AI BASED DIABETES PREDICTION SYSTEM

PHASE 3 Submission Document

Phase 3 : Development Part 1

Topic : AI Based Diabetes prediction system by loading and pre-processing the dataset.

Introduction

- Diabetes is a chronic disease that directly affects the pancreas, and the body is incapable of producing insulin. It is mainly responsible for maintaining the blood glucose level.
- Diabetes is generally characterized by either the body not making enough insulin or being unable to use the insulin that is made as effectively as needed.
- To use machine learning classification methods, that is, decision tree, SVM, Random Forest, Logistic Regression, KNN, and various ensemble techniques, to determine which algorithm produces the best prediction results.
- In this paper, we have employed machine learning and explainable AI techniques to detect diabetes.

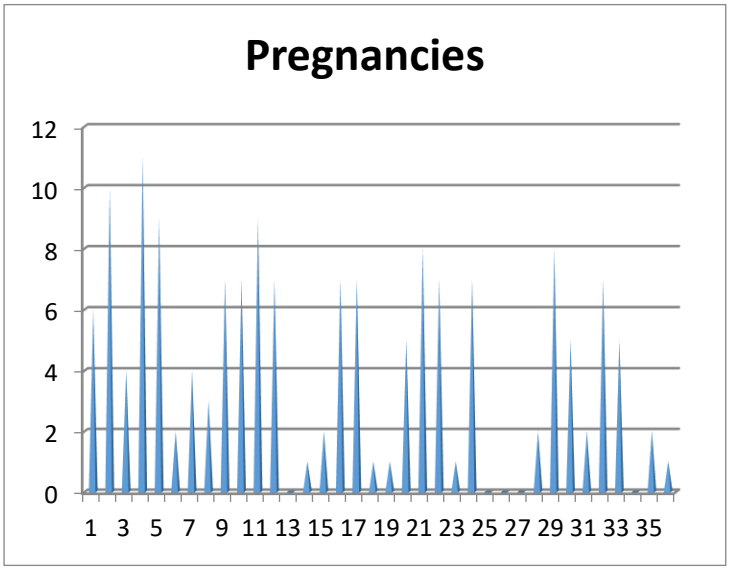
Data source

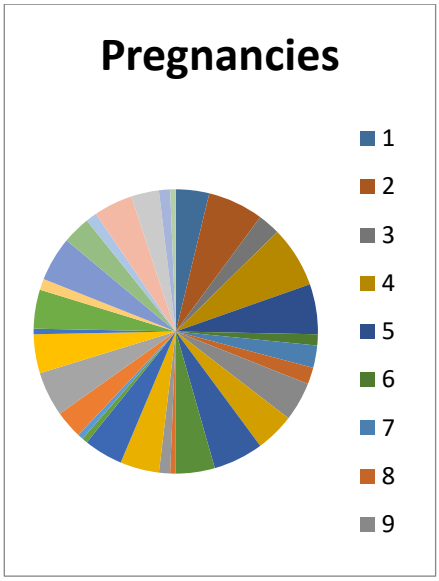
AI-powered diabetes prediction system that leverages machine learning algorithms to analyze medical data and predict the likelihood of an individual developing diabetes. the system aims to provide early risk assessment and personalized preventive measures, allowing individuals to take proactive actions to manage their health.

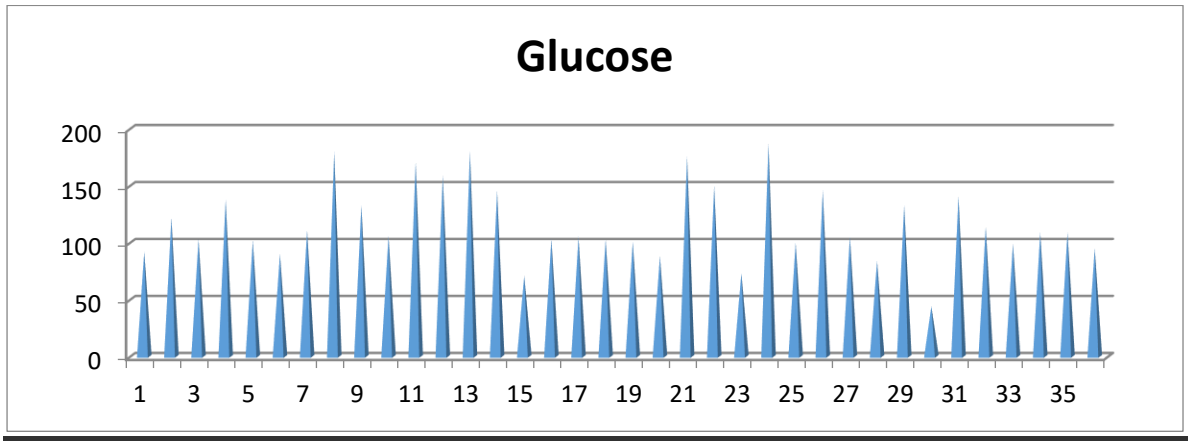
DatasetLink: https://www.kaggle.com/datasets/mathchi/diabetes-data-set

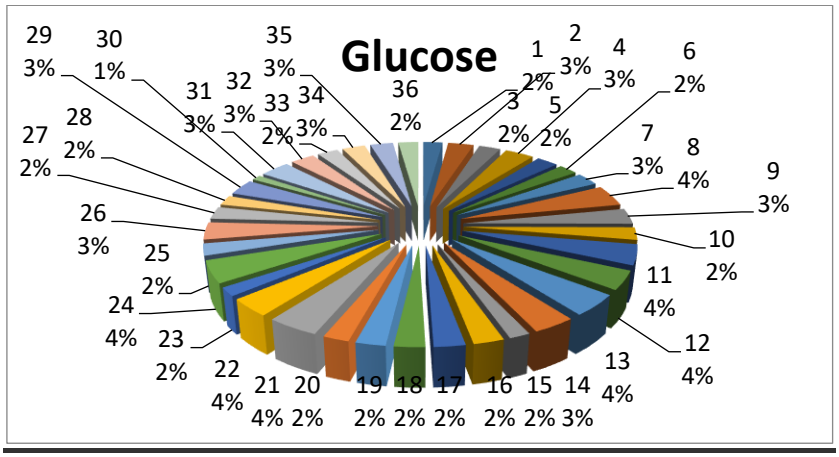
Pregnanci	Glucose	BloodPres	SkinThickr	Insulin	BMI	DiabetesP	Age	Outcome
6	92	92	0	0	19.9	0.188	28	0
10	122	78	31	0	27.6	0.512	45	0
4	103	60	33	192	24	0.966	33	0
11	138	76	0	0	33.2	0.42	35	0
9	102	76	37	0	32.9	0.665	46	1
2	90	68	42	0	38.2	0.503	27	1
4	111	72	47	207	37.1	1.39	56	1
3	180	64	25	70	34	0.271	26	0
7	133	84	0	0	40.2	0.696	37	0
7	106	92	18	0	22.7	0.235	48	0
9	171	110	24	240	45.4	0.721	54	1
7	159	64	0	0	27.4	0.294	40	0
0	180	66	39	0	42	1.893	25	1
1	146	56	0	0	29.7	0.564	29	0
2	71	70	27	0	28	0.586	22	0
7	103	66	32	0	39.1	0.344	31	1
7	105	0	0	0	0	0.305	24	0
1	103	80	11	82	19.4	0.491	22	0
1	101	50	15	36	24.2	0.526	26	0
5	88	66	21	23	24.4	0.342	30	0
8	176	90	34	300	33.7	0.467	58	1
7	150	66	42	342	34.7	0.718	42	0
1	73	50	10	0	23	0.248	21	0
7	187	68	39	304	37.7	0.254	41	1
0	100	88	60	110	46.8	0.962	31	0
0	146	82	0	0	40.5	1.781	44	0
0	105	64	41	142	41.5	0.173	22	0
2	84	0	0	0	0	0.304	21	0
8	133	72	0	0	32.9	0.27	39	1
5	44	62	0	0	25	0.587	36	0
2	141	58	34	128	25.4	0.699	24	0
7	114	66	0	0	32.8	0.258	42	1
5	99	74	27	0	29	0.203	32	0
0	109	88	30	0	32.5	0.855	38	1
2	109	92	0	0	42.7	0.845	54	0
1	95	66	13	38	19.6	0.334	25	0

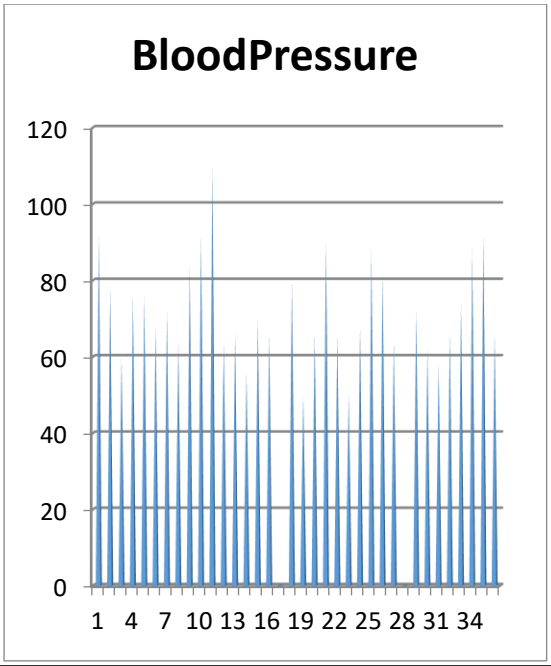
Preprocessing Dataset

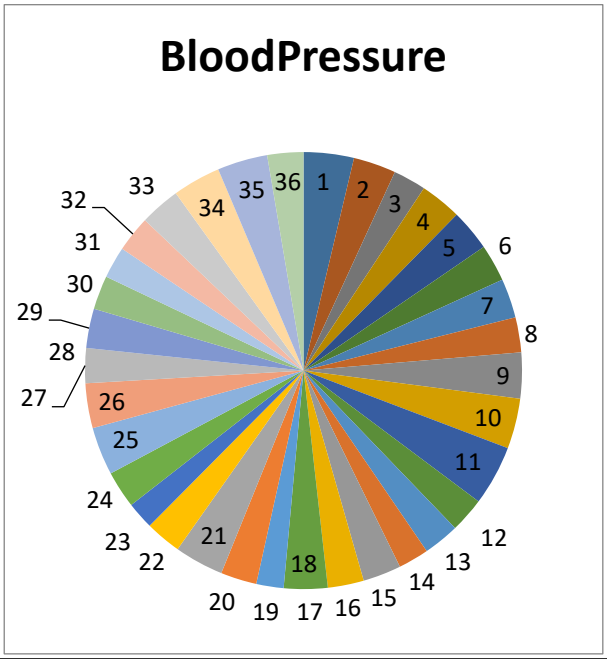


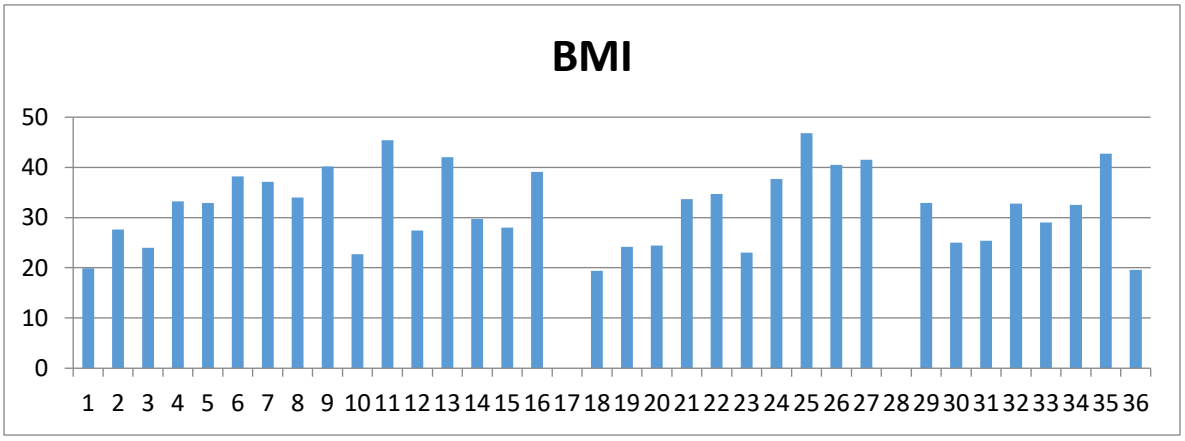


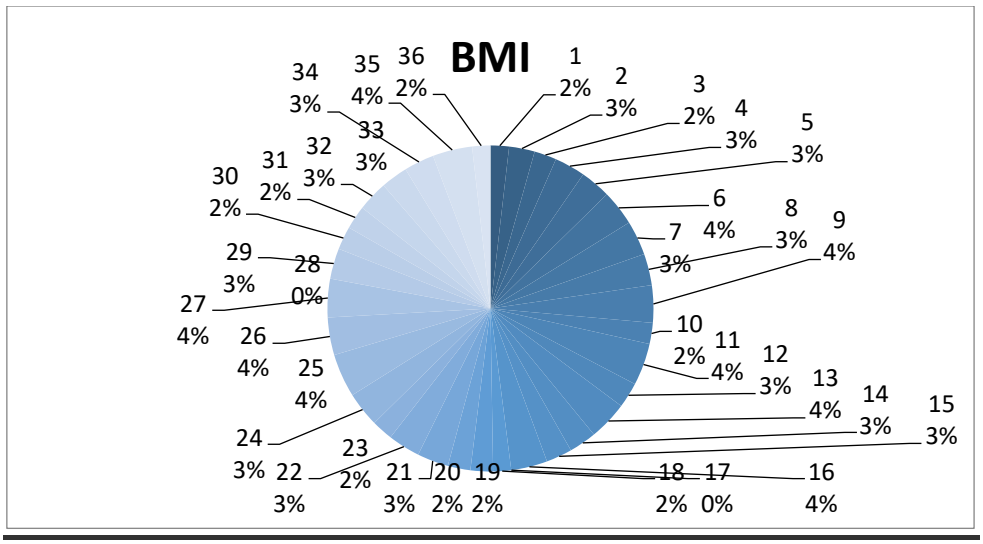





BloodPressure
120
100
80
60
40
20
0
1 4 7 10 13 16 19 22 25 28 31 34


BloodPressure
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36


BMI
50
40
30
20
10
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36


BMI
1 2% 2 3% 3 2% 4 3% 5 3% 6 7 4% 8 3% 9 4% 10 2% 11 4% 12 3% 13 4% 14 3% 15 3% 16 4% 17 0% 18 2% 19 2% 20 2% 21 3% 22 3% 23 2% 24 3% 25 4% 26 4% 27 4% 28 0% 29 3% 30 2% 31 2% 32 3% 33 3% 34 3% 35 4% 36 2%

Data Collection:

We need a dataset containing medical features such as glucose levels, blood pressure, BMI, etc., along with information about whether the individual has diabetes or not.

Data Preprocessing:

The medical data needs to be cleaned, normalized, and prepared for training machine learning models.

Feature Selection:

We will select relevant features that can impact diabetes risk prediction.

Model Selection:

We can experiment with various machine learning algorithms like Logistic Regression, Random Forest, and Gradient Boosting.

Evaluation:

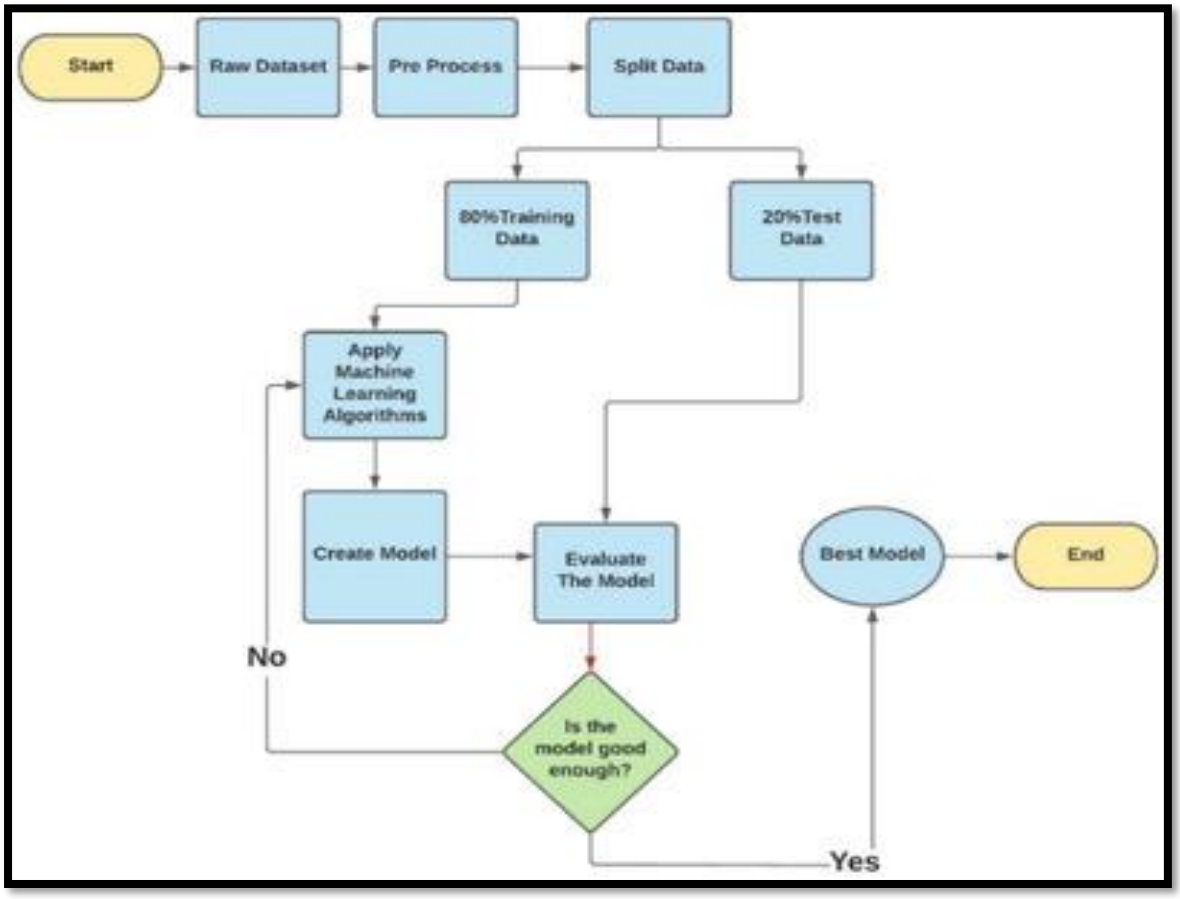
We will evaluate the model's performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.

Iterative Improvement:

We will fine-tune the model parameters and explore techniques like feature engineering to enhance prediction accuracy.

Proposed System

- ❖ This section describes the working procedures and implementation of various machine learning techniques to design the proposed automatic diabetes prediction system.
- ❖ First, the dataset was collected and preprocessed to remove the necessary discrepancies from the dataset
- ❖ Then the dataset was separated into the training set and test set using the holdout validation technique.
- ❖ Next, different classification algorithms were applied to find the best classification algorithm for this dataset.



Program

AI Based Diabetes Prediction System

```
# Import libraries
import numpy as np # for linear algebra
import pandas as pd # for data processing, CSV file I/O (e.g. pd.read_cs
v)
import seaborn as sns # for data visualization
import matplotlib.pyplot as plt # to plot data visualization charts
from collections import Counter
import os

# Modeling Libraries
from sklearn.metrics import confusion_matrix, accuracy_score, precisio
n_score
from sklearn.preprocessing import QuantileTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClass
ifier, GradientBoostingClassifier
from sklearn.model_selection import GridSearchCV, cross_val_score,
StratifiedKFold, learning_curve, train_test_split
from sklearn.svm import SVC
```

Importing the Dataset

```
# Importing the dataset from Kaggle
data = pd.read_csv("../input/pima-indians-diabetes-
database/diabetes.csv")

# First step is getting familiar with the structure of the dataset
data.info()
```

Output

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Code

```
# Showing the top 5 rows of the dataset
data.head()
```

Output

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35	30.5	33.6	0.627	50	1
1	1	85.0	66.0	29	30.5	26.6	0.351	31	0
2	8	183.0	64.0	23	30.5	23.3	0.672	32	1
3	1	89.0	66.0	23	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35	168.0	43.1	2.288	33	1

Filling the Missing Values Code

```
# Exploring the missing values in the diabetes dataset
data.isnull().sum()
```

Output

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Code

```
# Replacing 0 values with the mean of that column
# Replacing 0 values of Glucose
```

```
data['Glucose'] = data['Glucose'].replace(0, data['Glucose'].median())

# Filling 0 values of Blood Pressure

data['BloodPressure'] = data['BloodPressure'].replace(0, data['BloodPress
ure'].median())

# Replacing 0 values in BMI

data['BMI'] = data['BMI'].replace(0, data['BMI'].mean())

# Replacing the missing values of Insulin and SkinThickness

data['SkinThickness'] = data['SkinThickness'].replace(0, data['SkinThick
ness'].mean())

data['Insulin'] = data['Insulin'].replace(0, data['Insulin'].mean())

data.head()
```

Output

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35.000000	79.799479	33.6	0.627	50	1
1	1	85	66	29.000000	79.799479	26.6	0.351	31	0
2	8	183	64	20.536458	79.799479	23.3	0.672	32	1
3	1	89	66	23.000000	94.000000	28.1	0.167	21	0
4	0	137	40	35.000000	168.000000	43.1	2.288	33	1

Code

```
# Reviewing the dataset statistics

data.describe()
```

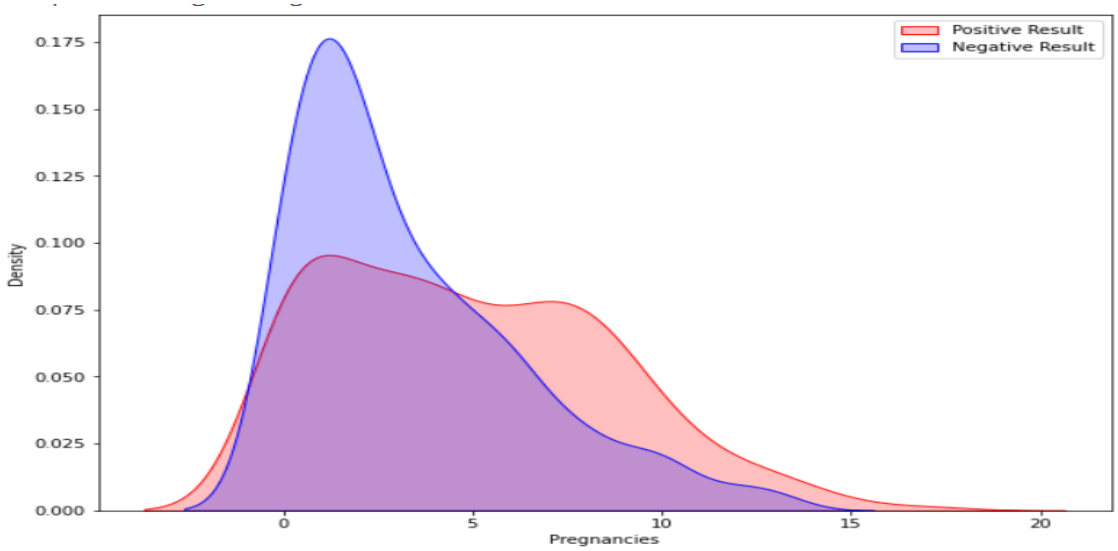
Output

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	121.656250	72.386719	26.606479	118.660163	32.450805	0.471876	33.240885	0.348958
std	3.369578	30.438286	12.096642	9.631241	93.080358	6.875374	0.331329	11.760232	0.476951
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	0.078000	21.000000	0.000000
25%	1.000000	99.750000	64.000000	20.536458	79.799479	27.500000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	79.799479	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Pregnancy Code

```
# Exploring Pregnancy and target variables together

  plt.figure(figsize = (10, 8))

# Plotting density function graph of the pregnancies and the target variab
le

kde = sns.kdeplot(data["Pregnancies"][data["Outcome"] == 1], color = "
Red", shade = True)

kde = sns.kdeplot(data["Pregnancies"][data["Outcome"] == 0], ax = kde,
 color = "Blue", shade= True)

kde.set_xlabel("Pregnancies")

kde.set_ylabel("Density")

kde.legend(["Positive Result", "Negative Result"])
```

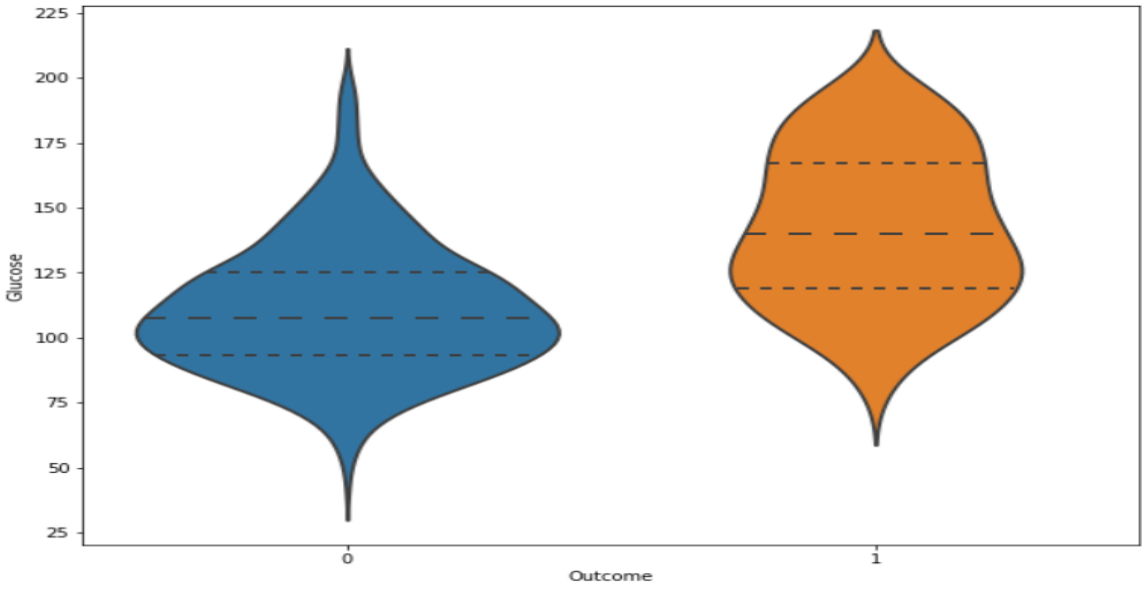
Output



Glucose code

```
# Exploring the Glucose and the Target variables together
plt.figure(figsize = (10, 8))
sns.violinplot(data = data, x = "Outcome", y = "Glucose",
               split = True, inner = "quart", linewidth = 2)
```

Output

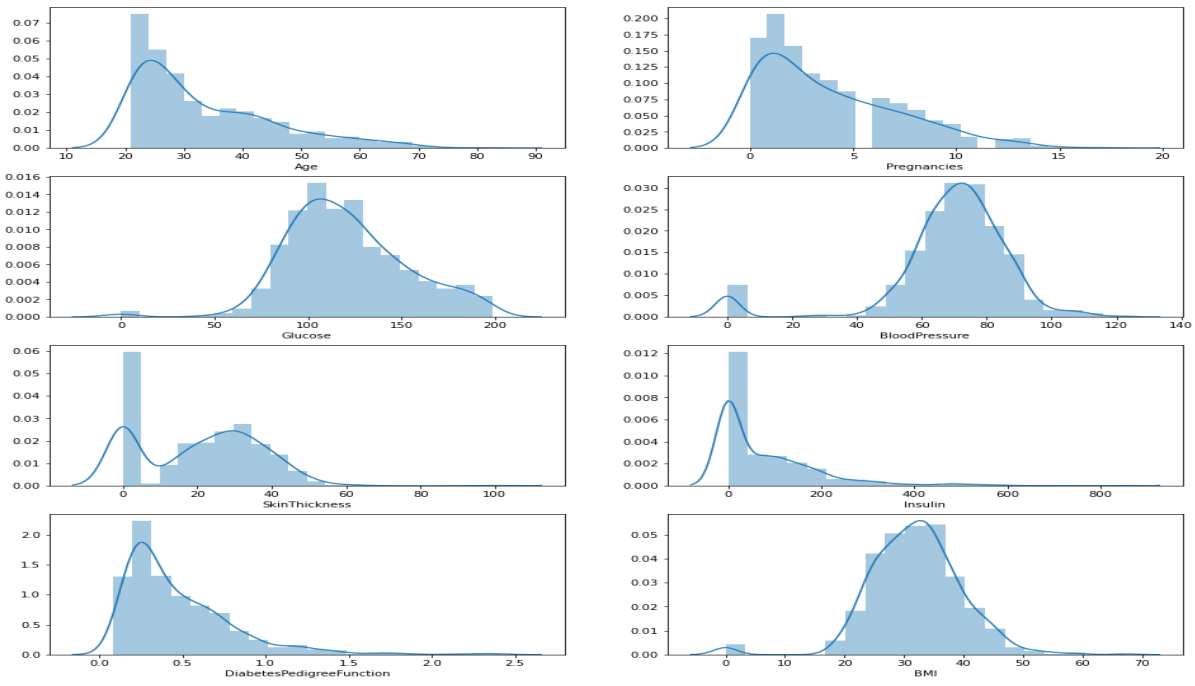


Code

```
# Histogram and density graphs of all variables were accessed.
fig, ax = plt.subplots(4,2, figsize=(16,16))
sns.distplot(df.Age, bins = 20, ax=ax[0,0])
sns.distplot(df.Pregnancies, bins = 20, ax=ax[0,1])
sns.distplot(df.Glucose, bins = 20, ax=ax[1,0])
sns.distplot(df.BloodPressure, bins = 20, ax=ax[1,1])
sns.distplot(df.SkinThickness, bins = 20, ax=ax[2,0])
sns.distplot(df.Insulin, bins = 20, ax=ax[2,1])
sns.distplot(df.DiabetesPedigreeFunction, bins = 20, ax=ax[3,0])
sns.distplot(df.BMI, bins = 20, ax=ax[3,1])
```

Output

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f77b83d5950>
```



Future for diabetes prediction system

- Researchers are motivated to create a Machine Learning methodology that can predict diabetes in the future.
- Exploiting Machine Learning Algorithms (MLA) is essential if healthcare professionals are able to identify diseases more effectively.

Conclusions

1. One of the risks during pregnancy is diabetes. It will have to be diagnosed to avoid problems.
2. An increase in glucose levels is strongly correlated to a rise in diabetes.
3. Diabetes can be a reason for reducing life expectancy and quality.
4. Predicting this chronic disorder earlier can reduce the risk and complications of many diseases in the long run.