

Marathwada Shikshan Prasarak Mandal's
**Deogiri Institute of Engineering and Management Studies,
Aurangabad**

Project Report

on

Diabetes Prediction of Female Patients using Logistic Regression

Submitted By

**Vaishnavi Dattatraya Muley (36016)
Swapna Subhash Salunke (36017)**

for

**Continuous Assessment of
Machine Learning (TY CSE)**

**Dr. Babasaheb Ambedkar Technological University
Lonere (M.S.)**



Department of Computer Science and Engineering
**Deogiri Institute of Engineering and Management Studies,
Aurangabad**
(2020- 2021)

Project Report
on
Diabetes Prediction of Female Patients
using Logistic Regression

Submitted By

Vaishnavi Dattatraya Muley (36016)
Swapna Subhash Salunke (36017)

In partial fulfillment of
Bachelor of Technology
(Computer Science & Engineering)

Guided By

Dr. Padmapani P. Tribhuvan

Department of Computer Science & Engineering
Deogiri Institute of Engineering and Management Studies,
Aurangabad
(2020- 2021)

CERTIFICATE

This is to certify that, the Survey entitled “**Diabetes Prediction of Female Patients using Logistic Regression**” submitted by **Vaishnavi Dattatraya Muley, Swapna Subhash Salunke**, is a bonafide work completed under my supervision and guidance in partial fulfillment for award of Bachelor of Technology (Computer Science and Engineering) Degree of Dr. Babasaheb Ambedkar Technological University, Lonere.

Place: Aurangabad

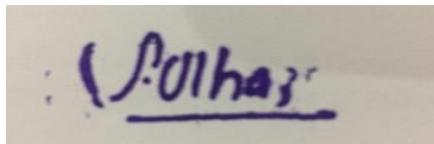
Date: 21/12/2020



Dr. Padmapani P. Tribhuvan
Guide



Mr. Sanjay B. Kalyankar
Head



Dr. Ulhas D. Shiurkar
Director,
Deogiri Institute of Engineering and Management Studies,
Aurangabad

Contents

List of Figures	i
List of Screen Shots	ii
List of Table	iii
1. INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Applications	4
2. DATA COLLECTION AND ANALYSIS	3
2.1 Dataset	5
2.2 Graph Represent Data	6
3. FINAL DESIGN AND IMPLEMENTATION	9
3.1 Module Implementation	9
3.2 Final Design	11
4. PERFORMANCE ANALYSIS (TNR 14)	13
5. CONCLUSION AND FUTURE SCOPE	15
Annexure	
Project Code	

List of Figures

Figure	Illustration	Page
1.2.1	System Flow	3
3.1.1	Logistic Regression sigmoid function	7
4.1	Confusion Matrix	13
4.2	Classification Report	14
4.3	ROC Curve	14

List of Screenshots

Screenshot	Illustration	Page
2.2.1	Diabetes and no Diabetes Patient	6
2.2.2	Compare Glucose with the outcome	7
2.2.3	Blood Pressure and age of entries who have diabetes	7
2.2.4	Pair plotting of Data frame	8
2.2.5	Histogram of all columns when the outcome is 1	8
3.2.1	System Flow	11
3.2.2	Logistic Regression sigmoid function	12
3.2.3	Confusion Matrix	12

List of Table

Table No	Illustration	Page
3.2.1	Dataset summary.	5
3.2.2	Pima dataset features.	6

1. INTRODUCTION

1.1 Introduction

A major public health problem, diabetes, is the surfeit rise of sugar level in blood and occurs when pancreas does not produce insulin, or in spite of the pancreas producing insulin, the body cannot use it effectively. It is the root cause of many associated health diseases. Diabetic peripheral neuropathy, for example, is a form of nerve pain caused by diabetes. Risk of eye problems increases due to the presence of diabetes and can lead to diabetic retinopathy and diabetic macular edema. Diabetic nephropathy which is the primary cause behind kidney failure is a consequence of diabetes. People with diabetes are more prone to coronary heart diseases and have heart attacks. Polycystic Ovary Syndrome (PCOS), a physical condition hampering the overall ovulation process, also increases the chances of having diabetes.

There are three main categories of diabetes:

- Type 1 diabetes occurs mostly to children and adolescents. In this case, the body produces very little or no insulin at all. As a result, daily insulin injections are needed to keep glucose levels under control. Frequent urination, sudden weight loss, abnormal thirst, constant hunger, blurred vision, and tiredness are common symptoms of this kind of diabetes. This can be treated with the help of insulin therapy.
- Type 2 diabetes is more prevalent in adults (90% cases). The body does not fully respond to insulin resulting in higher glucose levels. Obesity, unhealthy diet, high blood pressure, and physical inactivity are considered to be major risk factors that lead to type 2 diabetes. Insulin injections are required when oral medication is not sufficient enough to control blood sugar levels.
- Gestational Diabetes Mellitus (GDM), or simply gestational diabetes consists of high blood pressure during pregnancy and can cause health complications to both mother and children. It usually disappears during the pregnancy stage but the affected ones along with their children

have a risk of developing Type 2 diabetes in their later life. According to a survey in 2017, approximately 204 million women suffers from GDM. About 21.3 million live births had some form of hyperglycemia in pregnancy, among which about 85.1% occurred due to gestational diabetes. GDM typically affects around one out of seven births. Diabetes, of all types, can result in different complications in the body and also increase the overall risk of premature death. A recent research of 2017 shows that people with PCOS have an increased chance of having Type 2 Diabetes at a later age. According to the International Diabetes Federation (IDF) atlas 2019, one out of 11 adults (20–79 years) have diabetes which can be approximated as 463 million people. The global health report on diabetes from WHO shows that in addition to the 1.5 million deaths from diabetes in 2012, another 2.2 million deaths are resulted from cardiovascular and other diseases due to increased blood sugar levels. For the past three decades, diabetes has been increasing steadily and is growing more rapidly in low and middle-income countries. The WHO report, in 2016, shows that 1.6 million deaths occurred directly due to diabetes and in 2012, high blood glucose resulted in 2.2 million deaths.

1.2 Problem statement

Diabetes is a chronic disease with the potential to cause a worldwide health care crisis. According to International Diabetes Federation 382 million people are living with diabetes across the whole world. By 2035, this will be doubled as 592 million. Diabetes mellitus or simply diabetes is a disease caused due to the increase level of blood glucose. Various traditional methods, based on physical and chemical tests, are available for diagnosing diabetes. However, early prediction of diabetes is quite challenging task for medical practitioners due to complex interdependence on various factors as diabetes affects human organs such as kidney, eye, heart, nerves, foot etc. Data science methods have the potential to benefit other scientific fields by shedding new light on common questions. One such task is help to make predictions on medical data. Machine learning is an emerging scientific field in data science dealing with the ways in which machines learn from experience. The aim of this project is to develop a system which can perform early prediction of diabetes for a patient with a higher accuracy using different machine learning techniques. This project aims to predict diabetes via two different supervised machine

learning methods including: Naïve Bayes, Logistic regression. This project also aims to propose an effective technique for earlier detection of the diabetes disease.

System Flow

Working

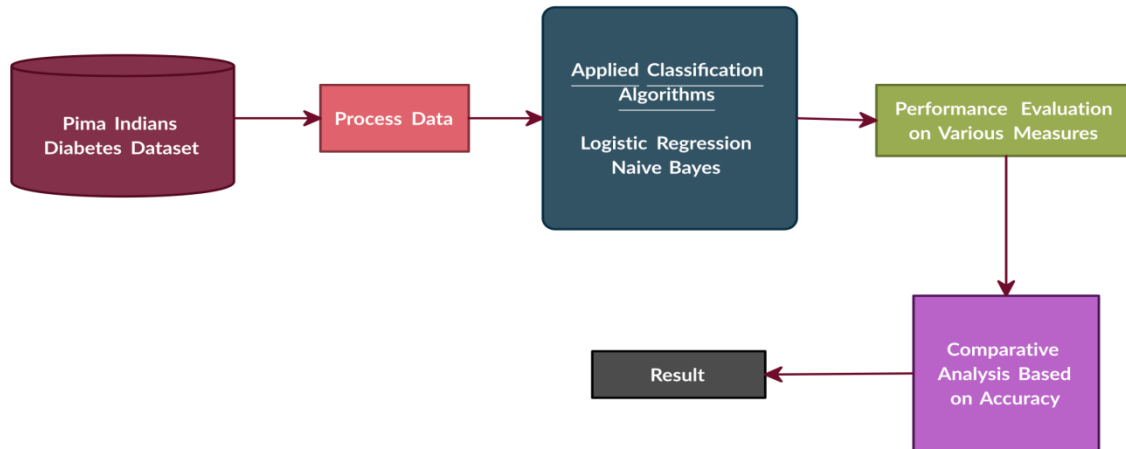


Figure 1.2.1 System Flow

1.3 Objectives

In this article, we aim to show the analytical results of how different physical factors and conditions can affect as well as give rise to the chances of diabetes considering the female population. In this, the analytical results of how different physical factors and conditions can contribute and give rise to the chances of diabetes considering the female population have been interpreted. Furthermore, we wanted to assess how classifiers built on the PIMA dataset perform on Indian diabetic patients. Our research objective is to answer the following research question: Can Machine Learning techniques be applied to predict the occurrence of diabetes among female patients? Can Machine Learning techniques be applied to predict diabetes in patients based on the PIMA Indian dataset? Data availability is a huge concern, predominantly noticed in developing countries. Accomplishment of the above-mentioned points can conclude that a single dataset taken from India can be used to train Machine Learning models which can then be applied to the female population in diagnosis and detection of diabetes. Thus the

problem of data unavailability can be resolved to some extent. This will give the mass population an in-depth knowledge and a close overview about the dependencies of various health conditions so that they can be aware and take necessary precautions in order to avoid the chances of diabetes occurring at an early age. The rest of the article is organized as follows

1.4 Application

Machine Learning (ML) has now become increasingly popular and has been reported as one of the most effective methods in a wide range of applications in preventive healthcare. It has associated advantages such as relatively low-cost computation, robustness, generalization ability and high performance. With the development of medical devices, equipment and tools, advanced knowledge can be gained in the disease diagnosis field. Computer-assisted decision making, i.e., Machine Learning aids humans by processing complex medical datasets and analyzing them to provide clinical insights. The knowledge extraction from data is a crucial factor for the prediction and diagnosis of disease in the medical industry. Through acquisition of required data and then necessary training and testing, some observations are obtained that can help reach a conclusion. In this article, we aim to detect whether any female patient residing in Bangladesh is suffering from diabetes or not. Inductive learning best suits this kind of work. In inductive learning, a learner learns some rules from observation of a set of instances. In inductive learning, the output can be predicted for new samples in the future through generalization and mapping. Machine Learning is one of the most conducive approaches to be applied for this.

2. DATA COLLECTION AND ANALYSIS

2.1 Dataset

The PIMA Indians Diabetes dataset, obtained from Kaggle, has been originally collected from the National Institute of Diabetes, Digestive and Kidney Diseases, India. The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age.

Table 2.1.1. Dataset summary.

Dataset	Number of instances	Number of features	Positive	Negative
PIMA	768	8	268	500

Columns

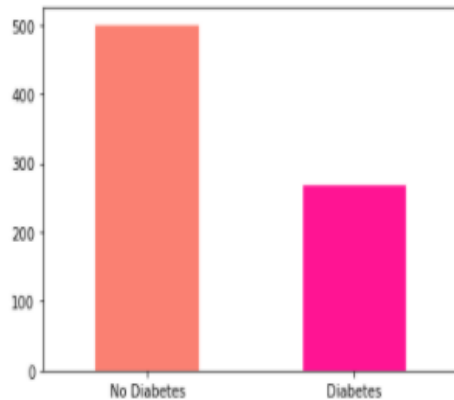
- **Pregnancies:** Number of times pregnant
- **Glucose:** Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- **blood pressure:** Diastolic blood pressure (mm Hg)
- **SkinThickness:** Triceps skinfold thickness (mm)
- **Insulin:** 2-Hour serum insulin (mu U/ml)
- **BMI:** Body mass index (weight in kg/(height in m)²)
- **DiabetesPedigreeFunction:** It provided some data on diabetes mellitus history in relatives and the genetic relationship of those relatives to the patient.
- **Age:** Age (years)
- **Outcome:** Class variable (0 or 1) 268 of 768 is 1, the others are 0

SL	Feature Name	Description	Min val	Max val	Mean
1	Number of pregnancy	Number of times pregnant	0	17	3.85
2	Glucose concentration	2-h oral glucose test (mg/dL)	0	199	120.89
3	Blood Pressure	Diastolic blood pressure (mm Hg)	0	122	69.11
4	Skin thickness	Triceps skin fold thickness (mm)	0	99	20.54
5	Serum Insulin	2-H serum insulin (mu U/mL)	0	846	79.80
6	BMI	Body mass index (kg/m ²)	0	67.10	31.99
7	Diabetes Pedigree Function	Diabetes in family history	0.08	2.42	0.47
8	Age	Age in Years	21	81	33.42

Table 2.1.2 Pima dataset features.

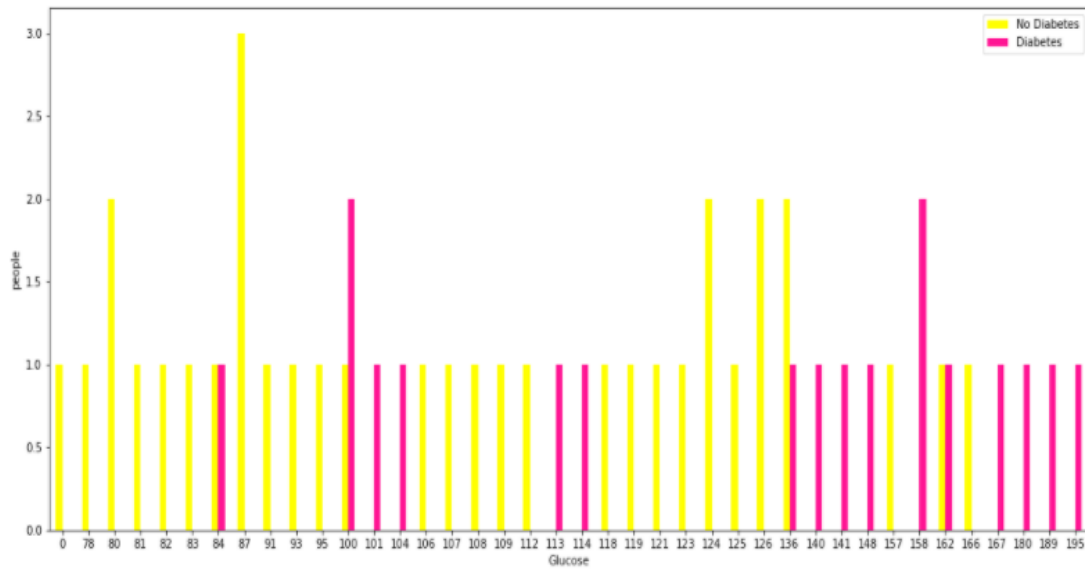
2.2 graphs representing the insights of the collected data.

```
In [6]: data["Outcome"].value_counts().plot(kind="bar",color=["salmon","deeppink"])
plt.xticks(np.arange(2), ('No Diabetes', 'Diabetes'),rotation=0);
```



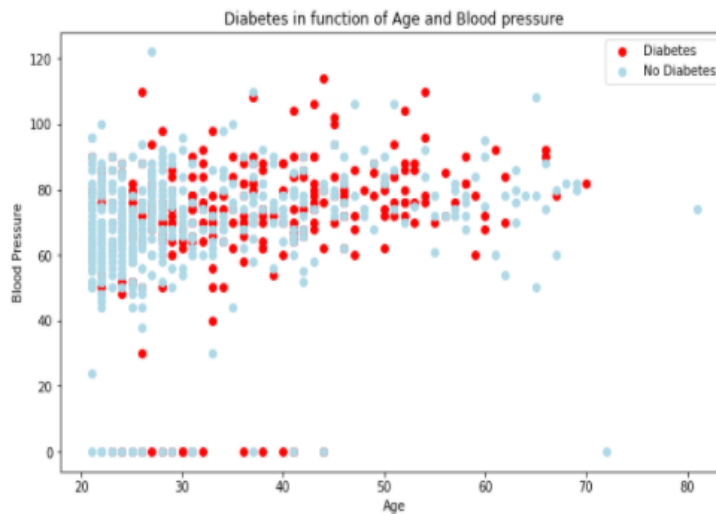
Screenshot 2.2.1 Diabetes and no Diabetes Patient

```
In [7]: # Comparing Glucose with the Outcome
pd.crosstab(data.Glucose[:,15],data.Outcome).plot(kind="bar",figsize=(18,8),color=["yellow","deeppink"])
plt.ylabel("people");
plt.xticks(rotation=0);
plt.legend(['No Diabetes', 'Diabetes']);
```



Screenshot 2.2.2 Compare Glucose with the outcome

```
In [8]: #find out Blood Pressure and age of entries who have diabetes
plt.figure(figsize=(10,6))
# Scatter with positive example
plt.scatter(data.Age[data.Outcome==1],data.BloodPressure[data.Outcome==1],c="Red");
# Scatter with negative example
plt.scatter(data.Age[data.Outcome==0],data.BloodPressure[data.Outcome==0],c="lightblue");
# Add some helpful info
plt.title("Diabetes in function of Age and Blood pressure")
plt.xlabel("Age")
plt.ylabel("Blood Pressure")
plt.legend(["Diabetes", "No Diabetes"]);
```



Screenshot 2.2.3 Blood Pressure and age of entries who have diabetes



Screenshot 2.2.4 Pair plotting of Data frame



Screenshot 2.2.5 Histogram of all columns when the outcome is 1

3.FINAL DESIGN AND IMPLEMENTATION

3.1 Model Implementation

In this Project, we have used Logistic Regression and Naïve Bayes to train, and then evaluated the classifiers to find the performance on PIMA Indians diabetes dataset. Implementation of the classifiers, calculation, and results generation have been conducted using Sklearn, a Python-based Machine Learning library while figure and graph generation have been done using Matplotlib and Seaborn library. After completing the pre-processing stage, the training set consisting of 613 instances was used to train the classifiers.

These classifiers have been explained shortly below:

Logistic Regression

In statistics Logistic regression is a regression model where the dependent variable is categorical, namely binary dependent variable-that is, where it can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick. Logistic regression is used in various fields, including machine learning, most medical fields, and social sciences. For example, the Trauma and Injury Severity Score (TRISS), which is widely used to predict mortality in injured patients, was originally developed using logistic regression. Many other medical scales used to assess severity of a patient have been developed using logistic regression.

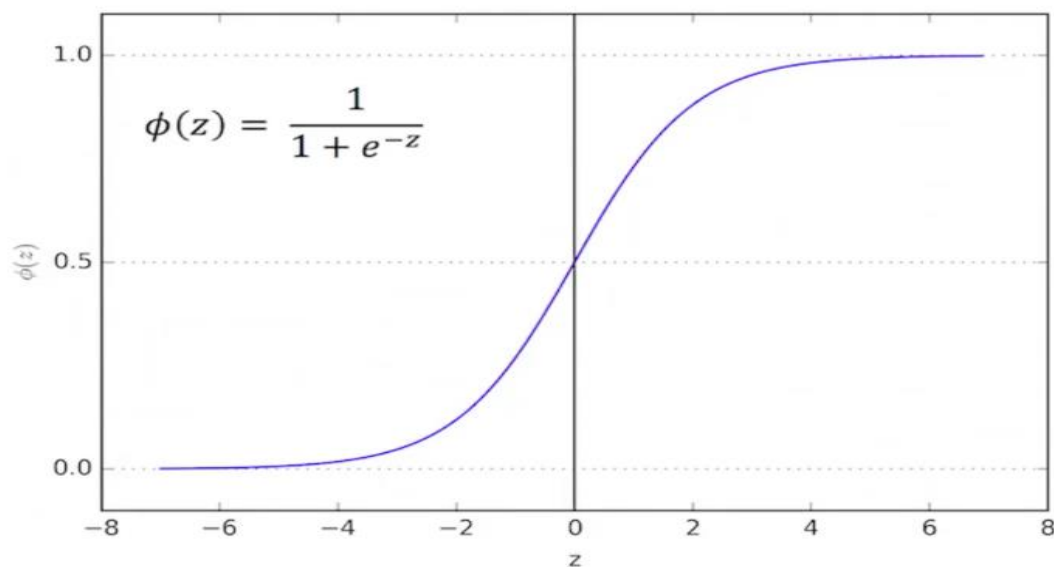


Figure 3.1.1 Logistic Regression Sigmoid Function

Naïve Bayes

Naïve Bayes classifier is a probabilistic classifier which is based on Bayes theorem with the independence assumption between the predictors. Naïve Bayesian method takes the dataset as input, performs analysis and predicts the class label using Bayes' Theorem. It calculates a probability of class in input data and helps to predict the class of the unknown data sample. It is a powerful classification technique suitable for large datasets. The Bayes Theorem formula calculates the posterior probability for each class using below formula.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} = \frac{P(X_1, X_2, \dots, X_n|Y)P(Y)}{P(X)}$$

Here Y is the class variable that we wanted to predict and X is a dependent vector of n attributes such that $X = \langle X_1, X_2, \dots, X_n \rangle$. Naïve Bayes classifier assumes that a predictor on a given class is independent of the values of other predictors. This conditional independency results in

$$P(X_1, X_2, \dots, X_n|Y) = \prod_{i=1}^n P(X_i | Y)$$

The naïve bayes classifier does not require any hyperparameter tuning.

The Platform used for creating the model is Anaconda Platform. Following packages are imported to create the model:

- Pandas
- Numpy
- Matplotlib
- Scikit-learn
- Seaborn

Requierments to create flask app are as given below:

- Flask==1.1.1
- Unicorn==19.9.0
- Jinja2==2.10.1
- MarkupSafe==1.1.1
- Werkzeug==0.15.5
- Numpy>=1.9.2
- Scipy>=0.15.1
- Scikit-learn>=0.18
- Matplotlib>=1.4.3
- Pandas>=0.19

3.2 Final Design

Following images shows the diabetes prediction application created using flask. There are 8 fields given named as Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction and Age. After entering all these values it will predict the probability of having diabetes by clicking the button “PREDICT PROBABILITY”.

Screenshot 3.2.1 Diabetes Prediction Application

Following Image shows Diabetes Prediction Application in which all the 8 fields are set with the values. These values are moderate that is neither too large nor too small. Hence the model predicts the output as “You are safe. Probability of having diabetes is 0.06”

Diabetes prediction

127.0.0.1:5000/predict

Diabetes Prediction Home

You are safe. Probability of having diabetes is 0.06

Diabetes Prediction

Predict the probability of having Diabetes

Pregnancies 1	Glucose 78	BloodPressure 50
SkinThickness 25	Insulin 88	BMI 26.6
DiabetesPedigreeFunction 0.248	Age 22	

PREDICT PROBABILITY

Type here to search

3:00 PM 12/20/2020

Screenshot 3.2.2 Diabetes Prediction Application Predicting “No diabetes”

Following Image shows Diabetes Prediction Application in which all the 8 fields are set with the values. These values are large than the excepted values to have a safe health report. Hence the model predicts the output as “You have chance of having diabetes. Probability of having diabetes is 0.63”.

Diabetes prediction

127.0.0.1:5000/predict

Diabetes Prediction Home

You have chance of having diabetes. Probability of having Diabetes is 0.63

Diabetes Prediction

Predict the probability of having Diabetes

Pregnancies 3	Glucose 234	BloodPressure 245
SkinThickness 32	Insulin 300	BMI 36
DiabetesPedigreeFunction 0.700	Age 22	

PREDICT PROBABILITY

Type here to search

2:58 PM 12/20/2020

Screenshot 3.2.3 Diabetes Prediction Application predicting Diabetes

4. PERFORMANCE ANALYSIS

Following figure displays the confusion matrices for PIMA indians diabetes dataset. Many valuable information can be extracted from a confusion matrix.

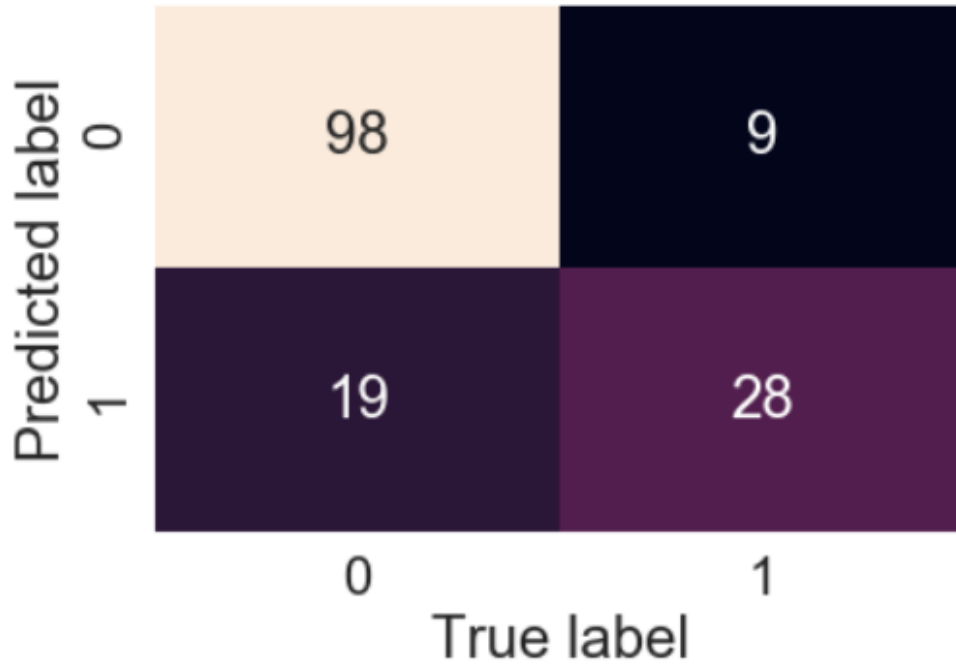


Figure 4.1 Confusion Matrix

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F1\ Score = \frac{2 * Recall * Precision}{Recall + Precision}$$

Here,

- TP = True Positive
- TN = True Negative
- FP = False Positive

- FN = False Negative

Following table shows the classification report of the diabetes prediction model.

Classification Report:				
	precision	recall	f1-score	support
0	0.84	0.92	0.88	107
1	0.76	0.60	0.67	47
accuracy			0.82	154
macro avg	0.80	0.76	0.77	154
weighted avg	0.81	0.82	0.81	154

Figure 4.2 Classification Report

ROC curve is a performance measurement for classification problem. It is plotted with True Positive Rate (TPR) against the False Positive Rate (FPR) where TPR is on y-axis and FPR is on the x-axis. The area under the ROC curve, called AUC, represents the degree of separability which provides a notion of how much a model is capable of distinguishing between classes. Higher AUC indicates better model at predicting classes.

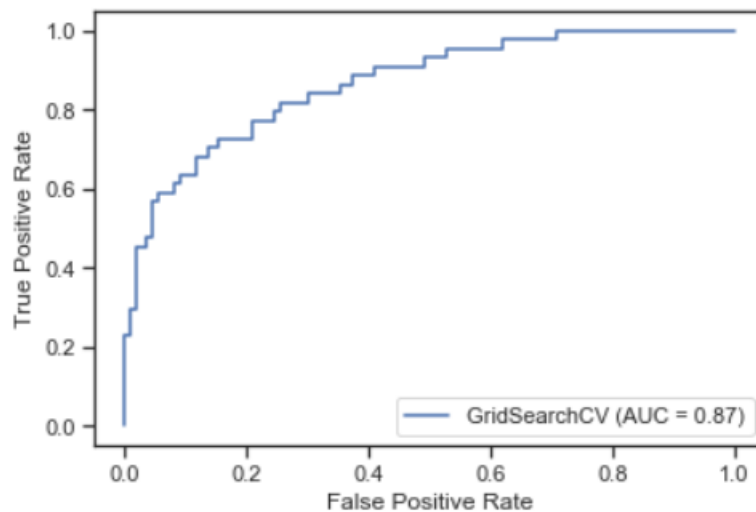


Figure 4.3 ROC Curve

5. Conclusion

5.1 Conclusion

Diabetes, although a non-communicable disease, is a serious condition with no cure. It can develop slowly in the body and further enhance the risk of other associated diseases. Obesity, chemical toxins in food, lack of physical exercise, sedentary lifestyle, and poor nutrition are all these risk factors for a person to suffer from diabetes eventually. Taking preventive measures and raising proper awareness can help reduce the risk of this complication. In a developing country, most of the people have poor knowledge regarding a healthy lifestyle and are unaware of the fact that they are already suffering from diabetes or developing the condition. Early prediction of diabetes will allow a patient to take necessary precautionary measures and control the condition from getting worse.

In order to answer the research question posed, a series of experiments have been conducted. The purpose of these experiments is to evaluate whether female patients having diabetes can be detected with high confidence using Machine Learning techniques. we have demonstrated that Machine Learning techniques are reliably effective in detecting diabetes. This infers that the unavailability of the dataset will not be a big issue if Machine Learning techniques can be well applied. We worked on a relatively small dataset.

One of the important real-world medical problems is the detection of diabetes at its early stage. In this study, systematic efforts are made in designing a system which results in the prediction of disease like diabetes. During this work, Two machine learning classification algorithms are studied and evaluated on various measures.

5.2 Future Scope

In future, the designed system with the used machine learning classification algorithms can be used to predict or diagnose other diseases. The work can be extended and improved for the automation of diabetes analysis including some other machine learning algorithms. plan to collect a more enriched dataset which will help us to predict diabetes detection with higher confidence. extend this work to evaluate how complex classifiers based on Artificial Neural Network (ANN) or other deep learning techniques perform when a certain dataset is used for training and another dataset is used for testing. This will help us understand whether such complex classifiers yield better predictions for such purposes. Furthermore, our idea can be generalized to any disease prediction, not just diabetes, especially if there is a data insufficiency problem.

Project Code

Importing Libraries

```
import numpy as np
import pandas as pd
np.random.seed(42)
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

#for warning

```
from warnings import filterwarnings
filterwarnings("ignore")
```

Load dataset

```
data=pd.read_csv("diabetes.csv")
```

Data Exploration

```
data.shape

len(data)

data.head()

data.tail()

data.isna().sum()

data.describe()

data["Outcome"].value_counts()

data["Outcome"].value_counts().plot(kind="bar",color=["salmon","deeppink"])

plt.xticks(np.arange(2), ('No Diabetes', 'Diabetes'),rotation=0);

data.info()
```

Comparing Glucose with Outcome

```
pd.crosstab(data.Glucose[:,15],data.Outcome).plot(kind="bar",figsize=(18,8),color=["yellow","deeppink"])

plt.ylabel("people");

plt.xticks(rotation=0);

plt.legend(['No Diabetes', 'Diabetes']);
```

Finding out the Blood Pressure and age of entries who have diabetes

```
plt.figure(figsize=(10,6))
```

Scatter with positive example

```
plt.scatter(data.Age[data.Outcome==1],data.BloodPressure[data.Outcome==1],c="Red");
```

Scatter with negative example

```
plt.scatter(data.Age[data.Outcome==0],data.BloodPressure[data.Outcome==0],c="lightblue");
```

Add some helpful info

```
plt.title("Diabetes in function of Age and Blood pressure")
```

```
plt.xlabel("Age")
```

```
plt.ylabel("Blood Pressure")
```

```
plt.legend(["Diabetes","No Diabetes"]);
```

Histogram of all columns when the Outcome is 1 [Diabetes]

```
fig, ax = plt.subplots(nrows=4, ncols=2, figsize=(12, 10))
```

```
fig.tight_layout(pad=3.0)
```

```
ax[0,0].set_title('Glucose')
```

```
ax[0,0].hist(data.Glucose[data.Outcome==1]);
```

```
ax[0,1].set_title('Pregnancies')
```

```
ax[0,1].hist(data.Pregnancies[data.Outcome==1]);
```

```
ax[1,0].set_title('Age')
```

```
ax[1,0].hist(data.Age[data.Outcome==1]);
```

```
ax[1,1].set_title('Blood Pressure')
```

```
ax[1,1].hist(data.BloodPressure[data.Outcome==1]);
```

```
ax[2,0].set_title('Skin Thickness')
```

```
ax[2,0].hist(data.SkinThickness[data.Outcome==1]);
```

```
ax[2,1].set_title('Insulin')
```

```
ax[2,1].hist(data.Insulin[data.Outcome==1]);
```



```
ax[3,0].set_title('BMI')

ax[3,0].hist(data.BMI[data.Outcome==1]);

ax[3,1].set_title('Diabetes Pedigree Function')

ax[3,1].hist(data.DiabetesPedigreeFunction[data.Outcome==1]);
```

#corelation matrix

```
data.corr()
```

make our correlation matrix visual

```
corr_matrix = data.corr()
```

```
fig,ax = plt.subplots(figsize=(15,10))
```

```
ax = sns.heatmap(corr_matrix, annot=True, linewidth=0.5, fmt=".2f", cmap="YlGnBu")
```

Modelling

#Evaluation

```
from sklearn.model_selection import train_test_split, cross_val_score
```

#Splitting the data

```
X = data.drop("Outcome",axis=1)
```

```
y = data["Outcome"]
```

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=0)
```

```
(len(X_train),len(X_test))
```

Build an Model (Naive Bayes)

```
from sklearn.naive_bayes import GaussianNB
model=GaussianNB()

model.fit(X_train , y_train)

y_pred=model.predict(X_test)

from sklearn import metrics
naive=metrics.accuracy_score(y_test,y_pred)
```

printing the accuracy

```
acc_nb = round( naive * 100, 2 )
print( 'Accuracy of Gaussian Naive Bayes model : ', acc_nb )
```

Build an Model (Logistic Regression)

```
from sklearn.linear_model import LogisticRegression
log_reg = LogisticRegression(random_state=0)

log_reg.fit(X_train,y_train);
```

Evaluating the model

```
log_reg = log_reg.score(X_test,y_test)
```

Printing Accuracy

```
acc_lg = round( log_reg * 100, 2 )
print( 'Accuracy of Logistic Regression model : ', acc_lg )
```

```
model_compare = pd.DataFrame({"Naive Bayes":naive,
                               "Logistic Regression":log_reg,
                               },index=["accuracy"])
```

```
model_compare
```

```
model_compare.T.plot.bar(figsize=(15,10));
```

Parameter Tuning

```
from sklearn.model_selection import RandomizedSearchCV,GridSearchCV
```

```

#Create a hyperparameter grid for LogisticRegression

log_reg_grid = {"C": np.logspace(-4, 4, 20),

               "solver": ["liblinear"]}}

# Tune LogisticRegression

log_reg_grid = {"C": np.logspace(-4,4,30),

               "solver":["liblinear"]}}


#setup the grid cv

gs_log_reg = GridSearchCV(LogisticRegression(),

                          param_grid=log_reg_grid,

                          cv=5,

                          verbose=True)


#fit grid search cv

gs_log_reg.fit(X_train,y_train)

gs_log_reg.score(X_test,y_test)

y_preds = gs_log_reg.predict(X_test)

y_preds

np.array(y_test)


# plot ROC curve

from sklearn.metrics import plot_roc_curve

plot_roc_curve(gs_log_reg,X_test,y_test)


#Confusion matrix

from sklearn.metrics import confusion_matrix, classification_report

print(confusion_matrix(y_test,y_preds))

sns.set(font_scale=2)

import seaborn as sns

sns.heatmap(confusion_matrix(y_test,y_preds), annot=True,cbar=False, fmt='g')

```

```

plt.xlabel("True label")

plt.ylabel("Predicted label");

print("Classification Report:")

print(classification_report(y_test, y_preds))

# Check best hyperparameters

gs_log_reg.best_params_

# Create a new classifier with best parameters

clf = LogisticRegression(C=0.20433597178569418,

                        solver="liblinear")

# Cross-validated accuracy

cv_acc = cross_val_score(clf, X, y, cv=10, scoring="accuracy")

cv_acc

cv_acc = np.mean(cv_acc)

cv_acc

# Cross-validated precision

cv_precision = cross_val_score(clf, X, y, cv=10, scoring="precision")

cv_precision=np.mean(cv_precision)

cv_precision

# Cross-validated recall

cv_recall = cross_val_score(clf, X, y, cv=10,scoring="recall")

cv_recall = np.mean(cv_recall)

cv_recall

# Cross-validated f1-score

cv_f1 = cross_val_score(clf,X, y, cv=10, scoring="f1")

cv_f1 = np.mean(cv_f1)

cv_f1

# Visualize cross-validated metrics

cv_metrics = pd.DataFrame({"Accuracy": cv_acc, "Precision": cv_precision, "Recall": cv_recall, "F1": cv_f1 },

```

```
index=[0])
```

```
cv_metrics.T.plot.bar(title="Cross-validated classification metrics",legend=False);
```

Fit an instance of LogisticRegression

```
clf = LogisticRegression(C=0.20433597178569418, solver="liblinear")
```

```
clf.fit(X_train, y_train);
```

```
clf.coef
```

```
feature_dict = dict(zip(data.columns, list(clf.coef_[0])))
```

```
feature_dict
```

Visualize feature importance

```
feature_df = pd.DataFrame(feature_dict, index=[0])
```

```
feature_df.T.plot.bar(title="Feature Importance", legend=False);
```

```
import pickle
```

Save trained model to file

```
pickle.dump(gs_log_reg, open("Diabetes.pkl", "wb"))
```

```
loaded_model = pickle.load(open("Diabetes.pkl", "rb"))
```

```
loaded_model.predict(X_test)
```

```
loaded_model.score(X_test,y_test)
```

Enter the new data

```
X_test.head(1)
```

```
Pregnancies = input()
```

```
Glucose = input()
```

```
BloodPressure = input()
```

```
SkinThickness = input()
```

```
Insulin = input()
```

```
BMI = input()
```

```
DiabetesPedigreeFunction = input()

Age = input()

row_df

prob = loaded_model.predict_proba(row_df)[0][1]

print(f"The probability of you having Diabetes is {prob}")

loaded_model.predict(row_df)[0]

#row_df.to_csv('./diabetes.csv', mode='a', header=False)
```

Github Link

https://github.com/swapnasalunke/ML_FLASKAPP

ACKNOWLEDGEMENT

We would like to place on record our deep sense of gratitude to Mr. Sanjay Kalyankar,
Head of Department Computer Science and Engineering, Deogiri Institute of Engineering

and management Studies Aurangabad, for his generous guidance, help and useful suggestions.

We express our sincere gratitude to Dr. Padmapani P. Tribhuvan, Dept. of Computer Science and Engineering, Deogiri Institute of Engineering and management Studies Aurangabad, for her stimulating guidance, continuous encouragement and supervision throughout the course of present work.

We are extremely thankful to Dr. Ulhas Shiurkar, Director, Deogiri Institute of Engineering and management Studies Aurangabad, for providing me infrastructural facilities to work in, without which this work would not have been possible.

Signature of Student

Vaishnavi Dattatraya Muley



Swapna Subhash Salunke

