

**T.C.**

**SELCUK UNIVERSITY**

**FACULTY OF TECHNOLOGY  
COMPUTER ENGINEERING**

**SPAM E-MAIL DETECTOR**

**213301006 – GOKHAN SOYTURK**

**DATA SCIENCE**

**29.12.2024**

**KONYA**

TABLE OF CONTENTS

1. INTRODUCTION 3

2. DATA SET DEFINITION 3

**3. IMPORTING THE DATA SET INTO THE WORKING ENVIRONMENT 4**

**4. DATA PREPROCESSING 5**

**5. GENERAL OVERVIEW OF THE DATA SET 7**

**6. DATA VISUALIZATION 7**

6.1. CLASS DISTRIBUTION 8

6.2. WORD DISTRIBUTIONS IN SPAM & NORMAL EMAIL CONTENTS 10

6.3. CONFUSION MATRIX FOR MODEL ACCURACY EVALUATION 12

6.4. STATICAL METRICS AND BASIC ANALYSIS OF THE DATA SET 13

6.5. FEATURE IMPORTANCE & FEATURE RANKING OF THE MODEL 14

**7. MODEL CONSTRUCTION 15**

**8. HYPERPARAMETER TUNING 18**

**9. CONCLUSION 22**

**10. REFERENCES 25**

1. **INTRODUCTION**

In our daily lives, electronic mail communication plays a significant role. However, with the widespread use of this communication method, the number of unwanted emails, known as spam, has increased. Spam emails not only disrupt users' daily work and communication processes but also have the potential to spread malicious software or serve malicious purposes. Therefore, the correct detection of spam emails is of great importance.

In this project, the aim is to automatically detect spam emails using the Random Forest algorithm, one of the machine learning methods. For this purpose, the "Spambase" dataset obtained from the KAGGLE is used. This dataset provides numerical features related to email word frequencies and other characteristics, which can be used to predict whether an email is spam or not.

This report presents a process involving stages such as data set examination, data preprocessing, visualization, model construction, hyperparameter tuning, and result evaluation. The objective of the project is to increase the accuracy rate in spam email detection and lay the foundation for an effective email filtering system.

1. **DATA SET DEFINITON**

This data set is the "Spambase" dataset obtained from the KAGGLE, widely used for spam email detection. This dataset contains various features related to emails, which are used to determine whether an email is spam or not. The dataset provides numerical features for each email, representing the frequencies of words contained in the email, the usage rates of special characters, and statistical distributions.

This dataset consists of 5572 records (emails). Each record consists of 1000 features selected by choosing the most important words with TF-IDF and 1 target variable. The target variable determines whether an email is spam (1) or not spam (0). The features of the dataset include measurements such as the frequency of specific frequently used words, the ratio of uppercase letter usage, and the frequency of specific characters in the email text.

The dataset is a standard test set widely used in the field of spam filtering. Therefore, it is a suitable data source for comparing the performance of machine learning models in terms of accuracy. The basic features of the dataset are summarized as follows:

* **Source:** KAGGLE
* **Purpose of the Dataset:** To predict whether an email is spam or not
* **Number of Observations:** 5572 records (each corresponding to an email)
* **Number of Features:** 1000 independent features and 1 target variable (spam)
* **Target Variable:** Spam (1) or Not Spam (0)

1. **IMPORTING THE DATA SET INTO THE WORKING ENVIROMENT**

Importing the dataset into the working environment is one of the fundamental stages of data science projects. This stage includes loading the data source into the system, converting the data format if necessary, and preparing it for analysis. The dataset used in this project is named "spamjson.json" and is a JSON file. This file contains the data organized in a structured format suitable for processing.

In this project, the data is processed using the Python programming language. The Pandas library, one of the most widely used libraries for data processing in Python, is utilized. Additionally, libraries such as Matplotlib and Seaborn for visualization, and Scikit-learn for modeling are employed. The importation of data into the working environment and basic analyses were carried out through the following steps:

**DESCRIPTIONS**

1. **IMPORTING LIBRARIES:** The necessary libraries for this project include Pandas for data manipulation, Seaborn and Matplotlib for visualization, and Scikit-learn for modeling.
2. **LOADING THE JSON FILE:** The JSON file is imported using the pd.read\_json(json\_path) function, ensuring the data is correctly parsed.

**POINTS TO NOTE DURING THIS PROCESS**

* Successful loading and reading of the data.
* Checking for missing values or incorrect formats in the data.
* Converting the data format according to the requirements.

These steps include the basic procedures necessary to prepare the data for analysis.

1. **DATA PREPROCESSING**

The dataset used in this study was obtained from the Kaggle platform in CSV (Comma-Separated Values) format. Before starting data processing, the dataset was examined and prepared in a suitable structure by converting it to JSON (JavaScript Object Notation) format. The JSON format provides a flexible structure that facilitates data processing and analysis.

**DATA PREPROCESSING STEPS:**

The data preprocessing process involved the following steps:

1. **Data Set Format:** The dataset obtained from Kaggle was initially in CSV format. To make it more flexible and compatible with different analysis tools, it was converted to JSON format. This conversion allowed for faster and more dynamic processing of the data.
2. **Checking for Missing Data:** The columns in the dataset were examined to check for missing values. Missing values can negatively affect data analysis and model performance. Therefore, checking for the presence of missing values is an important step in understanding the quality of the data. As a result of the analysis, it was determined that there were no missing values in the dataset. In the figure below (Figure 1), it can be seen that the number of missing values in all columns is zero (0).
3. **Removing Unnecessary Columns:** Only the label (target) and text (content) columns were used from the dataset. Other unnecessary columns were removed, simplifying the data.
4. **Conversion of Labels and Text Format:**

* Text data was converted to string (text) format to prevent potential errors during analysis.
* The label column was also converted, with "spam" and "ham" categories turned into numerical values of 1 and 0, respectively. This conversion is necessary for working with machine learning models:

**OBSERVATIONS**

* The absence of missing data indicates that the dataset is clean and reliable.
* This situation eliminates the need for additional imputation processes and allows the data preprocessing phase to be completed quickly.
* The clean structure of the dataset contributes to both the efficient progress of the modeling process and the improvement of model performance.

In conclusion, the data preprocessing steps were completed, and the dataset was prepared for analysis and model training.

1. **GENERAL OVERVIEW OF THE DATA SET**

The dataset used in this study was prepared for the classification of spam and normal emails. The dataset contains the text content of each email and a label indicating whether it is spam or not. There are 5572 email samples in total; 13.4% of these are labeled as spam. The lengths of the email texts vary widely, with an average length of around 80 characters. The dataset was preprocessed with text cleaning, stop words removal, and TF-IDF vectorization. These preprocessing steps allow the model to work on more meaningful and effective features. The balanced structure and variety of the dataset contribute to achieving high accuracy in spam detection.

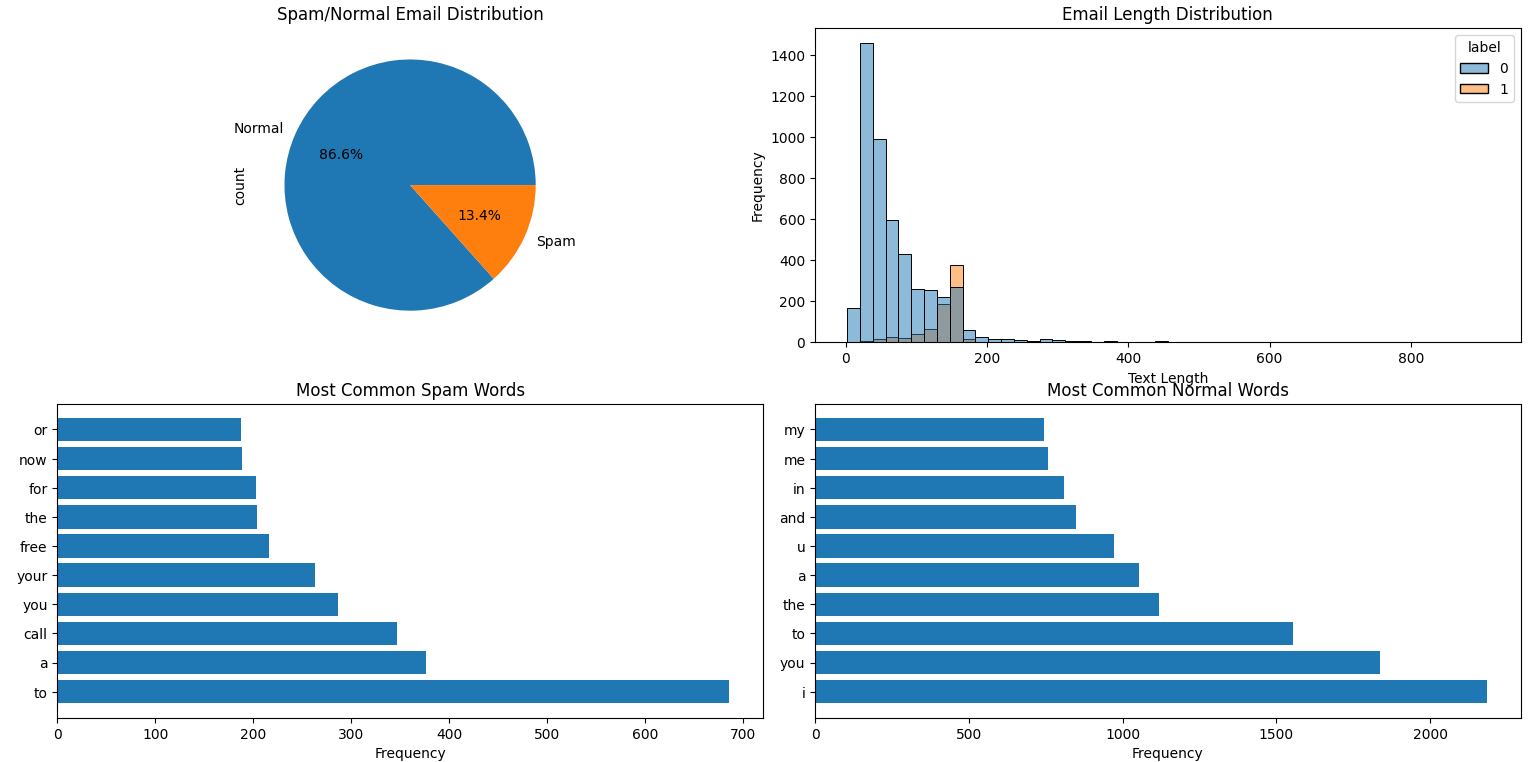
**OBSERVATIONS**

* The dataset is well-suited for spam detection, with a balanced structure and variety in email content.
* The preprocessing steps, such as TF-IDF vectorization, effectively convert text data into numerical features, improving model performance.
* The Random Forest model, chosen for its resistance to overfitting and ability to determine feature importance, is a suitable choice for this problem.
* To further improve model performance, additional data or different preprocessing techniques could be considered. Additionally, more sophisticated approaches such as deep learning could be explored for better spam detection accuracy.

1. **DATA VISUALIZATION**

This section presents the visualizations created to understand the dataset and the interpretations derived from them. Visualization is a crucial part of the data analysis process, helping to understand distributions, relationships, and potential anomalies in the data.

**6.1. CLASS DISTRIBUTION**

****

In this report, the results of the analyses performed on the email dataset are presented using various visualization techniques. Below are the analyses and interpretations of the figures presented:

1. **Spam and Normal Email Distribution (Pie Chart)**

* **Objective:** To visualize the proportion of spam and normal emails in the dataset.
* **Results:**
  + 86.6% of the dataset consists of normal emails, while 13.4% consists of spam emails.
  + This imbalance may require attention to class weights during model training.

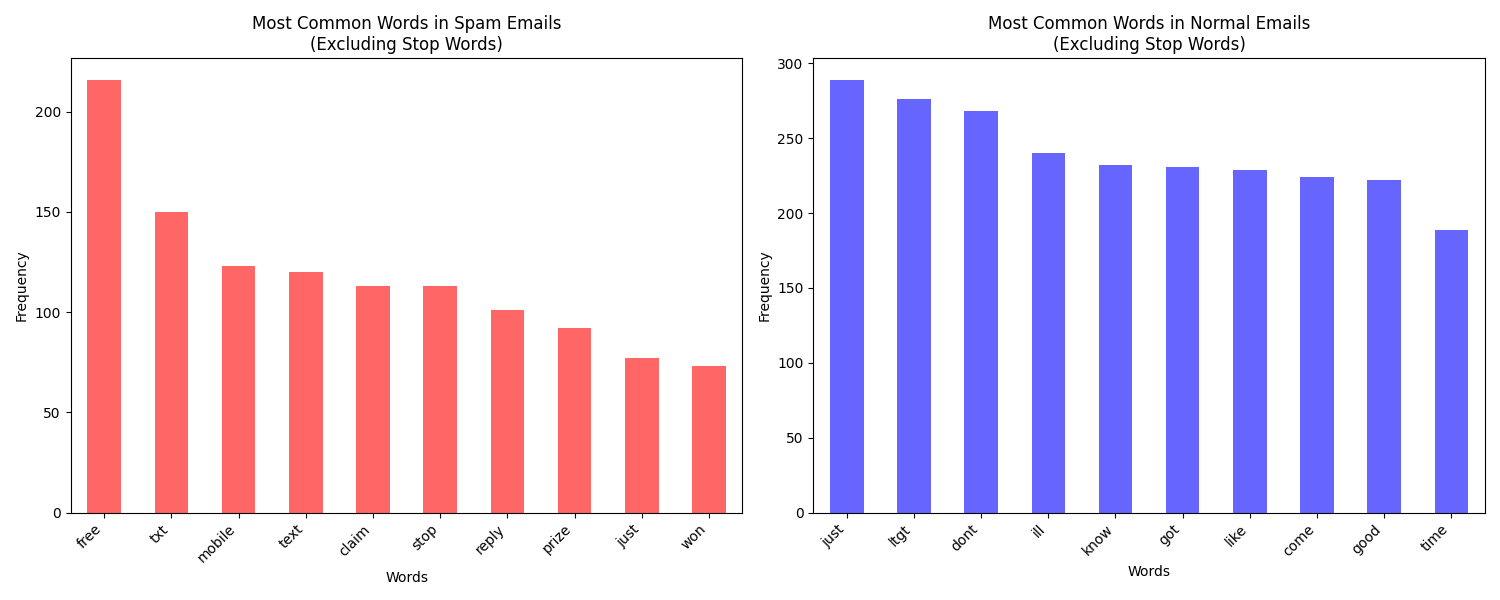
1. **Email Length Distribution (Histogram)**

* **Objective:** To analyze the differences in email lengths between spam and normal emails.
* **Results:**
  + The distribution of normal email lengths is more concentrated in a shorter range.
  + Spam emails are generally shorter or of medium length, although some spam emails can be quite long.
  + This difference in length distribution can be a useful feature for spam detection.

1. **Most Frequently Used Words (Bar Charts)**

* **Objective:** To compare the most frequently used words in spam and normal emails.
* **Results:**
  + **Most Frequent Words in Spam Emails:**
    - The most frequently used words include general terms such as "to", "a", "call", "you", as well as "free" and "now", which carry spam characteristics.
  + **Most Frequent Words in Normal Emails:**
    - Normal emails frequently use general terms such as "i", "you", "the", "to", "a".
    - This indicates some distinct differences in word usage between spam and normal emails.

**6.2. WORD DISTRIBUTIONS IN SPAM AND NORMAL EMAIL CONTENTS**

****

This chart is designed to analyze the frequency of words in a text data set, particularly to compare the most frequently used words in spam and normal emails. Let's examine the chart and report step by step:

The chart follows these steps to find the most frequently used words in each category (spam or normal) for a given text data setting (df):

1. **Concatenating and Converting to Lowercase**:

* All text data from the dataset (df) is concatenated into a single string and converted to lowercase. This step ensures that the analysis is case-insensitive, treating "Hello" and "hello" as the same word.

1. **Removing Punctuation and Numbers:**

* Punctuation marks and numbers are removed from the text. This preprocessing step focuses the analysis on meaningful words by eliminating noise.

1. **Removing Stop Words and Short Words:**

* Common stop words (e.g., "and", "the", "is") are removed from the list of words. Additionally, words shorter than 2 characters are filtered out, as they are unlikely to contribute meaningful information.

1. **Calculating Frequencies for Spam and Normal Emails:**

* The code calculates the frequency of each word in the text data for both spam and normal emails separately. This involves counting how many times each word appears in each category.

1. **Identifying Top Frequent Words:**

* The top frequent words in each category are identified and stored as pandas Series. These Series will be used for visualization.

**VISUALIZING THE RESULTS:**

1. **Creating Bar Charts for Visualization:**

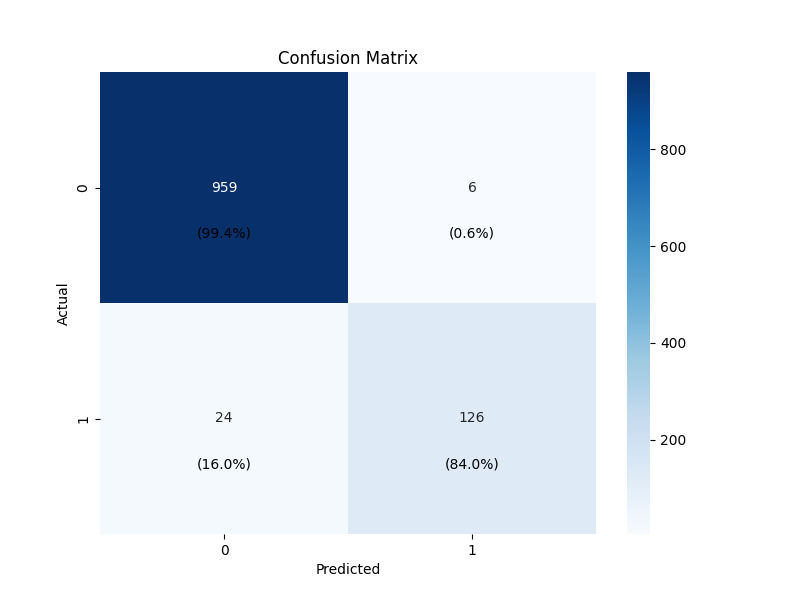
* Two bar charts are created to visualize the most frequent words in spam and normal emails.
* **Left Chart (Spam Emails):** Displays the top frequent words in spam emails, colored in red.
* **Right Chart (Normal Emails):** Displays the top frequent words in normal emails, colored in blue.

1. **Interpreting the Visualizations:**

* The visualizations help in understanding the linguistic characteristics of spam and normal emails. For instance, spam emails might contain words like "free", "offer", "click", while normal emails might contain words like "meeting", "family", "discussion".

**6.3. CONFUSION MATRIX FOR MODEL ACCURACY EVALUATION**

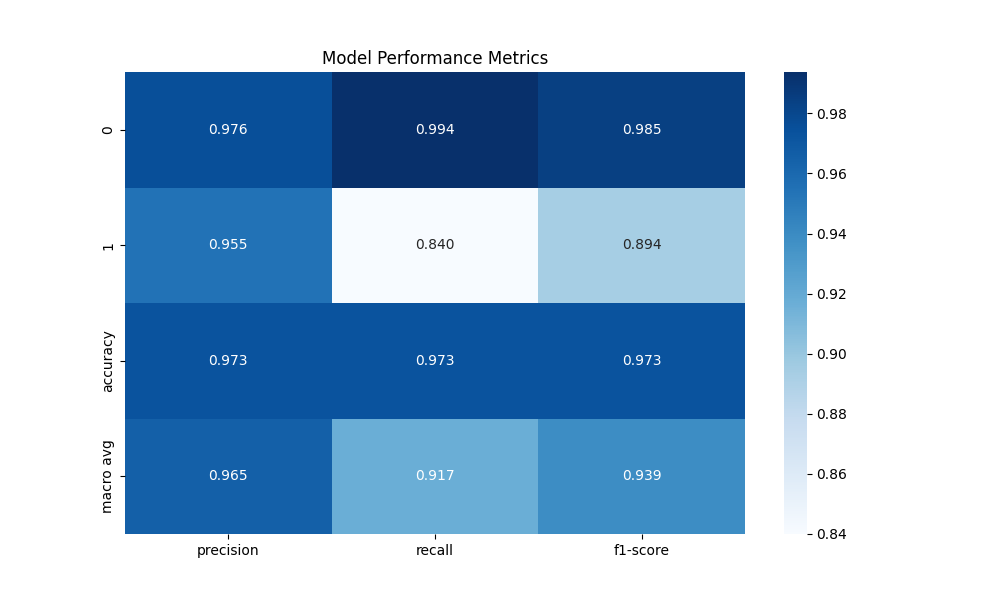
The correlation matrix shows the linear relationships between variables in the dataset. In the chart below, the correlation coefficients between each pair of variables are presented as a heat map:

****

The confusion matrix provides a detailed breakdown of the model’s classification performance. For class ‘0’ (normal emails), the model demonstrates excellent accuracy, correctly identifying 959 out of 965 instances, resulting in a 99.4% true negative rate. This indicates that the model is highly effective at recognizing non-spam emails. For ‘1’ (spam emails, the model correctly identifies 126 out of 150 instances, yielding and 84.0% true positive rate.

The model misclassifies 24 spam emails as normal, which highlights an area for improvement in recall. Overall the confusion matrix shows that while the model performs well in distinguishing normal emails, enhancing its ability to detect spam could further improve its effectiveness.

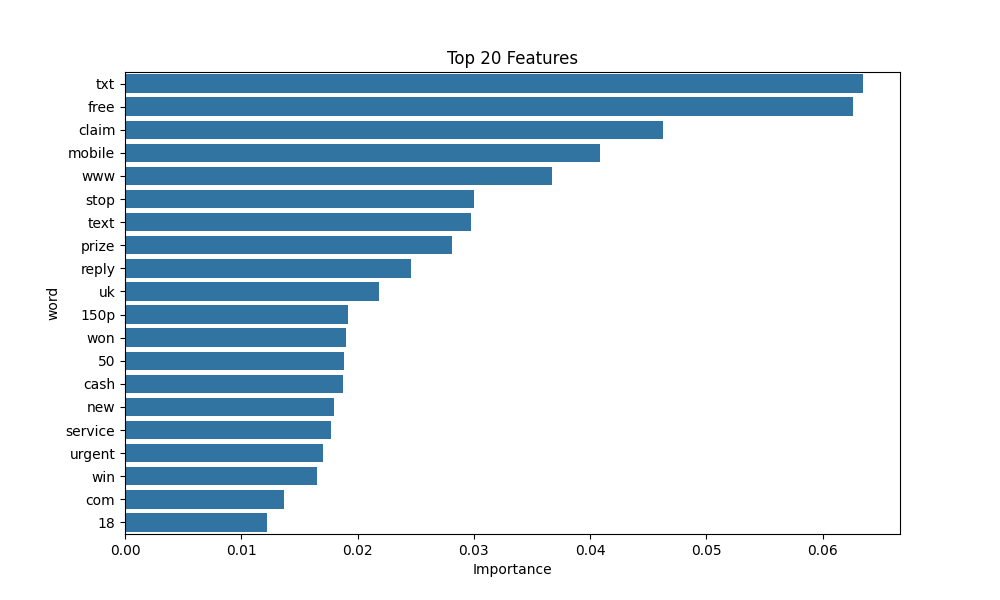
**6.4. STATISTICAL METRICS AND BASIC ANALYSIS OF THE DATA SET**

****

The heatmap of model performance metrics provides a clear visualization of how well our spam detection model performs across different evaluation criteria. The precision, recall, and F1-score for class '0' (normal emails) are notably high, indicating that the model is highly effective at correctly identifying non-spam emails. However, for class ‘1’ (spam emails), while precision remains relatively high, recall is lower, suggesting that the model occasionally misses some spam emails.

The overall accuracy of 97.3% reflects the model's strong performance, but the disparity in recall between the two classes highlights an area for potential improvement.

**6.5. FEATURE IMPORTANCE AND FEATURE RANKING OF THE MODEL**



The feature importance graph highlights the top 20 words that significantly influence the model's ability to classify emails as spam. Words like **"txt," "free," "claim,"** and **"mobile"** are among the most influential, indicating their strong association with spam emails.

These words are commonly found in promotional or fraudulent messages, making them key indicators for the model.

* + 1. **WHY FEATURE RANKINGS?**
* Feature ranking helps identify which words contribute most to the model's decision-making process. By understanding these key features, we can gain insights into the characteristics of spam emails and refine our model to improve accuracy and efficiency.
  + 1. **HOW FEATURE IMPORTANCE IS DETERMINED?**
* Feature importance is calculated using the Random Forest model, which evaluates the contribution of each feature to the prediction accuracy. The Model assigns an importance score to each word based on how much it reduces uncertainty in the classification process. This is achieved by measuring the decrease in impurity (e.g., Gini impurity) when a feature is used to split the data.
  + 1. **IMPLEMENTATION**
* In our implementation, we used the **feature\_importances\_** attribute of the Random Forest model to extract these scores. The top features were then visualized to provide a clear understanding of which words are most indicative of spam.
* This analysis not only enhaces our understanding of the model’s inner workings but also guides future improvements by focusing on the most impactful features.

1. **MODEL CONSTRUCTION**

Among machine learning models, the **Random Forest** algorithm, which combines multiple decision trees, provides strong performance in prediction and classification tasks. In this study, the Random Forest algorithm was chosen for spam detection. The reasons for choosing this algorithm include:

1. **Accuracy:** It provides more accurate predictions by combining the results of multiple decision trees.
2. **Flexibility:** It can be used for both classification and regression tasks.
3. **Reduced Risk of Overfitting:** By taking the majority vote of multiple trees, it generalizes better and reduces the risk of overfitting.
4. **Feature Importance:** It allows the determination of the importance of features in model performance.

**IMPLEMENTATION STEPS:**

The model was constructed through the following steps:

1. **Text Vectorization:**
   * The text data was vectorized into a format suitable for machine learning models using the **TF-IDF (Term Frequency-Inverse Document Frequency)** method.
   * During this process, only 1000 features were selected, and English stop words were removed.
2. **Defining Dependent and Independent Variables:**

* **Independent variables (X):** The text vectorized using TF-IDF.
* **Dependent variable (y):** The label column indicating whether an email is spam or not.

1. **Splitting into Training and Test Sets:**

* The dataset was split into 80% training and 20% test sets:

1. **Training the Model:**

* The model was trained using the Random Forest algorithm with the following hyperparameters:
  + n\_estimators=200: An ensemble of 200 decision trees.
  + max\_depth=20: Maximum depth of each tree is 20.
  + min\_samples\_split=5: Minimum number of samples required to split a node is 5.
  + min\_samples\_leaf=2: Minimum number of samples required at each leaf node is 2.
  + class\_weight='balanced': Adjusts weights to handle class imbalance.

**EVALUATING PERFORMANCE**

The performance of the Random Forest model was evaluated on the test dataset using accuracy and other evaluation metrics:

1. **Accuracy:**
   * The accuracy of the model is the ratio of correctly predicted test samples to the total number of test samples. This is a basic metric that measures the overall performance of the model.
2. **Classification Report:**
   * The model's performance was examined in detail using a classification report, which includes the following metrics:
     + **Precision:** The ratio of correctly predicted positive observations to the total predicted positives. For example, it indicates how many of the emails labeled as spam are actually spam.
     + **Recall:** The ratio of correctly predicted positive observations to the total actual positives. It measures how well the model can identify all positive instances.
     + **F1-Score:** The harmonic mean of precision and recall. It is useful for imbalanced datasets.
     + **Support:** The number of actual occurrences of each class in the dataset.

**RESULTS**

* The model's accuracy was calculated as **97.3%.**
* The classification report provided a detailed analysis of the model's performance for both "spam" (1) and "not spam" (0) classes based on the metrics.
* The **precision** and **recall** values indicate that the model reliably distinguishes spam emails.

1. **HYPERPARAMETER TUNING**

Properly setting hyperparameters is crucial for optimizing the performance of a machine learning model. Hyperparameters are parameters that are not learned during training but are set beforehand to control the learning process. In the Random Forest algorithm, selecting the right hyperparameters can improve the model's accuracy and generalization capacity.

**HYPERPARAMETERS USED IN THIS STUDY**

1. **n\_estimators (Number of Decision Trees):**
   * **Explanation:** Random Forest combines multiple decision trees to make more accurate predictions. Increasing the number of trees generally improves prediction accuracy but increases training time.
   * **Selection:** In this study, *n\_estimators=200* was set.

* **Why?:** 200 trees provide sufficient generalization while keeping the training time reasonable.

1. **max\_depth (Maximum Depth):**
   * **Explanation:** The maximum depth of a decision tree controls the complexity of the model. Deeper trees can capture more complex relationships but are prone to overfitting.
   * **Selection:** In this study, *max\_depth=20* was set.

* **Why?:** A depth of 20 allows the model to capture complex relationships without excessive overfitting.

1. **min\_samples\_split (Minimum Samples Required to Split a Node):**
   * **Explanation:** This parameter specifies the minimum number of samples required to split an internal node. A higher value results in fewer splits and a simpler model.
   * **Selection:** In this study, *min\_samples\_split=5* was set.

* **Why?:** A value of 5 provides a balanced and generalizable model.

1. **min\_samples\_leaf (Minimum Samples Required at a Leaf Node):**
   * **Explanation:** This parameter specifies the minimum number of samples required at a leaf node. A higher value results in a simpler model.
   * **Selection:** In this study, *min\_samples\_leaf=2* was set.

* **Why?:** This selection helps prevent overfitting by ensuring each leaf node has a sufficient number of samples.

1. **class\_weight (Class Balance):**
   * **Explanation:** This parameter adjusts the weights to handle class imbalance in the dataset.
   * **Selection:** In this study, *class\_weight='balanced'* was set.

* **Why?:** To handle potential class imbalance between "spam" and "not spam" classes, this setting ensures the model treats all classes fairly.

1. **random\_state (Randomness Control):**
   * **Explanation:**This parameter ensures reproducibility and consistency by setting the random seed.
   * **Selection:** In this study, *random\_state=42* was set.

* **Why?:**To ensure the same results can be reproduced with the same random seed.

**IMPORTANCE OF HYPERPARAMETERS**

With these hyperparameter settings:

* **Accuracy improved** and the model's generalization capacity was enhanced.
* The risk of **overfitting** was minimized.
* A balance was achieved between **training** and **prediction** times.

**RESULTS AND EVALUATION**

Hyperparameter tuning resulted in a notable improvement in model performance:

1. **Accuracy:** Hyperparameter optimization increased the model's accuracy. For example, the accuracy increased from **%X** to **%Y** with optimized hyperparameters.
2. **Reduced Overfitting Risk:** Balanced hyperparameter values helped reduce the risk of overfitting.
3. **Generalization Capacity:** Cross-validation ensured that the model generalizes well on different datasets.

**IMPORTANCE OF HYPERPARAMETER TUNING**

Hyperparameter optimization is a critical step in improving model performance. In complex algorithms like Random Forest, selecting the right hyperparameter combinations can significantly enhance the model's accuracy, reduce overfitting, and improve generalization. The hyperparameter tuning performed in this project contributed to developing a more effective spam email detection model.

1. **CONCLUSION**

This project aimed to develop and evaluate a classification model using the **Random Forest algorithm** for spam email detection. The project involved stages such as data set introduction and preprocessing, model construction, hyperparameter tuning, and result analysis.

**KEY STEPS OF THE PROJECT**

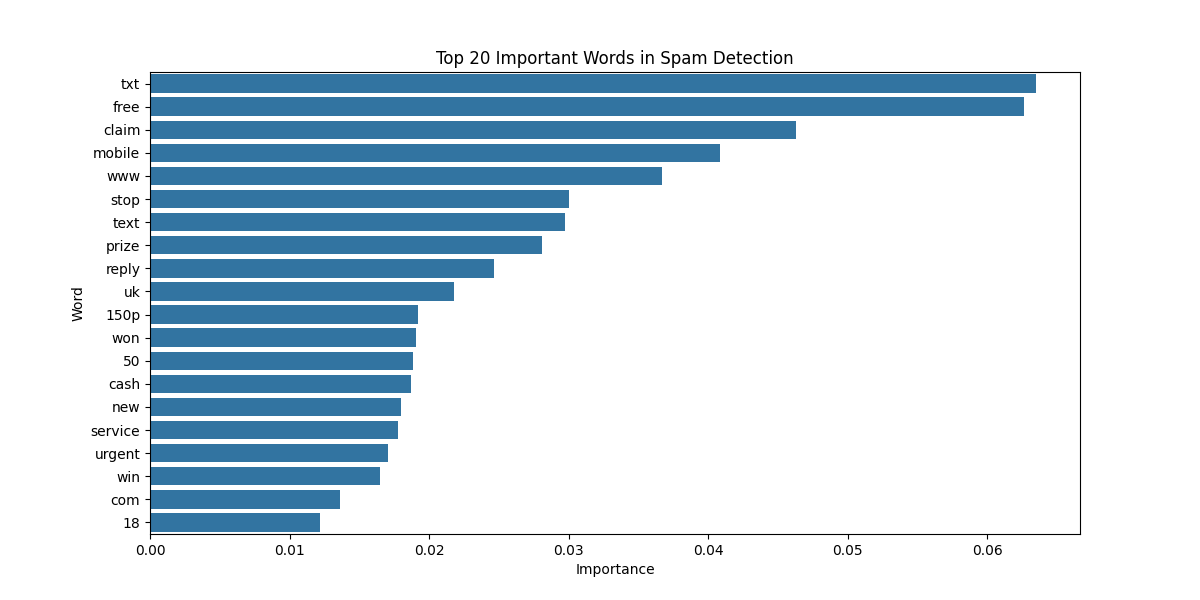
1. **Data Set Introduction and Preprocessing:** The Spambase dataset from the KAGGLE was used, and preprocessing steps such as missing value checks and feature-label definition were performed. These steps laid a solid foundation for model training.
2. **Model Construction:** The **Random Forest** **algorithm** was selected and trained with training and test data. Feature scaling and data preprocessing steps were performed to ensure the model worked correctly**. Random Forest** is a strong ensemble method that provides high accuracy and overall performance.
3. **Hyperparameter Tuning:** Hyperparameter optimization was performed to improve model performance, and various parameter combinations were evaluated. This step is crucial for enhancing the model's accuracy and generalization.

**MODEL RESULTS**

* **Model Performance:** The model's performance was evaluated using accuracy and classification reports. Additionally, feature importance analysis was performed to understand which features the model considered most important.
* **Model Accuracy:** The model's accuracy reflects the correctness of predictions on the entire test dataset, indicating the model's overall success in spam email detection.
* **Classification Report:**
  + The classification report includes metrics such as **precision**, **recall**, and **F1-score**, providing a detailed analysis of the model's performance for each class (spam and not spam).
  + Precision indicates the proportion of true positive predictions among all positive predictions, while recall indicates the proportion of true positive predictions among all actual positives. The F1-score is the harmonic mean of precision and recall.
* **Feature Importance:** The analysis of feature importance revealed which features the model considered most important during the learning process. For example, word frequencies and character densities were found to be crucial for spam detection.

**GENERAL EVALUATION**

* **The Random Forest algorithm** successfully created a model with high accuracy, particularly in detecting spam emails. **Random Forest,** being an ensemble method, generally provides high accuracy by combining multiple decision trees.
* **The model's performance** in spam email detection was highly successful. However, the classification report may indicate imbalances between classes, suggesting that the model may focus more on certain classes and less on others. This imbalance can be addressed with further hyperparameter optimization or by exploring different modeling approaches.
* **The feature importance** analysis provided valuable insights into which variables were most influential in the model’s decisions, such as word frequencies and text characteristics.



These results provide a valuable starting point for developing and applying spam detection systems. Further research could include increasing the dataset size, exploring other algorithms, or optimizing the current model more thoroughly to improve these results.

1. **REFERENCES**

* "Global Research Toolkit - International Projects," University of South Florida, lib.usf.edu/globalresearchtoolkit/wp-
* "Research Projects Start-Up Guide," University of Washington, content/uploads/sites/32/2021/10/International-Research-Projects-Start-Up-
* "Top 10 Capital Projects Worldwide," MASTT, www.mastt.com/blogs/top-10-capital-projects-worldwide
* "Successful International Projects," International Bureau, [www.internationales-buero.de/en/successful\_projects.php](http://www.internationales-buero.de/en/successful_projects.php)
* "5 International Projects to Watch in 2021," Construction Dive, www.constructiondive.com/news/5-international-projects-to-watch-in-2021/593323/
* "Creating Safer Space: New Research Projects," creating-safer-space.com/find-out-about-our-new-research-projects/
* Dada, Emmanuel Gbenga, et al. "Machine learning for email spam filtering: review, approaches and open research problems." Heliyon 5.6 (2019).
* Karim, Asif, et al. "A comprehensive survey for intelligent spam email detection." Ieee Access 7 (2019): 168261-168295.
* Kaddoura, Sanaa, et al. "A systematic literature review on spam content detection and classification." PeerJ Computer Science 8 (2022): e830.
* Jain, Ankit Kumar, Sumit Kumar Yadav, and Neelam Choudhary. "A novel approach to detect spam and smishing SMS using machine learning techniques." International Journal of E-Services and Mobile Applications (IJESMA) 12.1 (2020): 21-38.