

# Comparative Approaches for Classification of Diabetes Mellitus Data

Study of machine learning algorithms and application on the Pima Indians Diabetes  
Dataset.

**Alexander Roque Rodrigues**

A dissertation presented for the degree of  
Bachelor in Computer Science Department of Computer Science

Smt. Parvatibai Chowgule College of Arts and Science  
India

18 August 2019

# Contents

|            |  |           |
|------------|--|-----------|
| <b>I</b>   | <b>Acknowledgements</b>                            | <b>3</b>  |
| <b>II</b>  | <b>List of Abbreviations</b>                       | <b>4</b>  |
| <b>III</b> | <b>Introduction</b>                                | <b>5</b>  |
| <b>IV</b>  | <b>Review of Literature</b>                        | <b>6</b>  |
| <b>V</b>   | <b>Machine Learning Algorithms</b>                 | <b>8</b>  |
| 1          | Linear Regression                                  | 8         |
| 2          | Logistic Regression                                | 9         |
| 3          | K-Nearest Neighbours                               | 10        |
| 3.1        | Finding Optimum Number of Clusters . . . . .       | 10        |
| 3.2        | Implementation . . . . .                           | 10        |
| 4          | Decision Tree                                      | 11        |
| 4.1        | Introduction . . . . .                             | 11        |
| 5          | Random Forest                                      | 13        |
| 6          | Gradient Boosting                                  | 14        |
| 7          | Support Vector Machine                             | 15        |
| 7.1        | Linear Support Vector Machines . . . . .           | 15        |
| 8          | Perceptron   | 16        |
| 9          | Multilayered Perceptron                            | 17        |
| <b>VI</b>  | <b>Real World Application</b>                      | <b>18</b> |
| <b>VII</b> | <b>Building Information Systems for Prediction</b> | <b>20</b> |
| 1          | Introduction                                       | 20        |
| 2          | Feature Selection                                  | 20        |
| 3          | Models   | 21        |
| 4          | Conclusion   | 22        |

|             |                     |           |
|-------------|---------------------|-----------|
| <b>VIII</b> | <b>Conclusion</b>   | <b>23</b> |
| <b>IX</b>   | <b>Bibliography</b> | <b>24</b> |

## Part I

# Acknowledgements

I, Alexander Roque Rodrigues, would like to acknowledge the role of the Department of Computer Science in guiding me and helping me to achieve the completion of this project. I would also like to acknowledge the efforts put in by my project guide Ms. Ashweta Fondekar, who assisted me throughout the span of the project and helped me achieve my goal. I also would like to thank my parents for their support and others who have indirectly assisted me in this project.

## Part II

# List of Abbreviations

|      |   |
|------|---|
| T1D  | Type 1 Diabetes.                          |
| T2D  | Type 2 Diabetes.                          |
| GDM  | Gestational Diabetes.                     |
| AEE  | Activity Energy Expenditure.              |
| CSII | Continuous Subcutaneous Insulin Infusion. |
| MDI  | Multiple Daily Injections.                |

## Part III

# Introduction

Data is everywhere. International initiatives coupled with global disruptive innovation are the leading causes for pushing datafication forward. Datafication refers to the modern-day trend of digitalizing (or datafying) every aspect of life. This data creation is enabling the transformation of data into new and potentially valuable forms. Entire municipalities are being incentivized to become smarter. In the not too distant future, our towns and cities will collect thousands of variables in real time to optimize, maintain, and enhance the quality of life for entire populations. One would reasonably expect that as well as managing traffic, traffic lights may also collect other data such as air quality, visibility, and speed of traffic. As a result of big data from connected devices, embedded sensors, and the IoT, there is a global need for the analysis, interpretation, and visualization of data.

## Part IV

# Review of Literature

As defined by the American Diabetes Association[1],

Diabetes is a group of metabolic diseases characterized by hyperglycemia resulting from defects in insulin secretion, insulin action, or both. The chronic hyperglycemia of diabetes is associated with long-term damage, dysfunction, and failure of different organs, especially the eyes, kidneys, nerves, heart, and blood vessels.

Diabetes Mellitus refers collectively to a group of diseases resulting from dysfunction of the glucoregulatory system. Hyperglycemia, the hallmark of diabetes, is the primary consequence of this dysregulation. Chronic hyperglycemia in diabetes is associated with long-term complications involving tissue damage and organ failure, which can decrease life expectancy and even cause death. The International Diabetes Federation estimates that, by 2017, diabetes affected 425 million people worldwide, of whom, 4 million died in the same year. These figures are expected to increase dramatically in the coming decades, placing a rising burden on health care systems. According to the growing morbidity in recent years, in 2040, the world's diabetic patients will reach 642 million, which means that one of the ten adults in the future is suffering from diabetes. There is no doubt that this alarming figure needs great attention. With the rapid development of machine learning, machine learning has been applied to many aspects of medical health.[2]

Most diabetes cases can be categorized into 3 subgroups:

1. T1D
2. T2D
3. GDM

Over the long term, T2D patients become resistant to the normal effects of insulin and gradually lose their capacity to produce enough of this hormone. A wide range of therapeutic options are available for patients with T2D. At the early stages of disease, they commonly receive medications that improve insulin secretion or insulin absorption, but eventually they must receive external doses of insulin. On the other hand, T1D patients have severe impairments in insulin production, and must use external insulin exclusively to manage their blood glucose. Treatment of T1D requires consistent doses of insulin through MDI or CSII using a pump. GDM is treated similarly to T2D, but only occurs during pregnancy due to the interaction between insulin and hormones released by the placenta.

In each class of diabetes, timely diagnosis, education of patients in self-management, and continuous medical care are required to prevent acute complications (eg, ketoacidosis) and minimize the risk of long-term complications (eg, nephropathy, retinopathy, diabetic foot, cardiovascular disease, or stroke). In addition to medication, management of diabetes requires adherence to an array of self-care behaviors that are often very burdensome for patients: carefully scheduling meals, counting carbohydrates, exercising, monitoring BG levels, and adjusting endeavors on a daily basis. The effects of

non-adherence to recommended treatment are not immediately evident and long-term complications may take years to develop. Accordingly, diabetes therapy is complex, and therapeutic decisions need to take into account diverse medical factors and lifestyle-related activities that must be optimized to improve diabetic patients' quality of life.



## Part V

# Machine Learning Algorithms

## 1 Linear Regression

Linear regression is a forecasting technique that can be used to predict the future of a number series based on the historic data given. The perks of using a linear regression model are as follows:

- produces decent and easy to interpret results.
- is computationally inexpensive.
- conversion of algorithm into code does not take much effort or time.
- numeric values as well as nominal values support is offered.

However, a major drawback of linear regression is that it **poorly models nonlinear data**.

Considering a dataset that has values ranging from  $X = \{x_1 + x_2 + x_3 + \dots + x_n\}$  where all the entries of the dataset are real numbers. Each  $x_i$  is associated with a corresponding value of  $y_i$  from the dataset  $Y = \{y_1 + y_2 + y_3 + \dots + y_n\}$ .

The most basic equation for linear regression can be expressed via this simple equation.

$$y = \beta_0 x + \beta_1 + \epsilon$$

So to minimize the error in the predictions, a way to calculate the error should be formulated. A loss function in machine learning is simply a measure of how different the predicted value is from the actual value. The Quadratic Loss Function to calculate the loss or error in our linear regression model. It can be defined as:

$$L(x) = \sum_{i=1}^n (y_i - p_i)^2$$

Therefore using the method of Least Squares, we can find the values of  $\beta_0$  and  $\beta_1$ .

$$\beta_0 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

The linear regression model with an error value close to 1.00 indicates a perfect model and those with values closer to 0.00 indicates a model that delivers poor performance.

## 2 Logistic Regression

Logistic Regression is a classification method which is based on the probability for a sample to belong to a class. As our probabilities must be continuous in  $R$  and should be bounded between the lower limit of 0 and upper limit of 1 it's necessary to introduce a threshold function to filter the term  $z$ . The name logistic comes from the decision to use the sigmoid (or logistic) function, defined as shown below.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

The sigmoid function in its domain is  $R$  has two asymptotes<sup>1</sup> that occur at 0 and 1. So, the probability for a sample to belong to a class from 0 and 1 can be defined as:

$$P(y|\bar{x}) = \sigma(\bar{x}; \bar{w})$$

At this point, finding the optimal parameters is equivalent to maximizing the log-likelihood<sup>2</sup> given the output class:

$$L(\bar{w}; y) = \log P(y|\bar{w}) = \sum_i \log P(y_i|\bar{x}_i, \bar{w})$$

---

<sup>1</sup>In analytic geometry, an asymptote of a curve is a line such that the distance between the curve and the line approaches zero as one or both of the x or y coordinates tends to infinity.

<sup>2</sup>The log-likelihood is, as the term suggests, the natural logarithm of the likelihood.

## 3 K-Nearest Neighbours

The k-means algorithm is based on the (strong) initial condition to decide the number of clusters through the assignment of k initial centroids or means:

Then the distance between each sample and each centroid is computed and the sample is assigned to the cluster where the distance is minimum. This approach is often called minimizing the inertia of the clusters, which is defined as follows:

The process is iterative—once all the samples have been processed, a new set of centroids K (1) is computed (now considering the actual elements belonging to the cluster), and all the distances are recomputed. The algorithm stops when the desired tolerance is reached, or in other words, when the centroids become stable and, therefore, the inertia is minimized. Of course, this approach is quite sensitive to the initial conditions, and some methods have been studied to improve the convergence speed. One of them is called k-means++ (Karteeika Pavan K., Allam Appa Rao, Dattatreya Rao A. V., and Sridhar G.R., Robust Seed Selection Algorithm for K-Means Type Algorithms, International Journal of Computer Science and Information Technology 3, no. 5, October 30, 2011), which selects the initial centroids so that they are statistically close to the final ones. The mathematical explanation is quite difficult; however, this method is the default choice for scikit-learn, and it's normally the best choice for any clustering problem solvable with this algorithm.

### 3.1 Finding Optimum Number of Clusters

One of the most common disadvantages of k-means is related to the choice of the optimal number of clusters. An excessively small value will determine large groupings that contain heterogeneous elements, while a large number leads to a scenario where it can be difficult to identify the differences among clusters. Therefore, we're going to discuss some methods that can be employed to determine the appropriate number of splits and to evaluate the corresponding performance.

### 3.2 Implementation

```
1 from sklearn.neighbors import KNeighborsClassifier
2 training_accuracy = []
3 test_accuracy = []
4 neighbors_settings = range(1, 11)
5
6 for n_neighbors in neighbors_settings:
7     # build the model
8     knn = KNeighborsClassifier(n_neighbors=n_neighbors)
9     # train the model using the train data
10    knn.fit(X_train, y_train)
11    # record training set accuracy
12    training_accuracy.append(knn.score(X_train, y_train))
13    # record test set accuracy
14    test_accuracy.append(knn.score(X_test, y_test))
```

## 4 Decision Tree

### 4.1 Introduction

Decision trees are flowcharts that represent the decision-making process as rules for performing categorization. Decision trees start from a root and contain internal nodes that represent features and branches that represent outcomes. As such, decision trees are a representation of a classification problem. Decision trees can be exploited to make them easier to understand. Each decision tree is a dis-junction of implications (i.e., if-then statements), and the implications are Horn clauses that are useful for logic programming. A Horn clause is a dis-junction of literals. On the basis that there are no errors in the data in the form of inconsistencies, we can always construct a decision tree for training datasets with 100% accuracy. However, this may not roll out in the real world and may indicate overfitting, as we will discuss.

Classifying an example involves subjecting the data to an organized sequence of tests to determine the label. Trees are built and tested from top to bottom as so:

1. Start at the root of the model
2. Wait until all examples are in the same class
3. Test features to determine best split based on a cost function
4. Follow the branch value to the outcome
5. Repeat number 2
6. Leaf node output

The central question in decision tree learning is which nodes should be placed in which positions, including the root node and decision nodes. There are three main decision tree algorithms. The difference in each algorithm is the measure or cost function for which nodes, or features, are selected. The root is the top node. The tree is split into branches; evaluated through a cost function; and a branch that doesn't split is a terminal node, decision, or leaf.

Decision trees are useful in the way that acquired knowledge can be expressed in an easy to read and understandable format (see Figure 4-1). It mimics human decision-making whereby priority—determined by feature importance, relationships, and decisions—is clear. They are simple in the way that outcomes can be expressed as a set of rules. Figure 4-1. Decision tree of  $n = 2$  nodes Decision trees provide benefits in how they can represent big datasets and prioritize the most discriminatory features. If a decision tree depth is not set, it will eventually learn the data presented and overfit. It is recommended to set a small depth for decision tree modeling. Alternatively, the decision tree can be pruned, typically starting from the least important feature, or the incorporation of dimensionality reduction techniques.

Overfitting is a common machine learning obstacle, and not limited to decision trees. All algorithms are at risk of overfitting, and a variety of techniques exist to overcome this problem. Random forest or jungle decision trees can be extremely useful in this. Pruning reduces the size of a decision tree by removing features that provide the least information. As a result, the final classification rules are less complicated and improve predictive accuracy. The accuracy of a model is calculated as the percentage of examples in the test dataset that is classified correctly.

- True Positive: Where the actual class is yes, and the value of the predicted class is also yes.
- False Positive: Actual class is no, and predicted class is yes.
- True Negative: The value of the actual class is no, and the value of the predicted class is no.
- False Negative: When the actual class value is yes, but predicted class is no.

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

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

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

Classification is a common method used in machine learning; and ID3 (Iterative Dichotomizer 3), C4.5 (Classification 4.5), and CART (Classification And Regression Trees) are common decision tree methods where the resulting tree can be used to classify future samples.

## 5 Random Forest

## 6 Gradient Boosting

## 7 Support Vector Machine

Support Vector Machines is an algorithm that is capable of handling linear as well as data that occurs non-linearly. For example, for a long time, SVMs were the best choice for MNIST dataset classification, thanks to the fact that they can capture very high non-linear dynamics using a mathematical trick, without complex modifications in the algorithm.

### 7.1 Linear Support Vector Machines

Let us consider a dataset of features we want to classify.

$$X = \{x_1, x_2, x_3, \dots, x_n\}$$

For the target variable, we will consider the dataset  $Y$ , with target outcomes as  $\{0, 1\}$  indicating a true or false condition.

$$Y = \{y_1, y_2, y_3, \dots, y_n\}$$



## 8 Perceptron

## 9 Multilayered Perceptron

## Part VI

# Real World Application

This is the part on how the project results were obtained.

| Algorithm              | Additional Parameters                              | Train Set Accuracy | Test Set Accuracy |
|------------------------|--|--------------------|-------------------|
| K Nearest Neighbour    | -  | 0.79               | 0.78              |
| Logistic Regression    | $C = 1$  | 0.781              | 0.771             |
| Logistic Regression    | $C = 0.01$   | 0.700              | 0.703             |
| Logistic Regression    | $C = 100$  | 0.785              | 0.766             |
| Decision Tree          | -  | 1.00               | 0.714             |
| Decision Tree          | Max Depth = 3                                      | 0.773              | 0.740             |
| Random Forest          | Estimators = 100                                   | 1.000              | 0.786             |
| Random Forest          | Estimators = 100; Max Depth = 3                    | 0.800              | 0.755             |
| Gradient Boosting      | -  | 0.917              | 0.792             |
| Gradient Boosting      | Max Depth = 1                                      | 0.804              | 0.781             |
| Gradient Boosting      | Learning Rate = 0.01                               | 0.802              | 0.776             |
| Support Vector Machine | -  | 1.00               | 0.65              |
| Support Vector Machine | Train and Test set scaled using MinMaxScaler       | 0.77               | 0.77              |
| Support Vector Machine | $C = 1000$   | 0.790              | 0.797             |
| MLP Classifier         | Random State = 42                                  | 0.73               | 0.72              |
| MLP Classifier         | Random State = 0                                   | 0.823              | 0.802             |
| MLP Classifier         | Max Iterations = 1000                              | 0.908              | 0.792             |
| MLP Classifier         | Max Iterations = 1000; Alpha = 1; Random State = 0 | 0.806              | 0.797             |

Table 1: Test and Train accuracy's using various machine learning algorithms with various parameters.

## **Part VII**

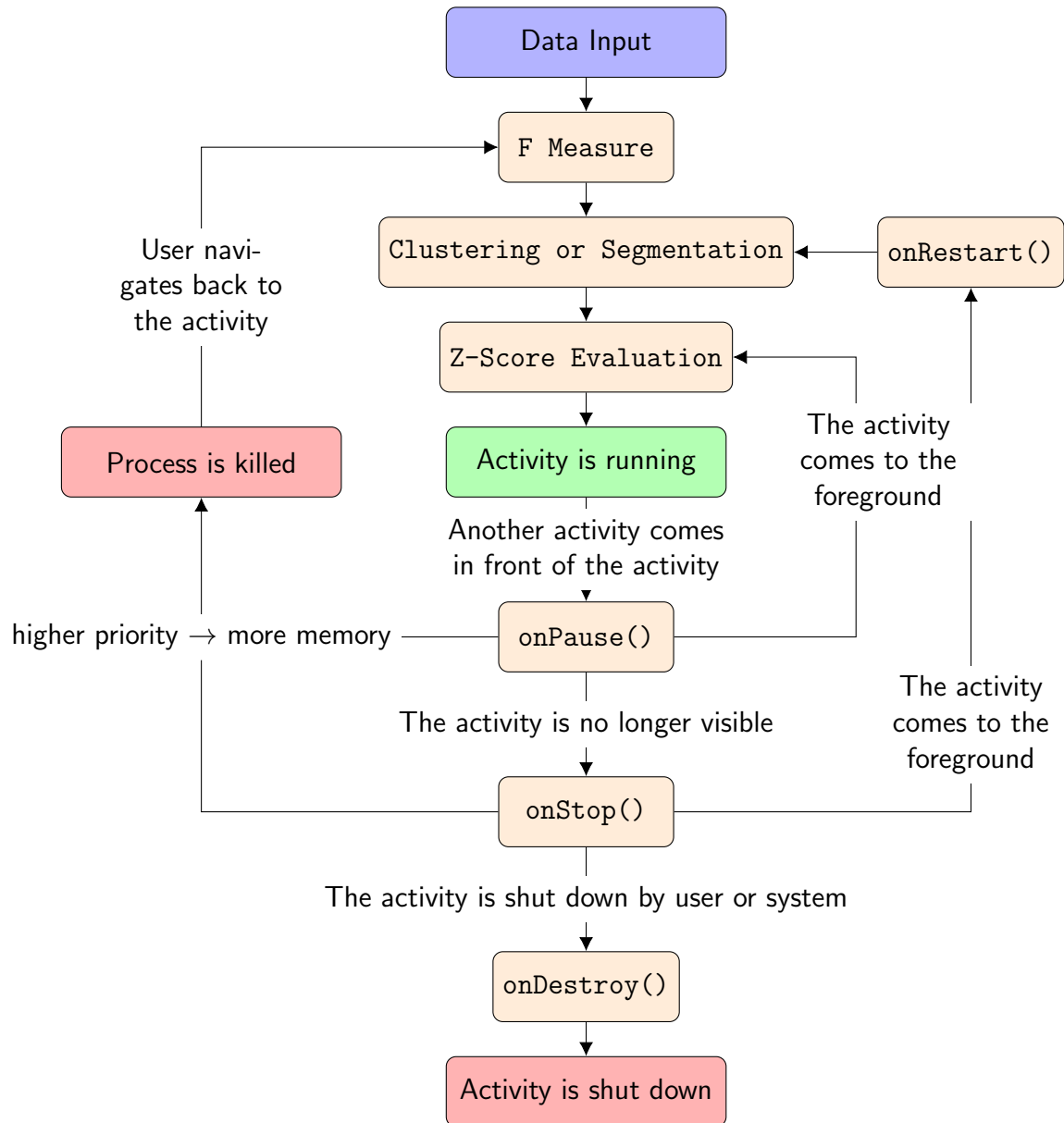
# **Building Information Systems for Prediction**

## **1 Introduction**

In today's world we have many algorithms and multiple datasets along with sufficient data available for testing and training the algorithms for classifying whether or not a person from the healthy population has the risk of contracting diabetes or not. The system should be able to firstly mine the data or handle the incoming data and filter or process it according to the parameters described by the required specifications.

## **2 Feature Selection**

### 3 Models



## 4 Conclusion

## Part VIII

# Conclusion



## Part IX

# Bibliography

### References

- [1] Diagnosis and Classification of Diabetes Mellitus. (2010). Retrieved November 28, 2019, from [https://care.diabetesjournals.org/content/33/Supplement\\_1/S62](https://care.diabetesjournals.org/content/33/Supplement_1/S62).
- [2] Home. (n.d.). Retrieved from <https://www.idf.org/e-library/epidemiology-research/diabetes-atlas/134-idf-diabetes-atlas-8th-edition.html>.
- [3] Wei M, Gibbons LW, Mitchell TL *et al.* (1999) The Association between cardiorespiratory fitness and impaired fasting glucose and type 2 diabetes mellitus in men. *Ann Intern Med* **130**, 427-34.
- [4] Jr., W. C. S. (2017, January 26). Definition of Diabetes mellitus. Retrieved November 17, 2019, from <https://www.rxlist.com/script/main/art.asp?articlekey=2974>.
- [5] Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., & Tang, H. (2018, November 6). Predicting Diabetes Mellitus With Machine Learning Techniques. Retrieved November 17, 2019, from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6232260/>.
- [6] Hinton, Geoffrey; Sejnowski, Terrence (1999). Unsupervised Learning: Foundations of Neural Computation. MIT Press. ISBN 978-0262581684.