Bachelor's Thesis Computing Science

Radboud University Nijmegen

Utilizing machine learning and SHAP to uncover key variables for a healthier lifestyle in Type 2 Diabetes management

---

Author:
Hugo Hakkenberg
s1027629

First supervisor/assessor:
dr. Ilona Wilmont

Second assessor:
dr. Gabriel Bucur

January 9, 2024

Abstract

Type 2 diabetes(T2D) is the most prevalent form of diabetes, with 1.1 million individuals currently diagnosed in the Netherlands. The risk of T2D significantly increases with an unhealthy lifestyle. This research, conducted in collaboration with "Je Leefstijl Als Medicijn"(JLAM), aims to uncover key variables for a healthier lifestyle in T2D management. Utilizing a unique data set provided by JLAM, featuring variables like HbA1c, waist-to-height-ratio(WtHr), BMI and variables indicating individuals activity on JLAM's website. Our research has been conducted in multiple steps. Firstly, we visualized the average weight, WtHr, HbA1c and glucose difference over a 30-month period, using measurements from JLAM members. Next, we observed a positive correlation between increased online activity and improved measurement results, emphasizing the benefits of online health communities like JLAM's website. Finally, employing six machine learning classification algorithms, we achieved accuracy scores ranging from 72.9% to 83.6% for T2D prediction. Utilizing Shapley Additive exPlanations(SHAP), we uncovered the most impactful variables contributing to the predictions, which were BMI and WtHr. Further research should include an expansion of the number of members and variables in the data set.

Contents

Chapter 1

Introduction

In the Netherlands, diabetes poses a massive public health challenge, affecting 1.1 million individuals, with this number increasing by 52.000 each year. Currently, nine out of ten individuals diagnosed with diabetes have type 2 diabetes (T2D). It is even expected that 1 out of 3 Dutch adults above 45 (2.8 million individuals) will get T2D in the future[9].

High blood sugar levels causes T2D, this occurs either when your body is not able to make enough insulin or due to the insulin made not working properly. Anybody can develop T2D; however, an unhealthy lifestyle significantly increases the chance of T2D. If T2D is not treated accordingly, significant physical complications are likely to occur, e.g., damage to feet, eyes, kidneys and cardiovascular diseases[44][9].

The foundation "Je Leefstijl Als Medicijn"(JLAM) encourages people to become healthier through lifestyle changes and therefore prevent diseases like T2D. With their "Saturday Weight and Measure" activity, they aim to have members measure their body weight, waist circumference, fasting glucose levels, and when possible, HbA1c levels[1] every Saturday[27]. The product of this activity is a large unexplored data set containing several hundred individuals who regularly measure these variables.

Nowadays, AI and especially machine learning classification algorithms are used more and more in the healthcare sector. They are well-suited for disease diagnosis, where the goal is to classify patients into classes. Machine learning classification algorithms can detect patterns in large amounts of data, which would be difficult for humans to find, and make classification predictions according to these patterns. Developing these algorithms could potentially lead to quicker and more accurate diagnoses while also improving the efficiency and decreasing the costs of healthcare. Previous research

[1] HbA1c is a measure of the average blood glucose levels of the past two/three months

has shown positive results when employing machine learning algorithms for T2D diagnoses, with accuracy scores ranging from 70% up to 85%[31][42]. However, little attention has been paid to uncovering the key variables responsible for T2D diagnoses. Therefore, to get a better understanding of the predictions made by machine learning algorithms, interpretability techniques like SHapley Additive exPlanations (SHAP) are used. For each variable in the data set, SHAP determines a Shapley value[2], which indicates the impact of that variable on the prediction. This allows us to uncover the key variables from a predictive classification made by a machine learning algorithm.

When uncovered, these key variables will be vital in T2D management. They give clarity and allow individuals to monitor their risk of developing T2D by keeping track of them. Therefore, in collaboration with JLAM, this research investigates how predictive machine learning algorithms and SHAP, can be employed to uncover key variables responsible for T2D diagnoses. Furthermore, we will analyze the measurements and possible links between measurements and members' activity on the website of JLAM.

This thesis is structured in the following manner:

- Chapter 2 gives background information about JLAM and explains technical concepts such as machine learning algorithms, data preprocessing, evaluation metrics and more.
- Chapter 3 discusses similar research done which we built upon.
- Chapter 4 explains the data set we used, how we retrieved the data, preprocessing of the data and feature selection to complete our data set.
- Chapter 5 explains how we visualized the measurements done by the members of JLAM, explains our method of finding links between online activity and measurements, and shows how we employed the machine learning algorithms and SHAP.
- Chapter 6 displays the results obtained from the data visualisation, online activity and machine learning and SHAP.
- Chapter 7 discusses the results and compares the results to previous research done.
- Chapter 8 presents the conclusions that can be drawn from the results obtained in our research.

[2]Shapley values assigns a fair contribution of each variable to the machine learning algorithm's prediction by considering all possible combinations of variables to ensure a comprehensive understanding of their impact on the predictive outcome

Chapter 2

Preliminaries

2.1 Je Leefstijl Als Medicijn

The foundation JLAM promotes improvement of health by living a healthier lifestyle[26]. They focus on important aspects of a healthy lifestyle like nutrition, exercise, mental health, relaxation and sleep. During our research, we have been working with their chairman Wim Tilburgs, who is someone who struggled with being overweight and was diagnosed with T2D. Medication and insulin injections were not giving the desired results. Therefore he went on a mission to gain knowledge about changing his lifestyle. After changing his diet he lost a lot of weight and was able to decrease the amount of medication needed.

JLAM has multiple support groups, one such group is called 'Diabetes2doorbreken'. This support group is a Facebook community that aims to inspire, motivate and spread knowledge about T2D. Furthermore, 'Diabetes2doorbreken' offers further guidance with its 'Zaterdag Wegen en Meten'(ZWeM) translated to 'Saturday Weighing and Measuring' activity[27]. Ideally, every Saturday, members of 'Diabetes2doorbreken' weigh themselves and measure their waist circumference and fasting glucose levels. When possible, members are also able to fill in their HbA1c levels. However, HbA1c can only be measured by a GP.

Keeping track of all of these values is important for T2D management. As can be seen in Table 2.1, these variables have normal ranges, increased diabetes risk ranges and high diabetes risk ranges. In this research, we will be focusing on normal ranges and high risk ranges. These values are based on research done by JLAM[27] and the World Health Organization[45][37]. Fasting glucose levels and BMI(created using weight and height) are reliable indicators used in basically every T2D prediction algorithm. These two variables are included in the Pima Indians Diabetes Database[25], which is an

Table 2.1: Variable Ranges Indicating Type 2 Diabetes

Variable	Normal Range	Increased Risk Range	High Risk Range
BMI	18.5-25	25-30	>30
WtHr (cm, Male/Female)	<0.53 / <0.49	0.53-0.57 / 0.49-0.53	>0.57 / >0.53
Glucose (mmol/L)	<5.6	5.6-6.9	>7
HbA1c (mmol/mol)	<42	42-53	>53

open source database that is used in most diabetes classification researches[42][10][39][20]. Waist to Height ratio(WtHr) has been found to be a strong indicator for T2D, with even more advantages than waist circumference[46]. Furthermore, previous research shows that using HbA1c as a variable increases the performance of machine learning algorithms for T2D classification significantly[21].

2.2 Machine Learning

Machine learning is a part of artificial intelligence that builds algorithms and models that enable computers to learn from data and make decisions based on that data. Machine learning classification algorithms are used to make predictive classification using techniques to learn patterns and relationships within the data. Firstly, machine learning algorithms get fed training data from which they learn the patterns of the data. Secondly, once the algorithm has been trained, it can be used to make predictions or decisions on unseen test data. Based on the results on the test data, the performance of the algorithm can be measured[29].

2.2.1 Supervised and Unsupervised

There are multiple ways an algorithm can learn from data. The first is by supervised learning, where the algorithm is trained on a labeled data set. This entails that the input data is paired with corresponding output labels. This way the algorithm learns from examples. The second way is by unsupervised learning, where the algorithm is given unlabeled data and is tasked with finding patterns or relationships within the data on its own. In this case, there are no predefined output labels provided during the training phase. In conclusion, supervised learning is able to make predictions or classifications by learning from examples, whereas unsupervised learning makes predictions by exploring patterns and structures in unlabeled data[8].

2.2.2 Classification and Regression

There are two types of supervised learning algorithms in machine learning. Firstly, we have classification algorithms, where to goal is to predict the correct categorical label of the given input data. Secondly, we have regression algorithms, which aims to predict a continuous numerical value based on input data[8].

2.2.3 Logistic regression

Logistic regression is a supervised machine learning algorithm for classification. The target variable is binary (0 or 1). The goal is to predict the target variable using a given data set of independent variables. This is achieved by transforming the linear regression[1] function continuous value output into a binary value output by applying a sigmoid function[2]. After applying the sigmoid function, we end up with a value between 0 and 1, resulting in a classification[14].

2.2.4 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression tasks. SVM can efficiently perform non-linear classification by implicitly mapping the input data into high-dimensional variable spaces. The main goal of classification is to find the optimal hyperplane that best separates the data into classes. The hyperplane found with SVM is an (n-1)-dimensional flat subspace in an n-dimensional space, e.g. in a 2D space, the SVM hyperplane is a line. The points closest to the hyperplane from both the classes are called support vectors. Computing the distance between the hyperplane and support vectors is called the margin. The optimal hyperplane is where the margin reaches the maximum[41].

2.2.5 Decision Tree

Decision tree is a supervised machine learning algorithm that can be used for classification and regression. It builds models in the form of a tree structure, starting at the root node (entire data set) and ending up at the leaf nodes. Each internal decision node represents a variable of the data set, the branches between the nodes represent the decision rules and the leaf nodes represent the outcome[22].

[1]Linear regression is a supervised machine learning algorithm that assumes a linear relationship between the input variables and the output variable, and aims to find the best-fitting straight line through the data

[2]Sigmoid function maps input values to a value between 0 and 1, which is used for a binary classification

Figure 2.1: Optimal Hyperplane using SVM

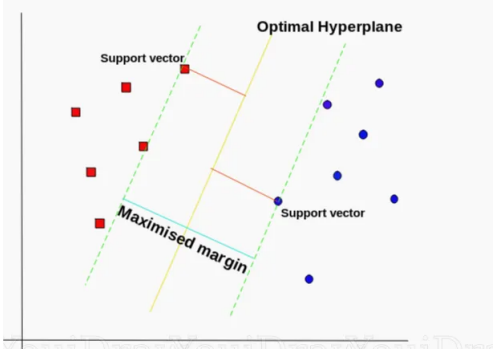
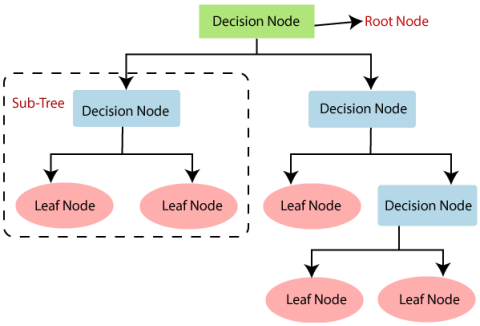


Figure 2.2: Decision Tree Structure



2.2.6 Naive Bayes

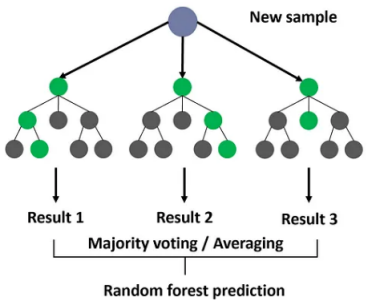
Naive Bayes algorithm is a supervised machine learning algorithm used for classification tasks. It is called Naive since the algorithm assumes that the variables in the data set are independent of each other. Bayes comes from the principle of Bayes' theorem, which Naive Bayes is based on[23].

2.2.7 Random Forest

Random forest is a supervised machine learning algorithm that is used for classification and regression. Random forest performs classification tasks using multiple decision trees combined with bagging[3]. The idea is to combine multiple decision trees to get a more accurate output prediction[15].

[3]Bagging is an ensemble technique that trains multiple instances of the random forest algorithm on random subsets of the training data, then combines their individual predictions to improve overall performance and accuracy of the algorithm

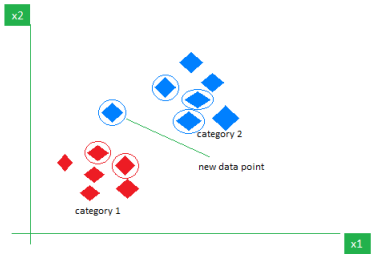
Figure 2.3: Random Forest Structure



2.2.8 K-Nearest Neighbor

K-Nearest Neighbor (KNN) is a supervised machine learning algorithm used for both classification and regression tasks. KNN works on the principles of similarity. It predicts the value of new data points by considering the values of its K nearest neighbors in the training data. KNN calculates the distance (often using Euclidean distance[4]) between each new test data point and all training data points. After finding the K nearest neighbors of the test data point, the algorithm makes a prediction based on the values of these neighbors[13].

Figure 2.4: Classification of new data point using KNN



2.3 Data Gathering

In order to analyze data using machine learning algorithms, we first need to gather the data. This is where the data sources of JLAM come into play. JLAM has two main data sources: Hubspot and Microsoft Azure SQL Database. Data about the measurements is stored in Microsoft Azure SQL Database, while online activity variables are stored in Hubspot.
"Hubspot is a customer platform with all the software, integration's, and resources you need to connect your marketing, sales, content management,

[4]Euclidean distance is the length of a line segment between two data points

and customer service."[17] In Hubspot you are able to create lists e.g. JLAM has a list containing members of the diabetes group. Using the Hubspot API, you are able to extract data from such lists.
"Microsoft's Azure SQL Database is an always up-to-date, fully managed relational database service built for the cloud."[33] When connected to the server, you are able to use SQL queries to extract data from a specific database.

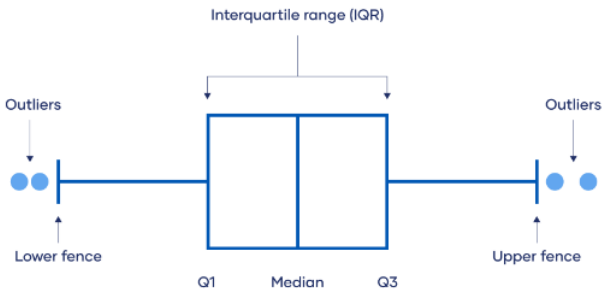
2.4 Data Preprocessing

Data preprocessing is a vital step in data analysis and machine learning. It involves transforming and cleaning raw data into data that can effectively, accurately and efficiently be utilized for algorithm training. Data preprocessing deals with outliers, missing, incorrect or duplicate values and inconsistencies[30].

2.4.1 Data Cleaning

Data cleaning is the process of removing or adjusting incorrect, missing or duplicate data within a data set. Data cleaning also involves dealing with outliers. Incorrect data, missing data and outliers can result in unreliable predictions. Therefore, it is important that this process is performed effectively. Identifying outliers can be done using the interquartile range (IQR)[6]. The IQR tells you the range of the middle half of your data set (figure 2.5). This is a helpful method to identify values on the extreme ends of the data set. The method works as follows: sort the data set from low to high; identify the first quartile[5] (Q1), the median, and the third quartile[6] (Q3); calculate IQR = Q3 - Q1; calculate the upper fence: Q3 + (1.5*IQR) and lower fence: Q1 - (1.5*IQR); all values outside the fences are outliers.

Figure 2.5: IQR



[5]The number halfway between the minimum and median number
[6]The number halfway between the median and maximum number

2.4.2 Data Transformation

Data transformation is the process of converting data from one format to another. It can be seen as mapping one data form into another. There are multiple methods of data transformation:

1. Normalization/scaling: scaling variables to ensure that variables are on a comparable scale. Scaling can be done using the StandardScaler function from Python's scikit-learn library.

2. Feature Engineering: process of deciding which variables are most important to use to train the machine learning algorithms. This can also mean combining two existing variables into a new variables e.g. length and weight into BMI.

2.5 Feature Selection

Feature selection is the process of selecting the most important, consistent and relevant variables (features) to use in a machine learning algorithm, resulting in the removal of redundant or irrelevant variables. Often, having too many variables selected results in a longer training process and sometimes less accurate results. This can happen because some variable characteristics may overlap or be less present in the data. Therefore, the main goal of variable selection is to reduce computational costs and improve the performance of a predictive machine learning algorithm[16]. One component of variable selection is feature engineering, explained in Section 2.4.2.

2.6 Hyperparameter Tuning

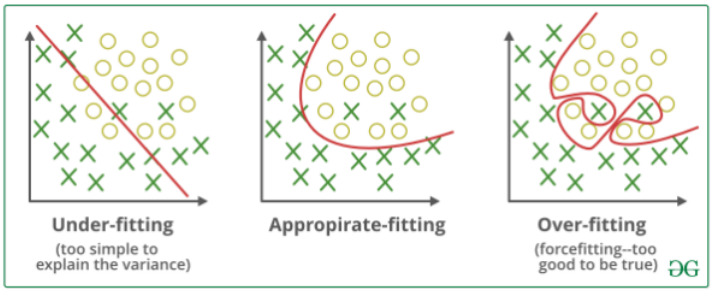
Hyperparameters are parameters that have a huge impact on the learning process of a machine learning algorithm. Hyperparameters can not be learned from data but need to be set prior to training. Hyperparameter tuning involves finding the best combination of hyperparameter values to optimize a machine learning algorithm's performance[36]. An example of a hyperparameter is K in the K-Nearest Neighbor algorithm. Choosing the right value for K will lead to significantly better performance.

2.7 Underfitting and Overfitting

Underfitting and overfitting are often responsible for poor performance of machine learning algorithms. Underfitting happens when an algorithm does not have enough variables and therefore is too simple to find data complexities. The algorithm is unable to learn the training data effectively resulting in poor performance in both training and test data. Underfitting can be

reduced by using more relevant variables that represent underlying patterns in the data, as well as, increasing the size of the data set.
Overfitting happens when an algorithm is too complex relative to the size of the data set. The algorithm begins to learn from noise and outliers, resulting in poor performance in test data. Overfitting can be prevented using regularization[7], feature selection and feature engineering to reduce the number of redundant variables.

Figure 2.6: Underfitting and Overfitting



2.8 Cross Validation

Cross validation is a method used in machine learning to evaluate the performance of an algorithm on data points. One form of cross validation is K-Fold cross validation, where the data set gets split into K number of subsets. One subset is used as a validation set, the remaining subsets are used to train the algorithm. This process is then repeated K times with a different validation set each time. Then the results of each validation step are taken and averaged to give a better estimate of the algorithm's performance[12].

2.9 Evaluation Metrics

Evaluation metrics are measures used to assess the performance of machine learning algorithms on a specific task. They provide insight into the performance of the algorithm and can help compare performance of different algorithms. Furthermore, they can be used to fine tune the algorithms to get better results. A confusion matrix is an example of an evaluation metric[2]. As can be seen in Figure 2.8, a confusion matrix is a table with combinations of predicted and actual values. Positive and Negative refer to the two classes a data point can be classified as. In our case, Positive refers to low risk diabetes and Negative refers to high risk diabetes. A confusion matrix is useful for calculating different evaluation metrics:

[7]Regularization is a set of techniques that pushes the algorithm to reduce its complexity as it is being trained, this helps prevent overfitting

1. Accuracy measures how often the algorithms correctly classify a data point.
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Precision shows how many positively predicted cases actually turned out to be positive.
$$Precision = \frac{TP}{TP + FP}$$

3. Recall shows how many of the actual positive cases were predicted positively by the algorithm.
$$Recall = \frac{TP}{TP + FN}$$

4. F1 score combines precision and recall.
$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

5. A Receiver Operator Characteristic(ROC) curve displays a machine learning algorithm's performance across different threshold[8] settings. ROC plots the True Positive Rate (sensitivity)[9] against the False Positive Rate[10]. Using the ROC plot, we can calculate its Area Under the Curve (AUC). The AUC value indicates how well an machine learning algorithm performed. An AUC value close to 1 indicates that the model is performing well, an score of 0.5 means the algorithm performs like a random classifier.

Another evaluation metric is a classification report, which is a useful table format that displays three other evaluation metrics: precision, recall and F1-score. These three evaluation metrics have a score between 0 and 1, where a higher score indicates a better performing algorithm. Support indicates the actual amount of instances of a specific class, e.g., in Figure 2.7 class 0 has 37584 instances and class 1 has 37577 instances. For each evaluation metric, there is a macro average and weighted average. Macro average returns the average of the results of the classes added up together, e.g., for precision the macro average score $= \frac{0.77+0.84}{2} = 0.81$. The weighted average score also return the average of the classes, however, it takes into account the number of instances of each class. Again if we take precision, the weighted average $= \frac{(0.77*37584)+(0.84*37577)}{75161} = 0.80$.

[8]Threshold is used to classify the predicted probabilities into different classes, e.g., if the predicted probability is above the threshold, it is classified as the Positive class

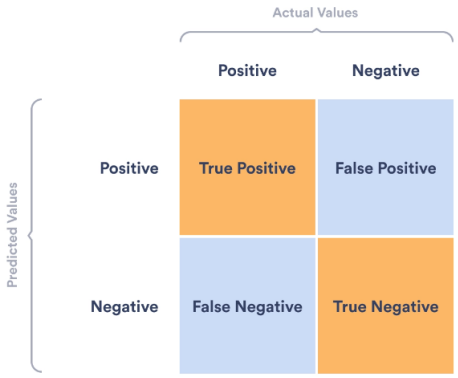
[9]TPR is used to measure the percentage of actual positive cases which are correctly classified by the algorithm

[10]FPR is used to measure the percentage of actual negative instances incorrectly predicted as positive by the algorithm

Figure 2.7: Classification Report

	precision	recall	f1-score	support
0	0.77	0.86	0.81	37584
1	0.84	0.75	0.79	37577
accuracy			0.80	75161
macro avg	0.81	0.80	0.80	75161
weighted avg	0.81	0.80	0.80	75161

Figure 2.8: Confusion Matrix



2.10 Interpretability

Interpretability of a machine learning algorithm refers to the ability to understand and explain the prediction made by an algorithm. It involves making machine learning algorithms more understandable to the users of the algorithms. One interpretability technique is Shapley Additive exPlanations (SHAP). SHAP is a method based on cooperative game theory principles that is used to explain predictions made by machine learning algorithms. "The goal of SHAP is to explain the prediction of an instance x by computing the contribution of each variable to the prediction[34]". Computing the contribution of each variable is done with Shapley values. Shapley values are determined by computing the average marginal contribution of a variable across all possible combinations of variable subsets.

Chapter 3

Related Work

In recent years, machine learning has become popular and widely used across multiple different industries. Machine learning has the great ability to recognize patterns in data sets to find solutions to a certain problem. Think of companies like Spotify, which uses machine learning algorithms to recommend music based on songs you have previously listened to. Supervised machine learning algorithms(Chapter 2.2.1) are used as a predictive tool in many different sectors. With the huge amount of data in the healthcare sector, utilizing machine learning algorithms could significantly help the healthcare sector. The predictions of the algorithms can be used to help doctors make more accurate decisions, improve outcomes for the patients and reduce costs. Already, predictive machine learning algorithms are used to help detect diseases or high risk patients for diseases like skin cancer, covid-19 and T2D[4].

3.1 PIMA Indians Diabetes Database

Previous research has shown that predictive T2D algorithms have been limited by the amount of open source data sets. Almost all scientific papers that research the use of machine learning algorithms to predict T2D, that we have found, use the PIMA Indians Diabetes Database[42][10][31]. The PIMA database is an open source database from the National Institute of Diabetes and Digestive and Kidney Diseases[25]. It contains a total of 768 women, where the division positively/negatively tested for T2D is 268/500. It uses the following variables to predict T2D: amount of pregnancies, blood pressure, skin thickness, insulin, BMI, age, pedigree diabetes function.

The PIMA database is one of the only relatively large open source T2D database, therefore, it is used in most research about T2D. However, the database only contains women and is limited in terms of its variables. The database does not include HbA1c, while HbA1c has been recommended

by an International Committee and by the ADA as a means to diagnose diabetes[38]. Furthermore, research done by J. Hou, Y. Sang, Y. Liu and L. Lu, who used different variables from the PIMA database, showed that including HbA1c as a variable results in better performance of the machine learning algorithms for T2D diagnoses[21]. With their research, they proved the importance of HbA1c in T2D prediction algorithms. Another important variable used for T2D classification is waist-circumference or WtHr. Especially WtHr is an important indicator for T2D and has even been called as a better indicator compared to BMI, waist circumference and waist-hip ratio[47].
Therefore, only using the PIMA database limits the full potential of predictive machine learning algorithms for T2D diagnoses. Having a database including males, females and essential variables such as HbA1c and WtHr will be highly beneficial for T2D diagnoses.

3.2 Predictive Machine learning Algorithms for T2D Diagnoses

Using machine learning algorithms to predict T2D has been done multiple times in the past. Machine learning algorithms such as, SVM, LR, KNN, NB, RF and DT have all been employed to predict T2D. The accuracy scores obtained from these papers vary from 74% up to 87%[42][31]. In several papers, including a review paper about the diagnosis of diabetes by machine learning algorithms, SVM has been found to be the best performing algorithm[1]. Therefore, we will be using these six machine learning algorithms for our research, however, we will also use SHAP to uncover the most impactful variables to the prediction.
As explained in Chapter 2.10, SHAP is an interpretability tool used to explain the predictions made by machine learning algorithms. SHAP has been used in all types of machine learning researches, e.g., to find the most impactful variables leading to suicide attempts[35]. Furthermore, in a research done by I.Tasin, T.U.Nabil, S.Islam and R.Khan, the PIMA database alongside interpretability tools LIME and SHAP were used to understand how the algorithms predict their final results. They applied nine machine learning algorithms to predict T2D and concluded accuracy scores between 75% and 81%. Furthermore, they found that glucose, BMI and age were the most impactful variables, whereas blood pressure and insulin were the least impactful variables resulting to T2D classification[20].

3.3 Our Contribution

In this research, we will be using machine learning algorithms to predict high risk T2D and SHAP to determine the most impactful variables contributing to that prediction. We will not be using the PIMA database which is used in most other similar researches that we found. We will be using the data set fetched from the data sources of JLAM(Chapter 2.1). This data set includes both males and females and uses some different variables than the ones used in PIMA, e.g., HbA1c and WtHr. Furthermore, the data set also includes variables indicating the 'online activity' of the members of JLAM. Online activity includes a member's activity on the website of JLAM where useful information about a better lifestyle and more is posted. Finding out how online activity correlates to measurement results will be useful information for the members to know.

Chapter 4

Data

4.1 Data Sources

The data set we have used in this research has been acquired from the foundation JLAM (Chapter 2.1). As mentioned in Chapter 2.3, their data is spread across two main data sources: Microsoft Azure SQL Database and Hubspot. The data from Microsoft Azure is especially interesting since it includes thousands of measurements done by the members of JLAM. The data from Hubspot is useful since it contains statistics about someone's activity on the website of JLAM.
In order to retrieve that data, we first had to extract the data from the data sources. Extracting data from Microsoft Azure SQL Database is relatively straight forward. Connecting to their server and using SQL statements, enabled us to extract data from different tables. The available data includes all measurements members have done over the years. These measurements can be related to: weight(kg), waist-circumference(cm), fasting glucose(mmol/L) and HbA1c(mmol/mol). For each measurement the following data can be extracted:

- Measurement type(mtype): variable that has been measured represented by a number, e.g., 1 indicates weight.
- Measured value(mvalue): actual value of the variable that has been measured, e.g., 94kg.
- Measurement date(mdate): date of the measurement.
- Birth date: specific date the member was born, e.g., 01-01-1975.
- Gender: 'M' for males and 'F' for females.
- Length: height of the member in centimeters.

Table 4.1: Example of Extracted Data from Microsoft Azure Database

first name	last name	gender	birth-date	mtype	mvalue	mdate	length
Jan	Janssen	M	01-01-1975	1	100	10-10-2020	180
Ana	Nas	V	06-06-1990	4	50	1-1-2022	167
...	...	...	...	...	...	...	...
Willem	Willemsen	M	02-02-1960	2	110	20-10-2023	185

Table 4.2: Example of Extracted Data from Hubspot

first name	last name	num-visits	num-views
Jan	Janssen	40	600
...	...	...	...
Willem	Willemsen	12	120

In total there are 22610 measurement done by 234 members, an example of a measurement can be seen in Table 4.1.

Extracting data from Hubspot is done by creating a private Hubspot app with a password token[19]. Using the Hubspot API we were able to send GET requests and retrieve data from specific lists. The foundation has a list called 'Diabetesgroup' in Hubspot which has 284 members. This is a list of people who actively want to improve their health by changing their lifestyle. After extraction of the data, we ended up with Table 4.2. Num-visits and num-views are interesting variables which show online activity of a member on the foundation's website. On this website, there are multiple health specialists who post tips about improving your health and decreasing risk of developing T2D. These variables indicating online activity will be used to research if there is a link between online activity and reducing the risk of T2D.

After collecting the data from Microsoft Azure and Hubspot, we merged them into a single DataFrame[1] (Table 4.3) using an inner-join[2] based on the first and last name variables.

4.2 Data Preprocessing

Before we are able to use the data set, we will first have to preprocess our data. This includes the following processes: data cleaning and data

[1] A dataframe is a 2D labeled data structure similar to SQL table

[2] An inner-join returns all records from both dataframes based on one or more related columns between them

Table 4.3: Example of Merged DataFrame

first name	last name	gender	birth-date	mtype	mvalue	mdate	length	num-visits	num-views
Jan	Janssen	M	01-01-1975	1	100	10-10-2020	180	40	600
Ana	Nas	V	06-06-1990	4	50	1-1-2022	167	null	null
...	...	...	...	...	...	...	...	...	...
Willem	Willemsen	M	02-02-1960	2	110	20-10-2023	185	12	120

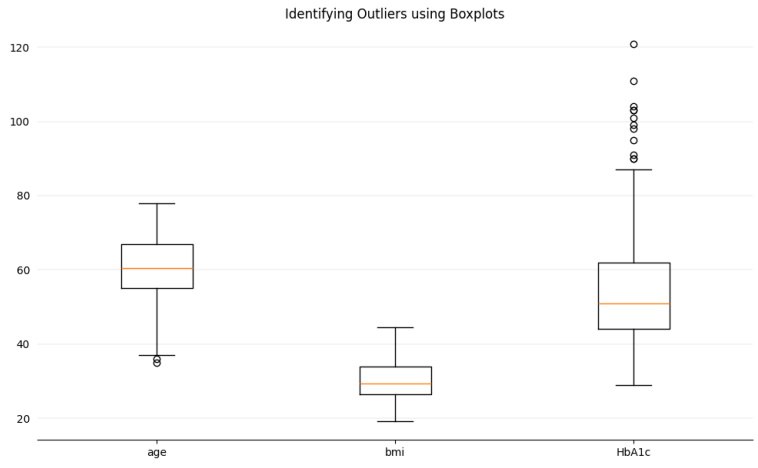
transformation(Chapter 2.4).

4.2.1 Data Cleaning

Data cleaning involves removing outliers, duplicates, missing(null) or incorrect data. Since we used an inner join to merge the DataFrames into a single DataFrame we know that there will be no inserted null values. Furthermore, we checked and removed any missing, incorrect or duplicate values using python functions.

To identify outliers we used the IQR method (Chapter 2.4.1) along with boxplots for visualisation. We used IQR on all relevant variables and had the following results: no outliers found for BMI and WtHr; outliers found for age, glucose, HbA1c, num-visits, num-views. Figure 4.1 shows a boxplot containing the variables age, BMI and HbA1c, where the circles outside the fences are outliers.

Figure 4.1: IQR Boxplot for Age, BMI and HbA1c



Now that we have identified the outliers we have two options: retaining or removing. Removing outliers in case of unreasonable or extreme values can lead to a cleaner data set, however, it can also lead to a biased data set and inaccurate conclusions. In most cases, retaining outliers as much as possible is better unless it is certain that they represent incorrect data. Considering that we have to work with a relatively smaller data set, which does not contain incorrect data, we decided not to remove any extreme values such as an HbA1c value of 120 mmol/mol or a BMI of 38. This way we would not create a biased data set, since these extreme values are real values measured by members of JLAM.

4.2.2 Data Transformation

As explained in Chapter 2.4.2, data transformation is used to transform data from one format to another. This can be done with feature engineering and normalization. Feature engineering is the process of creating new variables from existing ones. We merged the following existing variables(Table 4.3) into new variables:

1. Birth date transformed into age
2. Weight and length transformed into BMI
3. Waist circumference and length transformed into waist to height ratio (WtHr)

Variable scaling/normalization can significantly improve our algorithms performance, especially for machine learning algorithms that are sensitive to the scale of the input variables (SVM, KNN and more). Scaling transforms the variables so that they are all on a similar scale. This is done to ensure that no single variable dominates the training process of the algorithm simply because its values are larger. Since we have variables with different ranges, e.g. WtHr (mostly smaller than 1) and num-views (ranging from 0 to hundreds), it is essential that we scale our variables.

4.3 Feature Selection

Choosing the right amount and most important variables to train our machine learning algorithms with is a vital step for our research. The first five selected variables are:

1. HbA1c: hemoglobin A1c is similar to fasting glucose, as it is related to the measurement of blood glucose. However, HbA1c is the average blood glucose (sugar) levels of the last couple of months, while fasting glucose is the concentration of glucose in the blood after an overnight fast. Therefore, HbA1c provides an indication of long-term glucose

control. HbA1c is difficult to measure since it has to be measured by a GP, therefore HbA1c is barely used in other T2D research. However, as stated before, when used it has been proven to increase performance of T2D classification algorithms[21].

2. Waist-to-height ratio: calculated by dividing waist circumference by height. In previous research, BMI, waist circumference and waist-hip ratio have been compared to waist to height ratio as indicators for T2D. The research concluded that while BMI, waist circumference and waist-hip ratio are good indicators, waist to height ratio is a slightly better indicator for T2D diagnosis[47].

3. Age: the number of T2D diagnoses often increases as age increases. Middle aged individuals (50-70) are more likely to develop T2D than younger individuals. Research has also shown that the prevalence of diabetes and prediabetes is even higher for the elderly individuals(70+) than the middle aged individuals[48]. Therefore, age is an important variable to select.

4. Fasting glucose levels: amount of glucose in your blood when you have not eaten for 8-12 hours. It provides the blood glucose levels at a specific point in time and is often used to asses short-term glucose control. Testing fasting glucose levels is a common way to diagnose diabetes[37].

5. Body mass index (BMI): weight divided by the square of length. Having a high BMI indicates high body fatness. According to research, there is a positive association between BMI and the risk of T2D[32].

These five variables are indicators for T2D and are therefore selected. They have normal value ranges, ranges that increase the risk of T2D and ranges that highly increase the risk of T2D[27][37]. These ranges can be seen in Table 4.4.

The last two variables selected indicate a member's online activity. The website of the foundation JLAM contains a lot of useful and relevant information about T2D and more. Information like past experiences, support groups, lifestyle and diet benefits are all shared on the website. With the use of the following two variables we will research possible links between online activity and measurement results:

6. Num-visits: number of times an individual has visited JLAM's website.

7. Num-views: number of pages an individual has viewed on JLAM's website.

Now that we have selected seven variables, the data set is complete and an example is displayed in Table 4.5.

Table 4.4: Variable Statistics

Variable	no/low risk	risk at diabetes	high risk at diabetes	Average (data set)
BMI	18.5-25	25-30	>30	30
WtHr (in cm, Male/Female)	<0.53 / <0.49	0.53-0.57 / 0.49-0.53	>0.57 / >0.53	0.6
Glucose (in mmol/L)	<5.6	5.6-6.9	>7	8
HbA1c (in mmol/mol)	<42	42-53	>53	56
Age (in years)	18-44	45-70	>70	60
Num-visits	-	-	-	36
Num-views	-	-	-	88

Table 4.5: Example of Final DataFrame

age	BMI	WtHr	glucose	HbA1c	num-visits	num-views	classification
60	33.0	0.58	7.5	95	180	40	1
35	26.2	0.53	11.3	71	28	89	0
...	...	...	...	...	...	...	...
58	35.5	0.65	16.4	70	1	10	1

4.3.1 Diabetes Classification

Now that we have selected the most relevant and important variables, we have almost finalized our data set. The only variable we need to add is the target variable called 'classification'. Since we do not have a variable indicating a T2D diagnosis, we have to make that classification ourselves. Classification can be either 0 or 1, where 0 indicates a low risk of T2D and 1 indicates a high risk at T2D. The classification is done based on a points system displayed in Table 4.6. This point system is similar to the following tests: 'Diabetes-test' by Diabetes Fond[11] and 'Type 2 Diabetes Know Your Risk' by Diabetes UK[43]. In those tests around eight different questions about e.g., age, BMI and waist circumference are asked, which eventually determine the risk of T2D. We modified their test slighty by removing variables such as ethnicity, medicine used or diagnosed relatives and adding glucose and HbA1c. For each variable (except gender) in Table 4.6, there is a normal range (0 points), low risk range (4 points), increased risk range (6 points) and high risk range(9 points). If the total sum of points is more than 30, the classification variable is 1, otherwise the classification variables is 0.
After classifying the data instances using Table 4.6, we end up with a data

Table 4.6: Diabetes Risk Score

Variable	Value	Points
Gender	Female	0
	Male	1
Age (years)	< 49	0
	50-59	4
	60-69	6
	> 70	9
Waist (cm)	< 90	0
	90-99.9	4
	100-109.9	6
	> 110	9
BMI	< 25	0
	25-29.9	4
	30-34.9	6
	> 35	9
Glucose (mmol/L)	< 5.6	0
	5.6-6.1	4
	6.1-6.9	6
	> 7	9
HbA1c (mmol/mol)	< 42	0
	42-49	4
	49-53	6
	> 53	9

set containing 85 positive classifications and 69 negative classifications. This means the data set is split 55%/45%, and is therefore nicely balanced. We can state that there is no undersampling or oversampling[3] needed for our data set.

[3]Undersampling and oversampling are techniques to adjust the class distribution of a data set. These techniques should be employed if the data set is unbalanced.

Chapter 5

Research

Our research consists of the following three parts: measurements visualisation, online activity and machine learning. In the first part, we explain how we analyze all of the measurements done by members of JLAM. In the second part, we try to discover links between the members' online activity and measurement outcomes. In the last part, we explain how we employ predictive machine learning algorithms to uncover the key variables for T2D diagnoses.

5.1 Measurement Visualisation

The data set for the first part of our research contains 22610 measurements from 234 unique members of JLAM. From these 22610 measurements, 7717 are related to body weight, 7387 to waist circumference, 6808 to glucose levels and 698 to HbA1c levels. These measurements started in 2015 and still continue to this day. Since we are dealing with thousands of different measurements done by hundreds of different individuals, we want to be able to visualize these measurements. Therefore, we extracted the measurements using SQL and created a python program to visualize the data. Since we want to visualize the average trend of each variable, the python program first splits the measurements into four different data sets. One data set for each of the following variables: body weight, WtHr, glucose and HbA1c.

For each person in these data sets, we take the first measurement(using the measurement date) as a starting point, after that, we average the measurements done one month since the first measurement up-to 30 months since the first measurement. For example, if Jan Janssen has a first body weight measurement of 80kg and the following month he measures his body weight three times: 79.5, 79 and 78, we take the average of these three measurements as Jan's difference for month 1, which is: $\frac{79.5+79+78}{3} - 80 = -1.17$. If someone has not done any measurements in a specific month, that person is

not included in the average value difference for that month.

Since most members have not been measuring for more than 30 months, we decided to analyze the data from the start of measuring until 30 months into the process. Finally, for each data set, we took the average value difference of all members over the course of 30 months and displayed it using a graph. This will visualize the thousands of measurements done and allow us to get an understanding of a possible value difference trend.

5.2 Online Activity

In the second part of our research, we try to find links between members' online activity and their measurement outcomes. As explained in Chapter 4.3, to indicate a member's online activity we have the following two variables: num-views and num-visits. We are going to research if there are links between positive measurement results and a high level of online activity in the following way:

1. Collect the measurements from the Microsoft Azure Database, collect the online activity variables from Hubspot and merge them into a single python dataframe.
2. Calculate the value difference (BMI, WtHr, glucose and HbA1c) for each individual, from the first measurement and last measurement.
3. Use a bar chart to display the relation between online activity and measurement outcomes.

5.3 Machine Learning Algorithms

During our research, we employed the following six supervised machine learning classification algorithms to predict the risk of developing T2D: Logistic Regression, Support Vector Machine, Decision Tree, Naive Bayes, K-Nearest Neighbor, Random Forest (Chapter 2.2.3 - Chapter 2.2.8). We decided to use these six algorithms since previous research has shown that they are the most used algorithms for T2D predictions[1]. Alongside these algorithms we used SHAP to uncover the most impactful variables to the prediction made by the algorithms.

5.3.1 Optimization

Train-Test Split

Before training and testing the six algorithms mentioned above, we first have to split our data set into train and test data. Usually, the data set gets split 70/30, 80/20 or 90/10. This depends on the size of the data set and the complexity of the algorithm. For smaller data sets is it often better to have more train data(90% or 80%) and less test data(10% or 20%) to ensure that the algorithm has enough data to learn from[3]. For larger data sets a split of 70/30 is often used to get more accurate test results. Since we have a relatively small data set we decided to use a split of 80%/20%. Splitting the data can be done using scikit-learn's(sklearn)[1] 'train_test_split' function.

Underfitting and Overfitting

To prevent the occurrence of underfitting and overfitting(Chapter 2.6), we used feature engineering(Chapter 2.4.2) and feature selection(Chapter 2.5). This resulted in the removal of redundant variables, while all relevant variables remained. To prevent underfitting and uncover potential links between online activity and T2D classification, we added the variables num_views and num_visits.

Hyperparameter Tuning

Tuning hyperparameters leads to significantly better performance(Chapter 2.6). For some algorithms, like Naive Bayes and Logistic Regression, changing hyperparameters did not have an impact on performance, which means we do not have to tune any hyperparameters for those algorithms. On the other hand, we were able to tune hyperparameters for the following four machine learning algorithms:

- The only hyperparameter which needs to be optimized for the Support Vector Machine algorithm is the kernel function. The three most used options for the kernel function are linear, polynomial and radial basis function(RBF). Linear kernels are used for linear classifications, while polynomial and RBF are used for non-linear classifications[7]. After testing, the linear kernel option resulted in the best performance.

- K-Nearest Neighbor has an obvious hyperparameter, namely K. The amount of neighbors K chosen, has a big influence on the algorithm's performance. Therefore, we iterated from K = 1 up to K = 30 and found the best performance for K = 14.

[1]Scikit-learn is a popular python library used to implement machine learning algorithms[24]

- For the Decision Tree algorithm, the maximum depth of the tree is an important hyperparameter that we need to tune to get the best performance of the algorithm. We took a wide range of max depth = 1 up to 20 and tested to find the best performance. In the end, we found that the best accuracy scores happen for a max depth of 7 or higher. Therefore, we chose a max depth of 7.

- Random Forest has two important hyperparameters: max depth of trees and number of trees. After applying a GridSearch[2] using scikit-learn's 'GridsearchCV' function, we found the best combination of hyperparameters was: max depth = 7 and number of trees = 50.

5.3.2 Scikit-learn and SHAP

After optimizing and scaling the machine learning algorithms using sklearn, we are also able to efficiently execute them using sklearn. Since sklearn is a library containing functions to implement machine learning algorithms, we were able to execute Support Vector Machine('SVC'), Logistic Regression('LogisticRegression'), K-Nearest Neighbor('KNeighborsClassifier'), Naive Bayes('GaussianNB'), Decision Tree('DecisionTreeClassifier') and Random Forest('RandomForestClassifier') with it.

To explain the predictions made by the machine learning algorithms and determine the most impactful variables, we use SHAP. SHAP can be applied to any machine learning algorithm (linear, tree-based and other). SHAP uses Shapley values to indicate a variable's contribution to the prediction. The Shapley values are calculated for each variable by considering all possible variable combinations, by measuring how including a particular variable changes the prediction compared to the predictions of all possible subsets of variables. However, calculating Shapley values for all possible variable combinations is computationally expensive. Therefore, the Kernel SHAP method was created[28]. Kernel SHAP approximates the Shapley values using the weighted average difference(Chapter 2.9) of an algorithm's output for different subsets of variables.
After executing the six different machine learning algorithms, we are able to calculate and plot the Shapley values using 'KernelExplainer' and 'summary_plot' from the 'shap' library. This allows us the get the print the specific Shapley values of each variable per class.

[2]Grid search is a hyperparameter tuning approach that searches through a set of hyperparameter values in order to find the combination that results in the best performance.

5.3.3 K-Fold Cross Validation

One way to get more reliable accuracy scores is by using K-fold cross validation(Chapter 2.8). Two popular values for K in K-fold cross validation are 5 and 10. 5-fold cross validation is computationally less expensive since it validates the algorithm 5 times instead of 10. Therefore, in case of a large data set 5-fold cross validation is a good choice. On the other hand, 10-fold cross validation provides more reliable estimates of an algorithm's performance compared to 5-fold cross validation since it validates the algorithm 10 times instead of 5. Especially dealing with a smaller data set, 10-fold cross validation gives a more robust evaluation at the cost of more computational costs. Since there are benefits to both 5 and 10 fold cross validation, we tried both. If the performance is consistent across both options, 5-fold cross validation is preferred since it uses less computational power. However, if the performance estimates differ, 10-fold cross validation is likely more accurate and therefore preferred.

5.3.4 Evaluation Metrics

There are multiple different evaluation metrics(Chapter 2.9) that can be used to evaluate the performance of the machine learning algorithms. We will be using the following evaluation metrics to evaluate the performance of the algorithms used in our research:

- Classification report is a great evaluation metric since it includes the precision, recall, F1-score and support. It is especially useful to evaluate the performance for the two classes and compare performance of those two classes.

- ROC plots the true positive rate against the false positive rate at different threshold values. ROC is the probability curve while AUC represents the degree of separability(with a higher AUC indicating greater ability to separate the classes). Therefore, ROC-AUC is a great way to evaluate how well the algorithm distinguishes the two classes. The higher the AUC score, the better the algorithm is predicting classes correctly.

- A confusion matrix gives a more specific evaluation of an algorithm's performance. Since it displays the number of True Positives(TP), False Positives(FP), True Negatives(TN) and False Negatives(FN).

Chapter 6

Results

6.1 Measurement Visualisation

As explained in Chapter 5.1, we created the following four figures from the measurements done by members of JLAM:

1. Figure 6.1a displays the average body weight difference from the first measurement up to 30 months of measuring in kilograms. This figure displays the average trend of 7717 measurements from 234 individuals. However, there is a decline in the number of individuals measuring their weight over the course of the 30 months, resulting in the first 12 months(average of 150 individuals) containing more individuals then the last 18 months(average of 70 individuals). We can see that the weight difference drops to -9kg and eventually stabilises between -6kg and -9kg.

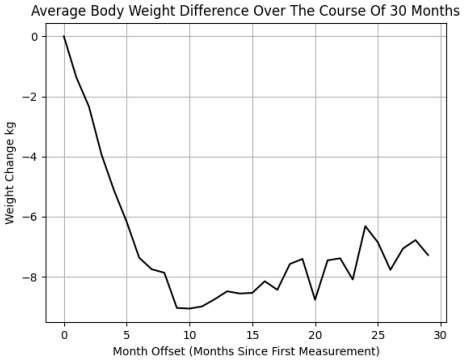
2. Figure 6.1b displays the average WtHr difference from the first measurement up to 30 months of measuring in centimeters. This figure displays the average trend of 7387 measurements from 231 individuals. Again, there is a drop in the number of unique individuals measuring their WtHr, which means the first 12 months are better represented then the last 18 months. We can see a similar trend to Figure 6.1a, WtHr difference drops to -0.07cm and stabilises between -0.045cm and -0.065cm.

3. Figure 6.1c displays the average glucose difference from the first measurement up to 30 months of measuring in mmol/L. This figure displays the average trend of 6808 measurements from 212 individuals. The first 12 months are represented by an average of 140 individuals and the last 18 months are represented by an average of 59 individuals. The average glucose levels difference drop to around -1.55mmol/L before stabilizing between -1mmol/L and -1.55mmol/L.

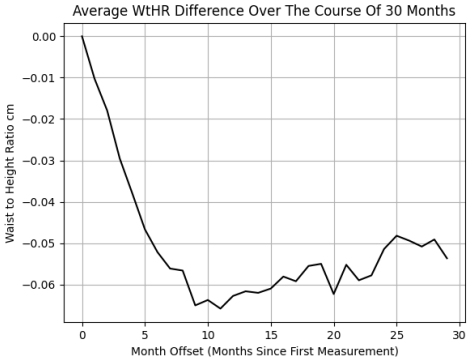
4. Figure 6.1d displays the average HbA1c difference from the first measurement up to 30 months of measuring in mmol/mol. This figure displays the average trend of 698 measurements from 41 individuals. The number of measurements is significantly less than for the other variables because HbA1c can only be measured by a GP. This also means that the figure is not as well represented as the other three figures and thus fluctuates more. As can be seen in figure 6.1d, there is a significant drop to -25mmol/mol, however, the figure mostly ranges between -10mmol/mol and -20mmol/mol.

Figure 6.1: Measurement Visualisation over the Course of 30 Months

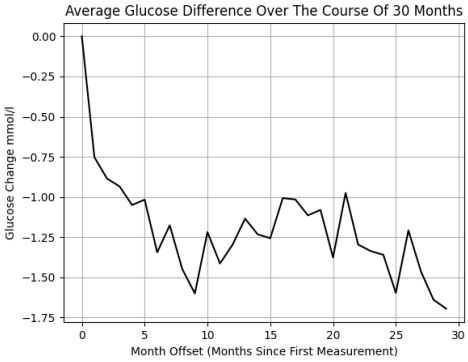
(a)



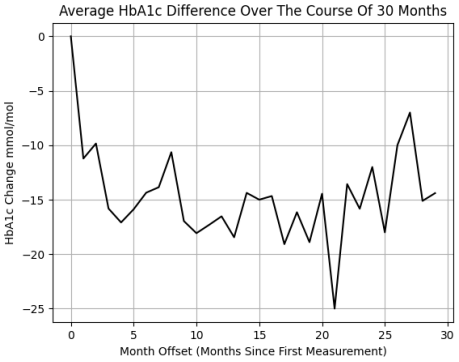
(b)



(c)



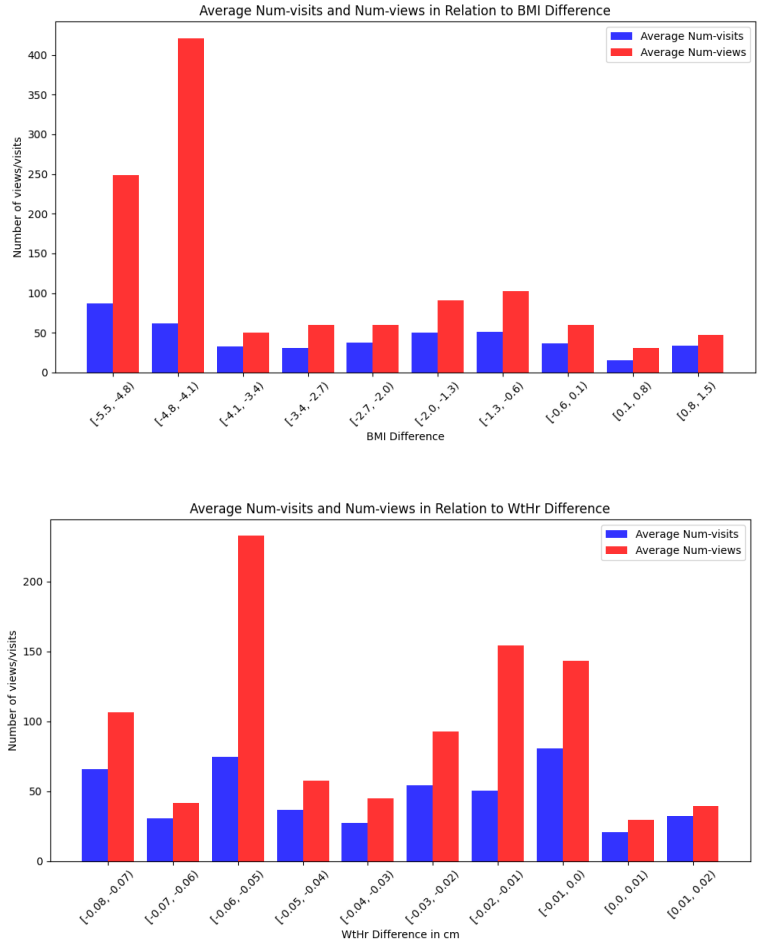
(d)



6.2 Online Activity

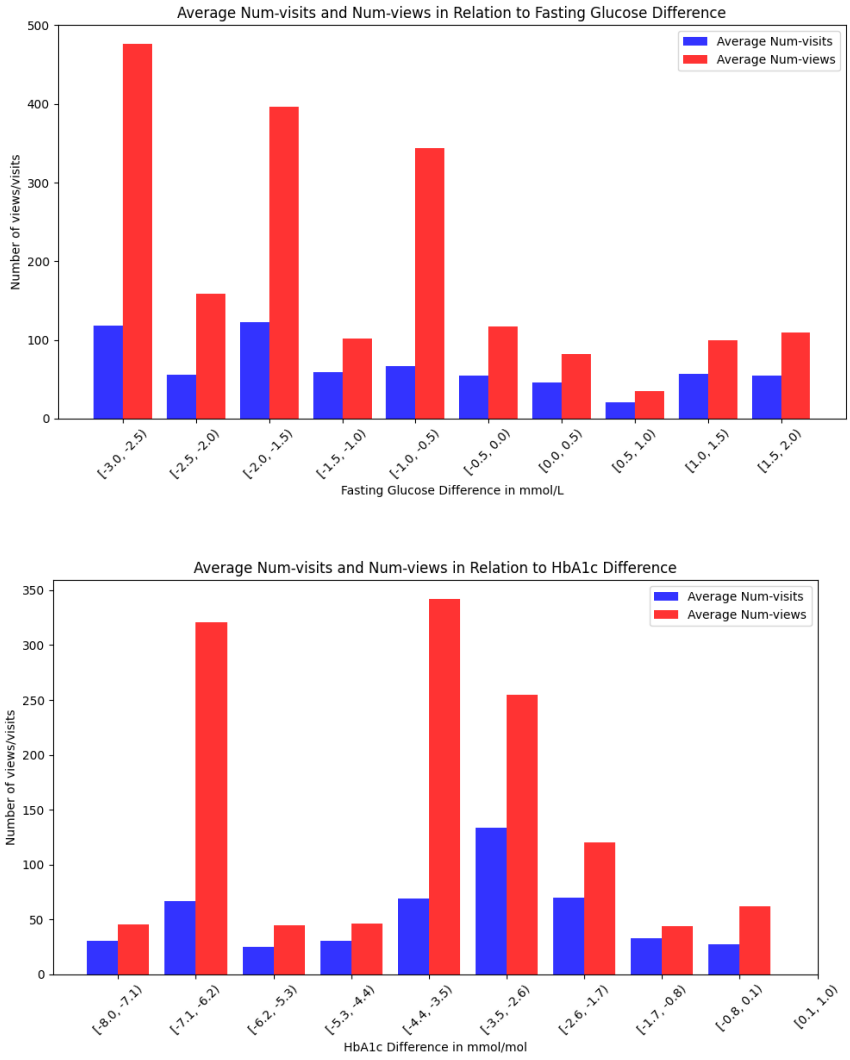
Figure 6.2 displays the average number of views and visits in relation to different BMI/WtHr difference intervals. The value difference of an individual is calculated by subtracting their last measured value by their first measured value. For BMI the amount of online activity is relatively consistent between -4 to 1.5. We do however see a slight increase for interval [-2.0, -1.3] and [-1.3, -0.6]. Furthermore, we do see a clear increase in online activity at interval [-5.5, -4.8] and [-4.8, -4.1]. For WtHr, online activity is overall higher when WtHr has decreased. With higher spikes at interval [-0.08, -0.07], [-0.06, -0.05], [-0.03, -0.02], [-0.02, -0.01] and [-0.01, 0.0].

Figure 6.2: Online Activity Related to BMI and WtHr



In figure 6.3, we see the online activity in relation to glucose levels and HbA1c levels. The bar chart containing glucose has a clear pattern, a decline in fasting glucose levels results in an increased amount of online activity. HbA1c is a bit inconsistent, with high spikes at interval [-7.1, -6.2], [-4.4, -3.5] and [-3.5, -2.6] and low spikes at the other intervals.

Figure 6.3: Online Activity Related to Fasting Glucose and HbA1c



6.3 Machine Learning

6.3.1 Evaluation Metrics

The evaluation metrics used in our research are evaluated on the test set of our data set. As stated before, the data set gets split as follows: 80%(125 individuals) of the data is used to train the algorithms and 20%(31 individuals) is used to test the algorithms. Furthermore, all evaluation metrics have been employed using 10-fold cross validation.

Classification Report

As explained in Chapter 2.9, classification reports are a great way to evaluate the performance of predictive machine learning algorithms. Therefore, we will now display the classification reports for the six predictive machine learning algorithms used in our research. Table 6.1 shows the classification report of Support Vector Machine and Logistic Regression. Their classification report were identical to each other. Both algorithms are slightly better at classifying class 0 (low risk diabetes) than classifying class 1 (high risk diabetes). Overall, the weighted average and macro average scores are between 0.81 and 0.84.

Table 6.1: Classification Report SVM and LR

	Precision	Recall	F1-score	Support
Low risk diabetes	0.86	0.9	0.88	20
High risk diabetes	0.80	0.73	0.76	11
Accuracy			0.84	31
Macr avg	0.83	0.81	0.82	31
Weighted avg	0.84	0.84	0.84	31

The classification report of the K-Nearest Neighbor algorithm is displayed in Table 6.2. The performance of KNN is slightly better then the performance of SVM and LR, with the macro and weighted average being between 0.84 and 0.88. However, there is a significant difference in recall scores for the two classes.

Table 6.3 displays the classification report for the Random Forest algorithm. RF scores even higher then the previous three algorithms, with macro and weighted average scores between 0.89 and 0.91. Furthermore, the evaluation metric scores per class are very similar.

Table 6.4 shows the classification report of Naive Bayes algorithm. Only the precision scores of the two classes differ significantly. The macro and weighted average scores are between 0.86 and 0.88.

Table 6.2: Classification Report KNN

	Precision	Recall	F1-score	Support
Low risk diabetes	0.86	0.95	0.90	20
High risk diabetes	0.89	0.73	0.80	11
Accuracy			0.87	31
Macr avg	0.88	0.84	0.85	31
Weighted avg	0.87	0.87	0.87	31

Table 6.3: Classification Report RF

	Precision	Recall	F1-score	Support
Low risk diabetes	0.95	0.90	0.92	20
High risk diabetes	0.83	0.91	0.87	11
Accuracy			0.90	31
Macr avg	0.89	0.90	0.90	31
Weighted avg	0.91	0.90	0.90	31

Table 6.4: Classification Report NB

	Precision	Recall	F1-score	Support
Low risk diabetes	0.94	0.85	0.89	20
High risk diabetes	0.77	0.91	0.83	11
Accuracy			0.87	31
Macr avg	0.86	0.88	0.86	31
Weighted avg	0.88	0.87	0.87	31

As can be seen in Table 6.5, Decision Tree algorithm has the worst performance. The evaluation metric scores show that the algorithm performs poorly especially for class 1(high risk diabetes). The macro and weighted average scores are between 0.72 and 0.76.

Table 6.5: Classification Report DT

	Precision	Recall	F1-score	Support
Low risk diabetes	0.83	0.75	0.79	20
High risk diabetes	0.62	0.73	0.67	11
Accuracy			0.74	31
Macr avg	0.72	0.74	0.73	31
Weighted avg	0.76	0.74	0.75	31

ROC-AUC

The ROC curves were plotted using the hyperparameter values mentioned in Chapter 5.3.1. The following AUC scores are retrieved from the ROC-curves displayed in Figure 6.4:

1. RF, AUC = 0.959
2. NB, AUC = 0.95
3. LR, AUC = 0.923
4. SVM, AUC = 0.909
5. KNN, AUC = 0.909
6. DT, AUC = 0.739

All algorithms except DT have an AUC of >0.9. RF and NB performed best, while DT performed worst.

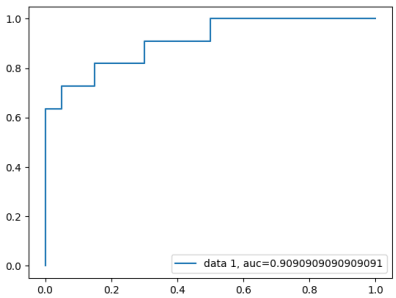
Confusion Matrices

Figures 6.5 and 6.6 display the confusion matrix of every machine learning algorithm used. In the confusion matrices, 0(Negative) corresponds to low risk T2D and 1(Positive) corresponds to high risk T2D. The confusion matrix consists of four squares, where:

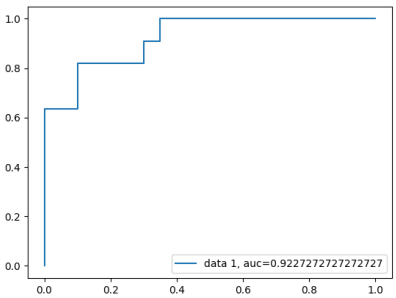
- The top-left square displays the number of TN values, we want this number to be high since it correctly predicts a data point as 'low risk diabetes'.
- The top-right square displays the number of FN values, we want this number to be low since it wrongfully predicts a data point as 'high risk diabetes'.
- The bottom-right square displays the number of TP values, we want this value to be high since it correctly predicts a data points as 'high risk diabetes'.
- The bottom-left square displays the number of FP values, we want this number to be low since it wrongfully predicts a data point as 'low risk diabetes'.

Figure 6.4: ROC-curves

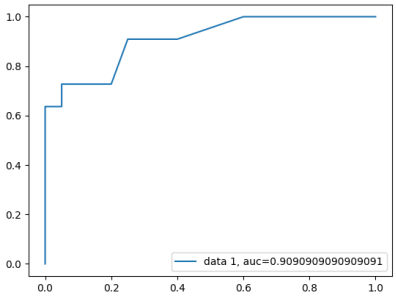
(a) SVM



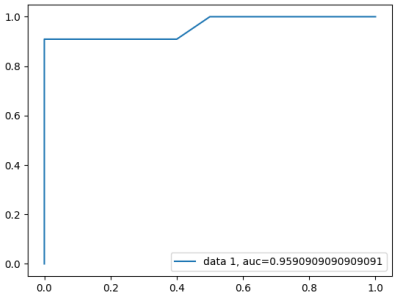
(b) LR



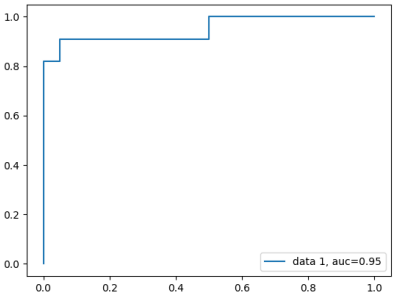
(c) KNN



(d) RF



(e) NB



(f) DT

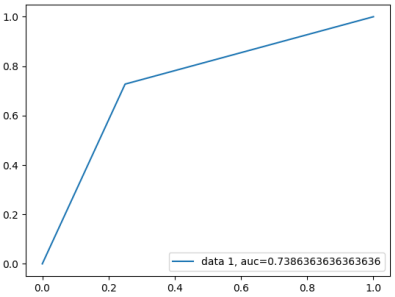
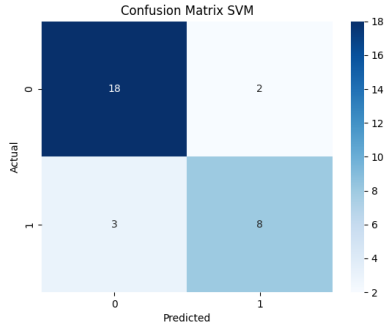
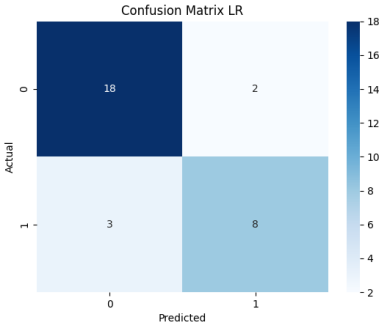


Figure 6.5: Confusion Matrices

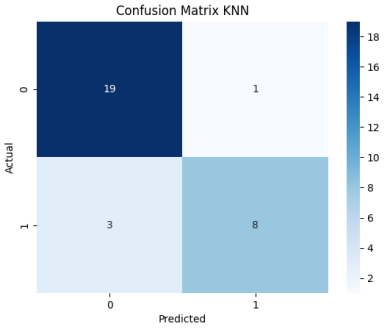
(a) SVM



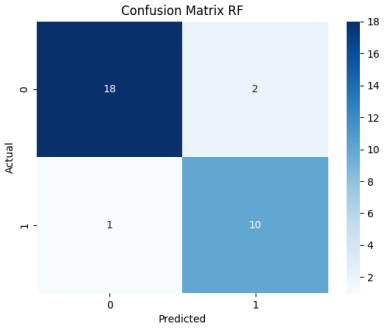
(b) LR



(c) KNN



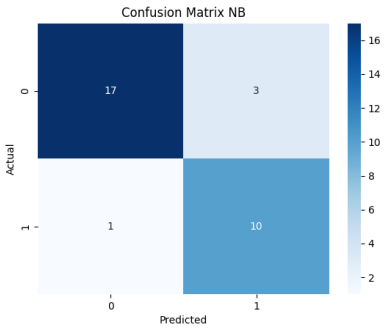
(d) RF



SVM and LR have identical confusion matrices(Figure 6.5a and Figure 6.5b). They both correctly predicted 18 low risk diabetes data points and 8 high risk diabetes data points, while incorrectly predicting 2 low risk diabetes data points as high risk diabetes and 3 high risk diabetes data points as low risk diabetes. Figure 6.5c shows that KNN is the most accurate algorithm correctly predicting low risk diabetes since it has the most TN values(19). Figure 6.5d and Figure 6.6a show that NB and RF are most accurate at correctly predicting high risk diabetes since they have the highest TP value(10). Figure 6.6b displays the performance of DT. DT is the least accurate at correctly predicting low risk diabetes and high risk diabetes.

Figure 6.6: Confusion Matrices

(a) NB



(b) DT

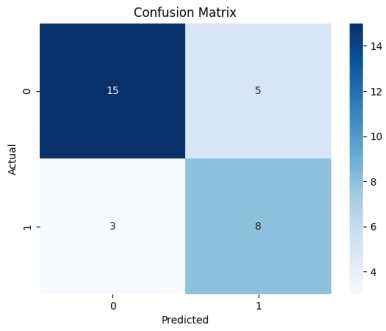


Table 6.6: Accuracy Scores using 5-Fold Cross Validation

Algorithm	Accuracy
Logistic Regression	0.837
Random Forest	0.820
Support Vector Machine	0.820
Decision Tree	0.755
K-Nearest Neighbor	0.747
Naive Bayes	0.723

K-fold Cross Validation Accuracy Scores

Table 6.6 displays the accuracy scores using 5-fold cross validation. NB performed worst, with an accuracy score of 0.723. LR, RF and SVM are the best performing algorithms with accuracy scores of 0.837, 0.82 and 0.82 respectively. Table 6.7 is very similar to Table 6.6, however using 10-fold cross validation resulted in a better accuracy score for KNN compared to 5-fold cross validation. For both K values, the top three best performing algorithms remained the same: LR, RF and SVM with accuracy scores between 0.820 and 0.837.

Table 6.7: Accuracy Scores using 10-Fold Cross Validation

Algorithm	Accuracy
Logistic Regression	0.836
Random Forest	0.829
Support Vector Machine	0.827
K-Nearest Neighbor	0.779
Decision Tree	0.748
Naive Bayes	0.729

6.3.2 SHapley Additive exPlanations

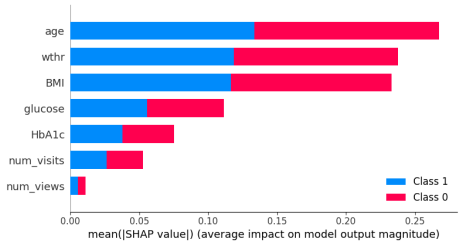
Table 6.8 displays the average absolute Shapley value of each variable for the absolute Shapley values obtained from the six different machine learning algorithms. Figure 6.7a up to 6.7e have similar results, with WtHr and BMI having the highest Shapley value and num-visits and num-views having the lowest Shapley value. Only figure 6.7f deviates from the other figures by giving WtHr a low Shapley value while giving BMI almost double the Shapley value compared to the Shapley values of the other variables. Therefore, Table 6.8 contains one column including and one column excluding the results of Figure 6.7f. As can been seen, the average absolute Shapley value of BMI drops significantly excluding the Shapley values obtained from DT. On the other hand, the average absolute Shapley value of WtHr increases quite a bit excluding the Shapley values of DT.

Table 6.8: Average absolute SHAP values

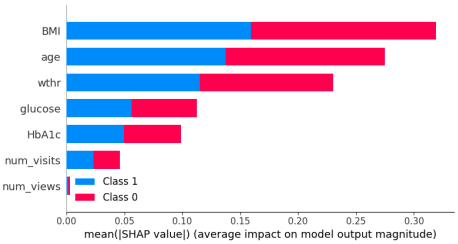
Variable	Average Shapley value including DT	Average Shapley value excluding DT
BMI	0.3	0.25
WtHr	0.24	0.275
Age	0.19	0.175
HbA1c	0.14	0.12
Glucose	0.12	0.12
Num-visits	0.06	0.05
Num-views	0.03	0.02

Figure 6.7: SHAP values

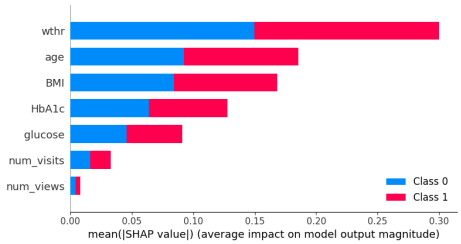
(a) SVM



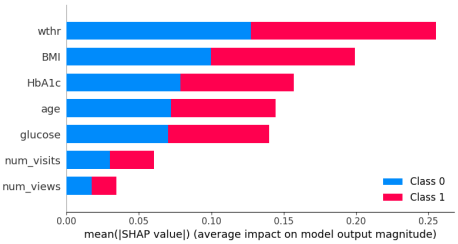
(b) LR



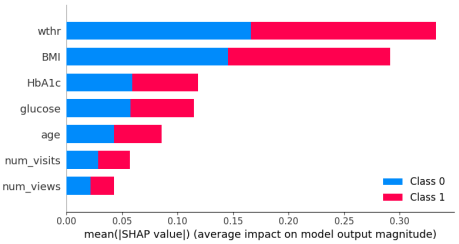
(c) KNN



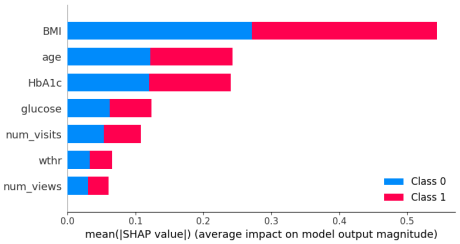
(d) RF



(e) NB



(f) DT



Chapter 7

Discussion

Our research aimed to uncover the most impactful variables leading to T2D classification by employing machine learning algorithms and SHAP. Initially, we visualized the measurements of BMI, WtHr, glucose and HbA1c. Our findings revealed that, on average, members of JLAM were able to significantly reduce these variable values over a 30-month period. Secondly, we investigated potential connections between a member's online activity and their measurements. Our analysis indicated that increased online activity correlated with improved measurement results. Lastly, we achieved accuracy results ranging from 72.9% to 83.6% using six different machine learning algorithms. Notably, the obtained Shapley values indicates that BMI and WtHr were the most impactful variables for T2D prediction.

While previous research predominantly relied on the PIMA database[42][31], our research utilized a data set provided by JLAM. This data set includes both males and females and uses variables such as HbA1c, WtHr and online activity. Using this data set, we could assess the individual impact of each variable, including online activity, on T2D prediction. With our research, we were able to further explore the use of machine learning for predicting T2D. As well as, offer insights into average measurement trends and highlight key variables that individuals should monitor to mitigate the risk of T2D.

7.1 Online Activity

Making healthy lifestyle changes becomes challenging in a world marked by reduced physical activity and easy access to unhealthy food. One effective way to improve your lifestyle is to establish a motivational environment that encourages individuals to lead healthier lives. Community groups, such as those available on JLAM's website, demonstrate their effectiveness in motivating and inspiring individuals by connecting them with others facing similar challenges[26]. Previous research has demonstrated that support

and information from online health communities positively affects healthy lifestyle changes[40]. Hence, utilizing the data provided by JLAM presented an ideal opportunity to explore the potential benefits of active participation in such communities for JLAM members.

The results derived from SHAP analysis indicate that online activity has minimal to no influence on the prediction of T2D risk. This validates the notion that online activity is not associated with a increased or reduced risk of T2D development. Nevertheless, when comparing the measurements alongside online activity, it became clear that members with positive measurement differences showed significantly higher online activity. In contrast, members with the same or worsened measurement differences showed lower online activity. This finding supports the claim that online communities, such as JLAM's website, contribute positively to encouraging healthy lifestyle changes.

These results suggest that online activity is not a reliable indicator for classifying T2D; however, increased online activity is related to positive measurement differences. This aligns with previous research stating that health communities have a positive impact on encouraging a healthier lifestyle.

7.2 Machine Learning Results

We employed six machine learning classification algorithms to predict T2D: Support Vector Machine, Logistic Regression, K-Nearest Neighbor, Random Forest, Naive Bayes and Decision Tree. These algorithms are commonly utilized in research related to the prediction of disease diagnoses. SVM has been the most frequently used algorithm for predicting T2D, often providing more accurate results compared to other algorithms[1][31][18].

In previous research, accuracy results for the six algorithms mentioned above, typically ranged between 0.7 and 0.76 without using pre-processing, feature selection and hyperparameter tuning. However, using these techniques resulted in accuracy scores between 0.8 and 0.87[42][20][31]. Notably, accuracy results of 90%(KNN) and 82.5%(DT) were achieved with extensive pre-processing to eliminate irrelevant variables and reduce dimensions[39]. Another research conducted by Faizan Zafar et al. focused on F1-score as an evaluation metric, yielding the following results: 0.807(KNN), 0.793(RF), 0.775(LR), 0.772(DT) and 0.64(NB)[10].

In our research, we obtained the following accuracy scores: 83.6%(LR), 82.9%(RF), 82.7%(SVM), 77.9%(KNN), 74.8%(DT) and 72.9%(NB). These accuracy scores fall within the range of 70-87%, and are thus in line with

the accuracy scores observed in previous research. Although our new data set did not lead to an improvement in accuracy scores, there is potential for enhanced results as the data set expands in the future.

The two worst performing algorithms were Decision Tree and Naive Bayes, which is a common trend in most research[10][20]. NB performed poorly according to the accuracy scores obtained using 10-fold cross validation, likely due to the violation of NB's assumption of independence between variables. This assumption is challenged by the relationship between variables such as BMI and WtHr, as well as HbA1c and glucose. Consequently, NB probably performs suboptimally, resulting in lower accuracy results than expected. One reason for DT being among the worst performers is its tendency to struggle with relatively smaller data sets. In contrast, algorithms like SVM and RF perform well even with smaller data set.

Two of the best performing algorithms are Logistic Regression and Support Vector Machine, with very similar accuracy scores. The results from the confusion matrix and classification report are identical for these two algorithms. This similarity likely occurs from the use of a linear kernel in SVM after tuning the kernel hyperparameter, making SVM structurally similar to LR[5]. The strong performance of LR, RF and SVM aligns with findings from prior research[1].

7.3 Key Variables

SHapley Additive exPlanations is an interpretability tool used for explaining machine learning algorithms. In the healthcare sector, SHAP has been employed to determine the most impactful variables associated with suicide attempts, death by sepsis and T2D diagnoses[18][35][20]. I. Tasin et al. utilized SHAP to determine the most impactful variables for T2D prediction in the PIMA Indians Database, resulting in the following Shapley values: $\sim$0.33 (Glucose), $\sim$0.235 (BMI), $\sim$0.125 (Age), $\sim$0.09 (SkinTickness), $\sim$0.08 (Pregnancies), $\sim$0.08 (BloodPressure) and $\sim$0.06 (Insulin).

In our research, we used similar variables (Age, BMI, Glucose) and introduced new variables (HbA1c, WtHr, num_views, num_visits). BMI and WtHr had the most impact, followed by age, HbA1c and glucose. As previously mentioned, num_views and num_visits had the least impact, suggesting that the amount of online activity does not correlate with T2D classification. Furthermore, in our research BMI had the most significant impact, while glucose had the least impact(excluding online activity). This differs from the findings of I. Tasin et al., who concluded that glucose had the most

impact. This difference is likely due to the introduction of new variables in our research, with HbA1c being similar to glucose.

7.4 Limitations and Future Work

Our research has a few limitations, with the first one being related to the data set. Although the data set is unique and includes valuable new variables like HbA1c and WtHr, it is somewhat incomplete and small. There is a significant lack of measurement for HbA1c, as it is not frequently measured by most individuals. Furthermore, the data set could benefit from expansion, and we cannot guarantee that every member consistently does measurements. It is possible that individuals with negative measurement results stopped measuring completely due to a decline in motivation. Therefore, a larger and more consistent group of members would improve the data set.

7.4.1 Recommendations for JLAM

Considering that the variables indicating online activity had minimal impact on the predictions made by the machine learning algorithms, it would be advisable to replace them in future research. JLAM could ask their members to fill in relevant information such as: ethnic background, family members with T2D and if a person smokes[11][43]. Using these variables, their importance can be determined using SHAP and they might help make T2D predictions even more accurate.

Moreover, showcasing the measurement graphs, online activity linked to measurement outcomes, and highlighting the most impactful variables contributing to a reduced risk of developing T2D will not only motivate current JLAM members but also potentially draw in new members. This, in turn, will contribute to an expanded data set.

Chapter 8

Conclusions

Given the increasing amount of T2D diagnoses globally, it is essential to further investigate the use of machine learning algorithms alongside interpretability tools, such as SHAP, to uncover key variables related to T2D management. In our research, we shifted away from the usual use of the PIMA database and instead opted for a new data set provided by JLAM. Our research yielded noteworthy insights. Firstly, we observed a consistent decrease in the average measurements difference related to weight, waist-to-height-ratio, HbA1c and glucose over a 30-month period. Secondly, increased online activity often translated to improved measurement outcomes, suggesting that the beneficial diet and lifestyle tips and tricks shared online have a positive influence on JLAM members who read and use them. Lastly, employing six machine learning classification algorithms, we achieved an accuracy score of 83.6% using Logistic Regression. Additionally, we achieved an F1-score, Recall, Precision and AUC of 0.9, 0.9, 0.89 and 0.959 respectively, with Random Forest. The application of SHapley Additive exPlanations uncovered the most impactful variables for T2D classification, highlighting BMI and WtHr as having the greatest impact, while variables indicating online activity had the least impact.

Our findings serve as valuable information for both current and future members of JLAM, with the intention of inspiring and motivating them to adopt a healthier lifestyle. This is achieved by showcasing the positive trends in average measurements over time. As well as, promoting active participation in online communities, such as those provided by JLAM's website, which has been proven to be beneficial. Additionally, the importance of monitoring BMI and WtHr in T2D management should be emphasized. Our findings are based on data of JLAM's members, but they are relevant to anyone interested in improving their lifestyle. Joining online health communities or monitoring BMI, WtHr, glucose and HbA1c can help individuals keep track of their risk of T2D development.

Bibliography

[1] Zaihisma Binti Che Cob Abir Al-Sideiri and Sulfeeza Bte Mohd Drus. Machine learning algorithms for diabetes prediction: A review paper. *AMC Digital Library*, pages 27–32, 2020.

[2] Sumeet Kumar Agrawal. Metrics to evaluate your classification model to take the right decisions. 2023.

[3] Ahmed. The motivation for train-test split. Technical report, Medium, 2022.

[4] Abdullah Alanazi. Using machine learning for healthcare challenges and opportunities. *ScienceDirect*, 30, 2022.

[5] Tarun Bawa. Machine learning: How are logistic regression and linear svm's similiar? `https://www.quora.com/Machine-Learning-How-are-logistic-regression-and-linear-SVMs-similiar/answer/Tarun-Bawa`.

[6] Pritha Bhandari. How to find outliers — 4 ways with examples and explanation. 2021.

[7] Datacamp. Support vector machines with scikit-learn tutorial. `https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python`.

[8] Julianna Delua. Supervised vs. unsupervised learning: What's the difference? 2021.

[9] Diabetes Fonds. Diabetes in cijfers. `https://www.diabetesfonds.nl/over-diabetes/diabetes-in-het-algemeen/diabetes-in-cijfers`.

[10] Faizan Zafar, Saad Raza, Muhammad Umair Khalid and Muhammad Ali Tahir. Predictive analytics in healthcare for diabetes prediction. *AMC Digital Library*, pages 253–259, 2019.

[11] Diabetes Fonds. Diabetes test. `https://www.diabetesfonds.nl/over-diabetes/diabetes-test`.

[12] GeeksforGeeks. Cross Validation in Machine Learning. https://www.geeksforgeeks.org/cross-validation-machine-learning/.

[13] GeeksforGeeks. K-nearest neighbor(knn) algorithm. https://www.geeksforgeeks.org/k-nearest-neighbours/.

[14] GeeksforGeeks. Logistic regression in machine learning. https://www.geeksforgeeks.org/understanding-logistic-regression/.

[15] GeeksforGeeks. Random forest regression in python. https://www.geeksforgeeks.org/random-forest-regression-in-python/.

[16] Heavy.AI. Feature selection. https://www.heavy.ai/technical-glossary/feature-selection.

[17] HubSpot. Grow better with hubspot. https://www.hubspot.com/.

[18] Igor Pereira Vidal, Marluce Rodrigues, André Pimenta Freire, Uanderson Resende and Erick Galani Maziero. Comparison of explainable machine-learning models for decision-making in health intensive care using shapley additive explanations. *AMC Digital Library*, pages 300–307, 2023.

[19] Python Package Index. hubspot-api-python. https://pypi.org/project/hubspot-api-client/.

[20] Sanjida Islam Isfafuzzaman Tasin, Tansin Ullah Nabil and Riasat Khan. Diabetes prediction using machine learning and explainable ai techniques. *PMC PubMed Central*, pages 1–10, 2022.

[21] Jaiqi Hou, Yongsheng Sang, Yuping Liu and Li Lu. Feature selection and prediction model for type 2 diabetes in the chinese population with machine learning. *AMC Digital Library*, pages 1–7, 2020.

[22] Javapoint. Decision tree classification algorithm. https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm.

[23] Javapoint. Naïve bayes classifier algorithm. https://www.javatpoint.com/machine-learning-naive-bayes-classifier.

[24] Scikit Learn. Machine learning in python. https://scikit-learn.org/stable/.

[25] UCI MACHINE LEARNING. Pima indians diabetes database. https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database.

[26] Je leefstijl als medicijn. Je leefstijl als medicijn. https://jeleefstijlalsmedicijn.nl/.

[27] Je leefstijl als medicijn. Zaterdag wegen en meten. zwem. https://landing.jeleefstijlalsmedicijn.nl/zaterdag-wegen-en-meten-zwem.

[28] Scott Lundberg. Shap documentation. https://shap-lrjball.readthedocs.io/en/latest/index.html.

[29] Batta Mahesh. Machine learning algorithms -a review. Technical Report DOI:10.21275/ART20203995, 2019.

[30] Tobias Geisler Mesevage. What is data preprocessing and what are the steps involved? 2021.

[31] Yuxin Miao. Using machine learning algorithms to predict diabetes mellitus based on pima indians diabetes dataset. *AMC Digital Library*, pages 47–53, 2021.

[32] Michael L Ganz, Neil Wintfeld, Qian Li, Veronica Alas, Jakob Langer and Mette Hammer. The association of body mass index with the risk of type 2 diabetes: a case–control study nested in an electronic health records system in the united states. *BMC*, 6:50, 2014.

[33] Microsoft. Azure sql database. https://azure.microsoft.com/en-us/products/azure-sql/database.

[34] Christoph Molnar. *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*. Independently published, 2022.

[35] Noratikah Nordin, Zurinahni Zainol, Mohd Halim Mohd Noor and Chanlai Fong. Explainable machine learning models for suicidal behavior prediction. *ACM Digital Library*, pages 118–123, 2022.

[36] Kizito Nyuytiymbiy. Parameters and hyperparameters in machine learning and deep learning. *Medium*, pages 1–7, 2020.

[37] World Health Organization. Mean fasting blood glucose. https://www.who.int/data/gho/indicator-metadata-registry/imr-details/2380.

[38] World Health Organization. *Use of Glycated Haemoglobin (HbA1c) in the Diagnosis of Diabetes Mellitus: Abbreviated Report of a WHO Consultation.* Use of Glycated Haemoglobin (HbA1c) in the Diagnosis of Diabetes Mellitus. WHO Press, 2011.

[39] Bouchaib Cherradi Othmane Daanouni and Amal Tmiri. Diabetes diseases prediction using supervised machine learning and neighbourhood components analysis. *AMC Digital Library*, pages 1–5, 2020.

[40] Ping Zhou, Yujie Zhao, Suping Xiao and Kangsheng Zhao. The impact of online health community engagement on lifestyle changes: A serially mediated model. *PubMed Central*, 10, 2022.

[41] Rushikesh Pupale. Support vector machines(svm) — an overview. 2018.

[42] S.M. Hasan Mahmud, Md. Altab Hossin, Md. Razu Ahmed, Sheak Rashed Haider Noori and Md. Nazirul Islam Sarkar. Machine learning based unified framework for diabetes prediction. *AMC Digital Library*, pages 46–50, 2018.

[43] Diabetes UK. Diabetes type 2 - know your risk. `https://riskscore.diabetes.org.uk/start`.

[44] Diabetes UK. Type 2 diabetes. `https://www.diabetes.org.uk/diabetes-the-basics/types-of-diabetes/type-2/preventing`.

[45] World Health Organization. Body mass index (BMI). `https://www.who.int/data/gho/data/themes/topics/topic-details/GHO/body-mass-index`.

[46] Yoon Jeong Son, Jihyun Kim , Hye-Jeong Park, Se Eun Park, Cheol-Young Park, Won-Young Lee, Ki-Won Oh, Sung-Woo Park, and Eun-Jung Rhee. Association of waist-height ratio with diabetes risk: A 4-year longitudinal retrospective study. *PubMed Central*, page 127–133, 2016.

[47] Z Xu, X Qi, A K Dahl, W Xu. Waist-to-height ratio is the best indicator for undiagnosed type 2 diabetes. *Wiley Online Library*, 30:201–207, 2013.

[48] Zihui Yan, Mengjie Cai, Xu Han, Qingguang Chen and Hao Lu. The interaction between age and risk factors for diabetes and prediabetes: A community-based cross-sectional study. *PubMed Central*, 16:85–93, 2023.