Comparative Analysis of Machine Learning Algorithms for SMS Spam Classifications

Doğan Dinçer Demirci
Department of Computer Engineering
Sabahattin Zaim University
İstanbul,Turkey
demirci.dogan@std.izu.edu.tr

Nurettin Resul Tanyıldızı
Department of Computer Engineering
Sabahattin Zaim University
İstanbul,Turkey
tanyildizi.nurettin@std.izu.edu.tr

Muhammed Farmedi Tengirşek
Department of Computer Engineering
Sabahattin Zaim University
İstanbul,Turkey
tengirsek.muhammed@std.izu.edu.tr

Selim Gülce
Department of Computer Engineering
Sabahattin Zaim University
İstanbul,Turkey
gulce.selim@std.izu.edu.tr

Abstract --- In this work, we aimed to create a distinct and effective preprocessing method and apply the method along with machine learning algorithms to our dataset to achieve a high categorization rate. When we compared our precision rates among other studies we concluded that our preprocessing method was indeed effective. After we have processed the raw data we implemented specific machine learning supervised methods such as Guassian Naive Bayes (GNB), Multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), KNN Classifier (KNN), Random Forest (RF), Bagging Classifier (BC), AdaBoost Classifier (AC), Logistic Regression (LGR), Perceptron (PR), Stochastic Gradient Descent (SGD), Ridge Classifier Cross-Validation (RCCV), Passive Aggressive Classifier (PAC), AdaBoost Classifier (AC), Support Vector Classification (SVC) and Linear Support Vector Classification (LSVC). We visualized the success of each classifying algorithm and compared them in order to figure out which algorithm was more effective.

Keywords --- SMS Spam Detection, Text Preprocessing, Machine Learning, CountVectorizer, TfidfTransformer

I. INTRODUCTION

In today's world where 5.27 billion people own mobile phones [1], SMS messages became a popular tool among scammers and spammers. According to [2] 60% of mobile phone users receive spam messages periodically. In some parts of the world, spam SMS messages cover 20-30% of all the SMS traffic in the region [3]. These spam messages cause frustration, annoyance [4] and may even harm the user by leaking information, obtaining a virus or subscribing to paid subscriptions [5].

In this turmoil of scammers and spammers, mis-categorizing a non-spam SMS as spam may cause harm as much as none-spamming a spam SMS. So, in order to create an SMS filter, we need our classification method to be precise.

In this study, we have aimed for precision on classifying the SMS messages. In order to achieve this desired precision, we came out with a plan that consists of 2 steps. First step was to process the initial data to achieve a more pure and relevant dataset. The 2nd step was to figure which classification method provided the most precision. We implemented nearly a dozen well known classification methods. Multiple of these machine learning algorithms did provide precision but some of them were not as effective as others.

There have been previous studies on this topic. One of these studies *was "Towards SMS Spam Filtering: Results under a New Dataset"* [6]. Two of the researchers who have contributed in this mentioned study were respectable Tiago A. Almeida and respectable José María Gómez Hidalgo who were the creators of the SMS Spam Collection Dataset which we have used in our study. In this study, researchers indicated that Support Vector Machines outperformed other evaluated techniques like Naïve Bayes, Logistic Regression, Linear SVM, K-NN, C4.5 and Random Forest. Hence the study states that Support Vector Machines can be used as a good baseline for future comparisons. [7] This study was inspired upon their work.

Another study regarding this topic is *"Automated SMS Classification and Spam Analysis using Topic Modeling"* [8]. In this study, researchers analyze different machine learning algorithms. Classifiers in this study are categorized into three groups based on their characteristics. Among these methods, researchers state that the Random Forest algorithm has achieved the best performance.

As we compared our results to similar studies, we concluded that our precision rate was higher than most of them. After we compared the methods used on these aforementioned studies and ours, we concluded that our preprocessing method made the difference between the outcomes.

The structure of this paper will be as follows. Section 2 offers details about the machine learning algorithms we have implemented. Section 3 will be brief information about the dataset and the preprocessing method we have used in order to purify it. Section 4 will be about the results and visualization of success rates of each machine learning algorithm we have used and comparison. Section 5 will include the conclusion of this work.

II. METHODOLOGY

1. NAÏVE BAYES:

The Naïve Bayes classifier is a supervised learning algorithm. The Bayesian classifier based on the dependent events possibilities which are going to happen in the future that can be appointed from the same event that happened already. The Naive Bayes method of work is always to calculate the chances of each class and the class having the maximum chance is then chosen as an output. [9]

The types of Naive Bayes Classifiers:

1.1 **Multinomial Naive Bayes** - Multinominal Naive Bayes is a natural language processing learning system that is widely used. The Bayes Theorem contains to Multinomial Naive Bayes, which determines the class of a file, such as a text message or a newspaper article.

1.2 **Bernoulli Naive Bayes** - Bernoulli Naive Bayes is used for discrete data which features are just binary form.

1.3 **Gaussian Naive Bayes** - When handling continuous data, Gaussian Naive Bayes is used. Gaussian Naive Bayes is a sort of Naive Bayes and It is utilizes Gaussian normal distribution. [10]

2. K- NEAREST NEIGHBOR

KNN is a lazy algorithm which means it tries to only memorize the process it doesn't learn by itself. It doesn't take its own decisions. K-NN algorithm classifies new data by measuring Euclidean or Manhattan distance between other classes. Data is classified as the same class as its closest neighbour. To prevent misclassification a value of N is determsined This N value is the number of neighbors classes. Algorithm calculates the distance of test data to the N neighbors and identifies the data depending on the calculated distance or majority. [11]

3. RANDOM FOREST CLASSIFIER

Random Forest Classifier consists of decision trees that are of different shapes and sizes. These combined trees are defined as a forest. In random forest, every tree is contingent on a random vector which has the same distribution for all the participants in the forest. For Random Forest Classifier, generalization error is based on the strength of each tree and the relevance between them. Number of trees in this algorithm is critical. In the case of trees being short on number overfitting may occure and negatively affect the results. [12]

4. BAGGING

Bagging classifier is another ensemble classifier that fits base classifiers each on random subsets of the original data sets and then combines their individual predictions by voting or by averaging to form a final prediction. Bagging is a mixture of bootstrapping and aggregating. Bootstrapping helps to lessen both the variance of the classifier and overfitting by just resampling the data from the training data with the same cardinality as in the original data set.

5. ADABOOST CLASSIFIER

Adaboost Classifier Works by fitting weak learner models (models thet are so terrible at guessing that they are performing slightly better than random guessing.) to repeatedly modified forms of the data. After that, all predictions are combined by a weighted majority vote to finalize the prediction. Modification of the data at for each iteration consist of applying weights to every sample itself. After every successful iteration, weights are modified and the algorithm is applied again to this data thats weight have been modified. [13]

6. LOGISTIC REGRESSION

Logistic Regression is one of the well known and used statistical methods that are used in examining data. Model itself is rather a simple categorisation as it classifies the data as ham or spam, good or bad, ham or no longer ham etc. Simply put, this classification method is used in cases when there are only two possible outcomes. So the data can only be coded as 1 or 0. Main goal of logistic regression is finding the most optimal fitting model. By optimising the fitting model, overfitting is reduced. [14]

7. PERCEPTRON

Perceptron is a supervised learning algorithm which is also binary classifier. Its contents are called neurons and neurons take a row of data as input and predicts a class. First step of this algorithm is multiplying the inputs with their weights and adding all of them with eachother in order to achieve weighted sum. After that, weighted sum is applied to an activation function. This function is used for mapping the inputs between needed values like (0,1) or (-1,1). [15]

8. SGD CLASSIFIER

SGD is a linear classifier that calculates the minimums of the cost function by calculating the gradient at each iteration and updating the model at a decreasing rate.
SGD is widely popular among large and spooradic machine learning problems that emerge in text classification. Although being efficient and easy to implement there are some downsides such as requiring some hyper parameters and being sensitive in feature scaling. [16]

9. RIDGE CLASSIFIER

Ridge Classifier is based on Ridge Regression. This method is used for regularizing the linear regression results to be more stable. To put it simply, Ridge Regression is a regularized linear regression. Regularization means retaining the parameters usual or regularized. There are different regularization methods that use different techniques. Ridge classifier is one of these regularization methods. [17]

10. PASSIVE AGGRESSIVE CLASSIFIER

Passive-aggressive classification is an incremental learning algorithm which is very simple to implement, since it has a closed-form update rule.This method incrementally trains the model while permitting for parameters to be altered when needed. In the case of an update not being able to chage the steadiness, the update is dropped. The core concept is that the classifier adjusts its weight vector for each misclassified training sample it receives, trying to correct it. [18]

III. MATERIALS AND METHODS

A. Dataset:

The SMS Spam Collection dataset is a public dataset created by *Tiago A. Almeida* and *José María Gómez Hidalgo* which contains different SMS samples from various sources. The dataset is a text file where each line has the correct label of ham or spam followed by the message itself. The dataset has 4,827 ham messages and 747 spam messages, a total of 5,574 short messages. [19] Basic statistics of the dataset are shown in Table 1 and there are some example data provided in Table 2.

TABLE 1: Basic statistics of the dataset

Msg	Amount	%
Hams	4,827	86.60
Spams	747	13.40
Total	5,574	100.00

TABLE 2: Example messages from the dataset

ham	I'm leaving my house now...
ham	Hello, my love. What are you doing? Did you get to that interview today? Are you you happy? Are you being a good boy? Do you think of me?Are you missing me ?
spam	Customer service annoncement. You have a New Years delivery waiting for you. Please call 07046744435 now to arrange delivery
spam	You are a winner U have been specially selected 2 receive £1000 cash or a 4* holiday (flights inc) speak to a live operator 2 claim 0871277810810
ham	Keep yourself safe for me because I need you and I miss you already and I envy everyone that see's you in real life

B: Text Preprocessing

Text processing is one of the key elements to achieve a high success classification rate. Therefore, the following steps were applied on the SMS messages to get the best classification result.

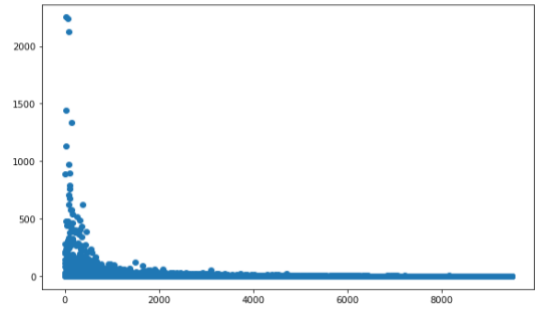

Figure 1: The distribution of processed words

First and foremost, whitespaces have been used to separate the messages in the dataset into words. The messages were then reformed by eliminating all special characters from their phrases, ensuring that there were no words that were actually identical but perceived as different due to these special characters. Following that, the remaining non-alphabetic words were omitted from the messages because they were infrequent and had no impact on the classification result. The non-alphabetical words were not removed at first because the goal was to save as many words as possible for classification. Figure 1 shows the distribution of processed words after these steps have been applied, while Figure 2 shows a word cloud of ham and spam messages with the most repetitive 100 words.

Figure 2: Word Cloud of Ham and Spam Messages

Text data must be transformed into integers or floating-point numbers before being used as an input in machine learning algorithms in order to be used for classification modeling. That process is known as vectorization. [20] Scikit-learn's CountVectorizer is a well-known vectorization model which converts text data to a vector of token counts. In this study, the dataset was converted to vectorized form using Scikit-learn's CountVectorizer model.

Following the vectorization of text data, there is one more important step to improve classification success rates. The raw token frequencies of a text are converted to their normalized tf-idf form in this process. The aim of this move is to reduce the impact of tokens that appear frequently in a corpus and thus are empirically less informative than features that appear in a small percentage of the corpus. [21] Before applying any machine-learning algorithms, the vectorized dataset was transformed into a tf-idf representation using Scikit-learn TfidfTransformer model.

IV. EXPERIMENTAL RESULTS

In this study, we aimed to identify which machine learning classification algorithm performs best on our dataset. So for that purpose, we have implemented different classification methods to our dataset and compared them in order to determine which method had the best ham-spam classification performance. Before we implemented and compared these methods, we transformed our data to vectorial structures. In order to accomplish that, CountVectorizer and TfidfTransformer models were used. Afterwards we split our dataset as train and test by a ratio of 70% (Train) - %30 (Test). We fed this test data to the 13 aforementioned methods and evaluated each of their performance by their precision, recall, accuracy and F1 score values as shown in Table 3. The comparison of the methods based on these parameters is shown in Figure 5. We will be explaining these parameters later in this section.

TABLE 3: Measures

	Name	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
6	SGDClassifier	0.9992	0.9821	0.977477	0.896694	0.935345
5	RidgeClassifierCV	1.0000	0.9815	1.000000	0.871901	0.931567
14	LinearSVC	0.9995	0.9809	0.981651	0.884298	0.930435
4	PassiveAggressiveClassifier	1.0000	0.9773	0.951327	0.888430	0.918803
13	SVC	0.9974	0.9761	0.995098	0.838843	0.910314
7	Perceptron	1.0000	0.9743	0.916318	0.904959	0.910603
0	RandomForestClassifier	1.0000	0.9659	1.000000	0.764463	0.866511
8	BernoulliNB	0.9856	0.9653	0.994624	0.764463	0.864486
1	AdaBoostClassifier	0.9785	0.9600	0.958115	0.756198	0.845266
3	LogisticRegression	0.9669	0.9588	0.983240	0.727273	0.836105
2	BaggingClassifier	0.9941	0.9540	0.945946	0.723140	0.819672
10	MultinomialNB	0.9644	0.9456	1.000000	0.623967	0.768448
11	KNeighborsClassifier	1.0000	0.9414	1.000000	0.595041	0.746114
12	KNeighborsClassifier	0.9464	0.9145	1.000000	0.409091	0.580645
9	GaussianNB	0.9582	0.9085	0.631268	0.884298	0.736661

When evaluating a classification method's effectiveness, accuracy is a critical factor to consider. It can point out whether a model has been properly trained and how well it performs overall on the data it has been provided. However, we can't judge a method's applicability solely on the basis of its precision. Aside from accuracy, there are a variety of other factors to consider. We also need to know the precision, recall, and F1 score of the methods in order to improve our analysis.

Simply put, value of accuracy is the ratio of correctly estimated data in the model to the total of our dataset. On the other hand, precision tells us about the rate of how much data which our methods deemed positive was actually truly positive. Value of precision is extremely cardinal when the cost of a possible false positive is high.

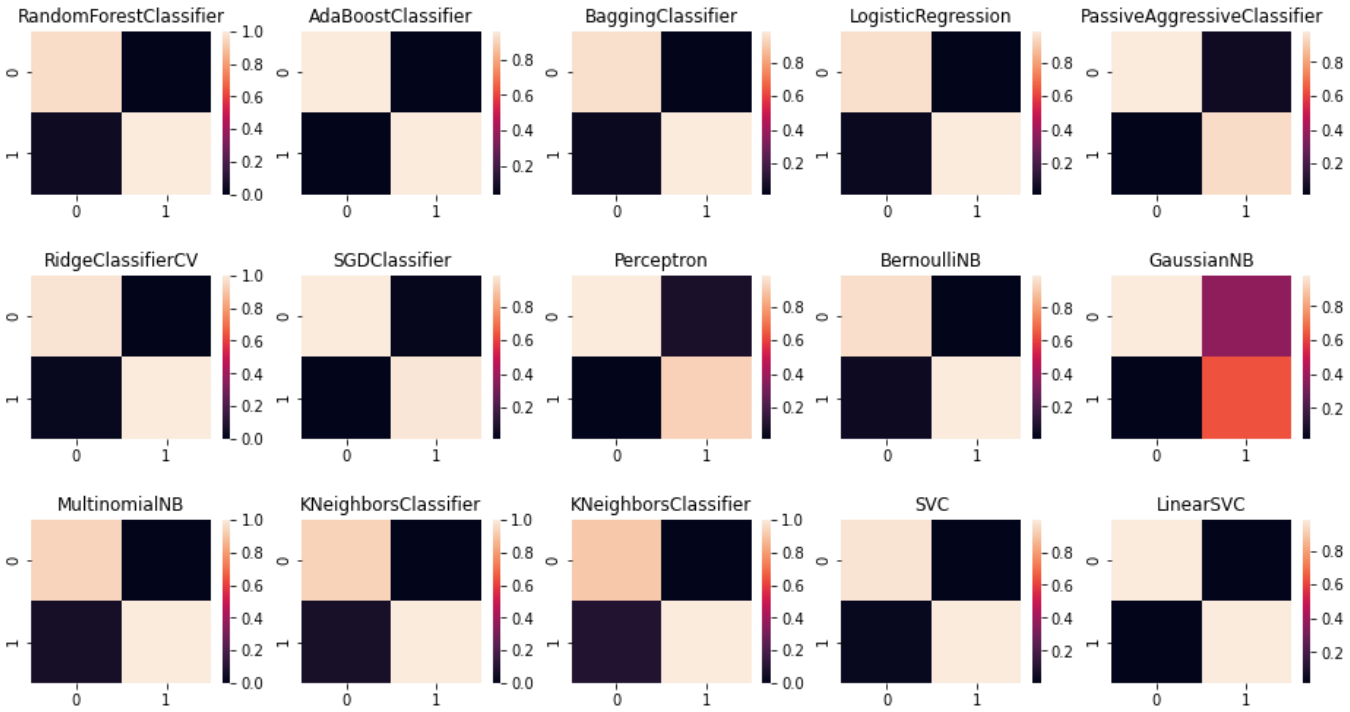

Figure 4: Confusion Matrices of Machine Learning Algorithms

In contrast to precision, recall is a measurement that we benefit in cases where the cost of false negative is high. Recall is a measurement which tells us about the number of transactions that we need to predict as positive. The more this value is, the better it is. The harmonic mean of precision and recall is used to calculate the F1 score. Extreme cases should never be overlooked, so harmonic-mean rather than average is being used. [22]

When we used the above parameters to evaluate our classification algorithms, we found that all of them had a train accurate rate of more than 95% and test accuracy rate of more than 90% as seen in Figure 5. SGD, RCCV, LSVC, PAC, SVC and PR even had a test accuracy of more than 97%. The best test accuracy rate was achieved by SGD which has classified the data with an astounding 98.21% test accuracy. GNB had the lowest test accuracy rate, with a score of nearly 90%, due to its misclassification of spam messages.

When we compared the algorithms according to their success of guessing ham messages and spam messages, we discovered that RF, BC, LR, BC, MNB, and KNN were slightly better at guessing spam messages, while PAC, PR, and GNB did a better job guessing ham messages. In general, all but GNB were successful at guessing spam messages. As seen on the Confusion Matrices in Figure 4, even though GNB performed a fine job on predicting ham samples, it was struggling predicting the spam samples.

When we examined the algorithms according to their F1 Scores, we saw that the six successful algorithms at accuracy rate mentioned above have a score of 91%, and the highest score is again belonging to the SGD Classifier with a rate of 93.35%. In general, there is a linear relationship between test accuracy and F1 score. However, it is observed that the KNN with k value of 3, which performs better than the GNB in test accuracy, has a worse result than the GNB, which shows a success rate of 73% with a 58% success rate compared to the F1 Score. This situation stems from the fact that the recall value of KNN algorithms in general and especially the KNN with k value of 3 is very low with a rate of 40%.

For precision rate, we can see that the GNB has the lowest precision value, with a rate of 63%. The inadequacy of the GNB algorithm to correctly classify spam SMS is the cause of this loss. In addition, the RCCV, RF MNB, KNN with k value of 3 and KNN with k value of 1 algorithms all have a precision rate of 100%. This demonstrates that all of the values that these algorithms consider to be positive are in fact true positives.

V. Conclusion

To summarize, the SGD outperforms other algorithms in terms of test accuracy as well as the F1 score, which is a harmonic average of precision and recall values. When they are compared according to their precision and recall rates, it is clear that although it does not give the best result, it gives a result quite close to the best. Based on all of these factors, it can be concluded as SGD performs better than the other algorithms on SMS Spam Dataset with aforementioned text preprocessing techniques in general

After the text preprocessing state of SMS Spam Collection dataset, messages have been converted into a vectorized form by CountVectorizer model and tfidf normalized form by TfidfTransformer model. After that, the accuracy, F1 score, precision and recall rates of machine learning algorithms LNR, LR, KNN, SVM, GNB, MNB, BNB, PAC, SGD, SVC, PR, RCCV, RF, BC, and AC were evaluated. As a result of this evaluation, it is noted that SGD classifier panned out the best result for classifying ham-spam SMS messages with an accuracy rate of %98.21.

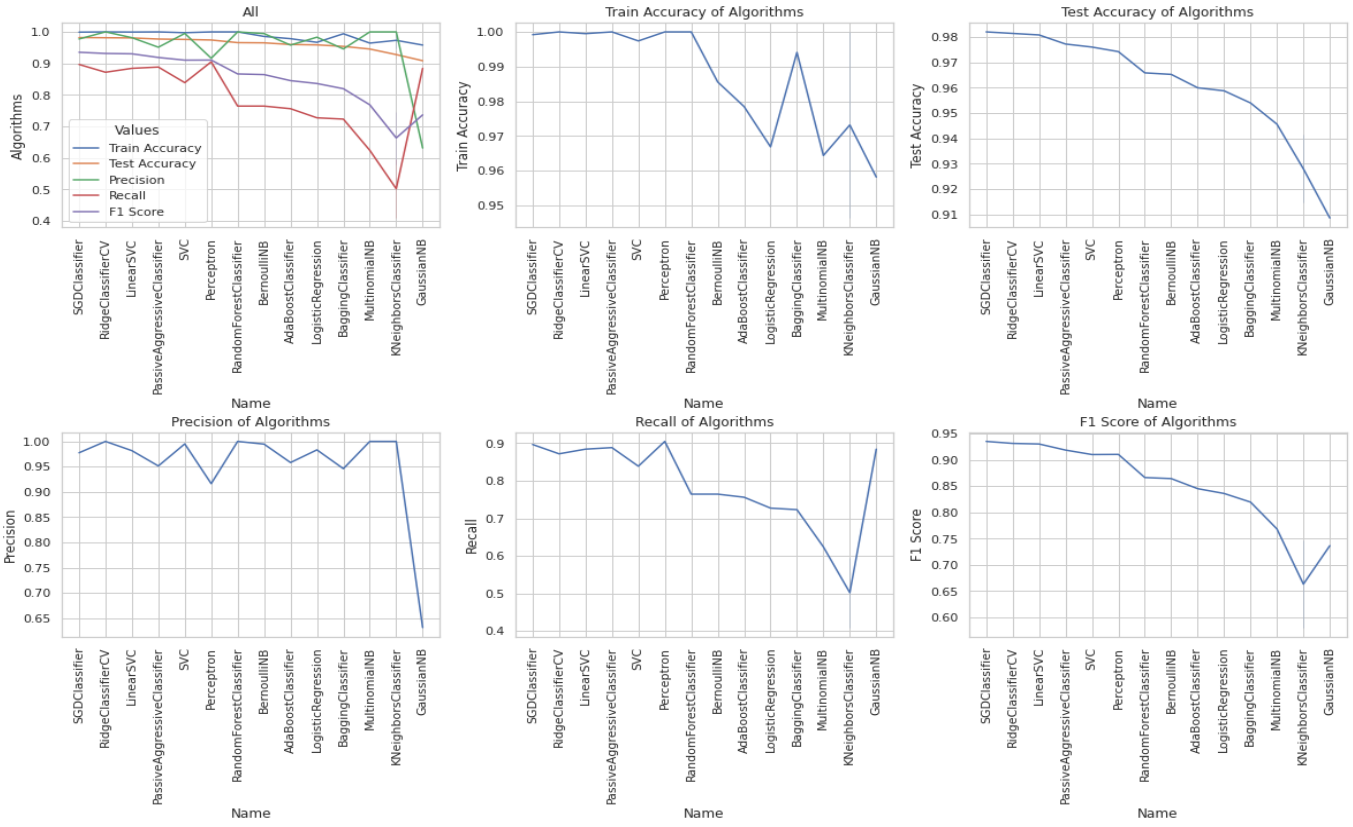

Figure 5: Comparison of accuracy rates of algorithms

References:

[1] https://datareportal.com/global-digital-overview , Digital Around the World, 2021.

[2] https://www.emarketer.com/Article/How-Frequently-SMS-Messaging-App-Users-Spammed/1014582 , How Frequently Are SMS, Messaging App Users Spammed, 2021.

[3] Sarah Jane Delany, Mark Buckley & Derek Greene. (2012). SMS SPAM FILTERING: METHODS AND DATA, EXPERT SYSTEMS WITH APPLICATIONS.

[4] Abayomi-Alli, Olusola & Onashoga, Saidat & Sodiya, Adesina Simon & Ojo, Da & Ng,. (2015). A CRITICAL ANALYSIS OF EXISTING SMS SPAM FILTERING APPROACHES.

[5] https://blogs.quickheal.com/infographic-how-can-sms-spam-harm-you , Infographic: How can an SMS Spam Harm You?, 2021.

[6] Tiago A. Almeida, José María Gómez Hidalgo, Tiago P. Silva. Towards SMS Spam Filtering: Results under a New Dataset(2013).

[7] Tiago A. Almeida, José María Gómez Hidalgo, Tiago P. Silva. Towards SMS Spam Filtering: Results under a New Dataset(2013.

[8] Dilip Singh Sisodia & Shreya Mahapatra, Arpita Sharma.(2014) Automated SMS Classification and Spam Analysis using Topic Modeling.

[9] Hand, David & Yu, Keming. (2007). Idiot's Bayes: Not So Stupid after All?. International Statistical Review. 69. 385 - 398. 10.1111/j.1751-5823.2001.tb00465.x.

[10] George H. John and Pat Langley. (1995). Estimating continuous distributions in Bayesian classifiers.

[11] Silverman, B. W., and M. C. Jones. "E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951)."

[12] Breiman, L. Random Forests (2001). https://doi.org/10.1023/A:1010933404324

[13] Y. Freund, and R. Schapire,(1997) A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.

[14] Cox, DR (1958). "The regression analysis of binary sequences".

[15] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain.

[16] Herbert Robbins. Sutton Monro. "A Stochastic Approximation Method." Ann. Math. Statist. 22 (3) 400 - 407, September, 1951. https://doi.org/10.1214/aoms/1177729586

[17] Hoerl, Arthur E., and Robert W. Kennard. "Ridge Regression: Biased Estimation for Nonorthogonal Problems."(1970) www.jstor.org/stable/1267351

[18] Crammer, Koby & Dekel, Ofer & Keshet, Joseph & Shalev-Shwartz, Shai & Singer, Yoram. (2006). Online Passive-Aggressive Algorithms.

[19] Almeida, T.A., Gómez Hidalgo, J.M., Silva, T.P. Towards SMS Spam Filtering: Results under a New Dataset. International Journal of Information Security Science (IJISS), 2(1), 1-18, 2013

[20] https://www.educative.io/edpresso/countvectorizer-in-python , CountVectorizer in Python, 2021.

[21] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html , 2021.

[22] Hasibe Büşra Doğru, Alaa Ali Hameed, Sahra Tilki, Akhtar Jamil. "Comparative Analysis Of Deep Learning And Traditional Machine Learning Methods For Turkish Text Classification". ICMS21