SMS Spam Classification

CMPE 255 – Data Mining

Prof. Dr. Gheorghi Guzun

San Jose State University

San Jose, CA - 95134

Group 07

Kavya Balihallimatta (013819923)

Sahana Ramesh (013832065)

1. Introduction

1.1 Motivation

Text Messaging has greatly increased in popularity over the past years and has become one of the main means of communication. A lot of junk messages get delivered to a mobile phone as text messaging through the Short Message Service (SMS) [1]. These junk messages are called spam messages and it becomes essential to segregate spam messages from the ham messages(non spam messages). These spam messages could be from a scammer trying to steal your personal information.

Scammers send fake text messages to trick you into giving them your personal information such as password, account number, or Social Security number. If they get that information, they could gain access to your email, bank, or other accounts [2]. Or they could sell your information to other scammers. Sometimes, just clicking on the spam messages could lead to fake sites and inturn could victimize innocent people into giving away their personal information without their knowledge and could even lead to loss of money.

Technology is evolving constantly and has numerous applications in almost all fields. Machine learning is one of the cutting edge technologies which provides us with a lot of classification techniques. As part of this project, we explored various machine learning classification models and compared the results to find the best model for SMS spam classification.

1.2 Objective

The objective of the project is to classify SMS as spam or ham and also to provide an evaluation and analysis of various classification models. The classification models we have considered are as follows:

- Naive Bayes
- Decision Tree
- K- Nearest Neighbours (KNN)
- Support Vector Machine (SVM)
- Logistic Regression
- Neural Networks

1.3 Literature / Market Review

D. K. Renuka et al, provided a comprehensive analysis of various classifiers such as MLP, J48 and Naive Bayes. The authors of this paper have implemented the classification methods on UCI dataset gathered from UCI repository that contains 2788 non-spam emails and 1813 spam emails. They have evaluated the models using evaluation metrics such as precision, recall and F-1 score and have presented MLP to be the best performing algorithm for the purpose of classification. [5]

N. Amir Sjarif et al, in their research paper have proposed SMS classification using various machine learning and text mining algorithms such as TF-IDF, Multinomial Naive Bayes, KNN, SVM, Decision Tree and Random Forest. According to [6], TF-IDF along with MNB outperformed the remaining algorithms with precision of 0.98 and f1-score of 0.97. They have inferred that including more features such as SMS length might help classifiers to perform well. [6]

S. Misra et al, in their paper reviewed various start-of-the-art technologies for SMS spam classification. Machine learning algorithms specifically Naive Bayes and SVM are used heavily. As part of feature selection TF- IDF and n-grams were the most utilized techniques. It was also inferred that around 8.23% of android users use anti-spam applications. [8]

2. System Design and Implementation Details

2.1 Algorithms Considered

For SMS spam classification we have implemented the following supervised classification algorithms as our dataset contains labeled records.

2.1.1 Naive Bayes

Naive Bayes is a classification technique which works on the principle of Bayes' theorem. This algorithm assumes that all attributes are independent which means that it looks at each attribute individually irrespective of presence of other attributes. [3]

2.1.2 Decision Tree

Decision Tree is a predictive modelling approach where class prediction is done on the basis of a tree-like structure where leaves represent class labels and nodes represent attributes or combination of attributes that leads to the respective class label. sklearn library's DecisionTreeClassifier algorithm is used to implement this model.

2.1.3 SVM

Support Vector Machine(SVM) constructs a hyperplane or a set of hyperplanes that can be used for classification. SVM performs both linear as well as non-linear classification using kernel trick. SVM is an effective classification for high dimensional dataset.

2.1.4 KNN

K-Nearest neighbor is a classification algorithm that classifies data by computing the proximity of each record against every other record and predicts the class by taking a majority vote of k nearest records. As part of this project, Euclidean distance is used as a proximity measure.

2.1.5 Logistic Regression

Logistic Regression is a predictive algorithm which is based on the concept of probability. Classification is decided based on the value of the probability obtained of whether it is spam or ham.

2.1.6 MLP (Multi-Layer Perceptron)

Multi-layer perceptron (MLP) is also known as a feed forward artificial neural network. It consists of an input layer, output layer and hidden layers. Beyond the input layer, each node acts as a neuron and uses an activation function such as tanh, ReLU, sigmoid, etc. The input to each neuron is the weighted sum of outputs from the previous layer.

2.2 Technologies & Tools Used

Tools & Technologies	**Purpose**
Jupyter Notebook (version 4.7.12)	For running of algorithms
Google Colaboratory	For running of algorithms
Pandas	To read data from the JSON file and for data preprocessing and data manipulation.
sklearn	For implementing classification models
NumPy	For data pre-processing
NLTK	For tokenization and stemming (to remove stopwords)
Wordcloud	For data visualization. It represents text data with the frequency of each text representing the size of the text.
Matplotlib	For data visualization
TfidfVectorizer	For applying tf-idf on textual data

Table 2: Technologies & Tools

2.3 System Design

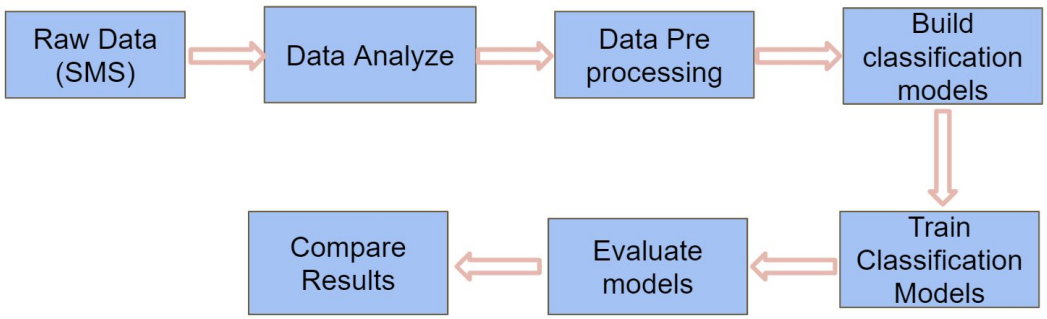


Figure 1: System Design

The above figure shows the system design that we have followed for the implementation of SMS spam classification. The first and foremost step is to collect the raw data i.e, collection of SMS. The collected data is then analyzed in order to get insights and deduce the summary statistics of data. The next step is data preprocessing where the collected and analyzed data is converted to the required format. The preprocessed data is then split into the train and test datasets before feeding them to classification models. We then build various classification models and train them using the train dataset. The models are then evaluated using appropriate evaluation metrics using the test dataset. We then compare the results of models and choose the one that gives the best accuracy.

2.4 Use cases / GUI / screenshots

Data Visualization is a graphical representation of the data and helps us to visualize and analyze the patterns, trends and outliers in data. Data visualization gives us an insight and helps us with an intuitive analysis.

2.4.1. Class Distribution

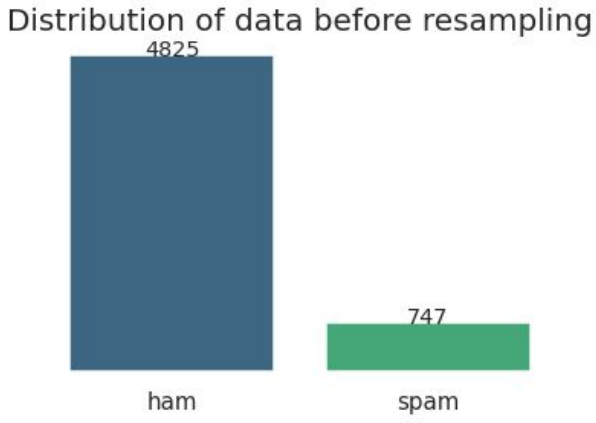


Figure 3: class distribution

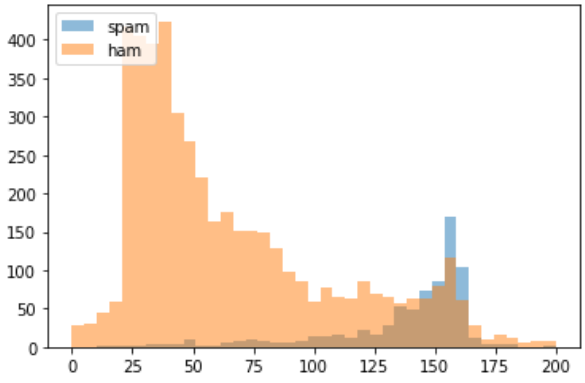


Figure 4: class distribution for word count

2.4.2. Word Cloud

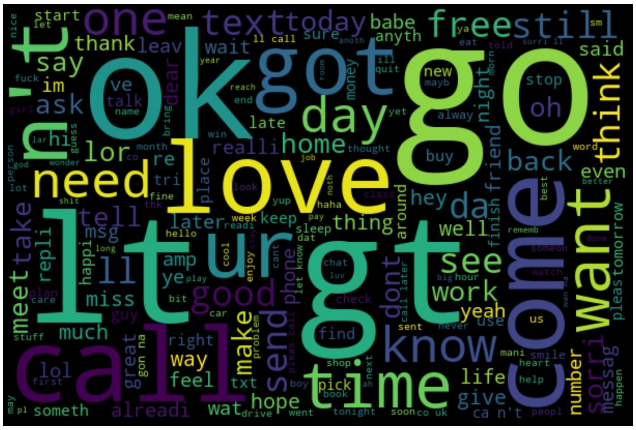


Figure 2: Word Cloud

3. Experiments / Proof of concept evaluation

3.1 Dataset

- **Name**: SMS Spam Collection data set
- **Source**: https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection
- **Size**: 486 KB
- **Number of instances:** The dataset consists of 5572 labeled SMS records.

	label	SMS
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
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 aro...
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will ü b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name

5572 rows × 2 columns

Since our dataset consists of multivariate textual data, many word preprocessing techniques are applied before implementing machine learning classification algorithms.

3.2 Data Preprocessing and Feature Engineering

Step 1: Remove Punctuation

Each text sentence consists of punctuations which only provides grammatical context. Punctuation does not add any mathematical value as the output of the classifier depends on the words in the sentence. So it is essential to make the sentences punctuation free before feeding it to our classifiers.

Step 2: Tokenization

Tokenization is a preprocessing technique which is used to split each textual record into an array of words.

Step 3: Remove Stop words

Stop words are the common words (the, and, or etc) that are present in the textual data. These stop words are removed as they do not add any value to the data.

Natural Language Toolkit (NLTK) library's stopword function is used for this purpose.

Step 4: Stemming

Stemming is a pre-processing technique which is used to convert or reduce each word to its root format. For example, the word sleeping is reduced to its root word sleep.

Natural Language Toolkit (NLTK) library's PorterStemmer function is used for this purpose.

Step 5: Handling Imbalanced Data

If the dataset is imbalanced , then the accuracy of the model is skewed towards majority class. Therefore, it is essential to balance the data before applying classification algorithms. As the SMS data was imbalanced, we have applied Synthetic Minority Over-sampling Technique (SMOTE). SMOTE is an oversampling technique to produce synthetic records from minority class i.e, spam in our case. After applying this technique we get a nearly balanced dataset.

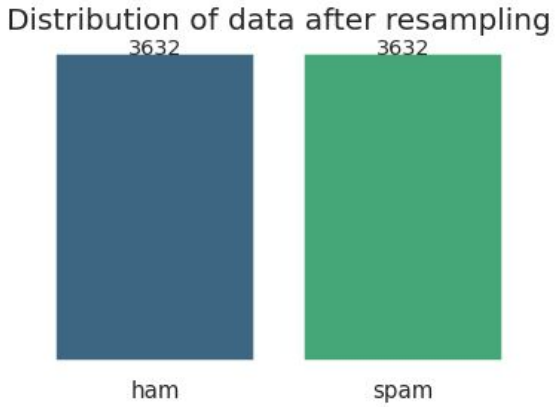


Figure 5: Balanced data distribution

Step 6: TF-IDF

Term Frequency-Inverse Document frequency indicates the importance of each word in a collection of records.

- **Term Frequency(TF)** measures the frequency of a word in each record. It is calculated as follows: *TF = (Number of times a word w appears in a record) / (Total number of words in a record)*

- **Inverse Document Frequency(IDF)** measures how important a word is. It is computed as follows: *IDF = log_e (Total number of records / Number of records with word w in it)*.

sklearn library's TfidfVectorizer function is used for this purpose.

3.3 Methodology

3.1 Train-Test Split

The dataset is divided into train data and test data as 75% and 25% respectively. The classification models are trained using the train data. The trained models are then evaluated using the test data.

3.2 Model Training

We have trained the following classification models by hyper tuning the parameters.

3.2.1 Naive Bayes

sklearn.naive_bayes.MultinomialNB(alpha = 0.2, fit_prior=True, class_prior=None).

3.2.2 Decision Tree

sklearn.tree.DecisionTreeClassifier(criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort='deprecated', ccp_alpha=0.0)

3.2..3 SVM

sklearn.svm.SVC(C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', break_ties=False, random_state=None)

3.2.4 KNN

In order to select the appropriate value of k in KNN, we have evaluated the model using MSE for different values of k. We have observed that MSE value is least when k is set to 3 as shown in the below graph.

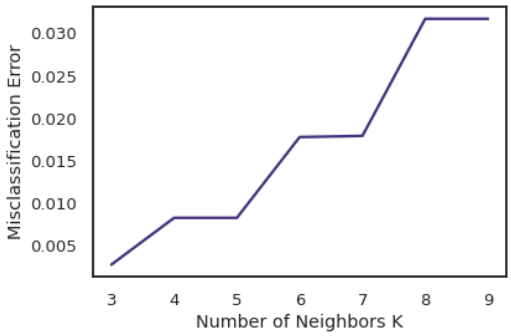


Figure 6: Graph of k vs MSE

The hyperparameters are set as follows:

*sklearn.neighbors.KNeighborsClassifier(n_neighbors=3, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None, **kwargs)*

3.2.5 Logistic Regression

sklearn.linear_model.LogisticRegression(*penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1,class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*)

3.2.6 MLP (Multi-Layer Perceptron)

class sklearn.neural_network.MLPClassifier(*hidden_layer_sizes=(500,), activation='relu', solver='adam',alpha=0.0001,batch_size='auto',learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000)*

3.3 Model Evaluation

Accuracy and F-1 score are used as evaluation metrics

We can visualize the performance of a classification model on a set of test data for which the true values are known by plotting a confusion matrix as shown in the diagram on the right.

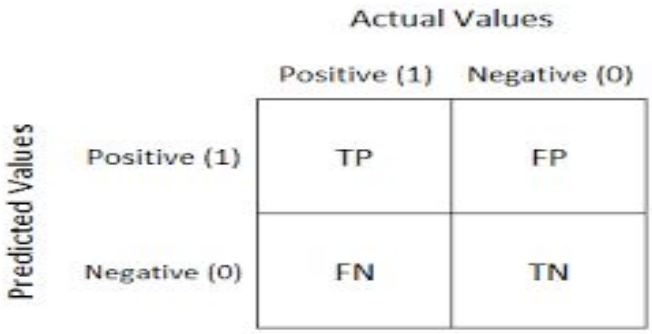


Figure 7: Confusion matrix

Accuracy measures the number of true predictions by total no of predictions as follows:

- **Accuracy = (TP+TN)/(TP+FP+TN+FN)**

where TP = True Positive , TN = True Negative, FP = False Positive, FN = False Negative

Accuracy is not a very well-suited evaluation metric for imbalanced dataset. So we decided to use F-1 Score too as an evaluation metric which is very well suited for imbalanced data.

F1 score is calculated as the weighted average of precision and recall as follows:

- **F1-score = 2pr/p+r**
  where precision (p) = TP/TP+FP and Recall (r) = TP/TP+FN

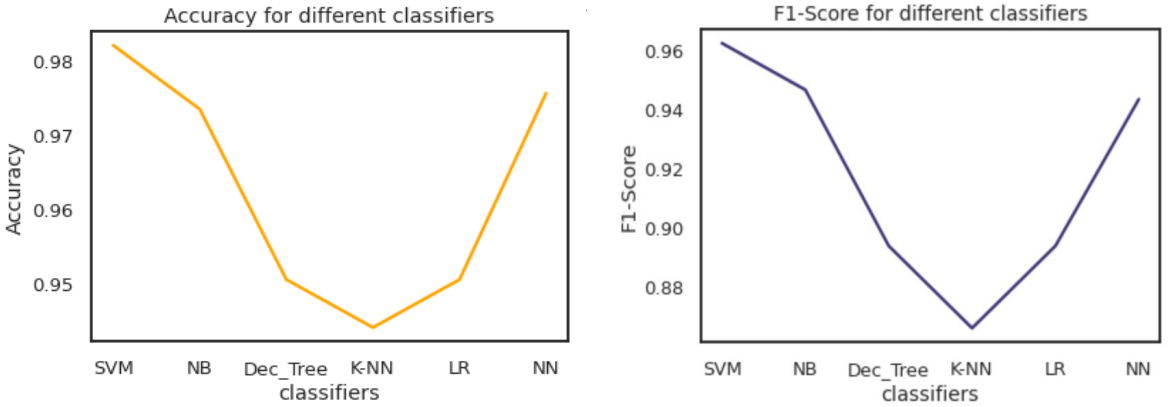
Precision p is the number of correctly classified positive examples divided by the total number of examples that are classified as positive.

Recall r is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.

Model	Accuracy(%)	F-1 Score(%)
Naive Bayes	97.34	94.65
Decision Tree	94.69	89.38
SVM	98.21	96.23
KNN	94.40	86.59
Logistic Regression	94.69	89.38
MLP	97.42	94.34

Table 1: Model Comparison

To compare and analyze the different models that we have used, we even plotted graphs of accuracy and f-1 score against. It is clear from this graph that SVM is the best model with highest accuracy of 98.21% and highest F-1 score of 96.23%.



4. Discussion and Conclusions

4.1 Decisions made

- As the dataset was imbalanced we have decided to use SMOTE technique to balance the dataset.
- As the dataset was textual, we have used TF-IDF vectorizer to find the most important words.

4.2 Conclusion

Various classifier models were implemented for the purpose of spam classification. After evaluating each model, most of the models were tuned to perform fairly well. However, the MLP model was found to perform the best among all the models with the highest accuracy of 98.21% and highest F-1 score of 96.23%. There is still a lot of research going on in the field of sms spam collection and detection as this is a hot topic when it comes to cyber crimes and security.

5. Project Plan / Task Distribution

Task Name	Task Done By
Project Proposal	Team
Data Analysis and Visualization	Team
Data Preprocessing and Feature Engineering	Kavya
Research and analysis of classification algorithms	Sahana
Implementation of Naive Bayes, Decision Tree, SVM	Kavya
Implementation of KNN, Logistic Regression, MLP	Sahana
Report and Presentation	Team

Table 3: Task Distribution

6. References

[1]"What is SMS spam (cell phone spam or short messaging service spam)? - Definition from WhatIs.com", *SearchMobileComputing*, 2020. [Online]. Available: https://searchmobilecomputing.techtarget.com/definition/SMS-spam. [Accessed: 01- May- 2020].

[2]"How to Recognize and Report Spam Text Messages", *Consumer Information*, 2020. [Online]. Available: https://www.consumer.ftc.gov/articles/how-recognize-and-report-spam-text-messages. [Accessed: 01- May- 2020].

[3] 6. R, "Learn Naive Bayes Algorithm | Naive Bayes Classifier Examples", *Analytics Vidhya*, 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/. [Accessed: 01- May- 2020].

[4]"Spam Filtering Using Naive Bayes", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/spam-filtering-using-naive-bayes-98a341224038. [Accessed: 01- May- 2020].4

[5] D. K. Renuka, T. Hamsapriya, M. R. Chakkaravarthi and P. L. Surya, "Spam Classification Based on Supervised Learning Using Machine Learning Techniques," 2011 International Conference on Process Automation, Control and Computing, Coimbatore, 2011, pp. 1-7.

[6] N. Amir Sjarif, N. Mohd Azmi, S. Chuprat, H. Sarkan, Y. Yahya and S. Sam, "SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm", *Procedia Computer Science*, vol. 161, pp. 509-515, 2019. Available: 10.1016/j.procs.2019.11.150.

[7] S. M. Abdulhamid *et al*., "A Review on Mobile SMS Spam Filtering Techniques," in *IEEE Access*, vol. 5, pp. 15650-15666, 2017, doi: 10.1109/ACCESS.2017.2666785.

[8] O. Abayomi-Alli, S. Misra, A. Abayomi-Alli and M. Odusami, "A review of soft techniques for SMS spam classification: Methods, approaches and applications", *Engineering Applications of Artificial Intelligence*, vol. 86, pp. 197-212, 2019. Available: 10.1016/j.engappai.2019.08.024.

Appendix

1. Screenshots of data preprocessing steps

1.1 Removal of Punctuation

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

1.2 Tokenization

	label	SMS	SMS_clean	SMS_tokenized
0	ham	Go until jurong point, crazy.. Available only ...	Go until jurong point crazy Available only in ...	[go, until, jurong, point, crazy, available, o...
1	ham	Ok lar... Joking wif u oni...	Ok lar Joking wif u oni	[ok, lar, joking, wif, u, oni]
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...
3	ham	U dun say so early hor... U c already then say...	U dun say so early hor U c already then say	[u, dun, say, so, early, hor, u, c, already, t...
4	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...

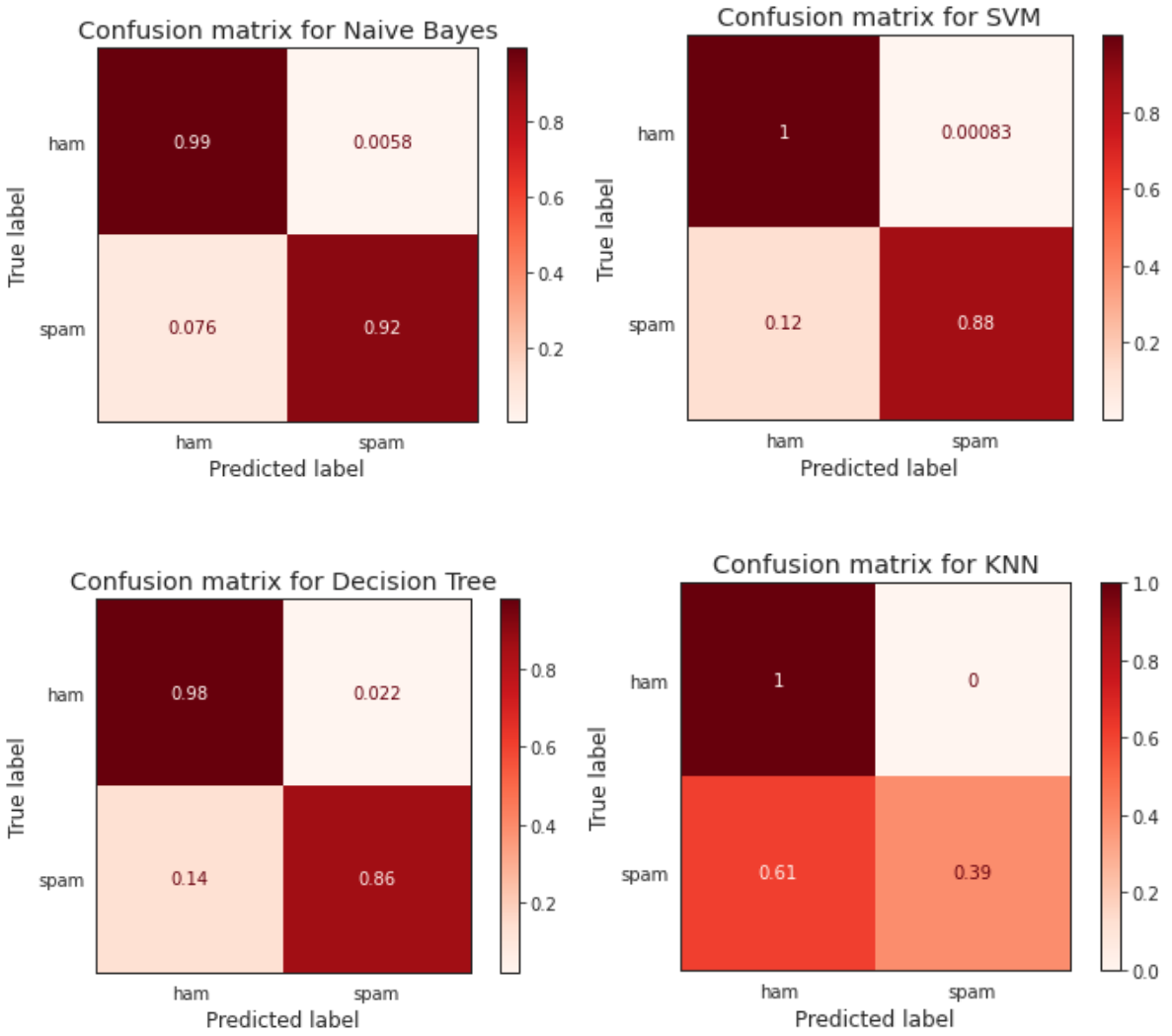
1.3 Remove Stopwords

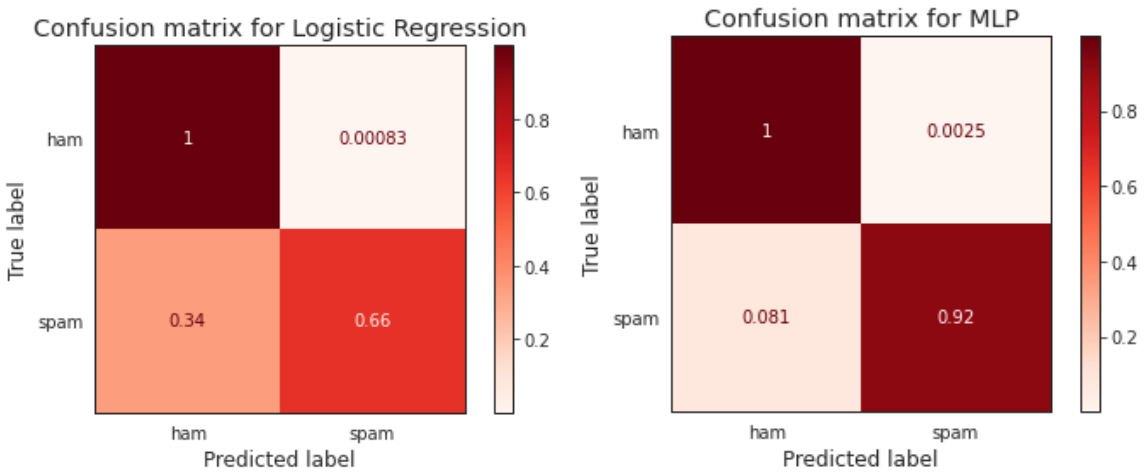
	label	SMS	SMS_clean	SMS_tokenized	length	SMS_nostopwords
0	ham	Go until jurong point, crazy.. Available only ...	Go until jurong point crazy Available only in ...	[go, until, jurong, point, crazy, available, o...	111	[go, jurong, point, crazy, available, bugis, n...
1	ham	Ok lar... Joking wif u oni...	Ok lar Joking wif u oni	[ok, lar, joking, wif, u, oni]	29	[ok, lar, joking, wif, u, oni]
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...	155	[free, entry, 2, wkly, comp, win, fa, cup, fin...
3	ham	U dun say so early hor... U c already then say...	U dun say so early hor U c already then say	[u, dun, say, so, early, hor, u, c, already, t...	49	[u, dun, say, early, hor, u, c, already, say]
4	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...	61	[nah, dont, think, goes, usf, lives, around, t...

1.4 Stemming

	label	SMS	SMS_clean	SMS_tokenized	length	SMS_nostopwords	SMS_stemmed
0	ham	Go until jurong point, crazy.. Available only ...	Go until jurong point crazy Available only in ...	[go, until, jurong, point, crazy, available, o...	111	[go, jurong, point, crazy, available, bugis, n...	[go, jurong, point, crazi, avail, bugi, n, gre...
1	ham	Ok lar... Joking wif u oni...	Ok lar Joking wif u oni	[ok, lar, joking, wif, u, oni]	29	[ok, lar, joking, wif, u, oni]	[ok, lar, joke, wif, u, oni]
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...	155	[free, entry, 2, wkly, comp, win, fa, cup, fin...	[free, entri, 2, wkli, comp, win, fa, cup, fin...
3	ham	U dun say so early hor... U c already then say...	U dun say so early hor U c already then say	[u, dun, say, so, early, hor, u, c, already, t...	49	[u, dun, say, early, hor, u, c, already, say]	[u, dun, say, earli, hor, u, c, alreadi, say]
4	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...	61	[nah, dont, think, goes, usf, lives, around, t...	[nah, dont, think, goe, usf, live, around, tho...

2. Confusion Matrix for various classifiers





Link to Github: