DIABETES PREDICTION MACHINE LEARNING PROJECT

Harnessing Machine Learning to Diagnose Diabetes in Women

Shreyas Sonwane

7th June 2024

Project Overview

- **Objective:** Predict whether a person is diabetic based on health metrics.
- **Dataset:** Pima Indian Diabetes dataset.
- **Features:** Pregnancies, Glucose levels, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age, Outcome.

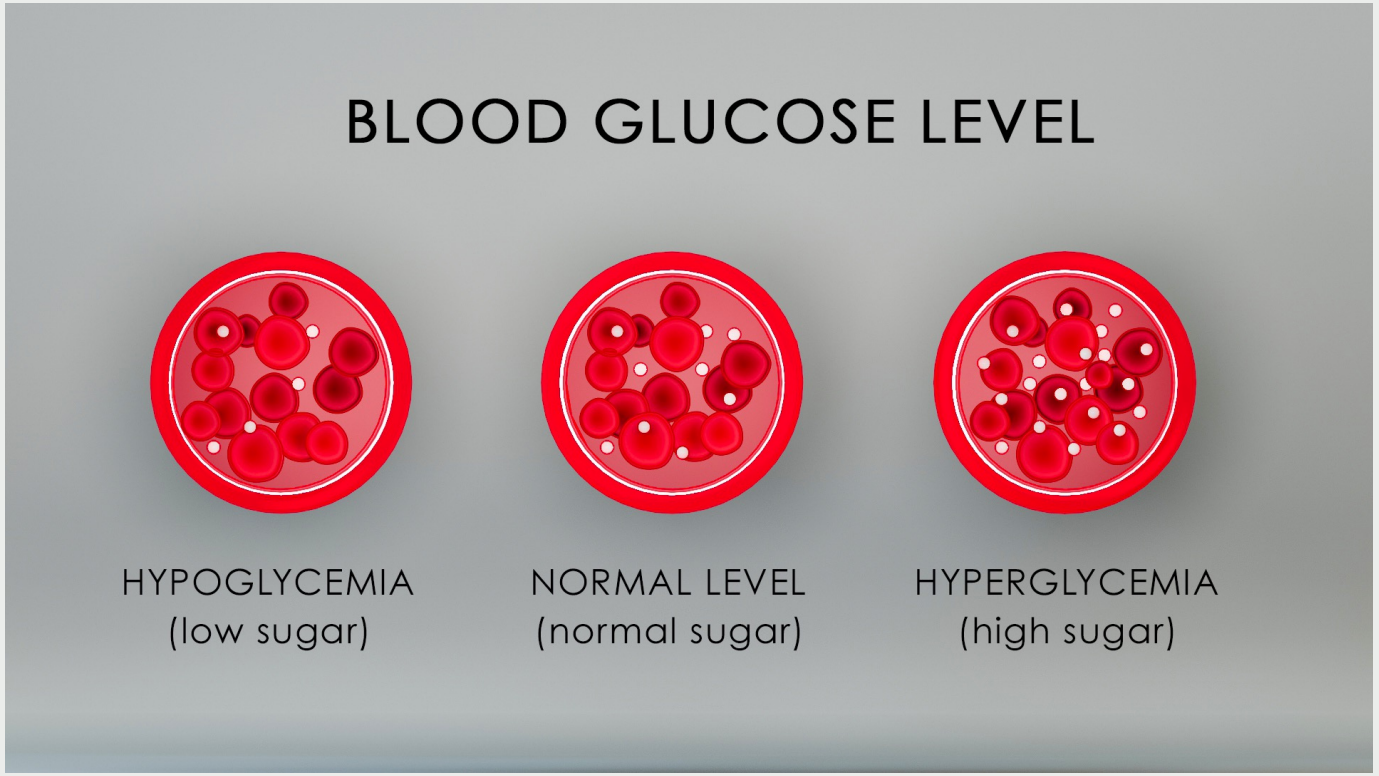


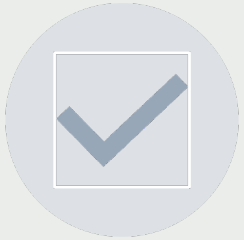
TABLE OF CONTENTS

INSTALLATION

DATASET

DATA PREPROCESSING

MODEL TRAINING



EVALUATION

PREDICTION

USAGE

INSTALLATION

- **Requirements:** Python, Numpy, Pandas, Scikit-Learn
- **Installation Command:** pip install numpy pandas scikit-learn
- **Alternative:** Use Google Colab to run the cells directly

Dataset

Name: Pima Indian Diabetes Dataset

Columns

- Pregnancies
- Glucose
- BloodPressure
- SkinThickness
- Insulin
- BMI
- Age
- Outcome
- DiabetesPedigreeFunction

Data Preprocessing

Steps:

1. Loading the dataset

```
diabetes_dataset = pd.read_csv("/content/diabetes.csv")
```

2. Inspecting the dataset

```
diabetes_dataset.head()
diabetes_dataset.shape
diabetes_dataset.describe()
diabetes_dataset["Outcome"].value_counts()
```

Data Preprocessing (Cont.)

3. Separating features and labels

```
X = diabetes_dataset.drop(columns="Outcome", axis=1)
Y = diabetes_dataset["Outcome"]
```

4. Standardizing the data

```
scaler = StandardScaler()
scaler.fit(X)
standardized_data = scaler.transform(X)
X = standardized_data
```

Model Training

Steps:

1. Splitting the data

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
```

2. Training the SVM(Support Vector Machine) Model

```
classifier = svm.SVC(kernel='linear’)
classifier.fit(X_train, Y_train)
```

Evaluation

Training Data Accuracy

```
X_train_prediction = classifier.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
print('Accuracy score of the training data:', training_data_accuracy)
```

Test Data Accuracy

```
X_test_prediction = classifier.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
print('Accuracy score of the test data:', test_data_accuracy)
```

Prediction

Steps:

1. Input Data

```
input_data = (1, 163, 72, 0, 0, 39, 1.222, 33)
input_data_nmp = np.asarray(input_data)
input_data_reshape = input_data_nmp.reshape(1, -1)
```

2. Standardize and Predict

```
std_data = scaler.transform(input_data_reshape)
prediction = classifier.predict(std_data)
```

3. Output

```
if prediction[0] == 0:
        print("The Person is not diabetic")
else:
        print("The Person is diabetic")
```

USAGE

Steps to run the project:

- **Clone the repository or download the script.**
- **Install the required libraries.**
- **Run the script in a Jupyter notebook or any Python environment.**

Link to Github

Click here