**CSCE 5215 – MACHINE LEARNING**

**Project Title – Spam Message Classification using Machine Learning**

**GROUP-25**

**ABDUL AHAD MOHAMMED | ANIL KUMAR BHOGADI | PRANAY KUMAR REDDY MALREDDY**

**GOALS & OBJECTIVES:**

**MOTIVATION:**

The motivation for a project on spam message classification using machine learning typically stems from the need to address the pervasive issue of unwanted or unsolicited messages, commonly known as spam. Here are some key motivations for undertaking such a project:

* Security Concerns
* Resource Efficiency
* User Experience Improvement

**SIGNIFICANCE:**

The project "Spam Message Classification using Machine Learning" holds significant importance in the field of natural language processing (NLP) and cybersecurity. Here are some key aspects of its significance:

* Email filtering
* Spam Detection
* Text Classification

**OBJECTIVES:**

Spam messages are becoming highly common in different means of communication, including email, text messaging, and social media. The objective of this project is to reduce and filter undesired communication by using machine learning techniques to accurately classify messages as spam or non-spam. We can improve user experiences and safeguard individuals and businesses from spam-related hazards by building an effective spam message classifier.

**FEATURES:**

A project on spam message classification using machine learning typically involves the development of a model that can automatically identify whether a given message is spam or not.

Here are some common features and steps involved:

* Data collection
* Preprocessing data
* EDA
* Model training **INCREMENT – 2:**

**Related Work (Background)**

The detection of SMS spam has been the subject of much study; machine learning classifiers have been employed in several studies to tackle the issue of distinguishing spam from authentic (ham) SMS communications. Regarding this, Gupta et al. (2018) assessed the effectiveness of several machine learning models in recognizing spam emails and conducted research on spam SMS identification using machine learning classifiers that were highlighted at the 2018 International Conference on Contemporary Computing. The study offers useful information on classifier performance and serves as a benchmark for comparing the efficacy of related algorithms.

In addition, Shafi'I et al2017 .'s analysis of mobile SMS spam filtering techniques, which was published in IEEE Access, emphasized the variety of strategies used by writers to stop SMS spam in the literature. It offers a comprehensive analysis of the advantages and disadvantages of several approaches, including those based on machine learning. This evaluation aids in elucidating the variety of SMS spam filtering techniques and guides the choice of useful tactics.

Machine learning approaches were employed by Saha et al. (2021) to address the problem of SMS spam detection. As they attempted to apply machine learning approaches to increase the accuracy of spam identification, this effort adds to the body of knowledge and provides insightful information to improve the functionality of spam detection systems while offering a thoughtful perspective on the most recent developments in SMS filtering.

**Dataset**

With the variety of messages provided by this collection, it is possible to swiftly train and assess the machine learning model utilizing data from several websites. Kaggle donated the SMS Spam Collection for this inquiry, which contains over 5,500 English-language SMS messages classed as "spam" or "ham" (authentic). This dataset is used to train and test a machine learning model that detects spam messages. During the data collection and preparation phases, the dataset is cleaned, tokenized, and appraised, which increases the model's accuracy and longevity.

**Detail Design of Methods**

The approach to analyzing the SMS dataset involves several steps, starting from data preprocessing and exploration, followed by model construction, training, evaluation, and deployment for predictions.

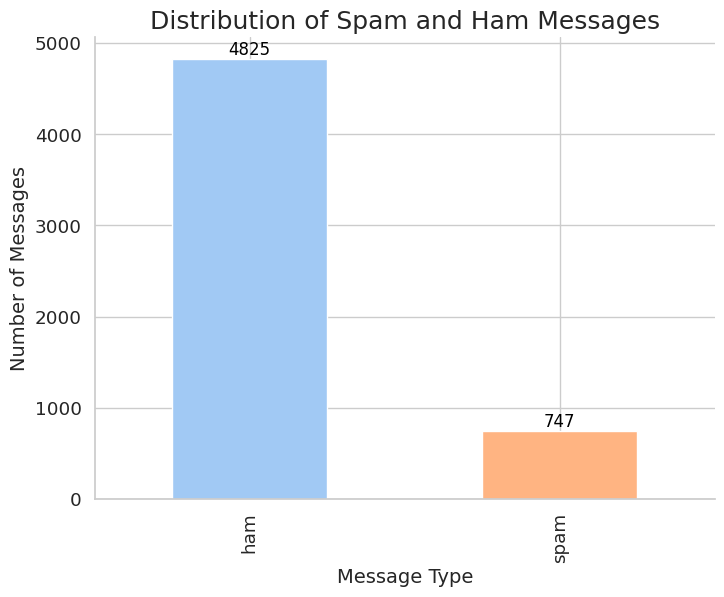
1. **Data Preprocessing and Exploration:**
   * **Data Collection:** The dataset is acquired via the Kaggle API, utilizing a custom function that interfaces with Kaggle's API to download the required dataset.
   * **Preprocessing:** Initial preprocessing involves reading the dataset using Pandas, renaming columns, and performing text preprocessing tasks such as tokenization, lowercasing, punctuation removal, stopword elimination, and stemming using NLTK.
   * **Exploratory Data Analysis (EDA):** Utilizing Seaborn and Matplotlib, the EDA phase encompasses statistical analysis, including message length distribution visualization using histograms, pie charts, and bar plots to identify common words in spam and non-spam messages.
2. **Feature Engineering and Model Construction:**
   * **Feature Extraction:** The CountVectorizer from Scikit-learn transforms text data into numerical features for model training.
   * **Model Architecture:** Logistic Regression is employed as the classification algorithm, with hyperparameter tuning through GridSearchCV to optimize the model's performance.
   * **Model Evaluation:** To measure the model's performance on the test set, evaluation metrics like as accuracy and recall scores are produced. A confusion matrix depicts the predicted accuracy of the model.
3. **Model Saving and Usage:**
   * **Model Serialization:** To enable future use without retraining, the learned model is serialized using joblib.
   * **Prediction on Sample Texts:** A set of example texts is used to show the model's capacity to classify messages as spam or non-spam, offering insights into the model's accuracy and potential for real-world application.
   * **Real-time Predictions:** By accepting user input, the model may be used to make real-time predictions, allowing for realistic implementation of spam detection across numerous communication channels.

**Analysis**

The analysis step includes a thorough assessment of the dataset, model performance, and significant insights generated from the machine learning approach utilized for spam message categorization.

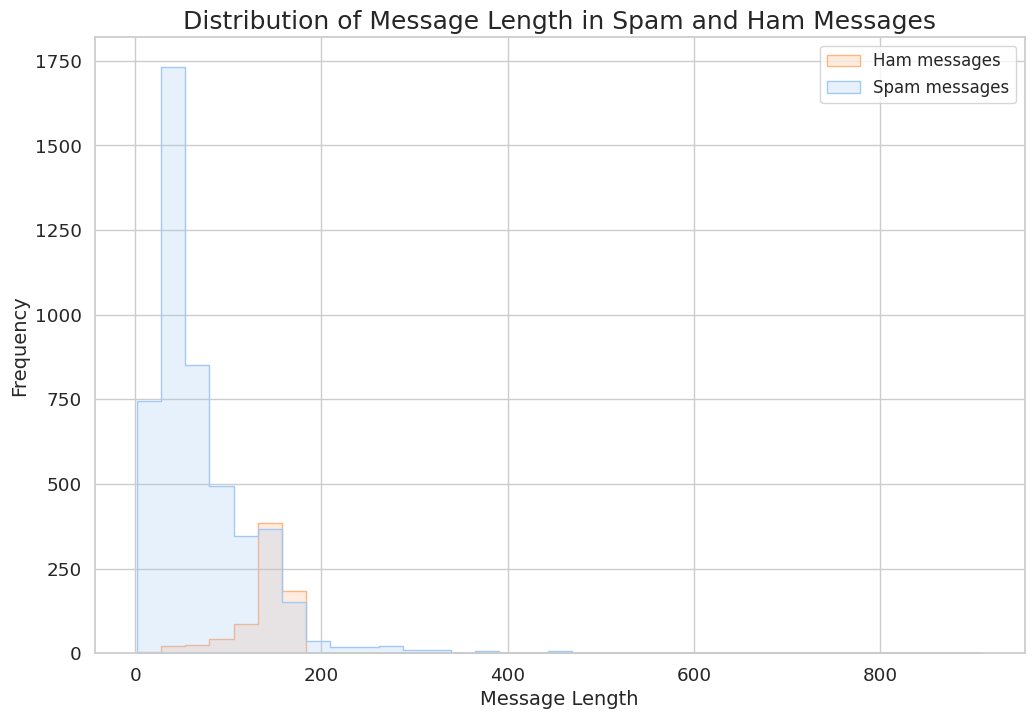
**1. Label Distribution**

The initial analysis focuses on how labels (spam and ham) are distributed across the SMS collection. Using pie charts and bar charts, this section clearly illustrates the dataset's composition.



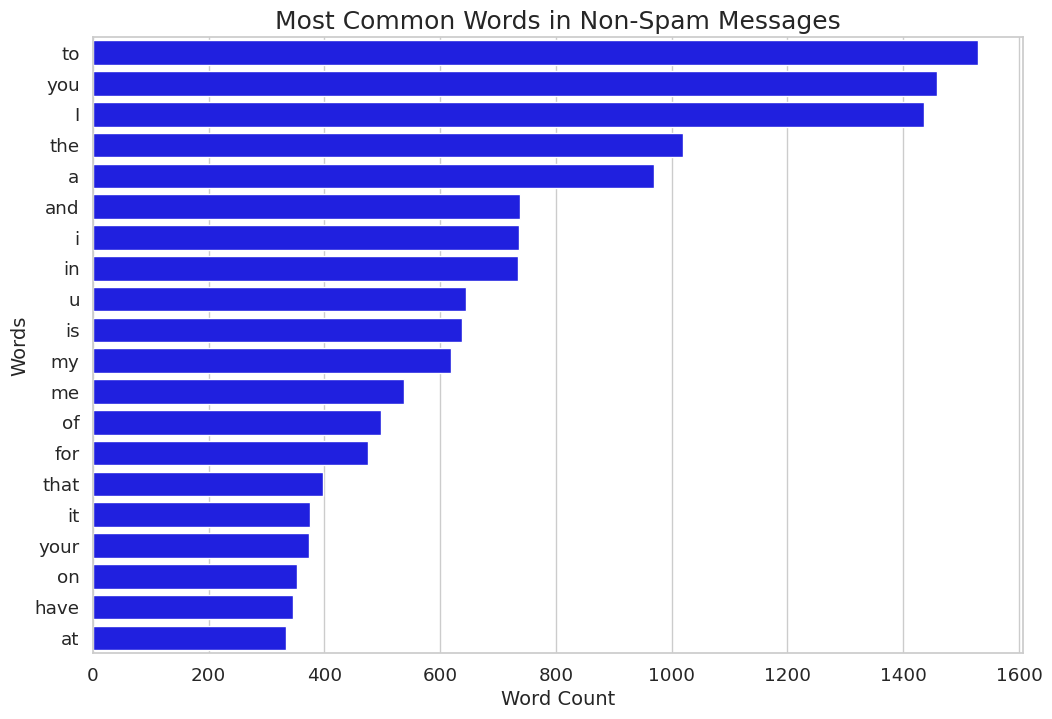
**2. Message Length Distribution**

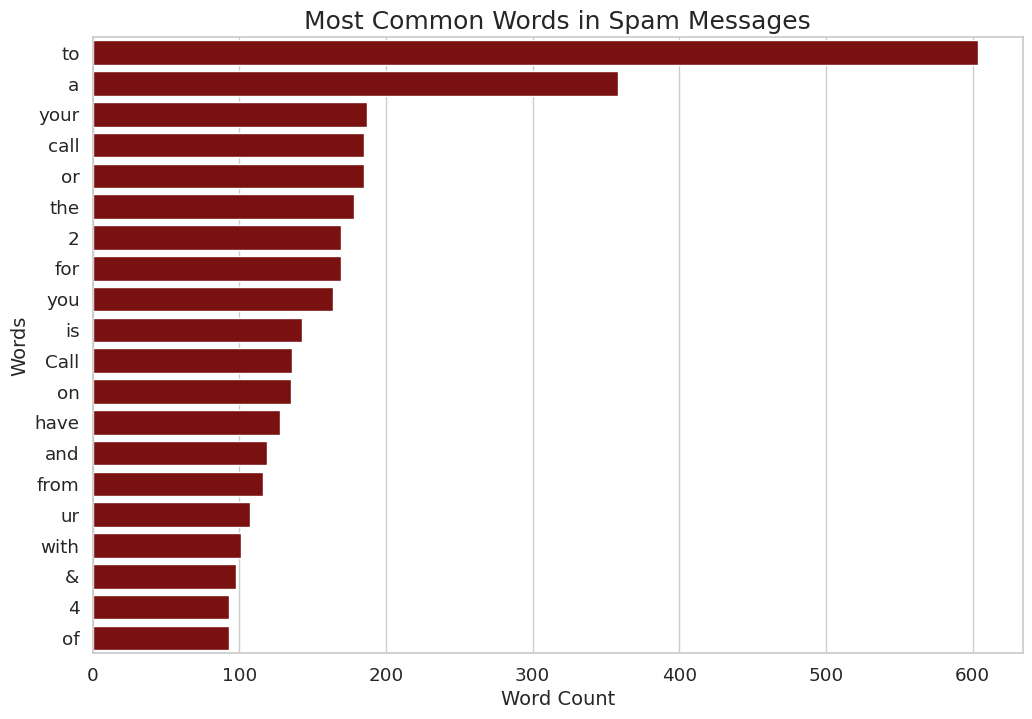
Examining the distribution of message lengths in both spam and ham transmissions is crucial. Histograms are utilized to visualize the frequency of messages across various lengths, allowing for the identification of patterns and potential correlations between message types and lengths.



**3. Common Words Analysis**

Further investigation focuses on determining the most common terms in spam and ham transmissions. The study offers insight on the unique linguistic patterns used in spam and non-spam communications using bar charts displaying top terms and their related counts.





This step aids in feature selection and comprehension of influential vocabulary for predictions.

**Model Training and Evaluation**

The model training and evaluation process in the spam message classification task involve several crucial steps:

**1. Data Transformation and Splitting**

The initial phase encompasses data preprocessing tasks, including text transformation using NLTK for stemming and tokenization. After transforming the text data, the dataset is split into training and test sets using **train\_test\_split()** from Scikit-learn. The transformed text is then converted into numerical features using **CountVectorizer()**, preparing it for the machine learning model.

**2. Logistic Regression Model Initialization and Hyperparameter Tuning**

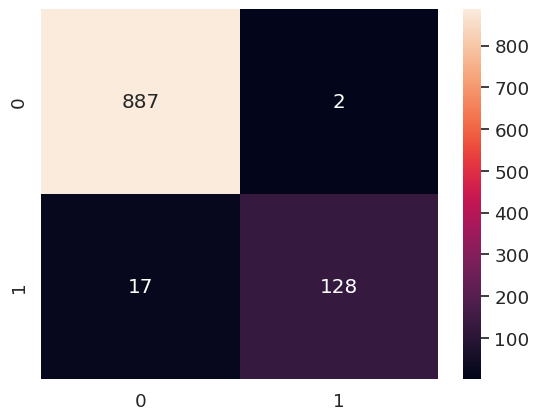
A logistic regression model is initialized using Scikit-learn's **LogisticRegression()** with the 'lbfgs' solver and a maximum iteration limit. Hyperparameter tuning is carried out through GridSearchCV, exploring a set of hyperparameters (**C** and **penalty**) to find the optimal configuration that enhances the model's performance.

**3. Model Training and Cross-Validation**

The GridSearchCV performs cross-validation on the training data, assessing the model's performance using different hyperparameter combinations. This process involves training the model multiple times on different subsets of the training data, validating it on the remaining subsets, and iterating to find the best hyperparameters that yield the highest performance.

**4. Performance Evaluation**

Upon completing the training phase, the best model, determined through cross-validation, is evaluated using the test dataset. Performance metrics such as precision and recall are calculated using **precision\_score()** and **recall\_score()** from Scikit-learn, respectively. The confusion matrix, visualized using **sns.heatmap()**, provides insights into the model's classification results, displaying true positives, true negatives, false positives, and false negatives.



**5. Model Persistence**

The trained and optimized model is persisted using joblib, allowing for future reusability without the need to retrain the model each time. This ensures easy access to the well-performing model for making predictions on new, unseen data.

**Machine Learning: Spam Message Classification**

The project, developed in a Python notebook within Google Colab, focuses on spam message classification using various libraries for data manipulation, visualization, and machine learning.

**Data Collection and Preprocessing**

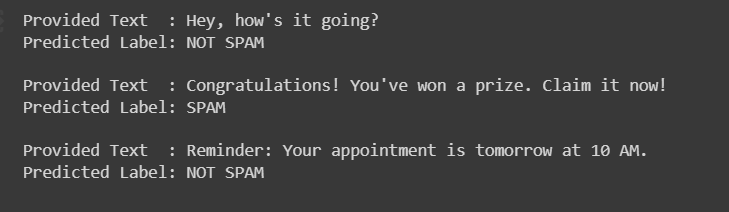
The project starts with data acquisition from Kaggle using a custom function, **get\_kaggle\_dataset()**, which facilitates the download and extraction of the SMS Spam Collection dataset. When the data is loaded into a Pandas DataFrame, it consists of labeled SMS messages that have been classified as "spam" or "ham" (non-spam). NLTK is used for initial preprocessing activities such as label encoding and text transformations to lowercase, tokenization, and stemming. To grasp label distribution, message length distribution, and identification of frequent terms in both spam and non-spam communications, statistical studies are performed, including visualizations using Seaborn and Matplotlib.

**Model Development and Training**

The machine learning model, constructed using Scikit-learn and logistic regression, undergoes rigorous training and evaluation. After analyzing message lengths and identifying frequent words in spam and non-spam messages, the CountVectorizer transforms the text data into numerical features. The dataset is split into training and test sets using **train\_test\_split()**. Hyperparameter tuning using GridSearchCV optimizes the logistic regression model. Performance evaluation metrics, including precision and recall scores, are computed to assess the model's efficacy in predicting spam messages. Confusion matrix visualization aids in understanding model performance on the test set.

**Model Deployment and Predictions**

The trained model is saved using joblib for future use and reloaded for making sample predictions. A set of sample texts, both spam and non-spam, is processed using the same transformations as the training data. The model predicts whether each message is spam or non-spam, providing predicted labels along with confidence scores. This final step demonstrates the model's capability in categorizing messages and its practical applicability in real-world scenarios.



**Preliminary Results**

The initial evaluation of the spam message classification model unveils compelling performance metrics that signify its proficiency in discerning spam messages:

**Precision: 98.46%**

Precision denotes the accuracy of the model in correctly identifying spam messages among all messages predicted as spam. Scoring an exceptional precision rate of 98.46%, the model showcases an extremely low false positive rate. This implies that when the model labels a message as spam, it is accurate 98.46% of the time, minimizing the instances of misclassifying non-spam messages as spam.

**Recall: 88.28%**

Recall, also known as sensitivity, signifies the model's capability to capture all actual spam messages. With a recall rate of 88.28%, the model demonstrates a commendable ability to identify a substantial portion of spam messages present in the dataset. While it captures a high proportion of spam messages, there's a minor fraction of actual spam that the model may not classify, highlighting a slight trade-off between precision and recall.

These preliminary findings underscore the model's impressive precision in accurately labeling spam messages, mitigating the false positive rate significantly. However, while maintaining a high precision, there's a balance with recall, capturing the majority of actual spam messages. Further iterations and optimizations could potentially fine-tune the model's performance for even better precision-recall equilibrium.

**Project Management:**

Implementation Status Report:

Work Completed:

* Description: As per our knowledge we have finished the project as there is no complications involved in this project.
* Responsibility:

Background Work, Dataset – Pranay

Detail Design, Analysis – Anil, Ahad Implementation – Ahad, Anil, Pranay

* Contribution:

Ahad – 100% (Implementation – 50%, Detail design, Analysis – 50%)

Anil – 100% (Detail design, Analysis – 50%, Implementation – 50%)

Pranay – 100% (Background work – 50%, Implementation – 50%)

Work to be Completed:

> Our work is completed by now.

> Issues/Concerns: No issues/concerns, all of our team members cooperated very well.

**References**

Gupta, M., Bakliwal, A., Agarwal, S., & Mehndiratta, P. (2018, August). A comparative study of spam SMS detection using machine learning classifiers. In *2018 eleventh international conference on contemporary computing (IC3)* (pp. 1-7). IEEE.

Shafi’I, M. A., Abd Latiff, M. S., Chiroma, H., Osho, O., Abdul-Salaam, G., Abubakar, A. I., & Herawan, T. (2017). A review on mobile SMS spam filtering techniques. *IEEE Access*, *5*, 15650-15666.

Gupta, S. D., Saha, S., & Das, S. K. (2021, February). SMS spam detection using machine learning. In *Journal of Physics: Conference Series* (Vol. 1797, No. 1, p. 012017). IOP Publishing.

GitHub-Link: