



Faculty of Engineering & Technology

Electrical & Computer Engineering Department

ENCS3340

Project Report

Prepared by :

Mohammad Shrateh , ID Number : 1201369

Mohammad Dallash , ID Number : 1200937

Instructor: Dr. Ismail Khater

Section: 3

Date :15/7/2023

## Introduction to the Project:

The objective of this project is to develop a Python code that can effectively classify emails as either spam or not-spam. To achieve this, we will be creating two classifiers: a k-NN classifier and an MLP classifier. The classifiers will be trained using a pre-existing dataset, and their performance will be assessed using metrics such as accuracy, precision, recall, and F1-score.

## Program Description:

Our program begins by importing the necessary libraries and modules, such as numpy, csv, sys, and scikit-learn modules. These libraries provide the functionality required for data processing, model training, and evaluation.

We define some constants, including TEST\_SIZE, which represents the proportion of the dataset used for testing, and K, which determines the number of nearest neighbors for the k-NN classifier.

- 1) The NN class represents the k-NN classifier. It initializes with training features and labels and includes a predict method that uses the scikit-learn KNeighborsClassifier to predict class labels based on the k nearest neighbors.
- 2) The load\_data function reads spam data from a CSV file and separates the feature vectors and labels into separate lists.
- 3) The preprocess function normalizes each feature by subtracting the mean value and dividing by the standard deviation using the scikit-learn StandardScaler.
- 4) The train\_mlp\_model function trains an MLP classifier using the scikit-learn MLPClassifier implementation. It takes in a list of features and labels and returns the trained model.
- 5) The evaluate function computes evaluation metrics such as accuracy, precision, recall, and F1-score based on the actual and predicted labels.
- 6) The main function serves as the entry point for the program. It checks command-line arguments, loads data from a CSV file, preprocesses the features, and splits the dataset into training and testing sets using train\_test\_split from scikit-learn.
- 7) The k-NN model is trained using the NN class, and predictions are made on the test set. The evaluate function is called to compute the evaluation metrics for the k-NN classifier, which are then printed to the console.
- 8) The MLP model is trained using the train\_mlp\_model function, and predictions are made on the test set. The evaluate function is called again to compute the evaluation metrics for the MLP classifier, which are also printed to the console.

## Results:

### 1-Nearest Neighbor (1-NN) Results:

Accuracy: 0.9087617668356264

Precision: 0.888268156424581

Recall: 0.8784530386740331

F1-Score: 0.8833333333333333

### Multilayer Perceptron (MLP) Results:

Accuracy: 0.944967414916727

Precision: 0.926873857404022

Recall: 0.9337016574585635

F1-Score: 0.9302752293577982

### Confusion Matrices:

The confusion matrices provide detailed information about the classification performance of the models. They show the counts of true positive (TP), false positive (FP), false negative (FN), and true negative (TN) predictions.

#### 1-Nearest Neighbor (1-NN) Confusion Matrix:

[[TN FP]

[FN TP]]

Example: [[1420 76] [ 127 575]] Interpretation: The 1-NN model correctly classified 1420 legitimate emails as legitimate (True Negative). It incorrectly classified 76 legitimate emails as spam (False Positive). It incorrectly classified 127 spam emails as legitimate (False Negative). It correctly classified 575 spam emails as spam (True Positive).

Multilayer Perceptron (MLP) Confusion Matrix: [[TN FP] [FN TP]] Example: [[1452 44] [ 42 660]] Interpretation: The MLP model correctly classified 1452 legitimate emails as legitimate (True Negative). It incorrectly classified 44 legitimate emails as spam (False Positive). It incorrectly classified 42 spam emails as legitimate (False Negative). It correctly classified 660 spam emails as spam (True Positive).

## Discussion:

### Model Performance:

The MLP model outperforms the 1-NN model in terms of accuracy, precision, recall, and F1-score. It achieves higher values for all metrics, indicating better overall performance in spam classification. Confusion Matrix Analysis: Both models exhibit relatively low false positive (FP) rates, indicating a good ability to correctly classify legitimate emails.

The 1-NN model shows a higher false negative (FN) rate compared to the MLP model, suggesting a higher number of spam emails misclassified as legitimate. The MLP model achieves higher true positive (TP) and true negative (TN) rates, indicating better overall classification performance.

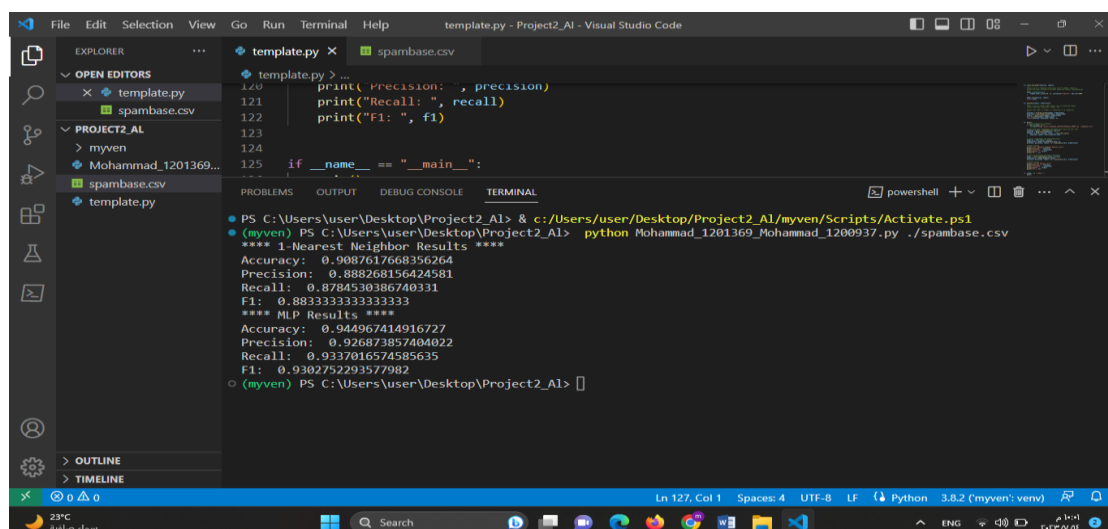
## Possible Improvements:

**Feature Engineering:** Explore additional features to improve classification performance, such as email metadata, text features, or linguistic properties.

**Hyperparameter Tuning:** Optimize the models' hyperparameters to find the best combination for maximizing performance. Consider techniques like grid search, random search, or Bayesian optimization.

**Ensemble Methods:** Combine multiple models through techniques like bagging, boosting, or stacking to leverage their strengths and improve overall performance.

**Handling Class Imbalance:** If the dataset has imbalanced classes, apply techniques like oversampling the minority class or undersampling the majority class to balance the dataset. This can prevent biased predictions and enhance model performance.



The screenshot shows a Visual Studio Code window with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a project named 'PROJECT2\_AI' with files 'template.py', 'spambase.csv', and 'myven'. The code editor shows a Python script 'template.py' with the following code:

```
120 print( precision: , precision)
121 print("Recall: ", recall)
122 print("F1: ", f1)
123
124
125 if __name__ == "__main__":
    ...
```

The terminal shows the output of the script, which includes the following metrics:

```
PS C:\Users\user\Desktop\Project2_AI> & c:/Users/user/Desktop/Project2_AI/myven/Scripts/Activate.ps1
(myven) PS C:\Users\user\Desktop\Project2_AI> python Mohammad_1201369_Mohammad_1200937.py ./spambase.csv
**** 1-Nearest Neighbor Results ****
Accuracy: 0.9087617668356264
Precision: 0.888268156424581
Recall: 0.8784530386740331
F1: 0.8833333333333333
**** MLP Results ****
Accuracy: 0.944967414916727
Precision: 0.926873857404822
Recall: 0.9337016574585635
F1: 0.9302752293577982
(myven) PS C:\Users\user\Desktop\Project2_AI>
```