

**데이터사이언스 HW3**

|  |  |
| --- | --- |
| 과목명: | 데이터사이언스 |
| 학과: | 독일어과 |
| 학번: | 201400867 |
| 이름: | 김태호 |
| 제출일: | 2020.07.01. |

**목 차**

**1. 데이터 분석 프로젝트의 목적 1**

2. 데이터 입수 절차 1

**3. 탐색적 데이터 분석 1**

3.1. 각 변수들에 대한 univariate 탐색적 분석 2

3.2. 종속변수 survival과 독립변수 간 bivariate 탐색적 분석 7

**3.2.1 Numerical Data 7**

**3.2.2 Categorical Data 9**

3.3. 종속변수와 통계적으로 유의미한 독립변수 12

**3.3.1 Numerical Data 12**

**3.3.2 Categorical Data 13**

4. 최종 모형 구축을 위한 탐색적 데이터 분석 결과 활용 방향 14

4.1 Numerical Data 14

4.2 Categorical Data 18

**5. 최적의 분류모형 구축 및 test 데이터에 대한 예측 22**

**6. Kaggle competition 결과 24**

**6.1 자체 테스트 24**

**6.2 개선과정 26**

**6.3 최대 점수 27**

**참고문헌 28**

**그 림 목 차**

**[그림 3-1] Train.csv 요약 1**

**[그림 3-2] Train, Test 파일 정보 요약 2**

**[그림 3-3] PassengerId** 2

**[그림 3-4] Survived 3**

**[그림 3-5] Pclass** 3

**[그림 3-6] Name 3**

**[그림 3-7] Sex** 4

**[그림 3-8] Age** 4

**[그림 3-9] SibSp 5**

**[그림 3-10] Parch 5**

**[그림 3-11] Ticket 5**

**[그림 3-12] Fare 6**

**[그림 3-13] Cabin 6**

**[그림 3-14] Sex 7**

**[그림 3-15] Fare 7**

**[그림 3-16] Age 8**

**[그림 3-17 SibSp 8**

**[그림 3-18] Parch 9**

**[그림 3-19] Pclass 10**

**[그림 3-20] Sex 10**

**[그림 3-21] Embarked 11**

**[그림 3-22] Cabin 11**

**[그림 3-23] Ticket 12**

**[그림 3-24] Family Size 13**

**[그림 3-25] Title 14**

**[그림 4-1] Fare 15**

**[그림 4-2] ] Fare 결측값 채우기 15**

**[그림 4-3] Fare 분포 15**

**[그림 4-4] Fare 구간 16**

**[그림 4-5] Age random 16**

**[그림 4-6]Age 분포 17**

**[그림 4-7] Age 구간 17**

**[그림 4-8] Family Size Mapping 17**

**[그림 4-9] One-hot-encoding 18**

**[그림 4-10] One-hot-encoding 처리 후 19**

**[그림 4-11] Title Mapping 20**

**[그림 4-12] Sex Mapping 20**

**[그림 4-13] Embarked Mapping 21**

**[그림 4-14] Cabin 21**

**[그림 4-15] Cabin Mapping 22**

**[그림 5-1] Cross Validation 22**

**[그림 5-2] kNN 22**

**[그림 5-3] Decision Tree 23**

**[그림 5-4] Random Forest 23**

**[그림 5-5] Naive Bayes 24**

**[그림 5-6] SVM 24**

**[그림 6-1] Test file 25**

**[그림 6-2] 다른 csv파일과 비교 25**

**[그림 6-3] 성능 개선 26**

**[그림 6-4] 최대 점수 및 등수** **27**

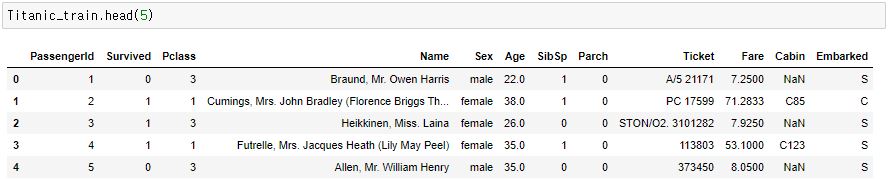
1. 데이터 분석 프로젝트의 목적

Tintanic호에 승선한 사람들의 다양한 인적 정보를 기반으로 생존 가능성을 파악할 수 있는 모형을 구축한다.

2. 데이터 입수 절차

Kaggle.com 에서 Titanic: Machine Learning from Disaster, Data 페이지에서 train.csv, test.csv, gender\_submission.csv 파일을 제공한다.

**3. 탐색적 데이터 분석**

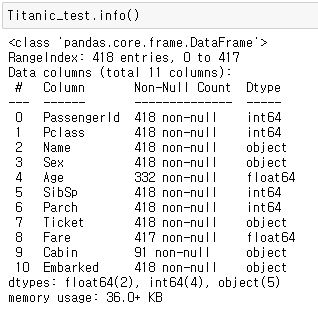
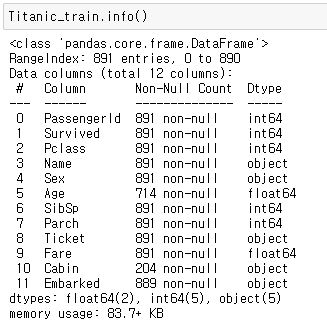
****

[그림 3-1] Train.csv 요약

[그림 3-1] 과 같이 파일 내 Data는 다음과 같은 의미를 갖는다.

* Survived: 0 = No, 1 = Yes
* pclass: Ticket class (1 = 1st, 2 = 2nd, 3 = 3rd )
* sibsp: Titanic 호에 같이 탑승한 siblings / spouses 수
* parch: Titanic 호에 같이 탑승한 parents / children 수
* ticket: Ticket number
* cabin: Cabin number
* embarked: 승선한 항구, C = Cherbourg, Q = Queenstown, S = Southampton

[그림 3-2] Train, Test 파일 정보 요약

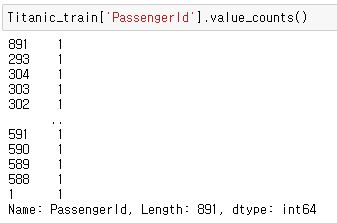


[그림 3-2]와 같이 train.csv는 891행, 12열로 이루어져 있으며, test.csv는 418행, 12열로 이루어져 있다.

train.csv에서 Age, Cabin, Embarked에 각각 177, 687, 2개 Null값이 존재하고, test.csv에서 Age, Fare, Cabin에서 각각 86, 1, 327개 Null 값이 있다.

3.1. 각 변수들에 대한 univariate 탐색적 분석

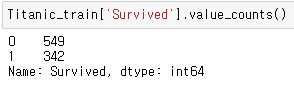
* PassengerId



[그림 3-3] PassengerId

[그림 3-3]과 같이 PassengerId는 승객을 구분하는 번호이며, 1번부터 891번까지 결측값 없이 모두 있다.

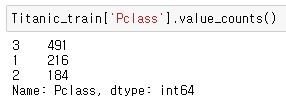
* Survived



[그림 3-4] Survived

[그림 3-4]와 같이 Survived는 0이면 사망, 1이면 생존을 나타내며 결측값 없이 891개 있다.

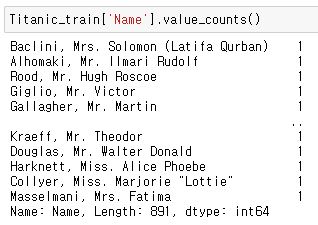
* Pclass



[그림 3-5] Pclass

[그림 3-5]와 같이 Pclass는 1이면 1등석, 2이면 2등석, 3이면 3등석을 나타내며 결측값 없이 891개 있다.

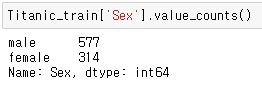
* Name



[그림 3-6] Name

[그림 3-6]과 같이 Name은 승객 이름을 나타낸다. Mr., Mrs. 등 구분될 수 있는 호칭이 포함되어 있다. train.csv에서는 Dona, test.csv에서는 Don 등 남녀가 구별되는 특별한 호칭도 포함되어 있다.

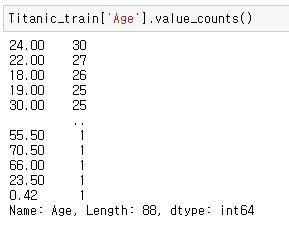
* Sex



[그림 3-7] Sex

[그림 3-7]과 같이 Sex는 남, 여로 구분되어 있다. 성별에 결측값은 없다.

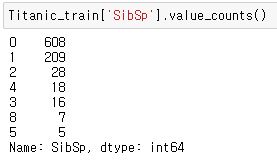
* Age



[그림 3-8] Age

[그림 3-8]과 같이 Age는 승객 나이를 표시하며, 소수점 이하 나이도 기록되어 있고, 177개 결측값이 있다.

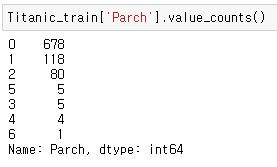
* SibSp



[그림 3-9] SibSp

[그림 3-9]와 같이 SibSp는 함께 탑승한 형제자매, 배우자이며 7개 크기로 구분되어 있다.

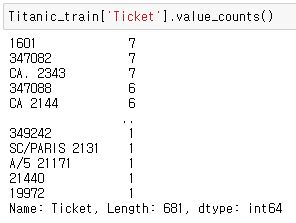
* Parch



[그림 3-10] Parch

[그림 3-10]과 같이 Parch는 함께 탑승한 부모, 자식 숫자이며, 마찬가지로 7개 크기로 구분된다.

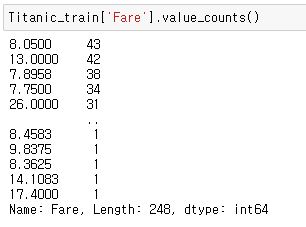
* Ticket



[그림 3-11] Ticket

[그림 3-11]과 같이 Ticket은 승객이 가지고 있는 티켓 번호이다. 같은 티켓 번호가 있으나, 대부분 다른 티켓 번호를 가지고 있다.

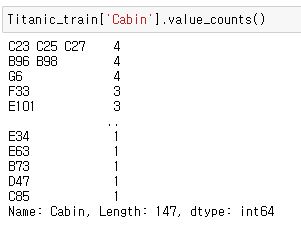
* Fare



[그림 3-12] Fare

[그림 3-12]와 같이 Fare는 승객이 지불한 요금이다. test.csv에서 1개의 결측값이 있다.

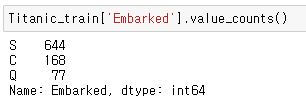
* Cabin



[그림 3-13] Cabin

[그림 3-13]과 같이 Cabin은 승객이 탑승한 구획 번호이다. 가장 많은 결측값을 보이는 변수이다.

* Embarked



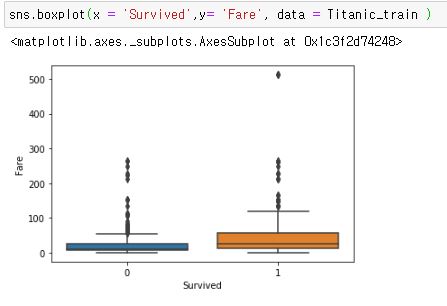
[그림 3-14] Sex

[그림 3-14]와 같이 Embarked는 승객이 탑승한 항구이다. train.csv에서 2개의 결측값이 있다.

3.2. 종속변수 survival과 독립변수 간 bivariate 탐색적 분석

**3.2.1 Numerical Data**

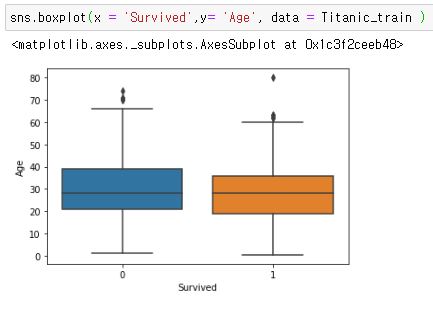
* **Fare**



[그림 3-15] Fare

[그림 3-15]와 같이 생존한 승객의 Fare가 전반적으로 약간 높다. 즉, 높은 요금을 지불한 사람이 살아남을 확률이 높았다.

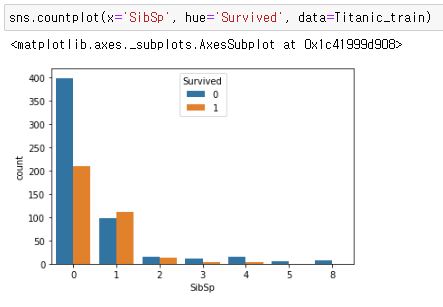
* **Age**



[그림 3-16] Age

[그림 3-16]과 같이 생존한 승객의 Age가 전반적으로 약간 낮다. 즉, 어린 사람이 살아남을 확률이 높았다.

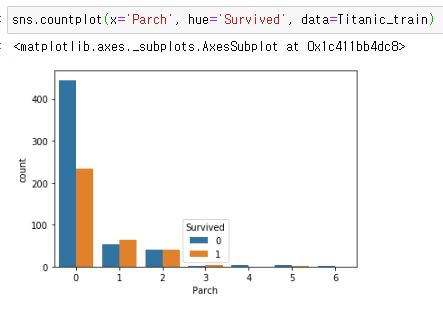
* **SibSp ( # of siblings and spouse)**



[그림 3-17 SibSp

[그림 3-17]과 같이 혼자 탑승한 승객에서 사망한 승객이 많다. 1명 이상 같이 탄 경우 생존한 사람이 더 많으나 1명보다 커지면 다시 사망한 승객이 많아진다.

* **Parch ( # of parents and children)**

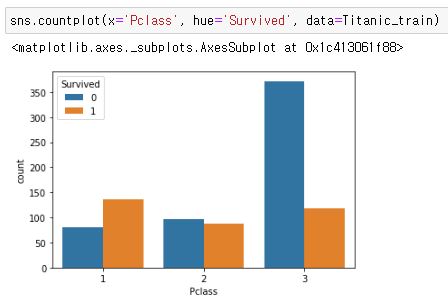


[그림 3-18] Parch

[그림 3-18]과 같이 부모, 자식 없이 탑승한 경우 사망한 사람이 많았으나. 살아남은 사람도 가장 많았다. 1명의 부모, 자식과 같이 탄 경우 생존한 승객이 더 많았다.

**3.2.2 Categorical Data**

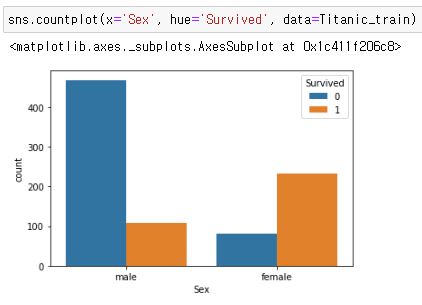
* **Pclass**



[그림 3-19] Pclass

[그림 3-19]와 같이 3등석에서 사망한 승객의 숫자가 많다. 1등석은 생존한 사람이 사망한 사람보다 많다. 따라서 등급이 높을수록 살아남을 확률이 크다.

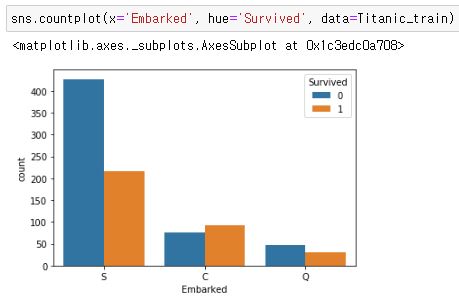
* **Sex**



[그림 3-20] Sex

[그림 3-20]과 같이 남자 승객보다 여자 승객이 많이 생존했다. 즉, 여성이면 살아남을 확률이 높다.

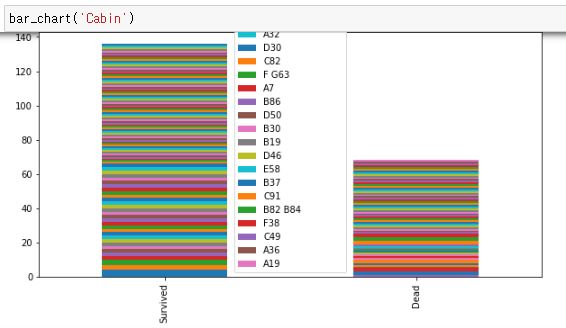
* **Embarked**



[그림 3-21] Embarked

[그림 3-21]와 같이 가장 많은 승객이 S에서 탑승했고, S에서 탑승한 승객이 가장 많이 사망했고, 가장 많이 생존했다. C에서 탑승한 승객의 경우 살아남은 사람이 약간 더 많다.

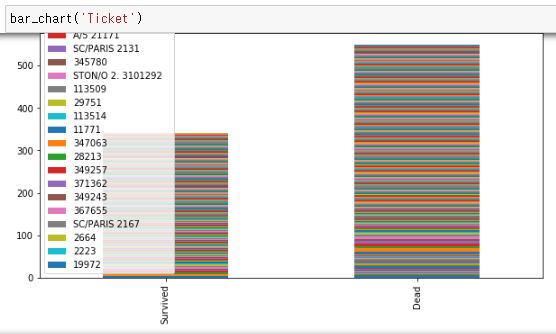
* Cabin



[그림 3-22] Cabin

[그림 3-22]와 같이 다양한 객실 승객이 살아남았음을 알 수 있으며, 특정 객실에서는 많은 승객이 사망했음을 알 수 있다.

* **Ticket**



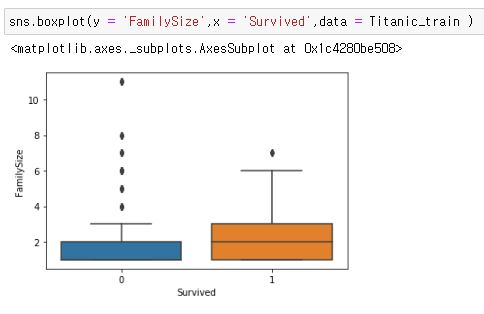
[그림 3-23] Ticket

[그림 3-23]과 같이 티켓 번호는 매우 다양하다. 전반적으로 많은 사람이 사망했음을 알 수 있다. 특정 티켓이 많이 사망했는지 찾기 어렵다.

3.3. 종속변수와 통계적으로 유의미한 독립변수

**3.3.1 Numerical Data**

Fare, Age 모두 종속변수에 영향을 주는 유의미한 독립변수이며, Fare가 높을수록, Age가 낮을수록 생존할 확률이 높았다.



[그림 3-24] Family Size

[그림 3-24]와 같이 Family Size는 SibSp( # of siblings and spouse)와 Parch( # of parents and children)를 하나로 묶어 ‘가족크기’로 볼 수 있다. FamilySize는 2 이하일 때 사망할 확률이 높았고, 4 미만인일 때 생존할 확률이 높았다. 그러나 가족 크기가 너무 큰 경우 사망한 경우가 많다.

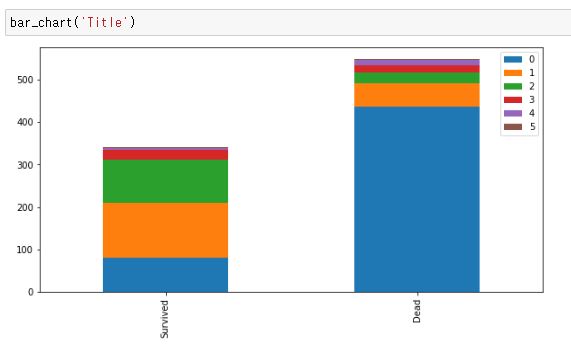
**3.3.2 Categorical Data**

Pclass는 1등석의 경우 생존할 확률이 높았고, 3등석의 경우 사망할 확률이 월등히 높았다.

Sex는 여성일 경우 살아남을 확률이 높았다.

Embarked는 C에서 탄 승객이 생존할 확률이 약간 높았고, S에서 탄 승객은 사망할 확률이 컸다.

Cabin에서 특정 객실에 탑승한 승객은 살아남을 확률이 높았다.



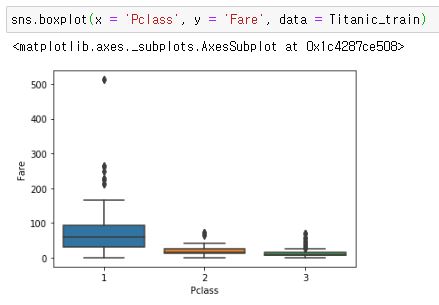
[그림 3-25] Title

[그림 3-25]와 같이 Name은 호칭으로 분류할 때, Title로 구분할 수 있으며, 특정 Title이 더 많이 살아남았다. 주로 Mr.이면 많이 사망했다. Miss., Mrs.이면 많이 생존했다.

4. 최종 모형 구축을 위한 탐색적 데이터 분석 결과 활용 방향

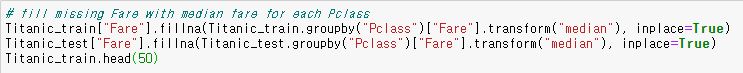
**4.1 Numerical Data**

* **Fare**



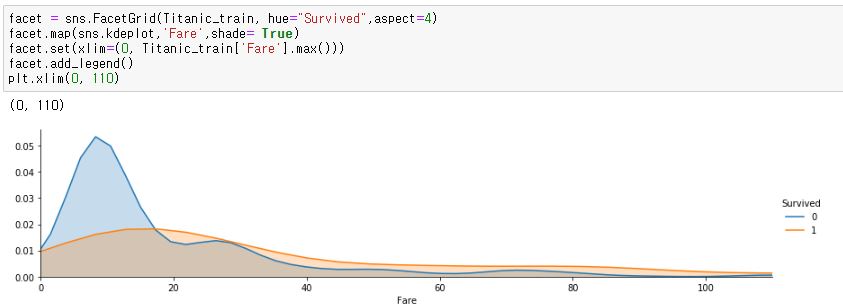
[그림 4-1] Fare

[그림 4-1]과 같이 Fare는 Pclass가 높은 등급일수록 평균 요금이 높아지고 구간도 넓어진다. Pclass에는 결측값이 없으므로 Pclass 요금의 중간값을 계산해 결측값을 채운다.



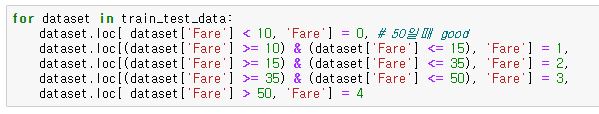
[그림 4-2] Fare 결측값 채우기

[그림 4-2]와 같이 Fare 결측값을 Pclass별 요금의 중간값으로 대체한다.



[그림 4-3] Fare 분포

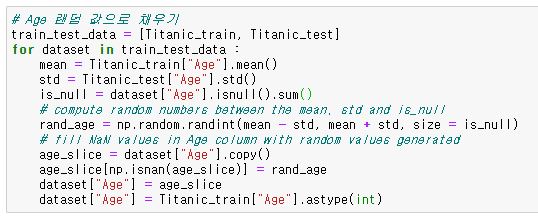
[그림 4-3]과 같이 Fare 분포를 보면 약 15 초과에서 생존자가 사망자보다 많아진다. 이후 25 정도에서 비슷한 값을 보이지만 뒤바뀌진 않는다. 10 정도에서는 사망자가 더 이상 늘지 않고 줄어드는 추세를 보이는 지점이다. 15 정도를 두 번째 지점으로 정할 수 있고, 35 초과부터는 생존자도 감소하는 모습을 보이므로 한 지점으로 정할 수 있다. 50 초과에서는 사망자, 생존자 모두 적으며 비율도 큰 변화를 보이지 않는다.



[그림 4-4] Fare 구간

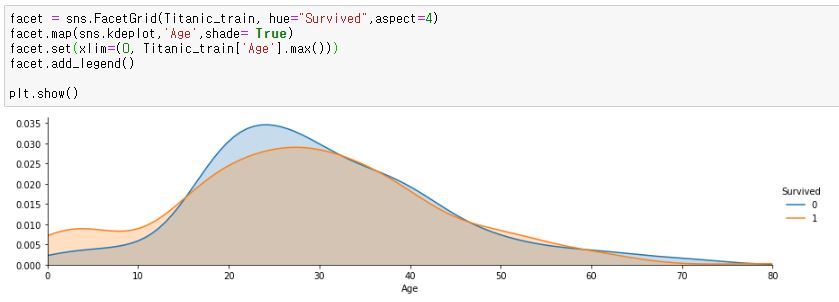
[그림 4-4]와 같이 Fare 분포를 활용해 Fare 요금을 구간마다 일정한 값으로 대체하여 이상값이 나타나지 않도록 Fare를 구간으로 만들 수 있다. 기존에 확인했을 때는 500에 가까운 이상값이 있었는데 Pclass 1등석 수준에 해당하는 4구간으로 포함시켜 계산할 수 있다.

* Age



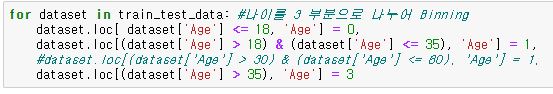
[그림 4-5] Age random

[그림 4-5]와 같이 나이는 train.csv에 존재하는 나잇값들의 평균값과 표준편차를 이용해 결측값을 채울 수 있다. 임의의 숫자는 평균에서 표준편차만큼 빼거나 더한 숫자 사이에서 결정된다. Pclass를 기준으로 등급별 나이 중간값을 채워 넣는 방법도 있으나 임의의 숫자를 넣는 방법이 더 높은 정확도를 보였다.



[그림 4-6]Age 분포

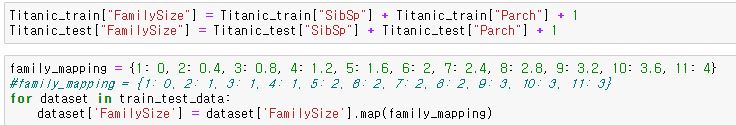
[그림 4-6]과 같이 나이는 약 15세 지점에서 사망자와 생존자의 숫자가 뒤바뀐다. 그리고 35세 이상에서 다시 만나며 이후로는 큰 변화를 보이지 않는다.



[그림 4-7] Age 구간

[그림 4-7]과 같이 Age 분포에 따라 나이를 3구간으로 나눌 수 있다. 18세 이하의 어린이, 18-35세 청년, 35세 초과 성인으로 나눌 수 있다. 35세 초과 구간이 넓기 때문에 2가 아닌 3을 부여했다.

* FamilySize : SibSp ( # of siblings and spouse) +Parch ( # of parents and children)



[그림 4-8] Family Size Mapping

[그림 4-8]과 같이 Family Size는 SibSp와 Parch 합이며 0과 구분하기 위해 최솟값을 1로 부여한 뒤 다시 1부터 11까지 숫자를 0.4 단위 또는 1 단위로 대응시킬 수 있다. 1단위씩 그대로 더하지 않는 이유는 컴퓨터가 계산할 때 1과 11 차이를 다른 컬럼들에 비해 과도하게 인식하는 것을 막기 위해서 다. 서로 다른 크기를 갖는 연속형 변수에 범주형 변수처럼 같은 숫자를 여러 번 대입하면 오히려 정확성이 떨어지므로 크기에 따라 0.4 단위로 대응시킨다.

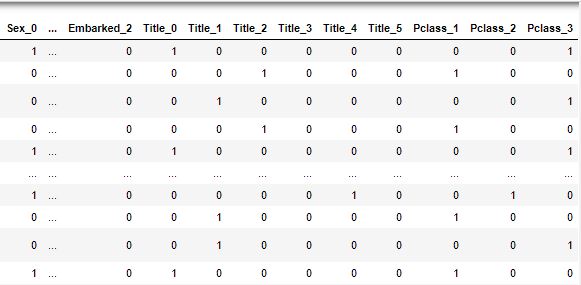
**4.2 Categorical Data**

* **One-hot-encoding**



[그림 4-9] One-hot-encoding

[그림 4-9]와 같이 범주형 변수를 컴퓨터가 처리할 때 똑같은 비중으로 처리할 수 있도록, 하나의 범주형 변수를 여러 개 컬럼으로 나누어 해당하는 컬럼 값에만 1을 갖도록 하고 나머지 값에는 0을 대입하는 것을 One-hot-encoding이라고 한다. 인코딩 처리 후 변수가 갖는 범주 개수만큼 컬럼 수가 증가한다.



[그림 4-10] One-hot-encoding 처리 후

[그림 4-10]과 같이 One-hot-encoding 처리 이후 컬럼 수가 증가한다. Sex는 Sex\_0, Sex\_1로 컬럼이 늘어나며 남자의 경우 컬럼 Sex\_0에만 1로 표시되며 여자는 0으로 표시되고, Sex\_1 컬럼에서 여자는 1로 표시되고, 남자는 0으로 표시된다. Title은 다섯개의 범주를 가지고 있어 5개 컬럼을 갖고, Pclass는 3가지 범주를 가지므로 3개 컬럼이 된다. Pclass, Title, Sex, Embarked, Cabin을 One-hot-encoding 처리한다. 개별적으로 변수들을 One-hot-encoding 처리를 위해 각 범주를 숫자로 대응시킨다. 개별 변수를 숫자로 처리한 이후, 머신러닝을 실행하기 직전에 One-hot-encoding 처리한다.

* **Pclass**

Pclass는 기존에 가지고 있는 숫자를 그대로 사용한다. 1등석은 1, 2등석은 2, 3등석은 3으로 입력되어 있으므로 그대로 사용한다.

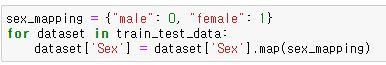
* **Title**



[그림 4-11] Title Mapping

[그림 4-11]과 같이 Title은 Name을 호칭에 따라 6가지 범주로 구분한 컬럼이다. Mr, Miss, Mrs, Master, Officer, Royalty 로 구분하여 이름에 따라 대응시키고 Name 컬럼은 사용하지 않고 Drop 한다.

* **Sex**



[그림 4-12] Sex Mapping

[그림 4-12]와 같이 Male은 0으로 Female은 1로 대응시킨다.

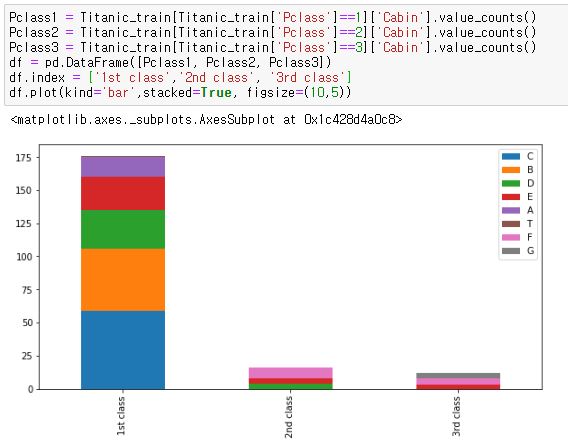
* **Embarked**



[그림 4-13] Embarked Mapping

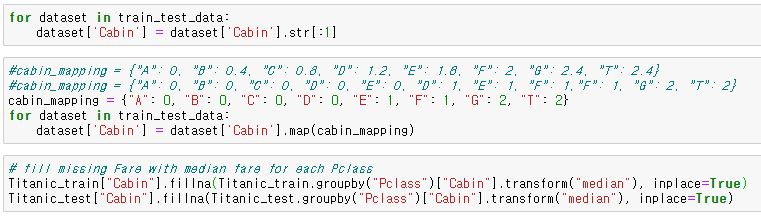
[그림 4-13]과 같이 Embarked는 S를 0, C는 1, Q는 2로 대응시킨다. 결측값은 해당 컬럼에서 가장 많이 등장한 값으로 채워 넣는다. Train 데이터셋에서는 결측값 2개에 S가 대입된다.

* **Cabin**



[그림 4-14] Cabin

[그림 4-14]와 같이 Cabin은 Pclass에 따라 구분할 수 있다. 2등석은 다른 등급과 겹치는 Cabin이 있으나 1등석과 3등석의 구분은 확실하다. 따라서 Cabin 결측값은 Pclass 등급에 따라 채워 넣을 수 있다.

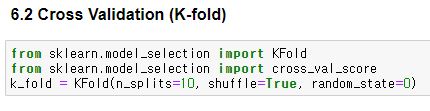


[그림 4-15] Cabin Mapping

[그림 4-15]와 같이 Cabin값은 가장 앞자리 알파벳으로 구분할 수 있다. 알파벳으로 구분한 뒤 1등석 Cabin에 해당하는 A, B, C, D에는 0을 부여한다. E의 경우 1등석도 포함되지만 2등석도 존재한다. E에 0을 대입하는 경우 One-hot-encoding시 train과 test 데이터셋에 서로 다른 컬럼이 생성되어 오류를 일으킨다. 따라서 E에는 1을 대응시키고, F에도 1을 대응시킨다. G, T는 3등석에 해당하므로 2를 대응시킨다. Cabin은 결측값이 많아 다르게 조정하면 정확도가 상승할 가능성이 있다.

**5. 최적의 분류모형 구축 및 test 데이터에 대한 예측**

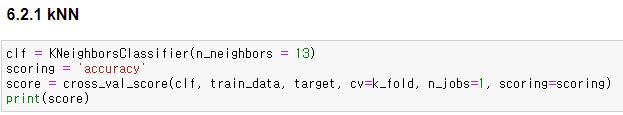
**- Cross Validation**

****

[그림 5-1] Cross Validation

[그림 5-1]과 같이 Cross Validation을 위해 k\_fold를 계산한다.

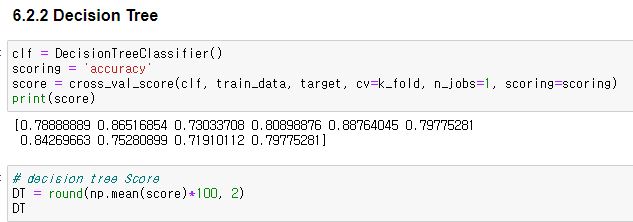
- kNN

****

[그림 5-2] kNN

[그림 5-2]와 같이 kNN을 사용해 점수를 측정해볼 수 있다.

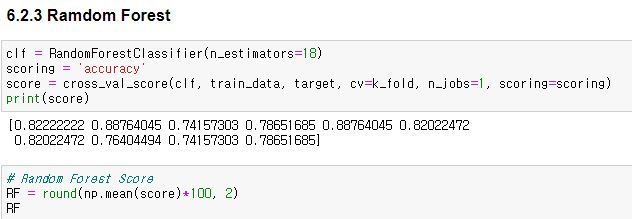
- Decision Tree



[그림 5-3] Decision Tree

[그림 5-3]과 같이 Decision Tree를 사용해 점수를 측정해볼 수 있다.

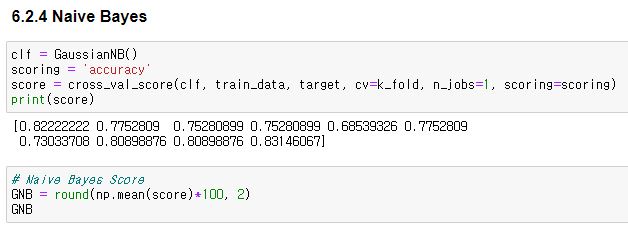
- Random Forest



[그림 5-4] Random Forest

[그림 5-4]와 같이 Random Forest를 사용해 점수를 측정해볼 수 있다.

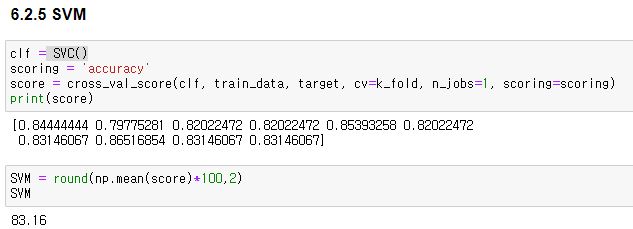
- Naive Bayes



[그림 5-5] Naive Bayes

[그림 5-5]와 같이 Naïve Bayes를 사용해 점수를 측정해볼 수 있다.

- SVM



[그림 5-6] SVM

[그림 5-6]과 같이 SVM을 사용해 점수를 측정해볼 수 있다. 가장 높은 점수를 보이며 일정하게 점수를 보이므로 SVM을 Competition에 활용한다.

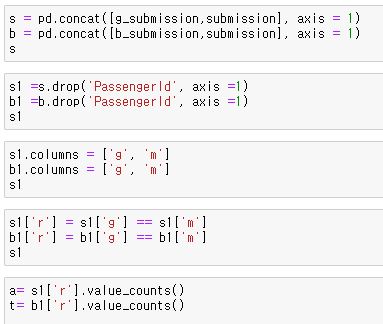
**6. Kaggle competition 결과**

**6.1 자체 테스트**



[그림 6-1] Test file

[그림 6-1]과 같이 SVM을 사용해 제출용 csv 파일을 만들고 gender\_submission 파일과 직접 비교해자체적으로 점수를 측정해볼 수 있다.



[그림 6-2] 다른 csv파일과 비교

[그림 6-2]와 같이 이미 존재하는 csv 파일 또는 gender\_submission과 동시에 점수를 비교해볼 수 있다. True, False 개수를 측정한다.

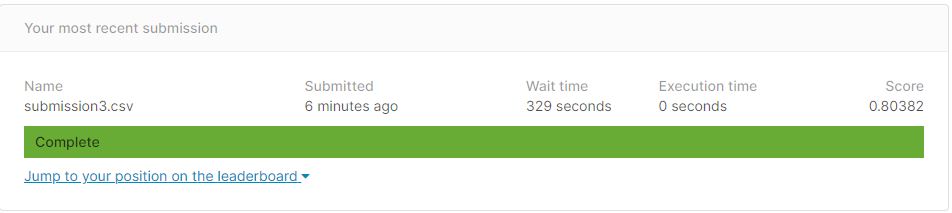
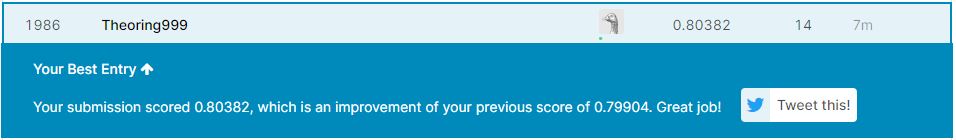
**6.2 개선과정**



[그림 6-3] 성능 개선

[그림 6-2]와 같이 성능 개선 과정 일부를 기록하였다. 개별 컬럼 처리 방법 조정 및 구간 조정, 다른 모델 사용으로 여러 조합을 통해 성능을 개선시키는 과정을 기록하였다.

**6.3 최대 점수**



[그림 6-4] 최대 점수 및 등수

[그림 6-4]와 같이 최대 80.3% 정확도를 보였으며, 최대 1986등을 기록했다. 실시간으로 순위는 변동된다.

**6.4 향후 연구 방향**

Cabin 결측값을 임의의 값으로 부여하는 방법을 사용해보고자 하였으나 실패하였다. 추가로 Cabin 값이나 다른 결측값, 구간 조정을 통해 정확도 추가 상승의 가능성이 있다.

**참 고 문 헌**

[1] [Mukesh ChapagainTitanic Solution: A Beginner's Guide](<https://www.kaggle.com/chapagain/titanic-solution-a-beginner-s-guide?scriptVersionId=1473689> )

[2] [How to score 0.8134 in Titanic Kaggle Challenge](<http://ahmedbesbes.com/how-to-score-08134-in-titanic-kaggle-challenge.html> )

[3] [Titanic: factors to survive](<https://olegleyz.github.io/titanic_factors.html> )

[4] [Titanic Survivors Dataset and Data Wrangling](<http://www.codeastar.com/data-wrangling/> )

[5] [캐글 - 타이타닉 생존자 예측하기](<https://www.youtube.com/watch?v=FAP7JOECfEE> )

[6] [Minseok Heo]([https://github.com/minsuk-heo/kaggle-titanic/tree/master](https://github.com/minsuk-heo/kaggle-titanic](https://github.com/minsuk-heo/kaggle-titanic/tree/master%5d(https://github.com/minsuk-heo/kaggle-titanic) )

[7] [Age Hint](<https://medium.com/@praveen.orvakanti/this-will-help-you-score-95-percentile-in-the-kaggle-titanic-ml-competition-aa2b3fd1b79b> )

[8] [Name Hint](<https://www.ahmedbesbes.com/blog/kaggle-titanic-competition> )

[9] [OneHotEncoding](<https://programmers.co.kr/learn/courses/21/lessons/11044> )