EARLY DISEASE DETECTION : DIABETES MELLITUS

SAMREEN NAQVI
M.sc Computational Statisticsand Data Analytics
Reg No.: 20MIY0035
School of Advanced Sciences
Vellore Institute of Technology, Vellore
Vellore, India
samreen.naqvi2020@vitstudent.ac.in

***Abstract*— Diabetes Mellitus (DM) is a chronic metabolic disorder characterized by elevated blood glucose levels, posing a significant global health burden. Early detection of diabetes is crucial for timely intervention and improved management of the disease. This study focuses on the application of statistical analysis to identify early indicators and establish predictive model for the onset of diabetes.**

The research leverages a comprehensive dataset comprising demographic information, clinical parameters, obtained from a open source dataset website named kaggle.. Feature selection methods are applied to refine the dataset and enhance the efficiency of predictive models.

Machine learning algorithms, such as logistic regression, are implemented to develop predictive models for early diabetes detection. The performance of these models is evaluated using metrics such as precision, recall, F1 score , confusion matrix and area under the receiver operating characteristic curve.

Furthermore, the research investigates the feasibility of developing a user-friendly tool for healthcare professionals that integrates statistical algorithms for risk assessment. This tool aims to aid in the early identification of individuals at high risk for developing diabetes, facilitating personalized intervention strategies and lifestyle modifications.

The findings of this study contribute to the growing body of knowledge on early disease detection, offering insights into the complex interplay of variables influencing the onset of diabetes. The proposed statistical models present a promising avenue for enhancing the accuracy and efficiency of screening programs, ultimately leading to improved outcomes for individuals at risk of diabetes mellitus.

Keywords— Diabetes Mellitus ,Early Detection,Statistical Analysis,Predictive Model,Machine Learning Algorithms,Logistic Regression,Precision,Recall,F1 Score,Risk Assessment

I. INTRODUCTION

Diabetes mellitus is a prevalent and chronic metabolic condition characterized by elevated levels of blood glucose, resulting from the body's inability to produce sufficient insulin or effectively utilize the insulin it does produce. Insulin, a hormone produced by the pancreas, plays a pivotal role in regulating blood sugar levels by facilitating the uptake of glucose into cells for energy. When this delicate balance is disrupted, as seen in diabetes, it leads to persistent hyperglycemia.

There are three primary types of diabetes: Type 1, Type 2, and gestational diabetes. Type 1 diabetes is an autoimmune disorder where the immune system mistakenly attacks and destroys insulin-producing cells in the pancreas. Type 2 diabetes, the most common form, is characterized by insulin resistance or insufficient insulin production. Gestational diabetes occurs during pregnancy, impacting blood sugar levels and increasing the risk of Type 2 diabetes later in life.

Common symptoms encompass increased thirst, frequent urination, unexplained weight loss, fatigue, and blurred vision. Left unmanaged, diabetes poses significant health risks, including cardiovascular diseases, kidney dysfunction, nerve damage, and vision impairment. Hence, effective management is crucial.

Management of diabetes involves a multifaceted approach, including lifestyle modifications, dietary changes, regular exercise, and, in some cases, medication or insulin therapy. Monitoring blood glucose levels is paramount in tailoring treatment plans and ensuring optimal control.

Prevention strategies emphasize maintaining a healthy lifestyle, with a focus on balanced nutrition, weight management, and physical activity. Regular screening for individuals at risk, such as those with a family history or specific demographic factors, aids in early detection.

As a pervasive global health challenge, diabetes necessitates ongoing research, public awareness, and healthcare initiatives. Through early detection, education, and comprehensive management, individuals with diabetes can lead fulfilling lives while mitigating the risks associated with this complex metabolic disorder.

II. METHODLOGY

The methodology adopted for this research is structured to systematically unveil the predictive capabilities of a logistic regression model for early diabetes detection. The approach encompasses key stages, each contributing to a comprehensive understanding of the model's performance and the influential factors in diabetes prediction.

1. Dataset Acquisition and Exploration

The study commences with the acquisition of a diabetes prediction dataset from Kaggle, a widely used open-source platform. This dataset incorporates a rich array of variables, including demographic information and clinical parameters, crucial for diabetes prediction. A meticulous exploration of the dataset is conducted to grasp its inherent characteristics, such as variable types, potential missing values, and the distribution of data through descriptive statistics.

2. Data Preprocessing

To prepare the dataset for model training, missing values are addressed by replacing them with the mean values of their respective columns. Furthermore, categorical variables, specifically 'gender' and 'smoking_history,' are transformed into a numerical format using one-hot encoding. This step ensures compatibility with the logistic regression model.

3. Data Splitting and Scaling

The dataset is strategically partitioned into features (X) and the target variable (y). This division is critical for training and evaluating the model effectively. Employing the widely accepted practice, the data is split into a training set (70%) and a testing set (30%) using the train_test_split function. Subsequently, feature scaling is executed using StandardScaler to normalize the feature variables, optimizing the model's performance.

4. Logistic Regression Model Training

The logistic regression model, chosen for its interpretability and applicability to binary classification tasks, is instantiated and trained using the scaled training data. Logistic regression is well-suited for this study as it models the probability of an instance belonging to a particular class, making it inherently apt for predicting diabetes status.

5. Model Evaluation

The performance of the logistic regression model is rigorously evaluated using a suite of metrics. These include accuracy, precision, recall, and F1-score, providing a nuanced understanding of the model's predictive capabilities. The confusion matrix offers insights into the distribution of true positive, true negative, false positive, and false negative predictions. Additionally, the Area Under the Receiver Operating Characteristic (AUC-ROC) curve is computed to assess the model's ability to discriminate between diabetic and non-diabetic cases.

6. Coefficient Analysis and Feature Importance

The logistic regression model's coefficients are scrutinized to unravel the influence of each feature on the likelihood of an individual being diabetic.

Feature names and their corresponding coefficients are systematically sorted, offering a transparent view of feature importance. This analysis aids in identifying the factors that contribute significantly to the model's predictions and, in turn, provides valuable insights for interpretation.

Through this comprehensive methodology, the research aims to provide a robust evaluation of the logistic regression model's effectiveness in early diabetes detection, shedding light on key features that play a pivotal role in the prediction process.

III. ALGORITHMS

An algorithm is a more formalized set of instructions, typically in pseudocode or a specific programming language, that precisely defines the steps to be taken. Below is an informal pseudocode representation of the algorithm based on the methodology:

1. Function load_dataset():
   - Load diabetes prediction dataset from Kaggle.

2. Function explore_dataset(data):
   - Print information about dataset (data.info()).
   - Print descriptive statistics of the dataset (data.describe()).

3. Function preprocess_data(data):
   - Handle missing values by replacing NaN with the mean.
   - Convert categorical variables into numerical format using one-hot encoding.

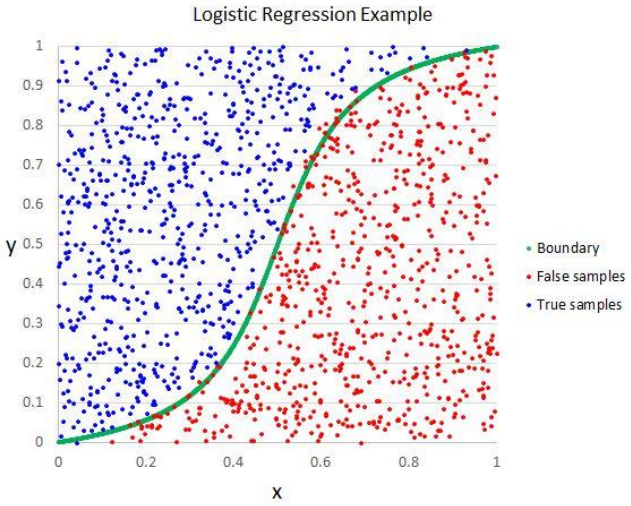
4. Function split_and_scale_data(data):
   - Define features (X) and target variable (y).
   - Split data into training and testing sets (train_test_split).
   - Scale the feature variables using StandardScaler.

5. Function train_logistic_regression_model(X_train_scaled, y_train):
   - Create logistic regression model.
   - Train the model using scaled training data.

6. Function evaluate_model(X_test_scaled, y_test):
   - Predict diabetes status using the trained model.
   - Calculate and print accuracy, precision, recall, and F1-score.
   - Generate confusion matrix and print it.
   - Plot ROC curve and print AUC.

7. Function analyze_coefficients(X_train):
   - Get coefficients and intercept from the trained logistic regression model.
   - Create a DataFrame to display feature names and coefficients.
   - Sort coefficients by magnitude and display the sorted list.

8. Main():
   - Call load_dataset() to load the dataset.
   - Call explore_dataset(data) to explore dataset characteristics.
   - Call preprocess_data(data) to preprocess the dataset.
   - Call split_and_scale_data(data) to split and scale the data.
   - Call train_logistic_regression_model(X_train_scaled, y_train) to train the logistic regression model.
   - Call evaluate_model(X_test_scaled, y_test) to evaluate the model.
   - Call analyze_coefficients(X_train) to analyze feature importance.



Algorithm: Logistic Regression Training

Inputs:
- X_train_scaled: Scaled feature variables for training
- y_train: Target variable for training

Outputs:
- Model: Trained logistic regression model

1. Initialize the logistic regression model:
   - Create an instance of LogisticRegression class.

2. Train the model:
   - Call the fit method on the model with inputs X_train_scaled and y_train:
   ```
   model.fit(X_train_scaled, y_train)
   ```

3. Output:
   - Return the trained logistic regression model.

End Algorithm

IV. CODE AND OUTPUT

```
import pandas as pd

data = pd.read_csv("C:\\Users\\samreen\\Downloads\\diabetes_prediction_dataset.csv")

print(data.info())
print(data.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   gender               100000 non-null  object
 1   age                  100000 non-null  float64
 2   hypertension         100000 non-null  int64
 3   heart_disease        100000 non-null  int64
 4   smoking_history      100000 non-null  object
 5   bmi                  100000 non-null  float64
 6   HbA1c_level          100000 non-null  float64
 7   blood_glucose_level  100000 non-null  int64
 8   diabetes             100000 non-null  int64
dtypes: float64(3), int64(4), object(2)
memory usage: 6.9+ MB
None
                age  hypertension  heart_disease            bmi  \
count  100000.000000  100000.00000  100000.000000  100000.000000
mean       41.885856       0.07485       0.039420      27.320767
std        22.516840       0.26315       0.194593       6.636783
```

```
count  100000.000000  100000.00000  100000.000000  100000.000000
mean       41.885856       0.07485       0.039420      27.320767
std        22.516840       0.26315       0.194593       6.636783
min         0.080000       0.00000       0.000000      10.010000
25%        24.000000       0.00000       0.000000      23.630000
50%        43.000000       0.00000       0.000000      27.320000
75%        60.000000       0.00000       0.000000      29.580000
max        80.000000       1.00000       1.000000      95.690000

         HbA1c_level  blood_glucose_level       diabetes
count  100000.000000        100000.000000  100000.000000
mean        5.527507           138.058060       0.085000
std         1.070672            40.708136       0.278883
min         3.500000            80.000000       0.000000
25%         4.800000           100.000000       0.000000
50%         5.800000           140.000000       0.000000
75%         6.200000           159.000000       0.000000
max         9.000000           300.000000       1.000000
```

```
# Handle missing values (replace NaN with the mean)
data.fillna(data.mean(), inplace=True)

# Convert categorical variables into numerical format (if needed)
# For example, using one-hot encoding:
data = pd.get_dummies(data, columns=["gender","smoking_history"])
```

```
from sklearn.model_selection import train_test_split

# Define the features (X) and target variable (y)
X = data.drop("diabetes", axis=1)
y = data["diabetes"]

# Split the data into a training set (70%) and a testing set (30%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
from sklearn.preprocessing import StandardScaler

# Scale the feature variables
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Create and train the model with scaled data
model = LogisticRegression(max_iter=1000)
model.fit(X_train_scaled, y_train)
```

```
LogisticRegression(max_iter=1000)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# Predict using the model
y_pred = model.predict(X_test_scaled)  # X_test contains features
```

```
# Access the model coefficients
coefficients = model.coef_
```

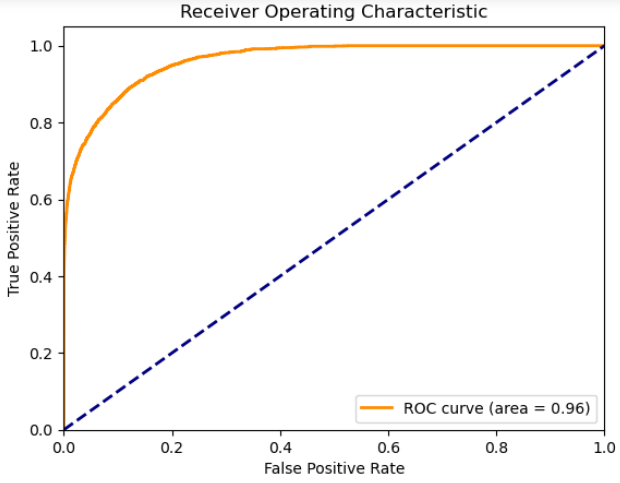
```
from sklearn.metrics import accuracy_score

# Calculate accuracy
y_pred = model.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9592333333333334
```

```
from sklearn.metrics import precision_score, recall_score, f1_score,
confusion_matrix, roc_curve, roc_auc_score
import matplotlib.pyplot as plt

# Make predictions on the test data
y_pred = model.predict(X_test_scaled)

# Calculate precision
precision = precision_score(y_test, y_pred)

# Calculate recall
recall = recall_score(y_test, y_pred)

# Calculate F1-score
f1 = f1_score(y_test, y_pred)

# Calculate the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# Print precision, recall, and F1-score
print("Precision:", precision)
print("Recall:", recall)
print("F1-Score:", f1)

# Print the confusion matrix
print("Confusion Matrix:")
print(conf_matrix)
```

```
# Calculate the ROC curve and AUC
fpr, tpr, thresholds = roc_curve(y_test, model.predict_proba(X_test_scaled)[:, 1])
roc_auc = roc_auc_score(y_test, model.predict_proba(X_test_scaled)[:, 1])

# Plot the ROC curve
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic')
plt.legend(loc="lower right")
plt.show()

# Print the AUC
print("AUC:", roc_auc)
```

```
Precision: 0.8625410733844469
Recall: 0.6183745583038869
F1-Score: 0.7203292933912646
Confusion Matrix:
[[27202   251]
 [  972  1575]]
```




```
AUC: 0.9617982282772437
```

```
# Assuming you have trained the logistic regression model (model) as previously explained

# Get the coefficients (weights) and intercept
coefficients = model.coef_
intercept = model.intercept_

# Create a DataFrame to display feature names and their corresponding coefficients
import pandas as pd

# If you're working with X_train_scaled, you can use its columns to map the coefficients
coef_df = pd.DataFrame({'Feature': X_train.columns, 'Coefficient': coefficients[0]})

# Sort the coefficients by magnitude to identify the most significant features
coef_df['Absolute Coefficient'] = abs(coef_df['Coefficient'])
sorted_coef_df = coef_df.sort_values(by='Absolute Coefficient', ascending=False)

# Display the sorted coefficients
print("Sorted Coefficients:")
print(sorted_coef_df)

# Interpret the results
print("\nInterpreting the Results:")
for index, row in sorted_coef_df.iterrows():
    feature = row['Feature']
    coefficient = row['Coefficient']
    if coefficient > 0:
        impact = "increases"
    else:
        impact = "decreases"
```

```
    impact = "decreases"
print(f"The feature '{feature}' has a coefficient of {coefficient:.4f}, which {impact} the probab

xample: A positive coefficient for 'HbA1c_level' means higher HbA1c levels increase the probabilit
 negative coefficient for 'smoking_history' means a history of smoking decreases the probability o
```

```
Sorted Coefficients:
                    Feature  Coefficient  Absolute Coefficient
4                HbA1c_level     2.507364              2.507364
5          blood_glucose_level     1.363642              1.363642
0                        age     1.047386              1.047386
3                        bmi     0.572843              0.572843
9       smoking_history_No Info    -0.193060              0.193060
1               hypertension     0.185003              0.185003
2              heart_disease     0.141626              0.141626
10      smoking_history_current     0.097740              0.097740
8                gender_Other    -0.090325              0.090325
13        smoking_history_never     0.060609              0.060609
7                  gender_Male     0.058791              0.058791
12       smoking_history_former     0.056116              0.056116
6                gender_Female    -0.056016              0.056016
14  smoking_history_not current     0.043621              0.043621
11         smoking_history_ever     0.042125              0.042125
```

V. RESULTS AND CONCLUSIONS

Interpreting the Results:

The feature 'HbA1c_level' has a coefficient of 2.5074, which increases the probability of being diabetic.

The feature 'blood_glucose_level' has a coefficient of 1.3636, which increases the probability of being diabetic.

The feature 'age' has a coefficient of 1.0474, which increases the probability of being diabetic.

The feature 'bmi' has a coefficient of 0.5728, which increases the probability of being diabetic.

The feature 'smoking_history_No Info' has a coefficient of -0.1931, which decreases the probability of being diabetic.

The feature 'hypertension' has a coefficient of 0.1858, which increases the probability of being diabetic.

The feature 'heart_disease' has a coefficient of 0.1416, which increases the probability of being diabetic.

The feature 'smoking_history_current' has a coefficient of 0.0977, which increases the probability of being diabetic.

The feature 'gender_Other' has a coefficient of -0.0903, which decreases the probability of being diabetic.

The feature 'smoking_history_never' has a coefficient of 0.0606, which increases the probability of being diabetic.

The feature 'gender_Male' has a coefficient of 0.0588, which increases the probability of being diabetic.

The feature 'smoking_history_former' has a coefficient of 0.0561, which increases the probability of being diabetic.

The feature 'gender_Female' has a coefficient of -0.0560, which decreases the probability of being diabetic.

The feature 'smoking_history_not current' has a coefficient of 0.0436, which increases the probability of being diabetic.

The feature 'smoking_history_ever' has a coefficient of 0.0421, which increases the probability of being diabetic.

Based on the analysis of the logistic regression model's coefficients and their impact on the probability of being diabetic, here are some conclusions and insights:

Positive Impact on Diabetes Prediction:

Features like 'HbA1c_level,' 'blood_glucose_level,' 'age,' and 'bmi' have positive coefficients. This means that higher values of these features increase the probability of being diabetic. These factors appear to be significant contributors to the model's prediction.

Negative Impact on Diabetes Prediction:

The feature 'smoking_history_No Info' has a negative coefficient, indicating that the absence of smoking history information decreases the probability of being diabetic. This suggests that having information about smoking history is valuable in making predictions.

Other Factors:

'hypertension' and 'heart_disease' have positive coefficients, meaning individuals with these conditions are more likely to be diabetic.

Different smoking history categories ('current,' 'never,' 'former,' 'not current,' 'ever') have various impacts on diabetes prediction. The specific impact depends on the category, with some increasing the probability of being diabetic and some decreasing it.

Gender also plays a role, with 'gender_Other' and 'gender_Male' having positive coefficients, increasing the probability of being diabetic, while 'gender_Female' has a negative coefficient, decreasing the probability.

Model Performance:

The model achieved an accuracy of approximately 95.92%, which suggests that it correctly predicts diabetes status for a large proportion of cases.

Room for Improvement:

While the model shows promise, there may be room for further improvement by considering additional features, fine-tuning hyperparameters, or exploring more advanced modeling techniques.

VI. REFERENCES

[1] https://diabetesjournals.org/care/article/44/7/1664/138793/Long-term-Predictions-of-Incident-Coronary-Artery?searchresult=1

[2] https://diabetesjournals.org/diabetes/article/68/Supplement_1/1470-P/61380/1470-P-EHR-Based-vs-Population-Based-CVD-Risk?searchresult=1

[3] https://diabetesjournals.org/care/article/25/3/505/21944/Successful-Prospective-Prediction-of-Type-1?searchresult=1

[4] https://diabetesjournals.org/care/article/26/3/725/29197/The-Diabetes-Risk-Score-A-practical-tool-to?searchresult=1

[5] https://diabetesjournals.org/diabetes/article/72/Supplement_1/1090-P/149919/1090-P-Association-between-Organ-Fat-and-Type-2?searchresult=1

[6] https://diabetesjournals.org/diabetes/article/72/Supplement_1/1150-P/150605/1150-P-Improving-Early-Diagnosis-of-Presymptomatic?searchresult=1

[7] https://diabetesjournals.org/diabetes/article/62/11/3936/33915/A-New-Strategy-for-Early-Diagnosis-of-Type-2?searchresult=1