Decision Tree Classification (Titanic Data Set)

August 13, 2022

0.1 Questions & Answers

-

0.1.1 Loading Libraries

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

```
[2]: import numpy as np
```

```
[3]: from matplotlib import pyplot as plt
```

```
[4]: %matplotlib inline
```

```
[5]: df = pd.read_csv("../../datasets/titanic.csv")
```

```
[6]: df.head()
```

```
[6]:    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            Ticket     Fare Cabin Embarked
     0      0         A/5 21171   7.2500   NaN        S
     1      0          PC 17599  71.2833   C85        C
     2      0  STON/O2. 3101282   7.9250   NaN        S
```

```
3       0             113803  53.1000  C123         S
4       0             373450   8.0500   NaN         S
```

0.1.2 Exploring Data

[7]:

```
#Droping colum because we do not need id column
df = df.drop(['PassengerId'], axis=1)
```

[8]:

```
df.head()
```

[8]:

```
   Survived  Pclass                                               Name  \
0         0       3                            Braund, Mr. Owen Harris
1         1       1  Cumings, Mrs. John Bradley (Florence Briggs Th…
2         1       3                             Heikkinen, Miss. Laina
3         1       1       Futrelle, Mrs. Jacques Heath (Lily May Peel)
4         0       3                           Allen, Mr. William Henry

      Sex   Age  SibSp  Parch            Ticket     Fare Cabin Embarked
0    male  22.0      1      0         A/5 21171   7.2500   NaN        S
1  female  38.0      1      0          PC 17599  71.2833   C85        C
2  female  26.0      0      0  STON/O2. 3101282   7.9250   NaN        S
3  female  35.0      1      0            113803  53.1000  C123        S
4    male  35.0      0      0            373450   8.0500   NaN        S
```

[9]:

```
len(df.columns) #columns
```

[9]:

```
11
```

[10]:

```
len(df) #Rows
```

[10]:

```
891
```

[11]:

```
#lets check the data set
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    int64
 1   Pclass    891 non-null    int64
 2   Name      891 non-null    object
 3   Sex       891 non-null    object
 4   Age       714 non-null    float64
 5   SibSp     891 non-null    int64
 6   Parch     891 non-null    int64
 7   Ticket    891 non-null    object
 8   Fare      891 non-null    float64
 9   Cabin     204 non-null    object
 10  Embarked  889 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 76.7+ KB
```

[12]:
```
#lets see the statistic overview
df.describe()
```

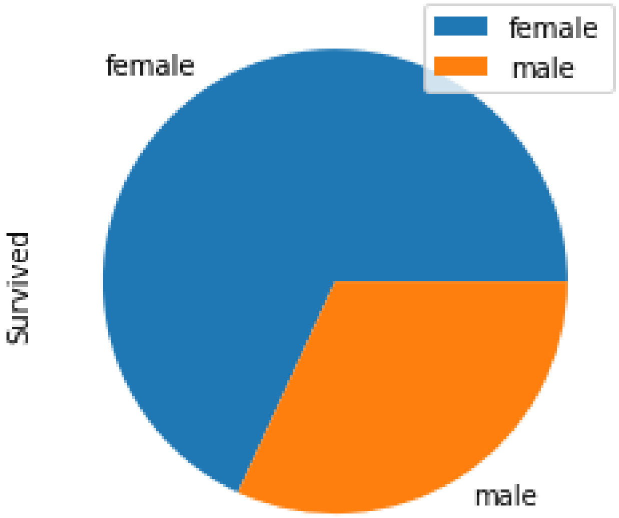
[12]:
```
         Survived      Pclass         Age       SibSp       Parch        Fare
count  891.000000  891.000000  714.000000  891.000000  891.000000  891.000000
mean     0.383838    2.308642   29.699118    0.523008    0.381594   32.204208
std      0.486592    0.836071   14.526497    1.102743    0.806057   49.693429
min      0.000000    1.000000    0.420000    0.000000    0.000000    0.000000
25%      0.000000    2.000000   20.125000    0.000000    0.000000    7.910400
50%      0.000000    3.000000   28.000000    0.000000    0.000000   14.454200
75%      1.000000    3.000000   38.000000    1.000000    0.000000   31.000000
max      1.000000    3.000000   80.000000    8.000000    6.000000  512.329200
```

[13]:
```
#finding null values
df.isnull().sum() #we saw there are null values in it
```

[13]:
```
Survived      0
Pclass        0
Name          0
Sex           0
Age         177
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin       687
Embarked      2
dtype: int64
```

[14]:
```
df.groupby(['Sex']).sum().plot(kind='pie', y='Survived')
```

[14]:
```
<AxesSubplot:ylabel='Survived'>
```



```
[15]: df.head()
```

```
[15]:    Survived  Pclass                                               Name  \
      0         0       3                            Braund, Mr. Owen Harris
      1         1       1  Cumings, Mrs. John Bradley (Florence Briggs Th…
      2         1       3                             Heikkinen, Miss. Laina
      3         1       1       Futrelle, Mrs. Jacques Heath (Lily May Peel)
      4         0       3                           Allen, Mr. William Henry

            Sex   Age  SibSp  Parch            Ticket     Fare Cabin Embarked
      0    male  22.0      1      0         A/5 21171   7.2500   NaN        S
      1  female  38.0      1      0          PC 17599  71.2833   C85        C
      2  female  26.0      0      0  STON/O2. 3101282   7.9250   NaN        S
      3  female  35.0      1      0            113803  53.1000  C123        S
      4    male  35.0      0      0            373450   8.0500   NaN        S
```

```
[17]: #we will drop the target column as inputs this is what we will find
      df.drop("Survived",axis="columns")
```

```
[17]:      Pclass                                               Name     Sex   Age  \
      0         3                            Braund, Mr. Owen Harris    male  22.0
      1         1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0
      2         3                             Heikkinen, Miss. Laina  female  26.0
      3         1       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
      4         3                           Allen, Mr. William Henry    male  35.0
      ..      …                                                  …     …     …
      886       2                              Montvila, Rev. Juozas    male  27.0
```

```
887       1                       Graham, Miss. Margaret Edith  female  19.0
888       3            Johnston, Miss. Catherine Helen "Carrie"  female   NaN
889       1                              Behr, Mr. Karl Howell    male  26.0
890       3                                Dooley, Mr. Patrick    male  32.0

     SibSp  Parch            Ticket     Fare Cabin Embarked
0        1      0         A/5 21171   7.2500   NaN        S
1        1      0          PC 17599  71.2833   C85        C
2        0      0  STON/O2. 3101282   7.9250   NaN        S
3        1      0            113803  53.1000  C123        S
4        0      0            373450   8.0500   NaN        S
..     …    …               …      …     …      …
886      0      0            211536  13.0000   NaN        S
887      0      0            112053  30.0000   B42        S
888      1      2        W./C. 6607  23.4500   NaN        S
889      0      0            111369  30.0000  C148        C
890      0      0            370376   7.7500   NaN        Q

[891 rows x 10 columns]
```

[19]:

```
#now lets check our target
target = df["Survived"]
```

[20]:

```
target #it come in shape of numpy form
```

[20]:

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

[82]:

```
df.head()
```

[82]:

```
   Pclass  Sex  Parch
0       3    1      0
1       1    0      0
2       3    0      0
3       1    0      0
4       3    1      0
```

```
[83]: df.head()
```

```
[83]:    Pclass  Sex  Parch
      0       3    1      0
      1       1    0      0
      2       3    0      0
      3       1    0      0
      4       3    1      0
```

```
[87]: from sklearn.preprocessing import LabelEncoder
```

```
[88]: t_Sex = LabelEncoder()
```

```
[89]: t_Pclass = LabelEncoder()
```

```
[90]: t_Parch = LabelEncoder()
```

```
[92]: df['sex'] = t_Name.fit_transform(df['Sex'])
```

```
[93]: df['Parch'] = t_fare.fit_transform(df['Parch'])
```

```
[37]: df['Age'] = t_age.fit_transform(df['Age'])
```

```
[94]: df.head()
```

```
[94]:    Pclass  Sex  Parch  sex
      0       3    1      0    1
      1       1    0      0    0
      2       3    0      0    0
      3       1    0      0    0
      4       3    1      0    1
```

```
[102]: df.drop("Sex",axis="columns")
```

```
[102]:      Pclass  Parch  sex
       0         3      0    1
       1         1      0    0
       2         3      0    0
       3         1      0    0
       4         3      0    1
       ..      …      …    …
       886       2      0    1
       887       1      0    0
       888       3      2    0
       889       1      0    1
       890       3      0    1
```

```
[891 rows x 3 columns]
```

```
[103]: from sklearn import tree
```

```
[104]: model = tree.DecisionTreeClassifier()
```

```
[105]: model.fit(df,target)
```

```
[105]: DecisionTreeClassifier()
```

```
[110]: model.score(df,target)
```

```
[110]: 0.8058361391694725
```

```
[112]: model.predict([[890,1,0,1]])
```

```
/home/muhammadsardardaudkhan/.local/lib/python3.8/site-
packages/sklearn/base.py:450: UserWarning: X does not have valid feature names,
but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

```
[112]: array([0])
```

```
[113]: df.tail()
```

```
[113]:      Pclass  Sex  Parch  sex
       886       2    1      0    1
       887       1    0      0    0
       888       3    0      2    0
       889       1    1      0    1
       890       3    1      0    1
```