Diabetes Prediction Using Machine Learning

Rakibul Hasan Rona
ID: 16155026
Department of IT,
University of Information
Technology and Sciences

Tarjuman Akter Liza
ID: 16155017
Department of IT,
University of Information
Technology and Sciences

Abstract:

Now a days Diabetes is a very common disease in all over the world of all age groups. To diagnose diabetes, a physician has to analyze many factors, data obtained from patients and has to take expert decisions that are very critical. But using Machine Learning technique it can be easily diagnose. In this paper we are going to discuss how to use Machine Learning to predict Diabetes. The prediction will predict whether a patient has a diabetes or not. We have used 'K-Nearest Neighbors' (KNN) machine learning algorithm for this work. We also measure the Accuracy of K-NN the algorithm and also compare it with another algorithm that is Decision Tree algorithm. Algorithms are works with python Machine learning modules like pandas, sklearn, matplotlib, Seaborn, numpy etc.

Keywords: Machine Learning, diabetes, prediction, KNN, Decision Tree, python, pnadas, sklearn, matplotlib, seaborn, numpy

INTORODUCTION:

425 million people have diabetes globally and 200 million of them do not even know they have the disease, according to the International Diabetes Federation (IDF-2017). Diabetes is a group of metabolic disease in which the body is unable to properly use and store glucose. Glucose backs up in the bloodstream causing one's blood glucose to rise too high. Symptoms of high blood sugar include frequent urination, increased thirst, and increased hunger. If left untreated, diabetes can cause many complications. Acute complications can include diabetic ketoacidosis, hyperosmolar hyperglycemic state, or death. Serious long-term complications include cardiovascular disease, stroke, chronic kidney disease, foot ulcers, and damage to the eyes.

The detection or diagnose of diabetes depends on the condition of Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Age etc. So it is hard for a doctor or physician to analysis those test reports and make a critical decision that a person has diabetes or not. Machine Learning can be help to analysis those reports and also used to make prediction that a person has a diabetes or not. In this paper we have briefly discussed about this Machine Learning project. We have developed a Machine Learning project that can predict about diabetes. We mainly follow K-NN algorithm. A data set is used to train the algorithm.

The FEATURES WE HAVE USED:

Technology:

- Anaconda (free open-source)
- Jupyter Notebook
- Python (3)

Data:

A dataset is use to analysis data and make a prediction. The dataset is also needed to train and test the algorithm. The dataset will be in csv form. The dataset we have used here is originally from the 'National Institute of Diabetes and Digestive and Kidney Diseases'. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. In this dataset, all patients here are females.

The datasets consist 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. The diabetes dataset consists of 768 data points, with 9 features each.

The 9 variables and it's meanings:

Pregnancies: Number of times pregnant

Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test

BloodPressure: Diastolic blood pressure (mm Hg)

SkinThickness: Triceps skin fold thickness (mm)

Insulin: 2-Hour serum insulin (mu U/ml)

BMI: Body mass index (weight in kg/(height in m)^2)

DiabetesPedigreeFunction: Diabetes pedigree function

Age: Age (years)

Outcome: Output variable (0 or 1)

ALGORITHM K-NN:

The k-NN algorithm is arguably the simplest machine learning algorithm. Building the model consists only of storing the training dataset. To make a prediction for a new data point, the algorithm finds the closest data points in the training dataset — its "nearest neighbors."

How it works:

1. Load the data
2. Initialize K to the chosen number of neighbors
3. For each example in the data
   3.1. Calculate the distance between the query example and the current example from the data.
   3.2. Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels
8. If classification, return the mode of the K labels

METHODOLOGY:

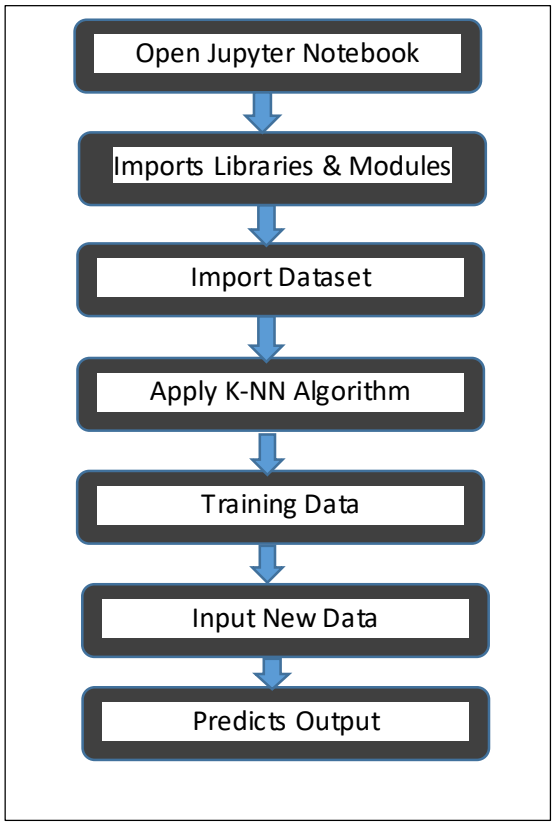


Fig 1: Working Block Diagram

Environment setup:

We have import some machine learning module and libraries:

```
import pandas as pd
import numpy as np
#!/usr/bin/env python
import matplotlib.pyplot as plt
```

Import Dataset and Analysis of Data:

```
diabetes=pd.read_csv('diabetes.csv')
print(diabetes.columns)
```

Output of the cell that will show columns index:

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

Show the dataset head (5 rows):

```
In [3]: diabetes.head()
```

Out[3]:

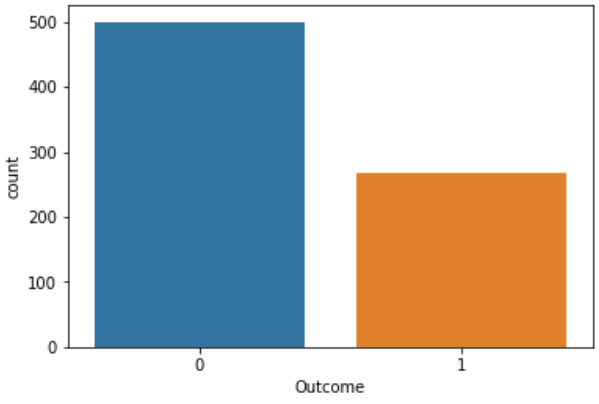
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Dimension of diabetes dataset:

```
print("dimension of diabetes data: {}".format(diabetes.shape))
dimension of diabetes data: (768, 9)
```

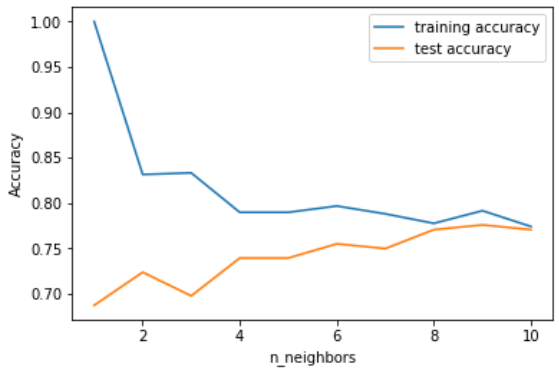
"Outcome" is the feature we are going to predict, 0 means No diabetes, 1 means diabetes. Of these 768 data points, 500 are labeled as 0 and 268 as 1:

```
print(diabetes.groupby('Outcome').size())
Outcome
0    500
1    268
dtype: int64
```



K-Nearest Neighbors Apply:

Visualization of training and test data accuracy:



As shown in the figure above, the plot shows the training and test set accuracy on the y-axis against the setting of neighbors on the x-axis. Imagine if we only choose one neighbor, the predictions in the training set are perfect. However, when more neighbors are added, the training accuracy will decrease, which means that the model obtained by selecting only one neighbor is too complicated. The best practice is to choose around 9 neighbors.

```
knn = KNeighborsClassifier(n_neighbors=9)
knn.fit(X_train, y_train)
print('Accuracy of K-NN classifier on training set: {:.2f}'.format(knn.score(X_trai
print('Accuracy of K-NN classifier on test set: {:.2f}'.format(knn.score(X_test, y_
```

Output:

Accuracy of K-NN classifier on training set: 0.79

Accuracy of K-NN classifier on test set: 0.78

Take user input to predict diabetes:

In this section of code, we have uses voice system that tells to give input. Here the program will take the report of 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age' as input.

The output of the code will predict whether a person has diabetes or not.

```
WelCome to Diabetes Prediction Software
Enter Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age 'with comma'
148,72,35,0,33.6,0.627,50
```

Output:

Analysis the user input the system will predict diabetes 'yes' or 'not,

If person have diabetes the output will this,

```
You Have Diabetes
You need to take this medicine
And immediate contact to your doctor
```

If the person has no diabetes output will this ,

```
You have not Diabetes
```

Now we will show the accuracy of another machine learning algorithm, 'Decision Tree' algorithm.

DECISION TREE:

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Accuracy Result:

The accuracy on the training set is 100%, while the test set accuracy is much worse. This is an indicative that the tree is overfitting and not generalizing well to new data. Therefore, we need to apply pre-pruning to the tree.

Accuracy on training set: 1.000

Accuracy on test set: 0.714

We set max_depth=3, limiting the depth of the tree decreases overfitting. This leads to a lower accuracy on the training set, but an improvement on the test set.

Accuracy on training set: 0.773

Accuracy on test set: 0.740

CONCLUSION AND FUTURE SCOPE:

In this paper we analyze the diabetes data and predicts about diabetes. We predict the outcome based on KNN algorithm and also compare the optimizations with Decision Tree Machine learning algorithms. Also implement different output accuracy in sikit learn modules.

REFRENCES:

1. https://www.kaggle.com/shriyaprasad/diabetes
2. Optimization of Diabetes Training DATA using Machine Learning Algorithms, DOI: 10.13140/RG.2.2.13442.58568
3. Improved J48 Classification Algorithm for the, International Journal of Computer Applications (0975 – 8887) Volume 98 – No.22, July 2014
4. A Machine Learning Approach to Predicting Blood Glucose Levels for Diabetes Management,
5. International Diabetes Federation: https://idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html
6. Machine Learning for Diabetes: https://towardsdatascience.com/machine-learning-for-diabetes-562dd7df4d42
7. hackernoon: https://hackernoon.com/ml-for-diabetes-from-bangladesh-d99d1d058d82
8. http://www.programmersought.com/article/7828640916/
9. k-nearest neighbors algorithm; https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
10. A Simple Introduction to K-Nearest Neighbors Algorithm:https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorithm-b3519ed98e
11. https://github.com/prakatheesh1234/Machine-learning-code-for-Diabetes/blob/master/main
12. https://www.youtube.com/watch?v=U1JIo8JSuYo&t=54s
13. https://www.youtube.com/watch?v=4HKqjENq9OU
14. https://classroom.google.com/u/0/c/MzU2NzU1Mjc1NTNa/m/MzU2NzU1Mjc1ODda/details