A Major Project
Report On

# FAKE REVIEW DETECTATION USING MACHINE LEARNING

***Project submitted in partial fulfillment of the requirements for the award of the degree of***

## BACHELOR OF TECHNOLOGY

**IN**

## COMPUTER SCIENCE AND ENGINEERING

**BY**


**U.BHARGAVI** **(18C91A05A3)**

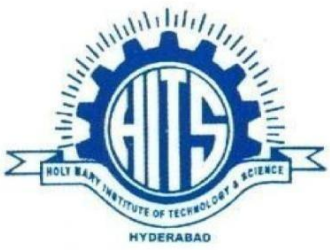
**Y.ABHISHEK** **(18C91A05B4)**


**Under the Esteemed guidance of**

**Mr. D. BHAGYARAJ M.Tech**

Assistant Professor

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**





## HOLY MARY INSTITUTE OF TECHNOLOGY & SCIENCE

**(COLLEGE OF ENGINEERING)**

***(Approved by AICTE New Delhi, Permanently Affiliated to JNTU Hyderabad, Accredited by NAAC with 'A' Grade)***

**Bogaram (V), Keesara (M), Medchal District -501 301.**


**2021 - 2022**

# HOLY MARY INSTITUTE OF TECHNOLOGY & SCIENCE

**(COLLEGE OF ENGINEERING)**

*(Approved by AICTE New Delhi, Permanently Affiliated to JNTU Hyderabad, Accredited by NAAC with 'A' Grade)*
**Bogaram (V), Keesara (M), Medchal Dist-501301.**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# CERTIFICATE

This is to certify that the major project entitled "FAKE REVIEW DETECTION USING MACHINE LEARNING" is being submitted by U.BHARGAVI (18C91A05A3), Y.ABHISHEK (18C91A05B4), in Partial fulfillment of the academic requirements for the award of the degree of Bachelor of Technology in "COMPUTER SCIENCE AND ENGINEERING" from HOLY MARY INSTITUTE OF TECHNOLOGY & SCIENCE, JNTU Hyderabad during the year 2021- 2022.

**INTERNAL GUIDE**

Mr.D.BHAGYARAJ M.Tech
Assistant Professor
Dept. of Computer Science & Engineering

**HEAD OF THE DEPARTMENT**

Dr .B.NARSIMHA
Professor & HOD
Dept. of Computer Science & Engineering

**EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, who's constant guidance and encouragement crowns all effort with success.

We take this opportunity to express my profound gratitude and deep regards to our Guide **Mr.D.BHAGYARAJ, Assistant Professor**, Dept. of Computer Science & Engineering, Holy Mary Institute of Technology & Science for his / her exemplary guidance, monitoring and constant encouragement throughout the project work.

Our special thanks to **Dr. B. Narsimha, Head of the Department**, Dept. of Computer Science & Engineering, Holy Mary Institute of Technology & Science who has given immense support throughout the course of the project.

We also thank Dr. **P. Bhaskar Reddy,** the **honorable Director** of my college Holy Mary Institute of Technology & Science for providing me the opportunity to carry out this work.

At the outset, we express my deep sense of gratitude to the beloved **Chairman A. Siddarth Reddy** of **Holy Mary Institute of Technology & Science**, for giving me the opportunity to complete my course of work

We are obliged to **staff members** of Holy Mary Institute of Technology & Science for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of my assignment.

Last but not the least we thank our **Parents**, and **Friends** for their constant encouragement without which this assignment would not be possible.

**U.BHARGAVI (18C91A05A3)**

**Y.ABHISHEK (18C91A05B4)**

# DECLARATION

This is to certify that the work reported in the present project titled **"FAKE REVIEW DETECTION USING MACHINE LEARNING"** is a record of work done by us in the Department of Computer Science & Engineering, Holy Mary Institute of Technology and Science.

To the best of our knowledge no part of the thesis is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred to in the text. The reports are based on the project work done entirely by us not copied from any other source.

**U.BHARGAVI (18C91A05A3)**

**Y.ABHISHEK (18C91A05B4)**

# CONTENTS

# LIST OF FIGURES

# LIST OF IMAGES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| LMS | Library Management System |
| DBMS | Database management system |
| UML | Unified Modeling Language |
| GCP | Google Cloud Platform |
| AWS | Amazon Web Services |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| PHP | Hypertext Preprocessor |
| SQL | Structured Query Language |
| ISO | International Organization for Standardization |
| EC2 | Amazon’s Elastic Compute Cloud |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |

# ABSTRACT

Users communicate over all social media, but messages are not secured when it passes through network. Intruder can access user's message easily. We want to secure users' communication over all social media. So here we proposed a system where user will enter the plain text and select the algorithm type from AES, DES, MD5, and provide the key, a chipper text will be formed that
can be sent via any communication application and end user can decrypt the text by selecting the same algorithm type and must enter the same sender secret key. User can use our application and can enter the plain text and must select the algorithm type and must enter the secret key to encrypt the message and receiver can decrypt the message by specifying the same algorithm used for encryption and must use the same secret key used by sender. Intruder will find difficult to decrypt the message. By using this method, you can double ensure that your secret message is sent securely without outside interference of hackers or crackers. If sender sends this image in public others will not know, what is it, and it will be received by receiver. You will need the key and algorithm type to decrypt the hidden text.

# 1. INTRODUCTION

## 1.1 Overview

Nowadays e-commerce sites have become very popular because a lot of products and services and their reviews are easily available online. Online reviews have become a good way for users for their decision making while making any purchase from these sites. Today because of the popularityof e-commerce sites, spammershavemadetheir target to these sites for review spam apart from other spam like email spam or web spam. Review spam means basically fake review that is written byfraudsters. Mostly e-commerce sites give section for review in order that users can write their opinion about products. There are also many review sites available like *TripAdvisor.com* which allows customer to write review for different hotels, *Zomato.com* which allows to write review about different restaurant, *Amazon.com* which allow users to write their opinion about their products and services, *Flipkart.com*, *Yelp.com* etc.

Such type of content provided by web is named as user-generated content. User-generated content contains a lot of valuable and important information about the products and services. Since there is no control on the quality of this content on the web and hence, these promote fraudsters to write fake and wrong information about the products. These fake and wrong information written by fraudsters is called as review spam. Fake reviews prevent customers and organizations reaching actual conclusions about the products. Hence, it highly affects the e-commerce business.

Hence, over the last few years, these review sites have been removing fake reviews about from their websites using their own spam detection technique.

Machine learning techniques have been more popular for spam detection. They use supervised (required all data set labelled), semi-supervised (require very few data set labelled) and unsupervised (works for unlabeled data set) learning technique.

Generally, fake reviews are written for two purposes one for promoting some target objects (positive fake review or positive spam) and another for damage the reputation of other targets (negative fake review or negative spam).

**What is Review?**

A review is a feedback or evaluation of a service, a company or a product such as a movie (a movie review), a book (a book review), a mobile phone (a mobile phone review), a hotel (a hotel review), a restaurant (a restaurant review) etc. There are many review sites available (like TripAdvisor, Tomato, Yelp etc.) which allow users to write their opinion about the products and services. Anyone who writes review is called as reviewer.

**Review 1:** We stay at Hilton for 4 nights last march. It was a pleasant stay. We got a large room with 2 double beds and 2 bathrooms, The TV was Ok, a 27' CRT Flat Screen. The concierges were very friendly when we need. The room was very cleaned when we arrived, we ordered some pizzas from room service and the pizza was Ok also. The main Hall is beautiful. The breakfast is charged, 20 dollars, kind of expensive. The internet access (Wi-Fi) is charged, 13 dollars/day. Pros: Low rate price, huge rooms, close to attractions at Loop, close to metro station. Cons: Expensive breakfast, Internet access charged. Tip: When leaving the building, always use the Michigan Av exit. It's a great view.

**Review 2:** My husband and I stayed for two nights at the Hilton Chicago, and enjoyed every minute of it! The bedrooms are immaculate, and the linens are very soft. We also appreciated the free wife, as we could stay in touch with friends while staying in Chicago. The bathroom was quite spacious, and I loved the smell of the shampoo they provided - not like most hotel shampoos. Their service was amazing, and we absolutely loved the beautiful indoor pool. I would recommend staying here to anyone.

There is no clear indication from above two reviews that which review is fake and which are actual. But Review 1 is actual however Review 2 is fake. This can be only identified by data mining and machine learning technique.

There are no clear indications or signals from the text of the two reviews that indicate to the casual reader that the first review is real while the second is a fake. Nevertheless, guides provided by the Consumerist2 and MoneyTalksNews3 websites offer tips to help consumers spot fake reviews. A computer scientist might seek to utilize this logic when training data mining and machine learning algorithms to find these features in the review that will determine if it is real or fake. Over 18 million reviews were created on Yelp 2014 and Trip Advisor currently has over 200 million reviews. Online reviews are constantly being generated on various web sites across the Internet. Consequently, Big Data techniques are needed to address the problem of review spam. Big Data, while an overused buzzword with an elusive definition, is often quantified with the Four V's : (1) Volume – the sheer size and scale of the data,

(2) Velocity – the rate at which new data is created and consumed by processing engines,

(3) Variety – the different formats that data may be stored in, and

(4) Veracity – the quality level of the data. The Volume and Velocity of online reviews are noted by merely visiting e-commerce and customer rating sites, such as Yelp and Amazon. There is great Variety across the possible industry sectors for reviews (such as hotels, restaurants, e-commerce, home services, etc.), along with the multiplicity of languages that reviews are written in. Veracity is a problem with online reviews, since the vast majority of reviews are unlabeled, which means it is not easily known whether the review is fake or not. Additionally, standard machine learning algorithms tend to break down and become ineffective when dealing with data of this size, which poses a problem when trying to apply these algorithms for review spam detection. Thus, review spam detection is a Big Data problem, as there are numerous challenges when analyzing and classifying varying reviews from disconnected sources.

Data mining and machine learning techniques, primarily those for web and text mining, offer an exciting contribution to detecting fraudulent reviews. According to Liu, web mining is "the process for finding useful information and relations from the contents available on the web by largely relying on the available machine learning techniques and methods". Web mining can be divided into three types of tasks: structure, content and usage mining. Content mining is concerned with knowledge and information extraction, and categorizing entities using data mining and machine learning approaches. A straightforward example of content mining is opinion mining. Opinion mining consists of attempting to ascertain the sentiment (i.e., positive or negative polarity) of a text passage by analyzing the features of that passage. A classifier can be trained to classify new instances by analyzing the text features associated with different opinions along with their sentiment. Constructing features to describe the text of the review involves text mining and Natural Language Processing (NLP). Additionally, there may be features associated with the review's writer, its post date/time and how the review deviates from other reviews for the same product or service. It is important to mention that while most existing machine learning techniques are not sufficiently effective for review spam detection, they have been found to be more reliable than manual detection. A common approach in text mining is to use a bag of words approach where the presence of individual words, or small groups of words are used as features; however, several studies have found that this approach is not sufficient to train a classifier with adequate performance in review spam detection.

Combining review spam detection through a review's features, and spammer detection through analysis of their behavior may be a more effective approach for detecting review spam than either approach alone.

Before addressing the challenges associated with improving review spam detection, we must first address collection of data. Data is a major part of any machine learning based model, and while a massive volume of reviews are available on the Internet, collecting and labeling a sufficient number of them to train a review spam classifier is a difficult task. An alternative to collecting and labeling data is to artificially create review spam datasets by using synthetic review spamming, which takes existing truthful reviews and builds fake reviews from them. The Feature Engineering for Review Spam Detection section provides an overview of feature engineering in this domain, both for review centric spam detection and reviewer centric spam detection. The Review Centric Review Spam Detection section discusses and analyzes current research using supervised, unsupervised and semi-supervised machine learning for review centric spam detection. The Reviewer Centric Review Spam Detection section provides an overview of studies using reviewer centric features.

**Types of Reviews**

**1.Positive Reviews**

If reviewers write positive things about the product or services, such review is called as Positive Reviews.

e.g. *The hotel is very nice. Room and services are too good. That is the awesome place to stay whole day and night. Rent is also affordable.*

**2. Negative Reviews**

If reviewers write negative things about the product or services, such review is called as Positive Reviews.

e.g. *Do not buy Samsung Galaxy S6. It is the worst mobile among all that I have used. No battery backup. Very bad camera quality. Touch pad is very hard.*

**Types of Spams**

**1.Email Spam**

If the sender sends unwantedand unsolicited email either directlyor indirectlyto user and there is no relationship of this email to the user is called as email spam. It is also called as junk email or unsolicited email. Email spam comes under the category of electronic spam. Example of such type of spam is phishing email.

**2. Web Spam**

Web spam (also called as search spam) refers to the action of the deceptive search engine so that the rank of a specific website becomes more than it deserves.

**3. SMS Spam**

If someone transmits unwanted and unsolicited messages over communication media (i.e. cell phone) is called as SMS spam. It comes under the category of electronic spam.

**4. Comment Spam**

Comment spams are generally written by spammers by posting their fake comments about the products and services.

**What is Review Spam?**

Today because of the popularity of e-commerce sites, spammers have made their target to these sites for review spam. Mostly e-commerce sites give section for review in order that users can write their opinion about products. There are also many review sites availablewhich allowusers to write their opinion about the products and services. Such type of content provided by web is named as user-generated content. User-generated content contains a lot of valuable and important information about the products and services. Since there is no control on the quality of this content on the web and hence, these promote fraudsters to write fake reviews.

**Types of Review Spam**

Review spams are generally categorizing in three categories:

**Type 1 (Untruthful opinions):**

It is also divided into two sub-categories:

**i. Hyper spam:** Fraudsters write positive fake opinions to promote some targets.

**ii. Defaming spam**: Fraudsters write negative fake opinions to damage the reputation of some targets.

**Type 2 (Reviews on brand only):**

Such type of review only focuses on brand name. Fraudsters write only about the brand, i.e the manufacturers of the products rather than the products.

**Type 3 (Non-reviews):**

Fraudsters write something that is totally unrelated to the products i.e. junk, such type of review spam comes under non-reviews. They have two forms:

i. advertisements, and

ii. irrelevant opinion.

**Types of Spammers**

A spammer is a person or a machine who writes spam (spam may be either email spam, web spam, review spam etc.). While finding fake review (spam) we can find two types ofspammers. These are:

**Individual Spammer:**

A single reviewer who uses different user-ids to register several times at a site for writing fake review.

They write either only positive reviews about a product for promotion or only negative reviews for damage the reputation of competitor's product.

They give too high rating for the prod

**A group of spammers:**

A group of reviewers who divide group in sub-group and each of these sub divisions work on different sites for writing fake reviews.

Every spam member gives lower rating to the product.

The spammers write spam during launch time so that they can take the control over the sale of the product.

### 1.2 Challenges in Review Spam Detection

- The fake reviews look like genuine reviews with a lot of similar keywords.
- Reviews are very subjective in nature and therefore can vary from a small paragraph to a long description. There are a number of review sites are available which provide space for writing reviews to reviewers, so it is very difficult to find out that reviewer has actual used the product and wrote the actual review or fake review.
- Both witty and sarcasm reviews present on a common place and hence, it is a very tough task to analyze such reviews.
- There is no labelled data set available online to train spam model. Even when people were asked to label reviews as spam, the concurrence rate was around 60%.

### 1.3 Problem Statement

Our main aim is to develop a model to detect review spams from review websites using review text. We have used the most apt data sets in the area of review spam detection research work. Both supervised and semi-supervised learning technique have been applied to obtain spam (review) from the data set.

For supervised learning, we try to obtain some feature sets from different automated approaches that can best distinguish the spam and non-spam reviews.

Along with these features, sentiment analysis and data mining technique have also been used. For semi-supervised learning, PU-learning algorithm along with different classifier are used to detect review spam from the data set. Finally, a comparison of proposed technique with some existing review spam detection techniques has been done.

## 1.4 Motivation and Objective

### Motivation

From the last few years, e-commerce sites have become very popular because a lot of products and services and their reviews are easily available online. Today because of the popularity of e-commerce sites, spammers have made their target to these sites for review spam. Mostly e-commerce sites give section for review in order that users can write their opinion about products. There are also many review sites available like *TripAdvisor.com, Zomato.com, Amazon.com, Yelp.com* which allow users to write their opinion about their products and services. Such type of content provided by web is named as user-generated content.

User-generated content contains a lot of valuable and important information about the products and services. Since there is no control on the quality of this content on the web and hence, these promote fraudsters to write fake and wrong information about the products. Fake reviews prevent customers and organizations reaching actual conclusions about the products. Hence, it highly affects the e-commerce business.

Hence, over the last few years, these review sites have been removing fake reviews from their websitesusingtheirownspamdetection technique. Machinelearningtechniqueshavebeenmore popular for spam detection and hence, maintenance team of these websites use supervised (required all data set labeled), semi-supervised (require very few data set labeled) and unsupervised (works for unlabeled data set) learning technique.

### Objectives

Ourmainobjectives arefollowing:

- To develop a model to detect review spams from review websites using review text.
- To obtain some feature sets from different automated approaches that can best
- distinguish the spam and non-spam reviews.
- To detect spam (review) from both labeled and partially labeled data set.
- Apply the concept of machine learning (supervised and semi-supervised learning), opinion mining, data mining and sentiment analysis.

# 2.LITERATURE REVIEW

**Authors:** A. Heydari, Mhd. A. Tavakoli, N. Salim, and Z. Heydari.

**Paper:** Detection of review spam: A survey, Expert Systems with Applications 42,no. 7 (2015): 3634-3642

- In recent years, online reviews have become the most important resource of customers' opinions. These reviews are used increasingly by individuals and organizations to make purchase and business decisions.
- Unfortunately, driven by the desire for profit or publicity, fraudsters have produced deceptive (spam) reviews. The fraudsters' activities mislead potential customers and organizations reshaping their businesses and prevent opinion-mining techniques from reaching accurate conclusions.
- The present research focuses on systematically analyzing and categorizing models that detect review spam. Next, the study proceeds to assess them in terms of accuracy and results.
- We find that studies can be categorized into three groups that focus on methods to detect spam reviews, individual spammers and group spam. Different detection techniques have different strengths and weaknesses and thus favor different detection contexts.

**Authors:** N. Jindal and B. Liu.

**Paper:** Analyzing and Detecting Review Spam. ICDM2007.

- Mining of opinions from product reviews, forum posts and blogs is an important research topic with many applications. However, existing research has been focused on extraction, classification and summarization of opinions from these sources.
- An important issue that has not been studied so far is the opinion spam or the trustworthiness of online opinions. In this paper, we study this issue in the context of product reviews.
- To our knowledge, there is still no published study on this topic, although Web page spam and email spam have been investigated extensively. We will see that review spam is quite different from Web page spam and email spam, and thus requires different detection techniques.

- Based on the analysis of 5.8 million reviews and 2.14 million reviewers from amazon.com, we show that review spam is widespread. In this paper, we first present a categorization of spam reviews and then propose several techniques to detect them.

**Authors:** N. Jindal and B. Liu. 2008.

**Paper:** "Opinion spam and analysis". In Proceedings of the international conference on Web search and web data mining, pages 219–230. ACM

- Evaluative texts on the Web have become a valuable source of opinions on products, services, events, individuals, etc. Recently, many researchers have studied such opinion sources as product reviews, forum posts, and blogs.
- However, existing research has been focused on classification and summarization of opinions using natural language processing and data mining techniques.
- An important issue that has been neglected so far is opinion spam or trustworthiness of online opinions. In this paper, we study this issue in the context of product reviews, which are opinion rich and are widely used by consumers and product manufacturers.
- In the past two years, several startup companies also appeared which aggregateopinions from product reviews. It is thus high time to study spam in reviews.
- To the best of our knowledge, there is still no published study on this topic, although Web spam and email spam have been investigated extensively.
- We will see that opinion spam is quite different from Web spam and email spam, and thus requires different detection techniques.
- Based on the analysis of 5.8 million reviews and 2.14 million reviewers from amazon.com, we show that opinion spam in reviews is widespread. This paper analyzes such spam activities and presents some novel techniques to detect them.

**Authors:** B. Liu.

**Paper:** "Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies", pages 1–167, 2012.

- Sentiment analysis and opinion mining is the field of study that analyzes people's opinions, sentiments, evaluations, attitudes, and emotions from written language.
- It is one of the most active research areas in natural language processing and is also widely studied in data mining, Web mining, and text mining.
- In fact, this research has spread outside of computer science to the management sciences and social sciences due to its importance to business and society as a whole.

- The growing importance of sentiment analysis coincides with the growth of social media such as reviews, forum discussions, blogs, micro-blogs, Twitter, and social networks.
- For the first time in human history, we now have a huge volume of opinionated data recorded in digital form for analysis.
- Sentiment analysis systems are being applied in almost every business and social domain because opinions are central to almost all human activities and are key influencers of our behaviors.
- Our beliefs and perceptions of reality, and the choices we make, are largely conditioned on how others see and evaluate the world.
- For this reason, when we need to make a decision we often seek out the opinions of others. This is true not only for individuals but also for organizations.

**Authors:** M. Crawford, T.M. Khoshgoftaar, J.D. Prusa, A.N. Richter, H. Al Najada,

**Paper:** "Survey of review spam detection using machine learning techniques", Journal Of Big Data, 2, pp. 1-24, 2015.

- Online reviews are often the primary factor in a customer's decision to purchase a product or service, and are a valuable source of information that can be used to determine public opinion on these products or services.
- Because of their impact, manufacturers and retailers are highly concerned with customer feedback and reviews. Reliance on online reviews gives rise to the potential concern that wrongdoers may create false reviews to artificially promote or devalue products and services.
- This practice is known as Opinion (Review) Spam, where spammers manipulate and poison reviews (i.e., making fake, untruthful, or deceptive reviews) for profit or gain. Since not all online reviews are truthful and trustworthy, it is important to develop techniques for detecting review spam.
- By extracting meaningful features from the text using Natural Language Processing (NLP), it is possible to conduct review spam detection using various machine learning techniques.
- Additionally, reviewer information, apart from the text itself, can be used to aid in this process. In this paper, we survey the prominent machine learning techniques that have been proposed to solve the problem of review spam detection and the performance of different approaches for classification and detection of review spam.

- The majority of current research has focused on supervised learning methods, which require labeled data, a scarcity when it comes to online review spam. Research on methods for Big Data are of interest, since there are millions of online reviews, with many more being generated daily.
- To date, we have not found any papers that study the effects of Big Data analytics for review spam detection. The primary goal of this paper is to provide a strong and comprehensive comparative study of current research on detecting review spam using various machine learning techniques and to devise methodology for conducting further investigation

**Authors:** Mukherjee A, Venkataraman V, Liu B, Glance NS (2013)

**Paper:** What yelp fake review filter might be doing? Boston, In ICWSM.

- Online reviews have become a valuable resource for decision making. However, its usefulness brings forth a curse – deceptive opinion spam. In recent years, fake review detection has attracted significant attention.
- However, most review sites still do not publicly filter fake reviews. Yelp is anexception which has been filtering reviews over the past few years. However, Yelp's algorithm is trade secret.
- In this work, we attempt to find out what Yelp might be doing by analyzing its filtered reviews. The results will be useful to other review hosting sites in their filtering effort.
- There are two main approaches to filtering: supervised and unsupervised learning. In terms of features used, there are also roughly two types: linguistic features and behavioral features.
- In this work, we will take a supervised approach as we can make use of Yelp's filtered reviews for training. Existing approaches based on supervised learning are all based on pseudo fake reviews rather than fake reviews filtered by a commercial Web site.
- Recently, supervised learning using linguistic n-gram features has been shown to perform extremely well (attaining around 90% accuracy) in detecting crowdsourced fake reviews generated using Amazon Mechanical Turk (AMT).
- We put these existing research methods to the test and evaluate performance on the real-life Yelp data. To our surprise, the behavioral features perform very well, but the linguistic features are not as effective.

### 2.1.1 EXISTING SYSTEM

- It is important to mention that while most existing machine learning techniques are not sufficiently effective for review spam detection, they have been found to be more reliable than manual detection.
- The primary issue is the lack of any distinguishing words (features) that can give a definitive clue for classification of reviews as real or fake
- A common approach in text mining is to use a bag of words approach where the presence of individual words, or small groups of words are used as features; however, several studies have found that this approach is not sufficient to train a classifier with adequate performance in review spam detection

### 2.1.2 PROPOSED SYSTEM

- In this paper we discuss machine learning techniques that have been proposed for the detection of online review spam, with an emphasis on feature engineering and the impact of those features on the performance of the spam detectors.
- Additionally, the merits of supervised, unsupervised and semi-supervised learning methods are analyzed and results of current research using each approach presented along with a comparative analysis.
- Finally, we provide suggestions for aspects of review spam detection requiring further investigation, and best practices for conducting future research.
- To the best of our knowledge, this paper includes information about all of the datasets that have been used, or generated for use, in the reviewed literature.

**Bag of words** approach considers words or sequence of words used in reviews as features. Sequences of words are called n-grams (where n denotes number of words in a sequence). Values of n=1, 2, 3 are the most common. Term frequency includes n-grams as well as the number of their occurrences. This additional information can improve bag of words approach.**Semantic features** are focused on the meaning of words. They include synonyms and similarphrases. The idea of using these features is that spammers usually replace some words with similar ones, conveying the message, while making it harder to identify duplicate reviews.

# 3. SOFTWARE REQUIREMENT AND SPECIFICATION

## 3.1 SOFTWARE REQUIREMENTS

- PYTHON
- LABELLED SAMPLE DATA
- FEATURE ENGINEERING MODEL
- MY SQL
- PYCHARM IDE.

## 3.2HARDWARE REQUIREMENTS

- PROCESSOR : INTEL I5 OR ABOVE, RYZEN 3 OR ABOVE, ANY MACBOOK
- SPEED : >=2GHZ
- RAM : 8GB
- HARDDISK : 256GB

# 4.SYSTEM DESIGN

In System Design has divided into three types like GUI Designing, UML Designing with avails in development of project in facile way with different actor and its utilizer case by utilizer case diagram, flow of the project utilizing sequence, Class diagram gives information aboutdifferent class in the project with methods that have to be utilized in the project if comes to ourproject our UML Will utilizable in this way The third and post import for the project in systemdesign is Data base design where we endeavor to design data base predicated on the number of modules in our project.

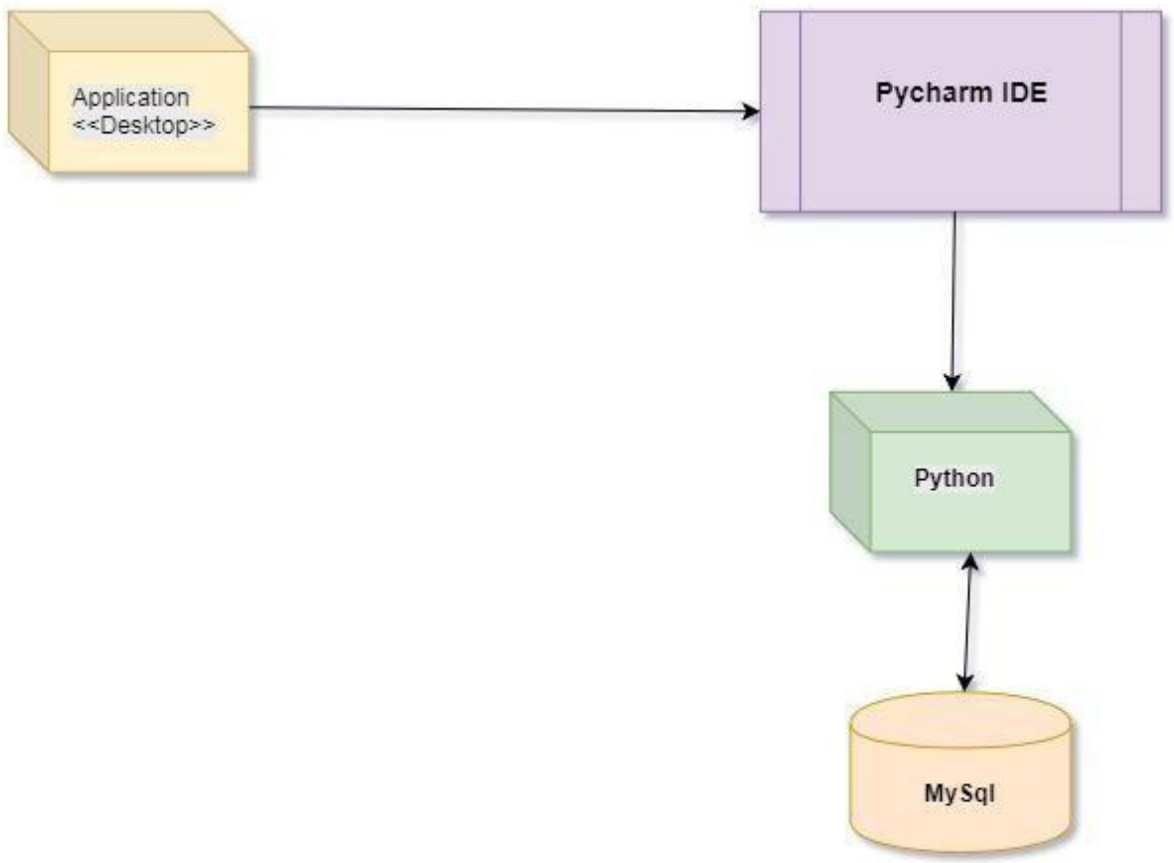
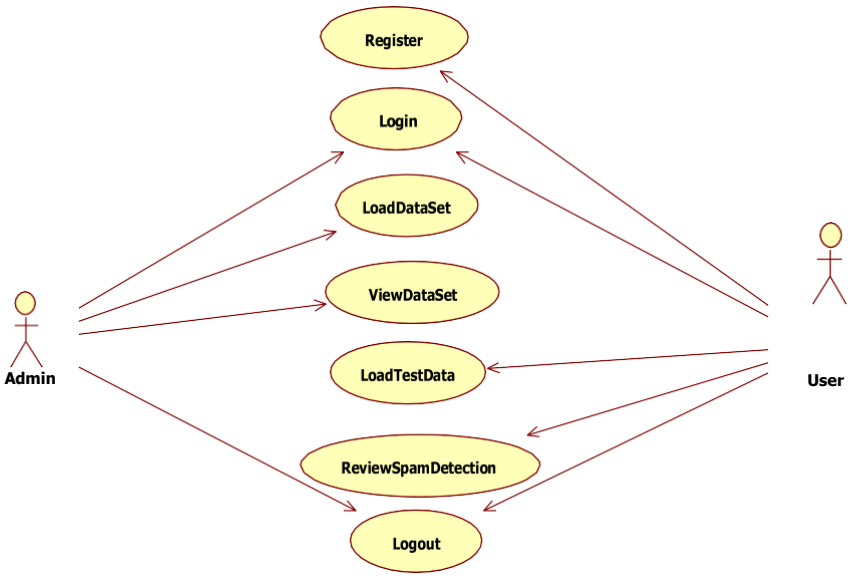



Fig 4.1. Application Block Diagram

## 4.1 UML DIAGRAMS:

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

## 4.1 Use case diagrams:




4.2. Use Case Diagram

## 4.2 Sequence diagrams:

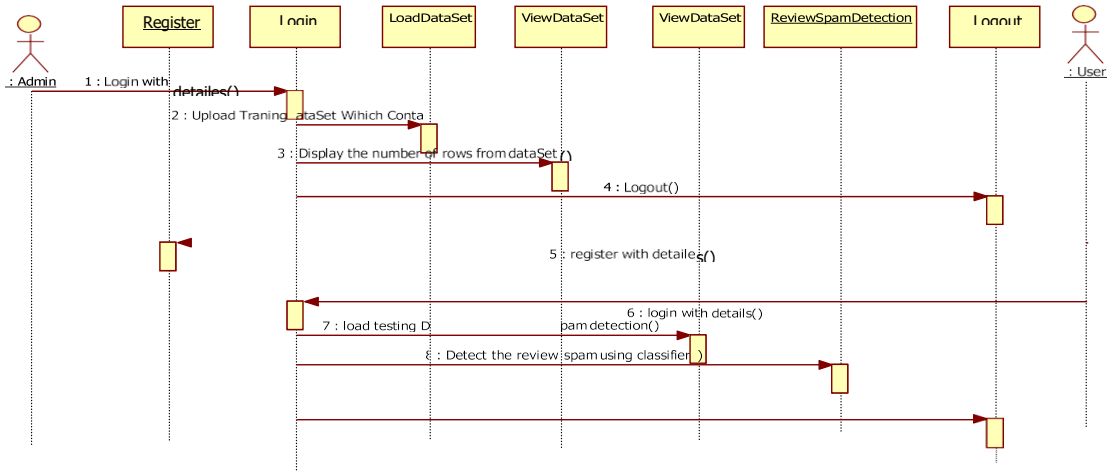



Fig 4.3. Sequence diagram

## 4.3 Collaboration diagram:

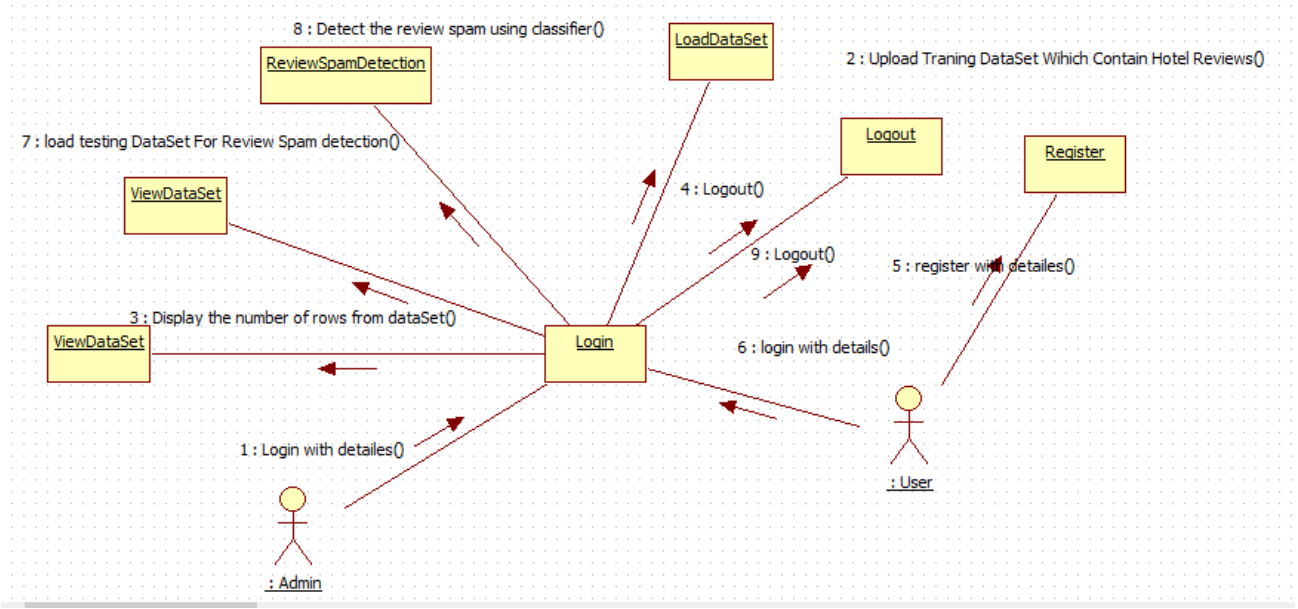



Fig 4.4 Collaboration diagram

## 4.4 State chat diagram:

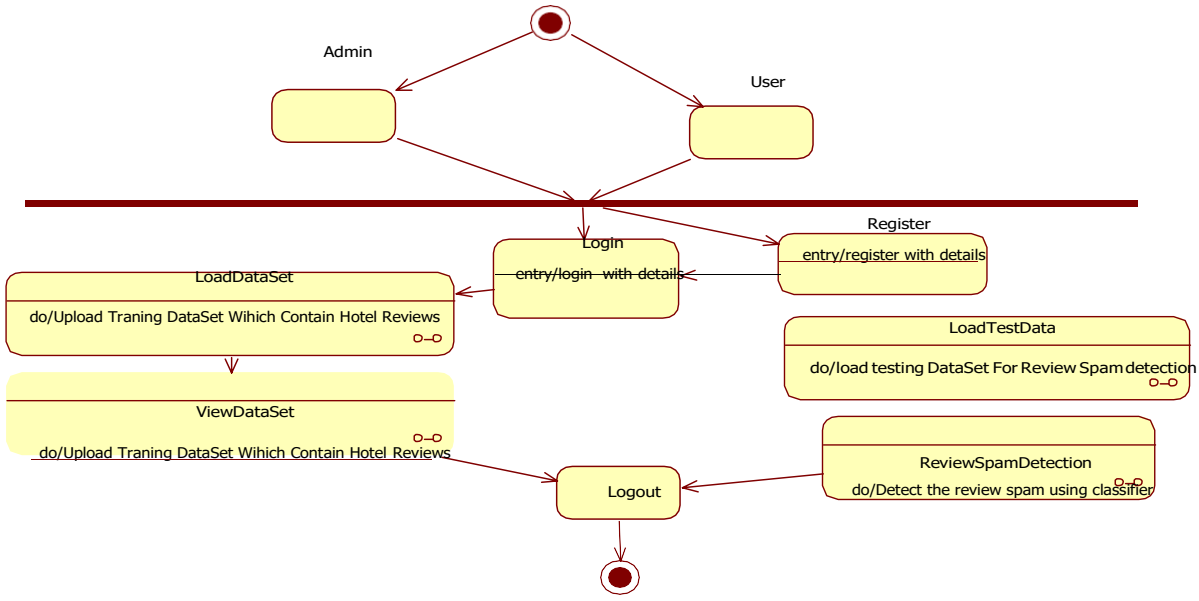



Fig 4.5 State chat diagram

## 4.5 Activity diagram:

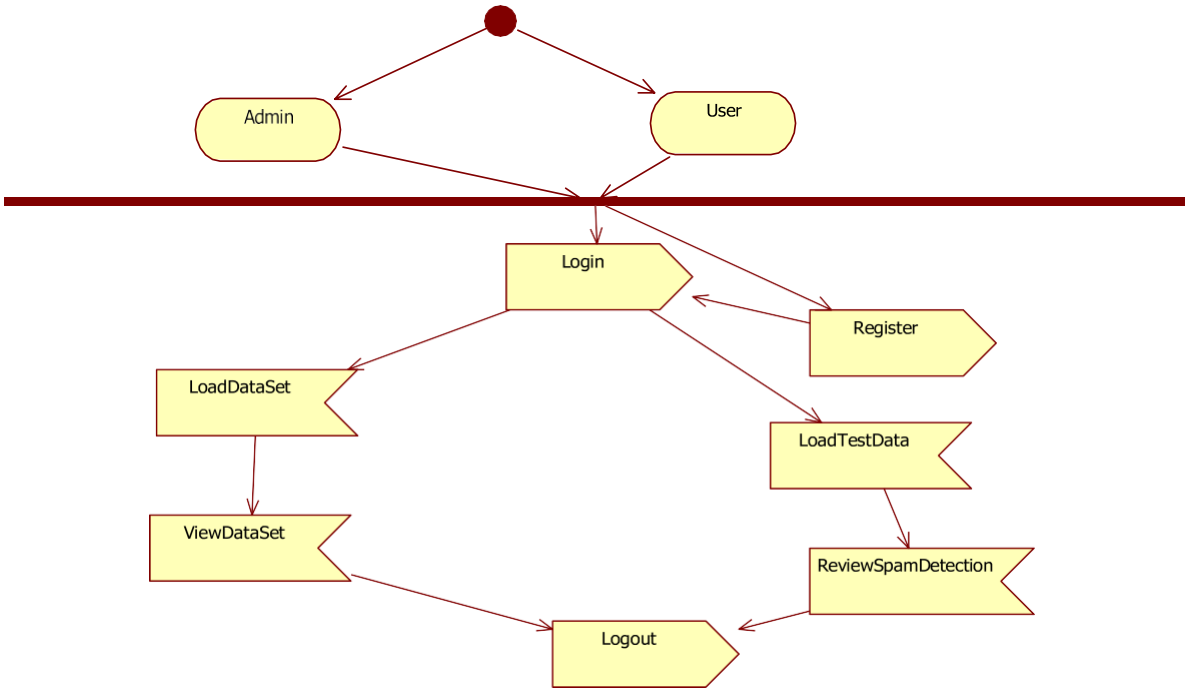
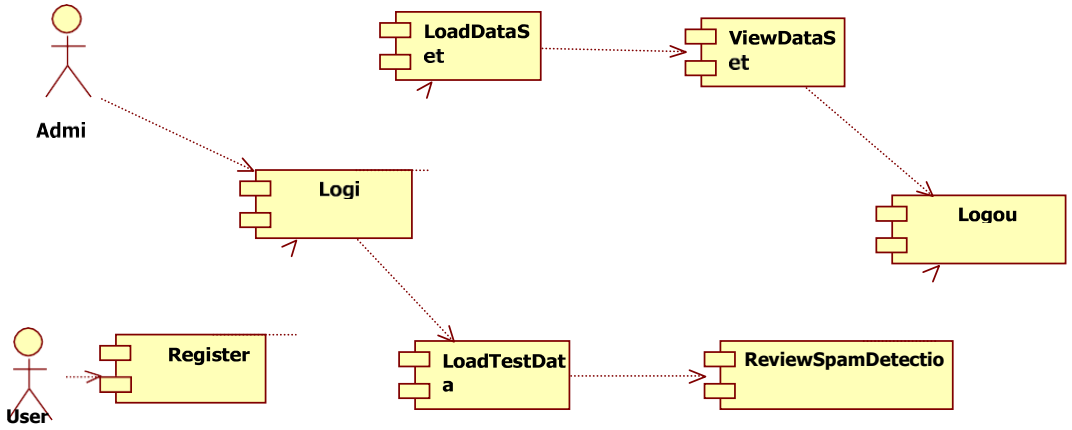



Fig 4.6 Activity diagram

## 4.6 Component diagram:



# 5. SPAM DECTECTION TECHNIQUE

Basically three machine learning techniques are used to detect spam. These are:

## 5.1 Supervised Learning Technique:

In supervised learning technique, we need labelled reviews or data set. We extracted a set of features from these data set. These features are generally LIWC, POS tagging, N-gram and sentiment score. After thesestepsdifferentclassifierslike SVM, decisiontree, logisticregression, Naive Bayes etc., are trained and accuracy is calculated. This is verysimple form among all spam detection techniques.

Supervised learning can be used to detect review spam by looking at it as the classification problem of separating reviews into two classes: spam and non-spam reviews. To the best of our knowledge, the first researchers to have studied deceptive opinion spam using supervised learning were Jindal et al. [1]. They discuss the evolution of opinion mining, which had primarily focused on extracting or summarizing the opinions from text by using Natural Language Processing (NLP). Prior to their contribution, the content characteristics of the text that might indicate abnormal activities, such as creating review spam, had not been addressed. In an effort to investigate opinion spam in reviews and devise techniques for review spam detection, Jindal et al. collected

| Method | Attributes |
|---|---|
| Supervised Learning | Learning from a set of labeled data |
| | Requires labeled training data |
| | Most common form of learning |
| Unsupervised Learning | Learning from a set of unlabeled data |
| | Finds unseen relationships in the data independent of class label |
| | Most common form is clustering |
| Semi-supervised Learning | Learning from labeled and unlabeled data |
| | Only requires a relatively small set of labeled data which is supplemented with a large amount of unlabeled data |
| | Ideal for cases such as review spam where vast amounts of unlabeled data exist |

Figure 5.1: Types of Machine Learning techniques.

Review spam can be found in multiple languages, as reviewers from all around the world can write online reviews in any language they want. While many of the features will remain unchanged (i.e., spammers characteristics and behaviors), word features will change to reflect each language.

## 5.1. Semi-supervised Learning Technique:

Semi-Supervised learning technique is same as supervised learning technique with slightly difference is that we do not need to label all the data set. If we have a very few labelled data set, then we can use such learning technique. Very few works have been done in this area.

In other domains, it has been found that using unlabeled data in conjunction with a small amount of labeled data can considerably improve learner accuracy compared to completely supervised methods. In a study, a two-view semi supervised method for review spam detection was created by employing the framework of a co-training algorithm to make use of the large amount of unlabeled reviews available.

The co-training algorithm is a bootstrapping method that uses a set of labeled data to incrementally apply labels to unlabeled data. It trains 2 classifiers on 2 distinct sets of features and adds the instances most confidently labeled by each classifier to the training set. This effectively allows large datasets to be generated and used for classification, reducing the demand to manually produce labeled training instances. A modified version of the co-training algorithm that only adds instances that were assigned the same label by both classifiers was also proposed. Their dataset was generated with the assistance of students who manually labeled 6000 reviews collected from Epinions.com, 1394 of which were labeled as review spam. Four groups of review centric features were created: content, sentiment, product and metadata. Another two groups of reviewer centric features
were created: profile and behavioral.

In order to use the two-view method for adding unlabeled instances to the training set, classifiers were trained on each set of features (i.e., one with review centric features and another with reviewer centric ones). Note that these 2 classifiers are only used to add instances to the labeled data and the final classifier is trained using all available features, both review centric and reviewer centric. Experiments were conducted using Naïve Bayes, Logistic Regression and SVM with 10-fold cross validation, and it was found that Naïve Bayes was the best performer, so all additional work was performed with Naïve Bayes.

PU-Learning is a second type of semi-supervised learning approach, developed by Liu et al. [22], to learn from a few positive examples and a set of unlabeled data.Montes-y-Gómez and

Rosso adapt this approach for review spam detection in their work “Using PU-Learning “

PU-learning is an iterative method which tries to identify a set of reliably negative instances in the unlabeled data. The model is trained and evaluated using all of the unlabeled data as the negative class and any instances that are classified as positive are removed.

### 5.2. Unsupervised Learning Technique:

If we haveunlabeleddataset, thenwego for unsupervised learningtechnique where we find some hidden pattern. It includes k-mean clustering and mixture models etc. Because of the difficulty of producing accurately labeled datasets of review spam, the use of supervised learning is not always applicable. Unsupervised learning provides a solution for this, as it doesn’t require labeled data.

### 5.3 Feature Sets from Different Automated Approaches Linguistic Inquiry Word Count

The Linguistic Inquiry and Word Count (LIWC) is a text analyzing tool which analyzes 80 different types of features like linguistic dimension (i.e. words count, words per sentence etc.), psychological processes (i.e. positive emotion, negative emotion, perceptual processes, biological processes etc.), personal concerns (i.e. home, money, religion, death etc.) and spoken categories (i.e. assent, non-fluencies, fillers etc).

**POS Tags**

Work in linguistics has already proved that the distribution of frequency of parts of speech (POS) tagging of any text is directly dependent on the genre of that text [Biber et al., 1999; Rayson et al., 2001]. Hence, according to this approach, feature made for every review is primarilybased onthe frequencyof everyPOS tag fortesting relationship this feature and actual and fake reviews.

**N-gram Feature**

In n-gram feature, we select n contiguous words from a text as a feature. If one word at a time is being considered as a feature then, it is called as unigram; if two contiguous words at a time is being selected then, it is bigram and similarly if we select three contiguous words at a time as a features then, it is called as trigram. These features help us to model all the content and its context. In this work, only unigram as a feature has been used.

**Sentiment Score**

The negative spammers generally used to write more negative words in their review like horrible, disappointed etc. and hence, show more negative sentiment than a truthful negative

Similarly, positive spammers used to write more positive words like beautiful, great etc. and show more positive sentiment than an actual positive review.

**Classification Techniques**

Features from above approaches are used to train 6 classifiers i.e. Decision Tree, Naive Bayes, Support Vector Machine (SVM), k-NN, Random Forest and Logistic Regression.

**Decision Tree**

Decision tree is one of the simplest classification algorithm used in machine learning technique. It is based on tree structure in which internal nodes represent test sets and leavesrepresentclass label(decisionthatistakenafter calculatingall attributes). Each branch represents output of test. A decision tree contains three types of node

i.e. root, branch and leaf node. These are basic steps of decision tree algorithm:

**Steps:**

1. Construct the tree in top-down divide and conquer recursive manner.
2. Initially, put all training set at root node.
3. Partition the input data recursively based on selected attributes.
4. Select test set at each node based on statically measure i.e. information gain.
5. These are terminating conditions:
    - All inputs are member of same class.
    - There is no input for partitioning.
    - No sample is left.

## 5.4 Naive Bayes

It is a probabilistic classifier based on Bayes theorem with strong assumption that all the features are not dependent on each other. Such assumption is known as class conditional independence. An important advantage of Naive Bayes is that it requires a very less amount of training data set for classification. It is one of the fast classifier since it works in a single scan. Bayes theorem give a way of finding posterior probability $P(c \mid x)$ from $P(x \mid c)$, P(c) and P(x). Naive Bayes classifier consider that effect of a predictor x (only value of x) on a given class c is not dependent on other predictors. Following is the formula for calculating posterior probability:

$$P(c \mid x) = \frac{P(x \mid c)\, P(c)}{P(x)}$$

where:

*P* (*c* | *x*): posterior probability of target class on given attribute.

*P* (*x* | *c*): probability of predictor on given class (like hood).

*P* (c): prior probability of the class.

*P* (x): prior probability of the predictor.

**Support Vector Machine**

Support Vector Machine (SVM) also known as Support Vector Network in machine learning is a supervised learning technique used for classification and regression. In simple, given a training examples set, each of them marked belonging to one of two categories. SVM training algorithm constructs a model that decides and assigns a new example falls into one category or the other. Hence SVM classifier is represented bya separating hyperplane. This hyperplane generatedfromtrainingset then classifies data from test set.

Suppose we have two classes shown in Figure 5.1, denoted by square and circle and two axis's x and y denoting features. SVM finds a hyperplane that classify all the training set into two classes.

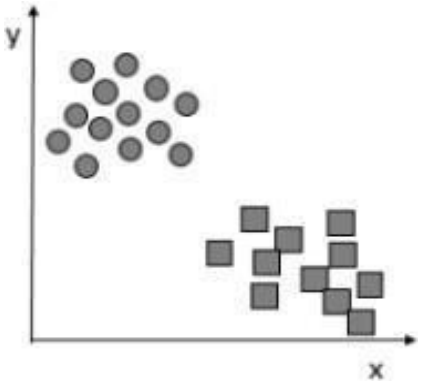



Figure 5.2: SVM Classifier with Two Classes

Figure 5.2 denotes some separable hyperplane according to SVM classifier. Among all hyperplanes, the best choice will be the hyperplane that leaves maximum margin from both the classes.

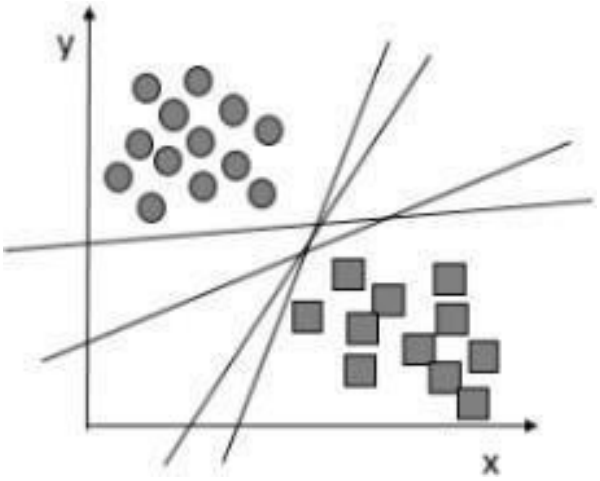



Figure 5.3: SVM Classifier with Hyperplane

**k-Nearest Neighbor**

k-Nearest Neighbor (k-NN) classifier is the simplest among all the classifiers and is used for both classification and regression. In this, input consists of k closest training sets in the feature space. Its output is class membership. If k=1, then object is directly assigned to single nearest neighbor class else object is assigned to that class in which object is most common in its k nearest neighbor.

**Random Forest**

Random Forest classifier works where Decision Tree fails. In other word, if trees are grown very deep or taken irregular shape i.e. over fit training set then for averaging multiple deep decision tree, random forests work on different part of same training set by generating multitude of decision trees during training time. The major belief with random forest method is that most of the tree can provide correct prediction of class for most of the data.

Figure 3.3 shows that three having node Y provide correct prediction because of their majority and tree having node N provide wrong prediction.

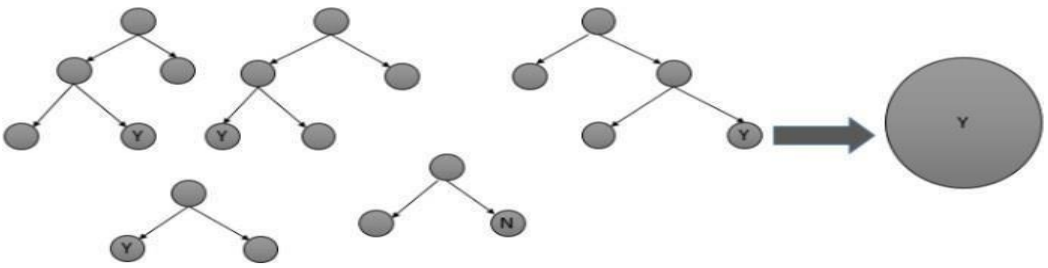



Figure 5.4: Random Forest Classifier

**Logistic Regression**

Logistic Regression, also known as logit regression is very popular technique used for classification and regression. This is simple and provides good performance. It is a discriminative probabilistic model that operates over vector inputs which are real valued and predicts the probability of an outcome that can have only two values (i.e. a dichotomy). The dimension of input vectors are features having no restriction against them being correlated.

Logistic Regression produces a logistic curve, which values lies between 0 and 1 as shown in Figure 3.4. Logistic regression is similar to linear regression, but the curve is constructed using natural logarithm rather than probability. The predictors do not have to be normally distributed or equal variance in each group.

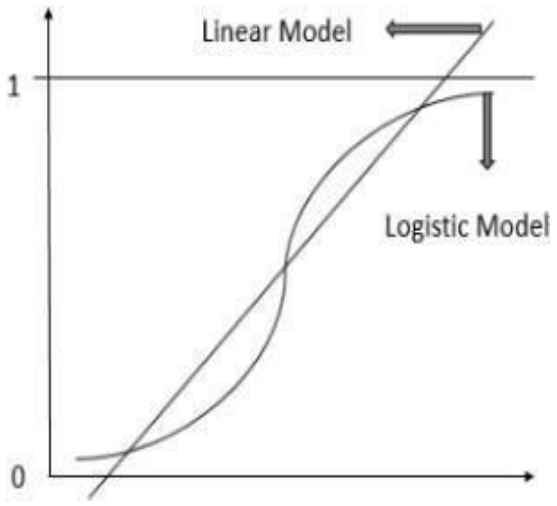



Figure 5.5: Logistic Model

**Semi-supervised Learning Technique**

Semi-supervised learning technique is a machine learning technique that uses a large amount of unlabeleddataandaveryfewlabeleddatasetfortraining. Semi-supervised learning lies between supervised learning (completely labeled data) and unsupervised learning (completely unlabeled data). Many researchers found that if a large amount of unlabeled data, when used with a few labeled data set, can produce good accuracy in term of learning problem.

**Assumptions in Semi-Supervised Technique**

Thereare threemainassumptionsinsemi-supervisedlearningtechniquewhichmakeitsimplerand easier. These are:

**Smoothness Assumption**

In the case of supervised learning, output varies smoothly with the distance on the basisofprior belief. In case ofsemi-supervised learning, densityof input is also taken into account. Hence, we can say that if two points x and y are in a high density region are considered to be close rather than x is in high density region and y is in low density region or vice-versa.

Figure 3.5 shows that x and y are close since they in high density region and x and z or y and z are not close since one is in high density region, others in low density.

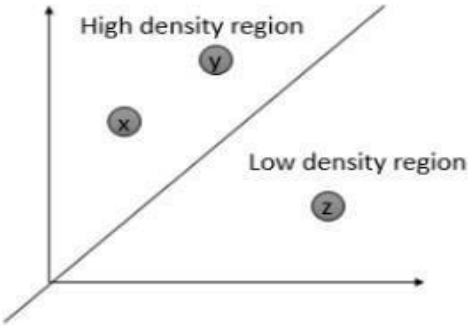



Figure 5.6: Smooth Assumption.

**Cluster Assumption**

Since points of each class form a cluster. Now assumption is that if two points x and y are in same cluster are considered to be in the same class, however two points in different cluster are not considered in the same class. Figure 3.6shows x and y are member of same class however x and z are not.

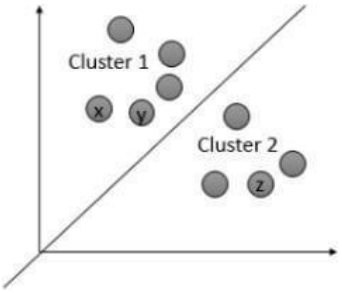



Figure 5.7: Cluster Assumption

**Manifold Assumption**

Manifold assumption is different from above two assumptions. The assumption is that, a high dimensional data lies in approximately low dimension manifold. Such assumption is useful when we have a high dimensional data and it is hard to model.

**Semi-supervised Learning Methods**

There are many semi-supervised learning methods are used in the area of machine learning. Some of them are generative method, self-training method, co-training method, graph based method etc.

## 5.6 Generative Method

Generative methods are one of the oldest semi-supervised learning method. This method is based on p(x,y) = $p(x \mid y)$p(y), where $p(x \mid y)$ is a recognizable distribution. In this, first mixture component of large volume of unlabeled data is recognized then perform labeling. It is an inductive mixture with very less parameter.

**Self-Training**

Self-Training is very common method used in semi-supervised learning method. In this, first classifier is trained with few selected labeled data set and then classifier classifies unlabeled data sets. Now predicted data sets are appending with selected label data and then classifier is retrained with this data set and the process is repeated. The process of retraining the data again and again is called as bootstrapping or self-teaching. These are the basic steps for Self-Training method.

**Steps:**

1) The classifier is trained with few labeled data (completely positive and completely negative).
2) The classifier is run with that data set which is weak label on the basis of maximum likelihood ratio.
3) Un label data set is label with the output of detector.
4) A subset is selected from these labeled data set using some featuresmetric.
5) Process is repeated until all data set to be trained.

**Co-Training**

Co-Training method is based on different features containing by data. It is assumed that each sample consists two different feature sets that give different information about the instances. These two views should be conditional independent. From each view, class of instances are predicted accurately. Co-training begins with learning an individual classifier for each view. With the help of these classifiers, we label unlabeled data set.

**Multiview Learning**

It is extended version of Co-Training method in which we use multiple views rather than two views. Rest steps are same as Co-Training method.

**Graph Based Method**

Graph based method is totally based on graph where each node represents data set (labeled and unlabeled) both and edges represent similarity between data. This method follows smoothness assumption.One important advantage with this method is that it does not require any parameters

# 6. IMPLEMENTATATION AND RESULTS

Reading data from CSV(comma separated values) is a fundamental necessity in Data Science. Often, we get data from various sources which can get exported to CSV format so that they can be used by other systems. The Panadas library provides features using which we can read the CSV file in full as well as in parts for only a selected group of columns and rows.

The CSV file is a text file in which the values in the columns are separated by a comma. Let's consider the following data present in the file named input.csv. You can create this file using windows notepad by copying and pasting this data. Save the file as input.csv using the save As All files(*.*) option in notepad.

```
import pandas as pd

data = pd.read_csv('path/input.csv')

print (data)
```

**Operations using NumPy**

NumPy is a Python package which stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Using NumPy, a developer can perform the following operations −

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

**KeyFeaturesofPandas:**

- Fast and efficient DataFrame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.

- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

**6.2 About MySQL:**

**MySQL** is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. The SQL phrase stands for Structured Query Language. Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, phpBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World WideWeb products, including Wikipedia, Google, Facebook, and Twitter.

MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout it's history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0,SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration formodern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software — including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production-tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription.

MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

**6.3 Project Data Base:**

```
CREATE DATABASE `spamreview` /*
/*Table structure for table `reviews` */
CREATE TABLE `reviews` (
  `REVIEWID` int(100) NOT NULL AUTO_INCREMENT,
  `HOTELNAME` varchar(5000) DEFAULT NULL,
  `REVIEWTEXT` varchar(5000) DEFAULT NULL,
  `SPAM` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`REVIEWID`)
)
```

## SAMPLE CODE:

```
Import
  re

        import random
        import json
        from DatabaseCreator import getHandle,clearHandle,DATABASE
        import features
        import SentimentScorer
        limit = ""
        i = 0


        #def main():
         #    pass









        # debug mode


        ##def getFeatures():
        ##     text = "I hate this hotel."
        ##     print SentimentScorer.senti(text)
```

```
def getFeatures(text):
    return features.majorfunc(text)


def getSentiment(text):
    sentences = re.split(r' *[\.\?!][\'"\)\]]* *', text)
    count = text.count('.')+ text.count ('!')+ text.count('?')
    avg = 0
    for sent in sentences:
        avg += SentimentScorer.senti(sent)
    return avg/(len(sentences))




def makeFeatureVectors(TrainingSet, TestSet):


    #REVIEWID, HOTELNAME, REVIEWTEXT, POLARITY, SPAM
    TrainingFeatures = []
    TestFeatures = []
    TrainingClass = []
    TestClass = []
    Features1 = []
    Features2 = []
    list1 = []


    for entry in TrainingSet:
        Features1 = getFeatures(entry[2]) # entry[2] has review text
        Features1.append(getSentiment(entry[2]))
        Features1.append(entry[3])   # ----------POLARITY
        TrainingFeatures.append(Features1)
        Features1 = []
        TrainingClass.append(entry[4])


    for entry in TestSet:
        Features2 = getFeatures(entry[2]) # entry[2] has review text
        Features2.append(getSentiment(entry[2]))
        Features2.append(entry[3])   # ----------POLARITY
        TestFeatures.append(Features2)
        Features2 = []
        TestClass.append(entry[4])
```

```
    return TrainingFeatures, TestFeatures, TrainingClass, TestClass


# small dataset
##
### make TrainingSet[] and TestSet[]
##
##def makeTrainingSet(K):
##
##    TrainingSet = []
##    TestSet = []
##    conn = getHandle(DATABASE)
##
##    # Positive Spam
##    end = 40
##
##    query = "select REVIEWID, HOTELNAME, REVIEWTEXT, POLARITY, SPAM from REVIEWS
where POLARITY = 1 and SPAM = 1 "+limit+";"
##    cursor = conn.execute(query) # connect the database
##    data = cursor.fetchall() # fetch all queries
##    random.shuffle(data) # randomly shuffle all the data
##    length = len(data)
##    train = int((K * length) / 1000) # number of queries fetched
##    temp = data[:train]
##    for entry in temp:
##        TrainingSet.append(entry) # append in TrainingSet[]
##    temp = data[train:end]
##    for entry in temp:
##        TestSet.append(entry) # append in TestSet[]
##
##    # Positive Ham
##
##    query = "select REVIEWID, HOTELNAME, REVIEWTEXT, POLARITY, SPAM from REVIEWS
where POLARITY = 1 and SPAM = 0 "+limit+";"
##    cursor = conn.execute(query)
##    data = cursor.fetchall()
##    random.shuffle(data)
##    length = len(data)
##    train = int((K * length) / 1000)
##    temp = data[:train]
##
##    for entry in temp: TrainingSet.append(entry)
##    temp = data[train:end]
##    for entry in temp: TestSet.append(entry)
```

```
##
##    # Negative Spam
##
##    query = "select REVIEWID, HOTELNAME, REVIEWTEXT, POLARITY, SPAM from REVIEWS
where POLARITY = -1 and SPAM = 1 "+limit+";"
##    cursor = conn.execute(query)
##    data = cursor.fetchall()
##    random.shuffle(data)
##    length = len(data)
##    train = int((K * length) / 1000)
##    temp = data[:train]
##    for entry in temp: TrainingSet.append(entry)
##    temp = data[train:end]
##    for entry in temp: TestSet.append(entry)
##
##    # Negative Ham
##
##    query = "select REVIEWID, HOTELNAME, REVIEWTEXT, POLARITY, SPAM from REVIEWS
where POLARITY = -1 and SPAM = 0 "+limit+";"
##    cursor = conn.execute(query)
##    data = cursor.fetchall()
##    random.shuffle(data)
##    length = len(data)
##    train = int((K * length) / 1000)
##    temp = data[:train]
##    for entry in temp: TrainingSet.append(entry)
##    temp = data[train:end]
##    for entry in temp: TestSet.append(entry)
##
##    clearHandle(conn)
##    print(TestSet)
##    return TrainingSet, TestSet


#large dataset


def makeTrainingSet(K):


    TrainingSet = []
    TestSet = []
    conn = getHandle(DATABASE)


    # Positive Spam
```

```
    query = "select REVIEWID, HOTELNAME, REVIEWTEXT, POLARITY, SPAM from REVIEWS where
POLARITY = 1 and SPAM = 1 "+limit+";"
    cursor = conn.execute(query) # connect the database
    data = cursor.fetchall() # fetch all queries
    random.shuffle(data) # randomly shuffle all the data
    length = len(data)
    train = int((K * length) / 100) # number of queries fetched
    temp = data[:train]
    for entry in temp:
        TrainingSet.append(entry) # append in TrainingSet[]
    temp = data[train:]
    for entry in temp:
        TestSet.append(entry) # append in TestSet[]


    # Positive Ham


    query = "select REVIEWID, HOTELNAME, REVIEWTEXT, POLARITY, SPAM from REVIEWS where
POLARITY = 1 and SPAM = 0 "+limit+";"
    cursor = conn.execute(query)
    data = cursor.fetchall()
    random.shuffle(data)
    length = len(data)
    train = int((K * length) / 100)
    temp = data[:train]


    for entry in temp: TrainingSet.append(entry)
    temp = data[train:]
    for entry in temp: TestSet.append(entry)


    # Negative Spam


    query = "select REVIEWID, HOTELNAME, REVIEWTEXT, POLARITY, SPAM from REVIEWS where
POLARITY = -1 and SPAM = 1 "+limit+";"
    cursor = conn.execute(query)
    data = cursor.fetchall()
    random.shuffle(data)
    length = len(data)
    train = int((K * length) / 100)
    temp = data[:train]
```

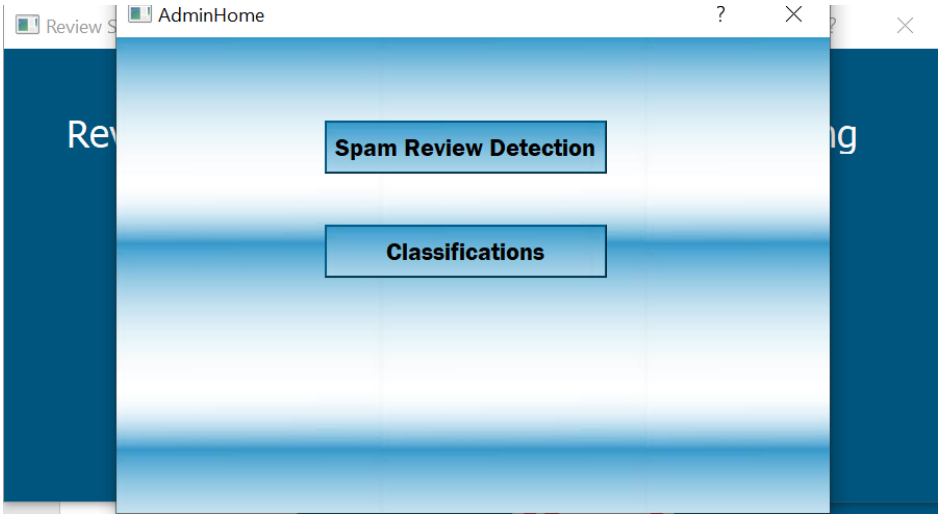
```
    for entry in temp: TrainingSet.append(entry)
    temp = data[train:]
    for entry in temp: TestSet.append(entry)


    # Negative Ham


    query = "select REVIEWID, HOTELNAME, REVIEWTEXT, POLARITY, SPAM from REVIEWS where
POLARITY = -1 and SPAM = 0 "+limit+";"
    cursor = conn.execute(query)
    data = cursor.fetchall()
    random.shuffle(data)
    length = len(data)
    train = int((K * length) / 100)
    temp = data[:train]
    for entry in temp: TrainingSet.append(entry)
    temp = data[train:]
    for entry in temp: TestSet.append(entry)


    clearHandle(conn)
    print(TestSet)
    return TrainingSet, TestSet




def saveToFile(filename,content):
    f = open(filename,'w')
    json.dump(content,f)
    f.close()


# creating training and test data and saving features to files


def main():
    TrainingSet = []
    TestSet = []


# debug
```

```
##     TrainingSet = [[1889, "sheraton", "I walked into this beautiful hotel and knew
that I would have a wonderful stay in Chicago. After walking around the city all day,
I was always excited to relax in this cleanly and superbly comforting place. The
service was friendly and I especially liked the location. What a Treat!\n", 1, 1],
[1802, "hyatt", "The Hyatt Regency in Chicago was a wonderful experience for my
husband and I as we traveled through the area on vacation. After checking in we simply
relaxed in the beautiful atrium for a while before going to our room. To our delight
the room was very clean, ultra-modern and furnished with sleek and comfortable
furniture. What really impressed us about the room was how spacious it was and the
stunning river front views. We were pressed for time to make dinner so decided the
Bistro sounded like the best option and it turned out to be a great choice! The food
was simply fabulous. After dinner we were able to try out the fitness center, which
ended up making our gym at home seem inadequate. This was just what we needed as a
restful base for our Chicago vacation and we will definitely stay here again.\n", 1,
1]]
##     TestSet = [[1597, "talbott", "We loved the Talbott. The location was
fabulous...very close to the shopping. The hotel is smaller, but the rooms are very
nice. They are very spacious and are able to be darkened to sleep well. Also, the beds
are some of the most comfortable I have ever slept in. I might also add that our room
at queen beds. The staff is all friendly and puts them themselves out for you. The
hotel has a cute outdoor eating area on the sidewalk that has flowers, etc. It looks
like a scene out of Europe. I would highly recommend the hotel. We were there with our
daughters; ages 10 and 13. \n", 1, 0], [1535, "sheraton", "i stayed at this hotel for
a week with my family this hotel is huge so clean has comfy beds foods great staff
couldn,t of being any nicer. stayed here for thanksgiving it was fabulous second time
in chicago and still want to go back again ... the hotel is within walkin distance to
both state street and michicgan avenue.There,s taxis right outside the hotel and the
concierge can also ring for mini buses if your travelling with a big crowd.there,s also
a cinema 2 mins away opposite the hotel . Perfect Hotel for a great get away ...
\n", 1, 0]]


    TrainingSet, TestSet = makeTrainingSet(90) #create the training and test files
# To make training and test data


    saveToFile('./machinelearn/TrainingSet',TrainingSet)
    saveToFile('./machinelearn/TestSet',TestSet)


    print"Training and Test sets created"


# To make feature files
```

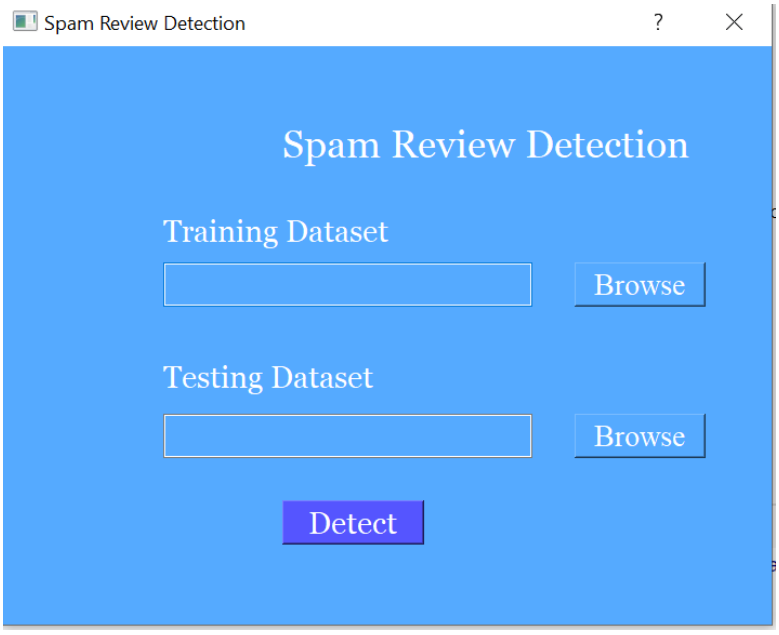
```
    TrainingFeatures = []
    TestFeatures = []
    TrainingClass = []
    TestClass = []


    TrainingFeatures,TestFeatures, TrainingClass, TestClass =
makeFeatureVectors(TrainingSet, TestSet)


    saveToFile('./machinelearn/TrainingFeatures',TrainingFeatures)
    saveToFile('./machinelearn/TestFeatures',TestFeatures)
    saveToFile('./machinelearn/TrainingClass',TrainingClass)
    saveToFile('./machinelearn/TestClass',TestClass)


    print"Training and Test features created"


##    getFeatures()


if __name__== "__main__":
    print('This works")
    main()
    print('and stops')
```

## RESULTS:

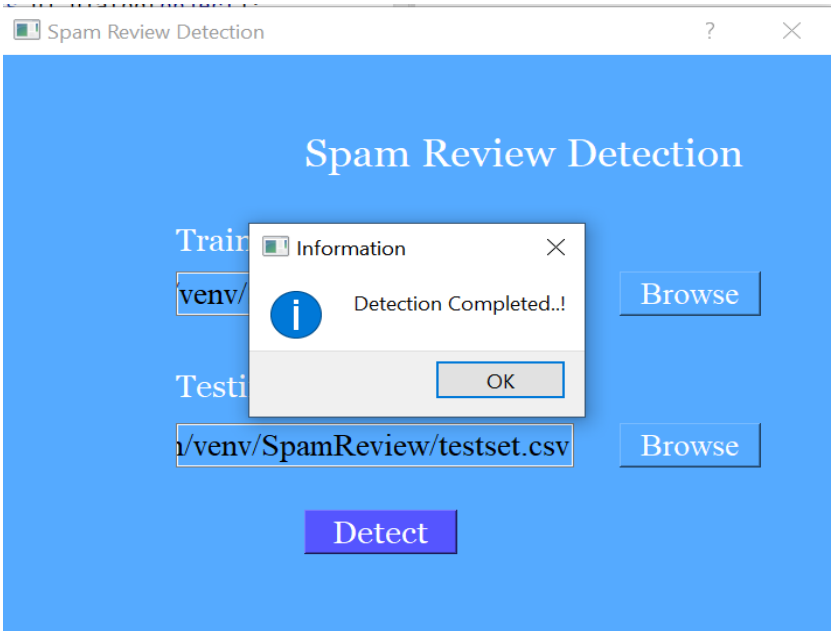
Step 1: Click on spam icon.



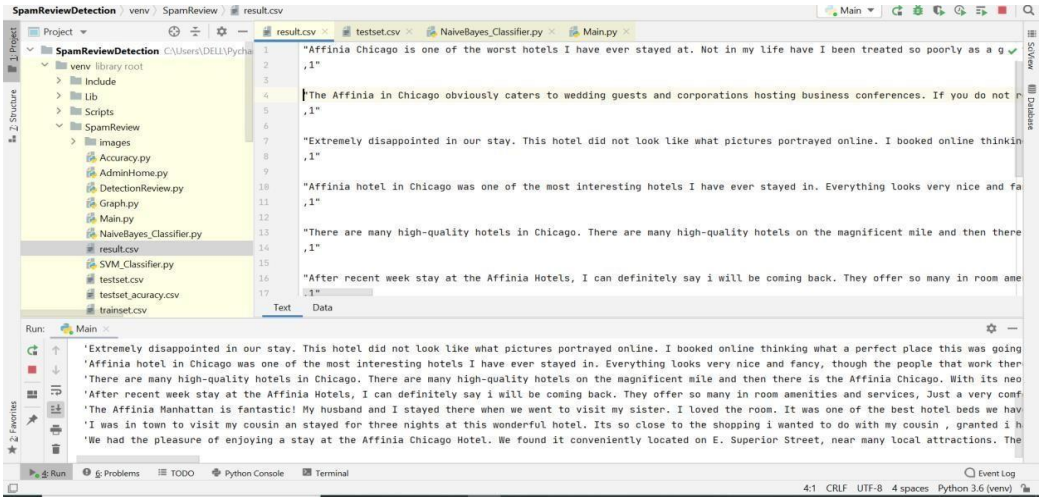

Step 2: Click on " Spam Review Detection" to check spam reviews.



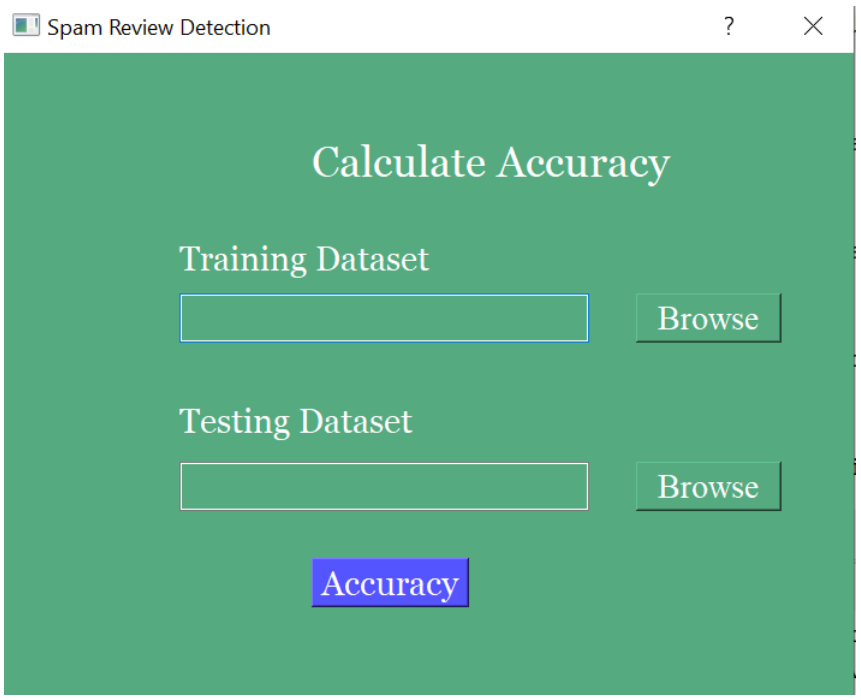
Step 3: Select training and testing datasets for detection.



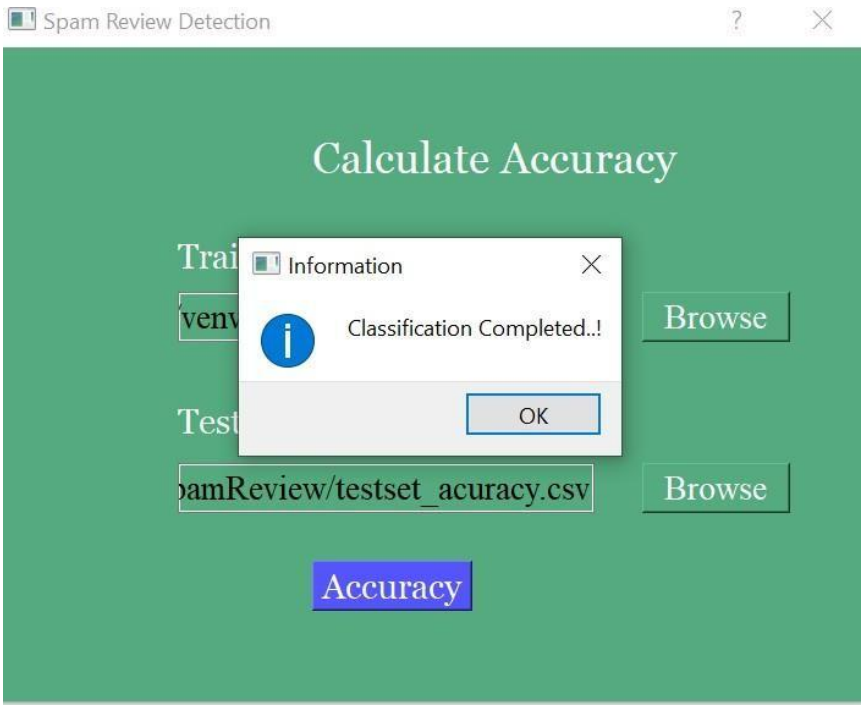

Step 4: After selection, click on detect.

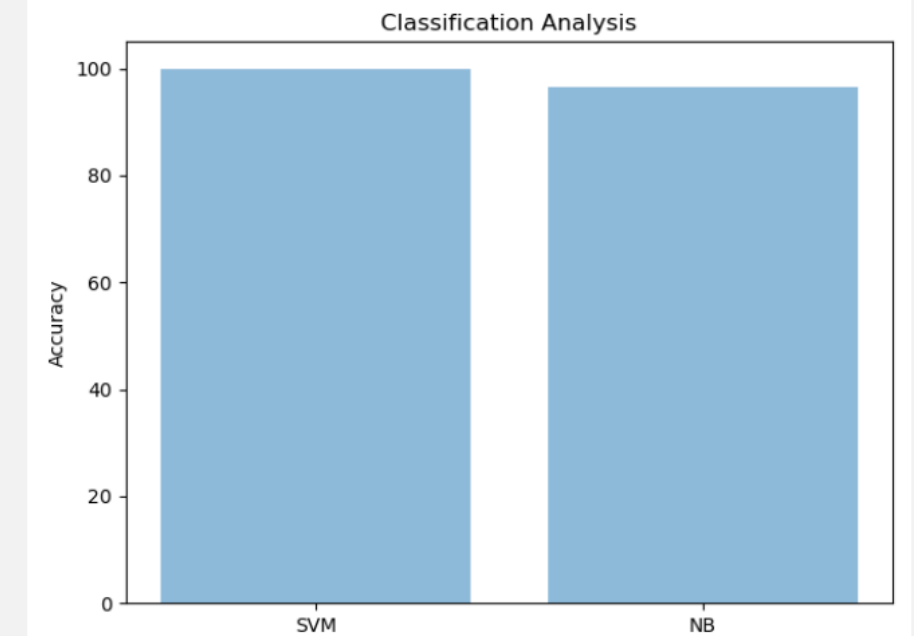


Step 6: This is the results, where it shows the spam and non spam reviews.




Step 7: To check accuracy, click on classification.






Step 8: Click on Accuracy, for the analysis of classification.

# 7. SYSTEM TESTING

Testing is the debugging program is one of the most critical aspects of the computer programming triggers, without programming that works, the system would never produce an output of which it was designed. Testing is best performed when user development is asked to assist in identifying all errors and bugs. The sample data are used for testing. It is not quantity but quality of the data used the matters of testing. Testing is aimed at ensuring that the system was accurately an efficiently before live operation commands.

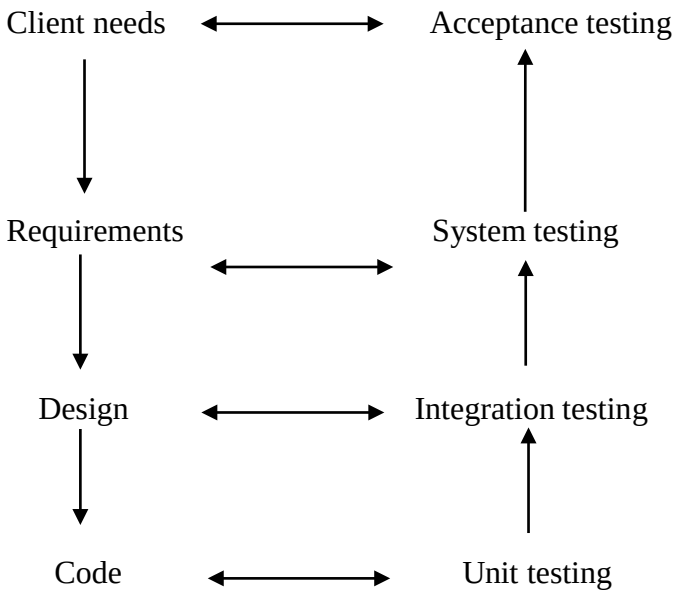
## 7.1 Testing objectives:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say, testing is a process of executing a program with intent of finding an error.

A successful test is one that uncovers an as yet undiscovered error. A good test case is one that has probability of finding an error, if it exists. The test is inadequate to detect possibly errors. The software more or less confirms to the quality and reliable standards.

**Levels of Testing:** In order to uncover present in different phases we have the concept of levels of testing.

**The basic levels of Testing:**




7.1 Figure: Levels of Testing

# 8.PROS AND CONS

i. We can identify fake products.
ii. Users are safe from getting scammed.
iii. We can identify and take action on fake sellers

It is hard to determine if it is a genuine or a fake product if the product is too good and all reviews are positive.
It uses user data to determine if a review is fake or not which brings privacy issues.

# 9.RESULTS SCREENSHOTS

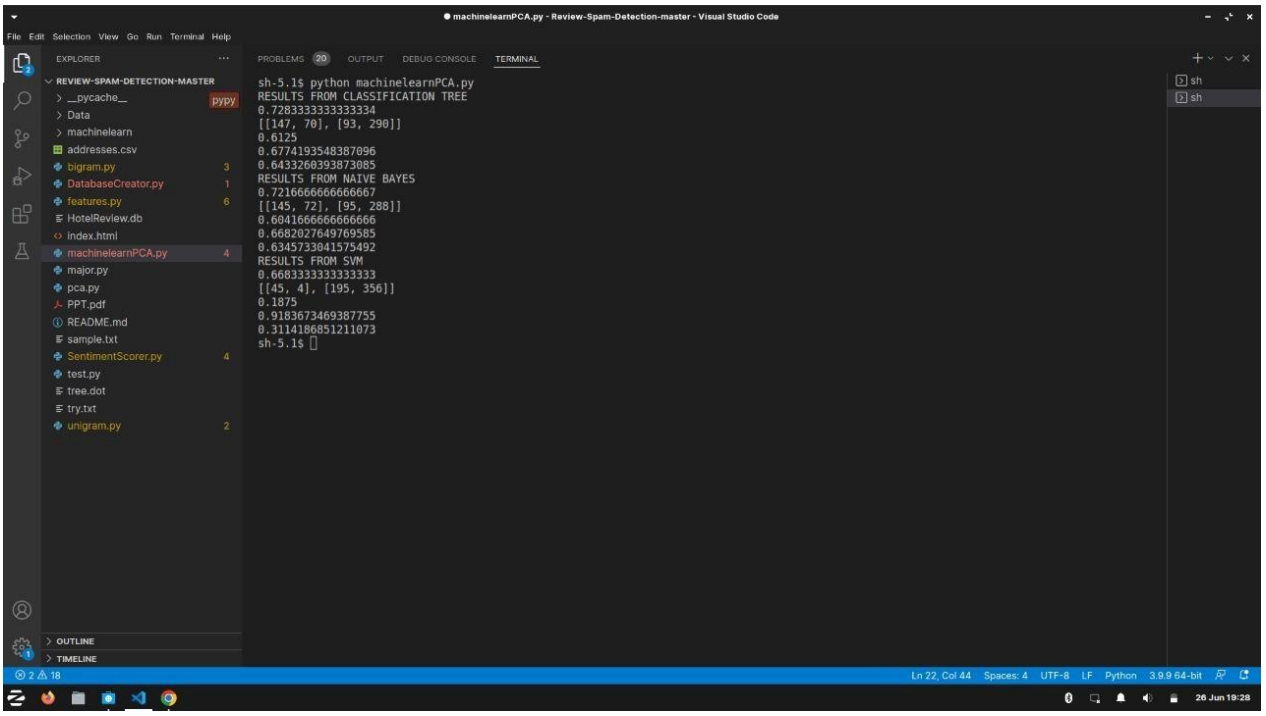



Fig.8.1 Results screenshot

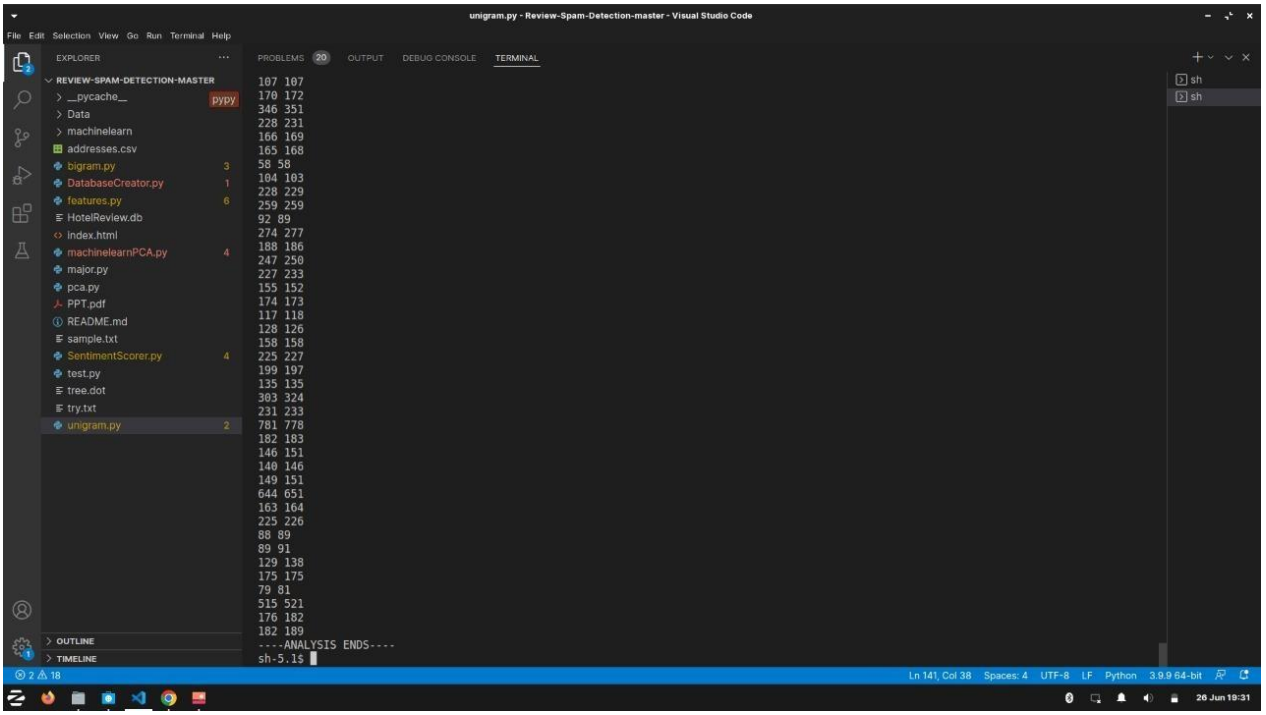



Fig.8.1.2 Results screenshot

# 10.CONCLUSION

In recent years, review spam detection has received significant attention in both business and academia due to the potential impact fake reviews can have on consumer behavior and purchasing decisions. Supervised learning is the most frequent machine learning approach for performing review spam detection; however, obtaining labeled reviews for training is difficult and manual identification of fake reviews has poor accuracy. This has led to many experiments using synthetic or small datasets. Features extracted from review text (e.g., bag of words, POS tags) are often used to train spam detection classifiers. An alternative approach is to extract features related to the metadata of the review, or features associated with the behavior of users who write the reviews. Disparities in performance of classifiers on different datasets may indicate that review spam detection may benefit from additional cross domain experiments to help develop more robust classifiers. Multiple experiments have shown that incorporating multiple types of features can result in higher classifier performance than using any single type of feature.

One of the most notable observations of current research is that experiments should use real world data if possible. Despite being used in many studies, synthetic or artificially generated datasets have been shown to give a poor indication of performance on real world data. As it is difficult to procure accurately labeled real-world datasets, unsupervised and semi-supervised methods are of interest. While unsupervised and semi-supervised methods are currently unable to match the performance of supervised learning methods, research is limited and results are inconclusive, warranting further investigation. A possibility for a less labor-intensive means of generating labeled training data is to find and label duplicate reviews as spam. Multiple studies have shown duplication, or near duplication, of review content is a strong indicator of review spam. Another data related concern is that real world data may be highly class imbalanced, as there are currently many more truthful than fake reviews online. This could be addressed through data sampling and ensemble learning techniques. A final concern related to quality of data is the presence of noise, particularly class noise due to mislabeled instances. Ensemble methods, and experiments with different levels of class noise, could be used to

evaluate the impact of noise on performance and how its effects may be reduced.

As review text is an important source of information and tens of thousands of text features can easily be generated based on this text, high dimensionality can be an issue. Additionally, millions of reviews are available to be used to train classifiers, and training classifiers from a large, highly dimensional dataset is computationally expensive and potentially impractical. Despite this, feature selection techniques have received little attention. Many experiments have avoided this issue by extracting only a small number of features, avoiding the use of n-grams, or by limiting number of features through alternative means such as using term frequencies to determine what n-grams are included as features. Further work needs to be conducted to establish how many features are required and what types of features are the most beneficial. Feature selection should not be considered optional when training a classifier in a big data domain with potential for high feature dimensionality. Additionally, we could find no studies that incorporated distributed or streaming implementations for learning from Big Data into their spam detection frameworks.

# 11.Future Scope

Supervised learning is the most frequent machine learning approach for performing review spam detection; however, obtaining labeled reviews for training is difficult and manual identification of fake reviews has poor accuracy. This has led to many experiments using synthetic or small datasets. Features extracted from review text (e.g., bag of words, POS tags) are often used to train spam detection classifiers. An alternative approach is to extract features related to the metadata of the review, or features associated with the behavior of users who write the reviews.unsupervised and semi-supervised methods are currently unable to match the performance of supervised learning methods, research is limited and results are inconclusive, warranting further investigation. Additionally, we could find no studies that incorporated distributed or streaming implementations for learning from Big Data into their spam detection frameworks

# 12.References

[1] E. F. Cardoso, R. M. Silva, and T. A. Almeida, ‘‘Towards automatic filtering of fake reviews,’’ Neurocomputing, vol. 309, pp. 106–116, Oct. 2018.
[2] L. Da Xu, W. He, and S. Li, ‘‘Internet of Things in industries: A survey,’’
[3] Y. Ren and Y. Zhang, ‘‘Deceptive opinion spam detection using neural network,’’ in Proc. 26th Int. Conf. Comput. Linguistics: Tech.
[4] N. Jindal and B. Liu, ‘‘Opinion spam and analysis,’’ in Proc. Int. Conf.
[5] A. Heydari, M. Tavakoli, and N. Salim, ‘‘Detection of fake opinions using time series,’’ Expert Syst. Appl., vol. 58, pp. 83–92, Oct. 2016.
[6] L. Li, W. Ren, B. Qin, and T. Liu, ‘‘Learning document representation for deceptive opinion spam detection,’’in Chinese Computational Linguistics
[7] H. Aghakhani, A. Machiry, S. Nilizadeh, C. Kruegel, and G. Vigna, ‘‘Detecting deceptive reviews using generative adversarial networks,’’ in
[8] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, ‘‘Fake review detection: Classification and analysis of real and pseudo reviews,’’ Univ. Illinois Chicago, Chicago, IL, USA, Tech. Rep. UIC-CS-03-2013, 2013.
[9] R. Yafeng, J. Donghong, Z. Hongbin, and Y. Lan, ‘‘Deceptive reviews detection based on positive and unlabeled learning,’’ J. Comput. Res.
[10] R. Y. K. Lau, S. Y. Liao, R. C.-W. Kwok, K. Xu, Y. Xia, and Y. Li, ‘‘Text mining and probabilistic language modeling for online review spam detection,’’ ACM Trans. Manage. Inf. Syst., vol. 2, no. 4, pp. 1–30, Dec. 2011