후반부 Week3. 타이타닉에서는 누가 살아 남았을까?

Stage1

In [1]:

```
data = {
    'name': ['고양이', '펭귄', '닭', '타조', '참새'],
    'wing': [False, True, True, True, True],
    'weight': ['light', 'heavy', 'light', 'heavy', 'light'],
    'density': ['low', 'high', 'high', 'high', 'low'],
    'fly': [False, False, False, False, True]
}
```

In [2]:

```
print('고양이는 날개가 있을까?', data['wing'][0] )
print('닭은 날개가 있을까?', data['wing'][2] )

# 펭귄의 날개,무게,골밀도,비행가능여부 정보를 모두 보여주기
print(data['wing'][1],
      data['weight'][1],
      data['density'][1],
      data['fly'][1])

# 참새의 모든 정보 보여주기
bird_no = 4
print(data['wing'][bird_no],
      data['weight'][bird_no],
      data['density'][bird_no],
      data['fly'][bird_no])
```

```
고양이는 날개가 있을까? False
닭은 날개가 있을까? True
True heavy high False
True light low True
```

In [4]:

```
print('고양이는 날개가 있을까?', data['wing'][1] )
print('닭은 날개가 있을까?', data['wing'][0] )

# 펭귄의 날개,무게,골밀도,비행가능여부 정보를 모두 보여주기
print(data['wing'][1],
      data['weight'][1],
      data['density'][1],
      data['fly'][1])

# 참새의 모든 정보 보여주기
bird_no = 4
print(data['wing'][bird_no],
      data['weight'][bird_no],
      data['density'][bird_no],
      data['fly'][bird_no])
```

```
고양이는 날개가 있을까? True
닭은 날개가 있을까? False
True heavy high False
True light low True
```

In [5]:

```
target_index = 0

print(data['name'][target_index], ': 날 수 있는지 확인합니다.')

# 날개 유무
if data['wing'][target_index]: # 날개 있음
    # 몸무게
    if data['weight'][target_index] == 'heavy': # 몸무게 heavy
        print('날 수 없다')
    else: # 몸무게 light
        # 골밀도
        if data['density'][target_index] == 'high': # 골밀도 high
            print('날 수 없다')
        else: # 골밀도 low
            print('날 수 있다!!!')
else: # 날개 없음
    print('날 수 없다')
```

```
고양이 : 날 수 있는지 확인합니다.
날 수 없다
```

In [6]:

```
target_index = 1

print(data['name'][target_index], ': 날 수 있는지 확인합니다.')

# 날개 유무
if data['wing'][target_index]: # 날개 있음
    # 몸무게
    if data['weight'][target_index] == 'heavy': # 몸무게 heavy
        print('날 수 없다')
    else: # 몸무게 light
        # 골밀도
        if data['density'][target_index] == 'high': # 골밀도 high
            print('날 수 없다')
        else: # 골밀도 low
            print('날 수 있다!!!')
else: # 날개 없음
    print('날 수 없다')
```

펭귄 : 날 수 있는지 확인합니다.
날 수 없다

In [7]:

```
target_index = 2

print(data['name'][target_index], ': 날 수 있는지 확인합니다.')

# 날개 유무
if data['wing'][target_index]: # 날개 있음
    # 몸무게
    if data['weight'][target_index] == 'heavy': # 몸무게 heavy
        print('날 수 없다')
    else: # 몸무게 light
        # 골밀도
        if data['density'][target_index] == 'high': # 골밀도 high
            print('날 수 없다')
        else: # 골밀도 low
            print('날 수 있다!!!')
else: # 날개 없음
    print('날 수 없다')
```

닭 : 날 수 있는지 확인합니다.
날 수 없다

In [8]:

```
target_index = 3

print(data['name'][target_index], ': 날 수 있는지 확인합니다.')

# 날개 유무
if data['wing'][target_index]: # 날개 있음
    # 몸무게
    if data['weight'][target_index] == 'heavy': # 몸무게 heavy
        print('날 수 없다')
    else: # 몸무게 light
        # 골밀도
        if data['density'][target_index] == 'high': # 골밀도 high
            print('날 수 없다')
        else: # 골밀도 low
            print('날 수 있다!!!')
else: # 날개 없음
    print('날 수 없다')
```

타조 : 날 수 있는지 확인합니다.
날 수 없다

Stage2

In [10]:

```
import pandas as pd

df = pd.read_csv('data/train.csv')
df.head(10)
```

Out[10]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708

In [12]:

```
df[['Sex', 'Age', 'SibSp', 'Parch']].head()
```

Out[12]:

	Sex	Age	SibSp	Parch
0	male	22.0	1	0
1	female	38.0	1	0
2	female	26.0	0	0
3	female	35.0	1	0
4	male	35.0	0	0

In [13]:

```
df_test = pd.read_csv('data/test.csv')
df_test.head()
```

Out[13]:

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

In [14]:

```
df['Age'] = df['Age'].fillna(df['Age'].mean())
df.head(10)
```

Out[14]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8
5	6	0	3	Moran, Mr. James	male	29.699118	0	0	330877	8
6	7	0	1	McCarthy, Mr. Timothy J	male	54.000000	0	0	17463	51
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.000000	3	1	349909	21
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.000000	0	2	347742	11
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.000000	1	0	237736	30

In [15]:

```
df_test['Age'] = df_test['Age'].fillna(df_test['Age'].mean());
```

In [16]:

```
df.loc[df['Age'] < 10, 'Age'] = 0
df.loc[(df['Age'] >= 10) & (df['Age'] < 20), 'Age'] = 1
df.loc[(df['Age'] >= 20) & (df['Age'] < 30), 'Age'] = 2
df.loc[(df['Age'] >= 30) & (df['Age'] < 40), 'Age'] = 3
df.loc[(df['Age'] >= 40) & (df['Age'] < 50), 'Age'] = 4
df.loc[df['Age'] >= 50, 'Age'] = 5
```

In [17]:

```
df_test.loc[df['Age'] < 10, 'Age'] = 0
df_test.loc[(df['Age'] >= 10) & (df['Age'] < 20), 'Age'] = 1
df_test.loc[(df['Age'] >= 20) & (df['Age'] < 30), 'Age'] = 2
df_test.loc[(df['Age'] >= 30) & (df['Age'] < 40), 'Age'] = 3
df_test.loc[(df['Age'] >= 40) & (df['Age'] < 50), 'Age'] = 4
df_test.loc[df['Age'] >= 50, 'Age'] = 5
```

In [18]:

```
df['FamilySize'] = df['SibSp'] + df['Parch']
df.head()
```

Out[18]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	2.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	3.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	2.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	3.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	3.0	0	0	373450	8.0500

In [19]:

```
df_test['FamilySize'] = df_test['SibSp'] + df_test['Parch']
```

In [20]:

```
train = df[['Survived', 'Sex', 'Age', 'FamilySize']]
test = df_test[['Sex', 'Age', 'FamilySize']] # test데이터는 애초에 Survived가 없음

train.head()
```

Out[20]:

	Survived	Sex	Age	FamilySize
0	0	male	2.0	1
1	1	female	3.0	1
2	1	female	2.0	0
3	1	female	3.0	1
4	0	male	3.0	0

In [21]:

```
df['Fare'] = df['Fare'].fillna(df['Fare'].mean())
df_test['Fare'] = df_test['Fare'].fillna(df_test['Fare'].mean())
```

In [22]:

```
df.isnull().sum()
```

Out[22]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
FamilySize       0
dtype: int64
```

In [23]:

```
df['Embarked'].value_counts()
```

Out[23]:

```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [24]:

```
df['Embarked'] = df['Embarked'].fillna('S')
df_test['Embarked'] = df_test['Embarked'].fillna('S')
```

In [26]:

```
df.isnull().sum()
```

Out[26]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         0
FamilySize       0
dtype: int64
```

In [28]:

```
train = df[['Survived', 'Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']]
test = df_test[['Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']]
train.head()
```

Out[28]:

	Survived	Sex	Age	FamilySize	Fare	Embarked
0	0	male	2.0	1	7.2500	S
1	1	female	3.0	1	71.2833	C
2	1	female	2.0	0	7.9250	S
3	1	female	3.0	1	53.1000	S
4	0	male	3.0	0	8.0500	S

Stage3

In [49]:

```
x_train = train[['Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']]
y_train = train['Survived'] # 선택할 열이 하나면, []를 한번만 써주세요.

x_train
```

Out[49]:

	Sex	Age	FamilySize	Fare	Embarked
0	0	0.0	1	7.2500	0
1	1	0.0	1	71.2833	1
2	1	0.0	0	7.9250	0
3	1	0.0	1	53.1000	0
4	0	0.0	0	8.0500	0
...	...	...	...	...	...
886	0	0.0	0	13.0000	0
887	1	0.0	0	30.0000	0
888	1	0.0	3	23.4500	0
889	0	0.0	0	30.0000	1
890	0	0.0	0	7.7500	2

891 rows × 5 columns

In [50]:

```
from sklearn.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier()
tree.fit(x_train, y_train)

print('training set accuracy:', tree.score(x_train, y_train))
```

```
training set accuracy: 0.9191919191919192
```

In [31]:

```
df['Age'] = df['Age'].fillna(df['Age'].mean())
df.head(10)
```

Out[31]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	2.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	3.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	2.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	3.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	3.0	0	0	373450	8.0500
5	6	0	3	Moran, Mr. James	male	2.0	0	0	330877	8.4583
6	7	0	1	McCarthy, Mr. Timothy J	male	5.0	0	0	17463	51.8625
7	8	0	3	Palsson, Master. Gosta Leonard	male	0.0	3	1	349909	21.0750
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	2.0	0	2	347742	11.1333
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	1.0	1	0	237736	30.0708

In [52]:

```
x_test = test[['Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']]
x_test
```

Out[52]:

	Sex	Age	FamilySize	Fare	Embarked
0	0	0.0	0	7.8292	2
1	1	0.0	1	7.0000	0
2	0	0.0	0	9.6875	2
3	0	0.0	0	8.6625	0
4	1	0.0	2	12.2875	0
...	...	...	...	...	...
413	0	0.0	0	8.0500	0
414	1	0.0	0	108.9000	1
415	0	0.0	0	7.2500	0
416	0	0.0	0	8.0500	0
417	0	0.0	2	22.3583	1

418 rows × 5 columns

In [53]:

```
prediction = tree.predict(x_test)
prediction
```

Out[53]:

```
array([0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0,
       1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
       1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1,
       0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0,
       1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1,
       1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0,
       1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0],
      dtype=int64)
```

In [54]:

```
submit = pd.DataFrame({
    'PassengerId': df_test['PassengerId'],
    'Survived': prediction
})

submit.to_csv('submit.csv', index=False)
```

In [55]:

```
my_prediction = pd.read_csv('submit.csv')
my_prediction.head()
```

Out[55]:

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

In [36]:

```
df_test['FamilySize'] = df_test['SibSp'] + df_test['Parch']
```

In [38]:

```
train = df[['Survived', 'Sex', 'Age', 'FamilySize']]
test = df_test[['Sex', 'Age', 'FamilySize']]

train.head()
```

Out[38]:

	Survived	Sex	Age	FamilySize
0	0	male	0.0	1
1	1	female	0.0	1
2	1	female	0.0	0
3	1	female	0.0	1
4	0	male	0.0	0

Stage4

In [56]:

```
import pandas as pd
import numpy as np

df = pd.read_csv('data/train.csv')
df_test = pd.read_csv('data/test.csv')
```

In [57]:

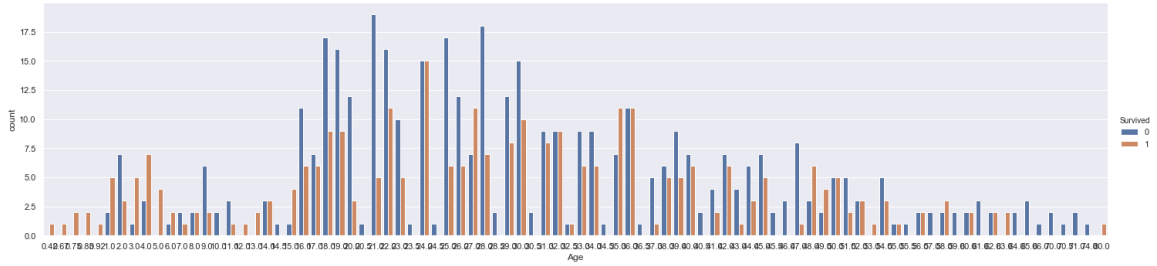
```
import matplotlib.pyplot as plt
%matplotlib inline

import seaborn as sns
sns.set()
```

In [58]:

```
sns.catplot(data=df, x='Age', hue='Survived', kind='count', aspect=4)
```

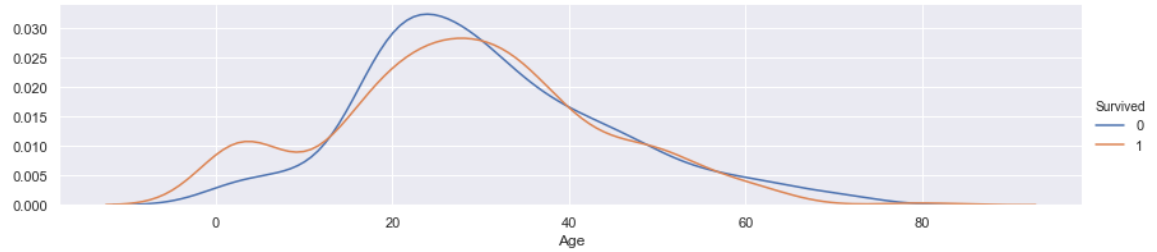
Out[58]:

<seaborn.axisgrid.FacetGrid at 0x22ed5cb2a48>



In [59]:

```
facet = sns.FacetGrid(df, hue="Survived", aspect=4)
facet.map(sns.kdeplot, 'Age')
facet.add_legend()

plt.show()
```



Challenge 1

In [60]:

```
df.loc[df['Sex'] == 'male', 'Sex'] = 0
df.loc[df['Sex'] == 'female', 'Sex'] = 1

df_test.loc[df_test['Sex'] == 'male', 'Sex'] = 0
df_test.loc[df_test['Sex'] == 'female', 'Sex'] = 1

df.head()
```

Out[60]:

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

In [61]:

```
df.loc[df['Embarked'] == 'S', 'Embarked'] = 0
df.loc[df['Embarked'] == 'C', 'Embarked'] = 1
df.loc[df['Embarked'] == 'Q', 'Embarked'] = 2

df_test.loc[df_test['Embarked'] == 'S', 'Embarked'] = 0
df_test.loc[df_test['Embarked'] == 'C', 'Embarked'] = 1
df_test.loc[df_test['Embarked'] == 'Q', 'Embarked'] = 2

df.head()
```

Out[61]:

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

In [65]:

```
train = df[['Survived', 'Sex', 'Age', 'Fare', 'Embarked']]
test = df_test[['Sex', 'Age', 'Fare', 'Embarked']]

train.head()
```

Out[65]:

	Survived	Sex	Age	Fare	Embarked
0	0	0	22.0	7.2500	0
1	1	1	38.0	71.2833	1
2	1	1	26.0	7.9250	0
3	1	1	35.0	53.1000	0
4	0	0	35.0	8.0500	0

Challenge 2

In [66]:

```
import pandas as pd
import numpy as np

df = pd.read_csv('data/train.csv')
df_test = pd.read_csv('data/test.csv')
```

In [67]:

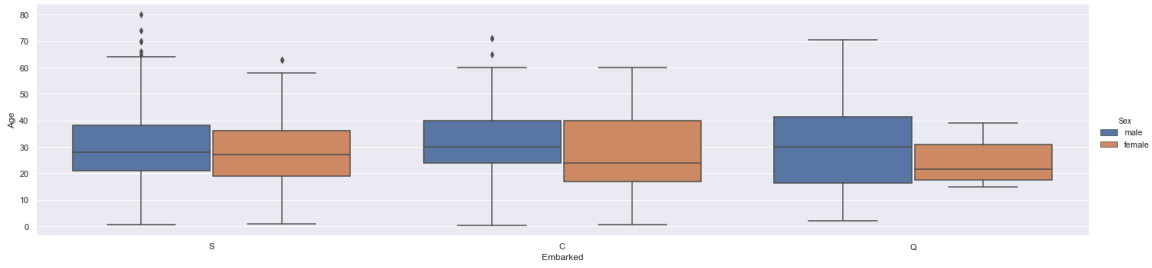
```
import matplotlib.pyplot as plt
%matplotlib inline

import seaborn as sns
sns.set()
```

In [68]:

```
sns.catplot(data=df, x='Embarked', y='Age', hue='Sex', kind='box', aspect=4)
```

Out[68]:

<seaborn.axisgrid.FacetGrid at 0x22ed6da1dc8>

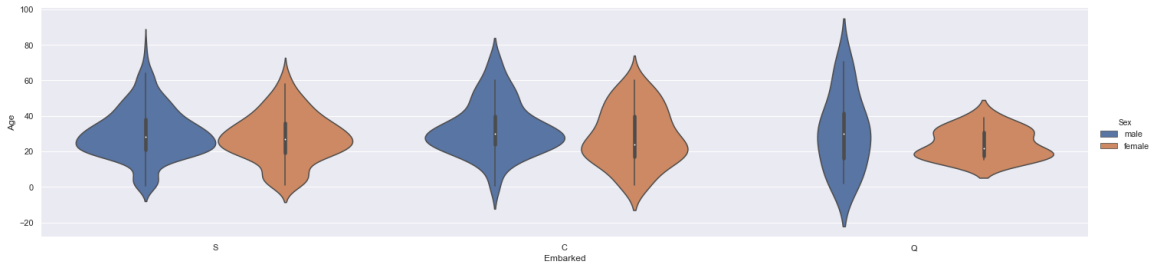


In [69]:

```
sns.catplot(data=df, x='Embarked', y='Age', hue='Sex', kind='violin', aspect=4)
```

Out[69]:

<seaborn.axisgrid.FacetGrid at 0x22ed6cf2ac8>



Homework1

In [71]:

```
df = pd.read_csv('data/train.csv')
df.head(10)
```

Out[71]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708

In [72]:

```
df[['Sex', 'Age', 'SibSp', 'Parch']].head()
```

Out[72]:

	Sex	Age	SibSp	Parch
0	male	22.0	1	0
1	female	38.0	1	0
2	female	26.0	0	0
3	female	35.0	1	0
4	male	35.0	0	0

In [73]:

```
df_test = pd.read_csv('data/test.csv')
df_test.head()
```

Out[73]:

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

In [74]:

```
df['Age'] = df['Age'].fillna(df['Age'].mean())
df.head(10)
```

Out[74]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8
5	6	0	3	Moran, Mr. James	male	29.699118	0	0	330877	8
6	7	0	1	McCarthy, Mr. Timothy J	male	54.000000	0	0	17463	51
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.000000	3	1	349909	21
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.000000	0	2	347742	11
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.000000	1	0	237736	30

In [75]:

```
df_test['Age'] = df_test['Age'].fillna(df_test['Age'].mean());
```

In [76]:

```
df.loc[df['Age'] < 10, 'Age'] = 0
df.loc[(df['Age'] >= 10) & (df['Age'] < 20), 'Age'] = 1
df.loc[(df['Age'] >= 20) & (df['Age'] < 30), 'Age'] = 2
df.loc[(df['Age'] >= 30) & (df['Age'] < 40), 'Age'] = 3
df.loc[(df['Age'] >= 40) & (df['Age'] < 50), 'Age'] = 4
df.loc[df['Age'] >= 50, 'Age'] = 5
```

In [77]:

```
df_test.loc[df['Age'] < 10, 'Age'] = 0
df_test.loc[(df['Age'] >= 10) & (df['Age'] < 20), 'Age'] = 1
df_test.loc[(df['Age'] >= 20) & (df['Age'] < 30), 'Age'] = 2
df_test.loc[(df['Age'] >= 30) & (df['Age'] < 40), 'Age'] = 3
df_test.loc[(df['Age'] >= 40) & (df['Age'] < 50), 'Age'] = 4
df_test.loc[df['Age'] >= 50, 'Age'] = 5
```

In [79]:

```
df['FamilySize'] = df['SibSp'] + df['Parch']
df.head()
```

Out[79]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	2.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	3.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	2.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	3.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	3.0	0	0	373450	8.0500

In [80]:

```
df_test['FamilySize'] = df_test['SibSp'] + df_test['Parch']
```

In [81]:

```
train = df[['Survived', 'Sex', 'Age', 'FamilySize']]
test = df_test[['Sex', 'Age', 'FamilySize']]

train.head()
```

Out[81]:

	Survived	Sex	Age	FamilySize
0	0	male	2.0	1
1	1	female	3.0	1
2	1	female	2.0	0
3	1	female	3.0	1
4	0	male	3.0	0

In [82]:

```
df['Fare'] = df['Fare'].fillna(df['Fare'].mean())
df_test['Fare'] = df_test['Fare'].fillna(df_test['Fare'].mean())
```

In [83]:

```
df.isnull().sum()
```

Out[83]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
FamilySize       0
dtype: int64
```

In [84]:

```
df['Embarked'].value_counts()
```

Out[84]:

```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [85]:

```
df['Embarked'] = df['Embarked'].fillna('S')
df_test['Embarked'] = df_test['Embarked'].fillna('S')
```

In [86]:

```
df.isnull().sum()
```

Out[86]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         0
FamilySize       0
dtype: int64
```

In [87]:

```
train = df[['Survived', 'Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']]
test = df_test[['Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']]
train.head()
```

Out[87]:

	Survived	Sex	Age	FamilySize	Fare	Embarked
0	0	male	2.0	1	7.2500	S
1	1	female	3.0	1	71.2833	C
2	1	female	2.0	0	7.9250	S
3	1	female	3.0	1	53.1000	S
4	0	male	3.0	0	8.0500	S

In [88]:

```
df.loc[df['Sex'] == 'male', 'Sex'] = 0
df.loc[df['Sex'] == 'female', 'Sex'] = 1

df_test.loc[df_test['Sex'] == 'male', 'Sex'] = 0
df_test.loc[df_test['Sex'] == 'female', 'Sex'] = 1

df.head()
```

Out[88]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
0	1	0	3	Braund, Mr. Owen Harris	0	2.0	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	3.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	1	2.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	3.0	1	0	113803	53.1000	
4	5	0	3	Allen, Mr. William Henry	0	3.0	0	0	373450	8.0500	

In [89]:

```
df.loc[df['Embarked'] == 'S', 'Embarked'] = 0
df.loc[df['Embarked'] == 'C', 'Embarked'] = 1
df.loc[df['Embarked'] == 'Q', 'Embarked'] = 2

df_test.loc[df_test['Embarked'] == 'S', 'Embarked'] = 0
df_test.loc[df_test['Embarked'] == 'C', 'Embarked'] = 1
df_test.loc[df_test['Embarked'] == 'Q', 'Embarked'] = 2

df.head()
```

Out[89]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
0	1	0	3	Braund, Mr. Owen Harris	0	2.0	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	3.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	1	2.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	3.0	1	0	113803	53.1000	
4	5	0	3	Allen, Mr. William Henry	0	3.0	0	0	373450	8.0500	

In [90]:

```
train = df[['Survived', 'Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']]
test = df_test[['Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']] # test데이터는 애초에 Survived가 없음

train.head()
```

Out[90]:

	Survived	Sex	Age	FamilySize	Fare	Embarked
0	0	0	2.0	1	7.2500	0
1	1	1	3.0	1	71.2833	1
2	1	1	2.0	0	7.9250	0
3	1	1	3.0	1	53.1000	0
4	0	0	3.0	0	8.0500	0

In [91]:

```
df[['Name', 'Age']].head(10)
```

Out[91]:

	Name	Age
0	Braund, Mr. Owen Harris	2.0
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	3.0
2	Heikkinen, Miss. Laina	2.0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	3.0
4	Allen, Mr. William Henry	3.0
5	Moran, Mr. James	2.0
6	McCarthy, Mr. Timothy J	5.0
7	Palsson, Master. Gosta Leonard	0.0
8	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	2.0
9	Nasser, Mrs. Nicholas (Adele Achem)	1.0

In [92]:

```
df.loc[ df['Name'].str.contains('MrW.'), 'Name' ] = 'Mr'
df.loc[ df['Name'].str.contains('MrsW.'), 'Name' ] = 'Mrs'
df.loc[ df['Name'].str.contains('MissW.'), 'Name' ] = 'Miss'
df.head()
```

Out[92]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	Mr	0	2.0	1	0	A/5 21171	7.2500	Nal
1	2	1	1	Mrs	1	3.0	1	0	PC 17599	71.2833	C8
2	3	1	3	Miss	1	2.0	0	0	STON/O2. 3101282	7.9250	Nal
3	4	1	1	Mrs	1	3.0	1	0	113803	53.1000	C12
4	5	0	3	Mr	0	3.0	0	0	373450	8.0500	Nal

In [93]:

```
df['Name'] = df['Name'].map({
    'Mr': 0,
    'Mrs': 1,
    'Miss': 2
})

df.head(10)
```

Out[93]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	0.0	0	2.0	1	0	A/5 21171	7.2500	Nal
1	2	1	1	1.0	1	3.0	1	0	PC 17599	71.2833	C8
2	3	1	3	2.0	1	2.0	0	0	STON/O2. 3101282	7.9250	Nal
3	4	1	1	1.0	1	3.0	1	0	113803	53.1000	C12
4	5	0	3	0.0	0	3.0	0	0	373450	8.0500	Nal
5	6	0	3	0.0	0	2.0	0	0	330877	8.4583	Nal
6	7	0	1	0.0	0	5.0	0	0	17463	51.8625	E4
7	8	0	3	NaN	0	0.0	3	1	349909	21.0750	Nal
8	9	1	3	1.0	1	2.0	0	2	347742	11.1333	Nal
9	10	1	2	1.0	1	1.0	1	0	237736	30.0708	Nal

In [94]:

```
df['Name'] = df['Name'].fillna(3)
df.head(10)
```

Out[94]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	0.0	0	2.0	1	0	A/5 21171	7.2500	Nal
1	2	1	1	1.0	1	3.0	1	0	PC 17599	71.2833	C8
2	3	1	3	2.0	1	2.0	0	0	STON/O2. 3101282	7.9250	Nal
3	4	1	1	1.0	1	3.0	1	0	113803	53.1000	C12
4	5	0	3	0.0	0	3.0	0	0	373450	8.0500	Nal
5	6	0	3	0.0	0	2.0	0	0	330877	8.4583	Nal
6	7	0	1	0.0	0	5.0	0	0	17463	51.8625	E4
7	8	0	3	3.0	0	0.0	3	1	349909	21.0750	Nal
8	9	1	3	1.0	1	2.0	0	2	347742	11.1333	Nal
9	10	1	2	1.0	1	1.0	1	0	237736	30.0708	Nal

In [95]:

```
df['Name'].value_counts()
```

Out[95]:

```
0.0    517
2.0    182
1.0    125
3.0     67
Name: Name, dtype: int64
```

In [96]:

```
df['Age'] = df['Age'].fillna( df.groupby('Name')['Age'].transform('mean') )
df.head(10)
```

Out[96]:

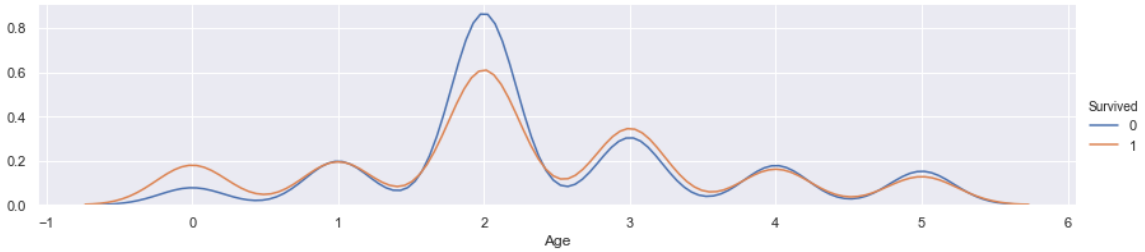
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	0.0	0	2.0	1	0	A/5 21171	7.2500	Nal
1	2	1	1	1.0	1	3.0	1	0	PC 17599	71.2833	C8
2	3	1	3	2.0	1	2.0	0	0	STON/O2. 3101282	7.9250	Nal
3	4	1	1	1.0	1	3.0	1	0	113803	53.1000	C12
4	5	0	3	0.0	0	3.0	0	0	373450	8.0500	Nal
5	6	0	3	0.0	0	2.0	0	0	330877	8.4583	Nal
6	7	0	1	0.0	0	5.0	0	0	17463	51.8625	E4
7	8	0	3	3.0	0	0.0	3	1	349909	21.0750	Nal
8	9	1	3	1.0	1	2.0	0	2	347742	11.1333	Nal
9	10	1	2	1.0	1	1.0	1	0	237736	30.0708	Nal

In [97]:

```
import matplotlib.pyplot as plt
%matplotlib inline

import seaborn as sns
sns.set()
```

In [98]:

```
facet = sns.FacetGrid(df, hue="Survived", aspect=4)
facet.map(sns.kdeplot, 'Age')
facet.add_legend()

plt.show()
```



Homework2

In [99]:

```
df['Age'] = df['Age'].fillna(df['Age'].mean())
df.head(10)
```

Out[99]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	0.0	0	2.0	1	0	A/5 21171	7.2500	Nal
1	2	1	1	1.0	1	3.0	1	0	PC 17599	71.2833	C8
2	3	1	3	2.0	1	2.0	0	0	STON/O2. 3101282	7.9250	Nal
3	4	1	1	1.0	1	3.0	1	0	113803	53.1000	C12
4	5	0	3	0.0	0	3.0	0	0	373450	8.0500	Nal
5	6	0	3	0.0	0	2.0	0	0	330877	8.4583	Nal
6	7	0	1	0.0	0	5.0	0	0	17463	51.8625	E4
7	8	0	3	3.0	0	0.0	3	1	349909	21.0750	Nal
8	9	1	3	1.0	1	2.0	0	2	347742	11.1333	Nal
9	10	1	2	1.0	1	1.0	1	0	237736	30.0708	Nal

In [100]:

```
df_test['Age'] = df_test['Age'].fillna(df_test['Age'].mean());
```

In [101]:

```
df.loc[df['Age'] < 10, 'Age'] = 0
df.loc[(df['Age'] >= 10) & (df['Age'] < 20), 'Age'] = 1
df.loc[(df['Age'] >= 20) & (df['Age'] < 30), 'Age'] = 2
df.loc[(df['Age'] >= 30) & (df['Age'] < 40), 'Age'] = 3
df.loc[(df['Age'] >= 40) & (df['Age'] < 50), 'Age'] = 4
df.loc[df['Age'] >= 50, 'Age'] = 5
```

In [102]:

```
df_test.loc[df['Age'] < 10, 'Age'] = 0
df_test.loc[(df['Age'] >= 10) & (df['Age'] < 20), 'Age'] = 1
df_test.loc[(df['Age'] >= 20) & (df['Age'] < 30), 'Age'] = 2
df_test.loc[(df['Age'] >= 30) & (df['Age'] < 40), 'Age'] = 3
df_test.loc[(df['Age'] >= 40) & (df['Age'] < 50), 'Age'] = 4
df_test.loc[df['Age'] >= 50, 'Age'] = 5
```

In [103]:

```
df['FamilySize'] = df['SibSp'] + df['Parch']
df.head()
```

Out[103]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	0.0	0	0.0	1	0	A/5 21171	7.2500	Nal
1	2	1	1	1.0	1	0.0	1	0	PC 17599	71.2833	C8
2	3	1	3	2.0	1	0.0	0	0	STON/O2. 3101282	7.9250	Nal
3	4	1	1	1.0	1	0.0	1	0	113803	53.1000	C12
4	5	0	3	0.0	0	0.0	0	0	373450	8.0500	Nal

In [104]:

```
df_test['FamilySize'] = df_test['SibSp'] + df_test['Parch']
```

In [105]:

```
train = df[['Survived', 'Sex', 'Age', 'FamilySize']]
test = df_test[['Sex', 'Age', 'FamilySize']]

train.head()
```

Out[105]:

	Survived	Sex	Age	FamilySize
0	0	0	0.0	1
1	1	1	0.0	1
2	1	1	0.0	0
3	1	1	0.0	1
4	0	0	0.0	0

In [106]:

```
df['Fare'] = df['Fare'].fillna(df['Fare'].mean())
df_test['Fare'] = df_test['Fare'].fillna(df_test['Fare'].mean())
```

In [107]:

```
df.isnull().sum()
```

Out[107]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         0
FamilySize       0
dtype: int64
```

In [108]:

```
df['Embarked'].value_counts()
```

Out[108]:

```
0    646
1    168
2     77
Name: Embarked, dtype: int64
```

In [109]:

```
df['Embarked'] = df['Embarked'].fillna('S')
df_test['Embarked'] = df_test['Embarked'].fillna('S')
```

In [110]:

```
df.isnull().sum()
```

Out[110]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         0
FamilySize       0
dtype: int64
```

In [111]:

```
train = df[['Survived', 'Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']]
test = df_test[['Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']]

train.head()
```

Out[111]:

	Survived	Sex	Age	FamilySize	Fare	Embarked
0	0	0	0.0	1	7.2500	0
1	1	1	0.0	1	71.2833	1
2	1	1	0.0	0	7.9250	0
3	1	1	0.0	1	53.1000	0
4	0	0	0.0	0	8.0500	0

In [112]:

```
df.loc[df['Sex'] == 'male', 'Sex'] = 0
df.loc[df['Sex'] == 'female', 'Sex'] = 1

df_test.loc[df_test['Sex'] == 'male', 'Sex'] = 0
df_test.loc[df_test['Sex'] == 'female', 'Sex'] = 1

df.head()
```

Out[112]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	0.0	0	0.0	1	0	A/5 21171	7.2500	Nal
1	2	1	1	1.0	1	0.0	1	0	PC 17599	71.2833	C8
2	3	1	3	2.0	1	0.0	0	0	STON/O2. 3101282	7.9250	Nal
3	4	1	1	1.0	1	0.0	1	0	113803	53.1000	C12
4	5	0	3	0.0	0	0.0	0	0	373450	8.0500	Nal

In [113]:

```
df.loc[df['Embarked'] == 'S', 'Embarked'] = 0
df.loc[df['Embarked'] == 'C', 'Embarked'] = 1
df.loc[df['Embarked'] == 'Q', 'Embarked'] = 2

df_test.loc[df_test['Embarked'] == 'S', 'Embarked'] = 0
df_test.loc[df_test['Embarked'] == 'C', 'Embarked'] = 1
df_test.loc[df_test['Embarked'] == 'Q', 'Embarked'] = 2

df.head()
```

```
C:₩Users₩user₩anaconda3₩lib₩site-packages₩pandas₩core₩ops₩array_ops.py:253: Future
Warning: elementwise comparison failed; returning scalar instead, but in the futur
e will perform elementwise comparison
  res_values = method(rvalues)
```

Out[113]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	0.0	0	0.0	1	0	A/5 21171	7.2500	Nal
1	2	1	1	1.0	1	0.0	1	0	PC 17599	71.2833	C8
2	3	1	3	2.0	1	0.0	0	0	STON/O2. 3101282	7.9250	Nal
3	4	1	1	1.0	1	0.0	1	0	113803	53.1000	C12
4	5	0	3	0.0	0	0.0	0	0	373450	8.0500	Nal

In [114]:

```
train = df[['Survived', 'Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']]
test = df_test[['Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']] # test데이터는 애초에 Survived가 없음

train.head()
```

Out[114]:

	Survived	Sex	Age	FamilySize	Fare	Embarked
0	0	0	0.0	1	7.2500	0
1	1	1	0.0	1	71.2833	1
2	1	1	0.0	0	7.9250	0
3	1	1	0.0	1	53.1000	0
4	0	0	0.0	0	8.0500	0

In [115]:

```
df['Fare'] = df['Fare'].fillna( df.groupby('Pclass')['Fare'].transform('mean') )
```

In [116]:

```
df_test['Fare'] = df_test['Fare'].fillna( df_test.groupby('Pclass')['Fare'].transform('mean') )
```

In [117]:

```
print(df_test.isnull().sum())
```

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          327
Embarked         0
FamilySize       0
dtype: int64
```

In [118]:

```
x_train = train[['Sex', 'Age', 'FamilySize', 'Fare', 'Embarked']]
y_train = train['Survived']

x_train
```

Out[118]:

	Sex	Age	FamilySize	Fare	Embarked
0	0	0.0	1	7.2500	0
1	1	0.0	1	71.2833	1
2	1	0.0	0	7.9250	0
3	1	0.0	1	53.1000	0
4	0	0.0	0	8.0500	0
...	...	...	...	...	...
886	0	0.0	0	13.0000	0
887	1	0.0	0	30.0000	0
888	1	0.0	3	23.4500	0
889	0	0.0	0	30.0000	1
890	0	0.0	0	7.7500	2

891 rows × 5 columns

In [119]:

```
from sklearn.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier()
tree.fit(x_train, y_train)

print('training set accuracy:', tree.score(x_train, y_train))
```

```
training set accuracy: 0.9191919191919192
```

In [120]:

```
prediction = tree.predict(x_test)
prediction
```

Out[120]:

```
array([0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0,
       1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
       1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1,
       0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0,
       1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0,
       1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0],
      dtype=int64)
```

In [121]:

```
submit = pd.DataFrame({
    'PassengerId': df_test['PassengerId'],
    'Survived': prediction
})

submit.to_csv('submit.csv', index=False)
```

In [122]:

```
my_prediction = pd.read_csv('submit.csv')
my_prediction.head()
```

Out[122]:

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

In [ ]:

```

```