Skill Genie

Data Science & AI /ML Internship

Samuel MATIA

Machine learning model for email spam classification

Introduction

The increasing volume of unsolicited emails, commonly known as spam, poses a significant challenge for email users and organizations. Spam emails clutter inboxes, compromise security, and hinder productivity. To address this issue, there is a need to develop an effective email spam classification system that can automatically identify and filter out spam emails from legitimate ones. This report aims to document the process of creating such a system, covering data preprocessing, feature engineering, model selection, and evaluation. By providing detailed explanations of the model architecture, algorithms used, and key insights gained, this report aims to contribute to the development of an email spam classification system that enhances email security and user experience. Additionally, the report will summarize project outcomes, discuss challenges faced, and provide recommendations for future improvements.

Data overview

The provided dataset consists of 5172 rows and 3002 columns, where each column represents a word from our vocabulary, and each row corresponds to the occurrence of that word in a particular email. There are two labels, with 0 indicating a ham (non-spam) email and 1 indicating a spam email. The ham class comprises 3672 data points, while the spam class contains 1500 data points.

Data preprocessing

I separated the features and the target from the dataset, and I applied a normalization of the features with the MinMaxScaler to prevent the outliers from lowering the performance of the model and I separated the data into training data and test (80-20)

Machine Learning model

I used two models and I took the one that had the best results in the metrics :

- **SupportVectorClassifier** used with default hyperparameters. I tried adjusting the hyperparameters with GridSearchCV, but the default ones gave the best results

- **RandomForestClassifier** : with n-estimators = 100

Results and evaluation

- **SupportVectorClassifier:**

-Training score : 98.4 %
-accuracy : 95 %
-precision : 85.48 %
-recall : 97.8 %
-f1 score : 91.22 %

Results and evaluation

- **RandomForestClassifier** :

-Training score : 100 %
-accuracy : 97.49 %
-precision : 95.81 %
-recall : 95.81 %
-f1 score : 95.81 %

Deployment

I used Flask and Html-css to deploy my machine learning model, where I created a single endpoint that takes the vector encoded email and passes it to the model to make the prediction.
The encoding function takes as a parameter the vocabulary of the dataset (each column) and creates a sparse vector with the number of occurrences of each word for each component.

Challenges and Limitations

For a spam classification model, the dataset provided was very poor in terms of quantity of data, and the format used reduced the performance for the embedding layer for recurrent neural networks, hence the use of random forest.

Conclusion

In conclusion, we used the Random Forest algorithm for spam classification. Despite the small dataset, our model showed promising performance.

Random Forest proved to be a suitable choice, effectively classifying emails as spam or non-spam.

However, the main limitation was the small dataset size, which may have limited the model's ability to capture all aspects of spam variability.

To improve results, obtaining a larger and more balanced dataset is recommended. Exploring additional data sources and considering alternative feature engineering techniques and classification algorithms could also enhance performance.

In summary, while Random Forest showed promise, improving the dataset remains a priority for better performance.

Thanks

Merci