

ICP5

UshaKiran Yadav Avula

700746114

Program1:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report

#using pandas importing the glass csv file
df = pd.read_csv('glass.csv')
X_train, X_test, Y_train, Y_test = train_test_split(df.drop("Type", axis=1),
df["Type"], test_size=0.2)

# train model using the Naive_bayes
model = GaussianNB()
model.fit(X_train, Y_train)

# evaluating the model
score = model.score(X_test, Y_test)
print('Accuracy: %.3f' % score)

# generating the classification report
y_pred = model.predict(X_test)
report = classification_report(y_true=Y_test, y_pred=y_pred)
print(report)
```

Output:

The screenshot shows a VS Code editor with a file named Program1.py. The code imports pandas, sklearn.model\_selection, sklearn.naive\_bayes, and sklearn.metrics. It reads a 'glass.csv' file, splits it into training and testing sets, trains a Gaussian Naive Bayes model, evaluates it, and generates a classification report. The terminal output shows the accuracy as 0.395 and a detailed classification report table.

```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.naive_bayes import GaussianNB
4 from sklearn.metrics import classification_report
5
6 #using pandas importing the glass csv file
7 df = pd.read_csv('glass.csv')
8 X_train, X_test, Y_train, Y_test = train_test_split(df.drop("Type", axis=1), df["Type"], test_size=0.2)
9
10 # train model using the Naive_bayes
11 model = GaussianNB()
12 model.fit(X_train, Y_train)
13
14 # evaluating the model
15 score = model.score(X_test, Y_test)
16 print('Accuracy: %.3f' % score)
17
18 # generating the classification report
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ngel Reddy\OneDrive\Documents\Usha Kiran\ICP5\Program1.py  
Accuracy: 0.395

	precision	recall	f1-score	support
1	0.29	0.70	0.41	10
2	0.50	0.10	0.16	21
3	0.14	0.33	0.20	3
5	0.50	0.33	0.40	3
6	1.00	1.00	1.00	1
7	1.00	1.00	1.00	5
accuracy			0.40	43
macro avg	0.57	0.58	0.53	43
weighted avg	0.50	0.40	0.36	43

PS C:\Users\Angel Reddy\OneDrive\Documents\Usha Kiran\ICP5> cd 'c:\Users\Angel Reddy\OneDrive\Documents\Usha Kiran\ICP5'; & 'C:\Users\Angel R

## Program2:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import LinearSVC
from sklearn.metrics import classification_report

# using pandas importing the dataset
df = pd.read_csv('glass.csv')
X_train, X_test, Y_train, Y_test = train_test_split(df.drop("Type", axis=1), df["Type"], test_size=0.2)

# train model using linear support vector machine
model = LinearSVC(dual=False)
model.fit(X_train, Y_train)

# evaluating the model
score = model.score(X_test, Y_test)
print('Accuracy: %.3f' % score)

# generating the classification report
y_pred = model.predict(X_test)
```

```
report = classification_report(y_true=Y_test, y_pred=y_pred)
print(report)
```

Output:

The screenshot shows a VS Code editor with a Python script named 'Program2.py' open. The script imports pandas, sklearn.model\_selection, sklearn.svm, and sklearn.metrics. It reads a CSV file 'glass.csv', splits it into training and testing sets, and trains a LinearSVC model. The output in the terminal shows the accuracy of the model and a classification report.

```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.svm import LinearSVC
4 from sklearn.metrics import classification_report
5
6
7 # using pandas importing the dataset
8 df = pd.read_csv('glass.csv')
9 X_train, X_test, Y_train, Y_test = train_test_split(df.drop("Type", axis=1), df["Type"], test_size=0.2)
10
11 # train model using linear support vector machine
```

Accuracy: 0.512

C:\Users\Angel Reddy\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11\_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\sklearn\metrics\\_classification.py:1469: 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.

precision recall f1-score support

1	0.53	0.53	0.53	17
2	0.50	0.60	0.55	15
3	0.00	0.00	0.00	5
5	0.33	0.50	0.40	2
6	1.00	0.50	0.67	2
7	0.50	1.00	0.67	2
accuracy				
macro avg	0.48	0.52	0.47	43

GitHub: