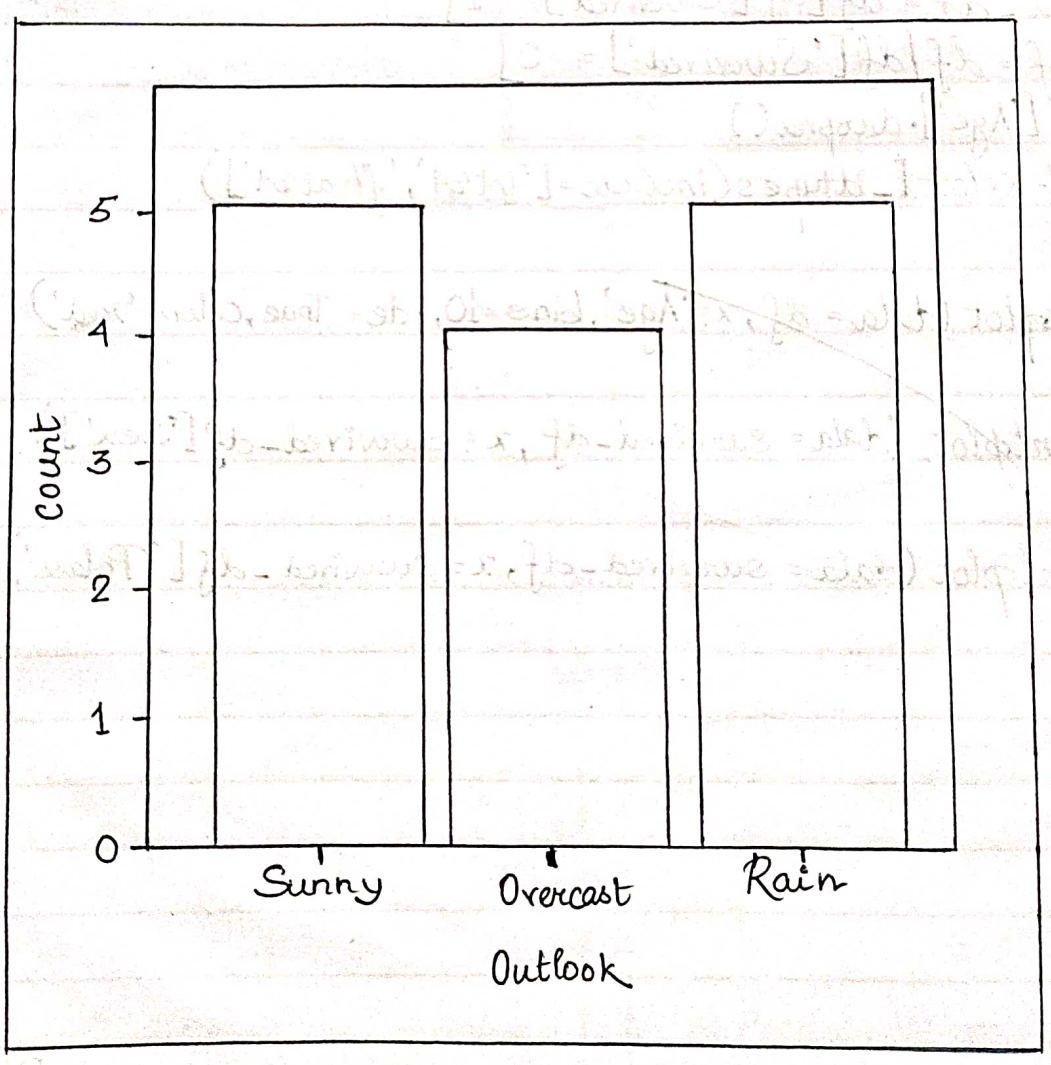
	Outlook	Temperature	Humidity	Wind	Play Tennis
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes



Expt, No.

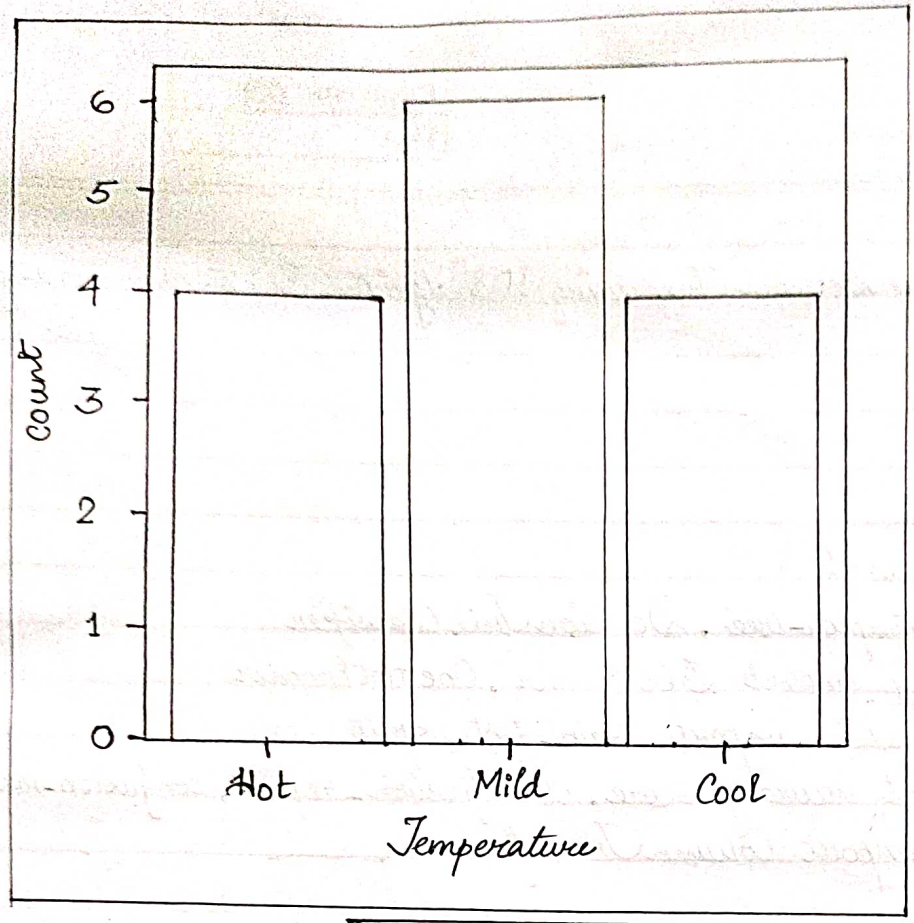
Page No. 09
Date

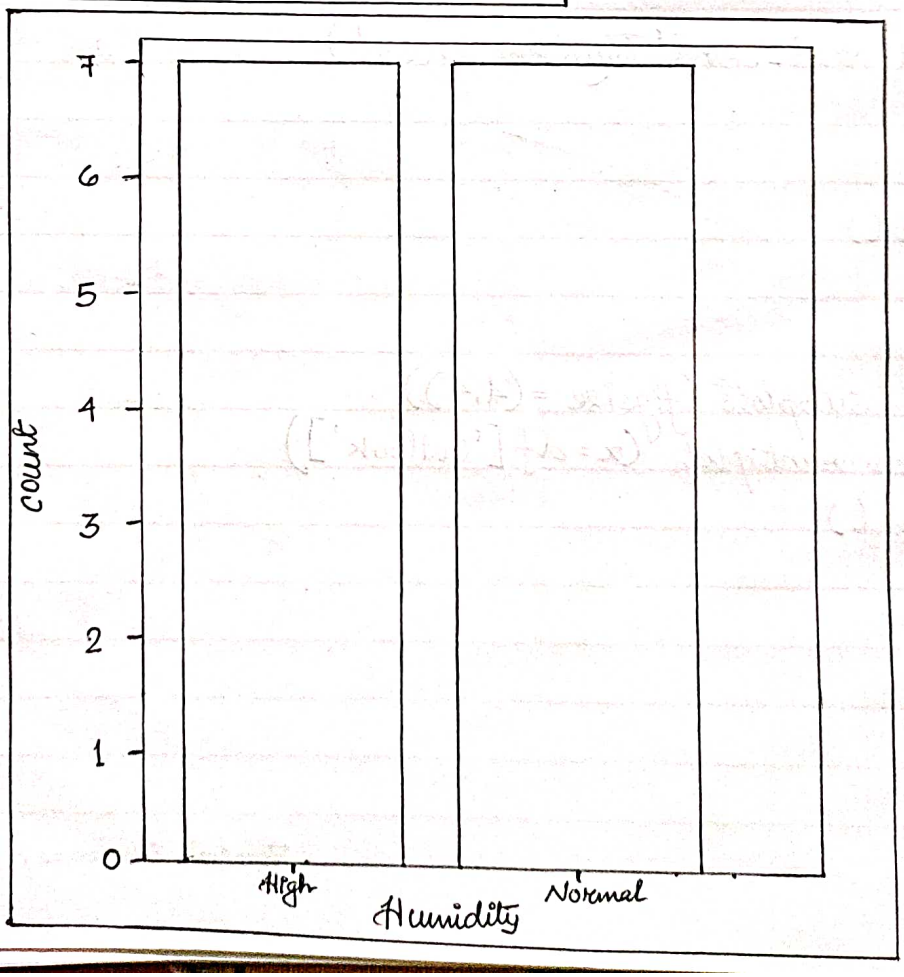
7) Write a python code for Decision Tree using ID3 algorithm & its corresponding csv file.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree, DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.compose import ColumnTransformer

df = pd.read_csv('PlayTennis.csv')

df.head()

ax = plt.subplots(figsize=(4,4))
ax = sns.countplot(x=df['Outlook'])
plt.show()
```

Avijit®

Teacher's Signature :.....


6
5
4
3
2
1
0
count
Hot
Mild
Cool
Temperature


7
6
5
4
3
2
1
0
count
High
Normal
Humidity

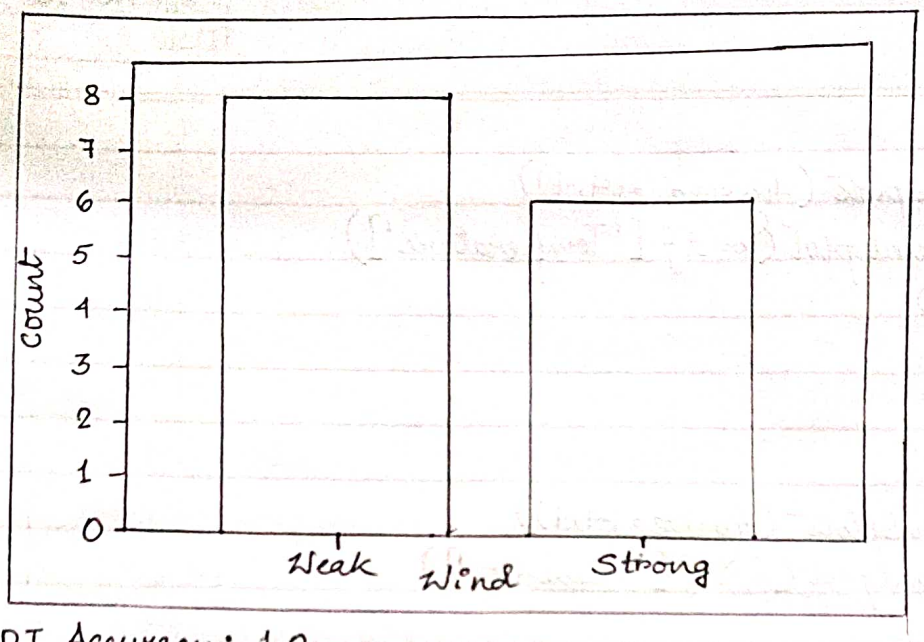
Expt, No.

Page No. 10
Date

```
ax = plt.subplots (figsize = (4,4))
ax = sns.countplot (x = df ['Temperature'])
plt.show ()


ax = plt.subplots (figsize = (4,4))
ax = sns.countplot (x = df ['Humidity'])
plt.show()


ax = plt.subplots (figsize = (4,4))
ax = sns.countplot (x = df ['Wind'])
plt.show()


X = df.iloc [:, :-1]
y = df.iloc[:, -1]


categorical_features = ['Outlook', 'Temperature', 'Humidity', 'Wind']
preprocessor = ColumnTransfer (
        transformers=[
            ('encoder', OneHotEncoder(), categorical_features)
        ],
```

Avijit®

Teacher's Signature :.....



DT Accuracy : 1.0

DT Confusion Matrix :

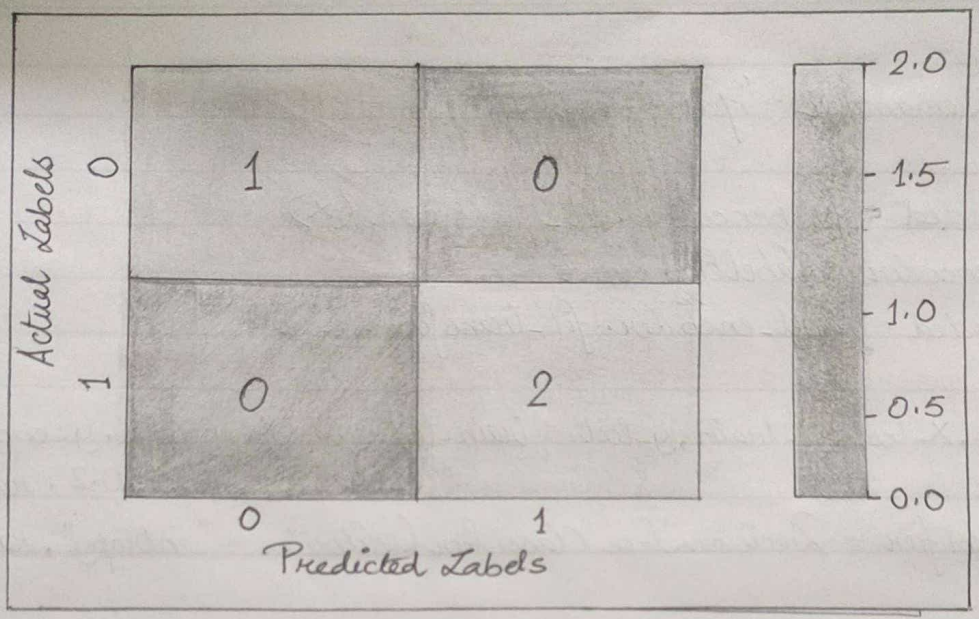
```
[[1 0]
 [0 2]]
```

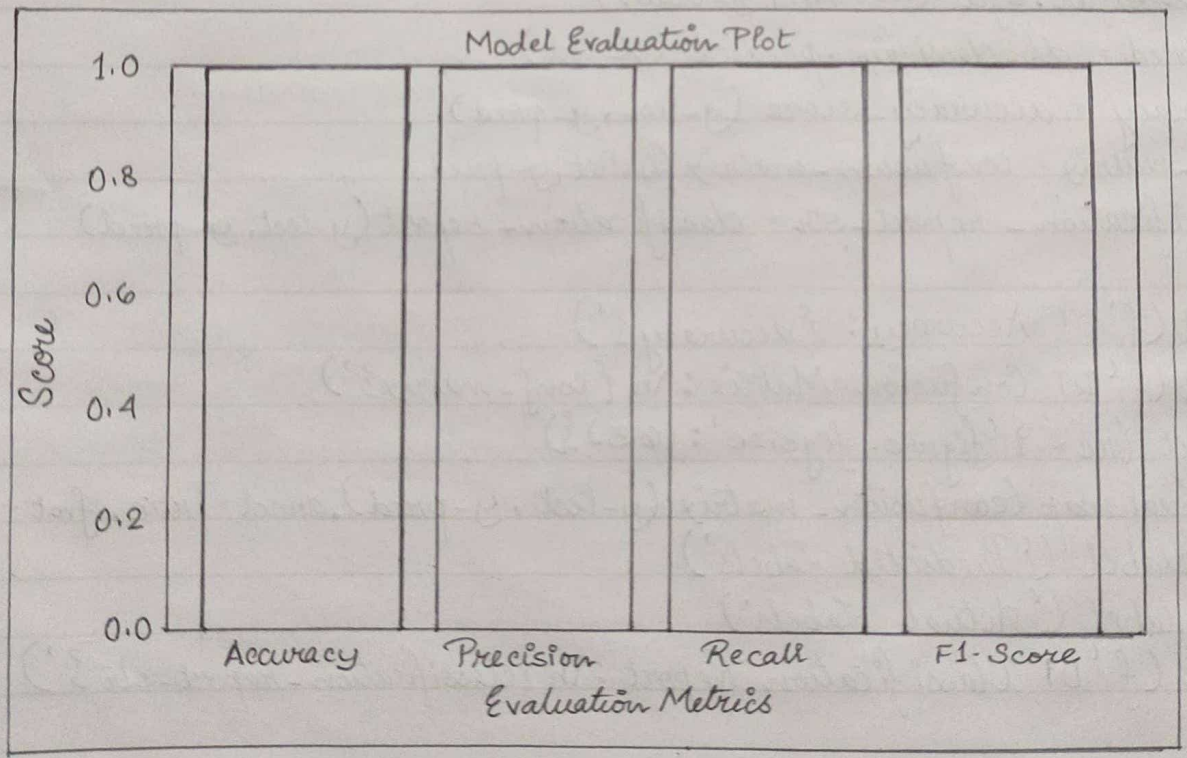
DT Classification Report :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1
1	1.00	1.00	1.00	2
accuracy			1.00	3
macro avg	1.00	1.00	1.00	3
weighted avg	1.00	1.00	1.00	3

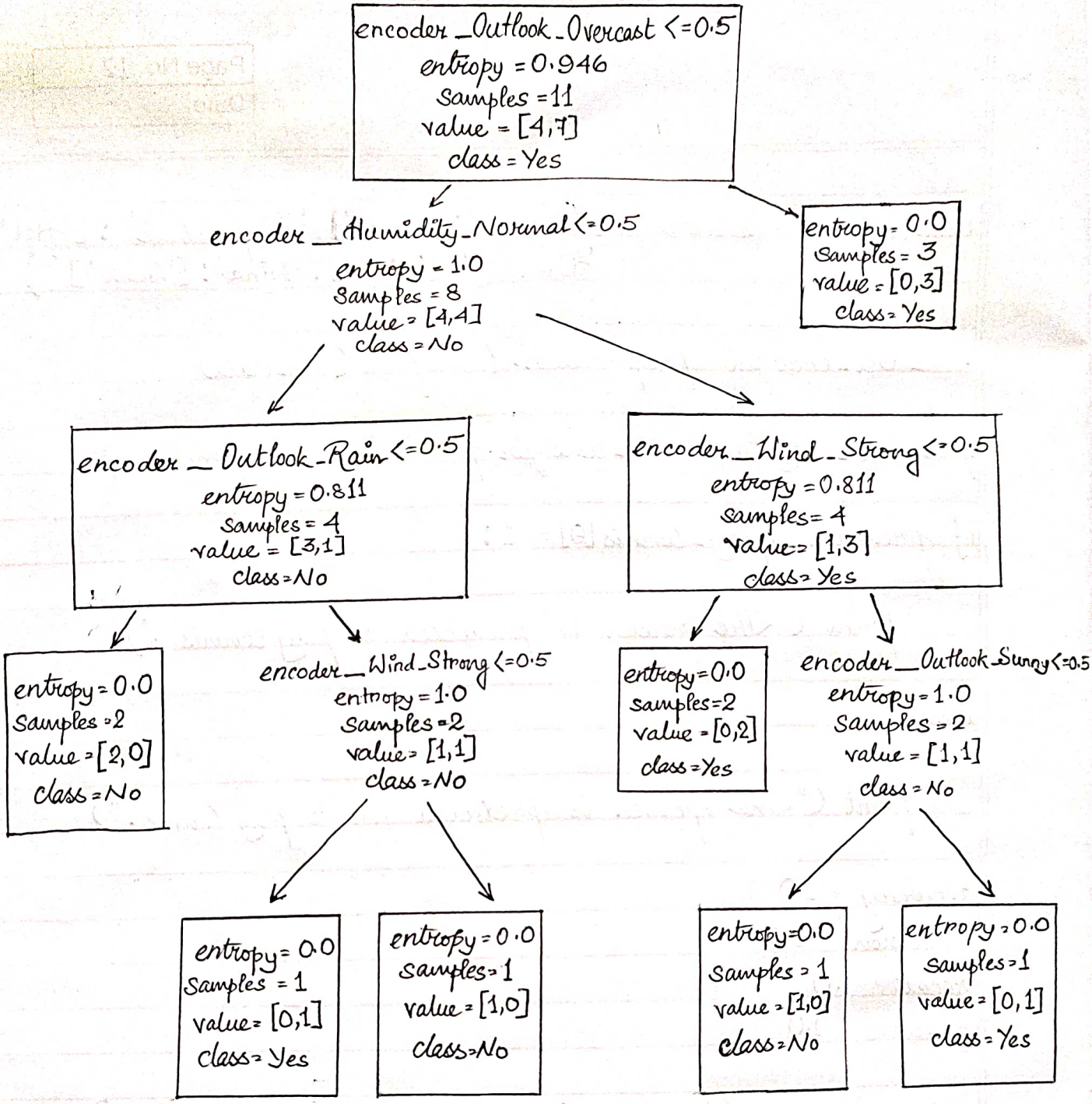
```
    remainder = 'passthrough'
)
X_encoded = preprocessor.fit_transform(X)
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

X_train, X_test, y_train, y_test = train_test_split(X_encoded, y_encoded,
                                   test_size = 0.2, random_state = 42)
dt_classifier = DecisionTreeClassifier(criterion = "entropy", random_state=42)

dt_classifier.fit(X_train, y_train)
y_pred = dt_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_report_str = classification_report(y_test, y_pred)

print(f'DT Accuracy : {accuracy}')
print(f'DT Confusion Matrix : \n {conf_matrix}')
sns.set(rc = {'figure.figsize':(6,3)})
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d')
plt.xlabel('Predicted Labels')
plt.ylabel('Actual Labels')
print(f'DT Classification Report : \n {classification_report_str}')
```


Teacher's Signature :.....


Actual Labels
0
1
1
0
0
2
0
1
Predicted Labels
2.0
1.5
1.0
0.5
0.0


Model Evaluation Plot
Score
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Precision
Recall
F1-Score
Evaluation Metrics

```
new_data = pd.DataFrame({'Outlook':['Sunny'], 'Temperature' : ['Hot'],
                         'Humidity':['High'], 'Wind':['Weak']})

new_data_encoded = preprocessor.transform(new_data)

predicted_play_tennis = dt_classifier.predict(new_data_encoded)

if predicted_play_tennis[0]==1:

    print("The person is predicted to play tennis.")

else:

    print("The person is predicted not to play tennis.")

accuracy = 1.0
precision = 1.0
recall = 1.0
f1_score = 1.0

metrics_names = ['Accuracy', 'Precision', 'Recall', 'F1-Score']
metrics_values = [accuracy, precision, recall, f1_score]

plt.bar(metrics_names, metrics_values, color=['blue','green','orange','red'])
plt.ylim([0,1])
plt.xlabel('Evaluation Metrics')
```


encoder_Outlook_Overcast <=0.5
entropy = 0.946
samples = 11
value = [4,7]
class = Yes
encoder_Humidity_Normal <=0.5
entropy = 1.0
samples = 8
value = [4,4]
class = No
entropy = 0.0
samples = 3
value = [0,3]
class = Yes
encoder_Outlook_Rain <=0.5
entropy = 0.811
samples = 4
value = [3,1]
class = No
encoder_Wind_Strong <=0.5
entropy = 0.811
samples = 4
value = [1,3]
class = Yes
entropy = 0.0
samples = 2
value = [2,0]
class = No
encoder_Wind_Strong <=0.5
entropy = 1.0
samples = 2
value = [1,1]
class = No
entropy = 0.0
samples = 2
value = [0,2]
class = Yes
encoder_Outlook_Sunny <=0.5
entropy = 1.0
samples = 2
value = [1,1]
class = No
entropy = 0.0
samples = 1
value = [0,1]
class = Yes
entropy = 0.0
samples = 1
value = [1,0]
class = No
entropy = 0.0
samples = 1
value = [1,0]
class = No
entropy = 0.0
samples = 1
value = [0,1]
class = Yes

```
plt.ylabel('Score')
plt.title('Model Evaluation Plot')
plt.show()

plt.figure(figsize=(12,8))
plot_tree(dt_classifier, filled=True, feature_names=list(preprocessor.
        get_feature_names_out(X.columns)), class_names=
        list(label_encoder.classes_))
plt.show()
```


Teacher's Signature :.....