Sentiment Analysis on Mobile Operators' Customer Review

by

Mohammed Mohibul Hossain
15301057
Md. Mostafizur Rahman
15301069
Samin Yeasar Seaum
15301105
Akil Ahmed
15201039

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
December 2019

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Mohammed Mohibul Hossain 15301057	Md. Mostafizur Rahman 15301069
Samin Yeasar Seaum 15301105	Akil Ahmed 15201039

Approval

The thesis titled "Sentiment Analysis on Mobile Operators' Customer Review" submitted by

1. Mohammed Mohibul Hossain (15301057)
2. Md. Mostafizur Rahman(15301069)
3. Samin Yeasar Seaum(15301105)
4. Akil Ahmed(15201039)

Of Fall, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on December 24, 2019.

Examining Committee:

Supervisor:
(Member)

Hossain Arif
Assistant Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Mahbub Alam Majumdar
Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Mobile is a great invention for the global village. Now, people from different corners of the world are interconnected over this device. Therefore, in Bangladesh, there are different telecommunication companies have emerged. They are competing among them for providing a better experience for their users. However, over the past few years, for the significant development of technology and software, we have got many advantages from this mobile phone using the internet. We live in a time when emotions are high because they're publicly shared, and displayed, in social media [1]. Among all of the social network sites, Facebook is very popular where people pass their time most. Researchers said on average a person spends 38 minutes every day on Facebook [2]. Telecommunication companies have opened their Facebook pages for providing a better experience to the customers. They post different types of services, packages and offers on their pages. On these Facebook pages, customers share their suggestions, complains and reactions on every post. Our research purpose is to analyze the sentiment of customers from these comments. We are analyzing whether a comment can be positive, negative or neutral. We have scrapped and labeled around 7 thousand comments ourselves from different verified pages of top telecommunication companies in Bangladesh. We use these as our dataset. In addition to that, we labeled each comment by positive, negative and neutral. Then, we have implemented Support Vector Machine (SVM), Naïve Bayes, Decision Tree and K-Nearest Neighbor (KNN) algorithms for getting our results. After implementing those algorithms, when we keep all language's comments in one dataset, we got 73% accuracy for SVM, 57.3% accuracy for KNN, 63.61% accuracy for Decision Tree and 64.11% accuracy for Naïve Bayes algorithm. In the same way, when we have separated Bangla language dataset, we got 70.65%, 65.58%, 55.07% and 56.70% accuracy respectively for SVM, Naïve Bayes, Decision Tree and KNN algorithms. In addition to that, we also have separated our Banglish (Bangla words typed using English letters) dataset and got 74%, 60.29%, 71.23% and 61.30% accuracy respectively for SVM, Naïve Bayes, Decision Tree and KNN algorithms. We do believe that our research has a great impact on the business of telecommunication companies. Through our research, a telecommunication company can understand customer sentiment on different new products, projects, and offers which is very important in the era of business competition. As a result, they can achieve the highest success through the highest client satisfaction.

Keywords: Sentiment Analysis, Customer Review Analysis, Machine Learning, Naive Bayes, K-nearest Neighbors, Support Vector Machine, Decision Tree.

Dedication

We are grateful to our parents for their countless sacrifices and utmost care. Without their guidance and prays we would have been lost. This book is dedicated in a heartfelt way to our parents, who deserve all the respect and honour for any of our success in life.

Acknowledgement

First of all, we are thanking almighty Allah for enabling us to complete our thesis. After that, we want to express our deepest love and respect to our supervisor Hossain Arif for introducing us with big data and machine learning algorithms. Without his constant support and direction, it was quite impossible to complete our thesis on time. His excellent supervision made our work easier. At the same time, we would like to thank the department of computer science and engineering of BRAC University for providing us with an excellent thesis lab where we got high configuration computer that accelerated our progress of thesis work. Moreover, we are grateful to our parents for providing their effort in every single day. Without their care and support, we would not able to complete this. Finally, we are expressing our thanks to our friends for sharing knowledge with us. They made it easy to learn different algorithms and visualize our work more clearly.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

Banglish Bangla words pronounced in English

KNN K-Nearest Neighbors

SVM Support Vector Machine

Chapter 1

Introduction

1.1 Thesis Overview

The appreciation of business organizations on the basis of the customers' feedback is becoming greater nowadays. The business organizations are now getting more aware on their customer satisfaction. Telecom operators often offer different types of packages for their customers and it becomes easier to update, change or bring new packages if they can have their customer review on their services. A big place to know the thoughts of the customers about their products or offer is the social media. In social media such as Facebook, Twitter and so forth people usually share their ideas and opinions which are also consist of huge amount of data about discussion of the issue. People usually used to watch or read review and buy the service. With the easy access of social media people nowadays also give review, complain or discuss about the service on the social networking groups [3]. This represents the social media phenomenon, which can significantly impact a firm's sales, reputation and even survival. In this research, we will build a framework which will be able to analyze the customer review on the telecom operators' services.

1.2 Thesis Motivation

Recently, customer active engagement in various telecommunication company has received a great attention. Customer engagement is generally understood as a process that effectively assembles, keeps up and manages relationships with customers. It becomes easier for telecom companies to make their firm top ranked by making their services customer friendly. Thus, it is well adopted by business intelligence of the firm to manage their customer engagement. Although many companies are not familiar with sentimental analysis processes in social media, some companies seem to be competing to engage consumers in corporate social media [4]. A successful organization must have the ability to process all available information such as customers' opinion, service prices from competitors and review of services to identify what has happened and predict what will happen in the near future [4]. A sentimental analysis using customers' discussion on social media can make the process much easier. This analysis will help to easily and successfully utilize customer need by understanding customers' positive, neutral and negative review.

1.3 Thesis Orientation

The rest of our research paper has been organized in the following way: Chapter 2 includes the literature review related to sentiment analysis on dierent social media. In addition, chapter 3 presents the methodology of the research which includes discussion about datasets, data scrapping and a brief description of our used algorithms. Then, in chapter 4, we have discussed how our used algorithms have been implemented and how we preprocessed our dataset with a brief description. Moreover, our experimental results, analysis with graphical representation have been shown in chapter 5. Finally, in Chapter 6 we have concluded the research paper by describing our future plans of extending the research.

Chapter 2

Literature Review

Several types of research have been done on sentiment analysis of social media data so far as millions of people share their thoughts and opinion on social media such as Twitter and Facebook. A machine learning based approach is simpler than knowledge based approach as the machine learning based approach doesn't need predefined database of entire emotions. Neethu and Rajasree in their research on sentiment analysis in tweeter worked with different machine learning algorithms [5]. First of all, they created a dataset collecting tweets using tweeter API. Then they preprocessed the dataset to avoid slang words and misspelling. Here they created a dictionary for the slang words with their meanings as the slang words should not be removed. The preprocessed dataset was used in creation of feature extraction where the emotion was given weight for separating the positive and negative words. After creating the feature vector, they used Naïve Bayes, SVM, Maximum Entropy and Ensemble Classifier where they found almost similar performance.

Two different frameworks was used to manage the huge dataset for sentiment analysis in a research paper [6]. Parveen and Pandey used Hadoop and MapReduce architecture to manage their huge data which was collected from Tweeter using tweeter API. In their dataset they also worked with emoticons. They have done sentiment analysis on feedback, reviews and comments which will help to provide some prediction on business intelligence. However, they used Naïve Bayes classifier with two different dataset. One dataset was consist of data with emoticons and another was without emoticons. Comparing the accuracy they found that the result was more accurate when the emoticons was used than the dataset without emoticons.
Troussas et al. introduced how sentiment analysis can help language learning, by invigorating the instructive procedure and exploratory outcomes on the Naive Bayes Classifier [7]. They have built up a framework that can arrange a conclusion utilizing sentence-level classification whether it involves positive, neutral or negative feelings. Their dataset was designed with 7000 status update from 90 users. The dataset was then manually labeled as positive, neutral or negative. Their main classifier was the Naïve Bayes classifier. However, the also worked with Rocchio classifier and Perception classifier. They divided their dataset into 50%-50% for training and testing part. Their experimental result shows that the Naïve Bayes and the Rocchio have about same accuracy and they give more accurate result than the Perception classifier.

Kaur, Mangat, and Nidhi in their research on sentiment analysis techniques introduced us to subjectivity, objectivity, and polarity [8]. Subjectivity is where we get some adjectives. On the other hand, objectivity is a normal statement. According to the paper, sentiment analysis can in sentence level, phrase level and document level. The sentiment analysis is used in the business application, support in decisions and prediction and trend analysis. In the feature selection method, an n-gram is used where it denotes consecutive n terms in the text. The next step was splitting the word according to parts of speech. After that, they have removed the prefix and suffix and the stop words from the text. In the sentiment classification, a dictionary-based approach can be used.

Communicating with friends and different types of people through social networks has become a major part of a person's daily life. Researchers have worked with mining data from shared social media posts and comments to identify the subjectivity of the sentences [9]. It helps to know whether the sentence reflects any emotional state of the writer or there is no emotional feelings in the sentence. Lexicon development deals with informal language where the lexicons have to be developed for emoticons, social acronyms, and expressions in unfamiliar languages. To make the data readable for machine learning algorithms data preprocessing is necessary as it removes redundant attributes and also normalizes data using max-min. To classify subjectivity into three categories such as neutral, moderately subjective and subjective, a training model has been created. This framework also predicts online friendship strength by using the SVM algorithm.

There are different techniques for sentiment analysis. For example; Linguistic inquiry and word count (LIWC), affective norms for English words (ANEW), Fuzzy logic, Emotion theories. This paper used the fuzzy inference technique. The inference technique had a linguistic processor that minimized semantic ambiguity [10]. It was combined with multi-source lexicon integration that derives dominant valance (positive, negative and neutral) and also prominent emotions (e.g., anger, sadness, anxiety, satisfaction, happiness, excitement). The "Data Collector" has been used to scrape the data. Meaningless data has been removed using the "Noise Filter".

A knowledge-based tool that does not require training is used in the research for analyzing data and modeling emotions in social media [11]. In this paper, they have present a generic framework for the analysis of big social data and the modeling of public emotions and mood. The framework consists of two main parts. Initially, an ensemble classifier schema, which combines a generic domain-independent, knowledge-based tool with machine learning methods such as Naïve Bayes, SVM, and Maximum Entropy is used to recognize emotional content in user-generated social data. A graph-based method is utilized to model the emotional level of a topic based on the emotions recognized and then it visualize the emotional graph of public emotions.

Besides categorizing the emotions, research has developed to measure the emotion intensity from the data of social media [12]. To compare sentences in the same emotion category there is a need of emotional intensity. Here for feature extraction, WordNet files for different emotions are used with a list of negative words and manually annotated enhancing/diminishing word list. Data preprocessing is done through the lower casing all words, removing hashtags, correcting letters and replacing user mention in @user. The Linear Regression algorithm is used for predicting the intensity of the sentences.

There is an enormous development in the generation of review information online [13]. We have now started to consider the internet as a big source of opinion evaluation. In this internet-based world, we choose anything by observing review, stars, rating. Mumtaz, and Ahuja in their research paper show us the sentiment analysis on movie review data. They show different levels of opinion mining. They describe document, sentence, and entity level opinion mining. However, the writers mention that data set from can be gathered from different blogs, social network sites, and review sites. In the machine learning approach, they use the Naive Bayes classifier algorithm. This is efficient for the large dataset. However, in the machine approach, the training dataset is difficult to obtain. Moreover, in the dictionary based approach, they collect some words with the positive, negative and neutral polarity. They collect the dataset of reviews and pre-processed the dataset. After performing the lexicon algorithm, they got their result. In the pre-processing, they remove URL, symbol, and hyperlinks. However, they got 70% correctness in their result.
Human decisions influenced by the assessment and judgment of others [14]. People observe the reaction of others when they buy any products. Bakshi, Navneet, Ravneet, and Gurpreet explained in their paper about opinion mining and sentiment analysis. Like other works, they worked on tweets. After extracting the tweets, they pre-processed it. Moreover, their sentiment analysis is dictionary based. By the pre-processed data they analyze the sentiment and give the sentiment score. The words of the dictionary have been categorized by positive words, negative words and neutral words. They calculate their output by N*(Positive word) - N*(Negative word)/ N*(Positive word) + N*(Negative word). However, they got 80.6% accuracy after testing their dataset.
Go, Bhayani and Huang the creators have utilized distant learning to figure out how to procure sentiment data [15]. They used tweets containing positive emoji's like ":)" ":-)" as positive and negative emoji's like ":(" ":- (" as negative at the end of the sentence. They construct models utilizing Naive Bayes, MaxEnt and SVM where they found that SVM gives more accuracy than other two classifiers. For feature space, they attempt a Unigram, Bigram model in combination with grammatical features. Their experiment shows that the unigram model is better than every single other model. In particular, POS features and bigrams are not very much useful.
Zainuddin and Selamat describes performance comparison of SVM classification in sentiment analysis [16]. Compares different performance of SVM using different feature selection process. They used TFIDF, Binary Occurrences (BO) and Term Occurrences (TO) as their feature selection method. From their research they found that SVM works best when the number of feature selection is fewer. Also selecting feature based on their chi-squared statistics value helped reduce the feature selection and noise from the data.

Chapter 3

Methodology

3.1 Data Scrapping

We will need data from the comments of different mobile operators Facebook page in Bangladesh. Among these comments there are majority of two language variants. One is the Bangla language and the other one is Banglish language. The Banglish language Figure 3.1 is the written in English but for Bangla words. As there is no

Messages	lable
Amar ekta gp welcome tune code lagbe.	neu
Amar ekta gp welcome tune code lagbe.	neu
Amar ekta gp welcome tune code lagbe.pirite kan eto jala. Ei ganta poramon2 movier.kindly taratary send koren	neu
amar ekta grameen sim,,,, ek mas jabod bondho ase,,,,,,ekon oi sim e ki bondho sim er offer pabo...	neg
Amar ekta phone churi hoise but bujhte parchina kon id theke seta sim registration kora ache	neg
Amar ekta sim hariye geche. Ami sim ta tulte chai. Koto taka lagbe??	neu

Figure 3.1: Showing a sample of Banglish

good dataset that contains the type of data, we need we proceeded to scrap and label our own data. We have imported text data from a Facebook page to a spreadsheet and the tool that was used for importing data was Facepager. Facepager can collect publicly available data which has JSON based APIs. This tool stores data in a SQLite database which later can be exported as excel sheet, csv etc. At first, we need an access token from Facebook. This token is provided in a website called Facebook from Graph API Explorer - Facebook for Developers. After installing Facepager into our system we created file which will be used as the name for the SQLite database. Then we create a node by putting the page url and access token. Finally, we can fetch any publicly available data from that Facebook page like posts, comments, photos etc.
As we are doing our research for mobile operators in our country we will focus on the pages of some operators in Bangladesh. We scrapped comments from the official pages of three giant mobile operators in Bangladesh and they are Grameenphone,

Rabi and Banglalink. Facepager uses the ID of a page and can fetch all the comments from a specified number of pages.
First, we scrapped 40 thousand comments in total from the three different pages. Now we will label these data manually so that we can implement this data into supervised machine learning approaches. Before labeling the comments, we reduced a lot of noise from the data. There were a lot of repetitions of comments of common words like: 'wow', 'nice', 'good offer' etc. Also, there are a lot of comments containing different links of pages. Furthermore, there are a lot of comments that are only mentions to friends. These comments are noise and will reduce the accuracy of our data. So, we removed these comments. After reducing all the unnecessary comments, we were left with 13 thousand comments. We will train our model to predict three different sentiment. They are neutral, negative and positive comments. These comments are direct sentiment of people.

message	label
ki bepar ajke speed nai keno mattro 36 40 aglaki . network koi	negative
ki bolbo sotti nijeo bje uthte parce na.dhonnobad tader k jara er nepothhe kaj korecen	positive
Kobe hobe ei khelata?	neutral
Kokhon janaben akhn ki publish hoice?	neutral
Love you gp	positive
Love you ma thanks gp	positive
mb thakar por o kno net browse kora jasce na?	negative
Alhumdulillah!	positive
all the best	positive
gp ta imo pack nai?	neutral
GP ta sms kivabe kinta hoiy	neutral
Gp te 3 taka diye sms kibabe kinbo	neutral
Grameenphone ইন্টারনেট ব্যবহারে খরচ খুব বেশী। খরচ কমানোর কোনো উদ্যোগ niben ki	negative
Grameenphone নেটওয়ার্ক খুব দুর্বল	negative
Grameenphone সিম নতুন গুলাতে রিচার্জ করা যাচ্চে না কেনো?	negative

Figure 3.2: Sample of data

Figure 3.2 shows some sample data of our dataset. In our dataset we separated the scrapped comments in three sentiments positive, negative and neutral. We then manually labeled these positive, negative and neutral comments as 'positive', 'negative' and 'neutral' respectively in the dataset. Positive comments are comments that are direct compliments or indicating some strong points of an offer or network etc. People gives good review on the services and they express their happiness and satisfaction on the services with their comments. For example, in posts by the telecom operators describing the new internet offers or packages if the people thinks it a good offer or package, they post comments like "Valo offer", "Dhonnobad" which are Bengali words written using English letter. These comments reflects positive reviews by the users which has been labeled as positive feedback. Sometimes there are posts which is not related to the service but good social works by the operators. People supports these works and praise about the initiative in the comments section. For instance, "dhonnobad tader k jara er nepothhe kaj korechen", "all the best" reflects the positive support and compliments by the users of the telecom service commented on a post related to social works. These supports and compliments are also labeled as positive feedback.
However, users also give negative feedback if they are not happy with the offers or the packages. Negative comments are direct criticism of the operators or complaining

about an offer price or network etc. These comments of the users are related to the offers, services or packages provide by the telecom operators where the users find the services are inappropriate or costly. Users directly complain on the poor network or about poor services provided by the telecom operators. To illustrate, " " which is a Bangla comment says about the poor network service, "mb thakar por o kno net browse kora jasce na" which is a Bangla comment written using English letters reflects about the poor technical service provided by the telecom operator. These clearly reflects that the users are not happy with the services. So, these comments have been labeled as negative comments.
Moreover, users also give many unusual comments. These comments are neither positive nor negative, which are neutral comments. Neutral comments are comments that may be general comment or asking some general questions. The user of the services provided by the telecom operators often comments which are not related to the service or users are asking irrelevant questions in the comment section. As an example, "GP te sms kivabe kinta hoiy", "GP te imo pack nai?" are Bangla comments written using English letters. Here the users are asking for information which are not related to any service or packages review. These types of comments have been labeled as neutral comments. In a nutshell, we have manually labeled the comments according to the positive, negative or neutral meanings of the comments. After labeling all these comments we still found some duplicate or similar comments. Moreover, in our data set the number of neutral, positive and negative comments are not equal. So, this uneven number of sentiments may bias our classification. So, we further removed some recurrent comments and equalized the number of comments of all three sentiment. After that we now have 7 thousand labeled comments.
The 7 thousand data that we have has the combination of Bangla, English and Banglish. For a comparative study we also created two additional datasets. One of them contains only Bangla language and the other one contains English and Banglish both. We used these three datasets in all our classification training and predicting.

3.2 Macihne Learning

For our research, we have implemented three machine learning algorithms.

3.2.1 Naïve Bayes Classifier

A classifier is a machine learning model that is utilized to separate various objects dependent on specific features. A Naive Bayes classifier is a probabilistic machine learning model that is utilized for characterization task. The essence of the classifier is mainly based on the Bayes theorem.The formula for Bayes theorem is as follows:

$$P(\mathbf{A}|\mathbf{B}) = P(\mathbf{A})\frac{P(\mathbf{B}|\mathbf{A})}{P(\mathbf{B})} \tag{3.1}$$

Where:
P (A) = the probability of A occurring
P (B) = the probability of B occurring
P (A |B) = the probability of A given B

P (B |A) = the probability of B given A

Types of Naive Bayes Classifier:

There are three types of Naive Bayes Classifier

Multinomial Naive Bayes

This is for the most part utilized for document classification problem, whether a record has a place with the classification of sports, governmental issues, innovation and so on. The features utilized by the classifier are the recurrence of the words present in the record.

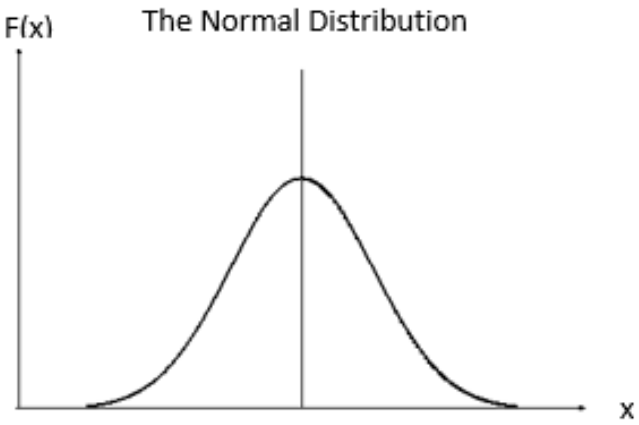

Figure 3.3: Gaussian distribution (Normal Distribution)
[17]

Bernoulli Naive Bayes

This is like the multinomial naive bayes yet the indicators are boolean factors. The parameters that we use to foresee the class variable take up just qualities yes or no, for instance if a word happens in the content or not.

Gaussian Naive Bayes

At the point when the predictors take up a nonstop worth and are not discrete, we accept that these qualities are examined from a Gaussian distribution Figure 3.3.

3.2.2 Support Vector Machine (SVM)

Support Vector Machine is a supervised machine learning algorithm that is used both to tackle regression or classification problems. They are called SVC (Support Vector Classification) and SVR (Support Vector Regression) that performs the two different challenges. SVM inherently is a binary classifier which means it will differentiate between two different classes. The fundamental principal of SVM is that first it

will plot all the data as a point on a graph corresponding to the values of the features. Then the algorithm will use different function which will depend on the characteristics of the dataset to create a hyper-plane that will divide the two classes. Thus from the position of the point with respect to the hyper-plane, the algorithm will classify a data from class A to B.

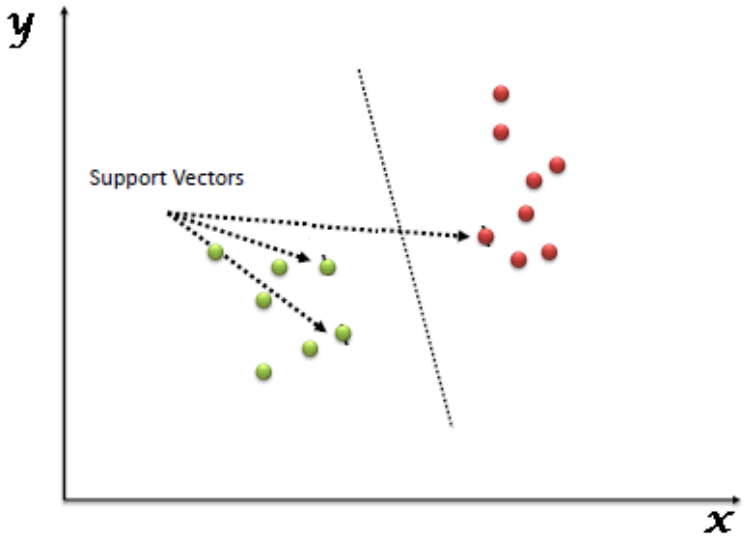

Figure 3.4: Support Vector Machine classification
[18]

SVM terminologies:

Hyper-plane, Support Vectors and Margin is explained below

Hyper-plane

Hyper-plane is a plane that linearly separates or classifies a set of data. So data at one side will be classified as Class A and the data at the opposite side of the hyper-plane will be classified as Class B. The further away from the hyper-plane a data resides the more confident the algorithm will be at classifying the data.

Support Vectors

Support Vectors are the data points that situated close to the hyper-plane. If these data points are removed it will affect the position of the hyper-plane. These support vectors are thus considered critical elements of the data set.

Margin

The distance between the hyper-plane and the closest data point to the either side of the hyper-plain is called margin. The objective is to determine the largest possible margin so that the classification can be more effective.

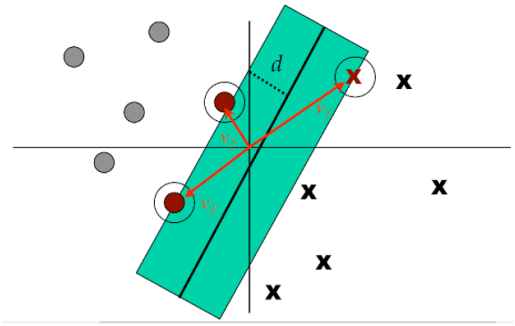

Figure 3.5: Green boundary is the margin and the red dots are support vector
[19]

Determining hyper-plane:

Goal of SVM is to determine the optimal hyperplane and to maximize the margin.

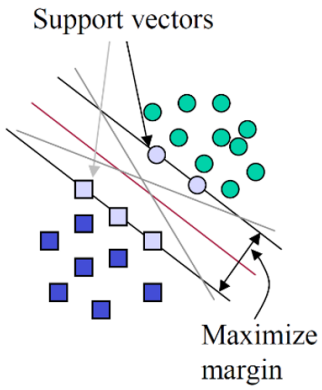

Figure 3.6: Maximizing margin
[19]

The inputs for SVM are called input sample feature x1, x2. . . xn and the output will be w. There can be n numbers of input features. Output w is an assigned weight for each feature. To optimize the algorithm we need to minimize the number of nonzero w so that the important features gets the highest priority.

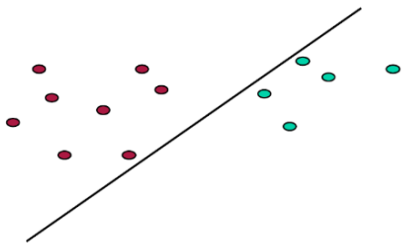

Figure 3.7: Determining hyper-plane
[19]

We determine the values for a, b, c such that:

For red points ax + by ≥ c and for green points ax + by < c. The definition of the hyper-plane is such that:

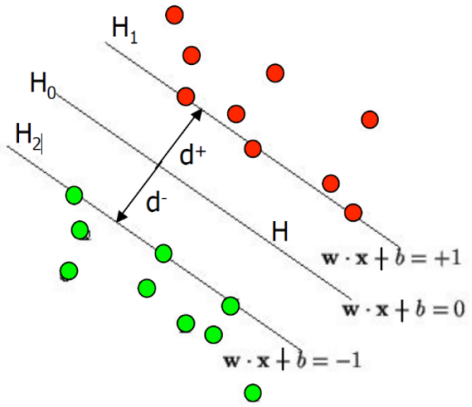

Figure 3.8: Determine the margin and the median plane
[19]

w * $x_i + b \geq +1$ when $y_i = +1$
w * $x_i + b \leq -1$ when $y_i = -1$
H_1 and H_2 are the two plains of the margin.
H_1 = w * x_i + b ≥ +1 and H_2 = w * xi + b ≤ -1.
Plain H_0 is the median where w * x_i + b = 0.
Again d+ is the shortest distance to the nearest positive point and d- the shortest distance to the nearest negative point.

3.2.3 K-Nearest Neighbors

K-Nearest Neighbor is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure [20]. It is mostly used for classification problem. However, K is the number of closest neighbor of the test case. Based on the maximum number of votes of the k-neighbor we classify the test case.
Let's say, we have two kinds of balls. For example, the yellow color ball and the purple color ball. So, here the red color ball is the test case that we will predict whether it's a yellow color ball or a purple color ball.
First of all, we will measure the distances between the test case and the all available balls. Then, we will set a value of the K for counting the vote. It can be any integer number greater than zero. So, for now let's assume it K=3. That means, we will consider top 3 closest neighbor of the "X" test case. Here, X2 refers to diameter of the balls and X1 refers to color of the balls.
After this, we will check the maximum number vote goes for which colored ball. Here, we are seeing that "X" has the maximum vote for the purple color ball. So the ball will be classified as purple color ball.
That means, we have some steps for this algorithm.

- Measure the distance between test case and training case.
- Select the value of K for the test case.

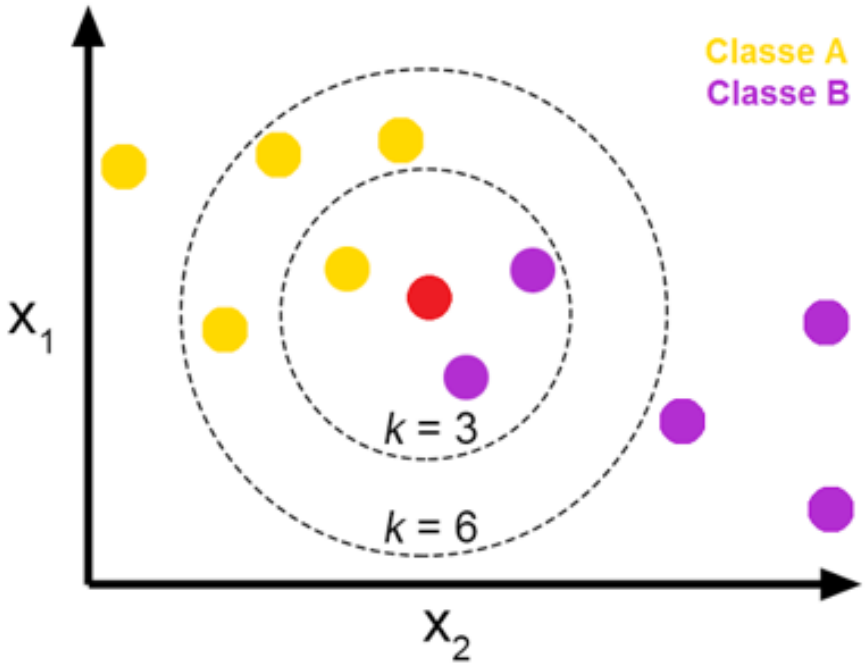

Figure 3.9: KNN vote counting
[21]

- Check the maximum votes for the test case and provide result.

However, for measuring the distance between test case and the training case in K-nearest neighbor, there are different kinds of equations. For example, Euclidean equation, Manhattan equation and the Minkowski equation. The formula of those equations are given below.

Euclidean Distance:

$$\sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \tag{3.2}$$

Minkowski Distance:

$$\left(\sqrt{\sum_{i=1}^{k} (\mid x_i - y_i \mid)^2} \right)^{\frac{1}{q}} \tag{3.3}$$

Manhattan Distance:

$$\sum_{i=1}^{k} \mid x_i - y_i \mid \tag{3.4}$$

3.2.4 Decision Tree

Decision tree is a supervised machine learning algorithm. Classification and regression both can be done by this algorithm. The decision tree uses a tree for coming to a result. Here, we have nodes and attributes. Each leaf nodes are the class label and the internal nodes are the attributes. Decision tree works based on sum of product rule. However, for constructing a decision tree we need to find the best root node. We cannot select the root node randomly. For determining the best root node we use information gain. Let us discuss information gain.
There are two steps of information gain.

- Calculate the entropy of target.
- Entropy of all attribute must be calculated and we will subtract the entropy of attributes from the entropy of target. Then we will get the information gain.

Entropy is the randomness and uncertainty of a random variable.
Equation of Entropy is:

$$E(X) = -\sum p(X) \log p(X) \tag{3.5}$$

Here, X means a random variable.
Equation of Information Gain is;

$$I(X, Y) = E(X) - E(X|Y) \tag{3.6}$$

Chapter 4

Implementation

4.1 Naïve Bayes Classifier:

The Comments Dataset involves classifying the sentiment as positive, negative or neutral given comments of customers in Facebook pages. At first we need to tokenize the comment data so that we can use that as features. In this process, to make the data more manageable we start by cleaning the data. First we read the data from the Comment Dataset by using a Python Library - Pandas. We read the comment and output data and build a data frame by these in the python script. Then we drop any null valued row in the data frame. Also, for cleaning the data for any noise or unnecessary text/characters for English (English and Banglish) text data we changed the data frame column of comment data from panda object to string and transformed all the strings to lowercase, removed any html tags and punctuations. Then by the help of NLTK stop words module we remove any stop words for example- 'but', 'again', 'there' etc. from the data. After that the data is ready for analyzing. However for Bangla text data we had to use our own set of stop words and punctuation, so that we can clean that Bangla text. Thus we have created three data frame consisting Banglish, Bangla and one with both.
Next, we used NLTK tokenizer module to extract features from our text data. The features were used to build a new data frame where column heads were the features and the values were true/false for every row of data. If the column head matches with the words in a row of previous data frame we put true in that corresponding row in the new data frame. This new data frame was used for our algorithm implementation.
The data frame was split into training and testing dataset. Then we fitted the training dataset to naive bayes algorithm. There are three types of Naive Bayes methods - Gaussian, Multinomial and Bernoulli. In our thesis we have implemented Gaussian Naive Bayes. The algorithms predict method was applied to predict the outcome of the test dataset.

4.2 Support Vector Machine (SVM) Classifier:

We cannot use the data we got in our training model just as is, we have to pre-process our data. For that we used the python library "Scikit-learn". Scikit-learn is a free machine learning library for Python programming language. For our pre-processing and feature selection and also for training and predicting we used different methods

provided by the Scikit-learn library. For pre-processing and feature selection we used the 'CountVectorizer' method. This method converts text into a matrix of tokens. Tokens are result of tokenizing a text. Tokenizing refers to the process where a series of string is broken up into words, keywords. In the process the punctuations are discarded. After tokenizing the text data the 'CountVectorizer' method creates a sprase representation of the counts. The head of the matrix are the tokens and the contents of the matrix are the counts of token occurrences. 'CountVectorizer' also creates features. Here features are all the tokens from the text data.

Figure 4.1: Sample of the features created using CountVectorizer method

We used the 'train_test_split' method from Scikit-learn to split our data into training and testing set. This method splits matrices into random train test subsets. We here reserved 20The first model that we trained was the SVM (Support Vector Machine). Scikit-learn library here also gives us convenient way to fit our data into the SVM model. Before fitting our data we need to specify some parameters for the SVM classifier.

Firstly SVM is inherently a two-class model which means it can differentiate between two different classes. However we are dealing with three class which are 'Positive', 'Negative' and 'Neutral'. The SVM here uses a technique call the OVR (One vs Rest) approach. This technique fits one classifier per class. For every classifier, one class is fitted against all the other classes. It gives the multiclass problem a significant computational advantage.

Secondly SVM has different kernel such as 'poly', 'sigmoid', 'linear' etc. Here we used all the kernels for our particular data and found the best result for the linear kernel showing that out data is indeed linearly separable. So we used the 'linear' kernel for our research.

Thirdly we give the value of C which determines the decision boundary for our classifier. Here we see the effect of C and accuracy.

After specifying all our parameters we can now fit our data and predict different sentiments using the SVM classifier.

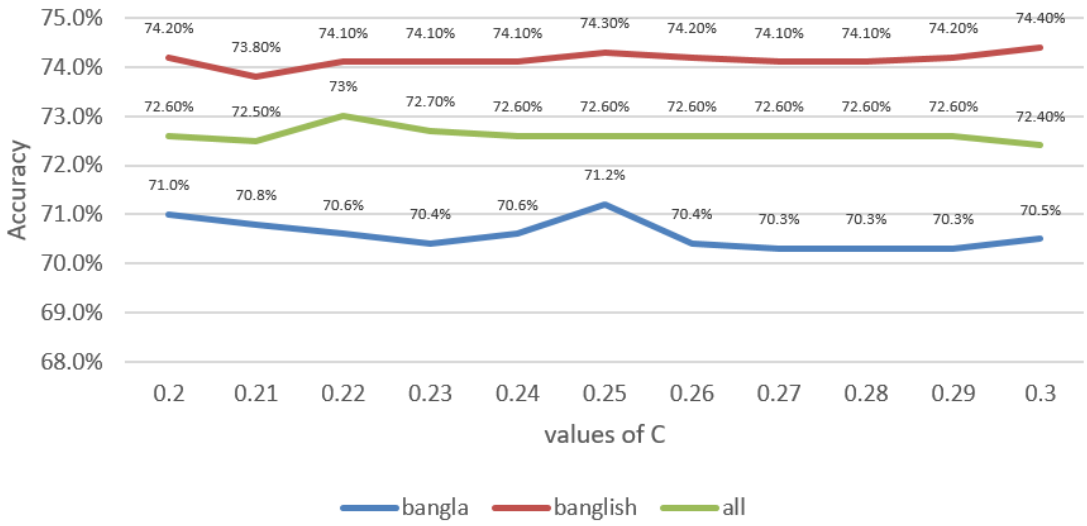

Figure 4.2: Effect of penalty parameter

4.3 K-Nearest Neighbors Classifier:

For implementing K-Nearest Neighbor, we keep our all comments of all languages in one dataset. So, now we need to extract our features. We got a method called "CountVectorizer" that helps us to tokenize our data and extract the features. In the figure 4.3, we have shown some features that are extracted from the comments of our dataset. In the same way, we also create different dataset for Banglish and Bangla language.

```
In [19]: from sklearn.feature_extraction.text import CountVectorizer
         cv = CountVectorizer()
         X = cv.fit_transform(data['message'])
         y = data['label']
         print(cv.vocabulary_)
```

```
{'আম': 6018, 'যদ': 6796, 'পক': 6615, 'উট': 6112, 'হল': 7044, 'এম': 6212, 'হয়': 7048, 'কত': 6280, '500mb': 465, '4day
s': 450, '১০ট': 7140, 'অফ': 5938, 'রট': 6837, '15': 215, 'mar': 3723, 'এর': 6222, 'আগ': 5978, '৫০০০': 7319, '২৯২': 724
9, 'ami': 801, 'ei': 2166, 'rokom': 4859, 'ekta': 2195, 'sms': 5121, 'paici': 4314, 'offer': 4147, 'ta': 5319, 'ki': 3229, 'e
kbar': 2178, 'nite': 4066, 'parbo': 4368, 'naki': 3932, 'joto': 3050, 'khushi': 3227, 'totobar': 5570, 'amar': 791, 'janar':
2932, 'code': 1656, 'bolen': 1345, 'shudo': 5061, 'matro': 3754, '49tk': 448, 'recharge': 4772, '5gb': 489, 'meyad': 3799, 'd
in': 2030, 'taki': 5349, 'khuno': 3226, 'diye': 2065, 'newya': 4004, 'jabeh': 2891, 'অভ': 5946, 'সফটওয়': 6976, 'জনক': 642
2, 'আপন': 6007, 'পত': 6621, 'সতর': 6965, 'ফণ': 6661, 'আঘ': 5980, 'টওয়': 6452, 'অবস': 5945, 'আশ': 6030, 'কর': 6304,
'এখন': 6184, '২৯৮': 7252, '৩০gb': 7259, '৩০দ': 7262, 'রত': 6842, 'ন১gb': 6612, 'এই': 6152, 'সম': 6990, 'পর': 6632, 'ন
ত': 6582, 'উন': 6122, 'আব': 6015, 'জন': 6421, 'যভভর': 6806, 'রক': 6823, 'উপ': 6127, 'হক': 7021, 'আমর': 6021, 'অল': 5
953, 'congratulations': 1701, 'কলর': 6316, 'আজক': 5989, '৪২tk': 7290, '৪৮প': 7304, 'tax': 5394, '১৫দ': 7177, 'আস': 603
2, 'করছ': 6305, 'রন': 6850, 'আন': 5996, 'সব': 6979, 'এব': 6210, 'ইম': 6089, 'সর': 7001, 'বন': 6691, 'জট': 6416, 'bogur
a': 1316, 'koi': 3343, 'grameenphone': 2539, 'nice': 4031, 'লন': 6898, 'বল': 6708, 'দয়': 6551, 'ইল': 6092, 'কঅয়': 6264,
'কব': 6293, 'কল': 6313, 'কচ': 6273, '121': 198, 'ওইট': 6241, 'অপ': 5931, 'টর': 6471, 'gp': 2520, 'soho': 5147, 'onnann
o': 4201, 'dial': 1982, 'korle': 3427, 'firti': 2356, 'kuno': 3551, 'message': 3790, 'ashchevna': 947, 'keno': 3154, '4610':
438, 'tay': 5395, 'kono': 3368, 'ache': 616, 'janaben': 2924, 'plz': 4541, 'ai': 676, 'pin': 4509, 'korar': 3379, 'por': 456
6, 'teke': 5416, 'phone': 4498, 'network': 3994, 'nei': 3976, 'kno': 3330, '১২১': 7164, '৫১৩৬': 7324, 'ডট': 6489, 'করল': 63
11, 'এমব': 6219, 'ওয়': 6262, 'কথ': 6287, 'er': 2235, 'belence': 1199, 'cheek': 1579, 'kora': 3375, 'jaitese': 2915, 'na': 3
918, 'erron': 2239, 'application': 883, 'dekhay': 1874, '1600': 226, 'মত': 6751, 'যবস': 6803, 'আছ': 5983, '333': 379, 'b
y': 1458, 'mistake': 3835, 'dia': 1980, 'disi': 2050, 'ekhn': 2181, 'out': 4259, 'going': 2492, 'call': 1481, 'barred': 114
```

Figure 4.3: Feature extraction using CountVectorizer

After getting features, we fit our model and split the dataset for training and testing. We keep 20% of data for testing. However, for implementing the KNN algorithm, our main task is to find the best value of K. K will decide how many votes will be counted. The lower value of K gives more flexibility and higher value of K gives chance to count more votes. We have calculated the mean error for the different value of K. For different value of K, we kept our range from 1 to 40. Finally, from the figure 4.4, we can see that, for the all language dataset, it provides lowest mean

error, when K=1. That means, we will get highest accuracy when K=1. The mean error increased when K=2, because, it takes 2 votes from the neighbors.

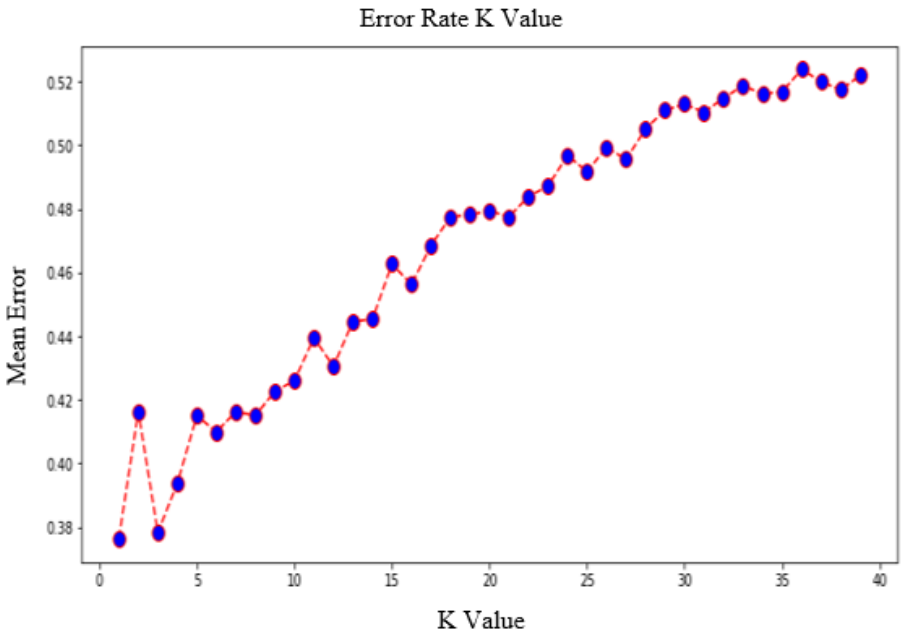

Figure 4.4: Mean error for all language dataset

In the same way, from the figure 4.5, we can see for Bangla language, it provides lowest mean error when K=7.

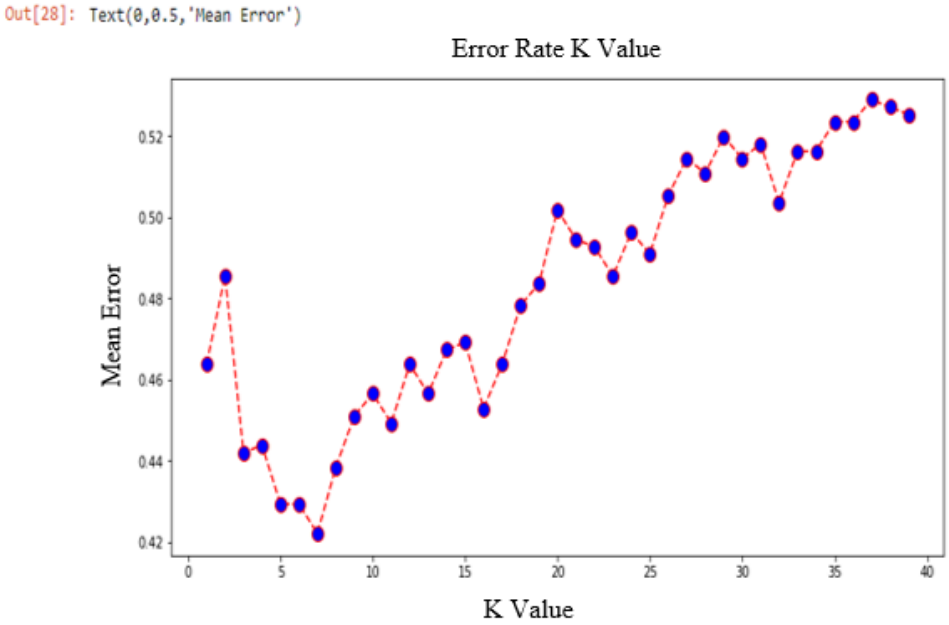

Figure 4.5: Mean error for Bangla language dataset

And from figure 4.6, we can see that, for Banglish Language we get the best value of K=2

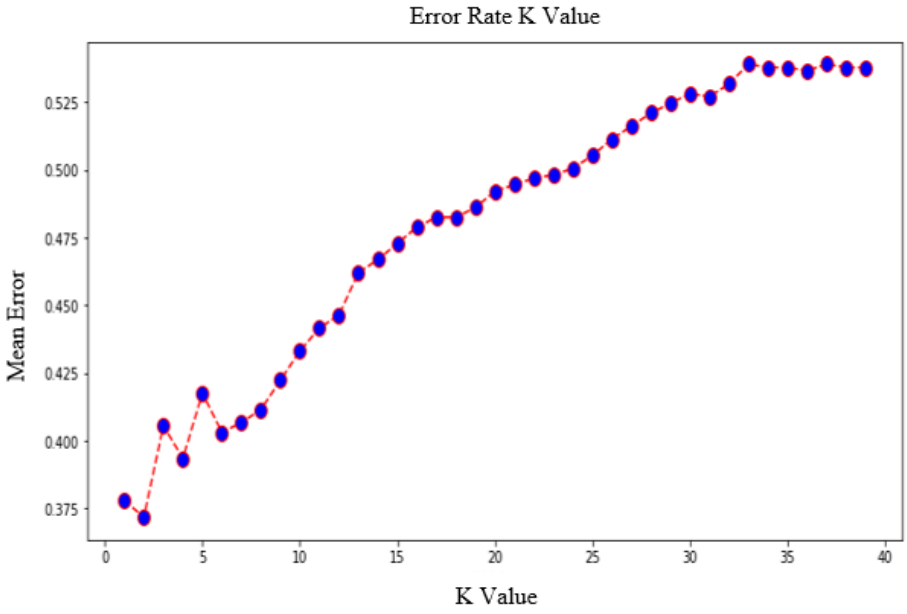

Figure 4.6: Mean error for Banglish language dataset

However, for calculating the distance from the test case to the train cases we have used the Euclidean equation (3.3).

4.4 Decision Tree Classifier:

After pre-processing the data with countvectorizer we also used it to train decision tree. We used information gain for getting the best node. This creates a tree with values of entropy. The nodes of the tree are the features of our dataset. As we have a lot of feature decision tree has a risk to overfit our data. So we limited the max_depth of our tree to 1000.

Chapter 5

Result Analysis

For our result analysis we will use different performance metrics to differentiate between different results of different classifier. The metrics that we will use are described below.
True Positives (TP): This is the correctly predicted positive value. That means the actual class is positive and the predicted value is also positive.
True Negative (TN): This is the correctly predicted negative value. That means the actual class is negative and the predicted value is also negative.
True Neutral (TNu): This is the correctly predicted neutral value. That means the actual class is neutral and the predicted value is also neutral.
False Positive (FP): False positive means, when actual class is positive but predicted class is negative or neutral.
False Negative (FN): False negative means, when actual class is negative but predicted class is positive or neutral.
False Neutral (FNu): False neutral means, when actual class is neutral but predicted class is positive or negative.
Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations [22].

$$Precision = \frac{\text{TP}}{\text{TP + FP}} \tag{5.1}$$

Recall (Sensitivity): Recall is the ratio of correctly predicted positive observations to the all observations in actual class [22].

$$Recall = \frac{\text{TP}}{\text{TP + FN}} \tag{5.2}$$

F1 score: F1 Score is the weighted average of Precision and Recall [22].

$$Recall = \frac{\text{TP2*(Recall*Precision)}}{\text{Recall + Precision}} \tag{5.3}$$

In our algorithm we reserved 20% of our whole dataset form testing. After training the model we used the library "metrics" from "scikit-learn". We implemented the method "classification_report" to get the different performance metrics for our three classifiers.

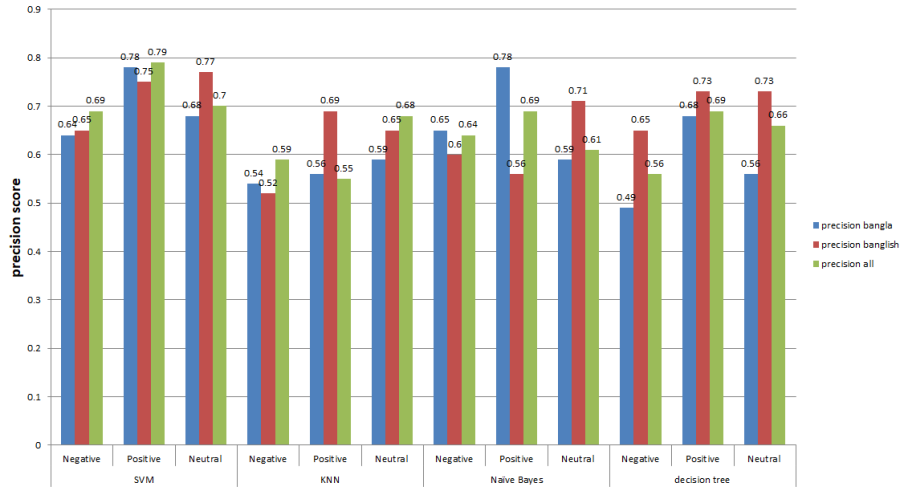

Figure 5.1: Precision score

From Figure 5.1 we can compare the precision score of our classifiers. First, we look at the precision for the dataset containing both the Bangla and Banglish language. Here the SVM has the overall high precision for all three classes [23]. From figure 5.1 we see the precision score for neutral, positive and negative classes are respectively 0.7, 0.79 and 0.69. The precision score for neutral, positive and negative classes of Naïve Bayes classifier are respectively 0.61, 0.69, 0.64 and for KNN are respectively 0.68, 0.55, 0.59. This means that among the predicted classes using SVM classifier most of the classes are correct class. For our purpose we need high precision as we are targeting live market analysis. Classifying a non-negative review negative will adversely affect the market analysis and may lead to false presumption of the market sentiment. From Figure 5.1 we see that SVM classifier also gives us better recall score for dataset containing only Bangla and only Banglish language.

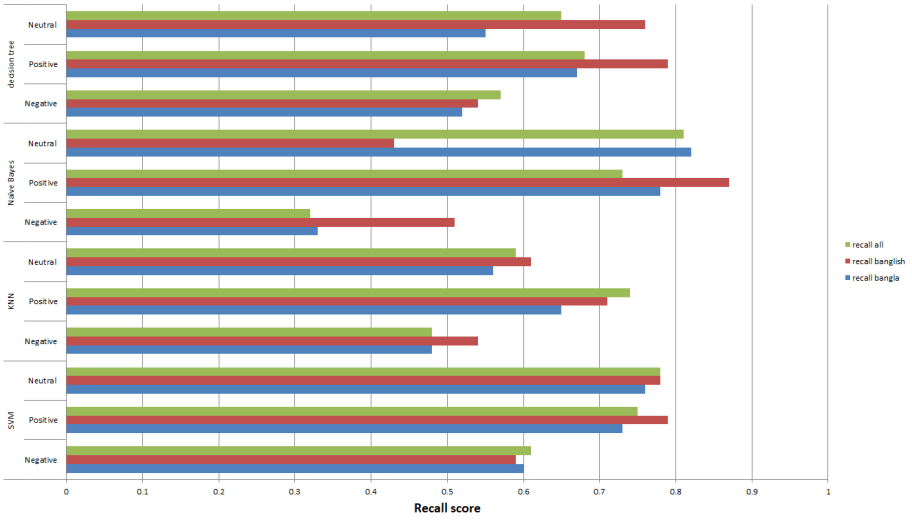

Figure 5.2: Recall score

Here we see the recall score. Recall score gives us the percentage of predicted class from the total number of that class in the dataset [1]. For example, from figure 5.2 we see that SVM classifier has a recall score of 0.61 for both Bangla English combined dataset. This score means that among the 100 negative comments 61%

negative comments are predicted. Form the Figure 5.2 we can see that Naïve Bayes classifier sometimes gives us better recall score. However, of our purpose we have to prioritize precision more than recall. The reason is that predicting the market correctly is more important than predicting all the data from a class.

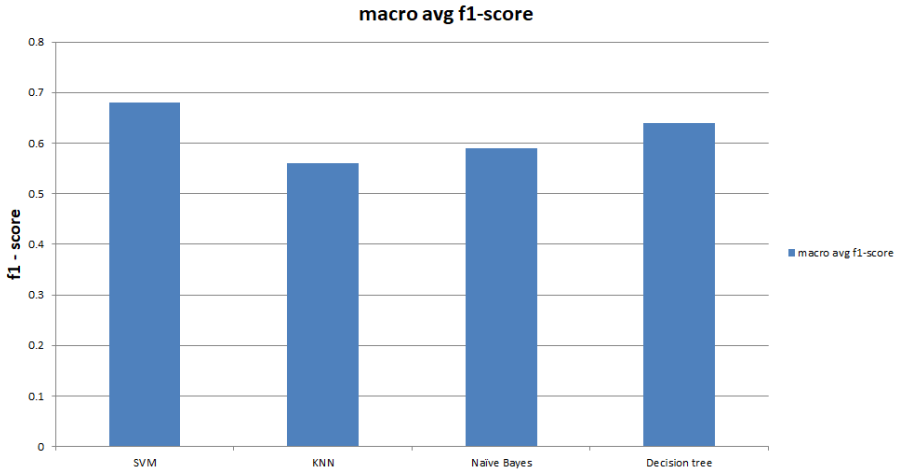

Figure 5.3: Macro average f1-score

Now we discuss the f1-score. This metric is the harmonic mean [24] of our previously found precision and recall score. This gives us a glace of the overall relative performance of our classifiers. From Figure 5.3 we see that SVM has the overall highest score among the three with 0.68. Also, this f1 score is the macro average f1 score of all the three classes.

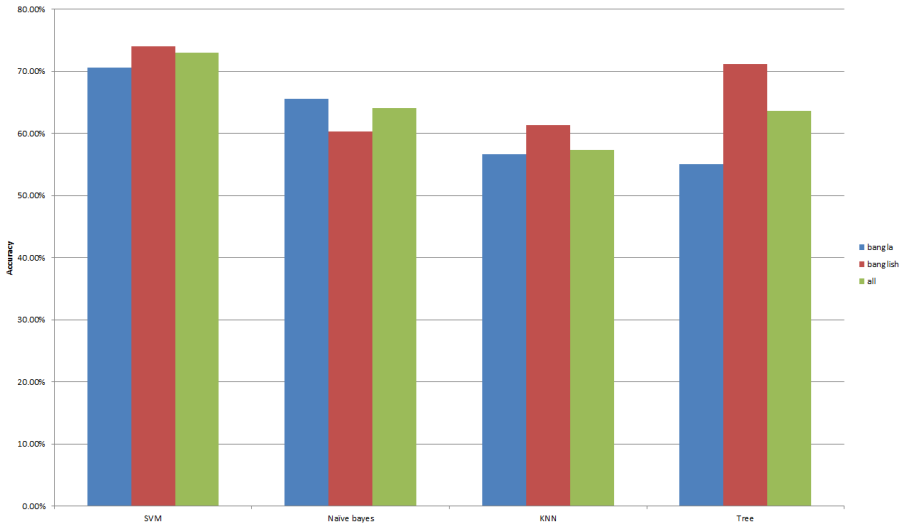

Figure 5.4: Accuracy score

From Figure 5.4 we see the accuracy score. The accuracy score gives us the overall performance of our classifiers [25]. This score defines the percentage of correctly predicted samples from all the test samples we provided. The formula used to calculate accuracy:

$$accuracy = \frac{\text{TP+TN+TNu}}{\text{TP+FP+TN+FN+TNu+FNu}} \tag{5.4}$$

From Figure 5.4 we see the common trend that we saw in our previous performances metrics. SVM has the highest accuracy in all the three datasets. SVM scored Bangla, Banglish and combined dataset respectively 70.65%, 74% and 73%. This means that SVM did correctly predicted in comparison to KNN and Naïve Bayes and decision tree. The reason SVM performed so well compared to others is the nature of our dataset. Our dataset is linearly separable. We came to this conclusion because we used different kernel of SVM and got the highest accuracy for the linear kernel. We used sigmoid, polynomial kernels. However they gave accuracy of 39% and 42% respectively. So it is clear that our data set is linearly separable.
Here Naïve Bayes also performs well compared to others. Naïve Bayes performs best when the dataset are conditionally independent. In our dataset the sentiments are not dependent with each other. One sentiment is conditionally independent with another on. Moreover Naïve Bayes works well with data that has plenty of input features but small number of classes. Our data has a lot of features which are all the words from the comments. Also we have only three classes i.e. 'positive', 'negative' and 'neutral'. That's why our data fitted well with Naïve Bayes.
Moreover we used decision tree to classify our data. Decision creates a tree from the feature sets of our data. As we have a lot of feature our tree had a depth of 4000. This is a problem with decision tree that a lot of times data can be overfitted. To tackle that we limited max_depth of our tree to 1000. From figure 5.5 we see that this improved the accuracy of decision tree.

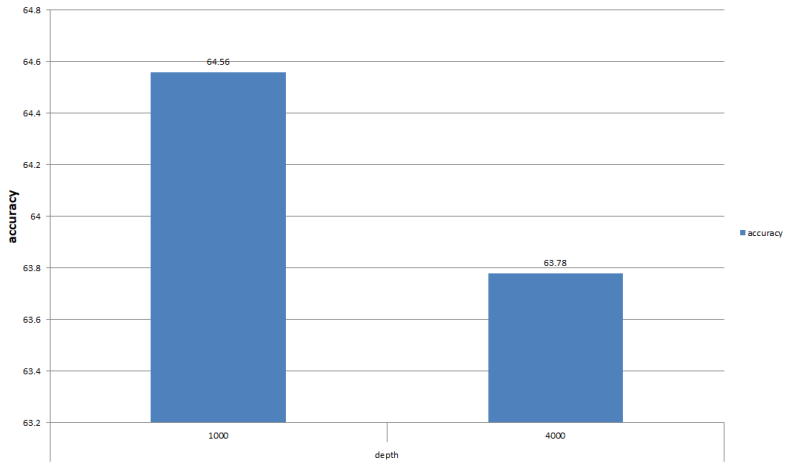

Figure 5.5: Decision Tree Accuracy

Moreover we used tested our dataset with KNN as this algorithm is a simple supervised algorithm. With our 'all' dataset we got accuracy of 57.3%. However for KNN we shorten our data to see the results. We divided our data into two smaller part one with 3000 comments and another with 4500. From figure 5.6 we see that when we reduce the data we get more accuracy. This suggests that our dataset has overfitted for KNN and that's why it is not performing as it is.

Furthermore if we change out dataset we get slightly different results. If we analyze all the algorithm used we see that 'bangla' data set performs the worst and 'banglish' dataset is performing the best. The reason is that when we scrapped the data from Facebook we did not get many comments that were written in Bangla. As a

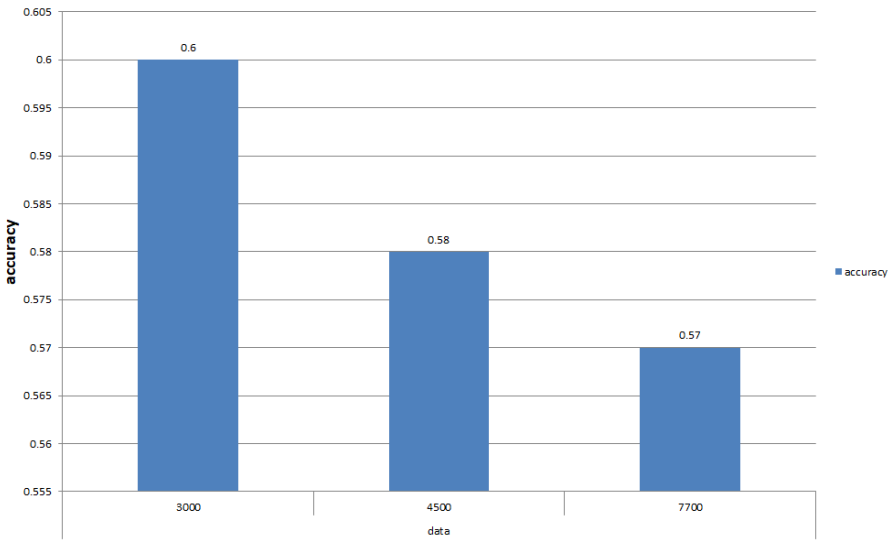

Figure 5.6: KNN Accuracy for Different Number of Data

result number of comments in 'bangla' dataset which consists of only pure Bangla comments is lower than the 'baglish'. Number of 'bangla' datasets are almost 3000 and number of comments in 'banglish' dataset is almost 5000. That why we see that for 'banglish' is performing the best. For the dataset that consists all the dataset give a slight worse accuracy than 'baglish' even after having comments more than 7000. This is because we get some noise tokenizing the English and Bangla language together.
From all the above assessments we can see a clear picture that SVM is giving us better performance in comparison to KNN and Naïve Bayes in all aspects.

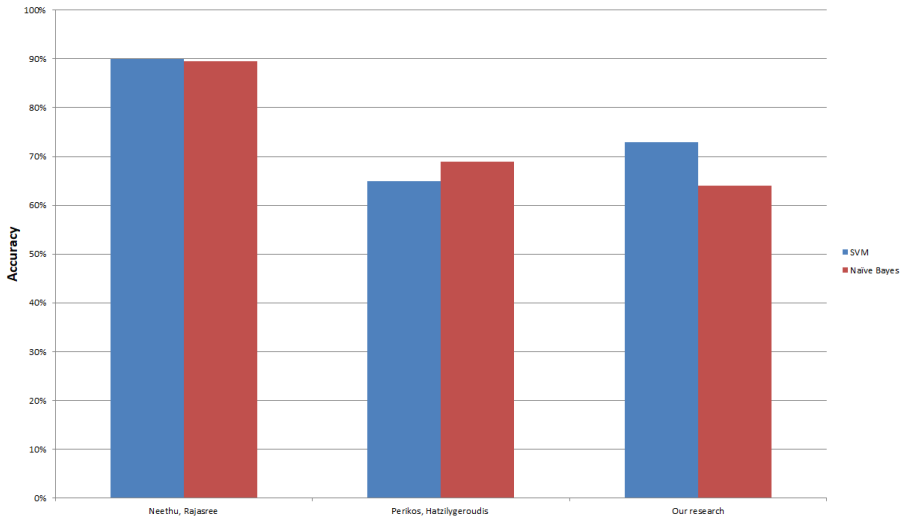

Figure 5.7: accuracy comparison

Here we compare our work with two of the paper that were somewhat related to our contribution. Neethu and Rajasree worked on sentiment analysis in twitter dataset using machine learning approach [5]. We have found some similarities and differences in between their work and our work. However, they worked on only two output classes. That means, their sentence can be either positive or negative.

But, in our paper, we have worked on three output classes. In addition to that, for feature extraction they use unigram method and create a collection of words. But, in our work we used countvectorizer method for feature extractions. Moreover, their dataset was only in English language. In contrary, we have used Banglish and Bangla language dataset. Neethu and Rajasree got higher accuracy for different algorithms compare to us. Because, they only have used English language dataset and their total dataset was 1200. Moreover, before implementing the algorithms they have corrected the misspelled words. But, in our case, we could not correct the misspelled words. A single word can be pronounced differently. For example, we trained "moja" in training time but in testing time we got "moza". Though the meaning of both words are same but spellings are different. Because of this ambiguity and using Bangla phonetic words we got less acuracy then them. In figure 5.7 we showed the comparison of the accuracy of with their work.
Perikos and Hatzilygeroudis in their research on analyzing big social data worked with dataset scrapped from twitter [11]. They indexed and analyzed 23,352 tweets where we used 7000 comments. Also they used a sentiment model which Ekman model which are "anger," "fear," "disgust," "joy," "sadness," and "surprise". However we worked with three sentiment 'Positive', 'Negative' and 'Neutral'. They also prioritize hash tags but we did not consider such parameters. Comparison of accuracy is shown in the figure 5.7.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

To conclude, people in today's world are highly engaged with social media and are active participants in giving their opinions on the services they buy through commenting on the Facebook pages or groups of the services. To understand user's opinion for providing better and user friendly services sentiment analysis on the comments related to the services can play an effective role. However, users also prefer to give their opinion in their native languages. So, in our research we have implemented a system that analyses the native language user's feedback using different machine learning algorithms. The dataset was consist of both Bangla and Banglish typed words. During our work, after implementing Support Vector Machine (SVM), Naïve Bayes and K-Nearest Neighbors (KNN) algorithms we have found that the SVM gives more accurate results than other three algorithms. The dataset was used in both ways, combining the two types of languages and separating them. Using the combined dataset the SVM gives 73% accuracy whereas the Naïve Bayes gives 64.11%, Decision Tree gives 63.61% and the KNN gives 61.60% accuracy. Moreover, while using dataset consisting only Bangla language, these algorithms give 70.65%, 65.58%, 55.07% and 56.70% accuracy respectively for SVM, Naïve Bayes, Decision Tree and KNN algorithms and for dataset with only Banglish language accuracy were 74%, 60.29%, 71.23% and 61.3% respectively. This system can be implemented as web or any platform application so that the service providers can get user's positive, negative or neutral feedback and improve their service efficiently.

6.2 Future Work

In our research, we have developed a system for sentiment analysis of users' feedback on various social media groups. We have used different machine learning classifier to predict the positive, negative and neutral feedback from the customer. For our future work we want to improve the dataset. We want to increase the data for training the classifier which will help to get more accuracy. As there is no Banglish dictionary and different people type one word in different way, we want to work on this issue to make the system more efficient. We also like to work on automatic data scrapping process.

With this system we want to add more functions which will make the analysis simpler and will help to learn the topics better. We like to work on a location based

classification where the mobile operators will be able to know area based issues. As in social media, people post comments from different location and in their account the location is given, we would like to work with the area based classification. It will help the operators to fix the issue specifically.
Furthermore, we want to make the system more user friendly by building a user interface. In future, we want to take this work in such a stage where getting users' opinion and taking decisions for improvement of the mobile operators will be easier enough through this system.

Reference

[1] B. Kramer, *The emotions of social sharing*, https://www.socialmediatoday.com/content/emotions-social-sharing, (Accessed on 17/11/2019), 2014.

[2] M. Novak, *Average user still spending 38 more minutes on facebook per day than they should.* https://gizmodo.com/average-user-still-spending-38-more-minutes-on-facebook-1835061024, (Accessed on 17/11/2019), 2019.

[3] J. H. Kietzmann, K. Hermkens, I. P. McCarthy, and B. S. Silvestre, "Social media? get serious! understanding the functional building blocks of social media", *Business Horizons*, vol. 54, no. 3, pp. 241–251, 2011.

[4] Y. Dai, T. Kakkonen, and E. Sutinen, "Minedec: A decision-support model that combines text-mining technologies with two competitive intelligence analysis methods", *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 3, pp. 165–173, 2011.

[5] M. S. Neethu and R. Rajasree, "Sentiment analysis in twitter using machine learning techniques", in *Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, IEEE, 2013.

[6] H. Parveen and S. Pandey, "Sentiment analysis on twitter data-set using naive bayes algorithm", in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology*, IEEE, 2016.

[7] C. Troussas, M. Virvou, K. J. Espinosa, K. Llaguno, and J. Caro, "Sentiment analysis of facebook statuses using naive bayes classifier for language learning", in *IISA 2013*, IEEE, 2013.

[8] H. Kaur and V. Mangat, "A survey of sentiment analysis techniques", in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, IEEE, 2017, pp. 921–925.

[9] M. Yassine and H. Hajj, "A framework for emotion mining from text in online social networks", in *2010 IEEE International Conference on Data Mining Workshops*, IEEE, 2010, pp. 1136–1142.

[10] Z. Wang, C. S. Chong, L. Lan, Y. Yang, S. B. Ho, and J. C. Tong, "Fine-grained sentiment analysis of social media with emotion sensing", in *2016 Future Technologies Conference (FTC)*, IEEE, 2016, pp. 1361–1364.

[11] I. Perikos and I. Hatzilygeroudis, "A framework for analyzing big social data and modelling emotions in social media", in *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*, IEEE, 2018, pp. 80–84.

[12] S. X. Mashal and K. Asnani, "Emotion intensity detection for social media data", in *Proceedings of 3rd international conference on document analysis and recognition*, IEEE, 2017, pp. 155–158.

[13] D. Mumtaz and B. Ahuja, "Sentiment analysis of movie review data using senti-lexicon algorithm", in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, IEEE, 2016, pp. 592–597.

[14] R. K. Bakshi, N. Kaur, R. Kaur, and G. Kaur, "Opinion mining and sentiment analysis", in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, 2016, pp. 452–455.

[15] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision", *Processing*, vol. 150, Jan. 2009.

[16] N. Zainuddin and A. Selamat, "Sentiment analysis using support vector machine", Sep. 2014, pp. 333–337. DOI: 10.1109/I4CT.2014.6914200.

[17] R. Gandhi, *Naive Bayes Classifier*, https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c, [Online; accessed 17-Nov-2019], 2018.

[18] R. Sunil, *Understanding Support Vector Machine algorithm from examples*, https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/, [Online; accessed 17-Nov-2019], 2017.

[19] R. Berwick, "An idiot's guide to support vector machines (svms)",

[20] D. Subramanian, *A Simple Introduction to K-Nearest Neighbors Algorithm*, https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorithm-b3519ed98e, [Online; accessed 17-Nov-2019], 2019.

[21] Genesis, *Pros and Cons of K-Nearest Neighbors*, https://www.fromthegenesis.com/pros-and-cons-of-k-nearest-neighbors/amp/?fbclid=IwAR2rWfa7E6M-0U3Mlx7KEwEKg6m9I3LlAgQwynfEhouQyN8jrLiUV5XKTBM, [Online; accessed 30-Nov-2019], 2018.

[22] A. Long, *Understanding Data Science Classification Metrics in Scikit-Learn in Python*, https://towardsdatascience.com/understanding-data-science-classification-metrics-in-scikit-learn-in-python-3bc336865019, [Online; accessed 17-Nov-2019], 2018.

[23] B. Shmueli, *Multi-Class Metrics Made Simple, Part I: Precision and Recall*, https://towardsdatascience.com/multi-class-metrics-made-simple-part-i-precision-and-recall-9250280bddc2, [Online; accessed 17-Nov-2019], 2019.

[24] B. Shmueli, *Multi-Class Metrics Made Simple, Part II: the F1-score*, https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1, [Online; accessed 17-Nov-2019], 2019.

[25] S. Ghoneim, *Accuracy, Recall, Precision, F-Score Specificity, which to optimize on?*, https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124, [Online; accessed 17-Nov-2019], 2019.