Machine Learning (2021 Fall semester)

Programming Assignment: **Classification of Titanic Data set**

# Benchmark Dataset

사용하지 않는 feature들을 drop하는 과정

train = pd.read\_csv("train.csv")

train\_drop = train.drop(['PassengerId', 'Name', 'Ticket', 'Cabin'], axis = 1)

# Preprocessing

2.1 train data의 결측 치(NULL) 처리

Train data의 결측 값은 Age와 Embarked feature이다. 해당 결측 값 확인은 isnull().sum() 함수를 통해 파악했다. 우선 Age feature 결측 값은, 나이의 평균값을 채워 넣는 방식으로 처리했다. 또한, embarked feature 결측 값은, age feature처럼 평균값으로 처리할 수 없으므로, 결측 된 column은 제거하였다.

age\_avg = round(train\_drop['Age'].mean())

values  = {"Age" : age\_avg}

train\_drop = train\_drop.fillna(value=values)

train\_drop\_ = train\_drop.dropna()

2.2 One-Hot encoding을 수행할 필요가 있는 feature

컴퓨터는 문자보다는 숫자 처리에 조금 더 용이하다. 따라서 자연어 처리에서는 문자를 숫자로 바꾸는 여러가지 기법들이 존재한다. 그 중에서, One-Hot encoding은 단어를 표현하는 가장 기본적인 방법이고, 표현하고 싶은 단어의 인덱스에 특정한 정수 값을 부여하는 단어의 벡터 표현 방식이다.

Titanic train data set에는, 성별을 나타내는 sex feature와 embarked feature가 문자열로 이루어진다. 따라서, 해당 feature의 수식계산을 통한 학습을 위해서는 One-Hot encoding의 처리가 필요하다. 우선적으로, sex feature의 value인 male과 female을 표현하는 정수 값으로 각각 0과 1을 부여했다. 그리고, embarked feature의 value인 S, C, Q에는 각각 0, 1, 2를 부여했다. 이후, mapping 함수를 이용해 train data의 feature값을 map한 value로 변환했다.

sex\_map = {"male":0, "female":1}

embark\_map = {"S":0, "C":1,"Q":2}

train\_drop\_map = train\_drop\_.copy()

train\_drop\_map['Sex'] = train\_drop\_map['Sex'].map(sex\_map)

train\_drop\_map['Embarked'] = train\_drop\_map['Embarked'].map(embark\_map)

2.3 Split train data

label = train\_scale.iloc[:, 0].values

feature = train\_scale.iloc[:, 1:7].values

train\_label, test\_label, train\_feature, test\_feature = train\_test\_split(label, feature, test\_size = 0.3, random\_state = 42)

2.4 그 외 진행한 전 처리 과정 (Remove Outlier)

Data outlier는 이상이 있는 데이터를 뜻한다. 일반적인 데이터 패턴과 다르게 매우 이상한 패턴을 가지고 있는 데이터를 뜻한다. Machine learning에 있어서, 이러한 outlier data로 인해 모델의 성능이 크게 영향을 받는다. 따라서 titanic data set에 **Sigma clipping** 연산을 수행하여, outlier를 제거하였다. Sigma clipping연산을 수행한 feature는 age feature이다.

quartiles = np.percentile(train\_drop\_map['Age'], [25, 50, 75])

mu = quartiles[1]

sig = 0.74 \* (quartiles[2] - quartiles[0])

train\_rm\_outlier = train\_drop\_map.query(' (Age > @mu - 5\*@sig) & (Age < @mu + 5\*@sig)')

# Machine Learning Models

## K-Nearest Neighbors (KNN)

KNN에서는, K(이웃)의 개수를 [1~5]까지 변화시키면서 test data의 결과가 어떻게 변하는지 분석하였다.

표 1: KNN 결과 [2]

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 1: KNN 결과 [2]

[표 1]과 [그림 1]을 통해 알 수 있는 결과는 다음과 같다. 우선, K의 값이 증가할수록 학습 데이터 상에서 정확도가 감소함을 알 수 있다. 하지만 반대로, 테스트 데이터의 정확도는 K의 값이 증가할수록, 비례하게 증가함을 알 수 있다. 또한, KNN모델의 테스트 데이터의 정확도는 68%에서 71% 전후임을 알 수 있다.

## Logistic Regression

Logistic Regression에서는, iteration 횟수를 [0~100] 범위에서 20씩 변화시키면서 test data에서 결과가 어떻게 변하는지 분석하였다. 또한, Iteration 횟수를 100으로 고정한 후 regularization term이 [0~5] 범위에서 1씩 변화하면서 test data에서 결과가 어떻게 변하는지 분석하였다.

표 2 : Logistic Regrssion 결과 [1] (iteration)

테이블이(가) 표시된 사진

자동 생성된 설명

그림 2: Logistic Regression 결과 [2] (iteration)

우선, iteration의 횟수를 변경하면서 변화되는test data의 결과이다. [표 2]와 [그림 2]를 통해 알 수 있는 결과는, iteration이 0인 것을 제외하고는 학습 데이터와 테스트 데이터 모두 변동이 크지 않음을 알 수 있다.

표 3: Logistic Regression 결과 [1] (C)

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 3: Logistic Regression 결과 [2] (C)

다음은. Iteration 횟수를 100으로 고정한 후 regularization term에 변화를 주어 변화되는 test data의 결과이다. 위의 결과와 유사하게, C값의 변동은 학습 데이터와 테스트 데이터의 변동이 거의 없었다.

## Decision Tree

Decision Tree 에서는, Information Gain을 통해 test data에서 결과가 어떻게 나오는지 분석하고, tree를 시각화 하는 과정을 거친다. 또한, Bagging 기법을 이용하여 bag의 수에 따라 test data의 결과가 어떻게 변하는지 decision tree와 비교하며 분석하는 과정을 가진다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 4: Decision Tree 결과

표 4: Decision Tree (Bagging) 결과 [1]

텍스트이(가) 표시된 사진

자동 생성된 설명

그림5: Decision Tree (Bagging) 결과 [2]

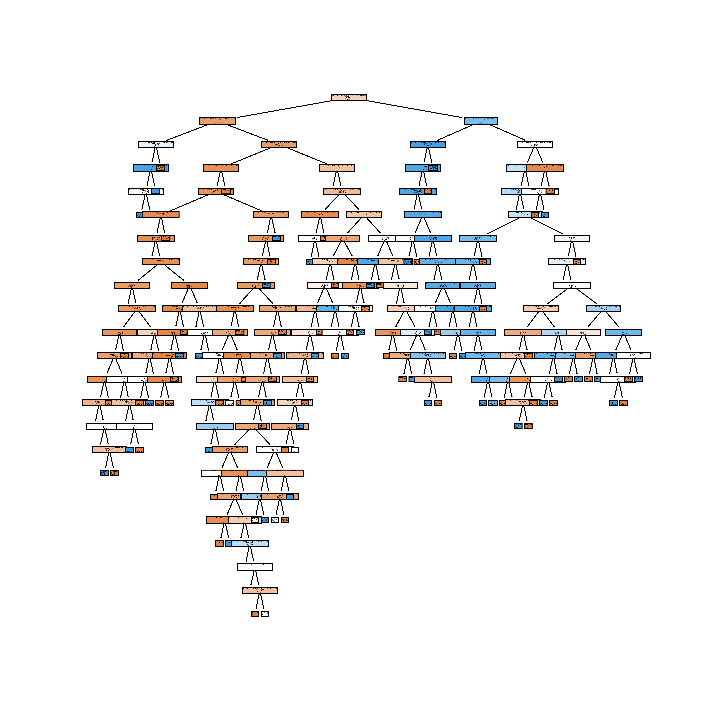
Information gain을 통한 test data의 결과는 [그림 4]를 통해 나타난다. 또한, Decision Tree 기반으로 Bagging 기법을 bag의 수에 따라 test data의 변화량은 [표6] 과 [그림 5]를 통해 나타난다. Information gain을 통한 test data의 학습데이터 정확도는 Bagging 기법을 통한 정확도 보다 높은 것으로 나타난다. 하지만 테스트 데이터의 정확도는 bag가 1일때를 제외하면, bagging 기법을 통한 정확도가 더 높음을 알 수 있다. 또한, bag의 수가 5일때가 학습데이터 정확도와 테스트 데이터의 정확도 모두 가장 높은 것을 알 수 있다. [그림 6]은 tree를 시각화 한 것이다.

그림 6: Visualize Tree