

PROFESSIONAL TRAINING REPORT

at

**Sathyabama Institute of Science and Technology
(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in

Computer Science and Engineering

By

SAURAV KUMAR

(Reg. No. 38110513)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMILNADU.**

MARCH 2022



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

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **SAURAV KUMAR(Reg. No. 38110513)** who has done the project in “**MACHINE LEARNING**” and carried out the project entitled “**Diabetes check up using different machine learning algorithm**” under our supervision from JANUARY 2022 to MARCH 2022.

Internal Guide

DR.R.M GOMATHI

Head of the Department

DR. S.VIGNESHWARI, M.E.,(Ph.D)

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **SAURAV KUMAR(Reg. No. 38110513)** hereby declare that the Professional Training Report on “**Diabetes check up using different machine learning algorithm**” done by me under the guidance of **DR.R.M GOMATHI**, at Sathyabama university is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science And Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing menecessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **DR.R.M GOMATHI** for her valuable guidance, suggestions which paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the department of **COMPUTER SCIENCE AND ENGINEERING** who are helpful in many ways for the completion of the project.

ABSTRACT

The remarkable advancements in biotechnology and public healthcare infrastructures have led to a momentous production of critical and sensitive healthcare data. By applying intelligent data analysis techniques, many interesting patterns are identified for the early and onset detection and prevention of several fatal diseases. Diabetes mellitus is an extremely life-threatening disease because it contributes to other lethal diseases, i.e., heart, kidney, and nerve damage. In this paper, a machine learning based approach has been proposed for the classification, early-stage identification, and prediction of diabetes. Furthermore, it also presents prediction based on different classification algorithm. For diabetes classification, six different classifiers have been employed, i.e., random forest (RF), Naïve bayes classifier (NB), logistic regression (LR), SVM classifier, Decision tree, and KNN. For experimental evaluation, a benchmark PIMA Indian Diabetes dataset is used. Here 80% data is used for training and remaining 20% data is used testing. During the analysis, approach in many public healthcare applications.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	9
1.1	<i>HISTORY</i>	10
1.1.1	PRINCIPLES	11
1.1.2	VERSIONS	12
1.2	<i>PY2 VS PY3</i>	13
1.2.1	PYTHON FEATURES	14
1.3	<i>EXECUTION SYSTEM</i>	14
1.3.1	KAGGLE	14
1.3.2	ABOUT THE PROJECT	15
1.3.3	THE DATA	16
1.3.3.1	DATA CLEANING	17
1.3.3.2	DATA PREPROCESSING	17
1.3.3.3	ML MODELS	17

1.3.2	PYTHON INTERPRETER	17
	1.4 PYTHON 3 INSTALLATION	
	SET UP GUIDE FOR WINDOW	18
1.4.1	SETTING UP PATH	20
21	1.5 LOGISTIC REGRESSION	
1.5.1	PYTHON LIBRARY	21
2.	LITERATURE SURVEY	22
3.	AIM AND SCOPE OF THE PRESENT INVESTIGATION	23
	3.1 OBJECTIVE	23
	3.1.1 INTRODUCTION	24
	3.2 WHY ML	24
25	3.2.1 K-MEANS ALGORITHM	
	3.2.3 RANDOM STATE REGRESSOR	26
27	3.2.4 TERMS AND SCOPE	
4.	EXPERIMENTAL SR MATERIALS AND METHODS, ALGORITHMS USED	28
	4.1 HARDWARE REQUIREMENTS	28
	4.2 SOFTWARE REQUIREMENTS	28
	4.2.1 OVERVIEW OF SOFTWARE REQUIREMENTS	28

	4.2.1.1 PYTHON	28
	4.2.1.2 METHODS	29
	4.2.1.3 IDLE PYTHON	29
5.	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	30
	5.1 ANALYSIS	30
	5.2 GOOGLE COLAB	31
	5.2.1 LANGUAGE CODE	33
	5.3 RESULT AND DISCUSSION	33
6.	SUMMARY AND CONCLUSIONS	34
	BIBLIOGRAPHY	35
	REFERENCES	35
	APPENDIX	36
	A. SOURCE CODE	36

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
1.1	PYTHON SYMBOL	10
1.2	STARTING PYTHON INSTALLATION	19
1.3	FINISH PYTHON INSTALLATION	20
5.1	GOOGLE COLAB	29
5.2	OPENING WINDOW	31
5.3	INPUT WINDOW	32

5.4	WINNING WINDOW	32
5.5	LOSING WINDOW	32
5.2.1	LANGUAGE CODE	33

CHAPTER 1

INTRODUCTION

Python is a general-purpose programming language that is class-based, object-oriented (although not a pure object-oriented language, as it contains primitive types), and designed to have as few implementation dependencies as possible. It is intended to let application developers *write once, run anywhere* (WORA), meaning that compiled Python code can run on all platforms that support Python without the need for recompilation. Python applications are typically compiled to by IDLE python that can run on any platform regardless of the underlying computer architecture.

This disease is known to almost everyone in today's era. WHO has told that in india by 2020 there will be nearly 32 million diabetic patients in India and by 2030 it will go up to 80 Million . A step can be taken by studying the data on a large scale through Machine learning Algorithm. The algorithm of this machine learning is used to analyze the data and to make predictions on the new data. Scholars have tried to prevent many diseases by using the algorithm of this machine. As a result, the prediction of this machine is very accurate. These days the machine learning algorithms are being used extensively in the medical field and not only this, it has made it easy to detect any disease. Doctors and nurses are also using it, which makes it easier for them to test but high quality technology is not being used in our country right now so Machine learning algorithm's techniques can bring the revolution in medical field.As of now 82% medical equipment is based on Machine.

The latest, and unreleased, documentation for versions of Python still under development:

<http://docs.python.org/dev/>

<http://docs.python.org/3/>

Python is a programming language. Python can be used on a server to create your own applications also.

1.1 HISTORY

Python is an interpreted, high-level, general-purpose programming language. Python has a design philosophy that emphasizes code readability, notably

using significant whitespace. It provides constructs that enable clear

programming on both small and large scales.

Python is a scripting language like PHP, Perl, and so much more. It can be

used for programming (django, Zope, Google App Engine, and much

more). But it also can be used for desktop applications (Blender 3D, or even

for games). **Python** can also be translated into binary code like java.

In February 1991, Van Rossum published the code (labeled version 0.9.0) to

alt.sources. Already present at this stage in development were [classes with inheritance](#), exception handling, functions, and the core datatypes

of `list`, `dict`, `str` and so on. Also in this initial release was a [module](#)

[system](#) borrowed from [Modula-3](#); Van Rossum describes the module as "one of Python's major programming units". Python's exception model also resembles

Modula-3's, with the addition of an `else` clause. In 1994 [comp.lang.python](#), the

primary discussion forum for Python, was formed, marking a milestone in the growth of Python's userbase..

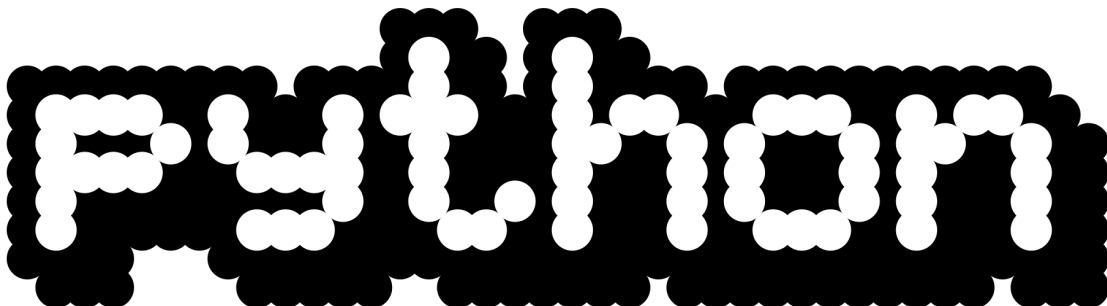


Fig 1.1 Python

Python has a simple syntax similar to the English language. **Python** has syntax that

allows developers to write programs with few lines than some other programming languages. **Python** runs on an interpreter system, meaning that code

can be executed as soon as it is written. It's easy for user to implement.

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. Python is a high level object-oriented, programming language. It has built-in data structures, combined with dynamic typing & binding which makes it an ideal choice for rapid application development. Python also offers support for modules and packages, which allows system modularity and code reuse.

It is one of the fastest programming languages as it requires very few lines of code. Its emphasis is on readability and simplicity, which make it a great choice for beginners.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Python 2.0 was released on October 16, 2000, with many major new features, including a cycle-detecting garbage collector (in addition to reference counting) for memory management and support for Unicode. However, the most important change was to the development process itself, with a shift to a more transparent and community-backed process.

Python 3.0, a major, backwards-incompatible release, was released on December 3, 2008 after a long period of testing. Many of its major features have also been backported to the backwards-compatible, while by now unsupported, Python 2.6 and 2.7

1.1.1 PRINCIPLES

There are so many primary goals in the creation of the Python language and some important ones are:

1. It must be Beautiful is better than ugly.
2. It must be Explicit is better than implicit.

3. It must be Simple is better than complex.
4. It must be Flat is better than nested.
5. It must be Sparse is better than dense.
6. It must be Readability counts.

1.1.2 VERSIONS

As of March 2020, Python3.8 is supported; and both Python 3.8 and 3.7 as Long Term Support (LTS) versions. Major release versions of Java, along with their release dates:

- Versions.
- Python 3.8.5, documentation released on 20 July 2020. •
- Python 3.8.4, documentation released on 13 July 2020. •
- Python 3.8.3, documentation released on 13 May 2020. •
- Python 3.8.2, documentation released on 24 February 2020. •
- Python 3.7.8, documentation released on 27 June 2020. •
- Python 3.7.7, documentation released on 10 March 2020. •
- Python 3.7.6, documentation released on 18 December 2019. •
- Python 3.7.5, documentation released on 15 October 2019. •
- Python 3.7.4, documentation released on 08 July 2019. •
- Python 3.7.3, documentation released on 25 March 2019. •
- Python 3.7.2, documentation released on 24 December 2018. •
- Python 3.7.1, documentation released on 20 October 2. •
- Python 2.7.18, documentation released on 20 April 2020 •
- Python 2.7.17, documentation released on 19 October 2019 •
- Python 2.7.16, documentation released on 02 March 2019 •
- Python 2.7.15, documentation released on 30 April 2018 •
- Python 2.7.14, documentation released on 16 September 2017 •

- Python 2.7.13, documentation released on 17

1.2 PY2 VS PY3

Let's begin with a brief timeline of Python 2 vs 3 usage.

- Python 2.0 was first released in 2000. Its latest version, 2.7, was released in 2010.
- Python 3.0 was released in 2008. Its neyoust version, 3.6, was released in 2016, and [version 3.7](#) is currently in development.
- Although Python 2.7 is still widely used, Python 3 adoption is growing
- quickly. In 2016, 71.9% of projects used Python 2.7, but [by 2017, it had fallen to 63.7%](#).
- This signals that the programming community is turning to Python
- 3—albeit gradually—when developing real-world applications.
- Notably, on January 1, 2018, Python 2.7 will “retire” and no longer be mentioned.

1.2.1 PYTHON FEATURES

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter.

These modules enable programmers to add to or customize their tools to be

more efficient.

- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

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

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

1.3 EXECUTION SYSTEM

1.3.1 KAGGLE

Inside Kaggle you'll find all the code & data you need to do your data science work. Use over 50,000 public [datasets](#) and 400,000 public **notebooks** to conquer any analysis in no time.

1.3.2 ABOUT THE PROJECT

Propose work:-Finding Highest accuracy rate using Different Algorithm and Implement in real life.

Diabetes Prediction Model

Dataset Collection

Data Pre-processing

Classification

Build Model

Evaluation

Outcome

Dataset Collection Through this data, it is easy to understand the collection of data and the input of data Through it's pattern we can predict it. A dataset is a collection of records which is usually presented in tabular form. This Diabetes dataset contains 768 rows and 9 columns in which attributes are given below.

TABLE

Dataset Information Column Type

Number of Pregnancies

N

Glucose value N

Blood Pressure N

Skin Thickness(mm) N

Insulin N

BMI N

AGE N

First Point is Data collection After importing important libraries we upload the dataset and study all the point and If any correction needed we remove that point in Data Pre-processing.

Data Pre-processing Data Preprocessing is a stage of process changing the unorganized data into perfect form so we should make sure data is clean. we feed up this data in a machine learning algorithms which will have the great

impact of the accuracy and efficiency of a machine learning module. aThis Data Preprocessing involves many steps like (i) Finding Missing values (ii) Feature Scaling (iii) Encoding Categorical Data etc.when we process the data ,we get the information.We organize the data in proper way.We group the data,sort the data then we send it to the machine.further we process the data on algorithm which is implemented by Data scientist.

Classification A classification is Supervised learning algorithm which is basically categories a bundle of data into classes. There are bunch of Classification algorithms in machine learning uring this part, we have implemented different classification algorithm .

Logistic Regression: This algorithm is used for one or more independent variables to determine an output.The output is based on a two variables meaning it can contain only two possible output.

1. Data Cleaning (Identifying null values, filling missing values and removing outliers)
2. Data Preprocessing (Standardization or Normalization)
3. ML Models: Linear Regression, Ridge Regression, Lasso, KNN, Random Forest Regressor, Bagging Regressor, Adaboost Regressor, and XGBoost
4. Comparison of the performance of the models
5. Some insights from data

1. THE DATA:

The dataset used in this project was downloaded from [Kaggle](#).

2. DATA CLEANING :

The first step is to remove irrelevant/useless features like 'URL', 'region_url', 'vin', 'image_url', 'description', 'county', 'state' from the dataset.

3. DATA PREPROCESSING:

Label Encoder: In our dataset, 12 features are categorical variables and 4 numerical variables (price column excluded). To apply the ML models, we need to transform these categorical variables into numerical variables and sklearn library is used to solve this problem.

4. ML MODELS:

In this section, different machine learning algorithms are used to predict price/target-variable.

1.3.3 PYTHON INTERPRETER

Python is an excellent programming language that allows you to be productive in A wide variety of fields..

Python is also a piece of software called an **interpreter**. The interpreter is the program you'll need to run Python code and scripts. Technically, the interpreter is a layer of software that works between your program and your computer hardware

to get your code running.

Depending on the Python implementation you use, the interpreter can be:

- A program written in C, like CPython, which is the core implementation of the language.
- A program written in Java, like JPython .
- A program written in Python itself, like PyPy.
- A program implemented in .NET, like Iron,Python .

Whatever form the interpreter takes, the code you write will always be run by this program. Therefore, the first condition to be able to run Python scripts is to have the interpreter correctly installed in your system.

The interpreter is able to run Python code in two different ways:

- As a script or module
- As a piece of code typed into an interactive session

1.4 PYTHON 3 INSTALLATION AND SETUP GUIDE FOR WINDOWS

It is highly unlikely that your Windows system shipped with Python already installed. Windows systems typically do not. Fortunately, installing does not involve much more than downloading the Python installer from the python.org website and running it. Let's take a look at how to install Python 3 on Windows:

Step 1: Download the Python 3 Installer

1. Open a browser window and navigate to the [Download page for Windows](https://python.org) at python.org.
2. Underneath the heading at the top that says **Python Releases for Windows**, click on the link for the **Latest Python 3 Release - Python 3.x.x**. (As of this writing, the latest is Python 3.6.5.)
- 2 Scroll to the bottom and select either **Windows x86-64 executable installer** for 64-bit or **Windows x86 executable installer** for 32-bit. (See below.)

Sidebar: 32-bit or 64-bit Python?

For Windows, you can choose either the 32-bit or 64-bit installer. Here's what the difference between the two comes down to:

- If your system has a 32-bit processor, then you should choose the 32-bit installer.
- On a 64-bit system, either installer will actually work for most purposes. The 32-bit version will generally use less memory, but the 64-bit version performs better for applications with intensive computation.
- If you're unsure which version to pick, go with the 64-bit version.

Note: Remember that if you get this choice “wrong” and would like to switch to another version of Python, you can just uninstall Python and then re-install it by downloading another installer from python.org.

Step 2:

Run the Installer Once you have chosen and downloaded an installer, simply run it by double-clicking on the downloaded file.

A dialog should appear that looks something like this:



Fig 1.2 Python installation

Important:

You want to be sure to check the box that says Add Python 3.x to PATH as shown to ensure that the interpreter will be placed in your execution path.

Then just click **Install Now**. That should be all there is to it. A few minutes later you should have a working Python 3 installation on your system.

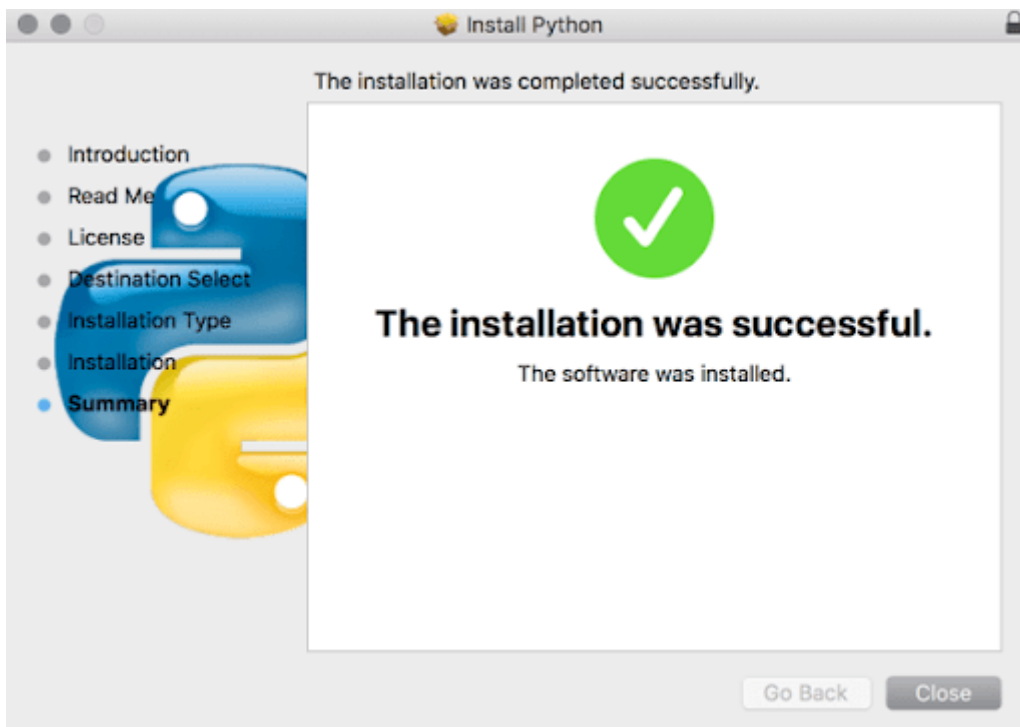


Fig 1.3 Finish installation

Garbage collection may happen at any time. Ideally, it will occur when a program is idle. It is guaranteed to be triggered if there is insufficient free memory on the heap to allocate a new object; this can cause a program to stall momentarily.

1.4.1 SETTING UP PATH

Programs and other executable files can be in many directories, so operating systems provide a search path that lists the directories that the OS searches for executables.

The path is stored in an environment variable, which is a named string maintained by the operating system. This variable contains information available to the command shell and other programs.

The **path** variable is named as PATH in Unix or Path in Windows (Unix is case sensitive; Windows is not).

In Mac OS, the installer handles the path details. To invoke the Python interpreter from any particular directory, you must add the Python directory to your path

To add the Python directory to the path for a particular session in Windows –

At the command prompt – type `path %path%;C:\Python` and press Enter.

Note – C:\Python is the path of the Python directory.

1.5 LOGISTIC REGRESSION

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on. Multinomial logistic regression can model scenarios where there are more than two possible discrete outcomes. Logistic regression is a useful analysis method for classification problems, where you are trying to determine if a new sample fits best into a category. As aspects of cyber security are classification problems, such as attack detection, logistic regression is a useful analytic technique.

1.5.1 PYTHON LIBRARY

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.

PANDAS:

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. In this tutorial, we will learn the various features of Python Pandas and how to use them in practice.

MATPLOTLIB:

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPython or Tkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

SEABORN:

Seaborn is an open source, BSD-licensed Python library providing high level API for visualizing the data using Python programming language. Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data.

CHAPTER 2

LITERATURE SURVEY

1. J. Neelaveni and M. S. G. Devasana applied machine learning for Alzheimer Prognosis.
2. V. K. Yarasuri et al proposed a machine learning based model for Hepatitis prognosis.
3. Wickramasinghe et al applied machine learning algorithms for diet prediction. The system was used for dietary prognosis for kidney diseases.
4. A. Maurya et al also applied ML for recommending diet plan for patients suffering from kidney disease prognosis.
5. V. Vats et al proposed an approach for prognosis of liver diseases using ML algorithms.

After going through this literature some issues are framed that include Diabetes is a very big challenge in Indian and some necessary action should be taken to prevent it.

Machine learning is very useful technique to predict diseases using medical and its power is to be explored more.

Logistic Regression is a very popular machine model. It works on supervised machine model.

In supervised machine learning model we have a teacher and the model is trained on the

instructions of a teacher/critic. It is very useful in solving the classification problems.

A lot of other authors also applied machine algorithm in prediction and detection of various diseases.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 OBJECTIVE

The main objective is to classify the data as diabetic or non-diabetic and improve the classification accuracy. It is observed that techniques like Support vector machine, knn and different algorithm and Artificial Neural Network are most suitable for implementing the Diabetes prediction system.

This Project is basically used for all person. The datasets consists of several medical parameter (independent) variables and one target variable, (Outcome). Predictor variables includes pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedegree Function, Age and Outcome).

I plan on gaining additional skills by taking related dataset and continuing my involvement with a variety of professional associations in organization and college. I noticed that my work provides in-diabetes training is good and showing accuracy rate high so I would certainly be interested in taking Diagnosis related dataset.

3.1.1 INTRODUCTION

Diabetes is a very familiar word in the present world and crucial challenges in both developed and developing countries . The insulin hormone in the body produced by the pancreas allows glucose to pass from the food into the bloodstream. The lack of that hormone due to malfunctioning of the pancreas forms diabetes which can result in coma, renal and retinal failure, pathological destruction of pancreatic beta cells, cardiovascular dysfunction, cerebral vascular dysfunction, peripheral vascular diseases, sexual dysfunction, joint failure, weight loss, ulcer, and pathogenic effects on immunity.

3.2 WHY ML

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning **algorithm** use historical data as input to predict new output values.

Machine learning is important because it gives enterprises a view of trends in customer behavior and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

TYPES:

Supervised Learning

Unsupervised Learning

Reinforcement Learning

- Supervised learning: In this type of machine learning data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.
- Unsupervised learning: This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.
- Semi-supervised learning: This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.
- Reinforcement learning: Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

3.2.1 K-means Algorithm

K-Means Clustering is an Unsupervised learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if $K=2$, there will be two clusters, and for $K=3$, there will be three clusters, and so on. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid.

The main aim of this algorithm is to minimize the sum of distances between the data

point and their corresponding clusters.

.

3.2.2 random State Regressor

Random_state is used to set the seed for the random generator so that we can ensure that the results that we get can be reproduced. Because of the nature of

splitting the data in train and test is randomised you would get different data assigned to the train and test data unless you can control for the random factor. For example, imagine you have numbers: 1,2,3,4,5 and you want to select two random numbers. Without using a `random_state` you would get different things: i.e. first 2,5 then 3,4 and so on.

When you control the random generator by setting the random state to 0, 1 or another number, everyone would get the same random numbers generated: i.e. `random_state=1` then the random numbers are 1,3 and everyone who will pick two random numbers using `random_state=1` from 1,2,3,4,5 will get the same two numbers as you: 1,3.

3.2.3

3.2.4 TERMS AND SCOPE

At present there are various terms used in the field to describe the diabetes prediction for creating record. In future this machine learning model may bind with various website which can provide real time data for Diabetes prediction. Also we added large historical data of Diabetes which can help to improve accuracy of the machine learning model. We can build an android app as user interface for interacting with user. For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset ,

CHAPTER 4

EXPERIMENTAL OR MATERIALS AND METHODS;

ALGORITHMS USED

4.1 HARDWARE REQUIREMENTS

1. Operating system : Windows 7,8,10
2. Processor : dual core 2.4 GHz (i5 or i7 series Intel processor or equivalent AMD)
3. RAM : Minimum 4GB primary memory
4. Hard Disk : Minimum 4GB hard disk space
5. Monitor : preferably colour monitor(16-bit colour) or above
6. Compact disk drive
7. A mouse and a keyboard

4.2 SOFTWARE REQUIREMENTS

1. Language Python 3.8.3/3.8.4
2. IDLE Python
3. Jupyter Notebook/chrome(google colab)

4.2.1 Overview Of Software Requirements

4.2.1.1 Python

Python is a general-purpose programming language in which you can use class also and designed to have as few implementation dependencies as possible. It is intended to let application developers *write once, run anywhere* (WORA), meaning that compiled Python code can run on all platforms that support Python without the need for recompilation. Python applications are typically compiled to IDLE python that can run on any virtual machine (VM) regardless of the underlying computer architecture. The syntax of Python is easier than C and C++. As of 2019, Python

was one of the most popular programming languages in use according to GitHub, particularly for client-server applications, with a reported 9 million developers.

METHODS

Data is collected from a local web portal OF KAGGLE for selling and buying cars. Since manual data collection is a time-consuming task, especially when there are numerous records to process, a “web scraper” as a part of this research is created to get this job done automatically and reduce the time for data gathering. Web scraping is a well-known technique to extract information from websites and save data into a local file or database. Manual data extraction is time-consuming and therefore web scrapers are used to do this job in a fraction of time. Web scrapers are programmed for specific websites and can mimic regular users from a website’s point of view. After raw data has been collected and stored to a local database, a data preprocessing step was applied.

4.2.1.2 IDLE PYTHON

Visual studio code is an integrated development environment (IDE) for Python. Visual studio code allows applications to be developed from a set of modular software components called *modules*. Visual studio program runs on Windows, macOS, Linux and Solaris. In addition to Python development, it has extensions for other languages like PHP, C, C++, HTML5, and JavaScript. Applications based on visual, including the visual studio code IDE, can be extended by third party developers.

CHAPTER 5

RESULTS AND DISCUSSION, PERFORMANCE

ANALYSIS

5.1 ANALYSIS

A python Diabetes prediction can be developed by using a number of components of Python. Here the major helping component is Google colab which is being used to construct the window and training and testing the data. Diabetes Prediction link(local host) has been generated using Spyder.

5.2 GOOGLE COLAB

Colab is basically a free Jupyter notebook environment running wholly in the cloud. Most importantly, Colab does not require a setup, plus the notebooks that you will create can be simultaneously edited by your team members — in a similar manner you edit documents in Google Docs. The greatest advantage is that Colab supports most popular Machine Learning Library which can be easily loaded in your notebook.



Fig 5.1 Google colab

What Colab Offers

As a developer, you can perform the following using Google Colab;

- Write and execute code in Python
- Create/Upload/Share notebooks
- Import/Save notebooks from/to Google Drive
- Import/Publish notebooks from GitHub
- Import external datasets
- Integrate PyTorch, TensorFlow, Keras, OpenCV
- Free Cloud service with free GPU

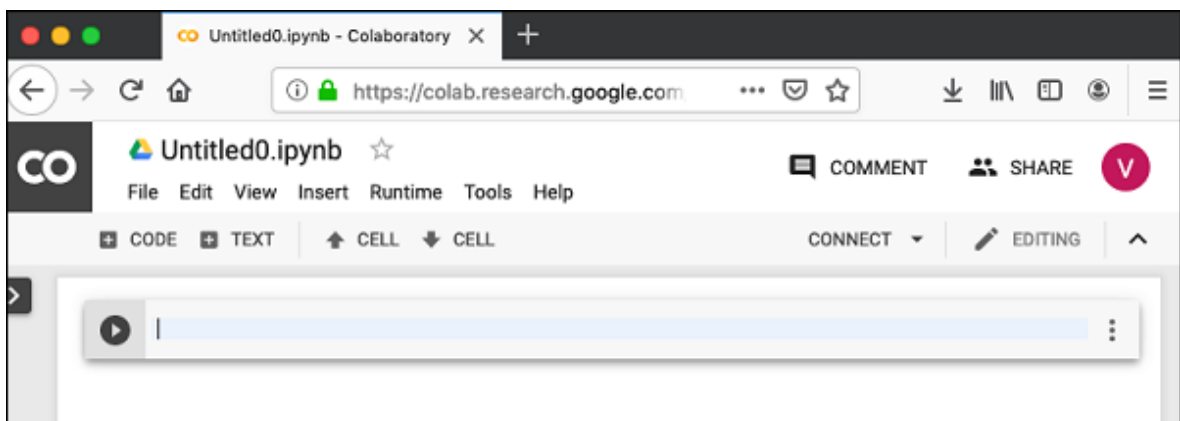


Fig 5.2 Opening window

In the above shown figure code is implemented successfully and it open in another window where we put number or data name.



Fig 5.3 Input Window

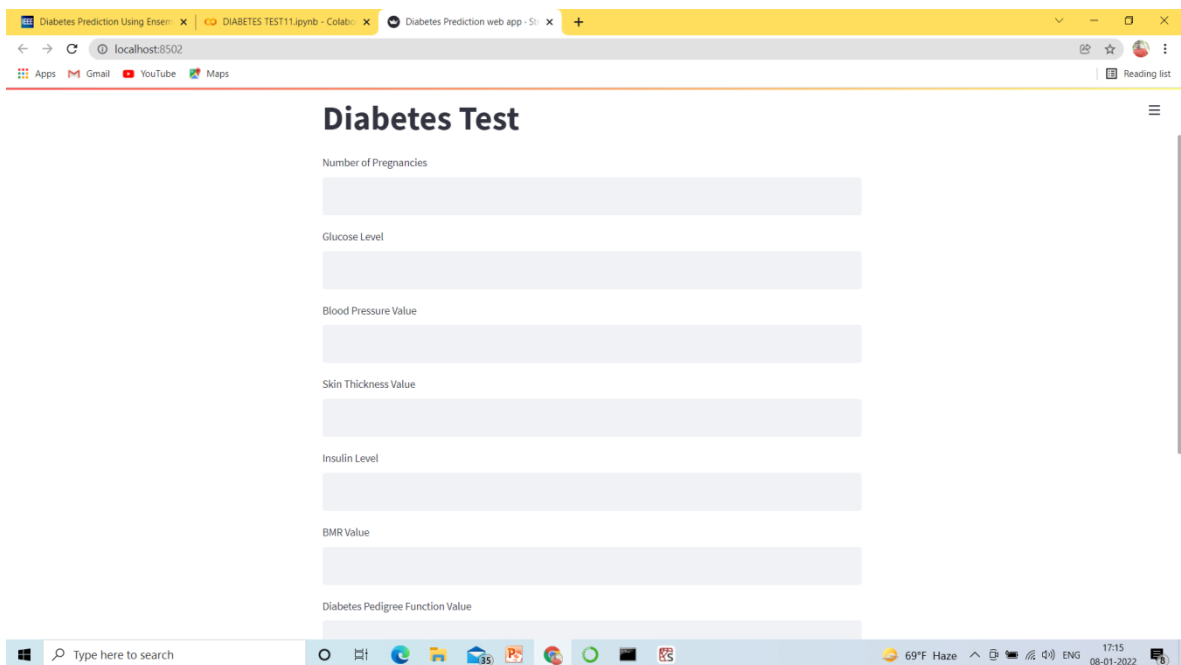


Fig 5.4 Winning Window

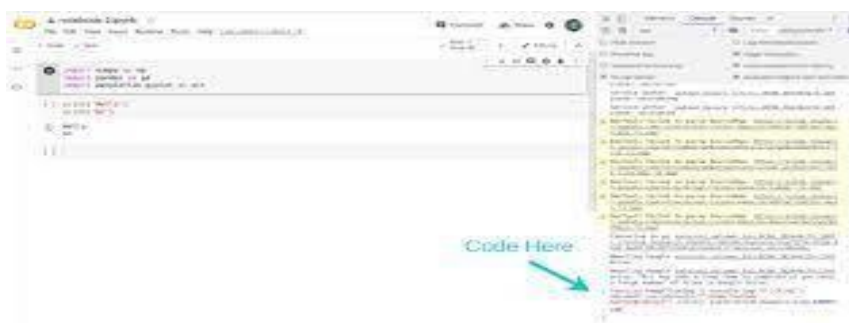
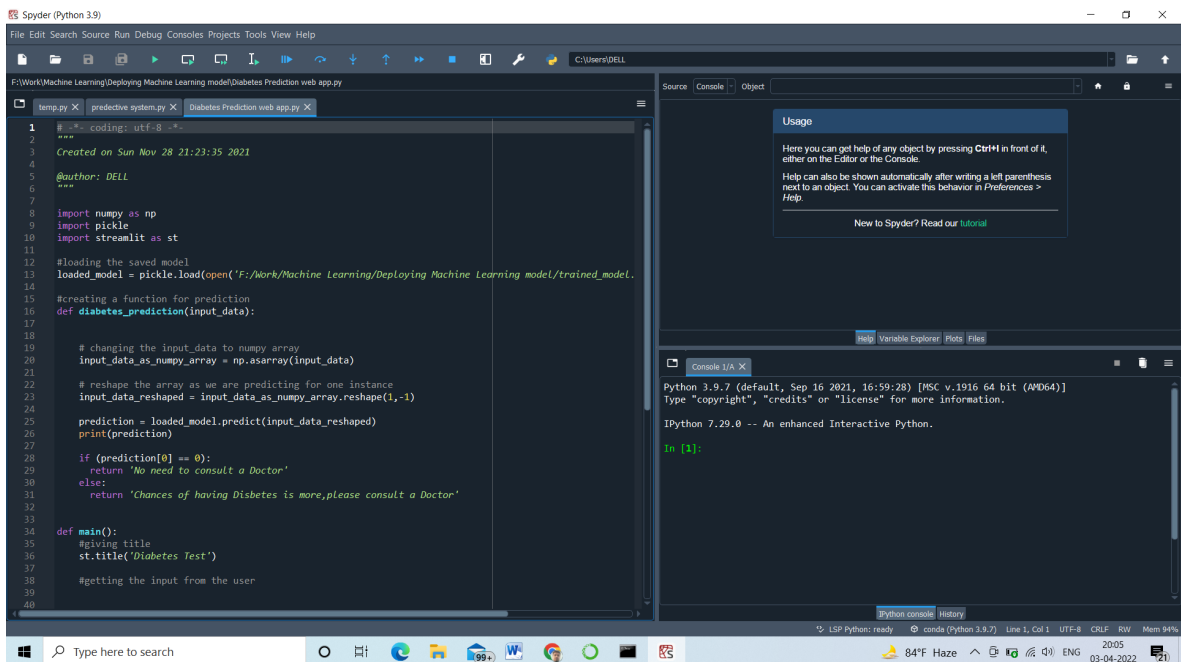
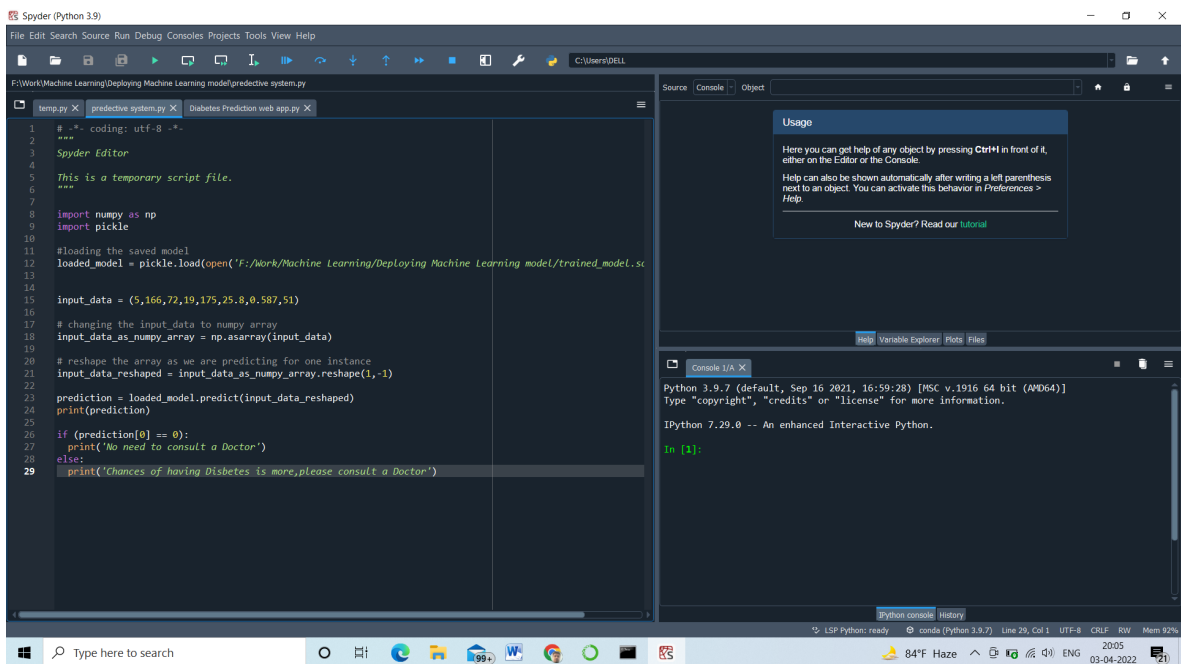


Fig 5.5 Losing Window

5.2.1 LANGUAGE CODE



5.3 RESULT AND DISCUSSION

No of Pregnancies	Glucose value	Blood pressure value	Skin thickness	Sugar level	BMI value	Diabetes Pedegree function	Age	Diabetic or Non-Diabetic
0	137	40	35	168	43.1	2.288	33	D
2	197	70	45	543	30.5	0.158	53	D
1	89	66	23	94	28.1	0.167	21	N.D

In this phase we get the result from machine after giving the eight input. Machine has been trained on the PIMA dataset. It is shown by taking three data in the above rows.

If the disease is detected early, then it can be reduced from becoming complicated.

This project is working very well on datasets. By improving it in the future, we can also find out the level of sugar. On the basis of sugar level machine can also suggest medicine.

CHAPTER 6

SUMMARY AND CONCLUSIONS

This project has greatly improved my skill in development of a prototype from making data.

After practise of some days it helps to improve to make the project. Some basics concepts of python. how google colab can be used to create an interactive window, how action listener can be used to perform different actions on the text field, and many more things.

In this study, various machine learning algorithms are applied on the dataset and the classification has been done using various algorithms of which SVM Regression gives highest accuracy of 96%. Application of spyder gave as best model with accuracy of %. We have seen comparison of machine learning algorithm accuracies with two different datasets. It is clear that the model improves accuracy and precision of diabetes prediction with this dataset compared to existing dataset. Further this work can be extended to find how likely non-diabetic people can have diabetes in next few years.

BIBLIOGRAPHY

1. <https://www.python.org/>
2. https://en.wikipedia.org/wiki/Machine_learning
3. <https://www.kaggle.com/>
4. <https://www.w3schools.com/python/pandas/default.asp>
5. <https://www.sciencedirect.com/science/article/pii/S2001037016300733>

REFERENCES

1. NPTEL – THE JOY OF COMPUTING USING PYTHON
3. COURSERA - PROGRAMMING FOR EVERYBODY
4. SOLOLEARN - PYTHON 3 TUTORIAL

SOURCE CODE

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score

# loading the diabetes dataset to a pandas DataFrame
diabetes_dataset = pd.read_csv('/content/diabetes.csv')

# printing the first 5 rows of the dataset
diabetes_dataset.head()

# number of rows and Columns in this dataset
diabetes_dataset.shape

# getting the statistical measures of the data
diabetes_dataset.describe()

diabetes_dataset['Outcome'].value_counts()

diabetes_dataset.groupby('Outcome').mean()

# separating the data and labels
X = diabetes_dataset.drop(columns = 'Outcome', axis=1)
Y = diabetes_dataset['Outcome']

print(X)
print(Y)

X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size =
0.2, stratify=Y, random_state=2)

print(X.shape, X_train.shape, X_test.shape)

classifier = svm.SVC(kernel='linear')

#training the support vector Machine Classifier
classifier.fit(X_train, Y_train)

# accuracy score on the training data
X_train_prediction = classifier.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
print('Accuracy score of the training data : ', training_data_accurac
y)
```

```

# accuracy score on the test data
X_test_prediction = classifier.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

print('Accuracy score of the test data : ', test_data_accuracy)

input_data = (5,166,72,19,175,25.8,0.587,51)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the array as we are predicting for one instance
input_data_resaped = input_data_as_numpy_array.reshape(1,-1)

prediction = classifier.predict(input_data_resaped)
print(prediction)

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

import pickle

filename = 'trained_model.sav'
pickle.dump(classifier, open(filename, 'wb'))

# loading the saved model
loaded_model = pickle.load(open('trained_model.sav', 'rb'))

input_data = (5,166,72,19,175,25.8,0.587,51)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the array as we are predicting for one instance
input_data_resaped = input_data_as_numpy_array.reshape(1,-1)

prediction = loaded_model.predict(input_data_resaped)
print(prediction)

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

```