**DATA VISUALIZATION**

|  |  |
| --- | --- |
| Date | 18 October 2023 |
| Team id | Proj-212176-Team-2 |
| Project Name | AI based Diabetes Prediction System |
| Maximum mark |  |

Data visualization is an important step in understanding your dataset when working on an AI-based diabetes detection project. You can use libraries like Matplotlib and Seaborn in Python to create various types of visualizations.

The following codes will provides examples of various data visualization techniques:

1. Displaying the first few rows of the dataset to get an overview.

2. Generating summary statistics for numerical features.

3. Creating histograms to visualize the distribution of numerical features.

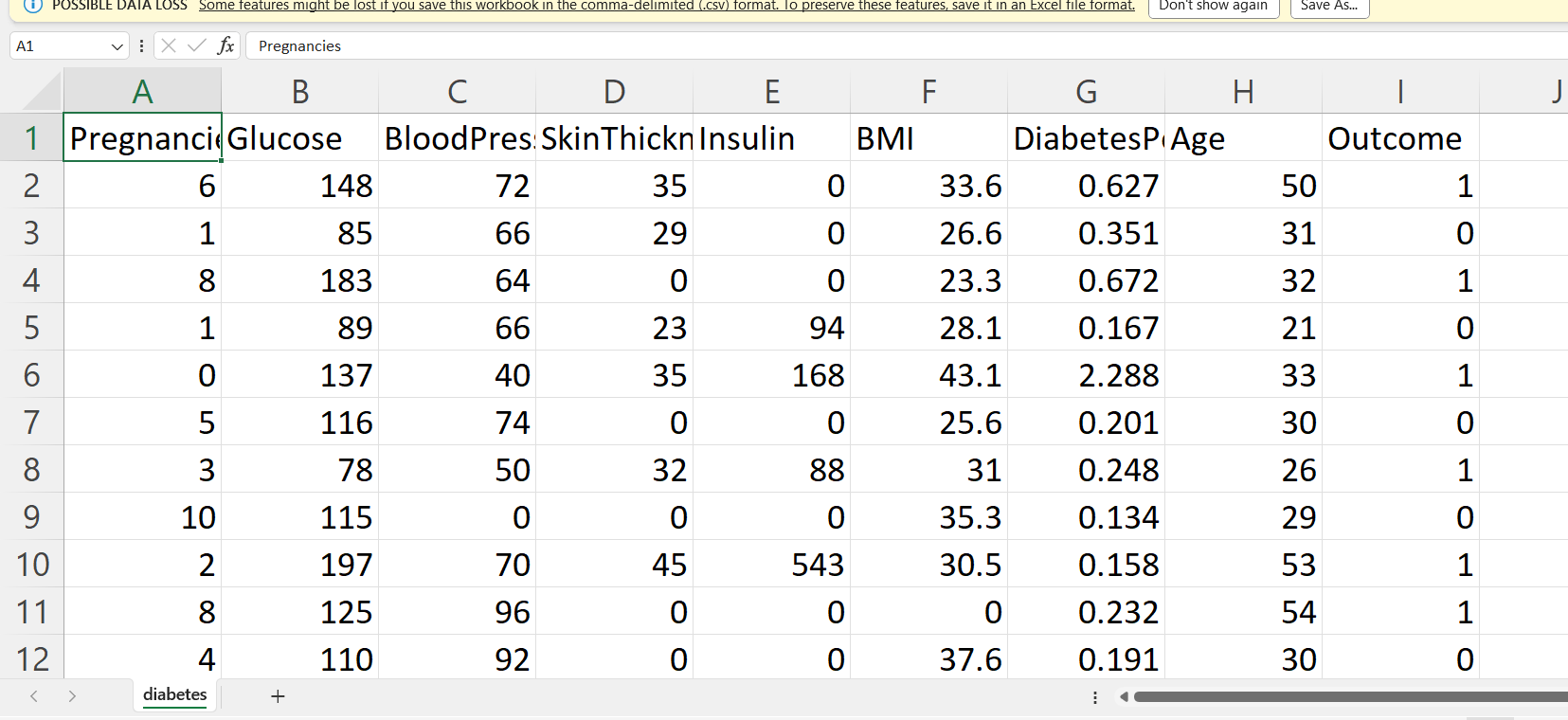
4. Generating boxplots to identify potential outliers.

5. Creating a pairplot to visualize relationships between features, with hue indicating the outcome class.

6. Creating a correlation heatmap to visualize feature correlations.

**Program :**

**Given dataset:**

****

**Import the necessary libraries:**

Numpy,pandas,sklearn ,matplotlib.pyplot

**Explaination:**

• Numpy :(import numpy as np) a library for mathematcal operatons and handling arrays.

• pandas :(import pandas as pd) a library for data manipulaton and analysis.

• Matplotlib.pyplot: (import as plt) a library for creatng visualiiaton.

• sklearn ( preproccesing and evaluate model )

**code:**

import numpy as np

import pandas as pd

from sklearn.preprocessing import StandardScaler , Normalizer

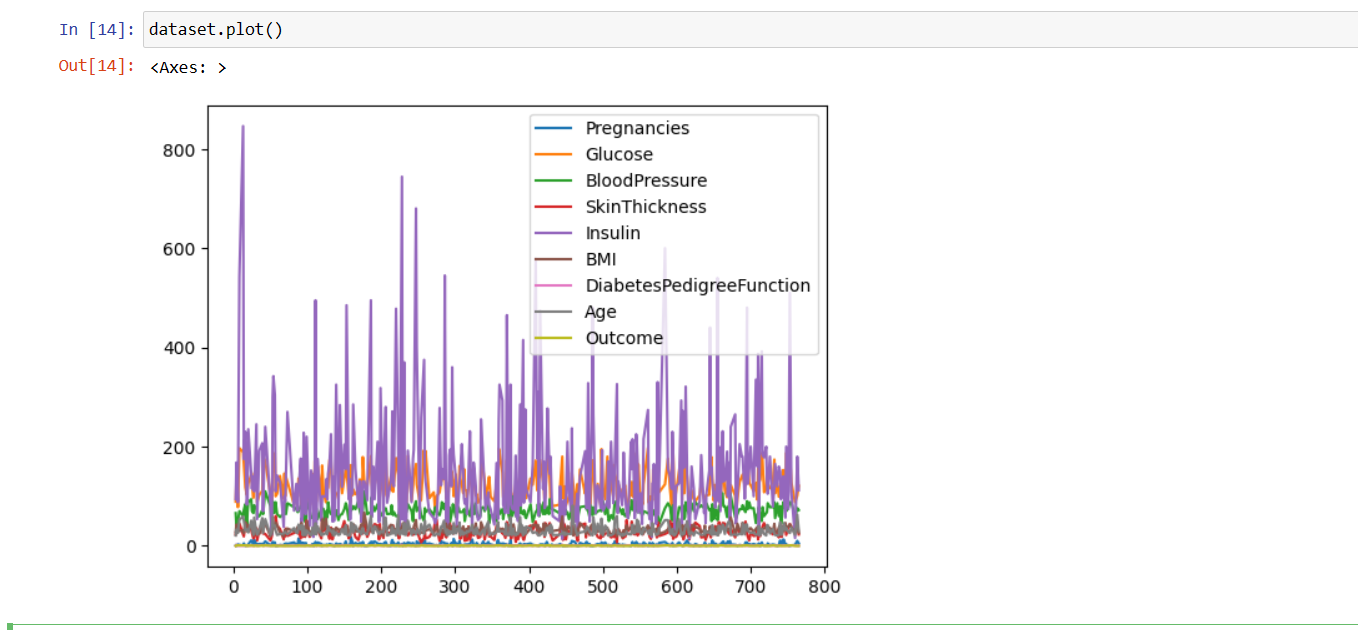
from sklearn.compose import make\_column\_transformer, make\_column\_selector from sklearn.model\_selection import train\_test\_split

**Import the dataset**

dataset = pd.read\_csv('C:/Users/91638/Documents/diabetes.csv')

**visualising of data:**

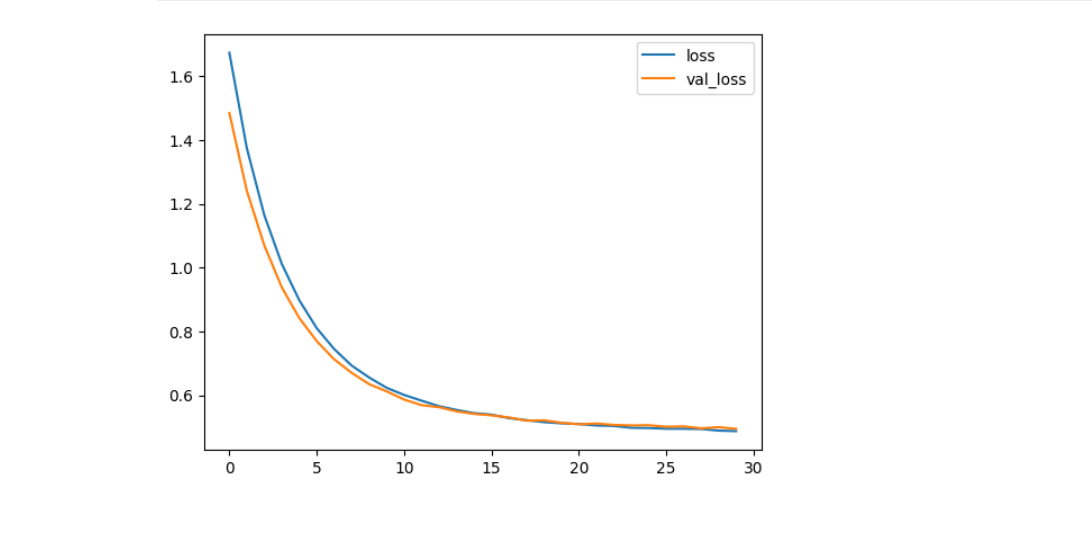
dataset.plot()

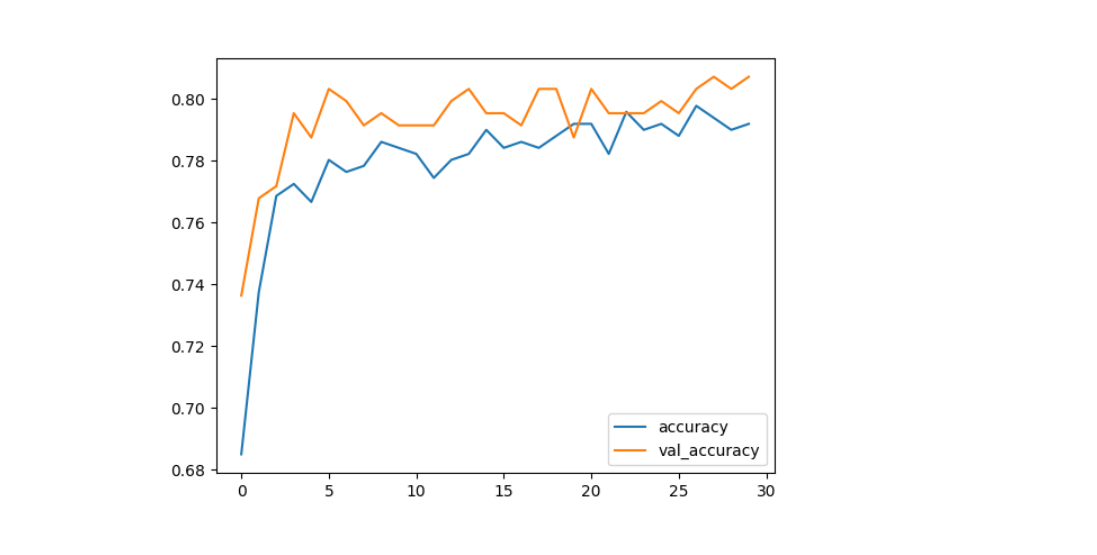


history\_df = pd.DataFrame(history.history)

history\_df.loc[:, ['loss','val\_loss']].plot();

history\_df.loc[:, ['accuracy','val\_accuracy']].plot();





from sklearn.metrics import confusion\_matrix

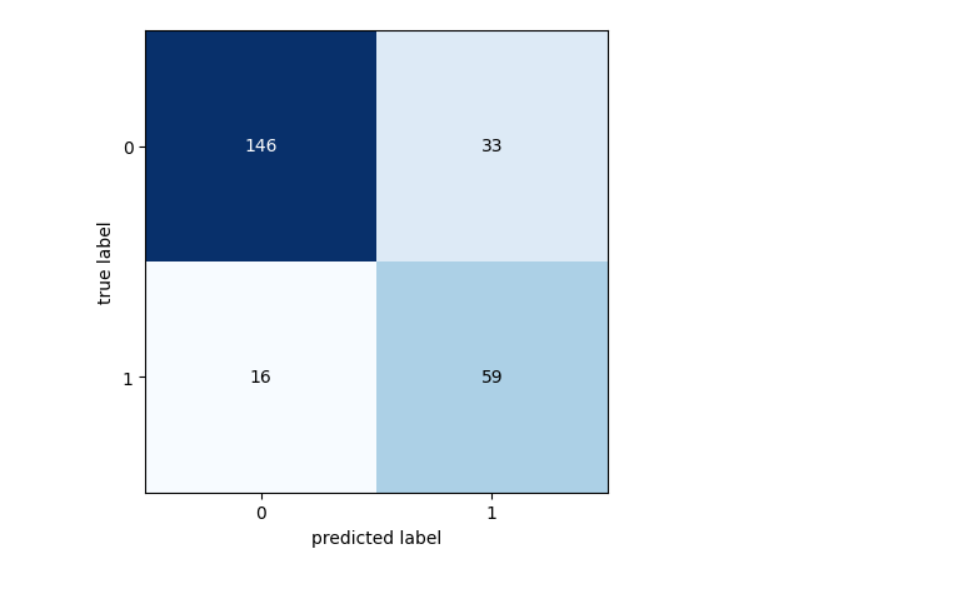
import matplotlib.pyplot as plt

cm = confusion\_matrix(y\_\_predict, y\_\_real)

from mlxtend.plotting import plot\_confusion\_matrix

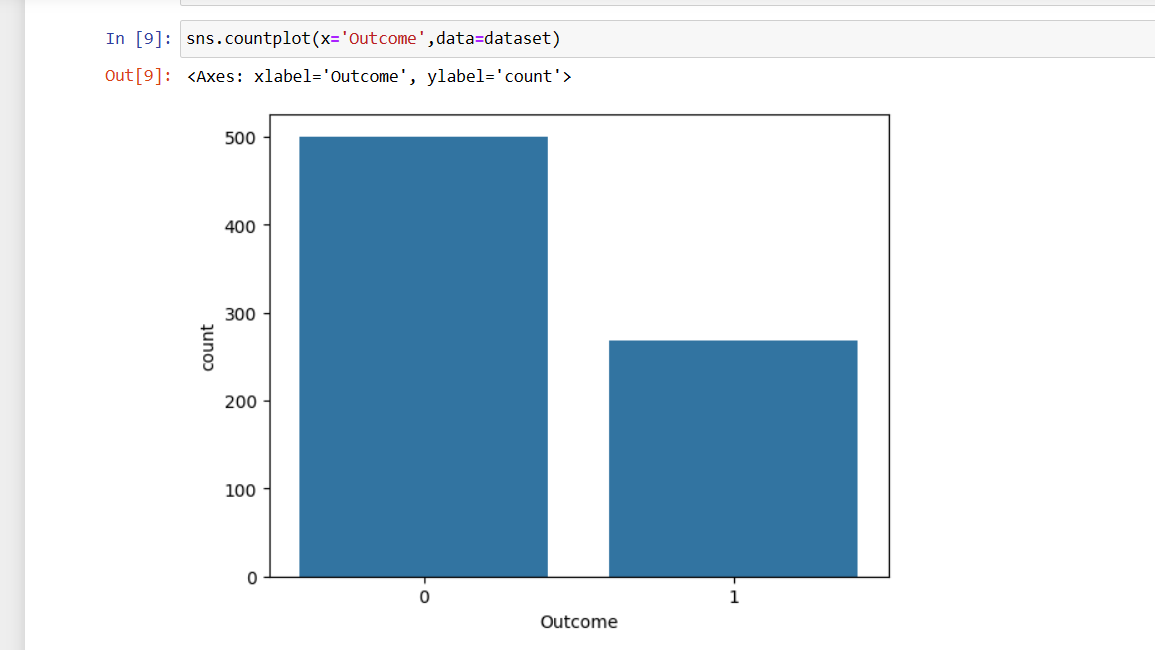
fig, ax = plot\_confusion\_matrix(conf\_mat=cm)

plt.show()



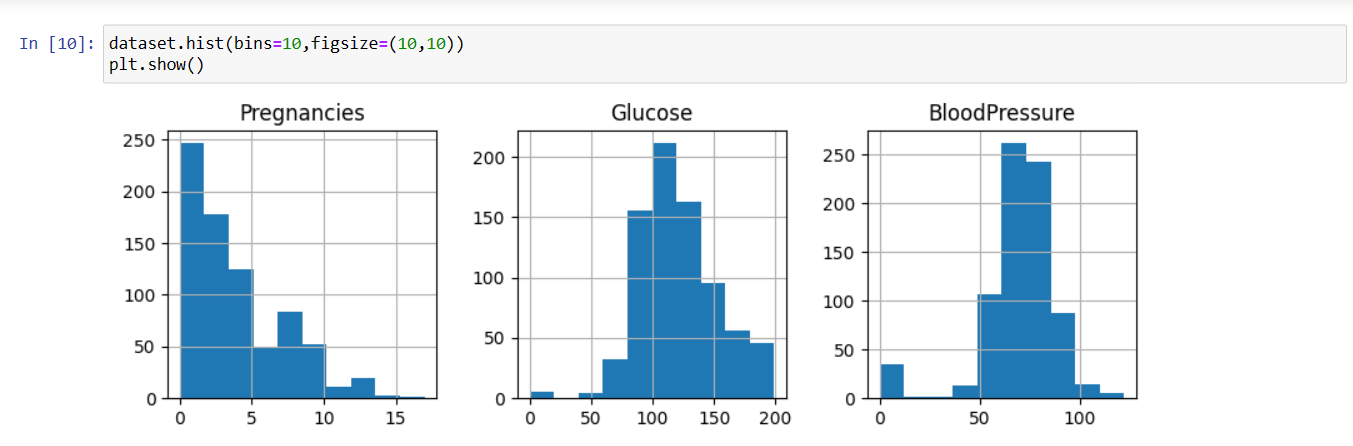
plots summarize the distribution of data and show key statistical measures like the median, quartiles, and potential outliers. They are useful for identifying data spread and skewness.

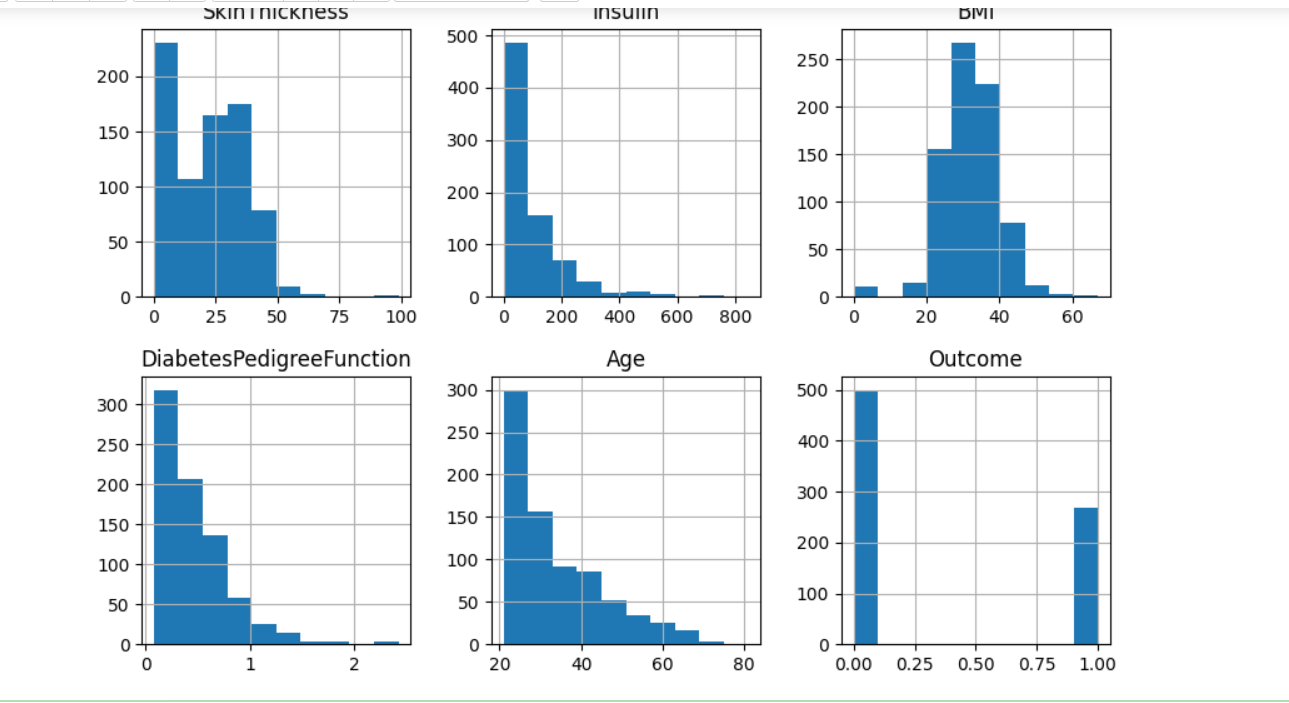
sns.countplot(x='Outcome',data=dataset)



dataset.hist(bins=10,figsize=(10,10))

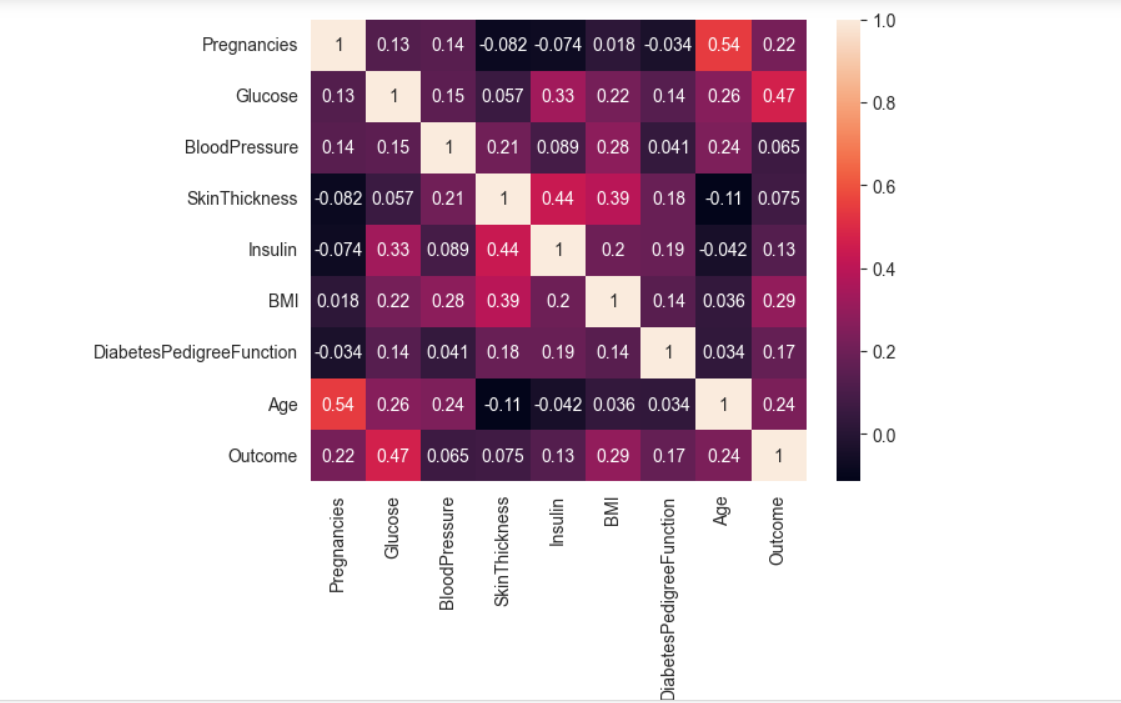
plt.show()





corrmat=dataset.corr()

sns.heatmap(corrmat, annot=True)



A well-crafted data visualization conclusion adds context to the visuals, the patients to better understand and act on the data-driven insights. It helps ensure that your data analysis and visualizations have a meaningful impact on decision-making and problem-solving.

By the data visualization a clear view of the diabetic patients details their blood pressure and their overall details can be visually represented in different graphs.