INTRODUCTION

Spam messages are a major problem for individuals and organizations worldwide. These messages can contain malware, phishing attacks, and other harmful content. Spam filters are used to protect users from unwanted emails by automatically identifying and separating them from legitimate emails. However, many spam emails still manage to bypass these filters, causing significant harm to individuals and organizations.

The objective of this research is to develop a robust SMS spam classification model using natural language processing techniques which are Multinomial Naive Bayes, Decision Trees and Random Forest. The study aims to collect a dataset of SMS messages, preprocess and clean the messages, and apply NLP techniques such as stemming, lemmatization, and tokenization for feature extraction. Different feature extraction techniques like bag-of-words and TF-IDF are used to represent the messages. The performance of three machine learning algorithms including is compared for SMS spam classification. The models are evaluated using metrics such as accuracy, precision, recall, and F1-score. Based on the evaluation, the best-performing model will be determined, which will serve as the recommended approach for SMS spam classification.

DATA SET

The dataset used for this classification task is the "*Spam SMS Data*" corpus. It consists of SMS messages labeled as spam or ham. The corpus provides a comprehensive collection of text messages, enabling us to train and evaluate our classification models effectively. The dataset includes features such as the message content and its corresponding label, which is essential for supervised learning.

The SMS Spam Collection Dataset: Source: UCI Machine Learning Repository Link: ***https://archive.ics.uci.edu/ml/datasets/sms+spam+collection***

RESOURCES USED IN THE IMPLEMENTATION

1. **Libraries**:

   numpy-Used for numerical computations in Python.

   pandas- Used for data manipulation and analysis.

   IPython.display- Used for displaying data frames in Jupyter Notebook.

   matplotlib- Used for data visualization.

   Seaborn- Used for statistical data visualization.

   Nltk- Natural Language Toolkit library for text processing.

   re- Regular expression library for pattern matching.

   sklearn- Scikit-learn library for machine learning.

2. **Dataset**

'Spam SMS Data': A CSV file containing SMS messages labeled as spam or ham (not spam).

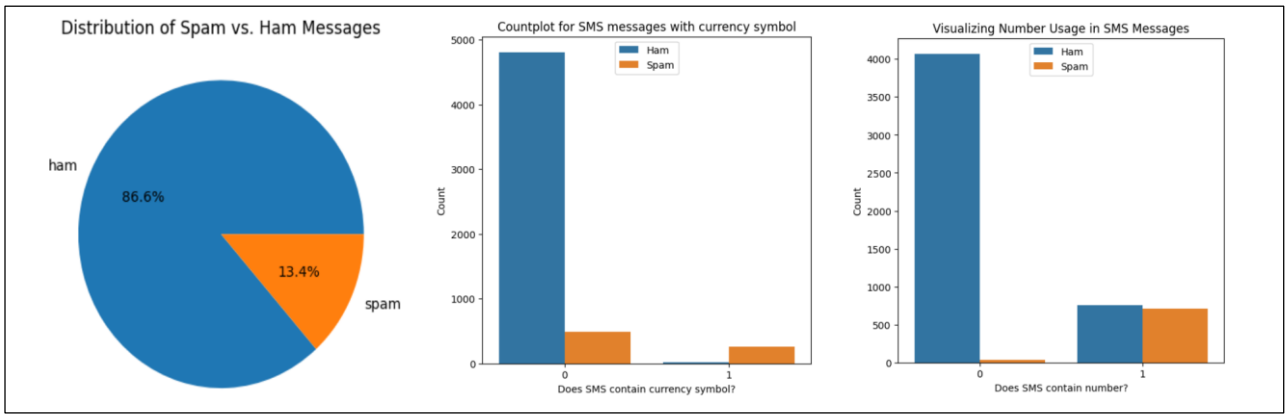
DATA VISUALIZATION

The data visualization step aims to gain insights into the distribution of messages based on their labels (spam or ham). This step is important because visualizing data can often reveal patterns and relationships that may not be immediately apparent from the raw data.

To achieve this, two specific types of visualizations are used:

Pie Chart: A pie chart is created to represent the percentage of messages for each label. In this case, the pie chart will display the proportion of spam and ham messages in the dataset. This visualization helps in understanding the overall distribution of the labels and provides a clear visual representation of the relative frequencies.

Count plot: A count plot is used to visualize the presence of currency symbols and numbers in the messages. It displays the frequency of occurrences of these specific elements in the dataset. This visualization can help in identifying patterns or trends related to the usage of currency symbols or numbers in spam or ham messages.



Also Feature engineering has been involved in this project where by new features are created from existing data to enhance model performance. In this case, two features are created: "contains_currency_symbol" (indicating if a message has a currency symbol) and "contains_number" (indicating if a message has a number). These features provide additional information for the models to better classify messages as spam or ham.

METHODOLOGY

During the model building and evaluation phase, the main focus was on training and assessing the performance of three distinct classifiers: *Multinomial Naive Bayes*, *Decision Tree*, and *Random Forest*. Each of these classifiers offers unique characteristics and advantages that make them suitable for our text classification problem.

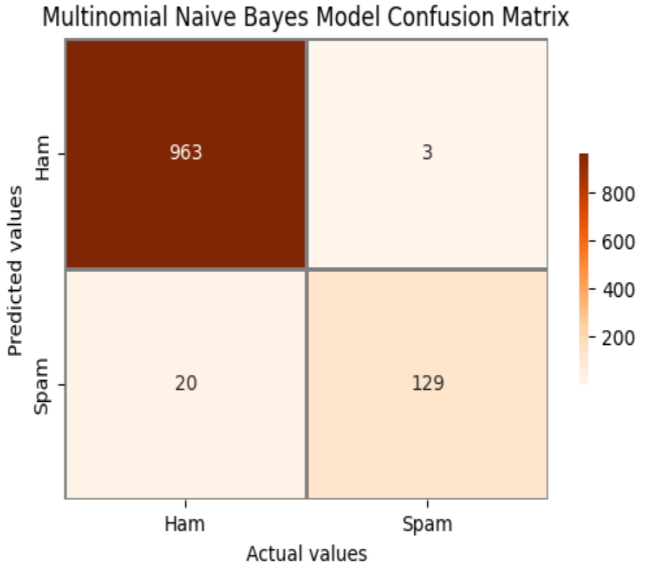
To ensure reliable performance assessment, cross-validation was employed during implementation. This technique involves dividing the dataset into multiple subsets or "folds" and iteratively training and evaluating the models on different combinations of these folds.

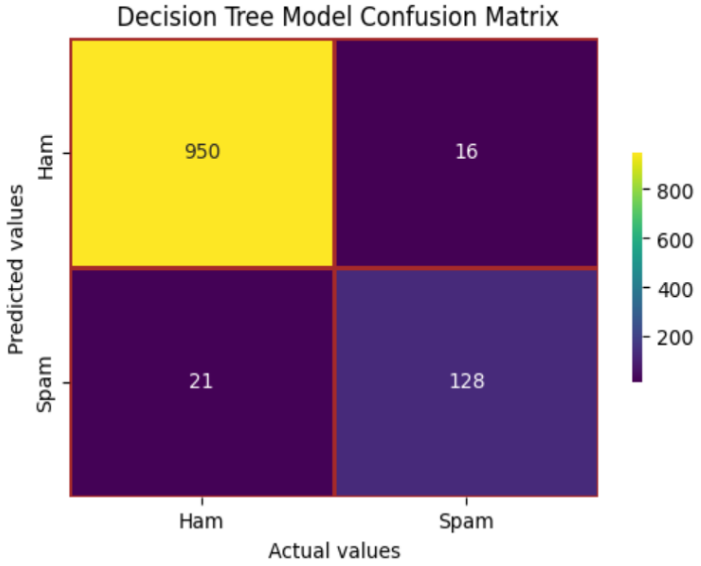
By doing so, a more robust evaluation of the classifiers' performance was obtained, as it accounts for variations in the data and helps mitigate overfitting. As part of the evaluation process, appropriate evaluation metrics was utilized, with the F1-score being one of them. The F1-score is a widely used metric for binary classification tasks that combines precision and recall into a single value. It provides a balanced measure of the classifiers' ability to correctly classify both positive and negative instances, accounting for both false positives and false negatives.

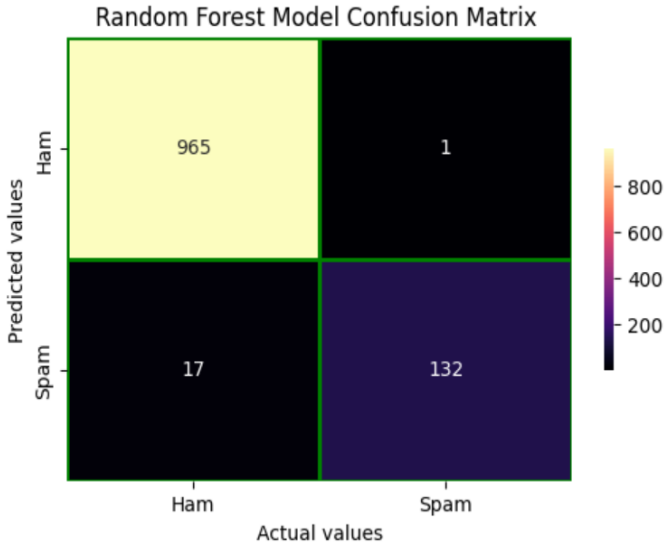
To provide a comprehensive assessment of each model's performance, classification reports and confusion matrices were also generated. The classification reports offer detailed information about various evaluation metrics, such as precision, recall, F1-score, and support, for each class (in our

case, spam and non-spam). These reports provide a holistic view of how well the classifiers performed across different metrics and help identify areas for improvement.

Additionally, confusion matrices were created to visualize the performance of the models. A confusion matrix displays the counts of true positives, true negatives, false positives, and false negatives. By examining the distribution of these values, we can gain insights into the types of errors made by the classifiers and assess their overall accuracy.







By training and evaluating the Multinomial Naive Bayes, Decision Tree, and Random Forest classifiers, performing cross-validation with appropriate evaluation metrics like the F1-score, and generating classification reports and confusion matrices, I conducted a comprehensive evaluation of each model's performance. This rigorous evaluation has made it possible to make informed decisions about the suitability of the classifiers for our text classification problem and guides further improvements in the given approach.

RESULTS

According to the project implementation, analysis and evaluation the performance of different machine learning models for spam SMS classification has been obtained. The models used in this study include Multinomial Naive Bayes (MNB), Decision Trees, and Random Forest. The evaluation is based on precision, recall, F1-score, and average F1-score metrics.

1. **Multinomial Naive Bayes Model** (MNB): The MNB model demonstrates high precision and recall for both spam and ham (non-spam) classes, with an overall accuracy of 98%. The F1-score for spam classification is 0.92, indicating a good balance between precision and recall. The average F1-score for the MNB model is 0.896, with a standard deviation of 0.015.

   The performance metrics of the Multinomial Naive Bayes (MNB) model indicate that it is effective in accurately classifying SMS messages as spam or non-spam. The high precision and recall scores for both spam and ham classes suggest that the model achieves a good balance between correctly identifying spam messages (precision) and capturing a high proportion of actual spam messages (recall). The overall accuracy of 98% further confirms the model's ability to make accurate predictions.

The F1-score for spam classification, which is 0.92, reflects the harmonic mean of precision and recall. It signifies that the MNB model achieves a good balance between correctly identifying spam messages and minimizing false positives. A higher F1-score indicates better overall performance.

The average F1-score for the MNB model is 0.896, with a standard deviation of 0.015. The average F1-score provides an overall measure of the model's performance across both spam and non-spam classes. The standard deviation indicates the variability or consistency of the model's performance across different iterations or subsets of the dataset.

Evidence from the code output

```
============================================================
Classification Report for MNB Model
============================================================
              precision    recall  f1-score   support

         ham       0.98      1.00      0.99       966
        spam       0.98      0.87      0.92       149

    accuracy                           0.98      1115
   macro avg       0.98      0.93      0.95      1115
weighted avg       0.98      0.98      0.98      1115


Average F1-Score for MNB model= 0.896
Standard Deviation= 0.015
```

2. Decision Trees Model

The Decision Trees model achieves an accuracy of 97% and shows a slightly lower F1-score for spam classification compared to the MNB model (0.87). The precision, recall, and F1-score for both classes are slightly lower than those of the MNB model. The average F1-score for the Decision Trees model is 0.869, with a standard deviation of 0.028.

The performance metrics of the Decision Trees model indicate that it is effective in classifying SMS messages as spam or non-spam but performs slightly lower compared to the MNB model. The model achieves an accuracy of 97%, which means it correctly classifies 97% of the SMS messages. However, the F1-score for spam classification is slightly lower at 0.87 compared to the MNB model, indicating that the Decision Trees model may have a slightly lower balance between precision and recall for identifying spam messages.

The precision, recall, and F1-score for both spam and non-spam classes are slightly lower compared to the MNB model. This suggests that the Decision Trees model may have slightly more false positives and false negatives compared to the MNB model.

The average F1-score for the Decision Trees model is 0.869, with a standard deviation of 0.028. The average F1-score provides an overall measure of the model's performance across both classes, and the standard deviation indicates the variability or consistency of the model's performance.

Evidence from the code output

```
============================================================
Classification Report for Decision Tree Model
============================================================
              precision    recall  f1-score   support

         ham       0.98      0.98      0.98       966
        spam       0.89      0.86      0.87       149

    accuracy                           0.97      1115
   macro avg       0.93      0.92      0.93      1115
weighted avg       0.97      0.97      0.97      1115


Average F1-Score for Decision Tree model: 0.869
Standard Deviation: 0.028
```

3. Random Forest Model

The Random Forest model performs well, achieving an accuracy of 98% and demonstrating high precision, recall, and F1-scores for both spam and ham classes. The F1-score for spam classification is 0.94, which is the highest among the evaluated models. The average F1-score for the Random Forest model is 0.910, with a standard deviation of 0.024.

The performance metrics of the Random Forest model indicate that it is highly effective in classifying SMS messages as spam or non-spam. The model achieves an impressive accuracy of 98%, meaning it correctly classifies 98% of the SMS messages. This indicates a high level of overall accuracy in predicting the correct class for each message.

The Random Forest model also demonstrates high precision, recall, and F1-scores for both spam and non-spam classes. Precision measures the proportion of correctly identified spam messages out of all the messages classified as spam. Recall measures the proportion of actual spam messages that were correctly identified. The F1-score is the harmonic mean of precision and recall, providing an overall measure of the model's performance. The F1-score for spam classification in the Random Forest model is 0.94, which is the highest among the evaluated models. This indicates a strong balance between precision and recall for identifying spam messages accurately.

Evidence from the code output

```
============================================================
Comprehensive Classification Report for Random Forest Model
============================================================
              precision    recall  f1-score   support

         ham       0.98      1.00      0.99       966
        spam       0.99      0.89      0.94       149

    accuracy                           0.98      1115
   macro avg       0.99      0.94      0.96      1115
weighted avg       0.98      0.98      0.98      1115


Average F1-Score for Random Forest Model= 0.910
Standard Deviation= 0.024

Accuracy: 0.98
Prediction: ['ham' 'spam' 'ham' 'spam' 'spam']
```

CONCLUSION

Among the evaluated models, the Random Forest model outperforms the others in terms of F1-score for spam classification, with an F1-score of 0.94. It demonstrates the highest precision, recall, and F1-scores for both spam and ham classes. The Random Forest model's superior performance can be attributed to its ensemble approach, which combines multiple decision trees to reduce overfitting and increase predictive accuracy. The model's ability to capture complex relationships in the data enhances its performance in identifying spam SMS messages accurately.

While the Multinomial Naive Bayes model also shows good performance with an F1-score of 0.92, the Random Forest model provides a slight improvement in spam classification. The Decision Trees model, although performing well, falls slightly behind the other two models in terms of overall performance.

Based on the results and comparison, the Random Forest model is recommended as the best algorithm for spam SMS classification due to its higher F1-score, precision, and recall and accuracy. An accuracy of 0.98 indicates that the Random Forest Classifier model achieved a high level of accuracy in correctly classifying the SMS messages as either spam or non-spam. In other words, the model accurately predicted the labels of 98% of the test set messages. This was clear, when the model was used to predict the labels for a set of 5 sample messages, all of the predictions were correct. This means that the model accurately identified whether each sample message was spam or not. The combination of a high accuracy on the test set and correct predictions on the sample messages indicates that the Random Forest Classifier model performs well in accurately classifying SMS messages and can be considered reliable for spam detection.

REFERENCES

- UCI Machine Learning Repository: *archive.ics.uci.edu/ml/index.php*
- Carrillo-de-Albornoz, J., Plaza, L., Martínez-Romo, J., and Castro, J.L. (2007). SMS Spam Filtering using Machine Learning Techniques. Proceedings of the 6th Conference on Information Technology and Telecommunications (IT&T 2007), 1-6.
- Cormack, G.V., and Lynam, T.R. (2007). Spam Filtering for Short Messages. Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), 871-872.
- El-Gayar, O.F., and Khedr, A. (2017). SMS Spam Filtering using Machine Learning Techniques: A Review. International Journal of Computer Science and Information Security, 15(5), 18-24.
- Kwok, I.C., Wang, Y., Chen, L., and Chan, P.P. (2012). Word Segmentation and Feature Selection in Chinese SMS Spam Filtering. International Journal of Digital Content Technology and its Applications, 6(20), 227-236.
- Olaleye, S., and Adebiyi, A.A. (2014). A Machine Learning Approach to SMS Spam Filtering using the N-gram Based Feature Selection. In Proceedings of the 5th International Conference on ICT for Africa (ICT4Africa 2014), 1-7.
- Resende, H.P., Valente, T., and Fernandes, E.M. (2013). Feature Extraction for SMS Spam Filtering. Proceedings of the 12th International Conference on Machine Learning and Applications (ICMLA 2013), 228-233.