**DOMAIN : DEEP LEARNING**

**TECHNOLOGY : PYTHON**

**FAKE REVIEWS DETECTION USING SUPERVISED MACHINE**

**LEARNING**

**OJECTIVE:**

The objective of this project is to create an Effective Detection system for Fake reviews from the text in order to get rid from fake reviews while purchasing a product.

**ABSTRACT:**

With the continuous evolve of E-commerce systems, online reviews are mainly considered as a crucial factor for building and maintaining a good reputation. Moreover, they have an effective role in the decision making process for end users. Usually, a positive review for a target object attracts more customers and lead to high increase in sales. Nowadays, deceptive or fake reviews are deliberately written to build virtual reputation and attracting potential customers. Thus, identifying fake reviews is a vivid and ongoing research area. Identifying fake reviews depends not only on the key features of the reviews but also on the behaviours of the reviewers. This paper proposes a machine learning approach to identify fake reviews. In addition to the features extraction process of the reviews, this paper applies several features engineering to extract various behaviours of the reviewers. The paper compares the performance of several experiments done on a real Yelp dataset of restaurants reviews, we compare the performance of machine learning classifiers; KNN, Naive Bayes (NB), Logistic Regression. The results reveal that Logistic Regression outperforms the rest of classifiers in terms of accuracy achieving best. The results show that the system has better ability to detect a review as fake or original.

**KEYWORDS**: Machine learning, fake, reviews, Logistic Regression….

**INTRODUCTION**

Nowadays, when customers want to draw a decision about services or products, reviews become the main source of their information. For example, when customers take the initiation to book a hotel, they read the reviews on the opinions of other customers on the hotel services. Depending on the feedback of the reviews, they decide to book room or not. If they came to a positive feedback from the reviews, they probably proceed to book the room. Thus, historical reviews became very credible sources of information to most people in several online services. Since, reviews are considered forms of sharing authentic feedback about positive or negative services, any attempt to manipulate those reviews by writing misleading or inauthentic content is considered as deceptive action and such reviews are labelled as fake Such case leads us to think what if not all the written reviews are honest or credible. What if some of these reviews are fake. Thus, detecting fake review has become and still in the state of active and required research area.

The rise of social media has blurred the line between authentic content and advertising, leading to an explosion in deceptive endorsements across the marketplace. Fake online reviews and other deceptive endorsements often tout products throughout the online world. Consequently, the FTC is now using its Penalty Offense Authority to remind advertisers of the law and deter them from breaking it. By sending a [Notice of Penalty Offenses](https://www.ftc.gov/system/files/attachments/penalty-offenses-concerning-endorsements/npo_endorsement_template_letter.pdf) to more than 700 companies, the agency is placing them on notice they could incur significant civil penalties—up to $43,792 per violation—if they use endorsements in ways that run counter to prior FTC administrative cases.

“Fake reviews and other forms of deceptive endorsements cheat consumers and undercut honest businesses,” said Samuel Levine, Director of the FTC’s Bureau of Consumer Protection. “Advertisers will pay a price if they engage in these deceptive practices.”

The Notice of Penalty Offenses allows the agency to seek civil penalties against a company that engages in conduct that it knows has been found unlawful in a previous FTC administrative order, other than a consent order.

The Notice sent to the companies outlines a number of practices that the FTC determined to be unfair or deceptive in prior administrative cases. These include, but are not limited to: falsely claiming an endorsement by a third party; misrepresenting whether an endorser is an actual, current, or  recent user; using an endorsement to make deceptive performance claims; failing to disclose an unexpected material connection with an endorser; and misrepresenting that the experience of endorsers represents consumers’ typical or ordinary experience.

Companies receiving the notice represent an array of large companies, top advertisers, leading retailers, top consumer product companies, and major advertising agencies. A full list of the businesses [receiving the Notice](https://www.ftc.gov/system/files/attachments/penalty-offenses-concerning-endorsements/list-recipients-endorsement-notice.pdf) from the FTC is available on the FTC’s website. A recipient’s presence on this list does not in any way suggest that it has engaged in deceptive or unfair conduct.

In addition to the Notice, the FTC has created [multiple resources for business](https://ftc.gov/endorsements) to ensure that they are following the law when using endorsements to advertise their products and services, which can be found on the FTC’s website.

To this end, this paper applies several machine learning classifiers to identify fake reviews based on the content of the reviews as well as several extracted features from the reviewers. We apply the classifiers on real corpus of reviews taken from open source sites. Besides the normal natural language processing on the corpus to extract and feed the features of the reviews to the classifiers, the paper also applies several features engineering on the corpus to extract various behaviours of the reviewers. The paper compares the impact of extracted features of the reviewers if they are taken into consideration within the classifiers. The papers compares the results in the absence and the presence of the extracted features in two different language models namely TF-IDF. The results indicates that the engineered features increase the performance of fake reviews detection process.

The rapid growth of the Internet influenced many of our daily activities. One of the very rapid growth area is

ecommerce. Generally e-commerce provide facility for customers to write reviews related with its service. The existence of

these reviews can be used as a source of information. For examples, companies can use it to make design decisions of their

products or services, while potential customers can use it to decide either to buy or to use a product. Unfortunately, the

importance of the review is misused by certain parties who tried to create fake reviews, both aimed at raising the popularity

or to discredit the product. This research aims to detect fake reviews for a product by using the text and rating property

from a review.

The rapid growth of the Internet influenced many of our daily activities. One of the very rapid growth area is

ecommerce. Generally e-commerce provide facility for customers to write reviews related with its service. The existence of

these reviews can be used as a source of information. For examples, companies can use it to make design decisions of their

products or services, while potential customers can use it to decide either to buy or to use a product. Unfortunately, the

importance of the review is misused by certain parties who tried to create fake reviews, both aimed at raising the popularity

or to discredit the product. This research aims to detect fake reviews for a product by using the text and rating property

from a review.

The rapid growth of the Internet influenced many of our daily activities. One of the very rapid growth area is ecommerce. Generally e-commerce provide facility for customers to write reviews related with its service. The existence of these reviews can be used as a source of information. For examples, companies can use it to make design decisions of their products or services, while potential customers can use it to decide either to buy or to use a product. Unfortunately, the importance of the review is misused by certain parties who tried to create fake reviews, both aimed at raising the popularity or to discredit the product. This research aims to detect fake reviews for a product by using the text and rating property from a review. Machine learning techniques can provide a big contribution to detect fake reviews of web contents. Generally, web mining techniques find and extract useful information using several machine learning algorithms. One of the web mining tasks is content mining. A traditional example of content mining is opinion mining which is concerned of finding the sentiment of text (positive or negative) by machine learning where a classifier is trained to analyse the features of the reviews together with the sentiments. Usually, fake reviews detection depends not only on the category of reviews but also on certain features that are not directly connected to the content. Building features of reviews normally involves text and natural language processing NLP. However, fake reviews may require building other features linked to the reviewer himself like for example review time/date or his writing styles. Thus the successful fake reviews detection lies on the construction of meaningful features extraction of the reviewers.

Usually, fake reviews detection depends not only on the category of reviews but also on certain features that are not directly connected to the content. Building features of reviews normally involves text and natural language processing NLP. However, fake reviews may require building other features linked to the reviewer himself like for example review time/date or his writing styles. Thus the successful fake reviews detection lies on the construction of meaningful features extraction of the reviewers.

**LITERATURE SURVEY**

**[1] R. Barbado, O. Araque, and C. A. Iglesias, “A framework for fake review detection in online consumer electronics retailers,” Information Processing & Management, vol. 56, no. 4, pp. 1234 – 1244, 2019.**

The impact of online reviews on businesses has grown significantly during last years, being crucial to determine business success in a wide array of sectors, ranging from restaurants, hotels to e-commerce. Unfortunately, some users use unethical means to improve their online reputation by writing fake reviews of their businesses or competitors. Previous research has addressed fake review detection in a number of domains, such as product or business reviews in restaurants and hotels. However, in spite of its economic interest, the domain of consumer electronics businesses has not yet been thoroughly studied. This article proposes a feature framework for detecting fake reviews that has been evaluated in the consumer electronics domain. The contributions are fourfold: (i) Construction of a dataset for classifying fake reviews in the consumer electronics domain in four different cities based on scraping techniques; (ii) definition of a feature framework for fake review detection; (iii) development of a fake review classification method based on the proposed framework and (iv) evaluation and analysis of the results for each of the cities under study. We have reached an 82% F-Score on the classification task and the Ada Boost classifier has been proven to be the best one by statistical means according to the Friedman test.

**Summary:** R. Barbado, O. Araque, and C. A. Iglesias proposes a feature framework for detecting fake reviews that has been evaluated in the consumer electronics domain**.**

**[2] Tadelis, “The economics of reputation and feedback systems in e-commerce marketplaces,” IEEE Internet Computing, vol. 20, no. 1, pp. 12–19, 2016.**

Online marketplaces have become ubiquitous, as sites such as eBay, Taobao, Uber, and Airbnb are frequented by billions of users. The success of these marketplaces is attributed to not only the ease in which buyers can find sellers, but also the trust that these marketplaces help facilitate through reputation and feedback systems. I begin by briefly describing the basic ideas surrounding the role of reputation in facilitating trust and trade, and offer an overview of how feedback and reputation systems work in online marketplaces. I then describe the literature that explores the effects of reputation and feedback systems on online marketplaces and highlight some of the problems of bias in feedback and reputation systems as they appear today. I discuss ways to address these problems to improve the practical design of online marketplaces and suggest some directions for future research.

**Summary**: Tadelis discuss ways to address these problems to improve the practical design of online marketplaces and suggest some directions for future research.

**[3] M. J. H. Mughal, “Data mining: Web data mining techniques, tools and algorithms: An overview,” Information Retrieval, vol. 9, no. 6, 2018.**

Web data mining became an easy and important platform for retrieval of useful information. Users prefer World Wide Web more to upload and download data. As increasing growth of data over the internet, it is getting difficult and time consuming for discovering informative knowledge and patterns. Digging knowledgeable and user queried information from unstructured and inconsistent data over the web is not an easy task to perform. Different mining techniques are used to fetch relevant information from web (hyperlinks, contents, web usage logs). Web data mining is a sub discipline of data mining which mainly deals with web. Web data mining is divided into three different types: web structure, web content and web usage mining. All these types use different techniques, tools, approaches, algorithms for discover information from huge bulks of data over the web.

**Summary**: M. J. H. Mughal concluded that using different techniques, tools, approaches, algorithms for discover information from huge bulks of data over the web performed well.

**[4] C. C. Aggarwal, “Opinion mining and sentiment analysis,” in Machine Learning for Text. Springer, 2018, pp. 413–434.**

The recent proliferation of social media has enabled users to post views about entities, individuals, events, and topics in a variety of formal and informal settings. Examples of such settings include reviews, forums, social media posts, blogs, and discussion boards. The problem of opinion mining and sentiment analysis is defined as the computational analytics associated with such text.

**Summary**: C. C. Aggarwaldefined problem of opinion mining and sentiment analysis as the computational analytics associated with such text.

**[5] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, “What yelp fake review filter might be doing?” in Seventh international AAAI conference on weblogs and social media, 2013.**

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 reviews. The results will be useful to other review hosting sites in their filtering effort. There are two main approaches to filtering: supervised and unsupervised learning. In terms of features used, there are also roughly two types: linguistic features and behavioral features. In this work, we will take a supervised approach as we can make use of Yelp’s filtered reviews for training. Existing approaches based on supervised learning are all based on pseudo fake reviews rather than fake reviews filtered by a commercial Web site. Recently, supervised learning using linguistic n-gram features has been shown to perform extremely well (attaining around 90% accuracy) in detecting crowdsourced fake reviews generated using Amazon Mechanical Turk (AMT). We put these existing research methods to the test and evaluate performance on the real-life Yelp data. To our surprise, the behavioral features perform very well, but the linguistic features are not as effective. To investigate, a novel information theoretic analysis is proposed to uncover the precise psycholinguistic difference between AMT reviews and Yelp reviews (crowdsourced vs. commercial fake reviews). We find something quite interesting. This analysis and experimental results allow us to postulate that Yelp’s filtering is reasonable and its filtering algorithm seems to be correlated with abnormal spamming behaviors.

**Summary:** A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance analysed experimental results allow us to postulate that Yelp’s filtering is reasonable and its filtering algorithm seems to be correlated with abnormal spamming behaviors.

**[6] N. Jindal and B. Liu, “Review spam detection,” in Proceedings of the 16th International Conference on World Wide Web, ser. WWW ’07, 2007.**

It is now a common practice for e-commerce Web sites to enable their customers to write reviews of products that they have purchased. Such reviews provide valuable sources of information on these products. They are used by potential customers to find opinions of existing users before deciding to purchase a product. They are also used by product manufacturers to identify problems of their products and to find competitive intelligence information about their competitors. Unfortunately, this importance of reviews also gives good incentive for spam, which contains false positive or malicious negative opinions. In this paper, we make an attempt to study review spam and spam detection. To the best of our knowledge, there is still no reported study on this problem.

**Summary**: They make an attempt to study review spam and spam detection. To the best of our knowledge, there is still no reported study on this problem.

**SYSTEM ANALYSIS & FEASIBILITY STUDY**

**Existing Method:**

The increasing growth of machine learning, computer techniques divided into traditional methods and machine learning methods. This section describes the related works of fake review detections and how machine learning methods are better than traditional methods. The existing method in this project have a certain flow and also RF is used for model development. But it requires large memory and result is not accurate.

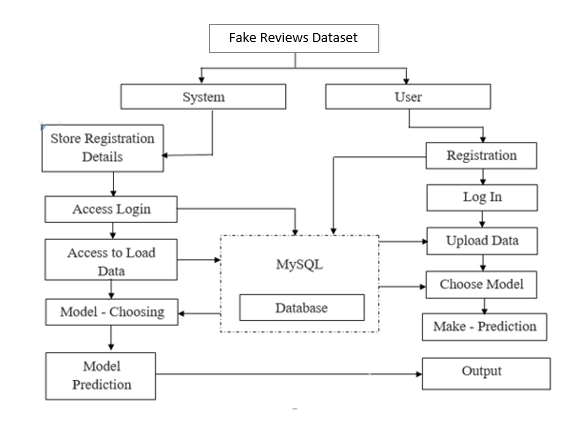
**Disadvantages:**

* Low Accuracy
* High complexity.
* Highly inefficient.
* Requires skilled persons

**Proposed System:**

We propose this application that can be considered a useful system since it helps to reduce the limitations obtained from traditional and other existing methods. The objective of this study to develop fast and reliable method which detects and estimates anaemia accurately. To design this system is we used a powerful algorithm in a based Python environment with Django frame work.

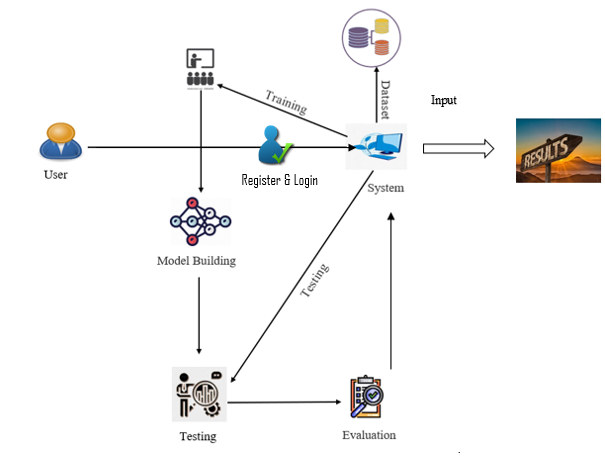
**Block Diagram:**



**Advantages**:

* Accuracy is good.
* Low complexity.
* Highly efficient.
* No need of skilled persons

**ARCHITECTURE**



Register & Login

**METHODOLOGY AND ALGORITHMS:**

**1. Naive Bayes:**

A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification task. The crux of the classifier is based on the Bayes theorem.

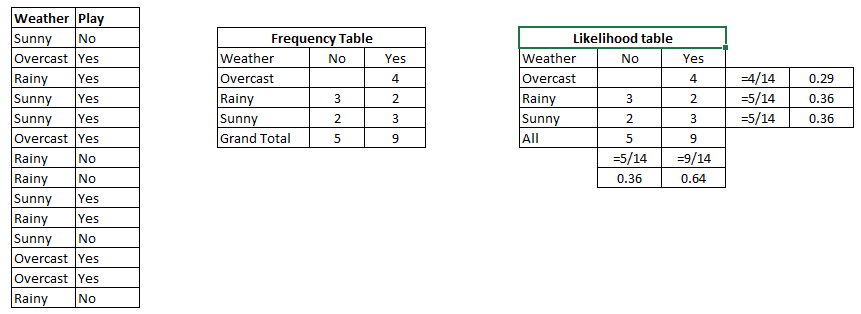


Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

Let’s understand it using an example. Below I have a training data set of weather and corresponding target variable ‘Play’ (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let’s follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.



Step 3: Now, use [Naive Bayesian](https://courses.analyticsvidhya.com/courses/naive-bayes?utm_source=blog&utm_medium=naive-bayes-explained) equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem**:**Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

P(Yes | Sunny) = P( Sunny | Yes) \* P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P (Sunny) = 5/14 = 0.36, P (Yes) = 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 \* 0.64 / 0.36 = 0.60, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

* It is easy and fast to predict class of test data set. It also perform well in multi class prediction
* When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
* It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

**Applications of Naive Bayes Algorithms**

**Real time Prediction:**Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.

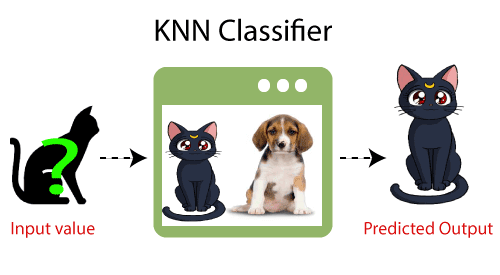
**Multi class Prediction:**This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.

**Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)

**Recommendation System:**Naive Bayes Classifier and [Collaborative Filtering](https://en.wikipedia.org/wiki/Collaborative_filtering) together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

KNN:

* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
* It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
* **Example**: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

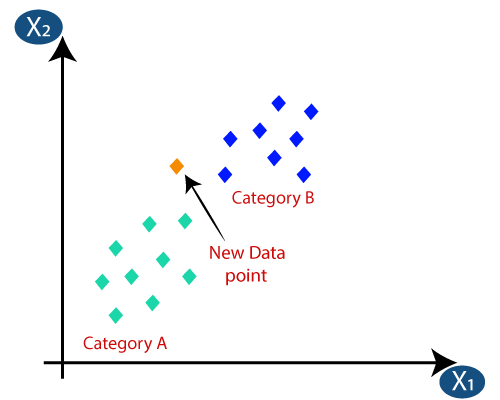


How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

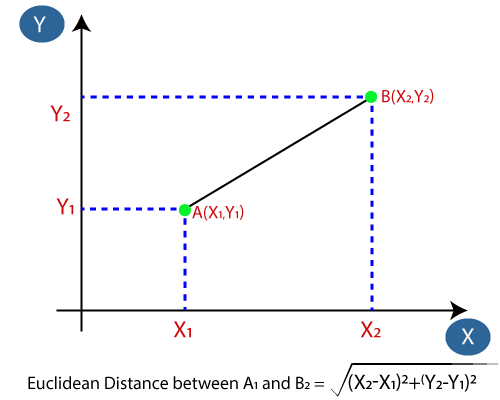
* **Step-1:** Select the number K of the neighbors
* **Step-2:**Calculate the Euclidean distance of K number of neighbors
* **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
* **Step-4:** Among these k neighbors, count the number of the data points in each category.
* **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
* **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:

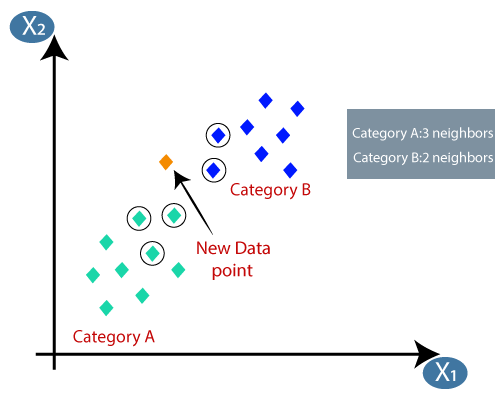


Firstly, we will choose the number of neighbors, so we will choose the k=5.

Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

How to select the value of K in the K-NN Algorithm?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

* There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
* A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
* Large values for K are good, but it may find some difficulties.

Advantages of KNN Algorithm:

* It is simple to implement.
* It is robust to the noisy training data
* It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

* Always needs to determine the value of K which may be complex some time.
* The computation cost is high because of calculating the distance between the data points for all the training samples.

**3. Logistic Regression:**

Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable (target) is categorical.

For example,

To predict whether an email is spam (1) or (0)

Whether the tumor is malignant (1) or not (0)

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time.

From this example, it can be inferred that linear regression is not suitable for classification problem. Linear regression is unbounded, and this brings logistic regression into picture. Their value strictly ranges from 0 to 1.

**Purpose and examples of logistic regression:**

Logistic regression is one of the most commonly used machine learning algorithms for binary classification problems, which are problems with two class values, including predictions such as “this or that,” “yes or no” and “A or B.”

The purpose of logistic regression is to estimate the probabilities of events, including determining a relationship between features and the probabilities of particular outcomes.

One example of this is predicting if a student will pass or fail an exam when the number of hours spent studying is provided as a feature and the variables for the response has two values: pass and fail.

Organizations can use insights from logistic regression outputs to enhance their business strategies so they can achieve their business goals, including reducing expenses or losses and increasing [ROI](https://searchcio.techtarget.com/definition/ROI) in marketing campaigns, for example.

An [e-commerce](https://searchcio.techtarget.com/definition/e-commerce) company that mails expensive promotional offers to customers would like to know whether a particular customer is likely to respond to the offers or not. For example, they’ll want to know whether that consumer will be a “responder” or a “non responder.” In marketing, this is called propensity to respond modeling.

Likewise, a credit card company develops a model to decide whether to issue a credit card to a customer or not will try to predict whether the customer is going to default or not on the credit card based on such characteristics as annual income, monthly credit card payments and number of defaults. In banking parlance, this is known as default propensity modeling.

**Uses of logistic regression:**

Logistic regression has become particularly popular in online advertising, enabling marketers to predict the likelihood of specific website users who will click on particular advertisements as a yes or no percentage.

* Logistic regression can also be used in:
* Healthcare to identify risk factors for diseases and plan preventive measures.
* Weather forecasting [apps](https://searchsoftwarequality.techtarget.com/definition/application) to predict snowfall and weather conditions.
* Voting apps to determine if voters will vote for a particular candidate.

Insurance to predict the chances that a policy holder will die before the term of the policy expires based on certain criteria, such as gender, age and physical examination.

Banking to predict the chances that a loan applicant will default on a loan or not, based on annual income, past defaults and past debts.

**Logistic regression vs. linear regression:**

The main difference between logistic regression and linear regression is that logistic regression provides a constant output, while linear regression provides a continuous output.

In logistic regression, the outcome, such as a dependent variable, only has a limited number of possible values. However, in linear regression, the outcome is continuous, which means that it can have any one of an infinite number of possible values.

Logistic regression is used when the response variable is categorical, such as yes/no, true/false and pass/fail. Linear regression is used when the response variable is continuous, such as number of hours, height and weight.

For example, given data on the time a student spent studying and that student’s exam scores, logistic regression and linear regression can predict different things.

With logistic regression predictions, only specific values or categories are allowed. Therefore, logistic regression can predict whether the student passed or failed. Since linear regression predictions are continuous, such as numbers in a range, it can predict the student’s test score on a scale of 0 -100.

**SOFTWARE DEVELOPMENT LIFE CYCLE – SDLC:**

In our project we use waterfall model as our software development cycle because of its step-by-step procedure while implementing.



**Fig1**: Waterfall Model

* **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
* **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
* **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

**FEASIBILITY STUDY**

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**Economic feasibility:**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **Technical feasibility:**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**Social feasibility:**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**SYSTEM REQUIREMENTS SPECIFICATION**

**Functional and non-functional requirements:**

Requirement’s analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

**Functional Requirements**: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1. Authentication of user whenever he/she logs into the system
2. System shutdown in case of a cyber-attack
3. A verification email is sent to user whenever he/she register for the first time on some software system.

**Non-functional requirements**: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.  
They basically deal with issues like:

* Portability
* Security
* Maintainability
* Reliability
* Scalability
* Performance
* Reusability
* Flexibility

Examples of non-functional requirements:

1. Emails should be sent with a latency of no greater than 12 hours from such an activity.
2. The processing of each request should be done within 10 seconds
3. The site should load in 3 seconds whenever of simultaneous users are > 10000

**SYSTEM SPECIFICATIONS:**

# **H/W SPECIFICATIONS:**

# Processor - I3/Intel Processor

# RAM - 8GB (min)

* Hard Disk - 128 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - Any

**S/W SPECIFICATIONS:**

* Operating System : Windows 10
* Server-side Script : Python 3.6
* IDE : PyCharm
* Libraries Used : Pandas, NumPy, Scikit-Learn.
* Frame Work : Django
* Data Base : MySql

**SYSTEM DESIGN:**

## **Input Design:**

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

* It should serve specific purpose effectively such as storing, recording, and retrieving the information.
* It ensures proper completion with accuracy.
* It should be easy to fill and straightforward.
* It should focus on user’s attention, consistency, and simplicity.
* All these objectives are obtained using the knowledge of basic design principles regarding −
  + What are the inputs needed for the system?
  + How end users respond to different elements of forms and screens.

### **Objectives for Input Design:**

The objectives of input design are −

* To design data entry and input procedures
* To reduce input volume
* To design source documents for data capture or devise other data capture methods
* To design input data records, data entry screens, user interface screens, etc.
* To use validation checks and develop effective input controls.

**Output Design:**

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

### **Objectives of Output Design:**

The objectives of input design are:

* To develop output design that serves the intended purpose and eliminates the production of unwanted output.
* To develop the output design that meets the end user’s requirements.
* To deliver the appropriate quantity of output.
* To form the output in appropriate format and direct it to the right person.
* To make the output available on time for making good decisions.

**MODULES:**

**1. System**

**1.1 Take Dataset:**

* The System receives data given by the user.

**1.2 Model Training:**

* The user can select model based on the variables to get response from the data and can predict the outputs from the model.

**1.3 Generate Results:**

* The system can deliver the predicted results and can be displayed to the user.

**2. User**

**2.1 Registration:**

* Every new user needs to register themselves in the system with a unique name and email.

**2.2 Login:**

* Registered users can log in to the system.

**2.3 Upload and View Data:**

* After logging in successfully, the user can upload and view their uploaded dataset.

**2.4 Splits and Select Model:**

* The user can select the desired split ratio and can choose the required model to view the result from the dataset.

**2.5 Prediction:**

* The user can predict outcomes from the system.

**2.6 Logout**

* User logout from the system

**UML DIAGRAMS**

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

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 Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. 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:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**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.



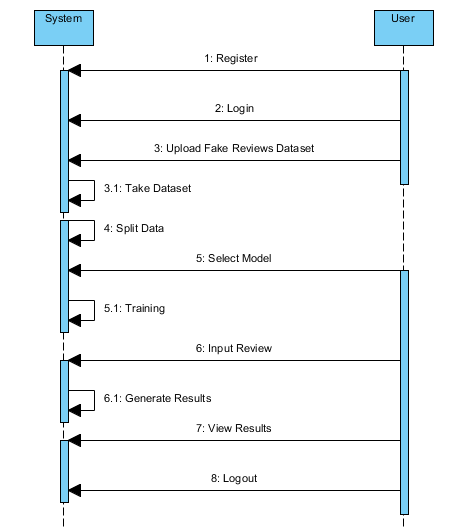
**CLASS DIAGRAM**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information



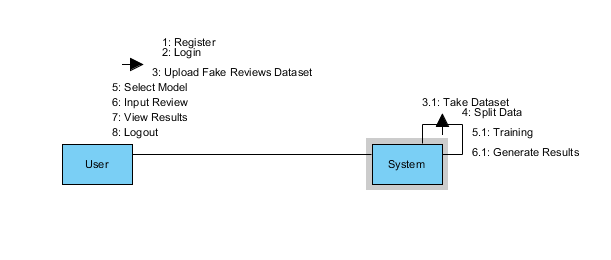
**SEQUENCE DIAGRAM**

* 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. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



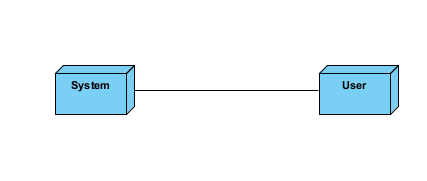
**COLLABORATION DIAGRAM:**

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.



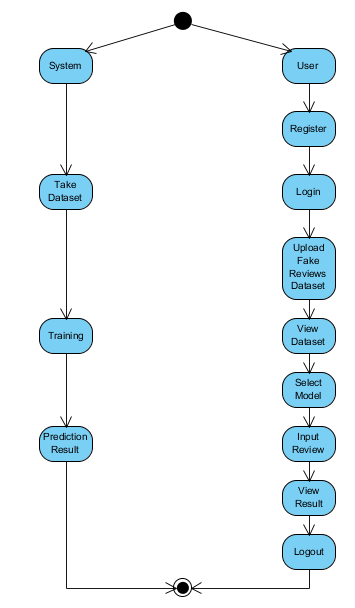
**DEPLOYMENT DIAGRAM**

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware’s used to deploy the application.



**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**COMPONENT DIAGRAM**:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.



**ER DIAGRAM:**

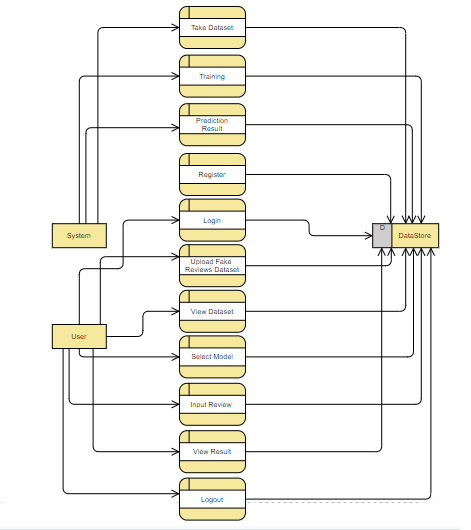
An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

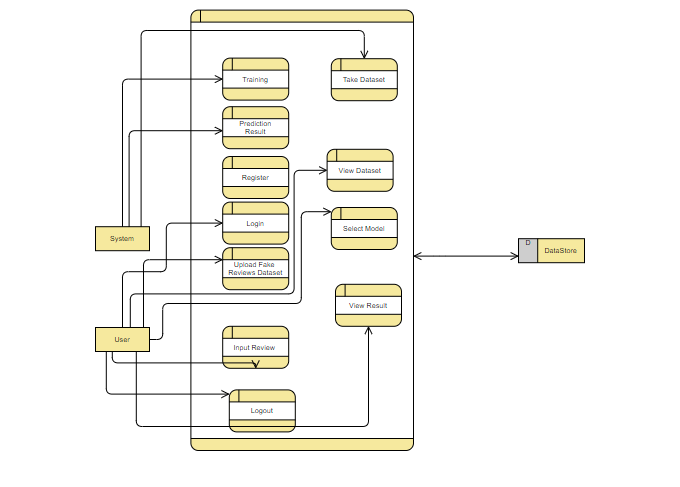
An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let’s have a look at a simple ER diagram to understand this concept.

# Picture 30

**DFD DIAGRAM:**

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

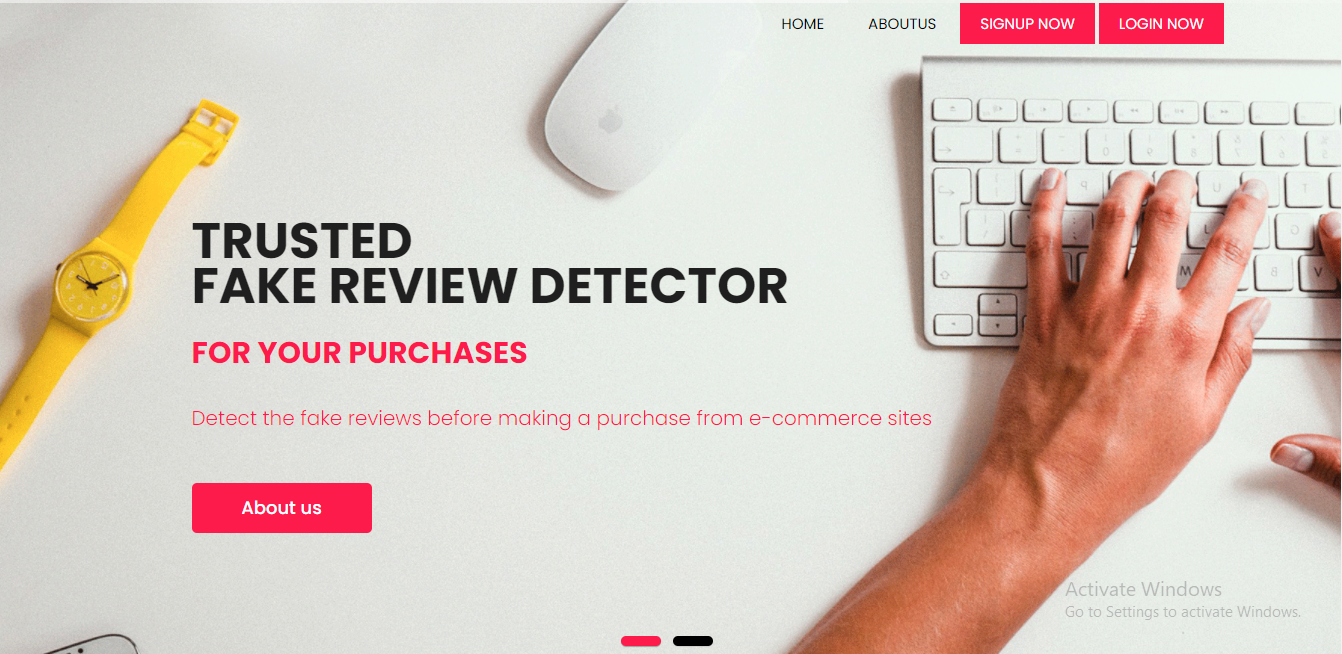




**OUTPUT SCREEN SHOTS WITH DESCRIPTION.**

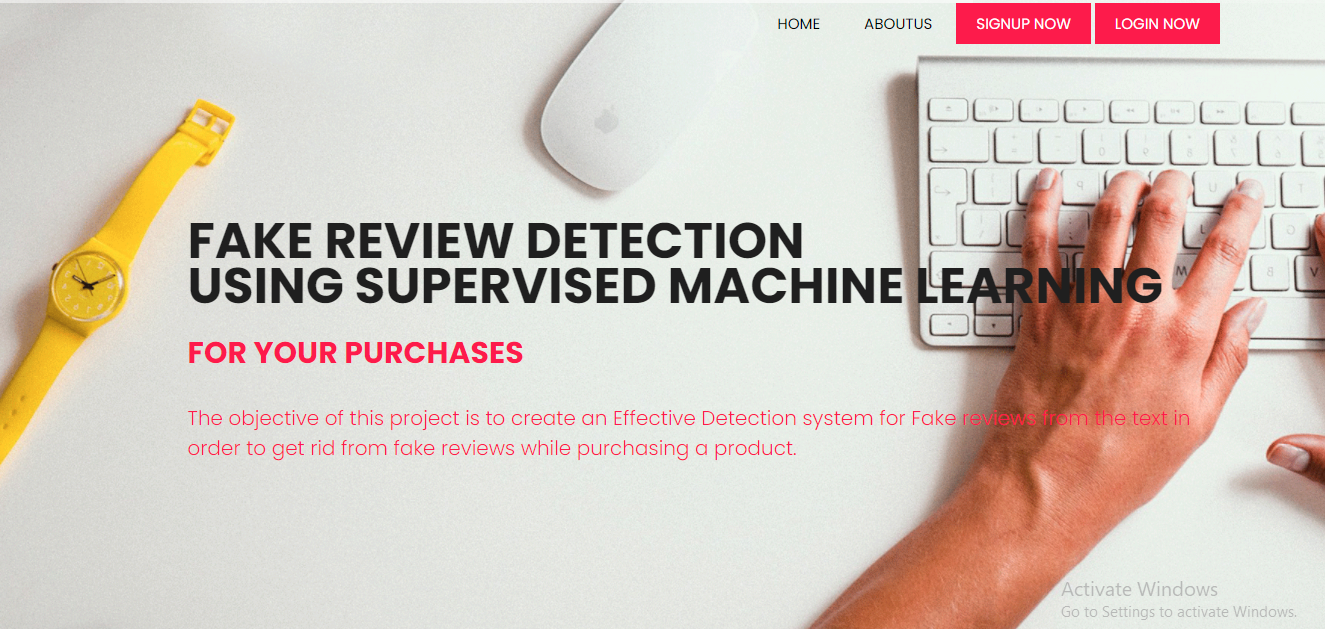
**Home:**

In our project, we are detecting the fake reviews from the review entered by the user.



**Fig1**: Home

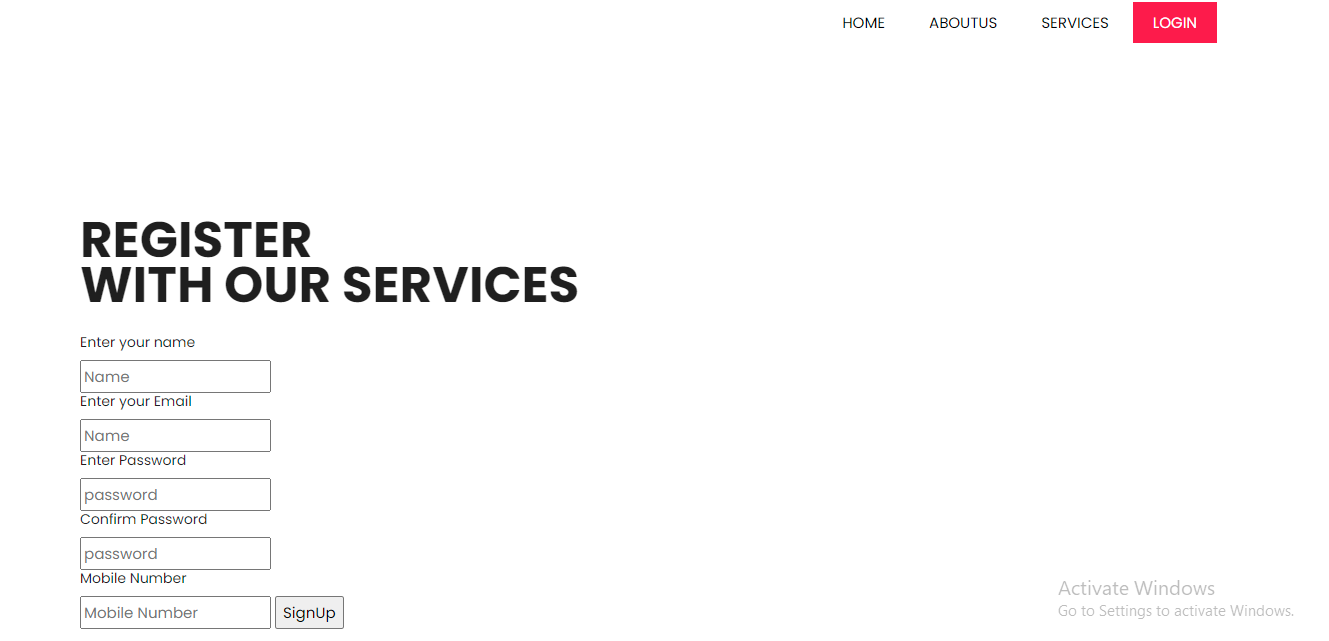
Home Page of our project

**About:** 

**Fig2**: About

Here the application describes what main objective of this project is.

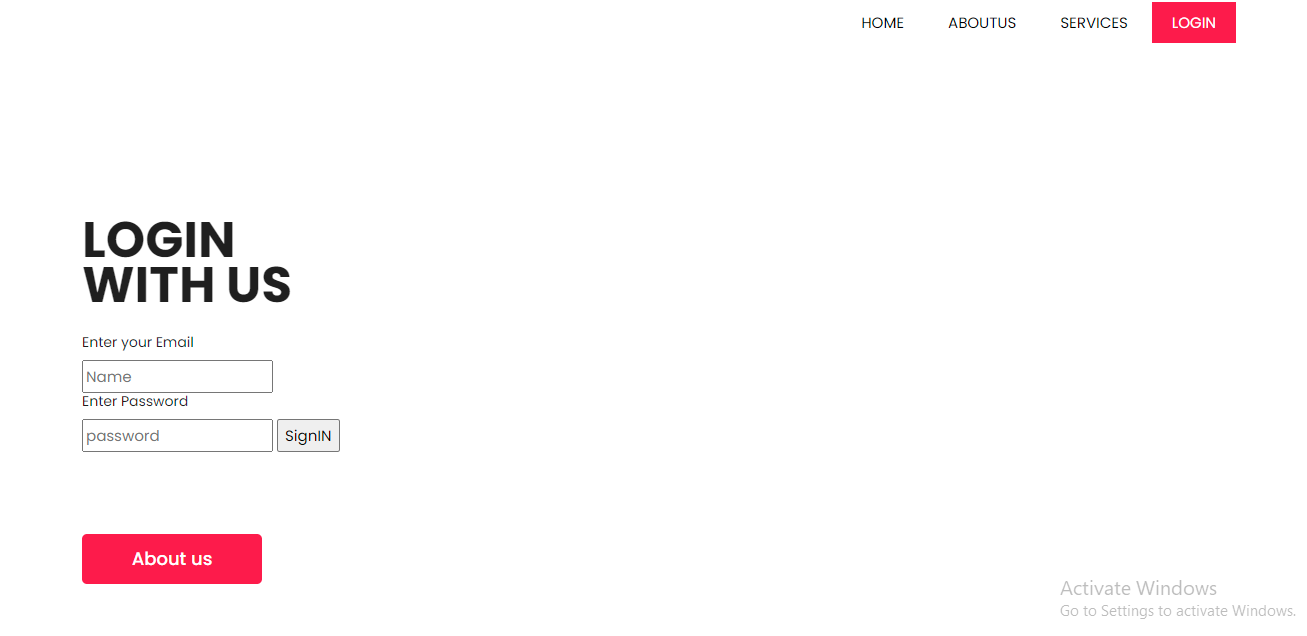
**Registration**:



**Fig3**: Registration

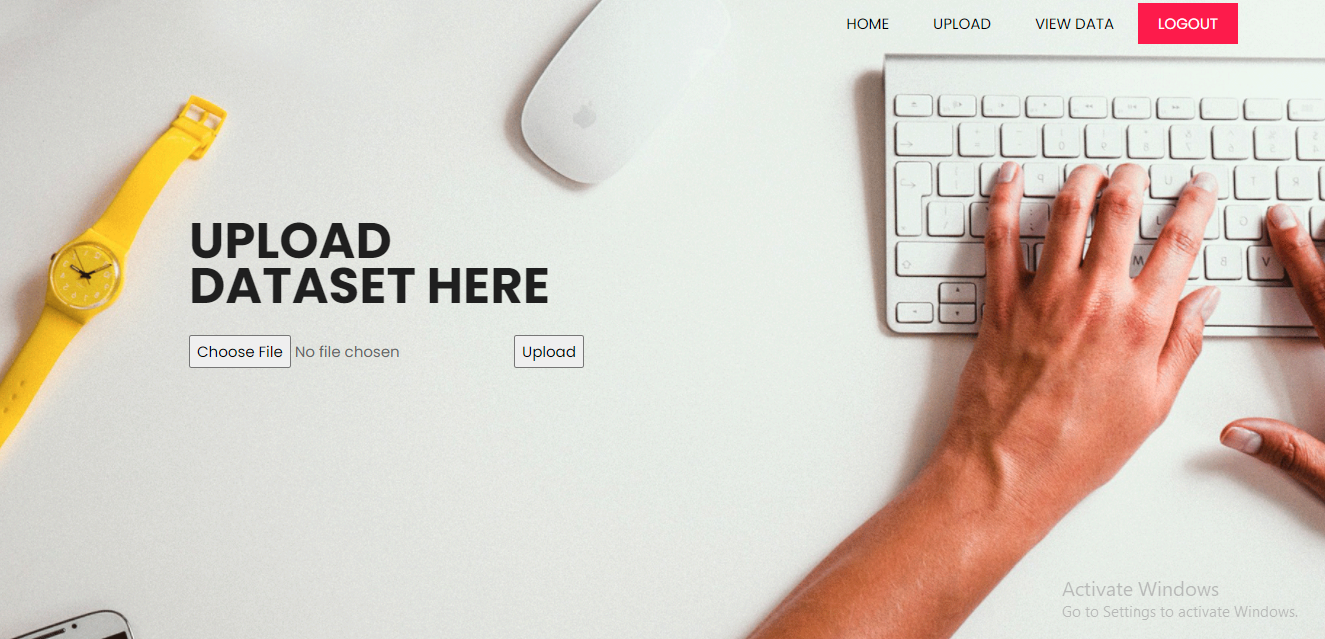
Registration page in which user need to register to start.

**Login:**



**Fig4**: Login Page

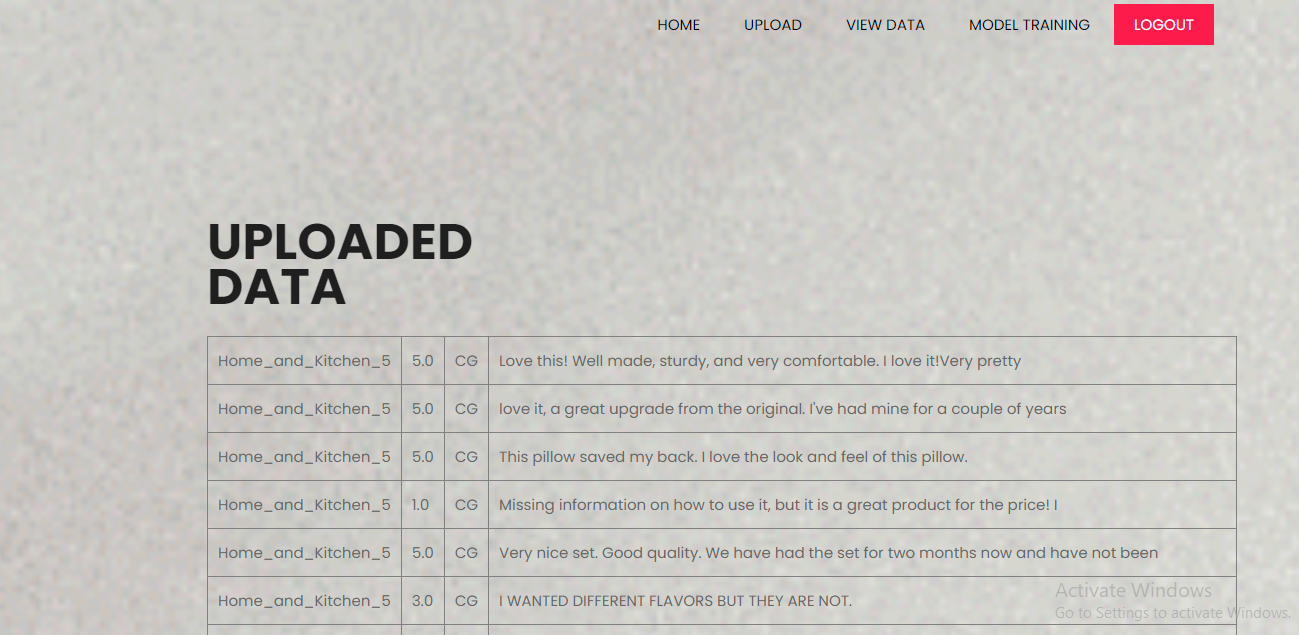
Login page, user need to enter valid credentials in order to enter.



**Fig5**: Upload Page

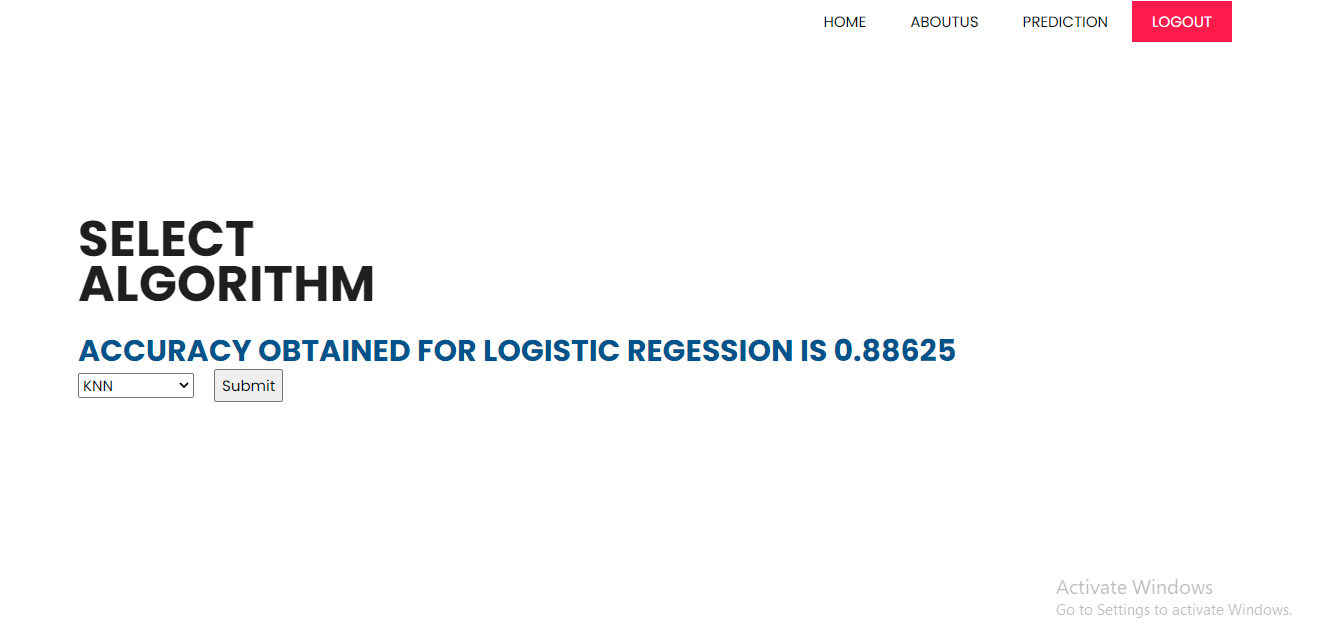
Upload Page in order to upload the dataset.

**View Data:**



**Fig6**: View Dataset Page

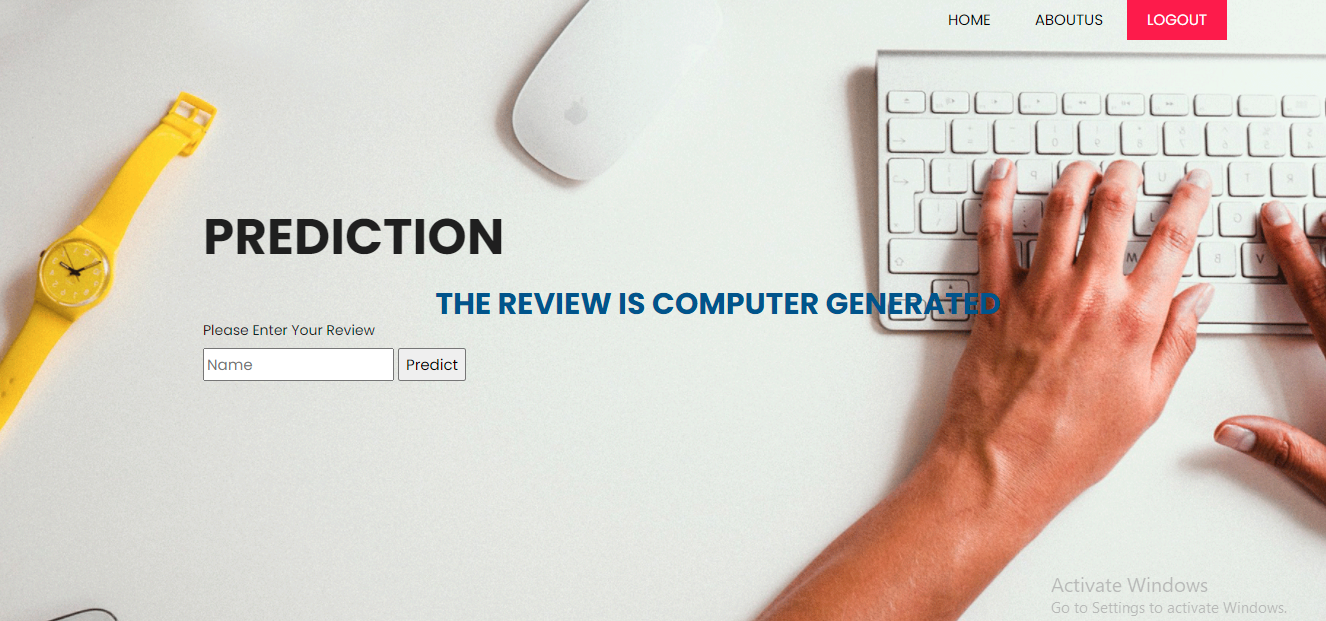
User views the data which he was uploaded to the system.



**Fig7**: Model Training

Here training of your model takes place and display the model’s accuracy

**Prediction:**



**Fig6**: Prediction Page

User need to enter the required fields in order to get the response from the data whether the review is computer generated or original.

**TEST CASES:**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Result** |
| Input text | Tested for whether the input tells us whether the review is fake or not | Success |

**Test cases Model building:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Test cases** | **I/O** | **Expected O/T** | **Actual O/T** | **P/F** |
| 1 | Read the dataset. | Dataset. | Dataset need to read successfully. | Dataset fetched successfully. | P |
| 2 | Performing pre-processing on the dataset | Pre-processing part takes place | Pre-processing should be performed on dataset | Pre-processing successfully completed. | P |
| 3 | Model Building | Model Building for the clean data | Need to create model using required algorithms | Model Created Successfully. | P |
| 4 | Fake Review Estimation | Input fields provided. | Output should be whether review is fake or not | Model predicted as review is fake | P |
| 5 | Anaemia Estimation | Input fields provided. | Output should be whether fake or not | Model predicted as review is not fake | P |

**CONCLUSION:**

We have successfully developed a system to detect fake reviews in this application. This is created in a user-friendly environment with Python programming and Django framework. The system is likely to gather data from the user in order to determine whether the review is fake or not.

**FUTURE SCOPE**

The ability to identify numerous classifications could be added to this application in the future. We intend to investigate prediction approach with the revised data set and employ the most accurate and relevant machine learning algorithms for detection.

**REFERENCES:**

[1] R. Barbado, O. Araque, and C. A. Iglesias, “A framework for fake review detection in online consumer electronics retailers,” Information Processing & Management, vol. 56, no. 4, pp. 1234 – 1244, 2019.

[2] S. Tadelis, “The economics of reputation and feedback systems in e-commerce marketplaces,” IEEE Internet Computing, vol. 20, no. 1, pp. 12–19, 2016.

[3] M. J. H. Mughal, “Data mining: Web data mining techniques, tools and algorithms: An overview,” Information Retrieval, vol. 9, no. 6, 2018.

[4] C. C. Aggarwal, “Opinion mining and sentiment analysis,” in Machine Learning for Text. Springer, 2018, pp. 413–434.

[5] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, “What yelp fake review filter might be doing?” in Seventh international AAAI conference on weblogs and social media, 2013.

[6] N. Jindal and B. Liu, “Review spam detection,” in Proceedings of the 16th International Conference on World Wide Web, ser. WWW ’07, 2007.

[7] E. Elmurngi and A. Gherbi, Detecting Fake Reviews through Sentiment Analysis Using Machine Learning Techniques. IARIA/DATA ANALYTICS, 2017.

[8] V. Singh, R. Piryani, A. Uddin, and P. Waila, “Sentiment analysis of movie reviews and blog posts,” in Advance Computing Conference (IACC), 2013, pp. 893–898.

[9] A. Molla, Y. Biadgie, and K.-A. Sohn, “Detecting Negative Deceptive Opinion from Tweets.” in International Conference on Mobile and Wireless Technology. Singapore: Springer, 2017.

[10] S. Shojaee et al., “Detecting deceptive reviews using lexical and syntactic features.” 2013. [11] Y. Ren and D. Ji, “Neural networks for deceptive opinion spam detection: An empirical study,” Information Sciences, vol. 385, pp. 213– 224, 2017.

[12] H. Li et al., “Spotting fake reviews via collective positive-unlabeled learning.” 2014.

**BIBLIOGRAPHY:**

**SOFTWARE INSTALLATION FOR MACHINE LEARNING PROJECTS:**

**Installing Python:**

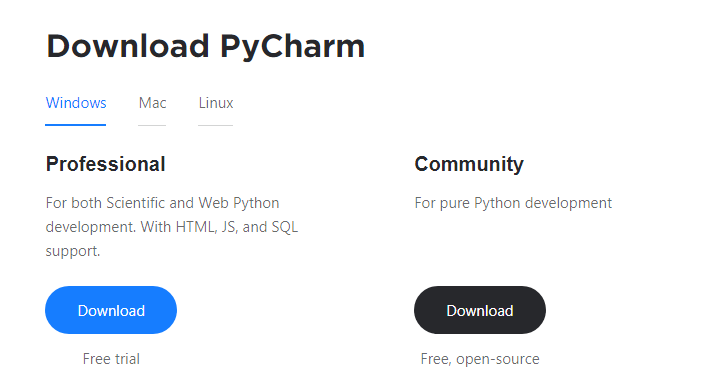
1. To download and install Python visit the official website of Python <https://www.python.org/downloads/> and choose your version.



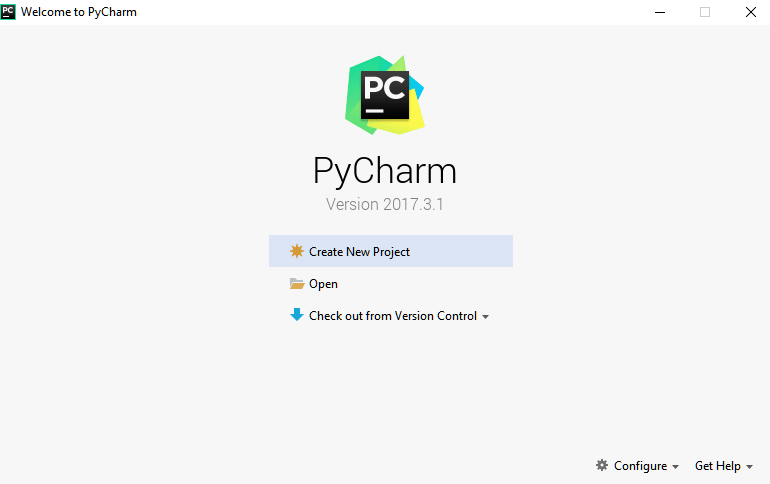
1. Once the download is complete, run the exe for install Python. Now click on Install Now.
2. You can see Python installing at this point.
3. When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

**Installing PyCharm:**

1. To download PyCharm visit the website <https://www.jetbrains.com/pycharm/download/> and click the "DOWNLOAD" link under the Community Section.



1. Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.
2. On the next screen, Change the installation path if required. Click “Next”.
3. On the next screen, you can create a desktop shortcut if you want and click on “Next”.
4. Choose the start menu folder. Keep selected JetBrains and click on “Install”.
5. Wait for the installation to finish.
6. Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.
7. After you click on "Finish," the Following screen will appear.

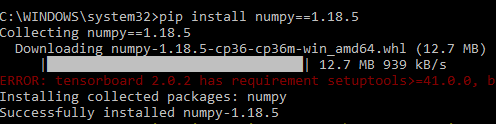


9. You need to install some packages to execute your project in a proper way.

10. Open the command prompt/ anaconda prompt or terminal as administrator.

11. The prompt will get open, with specified path, type “pip install package name” which you want to install (like NumPy, pandas, sea born, scikit-learn, Matplotlib, Pyplot)

Ex: Pip install NumPy



# **INTRODUCTION TO PYTHON**

* Python

### What Is a Script?

Up to this point, I have concentrated on the interactive programming capability of Python.  This is a very useful capability that allows you to type in a program and to have it executed immediately in an interactive mode

Scripts are reusable

Basically, a script is a text file containing the statements that comprise a Python program.  Once you have created the script, you can execute it over and over without having to retype it each time.

Scripts are editable

Perhaps, more importantly, you can make different versions of the script by modifying the statements from one file to the next using a text editor.  Then you can execute each of the individual versions.  In this way, it is easy to create different programs with a minimum amount of typing.

You will need a text editor

Just about any text editor will suffice for creating Python script files.

You can use *Microsoft Notepad, Microsoft WordPad, Microsoft Word,*or just about any word processor if you want to.

Difference between a script and a program

Script:

Scripts are distinct from the core code of the application, which is usually written in a different language, and are often created or at least modified by the end-user. Scripts are often interpreted from source code or byte code, whereas the applications they control are traditionally compiled to native machine code.

Program:

The program has an executable form that the computer can use directly to execute the instructions.

The same program in its human-readable source code form, from which executable programs are derived (e.g., compiled)

Python

What is Python? Chances you are asking yourself this. You may have found this book because you want to learn to program but don’t know anything about programming languages. Or you may have heard of programming languages like C, C++, C#, or Java and want to know what Python is and how it compares to “big name” languages. Hopefully I can explain it for you.

Python concepts

If you’re not interested in the how’s and whys of Python, feel free to skip to the next chapter. In this chapter I will try to explain to the reader why I think Python is one of the best languages available and why it’s a great one to start programming with.

• Open-source general-purpose language.

• Object Oriented, Procedural, Functional

• Easy to interface with C/ObjC/Java/Fortran

• Easy-is to interface with C++ (via SWIG)

• Great interactive environment

• Great interactive environment

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* 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 is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Possum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and UNIX shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Possum still holds a vital role in directing its progress.

Python Features

Python's features include −

* Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* Easy-to-read − Python code is more clearly defined and visible to the eyes.
* Easy-to-maintain − Python's source code is fairly easy-to-maintained.
* A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* Extendable − you can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* Databases − Python provides interfaces to all major commercial databases.
* GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* Scalable − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Dynamic vs. Static

Types Python is a dynamic-typed language. Many other languages are static typed, such as C/C++ and Java. A static typed language requires the programmer to explicitly tell the computer what type of “thing” each data value is.

For example, in C if you had a variable that was to contain the price of something, you would have to declare the variable as a “float” type.

This tells the compiler that the only data that can be used for that variable must be a floating point number, i.e. a number with a decimal point.

If any other data value was assigned to that variable, the compiler would give an error when trying to compile the program.

Python, however, doesn’t require this. You simply give your variables names and assign values to them. The interpreter takes care of keeping track of what kinds of objects your program is using. This also means that you can change the size of the values as you develop the program. Say you have another decimal number (a.k.a. a floating point number) you need in your program.

With a static typed language, you have to decide the memory size the variable can take when you first initialize that variable. A double is a floating point value that can handle a much larger number than a normal float (the actual memory sizes depend on the operating environment).

If you declare a variable to be a float but later on assign a value that is too big to it, your program will fail; you will have to go back and change that variable to be a double.

With Python, it doesn’t matter. You simply give it whatever number you want and Python will take care of manipulating it as needed. It even works for derived values.

For example, say you are dividing two numbers. One is a floating point number and one is an integer. Python realizes that it’s more accurate to keep track of decimals so it automatically calculates the result as a floating point number

Variables

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Standard Data Types

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

Python has five standard data types −

* Numbers
* String
* List
* Tuple
* Dictionary

## Python Numbers

Number data types store numeric values. Number objects are created when you assign a value to them

## Python Strings

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

## Python Lists

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.

The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (\*) is the repetition operator.

## Python Tuples

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

The main differences between lists and tuples are: Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses (( )) and cannot be updated. Tuples can be thought of as read-only lists.

## Python Dictionary

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

Different modes in python

Python has two basic modes: normal and interactive.

The normal mode is the mode where the scripted and finished .pie files are run in the Python interpreter.

Interactive mode is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory. As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole

# 20 Python libraries

1. Requests. The most famous http library written by Kenneth remits. It’s a must have for every python developer.

2. Scrappy. If you are involved in web scraping then this is a must have library for you. After using this library you won’t use any other.

3. Python. A guy toolkit for python. I have primarily used it in place of tinder. You will really love it.

4. Pillow. A friendly fork of PIL (Python Imaging Library). It is more user friendly than PIL and is a must have for anyone who works with images.

5. SQL Alchemy. A database library. Many love it and many hate it. The choice is yours.

6. Beautiful Soup. I know it’s slow but this xml and html parsing library is very useful for beginners.

7. Twisted. The most important tool for any network application developer. It has a very beautiful ape and is used by a lot of famous python developers.

8. Numbly. How can we leave this very important library? It provides some advance math functionalities to python.

9. Skippy. When we talk about numbly then we have to talk about spicy. It is a library of algorithms and mathematical tools for python and has caused many scientists to switch from ruby to python.

10. Matplotlib. A numerical plotting library. It is very useful for any data scientist or any data analyzer.

11. Pygmy. Which developer does not like to play games and develop them? This library will help you achieve your goal of 2d game development.

12. Piglet. A 3d animation and game creation engine. This is the engine in which the famous [python port](https://github.com/fogleman/Minecraft) of mine craft was made

13. Pit. A GUI toolkit for python. It is my second choice after python for developing GUI’s for my python scripts.

14. Pit. Another python GUI library. It is the same library in which the famous Bit torrent client is created.

15. Scaly. A packet sniffer and analyzer for python made in python.

16. Pywin32. A python library which provides some useful methods and classes for interacting with windows.

17. Notch. Natural Language Toolkit – I realize most people won’t be using this one, but it’s generic enough. It is a very useful library if you want to manipulate strings. But its capacity is beyond that. Do check it out.

18. Nose. A testing framework for python. It is used by millions of python developers. It is a must have if you do test driven development.

19. Simply. Simply can do algebraic evaluation, differentiation, expansion, complex numbers, etc. It is contained in a pure Python distribution.

20. I Python. I just can’t stress enough how useful this tool is. It is a python prompt on steroids. It has completion, history, shell capabilities, and a lot more. Make sure that you take a look at it.

Numpy

Humpy’s main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In numbly dimensions are called *axes*. The number of axes is *rank*.

• Offers Matlab-ish capabilities within Python

• Fast array operations

• 2D arrays, multi-D arrays, linear algebra etc.

Matplotlib

• High quality plotting library.

Python class and objects

These are the building blocks of OOP. Class creates a new object. This object can be anything, whether an abstract data concept or a model of a physical object, e.g. a chair. Each class has individual characteristics unique to that class, including variables and methods. Classes are very powerful and currently “the big thing” in most programming languages. Hence, there are several chapters dedicated to OOP later in the book.

The class is the most basic component of object-oriented programming. Previously, you learned how to use functions to make your program do something.

Now will move into the big, scary world of Object-Oriented Programming (OOP). To be honest, it took me several months to get a handle on objects.

When I first learned C and C++, I did great; functions just made sense for me.

Having messed around with BASIC in the early ’90s, I realized functions were just like subroutines so there wasn’t much new to learn.

However, when my C++ course started talking about objects, classes, and all the new features of OOP, my grades definitely suffered.

Once you learn OOP, you’ll realize that it’s actually a pretty powerful tool. Plus many Python libraries and APIs use classes, so you should at least be able to understand what the code is doing.

One thing to note about Python and OOP: it’s not mandatory to use objects in your code in a way that works best; maybe you don’t need to have a full-blown class with initialization code and methods to just return a calculation. With Python, you can get as technical as you want.

As you’ve already seen, Python can do just fine with functions. Unlike languages such as Java, you aren’t tied down to a single way of doing things; you can mix functions and classes as necessary in the same program. This lets you build the code

Objects are an encapsulation of variables and functions into a single entity. Objects get their variables and functions from classes. Classes are essentially a template to create your objects.

Here’s a brief list of Python OOP ideas:

• The class statement creates a class object and gives it a name. This creates a new namespace.

• Assignments within the class create class attributes. These attributes are accessed by qualifying the name using dot syntax: ClassName.Attribute.

• Class attributes export the state of an object and its associated behavior. These attributes are shared by all instances of a class.

• Calling a class (just like a function) creates a new instance of the class.

This is where the multiple copies part comes in.

• Each instance gets ("inherits") the default class attributes and gets its own namespace. This prevents instance objects from overlapping and confusing the program.

• Using the term self identifies a particular instance, allowing for per-instance attributes. This allows items such as variables to be associated with a particular instance.

Inheritance

First off, classes allow you to modify a program without really making changes to it.

To elaborate, by sub classing a class, you can change the behaviour of the program by simply adding new components to it rather than rewriting the existing components.

As we’ve seen, an instance of a class inherits the attributes of that class.

However, classes can also inherit attributes from other classes. Hence, a subclass inherits from a superclass allowing you to make a generic superclass that is specialized via subclasses.

The subclasses can override the logic in a superclass, allowing you to change the behavior of your classes without changing the superclass at all.

Operator Overloads

Operator overloading simply means that objects that you create from classes can respond to actions (operations) that are already defined within Python, such as addition, slicing, printing, etc.

Even though these actions can be implemented via class methods, using overloading ties the behavior closer to Python’s object model and the object interfaces are more consistent to Python’s built-in objects, hence overloading is easier to learn and use.

User-made classes can override nearly all of Python’s built-in operation methods

Exceptions

I’ve talked about exceptions before but now I will talk about them in depth. Essentially, exceptions are events that modify program’s flow, either intentionally or due to errors.

They are special events that can occur due to an error, e.g. trying to open a file that doesn’t exist, or when the program reaches a marker, such as the completion of a loop.

Exceptions, by definition, don’t occur very often; hence, they are the "exception to the rule" and a special class has been created for them. Exceptions are everywhere in Python.

Virtually every module in the standard Python library uses them, and Python itself will raise them in a lot of different circumstances.

Here are just a few examples:

• Accessing a non−existent dictionary key will raise a Key Error exception.

• Searching a list for a non−existent value will raise a Value Error exception

. • Calling a non−existent method will raise an Attribute Error exception.

• Referencing a non−existent variable will raise a Name Error exception.

• Mixing data types without coercion will raise a Type Error exception.

One use of exceptions is to catch a fault and allow the program to continue working; we have seen this before when we talked about files.

This is the most common way to use exceptions. When programming with the Python command line interpreter, you don’t need to worry about catching exceptions.

Your program is usually short enough to not be hurt too much if an exception occurs.

Plus, having the exception occur at the command line is a quick and easy way to tell if your code logic has a problem.

However, if the same error occurred in your real program, it will fail and stop working. Exceptions can be created manually in the code by raising an exception.

It operates exactly as a system-caused exceptions, except that the programmer is doing it on purpose. This can be for a number of reasons. One of the benefits of using exceptions is that, by their nature, they don’t put any overhead on the code processing.

Because exceptions aren’t supposed to happen very often, they aren’t processed until they occur.

Exceptions can be thought of as a special form of the if/elf statements. You can realistically do the same thing with if blocks as you can with exceptions.

However, as already mentioned, exceptions aren’t processed until they occur; if blocks are processed all the time.

Proper use of exceptions can help the performance of your program.

The more infrequent the error might occur, the better off you are to use exceptions; using if blocks requires Python to always test extra conditions before continuing.

Exceptions also make code management easier: if your programming logic is mixed in with error-handling if statements, it can be difficult to read, modify, and debug your program.

User-Defined Exceptions

I won’t spend too much time talking about this, but Python does allow for a programmer to create his own exceptions.

You probably won’t have to do this very often but it’s nice to have the option when necessary.

However, before making your own exceptions, make sure there isn’t one of the built-in exceptions that will work for you.

They have been "tested by fire" over the years and not only work effectively, they have been optimized for performance and are bug-free.

Making your own exceptions involves object-oriented programming, which will be covered in the next chapter

. To make a custom exception, the programmer determines which base exception to use as the class to inherit from, e.g. making an exception for negative numbers or one for imaginary numbers would probably fall under the Arithmetic Error exception class.

To make a custom exception, simply inherit the base exception and define what it will do.

Python modules

Python allows us to store our code in files (also called modules). This is very useful for more serious programming, where we do not want to retype a long function definition from the very beginning just to change one mistake. In doing this, we are essentially defining our own modules, just like the modules defined already in the Python library.

To support this, Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a *module*; definitions from a module can be *imported* into other modules or into the *main* module.

Testing code

As indicated above, code is usually developed in a file using an editor.

To test the code, import it into a Python session and try to run it.

Usually there is an error, so you go back to the file, make a correction, and test again.

This process is repeated until you are satisfied that the code works. T

His entire process is known as the development cycle.

There are two types of errors that you will encounter. Syntax errors occur when the form of some command is invalid.

This happens when you make typing errors such as misspellings, or call something by the wrong name, and for many other reasons. Python will always give an error message for a syntax error.

Functions in Python

It is possible, and very useful, to define our own functions in Python. Generally speaking, if you need to do a calculation only once, then use the interpreter. But when you or others have need to perform a certain type of calculation many times, then define a function.

You use functions in programming to bundle a set of instructions that you want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed. That means that a function is a piece of code written to carry out a specified task.

## To carry out that specific task, the function might or might not need multiple inputs. When the task is carved out, the function can or cannot return one or more values.

## There are three types of functions in python:

## Help (), min (), print ().

Namespaces in Python are implemented as Python dictionaries, this means it is a mapping from names (keys) to objects (values). The user doesn't have to know this to write a Python program and when using namespaces.

Some namespaces in Python:

* global names of a module
* local names in a function or method invocation
* built-in names: this namespace contains built-in functions (e.g. abs(), camp(), ...) and built-in exception names

Garbage Collection

Garbage Collector exposes the underlying memory management mechanism of Python, the automatic garbage collector. The module includes functions for controlling how the collector operates and to examine the objects known to the system, either pending collection or stuck in reference cycles and unable to be freed.

Python XML Parser

XML is a portable, open source language that allows programmers to develop applications that can be read by other applications, regardless of operating system and/or developmental language.

What is XML? The Extensible Markup Language XML is a markup language much like HTML or SGML.

This is recommended by the World Wide Web Consortium and available as an open standard.

XML is extremely useful for keeping track of small to medium amounts of data without requiring a SQL-based backbone.

XML Parser Architectures and APIs the Python standard library provides a minimal but useful set of interfaces to work with XML.

The two most basic and broadly used APIs to XML data are the SAX and DOM interfaces.

Simple API for XML SAX: Here, you register callbacks for events of interest and then let the parser proceed through the document.

This is useful when your documents are large or you have memory limitations, it parses the file as it reads it from disk and the entire file is never stored in memory.

Document Object Model DOM API : This is a World Wide Web Consortium recommendation wherein the entire file is read into memory and stored in a hierarchical tree − based form to represent all the features of an XML document.

SAX obviously cannot process information as fast as DOM can when working with large files. On the other hand, using DOM exclusively can really kill your resources, especially if used on a lot of small files.

SAX is read-only, while DOM allows changes to the XML file. Since these two different APIs literally complement each other, there is no reason why you cannot use them both for large projects.

Python Web Frameworks

A web framework is a code library that makes a developer's life easier when building reliable, scalable and maintainable web applications.

## Why are web frameworks useful?

Web frameworks encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application.

Common web framework functionality

Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications. These common operations include:

1. URL routing
2. HTML, XML, JSON, and other output format tinplating
3. Database manipulation
4. Security against Cross-site request forgery (CSRF) and other attacks
5. Session storage and retrieval

Not all web frameworks include code for all of the above functionality. Frameworks fall on the spectrum from executing a single use case to providing every known web framework feature to every developer. Some frameworks take the "batteries-included" approach where everything possible comes bundled with the framework while others have a minimal core package that is amenable to extensions provided by other packages.

## Comparing web frameworks

There is also a repository called [compare-python-web-frameworks](https://github.com/mattmakai/compare-python-web-frameworks) where the same web application is being coded with varying Python web frameworks, tinplating engines and object.

## Web framework resources

* When you are learning how to use one or more web frameworks it's helpful to have an idea of what the code under the covers is doing.
* Frameworks is a really well done short video that explains how to choose between web frameworks. The author has some particular opinions about what should be in a framework. For the most part I agree although I've found sessions and database ORMs to be a helpful part of a framework when done well.
* What is a web framework? Is an in-depth explanation of what web frameworks are and their relation to web servers?
* Jingo vs. Flash vs. Django: Choosing a Python web framework contains background information and code comparisons for similar web applications built in these three big Python frameworks.
* This fascinating blog post takes a look at the code complexity of several Python web frameworks by providing visualizations based on their code bases.
* Python’s web frameworks benchmarks  is a test of the responsiveness of a framework with encoding an object to JSON and returning it as a response as well as retrieving data from the database and rendering it in a template. There were no conclusive results but the output is fun to read about nonetheless.
* What web frameworks do you use and why are they awesome? Is a language agnostic Reedit discussion on web frameworks? It's interesting to see what programmers in other languages like and dislike about their suite of web frameworks compared to the main Python frameworks.
* This user-voted question & answer site asked "What are the best general purpose Python web frameworks usable in production?” The votes aren't as important as the list of the many frameworks that are available to Python developers.

## Web frameworks learning checklist

1. Choose a major Python web framework (Jingo or Flask are recommended) and stick with it. When you're just starting it's best to learn one framework first instead of bouncing around trying to understand every framework.
2. Work through a detailed tutorial found within the resources links on the framework's page.
3. Study open source examples built with your framework of choice so you can take parts of those projects and reuse the code in your application.
4. Build the first simple iteration of your web application then go to the [deployment](https://www.fullstackpython.com/deployment.html) section to make it accessible on the web.

2. SYSTEM STUDY

### SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

Features to be tested

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# 6.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.