



Grofers Housefull Sale Is Live!

Flat Rs.100 Cashback!  
Code HF100  
Min Order: Rs.[1000](#)

Offer valid till midnight!

Lowest Prices on Grocery!

TCA  
[bit.ly/Gro](#)

2 19:18



Grofers Housefull Sale Is Live!

Flat Rs.100 Cashback!  
Code HF100  
Min Order: Rs.[1000](#)

Offer valid till midnight!

Lowest Prices on Grocery!

TCA  
[bit.ly/Gro](#)

2 19:18



Grofers Housefull Sale Is Live!

Flat Rs.100 Cashback!  
Code HF100  
Min Order: Rs.[1000](#)

Offer valid till midnight!

Lowest Prices on Grocery!

TCA  
[bit.ly/Gro](#)

2 19:1



Grofers Housefull Sale Is Live!

Flat Rs.100 Cashback!  
Code HF100  
Min Order: Rs.[1000](#)

Offer valid till midnight!

Lowest Prices on Grocery!

TCA  
[bit.ly/Gro](#)

2 19:18



Grofers Housefull Sale Is Live!

Flat Rs.100 Cashback!  
Code HF100  
Min Order: Rs.[1000](#)

Offer valid till midnight!

Lowest Prices on Grocery!

TCA  
[bit.ly/Gro](#)

2 19:18

# SMS SPAM CLASSIFIER



Grofers Housefull Sale Is Live!

Flat Rs.100 Cashback!  
Code HF100  
Min Order: Rs.[1000](#)

Offer valid till midnight!

Lowest Prices on Grocery!

TCA  
[bit.ly/Gro](#)

2 19:18



Grofers Housefull Sale Is Live!

Flat Rs.100 Cashback!  
Code HF100  
Min Order: Rs.[1000](#)

Offer valid till midnight!

Lowest Prices on Grocery!

TCA  
[bit.ly/Gro](#)

2 19:18



Grofers Housefull Sale Is Live!

Flat Rs.100 Cashback!  
Code HF100  
Min Order: Rs.[1000](#)

Offer valid till midnight!

Lowest Prices on Grocery!

TCA  
[bit.ly/Gro](#)

2 19:18



Grofers Housefull Sale Is Live!

Flat Rs.100 Cashback!  
Code HF100  
Min Order: Rs.[1000](#)

Offer valid till midnight!

Lowest Prices on Grocery!

TCA  
[bit.ly/Gro](#)

2 19:1



Grofers Housefull Sale Is Live!

Flat Rs.100 Cashback!  
Code HF100  
Min Order: Rs.[1000](#)

Offer valid till midnight!

Lowest Prices on Grocery!

TCA  
[bit.ly/Gro](#)

2 19:1

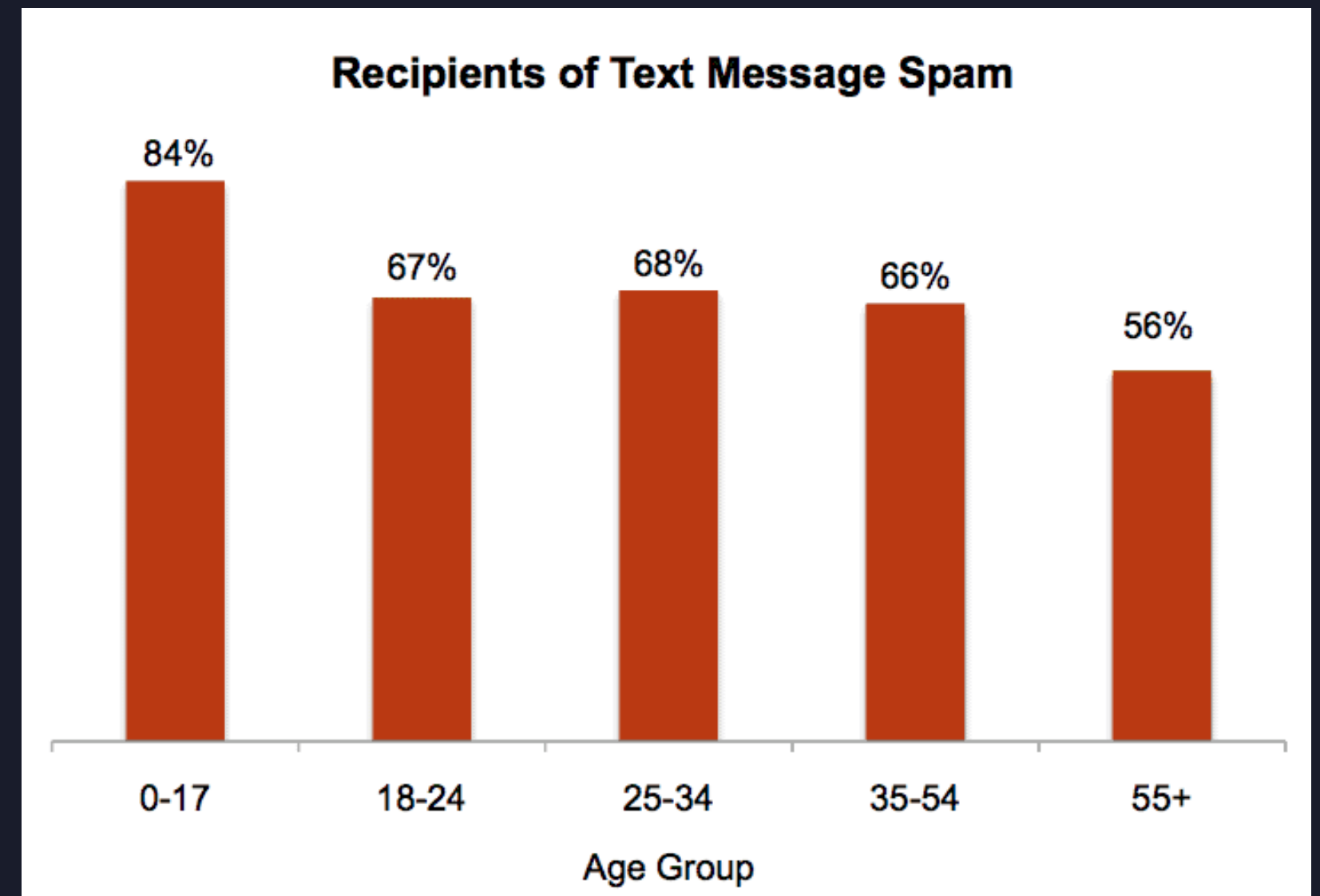
# WHAT IS SPAM ?

**Spam SMS are unwanted text messages that are usually either someone promoting a product or service, or someone attempting to scam you into providing personal information. If you've received a message that doesn't identify the sender, you didn't consent to receiving, or that makes an offer too good to be true, it's likely to be spam.**

According to a survey ,500 U.S. consumers were asked to share their experience with text message spam.

**Interesting text message spam statistics from the survey:**

- **68% of survey respondents say they've received text message spam.**
- **Women under 17 are the most likely to have received text message spam (86%).**
- **Women 55+ are the least likely to receive text message spam (51%) .**
- **Men & Women are equally likely to be the recipients of text message spam.**



# WHY SPAMMING IS A PROBLEM ?

- Spamming wastes people's time with unwanted email or text messages.
- In addition to that, spam also eats up a lot of network bandwidth.
- Spamming can also be used to spread computer viruses or other malicious software.
- Some spam attempts to capitalize on human greed, while some takes advantage of the victims inexperience with computer technology to trick them.
- In some countries SMS Spam contribute to a cost for the receiver as well.
- Spam is not just a nuisance. It absorbs bandwidth and the mail servers of ISP. The cost is now widely estimated in the billions of dollars a year.
- Other problems include Fraud, Theft, Causing harm to the market place and global implications etc.

# OBJECTIVE

The aim is to distinguish between ham messages and spam messages by making an efficient and sensitive classification model that gives good accuracy with low false positive rate.

We will present the same mechanism which can filter spam and non-spam messages using two different machine learning methods and will compare them both. Our proposed algorithm generates dictionaries and features and trains them through machine learning for effective results.



# RELATED WORK

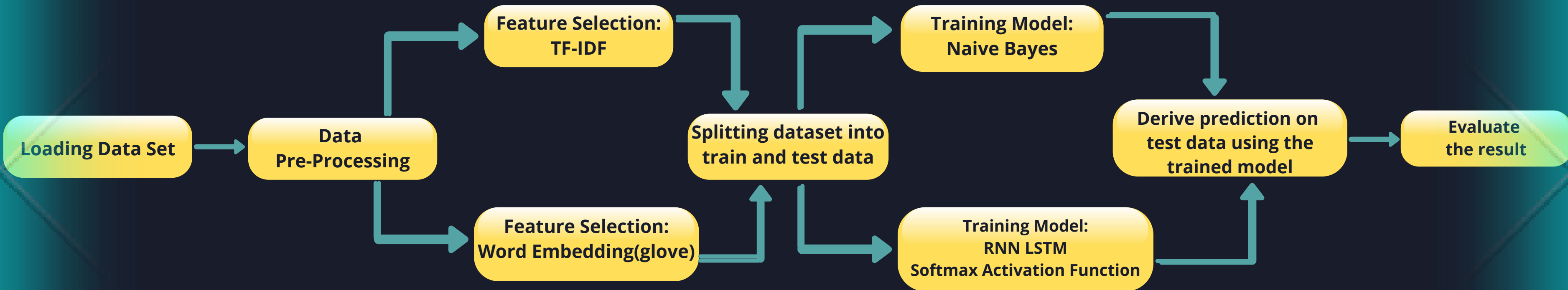
SERIAL NO	AUTHOR	PURPOSE	SOLUTION
1	GOMATHAM SAI SRAVYA, G PRADEEPINI, VADDESWARAM, GUNTUR - 2020	TO GIVE THE BETTER ACCURACY	BY USING CLASSIFICATION ALGORITHMS LIKE RANDOM FOREST CLASSIFIER, DECISION TREE CLASSIFIER, SUPPORT VECTOR MACHINES, ETC.
2	PRADEEP KUMAR ROY, JYOTI PRAKASH SINGH, SNEHASISH BANERJEE - 2020	TO FILTER SMS SPAM EFFICIENTLY	BY USING DEEP LEARNING-BASED MODELS SUCH AS CNN AND LSTM ALONG WITH MACHINE LEARNING BASED CLASSIFIERS SUCH AS NB, RF, GB.
3	PAVAS NAVANEY, GAURAV DUBEY, AJAY RANA - 2018	TO DETECT BETTER ACCURACY	BY USING VARIOUS SUPERVISED MACHINE LEARNING ALGORITHMS LIKE NB, SVM, AND THE MAXIMUM ENTROPY ALGORITHM.
4	MEHUL GUPTA ADITYA BAKLIWAL, SHUBHANGI AGARWAL, PULKIT MEHNDIRATTA - 2018	EVALUATING MACHINE LEARNING TECHNIQUES FOR SPAM SMS DETECTION.	BY COMPARING BETWEEN TRADITIONAL MACHINE LEARNING TECHNIQUES (NB, SVM) AND DEEP LEARNING METHODS (CNN).
5	S.M. ABDULHAMID, M.S. ABD LATIFF, H. CHIROMA, O. OSHO, G.A. SALAAM, A.I. ABUBAKAR, T.HERAWAN - 2017	DETECTION AND FILTERING OF SMS SPAMS	BY USING DIFFERENT MACHINE LEARNING ALGORITHMS LIKE SVM AND BAYESIAN CLASSIFIERS.
6	LUTFUN NAHAR LOTA, B M MAINUL HOSSAIN - 2017	INCREASING THE ACCURACY AND DECREASING THE COMPLEXITY	BY HAVING THE SVM ALGORITHM, IT GIVES BETTER ACCURACY BUT SUFFERS FROM IMPLEMENTATION COMPLEXITY.

SERIAL NO	AUTHOR	PURPOSE	SOLUTION
7	NEELAM CHOUDHARY, ANKIT KUMAR JAIN - 2017	FOR BETTER ACCURACY	USED FIVE MACHINE LEARNING ALGORITHMS NAMELY LOGISTIC REGRESSION, NAIVE BAYES, J48, DECISION TREES AND RANDOM FOREST.
8	NARESH KUMAR NAGWANI, AAKANKSHA SHARAFF - 2017	DETECTING THE SPAM AND NONSPAM MESSAGES	BY USING CLUSTERING TECHNIQUES, WE CAN DETECT THE SMS SPAMS AND FIND THE BETTER ACCURACY.
9	SAKSHI AGARWAL, SANMEET KAUR, SUNITA GARHWAL - 2015	TO GIVE THE BETTER ACCURACY	SVM AND MULTINOMIAL NAIVE BAYES ARE USED BY HAVING THE DATASET AND CALCULATED THE ACCURACY FOR BETTER SCORE
10	DR. GHULAM MUJTABA, MAJID YASIN - 2014	FOR BETTER ACCURACY AND PERFORMANCE	USED NAIVE BAYES CLASSIFIER WITH HYPERTUNED PARAMETERS TO ACHIEVE BETTER PERFORMANCE AND ACCURACY
11	HOUSHMAND SHIRANI-MEHR - 2013	REDUCE THE SPAM MESSAGES AND FOR BETTER ACCURACY	BY USING UCI REPOSITORY DATASET WITH DIFFERENT MACHINE LEARNING ALGORITHMS
12	KULDEEP YADAV, S.K. SAHA, PONNURANGAM KUMARAGURU, ROHIT KUMRA - 2012	TO DETECT THE SMS SPAM MESSAGES AND CALLS	BY USING SVM, WE CAN DETECT THE SMS AND GIVE THE BETTER ACCURACY

# METHODOLOGY

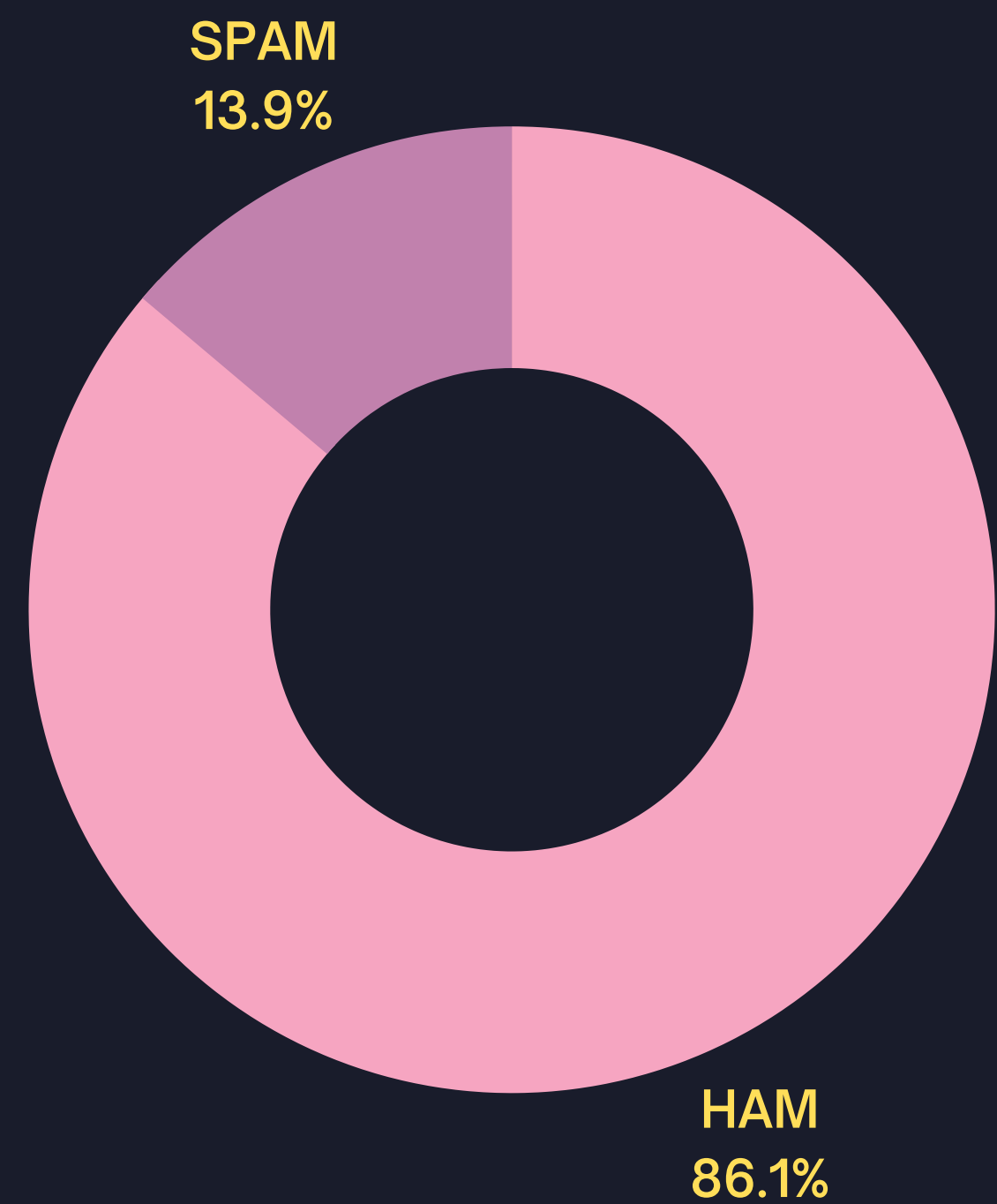
# ARCHITECTURE

Architecture using Naive Bayes



Architecture using LSTM unit of RNN

# DATASET DESCRIPTION



	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

Label	No. of Entries
Ham	4802
Spam	772
Total	5574

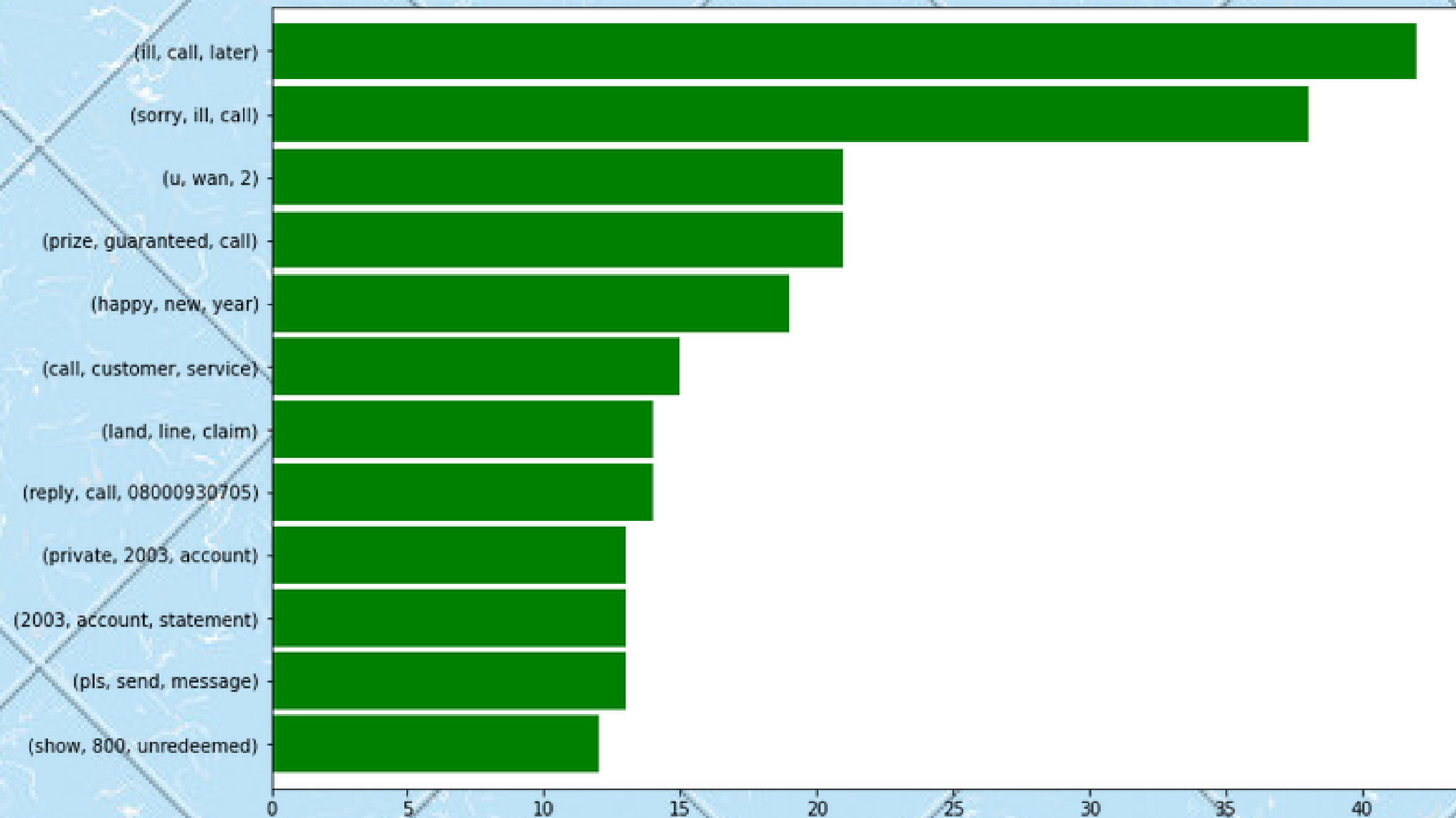


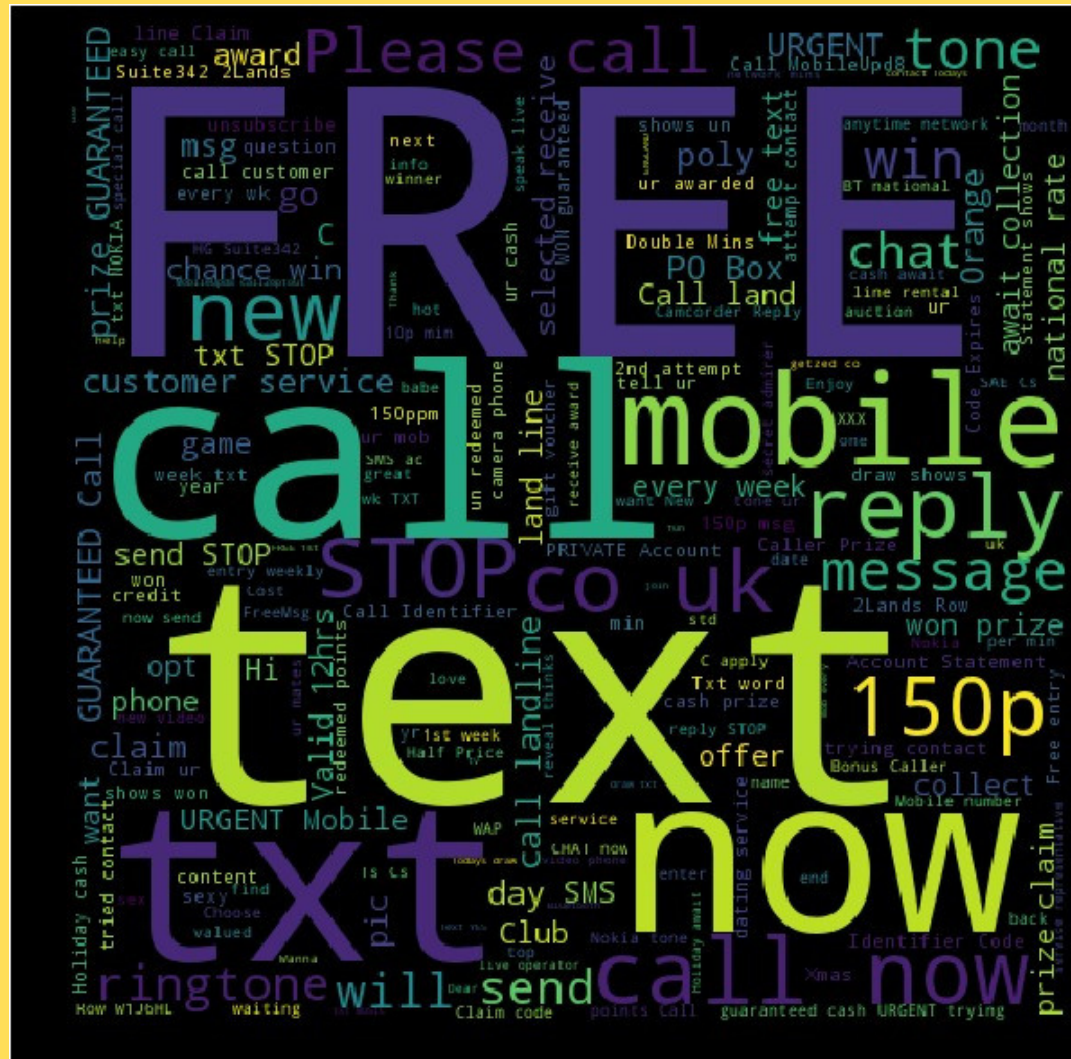
# N GRAM VISUALIZATION

In n-gram ranking, we simply rank the n-grams according to how many times they appear in a body of text. This is extremely useful in predicting the next set of words when a word is given. For example, we see that the words "**please call**" is a bi-gram which occurs the maximum number of times in the dataset hence there is a very good chance that the word please will be followed by call and this helps us get a spatial understanding of the dataset.

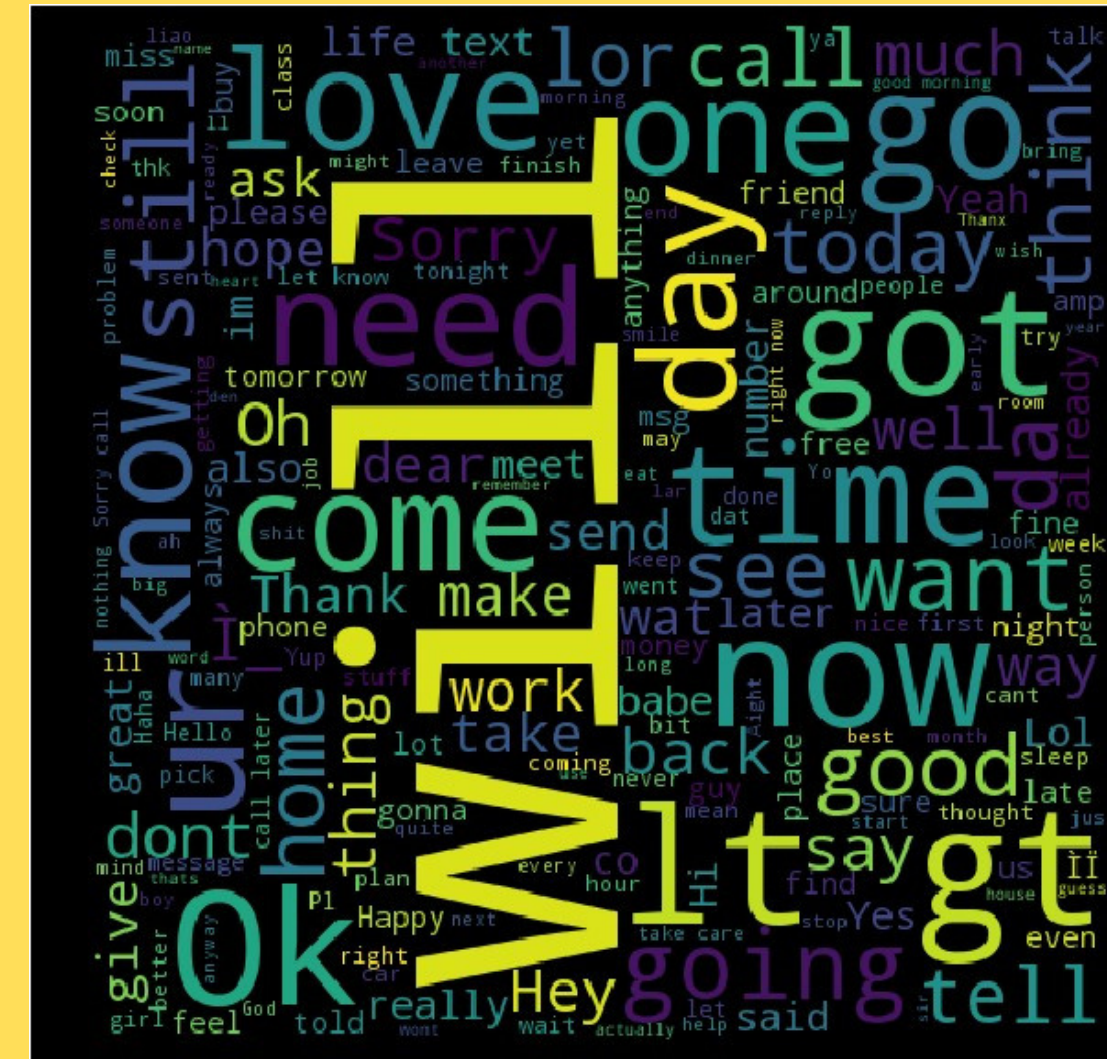


20 most frequently occurring tri-grams.





# Wordcloud for Spam

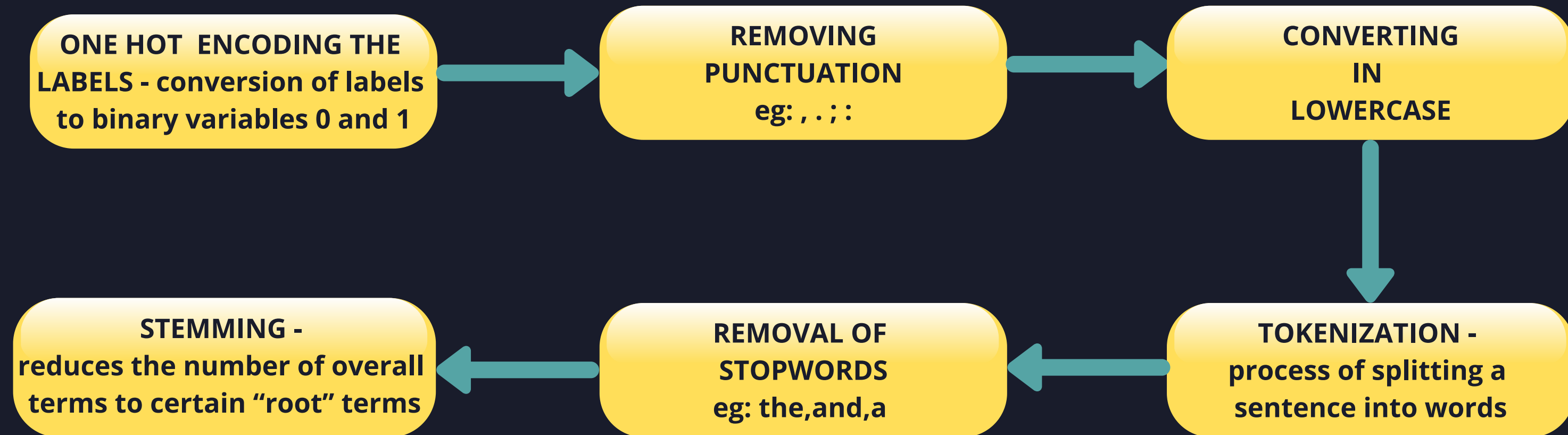


## Wordcloud for Ham

A word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it's mentioned within a given text and the more important it is. Also known as tag clouds or text clouds, these are ideal ways to pull out the most pertinent parts of textual data, from blog posts to databases.



# DATA PRE-PROCESSING



The 6 steps involved in data pre-processing the dataset

# **FEATURE EXTRACTION with TF-IDF**

## **( Term Frequency Inverse Document Frequency )**

Machine learning with natural language is faced with one major hurdle – its algorithms usually deal with numbers, and natural language is, well, text. So we need to transform that text into numbers, otherwise known as text vectorization. It's a fundamental step in the process of machine learning for analyzing data.

Once we've transformed words into numbers, in a way that's machine learning algorithms can understand, the TF-IDF score can be fed to algorithms such as Naive Bayes or Support Vector Machines, greatly improving the results of more basic methods like word counts.

# TF-IDF calculation and algorithm

*TFIDF score for term  $i$  in document  $j = TF(i, j) * IDF(i)$*

*where*

*IDF = Inverse Document Frequency*

*TF = Term Frequency*

$$TF(i, j) = \frac{\text{Term } i \text{ frequency in document } j}{\text{Total words in document } j}$$

$$IDF(i) = \log_2 \left( \frac{\text{Total documents}}{\text{documents with term } i} \right)$$


*and*

*$t$  = Term*

*$j$  = Document*

TF-IDF Calculation formulae

Scikit Learn library already has a tf-idf vectorizer package



```
#import the TfidfVectorizer from Scikit-Learn.  
from sklearn.feature_extraction.text import  
TfidfVectorizer
```

```
vectorizer = TfidfVectorizer(max_df=.65,  
min_df=1, stop_words=None, use_idf=True,  
norm=None)  
transformed_documents =  
vectorizer.fit_transform(all_docs)
```

Code snippet to compute TF-IDF

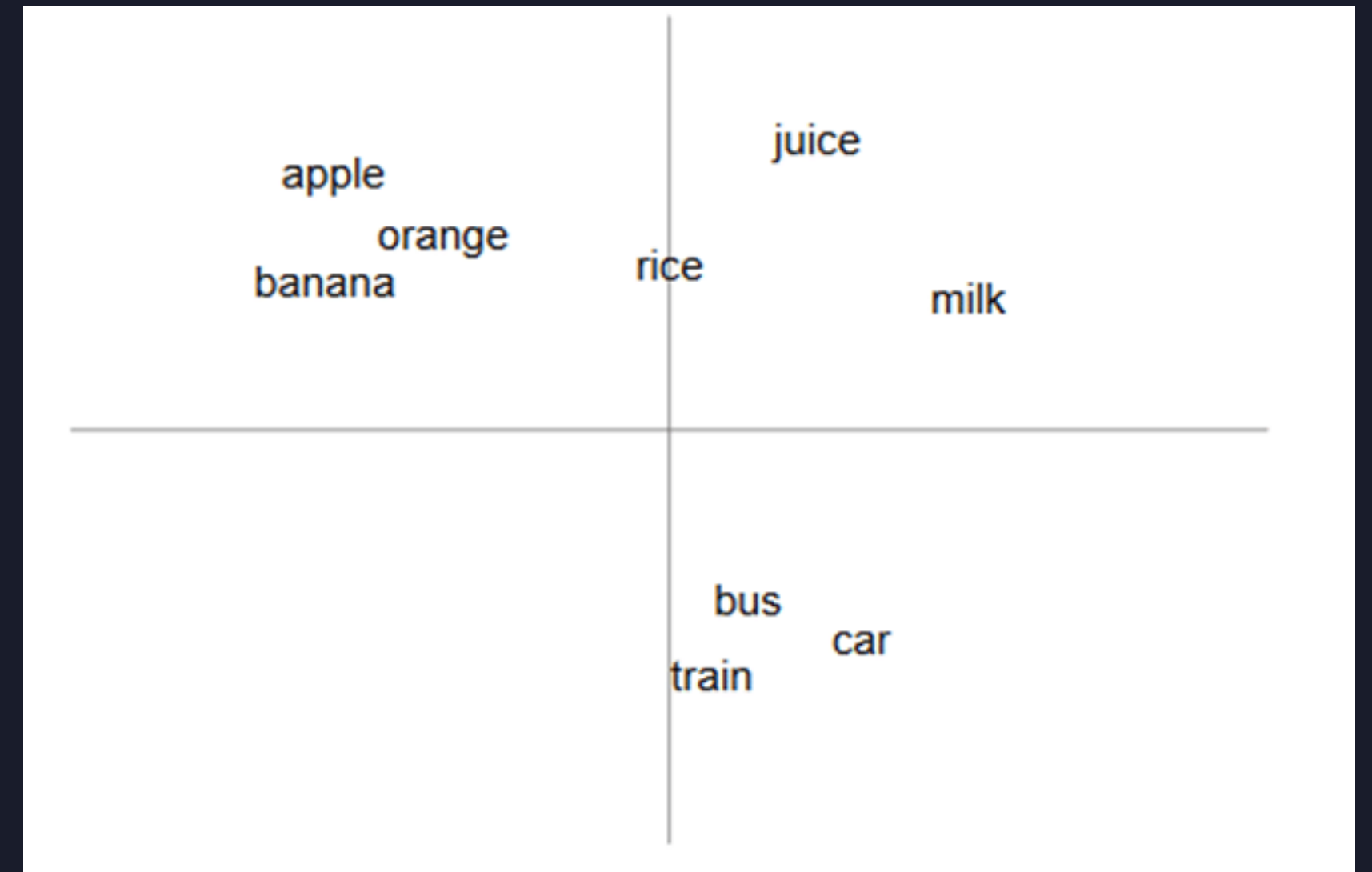


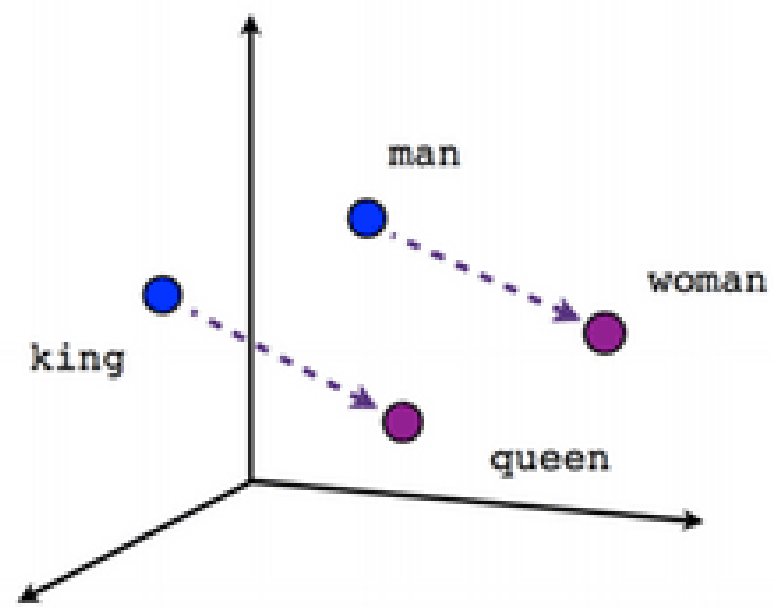
# WORD EMBEDDING

Stores each word in as a point in space, where it is represented by a vector of fixed number of dimensions (generally 300). Dimensions are basically projections along different axes, more of a mathematical concept. Unsupervised, built just by reading huge corpus of words.

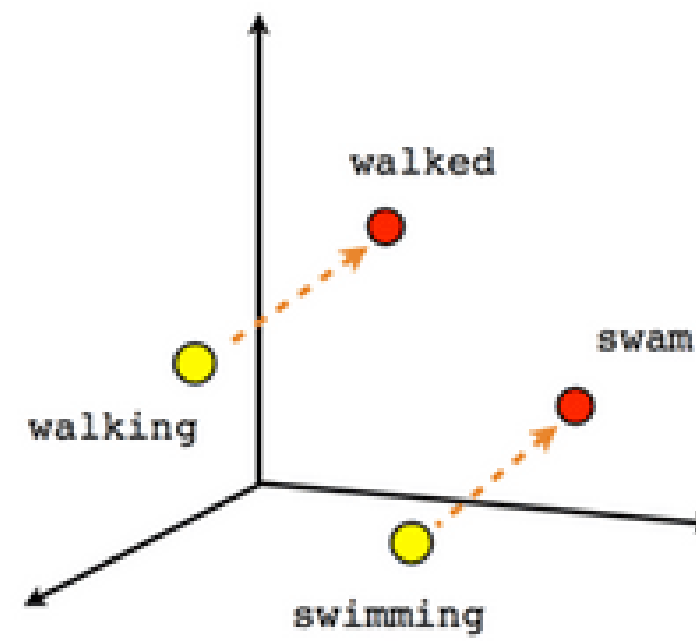
For example, “Hello” might be represented as : [0.4, -0.11, 0.55, 0.3 . . . 0.1, 0.02]

**A word is known by the company it keeps”**

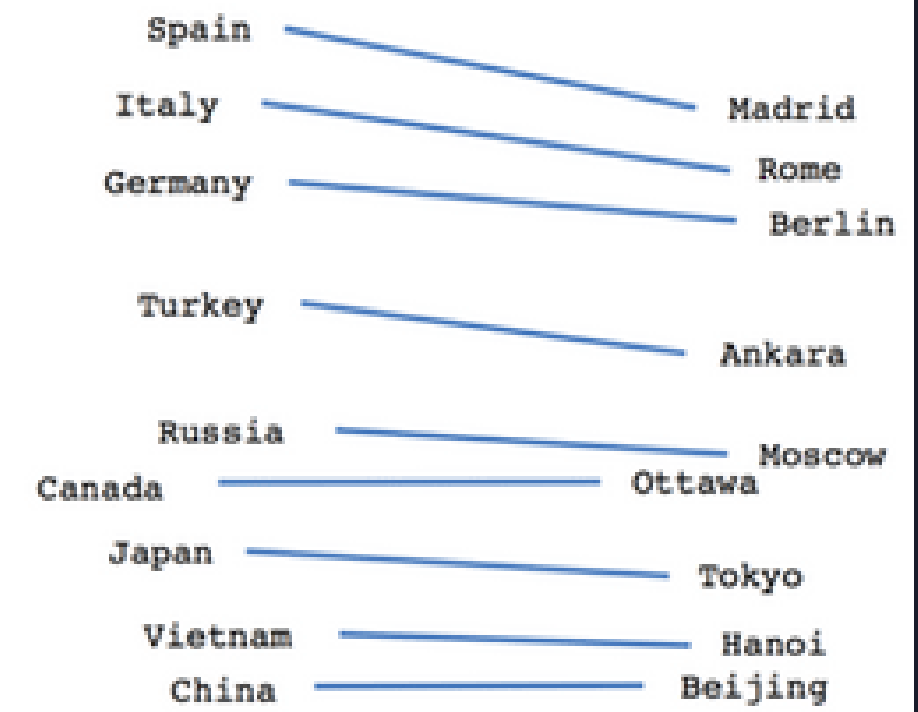




Male-Female

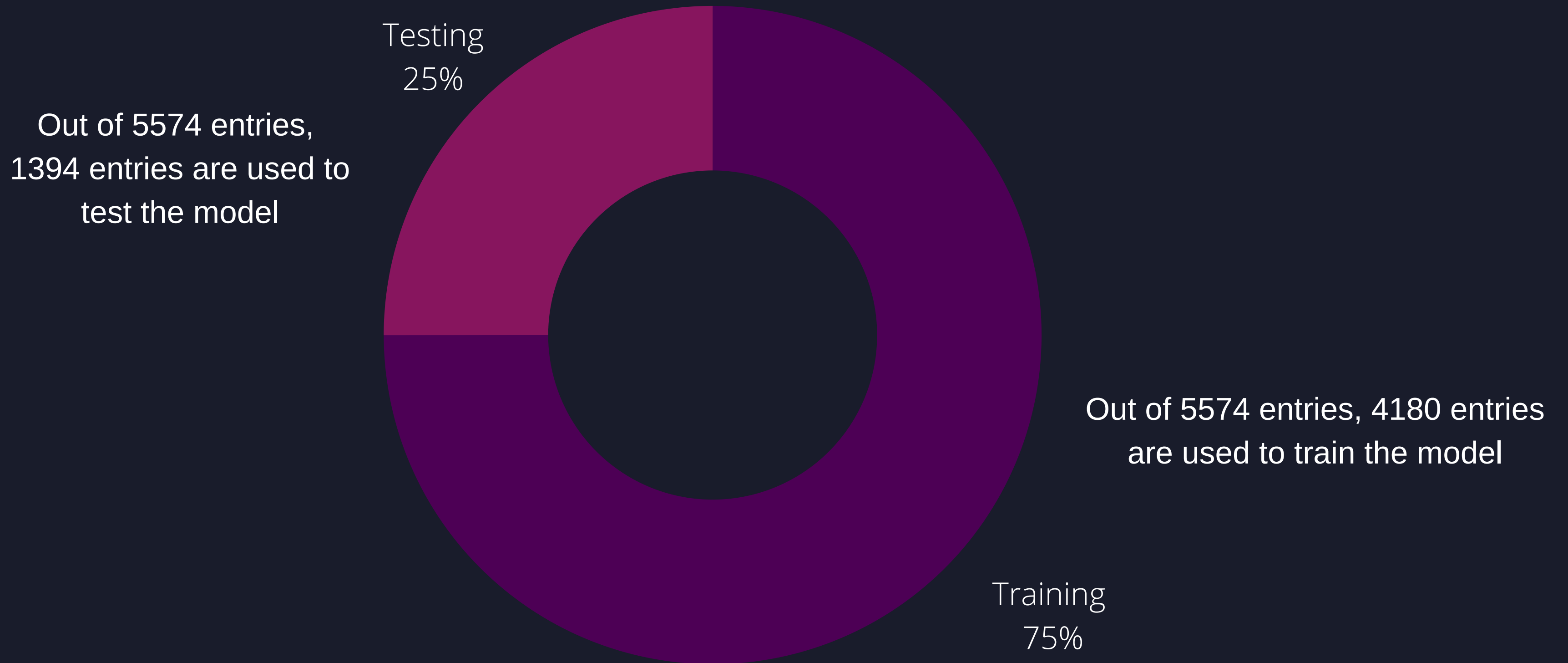


Verb tense



Country-Capital

# TRAINING AND TEST DATASETS



# DIFFERENT CLASSIFIERS

Two different pre-processing approaches have been applied to different classifiers based on their requirement of input data.

Following is a brief description of these approaches.

# 1. NAIVE BAYES CLASSIFIER

In machine learning, Naive Bayes classifiers are a family of simple probabilistic classifiers based on Bayes's Theorem with strong(naive) independence assumptions between the features. The reason the algorithm is called naïve is because it is assumed that :

- 1.) Each feature is linearly independent of the other features, and
- 2.) Each feature in the dataset are as important as any other feature (often neither is the case).

Or in other words, with Naïve Bayes, we assume that the predictor variables are conditionally independent of one another given the response value. This is an extremely strong assumption. Fortunately enough Naïve Bayes is still a strong enough algorithm that has strong results even when this assumption is violated.

Naive Bayes Classifier suitable for following use cases -

- News Classification||Spam filtering||Object Detection||Medical DiagnosisWeather Prediction

# NAIVE BAYES CLASSIFIER - BAYES THEOREM

Naive Bayes Classifier is based on Bayes Theorem which gives conditional probability of an event A given B

*Bayes Theorem*

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$

*where:*

$P(A|B)$  = Conditional Probability of A given B

$P(B|A)$  = Conditional Probability of A given B

$P(A)$  = Probability of event A

$P(B)$  = Probability of event A

$$P_r(spam|word) = \frac{P_r(word|spam)P_r(spam)}{P_r(word)}$$



# NAIVE BAYES CLASSIFIER - IMPLEMENTATION

Given a class variable  $y$  (spam in our case) and a dependent feature vector  $x_1$  through  $x_n$  (the words in each sentence play as a feature), Bayes' theorem states the following relationship:

$$P(y \mid x_1, \dots, x_n) = \{P(y) P(x_1, \dots, x_n \mid y)\} / \{P(x_1, \dots, x_n)\}$$

Using the naive independence assumption that

$$P(x_i \mid y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i \mid y),$$

for all  $i$ , this relationship is simplified to

$$P(y \mid x_1, \dots, x_n) = \{P(y) \prod_{i=1}^n P(x_i \mid y)\} / \{P(x_1, \dots, x_n)\}$$

Since  $P(x_1, \dots, x_n)$  is constant given the input, we can use the following classification rule:

$$P(y \mid x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i \mid y)$$

$$\hat{y} = \arg \max_y P(y) \prod_{i=1}^n P(x_i \mid y),$$

**So how does Naive Bayes work on an actual message according to the formula we saw on the previous slide ??**

**Eg sentence : FREE Money Money Money Money!**

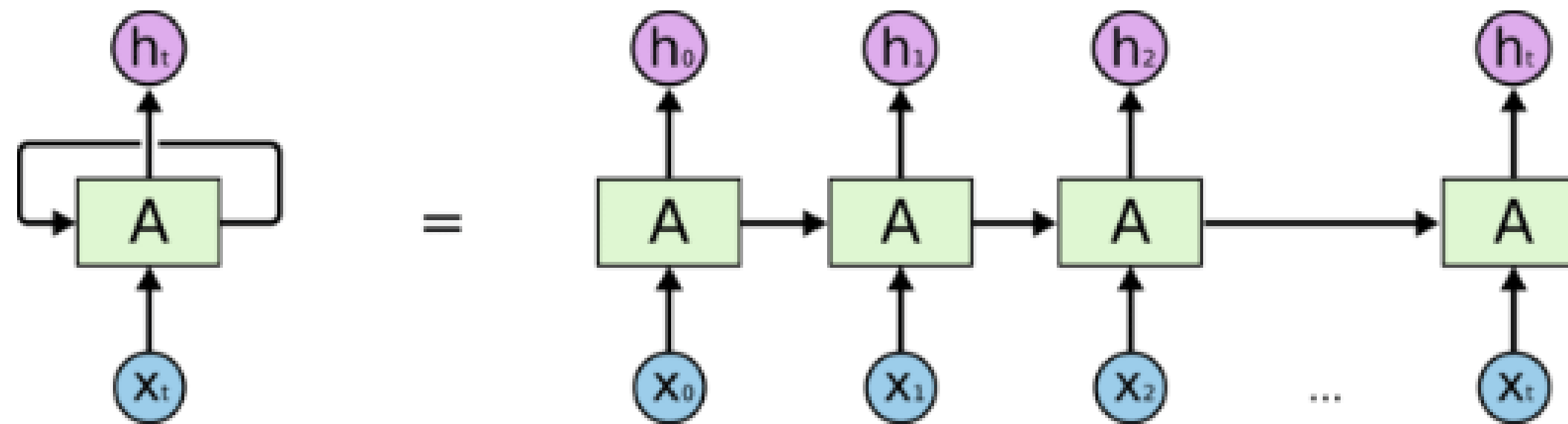
$$P(\text{Spam} | \text{FREE}, \text{Money}, \text{Money}, \text{Money}) = \frac{P(\text{FREE} | \text{Spam}) P(\text{Money} | \text{Spam})^4 P(\text{Spam})}{P(\text{FREE}) * P(\text{Money})^4}$$

$$P(\text{Ham} | \text{FREE}, \text{Money}, \text{Money}, \text{Money}) = \frac{P(\text{FREE} | \text{Ham}) P(\text{Money} | \text{Ham})^4 P(\text{Ham})}{P(\text{FREE}) * P(\text{Money})^4}$$

After calculating the probability of the message being HAM or SPAM we take probability which has the highest value

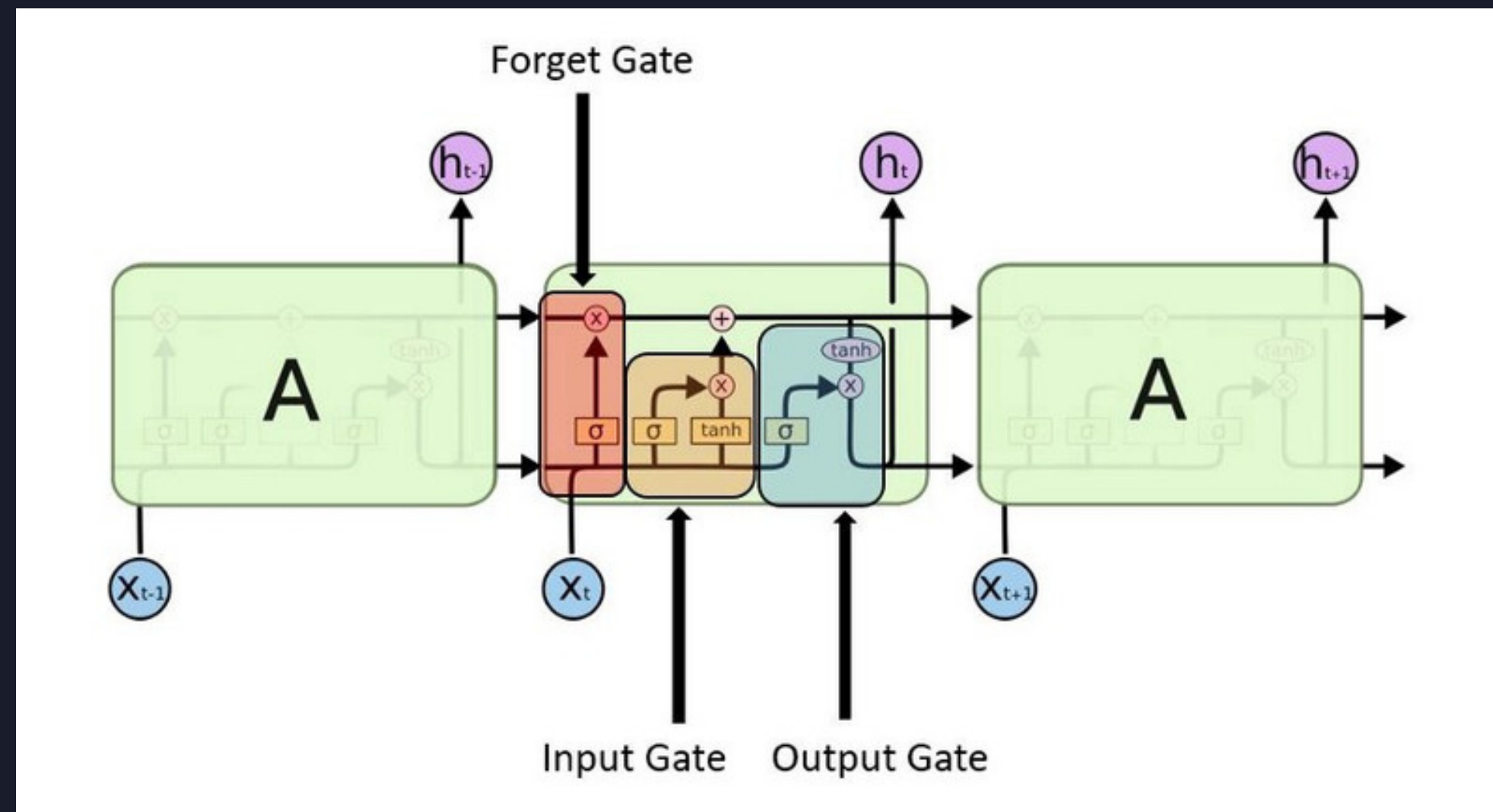
## 2. RNN WITH LSTM UNITS

Recurrent Neural Network is a generalization of feedforward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation.



An unrolled recurrent neural network.

Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using back-propagation. In an LSTM network, three gates are present:



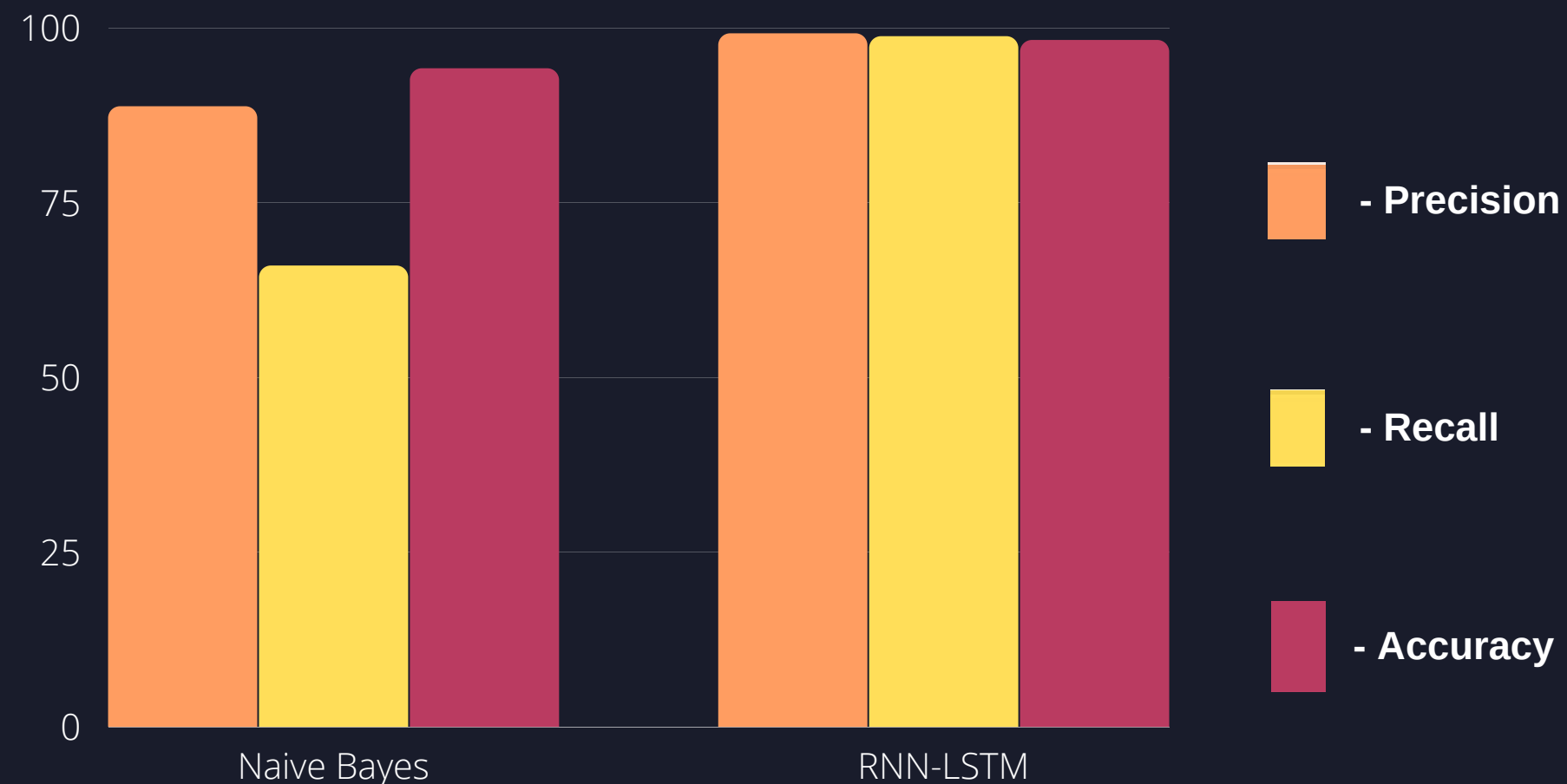
# APPLICATIONS OF LSTM -

Applications include -

- Robot Control
- Time series prediction
- Speech recognition
- Handwriting recognition
- Human action recognition
- Sign language translation
- Airport passenger management
- Drug design
- Market prediction

# TESTING AND EVALUATION

Model	PRECISION	RECALL	ACCURACY
Naive Bayes	88.73 %	65.96 %	94.16 %
RNN (LSTM)	99.16 %	98.75 %	98.21 %



From the table aside, we can clearly see that the LSTM model is doing way better than the Naive Bayes Algorithm. The reasons can be due to:

- Tf-idf Vectorizer did not take into account the ordering of the words in the sentence, and it is losing a great deal of information.
- LSTM has been one of the greatest algorithms in sequence data (text, speech, time-series data) in the recent years.
- Word embedding is definitely a good feature extraction tool for text data and with LSTM model, as we have built a spam filtering system with very decent performance.



# CONFUSION MATRIX

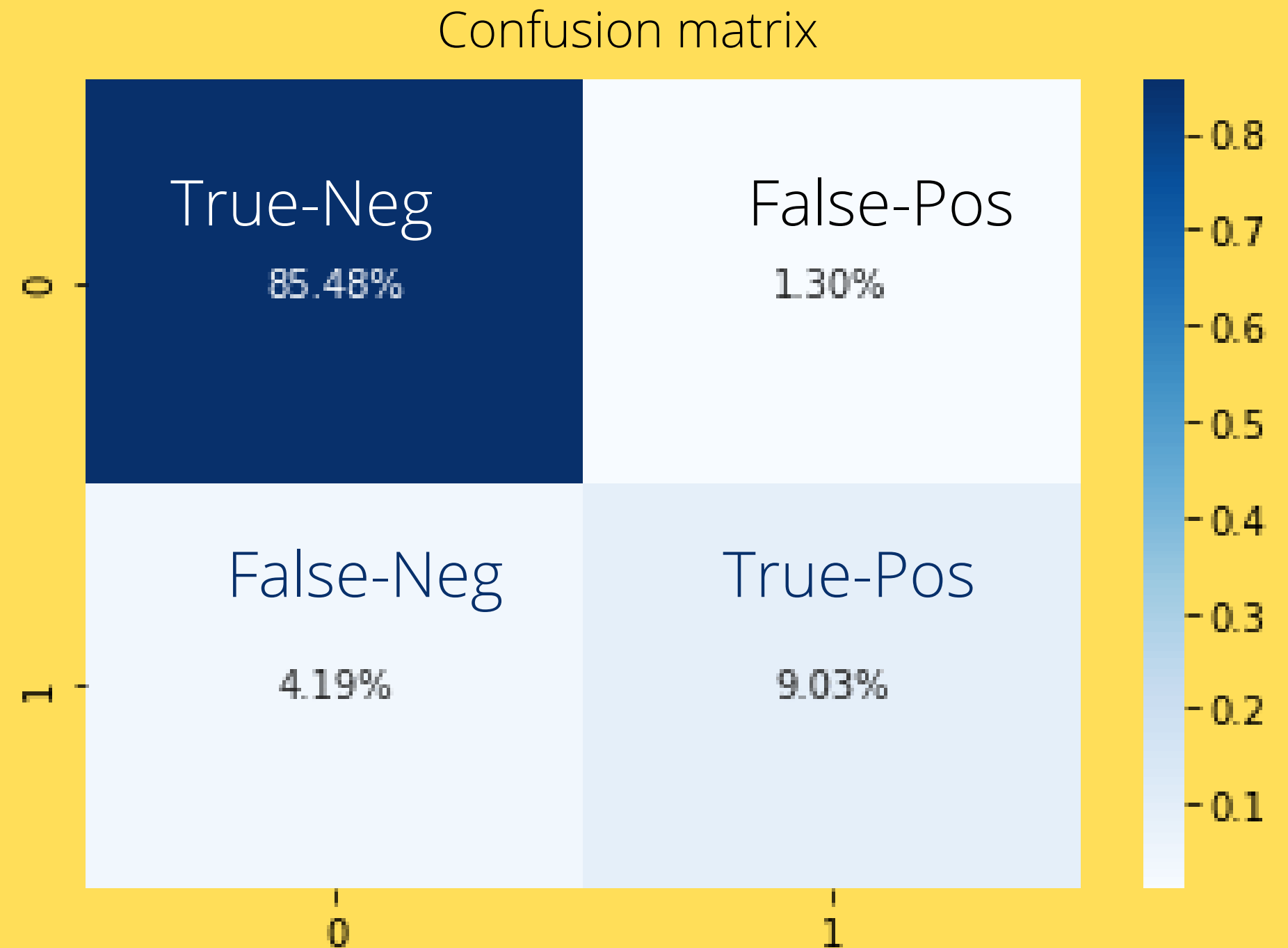
The confusion matrix is a 2 dimensional array comparing predicted category labels to the true label. For binary classification, these are the True Positive, True Negative, False Positive and False Negative categories.

True-Negative ----> 1183

False-Positive ----> 58

False-Negative ----> 18

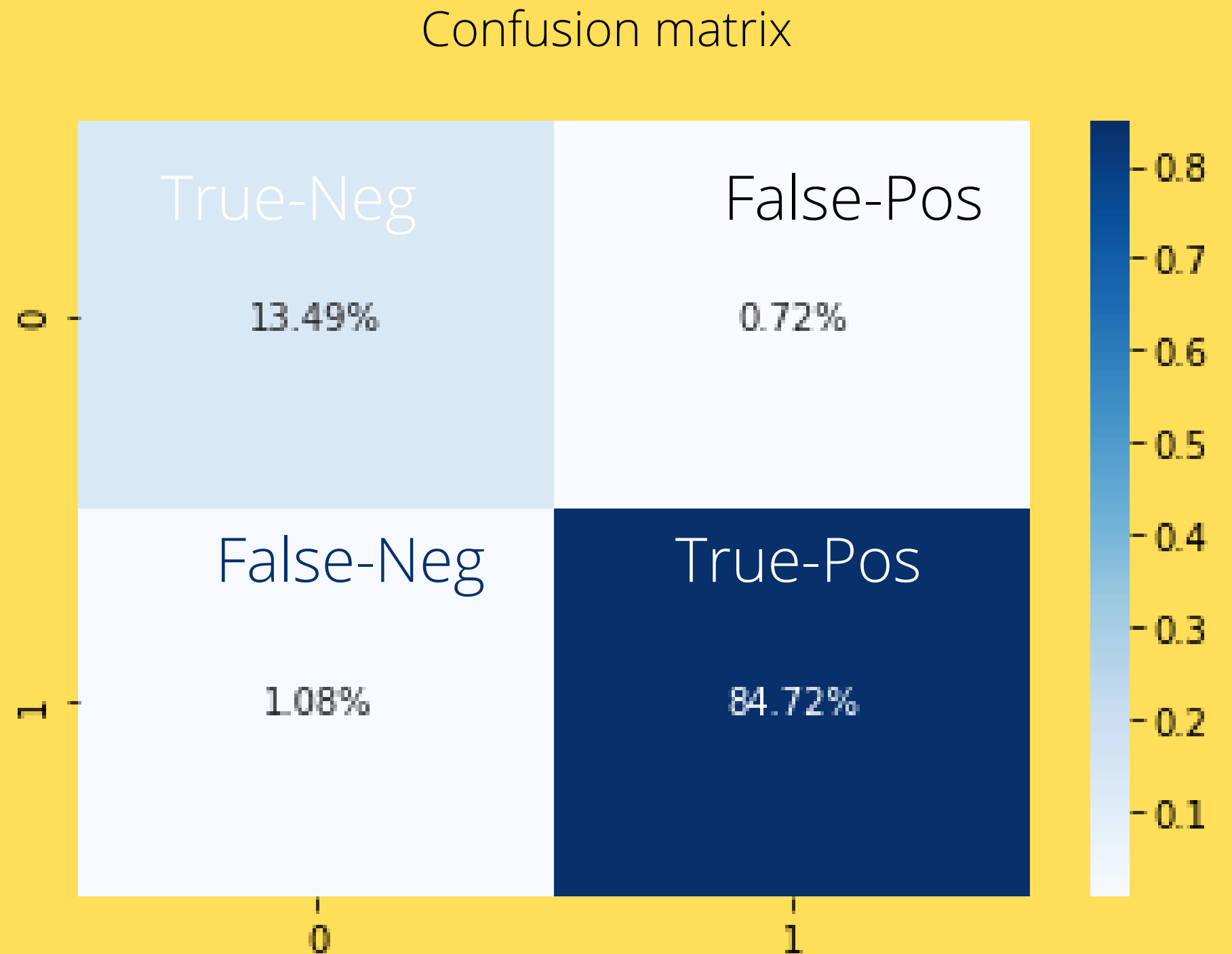
True-Positive ----> 125



Confusion matrix for Naive Bayes Classifier

# CONFUSION MATRIX

True-Negative ----> 188  
False-Positive ----> 10  
False-Negative ----> 15  
True-Positive ----> 1181



```
[31] pm = process_message('I cant pick the phone right now. Pls send a message')  
     sc_tf_idf.classify(pm)
```

False

```
pm = process_message('Congratulations ur awarded $5000 ')  
sc_tf_idf.classify(pm)
```

True

Testing the Naive Bayes Classifier with a spam and a ham message

```
text = "Congratulations! you have won 100,000$ this week, click here to claim fast"  
print(get_predictions(text))
```

Output:

spam

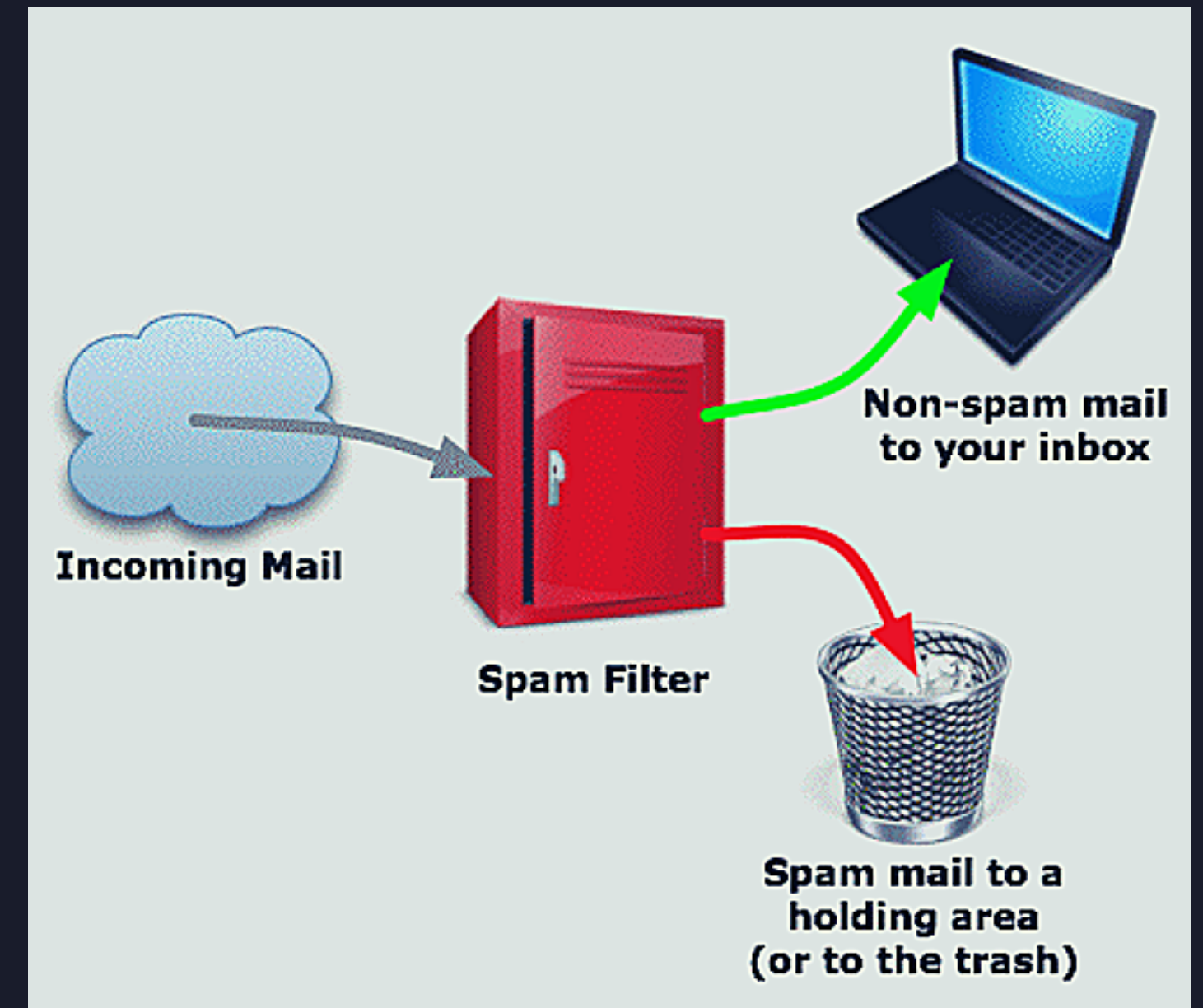
Testing the LSTM model with a spam message

# CONCLUSION

- SMS Spam filtering is the big challenge these days. The main aim of this paper is to compare two different machine learning algorithms i.e., Naïve Bayes and RNN-LSTM with better accuracy score.
- The dataset that we have used in our work consists of 5574 messages which were collected from UCI datasets.
- The text messages are differentiated with Ham or Spam. It predicts whether the message in the dataset is Ham or Spam and predicts the performance through accuracy criterion.
- We can safely conclude after checking the result that building an SMS spam classifier using RNN-LSTM algorithm gives us the better results with an accuracy of 98.21%

# FUTURE SCOPE

- A limitation of the work is that it was dependent on text messages written in English only. Therefore, this paper invites future research to employ similar deep learning approaches to filter Spam and Not-Spam text messages written in other languages too.
- From the initial dataset description we see that the dataset is imbalanced and there are several ways to handle the imbalance data, for instance.
- Use of appropriate evaluation metrics||resampling the training set||oversampling/upsampling or undersampling/downsampling||ensemble different resample datasets, implementing these could further improve the performance of the dataset
- We will also add more machine learning algorithm techniques to obtain best results.



- A better and more reliable classifier can be made if we take a larger training data. A larger corpus will ensure that our model trains from a vast number of words.
- We have used a vanilla architecture for our LSTM model because it has only one layer LSTM we can use advanced models like GRU, bidirectional LSTM to build a better classifier



**THANK YOU !**