Neural Network & Deep Learning

(ICP Assignment # 5)

CS 5720

Name: Satya Ishyanth Kadali

Id: 700735513

CRN: 23259

Video link:
https://drive.google.com/file/d/1SpRvqImW5l80uC5B-XNe60wjC02RsZYo/view?usp=share_link

Github link:

https://github.com/Ishyanth/Deep-Learning-Assignments

Question 1:

Naïve Bayes

ICP Assignment-5

CS 5720 Neural Network & Deep Learning 700735513, Satya Ishyanth Kadali

Question 1

In [1]:

```python
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 glass dataset
data = pd.read_csv("glass.csv")


# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data.drop("Type", axis=1),data["Type"], test_size=0.2, random_state=

# Create a Gaussian Naive Bayes model
model = GaussianNB()

# Train the model on the training data
model.fit(X_train, y_train)

# Predict the labels on the test data
y_pred = model.predict(X_test)

# Evaluate the model using accuracy score and classification report
print("Accuracy score:", accuracy_score(y_test, y_pred))
print("\n")
print("Classification Report:\n", classification_report(y_test, y_pred))

```

```
Accuracy score: 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
```

Importing the required libraries pandas, train_test_split from sklearn.model_selection, GaussianNB from sklearn.naive_bayes, and classification_report and accuracy_score from sklearn.metrics.

Loading the glass dataset into a pandas dataframe using pd.read_csv("glass.csv").

Spliting the dataset into training and testing sets using the train_test_split function. Here, the data.drop("Type", axis=1) drops the Type column (target variable) from the data and stores the rest of the columns in X_train and X_test.

The target variable data["Type"] is stored in y_train and y_test. The test_size parameter is set to 0.2, meaning 20% of the data will be used for testing and 80% for training. The random_state parameter is set to 42 for reproducibility.

Creating a Gaussian Naive Bayes model using model = GaussianNB().

Now, Training the model on the training data using model.fit(X_train, y_train).

Predicting the labels of the test data using y_pred = model.predict(X_test).

Finally, evaluating the model using the accuracy score and classification report by printing "Accuracy score:" followed by accuracy_score(y_test, y_pred) and then printing the classification report using classification_report(y_test, y_pred).

Question 2:

Linear SVM

```
Question 2
```

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

# Load the dataset
df = pd.read_csv("glass.csv")

# Split the dataset into training and testing parts
X_train, X_test, y_train, y_test = train_test_split(df.drop("Type", axis=1), df["Type"], test_size=0.2, random_state=42)

# Train the linear SVM model
linear_svm = SVC(kernel='linear')
linear_svm.fit(X_train, y_train)

# Make predictions on the test set
y_pred = linear_svm.predict(X_test)

# Evaluate the model
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\n")
print("Classification Report:\n", classification_report(y_test, y_pred, zero_division=1))

```

```
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       1.00      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       0.84      0.68      0.67        43
weighted avg       0.77      0.74      0.72        43
```

Importing the required libraries as pandas, train_test_split, SVC, and accuracy_score from the scikit-learn library.

Loading the dataset glass.csv dataset is loaded into a pandas DataFrame as df.

The dataset is split into training and testing parts using the train_test_split function. The training data is stored in X_train and y_train, while the test data is stored in X_test and y_test.

Training the model on data using the fit() method. The SVC class is used for this purpose and the kernel parameter is set to "linear".

Making predictions on the test data using the predict() method. The predictions are stored in the y_pred variable.

Evaluating the model using the accuracy_score function and the results are stored in the accuracy variable.

Finally, evaluating the model using the accuracy score and classification report by printing "Accuracy score:" followed by accuracy_score(y_test, y_pred) and then printing the classification report using classification_report(y_test, y_pred).

3. Which algorithm you got better accuracy? Can you justify why?

The accuracy of the Linear Support Vector Machine (SVM) algorithm is better than the Naive Bayes algorithm in this example.

SVM is a discriminative algorithm which means it aims to find the boundary between the classes that separates the data points. On the other hand, Naive Bayes is a generative algorithm that models the data distribution of each class.

The accuracy of SVM can be improved by fine-tuning the hyperparameters such as the kernel, regularization parameter, and margin, while Naive Bayes has limited options for improving the accuracy.