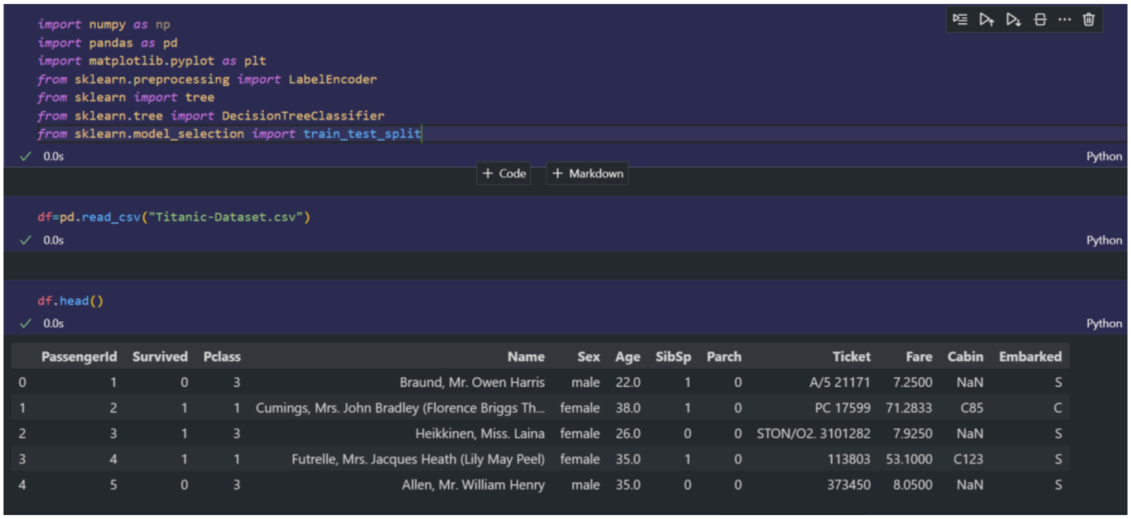
M.L_Assignment_4_ [Decision Tree algorithm implementation]
Name: Arya Chakraborty
Regd. No: 22MSD7020
Date: 3/9/2023

Question:

Consider the Titanic dataset, summarized according to economic status (class), sex,age and survival. Using Decision Tree algorithm you are asked to predict whether a passenger on the titanic would have been survived or not. Justify your answers with your model accuracy.

Ans:

Importing all required libraries and the dataset into the local machine and viewing the top 5 rows.



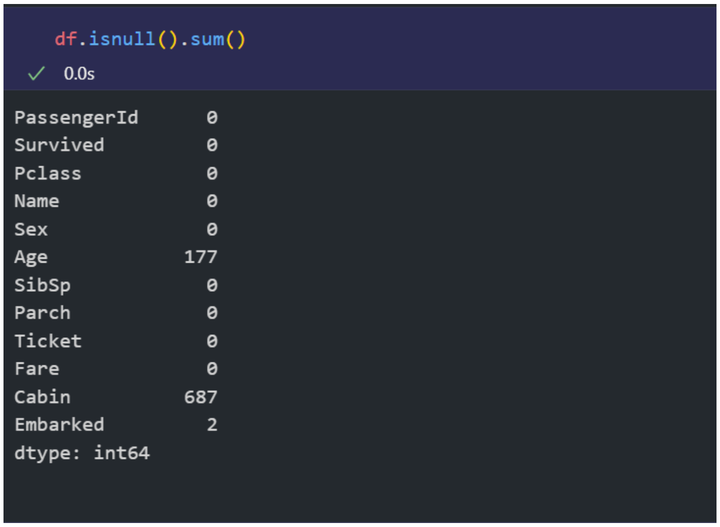
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
```

```python
df=pd.read_csv("Titanic-Dataset.csv")
```

```python
df.head()
```

	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

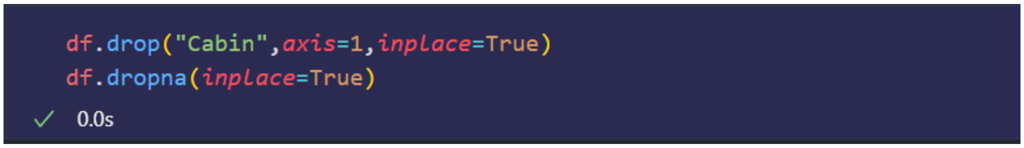
To check for null values in the dataset we use isnull() function.



```python
df.isnull().sum()
```

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

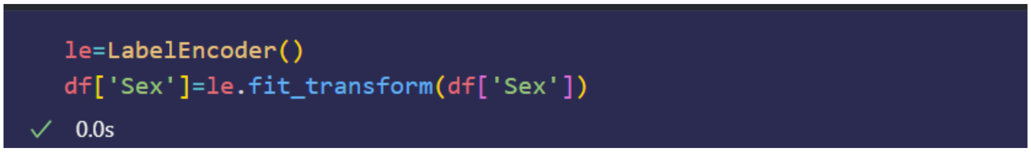
We may drop the “Cabin” Column as it isn’t relevant to our modelling and we delete other rows consisting null values of “Age” attribute.



```
df.drop("Cabin",axis=1,inplace=True)
df.dropna(inplace=True)
```

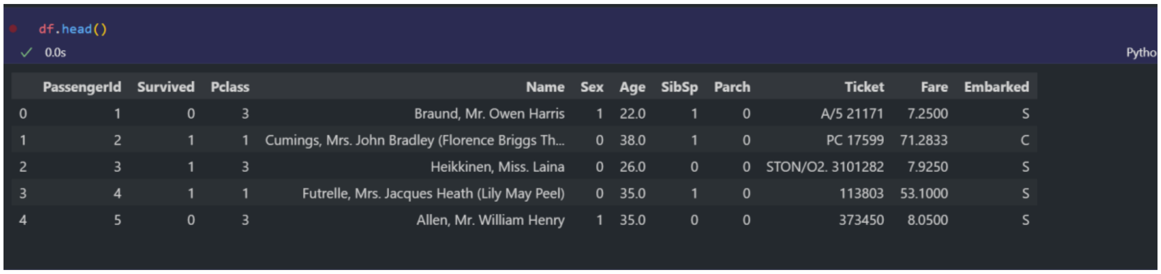
✓ 0.0s

As the “Sex” attribute is important and it’s categorical we transform it to numerical values using LabelEncoder module from sklearn.



```
le=LabelEncoder()
df['Sex']=le.fit_transform(df['Sex'])
```

✓ 0.0s

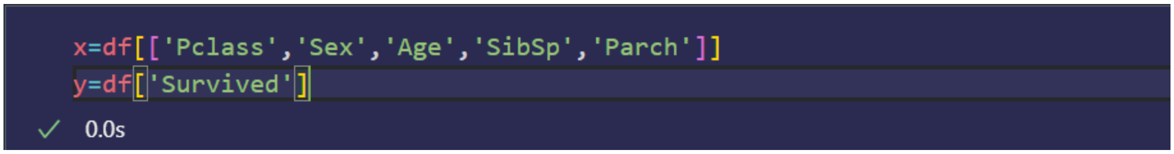


```
df.head()
```

✓ 0.0s

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

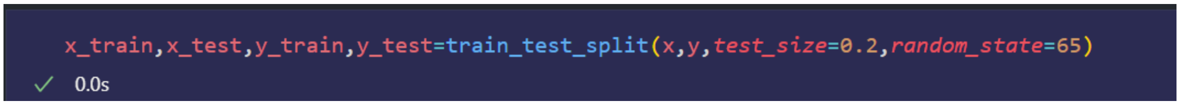
We take “Pclass”,”Sex”,”Age”,”SibSp”,”Parch” in X variable as concepts and “Survived” attribute in Y as target variable.



```
x=df[['Pclass','Sex','Age','SibSp','Parch']]
y=df['Survived']
```

✓ 0.0s

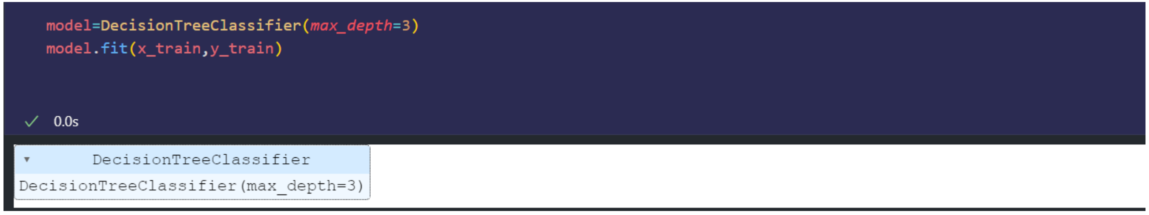
We divide both x & y into train and test data set using train_test_split.



```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=65)
```

✓ 0.0s

Then fit the DicisionTreeClassifier model with x_train & x_test data. Here the max depth of the tree is taken 3.

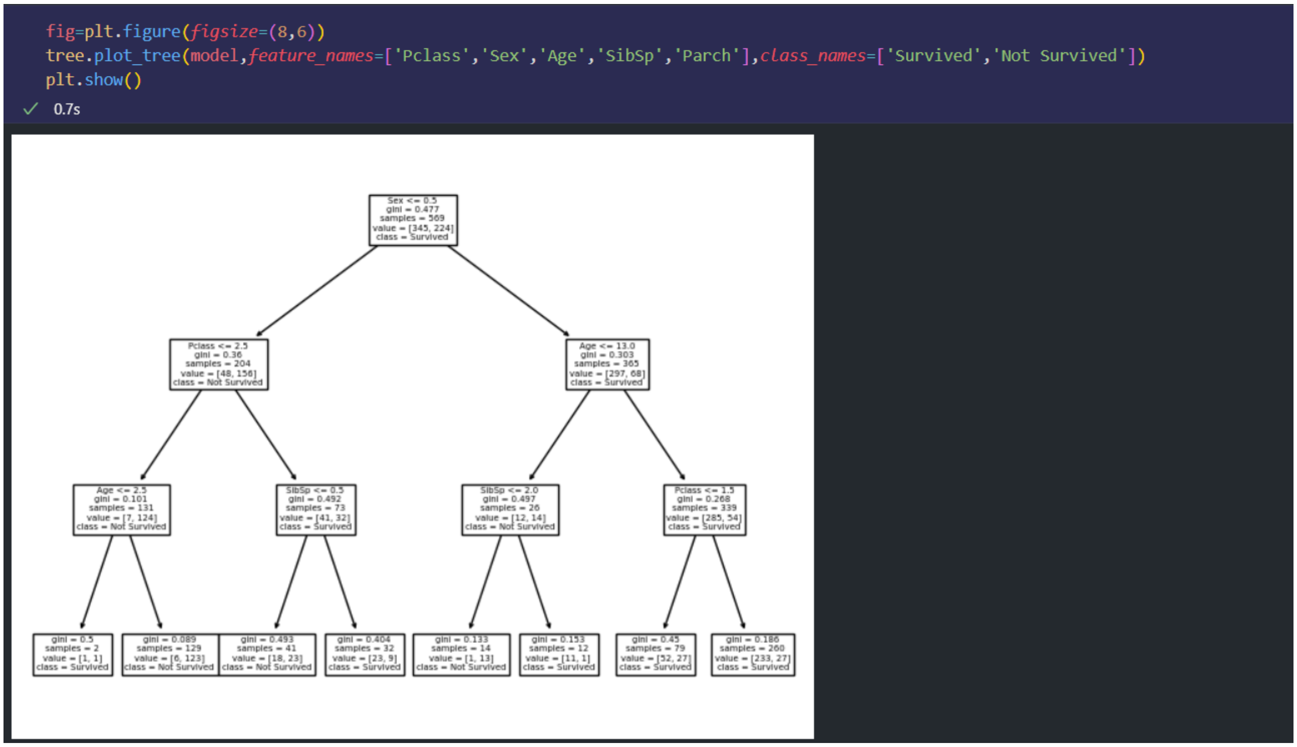


```
model=DecisionTreeClassifier(max_depth=3)
model.fit(x_train,y_train)
```

✓ 0.0s

```
▾ DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3)
```

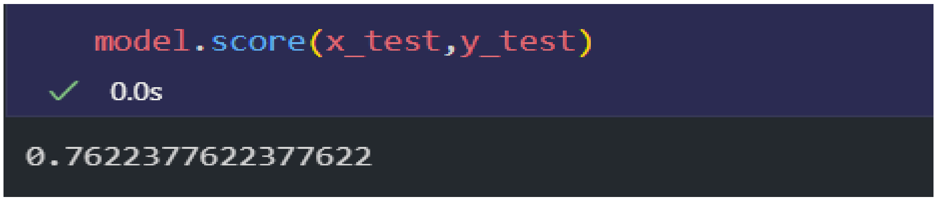
Then we visualize the tree using tree.plot_tree function.



```
fig=plt.figure(figsize=(8,6))
tree.plot_tree(model,feature_names=['Pclass','Sex','Age','SibSp','Parch'],class_names=['Survived','Not Survived'])
plt.show()
```

✓ 0.7s

Then we find the accuracy score of the model using test data set.

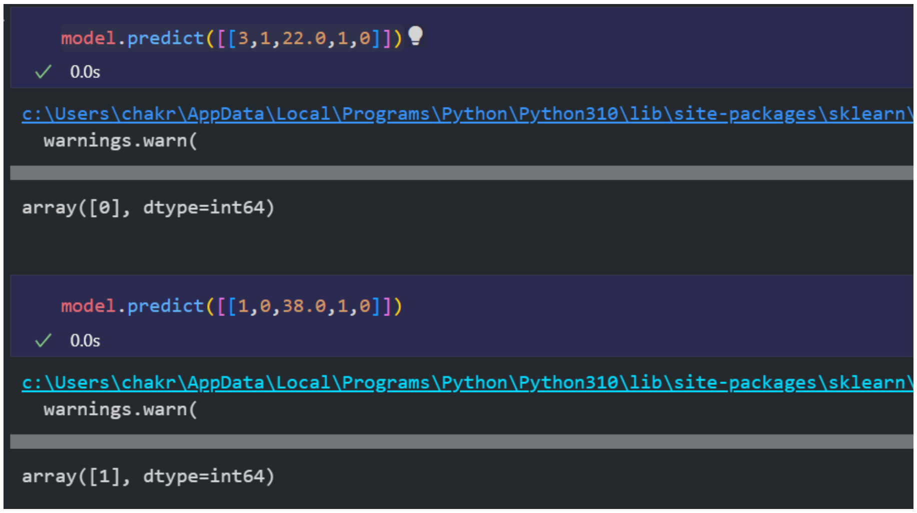


```
model.score(x_test,y_test)
```

✓ 0.0s

```
0.7622377622377622
```

It concludes that the model is roughly 76.2% accurate with the underlying data.Further to test the prediction capability we chose two sample rows from x variable and give the model to predict based on those values.



```
model.predict([[3,1,22.0,1,0]])
```

✓ 0.0s

```
c:\Users\chakr\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\
  warnings.warn(

array([0], dtype=int64)
```

```
model.predict([[1,0,38.0,1,0]])
```

✓ 0.0s

```
c:\Users\chakr\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\
  warnings.warn(

array([1], dtype=int64)
```

So we can see our model is classifying the target attribute pretty good.