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

I Acknowledgements **3**

II Background **4**

III Review of Literature **5**

IV Machine Learning Algorithms **6**

1 Linear Regression **6**
1.1 Introduction 6

2 Logistic Regression **7**

3 K-Nearest Neighbours **8**
3.1 Finding Optimum Number of Clusters 8

4 Decision Tree **9**
4.1 Introduction 9
4.2 Overfitting in Trees 9

5 Random Forest **11**

6 Gradient Boosting **12**

7 Support Vector Machine **13**
7.1 Linear Support Vector Machines 13

8 Perceptron **14**

9 Multilayered Perceptron **15**

V Real World Application **16**

VI Building Information Systems for Prediction **17**

1 Introduction **17**

2 Feature Selection **17**

3 Models **18**

4 Conclusion **19**

VII Conclusion **20**

VIII Bibliography **21**

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 acheive 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 throught the span of the project and helped me acheive 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
Background

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 III
Review of Literature

Part IV
Machine Learning Algorithms

1 Linear Regression

1.1 Introduction

Linear regression is a forecasting technique that can be use 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 $X = \{x_1 + x_2 + x_3 + ... + x_n\}$ 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 $Y = \{y_1 + y_2 + y_3 + ... + y_n\}$.

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 β_0 and β_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 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[1] 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[2] given the output class:

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

[1] 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. Some sources include the requirement that the curve may not cross the line infinitely often, but this is unusual for modern authors.

[2] 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++ (Karteeka 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.

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 disjunction of implications (i.e., if–then statements), and the implications are Horn clauses that are useful for logic programming. A Horn clause is a disjunction 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.

4.2 Overfitting in Trees

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 / TP: Where the actual class is yes, and the value of the predicted class is also yes.
- False Positive / FP: Actual class is no, and predicted class is yes
- True Negative / TN: The value of the actual class is no, and the value of the predicted class is no
- False Negative / FN: When the actual class value is yes, but predicted class is no
- Accuracy: (correctly predicted observation)/(total observation) = (TP + TN)/(TP + TN + FP + FN)
- Precision: (correctly predicted Positive)/(total predicted Positive) = TP/TP + FP
- Recall: (correctly predicted Positive)/(total correct Positive observation) = 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.

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

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

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

8 Perceptron

The perceptron is the foundation of neural networks.

9 Multilayered Perceptron

Part V
Real World Application

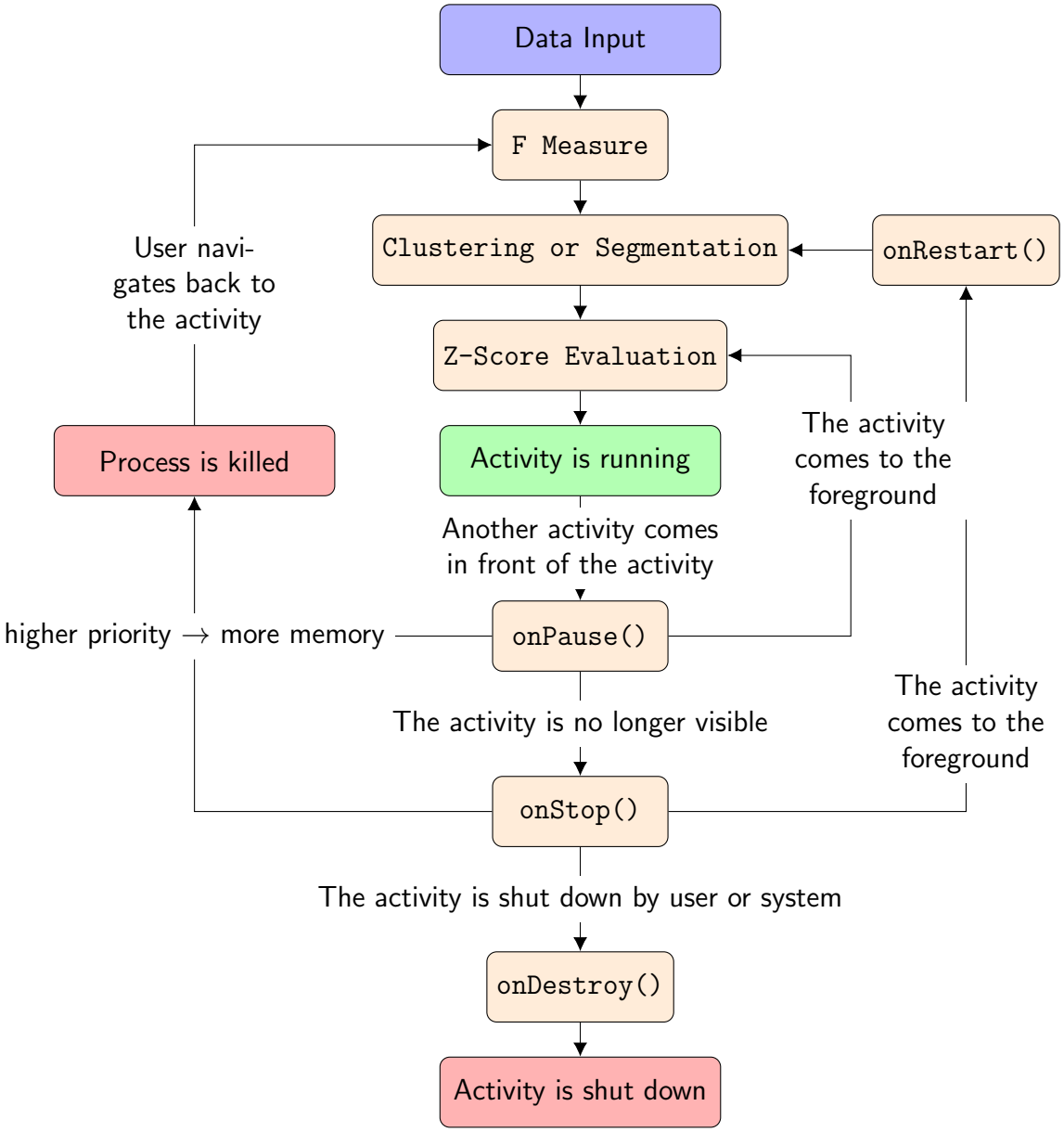
Part VI
Building Information Systems for Prediction

1 Introduction

In today's world we have many algorithms and multiple datasets along with sufficient data available for testing and training the algorithms.

2 Feature Selection

3 Models


Data Input
F Measure
Clustering or Segmentation
onRestart()
User navi-
gates back to
the activity
Z-Score Evaluation
Process is killed
Activity is running
The activity
comes to the
foreground
Another activity comes
in front of the activity
higher priority → more memory
onPause()
The activity is no longer visible
The activity
comes to the
foreground
onStop()
The activity is shut down by user or system
onDestroy()
Activity is shut down

4 Conclusion

Part VII
Conclusion

Part VIII
Bibliography

References

[1] 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.

[2] 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.

[3] 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/.