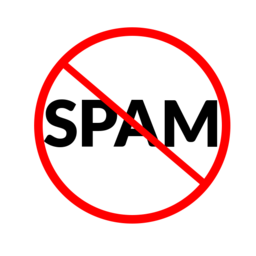
**SPAM / HAM Classification - Natural Language Processing (NLP)**

Author: Balaji Shanmugam



Contents

[INTRODUCTION 2](#_Toc32980810)

[Model Diagram 3](#_Toc32980811)

[Dataset: 3](#_Toc32980812)

[Process Flow Diagram: 4](#_Toc32980813)

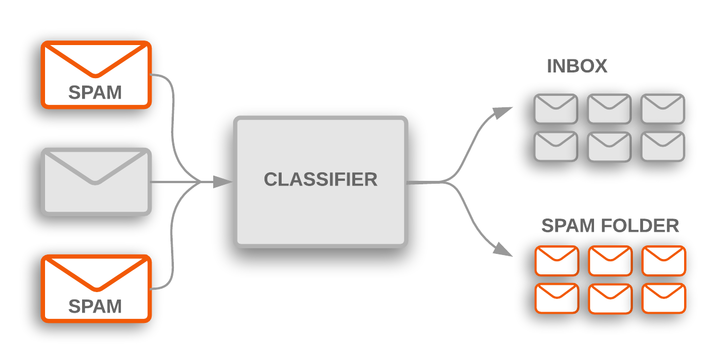
[Tools and Algorithms: 4](#_Toc32980814)

[Technical Details 5](#_Toc32980815)

# INTRODUCTION

We get hundreds of messages from unknown sources and our inbox is filled with unwanted messages. These unwanted messages are called spam and essential messages are called ham mails. We will prepare a model that will categorize messages in mobile devices as spam or ham. In order to achieve this, data from the messages is to be collected first and natural language processing techniques are to be applied on it.

The spam filtering among messages helps the mobile user to have a good visualization of the inbox. Unnecessary messages will be marked as spam so users need not waste their time reading them. In this paper, we propose to classify data in the messages as either spam (unwanted) or ham(wanted) messages.

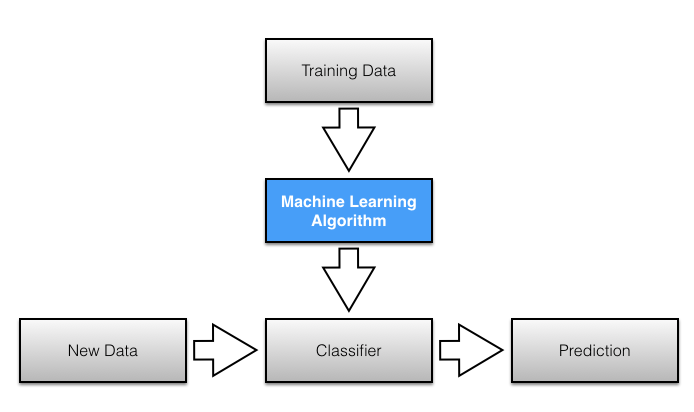


This problem comes under classification problem since here we are going to identify whether a message is spam or ham (which is binary 0 or 1). We use below three algorithms to train our model.

1. Support Vector Machine (SVM)
2. Naive Bayes (NB)
3. Logistic Regression

# Model Diagram

Below flow diagram illustrates how predictive model works.



# Dataset:

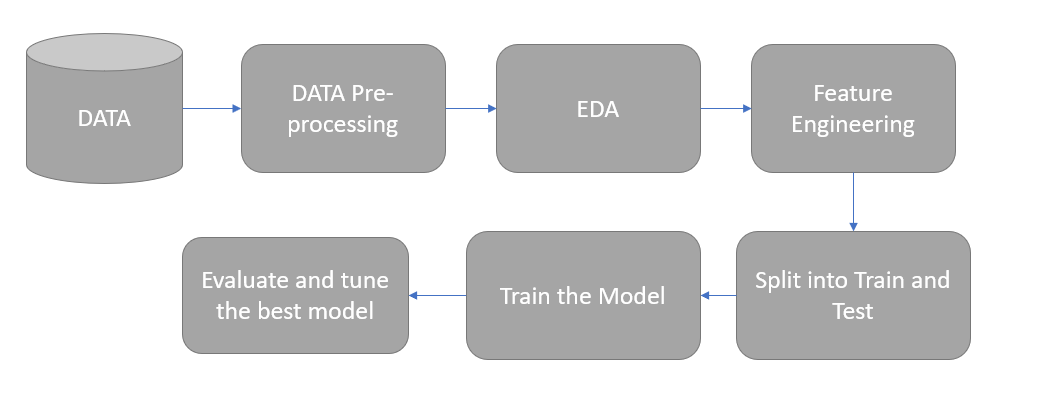
This dataset made available at <https://www.kaggle.com/uciml/sms-spam-collection-dataset>

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 according 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.

# Process Flow Diagram:



1. Data Loading & Preprocessing - Loaded textual data into Panda DataFrames. Anotated lebels. Performed lower casing, punctuation removal, and stop words deletions.
2. EDA – Analyze the data which is manipulated using visualizations(Descriptive analysis,text analysis, stopwords).
3. Feature Engineering - TF-IDF(term frequency - inverse document frequency) encoding & simply word length feature
4. ML Classifiers - Considered Logistic Regression, SVM and Naive Bayes.
5. Train/Test & Cross Validation - Considered 70/30 train/test ratios.
6. Compared the different models and did confusion matrices for the best method. Also carried out parameter tuning using GridSearchCV

# Tools and Algorithms:

Python – Jupyter Notebook

The machine learning algorithms used in the work are described in detail in this section.

**Naïve Bayes (NB)**

Naive Bayes Classifier uses Bayes Theorem, which determines the occurrence probability of an event considering the probability of an occurred event. Linearly separable problems are solved extremely well by Naïve Bayes classifier and for non-linearly separable questions, it performs reasonably good.

Multinomial Naive Bayes classifier uses a multinomial distribution for each one of the features generated on data. This is a particular instance of a Naive Bayes classifier.

**Support Vector Machine (SVM)**

The SVM classifier creates an N-dimensional hyperplane which divides the data into two categories. SVM models are similar to a Neural Network. SVM classifier usually takes the input data and for every input taken it outputs the class to which this input belongs. Two class problems are solved by SVM which is a non-probabilistic binary linear classifier.

**Logistic Regression**

Although many sophisticated statistical models exit,Logistic regression( or logit regression) uses a logistic function to model a binary dependent variable. In regression analysis, it estimates the parameters of a logit model which is in the form of binary regression. In mathematical words a binary logistic model has a dependent variable with two possible outcomes. These outcomes can be labelled as “0” and “1” which usually represent two opposite classes such as pass/fail, win/loose.

# Technical Details

**Attributes in the Dataset:**

* type: Label (ham or spam)
* text: The plain text

**Python Libraries used:**

* Pandas: To process the csv file
* Matplotlib and Seaborn: To visualise the data
* Sklearn: To use the algorithms
* Numpy: To process the data in arrays
* Nltk and spacy: Used for natural language processing
* Counter: used to count hashable objects

**Additional Feature TF–IDF**

Tf–idf stands for "Term Frequency–Inverse Document Frequency" is a numerical statistic used to reflect how important a word is to a document in a collection or corpus of documents.

TFIDF is used as a weighting factor during text search processes and text mining.

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.

**1.Data processing and Statistical summary:**

From our dataset we load the data into dataframe using pandas library and also study the data present in the dataset. Below functions are carried out.

* Read the csv file
* No. of data present in the file. No of columns and its datatypes.

**2.Descriptive analysis:**

We descriptive analysis to check the source data accuracy and feasibility of predicting a useful information and under the words present in the raw text. We perform feature extraction using text data like No. of stopwords,No.of numeric characters.

**3.Manipulation of textual data:**

Pre-processing the source data is always essential in order to improve the performance of our model. Here we clean the data –

* by removing punctuations/ stopwords and stemming words
* Text Analysis - Frequencies of words in the spam and non-spam messages. The words of the messages will be model features. We use the function Counter.
* Converting our clean text into a representation that a machine learning model can understand. Using TFIDF Vectorizer

**4.Model development**

* SVM

Cost and Gamma are the hyper-parameters that decide the performance of an SVM model. There should be a fine balance between Variance and Bias for any ML model.

Kernel: RBF and Sigmoid

* Naïve Bayes
* Logistic Regression – LG1 and LG2

**5.Comparison between models**

* Accuracy -Measure of correctly identified cases
* F1 score – Harmonic mean of precision and recall, gives a better measure of the incorrectly classified cases than accuracy.
* Confusion Matrices - performance measurement for machine learning classification problem

**6.Parameter tuning using GridSearchCV:**

Grid search is an approach to parameter tuning that will methodically build and evaluate a model for each combination of algorithm parameters specified in a grid. Achieve good results with an SVM by first performing parameter tuning methods.

**References:**

<https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72>

<https://data-flair.training/blogs/svm-kernel-functions/>

<https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c>

<https://towardsdatascience.com/logistic-regression-classifier-8583e0c3cf9>

<https://towardsdatascience.com/spam-or-ham-introduction-to-natural-language-processing-part-2-a0093185aebd>

<https://towardsdatascience.com/spam-detection-with-logistic-regression-23e3709e522>

<https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>

<https://www.datacamp.com/community/tutorials/wordcloud-python>