In [1]:
```
import pandas as pd

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

Out[1]:

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

In [2]:
```
df_test = pd.read_csv('data/test.csv')
df_test.head(5)
```

Out[2]:

	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

In [3]:
```
print("train data:", df.shape)
print("test data:", df_test.shape)
```

```
train data: (891, 12)
test data: (418, 11)
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [5]:
```
df_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Name         418 non-null    object 
 3   Sex          418 non-null    object 
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64  
 6   Parch        418 non-null    int64  
 7   Ticket       418 non-null    object 
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object 
 10  Embarked     418 non-null    object 
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

In [6]:
```
df_test['Fare']=df_test['Fare'].fillna(df_test['Fare'].mean())
```

In [7]:
```
#  Age 의 빈값(Nan)을 평균(mean) 값으로 채워주기

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

In [8]:
```
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 [9]:
```
df_test.loc[df_test['Age'] < 10, 'Age'] = 0
df_test.loc[(df_test['Age'] >= 10) & (df_test['Age'] < 20), 'Age'] = 1
df_test.loc[(df_test['Age'] >= 20) & (df_test['Age'] < 30), 'Age'] = 2
df_test.loc[(df_test['Age'] >= 30) & (df_test['Age'] < 40), 'Age'] = 3
df_test.loc[(df_test['Age'] >= 40) & (df_test['Age'] < 50), 'Age'] = 4
df_test.loc[df_test['Age'] >= 50, 'Age'] = 5
```

In [10]:
```
# FamilySIze

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

Out[10]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Eml
0	1	0	3	Braund, Mr. Owen Harris	male	2.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	3.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	2.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	3.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	3.0	0	0	373450	8.0500	NaN	

In [11]:
```
# 필요한 변수만 추출


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

Out[11]:

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

In [12]:
```
test=df_test[['Pclass', 'Sex', 'Age', 'Fare', 'FamilySize']]
test.head()
```

Out[12]:

	Pclass	Sex	Age	Fare	FamilySize
0	3	male	3.0	7.8292	0
1	3	female	4.0	7.0000	1
2	2	male	5.0	9.6875	0
3	3	male	2.0	8.6625	0
4	3	female	2.0	12.2875	2

In [13]:
```
train.loc[train['Sex'] =='male', 'Sex'] = 0
train.loc[train['Sex'] =='female', 'Sex'] = 1

train.head()
```

```
C:WProgramDataWAnaconda3WlibWsite-packagesWpandasWcoreWindexing.py:1765: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  isetter(loc, value)
```

Out[13]:

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

In [14]:
```
test.loc[test['Sex'] =='male', 'Sex'] = 0
test.loc[test['Sex'] =='female', 'Sex'] = 1

test.head()
```

```
C:WProgramDataWAnaconda3WlibWsite-packagesWpandasWcoreWindexing.py:1765: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  isetter(loc, value)
```

Out[14]:

	Pclass	Sex	Age	Fare	FamilySize
0	3	0	3.0	7.8292	0
1	3	1	4.0	7.0000	1
2	2	0	5.0	9.6875	0
3	3	0	2.0	8.6625	0

	Pclass	Sex	Age	Fare	FamilySize
4	3	1	2.0	12.2875	2

In [15]:
```
train.isnull().sum()
```

Out[15]:
```
Survived      0
Pclass        0
Sex           0
Age           0
Fare          0
FamilySize    0
dtype: int64
```

In [16]:
```
test.isnull().sum()
```

Out[16]:
```
Pclass        0
Sex           0
Age           0
Fare          0
FamilySize    0
dtype: int64
```

In [17]:
```
from sklearn.model_selection import train_test_split

#1. train-target 구분

x_train = train.drop(['Survived'], axis=1)
y_train = train['Survived']

x_train.shape
```

Out[17]: (891, 5)

In [18]:
```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(x_train, y_train)

print("train 데이터 예측결과:", lr.score(x_train, y_train))
```

train 데이터 예측결과: 0.3911535087085106

In [19]:
```
from sklearn.ensemble import RandomForestRegressor

forest = RandomForestRegressor()
forest.fit(x_train, y_train)

print("학습된 데이터 예측결과:", forest.score(x_train, y_train))
```

학습된 데이터 예측결과: 0.7986898613568904

In [20]:
```
test.shape
```

Out[20]: (418, 5)

In [21]:
```
test.head()
```

Out[21]:

	Pclass	Sex	Age	Fare	FamilySize

	Pclass	Sex	Age	Fare	FamilySize
0	3	0	3.0	7.8292	0
1	3	1	4.0	7.0000	1
2	2	0	5.0	9.6875	0
3	3	0	2.0	8.6625	0
4	3	1	2.0	12.2875	2

In [28]:
```
x_test=test[['Pclass', 'Sex', 'Age', 'Fare', 'FamilySize']]

prediction = lr.predict(x_test)
prediction[prediction > 0.5] = 1
prediction[prediction <= 0.5] = 0
prediction = prediction.astype(int)

prediction
```

Out[28]:
```
array([0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 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, 0, 0, 0, 0, 0, 0, 1, 1,
       1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0,
       1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 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, 0, 0, 1, 0, 0,
       1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0])
```

In [29]:
```
submit = pd.DataFrame({
    'PassengerId': df_test['PassengerId'],
    'Survived': prediction
})

submit.to_csv('submit_wk5hw2', index=False)
```

In [30]:
```
my_prediction = pd.read_csv('submit_wk5hw2')
my_prediction.head()
```

Out[30]:

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

kaggle score

0.77751

In [ ]:

In [ ]: