Simulation Approach of Algorithm Theories for Predictive Model's Reliability

Abdulrehman A. Mohamed[1], Dr. George O. Okeyo PhD[2], & Dr. Michael W. Kimwele, PhD[3]
School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology,
P. O. Box 62000- 00200, Nairobi, Kenya
[1]almutwafy@gmail.com, [2]gokeyo@jkuat.ac,ke and [3]mkimele@jkuat.ac.ke

Abstract - In recent times, wide adoption of machine learning algorithms for predictive modeling has been successfully used in numerous classification problems. While most of them outperform classification accuracy to an approximately hundred percent but are never use in real world situations. This is attributed to the difficulty of obtaining unbiased accuracy of predictive model's reliability due to machine learning techniques sensitive to worse-case scenarios yielding unreliable results. Even though, most techniques use extensive testing of infinite domains against infinite machine learning algorithms to yield improvement in reliability estimation performance, but most models could not fit the reliability estimation performance in comparison with traditional approaches such as Weibull distribution in reliability engineering. It is against this background that the study developed a simulation approach of algorithm theories for predictive model's reliability that will ensure model reliability to censor data before it is developed. In order to realize this objective the study implemented crowd-sourcing analytic and OSEMN (Obtain, Scrub, Explore, Model, and iNterpret) model for simulation of the model. Specifically, the study computed the mathematical theories behind the data-driven ensemble model using tweet dataset to show its reliability to improve censoring of profane words. Thereafter, algorithm tuning and spot-checking approach - a process of finding optimal classifier from a group of algorithm categories was implemented for model development. The results of simulation and modeling yielded three single prediction models: Logistic classifier (96.71%), Naïve Bayes classifier (98.51%) and k-NN classifier (94.66%). However, on further analysis of the three models using ensemble techniques of bootstrap mean and direct mean, the results showed an improvement of about 1% from direct mean (96.63%) to bootstrap mean (97.58%).

Index Terms – Algorithm theories, Model reliability, Model Accuracy and Simulation & modeling

1 INTRODUCTION

1.1 Background Study

Model reliability refers to a model that yields consistent results and is said to be reliable if it has internal consistency, where the predictor variables are sufficiently contributing to the model predictability.

However, the accuracy of a prediction model does not necessary mean it is reliable since, most of the near hundred percent accurate prediction models are never used in real world application due to sensitivity of worse-case scenario [18].

Even though numerous machine learning techniques have been developed to improve reliability estimation performance, but still some optimal prediction model have found no use in industries.

It is against this background that the study developed a simulation approach for algorithm theories for predictive model's reliability that will ensure model reliability to censor data before it is developed.

1.2 Model Accuracy

According to [10], machine learning model accuracy can be defined as the measure to determine which model is best at identifying relationships and patterns between variables in a dataset based on the input, or training data. The better a model can generalize to unseen data, the better predictions and insights it produces that deliver more business value.

However, statistically, accuracy can be defined as the measure of the ratio of correct predictions to the total number of cases evaluated.

1.3 Model Reliability

Statically, reliability is the overall consistency of a measure hence a model is said to have a high reliability if it produces similar results under consistent conditions [16].

Alternatively, reliability is the characteristic of a set of test scores that relates to the amount of random error from the measurement process that might be embedded in the scores. Therefore, scores which are highly reliable are accurate, reproducible, and consistent from one testing occasion to another.

Moreover, if the testing process were repeated with a group of test, then the same results would be obtained. Finally, there exists various tests of reliability coefficients, with values ranging between 0.00 (high error) and 1.00 (no error), which are used to indicate the amount of error in the scores [11].

1.4 Spot-checking Approach

According to [4], spot-checking approach is a technique that involves testing a large suite of algorithms against variant dataset of specific domain problem in order to quickly suggest which types of algorithms are fit for the domain in question.

However, spot-checking does not give an optimum algorithm, rather than is a starting point for searching for the best algorithm from the result of all algorithms which were suitable for the domain problem.

Therefore, the main objective of spot-checking algorithm is to discover a suite of algorithm that might work well with the domain problem, rather than selecting a specific or popular algorithm used by other researchers or in their own interest [1].

Moreover, the result of the spot-checking approach gives a baseline for starting experimental search for an optimal algorithm. Hence, the paradox of spot-checking approach is analytical focused rather than result focused. Unlike grid search that is focused on optimal algorithm or algorithm tuning that is focused on optimal configuration of an algorithm.

2 METHODOLOGY OF THE STUDY

2.1 Introduction

It is asserted by [8] that, machine learning should focus on the entire data analytical process either undertaking it holistically or each step at a time, whilst considering the reliability of the predictions produced. However, in the larger scope of data science, several processes must be taken that included data acquisition, data cleaning, exploratory visualization, data integration, model criticism and revision, and presentation of results to domain experts.

2.2 Model Design Approach

The main objective of this methodology was to design a data-driven ensemble model and demonstrate its reliability to detect profane words in social media. The study employed an OSEMN (Obtain, Scrub, Explore, Model, and iNterpret) model (Manson & Wiggins, 2010b) for designing the data-driven approach model as shown in the Figure 1.

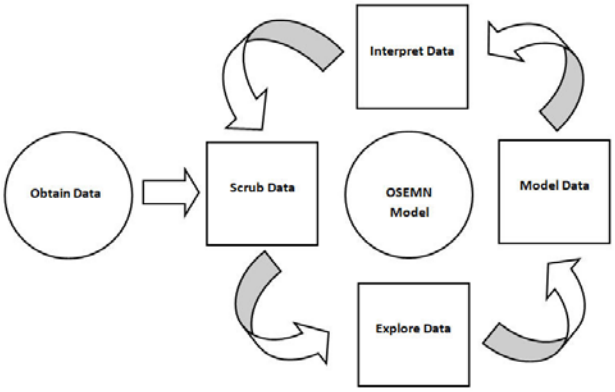


Figure 1: OSEMN Model (Manson & Wiggins, 2010a)

The OSEMN model has five stages of: obtaining data, scrubbing data, exploring data, modeling data and interpreting data.

2.2.1 Obtain Data

Obtaining data stage involved retrieving data from various sources such as databases, web applications, social media platforms, web pages or servers, and among others. Nevertheless, clean or treated datasets can also be obtained from data centers or open source databases ready to be used in machine learning algorithm for prediction or classification of a domain problem.

However, obtaining raw data, requires a researcher to be equipped with various data retrieval skills such querying languages, programming languages, scripting languages and API manipulation techniques, in order to support automatic retrieval of raw data.

Specifically, the process involves downloading data from remote locations, querying data from databases or API, extracting data from another file and generating one own data from reading sensors or taking surveys.

2.2.2 Scrub Data

Scrubbing data stage involved simply data cleaning (scrubbing). Raw data is usually complex and disorganized consisting of inconsistent labels, extraneous characters, unsupported data formats, missing data or unwanted rows and columns. Nonetheless, Machine leaning algorithms are sensitive to these issues, and it is a prerequisite for the raw data to be cleaned (scrubbed) before conducting experiments for prediction or classification problems.

However, cleaning raw data requires a researcher to be equipped with various data cleaning skills such querying languages, programming languages, and scripting languages, in order to support automation of raw data cleaning.

Specifically, the scrubbing process involves filtering lines, extracting certain rows or columns, replacing values, extracting words, handling missing values and converting data from one format to another.

2.2.3 Explore Data

Exploring of data stage involved creating graphical representation of clean data by summarizing the dataset in to charts and performing dimensionality reduction on the dataset. Machine learning algorithms diversity requires various dataset assumptions to be met before using it, such as the choice of binning, feature distribution, outliers, and data type.

However, exploring data requires a researcher to be equipped with various data exploration skills such statistics, data analytics and visualization techniques in order to support the automation of data exploration.

Specifically, the exploration process involves creating histograms, whisker plots, scatter plots, clustering dataset by grouping nodes of graphs and performing dimensionality reduction.

.

2.2.4 Model Data

Modeling of data stage involved building prediction or classification models using training datasets and test datasets. The training dataset is used by the algorithm to learn relationships and patterns and acquire knowledge to predict complex phenomenon. The test dataset is the unseen dataset and is used by the researcher to evaluate the performance of the algorithm's ability to learn from the training dataset.

However, modeling data requires a researcher to be equipped with various data modeling skills such as supervised learning models, semi-supervised learning models and unsupervised learning models in order to support the automation of data modeling.

Specifically, the data modeling process involves tuning of algorithm's parameters for best configuration, use of single classifiers or multiple classifiers for optimal results and the use of various performance metrics to find an optimum model.

2.2.2 Interpret Data

Interpreting of data stage involved analyzing the predictive power of the developed model and its interpretability. The predictive power of model lies in its ability to generalized quantitative data by making accurate quantitative predictions of data in repeated experiments. On the other hand, the interpretability of a model lies in its ability to generalize data by suggesting to the researcher which would be the most appropriate experiment to perform next for an optimal model choice.

However, interpreting data requires a researcher to be knowledgeable in numerous domain aress and equipped with different interpretation skills such as representing text as bag-of-words, rather than bag-of-letters, representing graph as sub-graph, rather than spectrum of laplacian, choosing single model for classification, rather than ensemble model or choosing linear function algorithm for prediction, rather than logistical function algorithm.

Specifically, the interpretation process involves drawing conclusions from the developed model, evaluating the results, relating the model to the problem and communicating the results to the domain area.

3 SIMULATION & MODELING

3.1 Introduction

Industries have been using machine learning models to make informed business decisions in order to have a competitive advantage in their field. Therefore, the more accurate model outcomes result means better decisions. Nevertheless, the cost of errors can be huge, but optimizing model accuracy also alleviates that cost. Moreover, there is a point of diminishing returns when the value of developing a more accurate model wouldn't result in a corresponding profit increase in some cases [10].

However, model accuracy doesn't translate to a reliable model where the cost of testing reliability is too high. Therefore, it is against this background the study conducted simulation and modeling to prove its reliability to sensor profane word in social media before its final development

3.2 Proposed Model

The main objective of simulation and modeling was to developed a data-driven ensemble model and demonstrate its reliability to detect profane words in social media. The study employed the OSEMN (Obtain, Scrub, Explore, Model, and iNterpret) model, crowd sourcing analytic and spot-checking approach for development of the model as shown in the Figure 2.

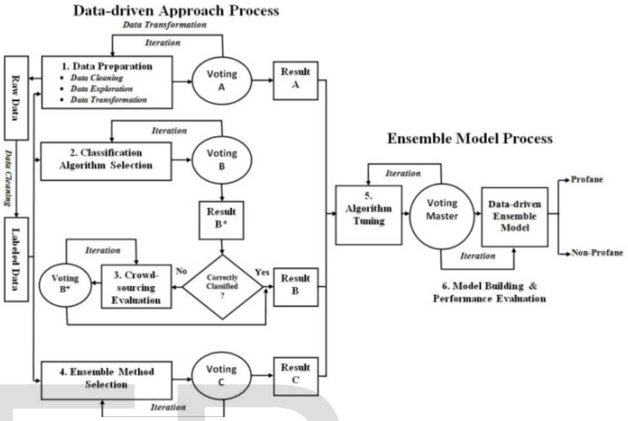


Figure 2: Data-driven Ensemble Model

The data-driven ensemble model for detection of profane words in social media could be inferred as a combination of two main processes of:

1. Data-driven Approach Process
2. Ensemble Model Process

3.2.1 Data-driven Approach Process

The data-driven approach process shown in Figure 2 consists of two main tasks:

A. **Input**: The input task accepts data as an input for the following 4 processes

1. **Data preparation process**: It contains the following sub processes:
   a. Data cleaning: It accepts raw data tweets in text format and apply a suit of MS Excel text manipulation formula to remove noise to output labeled data
   b. Data exploration: It receives labeled data and apply statistical and visualization methods to output normalized data as precondition for algorithm execution
   c. Data transformation: It receives labeled data and applies a suite of transformation methods to output optimal feature selected data set.
2. **Classification algorithm selection**: It receives labeled data and applies a suite of binary classification algorithms to a series of categories to output optimal classifier for each category

3. **Crowd-sourcing evaluation:** It accepts misclassified labeled data from classification algorithm selection process and applies human expert skills to classify data to output optimal classification.
4. **Ensemble method selection**: It receives labeled data and applies a suite of ensemble methods to output optimal method for model building

B. Loop: The loop iterate the 4 processes of: classification algorithm selection, crowd-sourcing evaluation ensemble method selection and data transformation sub-process in a series of alteration in majority voting to output optimal task for each process.

3.2.2 Ensemble Model Process

The ensemble model process shown in Figure 2 also consists of two main tasks:

A. **Input**: The input task accepts data as an input for the following 4 processes

1. **Algorithm Tuning:** It receives 3 results from data preparation, classification algorithm selection and ensemble method selection and applies a suite of parameterized algorithm options in a majority voting to output optimal setting for each optimal algorithm
2. **Model Building and Performance Evaluation**: It receives results from algorithm tuning and applies a serious of performance metrics in a majority voting to output the optimal ensemble model

B. **Loop:**

3. **The loop iterate the 2 processes of:** algorithm tuning and model building and performance evaluation in a series of alteration in majority voting to output optimal task for each process for data-driven ensemble building

3.3 Model Development

The study's model development was based on the proposed model's processes of the data-driven approach process and ensemble model process and hence, simulation & modeling was anchored on the following six processes:

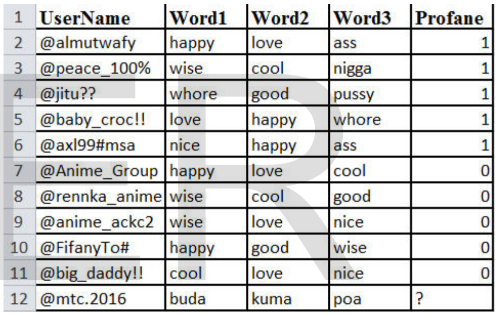
1. Data Preparation (data cleaning, data exploration and data transformation)
2. Classification Algorithm Selection
3. Crowd-sourcing Evaluation
4. Ensemble Method Selection
5. Algorithm Tuning
6. Model Building & Performance Evaluation

3.4 Data Preparation

Raw data was retrieved using Tweeter API software Tweet Archivist and then used for data preparation. Nonetheless, there exist several preprocessing algorithm filters in WEKA such as AddClassification, AttributeSelection, Discretize, NominalToBinary, and PLSFilter (Partial Least Square Regression) filters in machine learning for data cleaning. However, in order to simulate the theories behind the data-driven preprocessing algorithm filters, the study explored statistical methods of encoding implemented by NominalToBinary algorithm filters and crowd-sourcing technique using human judge to remove noise.

In order to achieve this objective the study used Tweet dataset of 10 correctly classified instances, one misclassified instance and one attribute of no interest. The dataset consists of four input variables of: username, word1, word2, word3 and one output variable with two values of 0 and 1, to create a binary classification problem as shown in Table 1.

Table 1: Tweet Dataset



1	UserName	Word1	Word2	Word3	Profane
2	@almutwafy	happy	love	ass	1
3	@peace_100%	wise	cool	nigga	1
4	@jitu??	whore	good	pussy	1
5	@baby_croc!!	love	happy	whore	1
6	@axl99#msa	nice	happy	ass	1
7	@Anime_Group	happy	love	cool	0
8	@rennka_anime	wise	cool	good	0
9	@anime_ackc2	wise	love	nice	0
10	@FifanyTo#	happy	good	wise	0
11	@big_daddy!!	cool	love	nice	0
12	@mtc.2016	buda	kuma	poa	?

3.4.1 Target-based Encoding (TBE)

According to [6] target-based encoding is a statistical theory of numerization of categorical variables through target variable. The theory replaces the categorical variables with only one new numerical variable and then replaces each category of the categorical variable with its corresponding probability of the target (if categorical) or average of the target (if numerical). The theory can be represented mathematically using Tweet profane word dataset in Table 1 as follows:

TBE (V) = CVF (T1) / [CVF (T1) + CVF (T0)]

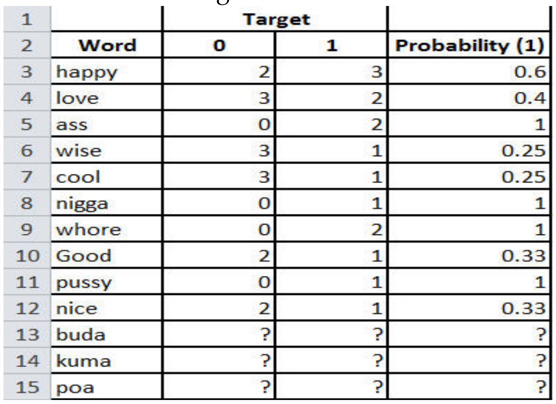
Where:

- TBE = Target-based Encoding
- V = Variable
- CVF (T1) = Total Categorical Variable Frequency for default target class (1)
- CVF (T0) = Total Categorical Variable Frequency for target class (0)

Therefore, using the formula and the Tweet dataset, it could be observed that, the variable happy had a frequency of 2 for class (0) and 3 for class (1). Hence applying the function: TBE (happy) = 3 / (2+3) = 0.60.

However, practically, the NominalToBinary algorithm filter will iterate through the whole dataset to calculate the value for each variable as shown in Table 2.

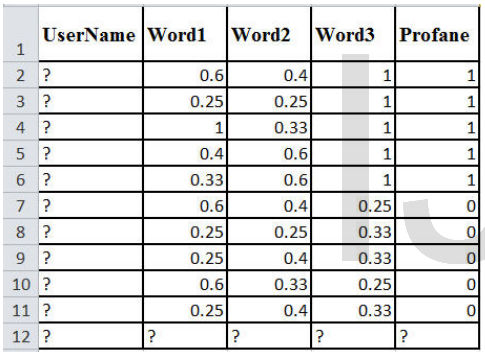
Table 2: Calculating TBE Value



1		Target		
2	Word	0	1	Probability (1)
3	happy	2	3	0.6
4	love	3	2	0.4
5	ass	0	2	1
6	wise	3	1	0.25
7	cool	3	1	0.25
8	nigga	0	1	1
9	whore	0	2	1
10	Good	2	1	0.33
11	pussy	0	1	1
12	nice	2	1	0.33
13	buda	?	?	?
14	kuma	?	?	?
15	poa	?	?	?

After the NominalToBinary algorithm has completed iteration for calculating TBE values for each variable, then it would replace the categorical variables with values such as happy = 0.60, love = 0.40, ass = 1.00 and so on. The result of the final transformation of the algorithm is a Numerized Tweet Dataset1 shown in Table 3.

Table 3: Numerized Tweet Dataset1



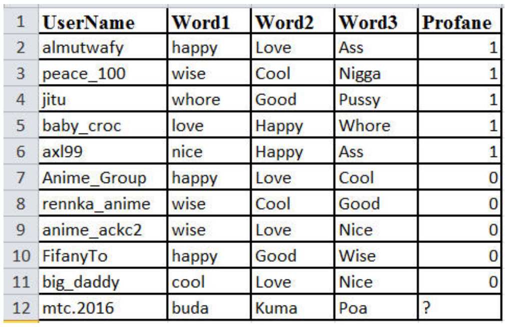
1	UserName	Word1	Word2	Word3	Profane
2	?	0.6	0.4	1	1
3	?	0.25	0.25	1	1
4	?	1	0.33	1	1
5	?	0.4	0.6	1	1
6	?	0.33	0.6	1	1
7	?	0.6	0.4	0.25	0
8	?	0.25	0.25	0.33	0
9	?	0.25	0.4	0.33	0
10	?	0.6	0.33	0.25	0
11	?	0.25	0.4	0.33	0
12	?	?	?	?	?

3.4.2 Crowd-sourcing for Noise Removal

Even though, there are several algorithm filters to transform dataset such as Supervised Attribute-based filters including Discretize, NominalToBinary, or PLSFilter, but most of them are short of completely classifying the dataset to 100%. Therefore, the study preferred a combination of algorithm filters with crowd-sourcing technique to completely transform the dataset to 100%. Crowd-sourcing here is used to manually remove noise such as attributes of no interest, or symbols such as ", ', *, +,-, and %, in order to improve classification.

The crowd-sourcing technique could be simulated using Tweet dataset in Table 1, where it has 10 out of 11 instances correctly classified (91%) and lot of noise in the variable username. Therefore, the human judge manually removes noise such as #, %, or @ symbols from the username attribute of non-interest from the dataset. The result of combination of the algorithm and crowd-sourcing had an improved classification with reduced noise as shown in Table 4.

Table 4: Non-Noise Crowd-sourced Dataset



1	UserName	Word1	Word2	Word3	Profane
2	almutwafy	happy	Love	Ass	1
3	peace_100	wise	Cool	Nigga	1
4	jitu	whore	Good	Pussy	1
5	baby_croc	love	Happy	Whore	1
6	axl99	nice	Happy	Ass	1
7	Anime_Group	happy	Love	Cool	0
8	rennka_anime	wise	Cool	Good	0
9	anime_ackc2	wise	Love	Nice	0
10	FifanyTo	happy	Good	Wise	0
11	big_daddy	cool	Love	Nice	0
12	mtc.2016	buda	Kuma	Poa	?

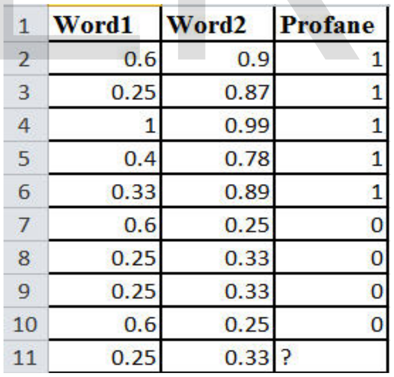
3.5 Classification Algorithm Selection

According to [3], algorithm selections in machine learning are affected by various attributes such as noisy dataset, number of attributes, and size of the dataset. However, in order to simulate the theories behind the formulated model, the study evaluated two techniques of algorithm tuning using k-NN algorithm parameterization and data exploration by generating box plot whisker with outliers using Tweet dataset

3.5.1 Data-driven Algorithm Tuning

To demonstrate data-driven algorithm tuning the study preferred a Tweet dataset consisting of 9 classified training dataset, one unseen test data of profane words. The dataset have been converted to numerized and its attributes reduced to three: word1, word2 and profane for binary classification of 1 or 0, as shown in Table 5.

Table 5: AttributeReduced Numerized Tweet Dataset



1	Word1	Word2	Profane
2	0.6	0.9	1
3	0.25	0.87	1
4	1	0.99	1
5	0.4	0.78	1
6	0.33	0.89	1
7	0.6	0.25	0
8	0.25	0.33	0
9	0.25	0.33	0
10	0.6	0.25	0
11	0.25	0.33	?

3.5.1.1 k-Nearest Neighbor

It was elaborated by [19], that K-Nearest Neighbor is a supervised learning algorithm for classification, where the querying of unseen test dataset is classified based on majority of k-nearest neighbor. The classifier is not based on any model but on its memory. Given unseen instance point, and k-numbers of training instances closest to the unseen instance point, it can calculate a prediction of the unseen dataset by voting from the k-nearest neighbor using minimal distance.

3.5.1.2 k-Nearest Neighbor Computation

In order to compute the k-nearest neighbor algorithm the following pseudo-code was implemented as follows:

1. To determine parameter values of k, where k = number of nearest neighbors
2. To calculate the distance between unseen data and all the training dataset
3. To rank the training data based on K^th minimum distance to unseen data
4. To match ranked training instances to their respective classes
5. To predict unseen data by calculating the majority match ranked training data

3.5.1.3 k-Nearest Neighbor Prediction

In order to simulate the prediction of the 10th unseen instance of profane word in the Table 5 AttributeReduced Numerized Tweet Dataset, the study implemented the following:

1. The study preferred three parameter values of k = 1, k = 3 and k = 5 to demonstrate algorithm tuning.
2. The study calculated the distance between 10th unseen instance and all other training instances, by assuming the Euclidean distance, which is the root of the square difference between coordinates of a pair of points, and presented mathematical as:

$$d_{ij} = \sqrt{\sum_{k=1}^{n}\left(x_{ik} - x_{jk}\right)^2}$$

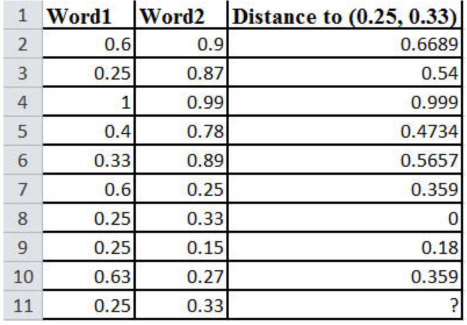
In order to evaluate the distance between 10th unseen data with coordinates (0.25, 0.33) and the 1st instance of training data with coordinates (0.60, 0.90), using Euclidean distances the value is calculated as follow:

$$\text{Euclidean Distance} = \sqrt{(0.60 - 0.25)\ 2\ +\ (0.90 - 0.33)\ 2}$$

$$\text{Hence, } \sqrt{0.4474} = 0.6689$$

Therefore, the Euclidean distance was calculated for each for the remaining training instances, and the results are shown in Table 6 distance computation.

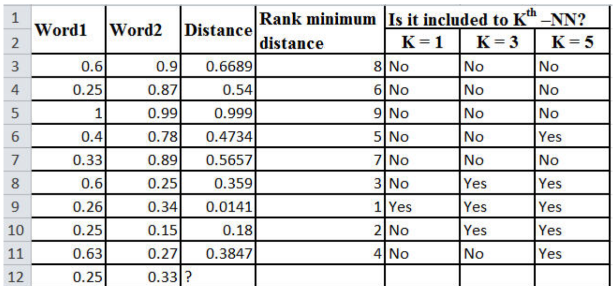
Table 6: Distance Computation



1	Word1	Word2	Distance to (0.25, 0.33)
2	0.6	0.9	0.6689
3	0.25	0.87	0.54
4	1	0.99	0.999
5	0.4	0.78	0.4734
6	0.33	0.89	0.5657
7	0.6	0.25	0.359
8	0.25	0.33	0
9	0.25	0.15	0.18
10	0.63	0.27	0.359
11	0.25	0.33	?

1. The study ranked the training data based on K^th minimum distance to unseen data, using k = 1, k = 3 and k = 5, as shown in Table 7.

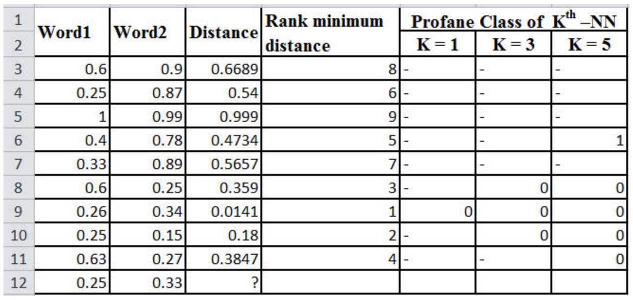
Table 7: Ranked Minimum Distance



1	Word1	Word2	Distance	Rank minimum distance	Is it included to K^th –NN?		
2					K = 1	K = 3	K = 5
3	0.6	0.9	0.6689	8	No	No	No
4	0.25	0.87	0.54	6	No	No	No
5	1	0.99	0.999	9	No	No	No
6	0.4	0.78	0.4734	5	No	No	Yes
7	0.33	0.89	0.5657	7	No	No	No
8	0.6	0.25	0.359	3	No	Yes	Yes
9	0.26	0.34	0.0141	1	Yes	Yes	Yes
10	0.25	0.15	0.18	2	No	Yes	Yes
11	0.63	0.27	0.3847	4	No	No	Yes
12	0.25	0.33	?				

2. The study matched ranked training instances to their respective classes as follows shown in Table 8.

Table 8: Matched to Profane Classes



1	Word1	Word2	Distance	Rank minimum distance	Profane Class of K^th –NN		
2					K = 1	K = 3	K = 5
3	0.6	0.9	0.6689	8	-	-	-
4	0.25	0.87	0.54	6	-	-	-
5	1	0.99	0.999	9	-	-	-
6	0.4	0.78	0.4734	5	-	-	1
7	0.33	0.89	0.5657	7	-	-	-
8	0.6	0.25	0.359	3	-	0	0
9	0.26	0.34	0.0141	1	0	0	0
10	0.25	0.15	0.18	2	-	0	0
11	0.63	0.27	0.3847	4	-	-	0
12	0.25	0.33	?				

3. Finally, the study predicted unseen data by calculating the majority match ranked training data using mean as follows:

K = 1, Mean = 0/ 1 = 0 hence, belongs to class 0

K = 3, Mean = (0 + 0 + 0) /3 = 0 hence, belongs to class 0

K = 5, Mean = ((0 + 0 + 0 + 0 + 1) /5 = 0.2 which belongs to class 0, since it is < 0.5

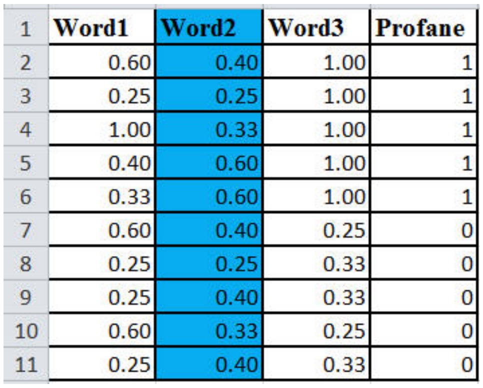
The prediction result showed that, higher K^th values tend to be more accurate as oppose to lower values. Therefore, the results inferred that algorithm tuning had some effects on classification prediction in machine learning.

3.5.2 Data-driven Exploration

It was explained by [12], that data exploration is a technique that is used to describe data by means of statistical and visualization techniques such as mean, standard deviation, Gaussian distribution, histogram, scatter charts and box plots, which is prerequisite for most machine learning algorithms for further analysis.

In order to simulate the theories behind data-driven exploration, the study preferred to generate a box plot from numerized Tweet dataset shown in Table 9.

Table 9: Tweet Numerized Tweet Dataset



1	Word1	Word2	Word3	Profane
2	0.60	0.40	1.00	1
3	0.25	0.25	1.00	1
4	1.00	0.33	1.00	1
5	0.40	0.60	1.00	1
6	0.33	0.60	1.00	1
7	0.60	0.40	0.25	0
8	0.25	0.25	0.33	0
9	0.25	0.40	0.33	0
10	0.60	0.33	0.25	0
11	0.25	0.40	0.33	0

3.5.2.1 Box Plot (Box & Whisker Plots with Outliers)

It was elaborated by [17], that a box Plot is a one-dimensional numerical data-based graphical methodology of displaying variation in samples of a statistical population without making any assumptions of the underlying statistical distribution.

Moreover, it was explained by (Rumsey, 2010), that the box and whisker plot with outliers, displays the distribution of data based on the five statistical summaries: minimum value, 25th percentile (known as Q1), median, 75th percentile (Q3), and maximum. These statistical summaries were presented in a box plot as: the rectangle that extent Q1 to the Q3; (interquartile range - IQR); the part inside the rectangle showing the median; and "whiskers" above and below the box showing the positions of the minimum and maximum respectively as shown in Figure 3.

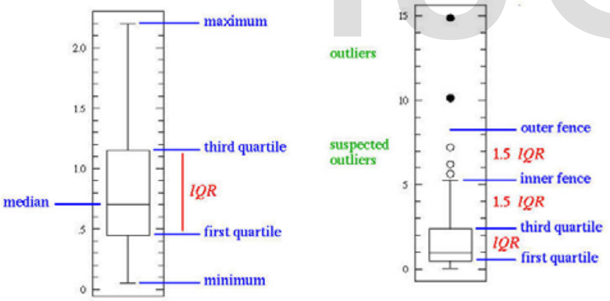


Figure 3: Box and whisker plot with outliers [17]

It is commonly observed in real world dataset to contain unusually very high maximum and minimum value called outliers. [17], has provided a precise definition for two types of outliers:

1. Outliers are either 3×IQR or more above the third quartile or 3×IQR or more below the first quartile.
2. Suspected outliers are either 1.5×IQR or more above the third quartile or 1.5×IQR or more below the first quartile.

The individual outlying data points are presented as unfilled circles for suspected outliers or filled circles for outliers. Outliers are not necessarily "bad" data-points; indeed they may well be the most important, and most information rich part of the dataset, which may need special attention depending on the domain study.

3.5.2.2 Box Plot Computation

In order to compute the box plot the following three steps were implemented as follows:

1. To compute the five statistical summary
2. To identify outliers
3. To generate a box and whisker plot with outliers
4. To interpret the plot

In order to simulate the five statistical summaries, the study preferred the 2nd attributes word2 Tweet Numerized Profane Dataset: Word2 = {0.40, 0.25, 0.33, 0.60, 0.60, 0.40, 0.25, 0.40, 0.33, and 0.40}

1. To compute the five statistical summary: In order to compute the five statistical summaries, the study rearranged the Word2 dataset into ascending order from the smallest to the largest as shown below:

Word2 = {**0.25, 0.25, 0.33, 0.33, 0.40, 0.40, 0.40, 0.40, 0.60, 0.60**}

Given that:

- Q1 = Middle number for the 1st half of the dataset
- Q2 = Middle number of the whole dataset
- Q3 = Middle number of the 2nd half of the dataset
- Q4 = Largest value of the dataset

Therefore,

Median (Q2) = (0.40 + 0.40) /2 = 0.40, hence, Minimum = 0.25, 25th Percentile (Q1) = 0.33, 75th Percentile (Q3) = 0.40, and Maximum = 0.6

2. To identify outliers: In order to identify the outliers, study computed the inter-quartile range (IQR), which is the width of box in the box and whisker plot. The IQR can be used to measure how spread out is the dataset values from central value and how far away from central value, known as outlier. The IQR can be mathematically represented as follows:

- IQR = Q3 – Q1, hence
- IQR = 0.40 – 0.33 = 0.07

Therefore, to bound the range of outlier the following functions are given for upper bound and lower bond:

- Lower Fence: Q1 − 1.5 × IQR = 0.33 – (1.5*0.07) = 0.225
- Upper Fence: Q3 + 1.5 × IQR = 0.40 + (1.5*00.07) = 0.505

Therefore, from the Tweet dataset there two outliers of 0.6 and 0.6 since the upper bound is 0.505 and no outliers below the lower bound of 0.225

3. To generate a box and whisker plot with outliers

In order to generate a box and whisker plot with outliers, the rest of the attributes word1 and word2 datasets are computed by repeating the step1 and step2 above. The results of the computation are used to generate the plot as shown in Figure 4:

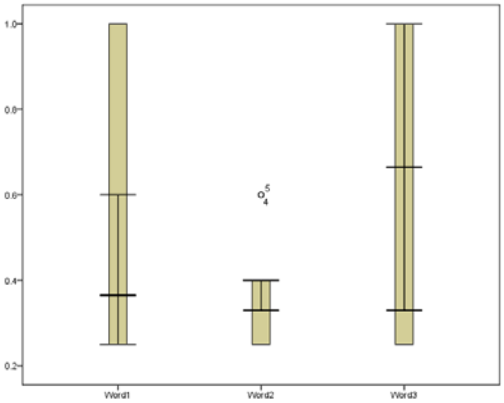


Figure 4: Box and Whisker Plot with Outliers

4. To interpret the plot

In order to interpret the box and whisker with outlier plot, the study evaluated the following attributes as defined by [15],

1. **Measure of Centrality –** It is the median, which is indicated by the vertical line that runs down the center of the box. Therefore, the results indicates samples of 10 from population centered on Word1 = 0.365, Word2 = 0.40 and Word3 = 0.665 respectively.
2. **Measure of Spread –** It is the box length, which is an indication of the sample variability (standard deviation). Therefore, the results indicate sample of 10 from centered on Word1 = 0.365 with standard deviation = 0.75, Word2 = 0.40 with standard deviation = 0.15, and Word3 = 0.665 with standard deviation = 0.75 respectively.
3. **Measure of Skewness –** It is the position of the box in its whiskers and the position of the line in the box indicate whether the sample is symmetric or skewed, either to the right or left. Therefore, the results indicate Word1 and Word2 are skewed to the right and Word3 is symmetric. Since, if the line is close to the center of the box and the whisker lengths are the same is symmetric. If the top whisker is much longer than the bottom whisker and the line is dropping towards the bottom of the box, then is skewed to the right. If the bottom whisker is much longer than the top whisker and the line is rising to the top of the box, then it is skewed to the left.
4. **Types of Population –** It is referred to as being heavy-tailed or light-tailed, where the normal population, is either too heavy or too light, and represented by the bell shaped curve. The results indicate that Word1 and Word2 are light-tailed and Word3 is heavy-tailed with a normal distribution. Since, normal population has whiskers the same length as the box, or slightly longer. If the length of the whiskers is shorter than the length of the box, then is lightly-tailed and if the whiskers are extremely short or absent then is slightly light-tailed.

3.6 Data-driven Classification & Crowd-sourcing Evaluation

It was reported by[3], that there are several suites of algorithms, such as ruled-based algorithms, k-nearest neighbours, Bayesian algorithms, logistic regression algorithms and ensemble models, which support binary classification problem. However, there is a huge diversity among algorithm performances such as high predicting power, or high flexibility or high adaptively or high self-tuning. As a result of which, it is impossible to know in advance which algorithms are suitable for classification.

Therefore, in order to discover the best algorithm classification, an empirical study should be conducted, that will simulate the theories behind the proposed model. Hence, the study evaluated the mathematical theories of algorithm functions of logistic regression classification and crowd-sourcing classification.

3.6.1 Logistic Regression Classification

It was elaborated by [9], that the logistic regression classification model accepts real-valued inputs to generate prediction as to the probability of the input belonging to the default class (class 1). If the probability is > 0.5, the output is inferred as a prediction for the default class (class 1), otherwise the prediction is for the other class (class 0).

By simulating the numarized Tweet dataset2, the logistic regression has four coefficients similar to linear regression, which are mathematical presented as follows: Output = b0 + b1x1 + b2x2 + b3x3

Therefore, the objective of the learning algorithm is to discover the best values for the coefficients (b0, b1, b2, and b3) based on the training data. However, unlike in linear regression, the output is transformed into a probability using the logistic function: P (class=1) = 1 / (1 + e - (b0 + b1x1 + b2x2 + b3x3))

In order to make prediction, the logistic function uses stochastic gradient descent by calculating prediction and error for each instances of the training dataset.

3.6.2 Stochastic Gradient Descent

The stochastic gradient descent uses logistic regression model to estimate a prediction for each instance in the training dataset and calculate error for each prediction. It is implemented by estimating the values of the coefficient in logistic regression model using the following algorithm as defined by [9]:

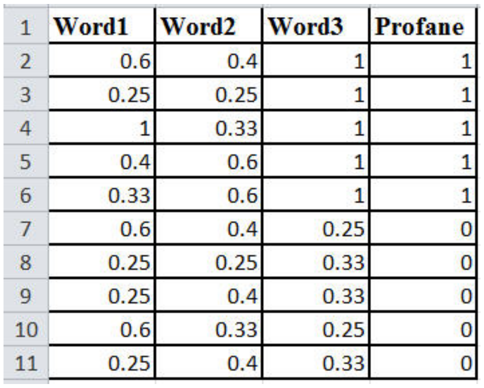
- Given each training instance:
- Calculate a prediction using the current values of the coefficients.
- Calculate new coefficient values based on the error in the prediction.

Thereafter, the algorithm is repeated until the model is sufficient accurate with minimal error or a specified number iterations.

3.6.2.1 Calculating Prediction

In order to simulate the stochastic gradient descent algorithm, study preferred the Numerized Tweet Dataset2 shown in Table 10.

Table 10: Numerized Tweet Dataset2



1	Word1	Word2	Word3	Profane
2	0.6	0.4	1	1
3	0.25	0.25	1	1
4	1	0.33	1	1
5	0.4	0.6	1	1
6	0.33	0.6	1	1
7	0.6	0.4	0.25	0
8	0.25	0.25	0.33	0
9	0.25	0.4	0.33	0
10	0.6	0.33	0.25	0
11	0.25	0.4	0.33	0

Thereafter, the study assigned 0.0 to each coefficient to calculate the probability of the first training instance that belongs to class (1). Hence b0 = 0.0, b1 = 0.0, b2 = 0.0 and b3 = 0.0 and the first training instance was: word1 = 0.60, word2 = 0.40, word3 = 1.00 and profane = 1.Therefore, using the logistic regression function, prediction could be calculated as follows:

- Prediction = 1 / (1 + e - (b0 + b1x1 + b2x2 + b3x3))
- Prediction = 1 / (1 + e - (0.0 + 0.0*0.60 + 0.0*0.40 + 0.0*1.00))
- Prediction = 1 / (1 + e - (0))
- Prediction = 1 / (1 + 1) = 0.5

In practice, the algorithm iterates through the whole dataset to calculate the prediction for each instance of the training dataset by calculating new coefficients.

3.6.2.2 Calculating New Coefficients

In order to calculate the new coefficients values the study implemented the following function derived from the prediction function: b = b0 + alpha * (y – prediction) * prediction * (1 – prediction) * x

Where:

- b = coefficient we are updating
- b0 = intercept (preferred 1.0)
- alpha = learning rate (0.1- 0.3)
- prediction = output of making a prediction using the model
- y = predicted class
- x = input value for coefficient

The alpha value is the learning rate that controls the number of changes in the coefficients it learns each time it updates. It should be specified at the start of the training of the dataset. The best values are commonly given in the range 0.1 to 0.3, and for the purpose of demonstration the study uses 0.3 as a value.

Therefore, in order to update the coefficient using the prediction (0.5) and coefficient values (0.0) from the previous section, the following new values are obtained

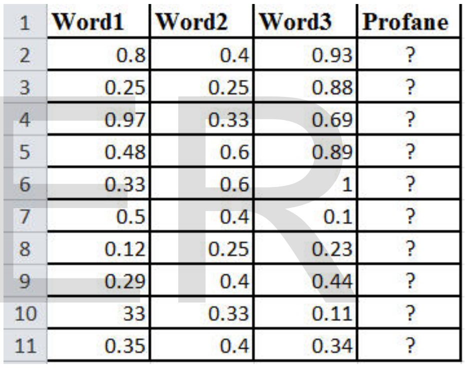
- b0 = b0 + 0.3 * (1 – 0.5) * 0.5 * (1 – 0.5) * 1.0 = 0.0375
- b1 = b1 + 0.3 * (1 – 0.5) * 0.5 * (1 – 0.5) * 0.60 = 0.0225
- b2 = b2 + 0.3 * (1 – 0.5) * 0.5 * (1 – 0.5) * 0.40 = 0.0150
- b3 = b3 + 0.3 * (1 – 0.5) * 0.5 * (1 – 0.5) * 1.0 = 0.0375

In practice, the algorithm has predefined number of iteration called epoch for updating the model for each training instance in the dataset. At the end of each epoch, error values for the model are calculated and adjusted to determine the accuracy of the model.

3.6.2.3 Making Prediction

In order to simulate making prediction, the study preferred the previous section trained model to evaluate its accuracy using unseen dataset. In order to evaluate this process the study uses a Tweet Test dataset as shown in Table 11.

Table 11: Tweet Test Dataset



1	Word1	Word2	Word3	Profane
2	0.8	0.4	0.93	?
3	0.25	0.25	0.88	?
4	0.97	0.33	0.69	?
5	0.48	0.6	0.89	?
6	0.33	0.6	1	?
7	0.5	0.4	0.1	?
8	0.12	0.25	0.23	?
9	0.29	0.4	0.44	?
10	33	0.33	0.11	?
11	0.35	0.4	0.34	?

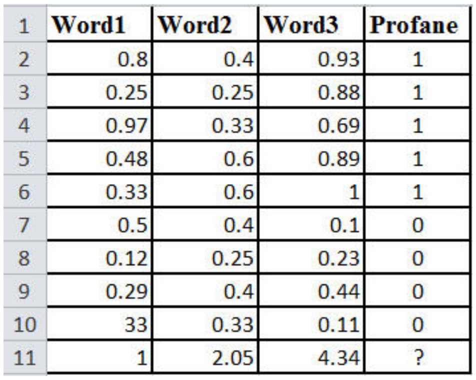
Therefore, using the new coefficient calculated from the previous section of: b0 = 0.0375, b1 = 0.0225, b2 = 0.0150 and b3 = 0.0375, it is possible to calculate the prediction values for each instances of the unseen test dataset. To simulate this process, the study calculates the prediction class of the first instance: word1 = 0.80, word2 = 0.40 and word3 = 0.93 of the unseen test dataset as follows:

- Prediction = 1 / (1 + e - (b0 + b1x1 + b2x2 + b3x3))
- Prediction = 1 / (1 + e - (0.0375 + 0.0225*0.80 + 0.0150*0.40 + 0.0375*0.93))
- Prediction = 1 / (1 + e - (0.096375))
- Prediction = 1 / (1 + 0.908) = 0.524

In order to interpret these results, the prediction probabilities values can be converted into crisp class value by bounding them using the following expression: Prediction = IF (output > 0.5) Then 1 Else 0

Therefore, the first instance of the Tweet profane word test dataset can be predicted to belong to class 1 according to the trained model. In practice, the above process is repeated for each instance in the predicted profane word dataset by calculating the values of each instance to predict the class. The output of this process can be simulated and presented as shown in Table 12.

Table 12: Predicted Tweet Test Dataset



1	Word1	Word2	Word3	Profane
2	0.8	0.4	0.93	1
3	0.25	0.25	0.88	1
4	0.97	0.33	0.69	1
5	0.48	0.6	0.89	1
6	0.33	0.6	1	1
7	0.5	0.4	0.1	0
8	0.12	0.25	0.23	0
9	0.29	0.4	0.44	0
10	33	0.33	0.11	0
11	1	2.05	4.34	?

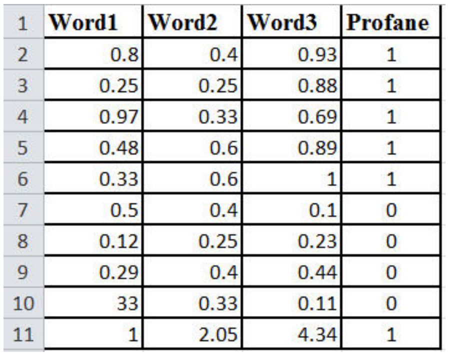
Finally, in order to simulate the evaluation of the accuracy of the trained model the study calculates the accuracy of the model as follows:

- Accuracy = (correct predictions / # predictions made) * 100
- Accuracy = (9 /10) * 100
- Accuracy = 90%

3.6.3 Crowd-sourcing for Classification

It was asserted by [13], that crowd-sourcing is used to manually remove outliers by classifying any misclassified instances. Using the previous section classified predicted profane test data, the human judge in crowd-sourcing can be simulated by labeling the misclassified instance outlier of Kiswahili or Sheng (Kiswahili slang) words: “buda” (father) numerized as 1.00, “kuma” (pussy) numerized as 2.05, and “poa” (cool) numerized as 4.34 to belong to class 1, as shown in Table 13 of Crowdsource Tweet Test dataset.

Table 13: Crowd-sourced Tweet Test Dataset



1	Word1	Word2	Word3	Profane
2	0.8	0.4	0.93	1
3	0.25	0.25	0.88	1
4	0.97	0.33	0.69	1
5	0.48	0.6	0.89	1
6	0.33	0.6	1	1
7	0.5	0.4	0.1	0
8	0.12	0.25	0.23	0
9	0.29	0.4	0.44	0
10	33	0.33	0.11	0
11	1	2.05	4.34	1

Therefore, combining classification and crowd-sourcing generated results will improve results. This is evident, when the accuracy of the crowd-sourced tweet test dataset is calculated as follows:

- Accuracy = (correct predictions / # predictions made) * 100
- Accuracy = (10 /10) * 100
- Accuracy = 100%

3.7 Ensemble Method Selection & Performance Evaluation

It was explained by [2], that a single model evaluation involves assessing performance measurements of individual algorithms using metrics such as accuracy, Kappa, and F-Measure against baseline algorithm. Baseline algorithm classifications such as base rate, random rate and null rate are prerequisite for evaluation of optimal algorithm from a suite of algorithm for a particular domain problem. The baseline classification defines a minimal level of performance from which other machine learning algorithms are compared during evaluation. In order to simulate the proposed model evaluation,

Therefore, in order to simulate single model evaluation, the study preferred the base rate of Zero R classifier algorithm as the benchmark and accuracy as the performance metric to demonstrate the theories behind individual model evaluation.

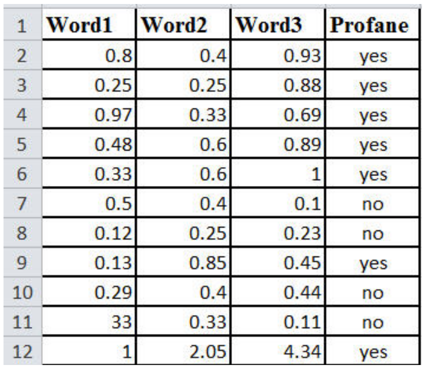
3.7.1 Zero R Classifier Benchmark

The Zero R classifier is a rule-based algorithm with null rules using base rate. It predicts the majority target class and ignores all predictor attributes. It does not use a model for prediction but only its memory. Hence, it is useful for determining a baseline performance as a benchmark for other machine learning algorithms.

3.7.1.1 Zero R Classifier Prediction

The Zero R classifier simply assigns every value to the most common class by examining the training dataset. Thereafter, it calculates the mean of the matched classes to make prediction by selecting the largest mean value. In order to simulate the theory behind the Zero R classifier, the study preferred the Numerized Tweet dataset2 in Table 14.

Table 14: Numerized Tweet Dataset2



1	Word1	Word2	Word3	Profane
2	0.8	0.4	0.93	yes
3	0.25	0.25	0.88	yes
4	0.97	0.33	0.69	yes
5	0.48	0.6	0.89	yes
6	0.33	0.6	1	yes
7	0.5	0.4	0.1	no
8	0.12	0.25	0.23	no
9	0.13	0.85	0.45	yes
10	0.29	0.4	0.44	no
11	33	0.33	0.11	no
12	1	2.05	4.34	yes

The means of the target class Profane of the Numerized Tweet dataset2 of 11 instances are calculate as follows:

- Mean Profane (yes) = 7/11 = 0.636
- Mean Profane (no) = 4/11 = 0.364

Therefore, the Zero R classifier will predict “yes” of the target class profane, since its mean is the largest compared to “no” of the profane class.

3.7.2 Accuracy Performance Metric

The accuracy performance evaluation involved assessing a suite of algorithms against the benchmark algorithm with respect to accuracy performance metric. The accuracy performance is defined by the confusion matrix for binary classification as: Accuracy = {tp+tn}/{tp+fp+fn+tn}

Where, tp are true positive, fp – false positive, fn – false negative, and tn – true negative counts.

Hence, calculating accuracy the Zero R classifier in the previous section will be as follows: Accuracy = {7/11} * 100 = 63.6%

Therefore, in order to simulate evaluation process, the study preferred the following:

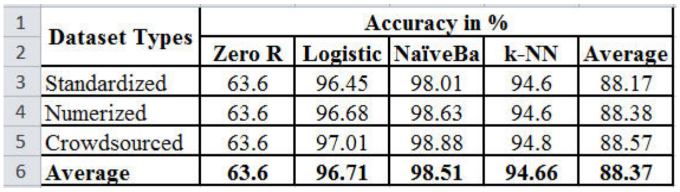
1. Three Tweet dataset types: Standardized Dataset, Numerized Dataset, and Crowd-sourced Dataset
2. Three algorithms of Tweet performance values for each dataset type: logistic, Naive Bayes and k-NN
3. The baseline algorithm as the Zero R with accuracy value of 63.6% for all the three dataset types

The objective of these assumptions was to compute an evaluation matrix consisting of Tweet dataset types, suite of algorithms and baseline algorithm to select the optimal algorithm suitable for the domain problem.

3.7.2.1 Accuracy Evaluation Matrix

The objective of accuracy evaluation matrix was to assess the accuracy of each algorithm against each dataset type with respect to the baseline algorithm. The study preferred Logistics values as (96.45, 96.68, and 97.01), Naïve Bayes (98.01, 98.63 and 98.88) and k-NN (94.6, 94.6 and 94.88) with respect to standardized, Numerized and Crowd-sourced dataset respectively. Thereafter, calculate the average performance for each algorithm and select the optimum algorithm for the domain problem. The result of the simulation is shown in model evaluation matrix in Table 15.

Table 15: Accuracy Evaluation Matrix for Formulated Model



1	Dataset Types	Accuracy in %				
2		Zero R	Logistic	NaïveBa	k-NN	Average
3	Standardized	63.6	96.45	98.01	94.6	88.17
4	Numerized	63.6	96.68	98.63	94.6	88.38
5	Crowdsourced	63.6	97.01	98.88	94.8	88.57
6	**Average**	**63.6**	**96.71**	**98.51**	**94.66**	**88.37**

The results of the above accuracy evaluation were used for selection of benchmarking algorithm in the ensemble model. The lowest performance in the suit of algorithm, the k-NN (94.66) algorithm was used as base-algorithm, while the logistic (96.71) and Naïve Bayes (98.51) as the Meta-algorithm in the ensemble model prediction, depending on the ensemble method used in the next section

3.8 Algorithm Tuning

It was elaborated by [14] that, the old paradigm in machine learning was to learn a single model such as Naïve Bayes, k-NN, or Logistic regression to make prediction, while the new paradigm is to learn a set of combined models called ensemble to make prediction. There are several methods of combining classification algorithm such majority votes, weighted votes or combiner function using different learning algorithm or same learning algorithm either trained in different ways or the same way. There are various models implementing these methods such as stacking model (combiner function), bagging (weighted vote) and boosting (majority votes). The objective of these models is to minimize variance (bagging), and to increase predictive power (boosting) or both (stacking).

Therefore, in order to simulate the combined model evaluation, the study preferred bagging (Bootstrap Aggregation) method to demonstrate the theories behind ensemble model

3.8.1 Bagging (Bootstrap Aggregation)

It was elaborated by [5], that bagging attempts to implement similar algorithm learners on small sample populations and then takes a mean of all the predictions. The bagging algorithm takes the original training dataset and creates multiple sub-dataset, from which it builds multiple classifiers for each sub-dataset, and then combined all the classifiers by taking the average of all to make a prediction. The output of the algorithm is expected to facilitate the reduction of variance error and hence, improved predictive power of the new model.

3.8.2 Bootstrap Method

The bootstrap is a statistical method for estimating quantity such as mean, standard deviation or learned coefficients from a data sample. For example, if the study preferred a sample of 100 values (x) and the goal is to estimate the mean of the sample. Then the mean can be directly from the sample as: Mean(x) = 1/100 * sum(x)

However, since the sample is small then, the mean will contained some error in it. Therefore, in order to improve the estimate of the mean, the study uses the bootstrap procedure:

Create many (e.g. 1000) random sub-samples of the dataset with replacement (meaning one can select the same value multiple times).

1. Calculate the mean of each sub-sample.
2. Calculate the average of all of the collected means and use that as our estimated mean for the data

3.9 Model Building & Performance Evaluation

The study used bootstrap prediction for model building & performance evaluation. In order to simulate Bootstrap prediction, the study preferred the following:

1. Create multiple dataset from profane dataset of: profane dataset1, profane dataset2 and profane dataset3
2. Thereafter, builds multiple classifiers by assuming the previous section results of individual model evaluation performances of: Logistic classifier

(96.71%), Naïve Bayes classifier (98.51%), and k-NN classifier (94.66%)

3. Finally, calculates the sample average of combined classifiers as following:

Given that: Sample average = population mean + random error

And:

- Population mean: $\mu = \sum(Xi)/N$
- Population standard deviation: $\sigma = \sqrt{\sum(Xi - \mu)2/N}$
- Standard Error of Mean: $SEM = \sigma/\sqrt{N}$

Then from the study's assumptions: Xi = {Logistic = 96.71, Naïve Bayes = 98.51, and k-NN = 94.66} and N = 3

Therefore calculating the bootstrap mean:

$\mu = \sum(96.71 + 98.51 + 94.66)/3 = 96.63$

$\sigma =$
$\sqrt{\sum[(96.71 - 96.63)2 + (98.51 - 96.63)2 + (94.66 - 96.63)2]/3}$
= 1.639

$SEM = 1.639/\sqrt{3} = 0.946$

Hence: Sample average (Bootstrap Mean) = 96.63 + 0.946 = 97.58

The simulation result of the bootstrap process is shown the Figure 5 where, there is an improved of about 1% from direct mean of 96.63% to 97.58% of bootstrap mean.

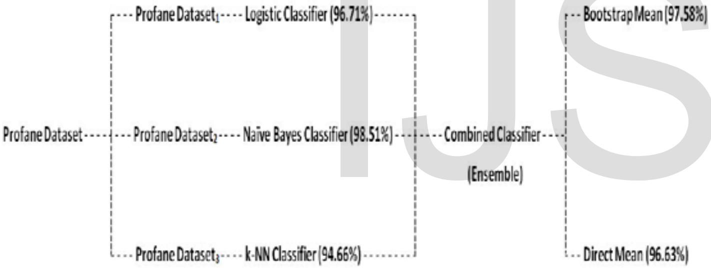


Figure 5: Bootstrap Process

4 SUMMARY & CONCLUSION

The objective of this study was to formulate a data-driven ensemble model for detection of the profane words in social media. In order to achieve this objective the study simulated the theories behind the formulated model using fictitious dataset in the context of the five stages of the proposed model to demonstrate its reliability to censor profane words in social media.

It was evident from the discussion that, all the five stages of the formulated model were grounded on mathematical theories and specifically statistics. The Data-Driven Preprocessing stage used probability theory for numerization of categorical dataset via class variable in target-based encoding. While, the Data-Driven Algorithm Selection stage, in algorithm tuning used Euclidean distance for parameterization of k-NN algorithm using various values of K, and in data exploration used statistical summaries such as min, max, IQR, Q1, and Q3 to generate boxplot and whiskers with outliers.

Moreover, the Data-Driven Classification used probability and regression equation in logistic algorithm to classify dataset. In the other side, the Single Model Evaluation used mean and percentage to calculate individual algorithm performance. Finally, the Combined Model Evaluation (Ensemble) used sampling and sample average in bootstrap for prediction of profane words

BIBLIOGRAPHY

[1]. Brownlee, J. (2014a). A Data-Driven Approach to Choosing Machine Learning Algorithms. Retrieved from http://machinelearningmastery.com/

[2]. Brownlee, J. (2014b). How To Choose The Right Test Options When Evaluating Machine Learning Algorithms. Retrieved from http://machinelearningmastery.com/how-to-choose-the-right-test-options-when-evaluating-machine-learning-algorithms/

[3]. Brownlee, J. (2015). Supervised and Unsupervised Machine Learning Algorithms. Retrieved from http://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

[4]. Brownlee, J. (2018). How to Develop a Reusable Framework to Spot-Check Algorithms in Python. Retrieved from https://machinelearningmastery.com/spot-check-machine-learning-algorithms-in-python/

[5]. Dror, G., Koren, Y., & Weimer, M. (2012). A Linear Ensemble of Individual and Blended Models for Music Rating Prediction.

[6]. Kardi, T. (2015). Similarity Measurement. Retrieved from http:\\people.revoledu.comkarditutorialSimilarity

[7]. Manson, H., & Wiggins, C. (2010a). A Taxonomy of Data Science. Retrieved from http://www.dataists.com/2010/09/a-taxonomy-of-data-science/

[8]. Manson, H., & Wiggins, C. (2010b). A Taxonomy of Data Science.

[9]. Mitchell, T. M. (2005). Generative and discriminative classifiers: Naive Bayes and logistic regression.

[10]. Refaat, M. (2017). Model Accuracy: Basic Concepts. Retrieved from http://www.angoss.com/model-accuracy-basic-concepts/

[11]. Ritter, N. (2010). Understanding a widely misunderstood statistic: Cronbach's alpha. Presented at the Educational Research Association (SERA) Conference 2010, New Orleans, LA (ED526237): SERA.

[12]. Rumsey, D. J. (2010). Statistics Essentials for Dummies (Second Edition).

[13]. Sood, S. O., Antin, J., & Churchill, E. (2012). Using Crowdsourcing to Improve Profanity Detection. Association for the Advancement of Artificial Intelligence.

[14]. Srivastava, T. (2015). Basics of Ensemble Learning Explained in Simple English.

[15]. Stapel, E. (2016). Box and Whisker Plots; Interquartile Ranges and Outliers. Retrieved from http://www.purplemath.com/modules/boxwhisk3.htm

[16]. Trochim, W. M. K. (2006). Reliability. Retrieved from http://www.socialresearchmethods.net/kb/reliable.php

[17]. Tukey, J. W. (1977). Exploratory Data Analysis.

[18]. Ursani, Z., & Corne, D. W. (2018). Use of reliability engineering concepts in machine learning for classification. Presented at the 2017 IEEE 4th International Conference on Soft Computing & Machine Intelligence (ISCMI), Mauritius: IEEE Computer Society. https://doi.org/10.1109/ISCMI.2017.8279593

[19]. Zhang, L., & Srihari, S. (2014). A Fast Algorithm for Finding k-Nearest Neighbors wiht Non-metric Dissimilarity.