Faculty of Engineering and Technology

Electrical and Computer Engineering Department

ARTIFICIAL INTELLIGENCE – ENCS3340

Project # 2

Evaluating KNN and MLP Classifiers for Effective Spam Detection

---

Prepared by:

Mohammed Owda 1200089

Osaid Hamza 1200875

Instructor: Dr. Yazan Abu Farha

Section: 2

Date: 14/7/2023

Birzeit

Introduction:

As the world embraces digital communication, the challenge of managing spam emails has grown increasingly critical. Machine learning, with its potential to automate and enhance spam detection, has emerged as a potential solution to this issue. This study examines the efficiency of two distinct machine learning algorithms, K-Nearest Neighbors (KNN) and Multi-Layer Perceptron (MLP), in the classification and detection of spam emails. While KNN operates on the principle of similarity, classifying new inputs based on their proximity to known data points, MLP uses a network of interconnected layers to make predictions.

The goal of this research is to perform an in-depth comparison of these two algorithms by assessing their performance metrics—accuracy, precision, recall, and F1 score—and evaluating their confusion matrices to understand each algorithm's strengths and weaknesses in spam detection. By comparing these two diverse approaches, the study aims to discern the best suited method for spam detection and to shed light on the error-prone aspects of each algorithm.

The paper concludes by exploring potential improvements for both KNN and MLP models based on their individual performance, their underlying principles, and accepted machine learning best practices. This comparative study provides a fresh perspective on the application of machine learning in spam detection and aims to pave the way for the development of more effective spam filtering systems in the future.

Table of Contents

Methods:

The first method applied in this project is K-Nearest Neighbors (KNN). This non-parametric algorithm uses 'K' nearest data points to make classifications. Here, 'K' is a hyperparameter referring to the number of closest training examples that contribute to the prediction, which was set to 3. The algorithm operates on the principle that similar emails likely fall within the same category. When classifying a new email, the KNN algorithm looks at the categories of its K closest neighbors from the training set and assigns the most common category.

The second method utilized is the Multi-Layer Perceptron (MLP), a type of artificial neural network. This approach learns not through comparison to close neighbors, but through the learning of a series of transformations applied to the input data across multiple layers, leading to a final classification. For our MLP model, two hidden layers were applied with sizes of 10 and 5 neurons respectively, using a 'logistic' activation function. The maximum iterations for the solver were set to 2000 to ensure convergence.

Prior to the application of both algorithms, the data was preprocessed. This included standardization of features by subtracting the mean and dividing by the standard deviation, a common practice in machine learning pipelines to ensure the algorithms operate on a level playing field. After standardization, a train/test split of 70/30 was employed to assess the performance of the KNN and MLP models, providing a measure of their generalization capabilities. Performance was evaluated based on accuracy, precision, recall, and the F1 score. The confusion matrices provided further details about the classifications made by both models.

Results:

The KNN model achieved an accuracy of 90.65%, precision of 89.18%, recall of 86.71%, and F1 score of 87.93%. The confusion matrix was as follows:

Actual\Predicted	**Non-Spam**	**Spam**
Non-Spam	782 (TN)	57 (FP)
Spam	72 (FN)	470 (TP)

The MLP model achieved an accuracy of 93.98%, precision of 91.65%, recall of 93.17%, and F1 score of 92.40%. The confusion matrix was as follows:

Actual\Predicted	Non-Spam	Spam
Non-Spam	793 (TN)	46 (FP)
Spam	37 (FN)	505 (TP)

The following Screenshot show the output of the code:

```
**** 1-Nearest Neighbor Results ****
Accuracy:  0.9065894279507604
Precision:  0.8918406072106262
Recall:  0.8671586715867159
F1:  0.8793264733395697
**** MLP Results ****
Accuracy:  0.939898624185373
Precision:  0.9165154264972777
Recall:  0.9317343173431735
F1:  0.9240622140896616
KNN Confusion Matrix:
[[782  57]
 [ 72 470]]
MLP Confusion Matrix:
[[793  46]
 [ 37 505]]
```

Discussion:

The discussion on our results brings several important observations to light. Firstly, both the K-Nearest Neighbors (KNN) and Multi-Layer Perceptron (MLP) models demonstrated respectable performance on the dataset. However, the MLP classifier had a slight edge over KNN in terms of accuracy, precision, and recall. Notably, MLP showcased a higher precision rate, indicating its superior ability in reducing false positive predictions.

KNN's performance heavily depends on the chosen parameters, particularly the number of neighbors (K) and the distance metric employed. In our experimentation, we selected K as 3 and used Euclidean distance as the metric. This choice of parameters had an impact on the results obtained, and different configurations could possibly lead to varying outcomes.

Similarly, MLP's performance is dictated by several factors, including the activation function, the architecture of the hidden layers, and the number of training iterations. For our MLP model, we chose a logistic activation function, set up a two-layered hidden structure with 10 and 5 neurons respectively, and trained the model for 2000 iterations. The MLP's performance could potentially be enhanced or adjusted by altering these factors.

The comparative analysis of the two models brings valuable insights into their predictive capabilities and the influence of hyperparameters. Future work should consider hyperparameter tuning and feature selection to optimize the performance of these models. Additionally, experimenting with more advanced or diverse models could provide further improvements or interesting comparisons.

Possible Ways to Improve the Performance:

1. Feature Engineering:

Transforming raw data into a format better suited for our models can improve accuracy. For example, non-linear transformations, interaction terms, or using methods like Principal Component Analysis (PCA) to reduce dimensionality could enhance the model's performance.

2. Hyperparameter Tuning:

Hyperparameters are set before learning from the data. For KNN, this is the number of neighbors, K, and for the MLP, it's the structure of the network. Systematic tuning of these parameters using techniques like Grid Search can lead to significant performance improvements.

3. Ensembling:

Ensembling methods combine multiple model predictions to improve accuracy and robustness. An ensemble that includes our KNN and MLP models, possibly along with other models, could yield better results.

4. Advanced Models:

Exploring other machine learning models such as Support Vector Machines, Random Forests, or Gradient Boosting Machines might provide higher predictive accuracy. If ample data and computational resources are available, deep learning techniques could also offer significant improvements.

Conclusion:

In conclusion, the experiment underscored the effectiveness of both K-Nearest Neighbors (KNN) and Multi-Layer Perceptron (MLP) in tackling the spam classification problem. Even though the MLP model emerged as a slightly superior choice in this specific scenario, it is essential to acknowledge that the performance of these models can vary significantly based on the characteristics of the dataset and the configuration of hyperparameters.

KNN and MLP classifiers both have their unique advantages, and choosing between them should be informed by the specific requirements and constraints of the problem at hand. For instance, while KNN is a simple and interpretable model, MLP allows for greater complexity and can model non-linear relationships more effectively.

Looking ahead, there are multiple pathways to optimize and enhance the performance of these models. Feature engineering, for example, could reveal more potent predictors in the dataset. Hyperparameter tuning might also yield better results by more closely tailoring the models to the specific features of the data.

Furthermore, ensembling, or the practice of combining multiple models, may present an opportunity to exploit the strengths of different classifiers and potentially deliver more accurate and reliable predictions.

Finally, it could also be beneficial to explore more advanced machine learning models. Sophisticated techniques like Random Forests or Gradient Boosting machines could offer improvements in performance, particularly for more complex or high-dimensional datasets.

As the field of machine learning continues to advance rapidly, it is likely that even more powerful and efficient algorithms will be developed, providing even more tools for tackling challenging prediction problems like spam classification.