Name: Pagadala Sahil Naidu

Subject: Neural Networks & Deep Learning

Course ID: CS-5720

Email: sxp86940@ucmo.edu

CRN: 22317

Click the ICP5 named link to access the assignment in GitHub.

https://github.com/Sahilnaidupagadala03/Neural_Networks_Deeplearning

Below is Voice over video.

https://youtu.be/3IgSbw1pbw8

Question 1

```python
In [13]: 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

         #Read csv file
         glass_data = pd.read_csv('glass.csv')

         # Split the data into training and testing sets
         x_train = glass_data.drop("Type", axis=1)
         y_train = glass_data['Type']


         x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.2, random_state=42)

         # Train the model using the training sets
         gaussian_nb = GaussianNB()
         gaussian_nb.fit(x_train, y_train)


         y_pred = gaussian_nb.predict(x_test)
         # Classification report
         Report = classification_report(y_test, y_pred)
         print(Report)
         print("NBs accuracy : ",  (accuracy_score(y_test, y_pred))*100)
```

	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

Naive Bayes accuracy is: 55.81395348837209

Question 2:

```
In [14]: 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


         glass_data = pd.read_csv('glass.csv')

         x_train = glass_data.drop("Type", axis=1)
         y_train = glass_data['Type']
         # splitting train and test data using train_test_split
         x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.2, random_state=42)

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

         # Classification report
         qual_report = classification_report(y_test, y_pred, zero_division = 0)
         print(qual_report)
         print("SVM accuracy is: ", accuracy_score(y_test, y_pred)*100)
```

```
              precision    recall  f1-score   support

           1       0.00      0.00      0.00        11
           2       0.33      1.00      0.49        14
           3       0.00      0.00      0.00         3
           5       0.00      0.00      0.00         4
           6       0.00      0.00      0.00         3
           7       0.00      0.00      0.00         8

    accuracy                           0.33        43
   macro avg       0.05      0.17      0.08        43
weighted avg       0.11      0.33      0.16        43

SVM accuracy is:  32.55813953488372
```

Accuracy of Naive Bayes: 55.81%

SVM precision: 32.56%

Naive Bayes outperforms SVM. The ratio of properly predicted instances to the total number of instances is used to determine accuracy, which is a measure of the classifier's overall correctness. In this instance, the SVM classifier only accurately predicts around 32.56% of the cases, compared to roughly 55.81% for the Naive Bayes classifier. For this reason, Naive Bayes performs more accurately on this specific dataset.