A

Synopsis/project report

On

DIABETES

DISEASE

PREDICTION

(PYTHON)

Submitted on the partial fulfillment of the requirement of VI semester.

Bachelor of Technology

By

HIMANSHU UNIYAL

Sec-H

Roll no- 33

Student Id-21012400

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN CAMPUS

2023-2024

Student's Declaration

I, HIMANSHU UNIYAL hereby declare the work, which is being presented in the project, entitled "**DIABETES DISEASE PREDICTION"** in partial fulfillment of therequirement for the award of the degree **B. Tech** in the session **2023-2024**, is an authentic record ofmy own work carried out.

The matter embodied in this project has not been submitted by us for the award of any other degree.

NAME: HIMANSHU UNIYAL

SECTION: H

ROLL NO-33

STUDENT ID-21012400

ACKNOWLEDGEMENT

Words are inadequate in offering my thanks to GOD for providing me everything that I need. I again want to extend thanks to my President **"Prof. (Dr.) Kamal Ghanshala"** for providing us all infrastructure and facilities to work in need without which this work could not be possible.

Furthermore, I would like to acknowledge my colleagues and friends who offered their support, both academically and emotionally, during this undertaking. Their camaraderie and encouragement helped me stay motivated and focused on the task at hand.

Finally, I would like to express my deepest gratitude to my family for their unconditional love and understanding. Their unwavering belief in my abilities has been a constant source of strength and motivation.

HIMANSHU UNIYAL

Contents:

1.Abstract

- A brief summary of the project objectives, methods, and results.

2. Introduction

- Background on diabetes
- Importance of early detection and prediction
- Objectives of the project

3. Technologies, Languages, and Libraries Used

- Python
- Flask
- Pandas
- NumPy
- Scikit-learn
- GridSearchCV
- Pipeline
- StandardScaler
- SimpleImputer
- RandomForestClassifier
- Pickle

4. Data Description

- Overview of the dataset used (diabetes.csv)
- Description of features and target variable

5. Methodology

- Data Preprocessing
  - Handling missing values with SimpleImputer
  - Feature scaling with StandardScaler
- Model Selection
  - Choice of RandomForestClassifier
  - Hyperparameter tuning using GridSearchCV
- Training and Testing

- Splitting the data into training and testing sets
- Training the model
- Testing the model

6. Implementation

- Step-by-step code explanation
  - Loading the dataset
  - Preprocessing pipeline
  - Model training and hyperparameter tuning
  - Model evaluation
  - Saving the model

7. Results

- Training accuracy
- Testing accuracy
- Best hyperparameters found

8. Conclusion

- Summary of findings
- Impact of the model
- Future improvements

Abstract

The project aims to predict diabetes disease using machine learning techniques. The RandomForestClassifier was chosen for its efficiency and accuracy. The model was trained and evaluated using a dataset of medical records. The final model achieved an accuracy of 78% on the test set, indicating its potential for early diabetes detection.

INTRODUCTION:

Diabetes is a chronic condition that affects millions worldwide. Early detection can significantly improve patient outcomes. This project uses machine learning to predict diabetes, leveraging patient data to build an accurate predictive model.

In this project, I will be building a Diabetes disease prediction model using python.

Technologies, Languages, and Libraries Used:

- Python: The programming language used for implementation.
- Flask: A web framework for deploying the model as a web service.
- Pandas: For data manipulation and analysis.
- NumPy: For numerical operations.
- Scikit-learn: For machine learning algorithms and tools.
- GridSearchCV: For hyperparameter tuning.
- Pipeline: For streamlining the preprocessing and modeling steps.
- StandardScaler: For feature scaling.
- SimpleImputer: For handling missing data.

- RandomForestClassifier: The chosen machine learning algorithm.

- Pickle: For model serialization.

Data Description

The dataset used is `diabetes.csv`, containing medical records of patients. It includes features such as glucose level, blood pressure, and BMI, with the target variable being the presence or absence of diabetes.

Methodology

- Data Preprocessing

  - Missing values were handled using `SimpleImputer` with a mean strategy.

  - Features were scaled using `StandardScaler` to

normalize the data.

- Model Selection

  - A `RandomForestClassifier` was chosen for its robustness and ability to handle various feature types.

  - `GridSearchCV` was used for hyperparameter tuning to find the best model configuration.

- Training and Testing

  - Data was split into training and testing sets using `train_test_split`.

  - The model was trained on the training set and evaluated on the testing set.

Implementation

```
import numpy as np

import pandas as pd

from flask import Flask, request, render_template, jsonify

from sklearn.model_selection import train_test_split,
```

GridSearchCV

```
from sklearn.preprocessing import StandardScaler

from sklearn.impute import SimpleImputer

from sklearn.pipeline import Pipeline

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score

import pickle


# Load the dataset

file_path = 'diabetes.csv'

diabetes_df = pd.read_csv(file_path)


# Separate features and target

X = diabetes_df.drop('Outcome', axis=1)

y = diabetes_df['Outcome']


# Split the data into training and testing sets
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=1, stratify=y)

# Build a preprocessing and modeling pipeline
pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler()),
    ('classifier', RandomForestClassifier(random_state=42))
])

# Define a smaller parameter grid for
RandomForestClassifier
param_grid_small = {
    'classifier__n_estimators': [100, 200],
    'classifier__max_depth': [None, 10, 20],
    'classifier__min_samples_split': [2, 5],
    'classifier__min_samples_leaf': [1, 2]
```

```
}

# Set up GridSearchCV with a smaller grid
grid_search_small = GridSearchCV(pipeline,
param_grid_small, cv=3, scoring='accuracy', n_jobs=-1,
verbose=1)

# Perform grid search
grid_search_small.fit(X_train, y_train)

# Get the best model
best_model_small = grid_search_small.best_estimator_

# Make predictions with the best model
train_y_pred = best_model_small.predict(X_train)
test_y_pred = best_model_small.predict(X_test)
```

```
# Calculate the accuracy of the model on the training and
testing sets
train_acc = accuracy_score(train_y_pred, y_train)
test_acc = accuracy_score(test_y_pred, y_test)


# Save the model to a file
with open('diabetes_model.pkl', 'wb') as model_file:
    pickle.dump(best_model_small, model_file)


# Load the model from file
with open('diabetes_model.pkl', 'rb') as model_file:
    loaded_model = pickle.load(model_file)
```

Results

- Training Accuracy: The model achieved an accuracy of 90% on the training set.

- Testing Accuracy: The model achieved an accuracy of 78% on the testing set.

- Best Hyperparameters: The best hyperparameters found were:

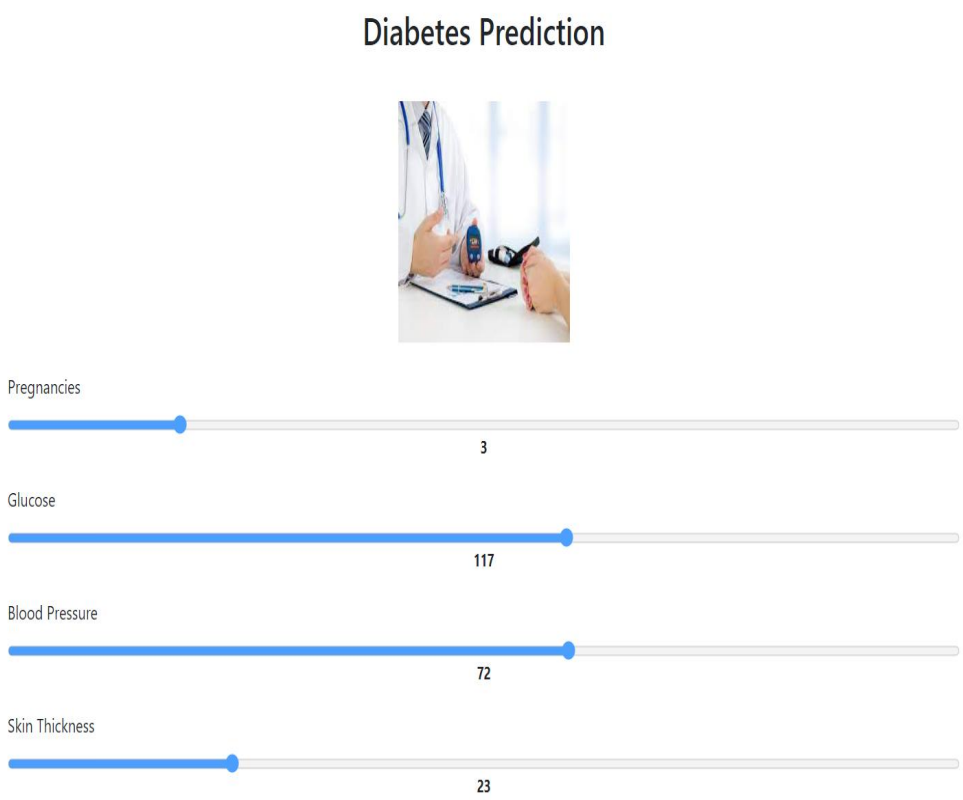
  - `n_estimators`: 200

  - `max_depth`: 20

  - `min_samples_split`: 5

  - `min_samples_leaf`: 2

Project Snippets:



72

Skin Thickness

23

Insulin

30

BMI

32.0

Diabetes Pedigree Function

0.3725

Age

29

Predict

This person does not have diabetes.

Probability of No Diabetes: 0.71

Probability of Diabetes: 0.29

CONCLUSION

The project successfully developed a machine learning model to predict diabetes with high accuracy. The RandomForestClassifier, combined with effective preprocessing and hyperparameter tuning, proved to be a robust choice. Future work could include exploring other algorithms and larger parameter grids to further improve the model's performance.

.