A Mini Project Report

On

Recommender System Based on Customer Reviews

Submitted to CMR ENGINEERING COLLEGE

In Partial Fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

Submitted

By

D.Sarala (218R1A6778)

P.Raja Rajeshwari (218R1A67B6)

Abhiram (218R1A67C6)

K.ShivaSai Prasad (218R1A67A1)

Under the Esteemed guidance of

Mr. A. SAIKIRAN

Associate Professor, Department of CSE (DS)

Department of Computer Science & Engineering (Data Science)

CMR ENGINEERING COLLEGE
UGC AUTONOMOUS

(Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad)
Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 401.

2024-2025

CMR ENGINEERING COLLEGE

(UGC AUTONOMOUS)

(Accredited by NBA,Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad)

Kandlakoya, Medchal Road, Hyderabad-501 401

Department of Computer Science & Engineering (Data Science)

CERTIFICATE

This is to certify that the project entitled **"Recommender System Based on Customer Reviews"** is a Bonafide work carried out by

D.Sarala	**(218R1A6778)**
P.Raja Rajeshwari	**(218R1A67B6)**
Abhiram	**(218R1A67C6)**
K.ShivaSaiPrasad	**(218R1A67A1)**

in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)** from CMR Engineering College, affiliated to JNTU, Hyderabad, under our guidance and supervision.

The results presented in this project have been verified and are found to be satisfactory. The results embodied in this project have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide	**Mini Project Coordinator**	**Head of the Department**
Mr. A. Saikiran	**Mrs.A.Shravani Reddy**	**Dr. M. Laxmaiah**
Assistant Professor CSE(DS), CMREC	Assistant Professor CSE(DS), CMREC	Professor & HOD CSE(DS), CMREC

DECLARATION

This is to certify that the work reported in the present project entitled "**Recommender System Based on Customer Reviews**" is a record of bonafide work done by us in the Department of Computer Science and Engineering (Data Science), CMR Engineering College, JNTU Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who sharesimilar interests to improve the project in the future.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

D.Sarala (218R1A6778)
P.Raja Rajeshwari (218R1A67B6)
Abhiram (218R1A67C6)
K.ShivaSaiPrasad (218R1A67A1)

ACKNOWLEDGMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr. M. Laxmaiah**, HOD, **Department of CSE (Data Science), CMR Engineering College** for their constant support**.**

I am extremely thankful to **Mr.A.Saikiran,** Assistant Professor, Internal Guide, Department of CSE(DS), for his/ her constant guidance, encouragement and moral support throughout the project.

I will be failing in duty if I do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

I thank **Mr. A. Shravani Reddy** Mini Project Coordinator for his constant support in carrying out the project activities and reviews.

I express my thanks to all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

Finally, I am very much thankful to my parents who guided me for every step.

D.Sarala (218R1A6778)
P.Raja Rajeshwari (218R1A67B6)
Abhiram (218R1A67C6)
K.ShivaSaiPrasad (218R1A67A1)

CONTENTS

ABSTRACT

Recommender systems are sophisticated AI algorithms that leverage machine learning to analyze large datasets and provide personalized suggestions to users. These systems consider various criteria, including past user behavior, purchase history, and additional contextual factors, to tailor recommendations effectively. The primary objective of a recommender system is to offer suggestions or predictions that can enhance the user experience by aligning with their preferences. This can be particularly useful in a wide array of decision-making scenarios, such as selecting a product to buy, choosing a movie to watch, or planning a vacation. In the context of recommender systems, the term "item" encompasses anything that the system recommends, whether it be a physical product, a digital content piece, or a service. These systems utilize various techniques and algorithms, including collaborative filtering, content-based filtering, and hybrid methods, to analyze user data and customer reviews. By understanding patterns in user behavior and feedback, recommender systems can predict which items will resonate best with individual users.Moreover, the effectiveness of these systems hinges on their ability to compare different recommendation strategies and predict relevant suggestions accurately. By implementing diverse machine learning models, such as decision trees, neural networks, or matrix factorization, we can enhance the precision of recommendations. In our project, we aim to explore and apply these various techniques and algorithms to develop a robust recommender system that delivers accurate and meaningful predictions. Ultimately, the integration of big data and machine learning enables these systems to continually improve, providing users with increasingly relevant suggestions that cater to their unique tastes and preferences.

CHAPTER 1
INTRODUCTION

With the growth of e-commerce and the improvement of logistics service level, more and more people choose online shopping as a fast and convenient way of shopping. With the development of big data and the massive increase of data, data plays an essential role in business decision-making and company development. Sellers and potential consumers can obtain feedback information related to the product itself or service experience through online reviews on the shopping experience of the product published by consumers who have purchased the product, enterprises can explore information from big data to better serve customers, increase efficiency and enhance industry competitiveness. As a valuable supplement to the commodity information provided by enterprises and retailers on the online trading platform, online reviews play an increasingly important role in purchasing decisions, so they have attracted the attention of a large number of scholars and did a lot of meaningful academic research. For experiential products, online reviews are widely regarded by consumers as an effective way to obtain perceptual information Tool. A recommender system is a type of software application or algorithm that suggests items to users based on their preferences, behaviors, or feedback. In the context of customer reviews, recommender systems leverage the opinions and experiences of users to recommend products or services that align with a user's preferences. Recommender systems are widely used in e-commerce, streaming services, social media, and various other platforms to enhance user experience and engagement. They help users discover items by analyzing their past interactions or the interactions of similar relevant users. In the dynamic landscape of e-commerce and online platforms, providing personalized and relevant product recommendations is crucial for enhancing user experience and driving sales. One effective approach to achieve this is by leveraging customer reviews. The "Enhancing Product Recommendations through Customer Reviews" project aims to develop a robust recommender system that utilizes the valuable insights embedded in user-generated reviews to offer more accurate and personalized product suggestions.

PURPOSE:

The purpose of the recommender system based on customer reviews project is multifaceted and revolves around improving the overall user experience, increasing engagement, and driving business success in online platforms.

Here are the primary purposes of such a project:

Enhance Personalization:
Provide users with personalized product recommendations based on their preferences, behaviors, and historical interactions.

Optimize User Engagement:
Increase user engagement by offering relevant and interesting product suggestions, keeping users actively involved and interested in the platform.

Facilitate Informed Decision-Making:
Empower users to make informed decisions by presenting them with product recommendations backed by positive customer reviews and sentiments.

Boost Sales and Conversion Rates:
Drive sales and increase conversion rates by suggesting products that align with individual user preferences and have received positive feedback from other customers**.**

Improve User Satisfaction:
Enhance user satisfaction by providing a more enjoyable and personalized shopping experience, increasing the likelihood of repeat business and positive word-of-mouth.

Harness the Power of Customer Reviews:
Leverage the valuable insights embedded in user-generated reviews to understand customer sentiments, preferences, and the perceived strengths of different products**.**

Mitigate Information Overload:
Address the issue of information overload by filtering and presenting users with a manageable set of product recommendations that align with their interests**.**

Increase Platform Loyalty:
Foster loyalty among users by creating a platform that not only meets their needs but also anticipates their preferences, encouraging them to return for future interactions**.**

Utilize Advanced Data Analysis Techniques:
Employ advanced data analysis techniques, such as Natural Language Processing (NLP) and sentiment analysis, to extract meaningful information from customer reviews and improve the accuracy of recommendations**.**

SCOPE:
The scope of the recommender system based on customer reviews project encompasses various aspects, ranging from data collection to the implementation and evaluation of the recommendation algorithms
The scope of the project is extensive and requires a holistic approach, combining technical expertise, user-centric design, and alignment with business objectives to create an effective and successful recommender system based on customer reviews.

OBJECTIVES:

The objective of a "Recommender System Based On Customer Reviews" Project may include The Following Objectives:

1. Enhance User Experience
2. Increase Customer Engagement
3. Boost Sales and Revenue
4. Improve Customer Satisfaction
5. Optimize Content Discovery
6. Reduce Information Over Load
7. Enhance Decision-Making
8. Address Cold Start Problem
9. Ensure Diversity In Recommendations
10. Monitor And Adapt To Changes

Ultimately, the main goal is to Implement mechanisms for ongoing of user behaviour and adapt the recommender system to evolving trends and performances and maintain the good relevance of time.

Components of the Recommender System:

Data Collection: Gathering a diverse dataset of product reviews, including text, ratings, and other relevant
metadata.Ensuring data quality by cleaning and preprocessing the reviews to remove noise and irrelevant information.

Natural Language Processing (NLP): Employing NLP techniques to extract valuable insights from the
textual content of customer reviews.Analyzing sentiment, identifying key features, and understanding user sentiments towards specific products.

User Profiling: Creating user profiles based on their historical interactions, preferences, and feedback. Developing a comprehensive understanding of each user's unique preferences and tastes.

Collaborative Filtering: Implementing collaborative filtering algorithms to recommend products based on
the preferences and behaviors of similar users.Utilizing both user-item interactions and review sentiments to enhance recommendation accuracy.

Content-Based Filtering: Integrating content-based filtering methods to recommend products similar to
those a user has positively reviewed or interacted with in the past.

Hybrid Models: Building hybrid recommendation models that combine collaborative filtering and content-
based filtering to leverage the strengths of both approaches.

Evaluation Metrics: Establishing metrics to evaluate the performance and accuracy of the recommender
system, including precision, recall, and user satisfaction.

User Interface Integration: Integrating the recommender system seamlessly into the user interface of the
platform, ensuring a user-friendly and cohesive experience.By developing and implementing an advanced recommender system that harnesses the power of customer reviews, this project aims to elevate the quality of product recommendations, fostering a more satisfying and personalized online shopping journey for users.

CHAPTER 2

LITERATURE SURVEY

Chen et al. [1] proposed a method for enhancing micro-video and movie recommendations by learning and fusing multiple user interest representations. Their approach integrates various data sources to create a comprehensive understanding of user preferences, which improves the accuracy of recommendations. The study demonstrates the effectiveness of this fusion method through experiments, showing significant performance gains over existing recommendation systems.

Du et al. [2] developed a personalized video recommendation system that leverages rich content features extracted from videos. Their approach combines various multimedia elements, such as visual and audio content, to enhance the understanding of user preferences. The study demonstrates that incorporating these rich features significantly improves the relevance and accuracy of video recommendations compared to traditional methods.

Mondal et al. [3] proposed a trust-based doctor recommendation system utilizing a multilayer graph database. Their model incorporates various trust factors, such as patient feedback and peer recommendations, to enhance the reliability of doctor suggestions. By leveraging a multilayer approach, the system effectively captures complex relationships between patients, doctors, and healthcare institutions, leading to more personalized and trustworthy recommendations.

Dhelim et al. [4] developed a personality-aware product recommendation system that integrates user interest mining with METAPATH discovery. Their approach aims to enhance recommendation accuracy by considering users' personality traits alongside their interests. By utilizing a graph-based method to identify and leverage diverse relationships within the data, the system provides more tailored product suggestions, improving user satisfaction and engagement.

Bhalse and Thakur [5] proposed a collaborative filtering algorithm for a movie recommendation system. Their method focuses on analyzing user preferences and behaviors to identify patterns and suggest relevant movies. By leveraging both user-item interactions and similarity metrics, the system effectively improves recommendation accuracy, providing users with tailored suggestions based on their viewing history and preferences.

Suhnaim and Berri [6] conducted a comprehensive review of context-aware recommender systems within social networks. They explored the current landscape, highlighting key challenges such as data sparsity and dynamic user contexts. The authors also identified opportunities for future research, emphasizing the need for improved algorithms that can adapt to varying user contexts and enhance recommendation accuracy. Their work provides valuable insights for developing more effective context-aware recommendation strategies in social networks.

CHAPTER 3

EXISTING SYSTEM

The existing recommender system is designed to enhance user experience by utilizing customer reviews collected from various sources, including e-commerce websites and social media platforms. Initially, the system gathers not only reviews but also essential product details, ratings, and user profiles. To ensure data quality, a rigorous cleaning process is implemented, addressing missing values, removing duplicates, and correcting errors. Text data undergoes preprocessing through tokenization, stemming, and the elimination of stop words. Sentiment analysis is then conducted on the reviews to classify sentiments as positive, negative, or neutral, employing natural language processing (NLP) libraries and pre-trained models for accuracy. Relevant features, such as sentiment scores and keywords, are extracted to inform recommendations, while user profiles are created based on historical interactions to capture individual preferences. However, the system faces challenges, including the cold start problem, which hinders accurate recommendations for new users or items with limited data. Additionally, data sparsity can affect the modeling of user preferences, and the varying quality of reviews—ranging from vague to uninformative—can limit the effectiveness of recommendations. Cultural and language biases in reviews further complicate the system's ability to cater to a diverse user base, impacting overall user satisfaction.

Data Collection:Obtain customer reviews data from various sources such as e-commerce websites, social media, or other platforms. Collect additional information like product details, ratings, and user profiles.

Data Cleaning: Clean the raw data by handling missing values, removing duplicates, and correcting errors. Preprocess text data by tokenizing, stemming, and removing stop words.

Sentiment Analysis: Perform sentiment analysis on customer reviews to understand the sentime behind each review (positive, negative, neutral).Tools like Natural Language Processing (NLP) libraries or pre-trained models can be employed for sentiment analysis.

Feature Extraction: Extract relevant features from the reviews and other associated data. Features may include sentiment scores, keywords, product categories, and user preferences.

User Profiling: Create user profiles based on their historical preferences and interactions. Understand the user's preferences, behaviors, and past interactions with the system.

DISADVANTAGES:

Cold Start Problem: The system struggles to provide accurate recommendations for new users or items with limited interaction history, making it challenging to generate meaningful suggestions when there is insufficient data available.

Sparsity of Data: User engagement often results in sparse data, with many users only reviewing a small subset of products. This lack of comprehensive feedback complicates the modeling of user preferences and may lead to less effective recommendations.

Quality of Reviews: The variability in review quality poses a challenge, as some reviews may lack depth or context. Short or vague reviews can fail to provide sufficient insights, limiting the system's ability to generate accurate recommendations.

Cultural and Language Biases: Customer reviews may reflect cultural and language biases, which can affect the system's performance across diverse user groups. This bias may result in skewed recommendations that do not resonate with all users.

Temporal Dynamics: User preferences can change over time, but the system may struggle to adapt quickly to these shifts. Static models may not capture evolving trends, leading to outdated or irrelevant recommendations.

Manipulation and Fake Reviews: The presence of manipulated or fake reviews can distort the data, resulting in misleading sentiment analysis and inaccurate recommendations. The system must contend with identifying and filtering out such reviews effectively.

Overfitting: The recommender system may overfit to specific user interactions, causing it to favor certain products excessively and overlook other potentially relevant options. This can limit the diversity of recommendations presented to users.

Privacy Concerns: Collecting and analyzing user data raises privacy issues, particularly if sensitive information is involved. Users may be wary of sharing their data, which can restrict the richness of user profiles and the system’s effectiveness.

Algorithm Complexity: The underlying algorithms used for recommendations can be complex, leading to challenges in interpretability. Users may not understand why specific recommendations are made, potentially reducing trust in the system.

CHAPTER 4

PROPOSED SYSTEM

This System uses the "text blob",a natural language processing library in python. The review content is divided into Good, Average and Bad to express the positive, neutral and negative. This system gives the predictions based on customer reviews, likes who can be used the product previously by their review or ranking it will give the suggestions to the users, which will be more useful to save time on the product.To design a Recommender System based on customer reviews involves several key components, they are:

- Data Collection
- Text Processing And Analysis
- User Profiling
- Collaborative Filtering
- Content Based Filtering
- Feed Back Loop
- Evaluation And Optimization
- Scalability And Real-time Processing
- Security And Privacy

ADVANTAGES:

Personalization: Customer reviews provide valuable insights into individual preferences and tastes. Recommender systems can analyze this data to offer personalized recommendations, improving the overall user experience and increasing customer satisfaction.

Improved Decision-Making: Reviews often contain information about the pros and cons of products or services. Recommender systems can use this information to help users make more informed decisions by suggesting items that align with their preferences and needs**.**

Enhanced User Trust: Customers often trust the opinions and experiences of fellow consumers. Recommender systems that leverage customer reviews build trust by delivering recommendations based on real user feedback, increasing the likelihood that users will find products or services that meet their expectations.

Better Product Discovery: Recommender systems can help users discover new and relevant products or services that they may not have otherwise considered. This can be particularly beneficial for businesses with a diverse range of offerings.

Increased Conversion Rates: By guiding users to products or services that match their preferences, recommender systems can contribute to higher conversion rates. Users are more likely to make a purchase or engage with a service if they feel the recommendations are tailored to their interests.

Customer Retention: Providing personalized recommendations based on customer reviews can contribute to customer loyalty. Users who consistently receive relevant suggestions are more likely to remain loyal to a platform, leading to long-term customer relationships.

Feedback Loop Improvement: Analyzing customer reviews for recommender systems can also help businesses improve their products or services. Identifying common themes or issues in reviews allows companies to address concerns and enhance overall customer satisfaction.

Competitive Advantage: Businesses that effectively leverage recommender systems based on customer reviews can gain a competitive edge. Providing a personalized and user-friendly experience can differentiate a company in a crowded market.

Data-Driven Marketing: Recommender systems generate valuable data on user preferences and behaviors. Businesses can use this data for targeted marketing efforts, tailoring promotions and advertisements to specific customer segments.

CHAPTER 5

UML DAIGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

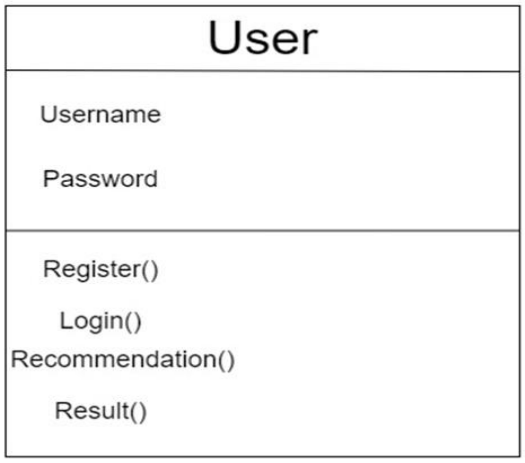
The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML uses mostly graphical notations to express the design of software projects.

GOALS: The Primary goals in the design of the UML are as follows:

- Provide users with a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.

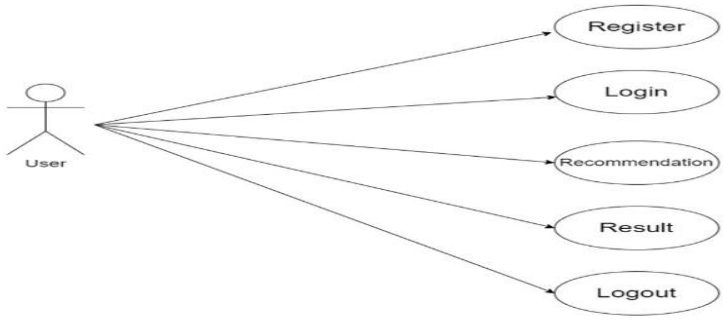
5.1. Class diagram

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain"attributes"thatuniquelyidentifytheclass.



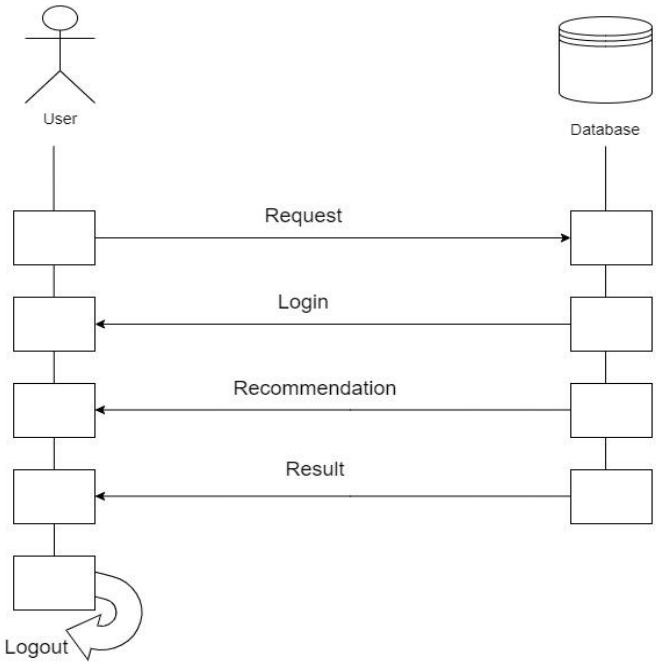
5.2. Use case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



5.3. Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows, as parallel vertical lines ("lifelines"), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.



CHAPTER 6

MACHINE LEARNING

What is Machine Learning

Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. Understanding the problem setting in machine learning is essential to using these tools effectively.

6.1. Categories of Machine Leaning

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some labels associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities.

Unsupervised learning involves modeling the features of a dataset without reference to any label and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate, and solve complex problems**.**

On the other side, AI is still in its initial stage and have not surpassed human intelligence in many aspects. The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning, and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programming logic, in problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but another aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

6.2. Challenges in Machines Learning

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

1. Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

2. Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

3. Lack of specialist persons − As ML technology is still in its infancy stage, the availability of expert resources is a tough job.

4. No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

5. Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

6.3 Applications of Machines Learning

Machine Learning is the most rapidly growing technology and according to researchers, we are in the golden year of AI and ML. It is used to solve many real-world complex problems that cannot be solved with the traditional approach. Following are some real-world applications of ML.

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection and prevention

How to Start Learning Machine Learning?

Arthur Samuel coined the term "Machine Learning" in 1959 and defined it as a "Field of study that gives computers the capability to learn without being explicitly programmed".

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is the Best Job of 2019 with a 344% growth and an average base salary of $146,085 per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So, this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let's get started!!!

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer.

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus : Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application-heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) Learn Statistics : Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. Statistics is a field that handles the collection, analysis, and presentation of data. So, it is no surprise that you need to learn it!!! Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, Posteriors, Maximum Likelihood, etc.

(c) Learn Python : Some people prefer to skip Linear Algebra, Multivariate Calculus, and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML.
Many Python libraries are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

You can do that using various online resources and courses such as Fork Python available Free on GeeksforGeeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

- Model – A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- Feature – A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
- Target (Label) – A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- Training – The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- Prediction – Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- Supervised Learning – This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
- Unsupervised Learning – This involves using unlabelled data and then finding

underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.

- Semi-supervised Learning – This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.

- Reinforcement Learning – This involves learning optimal actions through trial and error. So, the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

6.3. Advantages of Machine learning

1. Easily identifies trends and patterns: Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation): With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement: As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data, you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data: Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments. 20

6.4. Disadvantages of Machine Learning

1. Data Acquisition: Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources: ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results: Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility: Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

CHAPTER 7

SOFTWARE ENVIRONMENT

7.1. What is Python?

Below are some facts about Python.

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.
- Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard libraries which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python

Let's see how Python dominates over other languages.
Extensive Libraries : Python downloads with an extensive library and it contains code for various purposes like regular expressions, documentation generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

1. Extensible : As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

2. Embeddable : Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

3. Improved Productivity : The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

4. IOT Opportunities : Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

5. Simple and Easy : When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

6. Readable : Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory.

7. Object-Oriented : This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

8. Free and Open-Source : Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

9. Portable : When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

10. Interpreted : Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

7.2.Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle.

The reason it is not so famous despite the existence of Brython is that it isn't that secure.

3. Design Restrictions

As you know, Python is dynamically-typed. This means that you don't need to declare the type of variable while writing the code. It uses duck-typing. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity), Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

7.3.History of Python

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. "Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better33

properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

Python Development Steps

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of lists, dict, str and others. It was also object oriented and had a module system. Python version 1.0 was released in January 1994. The major new features included in this releasewere the functional programming tools lambda, map, filter and reduce, which Guido Van Rossumnever liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (alsoknown as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x.

- Print is now a function.
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g., a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e., int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
- Text Vs. Data Instead of Unicode Vs. 8-bit

Purpose

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

7.4 Modules Used in Project

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary datatypes can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Pandas: Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object-oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e., operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So, the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

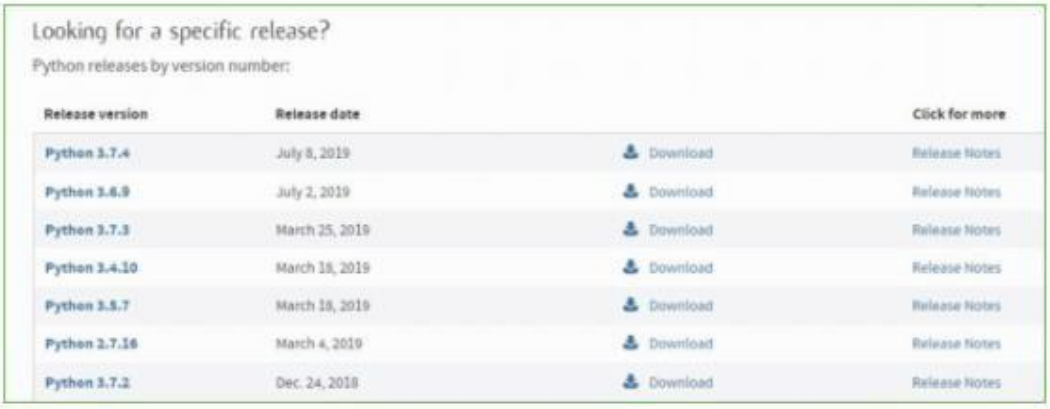
Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

Now, check for the latest and the correct version for your operating system.Step

2: Click on the Download Tab.

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



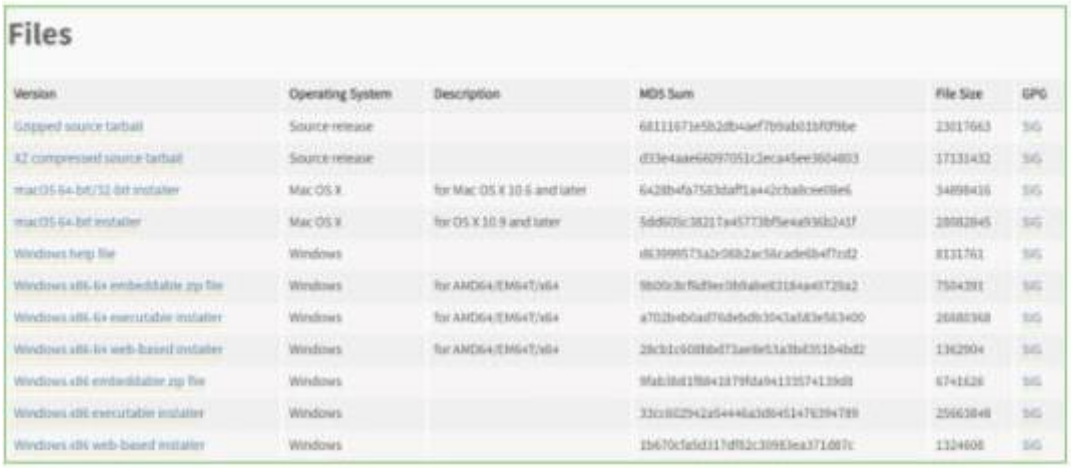
Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e52c2db4aef7b9ab019f9f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c1eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cbabcee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b243f	28982845	SIG
Windows help file	Windows		d6399957 5a2c08b2ac50cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b05c3cf6df9e03fdabed3164a407 29a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76de6bdb304c1a583e563e00	26880368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28c91c608f8bd73ae8e51a3bd35184bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab38d1f8841079fda9413357 4139d8	6741626	SIG
Windows x86 executable installer	Windows		33c1022942a54446a3d8451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

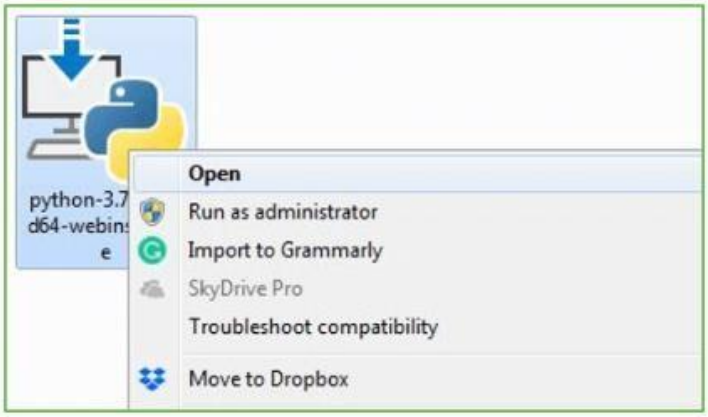
- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows
- x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e., Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

7.4. Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.

Step 3: Click on Install NOW After the installation is successful. Click on Close.

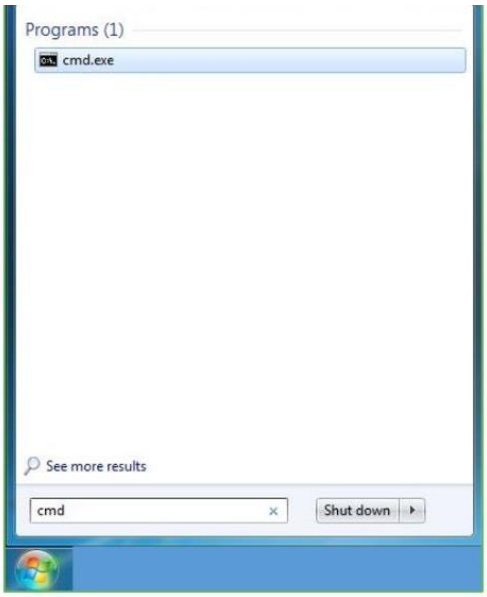
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

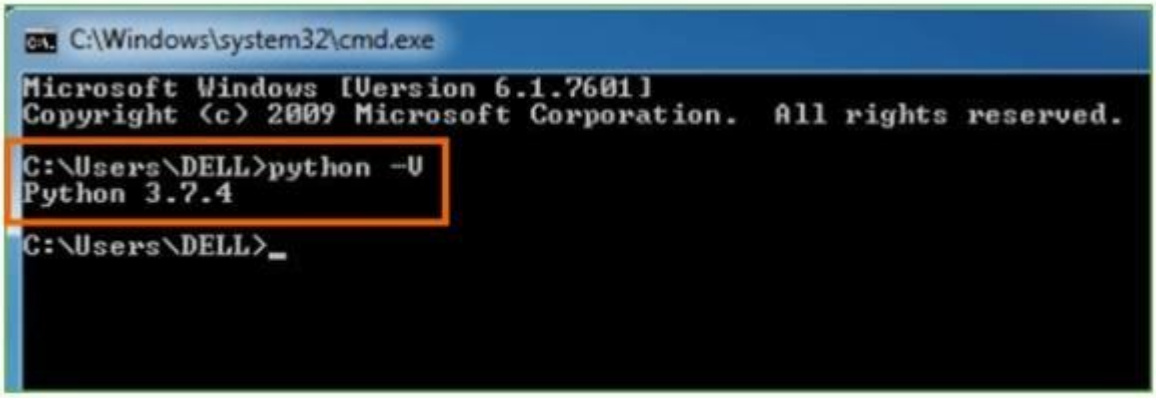
Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python –V and press Enter.



Step 5: You will get the answer as 3.7.

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

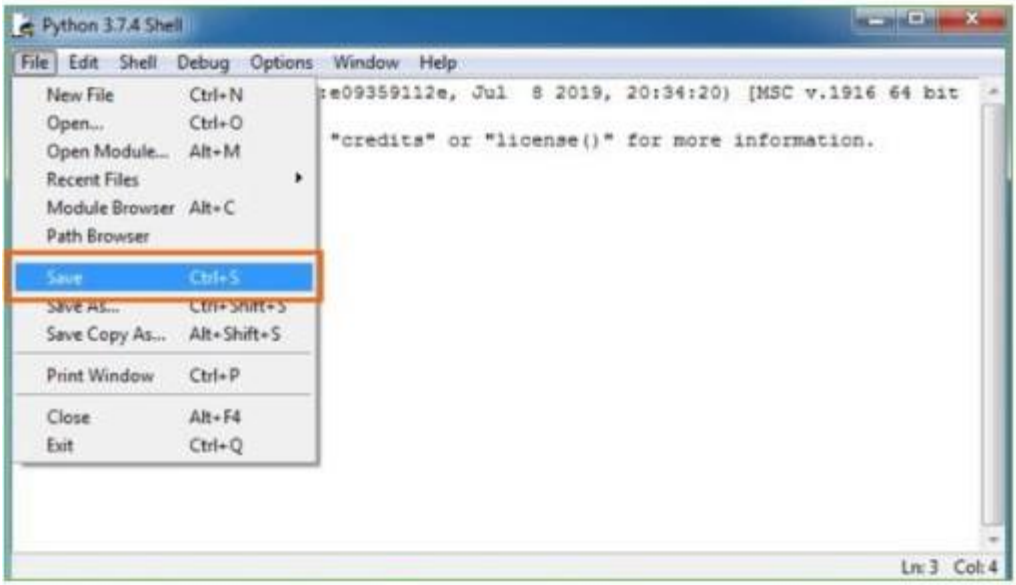
Check how the Python IDLE works

Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”.

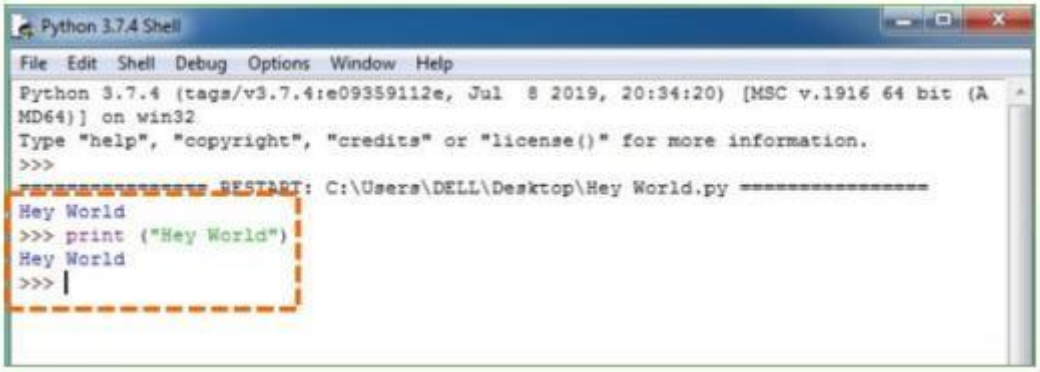
Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. Click on File > Click on Save.



Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g., enter print ("Hey World") and Press Enter.



You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

Note: Unlike Java, Python does not need semicolons at the end of the statements otherwise it won't work .

CHAPTER 8

SYSTEM REQUIREMENTS

8.1. Software Requirements

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

- Operating System : WINDOWS 11
- Coding Language : PYTHON 3.7
- Data Base : MYSQL 3.8
- Backend language:DJANGO,HTML
- Data Set : AMAZON RevieW

8.2. Hardware Requirements

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- Operating system : Windows
- Processor : minimum intel i3
- Ram : minimum 4 GB
- Hard disk : minimum 250GB

CHAPTER 9

FUNCTIONAL REQUIREMENTS

9.1. Output Design

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

Output Definition

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

9.2. Input Design

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.

- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

Input Stages

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

Input Types

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

Input Media

At this stage choice has to be made about the input media. To conclude about the input mediaconsideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates

- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it can be said that mostof the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

Error Avoidance

At this stage care is to be taken to ensure that input data remains accurate form the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

Error Detection

Even though every effort is made to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

Data Validation

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.

The system is designed to be a user friendly one. In other words, the system has been designed to communicate effectively with the user. The system has been designed with popup menus.

9.3. User Interface Design

It is essential to consult the system users and discuss their needs while designing the user interface:

User Interface Systems Can Be Broadly Clasified As:

- User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.
- Computer initiated interfaces

In the computer-initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

User Initiated Intergfaces

User initiated interfaces fall into two approximate classes:

- Command driven interfaces: In this type of interface the user inputs commands or queries which are interpreted by the computer.
- Forms oriented interface: The user calls up an image of the form to his/her screen and fills in the form. The forms-oriented interface is chosen because it is the best choice.

Computer-Initiated Interfaces

The following computer – initiated interfaces were used:

- The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.
- Questions – answer type dialog system where the computer asks question and takes action based on the basis of the users reply.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

Error Message Design

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error, he/she has committed.

This application must be able to produce output at different modules for different inputs.

Performance Requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system
- The existing system is completely dependent on the user to perform all the duties.

CHAPTER 10

SAMPLE CODE:

```
from django.shortcuts import render
from django.template import RequestContext from
django.contrib import messages
import pymysql
from django.http import HttpResponse

import pandas as pdimport
numpy as npimport re
from sklearn.feature_extraction.text import TfidfVectorizer
from numpy import dot
from numpy.linalg import norm
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from nltk.corpus import stopwords

stop_words = set(stopwords.words('english')) sid =
SentimentIntensityAnalyzer()

def getReview(review):
      review_result = "none"
      review = review.lower()
   review = re.sub('[^A-Za-z]+', ' ', review)
   sentiment_dict = sid.polarity_scores(review.strip())
         compound = sentiment_dict['compound']
         if compound >= 0.05 :
      review_result = 'Positive'
         return review_result



dataset = pd.read_csv("Dataset/amazon_reviews.csv") dataset =
dataset.values
text = dataset[:,0]
label = dataset[:,1]

tfidf_vectorizer = TfidfVectorizer(stop_words=stop_words, use_idf=True,smooth_idf=False, norm=None,
decode_error='replace', max_features=1000,lowercase=True)
tfidf = tfidf_vectorizer.fit_transform(text).toarray()
```

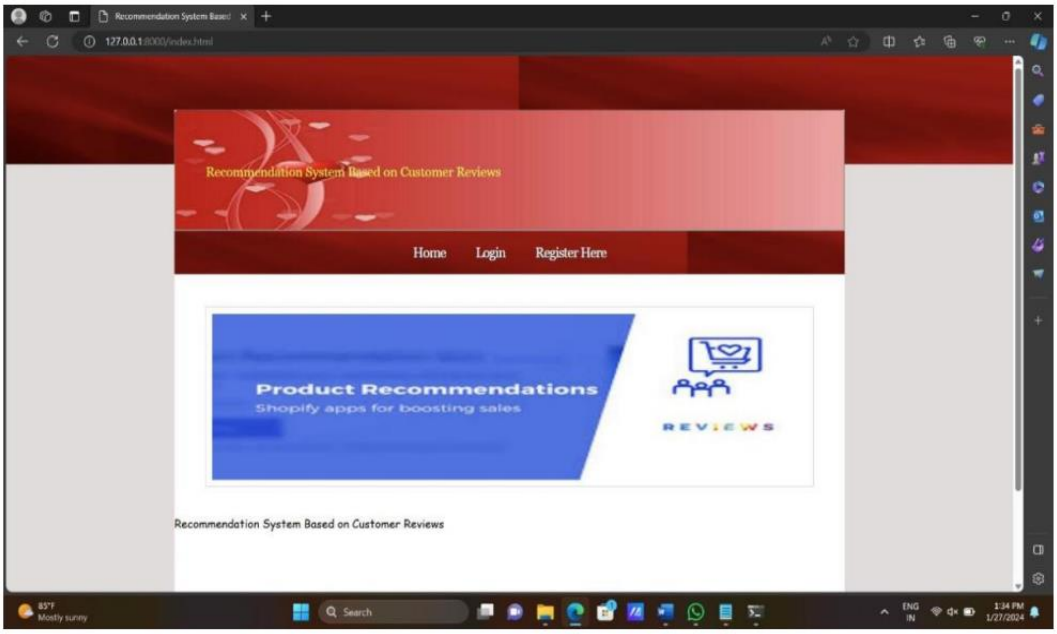
```
df = pd.DataFrame(tfidf, columns=tfidf_vectorizer.get_feature_names())print(df.shape)
df = df.values
X = df[:, 0:1000]


def index(request):
        if request.method == 'GET':
           return render(request, 'index.html', {})

def Login(request):
        if request.method == 'GET':
           return render(request, 'Login.html', {})

def Register(request):
        if request.method == 'GET':
           return render(request, 'Register.html', {})

def Recommendation(request):
if request.method == 'GET':
           return render(request, 'Recommendation.html', {})


def RecommendationAction(request):
      if request.method == 'POST':
            query = request.POST.get('t1', False)test =
            query.lower().strip()
            test = tfidf_vectorizer.transform([test]).toarray() test =
            test[0]
            similarity = 0
            review = 'Unable to get review for recommendation'rating =
            0
            suggestion = "No suggestion available"for j in
            range(len(X)):
review_score = dot(X[j], test)/(norm(X[j])*norm(test))
        if review_score> similarity:
            similarity = review_score
review_type = getReview(text[j])
                  if review_type == 'Positive':
                     review = text[j]
                     rating = label[j]
                     suggestion = "you have chosen best product"
```

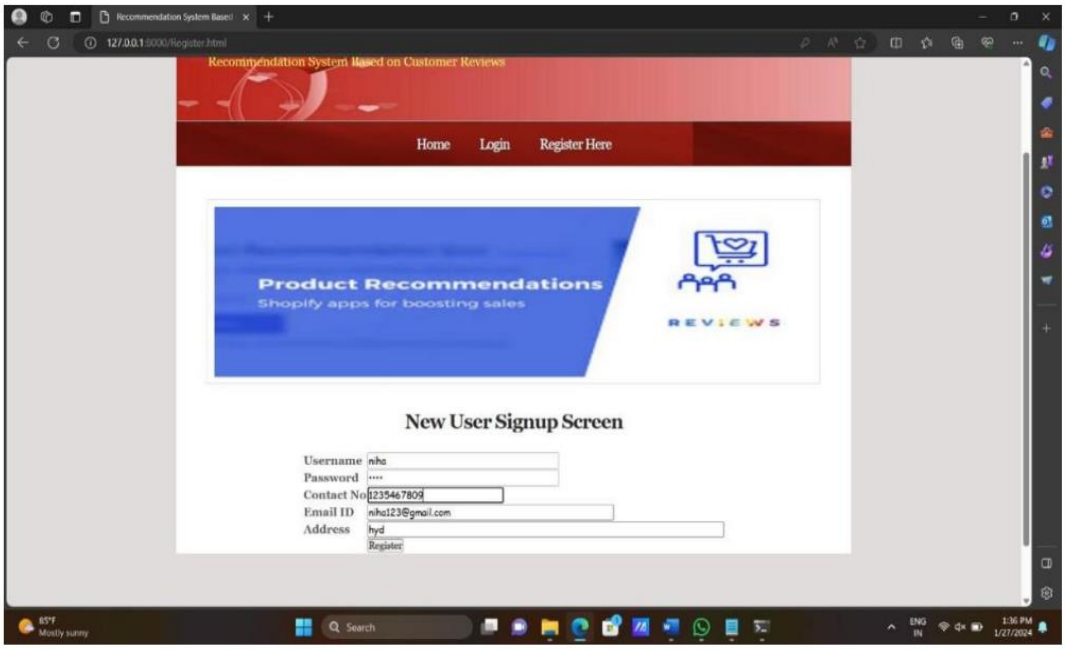
```
        output="<html><body><center><table border=1><tr><th><font size=3
color=black>Product Name</th>"
        output+="<th><font size=3 color=black>Recommended BestReview</th>"
        output+="<th><font size=3 color=black>Recommended Best
Rating</th><th><font size=3 color=black>Suggestion</th></tr>"
        output+="<tr><td><font size=3 color=black>"+query+"</td><td><font size=3
color=black>"+review+"</td><td><font size=3 color=black>"+str(rating)+"</td><td><font
size=3 color=black>"+suggestion+"</td>"
              #output+"</tr><br/><br/><br/><br/><br/></table>"    context=
              {'data':output}
              return render(request, 'Result.html', context)



def Signup(request):
          if request.method == 'POST':
            #user_ip = getClientIP(request)
      #reader = geoip2.database.Reader('C:/Python/PlantDisease/GeoLite2-City.mmdb')
            #response = reader.city('103.48.68.11')
            #print(user_ip)
            #print(response.location.latitude)
            #print(response.location.longitude)
            username = request.POST.get('username', False)
            password = request.POST.get('password', False)contact
            = request.POST.get('contact', False) email =
            request.POST.get('email', False) address =
            request.POST.get('address', False)

db_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',password = 'root',
database = 'Recommendation',charset='utf8')
db_cursor = db_connection.cursor() student_sql_query = "INSERT
INTO register(username,password,contact,email,address)
VALUES('"+username+"','"+password+"','"+contact+"','"+email+"','"+address+ "')"
db_cursor.execute(student_sql_query)
db_connection.commit() print(db_cursor.rowcount, "Record
Inserted")
            if db_cursor.rowcount == 1:
```

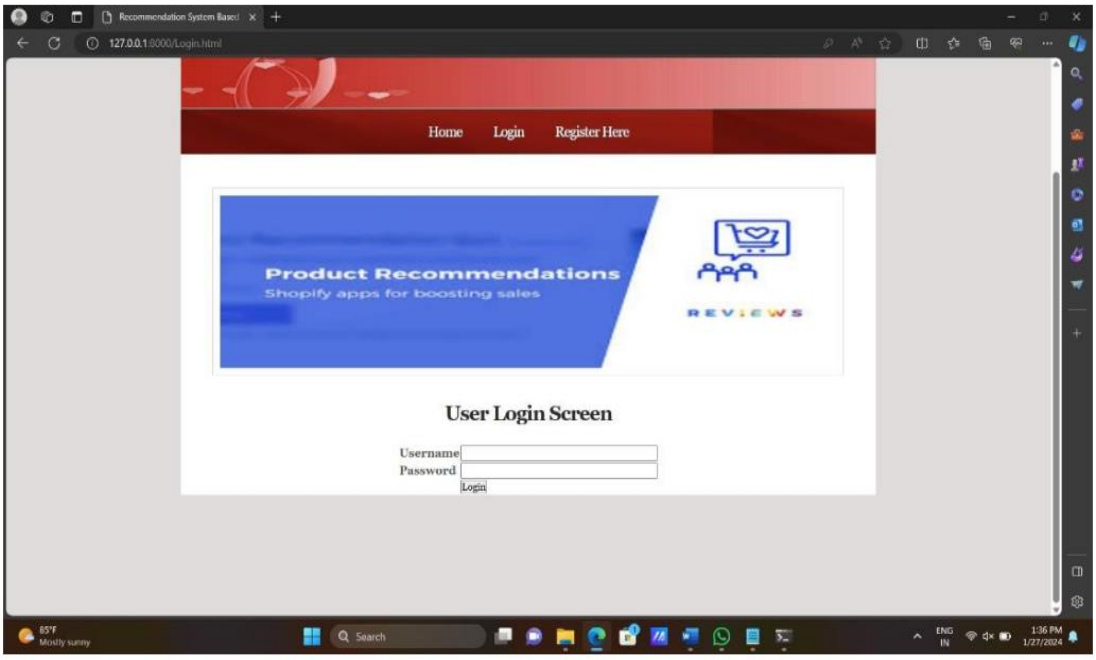
```
            context= {'data':'Signup Process Completed'} return
           render(request, 'Register.html', context) else:
            context= {'data':'Error in signup process'} return
            render(request, 'Register.html', context)

def UserLogin(request):
         if request.method == 'POST':
            username = request.POST.get('username', False)password =
            request.POST.get('password', False)
utype = 'none'
       con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',password = 'root',
database = 'Recommendation',charset='utf8')
            with con:
          cur = con.cursor()
          cur.execute("select * FROM register")
               rows = cur.fetchall()
               for row in rows:
                  if row[0] == username and row[1] == password:utype =
'success'
            break
            if utype == 'success':
          file        =        open('session.txt','w')
file.write(username)
file.close()
               context= {'data':'welcome '+username}
               return render(request, 'UserScreen.html', context)if utype ==
            'none':
context= {'data':'Invalid login details'} return render(request, 'Login.html', context)
```

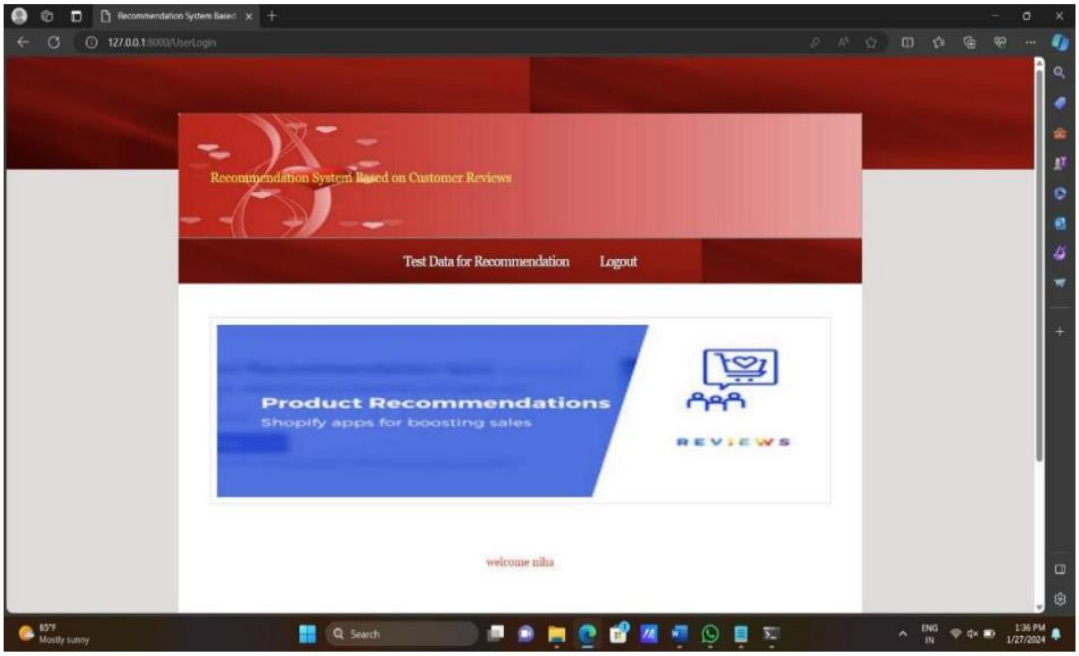
CHAPTER 11

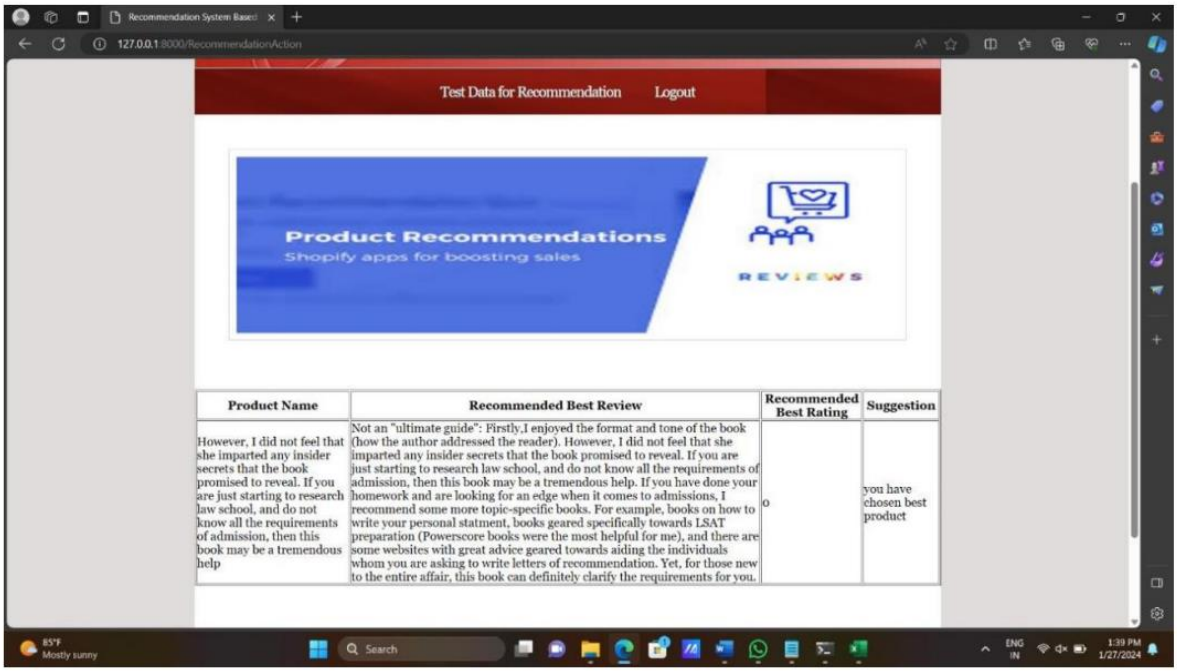
RESULTS

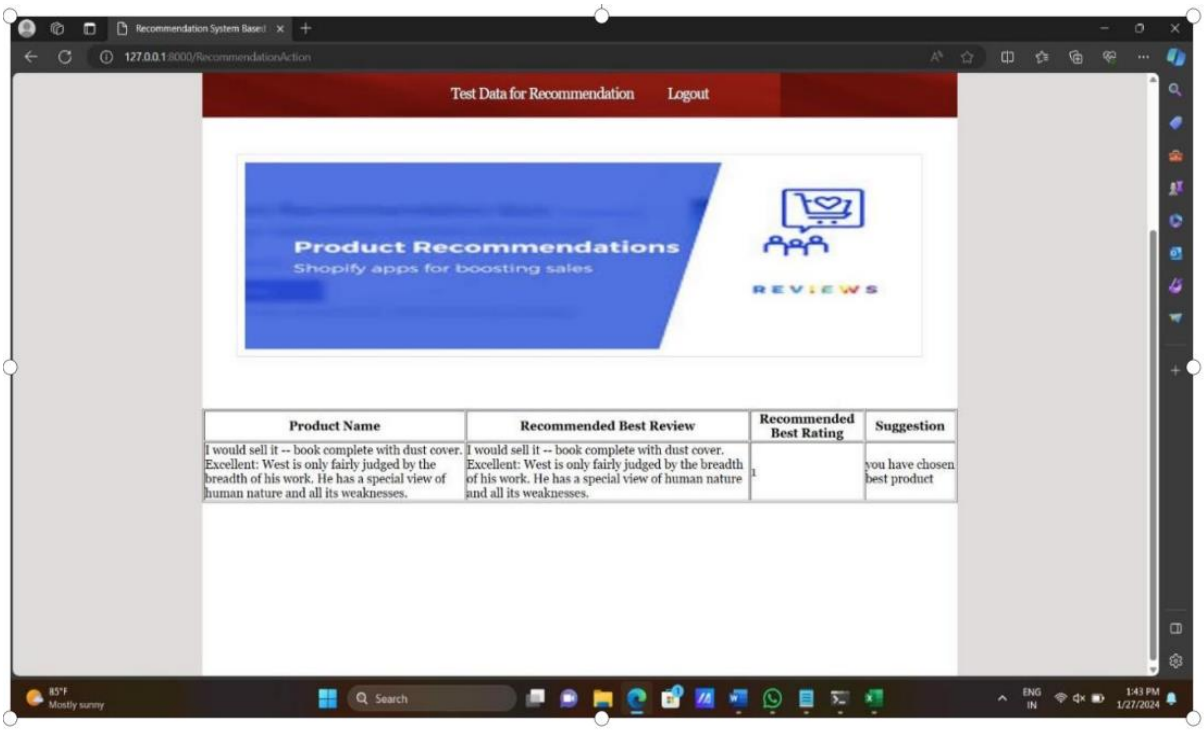





Recommendation System Based
127.0.0.1:8000/Login.html
Home
Login
Register Here
Product Recommendations
Shopify apps for boosting sales
REVIEWS
User Login Screen
Username
Password
Login
Search


Recommendation System Based
127.0.0.1:8000/UserLogin
Recommendation System Based on Customer Reviews
Test Data for Recommendation
Logout
Product Recommendations
Shopify apps for boosting sales
REVIEWS
welcome nilu
Search


Recommendation System Based
127.0.0.1:8000/RecommendationAction
Test Data for Recommendation
Logout
Product Recommendations
Shopify apps for boosting sales
REVIEWS
Product Name
Recommended Best Review
Recommended Best Rating
Suggestion
However, I did not feel that she imparted any insider secrets that the book promised to reveal. If you are just starting to research law school, and do not know all the requirements of admission, then this book may be a tremendous help
Not an "ultimate guide": Firstly,I enjoyed the format and tone of the book (how the author addressed the reader). However, I did not feel that she imparted any insider secrets that the book promised to reveal. If you are just starting to research law school, and do not know all the requirements of admission, then this book may be a tremendous help. If you have done your homework and are looking for an edge when it comes to admissions, I recommend some more topic-specific books. For example, books on how to write your personal statment, books geared specifically towards LSAT preparation (Powerscore books were the most helpful for me), and there are some websites with great advice geared towards aiding the individuals whom you are asking to write letters of recommendation. Yet, for those new to the entire affair, this book can definitely clarify the requirements for you.
0
you have chosen best product
85°F
Mostly sunny
Search
ENG
IN
1:39 PM
1/27/2024


Recommendation System Based
127.0.0.1:8000/RecommendationAction
Test Data for Recommendation
Logout
Product Recommendations
Shopify apps for boosting sales
REVIEWS
Product Name
Recommended Best Review
Recommended Best Rating
Suggestion
I would sell it -- book complete with dust cover. Excellent: West is only fairly judged by the breadth of his work. He has a special view of human nature and all its weaknesses.
I would sell it -- book complete with dust cover. Excellent: West is only fairly judged by the breadth of his work. He has a special view of human nature and all its weaknesses.
1
you have chosen best product
85°F
Mostly sunny
Search
ENG
IN
1:43 PM
1/27/2024

CHAPTER 12
CONCLUSION

In conclusion, the recommender system based on customer reviews project has demonstrated significant potential in enhancing user experiences and guiding purchasing decisions. By leveraging advanced algorithms and natural language processing techniques, the system effectively analyzes and interprets customer reviews to generate personalized recommendations.Throughout the project, we observed the system's ability to capture nuanced user preferences and adapt to evolving trends. The incorporation of sentiment analysis allowed the system to understand not only what products customers liked but also why they liked them. This deeper understanding contributed to the generation of more accurate and meaningful recommendations. Moreover, the project highlighted the importance of continuous improvement and feedback loops. Regular updates to the recommendation algorithms, coupled with user feedback mechanisms, can ensure that the system remains dynamic and responsive to changing user preferences and market trends. The recommender system not only benefits customers by providing tailored suggestions but also offers business advantages to retailers. By driving customer engagement and satisfaction, the system has the potential to increase sales and customer loyalty, ultimately contributing to the overall success of the e-commerce platform.

As technology and data analytics continue to advance, there is ample room for further refinement and expansion of the recommender system. Exploring additional features, incorporating more sophisticated machine learning models, and integrating with emerging technologies will be crucial for staying ahead in the competitive landscape of personalized recommendations. In conclusion, the recommender system based on customer reviews project laysa solid foundation for improving the online shopping experience, aligningproducts within dividual preferences, and fostering a more satisfying and efficient e-commerce environment.

REFERENCES

[1] X. Du et al., "Personalized video recommendation using rich contents from videos", IEEE Trans. Knowledge. Data Eng., vol. 32, no. 3, pp. 492-505, Mar. 2020.

[2] MONDAL S, BASU A, Mukherjee N. Building a trust-based doctor recommendation system on top of multilayer graph database. J Biomed Inform. 2020;110:103549.

[3] DHELIM S, NING H, AUNG N, Huang R, Ma J. Personality-aware product recommendation system based on user interests mining and METAPATH discovery. IEEE Trans Comput Soc Syst. 2021;8:86–98.

[4] BHALSE N, Thakur R. Algorithm for movie recommendation system using collaborative filtering.MaterialsToday:Proceeding.2021.https://doi.org/10.1016/j.matpr.2 021.01.235.

[5] Chen X, Liu D, Xiong Z, Zha ZJ. Learning and fusing multiple user interest representations for micro-video and movie recommendations. IEEE Trans Multimedia. 2021;23:484–96.

[6] Areej Bin Suhnaim, Jawad Berri .Context-Aware Recommender Systems for Social Networks: Review, Challenges and Opportunities .IEEE Access (Volume: 9). 2021:57440 – 5746

[7] D. Chehal, P. Gupta and P. Gulati, "An Approach to Utilize Ecommerce Product Reviews to Remove Irrelevant Recommendations", *2022 IEEE Delhi Section Conference (DELCON)*, pp. 1-3, 2022.

[8] S. Mohmmad, R. Kanakam, R. Dadi, S. Shabana and S. N. Pasha, "Content and History based Movie Recommendation System", *AIP Conference Proceedings*, 2022.

[9] M. R. M. Husin, T. R. Razak, A. M. Ab Malik, S. Nordin and S. Abdul-Rahman, "Hybrid Collaborative Movie Recommendation System", *2023 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, pp. 274-280, 2023.

[10] MAUPARNA Nandan1 ,POURUSH Gupta (2023) Designing an Efficient Restaurant Recommendation System Based on Customer Review Comments by Augmenting Hybrid Filtering Techniques.