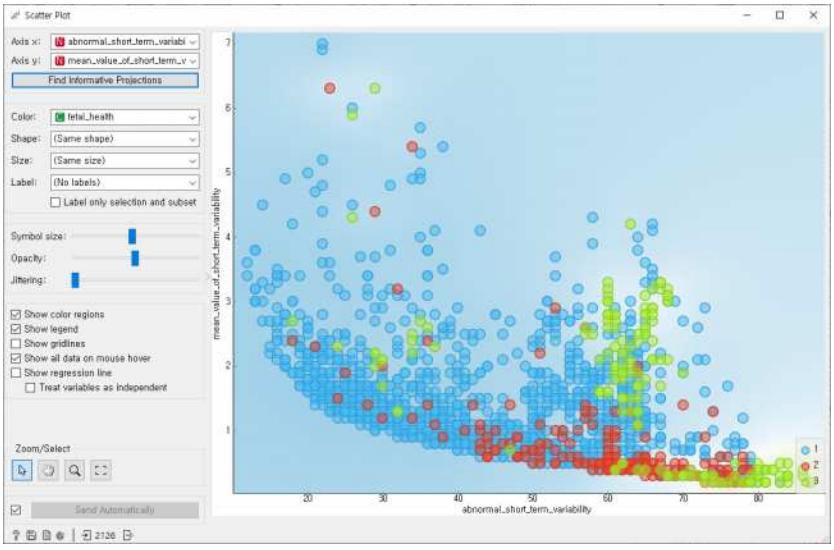
② 산점도로 나타내기

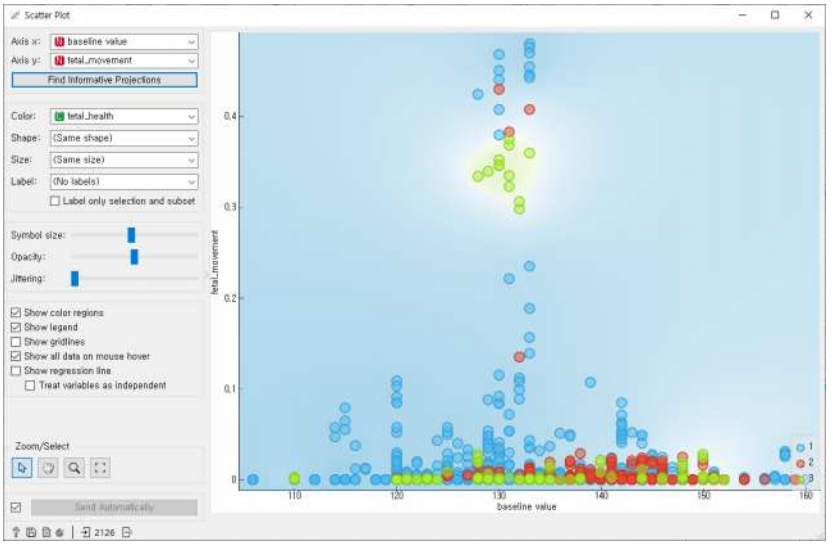
산점도는 두 변수의 관계를 통해 x축과 y축으로 설정하고 두 값이 만나는 곳에 점으로 나타난 그래프이다. 산점도를 통해 데이터의 관계를 알아볼 수 있으며, 다양한 속성 중 데이터를 잘 분류할 수 있는 속성을 찾아 x축과 y축으로 선정하는 것이 좋다.

② File 위젯에서 오른쪽 괄호를 드래그하고 팝업에서 Scatter Plot 위젯을 만든다.



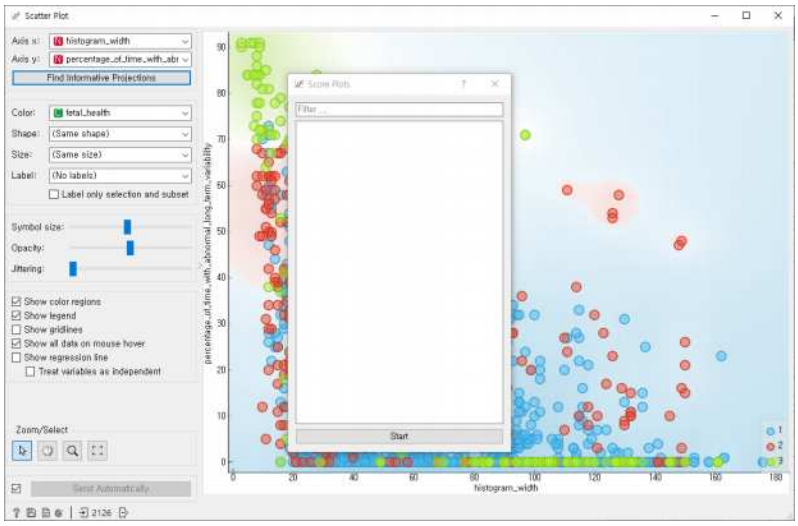
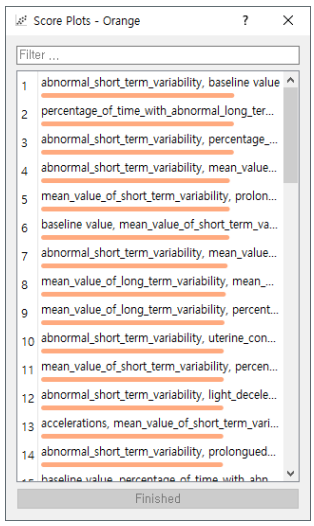
[그림 6-9] Scatter Plot 추가하기

2가지 속성을 이용한 산점도	데이터 해석
	품종을 잘 분류하는 속성으로 생각되는 abnormal과 mean_value를 이용하여 산점도로 표현

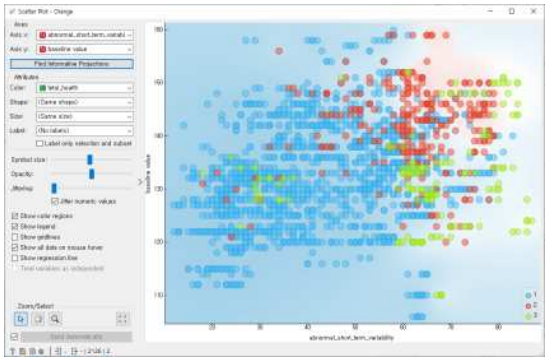
2가지 속성을 이용한 산점도	데이터 해석
	품종을 잘 분류하지 못하는 속성으로 생각되는 baseline value와 fetal_movement를 이용하여 산점도로 표현

태아 건강 상태 데이터처럼 데이터 세트에 많은 속성이 있는 경우 모든 쌍을 수동으로 스캔하여 의미있는 산점도를 찾기는 어렵다.. 오렌지3은 위젯의 정보 투영 찾기 옵션으로 지능형 데이터 시각화를 지원한다.

[Find Informative Projections]을 클릭하면 클래스 분류가 잘되어있는 점수로 결과를 보여준다.

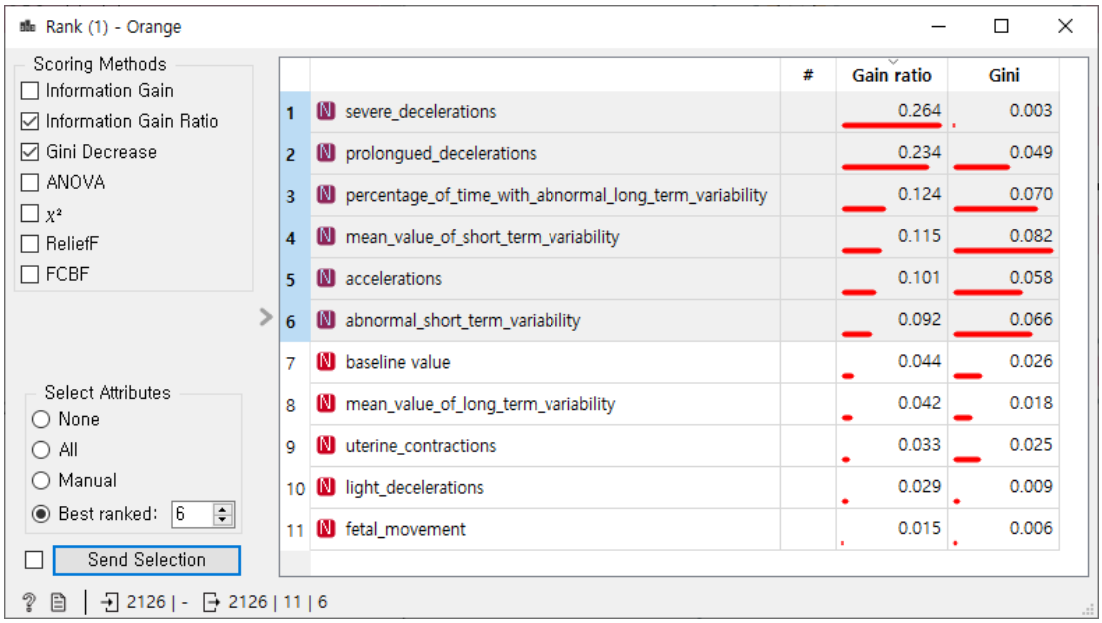
속성 추천	추천 결과
	

오렌지에서 추천한 속성을 이용해 분류한 결과를 산점도로 표시하면 다음과 같다.

1순위	3순위
	
abnormal_short_term_variability와 baseline value	abnormal_short_term_variability와 percentage_of_time_with_abnormal_long_term_variability

2 Rank 위젯으로 속성 추출하기

Rank 위젯을 이용하면 데이터 속성의 관련성을 바탕으로 점수를 산출하여 순위를 지정하고 속성을 필터링해준다. 이중 관련 있는 속성 상위 6개만 사용하도록 설정해 준다.



[그림 6-10] Rank 확인하기

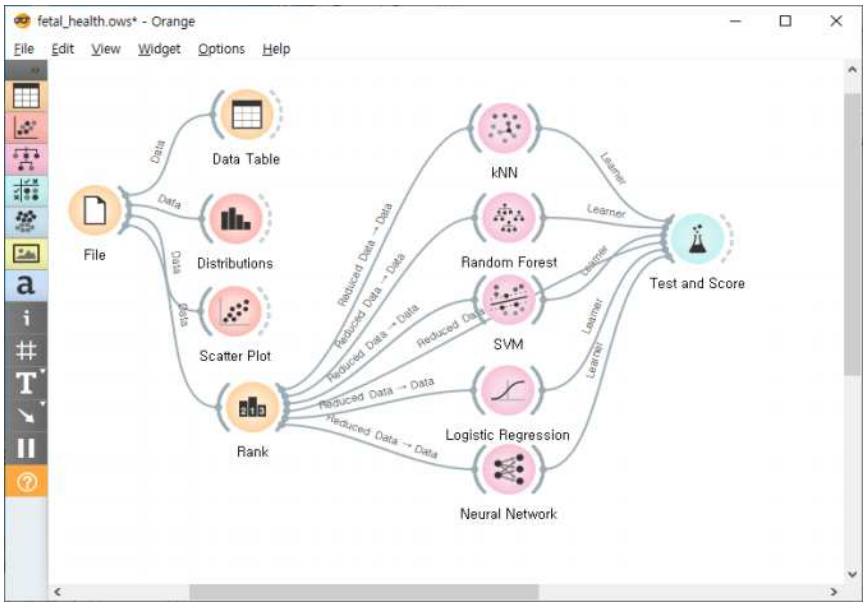
03 모델 학습과 성능 평가하자

1 모델학습

추출한 데이터 속성을 바탕으로, 기계학습 알고리즘과 데이터를 연결하여 모델 학습한다. 오렌지에서는 다양한 기계학습 알고리즘을 한꺼번에 연결하여 모델을 만들 수 있다. 여기서는 분류에 자주 사용하는 k-NN, random Foreset, SVM, Logistic Regression, Neural Network를 이용하여 모델을 구성하였다. model에서 해당하는 학습 알고리즘을 선택하여 File과 연결한다.

① Rank를 통해 추출한 6개의 속성을 사용하여 모델과 연결할 수 있다. 모델 메뉴에서 몇 가지의 위젯을 캔버스에 가져와 Rank와 연결하도록 하자.

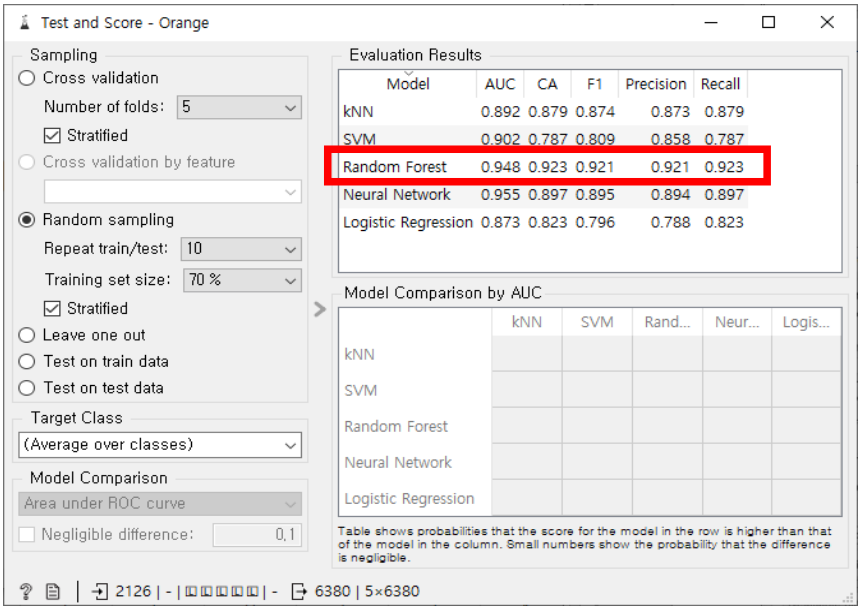
② 여러 개의 모델의 성능을 확인하기 위해서 Test and Score를 사용해 보도록 하자. 각 모델과 연결하고 Preprocess(전처리)와 연결하여 모델과 데이터를 연결해 준다. 그러면 학습이 시작되고 100%가 되면 완료된다.



[그림 6-11] 모델 학습하기

2 성능평가

샘플링 방식 중 Random Sampling은 전체 데이터를 섞어서 무작위로 훈련 데이터와 테스트 데이터를 분리한다. 또한, 훈련과 테스트의 반복 횟수를 설정할 수 있다. 테스트 데이터를 이용한 계산을 쉽게 하려면 반복(repeat train/test)를 10회로 설정하였다. 성능 평가 지표 중 CA는 분류 정확도를 나타내므로 이를 기준으로 판단했다.



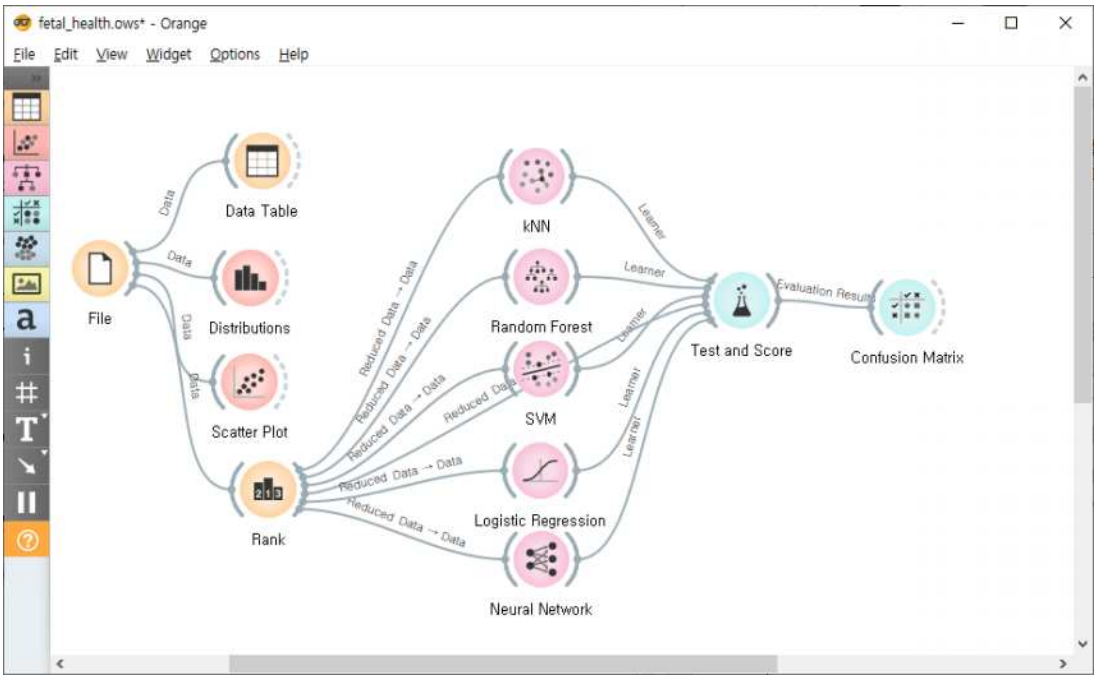
[그림 6-12] 성능 확인하기

- Evaluation Results(성능평가 결과)를 보면 좌측에 각 모델이 나타나있고 각 모델별 성능지표(AUC, CA, F1, Precision, Recall 등)을 확인할 수 있다.
- 모델의 성능을 종합적으로 평가할 수 있는 CA를 보면 Random Forest가 0.923으로 가장 높은 것을 알 수 있다.
- CA는 모델이 입력된 데이터에 대해 얼마나 정확하게 분류하는지를 나태나는 값이다. 1의 값에 가까울수록 모델의 성능을 좋다고 판단한다.

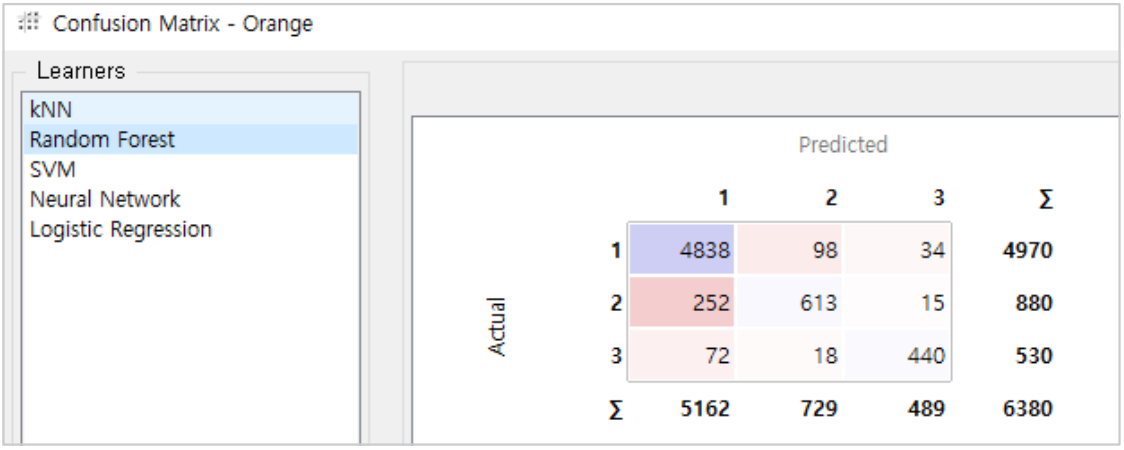
$$\text{CA(정확도)} = \frac{TP + TN}{TP + FP + TN + FN}$$

① Confusion Matrix 이용하기

- Confusion Matrix를 이용하면 각 모델별 성능을 훨씬 구체적으로 확인할 수 있다.
- 좌측 항목 Evaluate에서 Confusion Matrix를 클릭하여 오른쪽 화면에 끌어 놓은 후 Test and Score와 연결한다.
- Confusion Matrix를 더블클릭하면 각 모델별 정확도를 넘어 구체적으로 어떤 모델이 어떤 경우에는 맞혔고 어떤 경우에는 틀렸는지 확인할 수 있다.



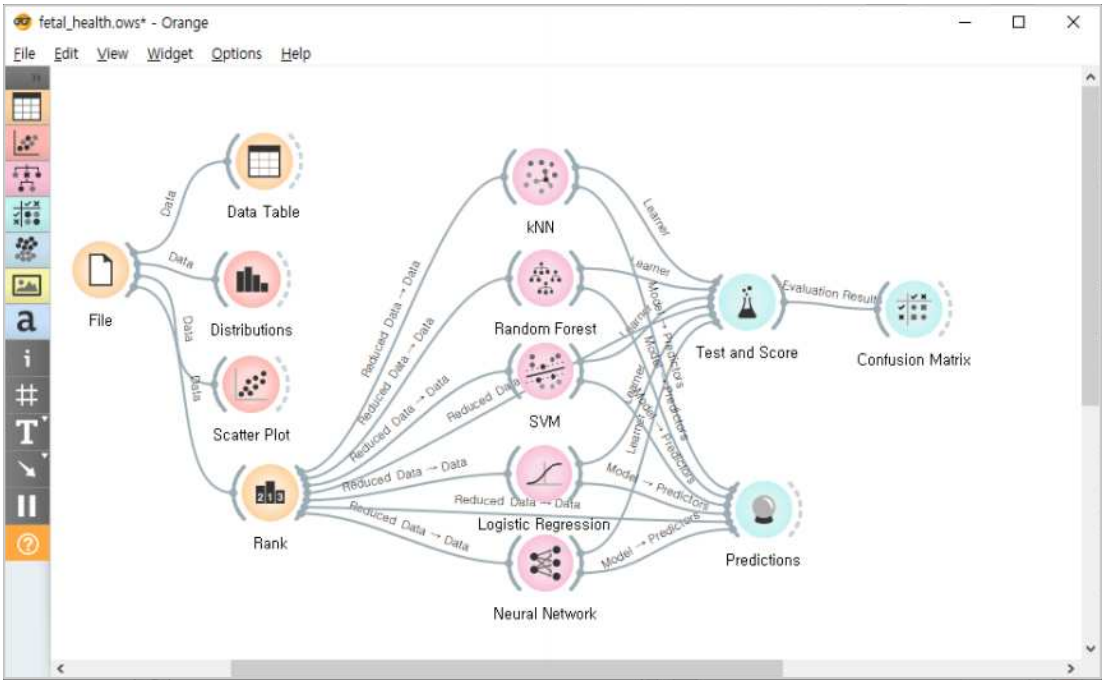
[그림 6-13] Confusion Matrix 위젯 추가하기



[그림 6-14] Confusion Matrix 결과 확인하기

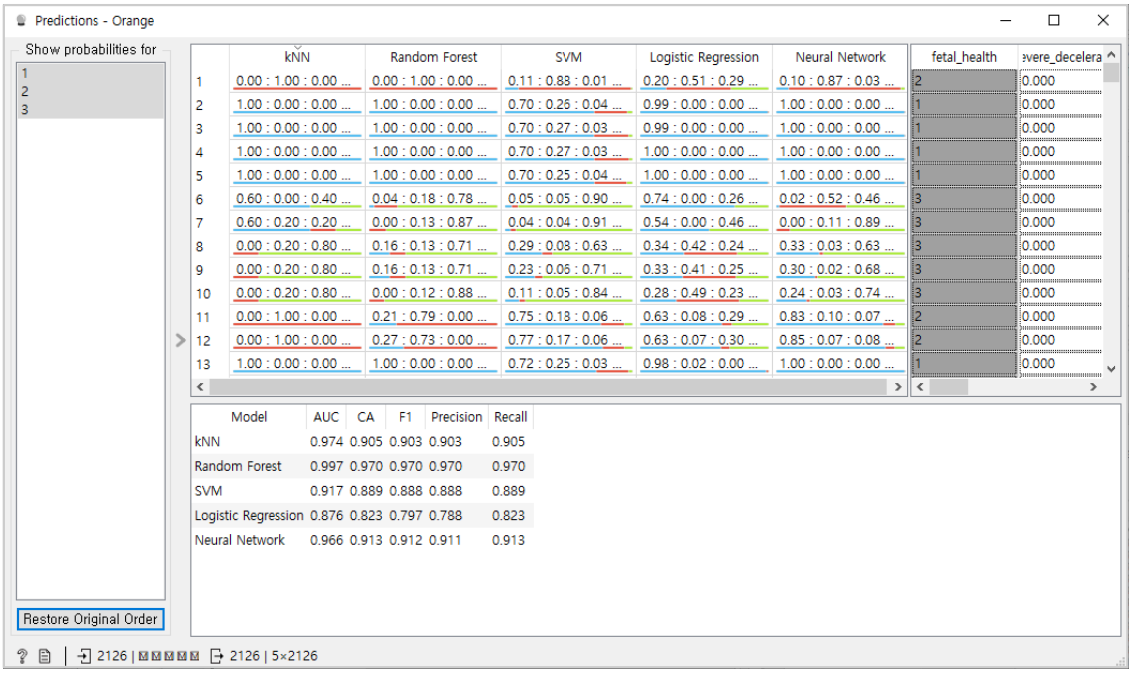
② 건강상태 예측하기

- 이제 학습시킨 모델들을 이용하여 태아의 건강상태를 예측해보자.
- Evaluate에서 Predictions를 가져오고 기계학습 모델과 Rank를 연결해준다.



[그림 6-15] Predictions 추가하기

- 앞에서 학습시킨 모델을 Predictions에 이어주면 학습한 모델에 기반하여 값을 다음과 같이 예측할 수 있다.



[그림 6-16] Predictions 결과 확인하기

- Predictions 창을 더블클릭해서 열어보면 위와 같이 태아의 건강상태를 어떻게 예측했는지 확인해볼 수 있다.
- 앞서 성능평가 결과를 확인했을 때 사용한 모델 중 Random Forest가 가장 우수했던 것과 같이 실제 태아의 데이터와 학습한 모델로 예측한 값을 비교해 보았을 때 성능이 0.970으로 우수한 것을 확인할 수 있다.

06. 태아의 건강 상태를 미리 알 수 있을까?

정리하기

태아의 수치 정보를 이용해 태아의 건강상태를 파악해보는 기계학습 모델을 만들었다. 모델의 학습 결과와 테스트 결과 Random Forest 모델의 성능이 가장 우수한 것으로 나타났다. 태아의 수치 정보를 이용해 건강 상태를 파악할 때는 다양한 모델 중 Random Forest를 사용하면 더 정확한 예측 결과를 얻을 수 있을 것이다.

[참고 문헌]

1. 서울과학종합대학원 디지털혁신처(2021). 3시간 만에 배우는 인공지능 데이터분석. 오렌지. 서울경제경영.
2. 손원성외 3인(2021). 오렌지3로 알아가는 머신러닝 데이터 분석. 홍릉.
3. 이고잉외 2인(2021). 생활코딩 머신러닝. 위키북스.
4. Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/andrewmvd/fetal-health-classification
5. 오렌지. https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/data/rank.html
6. 태아 사진. https://www.ac-illust.com/ko/clip-art/383590/%EC%9E%84%EC%82%B0%EB%B6%80%EC%99%80-%ED%83%9C%EB%8F%99

07. 교통사고 정보를 통해 운전자의 피해 정보를 구분할 수 있을까?

상모중학교 교사 황 상 연

학습 진행 과정

1단계	데이터 준비	- 사용 데이터: 제주도 개별 교통사고 - 수집: 공공데이터포털
2단계	데이터 불러오기	- 학습 데이터 불러오기 - 데이터의 속성별 Role(역할) 설정하기
3단계	모델 학습	- 분류를 이용한 모델 학습 - 사용된 알고리즘: Naive Bayes, Neural Network, Random Forest, k-NN
4단계	성능 평가	- test and score를 이용한 성능 평가 - 혼동 행렬을 이용한 성능 평가

학습 내용 요약

데이터 종류	문제 유형	사용된 학습 알고리즘	성능 평가 도구
정형 데이터(문자형)	분류	k-NN Random Forest Neural Network Naive Bayes	혼동 행렬

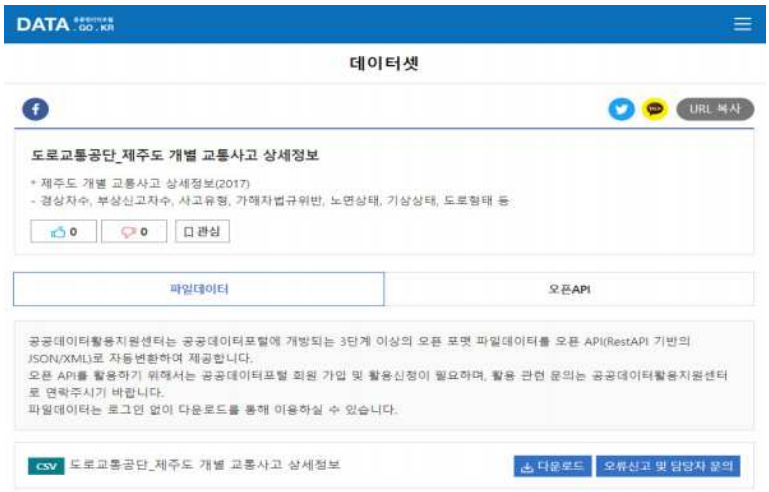
문제 상황

제주도에서는 매년 많은 관광객들이 찾고있고, 늘어난 렌트카, 교통들로 인해 제주도의 도로교통 안전 문제가 여러 차원에서 제기되고 개선책이 논의되고 있다. 하루 평균 3만명 이상의 관광객이 모두 렌터카를 사용하는 것은 아니지만 렌터카를 직접 운전하는 관광객과 현지 운전자들의 다른 운전 습관이 충돌하기 쉽다. 게다가 많은 도로가 높은 한라산을 중심으로 이루어진 산비탈이어서 교통불안 지점이 많다. 사고에는 많은 유형들이 있고, 사고의 원인에도 다양한 유형이 있을 것이다. 충돌종류와 가해자의 위법사유, 도로의 노면상태, 기상상태, 도로형태의 분류 등의 정보를 통해서 피해자의 신체상해 정도를 파악할 수는 없을까?

01 데이터 준비하기

1 교통사고 데이터 세트

공공데이터포털(https://www.data.go.kr/index.do)은 공공기관이 생성 또는 취득하여 관리하고 있는 공공데이터를 한 곳에서 제공하는 통합 창구이다. 포털에서는 국민이 쉽고편리하게 공공데이터를 이용할 수 있도록 파일데이터, 오픈API, 시각화 등 다양한 방식으로 제공하고 있으며, 누구라도 쉽고 편리한 검색을통해 원하는 공공데이터를 빠르고 정확하게 찾을 수 있다. 여기에서 기계학습을 위한 다양한 데이터 세트를 다운로드할 수 있다. 이곳에서 제공하는 제주도 교통사고 데이터 세트를 이용하여 교통사고 유형별 피해자의 상해 정보를 분류할 수 있는지 알아보자.



[그림 7-1] 공공데이터포털 제주도 개별 교통사고 상세정보 화면

다운로드를 클릭한 후 2017년 제주도 교통사고 개별정보 파일을 다운로드 할 수 있다.

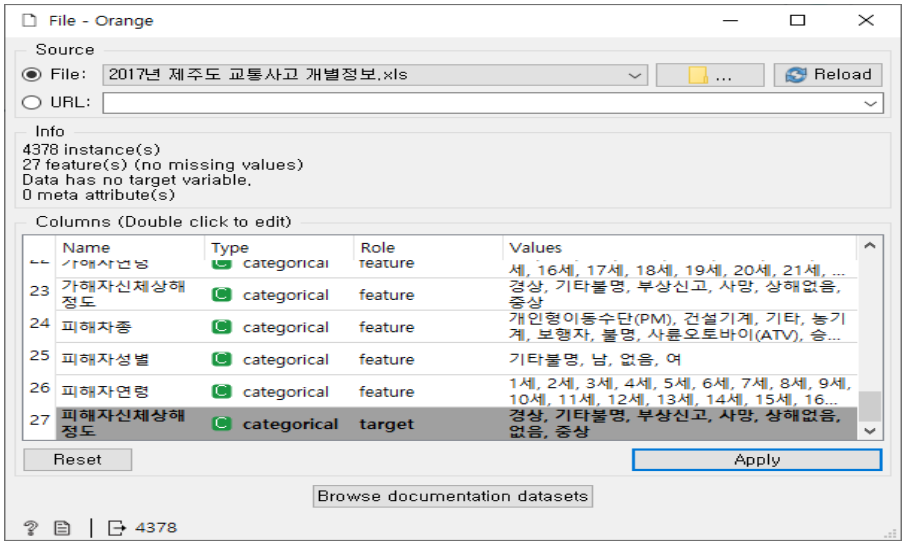
교통사고 데이터 세트는 분류 또는 클러스터링에 활용할 수 있다. 총 27개의 데이터 속성을 가지고 있다. 각 속성들은 개별적인 사건에 대한 기록들이 있다. 사고유형, 도로상태, 기상상태 등의 정보들이 있다. 마지막 열에는 피해자신체상해정보가 있다. 경상, 기타불명, 부상신고, 사망, 상해없음, 없음, 중상 이렇게 7가지의 정보로 구성되어 있다. 사고정보들로 피해자의 상해 정보를 분류할 수 있는지 확인해 보도록 하자.

2 데이터 불러오기

① Orange3을 실행하여 Data - File을 선택하여 아이콘을 캔버스에 배치한다.

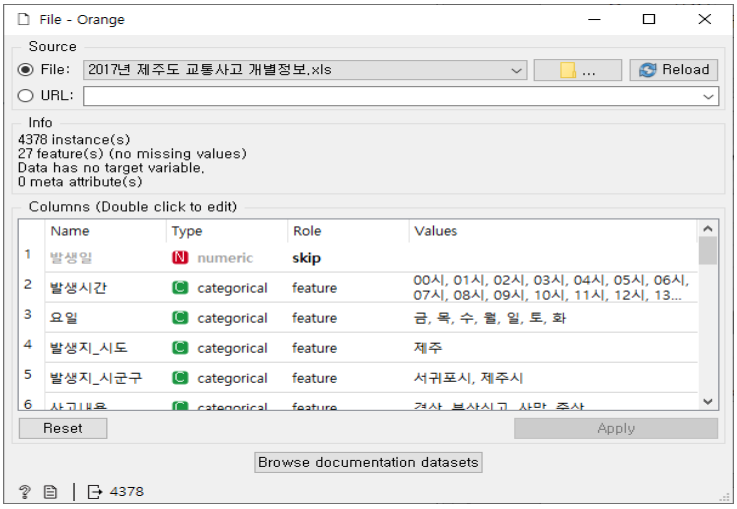
[그림 7-2] 캔버스에 데이터 세트 추가하기

② File 위젯을 더블클릭하여 다운로드 받은 교통사고 파일을 추가하고 피해자신체상해정보를 예측하기 위해 피해자신체상해정보 속성의 Role(역할)을 target으로 지정한다.



[그림 7-3] 속성 역할 설정1

③ 데이터를 분류하기 전에 교통사고와 관련 없는 속성인 발생일의 Role(역할)을 skip으로 지정한다.

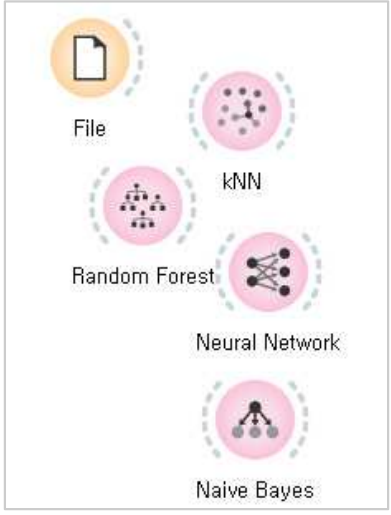


[그림 7-4] 속성 역할 설정2

02 모델학습하고 성능 평가하자

1 모델 학습

기계학습 알고리즘과 데이터를 연결하여 모델 학습한다. 오렌지3에서는 다양한 기계학습 알고리즘을 한꺼번에 연결하여 모델을 만들 수 있다. 여러 가지 모델 중 k-NN, Random Forest, Neural Network. Naive Bayes을 이용하여 모델을 구성한다.



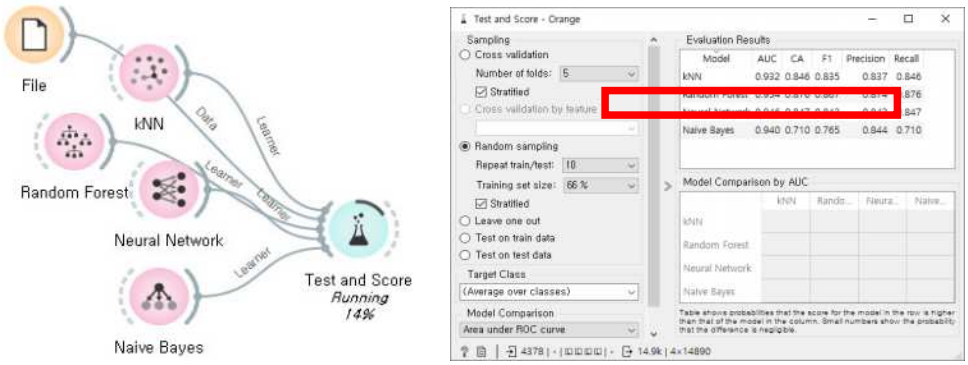
[그림 7-5] 모델 추가하기

2 성능 평가

① 모델 학습의 정확도를 확인하기 위해서 Evaluate 메뉴에서 Test and Score 위젯을 캔버스로 가져와서 가져온 모델과 Flie을 연결해 주도록 하자.

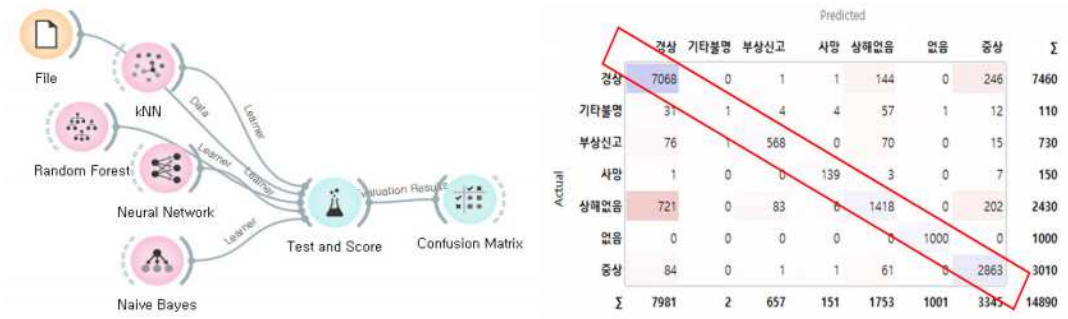
Test and Score 위젯에서 모델 학습과 테스트 데이터의 비율을 설정할 수 있다. Test and Score위젯을 더블클릭하여 Random sampling을 선택한다. Repeat train/test는 10번으로 설정하고 Training set size는 66%로 설정한다.

분류 유형에서는 CA, AUC, F1, Precision, Recall, LogLoss와 같은 성능 평가 지표가 있다. 이 중 CA(분류 정확도)는 '맞게 분류한 경우의 수'를 '전체 경우의 수'로 나눈 정확도이다. 학습 결과 Random Forest가 CA값이 가장 높게 나타났다.



[그림 7-6] 모델 학습하기

② Random Forest의 분류의 정확도를 확인하기 위해 혼동 행렬(Confusion Matrix)를 사용해 보자. Confusion Matrix 위젯을 캔버스에 놓고 Test and Score 위젯과 연결한다. 위젯을 더블클릭하여 왼쪽에서 알고리즘을 선택하고 혼동 행렬을 확인한다. 여기서 테스트 데이터의 수가 14890이다.



[그림 7-7] Confusion Matrix 확인하기

Random Forest알고리즘으로 만든 모델의 혼동 행렬을 살펴보면, 정확하게 분류한 경우가 13057개이다. 13057/14890을 계산하면 약 0.876이 나오는 것을 확인할 수 있다.

정리하기

제주도에서 발생한 교통사고 정보를 활용하여 부상자의 정도를 파악해 보는 기계학습 모델을 만들어 보았다. 모델의 학습 결과 Random Forest 모델의 성능이 가장 우수한 것으로 나타났다. 이러한 정보를 통해 교통사고의 정보를 통해 피해자의 신체피해정보를 신뢰도있게 예측하는 것을 확인할 수 있다. 이를 이용하여 운전자가 조심해야하는 교통의 정보들을 보고 사고를 예방한다면 더욱 안전한 제주도에서의 여행과 제주도민의 교통안전이 보장될 것이다.

[참고 문헌]

1. 서울과학종합대학원 디지털혁신처(2021). 3시간 만에 배우는 인공지능 데이터분석. 오렌지. 서울경제경영.
2. 손원성 외 3인(2021). 오렌지3로 알아가는 머신러닝 데이터 분석. 홍릉.
3. 이고잉 외 2인(2021). 생활코딩 머신러닝. 위키북스.
4. 조태호(2020). 모두의 딥러닝 개정 2판. 길벗.
5. 서민구(2014). R을 이용한 데이터 처리&분석 실무. 길벗.
6. 공공데이터포털. https://www.data.go.kr/index.do
7. 오렌지. https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/data/rank.html
8. 교통사고사진. https://www.photo-ac.com/ko/photo/2637973/%EA%B5%90%ED%86%B5-%EC%82%AC%EA%B3%A0-%EC%9E%90%EB%8F%99%EC%B0%A8-%EC%82%AC%EA%B3%A0-%EC%B6%94%EB%8F%8C-%EC%B6%A9%EB%8F%8C

08. 영화 평점 리뷰에서 많이 등장하는 단어는?

사동고등학교 교사 서 정 민

학습 진행 과정

1단계	데이터 준비	- 사용 데이터: 영화 평점 리뷰 - 수집: N사 (타사 리뷰 데이터 수집하여 분석해도 가능) - 데이터 편집: 웹크롤링과 코랩을 이용하여 원하는 속성만 추출해 .csv형태의 파일로 저장
▼		
2단계	데이터 불러오기	- 저장한 .csv 데이터 불러오기 - 데이터의 사용되는 특성, 사용되지않는 특성 설정하기
▼		
3단계	1차 데이터 시각화	- 시각화 도구을 이용한 데이터 탐색 - 사용한 시각화 도구: Word Cloud
▼		
4단계	데이터 전처리	- Process Text를 이용하여 검색에 불필요한 단어 제외
▼		
5단계	2차 데이터 시각화	- 시각화 도구을 이용한 데이터 탐색 - 사용한 시각화 도구: Word Cloud
▼		
6단계	감정 분석	- 감정분석 도구을 이용한 데이터 감정 분석 - 사용한 감정분석 도구: Sentiment Analysis

학습 내용 요약

데이터 종류	문제 유형	사용된 도구
비정형 데이터 (수치형, 문자형)	분류	Text - Word Cloud Text - Sentiment Analysis

문제 상황

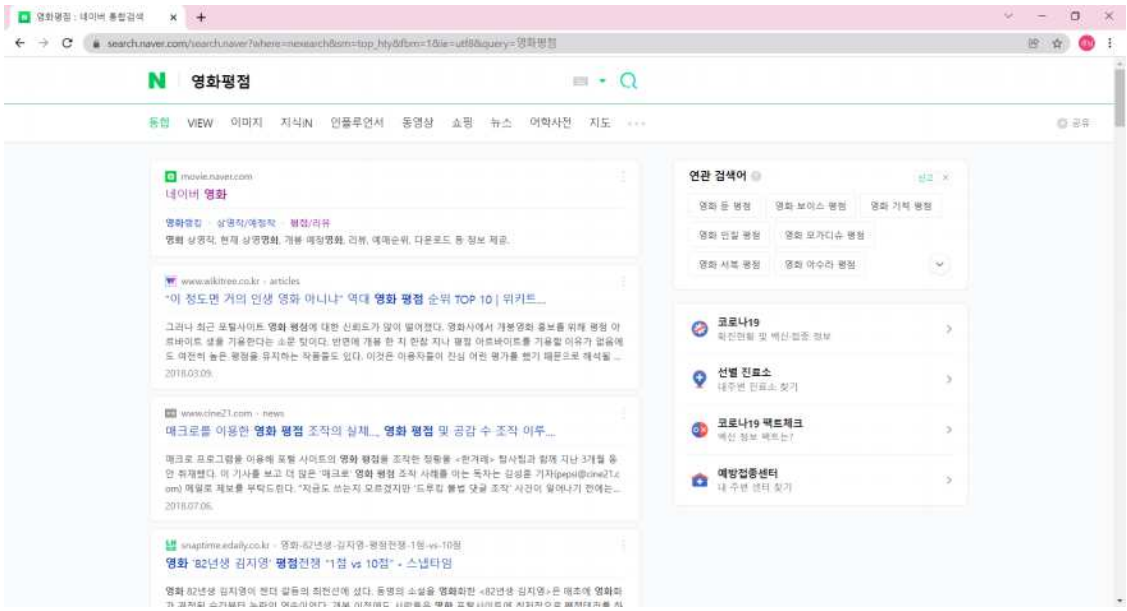
영화를 보기 전 미리 줄거리를 읽거나 평점리뷰를 주로 확인하며 영화가 재미있을지, 없을지를 판단한다.

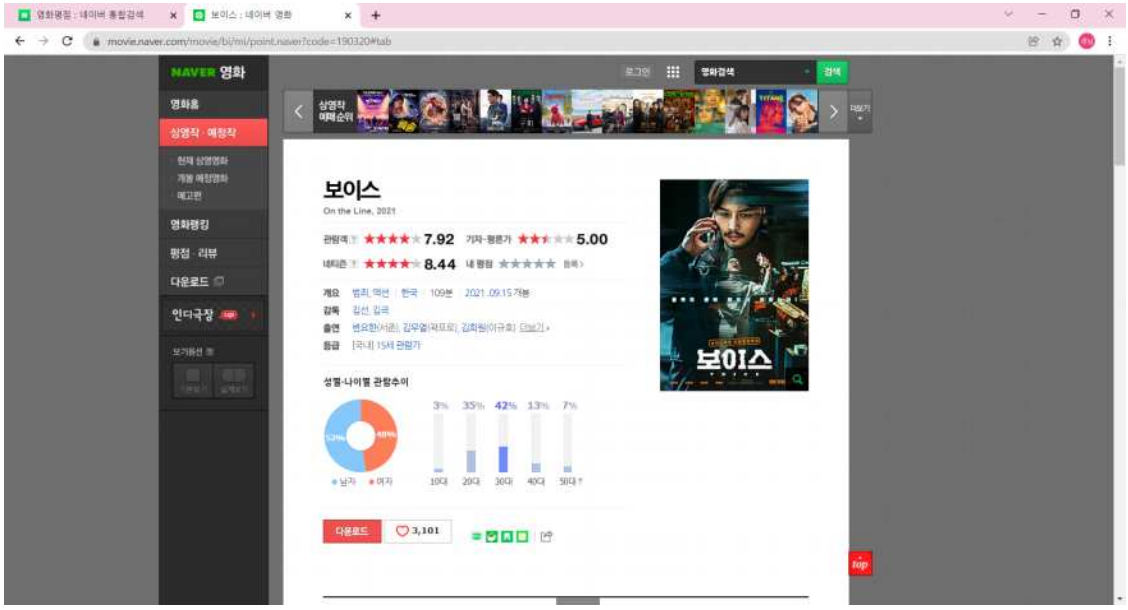
많은 평점 리뷰를 다 읽는데 많은 시간이 걸리기에 사람들이 어떤 리뷰를 작성했는지 빠르고 쉽게 확인할 수 있는 방법은 없을까?

01 데이터 준비하기

1 리뷰 데이터 세트 웹 크롤링

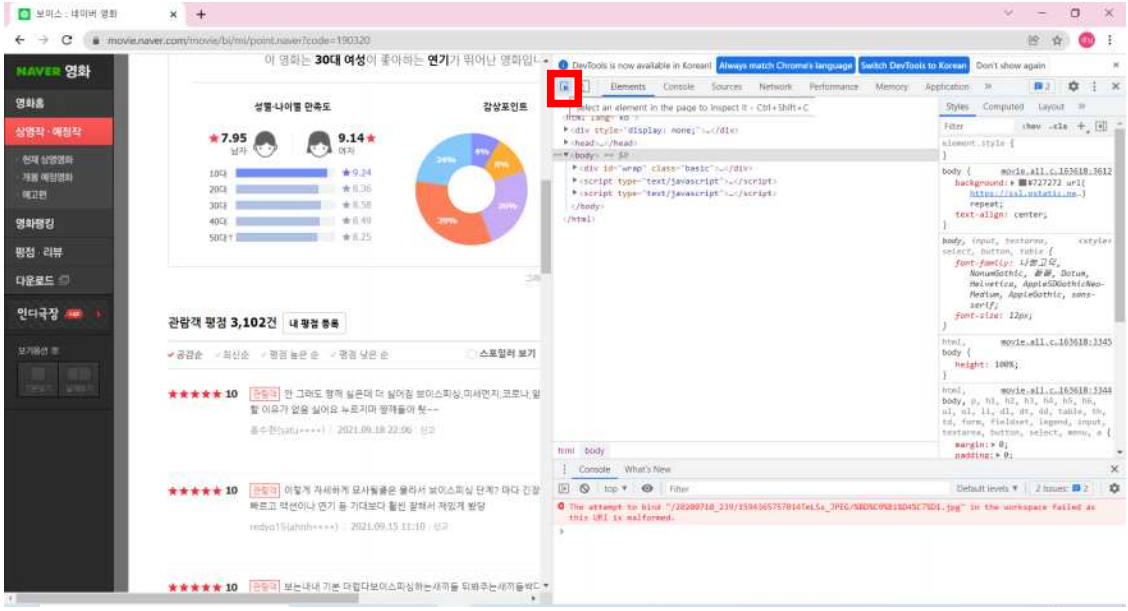
N사 검색창에 '영화평점'을 검색하여 네이버영화를 클릭한다.



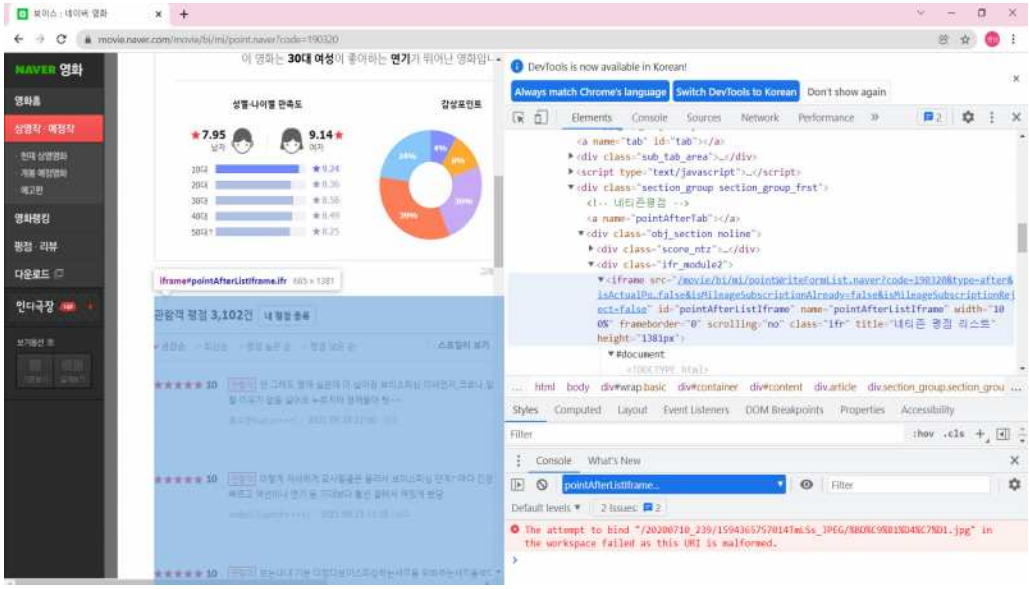


원하는 영화를 검색하고, 평점을 클릭한 후 키보드의 [F12] 버튼을 눌러 개발자도구를 띄워준다.

URL에서 code값인 190320은 N사에 지정된 보이스의 영화코드번호이다. 다른영화를 검색하여 평점을 분석하게되면 코드값은 해당 영화에 맞게 변경된다.

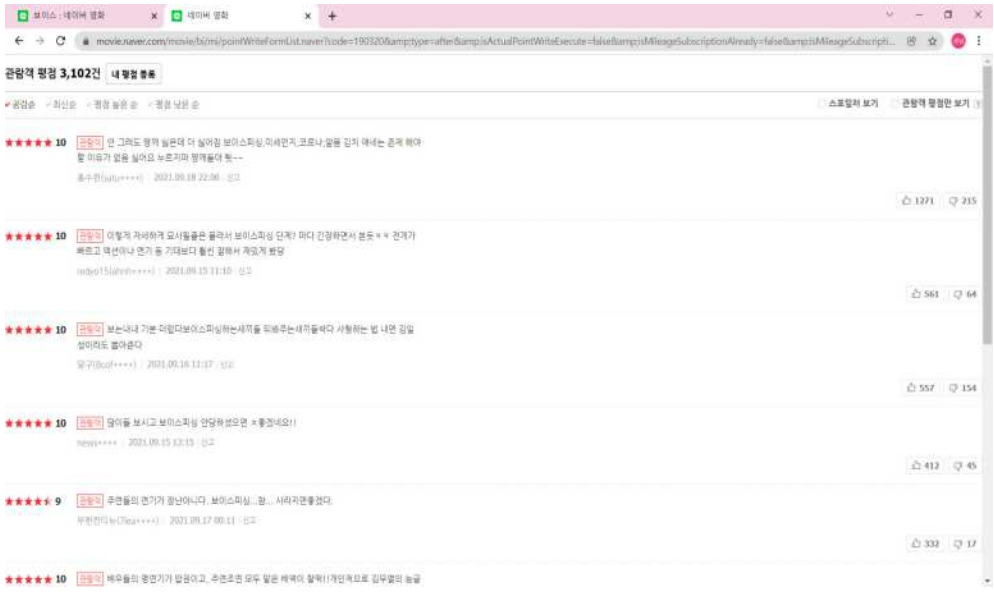


[F12]버튼을 누르면 우측에 개발자도구가 뜨고, Ctrl+Shift+C를 눌러 관람객평점 데이터 있는 부분을 선택하면 아래와 같이 평점 부분의 웹 코드를 확인해볼 수 있다.



iframe src에 있는 주소를 메모장에 붙여놓으면 /movie/bi/mi/pointWriteFormList.naver?code=190320&type=after&ActualPointWriteExecute=false&isMileageSubscriptionAlready=false&isMileageSubscriptionReject=false 이고, 새 탭을 열어 N사 영화 사이트의 평점이기에 앞에 movie.naver.com 뒤로 붙여넣는다.

movie.naver.com/movie/bi/mi/pointWriteFormList.naver?code=190320&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=false&isMileageSubscriptionReject=false



[그림 8-1] 위 URL을 새 탭에 붙여넣은 결과

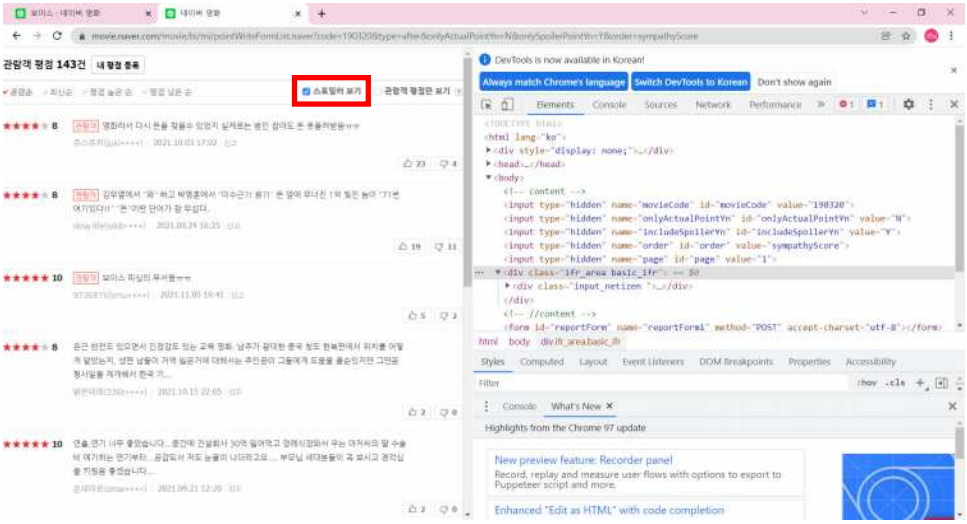
이제 공감순이 아니라 최신순, 높은평점순, 평점낮은순을 차례대로 눌러보고 URL에서 어떤 것이 바뀌는지 확인하고 비교해본다. 또한 F12를 눌러 개발자도구에서도 어떤게 바뀌는지, 스포일러 보기와 관람객 평점만 보기를 눌렀을 때 어떤게 변화하는지 확인한다.

[표 8-1] order 값 변화 확인

order	URL
공감순	https://movie.naver.com/movie/bi/mi/pointWriteFormList.naver?code=190320&type=after&onlyActualPointYn=N&onlySpoilerPointYn=N&order=***sympathyScore***
최신순	https://movie.naver.com/movie/bi/mi/pointWriteFormList.naver?code=190320&type=after&onlyActualPointYn=N&onlySpoilerPointYn=N&order=***newest***
평점높은순	https://movie.naver.com/movie/bi/mi/pointWriteFormList.naver?code=190320&type=after&onlyActualPointYn=N&onlySpoilerPointYn=N&order=***highest***
평점낮은순	https://movie.naver.com/movie/bi/mi/pointWriteFormList.naver?code=190320&type=after&onlyActualPointYn=N&onlySpoilerPointYn=N&order=***lowest***

스포일러 보기를 체크하면 URL에서 onlySpoilerPointYn값이 Y로 변경되고, 관람객 평점만 보기를 체크하면 URL에서는 onlyActualPointYn값이 Y로 변경되는 것을 볼 수 있다.

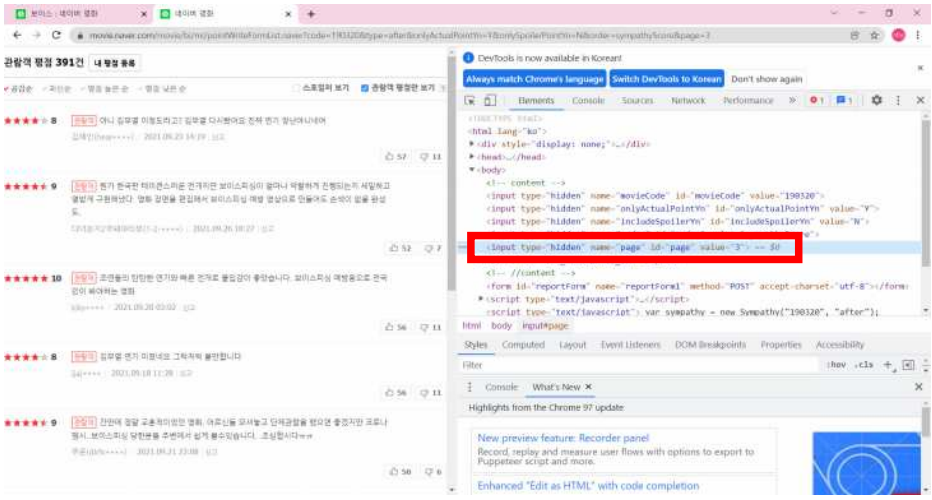
체크	URL
스포일러 보기	https://movie.naver.com/movie/bi/mi/pointWriteFormList.naver?code=190320&type=after&onlyActualPointYn=N&***onlySpoilerPointYn=Y***&order=sympathyScore
관람객 평점만 보기	https://movie.naver.com/movie/bi/mi/pointWriteFormList.naver?code=190320&type=after&***onlyActualPointYn=Y***&onlySpoilerPointYn=N&order=sympathyScore



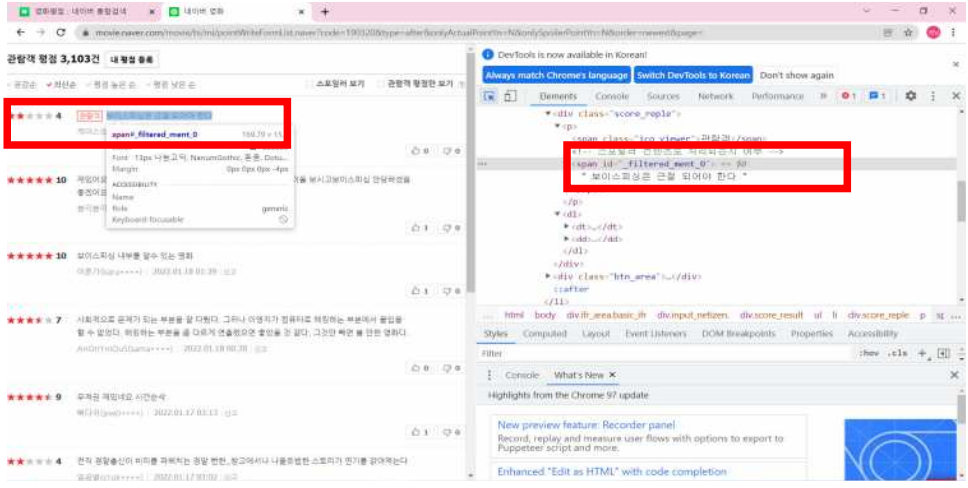
[그림 8-2] 스포일러보기 체크시 includeSpoilerYn 의 value값이 Y로 변경되는 것을 확인

더불어 밑으로 스크롤을 내려 2번째, 3번째 페이지를 넘겨보며 URL을 비교한다.
2번째 페이지를 클릭한 후 URL을 복사해서 메모장에 붙여넣어보면,
https://movie.naver.com/movie/bi/mi/pointWriteFormList.naver?code=190320&type=after&onlyActualPointYn=Y&onlySpoilerPointYn=N&order=sympathyScore&page=2

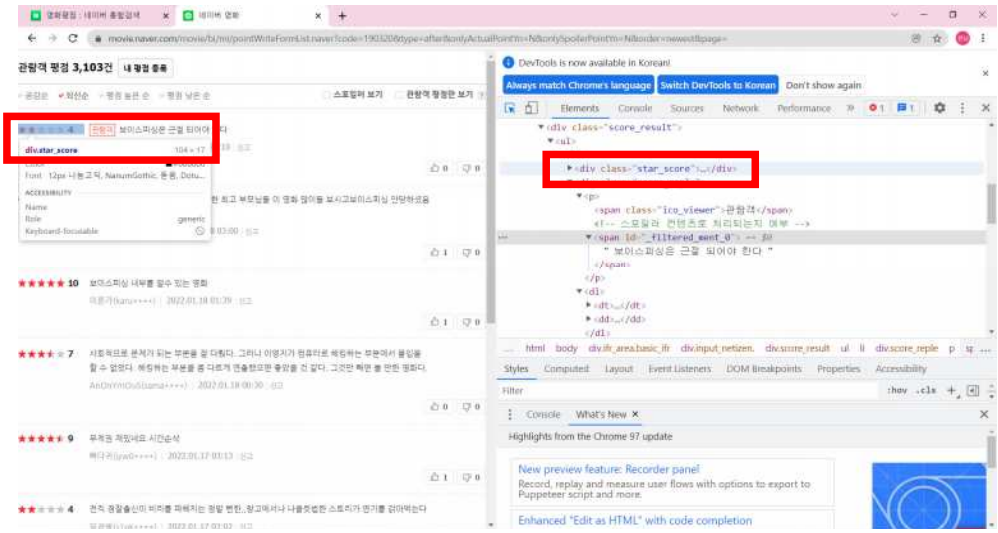
맨 뒤에 page가 추가되었고, 3페이지를 누르면 page값이 3으로 바뀌는걸 확인할 수 있다.
https://movie.naver.com/movie/bi/mi/pointWriteFormList.naver?code=190320&type=after&onlyActualPointYn=Y&onlySpoilerPointYn=N&order=sympathyScore&page=3



[그림 8-3] 페이지 값 3 확인

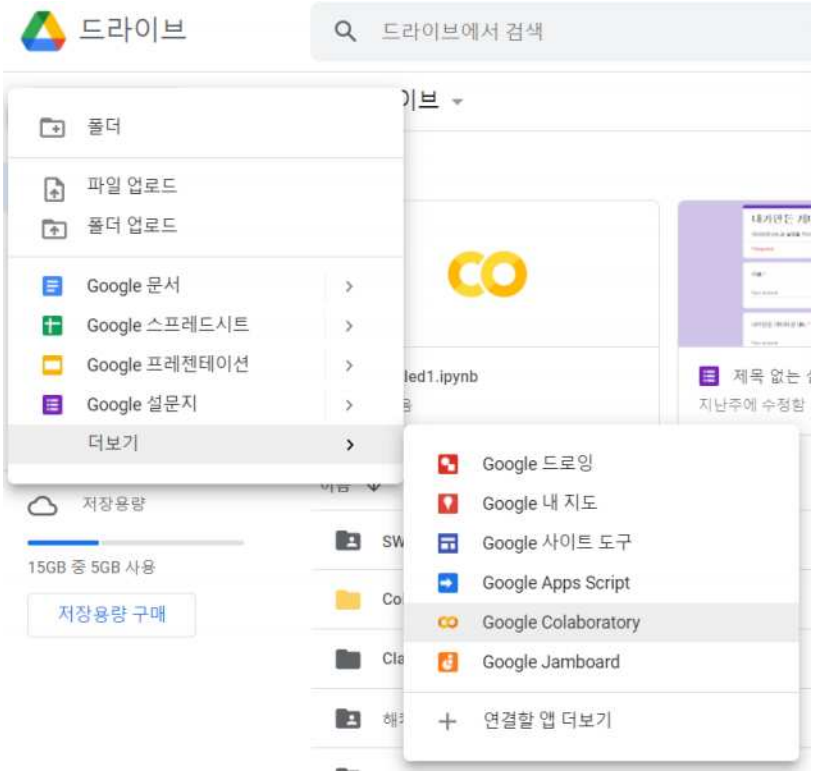


[그림 8-4] 한줄평은 span id가 _filtered_ment_0, _filtered_ment_1 순으로 _filtered_ment_뒤에 숫자가 0부터 증가



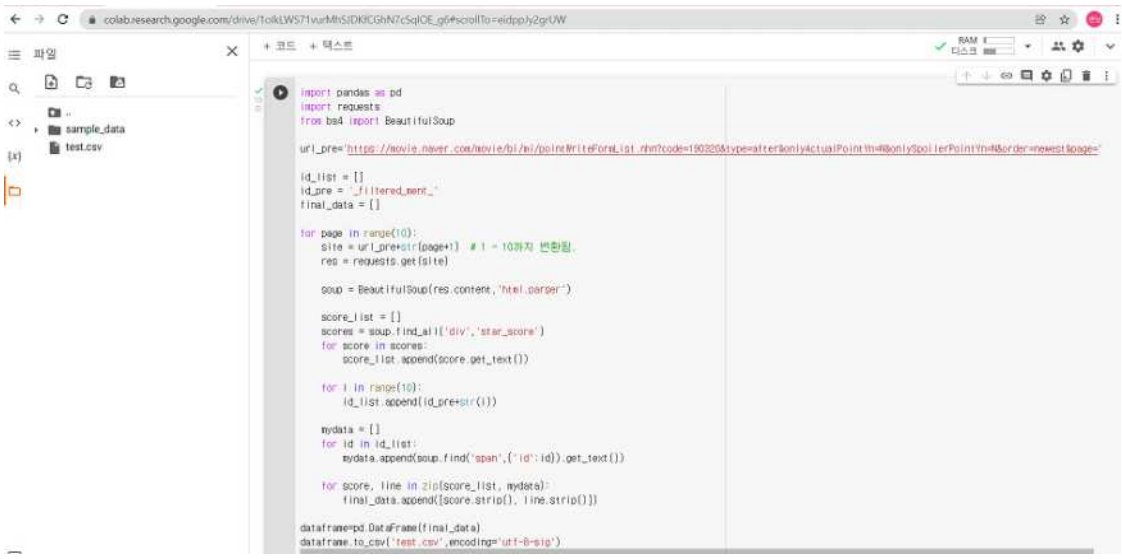
[그림 8-5] 별점은 div class로 star_score 인 것을 확인

대략적인 웹 페이지 분석이 끝났으면 실제적인 크롤링을 하여 위에서 확인했던 평점과 한 줄평을 가져오는 작업을 진행한다. 여기서는 다른 소프트웨어 설치 없이 간편한 구글 코랩을 사용했다. 구글 로그인을 한 후 드라이브로 들어간다.



[그림 8-6] 구글드라이브 - 더보기 - Google Colaboratory

구글코랩은 처음 사용할 때 기본적으로 설치되지 않기 때문에 설치를 해준다. 설치방법은 [연결할 앱 더보기]를 클릭하여 코랩을 설치하고 새로고침을 눌러준다. 설치가 불편하다면 구글코랩을 검색하고 로그인만 해도 사용 가능하다.



[+코드]를 눌러 아래의 코드를 작성한다.

사용할 클래스 패키지

1. **데이터프레임 형태로 자료를 저장하기 위한 pandas**
   pandas는 데이터를 쉽게 분석하고 조작할 수 있는 파이썬 라이브러리이다. panel datas의 약자이고 numpy를 기반으로 파이썬을 활용한 데이터 분석에서 가장 많이 활용되고있으며 데이터 분석을 위한 효율적인 데이터 구조를 제공한다. 그중 1차원 배열 형태의 데이터 구조를 Series(시리즈)라고 부르고, 2차원 배열 형태의 데이터 구조를 DataFrame(데이터프레임)이라고 부른다.

2. **웹크롤링을 위한 BeautifulSoup 과 request**
   BeautifulSoup은 웹 페이지의 정보를 쉽게 스크랩할 수 있도록 기능을 제공하는 라이브러리이고 Requests는 HTTP 요청을 보낼 수 있도록 기능을 제공하는 라이브러리이다.

◆ 100개의 리뷰만 가져오는 웹크롤링 소스코드

```
# 필요한 패키지 사용을 위한 import 작업
import pandas as pd
import requests
from bs4 import BeautifulSoup

# url_pre변수에 크롤링해 올 페이지 주소 문자열로 저장
```

```
url_pre='https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=
190320&type=after&onlyActualPointYn=N&onlySpoilerPointYn=N&order=newest
&page='

# id_list, final_data 변수를 만들어 비어있는 리스트형태로 저장, id_pre변수에는
_filtered_ment_라는 문자열 저장
id_list = []
id_pre = '_filtered_ment_'
final_data = []

for page in range(10): #반복문 (0~9까지 10번 실행)
    site = url_pre+str(page+1)
    res = requests.get(site)
    soup = BeautifulSoup(res.content,'html.parser')
    # 응답받은 res 바이너리 원문에 대해서 html 파싱
    score_list = []
    scores = soup.find_all('div','star_score')
    # 기존에 파싱된 soup에서 div태그의 star_score class를 호출하여 리스트로 반환
    for score in scores:
        score_list.append(score.get_text())
        # scores 리스트 값에서 유니코드 텍스트만 score_list에 추가
    for i in range(10):
        id_list.append(id_pre+str(i))
    mydata = []
    for id in id_list:
        mydata.append(soup.find('span',{'id':id}).get_text())
        # span 속성에서 id값 일치하는 것을 찾아 문자열 반환
    for score, line in zip(score_list, mydata):
  # zip함수를 이용하여 각 원소를 한쌍의 요소로 만들어 묶어준다.
        final_data.append([score.strip(), line.strip()])

dataframe=pd.DataFrame(final_data)
dataframe.to_csv('test.csv',encoding='utf-8-sig')
# test.csv의 형태로 저장하되 한글이 깨지지 않고 저장될 수 있도록 encoding='utf-8-sig'
인코딩 설정
```

◆ 100개의 리뷰만 가져오는 웹크롤링 소스코드 with 코드해석

```
import pandas as pd
import requests
from bs4 import BeautifulSoup
```

*** 패키지 import하여 사용하기**
구글 코랩 환경에서는 따로 pandas나 BeautifulSoup을 설치하지않았다. 하지만 다른개발환경에서 작성할 때는 pip을 이용하여 pandas나 BeautifulSoup을 설치해야할 수 있으니 참고해야한다.

```
>>>pip install beautifulsoup4
>>>pip install requests
>>>
```

```
url_pre='https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=190320&type=after&onlyActualPointYn=N&onlySpoilerPointYn=N&order=newest&page='
```

* 영화 검색 시 코드 : 보이스 - 190320
* 관람객 평점 목록

sympathyScore	공감순
newest	최신순
highest	평점 높은 순
lowest	평점 낮은 순

```
id_list = []
final_data = []

#한줄평은 _filtered_ment_0, _filtered_ment_1, _filtered_ment_2 숫자가 증가했었다.
id_pre = '_filtered_ment_'

for page in range(10):
    site = url_pre+str(page+1)  # 1 ~ 10 페이지까지 변환됨.
    res = requests.get(site)
```

* requests 라이브러리
requests는 파이썬으로 HTTP 호출하는 프로그램을 작성할 때 가장 많이 사용되는 라이브러리이다. requests 라이브러리는 매우 직관적인 API로 어떤 방식의 HTTP 요청을 하느냐에 따라서 해당하는 이름의 함수를 사용하면 된다. 우리는 get방식을 이용하여 데이터를 불러온다.
1. GET 방식: **requests.get()**
 - 응답
  가. **content 속성을 통해 바이너리 원문**을 얻는다.
  나. text 속성을 통해 UTF-8로 인코딩된 문자열을 얻는다.
  다. 응답 데이터가 JSON 포멧이라면 json() 함수를 통해 사전(dictionary) 객체를 얻는다.
2. POST 방식: requests.post()
3. PUT 방식: requests.put()
4. DELETE 방식: requests.delete()

```
#응답받은 res 바이너리 원문에 대해서, html 파싱하겠다.
soup = BeautifulSoup(res.content,'html.parser')

score_list = []

#기존에 파싱된 soup에서 div태그의 star_score class를 호출하여 리스트로 반환
scores = soup.find_all('div','star_score')
```

* find_all로 해당 태그의 class를 기준으로 호출하기
  기준에 맞는 태그를 모두 가져오기 때문에 리스트 타입을 반환한다.

```
for score in scores:
    #scores 리스트 값에서 유니코드 텍스트만 score_list에 추가
    score_list.append(score.get_text())
```

* 변수명.append(값) : 해당 변수에 값 추가
* get.text() : get_text()를 이용하면 한방에 현재 HTML 문서의 모든 텍스트를 추출할 수 있다. 조금 더 정확히 표현하면 get_text() 메서드는 현재 태그를 포함하여 모든 하위 태그를 제거하고 유니코드 텍스트만 들어있는 문자열을 반환한다.

```
#한 페이지당 10개의 리뷰가 있기 때문에 중첩 반복문을 사용
for i in range(10):
    id_list.append(id_pre+str(i))

mydata = []
for id in id_list:
    #span 속성에서 id값 일치하는 것을 찾아 문자열 반환
    mydata.append(soup.find('span',{'id':id}).get_text())
```

* find() : 해당 태그 추출

for score, line in zip(score_list, mydata):

* zip() : 각 원소를 한쌍의 요소로 만들어 묶어준다.

```
>>> numbers = [1, 2, 3]
>>> letters = ["A", "B", "C"]
>>> for pair in zip(numbers, letters):
...     print(pair)
...
(1, 'A')
(2, 'B')
(3, 'C')
```

final_data.append([score.strip(), line.strip()])

* strip() : 공백 및 문자열 제거

```
>>> ex_str = "                    hello          "
>>> ex_str.strip()
# 'hello'
```

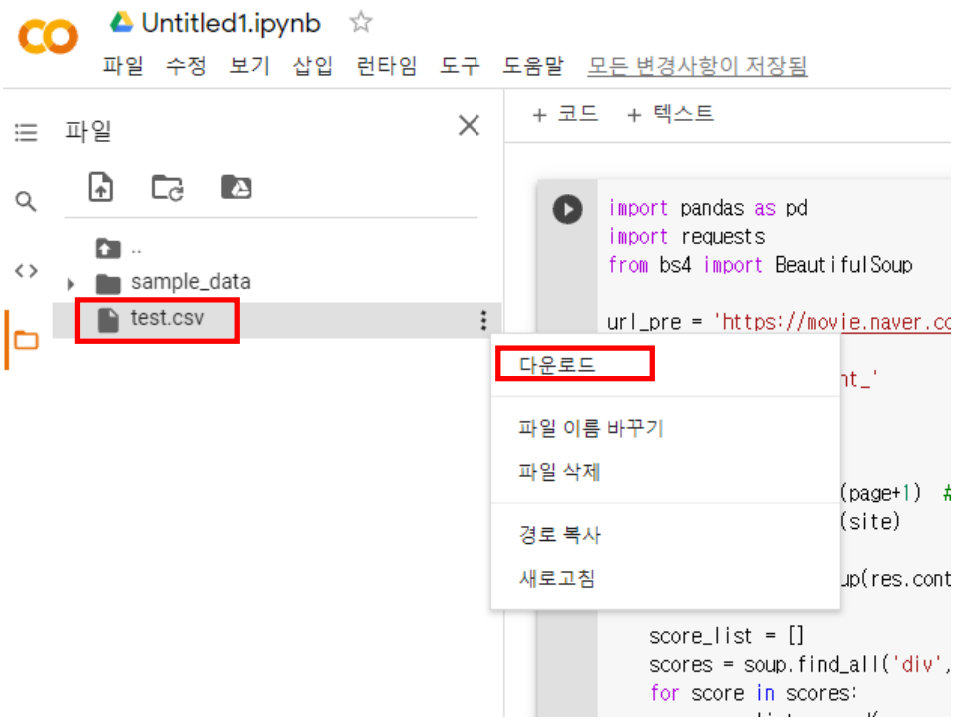
```
dataframe=pd.DataFrame(final_data)

#test.csv의 형태로 저장하되 한글이 깨지지 않고 저장될 수 있도록 encoding='utf-8-sig' 인코딩 설정
dataframe.to_csv('test.csv',encoding='utf-8-sig')
```

◆ 리뷰 전체 페이지-1 페이지까지 가져오는 웹크롤링 소스코드

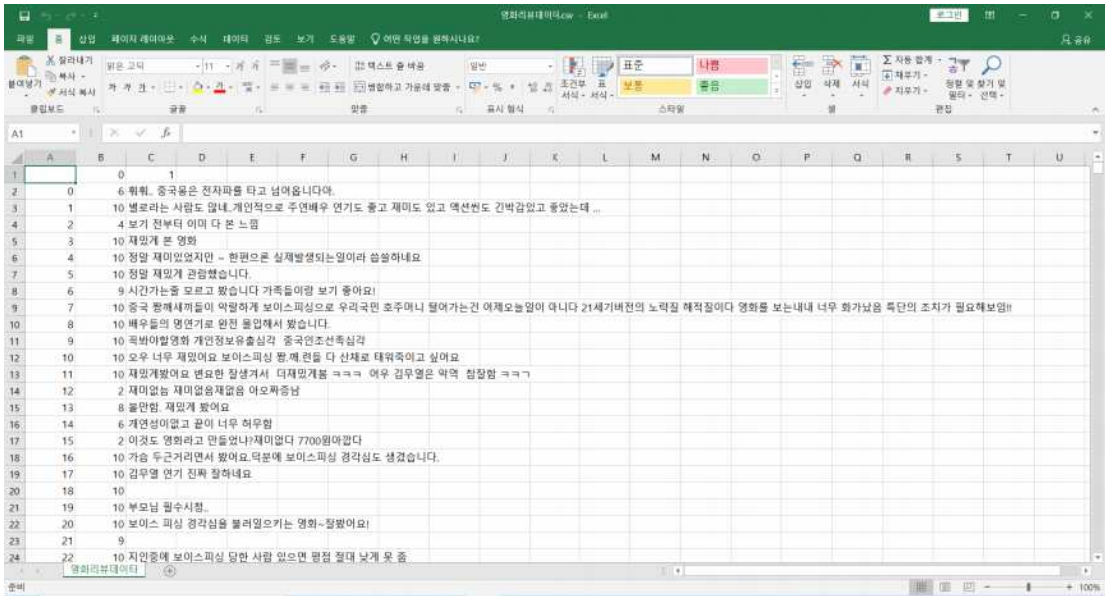
(끝페이지-1페이지를 한 이유는 예외처리를 넣지 않았기 때문이다.)

```
import pandas as pd
import requests
from bs4 import BeautifulSoup

url_pre=
'https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=190320
&type=after&onlyActualPointYn=N&onlySpoilerPointYn=N&order=newest&page='
id_list = []
id_pre = '_filtered_ment_'
final_data = []

result = html.find('div', {'class':'score_total'}).find('strong').findChildren('em')[0].getText()
total_count=int(result.replace(',', ''))

for page in range(int(total_count / 10)):
    site = url_pre+str(page+1)  # 1 ~ 마지막 페이지 -1 까지 변환됨.
    res = requests.get(site)

    soup = BeautifulSoup(res.content,'html.parser')

    score_list = []
    scores = soup.find_all('div','star_score')
    for score in scores:
        score_list.append(score.get_text())

    for i in range(10):
        id_list.append(id_pre+str(i))

    mydata = []
    for id in id_list:
        mydata.append(soup.find('span',{'id':id}).get_text())

    for score, line in zip(score_list, mydata):
        final_data.append([score.strip(), line.strip()])

dataframe=pd.DataFrame(final_data)
dataframe.to_csv('test.csv',encoding='utf-8-sig')
```

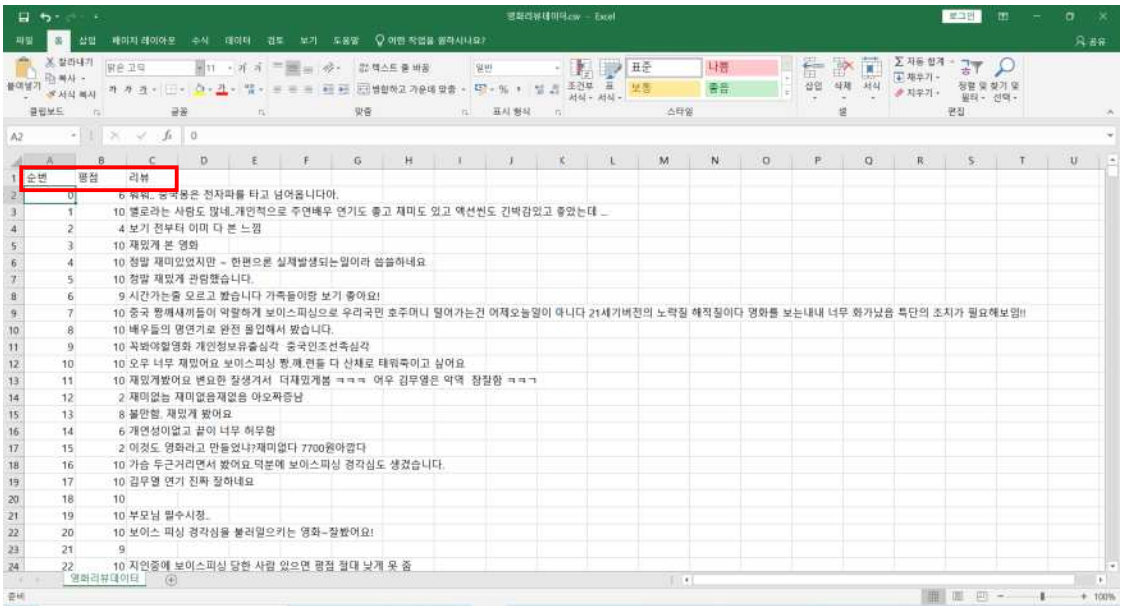
소스코드를 실행시킨 후 코랩 왼쪽의 파일 모양을 클릭하면 웹크롤링한 결과인 csv형태의 파일이 보인다.



코랩의 파일에 test.csv 파일을 다운로드받아 내 컴퓨터에 저장한 후 원하는 데이터만 크롤링 해 왔는지 확인한다.



[그림 8-7] 다운받은 데이터를 확인하고 속성값에 알맞게 속성명을 변경



순번	평점	리뷰
0	6	뭐워.. 중국몽은 전자파를 타고 넘어옵니다아.
1	10	멜로라는 사람도 많네..개인적으로 주연배우 연기도 좋고 재미도 있고 액션씬도 긴박감있고 좋았는데 _
2	4	보기 전부터 이미 다 본 느낌
3	10	재밌게 본 영화
4	10	정말 재미있었지만 ~ 한편으론 실제벌생되는일이라 씁쓸하네요
5	10	정말 재밌게 관람했습니다.
6	9	시간가는줄 모르고 봤습니다 가족들이랑 보기 좋아요!
7	10	중국 짱깨새끼들이 악랄하게 보이스피싱으로 우리국민 호주머니 털어가는건 어제오늘일이 아니다 21세기버전의 노략질 해적질이다 영화를 보는내내 너무 화가났음 특단의 조치가 필요해보임!!
8	10	배우들의 명연기로 완전 몰입해서 봤습니다.
9	10	꼭봐야할영화 개인정보유출심각 중국인조선족심각
10	10	오우 너무 재밌어요 보이스피싱 짱깨,한들 다 산채로 태워죽이고 싶어요
11	10	재밌게봤어요 변요한 잘생겨서 더재밌게봄 ㅋㅋㅋ 여우 김무열은 악역 잘잘함 ㅋㅋㅋ
12	2	재미없는 재미없음재없음 아오짜증남
13	8	불만함. 재밌게 봤어요
14	6	개연성이없고 끝이 너무 허무함
17	15	2 이것도 영화라고 만들었냐?재미없다 7700원아깝다
18	16	10 가슴 두근거리면서 봤어요 덕분에 보이스피싱 경각심도 생겼습니다.
19	17	10 김무열 연기 진짜 잘하네요
20	18	10
21	19	10 부모님 할수시정.
22	20	10 보이스 피싱 경각심을 불러일으키는 영화~잘봤어요!
23	21	9
24	22	10 지인중에 보이스피싱 당한 사람 있으면 평점 절대 낮게 못 줌

[그림 8-8] 가독성을 높이기 위해 속성명을 [순번, 평점, 리뷰]로 변경하고, 파일명도 영화 리뷰 데이터로 변경

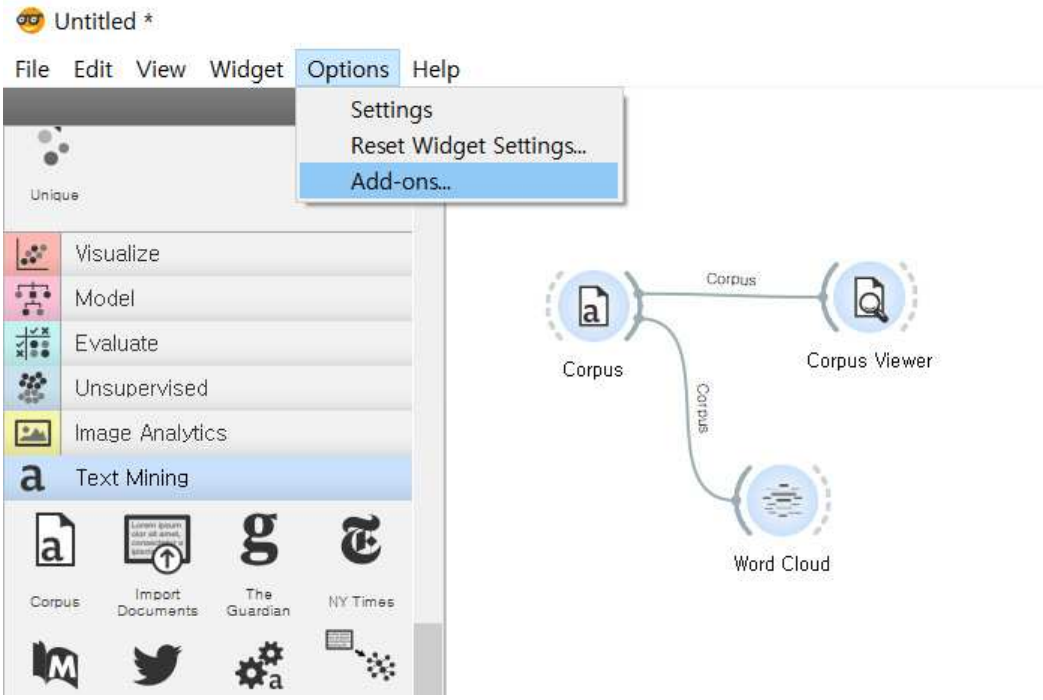
	A	B	C
2895	2893	9	하고자 하는 이야기를 아주 짙요하게 담아냈어요. 특히 변요한 배우의 눈빛 연기가 아주 애절함ㅋ
2896	2894	10	그어떤 공포영화보다 무섭다. 영화보다 드라마같이 좀 더 많은 사람들이 볼수있으면 좋을것같은작품.
2897	2895	10	생각보다 꿀잼이고 개사이다네ㅋㅋ
2898	2896	10	
2899	2897	9	보이스피싱에대해 경각심을 갖게하는 영화?
2900	2898	10	많이들 보시고 보이스피싱 안당하셨으면 ㅈ좋겠네요!!
2901	2899	10	개인적으로 코로나 이후론 제일 재미있게 봤고 느낀것도 많고 역대급으로 봐도 몇손가락안에 드는 영화입니다 강력추천합니다 평점알바아닙니다
2902	2900	9	변요한씨 팬이 아니라서가 아니라 스토리 지루하지않고 재밌게잘봤습니다10점은 아니지만 9점은 줄수있는 영화에요상치안보고ㅋ보길잘했어요코시국 언능 끝나서 영화
2903	2901	10	보이스피싱 사기는 녀석들이 돈시작하여 돈으로 끝나는 대화입니다 국민여러분 의심하시고 국번없이 112 바로 전화하세요 모든 세상 평화 시작 됩니다 감사합니다
2904	2902	10	어? 재밌네~~^^코로나로 간만에 본 영화인데지인, 가족에게 보라고 이야기해줄만한 영화
2905	2903	10	진짜진짜 재미있음 꼭보세용
2906	2904	10	변요한 넘 오랜만..ㅋㅋ 잘봤씁니다
2907	2905	10	쫄깃한 액션 빠른 전개 연출 좋았습니다. 재밌게 보고 갑니다.
2908	2906	10	꼭 보세요 강력 추천! 많은분들이 보시고 보이스피싱 예방에 도움이 되셨으면 좋겠습니다!
2909	2907	10	현실고증 제대로임 빡침과 동시에 몰입하게됨
2910	2908	10	
2911	2909	10	여러 의미로 보면 참 좋을거 같은 영화.. 배우님들 연기도 잘해서 화도 나고 암튼 재밌게 봤다
2912	2910	10	하아.. 열받네 정말 보이스 피싱은 왜 하는거야 대체
2913	2911	10	보는데 현실 이입 제대루 ㅜ 쫄리고 난리
2914	2912	10	완전 박진감 넘침~^^재미있고 스릴♡♡보이스피싱 너무 가슴아프지만 이런 영화보고 경각심을ㅡㅡㅜ
2915	2913	10	겁나 재밌음! 전개가 빨라서 몰입감도 좋았고 긴장감도 엄청났음. 특히 영화 그 특유의 쨍한 미장센이 너무 좋았음. 그리고 모든 배우분들이 연기를 잘하셔서 더 몰입해서
2916	2914	10	재밌게 잘봤어요 느낀게많은영화
2917	2915	9	재밌게 봤습니다. 돈 관련된 전화오면 무조건 조심합시다!!
2918	2916	10	진짜 보이스피싱..놈들..진짜 열받아..
2919	2917	7	김무열님 연기 너무 좋았습니다. 진짜 캐릭터 그 자체로 보여서 몰입이 잘 되었구요 ㅎㅎ 반생이나 콜센터 같은 디테일은 좋았는데 사건 해결이 너무 쉽게 풀어나가지는
2920	2918	10	이렇게 자세하게 모사될줄은 몰라서 보이스피싱 단계? 마다 긴장하면서 본듯ㅋㅋ 전개가 빠르고 액션이나 연기 등 기대보다 훨씬 잘해서 재밌게 봤당
2921	2919	10	괜차나요용 재미떠요용

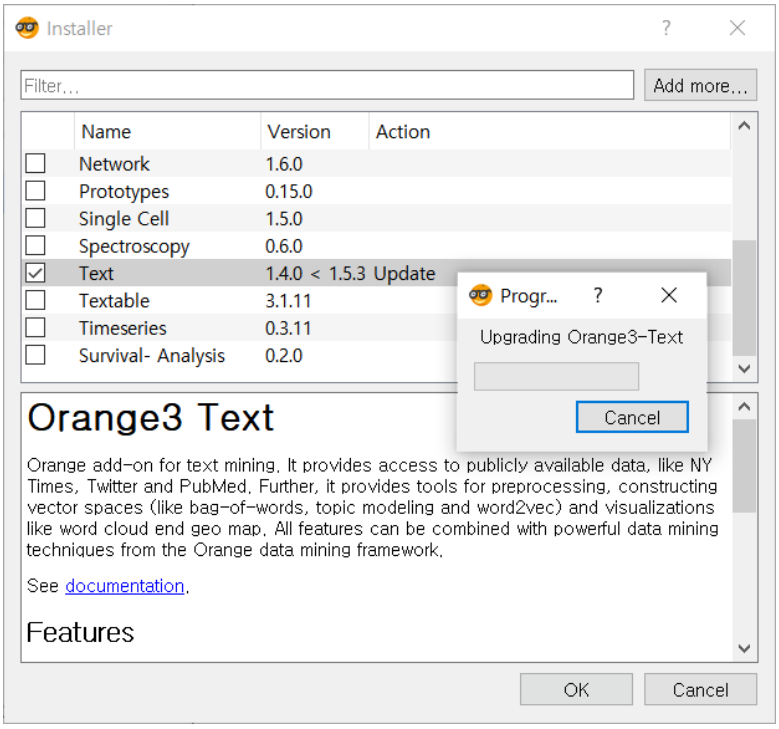
[그림 8-9] 다운로드한 평점 데이터는 총 2920개(순번이 0부터 시작했기 때문에)

다운받은 데이터를 이용하여 오렌지3로 어떤 단어가 많이 등장하는지 확인해보자.

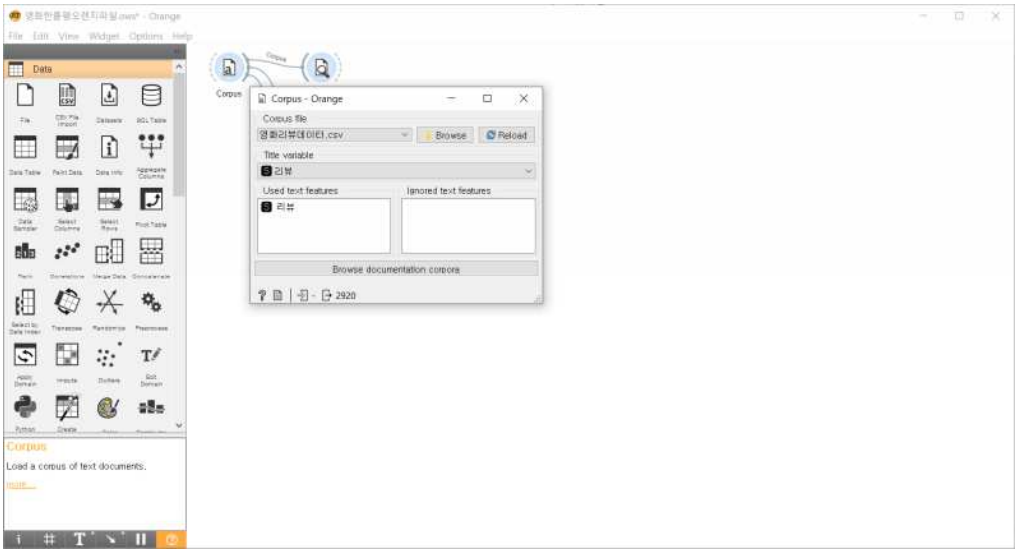
2 텍스트 데이터 불러오기

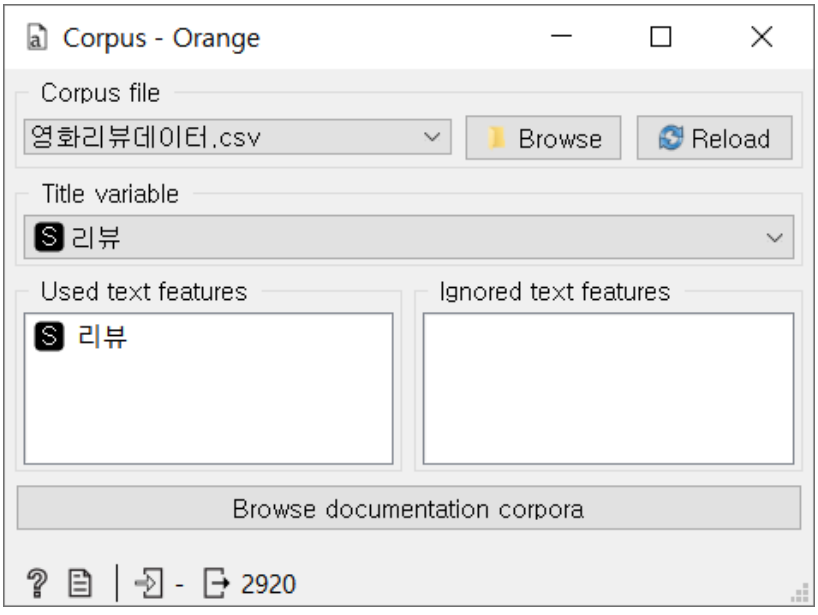
① Orange3을 실행하여 텍스트 분석을 위해 [Options]-[Add ons]-[Text] 체크 후 OK를 눌러 분석에 필요한 메뉴를 다운받는다.





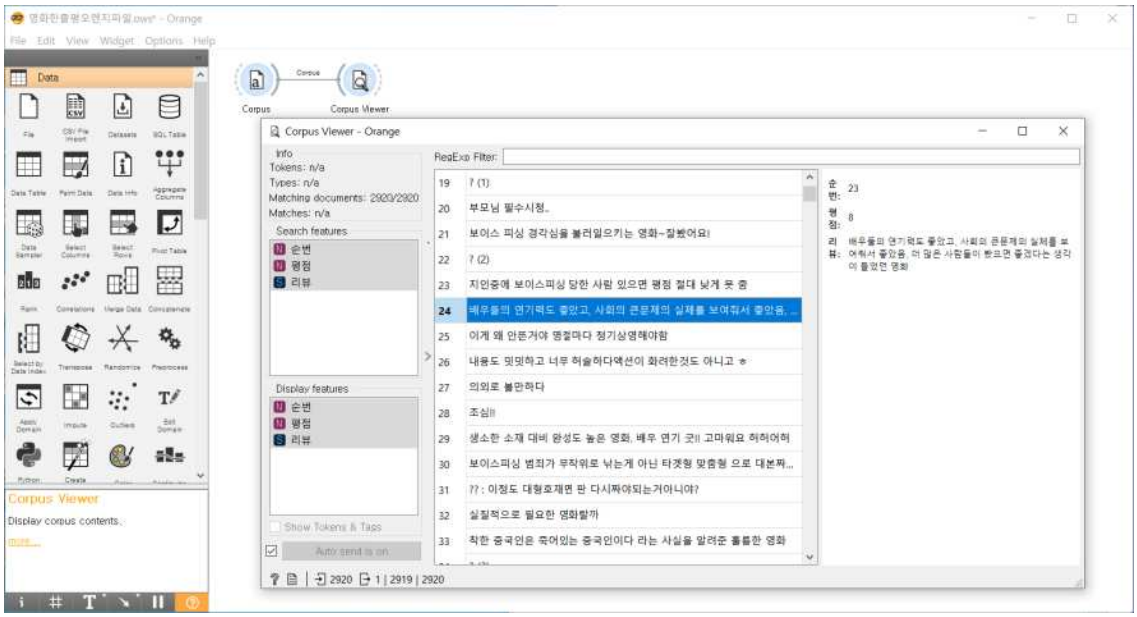
② Coupus를 이용하여 데이터 속성을 확인하고 텍스트분석에 사용할 특성과 사용하지 않을 특성을 선택한다. corpus 위젯은 Excel(.xlsx), 쉼표로 구분된(.csv) 및 탭으로 구분된 기본(.tab) 파일에서 데이터를 읽는다. 구글 코랩을 통해 웹크롤링 한 N사 영화 평점 리뷰데이터를 불러온다.



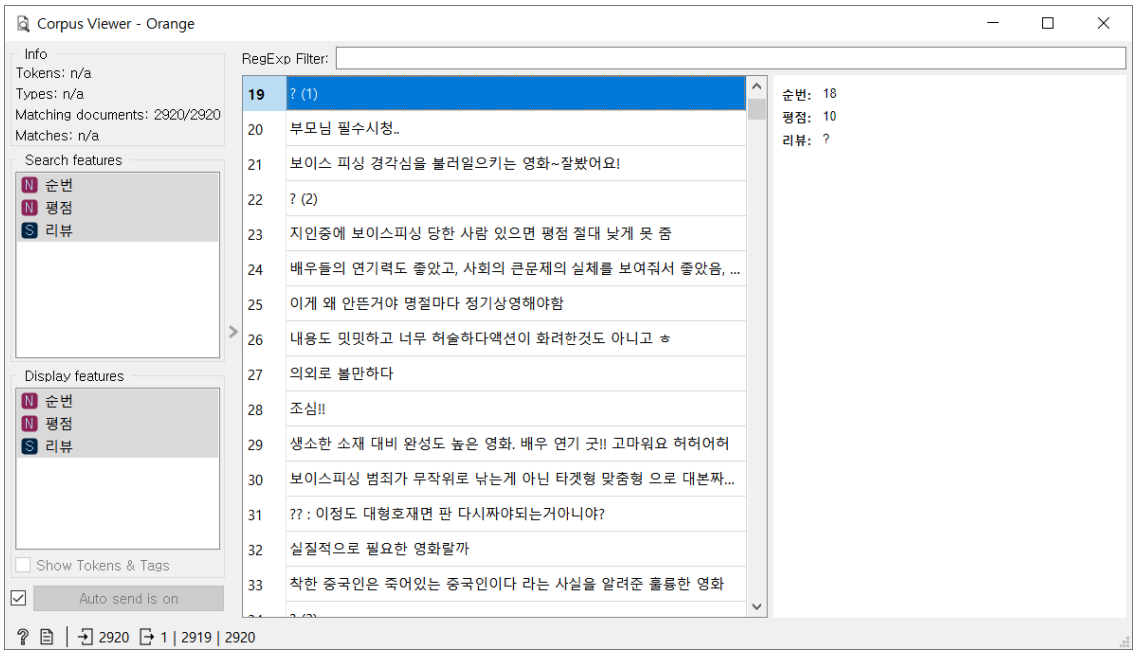


Title variable = Corpus Viewer에서 문서 제목으로 표시되는 변수
Used text features = 텍스트분석에 사용되는 특성
Ignored text features = 텍스트분석에 사용되지않는 특성

③ Corpus Viewer를 이용하여 데이터 속성과 속성 값을 확인한다.
업로드 한 파일의 데이터와 데이터 속성을 확인할 수 있다.



– 업로드 한 파일 뷰어 결과
속성이 순번, 평점, 리뷰로 이루어져 있으며 19행과 22행에 ? 는 별점만 넣고 리뷰는 작성하지 않아 비어있는 결측치 값을 의미한다.

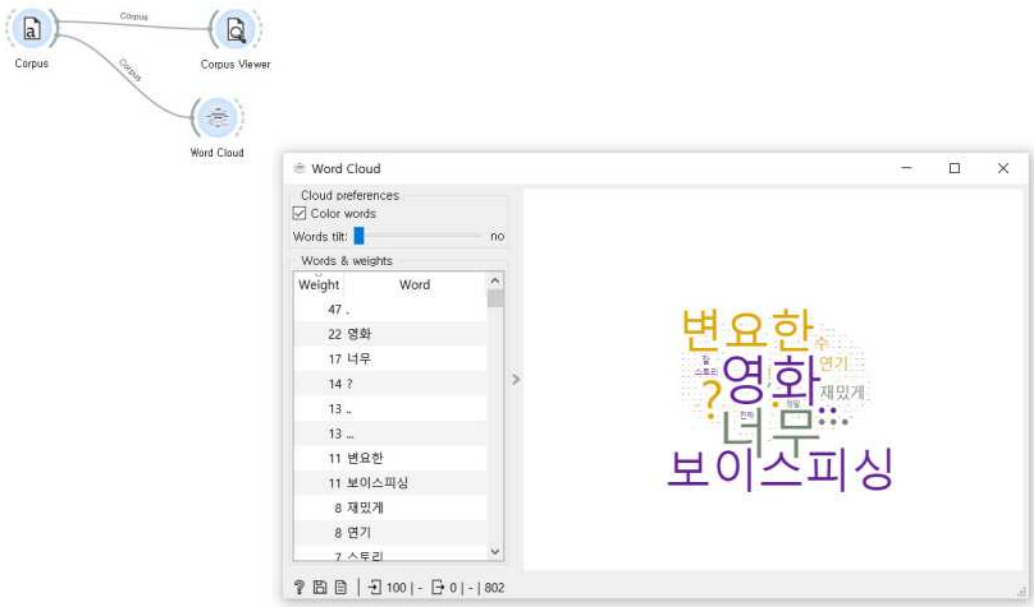


[그림 8–10] 19행(순번 18번)에 리뷰는 ? 인 것을 확인

3 Word Cloud를 이용해 데이터 시각화하기

워드클라우드를 통해 단어들의 등장한 빈도에 따라 나열되어 시각화되는 것을 확인할 수 있다. 워드클라우드를 통한 텍스트 분석 결과로 마침표(.)가 47건으로 가장 많았고, '영화'라는 단어가 22건, '너무'라는 단어가 17건, 비어있던 결측값을 채운 물음표(?)가 14건 등 어떤 단어들이 많이 등장했는지 한눈에 확인할 수 있다.

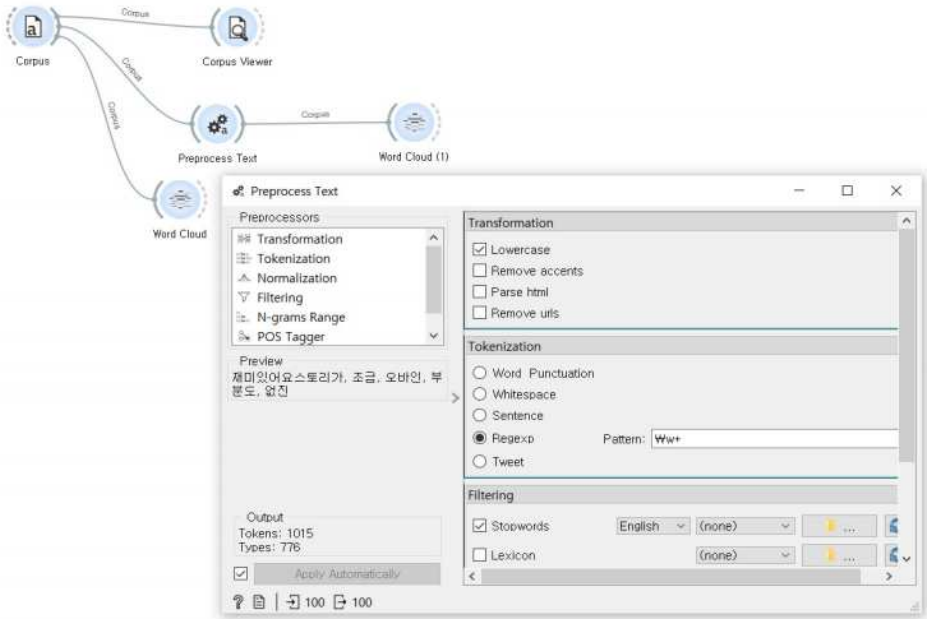
▸ **100개의 리뷰만 추출한 파일을 워드클라우드 처리한 결과**



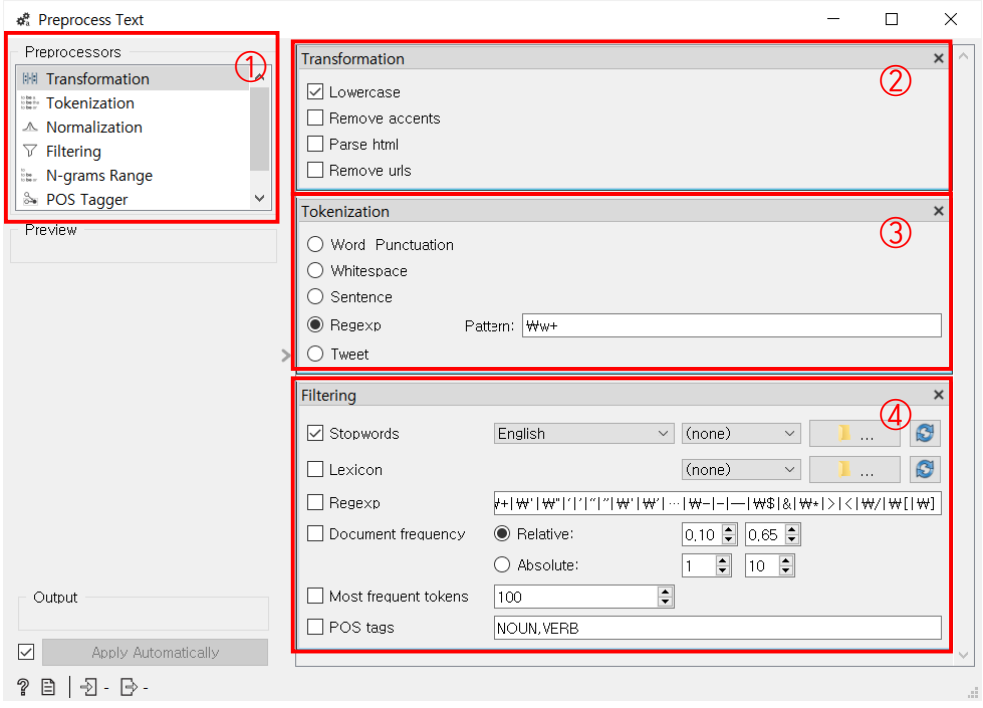
▸ **전체 리뷰 추출한 데이터 파일을 워드클라우드 처리한 결과**

4 Process Text로 데이터 전처리하기

많이 등장하는 단어 분석에서 불필요한 단어들은 검색에서 제외하는 방법을 통해 데이터 전처리 작업을 진행한다.



▸ Preprocess Text 기능 중 사용한 기능 설명



1. Preprocessors : 사용 가능한 전처리기

사용가능할 전처리기가 많지만 기본적으로 설정되어있는 값으로 전처리를 진행하였다.

2. Transformation : 입력 데이터를 변환하며 기본적으로 소문자 변환을 적용한다.

가. Lowercase : 모든 텍스트를 소문자로 바꾼다.
나. Remove accents : 텍스트의 모든 발음 구별 부호/악센트를 제거한다.
다. Parse html : html 태그를 감지하고 텍스트만 구문 분석한다.
[예시] 〈a href…〉일부 텍스트〈/a〉 → 일부 텍스트
라. Remove urls : 텍스트에서 URL 을 제거한다.
[예시] http://orange.biolab.si/ url입니다. → url입니다.

3. Tokenization : 텍스트를 더 작은 구성요소(단어, 문장, 빅그램)로 나누는 방법

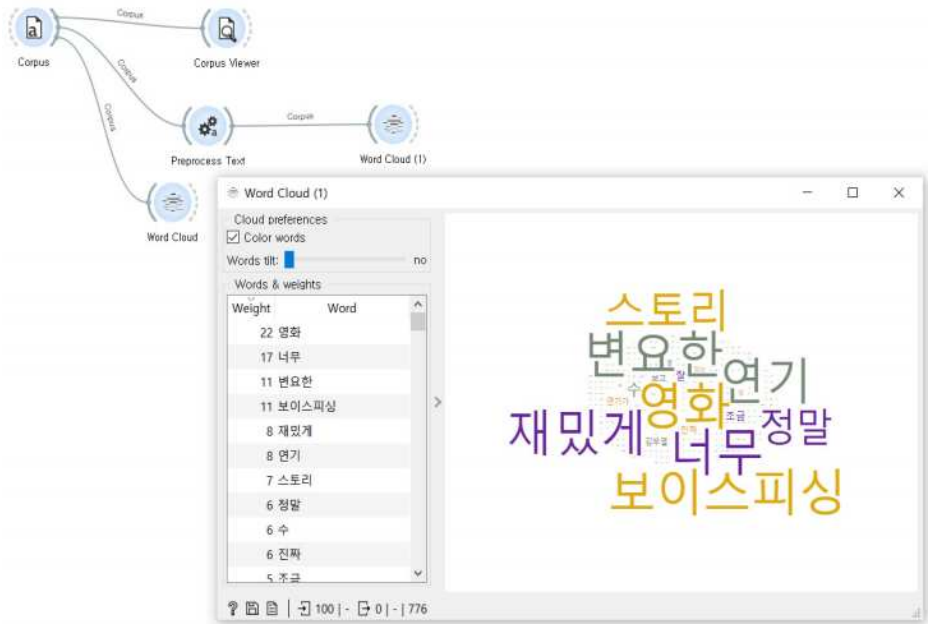
가. Word & Punctuation : 텍스트를 단어로 나누고 구두점 기호를 유지한다.
나. Whitespace : 공백으로만 텍스트를 분할한다.
다. Sentence : 전체 문장만 유지하면서 마침표로 텍스트를 분할한다.
라. Regexp : 제공된 정규식으로 텍스트를 분할한다. 기본적으로 단어로만 분할된다. (구두점 생략)
마. Tweet : 해시태그, 이모티콘 및 기타 특수 기호를 유지하는 사전 훈련된 Twitter 모델로 텍스트를 분할한다.

4. Filtering : 단어 선택을 제거하거나 유지한다.

가. Stopwords : 텍스트에서 접속사('와/과', '또는')를 제거한다.
나. Lexicon : 파일에 제공된 단어만 보관한다.
다. Regexp : 정규식과 일치하는 단어를 제거한다. 기본값은 구두점을 제거하도록 설정되어 있다.
라. Document frequency : 지정된 문서 수/백분율보다 작거나 크지 않은 토큰을 유지한다. Absolute는 지정된 문서 수에 나타나는 토큰만 유지한다. Relative는 문서의 지정된 백분율에 나타나는 토큰만 유지한다.
마. Most frequent tokens : 지정된 수의 가장 빈번한 토큰만 유지한다. 기본값은 가장 자주 사용되는 토큰 100개이다.

5 전처리 결과 Word Cloud 확인

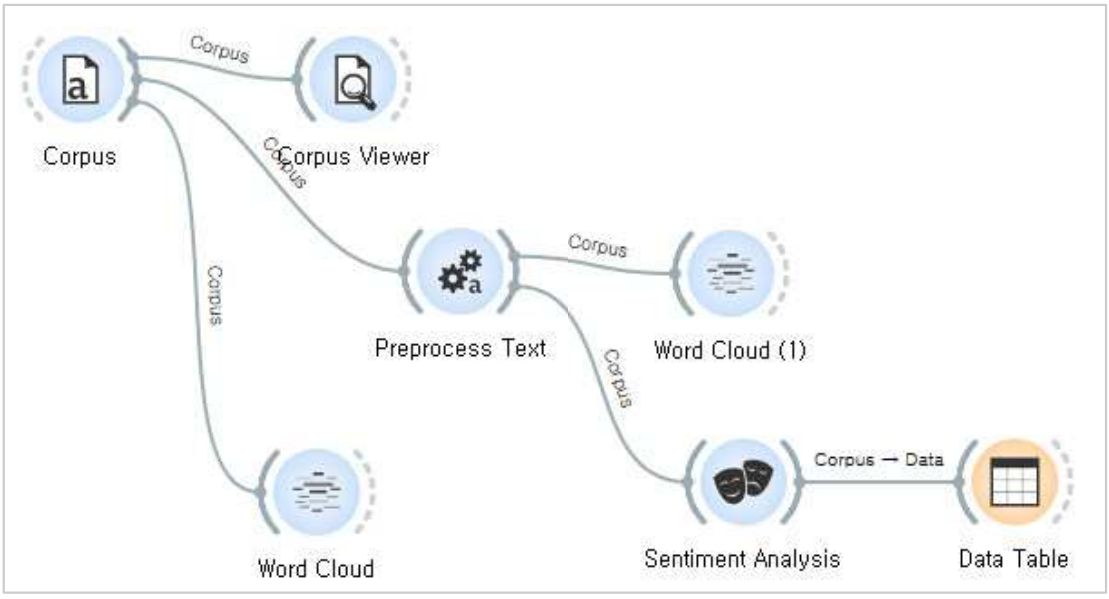
▸ **100개의 리뷰만 추출한 파일을 전처리 후 워드클라우드 결과**



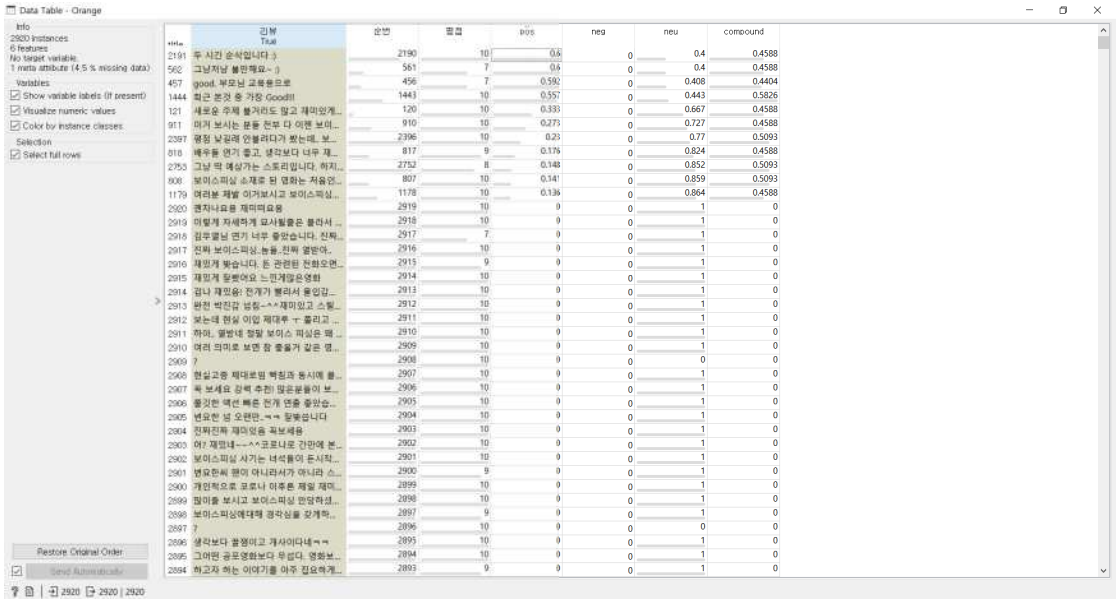
▸ **전체 리뷰 추출한 데이터 파일을 전처리 후 워드클라우드 결과**

데이터 전처리 후 워드클라우드를 결과를 보면 분석에 필요 없는 단어들이 정리되어있음을 확인할 수 있다. 한줄평의 단어를 분석하는 활동이기에 영화의 모든 줄거리를 파악할 수는 없겠지만 무엇에 관련된 영화이고 등장인물은 누구인지, 어떤 내용을 담고 있는지 이해할 수 있다.

6 전처리 결과 데이터로 텍스트 감정 분석



전처리 한 데이터로 감정분석한 결과를 테이블로 확인해볼 수 있다. 감정분석 후 데이터테이블을 확인해보면 pos(긍정), neg(부정), neu(중립) 세가지 특성 중 해당되는 영역에 1로 표시가 되어있는 것을 볼 수 있다.



	리뷰 True	순번	평점	pos	neg	neu	compound
562	그냥저냥 볼만해요~ :)	561	7	0.6	0	0.4	0.4588
2191	두 시간 순삭입니다 :)	2190	10	0.6	0	0.4	0.4588
457	good. 부모님 교육용으로	456	7	0.592	0	0.408	0.4404
1444	최근 본것 중 가장 Good!!!	1443	10	0.557	0	0.443	0.5826
121	새로운 주제 볼거리도 많고 재미있게 봤음 :)	120	10	0.333	0	0.667	0.4588
911	이거 보시는 분들 전부 다 이젠 보이스피싱 안 ...	910	10	0.273	0	0.727	0.4588
2397	평점 낮길래 안볼려다가 봤는데.. 보길 잘했어...	2396	10	0.23	0	0.77	0.5093
818	배우들 연기 좋고, 생각보다 너무 재밌게 잘 봤...	817	9	0.176	0	0.824	0.4588
2753	그냥 딱 예상가는 스토리입니다. 하지만 연기와...	2752	8	0.148	0	0.852	0.5093
808	보이스피싱 소재로 된 영화는 처음인거 같은데...	807	10	0.141	0	0.859	0.5093
1179	여러분 제발 이거보시고 보이스피싱에 속지않...	1178	10	0.136	0	0.864	0.4588
2881	개연성 0도 없는 진짜 우당탕하다가 그냥 악역 ...	2880	1	0	0	1	0
2829	감각적인 척...센스있는 척...척척척...	2828	1	0	0	1	0
2825	어휴 이게뭐냐.... 얼마만에 본 영환데 킹받네 ㅡ...	2824	1	0	0	1	0
2796	이거 영화가 맞냐? 평점 진심이냐?알바 아니고?	2795	1	0	0	1	0
2775	최악.... 영화가 허술하기 짝이 없네요. 그냥 악...	2774	1	0	0	1	0
2743	?에라이 ㅋㅋ 이래서 개봉 당일 평점은 믿으면 ...	2742	1	0	0	1	0
2723	믿고 거르는 개봉첫날 k영화 평점	2722	1	0	0	1	0

감정분석 결과 중립이 가장 많고, 긍정적인 감정으로 분류되어진 텍스트를 확인해보니 웃음 이모티콘 ;), good 등이 존재하는 것을 확인하였다. 감정분석 시 한국어보다는 영어로 된 데이터로 감정분석을 하면 더 정확하게 확인할 수 있다.

08. 영화 평점 리뷰에서 많이 등장하는 단어는?

정리하기

웹크롤링을 이용하여 N사 영화리뷰 데이터를 수집한 후 어떤 단어, 내용이 리뷰에서 많이 등장하는지 워드클라우드를 이용하여 확인해보았다. 나아가 리뷰들의 감정분석을 통해 긍정적인지 중립인지, 부정적인지도 확인해보았다. 한글로 이루어진 내용의 감정분석은 잘 되진않았지만 음식점이나 호텔 등 다양한 리뷰를 크롤링해보고 적용해보거나 영어로 된 리뷰로 분석을 적용해볼 수 있을 것이다.

[참고 문헌]

1. 서울과학종합대학원 디지털혁신처(2021). 3시간 만에 배우는 인공지능 데이터분석. 오렌지. 서울경제경영.
2. 손원성. 손경호. 황희진. 백우정. 오렌지3로 알아가는 머신러닝 데이터 분석. 홍릉출판.
3. 오렌지. https://orange3-text.readthedocs.io/en/latest/widgets/preprocesstext.html
4. 네이버 영화 평점. https://movie.naver.com/movie/point/af/list.naver
5. strip() 문자열 및 공백제거. https://wikidocs.net/33017
6. zip(). https://www.daleseo.com/python-zip/
7. 영화리뷰 웹크롤링. https://velog.io/@changhtun1/%ED%8C%8C%EC%9D%B4%EC%8D%AC%EC%9D%84-%ED%99%9C%EC%9A%A9%ED%95%9C-%EC%9B%B9-%ED%81%AC%EB%A1%A4%EB%A7%81-3
8. find(). https://seungjuitmemo.tistory.com/203
9. get_text(). https://hogni.tistory.com/21
10. 네이버영화리뷰 웹크롤링. https://kimdingko-world.tistory.com/77

09. 생선 눈 이미지로 생선의 신선도를 구분해볼 수 있을까?

사동고등학교 교사 서 정 민

학습 진행 과정

1단계	데이터 준비	- 사용 데이터: fish data - 수집: 캐글
2단계	데이터 불러오기	- 학습 데이터 불러오기 - 이미지 데이터 임베딩(Embedding)하기
3단계	모델 학습	- 분류를 이용한 모델 학습 - 사용된 알고리즘: k-NN, Random Foreset, Logistic Regression, Neural Network
4단계	성능 평가	- test and score를 이용한 성능 평가 - 혼동 행렬을 이용한 성능 평가
5단계	예측	- Prediction을 이용한 테스트 데이터로 예측하기

학습 내용 요약

데이터 종류	문제 유형	사용된 학습 알고리즘	성능 평가 도구
비정형 데이터(이미지)	분류	k-NN, Random Foreset, Logistic Regression, Neural Network	혼동 행렬

문제 상황

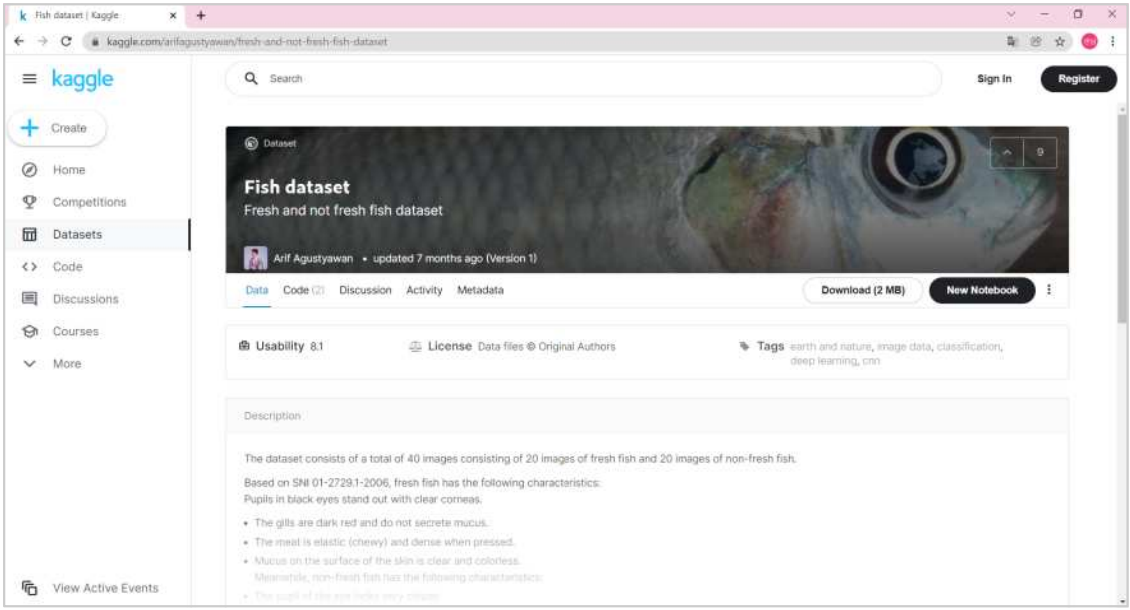
생선의 신선도를 판별하는 요인중에는 몸통의 색과 광택, 눈, 항문, 껍질 등이 있다. 이 중 우리가 가장 흔하고 쉽게 신선도를 판단하는 기준은 눈 일 것이다.

신선한 생선과 오래된 생선의 눈 이미지를 인공지능으로 구분할 수 있을까?

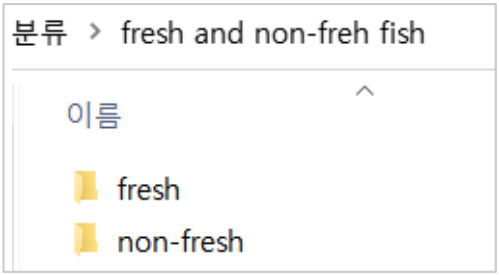
01 데이터 준비하기

1 Fish dataset 데이터 세트

캐글을 이용하여 생선 눈 이미지를 다운받는다. 검색창에 fish를 입력하면 다양한 내용이 검색된다. 검색 내용 중 인공지능 학습에 필요한 데이터가 필요하므로 유형은 [dataset]을 선택하여 dataset을 클릭 후 스크롤을 내려 데이터를 찾는다. 다운받은 데이터를 확인하고 훈련데이터와 테스트 데이터를 나누기위해 새폴더(테스트 데이터 저장)를 지정하고, 각각 5개의 이미지를 새로 만든 폴더로 이동시킨다.

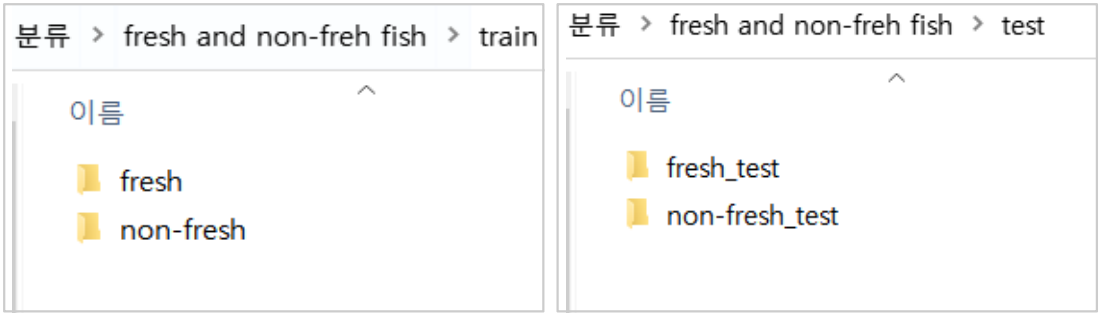


[그림 9-1] 캐글 사이트에서 fish를 검색



[그림 9-2] 다운받은 데이터 구성

[그림 9-3] fresh(좌), non-fresh(우) 각각 20장씩 있다.



[그림 9-4] fresh_test와 non-fresh_test 폴더를 새로 만들어 기존 데이터에서 5개씩 테스트데이터로 이동하여 학습데이터는 15개의 이미지가 테스트데이터는 5개의 이미지가 저장되도록 만든다.

즉 train 폴더 안에는 15개씩 이미지가 들어있는 fresh, non-fresh 폴더가있고, test 폴더 안에는 5개씩 이미지가 들어있는 fresh_test, non-fresh_test 폴더가 있다.

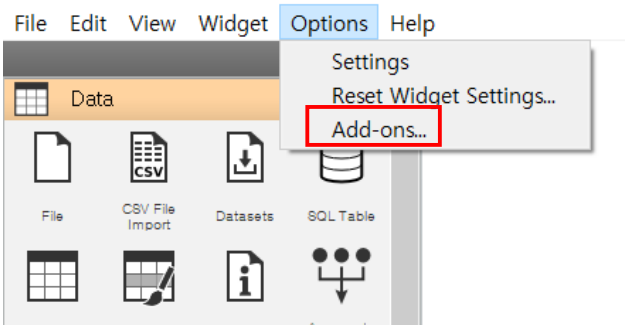
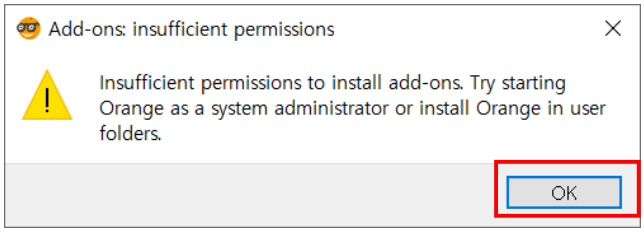
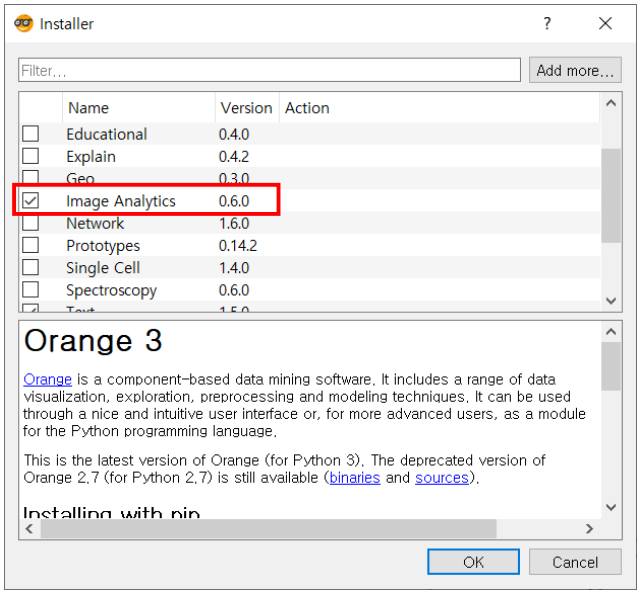
2 데이터 불러오기

① 이미지 분석 위젯 추가하기

이미지 데이터를 불러오기 위해서는 오렌지3의 이미지 분석 위젯을 추가해야 한다. 이 때 사용하는 컴퓨터가 인터넷에 연결된 상태일 때 위젯을 다운로드 받아 설치할 수 있다. 이미

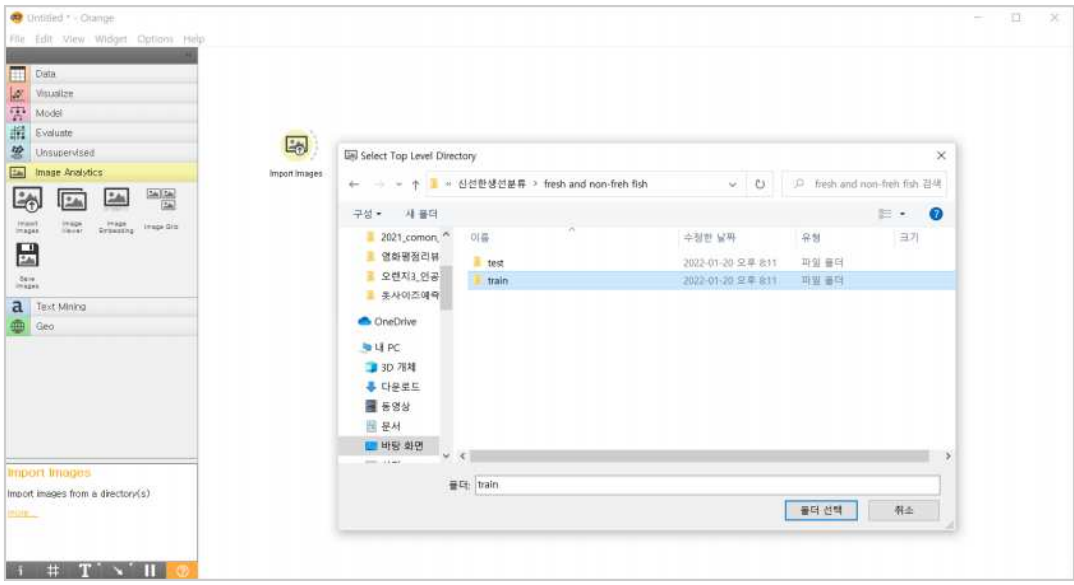
지 분석 위젯을 추가하는 방법은 아래와 같다.

[표 9-1] 이미지 분석 위젯 설치

단계	설명
	[메뉴] - [Options]- [Add-ons..]를 선택한다.
	설치 허가와 관련된 내용을 [OK]를 눌러 확인하고 설치를 진행한다.
	설치 가능한 부가기능 목록에서 [Image Analytics]를 선택하여 설치를 진행한다.

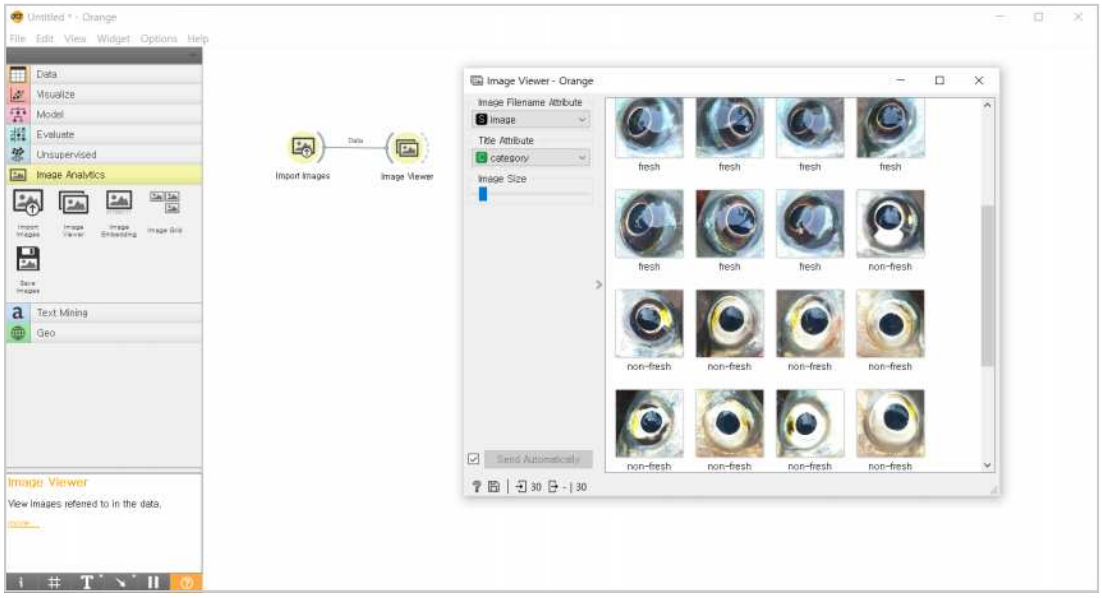
② 이미지 데이터 불러오기

이미지 데이터를 불러오기 위해 Image Analytics - Import Image를 선택하여 캔버스(작업 공간)에 배치한 후 위젯을 더블클릭하여 train 폴더를 가져온다. 그러면 30개의 이미지가 있고 2개의 카테고리로 분류되어있다는 것을 확인할 수 있다.



[그림 9-5] Info를 확인하면 30개의 이미지, 2개의 카테고리 확인

Image Viewer 위젯을 이용하여 가져온 이미지를 확인할 수 있다.

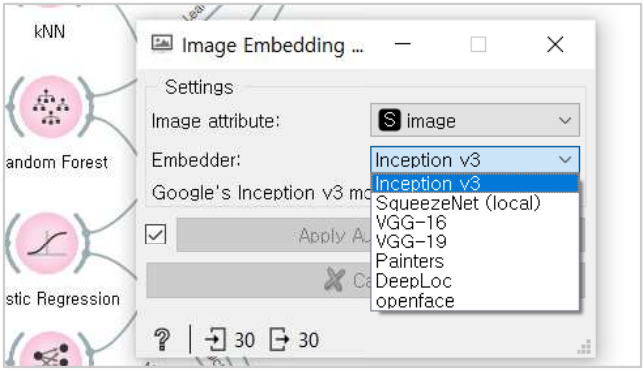


[그림 9-6] Image Viewer 확인

02 데이터 전처리하기

1 이미지 데이터 임베딩 하기

사람은 이미지 그 자체를 인식하지만 컴퓨터는 이미지를 숫자의 배열로 인식하기 때문에 이를 변환하는 과정이 필요하고 이 과정을 이미지 임베딩(Image Embedding)이라고 한다. 이미지를 임베딩하기 위해서는 Image Analytics - Image Embedding을 선택하여 Import Image 위젯과 연결한다. 해당 위젯을 더블클릭하면 설정을 확인할 수 있다. 여기서 Embedder를 이용해 이미 훈련된 이미지 인식 모델을 선택하여 사용할 수 있다.



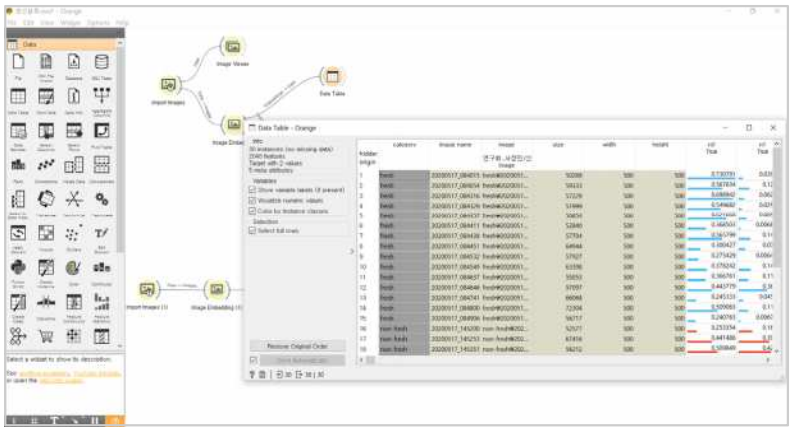
[그림 9-7] 임베딩 종류 설명

[표 9-2] 임베딩 종류 설명

임베더	설명
SqueezeNet	ImageNet에서 훈련된 이미지 인식을 위한 작고 빠른 모델 (Local)
Inception v3	ImageNet에서 훈련된 Google의 Inception v3 모델 (기본값)
VGG-16	ImageNet에서 훈련된 16계층 이미지 인식 모델
VGG-19	ImageNet에서 훈련된 19계층 이미지 인식 모델
Painters	예술 작품 이미지에서 화가 를 예측 하도록 훈련된 모델
DeepLoc	효모 세포 이미지 를 분석하도록 훈련된 모델

2 이미지 속성 확인하기

Data - Data Table 위젯을 선택하여 Image Embedding에 연결한 후 로드된 이미지 정보를 확인한다. 제시된 데이터 속성 중 category는 target 값으로 사용되는 속성이다. 학습 데이터 폴더 하위에 분류 항목을 폴더로 구성해야 카테고리로 인식되어 학습에 사용할 수 있다. 이미지 임베딩이 과정을 거쳤기 때문에 이미지가 숫자값으로 변환된 것을 확인할 수 있다.

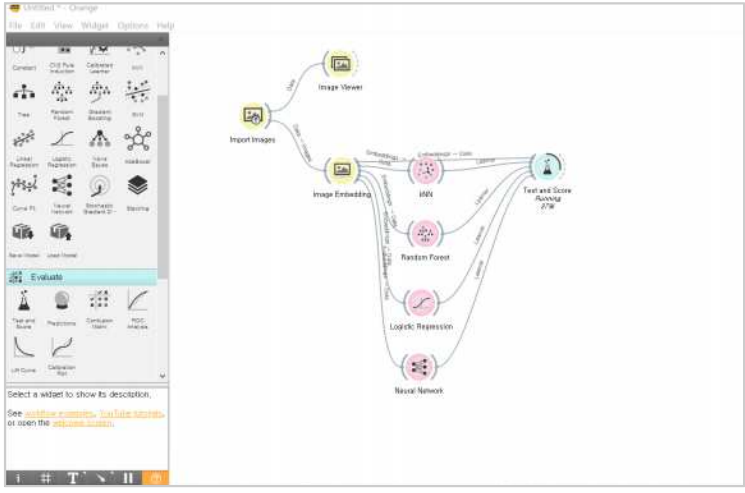


[그림 9-8] 이미지 임베딩 후 데이터테이블 확인

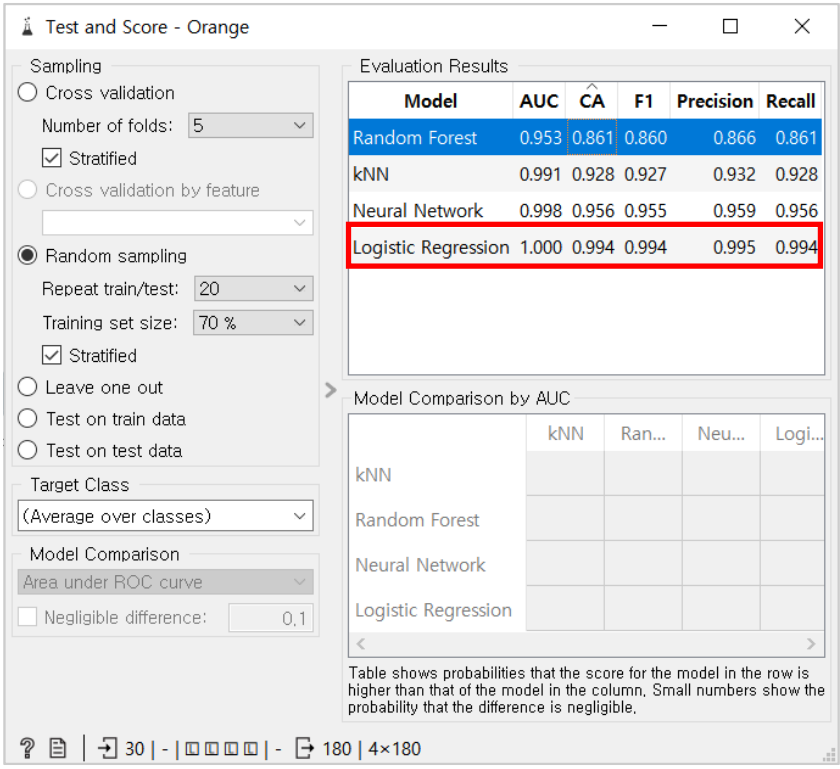
03 모델 학습 및 평가

1 모델 학습하기

기계학습 알고리즘과 데이터를 연결하여 모델 학습한다. 오렌지에서는 다양한 기계학습 알고리즘을 한꺼번에 연결하여 모델을 학습시킬 수 있다. 여기서는 분류에 자주 사용하는 kNN, Random Foreset, Logistic Regression, Neural Network을 이용하여 모델을 구성하였다. model에서 해당하는 학습 알고리즘을 선택하여 File과 연결시킨다. 성능 평가는 Test and Score를 이용하면 된다. Evaluate - Test and Score 위젯을 선택하고 File과 각 학습 알고리즘을 연결시켜 준다. Test and Score 위젯에서 모델 학습과 테스트 데이터의 비율을 설정할 수 있다. 아래는 Cross validation도 체크해주었고, Test and Score 위젯에서 Random sampling에서 Repeat train/test값을 기본 10에서 20으로 늘려 학습시켰다.



[그림 9-9] 인공지능 모델 학습



[그림 9-10] test and score 값 확인

2 성능 평가하기

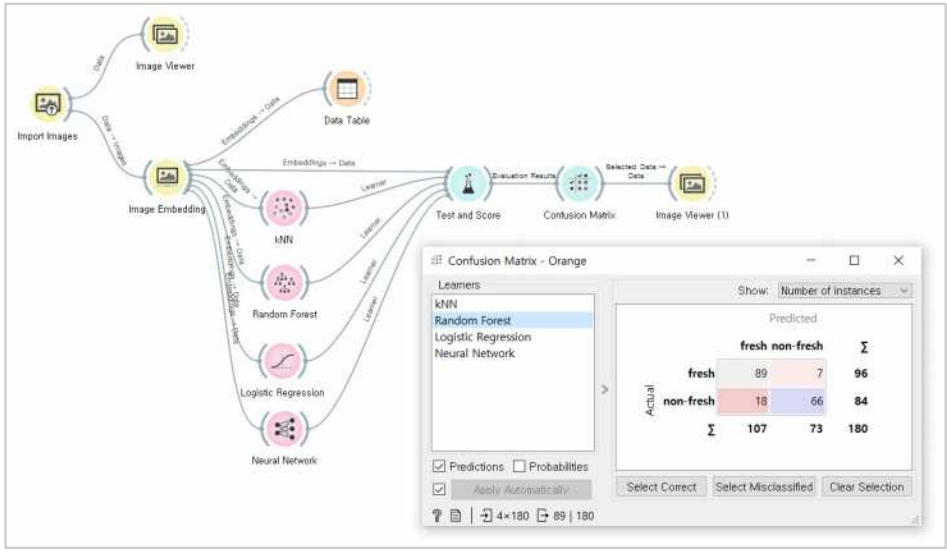
4개의 학습 모델의 성능을 평가한 결과 정확도 순으로 모델을 나열하면 Logistic Regression이 가장 높고 Neural Network, kNN, Random Forest로 나타났다. Logistic Regression의 경우 정확도가 0.994으로 나왔는데 1에 가까울수록 정확히 분류한다는 의미로 신선한 생선 눈 이미지와 오래된 생선 눈 이미지를 가장 정확하게 분류했다는 뜻이다. 성능을 평가하는 척도는 정확도 이외에 다양한 지표가 존재하는데 각 지표에 대한 설명은 다음과 같다.

[표 9-3] Evaluation Results 속성 설명

분류 성능 평가 척도	상세 설명
AUC(Area under ROC)	가능한 모든 분류 임계값에 대한 종합적인 성능 측정값
CA(Classification accuracy)	올바르게 분류된 예(TN, TP)의 비율
Precision(정밀도)	Positive(양성)로 분류된 인스턴스 중 참 양성(True Positive)의 비율
Recall(재현율)	데이터의 모든 Positive(양성) 사례 중 참 양성(True Positive)의 비율
F1	Precision(정밀도)와 Recall(재현율)의 가중 조화 평균
LogLoss	모델 예측과 목표 값 간의 교차 엔트로피. 이 범위는 0에서 무한대까지이며 값이 낮을수록 모델의 품질이 더 높음을 나타냄

혼동 행렬을 이용하여 성능 평가 척도를 계산하는 방법을 이해해 보기 위해 Random Forest의 혼동 행렬(confusion matrix)을 확인해 보자. Evaluate - confusion Matrix 위젯을 선택하고 Test and Score와 연결시킨다. 위젯을 더블클릭하여 왼쪽에서 알고리즘을 선택하고 혼동 행렬을 확인한다. 여기서 테스트 데이터의 수가 180인 이유는 30개의 데이터 중 30%에 해당하는 9개의 데이터를 20번 반복하여 테스트했기 때문이다.

Test and Score의 결과를 혼동행렬[Confusion Matrix]과 연결하여 정확도를 확인한다.



[그림 9-11] 학습과 평가

Random Forest 모델의 혼동 행렬을 살펴보면 그림과 같다. 이 중 CA(Classification Accuracy)을 계산해 보자. CA는 올바르게 분류한 비율에 해당한다. 혼동 행렬에서 실제 신선한 생선의 눈을 fresh로 예측한 경우를 TP(True Positive), 실제 오래된 생선의 눈을 fresh으로 예측한 경우 FN(False Negative), 실제 신선한 생선의 눈을 non-fresh로 예측한 경우 FP(False Positive), 실제 오래된 생선의 눈을 non-fresh로 예측한 경우 TN(True Negative)라고 한다.

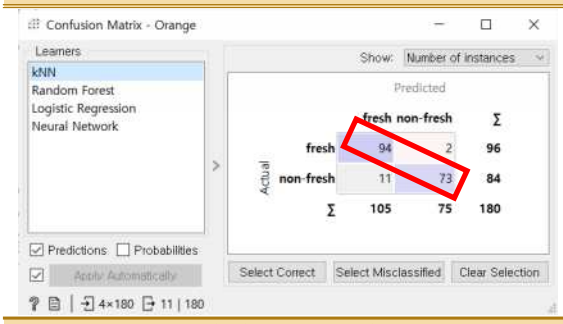
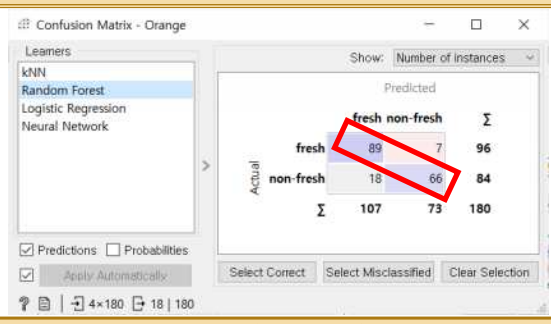
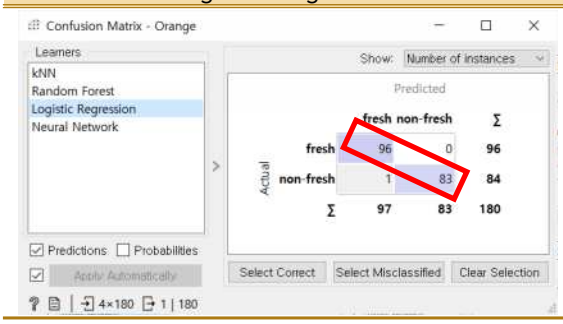
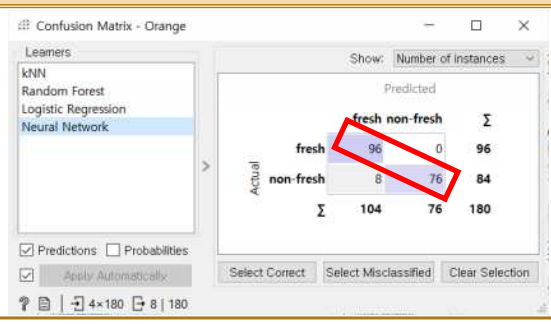
[표 9-4] 혼동 행렬 분석에 필요한 속성

예측 데이터 / 실제 데이터	신선한 눈(Positive)	오래된 눈(Negative)
신선한 눈(True)	TP	FN
오래된 눈(False)	FP	TN

$$CA = \frac{TP + TN}{TP + FN + FP + TN} = \frac{107 + 84}{107 + 18 + 7 + 84} = \frac{191}{216} = 0.88$$

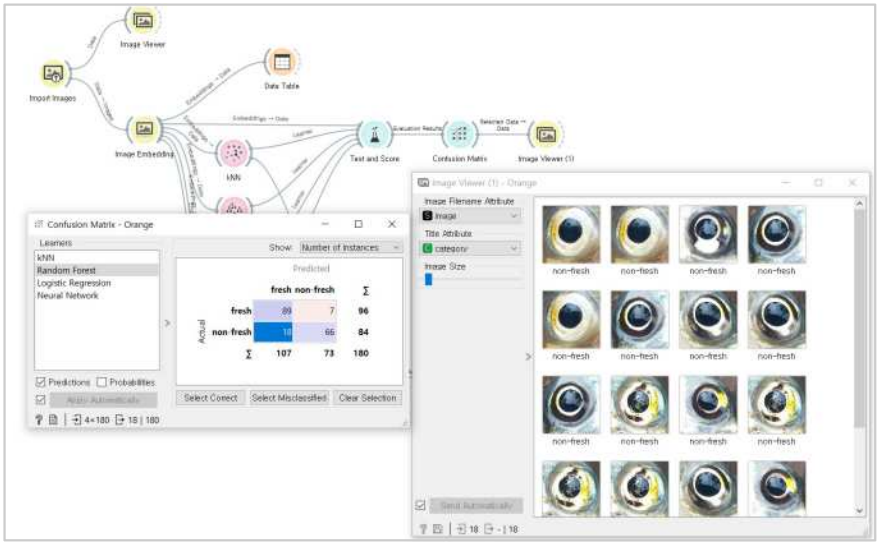
[그림 9-12] 혼동 행렬을 이용한 정확도 계산

[표 9-5] 인공지능 모듈 학습 결과 혼동행렬 확인

kNN	Random Foreset
	
Logistic Regression	**Neural Network**
	

4가지 혼동행렬을 비교해보아도 위와 같이 Logistic Regression이 우수한 것을 확인할 수 있다. 정확도가 가장 높았던 Logistic Regression의 성능평가척도는 위와 같은 방법으로 스스로 계산해보자.

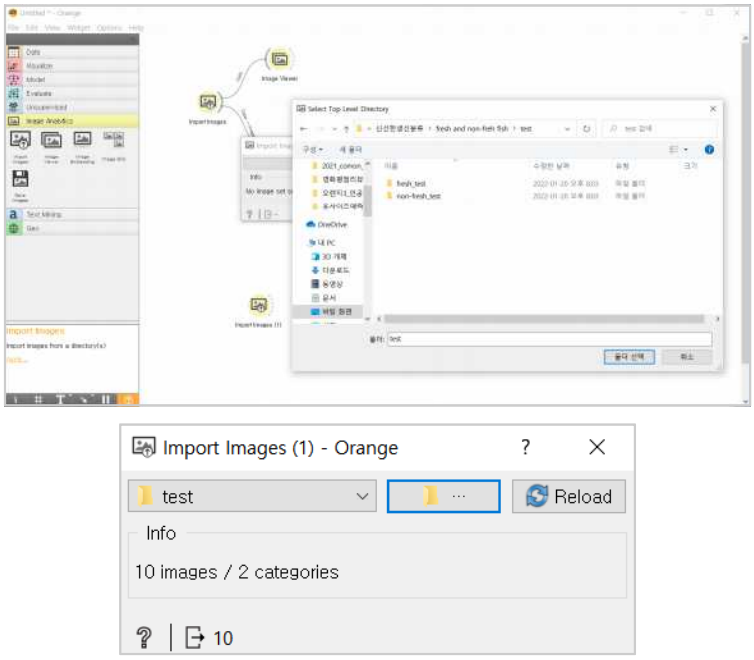
잘못 분류된 모델 확인을 위해 혼동행렬과 이미지뷰어 위젯을 연결시켜 혼동행렬에서 잘못 분류된 부분을 선택하면 어떤 이미지를 잘못 분류했는지 확인할 수 있다



[그림 9-13] confusion matrix - image viewer 확인 결과

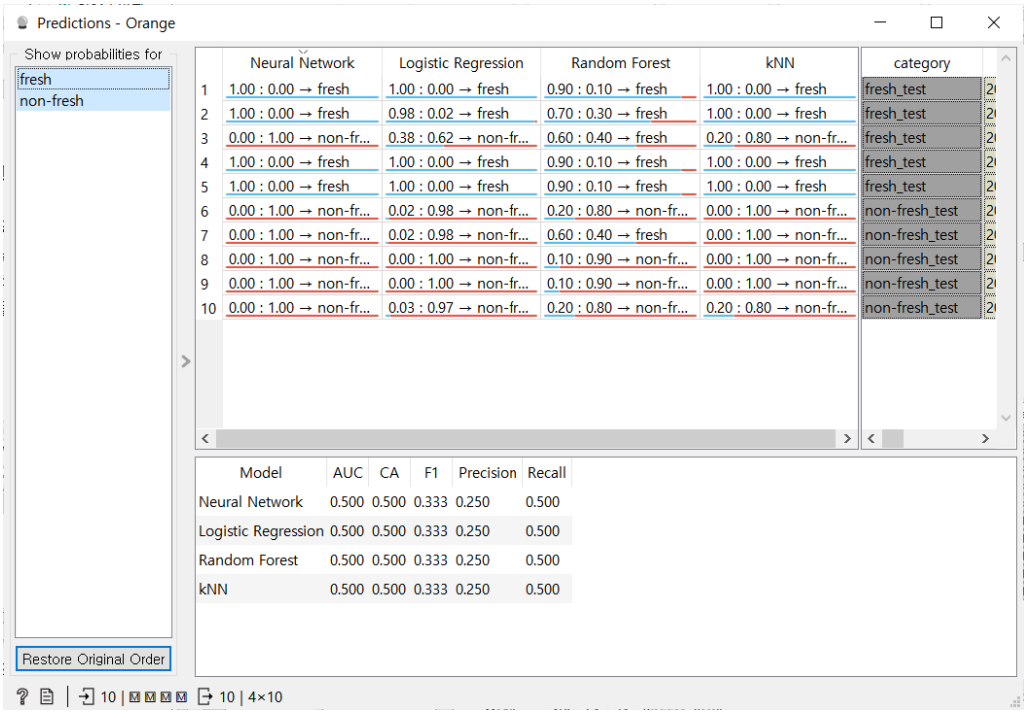
그렇다면 실제 테스트 데이터를 이용해 확인해 보자.

1) Image Analytics - Import Image 아이콘을 선택하여 캔버스에 배치한 후 dataset (test)를 로드한다.
2) Image Analytics - Image Embedding 위젯을 선택하여 Import Image 위젯에 연결한다.
3) Evaluate - Predictions 위젯을 선택해 캔버스에 배치한다.
4) Image Embedding과 기존 모델 생성에 사용했던 4가지 학습 알고리즘을 모두 Predictions에 연결한다.
5) Predictions를 클릭하여 결과를 확인한다.



[그림 9-14] 테스트 데이터를 이용한 예측

Predictions 위젯을 눌러 결과를 확인해 보자. 카테고리는 실제 값이고 각 모델별로 예측값이 표시되어 있다. 위에서 5개씩 분류해두었던 10개의 데이터를 이용해 테스트를 해 본 결과 10개의 사진을 모두 신선한 생선의 눈 사진이라고 판단하였고 정확도는 0.5가 나왔다. 이 모델의 성능을 평가할 때는 세 가지 모델이 90%가 넘는 정확도를 보인 것에 비하면 실제 테스트 결과는 좋지 않음을 알 수 있다. 이러한 결과가 나온 이유는 데이터의 양이 충분하지 못하여 성능이 낮게 측정된 것이다. 따라서 이 모델의 정확한 성능을 확인하기 위해서는 더 많은 데이터를 이용해 테스트를 진행해야 할 것이다.



[그림 9-15] 예측값 확인

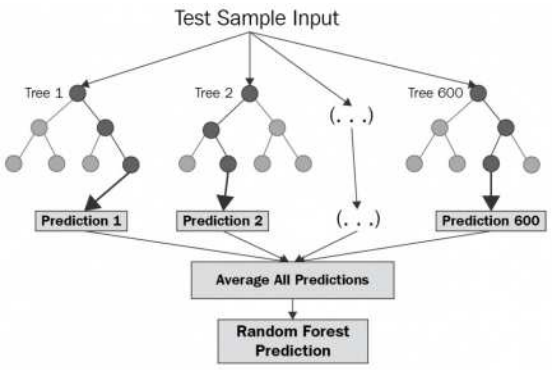
09. 생선 눈 이미지로 생선의 신선도를 구분해볼 수 있을까?

정리하기

생선의 신선도를 파악하기위해 생선의 눈 사진 데이터를 수집하여 훈련데이터와 테스트데이터로 분할하고 4가지 분류모델을 이용하여 훈련하였다. 훈련한 결과 로지스틱 회귀(Logistic Regression)모델의 성능이 1로 가장 좋았다. 나누어 둔 테스트 데이터를 이용하여 예측한 결과는 0.5로 성능이 좋지 않았지만 데이터의 양이 많았다면 올바른 결과를 예측했을 것이다.

AI 더 알아보기

◆ Random Forest



랜덤포레스트란 분류, 회귀 분석 등에 사용되는 앙상블 학습 방법의 일종으로, 훈련 과정에서 구성한 다수의 결정 트리로부터 부류(분류) 또는 평균 예측치(회귀 분석)를 출력함으로써 동작한다. 오버피팅을 피하기 위해 임의(random)의 숲을 구성하고 모든 의사결정 트리는 학습 데이터 세트에서 임의로 하위 데이터 세트를 추출하여 생성된다. 중복을 허용하기 때문에 단일 데이터가 여러번 선택될 수도 있다. 이 과정을 배깅(bagging)이라고 한다. 나무를 만들 때는 모든 속성(feature)들에서 임의로 일부를 선택하고 그 중 정보 획득량이 가장 높은 것을 기준으로 데이터를 분할한다. 만약 데이터 세트에 n개의 속성이 있는 경우 n제곱근 개수만큼 무작위로 선택하는 것이 일반적이다.

[참고 문헌]

1. 손원성 외 3인(2021). 오렌지3로 알아가는 머신러닝 데이터 분석. 홍릉.
2. 서울과학종합대학원 디지털혁신처(2021). 3시간 만에 배우는 인공지능 데이터분석. 오렌지. 서울경제경영.
3. 오렌지. https://orangedatamining.com/widget-catalog/
4. 데이터 수집(kaggle). https://www.kaggle.com/nabeelsajid917/covid-19-x-ray-10000-images
5. 이미지 학습 방법 설명. https://orangedatamining.com/widget-catalog/image-analytics/imageembedding/
6. 랜덤포레스트 개념 및 특징. https://hleecaster.com/ml-random-forest-concept/
7. 랜덤포레스트 그림. https://itwiki.kr/w/%EB%9E%9C%EB%8D%A4_%ED%8F%AC%EB%A0%88%EC%8A%A4%ED%8A%B8

MEMO

10. 우리나라 80%가 심장병 등의 만성질환으로 사망한다?

구미여자고등학교 교사 조 예 린

학습 진행 과정

1단계	데이터 준비	- 사용 데이터: 심부전증 dataset - 수집: 캐글
2단계	데이터 불러오기	- 학습 데이터 불러오기 - 데이터의 속성별 Role(역할) 설정하기
3단계	데이터 시각화	- 시각화 도구을 이용한 데이터 탐색 - 사용한 시각화 도구: Violin Plot
4단계	속성 추출	- 데이터 시각화 결과
5단계	모델 학습	- 분류를 이용한 모델 학습 - 사용된 알고리즘:Logistic Regression, kNN, SVM
6단계	성능 평가	- test and score를 이용한 성능 평가 - 혼동 행렬을 이용한 성능 평가

학습 내용 요약

데이터 종류	문제 유형	사용된 학습 알고리즘	성능 평가 도구
정형 데이터	분류	Logistic Regression kNN SVM	test and score

문제 상황

우리는 심정지 증상을 보이는 사람을 심폐 소생술로 살린다는 뉴스를 자주 접할 수 있다. 우리나라 인구의 10명 중 8명은 암, 심장병 등의 만성질환으로 사망한다고 한다.
이에 따라 연구원 및 의료진들은 심장병에 관한 연구를 끊임없이 진행하고 있는데, 우리도 함께 이 연구에 동참해보고자 한다.
심부전증은 심장질환의 마지막 단계에서 나타나는 질환으로, 심장의 기능이 쇠약해져서 혈액의 공급이 불안정해지는 병이다. 심부전증은 어떤 전조 증상을 보일 때 발생하는지 미리 알려주는 프로그램이 있다고 하면 안타깝게 심부전증으로 사망하는 사람을 살릴 수 있지 않을까? 심부전증이 발생하기 전의 여러 가지 증상을 분석하여 심부전증 발생 판별 인공지능 모델을 만들어 보자.

이제 심부전증이 발생하기 전의 증상들을 가지고 심부전증 발생 판별 인공지능 모델을 만들어 보자.

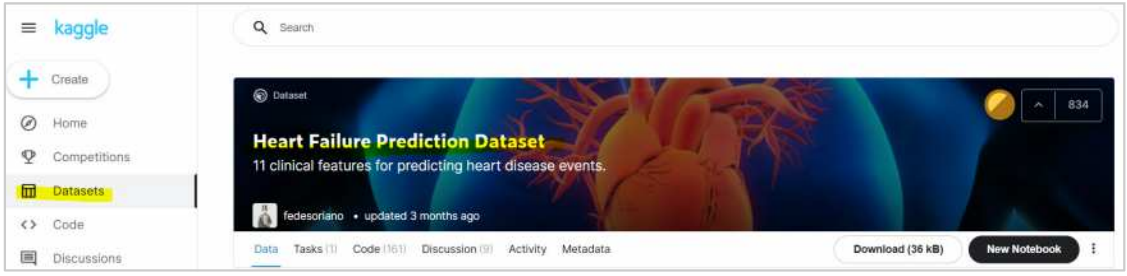
01 데이터를 준비하자

1 심부전증 데이터세트

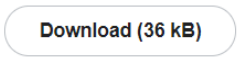
① 캐글(kaggle) 홈페이지 접속

- https://www.kaggle.com/

② dataset 클릭 〉 Heart Failure Prediction Dataset 검색



③ 심부전증 데이터세트(Heart Dataset) 다운로드



④ 심부전증 데이터세트(Heart Dataset) 속성 확인

심부전증 데이터세트를 다운로드 받아 파일을 열어보면 [그림 10-1]과 같다.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
2	40	M	ATA	140	289	0	Normal	172	N	0	Up	0
3	49	F	NAP	160	180	0	Normal	156	N	1	Flat	1
4	37	M	ATA	130	283	0	ST	98	N	0	Up	0
5	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
6	54	M	NAP	150	195	0	Normal	122	N	0	Up	0
7	39	M	NAP	120	339	0	Normal	170	N	0	Up	0
8	45	F	ATA	130	237	0	Normal	170	N	0	Up	0
9	54	M	ATA	110	208	0	Normal	142	N	0	Up	0
10	37	M	ASY	140	207	0	Normal	130	Y	1.5	Flat	1
11	48	F	ATA	120	284	0	Normal	120	N	0	Up	0
12	37	F	NAP	130	211	0	Normal	142	N	0	Up	0
13	58	M	ATA	136	164	0	ST	99	Y	2	Flat	1

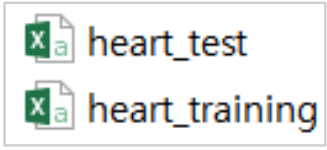
[그림 10-1] 심부전증 데이터 세트

다운로드 받은 심부전증 데이터 세트의 속성을 살펴보면 다음과 같다. 다음 설명을 보고 심부전증이 발생하기 전 증상을 살펴보자. 심장병 분류를 위해 Heart Disease 속성이 0, 1로 저장되어 있다.

속성	설명	비고
Age	나이	
Sex	성별	
ChestPainType	가슴 통증 유형	ATA-협심증 NAP-비협심증 흉통
RestingBP	안정된 혈압의 수치	정상수치 120 mmHg 미만
Cholesterol	콜레스테롤 수치	정상수치 200mg/dL 미만
FastingBS	과호흡	0-정상호흡 1-과호흡 발생
RestingECG	안정된 심전도 수치	Nomal-정상 ST-심전도의 QRS파 끝에서 T파 시작까지의 부분, 심실 전체의 흥분
MaxHR	최고 심박수	심박수의 최고 수치
ExerciseAngina	운동	Y-꾸준한 운동을 함 N-꾸준한 운동을 하지 않음
Oldpeak	비교적 안정되기까지 운동으로 유발되는 ST depression(부분 하강)	ST depression(부분 하강) 수치
ST_Slope	언어 장애 증상	Up-증상 있음 Flat-증상 없음
Heart Disease	심부전증 발생	0-심부전증 발생하지 않음 1-심부전증 발생함

2 기계학습을 위한 데이터 준비

훈련 데이터와 테스트 데이터를 만든다. 훈련 데이터는 지도 학습을 위한 데이터이고, 테스트 데이터는 우리가 만들 모델의 정확성을 파악하기 위한 데이터이다. 다운로드 받은 데이터에서 [그림 10-2]와 같이 훈련 데이터와 테스트 데이터를 분류해보자.



[그림 10-2] 훈련 데이터와 테스트 데이터

먼저 다운로드 받은 데이터세트 파일을 복사하여 2개의 파일로 만든다. 한 개의 파일은 훈련 데이터 파일이 될 것이고, 나머지 한 개의 파일은 테스트 데이터 파일이 될 것 이다.

① 훈련 데이터

다운로드한 파일을 열어 920개의 행 중에서 아래의 149개의 행을 삭제한다. 남은 769개의 행의 데이터는 훈련 데이터가 된다. [그림 10-3]처럼 1행의 속성을 포함한 770행까지의 데이터를 남겨두고 저장한다. 파일의 구분을 위해 파일 이름은 'heart_training'으로 변경한다.

	A	B	C	D	E	F	G	H	I	J	K	L
757	54	M	NAP	150	232	0	LVH	165	N	1.6	Up	0
758	46	F	NAP	142	177	0	LVH	160	Y	1.4	Down	0
759	67	F	NAP	152	277	0	Normal	172	N	0	Up	0
760	56	M	ASY	125	249	1	LVH	144	Y	1.2	Flat	1
761	34	F	ATA	118	210	0	Normal	192	N	0.7	Up	0
762	57	M	ASY	132	207	0	Normal	168	Y	0	Up	0
763	64	M	ASY	145	212	0	LVH	132	N	2	Flat	1
764	59	M	ASY	138	271	0	LVH	182	N	0	Up	0
765	50	M	NAP	140	233	0	Normal	163	N	0.6	Flat	1
766	51	M	TA	125	213	0	LVH	125	Y	1.4	Up	0
767	54	M	ATA	192	283	0	LVH	195	N	0	Up	1
768	53	M	ASY	123	282	0	Normal	95	Y	2	Flat	1
769	52	M	ASY	112	230	0	Normal	160	N	0	Up	1
770	40	M	ASY	110	167	0	LVH	114	Y	2	Flat	1

[그림 10-3] 훈련 데이터

② 테스트 데이터

저장한 또다른 파일을 열어 아래에서 149개의 행만 남겨두고 삭제한다. 남겨진 데이터는 테스트 데이터가 된다. [그림 10-4]처럼 속성을 포함한 150행까지의 데이터를 남기고 저장한다. 파일의 구분을 위해 파일 이름은 'heart_test'로 변경한다.

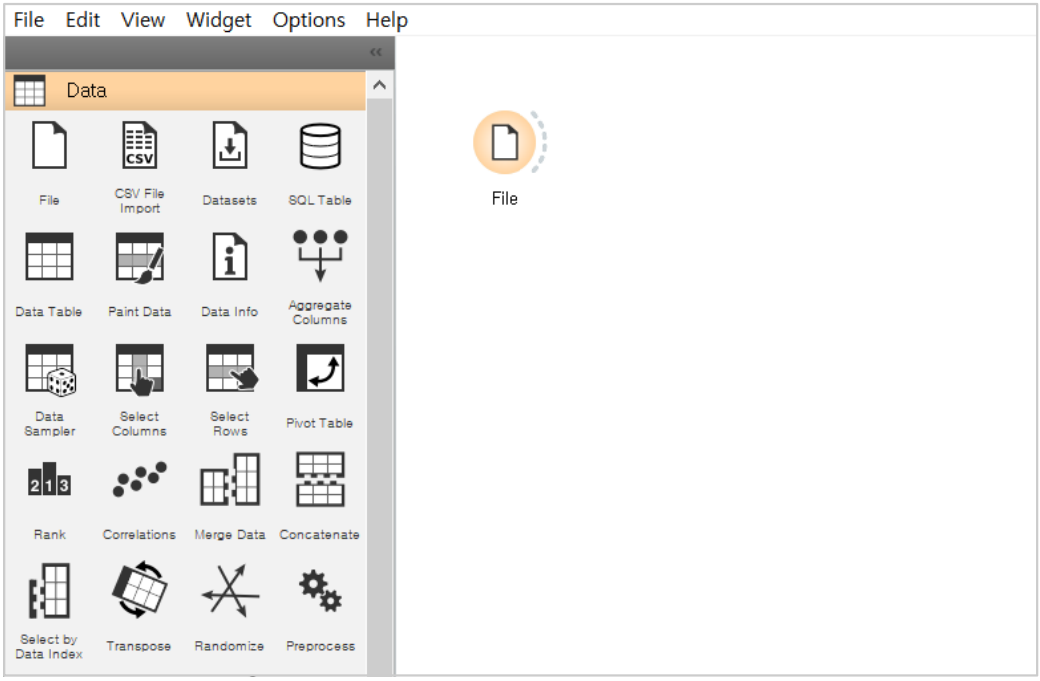
	A	B	C	D	E	F	G	H	I	J	K	L
136	56	M	ATA	130	221	0	LVH	163	N	0	Up	0
137	56	M	ATA	120	240	0	Normal	169	N	0	Down	0
138	67	M	NAP	152	212	0	LVH	150	N	0.8	Flat	1
139	55	F	ATA	132	342	0	Normal	166	N	1.2	Up	0
140	44	M	ASY	120	169	0	Normal	144	Y	2.8	Down	1
141	63	M	ASY	140	187	0	LVH	144	Y	4	Up	1
142	63	F	ASY	124	197	0	Normal	136	Y	0	Flat	1
143	41	M	ATA	120	157	0	Normal	182	N	0	Up	0
144	59	M	ASY	164	176	1	LVH	90	N	1	Flat	1
145	57	F	ASY	140	241	0	Normal	123	Y	0.2	Flat	1
146	45	M	TA	110	264	0	Normal	132	N	1.2	Flat	1
147	68	M	ASY	144	193	1	Normal	141	N	3.4	Flat	1
148	57	M	ASY	130	131	0	Normal	115	Y	1.2	Flat	1
149	57	F	ATA	130	236	0	LVH	174	N	0	Flat	1
150	38	M	NAP	138	175	0	Normal	173	N	0	Up	0

[그림 10-4] 테스트 데이터

02 데이터를 불러오자

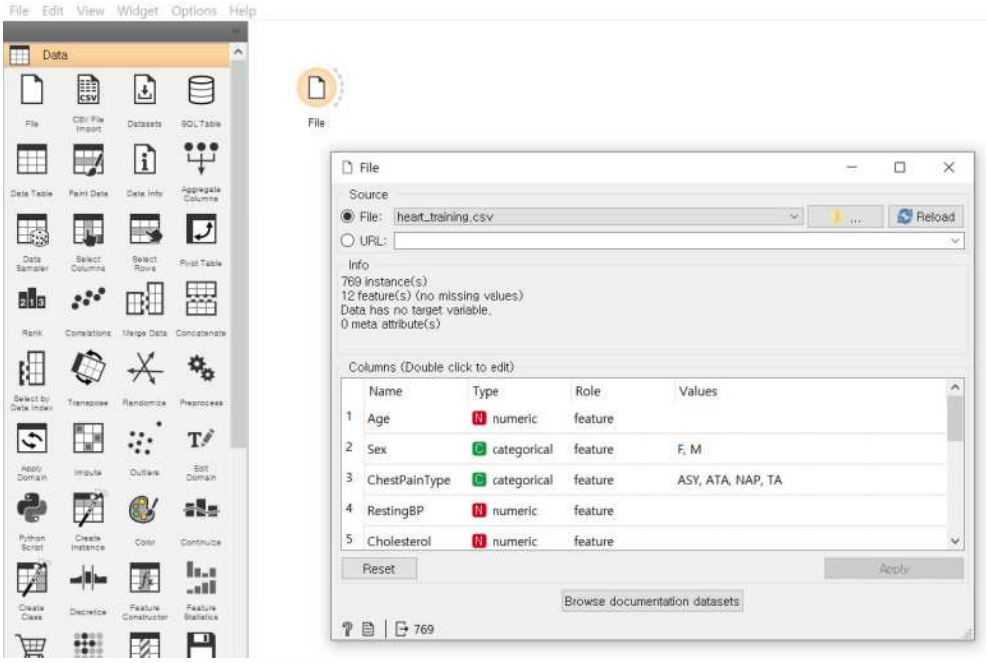
1 오렌지에 학습 데이터 불러오기

먼저 파일 업로드를 위해 왼쪽 위젯 도구상자에 [Data]에서 [File]을 선택하여 활동 창에 올려놓는다.

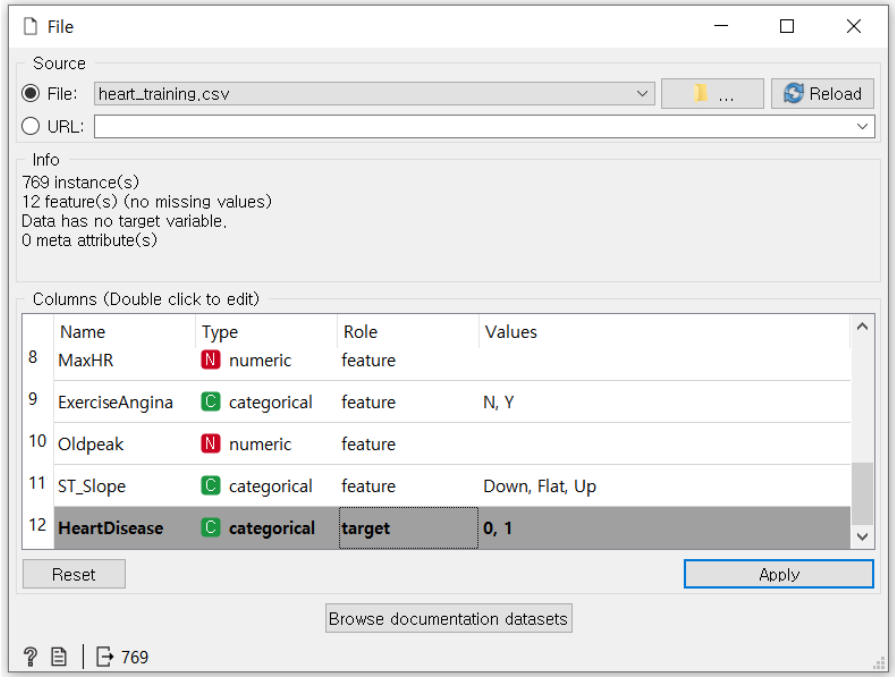


① 학습 데이터 불러오기

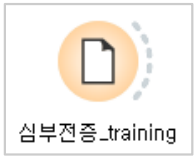
[File]을 클릭하여 다운로드한 'heart_training' 파일을 불러온다.



심부전증 발생을 알아보기 위해 Role에서 heart 속성을 target으로 변경한 후 Apply를 눌러 적용한다.



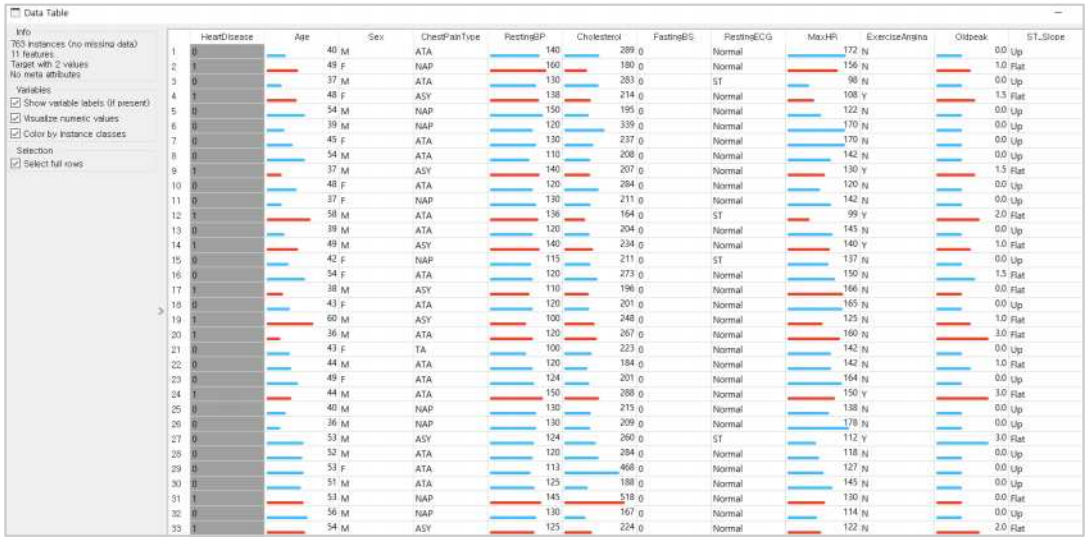
테스트 데이터 파일과 구분하기 위해 [File]위젯 위에서 오른쪽 마우스를 클릭하여 rename(파일 이름 변경)을 한다. 파일 이름은 '심부전증_training'으로 한다.



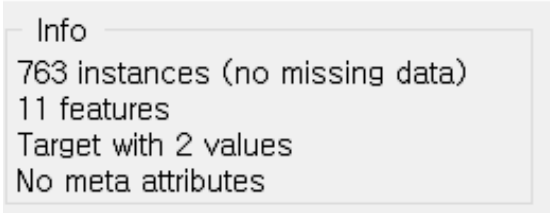
② 데이터 테이블 확인하기

데이터를 확인하기 위해 [Data]위젯에서 [Data table]위젯을 끌어내어 [File]위젯과 연결한다.

Data Table을 더블 클릭하면 심부전증_training 데이터를 확인할 수 있다. [그림 10-5]와 같이 이름과 크기, 가로 세로 길이 등을 테이블 형태로 보여준다. 정보(Info)를 살펴보면 2개의 값을 지닌 종속변수(Target)가 있으며, 기계학습을 위한 독립변수(features)는 11개이고, meta 속성 데이터는 없다.



[그림 10-5] 데이터 테이블 확인



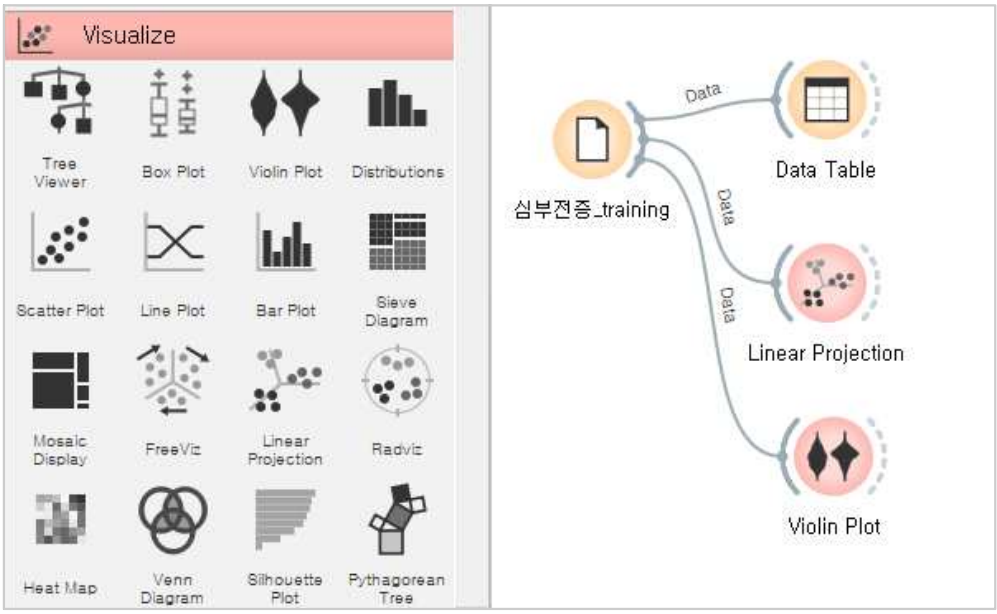
[그림 10-6] 데이터 테이블 속성 확인

03 데이터를 탐색하자

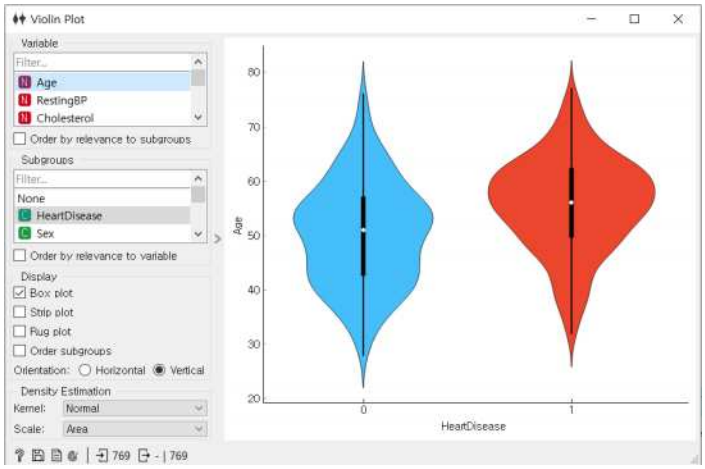
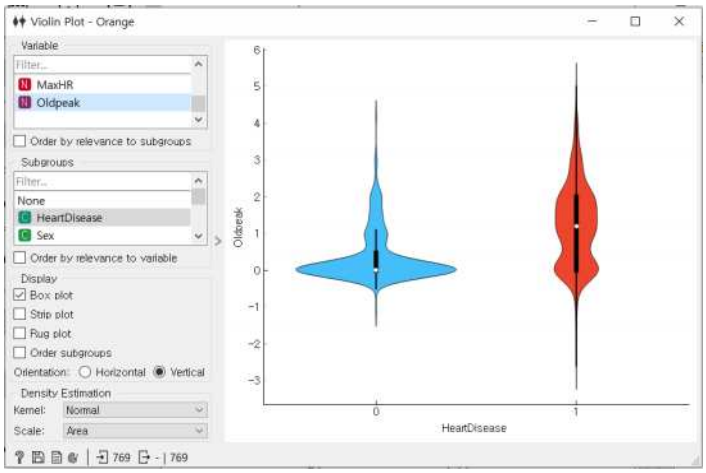
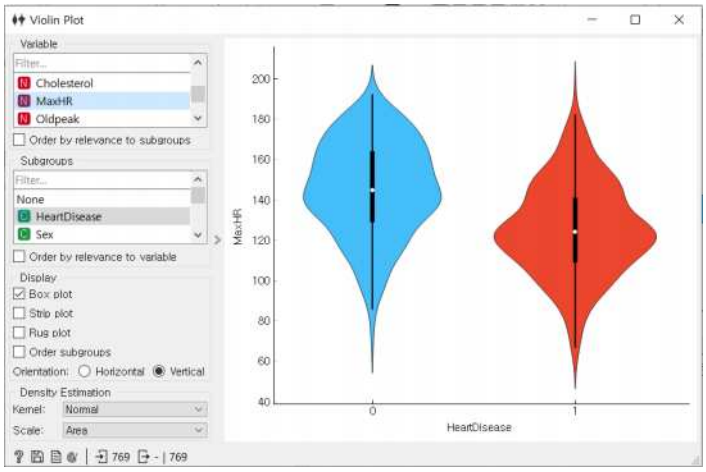
1 이미지 특징 시각화

① [Violin Plot] 위젯을 이용하여 시각화해보자.

[Violin Plot] 위젯은 2개의 속성의 데이터를 시각화해 준다. [Visualize] 위젯에서 [Violin Plot]]을 클릭하여 [File]위젯과 연결하여 데이터를 확인해보자.



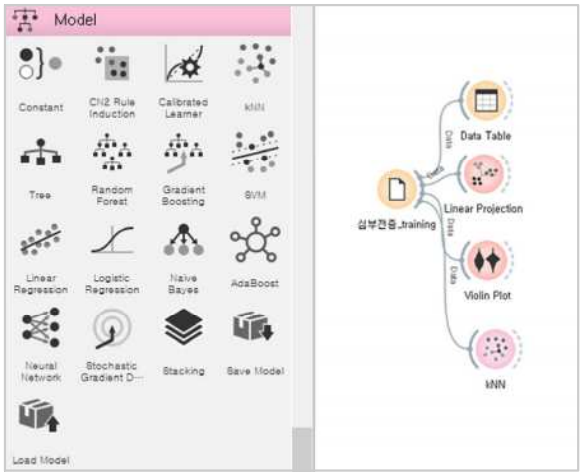
왼쪽 탭의 Vailable에서 여러 가지 속성을 클릭하여 해당 속성과 심부전증 발생 연관성을 확인할 수 있다.

2가지 속성을 이용한 산점도	데이터 해석
 Age와의 연관성	Age와 심부전증 발생의 연관성을 [Violin Plot]을 통해 시각화하여 보았을 때 Age가 심부전증 발생에 영향을 미칠 수 있다는 것을 알 수 있음.
 Oldpeak과의 연관성	Oldpeak수치와 심부전증 발생의 연관성을 [Violin Plot]을 통해 시각화하여 보았을 때 Oldpeak 수치는 심부전증 발생에 영향을 미친다는 것을 알 수 있음.
 MaxHR과의 연관성	MaxHR 수치와 심부전증 발생의 연관성을 [Violin Plot]을 통해 시각화하여 보았을 때 MaxHR 수치는 심부전증 발생에 영향을 미칠 수 있다는 것을 알 수 있음.

04 모델 학습하고 성능 평가하자

1 학습 모델 선택하고 학습시키기

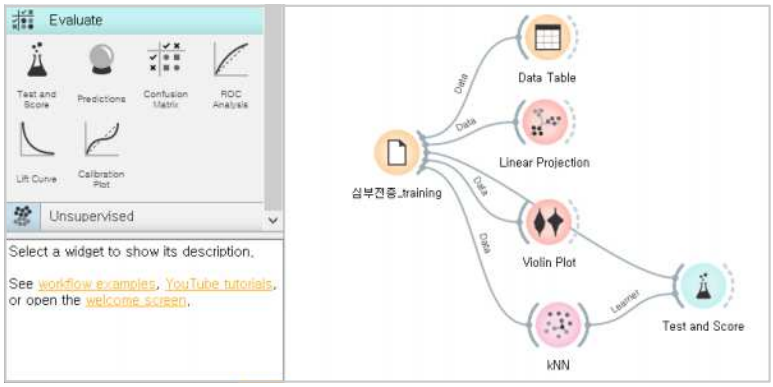
데이터와 기계학습 알고리즘을 연결하면 데이터를 분류할 수 있는 기계학습 모델을 만들 수있다. 모델 학습에 필요한 것은 데이터와 기계학습 알고리즘이다. 데이터와 기계학습 알고리즘 중 [kNN] 위젯과 [File] 위젯을 연결하여 모델 학습시킨다. [kNN] 위젯은 [Model] 위젯 하위 폴더에 있다.



모델을 연결한 후 얼마나 정확히 분류하는지 평가하기 위해 [Test ans Score] 위젯을 연결한다. [Test and Score]는 [Evaluate] 위젯에서 찾아볼 수 있다. [Test ans Score] 위젯과 [kNN] 위젯, [File] 위젯을 각각 연결하여 모델을 평가해보자.

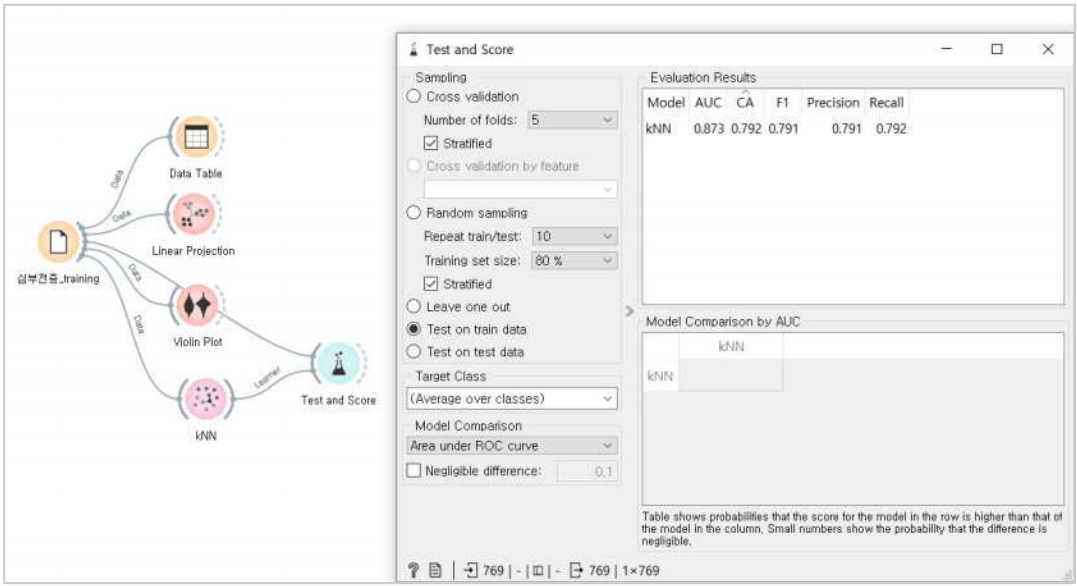
2 성능 평가하기

① 성능 좋은 모델을 결정하자



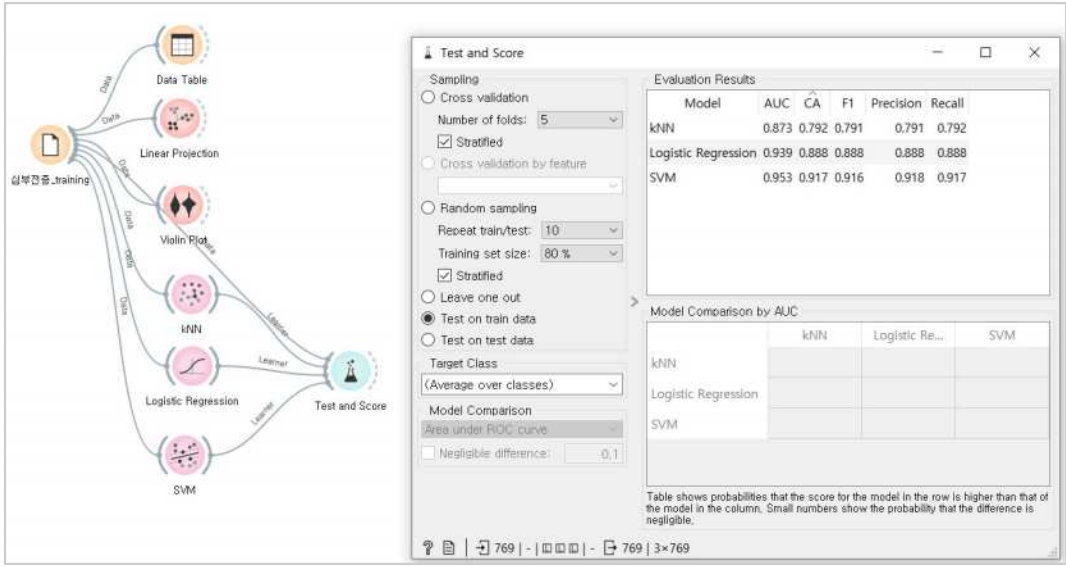
[그림10-7] 성능 평가

[Test and Score] 위젯에서 다양한 성능 평가 방법을 정할 수 있다. folds의 수를 10으로 설정하여 성능 평가였더니 재현율(Recall)이 0.792으로 나타났다.



오렌지3에서는 여러 알고리즘을 동시에 연결하여 어느 모델이 성능이 좋은지 비교해볼 수 있는 장점이 있다. [Model] 위젯에서 [Logistic Regression] 위젯과 [SVM] 위젯을 추가로 연결해서 여러 가지 모델의 성능을 비교해보자.

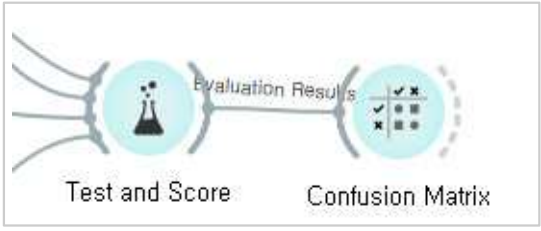
분류 성능 평가 척도 중 데이터의 모든 Positive(양성) 사례 중 참인 양성(True Positive)의 비율을 나타내는 Recall(재현율)을 기준으로 모델의 성능을 비교해보자.



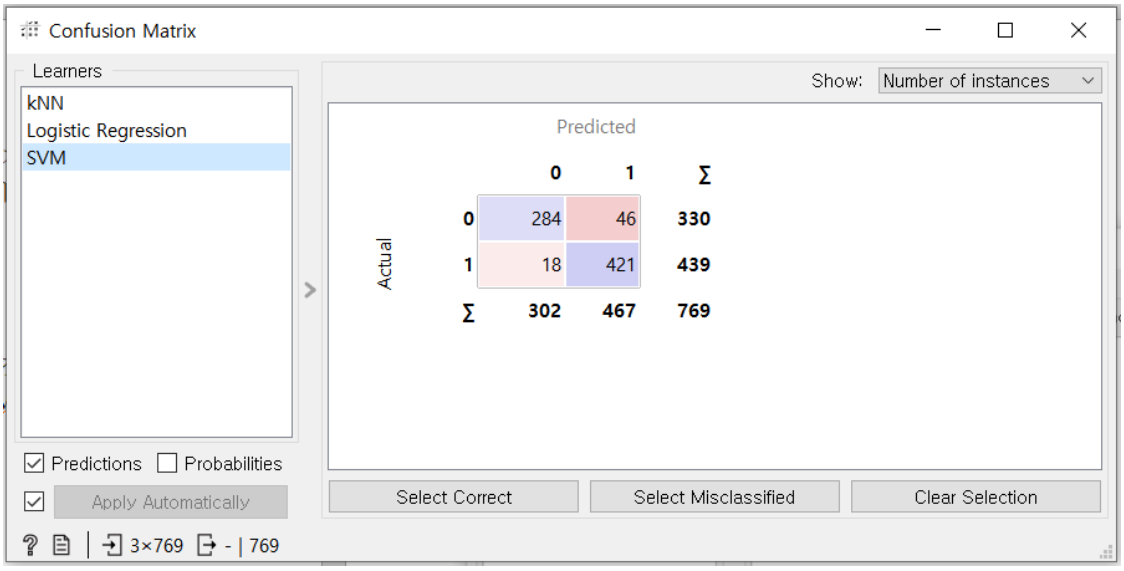
Logistic Regression, SVM모델을 학습시킨 후 Test and Score 위젯을 통해 성능을 확인해보자. Logistic Regression모델의 재현율(Recall)은 0.888, SVM모델의 재현율(Recall)은 0.917로 SVM 모델의 성능이 가장 우수한 것으로 나타났다. 따라서 심부전증 발생을 분류하는 문제는 SVM 알고리즘을 이용하여 모델을 만드는 것이 적합하다고 말할 수 있다.

② 어떤 것을 잘못 분류했나?

[Test and Score] 위젯으로 성능 평가한 결과를 [Confusion Matrix] 위젯에 연결하여 실제 데이터를 어떻게 예측하였는지 살펴보자. [Confusion Matrix] 위젯은 [Evaluate] 위젯에서 찾을 수 있다.



[Confusion Matrix] 위젯을 더블 클릭하여 성능 평가 결과를 확인해보자.



[그림 10-8] 훈련 데이터 성능 평가 결과

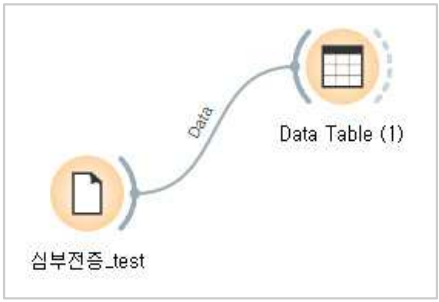
[그림 10-8]과 같이 심부전증이 발생하지 않는 데이터 330개 중 284개를 예측하였고, 심부전증이 발생하는 상황은 439개 중 421개를 예측하였다.

3 테스트 데이터로 예측하기

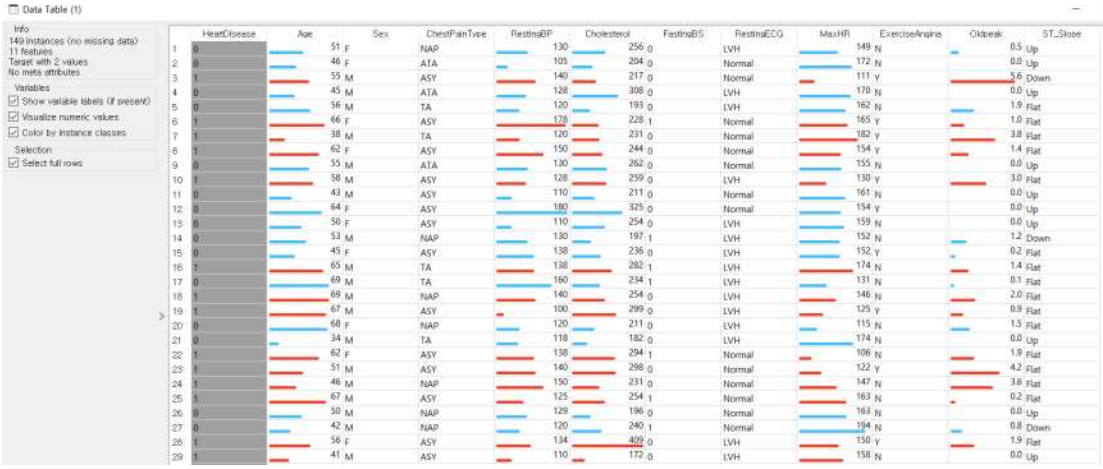
이제 훈련에 사용하지 않은 테스트 데이터로 심부전증 발생 확률을 얼마나 잘 예측할 수 있는지 확인해 보자.

① 테스트 데이터 불러오기

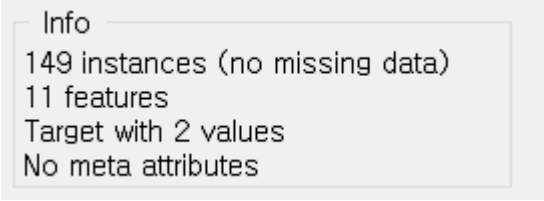
테스트 데이터도 훈련 데이터와 같은 방법으로 [Data]에서 [File]을 선택하여 'heart_test' 파일을 불러온다.



데이터 테이블을 확인하기 위해 [Data Table] 위젯을 추가하여 [File]과 연결한다. 데이터 테이블의 속성도 확인할 수 있다.



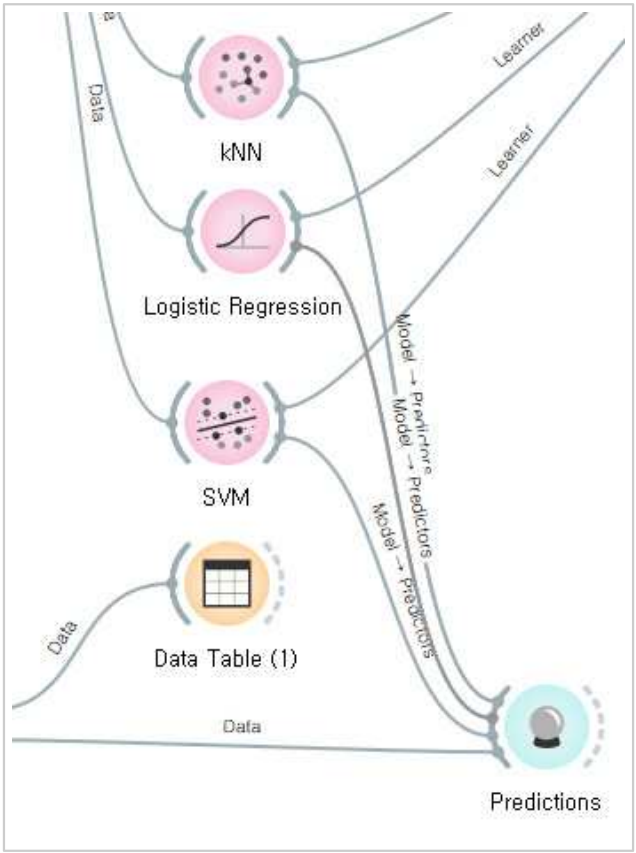
[그림 10-9] 데이터 테이블 확인



[그림 10-10] 데이터 테이블 속성 확인

② 테스트 데이터로 예측하기

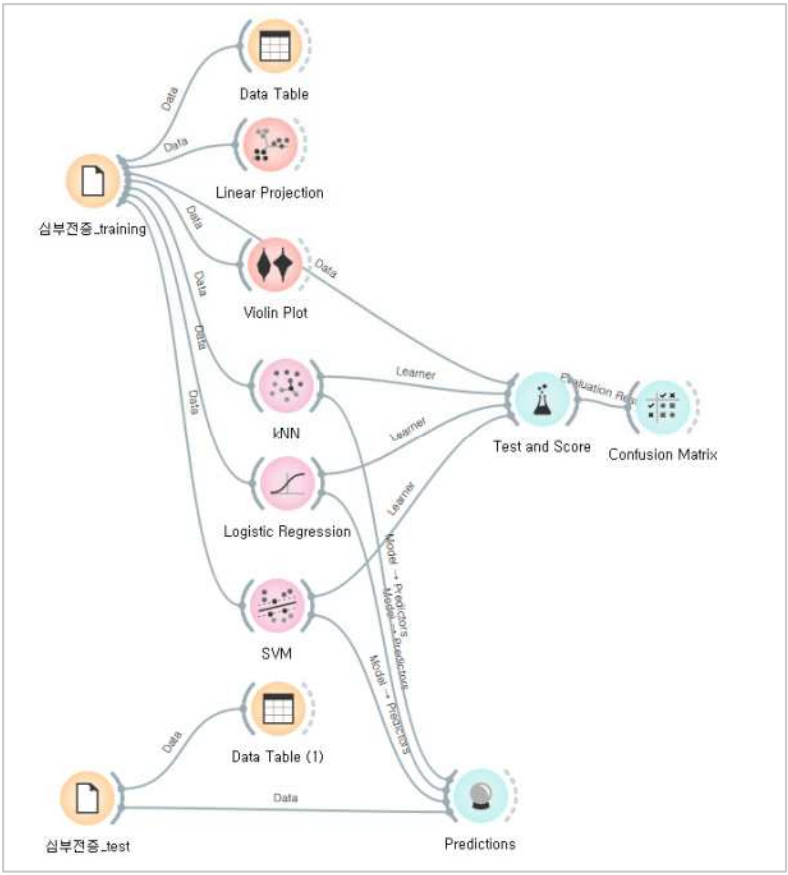
앞서 학습시켰던 세 가지 모델들과 테스트 데이터를 [Predictions] 위젯에 연결한다. 테스트 데이터를 얼마나 잘 예측하였는지 [Predictions]의 예측 결과를 통해 확인할 수 있다. [Predictions] 위젯은 [Evaluate] 위젯에서 찾을 수 있다.



[그림 10-11] [Predictions] 위젯 연결

[Predictions] 위젯을 클릭하여 재현율(Recall)을 확인해보자. Logistic Regression은 0.758, SVM은 0.745, kNN은 0.624로 확인할 수 있다.

Model	AUC	CA	F1	Precision	Recall
Logistic Regression	0.868	0.758	0.759	0.759	0.758
SVM	0.865	0.745	0.745	0.754	0.745
kNN	0.692	0.624	0.617	0.624	0.624



[그림 10-12] 심부전증 발생 판별 분류 모델 전체 과정

[참고 문헌]

1. 서울과학종합대학원 디지털혁신처(2021). 3시간 만에 배우는 인공지능 데이터분석, 오렌지. 서울경제경영.
2. 손원성 외 3인(2021). 오렌지3로 알아가는 머신러닝 데이터 분석. 홍릉.
3. 이고잉 외 2인(2021). 생활코딩 머신러닝. 위키북스.
4. 심부전증 데이터. Kaggle. https://www.kaggle.com/fedesoriano/heart-failure-prediction
5. 오렌지. https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/data/rank.html

MEMO

11. 바다 동물을 알아맞혀보자!

구미여자고등학교 교사 조 예 린

학습 진행 과정

1단계	데이터 준비	- 사용 데이터: 바다동물 데이터 세트 - 수집: 캐글 - 데이터 편집: 바다표범, 고래, 상어 이미지 데이터 추출
2단계	데이터 불러오기	- 학습 데이터 불러오기 - 데이터의 속성별 Role(역할) 설정하기
3단계	데이터 시각화	- 시각화 도구을 이용한 데이터 탐색 - 사용한 시각화 도구: Linear Projection
4단계	속성 추출	- 데이터 시각화 결과
5단계	모델 학습	- 분류를 이용한 모델 학습 - 사용된 알고리즘: Logistic Regression, k-NN, SVM, Tree, Random Forest
6단계	성능 평가	- test and score를 이용한 성능 평가 - 혼동 행렬을 이용한 성능 평가

학습 내용 요약

데이터 종류	문제 유형	사용된 학습 알고리즘	성능 평가 도구
정형 데이터	분류	Logistic Regression k-NN SVM Tree Random Forest	test and score

문제 상황

인터넷에서 강아지와 쿠키, 강아지와 식빵을 잘못 구분한 인공지능 모델이 화제가 되었다. 이와 같이 이미지는 분류하는 모델을 우리도 한번 만들어 보자! 바다 동물 중 생김새가 비슷하여 구분하기 어려워보이는 상어와 고래, 생김새가 조금 달라보이는 바다표범까지 과연 우리가 만든 인공지능 분류 모델은 이 셋을 잘 분류할 수 있을 것인가?

이제 바다표범, 고래, 상어를 분류하는 인공지능 모델을 만들어보자.

01 데이터를 준비하자

1 바다 동물 데이터세트(animal Dataset)

① 캐글(kaggle) 홈페이지 접속

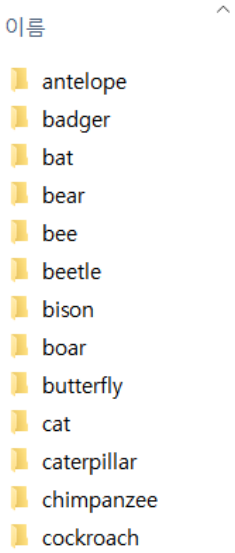
- https://www.kaggle.com/

② dataset 클릭 〉 Animal Image Dataset 검색

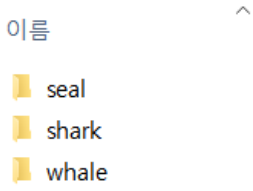
③ 동물 데이터세트(Heart Dataset) 다운로드

Download (36 kB)

④ 다운로드 받은 animal 폴더에서 seal, shark, whale 폴더만 남기고 삭제한다. 세 가지 바다 동물 이미지를 사용하여 바다표범, 상어, 고래를 분류하는 인공지능 모델을 만든다.

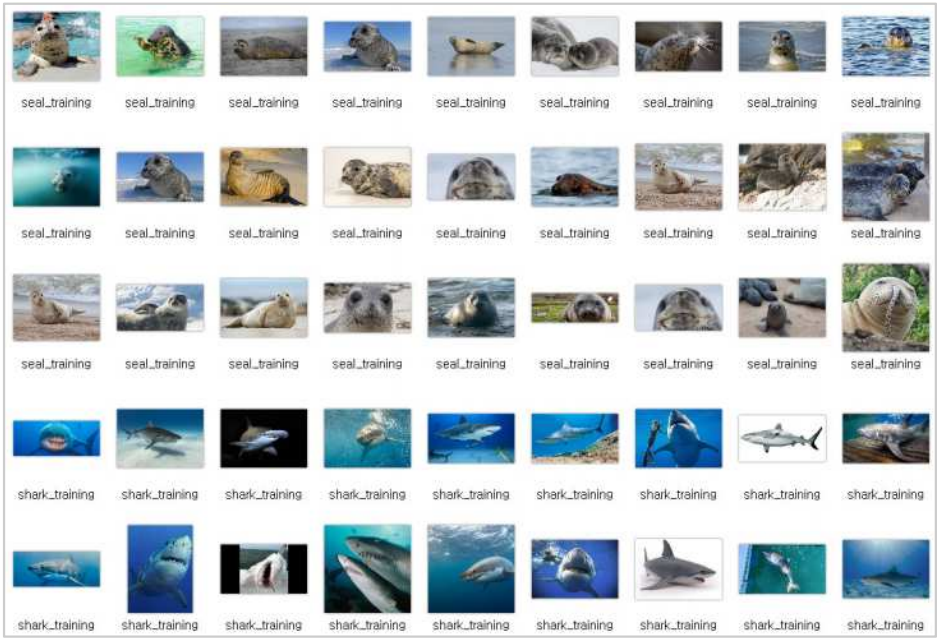


[그림 11-1] 다운로드한 animal 폴더의 하위 폴더



[그림 11-2] seal, shark, whale 폴더

바다 동물 데이터세트(animal Dataset)는 180개의 바다표범, 고래, 상어 이미지 데이터로 구성되어 있다. 180개의 이미지에 다음과 같이 파일 이름을 붙이고 전체 이미지 데이터를 Animal Data라고 하겠다.



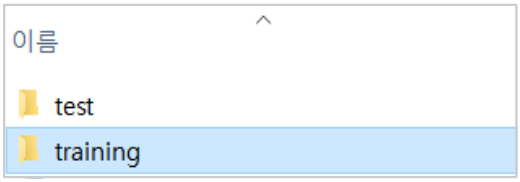
[그림 11-3] 바다 동물 데이터 세트

2 기계학습을 위한 데이터 준비

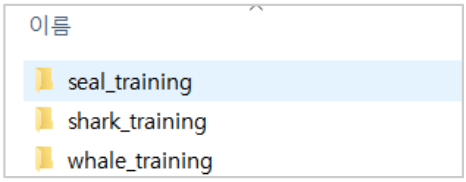
지도 학습을 위해서는 데이터에 정답 레이블을 붙여야 한다. 오렌지3 프로그램으로 기계학습을 하기 위해 Animal Dataset 폴더를 만든다.

① 훈련 데이터

아래 그림과 같이 다시 하위 폴더를 만들고 바다표범, 고래, 상어 이미지 파일을 추가하여 레이블을 붙이도록 한다. 아래 그림과 같이 training 이름의 폴더를 만들고, 하위 폴더에 seal_training, shark_training, whale_training 클래스 폴더를 만든다. 각각의 폴더에 훈련 데이터 이미지를 넣는다.

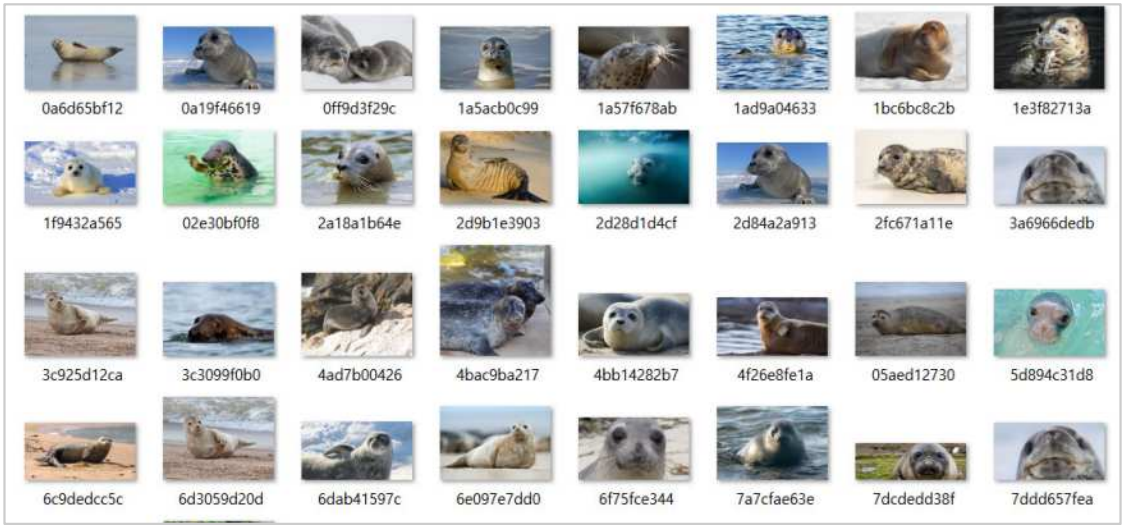


[그림 11-4] test 폴더와 training 폴더



[그림 11-5] 훈련 데이터 하위 폴더

훈련 데이터의 seal_training 폴더에는 [그림 11-6]과 같이 바다표범의 이미지 36장이 들어가도록 한다.

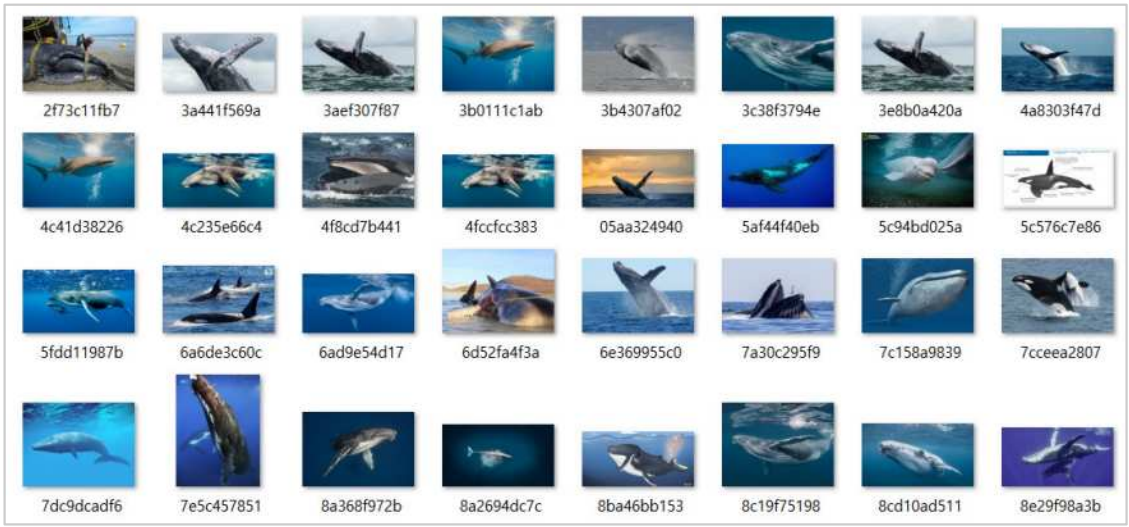


[그림 11-6] 바다표범 훈련 데이터

훈련 데이터의 shark_training 폴더에는 [그림 11-7]과 같이 상어의 이미지 36장이 들어가도록 한다.

[그림 11-7] 상어 훈련 데이터

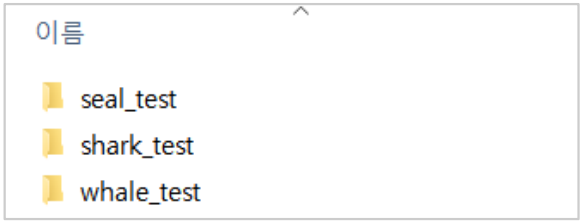
훈련 데이터의 whale_training 폴더에는 [그림 11-8]과 같이 고래의 이미지 36장이 들어가도록 한다.



[그림 11-8] 고래 훈련 데이터

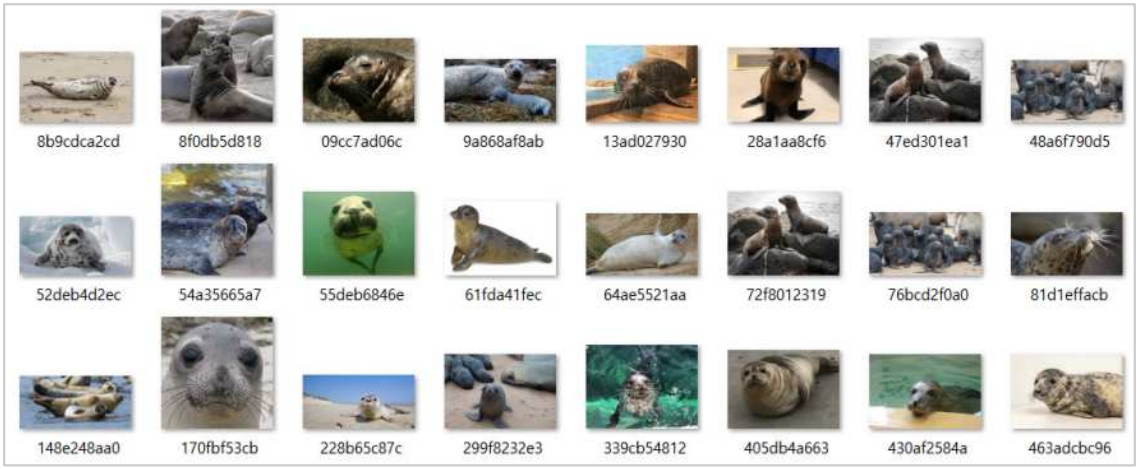
② 테스트 데이터

테스트 데이터도 훈련 데이터와 동일하게 [그림 11-9]와 같이 test 폴더를 만들고 하위 폴더에 seal_test, sharkseal_test, whaleseal_test 클래스 폴더를 만들어 이미지 파일을 넣는다.



[그림 11-9] 테스트 데이터 하위 폴더

테스트 데이터의 seal_test 폴더에는 [그림 11-10]과 같이 바다표범의 이미지 24개가 들어가도록 한다.



[그림 11-10] 바다표범 테스트 데이터

테스트 데이터의 shark_training 폴더에는 [그림 11-11]과 같이 상어의 이미지 24개가 들어가도록 한다.

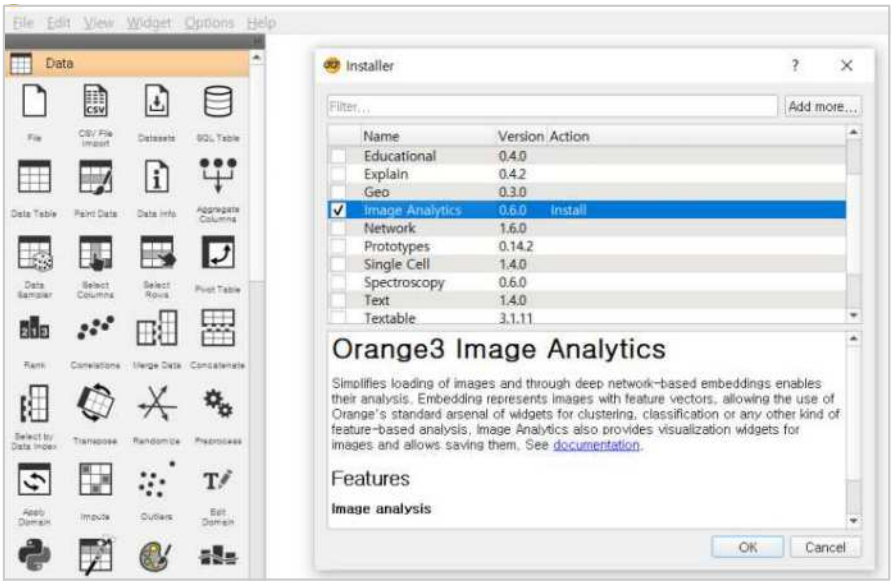
[그림 11-11] 상어 테스트 데이터

테스트 데이터의 whale_training 폴더에는 [그림 11-12]와 같이 고래의 이미지 24개가 들어가도록 한다.

[그림 11-12] 고래 테스트 데이터

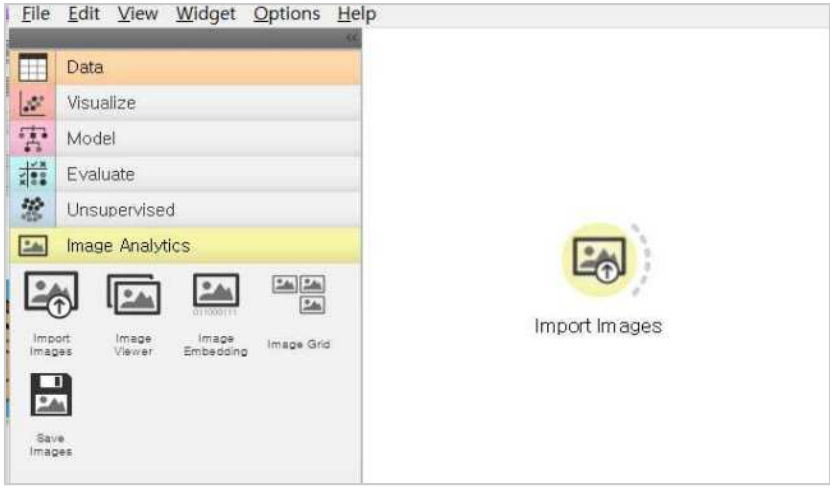
3 오렌지에 학습 데이터 불러오기

먼저 이미지 분석을 위하여 [Options] 메뉴에서 [Add-ons]을 클릭하면 다음과 같은 창이 나타난다. 추가 기능 중에서 [ImageAnaytics]앞에 체크(☑)를 눌러 추가 설치한다.



[그림 11-13] Image Anaytics 설치

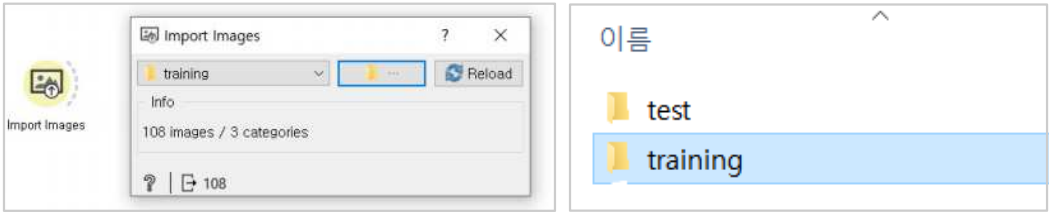
설치 후 오렌지를 종료하고 다시 실행해야 추가 설치된 기능을 사용할 수 있다. 그림과 같이 오렌지 실행 시 왼쪽 위젯 도구상자에 [Image Analytics]가 추가되었다.



① 학습 데이터 불러오기

이미지 학습 데이터를 불러오기 위해 [Import Images] 위젯을 선택하여 창에 놓는다. [Import Images]을 더블 클릭하면 [그림 11-14]와 같이 이미지를 업로드 하기 위해 폴더를 선택할 수 있다.

Animall Dataset에서 Training 폴더를 선택한다.

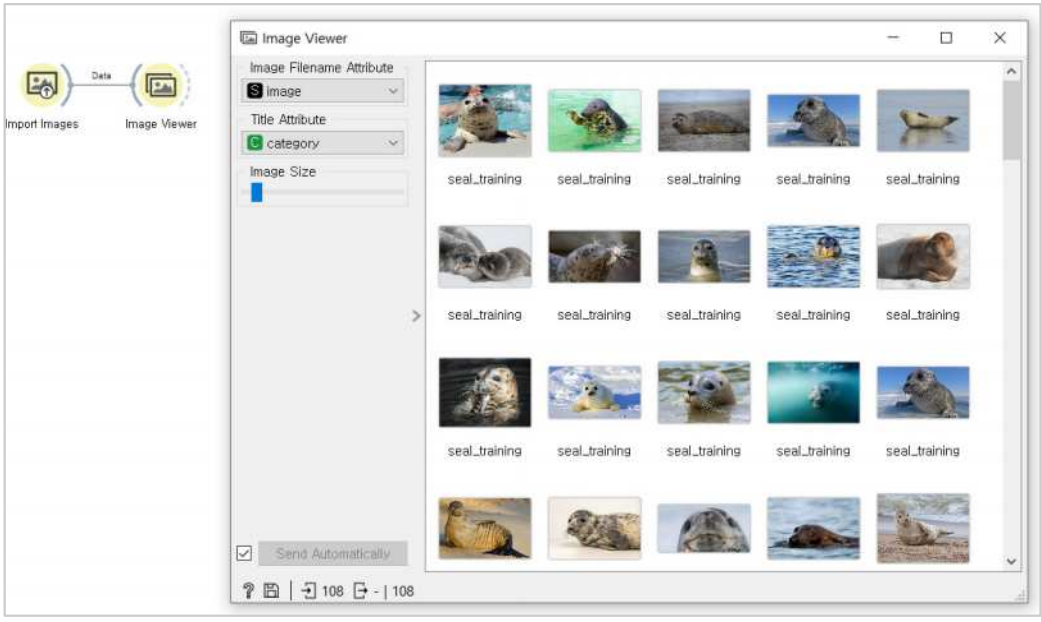


[그림 11-14] training 폴더 선택

② 이미지 데이터 보기

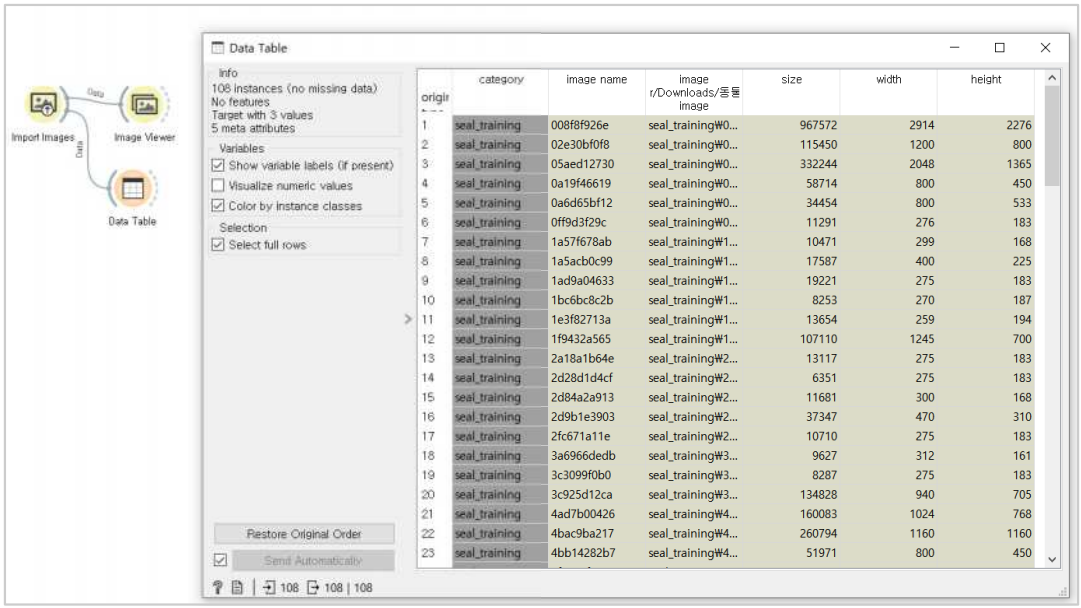
이미지를 확인하기 위해 [Image Analytics]위젯에서 [Image Viewer] 위젯을 끌어내어 [Import Images] 위젯과 연결한다.

[Image Veiwer] 위젯을 더블 클릭하면 폴더에 있는 이미지를 이름과 함께 볼 수 있다.

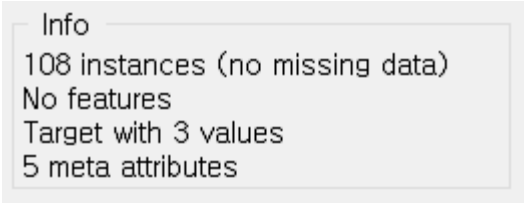


③ 데이터 테이블 보기

가져온 이미지에 [Data table]을 연결하면 [그림 11-15]와 같이 이미지 이름과 크기, 가로, 세로 길이 등을 테이블 형태로 보여준다. 정보(Info)를 살펴보면 3개의 값을 지닌 종속변수(Target)가 있으며, 기계학습을 위한 독립변수(features)는 없고, 5개의 meta 속성으로만 구성되어 있다.



[그림 11-15] Animall Dataset의 데이터 테이블

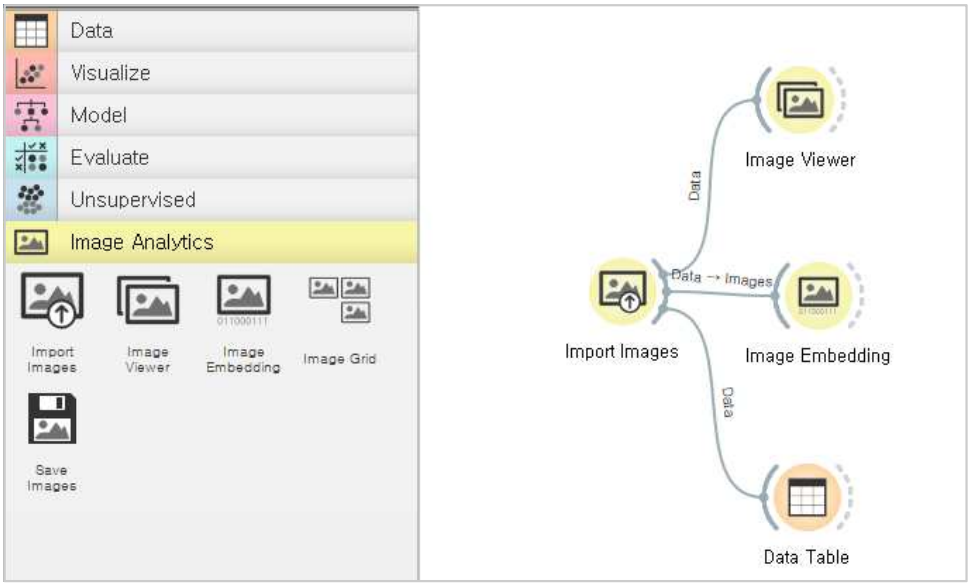


[그림 11-16] Animall Dataset의 데이터 테이블 속성

02 데이터를 탐색하고 전처리하자

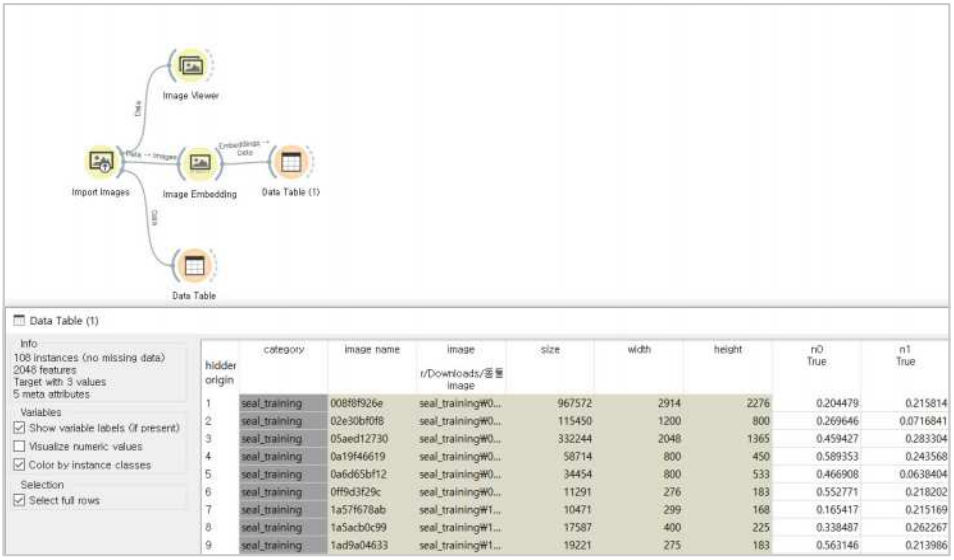
1 훈련 데이터 이미지 임베딩

이미지를 기계학습에 이용하려면 이미지 임베딩(Image Embedding)을 수행해야 한다. 이미지 임베딩은 딥러닝을 이용하여 각 이미지의 특징 벡터를 추출해 내는 역할을 한다. [Image Analytics] 위젯의 [Image Embedding]을 추가하고 [Import Images] 위젯과 연결한다.

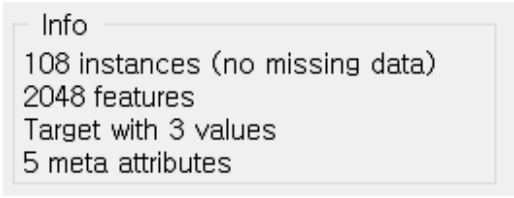


[그림 11-17] Image Embedding 위젯 연결

[Image Embedding]위젯과 [Data table] 위젯을 연결하여 이미지 임베딩 후의 데이터 테이블을 살펴보자. 데이터 테이블을 확인해 보면 벡터 속성이 추가되어 있다. [그림 11-18]과 같이 2048개의 특징(features)들이 추가된 것을 확인할 수 있다. 추가된 특성은 이미지 데이터의 내용에서 특징을 추출하여 수치화하여 나타낸 것이다. 인공지능은 이 속성을 이용하여 모델학습을 하게 된다.



[그림 11-18] 이미지 임베딩 후 데이터 테이블

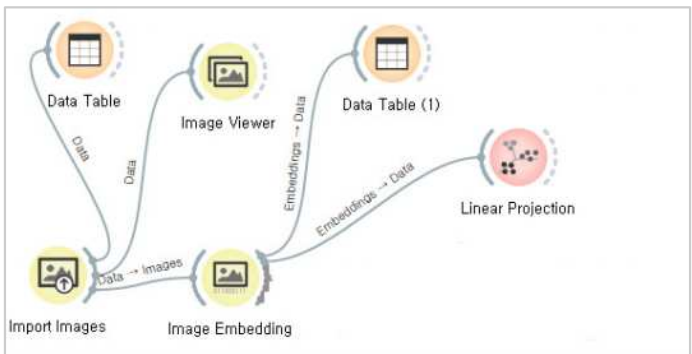


[그림 11-19] 이미지 임베딩 후 데이터 테이블 속성

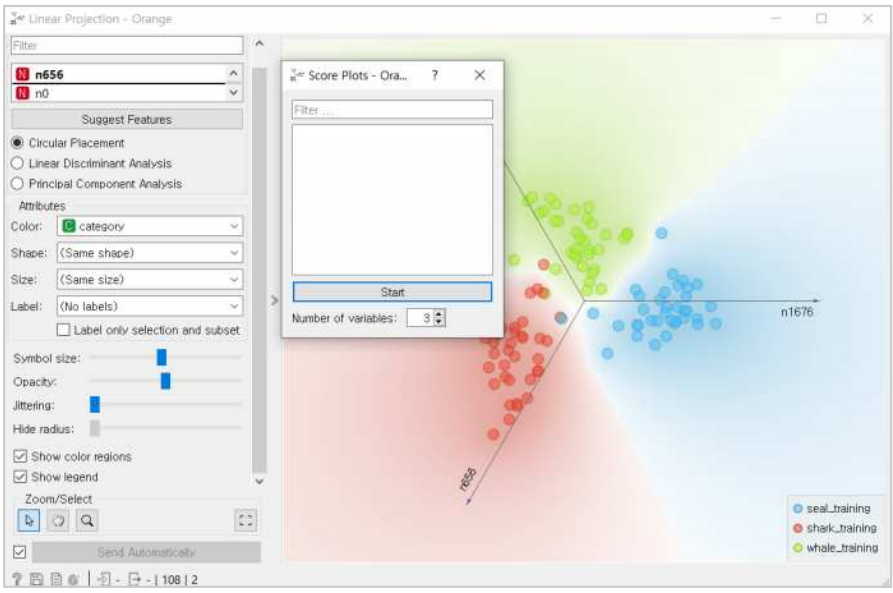
2 이미지 특징 시각화

이미지 임베딩한 결과 데이터 테이블을 살펴보면 2048개의 특징이 추가되고 수치화된 값으로 나타난다. 바다표범(seal)의 이미지에서 추출한 특징 벡터로 분류가 가능한지 알아보기 위해 Linear Projection 위젯을 이용하여 시각화해보자.

[Visualize] 위젯에서 [Linear Projection] 클릭하면 화면에 나타난다. 그림과 같이 [Image Embedding] 위젯과 연결한다.

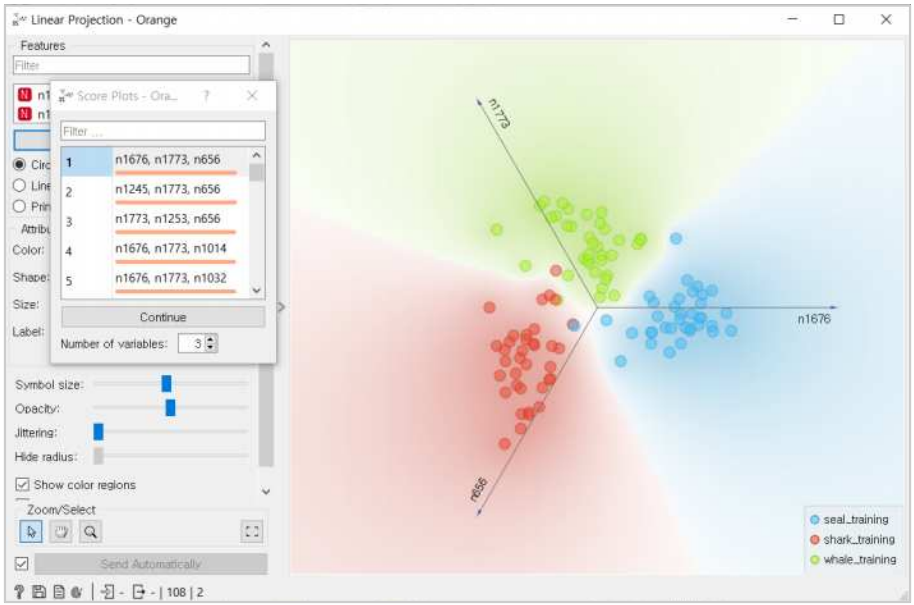


[Linear Projection] 위젯의 실행화면에서 'Suggest features'를 누르면 'Score Plots' 창이 나타난다. 여기에서 'start'를 누르면 많은 속성 중에 분류가 잘 이루어지는 어떤 속성의 조합을 찾아준다.



[그림 11-20] Linear Projection 위젯에서 분류가 잘 이루어지는 특징 조합 찾기

아래 [그림 11-21]은 2048개의 속성 중 n1676, n1773, n656의 세가지 속성으로 조합할 때, 바다표범, 상어, 고래를 분류할 수 있다는 것을 보여준다. 위에서 Show color regions 옵션을 선택하였기 때문에 배경 색이 3가지로 나타난다.



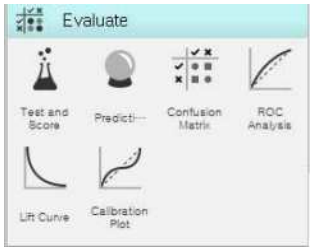
[그림 11-21] 분류가 잘 이루어지는 특징 조합 결과

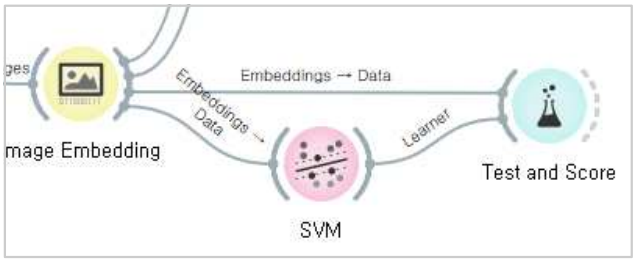
이처럼 [Linear Projection] 위젯을 이용하면 이미지 임베딩으로 수치화된 특징값으로 바다표범, 상어, 고래를 분류할 수 있다는 것을 시각적으로 확인할 수 있다.

03 모델 학습하고 성능 평가하자

1 학습 모델 선택하고 학습시키기

이미지 임베딩으로 처리한 이미지 데이터와 기계학습 알고리즘을 연결하면 이미지를 분류할 수 있는 기계학습 모델을 만들 수 있다. 모델 학습에 필요한 것은 데이터와 기계학습 알고리즘이다. 이미지 임베딩한 데이터와 기계학습 알고리즘 중 [SVM] 위젯과 [Test and Score] 위젯을 연결하여 모델 학습시킨다. [SVM] 위젯은 [Model] 위젯 하위 폴더에 있다.



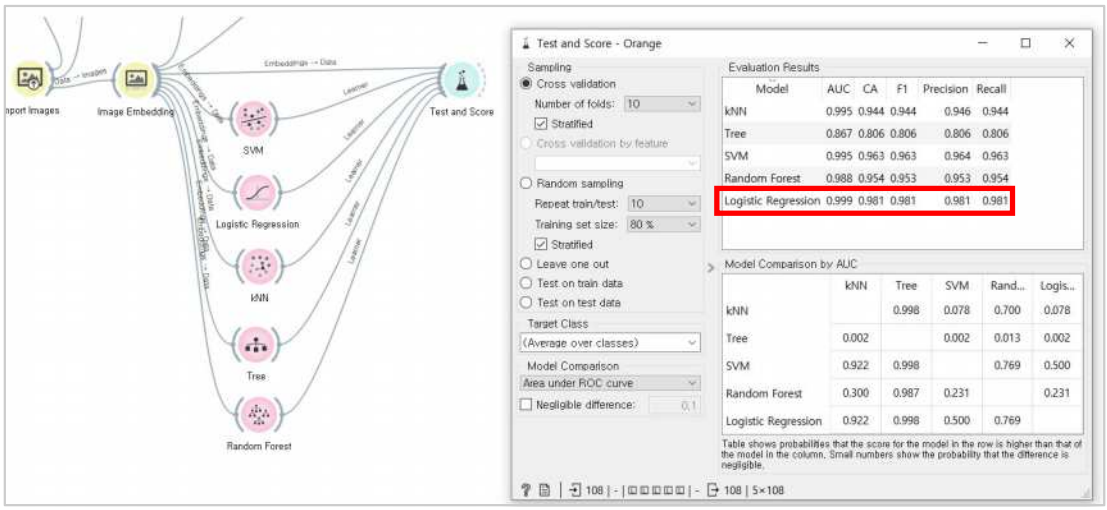


2 성능 평가하기

① 성능 좋은 모델을 결정하자

[Test and Score] 위젯에서 다양한 성능 평가 방법을 정할 수 있다. 바다 동물 훈련 데이터와 테스트 데이터를 분리했기 때문에 108개의 훈련 데이터를 이용하여 Cross validation을 선택하고 folds의 수를 10으로 설정하여 성능 평가였더니 분류 정확도(CA, Classficatin Accuracy)가 0.963으로 나타났다.

오렌지3에서는 여러 알고리즘을 동시에 연결하여 어느 모델이 성능이 좋은지 비교해볼 수 있다. [Logistic Regression]과 [kNN], [tree] 알고리즘을 동시에 연결해보자.

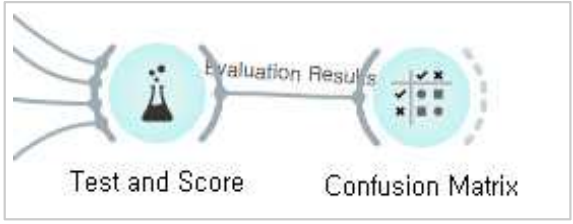


[그림 11-22] 여러 가지 모델을 연결한 결과

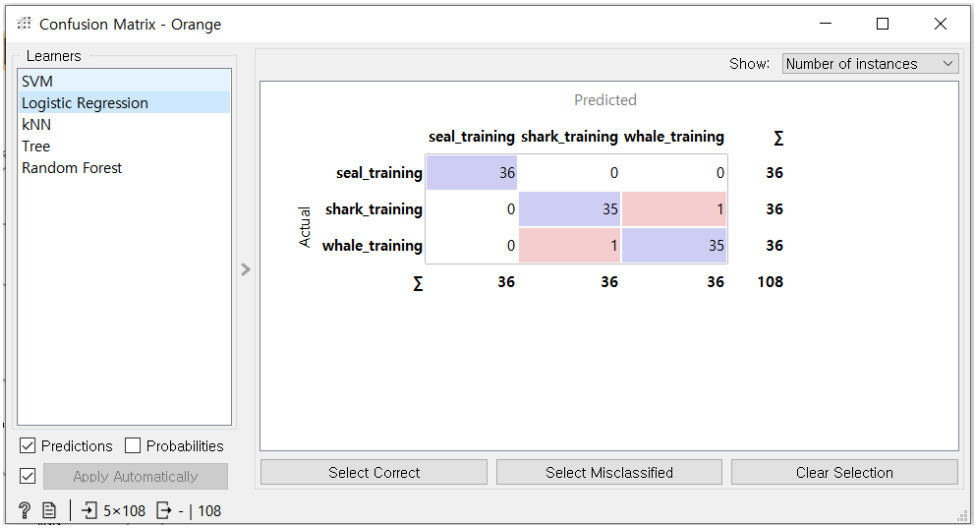
성능 평가를 비교해 보면 Logistic Regression 모델의 성능(CA)이 가장 우수한 것으로 나타났다. 따라서 바다표범, 상어, 고래를 분류하는 문제는 Logistic Regression 알고리즘을 이용하여 모델을 만드는 것이 적합하다는 것을 알 수 있다.

② 어떤 것을 잘못 분류했나?

[Test and Score] 위젯으로 성능 평가한 결과를 [Confusion Matrix] 위젯에 연결하여 실제 데이터를 어떻게 예측하였는지 살펴보자. [Confusion Matrix] 위젯은 [Evaluate] 위젯에서 찾을 수 있다.



[Confusion Matrix] 위젯을 더블 클릭하여 성능 평가 결과를 확인해보자.



[그림 11-23] 훈련 데이터 성능 평가 결과

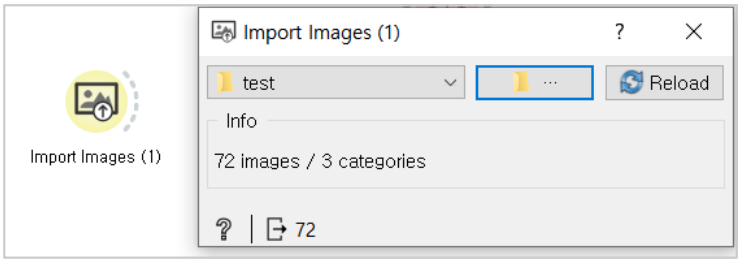
[그림 11-23]과 같이 바다표범은 36개 중 36개, 상어는 36개 중 35개, 고래는 36개 중 35개로 예측하였다. 따라서 분류 정확도(CA)는 36+35+35/108=0.981로 나타났다.

3 테스트 데이터로 예측하기

이제 훈련에 사용하지 않은 테스트 데이터로 바다표범, 상어, 고래를 얼마나 잘 예측할 수 있는지 확인해 보자.

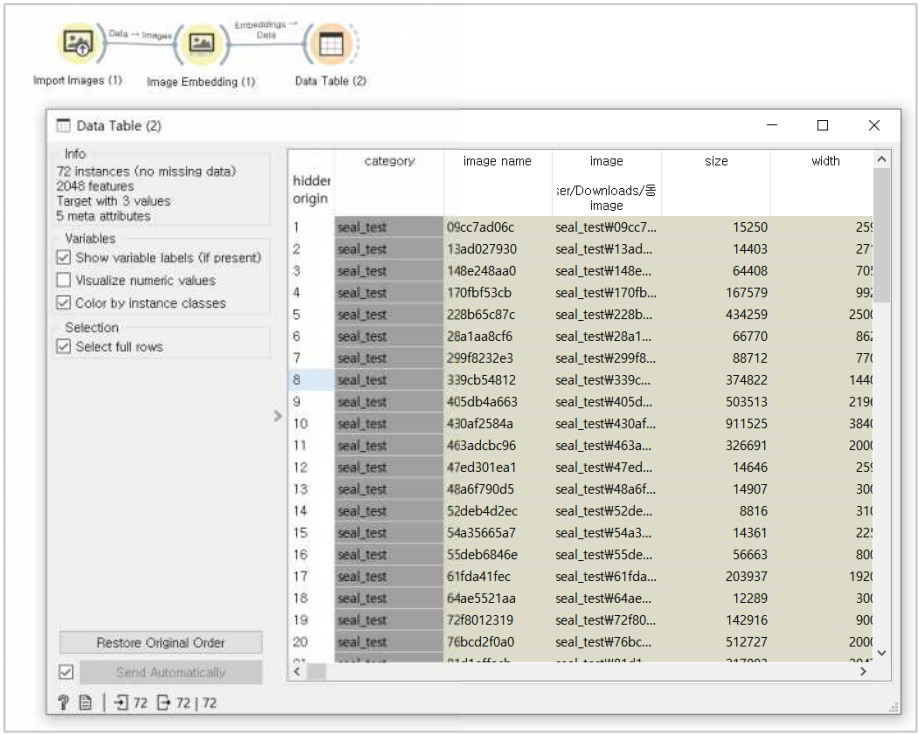
① 테스트 데이터 불러오기

테스트 데이터도 같은 방법으로 [Import Images] 위젯을 이용하여 [test] 폴더에 저장된 테스트 데이터 이미지를 불러온다.



② 테스트 데이터 이미지 임베딩

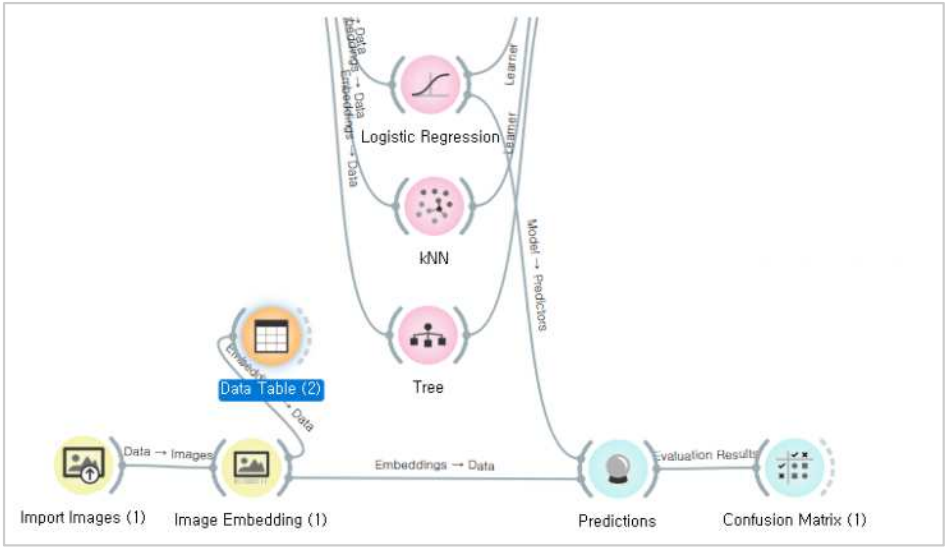
테스트 데이터도 훈련 데이터와 같은 방법으로 이미지 임베딩을 수행하고 데이터 테이블을 확인해보자. 데이터 테이블을 확인해보면 수치화된 데이터 테이블을 확인할 수 있다.



[그림 11-24] 테스트 데이터 임베딩한 후 데이터 테이블

③ 테스트 데이터로 예측하기

앞서 학습시켰던 모델들과 테스트 데이터를 [Predictions] 위젯에 연결한다. 테스트 데이터를 얼마나 잘 예측하였는지 [Predictions]의 예측 결과를 통해 확인할 수 있다. [Predictions] 위젯은 [Evaluate] 위젯에서 찾을 수 있다.



[그림 11-25] 테스트 데이터로 예측하기

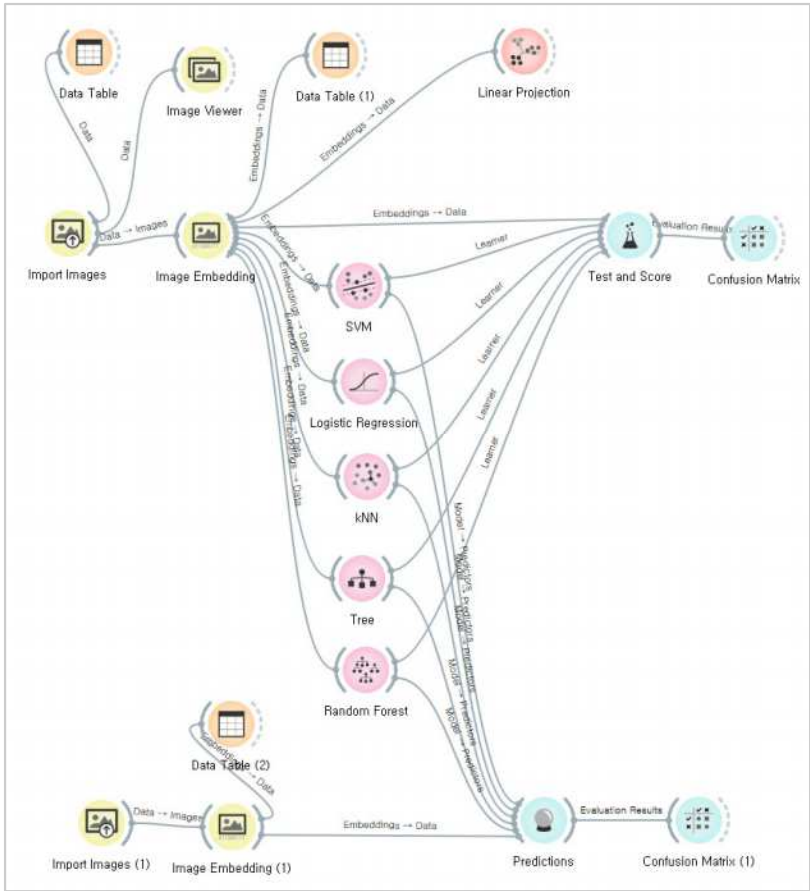
[Prediction] 위젯으로 예측한 결과를 나타내면 다음과 같다. 72개의 테스트 데이터 중에 2개가 잘 못 분류된 것을 확인할 수 있다. 분류 정확도(CA)가 24+24+22/72=0.972로 높게 나타났다.

	Logistic Regression	category
16	0.00 → seal_training	seal_test
17	0.00 → seal_training	seal_test
18	0.00 → seal_training	seal_test
19	0.00 → seal_training	seal_test
20	0.00 → seal_training	seal_test
21	0.00 → seal_training	seal_test
22	0.00 → seal_training	seal_test
23	0.00 → seal_training	seal_test
24	0.00 → seal_training	seal_test
25	0.05 → whale_traini...	shark_test
26	0.99 → shark_traini...	shark_test

	Logistic Regression	category
38	1.00 → shark_traini...	shark_test
39	0.93 → shark_traini...	shark_test
40	1.00 → shark_traini...	shark_test
41	1.00 → shark_traini...	shark_test
42	1.00 → shark_traini...	shark_test
43	1.00 → shark_traini...	shark_test
44	1.00 → shark_traini...	shark_test
45	0.99 → shark_traini...	shark_test
46	1.00 → shark_traini...	shark_test
47	1.00 → shark_traini...	shark_test
48	0.15 → whale_traini...	shark_test

[그림 11-26] 고래를 상어로 2개 잘 못 예측함.

바다표범, 상어, 고래를 분류하는 기계학습 모델을 구현하는 전체 과정은 [그림 11-27]과 같다.



[그림 11-27] 바다표범, 상어, 고래를 분류 기계학습 모델 전체 과정

정리하기

바다표범, 상어, 고래를 분류하는 기계학습 모델을 만들어보았다. 세 가지 바다 동물의 데이터를 훈련 데이터와 테스트 데이터로 분리하고, 이를 학습 알고리즘 5가지를 통하여 성능 분석을 해보았다. 그 결과 Logistic Regression 알고리즘의 성능이 가장 우수하였다. 이 모델을 이용하면 새로운 바다 동물 이미지를 입력하였을 때 바다표범, 상어, 고래 중 어떤 바다 동물인지 예측할 수 있을 것이다.

[참고 문헌]

1. 서울과학종합대학원 디지털혁신처(2021). 3시간 만에 배우는 인공지능 데이터분석. 오렌지. 서울경제경영.
2. 손원성 외 3인(2021). 오렌지3로 알아가는 머신러닝 데이터 분석. 홍릉.
3. 이고잉 외 2인(2021). 생활코딩 머신러닝. 위키북스.
4. 동물데이터. Kaggle. https://www.kaggle.com/iamsouravbanerjee/animal-image-dataset-90-different-animals
5. 오렌지. https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/data/rank.htm

12. 건강상태를 알면 당뇨병을 예측할 수 있을까?

경산과학고등학교 교사 임 진 숙

학습 진행 과정

1단계	데이터 준비	- 사용 데이터: 피마 인디언 당뇨병 데이터 - 수집: kaggle.com에서 'diabetes'로 검색 - 데이터 편집: 훈련 데이터와 테스트 데이터 분리
2단계	데이터 불러오기	- 데이터 불러오기 - 데이터의 속성의 Role(역할) 설정하기
3단계	데이터 시각화	- 시각화 도구을 이용한 데이터 탐색 - 사용한 시각화 도구: scatter, distributions, feature statistics
4단계	데이터 전처리	- 결측치 처리, 정규화
5단계	모델 학습	- 분류를 이용한 모델 학습 - 사용된 알고리즘: Logistic Regression, Random Forest, Neural Network
6단계	성능 평가	- test and score, cross validation을 이용한 성능 평가 - 혼동 행렬을 이용한 성능 평가
7단계	예측	- Prediction 위젯으로 테스트 데이터로 예측하기

학습 내용 요약

데이터 종류	문제 유형	기계학습 알고리즘	성능 평가 도구
정형 데이터	분류	Logistic Regression Neural Network	CA 혼동행렬

01 해결해야 할 문제는 무엇일까?

문제 상황

당뇨병은 고혈압과 함께 대표적인 만성질환으로 손꼽힌다. 2016년 국민건강영양조사 결과에 의하면 국내 30세 이상 성인 인구의 7명 중 1명(14.4%)이 당뇨병을 가지고 있는 것으로 나타났으며, 65세 이상 성인에서는 10명 중 3명이 당뇨병을 앓고 있다. 사회가 고령화됨에 따라 당뇨병 환자는 더욱 늘어날 것으로 예측된다. 또한 2030 젊은층의 당뇨병 유병률도 꾸준하게 상승 중이다. 당뇨병은 혈당이 매우 높지 않은 경우에는 증상을 거의 느끼지 못하는 경우가 많다. 하지만 몸이 지속적으로 고혈당에 노출되면 여러 장기에 위험한 합병증이 발생할 수 있어 주의가 필요하다. 정기검진으로 조기에 당뇨병을 발견하고 진단 후에는 혈당을 철저하게 관리해야 건강한 일상을 유지할 수 있다.

(출처 : 헬스인뉴스, http://www.healthinnews.co.kr)

피마인디언은 아리조나주에 살던 아메리가 원주민 부족이다. 유전적으로 당뇨병에 취약해 당뇨 연구에 자주 활용되었다. 피마 인디언 데이트 세트는 당뇨병 진단을 위한 미국 국가 건강 검진 프로그램에 참여한 21세 이상의 피마 인디언 출신 여성 768명의 데이터로 구성되어 있다. 피마 인디언 당뇨병 데이터세트를 활용하여 당뇨병을 예측해보자.

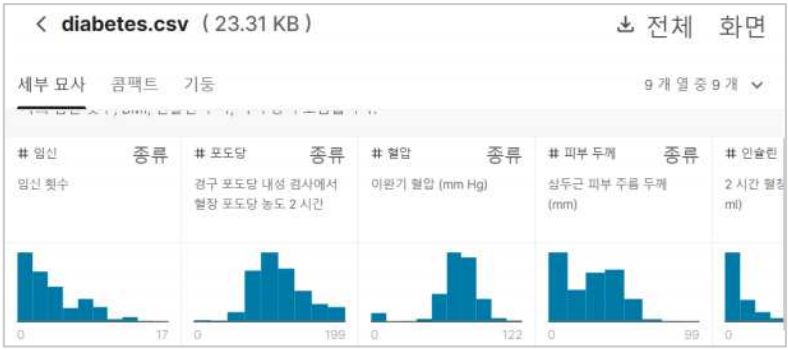
01 데이터를 준비하자

1 피마 인디언 당뇨병 데이터

당뇨병 데이터 세트는 캐글(https://www.kaggle.com/uciml/pima-indians-diabetes-database)에서 다운로드할 수 있다. 캐글 사이트에 로그인한 후 '당뇨병(diabetes)'으로 검색하면 다음과 같은 피마 인디언 당뇨병 데이터 세트를 다운로드할 수 있다.

2 기계학습을 위한 데이터 준비

캐글에서 '당뇨병'으로 검색하여 피마인디언 데이터세트를 다운로드하여 속성값의 의미를 살펴보자. 구글 변역의 자동 번역으로 속성명의 의미를 살펴보자.



① 데이터 살펴보기

데이터 파일 diabetes.csv을 다운로드하면 다음과 같은 속성명과 속성값을 확인할 수 있다.

피마 인디언 당뇨병 데이터를 살펴보면, 임신 횟수, BMI, 인슐린 수치, 나이 등 9개의 속성으로 구성되어 있다. 전체 데이터는 당뇨병 발병 여부를 나타내는 500개의 음성사례와 268개의 양성 사례로 구성되어 있다. 최초 측정 후 5년내 당뇨병이 발병하면 Outcome은 1, 그렇지 않으면 0이다.

[표 12-1] 당뇨병 데이터 속성과 의미

	속성명	의미	비고
1	Pregnancies	임신 횟수	
2	Glucose	혈당	70-110
3	BloodPressure	이완기 혈압	
4	SkinThickness	피부 두께	
5	Insulin	인슐린 농도	
6	BMI	체질량 지수	80-120
7	diabetesPedigreeFunction	당뇨병 혈통	
8	Age	나이	
9	Outcome	당뇨병 발병여부(0,1)	

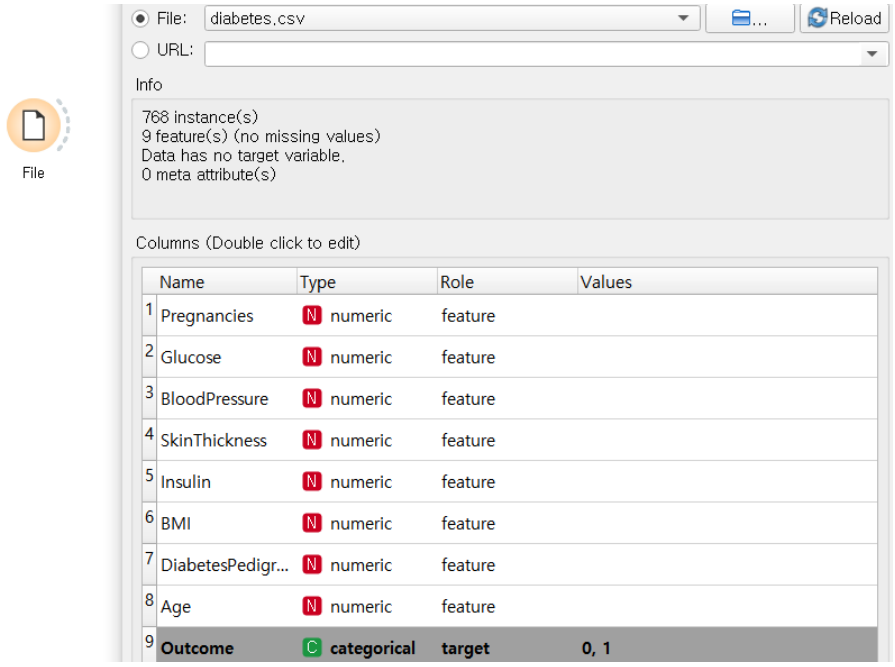
수 많은 사람들의 당뇨병에 영향을 미치는 요인들의 값을 수집하여 기계가 학습한다면, 당뇨병이 걸릴지 그렇지 않을지 예측할 수 있다.

	A	B	C	D	E	F	G	H	I
1	Pregnancie	Glucose	BloodPres	SkinThickn	Insulin	BMI	DiabetesP	Age	Outcome
2	6	148	72	35	0	33.6	0.627	50	1
3	1	85	66	29	0	26.6	0.351	31	0
4	8	183	64	0	0	23.3	0.672	32	1
5	1	89	66	23	94	28.1	0.167	21	0
6	0	137	40	35	168	43.1	2.288	33	1
7	5	116	74	0	0	25.6	0.201	30	0
8	3	78	50	32	88	31	0.248	26	1

[그림 12-1] 데이터 속성과 속성값 살펴보기

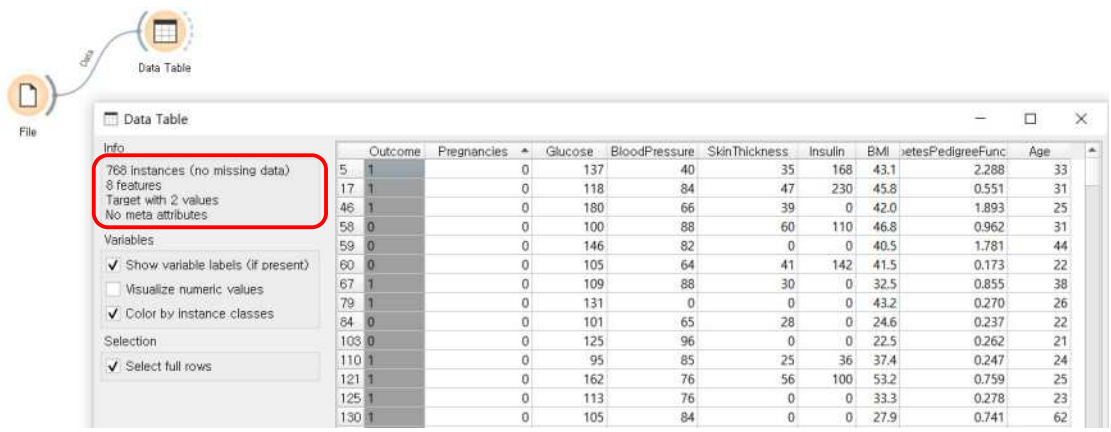
② 오렌지에서 당뇨병 데이터 가져오기

캐글에서 다운로드한 당뇨병 데이터 diaeties.csv파일을 오렌지에서 업로드하려면 file 위젯을 사용한다. 파일을 업로드하고 당뇨병 여부 속성은 종속 변수(target)로 나머지 속성은 독립 변수(feature)로 설정한다.



[그림 12-2] 데이터 입력과 속성의 역할 정하기

데이터 테이블을 살펴보면 768개의 데이터 샘플을 가지고 있으며, 8개의 독립변수와 1개의 독립변수로 구성되어 있다. 임신횟수, 혈당, 협압, 피부 두께, 인슐린, bmi, 당뇨병 혈통, 나이 등의 값이 당뇨병에 영향을 주는 독립변수가 될 수 있다.



[그림 12-3] 00이 포함된 당뇨병 데이터 테이블

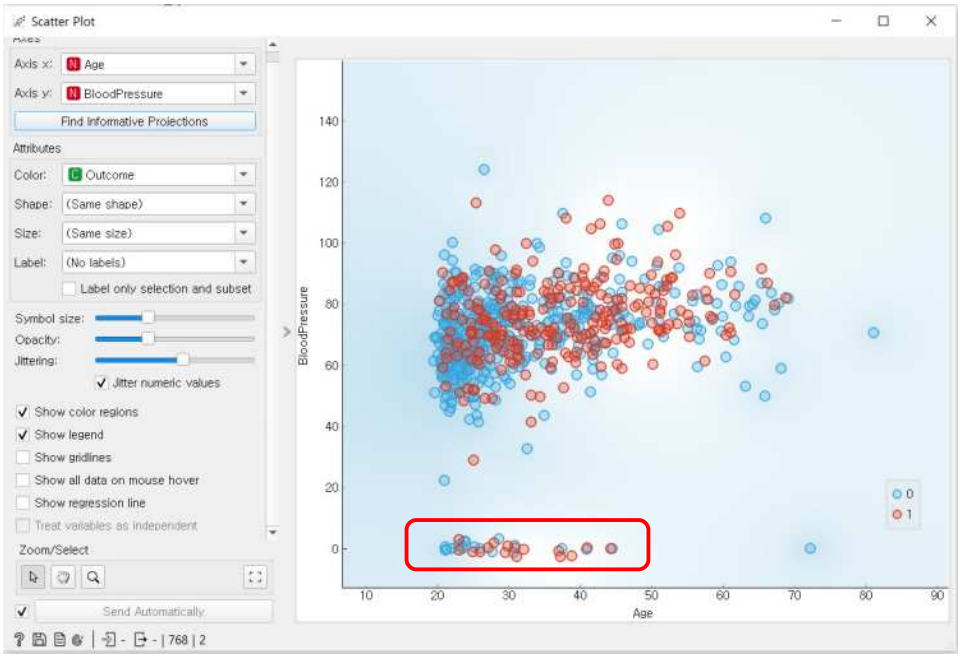
02 데이터 탐색하고 전처리하자

1 탐색적 데이터 분석

오렌지에서 당뇨병 데이터를 다양한 형태로 시각화해보자.

① 산점도로 나타내기

나이(Age)와 혈압(BloodPressure)을 x, y축에 놓고 산점도로 나타내면 [그림 12-4]와 같다.

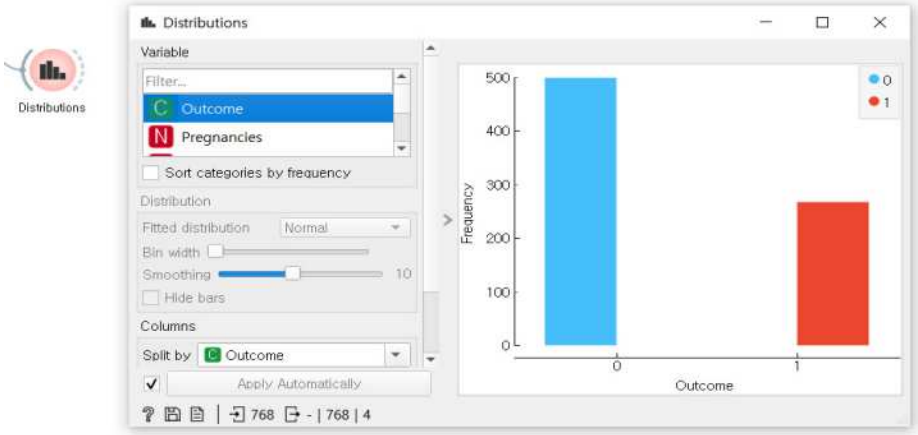


[그림 12-4] 나이와 혈압의 분포

위 그래프를 살펴보면 혈압이 0인 데이터가 많이 포함되어 있다. 보통 사람들의 혈압 정상범위는 80~120사이인데, 저혈압을 감안하더라도 그래프에서 혈압이 0인 데이터는 정상범위를 벗어난 이상치로, 결측치가 0의 값으로 채워진 것으로 유추해 볼 수 있다. 이처럼 데이터를 시각화해 보면 이상치를 발견할 수 있고, 전처리가 필요하다는 것을 알 수 있다.

② 도수 분포표로 나타내기

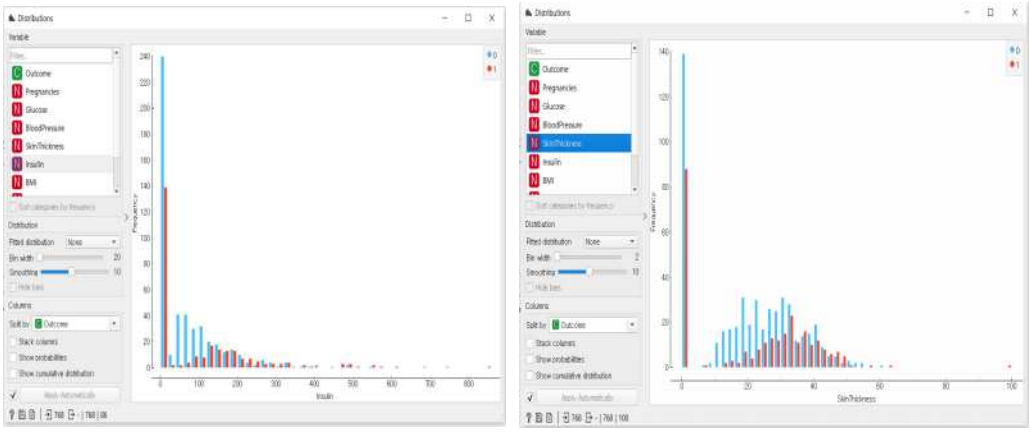
도수 분포표(distribution)을 활용하여 Outcome의 도수를 보면 당뇨병 걸리지 않은 사람의 데이터는 500개, 당뇨병 걸린 사람의 데이터는 268개이다. 두 클래스의 데이터 수에 불균형이 있다는 것을 알 수 있다. 당뇨병이 걸리지 않은 데이터가 거의 2배나 많다는 것도 알 수 있다. 가능하다면 데이터의 샘플 수를 비슷하게 하는 것이 좋다.



[그림 12-5] 당뇨병 여부(0, 1)에 따른 도수

데이터 탐색 단계에서는 데이터 시각화를 통해 데이터에 숨어 있는 의미나 이상치와 결측치 등을 발견할 수 있다. 피마 인디언 데이터 세트에서 데이터 분포 그래프를 그려보면 다음과 같이 이상치와 결측치 등을 확인할 수 있다.

아래 그래프를 보면 당뇨병 데이터에서 인슐린(Insulin)이 0인 데이터가 상당히 많다. 그리고 피부 두께도 0인 데이터가 상당히 많이 있다.



인슐린이나 피부두께가 0의 값은 이상치인데, 결측치가 0으로 채워진 것으로 판단할 수 있다. 기계학습을 위해 이러한 데이터는 적절한 데이터로 전처리를 해야 한다.

③ 특성 통계표로 나타내기

전체 데이터에 대한 특성 통계표를 살펴보면 [그림 12-6]과 같다.

특성 통계표는 평균, 표준편차, 최젓값과 최댓값 등 통계정보를 확인할 수 있으며, 도수분포표와 같은 그래프도 시각적으로 보여준다.

아래 그래프를 보면 최솟값이 0인 속성이 상당히 많이 있다. 임신 횟수는 0의 값을 가질 수 있다. 그러나 혈압(BloodPressure),피부두께(SkinThickness), BMI 등은 최솟값이 0을 가질 수 없으므로 이상치임을 알 수 있다.

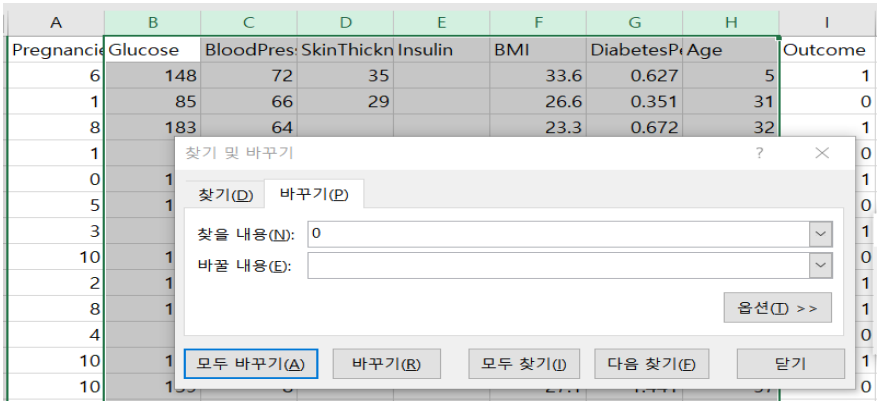
[그림 12-6] 특성 통계표로 살펴본 데이터 탐색

이러한 결과를 보면 0의 값으로 이상치가 있는 값에 대해 전처리가 필요함을 알 수 있다.

2 데이터 전처리

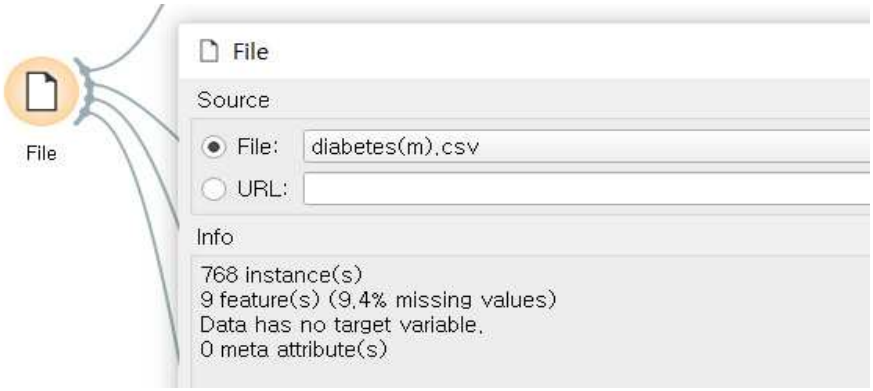
① 스프레드시트로 결측치 처리하기

데이터 테이블에서 임신 횟수와 당뇨병 발병여부는 0의 값을 가질 수 있기 때문에 이를 제외하고 나머지 값을 0의 값을 값이 없는 결측치로 처리하여 저장하고 파일 이름을 diabetes(m)으로 수정한다. m은 missing value의 의미로 이름을 붙였다.



[그림 12-7] 0의 값을 결측치 처리

이렇게 결측치를 포함하도록 저장한 데이터를 오렌지에서 불러온다.



[그림 12-8] 결측치 포함한 데이터 파일

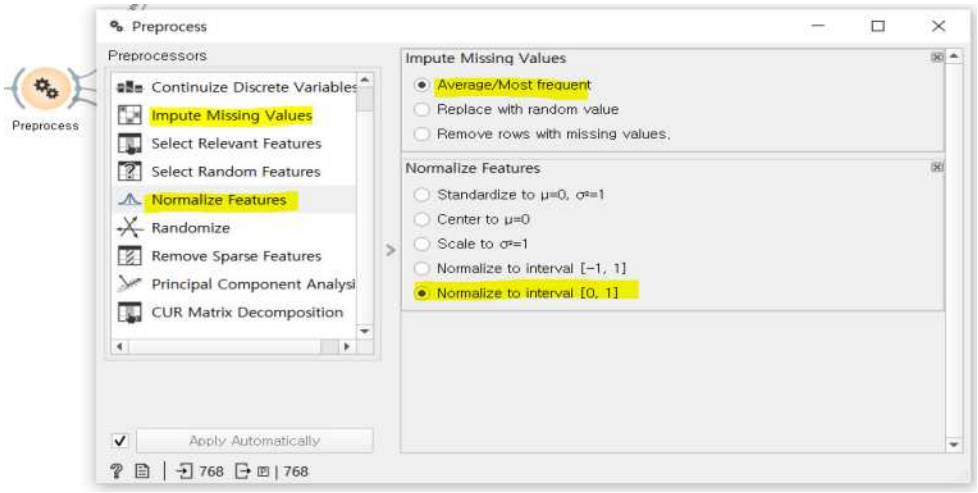
② 데이터 전처리(Preprocess)

결측치가 너무 많아서 이를 모두 삭제하면 너무 많은 데이터의 손실이 발생하기 때문에 다른 값으로 대체하는 것이 필요하다. 이를 위해 인슐린과 피부두께가 0인 값을 결측치(아무 값이 없는 상태)를 평균값으로 채우는 전처리를 수행한다.

결측치를 처리하는 여러 가지 방법 중에서 오렌지에서는 다음과 같은 세가지를 제공한다.

- 평균/최빈 값으로 채우기
- 랜덤한 값으로 채우기
- 결측치가 있는 행 삭제하기

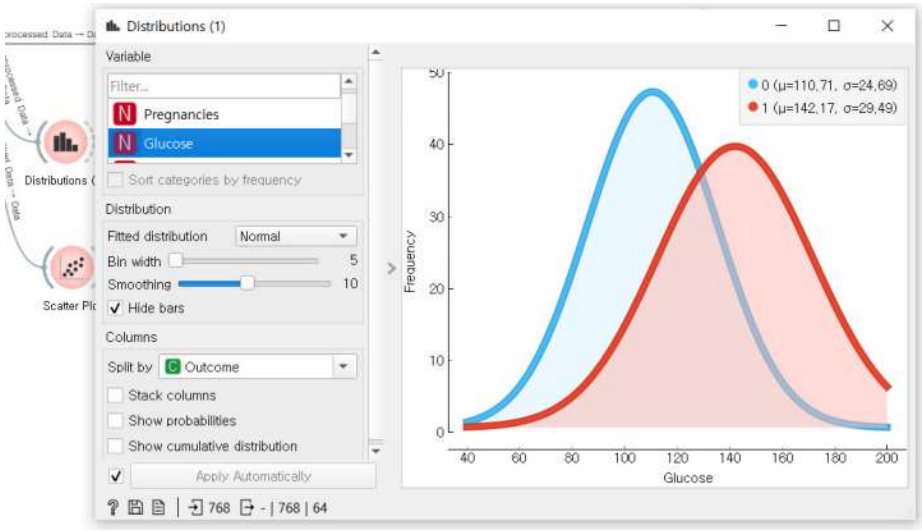
그리고 전체 데이터를 0~1사이의 값으로 정규화(Normalize)한다.



[그림 12-9] 결측치와 데이터 정규화 전처리

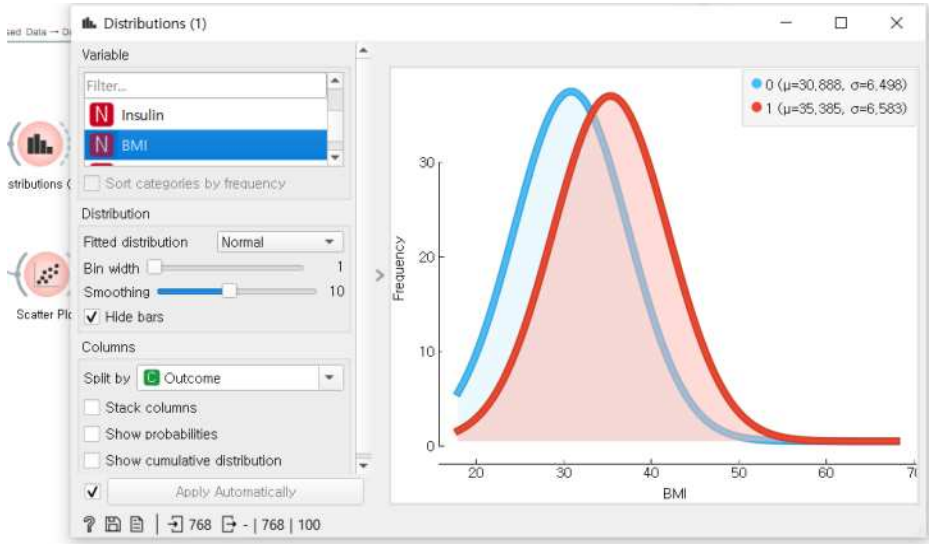
3 전처리후 데이터 분석

위 [그림 12-9]와 같이 도수분포표(distributin)위젯을 파일에 연결한 후 혈당(Glucose)을 살펴보니 당뇨병 발병 여부(Outcome)에 따라 혈당에 차이가 나는 것을 알 수 있다. 아래 그래프는 결측치만 처리하고 실젯값은 정규화하지 않은 값으로 시각화한 것이다.



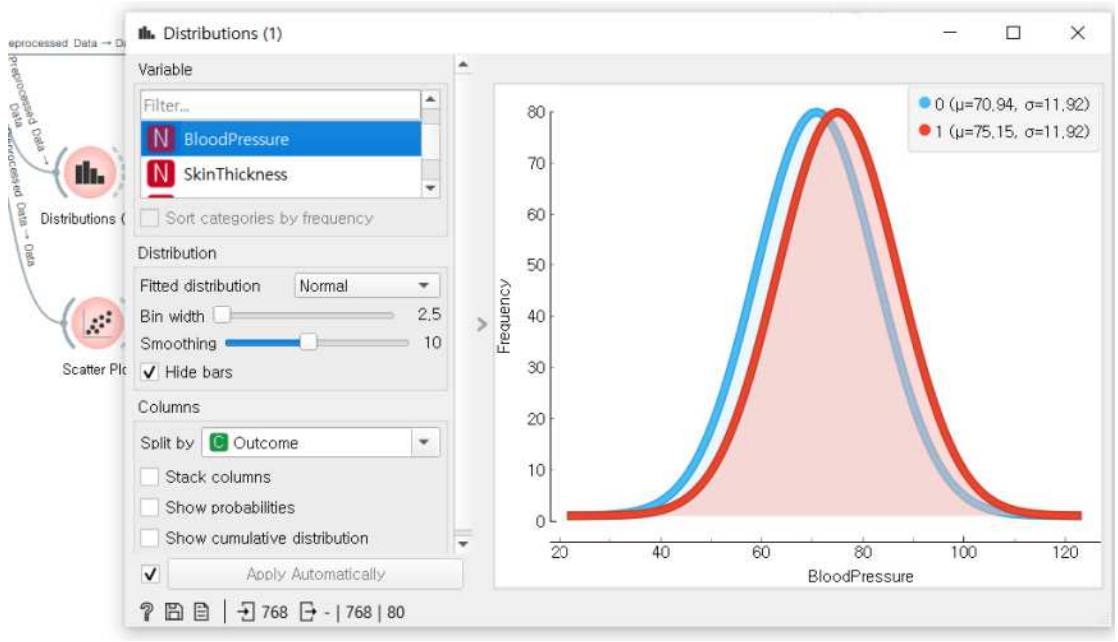
[그림 12-10] 당뇨병 여부에 따른 혈당의 분포

당뇨병이 발병한 클래스(1)는 혈당이 상대적으로 높았다. 이를 통해 혈당이 높으면 당뇨병 발병 확률이 높다는 것을 알 수 있다. 또한 BMI 값의 분포도 당뇨병 발병 여부에 따라 차이가 있다.



[그림 12-11] 당뇨병 여부에 따른 BMI의 분포

이러한 결과를 통해 혈당과 BMI 수치는 당뇨병을 예측하는 기계학습에 영향을 미치는 핵심 속성이라는 것을 알 수 있다. 반면 혈압(BloodPressure)은 당뇨병 여부에 따라 차이가 없어 당뇨병을 예측하기에 어려운 속성이다.



[그림 12-12] 당뇨병 여부에 따른 혈압의 분포

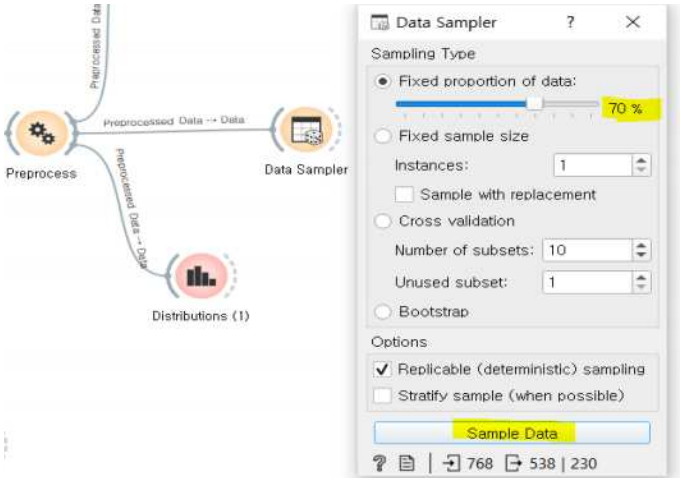
Rank 위젯을 이용하면 어떤 속성이 당뇨병 여부에 더 많은 영향을 미치는지 보여준다. ReliefF 를 기준으로 살펴보면 혈당, BMI, SkinThickness 순으로 나타났다.

[그림 12-13] 핵심 속성의 Rank

이와 같이 탐색적 데이터 분석을 통해 기계학습에 더 많은 영향을 미치는 속성이 무엇인지 확인할 수 있다.

03 모델 학습하고 성능 평가하자

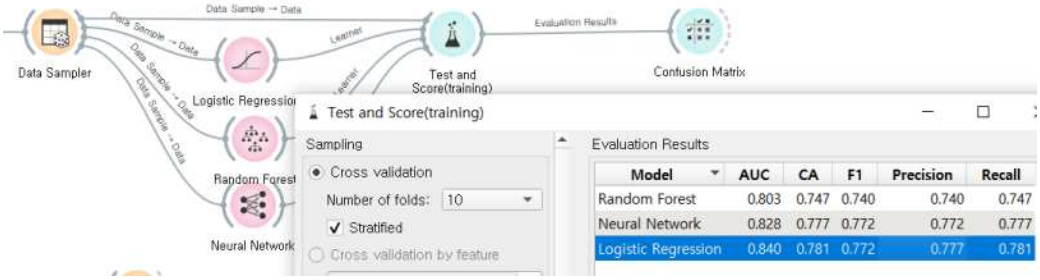
모델학습을 위해 데이터 샘플러(Data sampler)를 이용하여 훈련 데이터와 테스트 데이터를 분할한다. 훈련 데이터와 테스트 데이터의 비율을 조정할 수 있다.



[그림 12-14] 훈련 데이터와 테스트 데이터 분할

1 모델 학습

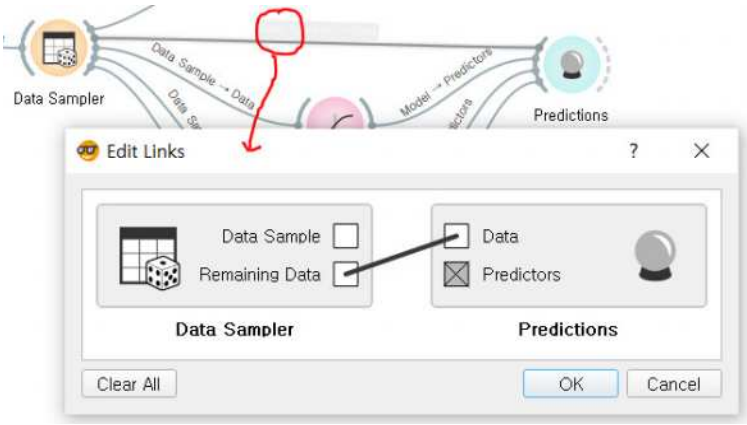
전처리한 데이터를 이용하여 모델 학습해보자. 데이터 샘플러에서 70%의 훈련 데이터는 기계학습 모델에 입력하고, 같은 데이터를 이용하여 모델 학습의 검증을 위해 크로스 밸리데이션(Cross validation)으로 설정한다. 검증 데이터를 활용하여 검증한 결과, 로지스틱 회귀(Logistic Regression)의 성능이 가장 좋았다.



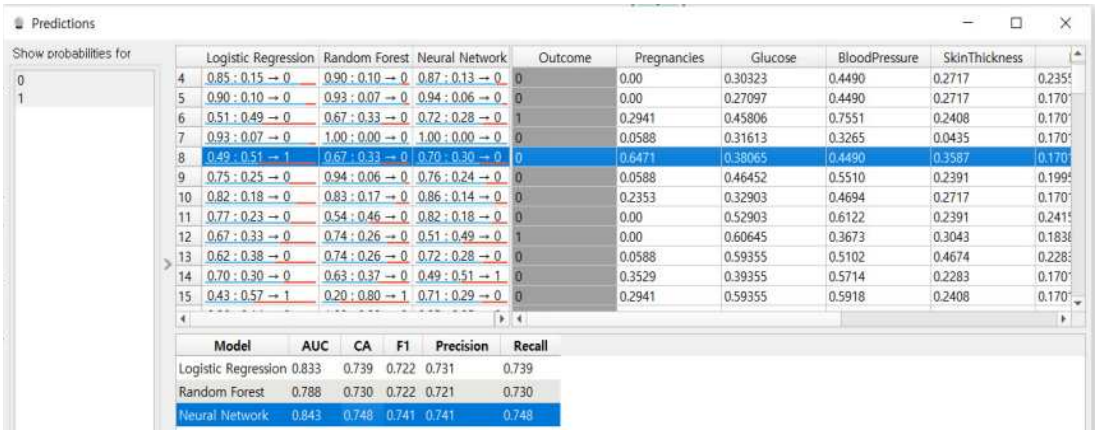
[그림 12-15] 모델 검증 결과

2 성능 평가

이제 데이터 샘플러에서 분할하여 남아 있는 30%의 데이터를 예측(Predictions)위젯에 연결하여 모델의 성능을 평가하고 예측한 결과를 확인해보자. 그림과 같이 Predictions 위젯에 데이터를 Remaining Data가 연결될 수 있도록 변경한다.



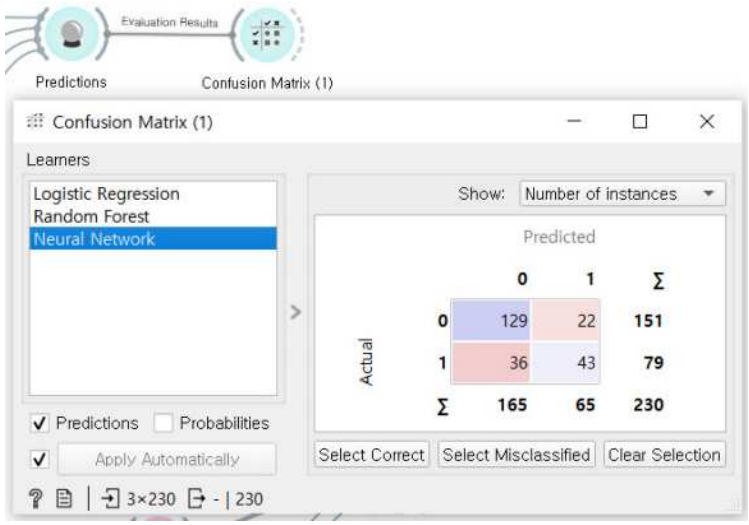
성능 평가 결과를 살펴보면 [그림 12-16]과 같다. 모델에 따라 예측되는 결과가 다르다는 것을 알 수 있다. 세가지 모델의 성능을 비교한 결과 Neural Network가 성능 중 분류 정확도(CA)가 0.740으로 상대적으로 높았다. 재현율(Recall)을 비교해 보아도 Neural Network이 가장 높게 나타났다.



[그림 12-16] 모델 예측과 성능평가 결과

조금 더 상세하게 성능 평가 결과를 알아보기 위해 혼동 행렬을 살펴보자. 신경망(Neural Network) 모델은 당뇨병이 걸리지 않은 사람의 데이터 151개 중 129개를 정확히 예측하였고, 당뇨병이 걸린 사람의 데이터 79개 중 43개를 정확히 예측하였다. 당뇨병과 같이 질병을 예측하는 모델은 당뇨병을 당뇨병으로 예측하는 것이 중요하다. 따라서 재현율이 높은 모델을 신경망(Neural Network) 모델을 선택하는 것이 적합하다.

혼동 행렬을 살펴보면 정확도, 정밀도, 재현율을 직접 계산할 수 있다. 혼동 행렬은 $CA = \frac{129+43}{129+22+36+43} = \frac{172}{230} = 0.748$ 로 계산할 수 있다.



[그림 12-17] 모델 예측의 혼동 행렬

피마 인디언 당뇨병 데이터는 클래스간 데이터 불균형 문제, 너무 많은 이상치(결측치)를 포함하고 있어 성능이 매우 높지는 않다. 모델의 성능을 개선하기 위해 이러한 문제를 개선한다면 더욱 성능이 좋은 기계학습 모델을 구현할 수 있을 것이다.

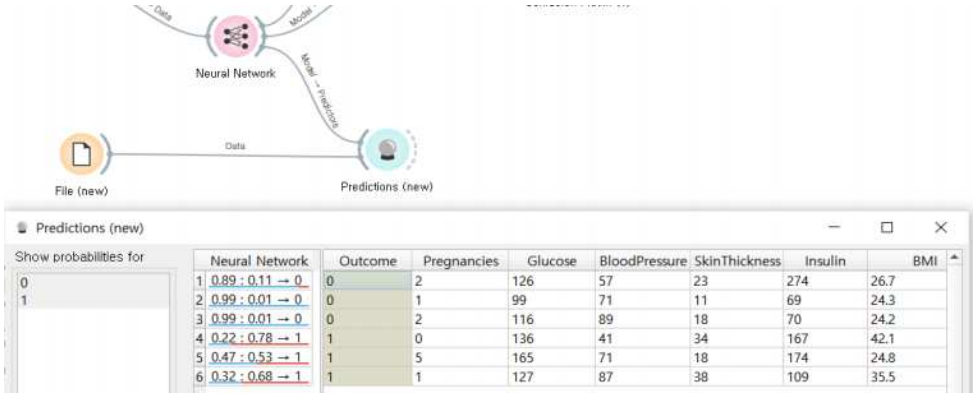
3 새로운 데이터로 당뇨병 예측하기

이렇게 만들어진 Neural Network 모델에 새로운 데이터 6개의 샘플을 입력하여 예측해 보자. 새로운 데이터 6개는 diabetes 데이터 세트에 결측치가 없었던 6개의 데이터를 임의로 뽑아서 0의 값이 없는 5개의 속성값에 ±1 계산하여 생성한 데이터이다. 예측 결과가 정확한지 알아 보기 위해 Outcome을 meta로 설정하였다.

A	B	C	D	E	F	G	H	I
Pregnancie	Glucose	BloodPres	SkinThickn	Insulin	BMI	DiabetesP	Age	Outcome
2	126	57	23	274	26.7	1.6	24	0
1	99	71	11	69	24.3	0.658	27	0
2	116	89	18	70	24.2	0.313	20	0
0	136	41	34	167	42.1	2.288	32	1
5	165	71	18	174	24.8	0.587	50	1
1	127	87	38	109	35.5	1.057	36	1

[그림 12-18] 새로운 데이터

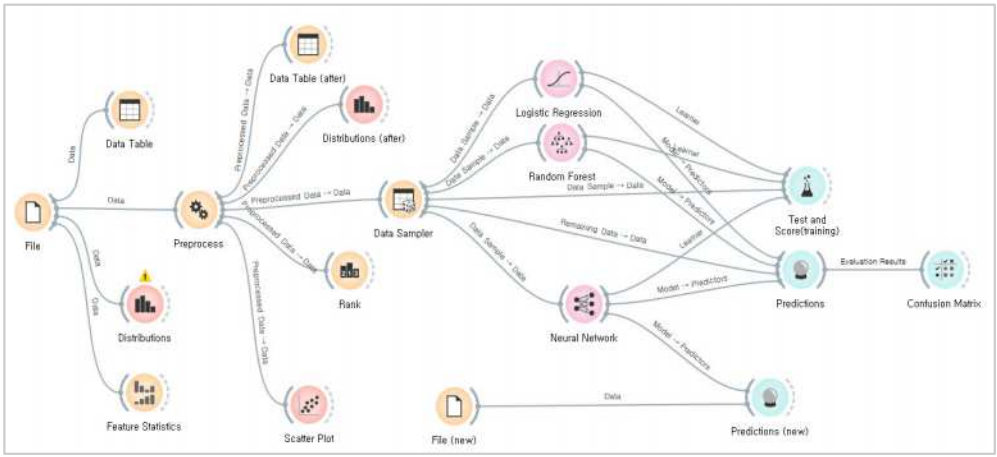
정답이 없기 때문에 성능평가 결과는 나타나지 않는다. meta로 처리한 Outcome과 비교해보면 모델의 예측이 정확하다는 것을 알 수 있다.



	Neural Network	Outcome	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
1	0.89 : 0.11 → 0	0	2	126	57	23	274	26.7
2	0.99 : 0.01 → 0	0	1	99	71	11	69	24.3
3	0.99 : 0.01 → 0	0	2	116	89	18	70	24.2
4	0.22 : 0.78 → 1	1	0	136	41	34	167	42.1
5	0.47 : 0.53 → 1	1	5	165	71	18	174	24.8
6	0.32 : 0.68 → 1	1	1	127	87	38	109	35.5

[그림 12-19] 새로운 데이터의 모델 예측

구현한 모델의 전체 구성도는 [그림 12-20]과 같다.



[그림 12-20] 당뇨병 예측을 위한 기계학습 모델 구현의 전체 과정

정리하기

이 장에서는 건강상태 데이터로 당뇨병을 예측하는 모델을 구현하기 위해 피마 인디언 당뇨병 데이터를 수집하여 훈련 데이터와 테스트 데이터로 분할하였다. 3가지 분류 모델을 이용하여 훈련하고 테스트 데이터로 성능 평가한 결과 신경망(Neural Network) 모델의 성능이 가장 좋았다. 이 모델을 활용하여 6개의 건강 데이터를 입력한 결과 당뇨병 여부를 정확히 예측하였다. 이와 같은 방법으로 우리의 건강 상태(혈당, 혈압, 인슐린, BMI 등)을 측정하여 인공지능 모델에 입력한다면 당뇨병을 예측할 수 있을 것이다.

[참고 문헌]

1. 이영준외 6인(2021). 고등학교 인공지능기초 지도서. 씨마스.
2. 임진숙외 3인(2021). 나는 오렌지로 데이터분석한다. 씨마스
3. 피마인디언당뇨병데이터. 캐글. https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database
4. 당뇨병뉴스. 헬스인 뉴스. http://www.healthinnews.co.kr

MEMO

13. 코로나19 확진자 수와 가장 관계성 있는 데이터는 무엇일까?

북삼중학교 교사 최 훈 주

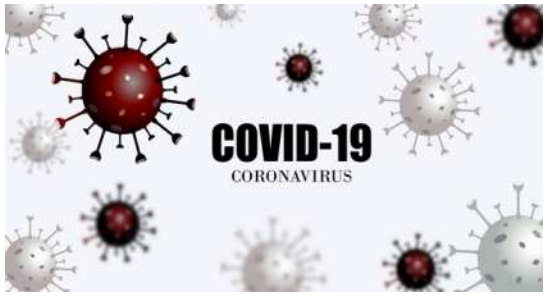
학습 진행 과정

1단계	데이터 준비	- 사용 데이터: 국가별 코로나19 확진자 수 데이터 - 수집: John Hopkins University github - 데이터 전처리: 국가별 지역 통합
2단계	데이터 불러오기	- 학습 데이터 불러오기 - 데이터의 속성별 Role(역할) 설정하기
3단계	데이터 시각화	- 시각화 도구을 이용한 데이터 탐색 - 사용한 시각화 도구: Scatter Plot, Line Plot, Bar Plot
4단계	세계지도에 데이터 나타내기	- 분석한 데이터를 세계 지도위에 나타내기 - 확진자 추이를 색으로 변화하며 애니메이션 나타내기 - 사용한 추가 위젯 : Geo, Timeseries
5단계	데이터 병합	- 추가 데이터: 국가별 HDI 데이터 - 수집: 오렌지3 자체 제공 - 데이터 전처리: 국가 이름 수정
6단계	데이터 관계성 찾기	- 데이터 시각화와 RANK로 주요 속성 추출하기 - 사용한 도구 : Scatter Plot, Rank
7단계	결론	- 코로나19 확진자 수와 가장 관계성 있는 데이터 찾기

학습 내용 요약

데이터 종류	문제 유형	사용된 학습 알고리즘
정형 데이터(수치형) 두 가지 데이터 병합	예측(회귀)	RReliefF

문제 상황



2019년 11월, 중국 후베이성 우한시에서 처음으로 발생하여 보고된 새로운 유형의 변종 코로나 바이러스에 의해 급성 호흡기 전염병이라 불리우는 코로나 19. 2021년이 된 현재도 우리는 계속되는 코로나 시대를 살아가고 있다.

코로나 19는 우리 삶에 수많은 변화를 불러 오게 되었다. 대표적으로 학교에서는 원격 수업이라는 새로운 형태의 수업을 진행되었으며 이는 교사와 학생 모두에게 신선한 충격을 가져다 줬으며 새로운 교육의 패러다임이 시작되었다. 이렇게 현재 우리 삶에 밀접한 영향을 끼치고 있는 코로나 19를 데이터 분석의 시점으로 들여다보자.

코로나 19로 인해서 발생하는 수 많은 데이터 중 확진자, 사망자, 완치자 수 등이 가장 큰 데이터 세트가 될 수 있을 것이다. 그렇다면 코로나 19 확진자 데이터를 가지고 전 세계적으로 확진자가 어떻게 변화하고 있는지 한 눈에 알아볼 수 있는 시각화 자료를 만들 수 없을까?

또한, 나라별로 어떠한 요소가 코로나 확진자 수에 영향을 끼치고 있는지 그 관계성을 오렌지3를 활용하여 살펴보고자 한다.

01 데이터 준비하기 - 국가별 코로나19 확진자 수 데이터 세트

실시간으로 코로나 확진자를 업데이트 하기 위해 데이터 세트를 가져온다. 실시간 코로나 확진자 데이터는 John Hopkins University github에서 제공하는 데이터 세트를 사용한다.

* John Hopkins University github의 코로나 19데이터 URL : https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data

아래 [그림 13-1]과 같이 전세계의 확진자, 사망자, 완치자 등의 실시간 데이터를 얻을 수 있다.

[그림 13-1] 코로나 확진자 수 데이터 다운로드 화면

전세계의 확진자 데이터를 가져와 다운로드 받는다.
time_series_covid19_confirmed_global.csv 파일에 들어가 RAW 데이터를 클릭한다.

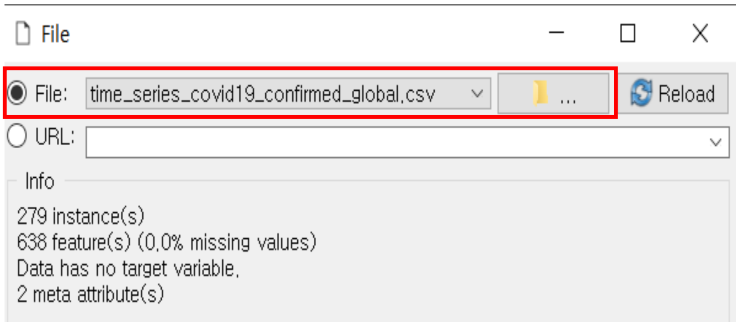
1 데이터 입력

File

위에서 찾은 데이터 세트를 오렌지3에 업로드하는 방법은 2가지이다.

① CSV 파일로 받아 업로드하는 방법.

1) Raw 파일이 열린 창에서 Ctrl + S를 눌러주면 파일로 저장할 수 있다.
2) 저장한 파일을 오렌지3의 file 위젯에서 업로드 시킨다.



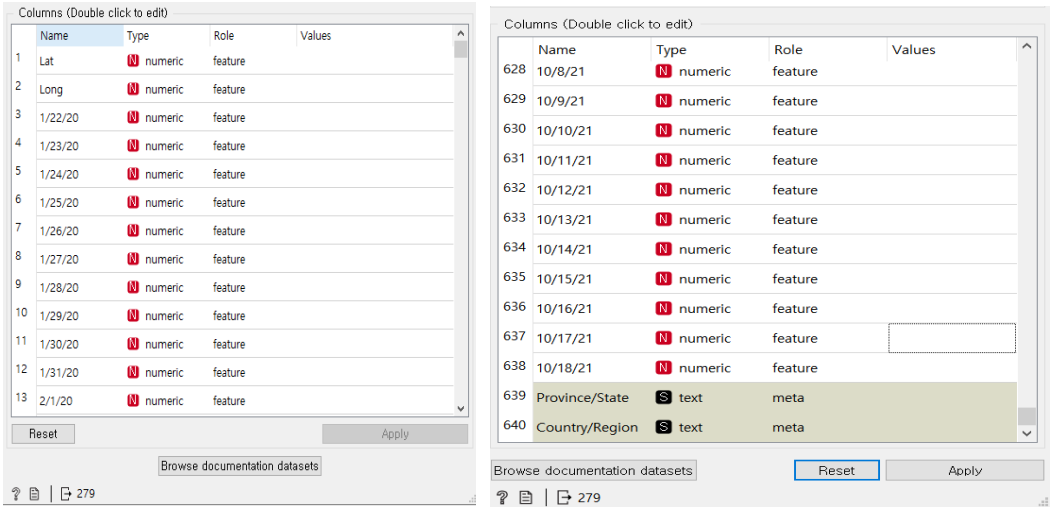
② github 링크로 바로 오렌지3에 업로드하는 방법.

1) Raw 파일이 열린 창의 url을 복사한다.
2) 오렌지3에서 file 위젯을 열어 url 부분에 붙여넣는다.

[그림 13-2] 코로나 확진자 수 RAW 데이터 클릭 화면

두 방법 모두 동일한 파일이 오렌지3에 입력된 것을 확인할 수 있다.

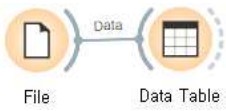
file의 Columns을 살펴보면 Lat과 Long이 상단에 보인다. 이것은 위도와 경도를 말한다. 두 개 다 numeric 즉 숫자 데이터로 이루어져 있고 Role(역할)은 feature이다.



[그림 13-3] File 속성 설정

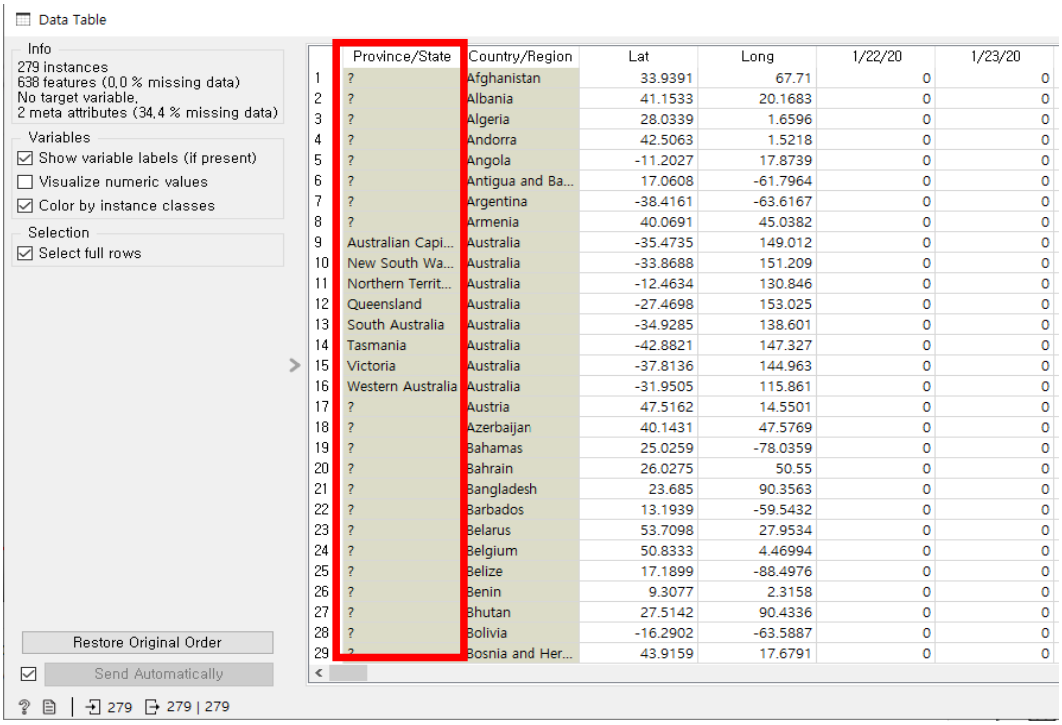
가장 하단에는 text(문자) Type의 meta 데이터를 확인할 수 있는데 이는 참고용으로 학습데이터로 사용되지 않는다.

2 데이터 출력



그렇다면 입력한 전세계 코로나 확진자 데이터를 Data Table을 활용하여 출력해보자.

다음의 Data Table을 보면 meta데이터로 설정되었던 Provice/State 열에 ?로 채워진 것이 보인다. 이는 아무런 데이터도 없어서 오렌지3에서 ?로 표현된 것이다. 자세히 보면 Australia가 여러 행에 걸쳐서 나눠져 있다. 큰 국가들은 지역을 더욱 나누어 표현한 상황이다.



	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20
1	?	Afghanistan	33.9391	67.71	0	0
2	?	Albania	41.1533	20.1683	0	0
3	?	Algeria	28.0339	1.6596	0	0
4	?	Andorra	42.5063	1.5218	0	0
5	?	Angola	-11.2027	17.8739	0	0
6	?	Antigua and Ba...	17.0608	-61.7964	0	0
7	?	Argentina	-38.4161	-63.6167	0	0
8	?	Armenia	40.0691	45.0382	0	0
9	Australian Capi...	Australia	-35.4735	149.012	0	0
10	New South Wa...	Australia	-33.8688	151.209	0	0
11	Northern Territ...	Australia	-12.4634	130.846	0	0
12	Queensland	Australia	-27.4698	153.025	0	0
13	South Australia	Australia	-34.9285	138.601	0	0
14	Tasmania	Australia	-42.8821	147.327	0	0
15	Victoria	Australia	-37.8136	144.963	0	0
16	Western Australia	Australia	-31.9505	115.861	0	0
17	?	Austria	47.5162	14.5501	0	0
18	?	Azerbaijan	40.1431	47.5769	0	0
19	?	Bahamas	25.0259	-78.0359	0	0
20	?	Bahrain	26.0275	50.55	0	0
21	?	Bangladesh	23.685	90.3563	0	0
22	?	Barbados	13.1939	-59.5432	0	0
23	?	Belarus	53.7098	27.9534	0	0
24	?	Belgium	50.8333	4.46994	0	0
25	?	Belize	17.1899	-88.4976	0	0
26	?	Benin	9.3077	2.3158	0	0
27	?	Bhutan	27.5142	90.4336	0	0
28	?	Bolivia	-16.2902	-63.5887	0	0
29	?	Bosnia and Her...	43.9159	17.6791	0	0

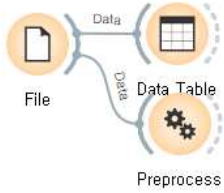
[그림 13-4] 코로나 확진자 수 Data Table

따라서 한 나라를 하나의 행에 만들어주기 위해 전처리 작업을 수행해야 한다. 전처리 작업은 오렌지3에서 할 수도 있고 직접 스프레드시트 파일에서 데이터들을 정리해 줄 수도 있다.

스프레드시트 파일에서 전처리 작업을 수행해보자.

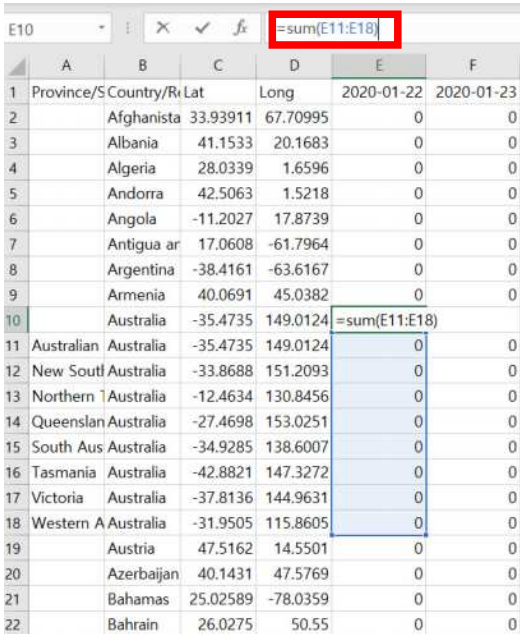
02 데이터를 전처리 하고 시각화 하자

1 데이터 전처리



[그림 13-4]와 같이 하나의 국가가 여러 지방으로 나뉘어져 있는 경우 해당 국가 전체에 대한 코로나 19 확진자를 확인할 수 없기 때문에 스프레드시트에서 Australia와 같이 여러 개의 주로 나뉘어진 국가에 대해서 모든 행의 합을 보여주는 하나의 행으로 바꾸어준다. 이때 행을 하나 더 삽입하여 해당 행에 sum 함수를 사용해주면 된다. 여러 개로 나뉘어진 다른 나라들도 하나로 만들어주어 데이터 전처리 작업을 완료한다.

- 전처리 작업 중 SUM 함수 사용 과정



E10 =sum(E11:E18)

	A	B	C	D	E	F
1	Province/S	Country/R	Lat	Long	2020-01-22	2020-01-23
2		Afghanista	33.93911	67.70995	0	0
3		Albania	41.1533	20.1683	0	0
4		Algeria	28.0339	1.6596	0	0
5		Andorra	42.5063	1.5218	0	0
6		Angola	-11.2027	17.8739	0	0
7		Antigua ar	17.0608	-61.7964	0	0
8		Argentina	-38.4161	-63.6167	0	0
9		Armenia	40.0691	45.0382	0	0
10		Australia	-35.4735	149.0124	=sum(E11:E18)	
11	Australian	Australia	-35.4735	149.0124	0	0
12	New Soutl	Australia	-33.8688	151.2093	0	0
13	Northern T	Australia	-12.4634	130.8456	0	0
14	Queenslan	Australia	-27.4698	153.0251	0	0
15	South Aus	Australia	-34.9285	138.6007	0	0
16	Tasmania	Australia	-42.8821	147.3272	0	0
17	Victoria	Australia	-37.8136	144.9631	0	0
18	Western A	Australia	-31.9505	115.8605	0	0
19		Austria	47.5162	14.5501	0	0
20		Azerbaijan	40.1431	47.5769	0	0
21		Bahamas	25.02589	-78.0359	0	0
22		Bahrain	26.0275	50.55	0	0

- 전처리가 완료된 스프레드시트 상황

Province/S	Country/R	Lat	Long	2020-01-22	2020-01-23	2020-01-24	2020-01-25	2020-01-26	2020-01-27
	Afghanista	33.93911	67.70995	0	0	0	0	0	0
	Albania	41.1533	20.1683	0	0	0	0	0	0
	Algeria	28.0339	1.6596	0	0	0	0	0	0
	Andorra	42.5063	1.5218	0	0	0	0	0	0
	Angola	-11.2027	17.8739	0	0	0	0	0	0
	Antigua a	17.0608	-61.7964	0	0	0	0	0	0
	Argentina	-38.4161	-63.6167	0	0	0	0	0	0
	Armenia	40.0691	45.0382	0	0	0	0	0	0
	Australia	-35.4735	149.0124	0	0	0	0	4	5
	Austria	47.5162	14.5501	0	0	0	0	0	0
	Azerbaijar	40.1431	47.5769	0	0	0	0	0	0
	Bahamas	25.02589	-78.0359	0	0	0	0	0	0

※ 또한 날짜의 형식이 월/일/연도의 형태로 되어 있으므로 년-월-일의 형태로 알아보기 쉽게 모습을 바꾸어 주었다.

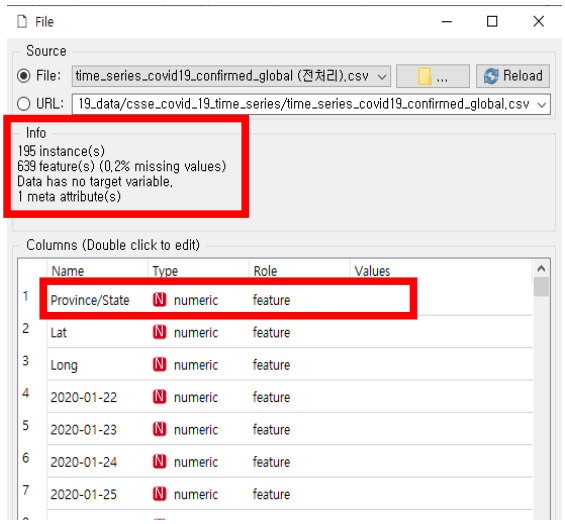
전처리를 완료한 후 다시 오렌지3에 전처리 완료한 스프레드시트 파일을 업로드 해보자.

- info를 보면 279 instans에서 195 instance로 데이터 개수가 줄어든 것을 확인할 수 있다. 행을 줄였기 때문에 데이터 개수가 줄어들었다.

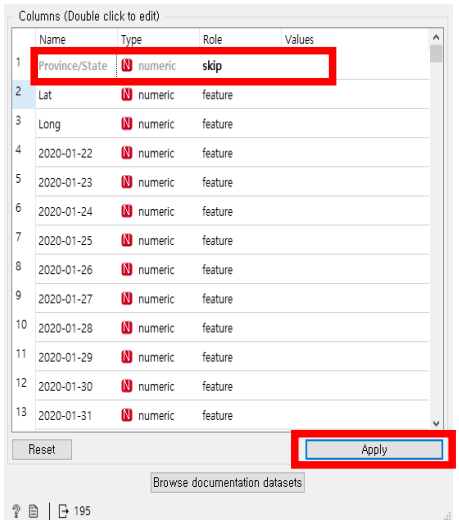
- contents를 보시면 Province/State 항목이 meta 데이터에서 numeric으로 바뀐 것을 확인할 수 있다. 이는 아무런 값도 들어가있지 않으므로 자체적으로 0으로 해석되었기 때문이다.

우리는 이번 데이터 분석에서 Province/State 항목을 사용하지 않을 것이므로 해당 항목의 Role을 skip으로 바꾸어 준다.

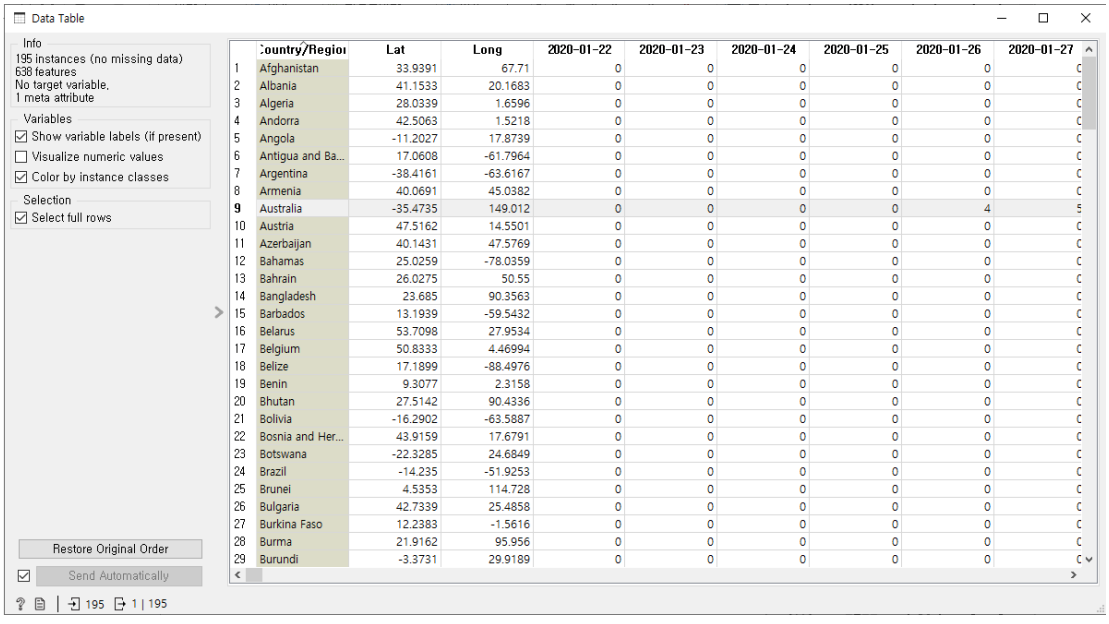
이렇게 전처리를 마치고 Data Table 출력을 다시 한 번 시켜보도록 한다.



[그림 13-5] File의 변경된 info



[그림 13-6] File 위젯에서 속성 수정

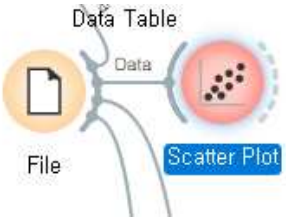


[그림 13-7] 전처리가 끝난 코로나 확진자 수 데이터의 Data Table

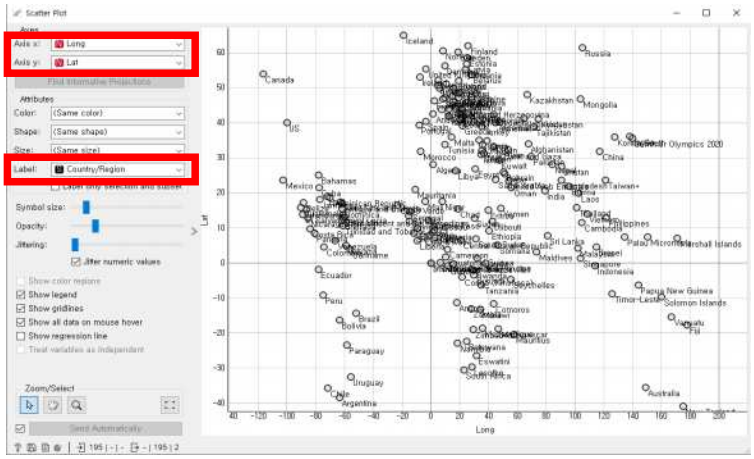
위의 [그림 13-7]과 같이 국가들이 모두 하나의 행에 적용되어 국가별 Lat(위도), Long(경도), 날짜 등으로 확진자의 추이를 확인할 수 있도록 수정되었다.

2 데이터 시각화

① Scatter Plot을 활용한 산점도 표현



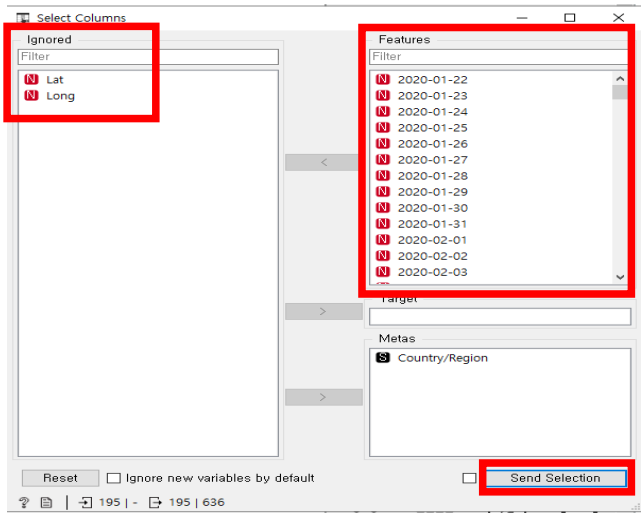
- Scatter Plot
  : 점을 이용하여 그래프를 그려주는 시각화 위젯



[그림 13-8] Scatter Plot으로 시각화

위도와 경도를 두 개의 축으로 설정해주면 그 값에 따라 각 나라들이 점으로 표현되게 된다. 위도, 경도 데이터를 통해 간단한 세계 지도를 표현할 수 있다. 해당 지도에서 우리나라 Korea를 찾아보자. 지도가 잘 그려졌는지도 확인해볼 수 있다.

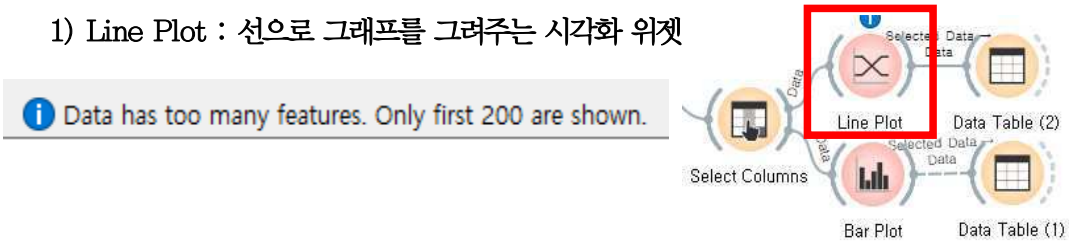
② 열을 선택하여 시각화



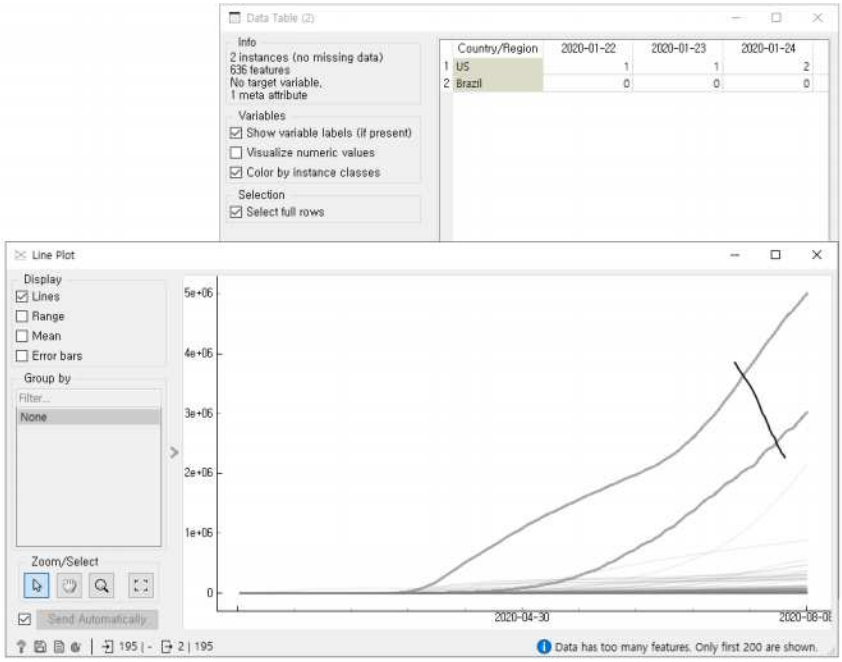
[그림 13-9] Select Columns 설정 내용

- Select Columns : 업로드 한 file에서 특정 열만을 선택하여 데이터를 분석할 수 있는 기능의 위젯
  위의 [그림 13-9]와 같이 Lat과 Long 열을 Ignored 즉 무시하는 열로 옮긴다. 그러면 Features들은 날짜들만 남게되고 해당 날에 국가별 코로나 확진자의 수만을 추출할 수 있다.
  Select Columns를 사용하여 국가별 확진자 추이를 2가지 방법으로 살펴보겠다.

1) Line Plot : 선으로 그래프를 그려주는 시각화 위젯



- Line Plot은 최대 200개의 features을 보여주는데 우리가 설정한 feature의 개수가 너무 많아 경고가 뜨고 있다.
  실제로 2020-01-22부터 2021-10-18까지의 features을 보여달라고 설정했지만 결과를 보면 2020-08-08까지밖에 보이지 않는다.
  아래의 [그림 13-10]과 같이 line plot에서 20년 8월 8일에 가장 확진자가 많은 나리 두 개를 선을 선택하면 line plot에서 이어진 data table에서 그 두 나라가 US와 Brazil임을 확인할 수 있다.



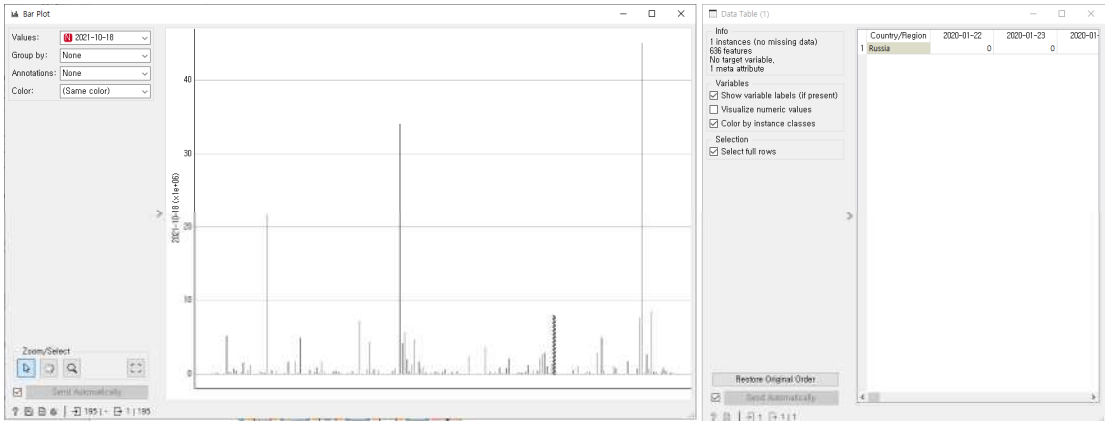
[그림 13-10] Line Plot 시각화

2) Bar Plot : 막대그래프를 그려주는 시각화 위젯

- bar plot은 [그림 13-11]과 같이 특정 날짜의 국가별 확진자수를 막대 그래프로 표현해준다. Annotations에 국가이름을 선택하여 가로축에 국가들의 이름을 볼 수 있지만 국가가 아주 많기 때문에 원활히 볼 수 없다.

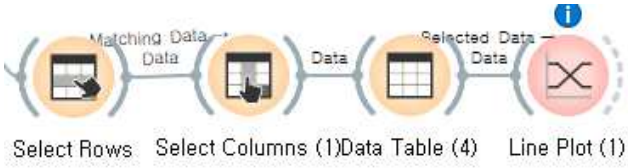
  따라서 bar plot도 data table과 묶어서 사용할 수 있다.

각 막대 그래프를 클릭하면 data table에서 어떤 나라인지 그 나라의 확진자 수를 표로 제공한다. 이 위젯은 몇 개의 행을 뽑아 일자 별 확진자 수를 비교할 때에 적절하다 할 수 있다.



[그림 13-11] Bar Plot 시각화

③ 열과 행을 모두 선택하여 시각화

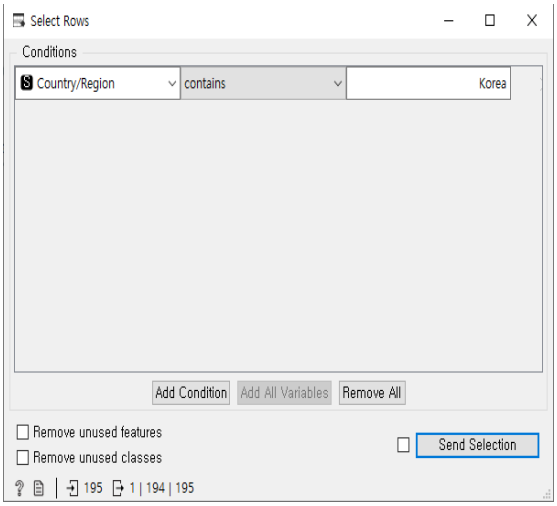


이번에는 한국의 확진자 변화를 살펴보기 위해서 열과 행을 모두 선택하여 Line Plot으로 시각화 해보도록 하자.

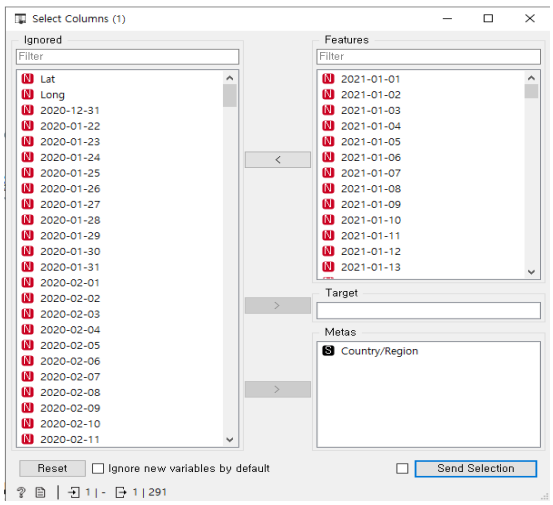
- Select Rows : 업로드 한 file에서 특정 행만을 선택하여 데이터를 분석할 수 있는 기능의 위젯

select rows는 여러 가지 조건으로 행을 선택할 수 있다. 우리는 South Korea를 찾기 위해 “Korea”라는 단어가 들어간 행을 추출하는 contains 조건을 사용하도록 한다.

select columns는 위와 같이 위도와 경도 feature을 ignored하도록 하겠다. 또한 feature를 줄여주기 위해 2021년도의 데이터만을 남기도록 한다.

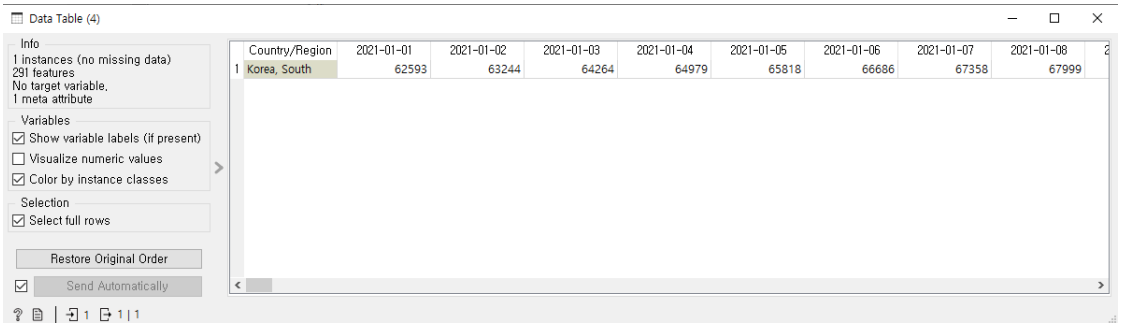


[그림 13-12] Select Rows



[그림 13-13] Select Columns

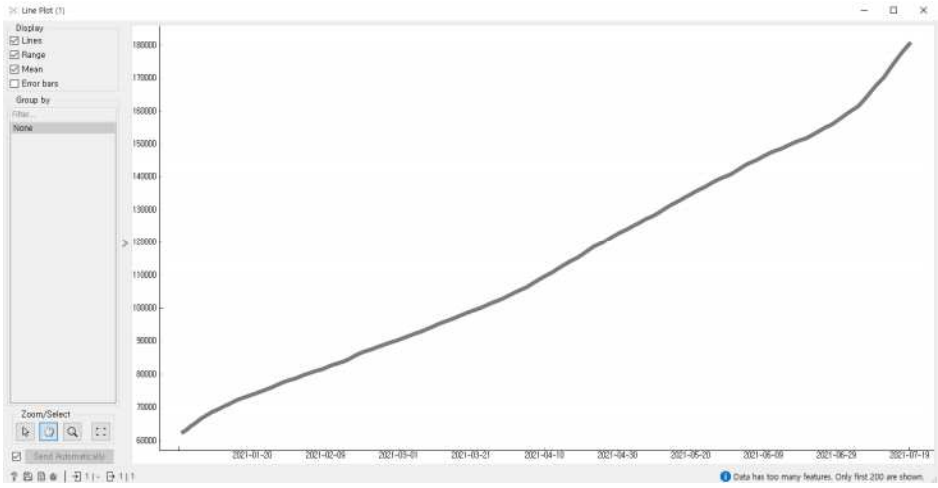
data table을 살펴보면 다음과 같다.



[그림 13-14] 변경된 Data Table

Korea, South 행만이 남은 것을 확인할 수 있고 Lat, Long은 사라진 후 2021년 1월부터 날짜가 시작되는 것을 확인할 수 있다. 이를 마지막으로 Line Plot으로 그래프를 확인해보자.

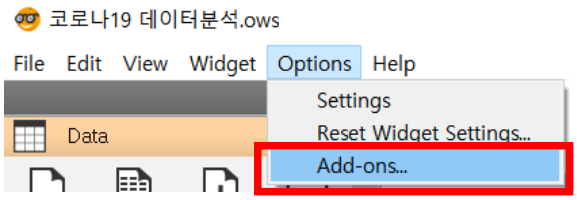
아래와 같이 21년도 들어와서는 가파른 상승세를 보이는 것을 확인할 수 있다.

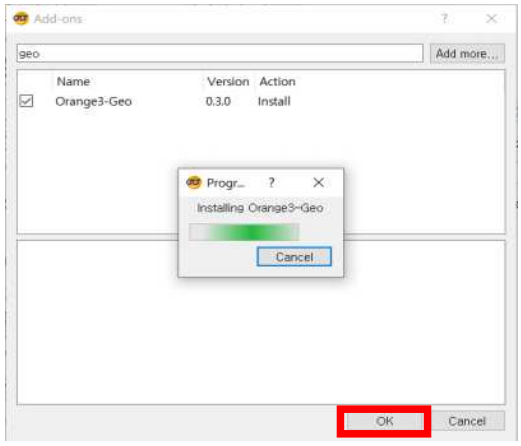


[그림 13-15] Line Plot 그래프 모습

3 데이터를 지도에 나타내기

이번에는 위도와 경도 데이터를 가지고 Add-ons에서 geo 기능을 추가하여 지도 맵 위에 데이터를 표현할 수 있도록 만들어본다.





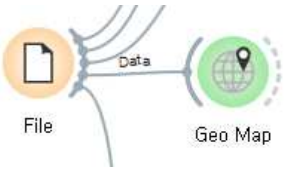
[그림 13-16] Add-ons에서 geo를 설치하는 모습

먼저 상단 Options에서 Add-ons를 클릭한다. 여기에서는 Orange3에서 제공하는 추가적인 기능들을 내가 선택해서 추가할 수 있다. 여기서 geo를 검색해보겠다. Orange3-Geo가 검색되어 나타났다. 이것을 체크하고 설치를 해보자.

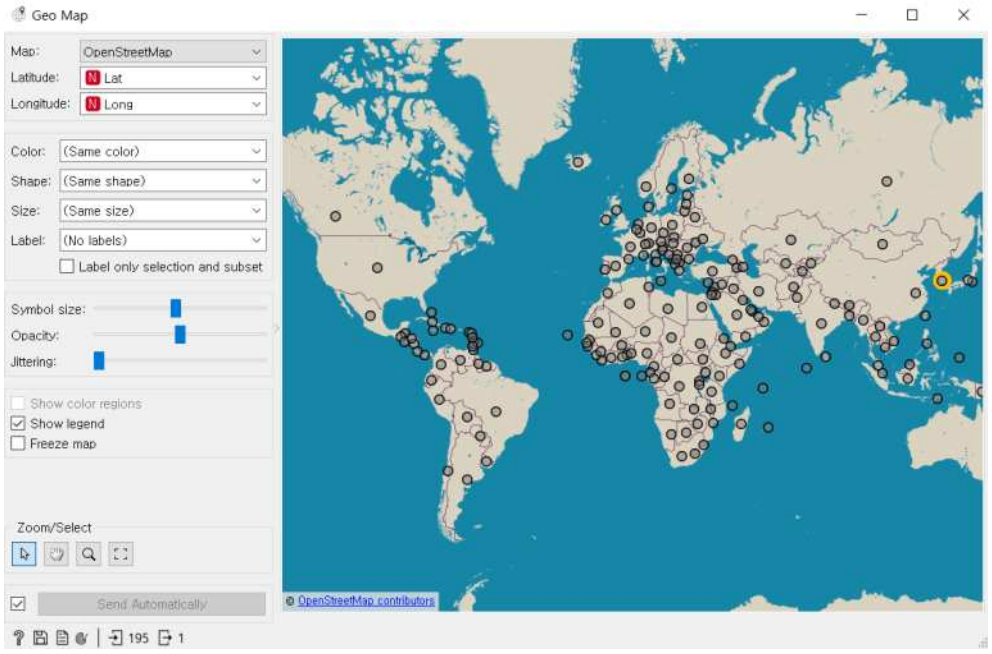
설치가 완료되면 다음과 같이 새로운 위젯탭이 생성된 것을 확인할 수 있다. Orange3에서 제공하는 지도맵 Geo를 추가하였다. 세 개의 위젯을 사용할 수 있다.

- Geocoding : 고유명칭(주소, 지명 등)을 위도, 경도의 좌표값으로 변환하는 위젯
- Geo Map : 세계지도에 데이터 점을 표시하는 위젯
- Choropleth Map : 통계 변수의 측정에 비례해 세계지도 위에서 데이터를 색깔로 표현해주는 위젯

① Geo Map

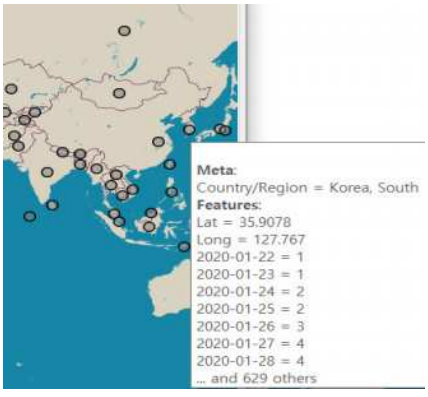


우리는 먼저 Geo Map에서 데이터를 점으로 표현해보도록 하겠다. [그림 13-17]과 같이 파일에서 Geo Map을 연결해준다.



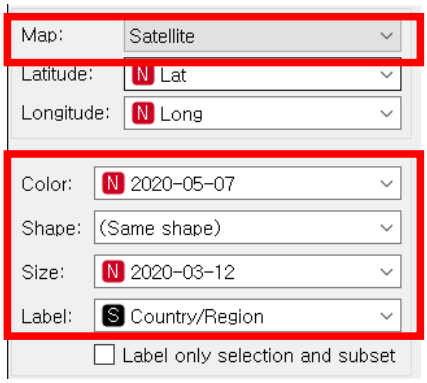
[그림 13-17] Geo Map

다음과 같이 자동으로 위도와 경도가 입력되어 세계지도 위에 점으로 데이터들이 표현되는 것을 볼 수 있다.



각 점들 위에 커서를 올려보면 다음과 같이 각 지역의 정보들이 보인다. 오른쪽 그림과 같이 한국에 갖다 대니 Korea,South로 정확하게 표시되는 것을 볼 수 있다.

왼쪽의 메뉴를 한번 살펴보겠다.



- Map에서 지도의 형태를 바꿀 수 있다.
- Color 에 날짜를 바꾸면 점의 색깔이 데이터의 값에 따라 달라진다.
- Size는 날짜에 따라 점의 크기가 달라진다.
- Label을 설정하면 지도에 나라의 이름이 표시되지만 지도가 작고 나라가 많이 겹쳐있기 때문에 잘 보이지 않게 된다. 이럴 때는 확대를 하면 잘 구분할 수 있다.

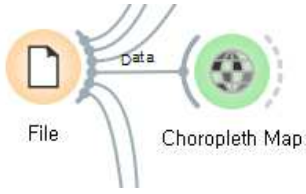
아래의 [그림 13-18]은 지도를 확대한 모습이다. 21년도에 들어와서 India에서 확진자가 많이 늘어났다는 것을 점의 크기와 색깔로 확인할 수 있다.

[그림 13-18] Geo Map 확대

Geo Map을 통해 점으로 데이터를 표현해 봤다. 하지만 코로나 19 확진자 데이터는 다음과 같이 점으로 표현했을 때 확진자 추이를 보기 쉽게 나타낼 수는 없음을 알 수 있었다.

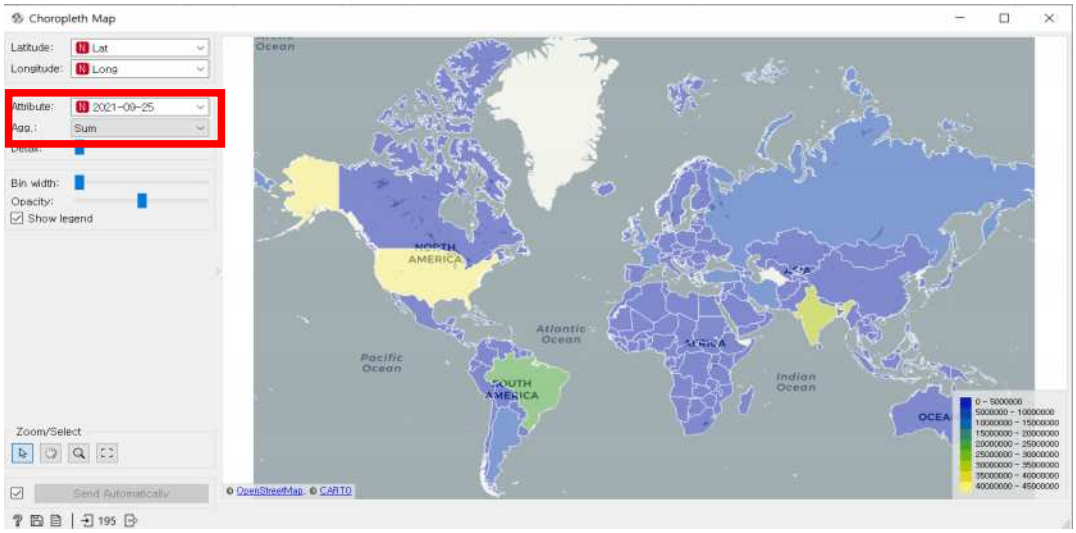
그래서 지도에서 색깔로 데이터를 나타내주는 Choropleth Map 위젯을 사용해보자.

② Choropleth Map



다음과 같이 Choropleth Map을 연결해보겠다. 아래의 [그림 13-19]와 같이 자동으로 위도와 경도가 적용되고 확진자 수가 색깔로 지도에 나타나는 것을 볼 수 있다. 여기서 날짜를

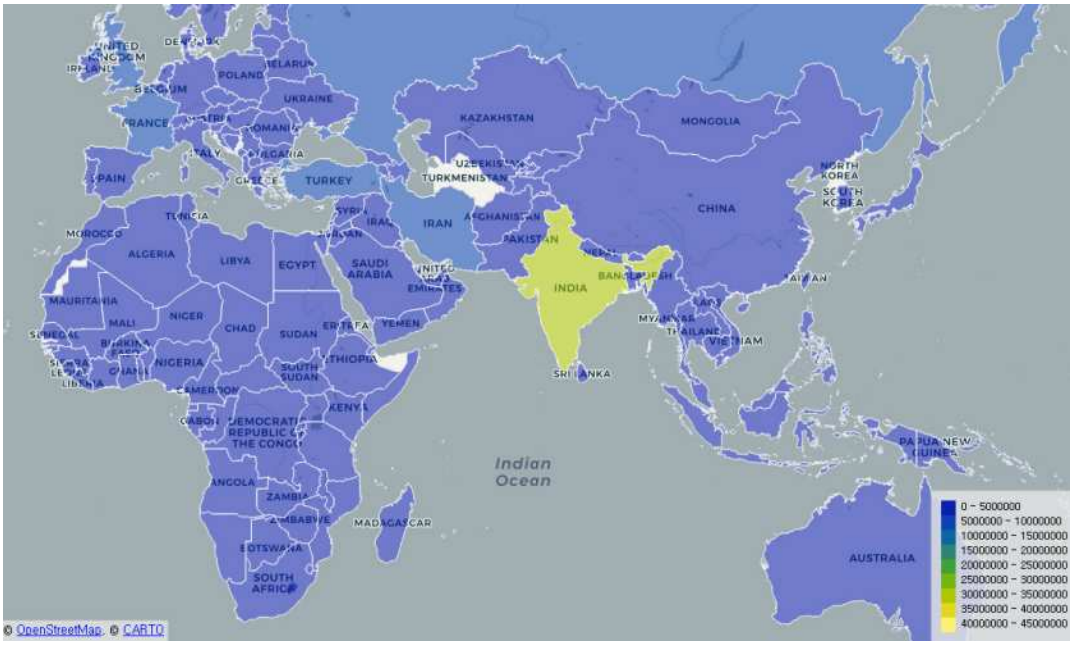
지정한 후 기능을 Sum으로 적용하였더니 해당 날짜까지의 누적 확진자가 지도에 다양한 색깔로 나타나게 된다.



[그림 13-19] Choropleth Map

2021년도 9월까지를 봤을 때, 눈에 띄는 확진자 수가 많은 지역은 미국, 브라질, 인도이다.

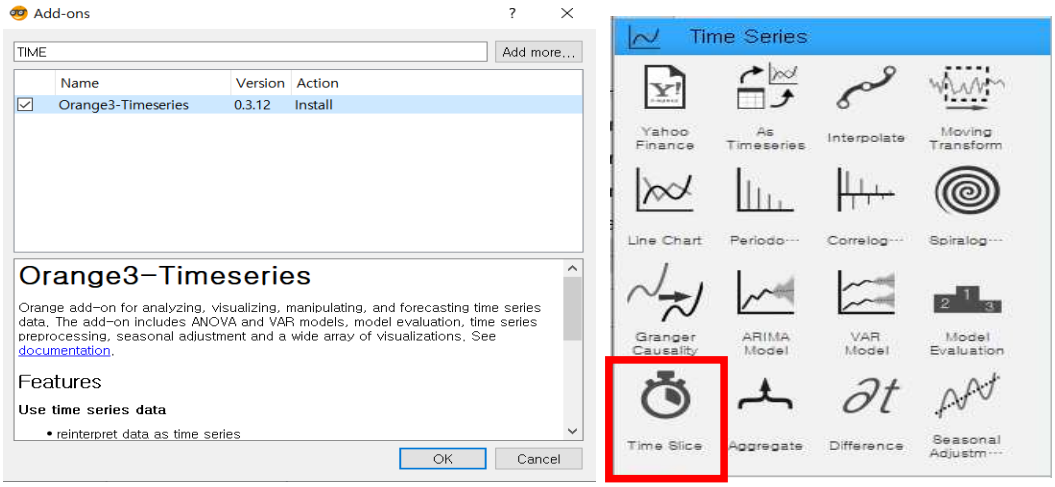
Choropleth Map에서도 마찬가지로 마우스 휠 또는 확대 기능을 통해 지도를 확대할 수 있다. 확대해서 살펴보면 아래의 [그림 13-20]처럼 각 나라가 모두 표시되어 있다.



[그림 13-20] Choropleth Map 확대한 모습

4 애니메이션 만들기

Timeseries를 이용하여 시간에 따라 변화하는 확진자 수를 애니메이션으로 지도 위에 나타내보겠다. 위의 geo를 추가했던 것처럼 Add-ons에서 Timeseries를 검색한다. 검색된 Timeseries를 설치해준다.

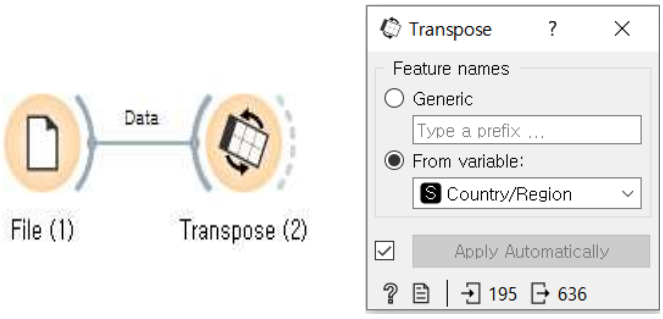


[그림 13-21] Timeseries 위젯 추가

설치하게 되면 위젯 탭에 다음과 같이 Time Series가 추가된다. 우리는 여기에서 Time Slice를 사용하겠다. Time Slice는 시간 간격으로 데이터를 선택할 수 있도록 해주는 위젯이다.

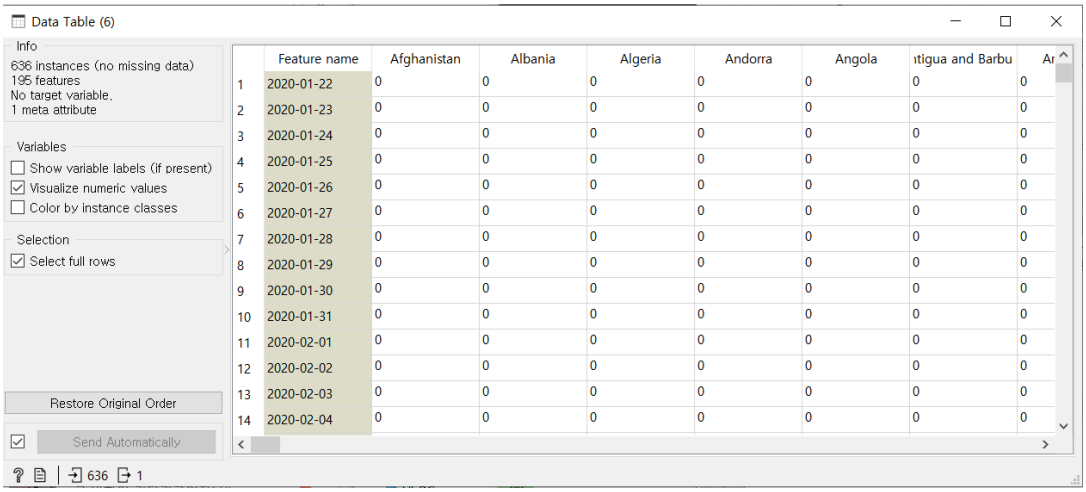
Time Slice는 행 형식의 시간 인스턴스가 필요하다. 그런데 우리가 가지고 있는 데이터를 살펴보게 되면 행은 지역, 열은 날짜로 구분되어 있다. 따라서 Time Slice를 사용하기 위해 우리는 데이터의 행과 열을 바꾸어주어야 한다.

이것을 바꾸어주기 위해 우리는 Transpose위젯을 사용한다. Transpose위젯을 연결한 뒤 파일에 값들이 feature(열)가 모두 날짜로 되어 있는지 확인해야 한다. 이 열들을 모두 행으로 바꿀 것이므로 다른 값이 feature에 들어가 있다면 meta로 바꾸어 준다.



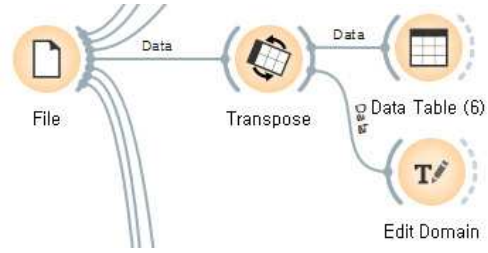
다음과 같이 Transpose 위젯을 연결하고 이동 할 행을 Country/Region으로 설정한다.

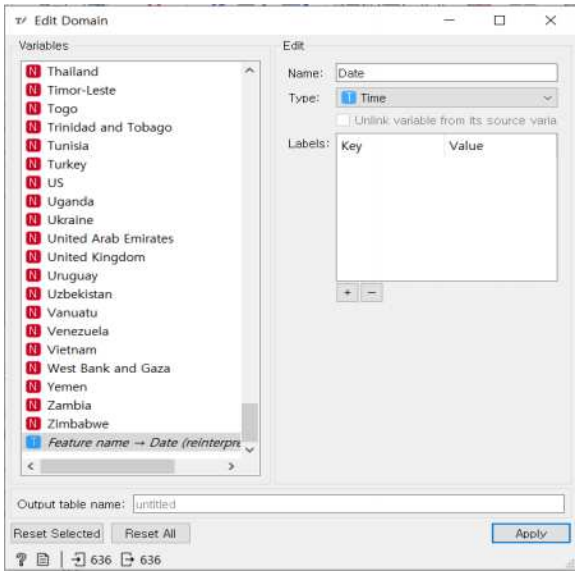
그리고 Data Table을 연결하여 정상적으로 행과 열이 바뀌었는지 확인해본다. Transpose를 한 뒤 데이터를 살펴보면 다음과 같이 행에는 날짜가 열에는 나라가 나열되어 행과 열이 치환된 것을 확인할 수 있다.



[그림 13-22] Transpose한 후 Data Table

현재 행이 날짜로 바뀌었지만 오렌지3에서는 날짜를 날짜 형식으로 인식하지 않고 텍스트 형식으로 인식하고 있다. 따라서 이것을 날짜 형식으로 바꾸기 위해 Edit Domain위젯을 사용한다.

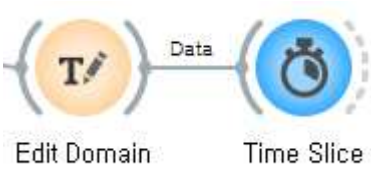


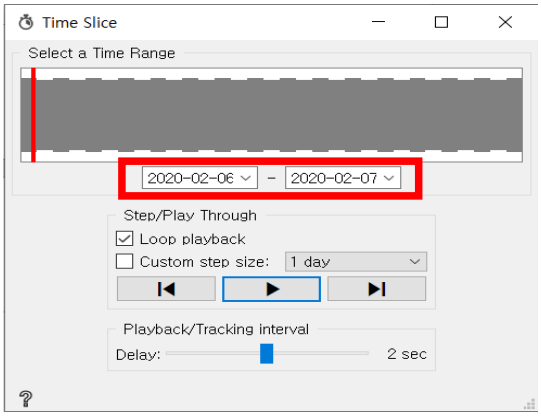


[그림 13-23] Edit Domain

다음과 같이 Transpose된 데이터 값을 Edit Domain으로 연결하여 Variables값 중에서 가장 아래의 feature name을 Date라는 이름으로 변경한 후 Type을 Date로 설정한다.

이렇게 행의 형식을 날짜 형식으로 수정하면 Time Slice를 연결하여 시간의 흐름을 만들 수 있다. 이제 Time Slice를 연결하겠다.



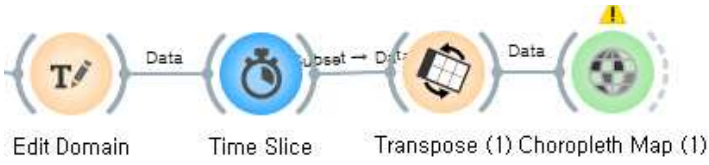


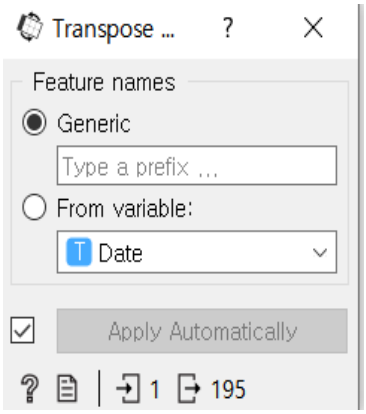
[그림 13-24] Time Slice

위 [그림 13-24]의 빨간 네모 안을 보면 시간 간격을 설정할 수 있다. 이것을 하루의 간격으로 설정해서 플레이를 눌러보면 하루 단위로 시간이 흘러가는 것을 볼 수 있다.

우리는 이것을 Choropleth Map에 연결하여 자동으로 시간이 흘러가도록 하여 누적 확진자 수 변화에 따라 세계지도에 표시된 색깔이 변화하도록 완성한다.

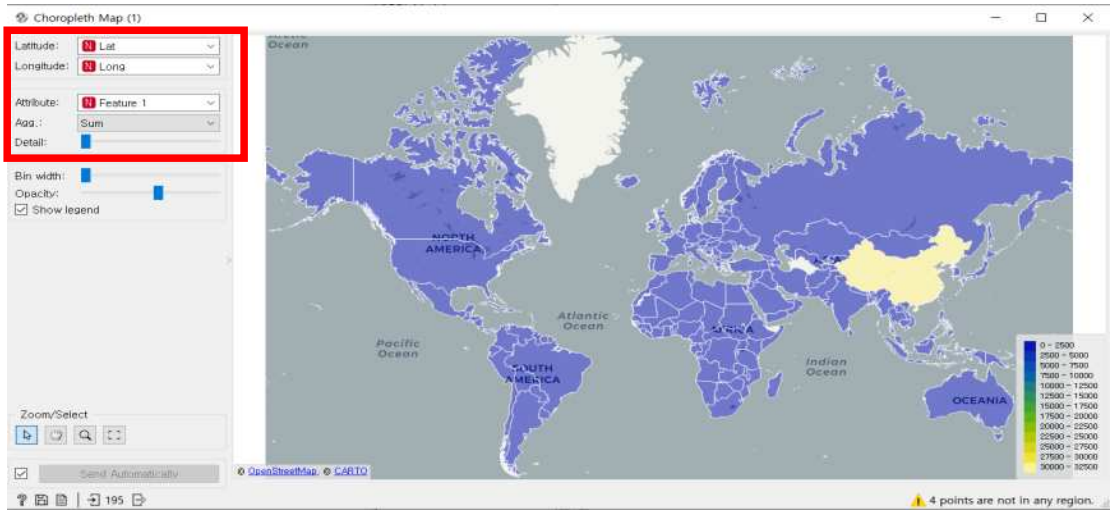
우리가 사용할 Choropleth Map은 위도와 경도가 필요하므로 다시 행과 열을 치환해 준 다음 연결해준다.





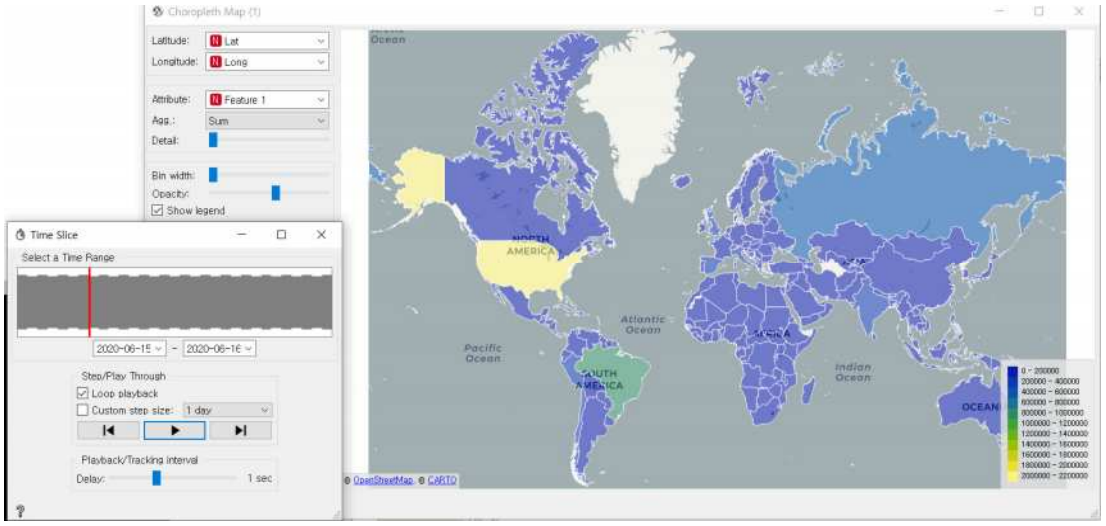
위도와 경도가 자동으로 들어가게 되고 사용할 데이터 값을 Feature 1로 기능은 Sum(누적값)으로 지정해준다.

이렇게 지도를 만들어두고 Time slice와 함께 창을 열어 시간의 흐름을 만들어준다.



[그림 13-25] Choropleth Map

* 시간이 흘러감에 따라 세계지도의 나라별 확진자 수가 변하는 것을 색깔변화로 확인할 수 있다.



[그림 13-26] Time slice에 시간이 흘러감에 따라 변화하는 Choropleth Map

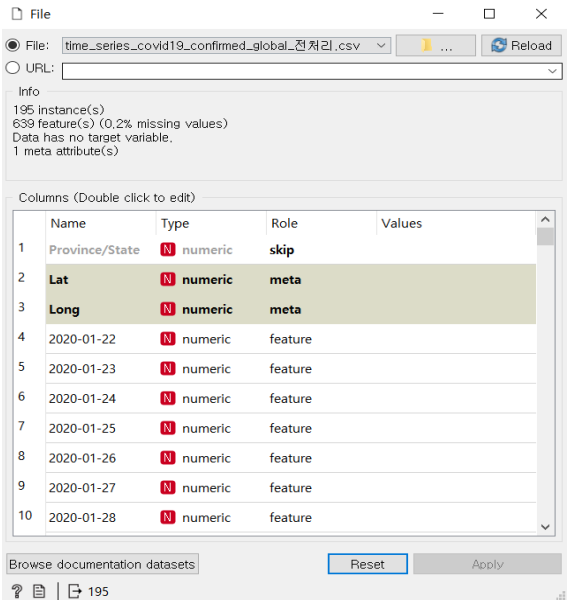
여기까지 우리는 코로나 확진자 수 데이터를 가지고 국가별 변화 과정을 애니메이션으로 보기 쉽게 표현해보았다. 다음 장에서는 단순히 시각화뿐만이 아닌 코로나 확진자 수 데이터와 국가별 HDI 데이터 간의 연관성을 찾아보도록 하자.

03 데이터를 분석하자

1 데이터 병합 후 연관성 살펴보기

이번 장에서는 2021년도 코로나 확진자 수를 다양한 지표와 병합하여 그 연관성을 살펴보려고 한다.

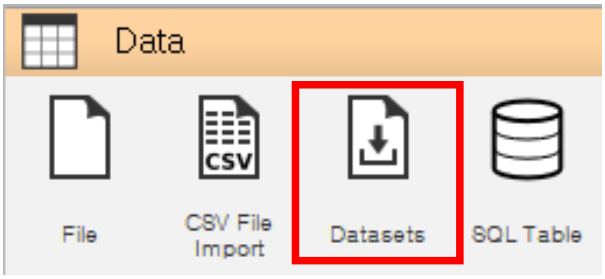
먼저 파일을 불러온 후 코로나19 확진자 수 데이터의 전처리 해두었던 파일을 불러온다.



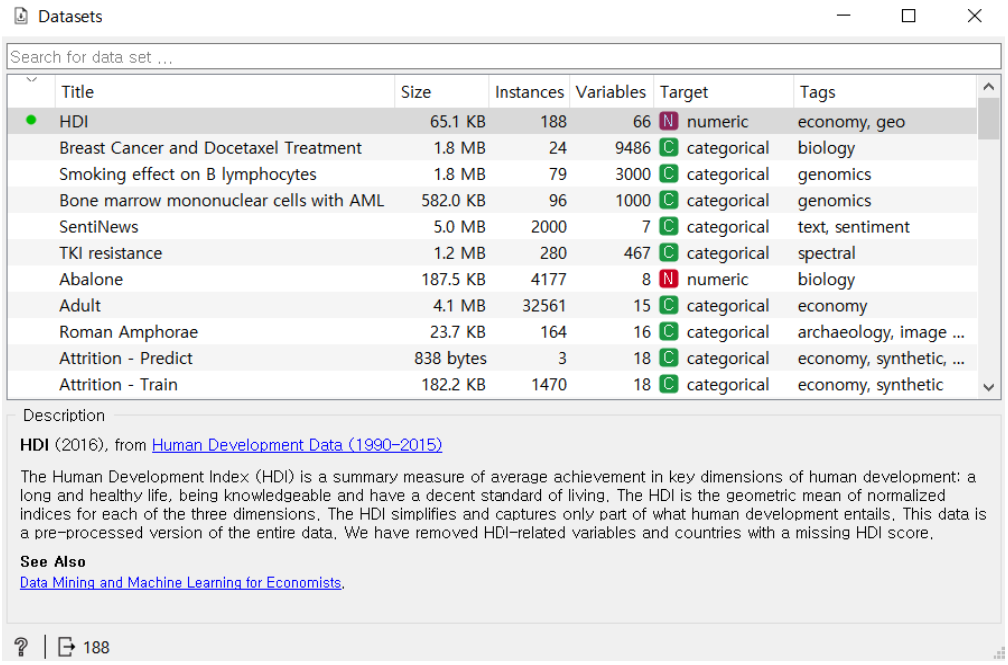
[그림 13-27] 코로나 19 확진자 수 데이터 File 속성

위의 [그림 13-27]과 같이 Province/State는 skip으로 Lat와 Long은 meta데이터로 설정한다.

Data 위젯에 보시면 Datasets이름의 위젯이 있다. 여기에는 오렌지3에서 제공하는 다양한 데이터들이 저장되어 있습니다. 어떤 것들이 있는지 한번 알아보자.



아래의 [그림 13-28]을 보시면 오렌지 3에서 제공하는 다양한 데이터 셋의 목록이 있다. 우리는 오늘 코로나 확진자 수와 전 세계의 HDI 지수를 병합해보도록 한다.



[그림 13-28] Datasets

* HDI : Human Development Index(인간 개발 지수)
  - 국제연합개발계획이 각국의 교육 수준 등을 조사해 인간 개발 성취 정도를 평가하는 지수

HDI 데이터 셋을 더블클릭하여 다운로드 받아준다. 위 사진의 하단에 보시면 HDI 데이터는 나라가 188개로 확진자 데이터의 195개 나라보다 적은 것을 확인할 수 있다. HDI 데이터 셋에 데이터 테이블을 연결하여 어떤 속성들로 구성되어 있는지 눈여겨 볼 만한 feature들로 정리해보겠다.

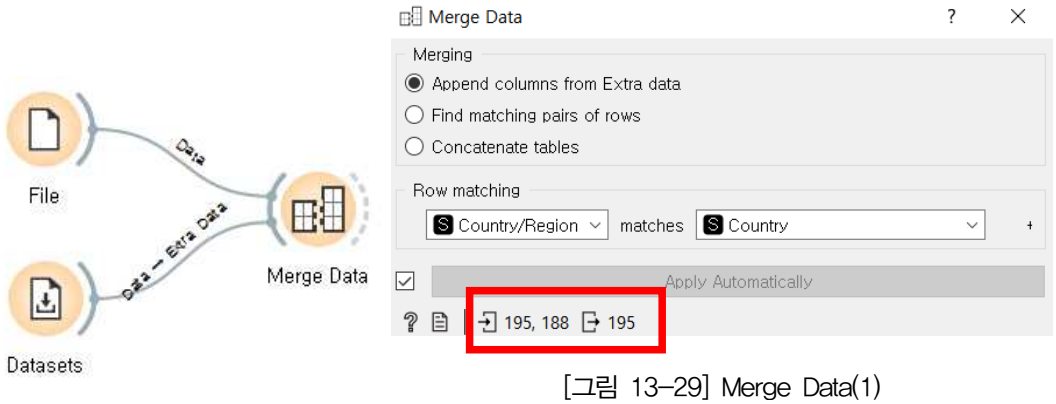
데이터 테이블을 확인하면 HDI 데이터는 66개의 속성들로 이루어져 있다. 아래의 표는 이 중에서 비슷한 속성들을 제외한 핵심 데이터 34가지를 정리하여 해석한 표이다. 국가별 HDI는 다음과 같은 속성들로 결정 된다.

[표 13-1] HDI 데이터의 속성

	속성	설명
1	Life expectancy	기대 수명
2	Mean years of schooling	평균 학력
3	Gross national income (GNI) per capita	1인당 국민총소득
4	Gender Development Index value	성 발달 지수 값

	속성	설명
5	Gender Development Index Group	성 발달 지수 그룹
6	Life expectancy at birth Female	출생 시 기대 수명 -여성
7	Life expectancy at birth Male	출생 시 기대 수명 -남성
8	Mean years of schooling Female	평균학력 -여성
9	Mean years of schooling Male	평균학력 -남성
10	Estimated gross national income per capita Female	1인당 추정 국민 총소득 -여성
11	Estimated gross national income per capita Male	1인당 추정 국민 총소득 -남성
12	Share of seats in parliament (% held by women)	여성의 의회 의석 비율
13	Population with at least some secondary education % (2005-2015) Female	최소한 중등 교육을 받은 인구 비율 -여성
14	Population with at least some secondary education % (2005-2015) Male	최소한 중등 교육을 받은 인구 비율 -남성
15	Labour force participation rate (% ages 15 and older) Female	여성의 경제활동참가율 (15세 이상)
16	Total Population (millions) 2015	총 인구 (백만단위) 2015
17	Population Average annual growth 2000/2005 (%)	인구 평균 연간 성장률
18	Dependency Ration Young age (0-14) /(per 100 people ages 15-64)	젊은 연령 부양비
19	Total fertility rate (birth per woman) 2000/2005	합계 출산율
20	Infants exclusively breastfed (% ages 0-5 months) 2010-2015	모유만 먹는 영유아 비율
21	Child malnutrition Stunting (moderate or severe) 2010-2015	아동 영양실조
22	Mortality rates Infant (per 1,000 live births) 2015	유아 사망률 2015
23	Deaths due to Malria (per 100,000 people)	말라리아로 인한 사망자
24	HIV prevalence, adult (% ages 15-49)	성인의 인체면역결핍바이러스 발병률
25	Physicians (per 10,000 people) 2001-2014	의사 수
26	Public health expenditure (% of GDP) 2014	공중 보건 지출
27	Unemployment Youth not in school or employment (% ages 15-24) 2010-2014	학교에 다니지 않거나 취업하지 않은 청년 비율
28	Child labour (% ages 5-14) 2009-2015	아동 노동 비율
29	Working poor at PPP$3.10 a day (%) 2004-2013	하루 3.1달러 이하로 생활하는 극빈층 비율
30	Internet users	인터넷 사용자
31	Coefficient of human inequality	인간 불평등의 계수
32	Inequality in life expectancy (%) 2010-2015	기대 수명의 불평등 비율
33	Inequality in education(%)	교육 불평등 비율
34	Income inequality (Quintile ratio) 2010-2015	소득 불평등 (5분위수 비율)

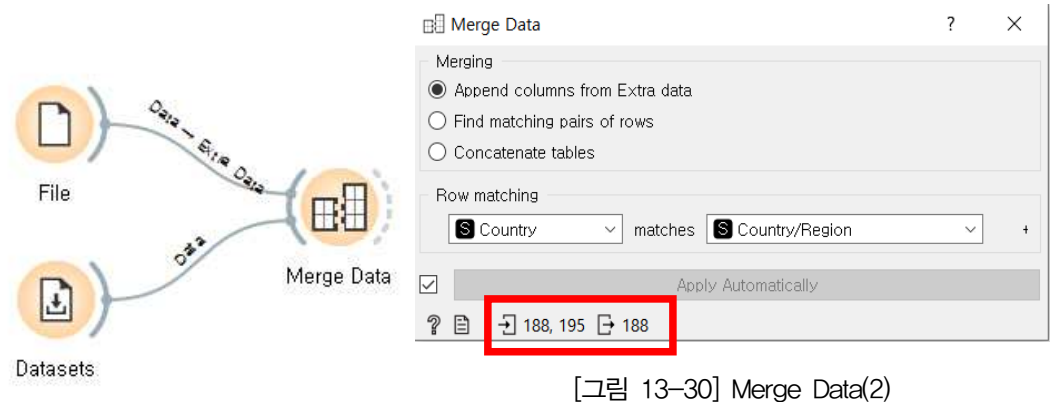
이 두 개의 데이터를 병합해보도록 한다.



[그림 13-29] Merge Data(1)

다음의 그림과 같이 두 데이터 셋을 Merge Data로 연결한다. 그러면 Datasets에서 나온 화살표에는 Data->Extra Data로 연결된 것을 볼 수 있다. 확진자 수는 있지만 HDI 데이터가 없는 나라를 함께 병합하게 되면 의미가 없으므로 없는 나라들을 그냥 없애도록 한다. File과 Datasets를 Merge Data로 연결하는 순서를 바꿔주면 된다.

Datasets에서 먼저 Merge Data로 연결해주고 그 다음 File에서 Merge Data를 연결한다.



[그림 13-30] Merge Data(2)

Merge Data를 더블 클릭하여 추가된 데이터를 열 수를 늘려 추가한다는 첫 번째 항목을 선택한다. 하지만 순서를 바꾸었기 때문에 매칭되지 않는 나라는 삭제된다. 하단의 매칭 후 데이터를 보면 출력 데이터의 수가 188개이다. Row matching은 File 데이터의 Country/Region과 Datasets의 Country를 연결하겠다고 설정해준다. 그러면 이름이 같은 나라들이 자동으로 매칭된다.

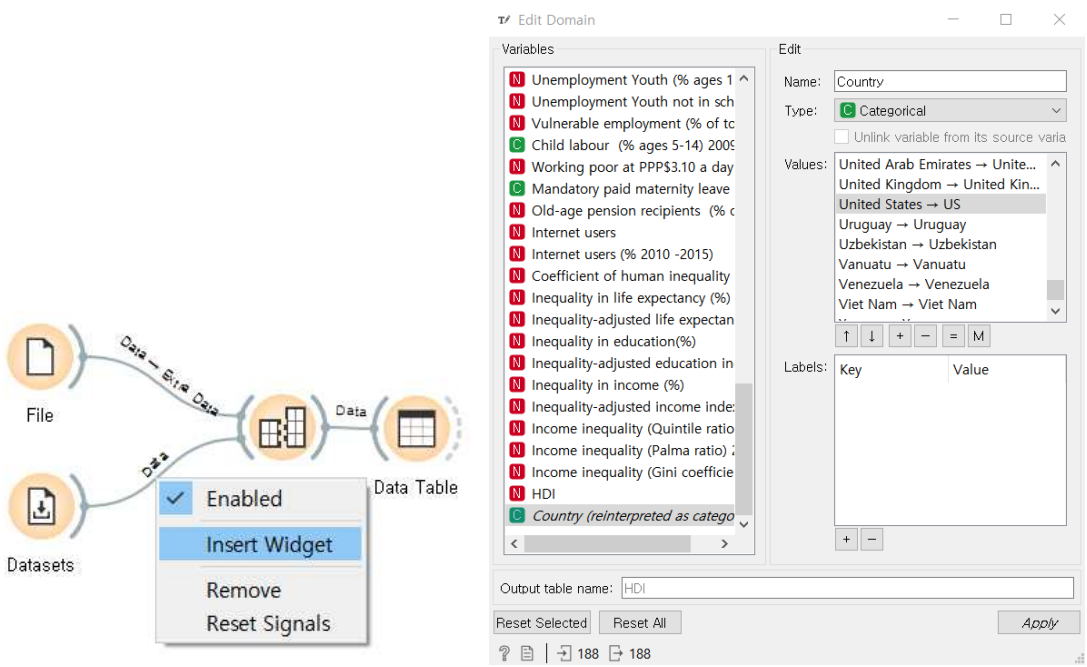
Data Table을 연결시켜 값이 제대로 잘 들어갔는지 확인해보자.

아래의 [그림 13-31]을 보시면 HDI 지수와 HDI를 결정하는 다양한 지표들의 데이터 뒤로 코로나 확진자 수 데이터가 붙어있는 것을 확인할 수 있다.

그런데 문제는 비어있는 데이터가 발생하게 된다. 아래의 [그림 13-31]에서도 United States와 Hong Kong 같은 경우에 위도 경도가 비어있고 코로나 확진자 수도 비어있는 것을 확인할 수 있다.

[그림 13-31] 두 데이터 병합 후 Data Table

이렇게 된 이유는 바로 나라 이름이 HDI 데이터에서는 United States지만 코로나 확진자 수 데이터에서는 US로 다르기 때문이다. Merge Data위젯의 경우 단순히 텍스트 비교를 해서 병합을 하기 때문에 이런 경우 다른 나라로 인식을 하게 되었다. 따라서 나라 이름을 바꾸어 준다.



[그림 13-32] Edit Domain

[그림 13-32]와 같이 Datasets에서 오는 연결선에 위젯을 추가해준다. 추가하는 위젯은 Edit Domain이다. Country 칼럼을 바꿀 것이기 때문에 왼쪽 Variables 항목에서 Country를 선택해 주시고 칼럼 타입을 Categorical로 바꿔준다. 항목의 이름을 바꾸는 방법은 변경할 나라 이름을 클릭하시고 변경하고픈 이름을 적어주시면 된다.

변경해야 하는 나라들은 다음 표와 같다.

[표 13-2] 수정해야 하는 나라 이름

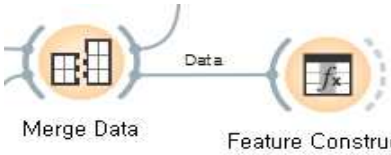
변경 전	변경 후
Antigua and Barb.	Antigua and Barbuda
Bosnia and Herz.	Bosnia and Herzegovina
Brunei Darussalam	Brunei
Cape Verde	Cabo Verde
Congo	Congo (Brazzaville)
Czech Rep.	Czechia
Cote d'Ivoire(억양표시 있음)	Cote d'Ivoire
Dem. Rep. Congo	Congo (Kinshasa)
Dominican Rep.	Dominican Republic
Korea	Korea, South
Leo People's Democratic Republic	Laos
Macedonia	North Macedonia
Myanmar	Burma
Palestine, State of	West Bnak and Gaza
Russian Federation	Russia
Sao Tome and Principe (억양표시 있음)	Sao Tome and Principe
St. Kitts and Nevis	Saint Kitts and Nevis
St. Vin. and Gren.	Saint Vincent and the Grenadines
Swaziland	Eswatini
Syrian Arab Republic	Syria
United States	US
Viet Nam	Vietnam

전체 변경한 뒤 Data Table을 다시 확인해보자. 아래의 [그림 13-33]과 같이 누락되는 값이 ?로 표시되지 않고 나라별로 잘 매칭된 것을 확인할 수 있다.

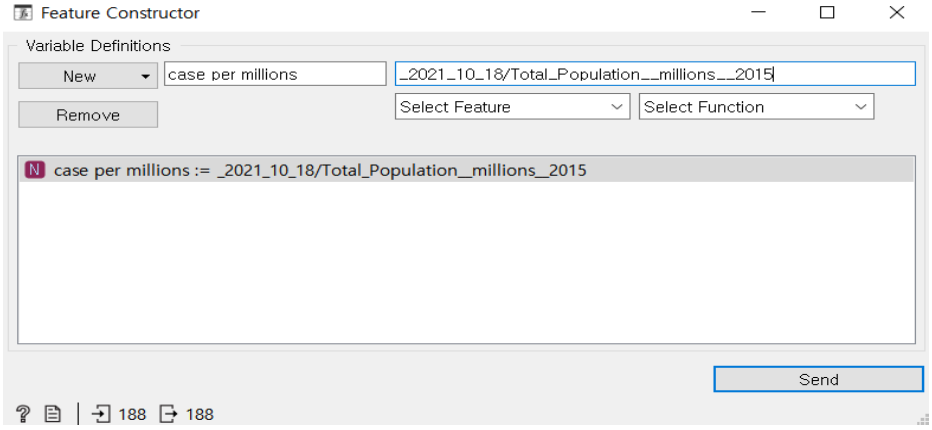
[그림 13-33] 나라 이름 통일 후 Data Table

이제, 관계성을 확인해 보도록 한다.

① 전체 인구수와 코로나 확진자 수의 관계



Feature Constructor 위젯을 추가하여 가장 최근 날짜의 확진자 수와 나라별 전체 인구의 관계성을 예측해보자.



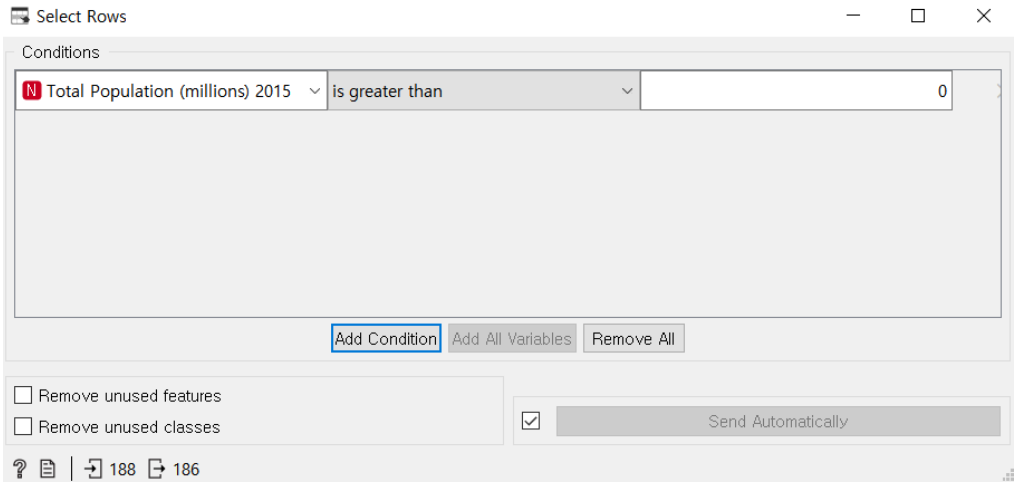
[그림 13-34] Feature Constructor

New 버튼을 눌러 feature를 Numeric으로 설정해준다. 이제 만들고자 하는 feature의 식을 설정해주면 되는데 우리는 가장 최근 날짜인 10월 18일 확진자수를 전체 인구수 (현재 오렌지3의 HDI데이터로는 2015년도 인구수가 가장 최근이다.)를 나눠준다.

ZeroDivisionError: float division by zero

Send를 누르면 위의 그림과 같은 오류가 발생하게 된다. 바로 인구수가 0명인 나라가 있어 0으로 나누게 되는 zerodivision 오류이다. 이를 방지하기 위해 인구수가 0명인 나라를 미리 제거해 준다.

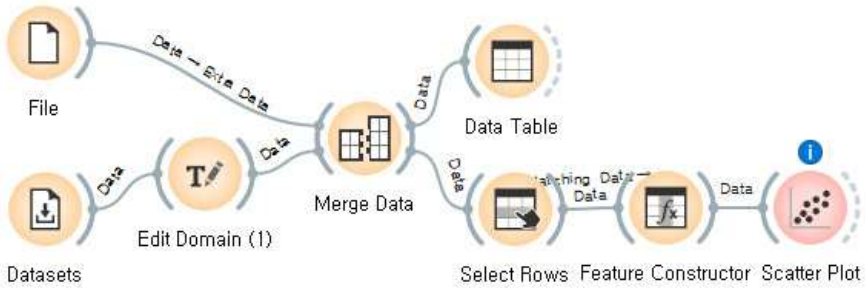
다음과 같이 Feature Constructor로 들어가기 전 병합 데이터에서 인구수가 0명인 나라를 걸러주기 위해 Select Rows 위젯을 추가한다.

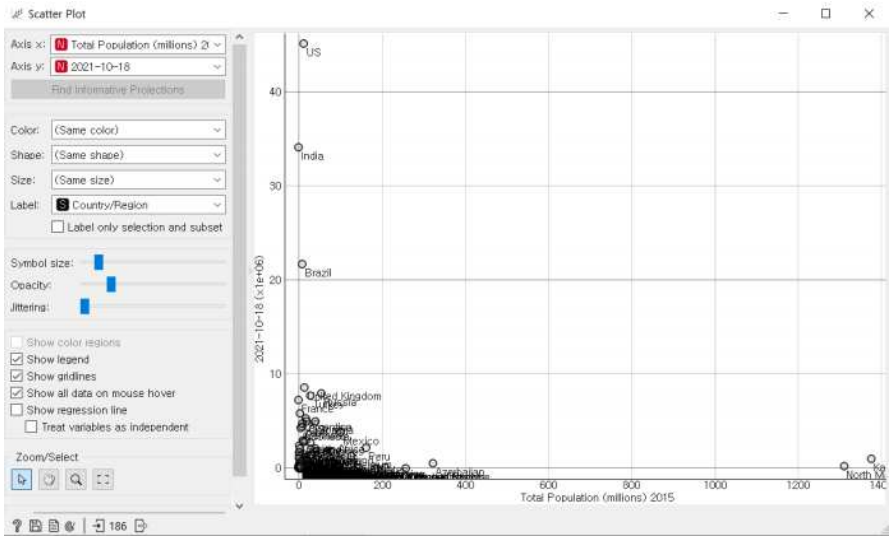


[그림 13-35] Select Rows

위의 [그림 13-35]와 같이 Total Population 항목을 0 이상인 것만 선택하도록 한다. 하단의 데이터 출력 개수를 확인하면 2개의 나라가 빠진 것을 볼 수 있다.

이제 산점도로 관계성을 살펴본다.



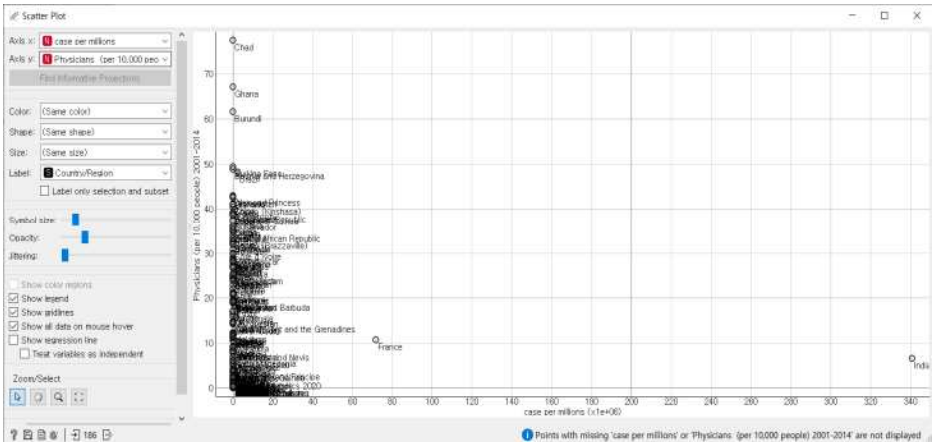


[그림 13-36] Scatter Plot

위의 [그림 13-36]에서처럼 전체 인구수에 따른 최근 코로나 확진자 수의 관계는 높지 않은 것으로 드러났다. 오히려 양극화되어 미국은 인구수가 많지 않음에도 불구하고 가장 높은 확진자 수를 보이고 있고 중국 같은 경우 인구수는 가장 많지만 코로나 확진자 수는 미국이나 인도보다 현저히 낮다.

물론 이것은 최근의 확진자 수를 기준으로 하였기 때문에 이러한 결과가 나타난다. X축과 Y축의 내용을 변경하여 다른 관계성도 알아볼 수 있다.

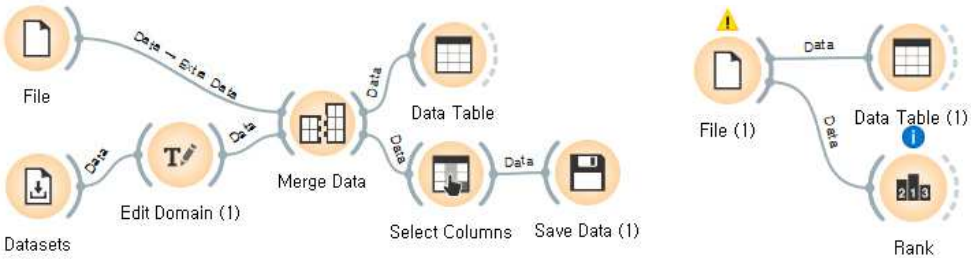
예를 들어, 우리가 생성한 인구수별 확진자 수와 국내 의료진의 명수를 비교해보자.



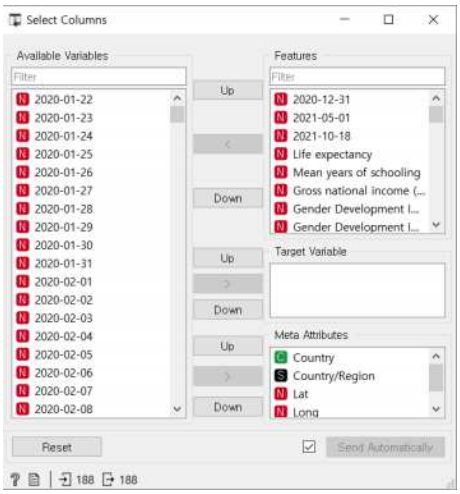
[그림 13-37] 코로나 확진자 수와 국가별 의료진 상황 분석 그래프

다음과 같이 의료진의 수가 적어서 코로나 확진자가 많이 걸렸다라고 볼 수 있는 나라는 인도뿐임을 확인할 수 있다.

② 코로나 확진자 수와 HDI 항목 간의 Rank



그렇다면, 어떤 항목이 코로나 확진자 수와 가장 관계가 있는지 오렌지3에서 제공하는 RReilefF 알고리즘으로 알아보도록 하자.



[그림 13-38] Select Columns

[그림 13-39] File 속성

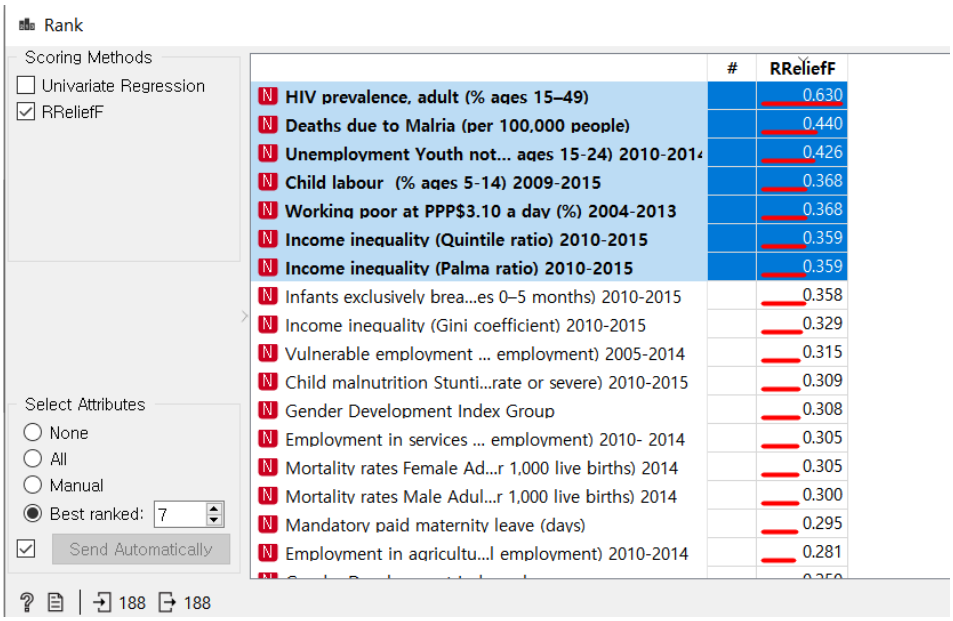
다음과 같이 내가 원하는 Column만을 선택해 준다. 저는 초기와 중기 후기의 확진자 변화를 알기 위해 20년 12월 31일, 21년 5월 1일, 21년 10월 18일 세 날짜와 함께 HDI 지표 데이터를 뺀 나머지 확진자 데이터를 모두 뺀다.

이렇게 선택한 데이터들을 새로운 파일로 저장한 후 새 파일에서 불러온다. 다른 날짜의 확진자 수를 Skip으로 가장 최근의 확진자 수를 Target으로 설정한다.

데이터 테이블을 연결시켜 데이터들이 제대로 들어왔는지 확인해본다.

* RReliefF : Rank 위젯에서 각각의 변수들 간의 상대적 거리에 대한 순위를 표시하는 것으로 RReliefF라는 지표가 있다. 회귀모델에서, RReliefF 지표는 Kononenko가 1994가 발표했으며 특징 선택(feature selection) 시 해당 feature 선택에 대한 적절함을 암시하며 알고리즘이 좋은 성능과 견고성을 가지고 있다고 알려져 있다. 따라서 이 지표가 높으면 주어진 두 변수 간의 상대적 거리가 가깝다는 것을 나타낸다.

[그림 13-40] 날짜와 HDI 지표를 병합한 Data Table



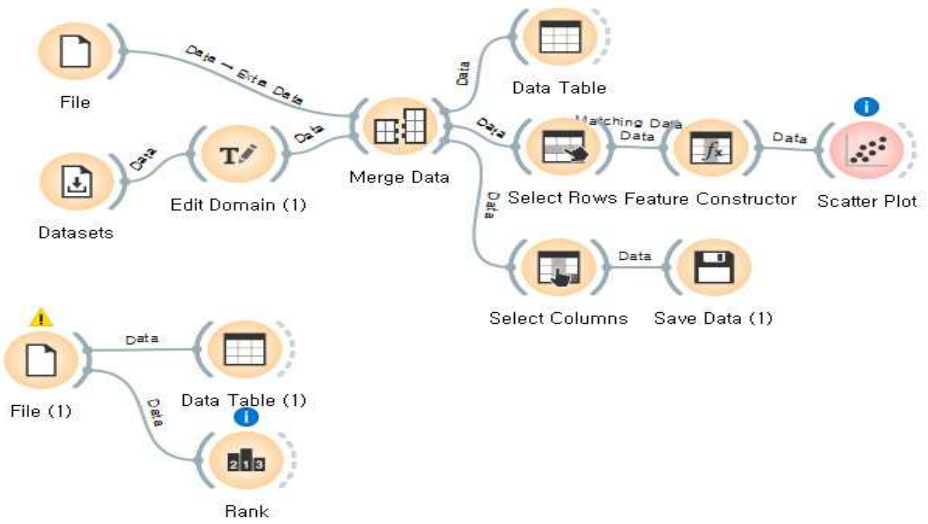
[그림 13-41] Rank 날짜와 HDI 지표를 병합한 Rank

위 [그림 13-41]을 확인해보면 가장 큰 상관도를 보이는 것이 바로 HIV prevalence이다. 즉, 인체면역결핍바이러스의 유행이 크게 있었던 나라에서 코로나 19 확진자 수가 크게 늘어날 가능성이 크다는 결과로 볼 수 있다. 또한 말라리아 감염으로 인해 죽은 사람의 수가 코로나 19와 연관성이 높게 나타나고 있다.

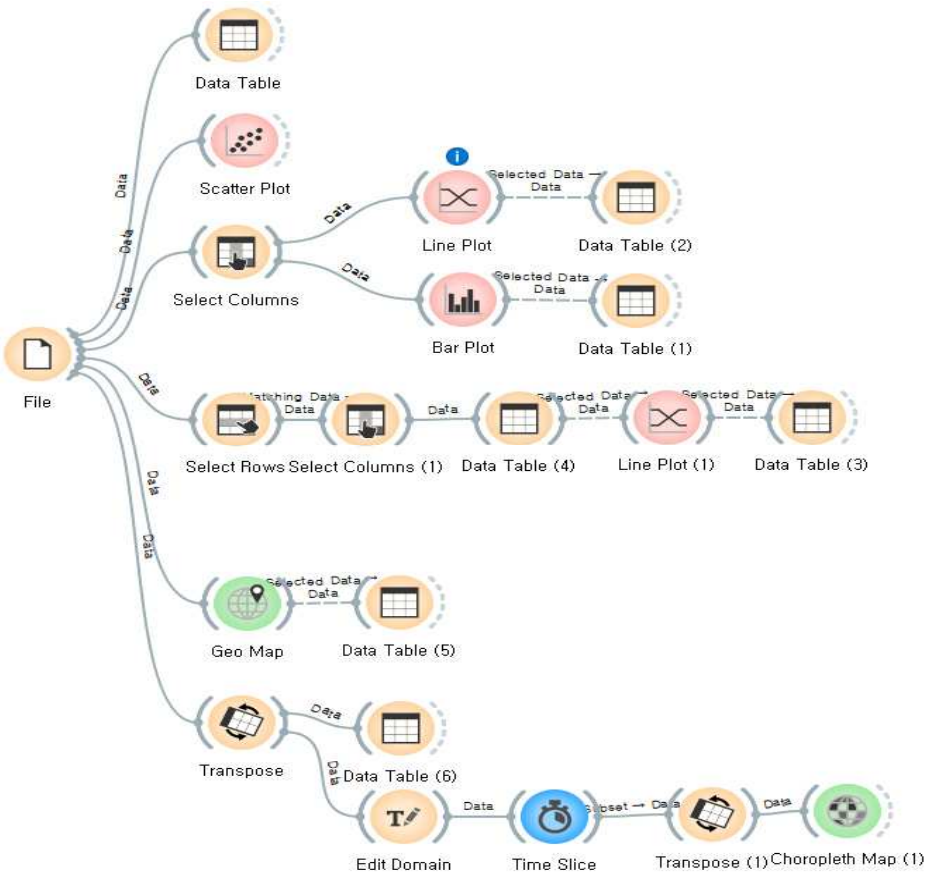
이 두 지표는 모두 국가의 위생환경과 관련이 있는 것으로 예측된다. 이외에도 다양한 연관성 랭크들이 있고 RReilefF를 내림차순으로 하면 순위를 확인할 수 있다.

이 외에도 청소년 실업률, 아동 노동률, 하루 3.1달러 이하로 생활하는 극빈층 비율, 소득 불평등 비율 등이 코로나19 확진자 수와 연관이 있는 데이터로 나타났다.

* 전체 오렌지3 흐름도



[그림 13-42] 코로나19 확진자 수와 국가별 HDI 관계성 확인을 위한 오렌지3 흐름도



[그림 13-43] 코로나19 확진자 수 데이터 시각화를 위한 오렌지3 흐름도

정리하기

인공지능은 데이터 분석을 통해 새로운 예측을 하거나 기존의 데이터를 군집화시켜 유용한 정보를 추출해 낼 수 있는 기술이다. 따라서 인공지능에서 데이터 수집 및 분석은 가장 중요한 단계가 된다. 잘 정돈된 데이터는 인공지능에 큰 영향을 줄 수 있기 때문이다. 이번 학습에서는 인공지능에서 쓰이는 데이터에 대해 심도 있게 분석해볼 수 있었다.

우리는 코로나 19 확진자 수 데이터를 가지고 오렌지3라는 데이터 분석 도구를 사용하여 확진자 수 변화를 시각화하여 살펴보고 7장에서와 같이 다른 데이터와 묶어 어떤 것이 코로나 19 확진자 수에 가장 많은 영향을 주는지를 찾아보았다. 이를 통해 성인의 인체면역결핍바이러스 발병률이 코로나 확진자 수와 연관성이 높은 것으로 확인되었다. 이러한 데이터 분석을 통해 얻어낸 자료는 실제로도 감염병을 예방하기 위해 사용되어질 수 있을 것이다.

더 나아가 John Hopkins University github에서는 국가별 확진자 수 뿐만 아니라 완치자와 사망자 수까지 얻을 수 있다. 이러한 데이터들을 가지고 어떤 예측을 할 수 있을지 찾아보는 것도 아주 좋은 공부가 될 수 있을 것이다.

[참고 문헌]

1. 엘리쌤의 [Orange3 데이터분석] 영상. Youtube. https://youtu.be/WZCpMYZlBKk
2. 코로나 확진자 데이터 세트. John Hopkins University github. https://github.com/CSSEGISandData/COVID-19
3. [문제상황] 코로나바이러스감염증-19. NAVER지식백과. https://terms.naver.com/entry.naver?docId=5920414&cid=40942&categoryId=32773
4. 서울과학종합대학원 디지털혁신처(2021). 3시간에 배우는 인공지능 데이터분석, 오렌지. 서울경제경영.
5. 이고잉, 이숙번 외 1명(2021). 헬로! 인공지능 생활코딩 머신러닝. 위키북스.

지도 위원

김 정 한(경상북도교육청 창의인재과장)
최 한 용(경상북도교육청 융합교육담당 장학관)

집필 위원

경북 SWAI교육 교사연구회
임 진 숙(경산과학고등학교 교사)
황 은 아(구미산동고등학교 교사)
박 윤 희(금오고등학교 교사)
조 예 린(구미여자고등학교 교사)
서 정 민(사동고등학교 교사)
황 상 연(상모중학교 교사)
최 훈 주(북삼중학교 교사)

기 획

손 유 경(경상북도교육청 장학사)

2022년 데이터 기반 인공지능 교육 자료

발 행 2022년 4월 일
발행처 경상북도교육청
주 소 (36759) 경상북도 안동시 풍천면 도청대로 511(갈전리)