

MACHINE LEARNING

Submitted by,

- Name:Jagannath v v
- Reg.No.-21122024
- class:1st MSc DataScience

LAB OVERVIEW

(1)Perform Classification using Decision Trees.

(2)Demonstrate Multiple Datasets, do the necessary EDA and show various evaluation metrics.

PROBLEM DEFINITION

- Used Pandas to Import the Dataset
- Do the necessary Exploratory Data Analysis
- Use the train_test_split method available in SCIKIT to split the dataset into Train Dataset and Test Dataset.
- AndS hows the accuracy_Score, classification_report,etc.

APPROACH

Here i reffered internet to find some more about the libraries and visualisation tools/functions And try to EDA on dataset

EDA

In [1]: `import pandas as pd`

In [92]: `df=pd.read_csv(r"C:\Users\jagan\OneDrive\Desktop\NFLX.csv")`

In [93]: `df`

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-02-05	262.000000	267.899994	250.029999	254.259995	254.259995	11896100

	Date	Open	High	Low	Close	Adj Close	Volume
1	2018-02-06	247.699997	266.700012	245.000000	265.720001	265.720001	12595800
2	2018-02-07	266.579987	272.450012	264.329987	264.559998	264.559998	8981500
3	2018-02-08	267.079987	267.619995	250.000000	250.100006	250.100006	9306700
4	2018-02-09	253.850006	255.800003	236.110001	249.470001	249.470001	16906900
...
1004	2022-01-31	401.970001	427.700012	398.200012	427.140015	427.140015	20047500
1005	2022-02-01	432.959991	458.480011	425.540009	457.130005	457.130005	22542300
1006	2022-02-02	448.250000	451.980011	426.480011	429.480011	429.480011	14346000
1007	2022-02-03	421.440002	429.260010	404.279999	405.600006	405.600006	9905200
1008	2022-02-04	407.309998	412.769989	396.640015	410.170013	410.170013	7782400

1009 rows × 7 columns

In [94]: df.tail()

	Date	Open	High	Low	Close	Adj Close	Volume
1004	2022-01-31	401.970001	427.700012	398.200012	427.140015	427.140015	20047500
1005	2022-02-01	432.959991	458.480011	425.540009	457.130005	457.130005	22542300
1006	2022-02-02	448.250000	451.980011	426.480011	429.480011	429.480011	14346000
1007	2022-02-03	421.440002	429.260010	404.279999	405.600006	405.600006	9905200
1008	2022-02-04	407.309998	412.769989	396.640015	410.170013	410.170013	7782400

In [95]: df.describe

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-02-05	262.000000	267.899994	250.029999	254.259995	254.259995	11896100
1	2018-02-06	247.699997	266.700012	245.000000	265.720001	265.720001	12595800
2	2018-02-07	266.579987	272.450012	264.329987	264.559998	264.559998	8981500
3	2018-02-08	267.079987	267.619995	250.000000	250.100006	250.100006	9306700
4	2018-02-09	253.850006	255.800003	236.110001	249.470001	249.470001	16906900
...
1004	2022-01-31	401.970001	427.700012	398.200012	427.140015	427.140015	20047500
1005	2022-02-01	432.959991	458.480011	425.540009	457.130005	457.130005	22542300
1006	2022-02-02	448.250000	451.980011	426.480011	429.480011	429.480011	14346000
1007	2022-02-03	421.440002	429.260010	404.279999	405.600006	405.600006	9905200
1008	2022-02-04	407.309998	412.769989	396.640015	410.170013	410.170013	7782400

```
4      16906900
...
1004   20047500
1005   22542300
1006   14346000
1007   9905200
1008   7782400
```

[1009 rows x 7 columns]>

In [96]: df.shape

Out[96]: (1009, 7)

In [97]: #df.drop(['Player Photo', 'Club Logo'], axis=1, inplace=True)

In [98]: #df.drop(['Flag Photo'], axis=1, inplace=True)

In [99]: df.isnull().sum()

Out[99]:

	Date	Open	High	Low	Close	Adj Close	Volume
Date	0						
Open	0						
High	0						
Low	0						
Close	0						
Adj Close	0						
Volume	0						
dtype:	int64						

In [100...]: df.describe(include="all")

Out[100...]:

	Date	Open	High	Low	Close	Adj Close	Volume
count	1009	1009.000000	1009.000000	1009.000000	1009.000000	1009.000000	1.009000e+03
unique	1009	NaN	NaN	NaN	NaN	NaN	NaN
top	2018-02-05	NaN	NaN	NaN	NaN	NaN	NaN
freq	1	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	419.059673	425.320703	412.374044	419.000733	419.000733	7.570685e+06
std	NaN	108.537532	109.262960	107.555867	108.289999	108.289999	5.465535e+06
min	NaN	233.919998	250.649994	231.229996	233.880005	233.880005	1.144000e+06
25%	NaN	331.489990	336.299988	326.000000	331.619995	331.619995	4.091900e+06
50%	NaN	377.769989	383.010010	370.880005	378.670013	378.670013	5.934500e+06
75%	NaN	509.130005	515.630005	502.529999	509.079987	509.079987	9.322400e+06
max	NaN	692.349976	700.989990	686.090027	691.690002	691.690002	5.890430e+07

Graphical representation

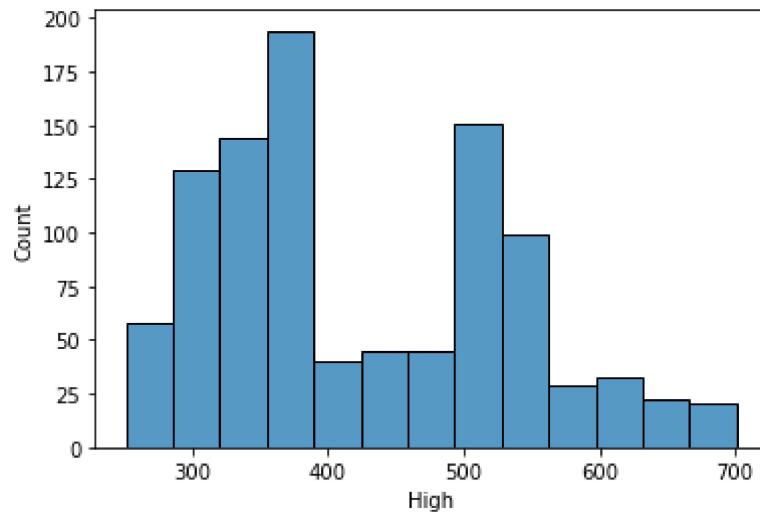
In [101...]

```
import seaborn as sns  
import matplotlib as plt
```

histogram

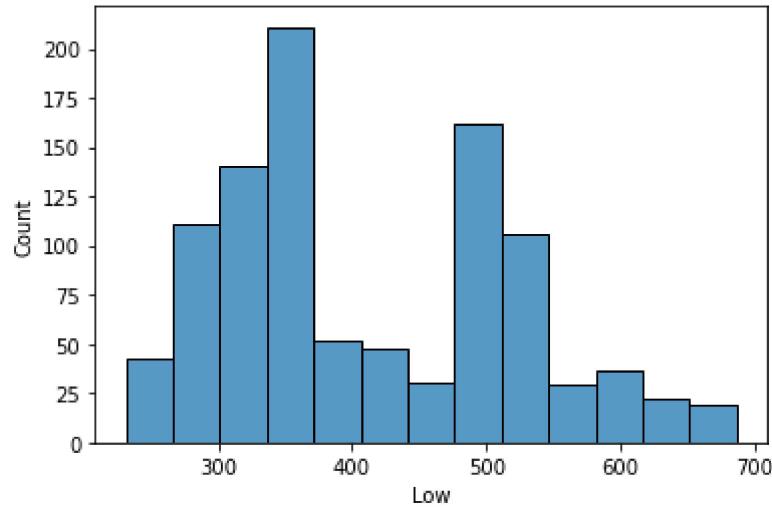
In [102...]

```
A=sns.histplot(data=df,x="High")
```



In [103...]

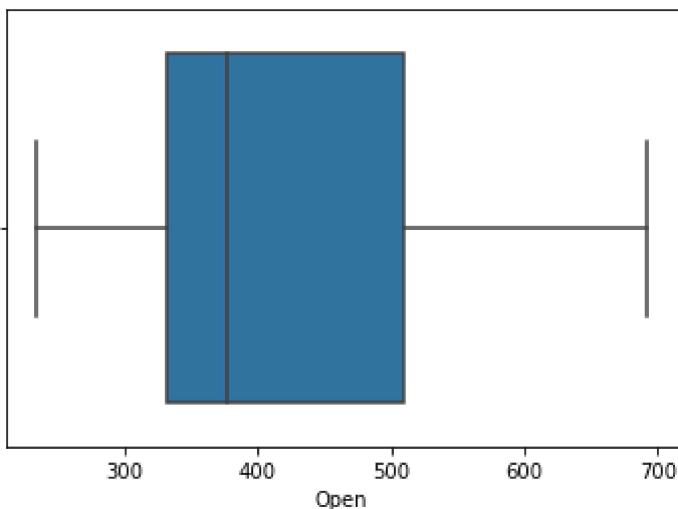
```
b=sns.histplot(data=df,x="Low")
```



boxplot

In [104...]

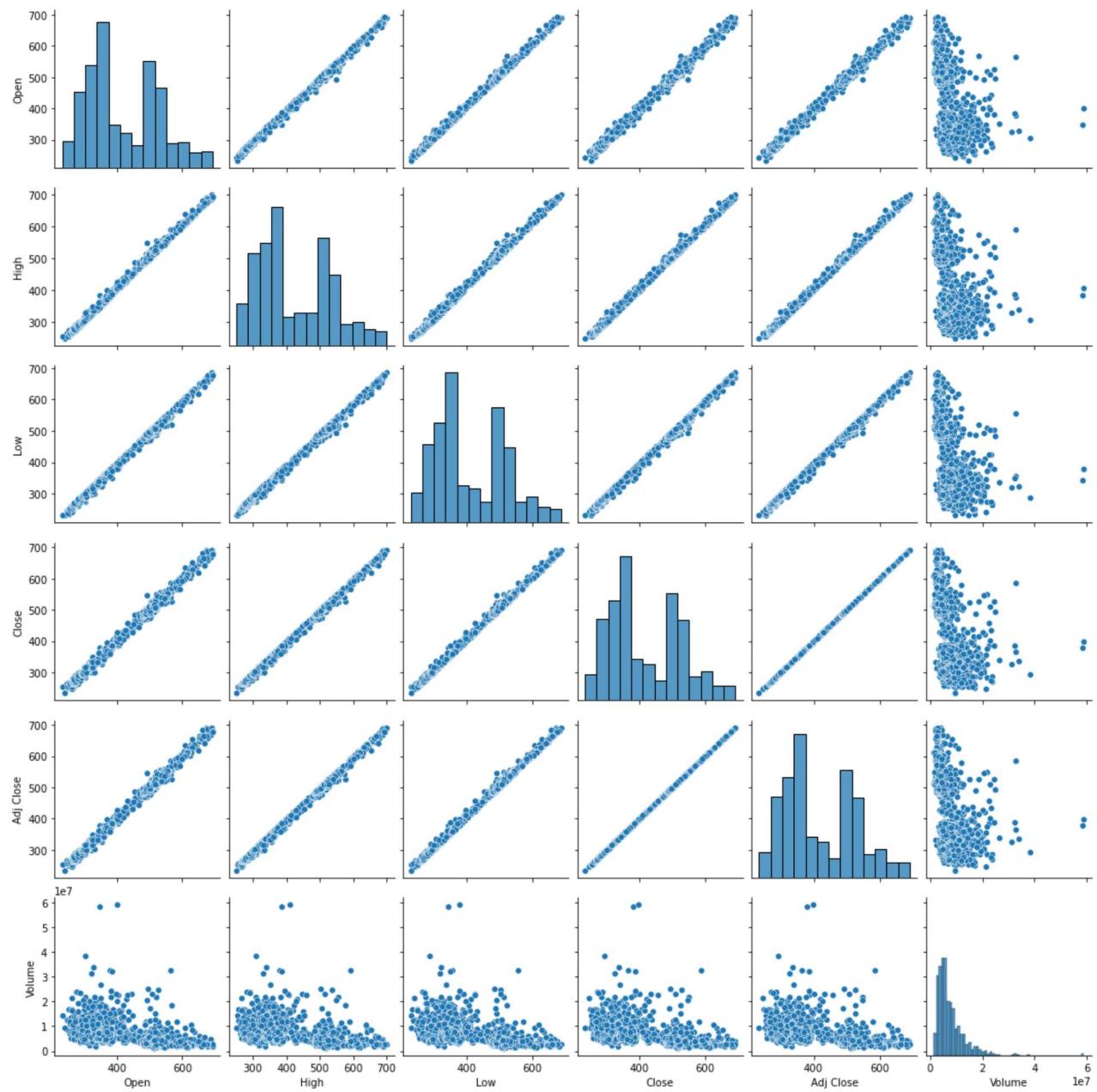
```
b1=sns.boxplot(x="Open",data=df)
```



PAIRPLOT

In [145...]

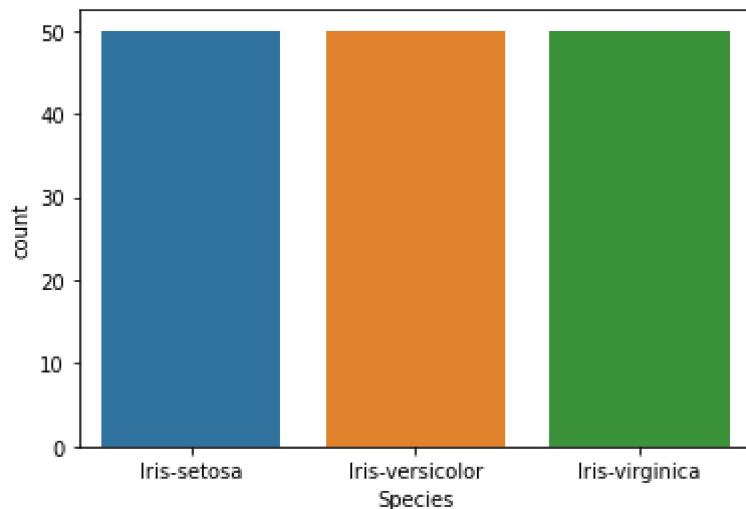
```
P=sns.pairplot(df)
```



countplot

```
In [148]: sns.countplot(data=df2,x="Species")
```

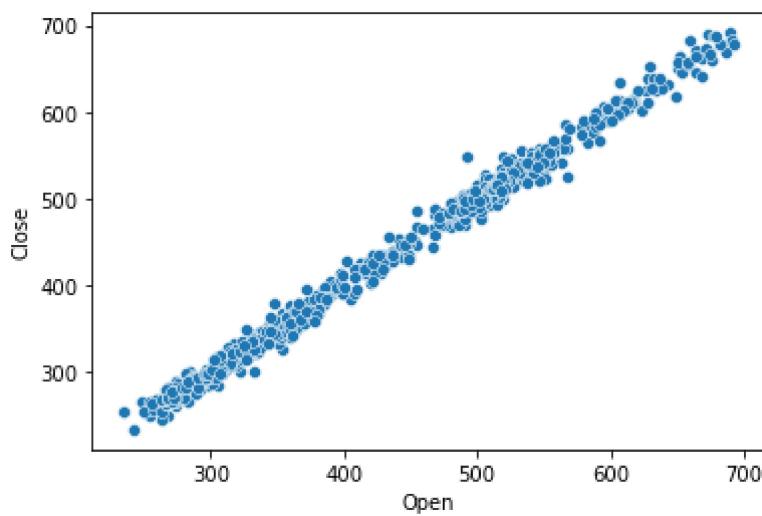
```
Out[148]: <AxesSubplot:xlabel='Species', ylabel='count'>
```



Scatter plot

```
In [149...]: sns.scatterplot(data=df, x="Open", y="Close")
```

```
Out[149...]: <AxesSubplot:xlabel='Open', ylabel='Close'>
```



DECISION TREE

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.

```
In [105...]: from sklearn.model_selection import train_test_split
```

```
In [106...]: X=df.iloc[:,2:6]
X
```

Out[106...]

	High	Low	Close	Adj Close
0	267.899994	250.029999	254.259995	254.259995
1	266.700012	245.000000	265.720001	265.720001
2	272.450012	264.329987	264.559998	264.559998
3	267.619995	250.000000	250.100006	250.100006
4	255.800003	236.110001	249.470001	249.470001
...
1004	427.700012	398.200012	427.140015	427.140015
1005	458.480011	425.540009	457.130005	457.130005
1006	451.980011	426.480011	429.480011	429.480011
1007	429.260010	404.279999	405.600006	405.600006
1008	412.769989	396.640015	410.170013	410.170013

1009 rows × 4 columns

In [107...]

y=df["Volume"]

In [108...]

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20)

In [109...]

from sklearn.tree import DecisionTreeClassifier

In [110...]

classifier = DecisionTreeClassifier()

In [111...]

classifier.fit(x_train, y_train)

Out[111...]

DecisionTreeClassifier()

In [112...]

y_pred=classifier.predict(x_test)
y_pred

Out[112...]

array([7873900, 3075700, 1595500, 9717900, 2579400, 5416600,
 8376600, 18871200, 11932100, 5657100, 2681200, 4431100,
 24499000, 7359100, 3124000, 3457000, 13508900, 20373700,
 4781300, 23177600, 2705200, 2482600, 6135800, 12655800,
 31287100, 5809300, 4353500, 10499600, 16878700, 7161500,
 10655200, 23429900, 7567500, 5752200, 9007500, 5164500,
 5990400, 2820200, 4621500, 4658900, 7643500, 8692500,
 4386400, 8629600, 8544000, 3530900, 5621900, 8968000,
 13478600, 2582000, 4584700, 8966800, 5969000, 5280300,
 23177600, 7970900, 9925200, 5263900, 6259500, 4705600,
 9222600, 1287200, 3820500, 5197600, 13639500, 12743300,
 4942200, 2504700, 7207300, 14919700, 5099600, 7643500,
 7817400, 7643500, 4584700, 5934500, 4444400, 4229000,

```
4229000, 8147200, 7817400, 14783200, 11340100, 3467800,
3042200, 7666500, 11444800, 5132500, 4308200, 9925200,
3686400, 10066700, 7088200, 6657300, 4096900, 5424500,
9222600, 4942200, 1144000, 5460500, 5633400, 4580400,
5663100, 3732200, 14919700, 9925200, 31287100, 2012900,
2376200, 3238900, 3124000, 7448400, 9071100, 4502700,
3512100, 6444100, 2712500, 5905900, 5280300, 5137300,
5948400, 8376600, 7405500, 4096900, 6541900, 4336300,
31287100, 1367800, 9234500, 1621100, 2320200, 2613700,
6135800, 10655200, 2350500, 2526200, 2032800, 4992600,
7207300, 2722500, 24324700, 5899500, 12914000, 4942200,
5669900, 4717100, 23177600, 10992800, 4388500, 10283000,
4393100, 7643500, 14758600, 8629600, 10759700, 23832800,
16853600, 9634700, 4820300, 3355900, 5642900, 9306700,
7561400, 5254900, 5113100, 2712500, 3322900, 5099600,
6954100, 3322900, 21698800, 8278000, 23429900, 3041400,
5230000, 9061100, 7405500, 3355900, 13893200, 21605600,
5230000, 3666200, 8353200, 5748400, 21084800, 5263900,
10992800, 9306700, 5158000, 3885800, 4089800, 8638700,
9222600, 3238900, 9158700, 8534900, 31287100, 5346200,
18222800, 14919700, 9925200, 7666500], dtype=int64)
```

In [113...]

```
confusion_matrix(y_test,y_pred)
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
1144000	0.00	0.00	0.00	0.0
1287200	0.00	0.00	0.00	0.0
1367800	0.00	0.00	0.00	0.0
1595500	0.00	0.00	0.00	0.0
1621100	0.00	0.00	0.00	0.0
1673900	0.00	0.00	0.00	1.0
1832000	0.00	0.00	0.00	1.0
1876300	0.00	0.00	0.00	1.0
1960500	0.00	0.00	0.00	1.0
1995900	0.00	0.00	0.00	1.0
2012900	0.00	0.00	0.00	0.0
2019300	0.00	0.00	0.00	1.0
2032800	0.00	0.00	0.00	0.0
2039400	0.00	0.00	0.00	1.0
2065600	0.00	0.00	0.00	1.0
2126200	0.00	0.00	0.00	1.0
2320200	0.00	0.00	0.00	0.0
2335700	0.00	0.00	0.00	1.0
2350500	0.00	0.00	0.00	0.0
2376200	0.00	0.00	0.00	0.0
2411700	0.00	0.00	0.00	1.0
2482600	0.00	0.00	0.00	0.0
2504700	0.00	0.00	0.00	0.0
2526200	0.00	0.00	0.00	0.0
2534900	0.00	0.00	0.00	1.0
2556700	0.00	0.00	0.00	1.0
2579400	0.00	0.00	0.00	0.0
2582000	0.00	0.00	0.00	0.0
2613700	0.00	0.00	0.00	0.0
2681200	0.00	0.00	0.00	0.0
2705200	0.00	0.00	0.00	0.0
2712500	0.00	0.00	0.00	0.0

2722500	0.00	0.00	0.00	0.0
2732800	0.00	0.00	0.00	1.0
2777200	0.00	0.00	0.00	1.0
2820200	0.00	0.00	0.00	0.0
2826100	0.00	0.00	0.00	1.0
2887500	0.00	0.00	0.00	1.0
2894000	0.00	0.00	0.00	1.0
3041400	0.00	0.00	0.00	0.0
3042200	0.00	0.00	0.00	0.0
3075700	0.00	0.00	0.00	0.0
3077800	0.00	0.00	0.00	1.0
3124000	0.00	0.00	0.00	0.0
3139100	0.00	0.00	0.00	1.0
3143200	0.00	0.00	0.00	1.0
3193400	0.00	0.00	0.00	1.0
3238900	0.00	0.00	0.00	0.0
3322900	0.00	0.00	0.00	0.0
3330000	0.00	0.00	0.00	1.0
3355900	0.00	0.00	0.00	0.0
3394900	0.00	0.00	0.00	1.0
3442100	0.00	0.00	0.00	1.0
3457000	0.00	0.00	0.00	0.0
3467800	0.00	0.00	0.00	0.0
3512100	0.00	0.00	0.00	0.0
3530900	0.00	0.00	0.00	0.0
3666200	0.00	0.00	0.00	0.0
3669700	0.00	0.00	0.00	1.0
3671700	0.00	0.00	0.00	1.0
3686400	0.00	0.00	0.00	0.0
3710000	0.00	0.00	0.00	1.0
3732200	0.00	0.00	0.00	0.0
3755600	0.00	0.00	0.00	1.0
3783700	0.00	0.00	0.00	1.0
3806200	0.00	0.00	0.00	1.0
3820500	0.00	0.00	0.00	0.0
3885800	0.00	0.00	0.00	0.0
3887400	0.00	0.00	0.00	1.0
3888600	0.00	0.00	0.00	1.0
3938600	0.00	0.00	0.00	1.0
3944800	0.00	0.00	0.00	1.0
3981800	0.00	0.00	0.00	1.0
4008000	0.00	0.00	0.00	1.0
4021800	0.00	0.00	0.00	1.0
4089800	0.00	0.00	0.00	0.0
4090800	0.00	0.00	0.00	1.0
4096900	0.00	0.00	0.00	0.0
4148700	0.00	0.00	0.00	1.0
4177400	0.00	0.00	0.00	1.0
4229000	0.00	0.00	0.00	0.0
4308200	0.00	0.00	0.00	0.0
4309800	0.00	0.00	0.00	1.0
4311500	0.00	0.00	0.00	1.0
4316000	0.00	0.00	0.00	1.0
4328100	0.00	0.00	0.00	1.0
4336300	0.00	0.00	0.00	0.0
4353500	0.00	0.00	0.00	0.0
4386400	0.00	0.00	0.00	0.0
4386900	0.00	0.00	0.00	1.0
4388500	0.00	0.00	0.00	0.0
4393100	0.00	0.00	0.00	0.0

4417500	0.00	0.00	0.00	1.0
4431100	0.00	0.00	0.00	0.0
4444400	0.00	0.00	0.00	0.0
4446400	0.00	0.00	0.00	1.0
4485800	0.00	0.00	0.00	1.0
4486100	0.00	0.00	0.00	1.0
4502700	0.00	0.00	0.00	0.0
4580400	0.00	0.00	0.00	0.0
4584700	0.00	0.00	0.00	0.0
4589700	0.00	0.00	0.00	1.0
4592500	0.00	0.00	0.00	1.0
4621500	0.00	0.00	0.00	0.0
4653800	0.00	0.00	0.00	1.0
4658900	0.00	0.00	0.00	0.0
4659500	0.00	0.00	0.00	1.0
4669100	0.00	0.00	0.00	1.0
4705600	0.00	0.00	0.00	0.0
4714500	0.00	0.00	0.00	1.0
4717100	0.00	0.00	0.00	0.0
4762200	0.00	0.00	0.00	1.0
4773500	0.00	0.00	0.00	1.0
4781300	0.00	0.00	0.00	0.0
4820300	0.00	0.00	0.00	0.0
4826200	0.00	0.00	0.00	1.0
4839100	0.00	0.00	0.00	1.0
4942200	0.00	0.00	0.00	0.0
4974200	0.00	0.00	0.00	1.0
4992600	0.00	0.00	0.00	0.0
5010900	0.00	0.00	0.00	1.0
5023400	0.00	0.00	0.00	1.0
5052700	0.00	0.00	0.00	1.0
5063700	0.00	0.00	0.00	1.0
5099600	0.00	0.00	0.00	0.0
5111800	0.00	0.00	0.00	1.0
5113100	0.00	0.00	0.00	0.0
5127800	0.00	0.00	0.00	1.0
5130300	0.00	0.00	0.00	1.0
5131600	0.00	0.00	0.00	1.0
5132500	0.00	0.00	0.00	0.0
5137300	0.00	0.00	0.00	0.0
5158000	0.00	0.00	0.00	0.0
5164500	0.00	0.00	0.00	0.0
5197600	0.00	0.00	0.00	0.0
5225700	0.00	0.00	0.00	1.0
5230000	0.00	0.00	0.00	0.0
5254900	0.00	0.00	0.00	0.0
5263900	0.00	0.00	0.00	0.0
5277300	0.00	0.00	0.00	1.0
5280300	0.00	0.00	0.00	0.0
5289400	0.00	0.00	0.00	1.0
5302300	0.00	0.00	0.00	1.0
5328100	0.00	0.00	0.00	1.0
5346200	0.00	0.00	0.00	0.0
5349100	0.00	0.00	0.00	1.0
5386500	0.00	0.00	0.00	1.0
5394700	0.00	0.00	0.00	1.0
5398200	0.00	0.00	0.00	1.0
5416600	0.00	0.00	0.00	0.0
5424500	0.00	0.00	0.00	0.0
5439200	0.00	0.00	0.00	1.0

5460500	0.00	0.00	0.00	0.0
5486000	0.00	0.00	0.00	1.0
5507900	0.00	0.00	0.00	1.0
5572700	0.00	0.00	0.00	1.0
5583500	0.00	0.00	0.00	1.0
5621900	0.00	0.00	0.00	0.0
5629900	0.00	0.00	0.00	1.0
5633400	0.00	0.00	0.00	0.0
5642900	0.00	0.00	0.00	0.0
5657100	0.00	0.00	0.00	0.0
5658900	0.00	0.00	0.00	1.0
5663100	0.00	0.00	0.00	0.0
5669900	0.00	0.00	0.00	0.0
5685500	0.00	0.00	0.00	1.0
5697100	0.00	0.00	0.00	1.0
5727700	0.00	0.00	0.00	1.0
5748400	0.00	0.00	0.00	0.0
5750400	0.00	0.00	0.00	1.0
5752200	0.00	0.00	0.00	0.0
5793100	0.00	0.00	0.00	1.0
5809300	0.00	0.00	0.00	0.0
5851500	0.00	0.00	0.00	1.0
5873100	0.00	0.00	0.00	1.0
5886900	0.00	0.00	0.00	1.0
5895800	0.00	0.00	0.00	1.0
5899500	0.00	0.00	0.00	0.0
5905900	0.00	0.00	0.00	0.0
5934500	0.00	0.00	0.00	0.0
5948400	0.00	0.00	0.00	0.0
5969000	0.00	0.00	0.00	0.0
5986700	0.00	0.00	0.00	1.0
5990400	0.00	0.00	0.00	0.0
5992700	0.00	0.00	0.00	1.0
6036000	0.00	0.00	0.00	1.0
6066500	0.00	0.00	0.00	1.0
6135800	0.00	0.00	0.00	0.0
6209300	0.00	0.00	0.00	1.0
6259500	0.00	0.00	0.00	0.0
6276400	0.00	0.00	0.00	1.0
6351500	0.00	0.00	0.00	1.0
6444100	0.00	0.00	0.00	0.0
6481200	0.00	0.00	0.00	1.0
6541900	0.00	0.00	0.00	0.0
6657300	0.00	0.00	0.00	0.0
6660700	0.00	0.00	0.00	1.0
6672500	0.00	0.00	0.00	1.0
6768100	0.00	0.00	0.00	1.0
6794400	0.00	0.00	0.00	1.0
6821600	0.00	0.00	0.00	1.0
6848100	0.00	0.00	0.00	1.0
6888900	0.00	0.00	0.00	1.0
6954100	0.00	0.00	0.00	0.0
7088200	0.00	0.00	0.00	0.0
7114900	0.00	0.00	0.00	1.0
7161500	0.00	0.00	0.00	0.0
7207300	0.00	0.00	0.00	0.0
7244000	0.00	0.00	0.00	1.0
7333700	0.00	0.00	0.00	1.0
7359100	0.00	0.00	0.00	0.0
7396600	0.00	0.00	0.00	1.0

7405500	0.00	0.00	0.00	0.0
7448400	0.00	0.00	0.00	0.0
7488400	0.00	0.00	0.00	1.0
7561400	0.00	0.00	0.00	0.0
7567500	0.00	0.00	0.00	0.0
7643500	0.00	0.00	0.00	0.0
7666500	0.00	0.00	0.00	0.0
7709500	0.00	0.00	0.00	1.0
7711300	0.00	0.00	0.00	1.0
7810300	0.00	0.00	0.00	1.0
7817400	0.00	0.00	0.00	0.0
7849400	0.00	0.00	0.00	1.0
7864500	0.00	0.00	0.00	1.0
7873900	0.00	0.00	0.00	0.0
7970900	0.00	0.00	0.00	0.0
8026700	0.00	0.00	0.00	1.0
8147200	0.00	0.00	0.00	0.0
8153700	0.00	0.00	0.00	1.0
8270100	0.00	0.00	0.00	1.0
8278000	0.00	0.00	0.00	0.0
8353200	0.00	0.00	0.00	0.0
8376600	0.00	0.00	0.00	0.0
8534900	0.00	0.00	0.00	0.0
8544000	0.00	0.00	0.00	0.0
8616600	0.00	0.00	0.00	1.0
8629600	0.00	0.00	0.00	0.0
8638700	0.00	0.00	0.00	0.0
8670300	0.00	0.00	0.00	1.0
8692500	0.00	0.00	0.00	0.0
8767200	0.00	0.00	0.00	1.0
8930700	0.00	0.00	0.00	1.0
8949500	0.00	0.00	0.00	1.0
8966800	0.00	0.00	0.00	0.0
8968000	0.00	0.00	0.00	0.0
9007500	0.00	0.00	0.00	0.0
9061100	0.00	0.00	0.00	0.0
9062900	0.00	0.00	0.00	1.0
9071100	0.00	0.00	0.00	0.0
9158700	0.00	0.00	0.00	0.0
9171100	0.00	0.00	0.00	1.0
9222600	0.00	0.00	0.00	0.0
9234500	0.00	0.00	0.00	0.0
9252500	0.00	0.00	0.00	1.0
9306700	0.00	0.00	0.00	0.0
9322400	0.00	0.00	0.00	1.0
9366500	0.00	0.00	0.00	1.0
9458200	0.00	0.00	0.00	1.0
9534300	0.00	0.00	0.00	1.0
9572700	0.00	0.00	0.00	1.0
9634700	0.00	0.00	0.00	0.0
9717900	0.00	0.00	0.00	0.0
9905200	0.00	0.00	0.00	1.0
9925200	0.00	0.00	0.00	0.0
10066700	0.00	0.00	0.00	0.0
10089800	0.00	0.00	0.00	1.0
10101200	0.00	0.00	0.00	1.0
10249400	0.00	0.00	0.00	1.0
10283000	0.00	0.00	0.00	0.0
10407900	0.00	0.00	0.00	1.0
10414000	0.00	0.00	0.00	1.0

10427100	0.00	0.00	0.00	1.0
10446300	0.00	0.00	0.00	1.0
10499600	0.00	0.00	0.00	0.0
10559900	0.00	0.00	0.00	1.0
10616000	0.00	0.00	0.00	1.0
10655200	0.00	0.00	0.00	0.0
10759700	0.00	0.00	0.00	0.0
10924800	0.00	0.00	0.00	1.0
10972000	0.00	0.00	0.00	1.0
10981000	0.00	0.00	0.00	1.0
10992800	0.00	0.00	0.00	0.0
11023000	0.00	0.00	0.00	1.0
11131600	0.00	0.00	0.00	1.0
11168600	0.00	0.00	0.00	1.0
11207400	0.00	0.00	0.00	1.0
11257600	0.00	0.00	0.00	1.0
11340100	0.00	0.00	0.00	0.0
11444800	0.00	0.00	0.00	0.0
11456700	0.00	0.00	0.00	1.0
11505200	0.00	0.00	0.00	1.0
11581000	0.00	0.00	0.00	1.0
11784500	0.00	0.00	0.00	1.0
11802100	0.00	0.00	0.00	1.0
11876800	0.00	0.00	0.00	1.0
11896100	0.00	0.00	0.00	1.0
11932100	0.00	0.00	0.00	0.0
11961800	0.00	0.00	0.00	1.0
11980500	0.00	0.00	0.00	1.0
11988300	0.00	0.00	0.00	1.0
12046600	0.00	0.00	0.00	1.0
12071600	0.00	0.00	0.00	1.0
12315800	0.00	0.00	0.00	1.0
12320200	0.00	0.00	0.00	1.0
12595800	0.00	0.00	0.00	1.0
12599200	0.00	0.00	0.00	1.0
12655800	0.00	0.00	0.00	0.0
12659000	0.00	0.00	0.00	1.0
12743300	0.00	0.00	0.00	0.0
12800600	0.00	0.00	0.00	1.0
12914000	0.00	0.00	0.00	0.0
12917200	0.00	0.00	0.00	1.0
13050200	0.00	0.00	0.00	1.0
13346900	0.00	0.00	0.00	1.0
13404600	0.00	0.00	0.00	1.0
13449400	0.00	0.00	0.00	1.0
13478600	0.00	0.00	0.00	0.0
13508900	0.00	0.00	0.00	0.0
13588100	0.00	0.00	0.00	1.0
13639500	0.00	0.00	0.00	0.0
13799700	0.00	0.00	0.00	1.0
13893200	0.00	0.00	0.00	0.0
14117400	0.00	0.00	0.00	1.0
14350300	0.00	0.00	0.00	1.0
14500200	0.00	0.00	0.00	1.0
14758600	0.00	0.00	0.00	0.0
14783200	0.00	0.00	0.00	0.0
14907300	0.00	0.00	0.00	1.0
14919700	0.00	0.00	0.00	0.0
15431500	0.00	0.00	0.00	1.0
16128700	0.00	0.00	0.00	1.0

16302500	0.00	0.00	0.00	1.0
16338200	0.00	0.00	0.00	1.0
16853600	0.00	0.00	0.00	0.0
16878700	0.00	0.00	0.00	0.0
16906900	0.00	0.00	0.00	1.0
17405700	0.00	0.00	0.00	1.0
17718000	0.00	0.00	0.00	1.0
17939700	0.00	0.00	0.00	1.0
18222800	0.00	0.00	0.00	0.0
18461000	0.00	0.00	0.00	1.0
18871200	0.00	0.00	0.00	0.0
19039300	0.00	0.00	0.00	1.0
19330100	0.00	0.00	0.00	1.0
19616000	0.00	0.00	0.00	1.0
20047500	0.00	0.00	0.00	1.0
20373700	0.00	0.00	0.00	0.0
21084800	0.00	0.00	0.00	0.0
21397600	0.00	0.00	0.00	1.0
21605600	0.00	0.00	0.00	0.0
21698800	0.00	0.00	0.00	0.0
21746300	0.00	0.00	0.00	1.0
22490900	0.00	0.00	0.00	1.0
22542300	0.00	0.00	0.00	1.0
22897400	0.00	0.00	0.00	1.0
23177600	0.00	0.00	0.00	0.0
23429900	0.00	0.00	0.00	0.0
23685700	0.00	0.00	0.00	1.0
23832800	0.00	0.00	0.00	0.0
24324700	0.00	0.00	0.00	0.0
24499000	0.00	0.00	0.00	0.0
26621000	0.00	0.00	0.00	1.0
31287100	0.00	0.00	0.00	0.0
accuracy			0.00	202.0
macro avg	0.00	0.00	0.00	202.0
weighted avg	0.00	0.00	0.00	202.0

```
C:\Users\jagan\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: Unde
finedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
C:\Users\jagan\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: Unde
finedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels wi
th no true samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
C:\Users\jagan\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: Unde
finedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
C:\Users\jagan\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: Unde
finedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels wi
th no true samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
C:\Users\jagan\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: Unde
finedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
C:\Users\jagan\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: Unde
finedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels wi
th no true samples. Use `zero_division` parameter to control this behavior.
```

```
th no true samples. Use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```

In [23]:

```
from sklearn.metrics import classification_report,confusion_matrix
```

In [24]:

```
confusion_matrix(y_test,y_pred)
```

Out[24]:

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

In [25]:

```
from pprint import pprint
pprint(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support\n'
'\n'				
1287200	0.00	0.00	0.00	0.0\n'
1367800	0.00	0.00	0.00	0.0\n'
1685700	0.00	0.00	0.00	0.0\n'
1867300	0.00	0.00	0.00	1.0\n'
1906600	0.00	0.00	0.00	1.0\n'
1919400	0.00	0.00	0.00	1.0\n'
1960500	0.00	0.00	0.00	0.0\n'
1975500	0.00	0.00	0.00	0.0\n'
2019300	0.00	0.00	0.00	1.0\n'
2061500	0.00	0.00	0.00	1.0\n'
2077400	0.00	0.00	0.00	1.0\n'
2151300	0.00	0.00	0.00	1.0\n'
2309800	0.00	0.00	0.00	1.0\n'
2314600	0.00	0.00	0.00	0.0\n'
2319400	0.00	0.00	0.00	0.0\n'
2334900	0.00	0.00	0.00	0.0\n'
2411700	0.00	0.00	0.00	0.0\n'
2412600	0.00	0.00	0.00	1.0\n'
2415600	0.00	0.00	0.00	1.0\n'
2418100	0.00	0.00	0.00	1.0\n'
2423500	0.00	0.00	0.00	0.0\n'
2434800	0.00	0.00	0.00	0.0\n'
2456200	0.00	0.00	0.00	0.0\n'
2465300	0.00	0.00	0.00	0.0\n'
2579400	0.00	0.00	0.00	0.0\n'
2613700	0.00	0.00	0.00	0.0\n'
2624100	0.00	0.00	0.00	0.0\n'
2681200	0.00	0.00	0.00	1.0\n'
2712500	0.00	0.00	0.00	1.0\n'
2720300	0.00	0.00	0.00	0.0\n'
2722500	0.00	0.00	0.00	1.0\n'
2730600	0.00	0.00	0.00	1.0\n'
2732800	0.00	0.00	0.00	1.0\n'
2751600	0.00	0.00	0.00	1.0\n'
2759000	0.00	0.00	0.00	1.0\n'
2773400	0.00	0.00	0.00	1.0\n'
2775100	0.00	0.00	0.00	1.0\n'

2791900	0.00	0.00	0.00	0.0\n'
2805400	0.00	0.00	0.00	0.0\n'
2841500	0.00	0.00	0.00	0.0\n'
2866200	0.00	0.00	0.00	0.0\n'
2887500	0.00	0.00	0.00	0.0\n'
2891900	0.00	0.00	0.00	0.0\n'
2904800	0.00	0.00	0.00	0.0\n'
2944100	0.00	0.00	0.00	1.0\n'
3002700	0.00	0.00	0.00	1.0\n'
3008900	0.00	0.00	0.00	1.0\n'
3018200	0.00	0.00	0.00	1.0\n'
3062900	0.00	0.00	0.00	0.0\n'
3075700	0.00	0.00	0.00	0.0\n'
3104100	0.00	0.00	0.00	0.0\n'
3129400	0.00	0.00	0.00	0.0\n'
3139100	0.00	0.00	0.00	1.0\n'
3227300	0.00	0.00	0.00	0.0\n'
3269000	0.00	0.00	0.00	1.0\n'
3271100	0.00	0.00	0.00	0.0\n'
3322900	0.00	0.00	0.00	0.0\n'
3349900	0.00	0.00	0.00	0.0\n'
3355900	0.00	0.00	0.00	0.0\n'
3358400	0.00	0.00	0.00	1.0\n'
3361200	0.00	0.00	0.00	0.0\n'
3381700	0.00	0.00	0.00	0.0\n'
3482300	0.00	0.00	0.00	0.0\n'
3484300	0.00	0.00	0.00	1.0\n'
3505500	0.00	0.00	0.00	1.0\n'
3550500	0.00	0.00	0.00	1.0\n'
3556900	0.00	0.00	0.00	1.0\n'
3589900	0.00	0.00	0.00	1.0\n'
3625000	0.00	0.00	0.00	0.0\n'
3629200	0.00	0.00	0.00	0.0\n'
3671700	0.00	0.00	0.00	0.0\n'
3682800	0.00	0.00	0.00	0.0\n'
3686400	0.00	0.00	0.00	0.0\n'
3703500	0.00	0.00	0.00	0.0\n'
3726400	0.00	0.00	0.00	0.0\n'
3743700	0.00	0.00	0.00	1.0\n'
3755600	0.00	0.00	0.00	0.0\n'
3769400	0.00	0.00	0.00	0.0\n'
3787400	0.00	0.00	0.00	0.0\n'
3817500	0.00	0.00	0.00	0.0\n'
3831000	0.00	0.00	0.00	1.0\n'
3870100	0.00	0.00	0.00	0.0\n'
3885800	0.00	0.00	0.00	1.0\n'
3888600	0.00	0.00	0.00	1.0\n'
3894000	0.00	0.00	0.00	0.0\n'
3948000	0.00	0.00	0.00	1.0\n'
3991300	0.00	0.00	0.00	1.0\n'
4008000	0.00	0.00	0.00	0.0\n'
4021800	0.00	0.00	0.00	0.0\n'
4062400	0.00	0.00	0.00	0.0\n'
4079400	0.00	0.00	0.00	1.0\n'
4134500	0.00	0.00	0.00	0.0\n'
4148700	0.00	0.00	0.00	0.0\n'
4177400	0.00	0.00	0.00	1.0\n'
4198500	0.00	0.00	0.00	0.0\n'
4291000	0.00	0.00	0.00	1.0\n'
4311500	0.00	0.00	0.00	1.0\n'

4325300	0.00	0.00	0.00	1.0\n'
4353500	0.00	0.00	0.00	1.0\n'
4386400	0.00	0.00	0.00	1.0\n'
4388500	0.00	0.00	0.00	1.0\n'
4388800	0.00	0.00	0.00	1.0\n'
4401000	0.00	0.00	0.00	1.0\n'
4408200	0.00	0.00	0.00	1.0\n'
4413200	0.00	0.00	0.00	0.0\n'
4432400	0.00	0.00	0.00	1.0\n'
4457800	0.00	0.00	0.00	0.0\n'
4467900	0.00	0.00	0.00	0.0\n'
4475100	0.00	0.00	0.00	1.0\n'
4502700	0.00	0.00	0.00	1.0\n'
4580400	0.00	0.00	0.00	1.0\n'
4592500	0.00	0.00	0.00	0.0\n'
4621500	0.00	0.00	0.00	0.0\n'
4659500	0.00	0.00	0.00	0.0\n'
4703200	0.00	0.00	0.00	0.0\n'
4717100	0.00	0.00	0.00	0.0\n'
4725400	0.00	0.00	0.00	0.0\n'
4756400	0.00	0.00	0.00	0.0\n'
4766600	0.00	0.00	0.00	0.0\n'
4777100	0.00	0.00	0.00	1.0\n'
4810500	0.00	0.00	0.00	1.0\n'
4820300	0.00	0.00	0.00	1.0\n'
4826200	0.00	0.00	0.00	0.0\n'
4830200	0.00	0.00	0.00	0.0\n'
4839100	0.00	0.00	0.00	1.0\n'
4865000	0.00	0.00	0.00	1.0\n'
4941400	0.00	0.00	0.00	0.0\n'
4944600	0.00	0.00	0.00	1.0\n'
4974200	0.00	0.00	0.00	0.0\n'
4992600	0.00	0.00	0.00	1.0\n'
4995900	0.00	0.00	0.00	0.0\n'
5049300	0.00	0.00	0.00	1.0\n'
5063700	0.00	0.00	0.00	1.0\n'
5111800	0.00	0.00	0.00	1.0\n'
5127800	0.00	0.00	0.00	1.0\n'
5131600	0.00	0.00	0.00	0.0\n'
5197600	0.00	0.00	0.00	1.0\n'
5245100	0.00	0.00	0.00	0.0\n'
5271400	0.00	0.00	0.00	0.0\n'
5277400	0.00	0.00	0.00	0.0\n'
5294100	0.00	0.00	0.00	0.0\n'
5328100	0.00	0.00	0.00	1.0\n'
5337000	0.00	0.00	0.00	1.0\n'
5346200	0.00	0.00	0.00	0.0\n'
5349100	0.00	0.00	0.00	0.0\n'
5358200	0.00	0.00	0.00	1.0\n'
5362400	0.00	0.00	0.00	0.0\n'
5368900	0.00	0.00	0.00	1.0\n'
5379300	0.00	0.00	0.00	1.0\n'
5386500	0.00	0.00	0.00	0.0\n'
5394700	0.00	0.00	0.00	1.0\n'
5416600	0.00	0.00	0.00	1.0\n'
5422300	0.00	0.00	0.00	1.0\n'
5487300	0.00	0.00	0.00	1.0\n'
5566200	0.00	0.00	0.00	0.0\n'
5572700	0.00	0.00	0.00	1.0\n'
5583500	0.00	0.00	0.00	1.0\n'

5589800	0.00	0.00	0.00	0.0\n'
5607300	0.00	0.00	0.00	0.0\n'
5608000	0.00	0.00	0.00	0.0\n'
5621900	0.00	0.00	0.00	1.0\n'
5633400	0.00	0.00	0.00	1.0\n'
5657100	0.00	0.00	0.00	1.0\n'
5667200	0.00	0.00	0.00	0.0\n'
5669900	0.00	0.00	0.00	0.0\n'
5697100	0.00	0.00	0.00	1.0\n'
5711800	0.00	0.00	0.00	1.0\n'
5752200	0.00	0.00	0.00	1.0\n'
5772800	0.00	0.00	0.00	1.0\n'
5798600	0.00	0.00	0.00	0.0\n'
5805200	0.00	0.00	0.00	1.0\n'
5821400	0.00	0.00	0.00	1.0\n'
5840500	0.00	0.00	0.00	0.0\n'
5851500	0.00	0.00	0.00	0.0\n'
5878800	0.00	0.00	0.00	1.0\n'
5882600	0.00	0.00	0.00	0.0\n'
5895800	0.00	0.00	0.00	1.0\n'
5899500	0.00	0.00	0.00	0.0\n'
5910300	0.00	0.00	0.00	1.0\n'
5928500	0.00	0.00	0.00	1.0\n'
5934500	0.00	0.00	0.00	1.0\n'
5948400	0.00	0.00	0.00	1.0\n'
5964400	0.00	0.00	0.00	1.0\n'
5969000	0.00	0.00	0.00	1.0\n'
5970100	0.00	0.00	0.00	1.0\n'
6029300	0.00	0.00	0.00	1.0\n'
6036600	0.00	0.00	0.00	1.0\n'
6135800	0.00	0.00	0.00	0.0\n'
6179900	0.00	0.00	0.00	1.0\n'
6186000	0.00	0.00	0.00	1.0\n'
6186100	0.00	0.00	0.00	1.0\n'
6209300	0.00	0.00	0.00	1.0\n'
6221000	0.00	0.00	0.00	1.0\n'
6227900	0.00	0.00	0.00	0.0\n'
6259500	0.00	0.00	0.00	0.0\n'
6272100	0.00	0.00	0.00	1.0\n'
6276400	0.00	0.00	0.00	0.0\n'
6333800	0.00	0.00	0.00	1.0\n'
6340100	0.00	0.00	0.00	0.0\n'
6347400	0.00	0.00	0.00	0.0\n'
6351500	0.00	0.00	0.00	0.0\n'
6437000	0.00	0.00	0.00	1.0\n'
6461100	0.00	0.00	0.00	1.0\n'
6481200	0.00	0.00	0.00	1.0\n'
6525600	0.00	0.00	0.00	1.0\n'
6563200	0.00	0.00	0.00	1.0\n'
6572000	0.00	0.00	0.00	0.0\n'
6612600	0.00	0.00	0.00	1.0\n'
6636900	0.00	0.00	0.00	1.0\n'
6657300	0.00	0.00	0.00	0.0\n'
6670600	0.00	0.00	0.00	0.0\n'
6672500	0.00	0.00	0.00	1.0\n'
6717700	0.00	0.00	0.00	1.0\n'
6727200	0.00	0.00	0.00	1.0\n'
6794400	0.00	0.00	0.00	1.0\n'
6821600	0.00	0.00	0.00	0.0\n'
6824800	0.00	0.00	0.00	1.0\n'

6855200	0.00	0.00	0.00	1.0\n'
6908900	0.00	0.00	0.00	1.0\n'
6954100	0.00	0.00	0.00	1.0\n'
6974900	0.00	0.00	0.00	0.0\n'
6993700	0.00	0.00	0.00	1.0\n'
6997900	0.00	0.00	0.00	0.0\n'
7014600	0.00	0.00	0.00	0.0\n'
7046400	0.00	0.00	0.00	0.0\n'
7088200	0.00	0.00	0.00	1.0\n'
7112300	0.00	0.00	0.00	1.0\n'
7114900	0.00	0.00	0.00	0.0\n'
7133500	0.00	0.00	0.00	0.0\n'
7161500	0.00	0.00	0.00	1.0\n'
7229600	0.00	0.00	0.00	1.0\n'
7301800	0.00	0.00	0.00	1.0\n'
7307700	0.00	0.00	0.00	0.0\n'
7329200	0.00	0.00	0.00	0.0\n'
7405900	0.00	0.00	0.00	1.0\n'
7448400	0.00	0.00	0.00	0.0\n'
7488400	0.00	0.00	0.00	1.0\n'
7541400	0.00	0.00	0.00	1.0\n'
7550800	0.00	0.00	0.00	1.0\n'
7569300	0.00	0.00	0.00	1.0\n'
7666500	0.00	0.00	0.00	0.0\n'
7685600	0.00	0.00	0.00	0.0\n'
7711300	0.00	0.00	0.00	1.0\n'
7746200	0.00	0.00	0.00	0.0\n'
7773500	0.00	0.00	0.00	1.0\n'
7782400	0.00	0.00	0.00	1.0\n'
7811100	0.00	0.00	0.00	0.0\n'
7817400	0.00	0.00	0.00	1.0\n'
7849600	0.00	0.00	0.00	0.0\n'
7863100	0.00	0.00	0.00	0.0\n'
7864500	0.00	0.00	0.00	0.0\n'
7891600	0.00	0.00	0.00	0.0\n'
7970900	0.00	0.00	0.00	0.0\n'
8110700	0.00	0.00	0.00	1.0\n'
8198100	0.00	0.00	0.00	0.0\n'
8209500	0.00	0.00	0.00	1.0\n'
8232700	0.00	0.00	0.00	1.0\n'
8278000	0.00	0.00	0.00	0.0\n'
8306100	0.00	0.00	0.00	1.0\n'
8437100	0.00	0.00	0.00	0.0\n'
8438800	0.00	0.00	0.00	0.0\n'
8444500	0.00	0.00	0.00	0.0\n'
8534900	0.00	0.00	0.00	1.0\n'
8629600	0.00	0.00	0.00	1.0\n'
8638700	0.00	0.00	0.00	1.0\n'
8650300	0.00	0.00	0.00	1.0\n'
8661300	0.00	0.00	0.00	1.0\n'
8692500	0.00	0.00	0.00	0.0\n'
8747000	0.00	0.00	0.00	0.0\n'
8842300	0.00	0.00	0.00	1.0\n'
8949500	0.00	0.00	0.00	1.0\n'
8951000	0.00	0.00	0.00	0.0\n'
9061100	0.00	0.00	0.00	1.0\n'
9071100	0.00	0.00	0.00	1.0\n'
9074400	0.00	0.00	0.00	0.0\n'
9099500	0.00	0.00	0.00	1.0\n'
9113700	0.00	0.00	0.00	1.0\n'

9117300	0.00	0.00	0.00	1.0\n'
9158900	0.00	0.00	0.00	0.0\n'
9222600	0.00	0.00	0.00	1.0\n'
9230000	0.00	0.00	0.00	1.0\n'
9234500	0.00	0.00	0.00	1.0\n'
9252500	0.00	0.00	0.00	1.0\n'
9322500	0.00	0.00	0.00	1.0\n'
9366500	0.00	0.00	0.00	1.0\n'
9371100	0.00	0.00	0.00	0.0\n'
9458200	0.00	0.00	0.00	0.0\n'
9499000	0.00	0.00	0.00	1.0\n'
9529900	0.00	0.00	0.00	0.0\n'
9547600	0.00	0.00	0.00	1.0\n'
9605600	0.00	0.00	0.00	1.0\n'
9693100	0.00	0.00	0.00	0.0\n'
9822300	0.00	0.00	0.00	1.0\n'
9843200	0.00	0.00	0.00	0.0\n'
9905200	0.00	0.00	0.00	0.0\n'
9925200	0.00	0.00	0.00	0.0\n'
10049100	0.00	0.00	0.00	0.0\n'
10089800	0.00	0.00	0.00	0.0\n'
10101200	0.00	0.00	0.00	1.0\n'
10249400	0.00	0.00	0.00	0.0\n'
10268600	0.00	0.00	0.00	1.0\n'
10283000	0.00	0.00	0.00	0.0\n'
10429400	0.00	0.00	0.00	1.0\n'
10446300	0.00	0.00	0.00	1.0\n'
10475100	0.00	0.00	0.00	1.0\n'
10480800	0.00	0.00	0.00	0.0\n'
10499600	0.00	0.00	0.00	1.0\n'
10559100	0.00	0.00	0.00	1.0\n'
10559900	0.00	0.00	0.00	1.0\n'
10616000	0.00	0.00	0.00	1.0\n'
10622000	0.00	0.00	0.00	1.0\n'
10655200	0.00	0.00	0.00	0.0\n'
10660500	0.00	0.00	0.00	1.0\n'
10798500	0.00	0.00	0.00	0.0\n'
10847500	0.00	0.00	0.00	0.0\n'
10924800	0.00	0.00	0.00	1.0\n'
10967700	0.00	0.00	0.00	1.0\n'
10992800	0.00	0.00	0.00	1.0\n'
11168600	0.00	0.00	0.00	0.0\n'
11207400	0.00	0.00	0.00	0.0\n'
11221100	0.00	0.00	0.00	1.0\n'
11336400	0.00	0.00	0.00	1.0\n'
11456700	0.00	0.00	0.00	0.0\n'
11860100	0.00	0.00	0.00	0.0\n'
11930600	0.00	0.00	0.00	1.0\n'
11961800	0.00	0.00	0.00	1.0\n'
11966600	0.00	0.00	0.00	0.0\n'
11988300	0.00	0.00	0.00	1.0\n'
12068600	0.00	0.00	0.00	1.0\n'
12071600	0.00	0.00	0.00	1.0\n'
12235200	0.00	0.00	0.00	0.0\n'
12466700	0.00	0.00	0.00	1.0\n'
12655800	0.00	0.00	0.00	1.0\n'
12694900	0.00	0.00	0.00	1.0\n'
12993800	0.00	0.00	0.00	1.0\n'
13328300	0.00	0.00	0.00	1.0\n'
13404600	0.00	0.00	0.00	1.0\n'

```

    13405800    0.00    0.00    0.00    1.0\n'
    13508900    0.00    0.00    0.00    0.0\n'
    13588100    0.00    0.00    0.00    0.0\n'
    13591100    0.00    0.00    0.00    0.0\n'
    14346000    0.00    0.00    0.00    0.0\n'
    14402700    0.00    0.00    0.00    1.0\n'
    14783200    0.00    0.00    0.00    1.0\n'
    14870800    0.00    0.00    0.00    1.0\n'
    14907300    0.00    0.00    0.00    1.0\n'
    15096700    0.00    0.00    0.00    0.0\n'
    15431500    0.00    0.00    0.00    1.0\n'
    16338200    0.00    0.00    0.00    1.0\n'
    16541400    0.00    0.00    0.00    0.0\n'
    16717200    0.00    0.00    0.00    0.0\n'
    16878700    0.00    0.00    0.00    1.0\n'
    16906900    0.00    0.00    0.00    0.0\n'
    17132200    0.00    0.00    0.00    0.0\n'
    17427300    0.00    0.00    0.00    1.0\n'
    18260700    0.00    0.00    0.00    0.0\n'
    18399000    0.00    0.00    0.00    0.0\n'
    18620100    0.00    0.00    0.00    0.0\n'
    18740200    0.00    0.00    0.00    1.0\n'
    19500400    0.00    0.00    0.00    1.0\n'
    20047500    0.00    0.00    0.00    0.0\n'
    20156400    0.00    0.00    0.00    0.0\n'
    20307900    0.00    0.00    0.00    1.0\n'
    21397600    0.00    0.00    0.00    1.0\n'
    21698800    0.00    0.00    0.00    0.0\n'
    22490900    0.00    0.00    0.00    1.0\n'
    22542300    0.00    0.00    0.00    0.0\n'
    23177600    0.00    0.00    0.00    1.0\n'
    24324700    0.00    0.00    0.00    0.0\n'
    24991400    0.00    0.00    0.00    1.0\n'
    26621000    0.00    0.00    0.00    0.0\n'

'\n'
'accuracy                      0.00    202.0\n'
'macro avg                     0.00    0.00    202.0\n'
'weighted avg                  0.00    0.00    202.0\n')

```

C:\Users\jagan\anaconda3\lib\site-packages\sklearn\metrics_classification.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

```
_warn_prf(average, modifier, msg_start, len(result))
```

C:\Users\jagan\anaconda3\lib\site-packages\sklearn\metrics_classification.py:1248: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.

```
_warn_prf(average, modifier, msg_start, len(result))
```

C:\Users\jagan\anaconda3\lib\site-packages\sklearn\metrics_classification.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

```
_warn_prf(average, modifier, msg_start, len(result))
```

C:\Users\jagan\anaconda3\lib\site-packages\sklearn\metrics_classification.py:1248: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.

```
_warn_prf(average, modifier, msg_start, len(result))
```

C:\Users\jagan\anaconda3\lib\site-packages\sklearn\metrics_classification.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

```
_warn_prf(average, modifier, msg_start, len(result))
```

C:\Users\jagan\anaconda3\lib\site-packages\sklearn\metrics_classification.py:1248: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels wi

```
th no true samples. Use `zero_division` parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```

IRIS DATASET

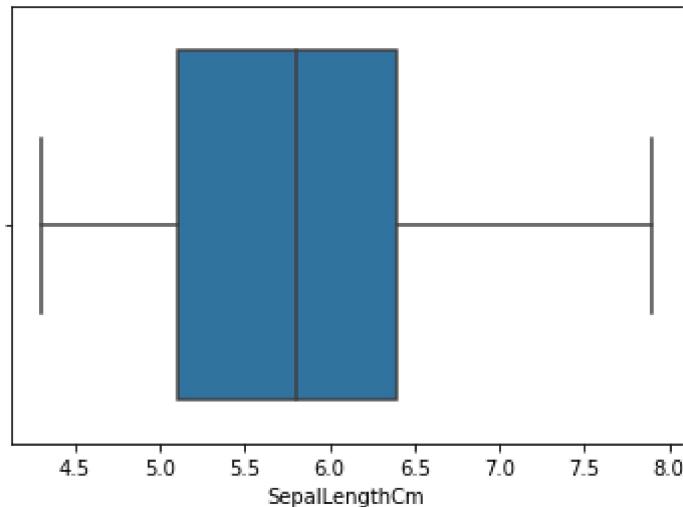
```
In [76]: df2=pd.read_csv(r"C:\Users\jagan\Downloads\Iris.csv")
```

```
In [77]: df2.head()
```

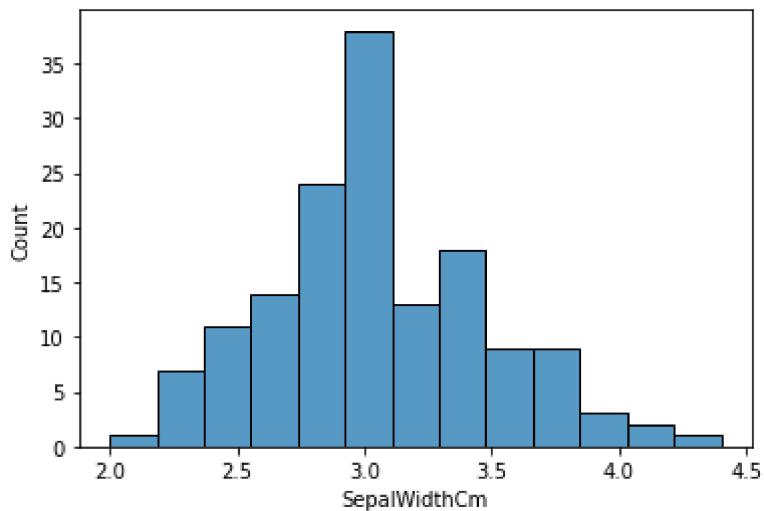
```
Out[77]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

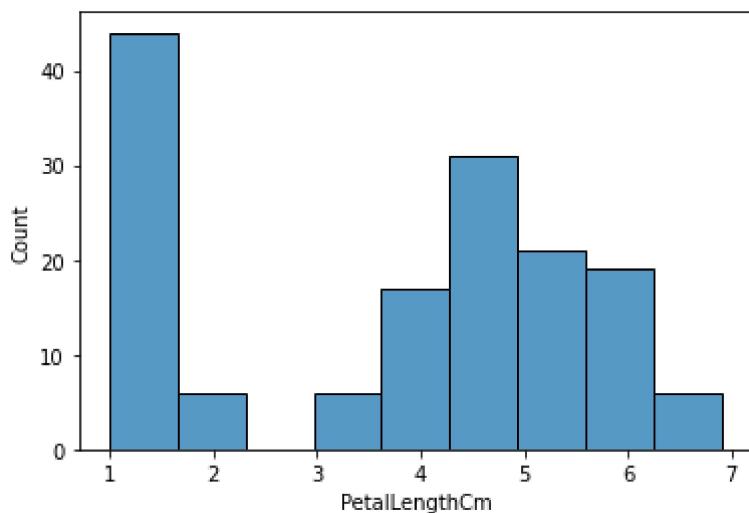
```
In [78]: b1=sns.boxplot(x="SepalLengthCm",data=df2)
```



```
In [79]: A=sns.histplot(data=df2,x="SepalWidthCm")
```



```
In [80]: A=sns.histplot(data=df2,x="PetalLengthCm")
```



```
In [81]: from sklearn.model_selection import train_test_split
```

```
In [121...]: df2
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
147	148	6.5	3.0	5.2	2.0 Iris-virginica
148	149	6.2	3.4	5.4	2.3 Iris-virginica
149	150	5.9	3.0	5.1	1.8 Iris-virginica

150 rows × 6 columns

In [122...]

```
X=df2.iloc[:,2:5]
X
```

Out[122...]

	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	3.5	1.4	0.2
1	3.0	1.4	0.2
2	3.2	1.3	0.2
3	3.1	1.5	0.2
4	3.6	1.4	0.2
...
145	3.0	5.2	2.3
146	2.5	5.0	1.9
147	3.0	5.2	2.0
148	3.4	5.4	2.3
149	3.0	5.1	1.8

150 rows × 3 columns

In [123...]

```
y=df2["Species"]
```

In [120...]

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
```

In [86]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix
```

In [87]:

```
classifier = DecisionTreeClassifier()
```

In [88]:

```
classifier.fit(X_train, y_train)
```

Out[88]:

```
DecisionTreeClassifier()
```

In [89]:

```
y_pred=classifier.predict(X_test)
y_pred
```

Out[89]: array(['Iris-virginica', 'Iris-virginica', 'Iris-versicolor',
 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
 'Iris-setosa', 'Iris-virginica', 'Iris-virginica',
 'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
 'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
 'Iris-setosa', 'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor',
 'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
 'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
 'Iris-virginica', 'Iris-setosa', 'Iris-versicolor', 'Iris-setosa',
 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica',
 'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
 'Iris-virginica', 'Iris-virginica', 'Iris-setosa',
 'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
 'Iris-versicolor', 'Iris-virginica'], dtype=object)

In [90]:

```
confusion_matrix(y_test,y_pred)
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	14
Iris-versicolor	0.88	1.00	0.93	14
Iris-virginica	1.00	0.88	0.94	17
accuracy			0.96	45
macro avg	0.96	0.96	0.96	45
weighted avg	0.96	0.96	0.96	45

WINEQT DATASET

In [64]:

```
df3=pd.read_csv(r"C:\Users\jagan\OneDrive\Desktop\WineQT.csv")
df3.head()
```

Out[64]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	

In [68]:

```
x=df3.iloc[:,2:12]
y=df3['quality']
```

```
In [73]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
classifier = DecisionTreeClassifier()
classifier.fit(X_train, y_train)
y_pred=classifier.predict(X_test)
y_pred
confusion_matrix(y_test,y_pred)
```

```
Out[73]: array([[ 1,  0,  0,  0,  0,  0],
   [ 0,  8,  0,  0,  0,  0],
   [ 0,  0, 151,  0,  0,  0],
   [ 0,  0,  0, 138,  0,  0],
   [ 0,  0,  0,  0, 41,  0],
   [ 0,  0,  0,  0,  0,  4]], dtype=int64)
```

```
In [75]: pprint(classification_report(y_test,y_pred))
```

```
('          precision    recall  f1-score   support\n'
 '\n'
 '          3       1.00      1.00      1.00       1\n'
 '          4       1.00      1.00      1.00       8\n'
 '          5       1.00      1.00      1.00      151\n'
 '          6       1.00      1.00      1.00      138\n'
 '          7       1.00      1.00      1.00      41\n'
 '          8       1.00      1.00      1.00       4\n'
 '\n'
 '  accuracy                           1.00      343\n'
 'macro avg       1.00      1.00      1.00      343\n'
 'weighted avg     1.00      1.00      1.00      343')
```

FINDING ACCURACY WITH SOME GIVEN TEST SIZE

```
In [142...]: from sklearn.metrics import accuracy_score
test_size1=[0.5,0.6,0.4,0.2,0.3]
for i in test_size1:
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=i,random_state=42)
    print("TEST_SIZE = ",i)
    print("\n")
    classifier = DecisionTreeClassifier()
    classifier.fit(X_train, y_train)
    y_pred=classifier.predict(X_test)
    m=confusion_matrix(y_test,y_pred)
    print("THE CONFUSION MATRICS IS : ",m)
    print("\n")
    a=accuracy_score(y_test,y_pred)
    print('ACCURACY SCORE : ',a)
    print("\n")
    print("THE CLASSIFICATION REPORT IS : ",classification_report(y_test,y_pred))
    print("-----")
```

TEST_SIZE = 0.5

THE CONFUSION MATRICS IS : [[22 0 0]
[0 25 3]
[0 0 25]]

ACCURACY SCORE : 0.96

THE CLASSIFICATION REPORT IS :			precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	22		
Iris-versicolor	1.00	0.89	0.94	28		
Iris-virginica	0.89	1.00	0.94	25		
			accuracy	0.96	75	
macro avg	0.96	0.96	0.96	75		
weighted avg	0.96	0.96	0.96	75		

TEST_SIZE = 0.6

THE CONFUSION MATRICES IS : [[27 0 0]
[0 29 4]
[0 3 27]]

ACCURACY SCORE : 0.9222222222222222

THE CLASSIFICATION REPORT IS :			precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	27		
Iris-versicolor	0.91	0.88	0.89	33		
Iris-virginica	0.87	0.90	0.89	30		
			accuracy	0.92	90	
macro avg	0.93	0.93	0.93	90		
weighted avg	0.92	0.92	0.92	90		

TEST_SIZE = 0.4

THE CONFUSION MATRICES IS : [[16 0 0]
[0 22 2]
[0 0 20]]

ACCURACY SCORE : 0.9666666666666667

THE CLASSIFICATION REPORT IS :			precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	16		
Iris-versicolor	1.00	0.92	0.96	24		
Iris-virginica	0.91	1.00	0.95	20		
			accuracy	0.97	60	
macro avg	0.97	0.97	0.97	60		
weighted avg	0.97	0.97	0.97	60		

TEST_SIZE = 0.2

THE CONFUSION MATRICES IS : [[12 0 0]

```
[ 0  8  1]
[ 0  1  8]]
```

ACCURACY SCORE : 0.9333333333333333

THE CLASSIFICATION REPORT IS :

	precision	recall	f1-score	support
--	-----------	--------	----------	---------

Iris-setosa	1.00	1.00	1.00	12
Iris-versicolor	0.89	0.89	0.89	9
Iris-virginica	0.89	0.89	0.89	9
accuracy			0.93	30
macro avg	0.93	0.93	0.93	30
weighted avg	0.93	0.93	0.93	30

TEST_SIZE = 0.3

THE CONFUSION MATRICES IS : [[13 0 0]

```
[ 0 12  1]
[ 0  0 19]]
```

ACCURACY SCORE : 0.9777777777777777

THE CLASSIFICATION REPORT IS :

	precision	recall	f1-score	support
--	-----------	--------	----------	---------

Iris-setosa	1.00	1.00	1.00	13
Iris-versicolor	1.00	0.92	0.96	13
Iris-virginica	0.95	1.00	0.97	19
accuracy			0.98	45
macro avg	0.98	0.97	0.98	45
weighted avg	0.98	0.98	0.98	45

CONCLUSION

Here i tried to do the Exploratory Data Analysis on perticular dataset.And i splitted the dataset into two which is training and testing data.After that i find the accuracy score , classification report,etc and i tried to predict

REFERENCE

<https://seaborn.pydata.org/>

<https://www.geeksforgeeks.org/matplotlib-tutorial/>

<https://scikit-learn.org/stable/>

In []: