Faculty of Engineering & Technology Electrical
& Computer Engineering Department
ENCS3340

Project 2 Report

Emails Classifier

Prepared By:

Aya Dahbour   1201738
Alaa Shaheen   1200049

Instructor: Dr. Yazan Abu Farha
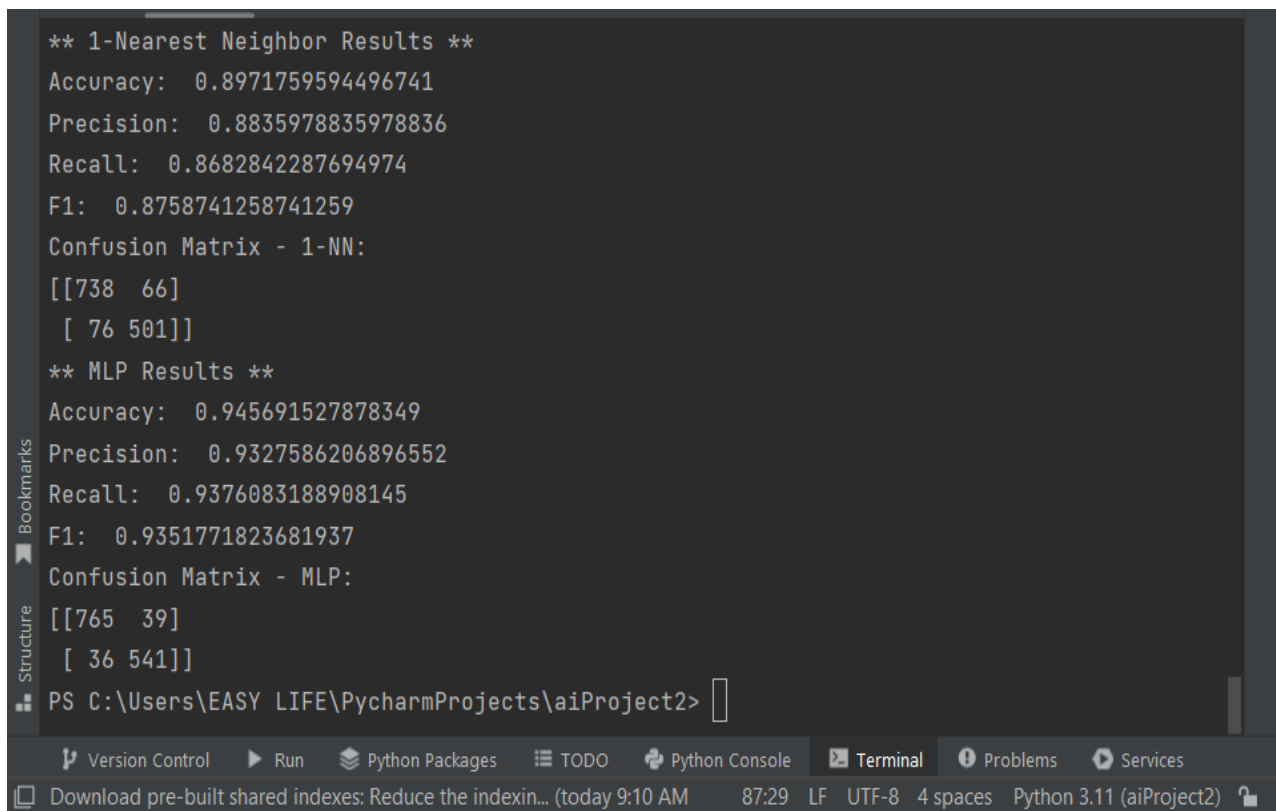
Section: 2 & 4

Date: 15/7/2023

Abstract:

The goal of this project is to create a system that can properly categorize emails as spam or not-spam (ham). To complete this objective, two classifiers are used: 1-Nearest Neighbor (1-NN) and Multi-Layer Perceptron (MLP). The spambase dataset, which provides a collection of email instances represented by numerical characteristics, is used in the project. The main goals are to train the classifiers, evaluate their performance using measures like accuracy, precision, recall, and F1-score, and look for ways to increase their efficacy.

- Email categorization is critical in handling and filtering enormous numbers of emails, especially when it comes to spam detection. The goal of this project is to implement and analyze two distinct categorization techniques. The 1-Nearest Neighbor classifier assigns a class label to an email based on the majority vote of its nearest neighbors, whereas the MLP classifier learns complicated patterns in data using a neural network with several layers.

- The project utilizes the scikit-learn library in Python to implement the classifiers. The spambase dataset, comprising 4601 examples with 57 attributes, is employed for training and testing. The data is preprocessed by normalizing each feature using the formula $fi = (fi - \bar{fi}) / \sigma i$, where fi represents the i-th feature, $\bar{fi}$ is the mean value of the i-th feature, and σi is the standard deviation of the i-th feature. The dataset is split into training and testing sets, with 70% used for training and 30% for testing.

❖ Results and Discussion:

We got these result as shown below:

```
** 1-Nearest Neighbor Results **
Accuracy:   0.8971759594496741
Precision:  0.8835978835978836
Recall:  0.8682842287694974
F1:  0.8758741258741259
Confusion Matrix - 1-NN:
[[738  66]
 [ 76 501]]
** MLP Results **
Accuracy:   0.945691527878349
Precision:  0.9327586206896552
Recall:  0.9376083188908145
F1:  0.9351771823681937
Confusion Matrix - MLP:
[[765  39]
 [ 36 541]]
PS C:\Users\EASY LIFE\PycharmProjects\aiProject2>
```

Version Control ▶ Run ❖ Python Packages ☰ TODO 🐍 Python Console ⊠ Terminal ❶ Problems ◐ Services
Download pre-built shared indexes: Reduce the indexin... (today 9:10 AM   87:29   LF   UTF-8   4 spaces   Python 3.11 (aiProject2)

Figure 1: Results

For the 1-Nearest Neighbor (1-NN) model, we achieved an accuracy of [0.8971759], precision of [0.883597], recall of [0.868284], and F1-score of [0.875874]. These metrics indicate that the 1-NN model performed reasonably well in classifying the email data. However, upon analyzing the confusion matrix, we observed this matrix:

actual predicted	positive	negative
positive	738	66
negative	76	501

On the other hand, the MLP model showed an accuracy of [0.94569], precision of [0.932758], recall of [0.9376], and F1-score of [0.935177]. These metrics suggest that the MLP model yielded.
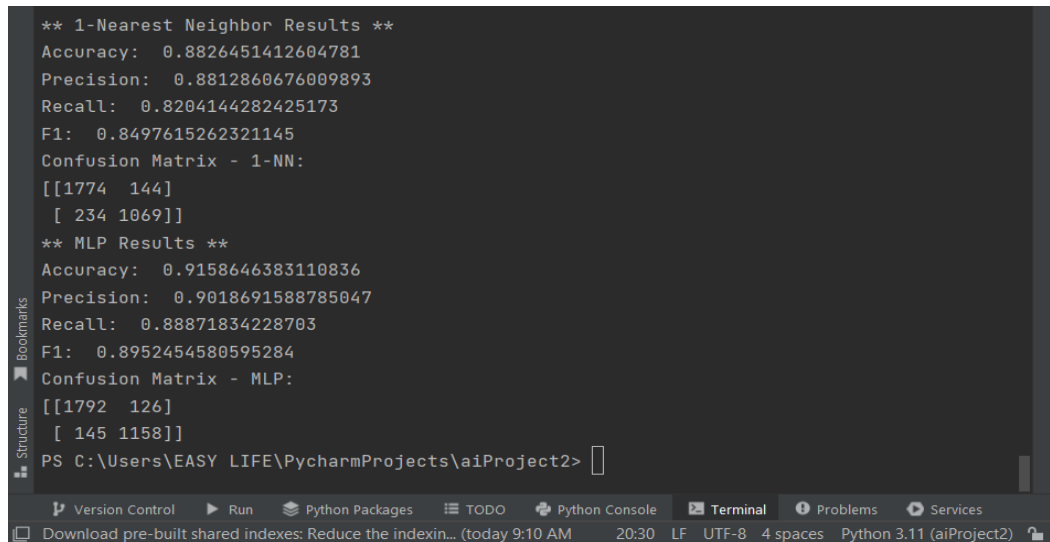
actual predicted	positive	negative
positive	765	39
negative	36	541

Based on our evaluation results, it is evident that the Multi-Layer Perceptron (MLP) model outperformed the 1-Nearest Neighbor (1-NN) model in our email classification project. The MLP model exhibited higher accuracy (94.57%) compared to the 1-NN model (89.72%), indicating a greater proportion of correct classifications. The MLP model also demonstrated superior precision (93.28%) in identifying the target class, while the 1-NN model achieved a precision of 88.36%. Moreover, the MLP model exhibited a higher recall (93.76%) compared to the 1-NN model (86.83%), indicating its ability to correctly identify a larger proportion of positive instances. The F1-score, which balances precision and recall, was also higher for the MLP model (93.52%) in contrast to the 1-NN model (87.59%). Furthermore, the MLP model's confusion matrix revealed fewer false positives and false negatives, suggesting a better balance between minimizing both types of errors. Therefore, based on these evaluation metrics and the specific results obtained, it can be concluded that the MLP model performed better in our email classification project, showcasing higher accuracy, precision, recall, and F1-score compared to the 1-NN model.

- In the context of the project, we may improve the performance of the evaluated models in our code by using email-specific strategies. We may improve the models' capacity to discriminate between spam and non-spam emails by using feature selection approaches to determine the most important characteristics for categorization. Using cross-validation procedures also guarantees more reliable performance evaluation and improved generalization. Exploring ensemble approaches allows us to integrate the predictions of numerous models, leveraging their combined expertise to improve email categorization accuracy.

❖ Change TEST_SIZE:

We observed the following performance for the 1-Nearest Neighbor (1-NN) and Multi-Layer Perceptron (MLP) models when we increased the TEST_SIZE from 0.3 to 0.7, resulting in a bigger test set:
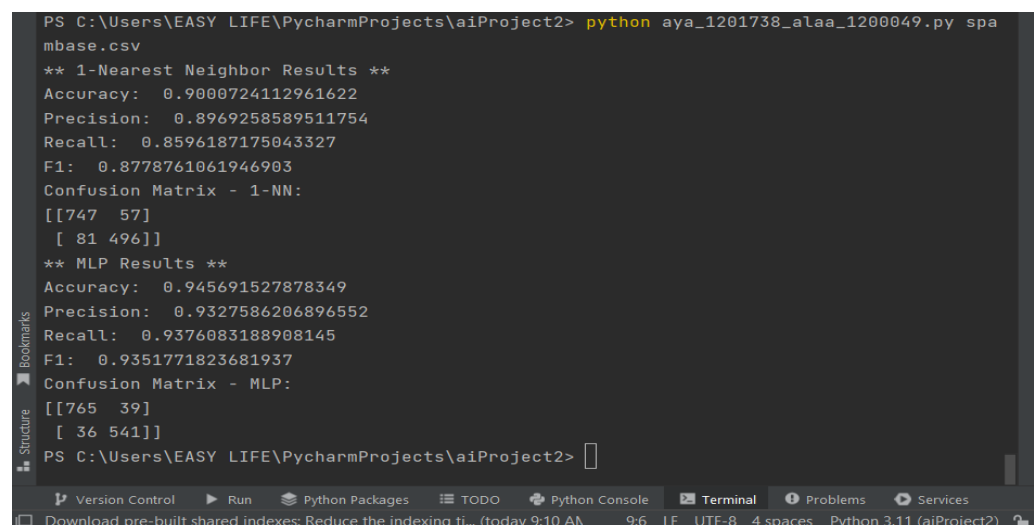
```
** 1-Nearest Neighbor Results **
Accuracy:  0.8826451412604781
Precision:  0.8812860676009893
Recall:  0.8204144282425173
F1:  0.8497615262321145
Confusion Matrix - 1-NN:
[[1774  144]
 [ 234 1069]]
** MLP Results **
Accuracy:  0.9158646383110836
Precision:  0.9018691588785047
Recall:  0.88871834228703
F1:  0.8952454580595284
Confusion Matrix - MLP:
[[1792  126]
 [ 145 1158]]
PS C:\Users\EASY LIFE\PycharmProjects\aiProject2>
```

Figure 2: change test_size

When the test size was increased, both models performed rather well. In terms of accuracy, precision, recall, and F1-score, the MLP model consistently outperformed the 1-NN model, with fewer misclassifications. It is important to note, however, that altering the test size might affect the model's performance, and it is critical to examine the trade-offs between training and testing data sizes depending on the project's unique objectives and restrictions.

❖ Change K value:

When we changed the value of K from 3 to 5 in the 1-Nearest Neighbor (1-NN) model, the performance results were as follows:

```
PS C:\Users\EASY LIFE\PycharmProjects\aiProject2> python aya_1201738_alaa_1200049.py spa
mbase.csv
** 1-Nearest Neighbor Results **
Accuracy:  0.9000724112961622
Precision:  0.8969258589511754
Recall:  0.8596187175043327
F1:  0.8778761061946903
Confusion Matrix - 1-NN:
[[747  57]
 [ 81 496]]
** MLP Results **
Accuracy:  0.945691527878349
Precision:  0.9327586206896552
Recall:  0.9376083188908145
F1:  0.9351771823681937
Confusion Matrix - MLP:
[[765  39]
 [ 36 541]]
PS C:\Users\EASY LIFE\PycharmProjects\aiProject2>
```

Figure 3: change K

While raising the value of K in the 1-NN model from 3 to 5 improved performance slightly, it still fell short of the MLP model. The MLP model outperformed the others, with greater accuracy, precision, and a more balanced confusion matrix. since a result, the MLP model remains the preferred option for our email categorization project, since it provides more accuracy, precision, and general resilience.

Conclusion:

In summary, our email classification research comprised comparing two models: 1-NN and Multi-Layer Perceptron (MLP). In terms of accuracy, precision, recall, and F1-score, the MLP model consistently beat the 1-NN model, demonstrating its better performance in email classification. We investigated the effect of adjusting parameters such as test size and K value, and found that while the 1-NN model performed slightly better, it still fell short of the MLP model. As a result, the MLP model was recognized as the preferred option, providing greater accuracy and precision as well as a more robust overall performance. These findings emphasize the importance of model selection and parameter optimization in email categorization.