VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY

UNIVERSITY OF ECONOMICS AND LAW

FINAL PROJECT

SUBJECT: DATA MINING

PROJECT: Build a model capable of automatically detecting and classifying spam emails

LECTURER	**:**	**M.S. Phan Huy Tam**
STUDENT	**:**	**Nguyen Quoc Huy**
CLASS	**:**	**231CN0601**
STUDENT ID	**:**	**K214142068**

Ho Chi Minh City, January 10 th, 2024

CONTENTS

Project Summary

The main goal of the account is to build a model or type of system that is capable of automatically detecting and classifying spam emails.

There are various methods and algorithms that can be used to perform this task, I have applied supervised algorithm computing to build an email classification spam model. I have tried many algorithms SVC, MNB, XGB, RFC, ADB, LR, GBC, DTC, KNC... Top models Naive Bayes, Support Vector Machines (SVM), Random Forest are used for text classification.

To evaluate model performance, methods such as accuracy, testing accuracy, training accuracy, and mixture matrix (confusion matrix) are used to measure accuracy and performance. of model types. The ultimate goal of the problem is to build a spam email classification model capable of accurately detecting and classifying spam emails, thereby helping users save time and prevent unwanted or malicious emails. damaging their mailboxes.

1. **Describe the dataset and the problem statement, what is the nature of this case?**

A common dataset used for spam email classification typically contains textual content from emails along with labels indicating whether they are spam or non-spam (referred to as ham). The dataset consists of two main components:

Textual Content (Features): This includes the body of the emails, which serves as the primary information used for classification. The text data contains various patterns, keywords, phrases, and potentially formatting peculiarities that are indicative of either spam or ham emails.

Labels (Target): Each email in the dataset is assigned a label indicating whether it is categorized as spam or ham. These labels serve as the target variable that the classification model aims to predict.

The objective of this dataset is to develop a machine learning model capable of accurately classifying emails into spam or non-spam categories. The nature of this case is a binary text classification problem, where the input is text data and the output is a binary label.

2. **Does data have any defects or issues? State the solution if any**

After examining the dataset, a few potential issues or defects can be identified:

Unnamed Columns with Null Values: The dataset contains three columns ('Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4') that mostly consist of null values. These columns do not seem to provide valuable information for the spam classification task. Solution: It is advisable to drop these columns from the dataset as they are likely irrelevant.

Encoding and Formatting Issues: The dataset uses 'latin-1' encoding. There might be issues related to text formatting or special characters that are not properly encoded. Solution: Ensure consistent encoding (like UTF-8) when processing the text data. Additionally, a preprocessing step to handle or remove special characters may be necessary.

Imbalanced Data: It's common in spam classification datasets to have an imbalance between the number of spam and ham messages, with ham messages often being more prevalent. Solution: Techniques such as resampling the dataset, using synthetic data generation (like SMOTE), or applying different weights to the classes during model training can help address this issue.

Text Preprocessing Needs: Text data typically requires various preprocessing steps to convert raw text into a format that's more suitable for analysis. Solution: Implement preprocessing steps such as tokenization, lowercasing, removing stop words, stemming or lemmatization, and handling of abbreviations and slang.

Feature Extraction: The raw text needs to be converted into a set of numerical features before it can be used in machine learning models. Solution: Use techniques like Bag of Words, TF-IDF, or word embeddings (like Word2Vec or GloVe) for feature extraction.

Model Overfitting: There is a risk that the model might overfit the training data, especially if the dataset is not large. Solution: Use techniques like cross-validation, regularization, and model selection based on validation data performance to mitigate overfitting.

In summary, while the dataset is fundamentally sound for the task of spam classification, addressing these issues through appropriate data preprocessing, feature engineering, and careful model selection is crucial to build an effective and robust classifier.

3. What kind of model could be used in this case? Explain!

When it comes to classifying spam emails, there are several types of models that can be used, each with its own strengths and considerations. Here are some popular approaches:

Naive Bayes Classifier: This is a probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features. It's particularly suited for text classification and is widely used for spam email filtering.

Advantages:

- **Simple and effective:** The Naive Bayes model has a straightforward structure and is easy to implement. It works well with high-dimensional data and is particularly suited for text classification tasks, making it widely used for spam email filtering.
- **Performs well with small datasets:** The Naive Bayes model can perform well even with small datasets, which is useful when the available training email samples are limited.
- **Feature independence assumption:** The Naive Bayes classifier assumes that the features (words or attributes) in the email are conditionally independent

given the class label (spam or ham). This assumption simplifies the modeling process and allows for efficient computation.

- **Probability-based classification:** The classifier calculates the probabilities of a given email belonging to each class and assigns it to the class with the highest probability. It estimates the required probabilities using Bayes' theorem and the training data.
- **Good performance with text data:** Naive Bayes classifiers are well-suited for text classification tasks, as they can handle high-dimensional and sparse data commonly found in text documents. They can effectively capture the occurrence of specific words or patterns that are indicative of spam or non-spam emails.

Support Vector Machine (SVM): SVMs are effective in high-dimensional spaces and in cases where the number of dimensions exceeds the number of samples. They work by finding the hyperplane that best separates the classes in the feature space.

Advantages:

- **Effective in high-dimensional spaces:** SVMs are known to be effective in high-dimensional feature spaces, especially when the number of dimensions exceeds the number of samples. In practice, text data often has high dimensionality due to the presence of various words and linguistic features.
- **Memory efficient:** SVMs use a subset of data points to determine the best separating hyperplane. As a result, they are memory efficient compared to storing the entire training dataset.
- **Robust against overfitting:** SVMs have the ability to resist overfitting, which means the model is not overly tuned to fit the training data and can generalize well to new data.
- **Margin-based separation:** SVMs aim to find the hyperplane that maximizes the margin between the two classes. The margin represents the distance between the hyperplane and the closest data points of each class. By maximizing the margin, SVMs achieve a clear separation between spam and non-spam emails.
- **Kernel trick for non-linear separation:** SVMs can efficiently handle non-linearly separable data by using the kernel trick. This technique allows SVMs to implicitly project the data into a higher-dimensional feature space where a linear separation can be achieved.
- **Effective with limited training samples:** SVMs perform well even when the number of training samples is smaller than the number of features. This is especially beneficial in cases where collecting a large number of labeled emails for training is challenging.

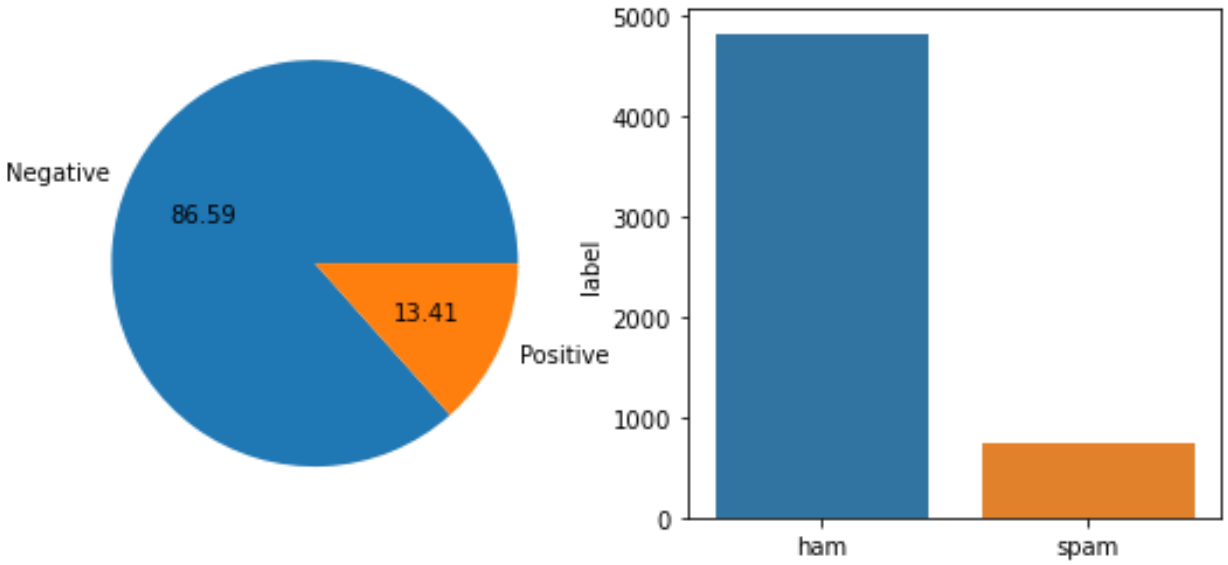
Random Forest: This is an ensemble learning method that operates by constructing a multitude of decision trees during training. For classification tasks, the output is the mode of the classes of the individual trees.

Advantages:

- **Handles numerical and categorical data:** Random Forest can handle both numerical and categorical data, which is useful when dealing with email attributes such as text content as well as other attributes like email addresses, subject lines, timestamps, etc.
- **Performs well with large and high-dimensional datasets:** Random Forest can handle large datasets with high dimensionality without sacrificing performance. It builds a collection of independent decision trees and combines the results to make final predictions.
- **Provides estimates of feature importance:** Random Forest can provide insights into the importance of different features for classification, helping to understand how specific attributes contribute to the classification decisions.
- **Ensemble of decision trees:** Random Forest is an ensemble learning method that combines multiple decision trees. Each tree is trained independently on a random subset of the training data and a random subset of features. The final prediction is made by aggregating the predictions of all the individual trees.
- **Robust against overfitting:** Random Forest models are less prone to overfitting compared to single decision trees. The randomness in the selection of training samples and features helps to reduce overfitting and improve generalization to unseen data.
- **Feature importance estimation:** Random Forest can provide estimates of feature importance based on how much each feature contributes to the overall performance of the model. This information can help identify the most relevant features for spam classification, allowing for better understanding and interpretation of the model.

When selecting a model, it is important to take into account various factors such as the size and characteristics of your data, the computational resources at your disposal, and the level of interpretability required. It is often advantageous to begin with simpler models like Naive Bayes or Logistic Regression and then gradually explore more complex models if needed. Additionally, feature engineering techniques such as TF-IDF or word embeddings for text data and fine-tuning the model parameters play a critical role in achieving optimal performance.

4. Perform data exploratory analysis



Conclusion: As seen Data is Imbalanced here first. I process with imbalanced data

4.1. Data Analysis

4.1.1. Feature Engineering

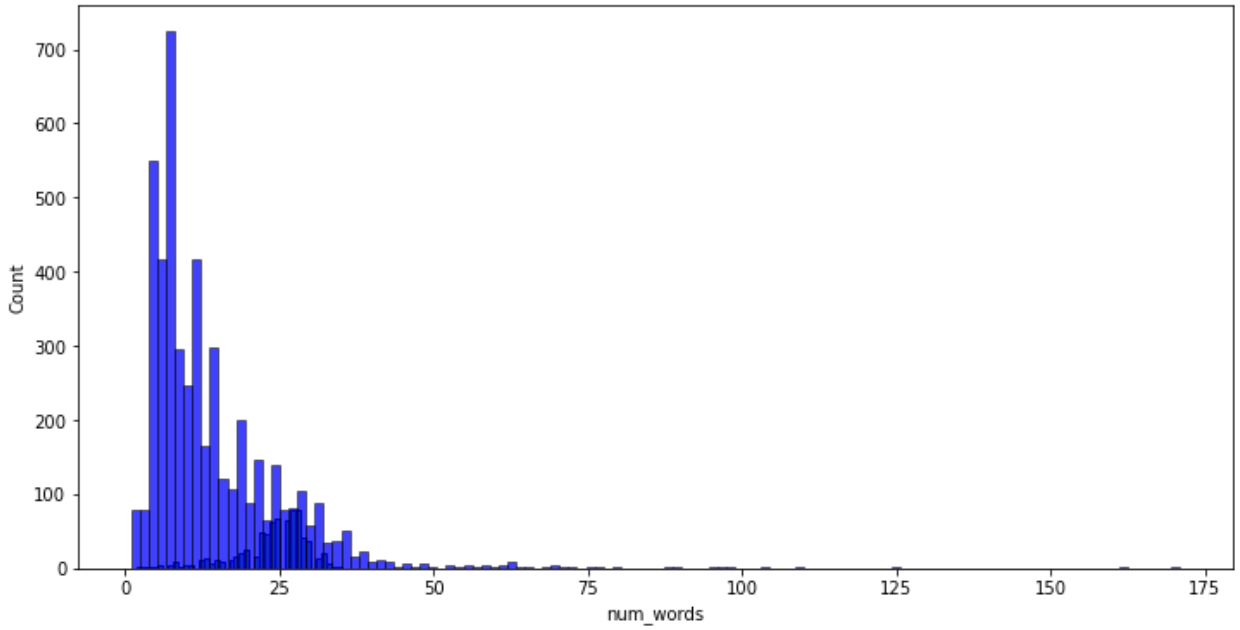
Out[22]:

	label	text	num_char	num_words	num_sen
0	ham	Go until jurong point, crazy.. Available only ...	111	20	2
1	ham	Ok lar... Joking wif u oni...	29	6	2
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	155	28	2
3	ham	U dun say so early hor... U c already then say...	49	11	1
4	ham	Nah I don't think he goes to usf, he lives aro...	61	13	1
...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	161	30	4
5568	ham	Will Ì_ b going to esplanade fr home?	37	8	1
5569	ham	Pity, * was in mood for that. So...any other s...	57	10	2
5570	ham	The guy did some bitching but I acted like i'd...	125	26	1
5571	ham	Rofl. Its true to its name	26	6	2

5572 rows × 5 columns

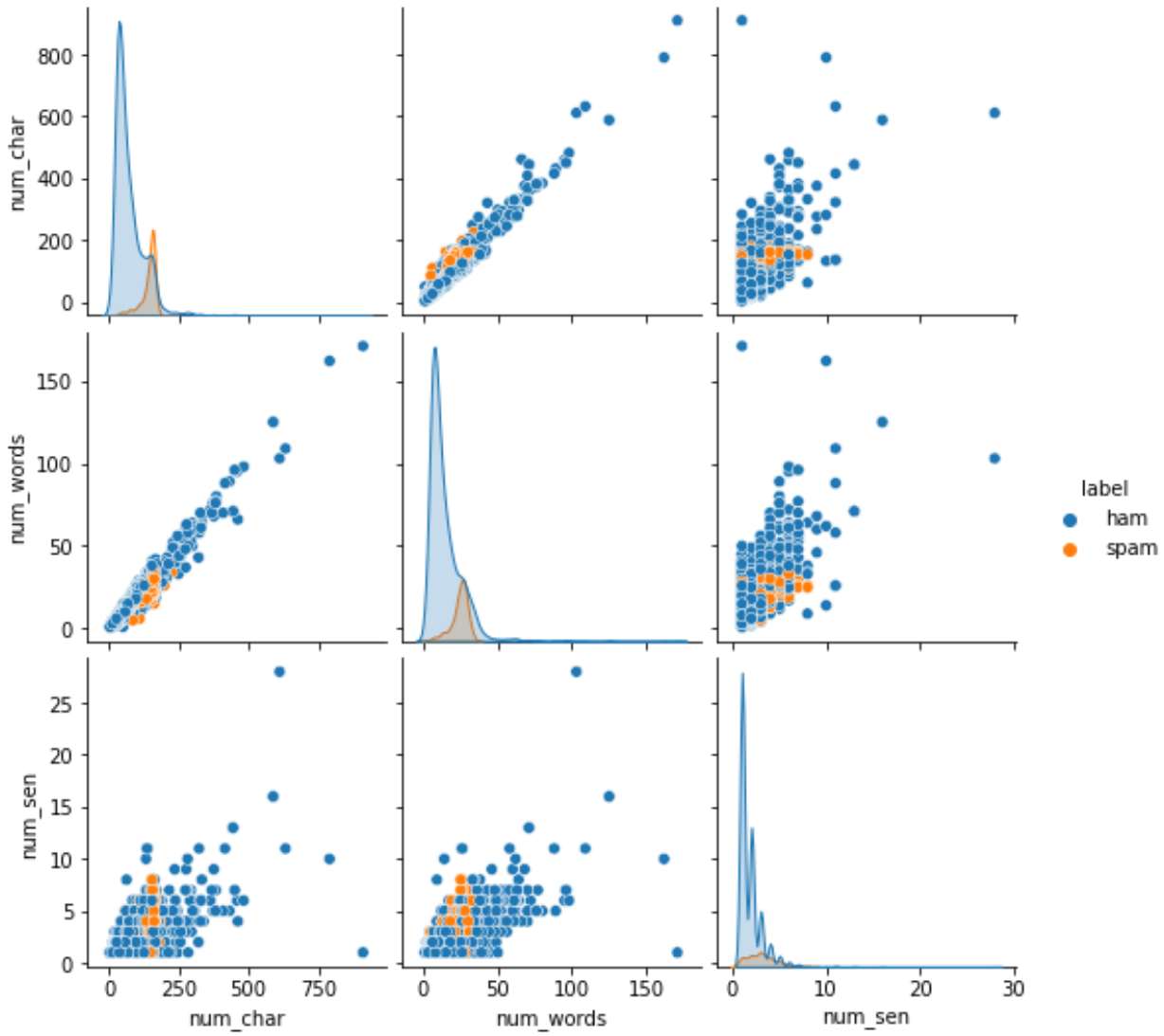
Conclusion: I Create Seperate features to extarct Some Information

- Total No of Characters
- Total No of Words
- Total No of Sentences



Conclusion: The chart shows the number of spam emails and is categorized by word count. On the x-axis is the number of words in the email, from 0 to 175. On the y-axis is the number of emails.

- Spam color : Yellow
- Ham color : Blue



Conclusion:

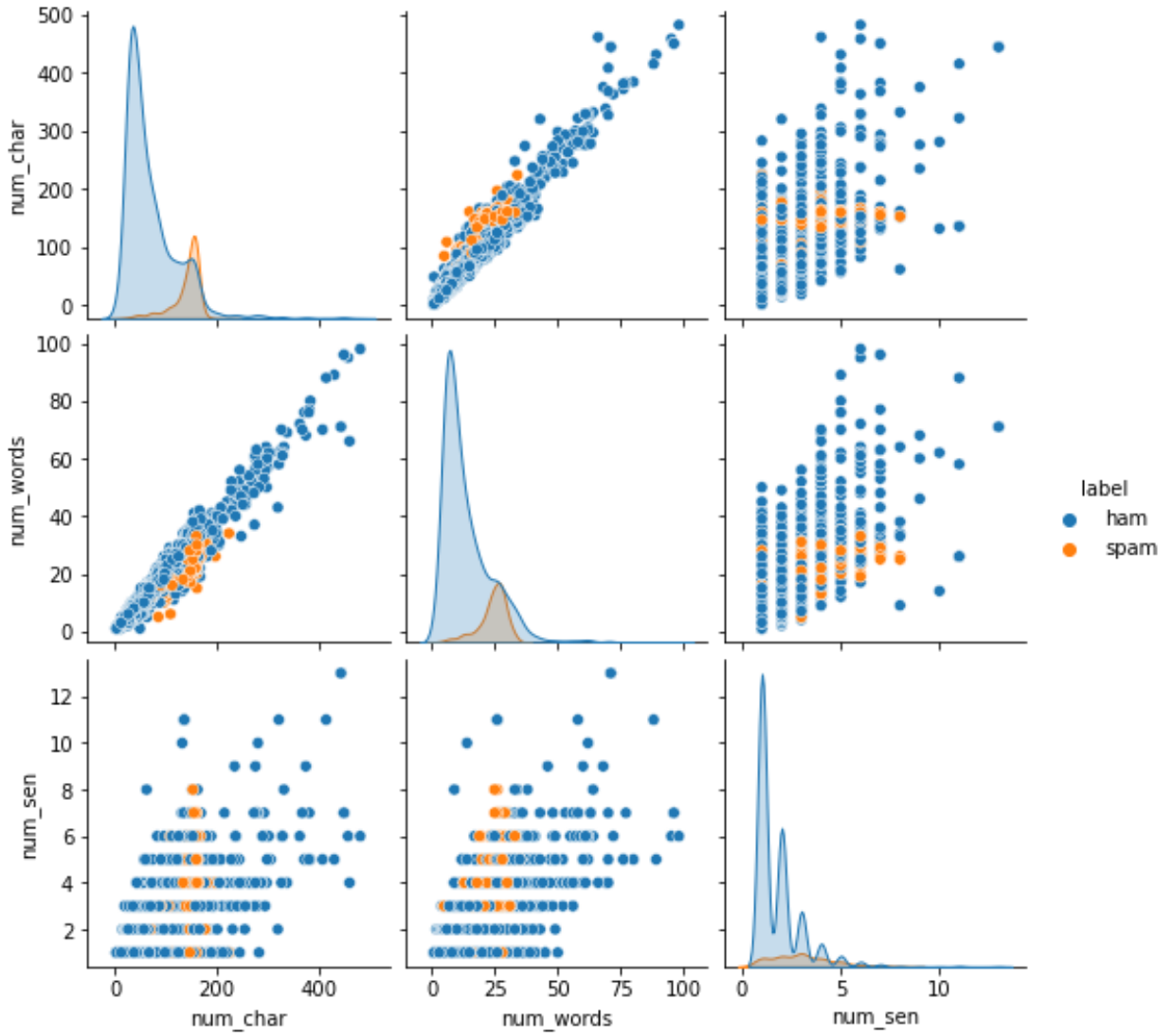
- Number of characters: Spam messages tend to have more characters than spam messages, as shown on the density plot and scatter plot. This may be because spam messages often contain a lot of advertising information or attractive invitations.
- Word count: Spam messages also tend to have more words than ham messages, but not as obvious as the character count. There are some ham messages with high word counts, possibly because they contain a lot of conversation or Q&A content.
- Number of sentences: Spam messages typically have fewer sentences than ham messages, as shown on the density plot and scatter plot. This may be because spam messages often use a lot of exclamation or question marks to create a sense of urgency or curiosity.

4.1.2. Handle Outliers

Out[33]:

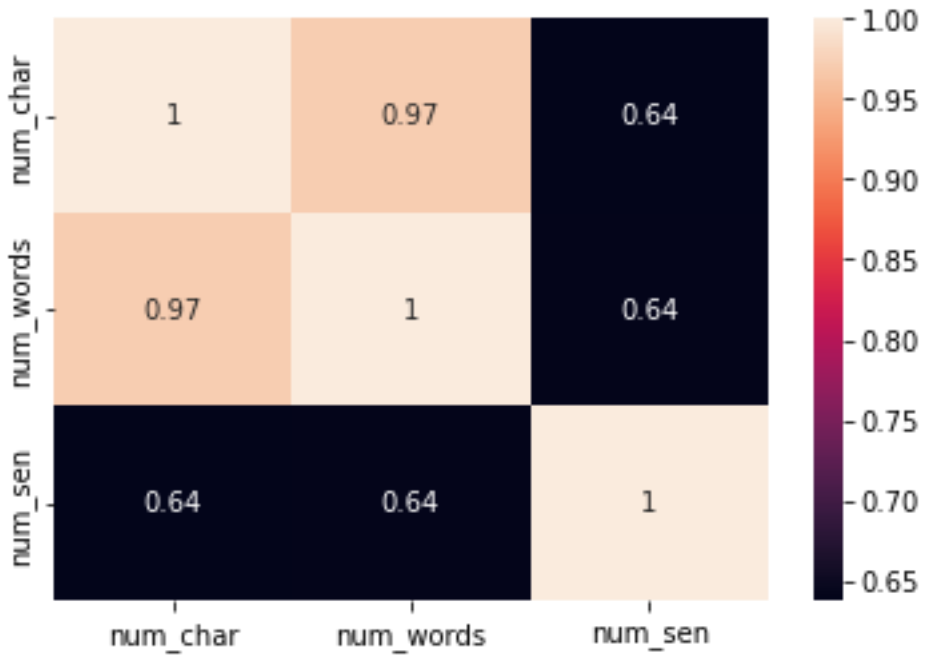
	label	text	num_char	num_words	num_sen
0	ham	Go until jurong point, crazy.. Available only ...	111	20	2
1	ham	Ok lar... Joking wif u oni...	29	6	2
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	155	28	2
3	ham	U dun say so early hor... U c already then say...	49	11	1
4	ham	Nah I don't think he goes to usf, he lives aro...	61	13	1
...	...	...	...	...	...
5561	spam	This is the 2nd time we have tried 2 contact u...	161	30	4
5562	ham	Will Ì_ b going to esplanade fr home?	37	8	1
5563	ham	Pity, * was in mood for that. So...any other s...	57	10	2
5564	ham	The guy did some bitching but I acted like i'd...	125	26	1
5565	ham	Rofl. Its true to its name	26	6	2

5566 rows × 5 columns



Conclusion: Remove few Outliers present in dataset because:

- Affects the correctness of the model
- Stability and performance of the model
- Consistent with model assumptions



4.2. Data Preprocessing

Out[47]:

	label	text	num_char	num_words	num_sen	num_words_transform
0	ham	go jurong point crazi avail bugi n great world...	111	20	2	16
1	ham	ok lar joke wif u oni	29	6	2	6
2	spam	free entri 2 wkli comp win fa cup final tkt 21...	155	28	2	24
3	ham	u dun say earli hor u c alreadi say	49	11	1	9
4	ham	nah think goe usf live around though	61	13	1	7
...	...	...	...	...	...	...
5561	spam	2nd time tri 2 contact u pound prize 2 claim e...	161	30	4	17
5562	ham	b go esplanad fr home	37	8	1	5
5563	ham	piti mood suggest	57	10	2	3
5564	ham	guy bitch act like interest buy someth els nex...	125	26	1	13
5565	ham	rofl true name	26	6	2	3

5566 rows × 6 columns

Conclusion:

Clean or Handle Text Data

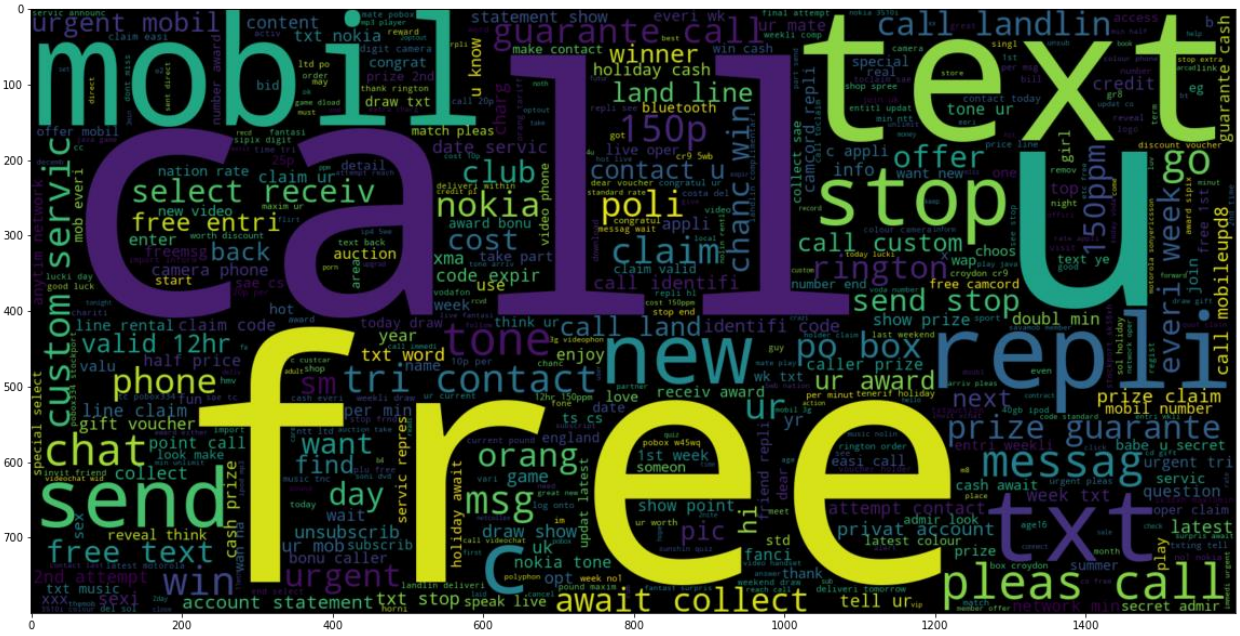
- Remove Web Links
- Remove Numbers
- Remove Emails

Create Common function to Clean or process

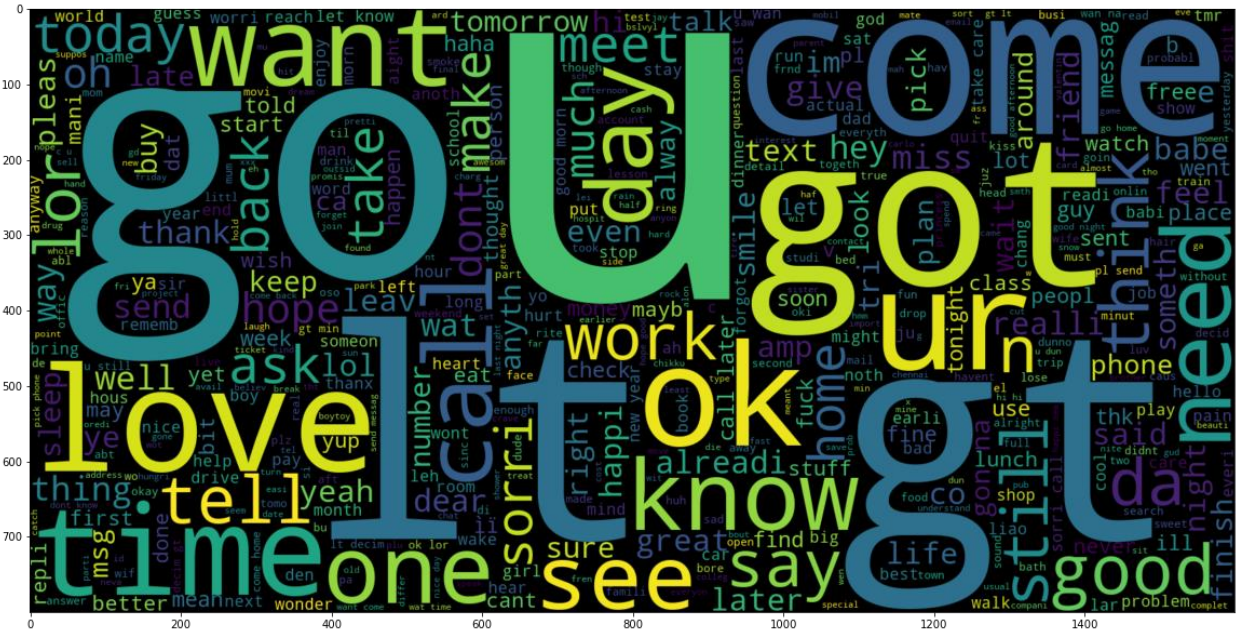
- Text Data Lower casing to avoids duplicates
- Tokenization sentences
- Remove Specials characters
- Remove Stopwords
- Remove punctuation Stemming

4.3. Data Visualization

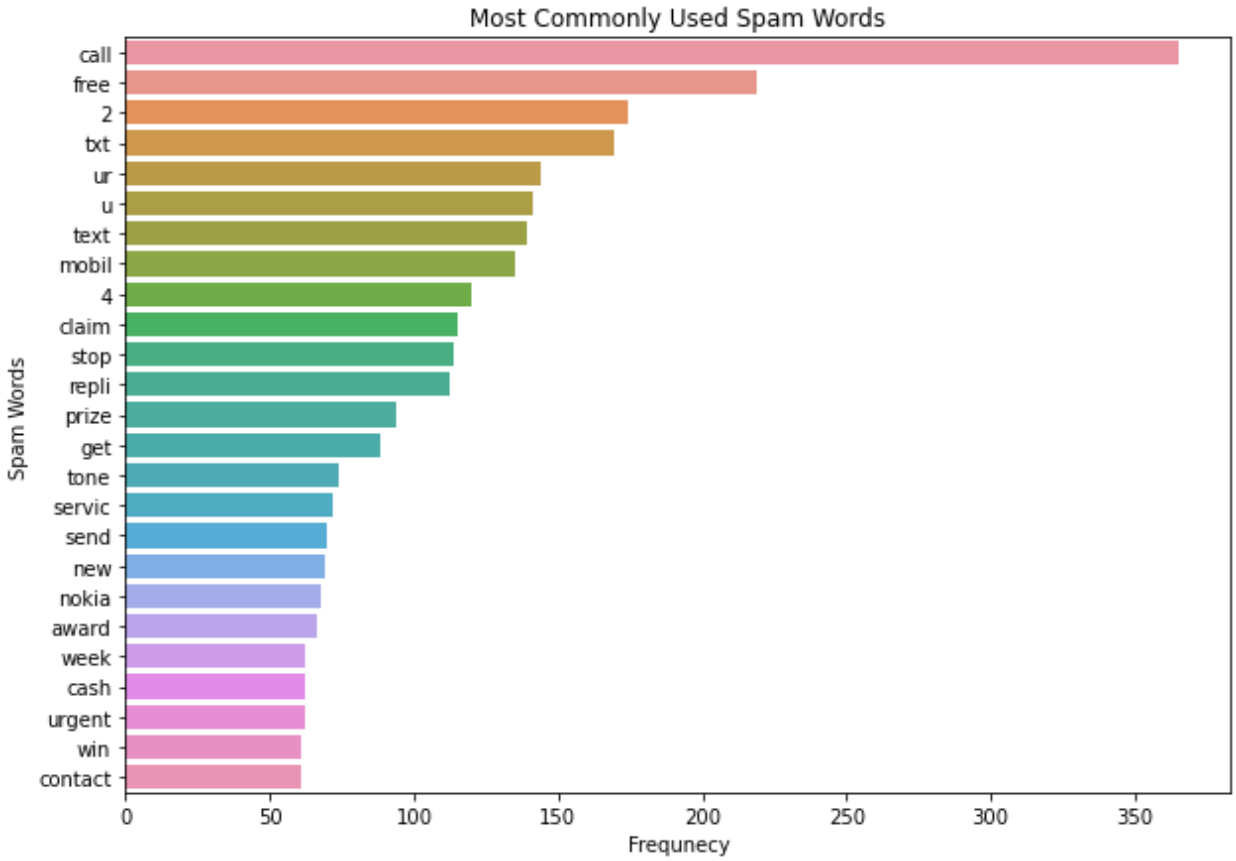
4.3.1. Most common words



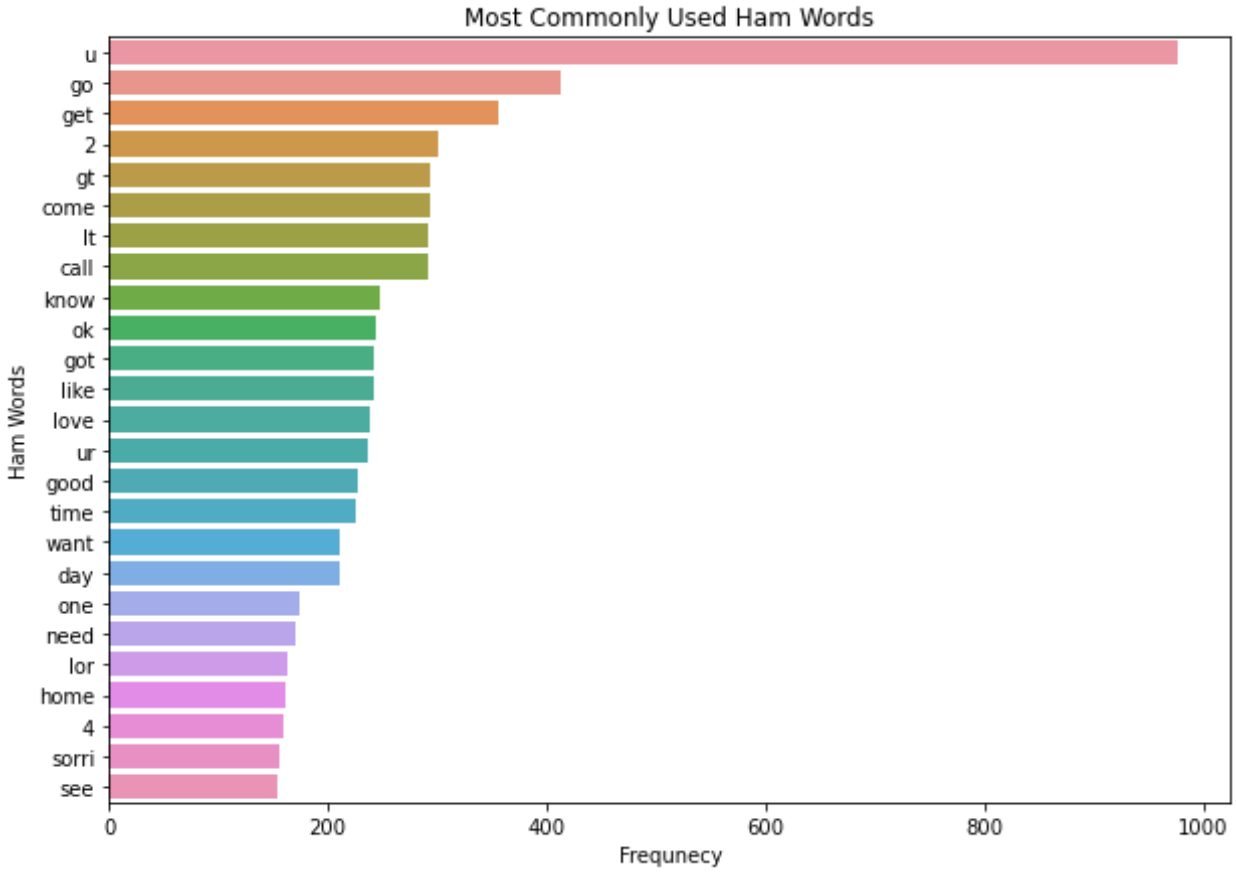
Conclusion: Most common words used for Spam Text words for example : Free, call, text, u , mobil, stop ….



Conclusion: Most common words used for Ham Text words for example: U, go, come, got, g, t …

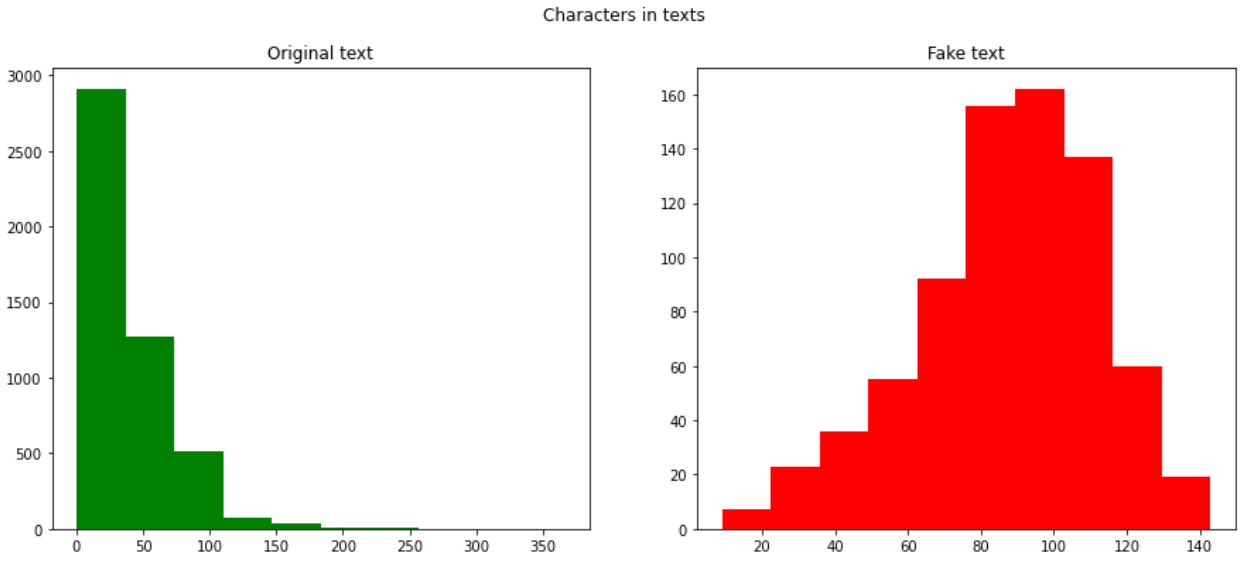


Conclusion: Most common words used for Spam Text words for example: Call, free, 2, txt, ur, u …

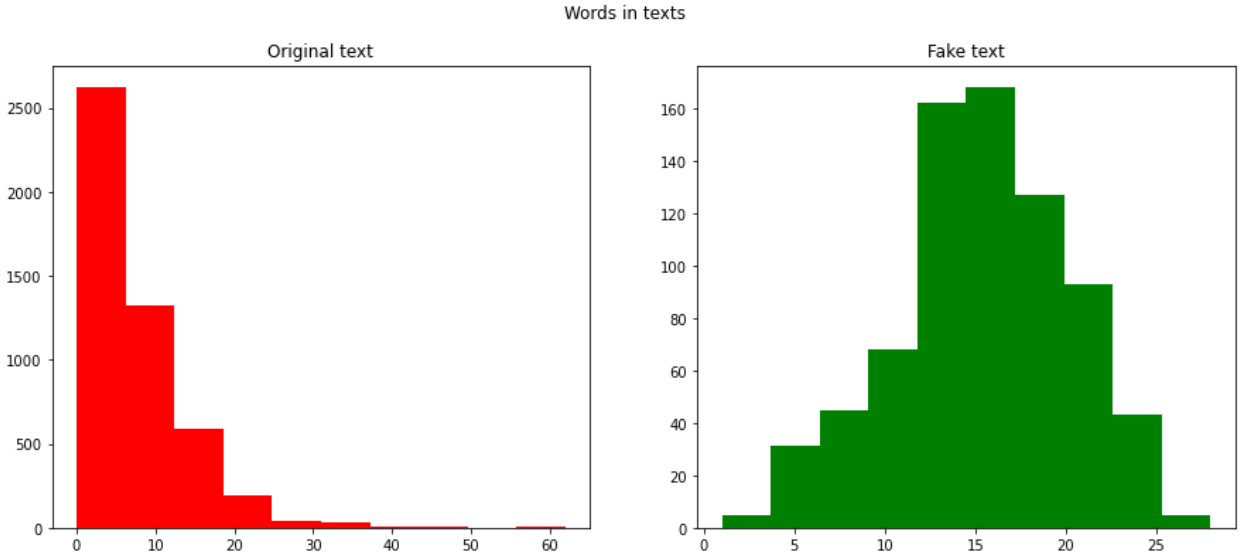


Conclusion: Most Commonly Used Ham Words
words for example: U, go, get, 2, gt, come ...

4.3.2. Compare Total No of charactres & words in spam and ham text

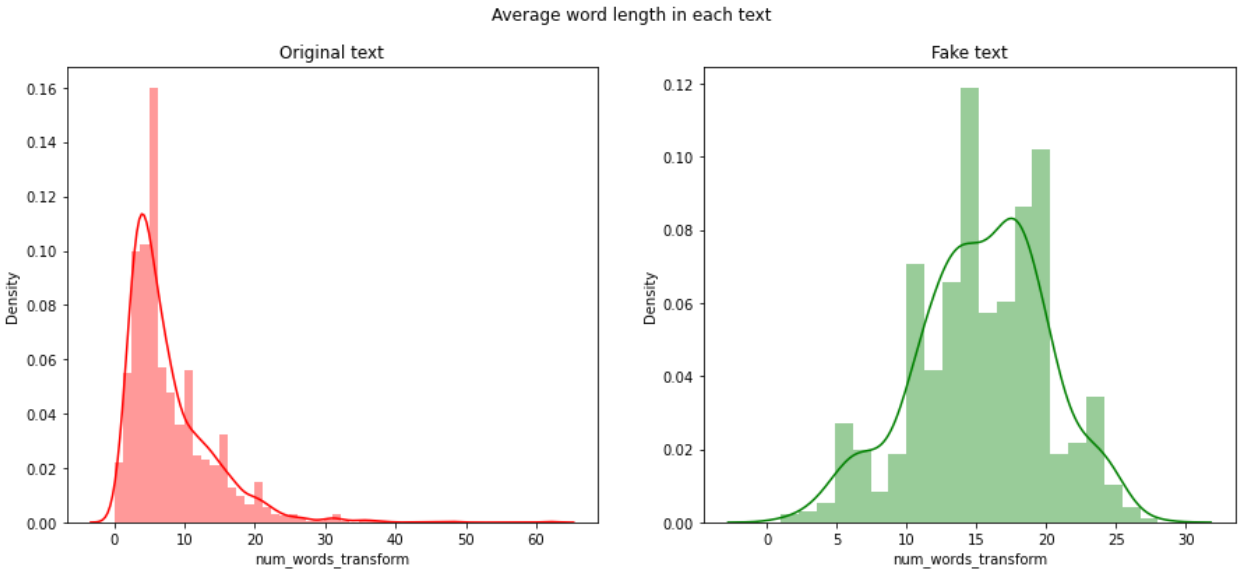


Conclusion: No of Characters in Spam and Ham Text
Left histogram (green): Shows the distribution of text lengths in original texts.
Right histogram (red): Shows the distribution of text lengths in fake texts.



Conclusion: No of Words in Spam and Ham Text

The graph shows that the word count distribution can be a factor in distinguishing original text from fake text. Fake text detection systems can use word count distribution to improve their effectiveness.



Conclusion: Average word length in each text

The graph shows that word length distribution can be a factor in distinguishing original text from fake text. Forged text detection systems can use word length distribution to improve their effectiveness.

Data Visualization Conclusion

- After visualize I can conclude that spam text has more words and characters as compare to Ham text
- Average characters includes in spam SMS is around 90 characters
- Average words includes in spam SMS is around 15 words

5. **Is there any special point or potential issue that the analyst must pay attention to?**

When analyzing a spam email dataset for classification, an analyst should be aware of several key points and potential issues:

Data Preprocessing and Cleaning:

- Text Data: Emails often contain various formats, informal language, misspellings, and special characters. Preprocessing steps like tokenization, stemming, lemmatization, and removal of stop words and non-textual elements are crucial.
- Encoding Issues: As encountered with your dataset, proper handling of text encoding is essential to avoid data corruption.

Feature Extraction:

- Text data requires conversion into a numerical format that machine learning models can understand. Techniques like Bag of Words, TF-IDF (Term Frequency-Inverse Document Frequency), and word embeddings (like Word2Vec or GloVe) are commonly used.
- Feature selection is crucial to remove irrelevant or redundant features that could negatively impact the model's performance.

Class Imbalance:

- Spam datasets often have a class imbalance (more non-spam than spam emails). This imbalance can bias the model towards the majority class. Techniques like resampling (either oversampling the minority class or undersampling the majority class), synthetic data generation (e.g., SMOTE), or using class weights can help mitigate this.

Model Selection and Tuning:

- Different models have different strengths and weaknesses. Choosing the right model and tuning its parameters for optimal performance is a critical step.

Evaluation Metrics:

- Accuracy alone can be misleading, especially with class imbalance. Metrics like Precision, Recall, F1-Score, and ROC-AUC provide a more nuanced understanding of model performance.

Overfitting and Generalization:

- Ensure the model doesn't just memorize the training data but generalizes well to new, unseen data. Techniques like cross-validation, regularization, and dropout (for neural networks) are important.

Ethical and Privacy Considerations:

- Ensure compliance with data privacy laws and regulations. Be cautious with personal and sensitive information in emails.

Model Interpretability:

- Depending on the use case, being able to interpret model decisions can be important. Some models like decision trees and logistic regression offer more interpretability than models like neural networks.

Real-world Application and Deployment:

- Consider how the model will be deployed in a real-world setting. This includes integration with existing systems, handling streaming data, and updating the model with new data over time.

Testing with Real-world Data:

- It's essential to test the model with real-world data that wasn't seen during the training process to ensure that the model performs well in actual use cases.

By paying attention to these aspects, an analyst can significantly improve the chances of developing a robust and effective spam email classification system

6. Bonus: perform the model to solve the problem, discuss the result, make conclusions or recommendations

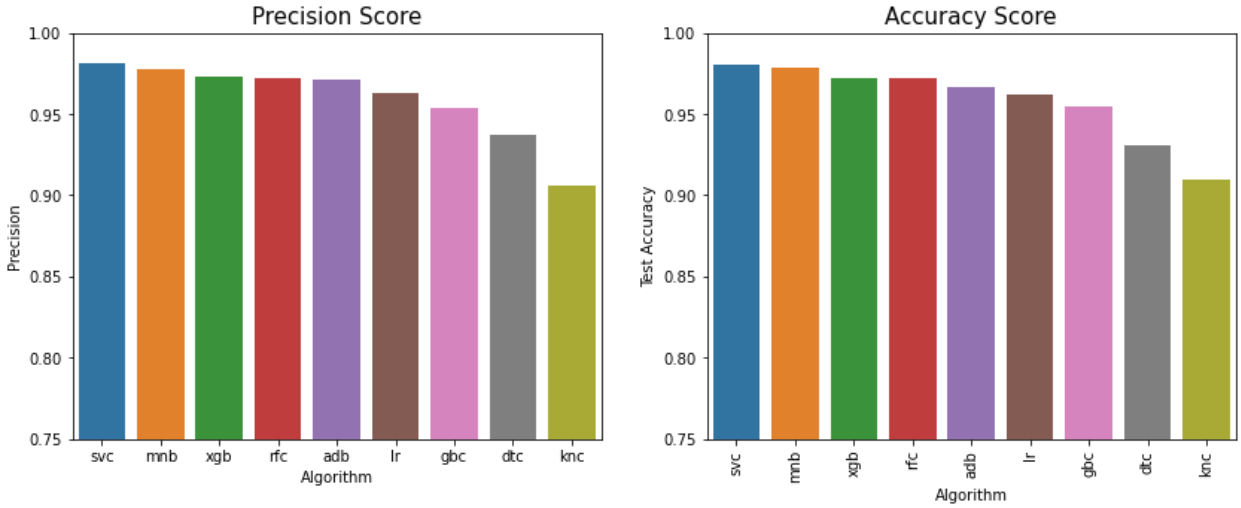
6.1. Model Evaluation

	Algorithm	Precision	Test Accuracy	Train Accuracy
0	svc	0.981	0.980	0.985
2	mnb	0.978	0.979	0.983
7	xgb	0.973	0.972	0.985
5	rfc	0.972	0.972	1.000
6	adb	0.971	0.967	0.975
4	lr	0.963	0.962	0.965
8	gbc	0.954	0.955	0.966
3	dtc	0.937	0.931	0.947
1	knc	0.906	0.910	0.931

Conclusion:

- Algorithm: This column names the different machine learning algorithms being compared. The algorithms listed in the table are svc, mnb, xgb, rfc, adb, lr, gbc, dtc, and knc.
- Precision: This column shows the precision of each algorithm. Precision is a measure of how often the algorithm correctly identifies a positive case. In this case, it likely refers to the percentage of times the algorithm correctly classified an example as belonging to the target class.

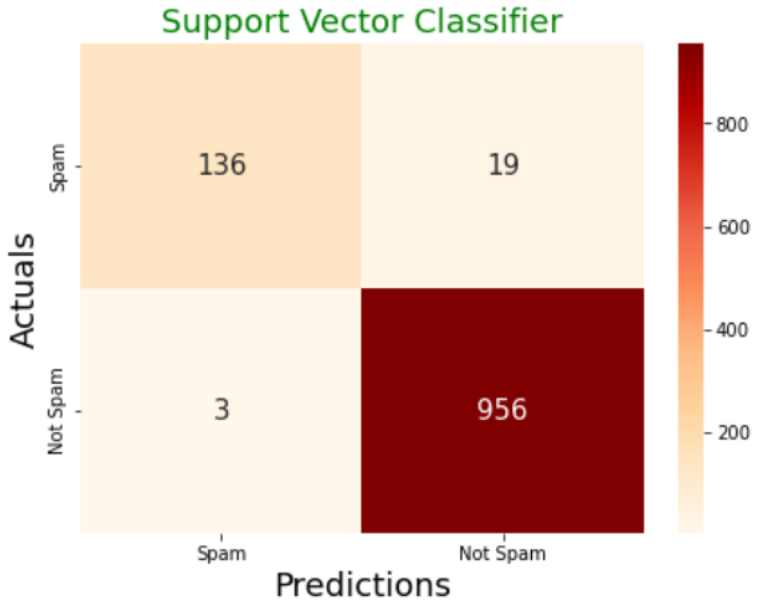
- Test Accuracy: This column shows the test accuracy of each algorithm. Test accuracy is a measure of how well the algorithm performs on unseen data. In this case, it likely refers to the percentage of examples in the test dataset that the algorithm correctly classified.
- Train Accuracy: This column shows the train accuracy of each algorithm. Train accuracy is a measure of how well the algorithm performs on the data it was trained on. In this case, it likely refers to the percentage of examples in the training dataset that the algorithm correctly classified.
- Overall, the algorithms have high accuracy. All of the algorithms have a test accuracy of at least 90%, and most of them have a test accuracy of over 95%.
- SVC appears to be the best performing algorithm. It has the highest test accuracy (98%) and train accuracy (98.5%).
- KNC has the lowest test accuracy (91%). However, it also has the lowest train accuracy (93.1%), which suggests that it may be overfitting the training data.



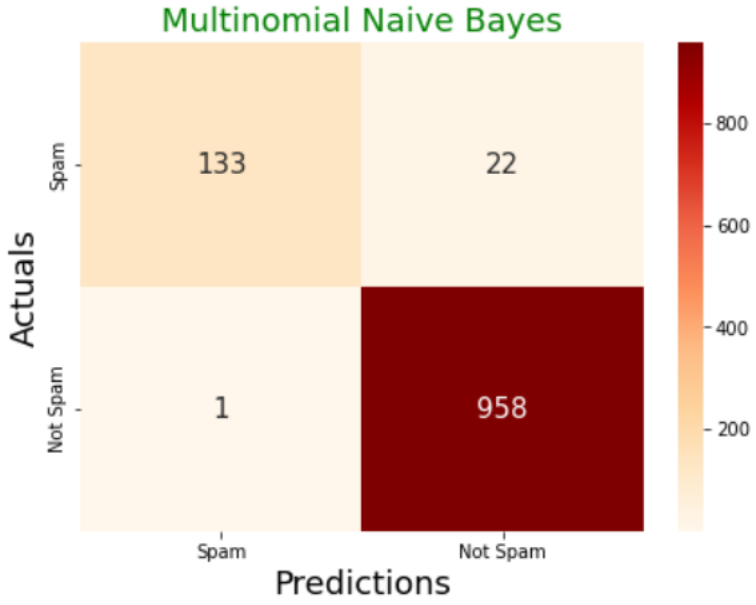
Conclusion: Visualize Accuracy of differents models

6.2. Cross Validation of Top Models

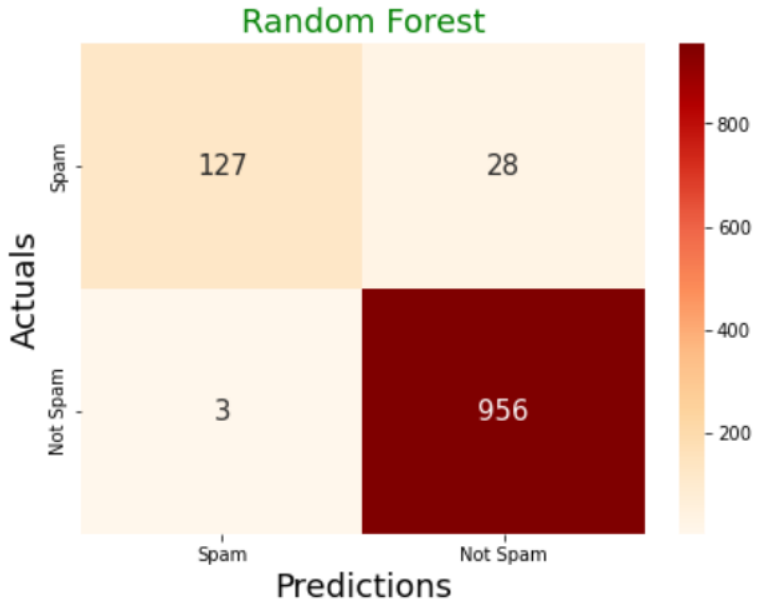
```
Classifier :  Support Vector Classifier
Processing Time :  42.048
Mean Accuracy :  0.978
Precision Score :  0.981
```




```
Classifier :  Multinomial Naive Bayes
Processing Time :  0.527
Mean Accuracy :  0.976
Precision Score :  0.978
```



```
Classifier :  Random Forest
Processing Time :  42.147
Mean Accuracy :  0.978
Precision Score :  0.972
```



Conclusion: Cross Validation of Top Models

- After Train with Multiple algorithms, I can conclude that Support Vector Classifer(SVC) and Multinomial Naive Bayes(mnb) are best algorithms to choose to sovle given NLP problem with very good Accuracy around 0.97
- If I want only higher precision accuracy there is No matter of Processing Time then prefer Support Vector Classifer(SVC) is Best Option.
- If Both Accuracy and processing time is Important Factors then Multinomial Naive Bayes(mnb) Perform Well With very less proceesing time and good accuracy
- Here In Our case I will consider Multinomial Naive Bayes(mnb) and Deploy

REFERENCE

Park, J., Choi, Y., Byun, J., Lee, J., & Park, S. J. I. S. (2023). Efficient differentially private kernel support vector classifier for multi-class classification. *619*, 889-907.

Odera, D., Odiaga, G. J. W. J. o. A. E. T., & Sciences. (2023). A comparative analysis of recurrent neural network and support vector machine for binary classification of spam short message service. *9*(1), 127-152.

Karyawati, A. E., Wijaya, K. D. Y., & Supriana, I. W. J. J. (2023). A Comparison of Different Kernel Functions of SVM Classification Method for Spam Detection. *8*(2), 91-97.

Kurani, A., Doshi, P., Vakharia, A., & Shah, M. J. A. o. D. S. (2023). A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *10*(1), 183-208.

Nagaraj, P., Muneeswaran, V., Reddy, G. S. S., Kumar, V. B., Mohan, B. M., & Kumar, S. (2023). *Automatic Email Spam Classification Using Naïve Bayes.* Paper presented at the 2023 International Conference on Computer Communication and Informatics (ICCCI).