**CMP6202**

**AI and Machine Learning Project**

**2023–2024**

Individual Report

Report Title

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Forename** | **Surname** | **Student ID** | **Instructor Name** | **ML Model(s) developed** |
| Aman | Zamarad | 21183681 | Khalid Ismail | K-nearest neighbour(KNN) |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table of Contents

[1 Report Introduction 3](#_Toc148277256)

[1.1 Dataset identification 3](#_Toc148277257)

[1.2 Supervised learning task identification 3](#_Toc148277258)

[2 Exploratory Data Analysis 3](#_Toc148277259)

[2.1 Question(s) identification 3](#_Toc148277260)

[2.2 Splitting the dataset 3](#_Toc148277261)

[2.3 Exploratory Data Analysis process and results 3](#_Toc148277262)

[2.4 EDA conclusions 4](#_Toc148277263)

[3 Experimental Design 4](#_Toc148277264)

[3.1 Identification of your chosen supervised learning algorithm(s) 4](#_Toc148277265)

[3.2 Identification of appropriate evaluation techniques 4](#_Toc148277266)

[3.3 Data cleaning and Pre-processing transformations 4](#_Toc148277267)

[3.4 Limitations and Options 4](#_Toc148277268)

[4 Predictive Modelling / Model Development 5](#_Toc148277269)

[4.1 The predictive modelling process 5](#_Toc148277270)

[4.2 Evaluation results on “seen” data 5](#_Toc148277271)

[5 Evaluation and further modelling improvements 5](#_Toc148277272)

[6 Conclusion 6](#_Toc148277273)

[6.1 Summary of results 6](#_Toc148277274)

[6.2 Reflection on Individual Learning 6](#_Toc148277275)

[7 References 6](#_Toc148277276)

# Report Introduction

This report encapsulates a reflective journey, documenting personal experiences, insights, and learnings acquired throughout the coursework. The purpose is to articulate a comprehensive understanding of the coursework, drawing upon personal perspectives and reflections.

Here I will delve into the details of a dataset by performing Exploratory Data Analysis(EDA) and forming an in depth understanding of the dataset’s nature.

Once completing the EDA, the dataset will be cleaned in the pre-processing phase and split into training and testing. I will then train a classification algorithm using this data to form a predictive model which can be evaluated.

## Dataset identification

The dataset I am working with is called ‘Diabetes Dataset’. The dataset contains 9 columns(attributes) and 769 rows (Women) , predicting whether an individual is likely to have diabetes or not. It is important to note that this dataset is specific to women aged 21 and older from the Pima Indian heritage. This means that the model may not generalize well to other populations.

This dataset is relevant to the project as I intend to use machine learning to predict whether an individual is likely to have diabetes, and this dataset has health attributes that are directly related to diabetes, meaning we the dataset is highly relevant to our goal.

The 'Outcome' column serves as the target variable in the dataset, indicating whether an individual has diabetes (coded as 1) or does not have diabetes (coded as 0), making it the focal point for predictive modelling in the context of this project. This makes it categorical.

This dataset is found on Kaggle. It is considered reliable as it is originally from the National Institute of Diabetes and Digestive and Kidney Diseases in the USA, with the source being listed (www.kaggle.com, n.d.).

**The dataset contains the following attributes**:

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Pregnancies (numerical) | The number of pregnancies an individual has had |
| Glucose (numerical) | The glucose level in an individual’s blood |
| BloodPressure (numerical) | The blood pressure measurement of an individual |
| SkinThickness (numerical) | The thickness of an individual’s skin |
| Insulin (numerical) | To express the Insulin level in blood |
| BMI (numerical) | The body mass index of an individual |
| DiabetesPedigreeFunction (numerical) | The likelihood of an individual having diabetes based on their family history |
| Age (numerical) | The age of an individual |
| Outcome (categorical | A binary attribute indicating whether an individual has diabetes (1) or doesn’t have diabetes (0) |

## Supervised learning task identification

The predictive problem I aim to solve is determining, for women aged 21 and older from the Pima Indian heritage, whether an individual is likely to have diabetes or not based on their health-related features.

Given that the outcome variable is categorical and involves discrete classes(Diabatic or not Diabetic), the appropriate modelling technique for this is classification. Models I can use that are well-suited for predicting categorical outcomes are, logistic regression, decision trees, random forests, and support vector machines (Sharma, 2021). Due to the nature of the outcome variable, regression models wouldn’t be suitable as they are designed to predict continuous numerical values, not discrete values (RapidMiner, n.d.).

The target variable used to establish the ground truth for this supervised learning task is "Outcome." The "Outcome" variable is binary, with values of 0 and 1, indicating the absence or presence of diabetes, respectively. This variable will be predicted by the model based on the other features in the dataset. The ability to predict the likelihood of diabetes for this demographic of women aged 21and older from Pima Indian Heritage will provide insights into the risk of diabetes within this specific group. This can lead to several implications, such as early intervention, customized healthcare, health planning, and resource allocation, to help lessen the risk of diabetes for these people (Laila et al., 2022).

# Exploratory Data Analysis

## Question(s) identification

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Question** | **Relevance** | **Hypothesis** |
| Q1 | Does Age increase the likelihood of diabetes? | Age is a well-established risk factor, and understanding its impact is crucial for risk assessment. (CDC, 2020) | The likelihood of diabetes increases with advancing age due to age-related changes in metabolism and insulin resistance. |
| Q2 | Is BMI a decisive factor in predicting the risk of diabetes, and how does it vary across different age groups? | BMI is a crucial measure of metabolic health and body composition, and changes with age can reveal information about the risk of age-related diabetes. (Gupta and Bansal, 2020) | BMI is a significant predictor, and its impact on diabetes risk varies across age groups, reflecting the interplay between body composition and age-related metabolic changes. |
| Q3 | Does more pregnancies have a positive correlation with likelihood of having diabetes? | Understanding the relationship between pregnancies and diabetes risk is vital for women's health, especially during and after childbearing years. (Liu et al., 2020) | A higher number of pregnancies correlates positively with an increased likelihood of diabetes, potentially influenced by hormonal changes and metabolic adaptations associated with multiple pregnancies. |
| Q4 | Does age impact the Diabetes pedigree function? | Age-specific genetic influences on diabetes risk can be identified by evaluating the age-genetic relationship to diabetes. | Diabetes pedigree function is affected by age, with genetic factors playing a role in the overall likelihood of diabetes. |
| Q5 | Is there a significant association between DPF and the likelihood of diabetes? | Understanding the significance of Diabetes Pedigree Function (DPF) will show how genetic factors contribute to the familial transmission of diabetic risk | Diabetes pedigree function is significantly associated with an increased likelihood of diabetes, highlighting the importance of genetic factors in determining diabetes susceptibility. |
| Q6 | Do individuals with higher blood pressure exhibit an increased likelihood of diabetes? | Investigating the link between blood pressure and diabetes risk to inform cardiovascular health assessments. | Higher blood pressure is associated with an increased likelihood of diabetes, indicating the interconnectedness of cardiovascular health and metabolic disorders. |
| Q7 | How does blood pressure correlate across various age brackets | Exploring the relationship between blood pressure and age-specific diabetes risk informs age-tailored cardiovascular health strategies | The older an individual gets, the more likely they are to have a higher blood pressure. |
| Q8 | What are the average insulin levels for various age brackets? | Understanding age-specific patterns in insulin levels aids in identifying age-related variations in insulin sensitivity and diabetes risk | Insulin levels vary across age brackets, and identifying these variations provides insights into age-specific metabolic health and diabetes risk profiles. |
| Q9 | What is the relationship between Insulin and glucose? | Insulin and Glucose provides a comprehensive understanding of diabetes, from its underlying mechanisms to diagnostic tools, risk assessment, and treatment strategies | There is a significant and direct correlation between insulin levels and glucose levels |

## Splitting the dataset

A Train-Test split will be used to assess the model’s performance accurately by simulating its ability to make predictions on data it has never encountered during training phase. Essentially, the training set is used to teach the model, while the testing set serves as a separate dataset which be used to assess the model’s performance on new unseen data.

Data leakage occurs when information that should not be available to the model during training is inadvertently included, leading to overly optimistic model performance. This could mean that the model’s learning patterns won’t generalize well to the new unseen data. Data leakage can occur from preprocessing errors as well as accidental use of test set information being used in the training phase.

Using a train-test split will prevent data leakage from happening as all the preprocessing steps, feature engineering, or model training are conducted only on the training set, preventing information from the testing set influencing the model. This is why splitting the dataset is crucial when trying to develop a predictive model.

In the context of the diabetes dataset, I will use the train-test split to segregate the data into two subsets. The train dataset will consist of 80% of the overall dataset whereas the test dataset will be 20%. Approximately 615 rows will be used for testing and 154 rows will be used for training.

## Exploratory Data Analysis process and results

Using the pandas library, the first query checked was the overall size of the dataset, which can be seen bellow.

A screenshot of a computer

Description automatically generated

As mentioned previously, there are 768 rows and 9 columns. Each feature and their corresponding data types are shown below.

A screenshot of a computer

Description automatically generated

In the dataset, the 'Outcome' column serves as a binary classification indicator, taking values of 0 or 1 to represent certain outcomes. However, when examining the data types using Pandas (df.dtypes), you'll find that the 'Outcome' column is categorized as an integer (int64). Pandas employs integer types for binary classification due to factors such as computational efficiency, memory optimization, and compatibility with various operations.

A screenshot of a computer

Description automatically generated

Here ,Min values in Glucose,BloodPressure,SkinThickness,Insulin and BMI columns are 0,which isn’t possible ,hence the zeros represent the missing values here.

Now, getting rid of these rows for now as they won’t contribute efficiently.

A screenshot of a computer

Description automatically generated

Now we can see there are no minimum values of 0 for any feature apart from “Pregnancies”.

**Univariate:**

Here we are focusing on individual features to understand the distribution, central tendency, and dispersion of a single variable, providing insights into its characteristics. Univariate analysis is a fundamental step in EDA and is particularly useful for gaining a preliminary understanding of the diabetes dataset.

**Age:**

The older an individual gets, the more prone they are to getting diabetes. This can be due to several factors that change the human body mechanics.

A screen shot of a computer screen

Description automatically generated

The box plot shows the presence of outliers above the horizontal line. We can also see there is a lot of data for individuals between 25 – 35 years. This is due to the way that the data was collected. The median age is just below 30, showing there is much more data for younger years than the those that are older than 50.

We can now assess the number of women in different age groups given whether they have diabetes or not via a histogram.

A blue graph with white text

Description automatically generated

We can see that this suggests most women with diabetes range from 22 to 30 years old. This makes sense as there was a lot of record that were conducted for these ages. This also suggests that the frequency of women with diabetes decreases as age increases which isn’t necessarily the case. This is due to the nature of the dataset as there isn’t a lot of data conducted for older ages.

**Blood Pressure:**

Blood Pressure can have an effect on an individuals insulin resistance which can make the risk of getting diabetes higher.

**A diagram of a box plot

Description automatically generated**

The average blood pressure rating is within the 60 to 80, shown by the box plot. There are a few outliers, some that are 0. This is impossible as one’s blood pressure cannot be 0. This shows that there are some missing values in the dataset which we need to impute later on. Removin these records isn’t practical as too much valuable data will be lost.

A diagram of a distribution of blood pressure

Description automatically generated

This histogram further supports the box plot as the frequencies are highest at points 60 to 80. There is a lot of data that covers that percentile of the histogram. We can see that there is a significant amount of records that have 0 as their blood pressure, further reinforcing the fact that removing these rows isn’t practical as it makes up a large chunk of our database.

**DiabetesPedigreeFunction:**

**A graph of a box with a line

Description automatically generated with medium confidence**

We can see there are quite a few outliers that are above the 1.25 mark of the DPF. The box plot shows that the majority of rows have a lower DPF value ranging between 0.25 and 0.75.

**Age:**

**A graph of a distribution of age

Description automatically generated**

Moving to the age feature, we can see there is a high frequency of individuals aged between 22 and 25 as shown from the histogram above. The frequency slowly decreases as the age increases, showing a negative correlation. This is due to the nature of the dataset as there were many records conducted of younger ages compared to older ages.

**BMI:**

Body Mass Index is the relation of body mass and the height of an individual. This can become a factor of an individuals having a high risk of diabetes as if they had a high BMI, they would be considered obese. Being obese has a direct correlation with those that are diabetic. By exploring the BMI data within this dataset, we can verify if this is true and give insight on how medical institutes could lower the risk of diabetes.

**A blue rectangular object with black lines

Description automatically generated**

We can see that the median BMI for those of Pima Indian Heritage is just over 30. This is considered obese, suggesting there is a high risk of most individuals of becoming diabetic. Although this is not a deciding factor, it is one that could contribute in the long run. This provides insight on the population of the Pima Indian Heritage and the average BMI ranges.

This provides valuable insight as medical institutes can setup dietary plans to tackle this factor and hopefully lower the risk of an individual becoming diabetic

**SkinThickness:**

It has been suggested that thick skin on the dorsum of the hands and fingers is a common finding in diabetes, and that affected persons have an increased prevalence of diabetes (Huntley and Walter, 1990).

A violin plot of skin thickness can help in understanding the distribution of this feature among individuals within the dataset.

A graph of a violin plot

Description automatically generated

This suggests distribution of this feature is high when skin thickness values are between 20 and 40. It is also worth noting that it is high when the value is 0, but that is due to the missing values which will need to be imputed.

We can see the outliers of this feature are at the higher end of the skin thickness values, showing a potential subset of individuals with unusually thick skin, which may be indicative of unique physiological characteristics or health conditions that could influence the risk or manifestation of diabetes.

**Outcome:**

**A blue and orange squares with white text

Description automatically generated**

Majority of the dataset isn’t diabetic, as shown by the count plot above. We can also see that the dataset isn’t imbalanced.

**Plotting the distribution of data:**

A group of blue and white graphs

Description automatically generated

The distribution for BP, Skin Thickness and BMI are normal, while for all the other features, it is skewed. Hence for these we can replace the missing values by mean, and for the rest by the median later.

**Multivariate:**

This analysis involves the simultaneous examination and analysis of multiple variables to understand the relationships and interactions between them. It helps researchers and analysts uncover more comprehensive information about the interplay of various factors and their impact on the overall dataset.

**Glucose levels and Outcome:**

In order to find the relationship between glucose levels and the likelihood of someone being diabetic (Outcome) can be represented with a heatmap. A heatmap shows the correlation between the two features as well as the strength. The heatmap is shown below.

A screenshot of a computer screen

Description automatically generated

Here you can see that there is a correlation coefficient of 0.46 which suggests there is a moderate correlation between “Glucose” and “Outcome” in the dataset. This could suggest that the higher the glucose levels in an individual, the higher the likelihood of someone of the Pima Indian heritage being diabetic.

**Age and Outcome:**

Understanding the impact of age on the likelihood of having diabetes is pivotal in identifying age-specific risk patterns and tailoring healthcare plans. To visualize the relationship between age and diabetes outcome we can use a count plot. This will show us information on different age ranges too. We can observe this below.

A screenshot of a computer screen

Description automatically generated

Here we can see that there are many records taken from ages ranging from 21 – 25.This may suggest that the data was collected from a demographic where there is more women from ages 21 – 25, or there was sampling bias where that age bracket is overrepresented.

After observing the ratios of these count plots, we can see that the likelihood of being nondiabetic is far greater than being diabetic. However, as the age increases, we can see that these proportions tend to approach each other, suggesting a potential age-related influence on diabetes prevalence. To further observe this, we can use a heatmap to see how the proportions vary from different age ranges.

While creating a heatmap based on the count of 'Outcome' might introduce age bias, I opted for a more insightful approach. The heatmap presented showcases the proportion of each outcome class, providing a balanced perspective irrespective of age distribution. This is shown below.

A screenshot of a data

Description automatically generated

Here we can see that from earlier ages, the proportion of being nondiabetic far outweighs that of being diabetic. However, as the age increases, the proportion of likely being diabetic increases, thus suggesting that age is directly proportionate to the likelihood of diabetes.

Its important to note that likelihood of being diabetic is extremely high between ages 35 – 63. This can be from other features that have an influence on the outcome feature. It is not clear what these features are exactly by observing this heatmap.

**BMI and Outcome:**

BMI(Body Mass Index) is a value derived from the mass and height of a person. BMI can be used to assess the relationship between body mass index and age. By visualizing this relationship through scatter plots, we can discern patterns that may contribute to a deeper understanding of the interplay between age, BMI, and the likelihood of diabetes.

A screenshot of a computer screen

Description automatically generated

After analysing the Box plots we can see that the box plot for individuals with diabetes is higher on the y axis than the box plot of those without diabetes. This suggests that, on average, individuals with diabetes tend to have higher BMI values than those without diabetes. In the context of BMI, this could imply that those with diabetes, may have higher body weight compared tot hose without. This observation aligns with the understanding that higher BMI is often related with increased risks of developing diabetes.

It is important to note that there are few outliers, some even having a BMI of 0 which is impossible. This is due to missing values when conducting the sampling of this dataset. During the preprocessing stage, we can tackle this by either imputing the missing values, or getting rid of the row completely.

**Pregnancy and outcome:**

Women with several pregnancies may be at an increased risk of developing type 2 diabetes later on in life. This can be due to hormonal changes, insulin resistance, and other factors associated with pregnancy. Below is a bar chart showing the relationship between the amount of pregnancies and the likelihood of diabetes.

**A graph of pregnancy and pregnancy

Description automatically generated**

We can see the proportions of the being diabetic and nondiabetic tend to draw closer to one another as the number of pregnancies increase. It starts off with a much higher proportion of non-diabetics than diabetics when there are 0 – 3 pregnancies. The proportion of diabetic likeliness becomes more prevalent as the pregnancies exceeds 3, suggesting the higher the pregnancies, the likely someone is to become diabetic. We can see the likelihood of someone having diabetes by having 7/8 children is much higher.

As the pregnancies increase, there are less records as it unlikely for large proportion of the population to have a lot of children. A lot of the records are of young individuals which shows why there are so many that don’t have children.

**DPF(Diabetes pedigree function) vs outcome:**

Diabetes Pedigree Function is a positively skewed variable with no zero values. We use the same violin plot to observe the characteristics.

This violin plot will display the distribution of Diabetes Pedigree Function values for each outcome class. The width of the plot at a particular DPF value represents the density of data points, providing an overview of the distribution

A diagram of a person's body

Description automatically generated

The violin plot suggests that there is a noticeable difference in the distribution of Diabetes Pedigree Function (DPF) values between diabetics (Outcome = 1) and non-diabetics (Outcome = 0). Specifically, diabetics tend to have a higher concentration of individuals with elevated DPF values compared to non-diabetics.

Individuals with a family history of diabetes, as indicated by a higher DPF, may be more predisposed to diabetes. This insight could be valuable in understanding the hereditary factors contributing to diabetes risk in the dataset.

**Glucose vs Insulin:**

Before we check the relation, its best we remove the records that have values of 0 as it will only make it harder to analyse the scatter graph.

A black background with white text

Description automatically generated

This gets rid of all records where insulin and glucose values are 0.

Now we can form a relational scattergraph.

A screen shot of a computer screen

Description automatically generated

This scatter graph shows a positive correlation between glucose and insulin levels. It is also worth noting that there are many more individuals that are likely to be diabetic as glucose levels high. This aligns with our hypothesis as Glucose is a main factor that contributes toa higher risk of diabetes.

**Spearman Rank correlation coefficient:**

Spearman rank correlation heatmap can reveal relationships between various features, helping to pinpoint which factors might be more strongly associated with diabetes likelihood and providing valuable insights for further investigation.

A screenshot of a computer screen

Description automatically generated

From the above graph, we can see that the relationship is very strong for the below features:

* Age - Pregnancies
* SkinThickness-BMI
* Glucose-Insulin

We can also see which features have no correlation whatsoever, such as:

* DiabetesPegireeFunction – Pregnancies
* BMI – Pregnancies
* DPF– Pregnancies

Glucose is the feature that has the highest correlation to the “Outcome” variable. This shows it is most likely the most important feature.

**Pair Plot:**

The last part of analysis in the EDA is analysis of a pair plot. This maps pairs of features in one graph, making it easy to spot trends between features.

A screenshot of a graph

Description automatically generated

Here we can see :

* Diabetic women tend to have higher value for each feature i.e. They are older in age, more obese, had more no. of pregnancies, had high BP, glucose and insulin levels.
* Also, if we see the plots of Glucose levels with other features, we see higher Glucose levels are a key factor for diabetes, regardless of the other factories. if the women have higher glucose levels, she is most likely to have diabetes.
* As Skin Thickness-glucose increases, the number of diabetic individuals increases. This suggests glucose is a prominent factor that contributes to the risk of diabetes. This makes sense as it is clinically proven that glucose has a strong correlation to diabetes.

## EDA conclusions

Now that we have completed the EDA, we can consider the questions previously identified in chapter 2.1 and identify any relevant insights that were obtained.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Question** | **Hypothesis** | **Result** |
| Q1 | Does Age increase the likelihood of diabetes? | The likelihood of diabetes increases with advancing age due to age-related changes in metabolism and insulin resistance. | Supported |
| Q2 | Is BMI a decisive factor in predicting the risk of diabetes, and how does it vary across different age groups? | BMI is a significant predictor, and its impact on diabetes risk varies across age groups, reflecting the interplay between body composition and age-related metabolic changes. | Opposed |
| Q3 | Does more pregnancies have a positive correlation with likelihood of having diabetes? | A higher number of pregnancies correlates positively with an increased likelihood of diabetes, potentially influenced by hormonal changes and metabolic adaptations associated with multiple pregnancies. | Supported |
| Q4 | Does age impact the Diabetes pedigree function? | Diabetes pedigree function is affected by age, with genetic factors playing a role in the overall likelihood of diabetes. | Opposed |
| Q5 | Is there a significant association between DPF and the likelihood of diabetes? | Diabetes pedigree function is significantly associated with an increased likelihood of diabetes, highlighting the importance of genetic factors in determining diabetes susceptibility. | Supported |
| Q6 | Do individuals with higher blood pressure exhibit an increased likelihood of diabetes? | Higher blood pressure is associated with an increased likelihood of diabetes, indicating the interconnectedness of cardiovascular health and metabolic disorders. | Supported |
| Q7 | How does blood pressure correlate across various age brackets | The older an individual gets, the more likely they are to have a higher blood pressure. | Supported |
| Q8 | Is Glucose a deciding factor that contributes to diabetes? | Glucose is a deciding factor that contributes to diabetes | Supported |
| Q9 | What is the relationship between Insulin and glucose? | There is a significant and direct correlation between insulin levels and glucose levels | Supported |

# Experimental Design

## Identification of your chosen supervised learning algorithm(s)

The goal of this research is to predict whether an individual is likely to have diabetes or not. This is a Classification-based task therefore a classification-based model was chosen. The following machine learning algorithms were considered before choosing the model:

1. **K-Nearest Neighbors(KNN):** A non-parametric supervised learning classifier that can be used for both classification and regression tasks. It is an instance-based learning algorithm which memorizes the entire training dataset. When making predictions for new data, it looks at the “K” nearest neighbours in the training set and assigns the most common class for classification tasks.

It works well with smaller dataset, which would be convenient as the diabetes dataset is considered small. It is also sensitive to irrelevant features because it relies on the distance between data points to make predictions.

It is considered a “lazy learning” language as it doesn’t make any assumptions about the data distribution during the training phase, and it doesn’t build an explicit model (uk.mathworks.com, n.d.).

1. **Logistic Regression:** As a statistical technique for binary classification, logistic regression is useful in situations where the outcome variable can be classified into two groups, such as true or false, or 0 and 1. Logistic regression, despite its name, is not used for regression, but for classification. This could be a good fit as the target variable is either 1 or 0, likely of being diabetic or not.

The goal is to model the probability that a given input belongs to a particular class. Logistic regression is simple, interpretable, and computationally efficient. It is extensively employed in many different domains for tasks like credit scoring, illness prediction, and spam detection. Although it works well for data that can be divided linearly, more complicated relationships might call for more sophisticated models.

1. **Decision Trees:** A popular machine learning algorithm that can be used for classification and regression tasks. Classification trees give responses that are nominal, such as true or false, whereas regression trees give numeric responses.

It models decisions as a tree-like structure, with each leaf node representing a final decision or prediction, each branch representing an outcome of the test, and each internal node representing a test on a specific feature. Using a decision tree, the dataset is divided into subsets that are progressively more like the target variable.

Decision trees are renowned for being easily visualised and interpreted. The decision-making process's reasoning can be easily understood by visualising them as flowcharts. Decision trees can, however, overfit and capture noise in the training set of data if they are permitted to grow too deeply. A technique like Random Forests is a way to address this problem and improve the resilience of decision tree models (uk.mathworks.com, n.d.).

1. **Random Forests**: The name "Random Forest" originates from the randomness introduced during the training process, which helps prevent overfitting and enhances generalization. It is an algorithm that combines the predictions of several different decision trees to increase overall robustness and accuracy.

A Random Forest's training phase involves building several decision trees using various subsets of the training data and a random subset of features for every split. The individual trees become more variable as a result, increasing their diversity. The predictions of each of the component trees are then combined to determine the Random Forest's final prediction(Donges, 2021).

The ensemble approach enhances predictive accuracy and makes the algorithm more robust to noise in the data. The randomization in feature selection and data sampling helps Random Forests overcome the overfitting tendency of individual decision trees(Team, 2020).

1. **Naïve Bayes:** Naive Bayes is a probabilistic machine learning algorithm based on Bayes' theorem, particularly suitable for classification tasks. It is considered "naive" because it assumes that features are conditionally independent given the class, even though this assumption might not hold in real-world scenarios. This algorithm usually performs well especially in text classification and spam filtering applications(Simplilearn.com, n.d.).

Due to the simplicity of this algorithm, it doesn’t overfit the data and can be trained very quickly, also making is a good choice when CPU and memory resources are limited.

This algorithm calculates the probability of a specific class given the values of the input features. The class with the highest probability is then assigned as the predicted class. It is computationally efficient and has low training time, making it useful for large datasets. It handles categorical and numerical features and requires a relatively small amount of training data to estimate parameters accurately (u k.mathworks.com, n.d.).

I have chosen to go with K-nearest neighbour (KNN) for my classification task as its suitable for small datasets. Since my dataset doesn’t have a large amount of data for training, this algorithm is better suited. There’s also no training phase as it is an instance based algorithm, and it has a non-parametric nature which allows for flexibility with smaller datasets.

## 

## Identification of appropriate evaluation techniques

When evaluating machine learning models, it is critical to choose the right metrics and evaluation methods. We have carefully selected a set of evaluation metrics for our approach that are specific to the features of our dataset and problem. Accuracy, precision, recall, F1-score, and the confusion matrix are among the selected metrics.

Accuracy is a fundamental metric that provides a high-level overview of the model's correctness. However, its suitability diminishes in the presence of imbalanced datasets. To address this, we incorporate precision and recall metrics.

By taking the mean of the two metrics, we use the F1-score to achieve a balance between recall and precision. This is useful when there is an uneven distribution of classes, requiring a good understanding of the trade-off between false positives and false negatives. Along with these metrics, we use the confusion matrix to give us the details of the true positives, true negatives, false positives, and false negatives. This insight allows us to fine-tune the model, and crack down on specific errors.

In conclusion, the chosen set of evaluation metrics collectively offers a comprehensive and sophisticated picture of the model’s performance. By considering these metrics, we can gain valuable insights to informed decisions.

## Data cleaning and Pre-processing transformations

Data cleaning:

After analysing the dataset, I could see that the “Insulin” and “Skin thickness” features have the highest amounts of records with values of “0”. This means these values weren’t collected during the conduction of the dataset. We can see the amount of values that are 0 for both columns below:

A computer screen shot of a computer code

Description automatically generated

Here we can see the number of 0 values are very high, almost half of our dataset. Getting rid of these rows isn’t practical as it will remove a lot of valuable data which would be useful to our machine learning model.

To clean this data, I decided to impute the missing values for insulin. I decided to use the mean value to be placed into all of the missing rows which was the value 155.

A computer screen with text on it

Description automatically generated

Once the 0 values were replaced with “nan”, they were then replaced with the mean value of the insulin and reshaped into the dataset.

We can check if this was successful.

A screenshot of a computer screen

Description automatically generated

The imputation was successful and all 0 values are replaced with mean values.

Now regarding the skin thickness feature, we have seen that it doesn’t have much of an affect on the target variable therefor I have chosen to drop the feature.

A screenshot of a computer

Description automatically generated

This column is no longer in the database. This is known as “feature elimination” or “feature dropping”. This can be beneficial in many ways as not only does it improve model performance due to focusing on less features, but it also gets rid of the missing values that would hinder the model performance. Additionally, it results in a smaller dataset which can lead to faster training and quicker predictions (kortical.com, n.d.).

**Outliers:**

Removing outliers is important when using KNN because they can disproportionately influence the distance calculations, potentially degrading the performance of the model.

An example, there can be an outlier when “Age” and “Pregnancies” come together as someone aged 17 with 5 pregnancies is unlikely. This is known as an outlier row.

To tackle this, the Local outlier factor will be considered, then the ”elbow method” will be implemented to determine threshold.

A screenshot of a computer program

Description automatically generated

Note that negative scores are given but can be changed to positive.

A screen shot of a graph

Description automatically generated

Observation scores on the x-axis and outlier scores on the y-axis. You can see the most marginal change is the third index, there is a slope.

A screen shot of a computer

Description automatically generated

Solution of outliers has now been complete with this process.

**Splitting the dataset:**

A screen shot of a computer

Description automatically generatedThe dataset can now be split, resulting in two training sets and two testing sets. One set contains the input feature set, with the other containing the target results. Here is how this was done.

With the use of “train\_test\_split” function , it does this for us. To verify this has been done successfully, we can check the shape of the testing set. From the snippet above, the testing set makes up 20 % of the rows from the entire dataset which was what we wanted.

**Data Scaling:**

Now that the missing values are handled, scaling numerical features would be the next step. This is especially important for algorithms sensitive to feature scales, such as KNN.

Scaling our dataset is necessary as when training a K-nearest neighbours(KNN) model because the algorithm relies on the concept of distance between data point. If the features in our dataset are on different scales, certain features might contribute more to the distance calculation than others(Filho, 2023.

This ensure all features are contributing equally to distance metric. Common scaling techniques are min-max scaling and standard scaling, both can be used by importing.

In this case , I chose min-max scaling as it transforms the features of the dataset to a common range between 0 to 1.

Additionally, it can help with interpretation of the model coefficients as well as speed up convergence, thus resulting in quicker training time if we were to use logistic regression.

Since I will be using the KNN algorithm, this step is essential.

**A screenshot of a computer program

Description automatically generated**

All data in columns are scaled apart from the target variable. This can be used for training our KNN model.

Just to check if this was done correctly, we can print the top 5 rows of the scaled sets.

A screenshot of a computer program

Description automatically generated

Here we can see at the data values have been scaled correctly as they all range between 0 and 1.

**Encoding:**

Due to the nature of the dataset, encoding isn’t necessary as all features are numerical. Although the target variable (“Outcome”) is categorical, it is represented with binary classifiers. Machine learning models typically handle numerical features directly without the need for encoding.

By ensuring all data within the dataset is numerical, and the outcome variable is binary, additional encoding features will not be needed.

For example, if our outcome feature had several classes, like low, medium, and high, then data encoding would be necessary as the values aren’t numerical. A common encoding method we would use in this instance would be to assign integer values according to their order. So it could be mapped as “low” for 0, “medium” for “1” and “high” for “2”.

A screenshot of a computer

Description automatically generated

Here we can see the data has been scaled, we just need to add unscaled target variable column to this data frame.

## Limitations and Options

While imputation, scaling, and feature dropping techniques were used in the pre-processing, there are potential limitations to be aware of.

Imputing any missing values may introduce bias, especially is the data is not missing completely at random. The imputed values might not accurately represent the true underlying distribution (Kline, 2022). We must also make sure we also don’t impute values that are meant to be 0, as it is normal for some features to have this, like “pregnancies”.

Dropping a feature can lead to a loss of valuable information, especially in a small dataset. Each feature can play an important role in capturing patterns. This could oversimplify the dataset and negatively impact the model.

Scaling can be sensitive to outliers, and since we have a small dataset, it can make a significant impact on the scaling process, potentially distorting overall distribution thus leading to a less accurate model.

# Predictive Modelling / Model Development

## The predictive modelling process

The pre-processed data, which includes missing values, scaling features,a dropped feature (“Skinthickness”),and split data, was used to train the model.

The Scaled and filled training set(“X\_train\_scaled\_filled”) was used to train the KNN model.

A black background with white text

Description automatically generated

The parameter specified for the KNN model is “n\_neighbours”, where the value is set to 5. This determines the number of neighbours the algorithm considers when making predictions. In this case, the KNN classifier will classify a data point based on the majority class of its five neighbours.

The feature matrix ” X\_train\_scaled\_filled” encapsulates the training set after Min-Max scaling and filling missing values. Simultaneously, the corresponding labels for the training instances are represented by “y\_train\_filled”.

This enabled the KNN model to grasp patterns and relationships within the training data, allowing for it to make predictions on new, unseen data.

## Evaluation results on “seen” data

In the evaluation of out model on the training set, the primary focus is on understanding how well the model has fit the training data and captured its patterns

A screenshot of a computer program

Description automatically generated

Results of the accuracy and a detailed classification report for the training set can be seen above. We can see that there is 83.88% accuracy. This means that the model correctly predicted the class of approximately 83.88% of the instances in the training set, which is considered Acceptible.

We can also see a classification report on the training set which provides additional metric, such as precision, recall, f1-score, and support, as well as the weighted average.

The target variable(Outcome) has 2 classes.

For class 0.0:  
 -Precision: 86% of predicted positives were true positives

-Recall: 90% of actual positives were correctly predicted

-f1 Score: The balance between the two metric above was 88%

For class 1.0:

-Precision: 79% of predicted positives were true positives

-Recall: 72% of actual positives were correctly predicted

-f1 Score: The balance between the two metric above was 76%

Here is a heatmap of the confusion matrix:

A graph of a diagram

Description automatically generated with medium confidence

We can see that the amount of true negatives predicted was 361 instances and the amount of true positive instances predicted was 154. The amount of false positive instances was 40 and false negatives was 59. This means the accuracy of the model is around 83% which is decent. Having perfect results would be an indicator of overfitting. Fortunately, this isn’t the case.

Now we test the how well the model performs on “unseen” data.

# Evaluation and further modelling improvements

## Evaluation results on Testing data

Having tested the model on the training dataset, the model was then tested on the testing dataset. Since the resting data is unseen, we can see how the model performs on a new set of data.

A screenshot of a computer program

Description automatically generated

After applying the model to the testing data, we can see the results. The model had a 74% accuracy rate. From the classification report, we can see the precision, recall, f1-score and support metrics, giving us a more in depth visualisation on the performance of our model on this dataset.

The model shows reasonable accuracy, but there is a trade-off between precision and recall, particularly for class 1. The model tends to perform better for class 0, where precision, recall, and F1-score are higher. However, for class 1, recall is relatively lower, indicating that there were less instances for this class during the training phase. A way to improve this result would be by training the model again but with more instances for class 1.

## Further Modelling Improvements

**Hyperparameter tuning and cross-validation**

K-Nearest neighbour (KNN) uses the parameters “n\_neighbours” (default = 5). This parameter defines the number of neighbours to consider when making predictions. We can tune this parameter to find which number of neighbours leads to the best performing model. By also using the technique, cross validation, we can split the dataset into different multiple folds where each are tested iteratively.



A screenshot of a computer program

Description automatically generated

Using the “grid\_search.fit” class performs a search over the “n\_neighbours” array and evaluates the model performance for each of the different hyperparameters using cross-validation. As you can see, the best performing parameter for “n\_neighbour” is “13”. After using this parameter and re-running our model, we get:

A screenshot of a computer

Description automatically generated

The model had an increase of approximately 0.6% efficiency. Although this seems very low, it isn’t uncommon due to the nature of the dataset and algorithm.

We can see that the recall metric for class 0.0 has increased by since our initial model, however, it has decreased for class 1.0. This could be due to the shift in model sensitivity towards each class. An increase in recall for class 0.0 suggests that the model is now better at identifying instances of this class, reducing the number of false negatives. On the other hand, the decrease in recall for class 1.0 shows that there may be missing instances in this class.

I believe a way of further improving this model is by making the classes more balanced, as the dataset originally had many more values of “0” than ”1”.

A way we can further improve the model even more is by:

**Feature engineering:**

During the data pre-processing phase, I decided to drop the “SkinThickness” feature as It had too many “0” values and I felt getting rid of it wouldn’t make too much of a difference.

This time around, I plan to keep the feature and impute the missing values in hopes that it’ll improve model performance since it provides more data.

Imputing the missing values for glucose will make a significant difference on the model as during our EDA, we found that glucose is the feature that contributes the most to our target variable.

I also now realised the target feature (Outcome) was imbalanced. There were many more instances for class “0” than f or class “1”, giving one class more influence over the other. I can apply oversampling to the minority class to match the same number of instances as the other class. This will provide the model more information on the minority class whilst also making both classes have an equal influence on the model.

Here is the process:

Imputing all missing data:

A screenshot of a computer program

Description automatically generated

Here you can see all the columns that shouldn’t have any “0” values are analysed and imputed with mean data from their corresponding columns. We can see this was successful as there’s no more 0 values for these columns.

Applying Min-Max Scaler:

A computer screen with white and green text

Description automatically generated

This enhances model interpretability and equalizes the feature magnitudes. Since KNN is a distance-based algorithm, features with lager magnitudes can dominate the distance metric, potentially influencing the model incorrectly.

Applying oversampling on the outcome class “1”:

A screen shot of a computer program

Description automatically generated

The classes are now balanced , 500 instances for each.

Now the pre-processing is complete we can train and test the model once again.

After finding the most efficient hyperparameter from before, “n\_neighbour = 13”, we can put this as out value for the training phase.

A black background with green and white text

Description automatically generated

**Results:**

A screenshot of a computer

Description automatically generated

The snippet above shows the accuracy of our model prediction is now 79.5% which is almost 5% more efficient than our previous model improvement.

## Final evaluation results

## Accuracy table

|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| First model | 0.741 |
| Second model improved (Hyperparameter) | 0.747 |
| Third model improved (Hyperparameter + oversampling + feature inclusion) | 0.795 |

The baseline model with no improvement was 74.1% accurate. The second improved model had a higher accuracy of approximately 0.6%. This is because there was only one adjustment made to the model, which was altering the n\_neighbour parameter.

Then with the third mode, we made two extra adjustments on top of our second model. Including our “skinthickness” feature, oversampling the target class, and imputing all 0 values, resulted in a 5% increase in prediction accuracy.

I am happy with this result as these adjustments made a significant improvement to out model. This shows the importance of the preprocessing phase and emphasizes how we clean our data makes an effect on the accuracy of our model.

Classification metrics table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Recall**  **(Class 0)** | **Recall**  **(Class 1)** | **Precision**  **(Class 0)** | **Precision**  **(Class 1)** | **F1-Score**  **(Class 0)** | **F1-Score**  **(Class 1)** |
| First model | 0.87 | 0.46 | 0.77 | 0.63 | 0.81 | 0.53 |
| Second model(Hyperparameter) | 0.89 | 0.46 | 0.77 | 0.67 | 0.82 | 0.55 |
| Third model(Hyperparameter + oversampling + feature inclusion) | 0.73 | 0.86 | 0.84 | 0.76 | 0.78 | 0.81 |

The metrics table presents a comparison of model performance across three iterations. In the first model, a baseline without specific enhancements, we observe a trade-off between precision and recall for class 1, indicative of the model's difficulty in correctly identifying positive instances.

The second model, with hyperparameter tuning, shows an improvement in recall for class 1, suggesting a better ability to capture positive instances. However, precision for class 1 continues to be moderate.

The third model, incorporating hyperparameter tuning, oversampling, and feature inclusion, demonstrates the most balanced performance, showcasing notable improvements in recall, precision, and F1-score for class 1.

Confusion matrix table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **TP** | **FP** | **TN** | **FN** |
| First model | 12 | 7 | 14 | 46 |
| Second model(Hyperparameter) | 12 | 6 | 14 | 47 |
| Third model(Hyperparameter + oversampling + feature inclusion) | 87 | 27 | 14 | 72 |

Here we can see how many instances each model got correct , and how many it got incorrect. We can see the third model did significantly better than the other models.

However, it is important to note that this isn’t as big of a difference as it may look. The numbers are much higher for our third model because we oversampled out minority class which in turn gave us a bigger testing set to work with. With that said, the third model is still more accurate than the others.

# Conclusion

## Summary of results

In concluding the model development process, we have gone through the process of defining a problem within a dataset that we wanted to fix and predict, to then refining a classification model with 79.5% accuracy. The overarching goal was to predict outcomes in a binary classification scenario, with a focus on enhancing the model's performance iteratively. There is still a lot to learn about the model and how to improve it as there are other parameters that can be tweaked.

The tuning process made for an iterative approach on improving model accuracy. This process is a good way of understanding how the KNN model interprets data and how we should manipulate the data we have so we can produce the best predictive model.

The final model produced a good score, indicating it can fulfil its intended role to predict whether a person is likely to have diabetes or not depending on their attributes. In terms of real world, we can use this model to predict the likeliness of individuals of the Pima Indian heritage of being diabetic. It will provide medical institutes with insight on how they can keep the likeliness as low as possible.

From the EDA, we found that Glucose is the most important feature that contributes to the Outcome feature. This tells us that regulating glucose levels is key as it contributes most to the risk of being diabetic.

## Suggested further improvements to the model development process?

A way we could possibly improve the model development process could be by introducing feature importance for the glucose feature. Since this feature had the biggest impact on the outcome, we can improve the model interpretability further as well as focusing on key variables driving the model’s decisions.

## Reflection on Individual Learning

As I was going through this coursework, I learned a lot about different machine learning algorithms and when best to use them. Going through the EDA helped me understand my chosen dataset on a deeper level and equipped me with the skills required to exploring any other datasets. I can now analyse what parts of a dataset can hinder model performance and adjust them accordingly via data pre-processing. These are all new skills that I never had before this module.

For me, the longest and trickiest part was the EDA at first as I didn’t know how to approach univariate and multivariate features since it was my first time encountering a dataset with 9 features. The more I progressed throughout the EDA phase, the more confident I became at visualising the data and finding imbalances or values that didn’t make sense.

The part I enjoyed most was the modelling process, as it was quite straight forward and practical. Building a machine learning model was fun and testing how efficient it was felt rewarding.

Another part that took time and effort was picking which algorithm to use as there was a few that I could choose from. Having to research the pros and cons of each and applying it to the context of the dataset was vital as picking your specific model is vital to produce a valid result.

To improve my skills in the future, I would repeat the process of this assignment with another dataset that is a regression task. This will allow me to work with regression models, a whole section I didn’t get to explore in this assignment. I would also like to play around more with the modelling as I believe I could’ve developed a more efficient ML model. During the modelling phase of this coursework, the coding was a bit scruffy since it took a while to grasp the concepts of what I was doing.

# References

* www.kaggle.com. (n.d.). Diabetes Dataset. [online] Available at: <https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset>.
* Laila, U. e, Mahboob, K., Khan, A.W., Khan, F. and Taekeun, W. (2022). An Ensemble Approach to Predict Early-Stage Diabetes Risk Using Machine Learning: An Empirical Study. Sensors, 22(14), p.5247. doi:https://doi.org/10.3390/s22145247
* Sharma, G. (2021). 5 Classification Algorithms you should know - introductory guide. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2021/05/5-classification-algorithms-you-should-know-introductory-guide/#:~:text=There%20are%20many%20different%20classification>
* RapidMiner. (n.d.). ML Classification vs Regression. [online] Available at: <https://rapidminer.com/glossary/ml-classification-vs-regression/#:~:text=Classification%20models%20are%20used%20to>.
* CDC (2020). Diabetes Risk Factors. [online] Centers for Disease Control and Prevention. Available at: <https://www.cdc.gov/diabetes/basics/risk-factors.html#:~:text=You>
* Gupta, S. and Bansal, S. (2020). Does a rise in BMI cause an increased risk of diabetes?: Evidence from India. PLOS ONE, 15(4), p.e0229716. doi:https://doi.org/10.1371/journal.pone.0229716
* Liu, B., Song, L., Zhang, L., Wang, L., Wu, M., Xu, S., Cao, Z. and Wang, Y. (2020). Higher Numbers of Pregnancies Associated With an Increased Prevalence of Gestational Diabetes Mellitus: Results From the Healthy Baby Cohort Study. Journal of Epidemiology, 30(5), pp.208–212. doi:https://doi.org/10.2188/jea.je20180245
* Huntley, A.C. and Walter, R.M. (1990). Quantitative determination of skin thickness in diabetes mellitus: relationship to disease parameters. Journal of Medicine, [online] 21(5), pp.257–264. Available at: <https://pubmed.ncbi.nlm.nih.gov/2079616/#:~:text=It%20has%20been%20suggested%20that>.
* uk.mathworks.com. (n.d.). Choosing the Best Machine Learning Classification Model and Avoiding Overfitting. [online] Available at: <https://uk.mathworks.com/campaigns/offers/next/choosing-the-best-machine-learning-classification-model-and-avoiding-overfitting.html>.
* Donges, N. (2021). A Complete Guide to the Random Forest Algorithm. [online] Built in. Available at: <https://builtin.com/data-science/random-forest-algorithm>.
* Team, G.L. (2020). Random Forest Algorithm- An Overview | Understanding Random Forest. [online] GreatLearning. Available at: <https://www.mygreatlearning.com/blog/random-forest-algorithm/>.
* uk.mathworks.com. (n.d.). Choosing the Best Machine Learning Classification Model and Avoiding Overfitting. [online] Available at: <https://uk.mathworks.com/campaigns/offers/next/choosing-the-best-machine-learning-classification-model-and-avoiding-overfitting.html>
* Simplilearn.com. (n.d.). Understanding Naive Bayes Classifier. [online] Available at: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/naive-bayes-classifier>
* kortical.com. (n.d.). Dropping Features Early: Smart, Time Saving and Efficient or Amateur, Bad Practice and Lazy? | Article. [online] Available at: <https://kortical.com/notebooks/dropping-features-early>
* Filho, M. (2023). Does Logistic Regression Require Feature Scaling? [online] forecastegy.com. Available at: <https://forecastegy.com/posts/does-logistic-regression-require-feature-scaling/>.
* Kline, A. (2022). Implementation and Limitations of Imputation Methods. [online] Medium. Available at: <https://towardsdatascience.com/implementation-and-limitations-of-imputation-methods-b6576bf31a6c>