titanic-classification

December 30, 2023

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
```

[3]:
```
print(train.shape)
print(test.shape)
```

```
(891, 12)
(418, 11)
```

[4]:
```
train.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
```

[5]:
```
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: 36.1+ KB
```

[6]:
```
train.drop(columns=['Cabin'],inplace=True)
test.drop(columns=['Cabin'],inplace=True)
```

[7]:
```
train.isnull().sum()
```

[7]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Embarked         2
dtype: int64
```

[8]:
```
test.isnull().sum()
```

[8]:
```
PassengerId     0
Pclass          0
Name            0
Sex             0
Age            86
SibSp           0
Parch           0
Ticket          0
```

```
Fare            1
Embarked        0
dtype: int64
```

[9]:
```
train['Embarked'].value_counts()
```

[9]:
```
Embarked
S    644
C    168
Q     77
Name: count, dtype: int64
```

[10]:
```
train['Embarked'].fillna('S',inplace=True)
```

[11]:
```
train.isnull().sum()
```

[11]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Embarked         0
dtype: int64
```

[12]:
```
test['Fare'].fillna(test['Fare'].mean(),inplace=True) ##filling null values in␣
↪fare
```

[13]:
```
test.isnull().sum()
```

[13]:
```
PassengerId     0
Pclass          0
Name            0
Sex             0
Age            86
SibSp           0
Parch           0
Ticket          0
Fare            0
Embarked        0
dtype: int64
```

[14]:
```
train_age=np.random.randint(train['Age'].mean()-train['Age'].std(),train['Age'].
↪mean()+train['Age'].std(),177)
```

[15]:
```
test_age=np.random.randint(test['Age'].mean()-test['Age'].std(),test['Age'].
↪mean()+test['Age'].std(),86)
```

[16]:
```
train['Age'][train['Age'].isnull()]=train_age ##filling of null values in age
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_11912\1168171242.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Age'][train['Age'].isnull()]=train_age ##filling of null values in age
```

[17]:
```
train.isnull().sum()
```

[17]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

[18]:
```
test['Age'][test['Age'].isnull()]=test_age
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_11912\3484201817.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Age'][test['Age'].isnull()]=test_age
```

[19]:
```
test.isnull().sum()
```

[19]:
```
PassengerId    0
Pclass         0
Name           0
Sex            0
```

```
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

[20]:
```
pd.crosstab(train['Pclass'], train['Survived']).apply(lambda r: round((r/r.
↪sum())*100,1), axis=1)
```

[20]:
```
Survived     0     1
Pclass
1         37.0  63.0
2         52.7  47.3
3         75.8  24.2
```

[21]:
```
train.groupby(['Sex'])['Survived'].mean()
```

[21]:
```
Sex
female    0.742038
male      0.188908
Name: Survived, dtype: float64
```

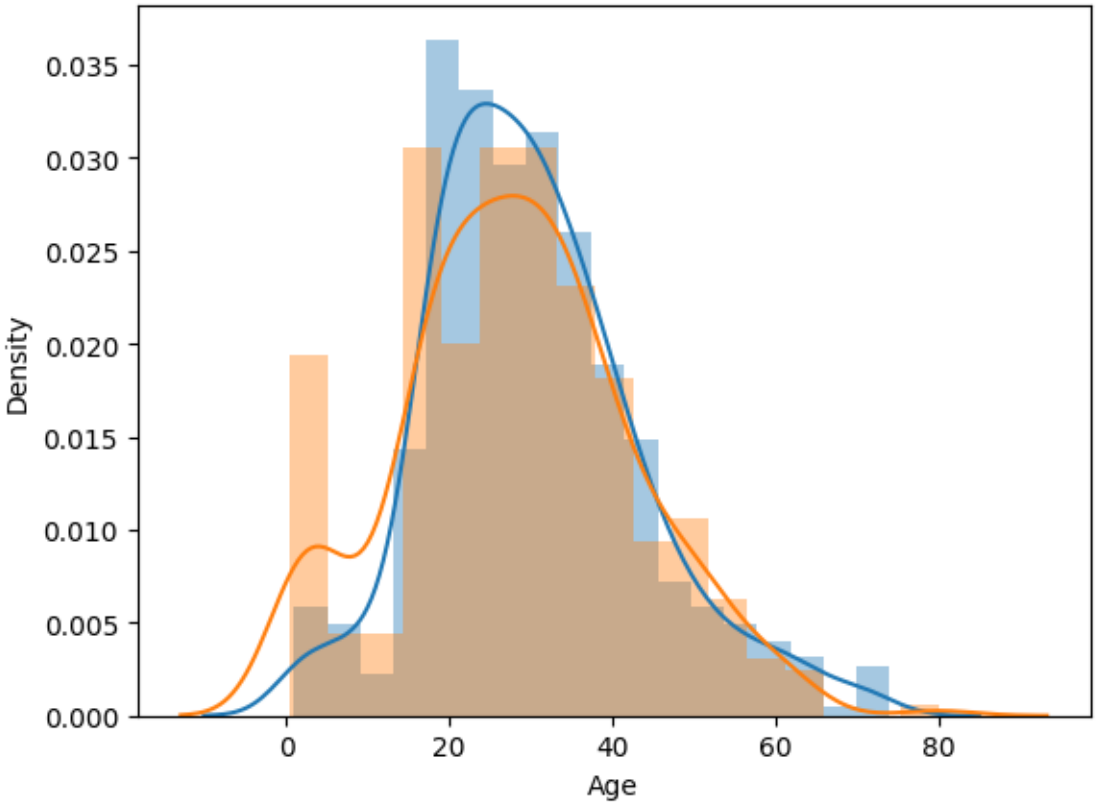
[22]:
```
#the above calculations show that females were saved in large number than mens
```

[23]:
```
sns.distplot(train['Age'][train['Survived']==0])#dead
sns.distplot(train['Age'][train['Survived']==1])#survived orange
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_11912\1589138790.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(train['Age'][train['Survived']==0])#dead
C:\Users\Dell\AppData\Local\Temp\ipykernel_11912\1589138790.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```

```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(train['Age'][train['Survived']==1])#survived orange
```

[23]:
```
<Axes: xlabel='Age', ylabel='Density'>
```



[24]:
```
# above graph shows that age group of 0-15 childrens has less death ratio
#age group of 20-35 died more
```

[25]:
```
train.drop(columns=['Ticket'],inplace=True)
test.drop(columns=['Ticket'],inplace=True)
```

[26]:
```
train['family']=train['SibSp']+train['Parch']+1
train['family'].value_counts()
```

[26]:
```
family
1     537
2     161
3     102
4      29
```

```
6      22
5      15
7      12
11      7
8       6
Name: count, dtype: int64
```

[27]:
```
#above calculations show that people who travelled with 4 members group had survived more
#and people with 8 and 11 group members all were died
```

[28]:
```
test['family']=test['SibSp']+test['Parch']+1
```

[29]:
```
def cal(number):
    if number==1:
        return "Alone"
    elif number>1 and number<5:
        return "Medium"
    else:
        return "Large"
```

[30]:
```
train['family_size']=train['family'].apply(cal)
```

[31]:
```
train.head()
```

[31]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch     Fare Embarked  family family_size
0      0   7.2500        S       2      Medium
1      0  71.2833        C       2      Medium
2      0   7.9250        S       1       Alone
3      0  53.1000        S       2      Medium
4      0   8.0500        S       1       Alone
```

[32]:
```
test['family_size']=test['family'].apply(cal)
```

[33]:
```
test.head()
```

[33]:
```
   PassengerId  Pclass                                          Name     Sex
0          892       3                              Kelly, Mr. James    male  \
1          893       3              Wilkes, Mrs. James (Ellen Needs)  female
2          894       2                     Myles, Mr. Thomas Francis    male
3          895       3                              Wirz, Mr. Albert    male
4          896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female

    Age  SibSp  Parch     Fare Embarked  family family_size
0  34.5      0      0   7.8292        Q       1       Alone
1  47.0      1      0   7.0000        S       2      Medium
2  62.0      0      0   9.6875        Q       1       Alone
3  27.0      0      0   8.6625        S       1       Alone
4  22.0      1      1  12.2875        S       3      Medium
```

[34]:
```
pd.crosstab(train['family_size'], train['Survived']).apply(lambda r: round((r/r.
↪sum())*100,1), axis=1)
```

[34]:
```
Survived        0     1
family_size
Alone        69.6  30.4
Large        83.9  16.1
Medium       42.1  57.9
```

[35]:
```
passengerId=train['PassengerId'].values
passengerId=test['PassengerId'].values
```

[36]:
```
train.drop(columns=['Name','PassengerId'],inplace=True)
test.drop(columns=['Name','PassengerId'],inplace=True)
```

[37]:
```
train
```

[37]:
```
     Survived  Pclass     Sex   Age  SibSp  Parch     Fare Embarked  family  \
0           0       3    male  22.0      1      0   7.2500        S       2
1           1       1  female  38.0      1      0  71.2833        C       2
2           1       3  female  26.0      0      0   7.9250        S       1
3           1       1  female  35.0      1      0  53.1000        S       2
4           0       3    male  35.0      0      0   8.0500        S       1
..        ...     ...     ...   ...    ...    ...      ...      ...     ...
886         0       2    male  27.0      0      0  13.0000        S       1
887         1       1  female  19.0      0      0  30.0000        S       1
888         0       3  female  30.0      1      2  23.4500        S       4
889         1       1    male  26.0      0      0  30.0000        C       1
890         0       3    male  32.0      0      0   7.7500        Q       1

    family_size
```

```
0        Medium
1        Medium
2         Alone
3        Medium
4         Alone
..          …
886       Alone
887       Alone
888      Medium
889       Alone
890       Alone

[891 rows x 10 columns]
```

[38]:
```
train=pd.
 ↪get_dummies(train,columns=['Pclass','Sex','Embarked','family_size'],drop_first=True)
```

[39]:
```
train #true->0 male
```

[39]:
```
     Survived   Age  SibSp  Parch     Fare  family  Pclass_2  Pclass_3  \
0           0  22.0      1      0   7.2500       2     False      True
1           1  38.0      1      0  71.2833       2     False     False
2           1  26.0      0      0   7.9250       1     False      True
3           1  35.0      1      0  53.1000       2     False     False
4           0  35.0      0      0   8.0500       1     False      True
..        …   …    …    …      …      …       …        …
886         0  27.0      0      0  13.0000       1      True     False
887         1  19.0      0      0  30.0000       1     False     False
888         0  30.0      1      2  23.4500       4     False      True
889         1  26.0      0      0  30.0000       1     False     False
890         0  32.0      0      0   7.7500       1     False      True

     Sex_male  Embarked_Q  Embarked_S  family_size_Large  family_size_Medium
0        True       False        True              False                True
1       False       False       False              False                True
2       False       False        True              False               False
3       False       False        True              False                True
4        True       False        True              False               False
..        …          …         …                 …                  …
886      True       False        True              False               False
887     False       False        True              False               False
888     False       False        True              False                True
889      True       False       False              False               False
890      True        True       False              False               False

[891 rows x 13 columns]
```

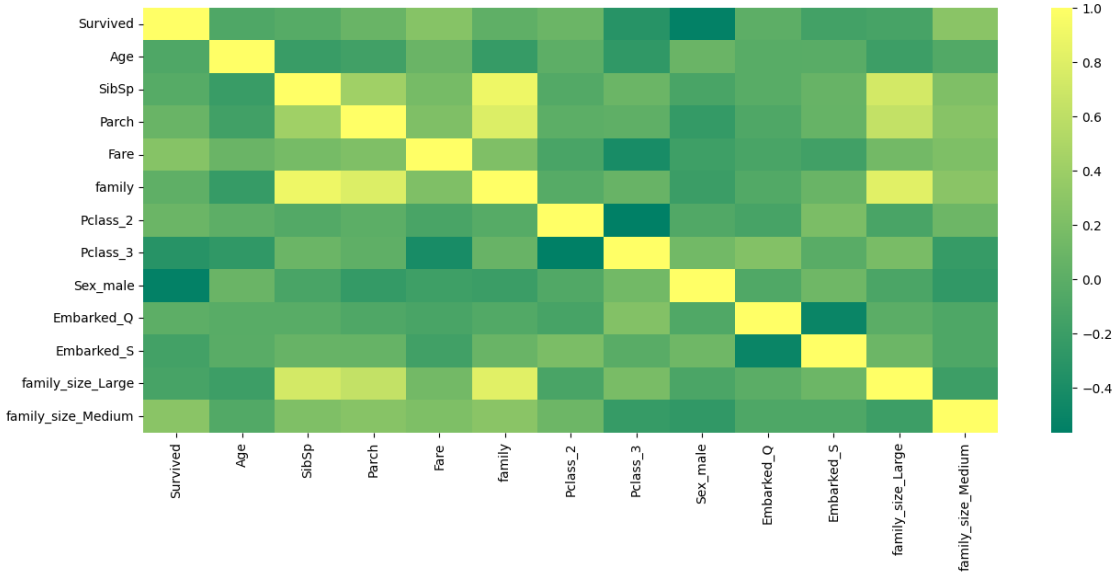
[40]:
```
test=pd.
 ↪get_dummies(test,columns=['Pclass','Sex','Embarked','family_size'],drop_first=True)
```

[41]:
```
test
```

[41]:
```
      Age  SibSp  Parch      Fare  family  Pclass_2  Pclass_3  Sex_male  \
0    34.5      0      0    7.8292       1     False      True      True
1    47.0      1      0    7.0000       2     False      True     False
2    62.0      0      0    9.6875       1      True     False      True
3    27.0      0      0    8.6625       1     False      True      True
4    22.0      1      1   12.2875       3     False      True     False
..    …    …     …        …     …       …        …       …
413  28.0      0      0    8.0500       1     False      True      True
414  39.0      0      0  108.9000       1     False     False     False
415  38.5      0      0    7.2500       1     False      True      True
416  20.0      0      0    8.0500       1     False      True      True
417  17.0      1      1   22.3583       3     False      True      True

     Embarked_Q  Embarked_S  family_size_Large  family_size_Medium
0          True       False              False               False
1         False        True              False                True
2          True       False              False               False
3         False        True              False               False
4         False        True              False                True
..          …         …                 …                  …
413       False        True              False               False
414       False       False              False               False
415       False        True              False               False
416       False        True              False               False
417       False       False              False                True

[418 rows x 12 columns]
```

[43]:
```
plt.figure(figsize=(15,6))
sns.heatmap(train.corr(),cmap='summer')
```

[43]:
```
<Axes: >
```



[44]:
```
x=train.iloc[:,1:].values
y=train.iloc[:,0].values
```

[45]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

[46]:
```
from sklearn.tree import DecisionTreeClassifier
classifier=DecisionTreeClassifier()
```

[47]:
```
classifier.fit(X_train,y_train)
```

[47]:
```
DecisionTreeClassifier()
```

[48]:
```
y_pred=classifier.predict(X_test)
```

[49]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)
```

[49]:
```
0.776536312849162
```

[50]:
```
Xf=test.iloc[:,:].values
```

[51]:
```
y_final=classifier.predict(Xf)
```

[52]:
```
y_final.shape
```

[52]:
```
(418,)
```

[53]:
```
passengerId.shape
```

[53]:
```
(418,)
```

[54]:
```
final=pd.DataFrame()
```

[55]:
```
final['passengerId']=passengerId
final['survived']=y_final
```

[56]:
```
final #0-indicates dead #1-indicates-survived
```

[56]:

	passengerId	survived
0	892	0
1	893	1
2	894	1
3	895	1
4	896	1
..	…	…
413	1305	0
414	1306	1
415	1307	0
416	1308	0
417	1309	1

[418 rows x 2 columns]

[ ]: