A Project Report
on
"SMS Spam Prediction using Machine Learning Techniques"

submitted for the partial fulfilment of the award of the degree of

BACHELOR OF TECHNOLOGY
in
Artificial Intelligence and machine Learning
(4th Semester)

Submitted to:-
Dr. Amit Choudhary
Asst. Prof., USAR

Submitted by:-
Name: Ekansh Juneja
Batch: AIML B1-b
Roll No: 05619011621

University School of Automation and Robotics
GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY
(EAST DELHI CAMPUS)
SURAJMAL VIHAR, NEW DELHI-110032

SMS SPAM PREDICTION USING MACHINE LEARNING MODELS

Abstract:

The problem of spam SMS, also known as unsolicited or unwanted text messages, is a pervasive issue that affects mobile phone users worldwide. Spam SMS refers to the unauthorized and often deceptive messages sent in bulk to a large number of recipients without their consent. These messages are typically commercial in nature, promoting various products, services, scams, or fraudulent activities.

This report aims to classify SMS messages as either spam or not. The classification task involves analysing a dataset of SMS messages and training a model to accurately predict whether a message is spam or legitimate. The report discusses the dataset used, different classification techniques employed, and concludes with an evaluation of the results.

The dataset is useful for researchers who want to conduct comparative studies on Spam and Legitimate SMS and also for training in the machine learning area.

Keywords:

Spam; Ham; Machine learning; Exploratory Data Analysis; Prediction.

1. Introduction:

Spam detection in SMS messages is a critical problem in the realm of mobile communication. With the increasing use of mobile phones and the convenience of text messaging, spam senders have found ways to exploit this medium for their own purposes. The problem involves identifying and distinguishing between legitimate messages and unsolicited spam messages.

The objective of spam detection in SMS messages is to develop robust algorithms and techniques that can automatically classify incoming messages as either spam or legitimate. This task requires analyzing the content and characteristics of each message to determine its nature and intent accurately.

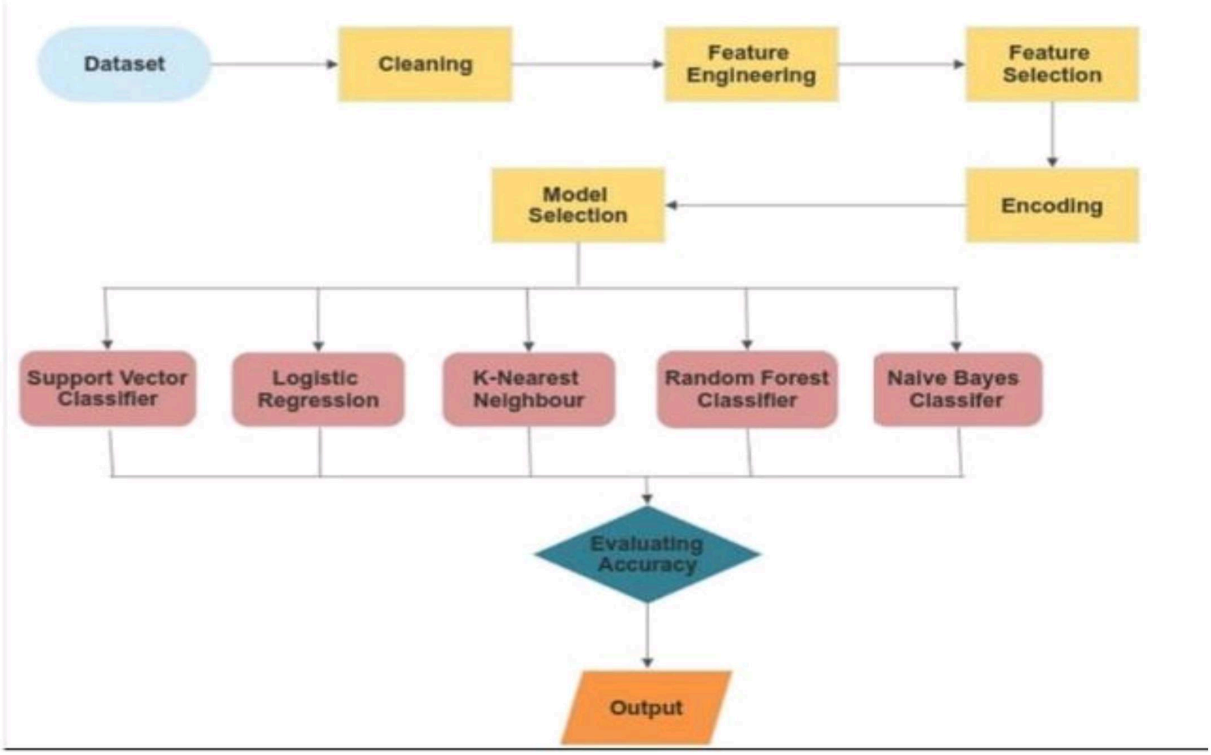
Spam messages often employ various tactics to deceive recipients, such as using misleading subject lines, offering fake products or services, or attempting to obtain sensitive information. These messages can be disruptive, annoying, and potentially harmful to users, both in terms of privacy invasion and security risks.

SMS Spam detection systems need to be adaptable to evolving spam techniques and capable of handling a large volume of messages efficiently. Moreover, they should minimize false positives, ensuring that legitimate messages are not mistakenly classified as spam.

Spam detection in SMS messages plays a crucial role in enhancing user experience, protecting privacy, and maintaining the integrity of mobile communication channels. By effectively detecting and filtering out spam, users can have a more secure and hassle-free messaging experience, ensuring that they receive only the messages that are relevant and desired.

The dataset contained 5573 records with 5 attributes, where each record represents a SMS and can be used for benchmarking the performance of different algorithms for solving the same type of problem and for training in the machine learning area.

2. ***Proposed Methodology:***



2.1 ***Dataset:***

The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged acording being ham (legitimate) or spam.

The files contain one message per line. Each line is composed by two columns: v1 contains the label (ham or spam) and v2 contains the raw text.

This corpus has been collected from free or free for research sources at the Internet:

A collection of 425 SMS spam messages was manually extracted from the Grumbletext Web site.

A subset of 3,375 SMS randomly chosen ham messages of the NUS SMS Corpus (NSC), which is a dataset of about 10,000 legitimate messages collected for research at the Department of Computer Science at the National University of Singapore.

A list of 450 SMS ham messages collected from Caroline Tag's PhD Thesis.

The final dataset is available as a comma separated values (CSV) file encoded as UTF8 and consists of 5573 records with 5 attributes and contains 3 unnamed empty columns that we remove later.

Attributes information in dataset:

```
Int64Index: 5169 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   target  5169 non-null   int64 
 1   text    5169 non-null   object
dtypes: int64(1), object(1)
```

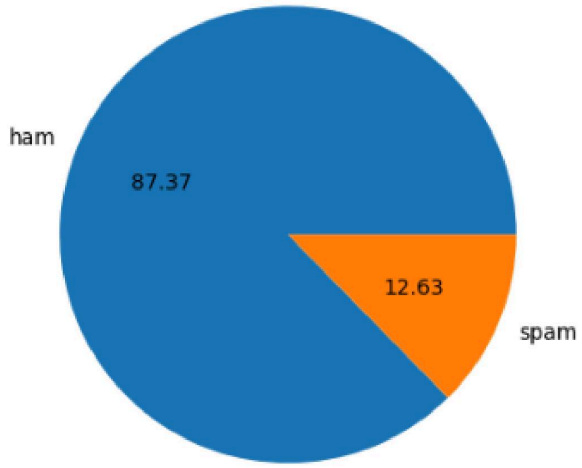
2.2 Exploratory Data Analysis:

A brief exploratory data analysis is performed in Python 3 using the Pandas library version 2.0.2, the Scikit-learn library version 1.2.2, nltk library, and the Seaborn library version 0.12.2 for visualizations.

Showing sample of data with head() function:

	target	text
2528	ham	jay says he'll put in <#>
212	ham	K:)k:)good:)study well.
1093	ham	Well the weather in cali's great. But its comp...
5260	ham	If anyone calls for a treadmill say you'll buy...
1371	ham	I though we shd go out n have some fun so bar ...

Pie Chart- ham and spam categories of text:



Added 3 new columns for classification namely- num_characters, num_words, num_sentences:

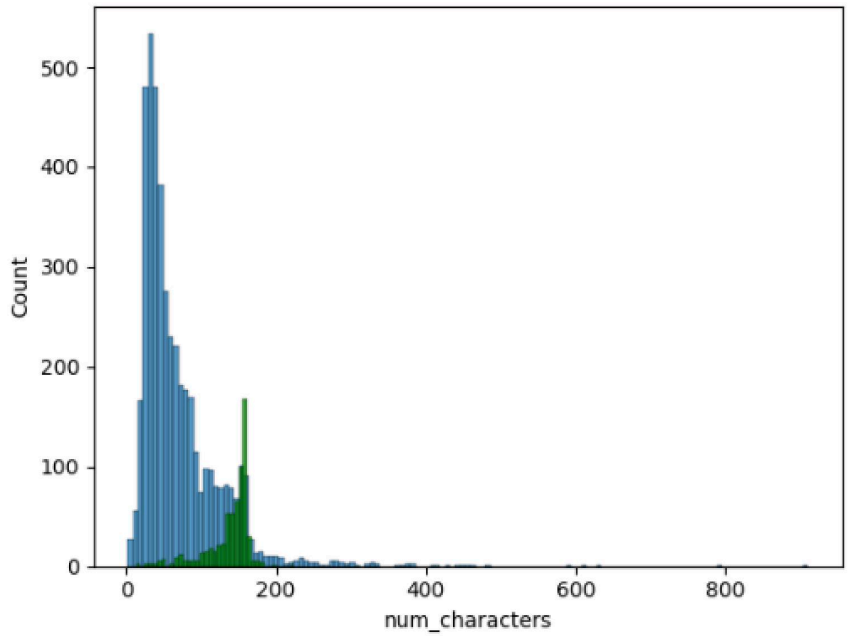
	target	text	num_characters	num_words	num_sentences
0	0	Go until jurong point, crazy.. Available only ...	111	24	2
1	0	Ok lar... Joking wif u oni...	29	8	2
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2
3	0	U dun say so early hor... U c already then say...	49	13	1
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1

Table showing describe of dataset including, the count of main attribute value, the mean of main attribute values, the standard deviation of the attributes values, and the minimum and maximum value for numerical attributes only,etc.

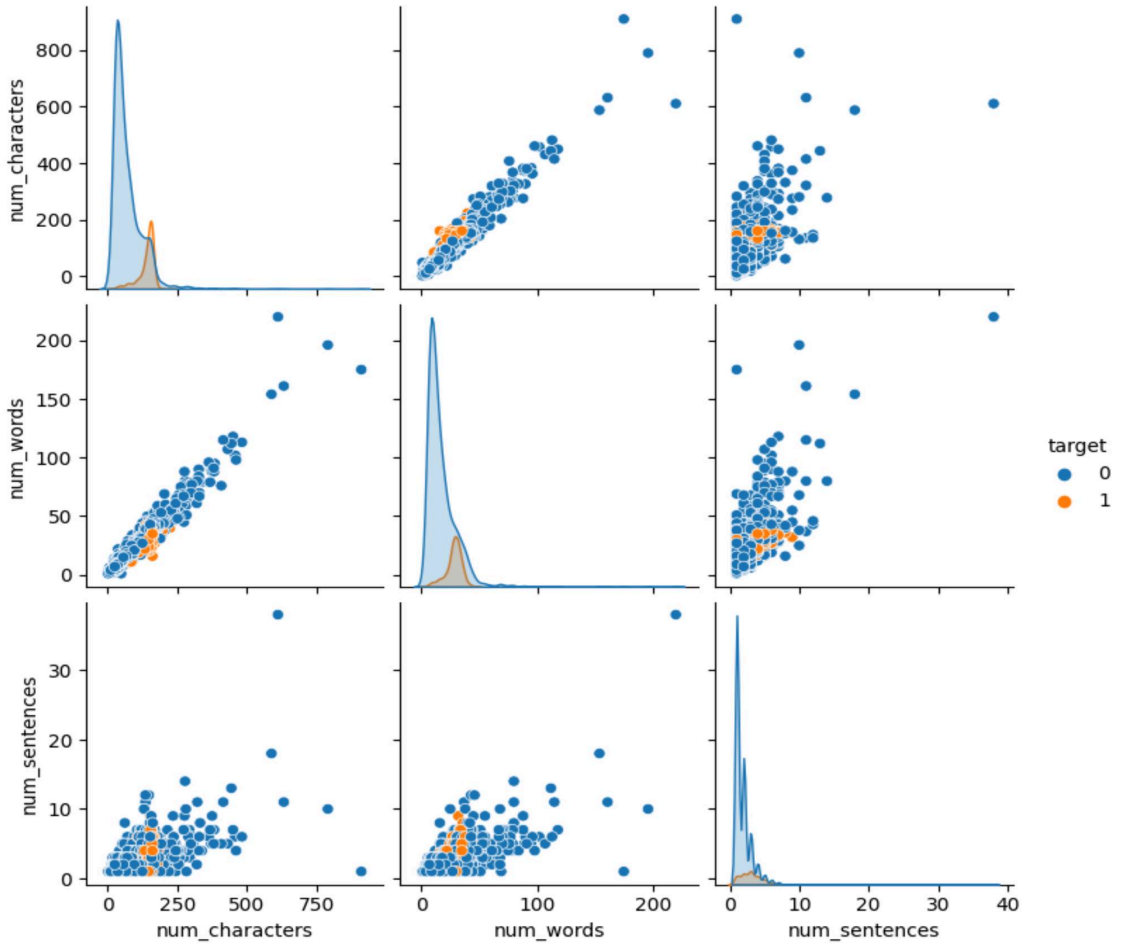
	target	num_characters	num_words	num_sentences
count	5169.000000	5169.000000	5169.000000	5169.000000
mean	0.126330	78.977945	18.455794	1.965564
std	0.332253	58.236293	13.324758	1.448541
min	0.000000	2.000000	1.000000	1.000000
25%	0.000000	36.000000	9.000000	1.000000
50%	0.000000	60.000000	15.000000	1.000000
75%	0.000000	117.000000	26.000000	2.000000
max	1.000000	910.000000	220.000000	38.000000

Histogram plot:

```
<Axes: xlabel='num_characters', ylabel='Count'>
```

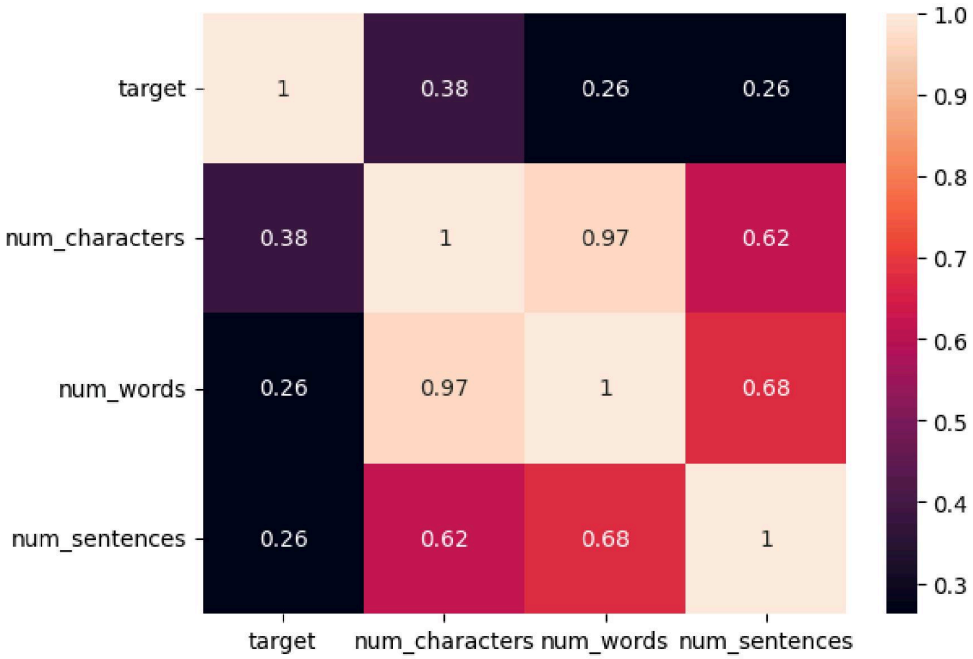


Pairplot:

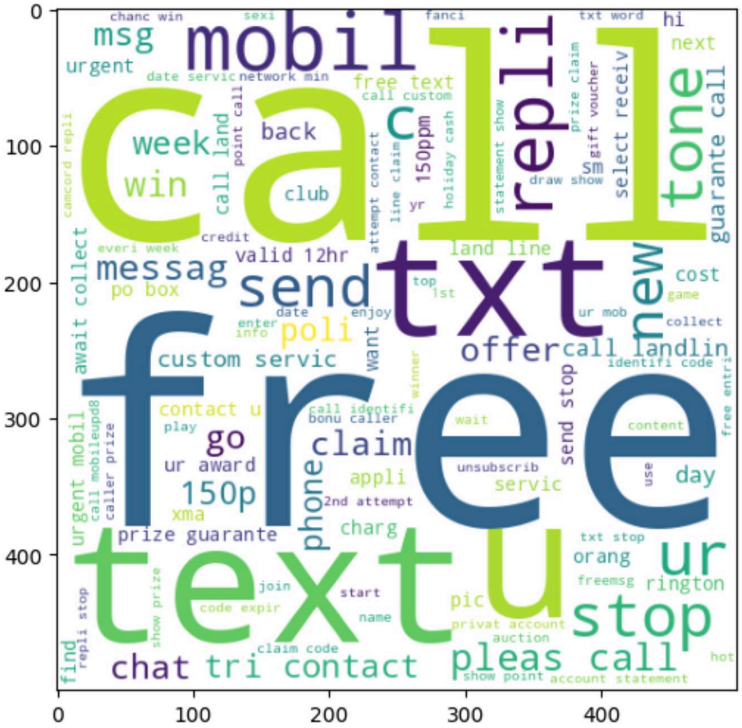


Heatmap:

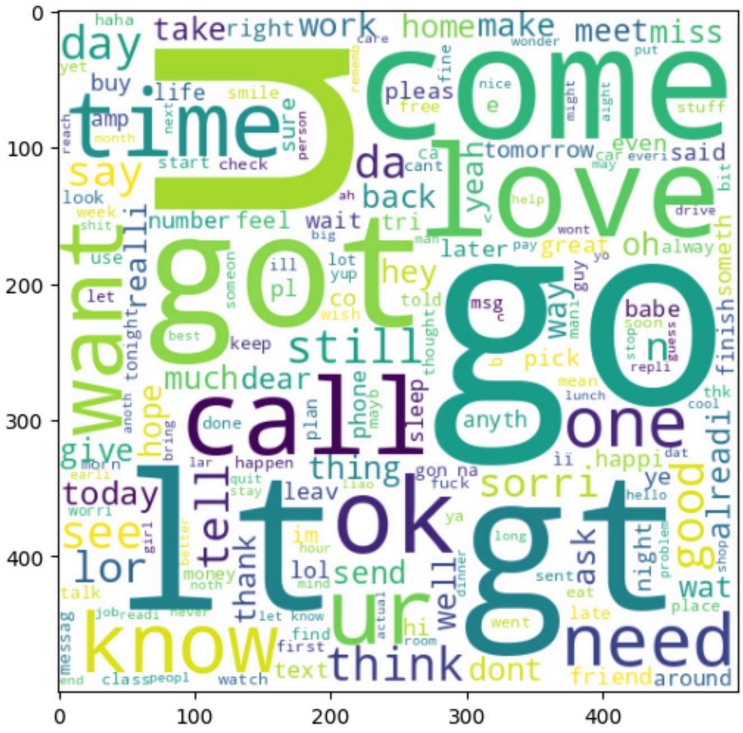
The analysis of the heatmap, using the Pearson correlation coefficient, shows that there are some pairs of features having high correlation coefficients, which increases multi-collinearity in the dataset.
The collinearity is strongest within the same group of features, but we can also find higher values of correlation between groups.



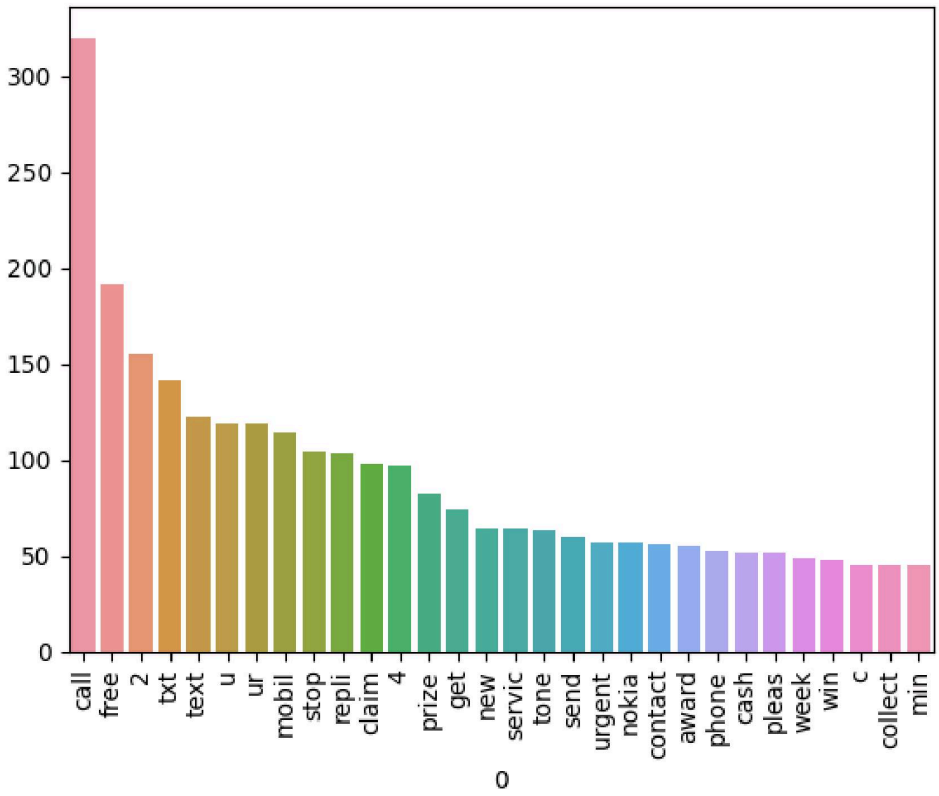
Spam Word Cloud:



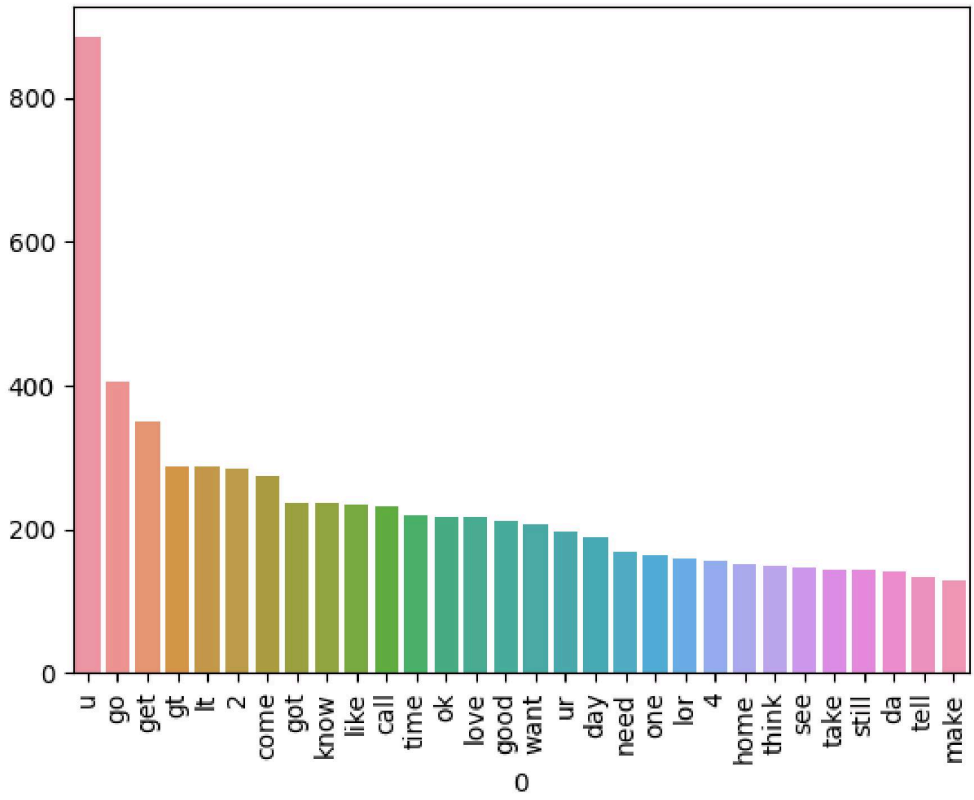
Ham Word Cloud:



30 most common words in spam SMS:



30 most common words in ham SMS:



2.3 Data Preprocessing:

Before training the machine learning models, several preprocessing steps were applied to the dataset:

2.3.1 Handling Missing and Duplicate Values:

Missing and duplicate values in the dataset were identified and treated appropriately. Different strategies such as mean imputation, or removal of instances and attributes were employed based on the nature and significance of the missing and duplicate values.

2.3.2 Feature Selection:

To enhance model performance and reduce computational complexity, feature selection techniques like correlation analysis, information gain, or stepwise regression were used to select the most relevant attributes for training the models. In this particular dataset all the attributes either have significant corelation with the target or among each other as in section 2.2 data analysis so, all the features were taken for the model training.

The target column contain two classes ham and spam. As per our assertion we are predicting whether the SMS is spam or ham.

2.3.3 Natural Language Processing:

Text data was tokenised, special characters were removed, stop wards and punctuation were removed, Data stemming was done.

All of this is done using nltk library of python to enable machine learning models to understand classify the SMS into ham and spam.

	target	text	num_characters	num_words	num_sentences	transformed_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2	free entri 2 wkli comp win fa cup final tkt 21...
3	0	U dun say so early hor... U c already then say...	49	13	1	u dun say earli hor u c alreadi say
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1	nah think goe usf live around though

2.3.4 Encoding:

Attributes having categorical data are encoded into numerical representations to enable machine learning algorithms to process them effectively. Technique like label encoding is employed.

The Target column which contain Text values are encoded using the label encoder in the scikit learn library which randomly assign numerical values to all the unique values.

Data vectorization is done on textual data CountVectorizer, TfidVectorizer functions of sklearn.feature_extraction.text library.

2.3.5 Data Splitting:

The pre-processed dataset is divided into training and testing sets, with 80% of the data allocated for training the models and 20% for evaluating their performance. Splitting is done using the train_test_split function of the sklearn library.

2.4 Model Training and Evaluation:

5 classification ML models, such as Support vector machines (SVM), naive-bayes classifier, logistic regression, KNN Classifier and random forests are trained using the training dataset.

Each model is trained with the goal of accurately predicting the provided text SMS as ham or spam.

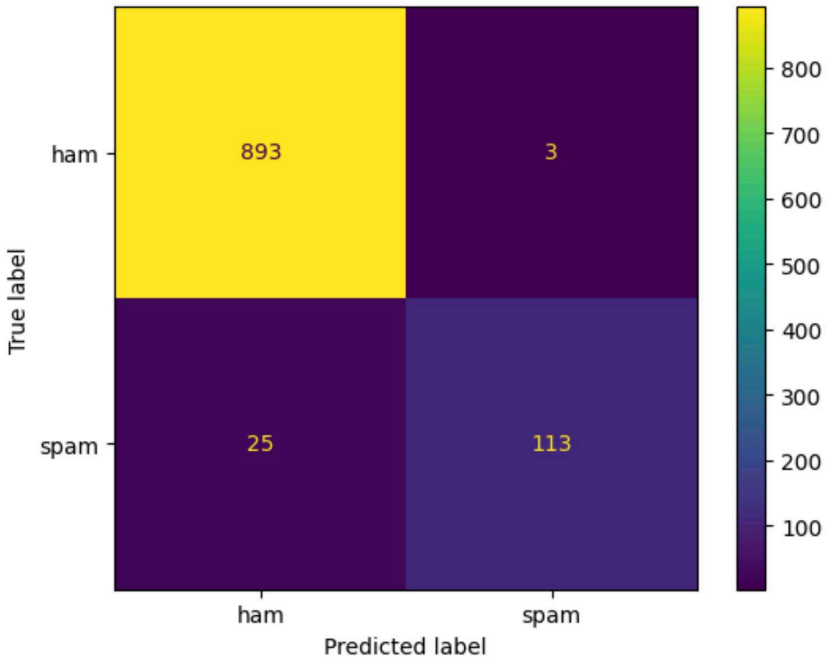
The trained models are evaluated using appropriate evaluation metrics, such as accuracy, precision, recall, and F1-score, to assess their performance and compare their effectiveness in predicting the result.

2.4.1 Support Vector Machine:

Support Vector Classifier (SVC), also known as Support Vector Machine (SVM), is a popular machine learning algorithm used for both binary and multi-class classification tasks. It is particularly effective when dealing with complex decision boundaries and datasets with high-dimensional feature spaces. Here we apply the SVC with the following hyper parameters gamma=scale, kernel='sigmoid'.

The Evaluation of model is as below:

```
For Support Vector Machine

**************************************************
               Classification Report
**************************************************
              precision    recall  f1-score   support

           0       0.97      1.00      0.98       896
           1       0.97      0.82      0.89       138

    accuracy                           0.97      1034
   macro avg       0.97      0.91      0.94      1034
weighted avg       0.97      0.97      0.97      1034

**************************************************

Accuracy -  0.9729206963249516
Precision -  0.9741379310344828
Recall -  0.9729206963249516
F1_score -  0.9729206963249516
```

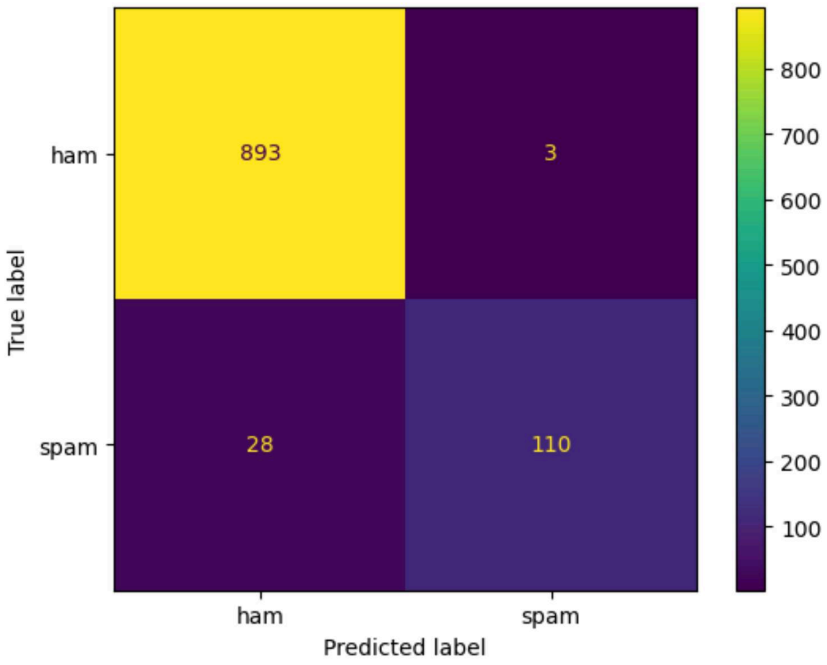


2.4.2 Naive-Bayes Classifier:

Naive Bayes is a popular machine learning algorithm used for classification tasks. It is based on Bayes theorem and assumes that features are conditionally independent given the class labels. Naive Bayes is particularly useful when dealing with large feature spaces and relatively small training datasets. Naive Bayes is used using the Sklearn library. Here we compared 3 types of Naïve Bayes and decided to use Bernoulli Naïve Bayes as it gives us best results out of those 3.

The Evaluation of model is as below:

```
For Bernoulli Naive Bayes

*****************************************************
                Classification Report
*****************************************************
              precision    recall  f1-score   support

           0       0.97      1.00      0.98       896
           1       0.97      0.80      0.88       138

    accuracy                           0.97      1034
   macro avg       0.97      0.90      0.93      1034
weighted avg       0.97      0.97      0.97      1034

*****************************************************

Accuracy -  0.9700193423597679
Precision -  0.9734513274336283
Recall -  0.9700193423597679
F1_score -  0.9700193423597679
```

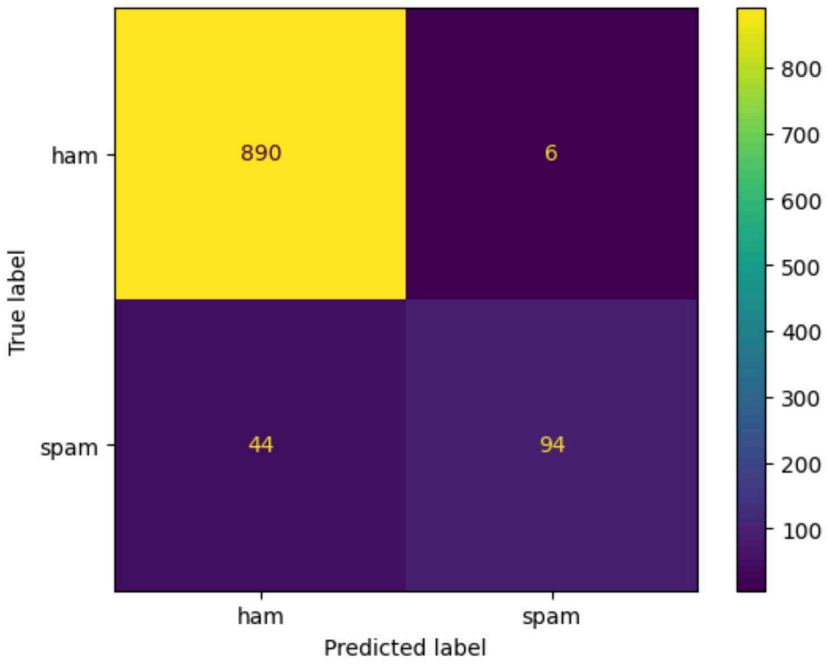


2.4.3 Logistic Regression:

Logistic regression is a widely used machine learning algorithm for binary classification problems. It is a type of regression analysis that models the relationship between a set of independent variables (features) and a binary dependent variable (the target variable) using the logistic function. Since it is a binary classification problem logistic regression would be one of the best for that.
Used: solver='liblinear',penalty='l1'.

The Evaluation of model is as below:

```
For Logistic Regression

*****************************************************
               Classification Report
*****************************************************
              precision    recall  f1-score   support

           0       0.95      0.99      0.97       896
           1       0.94      0.68      0.79       138

    accuracy                           0.95      1034
   macro avg       0.95      0.84      0.88      1034
weighted avg       0.95      0.95      0.95      1034

*****************************************************

Accuracy -  0.9516441005802708
Precision -  0.94
Recall -  0.9516441005802708
F1_score -  0.9516441005802708
```

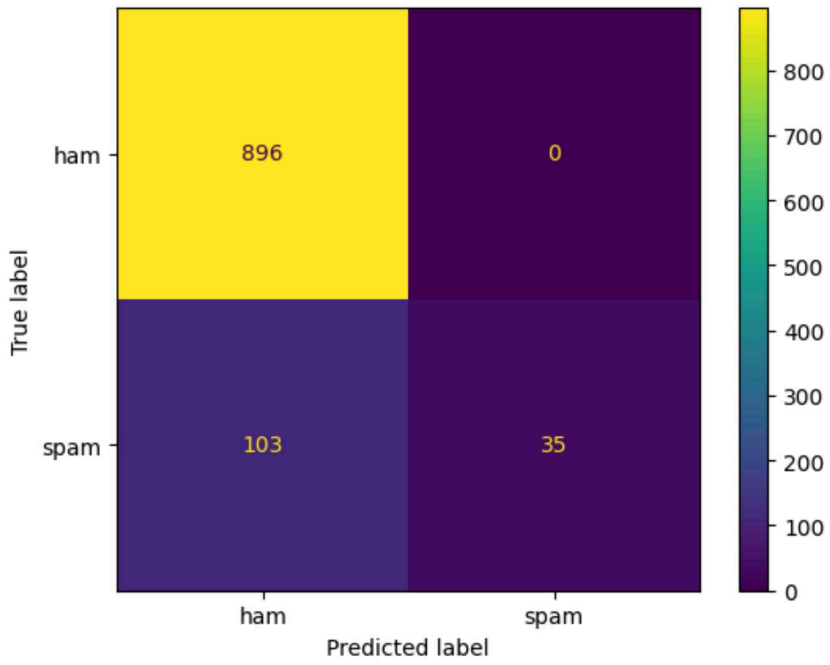


2.4.4 K-Nearest Neighbour:

The K-Nearest Neighbors (KNN) classifier is a versatile machine learning algorithm used for both classification and regression tasks. It is a non-parametric algorithm. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

The Evaluation of model is as below:

```
For K-Nearest Neighbour

*****************************************************
                Classification Report
*****************************************************
              precision    recall  f1-score   support

           0       0.90      1.00      0.95       896
           1       1.00      0.25      0.40       138

    accuracy                           0.90      1034
   macro avg       0.95      0.63      0.68      1034
weighted avg       0.91      0.90      0.87      1034

*****************************************************

Accuracy -  0.9003868471953579
Precision -  1.0
Recall -  0.9003868471953579
F1_score -  0.9003868471953579
```

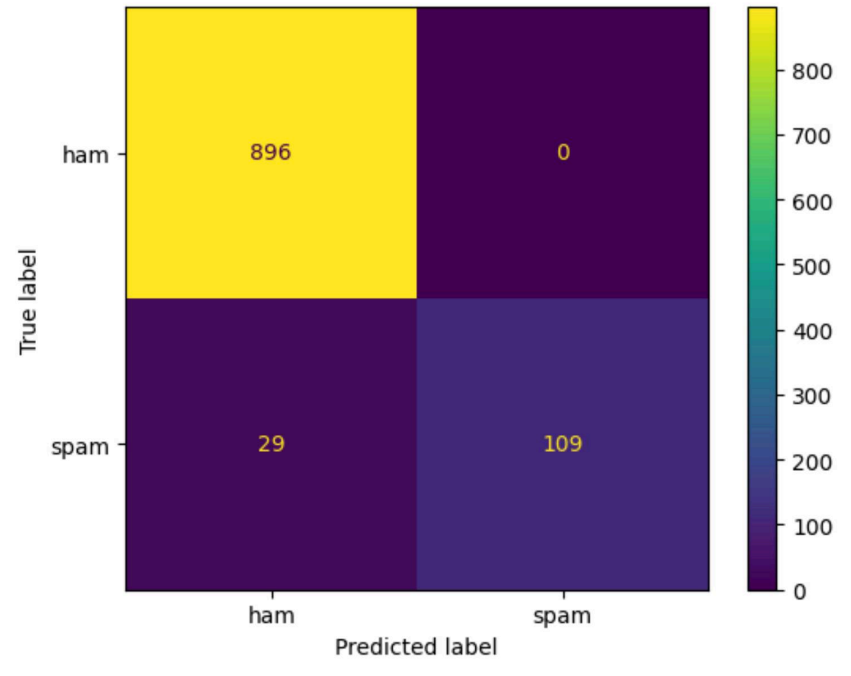


2.4.5 Random Forest:

Random Forest is a popular machine learning algorithm that is commonly used for both classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to make predictions. In this dataset we apply random Forest with following hyperparameters Random Forest n_estimators=50, random state=2.

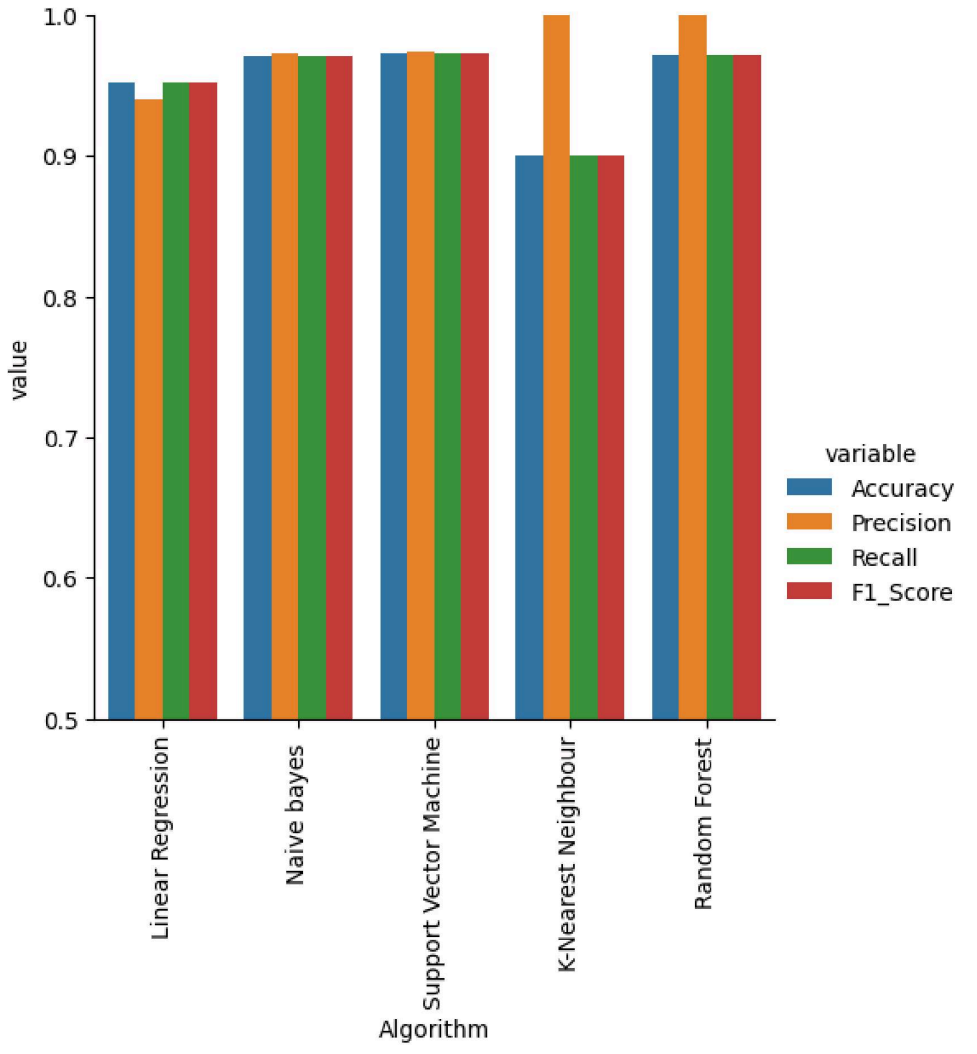
The Evaluation of model is as below:

```
For Random Forest

*********************************************************
                Classification Report
*********************************************************
              precision    recall  f1-score   support

           0       0.97      1.00      0.98       896
           1       1.00      0.79      0.88       138

    accuracy                           0.97      1034
   macro avg       0.98      0.89      0.93      1034
weighted avg       0.97      0.97      0.97      1034

*********************************************************

Accuracy -  0.971953578336557
Precision -  1.0
Recall -  0.971953578336557
F1_score -  0.971953578336557
```



2.4.6 ***Comparison between models:***

	Algorithm	Accuracy	Precision	Recall	F1_Score
2	Linear Regression	0.951644	0.940000	0.951644	0.951644
1	Naive bayes	0.970019	0.973451	0.970019	0.970019
0	Support Vector Machine	0.972921	0.974138	0.972921	0.972921
3	K-Nearest Neighbour	0.900387	1.000000	0.900387	0.900387
4	Random Forest	0.971954	1.000000	0.971954	0.971954



As per the above comparison table and graph we get the information that among all the selected classifiers **Random Forest (Accuracy: 0.971954 and Precision: 1.000000**) and **Support Vector Machine(Accuracy: 0.972921 and Precision: 0.974138)** performs the best.

3. ***Result and Discussion:***

During the evaluation phase, the performance of the trained machine learning models for predicting SMS Spam or Ham is assessed. The evaluation metrics used include accuracy, precision, recall, and F1 score, which provide insights into the models' effectiveness in classification tasks. The results obtained from the evaluation reveal the performance of each model on the given dataset.

The models are compared based on their metrics, allowing for the identification of the most suitable model for the task at hand. For example, the KNN model may demonstrate higher precision but lower accuracy.

Furthermore, the implications of the models' performance in real-world scenarios are discussed. The potential benefits of accurately predicting SMS categories are highlighted and emphasized. The limitations of the models pave path for improvement and future work in the field.

4. ***Conclusion and Future Work:***

In conclusion, this project successfully develops and evaluates machine learning models for predicting that **the SMS is ham or spam** using the provided dataset. The conclusions drawn from the results highlight the potential applications of machine learning in this domain.

In terms of future work, several avenues can be explored **to enhance the accuracy and robustness of the predictive models**.

Firstly, **during vectorization max_features could be limited to certain number in the dataset** for more accurate and precise result.

Secondly, experimenting with different machine learning algorithms or ensemble methods can be undertaken to identify the most optimal approach for prediction.
Techniques like **gradient boosting, deep learning, or recurrent neural networks** can be used for better results.

Lastly, integrating real-time data and implementing a feedback loop system would enable **continuous model refinement and adaptation to evolving dynamics**. This would provide more up-to-date predictions and allow for timely interventions.

By addressing these areas of improvement, the accuracy and applicability of the predictive models can be further enhanced, leading to **more effective prediction**.

5. ***References:***

1. https://en.wikipedia.org/wiki/Classification
2. https://www.javatpoint.com/classification-algorithm-in-machine-learning
3. https://scholar.google.co.in/scholar?q=ml+classification+learning&hl=en&as_sdt=0&as_vis=1&oi=scholart
4. https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
5. https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.
6. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
7. https://en.wikipedia.org/wiki/Logistic_regression
8. https://www.javatpoint.com/machine-learning-naive-bayes-classifier
9. https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/1822-b.e-cse-batchno-109.pdf
10. https://www.ijstr.org/final-print/feb2020/Spam-Detection-In-Sms-Using-Machine-Learning-Through-Text-Mining.pdf
11. https://www.tutorialspoint.com/scikit_learn/index.htm
12. https://matplotlib.org/
13. https://www.hotjar.com/heatmaps/
14. https://www.youtube.com/watch?v=YncZ0WwxyzU&pp=ygUpc21zIHNwYW0gZGV0ZWN0aW9uIHVzaW5nIG1hY2hpbmUgbGVhcm5pbmc%3D
15. https://www.youtube.com/watch?v=mPW9bjVXbPU&pp=ygUpc21zIHNwYW0gZGV0ZWN0aW9uIHVzaW5nIG1hY2hpbmUgbGVhcm5pbmc%3D