DATA VISUALIZATION

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

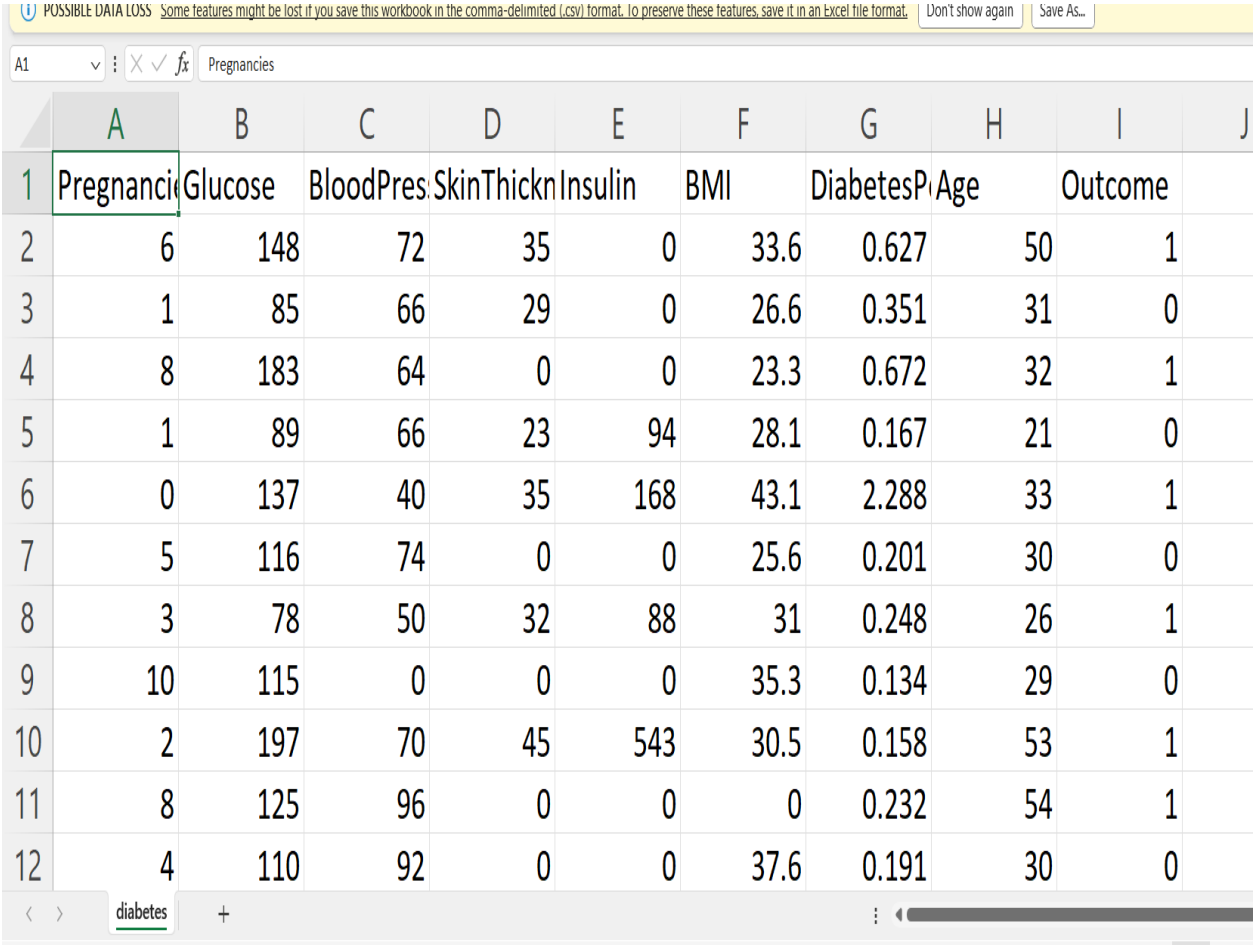
Data visualization is an important step in understanding your dataset when working on an AI-based diabetes detection project. You can use libraries like Matplotlib and Seaborn in Python to create various types of visualizations.

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:



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

A1 Pregnancies

	A	B	C	D	E	F	G	H	I	J
1	Pregnanci	Glucose	BloodPres	SkinThickn	Insulin	BMI	DiabetesP	Age	Outcome	
2	6	148	72	35	0	33.6	0.627	50	1	
3	1	85	66	29	0	26.6	0.351	31	0	
4	8	183	64	0	0	23.3	0.672	32	1	
5	1	89	66	23	94	28.1	0.167	21	0	
6	0	137	40	35	168	43.1	2.288	33	1	
7	5	116	74	0	0	25.6	0.201	30	0	
8	3	78	50	32	88	31	0.248	26	1	
9	10	115	0	0	0	35.3	0.134	29	0	
10	2	197	70	45	543	30.5	0.158	53	1	
11	8	125	96	0	0	0	0.232	54	1	
12	4	110	92	0	0	37.6	0.191	30	0	

diabetes +

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
```

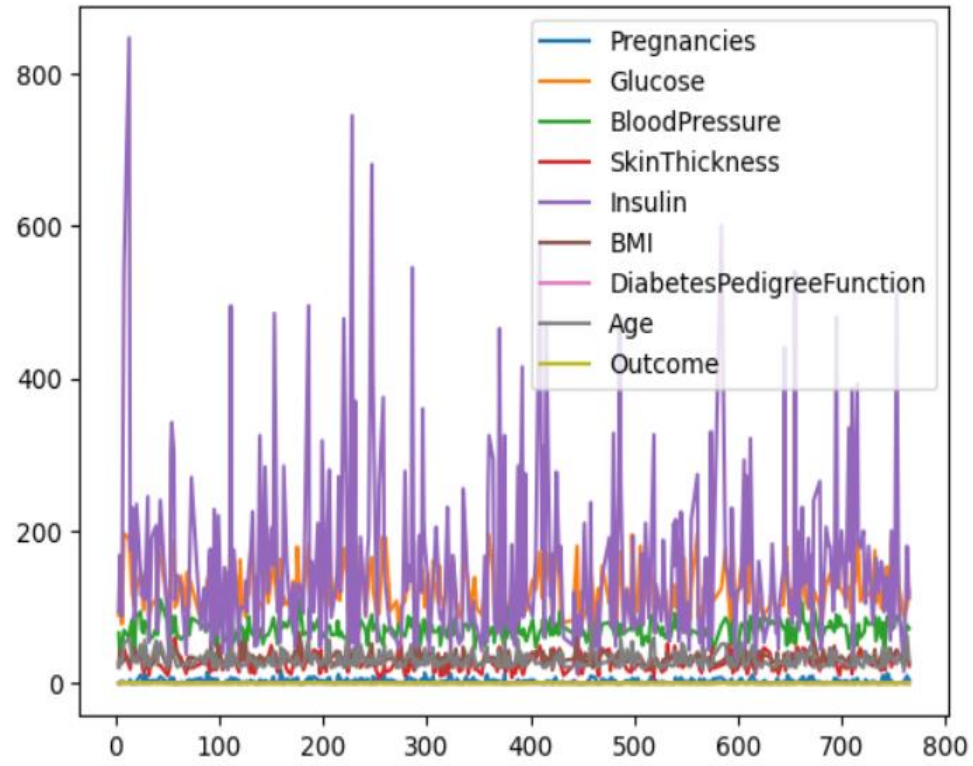
Import the dataset

```
dataset = pd.read_csv('C:/Users/91638/Documents/diabetes.csv')
```

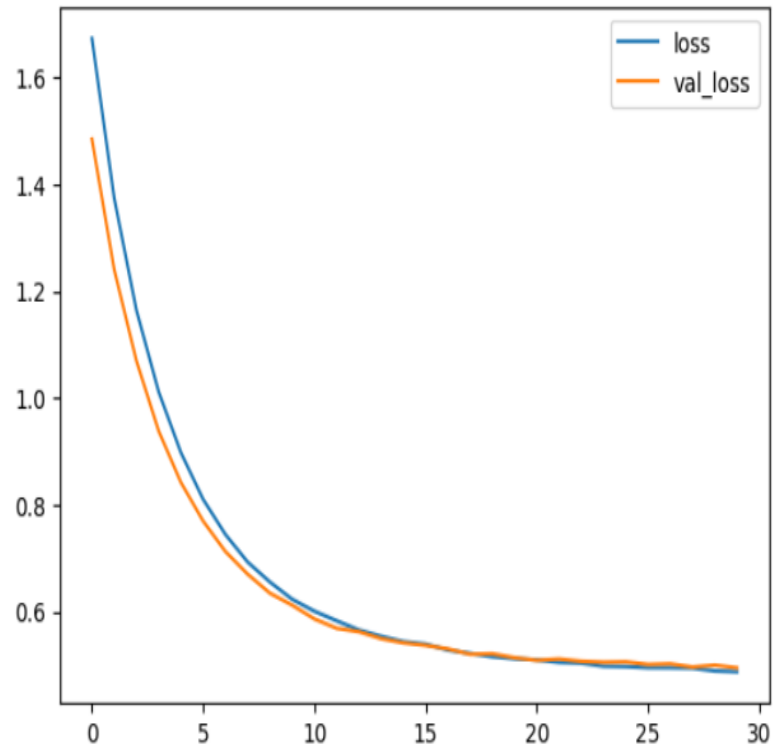
visualising of data:

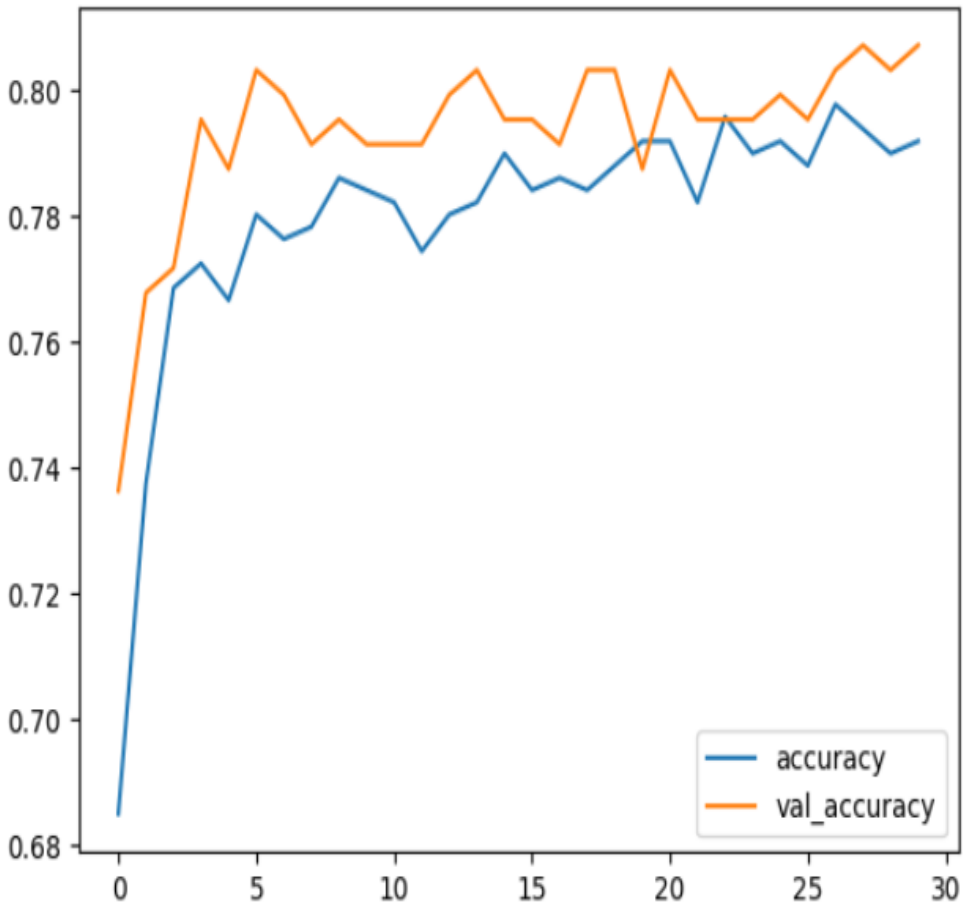
dataset.plot()

```
In [14]: dataset.plot()
Out[14]: <Axes: >
```

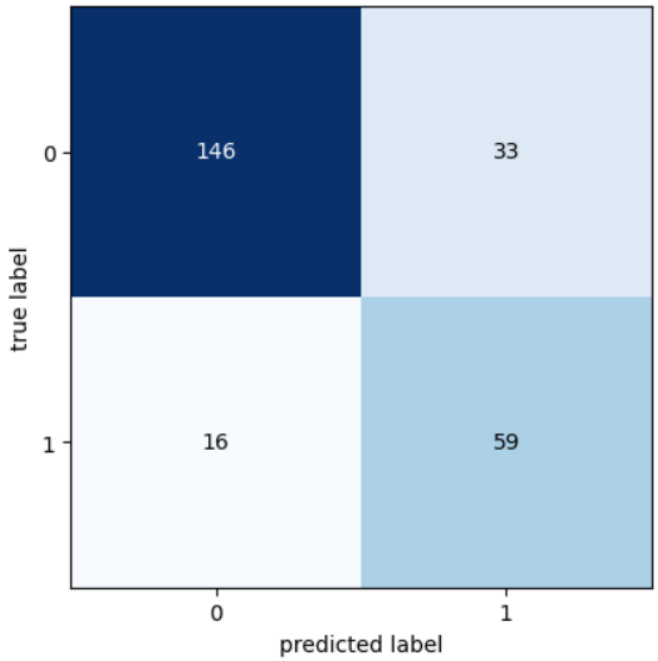


```
history_df = pd.DataFrame(history.history)
history_df.loc[:, ['loss','val_loss']].plot();
history_df.loc[:, ['accuracy','val_accuracy']].plot();
```





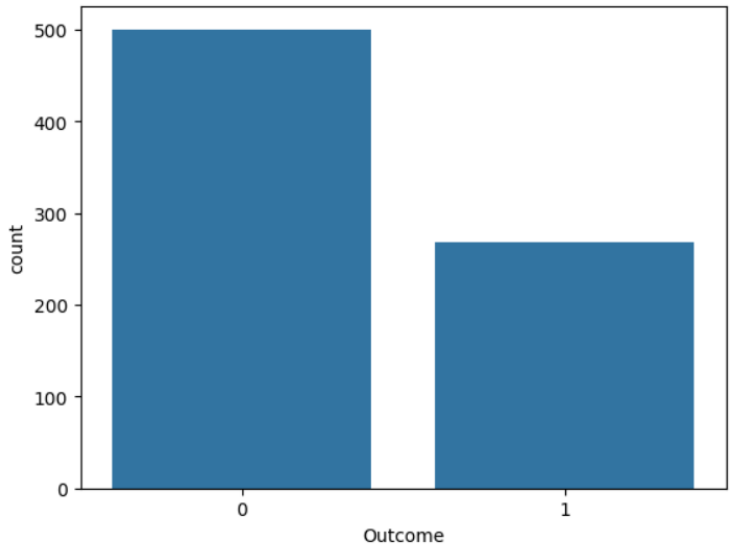

```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
cm = confusion_matrix(y__predict, y__real)
from mlxtend.plotting import plot_confusion_matrix
fig, ax = plot_confusion_matrix(conf_mat=cm)
plt.show()
```



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)

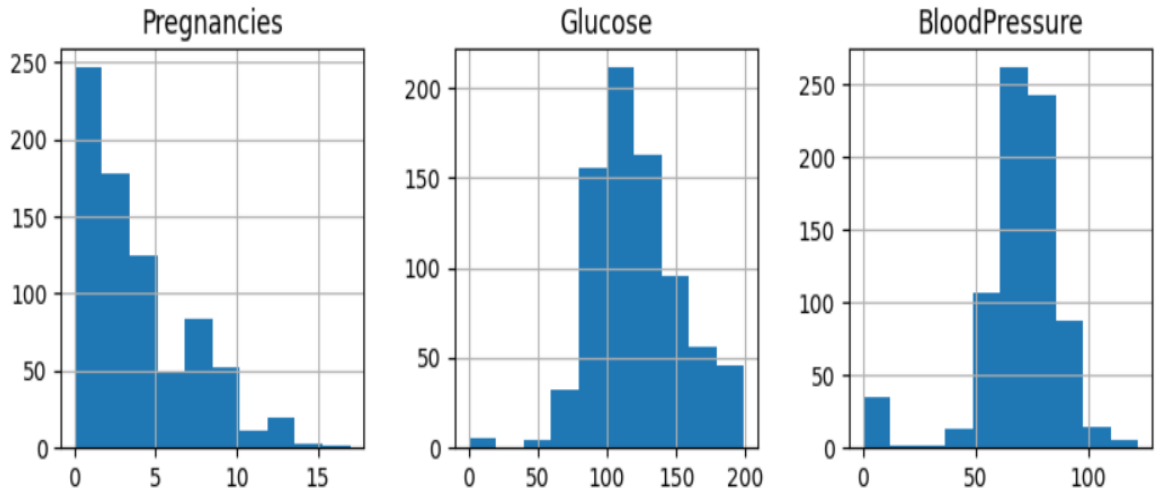
```
In [9]: sns.countplot(x='Outcome',data=dataset)
Out[9]: <Axes: xlabel='Outcome', ylabel='count'>
```

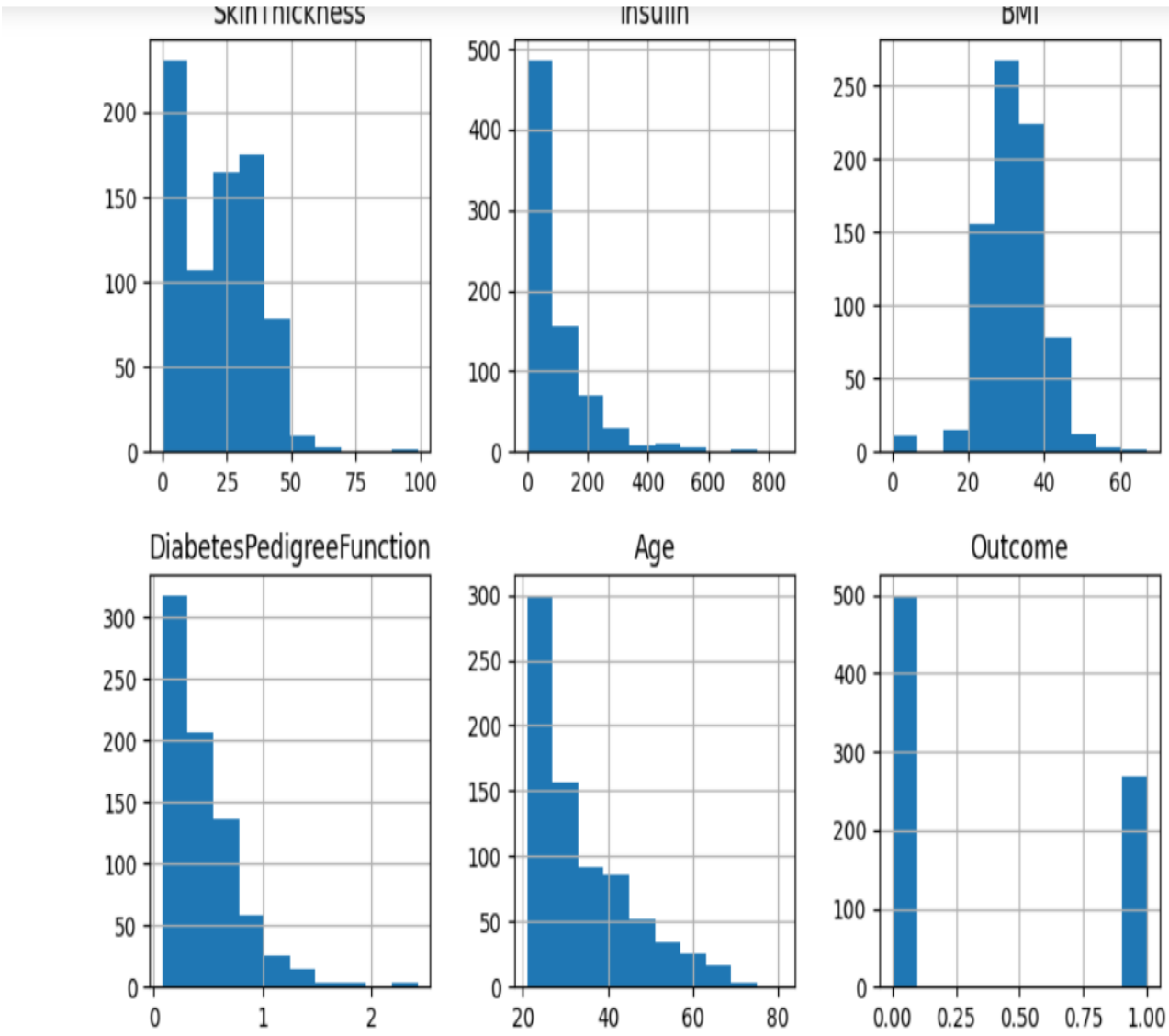


dataset.hist(bins=10,figsize=(10,10))

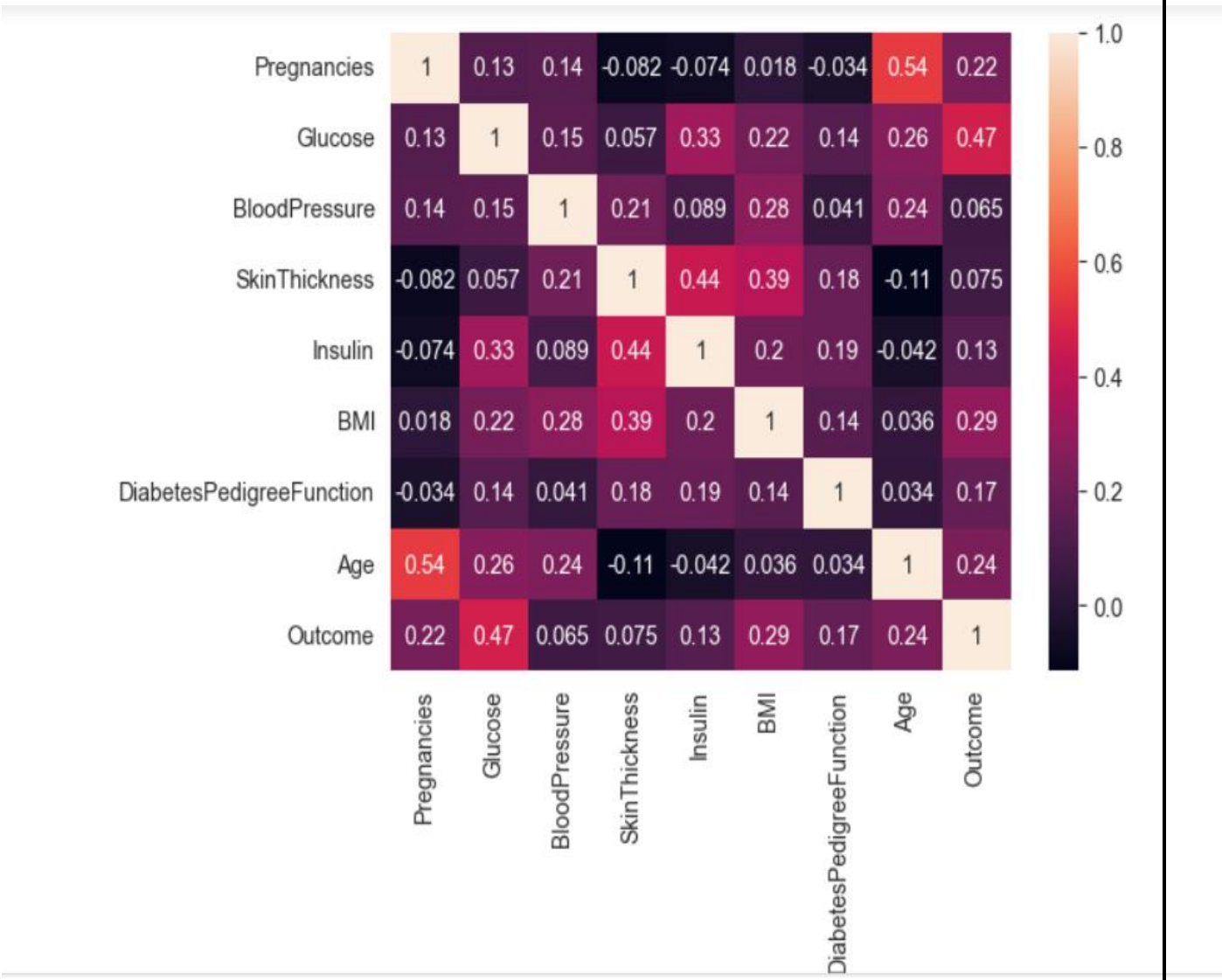
plt.show()

```
In [10]: 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.