Islamic University of Technology

Department of Computer Science and Engineering

Fake Review Detection Using Machine Learning Techniques

Sadat Shahriar Bari 170042056

Robiul Ahammed Sakib 170042058

Nabil Hossain Nico 170042076

Supervisor:
Lutfun Nahar Lota
Assistant Professor, CSE

A thesis proposal submitted in partial fulfilment of the requirements of the Islamic University of Technology for the degree of Bachelor of Science in *Computer Science and Engineering*

May 13, 2022

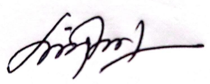
Abstract

Nowadays, review sites are increasingly confronted with the spread of disinformation, for example, opinion spam, which aims to promote or harm certain target businesses, by simultaneously deceiving the human readers. For this reason, over the past years, several data-driven approaches have been proposed to assess the credibility of user-generated content delivered through social media in the form of online reviews. Linked to both review and reviewers, as well as the network structure that links separate entities at the review site.This article aims to provide an analysis of various machine learning methods and deep learning methods for analyzing fake user review detection on bangla languages based on the reviewer and review-centric features.Additionally, this work offers to provide a synthesized dataset for fake user review detection in the Bangla language.

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Sadat Shahriar Bari, Robiul Ahammed Sakib and Nabil Hossain Nico under the supervision of Lutfun Nahar Lota, Assistant Professor of Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Gazipur, Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others have been acknowledged in the text and a list of references is given.

Authors:



Sadat Shahriar Bari
Student ID: 170042056

Robiul Ahammed Sakib
Student ID: 170042058

Nabil Hossain Nico
Student ID: 170042076

Supervisor:

13.05.22

Lutfun Nahar Lota
Assistant Professor
Department of Computer Science and Engineering
Islamic University of Technology

Acknowledgement

We would like express our gratitude towards IUT authority for granting us the fund and providing assistance required to implement our proposed system. We are indebted to our supervisor, Lutfun Nahar Lota for providing us with insightful knowledge and guiding us at every stage of our journey. Finally, we would like to express our heartiest appreciation towards our family members for their continuous support, motivation, suggestions and help, without which we could not have achieved the scale of implementation that we have achieved.

Contents

List of Figures

Chapter 1

Introduction

E-commerce has been severely affected by the rapid spread of the web and the internet technology that has enabled the production of user-generated content (UGC). Consumers continually state publicly and share their opinions about purchased products or services and rate quality and value for money. A recent study says that 86% of people hesitate to do business with a company if it has negative online reviews, 78% of people trust online reviews as much as a recommendation from a friend or a family member. Also On average, reviews produce an 18% uplift in sales [1]. The study [1] also says 54% of consumers would not buy a product if they suspected it to have fake reviews and 4% of fake online reviews caused $152 billion lose for world's leading e-commerce sites (including Trip Advisor, TrustPilot, Amazon, etc).

The impact of online reviews has given rise to "dishonest" practices aimed at profiting from or manipulating user reviews for certain products or services. Due to rivalry or commercial motives, professional "spammers" are frequently employed to saturate the online reviewing area with bogus evaluations. The widespread use of fraudulent reviews has become a major issue that has piqued the scientific community's curiosity. The majority of research efforts are aimed at improving fake review identification to re-establish the validity of online opinions.

The approaches that have generated the best results in terms of detecting opinion spam, and in particular false reviews, are often based on supervised or semi-supervised machine learning techniques that take into consideration both review and reviewer-centric information. The initial approaches were primarily linguistic, relying on simple textual features retrieved from review text, usually in the form of unigrams and/or bigrams. It is nearly impossible for a human reader to tell the difference between legitimate and non-credible information merely by reading it, especially in an era where the ability to write phony reviews is improving all the time. As a result, more successful multi-feature-based techniques have been proposed, which use a variety of features other than simply linguistic ones. Most of the fake review detection works are done for English fake reviews. So there is a scope for working in fake review detection in other languages.

Chapter 2

Literature Review

We have gone through many papers related to fake review detection. We are mentioning some of the important ones which align with our topic.

2.1 A Feature Analysis for Fake Reviews Detection using Supervised Machine Learning , International Conference on Data Science and Advanced Analytics

Julien FontanaravaJulien FontanaRava et. al [2] provides an analysis of review and reviewer-centric features. In this paper, they analyzed a lots of features those are already stated in other papers and they invented some new features. They tried to address singleton problem that is a user provide only one fake review on a system.

2.1.1 Problems They Identified

They thought that all existing solutions for detecting fake review detection rely on supervised machine learning techniques and distinct characteristics. Distinct characteristics mean features that are connected to a review and the reviewer who gave the review. They found that available solutions used a small set of features, used a small dataset. So all solutions were partial and review-dependant, they thought.

2.1.2 Their Solution Approach

In their solution approach, They emphasized features analysis. They selected both review and reviewer-centric features. They used a large dataset from YELP fake review dataset. They balanced the dataset by creating synthesized reviews from the minor class reviews. For the singleton problem, they used the burstiness feature and showed how the burst feature helps impact detecting singleton fake reviews. The burst feature is explained in the next section. They found an accuracy of 67.7% by using the burst feature where they got an accuracy of 54.2% without using a burst feature for detecting singleton fake reviews. For the overall solution, They used a supervised classifier based on Random Forests in their work. They got an accuracy of 80.6 using RFs classifier.

2.1.3 Limitations of Their Solution

They didn't analyze the reason behind the low accuracy of unsupervised machine learning techniques. They were also dependant on distinct features that is they didn't try to connect one reviewer with another reviewer or a community. For detecting singleton fake reviews, they weren't able to identify any reviewer-centric feature that may be accountable for this solution.

2.2 Fake Reviews Detection using Supervised Machine Learning,International Journal of Advanced Computer Science and Applications

Ahmed M. Elmogy et al [3] propose a machine learning approach to identify fake reviews in their paper. In addition to the reviews' feature extraction approach, this study uses different features engineering techniques to extract diverse behaviors of the reviewers.

2.2.1 Problems They Identified

Analyzing different papers they found, the successful fake reviews detection lies in the construction of meaningful features extraction of the reviewers. Fake reviews are usually detected not just by their category, but also by specific qualities that aren't directly related to the content. Text and natural language processing (NLP) is commonly used to create review features. Fake reviews, on the other hand, may need the development of additional characteristics related to the reviewer himself, such as the review time/date or his writing styles.

2.2.2 Their Solution Approach

Both the qualities of the reviews and the behavioral aspects of the reviewers are taken into account in their solution. The suggested method is evaluated using the Yelp dataset. In the developed technique, many classifiers are used. In the developed technique, the Bi-gram and Trigram language models are applied and contrasted. The findings show that the KNN(with K=7) classifier beats the other classifiers in detecting false reviews. In addition, the results reveal that taking into account the reviewers' behavioral characteristics increases the f-score by 3.80%.

2.2.3 Limitations of Their Solution

The limitations for this paper are that they implemented behavioral features but they have not considered the behavioral features of all reviewers. Moreover, they explained that the more is the behavioral features taken into consideration the more is the accuracy but they implemented just a few.

2.3 A Framework for Fake Review Detection in Online Cconsumer Electronics Retailers, Information Processing Management

Rodrigo Barbado et. al [4] proposes a feature framework for detecting fake reviews that have been evaluated in the consumer electronics domain in their paper. They divided their tasks into 4 different steps. First, they created a dataset using scraping techniques for different cities. Then they defined a true framework for fake review detection. After that, they developed a classification method and they analyzed the results for each of the cities.

2.3.1 Problems They Identified

They have found that no prior research has been done in the online consumer electronics domain where previous research was based only on hotel or restaurant reviews only. Analyzing different papers they have found user-centric features are more helpful in detecting fake reviews.

2.3.2 Their Solution Approach

To achieve their goal they propose a feature framework for detecting fake reviews in the online consumer electronic domain. They used scraping techniques to collect data from the web. They Mainly focused on user-centric features - Personal Features (P), Social Features (S), Reviewing Activity Features (RA) and Trusting Features (T). They used Logistic Regression, Decision Tree, Random Forest, Gaussian Naive Bayes, and AdaBoost They found AdaBoost works best for them with an 82% F-Score.

2.3.3 Limitations of Their Solution

The limitation of this paper is that they didn't propose any features to identify singleton fake reviews (One user provided one fake review). Moreover, Purchase information was not considered and it did not consider low rating fake reviews.

Chapter 3

Proposed Methodology

3.1 Dataset Creation

Dataset on any specific language is to be created. We intend to create a dataset on the Bangla language. Since there is no labeled dataset and having no way to label bangla fake reviews manually, we need to create a new dataset from an existing dataset that is already labeled. We chose YELP dataset that is already labeled and written in English. We converted it to Bangla. Reason behind chosing YELP dataset is that, some study found that YELP predictions are more accurate than others. But the algorithm they used is their trade secret.

3.2 Data Preprocess

Data preparation is the second stage in our proposed approach and it is one of the most important steps in machine learning approaches. Data preprocessing is essential since real-world data is never suitable for direct usage. To prepare the raw data from the Yelp dataset for computational operations, a series of preprocessing steps are performed in this study. It may be summarized as follows:

Figure 3.1: Methodology

3.2.1 Null Value Check

Null values indicate that there could be a value but you are not sure what it might be. They're placeholders until you've gathered enough information to populate the table field with a valid value. Any missing values are checked in this part.

3.2.2 Remove Stop Words

Stop words are widely used words (such as "the," "a," "an," or "in") that a search engine has been configured to disregard while indexing and retrieving results as the result of a search query. It is necessary to remove them to save space and save valuable processing time. We used NLTK for this purpose.

3.2.3 Remove Punctuation

Punctuation (. , ? ! ; : etc) are to be removed since these are not necessary.

3.2.4 Word Stemming

Word Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or the roots of words known as a lemma. For example "Eating" is a word and its suffix is "ing", if we remove "ing" from "Eating" we will get the root word which is "Eat". For this purpose, we will use the algorithm known as Snowball Stemmer.

3.3 Adding Features

3.3.1 Review Centric

Features that are built utilizing information from reviews are known as review-centric features. For Review Centric features we consider Text Statistics, Sentiment, Basic, Brust, and other features.

Text Statistics

Here we consider Review length, Number of capital words, Number of digits, and Number of nouns. Review length is the length of a review in terms of words. Number of capital words is the number of words where all the letters are uppercase. Digits include numbers like 1,2,3... and number of nouns indicating how many nouns are there in a sentence.

Sentiment

Sentiment analysis tries to quantify a speaker's or writer's attitude, sentiments, assessments, attitudes, and emotions using a computational approach of subjectivity in a text. For sentiment analysis, we consider Positive, Negative, Neutral, and Compound sentiment.

Basic

For basic features, we consider Rating and Rating Deviation[5]. The rating is the rating provided to the entity in the evaluation, expressed as a numerical number within a certain range(e.g.1-5 'stars'). Rating deviation is the difference between the entity's average rating and the evaluation offered in the review.

Burst

When there is a sudden concentration of reviews in a period, it is said that reviews for an entity are 'bursty.' The rapid popularity of the entity being evaluated, or spam assaults, might cause these review bursts. Because it has been demonstrated that reviews in the same burst tend to have the same type[6]. Evaluating the nature of the burst may quickly detect groupings of false reviews. If the average rating associated with an entity in a review (in a specified time window) differs considerably from the entity's average rating (in general, it declines, for example passing from 3.5/5 to 2/5), the review is more likely to be fraudulent. This assumption holds for all types of reviews, however, it has proven to be very useful in improving the identification of singleton phony reviews, which is extremely difficult to identify without taking burstiness into account.

Others

Here Review Density, Mean Review Deviation, Early Time Frame[5] are taken into consideration. Review Density is that, on a particular day of publication, the number of reviews for a certain entity. Mean Review Deviation is the difference between an entity's average rating on a given day and its average rating on that day. The Early Time Frame is the time it takes for a certain entity's first review to be posted. Spammers frequently review early to maximize the influence of their (false) thoughts on the audience.

3.3.2 Reviewer Centric

Reviewer-centric features consider information related to how users behave in a social network such as Yelp and which information users provide. This feature set is made up of features that are relevant to the behavior of the reviewers. This allows users to evaluate the behavior of users in general while submitting reviews, rather than only the text and meta-data linked with a review, which is restricted for classification.

Textual

For textual feature Maximum Content Similarity and Average Content, Similarity is taken into account. Maximum Content Similarity is the evaluation of the maximum similarity over the user's reviews whereas Average Content Similarity is the evaluation of the average similarity over the user's reviews.

Rating

For rating purposes, we include the Total number of reviews of a user and Average rating deviation from the entity's average rating. Average rating deviation is the average of a user's ratings assigned in reviews that is frequently considerably different from the mean of an entity's rating.

Temporal

In the Temporal feature, we consider the Activity time of a user[7], Maximum reviews per day[8], Date variance, Product purchase information, and Device information. Activity time of a user is the difference of timestamps of the last and first reviews for a given reviewer. Date variance is the squared deviation of the timestamps in which a user posts his or her reviews from the mean timestamps. Variance has been included as a temporal characteristic to better characterize how reviews for a certain user are dispersed over time frames. Product purchase information is about whether a person has posted a review by buying the product or just posted a review without even buying. Device information is the information of the device which may be device id or IP addresses so that it can be tracked whether a person cannot create multiple accounts to post a review for a single product.

3.4 Handling Unbalanced Data

When doing supervised classification, one of the key difficulties that must be addressed is unbalanced data. For handling the unbalanced data we would use Synthetic Minority Oversampling Technique oversampling(SMOTE)[9], which means Instead of duplicating examples from the minority class, SMOTE synthesizes new examples from the minority class.

3.5 Vectorization

We need some clever strategies to turn text input into numerical data, which are known as vectorization. Since machine learning classifiers don't work with raw words, we need to convert these words into numbers. That's what vectorization does. We will be using a countvectorizer and later on, we will use TF-IDF vectorization and Word2Vec.

3.6 Classification

A predictive modeling task in which a class label is anticipated for a given example of input data is referred to as classification. We will use different classification algorithms and we will analyze which one works best for us. In this reaserch, we used four classification models, those are given below,

- Logistic regression
- KNN
- Random forest
- Multinomial naive bayes

Chapter 4

Results

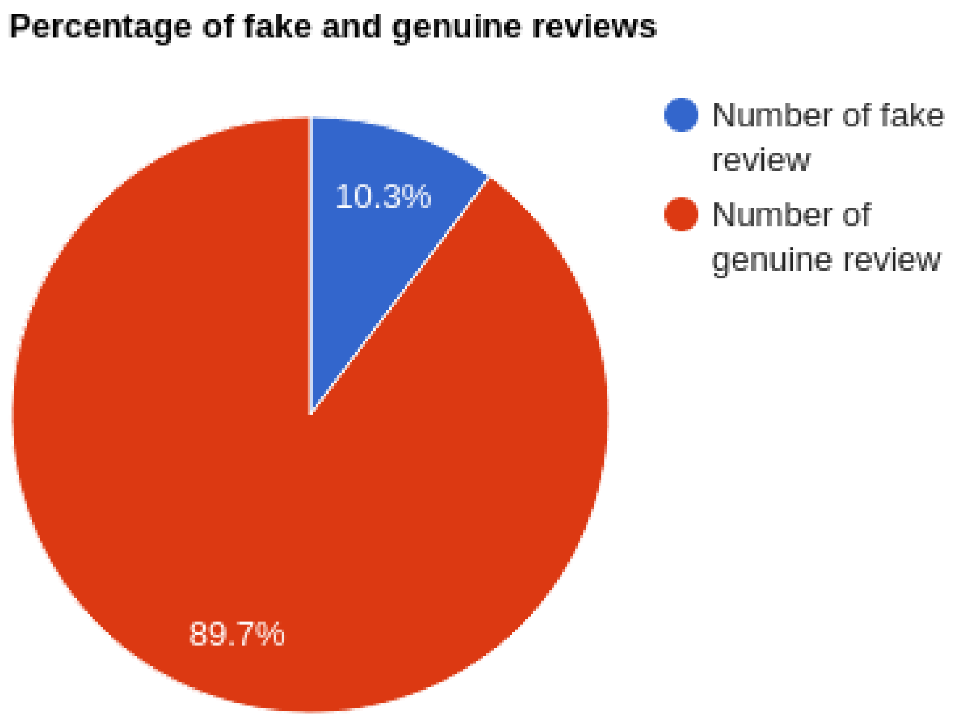


Figure 4.1: Percentage of Fake and Genuine Reviews

Before going into actual implementation, we've implemented an existing solution. For this, we've selected a YELP fake review dataset. Yelp.com is a popular online review site that filters fake reviews. Yelp's filter has also been claimed to be highly accurate by a study in BusinessWeek (Weise, 2011). That's why we've selected YELP's dataset. But Yelp's filter method is its trade secret. So, to find better accuracy, we wanted to find features that may be countable for detecting fake reviews. From the dataset, we can see there are a total of 359052 reviews where 322167 reviews are genuine and 36885 reviews are fake. From fig: 4.1, we can see that 10.3% of reviews are fake, which is alarming. Initially, we had a review and metadata dataset, then we merged these two

	Review_id	Product_id	Review_Date	Review_Text	Rating	Label
0	923	0	2014-12-08	The food at snack is a selection of popular Gr...	3	-1
1	924	0	2013-05-16	This little place in Soho is wonderful. I had ...	3	-1
2	925	0	2013-07-01	ordered lunch for 15 from Snack last Friday. Â...	4	-1
3	926	0	2011-07-28	This is a beautiful quaint little restaurant o...	4	-1
4	927	0	2010-11-01	Snack is great place for a Â casual sit down l...	4	-1
...	...	...	...	...	...	...
359047	161146	349	2014-02-06	I'm very spoiled with Pizza. Really, I have tr...	5	1
359048	116424	349	2014-01-31	Can't say enough good things about this place....	5	1
359049	161147	349	2014-01-30	Had a great dinner here- fantastic pizza, the ...	5	1
359050	97930	349	2014-01-25	Great foods and great drinks, they have even p...	5	1
359051	5260	349	2014-01-25	Pizza Loves Emily and I love Emily's pizza. Th...	5	1

359052 rows × 6 columns

Figure 4.2: Initial Dataset

	Review_id	Product_id	Review_Date	Review_Text	Rating	Label	cleaned_review
0	923	0	2014-12-08	The food at snack is a selection of popular Gr...	3	-1	[the, food, snack, select, popular, greek, dis...
1	924	0	2013-05-16	This little place in Soho is wonderful. I had ...	3	-1	[this, littl, place, soho, wonder, i, lamb, sa...
2	925	0	2013-07-01	ordered lunch for 15 from Snack last Friday. Â...	4	-1	[order, lunch, 15, snack, last, friday, â, on,...
3	926	0	2011-07-28	This is a beautiful quaint little restaurant o...	4	-1	[this, beauti, quaint, littl, restaur, pretti,...
4	927	0	2010-11-01	Snack is great place for a Â casual sit down l...	4	-1	[snack, great, place, â, casual, sit, lunch, e...

Figure 4.3: Preprocessed Dataset

into one fig: 4.2 . Since each review is raw data, so we need to preprocess it. For this, we first check for null values or any missing data. But there was no null value or missing data. Stop words are "a", "an", "the", "of", "in", etc. These words don't carry any useful information while detecting fake reviews. So we remove these stop words. Although punctuations may carry some good information, we also remove this for simplicity. Then we did word stemming. Word stemming is making any word to its original word such as 'Stemming' will be 'Stem'. Word stemming is important because while taking a word as a feature, we don't need to take all forms of a word as a feature rather than taking its original word. That is taking both 'Stemming' and 'Stem' as a feature, we can take only 'Stem' as a feature. After completing this preprocessing, we can see the result in fig: 4.3.

After that, we tried to implement some of the features such as number of capital words, number of digits, number of nouns, length of each review, average user rating, and average product rating. Average user rating can be calculated as follows,

$$Average\, user\, rating = \frac{total\, rating\, of\, a\, user}{total\, number\, of\, rating\, of\, a\, user}$$

$$Average\, product\, rating = \frac{total\, rating\, of\, a\, product}{total\, number\, of\, rating\, of\, a\, product}$$

Rating	Label	cleaned_review	number_of_cap_words	number_of_digits	number_of_nouns	review_length	avg_user_rating	avg_product_rating
3	-1	['the', 'food', 'snack', 'select', 'popular', ...	0	1	11	24	4.435897	4.009524
3	-1	['this', 'littl', 'place', 'soho', 'wonder', '...	0	0	15	28	3.000000	4.009524
4	-1	['order', 'lunch', '15', 'snack', 'last', 'fri...	0	1	14	23	4.000000	4.009524
4	-1	['this', 'beauti', 'quaint', 'littl', 'restaur...	0	0	22	53	4.000000	4.009524

Figure 4.4: Some Features Added

After adding these features, results can be seen in fig: 4.4.

avg_product_rating	user_total_review	neg_sentiment	pos_sentiment	neu_sentiment	compound_sentiment
4.009524	39	0.050	0.203	0.747	0.6486
4.009524	1	0.161	0.118	0.720	-0.1280
4.009524	2	0.055	0.253	0.692	0.7717
4.009524	1	0.035	0.203	0.762	0.8910

Figure 4.5: Sentiment Features Added

Then we added a new feature named the total number of reviews of a user. Then we tried to implement sentiment features. Sentiment features can be divided into four like positive sentiment, negative sentiment, neutral sentiment, and compound sentiment. Positive, negative, or neutral sentiment values determine how much a review contains positive, negative, or neutral sentiment. Compound sentiment can be found by adding positive, negative, and neutral sentiment values and then normalizing it between -1 and +1. Compound value -1 means the review is extremely negative and +1 means the review is extremely positive. We implement this feature using the VADER library of python. Results can be seen in fig: 4.5.

From the dataset, we can see the dataset contains around 10% data belonging to one class and the rest of the data belonging to another class. So there was a huge data unbalance. Since unbalanced data can lead to false results so we needed to balance the dataset first. To balance our dataset, we can use either oversampling technique in minor classes and the undersampling technique in major classes. If we undersample our data then there is a high probability that we can remove a lot of useful information. So we oversampled our data in minor classes. While oversampling, we can use either the random oversampling technique or the synthetic oversampling technique. Random oversampling is simply duplicate examples from minor class examples. But synthetic oversampling doesn't duplicate examples but it creates synthesized examples from minor classes, so we used a synthesis oversampling technique which was done using the SMOTE library of python.

Since machine learning classifiers don't work with raw words, we need to convert these words into numbers. That's why we need vectorization for this. Till now we've used 'Count vectorization' for that. It is pretty forward vectorization which counts the number of appearances of a word feature. We are willing to use TF-IDF vectorization and Word2Vec later on. Before vectorization, we first take the most common words between the two classes. Then while vectorization we remove these words to be a feature. It is important to know that we used the words bigram and trigram. Word bigram means taking two words as a feature and trigram mean taking three words as a feature. For example, We have a sentence like 'Product was good and nice.', then after preprocessing, bigram features would be 'Product good', 'good nice'. And trigram features would be 'Product good nice'. We limited the number of features that can't be greater than 15000.

Finally, we trained a classifier model that is logistic regression. By using this model, we've found an accuracy of 62.63%. The reason behind this low accuracy was using a small subset of features. We omitted a lot of important features but we're going to implement these features later for finding better accuracy.

From YELP dataset, we converted 100 thousand data into Bangla for our final research, where 70 thousand data was for bangla genuine reviews and 30 thousand data was for bangla fake reviews. We can see this by a pie chart in Fig:4.6. Among these 100 thousand data, we used 60 thousand data for our final implementation because of time and system limitaitions.

We first preprocess it, then added some features, those are given below,

- Content similarity
- Number of digits
- Rating deviation
- Burst feature
- Review length
- Average user rating
- Average product rating
- User total review
- Negative sentiment
- Positive sentiment

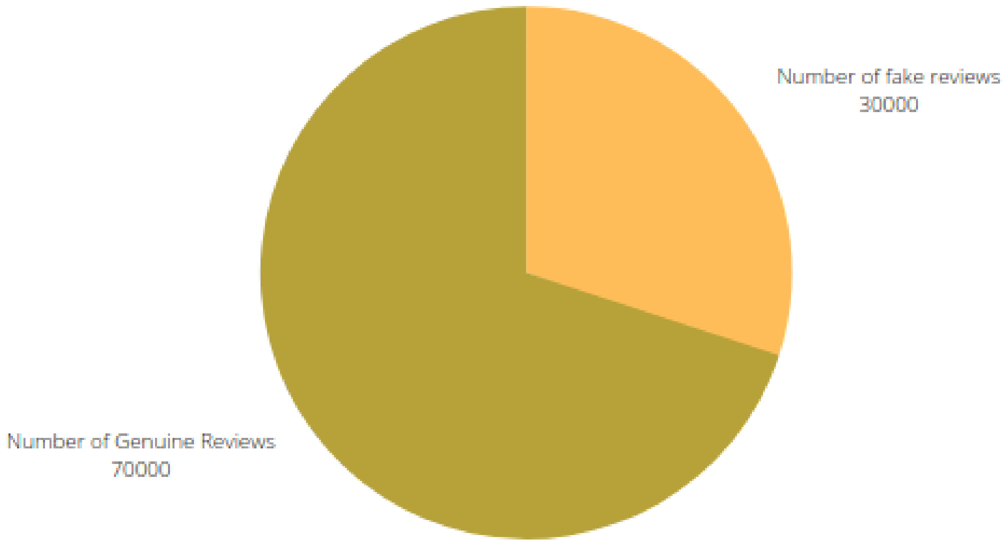


Figure 4.6: Bangla converted data classes

- Neutral sentiment
- Compound sentiment

After added those features, we vectorized it like previous one and finally we trained machine learning models.

By using Logistic Regression, we found the result in Fig:4.7

```
              precision    recall  f1-score   support

          -1       0.63      0.71      0.67      7466
           1       0.67      0.59      0.63      7532

    accuracy                           0.65     14998
   macro avg       0.65      0.65      0.65     14998
weighted avg       0.65      0.65      0.65     14998

array([[5287, 2179],
       [3083, 4449]])
```

Figure 4.7: Result for Logistic Regression

By using Random Forest, we found the result in Fig:4.8

By using KNN, we found the result in Fig:4.9

	precision	recall	f1-score	support
-1	0.76	0.75	0.76	7466
1	0.76	0.77	0.76	7532
accuracy			0.76	14998
macro avg	0.76	0.76	0.76	14998
weighted avg	0.76	0.76	0.76	14998

```
array([[5620, 1846],
       [1747, 5785]])
```

Figure 4.8: Result for Random Forest

	precision	recall	f1-score	support
-1	0.62	0.63	0.62	7466
1	0.63	0.62	0.62	7532
accuracy			0.62	14998
macro avg	0.62	0.62	0.62	14998
weighted avg	0.62	0.62	0.62	14998

```
array([[4685, 2781],
       [2869, 4663]])
```

Figure 4.9: Result for KNN

By using Multinomial naive bayes, we found the result in Fig:4.10
Overall result analysis can be seen by a bar chart given in Fig:4.11

```
              precision    recall  f1-score   support

          -1       0.68      0.60      0.64      8407
           1       0.56      0.64      0.59      6591

    accuracy                           0.62     14998
   macro avg       0.62      0.62      0.62     14998
weighted avg       0.62      0.62      0.62     14998

array([[5063, 2403],
       [3344, 4188]])
```

Figure 4.10: Result for Multinomial naive bayes

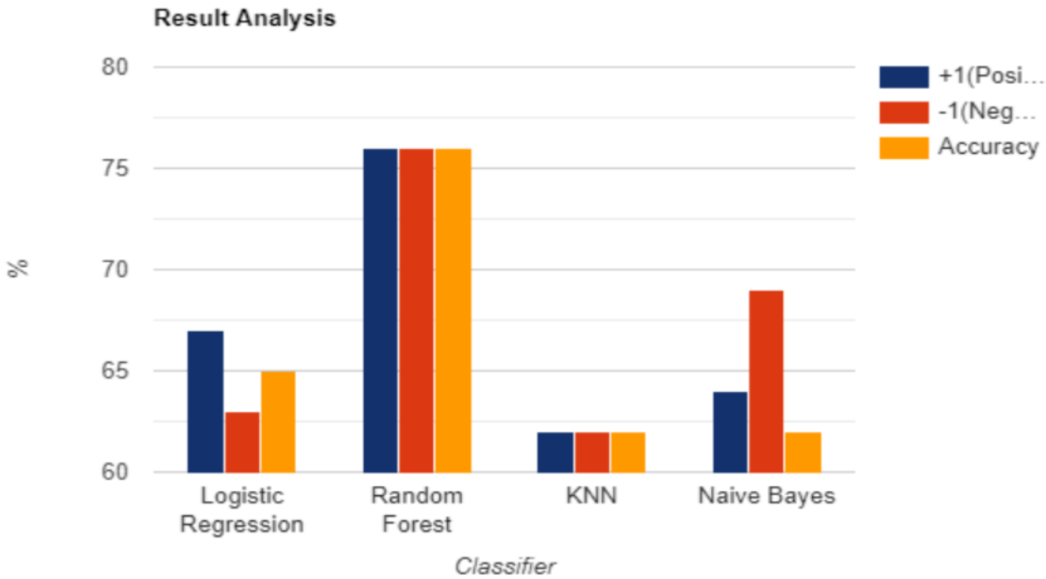


Figure 4.11: Bar chart showing overall result analysis for Bangla Reviews

Chapter 5

Contributions

Our contribution areas are given below,

- Creating bangla dataset
- Adding different review reviewer centric features
- Modify VADER (Valence Aware Dictionary and Sentiment Reasoner)
- Performance analysis for bangla fake review

5.0.1 Creating Bangla Datasete

As a key contribution, our first contribution was to create a bangla labeled dataset. As there is no bangla fake review dataset, and having no way to label a review dataset as a fake or genuine, we had to create a new dataset from a labeled dataset, we used YELP dataset which is already labeled and written in English, we converted it to bangla.

5.0.2 Adding Different Review and Reviewer Centric Features

Then our second contribution was to preprocess it and adding review and reviewer centric features so that we can get expected results.

5.0.3 Modify VADER

While adding features, we got a new feature called sentiment score value for each review. It basically finds out how much positive, negative or neutral a review is. We tried to use vader for that, but it wasn't working properly for bangla reviews. So we need to modify it so that we can get expected results.

5.0.4 Performance Analysis

And finally, we analyze the performance of different machine learning models to see the capability for detecting bangla fake reviews.

Chapter 6

Challenges we faced

To do the tasks given in Chapter:5, We had to face lots of challenges. Some are given below,

- Dataset selection
- Dataset conversion
- Feature selection
- Using VADER (Valence Aware Dictionary and Sentiment Reasoner)

6.0.1 Dataset Selection

First challenge was to select dataset. There are few datasets available, but we've select YELP dataset because some study found YELP predictions are more accurate than others, but the algorithm they used is their trade secret.

6.0.2 Dataset Conversion

After selecting dataset, we needed to convert it to bangla. Here we faced lots of challenges, like daily conversion limits, while converting, if electricity or network service interrupt, then whole things need to restart again. We overcome daily conversion limit by using open source python library deep-translator, which is free and unlimited. Then we divide our main task to sub tasks, so that any failure can't cause lots of time waste.

6.0.3 Feature Selection

Next challenge was to select features, we've collected as many features as we can. But while analyzing performance, we saw some features reduces the performance. For example, maximum reviews per day was decreasing the performance.

6.0.4 VADER

While using vader, we analyse that it takes lots of time and it produces almost same score for each review. We analyse that it was taking lots of time to convert bangla review to English review. That's why we were getting daily conversion limit again. So we modified it internally that don't need to convert anything. And that was also performing well for bangla review and it was working faster as well.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

We've reviewed some literatures to find the key features those are effective to detect fake reviews. We also tried to find out the scope of future work. Till now, we implemented an existing solutions using logistic regression and we got an accuracy of 62.67%. Since we used a small subset of features, that leads to this accuracy. We are going to implement more features and analyze more classifier.

7.2 Future Direction

- Singleton fake review analysis and detection.
- Detecting fake reviews written in other languages.
- Purchase, device and community features will be taken into consideration.

References

[1] M. Loiselle, “3 important statistics that show how reviews influence consumers,” 2021. [Online]. Available: https://www.dixa.com/blog/3-important-statistics-that-show-how-reviews-influence-consumers/

[2] J. Fontanarava, G. Pasi, and M. Viviani, “Feature analysis for fake review detection through supervised classification,” pp. 658–666, 2017.

[3] A. Elmogy, U. Tariq, A. Mohammed, and A. Ibrahim, “Fake reviews detection using supervised machine learning,” *International Journal of Advanced Computer Science and Applications*, vol. 12, 01 2021.

[4] R. Barbado, O. Araque, and C. Iglesias, “A framework for fake review detection in online consumer electronics retailers,” *Information Processing and Management*, vol. 56, 03 2019.

[5] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, “Spotting opinion spammers using behavioral footprints,” pp. 632–640, 08 2013.

[6] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, “Exploiting burstiness in reviews for review spammer detection,” *Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013*, pp. 175–184, 01 2013.

[7] A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance, “Fake review detection : Classification and analysis of real and pseudo reviews,” 2013.

[8] S. Rayana and L. Akoglu, “Collective opinion spam detection: Bridging review networks and metadata,” p. 985–994, 2015. [Online]. Available: https://doi.org/10.1145/2783258.2783370

[9] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, “Smote: Synthetic minority over-sampling technique,” *J. Artif. Intell. Res. (JAIR)*, vol. 16, pp. 321–357, 06 2002.