

Git Hub: https://github.com/HemanthLakkimsetti76/NNDL_Assignment5

Video:

<https://1drv.ms/v/c/3e414b35cd23de20/EYp1N08xEBxEpXxSeGyWWgUBg2AhFRsgWU07HkfoJv13Nw?e=BBjABH>

NNDL ASSIGNMENT 5

1. Implement the Naive Bayes method using scikit -learn library

```
[4]: #QUESTION1

#Implement the Naive Bayes method using scikit -Learn Library
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, accuracy_score

#use the dataset available with the name glass
glass_data = pd.read_csv('c://Users//keshu//.ipynb_checkpoints//NNDL_Code and Data (2)//NNDL_Code and Data//Glass.csv')

x_train = glass_data.drop("Type", axis=1)
y_train = glass_data['Type']

#use test_train_split to create training and testing part
x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.2, random_state=0)

# Train the model using the training sets
classifier = GaussianNB()
classifier.fit(x_train, y_train)
y_pred = classifier.predict(x_test)

# Classification report
c_report = classification_report(y_test, y_pred)
print(c_report)

#evaluate the model on test part using score
print("Naive Bayes accuracy is: ", (accuracy_score(y_test, y_pred))*100)
```

```
print( naive bayes accuracy is: , (accuracy_score(y_test, y_pred))*100)
```

	precision	recall	f1-score	support
1	0.19	0.44	0.27	9
2	0.33	0.16	0.21	19
3	0.33	0.20	0.25	5
5	0.00	0.00	0.00	2
6	0.67	1.00	0.80	2
7	1.00	1.00	1.00	6
accuracy			0.37	43
macro avg	0.42	0.47	0.42	43
weighted avg	0.40	0.37	0.36	43

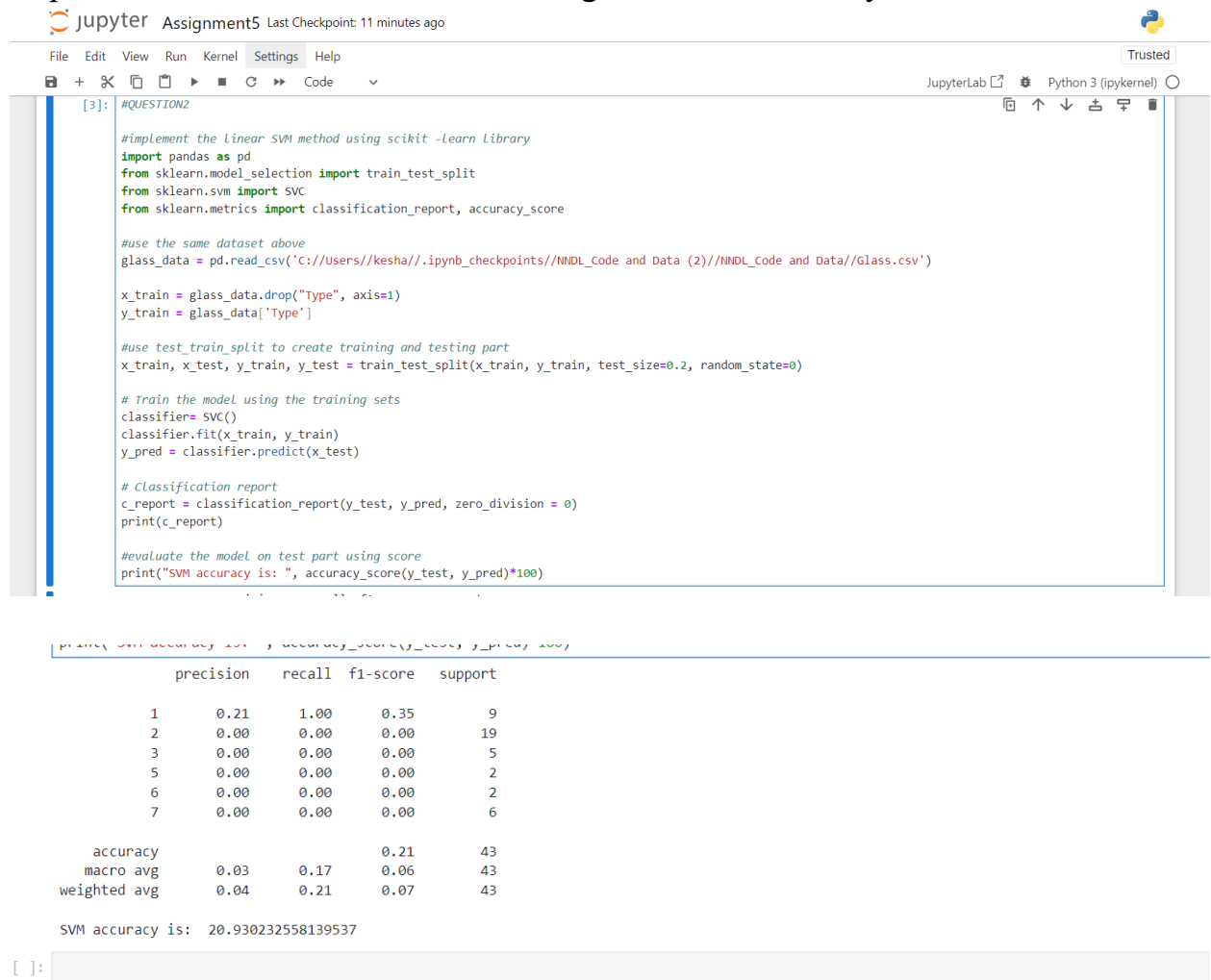
```
Naive Bayes accuracy is: 37.2093023255814
```

Git Hub: https://github.com/HemanthLakkimsetti76/NNDL_Assignment5

Video:

<https://1drv.ms/v/c/3e414b35cd23de20/EYp1N08xEBxEpXxSeGyWWgUBg2AhFRsgWU07HkfoJv13Nw?e=BBjABH>

2. Implement the linear SVM method using scikit -learn library



```
[3]: #QUESTION2

#implement the Linear SVM method using scikit -learn Library
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score

#use the same dataset above
glass_data = pd.read_csv('C:/Users/Kesha/.ipynb_checkpoints/NNDL_Code and Data (2)/NNDL_Code and Data/Glass.csv')

x_train = glass_data.drop("Type", axis=1)
y_train = glass_data['Type']

#use test_train_split to create training and testing part
x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.2, random_state=0)

# Train the model using the training sets
classifier= SVC()
classifier.fit(x_train, y_train)
y_pred = classifier.predict(x_test)

# Classification report
c_report = classification_report(y_test, y_pred, zero_division = 0)
print(c_report)

#evaluate the model on test part using score
print("SVM accuracy is: ", accuracy_score(y_test, y_pred)*100)
```

```
print('SVM accuracy is: ', accuracy_score(y_test, y_pred)*100)
```

	precision	recall	f1-score	support
1	0.21	1.00	0.35	9
2	0.00	0.00	0.00	19
3	0.00	0.00	0.00	5
5	0.00	0.00	0.00	2
6	0.00	0.00	0.00	2
7	0.00	0.00	0.00	6
accuracy			0.21	43
macro avg	0.03	0.17	0.06	43
weighted avg	0.04	0.21	0.07	43

```
SVM accuracy is:  20.930232558139537
```

```
[ ]:
```

Hemanth Lakkimsetti

700747872