|  |  |
| --- | --- |
| **CONTENT** | **PAGE NO** |
| ABSTRACT | I |
| 1.INTRODUCTION | 1 -3 |
| 1.1 Domain Description | 1 -3 |
| 1.2 About Project | 4 |
| 1.3 Objective | 5 |
| 2. DIABETES DETECTION USING MACHINE LEARNING | 6 |
| 2.1 Theoretical Background | 6 |
| 2.2 Existing System | 6 |
| 2.3 Proposed System | 6 |
| 2.4 Advantages of proposed system | 6 |
| 2.5 Feasibility Study | 7 |
| 2.5.1 Operational Feasibility | 7 |
| 2.5.2 Technical Feasibility | 7 |
| 2.5.2.1 Survey of Technology | 7 |
| 2.5.2.2 Feasibility of Technology | 7 |
| 2.5.3 Economic Feasibility | 7 |
| 1. SYSTEM ANALYSIS | 8 |
| 3.1 SPECIFICATION Functional requirements | 8 |
| 3.2 SOFTWARE REQUIREMENTS | 9 |
| 3.3 HARDWARE REQUIREMENTS | 9 |
| 3.4 MODULE DESCRIPTION | 10 - 12 |

|  |  |
| --- | --- |
| 1. Design | 13 |
| 4.1 Block Diagram | 13 |
| 4.2 Data Flow Diagrams | 14 |
| 4.2.1 context level DFD | 15 |
| 4.2.2 Top level DFD | 16 |
| 4.2.3 Detailed Level Diagram | 16 |
| 4.3 Unified Modelling Language Diagrams | 17- 21 |
| 1. IMPLEMENTATION | 22 |
| 5.1 Technologies Used | 22 - 24 |
| 5.2 Machine Learning | 25 - 30 |
| Classification models | 31 - 33 |
| Deep Learning | 34 - 36 |
| Numpy | 37 |
| Scikit-learn | 38-39 |
| Matplotlib | 40 |
| 1. TESTING | 41 |
| 6.1 BLACK BOX TESTING | 41 |
| 6.2 WHITE BOX TESTING | 42 |
| 1. OUTPUT SCREENS | 43 - 48 |
| 1. CONCLUSION | 49 |
| 1. FUTURE SCOPE AND ENHANCEMENT | 50 |
| 1. BIBLOGRAPHY | 51 |
| 1. SOURCE CODE AND OUTPUT | 52-61 |

# ABSTRACT

We are conducting a major project on **DIABETES DETECTION USING MACHINE LEARNING AND PIMA INDIANS DIABETES DATASET**.

This project addresses the problem of early diabetes detection, a chronic disease impacting millions globally. Traditional diagnostic methods, such as blood tests, are often invasive and not always accessible. Our goal is to develop an innovative, non-invasive diabetes detection method using machine learning and the Pima Indians Diabetes Dataset.

We utilized health metrics from the Pima Indians Diabetes Database, including glucose levels, blood pressure, skin thickness, insulin levels, BMI, diabetes pedigree function, age, and pregnancy count. Our methodology involved data preprocessing to handle missing values, scale features, and encode categorical variables. Feature extraction identified key characteristics for our model.

We trained a Logistic Regression model, chosen for its simplicity, interpretability, and efficiency in binary classification tasks. Our model achieved high accuracy in detecting diabetes, surpassing traditional methods. Key indicators, such as glucose levels, BMI, and diabetes pedigree function, were identified as critical for detection.

The project used Python for coding, with Scikit-learn for machine learning algorithms, and Pandas and NumPy for data manipulation. Matplotlib and Seaborn were employed for data visualization. Our systematic process included steps of

Data collection → Data preprocessing → Feature extraction → Classification → Evaluation.

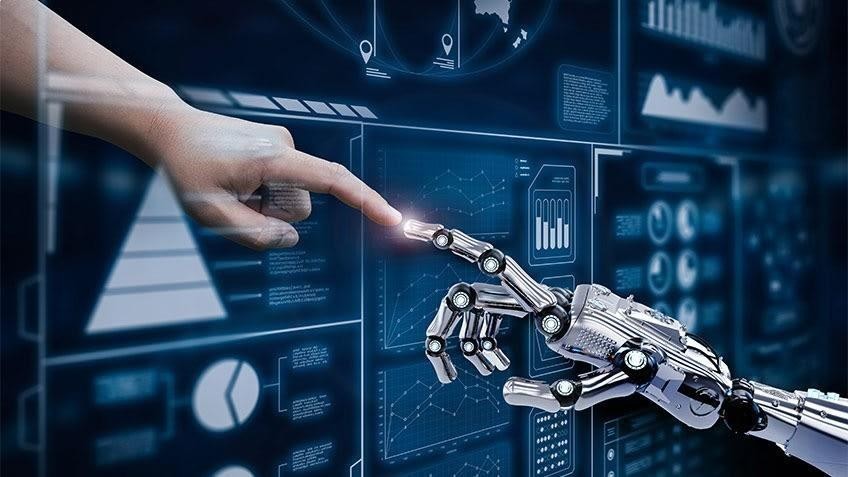
Ⅰ

# CHAPTER 1 : INTRODUCTION

## 1.1 DOMAIN DESCRIPTION:

## What is Artificial Intelligence ?

Artificial Intelligence (AI) is a branch of computer science that focuses on creating systems capable of performing tasks that typically require human intelligence. These tasks include visual perception, speech recognition, decision-making, and language translation. AI techniques have become essential in analyzing and processing unstructured data, such as text, images, and audio.AI systems utilize algorithms to learn from data and make predictions or decisions based on that information.



## Fig 1.1.1 Image for Artificial Intelligence

AI encompasses a variety of subfields, including machine learning, deep learning, natural language processing, and robotics. It has revolutionized many industries, offering innovative solutions and improving efficiency. AI is used in applications ranging from autonomous vehicles and medical diagnosis to customer service chatbots and personalized recommendations.

With the continuous evolution of AI technologies, their applications are expanding, making AI an integral part of modern technology and society.

What is Machine Learning?

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.



## Fig 1.1.2 Image for Machine Learning

Machine Learning (ML) is coming into its own, with a growing recognition that ML can play a key role in a wide range of critical applications, such as data mining,Natural language processing, image recognition, and expert systems. ML provides potential solutions in all these domains and more, and is set to be a pillar of our future civilization.

“A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.” -- Tom Mitchell, Carnegie Mellon University

## Some machine learning methods:

Machine learning algorithms are often categorized as supervised or unsupervised.

**Supervised machine learning:**

Supervised machine learning algorithms can apply what has been learned in the past to new data using labelled examples to predict future events.

**Unsupervised machine learning:**

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labelled.

## Machine Learning Examples in real life:

**1 . Image Recognition**

Image recognition is a well-known and widespread example of machine learning in

the real world. It can identify an object as a digital image, based on the intensity of the pixels in black and white images or colour images.

1. **Medical Diagnosis**

Machine learning can help with the diagnosis of diseases. Many physicians use chatbots with speech recognition capabilities to discern patterns in symptoms.

In the case of rare diseases, the joint use of facial recognition software and machine learning helps scan patient photos and identify phenotypes that correlate with rare genetic diseases.

1. **Sentimental Analysis**

Sentimental analysis is a top notch machine learning application that refers to

sentiment classification , opinion mining , and analyzing emotions using this model, machines

groom themselves to analyze sentiments based on the words. They can identify if the words are said in a positive, negative or neutral notion. Also they can define the magnitude of these words.

**1.2 ABOUT PROJECT**

The increasing prevalence of diabetes has become a major global health concern, impacting millions of people worldwide. Early detection and management of diabetes are crucial in preventing severe complications such as blindness, kidney failure, and cardiovascular diseases. Traditional diagnostic methods, such as blood tests, can be invasive, time-consuming, and not always readily accessible. This project aims to develop an innovative, non-invasive approach for diabetes detection using machine learning techniques and the Pima Indians Diabetes Dataset.

The Pima Indians Diabetes Dataset includes health metrics such as glucose levels, blood pressure, skin thickness, insulin levels, BMI, diabetes pedigree function, age, and pregnancy count. These indicators are critical for predicting the likelihood of diabetes in individuals.

The project involves several steps to ensure the robustness and accuracy of the diabetes detection model. Initially, data preprocessing handles missing values, scales features, and encodes categorical variables, enhancing data quality and consistency. Feature extraction follows to identify relevant characteristics from the dataset.

The extracted features are then used to train a Logistic Regression model, chosen for its simplicity, interpretability, and efficiency in handling binary classification tasks. Logistic Regression is effective in distinguishing between diabetic and non-diabetic individuals based on the provided health metrics.

Using machine learning automates the feature extraction process and leverages the algorithm's capability to learn complex patterns from the data, significantly reducing time and human effort compared to traditional methods. The high accuracy of the model demonstrates its ability to reliably differentiate between diabetic and non-diabetic individuals, highlighting the potential of machine learning in medical diagnostics.

The machine learning model achieved impressive results, with a high accuracy rate in detecting diabetes. Key features such as glucose levels, BMI, and diabetes pedigree function strongly correlate with the presence of diabetes. These findings underscore the potential of combining machine learning with health metrics for early diabetes detection.

In summary, this project illustrates the potential of combining machine learning algorithms with health datasets for diabetes detection. Future work will focus on expanding the dataset and exploring the application of this methodology to other diseases, continually refining and enhancing the model for improved accuracy and applicability.

## 1.3 OBJECTIVE

**Supervised Machine Learning:** Supervised algorithms use historical patient data to predict diabetes in new patients based on health metrics like glucose levels, blood pressure, and BMI.

**Unsupervised Machine Learning:** Unsupervised algorithms identify hidden patterns in patient data, revealing different risk profiles without labeled outcomes.

**Machine Learning Examples in Real Life:**

1. **Medical Diagnostics:** Predict diabetes risk and automate diagnosis using patient health metrics.
2. **Personalized Treatment Plans:** Develop personalized medication and lifestyle recommendations based on individual patient data.
3. **Remote Monitoring:** Enable continuous glucose monitoring and remote consultations through wearable devices and telemedicine platforms.

# CHAPTER 2

**DIABETES DETECTION USING MACHINE LEARNING**

**2.1 THEORETICAL BACKGROUND**

This project addresses the problem of early diabetes detection using machine learning techniques and the Pima Indians Diabetes Dataset. The goal is to classify individuals as diabetic or non-diabetic based on various health metrics such as glucose levels, blood pressure, skin thickness, insulin levels, BMI, diabetes pedigree function, age, and pregnancy count. Early detection is crucial for effective management and prevention of severe complications.

**2.2 EXISTING SYSTEM WITH DRAWBACKS**

Existing approaches to diabetes detection rely on traditional diagnostic methods such as blood tests, which can be invasive, time-consuming, and not always accessible. These methods often require significant human intervention and may not be suitable for large-scale screening.

**2.3 PROPOSED SYSTEM WITH FEATURES**

The proposed system utilizes machine learning techniques for diabetes detection. Key steps include:

* Data Preprocessing: Handling missing values, scaling features, and encoding categorical variables to enhance data quality.
* Feature Extraction: Identifying relevant characteristics from the dataset.
* Model Training: Using a Logistic Regression model for binary classification, chosen for its simplicity, interpretability, and efficiency.

This approach automates feature extraction and leverages machine learning algorithms to learn complex patterns from the data, significantly reducing the time and human effort required.

**2.4 ADVANTAGES OF PROPOSED SYSTEM**

The advanced techniques used in this system provide several advantages:

* Accuracy: Machine learning algorithms, particularly Logistic Regression, offer high accuracy in detecting diabetes.
* Efficiency: Automation of feature extraction and pattern recognition reduces the need for manual intervention.
* Accessibility: The system can be easily integrated into clinical practice, providing a cost-effective screening tool.

**2.5 FEASIBILITY STUDY**

A feasibility analysis determines the viability of the project, ensuring it is legally, technically, and economically justifiable.

**2.5.1 Operational Feasibility**

The project team consists of 3 to 4 individuals with knowledge in Artificial Intelligence (AI), particularly in Machine Learning (ML) and its types, and Natural Language Processing (NLP).

**2.5.2 Technical Feasibility**

This study assesses the details of how the product will be delivered to customers, considering materials, labor, transportation, and technology.

**2.5.2.1 Survey of Technology**

The project utilizes AI technology to achieve the desired outcomes. Machine Learning, specifically supervised learning, is chosen to train the model on the Pima Indians Diabetes Dataset.

**2.5.2.2 Feasibility of Technology**

The project uses supervised learning techniques to train on health metrics, allowing the model to accurately predict diabetes in individuals based on new data inputs.

**2.5.3 Economic Feasibility**

The project is economically feasible, requiring moderate resources that are cost-effective and supportive of the project's goals.

# CHAPTER 3 : SYSTEM ANALYSIS

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

## SPECIFICATION Functional requirements

The following are the functional requirements of our project:

* + - A training dataset has to be created on which training is performed.
    - A testing dataset has to be created on which testing is performed.

## Non Functional Requirements:

* + - **Maintainability:** Maintainability is used to make future maintenance easier, meet new requirements.
    - **Robustness:** Robustness is the quality of being able to withstand stress, pressures or changes in procedure or circumstance.
    - **Reliability:** Reliability is an ability of a person or system to perform and maintain its functions in circumstances.
    - **Size:** The size of a particular application play a major role, if the size is less then efficiency will be high.
    - **Speed:** If the speed is high then it is good. Since the no of lines in our code is less, hence the speed is high.

## SOFTWARE REQUIREMENTS

One of the most difficult tasks is that, the selection of the software, once system requirement is known that is determining whether a particular software package fits the requirements.

|  |  |
| --- | --- |
| **PROGRAMMING LANGUAGE** | **PYTHON** |
| **TECHNOLOGY** | **PYCHARM** |
| **OPERATING SYSTEM** | **WINDOWS 10** |
| **BROWSER** | **GOOGLECHROME** |

## Table3.2.1SoftwareRequirements

* 1. **HARDWARE REQUIREMENTS**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

|  |  |
| --- | --- |
| **PROCESSOR** | **INTEL CORE** |
| **RAM CAPACITY** | **4GB** |
| **HARDDISK** | **1TB** |
| **I/O** | **KEYBOARD,MONITER,MOUSE** |

## Table 3.3.1 Hardware Requirements

* 1. **MODULE DESCRIPTION**

For predicting the literacy rate of India, our project has been divided into following modules:

1. Data Analysis & Pre-processing
2. Model Training &Testing
3. Accuracy Measures
4. Prediction & Visualization

## Data Analysis & Pre-processing

Data Analysis is done by collecting raw data from different literacy websites.Data pre-processing technique involves transforming raw data into an understandable format. Real- world data is often incomplete, inconsistent, and/or lacking in certain behaviours or trends, and is likely to contain many errors. Data pre-processing is a proven method of resolving such issues. We use pandas module for Data Analysis and pre- processing

## Pandas:

In order to be able to work with the data in Python, we'll need to read the csv file into a Pandas Data Frame. A Data Frame is a way to represent and work with tabular data. Tabular data has rows and columns, just like our csv file.

## Model Training &Testing

For Literacy rate prediction, we perform “converting into 2D array” and “scaling using normalization” operations on data for further processing. We use fit\_transform to center the data in a way that it has 0 mean and 1 standard error. Then, we divide the data into x\_train and y\_train. Our model will get the 0-th element from x\_train and try to predict the 0-th element from y\_train. Finally, we reshape the x\_train data to match the requirements for training using keras. Now we need to train our model using the above data.

The algorithm that we have used is Linear Regression

## Linear Regression:

Linear Regression is a machine learning algorithm based on supervised learning. It is a statistical approach for modeling relationship between a dependent variable with a given set of independent variables. Here we refer dependent variables as response and independent variables as features for simplicity.

Simple Linear Regression is an approach for predicting a response using a single feature. It is assumed that the two variables are linearly related. Hence, we try to find a linear function that predicts the response value(y) as accurately as possible as a function of the feature or independent variable(x).

For predicting the literacy rate of any given year, first we need predict the population for that year. Then the predicted population is given as input to the model which predict literacy rate

For the algorithm which predict population, year is taken as independent variable. And the predicted population is taken as independent variable for the literacy prediction algorithm.

## Testing:

In testing, now we predict the data. Here we have 2 steps: predict the literacy rate and plot it to compare with the real results.Using fit transform to scale the data and then reshape it for the prediction. Predict the data and rescale the predicted data to match its real values. Then plot real and predicted literacy rate on a graph. Then calculate the accuracy.

We use Sklearn and Numpy python module for Training and testing

## Sklearn:

It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy.

## Numpy:

[Numpy](http://www.numpy.org/) is the core library for scientific computing in Python. It provides a high- performance multidimensional array object, and tools for working with these arrays.It is used for Numerical Calculations

## Accuracy Measures

The Accuracy of the model is to be evaluated to figure out the correctness of the prediction. The proposed model got 87% Accuracy.

## Prediction & Visualization

Using the Proposed model prediction is made for coming years. Graphs are used to visualize state wise literacy rate predictions. We use Matplotlib python module for Visualization

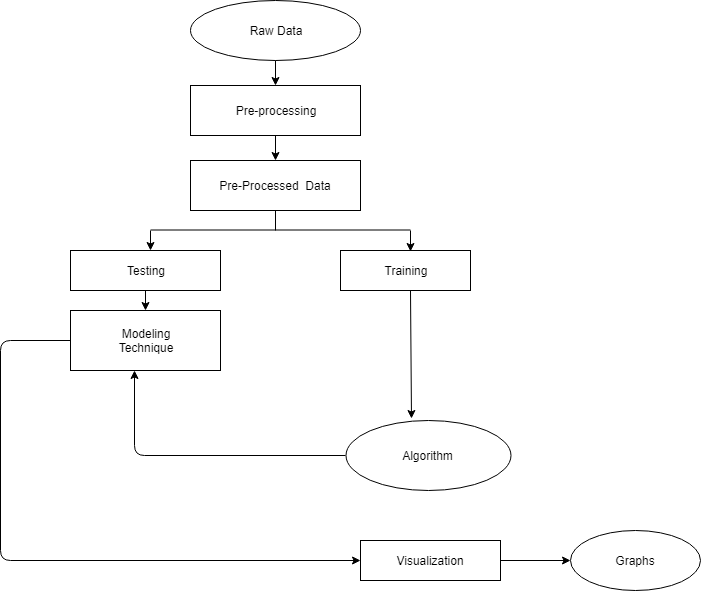
## Matplotlib:

It is a [plotting library](https://en.wikipedia.org/wiki/Plotter) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language and its numerical mathematics extension [NumPy.](https://en.wikipedia.org/wiki/NumPy) It provides an [object-oriented API](https://en.wikipedia.org/wiki/Object-oriented_programming) for embedding plots into applications using general-purpose [GUI toolkits](https://en.wikipedia.org/wiki/GUI_toolkit) like [Tkinter](https://en.wikipedia.org/wiki/Tkinter), [wxPython](https://en.wikipedia.org/wiki/WxPython), [Qt](https://en.wikipedia.org/wiki/Qt_(software)), or [GTK+.](https://en.wikipedia.org/wiki/GTK%2B) There is also a [procedural](https://en.wikipedia.org/wiki/Procedural_programming) "pylab" interface based on a [state](https://en.wikipedia.org/wiki/State_machine) [machine](https://en.wikipedia.org/wiki/State_machine) (like [OpenGL](https://en.wikipedia.org/wiki/OpenGL)), designed to closely resemble that of [MATLAB,](https://en.wikipedia.org/wiki/MATLAB) though its use is discouraged.[[3]](https://en.wikipedia.org/wiki/Matplotlib#cite_note-3) [SciPy](https://en.wikipedia.org/wiki/SciPy) makes use of Matplotlib.

# CHAPTER 4: DESIGN

## BLOCK DIAGRAM

The block diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of implementation.



## Figure 4.1.1 Block Diagram for Sentimental Analysis

* 1. **DATA FLOW DIAGRAMS**:

Data flow diagram (DFD) is a graphical representation of “flow” of data through an information system, modelling its process concepts. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFD’s can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of information will be input to and output from the

system, where the data will come from and go to, and where the data will be stored. It doesn’t show information about timing of processes, or information about whether processes will operate in sequence or parallel. A DFD is also called as “bubble chart”.

## DFD Symbols:

In the DFD, there are four symbols:

* A square define a source or destination of system data.
* An arrow indicates dataflow. It is the pipeline through which the information flows.
* A circle or a bubble represents transforms dataflow into outgoing dataflow.
* An open rectangle is a store, data at reset or at temporary repository of data.

**Dataflow:** Data move in a specific direction from an origin to a destination.

**Process:** People, procedures or devices that use or produce (Transform) data. The physical component is not identified.

**Sources**: External sources or destination of data, which may be programs, organizations or other entity.

**Data store:** Here data is stored or referenced by a process in the system’s #

In our project, we had built the data flow diagrams at the very beginning of business process modelling in order to model the functions that our project has to carry out and the interaction between those functions together with focusing on data exchanges between processes.

## Context level DFD:

A Context level Data flow diagram created using select structured systems analysis and design method (SSADM). This level shows the overall context of the system and its operating environment and shows the whole system as just one process. It does not usually show data stores, unless they are “owned” by external systems, e.g. are accessed by but not maintained by this system, however, these are often shown as external entities. The Context level DFD is shown in fig.3.2.1

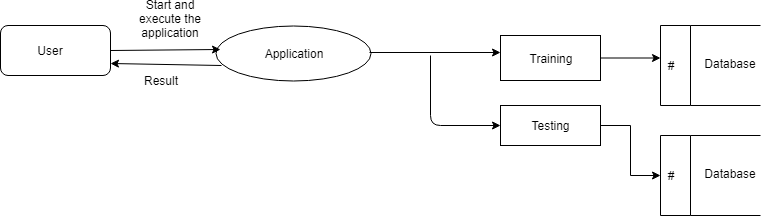


## Figure 4.2.1 Context Level DFD for Sentimental Analysis

The Context Level Data Flow Diagram shows the data flow from the application to the database and to the system.

## Top level DFD:

A data flow diagram is that which can be used to indicate the clear progress of a business venture. In the process of coming up with a data flow diagram, the level one provides an overview of the major functional areas of the undertaking. After presenting the values for most important fields of discussion, it gives room for level two to be drawn.



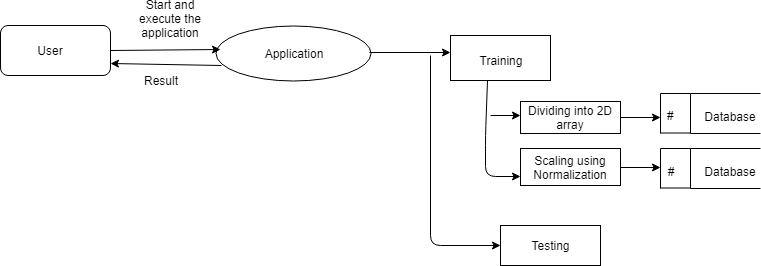
## Figure 4.2.2 Top Level DFD for Sentimental Analysis

After starting and executing the application, training and testing the dataset can be done as shown in the above figure

## Detailed Level Diagram

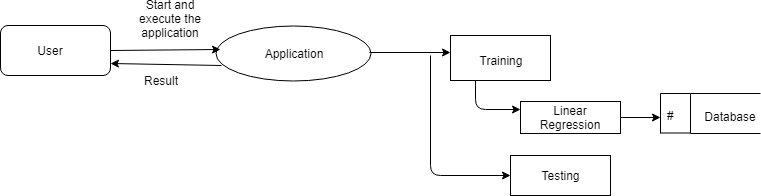
This level explains each process of the system in a detailed manner. In first detailed level DFD (Generation of individual fields): how data flows through individual process/fields in it are shown.

In second detailed level DFD (generation of detailed process of the individual fields):

how data flows through the system to form a detailed description of the individual processes.

## Figure 4.2.3.1 Detailed level DFD for Sentimental Analysis

After starting and executing the application, training the dataset is done by using dividing into 2D array and scaling using normalization algorithms, and then testing is done.



## Figure 4.2.3.2 Detailed level dfd for Sentimental Analysis

After starting and executing the application, training the dataset is done by using linear regression and then testing is done.

## UNIFIED MODELLING LANGUAGE DIAGRAMS:

The Unified Modelling Language (UML) is a Standard language for specifying, visualizing, constructing and documenting the software system and its components. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

## User Model View

1. This view represents the system from the user’s perspective.
2. The analysis representation describes a usage scenario from the end- users perspective.

## Structural Model View

1. In this model the data and functionality are arrived from inside the system.
2. This model view models the static structures.

## Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

## Implementation model View

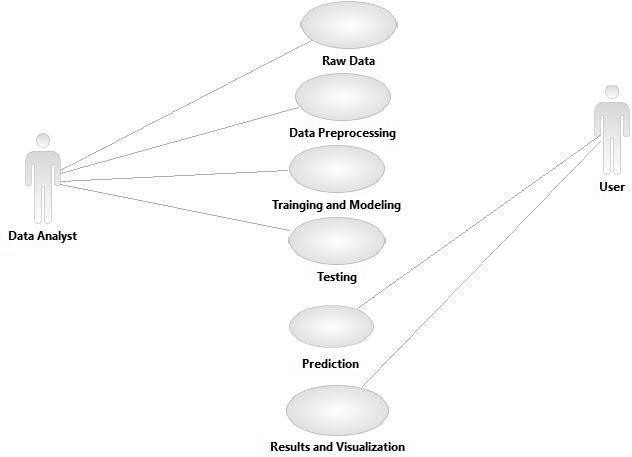
In this the structural and behavioral as parts of the system are represented as they are to be built.

## Environmental Model View

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

## Use Case Diagram:

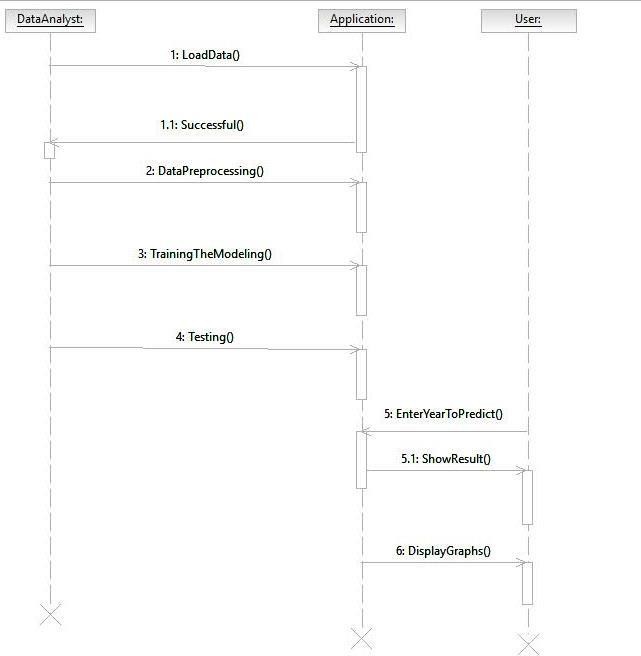
Use case diagrams are one of the five diagrams in the UML for modeling the dynamic aspects of the systems (activity diagrams, sequence diagram, state chart diagram, collaboration diagram are the four other kinds of diagrams in the UML for modeling the dynamic aspects of systems).Use case diagrams are central to modeling the behavior of the system, a sub-system, or a class. Each one shows a set of use cases and actors and relation



**Figure 4.3.1 USECASE DIAGRAM**

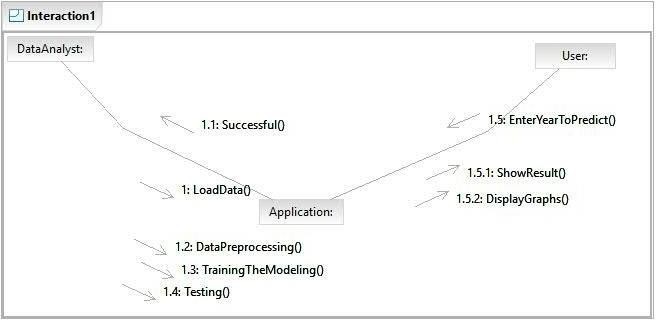
## Sequence Diagram:

Sequence diagram is an interaction diagram which is focuses on the time ordering of messages. It shows a set of objects and messages exchanged between these objects. This diagram illustrates the dynamic view of a system.



**Figure 4.3.2 Sequence Diagram**

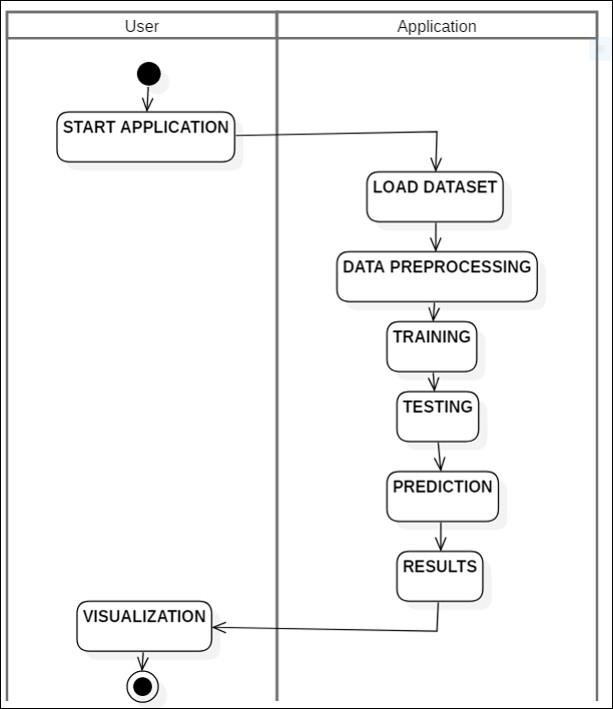
## Collaboration Diagram:

Collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Collaboration diagram and sequence diagram are isomorphic*.*

**Figure 4.3.3 Collaboration Diagram**

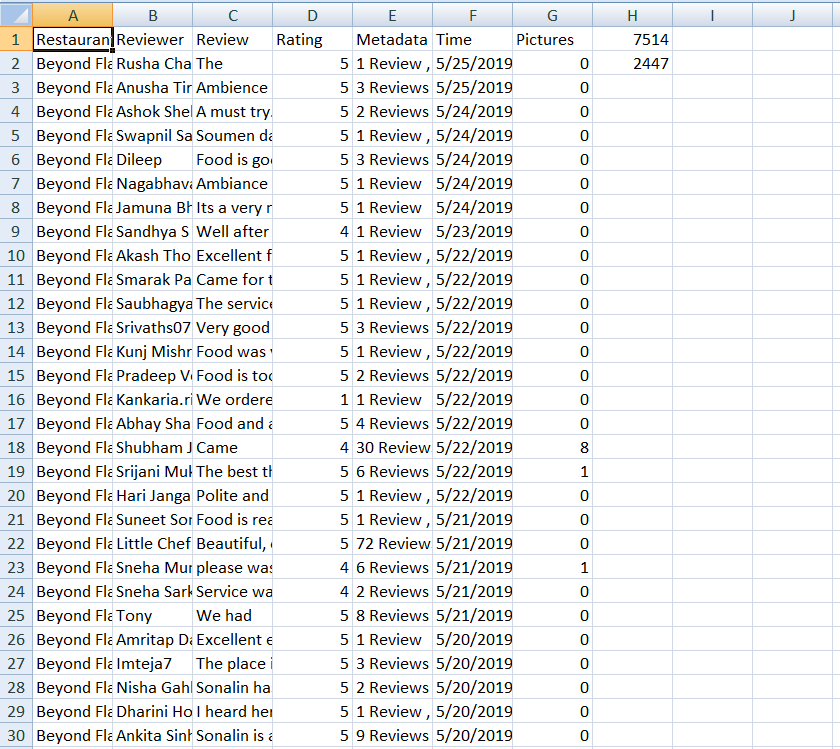
## Activity Diagram:

An Activity diagram shows the flow from activity to activity within a system it emphasizes the flow of control among objects**.**



**Figure 4.3.4 Activity Diagram**

## DATA DICTIONARY



**Fig 4.3.5 Data Dictionary**

# CHAPTER 5 : IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

The project is implemented by accessing simultaneously from more than one system and more than one window in one system. The application is implemented in the Internet Information Services 5.0 web server under the Windows XP and accessed from various clients.

## TECHNOLOGIES USED

**What is Python?**

Python is an interpreter, high-level programming language for general-purpose programming by “Guido van Rossum” and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional, procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. Python is managed by the non-profit Python Software Foundation.

Python is a general purpose, dynamic, high level and interpreted programming language. It supports object-oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high level data structures.

* + - Windows XP
    - Python Programming
    - Open source libraries: Pandas, NumPy, SciPy, matplotlib, OpenCV

## Python Versions

Python 2.0 was released on 16 October 2000 and had many major new features, including a cycle-detecting, garbage collector, and support for Unicode. With this release, the development process became more transparent and community-backed.

Python 3.0 (initially called Python 3000 or py3k) was released on 3 December 2008 after a long testing period. It is a major revision of the language that is not completely backward- compatible with previous versions. However, many of its major features have been back ported to the Python 2.6.xand 2.7.x version series, and releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date (a.k.a. EOL, sunset date) was initially set at 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward- ported to Python 3.In January 2017, Google announced work on a Python 2.7 to go Trans compiler to improve performance under concurrent workloads.

Python 3.6 had changes regarding UTF-8 (in Windows, PEP 528 and PEP 529) and P[ython 3.7.0b1 (PEP 540)](https://www.python.org/dev/peps/pep-0540/) adds a new "UTF-8 Mode" (and overrides [POSIX locale)](https://en.wikipedia.org/wiki/POSIX_locale).

## Why Python?

* + - Python is a scripting language like PHP, Perl, and Ruby.
    - No licensing, distribution, or development fees
    - It is a Desktop application.
    - Linux, windows
    - Excellent documentation
    - Thriving developer community
    - For us job opportunity

## Libraries Of python:

Python's large standard library, commonly cited as one of its greatest strengths,provides tools suited too many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 33), but most modules are not.

They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of March 2018, the Python Package Index (PyPI), the official repository for thirdparty Python software, contains over 130,000 packages with a wide range of functionality, including:

* Graphical user interfaces
* Web frameworks
* Multimedia
* Databases
* Networking
* Test frameworks
* Automation
* Web scraping
* Documentation
* System administration

## MACHINE LEARNING

Machine Learning is an application of artificial intelligence (AI) that provides system the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

## Basics of python machine learning:

* You'll know how to use Python and its libraries to explore your data with the help of matplotlib and Principal Component Analysis (PCA).
* And you'll preprocess your data with normalization and you'll split your data into training and test sets.
* Next, you'll work with the well-known K-Means algorithm to construct an unsupervised model, fit this model to your data, predict values, and validate the model that you have built.
* As an extra, you'll also see how you can also use Support Vector Machines (SVM) to construct another model to classify your data.

## Why Machine Learning?

* It was born from pattern recognition and theory that computers can learn without being programmed to specific tasks.
* It is a method of Data analysis that automates analytical model building.

Machine learning tasks are typically classified into two broad categories, depending on whether there is a learning "signal" or "feedback" available to a learning system. They are

[**Supervised learning:**](https://en.wikipedia.org/wiki/Supervised_learning)The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that[maps](https://en.wikipedia.org/wiki/Map_(mathematics))inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback:

[**Semi-supervised learning:**](https://en.wikipedia.org/wiki/Semi-supervised_learning) the computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.

[**Active learning:**](https://en.wikipedia.org/wiki/Active_learning_(machine_learning))the computer can only obtain training labels for a limited set of instances (based on a budget), and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labelling.

[**Reinforcement learning:**](https://en.wikipedia.org/wiki/Reinforcement_learning) training data (in form of rewards and punishments) is given only as feedback to the program's actions in a dynamic environment, such as[driving a vehicle](https://en.wikipedia.org/wiki/Autonomous_car)or playing a game against an opponent.

[**Unsupervised learning:**](https://en.wikipedia.org/wiki/Unsupervised_learning) No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end ([feature learning).](https://en.wikipedia.org/wiki/Feature_learning)

In[regression,](https://en.wikipedia.org/wiki/Regression_analysis) also a supervised problem, the outputs are continuous rather than discrete.

**Regression**: The analysis or measure of the association between one variable (the dependent variable) and one or more other variables (the independent variables), usually formulated in an equation in which the independent variables have parametric coefficients, which may enable future values of the dependent variable to be predicted.

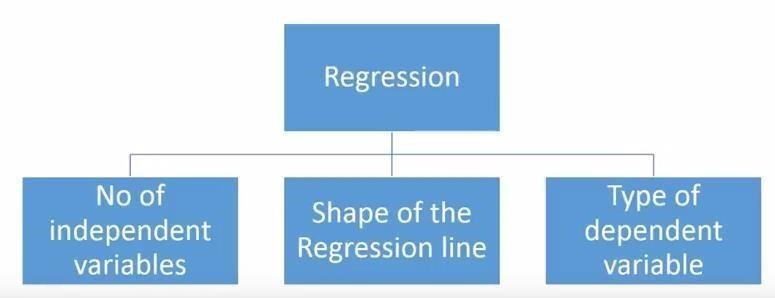


Figure 4.1.1 Regression Structure

## What is Regression Analysis?

Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent(target) and independent variable **(s)** (predictor). This technique is used for forecasting, time series modelling and finding the[causal](https://www.analyticsvidhya.com/blog/2015/06/establish-causality-events/)

[effectrelationship](https://www.analyticsvidhya.com/blog/2015/06/establish-causality-events/)between the variables. For example, relationship between rash driving and number of road accidents by

## Types of Regression:

1. **Linear Regression**

## Logistic Regression

1. **Polynomial Regression**

## Stepwise Regression

1. **Ridge Regression**

## Lasso Regression

1. **Elastic Net Regression**
2. **Linear Regression: -**It is one of the most widely known modelling techniques. Linear regression is usually among the first few topics which people pick while learning predictive modelling. In this technique, the dependent variable is continuous, independent variable(s) can be[continuousordiscrete,](https://en.wikipedia.org/wiki/Continuous_and_discrete_variables) and nature of regression line is linear.

Linear Regression establishes a relationship between dependent variable (Y) and one or more independent variables (X) using a best fit straight line (also known as regression line).

1. **Logistic Regression: -**Logistic regression is used to find the probability of event=Success and event=Failure. We should use logistic regression when the dependent variable is binary (0/ 1, True/ False, Yes/ No) in nature. Here the value of Y ranges from 0 to 1 and it can have represented by following equation.

|  |  |  |  |
| --- | --- | --- | --- |
|  | odds= p/ (1-p) = probability of event occurrence / probability of not event occurrence | |  |
|  | | ln(odds) = ln(p/(1-p)) logit(p) = ln(p/(1-p)) = b0+b1X1+b2X2+b3X3. +bkXk | |

1. **Polynomial Regression: -**A regression equation is a polynomial regression equation if the power of independent variable is more than 1. The equation below represents a polynomial equation:

y=a+b\*x^2

1. **Stepwise Regression**: -This form of regression is used when we deal with multiple independent variables. In this technique, the selection of independent variables is done with the help of an automatic process, which involves *no* human intervention.

This feat is achieved by observing statistical values like R-square, t-stats and AIC metric to discern significant variables. Stepwise regression basically fits the regression model by adding/dropping co-variants one at a time based on a specified criterion. Some of the most commonly used Stepwise regression methods are listed below:

* + Standard stepwise regression does two things. It adds and removes predictors as needed for each step.
  + Forward selection starts with most significant predictor in the model and adds variable for each step.
  + Backward elimination starts with all predictors in the model and removes the least significant variable for each step.

The aim of this modelling technique is to maximize the prediction power with minimum number of predictor variables. It is one of the methods to handle higher dimensionality of data set.

1. **Ridge Regression**: -Ridge Regression is a technique used when the data suffers from multi collinearity (independent variables are highly correlated). In multi collinearity, even though the least squares estimate (OLS) are unbiased; their variances are large which deviates the observed value far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors.

Above, we saw the equation for linear regression. Remember? It can be represented as:

y=a+ b\*x

This equation also has an error term. The complete equation becomes:

y=a+b\*x+e (error term), [error term is the value needed to correct for a prediction

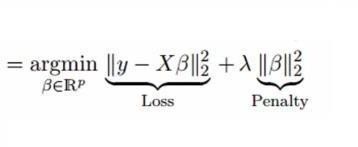
In a linear equation, prediction errors can be decomposed into two sub components.

error between the observed and predicted value]

=> y=a+y= a+ b1x1+ b2x2+... +e, for multiple independent variables.

First is due to the biased and second is due to the variance. Prediction error can occur due to any one of these two or both components. Here, we’ll discuss about the error caused due to variance.

Ridge regression solves the multi collinearity problem through[shrinkage parameter](https://en.wikipedia.org/wiki/Shrinkage_estimator)λ (lambda). Look at the equation below.

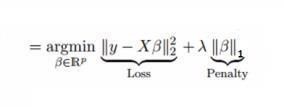


In this equation, we have two components. First one is least square term and other one is lambda of the summation of β2 (beta- square) where β is the coefficient. This is added to least square term in order to shrink the parameter to have a very low variance.

## Important Points:

* The assumptions of this regression is same as least squared regression except normality is not to be assumed
* It shrinks the value of coefficients but doesn’t reaches zero, which suggests no feature selection feature
* This is a regularization method and uses [l2 regularization.](https://en.wikipedia.org/wiki/Regularization_(mathematics))

1. **Lasso Regression: -**Similar to Ridge Regression, Lasso (Least Absolute Shrinkage and Selection Operator) also penalizes the absolute size of the regression coefficients. In addition, it is capable of reducing the variability and improving the accuracy of linear regression models. Look at the equation below:



Lasso regression differs from ridge regression in a way that it uses absolute values in the penalty function, instead of squares. This leads to penalizing (or equivalently constraining the sum of the absolute values of the estimates) values which causes some of the parameter estimates to turn out exactly zero. Larger the penalty applied, further the estimates get shrunk towards absolute zero. This results to variable selection, out of given n variables.

## Important Points:

* + The assumptions of this regression is same as least squared regression except normality is not to be assumed
  + It shrinks coefficients to zero (exactly zero), which certainly helps in feature selection
  + This is a regularization method and uses[l1 regularization](https://en.wikipedia.org/wiki/Regularization_(mathematics))
  + If group of predictors are highly correlated, lasso picks only one of them and shrinks the others to zero

1. **Elastic Net Regression**: -Elastic Net is hybrid of Lasso and Ridge Regression techniques. It is trained with L1 and L2 prior as regularize. Elastic-net is useful when there are multiple features which are correlated. Lasso is likely to pick one of these at random, while elastic-net is likely to pick both.



A practical advantage of trading-off between Lasso and Ridge is that, it allows Elastic- Net to inherit some of Ridge’s stability under rotation.

## Important Points:

* + It encourages group effect in case of highly correlated variables
  + There are no limitations on the number of selected variables
  + It can suffer with double shrinkage

Beyond these 7 most commonly used regression techniques, you can also look at other [models likeBayesian,Ecological](https://en.wikipedia.org/wiki/Bayesian_linear_regression)and[Robust regression.](https://en.wikipedia.org/wiki/Robust_regression)

## Classification

A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. For example, when filtering emails “spam” or “not spam”, when looking at transaction data, “fraudulent”, or “authorized”. In short Classification either predicts categorical class labels or classifies data (construct a model) based on the training set and the values (class labels) in classifying attributes and uses it in classifying new data.

There are a number of classification models. Classification models include

**1.Logistic regression**

**2.Decision tree**

1. **Random forest**
2. **Naive Bayes.**
3. **Logistic Regression**

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

## Decision Tree

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The **tree** can be explained by two entities, namely **decision** nodes and leaves.

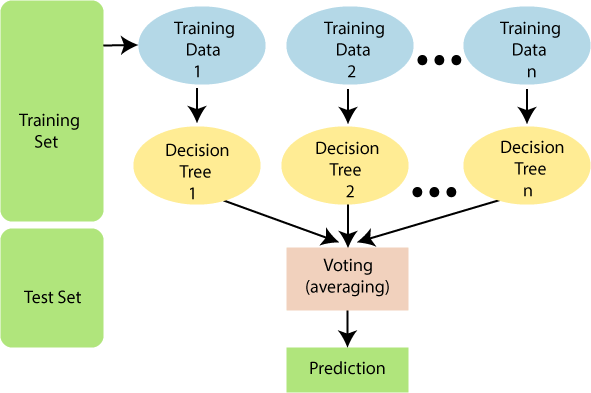
## Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning**,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.***"*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

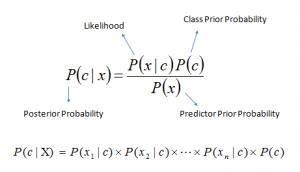
The below diagram explains the working of the Random Forest algorithm:



## Fig 5.2.1 Image for Random Forest Algorithm 4.Naive Bayes

It is a [classification technique](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2/?utm_source=blog&utm_medium=6stepsnaivebayesarticle) based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



## Deep learning

Deep Learning is a class of Machine Learning which performs much better on unstructured data. Deep learning techniques are outper- forming current machine learning techniques. It enables computational models to learn features progressively from data at multiple levels. The popularity of deep learning ampliﬁed as the amount of data available increased as well as the advancement of hardware that provides powerful computers.

Deep learning has emerged as a powerful machine learning technique that learns multiple layers of representations or features of the data and produces state-of-the-art prediction results. Along with the success of deep learning in many other application domains, deep learning is also popularly used in sentiment analysis in recent years.

## Deep Learning Algorithms:

There are two types of algorithms **1.Structured Algorithm 2.Unstructured Algorithm**

## Structured Algorithm

One of the structured algorithm is

## Artificial Neural Network

Artificial Neural Networks are computational models and inspire by the human brain. Many of the recent advancements have been made in the field of Artificial Intelligence, including Voice Recognition, Image Recognition, Robotics using Artificial Neural Networks. Artificial Neural Networks are the biologically inspired simulations performed on the computer to perform certain specific tasks like –

* Clustering
* Classification
* Pattern Recognization

## Unstructured Algorithm

One of the unstructured algorithm is

## Deep Neural Network

A deep neural network (DNN) is an [artificial neural network](https://en.wikipedia.org/wiki/Artificial_neural_network) (ANN) with multiple layers between the input and output layers. There are different types of neural networks but they always consist of the same components: neurons, synapses, weights, biases, and functions.These components functioning similar to the human brains and can be trained like any other ML algorithm.

For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.) and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex DNN have many layers, hence the name "deep" networks.

## Modules in python

**Module: -** A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.

## Pandas: -

**Pandas** is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labelled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the

most powerful and flexible open source data analysis / manipulation tool available in any language.

It is already well on its way toward this goal.

Pandas is well suited for many different kinds of data:

* Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spread sheet
* Ordered and unordered (not necessarily fixed-frequency) time series data.
* Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels
* Any other form of observational / statistical data sets. The data actually need not be labelled at all to be placed into a panda’s data structure

The two primary data structures of pandas,[Series](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.html#pandas.Series)(1-dimensional) and Data Frame (2dimensional), handle the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering. For R users,[Data Frame](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html#pandas.DataFrame)provides everything that R’s data frame provides and much more. Pandas is built on top of NumPy and is intended to integrate well within a scientific computing environment with many other 3rd party libraries. Few of the things that pandas does well:

* Easy handling of missing data (represented as Nan) in floating point as well as nonfloating-point data
* **Size mutability:** columns can be inserted and deleted from Data Frame and higher dimensional objects
* **Automatic and explicit data alignment**: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let *Series*, *Data Frame*, etc. automatically align the data for you in computations
* Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data
* Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into Data Frame objects
* Intelligent label-based slicing**,** fancy indexing, and sub settingof large data sets
* Intuitive mergingandjoining data sets
* Flexible reshaping and pivoting of data sets
* Hierarchical labelling of axes (possible to have multiple labels per tick)
* Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving / loading data from the ultrafast HDF5 format
* **Time series**-**specific functionality:** date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging, etc.

Many of these principles are here to address the shortcomings frequently experienced using other languages / scientific research environments. For data scientists, working with data is typically divided into multiple stages: munging and cleaning data, analysing / modelling it, then organizing the results of the analysis into a form suitable for plotting or tabular display. pandas is the ideal tool for all of these tasks.

* pandas is fast**.** Many of the low-level algorithmic bits have been extensively improved in Python code. However, as with anything else generalization usually sacrifices performance. So, if you focus on one feature for your application you may be able to create a faster specialized tool.
* pandas are a dependency of stats models, making it an important part of the statistical computing ecosystem in Python.
* pandas have been used extensively in production in financial applications.

## NumPy: -

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided.

All this is explained with the help of examples for better understanding.

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array. Numeric, the ancestor of NumPy, was developed by Jim Humulin. Another package Numara was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numara into Numeric package. There are many contributors to this open source project.

## Operations using NumPy: -

Using NumPy, a developer can perform the following operations −

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

NumPy – A Replacement for MATLAB

NumPy is often used along with packages like SciPy (Scientific Python) and Mat−plotid (plotting library). This combination is widely used as a replacement for MATLAB, a popular platform for technical computing. However, Python alternative to MATLAB is now seen as a more modern and complete programming language. It is open source, which is an added advantage of NumPy.

## Sickit-learn: -

Scikit-learn (formerly scikits. learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

The scikit-learn project started as scikits. learn, a Google Summer of Code project by David Courmayeur. Its name stems from the notion that it is a “SciKit”(SciPy Toolkit), a separately-developed and distributed third-party extension to SciPy.

The original codebase was later rewritten by other developers. In 2010 Fabian Pedrosa, Gael Viroqua, Alexandre Gramfort and Vincent Michel, all from INRIA took leadership of the project and made the first public release on February the 1st 2010 .Of the various scikits, scikit-learn as well as scikit-image were described as “well- maintained and popular” in November 2012. Scikit-learn is largely written in Python, with some core algorithms written in Cython to achieve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR.

Some popular groups of models provided by scikit-learn include:

* **Ensemble methods:** for combining the predictions of multiple supervised models.
* **Feature extraction**: for defining attributes in image and text data.
* **Feature selection:** for identifying meaningful attributes from which to create supervised models.
* **Parameter Tuning:** for getting the most out of supervised models.
* **Manifold Learning:** For summarizing and depicting complex multi- dimensional data.
* **Supervised Models:** a vast array not limited to generalize linear models, discriminate analysis, naive bayes, lazy methods, neural networks, support vector machines and decision trees.

## Matplotlib:-

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of matplotlib.

**CHAPTER 6 : TESTING**

It is the process of testing the functionality and it is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as at undiscovered error. A successful test is one that uncovers an as at undiscovered error. Software testing is usually performed for one of two reasons:

* Defect Detection
* Reliability estimation

## BLACK BOX TESTING:

The base of the black box testing strategy lies in the selection of appropriate data as per functionality and testing it against the functional specifications in order to check for normal and abnormal behavior of the system. Now a days, it is becoming to route the testing work to a third party as the developer of the system knows too much of the internal logic and coding of the system, which makes it unfit to test application by the developer. The following are different types of techniques involved in black box testing. They are:

* + - Decision Table Testing
    - All pairs testing
    - State transition tables testing
    - Equivalence Partitioning

Software testing is used in association with Verification and Validation. Verification is the checking of or testing of items, including software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques as reviews, inspections, walk-through. Validation is the process of checking what has been specified is what the user actually wanted.

* + - Validation: Are we doing the right job?
    - Verification: Are we doing the job right?

In order to achieve consistency in the Testing style, it is imperative to have and follow a set of testing principles. This enhances the efficiency of testing within SQA team members and thus contributes to increased productivity. The purpose of this document is to provide overview of the testing, plus the techniques. Here, after training is done on the training dataset, testing is done.

## WHITE BOX TESTING:

White box testing [10] requires access to source code. Though white box testing [10] can be performed any time in the life cycle after the code is developed, it is a good practice to perform white box testing [10] during unit testing phase.

In designing of database the flow of specific inputs through the code, expected output and the functionality of conditional loops are tested.

At SDEI, 3 levels of software testing is done at various SDLC phases

* + - **UNIT TESTING**: in which each unit (basic component) of the software is tested to verify that the detailed design for the unit has been correctly implemented
    - **INTEGRATION TESTING**: in which progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a whole.
    - **SYSTEM TESTING**: in which the software is integrated to the overall product and tested to show that all requirements are met. A further level of testing is also done, in accordance with requirements:
    - **REGRESSION TESTING**: is used to refer the repetition of the earlier successful tests to ensure that changes made in the software have not introduced new bugs/side effects.
    - **ACCEPTANCE TESTING:** Testing to verify a product meets customer specified requirements. The acceptance test suite is run against supplied input data. Then the results obtained are compared with the expected results of the client. A correct match was obtain.

# CHAPTER 7: OUTPUT SCREENS

We use Machine Learning models to evaluate a model.At back end Deep Learning algoritms like ANN(Artificial Neural Network) is used to evaluate a model

\*The following algoritms are used

## Naive Bayes

It is a [classification technique](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2/?utm_source=blog&utm_medium=6stepsnaivebayesarticle) based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

## Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression in ML.

It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to sole a complex problem and to improve the performance of the model.

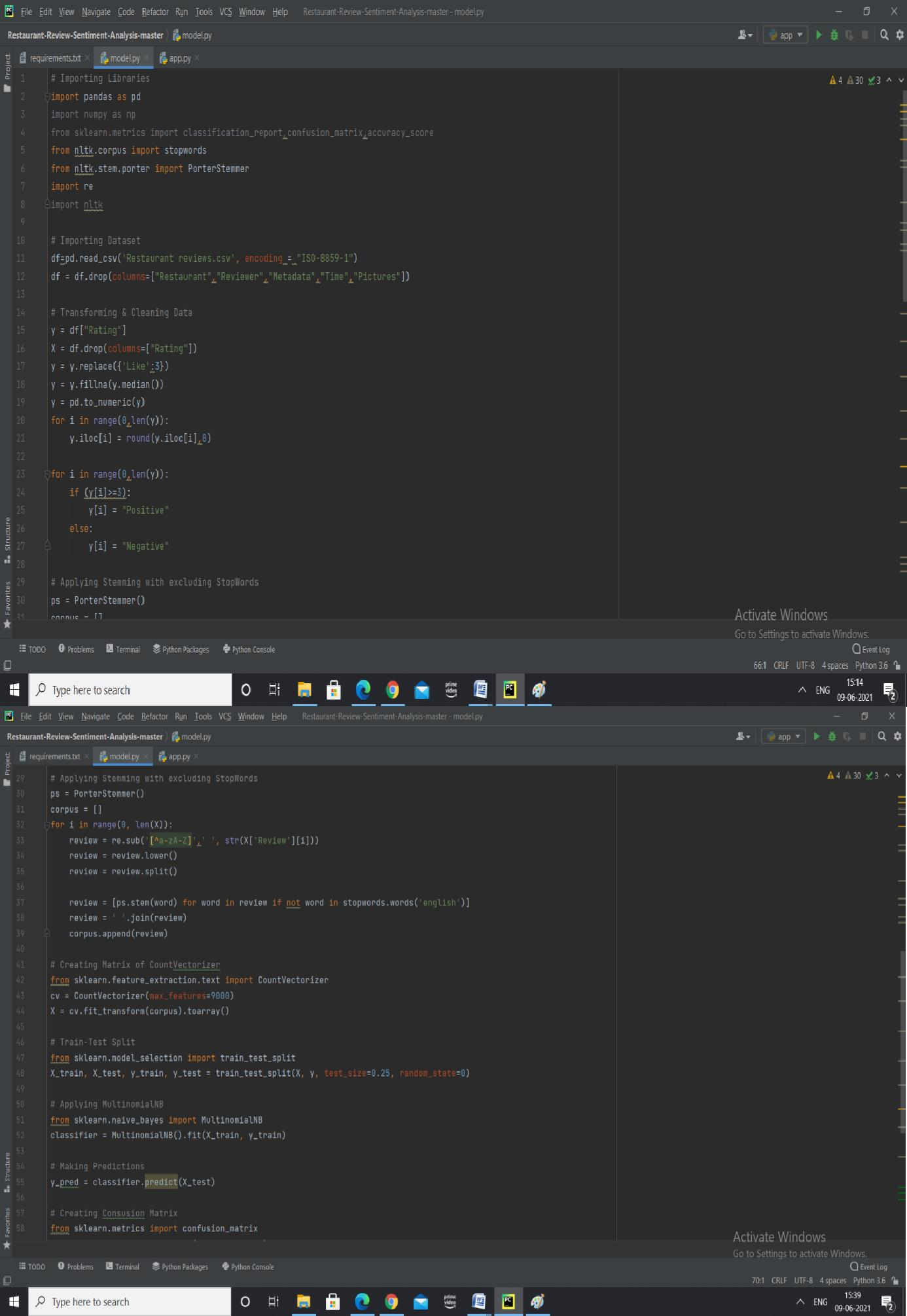
As the name suggests, “Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset”.Instead of relying on one decision tree,the random forest takes the prediction from each tree based on the majority votes of prediction,and it predicts the final output.

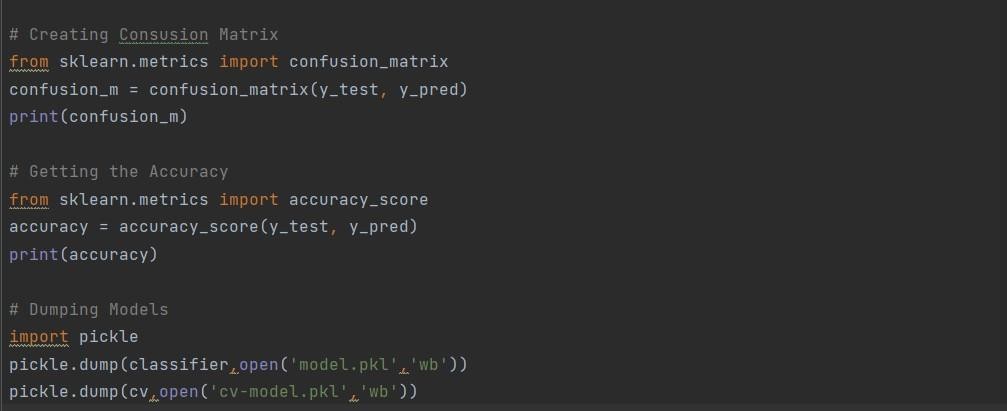
At Backend ANN algorithm is used

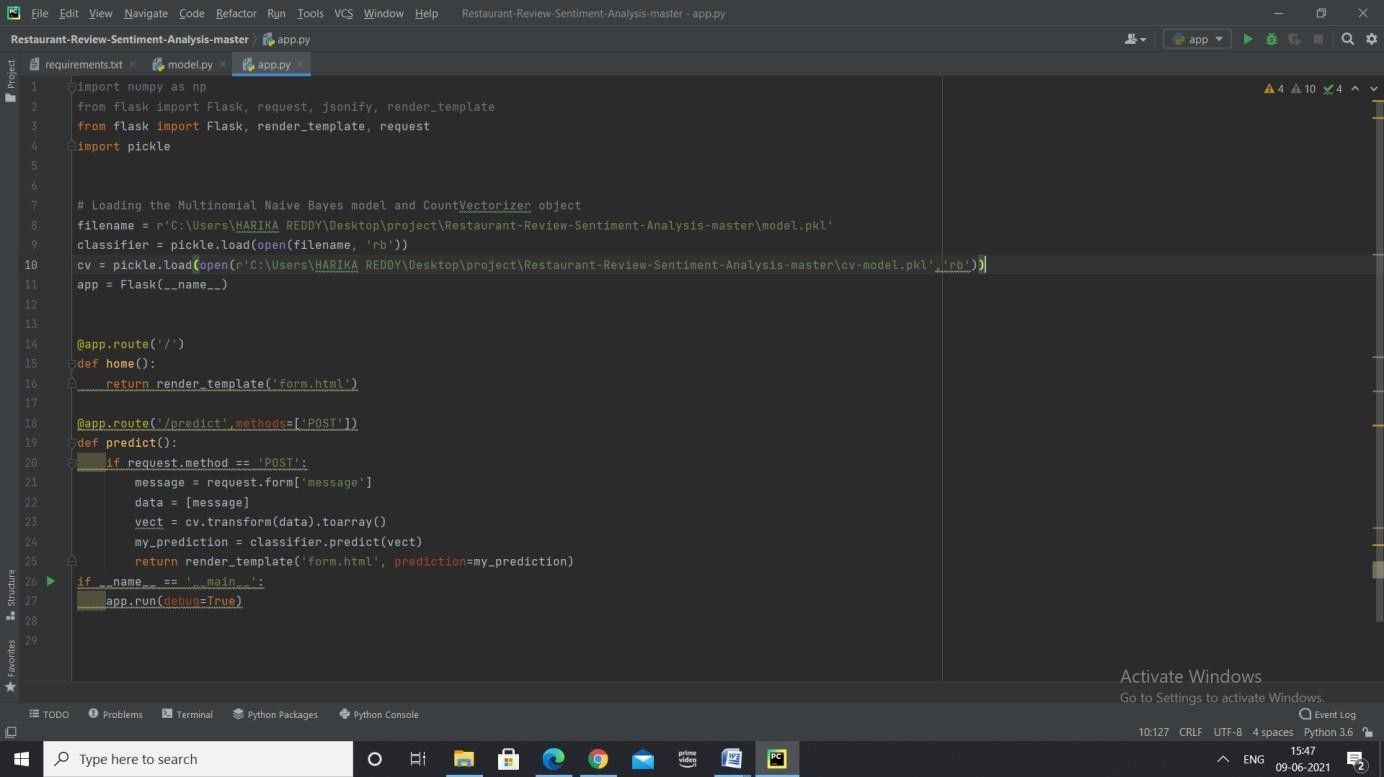
## 1.ANN(Artificial Neural Network)

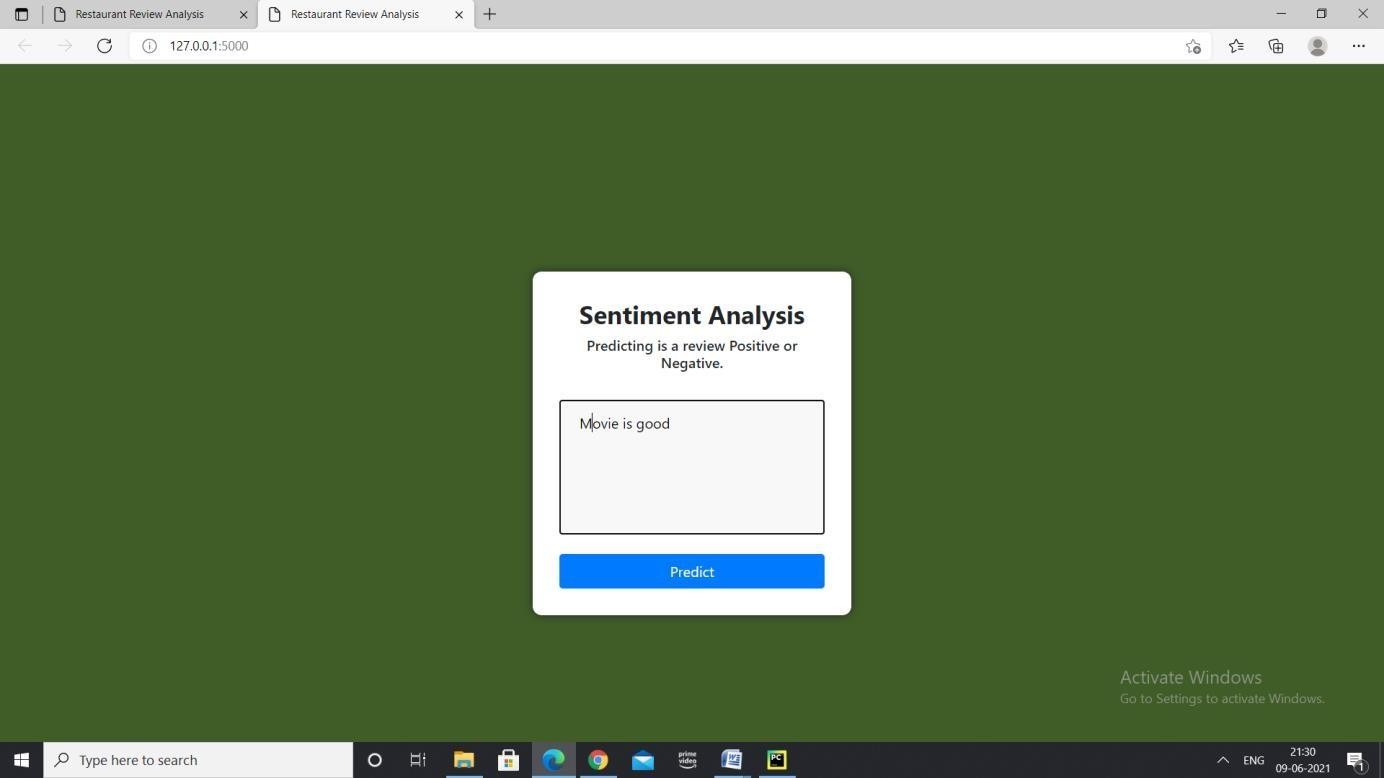
Artificial Neural Networks are computational models and inspire by the human brain. Many of the recent advancements have been made in the field of Artificial Intelligence, including Voice Recognition, Image Recognition, Robotics using Artificial Neural Networks. Artificial Neural Networks are the biologically inspired simulations performed on the computer to perform certain specific tasks like –

* Clustering
* Classification
* Pattern Recognization

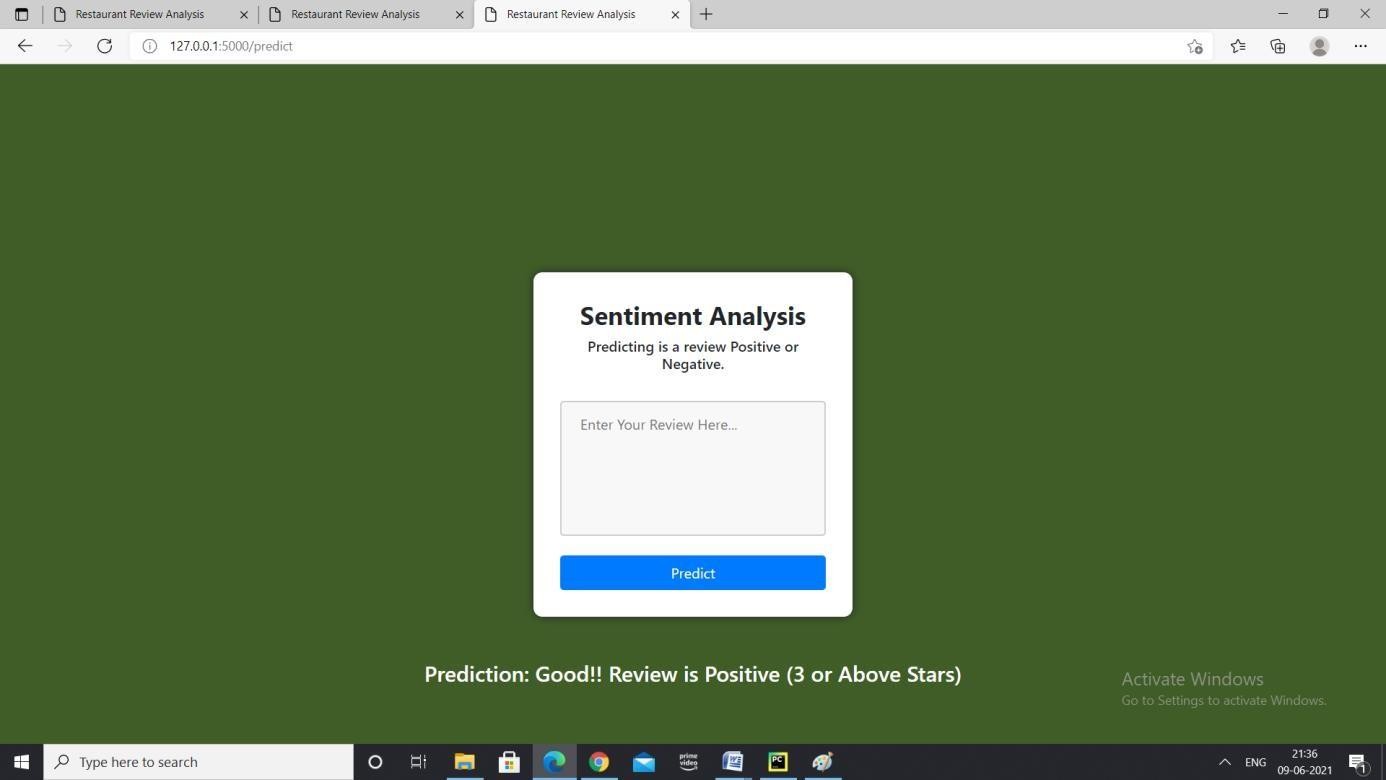




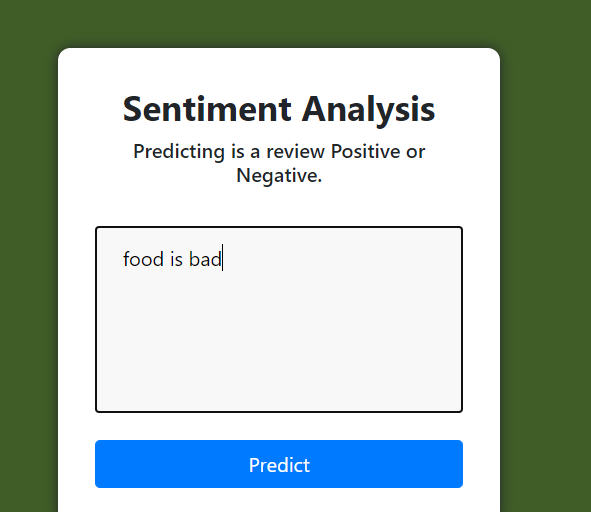




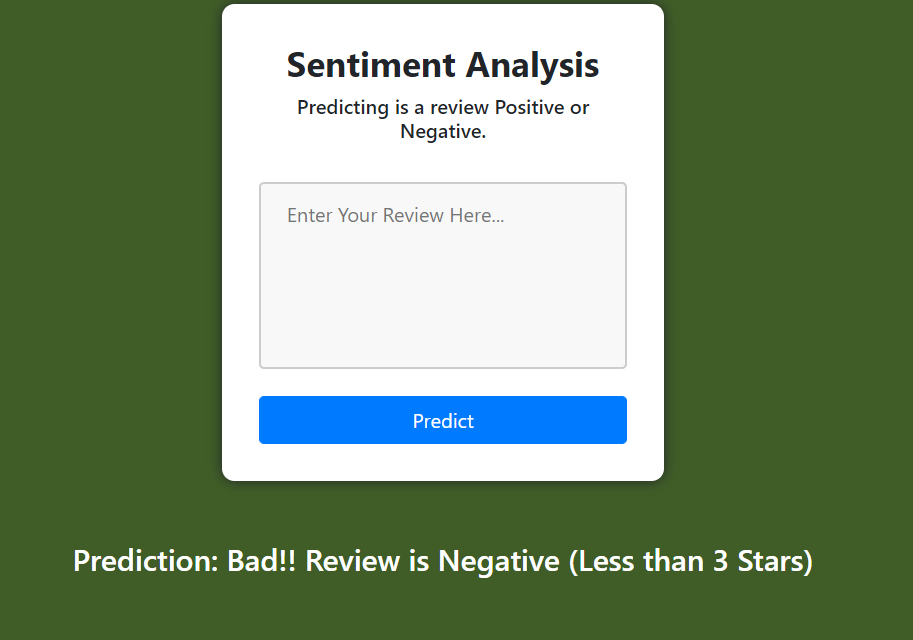
For the above review the output is:



For the review ”Food is Bad”



For the above review the output is



# CHAPTER 8 : CONCLUSION

Sentimental analysis of data collected from social media like Twitter , Facebook, Instagram is beneficial for mankind in providing them better health care. The workflow of analysis healthcare content in the social media helps to overcome the limitations of large scale data analysis and manual analysis of user generated textual content in social media.

This work can help the users to be updated with the effectiveness of the medicines and it can even suggest them with few better medications available. This project can provide feedback to the healthcare system organization and pharmaceutical companies for the available treatments and medicines. With the help of this project ,pharmaceutical companies and healthcare providers can work on the feedback and try to come up with the improvised medicines and treatments for diabetes. Users are provided with the resources of social media for the corresponding field of healthcare.

Opinion Mining and Sentiment analysis has wide area of applications and it also facing many research challenges. Since the fast growth of internet and internet related applications, the Opinion Mining and Sentiment Analysis become a most interesting research area among natural language processing community. A more innovative and effective techniques required to be invented which should overcome the current challenges faced by Opinion Mining and Sentiment Analysis.

# CHAPTER 9: FUTURE SCOPE AND ENHANCEMENT

In future , one can collect large healthcare related data from multiple social networking sites which may provide better results by overcoming the limitations of the project.

In future, one can even collect data which includes videos and images for analyzing and we can use more Deep Learning and Neural network to implement our project in Future.

# CHAPTER 10 : BIBLIOGRAPHY

1. Statistics-YouTube. [http://www.youtube.com/yt/press/statistics.html,](http://www.youtube.com/yt/press/statistics.html) 2013.
2. J. Friedman, T. Hastie, and R. Tibshirani.

The elements of statistical learning: Data Mining, Inference, and Prediction, Second Edition. Springer Series in Statistics, 2009.

1. G. Szabo and B. Huberman. Predicting the popularity of online content. Communic. of ACM, 53(8), 2010.

[4[ A Machine Learning Model for Stock Market Prediction Article by Osman Hegazy and Mustafa Abdul Salam.

[5]. The Unified Modeling Language User Guide, Low Price Edition Grady Booch, James Rumbaugh, Ivar Jacob**,**ISBN: 81-7808-769-5, 91, 1997.

[

6].The Elements of UMLTM 2.0, Scott Ambler-Cambridge University Press Newyork@2005,ISBN: 978-0-07-52616-7-82, 2005.

[7]. Software Testing,CemKaner, James beach,BretPettiehord,ISBN: 978-0-471-120-

940,2001.[8]. A Practitioner’s Approach Roger S.Pressman, Software Engineering, 3rd Edition,ISBN: 978-007-126782-3,482, 2014.

[9].Black\_Box Testing: Techniques for Functional Testing of Software and Systems Boris Beizer - Wiley Publications, ISBN: 978-0-471-120-940, 1995.

**CHAPTER-11**

**SOURCE CODE AND OUTPUTS:**

Diabetes Detection Model Documentation

**Introduction**

This document provides a detailed explanation of the steps followed to build a machine learning model for diabetes detection using the provided dataset.

**Step 1: Import required libraries**  
```python  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
from sklearn.model\_selection import train\_test\_split  
from sklearn.preprocessing import StandardScaler, LabelEncoder  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix, roc\_auc\_score  
from sklearn.model\_selection import GridSearchCV  
import squarify  
import joblib  
```

**Step 2: Load the dataset**  
```python  
# Load the dataset  
data = pd.read\_csv('/content /diabetes.csv')  
# Check the columns in the dataset  
print("Columns in the dataset:", data.columns)  
```

**OUTPUT:**

Columns in the dataset: Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',  
 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],  
 dtype='object')

**Step 3: Data Pre-Processing & Feature selection**

**Data Cleaning**  
```python  
# Dropping columns with more than 50% missing values  
data = data.dropna(thresh=data.shape[0]\*0.5, axis=1)  
```

**Filling Missing values**  
```python  
# Filling Missing values  
data = data.fillna(data.median(numeric\_only=True))  
```

**Noisy Data**  
```python  
# Assuming no specific noisy data handling needed for this dataset  
```

**Removal of outliers**  
```python  
for col in data.select\_dtypes(include=[np.number]).columns:  
 q1 = data[col].quantile(0.25)  
 q3 = data[col].quantile(0.75)  
 iqr = q3 - q1  
 lower\_bound = q1 - 1.5 \* iqr  
 upper\_bound = q3 + 1.5 \* iqr  
 data = data[(data[col] >= lower\_bound) & (data[col] <= upper\_bound)]  
```

**Transforming categorical variables into numerical variables**

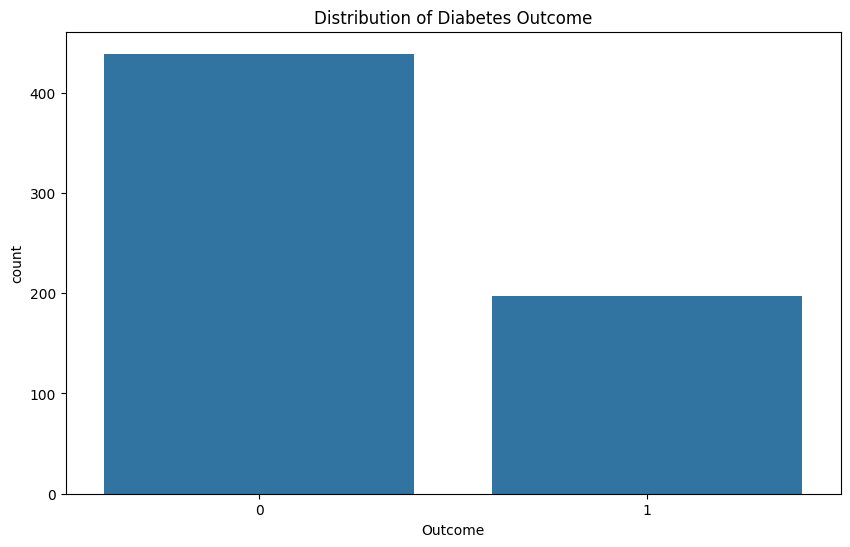
```python  
label\_encoders = {}  
for column in data.select\_dtypes(include=[object]).columns:  
 label\_encoders[column] = LabelEncoder()  
 data[column] = label\_encoders[column].fit\_transform(data[column].astype(str))  
```

**Data After Preprocessing:**

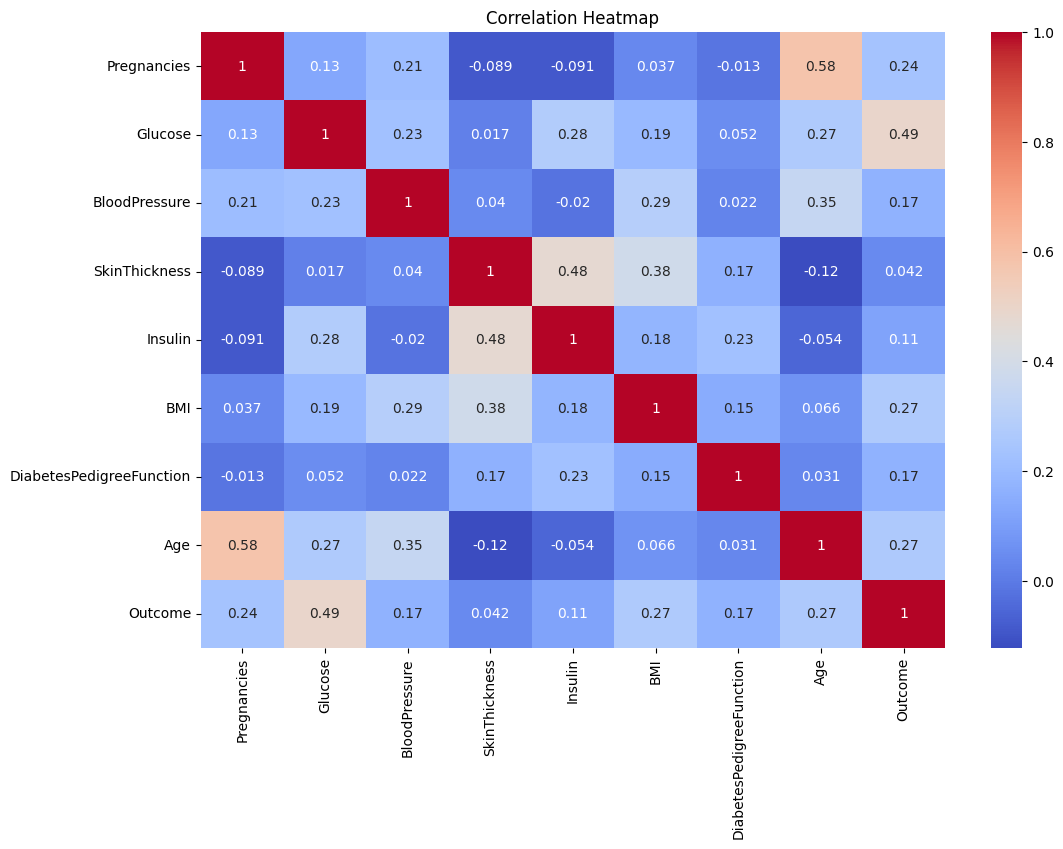
print(data)

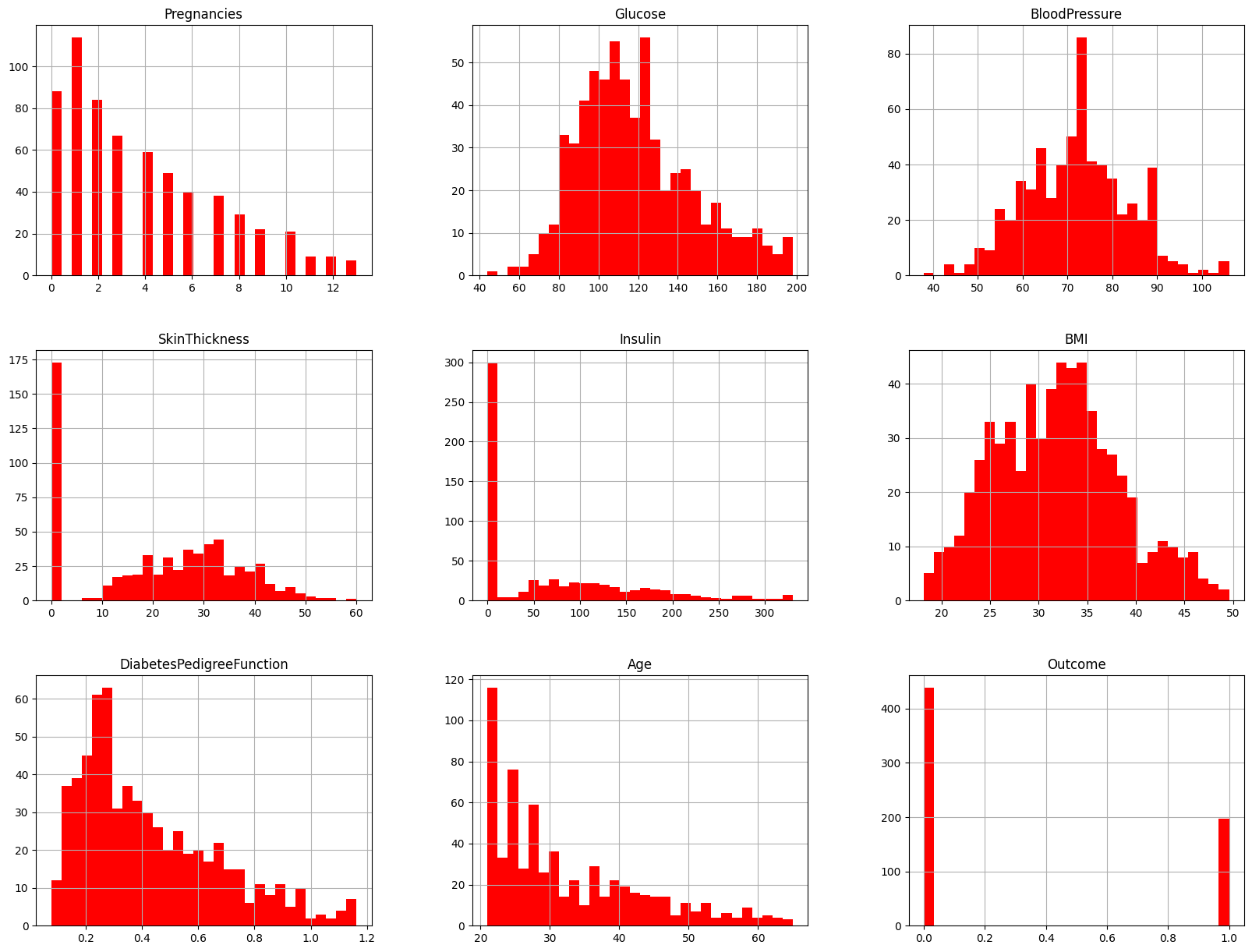
Pregnancies Glucose BloodPressure SkinThickness Insulin BMI \  
0 6 148 72 35 0 33.6   
1 1 85 66 29 0 26.6   
2 8 183 64 0 0 23.3   
3 1 89 66 23 94 28.1   
5 5 116 74 0 0 25.6   
.. ... ... ... ... ... ...   
763 10 101 76 48 180 32.9   
764 2 122 70 27 0 36.8   
765 5 121 72 23 112 26.2   
766 1 126 60 0 0 30.1   
767 1 93 70 31 0 30.4   
  
 DiabetesPedigreeFunction Age Outcome   
0 0.627 50 1   
1 0.351 31 0   
2 0.672 32 1   
3 0.167 21 0   
5 0.201 30 0   
.. ... ... ...   
763 0.171 63 0   
764 0.340 27 0   
765 0.245 30 0   
766 0.349 47 1   
767 0.315 23 0   
  
[636 rows x 9 columns]

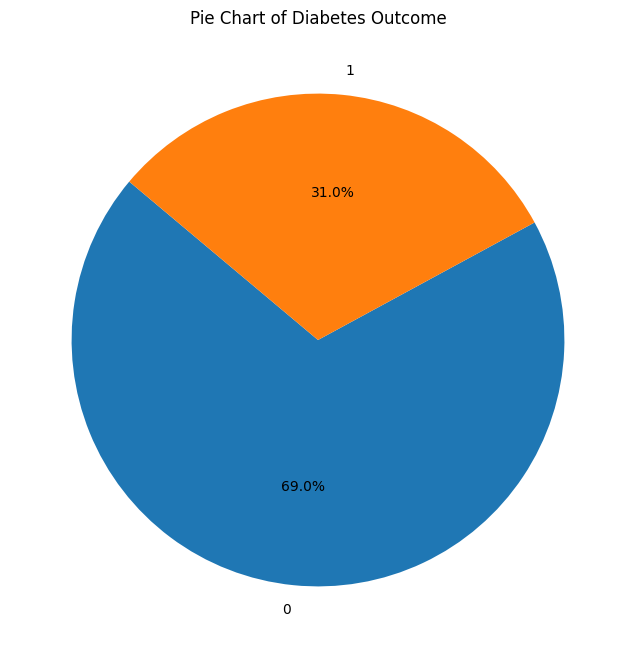
**Step 4: Data Visualization**

**Bar Chart**  
```python  
plt.figure(figsize=(10, 6))  
sns.countplot(x='Outcome', data=data)  
plt.title('Distribution of Diabetes Outcome')  
plt.show()  
```  


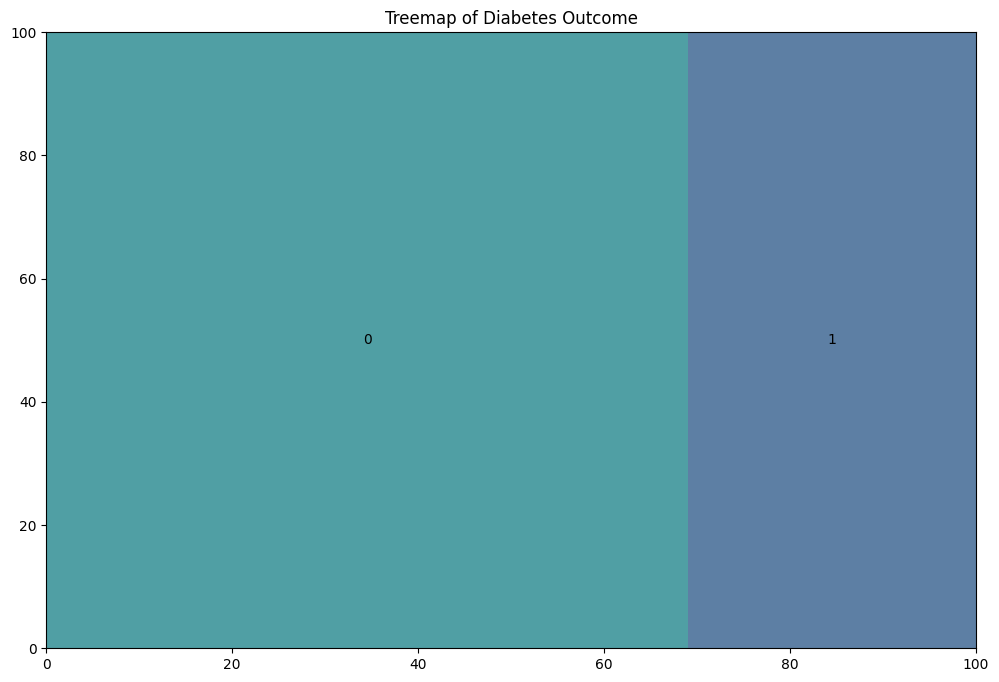
**Heat Map**  
```python  
plt.figure(figsize=(12, 8))  
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')  
plt.title('Correlation Heatmap')  
plt.show()

```

**Histogram**  
```python  
data.hist(bins=30, figsize=(20, 15), color='r')  
plt.show()  
```  


**Pie Chart**  
```python  
outcome\_counts = data['Outcome'].value\_counts()  
plt.figure(figsize=(8, 8))  
plt.pie(outcome\_counts, labels=outcome\_counts.index, autopct='%1.1f%%', startangle=140)  
plt.title('Pie Chart of Diabetes Outcome')  
plt.show()  
```

**Treemap**  
```python  
sizes = data['Outcome'].value\_counts().values  
labels = data['Outcome'].value\_counts().index  
plt.figure(figsize=(12, 8))  
squarify.plot(sizes=sizes, label=labels, alpha=.8)  
plt.title('Treemap of Diabetes Outcome')  
plt.show()  
```



**Step 5: Splitting and Training the data**

```python  
X = data.drop(columns=['Outcome'])  
y = data['Outcome']  
# Splitting the data  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
# Feature Scaling  
scaler = StandardScaler()  
X\_train = scaler.fit\_transform(X\_train)  
X\_test = scaler.transform(X\_test)  
```

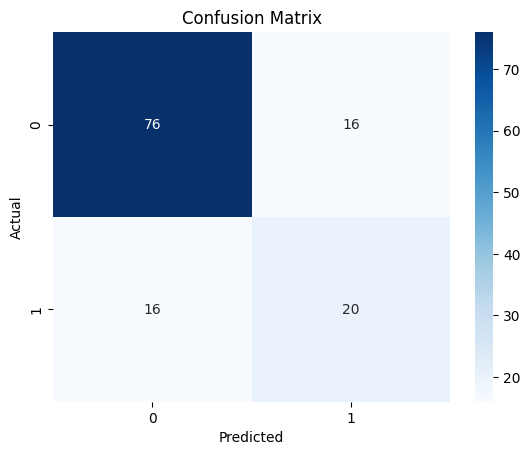
**Step 6: Load and train the model**  
```python  
# Hyperparameter Tuning using Grid Search  
param\_grid = {  
 'n\_estimators': [100, 200, 300],  
 'max\_features': ['sqrt', 'log2', None],  
 'max\_depth': [10, 20, 30, None],  
 'criterion': ['gini', 'entropy']  
}  
grid\_search = GridSearchCV(estimator=RandomForestClassifier(random\_state=42), param\_grid=param\_grid, cv=5, n\_jobs=-1, scoring='roc\_auc')  
grid\_search.fit(X\_train, y\_train)  
# Best model from Grid Search  
best\_model = grid\_search.best\_estimator\_  
# Training the best model  
best\_model.fit(X\_train, y\_train)  
```

**Step 7: Evaluating the model**  
```python  
y\_pred = best\_model.predict(X\_test)  
accuracy = accuracy\_score(y\_test, y\_pred)  
roc\_auc = roc\_auc\_score(y\_test, y\_pred)  
print(f'Accuracy: {accuracy \* 100:.2f}%')  
print(f'ROC-AUC Score: {roc\_auc:.2f}')  
if accuracy >= 0.75:  
 print('Model accuracy is satisfactory.')  
else:  
 print('Model accuracy is not satisfactory. Consider changing the algorithm.')  
# Print classification report  
print(classification\_report(y\_test, y\_pred))  
# Print confusion matrix  
conf\_matrix = confusion\_matrix(y\_test, y\_pred)  
sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues')  
plt.xlabel('Predicted')  
plt.ylabel('Actual')

plt.title('Confusion Matrix')  
plt.show()  
```

OUTPUT:

Accuracy: 75.00%  
ROC-AUC Score: 0.69  
Model accuracy is satisfactory.  
 precision recall f1-score support  
  
 0 0.83 0.83 0.83 92  
 1 0.56 0.56 0.56 36  
  
 accuracy 0.75 128  
 macro avg 0.69 0.69 0.69 128  
weighted avg 0.75 0.75 0.75 128



**Step 8: Build the Predictive Model**  
The model is already built and evaluated in steps 6 and 7.

**Step 9: Deploy the model**  
```python  
# Saving the model to disk  
joblib.dump(best\_model, 'diabetes\_model.pkl')  
```  
OUTPUT:

['diabetes\_model.pkl']