Research on Naive Bayes Classifier and Random Forest Models' Performance on Spam Message Recognization

Execuative Summary

This report analyzes the effectiveness of the Naive Bayes Classifier and Random Forest Model, in spam message recognition, using precision, recall, and confusion matrix as evaluation metrics on a labeled dataset. This research delves into different feature selection methods and hyperparameters searching methods for both of the dataset and the two classifier models.

Project Objectives

This research evaluates the performance of two machine learning algorithms, the Naive Bayes Classifier and the Random Forest Model, in identifying spam messages from multiple dimensions. The goal is to determine which model offers superior accuracy and efficiency for spam detection tasks.

Data Description

The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged acording being ham (legitimate) or spam. The file contains one message per line. Each line is composed by two columns: v1 contains the label (ham or spam) and v2 contains the raw text. This corpus has been collected from free or free for research sources at the Internet.

Methodology

1. TF-IDF

The text data, consisting of spam and ham messages, is transformed into numerical format using the Term Frequency-Inverse Document Frequency (TF-IDF) method. This technique converts text into a matrix of TF-IDF features, reflecting the importance of words within the dataset relative to their frequency across documents.

- Pros: TF-IDF effectively highlights the most relevant words in each document, reducing the influence of frequently occurring but less informative words across all documents. This helps in improving the signal-to-noise ratio for classification.
- Cons: TF-IDF can lead to high-dimensional feature spaces especially with large vocabularies, potentially increasing computational complexity and memory usage. It also ignores word order and context, which can be crucial for understanding text semantics.

2. Train and test sets Splitting

The train and test sets are splitted as a ratio of 80% to 20%.

3. Feature Selection

In the both case of two feature selection methods, there are 3000 features being chosen.

- Chi-Squared: This method evaluates the independence of each feature with respect to the target variable.
    - Pros: Effective in identifying features with a statistical significance to the target

variable, thus helping to reduce overfitting by eliminating irrelevant features.

- Cons: Relies heavily on the assumption that features are independent; however, in text data, words often have dependencies.

- Mutual Information: Measures the reduction in uncertainty for one variable given a known value of another variable.
    - Pros: Captures any kind of statistical dependency between variables, not just linear relationships like Chi-Squared.
    - Cons: Computationally intensive, especially with large datasets, and can be biased toward features with more categories.

3. Models

- Naive Bayes
    - Pros: Fast and efficient with large datasets. Performs well with an appropriate assumption of feature independence in text classification.
    - Cons: The assumption of feature independence is often violated in real-world data, potentially leading to poor performance.
- Random Forest
    - Pros: Handles non-linear data effectively, is less likely to overfit, and provides feature importance scores, aiding interpretability.
    - Cons: More resource-intensive, requiring more computational power and memory. Slower to train due to constructing multiple trees.

4. 5-fold Cross-Validation

The 5-fold cross-validation is served for searching the best parameters for the models, which is hyperparameters tuning. In this case, smoothing parameter α with a range of {0.01, 0.1, 1, 10, 100} is selected for the cross-validation of Naive Bayes Classifer, and the number of trees in the forest (n_estimators) and the number of features to consider when looking for the best split (max_features) are selected for the cross-validation of Random Forest.

- Pros: Helps in assessing the model's generalizability by using different subsets of the data for training and testing, reducing the risk of model overfitting.
- Cons: Increases computational cost as the model needs to be trained multiple times. It can also lead to variability in model performance depending on how the data is split.

5. Evaluation Methods

- Confusion Matrix: Provides a detailed breakdown of correct and incorrect classifications.
    - Pros: Allows detailed analysis of model performance, showing both errors and correct predictions across classes.
    - Cons: Alone, it does not provide measures of overall accuracy or other summary statistics.
- Precision and Recall Score: Precision measures the accuracy of positive predictions, and recall measures the ability to find all positive instances.
    - Pros: Useful for imbalanced datasets where positive cases are more critical.
    - Cons: Sometimes, a trade-off is required between precision and recall (precision/recall

trade-off). Focusing on one can detrimentally affect the other.

Results

TF-IDF vectorization generates a [5572, 8709] matrix as a result. The evaluation score of Random Forest model is fairly high enough to classify the spam text with 100% precision and 87.5% recall dealing with the chi-squared feature selection data. The best 'max_features' and 'n_estimators' are separatly 'sqrt' and 60. There are 0 case of predicting ham to spam, and only 17 cases of predicting spam to ham. Secondly, it is 100% precision and 86.029% recall dealing with the mutual information feature selection data, also using Random Forest model. The best 'max_features' and 'n_estimators' are 'sqrt' and 40. The only difference of evaluation score between this and chi-squared feature selection is that there are 2 more cases of predicting spam to ham. While the precision scores are both 100% on the test set (the predicted tagged as 'spam' are 100% correct), the recall score has not reached to a A range (90+ score) and the running times of this model for both of the feature selection methods are more than 40 seconds.

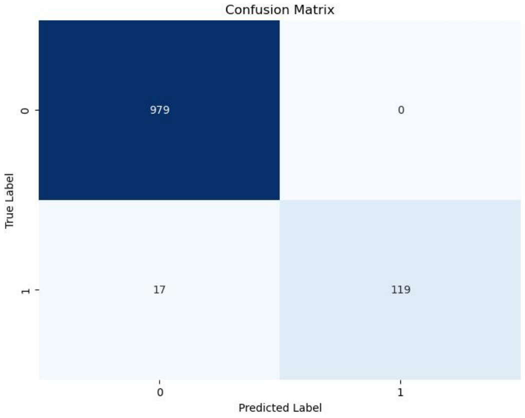

Figure 1. Confusion Matrix of Random Forest Model on Chi-Squared Feature Selection Data

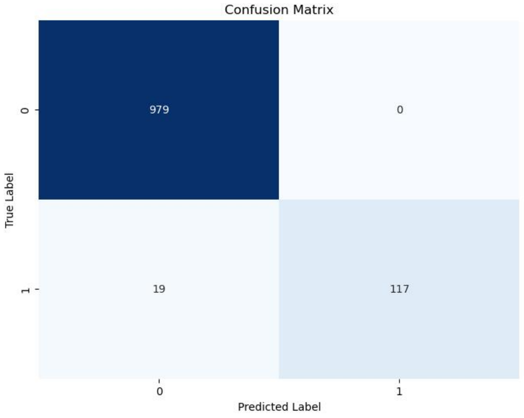

Figure 2. Confusion Matrix of Random Forest Model on Mutual Information Feature Selection Data

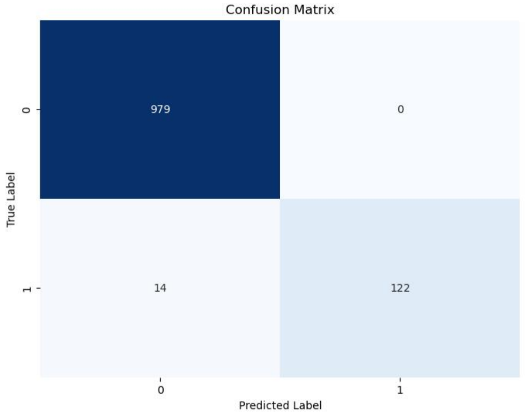

Figure 3. Confusion Matrix of Naive Bayes Model on Chi-Squared Feature Selection Data

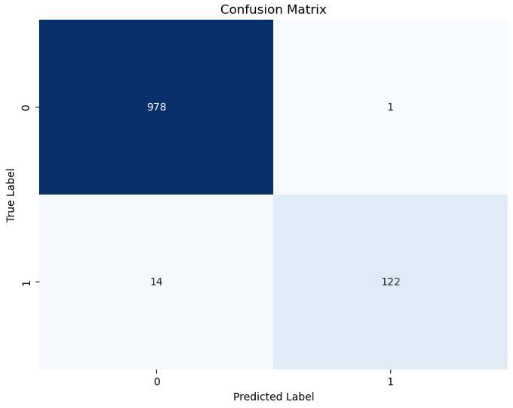

Figure 4. Confusion Matrix of Naive Bayes Model on Mutual Information Feature Selection Data

In regards of the next model, Naive Bayes Classifer has an unexpected prediction performance with 100% precision and 89.705% recall score on the chi-squared feature selected data. The best chosen smooth parameter is 0.1. There are only 14 cases of predicting spam to ham. However, Naive Bayes Classifier was almost the same for the mutual information feature selected data with 99.187% precision and 89.7% recall. The best chosen smooth parameter is also 0.1. Moreover, the average running time of Naive Bayes Classifier during the project is less than 2 seconds.

Conclusion

It is true that Random Forest model is considered to be robuster and less likely to overfit, Naive Bayes Classifier is considered to be the winner in this case of text recognization with the chi-squared feature selection methods not only for the precision and recall score but the much less running time than that of Random Forest. This research of spam message classification including the models and parameters only suit for the this specific data. It is needed to be researched whether the models can be used to recognize spam messages in the real life, and it is desired to look forward to discussing other machine learning models in the future.