A

Industrial Training Report

On

DATA ANALYSIS WITH MACHINE LEARNING

Submitted

In partial fulfilment

For the award of the Degree of

Bachelor of Technology

in Department of Computer Science and Engineering

Submitted To	**Submitted By**
Dr. Subhash Chandra	Ashita
HOD, CSE	20ERWCS011

Department of Computer Science and Engineering

Rajasthan College of Engineering for Women

Rajasthan Technical University

Session 2023-2024

DECLARATION

I hereby declare that the discussion entitled "**Data Analysis with Machine Learning**" being submitted by **me** toward the partial fulfilment of the Degree of Bachelor, In Department of Computer Science Engineering is a project work carried out by me. Under the supervision of Mr. **Vinod Todwal**, and haven't been submitted anywhere else. I will be solely responsible if any kind of plagiarisms is found.

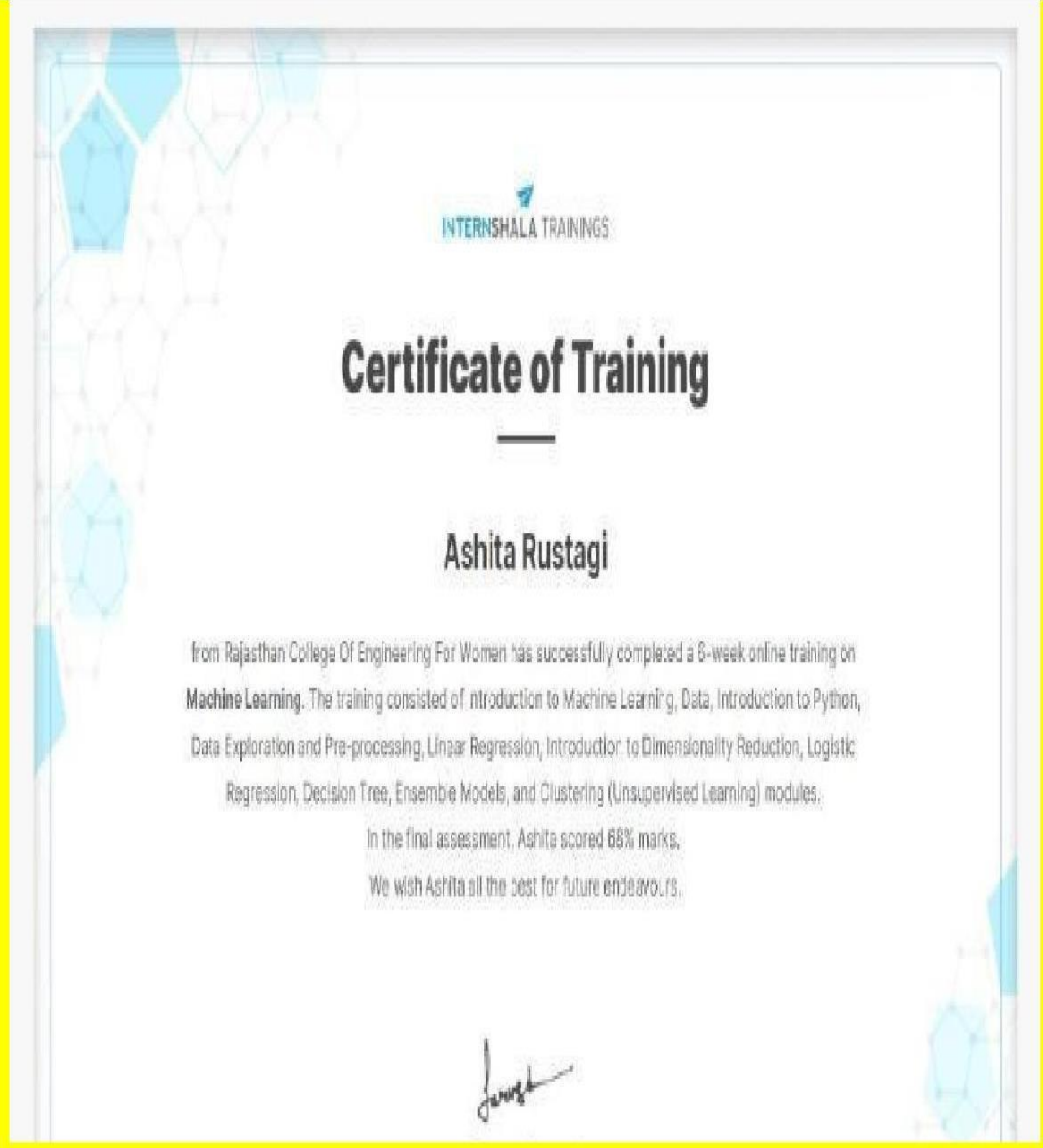
Ashita

20ERWCS011

Rajasthan College of Engineering for Women

Counter Signed by

Mr. Vinod Todwal



INTERNSHALA TRAININGS

Certificate of Training

Ashita Rustagi

from Rajasthan College Of Engineering For Women has successfully completed a 6-week online training on **Machine Learning.** The training consisted of Introduction to Machine Learning, Data, Introduction to Python, Data Exploration and Pre-processing, Linear Regression, Introduction to Dimensionality Reduction, Logistic Regression, Decision Tree, Ensemble Models, and Clustering (Unsupervised Learning) modules.

In the final assessment, Ashita scored 68% marks.

We wish Ashita all the best for future endeavours.

ACKNOWLEDGEMENT

I like to share my sincere gratitude toward all those who helps me in the completion of the project. During the project I have faced many challenges because of lack of experience but this training process helps me to get over from all the challenges and in the final compilation of my idea to shaped sculpture.

Minute aspects of the project work.

In the last I would like to thanks the management of Rajasthan College of Engineering for Women for Providing me such an opportunity to learn from these experiences.

I am also thankful to my friends and parents who have inspired me to face all the challenges and win over them in life.

TABLE OF CONTENTS

TABLE OF FIGURES

ABSTRACT

The linear Regression is one of the simplest and most widely used Machine Learning algorithm. This algorithm is a kind of Supervised Learning which deals with the labeled data to be analyzed and make the prediction.

Sir Francis Galton is the one who first proposed the concept of Linear Regression in 1894.

Linear Regression works with two kinds of variables x which is dependent variable and y is independent Variable. Linear Regression provides the linear relationship between them. Linear Regression used the Least Square method to determine the best fit for every specific part of the data.

The Linear Regression has two types simple and multiple regression. This type of regression is applicable or applied only with the cleansed data by means of data mining theory.

Machine Learning is a collection of algorithms which not only manage the data but also in finding new patterns to find out the new and best opportunities which is beneficial for the organizations or the company. Machine Learning is a branch of AI that are mostly concerns with the current or the most recent updates of the data of any particular sector.

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF PROJECT

Data Management on a large scale or we can say management of big data is very complex by simple techniques thus Machine Learning is used for the evaluation, analyzing and making prediction with the help of advanced algorithms and technologies which is physically impossible.

Machine Learning is simply all about data and its analyzing to make predictions and organizing the data in a proper format. Every change or the information which is related to any specific entity or topic is data for it. So, a million of byte of data is inserting, deleting and updating in every second all over the world as all the data are connected to each other through internet.

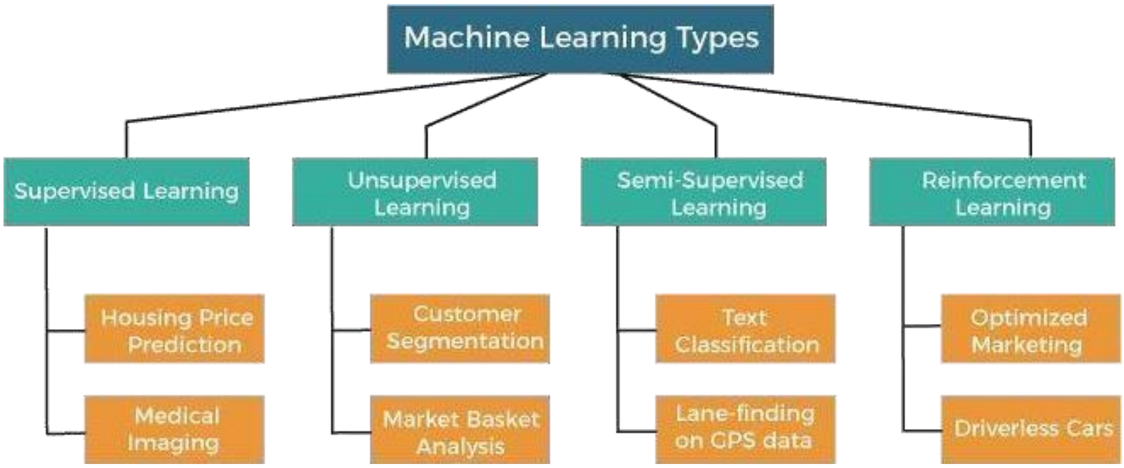
1.2 INTRODUCTION OF MACHINE LEARNING

Machine Learning is a branch of AI that enables computer to self-learn from the training data and improve over time without being explicitly programmed. Machine Learning is a programming computer to optimize a performance criterion using example data and past experiences. Machine Learning system builds the prediction model for analyzing based on algorithm applied.

The main purpose of Machine Learning is not only to train the model on previous data but also make the prediction value based on it for the best outcome. Machine Learning also helps in discovering the new patterns in the data. Machine Learning uses different kinds of algorithm to parse the data and it also helps in decision making. Machine Learning used in recognizing the practical benefits to the individuals in the real world.

There are mainly four types of Learning algorithms in Machine Learning

1. Supervised Machine Learning Algorithm
2. Unsupervised Machine Learning Algorithm
3. Semi Supervised Machine Learning Algorithm
4. Reinforcement Machine Learning Algorithm



1.3 INTRODUCTION OF DATA

Data is a raw collection of facts and figures. Data is basically any collection of different kind of facts in the forms such as numeric, alphabetical, relations, diagram, text, images, flowcharts and many more. Data is the main factor of Machine Learning on which almost all the operations and algorithms are being applied.

The unprocessed or unorganized information related to any specific is known as the data for that specific. As Machine Learning is used for prediction. So, as data is important for prediction, decision making, Problem solving, improving process to get the best outcome for the future.

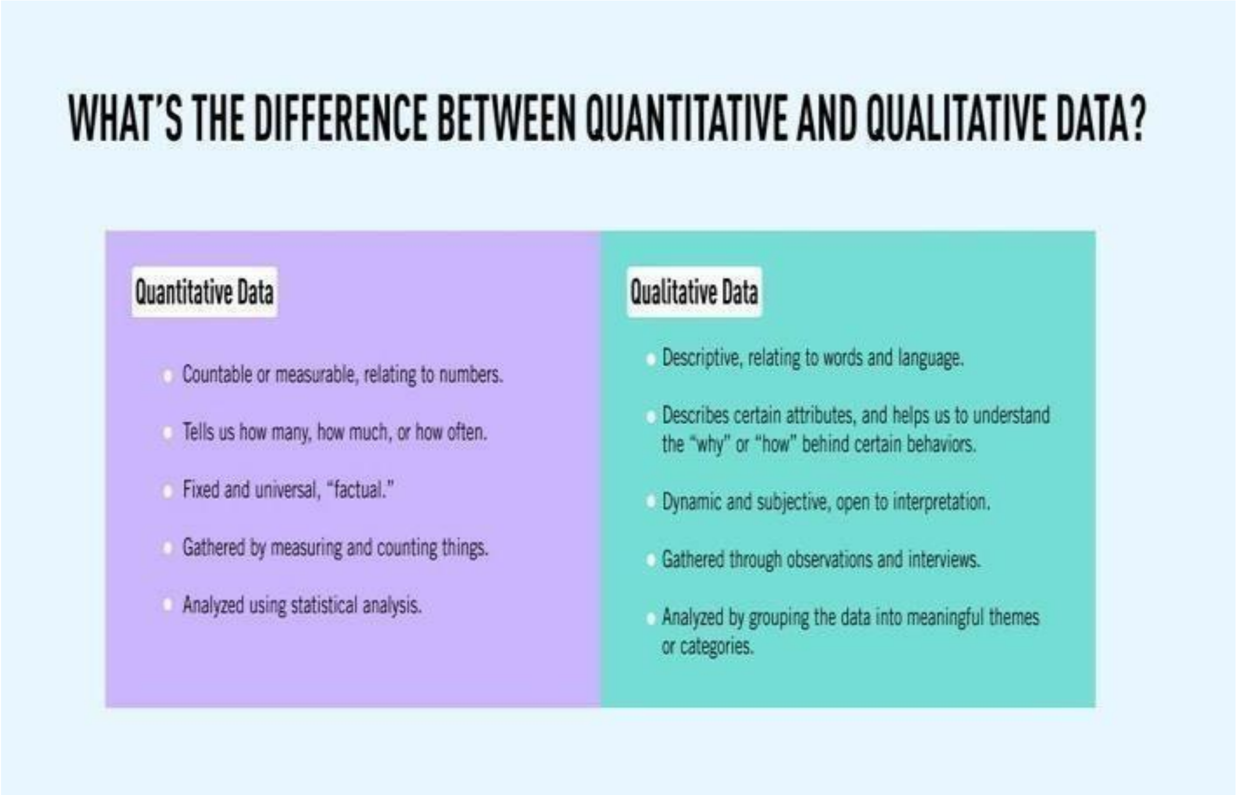
Data is generally represented into two forms, which are:

1. Qualitative Data

   Qualitative Data is descriptive which can be observed but not measured.

2. Quantitative Data

   Quantitative Data is any data that can be counted or measured



WHAT'S THE DIFFERENCE BETWEEN QUANTITATIVE AND QUALITATIVE DATA?

Quantitative Data

- Countable or measurable, relating to numbers.
- Tells us how many, how much, or how often.
- Fixed and universal, "factual."
- Gathered by measuring and counting things.
- Analyzed using statistical analysis.

Qualitative Data

- Descriptive, relating to words and language.
- Describes certain attributes, and helps us to understand the "why" or "how" behind certain behaviors.
- Dynamic and subjective, open to interpretation.
- Gathered through observations and interviews.
- Analyzed by grouping the data into meaningful themes or categories.

1.4 INTRODUCTION OF PYTHON

Python was developed by Guido Van Rossum in 1991. Python is a general-purpose language.

Python is a high-level interpreted language.

The most recent version if Python is Python3. Python is object-oriented language.

Python is simple as English in terms of reading and understanding.

There are different python libraries that are used during the project such as numpy, pandas, Matplotlib.pyplot, seaborn and many more.

CHAPTER 2

LITERATURE REVIEW

2.1 MACHINE LEARNING

Machine Learning is simply a branch of AI which uses different kinds of algorithm to maintain and organize a very vast amount of data in a proper format. Machine Learning builds different models according to algorithm used which help the user in decision making and predictions.
Machine Learning allows the data to self-learn.

2.2 OBJECTIVES OF MACHINE LEARNING

The main purpose of Machine Learning is to train the data for the model to predict the value of some quantity which gives the best outcome. The main purpose of Machine Learning also includes is to discover the new pattern in your data which leads you toward decision making.

2.3 SCOPE OF MACHINE LEARNING

Machine Learning algorithms are used in almost all the fields. Some of the application of the Machine Learning are:

1. Spam Filtering
2. Fraud Detection
3. Smart Healthcare System
4. Speech Recognition

5. Computer Vision
6. Smart Transportation

2.4 KINDS OF MACHINE LEARNING

Machine Learning is a collection of different algorithms which are classified into different types of Machine Learning.

2.4.1 SUPERVISED MACHINE LEARNING

Supervised Machine Learning algorithm is designed to train the data for the model to get The desired output. The algorithm measures its accuracy through the loss function and manipulates until the error has been sufficiently minimized.

There are mainly two types of Machine Learning algorithms are:

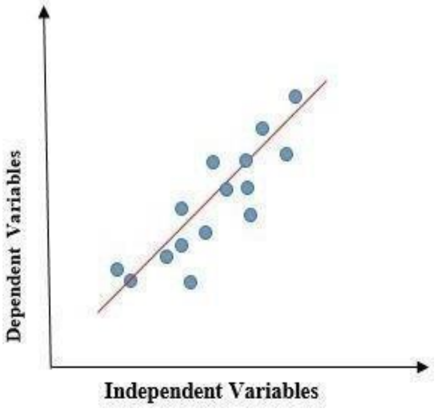
A. REGRESSION

Regression is a kind of supervised algorithm which deals with the numerical type data. Regression is a technique which is used to predict the continuous data. Regression algorithm is also of two kinds. They are:

A.1 LINEAR REGRESSION

Linear Regression is a model used to determine the regression analysis. Regression analysis tells us the relationship between dependent and independent variables.
Linear Regression only support the numerical type data. The regression is statistical in nature. It is used for predictive analysis. Linear Regression can be calculated with the equation:

y=mx +c y=independent variable x= dependent variable c= constant and m = slope



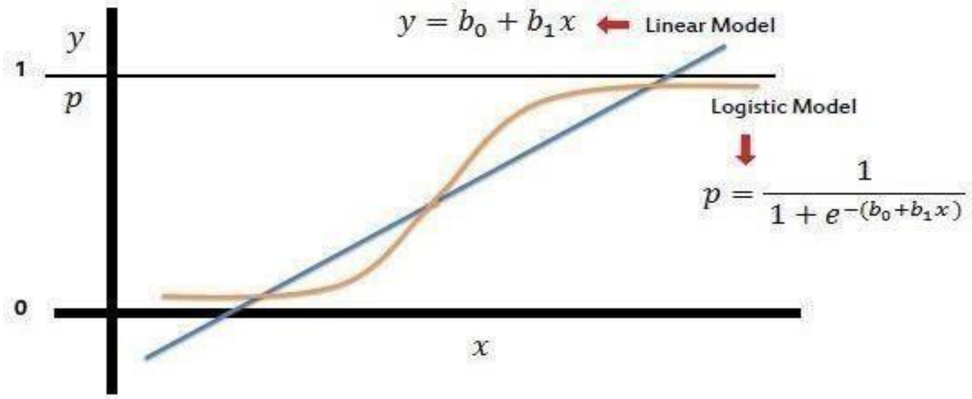
A.2 LOSS FUNCTION FOR LINEAR REGRESSION

The Loss Function is the difference between the predicted value and the actual value. It is the mean squared error value between them. This function is also known as cost function.

A.3 LOGISTIC REGRESSION

Logistic Regression is used for categorical variables. The output of Logistic Regression is either 0 or 1. This algorithm is used to solve the classification problem. The Logistic Regression equation is:

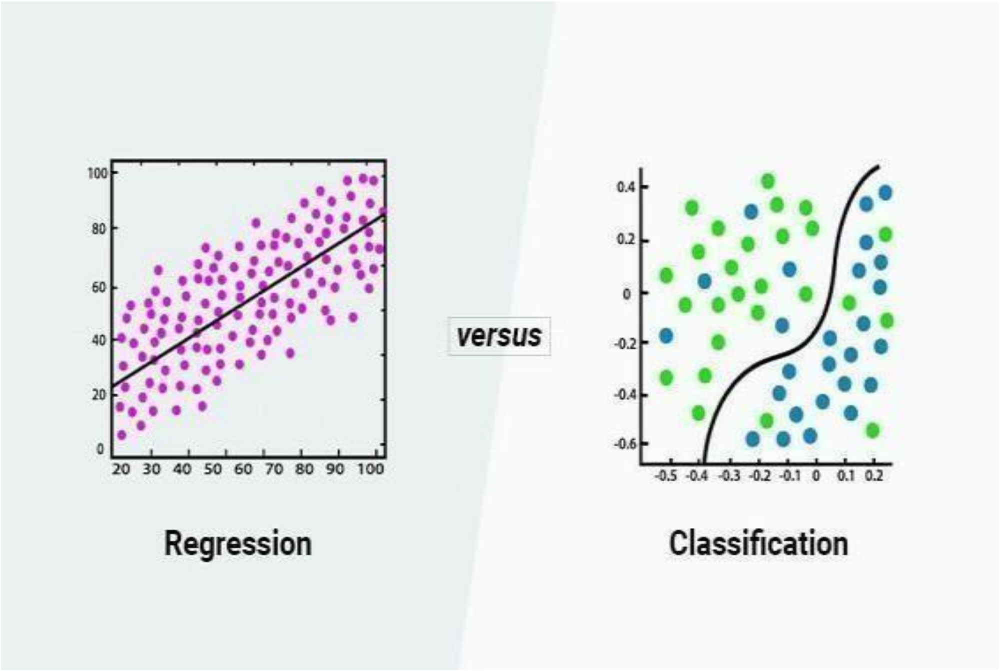
y=b0+b1x1+b2x2+ ----------------- +bnxn



B. CLASSIFICATION

In Classification, the model is built on training data but is also applied on test data before the use in general. Classification is a technique in which the data having same characteristics are placed in one group and the prediction is done accordingly on unseen data. The main algorithm that comes under the classification are:

Naïve Bayes, K Nearest Neighbor, Decision Tree, Support Vector Machine, Random Forest and so on.



2.4.2 UNSUPERVISED MACHINE LEARNING

Unsupervised learning is a type of Machine Learning that learns pattern from unlabeled data. It is used to find hidden pattern. There are mainly two types of Unsupervised Learning:

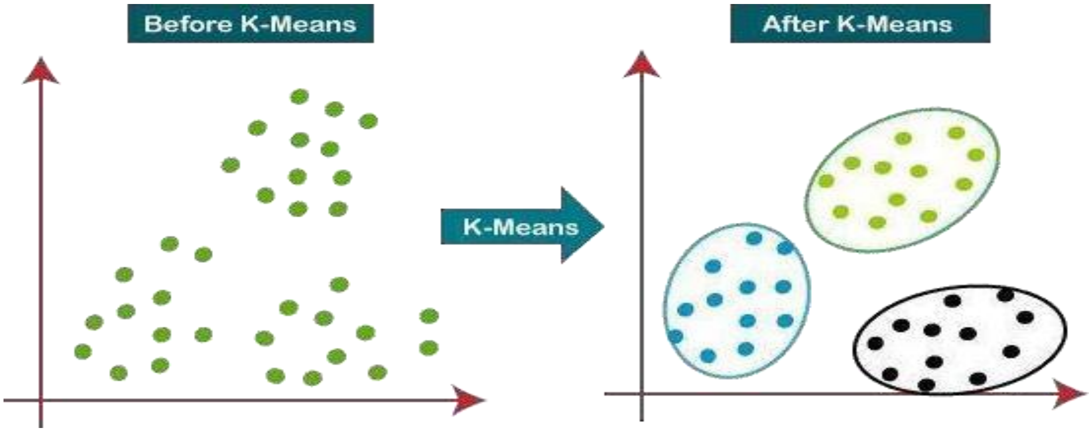
A. CLUSTERING

Clustering is a process in which the data having the same characteristics are placed into the same group with the help of different algorithms. They are:

A.1 K MEANS CLUSTERING ALGORITHM

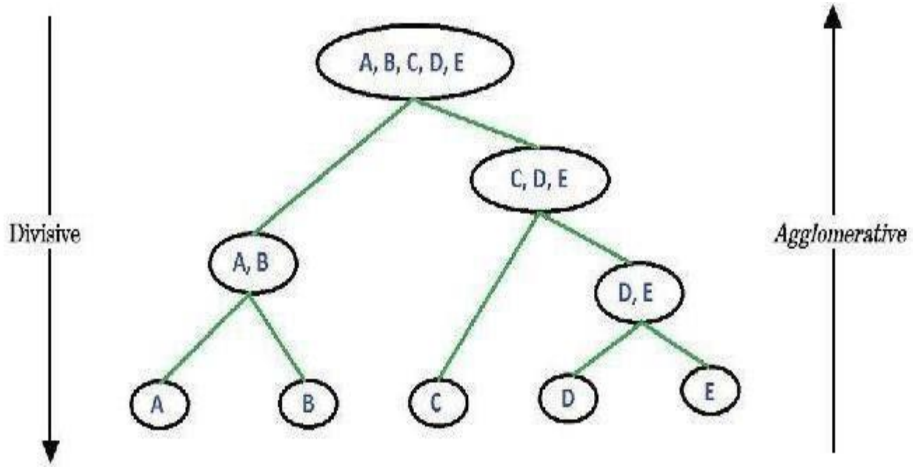
K Means Clustering algorithm is used to group the unlabeled data into different clusters. In this algorithm K defines the number of predefined values of clusters that are needed to be create in the process. This algorithm is also called Centre Base or Centroid algorithm. In this algorithm we have to find the Euclidean Distance for every new data from the Centroid to decide in which cluster the data is placed. The formula for Euclidean Distance is:

K=((x-x1) (x-x1) +(y-y1) (y-y1)) ^1/2



A.2 HIERARICHICAL CLUSTERING

Hierarchical Clustering is a type of clustering in which the algorithm is followed by the data as inherit of the characteristics of the data. There are mainly two types of Hierarchical clustering. They are: Agglomerative Divisive



B. ASSOCIATIVE RULE MINING

Associative Rule Mining is a kind of Unsupervised Learning algorithm which is used to check the dependency of one data element on another data element. The two main factor of this algorithm is:

B.1 SUPPORT

Support is a term which is defined as the percentage of the dataset in which a particular set of values is considered at a time.

B.2 CONFIDENCE

The percentage of transactions that contain a particular item or a set of items.

2.4.3 SEMI SUPERVISED MACHINE LEARNING

Semi Supervised learning is a collection of supervised and unsupervised learning. This type of learning is applicable for labeled and unlabeled data to train the model.

2.4.4 REINFORCEMENT MACHINE LEARNING

Reinforcement learning is a type of Machine Learning which allows machine and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance.

2.4.5 DEEP LEARNING

Deep Learning is a type of Machine Learning which guides the computer to work or process the data as a human brain does. Deep Learning have advance algorithm and features in it which help this learning to recognize the complex pictures, audio, video, text, graph and any other data source that lead to the better result and better prediction for future. Some application of Deep learning is as follow:

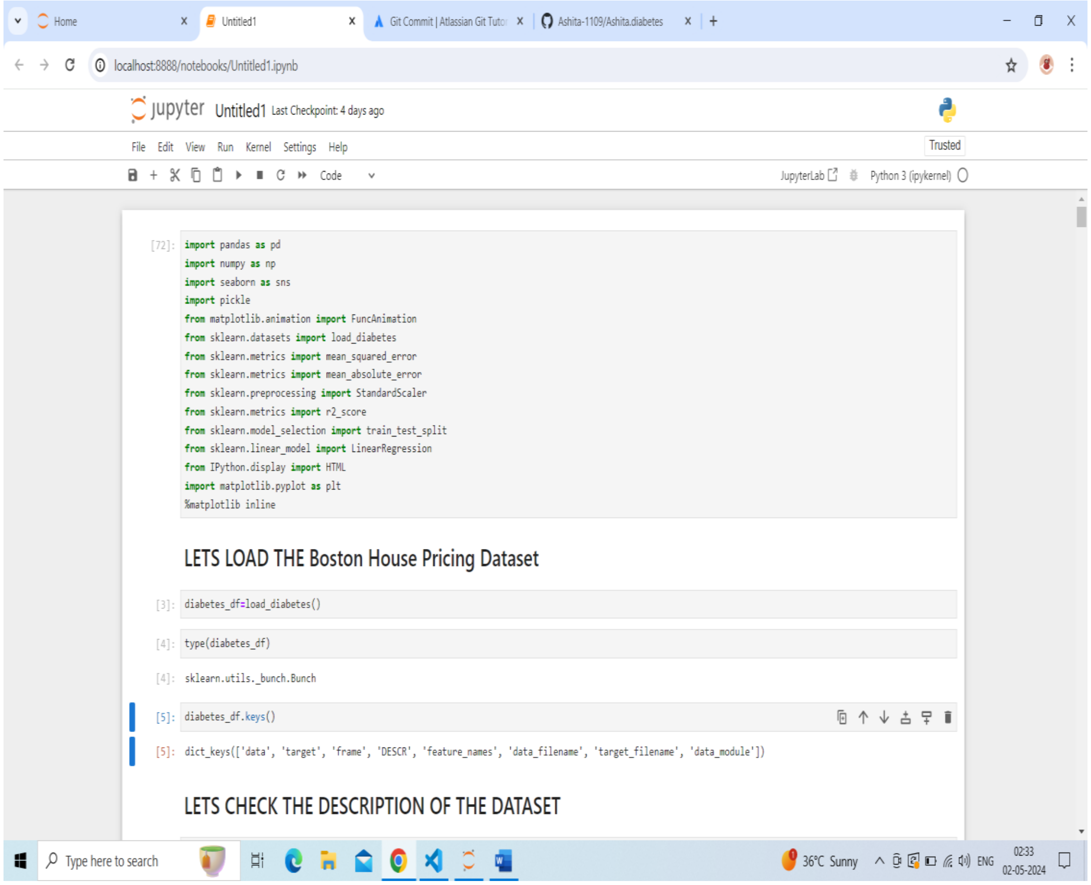
1. Digital Assistants
2. Fraud Detection
3. Automatic Face Recognition
4. Voice Activated Controller

CHAPTER 3

DESIGN OF PROJECT

The project in the training is of Linear Regression in Machine Learning. This project is Completed with stepwise process which include many steps. The project of Linear Regression is about how the data is managed to make the right decision as well as Prediction for the best.

3.1 IMPORTING PYTHON LIBRARIES



Home
Untitled1
Git Commit | Atlassian Git Tuto
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
36°C Sunny
ENG
02:33
02-05-2024

```
[72]: import pandas as pd
      import numpy as np
      import seaborn as sns
      import pickle
      from matplotlib.animation import FuncAnimation
      from sklearn.datasets import load_diabetes
      from sklearn.metrics import mean_squared_error
      from sklearn.metrics import mean_absolute_error
      from sklearn.preprocessing import StandardScaler
      from sklearn.metrics import r2_score
      from sklearn.model_selection import train_test_split
      from sklearn.linear_model import LinearRegression
      from IPython.display import HTML
      import matplotlib.pyplot as plt
      %matplotlib inline
```

LETS LOAD THE Boston House Pricing Dataset

```
[3]: diabetes_df=load_diabetes()

[4]: type(diabetes_df)

[4]: sklearn.utils._bunch.Bunch

[5]: diabetes_df.keys()

[5]: dict_keys(['data', 'target', 'frame', 'DESCR', 'feature_names', 'data_filename', 'target_filename', 'data_module'])
```

LETS CHECK THE DESCRIPTION OF THE DATASET

PANDAS

Pandas is a python library which is used in data analysis, manipulation and its cleansing Pandas support operation like sorting, re-indexing and concatenation.

NUMPY

Numpy is a python library which is used for numerical data. This library supports large Metrics and multi-dimensional data. This library consists of in-built mathematical function For easy computations.

MATPLOTLIB.PYPLOT

This library is used for analysis of data. Mat plot library helps to plot the data in the graph representation and helps us to determine the relationship between different elements.

SEABORN

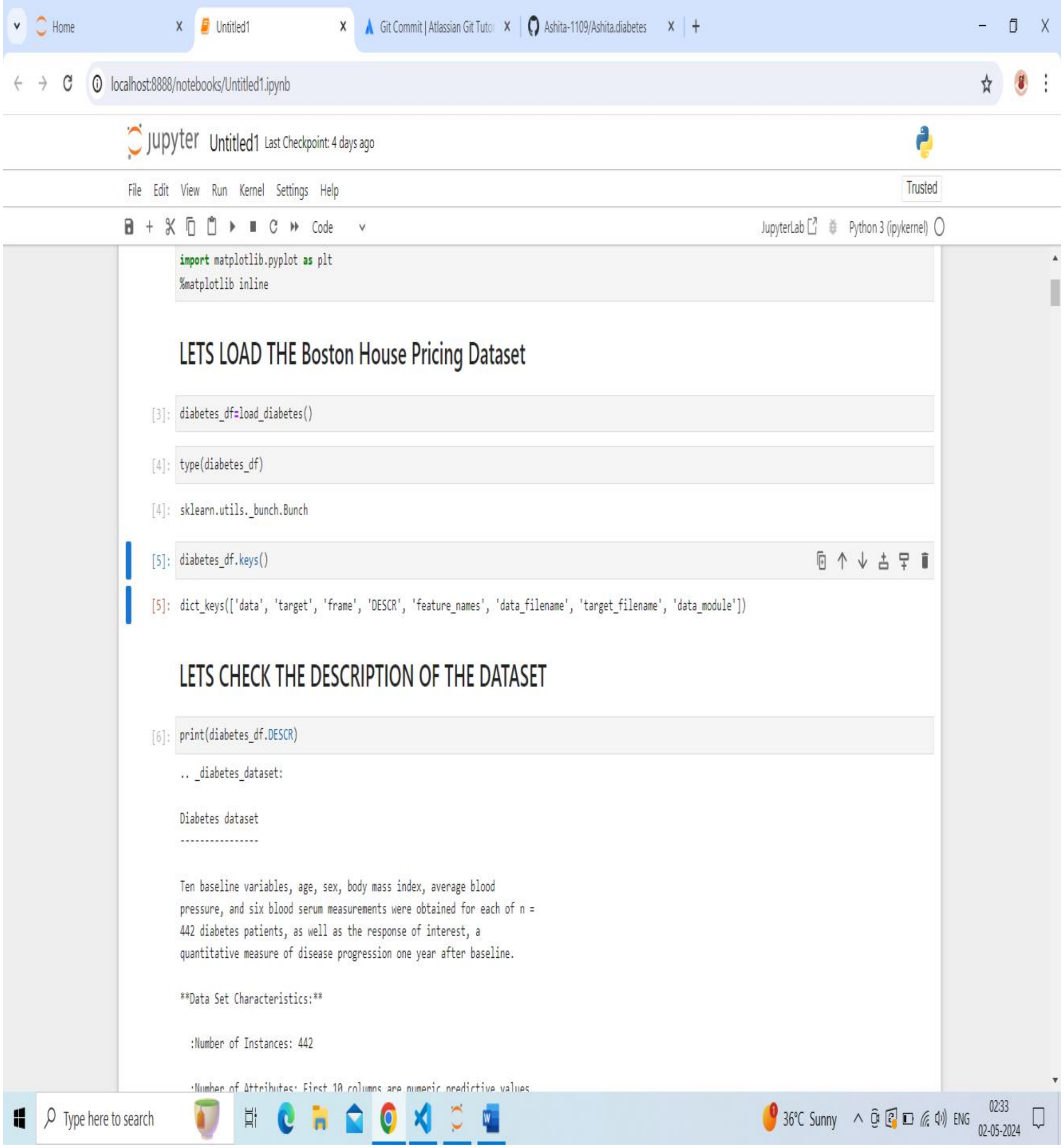
Seaborn library is a type of python library which provide a large interface for drawing attractive and informative statistical graphics.

SKLEARN

Sklearn is a python library used for the implementation of Machine Learning Model and Statistical model.

3.2 IMPORTING OF DATA

The data is imported by its address. The data is collected from different sources to analyze the real status of the environment and provides the appropriate suggestion. Data is imported by using the read file command used to display the data on the jupyter notebook with the help of python.


Home
Untitled1
Git Commit | Atlassian Git Tutor
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
36°C Sunny
ENG
02:33
02-05-2024

```
import matplotlib.pyplot as plt
%matplotlib inline
```

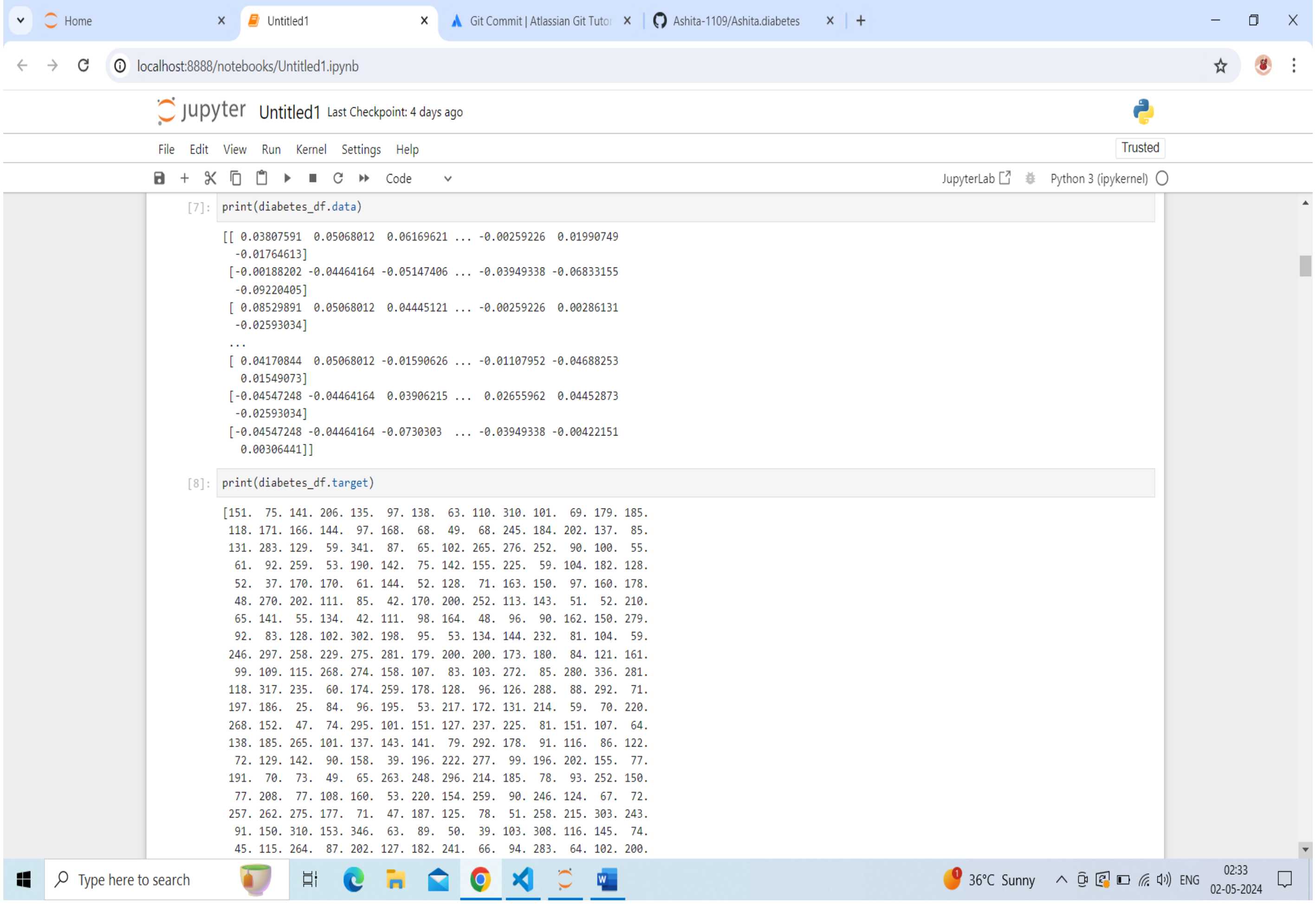
LETS LOAD THE Boston House Pricing Dataset

```
[3]: diabetes_df=load_diabetes()

[4]: type(diabetes_df)

[4]: sklearn.utils._bunch.Bunch

[5]: diabetes_df.keys()

[5]: dict_keys(['data', 'target', 'frame', 'DESCR', 'feature_names', 'data_filename', 'target_filename', 'data_module'])
```

LETS CHECK THE DESCRIPTION OF THE DATASET

```
[6]: print(diabetes_df.DESCR)
```

```
.. _diabetes_dataset:

Diabetes dataset
----------------

Ten baseline variables, age, sex, body mass index, average blood
pressure, and six blood serum measurements were obtained for each of n =
442 diabetes patients, as well as the response of interest, a
quantitative measure of disease progression one year after baseline.

**Data Set Characteristics:**

  :Number of Instances: 442

  :Number of Attributes: First 10 columns are numeric predictive values
```

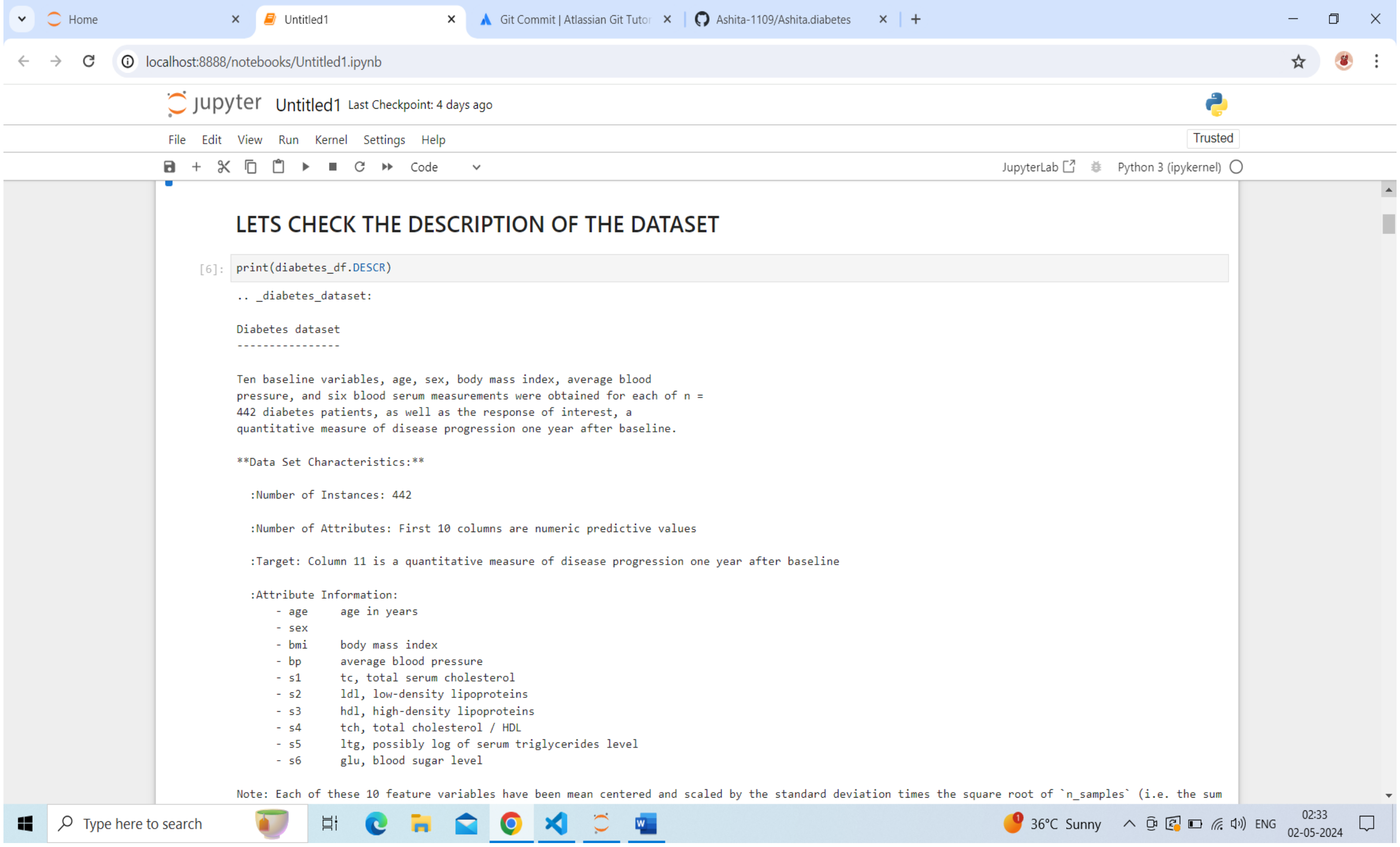


```
[7]: print(diabetes_df.data)
```

```
[[ 0.03807591  0.05068012  0.06169621 ... -0.00259226  0.01990749
  -0.01764613]
 [-0.00188202 -0.04464164 -0.05147406 ... -0.03949338 -0.06833155
  -0.09220405]
 [ 0.08529891  0.05068012  0.04445121 ... -0.00259226  0.00286131
  -0.02593034]
 ...
 [ 0.04170844  0.05068012 -0.01590626 ... -0.01107952 -0.04688253
   0.01549073]
 [-0.04547248 -0.04464164  0.03906215 ...  0.02655962  0.04452873
  -0.02593034]
 [-0.04547248 -0.04464164 -0.0730303  ... -0.03949338 -0.00422151
   0.00306441]]
```

```
[8]: print(diabetes_df.target)
```

```
[151.  75. 141. 206. 135.  97. 138.  63. 110. 310. 101.  69. 179. 185.
 118. 171. 166. 144.  97. 168.  68.  49.  68. 245. 184. 202. 137.  85.
 131. 283. 129.  59. 341.  87.  65. 102. 265. 276. 252.  90. 100.  55.
  61.  92. 259.  53. 190. 142.  75. 142. 155. 225.  59. 104. 182. 128.
  52.  37. 170. 170.  61. 144.  52. 128.  71. 163. 150.  97. 160. 178.
  48. 270. 202. 111.  85.  42. 170. 200. 252. 113. 143.  51.  52. 210.
  65. 141.  55. 134.  42. 111.  98. 164.  48.  96.  90. 162. 150. 279.
  92.  83. 128. 102. 302. 198.  95.  53. 134. 144. 232.  81. 104.  59.
 246. 297. 258. 229. 275. 281. 179. 200. 200. 173. 180.  84. 121. 161.
  99. 109. 115. 268. 274. 158. 107.  83. 103. 272.  85. 280. 336. 281.
 118. 317. 235.  60. 174. 259. 178. 128.  96. 126. 288.  88. 292.  71.
 197. 186.  25.  84.  96. 195.  53. 217. 172. 131. 214.  59.  70. 220.
 268. 152.  47.  74. 295. 101. 151. 127. 237. 225.  81. 151. 107.  64.
 138. 185. 265. 101. 137. 143. 141.  79. 292. 178.  91. 116.  86. 122.
  72. 129. 142.  90. 158.  39. 196. 222. 277.  99. 196. 202. 155.  77.
 191.  70.  73.  49.  65. 263. 248. 296. 214. 185.  78.  93. 252. 150.
  77. 208.  77. 108. 160.  53. 220. 154. 259.  90. 246. 124.  67.  72.
 257. 262. 275. 177.  71.  47. 187. 125.  78.  51. 258. 215. 303. 243.
  91. 150. 310. 153. 346.  63.  89.  50.  39. 103. 308. 116. 145.  74.
  45. 115. 264.  87. 202. 127. 182. 241.  66.  94. 283.  64. 102. 200.
```


Home
Untitled1
Git Commit | Atlassian Git Tutor
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
36°C Sunny
ENG
02:33
02-05-2024

LETS CHECK THE DESCRIPTION OF THE DATASET

```
[6]: print(diabetes_df.DESCR)
```

```
.. _diabetes_dataset:

Diabetes dataset
----------------

Ten baseline variables, age, sex, body mass index, average blood
pressure, and six blood serum measurements were obtained for each of n =
442 diabetes patients, as well as the response of interest, a
quantitative measure of disease progression one year after baseline.

**Data Set Characteristics:**

  :Number of Instances: 442

  :Number of Attributes: First 10 columns are numeric predictive values

  :Target: Column 11 is a quantitative measure of disease progression one year after baseline

  :Attribute Information:
      - age     age in years
      - sex
      - bmi     body mass index
      - bp      average blood pressure
      - s1      tc, total serum cholesterol
      - s2      ldl, low-density lipoproteins
      - s3      hdl, high-density lipoproteins
      - s4      tch, total cholesterol / HDL
      - s5      ltg, possibly log of serum triglycerides level
      - s6      glu, blood sugar level

Note: Each of these 10 feature variables have been mean centered and scaled by the standard deviation times the square root of `n_samples` (i.e. the sum
```

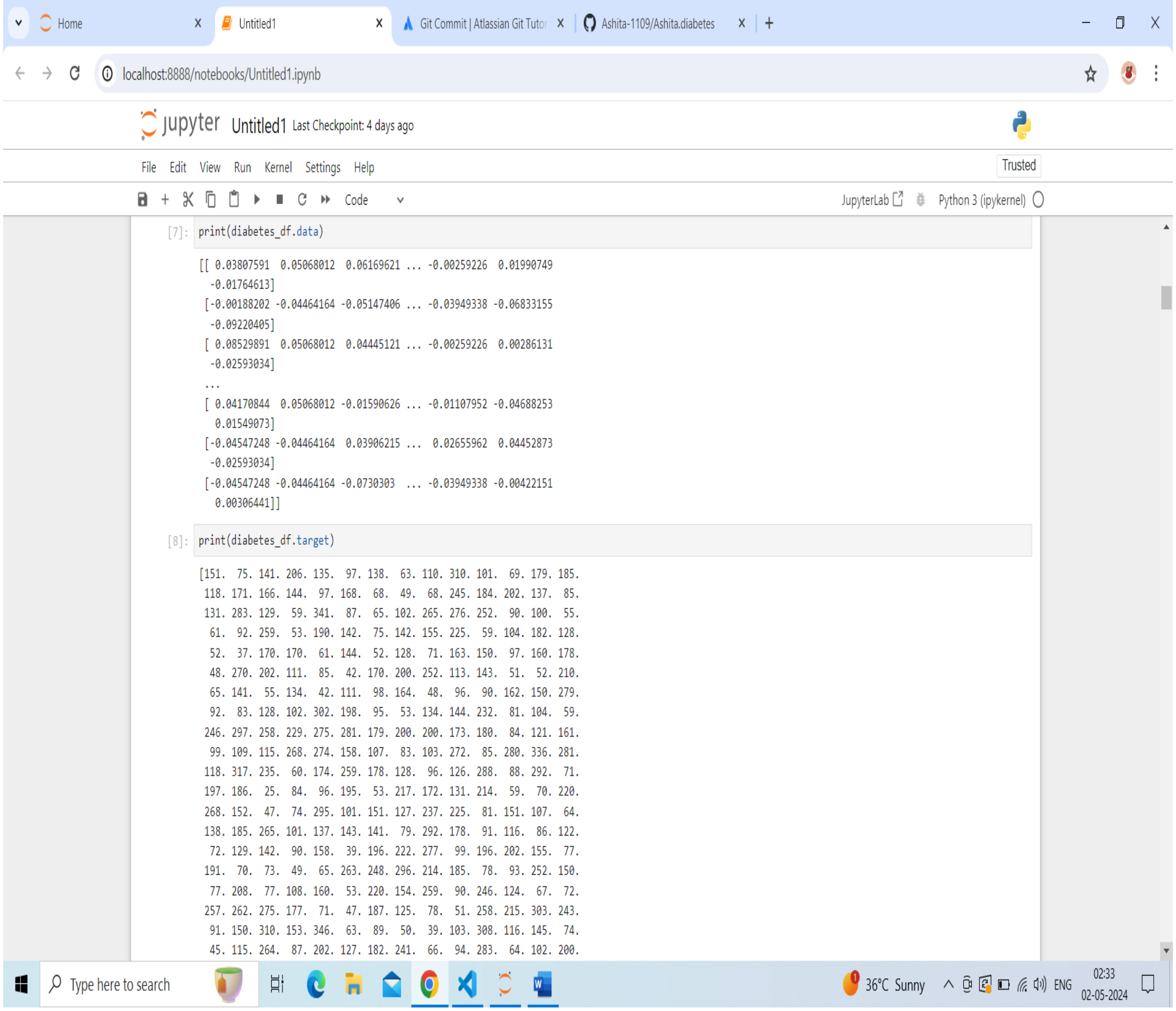
3.3 OPERATION ON TRAINING DATA

3.3.1 HEAD OPERATION

Head Operation is used to show the topmost rows of the data according to the value entered by the user. The default value of head is 5.

3.1.1 TAIL OPERATION

Tail Operation is used to display the bottommost rows of the data according to the value entered by the user. The default value for tail is 5.


Home
Untitled1
Git Commit | Atlassian Git Tutor
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
36°C Sunny
ENG
02:33
02-05-2024

```
[7]: print(diabetes_df.data)

[[ 0.03807591  0.05068012  0.06169621 ... -0.00259226  0.01990749
  -0.01764613]
 [-0.00188202 -0.04464164 -0.05147406 ... -0.03949338 -0.06833155
  -0.09220405]
 [ 0.08529891  0.05068012  0.04445121 ... -0.00259226  0.00286131
  -0.02593034]
 ...
 [ 0.04170844  0.05068012 -0.01590626 ... -0.01107952 -0.04688253
   0.01549073]
 [-0.04547248 -0.04464164  0.03906215 ...  0.02655962  0.04452873
  -0.02593034]
 [-0.04547248 -0.04464164 -0.0730303  ... -0.03949338 -0.00422151
   0.00306441]]
```

```
[8]: print(diabetes_df.target)

[151.  75. 141. 206. 135.  97. 138.  63. 110. 310. 101.  69. 179. 185.
 118. 171. 166. 144.  97. 168.  68.  49.  68. 245. 184. 202. 137.  85.
 131. 283. 129.  59. 341.  87.  65. 102. 265. 276. 252.  90. 100.  55.
  61.  92. 259.  53. 190. 142.  75. 142. 155. 225.  59. 104. 182. 128.
  52.  37. 170. 170.  61. 144.  52. 128.  71. 163. 150.  97. 160. 178.
  48. 270. 202. 111.  85.  42. 170. 200. 252. 113. 143.  51.  52. 210.
  65. 141.  55. 134.  42. 111.  98. 164.  48.  96.  90. 162. 150. 279.
  92.  83. 128. 102. 302. 198.  95.  53. 134. 144. 232.  81. 104.  59.
 246. 297. 258. 229. 275. 281. 179. 200. 200. 173. 180.  84. 121. 161.
  99. 109. 115. 268. 274. 158. 107.  83. 103. 272.  85. 280. 336. 281.
 118. 317. 235.  60. 174. 259. 178. 128.  96. 126. 288.  88. 292.  71.
 197. 186.  25.  84.  96. 195.  53. 217. 172. 131. 214.  59.  70. 220.
 268. 152.  47.  74. 295. 101. 151. 127. 237. 225.  81. 151. 107.  64.
 138. 185. 265. 101. 137. 143. 141.  79. 292. 178.  91. 116.  86. 122.
  72. 129. 142.  90. 158.  39. 196. 222. 277.  99. 196. 202. 155.  77.
 191.  70.  73.  49.  65. 263. 248. 296. 214. 185.  78.  93. 252. 150.
  77. 208.  77. 108. 160.  53. 220. 154. 259.  90. 246. 124.  67.  72.
 257. 262. 275. 177.  71.  47. 187. 125.  78.  51. 258. 215. 303. 243.
  91. 150. 310. 153. 346.  63.  89.  50.  39. 103. 308. 116. 145.  74.
  45. 115. 264.  87. 202. 127. 182. 241.  66.  94. 283.  64. 102. 200.
```

3.1.2 SHAPE OPERATION

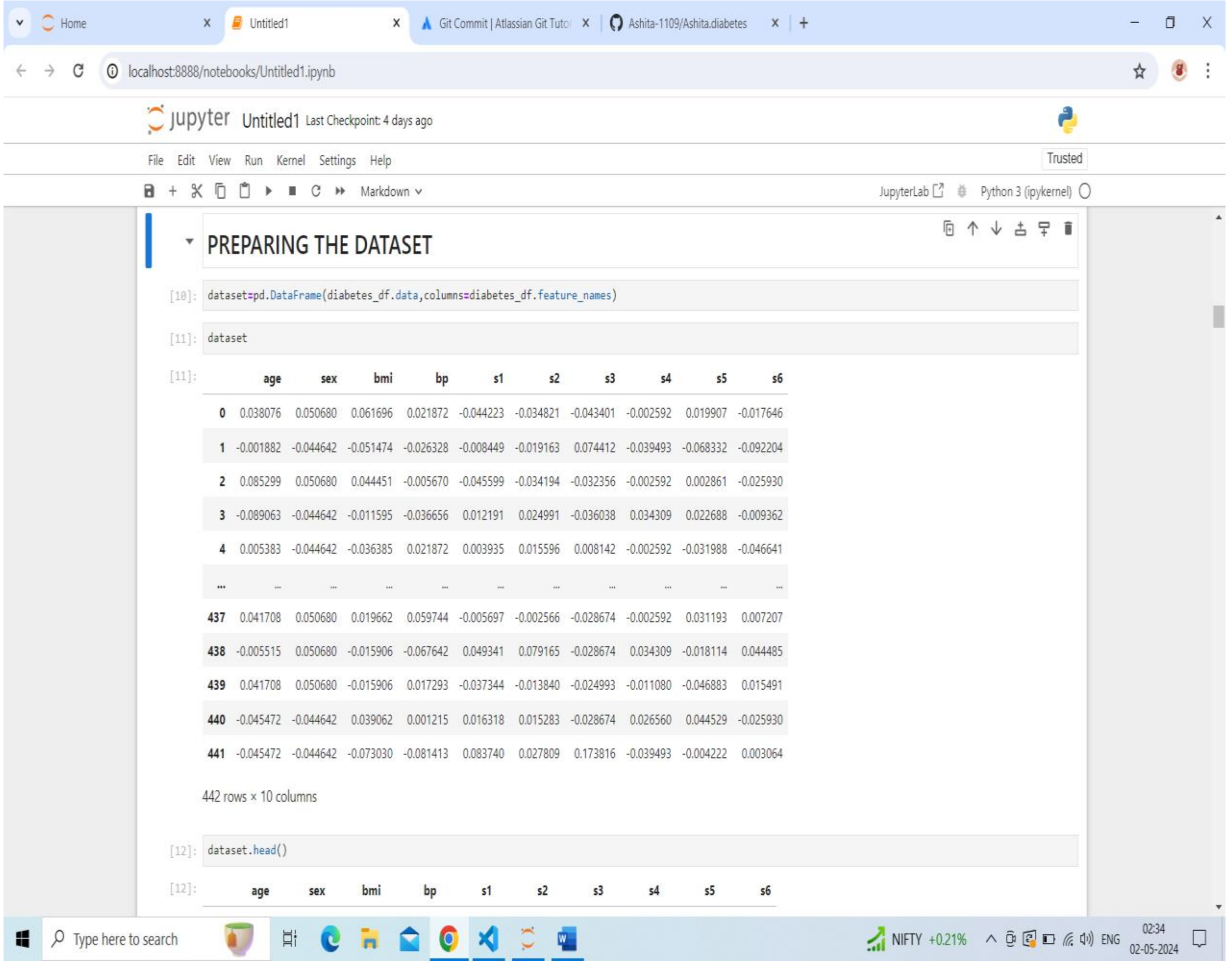
Shape Operation is used to find out the numbers of rows and column in the data.

3.1.3 DESCRIBE OPERATION

Describe Operation helps in finding the standard value for each of the element of operation like mean, mode, minimum, maximum, standard deviation and many more.

3.1.4 INFO OPERATION

Info Operation helps the user to find out all the data types with the number of not null values for each and every row and column along with the size of datatypes.


Home
Untitled1
Git Commit | Atlassian Git Tutor
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)
Type here to search
NIFTY +0.21%
ENG
02:34
02-05-2024

PREPARING THE DATASET

```
[10]: dataset=pd.DataFrame(diabetes_df.data,columns=diabetes_df.feature_names)
```

```
[11]: dataset
```

[11]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019907	-0.017646
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068332	-0.092204
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002861	-0.025930
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022688	-0.009362
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031988	-0.046641
...	...	...	...	...	...	...	...	...	...	...
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.031193	0.007207
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.018114	0.044485
439	0.041708	0.050680	-0.015906	0.017293	-0.037344	-0.013840	-0.024993	-0.011080	-0.046883	0.015491
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.044529	-0.025930
441	-0.045472	-0.044642	-0.073030	-0.081413	0.083740	0.027809	0.173816	-0.039493	-0.004222	0.003064

442 rows × 10 columns

```
[12]: dataset.head()
```

[12]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6

3.2 MANIPULATION OF DATA

3.2.1 IS NULL FUNCTION

The is null method return all the element having their values are replaced with True or False.

3.2.2 DROPNA FUNCTION

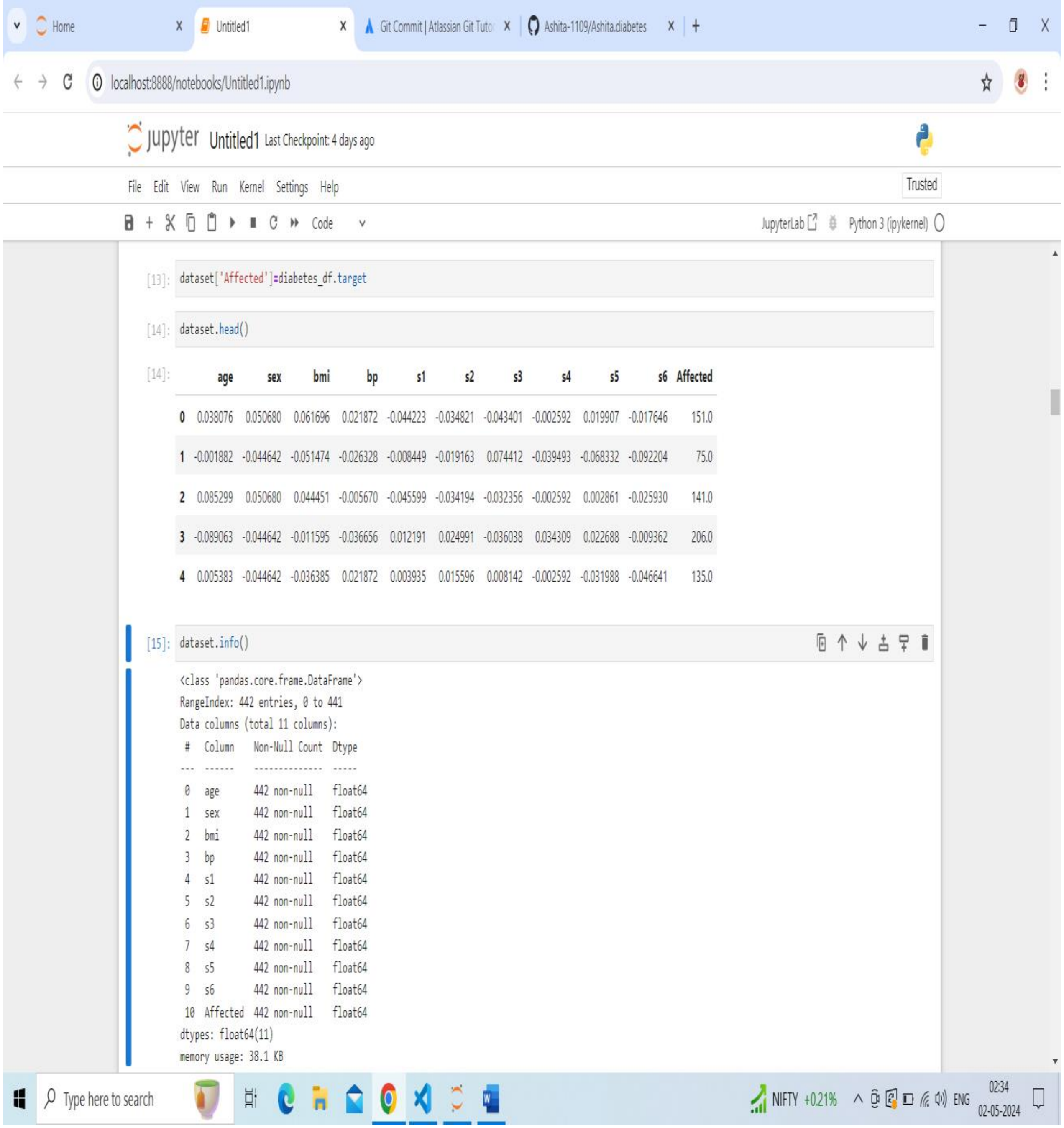
Dropna Function removes the cells which contains the null values.

3.2.3 INFO FUNCTION

Info Operation is used to determine the type of data and its size of each and every element of data.

3.2.4 COLUMN FUNCTION

Column function is used to find the number and names of the data.


Home
Untitled1
Git Commit | Atlassian Git Tuto
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
NIFTY +0.21%
ENG
02:34
02-05-2024

```
[13]: dataset['Affected']=diabetes_df.target
```

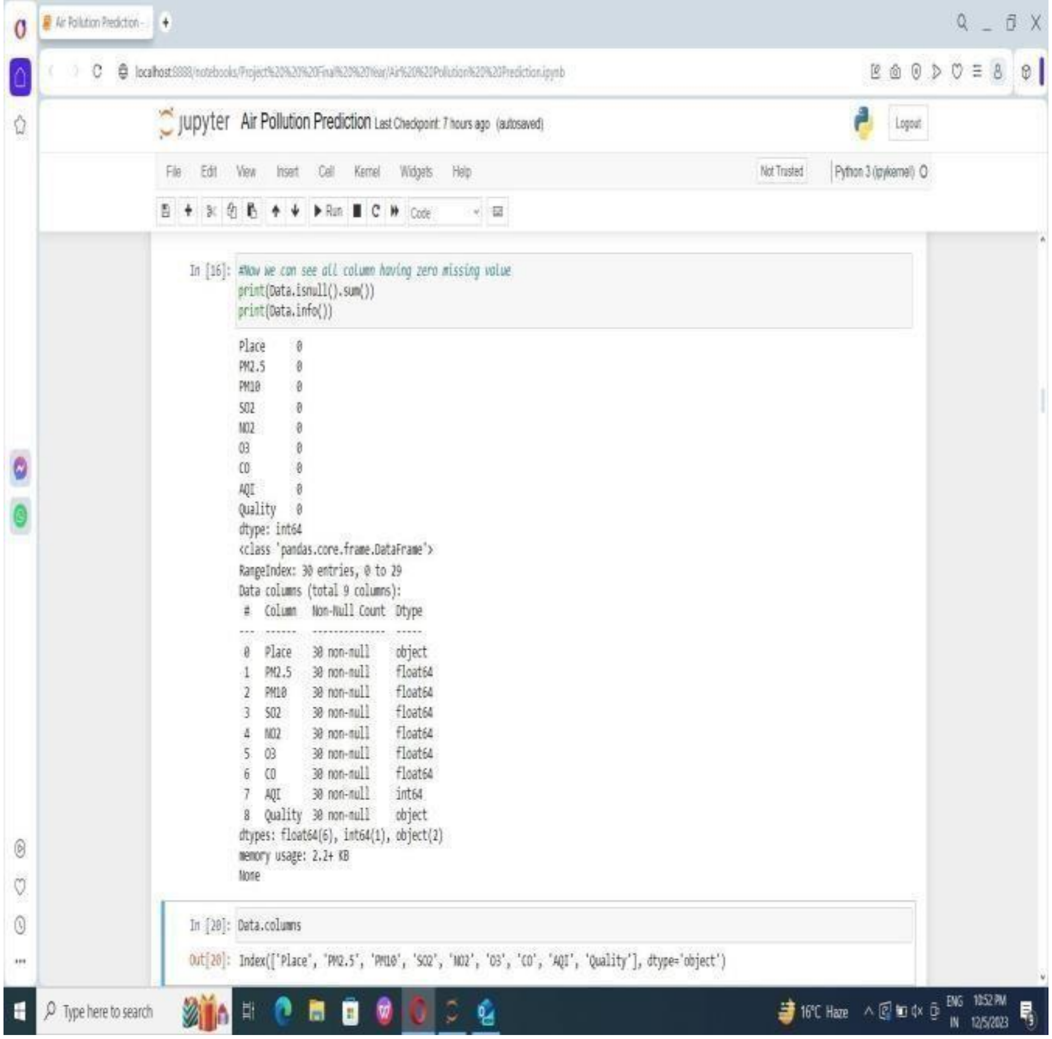
```
[14]: dataset.head()
```

[14]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	Affected
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019907	-0.017646	151.0
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068332	-0.092204	75.0
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002861	-0.025930	141.0
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022688	-0.009362	206.0
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031988	-0.046641	135.0

```
[15]: dataset.info()
```

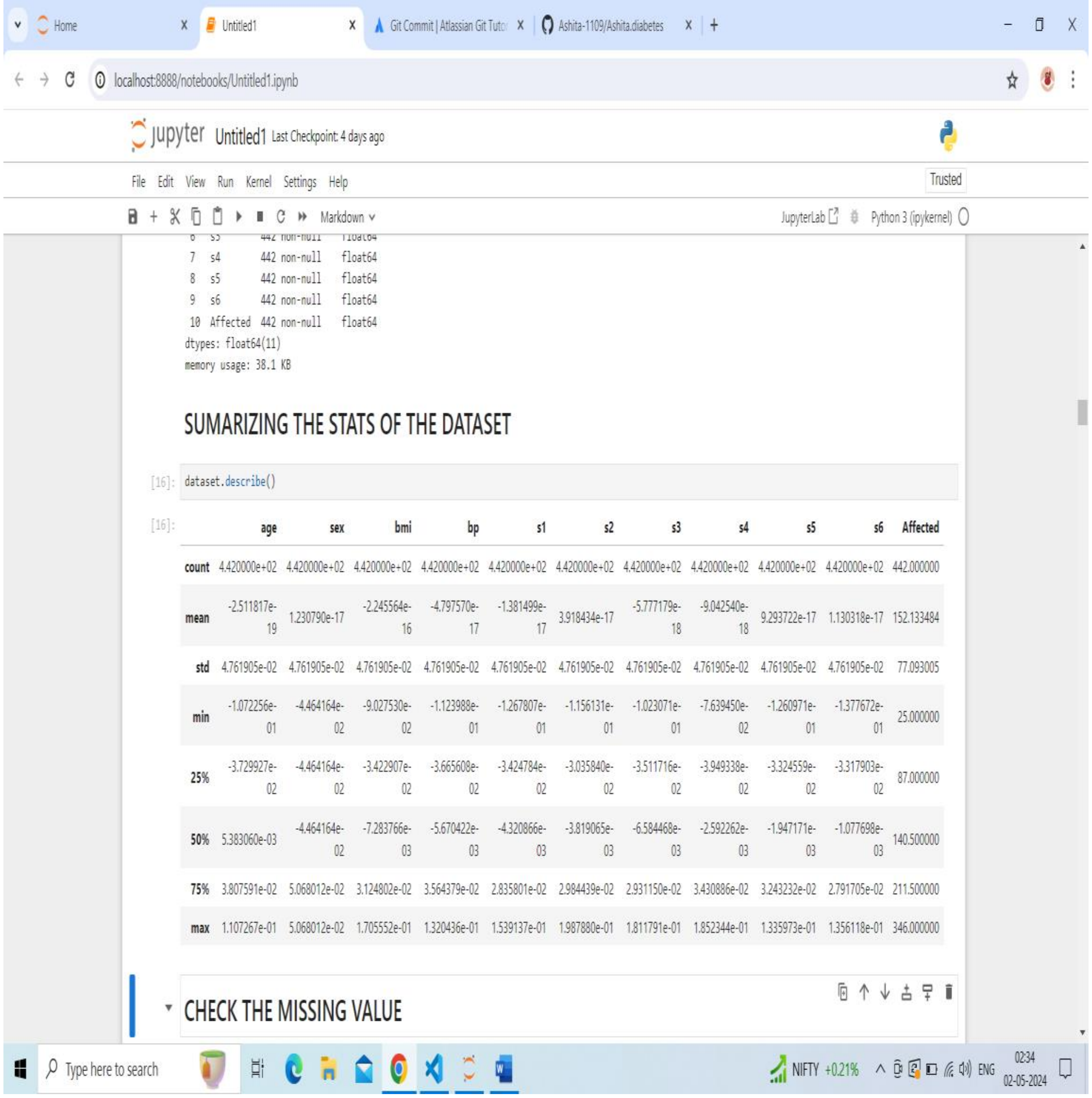
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 442 entries, 0 to 441
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       442 non-null    float64
 1   sex       442 non-null    float64
 2   bmi       442 non-null    float64
 3   bp        442 non-null    float64
 4   s1        442 non-null    float64
 5   s2        442 non-null    float64
 6   s3        442 non-null    float64
 7   s4        442 non-null    float64
 8   s5        442 non-null    float64
 9   s6        442 non-null    float64
 10  Affected  442 non-null    float64
dtypes: float64(11)
memory usage: 38.1 KB
```


Air Pollution Prediction
localhost:8888/notebooks/Project%20%20%20Final%20%20Year/Air%20%20Pollution%20%20Prediction.ipynb
jupyter Air Pollution Prediction Last Checkpoint: 7 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
16°C Haze
ENG IN
10:52 PM
12/5/2023

```
In [16]: #Now we can see all column having zero missing value
         print(Data.isnull().sum())
         print(Data.info())

         Place      0
         PM2.5      0
         PM10       0
         SO2        0
         NO2        0
         O3         0
         CO         0
         AQI        0
         Quality    0
         dtype: int64
         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 30 entries, 0 to 29
         Data columns (total 9 columns):
          #   Column   Non-Null Count  Dtype
         ---  ------   --------------  -----
          0   Place    30 non-null     object
          1   PM2.5    30 non-null     float64
          2   PM10     30 non-null     float64
          3   SO2      30 non-null     float64
          4   NO2      30 non-null     float64
          5   O3       30 non-null     float64
          6   CO       30 non-null     float64
          7   AQI      30 non-null     int64
          8   Quality  30 non-null     object
         dtypes: float64(6), int64(1), object(2)
         memory usage: 2.2+ KB
         None

In [20]: Data.columns

Out[20]: Index(['Place', 'PM2.5', 'PM10', 'SO2', 'NO2', 'O3', 'CO', 'AQI', 'Quality'], dtype='object')
```

3.2.5 VALUE COUNT FUNCTION

Value count function return the series containing the numbers and value of the unique data of the training dataset.


Home
Untitled1
Git Commit | Atlassian Git Tuto
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)
Type here to search
NIFTY +0.21%
ENG
02:34
02-05-2024

```
 6  s3        442 non-null   float64
 7  s4        442 non-null   float64
 8  s5        442 non-null   float64
 9  s6        442 non-null   float64
 10 Affected  442 non-null   float64
dtypes: float64(11)
memory usage: 38.1 KB
```

SUMARIZING THE STATS OF THE DATASET

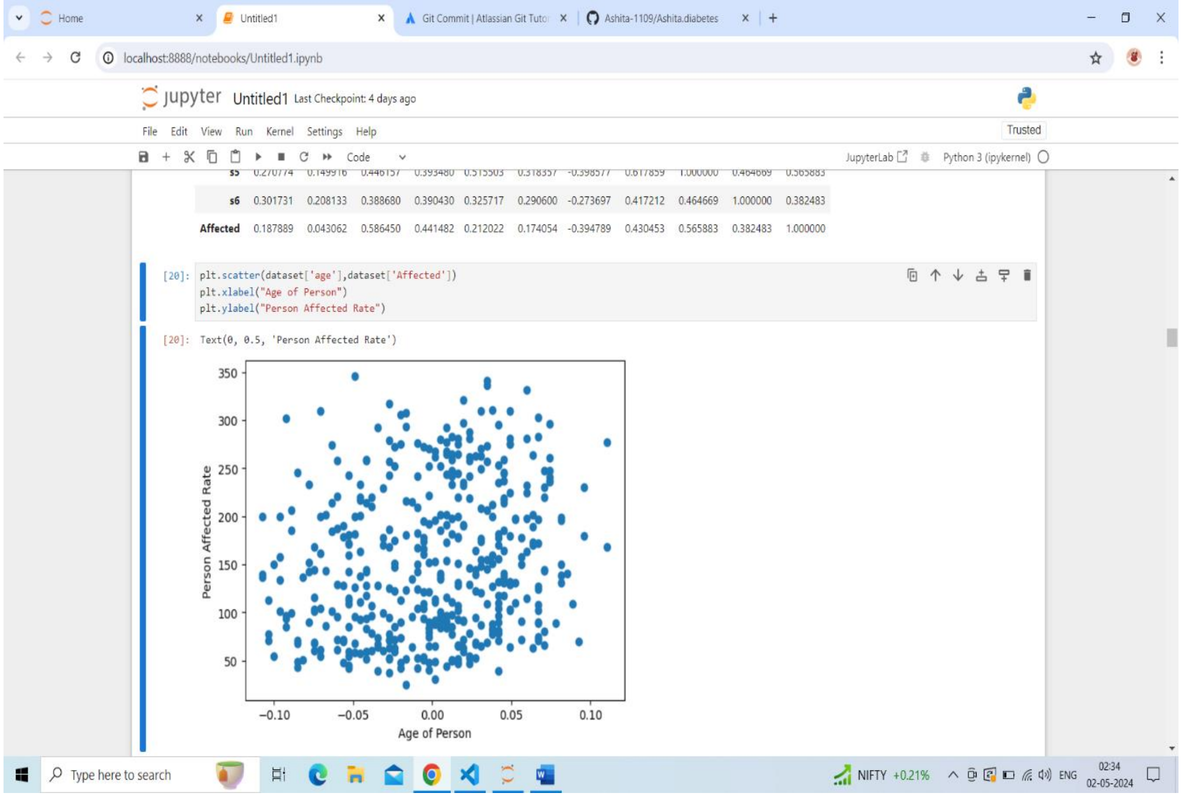
```
[16]: dataset.describe()
```

[16]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	Affected
count	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	442.000000
mean	-2.511817e-19	1.230790e-17	-2.245564e-16	-4.797570e-17	-1.381499e-17	3.918434e-17	-5.777179e-18	-9.042540e-18	9.293722e-17	1.130318e-17	152.133484
std	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	77.093005
min	-1.072256e-01	-4.464164e-02	-9.027530e-02	-1.123988e-01	-1.267807e-01	-1.156131e-01	-1.023071e-01	-7.639450e-02	-1.260971e-01	-1.377672e-01	25.000000
25%	-3.729927e-02	-4.464164e-02	-3.422907e-02	-3.665608e-02	-3.424784e-02	-3.035840e-02	-3.511716e-02	-3.949338e-02	-3.324559e-02	-3.317903e-02	87.000000
50%	5.383060e-03	-4.464164e-02	-7.283766e-03	-5.670422e-03	-4.320866e-03	-3.819065e-03	-6.584468e-03	-2.592262e-03	-1.947171e-03	-1.077698e-03	140.500000
75%	3.807591e-02	5.068012e-02	3.124802e-02	3.564379e-02	2.835801e-02	2.984439e-02	2.931150e-02	3.430886e-02	3.243232e-02	2.791705e-02	211.500000
max	1.107267e-01	5.068012e-02	1.705552e-01	1.320436e-01	1.539137e-01	1.987880e-01	1.811791e-01	1.852344e-01	1.335973e-01	1.356118e-01	346.000000

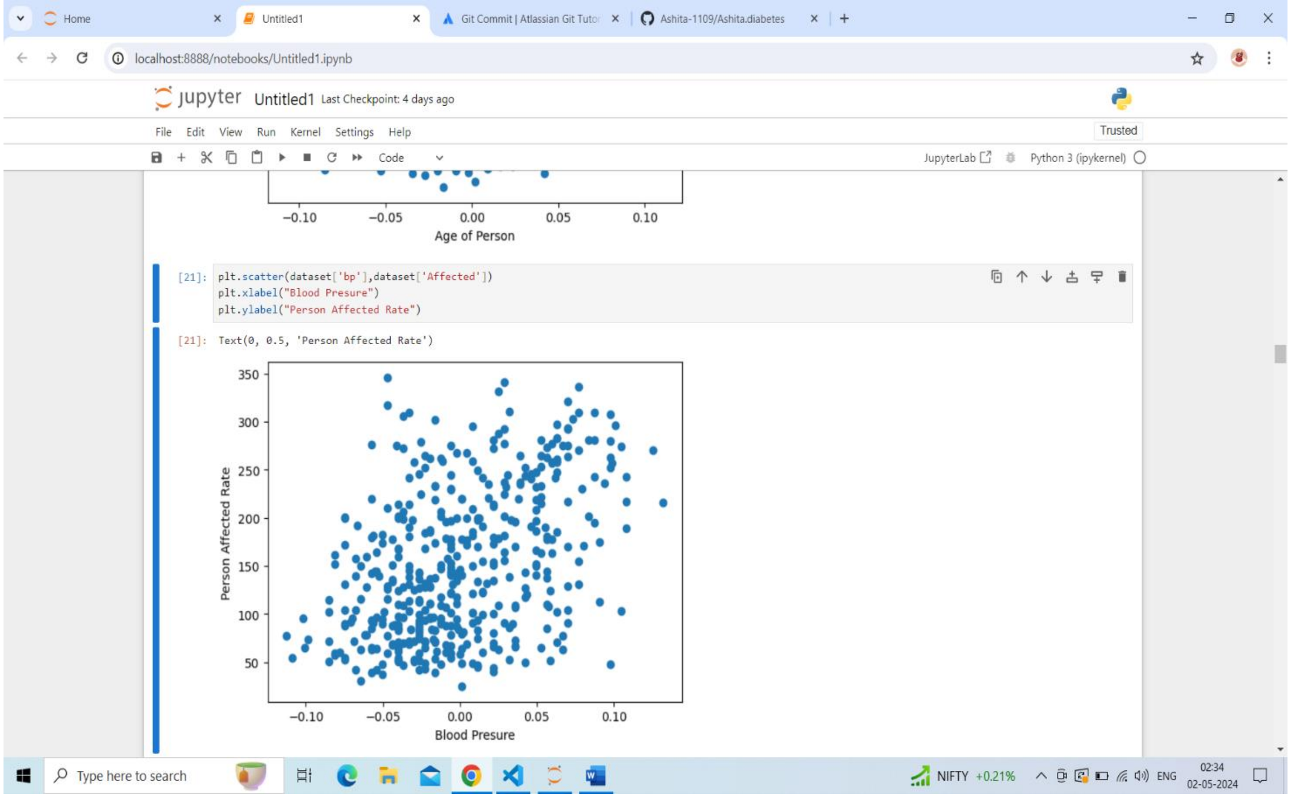
CHECK THE MISSING VALUE

3.3 REPRESENTATION OF DATA

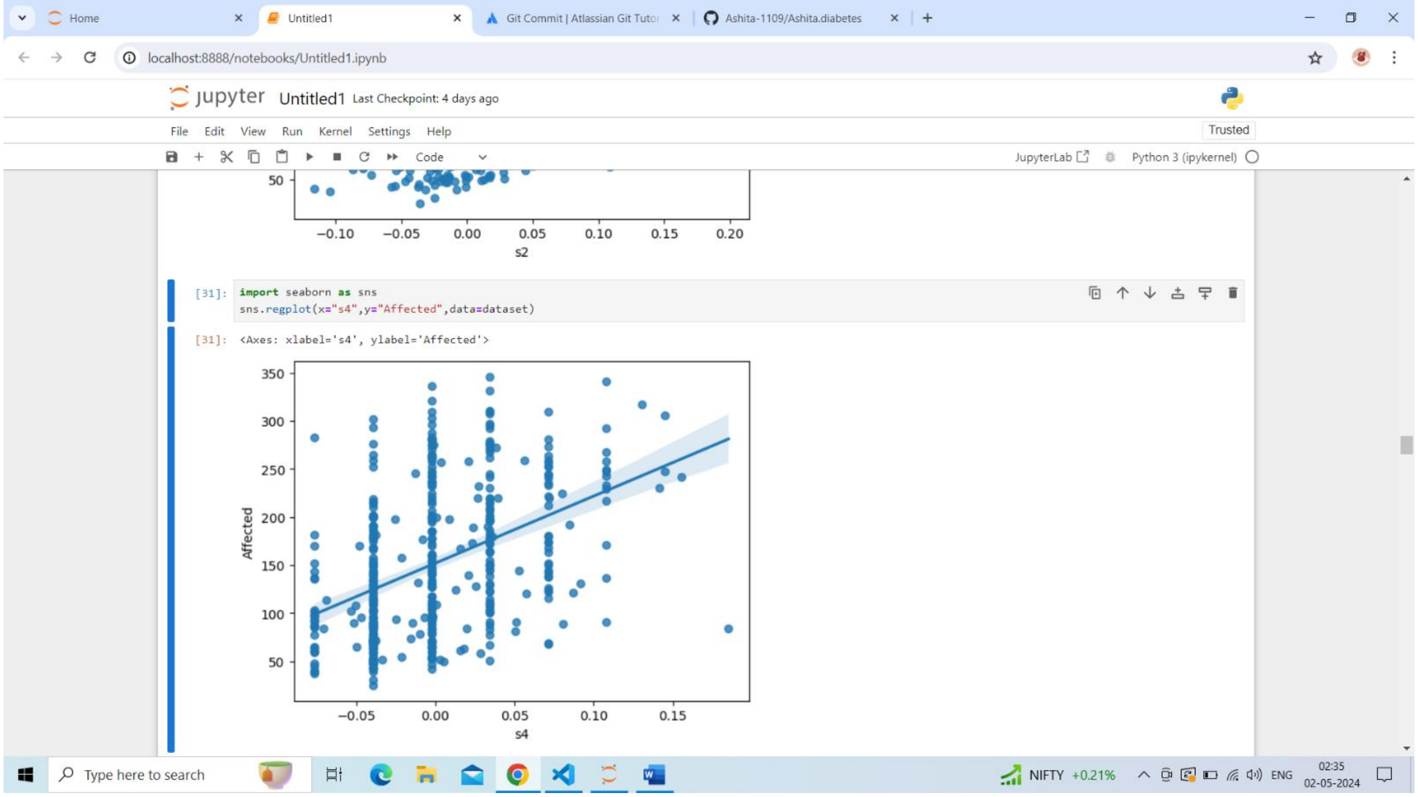

Home
Untitled1
Git Commit | Atlassian Git Tutor
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
350
300
250
200
150
100
50
Person Affected Rate
−0.10
−0.05
0.00
0.05
0.10
Age of Person
Type here to search
NIFTY +0.21%
ENG
02:34
02-05-2024

s5	0.270774	0.149916	0.446157	0.393480	0.515503	0.318357	-0.398577	0.617859	1.000000	0.464669	0.565883
s6	0.301731	0.208133	0.388680	0.390430	0.325717	0.290600	-0.273697	0.417212	0.464669	1.000000	0.382483
Affected	0.187889	0.043062	0.586450	0.441482	0.212022	0.174054	-0.394789	0.430453	0.565883	0.382483	1.000000

```
[20]: plt.scatter(dataset['age'],dataset['Affected'])
      plt.xlabel("Age of Person")
      plt.ylabel("Person Affected Rate")

[20]: Text(0, 0.5, 'Person Affected Rate')
```

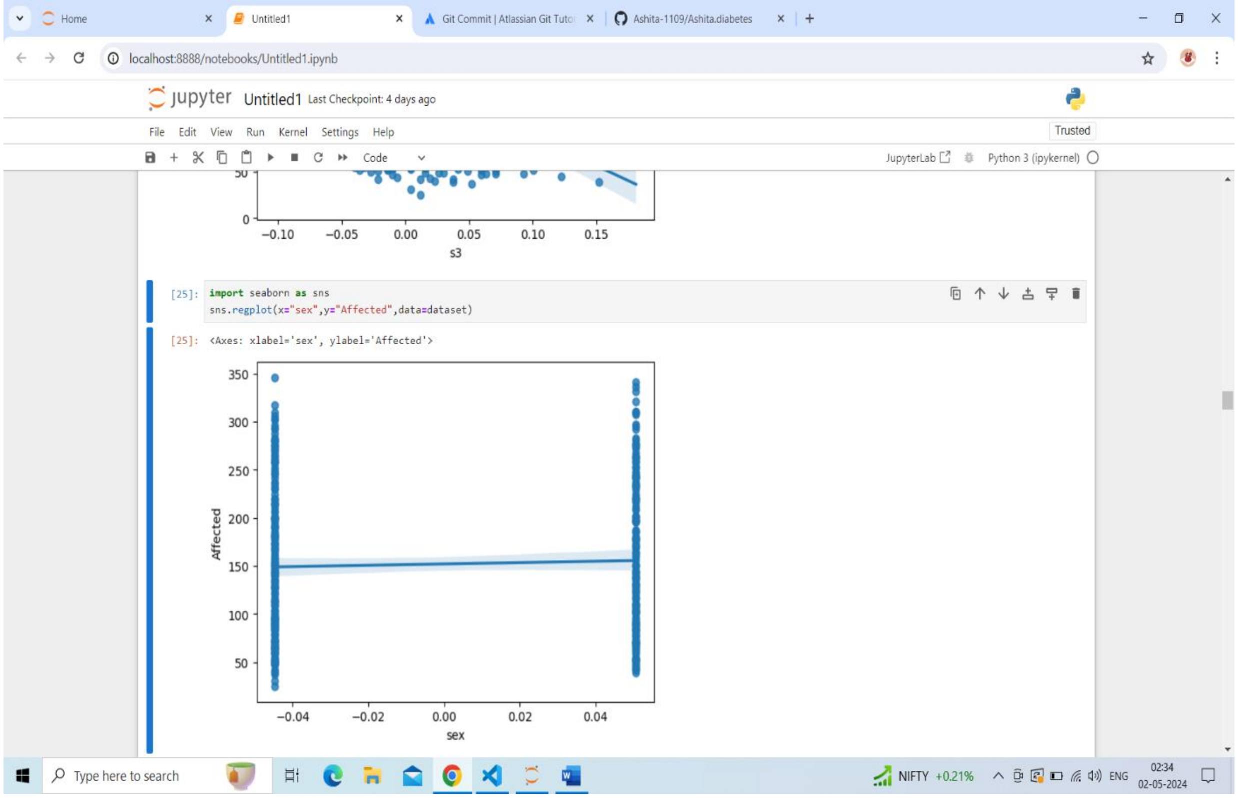


```
[21]: plt.scatter(dataset['bp'],dataset['Affected'])
      plt.xlabel("Blood Presure")
      plt.ylabel("Person Affected Rate")

[21]: Text(0, 0.5, 'Person Affected Rate')
```



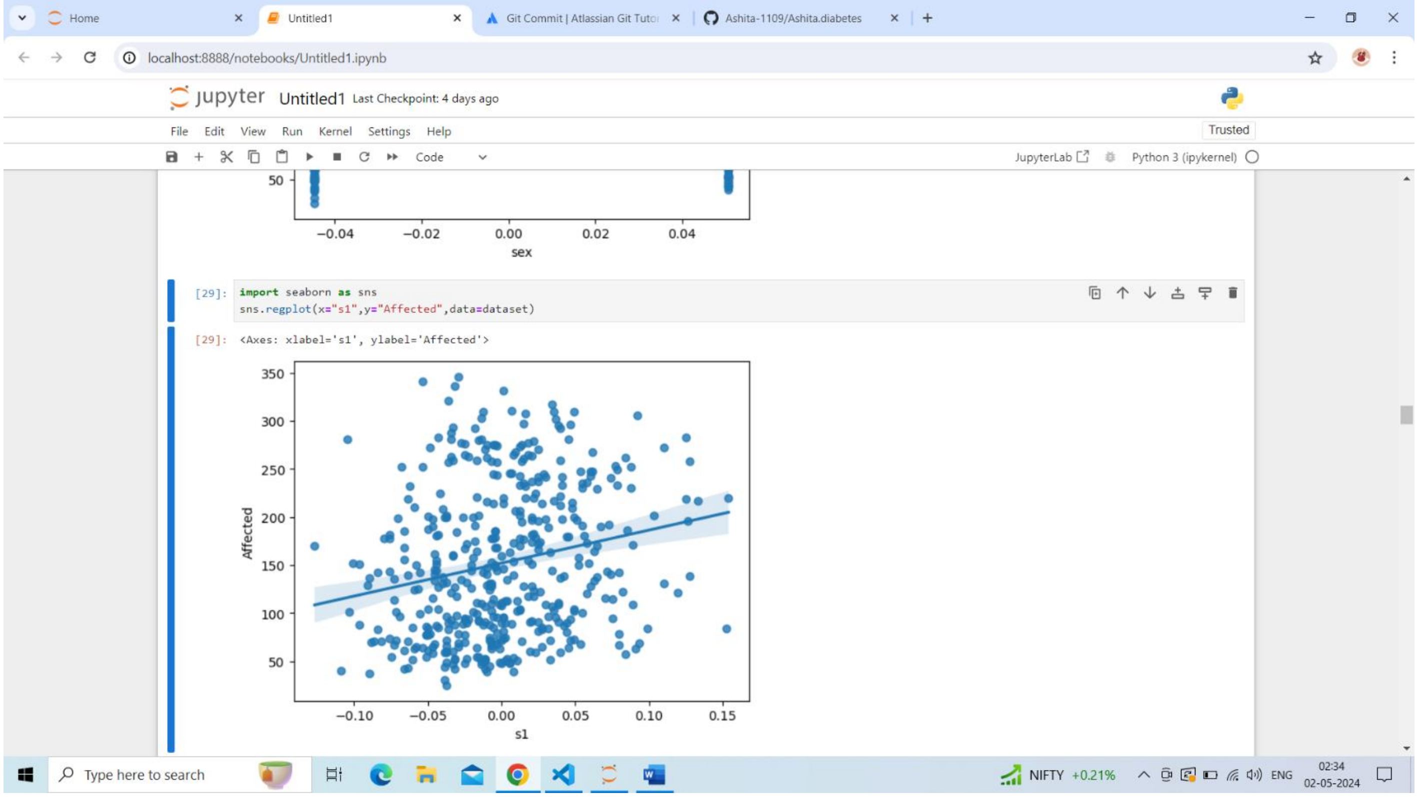
```
[31]: import seaborn as sns
      sns.regplot(x="s4",y="Affected",data=dataset)
```

```
[31]: <Axes: xlabel='s4', ylabel='Affected'>
```



```
[25]: import seaborn as sns
      sns.regplot(x="sex",y="Affected",data=dataset)
```

```
[25]: <Axes: xlabel='sex', ylabel='Affected'>
```

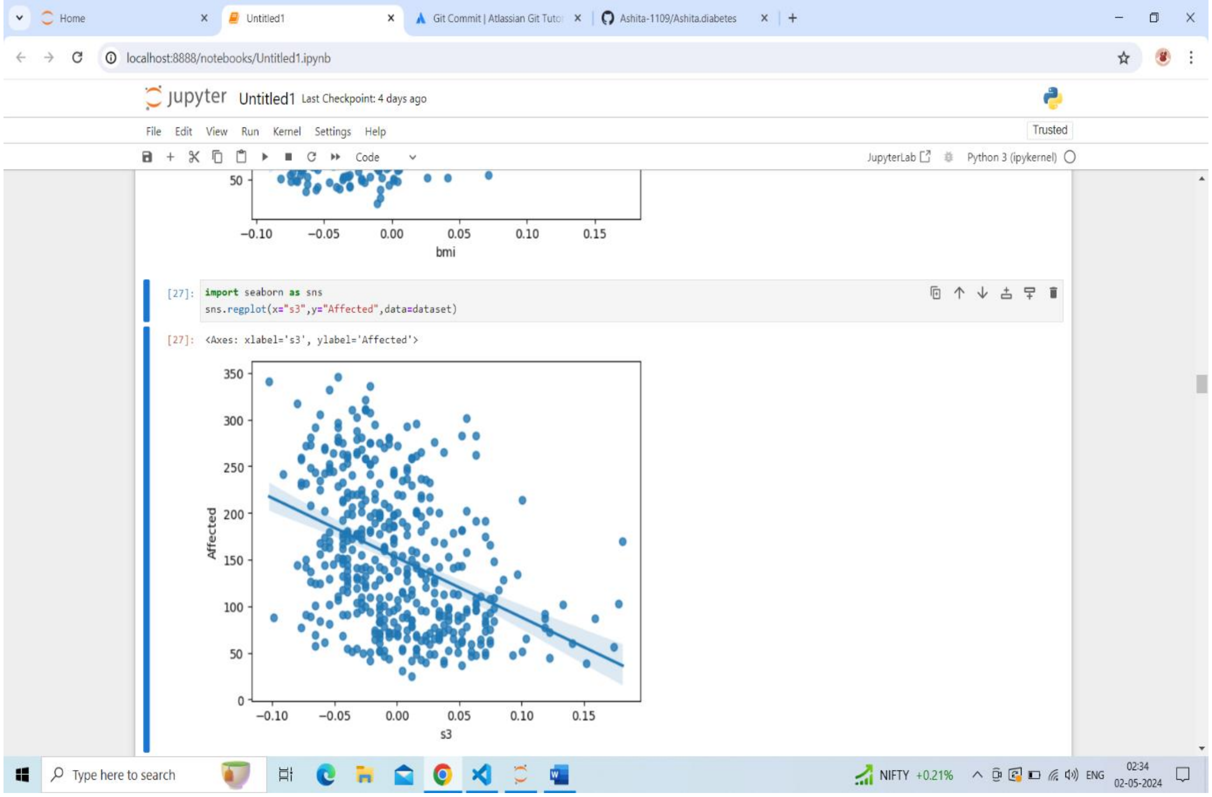


```
[29]: import seaborn as sns
      sns.regplot(x="s1",y="Affected",data=dataset)
```

```
[29]: <Axes: xlabel='s1', ylabel='Affected'>
```

3.3.1 BOX PLOT REPRESENTATION

Box plot is used to represent the relation between the data and the differences between the expected and actual value.

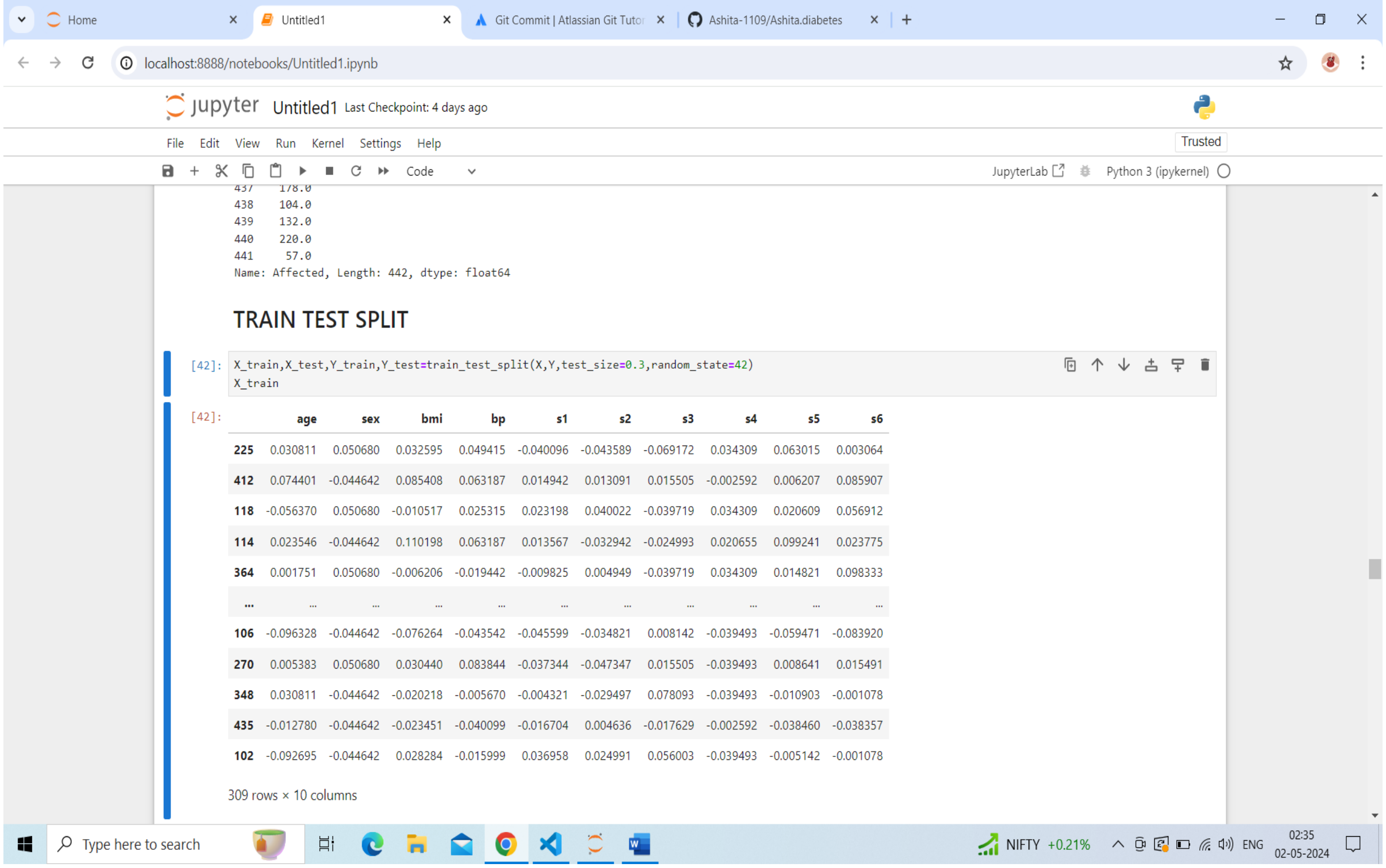


```
[27]: import seaborn as sns
      sns.regplot(x="s3",y="Affected",data=dataset)
```

```
[27]: <Axes: xlabel='s3', ylabel='Affected'>
```

3.3.2 CORRELATION

Correlation gives the direction of the linear line between the quantitative variables



Home
Untitled1
Git Commit | Atlassian Git Tutor
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
NIFTY +0.21%
ENG
02:35
02-05-2024

```
437     178.0
438     104.0
439     132.0
440     220.0
441      57.0
Name: Affected, Length: 442, dtype: float64
```

TRAIN TEST SPLIT

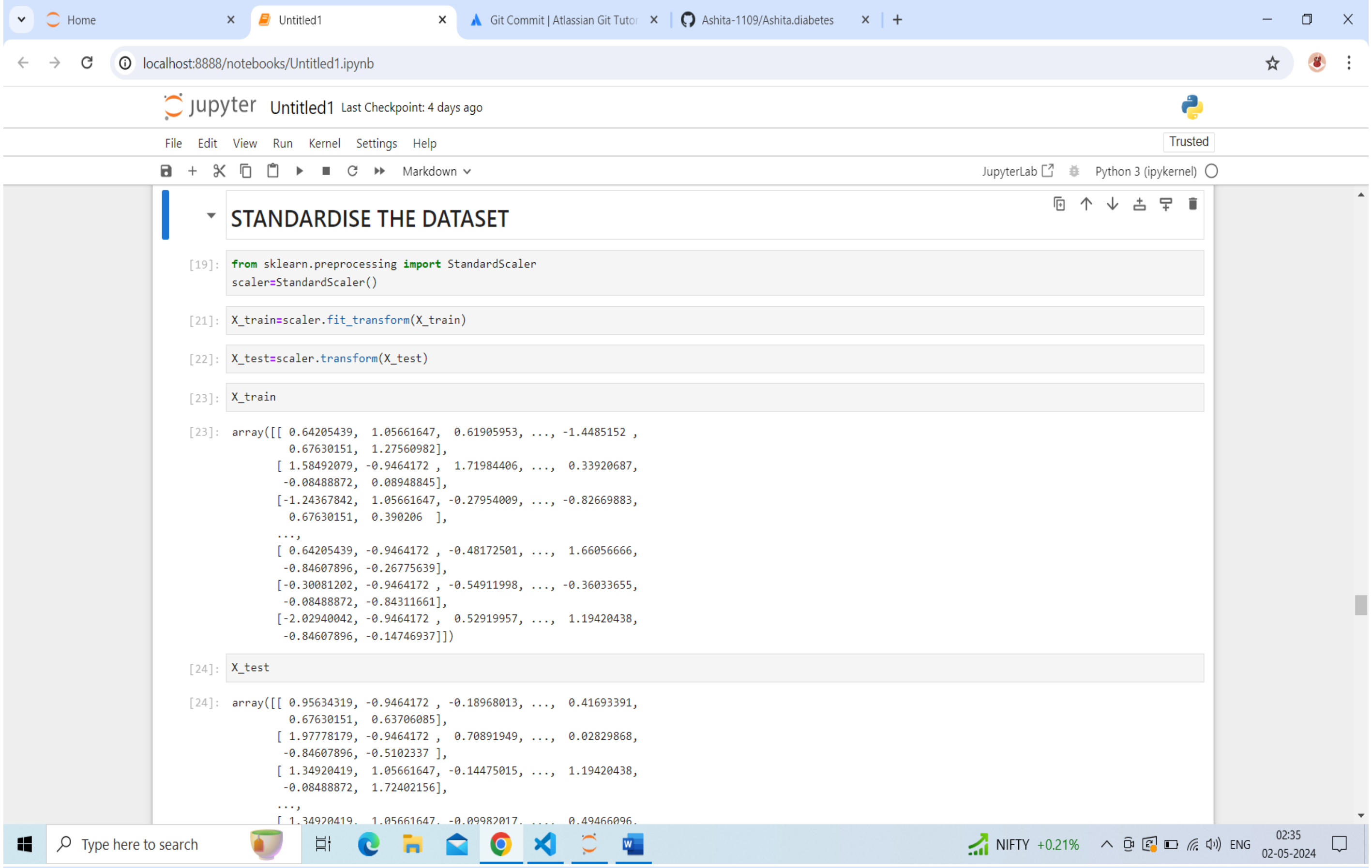
[42]:

```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3,random_state=42)
X_train
```

[42]:

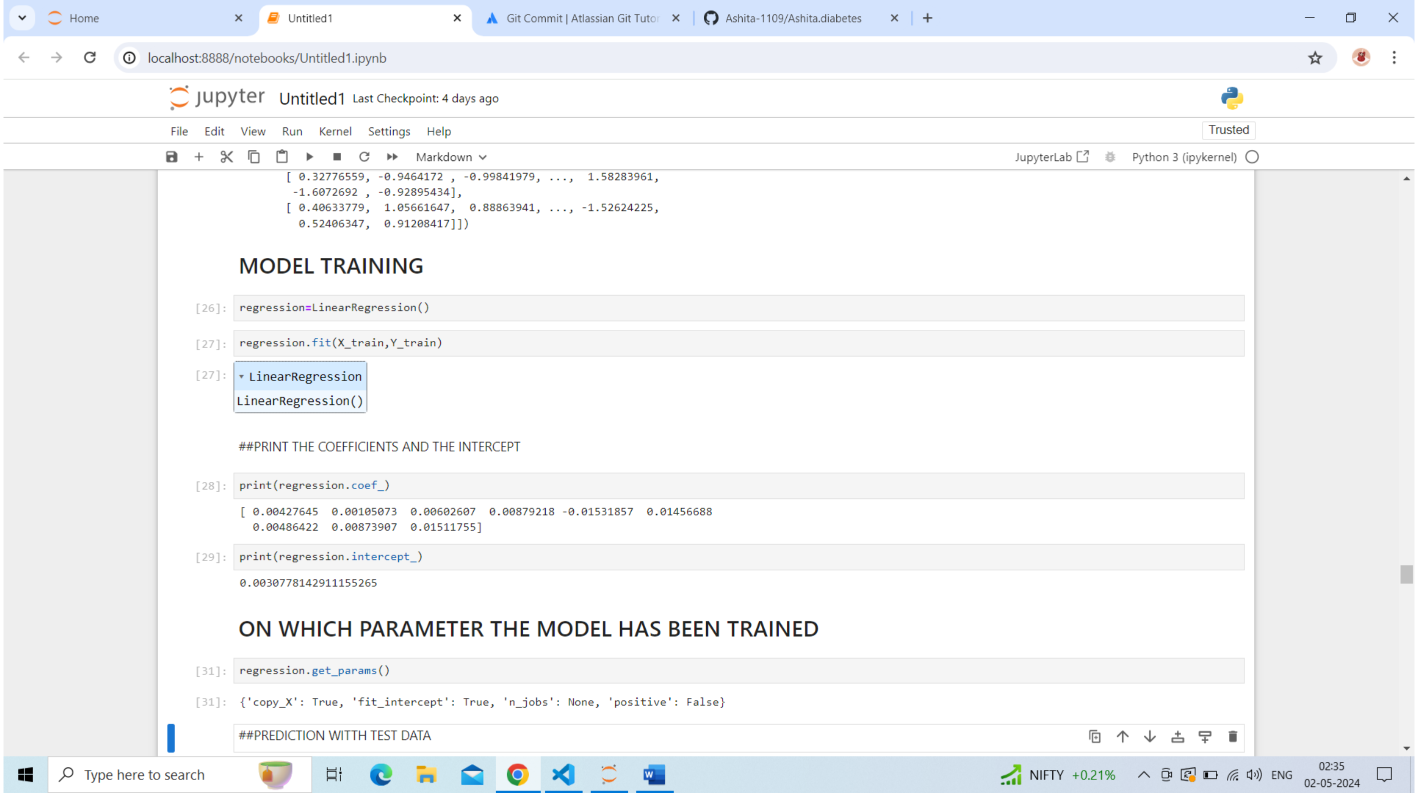
	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6
225	0.030811	0.050680	0.032595	0.049415	-0.040096	-0.043589	-0.069172	0.034309	0.063015	0.003064
412	0.074401	-0.044642	0.085408	0.063187	0.014942	0.013091	0.015505	-0.002592	0.006207	0.085907
118	-0.056370	0.050680	-0.010517	0.025315	0.023198	0.040022	-0.039719	0.034309	0.020609	0.056912
114	0.023546	-0.044642	0.110198	0.063187	0.013567	-0.032942	-0.024993	0.020655	0.099241	0.023775
364	0.001751	0.050680	-0.006206	-0.019442	-0.009825	0.004949	-0.039719	0.034309	0.014821	0.098333
...	...	...	...	...	...	...	...	...	...	...
106	-0.096328	-0.044642	-0.076264	-0.043542	-0.045599	-0.034821	0.008142	-0.039493	-0.059471	-0.083920
270	0.005383	0.050680	0.030440	0.083844	-0.037344	-0.047347	0.015505	-0.039493	0.008641	0.015491
348	0.030811	-0.044642	-0.020218	-0.005670	-0.004321	-0.029497	0.078093	-0.039493	-0.010903	-0.001078
435	-0.012780	-0.044642	-0.023451	-0.040099	-0.016704	0.004636	-0.017629	-0.002592	-0.038460	-0.038357
102	-0.092695	-0.044642	0.028284	-0.015999	0.036958	0.024991	0.056003	-0.039493	-0.005142	-0.001078

309 rows × 10 columns


Home
Untitled1
Git Commit | Atlassian Git Tutor
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
Trusted
File Edit View Run Kernel Settings Help
Markdown
JupyterLab
Python 3 (ipykernel)
Type here to search
NIFTY +0.21%
ENG
02:35
02-05-2024

STANDARDISE THE DATASET

```
[19]: from sklearn.preprocessing import StandardScaler
      scaler=StandardScaler()

[21]: X_train=scaler.fit_transform(X_train)

[22]: X_test=scaler.transform(X_test)

[23]: X_train

[23]: array([[ 0.64205439,  1.05661647,  0.61905953, ..., -1.4485152 ,
          0.67630151,  1.27560982],
        [ 1.58492079, -0.9464172 ,  1.71984406, ...,  0.33920687,
         -0.08488872,  0.08948845],
        [-1.24367842,  1.05661647, -0.27954009, ..., -0.82669883,
          0.67630151,  0.390206  ],
        ...,
        [ 0.64205439, -0.9464172 , -0.48172501, ...,  1.66056666,
         -0.84607896, -0.26775639],
        [-0.30081202, -0.9464172 , -0.54911998, ..., -0.36033655,
         -0.08488872, -0.84311661],
        [-2.02940042, -0.9464172 ,  0.52919957, ...,  1.19420438,
         -0.84607896, -0.14746937]])

[24]: X_test

[24]: array([[ 0.95634319, -0.9464172 , -0.18968013, ...,  0.41693391,
          0.67630151,  0.63706085],
        [ 1.97778179, -0.9464172 ,  0.70891949, ...,  0.02829868,
         -0.84607896, -0.5102337 ],
        [ 1.34920419,  1.05661647, -0.14475015, ...,  1.19420438,
         -0.08488872,  1.72402156],
        ...,
        [ 1.34920419,  1.05661647, -0.09982017, ...,  0.49466096,
```


Home
Untitled1
Git Commit | Atlassian Git Tutor
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
Trusted
File Edit View Run Kernel Settings Help
Markdown
JupyterLab
Python 3 (ipykernel)
Type here to search
NIFTY +0.21%
ENG
02:35
02-05-2024

```
        [ 0.32776559, -0.9464172 , -0.99841979, ...,  1.58283961,
         -1.6072692 , -0.92895434],
        [ 0.40633779,  1.05661647,  0.88863941, ..., -1.52624225,
          0.52406347,  0.91208417]])
```

MODEL TRAINING

```
[26]: regression=LinearRegression()

[27]: regression.fit(X_train,Y_train)

[27]: ▾ LinearRegression
      LinearRegression()
```

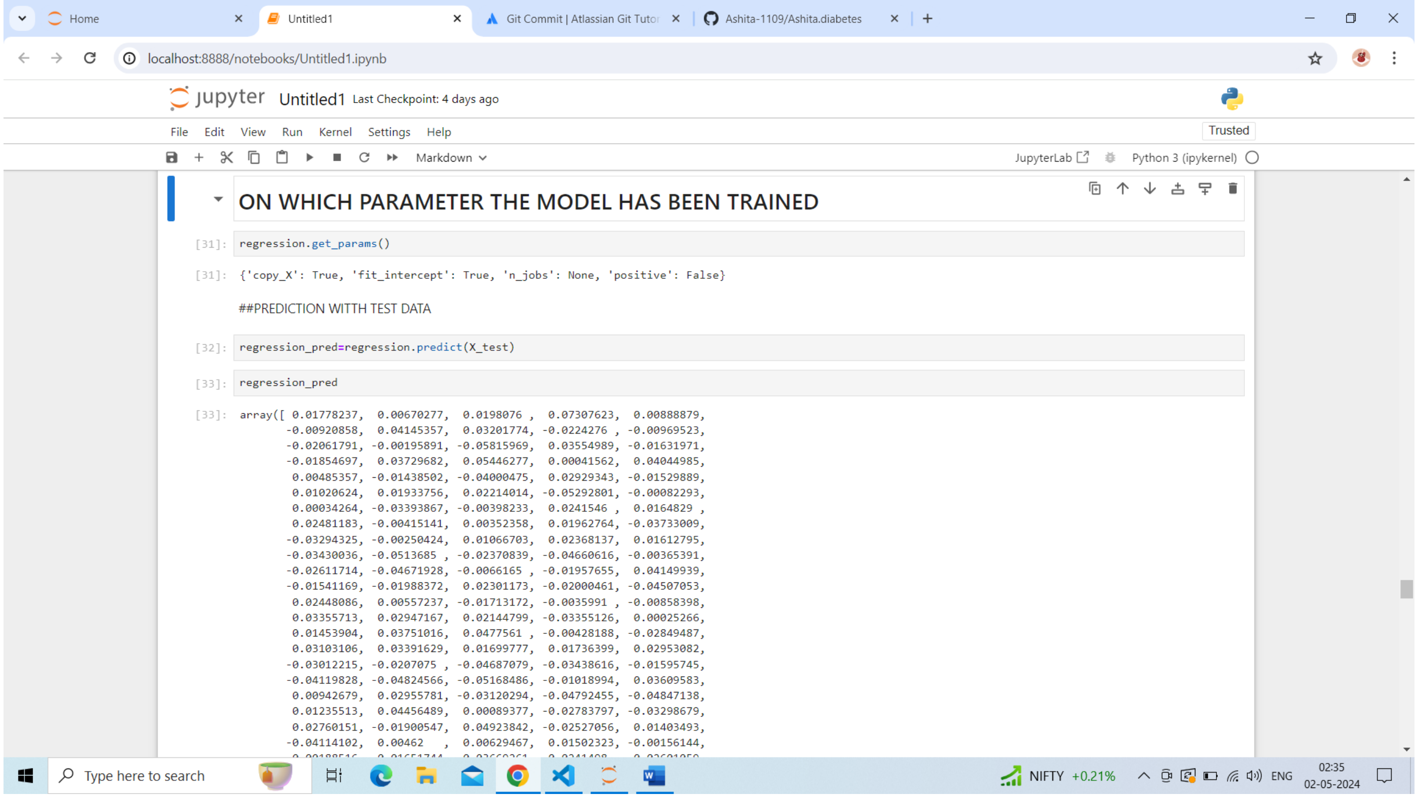
##PRINT THE COEFFICIENTS AND THE INTERCEPT

```
[28]: print(regression.coef_)

      [ 0.00427645  0.00105073  0.00602607  0.00879218 -0.01531857  0.01456688
        0.00486422  0.00873907  0.01511755]

[29]: print(regression.intercept_)

      0.0030778142911155265
```

ON WHICH PARAMETER THE MODEL HAS BEEN TRAINED

```
[31]: regression.get_params()

[31]: {'copy_X': True, 'fit_intercept': True, 'n_jobs': None, 'positive': False}
```

##PREDICTION WITTH TEST DATA

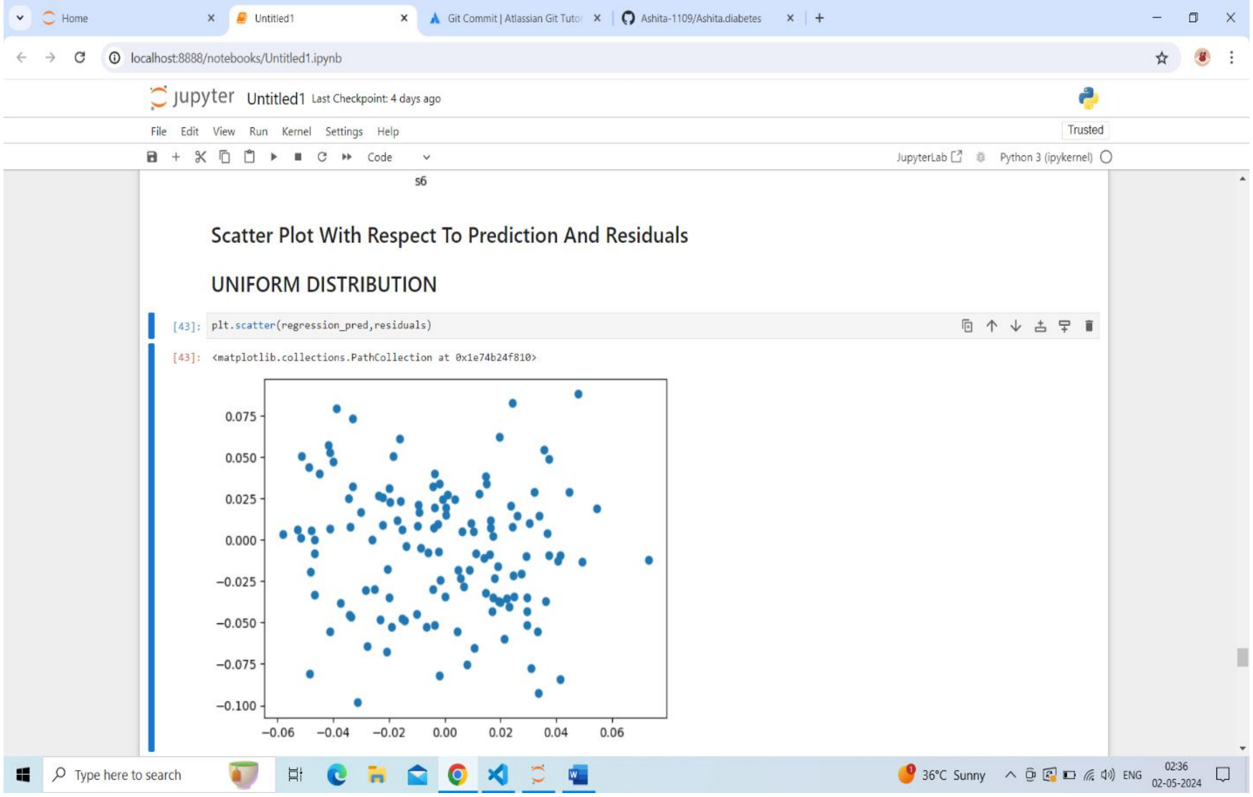

Home
Untitled1
Git Commit | Atlassian Git Tutor
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
Trusted
File Edit View Run Kernel Settings Help
Markdown
JupyterLab
Python 3 (ipykernel)
Type here to search
NIFTY +0.21%
ENG
02:35
02-05-2024

ON WHICH PARAMETER THE MODEL HAS BEEN TRAINED

```
[31]: regression.get_params()

[31]: {'copy_X': True, 'fit_intercept': True, 'n_jobs': None, 'positive': False}
```

##PREDICTION WITTH TEST DATA

```
[32]: regression_pred=regression.predict(X_test)

[33]: regression_pred

[33]: array([ 0.01778237,  0.00670277,  0.0198076 ,  0.07307623,  0.00888879,
             -0.00920858,  0.04145357,  0.03201774, -0.0224276 , -0.00969523,
             -0.02061791, -0.00195891, -0.05815969,  0.03554989, -0.01631971,
             -0.01854697,  0.03729682,  0.05446277,  0.00041562,  0.04044985,
              0.00485357, -0.01438502, -0.04000475,  0.02929343, -0.01529889,
              0.01020624,  0.01933756,  0.02214014, -0.05292801, -0.00082293,
              0.00034264, -0.03393867, -0.00398233,  0.0241546 ,  0.0164829 ,
              0.02481183, -0.00415141,  0.00352358,  0.01962764, -0.03733009,
             -0.03294325, -0.00250424,  0.01066703,  0.02368137,  0.01612795,
             -0.03430036, -0.0513685 , -0.02370839, -0.04660616, -0.00365391,
             -0.02611714, -0.04671928, -0.0066165 , -0.01957655,  0.04149939,
             -0.01541169, -0.01988372,  0.02301173, -0.02000461, -0.04507053,
              0.02448086,  0.00557237, -0.01713172, -0.0035991 , -0.00858398,
              0.03355713,  0.02947167,  0.02144799, -0.03355126,  0.00025266,
              0.01453904,  0.03751016,  0.0477561 , -0.00428188, -0.02849487,
              0.03103106,  0.03391629,  0.01699777,  0.01736399,  0.02953082,
             -0.03012215, -0.0207075 , -0.04687079, -0.03438616, -0.01595745,
             -0.04119828, -0.04824566, -0.05168486, -0.01018994,  0.03609583,
              0.00942679,  0.02955781, -0.03120294, -0.04792455, -0.04847138,
              0.01235513,  0.04456489,  0.00089377, -0.02783797, -0.03298679,
              0.02760151, -0.01900547,  0.04923842, -0.02527056,  0.01403493,
             -0.04114102,  0.00462   ,  0.00629467,  0.01502323, -0.00156144,
```


Home
Untitled1
Git Commit | Atlassian Git Tuto
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
Trusted
File Edit View Run Kernel Settings Help
Code
JupyterLab
Python 3 (ipykernel)
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
−0.06
−0.04
−0.02
0.00
0.02
0.04
0.06
Type here to search
36°C Sunny
ENG
02:36
02-05-2024

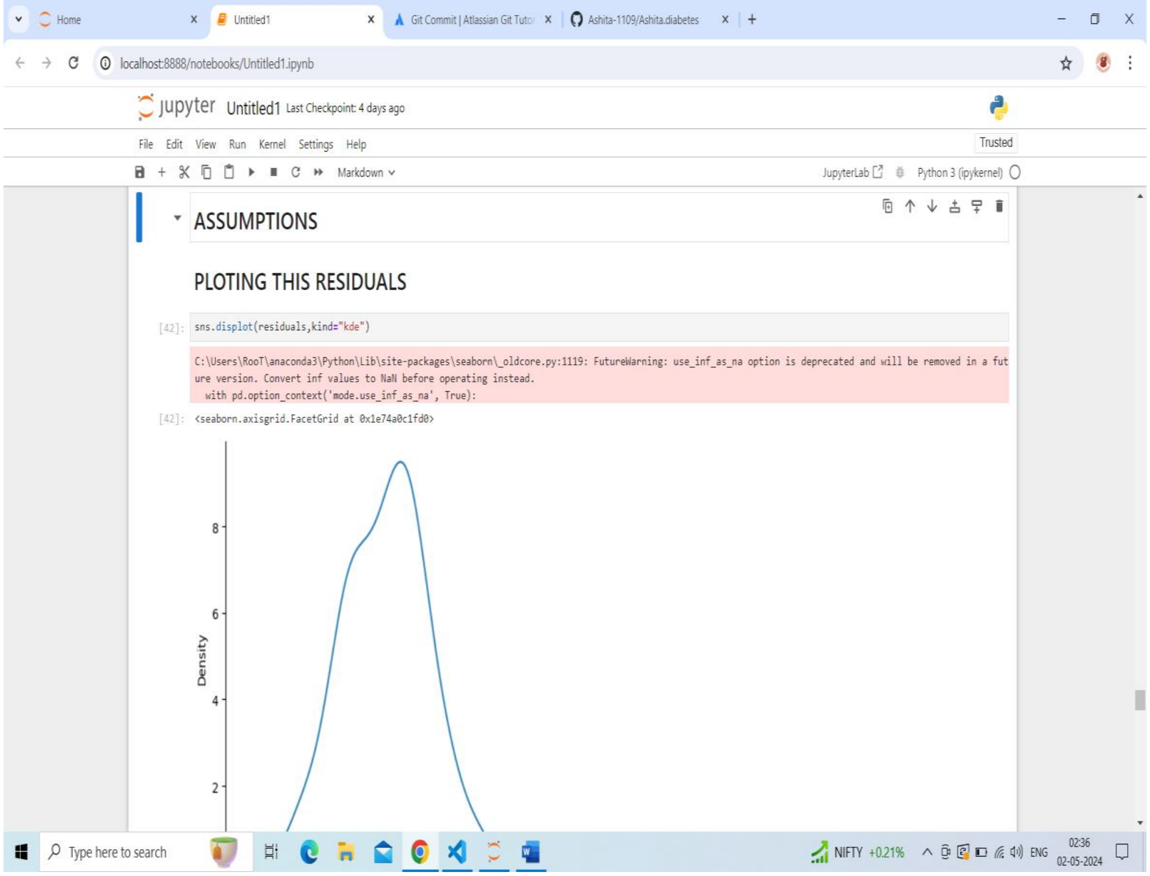
s6

Scatter Plot With Respect To Prediction And Residuals

UNIFORM DISTRIBUTION

```
[43]: plt.scatter(regression_pred,residuals)
```

```
[43]: <matplotlib.collections.PathCollection at 0x1e74b24f810>
```


Home
Untitled1
Git Commit | Atlassian Git Tuto
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
Trusted
File Edit View Run Kernel Settings Help
Markdown
JupyterLab
Python 3 (ipykernel)
8
6
4
2
Density
Type here to search
NIFTY +0.21%
ENG
02:36
02-05-2024

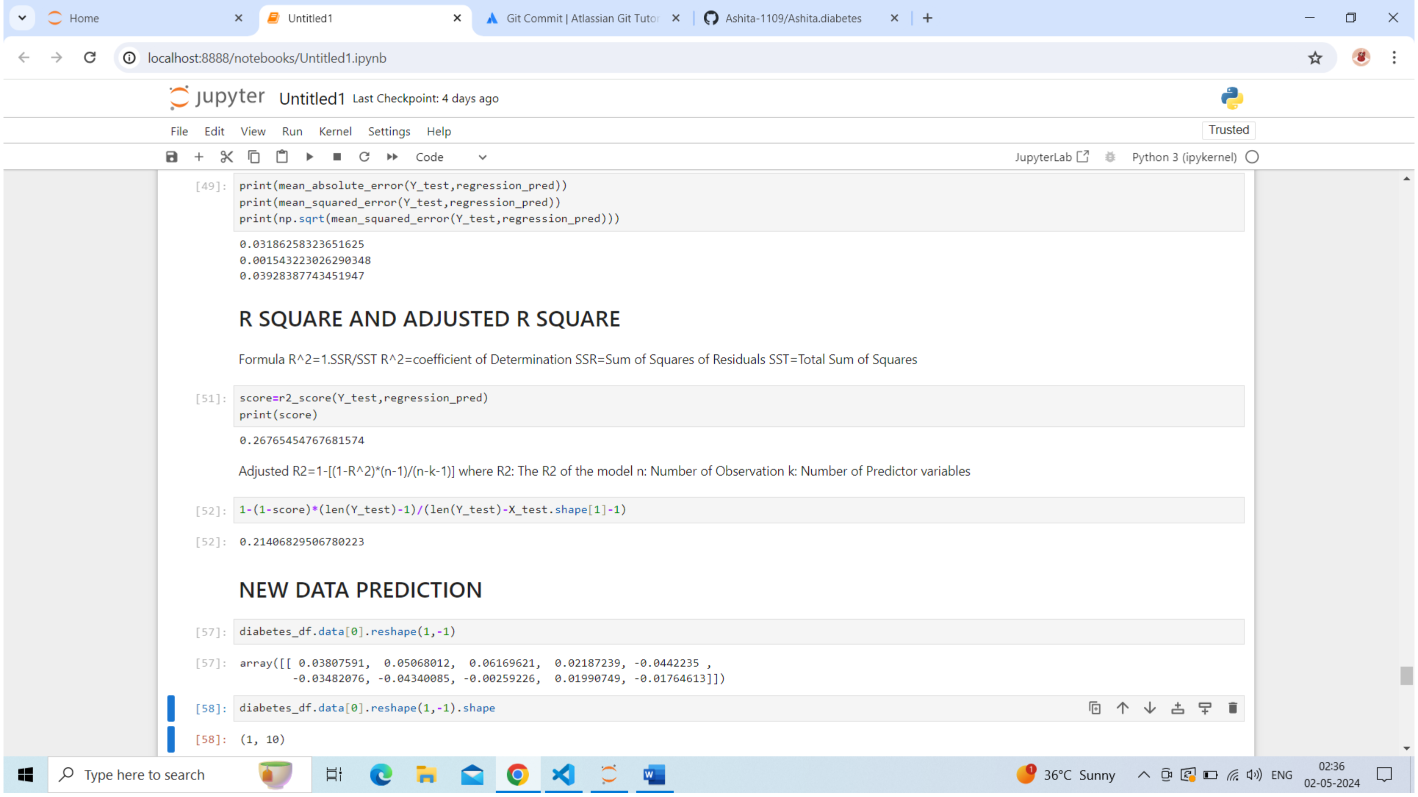
ASSUMPTIONS

PLOTING THIS RESIDUALS

```
[42]: sns.displot(residuals,kind="kde")
```

```
C:\Users\RooT\anaconda3\Python\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a fut
ure version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

```
[42]: <seaborn.axisgrid.FacetGrid at 0x1e74a0c1fd0>
```


Home
Untitled1
Git Commit | Atlassian Git Tutor
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

```
[49]: print(mean_absolute_error(Y_test,regression_pred))
      print(mean_squared_error(Y_test,regression_pred))
      print(np.sqrt(mean_squared_error(Y_test,regression_pred)))

      0.03186258323651625
      0.001543223026290348
      0.03928387743451947
```

R SQUARE AND ADJUSTED R SQUARE

Formula R^2=1.SSR/SST R^2=coefficient of Determination SSR=Sum of Squares of Residuals SST=Total Sum of Squares

```
[51]: score=r2_score(Y_test,regression_pred)
      print(score)

      0.26765454767681574
```

Adjusted R2=1-[(1-R^2)*(n-1)/(n-k-1)] where R2: The R2 of the model n: Number of Observation k: Number of Predictor variables

```
[52]: 1-(1-score)*(len(Y_test)-1)/(len(Y_test)-X_test.shape[1]-1)

[52]: 0.21406829506780223
```

NEW DATA PREDICTION

```
[57]: diabetes_df.data[0].reshape(1,-1)

[57]: array([[ 0.03807591,  0.05068012,  0.06169621,  0.02187239, -0.0442235 ,
              -0.03482076, -0.04340085, -0.00259226,  0.01990749, -0.01764613]])

[58]: diabetes_df.data[0].reshape(1,-1).shape

[58]: (1, 10)
```

3.4 PREDICTION FROM TRAINING DATA

3.4.1 LINEAR REGRESSION

Linear Regression is used to show the relationship between the dependent and the Independent variables.

3.4.2 TRAIN VALUES AND TEST VALUES

Train values and test values are very important evaluations of Linear Regression models by evaluating the training data with the concerns toward the requirement or the best fit outcomes.

3.4.3 FEATURES IN LINEAR REGRESSION

A. PARAMETERS

Linear Regression has two main parameters slope and intercepts. The slope represents the change in the dependent variable for a unit of change in independent variable. The intercept is the value of the variables when the independent variable is zero.

B. ASSUMPTIONS

Linear Regression is a parametric technique that relies on parameters learned from the data. The data must fulfil certain assumptions to obtain reliable results.

C. GRADIENT DESCENT

A Linear Regression model can be trained using the optimization algorithm gradient descent. The algorithm iteratively modifies the model parameters to reduce the mean square error of the model on a training set.

D. ORDINARY LEAST SQUARES

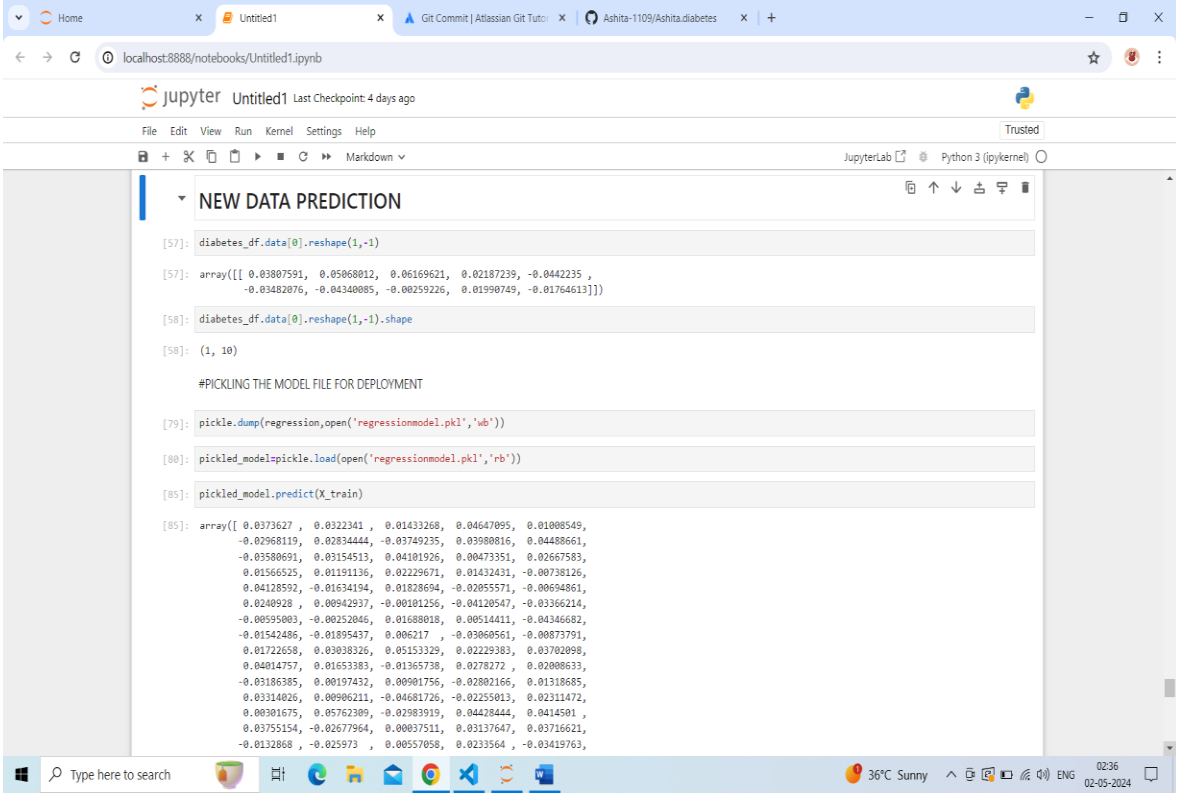
The ordinary least square procedure seeks to minimize the sum of the square residual.

E. COST FUNCTION

The cost function helps to figure out the best possible value which provide the best fit line for the training data.

F. MULTICOLLINEARITY

Multicollinearity in regression analysis occur when two or more predictors or independent Variable is highly correlated.



NEW DATA PREDICTION

```
[57]: diabetes_df.data[0].reshape(1,-1)

[57]: array([[ 0.03807591,  0.05068012,  0.06169621,  0.02187239, -0.0442235 ,
              -0.03482076, -0.04340085, -0.00259226,  0.01990749, -0.01764613]])

[58]: diabetes_df.data[0].reshape(1,-1).shape

[58]: (1, 10)
```

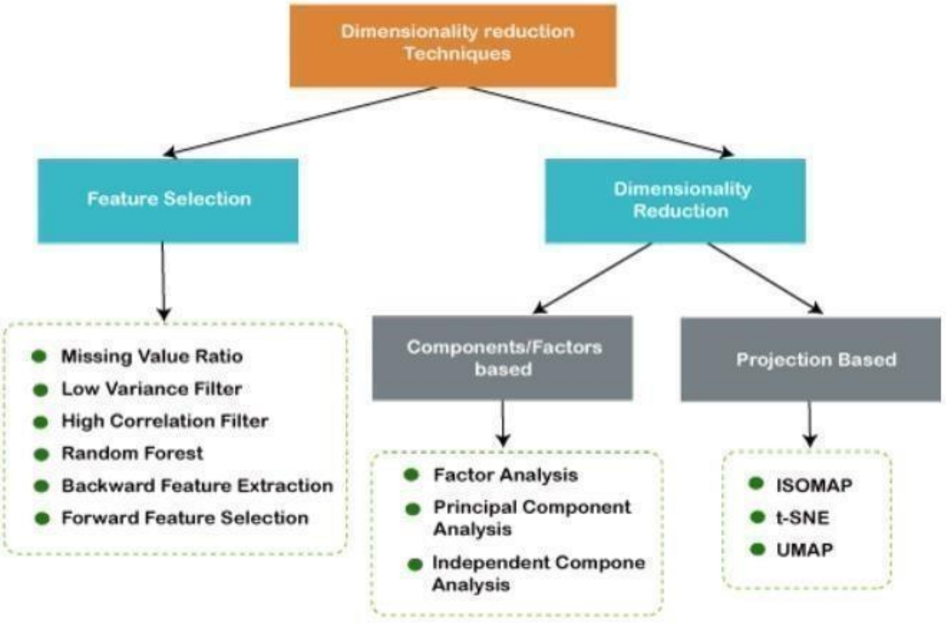
#PICKLING THE MODEL FILE FOR DEPLOYMENT

```
[79]: pickle.dump(regression,open('regressionmodel.pkl','wb'))

[80]: pickled_model=pickle.load(open('regressionmodel.pkl','rb'))

[85]: pickled_model.predict(X_train)

[85]: array([ 0.0373627 ,  0.0322341 ,  0.01433268,  0.04647095,  0.01008549,
             -0.02968119,  0.02834444, -0.03749235,  0.03980816,  0.04488661,
             -0.03580691,  0.03154513,  0.04101926,  0.00473351,  0.02667583,
              0.01566525,  0.01191136,  0.02229671,  0.01432431, -0.00738126,
              0.04128592, -0.01634194,  0.01828694, -0.02055571, -0.00694861,
              0.0240928 ,  0.00942937, -0.00101256, -0.04120547, -0.03366214,
             -0.00595003, -0.00252046,  0.01688018,  0.00514411, -0.04346682,
             -0.01542486, -0.01895437,  0.006217  , -0.03060561, -0.00873791,
              0.01722658,  0.03038326,  0.05153329,  0.02229383,  0.03702098,
              0.04014757,  0.01653383, -0.01365738,  0.0278272 ,  0.02008633,
             -0.03186385,  0.00197432,  0.00901756, -0.02802166,  0.01318685,
              0.03314026,  0.00906211, -0.04681726, -0.02255013,  0.02311472,
              0.00301675,  0.05762309, -0.02983919,  0.04428444,  0.0414501 ,
              0.03755154, -0.02677964,  0.00037511,  0.03137647,  0.03716621,
             -0.0132868 , -0.025973  ,  0.00557058,  0.0233564 , -0.03419763,
```

Home
Untitled1
Git Commit | Atlassian Git Tutor
Ashita-1109/Ashita.diabetes
localhost:8888/notebooks/Untitled1.ipynb
jupyter Untitled1 Last Checkpoint: 4 days ago
Trusted
File Edit View Run Kernel Settings Help
Markdown
JupyterLab
Python 3 (ipykernel)
Type here to search
36°C Sunny
ENG
02:36
02-05-2024

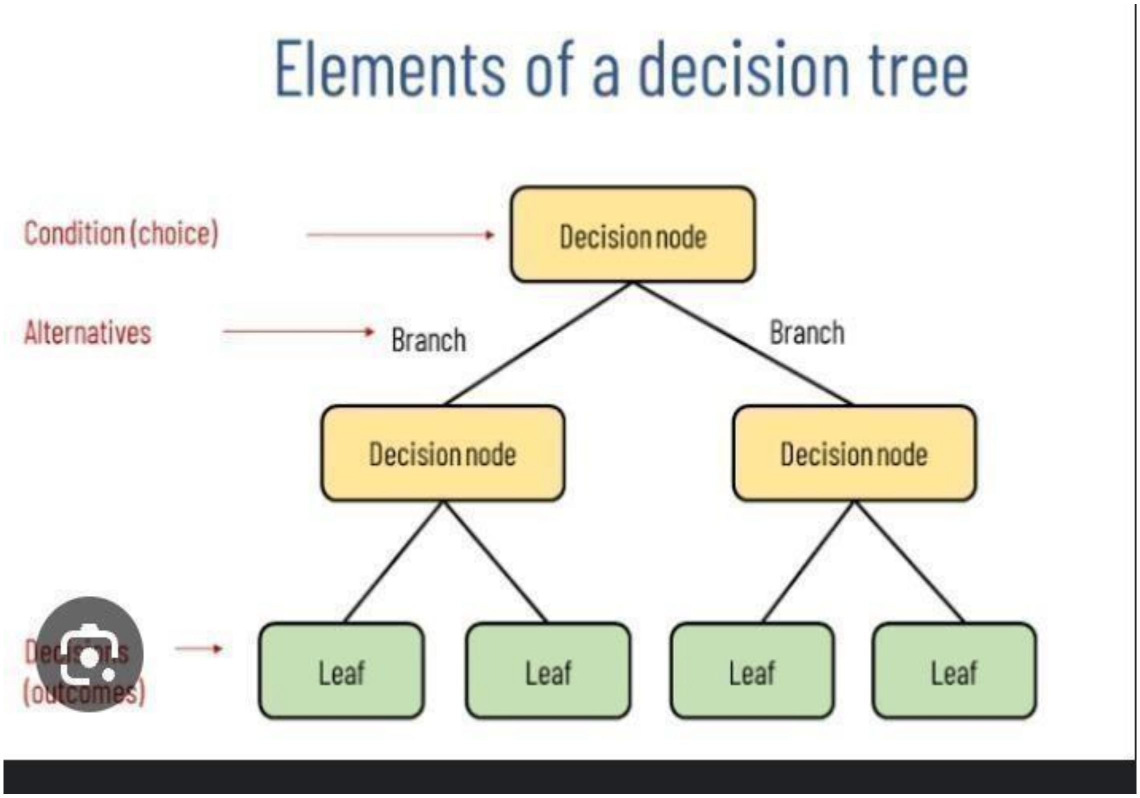
3.5 INTRODUCTION TO DIMENSIONALITY REDUCTION

Dimensionality Reduction is mainly the collection of the technique used to reduce the number of features, reduce dimension in a dataset. It improves the performance of any model by reducing it complexity and help in better visualization.



3.6 DECISION TREE

Decision Tree can be used for both regression and classification problem but mainly preferred for classification problems. In a decision tree, decision nodes are used to make any decision and have multiple branches. Leaf nodes are the output of those decision. In order to build a tree we use CART algorithm.



3.7 NAÏVE BAYES CLASSIFICATION THOREM

It is a probabilistic classifier algorithm which means its prediction is done on the basis of Probability of an object. Bayes theorem is also known as Bayes rule which is used to Determine the probability of a hypothesis. It depends on the conditional probability.

P(A/B) =(P(B/A) *P(A))/P(B)
P(A/B) =Posterior Probability P(A)=Prior Probability
P(B)= Marginal Probability P(B/A) =Likelihood Probability

WORKING PROCESS OF NAÏVE BAYES ARE:

1. Convert the given dataset into frequency table.

2. Generate likelihood table by finding the probability of the feature.
3. Now use the Bayes formula to calculate posterior probability

CONCLUSION

Linear Regression is able to predict values for both dependent and independent variables and determines the linear relationship between them along with the discussion of null or alternative hypothesis.

The cost function provides the best feasible value which leads to the best fit. But this Linear Regression is not applicable for complex computation.
It is very difficult to find the linear relationship between the variables.

REFERENCES

1. Types of Machine Learning from Java Point website

2. Quantitative vs Qualitative Data: What is the difference? By Emily Stevens

3. Writing Null Hypothesis in Research and Statistics by Joseph Quinones and Jennifer Mueller.

4. Logistics Regression from the Saedsayad.com website.

5. Regression vs Classification by Dhanush

6. K Means Clustering from Java Point website.