Titanic Code Analysis Summary

Jieun Choi

January 24, 2022

Chung-Ang University

CONTENTs

01. Titanic Data

Titanic Data

train.csv

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

test.csv

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

Titanic Data

Step 1. outlier detection (IQR 사용해서 이상치 제거 작업)

Step2. Joining train set and test set to obtain the same number of features during categorical conversion

Step3. check for null and missing values

Feature_name	Is.null_sum
PassengerId	0
Survived	418
Pclass	0
Name	0
Sex	0
Age	256
SibsP	0
Parch	0
Ticket	0
Fare	1
Cabin	1007
embarked	2

Age and Cabin features have an important part of missing values

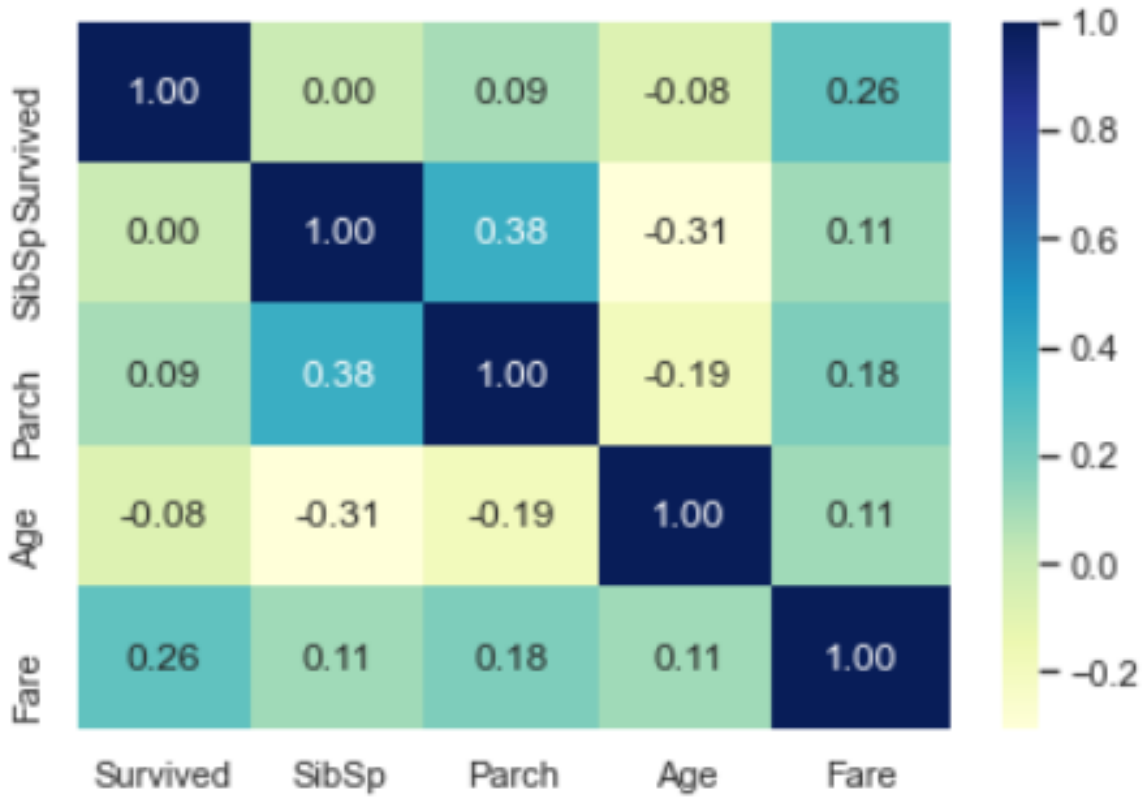
02. Feature Analysis

- Numerical values (SibSP, Parch, Age, Fare)
- Categorical values (Sex, Pclass, Embarked)

02. Feature Analysis - Numerical values

Correlation matrix between numerical values.

```
g = sns.heatmap(train[["Survived","SibSp","Parch","Age","Fare"]].corr(),annot=True, fmt = ".2f", cmap = "coolwarm")
```

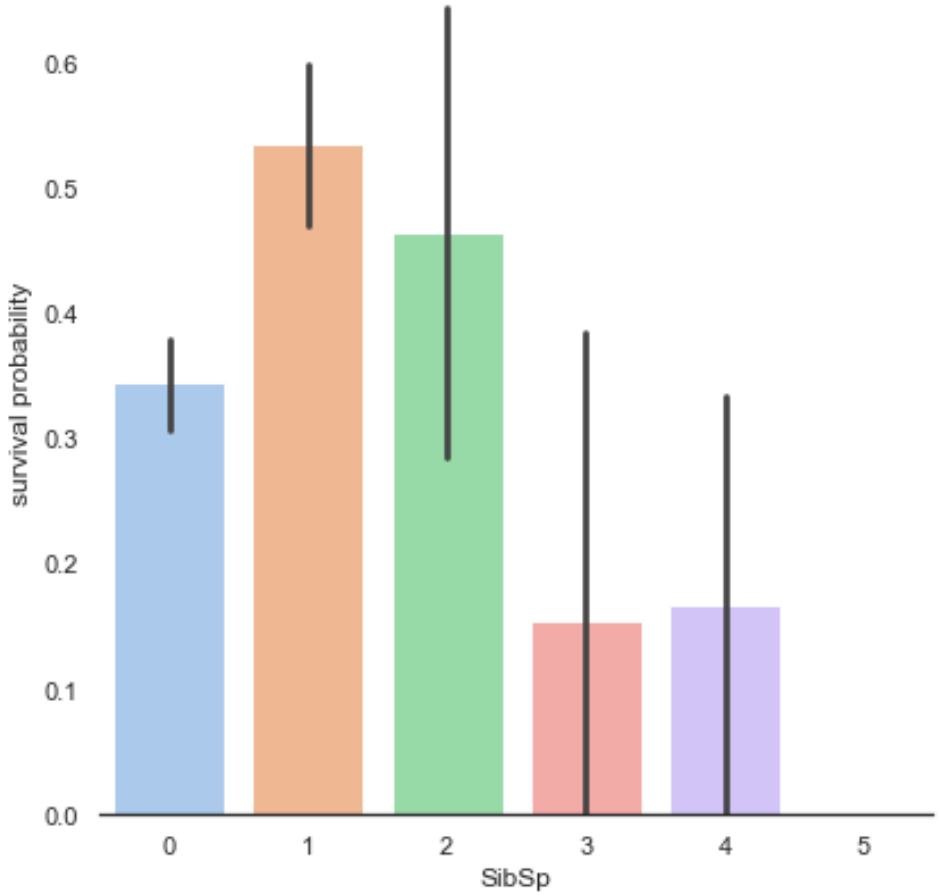


- Correlation matrix를 통해,
  Fare가 생존가능성 변수와 상관관계가
  유의미함을 알 수 있음.

- 다른 변수도 유용하지 않다는건 아니라서
  각 변수들을 세부적으로 분석하는 것이 중요함.

02. Feature Analysis - Numerical values

Explore SibSP features vs Survived

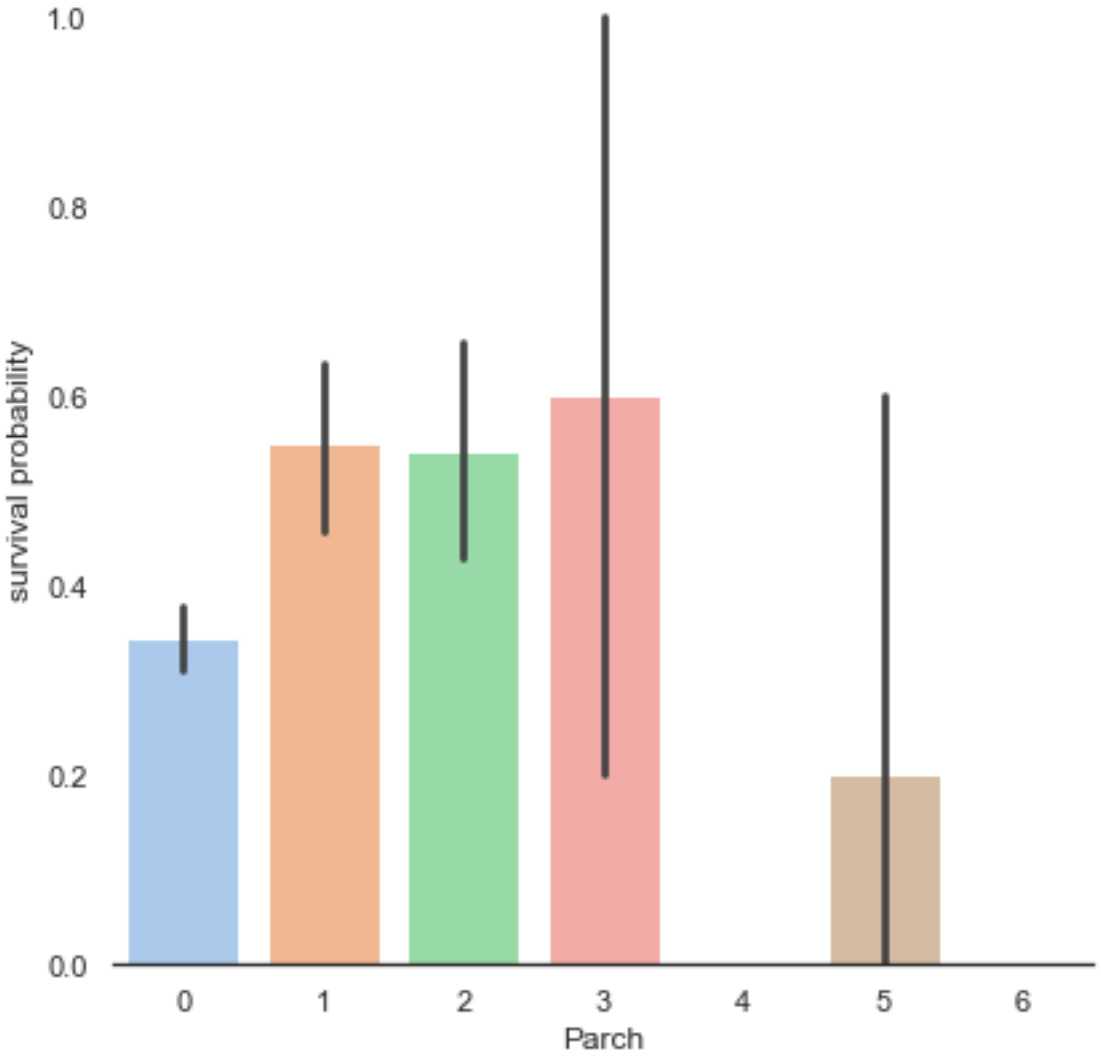


형제 및 배우자가 많은 승객은 생존가능성이 낮아보임.

혼자온 승객이나 둘이 같이 온 승객이 생존할 확률 더 높음.

02. Feature Analysis - Numerical values

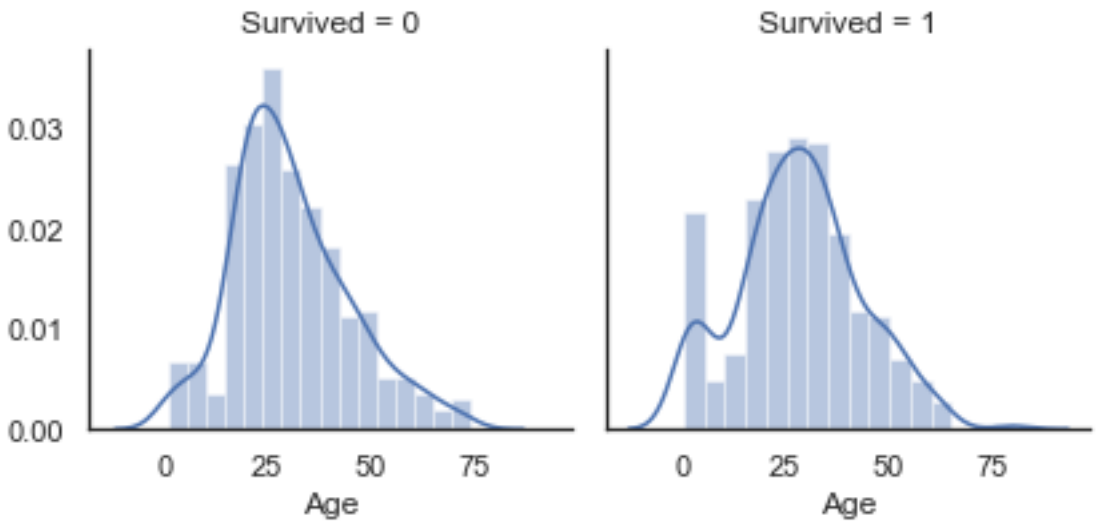
Explore Parch features vs Survived

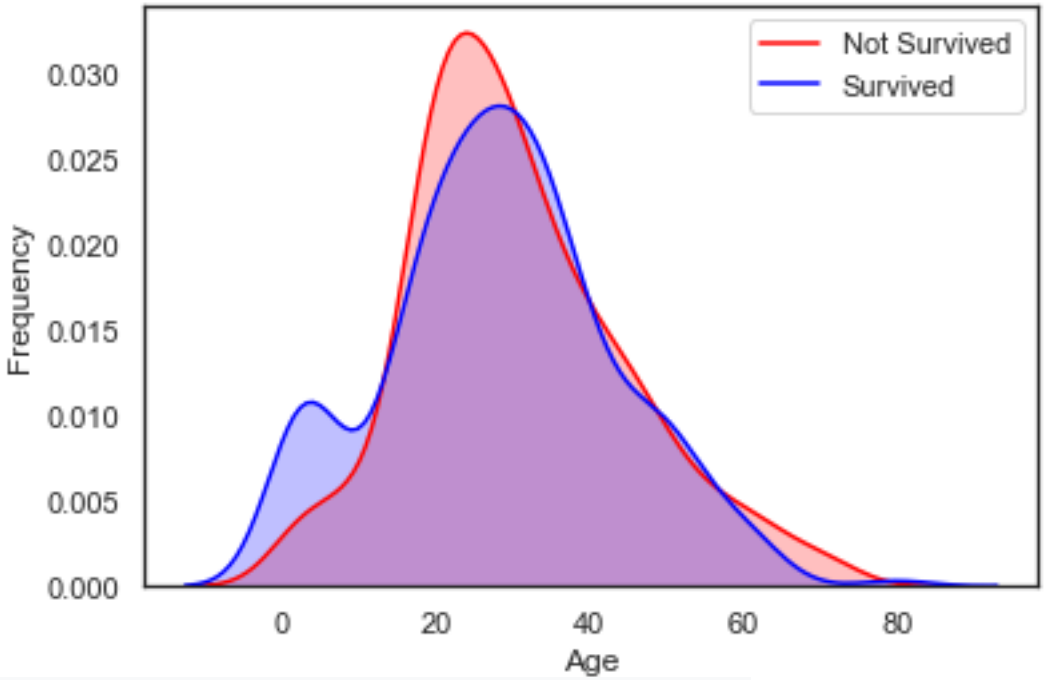


가족수가 적은 경우가 생존가능성 높음.
Parch가 3인 승객의 경우 생존에 중요한 표준편차가 있으므로 주의할것.

02. Feature Analysis - Numerical values

Explore Age features vs Survived





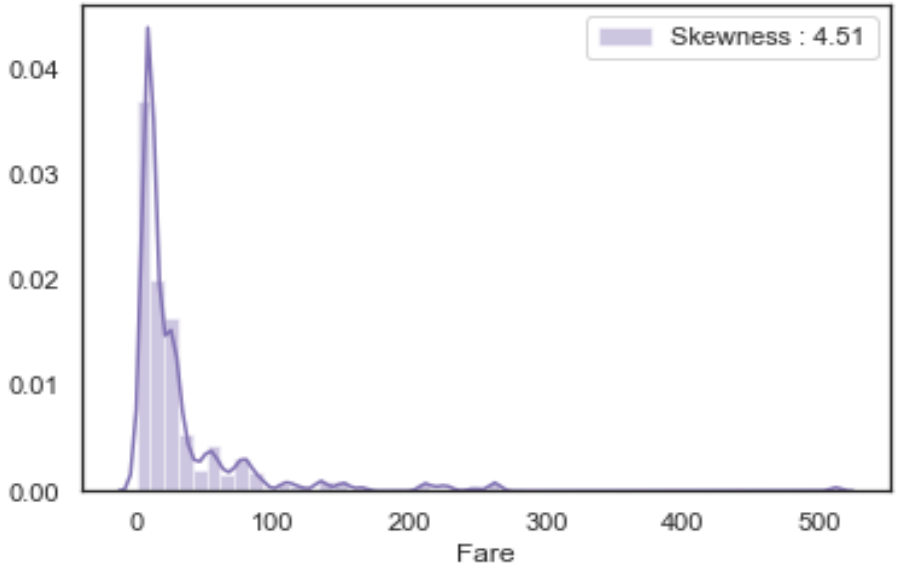
연령 분포는 생존집단과 생존하지 않은 집단 둘이 다름.

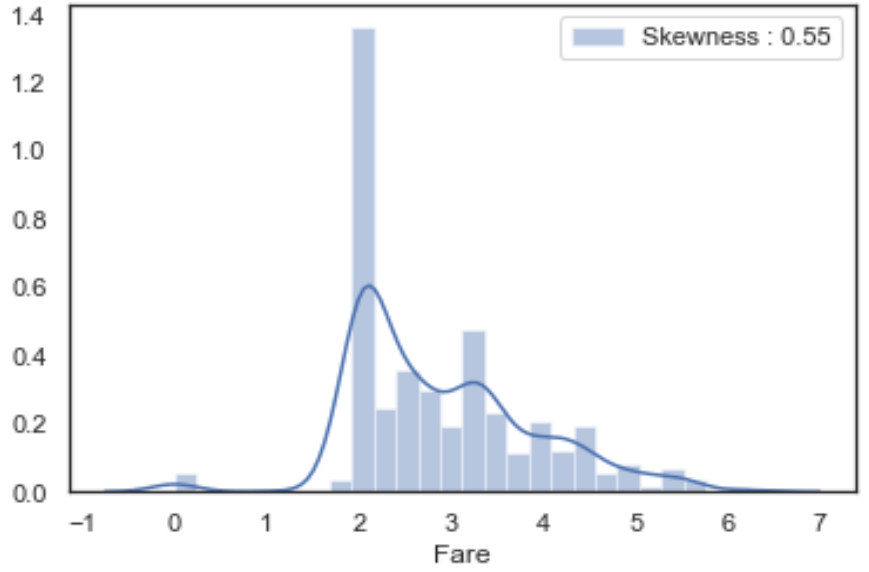
60-80세 사이의 승객은 생존가능성이 낮았음.

아주 어린 승객이 생존할 가능성이 더 높은 것 같음..

Explore Fare features

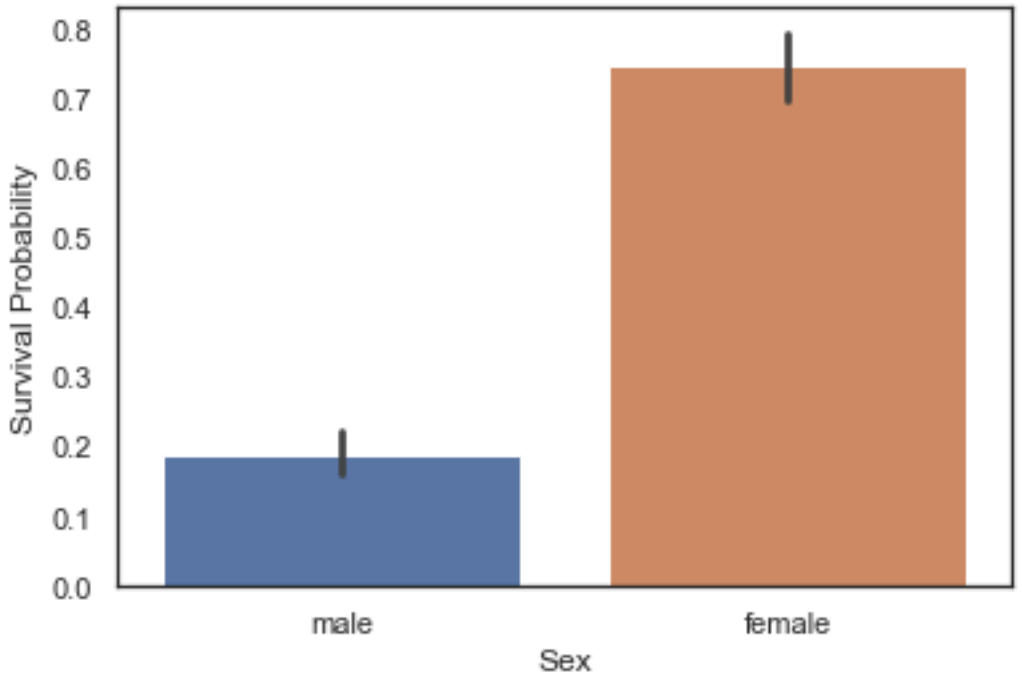
1. Fare 변수 null값 sum 확인.
2. Missing values을 median으로 채우기 (missin이 1개라서 하나정도 median으로 채우는건 예측에 딱히 영향없을듯)
3. Fare 변수 분포확인 후 편향이 심하고 scale하더라도 high values가 overweight하기 때문에 log function으로 변환 해주는게 더 나음.
4. Log function 적용 후 편향 줄어듬을 확인.





02. Feature Analysis - Categorical values

Explore Sex features



Female 평균 생존 가능성 0.748

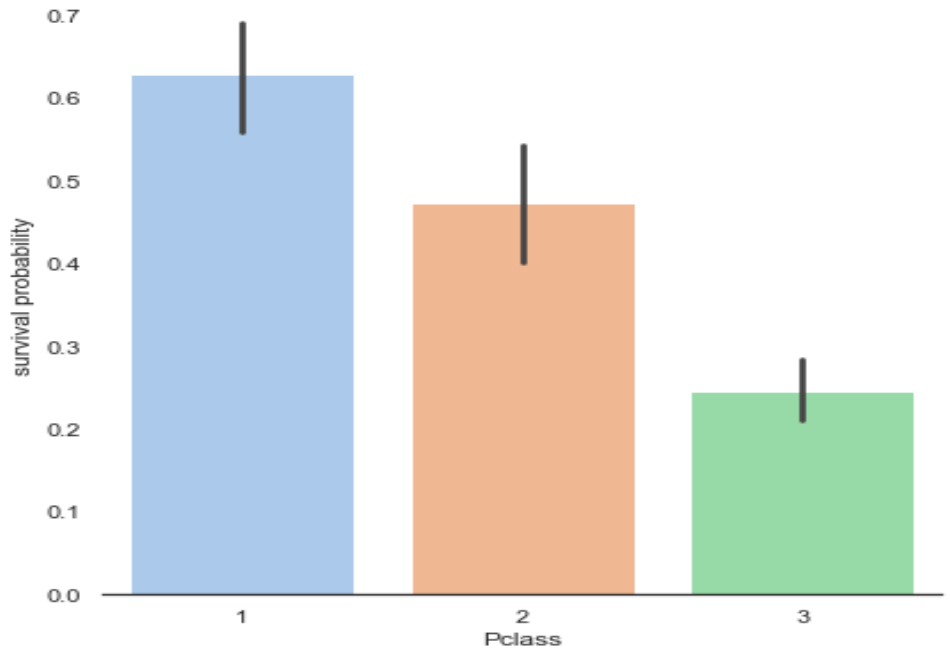
Male 평균 생존 가능성 0.191

남성이 여성보다 생존 확률이 낮음을 확인.

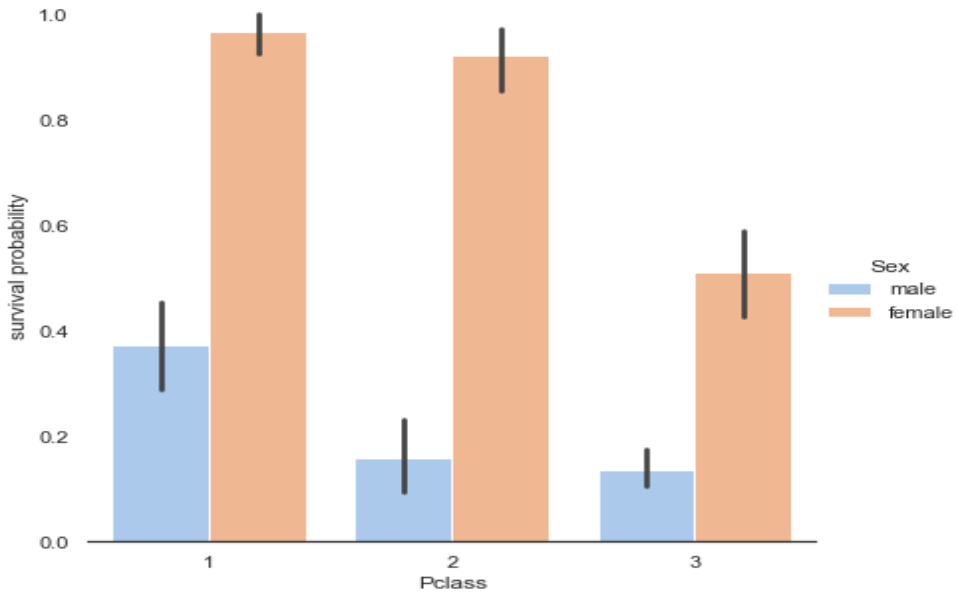
따라서 성별은 생존 예측에 중요한 역할을 할 수 있음.

02. Feature Analysis - Categorical values

Explore Pclass vs Survived



Explore Pclass vs Survived by sex



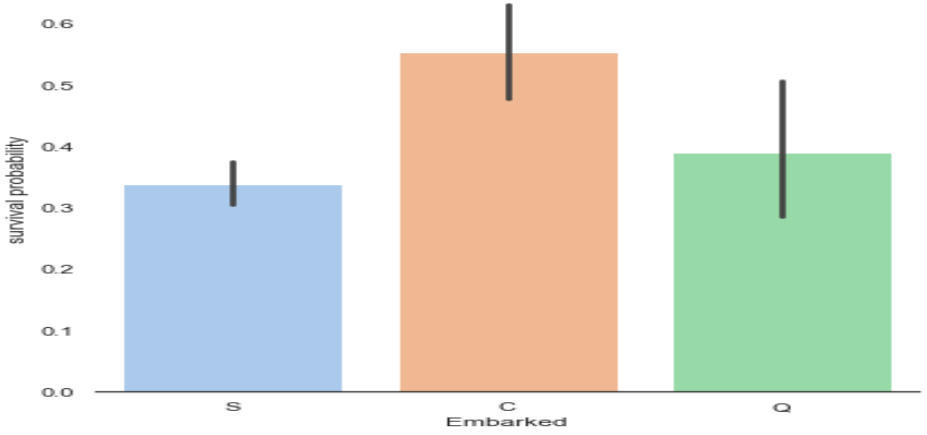
1등석 승객은 2등석 및 3등석 승객보다 생존 가능성이 더 높음.

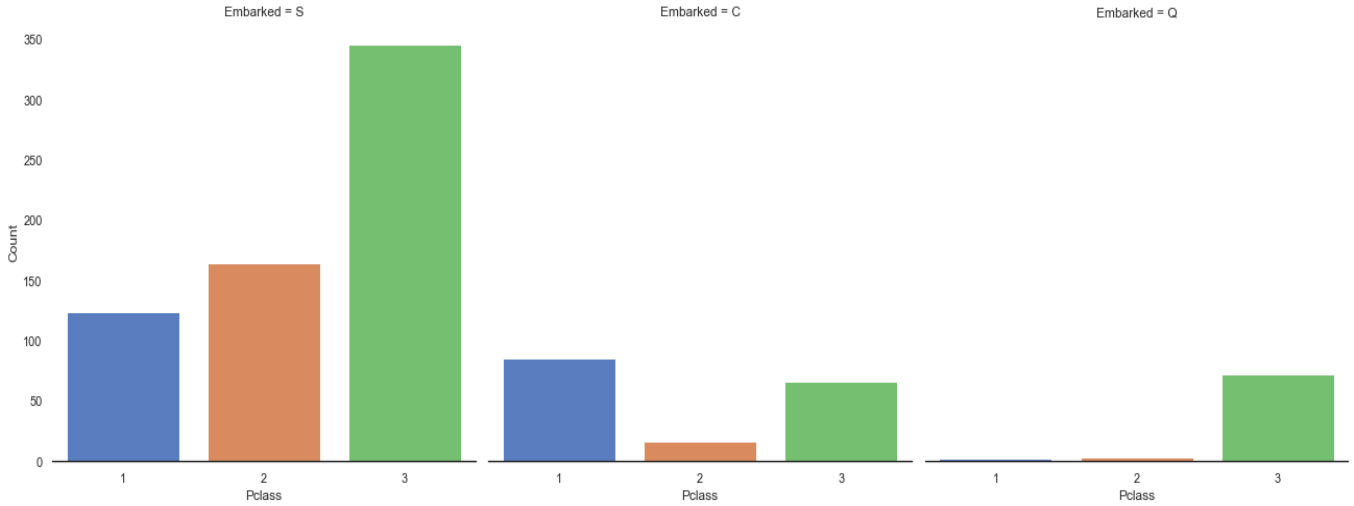
이러한 경향은 성별을 고려해도 마찬가지임.

02. Feature Analysis - Categorical values

1. Null 값 확인
2. 결측값 2개 있는것을 빈도가 가장 큰 'S' 지역으로 채우는 아이디어

Explore Embarked vs Survived





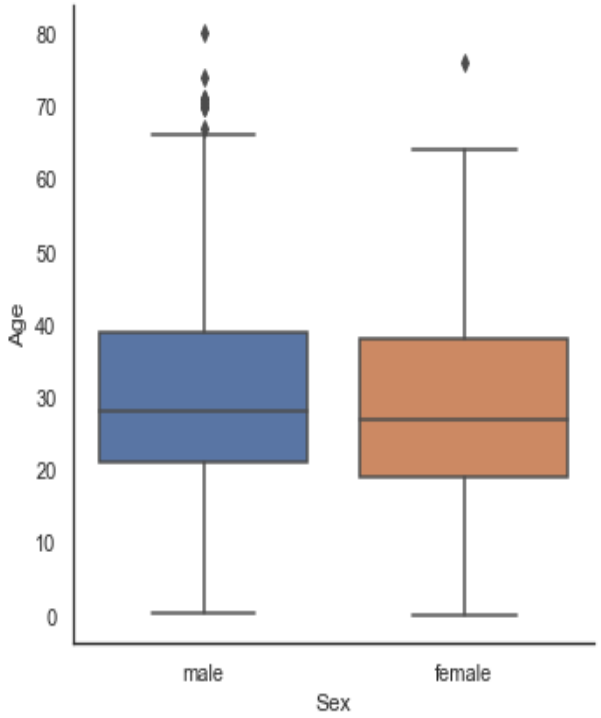
셰르부르(C)에서 오는 승객은 생존 가능성이 더 높음.
1등석 승객의 비율이 Queenstown(Q), Southampton(S)보다 Cherbourg에서 오는 승객의 비율이 더 높음.

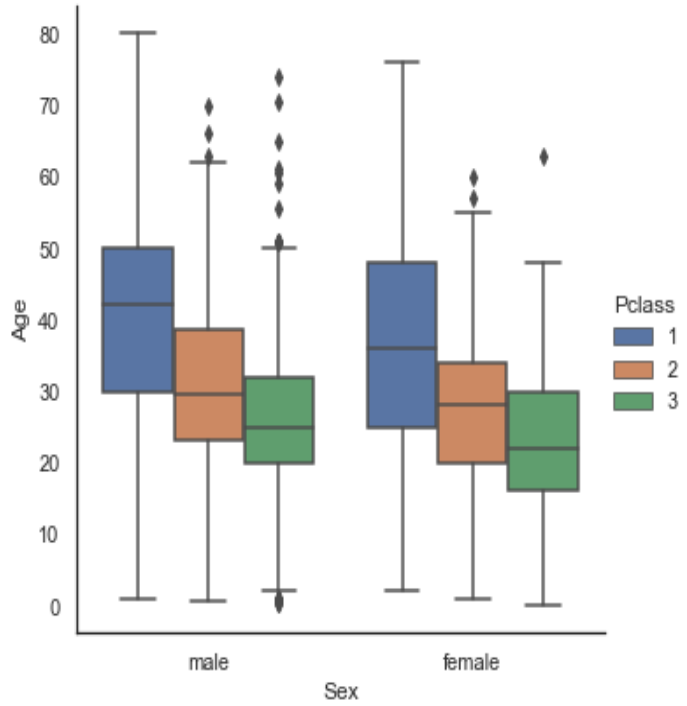
3등석은 Southampton(S) 및 Queenstown(Q)에서 오는 승객에게 가장 빈번한 반면 Cherbourg 승객은 대부분 생존율이 가장 높은 1등석임.
1등석 승객이 영향을 받아 대피 과정에서 우선 순위가 높은 것으로 보임.

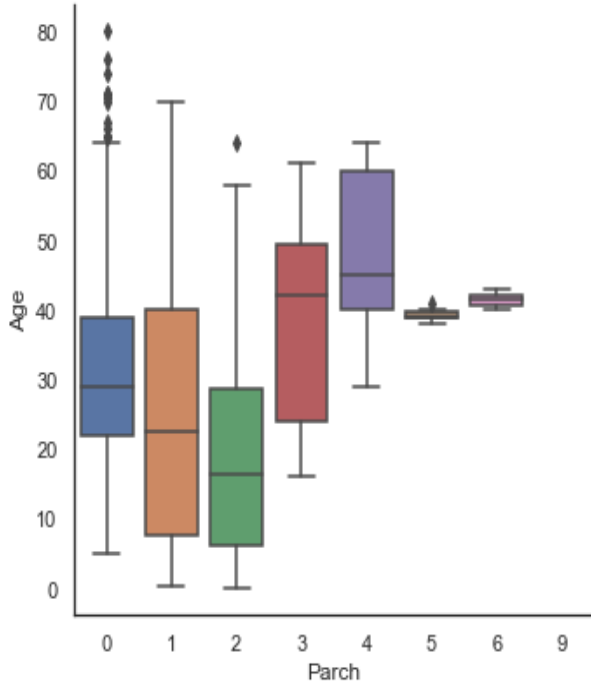
02. Feature Analysis – Filling missing Values

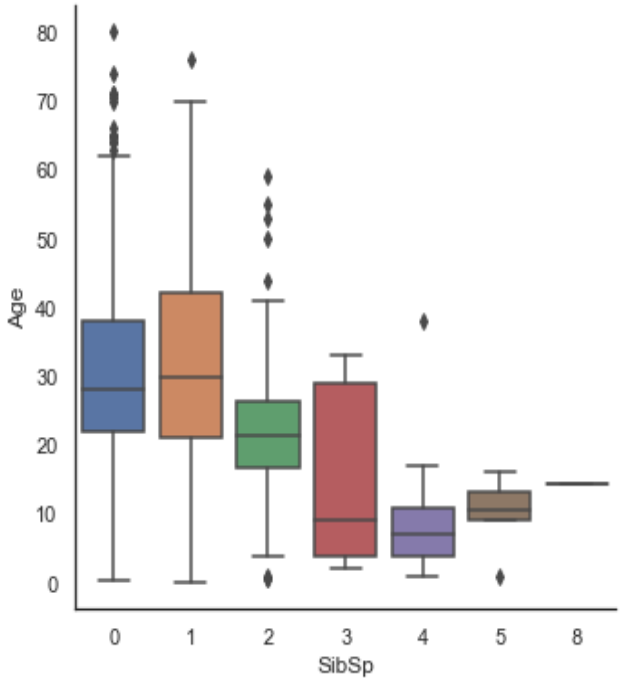
NA 처리 아이디어. 전체중에 256개가 missing value.

age 변수를 분석하면서 나이가 적을수록 생존가능성이 높아지는 특성을 발견했다.
이 특성을 유지하도록 하는 방법 중 하나로 age와 다른 변수들 간의 상관관계를 살펴본다.

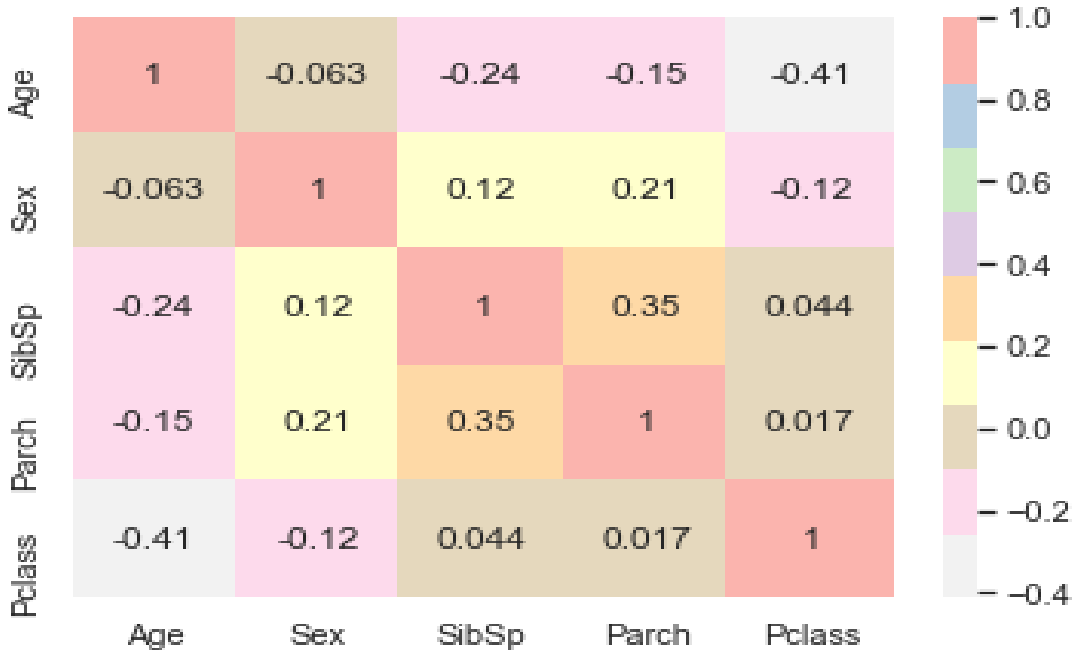








02. Feature Analysis – Filling missing Values




```
dataset["Sex"] = dataset["Sex"].map({"male": 0, "female":1})
```

성별 변수를 0,1로 범주화 한 후에 상관관계 살펴봄.

나이는 성별과 상관관계 없음.

Pclass, parch, sibsp와는 음의 상관관계가 있음.

⇒ Missing value 추정을 위해서 sibsp, parch, pclass 활용하기로 결정.

```
index_NaN_age = list(dataset["Age"][dataset["Age"].isnull()].index)

for i in index_NaN_age :
    age_med = dataset["Age"].median()
    age_pred = dataset["Age"][((dataset['SibSp'] == dataset.iloc[i]["SibSp"]) &
                               (dataset['Parch'] == dataset.iloc[i]["Parch"]) &
                               (dataset['Pclass'] == dataset.iloc[i]["Pclass"]))].median()
    if not np.isnan(age_pred) :
        dataset['Age'].iloc[i] = age_pred
    else :
        dataset['Age'].iloc[i] = age_med
```

<= 세 변수들과 유사한 행을 찾아서 그 행의 중앙값으로 Age를 채우겠다는 아이디어가 돋보임.

03. Feature Engineering

Feature Engineering

1. 승객의 Name에는 직위에 대한 정보가 있을 것이라고 생각하여 Title 변수를 새로 생성하여 4개로 카테고리화함.

2. 대가족은 서로의 가족들을 찾느라 대피하는데 어려움이 있을 것이라고 생각해서 family size라는 변수(sib네+parch+1)로 새로 생성하여 가족의 규모가 생존가능성에 영향을 줌을 알아냄.
(이 안에서 또 가족규모를 범주화하여 중규모가족이 대가족보다 생존기회가 더 많다는 것을 알아냄)

3. Cabin 변수에는 1007개의 missing value가 존재.
Cabin의 첫글자는 승객의 위치를 나타내서 이 정보를 활용하기로.
Cabin이 있는 승객이 없는 승객보다 생존가능성이 높음을 확인.

4. 티켓의 앞글자가 케빈을 예약할 수 있어서 케빈의 실제 배치로 이어질 수 있음. 서로 앞글자가 같은 티켓은 생존확률이 비슷할 수 있다고 생각해서 티켓 칼럼을 앞글자로 바꿔서 더많은 정보를 얻음.

Thank you ☺