DIABETES PREDICTION USING DATA SCIENCE IN PYTHON

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Computer Science

By

Name: NAVEENRANGASAMY B

(REGISTER NO 40290060)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600199

MAY 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **NAVEENRANGASAMY B (40290060)** who carried out the project entitled **"DIABETES PREDICTION USING DATA SCIENCE IN PYTHON"** under my supervision from January 2023 to May 2023.

29/4

INTERNAL GUIDE

Dr. V.ULAGAMUTHALVI M.E., Ph.D.

Rekha Chakravarthi

Dean and Head of the Department

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva Voce examination held on 08/05/2023

8/5/2023

Internal Examiner

8/5/23

External Examiner

DECLARATION

I **NAVEENRANGASAMY B** hereby declare that the Project Report entitled **"DIABETES PREDICTION USING DATA SCIENCE IN PYTHON"** done by me under the guidance of **Dr.V.ULAGAMUTHALVI M.E., Ph.D.** at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Computer Science.

DATE: 08.05.2023

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ABSTRACT

This is a classification problem of supervised machine learning. The objective is to predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Diabetes is a common chronic disease and poses a great threat to human health. The characteristic of diabetes is that the blood glucose is higher than the normal level , which is caused by defective insulin secretion. Diabetes can lead to chronic damage and dysfunction of various tissues, especially eyes, kidneys, heart, blood vessels and nerves. With the development in living standards, diabetes is increasingly common in our day-to-day life.

Therefore, how to quickly and accurately diagnose and analyze diabetes is a topic worthy studying. Machine learning can help people make a preliminary judgment about diabetes according to their daily physical examination data, and it can serve as a reference for doctors . For machine learning the selection of valid features and the correct classifier are the most important problems. So in this study, LogisticRegression and DecisionTree are implemented to predict the diabetes.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

This is a classification problem of supervised machine learning. The objective is to predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Diabetes is a common chronic disease and poses a great threat to human health. The characteristic of diabetes is that the blood glucose is higher than the normal level, which is caused by defective insulin secretion. Diabetes can lead to chronic damage and dysfunction of various tissues, especially eyes, kidneys, heart, blood vessels and nervesWith the development in living standards, diabetes is increasingly common in our day-to-day life. Therefore, how to quickly and accurately diagnose and analyze diabetes is a topic worthy studying. Machine learning can help people make a preliminary judgment about diabetes according to their daily physical examination data, and it can serve as a reference for doctors . For machine learning the selection of valid features and the correct classifier are the most important problems. So in this study, LogisticRegression and DecisionTree are implemented to predict the diabetes.

1.1Introduction to MACHINE LEARNING USING PYTHON

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data. In this article, we'll see basics of Machine Learning, and implementation of a simple machine-learning algorithm using python.

Machine learning is a method of teaching computers to learn from data, without being explicitly programmed. Python is a popular programming language for machine learning because it has a large number of powerful libraries and frameworks that make it easy to implement machine learning algorithms.

Python offers multiple great graphing libraries that come packed with lots of different features. No matter if you want to create interactive, live or highly

customized plots python has an excellent library for you.

To get a little overview here are a few popular plotting libraries:

MATPLOTLIB: It is a multi-platform data visualization library built on NumPy arrays, and designed to work with the broader SciPy stack. It was conceived by John Hunter in 2002, originally as a patch to I Python for enabling interactive MATLAB-style plotting via gnu plot from the I Python command line.

PLOTLY: Plotlib is the Python Library for interactive data visualizations. Plt allows you to plot superior interactive graphs than either Matplotlib or Seaborn.

1.2 Need for the MACHINE LEARNING

The field of artificial intelligence known as machine learning is concerned with the process by which computers attempt to forecast future events using historical data and the information they already possess. There are two distinct forms of machine learning. The first kind of learning is called supervised learning, and in this type of learning, the data itself serve as the instructor, and the system is constructed based on the dataset. The second kind of learning is called unsupervised learning, and it involves the data teaching itself by identifying certain patterns within the dataset and then categorising those patterns. Over the last several years, a large number of writers have reported and discussed their research on diabetes prediction by utilising machine learning algorithms.

1.3 MACHINE LEARNING ALGORITHM

The research that has been done on machine learning has resulted in the development of multiple data mining methods. These algorithms may be directly applied to a dataset in order to create some predictions or to derive important inferences and conclusions from such a dataset. Immediate use of these algorithms is possible. Decision tree,NaivE Bayes, k-means, neural network, and other similar algorithms are examples of prominent data mining techniques. In the part that comes after this one, we will talk about them.

The **Naive Bayes (NB)** method is a straightforward approach to the construction of classifiers. It is a Bayesian probabilistic classifier that uses Bayes' theorem as its foundation. Given the class variable, each and every Naive Bayes classifier operates on the assumption that the value of a single feature does not rely in any way on the value of any additional feature. The following statement illustrates Bayes theorem: Here X is the data tuple and C is the class, P(C|X) is calculated by multiplying P(X|C) by P(C)/P(X). This ensures that P(X) remains the same for all classes. Despite the fact that it operates on the assumption that feature values are critically independent, which is not a realistic assumption, it performs very well on huge datasets when this requirement is assumed and is true.

A **decision tree** is a kind of decision support system that utilises a tree-like structure or representation of actions and their potential repercussions, which may include the results of chance events as well as utility considerations. It is one of the methods in which an algorithm may be shown. In the field of operational research, decision trees are often used, particularly in the process of decision analysis, to assist with the identification of a strategy that would most likely result in the achievement of the objective. In the field of machine learning, it is also a widely used tool. The process of mapping first from tree's parent node to each of its leaf nodes individually may quickly and efficiently convert a decision tree to a collection of rules. When everything is said and done, reasonable judgments may

be obtained by adhering to these criteria.

Support Vector Machine (SVM) - an abbreviation for It is a kind of learning called supervised learning, and it divides data into two categories using a hyper plane as the dividing line. The Support Vector Machine (SVM) accomplishes the same goal as C4.5, with the exception that it does not make any use of Decision Trees. In order to reduce the likelihood of an incorrect classification being made, the support vector machine makes an effort to increase the margin, which is the distance across the hyper plane and the 2 data points that are closest to it from each class. Scikit-learn, MATLAB, and LIBSVM are examples of well-known software packages that may be used to create support vector machines.

A **Network Made of Artificial Neurons (ANN)** A computer model that replicates the architecture and functionality of biological neurons is referred to as an artificial neural network, or ANN for short. Since a neural network adapts or trains in some manner depending on the outputs and inputs, for that specific stage and subsequently for each phase, data that travels thru the network has an effect on the architecture of the ANN. ANNs are recognised as nonlinear statistical data modelling software because of their ability to simulate the intricate interactions that exist across inputs and outputs or to identify trends. ANNs are built using layers that are linked to one another. The artificial neural networks that are being used to improve current data analytics tools are very straightforward mathematical models.

1.4 INTRODUCTION TO PYTHON FOR DATA ANALYTICS

Python is a high level, dynamic programming language. Python3.4 version was used as it is a mature, versatile and robust programming language. It is an interpreted language which makes the testing and debugging extremely quickly as there is no compilation step. There are extensive open source libraries available for this version of python and a large community of users. Python is simple yet powerful, interpreted and dynamic programming language, which is well known for its functionality of processing natural language data, i.e. spoken English using NLTK. Other high level programming languages such as R and Matplotlib were considered because they have many benefits such as ease of use but they do not offer the same flexibility and freedom that Python can Deliver.

Features in Python

There are many features in Python, some of which are discussed below

- Easy to code.
- Free and Open Source.
- Object-Oriented Language.
- GUI Programming Support.
- High-Level Language.
- Extensible feature.
- Python is Portable language.
- Python is Integrated language.

1.5 ANACONDA

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, [12] as a graphical alternative to the Command Line Interface (CLI).

The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists.

When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages. It will install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail.

Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed.

Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories.

The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.

1.6 ANACONDA NAVIGATOR

Anaconda Navigator is a desktop Graphical User Interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda

Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab.
- Jupyter Notebook.
- QtConsole.
- Spyder.
- Glue.
- Orange.
- RStudio.
- Visual Studio Code.

1.7 JUPYTER NOTEBOOK

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. Usually ending with the ".ipynb" extension. Jupyter Notebook can connect to many kernels to allow programming in different languages. By default, Jupyter Notebook ships with the IPython kernel. As of the 2.3 release[11][12] (October 2014), there are currently 49 Jupyter-compatible kernels for many programming languages, including Python, R, Julia and Haskell. The Notebook interface was added to IPython in the 0.12 release[14] (December 2011), renamed to Jupyter notebook in 2015 (IPython 4.0 – Jupyter 1.0). Jupyter Notebook is similar to the notebook interface of other programs such as Maple, Mathematica, and SageMath, a computational interface style that originated with Mathematica in the 1980s. According to The Atlantic, Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.

Pandas is an open-source Python Library providing high-performance data

manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data. In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data. Prior to Pandas, Python was majorly used for data munging and preparation. It had very less contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data—load, prepare, manipulate, model and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Key Features of Pandas

- Fast and efficient Data Frame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and sub-setting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

Matplotlib is the most popular python plotting library. It is a low-level library with a Matlab like interface which offers lots of freedom at the cost of having to write more code.

Key features of Matplotlib

- Semantic way to generate complex, subplot grids.
- Settings the aspect ratio of the axes box.
- Colored labels in legends.
- Ticks and labels.
- RcParams can be passed as Decorators.
- 3D plots now support minor ticks.

Plotly allows users to import, copy and paste, or stream data to be analyzed and visualized. For analysis and styling graphs, Plotly offers a Python sandbox (NumPy supported), datagrid, and GUI. Python scripts can be saved, shared, and collaboratively edited in Plotly.

Key features of Plotly

- Charts.
- Dashboards.
- File Export.
- App Manager.
- Kubernetes Authentication.
- Jobs Queue.
- Snapshot Engine.

CHAPTER 2
LITRATURE SURVEY

Birjais *et al*.[8] experimented on PIMA Indian Diabetes (PID) data set. It has 768 instances and 8 attributes and is available in the UCI machine learning repository. They aimed to focus more on diabetes diagnosis, which, according to the World Health Organization (WHO) in 2014, is one of the world's fastest-growing chronic diseases. Gradient boosting, logistic regression, and naive Bayes classifiers were used to predict whether a person is diabetic or not, with gradient boosting having an accuracy of 86%, logistic regression having a 79% accuracy, and naive Bayes having a 77% accuracy.

Sadhu, A. and Jadli A.[9] experimented on a diabetes data set taken from the UCI repository. There were 520 occurrences and 16 attributes in all. They attempted to concentrate their efforts on predicting diabetes at an early stage. On the validation set of the employed data set, seven classification techniques were implemented: k-NN, logistic regression, SVM, naive Bayes, decision tree, random forests, and multilayer perceptron. The random forests classifier proved to be the best model for the concerned data set, with an accuracy score of 98%, followed by logistic regression at 93%, SVM at 94%, naive Bayes at 91%, decision tree at 94%, random forests at 98%, and multilayer perceptron at 98%, according to the results of training several machine learning models.

Xue *et al*.[10] experimented on the diabetes data set taken from the UCI repository; there were 520 patients and 17 qualities in it. They attempted to concentrate on early detection of diabetes. They trained on the actual data of 520 diabetic patients and probable diabetic patients aged 16–90 using supervised ML techniques such as SVM, naive Bayes classifiers, and LightGBM. The performance of the SVM is the best when comparing classification and recognition accuracy. The naive Bayes classifier is the most widely used classification algorithm, with an accuracy of 93.27%. SVM has the highest accuracy rate of 96.54%. LightGBM

has an accuracy of only 88.46%. This demonstrates that SVM is the best classification algorithm for diabetes prediction.

Le et al.[11] experimented on the early-stage diabetes risk prediction; the data set used in this research was taken from the UCI repository and consisted of 520 patients and 16 variables. They suggested a ML approach for predicting diabetes patients' early onset. It was a new wrapper-based feature selection method that employed grey wolf optimizer (GWO) and adaptive particle swarm optimization (APSO) to optimize the multilayer perceptron (MLP) and reduce the number of needed input attributes. They also compared the results obtained with this method to those obtained via a variety of traditional machine learning algorithms, including SVM, DT, k-NN, naive Bayes classifier (NBC), random forest classifier (RFC), and logistic regression (LR). LR achieved a 95% accuracy rate. k-NN had a 96% accuracy rate, SVM a 95% accuracy rate, NBC a 93% accuracy rate, DT a 95% accuracy rate, and RFC had a 96% accuracy rate. The suggested methods' computational findings show that not only are fewer features required but also that higher prediction accuracy may be attained (96% for GWO–MLP and 97% for APSO–MLP). This research has the potential to be applied in clinical practice and used as a tool to assist doctors and physicians.

Julius et al.[12] used the Waikato Environment for Knowledge Analysis (Weka) application platform to test a data set collected from the UCI repository. There were 520 samples in the data set, each with a collection of 17 attributes. The goal of this study was to use machine learning classification approaches based on observable sample attributes to predict diabetes at an early stage. The k-NN, SVM, functional tree (FT), and RFCs were employed as classifiers. k-NN had the highest accuracy of 98%, followed by SVM at 94%, FT at 93%, and RF at 97%.

Shafi et al.[13] reported that because diabetes is a serious illness, early detection is always a struggle. This study used machine learning classification methods to develop a model that could solve any problem and that could be used to identify

diabetes development early on. The authors of this research made concerted efforts to develop a framework that could accurately predict the likelihood of diabetes in patients. As part of this study, the three ML approach classification algorithms—DT, SVM, and NBC—were studied and assessed on various measures. In the study, the PID data set acquired from the UCI repository was used to save time and produce precise findings. The experimental results suggested that the NBC approach was adequate, with a 74% accuracy, followed by SVM with a 63% accuracy and the DT with a 72% accuracy. In the future, the built framework, as well as the ML classifiers used, could be used to identify or diagnose other diseases. The study, as well as several other ML methodologies, could be extended and improved for diabetes research, and the scientists intended to classify other algorithms with missing data.

Khanam et al.[14] experimented with diabetes illness prediction. Diabetes is a condition with no known cure; therefore early detection is essential. In this study, data mining, ML techniques, and neural network (NN) methodologies were utilized to predict diabetes. They developed a technique that could accurately predict diabetes. They used data from the UCI repository's PID data set. The data set included information on 768 patients and their 9 attributes. On the data set, they utilized seven ML methods to predict diabetes: DT, k-NN, RFC, NBC, AB, LR, and SVM. They used the Weka tool to preprocess the data. They discovered that a model combining LR and SVM is effective at predicting diabetes. They created a NN model with two hidden layers and varied epochs and found that the NN with two hidden layers gave 88.6% accuracy. ANN scored 88.57%, LR scored 78.85%, NBC scored 78.28%, and RFC scored 77.34%.

Sisodia et al.[15] used the PID data set available on the UCI repository. This data set contained 768 patients and 8 attributes. They employed three ML classifications to identify diabetic patients: DT, SVM, and NBC. NBC had the highest accuracy (76.30%) when compared to the other models.

Agarwal et al.[16] used the PID data set of 738 patients as well in their study. To analyze the effectiveness of this data set for identifying diabetic patients, the authors applied models such as SVM, k-NN, NBC, ID3, C4.5, and CART. The SVM and LDA algorithms were the most accurate, with an accuracy of 88%.

Rathore et al.[17] employed classification techniques like SVM and DTs to predict diabetes mellitus. The PID data set provided the data for this investigation. PIMA India prioritizes women's health. The SVM has an accuracy of 82%.
To predict diabetes mellitus, Hassan et al.[18] employed classification approaches such as the DT, k-NN, and SVM. The SVM outperformed the DT and KNN methods with a maximum accuracy of 90.23%.

Kandhasamy and Balamurali[19] investigated the prediction accuracy of J48, k-NN, RFC, and SVM on the diabetes data set. Before preprocessing the data, the author discovered that the J48 method had a higher accuracy than others, at 73.82%. After preprocessing, k-NN and RFC demonstrated improved accuracy.

Meng et al.[20] examined J48, LR, and k-NN algorithms on the diabetes data set. J48 was found to be the most accurate, with a classification accuracy of 78.27%.

Nai-Arun and Moungmai[21] created a web application based on the prediction accuracy for diabetes prediction. They compared prediction methods such as DTs, NNs, LR, NBC, and RFC, as well as, bagging and boosting. They discovered that RFC performed best in terms of accuracy and ROC score, with an accuracy of 85.558% and an ROC value of 0.912.

Saravananathan and Velmurugan[22] looked at J48, CART, SVM, and k-NN on a medical data set in their research. They compared them based on accuracy, specificity, sensitivity, precision, and error rate. With a score of 67.15%, they discovered that J48 algorithms were the most accurate, followed by SVM (65.04%), CART (62.28%), and k-NN (53.39%).

Kumari and Chitra[23] used SVM, RFC, DT, MLP, and LR, as well as four k-fold cross-validations (k = 2,4,5,10) in their research. According to the researchers, MLP with four-fold cross-validation achieves the best accuracy, at 78.7%. They discovered that MLP outscored all other algorithms.

To predict diabetes, Kavakiotis et al.[24] employed NBC, RFC, k-NN, SVM, DT, and LR methods. The algorithms were applied using a ten-fold cross-validation technique. SVM had the best accuracy of all the approaches, measuring 84%, according to the study.

The work on the classification of “Diabetes Prediction” based on eight attributes was done by Rawat et al.[25] In this study, five ML algorithms for the analysis and prediction of diabetic patients were described: AdaBoost, LogicBoost, RobustBoost, naive Bayes, and bagging. A group of diabetic PIMA Indians was used to test the proposed strategies. The computed results were found to be quite accurate, with a classification accuracy of 81.77% and 79.69% for the bagging and AdaBoost techniques, respectively. As a result, the proposed DM prediction algorithms were particularly appealing, effective, and efficient.

Perveen et al.[26] used a data set from the Canadian Primary Care Sentinel Surveillance Network (CPCSSN) database to do their research. The study employed the AdaBoost and bagging ensemble techniques using the J48 (C4.5) DT as a base learner and standalone data mining methodology J48 to categorize patients with diabetes mellitus based on diabetes risk indicators. This categorization was done across three separate ordinal adult groups in the CPCSSN. In terms of overall performance, the AdaBoost ensemble method surpassed both bagging and a single J48 DT, according to the findings.

Mujumdar and Vaidehi[27] presented a diabetes prediction model for better diabetes classification that included a few extrinsic factors that caused diabetes,

as well as regular components such as glucose, BMI, age, insulin, and so on. The new data set enhanced classification accuracy when compared to the old data set. Multiple ML approaches were used on the data set, and classification was done with a variety of algorithms, with LR yielding the highest accuracy at 96%. The AdaBoost classifier was found to be the most accurate, with a 98.8% accuracy rate. They used two separate data sets to compare the accuracy of ML techniques. When compared to the existing data set, it was clear that the model improved diabetes prediction accuracy and precision.

Mercaldo et al.[28] offered a strategy for classifying diabetic patients based on a set of features chosen according to the WHO criteria. Evaluating real-world data using state of the art machine learning algorithms. The model was trained using six alternative classification approaches, with the Hoeffding Tree method scoring 0.770 in precision and 0.775 in recall. They used data from the PIMA Indian community in Phoenix, Arizona, to evaluate the method.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 AIM: To accurately predict whether or not the patients in the dataset have diabetes or not.

3.2 SCOPE: The accuracy of the model is obtained by comparing the predicted values against the original set of values. Identifying diabetics or predicting the upcoming of a diabetic life can be propelled by using various machine learning techniques.

In this project we are going to use a dataset named Diabetes.csv file which contains the details such as, Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age and Outcome.

3.3 WHAT IS DIABETES

Diabetes is a condition that is brought on by having an abnormally high level of blood glucose in the body. Human bodies are in constant need of power, and sugar is one of the primary sources of vitality that is used in the construction of our muscles and other tissues. In individuals, the primary reasons of type 2 diabetes are often an unhealthy habit combined with a lack of physical activity. Diabetes is a condition that is brought on by an abnormally high level of glucose in the bloodstream. Diabetes occurs when the pancreas is failing to turn the meal into insulin; as a result, sugar is not taken into the body, leading to the condition. Diabetes may cause problems in a variety of body systems, including the kidneys, eyes, neurological system, arteries, and so on.

OBJECTIVE: The main objective of dataset is to classify that whether a particular person has diabetes or not with help of several independent variables given in

dataset.

Details about the dataset:

The datasets consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

- Pregnancies: Number of times pregnant
- Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- BloodPressure: Diastolic blood pressure (mm Hg)
- SkinThickness: Triceps skin fold thickness (mm)
- Insulin: 2-Hour serum insulin (mu U/ml)
- BMI: Body mass index (weight in kg/(height in m)^2)
- DiabetesPedigreeFunction: Diabetes pedigree function
- Age: Age (years)
- Outcome: Class variable (0 or 1)

CHAPTER 4

EXPERIMENTAL OR REQUIREMENTS AND METHODS

4.1 SYSTEM SPECIFICATION

Hardware Requirements

1. Processor – Intel Core processors or any AMD chips.
2. RAM – 1 GB
3. Hard Disk – 40GB
4. Operating System – Windows 7 and above

Software requirements

1. Jupyter notebook (anaconda 3) or Google collab.
2. Python 3 or latest version.

4.2 PROJECT DESCRIPTION

We are going to predict diadetes via three different supervised machine learning methods including:SVM,Logistic regression,KNN. We have got this dataset from Kaggle. We are going to analyze the information using Logistic Regression in this dataset and try to present in a visual manner using different libraries in python programming language like numpy,Pandas and matplotlib.

4.3 LIBRARIES USED

We will generally be working on this project using the standard libraries such as:

- Sklearn
- Pandas
- Numpy
- Matplotlib

Sklearn is the root important library from which we are going to import and make use of the algorithms or the metrics or the train_test_split for making predictions using the model generated using thee.

Pandas is the standard library one would be dealing with while doing a machine learning project. As I have mentioned above that I will be using the data set which is stored in an excel file, we can use pandas to read the excel file.

Similarly we will use Numpy to make use of the numpy arrays, while we are calculating the errors or handling the data of some kind.

Matplotlib is one of the major necessities as we are going to use it for creating plots and to analyze our predictions based on that.

We will break down more about the libraries and their usage once we start using them. I will try to describe that particular usage and its overall description too once we get into the coding part. As we have seen that the above are the necessary libraries which we will import as full but for some we will only import the necessary modules (just to have a quick knowledge of exactly what modules we will be using).

We will import a few specific functions and algorithms from modules of Sklearn library, they are namely:

- train_test_split from model_selection module of sklearn
- from sklearn.linear_model import LogisticRegression
- metrics module of sklearn

Model_selection is a method for setting a blueprint to analyze data and then using it to measure new data. Selecting a proper model allows you to generate accurate results when making a prediction.

train_test_split is a function in sklearn model selection for splitting data arrays into two subsets: for training data and for testing data.

With this function you don’t need to divide the dataset manually. By default, sklearn train_test_split will make random partitions for the two subsets. However, you can also specify a random state for the operation.

4.4 Logistic Regression

The predictive analysis which is used for the dependent variable is categorical called as Logistical Regression. Logistical Regression explains the relationship between one dependent variable and one or more independent variables.

System Architecture/ Ideation Map:

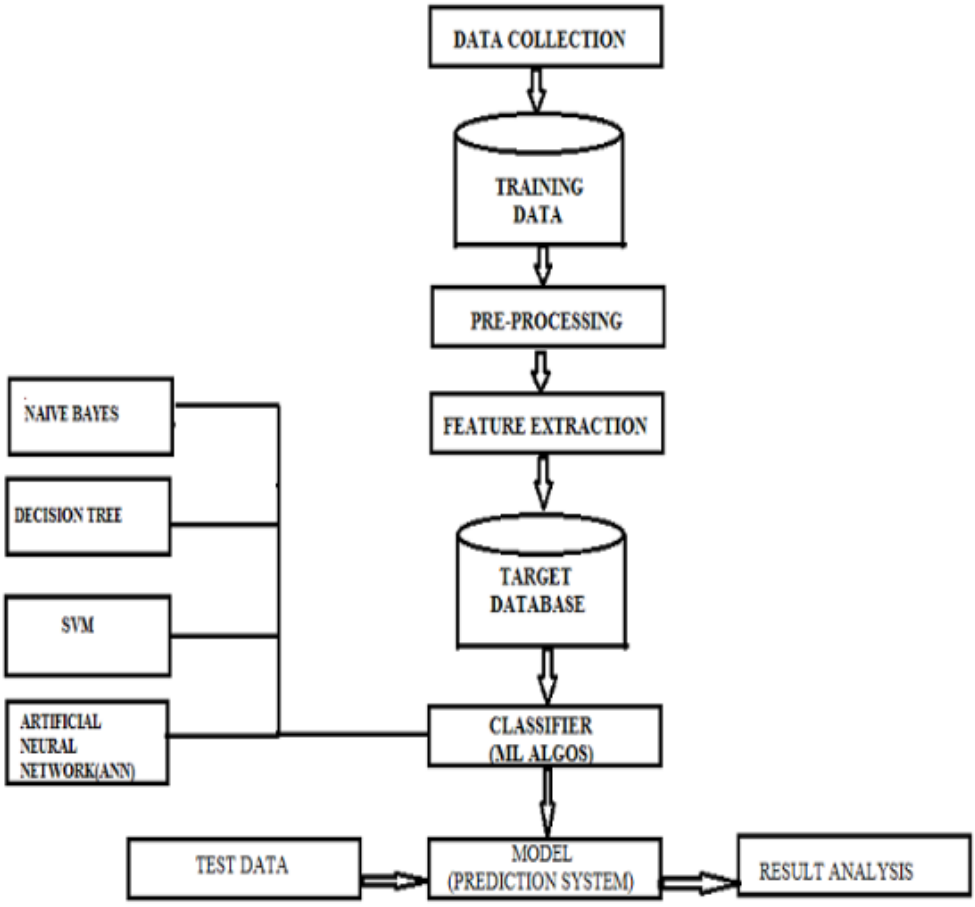


Fig 4.1: System Architecture

Project Implementation:

Steps followed during the implementation

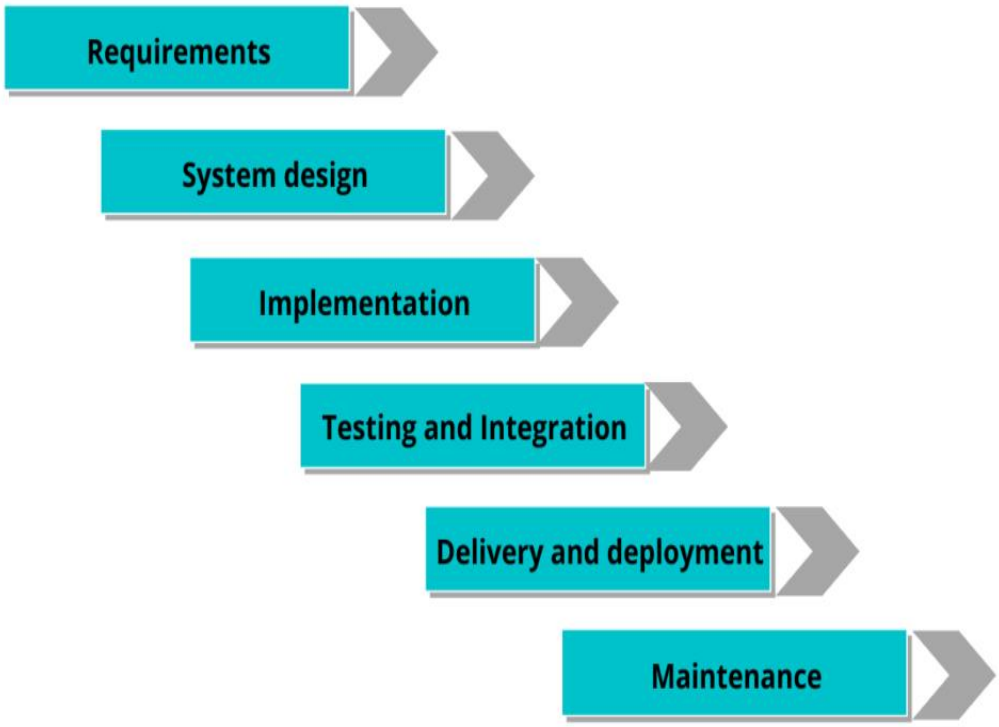


Fig 4.2: Project Implementation

Step1:

Went on a search for a diabetes dataset with all the labels given.

Step2:

Then made a flow char of how to implement the diabetes predicting system and made some conclusions like ,What are the ML algorithms or techniques that would perfectly fit the problem with good accuracy.

Step3:

Next installation of required software's like Anaconda ,Jupyter, Python.

Step4:

After installing the required software's ,started with Anaconda lunched Jupyter notebook with python 3.

Step5:

Now importing all the necessary libraries like Pandas,Numpy,Sklearn,Matplotlib and all the required modules.

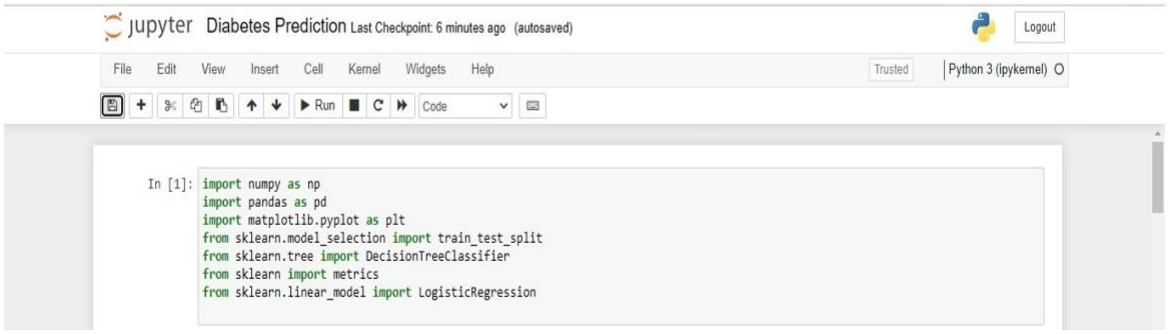


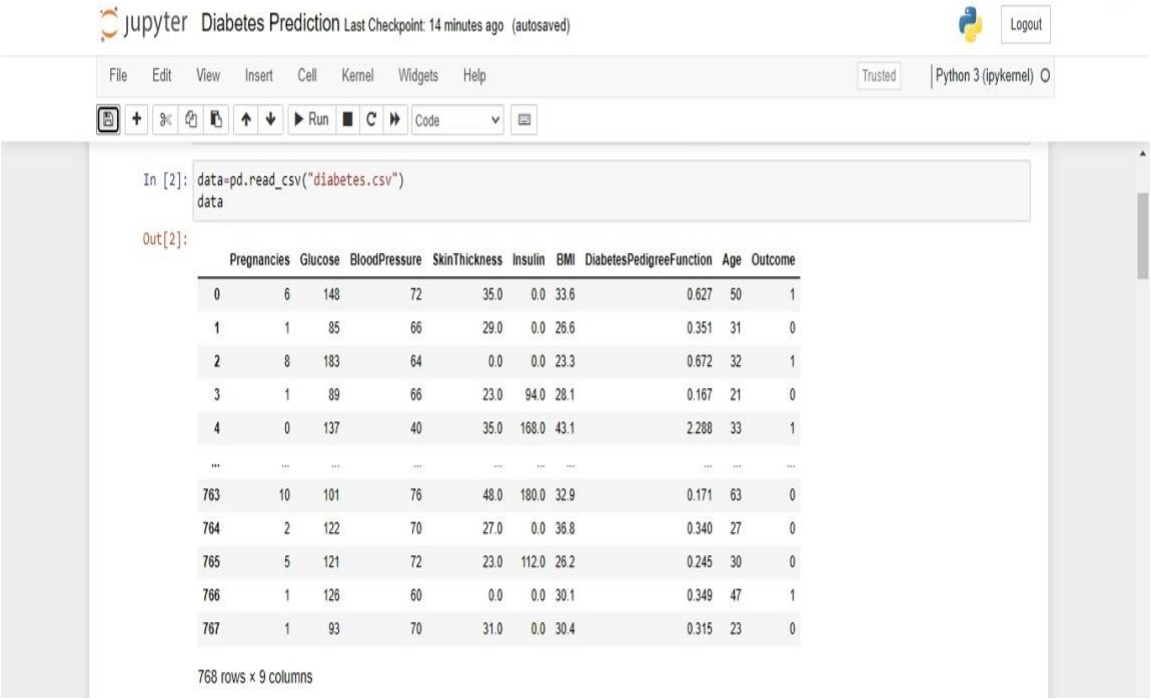
Fig 4.3: code of importing libraries

Step6:

Import the diabetes Dataset into Jupyter notebook.

Step7:

Now we need to see the structure of the data like how many columns and rows are present.



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35.0	0.0	33.6	0.627	50	1
1	1	85	66	29.0	0.0	26.6	0.351	31	0
2	8	183	64	0.0	0.0	23.3	0.672	32	1
3	1	89	66	23.0	94.0	28.1	0.167	21	0
4	0	137	40	35.0	168.0	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48.0	180.0	32.9	0.171	63	0
764	2	122	70	27.0	0.0	36.8	0.340	27	0
765	5	121	72	23.0	112.0	26.2	0.245	30	0
766	1	126	60	0.0	0.0	30.1	0.349	47	1
767	1	93	70	31.0	0.0	30.4	0.315	23	0

768 rows × 9 columns

Fig 4.4: code for importing data set

Step8:

Then started looking for missing values and filling those with mean of the particular column (here we have 2 missing values one in SkinThickness and the other on Insulin)

Fig 4.5: Example code for the pre-work of data set with null values

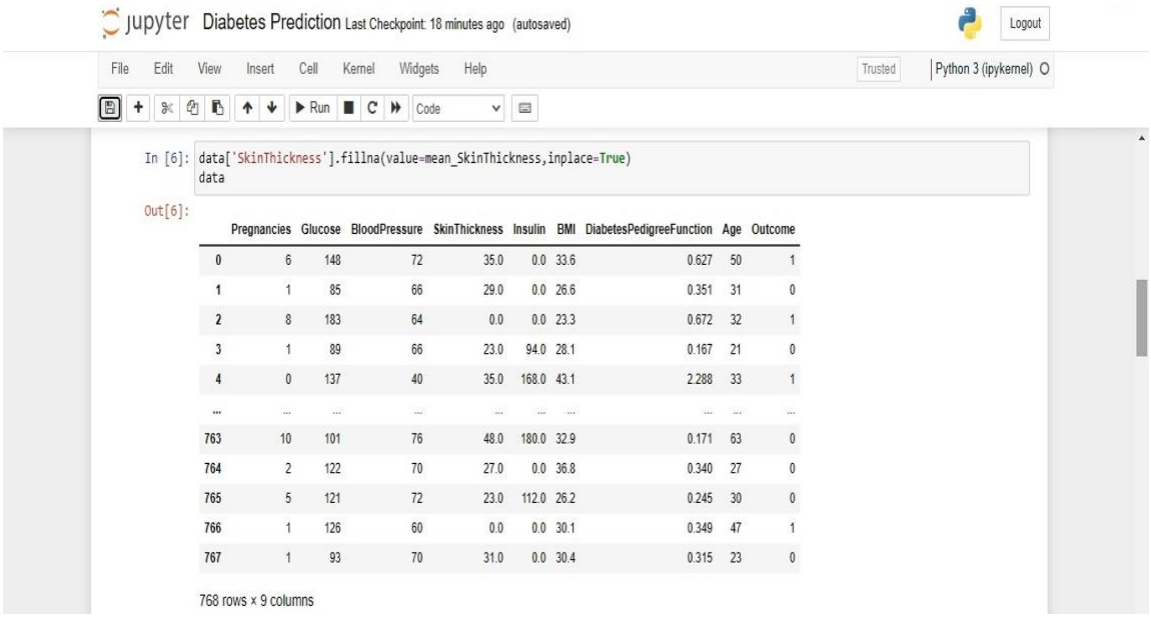


Fig 4.6: Example code for the pre-work of data set with null values

Step9:

Once we are good with the data with no null values and missing values and no string values then we can split the data into 2 parts.

Part1 : Training data

Part2: Testing data

Here our Target column is ‘Outcome’ i.e y so separating from the main data now we have y data frame with ‘Outcome’ and x data frame with all the other lables. We need at least 80% of data to train and 20% of data to test.

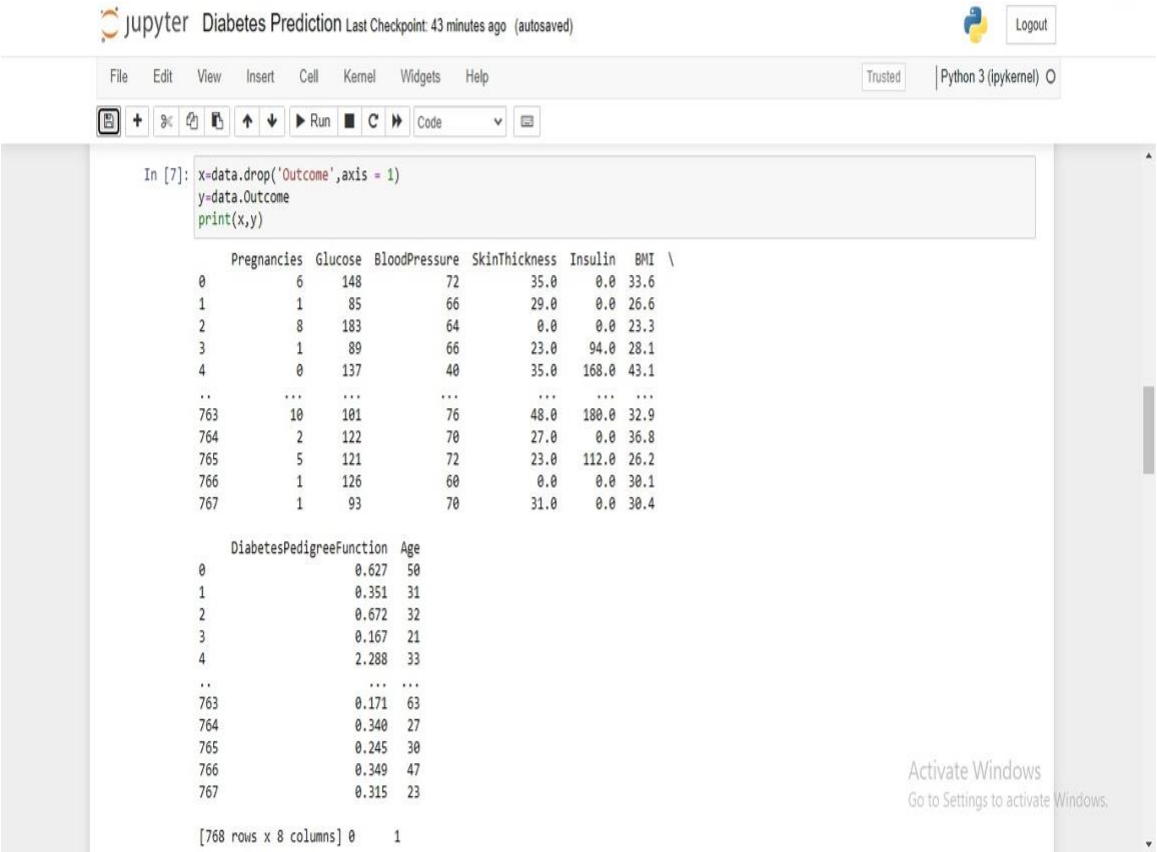


Fig 4.7 : code for splitting data frame into two parts based on the column values

Step10:

Now further splitting the data into

x_train,x_test

y_train,y_test

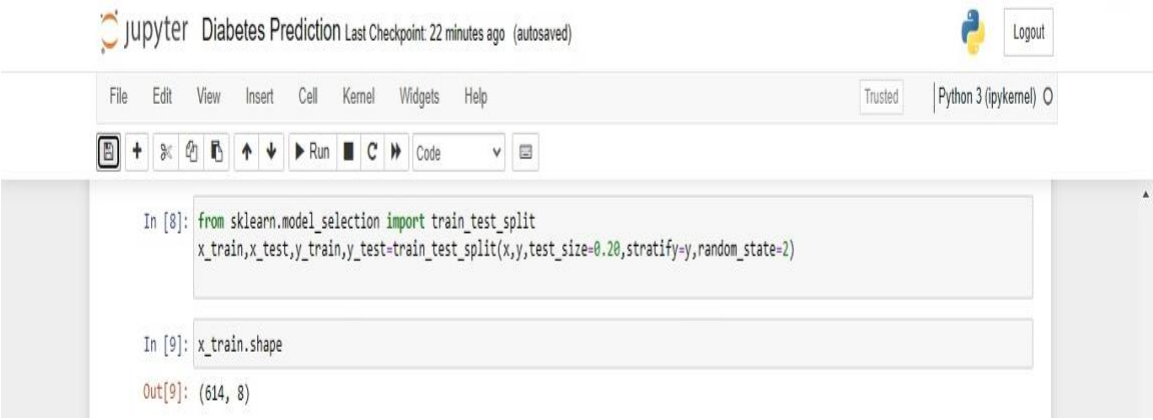


Fig 4.8: code for splitting data frame into two parts based on the column values

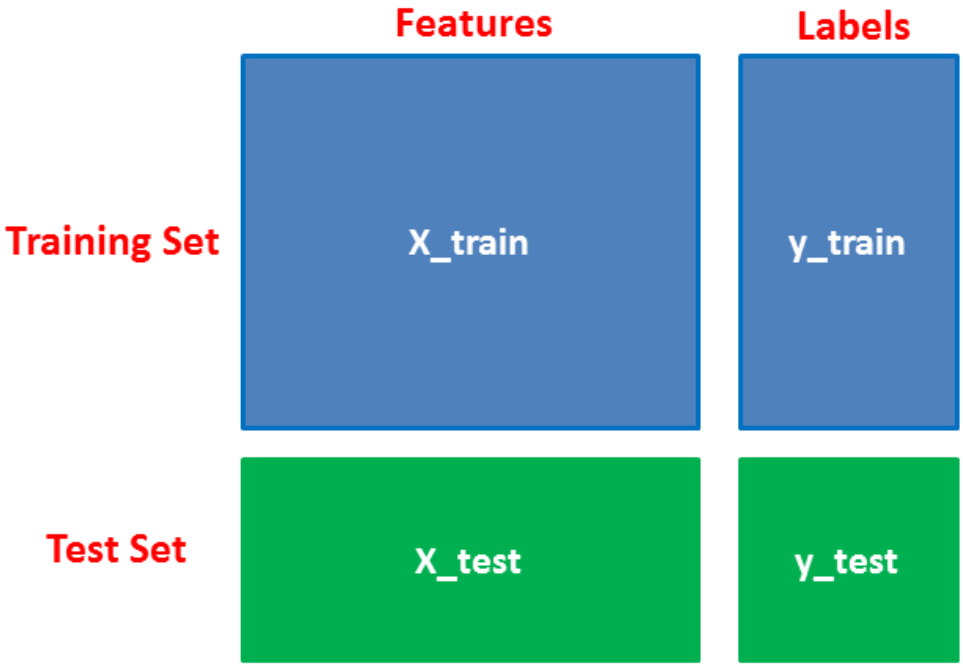


Fig 4.9: Splitting Datasets

Step11:

Now we will apply ML algorithm i.e. LogisticRegression from Sklearn.linear_model.

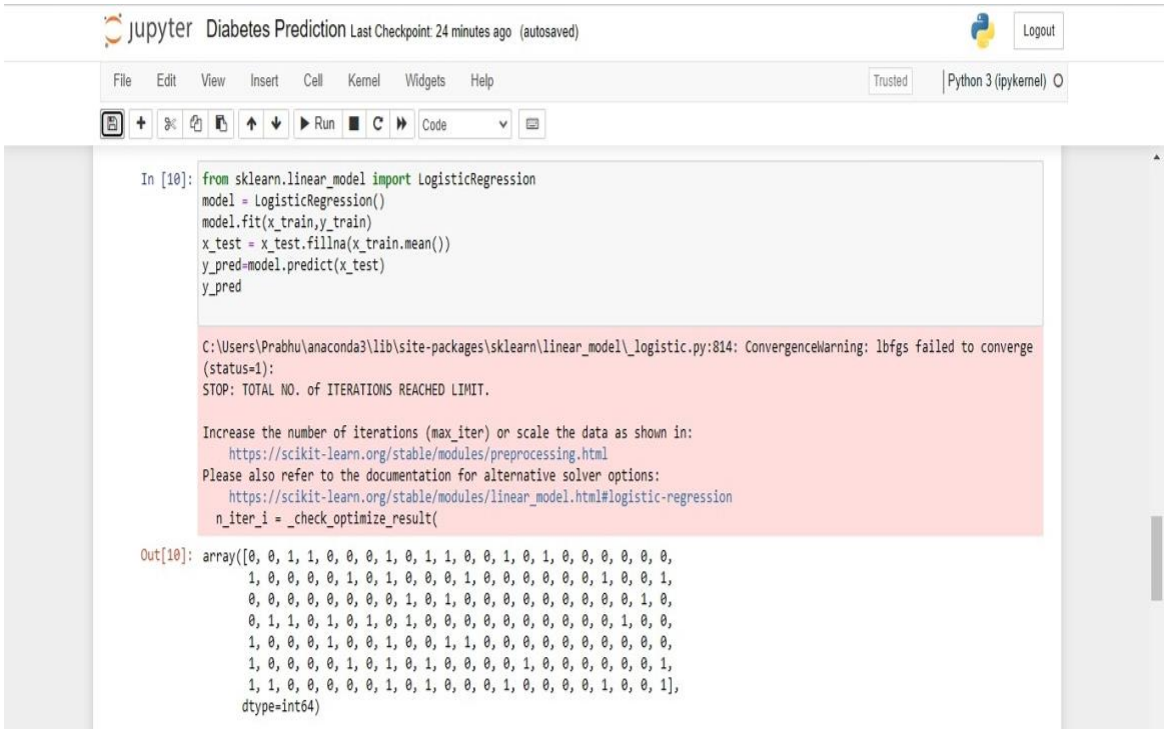


Fig 4.10: code to demonstrate fitting the algorithm to the model and resulting array of predictions made

Step12:

From the trained model we will calculate the accuracy of our model.

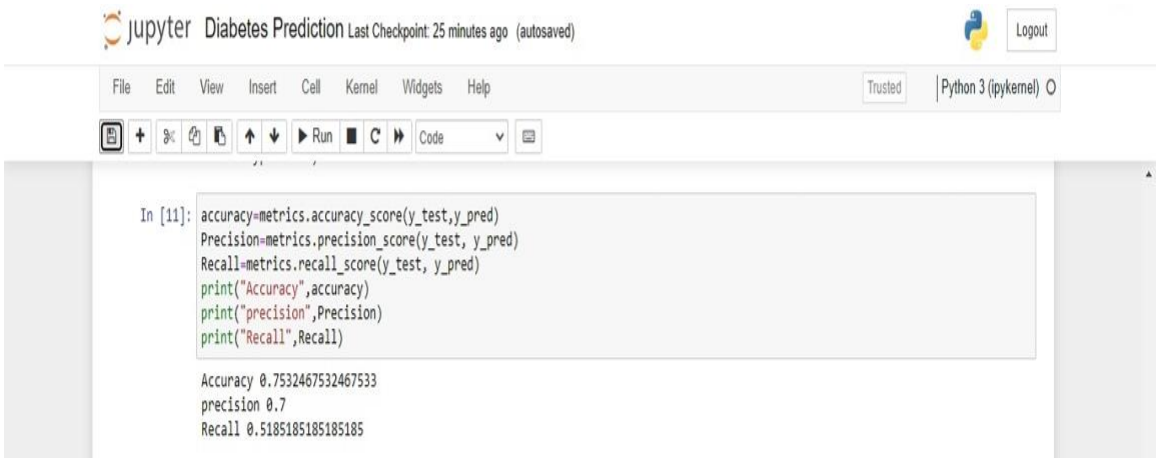


Fig 4.11: code to calculate the accuracy of the model

Step13:

Now will see the accuracy graph.

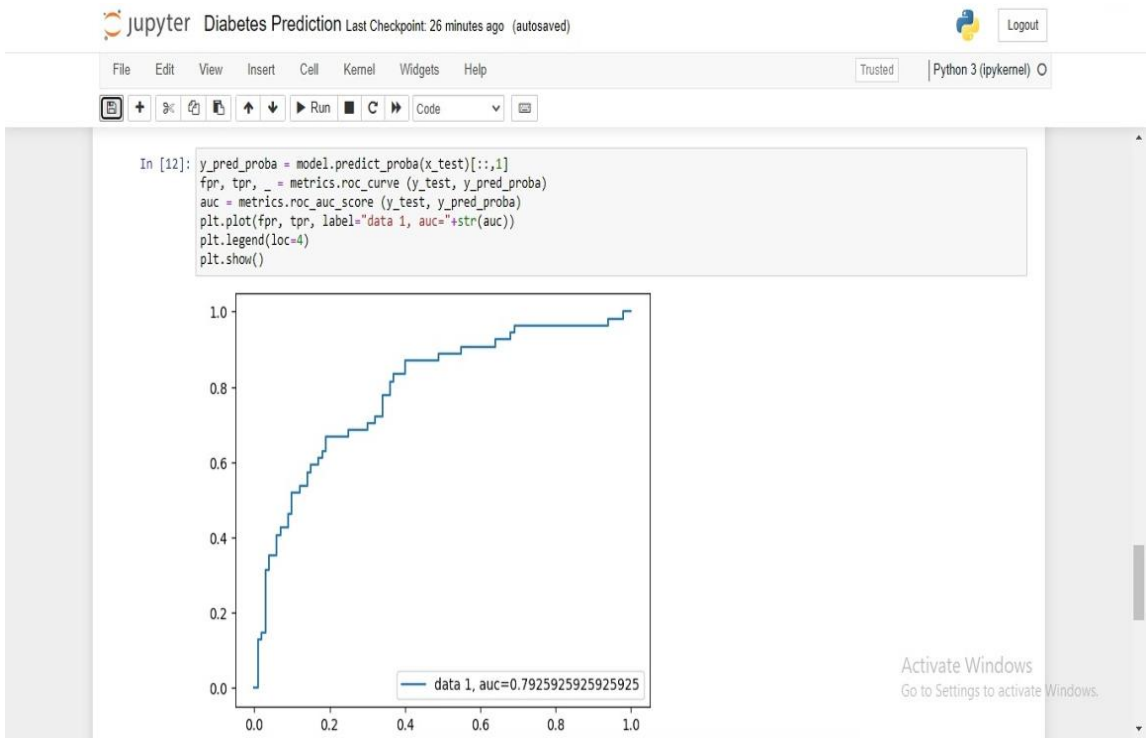


Fig 4.12: Accuracy graph model

Confusion Matrix

Fig 4.13: code for importing metrics to find the accuracy score

METHODOLOGY:

- Data Collection
- Data Preparation
- Choosing a model
- Training the model
- Evaluating the model
- Parameter turning
- Making prediction

- Data Collection : Collecting the Dataset from the Kaggle.com
- Data Preparation: We need to prepare data before training so preparation includes dealing with missing data filling the null values with the mean or median.
- Choosing a Model: model or algorithm selection is one of the most important step in predictions which is to select a particular ML Algorithm for training.

- Training the Model: After selecting the model we need to train the data.
- Evaluating the Model: After training we need to check weather the model is accurately predicting the outcomes(Accuracy Score=75%).
- Making Predictions: Finally we need to predict the diabetes of a particular row to check weather the model is working perfectly or not.

TECHOLOGIES USED:

- Data science is a multidisciplinary approach to extracting actionable insights from the large and ever-increasing volumes of data collected and created by today's organizations**.**
- Data science encompasses preparing data for analysis and processing, performing advanced data analysis, and presenting the results to reveal patterns and enable stakeholders to draw informed conclusions.
- Data preparation can involve cleansing, aggregating, and manipulating it to be ready for specific types of processing. Analysis requires the development and use of algorithms, analytics and AI models.
- It's driven by software that combs through data to find patterns within to transform these patterns into predictions that support business decision-making.
- The accuracy of these predictions must be validated through scientifically designed tests and experiments.
- The results should be shared through the skillful use of data visualization tools that make it possible for anyone to see the patterns and understand trends.

CHAPTER 5

RESULT, PERFORMANCE ANALYSIS

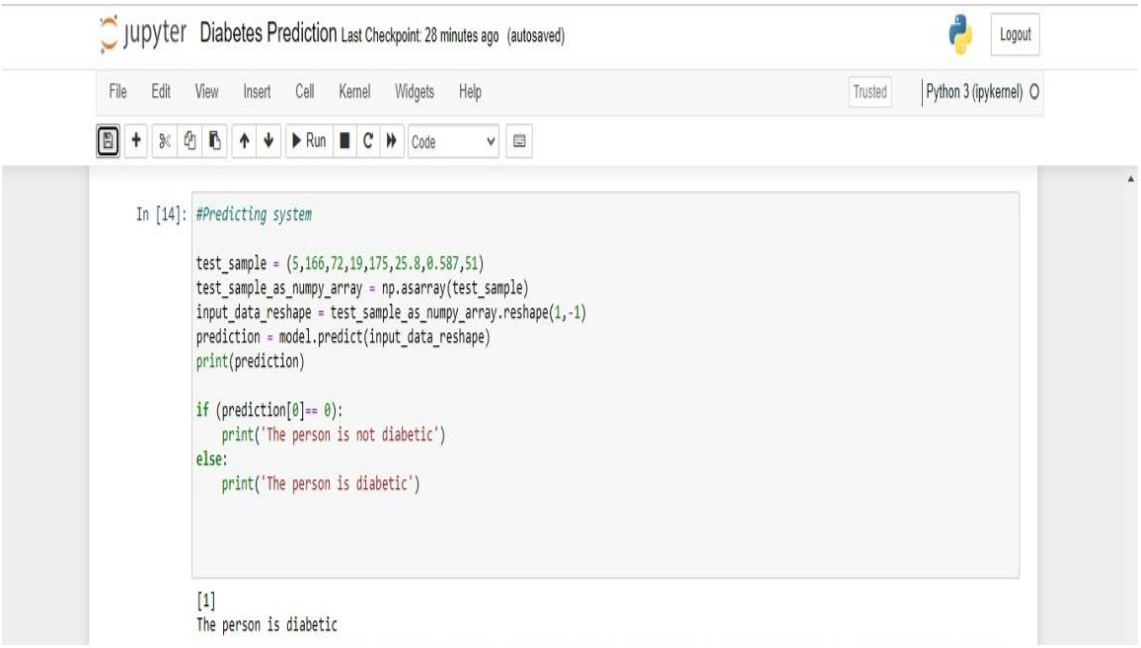


Fig 5.1: code for the prediction model

- From the above snippet we are successfully able to see the predictions made by the model.
- If the Output is ‘1’ that means the person is diabetic but if the Output is ‘0’ that means the person is not diabetic.
- Here the Output is based on considering the other labels like Glucose, Pregnancies , BloodPressure , SkinThickness , Insulin , BMI , DiabetesPedigreeFunction , Age.
- Successfully achieved an accuracy of 75%

```
accuracy=metrics.accuracy_score(y_test,y_pred)
Precision=metrics.precision_score(y_test, y_pred)
Recall=metrics.recall_score(y_test, y_pred)
print("Accuracy",accuracy )
print("precision",Precision )
print("Recall",Recall)
```

```
Accuracy 0.7532467532467533
precision 0.7
Recall 0.5185185185185185
```

Code to calculate the accuracy of the model

- Successfully achieved an accuracy of 75% using LogisticRegression and 69% accuracy using DecisionTreeClassifier.

CHAPTER 6
SUMMARY AND CONCLUSION

The main aim of this project was to design and implement Diabetes Prediction Using Machine Learning Methods and Performance Analysis of that methods and it has been achieved successfully. Successfully able to clean the data and split it into training and testing data. The proposed approach uses various classification and ensemble learning method in which Decision Tree, Logistic Regression are used. 75% classification accuracy has been achieved. The Experimental results can be asst health care to take early prediction and make early decision to cure diabetes and save humans life. In future I would like to move on to Deep Learning and upgrading myself with new technologies like TensorFlow and keras and NeuralNetwork and continue my research in the field of AI and built many more projects.

APPENDIX

A. Source Code

```
{#c1}
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.linear_model import LogisticRegression

{#c2}
data=pd.read_csv("diabetes.csv")
data

{#c3}
data.isna().sum()

{#c3}
mean_SkinThickness=data.SkinThickness.mean()

{#c4}
mean_Insulin=data.Insulin.mean()

{#c5}
data['SkinThickness'].fillna(value=mean_SkinThickness,inplace=True
)
data

{#c6}
x=data.drop('Outcome',axis = 1)
y=data.Outcome
print(x,y)

{#c7}
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,stratify
=y,random_state=2)

{#c8}
x_train.shape

{#c9}
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(x_train,y_train)
x_test = x_test.fillna(x_train.mean())
y_pred=model.predict(x_test)
y_pred

{#c10}
accuracy=metrics.accuracy_score(y_test,y_pred)
Precision=metrics.precision_score(y_test, y_pred)
Recall=metrics.recall_score(y_test, y_pred)
print("Accuracy",accuracy)
print("precision",Precision)
print("Recall",Recall)

{c11}
y_pred_proba = model.predict_proba(x_test)[::,1]
fpr, tpr, _ = metrics.roc_curve (y_test, y_pred_proba)
auc = metrics.roc_auc_score (y_test, y_pred_proba)
plt.plot(fpr, tpr, label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()

{#c12}
from sklearn import metrics

cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
cnf_matrix

{#c13}
#Predicting system
```

```
test_sample = (5,166,72,19,175,25.8,0.587,51)
test_sample_as_numpy_array = np.asarray(test_sample)
input_data_reshape = test_sample_as_numpy_array.reshape(1,-1)
prediction = model.predict(input_data_reshape)
print(prediction)

if (prediction[0]== 0):
      print('The person is not diabetic')
else:
      print('The person is diabetic')
```

B. Screen Shots

Fig 6.1: code of importing libraries

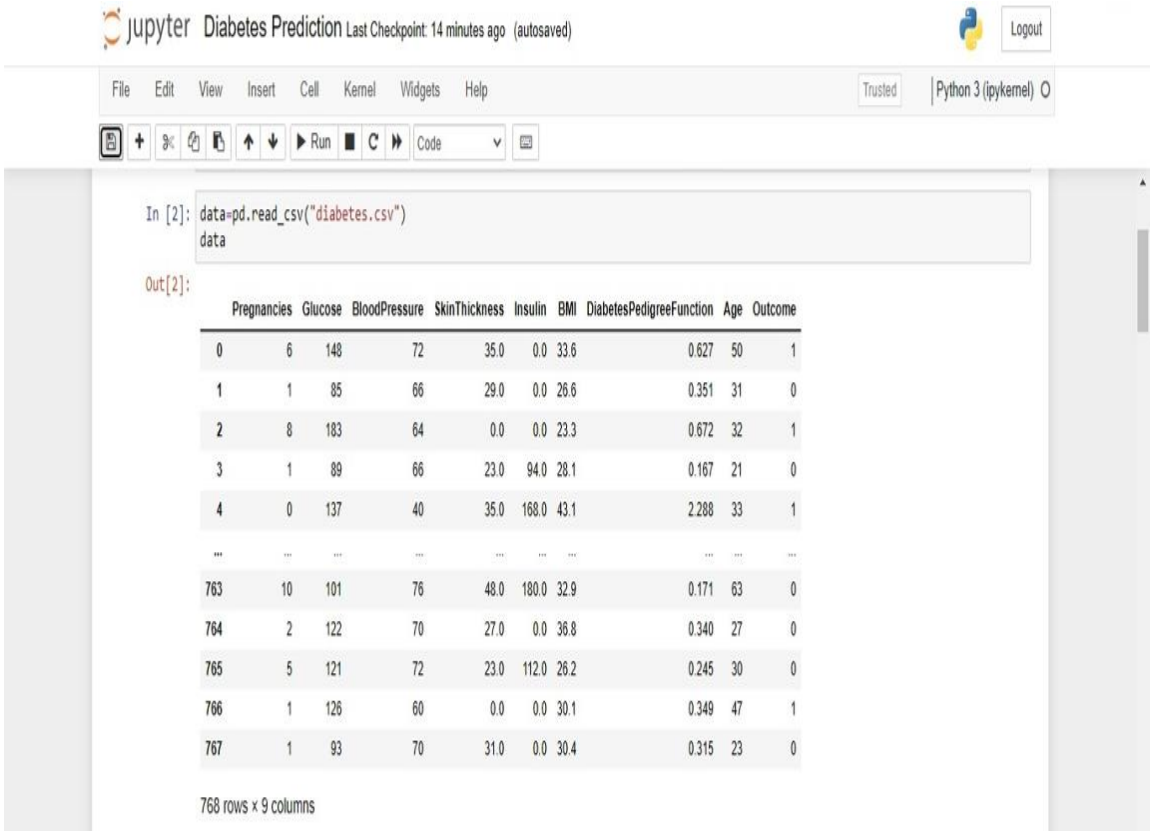


Fig 6.2: code for importing data set

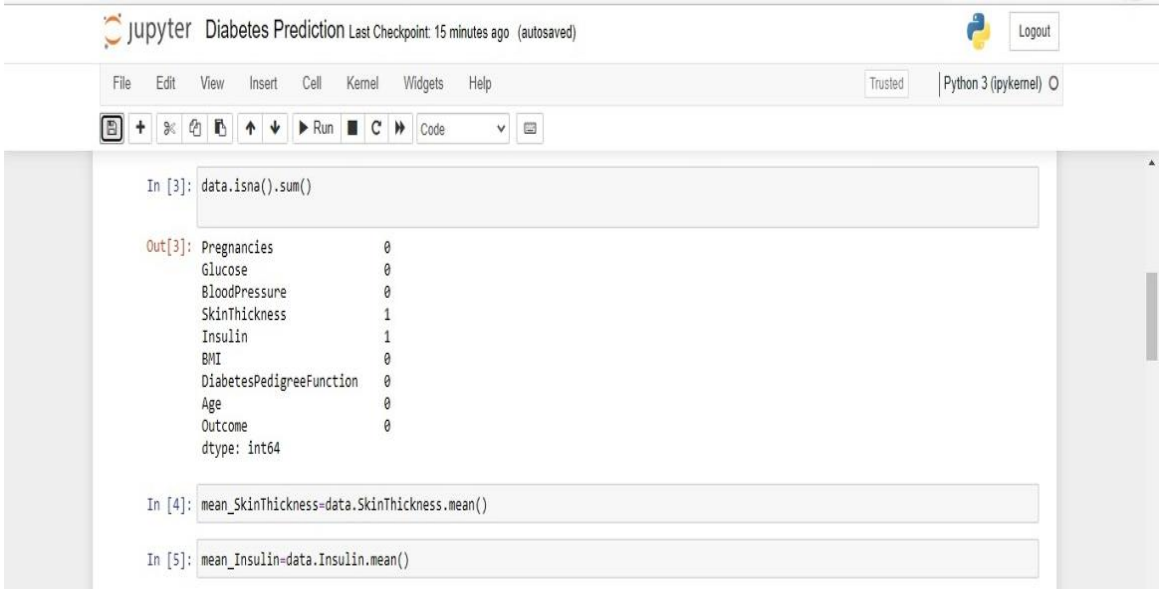


Fig 6.3: Example code for the pre-work of data set with null values

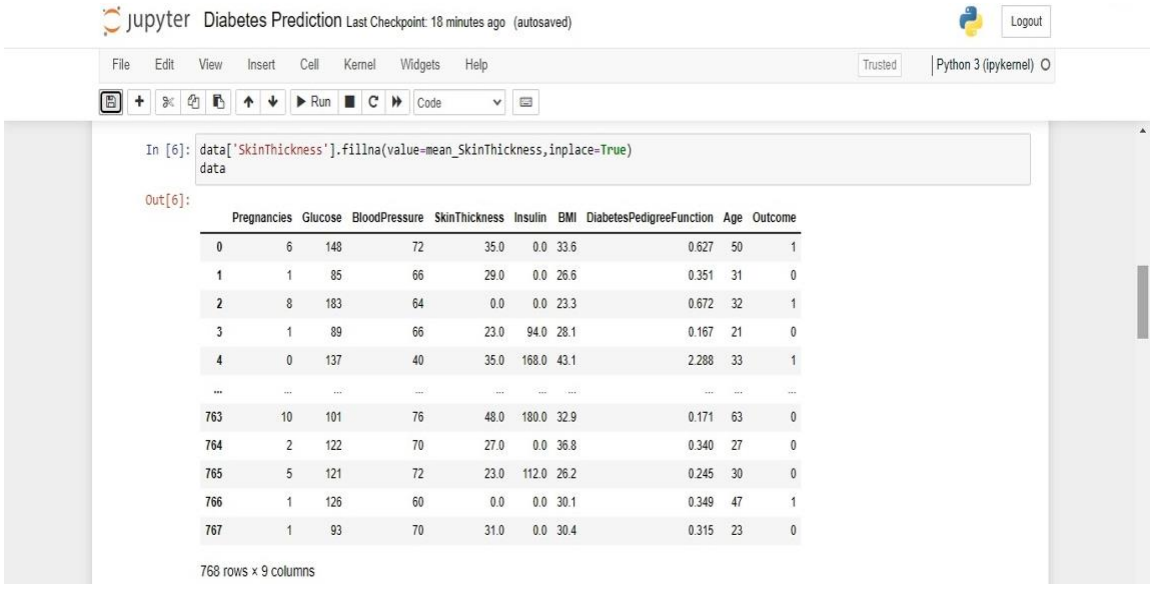


Fig 6.4: Example code for the pre-work of data set with null values

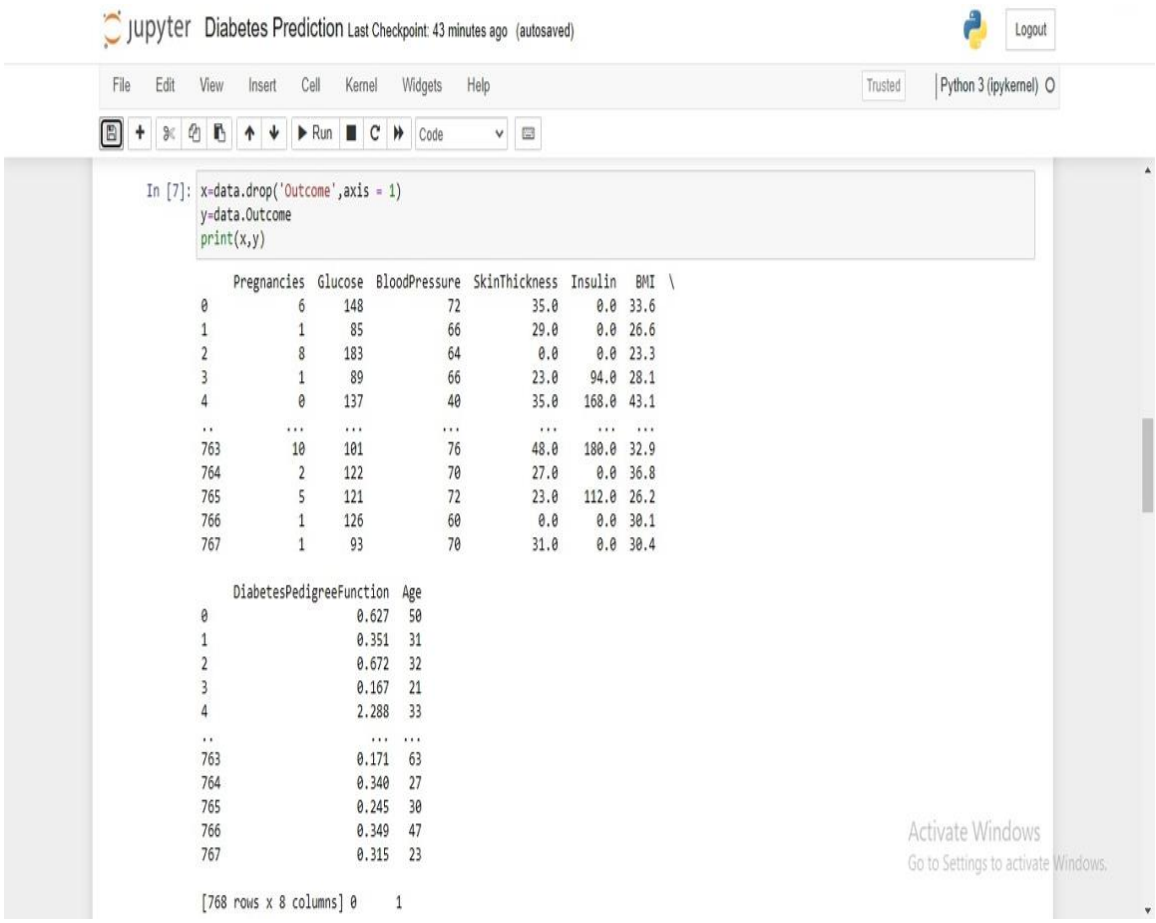


Fig 6.5: code for splitting data frame into two parts based on the column values

Fig 6.6: code for splitting data frame into two parts based on the column values

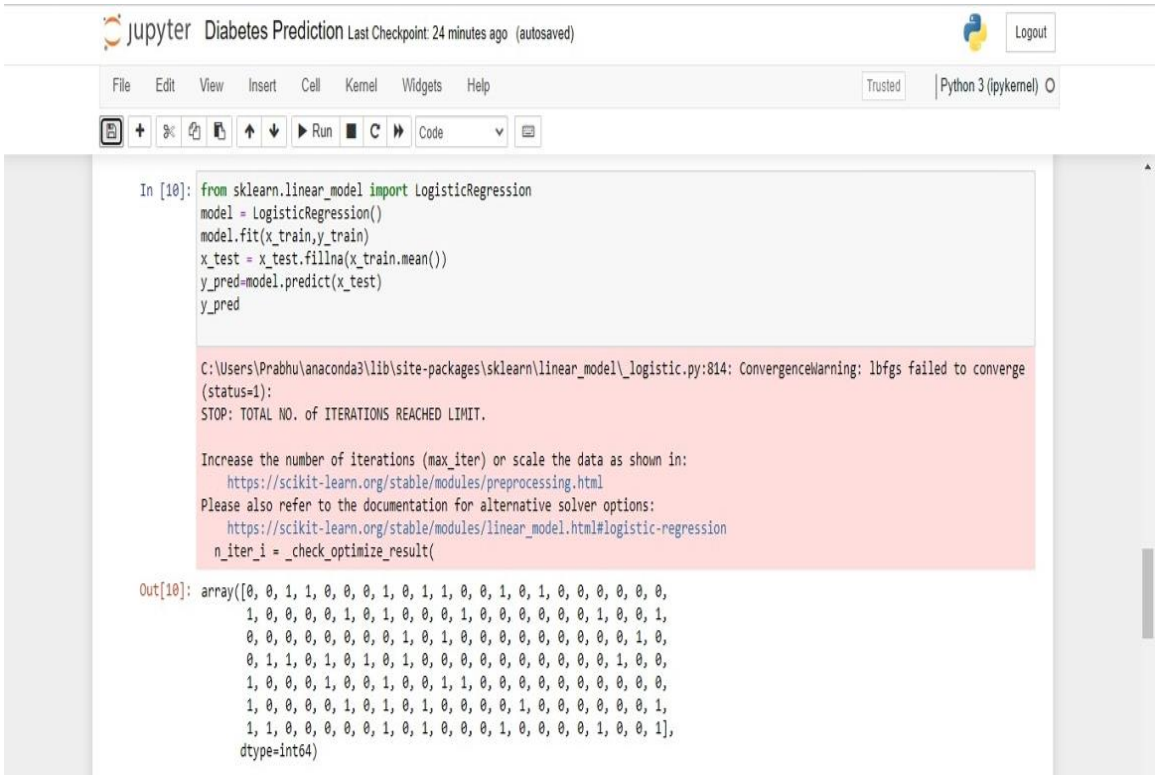


Fig 6.7: code to demonstrate fitting the algorithm to the model and resulting array of predictions made

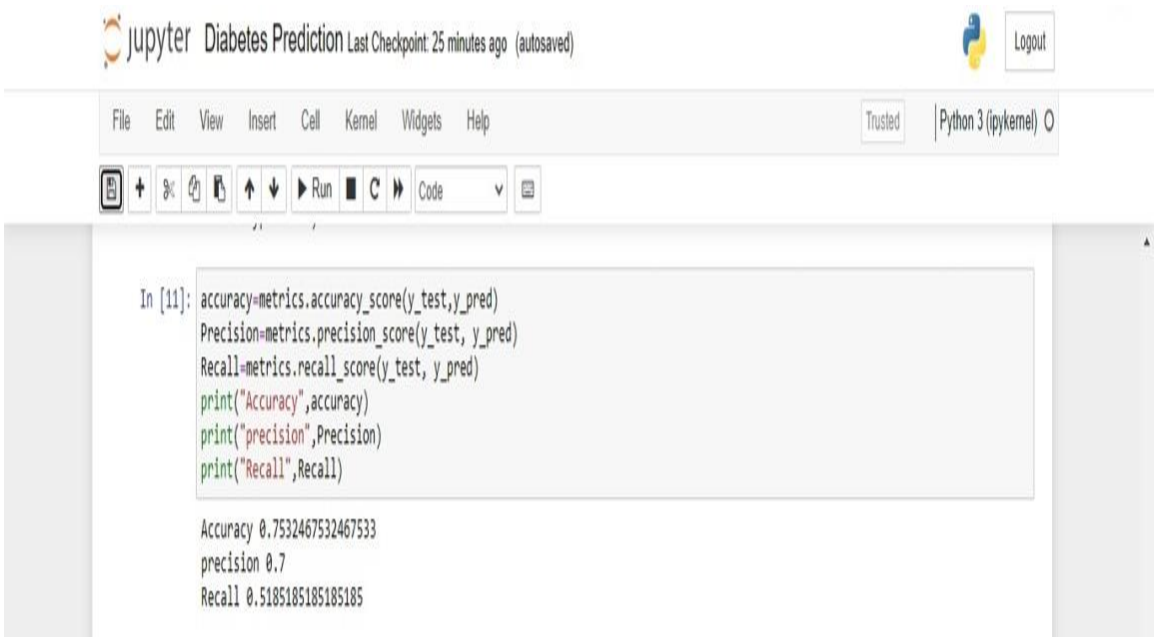


Fig 6.8: code to calculate the accuracy of the model

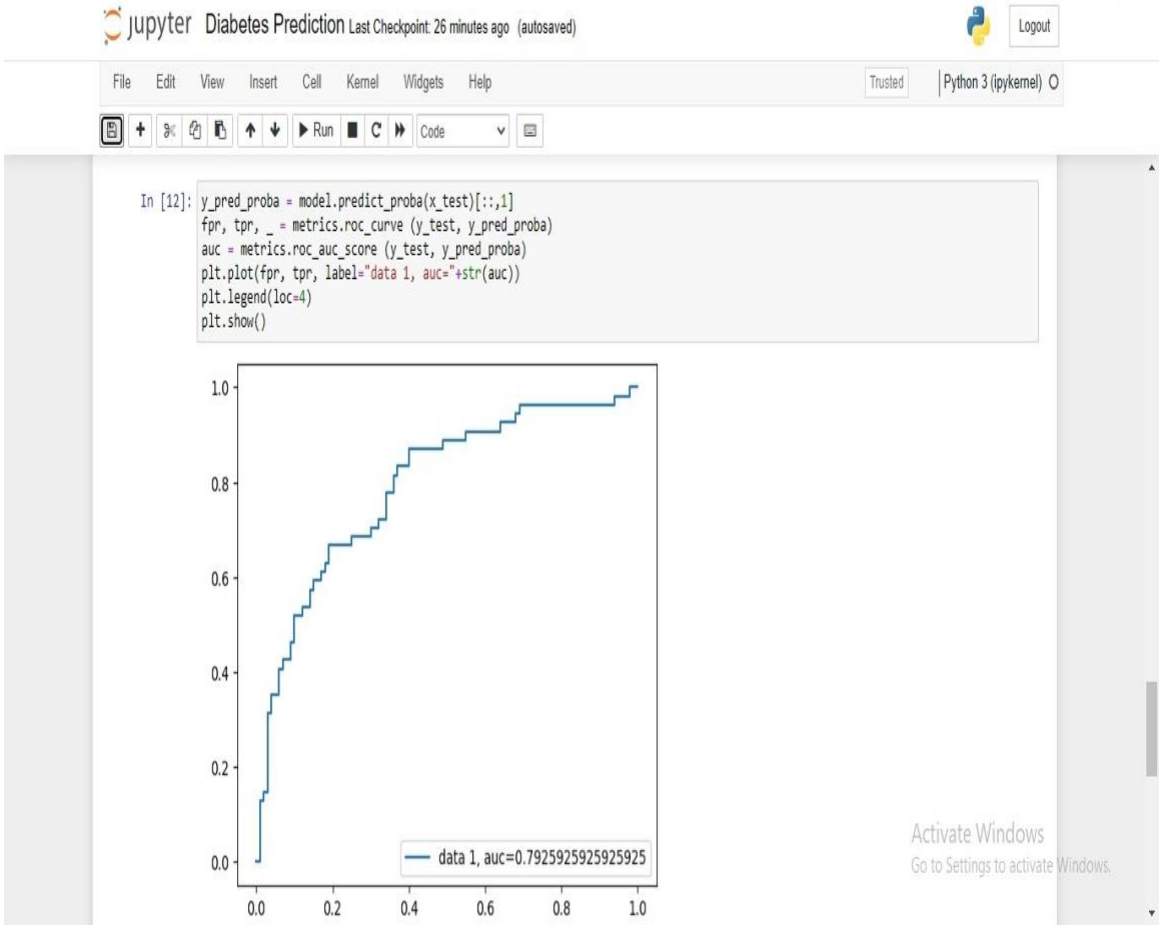


Fig 6.9: Accuracy graph model

Fig 6.10: code for importing metrics to find the accuracy score

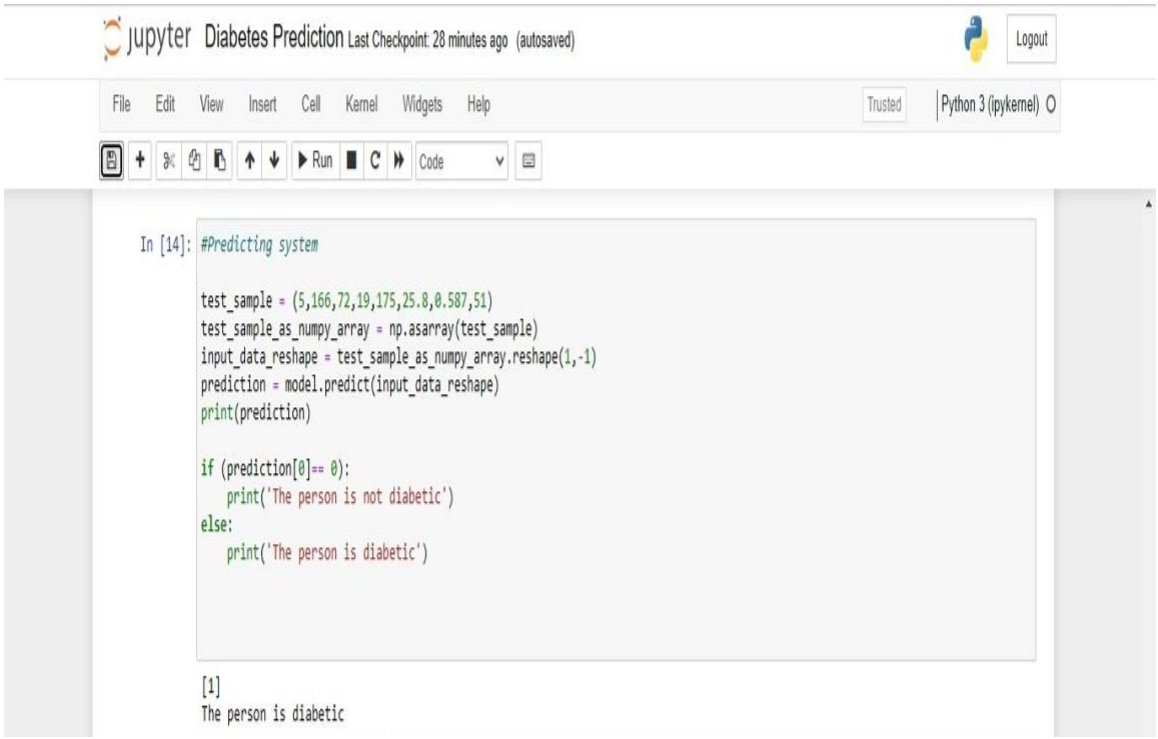


Fig 6.11: code for the prediction model

REFERENCES

1. DeeptiSisodia, Dilip Singh Sisodia,"Prediction of Diabetes Using Classification Algorithm", www.elsevier.com/locate/procedia, Procedia computer science 132(2018) 1578-1585.

2. Xue-Hui Meng,Yi-Xiang Huang,Dong-PingRao,Qiug Liu,2013,"Comparison of Three Data Mining Models For Predicting Diabetes of Prediabetes By Rick Factos",Kaohsiung journal of medical science(2013) 29,93-99.

3. V.AnujaKumari, R.Chithra."Classification of Diabetes Disease Using Support Vector Machine".vol 3.,Issue 2,March-April 2013,pp.1797-1801.www.ijera.com.

4. Monisha.A, S.ShalinChistina, Nirmala Santiago, "Decision support system for a chronic disease-Diabetes". International Journal of Computer &Mathematical Science(IJCMS),ISSN 2347-8527 Volume 7,Issue 3,March 2018.

5. S.Selvakumar, K.Senthamarai Kannan and S.GothaiNachiyar, "Prediction of Diabetes Diagnosis Using Classification Based Data Mining Techniques", International Journal of statistics and Systems,ISSN 0973-2675 Volume 12,Number 2(2017),PP.183-188.http://www.ripublication.com.

6. Aiswarya Iyar, S. Jeyalatha and RonakSumbaly, "Diagnosis of Diabetes Using Classification Mining Techniques", International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.5, No.1, January 2015.

7. B.Tamilvanan, Dr.V.MuraliBhaskaran, "An Experimental Study of Diabetes Disease Prediction System Using Classification Techniques", IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-

0661,p-ISSN: 2278-8727, Volume 19, Issue 1, Ver. IV (Jan.-Feb. 2017), PP 39-44, www.iosrjournals.org.

8. Rahul Joshi, MinyechilAlehegn,“Analysis and prediction of diabetes diseases using machine learning algorithm: Ensemble approach”,International Research Journal of Engineering and Technology (IRJET),Volume: 04 Issue: 10 | Oct -2017,e-ISSN: 2395-0056,p-ISSN: 2395-0072. www.irjet.net.

9. Amina Azar,Yasir Ali, Muhammad Awais, KhurramZaheer,“Data Mining Models Comparison for Diabetes Prediction”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 8, 2018.

10. VeenaVijayan.V, Anjali.C,”Decision Support Systems for Predicting Diabetes Mellitus –A Review”, Proceedings of 2015 Global Conference on Communication Technologies(GCCT 2015), 978-1-4799-8553-1/15/$31.00 © s2015 IEEE.

11. DeepikaVerma , Dr.Nidhi Mishra, “Analysis and prediction of breast cancer and diabetes disease dataset using data mining classification techniques”,Proceedings of the International Conference on Intelligent Sustainable Systems (ICISS 2017),IEEE Xplore Compliant - Part Number:CFP17M19-ART, ISBN:978-1-5386-1959-9.

12. VeenaVijayan V, Anjali c, “Prediction of diagnosis of diabetes mellitus –A machine learning approach ”, 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS) | 10-12 December 2015.

13. Ayman Mir, SudhirN.Dhage, “Diabetes Disease Prediction using MachineLearning on Big Data of Healthcare”, 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA).

14. S M Hasan Mahmud, MdAltabHossin, Md. Razu Ahmed,

SheakRashedHaiderNoori, MdNazirul Islam Sarkar, “Machine Learning Based Unified Framework for DiabetesPrediction”, BDET 2018, August 25–27, 2018, Chengdu, China. © 2018 Association for Computing Machinery. ACM ISBN 978-1-4503-6582-6/18/08…$15.00.DOI:https://doi.org/10.1145/3297730.3297737.

15. AakanshaRathore, Simran Chauhan, SakshiGujral, “Detecting and Predicting Diabetes Using Supervised Learning: An Approach towards Better Healthcare for Women”,Volume 8, No. 5, May-June 2017, ISSN No. 0976-5697, Available Online at www.ijarcs.info.