**Project : Ai based diabetic prediction system**

Diabetic prediction using Machine learning

# Introduction

In an era marked by the convergence of healthcare and cutting-edge technology, the application of machine learning in the prediction of diabetes has become a pivotal advancement. Diabetes, a chronic metabolic disorder affecting millions worldwide, poses significant challenges for early detection and management. Machine learning, a subset of artificial intelligence, offers a powerful toolset to address this challenge. By harnessing the potential of vast datasets and sophisticated algorithms, machine learning models can analyze a multitude of factors, including genetic markers, lifestyle choices, and medical history, to predict the likelihood of an individual developing diabetes.

This predictive capability not only aids in early intervention and personalized healthcare but also contributes to the broader mission of preventing and managing this widespread chronic condition. In this exploration of diabetes prediction through machine learning, we delve into the methodologies, data sources, and real-world implications of this innovative approach to improving healthcare outcomes.”

**Data analysis:** Here one will get to know about how the data analysis part is done in a data science life cycle.

**Exploratory data analysis:** EDA is one of the most important steps in the data science project life cycle and here one will need to know that how to make inferences from the visualizations and data analysis

**Model building:** Here we will be using 4 ML models and then we will choose the best performing model.

**Saving model:** Saving the best model using pickle to make the prediction from real data.

# Importing libraries

Import numpy as np

Import pandas as pd

Import matplotlib.pyplot as plt

Import seaborn as sns

Sns.set()

From mlxtend.plotting import plot\_decision\_regions

Import missingno as msno

From pandas.plotting import scatter\_matrix

From sklearn.preprocessing import StandardScaler

From sklearn.model\_selection import train\_test\_split

From sklearn.neighbors import KNeighborsClassifier

From sklearn.metrics import confusion\_matrix

From sklearn import metrics

From sklearn.metrics import classification\_report

Import warnings

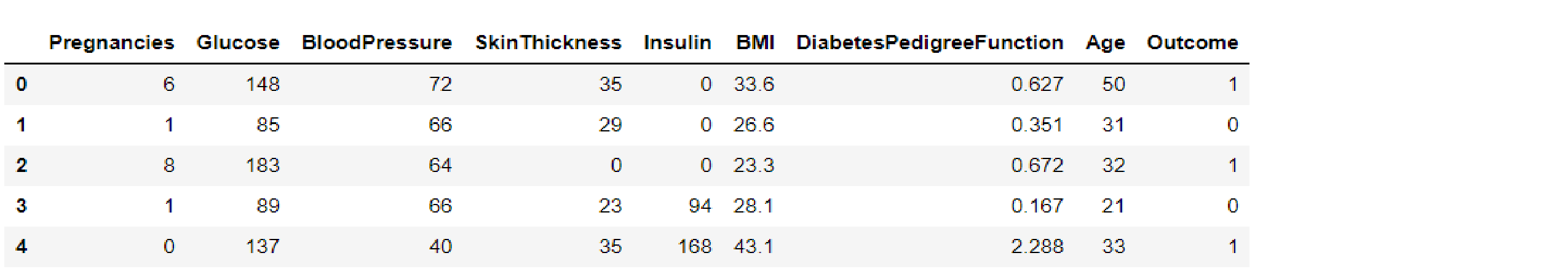
warnings.filterwarnings(‘ignore’)

%matplotlib inline

**Here we will be reading the dataset which is in the CSV format**

diabetes\_df = pd.read\_csv(‘diabetes.csv’)

diabetes\_df.head()

**Output:**

# Exploratory Data Analysis (EDA)

**Now let’ see that what are columns available in our dataset.**

diabetes\_df.columns **Output:**

Index([‘Pregnancies’, ‘Glucose’, ‘BloodPressure’, ‘SkinThickness’, ‘Insulin’,

‘BMI’, ‘DiabetesPedigreeFunction’, ‘Age’, ‘Outcome’],

Dtype=’object’)

----------------------------------------------------------------------------------------------------------------

**Information about the dataset**

diabetes\_df.info()

**Output:**

RangeIndex: 768 entries, 0 to 767 Data columns (total 9 columns):

# Column Non-Null Count Dtype

--- ------ -------------- ----- --------

1. Pregnancies 768 non-null int64
2. Glucose 768 non-null int64
3. BloodPressure 768 non-null int64
4. SkinThickness 768 non-null int64
5. Insulin 768 non-null int64
6. BMI 768 non-null float64
7. DiabetesPedigreeFunction 768 non-null float64
8. Age 768 non-null int64
9. Outcome 768 non-null int64

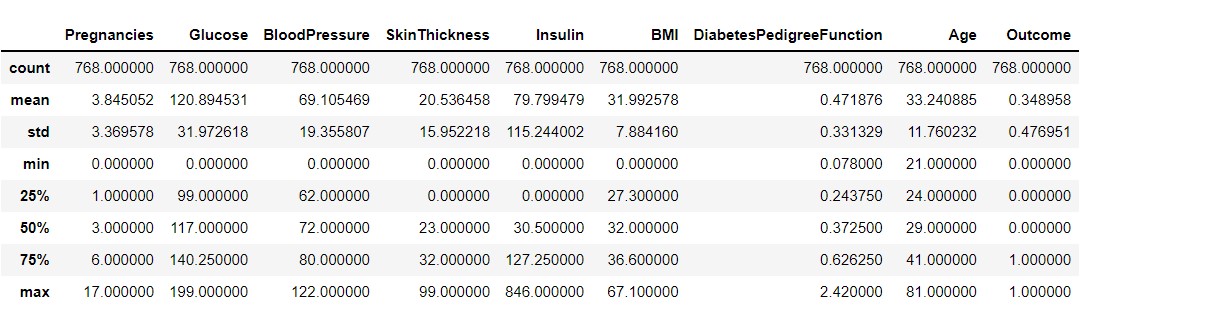
Dtypes: float64(2), int64(7)

Memory usage: 54.1 KB

**To know more about the dataset**

Diabetes\_df.describe()

**Output:**



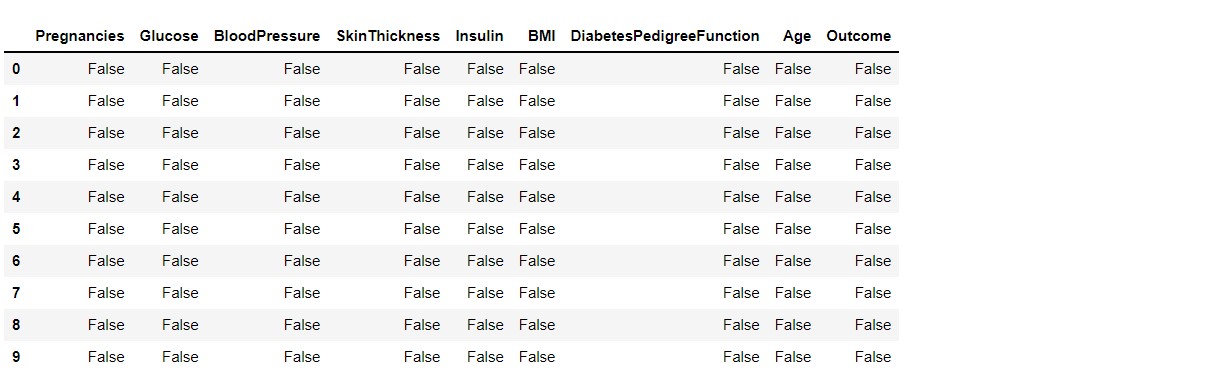
## **To know more about the dataset with transpose – here T is for the transpose**

diabetes\_df.describe().T **Output:**



## **Now let’s check that if our dataset have null values or not**

diabetes\_df.isnull().head(10)

**Output:** 

**Now let’s check the number of null values our dataset has.**

diabetes\_df.isnull().sum()

**Output:**

Pregnancies 0

Glucose 0

BloodPressure 0

SkinThickness 0

Insulin 0

BMI 0

DiabetesPedigreeFunction 0

Age 0

Outcome 0

dtype: int64

Here from the above code we first checked that is there any null values from the IsNull() function then we are going to take the sum of all those missing values from the sum() function and the inference we now get is that there are no missing values but that is actually not a true story as in this particular dataset all the missing values were given the 0 as a value which is not good for the authenticity of the dataset. Hence we will first replace the 0 value with the NAN value then start the imputation process.

Diabetes\_df\_copy = diabetes\_df.copy(deep = True)

Diabetes\_df\_copy[[‘Glucose’,’BloodPressure’,’SkinThickness’,’Insulin’,’BMI’]] = diabetes\_df\_copy[[‘Glucose’,’BloodPressure’,’SkinThickness’,’Insulin’,’BMI’]].replace(0,n p.NaN)

**# Showing the Count of NANs**

Print(diabetes\_df\_copy.isnull().sum())

**Output:**

Pregnancies 0

Glucose 5

BloodPressure 35

SkinThickness 227

Insulin 374

BMI 11

DiabetesPedigreeFunction 0

Age 0

Outcome 0

dtype: int64

As mentioned above that now we will be replacing the zeros with the NAN values so that we can impute it later to maintain the authenticity of the dataset as well as trying to have a better Imputation approach i.e to apply mean values of each column to the null values of the respective columns.

**Conclusion**

After using all these patient records, we are able to build a machine learning model (random forest – best one) to accurately predict whether or not the patients in the dataset have diabetes or not along with that we were able to draw some insights from the data via data analysis and visualization.