

NAME OF THE PROJECT : AI BASED DIABETES PREDICTION SYSTEM

PHASE - 1

PROBLEM STATEMENT:

The problem is to predict diabetes using machine learning techniques. The objective is to develop a model that accurately predicts the risk of developing diabetes in a person based on a set of features such as age, gender, weight, height, blood pressure, blood sugar levels, and other relevant factors. This project involves data preprocessing, feature engineering, model selection, training, and evaluation.

WHAT I UNDERSTAND

1. Data collection: The first step is to collect data about people's health and medical history. This data can be collected from a variety of sources, such as electronic health records (EHRs), clinical trials, and public health databases. It is important to collect data from a diverse population to ensure that the model is generalizable to a wide range of people.

2. Data preprocessing: Once the data has been collected, it needs to be preprocessed before it can be used to train a machine learning model. This involves cleaning the data, removing outliers, and handling missing values.

3. Feature engineering: Once the data has been preprocessed, you need to select the features that will be used to train the machine learning model. Not all features will be equally informative for the model, so it is important to select the features that are most relevant to predicting diabetes.

4. Model selection: There are many different machine learning algorithms that can be used for diabetes prediction. Some common choices include logistic regression, support vector machines, and random forests.

5. Model training: Once you have selected a model, you need to train it on your data. This involves feeding the model the data and allowing it to learn the relationships between the features and the target variable (i.e., whether or not a person has diabetes).

6. Model evaluation: Once the model is trained, you need to evaluate its performance on a held-out test set. This will give you an idea of how well the model will generalize to new data.

7. Model deployment: Once you are satisfied with the performance of the model, you can deploy it to production. This could involve integrating the model into an electronic health record (EHR) system or developing a stand-alone diabetes prediction app.

8. Model monitoring: It is important to monitor the performance of the model over time to ensure that it remains accurate. This is because new data may become available, and the prevalence of diabetes may change over time.

9. Model retraining: It may be necessary to retrain the model periodically with new data to ensure that it remains accurate. This is especially important if the prevalence of diabetes changes significantly over time.

Design Thinking:

Data Collection:

Source a dataset with medical features such as glucose levels, blood pressure, BMI, etc. Ensure the dataset includes information on whether the individual has diabetes or not. The dataset should be representative and follow constraints (females, at least 21 years old, Pima Indian heritage).

Data Preprocessing:

Clean the medical data, addressing missing values and outliers.

Normalize the data to bring all features to a standard scale.

Prepare the data for training by encoding categorical variables if necessary.

Feature Selection:

Identify relevant features affecting diabetes risk prediction.

Consider feature correlation and importance.

Prioritize features like glucose levels, BMI, and age based on medical literature.

Model Selection:

Experiment with multiple machine learning algorithms (Logistic Regression, Random Forest, Gradient Boosting).

Choose algorithms suitable for binary classification tasks.

Split the dataset into training and testing sets.

Model Evaluation:

Evaluate the model's performance using various metrics:

Accuracy: Overall correctness of predictions.

Precision: Proportion of true positives among positive predictions.

Recall: Proportion of true positives among actual positives.

F1-score: Harmonic mean of precision and recall.

ROC-AUC: Receiver Operating Characteristic - Area Under the Curve.

Iterative Improvement:

Fine-tune model parameters based on evaluation results.

Explore techniques like feature engineering to enhance prediction accuracy.

Consider cross-validation to ensure robustness.

Dataset Information:

Pregnancies: Number of times pregnant

Glucose: Plasma glucose concentration 2 hours after an oral glucose tolerance test

BloodPressure: Diastolic blood pressure (mm Hg)

SkinThickness: Triceps skinfold thickness (mm)
Insulin: 2-Hour serum insulin (mu U/ml)
BMI: Body mass index (weight in kg/(height in m)²)
DiabetesPedigreeFunction: Diabetes pedigree function
Age: Age in years
Outcome: Class variable (0 or 1)

SOURCE CODE

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression # Example algorithm

# Step 1: Data Collection
# Load the dataset containing information about people's health and medical history
data = pd.read_csv('diabetes_data.csv')

# Step 2: Data Preprocessing
# Handle missing values, duplicates, and encode categorical variables
# Address outliers if necessary

# Step 3: Feature Engineering
# Create new features or transform existing ones
# For example, you could calculate the body mass index (BMI) or the number of years a person
# has been overweight or obese.

# Step 4: Data Splitting
# Split the data into training and testing sets
X = data[['Age', 'Gender', 'Weight', 'Height', 'BloodPressure', 'BloodSugarLevels',
'FamilyHistoryDiabetes']] # Features
y = data['HasDiabetes'] # Target variable

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 5: Model Selection
# Choose a machine learning algorithm that is well-suited for classification tasks
model = LogisticRegression()

# Step 6: Model Training
model.fit(X_train, y_train)

# Step 7: Model Evaluation
# Evaluate the model's performance on the testing data
```

```
y_pred = model.predict(X_test)
# Calculate metrics such as accuracy, precision, recall, and F1 score

# Step 8: Hyperparameter Tuning (if needed)
# Fine-tune the model's hyperparameters using techniques like cross-validation

# Step 9: Deployment (Optional)
# Deploy the model for making real predictions in a production environment

# Step 10: Monitoring and Maintenance (Continuous)
# Continuously monitor the model's performance and retrain it with new data

# Predicting diabetes risk for new input data
new_data = pd.DataFrame({'Age': [35], 'Gender': ['Male'], 'Weight': [75], 'Height': [5.10],
                          'BloodPressure': [120], 'BloodSugarLevels': [100], 'FamilyHistoryDiabetes': [Yes]})
predicted_risk = model.predict_proba(new_data)[0][1]

print("Predicted Diabetes Risk:", predicted_risk)
```

DIABETES PREDICTION


To import the important python libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn as sk

import warnings
warnings.filterwarnings('ignore')
```

To upload the diabetes prediction dataset from the naan mudhalvan resources

```
diabetes=pd.read_csv('/content/karthie.csv')
diabetes.fillna(0)
```



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
0	6	148	72	35	0	33.6
1	1	85	66	29	0	26.6
2	8	183	64	0	0	23.3
3	1	89	66	23	94	28.1
4	0	137	40	35	168	43.1
...
763	10	101	76	48	180	32.9
764	2	122	70	27	0	36.6
765	5	121	72	23	112	26.4
766	1	126	60	0	0	30.1
767	1	93	70	31	0	30.4

768 rows x 7 columns

to get the head and tail of the dataset

```
diabetes.head(4)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
0	6	148	72	35	0	33.6
1	1	85	66	29	0	26.6
2	8	183	64	0	0	23.3
3	1	89	66	23	94	28.1

```
diabetes.tail(4)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
764	2	122	70	27	0	36.8
765	5	121	72	23	112	26.5

To explore the dataset using python

```
diabetes.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
```

To describe the dataset using python

```
diabetes.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin
count	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.7994
std	3.369578	31.972618	19.355807	15.952218	115.2440
min	0.000000	0.000000	0.000000	0.000000	0.0000
25%	1.000000	99.000000	62.000000	0.000000	0.0000
50%	3.000000	117.000000	72.000000	23.000000	30.5000
75%	6.000000	140.250000	80.000000	32.000000	127.2500
max	17.000000	199.000000	122.000000	99.000000	846.0000

To find the null values in the dataset

```
diabetes.isna().sum()

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

There is no any null values present in the dataset

To find the duplicated values in the dataset

```
diabetes.duplicated()
```

```
0      False
1      False
2      False
3      False
4      False
...
763    False
764    False
765    False
766    False
767    False
Length: 768, dtype: bool
```

To remove the all duplicated values in the dataset

```
diabetes.drop_duplicates(keep='first',inplace=True)
diabetes[diabetes.duplicated()]
```

```
Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin  BMI  D:
◀────────────────────────────────────────────────────────────────────────────────▶
```

There is no any duplicated values in the dataset

To identify the shape of the dataset

```
diabetes.shape
```

```
(768, 9)
```

The above dataset contains the 768 rows and 9 columns present in the dataset

To find the datatypes in the dataset

```
diabetes.dtypes
```

```
Pregnancies      int64
Glucose           int64
BloodPressure     int64
SkinThickness     int64
Insulin           int64
BMI              float64
DiabetesPedigreeFunction float64
Age              int64
Outcome          int64
dtype: object
```

To display the all the columns present in the dataset

```
diabetes.columns
```

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

to find the separate columns values present in the dataset

1. Pregnancies

Diabetes can cause problems during pregnancy for women and their developing babies. Poor control of diabetes during pregnancy increases the chances for birth defects and other problems for the baby. It can also cause serious complications for the woman. Proper health care before and during pregnancy can help prevent birth defects and other health problems.

```
diabetes.Pregnancies
```

```
0      6
1      1
2      8
3      1
4      0
...
763    10
764     2
765     5
766     1
767     1
Name: Pregnancies, Length: 768, dtype: int64
```

2.gulcose

A fasting blood sugar level from 100 to 125 mg/dL (5.6 to 6.9 mmol/L) is considered prediabetes. If it's 126 mg/dL (7 mmol/L) or higher on two separate tests, you have diabetes. Glucose tolerance test. For this test, you fast overnight.

```
diabetes.Glucose
```

```
0      148
1       85
2     183
3       89
4     137
...
763    101
764    122
765    121
766    126
767     93
Name: Glucose, Length: 768, dtype: int64
```

3.Blood pressure

Blood pressure target is usually below 140/90mmHg for people with diabetes or below 150/90mmHg if you are aged 80 years or above. For some people with kidney disease the target may be below 130/80mmHg. But it is important to speak to your healthcare team about your individual target.

```
diabetes.BloodPressure
```

```
0      72
1      66
2      64
3      66
4      40
...
763     76
764     70
765     72
766     60
767     70
Name: BloodPressure, Length: 768, dtype: int64
```

4.skin thickness

Skin thickness (epidermal surface to dermal fat inter- face), which is primarily determined by collagen con- tent, is greater in insulin-dependent diabetes mellitus (IDDM) patients who have been diabetic for >10 yr (11,12). This possibly reflects increased collagen cross- linkage and reduced collagen turnover (2,3).

diabetes.SkinThickness

```
0      35
1      29
2       0
3      23
4      35
...
763    48
764    27
765    23
766     0
767    31
Name: SkinThickness, Length: 768, dtype: int64
```

5.Insulin

With type 1 diabetes, the body does not make any insulin and therefore insulin has to be injected regularly every day to stay alive. With type 2 diabetes, the body does not make enough insulin, or the insulin that is made does not work well. Insulin injections are sometimes needed to manage blood glucose levels.

diabetes.Insulin

```
0      0
1      0
2      0
3     94
4    168
...
763   180
764     0
765   112
766     0
767     0
Name: Insulin, Length: 768, dtype: int64
```

6.BMI

BMI-Body Mass Index

BMI is one of the most commonly used tools to assess whether you are overweight. So, if your BMI is at 30 or more, you are at risk of getting type 2 diabetes. The higher your BMI goes from this value, the greater the chances of getting type 2 diabetes.

diabetes.BMI

```
0      33.6
1      26.6
2      23.3
3      28.1
4      43.1
...
763    32.9
764    36.8
765    26.2
766    30.1
```

```
767    30.4
Name: BMI, Length: 768, dtype: float64
```

7.diabetes pedigree function

Diabetes pedigree function (DPF) calculates diabetes likelihood depending on the subject's age and his/her diabetic family history. Very little is known about the determinants of DPF for gestational diabetes mellitus (GDM) and normal women.

```
diabetes.DiabetesPedigreeFunction

0      0.627
1      0.351
2      0.672
3      0.167
4      2.288
...
763    0.171
764    0.340
765    0.245
766    0.349
767    0.315
Name: DiabetesPedigreeFunction, Length: 768, dtype: float64
```

8.Age

Type 2 diabetes most often develops in people over age 45, but more and more children, teens, and young adults are also developing it.

```
diabetes.Age

0      50
1      31
2      32
3      21
4      33
..
763    63
764    27
765    30
766    47
767    23
Name: Age, Length: 768, dtype: int64
```

9.Outcomes

Over time, diabetes can damage blood vessels in the heart, eyes, kidneys and nerves. People with diabetes have a higher risk of health problems including heart attack, stroke and kidney failure.

```
diabetes.Outcome

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
```

ACCURACY:

Accuracy of the diabetes prediciton model

```
tree_accuracy
```

```
67.70833333333334
```

For diabetes prediction model the decision tree algorithm has accuracy of 68% of successful efficiency

Diabetes prediction using python excuted successfully