

Neural Networks & Deep Learning Assignment-5

Harshavardhan Reddy Patterpur
700754833

Repository Link :

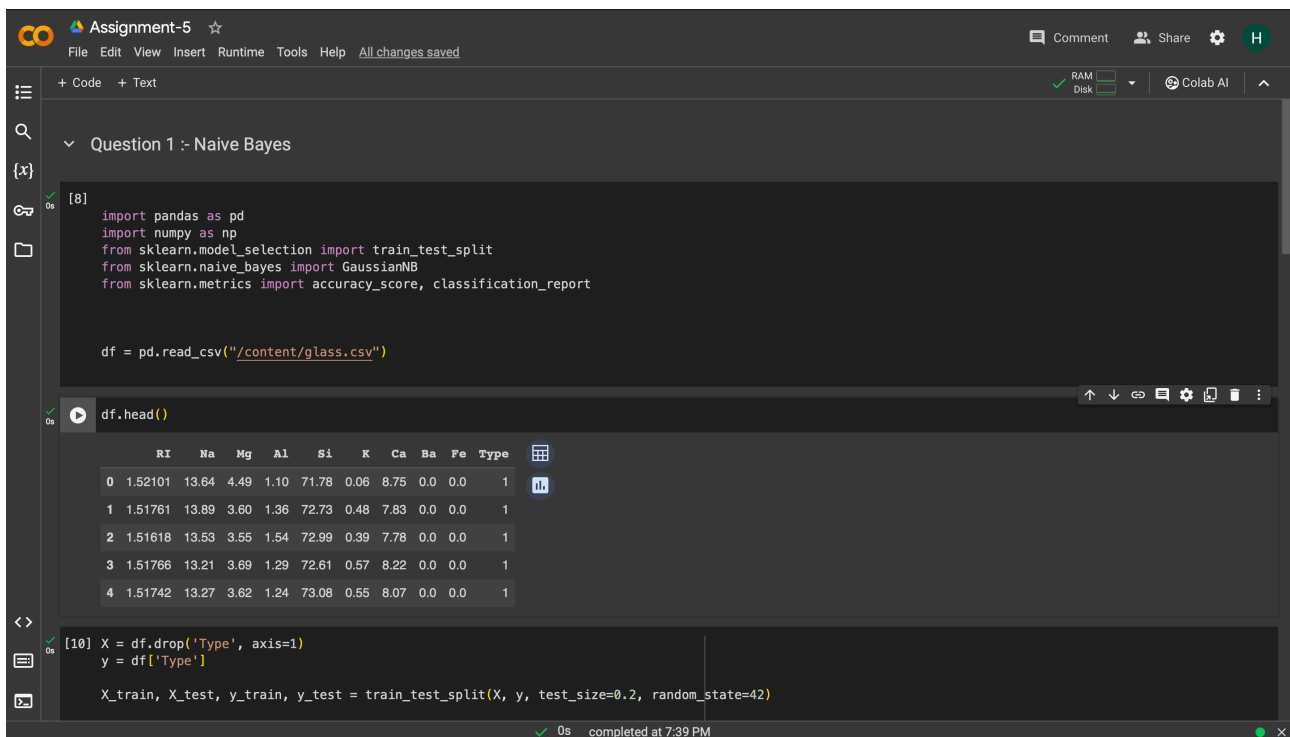
<https://github.com/harshavardhanreddy27/NNDL-Assignment-5>

Video Link:

https://drive.google.com/file/d/13ez2F4la4H5lCKAIlrQr5Lq-UIAPNfHm/view?usp=share_link

Code Screenshots:

Question:1



```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report

df = pd.read_csv("/content/glass.csv")

df.head()

[0] X = df.drop('Type', axis=1)
y = df['Type']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

	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	0.0	0.0	1
1	1.51761	13.89	3.60	1.36	72.73	0.48	7.83	0.0	0.0	1
2	1.51618	13.53	3.55	1.54	72.99	0.39	7.78	0.0	0.0	1
3	1.51766	13.21	3.69	1.29	72.61	0.57	8.22	0.0	0.0	1
4	1.51742	13.27	3.62	1.24	73.08	0.55	8.07	0.0	0.0	1

completed at 7:39 PM

Assignment-5

File Edit View Insert Runtime Tools Help All changes saved

Comment Share H

+ Code + Text

RAM Disk Colab AI

```
X = df.drop('Type', axis=1)
y = df['Type']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

naive_bayes_model = GaussianNB()

naive_bayes_model.fit(X_train, y_train)

y_pred = naive_bayes_model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
classification_report_output = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print("\nClassification Report:\n", classification_report_output)
```

Accuracy: 0.5581395348837209

Classification Report:

	precision	recall	f1-score	support
1	0.41	0.64	0.50	11
2	0.43	0.21	0.29	14
3	0.40	0.67	0.50	3
5	0.50	0.25	0.33	4
6	1.00	1.00	1.00	3
7	0.89	1.00	0.94	8
accuracy			0.56	43
macro avg	0.60	0.63	0.59	43
weighted avg	0.55	0.56	0.53	43

0s completed at 7:39 PM

Question-2

Assignment-5

File Edit View Insert Runtime Tools Help All changes saved

Comment Share H

+ Code + Text

RAM Disk Colab AI

```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
import pandas as pd

X = df.drop('Type', axis=1)
y = df['Type']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

svm_model = SVC(kernel='linear')

svm_model.fit(X_train, y_train)

y_pred_svm = svm_model.predict(X_test)

accuracy_svm = accuracy_score(y_test, y_pred_svm)
classification_report_svm = classification_report(y_test, y_pred_svm)

print("Linear SVM:")
print(f"Accuracy: {accuracy_svm}")
print("\nClassification Report:\n", classification_report_svm)
```

Linear SVM:
Accuracy: 0.7441860465116279

Classification Report:

	precision	recall	f1-score	support
1	0.69	0.82	0.75	11
2	0.67	0.71	0.69	14
3	0.00	0.00	0.00	3
5	0.00	1.00	0.00	4

0s completed at 7:39 PM

Result & Justification:

Assignment-5

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk Colab AI

[11]

Linear SVM:
Accuracy: 0.7441860465116279

Classification Report:

	precision	recall	f1-score	support
1	0.69	0.82	0.75	11
2	0.67	0.71	0.69	14
3	0.00	0.00	0.00	3
5	0.80	1.00	0.89	4
6	1.00	0.67	0.80	3
7	0.88	0.88	0.88	8
accuracy			0.74	43
macro avg	0.67	0.68	0.67	43
weighted avg	0.70	0.74	0.72	43

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to nan due to no samples predicted for class 3. Please use precision_recall_fscore_support with zero_division='warn'.

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to nan due to no samples predicted for class 3. Please use precision_recall_fscore_support with zero_division='warn'.

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to nan due to no samples predicted for class 3. Please use precision_recall_fscore_support with zero_division='warn'.

Justification

SVM algorithm has more accuracy (74%) than naive bayes Algorithm(55%) because SVM handles the small datasets very well and its robustness to handle outliers and noise makes SVM more accurate as compared to naive bayes

0s completed at 7:39 PM