Scientific Research Publishing

Journal of Data Analysis and Information Processing, 2019, 7, 153-173
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Support Vector Machine for Sentiment Analysis of Nigerian Banks Financial Tweets

Faithful Chiagoziem Onwuegbuche[1*], Joseph Muliaro Wafula[2], Joseph Kyalo Mung'atu[3]

[1]Department of Mathematics, Pan African University Institute for Basic Sciences, Technology and Innovation, Nairobi, Kenya
[2]Department of Computing, Jomo Kenyatta University of Agriculture & Technology, Nairobi, Kenya
[3]Department of Statistics & Actuarial Sciences, Jomo Kenyatta University of Agriculture & Technology, Nairobi, Kenya
Email: *faithful.onwuegbuche@gmail.com, muliaro@icsit.jkuat.ac.ke, j.mungatu@fsc.jkuat.ac.ke

How to cite this paper: Onwuegbuche, F.C., Wafula, J.M. and Mung'atu, J.K. (2019) Support Vector Machine for Sentiment Analysis of Nigerian Banks Financial Tweets. *Journal of Data Analysis and Information Processing*, **7**, 153-173.
https://doi.org/10.4236/jdaip.2019.74010

Received: July 23, 2019
Accepted: September 20, 2019
Published: September 23, 2019

Abstract

The rise of social media paves way for unprecedented benefits or risks to several organisations depending on how they adapt to its changes. This rise comes with a great challenge of gaining insights from these big data for effective and efficient decision making that can improve quality, profitability, productivity, competitiveness and customer satisfaction. Sentiment analysis is the field that is concerned with the classification and analysis of user generated text under defined polarities. Despite the upsurge of research in sentiment analysis in recent years, there is a dearth in literature on sentiment analysis applied to banks social media data and mostly on African datasets. Against this background, this study applied machine learning technique (support vector machine) for sentiment analysis of Nigerian banks twitter data within a 2-year period, from 1st January 2017 to 31st December 2018. After crawling and preprocessing of the data, LibSVM algorithm in WEKA was used to build the sentiment classification model based on the training data. The performance of this model was evaluated on a pre-labelled test dataset generated from the five banks. The results show that the accuracy of the classifier was 71.8367%. The precision for both the positive and negative classes was above 0.7, the recall for the negative class was 0.696 and that of the positive class was 0.741 which shows the prediction did better than chance in addition to other measures. Applying the model in predicting the sentiments of the five Nigerian banks twitter data reveals that the number of positive tweets within this period was slightly greater than the number of negative tweets. The scatter plots for the sentiments series indicated that, majority of the data falls between 0 and 100 sentiments per day, with few outliers above this range.

Keywords

Sentiment Analysis, Support Vector Machine (SVM), Nigerian Banks, Opinion Mining, Twitter, Social Media Analytics

1. Introduction

In the dynamic world that we now live in, the ideas, thoughts, beliefs, opinions and decisions of people are now being shared in real time on different platforms like Twitter, Facebook, LinkedIn, to name a few. However, gaining insights from these textual data for effective decision making may be a herculean task, partly due to the huge volume of information being shared from a large population and also from the complexity associated in quantifying the text data for modelling purposes. For instance, Twitter had over 310 million active users as at the first quarter of 2017 who posted over 500 million tweets per day [1]. Achieving this herculean task of gaining insights from textual data falls within the domain of sentiment analysis.

Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes [2]. Simply put, sentiment analysis is a field that is concerned with the classification and analysis of user generated text under defined polarities. There exist three major approaches to sentiment analysis, namely, machine learning approach [3] [4], lexicon-based approach [5] [6] [7] and hybrid approach [8] [9]. Machine learning approach can be further classified into supervised, unsupervised and semi-supervised machine learning. Supervised machine learning algorithms such as Support Vector Machine (SVM), Naïve Bayes Classifier, Maximum Entropy, to name a few, perform classification on the target corpus using an already labelled training data while unsupervised machine learning algorithms use unlabelled input data to find structure in the data, which is then used to determine text polarity. Lexicon-based approach utilizes dictionary based annotated corpus in classifying the polarity of a text while the hybrid approach combines both the lexicon-based approach and machine learning approach in determining the sentiment of a text.

The importance of sentiment analysis cannot be overemphasised as its application and impact span diverse fields and domains. Organisations, companies, agencies and governments can leverage on sentiment analysis in gaining insights which can enhance efficient and effective decision making. Also, by implementing sentiment analysis, organisations can take appropriate measures to ensure that they remain competitive in the market place, by determining product and services that customers are not satisfied with and improving such either through price reallocation, quality improvement or addition of new features. Similarly, in the academic domain, according to [2] there is a widespread and growing interests among researchers since the early 2000s on sentiment analysis and its applications in several fields with works like [10] [11] and later different works have been done. In fact, a review of sentiment analysis by [2] has been cited over 8000 times, this shows the growing interest and awareness of sentiment analysis.

Sentiment analysis has been shown to be beneficial in the finance sector with works such as [12] [13]. In a similar vein, [14] demonstrated that the stock market itself can be considered as a measure of social mood. [15] [16] [17] among others, have shown the promise of using sentiments in forecasting market movements. Similarly, the work of [18] applied several machine learning techniques such as Max Entropy, SVM and Naïve Bayes in the sentiment analysis of financial news articles. They also provided a database of financial news classified into positive and negative that can be used in training supervised machine learning algorithms for sentiment analysis. Other domains have also witnessed the adoption of sentiment analysis, such as, [19] [20] studies that implement sentiment analysis in learning about elections. Similarly, [21] presented a methodology to improve the quality of Twitter-based automatic polls, applying it to the prediction of municipal elections in 6 Brazilian cities. Meanwhile, [3] performed sentiment analysis on movie reviews, twitter and gold datasets using Nave Bayes, SVM and Radial Basis Function (RBF) kernel SVM. The study found that RBF kernel SVM with a modified hyper parameter value outperforms the SVM and Naïve Bayes algorithm.

However, despite the upsurge of research in sentiment analysis, there is a dearth in literature on sentiment analysis applied to banks social media data and mostly on African datasets. To the best of our knowledge, this paper is the first work to apply machine learning technique for sentiment analysis of Nigerian banks social media data. It is based on these findings that this research is carried out. This work will be of great benefit not only in expanding the domain of sentiment analysis but also be of profound help to Nigerian banks on customer intelligence and education so as to improve their satisfaction. It will also emphasis the utilization of social media analytics, in promoting their products and services, risk management, business forecasting, competitive analysis, product and service design.

The rest of the paper is structured accordingly: Section 2 presents our proposed framework for sentiment analysis, data and software used in the study. In Section 3, we present the twitter data preprocessing techniques employed in cleaning the data while in Section 4, we give a general overview of the support vector machine, the mathematical theory behind its solution, the libSVM algorithm in WEKA for its implementation and the model evaluation parameters. Section 5 presents, the results and discussions of the twitter classification model and its results when implemented in the five Nigerian Banks considered in the study. Finally, Section 6 presents the conclusion and recommendation of the study.

2. Framework

We have presented our proposed framework for sentiment analysis, illustrated in **Figure 1**. This shows the process of data gathering, using Twitter's Application Programming Interface (API), preprocessing of the gathered twitter data,

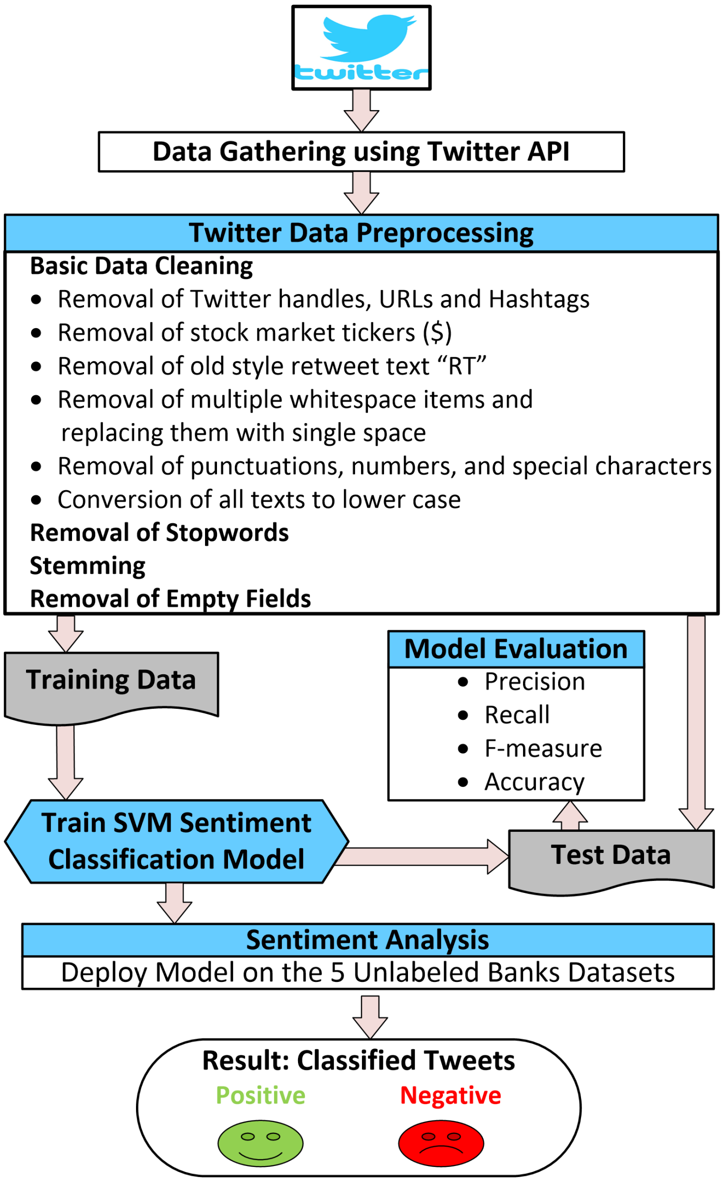
twitter
Data Gathering using Twitter API
Twitter Data Preprocessing
Basic Data Cleaning
Removal of Twitter handles, URLs and Hashtags
Removal of stock market tickers ($)
Removal of old style retweet text "RT"
Removal of multiple whitespace items and replacing them with single space
Removal of punctuations, numbers, and special characters
Conversion of all texts to lower case
Removal of Stopwords
Stemming
Removal of Empty Fields
Training Data
Model Evaluation
Precision
Recall
F-measure
Accuracy
Train SVM Sentiment Classification Model
Test Data
Sentiment Analysis
Deploy Model on the 5 Unlabeled Banks Datasets
Result: Classified Tweets
Positive
Negative

Figure 1. Proposed sentiment analysis framework.

training of the SVM classification model in Weka, evaluation of the model performance and deployment in the unlabelled Nigerian Banks twitter datasets, resulting in the classification of tweets as positive and negative.

Data Used in the Study

1) Nigerian banks twitter data

The data used for this study was retrieved from Twitter's API covering a 2-year period from 1st January 2017 to 31st December 2018. Since the data crawled were publicly available data and the usage was within the stipulated twitter data usage there was no need for ethical review. The data was retrieved from the following Nigerian banks using their specific twitter filter operators as shown in Table 1 below.

Table 1. Top five banks in Nigeria with their twitter handles and total number of crawled tweets from 1st January 2017 to 31st December 2018.

Bank Name	Stock Symbol	Filter Operator (Twitter Handle)	Number of Tweets
Access Bank	ACCESS	@myaccessbank	33,821
First Bank of Nigeria Holdings	FBNH	@FirstBankngr	51,995
Guaranty Trust Bank	GUARANTY	@gtbank	79,745
United Bank for Africa	UBA	@UBAGroup	21,048
Zenith Bank	ZENITHBANK	@ZenithBank	29,007

Table 1 shows the total number of retrieved tweets from 1st January 2017 to 31st December 2018 for the top five Nigerian Banks using their respective twitter handles. The rationale for using the five banks above was because as at June 30th 2017, they were the top five Nigerian banks when ranked by total assets [22].

2) Training data

Training Data used was the Sentiment140 dataset by [23] which can be downloaded at http://help.sentiment140.com/for-students. The Sentiment140 dataset was created by Alec Go, Richa Bhayani, and Lei Huang of Stanford University. It contains 1.6 million tweets automatically annotated as positive (800,000 tweets) and negative (800,000 tweets) using the assumption that tweets with positive emoticons are positive and tweets with negative emoticons are negative. Since training a classifier with 1.6 million tweets on a normal computer can take days before completing, after performing the preprocessing on this 1.6 million tweets dataset we then randomly select 75,000 positive and 75,000 negative tweets which forms our training dataset of 150,000 tweets.

3) Test data

The test dataset used in this study was created from the crawled Nigerian Banks Twitter datasets. We randomly select 100 tweets from each bank. Since we have five banks, this sums up our test data to 500. We then manually classified these 500 tweets into positive and negative. This test data was used to evaluate the performance of the SVM classifier, which was trained using the 150,000 tweets obtained from the Sentiment140 dataset.

4) Software used in the study

This study utilized Waikato Environment for Knowledge Analysis (WEKA) software for training the SVM classification model and deploying its results on the unlabelled datasets. WEKA is a data mining and machine learning software, developed by the University of Waikato, New Zealand. This software is used by most researchers to solve data mining and machine learning problems, because of its general public license and its graphical user interface for several functionalities such as data analysis, classification, predictive modelling and visualizations. In addition, the data preprocessing described in Section 3 was implemented in Python. Python is a very powerful open source programming language that is effective in solving diverse problems in different learning domains.

3. Twitter Data Preprocessing

In order to obtain accurate and reliable results, it is paramount that the twitter data be preprocessed so as to enable the classifier run smoothly. Data preprocessing, is a data mining technique that seeks to clean raw datasets by, removing the noise and uninformative parts from it, thereby making it suitable for data analysis with the aim of achieving accurate and reliable results.

Furthermore, [24] performed an elaborate research, to evaluate the effectiveness of different twitter data preprocessing techniques. The research found that adapting effective preprocessing of twitter data can improve system accuracy and also opined that preprocessing is the first step in sentiment analysis when given a raw datasets. In this work, we apply the following preprocessing techniques on the Twitter datasets.

3.1. Basic Data Cleaning

This phase involves removing unimportant parts from the tweets. They are tagged unimportant since they do not contain any sentiment and removing them does not change the meaning of the tweet. Basic twitter data cleaning involves removing:

1) Twitter handles such as @user,

2) Uniform Resource Locators (URLs): URLs such as http://www.twitter.com,

3) hashtags (#): in this case we remove only the hashtag symbol # and not the word, since most words after hashtags contains sentiments,

4) stock market tickers like $GE,

5) old style retweet text “RT”,

6) multiple whitespace items and replacing them with single space,

7) punctuations, numbers, and special characters, and

8) converting of all texts to lower case.

3.2. Removal of Stopwords

Stopwords are words which are used very frequently in a language and have very little meaning. They are mostly pronouns and articles, for example, words like, “is”, “and”, “was”, “the”, etc. Also, we created a stop list of frequent Nigerian stopwords such as “na”, “ooo”, “kuku”, etc. These words are filtered out from the tweets in other to save both space and time.

3.3. Stemming

Stemming is a technique that seeks to reduce a word to its word stem. For example, words like “quick”, “quickly”, “quickest”, and “quicker” are considered to be from the same stem “quick”. This helps in decreasing entropy and thereby increasing the relevance of the word. Before stemming is applied, the text has to be tokenized. Tokenization is the process of splitting a string of text into individual words.

3.4. Removal of Empty Fields

After performing the required preprocessing, some tweets may become empty. For example, if a tweet contains only URL and mention (@user). When URL and handles are removed, this tweet becomes empty. It is therefore important to remove all empty fields since an empty field contains no sentiment.

Table 2 shows the number of tweets for the different datasets before and after preprocessing. The reduction in the number of tweets is due to the removal of empty fields after applying all stated preprocessing techniques, since empty fields do not contain sentiments.

4. Support Vector Machine

Support Vector Machine (SVM) algorithm is a supervised machine learning approach that has proven to be very successful in tackling regression and classification problems. It was developed by [25].

Consider a training set $\mathcal{D}=\left\{\left(\vec{\boldsymbol{x}}_i, y_i\right) \mid \vec{\boldsymbol{x}}_i \in \mathbb{R}^n, y_i \in \{-1,1\}\right\}_{i=1}^{m}$ where $\vec{\boldsymbol{x}}_i$ is a vector and y_i is the associated class label, which can take values +1 or −1. The goal of the SVM is to find the optimal hyperplane that best separates the data between the two classes. This goal is achieved by maximizing the margin between the two classes (e.g. blue class and red class in Figure 2 below). The support vectors are the points lying on the boundaries and the middle of the margin is called the optimal separating hyperplane.

Definition 4.1 (Hyperplane). *The hyperplane is a subspace of one dimension less than its ambient space. It is a set of points satisfying the equation* $\vec{\boldsymbol{w}} \cdot \vec{\boldsymbol{x}} + b = 0$.

4.1. Formulation of the SVM Optimization Problem

From the foregoing, it has been buttressed that the goal of SVM is to find the optimal separating hyperplane that best segregates the data. Given a vector $\vec{\boldsymbol{w}}$ of any length that is constrained to be perpendicular to the median line and an unknown vector $\vec{\boldsymbol{u}}$. We are interested in knowing if $\vec{\boldsymbol{u}}$ belongs to class A or B, illustrated in Figure 2. Thus, we project $\vec{\boldsymbol{u}}$ to $\vec{\boldsymbol{w}}$ which is perpendicular to

Table 2. Number of tweets before and after preprocessing for the different datasets.

Data	Number of tweets before preprocessing	Number of tweets after preprocessing
Training	1,600,000	1,588,648
Test	500	490
ACCESS	33,821	33,046
FBNH	51,995	50,995
GUARANTY	79,745	77,999
UBA	21,048	20,395
ZENITHBANK	29,007	28,096

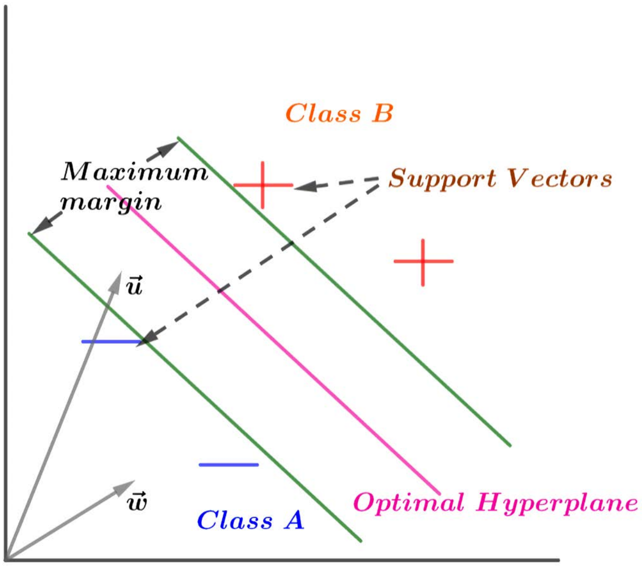

Figure 2. Illustration of the support vector machine approach.

the line and we have that $\vec{w}\cdot\vec{u}\geq c$. Where c is a constant. If $c=-b$, we can obtained the decision rule to be:

$$y_i\left(\vec{w}\cdot\vec{x}_i+b\right)\geq 1,\ \forall i \tag{1}$$

In which the variable y_i is the associated class label such that $y_i=+1$, for + samples and $y_i=-1$, for – samples.

Definition 4.2 (Point on a Hyperplane). *Given a data point* $\vec{x}_i$ *and a hyperplane defined by a vector* $\vec{w}$ *and bias b, we will get* $\vec{w}\cdot\vec{x}_i+b=0$ *if* $\vec{x}_i$ *is on the hyperplane.*

This implies that, for points on the hyperplane, Equation (1) becomes

$$y_i\left(\vec{w}\cdot\vec{x}_i+b\right)=1 \tag{2}$$

Therefore, we want the distance (margin) between the positive and negative samples to be as wide as possible.

The distance (margin) between the two support vectors (illustrated by the two green lines in **Figure 3** below) is the dot product of the difference vector and the unit vector:

$$\text{Margin}=\left(\vec{x}_+-\vec{x}_-\right)\cdot\frac{\vec{w}}{\|\vec{w}\|} \tag{3}$$

Utilizing Equation (2), by substituting for a positive sample, $y_i=1$ and negative sample, $y_i=-1$, respectively and later substituting both results in Equation (3) we obtain

$$\text{Margin}=\frac{2}{\|\vec{w}\|} \tag{4}$$

Rescaling $\vec{w}$ and b to $\text{Margin}=\frac{1}{\|\vec{w}\|}$ since it does not affect the optimization result. Therefore, the SVM optimization problem becomes:

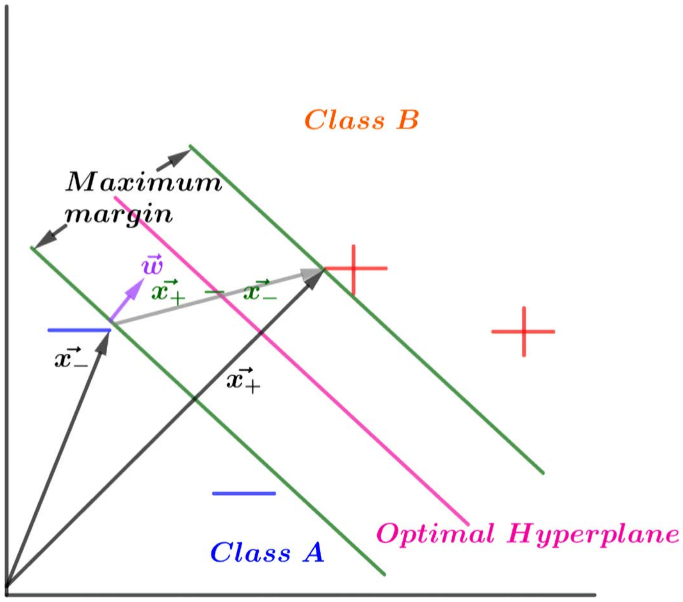

Figure 3. SVM optimization problem—distance between two support vectors.

$$\begin{aligned}&\underset{\vec{w},b}{\text{maximize}} && \frac{1}{\|\vec{w}\|}\\ &\text{subject to} && y_i\left(\vec{w}\cdot\vec{x}_i+b\right)-1\geq 0,\ i=1,\cdots,m.\end{aligned} \tag{5}$$

This maximization problem is equivalent to the following minimization problem:

$$\begin{aligned}&\underset{\vec{w},b}{\text{minimize}} && \|\vec{w}\|\\ &\text{subject to} && y_i\left(\vec{w}\cdot\vec{x}_i+b\right)-1\geq 0,\ i=1,\cdots,m.\end{aligned} \tag{6}$$

This minimization problem gives the same result as the following:

$$\begin{aligned}&\underset{\vec{w},b}{\text{minimize}} && \frac{1}{2}\|\vec{w}\|^2\\ &\text{subject to} && y_i\left(\vec{w}\cdot\vec{x}_i+b\right)-1\geq 0,\ i=1,\cdots,m.\end{aligned} \tag{7}$$

4.2. Solution of the SVM Optimization Problem

The strategy that helps in finding the local maxima and minima of a function subject to equality constraint was developed by the Italian-French mathematician Giuseppe Lodovico Langrangia, also known as Joseph-Louis Lagrange.

4.2.1. The SVM Lagrangian Problem

From Equation (7) our objective function is:

$$f\left(\vec{w}\right)=\frac{1}{2}\|\vec{w}\|^2 \tag{8}$$

and m constraint functions:

$$g_i\left(\vec{w},b\right)=y_i\left(\vec{w}\cdot\vec{x}_i+b\right)-1,\ i=1,\cdots,m. \tag{9}$$

Introducing the Lagrangian function, we have

$$\mathcal{L}(\vec{w},b,\alpha)=f(\vec{w})-\sum_{i=1}^{m}\alpha_i g_i(\vec{w},b) \tag{10}$$

$$\Rightarrow\ \mathcal{L}(\vec{w},b,\alpha)=\frac{1}{2}\|\vec{w}\|^2-\sum_{i=1}^{m}\alpha_i\left[y_i(\vec{w}\cdot\vec{x}+b)-1\right] \tag{11}$$

where α_i is a Lagrange multiplier for each constraint function.

The SVM Lagrangian problem in Equation (11) can be rewritten using the duality principle to aid its solvability.

$$\begin{aligned}&\underset{\vec{w},b}{\text{minimize}}\quad \underset{\alpha}{\max}\ \mathcal{L}(\vec{w},b,\alpha)\\ &\text{subject to}\quad \alpha_i\geq 0,\ i=1,\cdots,m.\end{aligned}$$

To obtain the solution of the primal problem, we need to solve the Lagrangian problem. The duality principle tells us that an optimization problem can be viewed from two perspectives. The first one is the primal problem, a minimization problem in our case, and the other one is a dual problem, which will be a maximization problem. Since we are solving a convex optimization problem, then Slater's condition holds for affine constraints, and Slater's theorem tells us that strong duality holds. This implies that the maximum of the dual problem is equal to the minimum of the primal problem.

The minimization problem is to be solved by taking the partial derivatives of $\mathcal{L}(\vec{w},b,\alpha)$ in Equation (11) with respect to $\vec{w}$ and *b*. Therefore, differentiating partially $\mathcal{L}(\vec{w},b,\alpha)$ with respect to $\vec{w}$ and setting it to zero we have:

$$\vec{w}=\sum_{i=1}^{m}\alpha_i y_i\vec{x} \tag{12}$$

Equation (12) shows that the vector $\vec{w}$, is a linear sum of the samples.

Also, differentiating partially $\mathcal{L}(\vec{w},b,\alpha)$ with respect to *b*, and setting it to zero we obtain:

$$\sum_{i=1}^{m}\alpha_i y_i=0 \tag{13}$$

Substitute Equation (12) into Equation (11), we have:

$$\begin{aligned}&\mathcal{L}(\vec{w},b,\alpha)\\ &=\frac{1}{2}\left(\sum_{i=1}^{m}\alpha_i y_i\vec{x}_i\right)\cdot\left(\sum_{j=1}^{m}\alpha_j y_j\vec{x}_j\right)-\sum_{i=1}^{m}\alpha_i\left[y_i\left\{\left(\sum_{j=1}^{m}\alpha_j y_j\vec{x}_j\right)\cdot\vec{x}_i+b\right\}-1\right]\\ &=\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\vec{x}_i\cdot\vec{x}_j-\sum_{i=1}^{m}\alpha_i y_i\left[\left(\sum_{j=1}^{m}\alpha_j y_j\vec{x}_j\right)\cdot\vec{x}_i+b\right]+\sum_{i=1}^{m}\alpha_i\\ &=\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\vec{x}_i\cdot\vec{x}_j-\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\vec{x}_i\cdot\vec{x}_j-b\sum_{i=1}^{m}\alpha_i y_i+\sum_{i=1}^{m}\alpha_i\end{aligned}$$

$$\mathcal{L}(\vec{w},b,\alpha)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\vec{\boldsymbol{x}}_{\boldsymbol{i}}\cdot\vec{\boldsymbol{x}}_{\boldsymbol{j}}-b\sum_{i=1}^{m}\alpha_i y_i \tag{14}$$

Substitute Equation (13) into Equation (14) gives:

$$\mathcal{L}(\vec{w},b,\alpha)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\vec{\boldsymbol{x}}_{\boldsymbol{i}}\cdot\vec{\boldsymbol{x}}_{\boldsymbol{j}} \tag{15}$$

Equation (15) is called the Wolfe Dual Lagrangian function. This shows that the optimization depends only on the dot product of pairs of samples (*i.e.* $\vec{x}_i \cdot \vec{x}_j$).

The optimization problem is now called the Wolfe Dual problem:

$$\begin{aligned} &\underset{\alpha}{\text{maximize}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j \\ &\text{subject to} \quad \alpha_i \geq 0, \text{ for any } i = 1,\cdots,m, \\ &\qquad\qquad\quad \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned} \tag{16}$$

The main advantage of the Wolfe dual problem over the Lagrangian problem is that the objective function $\vec{w}$ now depends only on the Lagrange multipliers. Also, the optimization problem depends only on the dot product of pairs of samples (*i.e.* $\vec{x}_i \cdot \vec{x}_j$). This aids computation using software.

4.2.2. KKT Optimality Condition for the SVM Optimization Solution

The Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient conditions for an optimal point of a positive definite Quadratic Programming problem. Thus, for a solution to be optimal, it has to satisfy the KKT conditions. According to [26], "solving the SVM problem is equivalent to finding a solution to the KKT conditions". Therefore, since we've solved the SVM optimization problem, by [26] we've also obtained the solution to the KKT conditions.

4.3. Soft Margin SVM

The hard margin SVM demands that the data be linearly separable. However, since in real-life data is often noisy, due to issues such as mistyped value, presence of outlier, etc. To solve this problem [27] developed a modified version of the original SVM which permits classifier to make some mistakes. Thus, the aim is not to make zero mistakes, but to make as few mistakes as possible. This modification is made possible by introducing a variable ζ. The constraint $y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$ becomes $y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \zeta_i$.

Therefore, the soft margin formulation becomes:

$$\begin{aligned} &\underset{\vec{w},b,\zeta}{\text{minimize}} \quad \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{m}\zeta_i \\ &\text{subject to} \quad y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \zeta_i, i = 1,\cdots,m, \\ &\qquad\qquad\quad \zeta_i \geq 0 \text{ for any } i = 1,\cdots,m \end{aligned}$$

The parameter C is called the SVM hyperparameter, it help us to determine how important the ζ should be since sometimes we would want to use the hard margin.

Therefore, the Wolfe dual problem is:

$$\begin{aligned} &\underset{\alpha}{\text{maximize}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j \\ &\text{subject to} \quad 0 \leq \alpha_i \leq C, \text{ for any } i = 1,\cdots,m, \\ &\qquad\qquad\quad \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned} \tag{17}$$

4.4. Non-Linearly Separable Data

When the data is not linearly separable in two dimensions and SVM is to be applied, it therefore becomes pertinent to transform the data to higher dimensions so that it can be separated. This can be done with the aid of kernel function. A kernel is a function that returns the result of a dot product performed in another space.

4.4.1. The Kernel Trick

Let $K(\vec{\boldsymbol{x}},\vec{\boldsymbol{x}}) = \vec{\boldsymbol{x}} \cdot \vec{\boldsymbol{x}}$ be a kernel, the soft-margin dual problem of 17 can be re-written as:

$$\begin{aligned} &\underset{\alpha}{\text{maximize}} && \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(\vec{\boldsymbol{x}}_i \cdot \vec{\boldsymbol{x}}_j) \\ &\text{subject to} && 0 \le \alpha_i \le C, \text{ for any } i = 1, \cdots, m, \\ & && \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned} \tag{18}$$

In Equation (18) above, C is called an SVM hyperparameter and $K(\vec{\boldsymbol{x}}_i \cdot \vec{\boldsymbol{x}}_j)$ is the kernel function, these are provided by the user; and the variables α_i are Lagrange multipliers. This change made to the dual problem is called the kernel trick. Thus, applying the kernel trick is the same as replacing the dot product of the two examples by a kernel function.

4.4.2. Types of Kernel

There are different types of Kernel that can be used to achieve the goal of the SVM optimization.

1) Linear kernel: linear kernel is known as the simplest kernel. Given two vectors $\vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{x}}'$, the linear kernel is defined as: $K(\vec{\boldsymbol{x}},\vec{\boldsymbol{x}}') = \vec{\boldsymbol{x}} \cdot \vec{\boldsymbol{x}}'$.

2) Polynomial kernel: Given two vectors $\vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{x}}'$, the polynomial kernel is defined as: $K(\vec{\boldsymbol{x}},\vec{\boldsymbol{x}}') = (\vec{\boldsymbol{x}} \cdot \vec{\boldsymbol{x}}' + c)^d$. Where $c \ge 0$ is a constant term and $d \ge 2$ represent the degree of the kernel.

3) Radial basis function (RBF) or Guassian kernel: A radial basis function is a function whose value depends only on the distance from the origin or from some point. Given two vectors $\vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{x}}'$, the RBF or Guassian kernel is defined as: $K(\vec{\boldsymbol{x}},\vec{\boldsymbol{x}}') = \exp\left(-\gamma \|\vec{\boldsymbol{x}} - \vec{\boldsymbol{x}}'\|^2\right)$. The RBF kernel returns the result of a dot product performed in $\mathbb{R}^\infty$.

4.5. LibSVM Algorithm

We utilized the LibSVM algorithm running under WEKA environment in solving the SVM optimization problem. LibSVM is a library for SVM developed by [28] of the National Taiwan University. LIBSVM implements the Sequential Minimal Optimization (SMO) algorithm for kernelize SVMs, supporting classification and regression. It is more flexible and faster when compared to the SMO algorithm invented by [29] at Microsoft Research.

4.6. Model Evaluation Parameters

After classification using SVM, it is needful to evaluate the performance of the classifier. In this work we do this by using a test data. Some of the measures used by text categorization algorithms for performance evaluation are Precision, Recall, F-measure and Accuracy. These parameters are calculated using elements of the confusion matrix or contingency table.

From Table 3, TP are positive tweets which have been correctly classified as positive and FP are positive tweets which have been misclassified as negative by the classifier. In the same vein, TN are negative tweets which have been correctly classified as negative while FN are negative tweets which have been misclassified as positive by the classifier.

With the values provided by the confusion matrix it is possible to calculate the performance evaluation parameters such as:

1) Precision: measures the exactness of the classifier with respect to each class. It is given as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{19}$$

2) Recall: measures the completeness of the classifier with respect to each class. It is given as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{20}$$

3) F-Measure: is the harmonic mean of precision and recall. It is given as:

$$\text{F-measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{21}$$

4) Accuracy: is the ratio of correctly classified example to total number of examples. It is given as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{22}$$

5) Kappa Statistics: the Cohen's Kappa coefficient measures how much better the classifier is compared with guessing with a random classifier. To do this, it compares the observed accuracy with an expected accuracy (random chance).

$$\text{Kappa Statistics} = \frac{P_o - P_e}{1 - P_e} \tag{23}$$

where $P_o = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$ is the observed accuracy and P_e = expected accuracy (which is the hypothetical probability of chance agreement).

6) Error Estimates: in general, error estimates are used to measure how close forecasts or predictions are to the eventual outcomes. Given θ as the true value,

Table 3. Confusion matrix.

	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

$\hat{\theta}$ as value estimated using the algorithm and $\bar{\theta}$ as the mean value of θ. The following error estimates are calculated as follows:

a) Mean absolute error (MAE)

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{\theta}_i - \theta_i\right| \tag{24}$$

b) Root mean square error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{\theta}_i - \theta_i\right)^2} \tag{25}$$

c) Relative absolute error (RAE)

$$\text{RAE} = \frac{\sum_{i=1}^{N}\left|\hat{\theta}_i - \theta_i\right|}{\sum_{i=1}^{N}\left|\bar{\theta}_i - \theta_i\right|} \tag{26}$$

d) Root relative square error (RRSE)

$$\text{RRSE} = \sqrt{\frac{\sum_{i=1}^{N}\left(\hat{\theta}_i - \theta_i\right)^2}{\sum_{i=1}^{N}\left(\bar{\theta}_i - \theta_i\right)^2}} \tag{27}$$

7) Matthews Correlation Coefficient (MCC): measures the quality of binary classifications. The MCC is a correlation coefficient and its value lies in the interval $-1 \le \text{MCC} \le 1$. A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and −1 indicates total disagreement between prediction and observation.

From the confusion matrix, the MCC can be calculated as:

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}} \tag{28}$$

5. Results and Discussions

From Table 4 we had a total of 490 instances (tweets) from the test data. The correctly classified instances also known as the accuracy is 352 instances (tweets) which forms 71.8367% while 138 instances (tweets) were incorrectly classified which forms 28.1633%. The Kappa statistic is a chance-corrected measure of agreement between the classifications and the true classes. It's calculated by taking the agreement expected by chance away from the observed agreement and dividing by the maximum possible agreement. Since the Kappa statistic is greater than 0 it means that the classifier is doing better than chance. The mean absolute error is used to measure how close predictions are to the eventual outcomes.

Table 5 shows that the precision of our model in classifying into both the positive and negative classes is above 0.7 which is very good. Also, the recall for the negative class is 0.696 and that of the positive class is 0.741 which are greater than 0.5 which shows the prediction did better than chance. Similarly, the F-measure scores for both classes are greater than 0.7, which shows our model

Table 4. Summary statistics of model performance on the test data.

Summary	Result
Correctly Classified Instances (Accuracy)	352 (71.8367%)
Incorrectly Classified Instances	138 (28.1633%)
Kappa statistic	0.4369
Mean absolute error	0.2816
Root mean squared error	0.5307
Relative absolute error	56.3265%
Root relative squared error	106.1381%
Total Number of Instances	490

Table 5. Performance measures on the test data.

Class	Precision	Recall	F-Measure	MCC
Negative	0.732	0.696	0.714	0.437
Positive	0.706	0.741	0.723	0.437
Weighted Avg.	0.719	0.718	0.718	0.437

performed better than chance. In a similar vein, the MCC (Matthews Correlation Coefficient) measures the quality of binary (two-class) classifications and mostly employed if the classes are of different sizes. Since our result for MCC is 0.437, which is positive and above 0, it indicates that our prediction is good.

From the foregoing, we have established that the model performed very well on our test data and hence we employed it in predicting the sentiments of the unlabelled five banks twitter data to obtain the result below.

Table 6 and Figure 4 shows the twitter sentiment analysis results for the five Nigerian banks between the period of 1st January 2017 to 31st December 2018. From the results, it can be seen that the number of positive tweets within this period is slightly greater than the number of negative tweets for each of the five banks. Based on the number of tweets crawled, GUARANTY had a greater number of tweets followed by FBNH, next to ACCESS, which was followed by ZENITHBANK and with UBA having the least number of crawled tweets within this period.

Figures 5-9 display the plots of the number of positive and negative sentiments for each bank. For each day within the period 1st January 2017 to 31st December 2018, we obtained the number of positive and negative sentiments, this data shows intuitively the distribution of sentiments for each bank within the period considered. From the plots, it can be seen that majority of the data falls between 0 and 100, with few outliers above this range.

6. Conclusions

This study proposed a framework for sentiment analysis of five Nigerian banks

Table 6. Sentiment analysis result for the five Nigerian banks.

BANK	ACCESS	FBNH	GUARANTY	UBA	ZENITHBANK
Number of Positive Sentiments	18,026	27,028	41,616	12,588	16,375
Number of Negative Sentiments	15,020	23,967	36,383	7807	11,721
Total Number of Sentiments	33,046	50,995	77,999	20,395	28,096

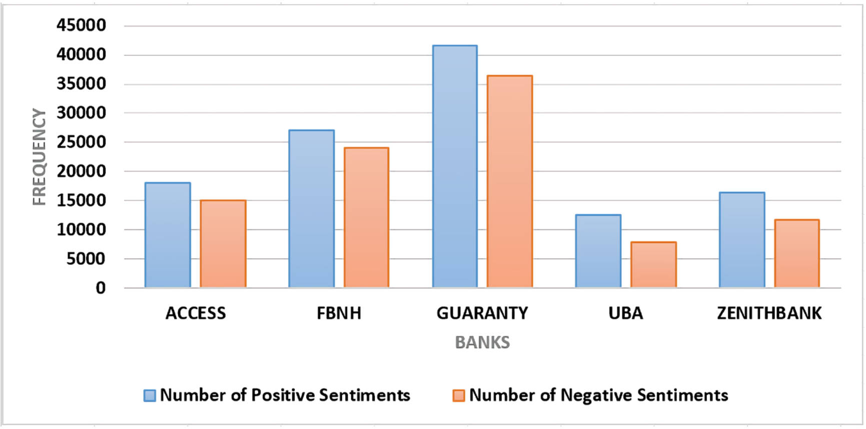

Figure 4. Number of sentiments for the different Nigerian banks.

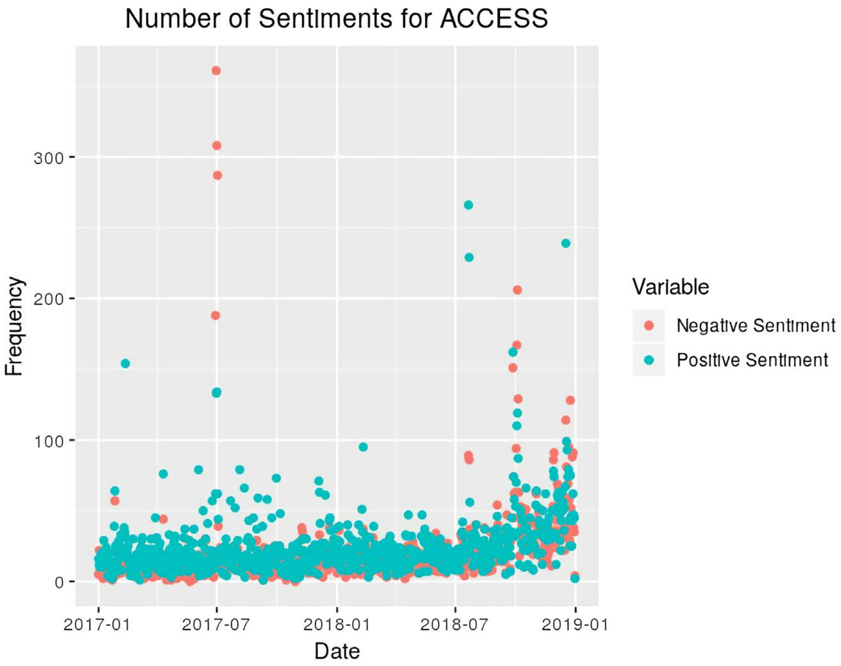

Figure 5. Number of sentiments for ACCESS.

tweets using Support Vector Machine. The data used for the study was crawled from Twitter API and spans a 2-year period, from 1st January 2017 to 31st December 2018. Based on the number of tweets crawled, GUARANTY had a greater

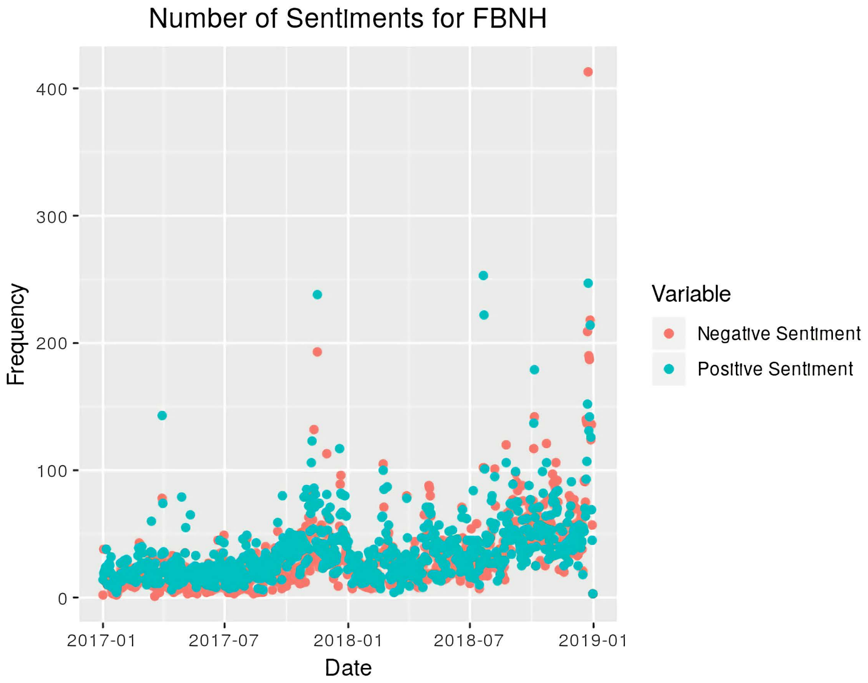

Figure 6. Number of sentiments for FBNH.

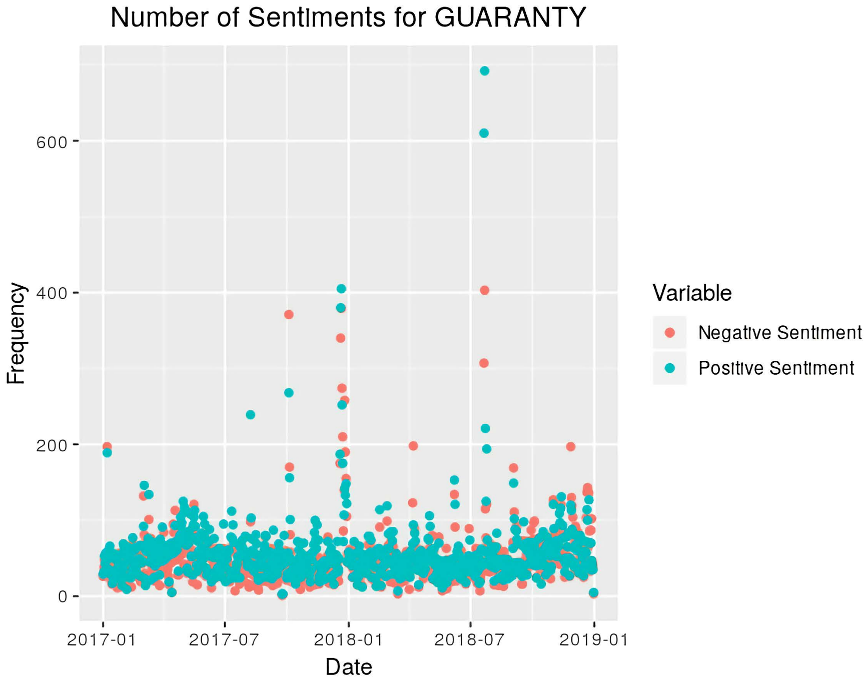

Figure 7. Number of sentiments for GUARANTY.

number of tweets followed by FBNH, next to ACCESS, which was followed by ZENITHBANK and with UBA having the least number of crawled tweets within this period. In order to obtain accurate and reliable results, and also ensure that the classifier runs smoothly, these datasets were preprocessed in Python in which twitter handles, URLs, hashtags, stock market symbols, "RT", multiple whitespaces,

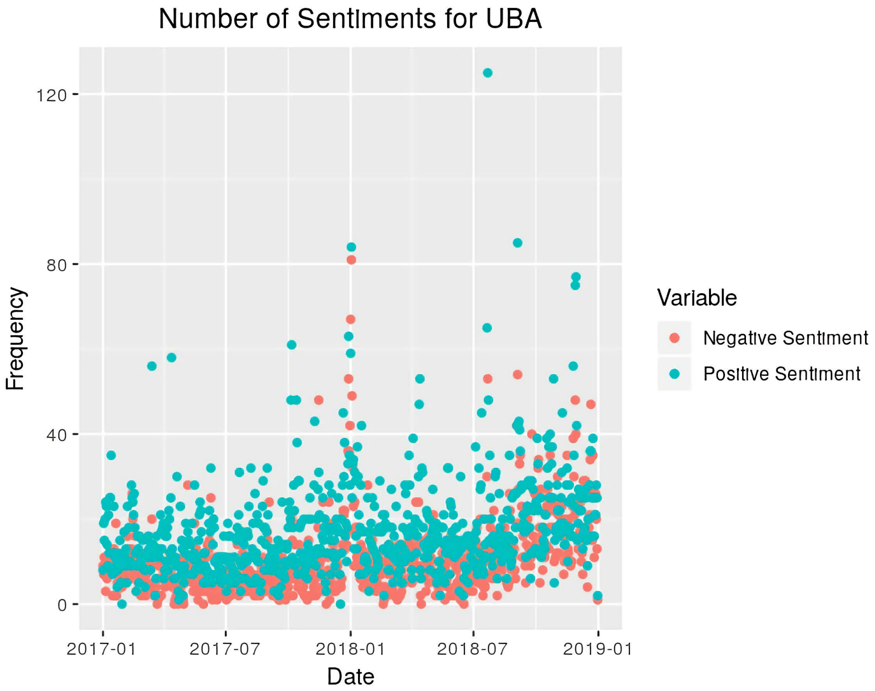

Figure 8. Number of sentiments for UBA.

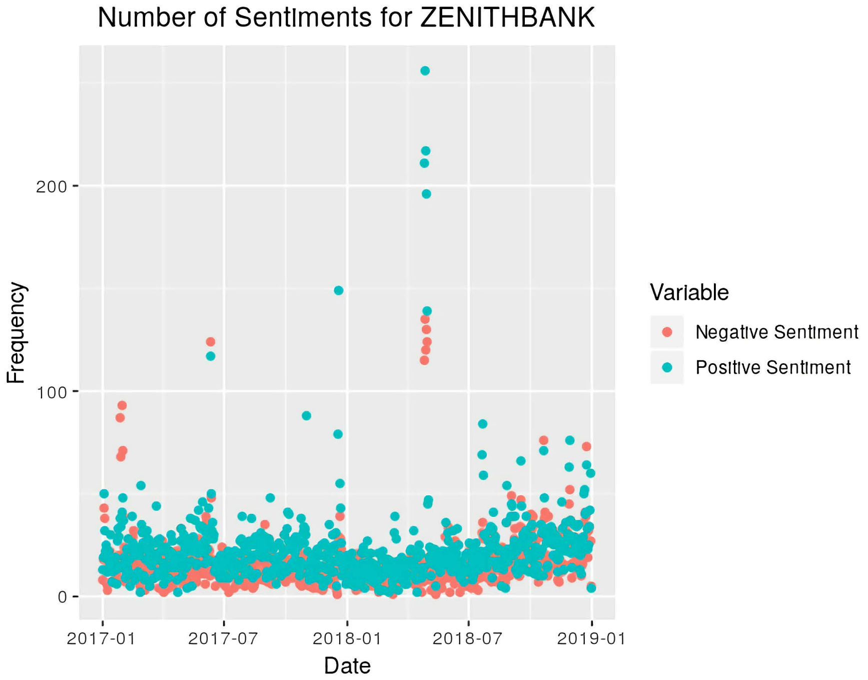

Figure 9. Number of sentiments for ZENITHBANK.

punctuations, numbers, and special characters were all removed. Similarly, stop-words were removed, and the tweets were stemmed so as to decrease the entropy of a word. Thereafter, empty fields were removed.

After preprocessing of the training, test and five banks datasets, LibSVM algorithm in WEKA was used to build the sentiment classification model based on the training data. The performance of this model was evaluated on a pre-labelled test dataset generated from the five banks. Our results show that the accuracy of the classifier was 71.8367%, the Kappa statistics was greater than 0 and implies that the classifier performed better than chance. The precision for both the positive and negative classes was above 0.7 which is very good. Also, the recall for the negative class is 0.696 and that of the positive class is 0.741 which are greater than 0.5 which shows the prediction did better than chance. Similarly, the F-measure scores for both classes are greater than 0.7, which shows our model performed better than chance. Since our result for Matthews Correlation Coefficient (MCC) is 0.437, which is positive and above 0, it indicates that our prediction is good.

Since the model performed very well on the test data, it was deployed in predicting the sentiments of the five Nigerian banks twitter data. Our results show that the number of positive tweets within this period was slightly greater than the number of negative tweets for each of the five banks. Plots for the sentiments series indicated that, majority of the data falls between 0 and 100 sentiments per day, with few outliers above this range.

This research will assist the Nigerian banks in better understanding their customers and foster risk management, business forecasting, competitive analysis, products and services improvements. Future studies can use several machine learning techniques and compare their performance on the datasets to ascertain the best classifier. Another research direction will be classifying the tweets into other polarities other than positive and negative.

Acknowledgements

The first author acknowledges the African Union Commission for the research grant to carry out this research under the Pan African University Scholarship Scheme.

Conflicts of Interest

The authors have no competing interests to declare.

References

[1] Statista (2017) Number of Monthly Active Twitter Users Worldwide from 1st Quarter 2010 to 1st Quarter 2019 (in Millions). https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[2] Pang, B., Lee, L., *et al.* (2008) Opinion Mining and Sentiment Analysis. *Foundations and Trends® in Information Retrieval*, **2**, 1-135. https://doi.org/10.1561/1500000011

[3] Jadav, B.M. and Vaghela, V.B. (2016) Sentiment Analysis Using Support Vector Machine Based on Feature Selection and Semantic Analysis. *International Journal of Computer Applications*, **146**, 26-30. https://doi.org/10.5120/ijca2016910921

[4] Tripathy, A., Agrawal, A. and Rath, S.K. (2016) Classification of Sentiment Reviews

Using n-Gram Machine Learning Approach. *Expert Systems with Applications*, **57**, 117-126.

[5] Taboada, M., Brooke, J., Tofiloski, M., Voll, K. and Stede, M. (2011) Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, **37**, 267-307. https://doi.org/10.1162/COLI_a_00049

[6] Al-Ayyoub, M., Essa, S.B. and Alsmadi, I. (2015) Lexicon-Based Sentiment Analysis of Arabic Tweets. *International Journal of Social Network Mining*, **2**, 101-114. https://doi.org/10.1504/IJSNM.2015.072280

[7] Khoo, C.S. and Johnkhan, S.B. (2018) Lexicon-Based Sentiment Analysis: Comparative Evaluation of Six Sentiment Lexicons. *Journal of Information Science*, **44**, 491-511. https://doi.org/10.1177/0165551517703514

[8] Prabowo, R. and Thelwall, M. (2009) Sentiment Analysis: A Combined Approach. *Journal of Informetrics*, **3**, 143-157. https://doi.org/10.1016/j.joi.2009.01.003

[9] Appel, O., Chiclana, F., Carter, J. and Fujita, H. (2016) A Hybrid Approach to the Sentiment Analysis Problem at the Sentence Level. *Knowledge-Based Systems*, **108**, 110-124. https://doi.org/10.1016/j.knosys.2016.05.040

[10] Cardie, C., Wiebe, J., Wilson, T. and Litman, D.J. (2003) Combining Low-Level and Summary Representations of Opinions for Multi-Perspective Question Answering. In: *New Directions in Question Answering*, 20-27.

[11] Dave, K., Lawrence, S. and Pennock, D.M. (2003) Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In: *Proceedings of the* 12*th International Conference on World Wide Web*, ACM, New York, 519-528. https://doi.org/10.1145/775152.775226

[12] Mishne, G. and Glance, N.S. (2006) Predicting Movie Sales from Blogger Sentiment. *AAAI Spring Symposium*: *Computational Approaches to Analyzing Weblogs*, 155-158.

[13] Tetlock, P.C. (2007) Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *The Journal of Finance*, **62**, 1139-1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x

[14] Nofsinger, J.R. (2005) Social Mood and Financial Economics. *The Journal of Behavioral Finance*, **6**, 144-160. https://doi.org/10.1207/s15427579jpfm0603_4

[15] Bollen, J., Mao, H. and Zeng, X. (2011) Twitter Mood Predicts the Stock Market. *Journal of Computational Science*, **2**, 1-8. https://doi.org/10.1016/j.jocs.2010.12.007

[16] Nofer, M. and Hinz, O. (2015) Using Twitter to Predict the Stock Market. *Business & Information Systems Engineering*, **57**, 229-242. https://doi.org/10.1007/s12599-015-0390-4

[17] Nisar, T.M. and Yeung, M. (2018) Twitter as a Tool for Forecasting Stock Market Movements: A Shortwindow Event Study. *The Journal of Finance and Data Science*, **4**, 101-119. https://doi.org/10.1016/j.jfds.2017.11.002

[18] Agaian, S. and Kolm, P. (2017) Financial Sentiment Analysis Using Machine Learning Techniques. *International Journal of Investment Management and Financial Innovations*, **3**, 1-9.

[19] Dokoohaki, N., Zikou, F., Gillblad, D. and Matskin, M. (2015) Predicting Swedish elections with Twitter: A Case for Stochastic Link Structure Analysis. 2015 *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Paris, 25-28 August 2015, 1269-1276. https://doi.org/10.1145/2808797.2808915

[20] Hasan, A., Moin, S., Karim, A. and Shamshirband, S. (2018) Machine Learning-Based

Sentiment Analysis for Twitter Accounts. *Mathematical and Computational Applications*, **23**, 11. https://doi.org/10.3390/mca23010011

[21] Almeida, J.M., Pappa, G.L., *et al.* (2015) Twitter Population Sample Bias and Its Impact on Predictive Outcomes: A Case Study on Elections. *Proceedings of the* 2015 *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Paris, 25-28 August 2015, 1254-1261. https://doi.org/10.1145/2808797.2809328

[22] Relbanks (2017) Top Ten Banks in Nigeria Ranked by Their Total Assets. https://www.relbanks.com/africa/nigeria

[23] Go, A., Bhayani, R. and Huang, L. (2009) Twitter Sentiment Classification Using Distant Supervision. CS224N Project Report.

[24] Angiani, G., Ferrari, L., Fontanini, T., Fornacciari, P., Iotti, E., Magliani, F. and Manicardi, S. (2016) A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter. KDWeb.

[25] Vapnik, V.N. (1995) The Nature of Statistical Learning Theory. Springer-Verlag, Berlin. https://doi.org/10.1007/978-1-4757-2440-0

[26] Burges, C.J. (1998) A Tutorial on Support Vector Machine for Pattern Recognition. *Data Mining and Knowledge Discovery*, **2**, 955-974. https://doi.org/10.1023/A:1009715923555

[27] Cortes, C. and Vapnik, V. (1995) Support-Vector Networks. *Machine Learning*, **20**, 273-297. https://doi.org/10.1007/BF00994018

[28] Chang, C.-C. and Lin, C.-J. (2011) LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, **2**, Article No. 27. https://doi.org/10.1145/1961189.1961199
https://dl.acm.org/citation.cfm?doid=1961189.1961199

[29] Platt, J. (1998) Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines.