# Diabetes Prediction Machine Learning Model

*A project submitted in partial fulfilment of the*

*requirements for the award of the degree of*

**Bachelor of Technology**
**in**
**COMPUTER SCIENCE AND ENGINEERING**


Submitted by:

VRISHAV GARG - 12111051
ANURAG HANDA - 12111058
JASKIRAT SINGH - 12111082

Supervised by:

**Dr. Sourabh Jain**

Assistant professor

**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY**
**SONEPAT - 131201, HARYANA, INDIA**

# ACKNOWLEDGEMENT

The success and the outcome of this project required ceaseless guidance and assistance, my team members and I are extremely privileged to have got this all along the project.

We would like to take this opportunity to acknowledge all the people who have helped us whole heartedly in every stage of this project.

We are indebtedly grateful to Dr. Sourabh Jain, Assistant professor, CSE, IIIT SONEPAT for providing this opportunity in the first place and giving us all the support and guidance possible, in spite of having a busy schedule. We are indebtedly grateful for his help which played a very foremost part in the project and for providing us all the indispensable information for developing the machine learning model.

Vrishav Garg - 12111051

Anurag Handa – 12111058

Jaskirat Singh - 12111082

# SELF DECLARATION

I hereby state that work contained in the project titled “Diabetes Prediction Machine Learning Model” is original. I have followed the standards of the project ethics to the best of my abilities. I have acknowledged all the sources of knowledge which I have used in the project.

Vrishav Garg - 12111051

Anurag Handa – 12111058

Jaskirat Singh – 12111082

Department of Computer Science and Engineering,
Indian Institute of Information technology,
Sonepat-131201, Haryana, India

# CERTIFICATE

This is to certify that Mr. Jaskirat Singh, Mr. Vrishav Garg and Mr. Anurag Handa has worked on the project entitled “Diabetes Prediction Machine Learning Model” under my supervision and guidance.

The contents of the project, being submitted to the Department of Computer Science and Engineering, IIIT SONEPAT, HARYANA, for the award of the degree of B. Tech in Computer Science and Engineering, are original and carried out by candidate himself. This project has not been submitted in full or part for award of any other degree or diploma to this or any other university.

Dr. Sourabh Jain

Supervisor

Department of Computer Science and Engineering,
Indian Institute of Information technology,
Sonepat-131201, Haryana, India

# ABSTRACT

Our project is a Diabetes Prediction Machine Learning Model which focuses on prediction of diabetes from 7 biological features. This project involves more of research mindset of how to integrate Machine Learning and its applications in the field of healthcare so as to obtain better results, contribute to the society and save lots of precious lives. The prime objective here is to achieve the accurate possible predictions so as to detect the lethal disease at its earliest possible stages, which in turn would lead to adoption of early preventive measures and cures required.

The prime Results which wished to achieve are: -

That the people are able to diagnose themselves for diabetes through a better and effective way.

# Table of Contents

# Chapter 1

# Introduction

# 1.1 INTRODUCTION

## Brief Introduction to the Topic:

The project under work(PUW) is a diabetes machine learning model for Mellitus-type diabetes prediction. The model works at the best of its abilities to predict whether a person has diabetes or not based on the biological characteristics like age, gender, glucose, insulin, diabetes pedigree function, skin thickness and BMI. The basic reason for choosing this topic in the field of ML was to get real-time analysis of a patient condition and medical history to get the idea of patient illness and health condition so that the treatment can be faster, better and more specific to avoid the side effects caused by hit and trial method. ML in health care allows professionals to automate the administrative duties to provide better patient care. This technology can help in the early stages of medication research.

# 1.2 PROBLEM OUTLINE

Diabetes, as we all know can be a very lethal disease which is hard to get rid of once affected.

It is a disease in which the body's ability to produce or respond to the hormone insulin is impaired, resulting in abnormal metabolism of carbohydrates and elevated levels of glucose in the blood. Diabetes mellitus (DM) is a metabolic disease, involving inappropriately elevated blood glucose levels.

If the diabetes is not diagnosed at its early stages, it could also affect the other significant body parts rather than just kidneys, it can lead to

the weakening of retina leading to malfunctioning of eyes. It can cause heart attack, heart failure, stroke, kidney failure and coma. These complications can lead to your death.

Diabetes can be identified by bringing into notice the various biological features of a human and taking their reading like that of Glucose levels, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age etc.

The primary objective or the problem outline here is to diagnose and predict the chances of diabetes at the earliest possible stage to avoid the condition getting worse.

## 1.3 PROJECT OBJECTIVE

The main objective of this project is to provide a real time working machine learning model which is capable of detecting whether a person has diabetes or not. This would lead to early detection of diabetes thus making the patient aware about his condition so that he/she can take all the preventive measures required to avoid the condition getting worse.

## 1.4 PROJECT METHODOLOGY

Some data will be fed into the computer which will then be refined and cleaned. The data will further be analysed with graphs and plots. A suitable ML algorithm be then be applied used to train the machine and make correct predictions on the real-life user data.

## 1.5 SCOPE OF PROJECT

The scope of ML-AI application in medical sciences is not confined by any boundaries. On a daily basis, we see improvements in technologies used medical sciences and giving better results. The use of computers and machine models to predict diseases, perform body scan, tests of various body organs like ENT etc. without human intervention decrease the chance of errors while giving the much accurate results. Our project also follows the similar trend where in its initial stages, it is capable of predicting diabetes but can be further expanded to a complete disease prediction model.

## 1.6 LIMITATIONS

The project has the following limitations:

- **Diabetes Type:** The project is able to predict mellitus-type diabetes but at the same time, it is incapable of detecting insipidus-type diabetes.
- **Data requirement:** ML models generally require vast amount of data to train the model to get more accurate results. Data collection generally becomes a major cause of concern as collecting real world data often lead to errors and is a tedious task. Often the data collected is also not clean due to human errors or some other factors.

# 1.7 HISTORY

Important developments in biotechnology and, more importantly, high throughout computing is constantly contributing to quick and affordable data production, taking computational biology research into the world of big data.

DM is among the most widespread diseases (World Health Organization, 20202) for the elderly in the country. In 2017, 451million individuals globally are diabetic as informed by the International Diabetes Federation. Expectations are that this figure will rise to 693 million citizens over the next 26 years.

Studies have been performed on ML procedures, but certain essential facets of ML, like databases, pre-processing methods, and feature extraction and selection approaches used to identify Diabetes Mellitus and AI solutions to the need for intelligent DM assistants, are not addressed. As a consequence, attempts have been made in the sense of this analysis to examine existing literature on ML and AI approaches to DM studies.

# Chapter 2

# Study and review of literature

# 2.1 INTRODUCTION

The entire project has been made, run and tested the entire project on vs code, jupyter notebook. The entire project has been written in python language. The project consists of famous Machine Learning libraries recommended for Machine Learning like NumPy, Pandas, Scikit-Learn, Seaborn, Matplotlib.

# 2.2 PYTHON

Python is a popular programming language.
It was created by Guido Van Rossum and released in 1991.
Python is a preferred programming language because of its extensive capabilities, applicability, and simplicity. Python programming, which has separate platforms and is well-liked in the programming community, is ideally suited for machine learning.
The dependable environment provided by Python frameworks and libraries speeds up programme development dramatically. When working on complicated projects, developers might use libraries to essentially employ pre-written code to speed up development.

# 2.3 VS CODE

V S Code is an IDE used for writing and editing the codes. It has a user-friendly interface and provides wide range of features and utility tools for writing effective codes and smooth execution. It is highly preferred by professionals in today's world.

## 2.3 JUPYTER NOTEBOOK

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

## 2.4 LIBRARIES USED

**2.4.1 NumPy:** NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

**2.4.2 Pandas**: Pandas is mainly used for data analysis and associated manipulation of tabular data in Data Frames. Pandas allows importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel.

**2.4.3 Scikit-Learn:** Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction.

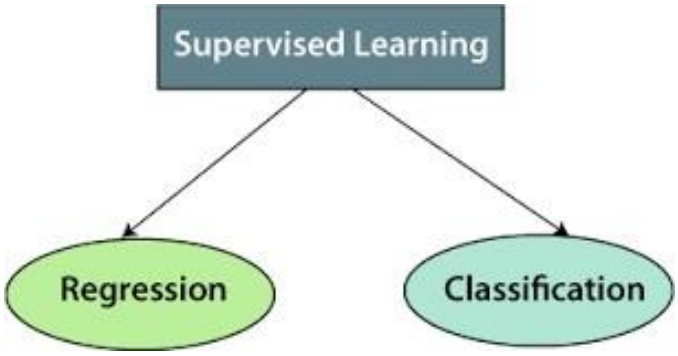
**2.4.4 Matplotlib:** Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication plots. Make interactive figures that can zoom, pan, update. Customize visual style and layout.

**2.4.5 Seaborn:** 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. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

## 2.5 Machine Learning

This project uses Classification Based Supervised Learning Model.

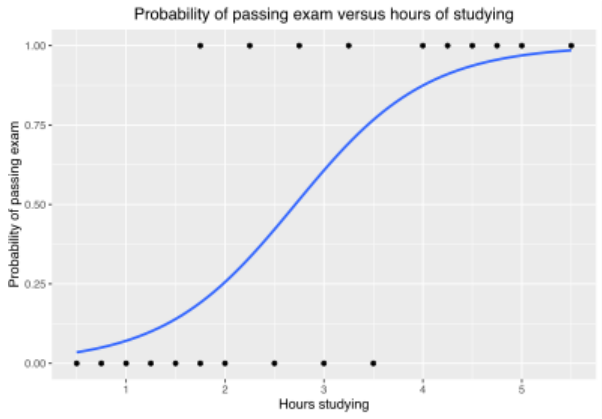
**2.5.1 Supervised Machine Learning** is the type of machine learning where we train the machine with training set which contains large number of training examples with features and targets. Here, we need to provide right answers to the input variables using which the machine learns.




**2.5.2 Classification** means predicting a small no. of outputs or more precisely it belongs to predicting a small finite limited set of possible Output Categories.

## 2.5.3 Logistic Regression

Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.



# Chapter 3

# Implementation

## 3.1 Flow of a general Machine Learning Project

1. **Data Collection –** The initial step of any machine learning model revolves collecting the real-world data and their respective outcomes so that we can train based on that data.

2. **Data Cleaning –** Data collected often has some absurd values which are easily observable and needs to be eliminated so that our model trains correctly. Here we clean and refine the data often be replacing the NULL/ absurd values with their mean values.

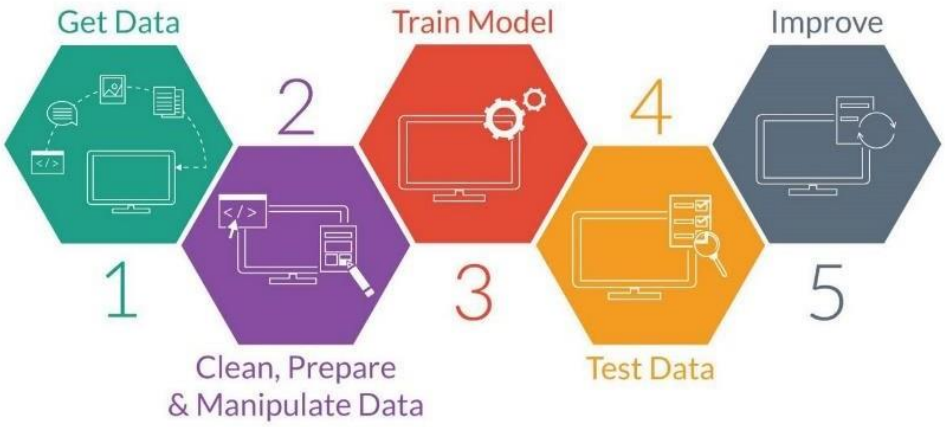
3. **Data Analysis –** After refining the data, we analyze it by drawing its various graphs, which help us to observe its trends and also that which independent variable majorly governs the output variable, this can be majorly predicted by using heatmaps, correlations and Pearson's Coefficient.

4. **Data Standardization -** Data standardization is a data processing workflow that converts the structure of different datasets into one common format of data. It deals with the transformation of datasets after the data are collected from different sources and before it is loaded into target systems.

5. **Splitting Data** – We split our model into two parts known as train data and test data. The train data is then used to train the model and the test data is used to test our model before it can be actually deployed.

6. **Model Selection –** Based on the type of problem we choose (Supervised or unsupervised), an appropriate ML algorithm is chosen which can achieve the highest accuracy

.



## 3.2 Code Snippets

This section contains the snippets from of the VS Code IDE where the entire code is written to explain its functioning.

1. Importing the Libraries required

Importing the Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
from pandas_profiling import ProfileReport
```

✓ 0.8s

2. Data Collection

DATA COLLECTION

```
data = pd.read_csv('Diabetes_dataset.csv')
```

# 3. Data Analysis

DATA ANALYSIS

```
tup = data.shape
rows = tup[0]
columns = tup[1]
print ("Rows ",rows)
print ("Columns ",columns)
```

```
Rows  768
Columns  8
```

```
data.columns
```

```
Index(['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI',
       'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

```
data.head(10)
```

| | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|
| 0 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| 2 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| 3 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| 4 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |
| 5 | 116 | 74 | 0 | 0 | 25.6 | 0.201 | 30 | 0 |
| 6 | 78 | 50 | 32 | 88 | 31.0 | 0.248 | 26 | 1 |
| 7 | 115 | 0 | 0 | 0 | 35.3 | 0.134 | 29 | 0 |
| 8 | 197 | 70 | 45 | 543 | 30.5 | 0.158 | 53 | 1 |
| 9 | 125 | 96 | 0 | 0 | 0.0 | 0.232 | 54 | 1 |

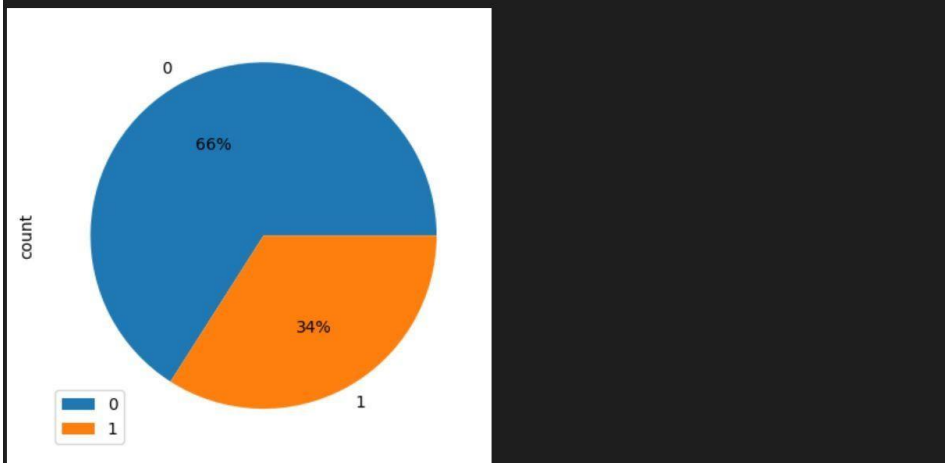
```
data["Outcome"].value_counts()
```

```
0    500
1    268
```

# 4. Plotting the graphs

```
datafr = pd.DataFrame({'Result' : ['0','1'],'count' : [500,258]})
datafr.groupby(['Result']).sum().plot(kind = "pie", y = "count", autopct = "%1.0f%%")
```

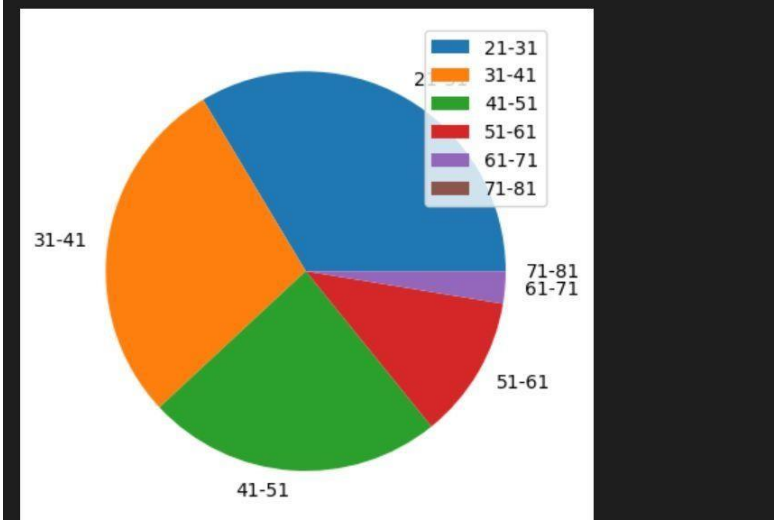
```
<AxesSubplot: ylabel='count'>
```




Pie Chart depicting the percentage of people having and not having diabetes

```
plt.pie(L,labels=label)
plt.legend()
```

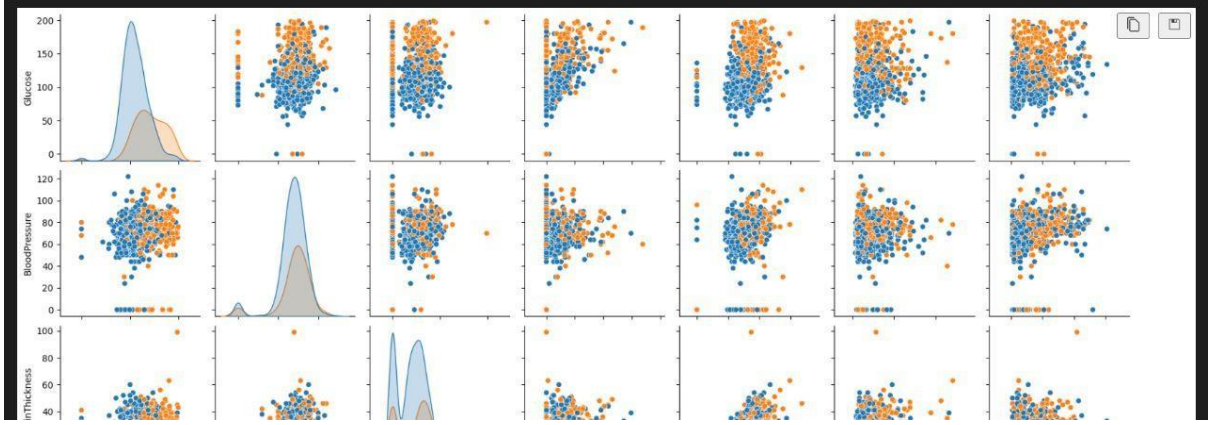
```
<matplotlib.legend.Legend at 0x16428187fa0>
```

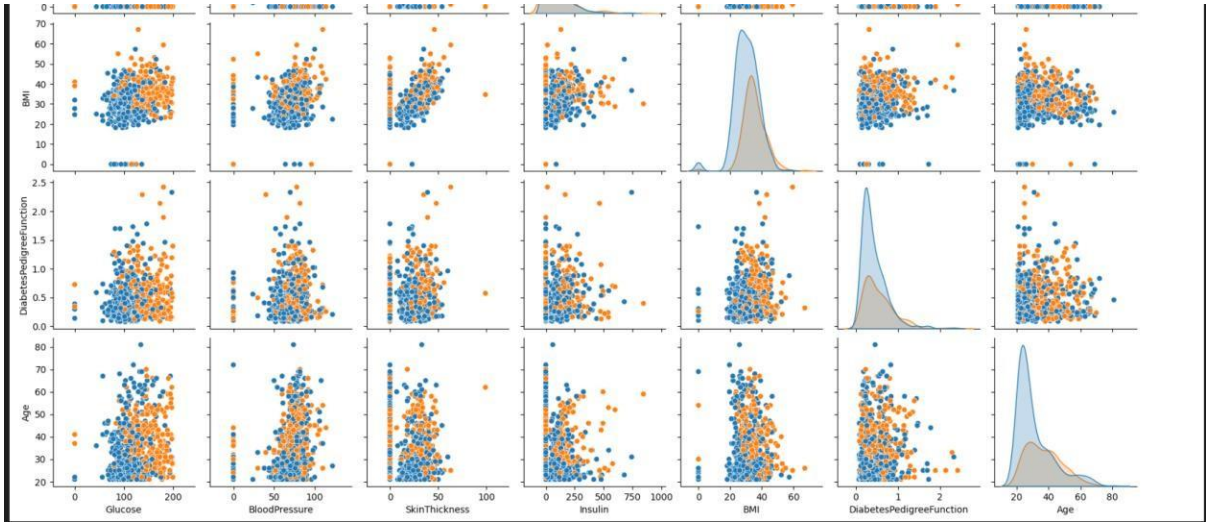



Graph depicting the affected people in different age groups

```
sb.pairplot(data, diag_kind='kde',hue = "Outcome")
```

```
<seaborn.axisgrid.PairGrid at 0x1643414bd30>
```




# Pair Plots

# 5. Data Cleaning

Splitting the data between Input and Output variables

```
input_variables = data.drop("Outcome",axis=1) # independent feature
result = data["Outcome"] # dependent feature
```

DATA CLEANING

Eliminating NULL (Zero) values in the data table

```
heads = input_variables.columns
for i in heads:
    avg = input_variables[i][input_variables[i] != 0].mean()
    input_variables[i].fillna(avg)
    for j in range(768):
        if (input_variables[i][j]==0):
            input_variables[i][j] = avg
```

```
input_variables
```

| | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age |
|---|---|---|---|---|---|---|---|
| 0 | 148.0 | 72.0 | 35.00000 | 155.548223 | 33.6 | 0.627 | 50 |
| 1 | 85.0 | 66.0 | 29.00000 | 155.548223 | 26.6 | 0.351 | 31 |
| 2 | 183.0 | 64.0 | 29.15342 | 155.548223 | 23.3 | 0.672 | 32 |
| 3 | 89.0 | 66.0 | 23.00000 | 94.000000 | 28.1 | 0.167 | 21 |
| 4 | 137.0 | 40.0 | 35.00000 | 168.000000 | 43.1 | 2.288 | 33 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 763 | 101.0 | 76.0 | 48.00000 | 180.000000 | 32.9 | 0.171 | 63 |
| 764 | 122.0 | 70.0 | 27.00000 | 155.548223 | 36.8 | 0.340 | 27 |
| 765 | 121.0 | 72.0 | 23.00000 | 112.000000 | 26.2 | 0.245 | 30 |
| 766 | 126.0 | 60.0 | 29.15342 | 155.548223 | 30.1 | 0.349 | 47 |
| 767 | 93.0 | 70.0 | 31.00000 | 155.548223 | 30.4 | 0.315 | 23 |

768 rows × 7 columns

# 6. Data Standardization

DATA STANDARDIZATION / TRANSFORMATION

```
scaler = StandardScaler()
```

```
scaler.fit(input_variables)
```

```
▾ StandardScaler
StandardScaler()
```

+ Code + Markdown

```
scaled_data = scaler.transform(input_variables)
```

```
scaled_data
```

```
array([[ 8.65108070e-01, -3.35182392e-02,  6.65502121e-01, ...,
         1.66291742e-01,  4.68491977e-01,  1.42599540e+00],
```

# 7. Different methods for showing decencies

## i) Correlations

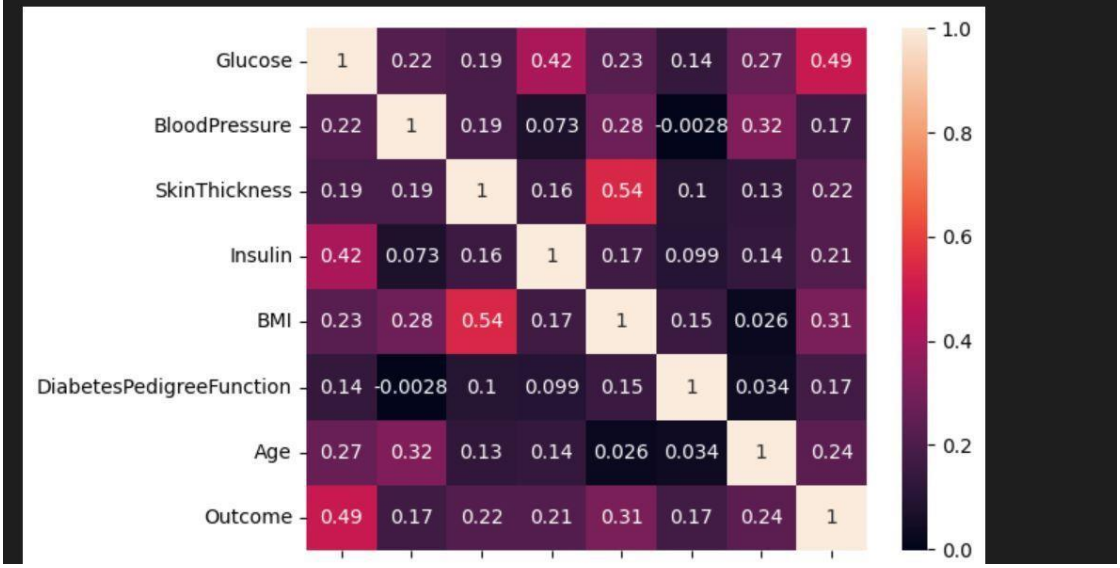
```
newdata.corr()
```

| | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|
| Glucose | 1.000000 | 0.218367 | 0.192991 | 0.420157 | 0.230941 | 0.137060 | 0.266534 | 0.492928 |
| BloodPressure | 0.218367 | 1.000000 | 0.192816 | 0.072517 | 0.281268 | -0.002763 | 0.324595 | 0.166074 |
| SkinThickness | 0.192991 | 0.192816 | 1.000000 | 0.158139 | 0.542398 | 0.100966 | 0.127872 | 0.215299 |
| Insulin | 0.420157 | 0.072517 | 0.158139 | 1.000000 | 0.166586 | 0.098634 | 0.136734 | 0.214411 |
| BMI | 0.230941 | 0.281268 | 0.542398 | 0.166586 | 1.000000 | 0.153400 | 0.025519 | 0.311924 |
| DiabetesPedigreeFunction | 0.137060 | -0.002763 | 0.100966 | 0.098634 | 0.153400 | 1.000000 | 0.033561 | 0.173844 |
| Age | 0.266534 | 0.324595 | 0.127872 | 0.136734 | 0.025519 | 0.033561 | 1.000000 | 0.238356 |
| Outcome | 0.492928 | 0.166074 | 0.215299 | 0.214411 | 0.311924 | 0.173844 | 0.238356 | 1.000000 |

## ii) Heatmap

```
sb.heatmap(newdata.corr(), annot=True)
```

```
<AxesSubplot: >
```




## iii) Pearson's Coefficient

Pearson's Coefficient

```
from scipy.stats import pearsonr
```

```
features = list(input_variables.columns)
target = "Outcome"
for i in features:
    print(i,pearsonr(data[i],data[target]))
```

```
Glucose PearsonRResult(statistic=0.4665813983068734 pvalue=8.935431645288658e-43)
BloodPressure PearsonRResult(statistic=0.06506835955033273 pvalue=0.071513900097732)
SkinThickness PearsonRResult(statistic=0.07475223191831933 pvalue=0.03834770482047633)
Insulin PearsonRResult(statistic=0.13054795488404775 pvalue=0.000286186460360213)
BMI PearsonRResult(statistic=0.29269466264444516 pvalue=1.2298074873112102e-16)
DiabetesPedigreeFunction PearsonRResult(statistic=0.1738440656529598 pvalue=1.2546070101484021e-06)
Age PearsonRResult(statistic=0.2383559830271976 pvalue=2.2099754606646917e-11)
```

## iv) Train – Test Split

Train Test Split

```
in_train, in_test, op_train, op_test = train_test_split(input_variables, result, test_size = 0.15, stratify = result, random_stat
```

Python

```
in_train.shape[0]
```

Python

```
652
```

```
in_test.shape[0]
```

Python

```
116
```

```
(input_variables.shape[0]) == (in_train.shape[0]+in_test.shape[0])
```

Python

# 8. Machine Learning Algorithm

## 8.1 Linear Regression

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
model_regression = LogisticRegression()
model_regression.fit(in_train,op_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
prediction = model_regression.predict(in_test)
from sklearn.metrics import classification_report
print(classification_report(op_test,prediction))
```

```
              precision    recall  f1-score   support

           0       0.75      0.91      0.82        76
           1       0.71      0.42      0.53        40

    accuracy                           0.74       116
   macro avg       0.73      0.67      0.68       116
weighted avg       0.74      0.74      0.72       116
```

9. Building a user defined function where the user can enter his biological values to predict his diabetes

Trying model for REAL USER DATA

```
def write_data():
    L=[]
    L.append(input("Enter the Glucose: "))
    L.append(input("Enter the BloodPressure: "))
    L.append(input("Enter the SkinThickness: "))
    L.append(input("Enter the Insulin: "))
    L.append(input("Enter the BMI: "))
    L.append(input("Enter the DiabetesPedigreeFunction: "))
    L.append(input("Enter the Age: "))
    L.append(input("Enter the Outcome: "))
    d={}
    i=0
    user_data = pd.read_csv('user_data.csv')
    for x in user_data.columns:
        d[x]=[]
        d[x].append(L[i])
        i+=1
    df=pd.DataFrame(d)
    user_data=pd.concat((user_data,df),ignore_index=True)
    user_data.to_csv('user_data.csv',index=False)
```

# Summary and Future Expansion of Project

The project is almost ready on the python shell on VS Code but there is still much work that can be done on this project like we can still improve by collecting more and more data to train machine more efficiently achieve high accuracy.

Also we can create a user-friendly interface by integrating this on some Website or some Application.

# References


- https://www.sciencedirect.com/science/article/pii/S1877050920300557
- https://towardsdatascience.com/predicting-diabetes-with-machine-learning-part-i-f151cb764aee
- https://www.kaggle.com/code/ahmetcankaraolan/diabetes-prediction-using-machine-learning
- https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html
- https://towardsdatascience.com/predicting-diabetes-with-machine-learning-part-i-f151cb764aee
- https://www.tensorflow.org/guide/core/logistic_regression_core
- https://www.geeksforgeeks.org/ml-logistic-regression-using-tensorflow/
- https://www.youtube.com/watch?v=xUE7SjVx9bQ&t=2792s