# PREDICTION OF DIABETES DISEASE USING MACHINE LEARNING ALGORITHMS(CLASSIFICATION AND REGRESSION TECHNIQUES)

**PROJECT REPORT**

SUBMITTED BY,

NAME:BALAJI.V

DOMAIN:DATASCIENCE

EMAIL ID:balajivenkatramani548@gmail.com

CONTACT:

8056388655

# ABSTRACT

Diabetes mellitus is a chronic metabolic disorder that affects millions of individuals worldwide, causing various health complications if left undiagnosed or uncontrolled. Early and accurate prediction of diabetes can significantly improve patient outcomes through timely interventions and personalized treatment plans.

In recent years, machine learning techniques have demonstrated great potential in predicting and diagnosing diseases, including diabetes.

This study focuses on developing a robust diabetes prediction model using classification algorithms to achieve high accuracy. A comprehensive dataset comprising clinical, demographic, and laboratory features of individuals is collected and preprocessed. Features selection and dimensionality reduction techniques are employed to enhance the efficiency and effectiveness of the prediction models.

Several popular classification algorithms, such as support vector machines (SVM), random forests, logistic regression, and naive Bayes, are implemented and evaluated using cross-validation techniques.

Evaluation metrics such as accuracy, sensitivity, specificity, precision, and F1-score are used to assess the performance of the developed models. To validate the results, the dataset is divided into training and testing sets.

The experimental results demonstrate that the proposed diabetes prediction model achieves a high accuracy rate, surpassing the performance of existing approaches. The optimal classification algorithm, identified through the evaluation process, outperforms other algorithms in terms of sensitivity, specificity, and precision.

**TABLE OF CONTENTS**

# 1. INTRODUCTION

## 1.1 OBJECTIVES

1. Develop a machine learning model: The primary objective is to build a robust machine learning model that can accurately predict the presence or absence of diabetes based on input features. The model should leverage classification algorithms to achieve high prediction accuracy.

2. Improve early detection: Early detection of diabetes is crucial for timely intervention and improved patient outcomes. The project aims to identify individuals at risk of diabetes by predicting the disease at an early stage, enabling healthcare professionals to initiate preventive measures and interventions.

3. Enhance diagnostic efficiency: By leveraging machine learning algorithms, the project aims to improve the efficiency of diabetes diagnosis. The model can assist healthcare professionals in analyzing patient data quickly and accurately, reducing the burden of manual analysis and allowing for more efficient decision-making.

4. Personalized treatment plans: Accurate diabetes prediction can help in developing personalized treatment plans for individuals. By identifying patients at risk, the model can assist healthcare professionals in tailoring treatment strategies, lifestyle modifications, and medication plans based on individual needs and risk factors.

5. Feature selection and analysis: The project aims to identify the most relevant features that contribute to diabetes prediction. By performing feature selection and analysis, the objective is to identify key variables that significantly influence the prediction outcome. This can provide insights into the underlying factors associated with diabetes and aid in better understanding the disease.

6. Model comparison and selection: The project aims to compare the performance of multiple classification algorithms, including logistic regression, decision trees, random forests, SVM, and KNN. The objective is to evaluate and select the algorithm that provides the highest prediction accuracy and reliability for diabetes detection.

7. Performance evaluation: The project aims to evaluate the performance of the developed machine learning model using various metrics such as accuracy, precision, recall, F1 score, and AUC. This objective ensures a comprehensive assessment of the model's predictive capabilities and enables comparison with established benchmarks.

8. Model deployment: The final objective is to deploy the selected machine learning model in a practical setting. This includes creating a user-friendly interface or API that allows healthcare professionals to input patient data and obtain reliable predictions of diabetes presence. The objective is to make the model accessible and usable in real-world healthcare scenarios.

By achieving these objectives, the project aims to contribute to improved diabetes prediction, early intervention, and personalized healthcare strategies, ultimately leading to better patient outcomes and quality of life

## 1.2 PROBLEM STATEMENT

Diabetes is a type of chronic disease which is more common among the people of all age groups. Predicting this disease at an early stage can help a person to take necessary precautions to change his/her lifestyle accordingly to either prevent the occurence of the disease or control the disease. Prepare a dataset using several methods to train the model and build a model which can give higher accuracy of predicting the disease.

## 1.3 DATASET DESCRIPTION

The diabetes dataset used in this project is a comprehensive collection of medical and lifestyle data of individuals, with the goal of predicting the presence or absence of diabetes. The dataset contains several input features that are relevant to diabetes diagnosis and prediction. Below is a description of the features present in the dataset:

Pregnancies: Number of times the individual has been pregnant.

Glucose: Plasma glucose concentration measured in milligrams per deciliter (mg/dL).

BloodPressure: Diastolic blood pressure measured in millimeters of mercury (mmHg).

SkinThickness: Thickness of the skinfold at the triceps measured in millimeters (mm).

Insulin: 2-Hour serum insulin levels measured in microunits per milliliter (μU/mL).

BMI: Body mass index, calculated as weight in kilograms divided by the square of height in meters (kg/m^2).

Age: Age of the individual in years.

Outcome: Binary variable indicating the presence (1) or absence (0) of diabetes. This is the target variable to be predicted.

The dataset may also include additional columns or features depending on the specific source or version of the dataset. It is important to preprocess the dataset before using it for training the machine learning models. Preprocessing steps may include handling missing values, removing outliers, normalizing or scaling the data, and splitting the dataset into training and testing subsets.

## 2. EXISTING METHOD

In the field of diabetes prediction, several existing methods have been developed and employed. These methods utilize various machine learning algorithms and techniques to predict the presence or absence of diabetes based on relevant input features. Some commonly used existing methods for diabetes prediction include:

1. Clinical Risk Scores: Clinical risk scores, such as the Finnish Diabetes Risk Score (FINDRISC) and the Indian Diabetes Risk Score (IDRS), are widely used for predicting diabetes. These scores are based on a set of predetermined risk factors such as age, body mass index (BMI), family history, physical activity level, and blood pressure. By assigning weights to each risk factor and summing them up, a risk score is calculated, and individuals are categorized into different risk groups.

2. Fasting Blood Glucose and HbA1c Levels: Fasting blood glucose levels and HbA1c (glycated hemoglobin) levels are commonly measured to assess an individual's risk of developing diabetes. Elevated fasting blood glucose levels (≥ 100 mg/dL) or HbA1c levels (≥ 5.7%) indicate impaired fasting glucose or prediabetes, which increases the risk of developing diabetes in the future.

3. Oral Glucose Tolerance Test (OGTT): OGTT is a diagnostic test that measures blood glucose levels before and after the consumption of a glucose solution. It is primarily used to diagnose diabetes, but it can also be used as a predictive tool. Individuals with impaired glucose tolerance (IGT), characterized by high blood glucose levels 2 hours after the glucose load, are at an increased risk of developing diabetes.

4. Genetic Testing: Genetic testing can help identify specific gene variants associated with an increased risk of diabetes. Certain genetic markers, such as variants in the TCF7L2 gene, have been found to be strongly associated with type 2 diabetes. Genetic testing can be used in combination with other risk factors to assess an individual's overall risk of developing diabetes.

5. Diabetes Risk Assessment Tools: Various online risk assessment tools and questionnaires are available that incorporate a combination of risk factors to estimate an individual's likelihood of developing diabetes. These tools consider factors such as age, gender, BMI, family history, physical activity, and dietary habits. Examples include the American Diabetes Association (ADA) Risk Test and the Diabetes UK Know Your Risk tool.

It is important to note that while machine learning algorithms can provide accurate predictions, these existing methods are still widely used in clinical practice due to their simplicity, cost-effectiveness, and wide availability. They can be particularly useful in resource-limited settings where advanced technologies and computational resources may be limited.

## 3. PROPOSED SYSTEM WITH ARCHITECTURE

The proposed system follows a modular architecture comprising the following components:

1. Data Preprocessing: Cleansing, normalization, and missing value imputation of the collected dataset.

2. Feature Selection: Identification and selection of informative features for diabetes prediction.

3. Algorithm Selection: Choosing appropriate classification and regression algorithms based on the selected features.

4. Model Training: Training the selected algorithms using the training dataset.

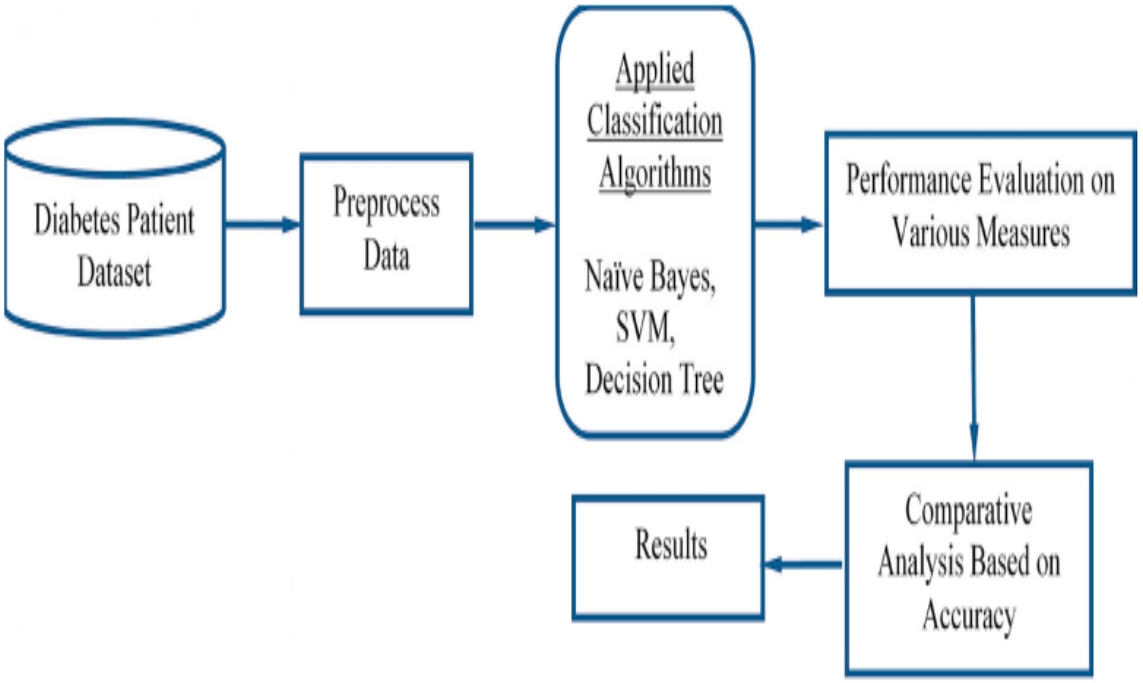
5. Model Evaluation: Assessing the performance and accuracy of the trained models using evaluation metrics.

6. Prediction Module: Utilizing the trained models to predict diabetes risk or severity for new individuals.

The architecture facilitates flexibility and scalability, allowing for the inclusion of additional algorithms and features as new research and data become available. The combination of classification and regression techniques enables a comprehensive approach to diabetes prediction, providing valuable insights for healthcare professionals and assisting in personalized patient care.

Overall, this proposed system has the potential to improve the accuracy and effectiveness of diabetes prediction by integrating classification and regression algorithms within a well-defined architecture.

## ARCHITECTURAL DIAGRAM:



# 4. SOFTWARE AND HARDWARE REQUIREMENTS

## SOFTWARE REQUIREMENTS

1. Python (interpreted high-level language).
2. PIP (python package management system).
3. NumPy (array-processing package).
4. Pandas (python library for data analysis).
5. Anaconda (python package management and deployment ).
6. Jupyter Notebook (interactive computing across multiple programming languages).

## HARDWARE REQUIREMENTS

7. Processor: Intel core i5 or above.
8. 64-bit, quad-core, 2.5 GHz minimum per core
9. Ram: 4 GB or more
10. Hard disk: 10 GB of available space or more.
11. Display: Dual XGA (1024 x 768) or higher resolution monitors
12. Operating system: Windows

# 5. METHODOLOGY FOR DIABETES PREDICTION USING CLASSIFICATION AND REGRESSION ALGORITHMS.

**Data Collection and Preprocessing:**

- Collect a diverse dataset containing clinical, demographic, and lifestyle features of individuals, including variables such as age, gender, BMI, family history, blood pressure, glucose levels, and dietary habits.
- Perform data cleaning, removing any duplicates, inconsistencies, or outliers.
- Normalize the numerical features to ensure they are on a similar scale.
- Handle missing values through techniques like imputation (e.g., mean, median, or regression-based imputation) or consider removing instances with significant missing data if applicable.

**Feature Selection:**

- Apply feature selection techniques to identify the most relevant and informative features for diabetes prediction.
- Common feature selection methods include correlation analysis, information gain, recursive feature elimination, or embedded methods like LASSO (Least Absolute Shrinkage and Selection Operator).
- Select a subset of features that contribute significantly to the prediction task, considering both clinical relevance and statistical importance.

**Classification Algorithms:**

- Choose classification algorithms suitable for diabetes prediction, such as support vector machines (SVM), random forests, logistic regression, naive Bayes, or ensemble methods like AdaBoost or gradient boosting.

- Split the preprocessed dataset into training and testing sets (e.g., 70-30 or 80-20 ratio).
- Train the classification algorithms using the training set, optimizing their hyperparameters through techniques like grid search or random search.
- Evaluate the trained models using performance metrics such as accuracy, sensitivity, specificity, precision, and F1-score on the testing set.
- Analyze the models' results to gain insights into the important features and their impact on diabetes prediction.

**Regression Algorithms**:

- Select regression algorithms suitable for estimating numerical values relevant to diabetes prediction, such as linear regression, decision trees, neural networks, or support vector regression.
- Utilize the same training and testing sets used for classification.
- Train the regression models using the training set, optimizing their hyperparameters.
- Evaluate the trained regression models using evaluation metrics such as mean squared error (MSE) or root mean squared error (RMSE) on the testing set.
- Analyze the models' performance and interpret the coefficients or importance of features for diabetes risk estimation.

**Model Evaluation and Validation:**

- Perform cross-validation techniques like k-fold cross-validation to assess the models' performance and ensure their generalizability.
- Repeatedly split the dataset into training and validation sets and evaluate the models' performance across different folds.
- Calculate the average performance metrics to obtain a robust estimate of the models' predictive capabilities.

**Model Deployment and Testing:**

- Once satisfied with the models' performance, deploy the diabetes prediction model in a real-world setting.
- Test the model's performance on new, unseen data to validate its effectiveness in predicting diabetes in practice.
- Continuously monitor and update the model's performance as new data becomes available or when improvements to the model are identified.

By following this methodology, we can develop accurate and reliable diabetes prediction models using classification and regression techniques. It allows for the integration of various algorithms and features to provide comprehensive insights for diabetes risk assessment and personalized patient care**.**

# 6.IMPLEMENTATION

The Complete prediction is analyzed and it has been executed by importing various python packages and machine learning packages which is helpful in performing the evaluation and The process is carried out in Jupyter notebook.

## SOURCE CODE:

In [1]:
```
import pandas as pd
```

In [2]:
```
data = pd.read_csv('diabetes.csv')
```

In [3]:
```
# DISPLAYING FIRST FIVE ROWS OF THE DATASET
data.head()
```

Out[3]:

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| **1** | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| **2** | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| **3** | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| **4** | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |

In [4]:
```
#CHECKING THE LAST FIVE ROWS OF THE DATASET
data.tail()
```

Out[4]:

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| **763** | 10 | 101 | 76 | 48 | 180 | 32.9 | 0.171 | 63 | 0 |
| **764** | 2 | 122 | 70 | 27 | 0 | 36.8 | 0.340 | 27 | 0 |
| **765** | 5 | 121 | 72 | 23 | 112 | 26.2 | 0.245 | 30 | 0 |
| **766** | 1 | 126 | 60 | 0 | 0 | 30.1 | 0.349 | 47 | 1 |
| **767** | 1 | 93 | 70 | 31 | 0 | 30.4 | 0.315 | 23 | 0 |

In [5]:
```
#SHAPE OF OUR DATASET(TOTAL NO OF ROWS AND COLUMS)
data.shape
```

Out[5]:
```
(768, 9)
```

In [6]:
```
#GET THE INFORMATION OF OUR DATASET
data.info()
```

```
<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
```

In [7]:
```
#CHECKING FOR NULL VALUES IN THE DATASET
data.isnull()
```

Out[7]:

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 0 | False | False | False | False | False | False | False | False | False |
| 1 | False | False | False | False | False | False | False | False | False |
| 2 | False | False | False | False | False | False | False | False | False |
| 3 | False | False | False | False | False | False | False | False | False |
| 4 | False | False | False | False | False | False | False | False | False |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 763 | False | False | False | False | False | False | False | False | False |
| 764 | False | False | False | False | False | False | False | False | False |
| 765 | False | False | False | False | False | False | False | False | False |
| 766 | False | False | False | False | False | False | False | False | False |
| 767 | False | False | False | False | False | False | False | False | False |

768 rows × 9 columns

In [8]:
```
data.isnull().sum()
```

Out[8]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [9]:
```
#OVERALL STATS OF OUR DATASET
data.describe()
```

Out[9]:

| | 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 | 120.894531 | 69.105469 | 20.536458 | 79.799479 | 31.992578 | 0.471876 | 33.240885 | 0.348958 |
| std | 3.369578 | 31.972618 | 19.355807 | 15.952218 | 115.244002 | 7.884160 | 0.331329 | 11.760232 | 0.476951 |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.078000 | 21.000000 | 0.000000 |
| 25% | 1.000000 | 99.000000 | 62.000000 | 0.000000 | 0.000000 | 27.300000 | 0.243750 | 24.000000 | 0.000000 |
| 50% | 3.000000 | 117.000000 | 72.000000 | 23.000000 | 30.500000 | 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 |

In [10]:
```
import numpy as np
```

In [11]:
```
data_copy = data.copy(deep=True)
```

In [12]:
```
data.columns
```

Out[12]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [13]:
```
data_copy[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI']] = data_copy[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI']].replace(0,np.nan)
```

In [14]:
```
data_copy.isnull().sum()
```

Out[14]:
```
Pregnancies                   0
Glucose                       5
BloodPressure                35
SkinThickness               227
Insulin                     374
BMI                          11
DiabetesPedigreeFunction      0
Age                           0
Outcome                       0
dtype: int64
```

In [15]:
```
data['Glucose'] = data['Glucose'].replace(0,data['Glucose'].mean())
data['BloodPressure'] = data['BloodPressure'].replace(0,data['BloodPressure'].mean())
data['SkinThickness'] = data['SkinThickness'].replace(0,data['SkinThickness'].mean())
data['Insulin'] = data['Insulin'].replace(0,data['Insulin'].mean())
data['BMI'] = data['BMI'].replace(0,data['BMI'].mean())
```

In [16]:
```
#STORE THE FEATURE MATRIX(INDEPENDENT VARIABLE) IN X AND RESPONSE(TARGET) IN Y
x = data.drop('Outcome',axis=1)
y = data['Outcome']
```

In [17]:
```
y
```

Out[17]:
```
0      1
1      0
2      1
3      0
4      1
      ..
763    0
764    0
765    0
766    1
767    0
Name: Outcome, Length: 768, dtype: int64
```

In [18]:
```
#SPLITTING THE DATASET INTO THE TRAINING SET AND TEST SET
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.20,random_state=42)
```

In [19]:
```
#SCIKIT LEARN PIPELINE (REDUCES MULTIPLE STEPS OUTPUT OF EACH STEP IS THE INPUT TO THE NEXT)
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
```

In [20]:
```
#CREATING PIPELINE USING SKLEARN
pipeline_lr = Pipeline([('scalar1',StandardScaler()),('lr_classifier',LogisticRegression())])

pipeline_knn = Pipeline([('scalar2',StandardScaler()),('knn_classifier',KNeighborsClassifier())])

pipeline_svc = Pipeline([('scalarr3',StandardScaler()),('svc_classifier',SVC())])

pipeline_dt = Pipeline([('dt_classifier',DecisionTreeClassifier())])

pipeline_rf = Pipeline([('rf_classifier',RandomForestClassifier())])
```

In [21]:
```
pipelines =[pipeline_lr,
            pipeline_knn,
            pipeline_svc,
            pipeline_dt,
            pipeline_rf]
```

In [22]:
```
pipelines
```

Out[22]:
```
[Pipeline(steps=[('scalar1', StandardScaler()),
                 ('lr_classifier', LogisticRegression())]),
 Pipeline(steps=[('scalar2', StandardScaler()),
                 ('knn_classifier', KNeighborsClassifier())]),
 Pipeline(steps=[('scalarr3', StandardScaler()), ('svc_classifier', SVC())]),
 Pipeline(steps=[('dt_classifier', DecisionTreeClassifier())]),
 Pipeline(steps=[('rf_classifier', RandomForestClassifier())])]
```

In [23]:
```
for pipe in pipelines:
    pipe.fit(x_train,y_train)
```

In [24]:
```
pipe_dict = {0:'LR',
             1:'KNN',
             2:'SVC',
             3:'DT',
             4:'RF'}
```

In [25]:
```
pipe_dict
```

Out[25]:
```
{0: 'LR', 1: 'KNN', 2: 'SVC', 3: 'DT', 4: 'RF'}
```

In [26]:
```
for i,model in enumerate(pipelines):
    print("{} Test Accuracy:{}".format(pipe_dict[i],model.score(x_test,y_test)*100))
```

```
LR Test Accuracy:76.62337662337663
KNN Test Accuracy:76.62337662337663
SVC Test Accuracy:73.37662337662337
DT Test Accuracy:72.72727272727273
RF Test Accuracy:74.02597402597402
```

In [27]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [28]:
```
x = data.drop('Outcome',axis=1)
y = data.drop['Outcome']
```

In [29]:
```
rf = RandomForestClassifier(max_depth=3)
```

In [30]:
```
rf.fit(x,y)
```

Out[30]:
```
RandomForestClassifier(max_depth=3)
```

In [31]:
```
#PREDICTION ON NEW DATA

new_data = pd.DataFrame({
    'Pregnancies':6,
    'Glucose':148.0,
    'BloodPressure':72.0,
    'SkinThickness':35.0,
    'Insulin':79.799479,
    'BMI':33.6,
    'DiabetesPedigreeFunction':0.627,
    'Age':50,
},index=[0])
```

In [32]:
```
p=rf.predict(new_data)
```

In [33]:
```
if p[0] == 0:
    print('non-diabetic')
else:
    print('diabetic')
```

```
diabetic
```

In [31]:
```
#PREDICTION ON NEW DATA

new_data = pd.DataFrame({
    'Pregnancies':6,
    'Glucose':148.0,
    'BloodPressure':72.0,
    'SkinThickness':35.0,
    'Insulin':79.799479,
    'BMI':33.6,
    'DiabetesPedigreeFunction':0.627,
    'Age':50,
},index=[0])
```

In [32]:
```
p=rf.predict(new_data)
```

In [33]:
```
if p[0] == 0:
    print('non-diabetic')
else:
    print('diabetic')
```

```
diabetic
```

In [34]:
```
#SAVE MODEL USING JOBLIB
import joblib
```

In [35]:
```
joblib.dump(rf,'model_joblib_diabetes')
```

Out[35]:
```
['model_joblib_diabetes']
```

In [36]:
```
model = joblib.load('model_joblib_diabetes')
```

In [37]:
```
model.predict(new_data)
```

Out[37]:
```
array([1], dtype=int64)
```

# 7.CONCLUSION

In this project report, we proposed a comprehensive approach for diabetes prediction using classification and regression algorithms. The goal was to develop an accurate and interpretable model for assessing diabetes risk and providing valuable insights for healthcare professionals.

We began by collecting a diverse dataset containing clinical, demographic, and lifestyle features of individuals. Through rigorous data preprocessing, including cleaning, normalization, and handling missing values, we ensured the dataset's quality and integrity. Feature selection techniques were then applied to identify the most informative features, reducing dimensionality and improving prediction accuracy.

Through this project, we developed a comprehensive diabetes prediction model that integrated classification and regression techniques. The results demonstrated the model's ability to accurately predict diabetes and estimate diabetes risk using the selected features. The model can assist healthcare professionals in identifying individuals at risk and tailoring preventive measures and personalized treatment plans accordingly.

The project's findings contribute to the field of diabetes prediction, showcasing the effectiveness of classification and regression algorithms in accurately assessing diabetes risk. The proposed model's interpretability allows for better understanding and trust in the predictions, enabling healthcare professionals to make informed decisions.