DEEP LEARNING – SE4050

ASSIGNMENT 02

B.Sc. (Hons) Degree in Information Technology

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

Group Members

Student ID	Name	Email
IT18149654	Rajapaksha T.N.	it18149654@my.sliit.lk
IT18148350	Naidabadu N.I.	it18148350@my.sliit.lk
IT18149272	Perera M.J.F.R.	it18149272@my.sliit.lk

Problem

Short Message Service (SMS) has become one of the most common and heavily used ways of communication in society today. According to the research conducted by Ghourabi, Mahmood, and Alzubi, mobile phone users around the world have sent more than 8.3 trillion SMS messages in the year 2017 [1]. Furthermore, the number of SMS messages sent monthly is more than 690 billion [1].

However, SMS spam has been widely spreading among most mobile phone users recently. Any undesired or unsolicited text message sent indiscriminately to your mobile phone, usually for commercial motives, is referred to as SMS spam [1, 2, 3]. According to a survey conducted in the research paper [1], more than 68% of mobile phone users around the globe are affected by SMS spam messages.

Most people face many difficulties in their day-to-day lives due to SMS spam messages. Spams annoy most people as their valuable time is wasted and their workflow is interrupted and disturbed when they have to go through unwanted messages [3]. Additionally, the important text messages can be missed due to the inbox being filled with unwanted spam SMS messages. Furthermore, these SMS spams waste network resources of the device [3]. Therefore, SMS spam can cause significant negative sociological and economical effects.

Furthermore, in many cases, SMS spams consist of malicious activities such as smishing (SMS + phishing) which is a cyber security threat for mobile users aimed at deceiving them via SMS spam messages that may include a link or malicious software [1]. Cyber-attackers are trying to steal users' secret and sensitive information such as credit card numbers, passwords, and bank account details [1] using these malicious SMS spam messages. Many individuals and even major organizations suffer huge financial losses due to the cyber security threats caused by SMS spam messages [1].

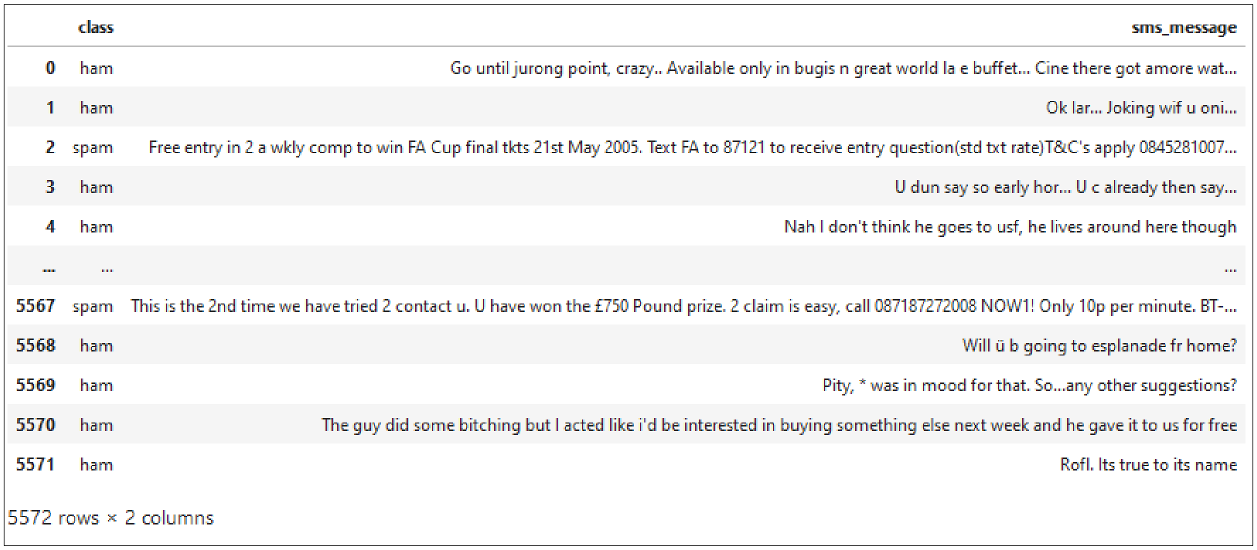
The filtration of SMS spam in smartphones is still not very robust compared to the filtration of email spam detection [1]. Identification of text spam messages is also proven to be a very hard and time-consuming task according to most research [1, 2, 3]. Most existing lexicon-based methodologies as well as traditional NLP (Natural Language Processing) and Machine Learning algorithms are not accurate and efficient enough. Therefore, addressing this real-world problem and coming up with a reliable technique to classify spam SMS messages from ham SMS messages can be very useful.

Dataset

Background

Link: http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

The dataset which we are going to use is the **SMS Spam Collection Dataset** which is hosted on UCI Machine Learning Repository [4, 5, 6]. The dataset has been published on the 22nd of June 2012 [4]. It contains 5,574 English SMS phone messages and each of these messages have already been labeled as either ham (legitimate) or spam [4, 6] as shown in Figure 1, Figure 2, and Figure 3. The main purpose of the SMS Spam Collection Dataset is to assist NLP and computational linguistics research focused on detecting mobile phone SMS text message spams [4].

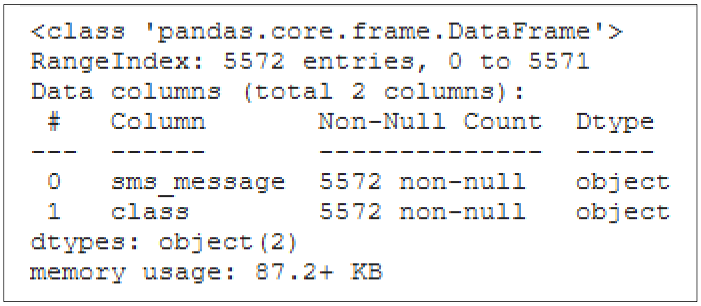


	class	sms_message
0	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 0845281007...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives around here though
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u. U have won the £750 Pound prize. 2 claim is easy, call 087187272008 NOW1! Only 10p per minute. BT-...
5568	ham	Will ü b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other suggestions?
5570	ham	The guy did some bitching but I acted like i'd be interested in buying something else next week and he gave it to us for free
5571	ham	Rofl. Its true to its name

5572 rows × 2 columns

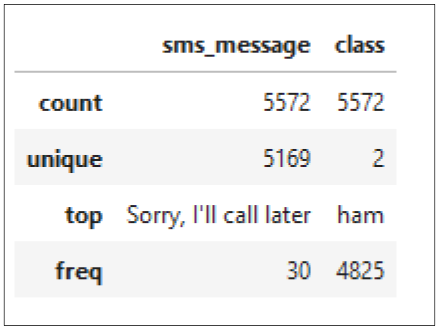
Figure 1: Sample data instances of SMS Spam Collection Dataset

The dataset has been constructed using multiple free resources from the internet. Out of the total of 5,574 messages, 425 SMS messages were extracted manually from the Grumbletext website [7] which is a forum for mobile phone users to discuss SMS spam messages. Further, 3,375 SMS ham messages have been chosen randomly from the NUS SMS Corpus (NSC) [8] which consists of about 10,000 ham SMS messages collected mostly from Singaporean students studying in the Department of Computer Science at the National University of Singapore [4]. Additionally, 450 SMS ham messages have been gathered from the Ph.D. thesis of Dr. Caroline Tagg [4]. Furthermore, 1,002 SMS ham messages and 322 spam messages have been accumulated from SMS Spam Corpus v.0.1 Big [4].



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   sms_message  5572 non-null   object
 1   class        5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

Figure 2: Concise summary of the dataframe



	sms_message	class
count	5572	5572
unique	5169	2
top	Sorry, I'll call later	ham
freq	30	4825

Figure 3: Descriptive statistics of the dataframe

Attributes

The **SMS Spam Collection Dataset** is composed of one text file consisting of only two columns.

1. SMS message text

This is the only input feature of the dataset. It contains the raw text of the mobile phone SMS message. The data type of this attribute is a string.

We have analyzed the text length distribution of each of these SMS messages in terms of the character count and plotted a graph as shown in Figure 4.

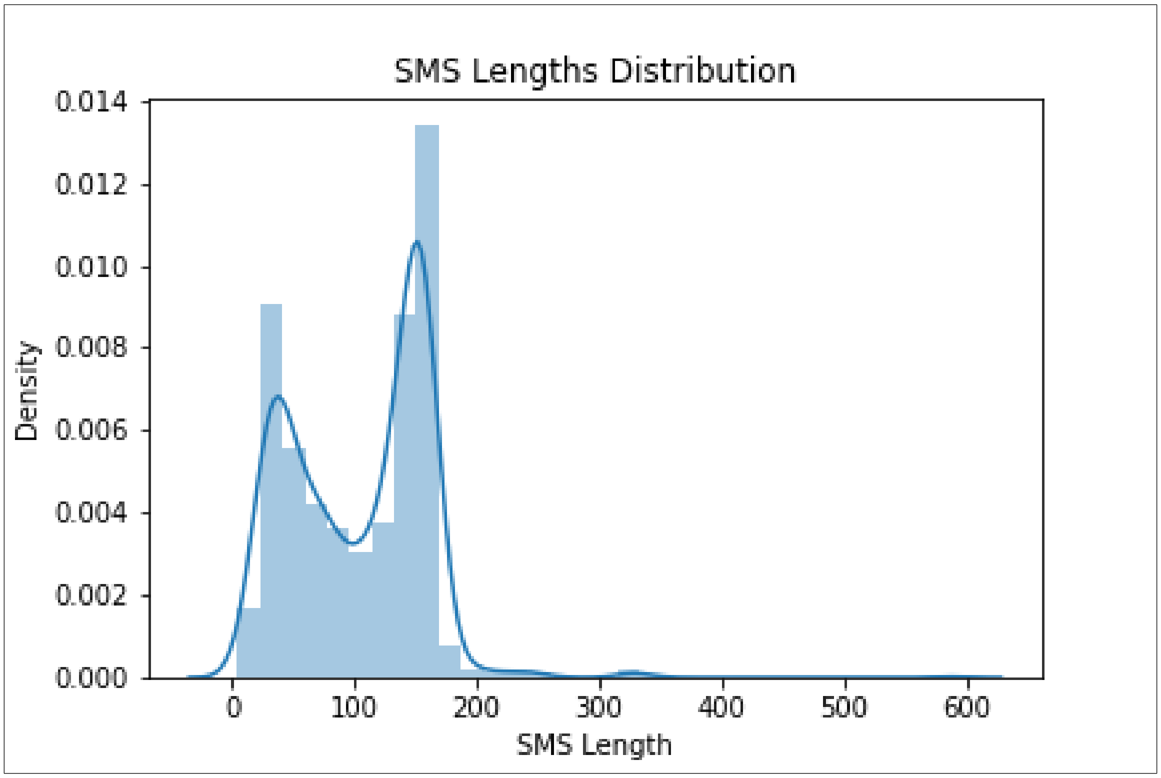


Figure 4: SMS lengths distribution plot

2. Label

This is the target attribute (class label) of the dataset. It has two distinct values: “ham” and “spam”. The data type of this attribute is a string.

We have analyzed the data distribution of the target attribute of the dataset and plotted the distributions in a pie chart.

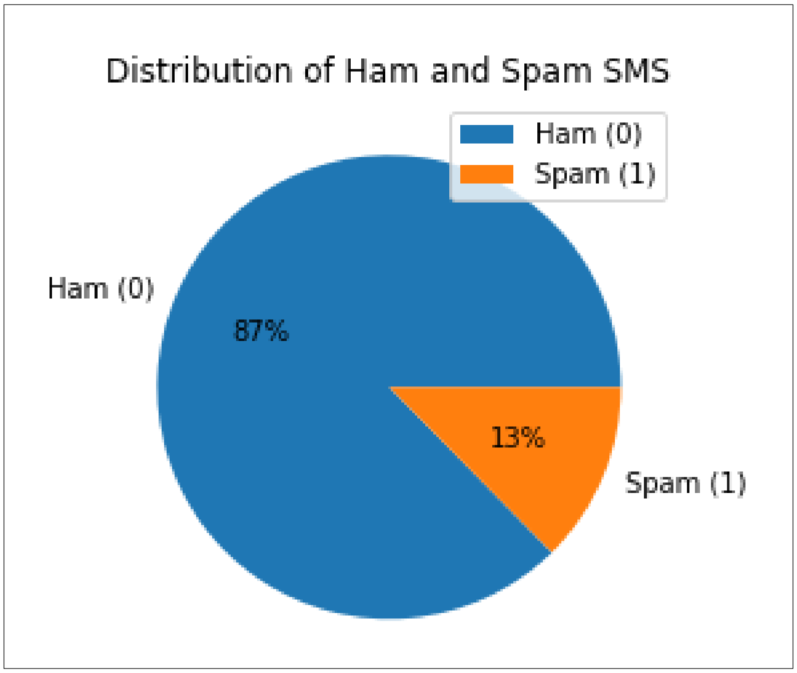
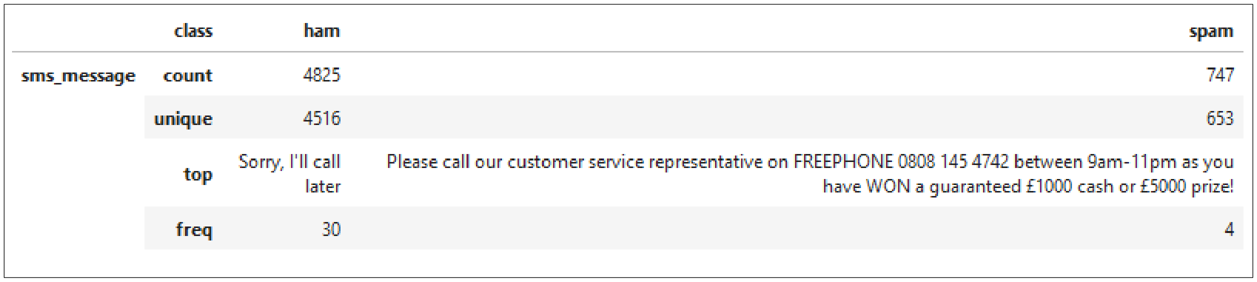


Figure 5: Distribution of ham and spam SMS plot

As shown in Figure 5, about 87% of SMS messages in the dataset are hams and the remaining 13% of SMS messages are spam. Therefore, the majority of the SMS messages in this dataset are labeled as ham as shown in Figure 6.



	class	ham	spam
sms_message	count	4825	747
	unique	4516	653
	top	Sorry, I'll call later	Please call our customer service representative on FREEPHONE 0808 145 4742 between 9am-11pm as you have WON a guaranteed £1000 cash or £5000 prize!
	freq	30	4

Figure 6: Descriptive statistics for each class of the dataframe

We further generated the Word Cloud plots for ham and spam SMS messages separately to visualize the most used words in ham and spam SMS messages as shown in Figure 3 and Figure 4 respectively.

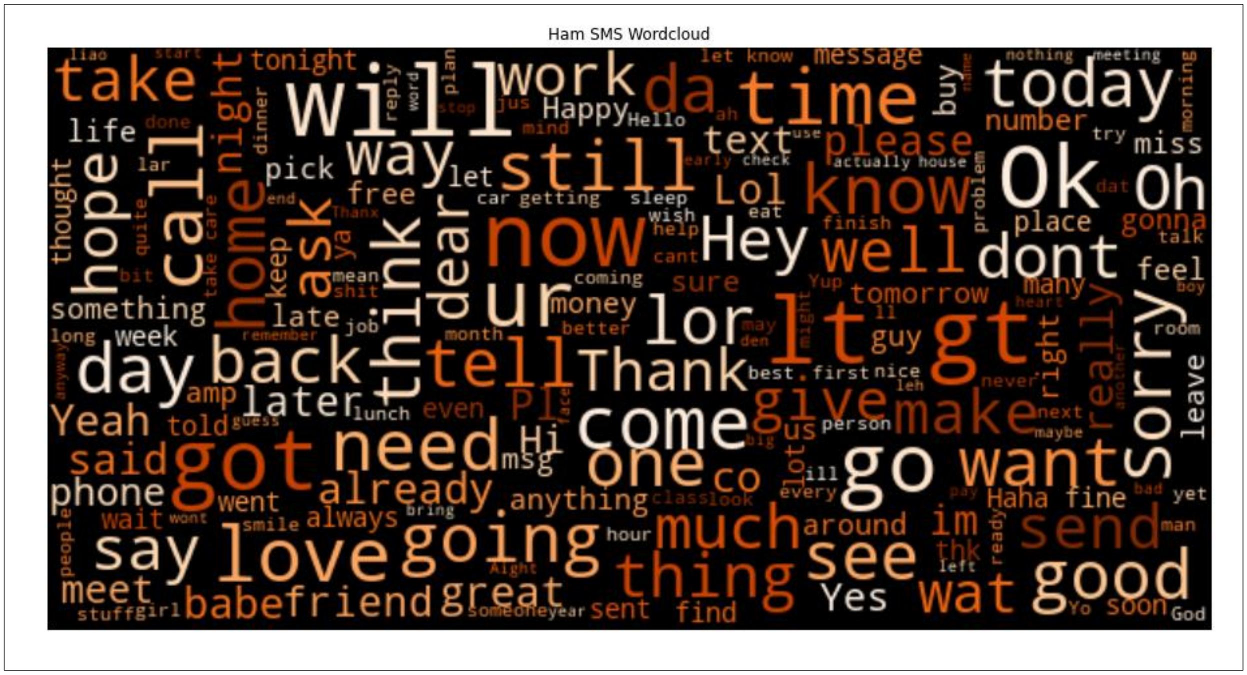


Figure 7: Ham SMS Word Cloud plot

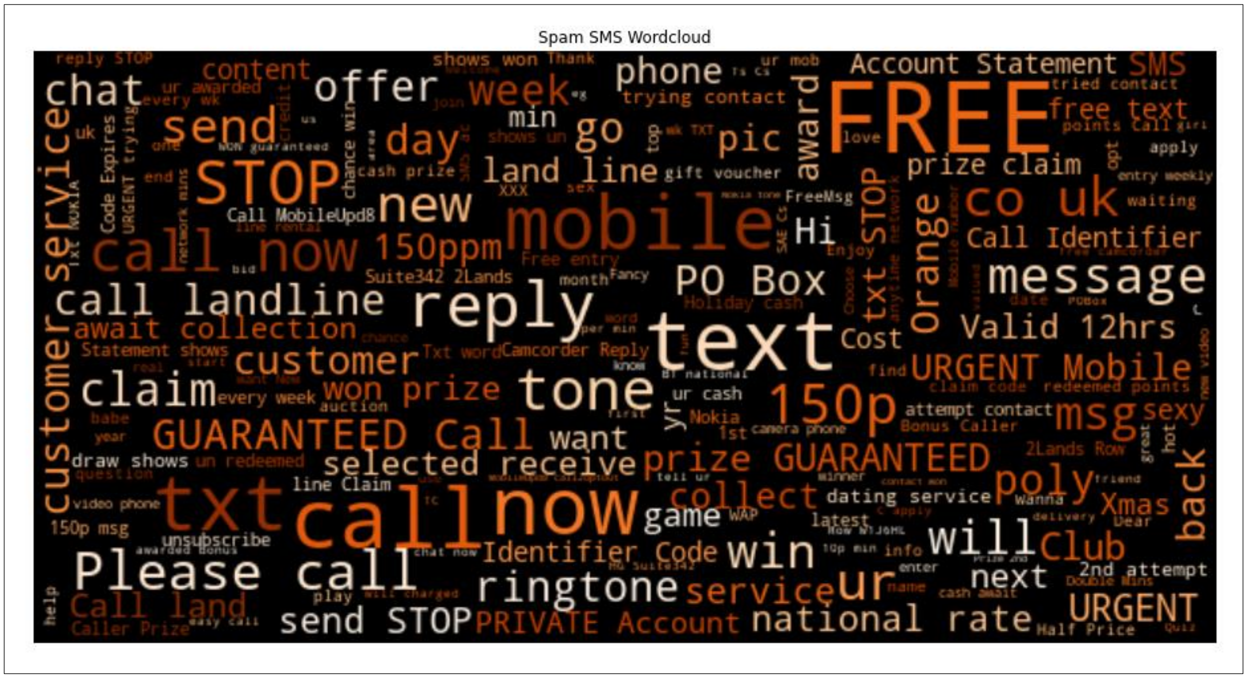


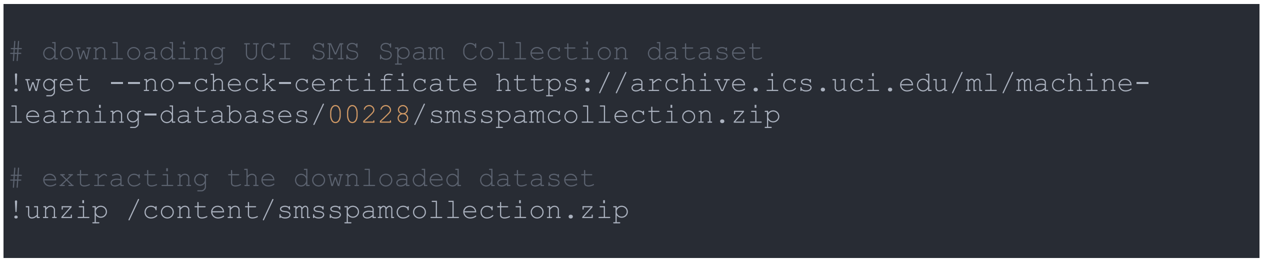
Figure 8: Spam SMS Word Cloud plot

Methodology

Data Cleaning and Preprocessing

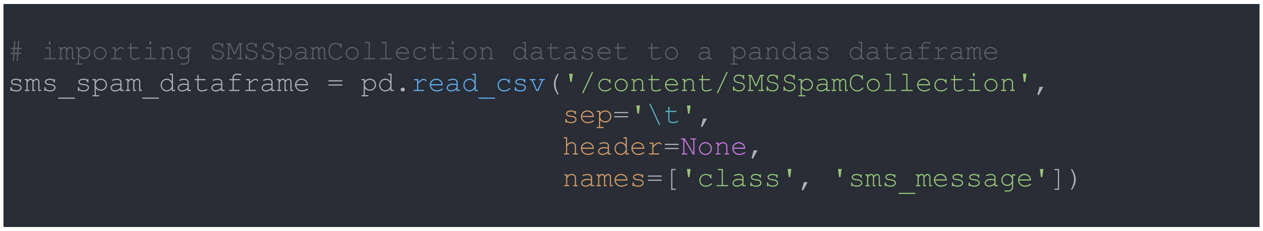
Real-world data sets contain several problems such as missing or null values, data inconsistency, incompleteness, and outliers of the dataset. Therefore, data preprocessing is a mandatory step before feeding the dataset into a deep learning model. Data preprocessing includes transforming row data into the machine-understandable and efficient format. Since the SMS Spam Collection dataset may also contain some of these issues, we had to do preprocessing tasks before using the dataset for model training purposes.

Initially, the dataset was downloaded and extracted.



```
# downloading UCI SMS Spam Collection dataset
!wget --no-check-certificate https://archive.ics.uci.edu/ml/machine-
learning-databases/00228/smsspamcollection.zip

# extracting the downloaded dataset
!unzip /content/smsspamcollection.zip
```

Then the dataset was imported to a Pandas dataframe.



```
# importing SMSSpamCollection dataset to a pandas dataframe
sms_spam_dataframe = pd.read_csv('/content/SMSSpamCollection',
                                 sep='\t',
                                 header=None,
                                 names=['class', 'sms_message'])
```

Then the data was analyzed to check whether there are any missing or null values existed within the dataset. If any null values existed within the dataset, there are usually two ways to deal with them. The first method is to delete the particular rows which contain multiple null values. The second method is to replace the missing values with mean, median, or most frequent values. Although our analysis proved that the SMS Spam Collection dataset did not contain any missing or null values as shown in Figure 9, we implemented the null or missing values removal step programmatically.

```
sms_spam_dataframe =
sms_spam_dataframe[sms_spam_dataframe.notna().all(axis=1)]
```

```
# plotting the heatmap for missing or null values in the dataframe
sns.heatmap(sms_spam_dataframe.isnull(),
            yticklabels=False,
            cbar=False,
            cmap='viridis')
```

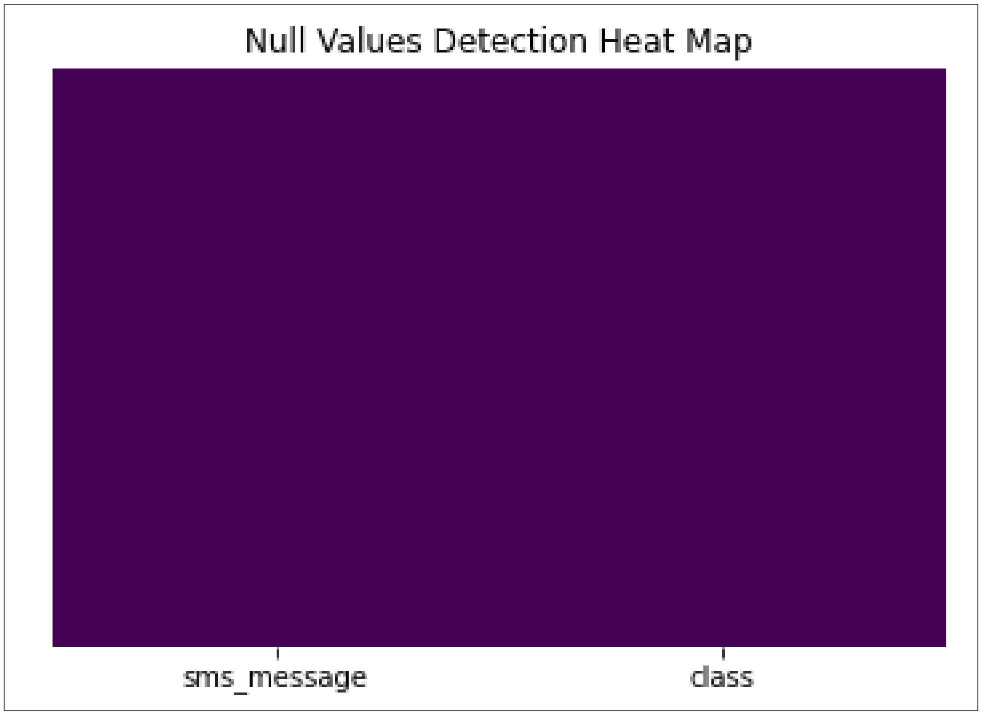
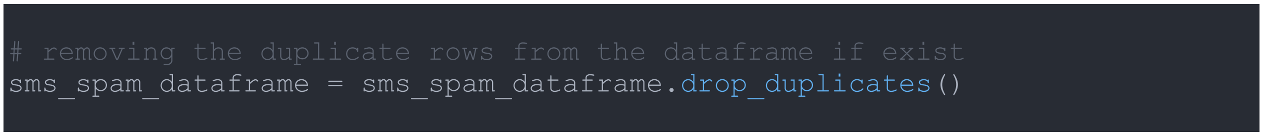


Figure 9: Null values detection heat map

Furthermore, the dataset was checked to identify duplicate rows.

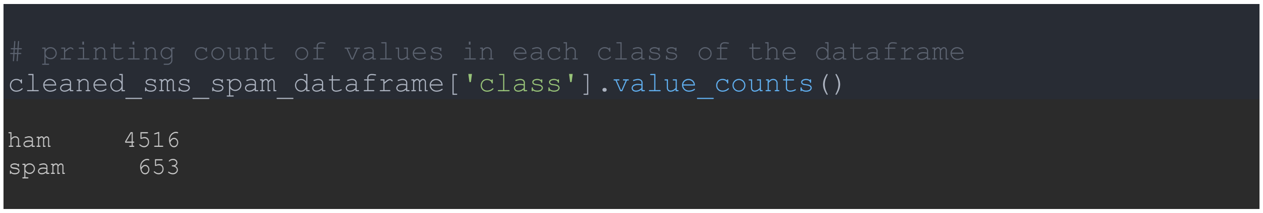
```
# detecting duplicate rows exist in the dataframe before cleaning
duplicated_records = sms_spam_dataframe[sms_spam_dataframe.duplicated()]

# checking the number of duplicate rows exist in the dataframe
# before cleaning
sms_spam_dataframe.duplicated().sum()
```

Since there were 403 duplicates in the data frame, we implemented the duplicate row removal step.

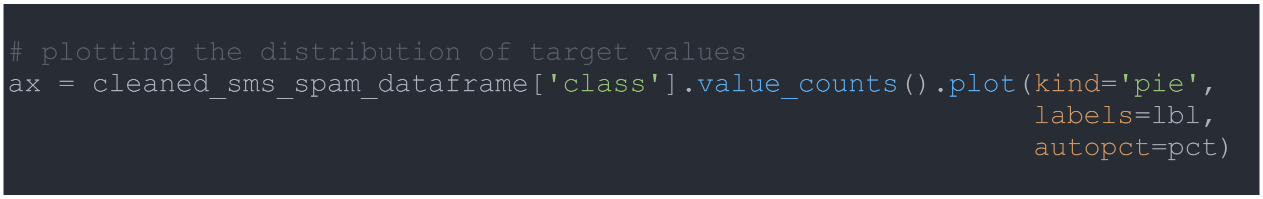


```
# removing the duplicate rows from the dataframe if exist
sms_spam_dataframe = sms_spam_dataframe.drop_duplicates()
```

When we analyzed the data distribution of ham and spam SMS messages, it clearly shows that the data is imbalanced between the two classes as shown in Figure 10.



```
# printing count of values in each class of the dataframe
cleaned_sms_spam_dataframe['class'].value_counts()

ham      4516
spam      653
```



```
# plotting the distribution of target values
ax = cleaned_sms_spam_dataframe['class'].value_counts().plot(kind='pie',
                                                             labels=lbl,
                                                             autopct=pct)
```

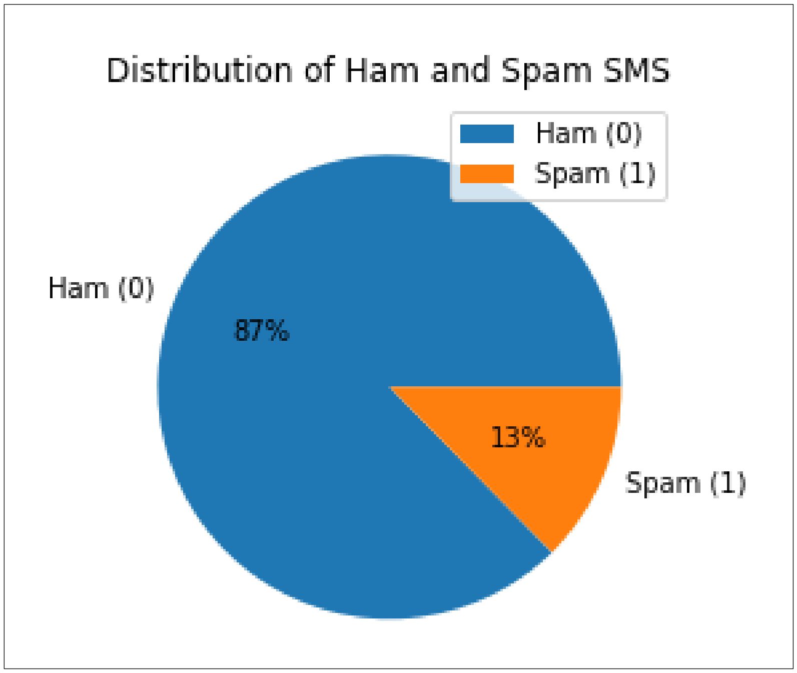
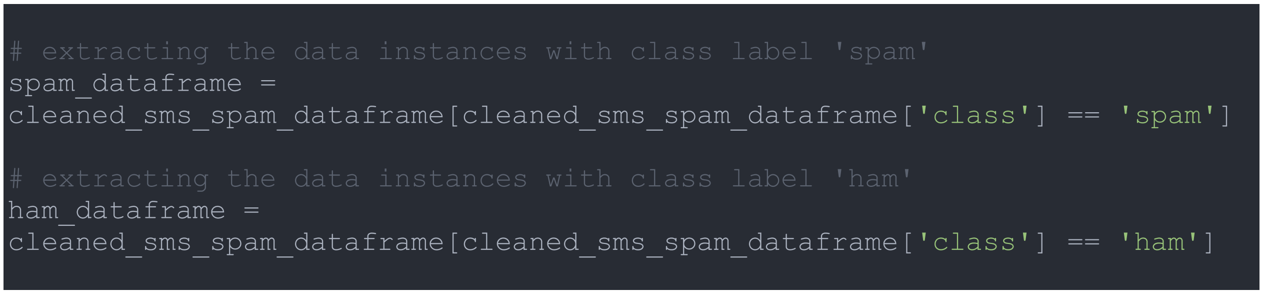
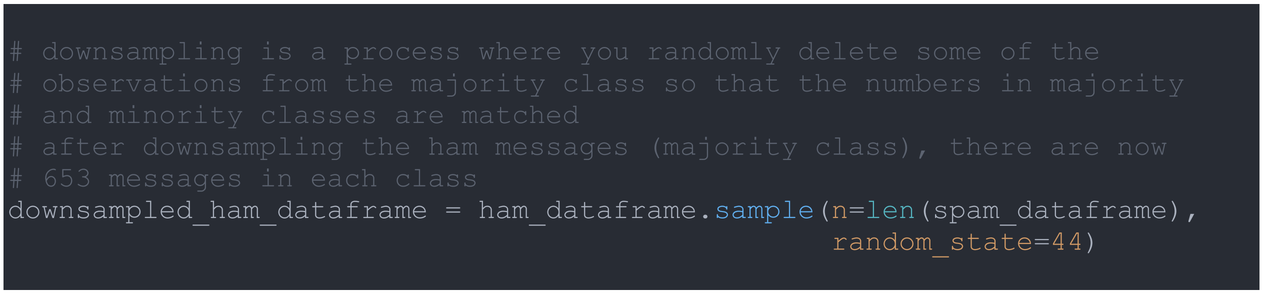


Figure 10: Distribution of ham and spam SMS plot before downsampling

To address this problem of imbalanced data, we decided to apply the downsampling technique which is a process where you randomly delete some of the observations from the majority class (ham) so that the numbers in majority and minority (spam) classes are matched. The dataset is separated into two dataframes based on the class label, and the majority class is downsampled.

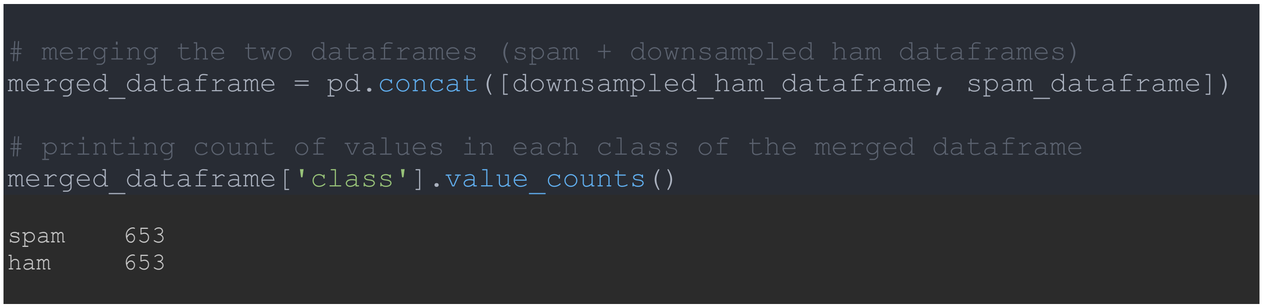


```
# extracting the data instances with class label 'spam'
spam_dataframe =
cleaned_sms_spam_dataframe[cleaned_sms_spam_dataframe['class'] == 'spam']

# extracting the data instances with class label 'ham'
ham_dataframe =
cleaned_sms_spam_dataframe[cleaned_sms_spam_dataframe['class'] == 'ham']
```



```
# downsampling is a process where you randomly delete some of the
# observations from the majority class so that the numbers in majority
# and minority classes are matched
# after downsampling the ham messages (majority class), there are now
# 653 messages in each class
downsampled_ham_dataframe = ham_dataframe.sample(n=len(spam_dataframe),
                                                 random_state=44)
```

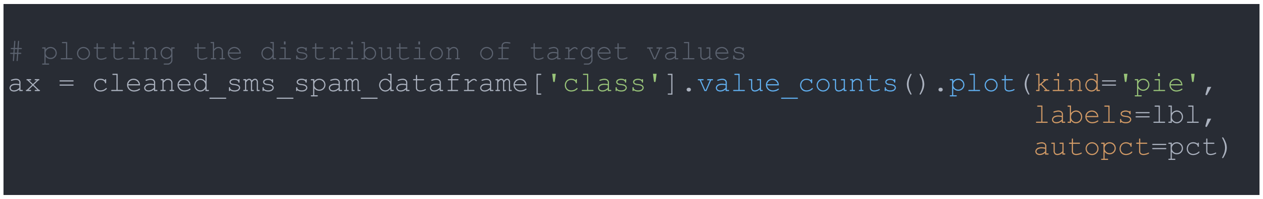
After applying downsampling and merging the two dataframes, there were the same amount of 653 messages for each class.



```
# merging the two dataframes (spam + downsampled ham dataframes)
merged_dataframe = pd.concat([downsampled_ham_dataframe, spam_dataframe])

# printing count of values in each class of the merged dataframe
merged_dataframe['class'].value_counts()
```

```
spam    653
ham     653
```

As shown in Figure 11, now the dataframe is balanced with 50% of data instances for each class.



```
# plotting the distribution of target values
ax = cleaned_sms_spam_dataframe['class'].value_counts().plot(kind='pie',
                                                             labels=lbl,
                                                             autopct=pct)
```

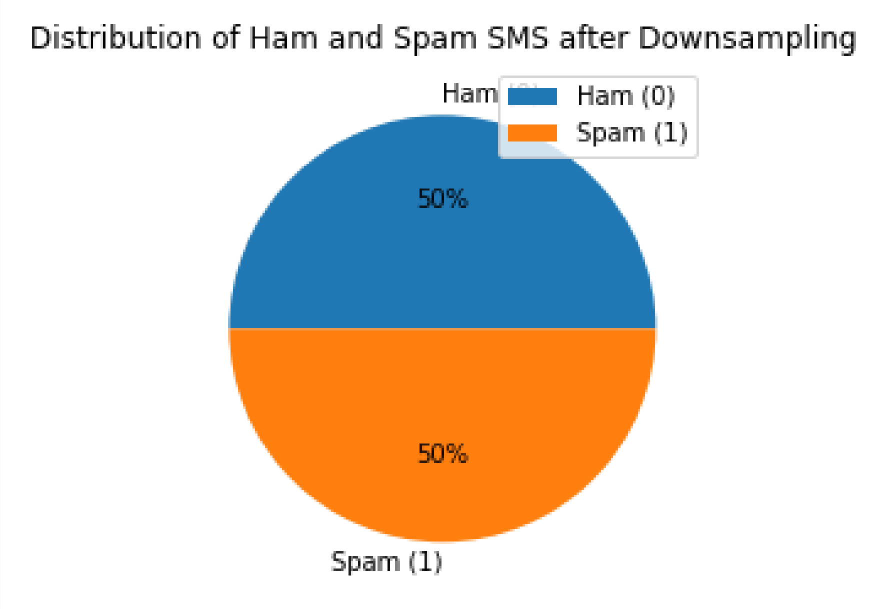


Figure 11: Distribution of ham and spam SMS plot after downsampling

Then we inserted a new column to the dataframe containing the length of the SMS message texts in terms of the number of characters.

```
# inserting a new column called 'length' to the merged dataframe
# the column contains the number of characters of the sms_message text
merged_dataframe['length'] = merged_dataframe['sms_message'].apply(len)
```

Then the text labels “ham” and “spam” were converted to numeric values 0 and 1 respectively as shown in Table 1.

```
# inserting a new column called 'label' to the merged dataframe
# if class is 'ham' label = 0
# if class is 'spam' label = 1
merged_dataframe['label'] = merged_dataframe['class'].map({'ham': 0,
'spam': 1})
```

Table 1: Numeric value for textual class labels

Class	Label
Ham	0
Spam	1

Then the dataset was split into train and test sets. The training set was 80% of the entire dataset while the test set was 20% of the entire dataset. Therefore, out of 1306 total records, 1044 records were selected as training set and 262 records were selected as testing set.

```
# splitting data into random train and test subsets
# train set - 80%, test set - 20%
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.2,
                                                    random_state=443)
```

Table 2: Row count after train test split

Total dataset rows	**1306**
Train dataset rows	1044
Test dataset rows	262

References

[1] A. Ghourabi, M. A. Mahmood, and Q. M. Alzubi, “A Hybrid CNN-LSTM Model for SMS Spam Detection in Arabic and English Messages,” Future Internet, vol. 12, no. 9, p. 156, Sep. 2020, doi: 10.3390/fi12090156.

[2] J. Ma, Y. Zhang, J. Liu, K. Yu, and X. Wang, "Intelligent SMS Spam Filtering Using Topic Model," 2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS), 2016, pp. 380-383, doi: 10.1109/INCoS.2016.47.

[3] M. Popovac, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Convolutional Neural Network Based SMS Spam Detection," 2018 26th Telecommunications Forum (TELFOR), 2018, pp. 1-4, doi: 10.1109/TELFOR.2018.8611916.

[4] Archive.ics.uci.edu. 2012. UCI Machine Learning Repository: SMS Spam Collection Data Set. [online] Available at: http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection.

[5] "SMS Spam Collection Dataset", Kaggle.com, 2016. [online]. Available: https://www.kaggle.com/uciml/sms-spam-collection-dataset.

[6] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, “Contributions to the study of SMS spam filtering,” presented at the 11th ACM symposium, 2011. doi: 10.1145/2034691.2034742.

[7] Grumbletext.co.uk, 2021. [Online]. Available: http://www.grumbletext.co.uk/.

[8] Comp.nus.edu.sg, 2021. [Online]. Available: https://www.comp.nus.edu.sg/.

Appendixes