**SPAM MAIL DETECTION**

**Group : xx**

**Names of the group member :**

**1.------------------**

**2.-----------------**

**3.-----------------**

**4.-----------------**

**ABSTRACT :**

As we are in the world of digital evolution and the digital communications all over the world, the spam messages have become a risky issue, which is cluttering inboxes and also potentially leading to phishing and fraud. This document will take into the details of the development of a system which perform well in detecting the spam text, where it automatically classify the messages as spam or ham(non-spam). As we have used a dataset which has numerous amount of messages in it. Then we have implemented and done the evaluation of the models which in our case we chose three models : Naive Bayes, Logistic Regression, and Support Vector Machine (SVM). Where we have trained each model with the combination of some techniques that typically includes text preprocessing, feature extraction with TF-IDF vectorization, and modelling of the predictive models. Then the performance of the each model was tested using the performance metrics, which typically includes accuracy, precision, recall, and F1-score, which are represented through confusion matrices and classification reports. So this system generally has the goal of enhancing the email filtering solutions through the providing of efficient methods and reliable as well which are responsible for the detection of unwanted images , which are providing the user experience in the digital era of communication environments. This document particularly aligns to show case the methodologies employed, discusses the models' performances, and suggests directions for future improvements and implementations.

**INTRODUCTION :**

In this digital era of the communication, there is a need for the filtering of the messages or emails to spam or ham(non-spam), which has been posing a main challenge which is not only to the individual perspective but also to the security of the personal information. As we speak about the spam mails they can be range from harmless advertisements to malicious scams designed to deceive recipients into leaking the sensitive data. To avoid the scenario we need to have effective spam detection systems, which are generally important for managing the safety of the users and enhancing the overall efficiency of the online communication.

The main goal of this project is to develop a robust spam text detection system that can accurately differentiate between spam and non-spam ("ham") messages. This document will be going to present a brief overview of the system we have implemented which includes the collection of a labelled dataset and also employs various preprocessing techniques to text data and also getting the machine learning algorithms into the picture by creating them as a predictive models. These models are generally chosen due to their ability in getting the classification tasks done seamlessly.

We have utilized the dataset, which is having the labelled text images, then this dataset is going to subject towards some of the preprocessing tasks, which will convert the raw data into a type of format which is suitable for the machine learning modeling. Then after that feature extraction is generally done using the Term Frequency-Inverse Document Frequency (TF-IDF) technique, which has the ability of transforming of the text into a set of variables which are representing the importance of specific words within the messages. Three distinct classification models—Multinomial Naive Bayes, Logistic Regression, and Support Vector Machine (SVM)—were trained and evaluated based on their ability to classify messages accurately.

This document will take us into the world of spam text detection using machine learning. We'll explore how to prepare data for analysis, then delve into different modeling techniques. Finally, we'll put these models to the test and see how well they perform in identifying spam messages. By the end, you'll gain a solid understanding of how machine learning can automate and significantly improve the accuracy of spam filtering systems.

**MOTIVATION :**

The main thing here is that, the primary mode of connection which has essentially using the electronic messages are the sole responsible for the incoming spam messages. These unwanted or the spam mails not only disturbs the user experience but also works on posing notable amount of risks In terms of security. Spam messages may involve phishing attempts, virus links, fraudulent schemes, and heavy promotional content. The total amount and variety of spam makes manual detection and filtering inefficient and time-consuming, demanding automated systems that can effectively distinguish between legitimate and spam messages.

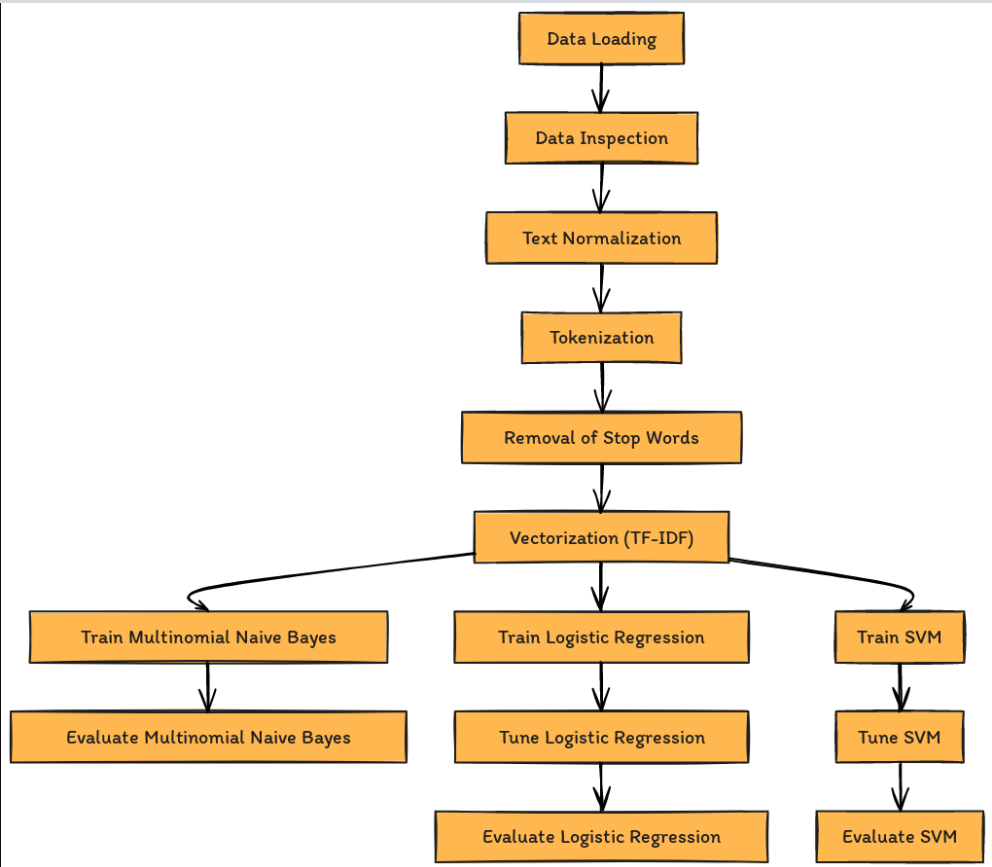
Spam detection has been in an one such area where active research and the development we are certainly looking now a days, which is ultimately to provide the security to the users and protect them in order to maintain the integrity of the systems. Before we used to have some traditional approaches which are rule based generally where the spam mails or the texts get filtered based on some set of rules, where it has some predefined criteria to filter the spam messages. However we all know how good the spammers are they come up with new approaches and tactics to send the spam mails that doesn’t effect with the inclusion of traditional filtering of the mails. So now as we speak about the inclusion of machine learning / Artificial intelligence this offers a dynamic way of solution by learning from the examples and it try to improve continuously on unseen data and the new tactics of the spammers. This adaptability makes machine learning models particularly suited for spam detection, where new patterns and tactics continually emerge.

The main motivation of this project hails from the desire to improve the accuracy and the efficiency of the spam detection systems using advanced machine learning techniques, where by implementing them and comparing with different models, this project also focus on getting the strengths and weakness of each approach under varying conditions and configurations. The goal is to develop a spam detection system that not only performs with high accuracy but also scales effectively with increasing data volumes and complexity. So generally this system will helps the users and the administrators in automating the spam mails in not getting into the main inbox. Which further concludes that it is offering security across communication channels.

**WORKFLOW DIAGRAM :**

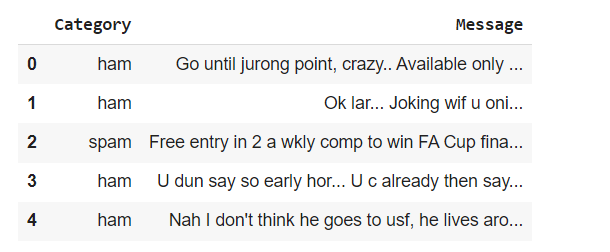
The below is the workflow diagram for the entire project where each and every step is discussed further in the subsequent sections.

It starts with the loading of the data and then doing some preprocessing tasks and then feeding it to the models and at last evaluating the performance of the models.

****

**DATASET AND EXPLORATORY DATA ANALYSIS(EDA) :**

The below is the dataset and the some of the descriptions of the dataset we made as part of the preprocessing analysis and as we move into the dataset to get to know more about the dataset. The dataset we are using has the collection of messages, which are labelled as spam or not spam(ham). This dataset has some numerous amount of text messages that has been collected from various sources, where they have been labelled.



From the above image , you can see the top 4 rows in the dataset and their attribute names on it. Basically this dataset has 5572 messages in it, which all of them are labelled and this huge dataset is sufficient enough to train and test the various machine learning algorithms effectively.

The two attributes as you see in the above image :

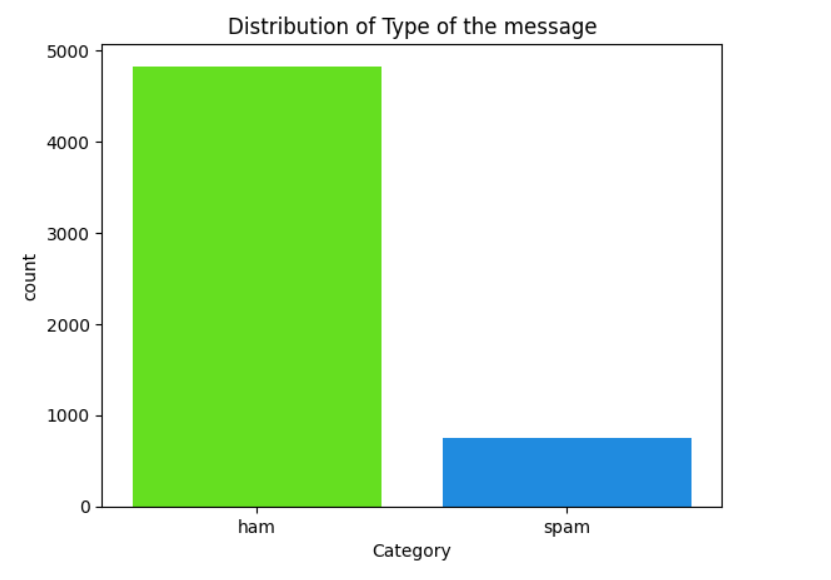
**Category:** This is the target variable, with each message labeled as 'spam' or 'ham'.

**Message:** This variable contains the text of the SMS message itself, which is the input used for feature extraction and model training.

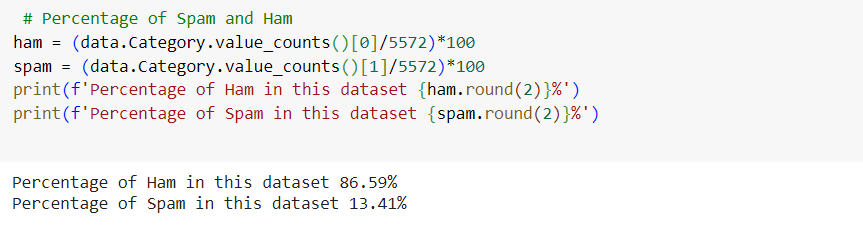
As we can see each message is stored in the form of plain text so there is a need of transforming it into some suitable format where our machine learning algorithms or the models can get trained effectively and able to predict the spam or ham. Where it typically involves cleaning, tokenization, and vectorization.

The dataset is imbalanced, with a much higher proportion of 'ham' messages compared to 'spam'. This imbalance reflects typical real-world conditions but poses challenges for model training, necessitating specific techniques to ensure model robustness.

The below attached image is the visual representation of the above stated statement which is clearly indicating the fact that there are more ham messages compare to the spam messages. This makes the dataset as an imbalance dataset.



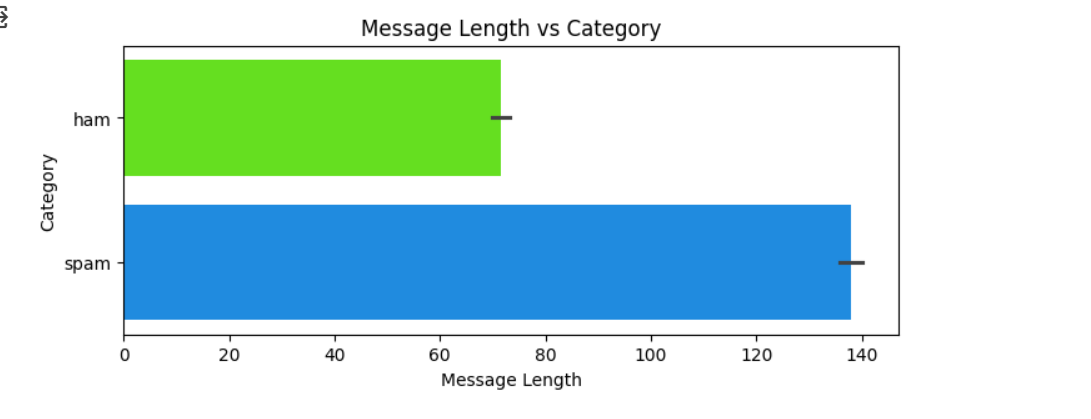
The below is the percentage of the proportion of the dataset, based on spam and ham.



Next as part of the EDA we have also calculated the length of the text in every message in the dataset and made it as a separate column in the dataset. The below is the result we got with respect to the length of the text.



As you can see we calculated the length of the text, now for the analysis part to find the lengths of the text with respect to the labelling of the message , In other words the below image is demonstrating the message length with respect to category.



So from the above visualization we can clearly state that the spam messages are high in length compared to the ham messages , we mean the messages which are potentially a spam one is typically the highr in length.

**DATA PREPROCESSING :**

The data preprocessing is such an important task as part of training the model and all the things are generally depends on how good our data is there to feed to it to the model, so pre processing is one such task where we transform our data inorder to make it adaptable to the chosen algorithm or the machine learning models. As part of our project we have done several preprocessing task which particularly includes :

Firstly some of the basic tasks that includes the removing of null values and the duplicates from the dataset.

Then we have performed the text normalization which involves the conversion of messages to the lower case, which is to make sure the text is uniform and also prevent the same words in different cases from being treated as different tokens.

Now as part of the tokenization, the text of each message was split into individual words or "tokens". This step breaks down the text into pieces that can be analyzed and used for feature extraction.

Next we moves our focus on to the removal of stop words which generally involves the Commonly occurring words in the English language that offer little value in distinguishing between spam and ham, which are "and", "is", "the", were removed in this phase of the preprocessing. So in return this makes the dataset even more efficient to feed it to the models.

The Term Frequency-Inverse Document Frequency technique was used to convert text tokens into a numeric format which helps in getting the importance of words within messages relative to the entire dataset. This method helps highlight the words that are important for classifying a message as spam or ham.

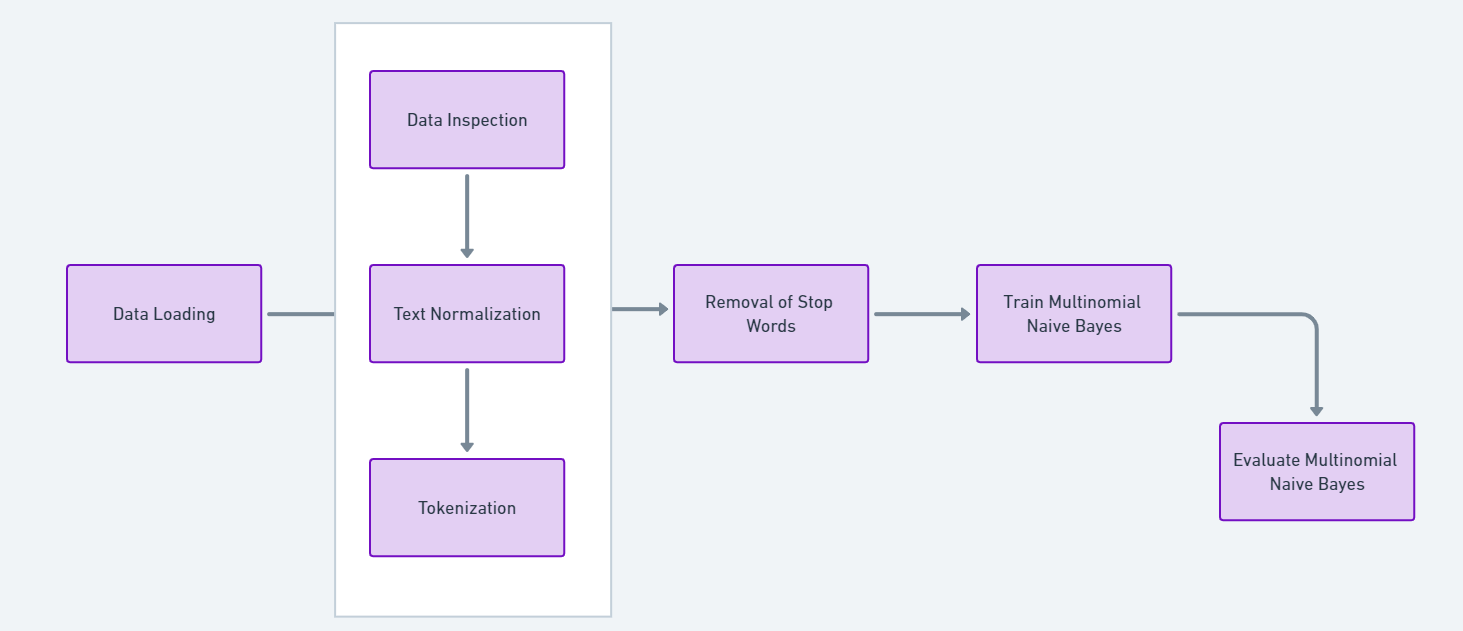
So all the above are the preprocessing steps we followed in feeding the efficient dataset to the models which we are going to implement. Which also allowing the algorithms to focus on the most informative features of the data.

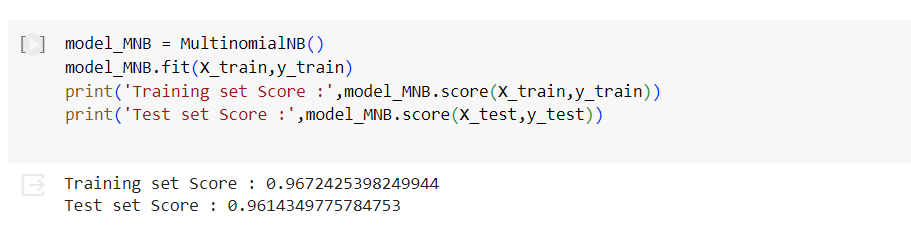
**MACHINE LEARNING MODELS :**

We have used three distinct machine learning models, as we name them they are : Multinomial Naive Bayes, Logistic Regression, and Support Vector Machine (SVM). So here each of the chosen model is there in our project based on their strengths as part of the handling of the text classification tasks. Below we will be having a discussion on the model implementation.

1. **Multinomial Naive Bayes (MNB):**

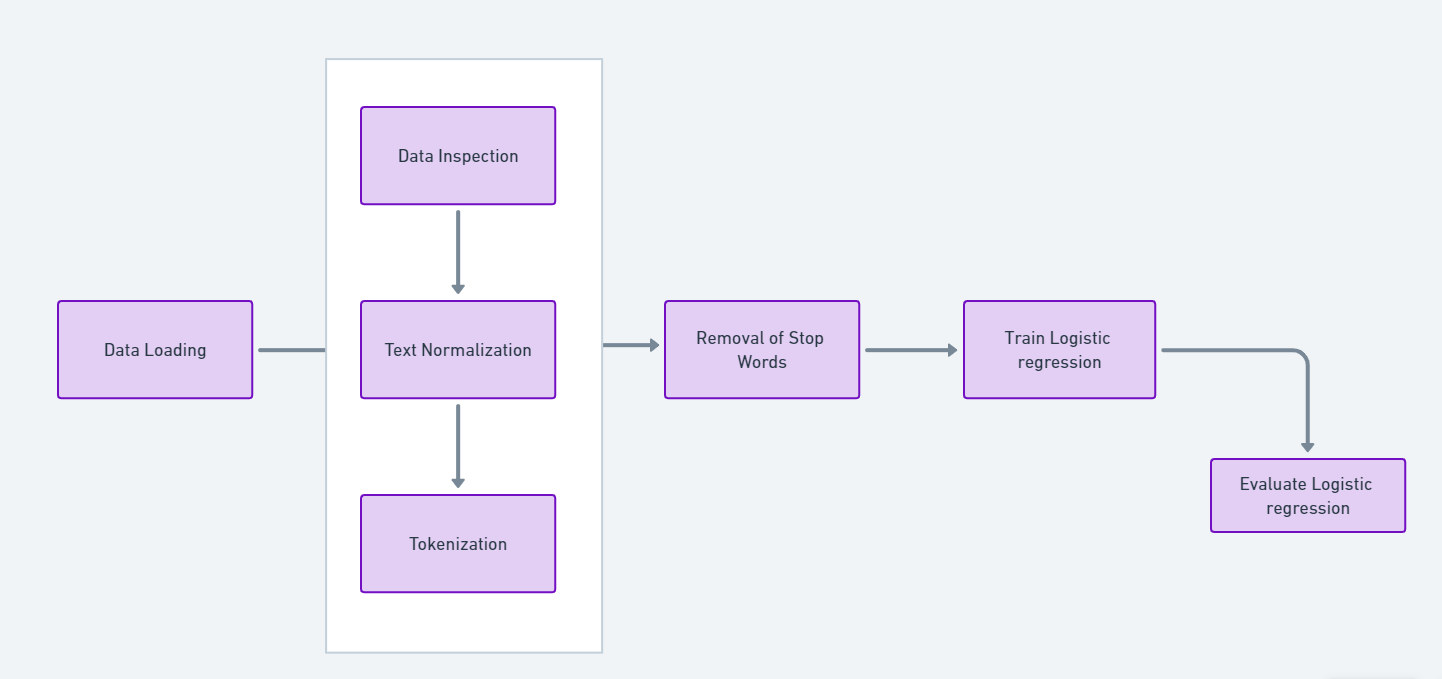
* Naive Bayes classifiers are probabilistic models that apply Bayes' Theorem, which assumes independence between the predictors. Multinomial Naive Bayes is particularly suited for classification with discrete features (e.g., text classification), where it models word counts and modifies the forecast based on word frequency.
* **Implementation :**
* **Vectorization :** The TF-IDF vectorization was used to transform text data into a format suitable for the model.
* **Training :** The model was trained using the MultinomialNB() class from sklearn.naive\_bayes.
* **Evaluation :** The model performance is demonstrated using the training and the testing datasets, in which their results are based on the accuracy, confusion matrix, and classification report.

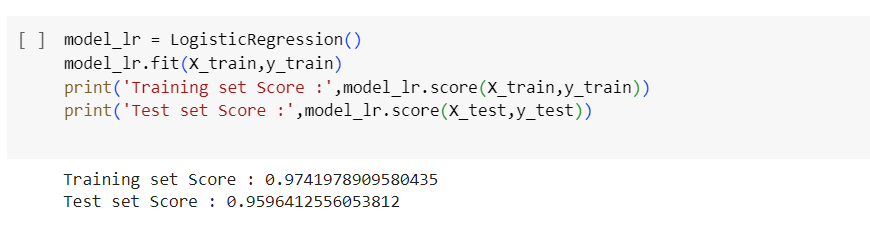




1. **Logistic Regression:**

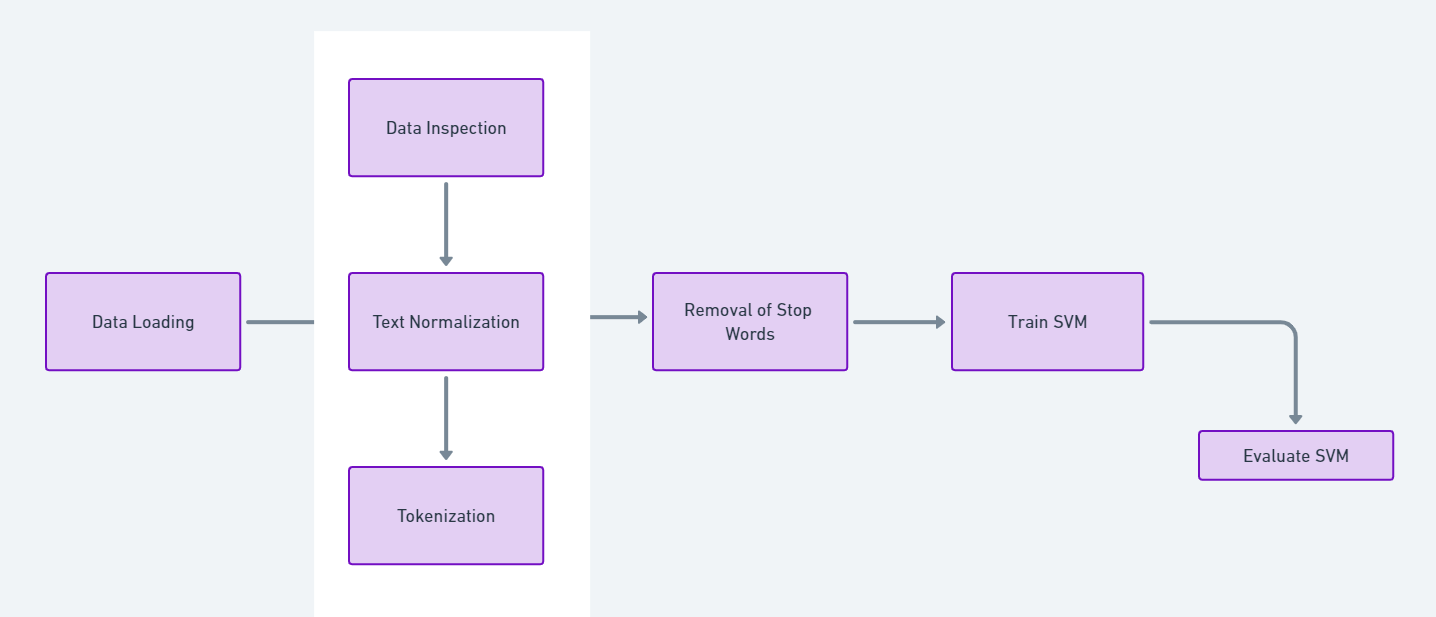
* In general the logistic regression is used for the tasks of the binary classification, where it estimates probabilities using a logistic function, which is particularly useful for cases where there is a clear delineation or boundary between the classes.
* **Implementation :**
* **Vectorization :** The TF-IDF vectorization was used to transform text data into a format suitable for the model.
* **Training :** we have used the LogisticRegression() class from sklearn.linear\_model, with the model trained on the training data.
* **Evaluation :** The model performance is demonstrated using the training and the testing datasets, in which their results are based on the accuracy scores and a confusion matrix, alongside a detailed classification report to analyze precision, recall, and F1-score.

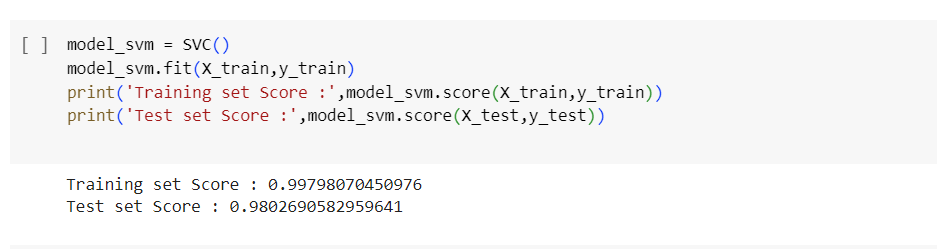




1. **Support Vector Machine (SVM):**

* As we describe about the SVM model , it is a powerful technique in terms of the classification tasks, which in practical terms works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable.
* **Implementation :**
* **Vectorization :** The TF-IDF vectorization was used to transform text data into a format suitable for the model.
* **Training :** The SVC() class from sklearn.svm was configured and trained with the kernel type set to 'linear' for better handling of linearly separable data.
* **Evaluation :** The model performance is demonstrated using the training and the testing datasets, in which their results are based on the accuracy, confusion matrix, and classification report



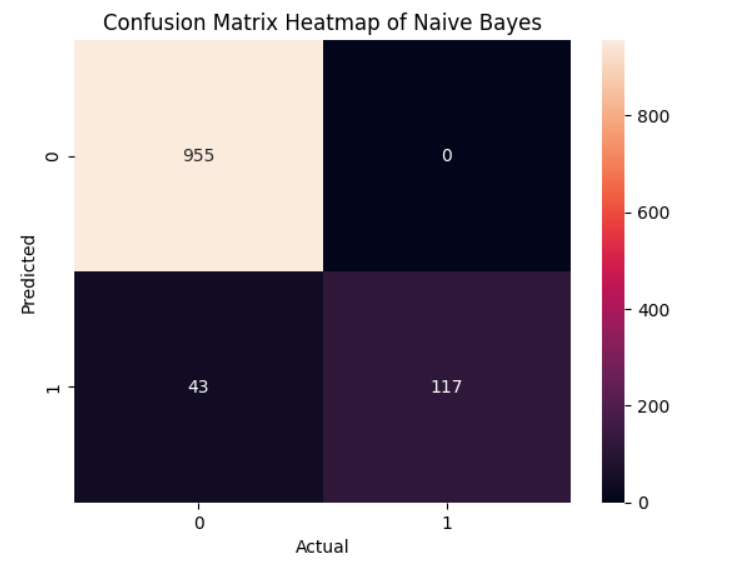


**MODEL EVALUATION :**

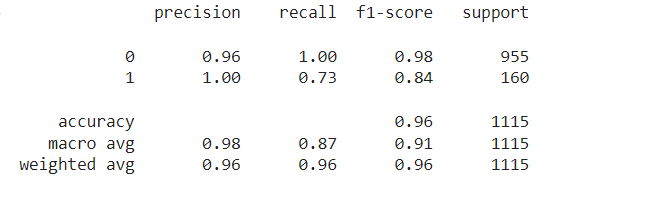
Below we will be noting all the performance metrics we got from the three distinct models, which are thoroughly assessed using a variety of metrics including accuracy, precision, recall, and F1-score, as well as confusion matrices for a detailed error analysis.

**Multinomial Naive Bayes :**

The below is the confusion matrix of the mentioned model :

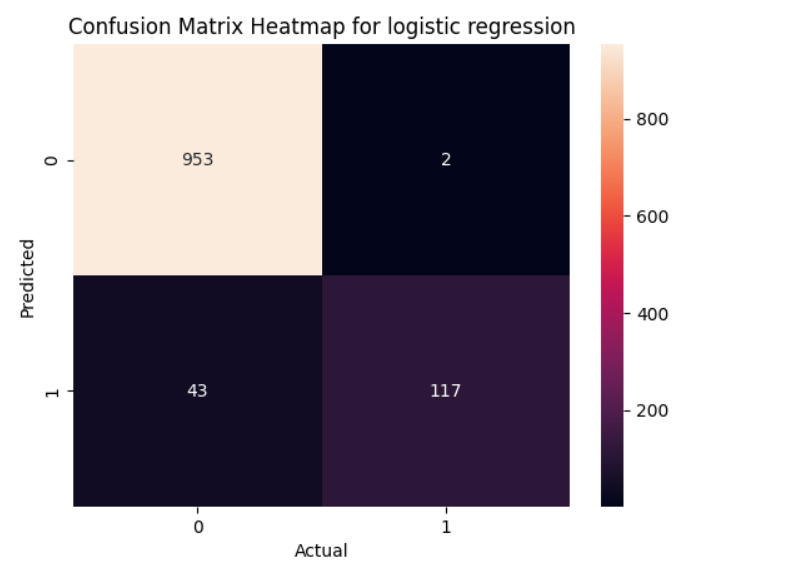


The below is the classification reports with respect to the Naïve Bayes :

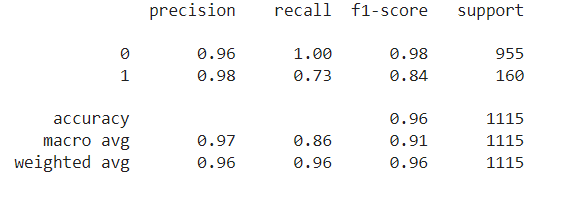


**Logistic Regression :**

The below is the confusion matrix of the mentioned model :

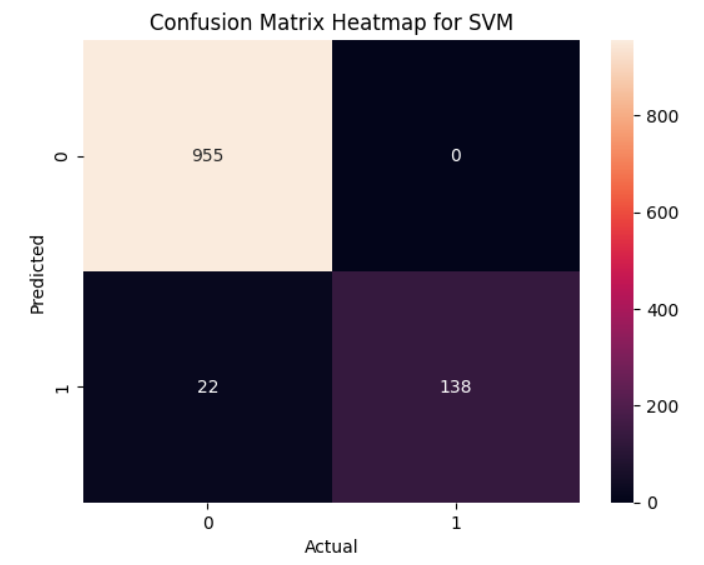
****

Below we can find the classification report with respect to the logistic regression model.

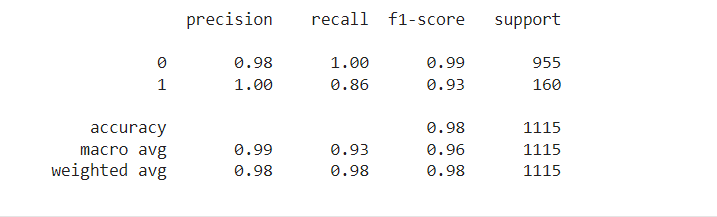


**SVM :**

The below is the confusion matrix of the mentioned model :

****

Below is the image that depicts the classification report with respect to SVM model.



As from the above performance metrics we can state that SVM model has the highest overall performance among the three models which we have chose to implement. Which achieved nearly 98% of the accuracy on the test set, which showcases the great balance between the recall and precision, in particularly for the spam detection. Notably, SVM managed to achieve high precision and recall for spam messages without misclassifying any ham messages as spam, indicating a strong ability to generalize from the training data.

Multinomial Naive Bayes and Logistic Regression also performed well, particularly in terms of precision, but they had lower recall for spam messages compared to SVM, indicating a higher number of spam messages being misclassified as ham.

High precision and recall scores for the ham class across all models showed the models' overall strong ability to accurately classify ham communications. To ensure that fewer spam messages pass through the filtering process, the main area for improvement is to raise the recall for spam without compromising the high precision.

**CONCLUSION :**

As we conclude the project, the spam text detection project demonstrated the effectiveness of machine learning techniques in distinguishing between spam and non-spam messages. As we implemented this project, where we have used three different machine learning models : Multinomial Naive Bayes, Logistic Regression, and Support Vector Machine (SVM). Where each model was feeded with many testing and evaluations using the performance metrics, which includes the accuracy, precision, recall, and F1-score.

The SVM model has shown more efficiency and effectiveness in classification of spam and ham text messages, where it achieved a huge accuracy of 98.03% on the test set, along with the highest precision and recall scores for spam detection. This superior performance highlights SVM's robustness and its ability to generalize well from the training data to unseen messages. Where on the other side the models Logistic regression and Multinomial Naïve Bayes also performed well by gaining the overall accuracy of nearly 96%, which is showcasing the ability of these two models as well in the classification of text.

As we conclude here, we can state that the development and the implementation of the spam detection models have significant implications for improving online security and user experience. By automating the process of identifying and filtering spam messages, these models help safeguard personal information and maintain the integrity of communication channels.

**FUTURE WORK**

As we speak about the future work that can be aligned to this spam detection models, we can further include the deep learning models, where it includes approaches such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs) could be explored which has the ability to understand the contextual relationships in text data better.

Also using word embedding models can provide a more detailed understanding of the text semantics which the models are : Word2Vec or GloVe. Word embeddings capture the semantic relationships between words, potentially improving the model’s ability to generalize from the training data to unseen messages.

There is a requirement of the updation of the dataset dynamically as we can see the evolving tactics from the spammers . so there is a requirement of training the models on the dataset which are continuously updating is one such are the future work might look into it.

Expanding the scope of the models to handle messages in multiple languages and adapt to different messaging platforms can significantly increase their utility

**REFERENCES :**

1. Patrice Samuel Rompas, Riza Satria Perdana, “Securing Confidential Documents in Local Network Using an Email Filtering Technique”, 2018 International Workshop on Big data and Information Security(IWBIS)
2. Wuxu Peng, Linda Huang, Julia Jia, Emma Ingram, “Enhancing the Naive Bayes Spam Filter through Intelligent Text Modification Detection”, 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications.
3. ]Aakash Atul Alurkar, Sourabh Bharat Ranade, Shreeya Vijay Joshi, Siddhesh Sanjay Ranade, “A proposed data science approach for email spam classification using machine learning techniques”, 2017 Internet of Things Business Models, Users, and Networks.
4. Ghulam Mujtaba, Liyana Shuib, Ram Gopal Raj, Nahdia Majeed; Mohammed Ali Al- Garadi, “Email Classification Research Trends: Review and Open Issues”, IEEE Access, 2017, Vol.5, Pg-9044 9064.
5. Fazira Ku Azir, “Multi Phishing Email Classification”, Journal of Theoretical and Applied Information Technology 20th January 2016. Vol.83. No.2
6. Ammar Yahya daeef, R. Badlishah Ahmad, Yasmin Yacob, Naimah Yaakob, Kunurul, Fazira Ku Azir, “Multi Phishing Email Classification”, Journal of Theoretical and Applied Information Technology 20th January 2016, Vol.83, No.2
7. G. Stringhini, C. Kruegel, and G. Vigna, “Detecting spammers on social networks,” in Proc. ACSC, Austin, Texas, 2010, pp. 1–9.
8. V. Metsis, I. Androutsopoulos, and G. Paliras, “Spam filtering with Naive Bayes”,Proc. of the 3rd CA, 2006, pp. 1–5.
9. Drucker H, Wu D, Vapnik VN. Support vector machines for spam categorization. IEEE Trans Neural Netw. 1999;10(5):1048–54. <https://doi.org/10.1109/72.788645>.
10. Blanzieri E, Bryl A. A survey of learning-based techniques of email spam fltering. Artif Intell Rev.