National College of Ireland

Learning to detect fake online reviews using readability tests and text analytics

MSc Research Project
Data Analytics

Siddhanth Chandrahas Shetty
Student ID: x17164036

School of Computing
National College of Ireland

Supervisor: Dr. Vladimir Milosavljevic

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Siddhanth Chandrahas Shetty
Student ID:	x17164036
Programme:	Data Analytics
Year:	2018-19
Module:	MSc Research Project
Supervisor:	Dr. Vladimir Milosavljevic
Submission Due Date:	12/08/2019
Project Title:	Learning to detect fake online reviews using readability tests and text analytics
Word Count:	7817
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	11th August 2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Learning to detect fake online reviews using readability tests and text analytics

Siddhanth Chandrahas Shetty
x17164036

Abstract

A customer highly relies on reviews when buying any product online, hence playing a crucial part in the customer's decision-making process. With the rise of online communities and portals, millions of reviews are getting posted and determining the credibility of them with such a high volume data is difficult. Although it is essential to classify them, as it profoundly impacts the business. Due to its hidden nature, fake reviews are used by companies to increase their market strength, which is a matter of concern. Many studies have been conducted with respect to this domain, where different statistical and textual analysis was performed to identify fake and genuine reviews. In this research, we propose the use of readability tests as features in combination with other general ratings and textual features on restaurant reviews datasets from Yelp for online spam review detection. We use supervised machine learning techniques such as Naïve Bayes, XGBoost, AdaBoost, and Gradient Boosting Machine for the classification of reviews using the mentioned feature sets. The results by the models are promising and displays the effectiveness of the proposed models in detecting fake reviews.

Keywords— *Data Mining, Machine Learning, Text Mining, Natural Language Process(NLP), Opinion spam, Fake review, XGBoost, Naïve Bayes, AdaBoost, Gradient Boosting Machine(GBM)*

1 Introduction

In today's world, the internet has become an integral part of our lives. Every person in the world uses the internet for some or the other purpose. Online purchasing of products has been on the rise since the inception of E-commerce, and online product reviews play a significant part in the e-commerce business. Due to E-commerce websites' ease of use, people are purchasing goods and services from the comfort of their houses. Online Reviews helps a customer to decide about buying a product. Since there are lots of similar product available in the market, it becomes difficult for the individual to rely upon only the specifications of the products (Banerjee et al.; 2015). Reviews are said to be an unbiased opinion of an individual about the personal experience with a particular product, hence playing a significant part in influencing the consumers buying behavior (Rastogi and Mehrotra; 2017).

Nevertheless identifying the authenticity of the reviews is essential, as there has been a rise in review spamming and spammers who are spreading false information over the internet. By generating fake reviews, a users perception is easy to manipulate regarding any product or services. As a consumer, we must understand which reviews are truthful and real personal opinions. Fake reviews are often used by companies to increase the online visibility of their

products and services. Also, some use it hamper the reputation of their immediate competition to stay atop in the market. Individuals also use fake reviews and regularly post it or spam to stay afloat and have a broader reach in the internet community. It is known as opinion spamming, where incentives are provided to individuals to write fake reviews having biased opinions about a particular product (Jindal and Liu; 2008). These reviews are written in such a way that it appears to be authentic when in actual they are not, this misleads the readers' views, making them inclined to a product which may not be suitable for their needs. Also, with the increase in the volume of such reviews identifying the authenticity of the reviews and manually differentiating them has become very difficult (Banerjee et al.; 2015). Even though separating fake and genuine reviews is challenging, the writing pattern often provides specific insights which allow us to distinguish their type. Reviews related to purchasing or experiences which are authentic are easy to understand as compared to the fake ones written based on the imagination of the writer (Yoo and Gretzel; 2009). Spam reviews often have fewer specific details than genuine ones (Hancock et al.; 2007). Fake reviews are usually more exaggerated or overblown, whereas honest reviews are simple and correctly written (Zhou et al.; 2004).

With the rise of online shopping portals, many researchers have been working on finding new techniques for understanding how authentic is the information mentioned on these web portals, especially for TripAdvisor, Yelp, Amazon. Such researches and improvements are essential as there is a substantial widespread of opinion spamming. Many researchers have found the reliability of online reviews to be questionable. For e.g., Yelp has estimated that 20% of the reviews present on their website are faked by paid writers (Yao et al.; 2017). False information related to the products and services on these websites can cause huge ramifications as the customers regularly use them for recommendations (Fontanarava et al.; 2017). One of the sectors that are most sensitive to reviews is the hospitality sector because it directly influences the customer irrespective of whether the review is genuine or fake (Lee et al.; 2018). The impact of reviews in the hospitality sector is seen in the study conducted by R. Filieri (Filieri; 2016) and P. OConnor (OConnor; 2008). Many studies have been conducted on the lines of fake review detection, spam filtering, etc. using different data mining and analysis techniques. The study conducted by K.Lee, J.Ham, S.B.Yang and C.Koo (Lee et al.; 2018) use fsQCA or fuzzy set Qualitative Comparative Analysis method on reviews from Yelp to identify configuration combinations of fake and genuine reviews. By this technique, they try to find the patterns which tell us about the characteristics of the reviews, which helps them to differentiate between real and fake reviews

N. Jindal and B. Liu (Jindal and Liu; 2008) initially explored opinion spamming, the primary concern nowadays for many online web portals by concentrating more on duplicate reviews. Similar kind of studies to identify individual and group spammers was conducted by A. Mukherjee, B. Liu, and N. Glance (Mukherjee et al.; 2012). Yelp has its fake review filter mechanism and to understand its functioning A.Mukherjee, V.Venkataraman B.Liu and N.Glance (Mukherjee, Venkataraman, Liu and Glance; 2013) conducted a study and explored behavioral and linguistic characteristics of the reviews with the help of supervised machine learning methods. Similar studies exploring features of the reviews have been conducted in the past, making a noteworthy contribution in the domain. It is also essential to find out new techniques to improve the existing system for identifying spam reviews.

The above approaches talk about different available and curated features related to reviews, but still, there are features which have not been explored and used much. One such feature is the readability, which indicates us how difficult it is to read and understand the text. Different readability tests can be conducted on the text to know how difficult it is to read and understand. These tests provide numeric results, and higher the value means the text is easy to read and understand (O'Mahony and Smyth; 2010). We will discuss these tests in detail in the later sections. For this study, a labeled dataset containing thousands of restaurant review from Yelp (Mukherjee, Venkataraman, Liu and Glance; 2013) is considered. Yelp being a trusted source

makes use of crowd-sourced reviews for its search recommender system, makes it the primary reason for selecting this dataset for the study. In the experiment, multiple feature sets such as rating features, structural features, readability tests, and n-grams are used in different combinations to build a novel fake review detection model. For this classification purpose, different supervised tree-based boosting algorithms such as XGBoost, AdaBoost, Gradient Boosting Machine (GBM) is used along with the Naïve Bayes model. Furthermore, K-fold cross-validation approach is implemented along with hyperparameter tuning of the different models using the Random Search technique. This research aims at creating an effective and quantifiable model for fake online review detection using readability features along with the rating, structural, and TF-IDF feature weights of n-grams.

1.1 Research Question

To what extent, can the features obtained from different readability tests on review text assist in enhancing the performance of machine learning algorithms to detect fake online reviews?

1.2 Research Objectives

1. *Discuss and review the work done in the field of fake online review detection.*
2. *Implementing machine learning models with readability features along with other review & text related features, analyzing and evaluating the results.*

The rest of the report is organized into the following sections: Section 2 will describe the related work and literature which will largely explain the details about the different techniques previously used in the fake review detection domain, In Section 3 we will discuss in detail about the considered dataset and the methodology used for the experiment, Section 4 will discuss about the design and architecture of the whole project, Section 5 will give the details about the Implementation process of various proposed features and approaches used for building the model. In Section 6, we evaluate the model performances based on multiple parameters and finally, Section 7 will focus on the conclusion and the future work.

2 Related Work

Over the past few decades, multiple researchers have demonstrated their approaches fro detecting fake reviews and spammers. Also, it is evident from the recent literature that Machine learning algorithms are playing a very pivotal role in classifying fake and genuine reviews. In this section, previous studies related to this domain is discussed in detail. This section is further divided into subsections based on the categories that are considered, such as Section 2.1 Machine Learning and Fake Review Detection, Section 2.2 Application of Tree Boosting algorithms in various fields, Section 2.3 Application of Readability tests and Section 2.4 Feature Extraction techniques with Machine Learning.

2.1 Machine Learning and Fake Review Detection

These days, social media is used for different purposes, such as posting news articles, posting blogs, and videos to share personal opinions about various day to day topics. It has been widely used for posting online reviews about the personal experience of any product bought from any shopping portal online or elsewhere. Customer experiences have been directly shared on the web nowadays, and people are using it and making their perspective on the product or service based on these reviews. Since then, there have been cases where fake reviews or opinion spamming

are seen, which is a huge threat for any business who depend mainly on its online operations. This issue of opinion spamming was first explored by (Jindal and Liu; 2008), wherein they investigated this issue considering product reviews which are highly influential as it is rich in opinions and are extensively used by both consumers and the manufacturers. In the literature, spams reviews were divided into three different categories Untruthful opinions, reviews specific to brands, and Non-reviews, which is further divided into two sub-categories advertisements and other irrelevant reviews which have no opinion. Supervised learning methods such as Logistic Regression, Naive Bayes and SVM were used to detect Type 2 - brand-specific and Type 3 - non-reviews, Type 1 - untruthful reviews was detected using a model which was built by using duplicate spam reviews. Logistic Regression was found to be better in differentiating between Type 2 and Type 3 categories.

Another study by (Jindal et al.; 2010) concentrated on finding suspicious behavioral patterns of the reviewers, which indicates spam reviewing activities. Being a domain independent technique, it made use of the dataset containing reviews from Amazon.com for analysis, which led them to find many suspicious reviewers. Various expectations were specified representing the suspicious behaviour of the reviewer, deviations from these expectations were then used for calculating the measure of unexpectedness, which was ultimately used for identifying abnormal behaviour in the review writers. The result of the case study, which was used to display the proposed system's effectiveness was found by making use of two different condition rules to test the reviewer's unexpectedness in terms of confidence and support.

A similar use of behavioral features of spammers was seen in the study (Lim et al.; 2010), where they used these features for detecting the review spammers. A scoring method was proposed, which measures the degree of spam for each reviewer, which was then later implemented on the Amazon review dataset. The model approach was said to be user-centric and driven by user behaviour. The study identified different types of abnormalities in the reviewer's behaviour. Based on helpfulness votes, it was derived that the proposed approach performed better than the baseline method.

A further study was made by (Lau et al.; 2011), where for the first time a model was built for fake review detection by applying text mining method integrated with semantic language models. This was an unsupervised model which was capable of addressing the missing features issue in an individual review. The study is a large scale analysis to check review trustworthiness, which had many features such as the proposed semantic language model was capable of taking the substituted terms into account while estimating the review content similarity. In addition to this, the study also addressed the knowledge acquisition problem by creating a concept association mining technique for extracting context-sensitive concept association knowledge, which was then utilized by the proposed model to determine the concept substitutions in fake reviews. Though this work suggested different aspects into the spam review detection, it was not a full-fledged commercial system; instead, it provides a prototype for the development of a fully functional spam review detection system.

Detection of fake review in unison with detecting the review spammer can be seen in the study done by (Lu et al.; 2013). The study proposed a review factor graph model for solving both the problem statements. It seems to be one kind of a literature wherein fake reviews, and the fake review spammers detection was simultaneously performed in a single framework. A set of features was defined between the reviews and reviewers. The considerable challenge of incorporating different features of both the problems into a united framework was solved by defining the review factor graph model. The proposed model here outperformed various baseline methods such as Support Vector Machine(SVM), Logistic Regression(LR), Conditional Random Field(CRF). The model learning was performed using the max-sum algorithm, which incorporates the belief propagation for the purpose.

In another research (Ko et al.; 2017), online paid reviews for restaurants was investigated using supervised machine learning methods such as Support Vector Machine(SVM) and Logistic

Regression(LR). A huge set of features from contents and metadata was proposed for detecting paid reviews, for paid writer detection the behavior of the content writer was captured. The results were significant in both the tasks and surpassed the considered baseline methods. The study explored a plethora of features related to reviews and the review writer, which will help in future researches, also in our study, we will use certain features as mentioned here.

Similarly, studies such as (Shu et al.; 2017) used data mining algorithms for fake news detection, (Chowdhary and Pandit; 2018) used Random Forest Classifier and Naïve Bayes to classify fake and genuine reviews with the help of Term frequency and user review frequency as the features sets. The study (Li et al.; 2011) presented a dual learning method for identifying reviews spams. Supervised learning methods are used to understand the effect of various features for identification of spam reviews, and then a semi-supervised method is used to co-train the model with a large amount of unlabeled dataset. The approach here proved to be better in terms of performance over the baseline models.

2.2 Application of Tree Boosting algorithms in various fields

Boosting algorithms are the most widely used machine learning methods for solving data mining problems. These algorithms convert weak learners into active learners to attain optimum performance results. There are many boosting algorithms currently being used, and XGBoost is the one which is used in most of the solutions. It is relatively a new algorithms, widely used in Kaggle competitions and Machine learning hackathons, was introduced by Tianqi Chen(Chen and Guestrin; 2016) in 2016. In study (Zhang and Zhan; 2017), a successful application of XGBoost can be seen for the classification of Rock facies based on geological features and constraints. Another study(Zheng et al.; 2017) presented the use of XGBoost for evaluating feature importance based on certain similarities. A framework was built wherein feature weights were learned using XGBoost based k-means framework and overcame the issue of dimensionality limitations in clustering.

Application of XGBoost in the field of medical science can be seen in study (Torlay et al.; 2017), where it is used for classification of patients with epilepsy by analyzing language networks. The analysis was based on neurophysiological features such as cerebral region, hemisphere, language representation processing, and the performance of the model was measured using the AUC curve. The model proved to be of significant potential in the said classification task with the AUC mean score of close to 91. Use of XGBoost is extensive, and never-ending literature can be found where it is used to solve realtime problems. Another application of XGBoost was demonstrated in study (Gumus and Kiran; 2017) where it was used for forecasting crude oil prices. For the experiment, a dataset containing crude oil prices from February 2010 to May 2017 was considered and also for analyses price trends of gold, silver, and natural gas of the same period was considered.

Employee attrition was predicted using XGBoost method in study (Jain and Nayyar; 2018). Using organizational data where there is a high chance of data redundancy, a precision model was implemented for the employee attrition prediction using XGBoost. With the accuracy of close to 90%, the presented model was very efficient in the prediction. Work is done by (Chen et al.; 2017) proposes a classification model for Radar emitter based on weighted XGBoost method. Here, a large dataset having different types of features was considered to train the model; also, a smooth weight function was used to tackle the data deviation issue. The dataset was divided 70%, 10% and 20% for training, validation, and testing, respectively, and with the accuracy score of 98.3%, it outperforms the traditional baseline models.

Another boosting algorithm is AdaBoost; it works on the same principle of converting weak learners into a strong learner and improve the model prediction. The weak learners here create the single split or the decision and perform the classification assigning more weights to observations that are incorrectly classified. Application of AdaBoost for solving different problems

can be seen in many pieces of literature, the researcher in the study (Haixiang et al.; 2016) proposed an ensemble algorithm where AdaBoost is used in a framework of BPSO-AdaBoost-KNN to perform multi-class classification on a highly imbalanced dataset. Here the researchers tried to perform boosting and feature selection in parallel by proposing this ensemble algorithm. The BPSO was used as the feature selection method, and the AdaBoost-KNN combined was used for the classification of oil reservoirs by using a boosting by resample strategy, where KNN was the weak learner. The performance evaluation was carried out using a novel AUC area metric.

Another good application of AdaBoost can be seen in the research (Nayak et al.; 2016). Here AdaBoost with random forest as its base classifier is used for classification of Brain Magnetic Resonance(MR) images. The proposed system used three MR image datasets for validation, and a stratified cross-validation scheme was used for enhancing the model's generalization capabilities. The results claimed that the proposed scheme is nearly 99% accurate across all the validation datasets.

Another member from the family of tree-based boosting algorithms is the Gradient Boosting Machine(GBM) which was initially derived by (Friedman; 2001). It has been used by the researchers over the years for solving many data mining, one such example can be seen in the study (Touzani et al.; 2018) where it was used for predicting energy consumption for a commercial building. The experiment was conducted on a large dataset of 410 commercial buildings, and for improving the results, k-fold cross-validation approach with some modifications is used. The results were found to be better, and good predictive accuracy was also seen.

Use of GBM in the Medical field can be seen in study (Atkinson et al.; 2012), where it was used for predicting bone fractures. The model incorporated measurements of bone density, the geometry of bone via the images, and other features to improve the model prediction. The experiment was conducted on two groups of 322 women in total who are in their postmenopause period having different bone structural conditions. The overall results were found to be good, and a stronger fracture prediction model was developed.

Similar applications of the above mentioned algorithms can be seen in many cases providing astonishing results, which is why they are preferred over other traditional machine learning techniques. In this study, the application of these methods for solving classification problems will be seen, and the evaluation of their performance will also be carried out.

2.3 Application of Readability Tests

In this research, we propose the use of Readability tests on review text as features for fake review detection. Readability tests indicate the difficulty level of a text to read and understand. It describes the ease with which the document can be read. There are various tests which are used to measure the readability of a text such as Flesch Reading Ease, Flesch Kincaid grade level, Gunning fog index, SMOG score, etc. All these tests will be discussed in detail in the latter part of the paper. Use of readability tests has not been seen much in many domains, especially in spam detection and opinion spamming to be precise. In 1998, the study (Courtis; 1998) used specific readability tests on annual stock reports of different companies. The Flesch reading ease formula was used to measure the readability and check for the coefficient of variation.

Application of readability tests on product reviews can be seen in study (O'Mahony and Smyth; 2010). In this study, four readability tests, namely Flesch Reading Ease, Flesch Kincaid Grade Level, Fog Index, and SMOG index were used as features along with structural features for classifying helpful and unhelpful product reviews. Random forest was used as the classification model for the task, and the performance was found to be improving with the addition of all the readability features instead of single elements.

In another research (François and Miltsakaki; 2012), NLP and machine learning was used to try improving the readability formulas. Traditional formulas like the Flesch formula was compared against the measure built using the proposed approach. The study was done on a

corpus of the text of the French language. The experiment results were found to be better were the new readability formulas developed using the proposed approach outperformed the traditional ones. For our experiment, we will be using readability features, as mentioned in this project, and in addition to this, additional readability tests will be used as features.

The study (Si and Callan; 2001) worked on the statistical aspects of the readability metrics. Three most widely used readability metrics FOG, SMOG, and Flesch Kincaid were worked upon, and a hypothesis was derived were it was said that readability metrics would be more accurate when the information about the content in the document is incorporated in it. The study also proposed a new method for estimating that combines these readability features into statistical models. By considering the above literature, it can be seen that readability features can be used as parameters for text classification. Various tests exist which can be incorporated and fed to the model to gain more accurate results for the problems.

2.4 Feature Extraction techniques with Machine Learning

Various literature (Ahmed et al.; 2017), (Gilda; 2017), (Ahsan et al.; 2016) and (Li et al.; 2014) adopted TF-IDF(Term Frequency - Inverse Document Frequency) technique for feature vector creation so as to improve the performance and accuracy of various classifiers.

In (Li et al.; 2014), opinion spam detection of Chinese reviews is done. For the study reviews from Dianping.com, a Chinese review hosting site equivalent to Yelp is being used. For the classification, a supervised learning and a PU learning approach were proposed. For the supervised learning, uni-grams and bi-grams were used with TF-IDF for feature weighting. The dataset consisted of restaurant reviews from 500 Chinese restaurants in Shanghai, China. The result showed that the proposed PU learning method outperformed the traditional SVM algorithm and also was successful in detecting a large number of fake reviews from unlabeled data.

In the literature (Ahsan et al.; 2016), an approach of active learning for spam review detection based on TF-IDF feature weighting was proposed. Active learning is an interactive learning method where new examples are gathered by making queries to the user, instead learning it from the training examples. For the experiment, an unlabeled dataset from Yelp was used for both training and testing purpose. Linear SVM, Stochastic Gradient Descent Classifier and Perceptron classifier were used for active learning out of which the Linear SVM performed better than other methods with an accuracy of 88%.

(Ahmed et al.; 2017) used Machine learning and n-gram analysis for detecting fake news. For the experiment, two different datasets from different sources reuters.com and kaggle were used, with over 24,000 articles of political news. The dataset consisted of labeled columns which said whether the article is reliable or not and all the articles in the dataset consisted of more than 250 characters. In this research, along with various other features, TF-IDF and TF were used for constructing feature vectors of different n-gram sizes such as Uni-gram, Bi-gram, Tri-gram, and Four-gram. Initially in the experiment the effect of these feature vectors on different classifiers such as Support Vector Machines (SVM), Stochastic Gradient Boosting (SGB), Linear Support Vector Machines (LSVM), Decision Trees (DT), K-Nearest Neighbour (KNN) and Linear Regression (LT) was observed and 5-fold cross-validation approach was applied to each of them. Out of all the classifiers, LSVM performed the best with an accuracy of 92% and also TF-IDF feature extraction technique was found to be better than the TF method.

The study (Gilda; 2017) explored natural language processing for detecting fake or misleading news articles. It used a dataset consisting of news articles from Signal media and also sources of these articles were found from OpenSources.co and applied TF-IDF of bigrams along with probabilistic context free grammar (PCFG) technique to a corpus of about 11,000 news articles. This dataset was tested on multiple classification algorithms such as Support Vector

Machines, Stochastic Gradient Descent, Gradient Boosting, Bounded Decision Trees, Random Forest and the Stochastic Gradient Descent classifier outperformed other classifiers when fed with the TF-IDF bigrams giving an accuracy of 77.2%. This experiment shows that TF-IDF approach gives promising results and have a good prediction power, though it creates a certain level of doubt of it being robust to the changing way of news articles around the world.

TF-IDF is a very popular approach in text mining and information retrieval domain. From the above literature, it is understood how useful TF-IDF is for solving spam detection and text classification problems. In addition to the above mentioned literature, (Kaur and Kaur; 2017), (Thu and New; 2017), (Barbado et al.; 2019), (Martinez-Torres and Toral; 2019) also demonstrated the successful application of the TF-IDF approach for their respective problems. In our study, we will be using the TF-IDF feature weight approach of bi-gram and tri-gram for feature extraction.

3 Methodology

In this study, we followed the Knowledge Discovery Databases (KDD) methodology for acquiring useful knowledge from the data. It was defined by (Fayyad et al.; 1996) as ***"The non-trivial process of identifying valid, novel, potentially useful, and ultimately pattern in data".*** It consists of various steps such as Data Selection, Preprocessing, and Transformation of the data, Data Mining, and finally, evaluation of the results. Figure 1 illustrates the different stages of the KDD approach.

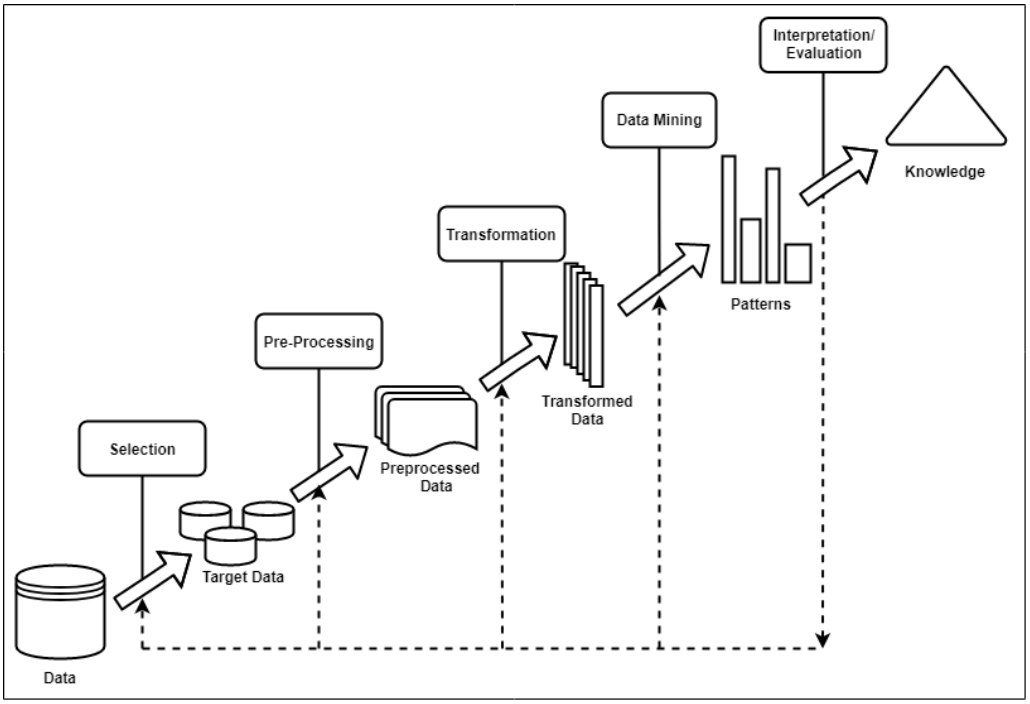

Figure 1: KDD Methodology (Fayyad et al.; 1996)

The rest of this section will explain the architecture of the applied methods in the study which briefly follows the KDD process life cycle.

3.1 Dataset Description

In this phase, for the classification of fake reviews, labeled dataset from Yelp[1] consisting of restaurant reviews has been used. This particular dataset was collected and is used as described by A.Mukherjee, V.Venkataraman, B.Liu, and N.Glance in (Mukherjee, Venkataraman, Liu and Glance; 2013).

The data consisted of two datasets having features relevant to the reviews and the restaurant that is reviewed. The review dataset contains around 67,000 entries and has 10 review related features. The restaurant dataset consists of 30 features with over 200,000 entries.

Total review count	**Fake**	**Genuine**	**Fake reviews %**	**Total reviewers**
67016	8301	58715	12.38%	34555

Table 1: Dataset Overview

Review Rating, Review Content, Total upvotes, Review Flag, etc. are the set of features directly used after preprocessing, additional features were generated using these input variables and a complete feature set was generated for the classification process.

3.2 Data Pre-processing

Before starting with the implementation of any data mining or machine learning process, it is essential to clean the data so that irrelevant data or noise is eliminated. This allows the models to perform better and present optimum results. In the initial stage, missing values in the dataset were checked and then dropped since there were a minimal amount of such entries in the dataset. Few of the review text entries were also found to be empty, were also dropped. After handling the missing values, irrelevant features were dropped from the dataset. To create the final dataset, the reviews and the restaurant dataset was merged with respect to the *restaurantID* column, which is common in both datasets.

For performing analysis on text, certain activities are required to be undertaken so that it is understandable to the model, and it can derive useful information from it. In our research, we undertook the following Text preprocessing activities:

- Removal of numeric values and punctuation from the reviews.
- Removal of special characters as the count of these characters will be extracted and used as a separate feature.
- The complete text of the reviews will be converted to lowercase for maintaining consistency in data. For example, the text *"FAke ReWIew DEtectiOn"* will be changed to *"fake review detection"*.
- Removal of stopwords from the text by using *stopwords* method from the *NLTK*[2] package in Python. Words such as *"the","a","an","in"* etc. are known as Stopwords. These words usually do not contain any information which adds noise to our data.
- *Lemmatization* - In this stage, the words are transformed back to its base form, for e.g. *"testing"="test"*, *"amazing"="amaze"*. It was preferred over stemming because stemming just cuts off the suffix e.g. *"amazing"="amaz"*. Lemmatization was performed using the *lemmatize* method from the *TextBlob*[3] package in Python.

[1] `http://www.yelp.com`

[2] `https://pypi.org/project/nltk/`

[3] `https://pypi.org/project/textblob/`

After the above text cleansing activities, the data was checked for class imbalance, where a considerable amount of class imbalance was observed, it can be seen in Figure 2. No of fake reviews are very minimal in number compared to the genuine ones, accounting only 12.38% of the whole dataset.

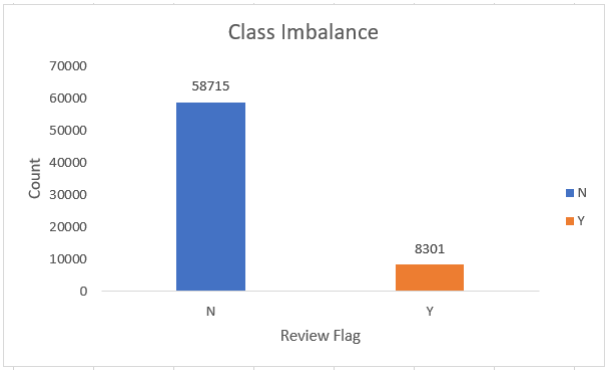

Figure 2: Class Imabalance Graph

With such a proportion of data, the analysis will not yield reliable results, appropriate steps need to be taken to tackle this problem. There are multiple techniques available for handling the class imbalance, we will be using the following techniques for individual approaches (Fontanarava et al.; 2017):

- ***Random Undersampling*** - This technique will be used for analysis on review text using n-grams. Here random samples will be selected from the majority class to match the count of the minority class. Figure 3 shows the Random undersampling result.
- ***SMOTE(Synthetic Minority Over-Sampling Technique)*** - It is an oversampling technique, where synthetic samples of the minority class is created equal to the majority class (Chawla et al.; 2002). Figure 4 shows the result after using SMOTE. This techinique was implemented using the SMOTE method from imbalanced-learn[4] package in Python.

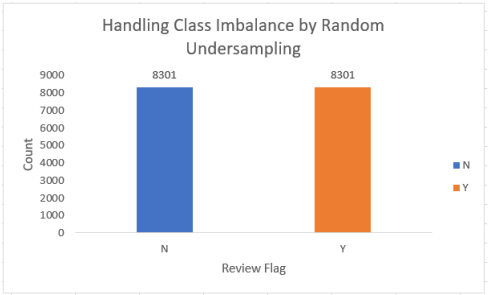

Figure 3: Random Undersampling

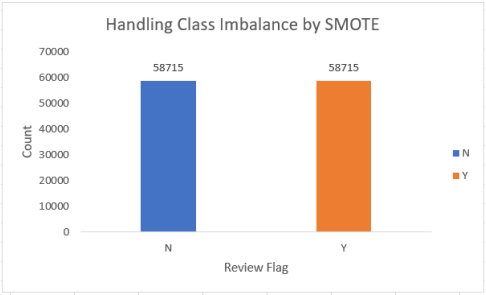

Figure 4: SMOTE

3.3 Feature Extraction and Transformation

Feature extraction is essential for text classification problems which have a considerable amount of data. Selecting relevant features helps in reducing the computational burden and enhance the accuracy of the classifier. In our study, multiple sets of features are considered and also

[4] `https://pypi.org/project/imbalanced-learn/`

propose the use of readability tests scores along with rating features and linguistic features such as TF-IDF using bi-grams and tri-grams. Following a set of features are used in this research:

1. **Rating Features**: These are the features containing ratings related to the restaurant and the reviews. Table 3 gives the details of these features.

Feature Name	Data Type	Description
Review Rating	Float	Rating of the individual review in the scale of 0 - 5
Total Upvotes	Integer	No of people in agreement with the review
Total Review Count	Integer	Total number of reviews received by the restaurant
Fake Review Count	Integer	Total number of fake reviews posted for the restaurant. It is filtered by Yelp
Average rating of restaurant	Float	Aggregate rating of the restaurant
Deviation from aggregate rating	Float	Deviation of the review rating from the aggregate restaurant rating

Table 2: Rating Features

2. **Structural Features**: These features are engineered from the existing review text so as to understand its influence. These features have proved to be important in many text classification problems (Li et al.; 2011)(O'Mahony and Smyth; 2010).

Feature Name	Data Type	Description
Review WordCount	Integer	Total number of words used in the review.
Total Character Count	Integer	Total number of characters used in the review.
Special Character Count	Integer	Total number of special characters used in the review.
Uppercase Character Count	Integer	Total number of uppercase characters used in the reviews.
Sentence Count	Integer	Total number of sentences used in the reviews.
Stopwords Count	Integer	Total number of stopwords used in the reviews.
POS tags count	Integer	No of different POS(Parts Of Speech) used in the review. In this research, we assume that the reviewers have a good command on the english language. We use counts of Noun, Verb, Adjective and Adverb as features for our classification model.

Table 3: Structural Features

3. **Text Features(n-grams)**: From the review text we extract the TF-IDF feature weights of bi-grams and tri-grams. This method assigns weights to the words in the document, which allows us to identify the unique words in the document. We used TF-IDF for feature weighting instead of TF, as it performs better(Li et al.; 2014). For executing this

task we used TfidfVectorizer package from the sklearn.feature_extraction package family in Python. Due to high dimensional features being generated, we used Chi-squared test for dimensionality reduction, allowing less computational time and optimum performance of the model.

4. **Readability Features**: Along with the above features, we propose an additional set of features using the readability tests(O'Mahony and Smyth; 2010). It is one of the interesting fields within Natural Language Processing which involves determining the readability of a text. These tests let us know how difficult is a particular text to read and understand. In our research, we will be using the following tests of readability:

    - ***Flesch Reading Ease(FRE)***: It calculates the reading ease of a text in the range of 1 to 100. Lower scores indicates that the text is harder to read.
    - ***Flesch Kincaid Grade Level***: This test indicates what US grade level of education is required to understand the text.
    - ***Gunning Fog Index***: The result of this test indicates the number of years of formal education(US grade level) required to understand the text.
    - ***SMOG score***: The result of this test indicates the years of formal education(US grade level) required to completely understand the text.
    - ***Automated Readability Index (ARI)***: This tests' results indicates the US grade level of education required for comprehending the text.
    - ***Coleman-Liau Index***: Using the Coleman-Liau formula, it indicates the grade level of education required to read the text.
    - ***Linsear write***: It indicates the US grade level of education required to read the text.
    - ***Dale Chall Score***: This test provides a numeric scale for measuring the comprehension difficulty of a reader on a particular text.

3.4 Data Mining Models

In order to distinctly classify fake and genuine reviews based on the labeled dataset from Yelp, a supervised learning approach is followed in the research. Since our other objective is to analyze the performance of boosting algorithms for text analysis, we consider ***XGBoost(XGB), AdaBoost(ADB) and Gradient Boosting Machine(GBM)*** for classification of reviews. The Boosting algorithms has always provided better results over traditional Machine Learning techniques (Zhang and Zhan; 2017), (Zheng et al.; 2017), (Haixiang et al.; 2016), (Atkinson et al.; 2012). Also, we will be using ***Naïve Bayes(NB)*** model for classification. Random Forest Classifier (RFC) used in the study (Mukherjee, Kumar, Liu, Wang, Hsu, Castellanos and Ghosh; 2013) will be the baseline model for our research.

3.5 Evaluation Metrics

The data is divided using the holdout approach of splitting dataset in 80:20 for training and testing the model, respectively. We perform Hyperparameter tuning for getting optimum results. For analyzing the results, similar parameters will be used as used in the previous related works. We evaluate the results of four different machine learning algorithms, namely XGB, ADB, GBM, and NB. For evaluating the performance of each of these models, we used six evaluation metrics, namely: ***Accuracy, Precision, Recall, F1 Score, AUC score*** and ***Kappa score***.

4 Design Specification

Figure 5 displays the architectural diagram of our research. In the first stage the data was gathered from the source, following that different preprocessing steps was undertaken such as eliminating missing values, normalizing the cases and other text preprocessing activities. Later, with the help of TfidfVectorizer package from Python we created bi-grams and tri-grams with TF-IDF feature weights. Due to large number of features being generated, we use Chi-square test to select the best bi-grams and tri-grams in terms of performance. The last step in the process is to train the classification models on the training set and predict the outcomes on the test set with 80-20 distribution of the total dataset. To achieve optimum performance of our models, we performed 10-fold cross validation and Hyperparameter tuning of our classification models.

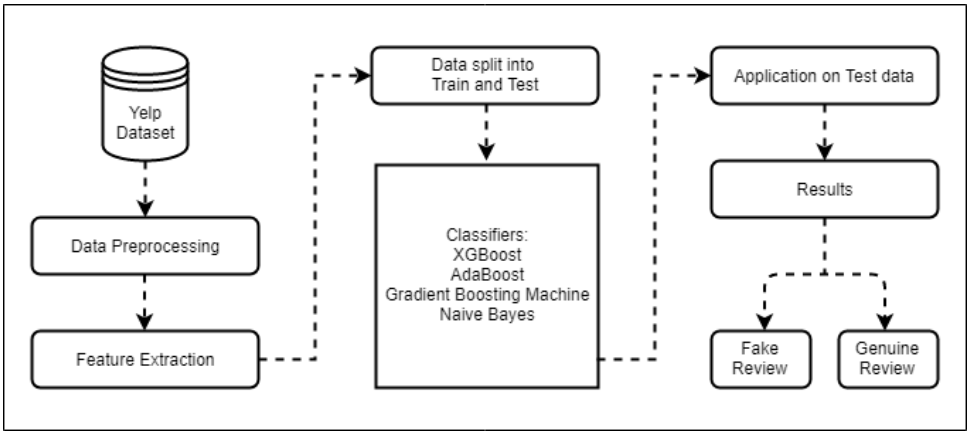

Figure 5: Architecture Diagram

5 Implementation

The section will provide details on how the implementation of the research was carried out to create an efficient spam review detection model. Also, it discusses the procedures undertaken for feature extraction and dimensionality reduction. For the implementation of the project, Python 3.7 was used, and Jupyter notebook was chosen as the Integrated Development Environment(IDE). Python was the default choice for implementation because of its ease in use and also because of the wide range of online support available it has from the active community. It is also the default choice for many Data Mining projects involving Natural Language Processing (NLP) because it provides a wide range of packages to perform NLP activities.

The data for the research contains labeled restaurant reviews from Yelp[5], collected and described by the researchers in the study (Mukherjee, Venkataraman, Liu and Glance; 2013). The database consisted of three different tables, and we used reviews and restaurant table for our research. These tables were imported into CSV files using SQLite[6] Database browser. Both the dataset were then imported as Dataframes and later was checked for any discrepancy (missing values). The reviews dataset consisted of additional labeled data, which were removed as it was not considered for the research. After basic data cleaning activities, both the datasets were then merged using the common column of *restaurantID*. The final dataset then consisted of total 67,016 reviews which were then used for performing the preprocessing and feature engineering tasks as explained in Section 3.2 and 3.3 respectively. After this task, the dataset was checked

[5] `https://www.yelp.com`

[6] `https://sqlitebrowser.org/`

for the class imbalance to avoid any biases in the data. It was found that approximately 12% of the reviews were fake, and the rest was genuine, which showed a huge imbalance in the predictive class. For handling this issue, we used the SMOTE technique for numeric features and Random Undersampling when using text features in the model.

Each review was cleaned for special characters, stopwords, numeric values, NA values, punctuation marks, etc. using relevant text preprocessing methods, as explained in section in 3.2. We also assured that all the required packages were installed and ready to execute different activities involved in the research. For e.g., Packages such as *pandas, numpy, seaborn* were used for basic data handling and preprocessing tasks, *textstat*[7] package was installed and used for extracting all the readability features. For performing classification using different machine learning methods packages such as *scikit-learn* was used for Naïve Bayes which is directly available within *sklearn* package, AdaBoost and GBM are present within *sklearn.ensemble* packages. We used *xgboost*[8] package which is independent of scikit-learn for implementing XGBoost algorithm.

Hyperparameter optimization helps in getting the best out of the classifiers in terms of the performance. There are many approaches that can be used for optimizing the parameters of the classifiers. For our research, we used the Random Search technique for hyperparameter optimization because of its efficiency over Grid Search in finding better classification models, and it also requires less computational time (Bergstra and Bengio; 2012). Parameters such as subsample, n_estimators, max_depth, learning_rate, etc. for XGBoost and GBM, n_estimators and learning_rate for AdaBoost. We also use Gaussian Naïve Bayes model where no Hyperparameter tuning is possible. For implementing this we used *RandomizedSearchCV* package from *sklearn.model_selection* in python. For creating n-grams vectors with Tfidf weights, *TfidfVectorizer* package from *sklearn.feature_extraction.text*, *SelectKBest* and *Chi2* packages from *sklearn.feature_selection* was used for implementing the Chi-square test.

We created three different feature sets consisting of textual and non-textual features. All the feature sets were then fed one by one to Naïve Bayes, XGBoost, AdaBoost and GBM classifiers for assessing the performance of these models. Before training and then testing the model, the dataset was split into training, and testing sets using the package *train_test_split* from *sklearn.model_selection*. Thus, it was divided into the proportion of 80:20, where 80% of the data was allocated for training the model and 20% for testing the model. To avoid overfitting of the data, we performed 10-fold cross-validation. By using the *predict* function accuracy of each classifier was then checked on test data. Finally, *confusion_matrix, precision, recall* and other evaluation metric packages were used for assessing the performance of the classifiers.

6 Evaluation

This research aims to build a fake review system by using various curated feature sets mentioned above, especially the readability tests. Also, we will evaluate the performances of different boosting algorithms on textual data and see how better do they classify the fake and genuine reviews. Once all the required features are formed, a supervised machine learning approach is undertaken. In our case, we specifically use boosting algorithms such as XGBoost, AdaBoost, and GBM for classification. These models are then trained with this data and using various evaluation metrics we compare the results of each of the model.

We use three different combinations of feature sets for our experiment, and the results are then evaluated against each other to obtain the best appropriate approach for this kind of problems.

[7] `https://pypi.org/project/textstat/`

[8] `https://pypi.org/project/xgboost/`

6.1 Experiment 1: Using Rating Features, Structural Features and Readability features

In the first feature set, we used rating features, structural features, and readability features. In all 25 different features where used in the initial experiment for classification of reviews. The class imbalance in the dataset was handled using SMOTE after which the data was divided into training and testing sets.

Rating Features + Structural Features + Redability Tests						
Model	Accuracy	Precision	Recall	F1 Score	AUC	Kappa
AdaBoost	0.85	0.85	0.84	0.84	0.85	0.7
XGBoost	0.93	0.96	0.88	0.92	0.92	0.85
GBM	0.92	0.97	0.87	0.92	0.92	0.85
NB	0.72	0.65	0.93	0.76	0.72	0.44

Figure 6: Only numeric features Model Results

As seen in Figure 6, XGBoost performed the best with this particular feature set, with the highest accuracy of 93% and highest F1 score of 0.92 same as the GBM model. On the other hand, Naïve Bayes displayed lower performance with accuracy with 72% and also NB clocked the lowest F1 score of 0.72. Also, by taking precision into consideration, XGBoost and GBM performed better with both scoring above 0.96 Precision value, and NB model scored low in the precision scale as well with only 0.65. AdaBoost as compared to other boosting algorithm performed significantly low with an accuracy of 85% and Precision values of 0.85. In terms of Recall, Naïve Bayes performed better with a Recall score of 0.93, XGBoost was next with a score of 0.88.

6.2 Experiment 2: Using Rating Features, Structural Features, Readability features and TF-IDF bigrams

For the second experiment, we used Bi-grams along with the numeric features considered in section 6.1. Selection of relevant bi-grams was made using the Chi-square test, which showed that using the top 4000 bigrams will give the optimum results. The dataset was then developed accordingly and was fed to the classification models. Initially, the class imbalance was also handled using the Random Undersampling technique, where random samples of the majority class were selected equal to the minority class samples. The results displayed in Figure 7 is after Hyperparameter optimization of these models.

Rating Features + Structural Features + Redability Tests + Bi-grams						
Model	Accuracy	Precision	Recall	F1 Score	AUC	Kappa
AdaBoost	0.75	0.69	0.91	0.78	0.75	0.5
XGBoost	0.75	0.7	0.89	0.78	0.75	0.51
GBM	0.76	0.71	0.86	0.78	0.76	0.52
NB	0.72	0.64	0.98	0.77	0.71	0.43

Figure 7: Numeric features and TF-IDF Bigrams Model Results

As seen in Figure 7, GBM performed the best with an accuracy of 76%, it also had a good recall score of 0.91. On the other hand Naïve Bayes showed a minimum accuracy out of all the

used models with 72%, although it had an astonishing recall score of 0.98 it lacked good precision with a score of 0.64 which was lower than all other models. With scores displayed in Figure 7, XGBoost, and GBM performance are found to be similar, but the main difference between these is the computational time as GBM requires higher computational time as compared to XGBoost. In terms of F1 Score, all the models seem to be performing in a similar way with the scores between 0.77 and 0.78.

6.3 Experiment 3: Using Rating Features, Structural Features, Readability features and TF-IDF trigrams

For the last experiment, tri-grams were used along with the numeric features as a single combined dataset. Similar steps were implemented as used in Section 6.2 and then was fed into the classifiers for training them. The results of our classifier are displayed in Figure 8.

Rating Features + Structural Features + Redability Tests + Tri-grams						
Model	Accuracy	Precision	Recall	F1 Score	AUC	Kappa
AdaBoost	0.75	0.7	0.87	0.78	0.75	0.5
XGBoost	0.76	0.7	0.88	0.78	0.76	0.52
GBM	0.76	0.71	0.86	0.77	0.75	0.5
NB	0.81	0.75	0.93	0.83	0.8	0.62

Figure 8: Numeric features and TF-IDF Trigrams Model Results

The Naïve Bayes model outperformed the boosting algorithms with an accuracy of 81% and the recall score of 0.93. Though the precision score of 0.75 was a matter of concern, it displayed a better Kappa score of 0.62, which was the maximum score out of all the models implemented for this dataset. XGBoost and GBM again showed similar performance scores with both having an accuracy 76% . The precision score remains low for both at 0.7. In this experiment, we could see that NB has outperformed the Boosting algorithms in terms of accuracy, precision, F1 Score, and Recall, also a good Kappa score was seen in NB.

6.4 Discussion

Below figure (Fig. 9) represents the comparison of all the proposed models using Experiment - 2 (Sec: 6.2) feature set and the baseline model (Mukherjee, Kumar, Liu, Wang, Hsu, Castellanos and Ghosh; 2013) in terms of accuracy, precision, recall and f1 score. It is evident from Fig. 9 that our proposed techniques outperform the baseline model. It can be observed that GBM and XGBoost specifically have better accuracy than other models with a score of 76% and 75% respectively. The computational time is the main difference between GBM and XGBoost, where the latter requires less time. Since our problem is of spam classification, we will concentrate more on the recall score. We can observe that the recall value of all the proposed models exceeds the baseline recall score, specifically Naïve Bayes is the best in this section with a recall score of 0.98. AdaBoost was next best with a recall score of 0.91 followed by XGBoost and GBM with scores of 0.89 and 0.86 respectively. In our case, recall is important because classifying few fake reviews as genuine will still be acceptable rather than classifying a genuine review as fake.

Similarly, in figure (Fig. 10) we will see the comparison of all the models with the baseline using Experiment - 3 (Sec: 6.3) feature set. In this feature set, we use different numeric features, including readability features along with the TF-IDF scores of tri-grams. As we can see in Fig. 10, the results achieved by our proposed models are better than the baseline model. It can be observed that Naïve Bayes performed better out of all the models with a maximum accuracy of

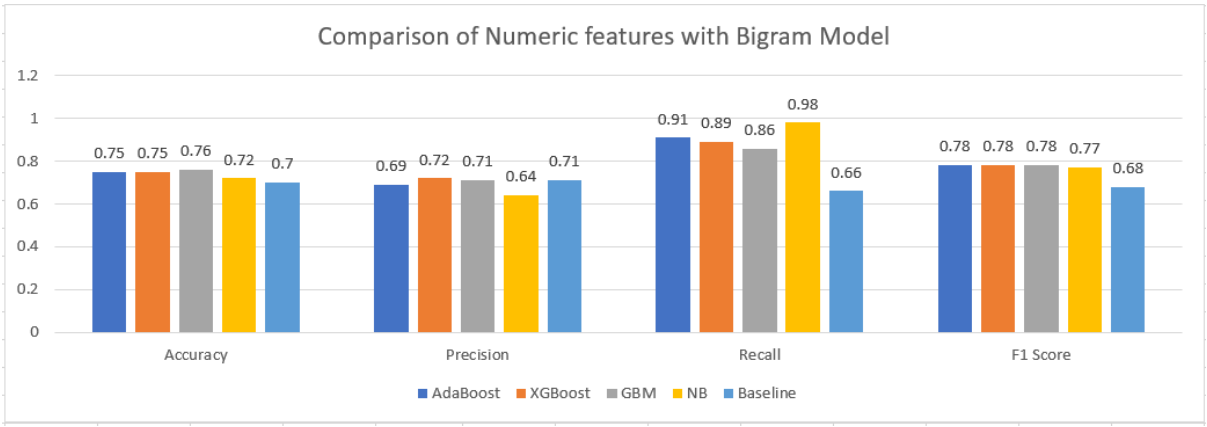

Figure 9: Result Comparison 1

81% and recall value of 0.93. Also, in terms of precision, NB was better with a score of 0.75. It can be observed that all the classification models showed better recall over the baseline model, and that too by a commendable margin. XGB had the second best recall score of 0.88, followed by AdaBoost and GBM with scores of 0.87 and 0.86 respectively.

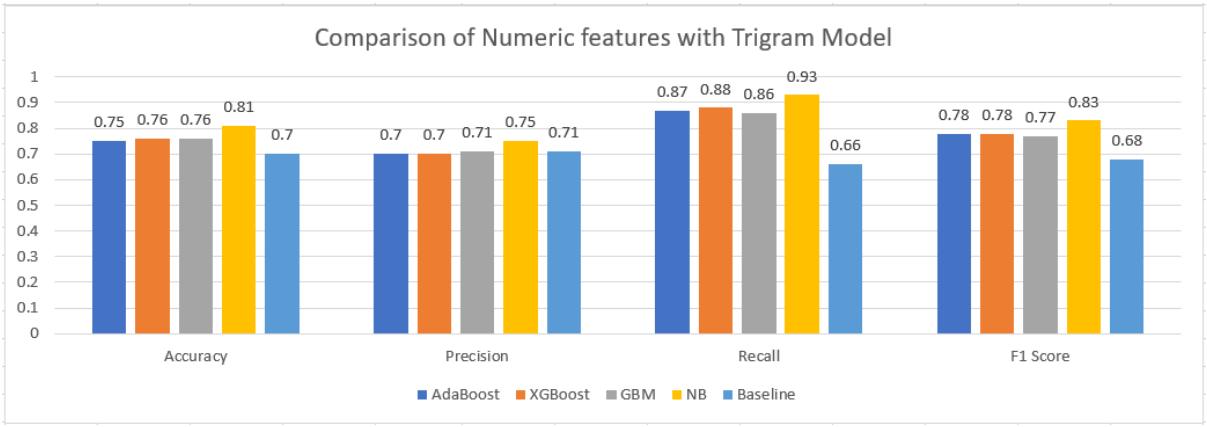

Figure 10: Result Comparison 2

From the above discussion, we can see that using readability features along with other features, was found to be effective over other approaches. We have used rating features as well, which is restaurant centric, apart from that all other features are general features which can be applied to reviews of the different domain for spam review detection.

7 Conclusion and Future Work

In this research, we propose an implementation of models for fake online review detection with readability tests as part of the feature set. Along with these features, we use rating features, structural features, and TF-IDF score for bi-gram & tri-gram. The results obtained from the above experiments were promising, showing that readability features are useful in detecting fake online reviews. Although this research accurately classifies fake and genuine reviews, still a certain amount of work is required to be done. We used eight different readability tests as features, in some cases, all the tests may not be necessary. We can use these features one by one in the classification models to check the best performing feature set within them. Secondly, we assumed that all the reviewers have a good command over English vocabulary while extracting

the POS tags count, which most of the times may not be the case as there can be tourists from other countries using their native language for writing the reviews. So language and grammar issue should be taken into consideration in the future works. In the n-gram section, we only used bi-grams and tri-gram, future works may include uni-grams and four-grams for better performance. This research primarily focuses on restaurant reviews, but by skipping the rating features which is only related to restaurants, remaining features along with some domain-specific features can be useful in online spam review detection

References

Ahmed, H., Traore, I. and Saad, S. (2017). Detection of online fake news using n-gram analysis and machine learning techniques, *International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, Springer, pp. 127–138.

Ahsan, M. I., Nahian, T., Kafi, A. A., Hossain, M. I. and Shah, F. M. (2016). Review spam detection using active learning, *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, IEEE, pp. 1–7.

Atkinson, E. J., Therneau, T. M., Melton III, L. J., Camp, J. J., Achenbach, S. J., Amin, S. and Khosla, S. (2012). Assessing fracture risk using gradient boosting machine (gbm) models, *Journal of Bone and Mineral Research* **27**(6): 1397–1404.

Banerjee, S., Chua, A. Y. and Kim, J.-J. (2015). Using supervised learning to classify authentic and fake online reviews, *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*, ACM, p. 88.

Barbado, R., Araque, O. and Iglesias, C. A. (2019). A framework for fake review detection in online consumer electronics retailers, *Information Processing & Management* **56**(4): 1234–1244.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization, *Journal of Machine Learning Research* **13**(Feb): 281–305.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research* **16**: 321357.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system, *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, ACM, pp. 785–794.

Chen, W., Fu, K., Zuo, J., Zheng, X., Huang, T. and Ren, W. (2017). Radar emitter classification for large data set based on weighted-xgboost, *IET Radar, Sonar & Navigation* **11**(8): 1203–1207.

Chowdhary, N. S. and Pandit, A. A. (2018). Fake review detection using classification, *International Journal of Computer Applications* **180**(50): 16–21.

Courtis, J. K. (1998). Annual report readability variability: tests of the obfuscation hypothesis, *Accounting, Auditing & Accountability Journal* **11**(4): 459–472.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996). From data mining to knowledge discovery in databases, *AI magazine* **17**(3): 37–37.

Filieri, R. (2016). What makes an online consumer review trustworthy?, *Annals of Tourism Research* **58**: 46–64.

Fontanarava, J., Pasi, G. and Viviani, M. (2017). Feature analysis for fake review detection through supervised classification, *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, pp. 658–666.

François, T. and Miltsakaki, E. (2012). Do nlp and machine learning improve traditional readability formulas?, *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*, Association for Computational Linguistics, pp. 49–57.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine, *Annals of statistics* pp. 1189–1232.

Gilda, S. (2017). Evaluating machine learning algorithms for fake news detection, *2017 IEEE 15th Student Conference on Research and Development (SCOReD)*, IEEE, pp. 110–115.

Gumus, M. and Kiran, M. S. (2017). Crude oil price forecasting using xgboost, *2017 International Conference on Computer Science and Engineering (UBMK)*, IEEE, pp. 1100–1103.

Haixiang, G., Yijing, L., Yanan, L., Xiao, L. and Jinling, L. (2016). Bpso-adaboost-knn ensemble learning algorithm for multi-class imbalanced data classification, *Engineering Applications of Artificial Intelligence* **49**: 176–193.

Hancock, J. T., Curry, L. E., Goorha, S. and Woodworth, M. (2007). On lying and being lied to: A linguistic analysis of deception in computer-mediated communication, *Discourse Processes* **45**(1): 1–23.

Jain, R. and Nayyar, A. (2018). Predicting employee attrition using xgboost machine learning approach, *2018 International Conference on System Modeling & Advancement in Research Trends (SMART)*, IEEE, pp. 113–120.

Jindal, N. and Liu, B. (2008). Opinion spam and analysis, *Proceedings of the 2008 international conference on web search and data mining*, ACM, pp. 219–230.

Jindal, N., Liu, B. and Lim, E.-P. (2010). Finding unusual review patterns using unexpected rules, *Proceedings of the 19th ACM international conference on Information and knowledge management*, ACM, pp. 1549–1552.

Kaur, G. and Kaur, E. P. (2017). Novel approach to text classification by svm-rbf kernel and linear svc, *International Journal of Advance Research, Ideas and Innovation in Technology* **3**(3).

Ko, M.-C., Huang, H.-H. and Chen, H.-H. (2017). Paid review and paid writer detection, *Proceedings of the International Conference on Web Intelligence*, ACM, pp. 637–645.

Lau, R. Y., Liao, S., Kwok, R. C.-W., Xu, K., Xia, Y. and Li, Y. (2011). Text mining and probabilistic language modeling for online review spam detection, *ACM Transactions on Management Information Systems (TMIS)* **2**(4): 25.

Lee, K., Ham, J., Yang, S.-B. and Koo, C. (2018). Can you identify fake or authentic reviews? an fsqca approach, *Information and Communication Technologies in Tourism 2018*, Springer, pp. 214–227.

Li, F. H., Huang, M., Yang, Y. and Zhu, X. (2011). Learning to identify review spam, *Twenty-second international joint conference on artificial intelligence*.

Li, H., Liu, B., Mukherjee, A. and Shao, J. (2014). Spotting fake reviews using positive-unlabeled learning, *Computación y Sistemas* **18**(3): 467–475.

Lim, E.-P., Nguyen, V.-A., Jindal, N., Liu, B. and Lauw, H. W. (2010). Detecting product review spammers using rating behaviors, *Proceedings of the 19th ACM international conference on Information and knowledge management*, ACM, pp. 939–948.

Lu, Y., Zhang, L., Xiao, Y. and Li, Y. (2013). Simultaneously detecting fake reviews and review spammers using factor graph model, *Proceedings of the 5th annual ACM web science conference*, ACM, pp. 225–233.

Martinez-Torres, M. and Toral, S. (2019). A machine learning approach for the identification of the deceptive reviews in the hospitality sector using unique attributes and sentiment orientation, *Tourism Management* **75**: 393–403.

Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M. and Ghosh, R. (2013). Spotting opinion spammers using behavioral footprints, *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 632–640.

Mukherjee, A., Liu, B. and Glance, N. (2012). Spotting fake reviewer groups in consumer reviews, *Proceedings of the 21st international conference on World Wide Web*, ACM, pp. 191–200.

Mukherjee, A., Venkataraman, V., Liu, B. and Glance, N. (2013). What yelp fake review filter might be doing?, *Seventh international AAAI conference on weblogs and social media*.

Nayak, D. R., Dash, R. and Majhi, B. (2016). Brain mr image classification using two-dimensional discrete wavelet transform and adaboost with random forests, *Neurocomputing* **177**: 188–197.

O'Mahony, M. P. and Smyth, B. (2010). Using readability tests to predict helpful product reviews, *Adaptivity, Personalization and Fusion of Heterogeneous Information*, Le Centre De Hautes Etudes Internationales D'Informatique Documentaire, pp. 164–167.

OConnor, P. (2008). User-generated content and travel: A case study on tripadvisor. com, *Information and communication technologies in tourism 2008* pp. 47–58.

Rastogi, A. and Mehrotra, M. (2017). Opinion spam detection in online reviews, *Journal of Information & Knowledge Management* **16**(04): 1750036.

Shu, K., Sliva, A., Wang, S., Tang, J. and Liu, H. (2017). Fake news detection on social media: A data mining perspective, *ACM SIGKDD Explorations Newsletter* **19**(1): 22–36.

Si, L. and Callan, J. (2001). A statistical model for scientific readability, *CIKM*, Vol. 1, pp. 574–576.

Thu, P. P. and New, N. (2017). Implementation of emotional features on satire detection, *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, IEEE, pp. 149–154.

Torlay, L., Perrone-Bertolotti, M., Thomas, E. and Baciu, M. (2017). Machine learning–xgboost analysis of language networks to classify patients with epilepsy, *Brain informatics* **4**(3): 159.

Touzani, S., Granderson, J. and Fernandes, S. (2018). Gradient boosting machine for modeling the energy consumption of commercial buildings, *Energy and Buildings* **158**: 1533–1543.

Yao, Y., Viswanath, B., Cryan, J., Zheng, H. and Zhao, B. Y. (2017). Automated crowdturfing attacks and defenses in online review systems, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ACM, pp. 1143–1158.

Yoo, K.-H. and Gretzel, U. (2009). Comparison of deceptive and truthful travel reviews, *Information and communication technologies in tourism 2009* pp. 37–47.

Zhang, L. and Zhan, C. (2017). Machine learning in rock facies classification: an application of xgboost, *International Geophysical Conference, Qingdao, China, 17-20 April 2017*, Society of Exploration Geophysicists and Chinese Petroleum Society, pp. 1371–1374.

Zheng, H., Yuan, J. and Chen, L. (2017). Short-term load forecasting using emd-lstm neural networks with a xgboost algorithm for feature importance evaluation, *Energies* **10**(8): 1168.

Zhou, L., Burgoon, J. K., Twitchell, D. P., Qin, T. and Nunamaker Jr, J. F. (2004). A comparison of classification methods for predicting deception in computer-mediated communication, *Journal of Management Information Systems* **20**(4): 139–166.