
Titanic Project Team 1



Sample Project Proposal

Topic | Titanic Project

Description | Data analysis, Machine learning,

Expected Duration | 4 weeks

Team Member | 허승준, 이연서, 이준성

- Timeline -

Before Start

Planning, Research

Week 6

Exploratory data analysis

Week 8

Building machine learning
model and prediction

Week 5

Data analysis
Data Visualization #1

Week 7

Group Presentation #1
(~2.9)
Data Visualization

Week 9

Group Presentation #2
Data Visualization #2

Feature Engineering

1. Fill Null

- Age의 평균을 이용해 Null value 를 채움
- loc + boolean + column 을 사용해 값을 치환하는 방법
- 각 initial의 mean value를 null data에 넣어줌

```
df_train.loc[(df_train.Age.isnull())&(df_train.Initial=='Mr'),'Age'] = 32.74
df_train.loc[(df_train.Age.isnull())&(df_train.Initial=='Mrs'),'Age'] = 35.98
df_train.loc[(df_train.Age.isnull())&(df_train.Initial=='Master'),'Age'] = 4.57
df_train.loc[(df_train.Age.isnull())&(df_train.Initial=='Miss'),'Age'] = 21.86
df_train.loc[(df_train.Age.isnull())&(df_train.Initial=='Other'),'Age'] = 45.88

df_test.loc[(df_test.Age.isnull())&(df_test.Initial=='Mr'),'Age'] = 32.74
df_test.loc[(df_test.Age.isnull())&(df_test.Initial=='Mrs'),'Age'] = 35.98
df_test.loc[(df_test.Age.isnull())&(df_test.Initial=='Master'),'Age'] = 4.57
df_test.loc[(df_test.Age.isnull())&(df_test.Initial=='Miss'),'Age'] = 21.86
df_test.loc[(df_test.Age.isnull())&(df_test.Initial=='Other'),'Age'] = 45.88
```

[395] ✓ 0.1s

Feature Engineering (Cont.)

2. Change age from continuous to categorical

- Age는 originally continuous feature이다
- 이대로 써도 모델을 세울 수 있지만, Age 를 몇개의 group 으로 나누어 category 화 시켜줄 수 도 있습니다
- Group the ages for prediction will yield a more promising result

How?

- 함수를 만들어 메소드에 넣어줌

```
def category_age(x):  
    if x < 10:  
        return 0  
    elif x < 20:  
        return 1  
    elif x < 30:  
        return 2  
    elif x < 40:  
        return 3  
    elif x < 50:  
        return 4  
    elif x < 60:  
        return 5  
    elif x < 70:  
        return 6  
    else:  
        return 7  
  
df_train['Age_cat_2'] = df_train['Age'].apply(category_age)  
✓ 0.8s
```

Feature Engineering (Cont.)

3. Change initial, Embarked, and Sex

- String to Numerical

- Initial: Mr., Mrs., Miss, Master, Other

Map method를 사용해 컴퓨터가 인지할 수 있도록 수치화

시킨후

순서대로 정리하여 mapping을 하였음

```
df_train['Initial'] = df_train['Initial'].map({'Master': 0, 'Miss': 1, 'Mr': 2, 'Mrs': 3, 'Other': 4})
df_test['Initial'] = df_test['Initial'].map({'Master': 0, 'Miss': 1, 'Mr': 2, 'Mrs': 3, 'Other': 4})
df_train["Initial"]
```

0	2
1	3
2	1
3	3
4	2
..	
886	4
887	1
888	1
889	2
890	2

Name: Initial, Length: 891, dtype: int64

map() function returns a map object (which is an iterator) of the results after applying the given function to each item of a given iterable (list, tuple etc.)

Syntax :

```
map(fun, iter)
```

Parameters :

fun : It is a function to which map passes each element of given iterable.

iter : It is a iterable which is to be mapped.

So what does map() do?

두 번째 인자로 들어온 반복 가능한 자료형

(리스트나 튜플)을 첫 번째 인자로 들어온 함수에

하나씩 집어넣어서 함수를 수행하는 함수입니다

여러 개의 데이터를 한 번에 다른 형태로

변환하기 위해서 사용됨

Feature Engineering (Cont.)

4. Pearson Correlation

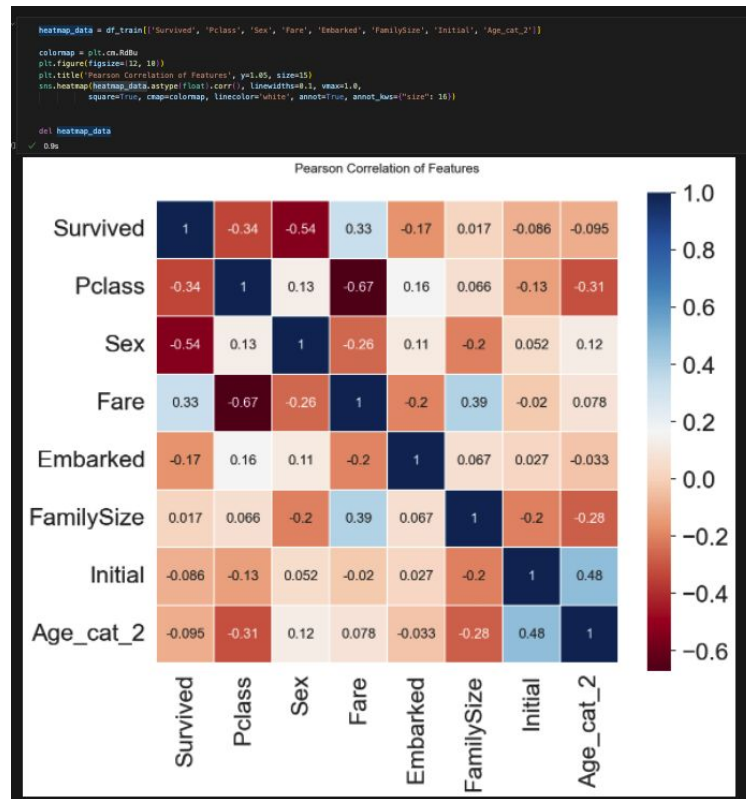
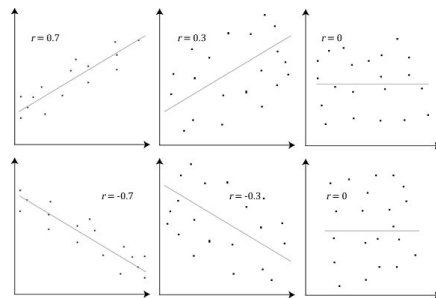
- Pearson correlation evaluates the linear relationship between two continuous variables

여기서, Sex 와 Pclass 가 Survived 에 상관관계가 어느 정도 있음을 볼 수 있습니다. 그리고 생각보다 fare 와 Embarked 도 상관관계가 있음을 볼 수 있습니다.

$$r = \frac{N\sum xy - (\sum x)(\sum y)}{\sqrt{[N\sum x^2 - (\sum x)^2][N\sum y^2 - (\sum y)^2]}}$$

Where:

- N = number of pairs of scores
- $\sum xy$ = sum of the products of paired scores
- $\sum x$ = sum of x scores
- $\sum y$ = sum of y scores
- $\sum x^2$ = sum of squared x scores
- $\sum y^2$ = sum of squared y scores



Feature Engineering (Cont.)

One-hot encoding is used for: converting numerical categorical variables into binary vectors

5. One-hot encoding and Embarked

- 서로 다른 다섯개의 Columns로 값 섹션
- pandas의 get dummies를 이용한 분리방법
- 같은 형식으로 Embarked에도 one-hot encoding 방식으로 적용

```
df_train = pd.get_dummies(df_train, columns=['Initial'], prefix='Initial')  
df_test = pd.get_dummies(df_test, columns=['Initial'], prefix='Initial')  
df_train.head()
```

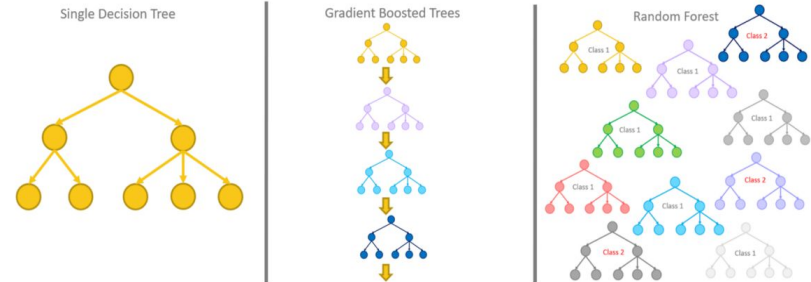
PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	Embarked	FamilySize	Age_cat_2	Initial_0	Initial_1	Initial_2	Initial_3	Initial_4	
0	1	0	3	Braund, Mr. Owen Harris	1	1	0	A/5 21171	1.981001	NaN	2	2	2	0	0	1	0	0
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	1	0	PC 17599	4.266662	C85	0	2	3	0	0	0	1	0
2	3	1	3	Heikkinen, Miss. Laina	0	0	0	STON/O2. 3101282	2.070022	NaN	2	1	2	0	1	0	0	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	1	0	113803	3.972177	C123	2	2	3	0	0	0	1	0
4	5	0	3	Allen, Mr. William Henry	1	0	0	373450	2.085672	NaN	2	1	3	0	0	1	0	0

Building Machine Learning Model

(Random Forest vs. Gradient Tree Boosting)

Gradient Boosting

- Similar to Random Forest
- Both are ensemble method that creates many decision trees
- Random Forest creates independent decision trees
- The difference is that Gradient Boosting builds decision trees one at a time, rather than independently, to correct errors made by previous trees.



Gradient boosting trees can be more accurate than random forests.

∴ ML modeling with Gradient Boosting Decision Tree Method

Building Machine Learning Model (Cont.)

- sklearn을 사용한 머신러닝 모델

```
#importing all the required ML packages
from sklearn.ensemble import RandomForestClassifier # 유명한 randomforestclassifier 입니다.
from sklearn import metrics # 모델의 평가를 위해서 씁니다
from sklearn.model_selection import train_test_split # traning set을 쉽게 나눠주는 함수입니다.
```

Building Machine Learning Model (Cont.)

1. Preparation

- 학습 데이터와 Target Label 분리로 준비 시작
- SKLearn의 Train Set 분리 단계 예시

****보통 train, test 만 언급되지만, 실제 좋은 모델을 만들기 위해서 저희는 valid set을 따로 만들어 모델 평가를 해보았습니다****

Why?

마치 축구대표팀이 팀훈련(train)을 하고 바로 월드컵(test)로 나가는 것이 아니라, 팀훈련(train)을 한 다음 평가전(valid)를 거쳐 팀의 훈련 정도(학습정도)를 확인하고 월드컵(test)에 나가는 것과 비슷합니다.

```
x_tr, x_vld, y_tr, y_vld = train_test_split(X_train,
                                             target_label,
                                             test_size=0.3,
                                             random_state=42,
                                             shuffle=True)
```

Building Machine Learning Model (Cont.)

2. Model Generation and Prediction


➤ 모델 제작과 데이터 예측

```
#Gradient Tree Boosting
#model = RandomForestClassifier()          #83.21% Accuracy

from sklearn.ensemble import GradientBoostingClassifier    #85.07 Accuracy

model = GradientBoostingClassifier(learning_rate=0.2)
model.fit(X_tr, y_tr)
model_prediction = model.predict(X_vld)
model.score(X_tr, y_tr)
```

➤ 제작된 모델과 함께 fit 함수를 이용한
학습
➤ 예측값 측정



```
(y_vld.shape[0], 100 * metrics.accuracy_score(model_prediction, y_vld)))
```

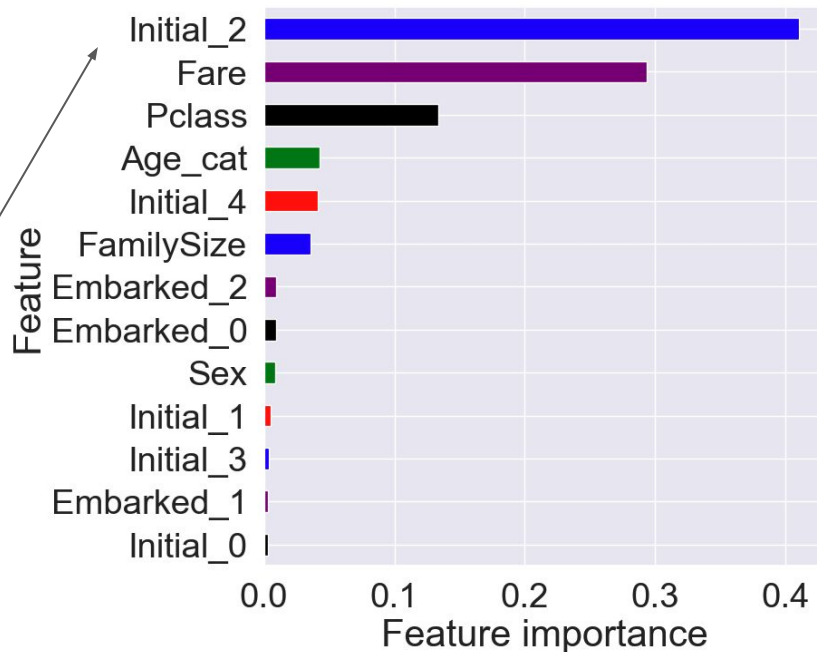
총 268명 중 84.33% 정확도의 결과를
보여줌

Building Machine Learning Model (Cont.)

3. Feature Importance Analysis

➤ 데이터 타입별 중요도 나열

- 예시에서는 **Fare**가 중요도가 가장 높았으나, modeling technique를 Gradient Tree Boosting로 바꾸었을때 값이 달라진것 볼 수 있음



Building Machine Learning Model (Cont.)

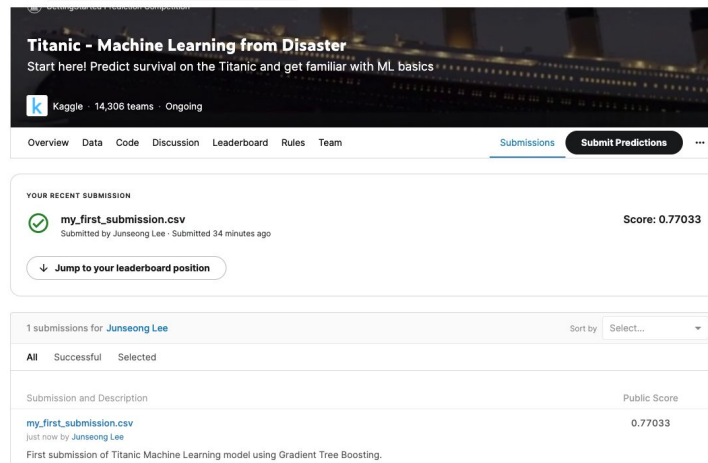
4. Prediction and Final Submission

- 마지막 결과값을 CSV파일 형식으로 추출과 함께 결과 점수 확인 → 77% accuracy

ML models accuracy is generally considered good anything greater than 70% or +

Overall... Good learning experience

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1
...
413	1305	0



The screenshot shows the Kaggle competition page for "Titanic - Machine Learning from Disaster". The header includes the competition title and a brief description: "Start here! Predict survival on the Titanic and get familiar with ML basics". Below the header, there are tabs for "Overview", "Data", "Code", "Discussion", "Leaderboard", "Rules", and "Team". A "Submit Predictions" button is visible. The main content area displays "YOUR RECENT SUBMISSION" with a green checkmark icon, the filename "my_first_submission.csv", and the submission details: "Submitted by Junseong Lee · Submitted 34 minutes ago". The score is shown as "Score: 0.77033". Below this, there is a button that says "Jump to your leaderboard position". At the bottom, a table shows the submission details, including the filename, the user "Junseong Lee", and the public score "0.77033".