SMS SPAM CLASSIFICATION

Assignment -4

Problem solving for SMS spam filtering using artificial intelligence:

- Mobile message is a way of communication among the people and billon of mobile device user exchange numerous messages.
- However, such type of communication is insecure due to lack of proper message filtering mechanisms.
- One cause of such insecurity is spam, and it makes the mobile message communication insecure.
- Spam is considered be one of the serious problem in email and instance message services.
- Spam is a junk or message.
- Spam e-mails and messages are unwanted for receivers which are sent to the users without their prior permission.
- The spam increased in these days due more mobile devices deployed in environment for e-mail and message communication.

- ❖ Currently,85% of mails and messages received by mobile users are spam.
- ❖ Due to these spam mails and messages, the values able e-mails and messages are affected because each user have limited internet services, short time, and memory.

Problem Handling

- ✓ To handle these problems caused by the spam, researchers proposed different techniques to detect the spam e-mails and messages and secure the communication.
- ✓ The e-mail classification method was proposed for the detection of spam.
- ✓ The system obtained good results.
- ✓ Designed ensemble methods based on technique such as bagging, boosting, and stacking for classification of spam and ham.

Methods and materials:

$$\text{accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \times 100\%,$$

$$\text{sensitivity } (Sn) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \times 100\%,$$

$$\text{specificity } (Sp) = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \times 100\%.$$

Experiments and Results Analysis:

Table 1: Description of SMS Spam collection dataset.

Sample no number	V1	V2
1	Ham	Did I forget to tell you? I want you , I need you , I crave you… But most of all…I love you my sweet Arabian steed…Mmmmmm…Yummy
2	Ham	I don't thnk it's a wrong calling between us Awesome, that gonna be soon or later tonight?
3	Spam	UpgrdCentre Orange customer, you may now claim your FREE CAMERA PHONE upgrade for loyalty. Call now on 0207 153 9153. Offer ends 26th July. T&C's apply. Opt-out available
. . . .	. . . .	. . . .
4458	Ham	Sorry, I'll call later

Table 2:Confusion matrix[20,21].

	Predicted spam(1)	Predicated ham(0)
Actual spam(1)	TP	FN
Actual ham(0)	FP	TN

- SMS spam application on the store of android app were assessed.
- We performed experiments to classify the ham and spam using the SMS spam collection dataset.
- Classifiers LR,decision tree, and k-nearest neighbor were used for the classification in this study.
- The dataset is divided as follow:30% for validation and 70% for training.
- The results obtained from experiments are shown in tables and presented in figures graphically.
- Visualization of SVM spam collection dataset.
- In the data, 4900 are harm samples and 672 are spam samples.

Table 3: Classification performance of classifiers.

Predictive model	Evaluation performance measures				
	Accuracy (%)	Specificity (%)	Sensitivity (%)	MCC (%)	Processing time (%)
Logistic regression (C=1)	99	93	86	93	0.494
K-nearest neighbor (K-NN, K=1)	95	80	60	80	0.630
DT	98	95	86	95	46.032

Import required library:

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline
```

In [3]:

```
data = pd.read_csv("../input/spam.csv",encoding='latin-1')
```

Research Objectives and Contributions:

One of the important objectives was to do an investigation on the work already done in the field of text analytics which helped in gaining the knowledge and achieving the below research objectives.

Objectives	Description	Evaluation Matrices
Obj-1	Spam SMS file dataset pre-processing	
Obj-2	Implement Natural Language Processing techniques	
Obj-2a	Implement Bag-Of-Words to create document term matrix for Spam SMS Detection	
Obj-2b	Implement TF-IDF to create document term matrix for Spam SMS Detection	
Obj-3	Apply feature selection technique i.e. Chi-Square method	
Obj-4	Implement, Evaluate and generate the outcome for several machine learning algorithms	
Obj-4a	Implementation, Evaluation, and Results of LightGBM	Accuracy, Precision, Recall, F1-Source, execution Time
Obj-4b	Implementation, Evaluation, and Results of Bernoulli Naive Bayes	Accuracy, Precision, Recall, F1-Source, execution Time
Obj-4c	Implementation, Evaluation, and Results of SVM	Accuracy, Precision, Recall, F1-Source, execution Time
Obj-4d	Implementation, Evaluation, and Results of Random Forest	Accuracy, Precision, Recall, F1-Source, execution Time
Obj-5	Compare the developed models(Obj-5a to Obj-5e)	Accuracy, execution Time
Obj-6	Compare the developed models with the existing state-of-the-art models	Accuracy

Data Pre-processing:

Data pre-processing is mainly cleaning of data by removing unwanted rows, columns, missing values, outliers, etc. For the research, the following pre-processing steps have been taken:

- ✓ Removal of Unwanted Columns
- ✓ Cleaning of Text Messages

In [15]:

```
import nltk, os

#if true it will download all the stopwords
if True:
    os.system('python -m nltk.downloader')
```

In [16]:

```
# importing Natural Language Toolkit
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
#if true will create vectorizer with stopwords
if True:
    stopset = set(stopwords.words("english"))
    vectorizer = TfidfVectorizer(stop_words=stopset,binary=True)
```

In [17]:

```
#if true will create vectorizer without any stopwords
if True:
    vectorizer = TfidfVectorizer()
```

In [18]:

```
# Extract feature column 'Text'
X = vectorizer.fit_transform(data.Text)
# Extract target column 'Class'
y = data.numClass
```

In [19]:

```
#Shuffle and split the dataset into the number of training and testing p
oints
if True:
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=
0.20, train_size=0.80, random_state=42)

# Show the results of the split
```

```
print("Training set has {} samples.".format(X_train.shape[0]))
print("Testing set has {} samples.".format(X_test.shape[0]))
```

Exploratory Data Analysis:

Exploratory Data Analysis is a crucial process where the data is analyzed to uncover underlying patterns, spot abnormality and test the hypothesis . It is the best practice to understand the data and then carry out the data mining process. Missing value analysis was carried out using libraries of python like Pandas. An additional feature of the “length of text messages” was considered as there was a substantial correlation of over 0.6 between the length and the type of SMS as seen in Figure 4. For visualization of most frequent words appearing in spam messages and ham messages, the Matplotlib library with WordCloud technique was used.

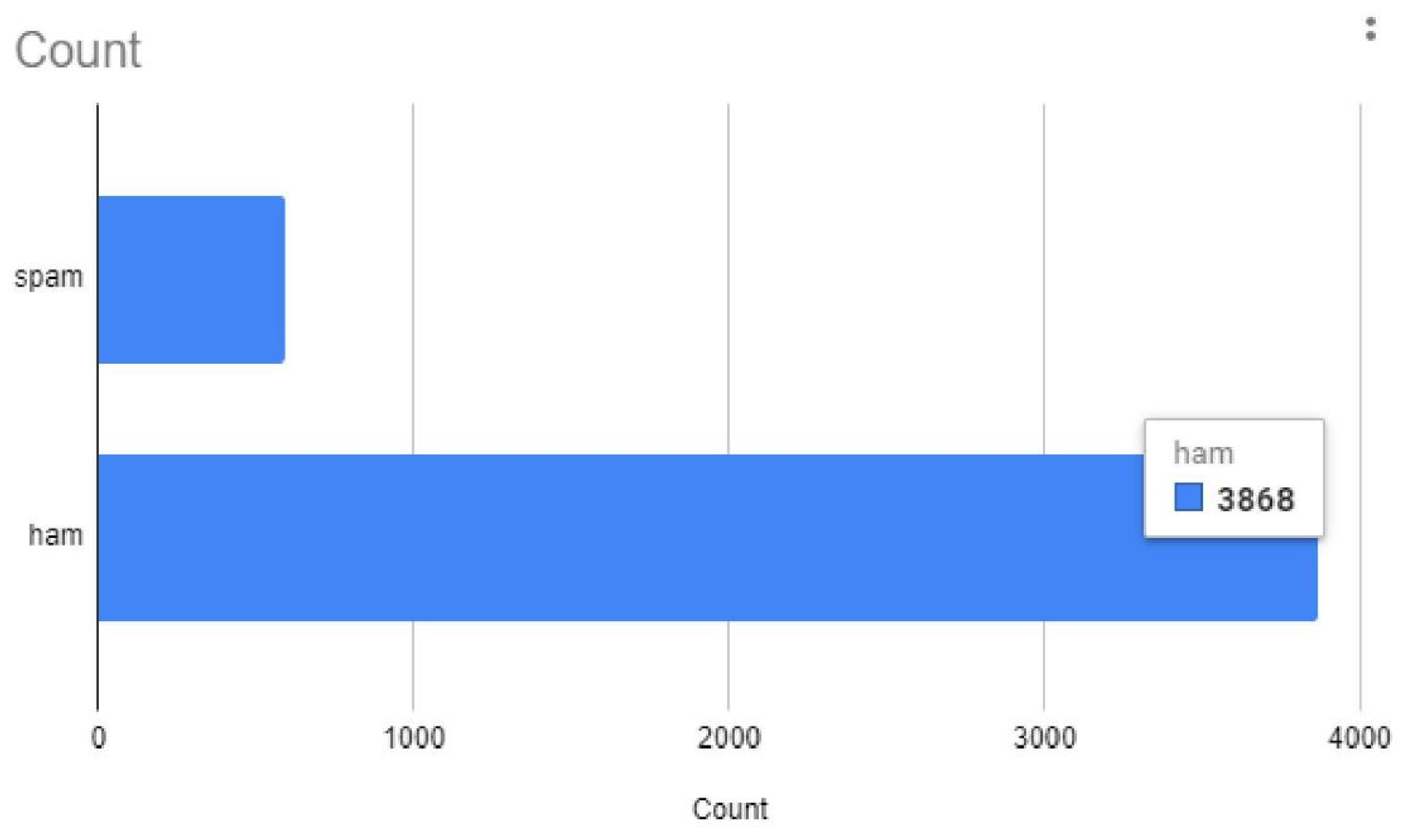


FIGURE 1: Classification of spam and ham message.

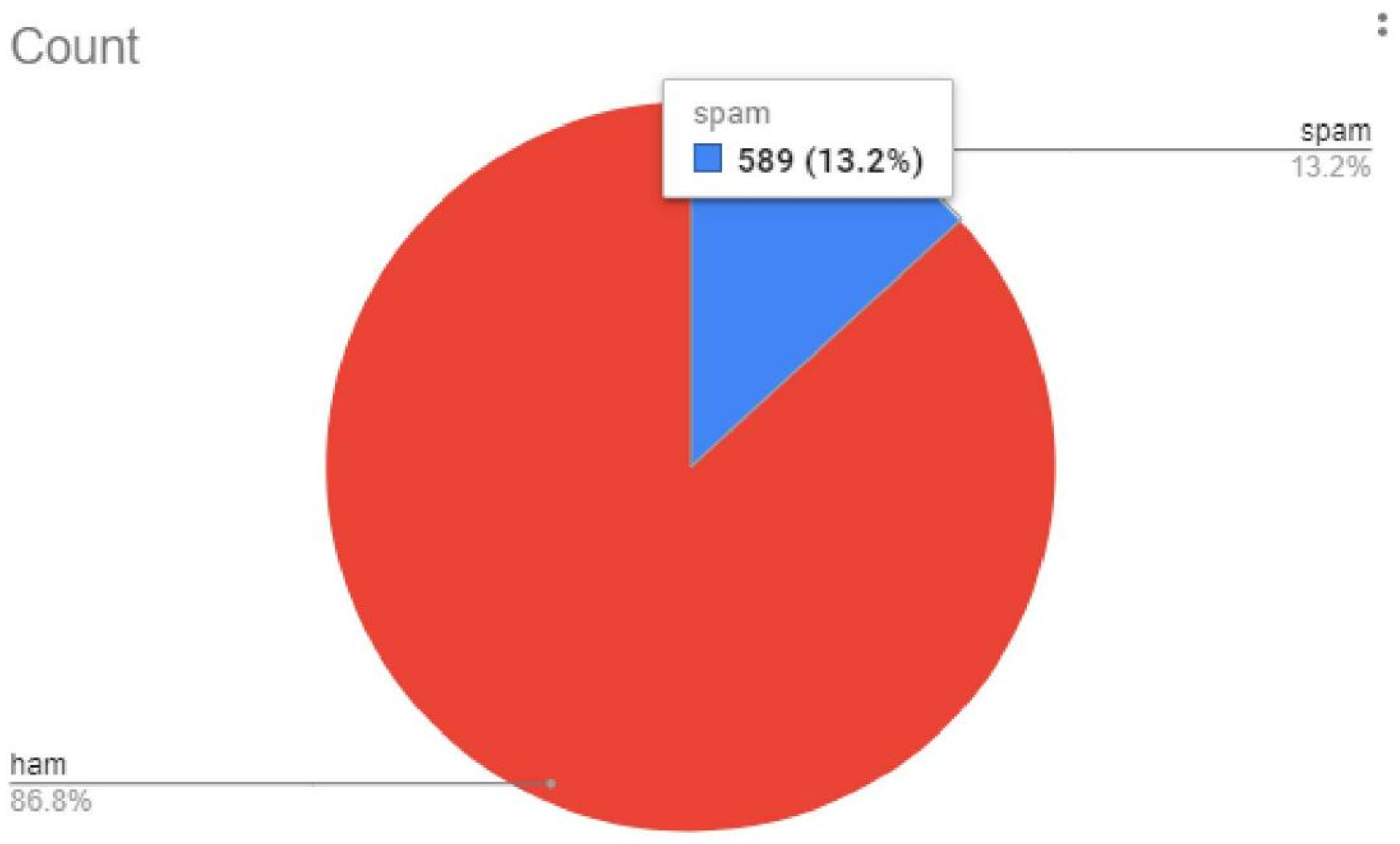
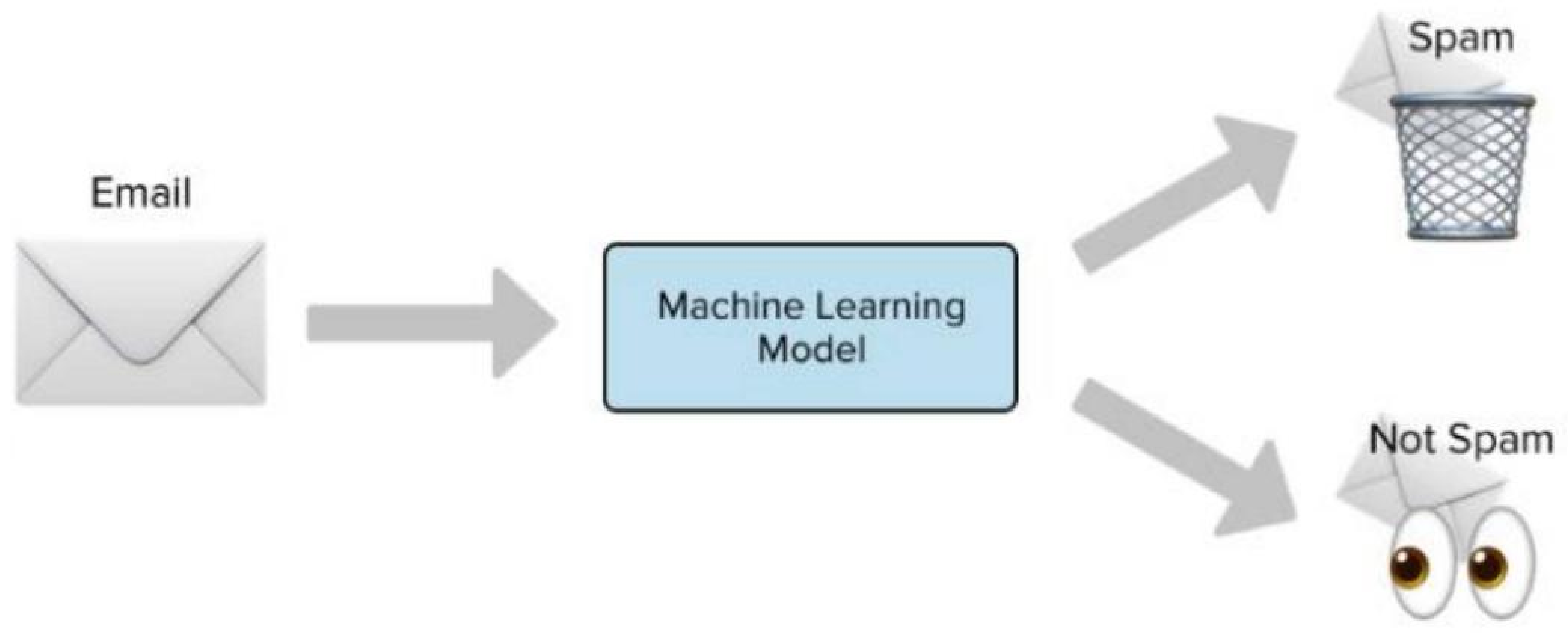


FIGURE 2: Radio of ham and spam message

- 🟦 Spam -589(13.2%)
- 🟥 Ham -3868(86.8%)

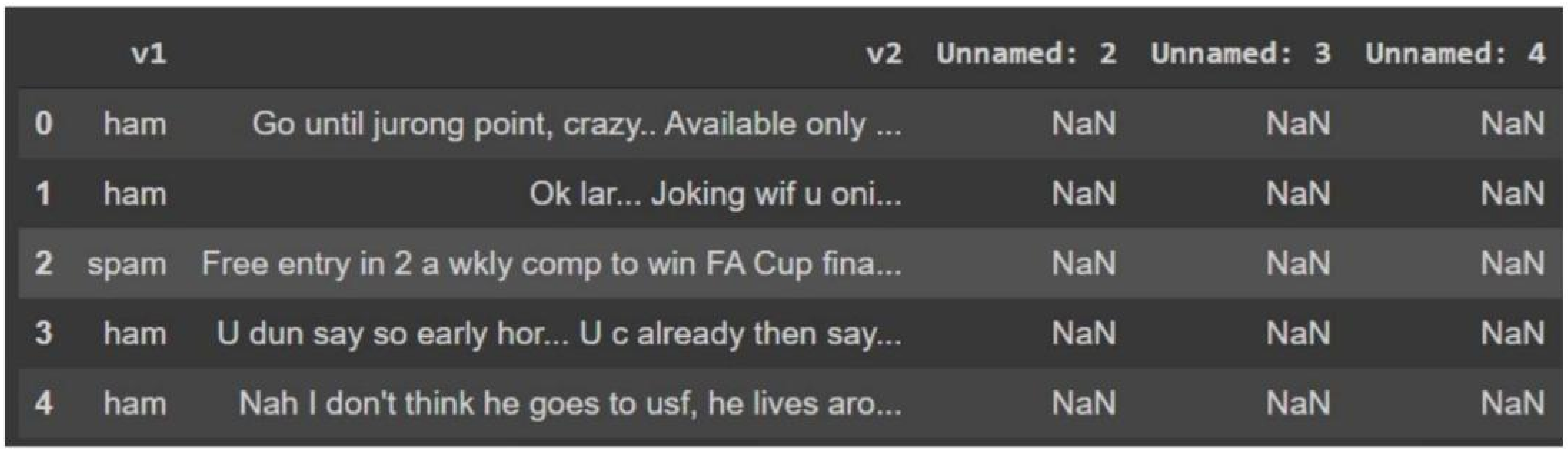
Create model:



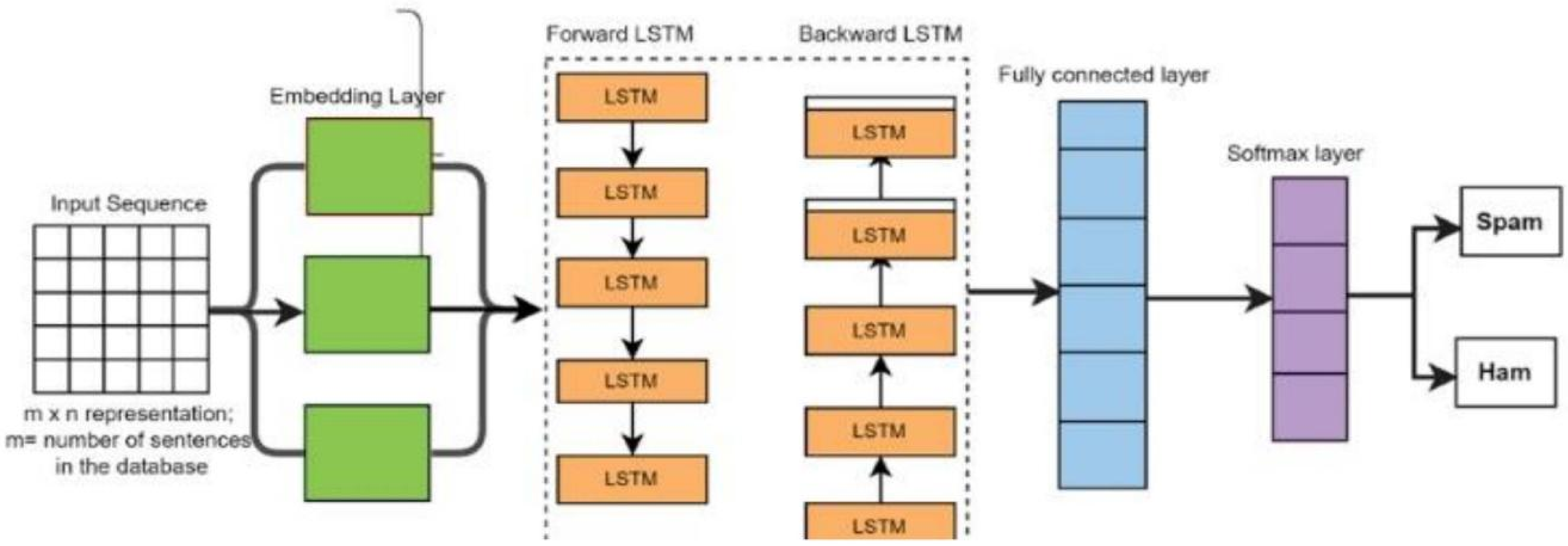
Long Short Term Memory (LSTM):

- ✓ In this article, we are going to create an SMS spam detection model which will help you to find whether an SMS is spam or not using LSTM.
- ✓ To address this, we proposed a novel method long short term memory (LSTM),which is an advanced structure of Recurrent Neural Network (RNN) that has gating mechanism

including memory.



	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



Result Analysis:

It can be observed that the length contributed to the increase in the performance of the model for all the 3 matrices. TF-IDF with length generated the best results in terms of Accuracy, Recall, and F1-Score. The Accuracy stood at 0.964, Recall at 0.955 and F1-Score at 0.96 Precision for Bigram without length feature was the highest amongst all at 0.993.

In [26]:

```
# ploating data for Accuracy Score
# ploating data for Accuracy of Models between 1.00 - 0.90 for better vi
sualization
objects = ('','Untunded', 'Tuned','')
y_pos = np.arange(4)
y_val = [0,0.03470790378,0.037062937063,0 ]
plt.bar(y_pos,y_val, align='center',width = 0.5, alpha=0.6)
plt.xticks(y_pos, objects)
plt.ylabel('Accuracy Score')
plt.title('Accuracy of AdaBoost')
plt.show()
```