Titanic Team B

Data science and Artificial Intelligence Society

Team Member: Yujin Choi, Sangmok Lee, Heera Lee, Nakyung Lee

Predicting the Survival of Titanic Passengers

Topic: Data Analysis, Machine Learning

Expected Duration: 4 weeks

Titanic Tutorial

- Titanic Tutorial Blog
- Divided into 5 parts
    - Part 1: Check data (WK 1)
    - Part 2: Exploratory Data Analysis (WK 2)
    - Part 3: Exploratory Data Analysis (WK 3)
    - Part 4: Feature engineering (WK 3,4)
    - Part 5: Build Machine Learning Model and Prediction (WK 4)

https://kaggle-kr.tistory.com/17?category=868316#2_5

0. Setting Library

- Titanic survival data from Kaggle
- Data visualization:
    - Matplotlib
    - Seaborn
    - Plotly
- Data analysis:
    - Pandas
    - Numpy
- Machine Learning Tool:
    - Sklearn

Source Link:
https://www.kaggle.com/c/titanic/data?select=test.csv

```
[7]  import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns

     plt.style.use('seaborn')
     sns.set(font_scale=2.5)

     import missingno as msno

     import warnings
     warnings.filterwarnings('ignore')

     %matplotlib inline
```

1. Check Data

- Titanic survival data from Kaggle
- Used the following data for analysis
  - Check NULL data
  - Target Label Distribution (Survived)

Source Link:
https://www.kaggle.com/c/titanic/data?select=test.csv

```
[ ] for col in df_train.columns:
        msg = 'column: {:>10}₩t Percent of NaN value: {:.2f}%'.format(col, 100 * (df_train[col].isnull().sum() / df_train[col].shape[0]))
        print(msg)

    column: PassengerId      Percent of NaN value: 0.00%
    column:   Survived       Percent of NaN value: 0.00%
    column:     Pclass       Percent of NaN value: 0.00%
    column:       Name       Percent of NaN value: 0.00%
    column:        Sex       Percent of NaN value: 0.00%
    column:        Age       Percent of NaN value: 19.87%
    column:      SibSp       Percent of NaN value: 0.00%
    column:      Parch       Percent of NaN value: 0.00%
    column:     Ticket       Percent of NaN value: 0.00%
    column:       Fare       Percent of NaN value: 0.00%
    column:      Cabin       Percent of NaN value: 77.10%
    column:   Embarked       Percent of NaN value: 0.22%
```

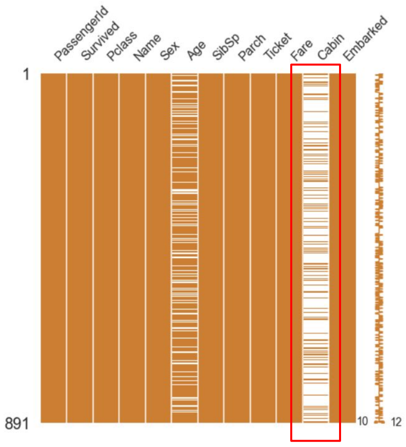
```
[ ] for col in df_test.columns:
        msg = 'column: {:>10}₩t Percent of NaN value: {:.2f}%'.format(col, 100 * (df_test[col].isnull().sum() / df_test[col].shape[0]))
        print(msg)

    column: PassengerId      Percent of NaN value: 0.00%
    column:     Pclass       Percent of NaN value: 0.00%
    column:       Name       Percent of NaN value: 0.00%
    column:        Sex       Percent of NaN value: 0.00%
    column:        Age       Percent of NaN value: 20.57%
    column:      SibSp       Percent of NaN value: 0.00%
    column:      Parch       Percent of NaN value: 0.00%
    column:     Ticket       Percent of NaN value: 0.00%
    column:       Fare       Percent of NaN value: 0.24%
    column:      Cabin       Percent of NaN value: 78.23%
    column:   Embarked       Percent of NaN value: 0.00%
```

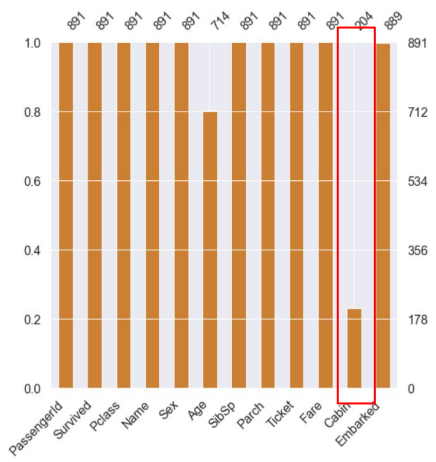
1. Check Data

- Titanic survival data from Kaggle
- Used the following data for analysis
    - Check NULL data
    - Target Label Distribution (Survived)

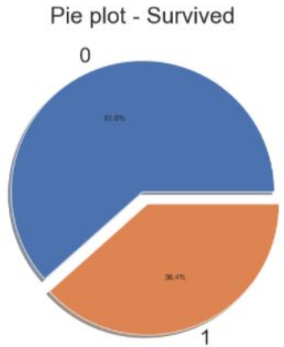
Source Link:
https://www.kaggle.com/c/titanic/data?select=test.csv

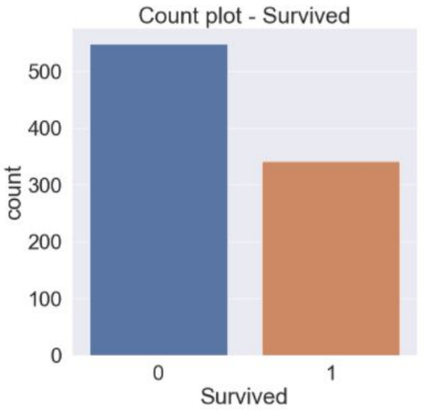
- Check NULL data





- Target label distribution (Survived)

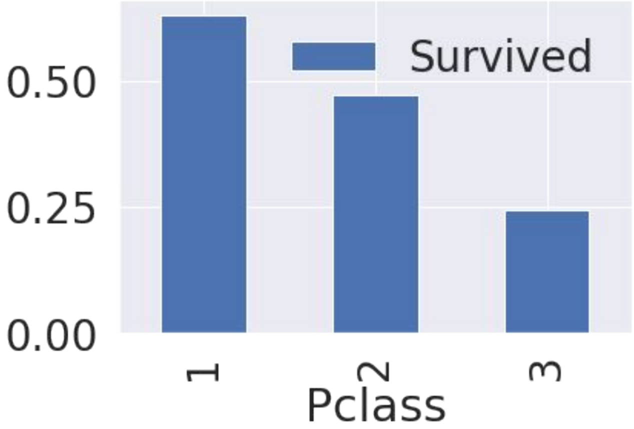


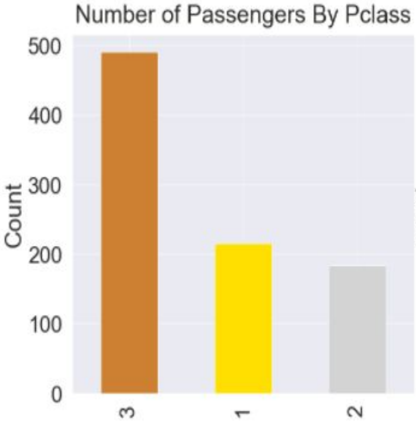


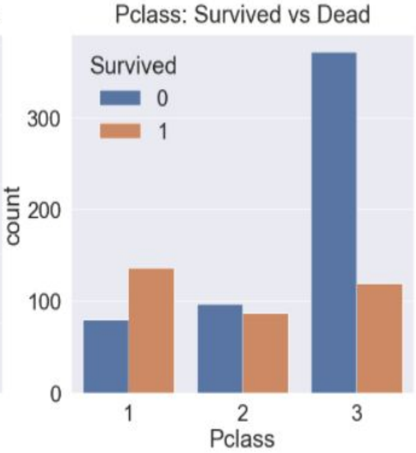
2. Exploratory Data Analysis - Pclass

- Graph Visualization using matplotlib and seaborn
- Higher the class, higher the survival rate
- Conclusion: Pclass is an important factor that should be considered for predicting survival

Survival Rate depend on Pclass

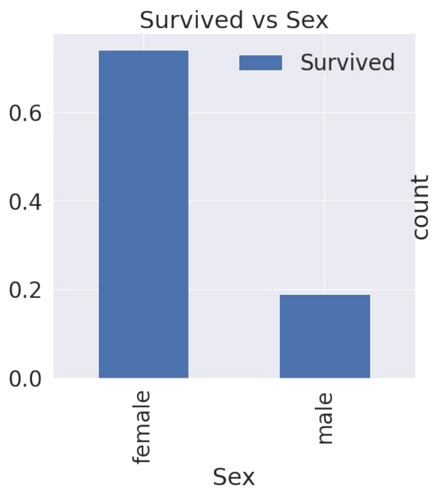


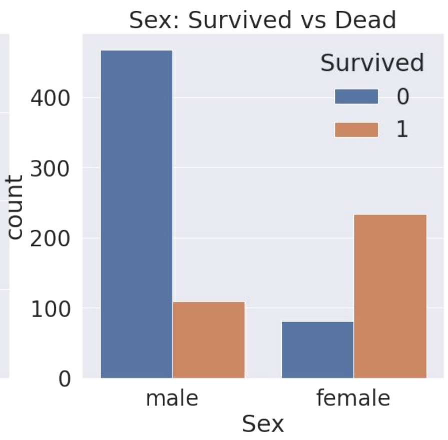




2. Exploratory Data Analysis
- Sex

- Graph Visualization using Pandas and seaborn
- Female survival rate is higher than that of Male
- Conclusion: Sex is an important factor that should be considered for predicting survival





Survival Rate depend on Sex

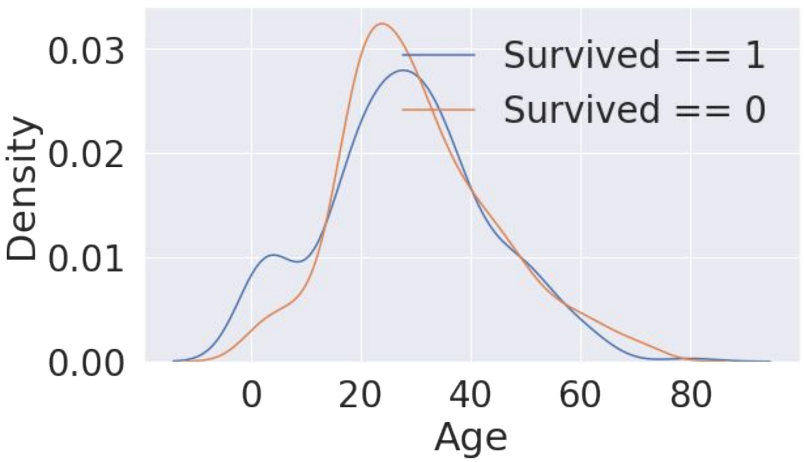
Survived	0	1	All
Sex			
female	81	233	314
male	468	109	577
All	549	342	891

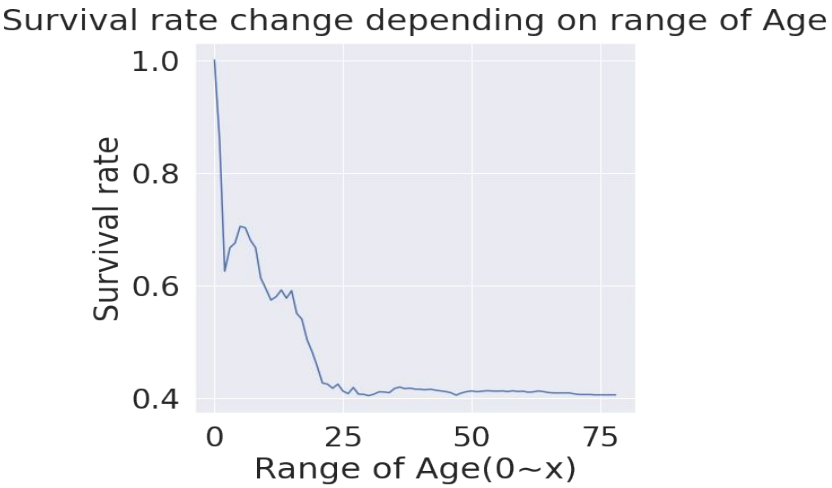
	Sex	Survived
0	female	0.742038
1	male	0.188908

2. Exploratory Data Analysis - Age

- Graph Visualization using matplotlib and seaborn
- Younger the age, higher the survival rate
- Conclusion: Age is an important factor that should be considered for predicting survival

Age Histogram





2. Exploratory Data Analysis
- Pclass, Sex, Age

- Graph Visualization using violinplot of seaborn

- Conclusion:
  - In all Pclasses, the younger passengers survived more
  - Female survived more than male

Titanic Tutorial

- Titanic Tutorial Blog
- Divided into 5 parts
    - Part 1: Check data (WK 1)
    - Part 2: Exploratory Data Analysis (WK 2)
    - **Part 3: Exploratory Data Analysis (WK 3)**
    - **Part 4: Feature engineering (WK 3,4)**
    - Part 5: Build Machine Learning Model and Prediction (WK 4)

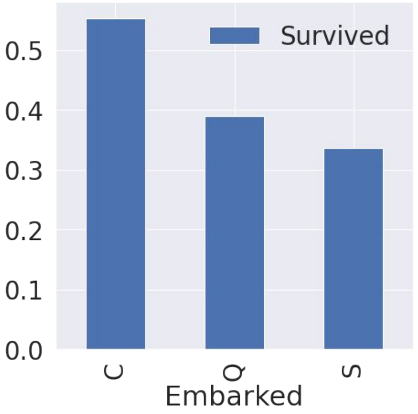
https://kaggle-kr.tistory.com/17?category=868316#2_5

2. Exploratory Data Analysis - Embarked

- Graph Visualization using matplotlib and seaborn

- Passenger embarked at C survived more than others
- Conclusion: Highest survival rate at C is because the higher Pclass passengers boarded at C.

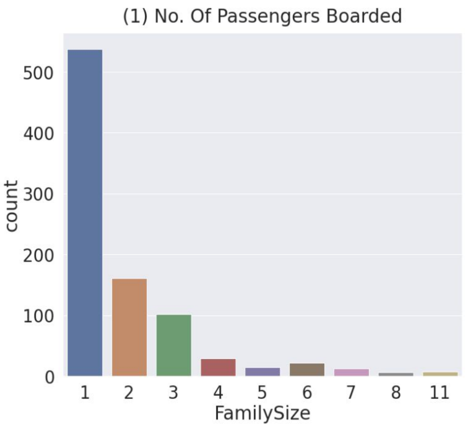
Survival Rate depend on Embarked

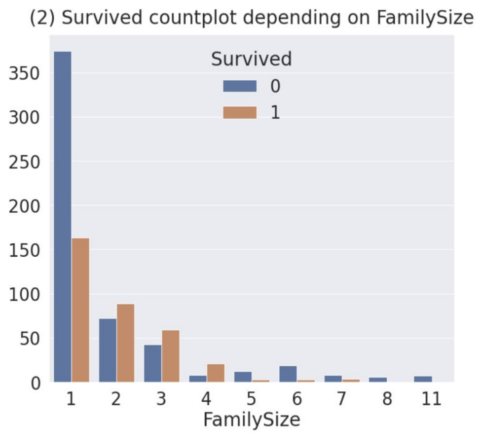


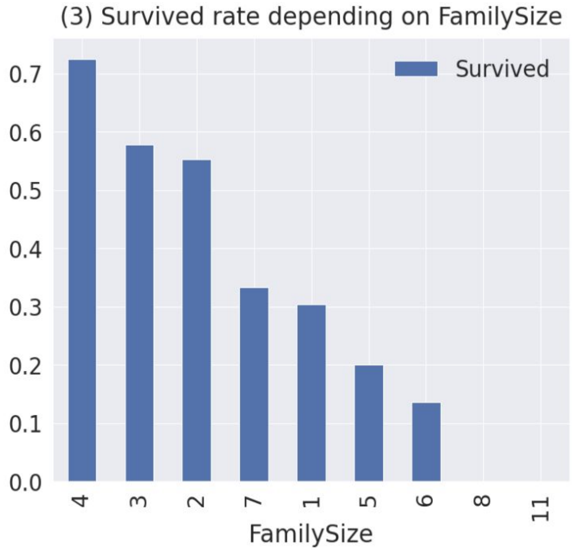
2. Exploratory Data Analysis
- Family (SibSp + Parch)

- Graph Visualization using Pandas and seaborn

- Family member = Sibling + Parent&Children
- Family size from 1 to 11
- Conclusion: Family size between 2 to 4 has the highest survival rate

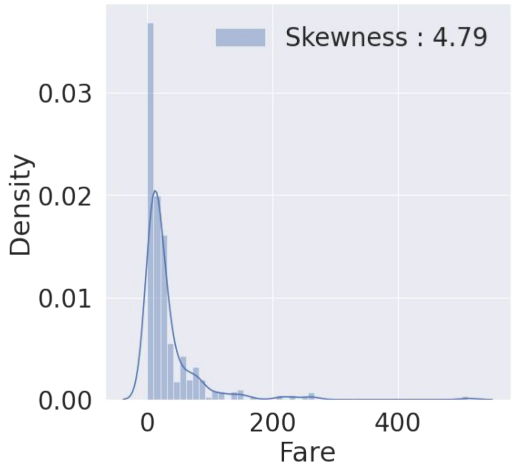


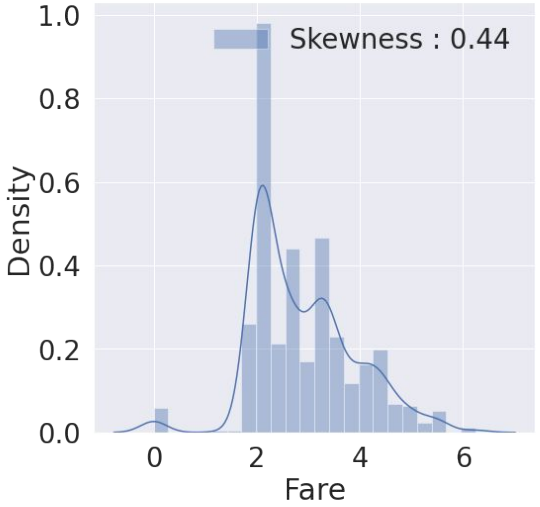




2. Exploratory Data Analysis - Fare

- Graph Visualization using matplotlib and seaborn
- Modifying skewness by having log on fare data

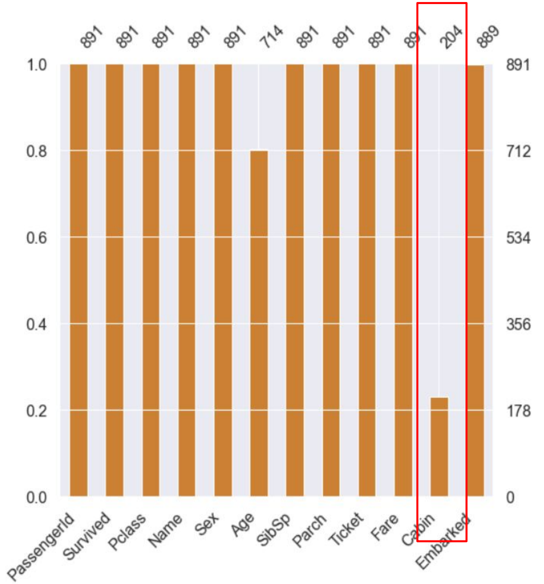




2. Exploratory Data Analysis
- Cabin & Ticket

- Cabin:
  Null value about 80%, difficult to relate the feature with survival
- Ticket:
  Various ticket numbers, difficult to relate the feature with survival

- Conclusion:
  Will not include Cabin and Ticket in model formation.




```
df_train['Ticket'].value_counts()

347082      7
CA. 2343    7
1601        7
3101295     6
CA 2144     6
           ..
9234        1
19988       1
2693        1
PC 17612    1
370376      1
Name: Ticket, Length: 681, dtype: int64
```

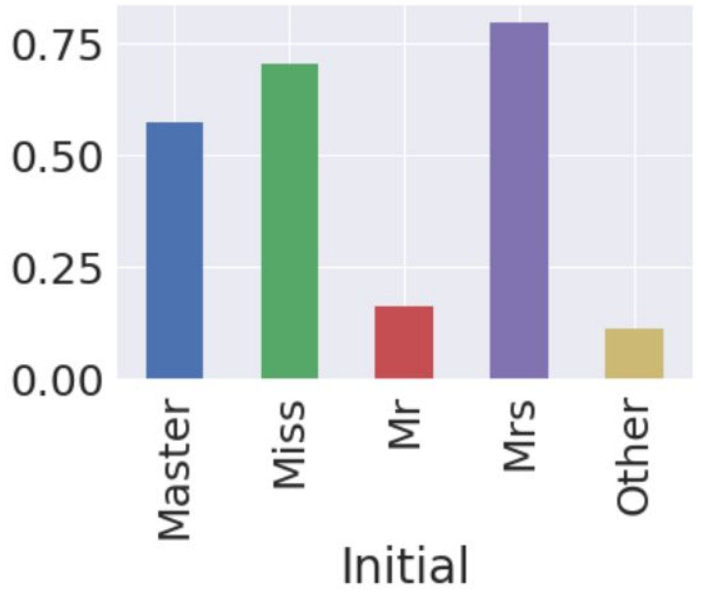
3. Feature engineering
- Replace initial title

Initial	Capt	Col	Countess	Don	Dr	Jonkheer	Lady	Major	Master	Miss	Mlle	Mme	Mr	Mrs	Ms	Rev	Sir
Sex																	
female	0	0	1	0	1	0	1	0	0	182	2	1	0	125	1	0	0
male	1	2	0	1	6	1	0	2	40	0	0	0	517	0	0	6	1

- 17 initial titles were replaced by 5 titles (Master, Miss, Mr, Mrs, Other)

- Graph Visualization using matplotlib

- Conclusion:
  - Female group(Miss, Mrs) has a higher survived rate

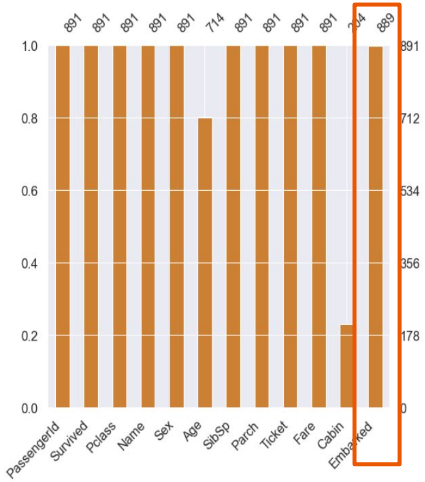


3. Feature engineering
- Fill Age, Embarked

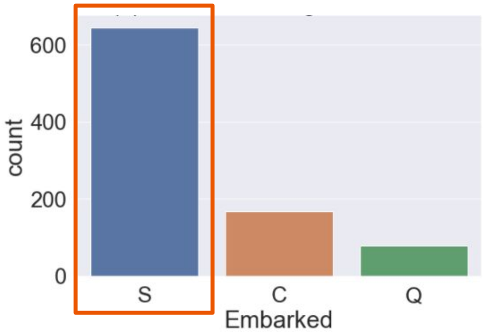
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	FamilySize
Initial								
Master	414.975000	0.575000	2.625000	4.574167	2.300000	1.375000	3.340710	4.675000
Miss	411.741935	0.704301	2.284946	21.860000	0.698925	0.537634	3.123713	2.236559
Mr	455.880907	0.162571	2.381853	32.739609	0.293006	0.151229	2.651507	1.444234
Mrs	456.393701	0.795276	1.984252	35.981818	0.692913	0.818898	3.443751	2.511811
Other	564.444444	0.111111	1.666667	45.888889	0.111111	0.111111	2.641605	1.222222

- Fill null value of age using the average age of each title

- Fill null value of embarked with the S value
  - The count of embarked null values is 2
  - The most embarked is S



Value count



Embarked count

3. Feature engineering

- Change Age(continuous to categorical) using ‘loc’ method

- Change Initial, Embarked, and Sex(string to numerical) using ‘map’ mehod

- Graph Visualization using the heatmap plot
- Sex, Pcalss, Fare are correlated with Survived

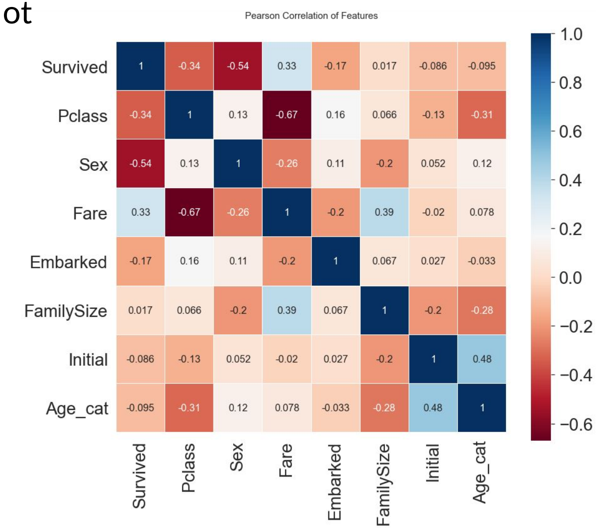
‘Loc’ method (continuous to categorical)

```
# Age_cat: category
df_train['Age_cat'] = 0
df_train.loc[df_train['Age'] < 10, 'Age_cat'] = 0
df_train.loc[(10 <= df_train['Age']) & (df_train['Age'] < 20), 'Age_cat'] = 1
df_train.loc[(20 <= df_train['Age']) & (df_train['Age'] < 30), 'Age_cat'] = 2
df_train.loc[(30 <= df_train['Age']) & (df_train['Age'] < 40), 'Age_cat'] = 3
df_train.loc[(40 <= df_train['Age']) & (df_train['Age'] < 50), 'Age_cat'] = 4
df_train.loc[(50 <= df_train['Age']) & (df_train['Age'] < 60), 'Age_cat'] = 5
df_train.loc[(60 <= df_train['Age']) & (df_train['Age'] < 70), 'Age_cat'] = 6
df_train.loc[70 <= df_train['Age'], 'Age_cat'] = 7
```

‘Map’ method (String to numerical)

```
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['Embarked'] = df_train['Embarked'].map({'C': 0, 'Q': 1, 'S': 2})
df_test['Embarked'] = df_test['Embarked'].map({'C': 0, 'Q': 1, 'S': 2})
df_train['Sex'] = df_train['Sex'].map({'female':0, 'male':1})
df_test['Sex'] = df_test['Sex'].map({'female':0, 'male':1})
```

Heatmap plot



3. Feature engineering _ One hot encoding

- Create fifth dimensional vector in train set regarding title and embarked
- Use get_dummies in Pandas

	Initial_Master	Initial_Miss	Initial_Mr	Initial_Mrs	Initial_Other
Master	1	0	0	0	0
Miss	0	1	0	0	0
Mr	0	0	1	0	0
Mrs	0	0	0	1	0
Other	0	0	0	0	1

```
df_train.head()
```

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	FamilySize	Age_cat	Initial_0	Initial_1	Initial_2	Initial_3	Initial_4	Embarked_0	Embarked_1	Embarked_2
0	1	0	3	Braund, Mr. Owen Harris	1.0	1	0	A/5 21171	1.981001	NaN	2	2	0	0	1	0	0	0	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	NaN	1	0	PC 17599	4.266662	C85	2	3	0	0	0	1	0	1	0	0
2	3	1	3	Heikkinen, Miss. Laina	NaN	0	0	STON/O2. 3101282	2.070022	NaN	1	2	0	1	0	0	0	0	0	1
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	NaN	1	0	113803	3.972177	C123	2	3	0	0	0	1	0	0	0	1
4	5	0	3	Allen, Mr. William Henry	1.0	0	0	373450	2.085672	NaN	1	3	0	0	1	0	0	0	0	1

3. Feature engineering - Drop columns

- Delete unnecessary columns using the 'drop' function
  - Before drop

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	Embarked	FamilySize	Age_cat	Initial_0	Initial_1	Initial_2	Initia
0	1	0	3	Braund, Mr. Owen Harris	1	1	0	A/5 21171	1.981001	NaN	2	2	2	0	0	1	
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	
2	3	1	3	Heikkinen, Miss. Laina	0	0	0	STON/O2. 3101282	2.070022	NaN	2	1	2	0	1	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	
4	5	0	3	Allen, Mr. William Henry	1	0	0	373450	2.085672	NaN	2	1	3	0	0	1	

  - After drop

	Survived	Pclass	Sex	Fare	FamilySize	Age_cat	Initial_0	Initial_1	Initial_2	Initial_3	Initial_4	Embarked_0	Embarked_1	Embarked_2
0	0	3	1	1.981001	2	2	0	0	1	0	0	0	0	1
1	1	1	0	4.266662	2	3	0	0	0	1	0	1	0	0
2	1	3	0	2.070022	1	2	0	1	0	0	0	0	0	1
3	1	1	0	3.972177	2	3	0	0	0	1	0	0	0	1
4	0	3	1	2.085672	1	3	0	0	1	0	0	0	0	1

Titanic Tutorial

- Titanic Tutorial Blog
- Divided into 5 parts
    - Part 1: Check data (WK 1)
    - Part 2: Exploratory Data Analysis (WK 2)
    - Part 3: Exploratory Data Analysis (WK 3)
    - Part 4: Feature engineering (WK 3,4)
    - **Part 5: Build Machine Learning Model and Prediction (WK 4)**

https://kaggle-kr.tistory.com/17?category=868316#2_5

0. Setting Library

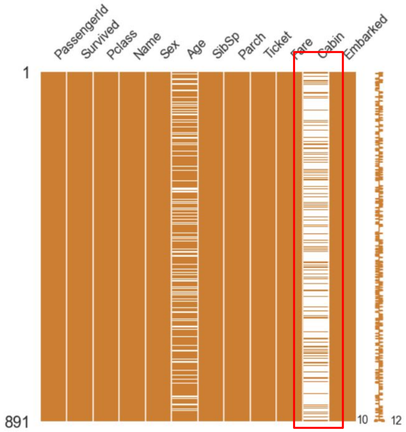
- Titanic survival data from Kaggle
- Data visualization:
  - Matplotlib
  - Seaborn
  - Plotly
- Data analysis:
  - Pandas
  - Numpy
- Machine Learning Tool:
  - Sklearn

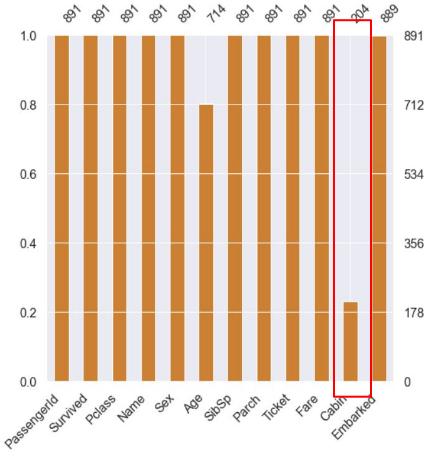
```
[7]  import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns

     plt.style.use('seaborn')
     sns.set(font_scale=2.5)

     import missingno as msno

     import warnings
     warnings.filterwarnings('ignore')

     %matplotlib inline
```

1. Check Dataset & Null

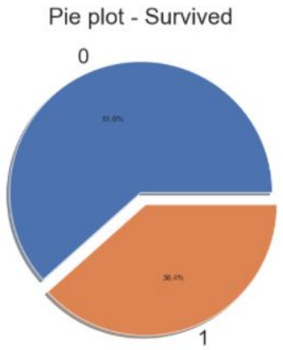
- Titanic survival data from Kaggle
- Used the following data for analysis
  - Check NULL data
  - Target Label Distribution (Survived)

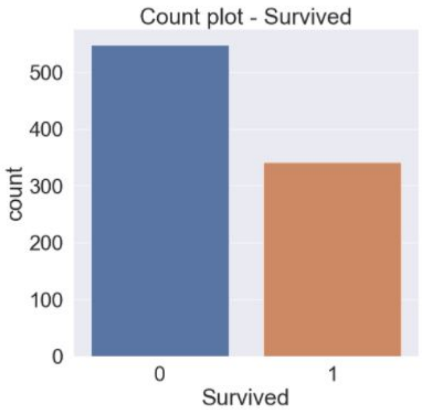
- Check NULL data





- Target label distribution (Survived)

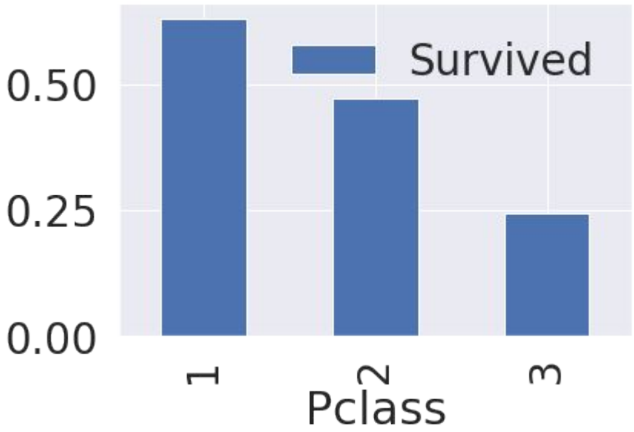


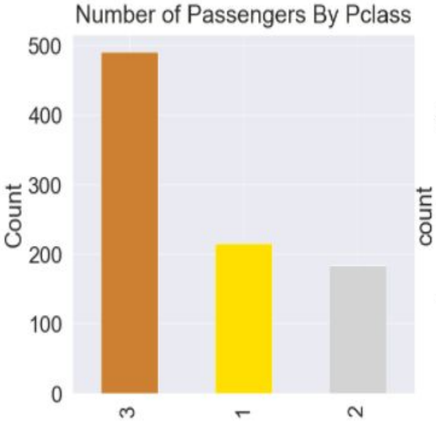


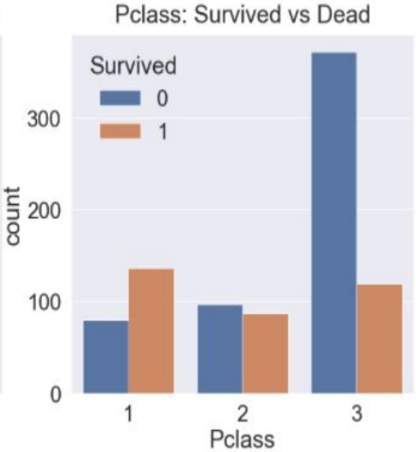
2. Exploratory Data Analysis
- Pclass

- Graph Visualization using matplotlib and seaborn
- Higher the class, higher the survival rate
- Conclusion: Pclass is an important factor that should be considered for predicting survival

Survival Rate depend on Pclass

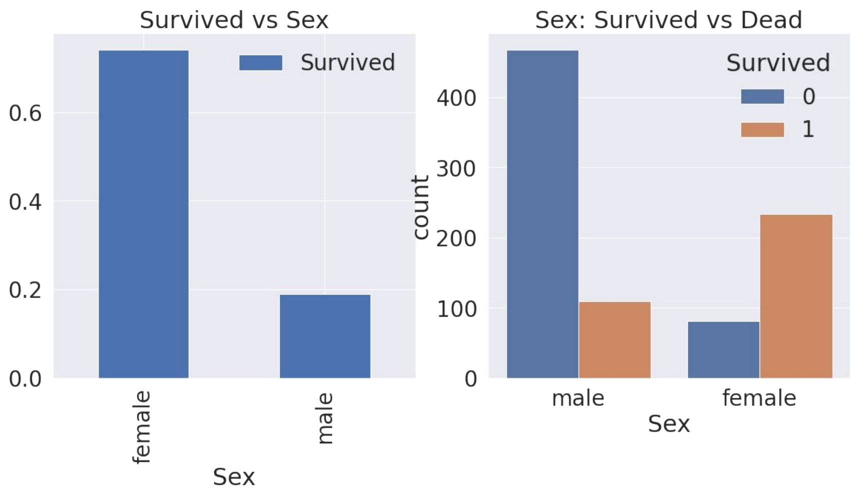






2. Exploratory Data Analysis
- Sex

- Graph Visualization using Pandas and seaborn
- Female survival rate is higher than that of Male
- Conclusion: Sex is an important factor that should be considered for predicting survival



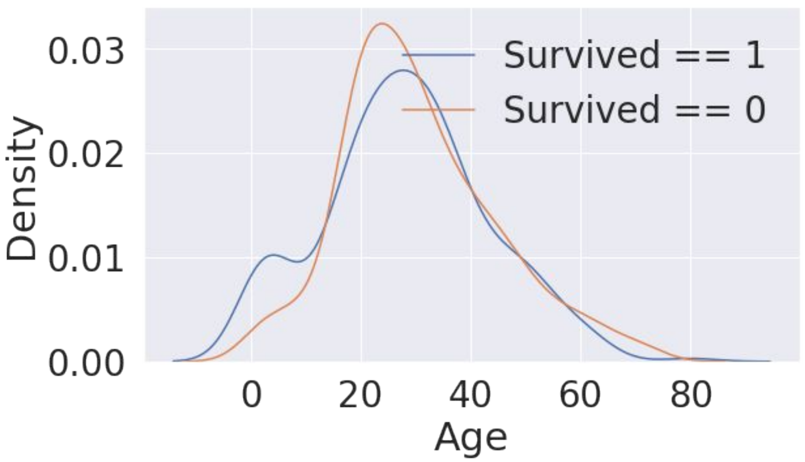
Survival Rate depend on Sex

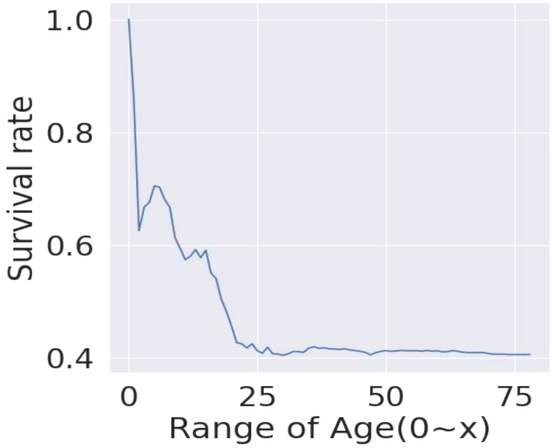
	Sex	Survived
0	female	0.742038
1	male	0.188908

2. Exploratory Data Analysis
- Age

- Graph Visualization using matplotlib and seaborn
- Younger the age, higher the survival rate
- Conclusion: Age is an important factor that should be considered for predicting survival

Age Histogram

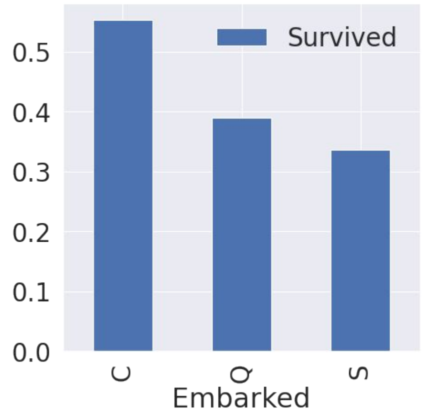




2. Exploratory Data Analysis - Embarked

- Graph Visualization using matplotlib and seaborn
- Passenger embarked at C survived more than others
- Conclusion: Highest survival rate at C is because the higher Pclass passengers boarded at C.

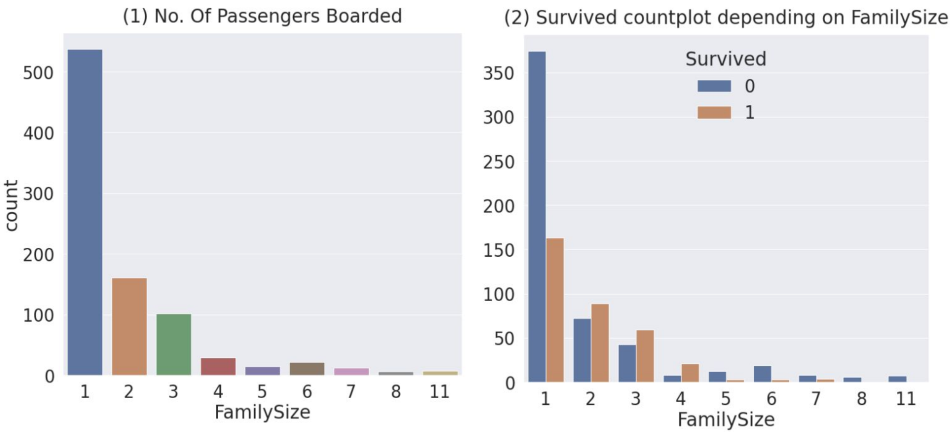
Survival Rate depend on Embarked

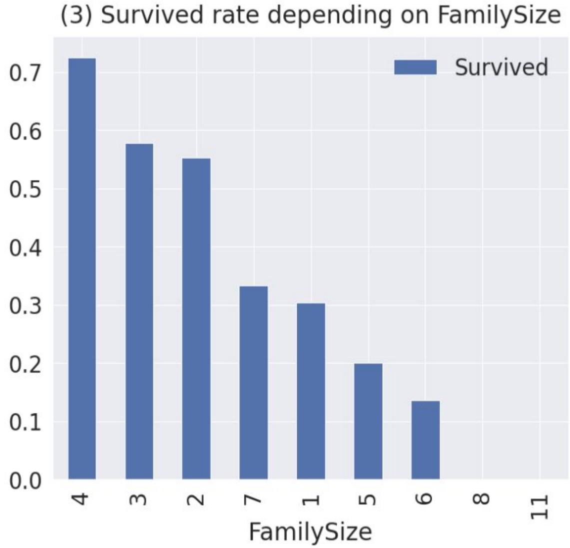


2. Exploratory Data Analysis - Family (SibSp + Parch)

- Graph Visualization using Pandas and seaborn

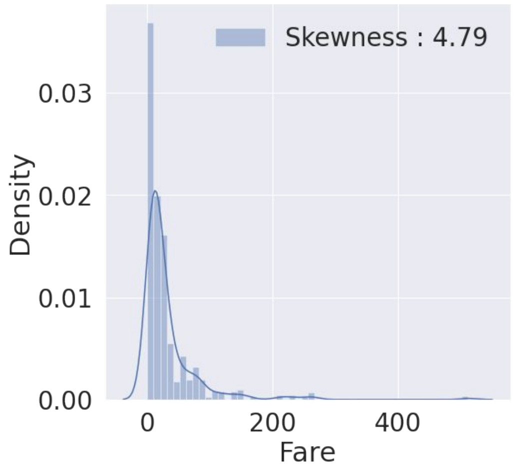
- Family member = Sibling + Parent&Children
- Family size from 1 to 11
- Conclusion: Family size between 2 to 4 has the highest survival rate

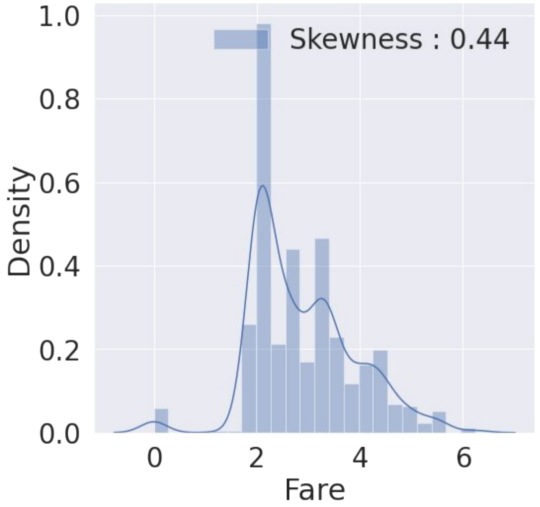




2. Exploratory Data Analysis
- Fare

- Graph Visualization using matplotlib and seaborn
- Modifying skewness by having log on fare data

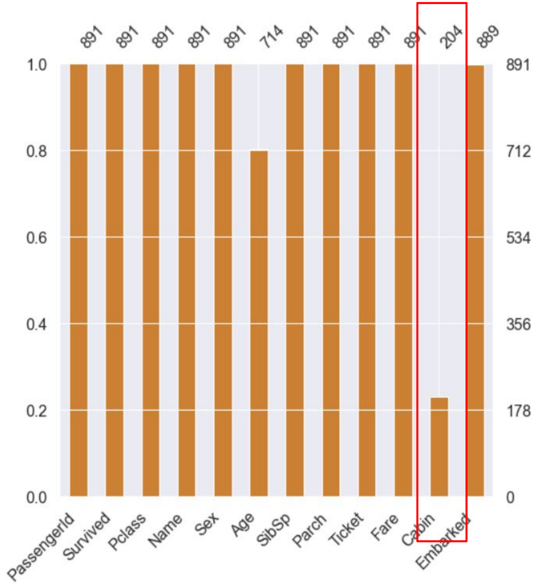




2. Exploratory Data Analysis
- Cabin & Ticket

- Cabin:
  Null value about 80%, difficult to relate the feature with survival
- Ticket:
  Various ticket numbers, difficult to relate the feature with survival

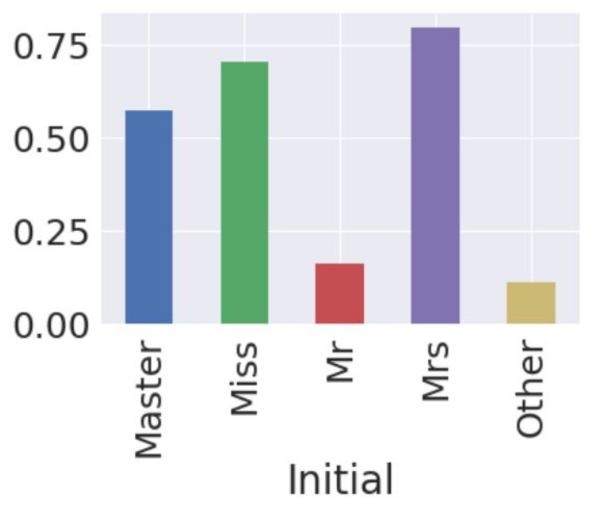
- Conclusion:
  Will not include Cabin and Ticket in model formation.



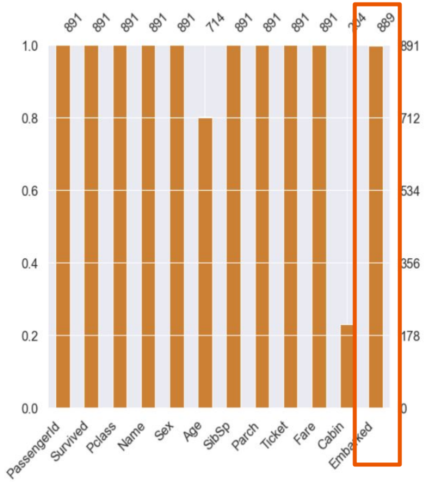

```
df_train['Ticket'].value_counts()

347082      7
CA. 2343    7
1601        7
3101295     6
CA 2144     6
           ..
9234        1
19988       1
2693        1
PC 17612    1
370376      1
Name: Ticket, Length: 681, dtype: int64
```

3. Feature engineering

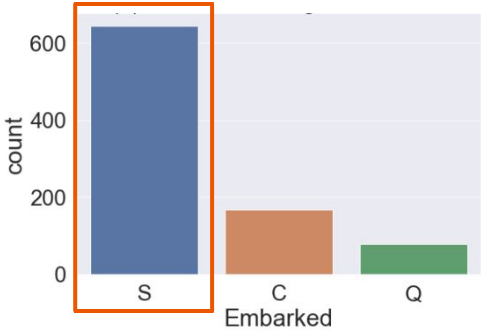
Initial Sex	Capt	Col	Countess	Don	Dr	Jonkheer	Lady	Major	Master	Miss	Mlle	Mme	Mr	Mrs	Ms	Rev	Sir
female	0	0	1	0	1	0	1	0	0	182	2	1	0	125	1	0	0
male	1	2	0	1	6	1	0	2	40	0	0	0	517	0	0	6	1



Initial	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	FamilySize
Master	414.975000	0.575000	2.625000	4.574167	2.300000	1.375000	3.340710	4.675000
Miss	411.741935	0.704301	2.284946	21.860000	0.698925	0.537634	3.123713	2.236559
Mr	455.880907	0.162571	2.381853	32.739609	0.293006	0.151229	2.651507	1.444234
Mrs	456.393701	0.795276	1.984252	35.981818	0.692913	0.818898	3.443751	2.511811
Other	564.444444	0.111111	1.666667	45.888889	0.111111	0.111111	2.641605	1.222222



Value count



Embarked count

3. Feature engineering

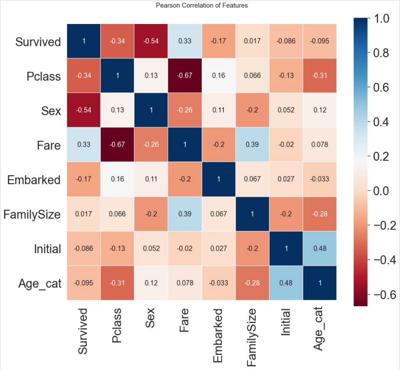
‘Loc’ method (continuous to categorical)

```
# Age_cat: category
df_train['Age_cat'] = 0
df_train.loc[df_train['Age'] < 10, 'Age_cat'] = 0
df_train.loc[(10 <= df_train['Age']) & (df_train['Age'] < 20), 'Age_cat'] = 1
df_train.loc[(20 <= df_train['Age']) & (df_train['Age'] < 30), 'Age_cat'] = 2
df_train.loc[(30 <= df_train['Age']) & (df_train['Age'] < 40), 'Age_cat'] = 3
df_train.loc[(40 <= df_train['Age']) & (df_train['Age'] < 50), 'Age_cat'] = 4
df_train.loc[(50 <= df_train['Age']) & (df_train['Age'] < 60), 'Age_cat'] = 5
df_train.loc[(60 <= df_train['Age']) & (df_train['Age'] < 70), 'Age_cat'] = 6
df_train.loc[70 <= df_train['Age'], 'Age_cat'] = 7
```

‘Map’ method (String to numerical)

```
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['Embarked'] = df_train['Embarked'].map({'C': 0, 'Q': 1, 'S': 2})
df_test['Embarked'] = df_test['Embarked'].map({'C': 0, 'Q': 1, 'S': 2})
df_train['Sex'] = df_train['Sex'].map({'female':0, 'male':1})
df_test['Sex'] = df_test['Sex'].map({'female':0, 'male':1})
```

Heatmap plot



- Delete unnecessary columns using the ‘drop’ function

  - Before drop

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	Embarked	FamilySize	Age_cat	Initial_0	Initial_1	Initial_2	Initia
0	1	0	3	Braund, Mr. Owen Harris	1	1	0	A/5 21171	1.981001	NaN	2	2	2	0	0	1	
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	
2	3	1	3	Heikkinen, Miss. Laina	0	0	0	STON/O2. 3101282	2.070022	NaN	2	1	2	0	1	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	
4	5	0	3	Allen, Mr. William Henry	1	0	0	373450	2.085672	NaN	2	1	3	0	0	1	

  - After drop

	Survived	Pclass	Sex	Fare	FamilySize	Age_cat	Initial_0	Initial_1	Initial_2	Initial_3	Initial_4	Embarked_0	Embarked_1	Embarked_2
0	0	3	1	1.981001	2	2	0	0	1	0	0	0	0	1
1	1	1	0	4.266662	2	3	0	0	0	1	0	1	0	0
2	1	3	0	2.070022	1	2	0	1	0	0	0	0	0	1
3	1	1	0	3.972177	2	3	0	0	0	1	0	0	0	1
4	0	3	1	2.085672	1	3	0	0	1	0	0	0	0	1

3. Feature engineering _ One hot encoding

- Create fifth dimensional vector in train set regarding title and embarked
- Use get_dummies in Pandas

	Initial_Master	Initial_Miss	Initial_Mr	Initial_Mrs	Initial_Other
Master	1	0	0	0	0
Miss	0	1	0	0	0
Mr	0	0	1	0	0
Mrs	0	0	0	1	0
Other	0	0	0	0	1

```
df_train.head()
```

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	FamilySize	Age_cat	Initial_0	Initial_1	Initial_2	Initial_3	Initial_4	Embarked_0	Embarked_1	Embarked_2
0	1	0	3	Braund, Mr. Owen Harris	1.0	1	0	A/5 21171	1.981001	NaN	2	2	0	0	1	0	0	0	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	NaN	1	0	PC 17599	4.266662	C85	2	3	0	0	0	1	0	1	0	0
2	3	1	3	Heikkinen, Miss. Laina	NaN	0	0	STON/O2. 3101282	2.070022	NaN	1	2	0	1	0	0	0	0	0	1
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	NaN	1	0	113803	3.972177	C123	2	3	0	0	0	1	0	0	0	1
4	5	0	3	Allen, Mr. William Henry	1.0	0	0	373450	2.085672	NaN	1	3	0	0	1	0	0	0	0	1

3. Feature engineering - Drop columns

- Delete unnecessary columns using the ‘drop’ function
  - Before drop

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	Embarked	FamilySize	Age_cat	Initial_0	Initial_1	Initial_2	Initia
0	1	0	3	Braund, Mr. Owen Harris	1	1	0	A/5 21171	1.981001	NaN	2	2	2	0	0	1	
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	
2	3	1	3	Heikkinen, Miss. Laina	0	0	0	STON/O2. 3101282	2.070022	NaN	2	1	2	0	1	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	
4	5	0	3	Allen, Mr. William Henry	1	0	0	373450	2.085672	NaN	2	1	3	0	0	1	

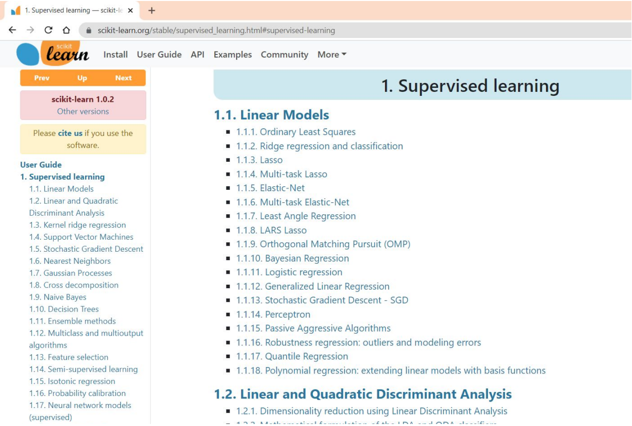
  - After drop

	Survived	Pclass	Sex	Fare	FamilySize	Age_cat	Initial_0	Initial_1	Initial_2	Initial_3	Initial_4	Embarked_0	Embarked_1	Embarked_2
0	0	3	1	1.981001	2	2	0	0	1	0	0	0	0	1
1	1	1	0	4.266662	2	3	0	0	0	1	0	1	0	0
2	1	3	0	2.070022	1	2	0	1	0	0	0	0	0	1
3	1	1	0	3.972177	2	3	0	0	0	1	0	0	0	1
4	0	3	1	2.085672	1	3	0	0	1	0	0	0	0	1

4. Building machine learning model and prediction

- Importing all the required ML packages (sklearn - RandomForest)

```python
#importing all the required ML packages
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
from sklearn.model_selection import train_test_split
```



http://scikit-learn.org/stable/supervised_learning.html#supervised-learning

- Split dataset into train, valid, test set

```python
X_train = df_train.drop('Survived', axis=1).values
target_label = df_train['Survived'].values
X_test = df_test.values

X_tr, X_vld, y_tr, y_vld = train_test_split
(X_train, target_label, test_size=0.3, random_state=2018)
```

- Model generation and prediction

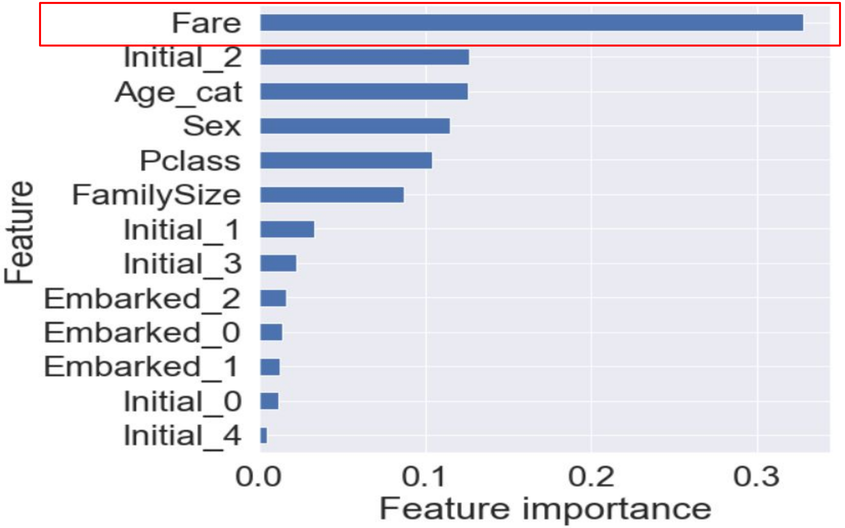
```python
model = RandomForestClassifier()
model.fit(X_tr, y_tr)
prediction = model.predict(X_vld)


print('총 {}명 중 {:.2f}% 정확도로 생존을 맞춤'.format
(y_vld.shape[0], 100 * metrics.accuracy_score(prediction, y_vld)))
```

```
총 268명 중 83.96% 정확도로 생존을 맞춤
```

4. Building machine learning model and prediction

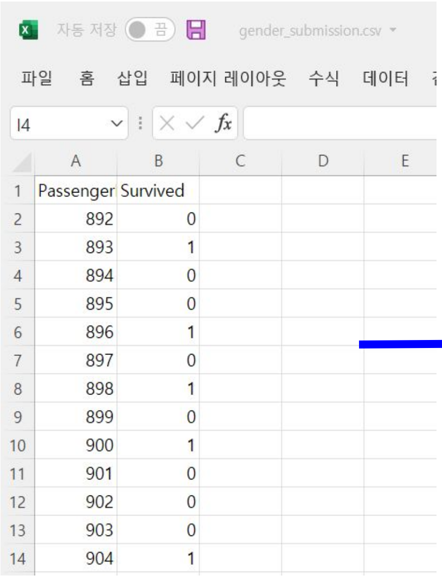
- Feature Importance

```python
from pandas import Series

feature_importance = model.feature_importances_
Series_feat_imp = Series(feature_importance, index = df_test.columns)

plt.figure(figsize=(8, 8))
Series_feat_imp.sort_values(ascending=True).plot.barh()
plt.xlabel('Feature importance')
plt.ylabel('Feature')
plt.show()
```

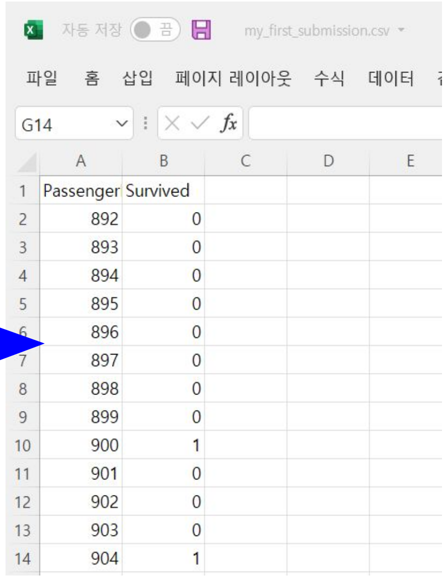


- Prediction on Test set (83% accuracy)

```python
submission = pd.read_csv('C:/DAIS/22SP/01_titanic/input/gender_submission.csv')

prediction = model.predict(X_test)
submission['Survived'] = prediction

submission.to_csv('./my_first_submission.csv', index=False)
```



	A	B	C	D	E
1	Passenger	Survived			
2	892	0			
3	893	1			
4	894	0			
5	895	0			
6	896	1			
7	897	0			
8	898	1			
9	899	0			
10	900	1			
11	901	0			
12	902	0			
13	903	0			
14	904	1			



	A	B	C	D	E
1	Passenger	Survived			
2	892	0			
3	893	0			
4	894	0			
5	895	0			
6	896	0			
7	897	0			
8	898	0			
9	899	0			
10	900	1			
11	901	0			
12	902	0			
13	903	0			
14	904	1			