# LP1 (Data Analytics)
# Mini-Project

## Title : Spam Detection and analysis in SMS

1.Kasturi Patil (41149)
2.Mayuresh Pingale (41151)
3.Piyush Kulkarni (41152)

## 1. Problem Definition

Spam detection and analysis in SMS.

## 2. Abstract

The growth of the mobile phone users has led to a dramatic increase in SMS spam messages. Though in most parts of the world, mobile messaging channel is currently regarded as clean" and trusted, on the contrast the volume of mobile phone spam is dramatically increasing year by year. It is an evolving setback. Nowadays, spam has become serious issue for computer security, because it becomes a main source for disseminating threats, including viruses, worms and phishing attacks. SMS spam filtering is a comparatively recent errand to deal such a problem.

## 3. H/W and S/W requirements

- Operating System : 64-bit Ubuntu 18.04
- Browser : Google Chrome
- Programming Language : Python 3
- Jupyter Notebook Environment : Google Colaboratory

## 4. Introduction

The Short Messaging Service (SMS), commonly referred to as \text mes- saging" is a service for transmitting short length messages of around 160 characters to different devices such as cellular phones, smartphones and PDAs using stan- dardized communications protocols. It is one of the most flourishing phone service engendering millions of dollars in perquisite for mobile operators yearly. Today's estimates signify that billions of SMS's are sent per day.
Spam is the virus infected SMS which results into malfunctioning of

mobiles.Ham is a virus free SMS. Mobile spam is originated from the text message and other communication services by mobile phones.The user cannot identify the spam and segregate legitimate messages mannually.
SMS spam detection is an important task where spam SMS messages are identified and filtered. As more significant numbers of SMS messages are communicated every day, it is challenging for a user to remember and correlate the newer SMS messages received in context to previously received SMS. In our proposed approach,the main aim is to filter the spam and ham SMS using machine learning algorithms. Classification algorithm used is Support Vector Machine(SVM) which has been proven successful in SMS filtering.

## 5. Objective

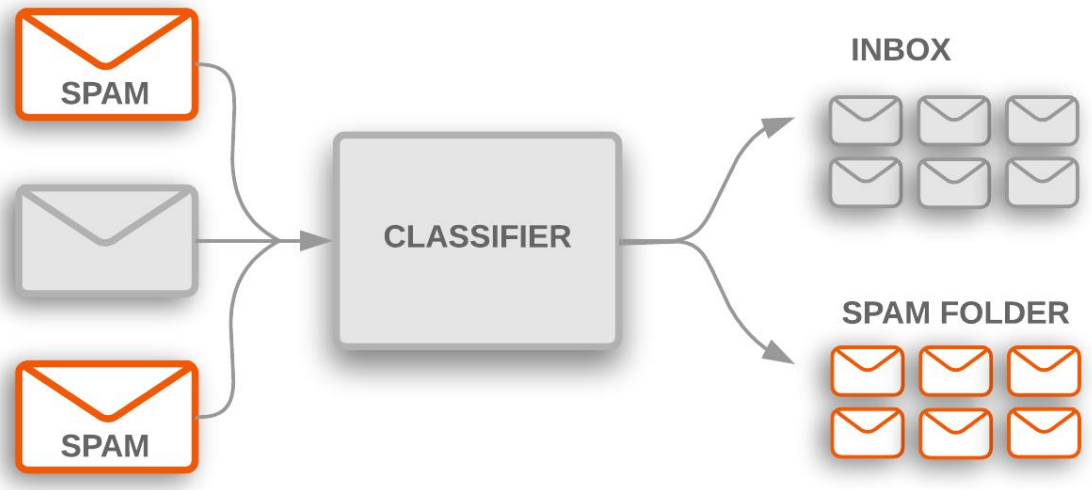
The aim is to distinguish between ham messages and spam messages by making an efficient and sensitive classification model that gives good accuracy with low false positive rate.

## 6. Dataset

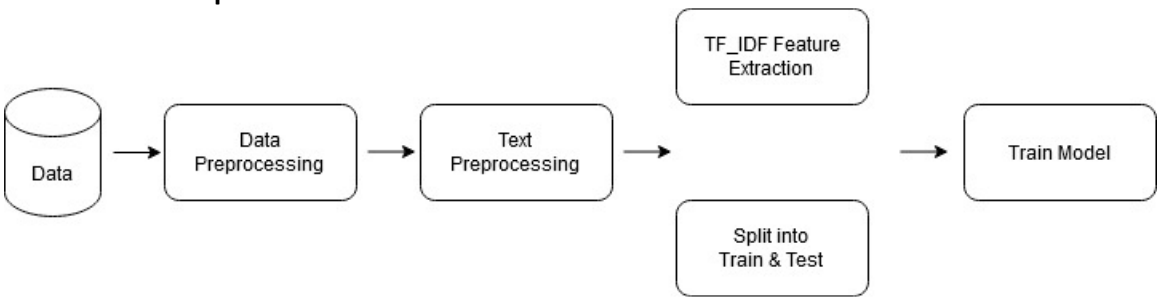
The dataset is a collection of 5567 SMS entries. The messages are tagged according being ham(legitimate) or spam.
The files contain one message per line. Each line is composed by two columns:
Class- contains the label (ham or spam)
Message - contains the raw text

## 7. System Architecture

Extract the text and the target class from the dataset. Extract the features of the test using TF-IDF vectorizer for the Input features.Split the skewed data into shuffled sets using stratified shuffle split in sklearn library. Use standard classifiers to classify the data into spam or ham.




## 8. Model Pipeline




## 9. Analysis of Classifier

1.Accuracy score:98.41%
2.Recall score:99.58%
3.Precision score:98.60%

## 10. Python Libraries and Functions

1.pandas
An open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures.
2.scikit-learn / sklearn
It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.
Functions and Algorithms:
1.train test split function in Sklearn model selection is used for splitting data ar-

rays into two subsets: for training data and for testing data.
2.T_dfVectorizer - Transforms text to feature vectors that can be used as input to estimator.
3.Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.
4.Confusion matrix Compute confusion matrix to evaluate the accuracy of a classification.
5.accuracy score:In multilabel classification, this function computes subset accuracy
3.matplotlib Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

# 11. Future Scope

1.Adding this feature in a dynamic website which supports contact-us typo feature.
2.Show live user inputs for Ham and Spam .

# 12. Output

In [14]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
import string
import matplotlib.pyplot as plt
import re
%matplotlib inline
```

In [15]:
```
dset = pd.read_csv("https://raw.githubusercontent.com/ShubhamPy/Spam-Classifier/master/spam.tsv",sep='\t',names=['Class'
dset.head(8)
```

Out[15]:

| | Class | Message |
|---|---|---|
| 0 | ham | I've been searching for the right words to tha... |
| 1 | spam | Free entry in 2 a wkly comp to win FA Cup fina... |
| 2 | ham | Nah I don't think he goes to usf, he lives aro... |
| 3 | ham | Even my brother is not like to speak with me. ... |
| 4 | ham | I HAVE A DATE ON SUNDAY WITH WILL!!! |
| 5 | ham | As per your request 'Melle Melle (Oru Minnamin... |
| 6 | spam | WINNER!! As a valued network customer you have... |
| 7 | spam | Had your mobile 11 months or more? U R entitle... |

In [16]:
```
dinfo=dset.info()
dinfo
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5567 entries, 0 to 5566
```

```
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Class    5567 non-null   object
 1   Message  5567 non-null   object
dtypes: object(2)
memory usage: 43.6+ KB
```

In [17]:
```
dset.describe()
```

Out[17]:

| | Class | Message |
|---|---|---|
| **count** | 5567 | 5567 |
| **unique** | 2 | 5164 |
| **top** | ham | Sorry, I'll call later |
| **freq** | 4821 | 30 |

In [18]:
```
dset['Length'] = dset['Message'].apply(len)
dset.head(8)
```

Out[18]:

| | Class | Message | Length |
|---|---|---|---|
| **0** | ham | I've been searching for the right words to tha... | 196 |
| **1** | spam | Free entry in 2 a wkly comp to win FA Cup fina... | 155 |
| **2** | ham | Nah I don't think he goes to usf, he lives aro... | 61 |
| **3** | ham | Even my brother is not like to speak with me. ... | 77 |
| **4** | ham | I HAVE A DATE ON SUNDAY WITH WILL!!! | 36 |
| **5** | ham | As per your request 'Melle Melle (Oru Minnamin... | 160 |
| **6** | spam | WINNER!! As a valued network customer you have... | 157 |
| **7** | spam | Had your mobile 11 months or more? U R entitle... | 154 |

In [19]:
```
dset.groupby('Class').count()
```

Out[19]:

| Class | Message | Length |
|---|---|---|
| **ham** | 4821 | 4821 |
| **spam** | 746 | 746 |

In [20]:
```
dset['Length'].describe()
```

Out[20]:
```
count    5567.000000
mean       80.450153
std        59.891023
min         2.000000
25%        36.000000
50%        62.000000
75%       122.000000
max       910.000000
Name: Length, dtype: float64
```

In [21]:
```
dObject = dset['Class'].values
dObject
```

Out[21]:
```
array(['ham', 'spam', 'ham', ..., 'ham', 'ham', 'ham'], dtype=object)
```

In [22]:
```
dset.loc[dset['Class']=="ham","Class"] = 1
```

In [23]:
```
dset.loc[dset['Class']=="spam","Class"] = 0
```

In [24]:
```
dObject2=dset['Class'].values
dObject2
```

Out[24]:
```
array([1, 0, 1, ..., 1, 1, 1], dtype=object)
```

In [25]:
```
dset.head(8)
```

Out[25]:

| | Class | Message | Length |
|---|---|---|---|
| 0 | 1 | I've been searching for the right words to tha... | 196 |
| 1 | 0 | Free entry in 2 a wkly comp to win FA Cup fina... | 155 |
| 2 | 1 | Nah I don't think he goes to usf, he lives aro... | 61 |
| 3 | 1 | Even my brother is not like to speak with me. ... | 77 |
| 4 | 1 | I HAVE A DATE ON SUNDAY WITH WILL!!! | 36 |
| 5 | 1 | As per your request 'Melle Melle (Oru Minnamin... | 160 |
| 6 | 0 | WINNER!! As a valued network customer you have... | 157 |
| 7 | 0 | Had your mobile 11 months or more? U R entitle... | 154 |

In [26]:
```
#clean message from punctuations
def cleanMessage(message):
    nonPunc = [char for char in message if char not in string.punctuation]
    nonPunc = "".join(nonPunc)
    return nonPunc
```

In [27]:
```
dset['Message'] = dset['Message'].apply(cleanMessage)
```

In [28]:
```
dset.head(8)
```

Out[28]:

| | Class | Message | Length |
|---|---|---|---|
| 0 | 1 | Ive been searching for the right words to than... | 196 |
| 1 | 0 | Free entry in 2 a wkly comp to win FA Cup fina... | 155 |
| 2 | 1 | Nah I dont think he goes to usf he lives aroun... | 61 |
| 3 | 1 | Even my brother is not like to speak with me T... | 77 |
| 4 | 1 | I HAVE A DATE ON SUNDAY WITH WILL | 36 |
| 5 | 1 | As per your request Melle Melle Oru Minnaminun... | 160 |
| 6 | 0 | WINNER As a valued network customer you have b... | 157 |
| 7 | 0 | Had your mobile 11 months or more U R entitled... | 154 |

In [29]:
```
from nltk.stem import PorterStemmer
stemmer = PorterStemmer()

def clean_sentences(text):
    text = text.lower()
    text = re.sub(r"[^a-z0-9^,!.\/']", " ", text)
    text = " ".join(text.split())
    text = " ".join(stemmer.stem(word) for word in text.split())
    return text
```

In [30]:
```
x = dset['Message']
y = dset['Class']
```

In [31]:
```
x = x.map(lambda a: clean_sentences(a))
```

In [32]:
```
x
```

Out[32]:
```
0       ive been search for the right word to thank yo...
1       free entri in 2 a wkli comp to win fa cup fina...
2       nah i dont think he goe to usf he live around ...
3       even my brother is not like to speak with me t...
4                       i have a date on sunday with will
                              ...
5562    thi is the 2nd time we have tri 2 contact u u ...
5563                         will b go to esplanad fr home
5564         piti wa in mood for that soani other suggest
5565    the guy did some bitch but i act like id be in...
5566                                rofl it true to it name
Name: Message, Length: 5567, dtype: object
```

In [33]:
```
y
```

Out[33]:
```
0       1
1       0
2       1
3       1
4       1
       ..
5562    0
5563    1
5564    1
5565    1
5566    1
Name: Class, Length: 5567, dtype: object
```

In [34]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,random_state=42)
```

```
In [35]: print(x_train)
```

```
4063    you are be contact by our date servic by someo...
585                          im in a meet call me later at
3412        uhhhhrmm isnt have tb test bad when your sick
5278                         yeah probabl here for a while
4898    free polyphon rington text super to 87131 to g...
                              ...
3772                             ok lor msg me b4 u call
5191    spook up your mob with a halloween collect of ...
5226    i realis you are a busi guy and im tri not to ...
5390       dunno lei shd b drive lor co i go sch 1 hr oni
860                dude ive been see a lotta corvett late
Name: Message, Length: 4175, dtype: object
```

```
In [36]: from sklearn.feature_extraction.text import TfidfVectorizer
```

```
In [37]: vectorizer = TfidfVectorizer(stop_words='english')
```

```
In [38]: x_train = vectorizer.fit_transform(x_train)
```

```
In [39]: x_test = vectorizer.transform(x_test)
```

```
In [40]: from sklearn.svm import LinearSVC
```

```
In [41]: model = LinearSVC(C=1.05, tol=0.5)
```

```
In [42]: print(x_train)
```

```
  (0, 4680)     0.4624572218570745
  (0, 211)      0.4624572218570745
  (0, 3593)     0.31412203882364786
  (0, 4043)     0.2553456131622503
```

```
In [43]: y_train=y_train.astype('int')
         y_test=y_test.astype('int')
         model.fit(x_train,y_train)
```

```
Out[43]: LinearSVC(C=1.05, tol=0.5)
```

```
In [44]: y_test
```

```
Out[44]: 1168    1
         765     1
         465     1
         1117    0
         4930    1
                ..
         668     0
         218     1
         4711    1
         2970    1
         3541    1
         Name: Class, Length: 1392, dtype: int32
```

```
In [45]: from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, f1_score, recall_score
         confusion_matrix(y_test,model.predict(x_test))
```

```
Out[45]: array([[ 164,   17],
                [   5, 1206]], dtype=int64)
```

```
In [46]: accuracy_score(y_test,model.predict(x_test))
```

```
Out[46]: 0.9841954022988506
```

```
In [47]: recall_score(y_test,model.predict(x_test))
```

```
Out[47]: 0.9958711808422791
```

# 13. Conclusion

Thus we successfully implemented Spam and Ham Filter in SMS and analysed the spam messages.

# 14. References

1.https://towardsdatascience.com/the-ultimate-guide-to-sms-spam-or-ham-detector-
aec467aecd85
2.cs229.stanford.edu
3.https://www.kaggle.com/adepvenugopal/detecting-sms-spam-using-machine-learning