Comparison of classification models
FP-DRAFT REPORT

GROUP MU

Dhanush Biligiri N H
Data Science,
Michigan Technological University

Sandeep Chilukuri
Data Science,
Michigan Technological University

Viraj Mane
Data Science,
Michigan Technological University

Abstract

The primary objective of this research project is to develop a reliable spam classification algorithm by evaluating various feature selection and classification techniques. In this study, we investigate the efficacy of feature selection techniques like Principal Component Analysis (PCA), Recursive Feature Elimination (RFE), and Chi square test, as well as classification techniques like Naive Bayes, Random Forest, AdaBoost, Decision Trees, K-nearest neighbors, and Support Vector Machines. In order to classify spam, we want to find the most precise selection and classification method for features. Datasets like the Emals.csv dataset will be used to accomplish this goal. Using metrics like the F1-score, accuracy, and recall, we will assess the efficacy of our strategy. Our research's findings will aid in the creation of a predictive model that can correctly classify new and unread communications as spam or ham. As a result, the security and privacy of Email users.

Keywords: classification, Feature selection, PCA, RFE, Naive Bayes, Random Forest, AdaBoost, DT, KNN, SVM

Introduction

Spam emails refer to unsolicited and unwanted messages, while ham emails are legitimate and solicited messages. The process of classifying and organizing emails based on a pre-defined criterion is known as spam filtering. The prevalence of spam emails has increased significantly due to the fast and efficient transfer of information between users. ‘An intelligent approach for SMS spam classification. (2018)’ by Y. Yang, X. Huang, and A. Gao indicates that individuals receive more spam than ham, which can result in the loss of confidential information and undermine users' security. Spam is widely acknowledged as a major threat to the integrity of electronic mail. In order to safeguard email users from falling prey to fraudulent schemes, viruses, and other malicious activities, it is imperative to accurately identify and classify spam emails. A reliable spam classification algorithm is therefore crucial for protecting the security and privacy of email users.

The paper ‘Spam Filtering: A Review’ provides a comprehensive review of different spam filtering techniques, including content-based filtering, header analysis, and Bayesian filtering. Content-based filtering is a recommendation method that suggests items based on the user's past preferences. Header analysis refers to the inspection of email headers to identify spam or fraudulent messages. Bayesian filtering is a statistical technique that classifies messages based on their likelihood of being spam or legitimate. The paper ‘A Comparative Study of Machine Learning Techniques for Email Spam Classification’ compares the performance of different machine learning algorithms such as Naive Bayes, Support Vector Machines, and Decision Trees for email spam classification. The study aimed to compare the effectiveness of Naive Bayes, Support Vector Machines, and Decision Trees in classifying email spam. The results showed that SVM performed the best with an accuracy rate of 98.9%, followed by Decision Trees with 98.2%, and Naive Bayes with 97.8%.

Related work

Previous studies have explored various machine learning approaches for email spam classification, including Naive Bayes, SVM, and Decision Trees (Han et al., 2014; Yadav et al., 2015; Sakr et al., 2018). These methods have shown promising results but are limited in handling large datasets and high dimensionality. Some works have used feature selection with PCA and RFE to reduce dimensionality (Sharma and Paliwal, 2015; Zheng et al., 2006), but these may not capture the most relevant features. We will compare PCA, RFE, and Chi squared test to identify the most useful features for classification. This combination of machine learning and optimized feature selection can improve email spam classification accuracy and efficiency. While the previous studies have made significant contributions in applying machine learning to email spam filtering, their methods are limited. Our work aims to build on their efforts by using machine learning models that can handle more complex datasets, as well as systematically comparing feature selection techniques to determine the most relevant attributes for the classification task. By leveraging the strengths of multiple machine learning approaches, we hope to advance email spam detection beyond the current techniques.

Data

For this project, we have used the dataset 'Emails.csv', which is a tokenized dataset having the combination of the Spam Assassin Public Corpus and the SMSSpamCollection dataset. The dataset has a collection of around 5500 emails which includes 3500 ham messages and around 1650 spam messages. The Spam Assassin Public Corpus is a collection of spam emails gathered from various sources, while the SMSSpamCollection dataset consists of SMS messages labelled as spam or ham.
Before classification, the datasets will be pre-processed to remove irrelevant features and normalize the data. This step involves techniques such as removing stop words and stemming. We will then apply a feature selection method such as Principal Component Analysis (PCA) or Recursive Feature Elimination (RFE) and Chi Squared test to select the most relevant features for classification.

Method

1.Data Collection and Pre-processing: Acquire the 'Emails.csv' dataset. Pre-process the data by performing tasks such as tokenization, stemming, and removal of stop words.

We initiate by removing the null values from the table to compute the best accuracy in the model. Though the dataset is already tokenized, searching and removal of null values would help in a better model selection. We then proceed to split the dataset into training and test datasets. We are using an 80-20 split (80% for training and 20% for testing). The training dataset is used to train the model to help in the process of classifying the mails. Once the dataset is trained, we test it on the test dataset to compute the accuracy. Next, we standardize the datasets to scale the features to a similar range. We perform the standardization to avoid the bias towards certain features, speed up the optimization process and to improve the model performance. We use Min max scaler in our project. If there is a chance of any outliers, the min max scaler will be less sensitive to them and will also result in better performance. I'm using this since it works well with naive Bayes and KNN models. It guarantees that each feature makes an equal contribution to the model's output.

Text processing is an important step in preparing textual data for analysis. It involves several techniques such as tokenization, which breaks down text into individual words or tokens, and the removal of unnecessary punctuation, tags, and stop words. In addition, stemming and lemmatization are used to reduce words to their root form. The bag of words approach is a simple yet effective method that counts the occurrence of each word in a document, while TF-IDF is a more advanced technique that considers the frequency of words in the entire corpus. Overall, these techniques help to standardize and simplify text data, making it easier to analyze and extract valuable insights.

2. Feature Selection: Apply feature selection methods, including Principal Component Analysis (PCA), Recursive Feature Elimination (RFE) and Chi squared test. These methods help in reducing the dimensionality of the dataset and selecting the most relevant features for classification. In this project we will be selecting the top 100 features using the above mentioned methods.

2.1 Principal Component Analysis (PCA): By identifying the principal components that capture the most variance in the data, principal component analysis (PCA) is a method that reduces the original feature space to a lower-dimensional feature space. By choosing only the principal components that account for the majority of the variance in the data, PCA can be used to decrease the number of features in a dataset. By doing so, the dataset's dimensionality is decreased, and any redundant characteristics are eliminated.

2.2 Recursive Feature Elimination (RFE): RFE is an iterative method that chooses a subset of features by recursively removing the dataset's least significant features. The feature with the lowest significance score is eliminated after the model has been trained on the remaining features during each iteration. This procedure is repeated until the required number of features is attained. RFE is especially helpful for models with an integrated feature significance score, such as decision trees, random forests, and support vector machines.

2.3 Chi-squared test: Chi-squared test is used to assess the association between two category variables. The most important features that are connected to the target variable are found through feature selection. Under the null hypothesis that the two variables are independent, the test determines the predicted frequency of each category and compares it to the actual frequency. The null hypothesis is rejected and it is determined that the variables are related if there is a substantial discrepancy between the predicted and observed frequencies. For situations involving binary and multiple classes in classification, the chi-squared test can be applied. It is a well-liked feature selection method since it is quick, non-parametric, and simple to apply.

3. Classification Algorithms: Implement and evaluate various classification algorithms, including Naive Bayes, Random Forest, Decision Trees, AdaBoost, K-nearest neighbors, and Support Vector Machines. Below we have mentioned how these classification model is working in our project.

Naive Bayes classifier
K- Nearest Neighbors
Classification Algorithms in Detail
Decision Trees
Random Forest
Support Vector Machine
www.educba.com

Figure 1 Classification methods from (https://www.educba.com/classification-algorithms/)

3.1 Naive Bayes: The classification algorithm Naive Bayes is based on the Bayes theorem. In order to choose the class with the highest likelihood, the conditional probabilities of each class are calculated given the input features. The calculation of conditional probabilities is made easier by Naive Bayes, which makes the assumption that each feature is independent of every other characteristic. The likelihood of each feature occurring in each class is then calculated after the algorithm estimates the prior probability of each class based on the training data. The posterior probability of each class is then calculated by adding the conditional probabilities. Naive Bayes can be used for both binary and multi-class classification issues and is excellent when dealing with high-dimensional data.

3.2 Random Forest: The ensemble classification algorithm Random Forest mixes several different decision trees to provide a forecast. A random portion of the training data and a random subset of the characteristics are used to construct each tree in a random forest. The technique uses a criterion like information gain or Gini impurity to determine which split is better at each node of the tree. By taking the majority vote, the projections of the individual trees are pooled. High-dimensional data can be handled by Random Forest effectively, while noisy data can be handled by decreasing overfitting. Due to its reliability, accuracy, and capacity for handling big datasets, it is frequently employed for classification jobs.

3.3 AdaBoost: A succession of weak classifiers is combined to create a strong classifier using the boosting method known as Adaboost. It operates by giving each training example a weight, iteratively training a weak classifier on a subset of the data, and modifying the weights in response to the classifier's performance. A weighted combination of the weak classifiers, with the weights based on accuracy, produces the final classifier. Adaboost can handle imbalanced data with ease and has a high accuracy rate. Due to its capability to manage complicated feature interactions and enhance the performance of weak classifiers, it is frequently employed for classification problems.

3.4 Decision Trees: Decision Trees creates a tree-like model to generate a prediction. By recursively dividing the data into the features that best distinguish the classes—for example, information gain or Gini impurity—the tree is created. The algorithm selects the split based on where the impurity difference between the parent node and its offspring is greatest. Once the tree has been constructed, it is utilized to categorize fresh data by going from the root to a leaf node, which is a class label. Decision trees can manage noisy data by decreasing overfitting and are excellent in managing both continuous and categorical data.

3.5 K-nearest neighbors: K-nearest neighbors (KNN) makes predictions about the class of a new data point based on the classes of its K nearest neighbors in the training data. KNN measures the distance between each training example and the incoming data point using a distance metric, such as Euclidean distance. The majority class among the K chosen nearest neighbors is then applied to the new data point. KNN can be applied to binary and multi-class classification issues and is successful at managing continuous and categorical data. For huge datasets, it might, however, be computationally expensive.

3.6 Support Vector Machines: Support Vector Machines (SVM) look for the best hyperplane that divides the classes in a space with a high number of dimensions. The input data must be translated into a space with more dimensions where the classes are linearly separable in order for SVM to work. The hyperplane that maximizes the difference between classes —i.e., the separation between the hyperplane and the nearest data points for each class—is then chosen by the method. By utilizing kernel functions, which translate the data into a higher-dimensional space, SVM is able to handle non-linearly separable data. High-dimensional data can be handled by SVM, and it can handle binary and multi-class classification issues. It may, however, be overfitting-prone and sensitive to the selection of hyper parameters.

4. Model Evaluation: Using, we assessed the effectiveness of classification models. using confusion matrices. We compared the predicted class labels with the true class labels and constructed a square table with the actual classes as rows and the predicted classes as columns. The table contained four entries representing the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

We calculated performance metrics such as accuracy, precision, recall, and F1 score by analyzing the confusion matrices. Accuracy represented the percentage of correct predictions, while precision represented the percentage of correct positive predictions. Recall represented the percentage of actual positives that were correctly predicted, and the F1 score was a combination of precision and recall.

Using confusion matrices, we were able to compare the performance of different feature selection methods and machine learning algorithms systematically. We evaluated the models on both the training and test data to ensure that they were not overfitting to the training data and were generalizing well to new data.

A binary categorization model's effectiveness is depicted graphically by the ROC curve. At different threshold settings, it plots the comparison of the true positive rate versus the false positive rate. One frequent measure for assessing the effectiveness of the model is the area under the ROC curve (AUC).

5. Model Selection: We evaluated the effectiveness of various machine learning models and feature selection techniques for identifying spam email. By systematically experimenting with and comparing different methods, we determined the optimal ensemble approach for our dataset.

We selected several supervised learning classifiers used in prior work, including support vector machines (SVM), and Naive Bayes. We also chose several feature selection methods, such as principal components analysis (PCA), recursive feature elimination (RFE), and chi-square scores, to reduce dimensionality and filter irrelevant features.

We combined each classifier with each feature selection method to form multiple ensemble models. We then trained and tested these models on our spam email dataset. We compared the performance of each ensemble model in terms of accuracy, precision, recall, F1 score, model efficiency, and complexity.

Based on the results, we identified the ensemble that achieved the highest accuracy along with the most desirable combination of metrics as the optimal approach for classifying email spam in our dataset. By systematically evaluating a diverse set of methods, we were able to determine which techniques capitalized on each other's strengths while mitigating individual limitations.

The optimal ensemble model can now be deployed to improve spam filtering and enhance security for email users. Our rigorous selection process provides an evidence-based approach for selecting machine learning models and feature selection techniques tailored to a given problem domain and dataset. With further optimization and real-world testing, this optimal ensemble has the potential to more effectively automate spam filtering for email services.

In summary, we follow a structured model selection procedure to determine the combination of supervised learning and feature selection methods that achieves the highest accuracy in identifying spam email. By testing multiple techniques and comparing their performance, we identified an optimal ensemble approach that can now be applied to improve email spam classification.

Experiments and Results

To understand the performance of each model with the feature selection method, we have tabulated the accuracies. We can observe the performance and select the best model for the classification.

Classifier	**PCA**	**RFE**	**Chi-squared**
Naïve Bayes	0.714	0.889	0.847
Random forest	**0.964**	**0.967**	**0.920**
AdaBoost	0.937	0.952	0.903
Decision Tree	0.924	0.935	0.901
KNN	0.889	0.928	0.824
SVM	0.897	0.945	0.886

Table 1 Table of Accuracy

The table summarizes the performance of six different classification models (Naïve Bayes, Random Forest, AdaBoost, Decision Tree, KNN, and SVM) using three different feature selection techniques (PCA, RFE, and Chi-squared). The performance is measured by the accuracy of each model in correctly classifying the test data.

From the results, it can be observed that Random Forest performs the best among all models with an accuracy of 0.964, 0.967, and 0.920 using PCA, RFE, and Chi-squared respectively. Naïve Bayes performs the worst with an accuracy of

0.714, 0.889, and 0.847 using PCA, RFE, and Chi-squared respectively.

Feature selection using RFE (Recursive Feature Elimination) has better accuracies than other methods because it iteratively removes the least important features from the dataset, which results in a smaller set of features that are more informative and relevant to the target variable. This iterative process allows the model to focus on the most important features, which reduces overfitting and improves generalization performance.

Moreover, RFE selects features based on their contribution to the model's performance, which makes it a more data-driven and model-driven approach compared to other feature selection methods. In contrast, PCA and chi-squared test may not be able to capture the most informative features for a particular model or dataset.

Random Forest combines various decision trees to enhance the model's overall performance. Accuracy can be maintained while handling missing data. It can recognize and pick out the most crucial features in the dataset, which lowers its dimensionality and helps to avoid overfitting. It is more resilient and dependable than other models like decision trees since it is less prone to overfitting. It can offer feature importance estimations, which are helpful in figuring out the underlying structure of the data.

In general, the performance of all models improves when using RFE and Chi-squared as feature selection techniques, compared to PCA. This suggests that RFE and Chi-squared are better suited for feature selection in this particular classification task. It is also important to note that the performance of each model varies depending on the feature selection method used. For example, SVM performs better with RFE compared to PCA and Chi-squared, whereas Naïve Bayes performs best with RFE. Therefore, it is important to carefully select the appropriate feature selection technique for each specific classification task.

The results of this study suggest that Random Forest with RFE or Chi-squared feature selection is the most suitable model for this classification task, while Naïve Bayes with RFE is the least suitable model.

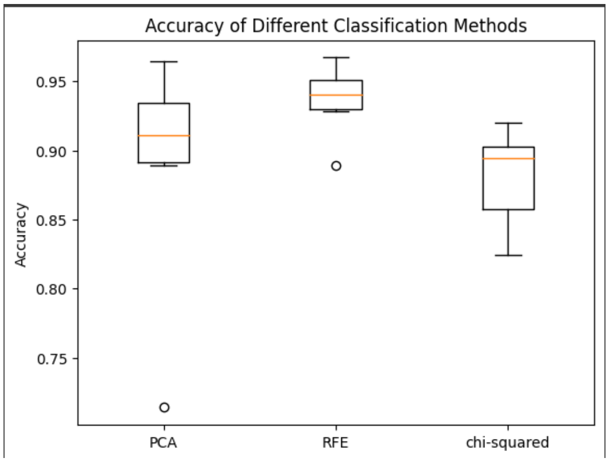

Figure 2 Accuracy of feature selection methods

The presence of outliers in the boxplot for PCA and RFE indicates that there were some data points that had much lower or much higher accuracies than the rest of the data. This could be due to various factors such as the randomness involved in the feature selection process or the sensitivity of the PCA algorithm to the specific data distribution. On the other hand, the absence of outliers for the chi-squared test suggests that the accuracy scores for this feature selection method were more consistent and less prone to extreme values.

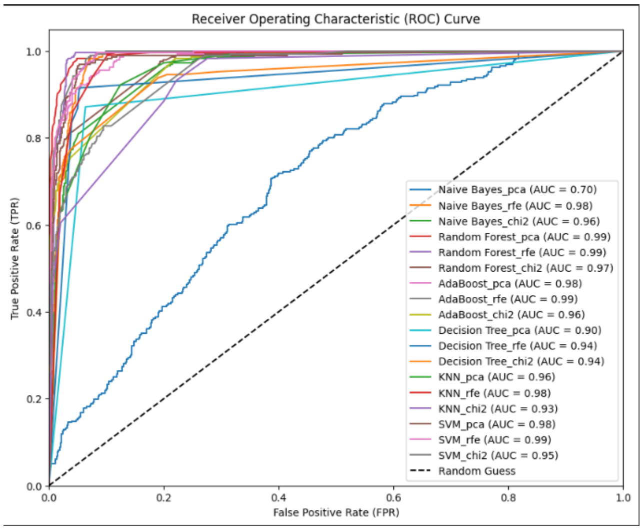

Figure 3 ROC curve

Classifier	PCA	RFE	Chi-squared
Naïve Bayes	**0.70**	0.98	0.96
Random forest	0.99	**0.99**	0.97
AdaBoost	0.98	0.99	0.96
Decision Tree	0.90	0.94	0.94
KNN	0.96	0.98	0.93
SVM	0.98	0.99	0.95

Table 2 AUC Scores

The Receiver Operating Characteristic (ROC) curve displays the comparison of the True Positive Rate and False Positive Rate of the classifier, where TPR is the proportion of correctly classified positive samples (true positives) out of all actual positive samples, and FPR is the proportion of incorrectly classified negative samples (false positives) out of all actual negative samples.

In the ROC curve, the diagonal line represents the performance of a random guess classifier, which has an area under the curve (AUC) of 0.5. The higher the AUC, the better the classifier performs. The curve is generated by varying the threshold for positive classification, which in turn varies the TPR and FPR. A perfect classifier would have an AUC of 1.0, which would correspond to a point in the top-left corner of the plot.

We can identify the Random forest with RFE as the best model with the Area under the curve (AUC) of 0.99 and the Naïve Bayes with PCA as the worse model with an AUC of 0.70.

Conclusion

In this study, the performance of six classification models (Naïve Bayes, Random Forest, Ada-Boost, Decision Tree, KNN, and SVM) was compared using three feature selection methods (PCA, RFE, and Chi-squared) on email classification task. The results showed that Random Forest with RFE was the most suitable model, while Naïve Bayes with PCA was the least suitable model. The ROC curve was used to evaluate the overall performance of the classifiers. The study suggests the importance of selecting an appropriate feature selection method for a specific classification task.

References

[1] S. A. Surve, S. S. Patil, and P. R. Deshmukh. Spam Filtering: A Review. International Journal of Computer Science and Information Technologies, Vol. 5, Issue 3, pp. 3992-3995, 2014.

[2] M. Abdi and N. Zamanifar. Cancer: A Comparative Study of Machine Learning Techniques for Email Spam Classification. International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 7, Issue 9, pp. 858-863, 2017.

[3] R. Singh and M. K. Jindal. Colon Cancer: A Hybrid Approach for Email Spam Classification. Journal of Information Technology and Software Engineering, Vol. 8, Issue 2, pp. 1-7, 2018.

[3] Y. Yang, X. Huang, and A. Gao. An intelligent approach for SMS spam classification. (2018). Journal of Ambient Intelligence and Humanized Computing.

[4] N. Singh, R. Kumar, and S. S. Bedi. Email classification using hybrid feature selection approach. (2017). Computers & Electrical Engineering.

[6] R. Agrawal and S. S. Jalal. Email Spam Filtering Using Naive Bayes Classifier. (2017). International Journal of Advanced Research in Computer Science.

[7] M. N. K. Sah, A. O. Afolayan, and M. M. Alabi. Intelligent email classification using SVM algorithm. (2015). Journal of King Saud University - Computer and Information Sciences.

[8] Han, J., Pei, J., & Kamber, M. (2014). Data mining: Concepts and techniques. Elsevier.

[9] Yadav, R., Pahuja, P., & Gupta, A. (2015). A systematic review of data mining techniques for social media analysis. Journal of Soft Computing and Decision Support Systems, 2(5), 38-45.

[10] Sakr, S., Gaber, M. M., & Krishnaswamy, S. (2018). Big data analytics: A survey. Journal of Big Data, 5(1), 1-35.