Sentiment Analysis of Mobile Apps Reviews through Natural Language Processing (NLP)

Abu Hanzala

Department Of Computer Science and Engineering

Daffodil International University

Dhaka, Bangladesh

ID: 211-15-4022

abu15-4022@diu.edu.bd

Mamunur Rashid

Department Of Computer Science and Engineering

Daffodil International University

Dhaka, Bangladesh

213-15-4556

rashid15-4556@diu.edu.bd

Md Abdullah Al Mazed

Department Of Computer Science and Engineering

Daffodil International University

Dhaka, Bangladesh

ID: 213-15-4478

mazed15-4478@diu.edu.bd

Md. Rafi Al Karim

Department Of Computer Science and Engineering

Daffodil International University

Dhaka, Bangladesh

ID: 213-15-4490

karim15-4490@diu.edu.bd

Abstract— In today's digital world, mobile apps play a crucial role in our daily lives. With the increasing popularity of app stores, users have become more vocal in expressing their opinions and experiences through app reviews. Sentiment analysis, a subfield of natural language processing (NLP), aims to extract and classify opinions from text, making it a valuable tool for understanding user sentiment towards apps. The performance was assessed using the ensemble learning method as we applied machine learning algorithms with the TF-IDF text representation scheme. Student reviews were used to test our model after it had been trained on Google Play Store reviews. On the trigram + TF-IDF scheme, SVM recorded the highest accuracy (90%) and F1-score (0.90).

Keywords— Sentiment analysis, Machine learning, students' reviews, Google Play store apps.

Introduction

App reviews provide a rich source of information for app developers, marketers, and researchers. By analysing app reviews, they can gain insights into user satisfaction, identify potential issues, and track app performance over time. Sentiment analysis is a powerful technique that can automate the process of extracting and classifying opinions from app reviews, providing valuable insights into user sentiment.

Sentiment analysis techniques can be broadly categorized into two main approaches: lexicon-based and machine learning-based.

Sentiment analysis employs both lexicon-based and machine learning-based approaches. A vocabulary-based strategy considers the words' semantic order while excluding labelled data. Words and phrases in a document are included in a dictionary that is manually developed. The objective of sentiment analysis using machine learning involves working with labelled data and assisting in the creation of models using supervised learning techniques, such as K-nearest neighbour (KNN), support vector machine (SVM), and naive bayes (NB).

- The TF-IDF text representation system was applied to the uni-gram, bi-, and tri-gram strategies.
- The text representation system was subjected to supervised machine learning techniques, including NB, SVM, logistic regression (LR), KNN, and Random Forest (RF), which were then applied, and their respective performances compared.

Lexicon-based sentiment analysis relies on a predefined dictionary of words, phrases, and their associated sentiment scores. These dictionaries, also known as lexicons, are typically manually curated or created using machine learning techniques. The sentiment score of a text is calculated by summing the sentiment scores of all the opinionated words or phrases it contains.

Machine learning-based sentiment analysis utilizes machine learning algorithms to classify the sentiment of text. These algorithms are trained on large datasets of labelled text, where each text has been assigned a sentiment label (e.g., positive, negative, or neutral). Once trained, the algorithm can classify the sentiment of new text based on the patterns it has learned from the training data.

The System Description

This section provides a comprehensive overview of the key components and functionalities of the system. The system graph is shown in below:

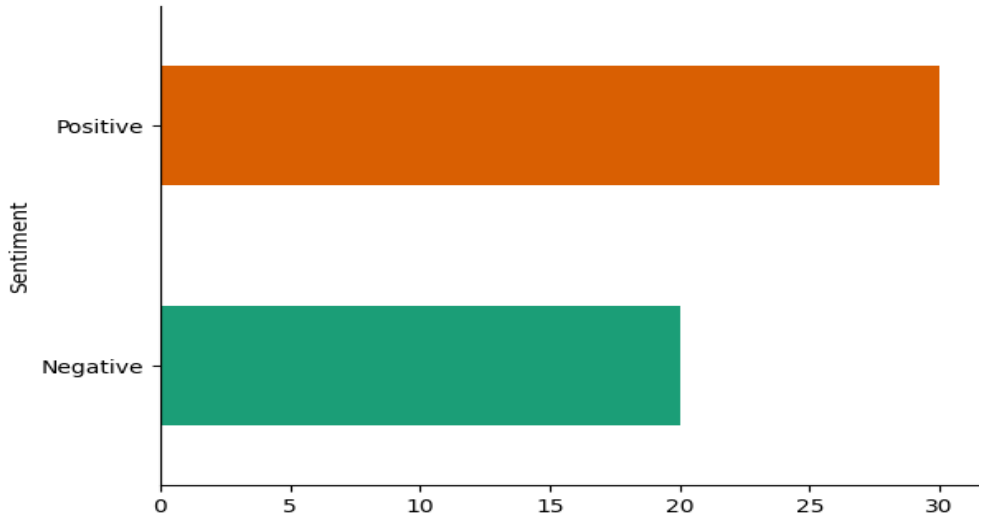

Fig1: Graph

Dataset Acquisition

App reviews were compiled and stored in a CSV file. The dataset contained one app and two features' columns. The dataset has the following columns: [reviews], and [Sentiment].

The objective is to comprehend the pattern and contrast it with test data by examining student evaluations of the Google Paly Store. As a result, we had gathered actual data from the Google Play Store. the evaluations of the often-used applications. Ratings-type and app(reviews) were among the fields. Students gave the apps a 5-point rating system, with a score of three or more being categorized as "positive" and a score of less than three as "negative."

Date Preprocessing Tools

I used regular expression for remove emails, remove html tags, remove URLs, remove accented chars, remove special chars and lowercase all word.

A very popular method for representing all the unique words found in the documents is TF-IDF. The feature extraction process from text documents is aided by this technique. Counter Vectorizer and TF-IDF (Term Frequency-Inverse Document Frequency) are the two weighted systems that are commonly used.

TF-IDF is essentially a combination of two metrics: IDF and TF. It's commonly utilized in text mining and information retrieval, and it aids in search engine query ranking. It's employed to assign a word's value based on its weight.

TF-IDF is defined:

Term Frequency (TF) = Number of repetition word / Total word in sentence

Inverse Document Frequency (IDF) = log (Total Number of sentence / Number of sentences contain in the word)

$$\text{TF-IDF} = \text{T F (w/ D)} * \log(\text{C/df(w)})$$

In this case, the TF-IDF score for word w in document D is mapped by T F − IDF (w, D). Therefore, if the term is uncommon, it will receive a higher score. IDF calculates a word's importance inside a corpus of documents. IDF is obtained by dividing the number of documents in a corpus C by the frequency of a word in a document w,

then applying the log transform. Words that appear frequently in many papers will be given a lower weight; otherwise, they will be given a higher weight.

Machine Learning Model Selection

SVM Architecture

Supervised learning is the process of training a machine learning algorithm on a dataset that has been labelled. In essence, labelling indicates the result based on the input parameters.

[SVM]The supervised machine learning algorithm is called SVM. It assists in resolving issues with regression and classification. In support vector machines (SVM), a hyper-plane is found to distinguish between data points by plotting the data points in an N-dimensional space, where N is the number of features. Although this method is computationally costly, it is employed when the number of dimensions is large in comparison to the number of data points.

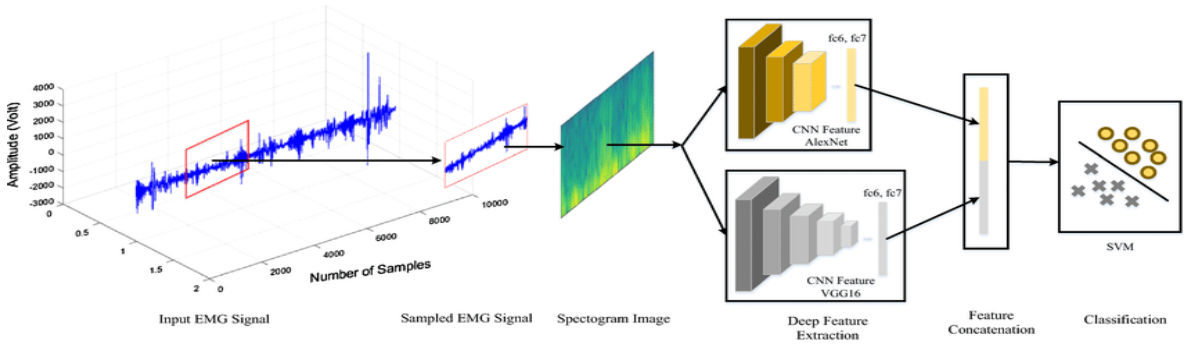

Fig2: SVM Classification Process

A set of labelled data points, each represented by a feature vector and a corresponding class label, make up the training data. Using this data, the SVM algorithm learns to determine the ideal hyperplane.

A mathematical operation called the kernel function raises the dimensionality of the input data. The SVM may now learn complex decision boundaries that are not linearly separable in the original input space thanks to this change. The radial basis function (RBF) kernel, polynomial kernel, and linear kernel are examples of popular kernel functions.

The data points that define the greatest margin and are closest to the hyperplane are known as support vectors. The location and orientation of the hyperplane are decided in large part by these data scores.

Naïve Bayes Architecture

[NB] NB is a probabilistic classifier that applies the Bayes theorem. According to a given possibility, objects with similar qualities are grouped together in one class, while others are placed in another class. Strong independence assumptions between the characteristics are made in this strategy. Because all keywords may be pre-computed and just a minimal amount of training data is needed, categorization is made simple, quick, and effective.

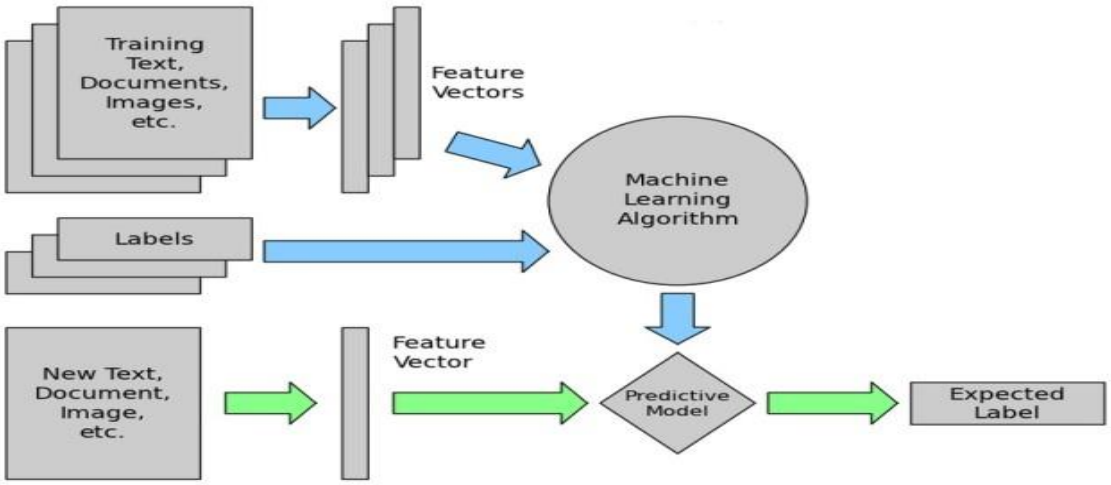

Fig3: NB Classification Process

It is assumed by the Naive Bayes classifier that each data point's features are independent of one another. Though it can make calculations easier and speed up the Naive Bayes classifier's training, this assumption is frequently untrue in real-world situations.

From the training data, the Naive Bayes method calculates the probability of each class and the probabilities of each feature given each class. New data points are then classified using these probabilities.

When a new data point is received, the Naive Bayes classifier assigns it to the class with the highest probability after calculating the probability of each class receiving the data point.

KNN Architecture

[KNN] By assigning the object to a class based on a plurality vote from its k (positive integer) neighbours, KNN solves the classification problem. The sum of an object's k nearest neighbours' values is its output result in a regression. The concept of resemblance between an object and its neighbours in terms of distance, proximity, or closeness is represented by KNN.

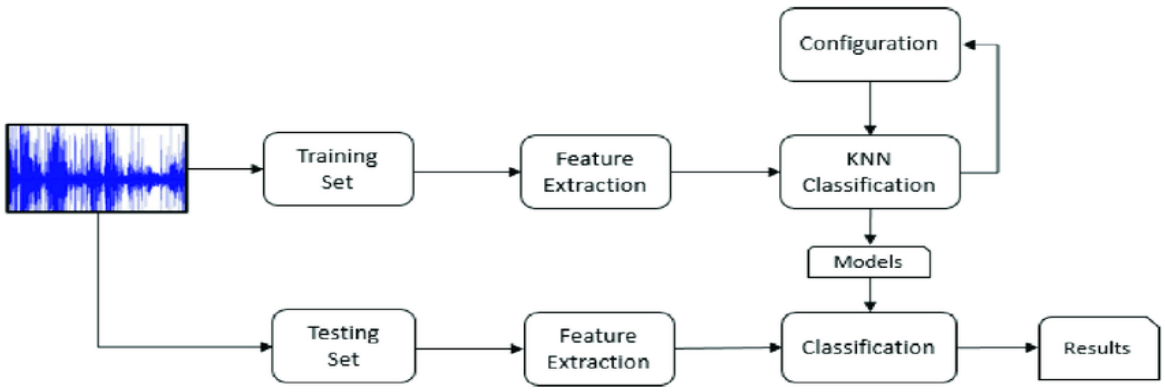

Fig 4: KNN Classification Process

The complete training dataset is just stored by the algorithm; no explicit model development is done. This is since the KNN method uses the data points itself for categorization or prediction rather than building a model from the data.

The algorithm determines the distance between each new unlabelled data point and every other data point in the training dataset during the prediction phase. Numerous metrics, including the Minkowski, Manhattan, and Euclidean distances, can be used to calculate the distance.

RESULTS ANALYSIS

The best algorithms on this dataset throughout all three unigram types with TF-IDF feature extraction were SVM and KNN. In terms of accuracy (90%) and F-score (0.90), SVM scored the best.NB achieved an accuracy of 84.08% and F-score of 0.82 on the unigram + Counter vectoraizer scheme, while KNN obtained an accuracy of 90.00% and an F-score of 0.88 on uni-gram + TF-IDF featurization. It didn't take much to train and is an average algorithm if compared to others.

CONCLUSIONS AND FUTURE WORK

Sentiment analysis was successfully carried out utilising the machine learning technique. For performance compares, three famous classification algorithms—NB, SVM, and KNN were employed. An ensemble method called bagging was used to improve the classifier's predictive performance. SVM fared better than other classification algorithms in this study in terms of accuracy (90%) on the TF-IDF + uni-gram feature.

Future Scope is I will add Jason file. Because anyone reviews our apps it will add our dataset.

Methodology

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

Acknowledgments

The developers of the Natural Language Toolkit (NLTK), a popular NLP library that offers a wide range of tools for sentiment analysis, including date parsing, normalization, and feature extraction.
The developers of Pandas, a popular data analysis library that provides date manipulation functions and tools for handling time series data like app reviews.
The researchers who have contributed to the development of machine learning models for sentiment analysis, including Naive Bayes, Support Vector Machines (SVMs) and KNN.