Experiment 3 – Implementation of Classification Algorithm

Aim: Implementation of Classification algorithm Using

1. Decision Tree ID3

2. Naïve Bayes algorithm

Theory: Classification is a fundamental task in machine learning and data analysis that involves categorizing data points into predefined classes or categories based on their features or attributes. It's widely used in various applications, such as spam email detection, sentiment analysis, and medical diagnosis. Two common classification algorithms are Naïve Bayes and ID3.

Naïve Bayes: Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes that the features used for classification are conditionally independent, which means that the presence of one feature does not affect the presence of another. Despite this "naïve" assumption, Naïve Bayes often works well in practice and is particularly useful for text classification and spam detection. It calculates the probability of a data point belonging to each class and selects the class with the highest probability.

ID3 (Iterative Dichotomiser 3): ID3 is a decision tree-based classification algorithm that recursively splits the dataset into subsets based on the most informative attributes. It selects attributes that maximize information gain (reduce uncertainty) at each step to create a tree structure. ID3 is used for both classification and feature selection and is advantageous in scenarios where the data consists of discrete values and can be easily visualized.

These classifiers serve as essential tools in the field of machine learning, each with its own strengths and weaknesses, making them suitable for different types of classification tasks.

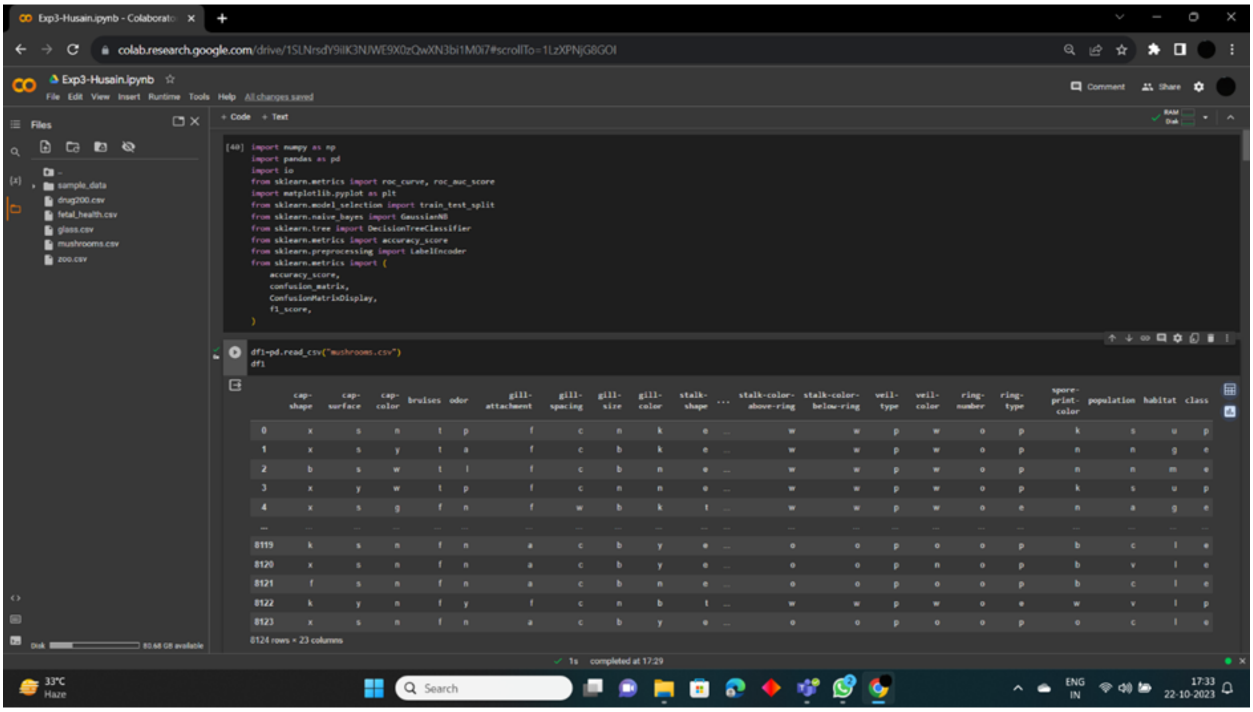
Implementation: For this experiment we were required to perform the following:

Part A:
Program using inbuilt functions.
Predict class of unseen samples.
Results should display
1. Confusion matrix
2. Classifier accuracy
Part B:
1. Compare results of DT and NB for 5 datasets.
2. Plot AUROC
3. Plot comparison graphs using the results of DT and NB
Part C:
Modify DT/NB to use k-fold cross validation and ensemble models

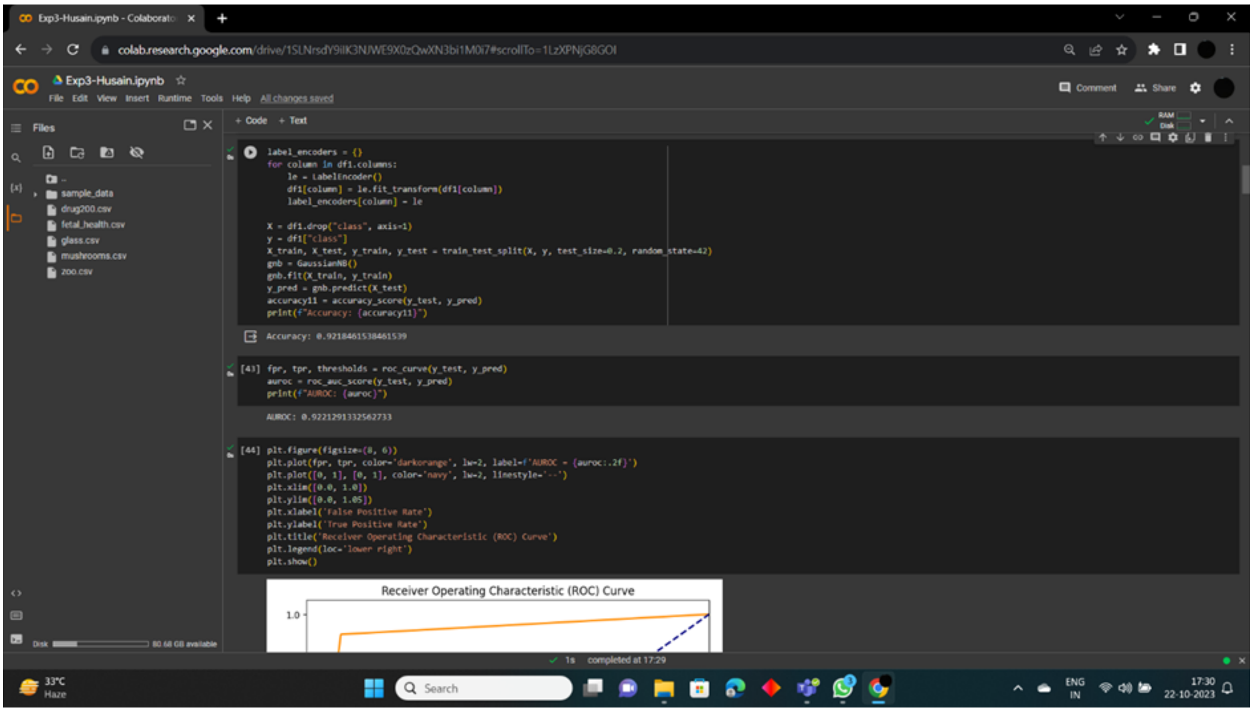
We have chosen the following 5 datasets:

- mushrooms.csv
- drug200.csv
- fetal_health.csv
- zoo.csv
- glass.csv

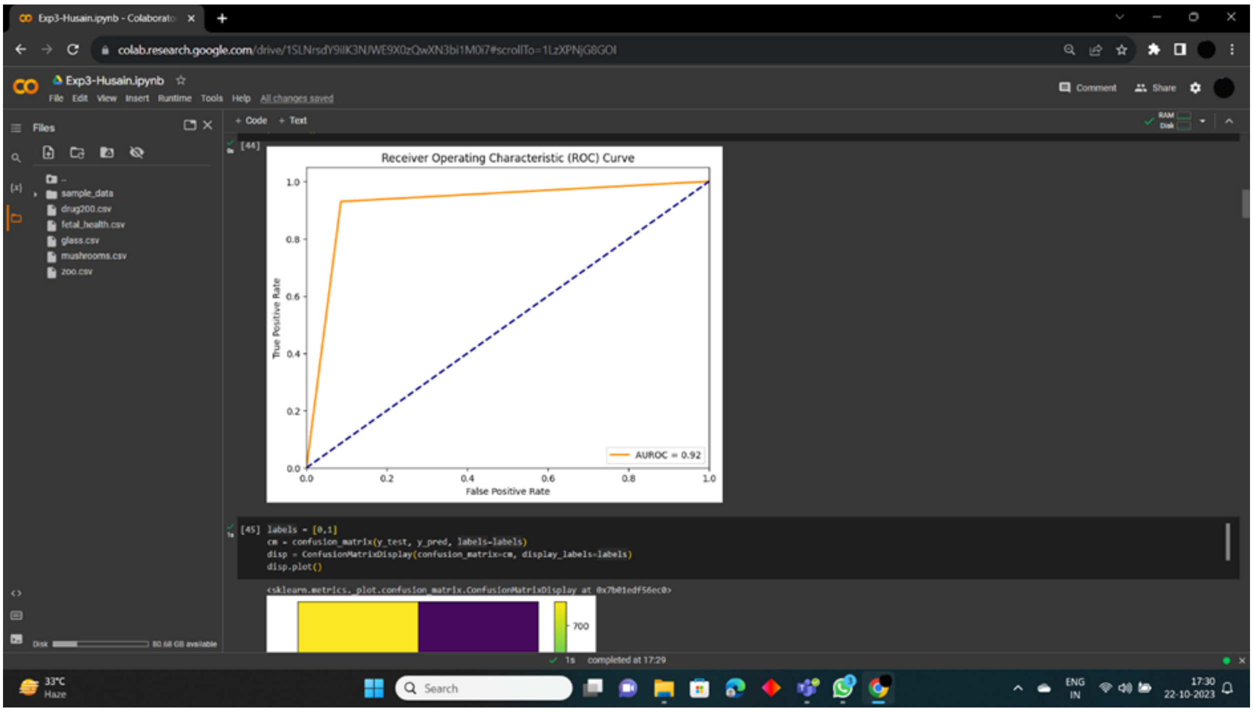
We first performed Naïve Bayes classification and then the Decision Tree classification and then compared the results of both. In the end we also performed k-fold validation.

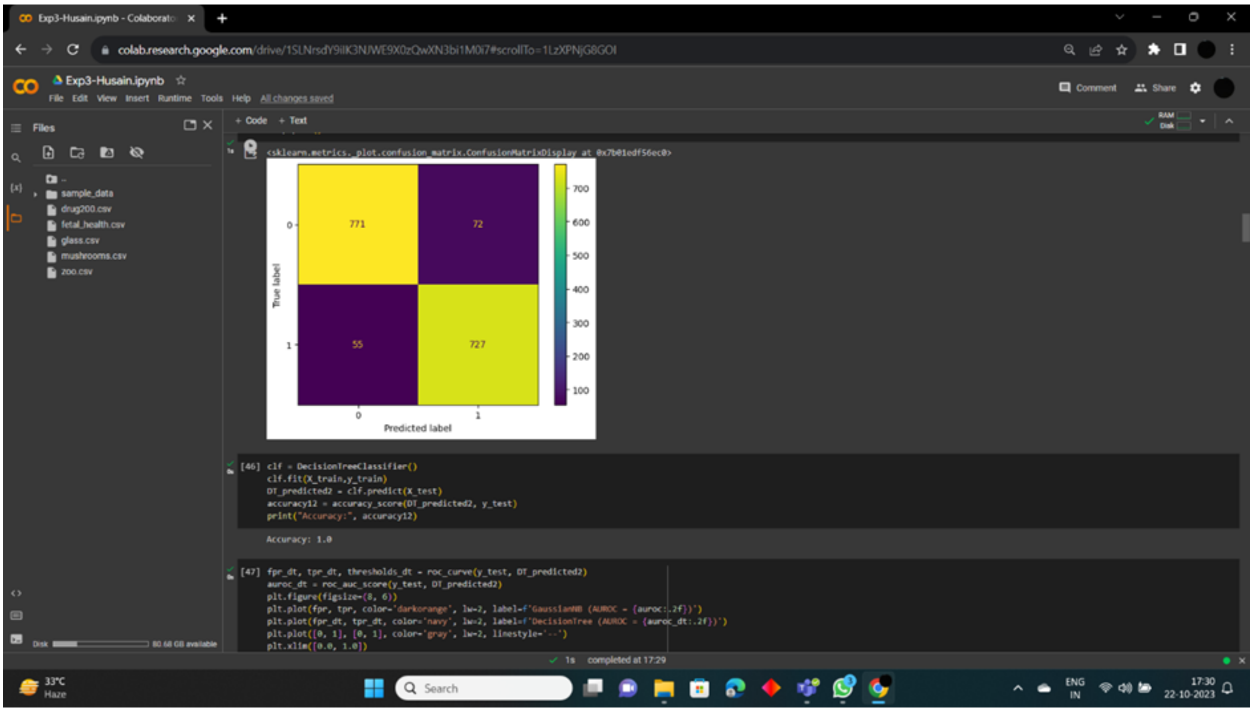


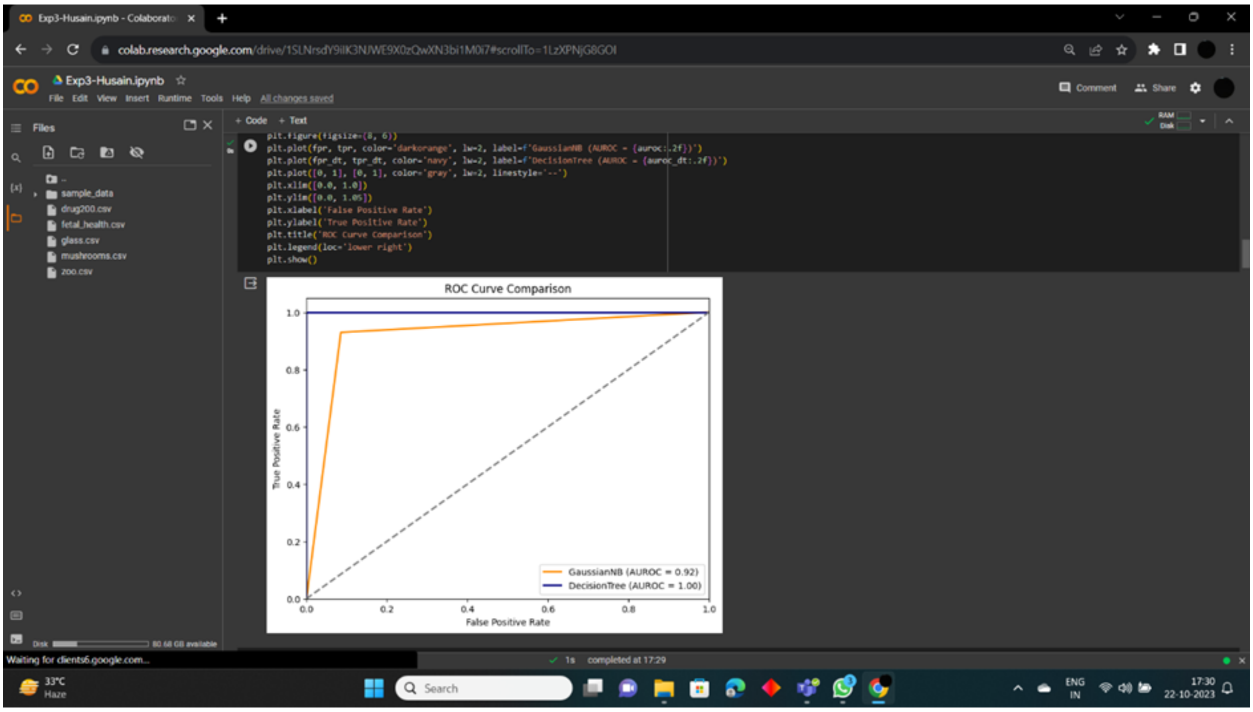
Exp3-Husain.ipynb
[40] import numpy as np
import pandas as pd
import io
from sklearn.metrics import roc_curve, roc_auc_score
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import (
accuracy_score,
confusion_matrix,
ConfusionMatrixDisplay,
f1_score,
)
df1=pd.read_csv("mushrooms.csv")
df1
8124 rows × 23 columns



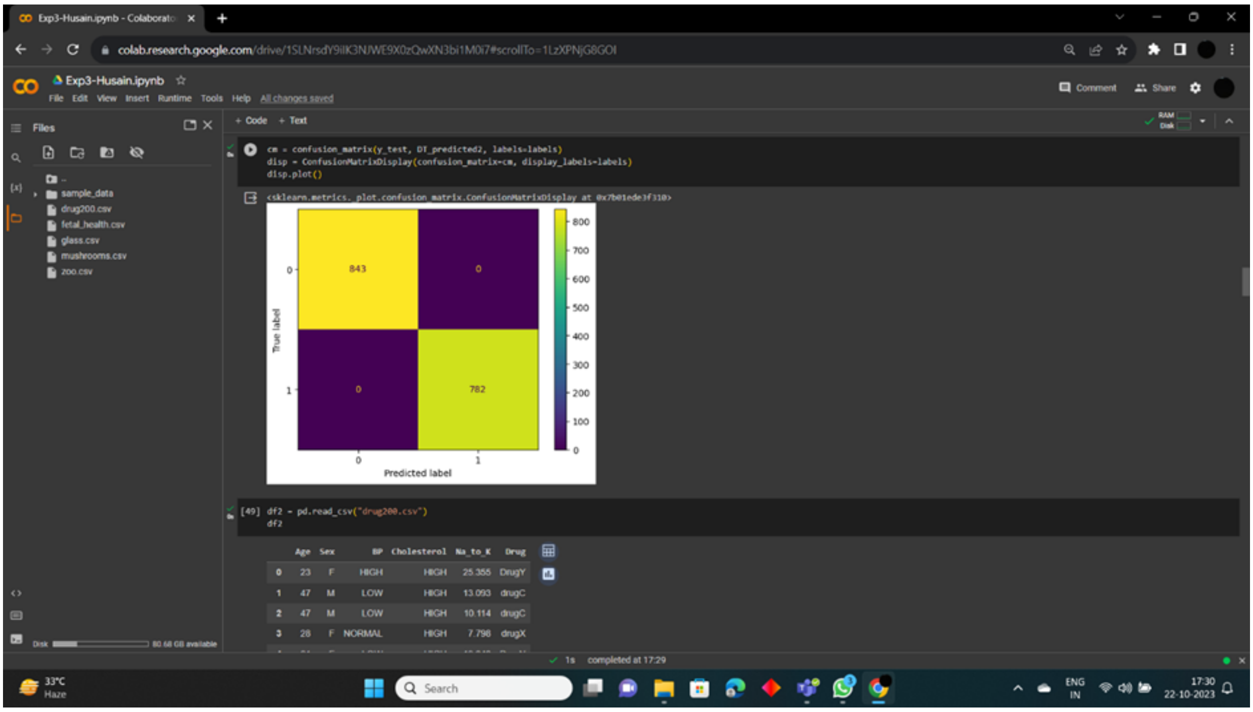
Exp3-Husain.ipynb
label_encoders = {}
for column in df1.columns:
le = LabelEncoder()
df1[column] = le.fit_transform(df1[column])
label_encoders[column] = le
X = df1.drop("class", axis=1)
y = df1["class"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
accuracy11 = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy11}")
Accuracy: 0.9218461538461539
[43] fpr, tpr, thresholds = roc_curve(y_test, y_pred)
auroc = roc_auc_score(y_test, y_pred)
print(f"AUROC: {auroc}")
AUROC: 0.9221291332562733
[44] plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'AUROC = {auroc:.2f}')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()
Receiver Operating Characteristic (ROC) Curve
1.0


Exp3-Husain.ipynb
Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
AUROC = 0.92
[45] labels = [0,1]
cm = confusion_matrix(y_test, y_pred, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
disp.plot()
completed at 17:29

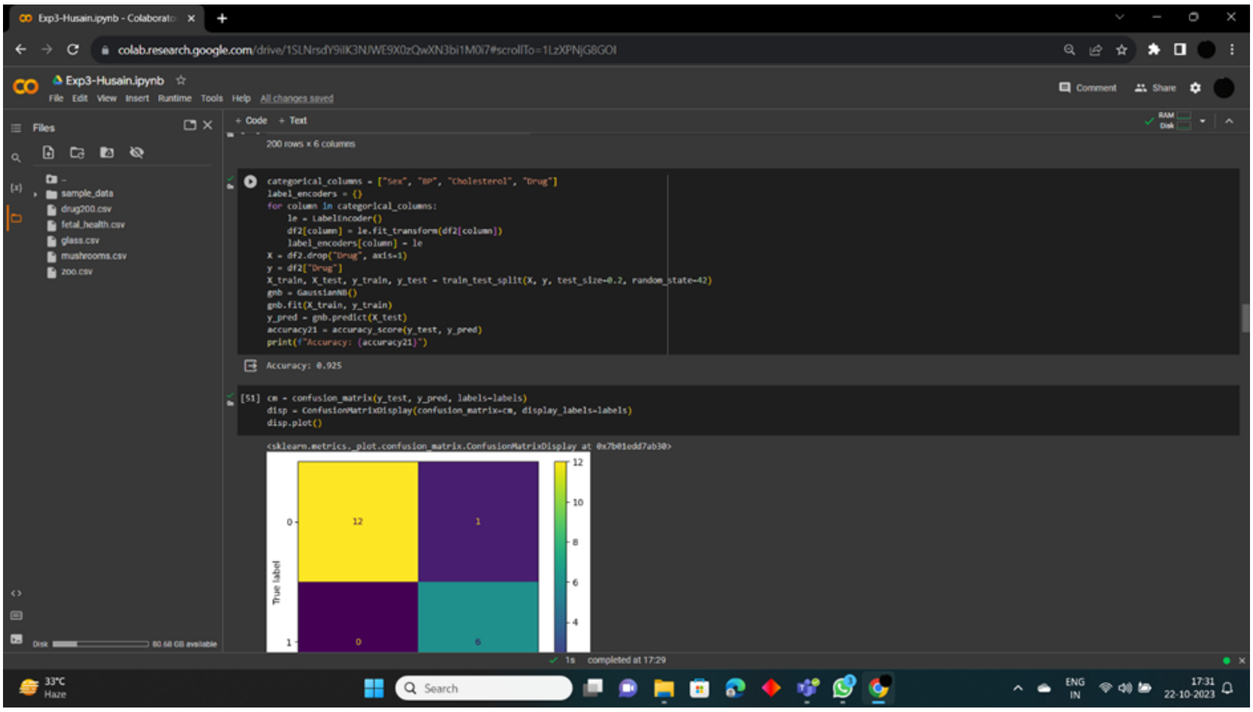

Exp3-Husain.ipynb
771
72
55
727
True label
Predicted label
[46] clf = DecisionTreeClassifier()
clf.fit(X_train,y_train)
DT_predicted2 = clf.predict(X_test)
accuracy12 = accuracy_score(DT_predicted2, y_test)
print("Accuracy:", accuracy12)
Accuracy: 1.0
[47] fpr_dt, tpr_dt, thresholds_dt = roc_curve(y_test, DT_predicted2)
auroc_dt = roc_auc_score(y_test, DT_predicted2)
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'GaussianNB (AUROC = {auroc:.2f})')
plt.plot(fpr_dt, tpr_dt, color='navy', lw=2, label=f'DecisionTree (AUROC = {auroc_dt:.2f})')
plt.plot([0, 1], [0, 1], color='gray', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
completed at 17:29

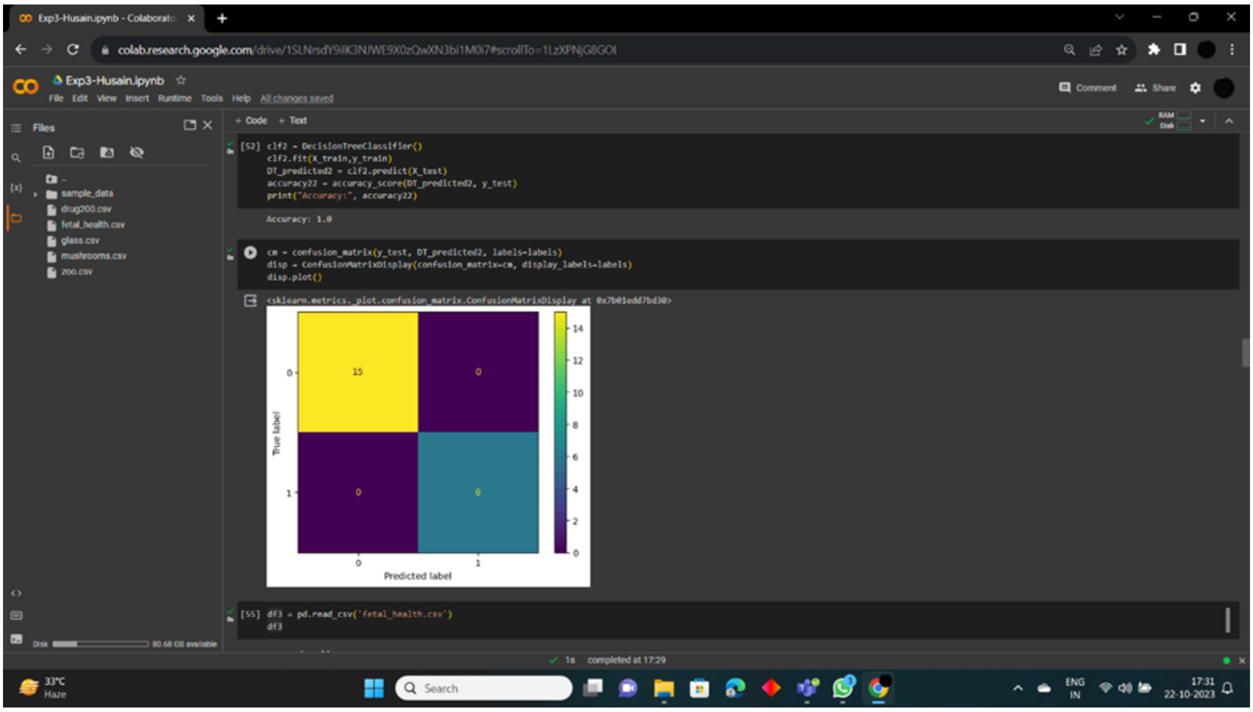


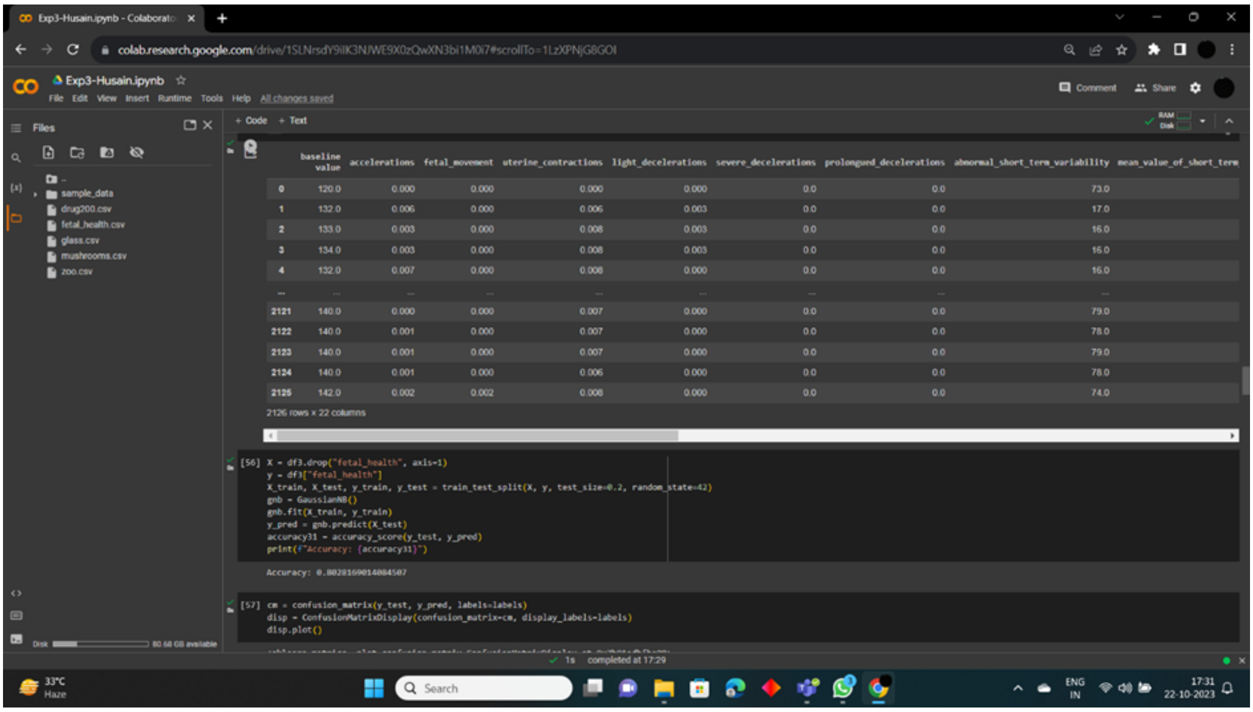
Exp3-Husain.ipynb - Colaborato
colab.research.google.com/drive/1SLNrsdY9ilK3NJWE9X0zQwXN3bi1M0i7#scrollTo=1LzXPNjG8GOI
Exp3-Husain.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
sample_data
drug200.csv
fetal_health.csv
glass.csv
mushrooms.csv
zoo.csv
+ Code + Text
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'GaussianNB (AUROC = {auroc:.2f})')
plt.plot(fpr_dt, tpr_dt, color='navy', lw=2, label=f'DecisionTree (AUROC = {auroc_dt:.2f})')
plt.plot([0, 1], [0, 1], color='gray', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve Comparison')
plt.legend(loc='lower right')
plt.show()
ROC Curve Comparison
True Positive Rate
False Positive Rate
GaussianNB (AUROC = 0.92)
DecisionTree (AUROC = 1.00)
Waiting for clients6.google.com...
1s completed at 17:29
33°C Haze
Search
17:30 22-10-2023



Exp3-Husain.ipynb - Colaborato
colab.research.google.com/drive/1SLNrsdY9ilK3NJWE9X0zQwXN3bi1M0i7#scrollTo=1LzXPNjG8GOI
Exp3-Husain.ipynb
File Edit View Insert Runtime Tools Help All changes saved
cm = confusion_matrix(y_test, DT_predicted2, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
disp.plot()
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7b01ede3f310>
843
0
0
782
True label
Predicted label
[49] df2 = pd.read_csv("drug200.csv")
df2
Age Sex BP Cholesterol Na_to_K Drug
0 23 F HIGH HIGH 25.355 DrugY
1 47 M LOW HIGH 13.093 drugC
2 47 M LOW HIGH 10.114 drugC
3 28 F NORMAL HIGH 7.798 drugX
1s completed at 17:29
33°C Haze
Search
17:30 22-10-2023


Exp3-Husain.ipynb - Colaborato
colab.research.google.com/drive/1SLNrsdY9ilK3NJWE9X0zQwXN3bi1M0i7#scrollTo=1LzXPNjG8GOI
Exp3-Husain.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
sample_data
drug200.csv
fetal_health.csv
glass.csv
mushrooms.csv
zoo.csv
+ Code + Text
200 rows × 6 columns
categorical_columns = ["Sex", "BP", "Cholesterol", "Drug"]
label_encoders = {}
for column in categorical_columns:
le = LabelEncoder()
df2[column] = le.fit_transform(df2[column])
label_encoders[column] = le
X = df2.drop("Drug", axis=1)
y = df2["Drug"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
accuracy21 = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy21}")
Accuracy: 0.925
[51] cm = confusion_matrix(y_test, y_pred, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
disp.plot()
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7b01edd7ab30>
12
1
0
6
True label
1s completed at 17:29
33°C Haze
Search
ENG IN
17:31 22-10-2023


Exp3-Husain.ipynb - Colaborato
colab.research.google.com/drive/1SLNrsdY9ilK3NJWE9X0zQwXN3bi1M0i7#scrollTo=1LzXPNjG8GOI
Exp3-Husain.ipynb
File Edit View Insert Runtime Tools Help All changes saved
[52] clf2 = DecisionTreeClassifier()
clf2.fit(X_train,y_train)
DT_predicted2 = clf2.predict(X_test)
accuracy22 = accuracy_score(DT_predicted2, y_test)
print("Accuracy:", accuracy22)
Accuracy: 1.0
cm = confusion_matrix(y_test, DT_predicted2, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
disp.plot()
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7b01edd7bd30>
15
0
0
6
True label
Predicted label
[55] df3 = pd.read_csv('fetal_health.csv')
df3
1s completed at 17:29
33°C Haze
Search
ENG IN
17:31 22-10-2023



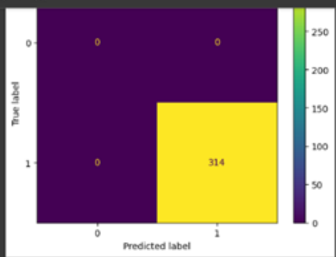
	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variability	mean_value_of_short_term
0	120.0	0.000	0.000	0.000	0.000	0.0	0.0	73.0	
1	132.0	0.006	0.000	0.006	0.003	0.0	0.0	17.0	
2	133.0	0.003	0.000	0.008	0.003	0.0	0.0	16.0	
3	134.0	0.003	0.000	0.008	0.003	0.0	0.0	16.0	
4	132.0	0.007	0.000	0.008	0.000	0.0	0.0	16.0	
...	...	...	...	...	...	...	...	...	
2121	140.0	0.000	0.000	0.007	0.000	0.0	0.0	79.0	
2122	140.0	0.001	0.000	0.007	0.000	0.0	0.0	78.0	
2123	140.0	0.001	0.000	0.007	0.000	0.0	0.0	79.0	
2124	140.0	0.001	0.000	0.006	0.000	0.0	0.0	78.0	
2125	142.0	0.002	0.002	0.008	0.000	0.0	0.0	74.0	

2126 rows × 22 columns

```
[56] X = df3.drop("fetal_health", axis=1)
     y = df3["fetal_health"]
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
     gnb = GaussianNB()
     gnb.fit(X_train, y_train)
     y_pred = gnb.predict(X_test)
     accuracy31 = accuracy_score(y_test, y_pred)
     print(f"Accuracy: {accuracy31}")
```

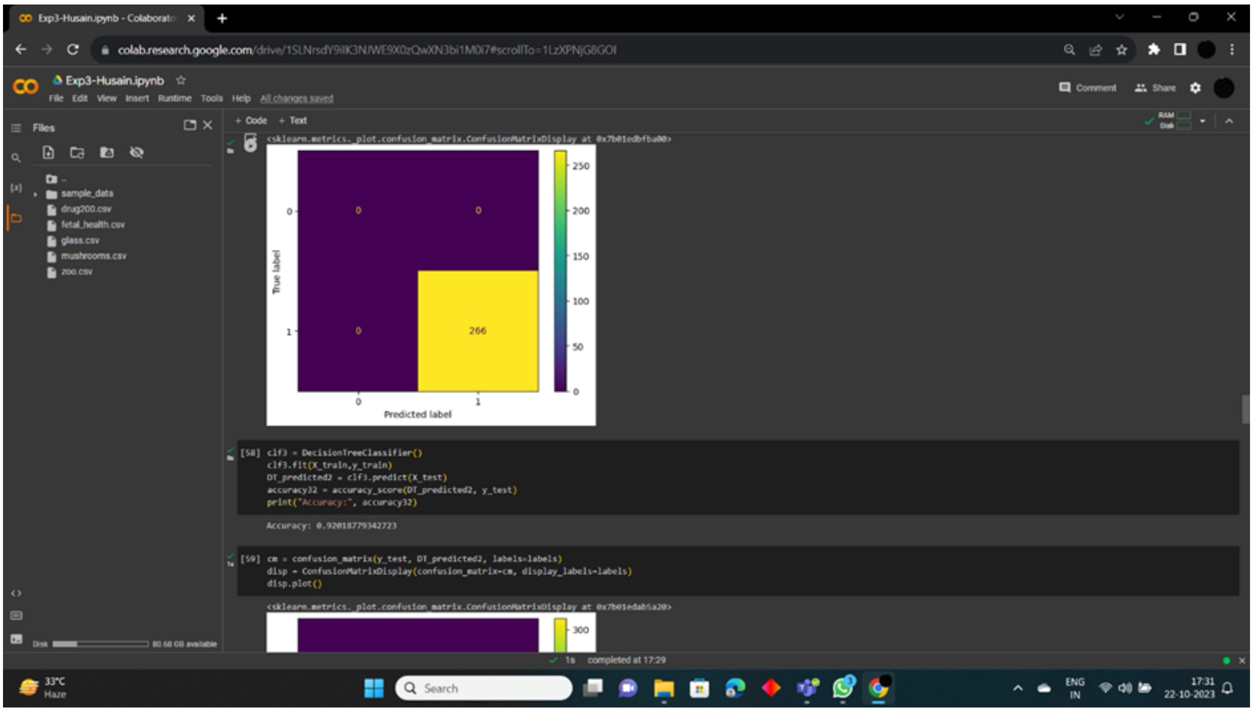
Accuracy: 0.8028169014084507

```
[57] cm = confusion_matrix(y_test, y_pred, labels=labels)
     disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
     disp.plot()
```

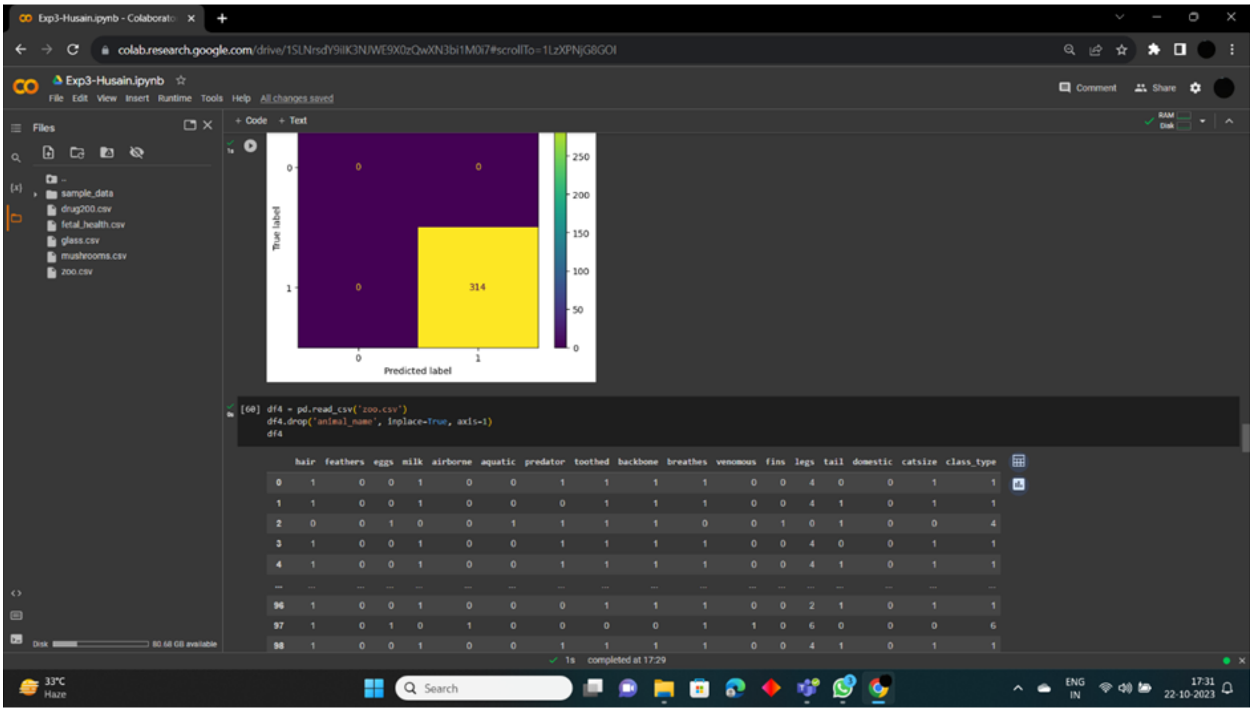



```
[60] df4 = pd.read_csv('zoo.csv')
     df4.drop('animal_name', inplace=True, axis=1)
     df4
```

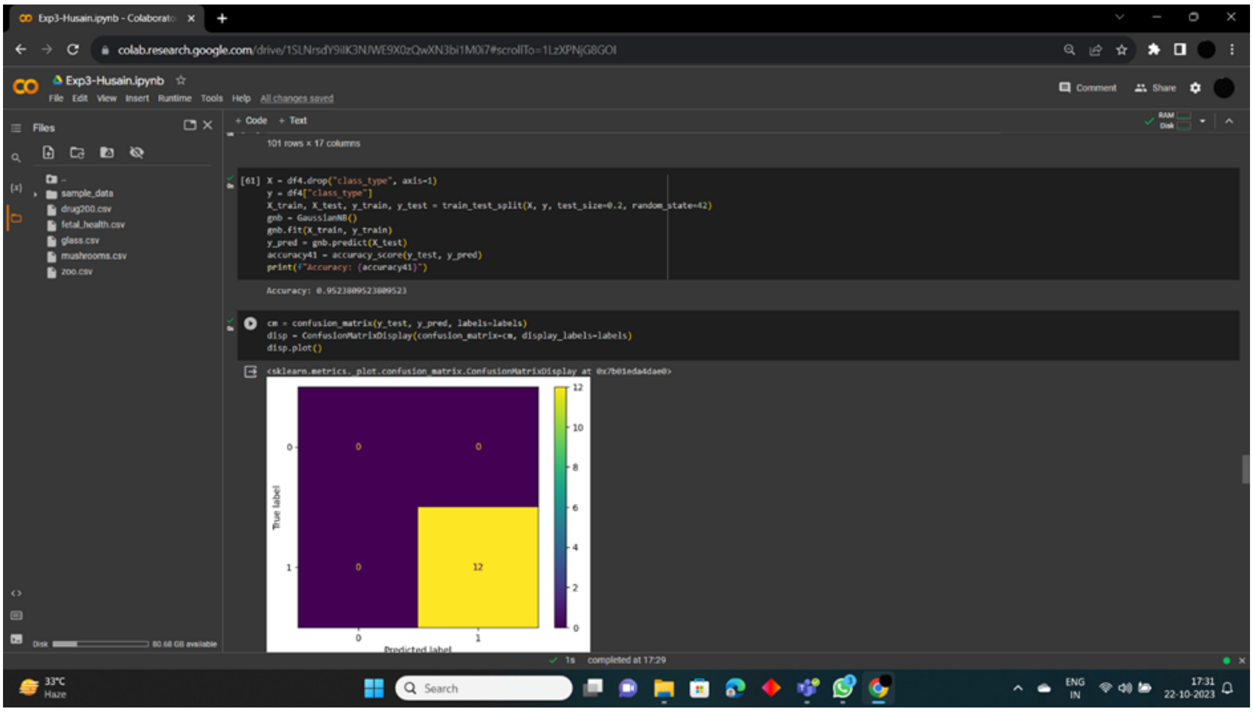
	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tail	domestic	catsize	class_type
0	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	1
1	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1
2	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	4
3	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	1
4	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
96	1	0	0	1	0	0	0	1	1	1	0	0	2	1	0	1	1
97	1	0	1	0	1	0	0	0	0	1	1	0	6	0	0	0	6
98	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1

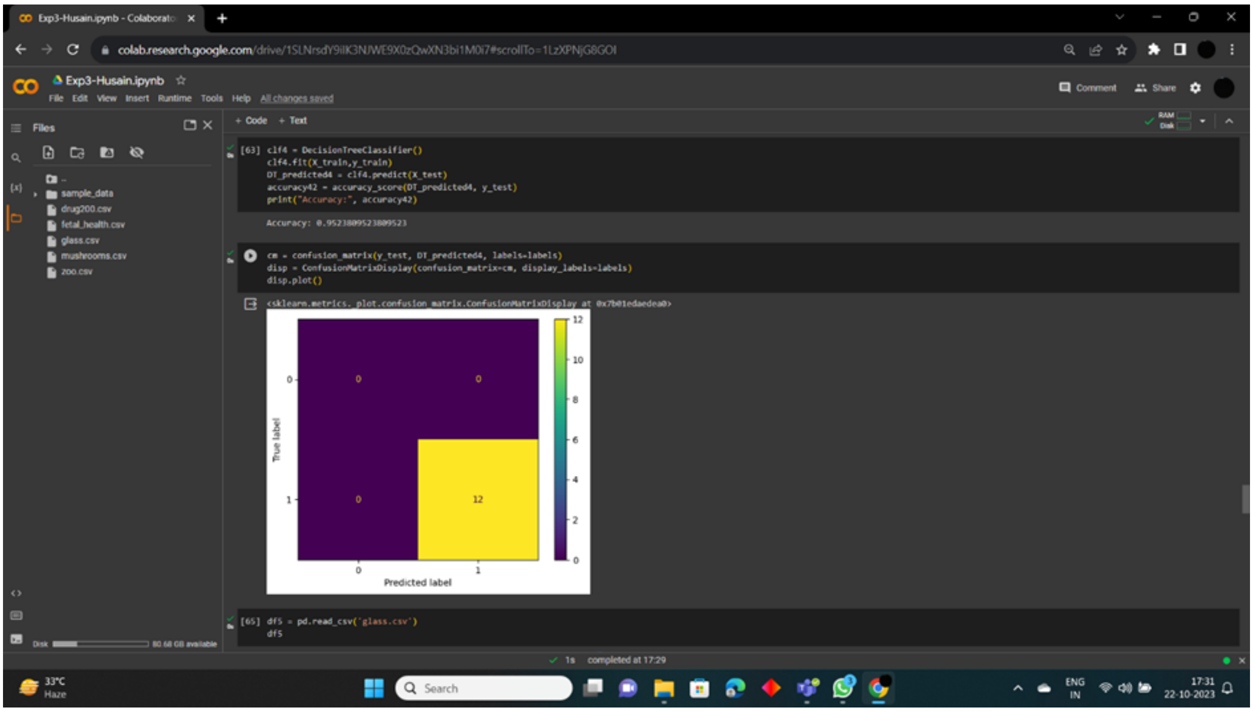


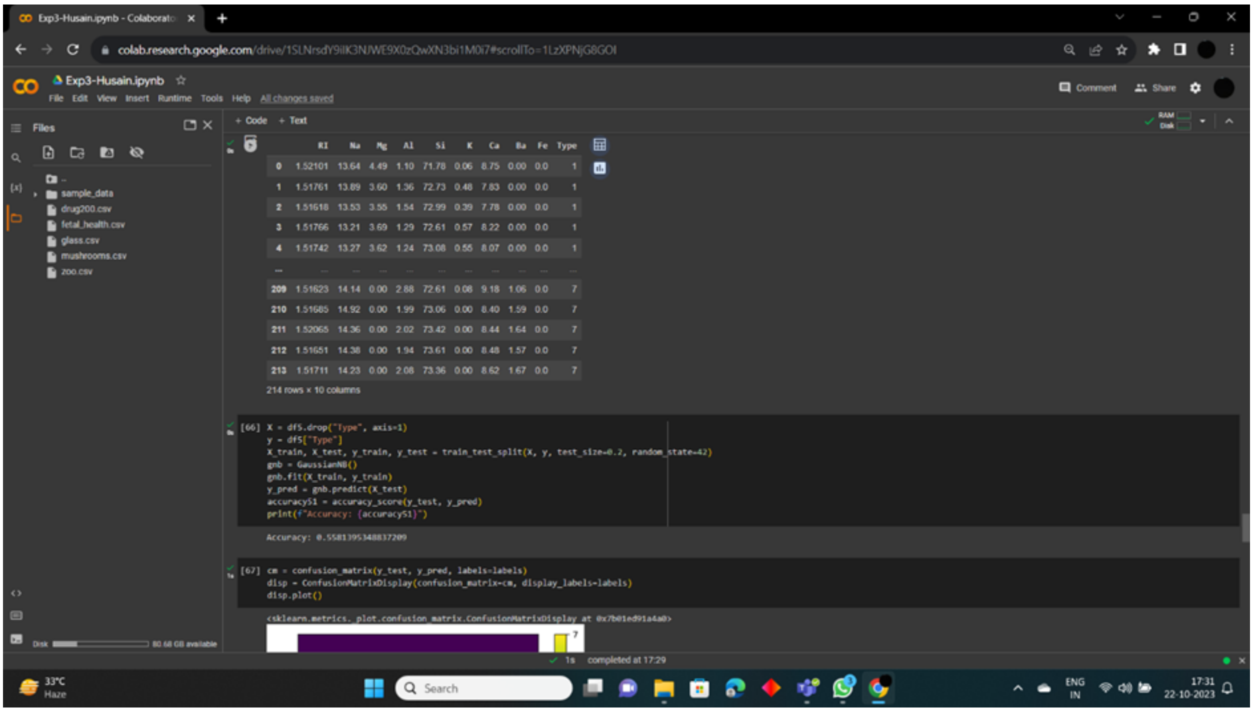
Exp3-Husain.ipynb
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7b01edbfba00>
True label
Predicted label
266
[58] clf3 = DecisionTreeClassifier()
clf3.fit(X_train,y_train)
DT_predicted2 = clf3.predict(X_test)
accuracy32 = accuracy_score(DT_predicted2, y_test)
print("Accuracy:", accuracy32)
Accuracy: 0.92018779342723
[59] cm = confusion_matrix(y_test, DT_predicted2, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
disp.plot()
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7b01edab5a20>
completed at 17:29

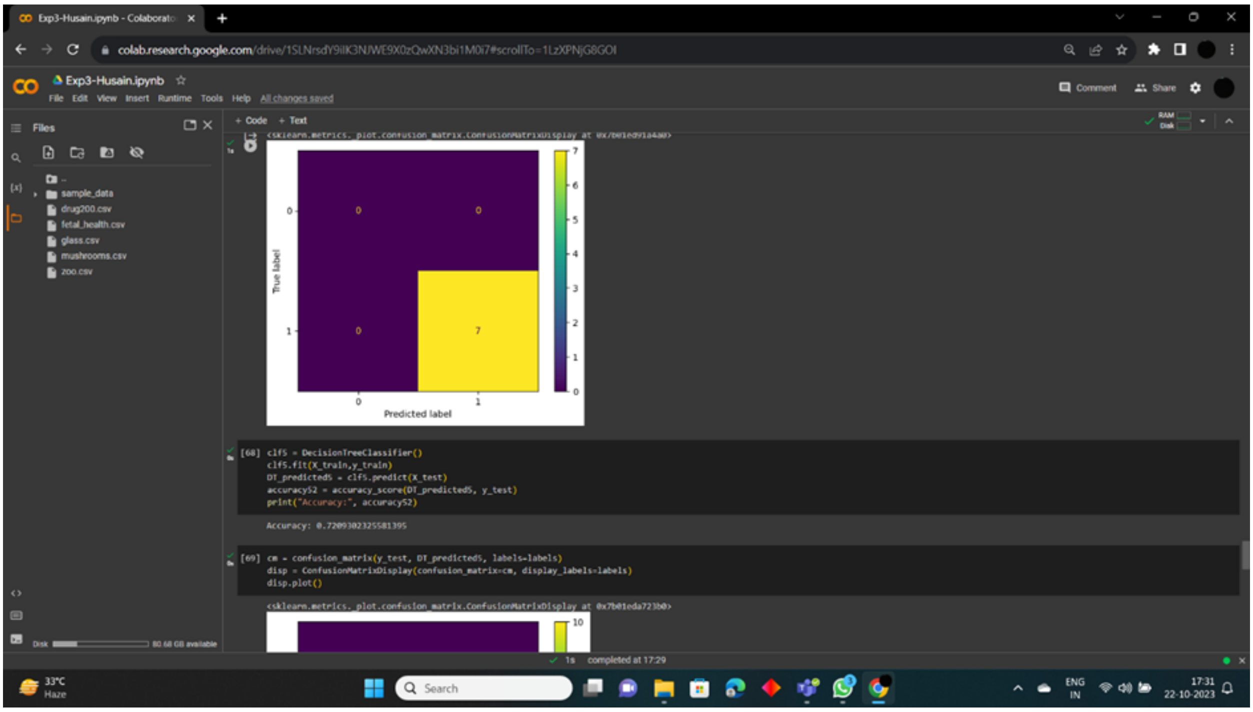


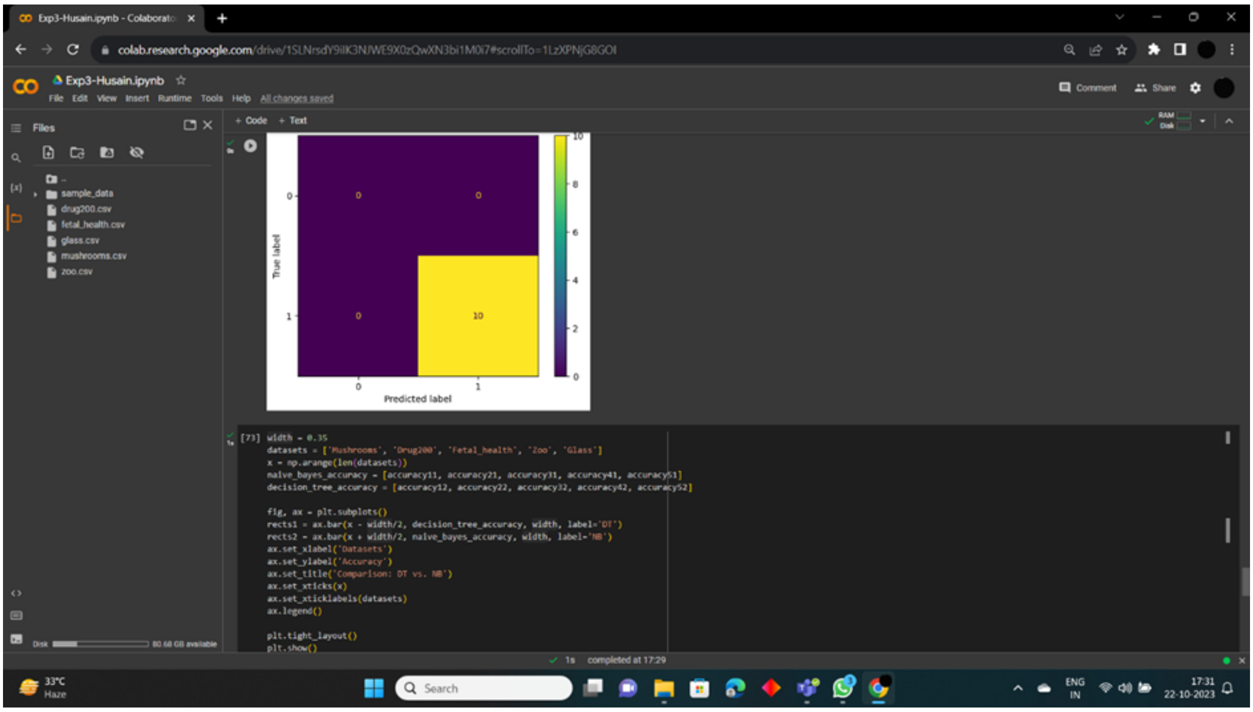
Exp3-Husain.ipynb
True label
Predicted label
314
[60] df4 = pd.read_csv('zoo.csv')
df4.drop('animal_name', inplace=True, axis=1)
df4
hair feathers eggs milk airborne aquatic predator toothed backbone breathes venomous fins legs tail domestic catsize class_type
0 1 0 0 1 0 0 1 1 1 1 0 0 4 0 0 1 1
1 1 0 0 1 0 0 0 1 1 1 0 0 4 1 0 1 1
2 0 0 1 0 0 1 1 1 1 0 0 1 0 1 0 0 4
3 1 0 0 1 0 0 1 1 1 1 0 0 4 0 0 1 1
4 1 0 0 1 0 0 1 1 1 1 0 0 4 1 0 1 1
96 1 0 0 1 0 0 0 1 1 1 0 0 2 1 0 1 1
97 1 0 1 0 1 0 0 0 0 1 1 0 6 0 0 0 6
98 1 0 0 1 0 0 1 1 1 1 0 0 4 1 0 1 1
completed at 17:29

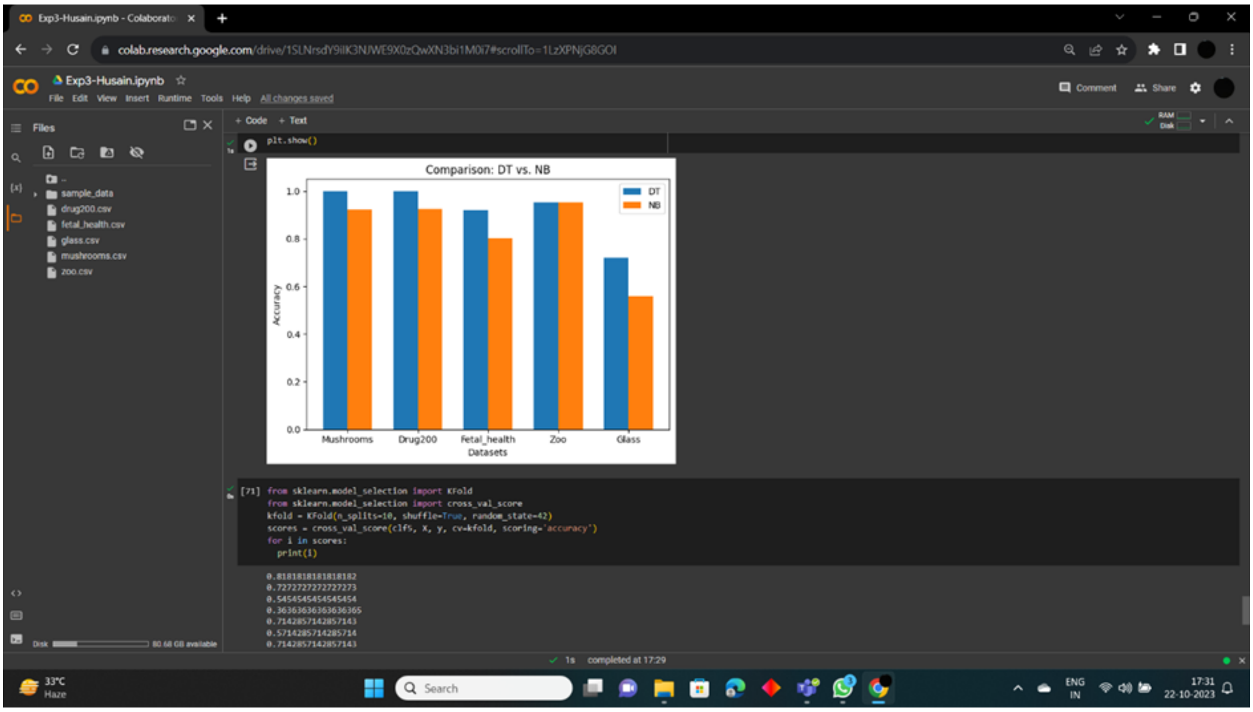

Exp3-Husain.ipynb
101 rows × 17 columns
[61] X = df4.drop("class_type", axis=1)
y = df4["class_type"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
accuracy41 = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy41}")
Accuracy: 0.9523809523809523
cm = confusion_matrix(y_test, y_pred, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
disp.plot()
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7b01eda4dae0>
True label
Predicted label

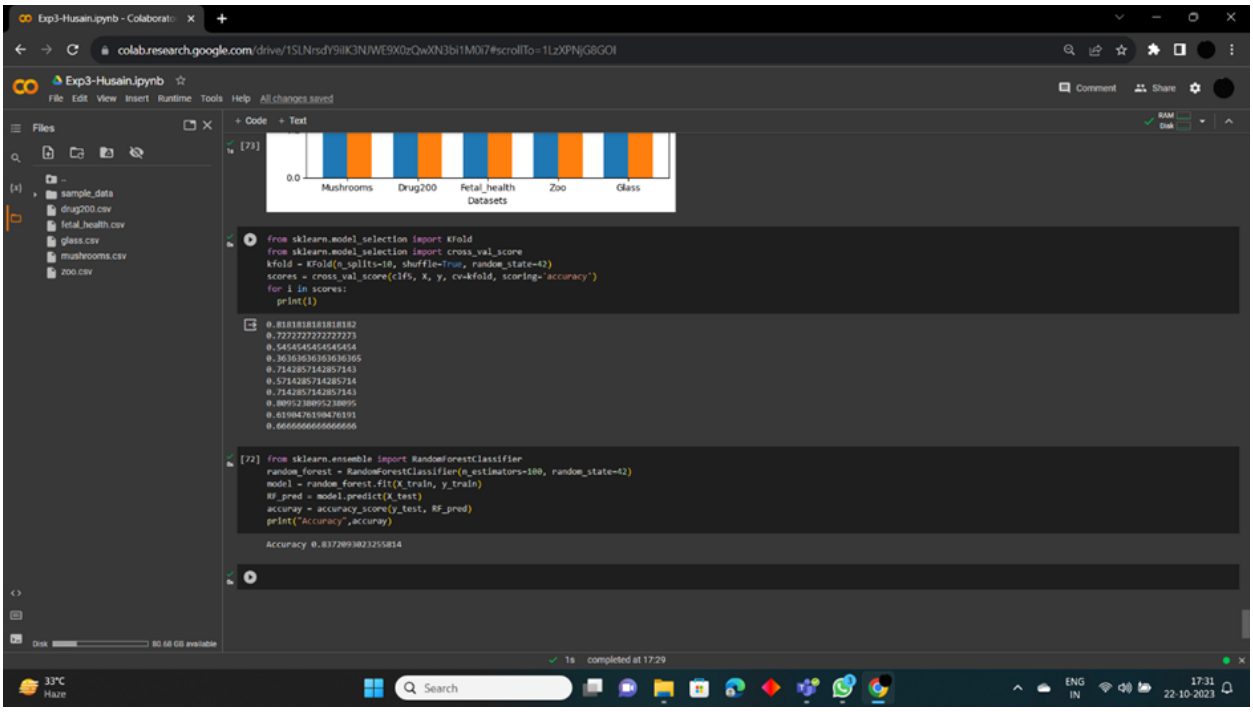

Exp3-Husain.ipynb
[63] clf4 = DecisionTreeClassifier()
clf4.fit(X_train,y_train)
DT_predicted4 = clf4.predict(X_test)
accuracy42 = accuracy_score(DT_predicted4, y_test)
print("Accuracy:", accuracy42)
Accuracy: 0.9523809523809523
cm = confusion_matrix(y_test, DT_predicted4, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
disp.plot()
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7b01edaedea0>
True label
Predicted label
[65] df5 = pd.read_csv('glass.csv')
df5

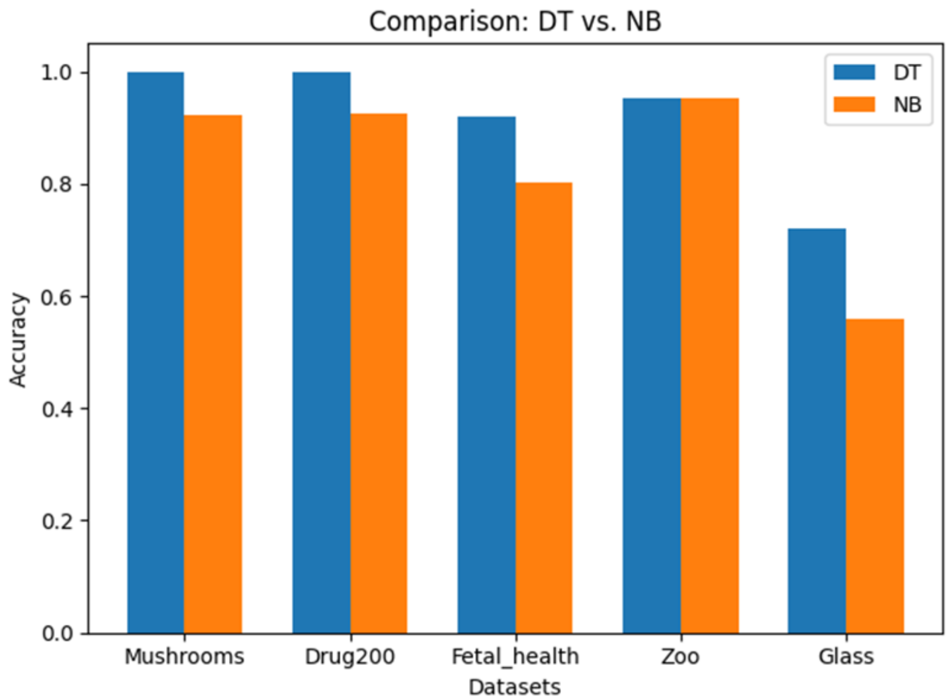

Exp3-Husain.ipynb - Colaborato
colab.research.google.com/drive/1SLNrsdY9iIK3NJWE9X0zQwXN3bi1M0i7#scrollTo=1LzXPNjG8GOI
Exp3-Husain.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
sample_data
drug200.csv
fetal_health.csv
glass.csv
mushrooms.csv
zoo.csv
+ Code + Text
RI Na Mg Al Si K Ca Ba Fe Type
0 1.52101 13.64 4.49 1.10 71.78 0.06 8.75 0.00 0.0 1
1 1.51761 13.89 3.60 1.36 72.73 0.48 7.83 0.00 0.0 1
2 1.51618 13.53 3.55 1.54 72.99 0.39 7.78 0.00 0.0 1
3 1.51766 13.21 3.69 1.29 72.61 0.57 8.22 0.00 0.0 1
4 1.51742 13.27 3.62 1.24 73.08 0.55 8.07 0.00 0.0 1
...
209 1.51623 14.14 0.00 2.88 72.61 0.08 9.18 1.06 0.0 7
210 1.51685 14.92 0.00 1.99 73.06 0.00 8.40 1.59 0.0 7
211 1.52065 14.36 0.00 2.02 73.42 0.00 8.44 1.64 0.0 7
212 1.51651 14.38 0.00 1.94 73.61 0.00 8.48 1.57 0.0 7
213 1.51711 14.23 0.00 2.08 73.36 0.00 8.62 1.67 0.0 7
214 rows × 10 columns
[66] X = df5.drop("Type", axis=1)
y = df5["Type"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
accuracy51 = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy51}")
Accuracy: 0.5581395348837209
[67] cm = confusion_matrix(y_test, y_pred, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
disp.plot()
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7b01ed91a4a0>
1s completed at 17:29
33°C Haze
Search
ENG IN
17:31 22-10-2023


Exp3-Husain.ipynb - Colaborato
colab.research.google.com/drive/1SLNrsdY9iIK3NJWE9X0zQwXN3bi1M0i7#scrollTo=1LzXPNjG8GOI
Exp3-Husain.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
sample_data
drug200.csv
fetal_health.csv
glass.csv
mushrooms.csv
zoo.csv
+ Code + Text
True label
Predicted label
0 0
0 7
[68] clf5 = DecisionTreeClassifier()
clf5.fit(X_train,y_train)
DT_predicted5 = clf5.predict(X_test)
accuracy52 = accuracy_score(DT_predicted5, y_test)
print("Accuracy:", accuracy52)
Accuracy: 0.7209302325581395
[69] cm = confusion_matrix(y_test, DT_predicted5, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
disp.plot()
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7b01eda723b0>
1s completed at 17:29
33°C Haze
Search
ENG IN
17:31 22-10-2023


Exp3-Husain.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
sample_data
drug200.csv
fetal_health.csv
glass.csv
mushrooms.csv
zoo.csv
True label
Predicted label
[73] width = 0.35
datasets = ['Mushrooms', 'Drug200', 'Fetal_health', 'Zoo', 'Glass']
x = np.arange(len(datasets))
naive_bayes_accuracy = [accuracy11, accuracy21, accuracy31, accuracy41, accuracy51]
decision_tree_accuracy = [accuracy12, accuracy22, accuracy32, accuracy42, accuracy52]
fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, decision_tree_accuracy, width, label='DT')
rects2 = ax.bar(x + width/2, naive_bayes_accuracy, width, label='NB')
ax.set_xlabel('Datasets')
ax.set_ylabel('Accuracy')
ax.set_title('Comparison: DT vs. NB')
ax.set_xticks(x)
ax.set_xticklabels(datasets)
ax.legend()
plt.tight_layout()
plt.show()
1s completed at 17:29
33°C Haze
17:31
22-10-2023


Exp3-Husain.ipynb
plt.show()
Comparison: DT vs. NB
DT
NB
Accuracy
Mushrooms
Drug200
Fetal_health
Zoo
Glass
Datasets
[71] from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
kfold = KFold(n_splits=10, shuffle=True, random_state=42)
scores = cross_val_score(clf5, X, y, cv=kfold, scoring='accuracy')
for i in scores:
  print(i)
0.8181818181818182
0.7272727272727273
0.5454545454545454
0.36363636363636365
0.7142857142857143
0.5714285714285714
0.7142857142857143
1s completed at 17:29
33°C Haze
17:31
22-10-2023





Thus it is seen from the comparison graph that Decision Tree classifier generally has a better performance than Naïve Bayes classifier for our chosen datasets.

<u>Conclusion:</u> In conclusion, classification is a fundamental task in machine learning, with Naïve Bayes and ID3 representing two distinct approaches. Naïve Bayes relies on probabilistic principles and conditional independence assumptions, while on the other hand, ID3 builds decision trees based on information gain, making it suitable for discrete data and interpretable models. The choice between these classifiers depends on the specific characteristics of the data and the goals of the classification task, highlighting the importance of understanding the underlying principles to select the most appropriate algorithm.