

Exercise 6

Exercise: Build decision tree model to predict survival based on certain parameters

Image Alt Text



In this file using following columns build a model to predict if person would survive or not

1. Pclass
2. Sex
3. Age
4. Fare

Calculate score of your model

```
In [64]: import pandas as pd
import matplotlib.pyplot as plt
from sklearn import tree
from sklearn.model_selection import train_test_split
```

```
In [65]: data = pd.read_csv('titanic.csv')
data
```

Out[65]:	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
	0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
	1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
	2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
	3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
	4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

	886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
	887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
	888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
	889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
	890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

Checking shape of the data

```
In [66]: data.shape
```

```
Out[66]: (891, 12)
```

Printing how many people survived and died

```
In [67]: print(f"Number of people survived : {len(data[data['Survived'] == 1])}")
print(f"Number of people death : {len(data[data['Survived'] == 0])}")
```

Number of people survived : 342

Number of people death : 549

Splitting into X and y

```
In [68]: X = data[['Pclass', 'Sex', 'Age', 'Fare']]
y = data['Survived']
```

Label Encoding on sex column and creating new le_sex column

```
In [69]: from sklearn.preprocessing import LabelEncoder
```

```
In [70]: le_sex = LabelEncoder()
X['le_sex'] = le_sex.fit_transform(X['Sex'])
X
```

C:\Users\User\AppData\Local\Temp\ipykernel_7844\1216391553.py:2: SettingWithCopyWarning:
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_guide/indexing.html#returning-a-view-versus-a-copy
X['le_sex'] = le_sex.fit_transform(X['Sex'])

```
Out[70]:
```

	Pclass	Sex	Age	Fare	le_sex
0	3	male	22.0	7.2500	1
1	1	female	38.0	71.2833	0
2	3	female	26.0	7.9250	0
3	1	female	35.0	53.1000	0
4	3	male	35.0	8.0500	1
...
886	2	male	27.0	13.0000	1
887	1	female	19.0	30.0000	0
888	3	female	NaN	23.4500	0
889	1	male	26.0	30.0000	1
890	3	male	32.0	7.7500	1

891 rows × 5 columns

Dropping the sex column

```
In [71]: X_new = X.drop(['Sex'], axis=1)
X_new
```

```
Out[71]:
```

	Pclass	Age	Fare	le_sex
0	3	22.0	7.2500	1
1	1	38.0	71.2833	0
2	3	26.0	7.9250	0
3	1	35.0	53.1000	0
4	3	35.0	8.0500	1
...
886	2	27.0	13.0000	1
887	1	19.0	30.0000	0
888	3	NaN	23.4500	0
889	1	26.0	30.0000	1
890	3	32.0	7.7500	1

891 rows × 4 columns

Train, Test dataset splitting

```
In [86]: from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_new, y, test_size=0.3)
```

Model instance and training the model

```
In [87]: model = tree.DecisionTreeClassifier()
```

```
In [88]: model.fit(X_train, y_train)
```

```
Out[88]: DecisionTreeClassifier
DecisionTreeClassifier()
```

Checking the score

```
In [89]: model.score(X_test, y_test)
```

```
Out[89]: 0.7947761194029851
```

Printing total number of people in the Titanic

```
In [90]: print(f"Total People present in the Titanic : {len(X_new['le_sex'])}")
```

Total People present in the Titanic : 891

Printing num of male and female in Titanic

```
In [91]: num_female = len(X_new[X_new['le_sex'] == 0])
print(f"Num of female in titanic : {num_female}")

num_male = len(X_new[X_new['le_sex'] == 1])
print(f"Num of male in titanic : {num_male}")
```

Num of female in titanic : 314

Num of male in titanic : 577

Printing how num of male and female survived

```
In [92]: num_female_survived = len(X_new[(X_new['le_sex'] == 0) & (y == 1)])
num_male_survived = len(X_new[(X_new['le_sex'] == 1) & (y == 1)])
print(f'Num of female survived : {num_female_survived}')
print(f'Num of male survived : {num_male_survived}')
```

Num of female survived : 233

Num of male survived : 109

Printing num of male and female death

```
In [93]: num_female_died = len(X_new[(X_new['le_sex'] == 0) & (y == 0)])
num_male_died = len(X_new[(X_new['le_sex'] == 1) & (y == 0)])
print(f'Num of female death : {num_female_died}')
print(f'Num of male death : {num_male_died}')
```

Num of female death : 81

Num of male death : 468

```
In [111] X_new.head()
```

```
Out[111]:
```

	Pclass	Age	Fare	le_sex
0	3	22.0	7.2500	1
1	1	38.0	71.2833	0
2	3	26.0	7.9250	0
3	1	35.0	53.1000	0
4	3	35.0	8.0500	1

Predicting a particular value

```
In [121] model.predict([[3, 22, 7.2500, 1]])
```

C:\Users\User\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:465: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
warnings.warn(

```
Out[121]: array([0], dtype=int64)
```

Printing the value at index 1 and checking it

```
In [127] model.predict([X_new.iloc[1]])
```

C:\Users\User\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:465: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
warnings.warn(

```
Out[127]: array([1], dtype=int64)
```

```
In [128] y[1]
```

```
Out[128]: 1
```

Predicting on the actual test set and making it a new DataFrame

```
In [132] predict = model.predict(X_test)
```

```
In [140] obj = {
    "Pclass" : X_test['Pclass'],
    "Age" : X_test['Age'],
    "Fare" : X_test['Fare'],
    "le_sex" : X_test['le_sex'],
```

```
"Actual Value" : y_test,  
"Predicted Value" : predict  
}  
pd.options.display.max_rows = 500  
newdf = pd.DataFrame(obj)  
newdf.shape
```

Out[140]: (268, 6)

Checking how many right and wrong predictions

```
In [141]: len(newdf[newdf['Actual Value'] == newdf['Predicted Value']])
```

Out[141]: 213

```
In [143]: len(newdf[newdf['Actual Value'] != newdf['Predicted Value']])
```

Out[143]: 55

```
In [150]: plt.hist(data['Age'], bins=10)
```

Out[150]: (array([54., 46., 177., 169., 118., 70., 45., 24., 9., 2.]),
array([0.42, 8.378, 16.336, 24.294, 32.252, 40.21, 48.168, 56.126,
64.084, 72.042, 80.]),
<BarContainer object of 10 artists>)

