

# NEURAL NETWORK & DEEP LEARNING

## ASSIGNMENT 5

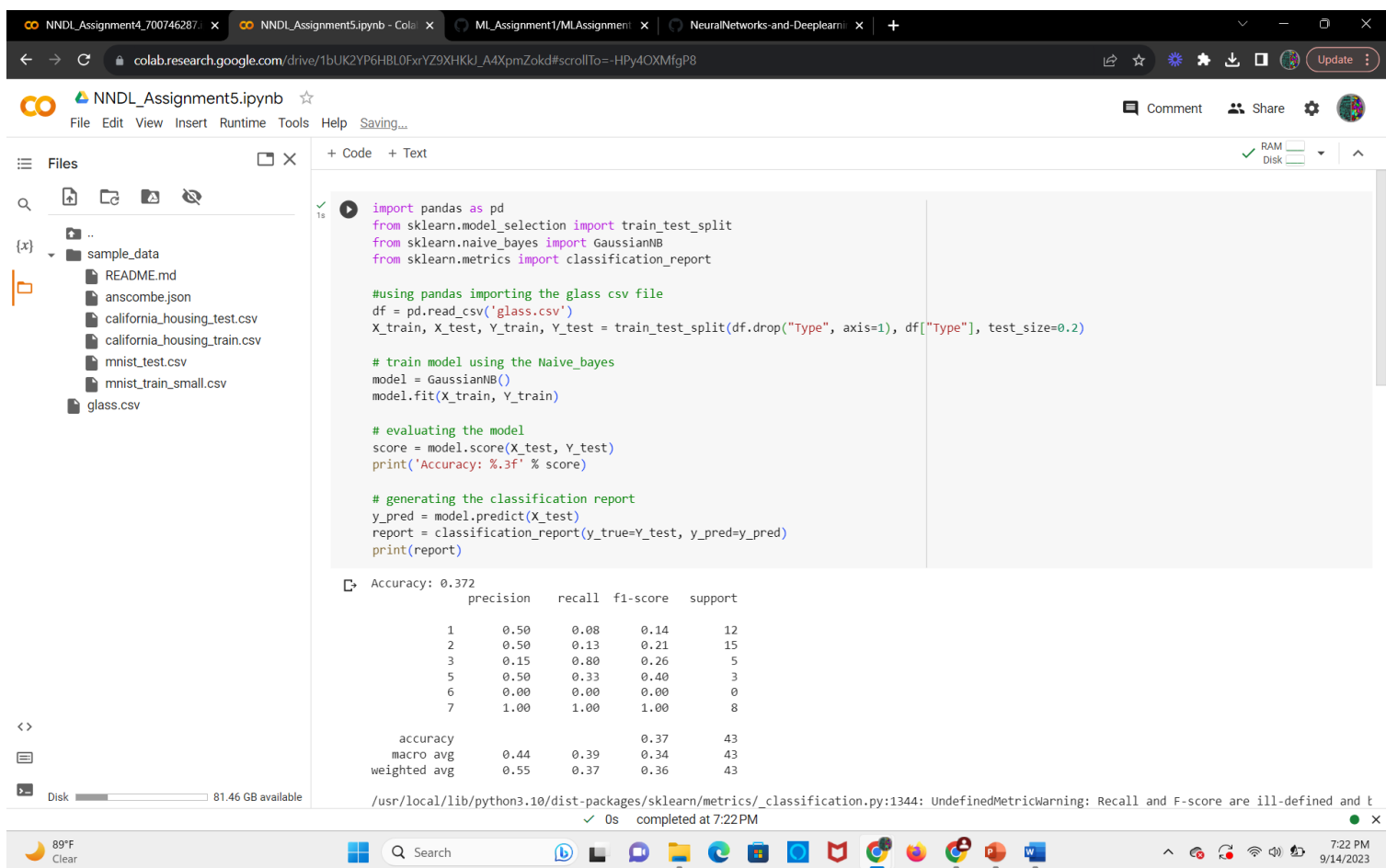
Name : SAI SNUSHA NAKKA

Student ID : 700746287

Git hub Link: [https://github.com/NSnusha/NNDL\\_Assignment5](https://github.com/NSnusha/NNDL_Assignment5)

Video link: [https://drive.google.com/file/d/1h-frN\\_pXtwlYzD9dzMMU2b9eNwNm8AYB/view?usp=sharing](https://drive.google.com/file/d/1h-frN_pXtwlYzD9dzMMU2b9eNwNm8AYB/view?usp=sharing)

1. . Implement Naïve Bayes method using scikit-learn library Use dataset available with name glass Use train\_test\_split to create training and testing part Evaluate the model on test part using score and classification\_report(y\_true, y\_pred)



The screenshot shows a Google Colab notebook titled "NNDL\_Assignment5.ipynb". The code in the notebook is as follows:

```
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)
```

The output of the code shows the accuracy and a classification report:

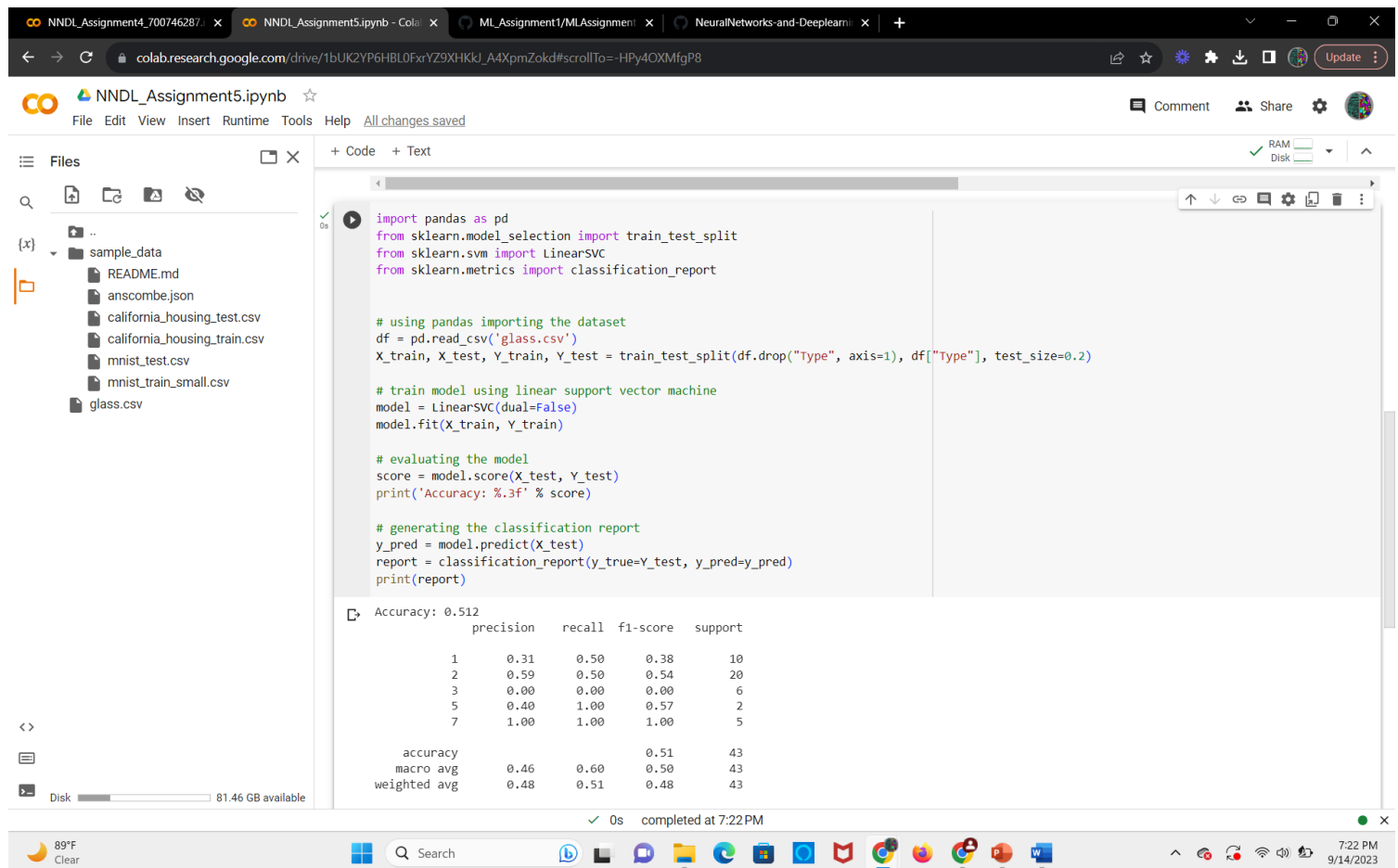
```
Accuracy: 0.372
precision    recall  f1-score   support

     1      0.50      0.08      0.14        12
     2      0.50      0.13      0.21        15
     3      0.15      0.80      0.26         5
     5      0.50      0.33      0.40         3
     6      0.00      0.00      0.00         0
     7      1.00      1.00      1.00         8

 accuracy          0.44          0.39          0.37          43
 macro avg          0.44          0.34          0.34          43
 weighted avg          0.55          0.37          0.36          43
```

The bottom of the screenshot shows a Windows taskbar with the date and time: 7:22 PM, 9/14/2023.

2. Implement linear SVM method using scikit library Use the same dataset above Use train\_test\_split to create training and testing part Evaluate the model on test part using score and classification\_report(y\_true, y\_pred)



The screenshot shows a Google Colab notebook titled "NNDL\_Assignment5.ipynb". The left sidebar displays a file explorer with a folder named "sample\_data" containing several CSV files: "README.md", "anscombe.json", "california\_housing\_test.csv", "california\_housing\_train.csv", "mnist\_test.csv", "mnist\_train\_small.csv", and "glass.csv". The main code area contains the following Python code:

```
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)
```

The output of the code is displayed below the code cell, showing the accuracy and a detailed classification report:

```
Accuracy: 0.512
```

	precision	recall	f1-score	support
1	0.31	0.50	0.38	10
2	0.59	0.50	0.54	20
3	0.00	0.00	0.00	6
5	0.40	1.00	0.57	2
7	1.00	1.00	1.00	5
accuracy			0.51	43
macro avg	0.46	0.60	0.50	43
weighted avg	0.48	0.51	0.48	43

The bottom of the notebook shows a status bar indicating "0s completed at 7:22 PM". The Windows taskbar at the very bottom shows the date and time as "7:22 PM 9/14/2023".