Texas A&M University - Commerce

Department of Computer Science

Performance Analysis of LR and SVM - Supervised Machine Learning Algorithms for Diabetes Prediction

Vyshnavi Sanikommu

Supervisor: Derek Harter, Ph.D.

A report submitted in partial fulfilment of the requirements of
Texas A&M University - Commerce for the degree of
Master of Science in *Computer Science*

February 3, 2024

Declaration

I, Vyshnavi Sanikommu, of the Department of Computer Science, Texas A&M University - Commerce, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of TAMUC and public with interest in teaching, learning and research.

Vyshnavi Sanikommu
February 3, 2024

Abstract

Diabetes is one of the most lethal diseases in the world. According to International Diabetes Foundation 537 million people are living with diabetes across the world. By 2045, this would be 783 million. Diabetes is a disease caused due to an increase in blood glucose level. High blood glucose produces the symptoms of frequent urination, increased hunger, and thirst. Diabetes is a leading cause of blindness, kidney failure, amputations, heart failure and stroke. When we eat, our body converts the food into sugars, or glucose. The pancreas releases insulin, a crucial hormone that unlocks cells to facilitate glucose entry. This mechanism enables our cells to utilize glucose as an energy source, supporting essential bodily functions. Machine learning is an emerging scientific field in data science dealing with the ways in which machines learn from experience. The aim of this project is to develop a system which can perform early prediction of diabetes for a patient with a higher accuracy by combining the results of supervised machine learning algorithms like logistic regression and support vector machine. Machine learning techniques provide better results for prediction by constructing models from datasets collected from patients. The accuracy of the model using each of the algorithms is calculated by considering the meta parameters to achieve the best performance. Then the one with a good accuracy is taken as the model for predicting diabetes.

Keywords: Diabetes, Machine Learning, Logistic Regression, Support Vector Machine, Accuracy

Acknowledgements

An acknowledgements section is optional. You may like to acknowledge the support and help of your supervisor(s), friends, or any other person(s), department(s), institute(s), etc. If you have been provided specific facility from department/school acknowledged so.

Contents

List of Figures

List of Tables

List of Abbreviations

SMPCS	School of Mathematical, Physical and Computational Sciences

Chapter 1

Introduction

Diabetes is a fast growing disease among the people even in youngsters nowadays. It is necessary to understand how it develops in our body. Firstly, we need to understand how a body works without diabetes. Sugar comes from the food that we eat, specially carbohydrates. When we eat this food, the body breaks them down to Sugars or Glucose. This glucose moves around the body in the bloodstream. This glucose was needed by body parts like brain, pancreas to function. The remainder of glucose is taken to the cells of our body and liver which is stored as energy for later use. In order to use glucose for our body, insulin is required which is a hormone generated by pancreas. Insulin is a key to a closed door which helps glucose moves from blood stream. If pancreas is not able to produce enough insulin or if our body cannot use insulin it produces then glucose levels increases in bloodstream which leads to diabetes.

According to World Health Organization, diabetes is major cause of death worldwide. Around 422 million people worldwide have diabetes. Indeed, it caused deaths of 2 million people in 2019.

Types of Diabetes

Type 1 diabetes appear most often during childhood and characterized by the partial functioning of pancreas. The cells fail to produce sufficient amounts of Insulin. Initially, we do not see any symptoms as the pancreas remains partially functional. There is no proven study and known methods for prevention.

Type 2 diabetes affects how the body uses sugars for energy. The cells produce low quantity of insulin or the body stops using insulin which can lead to high levels of sugar in bloodstream. High levels of glucose in the bloodstream and urine referred as Diabetes Mellitus. It is most common type of diabetes found in many people. It is caused by genetic factors and the lifestyle. It affects older adults and more obese or overweight people.

Early prediction of the disease can be controlled and save the life. To accomplish this, this work explores prediction of diabetes by taking various attributes related to diabetes disease. For this purpose, we use the Diabetes Dataset Johndasilva (2018) which has 2000 instances with 9 attributes - 'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome' for creating classification models using logistic regression

and support vector machine algorithms.

1.1 Background

This study is undertaken to address the crucial need for accurate and reliable methods in predicting diabetes using logistic regression and support vector machine algorithms on patient data. The motivation behind this study stems from the growing significance of leveraging machine learning algorithms to assist medical professionals in diagnosing and managing diabetes. Predictive models based on patient data offer the potential to identify individuals at risk or in the early stages of diabetes, enabling proactive and personalized healthcare strategies. Algorithms - logistic regression and support vector machine are chosen due to their widespread use in medical data analysis and their capability to handle classification tasks. The study outcomes have practical implications for healthcare practitioners and researchers, offering insights into algorithm selection for accurate and interpret diabetes prediction.

1.2 Research question

This section provides a detailed exploration of the research problem. The main question is about understanding how accurate logistic regression and support vector machine are in predicting diabetes from patient's data. The focus is on figuring out what factors cause differences in accuracy, especially looking at the impact of hyper parameters.

1.3 Aims and objectives

The primary aim of this study is to assess and compare the predictive accuracy of logistic regression and support vector machine algorithms based on diabetic patient's data to decide if a patient is diabetic or not. This study aims to provide valuable insights into the strengths and limitations of these supervised machine learning approaches, contributing to the enhancement of diabetes prediction methodologies.

Objectives: The main objectives of this study is to do data processing, implement logistic regression and support vector machine models, hyper parameter tuning to achieve highest possible predictive accuracy. At the end, we will evaluate the performance of both classification models based on accuracy and the best model is decided.

1.4 Solution approach

The study follows a systematic approach encompassing model implementation, data preprocessing, splitting of data, hyper parameter tuning and performance analysis. Thorough exploration of various meta parameter search strategies contribute to achieving the aims. I will make sure to provide a clear documentation to make sure results are reliable and can be easily reproducible.

1.4.1 Dataset Description

This Diabetes Dataset Johndasilva (2018) has 2000 instances with 9 attributes - 'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'. The 'Outcome' attribute indicates positive or negative for diabetes.

1.4.2 Data Preprocessing

Data preprocessing is most important process. Mostly healthcare related data contains missing values that causes effectiveness of data. This process is essential for accurate result and successful predication using machine learning techniques.

Missing values removal

This process is meant to identify instances with zero value and eliminate all such instances. Through eliminating irrelevant instances we make feature subset and this process is called feature subset selection which reduces the dimensionality of data.

Splitting of data

After cleaning the data, the data is normalized in training and testing the models. After split, we train algorithm with training dataset and keep testing dataset aside. This testing dataset is used to test the trained model.

1.4.3 Model Implementation

After data is ready, we apply machine learning techniques. Implement logistic regression and support vector machine models to predict diabetes on the dataset.

1.4.4 Hyper Parameter Tuning and Performance Analysis

To determine the best performance model, we should consider the factors like hyper parameters, meta parameter search strategies for logistic regression and support vector machine to achieve more predictive accuracy.

1.5 Summary of contributions and achievements

Describe clearly what you have done/created/achieved and what the major results and their implications are.

Chapter 2

Literature Review

The examination of relevant studies yields findings from various healthcare datasets, where researchers conducted analyses and predictions employing a range of methods and techniques. Numerous prediction models have been devised and applied by various researchers, utilizing different forms of data mining techniques, machine learning algorithms, or a combination of these methodologies.

2.1 Review of state-of-the-art

Mujumdar and Vaidehi (2019) aims to create a system using machine learning algorithm and deep learning techniques to provide accurate results and reduce human efforts. The diabetes dataset contains 800 instances with 10 attributes. This study implemented various machine learning algorithms include Support Vector Classifier, Random Forest Classifier, Decision Tree Classifier, Extra Tree Classifier, Ada Boost algorithm, Perceptron, Linear Discriminant Analysis algorithm, Logistic Regression, K-Nearest Neighbour, Gaussian Naïve Bayes,Bagging algorithm, Gradient Boost Classifier. The study incorporates the concept of pipelining and compares the diabetes dataset with the Pima dataset. Performance analysis includes metrics like classification accuracy, confusion matrix, f1-score, precision, and recall. The findings reveal that Logistic Regression achieves the highest accuracy of 96%, indicating an improvement in accuracy for the diabetes dataset compared to the Pima diabetes dataset. The study concludes that implementing a pipeline model enhances the accuracy of the classification performance, with the Ada Booster classifier identified as the best model, achieving an accuracy of 98.8%.

Soni and Varma (2020) aims to design and implement Diabetes prediction using machine learning methods and performance analysis of that methods for early prediction and to cure diabetes and save humans life. The diabetes dataset is gathered from UCI repository which is named as Pima Indian Diabetes dataset. The dataset have many attributes of 768 patients. The proposed methodology involves the utilization of various classification and ensemble learning methods, including SVM, Logistic Regression, KNN, Rndom Forest, Decision Tree, Gradient Boosting classifiers are used. The findings indicate a 77% accuracy achieved through an 80:20 split. The study concludes that Random Forest classifier exhibits highest accuracy when compared to other machine learning methods.

Swapna et al. (2018) aims to develop a methodology for classification of diabetic and normal Heart rate variability (HRV) signals using advanced deep learning architectures, specifically

employing Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and combinations. The extracted features are then passed into a Support Vector Machine (SVM) for accurate classification. The study demonstrates performance improvements in CNN and CNN-LSTM architectures compared to earlier work, achieving a high accuracy of 95.7%.

2.2 Critique of the review

The literature review provides insights related to diabetes prediction using machine learning and deep learning techniques. Each study aims to contribute to the development of reliable and accurate models for diabetes diagnosis, with a focus on enhancing classification performance. Mujumdar and Vaidehi (2019) focus on creating an efficient system, achieving a 96% accuracy with Logistic Regression and identifying Ada Booster as the best model. Soni and Varma (2020) design a predictive system, achieving 77% accuracy with Random Forest being the most accurate. Swapna et al. (2018) explore advanced deep learning architectures, obtaining a high accuracy of 95.7% with CNN-LSTM and SVM. The key findings include the effectiveness of pipeling models, ensemble methods and the importance of accurate diabetes prediction for early intervention and patient care. Future research directions involve anomaly prediction and the utilization of larger datasets.

2.3 Summary

Chapter 3

Methodology

We mentioned in Chapter 1 that a project report's structure could follow a particular paradigm. Hence, the organization of a report (effectively the Table of Content of a report) can vary depending on the type of project you are doing. Check which of the given examples suit your project. Alternatively, follow your supervisor's advice.

3.1 Examples of the sections of a methodology chapter

A general report structure is summarised (suggested) in Table 3.1. Table 3.1 describes that, in general, a typical report structure has three main parts: (1) front matter, (2) main text, and (3) end matter. The structure of the front matter and end matter will remain the same for all the undergraduate final year project report. However, the main text varies as per the project's needs.

3.1.1 Example of a software/Web development main text structure

Notice that the "methodology" Chapter of Software/Web development in Table 3.2 takes a standard software engineering paradigm (approach). Alternatively, these suggested sections can be the chapters of their own. Also, notice that "Chapter 5" in Table 3.2 is "Testing and Validation" which is different from the general report template mentioned in Table 3.1. Check with your supervisor if in doubt.

3.1.2 Example of an algorithm analysis main text structure

Some project might involve the implementation of a state-of-the-art algorithm and its performance analysis and comparison with other algorithms. In that case, the suggestion in Table 3.3 may suit you the best.

3.1.3 Example of an application type main text structure

If you are applying some algorithms/tools/technologies on some problems/datasets/etc., you may use the methodology section prescribed in Table 3.4.

Table 3.1: Undergraduate report template structure

Frontmatter		Title Page Abstract Acknowledgements Table of Contents List of Figures List of Tables List of Abbreviations
Main text	Chapter 1	Introduction
	Chapter 2	Literature Review
	Chapter 3	Methodology
	Chapter 4	Results
	Chapter 5	Discussion and Analysis
	Chapter 6	Conclusions and Future Work
	Chapter 7	Refection
End matter		References Appendices (Optional) Index (Optional)

Table 3.2: Example of a software engineering-type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	
		Requirements specifications
		Analysis
		Design
		Implementations
Chapter 4	Testing and Validation	
Chapter 5	Results and Discussion	
Chapter 6	Conclusions and Future Work	
Chapter 7	Reflection	

3.1.4 Example of a science lab-type main text structure

If you are doing a science lab experiment type of project, you may use the methodology section suggested in Table 3.5. In this kind of project, you may refer to the "Methodology" section as "Materials and Methods."

Table 3.3: Example of an algorithm analysis type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	
		Algorithms descriptions
		Implementations
		Experiments design
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

Table 3.4: Example of an application type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	
		Problems (tasks) descriptions
		Algorithms/tools/technologies/etc. descriptions
		Implementations
		Experiments design and setup
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

Table 3.5: Example of a science lab experiment-type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Materials and Methods	
		Problems (tasks) description
		Materials
		Procedures
		Implementations
		Experiment set-up
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

3.2 Example of an Equation in LaTeX

Eq. 3.1 [note that this is an example of an equation's in-text citation] is an example of an equation in LaTeX. In Eq. (3.1), s is the mean of elements $x_i \in \mathbf{x}$:

$$s = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{3.1}$$

Have you noticed that all the variables of the equation are defined using the **in-text** maths command $.$, and Eq. (3.1) is treated as a part of the sentence with proper punctuation? Always treat an equation or expression as a part of the sentence.

3.3 Example of a Figure in LaTeX

Figure 3.1 is an example of a figure in LaTeX. For more details, check the link:
wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions.
Keep your artwork (graphics, figures, illustrations) clean and readable. At least 300dpi is a good resolution of a PNG format artwork. However, an SVG format artwork saved as a PDF will produce the best quality graphics. There are numerous tools out there that can produce vector graphics and let you save that as an SVG file and/or as a PDF file. One example of such a tool is the "Flow algorithm software". Here is the link for that: flowgorithm.org.

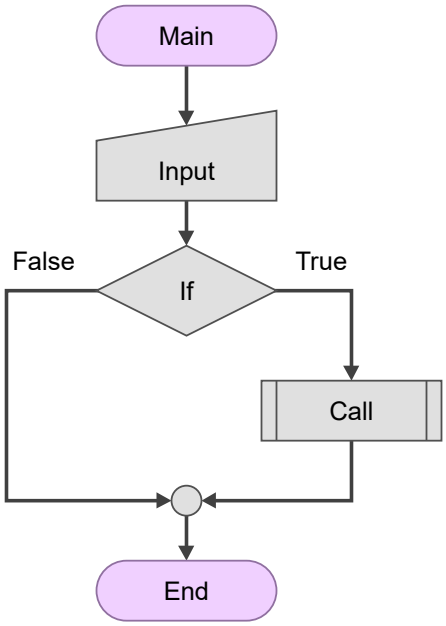

Figure 3.1: Example figure in LaTeX.

3.4 Example of an algorithm in LaTeX

Algorithm 1 is a good example of an algorithm in LaTeX.

Algorithm 1 Example caption: sum of all even numbers

Input: $\mathbf{x} = x_1, x_2, \ldots, x_N$
Output: $EvenSum$ (Sum of even numbers in $\mathbf{x}$)

1: **function** EvenSummation($\mathbf{x}$)
2: $EvenSum \leftarrow 0$
3: $N \leftarrow length(\mathbf{x})$
4: **for** $i \leftarrow 1$ to N **do**
5: **if** $x_i \mod 2 == 0$ **then** ▷ check if a number is even?
6: $EvenSum \leftarrow EvenSum + x_i$
7: **end if**
8: **end for**
9: **return** $EvenSum$
10: **end function**

3.5 Example of code snippet in LaTeX

Code Listing 3.1 is a good example of including a code snippet in a report. While using code snippets, take care of the following:

- do not paste your entire code (implementation) or everything you have coded. Add code snippets only.
- The algorithm shown in Algorithm 1 is usually preferred over code snippets in a technical/-scientific report.
- Make sure the entire code snippet or algorithm stays on a single page and does not overflow to another page(s).

Here are three examples of code snippets for three different languages (Python, Java, and CPP) illustrated in Listings 3.1, 3.2, and 3.3 respectively.

```python
import numpy as np

x  = [0, 1, 2, 3, 4, 5] # assign values to an array
evenSum = evenSummation(x) # call a function

def evenSummation(x):
    evenSum = 0
    n = len(x)
    for i in range(n):
        if np.mod(x[i],2) == 0: # check if a number is even?
            evenSum = evenSum + x[i]
    return evenSum
```

Listing 3.1: Code snippet in LaTeX and this is a Python code example

Here we used the "\clearpage" command and forced-out the second listing example onto the next page.

```java
public class EvenSum{
    public static int evenSummation(int[] x){
        int evenSum = 0;
        int n = x.length;
        for(int i = 0; i < n; i++){
            if(x[i]%2 == 0){ // check if a number is even?
                evenSum = evenSum + x[i];
            }
        }
        return evenSum;
    }
    public static void main(String[] args){
        int[] x  = {0, 1, 2, 3, 4, 5}; // assign values to an array
        int evenSum = evenSummation(x);
        System.out.println(evenSum);
    }
}
```

Listing 3.2: Code snippet in LaTeX and this is a Java code example

```cpp
int evenSummation(int x[]){
    int evenSum = 0;
    int n = sizeof(x);
    for(int i = 0; i < n; i++){
        if(x[i]%2 == 0){ // check if a number is even?
            evenSum = evenSum + x[i];
      }
    }
    return evenSum;
}

int main(){
    int x[]  = {0, 1, 2, 3, 4, 5}; // assign values to an array
    int evenSum = evenSummation(x);
    cout<<evenSum;
    return 0;
}
```

Listing 3.3: Code snippet in LaTeX and this is a C/C++ code example

3.6 Example of in-text citation style

3.6.1 Example of the equations and illustrations placement and reference in the text

Make sure whenever you refer to the equations, tables, figures, algorithms, and listings for the first time, they also appear (placed) somewhere on the same page or in the following page(s). Always make sure to refer to the equations, tables and figures used in the report. Do not leave them without an **in-text citation**. You can refer to equations, tables and figures more them once.

3.6.2 Example of the equations and illustrations style

Write **Eq.** with an uppercase "Eq" for an equation before using an equation number with (\eqref{.}). Use "Table" to refer to a table, "Figure" to refer to a figure, "Algorithm" to

refer to an algorithm and "Listing" to refer to listings (code snippets). Note that, we do not use the articles "a," "an," and "the" before the words Eq., Figure, Table, and Listing, but you may use an article for referring the words figure, table, etc. in general.

For example, the sentence "A report structure is shown in **the** Table 3.1" should be written as "A report structure is shown **in** Table 3.1."

3.7 Summary

Write a summary of this chapter.

Note: In the case of **software engineering** project a Chapter "**Testing and Validation**" should precede the "Results" chapter. See Section 3.1.1 for report organization of such project.

Chapter 4

Results

The results chapter tells a reader about your findings based on the methodology you have used to solve the investigated problem. For example:

- If your project aims to develop a software/web application, the results may be the developed software/system/performance of the system, etc., obtained using a relevant methodological approach in software engineering.
- If your project aims to implement an algorithm for its analysis, the results may be the performance of the algorithm obtained using a relevant experiment design.
- If your project aims to solve some problems/research questions over a collected dataset, the results may be the findings obtained using the applied tools/algorithms/etc.

Arrange your results and findings in a logical sequence.

4.1 A section

...

4.2 Example of a Table in LaTeX

Table 4.1 is an example of a table created using the package LaTeX"booktabs." do check the link: wikibooks.org/wiki/LaTeX/Tables for more details. A table should be clean and readable. Unnecessary horizontal lines and vertical lines in tables make them unreadable and messy. The example in Table 4.1 uses a minimum number of liens (only necessary ones). Make sure that the top rule and bottom rule (top and bottom horizontal lines) of a table are present.

Table 4.1: Example of a table in LaTeX

Bike		
Type	Color	Price (£)
Electric	black	700
Hybrid	blue	500
Road	blue	300
Mountain	red	300
Folding	black	500

4.3 Example of captions style

- The **caption of a Figure (artwork) goes below** the artwork (Figure/Graphics/illustration). See example artwork in Figure 3.1.
- The **caption of a Table goes above** the table. See the example in Table 4.1.
- The **caption of an Algorithm goes above** the algorithm. See the example in Algorithm 1.
- The **caption of a Listing goes below** the Listing (Code snippet). See example listing in Listing 3.1.

4.4 Summary

Write a summary of this chapter.

Chapter 5

Discussion and Analysis

Depending on the type of project you are doing, this chapter can be merged with "Results" Chapter as " Results and Discussion" as suggested by your supervisor.

In the case of software development and the standalone applications, describe the significance of the obtained results/performance of the system.

5.1 A section

Discussion and analysis chapter evaluates and analyses the results. It interprets the obtained results.

5.2 Significance of the findings

In this chapter, you should also try to discuss the significance of the results and key findings, in order to enhance the reader's understanding of the investigated problem

5.3 Limitations

Discuss the key limitations and potential implications or improvements of the findings.

5.4 Summary

Write a summary of this chapter.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Typically a conclusions chapter first summarizes the investigated problem and its aims and objectives. It summaries the critical/significant/major findings/results about the aims and objectives that have been obtained by applying the key methods/implementations/experiment set-ups. A conclusions chapter draws a picture/outline of your project's central and the most signification contributions and achievements.

A good conclusions summary could be approximately 300–500 words long, but this is just a recommendation.

A conclusions chapter followed by an abstract is the last things you write in your project report.

6.2 Future work

This section should refer to Chapter 4 where the author has reflected their criticality about their own solution. The future work is then sensibly proposed in this section.

Guidance on writing future work: While working on a project, you gain experience and learn the potential of your project and its future works. Discuss the future work of the project in technical terms. This has to be based on what has not been yet achieved in comparison to what you had initially planned and what you have learned from the project. Describe to a reader what future work(s) can be started from the things you have completed. This includes identifying what has not been achieved and what could be achieved.

A good future work summary could be approximately 300–500 words long, but this is just a recommendation.

Chapter 7

Reflection

Write a short paragraph on the substantial learning experience. This can include your decision-making approach in problem-solving.

Some hints: You obviously learned how to use different programming languages, write reports in LaTeXand use other technical tools. In this section, we are more interested in what you thought about the experience. Take some time to think and reflect on your individual project as an experience, rather than just a list of technical skills and knowledge. You may describe things you have learned from the research approach and strategy, the process of identifying and solving a problem, the process research inquiry, and the understanding of the impact of the project on your learning experience and future work.

Also think in terms of:

- what knowledge and skills you have developed
- what challenges you faced, but was not able to overcome
- what you could do this project differently if the same or similar problem would come
- rationalize the divisions from your initial planed aims and objectives.

A good reflective summary could be approximately 300–500 words long, but this is just a recommendation.

Note: The next chapter is "**References**," which will be automatically generated if you are using BibTeX referencing method. This template uses BibTeX referencing. Also, note that there is difference between "References" and "Bibliography." The list of "References" strictly only contain the list of articles, paper, and content you have cited (i.e., refereed) in the report. Whereas Bibliography is a list that contains the list of articles, paper, and content you have cited in the report plus the list of articles, paper, and content you have read in order to gain knowledge from. We recommend to use only the list of "References."

References

Johndasilva (2018), 'Diabetes dataset'. (accessed January 25, 2024).
URL:

Mujumdar, A. and Vaidehi, V. (2019), 'Diabetes prediction using machine learning algorithms', *Procedia Computer Science* **165**, 292–299.

Soni, M. and Varma, S. (2020), 'Diabetes prediction using machine learning techniques', *International Journal of Engineering Research & Technology (Ijert) Volume* **9**.

Swapna, G., Vinayakumar, R. and Soman, K. (2018), 'Diabetes detection using deep learning algorithms', *ICT express* **4**(4), 243–246.

Appendix A

An Appendix Chapter (Optional)

Some lengthy tables, codes, raw data, length proofs, etc. which are **very important but not essential part** of the project report goes into an Appendix. An appendix is something a reader would consult if he/she needs extra information and a more comprehensive understating of the report. Also, note that you should use one appendix for one idea.

An appendix is optional. If you feel you do not need to include an appendix in your report, avoid including it. Sometime including irrelevant and unnecessary materials in the Appendices may unreasonably increase the total number of pages in your report and distract the reader.

Appendix B

An Appendix Chapter (Optional)

...