Intelligent Interface for Fake Product Review Monitoring and Removal

PROJECT REPORT

Submitted by

ARUN KUMAR P	**(19104010)**
MANI PRABHU T	**(19104055)**
MOHAN M	**(19104060)**

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PAAVAI ENGINEERING COLLEGE,

NAMAKKAL.

ANNA UNIVERSITY: CHENNAI 600 025

DECEMBER-2022

BONAFIDE CERTIFICATE

Certified that this project report **"INTELLIGENT INTERFACE FOR FAKE PRODUCT REVIEW MONITORING AND REMOVAL"** is the Bonafede work of **ARUN KUMAR P (19104010), MANI PRABHU T (19104055), MOHAN M (19104060),** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.D.BANUMATHY,M.E.,Ph.D	Mrs.K.SANTHANALAKSHMI, M.TECH.,
HEAD OF THE DEPARTMENT	**ASSISTANT PROFESSOR,**
Department of Computer Science	Department of Computer Science
and Engineering	and Engineering
Paavai Engineering College,	Paavai Engineering College,
Namakkal-637 018	Namakkal-637 018

Submitted for the university project viva-voce held on ……………………………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

ARUN KUMAR P, MANI PRABHU T and **MOHAN M** hereby that the project report titled **INTELLIGENT INTERFACE FOR FAKE PRODUCT REVIEW MONIORING AND REMOVAL** done by us under the guidance of **Mrs K. SANTHANALAKSHMI., M.TECH.,** at **PAAVAI ENGINEERING COLLEGE, NAMAKKAL** is submitted in partial fulfillment of the requirements for the award of BACHELOR **OF ENGINEERING degree in COMPUTER SCIENCE AND ENGINEERING**. Certified further that, to the best of my knowledge, the work report edhere in does not form part of any other project report or dissertion on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

DATE:

PLACE: **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

With pleasure we express our profound gratitude to our beloved chairman **Shri.CA.N.V. NATARAJAN,B.Com,FCA.,** and gratitude to our beloved correspondent **Smt.N.MANGAINATARAJAN,M.Sc.,** for giving motivation and providing all necessary facility for the successful completion of this project. It is our privilege to thank our beloved Director Adiministration **Dr.K.K.RAMASAMY,M.E,Ph.D.,**and our principal **Dr.M.PREMKUMAR., M.E,Ph.D.,** for their moral support and deeds in bringing out this project successfully. We extend our greatfulness to **Dr.D.BANUMATHY., M.E,Ph.D** Head of Department, Department of Computer Science and Engineering,for her encouragement and constant support in successful completion of our project. We convey our thanks to our supervisor **Mrs.K.SANTHANA LAKSHMI.,M.TECH.,** Associate Professor,Department of Computer Science and Enginering for being more informative and providing suggestions regarding this project all the time. We thank our project coordinator **Mrs.P.RENUKADEVI.M.E., (Ph.D)** for her meticulous guidance and timely help during the project work. We express thanks to all our department staff members,friends and family members for their encouragement and advice to do the project work with full interest and enthusiasm.

Abstract

As the trend to shop online is increasing day by day and more people are interested in buying the products of their need from the online stores. This type of shopping does not take a lot of time of a customer. Customer goes to online store, search the item of his/her need and place the order. But, the thing by which people face difficulty in buying the products from online store is the bad quality of the product. Customer place the order only by looking at the rating and by reading the reviews related to the particular product. Such comments of other people are the source of satisfaction for the new product buyer. Here, it may be possible that the single negative review changes the angle of the customer not to buy that product. In this situation, it might possible that this one review is fake. So, in order to remove this type of fake reviews and provide the users with the original reviews and rating related to the products, we proposed a Fake Product Review Monitoring and Removal System (FaRMS) which is an Intelligent Interface and takes the Uniform Resource Locator (URL) related to products of Amazon, Flipkart and Daraz and analyzes the reviews, and provides the customer with the original rating. It is a unique quality of the proposed system that it works with the three e-commerce Websites and not only analyzes the reviews in English but also the reviews written in Urdu and Roman Urdu. Previous work on fake reviews does not support feature to analyze the reviews written in languages like Urdu and Roman Urdu and cannot handle the reviews of multiple e-commerce Websites. The proposed work achieved the accuracy of 87%in detecting fake reviews of written in English by using intelligent learning techniques which is greater than the accuracy of the previous systems.

TABLE OF CONTENTS

LIST OF FIGURE

Chapter	Title	Page NO
4	**OTHER REQUIREMENTS**	
	System requirements 1	
	System requirements 2	
	System requirements 3	
	System requirements 4	
6	**SYSTEM ARCHITECTURE**	
	System architecture	

1.INTRODUCTION:

- There are different ways to shop like you can buy a specific thing of your need by going to a store or mall. In this style of shopping the seller gives you the feedback of the product, you do not know whether he/she is giving a fake feedback or original. Because, it is upon seller honesty, how much the seller is true in his/her words and you have to carefully examine the product because you do not have any other option in examining the product. If you don't pay attention in buying that product then it may be proved a waste for you. On the other hand, nowadays source of shopping has been changed.
- You can buy the products from the online stores of different brands. Here, you have to place the order without seeing and examining the original product. You read the reviews and buy the product. Therefore, you are dependent on the reviews about the product. These reviews may be the original or fake. The customer wants to buy an original and reliable product, which is possible only when you get the original feedback related to that product. Research shows that U.S. shoppers spend $6 billion in Black Friday sale 2018. Americans spend 36% of the shopping budget online. In 2017, E-commerce stores earned $2.3 trillion in sales and expected to reach $4.5 trillion by 2021.
- Today, almost 12-24 million ecommerce stores are operating around the world. Study found that 61% of Amazon reviews that belongs to Electronics Category are fake. There are some websites which are working to detect the fake reviews. Fake spot is an online Website that detects fake reviews using suspicious patterns and reviewers activity. As in the process of buying the product from the online stores you have to read all the reviews one by one to check for the quality of that product and to get a good quality product. It is a very time consuming process. Here this possibility also falls that the reviews may be fake or original.

1.2 PURPOSE AND SCOPE:

- The purpose of an intelligent interface for fake product review monitoring and removal is to provide an efficient and effective tool for identifying and removing fake reviews posted on online platforms.
- The scope of this tool would cover various online platforms such as e-commerce websites, social media platforms, and other online forums where product reviews are posted.The intelligent interface would use machine learning algorithms and natural language processing techniques to analyze the reviews and identify patterns that indicate fake reviews.
- The tool would also have the capability to identify suspicious behavior such as a large number of reviews posted by a single user or a sudden spike in the number of reviews for a particular product.
- The ultimate goal of this tool is to provide a better and more reliable shopping experience for consumers by ensuring that the reviews they read are genuine and trustworthy. This would help to increase customer confidence in online shopping and ultimately benefit businesses that rely on positive reviews to drive sales.In summary, the purpose of an intelligent interface for fake product review monitoring and removal is to improve the quality and integrity of product reviews by automatically identifying and removing fake reviews from online platforms.
- The scope of this tool would cover various online platforms, and the ultimate goal is to provide a more reliable shopping experience for consumers and benefit businesses that rely on positive reviews.
- The system would also provide businesses with the ability to flag and remove fake reviews, allowing them to maintain a high level of authenticity in their product listings. This would ultimately improve the customer experience, as shoppers can trust that the reviews they are reading are genuine and reflective of the product's true quality.
- Overall, the intelligent interface for fake product review monitoring and removal would help businesses build credibility and trust with their customers while also providing them with a powerful tool to combat fraudulent reviews.

1.3 PROBLEM STATEMENT:

- The problem with fake product reviews is that they can mislead consumers, negatively impact a business's reputation, and ultimately result in lost sales. With the rise of online shopping, customer reviews have become a crucial part of the purchasing process, with many consumers relying on them to make informed decisions about products.
- Unfortunately, there are individuals and businesses who engage in the practice of posting fake reviews to manipulate a product's rating and deceive potential buyers. This not only hurts honest businesses who rely on genuine customer feedback but also undermines the trust that consumers have in online reviews.
- The problem is further compounded by the sheer volume of reviews that businesses receive, making it difficult and time-consuming to manually monitor and identify fake reviews. This creates a need for an intelligent interface that can quickly and accurately analyze reviews, detect suspicious patterns, and help businesses remove fraudulent reviews from their listings.
- Thus, the problem statement for fake product review monitoring and removal is: In the era of online shopping, businesses face the challenge of dealing with the negative effects of fake product reviews, which mislead consumers, harm a business's reputation, and result in lost sales. To address this issue, an intelligent interface is needed to assist businesses in identifying and removing fraudulent reviews, improving the integrity of online reviews, and ultimately enhancing the customer experience.
- Therefore, the problem statement for an intelligent interface for fake product review monitoring and removal is: Existing solutions for detecting and removing fake product reviews may be limited in their effectiveness, and businesses may lack the resources or expertise to implement and maintain such systems. This creates a need for an intelligent interface that is user-friendly, accessible, and capable of accurately analyzing reviews, detecting suspicious patterns, and assisting businesses in removing fraudulent reviews from their listings, thereby enhancing the integrity of online reviews and improving the customer experience.

2. PROJECT ANALYSIS:

2.1 Literature review:

- Fake Product Review Monitoring Using Opinion Mining Product reviews play an important role in deciding the sale of a particular product on the e-commerce websites or applications like Flipkart, Amazon, Snap deal, etc. In this paper, we propose a framework to detect fake product review or spam reviews by using Opinion Mining. The Opinion mining is also known as Sentiment Analysis.
- In sentiment analysis, we try to figure out the opinion of a customer through a piece of text. We first take the review and check if the review is related to the specific product with the help of Decision tree. We use Spam dictionary to identify the spam words in the reviews. In Text Mining we apply several algorithms and on the basis of these algorithms we get the specific results.

2.2 Black Friday Purchases:

- The top selling products on Black Friday were laptops, with lots of people taking advantage of the savings to pick up a new machine. Other top sellers were games like *God of War and Let's Go Pikachu*, and children's toys like Fingerlings.
- This is a reminder that although Black Friday in increasingly thought of as a day for adults to pick up big ticket items like electronics and appliances, the market for kid's toys is still a significant part of sales.

2.3 Study Finds 61 Percent of Electronics Reviews on Amazon Are Fake:

- As the trend to shop online is increasing day by day and more people are interested in buying the products of their need from the online stores. This type of shopping does not take a lot of time of a customer. Customer goes to online store, search the item of his/her need and place the order. But, the thing by which people face difficulty in buying the products from online store is the bad quality of the product.
- Customer place the order only by looking at the rating and by reading the reviews related to the particular product. Such comments of other people are the source of satisfaction for the new product buyer. Here, it may be possible that the single negative review changes the angle of the customer not to buy that product.

- In this situation, it might possible that this one review is fake. So, in order to remove this type of fake reviews and provide the users with the original reviews and rating related to the products, we proposed a Fake Product Review Monitoring and Removal System (FaRMS) which is an Interface and takes the Uniform Resource Locator (URL) related to products of Amazon, Flipkart and analyzes the reviews, and provides the customer with the original rating.

2.4 Urdu Sentiment Analysis:

- The entire world is transforming quickly under the present innovations. The Internet has become a basic requirement for everybody with the Web being utilized in every field. With the rapid increase in social network applications, people are using these platforms to voice them their opinions with regard to daily issues.
- Gathering and analyzing peoples' reactions toward buying a product, public services, and so on are vital. Sentiment analysis (or opinion mining) is a common dialogue preparing task that aims to discover the sentiments behind opinions in texts on varying subjects.
- In recent years, researchers in the field of sentiment analysis have been concerned with analyzing opinions on different topics such as movies, commercial products, and daily societal issues. Twitter is an enormously popular microblog on which clients may voice their opinions.
- Opinion investigation of Twitter data is a field that has been given much attention over the last decade and involves dissecting "tweets" (comments) and the content of these expressions. As such, this paper explores the various sentiments analysis applied to Twitter data and their outcomes.

2.5 Spotting Fake Reviewer Groups in Consumer Reviews:

- Opinionated social media such as product reviews are now widely used by individuals and organizations for their decision making. However, due to the reason of profit or fame, people try to game the system by opinion spamming (e.g., writing fake reviews) to promote or demote some target products.
- For reviews to reflect genuine user experiences and opinions, such spam reviews should be detected. Prior works on opinion spam focused on detecting fake reviews and individual fake reviewers.

- This paper studies spam detection in the collaborative setting, i.e., to discover fake reviewer groups. The proposed method first uses a frequent item set mining method to find a set of candidate groups.
- It then uses several behavioural models derived from the collusion phenomenon among fake reviewers and relation models based on the relationships among groups, individual reviewers, and products they reviewed to detect fake reviewer groups. Additionally, we also built a labelled dataset of fake reviewer groups. Although labelling individual fake reviews and reviewers is very hard, to our surprise labelling fake reviewer groups is much easier.
- We also note that the proposed technique departs from the traditional supervised learning approach for spam detection because of the inherent nature of our problem which makes the classic supervised learning approach less effective.
- Experimental results show that the proposed method outperforms multiple strong baselines including the state-of-the-art supervised classification, regression, and learning to rank algorithms.

2.6 What Yelp Fake Review Filter Might Be Doing?

- 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 an exception 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.

2.7 Fake Review Detection on Yelp Dataset and features:

- Online reviews have become an important factor when people make purchase and business decisions. The increasing popularity of online reviews also stimulates the business of fake review writing, which refers to paid human writers producing deceptive reviews to influence readers' opinions.
- Our project tackles this problem by building a classifier that takes there view text and the basic information of its reviewer as input and outputs whether the review is reliable.

- The learning algorithms we experimented include logistic regression, linear discriminate analysis, multinomial Naive Bayes, support vector machines and neural networks. The results show that the neural network performs the best with a detection accuracy of 81.92%.

2.8 Fake product review monitoring:

- As most of the people require review about a product before spending their money on the product. So people come across various reviews in the website but these reviews are genuine or fake is not identified by the user. In some review websites some good reviews are added by the product company people itself in order to make in order to produce false positive product reviews.
- They give good reviews for many different products manufactured by their own firm. User will not be able to find out whether the review is genuine or fake. To find out fake review in the website this "Fake Product Review Monitoring and Removal for Genuine Online Product Reviews Using Opinion Mining" system is introduced.
- This system will find out fake reviews made by posting fake comments about a product by identifying the IP address along with review posting patterns. User will login to the system using his user id and password and will view various products and will give review about the product.
- To find out the review is fake or genuine, system will find out the IP address of the user if the system observes fake review send by the same IP Address many a times it will inform the admin to remove that review from the system. This system uses data mining methodology. This system helps the user to find out correct review of the product.
- Effective fake review monitoring requires ongoing attention and analysis of product reviews, as well as the ability to quickly and accurately identify and remove fraudulent reviews. By implementing a system for fake review monitoring, businesses can help ensure that their product reviews accurately reflect the quality and reputation of their products, which can lead to increased customer trust and sales.

3. SYSTEM ANALYSIS:

3.1 Existing system:

- **T**he system in which you can find the original feedback and rating related to a specific product .Then, it is the source of satisfaction and reliability for you. In the proposed technique, the reviews related to a product for which the URL is given are extracted. After it, the system finds the fake reviews and finally by analyzing these reviews system finds the original reviews of the product. Previous researches detect fake reviews using different approaches including identification address, opinion mining and sentiment analysis, machine-learning approach.
- There are many researches available that detect the fake reviews related to English but no work is done so far that detect fake reviews for Urdu (highly spoken language in Asia) and Roman Urdu. Therefore, we have proposed Fake Product Review Monitoring and Removal System (FaRMS) in which a customer can get the best possible item from the online store in a short time and with the original reviews associated with that product.
- This system gives you the original words of people related to the product with genuine reviews. Some popular products can get hundreds of reviews at some large merchant sites and FaRMS gives you the promising reviews by filtering fake reviews and then you can decide whether you want to buy or not.

Disadvantages

- Fake reviews detection for the Yelp is worked with the intention to filter the fake reviews from the original reviews as this is becoming the need of the hour. The proposed system classifier takes the reviews text and other information and produces the output whether the reviews are reliable or not.
- The data set which is used in this project is taken for the Yelp.com which is firstly used by the Rayana and Akoglu. They use 16282 reviews and split these into 0.7 training set, 0.2 dev set and 0.1 test set. Extracting predictive features from the reviews is the most challenging part of the project.

- Basically they extract two types of features: review-centric feature and reviewer-centric features. Firstly they count the percentages of each unigram and bigram tokens for fake and non-fake reviews. They then take out the top 100 unigrams and bigrams that have the most different percentages in fake and non-fake reviews.
- The second approach leads to the better performance because it processed all the unigrams and bigrams. They tested multiple algorithms of machine learning but by using the Neural Networks they achieved the highest accuracy of about 81.92%. This system is good in finding the fake reviews but still there is a need to improve the accuracy in filtering the reviews.

3.2 Proposed system:

- A technique to ranks the product is worked to present a product ranking model that applies weights to product review factors to calculate a product ranking score. In this proposed system, the sentences that are not related with the quality of a product such as customer service or sentence related to the.
- In this paper the pre processing is done by Support Vector Machine (SVM). First of all it removes the comments which neither is nor related with the quality of the product. Second stage describes the weights of the reviews based on the votes. The final stage calculates the overall ranking of the product.
- The ranking score is calculated by the relevance of the review with quality of the product, review content, and posting date of the review. They use 10-fold cross validation on the training set. In the evaluation process they use two measures to quantify effectiveness of the ranking model which are as following: correlation between the ranking method and the Amazon's rank and second is the Mean Average Precision (MAP), which is a very commonly used technique for evaluating ranking accuracy.
- As this system is finding the fake reviews by using the only two properties of the reviews but as per the future work describes in the paper more properties can be used to find out the fake reviews more accurately.Spam reviews detection by using Temporal Pattern Discovery is proposed to observe the reviews related to the normal reviewers arrival pattern and fake reviewers arrival pattern and they observe that the normal reviewer arrival pattern is stable and uncorrelated to their rating pattern temporally.

- On the opposite side the spam attacks are usually bursty and either positively or negatively correlated with the rating pattern. The data set which they have taken is snapshot of a review Website on October 6, 2010. It includes 408469 reviews which are written by 343629 reviewers, which are written for 25034 stores of a Website.
- For each review they collect the following information like rating, postdate and whether it is a Spammer Review (SR) or not. In the evaluation process they select 53 stores each of which has more than 1000 reviews.
- Human evaluators make decision about the stores to be SR spam attack or not if two or more evaluators declared a store as SR spam attack than system considers the store a dishonest in it's selling. Out of 53 stores 34 are suspicious one and the remaining are normal ones. Out of 34 stores 22 stores have at least two votes for being suspicious.
- The recall related to the system is 75.86% which shows that the system detects most of the stores having SR spam attack. The precision related to the proposed approach is 61.11%. This proposed system is good in terms of the training of their model for finding the relation as the model is trained by using the large number of reviews contains in their dataset.

Advantages:

- The previous works detect fake reviews by using IP address, opinion mining, sentiment analysis, and some of them uses machine learning approaches. In some approaches dataset is very small and other uses few properties related to the reviews to find the fake reviews.
- In the proposed system the large dataset of English reviews is used to train the model. In this way, the system can find the hidden patterns in the reviews more efficiently. The accuracy which is achieved by using the proposed technique is greater than the accuracy of the previous systems in terms of the English reviews. This system is also worked to detect the fake reviews of Urdu and Roman Urdu reviews.
- Fake review monitoring system focuses on detecting spam and fake reviews by using sentimental analysis removes the reviews which have curse and vulgar words. In the proposed system web crawler is used to scrapped the data on the Website.
- In the pre processing, the data is converted into the required format and then the fake reviews are removed from the mixture of original and spam reviews. Fake reviews are detected by the Fake Review Detector.

4 Hardware requirements:

4.1 System requirements:

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer) **system requirements** and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements. A second meaning of the term of system requirements, is a generalisation of this first definition, giving the requirements to be met in the design of a system or sub-system.

4.2 Recommended system requirements:

Often manufacturers of games will provide the consumer with a set of requirements that are different from those that are needed to run a software. These requirements are usually called the recommended requirements. These requirements are almost always of a significantly higher level than the minimum requirements, and represent the ideal situation in which to run the software. Generally speaking, this is a better guideline than minimum system requirements in order to have a fully usable and enjoyable experience with that software.

4.3 Hardware requirements:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

4.4 Architecture

All computer operating systems are designed for a particular computer architecture. Most software applications are limited to particular operating systems running on particular architectures. Although architecture-independent operating systems and applications exist, most need to be recompiled to run on a new architecture. See also a list of common operating systems and their supporting architectures.

4.5 Processing power:

The power of the central processing unit (CPU) is a fundamental system requirement for any software. Most software running on x86 architecture define processing power as the model and the clock speed of the CPU. Many other features of a CPU that influence its speed and power, like bus speed, cache, and MIPS are often ignored. This definition of power is often erroneous, as AMD Athlon and Intel Pentium CPUs at similar clock speed often have different throughput speeds. Intel Pentium CPUs have enjoyed a considerable degree of popularity, and are often mentioned in this category.

4.6 Memory

All software, when run, resides in the random access memory (RAM) of a computer. Memory requirements are defined after considering demands of the application, operating system, supporting software and files, and other running processes. Optimal performance of other unrelated software running on a multi-tasking computer system is also considered when defining this requirement.

4.7 Secondary storage

Data storage device requirements vary, depending on the size of software installation, temporary files created and maintained while installing or running the software, and possible use of swap space (if RAM is insufficient).

4.8 Display adapter:

Software requiring a better than average computer graphics display, like graphics editors and high-end games, often define high-end display adapters in the system requirements.

4.9 Peripherals:

Some software applications need to make extensive and/or special use of some peripherals, demanding the higher performance or functionality of such peripherals. Such peripherals include CD-ROM drives, keyboards, pointing devices, network devices, etc.

4.10 Software requirements:

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

4.11 Platform:

A computing platform describes some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer's architecture, operating system, or programming languages and their runtime libraries. Operating system is one of the requirements mentioned when defining system requirements (software). Software may not be compatible with different versions of same line of operating systems, although some measure of backward compatibility is often maintained. For example, most software designed for Microsoft Windows XP does not run on Microsoft Windows 98, although the converse is not always true. Similarly, software designed using newer features of Linux Kernel v2.6 generally does not run or compile properly (or at all) on Linux distributions using Kernel v2.2 or v2.4.

4.12 APIs and drivers:

Software making extensive use of special hardware devices, like high-end display adapters, needs special API or newer device drivers. A good example is DirectX, which is a collection of APIs for handling tasks related to multimedia, especially game programming, on Microsoft platforms.

4.13 Web browser:

Most web applications and software depend heavily on web technologies to make use of the default browser installed on the system. Microsoft Internet Explorer is a frequent choice of software running on Microsoft Windows, which makes use of ActiveX controls, despite their vulnerabilities.

4.14 Other requirements:

Some software also has other requirements for proper performance. Internet connection (type and speed) and resolution of the display screen are notable examples.

4.15 Examples:

Following are a few examples of system requirement definitions for popular PC games and trend of ever-increasing resource needs:

For instance, while StarCraft (1998) requires:

System requirements [hide]

	Requirements
Windows	
Operating system	Windows 95 or NT or superior
CPU	Pentium processor at 90 MHz or higher
Memory	16 MB RAM
Free space	80 MB available
Media	CD-ROM, 2x or higher
Graphics hardware	DirectX 3.0 or higher

System requirements [hide]

	Requirements
Windows	
Operating system	Windows 2000/XP
CPU	Pentium 4 1.5 GHz or Athlon XP 1500+ processor or higher
Memory	384 MB RAM
Free space	2.2 GB free space
Media	8x Speed CD-ROM
Graphics hardware	3D Hardware Accelerator - 64MB of memory minimum DirectX 9.0b
Sound hardware	DirectX 9.0b compatible 16-bit sound card

System requirements [hide]

	Requirements
Windows	
Operating system	Windows Windows XP SP3, Windows Vista SP2, Windows 7
CPU	Core 2 Duo or Athlon X2 at 2.4 GHz
Memory	2 GB RAM
Free space	8 GB of free space, 23.8 GB + 1 GB Swap File space
Graphics hardware	DirectX 9.0c compatible video card. 3D Hardware Accelerator - 256MB of memory minimum
Sound hardware	DirectX 9.0c compatible sound card

System requirements [hide]

	Requirements
Windows	
Operating system	Windows 8.1 64 Bit, Windows 8 64 Bit, Windows 7 64 Bit Service Pack 1, Windows Vista 64 Bit Service Pack 2
CPU	Core 2 Quad Q6600 at 2.4 GHz or AMD Phenom 9850 at 2.5 GHz
Memory	4 GB RAM
Free space	65 GB of free space
Graphics hardware	DirectX 10-compatible GPU: GeForce 9800GT 1GB or ATI Radeon HD 4870 1GB
Sound hardware	DirectX 10 compatible sound card

5 Software requirements:

5.1Python:

- Python is an interpreter, high-level Object and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected.
- It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.
- Guido van Rossum began working on Python in the late 1980's, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.1.[31] Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting and was discontinued with version 2.7.18 in 2020.
- Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3. Python consistently ranks as one of the most popular programming languages

5.2 Design philosophy and features:

- Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)).
- Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

- Python's design offers some support for functional programming in the Lisp tradition. It has filter, map, and reduce functions; list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications.

- Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach. Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology.
- In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one— and preferably only one —obvious way to do it" design philosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, writes that "To describe something as 'clever' is *not* considered a compliment in the Python culture."
- Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler.

- Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter. An important goal of Python's developers is keeping it fun to use.
- This is reflected in the language's name—a tribute to the British comedy group Monty Python—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.
- A common neologism in the Python community is *pythonic*, which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python's minimalist philosophy and emphasis on readability.
- In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called unpythonic. Users and admirers of Python, especially those considered knowledgeable or experienced, are often referred to as *Pythonistas*

5.3 Libraries:

- Python's large standard library, commonly cited as one of its greatest strengths,[113] provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported.
- It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing.
- Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation `wsgiref` follows PEP 333), but most modules are not.
- They are specified by their code, internal documentation, and test suites. However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

- ➢ As of March 2021, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 290,000 packages with a wide range of functionality, including:

- Automation
- Data analytics
- Databases
- Documentation
- Graphical user interfaces
- Image processing
- Machine learning
- Mobile App
- Multimedia
- Computer Networking
- Scientific computing
- System administration
- Test frameworks
- Text processing
- Web frameworks
- Web scraping

5.4 Development environments:

See also: Comparison of integrated development environments § Python

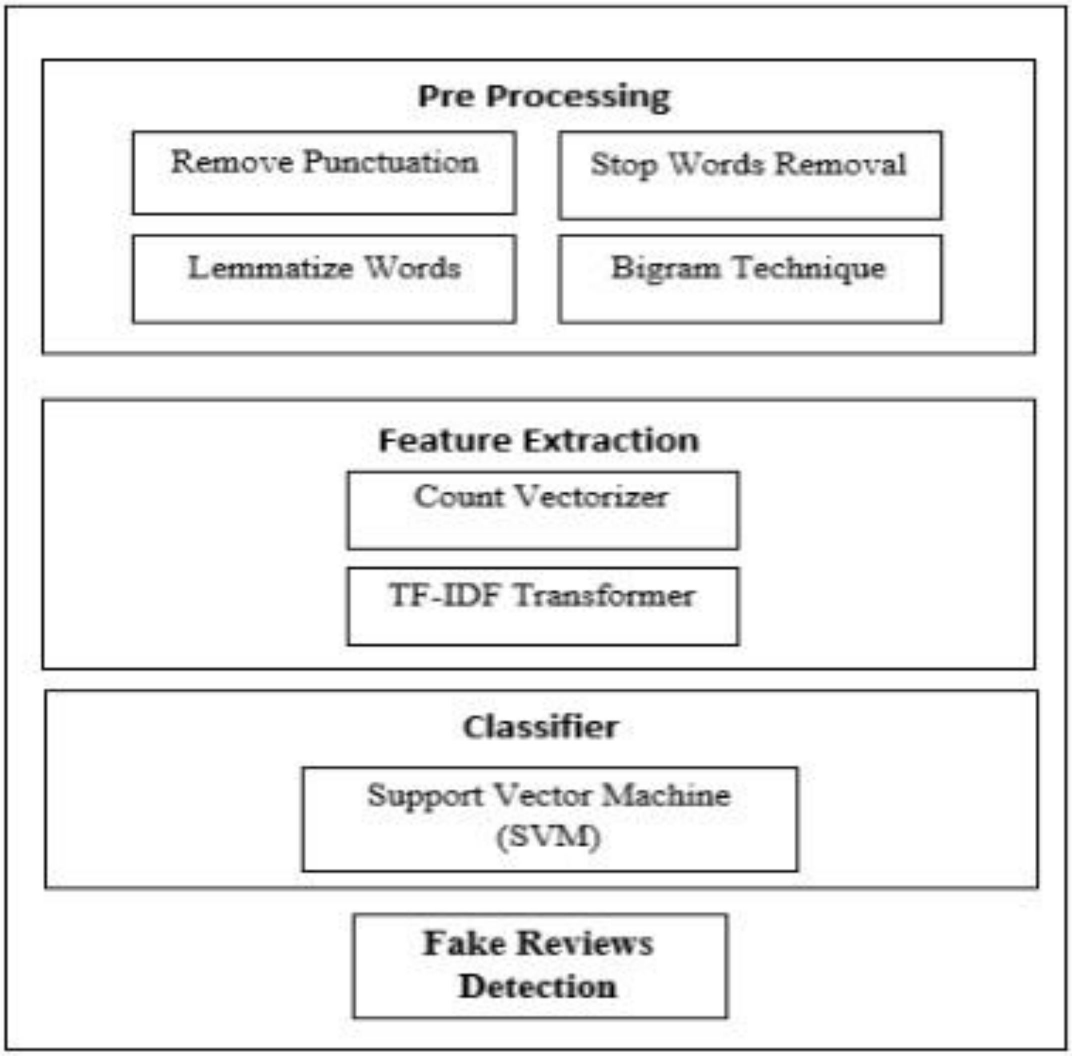
- ➢ Most Python implementations (including CPython) include a read–eval–print loop (REPL), permitting them to function as a command line interpreter for which the user enters statements sequentially and receives results immediately.
- ➢ Other shells, including IDLE and IPython, add further abilities such as improved auto-completion, session state retention and syntax highlighting.

6. System Architecture:

6.1 System Architecture:

- An intelligent interface for fake product review monitoring and removal would require a robust system architecture to effectively detect and remove fake reviews. The following is a proposed system architecture for such an interface:
- Data Collection: The system should collect data from various sources such as ecommerce websites, social media platforms, and other sources where product reviews are posted.Data Pre-processing: The collected data should be pre-processed to remove irrelevant information and extract important features such as text, rating, and sentiment.
- Machine Learning Model: A machine learning model such as a natural language processing (NLP) model or a deep learning model should be trained on the preprocessed data to detect fake reviews.
- User Interface: The system should have a user interface where users can interact with the system to view detected fake reviews and take necessary actions to remove them.
- Review Database: A review database should be created to store all the reviews collected from various sources.
- Review Analysis: The system should analyze each review and provide a score indicating the probability of it being a fake review.
- Review Removal: If a review is identified as fake, the system should automatically remove it from the website or platform where it was posted.Alert Generation: The system should generate alerts to notify the users of the system about the detection and removal of fake reviews.
- Feedback Mechanism: The system should have a feedback mechanism where users can provide feedback on the accuracy of the system's fake review detection and removal.
- Continuous Improvement: The system should continuously improve its accuracy and effectiveness by learning from the feedback provided by users and by analyzing new data. This can be achieved by retraining the machine learning model with new data and improving the pre-processing and analysis techniques used by the system.
- Overall, this proposed system architecture for an intelligent interface for fake product review monitoring and removal would require integration of various technologies and processes to effectively detect and remove fake reviews.

Fig : 6.1 System Architecture .



7. Module & Description:

7.1 Modules:

- Data set module
- Data pre processing module
- Feature extraction module
- Training module
- Classification module
- Output module

7.2 Module Description:

7.2.1Data set module:

- A data set (or dataset) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. Data sets can also consist of a collection of documents or files.
- In the open data discipline, data set is the unit to measure the information released in a public open data repository. The European Open Data portal aggregates more than half a million data sets. In this field other definitions have been proposed, but currently there is not an official one. Some other issues (real-time data sources, nonrelational data sets, etc.) increases the difficulty to reach a consensus about it.

7.2.2 Data pre processing module:

- The pre-processing module for an intelligent interface for fake product review monitoring and removal would involve several steps to ensure the accuracy and effectiveness of the system. Here are some potential steps that could be included in the pre-processing module:
- Feature selection: The pre-processing module may need to select the most relevant features for training the machine learning algorithms to improve the accuracy and efficiency of the system.
- Overall, the pre-processing module is a critical component of an intelligent interface for fake product review monitoring and removal, as it ensures that the system is trained on high-quality data and that the machine learning algorithms can effectively detect and remove fake reviews.

7.2.3 Feature extraction module:

- Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process.
- Feature extraction is the name for methods that select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set.
- The process of feature extraction is useful when you need to reduce the number of resources needed for processing without losing important or relevant information.
- Also, the reduction of the data and the machine's efforts in building variable combinations (features) facilitate the speed of learning and generalization steps in the machine learning process.

7.2.4Training module:

- A training module is one structured section of a course. The content within a training module should be designed, and created, to support the learner's intake and retention of the information it contains. Grouping training modules together is used to create step-by-step learning.

- Grouping training modules together is used to create step-by-step learning. Each module forms one part of an overall topic, enabling learners to gradually progress through a course, module by module, to reach their training goals. It's a tactic that makes training delivered through a learning management system more digestible.

7.2.5 Classification module:

- Classification is a machine learning method that uses data to determine the category, type, or class of an item or row of data. For example, you can use classification to:

- Classify email filters as spam, junk, or good.
- Determine whether a patient's lab sample is cancerous.
- Categorize customers by their propensity to respond to a sales campaign.
- Identify sentiment as positive or negative.

- Classification tasks are frequently organized by whether a classification is binary (either A or B) or multiclass (multiple categories that can be predicted by using a single model).

7.2.6 Output module:

- In signal communication within the output module, the external signal line must be isolated from the body. Optical-coupled photo couplers provide simple and reliable isolation.
- A transistor output photo coupler or an IC output photo coupler which has high-speed communication capability is recommended. In addition, it is essential to observe the safety standards established in each country.
- Toshiba Photo couplers have been approved by UL1577, VDE: EN60747-5-5, EN62368-1, and other organizations.
- The output stage also contributes to lower power dissipation through MOSFET of low on-resistance and higher output through the use of transistor arrays with high withstand voltage and large current.

8. Implementations:

See also: List of Python software § Python implementations

8.1Reference implementation:

- CPython is the reference implementation of Python. It is written in C, meeting the C89 standard with several select C99 features (with later C versions out, it's considered outdated; CPython includes its own C extensions, but third-party extensions are not limited to older C versions, can e.g. be implemented with C11 or C++[120]).
- It compiles Python programs into an intermediate bytecode which is then executed by its virtual machine. CPython is distributed with a large standard library written in a mixture of C and native Python.
- It is available for many platforms, including Windows (starting with Python 3.9, the Python installer deliberately fails to install on Windows 7 and 8; Windows XP was supported until Python 3.5) and most modern Unix-like systems, including macOS (and Apple M1 Macs, since Python 3.9.1, with experimental installer) and unofficial support for e.g. VMS. Platform portability was one of its earliest priorities, during the Python 1 and 2 time-frame, even OS/2 and Solaris were supported; support has since been dropped for a lot of platforms.

8.2 Other implementations:

- PyPy is a fast, compliant interpreter of Python 2.7 and 3.6. Its just-in-time compiler brings a significant speed improvement over CPython but several libraries written in C cannot be used with it.
- Stackless Python is a significant fork of CPython that implements microthreads; it does not use the call stack in the same way, thus allowing massively concurrent programs. PyPy also has a stackless version.
- MicroPython and CircuitPython are Python 3 variants optimized for microcontrollers, including Lego Mindstorms EV3.
- Pyston is a variant of the Python runtime that uses just-in-time compilation to speed up the execution of Python programs.

8.3 Unsupported implementations:

Other just-in-time Python compilers have been developed, but are now unsupported:

- Google began a project named Unladen Swallow in 2009, with the aim of speeding up the Python interpreter fivefold by using the LLVM, and of improving its multithreading ability to scale to thousands of cores,[135] while ordinary implementations suffer from the global interpreter lock.
- Psyco is a discontinued just-in-time specializing compiler (which didn't support Python 2.7 or later) that integrates with CPython and transforms bytecode to machine code at runtime. The emitted code is specialized for certain data types and is faster than the standard Python code.

- PyS60 was a Python 2 interpreter for Series 60 mobile phones released by Nokia in 2005. It implemented many of the modules from the standard library and some additional modules for integrating with the Symbian operating system. The Nokia N900 also supports Python with GTK widget libraries, enabling programs to be written and run on the target device.

`9.Cross-compilers to other languages:

9.1 Cross-compilers to other languages:

There are several compilers to high-level object languages, with either unrestricted Python, a restricted subset of Python, or a language similar to Python as the source language:

- Cython compiles (a superset of) Python 2.7 to C (while the resulting code is also usable with Python 3 and also e.g. C++).
- Nuitka compiles Python into C++.
- Pythran compiles a subset of Python 3 to C++.
- Pyrex (latest release in 2010) and Shed Skin (latest release in 2013) compile to C and C++ respectively.
- Google's Grumpy (latest release in 2017) transpires Python 2 to Go.
- Iron Python (now abandoned by Microsoft) allows running Python 2.7 programs on the .NET Common Language Runtime.
- Python compiles Python 2.7 to Java bytecode, allowing the use of the Java libraries from a Python program.
- Myrdal is a Python-based hardware description language (HDL), that converts MyHDL code to Verilog or VHDL code.
- Numba uses LLVM to compile a subset of Python to machine code.
- Brython, Transcrypt and Pyjs (latest release in 2012) compile Python to JavaScript.
- RPython can be compiled to C, and is used to build the PyPy interpreter of Python.

9.2 Performance:

A performance comparison of various Python implementations on a non-numerical (combinatorial) workload was presented at EuroSciPy '13. Python's performance compared to other programming languages is also benchmarked by The Computer Language Benchmarks Game.

9.3 Development:

- Python's development is conducted largely through the *Python Enhancement Proposal* (PEP) process, the primary mechanism for proposing major new features, collecting community input on issues and documenting Python design decisions. Python coding style is covered in PEP 8. Outstanding PEPs are reviewed and commented on by the Python community and the steering council.

- Enhancement of the language corresponds with development of the CPython reference implementation. The mailing list python-dev is the primary forum for the language's development. Specific issues are discussed in the Roundup bug tracker hosted at bugs.python.org. Development originally took place on a self-hosted source-code repository running Mercurial, until Python moved to GitHub in January 2017. CPython's public releases come in three types, distinguished by which part of the version number is incremented:

- Backward-incompatible versions, where code is expected to break and needs to be manually ported. The first part of the version number is incremented. These releases happen infrequently—version 3.0 was released 8 years after 2.0.
- Major or "feature" releases, occurred about every 18 months but with the adoption of a yearly release cadence starting with Python 3.9 are expected to happen once a year. They are largely compatible but introduce new features. The second part of the version number is incremented. Each major version is supported by bugfixes for several years after its release.
- Bugfix releases, which introduce no new features, occur about every 3 months and are made when a sufficient number of bugs have been fixed upstream since the last release. Security vulnerabilities are also patched in these releases. The third and final part of the version number is incremented.
- Many alpha, beta, and release-candidates are also released as previews and for testing before final releases. Although there is a rough schedule for each release, they are often delayed if the code is not ready. Python's development team monitors the state of the code by running the large unit test suite during development.
- The major academic conference on Python is PyCon. There are also special Python mentoring programmes, such as Pyladies. Pythons 3.10 deprecates wstr (to be removed in Python 3.12; meaning Python extensions need to be modified by then),[159] and also plans to add pattern matching to the language.

9.5 Naming:

- Python's name is derived from the British comedy group Monty Python, whom Python creator Guido van Rossum enjoyed while developing the language. Monty Python references appear frequently in Python code and culture;[161] for example, the metasyntactic variables often used in Python literature are *spam* and *eggs* instead of the traditional *foo* and *bar*. The official Python documentation also contains various references to Monty Python routines.
- The prefix *Py-* is used to show that something is related to Python. Examples of the use of this prefix in names of Python applications or libraries include Pygame, a binding of SDL to Python (commonly used to create games); PyQt and PyGTK, which bind Qt and GTK to Python respectively; and PyPy, a Python implementation originally written in Python.

9.6 Uses:

- Since 2003, Python has consistently ranked in the top ten most popular programming languages in the TIOBE Programming Community Index where, as of February 2021, it is the third most popular language (behind Java, and C). It was selected Programming Language of the Year (for "the highest rise in ratings in a year") in 2007, 2010, 2018, and 2020 (the only language to do so four times .

- An empirical study found that scripting languages, such as Python, are more productive than conventional languages, such as C and Java, for programming problems involving string manipulation and search in a dictionary, and determined that memory consumption was often "better than Java and not much worse than C or C++".Large organizations that use Python include Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify and some smaller entities like ILM and ITA.
- The social news networking site Reddit was written mostly in Python. Python can serve as a scripting language for web applications, e.g., via mod_wsgi for the Apache web server. With Web Server Gateway Interface, a standard API has evolved to facilitate these applications.
- Web frameworks like Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle and Zope support developers in the design and maintenance of complex applications. Pyjs and IronPython can be used to develop the client-side of Ajax-based applications.
- SQLAlchemy can be used as a data mapper to a relational database. Twisted is a framework to program communications between computers, and is used (for example) by Dropbox.
- Libraries such as NumPy, SciPy and Matplotlib allow the effective use of Python in scientific computing with specialized libraries such as Biopython and Astropy providing domain-specific functionality. SageMath is a mathematical software with a notebook interface programmable in Python: its library covers many aspects of mathematics, including algebra, combinatorics, numerical mathematics, number theory, and calculus. OpenCV has python bindings with a rich set of features for computer vision and image processing.
- Python is commonly used in artificial intelligence projects and machine learning projects with the help of libraries like TensorFlow, Keras, Pytorch and Scikit-learn. As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing
- Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like FreeCAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro, and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in ArcGIS.

- It has also been used in several video games, and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go.[194]
- Many operating systems include Python as a standard component. It ships with most Linux distributions, AmigaOS 4 (using Python 2.7), FreeBSD (as a package), NetBSD, OpenBSD (as a package) and macOS and can be used from the command line (terminal).
- Many Linux distributions use installers written in Python: Ubuntu uses the Ubiquity installer, while Red Hat Linux and Fedora use the Anaconda installer. Gentoo Linux uses Python in its package management system, Portage. Python is used extensively in the information security industry, including in exploit development.
- Most of the Sugar software for the One Laptop per Child XO, now developed at Sugar Labs, is written in Python. The Raspberry Pi single-board computer project has adopted Python as its main user-programming language. LibreOffice includes Python, and intends to replace Java with Python. Its Python Scripting Provider is a core feature since Version 4.0 from 7 February 2013.

9.7 Languages influenced by Python:

Python's design and philosophy have influenced many other programming languages:

- Boo uses indentation, a similar syntax, and a similar object model.
- Cobra uses indentation and a similar syntax, and its *Acknowledgements* document lists Python first among languages that influenced it.
- CoffeeScript, a programming language that cross-compiles to JavaScript, has Python-inspired syntax.
- ECMAScript/JavaScript borrowed iterators and generators from Python.
- GDScript, a scripting language very similar to Python, built-in to the Godot game engine.
- Go is designed for the "speed of working in a dynamic language like Python" and shares the same syntax for slicing arrays.
- Groovy was motivated by the desire to bring the Python design philosophy to Java.
- Julia was designed to be "as usable for general programming as Python".
- Nim uses indentation and similar syntax.
- Ruby's creator, Yukihiro Matsumoto, has said: "I wanted a scripting language that was more powerful than Perl, and more object-oriented than Python. That's why I decided to design my own language."
- Swift, a programming language developed by Apple, has some Python-inspired syntax.

10 Keras:

- Keras is an open source neural network library written in Python. It is capable of running on top of Tensor Flow, Microsoft Cognitive Toolkit, or Theano. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.
- It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System),[2] and its primary author and maintainer is François Chollet, a Google engineer.In 2017, Google's TensorFlow team decided to support Keras in TensorFlow's core library.
- Chollet explained that Keras was conceived to be an interface rather than a standalone machine-learning framework. It offers a higher-level, more intuitive set of abstractions that make it easy to develop deep learning models regardless of the computational backend used. Microsoft added a CNTK backend to Keras as well, available as of CNTK v2.0.

10.1 Key Features:

- Commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier.
- The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel.

10.2 Users view of Keras:

- Keras allows users to productize deep models on smart phones (iOS and Android), on the web, or on the Java Virtual Machine.It also allows use of distributed training of deep learning models on clusters of Graphics Processing Units (GPU) and Tensor processing units (TPU)Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano.
- It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

10.3 Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).Supports convolutional networks and recurrent networks, as well as combinations of the two.Runs seamlessly on CPU and GPU.

10.4 User friendliness.

- Keras is an API designed for human beings, not machines. It puts user experience front and center. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error.

10.5 Modularity.

- A model is understood as a sequence or a graph of standalone, fully-configurable modules that can be plugged together with as few restrictions as possible. In particular, neural layers, cost functions, optimizers, initialization schemes, activation functions, regularization schemes are all standalone modules that you can combine to create new models.

10.6 Easy extensibility.

- New modules are simple to add (as new classes and functions), and existing modules provide ample examples. To be able to easily create new modules allows for total expressiveness, making Keras suitable for advanced research.

10.7 Work with Python.

- No separate models configuration files in a declarative format. Models are described in Python code, which is compact, easier to debug, and allows for ease of extensibility.

10.8 CoCo Dataset

- Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos.[1] Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.

10.9 Uses:

- It is widely used in computer vision tasks such as image annotation, activity recognition, face detection, face recognition, video object co-segmentation. It is also used in tracking objects, for example tracking a ball during a football match, tracking movement of a cricket bat, or tracking a person in a video.

10.10 Concept:

- Every object class has its own special features that helps in classifying the class for example all circles are round. Object class detection uses these special features. For example, when looking for circles, objects that are at a particular distance from a point (i.e. the centre) are sought.
- Similarly, when looking for squares, objects that are perpendicular at corners and have equal side lengths are needed. A similar approach is used for face identification where eyes, nose, and lips can be found and features like skin colour and distance between eyes can be found.

10.11 Methods:

- Methods for object detection generally fall into either neural network-based or non-neural approaches. For non-neural approaches, it becomes necessary to first define features using one of the methods below, then using a technique such as support vector machine (SVM) to do the classification.
- On the other hand, neural techniques are able to do end-to-end object detection without specifically defining features, and are typically based on convolutional neural networks (CNN).

10.12 Non-neural approaches:

- Viola–Jones object detection framework based on Haar features
- Scale-invariant feature transform (SIFT)
- Histogram of oriented gradients (HOG) features.

11 . source coad:

```
import csv
import sqlite3
import numpy as np
import pandas as pd
from nltk.stem.porter import *
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import string
from PIL import Image
from os import path
import matplotlib.pyplot as plt
import matplotlib as mpl
import pickle

from sklearn.svm import LinearSVC
from sklearn.feature_extraction.text import TfidfVectorizer

from wordcloud import WordCloud, STOPWORDS
from collections import Counter

from nltk.stem import SnowballStemmer
from nltk.stem import WordNetLemmatizer
stemmer = SnowballStemmer('english')
wordnet_lemmatizer = WordNetLemmatizer()

##############################In/out#############################
def open_file(csvfile):
    reader = pd.read_csv(csvfile)
    return reader

def output_file(df,string):
    df.to_csv(string, index = False)

##############################Word Cloud & Feature
Extraction#############################

def text_process(data):
    '''
    Takes in a df in format of [text,stars] performs the following:
    1. Lower capital letters
    2. Remove all punctuation
    3. Remove all stopwords
    4. Reduce words to their word stem
    5. Return a list of words
```

```
    '''
    for i in range(len(data)):
        line = data[i]
        line = line.lower()  # lower case
        translation = str.maketrans("", "", string.punctuation);
        line = line.translate(translation)
        split = word_tokenize(line)
        # filter out any tokens not containing letters (e.g., numeric tokens, raw punctuation)
        filtered = []
        for token in split:
            if re.search('[a-zA-Z]', token):
                filtered.append(token)
        word = [i for i in filtered if i not in stopwords.words('english')]

        d = [stemmer.stem(word) for word in word]
        d = [wordnet_lemmatizer.lemmatize(word) for word in d]
        data[i] = d
    return data


def top_words(business_id,review_ml ):
    train = review_ml[review_ml['business_id'] == business_id][review_ml['True(1)/Deceptive(0)']
== 'True']
    text = list(train['Review'])  # text
    text = text_process(text)
    text = sum(text, [])

    counts = Counter(text)
    wordcloud = WordCloud(
        background_color='white',
        max_words=100,
        max_font_size=50,
        min_font_size=10,
        random_state=40,

    ).fit_words(counts)
    fig = plt.figure(1)
    plt.imshow(wordcloud)
    plt.axis('off')  # remove axis
    plt.show()


def change_label(x):
    for i in range(len(x)):
        if x[i] >= 3.0:  # good review: stars >=3.0
            x[i] = 1
        else:  # bad review: stars 3.0
```

```
        x[i] = 0
    return x


def bigram(business_id, review_ml):
    # only use true review

    train0 = review_ml[review_ml['business_id'] == business_id]
    train = train0[train0['True(1)/Deceptive(0)'] == 'True']
    #print(train.head())
    #train_data = list(train['Review'])  # text
    label = list(train['Stars'])  # ratings
    #print(label)
    train_label = change_label(label)
    #print(train_label)

    # TfidfVectorizer Transform
    transformer = TfidfVectorizer(stop_words='english',
                                  ngram_range=(2, 2))  # "ignore terms that appear in less than 1% of the
documents".
    #print(transformer)

    cvectorizer = transformer.fit(train['Review'])
    #print(cvectorizer)
    transformed = cvectorizer.transform(train['Review'])
    #print(transformed)

    # SVM regression
    clf = LinearSVC()
    clf.fit(transformed, train_label)
    coefficients = clf.coef_.ravel()
    #print(coefficients)
    pos_coefficients = np.argsort(coefficients)[-10:]
    neg_coefficients = np.argsort(coefficients)[:10]
    combine = np.hstack([neg_coefficients, pos_coefficients])
    #print("combine:, ",combine)
    #print("coefficients[combine]: ", coefficients[combine])

    plt.figure(figsize=(7, 4))
    #print("fisnish 1")

    colors = ['red' if i < 0 else 'blue' for i in coefficients[combine]]
    #print("finish 2")

    plt.bar(np.arange(len(coefficients[combine])), coefficients[combine], color=colors)
    #print("finish 3")
```

```
    feature_names = np.array(cvectorizer.get_feature_names())
    #print("finish 4")
    plt.title('why the restaurant is rated as bad or good ', fontsize=15)
    #print("finish 5")

    plt.xticks(np.arange(0, 2 * 10), feature_names[combine], rotation=40, ha='right')
    #print("finish 6")

    plt.show()
    #print("finish 7")


############################helper function############################
def load_database_data(c, zipcode, business_name_input):
    c.execute('''
            SELECT b_id,r.review, r.r_stars
            FROM business, review_fact_table r
            WHERE postal_code = ? AND name = ? AND r.business_id = b_id''', (zipcode,
business_name_input,))
    dataframe = pd.DataFrame(data=c.fetchall(), columns=['business_id', 'review', 'rating'])
    return dataframe
def select_data(c, zipcode, business_name):
    c.execute('''
                SELECT DISTINCT(b_id)
                FROM business, review_fact_table r
                WHERE postal_code = ? AND name = ? AND r.business_id = b_id''', (zipcode,
business_name,))
    single_df = pd.DataFrame(data=c.fetchall(), columns=['business_id'])
    return single_df['business_id'][0]

def fake_ratio(predict, single):
    # Load fake results
    predicted_fake = predict
    # reviews that has only that business id
    reviews = predicted_fake[predicted_fake['business_id'] == single]
    n = reviews.shape[0]
    # print(n)
    fake = reviews.groupby('True(1)/Deceptive(0)').count()['Review'][0]
    # print(fake)
    fake_percentage = fake / n
    # print(fake_percentage)
    return fake_percentage

##############################main#####################################
def main():
    #open states and income raw data
    zipcode = input("zipcode:")
```

```
    business_name = input("restaurant name:")
    print(zipcode,business_name)

    conn = sqlite3.connect('yelp.db')
    c = conn.cursor()
    predicted_fake = open_file('data/predicted_review.csv')
    # find the business id
    single = select_data(c, zipcode, business_name)
    # print(single)
    fake_review_ratio = fake_ratio(predicted_fake,single)
    print(fake_review_ratio)
    #top_words(single, predicted_fake)
    bigram(single, predicted_fake)

#if __name__=="__main__":
    #main()
main()
```

12 Conclusion & Reference :

12.1 Conclusion:

- In the proposed work, dataset is developed that contains Urdu and Roman Urdu reviews. It is difficult to detect fake reviews by yourself. So, n-gram approach is used to detect fake reviews for multiple languages.
- It is observed that the text categorization with SVM classifier is best approach for the detection of fake reviews. Now a days, as the technology is growing day by day and there are so many Websites and applications available in the online market by which seller can sell their products and on that products there are millions of reviews available.
- There are some organizations posting fake reviews for the products of the seller in order to increase or decrease the rating of the products. Therefore, the system is proposed that detects the fake reviews in multiple languages including English, Urdu, and Roman Urdu, classify the reviews in genuine.
- It helps the user to get the products from Daraz, Flipkart and Amazon with the satisfaction of their mind and pay for the good quality product. As, there are a lot of e-commerce stores like AliExpress and Alibaba which have reviews of multiple languages. It would be great if the proposed system finds the way, to process and filter the reviews for other multiple languages.

12.2 References:

[1] A. Sinha, N. Arora, S. Singh, M. Cheema, and A. Nazir, “Fake Product Review Monitoring Using Opinion Mining,” vol. 119, no. 12, pp. 13203–13209, 2018.

[2] Torbet, Georgina. “U.S. Customers Spent over $6 Billion on Black Friday Purchases.” Digital Trends, Digital Trends, 25 Nov. 2018, www.digitaltrends.com/web/shopping-totals-black-friday/

[3] Sterling, Greg. “Study Finds 61 Percent of Electronics Reviews on Amazon Are 'Fake'.” Marketing Land, 19 Dec. 2018, marketingland.com/study-finds-61-percent-of electronics-reviews-on-amazon-are-fake 254055.

[4] K. Khan, W. Khan,A. Rehman, A. Khan, Asfandyar. Khan, A. Ullah Khan, B. Saqia, "Urdu Sentiment Analysis," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 9, 2018.

[5] A. Mukherjee, B. Liu, and N. Glance, "Spotting Fake Reviewer Groups in Consumer Reviews," 2012.

[6] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, “What Yelp Fake Review Filter Might Be Doing?,” Aaai, pp. 409–418, 2013

[7] Z. Wang, Y. Zhang, and T. Qian, “Fake Review Detection on Yelp Dataset and features,” pp.

[8] S. Xie, G. Wang, S. Lin, and P. S. Yu, “Review spam detection via temporal pattern discovery,” p. 823, 2012.

[9] C. Paper, “Mining millions of reviews: A technique to rank products based on importance of reviews Mining Millions of Reviews: A Technique to Rank Products Based on Importance of Reviews,” no. November, 2015.

[10] V. K. Madhura N Hegde, Sanjeetha K Shetty, Sheikh Mohammed Anas, “Fake product review monitoring,” Int. Res. J. Eng. Technol., vol. 05, no. 06, p. 4, 2018.

[11] S. Rayana and L. Akoglu, “Collective Opinion Spam Detection: Bridging Review Networks and Metadata,” Sigkdd, pp. 985–994, 2015.