Neural Networks & Deep Learning: ICP5

Name: Lalitha Sowjanya Kamuju
ID: 700747213

1. Implement Naïve Bayes method using scikit-learn library
Use dataset available with name glass
Use train_test_split to create training and testing part
Evaluate the model on test part using score and
classification_report(y_true, y_pred)

```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

# Load dataset
glass_data = pd.read_csv("Desktop:\\Neural networks\\NNDL_Code and Data (2)\\NNDL_Code and Data\\glass.csv")

# Separate features and target variable
X = glass_data.drop('Type', axis=1)
y = glass_data['Type']

# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Initialize Gaussian Naive Bayes classifier
nb_classifier = GaussianNB()

# Train the classifier
nb_classifier.fit(X_train, y_train)

# Predict on the testing set
y_pred = nb_classifier.predict(X_test)

# Evaluate the model
accuracy = nb_classifier.score(X_test, y_test)
#print("Accuracy:", accuracy)
#print("SVM Accuracy: {:.2f}%".format(svm_acc * 100))
print("Accuracy: {:.2f}%".format(accuracy*100))
# Print classification report
print(classification_report(y_test, y_pred))
```

```
Accuracy: 30.77%
              precision    recall  f1-score   support

           1       0.00      0.00      0.00        19
           2       0.40      0.17      0.24        23
           3       0.08      0.75      0.15         4
           5       0.33      0.17      0.22         6
           6       0.75      1.00      0.86         3
           7       0.90      0.90      0.90        10

    accuracy                           0.31        65
   macro avg       0.41      0.50      0.40        65
weighted avg       0.35      0.31      0.29        65
```

2. Implement linear SVM method using scikit library
Use the same dataset above
Use train_test_split to create training and testing part
Evaluate the model on test part using score and
classification_report(y_true, y_pred)
Which algorithm you got better accuracy? Can you justify why?

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

# Load the glass dataset
glass_data = pd.read_csv("Desktop:\\Neural networks\\NNDL_Code and Data (2)\\NNDL_Code and Data\\glass.csv")

# Split the data into features and target
X = glass_data.iloc[:, :-1].values
y = glass_data.iloc[:, -1].values

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
svm = SVC(kernel='linear')
svm.fit(X_train, y_train)

# Make predictions on the test set
svm_y_pred = svm.predict(X_test)

# Evaluate the model performance
svm_acc = accuracy_score(y_test, svm_y_pred,)
print("SVM Accuracy: {:.2f}%".format(svm_acc * 100))
print("\nSVM Classification Report:")
print(classification_report(y_test, svm_y_pred, zero_division=1))
```

```
SVM Accuracy: 74.42%

SVM 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
```

GitHub Link : https://github.com/sowjanya-kamuju/NeuralNetworksAssignment5

Video Link : https://vimeo.com/912459337/a5ef73e6eb?share=copy