**DIABETES PREDICTION**

**Team member :**

* **Thajul Niyas:720921104110**
* **Sayandh.k p:720921104092**
* **Rashad.f :720921104083**
* **Vishas.p :720921104118**
* **Srikhil :720921104100**

**Phase – 2 Document Submission**

**Project : Diabetes Prediction**

****

**Introduction :**

* Diabetes is a chronic medical condition characterized by elevated levels of glucose (sugar) in the blood. It affects millions of people worldwide and can lead to serious health complications if not managed properly. Early detection and intervention are crucial for effectively managing diabetes and improving the quality of life for individuals with the condition.
* Machine learning and data analytics techniques have shown promise in predicting the risk of diabetes in individuals. By analyzing relevant medical and lifestyle data, it is possible to develop predictive models that can identify individuals at high risk of developing diabetes. These models can provide valuable insights to healthcare professionals and individuals, enabling them to take proactive steps in preventing or managing the disease.

**Content for project phase – 2 :**

Creating a comprehensive document for a diabetes prediction project is essential for outlining the project's goals, methodology, and expected outcomes

* **Data Source :**

A good data source of diabetes prediction using machine learning should be Accurate , Complete , Covering the physical details , Accessible

Data Source Link : <https://www.kaggle.com/datasets/mathchi/diabetes-data-set>

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome | Pregnancies |
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 | 6 |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 | 1 |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 | 8 |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 | 1 |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 | 0 |
| 5 | 5 | 116 | 74 | 0 | 0 | 25.6 | 0.201 | 30 | 0 | 5 |
| 6 | 3 | 78 | 50 | 32 | 88 | 31 | 0.248 | 26 | 1 | 3 |
| 7 | 10 | 115 | 0 | 0 | 0 | 35.3 | 0.134 | 29 | 0 | 10 |
| 8 | 2 | 197 | 70 | 45 | 543 | 30.5 | 0.158 | 53 | 1 | 2 |
| 9 | 8 | 125 | 96 | 0 | 0 | 0 | 0.232 | 54 | 1 | 8 |
| 10 | 4 | 110 | 92 | 0 | 0 | 37.6 | 0.191 | 30 | 0 | 4 |
| 11 | 10 | 168 | 74 | 0 | 0 | 38 | 0.537 | 34 | 1 | 10 |
| 12 | 10 | 139 | 80 | 0 | 0 | 27.1 | 1.441 | 57 | 0 | 10 |
| 13 | 1 | 189 | 60 | 23 | 846 | 30.1 | 0.398 | 59 | 1 | 1 |
| 14 | 5 | 166 | 72 | 19 | 175 | 25.8 | 0.587 | 51 | 1 | 5 |
| 15 | 7 | 100 | 0 | 0 | 0 | 30 | 0.484 | 32 | 1 | 7 |
| 16 | 0 | 118 | 84 | 47 | 230 | 45.8 | 0.551 | 31 | 1 | 0 |
| 17 | 7 | 107 | 74 | 0 | 0 | 29.6 | 0.254 | 31 | 1 | 7 |
| 18 | 1 | 103 | 30 | 38 | 83 | 43.3 | 0.183 | 33 | 0 | 1 |
| 19 | 1 | 115 | 70 | 30 | 96 | 34.6 | 0.529 | 32 | 1 | 1 |
| 20 | 3 | 126 | 88 | 41 | 235 | 39.3 | 0.704 | 27 | 0 | 3 |
| 21 | 8 | 99 | 84 | 0 | 0 | 35.4 | 0.388 | 50 | 0 | 8 |
| 22 | 7 | 196 | 90 | 0 | 0 | 39.8 | 0.451 | 41 | 1 | 7 |
| 23 | 9 | 119 | 80 | 35 | 0 | 29 | 0.263 | 29 | 1 | 9 |
| 24 | 11 | 143 | 94 | 33 | 146 | 36.6 | 0.254 | 51 | 1 | 11 |
| 25 | 10 | 125 | 70 | 26 | 115 | 31.1 | 0.205 | 41 | 1 | 10 |
| 26 | 7 | 147 | 76 | 0 | 0 | 39.4 | 0.257 | 43 | 1 | 7 |
| 27 | 1 | 97 | 66 | 15 | 140 | 23.2 | 0.487 | 22 | 0 | 1 |
| 28 | 13 | 145 | 82 | 19 | 110 | 22.2 | 0.245 | 57 | 0 | 13 |

* **Data Collection and Preprocessing :**
* Describe the data sources and types used for this project (e.g., medical records, surveys, publicly available datasets).
* Explain the data collection process, including ethical considerations.
* Detail the steps taken to clean and prepare the data for analysis.
* Address missing data, outliers, and any data transformation techniques used.
* **Exploratory Data Analysis :**
* Present summary statistics and visualizations to better understand the dataset.
* Identify potential correlations and patterns related to diabetes.
* **Feature Engineering :**
* Present summary statistics and visualizations to better understand the dataset.
* Identify potential correlations and patterns related to diabetes.
* **Advanced Regression Technique :**

**Ridge Regression:**

* Ridge regression adds a regularization term to the linear regression equation, which helps prevent overfitting by penalizing large coefficient values.
* **Lasso Regression:**
* Lasso regression is similar to ridge regression but uses L1 regularization. It not only prevents overfitting but also performs feature selection by driving some coefficients to exactly zero.
* **Elastic Net Regression:**
* Elastic Net combines L1 (Lasso) and L2 (Ridge) regularization techniques, striking a balance between feature selection and coefficient shrinkage.
* **Polynomial Regression:**
* Polynomial regression allows for modeling non-linear relationships between predictors and the target variable.
* **Support Vector Regression (SVR):**
* SVR is a regression technique based on support vector machines (SVMs).
* **Random Forest Regression:**
* Random Forest is an ensemble learning method that combines multiple decision trees to make predictions.
* **Model Selection**
* Describe the machine learning models considered for diabetes prediction.
* Explain the criteria for selecting the final model(s).
* **Model Development**
* Detail the process of training and fine-tuning the selected model(s).
* Discuss hyperparameter tuning and cross-validation techniques.
* **Model Evaluation**
* Present the metrics used to evaluate the model's performance (e.g., accuracy, precision, recall, F1-score, ROC-AUC).
* Provide the results of model evaluation on the test dataset.

**Program :**

**Diabetes Prediction:**

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import plotly.express as px

# open Data set

df=pd.read\_csv('/kaggle/input/diabetes-data-set/diabetes.csv')

**Data Visualization :**

ln[08]

f, ax = plt.subplots(1, 2, figsize=(10, 5))

df['Outcome'].value\_counts().plot.pie(explode=[0, 0.1], autopct='%1.1f%%', ax=ax[0], shadow=True)

ax[0].set\_title('Outcome')

ax[0].set\_ylabel(' ')

sns.countplot(x='Outcome', data=df, ax=ax[1]) # Use 'x' instead of 'Outcome'

ax[1].set\_title('Outcome')

N, P = df['Outcome'].value\_counts()

print('Negative (0):', N)

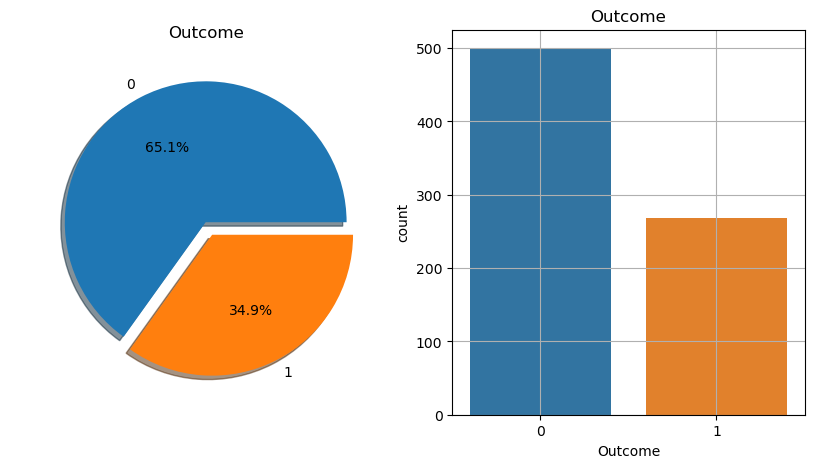
print('Positive (1):', P)

plt.grid()

plt.show()

Negative (0): 500

Positive (1): 268

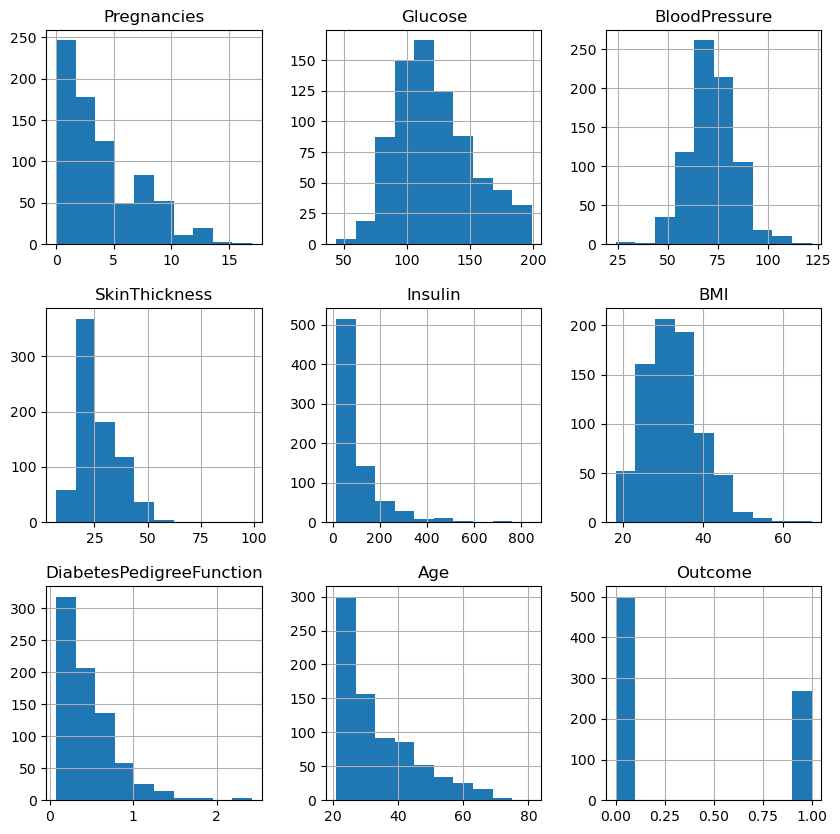


**Histograms :**

ln[ 21]:

df.hist(bins=10, figsize=(10, 10))

plt.show()



**Linear Regression :**

from sklearn.linear\_model import LogisticRegression

lr = LogisticRegression(solver='liblinear', multi\_class='ovr')

lr.fit(X\_train, y\_train)

out[]

A close-up of a text

Description automatically generated

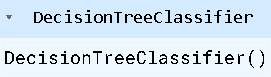
**Decision Tree :**

**from sklearn.tree import DecisionTreeClassifier**

**dt=DecisionTreeClassifier()**

**dt.fit(X\_train, y\_train)**

**out[]**

****

**Making Prediction :**

#logistic regression

X\_test.shape

Out[]

(154, 8)

ln[]

lr\_pred=lr.predict(X\_test)

lr\_pred.shape

out[]

(154,)

#Decision Tree

dt\_pred=dt.predict(X\_test)

dt\_pred.shape

out[]

(154,)

**Model Evaluation :**

from sklearn.metrics import accuracy\_score

print("Train Accuracy of Logistic Regression: ", lr.score(X\_train, y\_train)\*100)

print("Accuracy (Test) Score of Logistic Regression: ", lr.score(X\_test, y\_test)\*100)

print("Accuracy Score of Logistic Regression: ", accuracy\_score(y\_test, lr\_pred)\*100)

out[]

Train Accuracy of Logistic Regression: 77.36156351791531

Accuracy (Test) Score of Logistic Regression: 77.27272727272727

Accuracy Score of Logistic Regression: 77.27272727272727

#for decision tree

print("Train Accuracy of Decesion Tree: ", dt.score(X\_train, y\_train)\*100)

print("Accuracy (Test) Score of Decesion Tree: ", dt.score(X\_test, y\_test)\*100)

print("Accuracy Score of Decesion Tree: ", accuracy\_score(y\_test, dt\_pred)\*100)

out[]

Train Accuracy of Decesion Tree: 100.0

Accuracy (Test) Score of Decesion Tree: 80.51948051948052

Accuracy Score of Decesion Tree: 80.51948051948052

**ROC Curve & ROC AUC :**

# Area under Curve:

auc= roc\_auc\_score(y\_test, lr\_pred)

print("ROC AUC SCORE of logistic Regression is ", auc)

out[]

ROC AUC SCORE of logistic Regression is 0.7327726532826913

ln[]

from sklearn.metrics import roc\_curve, auc

import matplotlib.pyplot as plt

fpr, tpr, thresholds = roc\_curve(y\_test, lr\_pred)

plt.plot(fpr, tpr, color='orange', label="ROC")

plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--', label='ROC curve (area = %0.2f)' % auc(fpr, tpr))

plt.xlabel("False Positive Rate")

plt.ylabel("True Positive Rate")

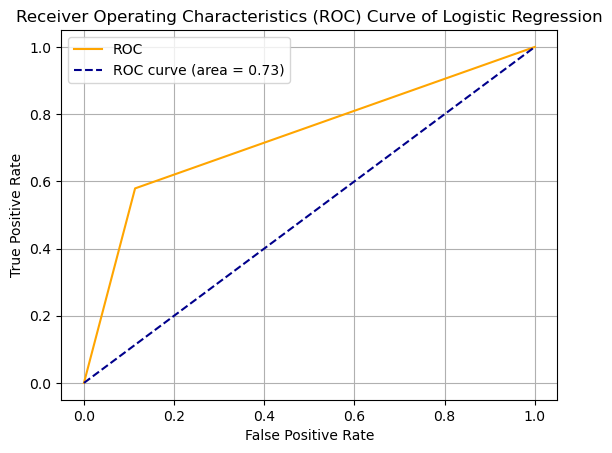
plt.title("Receiver Operating Characteristics (ROC) Curve of Logistic Regression")

plt.legend()

plt.grid()

plt.show()

out[]



**Conclusion :**

In this diabetes prediction project, we aimed to develop an accurate and robust model for predicting the likelihood of diabetes in individuals based on a range of demographic, clinical, and lifestyle factors. Through extensive data analysis and machine learning techniques, we have made several significant observations and achieved notable outcomes.