Titanic Survivor Prediction

November 8, 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)
```

[5]:
```
test.info()
```

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

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

[8]:
```
train['Embarked'].fillna('S',inplace=True)
```

[10]:
```
test['Fare'].fillna(test['Fare'].mean(), inplace=True)
```

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

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

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

```
[25]: train['Age'][np.isnan(train['Age'])]=gen_age
```

```
C:\Users\Nitish\Anaconda3\lib\site-packages\ipykernel_launcher.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: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
```

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

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

```
[27]: gen_age1=np.random.randint(test['Age'].mean()-test['Age'].std(),test['Age'].
        ↪mean()+test['Age'].std(), size=86)
```

```
[28]: test['Age'][np.isnan(test['Age'])]=gen_age1
```

```
C:\Users\Nitish\Anaconda3\lib\site-packages\ipykernel_launcher.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: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
```

[29]:
```
test.isnull().sum()
```

[29]:
```
PassengerId    0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

[30]:
```
train.isnull().sum()
```

[30]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

[31]:
```
train[['Pclass','Survived']].groupby('Pclass').mean()
```

[31]:
```
        Survived
Pclass
1       0.629630
2       0.472826
3       0.242363
```

[33]:
```
train[['Sex','Survived']].groupby('Sex').mean()
```

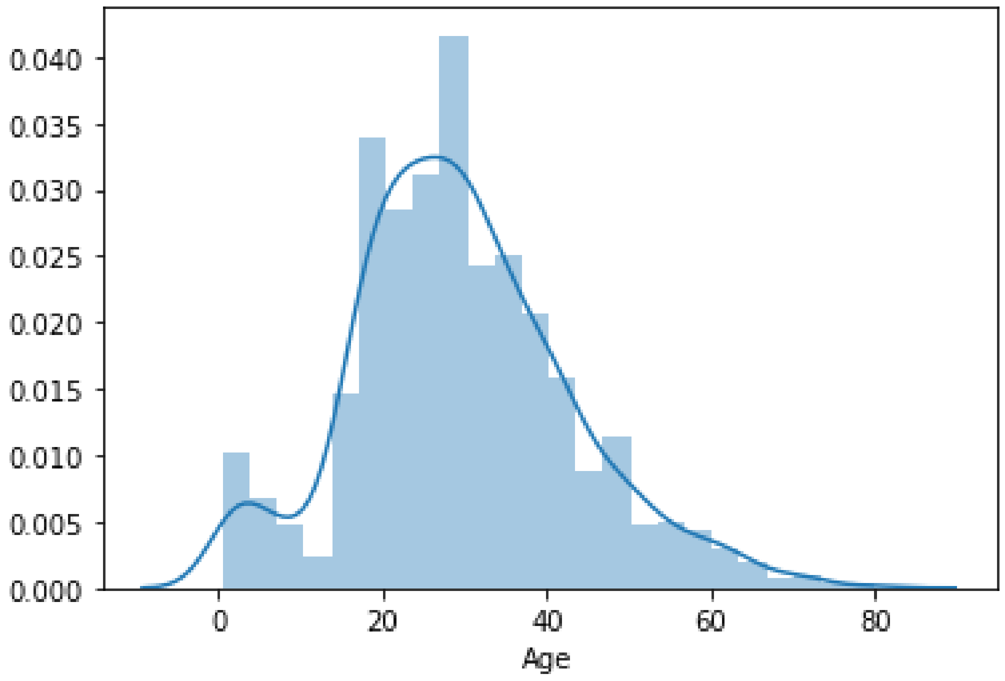
```
[33]:         Survived
      Sex
      female  0.742038
      male    0.188908
```

```
[34]: train[['Embarked','Survived']].groupby('Embarked').mean()
```

```
[34]:           Survived
      Embarked
      C         0.553571
      Q         0.389610
      S         0.339009
```

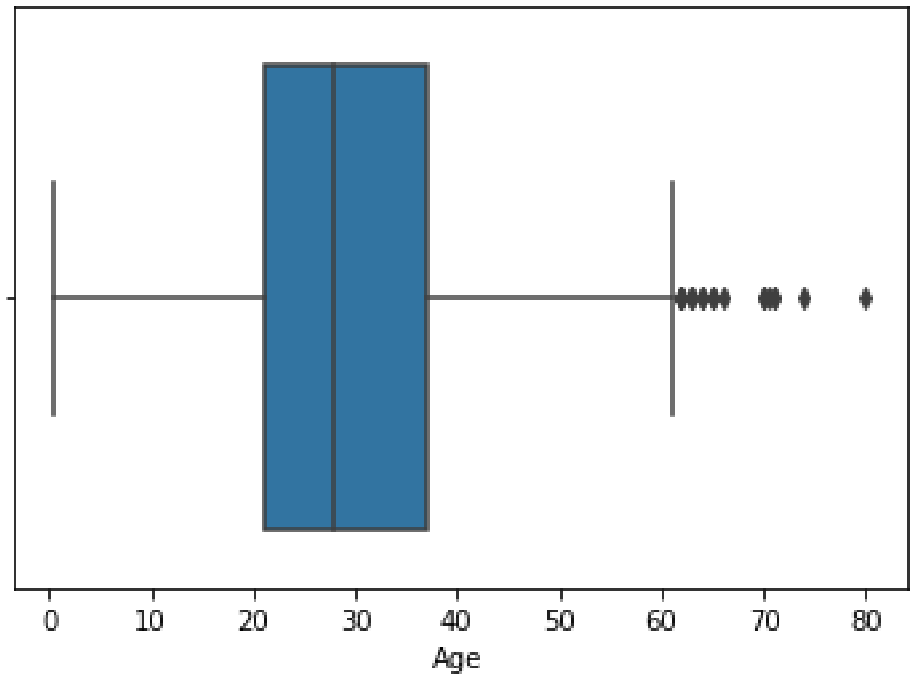
```
[35]: sns.distplot(train['Age'])
```

```
[35]: <matplotlib.axes._subplots.AxesSubplot at 0xdcf1301ba8>
```



```
[36]: sns.boxplot(train['Age'])
```

```
[36]: <matplotlib.axes._subplots.AxesSubplot at 0xdcf14362e8>
```

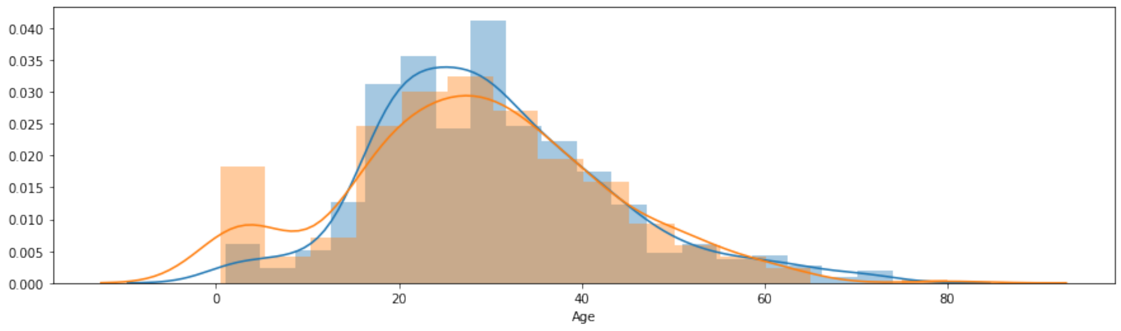


[45]:
```
train[train['Age']>75]['Survived'].value_counts()
```

[45]:
```
1    1
Name: Survived, dtype: int64
```

[49]:
```
plt.subplots(figsize=(15,4))
sns.distplot(train[train['Survived']==0]['Age'])
sns.distplot(train[train['Survived']==1]['Age'])
```

[49]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xdcf1f4ed68>
```



[51]:
```
passengerId=test['PassengerId'].values
```

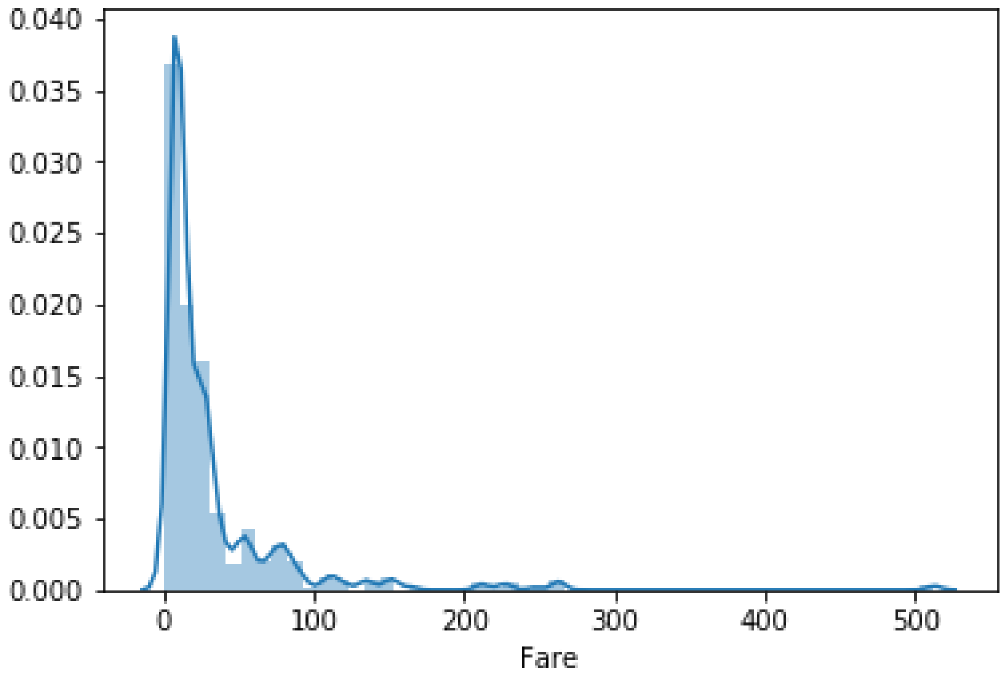
```
[53]: train.drop(columns=['PassengerId','Ticket'],inplace=True)
      test.drop(columns=['PassengerId','Ticket'],inplace=True)
```

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

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

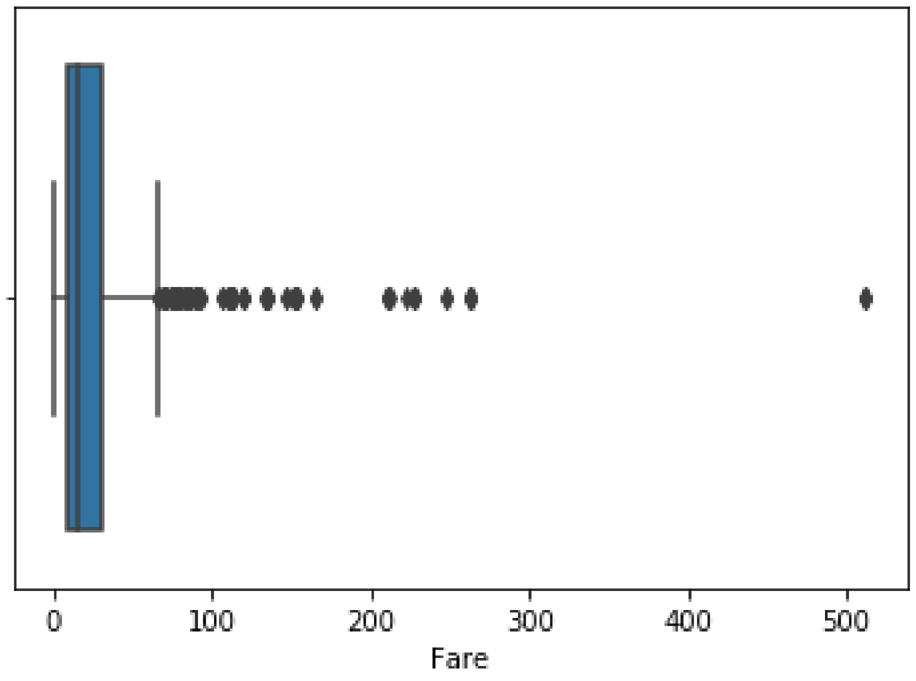
```
[55]: sns.distplot(train['Fare'])
```

```
[55]: <matplotlib.axes._subplots.AxesSubplot at 0xdcf03550f0>
```



```
[56]: sns.boxplot(train['Fare'])
```

```
[56]: <matplotlib.axes._subplots.AxesSubplot at 0xdcf20a3f60>
```

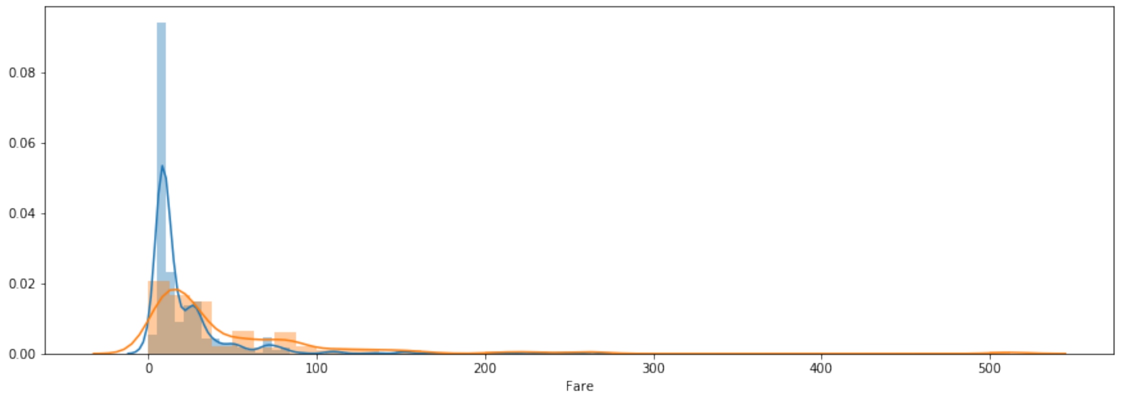


[67]:
```
train[train['Fare']>400]['Survived'].value_counts()
```

[67]:
```
1    3
Name: Survived, dtype: int64
```

[71]:
```
plt.subplots(figsize=(15,5))
sns.distplot(train[train['Survived']==0]['Fare'])
sns.distplot(train[train['Survived']==1]['Fare'])
```

[71]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xdcf24f1860>
```



```
[72]: # Don't delete this unless its 1st Jan
      train['Name']
```

```
[72]: 0                                  Braund, Mr. Owen Harris
      1      Cumings, Mrs. John Bradley (Florence Briggs Th…
      2                                   Heikkinen, Miss. Laina
      3         Futrelle, Mrs. Jacques Heath (Lily May Peel)
      4                                 Allen, Mr. William Henry
      5                                         Moran, Mr. James
      6                                  McCarthy, Mr. Timothy J
      7                           Palsson, Master. Gosta Leonard
      8      Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)
      9                     Nasser, Mrs. Nicholas (Adele Achem)
      10                       Sandstrom, Miss. Marguerite Rut
      11                              Bonnell, Miss. Elizabeth
      12                        Saundercock, Mr. William Henry
      13                           Andersson, Mr. Anders Johan
      14                  Vestrom, Miss. Hulda Amanda Adolfina
      15                      Hewlett, Mrs. (Mary D Kingcome)
      16                                  Rice, Master. Eugene
      17                          Williams, Mr. Charles Eugene
      18     Vander Planke, Mrs. Julius (Emelia Maria Vande…
      19                               Masselmani, Mrs. Fatima
      20                                  Fynney, Mr. Joseph J
      21                                 Beesley, Mr. Lawrence
      22                           McGowan, Miss. Anna "Annie"
      23                          Sloper, Mr. William Thompson
      24                         Palsson, Miss. Torborg Danira
      25     Asplund, Mrs. Carl Oscar (Selma Augusta Emilia…
      26                               Emir, Mr. Farred Chehab
      27                        Fortune, Mr. Charles Alexander
      28                         O'Dwyer, Miss. Ellen "Nellie"
      29                                   Todoroff, Mr. Lalio
                                    …
      861                          Giles, Mr. Frederick Edward
      862    Swift, Mrs. Frederick Joel (Margaret Welles Ba…
      863                      Sage, Miss. Dorothy Edith "Dolly"
      864                                Gill, Mr. John William
      865                              Bystrom, Mrs. (Karolina)
      866                          Duran y More, Miss. Asuncion
      867                  Roebling, Mr. Washington Augustus II
      868                            van Melkebeke, Mr. Philemon
      869                        Johnson, Master. Harold Theodor
      870                                     Balkic, Mr. Cerin
      871     Beckwith, Mrs. Richard Leonard (Sallie Monypeny)
      872                               Carlsson, Mr. Frans Olof
      873                            Vander Cruyssen, Mr. Victor
```

```
874                  Abelson, Mrs. Samuel (Hannah Wizosky)
875                       Najib, Miss. Adele Kiamie "Jane"
876                          Gustafsson, Mr. Alfred Ossian
877                                   Petroff, Mr. Nedelio
878                                     Laleff, Mr. Kristo
879         Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)
880          Shelley, Mrs. William (Imanita Parrish Hall)
881                                     Markun, Mr. Johann
882                           Dahlberg, Miss. Gerda Ulrika
883                          Banfield, Mr. Frederick James
884                                 Sutehall, Mr. Henry Jr
885                   Rice, Mrs. William (Margaret Norton)
886                                  Montvila, Rev. Juozas
887                           Graham, Miss. Margaret Edith
888               Johnston, Miss. Catherine Helen "Carrie"
889                                  Behr, Mr. Karl Howell
890                                    Dooley, Mr. Patrick
Name: Name, Length: 891, dtype: object
```

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

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

```
[76]: train.drop(columns=['SibSp','Parch'],inplace=True)
      test.drop(columns=['SibSp','Parch'],inplace=True)
```

```
[78]: train['family'].value_counts()
```

```
[78]: 1     537
      2     161
      3     102
      4      29
      6      22
      5      15
      7      12
      11      7
      8       6
      Name: family, dtype: int64
```

```
[79]: train[['family','Survived']].groupby('family').mean()
```

```
[79]:         Survived
      family
      1       0.303538
      2       0.552795
```

```
3       0.578431
4       0.724138
5       0.200000
6       0.136364
7       0.333333
8       0.000000
11      0.000000
```

[80]:
```
def family_size(number):
    if number==1:
        return "Alone"
    elif number>1 and number <5:
        return "Small"
    else:
        return "Large"
```

[81]:
```
family_size(5)
```

[81]:
```
'Large'
```

[82]:
```
train['family_size']=train['family'].apply(family_size)
```

[84]:
```
test['family_size']=test['family'].apply(family_size)
```

[85]:
```
train.drop(columns=['family'],inplace=True)
test.drop(columns=['family'],inplace=True)
```

[87]:
```
y=train['Survived'].values
y
```

[87]:
```
array([0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1,
       1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1,
       1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0,
       1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0,
       0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1,
       0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1,
       1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0,
       0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0,
       0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0,
       1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1,
       1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
       1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0,
       0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0,
       1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1,
       0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0], dtype=int64)
```

```
[88]: train.drop(columns=['Survived'],inplace=True)
```

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

```
(891, 6)
(418, 6)
```

```
[94]: final=train.append(test)
```

```
[95]: final.shape
```

```
[95]: (1309, 6)
```

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

```
[103]: Xf=final.tail(418).values
```

```
[106]: X=final.head(891).values
```

```
[107]: X.shape
```

```
[107]: (891, 9)
```

```
[108]: y.shape
```

```
[108]: (891,)
```

```
[109]: 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)
```

```
[110]: from sklearn.tree import DecisionTreeClassifier
       clf=DecisionTreeClassifier()
```

```
[111]: clf.fit(X_train,y_train)
```

```
[111]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                              max_features=None, max_leaf_nodes=None,
                              min_impurity_decrease=0.0, min_impurity_split=None,
                              min_samples_leaf=1, min_samples_split=2,
                              min_weight_fraction_leaf=0.0, presort=False,
                              random_state=None, splitter='best')
```

```
[112]: y_pred=clf.predict(X_test)
```

```
[113]: y_pred.shape
```

```
[113]: (179,)
```

```
[114]: y_test.shape
```

```
[114]: (179,)
```

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

```
[115]: 0.7877094972067039
```

```
[118]: yf=clf.predict(Xf)
```

```
[119]: yf.shape
```

```
[119]: (418,)
```

```
[120]: submission=pd.DataFrame()
```

```
[121]: submission['PassengerId']=passengerId
       submission['Survived']=yf
```

```
[124]: submission.to_csv('submission.csv', index=False)
```

[ ]:
```

```