Indians Diabetic Prediction

Predict the onset of diabetes based on diagnostic measures

Data Overview

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. URL for the dataset is
https://www.kaggle.com/kumargh/pimaindiansdiabetescsv
The Data Contains Mainly 9 columns and 768 records. Records contain several medical datas and a target varibale to predict whether diabetic present or not.

Objective of work

So what are the main objectives for this project. I have 2 objectives, first is to analyze the data, and see further information from the data to determine correlation between parameters and diabetes. The second is to attempt to get the best accuracy score using various supervised learning machine learning algorithms. This means to find out which algorithm is able to predict whether a Indian has diabetes or not based on this dataset.

Contents

The datasets consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

Variables

Variables are divided into two categories: X and y

X contains independet Variables and y contain target variable called dependent variable

Independed variables:

1.Number of times pregnant : Numeric data
2. Glucose Concentration : Numeric
3. Blood Pressure : Numeric
4. Skin thickness : Numeric
5. Insulin : Numeric
6. Body mass index : Numeric
7 .Diabetes Pedigree Function : Numeric

8. Age : Numeric

Dependent Variable

Diabetes outcome : boolean

Analysis of Data

	num_preg	glucose_conc	diastolic_bp	thickness	insulin	bmi	diab_pred	age	skin	diabetes
0	6	148	72	35	0	33.6	0.627	50	1.3790	True
1	1	85	66	29	0	26.6	0.351	31	1.1426	False
2	8	183	64	0	0	23.3	0.672	32	0.0000	True
3	1	89	66	23	94	28.1	0.167	21	0.9062	False
4	0	137	40	35	168	43.1	2.288	33	1.3790	True

The above screenshot is a view on how the data looks like when loaded into Pandas as a dataframe. Next, I will check if there are any missing data and account for any abnormalities in data. For example, in the above snapshot, for index no. 2, we can see that skinThickness=0, which cannot be the case in reality. As such, for each parameter with 0 value, I will be replacing it with the mean instead.

1. Import Libraries

Import all necessory libraries like pandas for reading datas,Numpy for creating arrays ,Seaborn,Matplot for visualization purpose etc.

2 . Load Data

Load dataset from location path as dataframe using Pandas

3. View Data

View data set using Head and Tail and verify whether data is correct or not. Check columns and rows count.

4. Data Preprocessing

Data Cleaning and EDA
1. Check whether have null values or not (using isna().sum())
2. Check datatypes of each variables
3. Maped boolen to numerical values [map_values = {True:1,False:0}]
4.Check whether data is imbalanced or not.(here data is 1:2 ratio)

5. Used imputation function to impute Zero values with mean values using SimpleImputer
6. Use heatmap to check whether have null values or not

5. Model Building

Split the dataset to independet and dependent variables and use train_test_split function to seperate the data to (70:30 ratio) training and testing

6. Algorithms

Random Forest Algorithm is used to check the accuracy.
For hyperparameter tuning used RandomizedSearchCv using Xgboost, and also verified with GridsearchCV

Result

There are no any null values in the data set. But have some zero values and had impute these values with mean values using SimpleImputer. Correlation was better. Using Heatmap check the null values present or not. Boolean values maped to numerical values. The data is balanced ratio 1:2. And finally split the test and train data set and perform training the dataset to predict the test data. Here applying RF algorithm in traing set and predicting test data(70% train data and 30% test data)

Used Random Forest Algorithm for training - Accuracy : 75%

For Hyperparameter tuning used RandomizedSearchCV with Xgboost -

Accuracy :77%

By using parameter tuning there is a little improvement on accuracy compared to RF.

(when used GridSearchCV for Hyperparameter tuning got only 74% accuracy. So for the best performnace ,used RandomizedSearchCV with XGboost).

Checked with other supevised learning algorithms like SVM,KNN,Logistic Regression but the accuracy was less compared to RF so here i chosen RF algorithm with parameter optimization for better performace

Model	Accuracy
Logistic Regression	74.67
Random Forest	75.3
Hyperparameter with Xgboost (Randomizedsearchcv)	77
SNN	67.57
KNN	73.4

Conclusion

Here i conlude that there is a relation between glucose and diabetic. So here Random forest is best algorithm with hyper parameter optimization.