Faculty of Engineering, Environment and Computing
6001CEM Individual Project

DIABETES PREDICTION WITH MACHINE LEARNING ALGORITHMS

Author: Sarkis Chichkoyan

Student ID: 9827770

Supervisors: Carey Pridgeon/Amanda Brooks

Date: 26/04/22

Coventry University

Table of contents

Faculty of Engineering, Environment and Computing
6001CEM Individual Project

6001CEM Declaration of originality

I Declare that This project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

Statement of copyright

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to Coventry University. Support, including funding, is available to commercialize products and services developed by staff and students. Any revenue that is generated is split with the inventor/s of the product or service. For further information, please see www.coventry.ac.uk/ipr or contact ipr@coventry.ac.uk.

Statement of ethical engagement

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (https://ethics.coventry.ac.uk/) and that the application number is listed below (Note: Projects without an ethical application number will be rejected for marking).

Signed:  Date: 26/04/2022

First Name:	Sarkis
Last Name:	Chichkoyan
Student ID number	9827770
Ethics Application Number	P130492
Supervisors	Carey Pridgeon/Amanda Brooks

Abstract

Diabetes is a disease that causes a person's blood sugar level to become too high. There are two types of diabetes, type 1 and type 2. Type 1 is when the body's immune system attacks and destroys the cells that produce insulin. Type 2 is when the body does not produce enough insulin, or the body's cells do not react to insulin. Type 2 is the more common one, in fact, this lifelong condition concerns around 90% of adults that have diabetes in UK. It is an important disease that touches more that 463 million people on earth on 2019, so it is a major problem to deal with.

This project is oriented on an analysis of a dataset which comes from the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK), which is based in the United States. The objective is to predict if the patients of the dataset have diabetes or not, with the help of diagnostic measurements. All the patients are females, are at least 21 years old and of Pima Indian heritage. In this paper we are going to use those Machine learning algorithms: Logistic Regression, SVC, kNN and Random Forest Classifier; to predict diabetes and compare the different accuracies and results that we get.

Acknowledgements

Thank you to my first supervisor Carey Pridgeon for his guidance to help me find the idea for the project. Thank you to Amanda Brooks, my second supervisor who helped me a lot on a small period of time to organise and finish this dissertation.

GitHub repository link:

https://github.coventry.ac.uk/chichkos/6001CEM_ML_Diabetes_PimaIndians

Introduction

Background and motivation

Diabetes is a disease that touches 463 million people in the world in 2019, it is an important illness that has two types. Type 1 usually comes at a young age and it's when the body stops producing insulin. Type 2 is when the body is not producing enough insulin to maintain a stable blood sugar level.

Machine Learning is more and more used in different fields, one of them is healthcare and it starts to have huge impacts on analysis and on prediction for diseases.

The motivation for this project is to be able to do a predictive analysis on diabetes. Another fact is that there are people in my family environment that have diabetes and it was a motivation for me to know more about the disease, understand it more and try to bring results that could help.

In this analysis we will be concentrated on one dataset which contain only females that are at least 21 years old and from Pima Indian heritage.

Project aim and objectives

The aim of this project is to be able to predict diabetes with the highest accuracy possible by using the appropriate machine learning algorithms and improving the models.

The first objective is to implement the different algorithms that has been chosen for the predictive analysis, then the second one is to separately evaluate the accuracies of each one of them. After that, the objective is to get a higher accuracy by applying methods that could enhance the result and make it more accurate. Finally, we compare the different algorithm results and give the best algorithm according to the analysis.

Structure

1 – Literature review:

The literature review is all the background research on the works previously done on the topics that will be treated, in this case diabetes, machine learning algorithms and techniques and also previous works done on the same question as the thesis.

2 – Research Methodology:

The research methodology part is all about finding the different ways of doing the work that we need to do and explain how they work and how they will be relevant for the thesis

3 – Implementation:

This part goes through the process of implementing all what was researched and studied in the research methodology part and how it fits into the project.

4 – Result Analysis:

It displays the different results and figures for every algorithm and explains what the best algorithm is, with the analysis of the different accuracies is that we get.

5 – Project Management:

It is a description of the project management methodology and time management. There is also a part that explains the different risk that we can encounter. Then, there is a part where it is explained how communication and feedback was done on the project, followed by all the legal, ethical and social parts of the project. In the end there is a reflexion on how the project was managed and how it could have been improved

6 – Conclusion:

Summarize the project and explain how it can be important and relevant in the real world, also explore the limitations of the project and the future work that could be done on the same topic. Finally, a reflexion on the whole thesis, how it impacted me and what changed after doing it.

Literature Review

This section of the report is an investigation on literature available for the different topics that are related to it. It will go through Diabetes, the National Institute of Diabetes and Digestive and Kidney Diseases Machine learning, Algorithms, Feature scaling, Confusion matrices and Cross validation. This section will also be a place to explain how Machine learning algorithms are related to Diabetes.

Diabetes

Diabetes affects approximately 5 million people in the UK, 90% of them are affected by Type 2, the rest of them are Type 1 or unknown. It is predicted that around 2030 there will be 5.5 million people affected by diabetes (Diabetes UK, 2019). Worldwide it is estimated that 415 million are living with diabetes and it is predicted that there will be 642 million by 2040 (Diabetes.co.uk, 2022).

The first mention of diabetes was in 1552 B.C., in fact an Egyptian physician named Hesy-Ra has documented a disease with frequent urination and that also caused emaciation. Centuries later, the technique used to detect the disease was to taste the urine if it was sweet then the person had diabetes. It was not until the 1800's that scientists found methods to be able to detect the presence of sugar in urine. In 1889 the first way to treat diabetes that was effective was found, the use of insulin. It was Oskar Minkowski and Joseph von Mering, researchers, that proved that removing a dog's pancreas would lead to diabetes (Krisha McCoy, 2009).

The pancreas is the part of the body that produces insulin. Insulin is very important for our bodies. And whenever the pancreas is not producing insulin or enough insulin, there is no insulin going into the blood and it causes the glucose levels to go too high and induce diabetes (Mayo Clinic). There are two main types of diabetes, Type 1 which can only be managed with medication and Type 2 which can be managed with both medication and diet.

Diabetes Type 1 is often diagnosed during childhood or adolescence (4-7 years old, 10-14 years old) even if sometimes it can develop in adults (Mayo Clinic). The problem with this type of diabetes is that the exact cause is not known. It's the body's own immune system that destroys the cells in the pancreas that produce the insulin for the blood (Mayo Clinic). The medication needed for Type 1 diabetes is insulin injections, it can be once or twice a day to do a long-acting effect, or it can be insulin taken with food or drink and it is more of a fast-acting process to reduce the amount of glucose in blood that is caused by eating or drinking (NHS, 2021).

Diabetes Type 2 is whenever the body is not producing enough insulin to low the level of glucose in the blood. The main difference with Type 1 diabetes is that Type 2 can be managed with healthy and active life, there is also medication like tablets or injections of insulin (Diabetes UK). To manage this type of diabetes it is advised to check the Glycaemic Index (GI) for the food that is going to be eaten, GI is the value that is used to know how much a specific food increase the blood sugar levels. The diet should be low on sugar, salt and fat. It is also very important to check the blood sugar levels to be able to know whenever it is more important to take medication or not. There are a few ways to check the blood sugar levels, finger-prick-tests, an electronic blood sugar monitor and an HbA1c which allows to measure the levels over the past three months. If it is too low, it is called hypoglycaemia also "hypo" and if it is too high it is called hyperglycaemia also "hyper" (Diabetes UK, 2019).

There are some things that will increase the chance of someone getting type 2 diabetes. First it is the age if you are white and more than 40 or even over 25 for the other ethnicities. Like it was said earlier eating is very important when we have diabetes but also when we don't have it because obesity can cause someone of having type 2 diabetes. It can also be due to a parent that already have diabetes. If you have a medical history, you can also have diabetes, for instance if you had heart attacks or high blood pressure it can increase the chances. There can be a high blood glucose that can develop whenever there is a pregnancy it is called gestational diabetes (NHS, 2022). This information is important because in the dataset there is a pregnancy parameter and since we have only women in the dataset it is relevant.

According to Diabetes.co.uk ethnicity has a big role in diabetes cases in fact it is two and a half times more likely that Indian women develop diabetes. This information is very important for this project because the dataset that is work with is composed of Pima Indian Heritage women. According to the International Diabetes Federation the top 5 countries with the highest amount of diabetes are:

- China (109 million)
- India (69 million)
- USA (29 million)
- Brazil (14 million)
- Russia (12 million)

They are multiple signs that show someone has diabetes, it can be a weight loss, blurred vision, to go more often to the toilet and feeling more and more tired (Diabetes UK). It is very important to diagnose diabetes early because it can lead to developing complications. UK expert committee says that people that are around 6.0-6.4 % HbA1c are considered high risk even if they don't have any symptoms, they are also called "prediabetes". Early identification can delay or even prevent the condition if it is not at a high-risk stage (Diabetes UK, 2021).

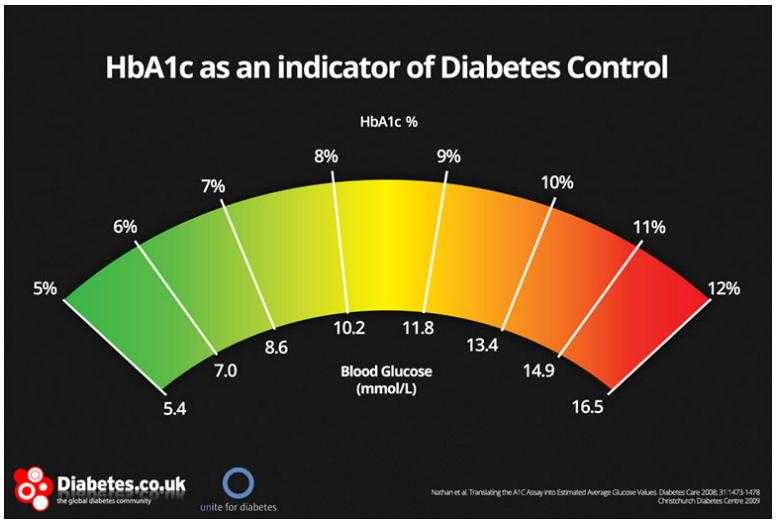

Figure 1 HbA1c indicator

Machine Learning

In 1950 Alan Turing proposed something named the Turing Test. This test is basically to be able to know if machines can think or not. The criteria to be able to know that was if the computer could convince that it was a human being (Wesley Chai, 2020). Now the term to define this is artificial intelligence (AI).

It is around 1957 that Frank Rosenblatt, at the Cornell Aeronautical Laboratory, created a software named perceptron, it was initially designed to be a machine. This software is considered to be the first machine learning algorithm, the objective is to give a valid output after analysing data. Unfortunately, this machine named Mark 1 perceptron couldn't give the good outputs and didn't meet the expectations, by the time it was very promising, but not working perfectly. Ten years later the nearest neighbour algorithm is created, it is the very beginning of basic pattern recognition (Cover and Hart, 1967). In this project you will find that k-nearest neighbour algorithm (kNN) is used, and it is a very similar algorithm to this one. The nearest neighbour algorithm was designed to find a solution for traveling salesperson's problem to find the best route it was used to map routes (Keith D. Foote, 2021). This model was a real improvement and one the first to provide a solution with an algorithm that works.

There are a lot of Machine Learning studies on diabetes. Machine learning algorithms are increasingly used in the medical environment. For almost all the studies that we are going to look they are using clinical datasets or medical records to make their work.
In those projects there are different methodologies to get accurate and relevant results when it comes to Machine Learning algorithms. If we look at the first steps there are always the same, extracting the needed data and then pre-process it to be sure that there are not duplicates or values that are not relevant to the work. Then the algorithms used are always different from a study to another. For example, there is a project which used Logistic Regression and SVM based algorithms to make diabetes prediction made in 2018 by Pramila M. Chawan (and two other collaborators). There is also the Henry Ford project (2017), the used algorithms were three different decision trees, Naïve Bayes, Random Forest, and Logistic Regression. In other works, we

can find analysis with Neural Network classifiers like Quan Zou's team project in 2018 on diabetes prediction. It shows us the different possibilities and the different ways to analyse same type of data. Finding the algorithm that fits the best to be able to compare the different works and make conclusions.

When Machine Learning is used to do prediction, it is called predictive modelling or predictive analysis. A predictive analysis can be a huge improvement and save a lot of time because it gives very precise results and saves also energy, if humans had to go through all the analysis, it would be too difficult. One of the drawbacks is that we are never sure that the result is going to be a 100% accurate because machine learning is all about the algorithm adapting to the data to give the most accurate results and it is not always the case. Another problem can be that not everyone is ready to adopt these methods, mostly because of their complexity and lack of knowledge about them.

As machine learning algorithms become more and more intelligent, they can be more precise and accurate when it comes to predict outcomes. There are multiple ways of doing predictive models with machine learning one of them is classification.

"Classification is the process of predicting the class of given data points" (Sidath Asiri, 2018). Classes will be equivalent to categories or labels.

There is a lot of classification algorithms available that's why we need to make relevant choices according to our research, the goal is to predict whether someone will have diabetes or not. Four different algorithms were chosen to do this work. First the k nearest neighbour classifier, then the logistic regression, also the support vector classifier and finally de random forest classifier.

K-Nearest Neighbour algorithm

The kNN is an improvement of the nearest neighbour algorithm, it stores the data available and classifies new data point based on the similarity that they have. “We find the set of K nearest neighbours in the training set to Xo and then classify Xo as the most frequent class among the K neighbours” (Trevor Hastie, Robert Tibshirani, 1996). Whenever new data comes in it is going to be suited to its category (javaTpoint/machine learning). The similarity is found using the proximity with the previous neighbours that it went through to make a classification or a prediction. The advantage of using this algorithm is that is it very simple fast and efficient but there is a disadvantage is that we must find manually the number of K’s that will match our expectation. There are examples of KNN algorithms used in real life problems, for instance they are used in the healthcare area to do predictions on the risk of heart attacks and prostate cancer by calculating gene expressions (IBM, 2022).

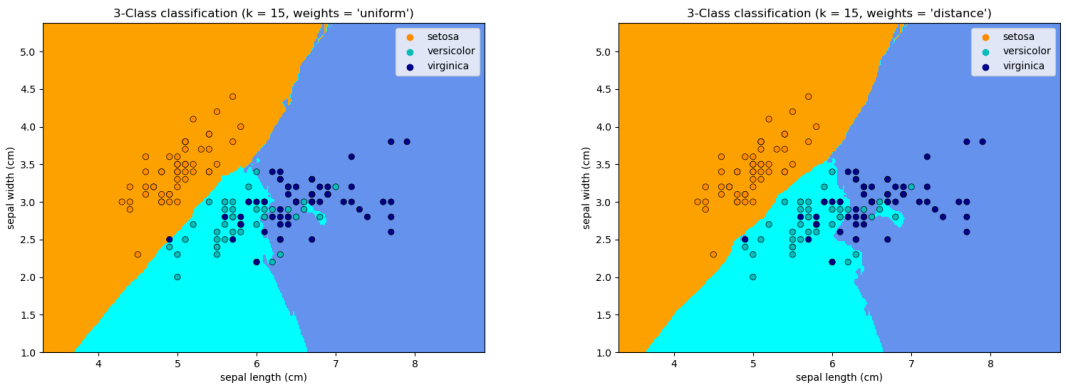

Figure 2 kNN classifier example on the iris dataset (Nearest Neighbors Classification— scikit-learn documentation, 2022).

Logistic Regression

The first appearance of logistic function was in 1838 by Pierre François Verhulst a Belgian mathematician.

Which then led to J.S. Cramer paper on "The Origins of Logistic Regression" (2002) where he explains in detail how does the function works.

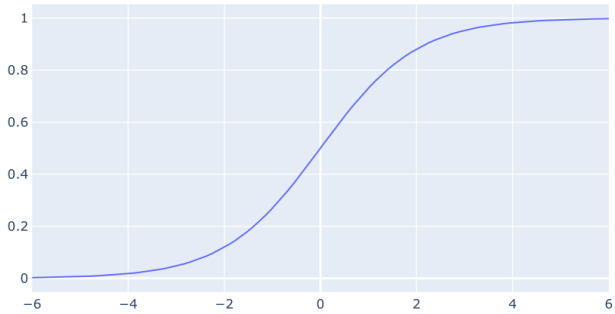

Figure 3 The logistic curve (Sigmoid function)

Logistic Regression is a machine learning algorithm that is used mainly for binary classification problems, whenever the target is categorical. Binary means the result is two outcomes so for example true/false or yes/no (Thomas W. Edgar, David O. Manz, 2017). Logistic regression is used in real life problems, for instance to predict a lung cancer or not by analysing the weight of someone and the cigarette packs that they smoke per day

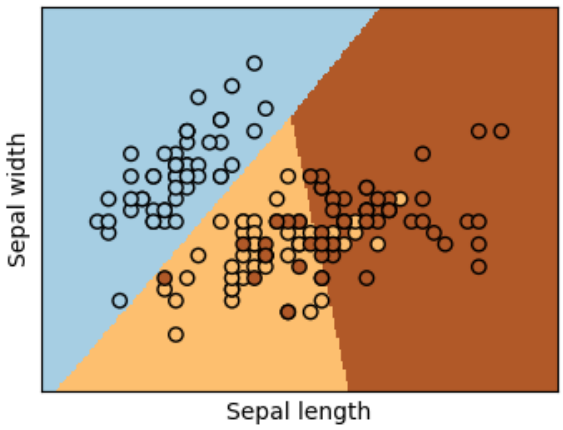

Figure 4 Logistic regression example on the iris dataset (Logistic Regression 3-class Classifier— scikit-learn documentation, 2022).

Support Vector Machine algorithm

It was in 1992 that Boser, Guyon and Vapnik introduced Support Vector Machines to the world with a paper that explain various parts of the algorithms.

Support vector Classifier (SVC) is a machine learning algorithm that follows the idea of support vector machines except it is a classifier, like kNN classifier, but the objective is to find a hyperplane that makes a clear division and distinctly classifies the different data points. The hyperplane is the best line or decision boundary that will divide space into classes. The algorithm chooses the best points that is going to create that hyperplane. The advantages of SVC are that it is performant, and it is not sensitive to overfitting which is an error in statistics whenever a function is too close to a limited set of data points. The disadvantages are that SVM is not the best choice when it comes to many features, and it is not adapted to non-linear problems.

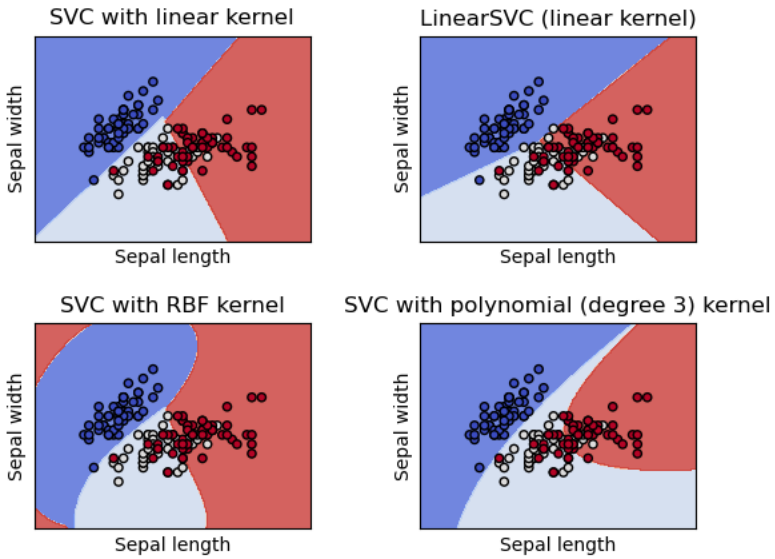

Figure 5 4 different types of SVC's applied on the iris dataset (Vector Machines — scikit-learn documentation, 2022).

Random Forest Classifier

The first appearance of an idea of random forest classifier was in 1995 (Ho, 1995), the method uses oblique decision trees which are adapted to optimize the training set accuracy. This method was initiated to overcome the problem of the traditional methods that were used, to be able to avoid the sacrifice of the generalization accuracy on unseen data. In randomly selected spaces the algorithm will build multiple trees and then it will combine the different classification of those trees. In 1997 Amit and German proposed a shape quantization and recognition approach for randomized trees and in 1998, Ho found a new method to solve the dilemma between overfitting and achieving a maximum accuracy

Random Forest is finally developed by Breiman in 2001, combining the methods of the previous works done on the subject to propose a new one. Each tree is acting as a classifier to determine the class of unlabelled instances by finding the majority where each classifier predicted a class label. The class that gets the most predicted is chosen to classify the instance. The pros if this algorithm is that it is powerful, accurate and has good performance on most of the problems, the cons are that we must find the number of trees that fits manually, and overfitting can occur easily. Random forest has a lot of real-world application, banking industry to detect credit card fraud or in healthcare for diabetes prediction or breast cancer prediction, it is also used in stock market and E-commerce (OpenGenius IQ).

Feature Scaling

Before training a machine learning model, we need to pre-process the data that we have. Feature scaling is one of the most important steps on pre-processing the data. There are two types of techniques for feature scaling, normalization and standardization.

"Normalization is used when we want to bound our values between two numbers, typically, between [0,1] or [-1,1]. While Standardization transforms the data to have zero mean and a variance of 1, they make our data unitless." (Baijayanta Roy, 2020)

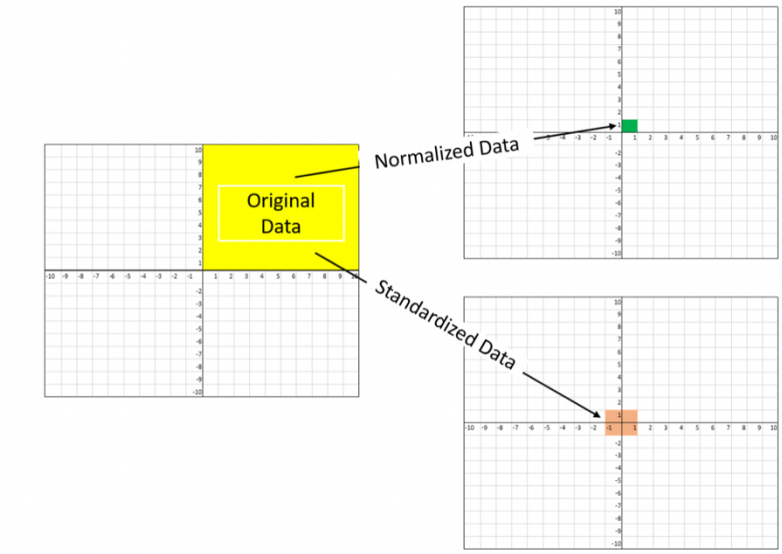

Figure 6 Difference between normalized and standerdized data

Machine learning algorithms work with numbers and therefore it does not understand what they represent, "A weight of 10 grams and a price of 10 dollars represents completely two different things" (Baijayanta Roy, 2020). Scaling helps the algorithm to find patterns easier and to give more precise and accurate results.

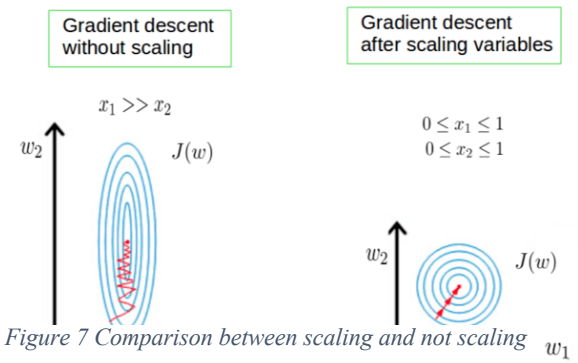

Figure 7 Comparison between scaling and not scaling

Confusion Matrices

A confusion matrix is a technique that summarize the performance of a classification algorithm. It is also a great tool to compare and display the different results which is why it is going to be an important technique to use in every algorithm of our research. Here is an example of a confusion matrix if we had to different categories named “positive” and “negative”.

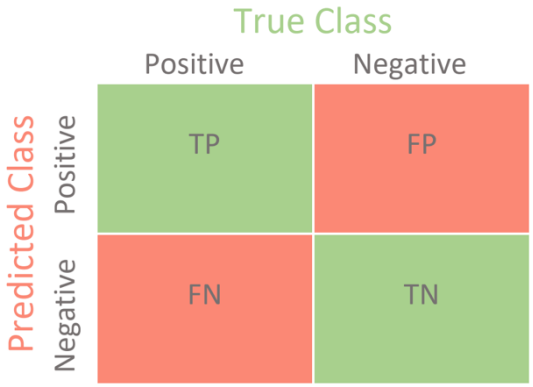

Figure 8 Confusion Matrix for Binary Classification

True Positive (TP): It refers to the number of predictions where the classifier correctly predicts the positive class as positive.

True Negative (TN): It refers to the number of predictions where the classifier correctly predicts the negative class as negative.

False Positive (FP): It refers to the number of predictions where the classifier incorrectly predicts the negative class as positive.

False Negative (FN): It refers to the number of predictions where the classifier incorrectly predicts the positive class as negative (Joydwip Mohajon, 2020).

Cross Validation

Cross validation is a model that evaluates and compares learning algorithms by using two segments of data, the first one is the one to train the model and the second one is the one to validate the model (Payam Refaeilzadeh, Lei Tang, Huan Liu, 2009). To get consistent accuracies it is usually ran multiple times so that the result is more consistent.

Research Methodology

Philosophy and strategy

The strategy for this project is to use the precision and the capacity of predicting of machine learning algorithms to predict whether someone has diabetes or not in the Pima Indians Dataset (Kaggle, 2016).

Machine learning is one of the most efficient ways of analysing data, it is fast and can handle large amount of data. There are multiple algorithms that can do predictive analysis and they are classification algorithms. This project will have 4 different algorithms analysing the same dataset, the strategy is to be able to have various results and compare them to find what is the best algorithm to predict. The scope of this project is to determine if machine learning algorithms can predict diabetes, and how precise and accurate they are.

National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK)

The dataset used for this project comes from this organisation created in 1950. Since then, the mission of the NIDDK is to do medical research and research training to spread scientific information on diabetes and other metabolic diseases (NIH, 2021). The institute is supporting government scientists that do research related with the institute mission, it also supports universities and other institutions that do medical research.

Pima Indians

Pima Indians are North American Indians who originally lived in Arizona. They called themselves the “River People” because they lived along the Gila and Salt rivers. They were traditionally farmers that lived in one room houses (Elizabeth Prine Pauls). In this paper, in the dataset that is used show patients that are Pima Indians heritage

Dataset

In this project the dataset that is going to be used is publicly available for everyone on the website Kaggle.com which is a platform where we can find various datasets and works done on them. It is real world data provided by the NIDDK. In this dataset we will find different and important parameters that are essential to be able to say if someone has diabetes:

- Pregnancy
- Glucose
- Blood Pressure
- Skin Thickness
- Insulin
- Body Mass Index (BMI)
- Diabetes pedigree
- Age

There is also an "Outcome" parameter available that shows who has or don't have diabetes. This dataset has a lot of work done on it, but it was chosen to propose another perspective and try to get new results.

Training

The dataset will need to be trained and prework to verify if everything is ready to be analysed. First it is important to extract the data from its original file without damaging it and retrieving every piece of information exactly like it is. Then it is time to check if there are not any columns that are not relevant for the analysis and select the ones that will be used. In fact, it is also very important to verify if there are no NULL values that could interfere with the algorithms and give results that would not be precise enough.

The perfect way to check the correlation with the different parameters is to plot the data in different ways and check how this will impact the result that is expected.

Software

Project Jupyter was born in 2014 and it is a non-profit and open-source software (Jupyter, 2022). Jupyter notebook is a software that uses Python as a language and that allows to run code in different parts for data analysis. This allows the user to run code one by one or to run everything from the beginning or even from a certain point for example. The benefit of this software is that it is both an IDE and a notebook that displays results with a lot of libraries available. This is important for the project to be able to have different parts for every algorithm that we will test so if there is an error it is possible to correct it without making the whole code fail.

Version Control

Version control within this project was done locally using Visual Studio Code. Since the project is not done with a group of people and that the Jupyter software allows us to be able to modify parts of the code without damaging the other outcomes, Visual Studio Code was the best option to code and to have control on the code file.

Algorithms

There are many classification algorithms available to work with but for this thesis only four of them were chosen: kNN, Logistic Regression, SVM and Random Forest. This choice was made because every single of those algorithms have their own particularity and they propose different methods to analyse data and they are also the best ones to do predictive analysis.

Logistic Regression

Logistic Regression is used to assign observations on discrete classes, it is only checking binary results. In the case of our thesis this is perfect because we want to know if yes or no someone has diabetes. It is often used in classification problems, and it uses probability to predict results. Logistic regression is different from a Linear regression, it is a more complex function that is used. The cost of the function is named “Sigmoid function”.

K-Nearest Neighbour algorithm

KNN is used for both classification and regression problems. The nearest neighbours are the data points that have the minimum distance with the new data point, K is going to be the number of data points that we will consider for the implementation of the algorithm. The algorithm is calculating the probability that the test data that belongs to the classes of the training data and then selects the highest probability. Choosing KNN in this thesis is relevant because the dataset has different categories that need to be correlated and tested to find the best accuracy.

Support Vector Machine algorithm

The objective of this algorithm is to find a hyperplane, among several features, that will classify the different data points. The goal is to find the perfect optimization line between the different categories and establish results from this optimization. The dimension of the hyperplane will depend on the number of features, it rarely exceeds 3 because whenever there is 2 features the hyperplane is a line and 3 means the hyperplane becomes a two-dimensional plane.

Random Forest Classifier

Random Forest Classifier is an algorithm that works both for classification and regression. How it works is simple, the algorithm builds an ensemble of decision trees and with a bagging method increases the overall result. The decision trees will merge to get the most accurate and stable result.

Implementation

For this project a Jupyter Project will be used running on Visual Studio Code which is a software that gives the opportunity to work with almost all the different programming languages and it is very easy to run and compile files on it. The project will run via Python Anaconda 3.8.8 and the computer used is a MacBook Pro M1 2020 with 8 Go memory.

Training and pre-processing the dataset

The first step when we get the data file (csv file) is to extract the dataset information into our code. The pandas library was imported in order to pull the data with the pd function. Simultaneously we display the head of the dataset to check if the columns are right and if there is none of them missing.

```
df = pd.read_csv('data/diabetes.csv')
df_name = df.columns
df.head()
```

✓ 0.3s Python

	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

Figure 9 Screenshot of the import of the dataset and printing the head of it

Then another important step is to run the info function which will give us a detail of the different number of columns but also the type of data that will be stored in those different categories.

```
df.info()
```

✓ 0.1s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Figure 10 Dataset global information

Another overview is interesting and useful to check is using the describe function to see more information about the dataset.

```
df.describe()
```

✓ 0.5s Python

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

+ Code + Markdown

Figure 11 Detailed description of the dataset

We can see the count of every category which have to be the same, also displays the mean, the standard deviation, the minimum value, the maximum value and finally ¼, ½, ¾ of every labels.

Then we display the first plot to see how all the categories are related to the Outcome label and how they have different behaviours.

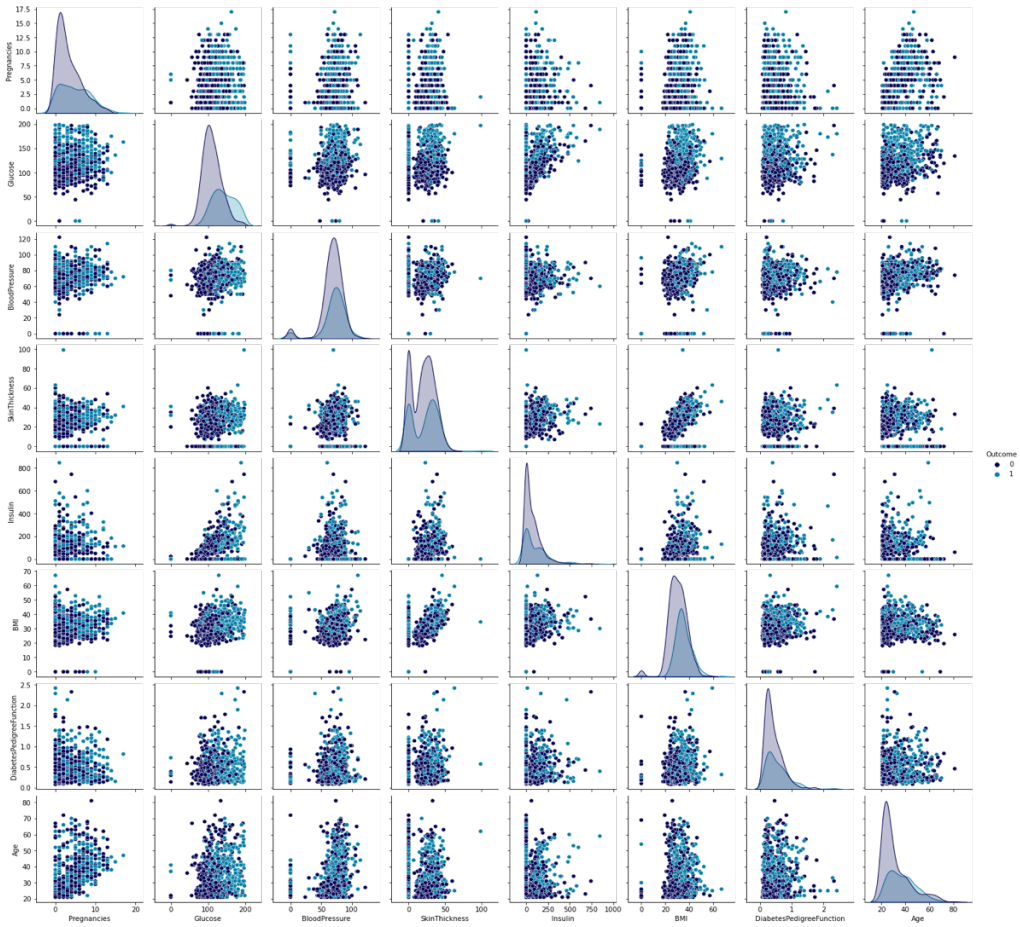

Figure 12 Every categories relation to the "Outcome" parameter

After that it was chosen to display another figure but this time it displays the correlation numbers between all the labels of the dataset, we chose to use colours to display them, so it is easier to understand the relation.

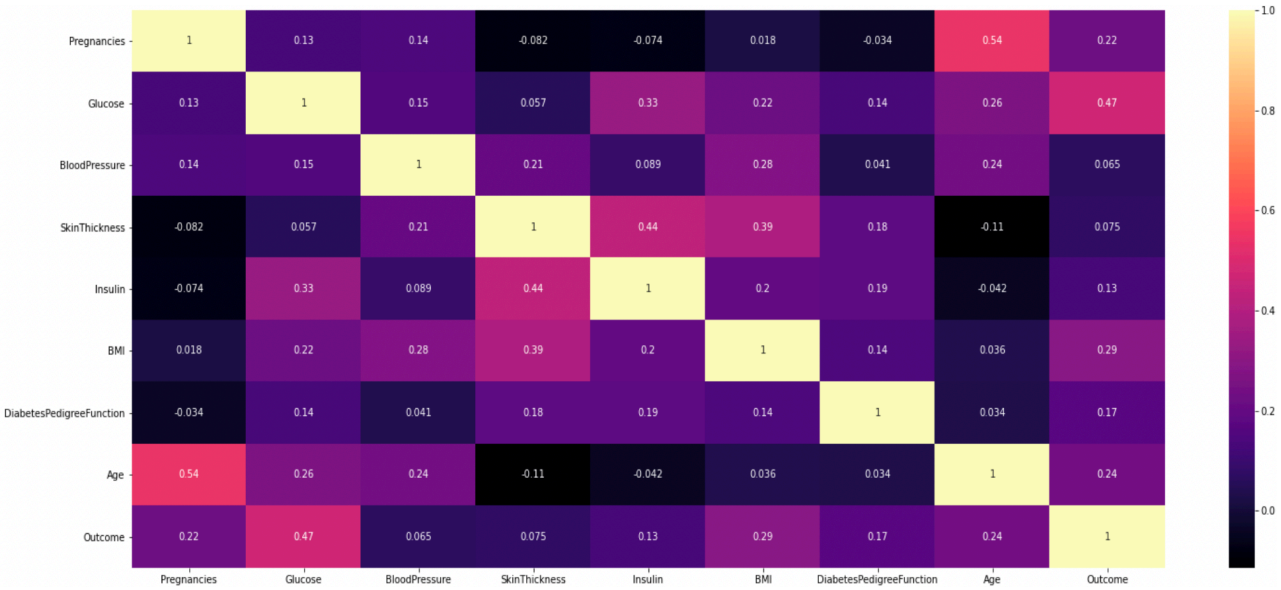

Figure 13 Correlation plot between all the parameters of the dataset

The process of pre-processing is an important step before applying the different algorithms to our dataset. It was decided to use a standard scaler for this dataset by importing the function from the sklearn pre-processing library.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train=sc.fit_transform(X_train)
X_test= sc.transform(X_test)
df
```

✓ 0.3s

	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
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

Figure 14 Standard scaling the dataset

The y value is only going to be the "Outcome" category because it is the value that is going to be predicted and to do that, we also need to delete that category from the X values.

```
y = df["Outcome"]
X = df.drop(["Outcome"], axis=1)
```
✓ 0.1s

Figure 15 Code for the drop of "Outcome" from the X values and adding it only to the y values

Applying the algorithms to the dataset.

Note that every piece of code for every algorithm will have the same structure when it comes to apply it to the dataset, first the import of the algorithm, the implementation and then displays the results with a classification reports and confusion matrices.

Logistic Regression code

The first implementation is logistic regression, to apply this algorithm first we need to import it from the sklearn linear model library. Then we create a name for the model (lr) and call the function. Then we attribute to that model the different values for X and y, and create a prediction value which will be named y_pred. Then we create two different scores, the training score and the testing score, we attribute the different values that we already defined to the respective scores. There is also the code to print the confusion matrix with the testing score and finally printing the classification report with the two different scores.

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

lr = LogisticRegression()
lr.fit (X_train, y_train)
y_pred = lr.predict(X_test)

train_score = accuracy_score(y_train, lr.predict(X_train))
test_score = accuracy_score(y_test, y_pred)


cfm = confusion_matrix(y_test,y_pred)
accuracy = round(accuracy_score(y_test, y_pred), 2)
```

```
sns.heatmap (cfm,annot=True,cmap='Blues',fmt=".0f")
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Accuracy Score: {0}'.format(accuracy), size=10)
plt.show()


print("Testing score : {:.3f}".format(test_score))
print("Training score : {:.3f}".format(train_score))
print('Classification Report: \n', classification_report(y_test, y_pred))
```

Figure 16 Logistic Regression implementation code

K-Nearest Neighbour algorithm code

For this algorithm there is something different, as we know kNN needs to have the K number to be able to run the model. After importing the necessary libraries, we create a list, here within a range from 1 to 30 to find the best K number for this model. There is then a function that goes through every K number and with cross validation shows what is the best one. In the end of this piece of code we can see that there are lines to print everything on a figure, and we can observe on that figure that the best k number seems to be 15 neighbours. That's the number that we will apply to the model.

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
k_list= range (1,30)
ave_scores = []
for k in k_list:
    knc = KNeighborsClassifier(n_neighbors= k)
    knc.fit(X_train, y_train)
    scores = cross_val_score(knc, X,y, cv= 5, scoring="accuracy")
    ave_scores.append(round(scores.mean(),3))

plt.grid
plt.figure(figsize = (10,10))
plt.plot(k_list,ave_scores, marker='o', markerfacecolor='orange', markersize=10)
plt.xlabel("Number of nearest neighbours")
plt.ylabel("Average model accuracy")
```

Figure 17 KNN implementation code to find the best k number

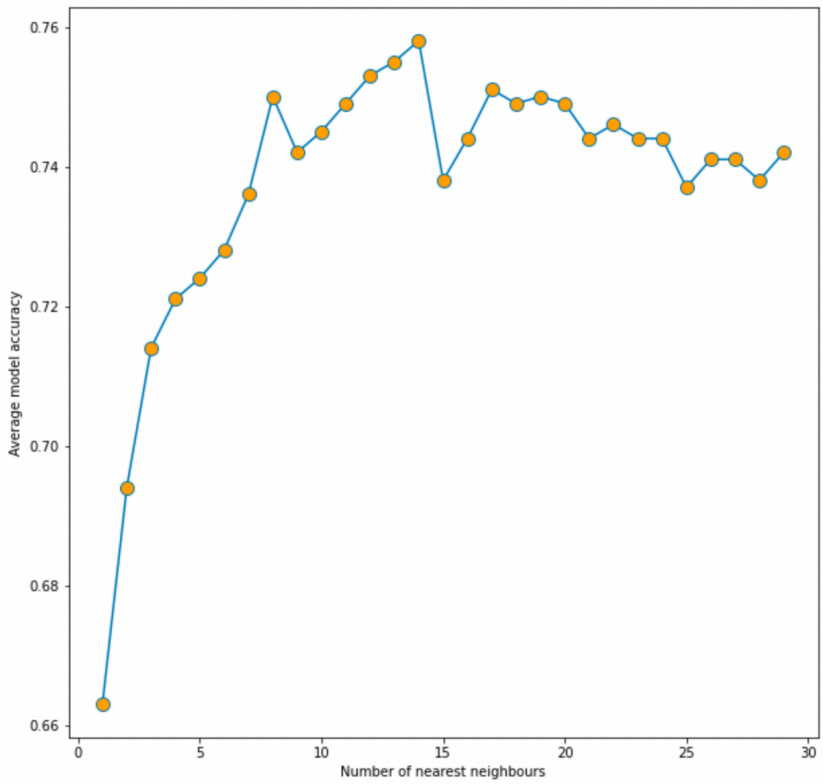

Figure 18 Plot of the KNN average accuracies depending on the number of nearest neighbours

```
knc = KNeighborsClassifier(n_neighbors= 15)
knc.fit(X_train, y_train)
y_pred = knc.predict(X_test)


train_score = accuracy_score(y_train, knc.predict(X_train))
test_score = accuracy_score(y_test, y_pred)


cfm = confusion_matrix(y_test,y_pred)
accuracy = round(accuracy_score(y_test, y_pred), 2)
sns.heatmap (cfm,annot=True,cmap='Blues',fmt=".0f")
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Accuracy Score: {0}'.format(accuracy), size=10)
plt.show()


print("Testing score : {:.3f}".format(test_score))
print("Training score : {:.3f}".format(train_score))
print('Classification Report: \n', classification_report(y_test,y_pred))
```

Figure 19 KNN algorithm implementation code

SVC algorithm code

The SVC code is using the same method of coding as the previous ones, we import the SVC function from the SVM library to apply it to the dataset.

```
from sklearn.svm import SVC

svc = SVC()
svc.fit(X_train, y_train)
y_pred = svc.predict(X_test)

train_score = accuracy_score(y_train, svc.predict(X_train))
test_score = accuracy_score(y_test, y_pred)

cfm = confusion_matrix(y_test,y_pred)
accuracy = round(accuracy_score(y_test, y_pred), 2)
sns.heatmap (cfm,annot=True,cmap='Blues',fmt=".0f")
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Accuracy Score: {0}'.format(accuracy), size=10)
plt.show()


print("Testing score : {:.3f}".format(test_score))
print("Training score : {:.3f}".format(train_score))
print('Classification Report: \n', classification_report(y_test,y_pred))
```

Figure 20 SVC algorithm implementation code

Random Forest algorithm code

The Random Forest code follows the same path as the previous ones, and we import the Random Forest Classifier function from the ensemble library to apply it to the dataset. There is one thing that changes when we create the function there are different parameters to enter. First parameter is the number of estimators which will be leaved at the default number which is a 100, then there is the criterion set as gini (default one) which measure the quality of a split, there is also the max_depth which represents the maximum depth of the tree.

There are also two other parameters, and they are called mini_samples_leaf and mini_samples_split, the first one is the minimum number of samples required to be at a leaf node and the second one is the minimum number of samples requires to split an internal node.

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators= 100, criterion = 'gini', max_depth = 3,
max_features = 'auto', min_samples_leaf = 2, min_samples_split = 4)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)

train_score = accuracy_score(y_train, rf.predict(X_train))
test_score = accuracy_score(y_test, y_pred)

cfm = confusion_matrix(y_test,y_pred)
accuracy = round(accuracy_score(y_test, y_pred), 2)
sns.heatmap (cfm,annot=True,cmap='Blues',fmt=".0f")
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Accuracy Score: {0}'.format(accuracy), size=10)
plt.show()

print("Testing score : {:.3f}".format(test_score))
print("Training score : {:.3f}".format(train_score))
print('Classification Report: \n', classification_report(y_test,y_pred))
```

Figure 21 Random Forest Classifier implementation code

Cross Validation

This part is basically where we select all the different functions that we created earlier and apply cross validation to it. Here we can see that the number of folds is equal to 10 so it will print 10 different accuracies for each of the cross-validation calls.

```
lr_score = cross_val_score (lr,X,y,cv=10,scoring='accuracy')
knc_score = cross_val_score (knc,X,y,cv=10,scoring='accuracy')
svc_score = cross_val_score (svc,X,y,cv=10,scoring='accuracy')
rf_score = cross_val_score (rf,X,y,cv=10,scoring='accuracy')
print(lr_score)
print(knc_score)
print(svc_score)
print(rf_score)
```

Figure 22 Cross validation code

Results and Analysis

Results and classification report for each algorithm

Logistic Regression

```
Testing score : 0.802
Training score : 0.764
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.91      0.86       130
           1       0.75      0.58      0.65        62

    accuracy                           0.80       192
   macro avg       0.78      0.74      0.76       192
weighted avg       0.80      0.80      0.79       192
```

Figure 23 Logistic regression accuracies and classification report

K-Nearest Neighbour algorithm

```
Testing score : 0.781
Training score : 0.767
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.89      0.85       130
           1       0.71      0.55      0.62        62

    accuracy                           0.78       192
   macro avg       0.76      0.72      0.73       192
weighted avg       0.77      0.78      0.77       192
```

Figure 24 KNN algorithm accuraies and classification report

Support Vector Classifier

```
Testing score : 0.776
Training score : 0.825
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.90      0.84       130
           1       0.71      0.52      0.60        62

    accuracy                           0.78       192
   macro avg       0.75      0.71      0.72       192
weighted avg       0.77      0.78      0.77       192
```

Figure 25 SVC algorithm accuracies and classification report

Random Forest Classifier

```
Testing score : 0.786
Training score : 0.800
Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.92      0.85       130
           1       0.76      0.50      0.60        62

    accuracy                           0.79       192
   macro avg       0.78      0.71      0.73       192
weighted avg       0.78      0.79      0.77       192
```

Figure 26 Random Forest algorithm accuracies and classification report

The results displayed show that logistic regression has the best testing accuracy with 80.2% and the best training score is the SVC with 82.5%.

Confusion matrices and analysis

The different confusion matrices for each algorithm.

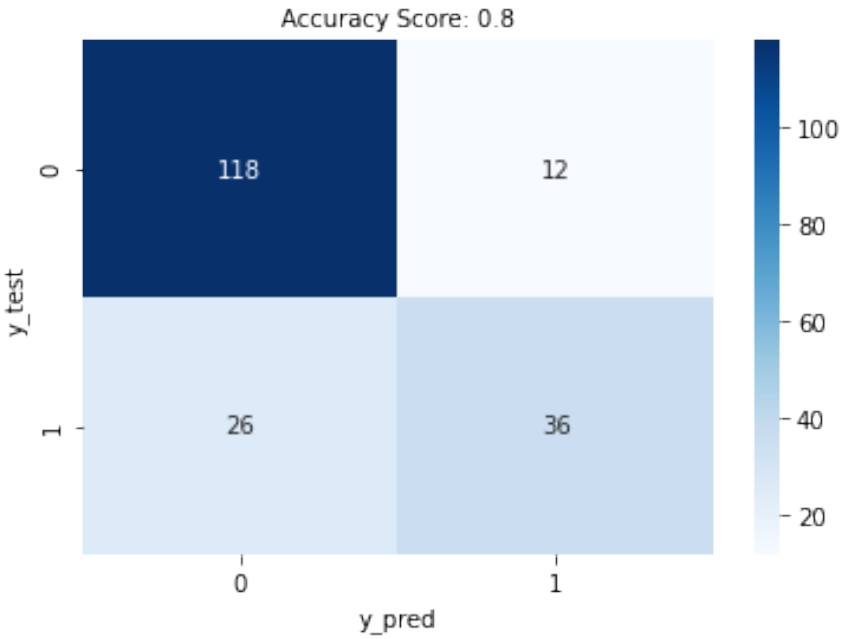

Figure 27 Logistic Regression confusion matrix

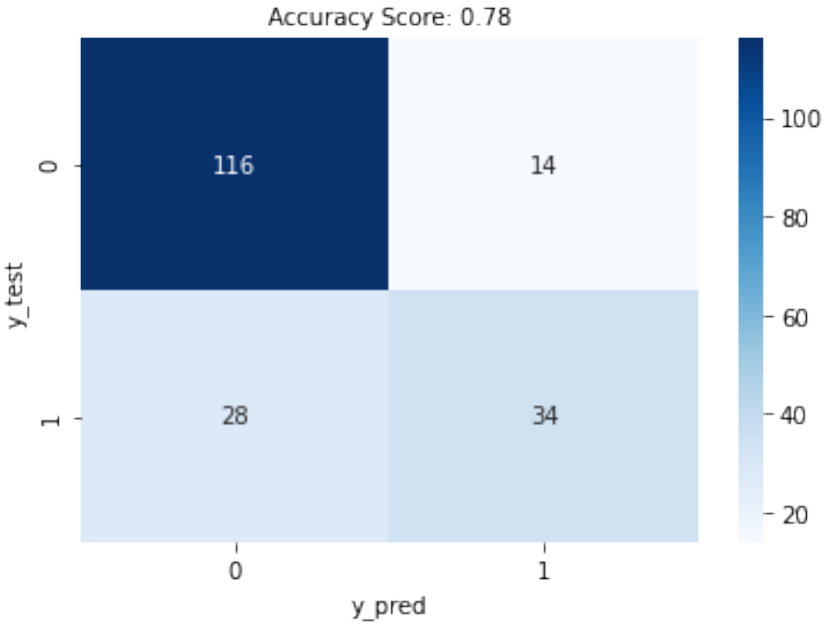

Figure 28 KNN confusion matrix

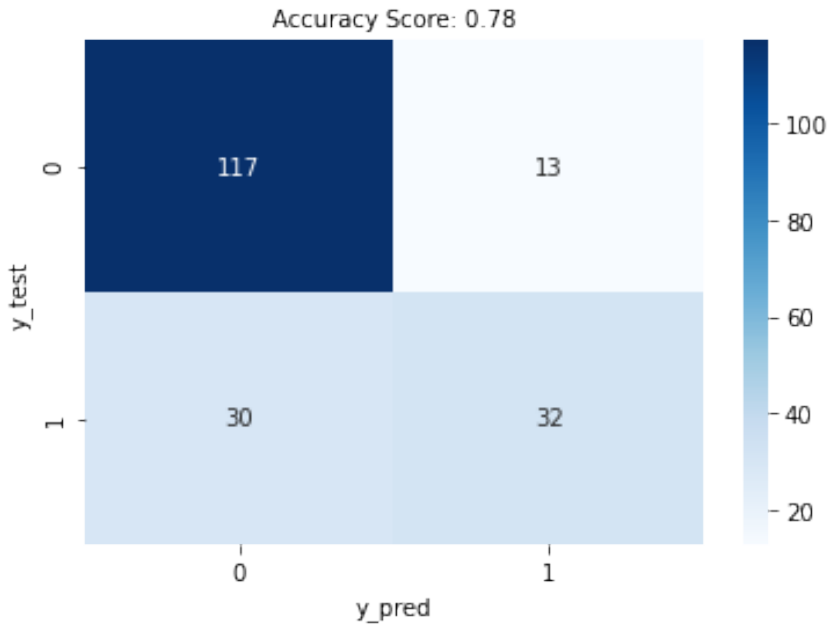

Figure 29 SVC confusion matrix

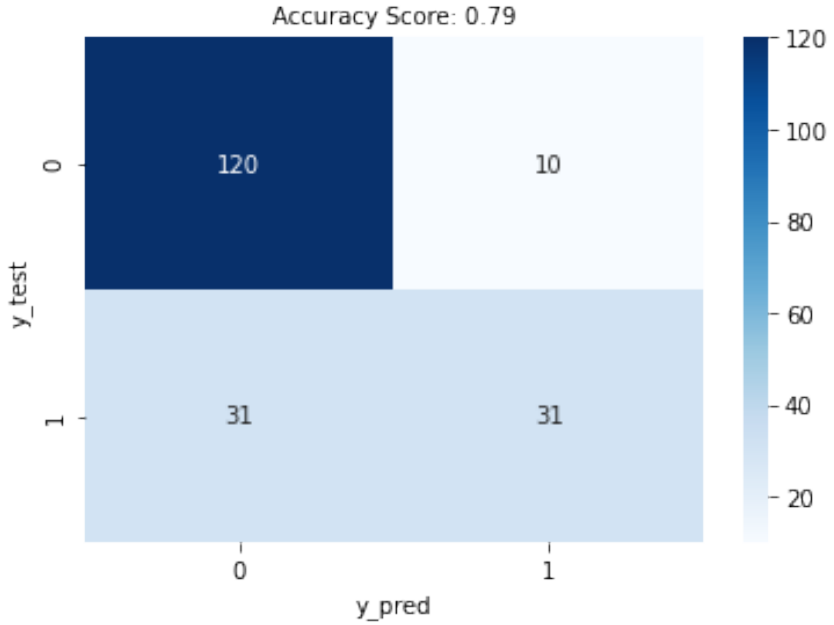

Figure 30 Random Forest confusion matrix

Cross Validation Scores and analysis

```
[0.71428571 0.77922078 0.80519481 0.71428571 0.74025974 0.76623377
 0.81818182 0.80519481 0.75       0.82894737]
[0.75324675 0.7012987  0.68831169 0.62337662 0.71428571 0.76623377
 0.75324675 0.79220779 0.78947368 0.73684211]
[0.74025974 0.74025974 0.74025974 0.71428571 0.72727273 0.80519481
 0.75324675 0.80519481 0.76315789 0.78947368]
[0.76623377 0.71428571 0.74025974 0.7012987  0.71428571 0.79220779
 0.77922078 0.79220779 0.73684211 0.81578947]
```

Figure 31 Cross Validation scores for every algorithm

In these four different arrays we can see the different results of cross validation with ten folds. The best results from every array are:

- Logistic Regression: 83%
- KNN: 79%
- SVC: 80.5%
- RF: 81.5%

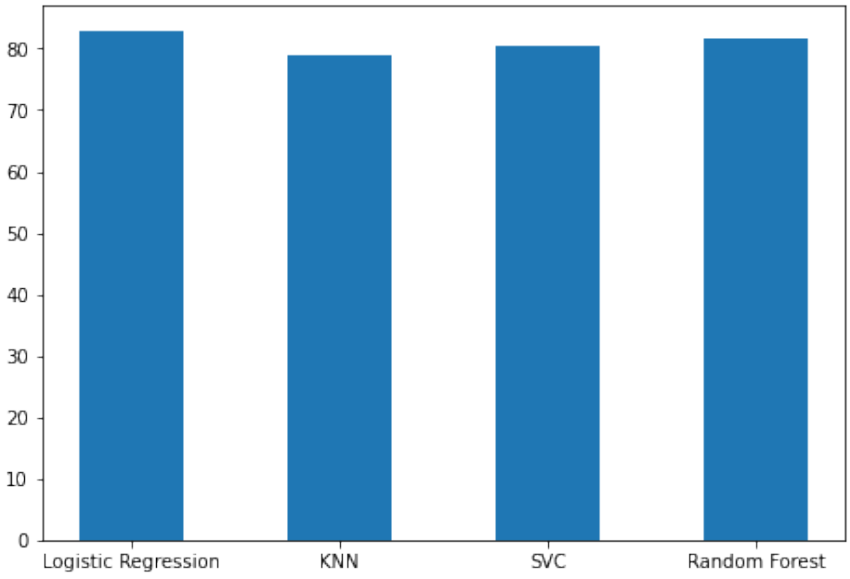

Figure 32 Best Cross Validation Plot

Like the testing scores, the best algorithm that predicts diabetes with cross-validation is Logistic Regression (83%). We can also see that every algorithm chosen were very close to the same results that shows the efficiency and the accuracy of the 4 algorithms that were chosen for this project.

Project Management

Methodology

The waterfall approach was used in this project. Every step was done in chronological order, from the research part at the beginning to the dissertation.

This method depends on the different requirements of the project that we are working on. On this case the first step was to search for a dataset that could suit the idea of the project. Then find the methodology that will be adopted for the analysis and the prediction, in this case classification algorithms were obvious, but some research had to be done on what algorithms will be used.

When those steps are done, we can move forward on the development of the project, and all the coding part. First the design must be investigated and prepared to be able to implement it well later. When the different coding methods were studied the implementation is the most important step, because we must make sure everything is in the correct place, and everything is running as planned. Then whenever we have the results, the next step is to start the dissertation.

Time Management

It wasn't very easy to work without a supervisor that helped to handle everything and be aware of how the project was going forward. The main focus for me was first, the research part and the coding part, to be able to have a solid thesis with concrete and working results prediction analysis. A major part of the semester was working on finding a suitable dataset for the problem, finding the algorithms and applying everything to the code and make it work properly. Then, the

last three weeks were only about the dissertation part of the thesis. There is a Gantt Chart to show how the time was managed for this project.

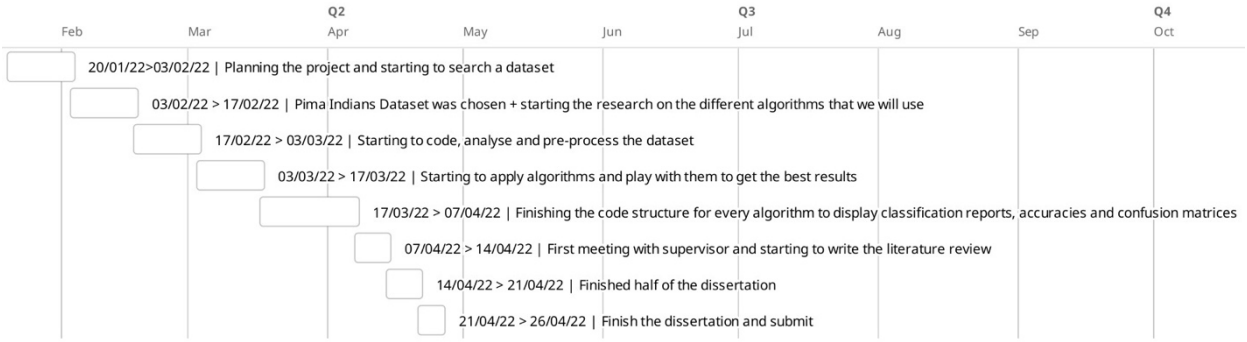

Figure 33 Time Management Gantt Chart

Risk Management

There are multiple risks while doing a project like this one, that is why we must be ready for any problems that could come during the making of the thesis.

The first one is the most obvious one, we don't want to lose all the work that has been done so we must save it, the issue is we are not sure that our computer will not crash and that we will not lose all the project that is why for the code portion an external hard drive save was made regularly to keep the work up to date and for the dissertation part the easiest way to keep the work safe was to use OneDrive. In fact, Microsoft Word propose an automatic saving every time the file is being changed, that is what we need.

Another risk is to not be able to finish the project the way that it was planned, to deal with that problem there are multiple solution depending on the degree of the problem. If we realise the work is a lot behind a couple of weeks before submitting the best way to deal with that is to ask for a deferral or an extension explaining why we are behind. To avoid this problem, we can also be prepared for that with a schedule to insure everything is planned to be on time.

One thing that we can't know in advance is health and if we are ill the best way is to rest and tell the supervisor and the module leader about the situation to have a deferral planned or an extension. There is also an important risk that we have to keep in mind it is the stress, a lot of students in this final part of the year have a lot of coursework and exams to submit, and it can be stressful. The best way to deal with it is to take time to rest and do activities, to be able to be efficient whenever the work has to be done.

Feedback and Communication

Most of the communication for this project was done with Microsoft teams meeting calls, messages and emails.

On the first semester of thinking about this thesis, I had a supervisor (C. Pridgeon) who helped me find my way to what I wanted to do for this project. The focus was on data analysis, I knew that it was the path that I wanted to take, so at first different tools were considered to find the dataset that I wanted to work with. After a few weeks of meetings with my first supervisor, predictive analysis was considered, and it was the idea to continue with that. Then I found this interesting subject, which was diabetes, the dataset was not chosen yet, but it was the main idea that was important. After that, the proposal was made to my supervisor which confirmed that it was a good path to take. Then, after submitting my ethics application and my proposal with a mini literature review, I didn't have any news of this supervisor and just got an email from him saying that he was living university.

The second semester was very quiet on feedback and communication because I wasn't given a new supervisor for the whole semester, some help was proposed by the module leader by giving the contacts of lecturers that matched with the thesis subject. It was only three weeks before submission that my new supervisor (A. Brooks) was attributed, and after that it was easier to go forward in the project. In a few weeks we were able to go through the most important aspects of the dissertation and it was the help that I needed to finish this project. This supervisor gave me feedback on the first version of my literature review and was answering very quickly whenever I had a question on whether the structure or the content of the thesis.

Legal, Ethical and Social

Legal and Social

The legal issues are considered in the project with the General Data Protection Regulation (GDPR) (Ico) tailored with the Data Protection Act (DPA) (2018). The dataset that we are using is completely free access and protect the identity of the patients that were involved.

Ethical

The project has been approved by the university ethics platform.

Ethics application number: P130492

Reflexion

The project management wasn't the easier part for me because of the lack of information and help. It was with the help of my second supervisor for the last 3 weeks before finishing this thesis that it became much clearer, especially the dissertation part which was not handled perfectly. Overall, the coding part was in my opinion very well managed because when starting the dissertation 99% of the code was already done and the results and analysis were done. The research part was done prior to the coding to be able to have a clear scope for the project even if it could have changed during in the flow of doing it.

Conclusion

Summary

This thesis objective was to predict diabetes using machine learning techniques using a dataset with different parameters involved.

Overall, the results are mostly convincing and high, and that shows us the power of machine learning when it comes to do predictive models.

The final results accuracies could be higher with more work done on it; a project that was done in this amount of time can never be perfect with 100% accuracies.

Relevance to the real world

Machine learning are more and more used in the healthcare department, this thesis shows an approach to predict diabetes with four different algorithms. The interesting this is that these different algorithms could be used on other real-world problems like breast cancer prediction. Predictive analysis is not only relevant in healthcare but also for instance in the cybersecurity area where prediction can be used to improve the performances and detect anomalies, understanding the behaviour and ameliorate the data security.

The importance of this type of thesis to the real world is huge because it could find information that was not identified by humans and change the scientific vision on the given problem.

Diabetes is a disease that touches a lot of people in the UK and in the world like it was told earlier and to bring another work, a new work on the subject is relevant because it brings different results and outcomes that could be analysed in the future. This thesis is also oriented only on women because of the dataset, and it is interesting to see the impact of the pregnancies and the ethnical influence when it comes to diabetes. We can also see that the results are very high on accuracy and that is also a confirmation that machine learning algorithms are very useful for this type of research.

Limitation

The major limitation with this thesis is that the dataset is oriented on only women from Pima Indians heritage. In fact, we don't have a large scope and the final conclusions are only about this range of information. Also, the lack of time to develop more and improve the models to get more accurate results was a huge limitation. Like it was told on the management part, the lack of feedback and communication on almost the whole time were the thesis had to be done was an important limitation for me because I didn't really know what was the most important to do or not and how it had to be done.

Future work

Future research could be done on larger datasets which will cover more than one gender and with more than one ethnicity to touch at different results. The research of more efficient algorithms could be a solution to push forward on this thesis, for example neural networks that were not used in this project. Furthermore, it could be interesting to have more work done on the different models, for example with the use of hyperparameter tuning which could have helped have sharper results.

Reflection

This project was very captivating and gave me more interest and knowledge about the diabetes disease. The programming part was very challenging and helped me progress in terms of knowledge but also application. The research part on the whole project was not easy but it was the most important part in my opinion to be able to understand clearly what the thesis was about and how it was going to be managed. Of course, the literature review is very helpful for the reader to understand the subject that we are dealing with, but it had a huge importance and influence on the vision that I had during the making of this thesis.

References

McCoy K. (2009) The History of Diabetes *Everyday Health*
https://www.everydayhealth.com/diabetes/understanding/diabetes-mellitus-through-time.aspx

Checking your blood sugar levels *Diabetes UK* https://www.diabetes.org.uk/guide-to-diabetes/managing-your-diabetes/testing

Preventing type 2 diabetes *Diabetes UK* https://www.diabetes.org.uk/preventing-type-2-diabetes

Written by an editor (15/01/2019) Diabetes prevalence *Diabetes.co.uk*
https://www.diabetes.co.uk/diabetes-prevalence.html

Mayo Clinic (2021) Type 2 diabetes https://www.mayoclinic.org/diseases-conditions/type-2-diabetes/symptoms-causes/syc-20351193
https://www.mayoclinic.org/diseases-conditions/type-2-diabetes/diagnosis-treatment/drc-20351199

Mayo Clinic (2021) Type 1 diabetes https://www.mayoclinic.org/diseases-conditions/type-1-diabetes/symptoms-causes/syc-20353011

NHS (2021) About insulin https://www.nhs.uk/conditions/type-1-diabetes/about-insulin/
https://www.bupa.co.uk/health-information/diabetes/type-2-diabetes

NHS (2021) Gestational diabetes https://www.nhs.uk/conditions/gestational-diabetes/

Manchester 1824 (2021) Type 2 diabetes missed or diagnosis delayed for 60,000 UK people in 2020
https://www.manchester.ac.uk/discover/news/type-2-diabetes-missed-or-diagnosis-delayed-for-60000-uk-people-in-2020/

Ali R. (2020) Predictive Modeling: Types, Benefits, and Algorithms *ORACLE NETSUITE*
https://www.netsuite.com/portal/resource/articles/financial-management/predictive-modeling.shtml

Marr B. (2020) How Is Artificial Intelligence and Machine Learning Used in Engineering? *Forbes* https://www.forbes.com/sites/bernardmarr/2020/02/07/how-is-artificial-intelligence-and-machine-learning-used-in-engineering/?sh=2592ece94a85

Asiri S. (2018) Machine Learning Classifiers *Towards Data Science* https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623

Java T point (2021) Classification algorithm in Machine Learning https://www.javatpoint.com/classification-algorithm-in-machine-learning

Java T point (2021) Support Vector Machine Algorithm https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

Keith D. Foote (2021) A brief History of Machine Learning *Dataversity* https://www.dataversity.net/a-brief-history-of-machine-learning/#

T M & Hart P E. (1967) Nearest neighbour pattern classification. IEEE Trans. Inform http://garfield.library.upenn.edu/classics1982/A1982NF37700001.pdf

Chai W. (2020) A Timeline of Machine Learning History *TechTarget* https://www.techtarget.com/whatis/A-Timeline-of-Machine-Learning-History

Thomas W. Edgar, David O. Manz (2017) Research Methods for Cyber Security https://www.sciencedirect.com/book/9780128053492/research-methods-for-cyber-security

Belyadi H., Haghighat A. (2021) Machine Learning Guide for Oil and Gas Using Python. https://www.sciencedirect.com/topics/computer-science/logistic-regression

Gandhi R. (2018) Support Vector Machine — Introduction to Machine Learning Algorithms *Towards Data Science* https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Scikit-learn: Machine Learning in Python, Pedregosa et al. (2011), JMLR 12, pp. 2825-2830

Ho, T. K. (1995). Random decision forests. *Document analysis and recognition Proceedings of the third international conference*, Montreal, Quebec, Canada (Vol. 1, pp. 278–282). New York City, NY: IEEE

Meena M. (2022) Applications of Random Forest. *OpenGenus IQ* https://iq.opengenus.org/applications-of-random-forest/

Mohajon J. (2020) Confusion Matrix for your multi-class machine learning model *Towards Data Science* https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826

Roy B. (2020) All about Feature Scaling. *Toward Data Science* https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35

Stone M. (1973) Cross-validatory Choice and Assessment of Statistical Prediction https://rss.onlinelibrary.wiley.com/doi/epdf/10.1111/j.2517-6161.1974.tb00994.x

Refaeilzadeh P., Tang L., Liu H. (2009) Cross-Validation *Encyclopedia of Database Systems*. https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9_565#howtocite

Chichkoyan S. (2021) Project Proposal/Mini literature Review. *6000CEM Submission*

HolyPython (2020) Support Vector Machine History. *Holy Python* https://holypython.com/svm/support-vector-machine-history/

The Editors of Encyclopaedia Britannica (1998) Pima. *Britannica* https://www.britannica.com/topic/Pima-people

Jupyter (2022) Project Jupyter's origins and governance *Jupyter About Us* https://jupyter.org/about

Kaggle (2016) Pima Indians Diabetes *UCI Machine Learning* https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

eshwitha_reddy (2020) Advantages and Disadvantages of different Classification Models. *GeeksforGeeks* https://www.geeksforgeeks.org/advantages-and-disadvantages-of-different-classification-models/

Kambria editor (2019) Logistic Regression for Machine Learning and Classification. *Kambria* https://kambria.io/blog/logistic-regression-for-machine-learning/

Gupta S. (2020) Pros and cons of various Machine Learning algorithms. *Toward Data Science* https://towardsdatascience.com/pros-and-cons-of-various-classification-ml-algorithms-3b5bfb3c87d6

Arora H. (2019) Pros and Cons of Predictive Analytics in Healthcare *Dataversity* https://www.dataversity.net/pros-and-cons-of-predictive-analytics-in-healthcare/#

Tutorialspoint (2022) Machine Learning - Logistic Regression. *Tutorialspoint* https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_logistic_regression.htm

Breiman L. (2001) Random Forests. *SpringerLink* https://link.springer.com/article/10.1023/A:1010933404324

Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., & Tang, H. (2018). Predicting diabetes mellitus with machine learning techniques. Frontiers in genetics, 515. https://www.frontiersin.org/articles/10.3389/fgene.2018.00515/full

National Institute of Diabetes and Digestive and Kidney Diseases (2021) *National Institute of Diabetes and Digestive and Kidney Diseases* (NIDDK) https://www.nih.gov/about-nih/what-we-do/nih-almanac/national-institute-diabetes-digestive-kidney-diseases-niddk

Brownlee J. (2018) A Gentle Introduction to k-fold Cross-Validation. *Machine Learning Mastery* https://machinelearningmastery.com/k-fold-cross-validation/

Narkhede S. (2018) Understanding Confusion Matrix. *Towards Data Science* *https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62*

GeeksforGeeks (2022) Understanding Logistic Regression. *GeeksforGeeks* https://www.geeksforgeeks.org/understanding-logistic-regression/

Christopher A. (2021) K-Nearest Neighbor. *The Startup* https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4

Appendices

Table of Figures

Project Proposal

This project will deal with a known disease named diabetes. Diabetes is a disease that causes a person's blood sugar level to become too high. There are two types of diabetes, type 1 and type 2. Type 1 is when the body's immune system attacks and destroys the cells that produce insulin. Type 2 is when the body does not produce enough insulin, or the body's cells do not react to insulin. Type 2 is the more common one, in fact, this lifelong condition concerns around 90% of adults that have diabetes in UK. It is an important disease that touches more that 463 million people on earth on 2019, so it is a major problem to deal with. Predicting who can maybe affected by this disease can help to prevent from severe cases of diabetes. By doing this study we could conclude with multiple algorithms outputs accuracies with predictions.

The best thing is to get organized before starting a project. So, I will have to plan dates for every step that I am predicting to go through. A Gantt chart will be used to do prepare every task that I will plan for the project.
This project is about making predictions on data so logically the main topic that we are going to use is Machine learning. Machine learning is a branch of AI that can learn from data, and it is a method of data analysis that automates model building. There are multiple algorithms that can be used with Machine Learning (e.g., Linear Regression, Logistic Regression, Decision Tree, SVM, Naive Bayes, NNK-Means, Random Forest). It also helps doing predictions on different types of data. Machine learning algorithms really intrigues me because it is so diversified, there is a tone of ways to work on data. That why I chose this method for my project, to be able to have different results. And of course, I think nowadays it is one of the best ways to make predictions on data. It becomes more and more used over the last 10 years and there is a reason for that.

The first step of our study would be to get informed on diabetes deeply so that we can find a suitable dataset for our project. After finding the data we must analyze it, there could be missing information or other problems. This step is the most important one because we must make sure before we apply anything on it that it is well prepared. Not getting pre-processing can lead to misleading results and the whole study could be compromised.

So, we pre-process it so it is ready to apply different types of algorithms on it and so that we will be able to get results. Once our data is ready for processing, we will apply the chosen algorithms on it to be able to get the higher possible prediction accuracy. Multiple algorithms will be taken to be able to make a comparison between them. The results that we are waiting for could be displayed in different ways, tables, graphics that will help people understand clearly what these results could mean if we present it.

Concerning this plan there are a lot of elements that can be discussed, first, how much data do we need to be able to do this study. The problem is we can't select something that will take years to analyze so we will have to choose data that can be worked with during the period of a semester, of course results will not be very precise. The solution that came to me is to take data from a specific location for example a country or even a city, that could help us to stay precise on the analysis but also to achieve it on time. Then, a disease is always very delicate to deal with, so we must be careful with the data analysis to be sure that everything is controlled. That we are not providing false data or that we are not showing something not true. The best way of doing that can be that we take data that has already work been done on it. The real-life objective is to add research on this subject maybe find new results for it.

Like we said at the beginning there are a lot of people with this disease. The objective of this project is also to find a new vision, a new method on this and so help being more effective predicting these types of diseases. To be able to be sure our solution and our results are successful we will have to compare with work that has already been done. With some research I found that it was a subject that interested a lot of people and that there are multiple different studies on it. Some of them use Machine Learning techniques and that is what is very interesting for us because our results will obviously not be the exact same as them so it will be good to see what is similar and what is very different between these studies and my project.

This is my proposal for the final year project. It is a very ambitious theme, because diabetes concerns a lot of people and so there has been already researches on this subject. But since every dataset is different, maybe it will lead to something new and useful for everyone. I understand that one semester will not be enough to cover all the analysis and model application possible on

a dataset. That is why I will choose to go deeply into some of the algorithms that I find interesting to work with. I haven't found a proper dataset that I want to work with yet, but I already saw one or two that will fit this idea and study, for example Linear regression. The next step for me will be to find a dataset and start thinking how to plan this project for next semester.

Meeting Records

3/11/21	Machine Learning: - Find a subject (examples: games, finance...) - C++ library to analyse codes and tweak with parameters - The most important is to have a good report - Find a dataset that has already been worked with or find one that's like be able to compare our work. - Search datasets (google scholar, Kaggle...)
10/11/21	- predictive data (nonlinearly separable) - maybe games past behaviours - sustainable energy sources etc - climate change Is the data acceptable
17/11/21	- diabetes subject was approved - start writing drafts - start filling the ethics
7/04/22	- helping on the dissertation structure, table of contents - what is important to go in the literature review
14/04/22	- more advice on literature review - more advice on research methodology
21/04/22	- project management content + conclusion content

Git Hub Repository link

https://github.coventry.ac.uk/chichkos/6001CEM_ML_Diabetes_PimaIndians