Classification of Diabetes

Contents

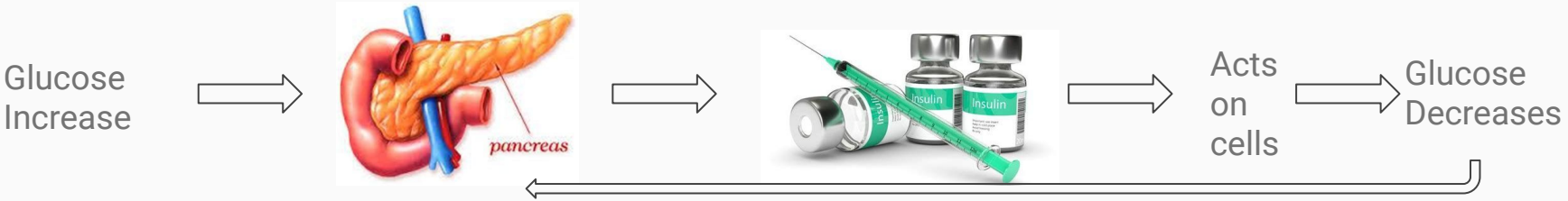
- Motivation
- What is Diabetes?
- Hardware & Software Requirements
- Methodology
- Algorithm Used
- Testing
- Conclusion
- References & Bibliography

Motivation

- Sedentary lifestyle of people, involving minimum physical activity.
- Early prediction of diabetes, can prevent adverse effects.
- Technology can be a reliable and efficient tool in prediction.

What is Diabetes?

Diabetes is a group of metabolic disorders characterized by a **high blood sugar level** over a prolonged period of time. Symptoms often include frequent **urination, increased thirst, and increased appetite**. If left untreated, diabetes can cause many complications.



Glucose Level:

A glucose level of less than **140 mg/dL** (7.8 mmol/L) is normal. A reading of **more than 200** mg/dL (11.1 mmol/L) **after two hours indicates diabetes**. A reading between **140 and 199 mg/dL** (7.8 mmol/L and 11.0 mmol/L) indicates **prediabetes**.

BMI:

Adult Body Mass Index (BMI) If your BMI is **less than 18.5**, it falls within the **underweight** range. If your BMI is **18.5 to <25**, it falls within the **normal**. If your BMI is **25.0 to <30**, it falls within the **overweight range**. If your BMI is 30.0 or higher, it falls within the **obese range**.

$$\text{BMI} = \frac{\text{weight (kg)}}{\text{height (m)}^2}$$

Hardware & Software Requirements

Processors : Intel Atom Or higher

Operating system : Windows 7 and Later, linux, MacOS .

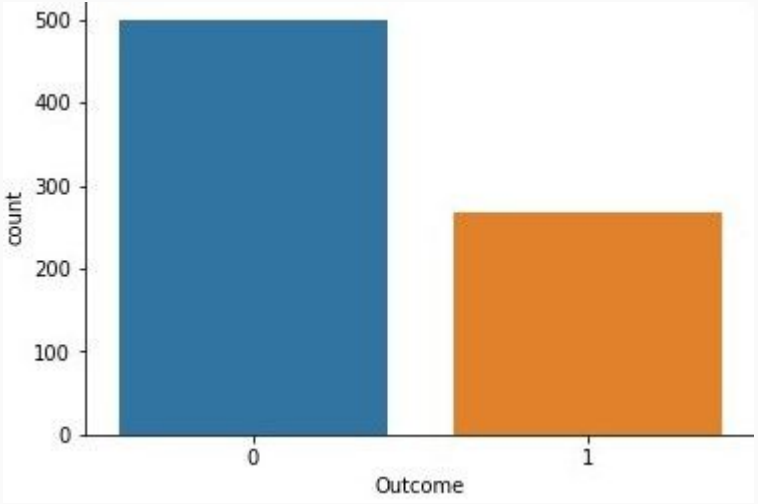
Python Version 2.7.x or later with pre-installed libraries here numpy, pandas,scikit-learn, flask, selenium, unittest

Methodology

1. **Data gathering and Importing libraries:** We used data set from here, it is a csv file which includes **Blood Pressure, Glucose, Insulin, BMI** as fields. We use **numpy** for linear algebra operations, **pandas** for using data frames, **matplotlib and seaborn** for plotting graphs.
2. **Descriptive Analysis:** Some features eg. Insulin, Glucose had zero values. Keeping values in int/float form. There are zero NaN entries.

Contd..

3. **Data Visualizations:** Plotting data wrt every column eg. in fig. 0 = Non Diabetic, 1 = Diabetic.



Contd..

4. **Data Preprocessing:** There were a number of zeroes in the independent dataset(dataset_X) hence we replaced those values with the mean.Glucose, Insulin, Age and BMI are highly correlated with the outcome. So, we select these features as X and the outcome as Y. The dataset is then split using train_test_split with an 80:20 ratio.

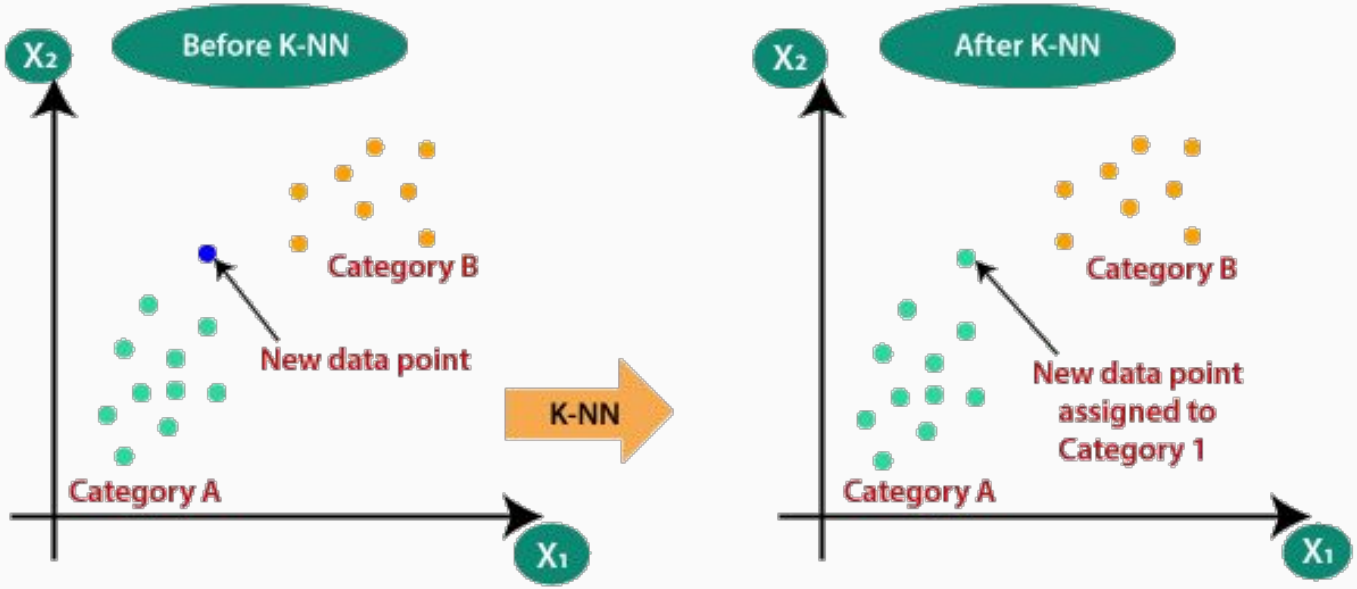
K-Nearest Neighbor(KNN) Algorithm

K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

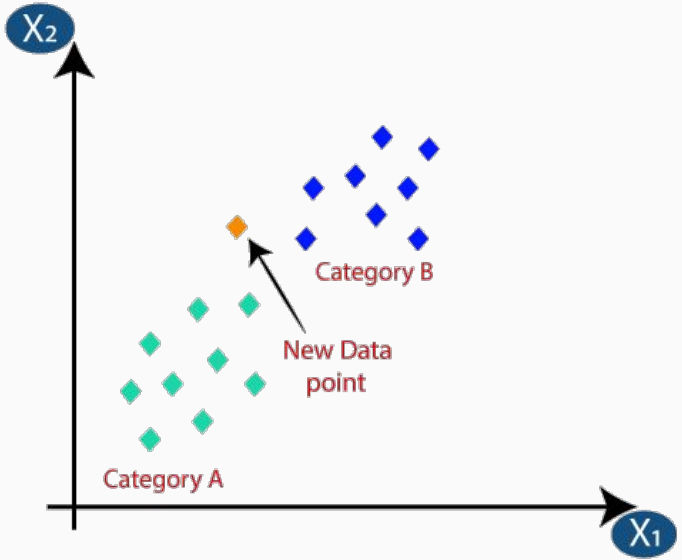
With the help of K-NN, we can easily identify the category or class of a particular dataset.

Contd..


Before K-NN
X2
Category B
New data point
Category A
X1
K-NN
After K-NN
X2
Category B
New data point
assigned to
Category 1
Category A
X1

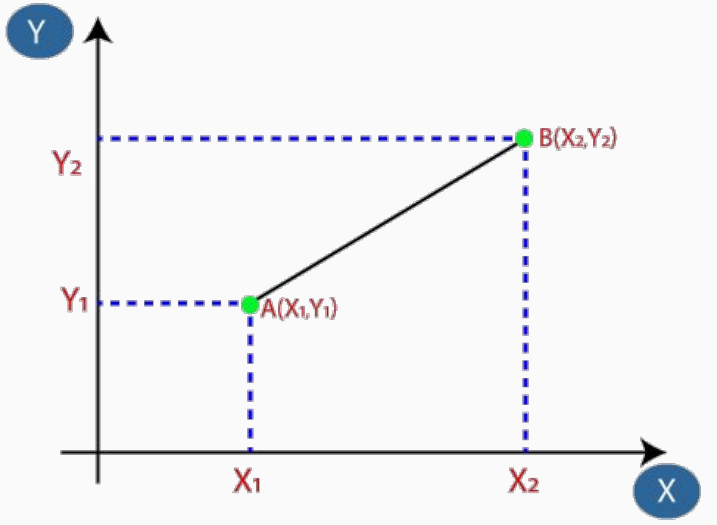
Working of KNN Algorithm

Step-1: Select the number K of the neighbors.



Contd..

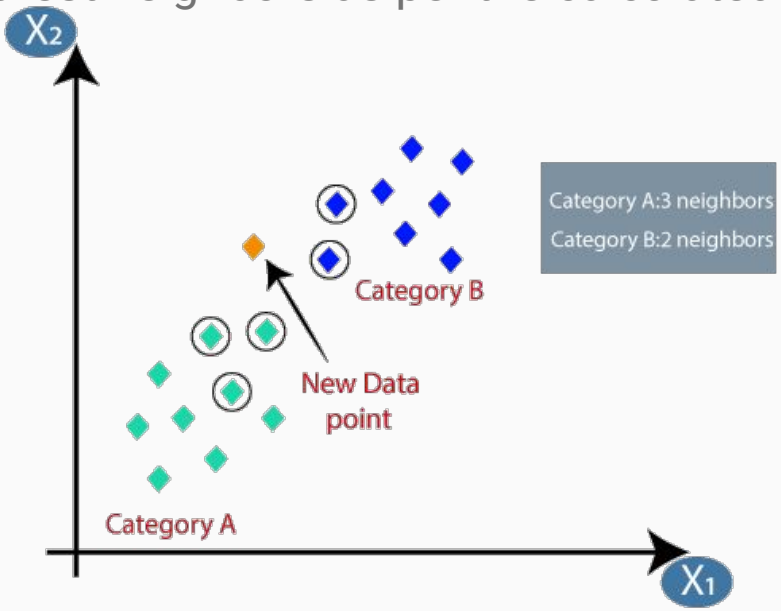
Step-2: Calculate the Euclidean distance of K number of neighbors.



Euclidean Distance between A_1 and $B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

Contd..

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.



Contd..

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

Advantages of KNN

1. It is simple to implement.
2. It is robust to the noisy training data.
3. It can be more effective if the training data is large.

Disadvantages of KNN

1. Always needs to determine the value of K which may be complex some time.
2. The computation cost is high because of calculating the distance between the data points for all the training samples.

Naïve Bayes Algorithm

Naïve Bayes method is the **probabilistic classifier algorithm** based on Baye's theorem. It works on **conditional probability**. The presence of a certain feature is independent on the presence of other features; hence it is called as Naïve. The models built are faster, particularly useful for very large data sets.

Contd..

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Types of Naïve Bayes Model:

Gaussian: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution. We have used gaussian method.

Multinomial: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed.

Bernoulli: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables

Working of Naïve Bayes Algorithm

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.

Advantages of Naïve Bayes

1. Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
2. It can be used for Binary as well as Multi-class Classifications.
3. It performs well in Multi-class predictions as compared to the other Algorithms.

Disadvantages of Naïve Bayes

1. Naïve Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.
2. If your test data set has a categorical variable of a category that wasn't present in the training data set, the Naïve Bayes model will assign it zero probability and won't be able to make any predictions in this regard. This phenomenon is called **'Zero Frequency**,' and you'll have to use a smoothing technique to solve this problem.

DIABETES PREDICTION

Glucose Level

Insulin

BMI

Age

SVC Predict

Naive Bayes Predict

DIABETES PREDICTION

148

0

33.6

50

SVC Predict | Naive Bayes Predict

SVC : You have Diabetes, please consult a Doctor.

DIABETES PREDICTION

85

0

22.6

31

SVC Predict

Naive Bayes Predict

Naive Bayes : You don't have Diabetes.

Testing

Unit Testing:

Unit testing is a software testing method by which individual units of source code are tested to determine whether they are fit for use.

Selenium is one of the most popular **automation testing tools**. Here automation testing is a process of **converting** any **manual test case into the test scripts** using automation tools such as Selenium.

For our testing purpose, we are creating our **sample test script using selenium tool** in python programming language.

Contd..

setUpClass()

A class method is called before any of the individual tests are called. **@classmethod** is the identifier with which you can identify a setUpClass(). setUpClass() has **only one argument** i.e. the **class name.**

tearDownClass()

This method is called **after all the tests in the class are executed**. Similar to setUpClass(), tearDownClass() also as **only one argument** i.e. the class name. The **'class name' should match** with the name which is **used in the setUpClass(),** else it might result in an error.

Contd..

Test Runner

Responsible for displaying the output of the executed test to an end user using a runnable interface.

Test Case

Test Case contains the actual implementation of the test code.

TestCase class is used to **create new tests** and the FunctionTestCase acts as a subclass to TestCase class and make use of tests which are appended to the existing unittest framework. **setUp()** and **tearDown()** are **important components** of the TestCase class that are used for **initialization & cleanup** of the test fixture (that was created using setUp()).

~/Desktop/diabetes_prediction — python • python wsgi.py

~/Desktop/diabetes_prediction — -zsh

```
(base) prashantkumar@Prashants-MacBook-Pro diabetes_prediction % python test.py

Running tests...
----------------------------------------------------------------------
 test_search_1 (__main__.Prediction) ... OK (15.074995)s
 test_search_2 (__main__.Prediction) ... OK (10.244029)s
 test_search_3 (__main__.Prediction) ... OK (8.262474)s
 test_search_4 (__main__.Prediction) ... OK (8.241232)s
 test_search_5 (__main__.Prediction) ... OK (8.260846)s
 test_search_6 (__main__.Prediction) ... OK (8.219738)s
 test_search_7 (__main__.Prediction) ... OK (8.236486)s
 test_search_8 (__main__.Prediction) ... OK (8.248508)s
 test_search_9 (__main__.Prediction) ... OK (8.257970)s

----------------------------------------------------------------------
Ran 9 tests in 0:01:24

OK


Generating HTML reports...
reports/TestResults___main__.Prediction_2020-12-28_19-38-37.html
(base) prashantkumar@Prashants-MacBook-Pro diabetes_prediction %
```

Unittest Results

Start Time: 2020-12-28 19:38:37

Duration: 83.05 s

Summary: Total: 9, Pass: 9

__main__.Prediction	Status
test_search_1	Pass
test_search_2	Pass
test_search_3	Pass
test_search_4	Pass
test_search_5	Pass
test_search_6	Pass
test_search_7	Pass
test_search_8	Pass
test_search_9	Pass

Total: 9, Pass: 9 -- Duration: 83.05 s

Conclusion

We have used accuracy score as metric for evaluating our model. The model has accuracy of **77%** by KNN algorithm and of **80%** by Naïve bayes algorithm which is quite satisfactory. Similar models can be used to predict other diseases like breast cancer, malaria etc.

References & Bibliography

Dataset:
https://www.kaggle.com/uciml/pima-indians-diabetes-database

K-Nearest Neighbour Images:
https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

Other Images: https://images.google.com/

Naive Bayes Algorithm:

https://www.javatpoint.com/machine-learning-naive-bayes-classifier

K-Nearest Neighbour Algorithm:
https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

Thanks

Aniruddh Bhagwat
405B005

Manav Chouhan
405B018

Rohit Kshirsagar
405B036

Prashant Kumar
405B049

BE - II
(Computer)