Faculty of Engineering and Technology

ENCS3340

Project 2 Report

Classifying emails as spam or not-spam using MLmodels

Prepared by:

Zainab Jaradat,1201766

Manar Shawahni, 1201086

Instructor: Yazan Abu Farha.

Section: 3

Date: 20/6/2023

1. Achieved Results:

```
Processing ...
**** k-NN Results ****
Accuracy:  0.8971759594496741
Precision:  0.8835978835978836
Recall:  0.8682842287694974
F1:  0.8758741258741259
************************
**** MLP Results ****
Accuracy:  0.944967414916727
Precision:  0.9417989417989417
Recall:  0.925476603119584
F1:  0.9335664335664337
```

The achieved results show that both the k-NN and MLP classifiers perform reasonably well on the spam email classification task. The MLP classifier outperforms the k-NN classifier in terms of accuracy, precision, recall, and F1-score, indicating that it has better overall predictive performance.

2. Experimental Approach:

- k-NN Classifier:

  - Euclidean distance metric was used to find the nearest neighbors.

  - k was set to 3.

- MLP Classifier:

  - Two hidden layers were used with 10 neurons in the first layer and 5 neurons in the second layer.

  - The activation function used was the logistic (sigmoid) function.

3. Confusion Matrix:

To evaluate the performance of the classifiers in more detail, it would be beneficial to calculate the confusion matrix. The confusion matrix provides information about the true positive, true negative, false positive, and false negative predictions made by the classifiers. Unfortunately, the code you provided does not include the calculation of the confusion matrix. However, you can use scikit-learn's `confusion_matrix` function to compute it.

4. Suggestions for Improving Performance:

- For k-NN:

- Experiment with different values of k to determine the optimal number of neighbors.

- Explore alternative distance metrics other than Euclidean distance, such as Manhattan distance or cosine similarity, to see if they yield better results.

- Consider applying feature selection or dimensionality reduction techniques to reduce the dimensionality of the feature space and potentially improve performance.

- For MLP:

- Adjust the architecture of the neural network, such as increasing the number of hidden layers or neurons, to capture more complex patterns in the data.

- Tune the hyperparameters of the MLP model, including the learning rate, regularization parameters, and batch size, to find the optimal configuration.

- Gather more training data to improve the generalization capability of the model.

- Consider using other activation functions or optimization algorithms to potentially enhance performance.

- General Suggestions:

- Perform feature engineering by extracting more meaningful features from the email data, such as word frequencies, presence of certain keywords, or text-based features.

- Use ensemble techniques, such as random forests or gradient boosting, to combine the predictions of multiple models and improve overall performance.

- Apply more advanced preprocessing techniques, such as removing outliers, handling imbalanced data, or addressing missing values, to ensure the data quality.

- Consider incorporating natural language processing (NLP) techniques, such as tokenization, stemming, or TF-IDF encoding, to better represent the text-based features of the emails.

5. Conclusion:

The implemented k-NN and MLP classifiers demonstrate promising results for the spam email classification task. The MLP classifier, in particular, achieved higher accuracy and performance metrics compared to the k-NN classifier. However, there is room for further improvement by exploring alternative techniques, optimizing hyperparameters, and incorporating more advanced preprocessing and feature engineering approaches.