

NEURAL NETWORK ASSIGNMENT 5

NAME:JYOSHNA YARRAGUNTLA

STUDENT ID:700758848

Question-1

```
In [1]: 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

        # Load the dataset
        glass_data = pd.read_csv('glass.csv')

        # Separate features (x_train) and target variable (y_train)
        x_train = glass_data.drop("Type", axis=1)
        y_train = glass_data['Type']

        # Split the data into training and testing sets
        x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.2, random_state=0)

        # Initialize the Gaussian Naive Bayes model
        gnb = GaussianNB()

        # Train the model using the training sets
        gnb.fit(x_train, y_train)

        # Make predictions on the test set
        y_pred = gnb.predict(x_test)

        # Generate and print the classification report
        qual_report = classification_report(y_test, y_pred)
        print("Classification Report:\n", qual_report)

        # Print the accuracy score
        accuracy = accuracy_score(y_test, y_pred)
        print("Naive Bayes accuracy is: {:.2f}%".format(accuracy * 100))
```

Classification Report:

	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

QUESTION-2

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: 37.21%

```
In [2]: 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=0)
```

```
svc = SVC(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	0.21	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

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

```
In [ ]: #Which algorithm you got better accuracy? Can you justify why?

#As compared to the accuracy naive bayes is better than the svm, even though svm uses a hyperplane to s
# where naive bayes assumes independence features naive bayes is fast and efficient it is used for lar
# doesnt require large data it can handle continuous data and it is also suitable for high dimensional da
```

VEDIO LINK:

<https://drive.google.com/file/d/1bFxV2jdWRm3ThS61RLdYw-0oaWFSPT05/view?usp=sharing>