

Pima Diabetes Data Analysis Report

Gloria Hawkins-Roberts

Media Design School

2023

DATA ANALYSIS

Definition and description of the project topic.

The project topic I have selected is Pima diabetes. As a leading non-communicable disease, diabetes Mellitus (commonly known as diabetes) affects individuals whose blood glucose is too high (Kharroubi & Darwish, 2015). The reason for this is entirely individual, hence the three primary types of diabetes: type 1 diabetes is caused by the pancreas' inability to react to insulin(JDRF UK, 2021), it is thought to be caused by an autoimmune reaction where the body attacks itself (Kharroubi & Darwish, 2015). On the other hand, there is type 2 diabetes which is caused by a built-up resistance to insulin due to lifestyle factors and the third common type is gestational diabetes (diabetes during pregnancy) (Kharroubi & Darwish, 2015). However, it is important to note that there are around 12 identifiable types of diabetes. Though each of these types has its own set of risk factors, the most common type of diabetes is type 2 which affects around 90% of people with diabetes (Kharroubi & Darwish, 2015). Risk factors for type 2 diabetes include (but are not limited to); being overweight, 45 years or older, family history of diabetes and having a history of gestational diabetes (Kharroubi & Darwish, 2015). Symptoms of diabetes can be life-threatening, some of these symptoms include high blood pressure, risk of heart disease, blood vessel damage and more(Kharroubi & Darwish, 2015). The Pima diabetes data set does not focus on a particular type of diabetes, but rather on some of the predictor variables for general diabetes. The data set focuses on Pima Indians who are a native american group mainly residing in Arizona and Mexico. Pima Indians have a particularly high incidence of diabetes, this has spiked an interest in research.

Explanation of why this is an appropriate problem to attempt with an AI.

The problem of Pima diabetes is appropriate to attempt with AI. As mentioned, diabetes can lead to many other life threatening health issues, this is something that can be prevented with the help of AI. Machine learning can aid in the diagnosis of Diabete and can allow professionals to identify patterns, quickly enter patient data, and predict the likelihood of a female patient having diabetes in general. For example, type 2 diabetes has mild symptoms at the beginning, these symptoms may go unnoticed leading to long-term complications(Kharroubi & Darwish, 2015), AI can be implemented to help physicians easily predict if a patient is likely to have diabetes based on a set of variables and training data. The use of AI has the potential to identify undiagnosed or less diagnosed patients, and reduce the incidence of serious complications.

Description of the source of the project data set.

I was able to source this data via Kaggle which is an online science platform and community of scientists and recognised datasets. However, the dataset was originally donated to the National Institute of diabetes and Digestive and Kidney diseases (better known as NIDDK). NIDDK is a part of the United States National Institutes of Health (NIH) which acts as the nation's medical research agency (*National Institute of Diabetes and Digestive and Kidney Diseases, 2021*). The NIDDK was first discovered by the NIH in 1950 as the National Institute of Arthritis and Metabolic diseases, 1986 saw the official name change to what is known today as the NIDDK (*National Institute of Diabetes and Digestive and Kidney Diseases, 2021*). As a branch of the NIH, NIDDK carries out medical support and research on diseases such as diabetes and other endocrine and metabolic diseases (*National Institute of Diabetes and Digestive and Kidney Diseases, 2021*). NIDDK's main objective is to improve the lives of those suffering from these (often) long-term (or chronic) diseases (*National Institute of Diabetes and Digestive and Kidney Diseases, 2021*).

Description of the project data set, including the number of variables, number of examples, and descriptive statistics where appropriate.

As I have mentioned, the data set focuses on Pima Indians. However, more specifically female Pima Indians. This has been an ongoing study since 1965. There are nine variables included in the dataset, independent variables being; number of pregnancies (number of times pregnant), plasma glucose level (concentration of plasma glucose, 2hr oral glucose tolerance test), blood pressure (diastolic blood pressure, mm Hg), skin thickness (mm), body Mass Index (BMI, weight in kg/height in metres), diabetes pedigree function (family history of diabetes), Age (years), Insulin (2-hr serum (U/ml), and the dependent variable: outcome of diabetes (1:yes/0:no). The data set includes 768 samples, all of which are at least 21 years of age, from a population living near Phoenix, Arizona USA. 500 of these samples do not have diabetes and 268 do, this means that this is an imbalanced dataset.

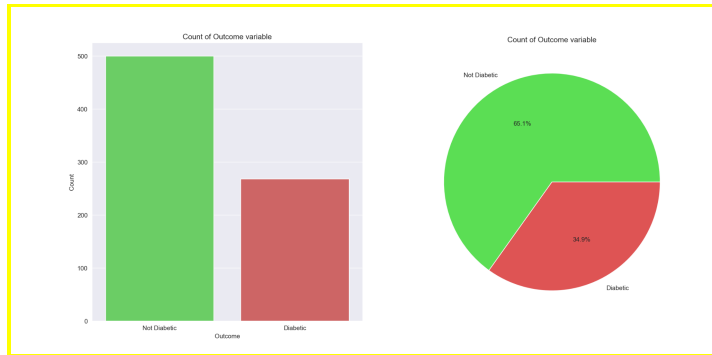
Description and application of, and justification for, data visualisation techniques. This will involve specifying and justifying the visualisation techniques used, and producing visualisations using those techniques.

In order to produce my visualisation techniques, I have used the libraries seaborn and matplotlib.

The first data visualisation techniques I have implemented are a count plot and a pie chart (F.1). The reason I have chosen to use both of these data visualisation techniques is to represent the categorical dependent variable of "outcome". I have chosen to use a pie chart as it represents data visually as a fractional part of a whole, this is a type of data visualisation technique that most people are familiar with and it is easy for an uninformed audience to interpret. The pie

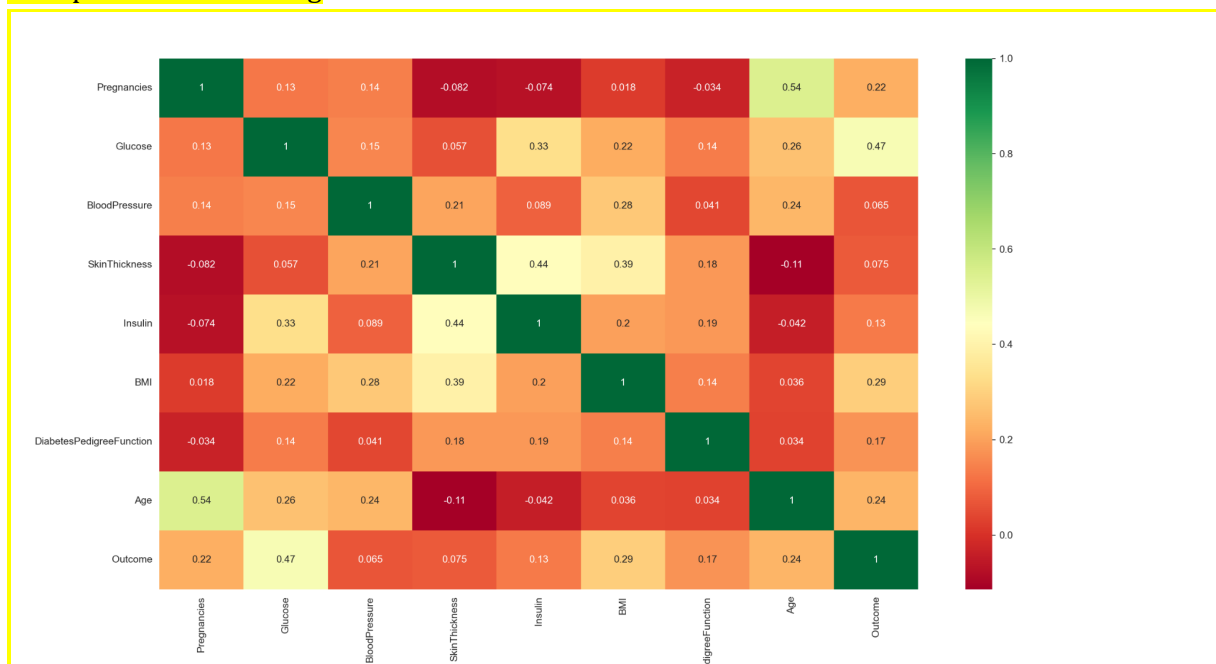
chart is advantageous for the representation of my “outcome” variable as it is easy for the audience to make quick analysis, it is familiar, and simple.

I am only comparing two categorical outcomes, having diabetes or not, thus, a pie chart is an appropriate measure for visualising these two outcomes.

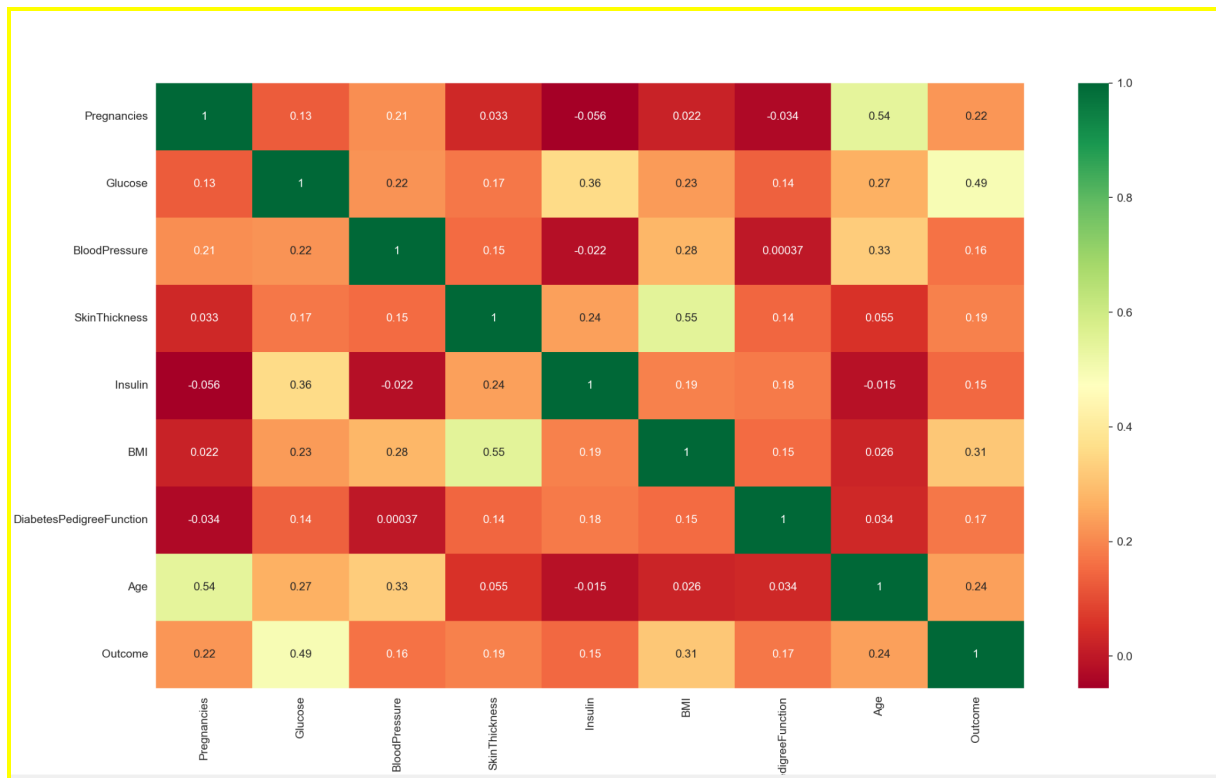


I further decided to use a count plot, this is effective for showing proportions of categorical data. The reason why I have chosen to include this is to aid in visually representing a number as well as a fraction to the audience. It provides a more in depth look at the categorical “outcome” for the data, while still maintaining simplicity and easy interpretation of the data. Both the countplot and the pie chart show that the data is unbalanced.

Heat plot Before cleaning



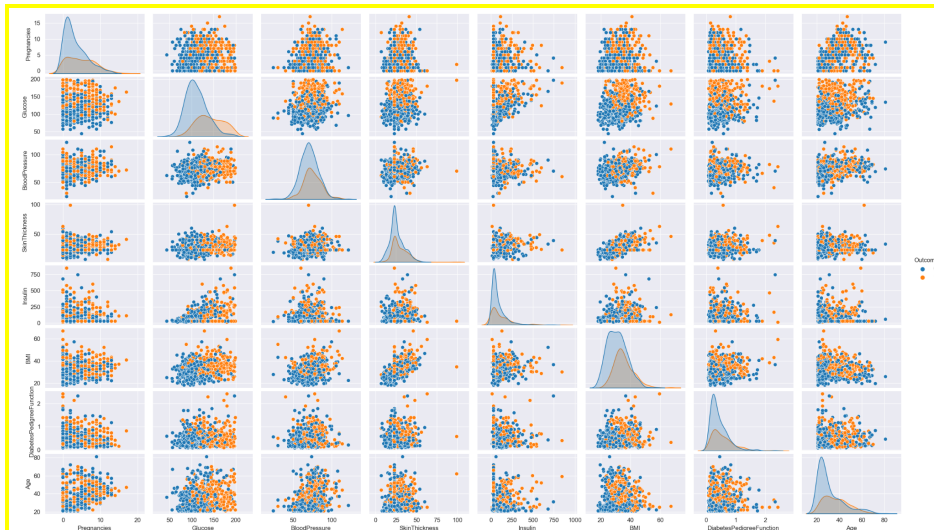
Heatmap/ correlation plot after cleaning



I have decided to use a heatmap to visualise correlations between the independent variables and the dependent variable. The green hue indicates a high correlation and the red hue indicates a low correlation. From both heatmaps (F.2, F.3), it can be seen that there does not seem to be many strong relationships between the variables. Some notable correlations within the matrix include; BMI and skin thickness (0.55) and glucose and insulin (0.36). Although the correlations to outcome are all relatively low, glucose and BMI have the strongest. The outcome seems to be most dependent on glucose, however, which could be a result of insulin's correlation to glucose. The effect of insulin on glucose could be a hidden pattern in this data set. Another notable correlation is between age and pregnancy (0.54), however this is irrelevant and is an obvious correlation in the general world.

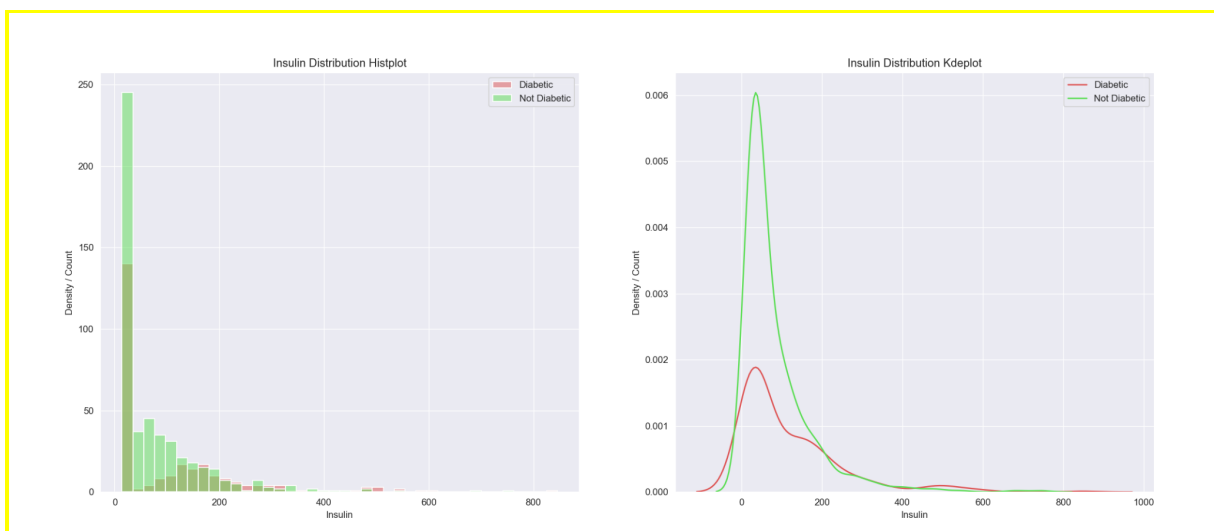
The heatmap is an effective tool to quickly and easily visualise correlations between independent and dependent variables. The colours on the map make this an effective technique for those unfamiliar with the dataset and aid in easy interpretation.

All Features Pair Plot:



I have decided to use an all features pair plot as a more indepth visualiser to the heat map. Though the heat map is easy to interpret for correlation, all it is telling us is a number and not allowing us to visualise the data. The all features pair plot is effective because it allows the viewer to compare multiple independent variables against each other and against the dependent variable. It is easy to see from the plot that Glucose is a predictor variable that shows a high and consistent correlation to diabetes when compared alongside other variables. This acts as a useful starting point for further investigation into the associations between variables.

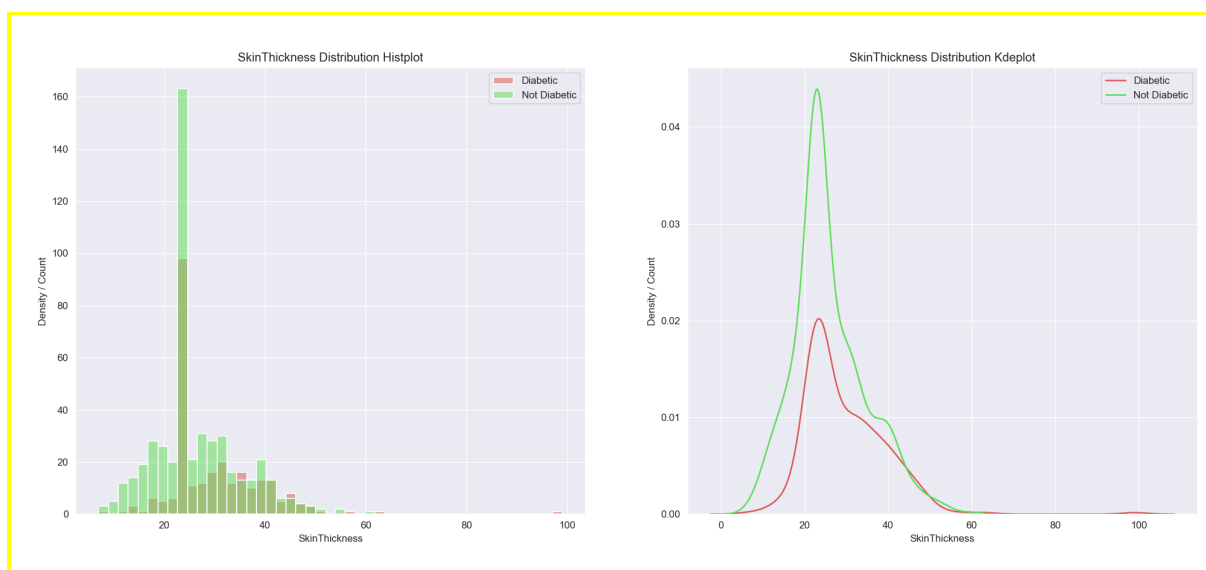
The following plots look into variables against the outcome at an individual scale. I have used distribution plots/ histograms and Kde plots because together they compliment each other. They offer two visual approaches to analysing the data. Histograms are useful for easy comparison of data and the same goes for Kde plots. Kde plots are also very effective for continuous variables, this is why I have chosen to use them for my independent variables as they are continuous.



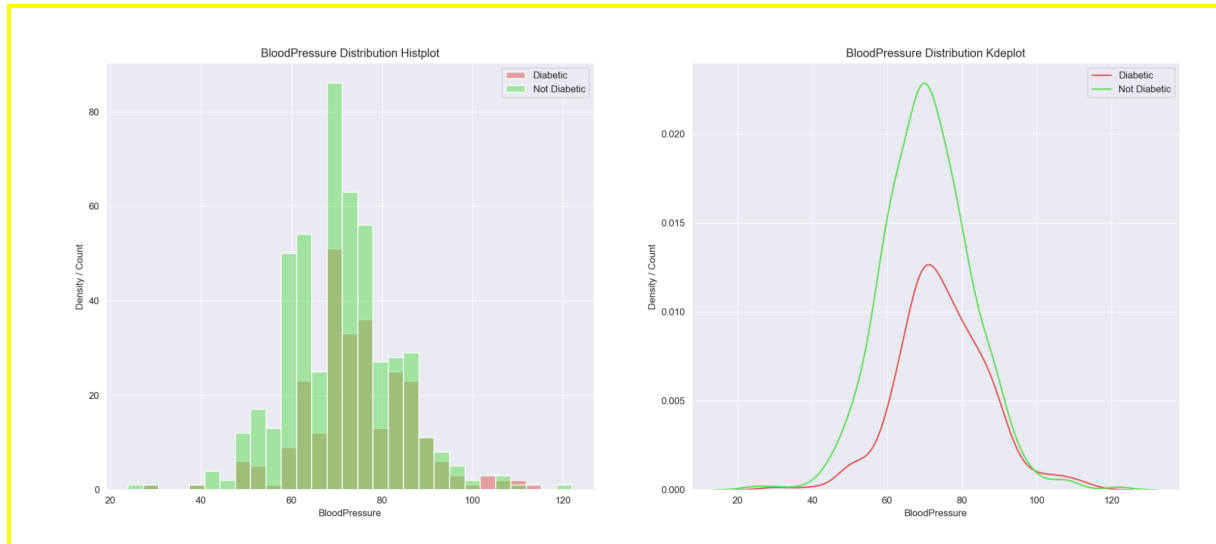
The histogram and the kde plot show that the distribution of data for insulin is skewed to the left. This also shows us that insulin doesn't have a high correlation to the outcome, but that doesn't mean that it isn't indirectly affecting the outcome of having diabetes or not.



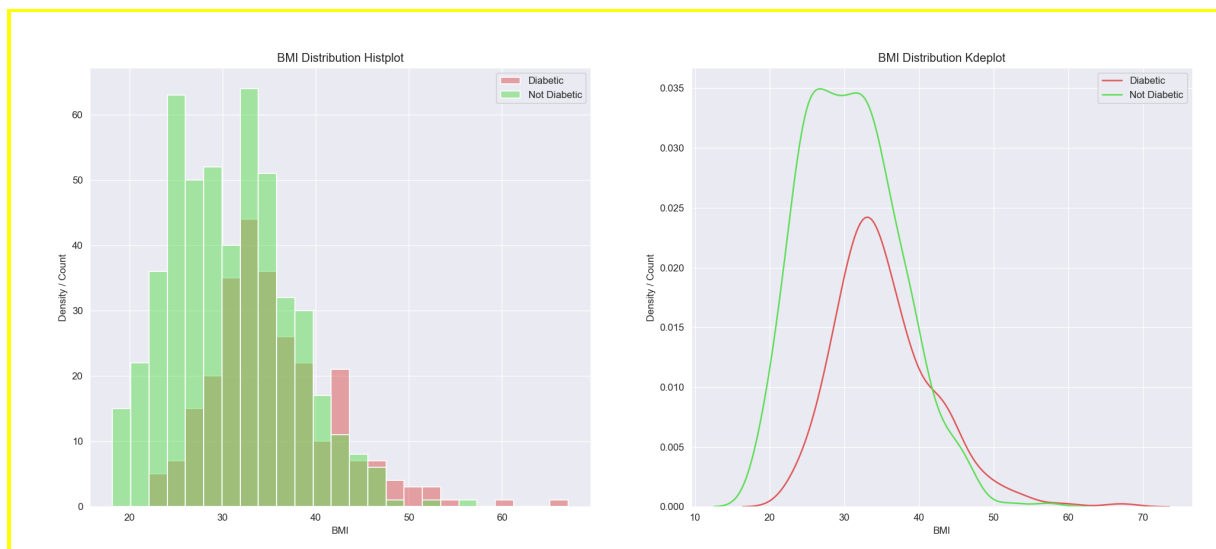
The histogram and Kde plot show that glucose has a slightly positive skewed distribution of not diabetic outcomes with lower numbers of glucose and a slightly negative distribution for higher numbers of glucose. This indicates that there is a correlation between glucose and outcome, these plots allow the viewer to see this relationship in more depth rather than just visualising the numbers on the heatmap.



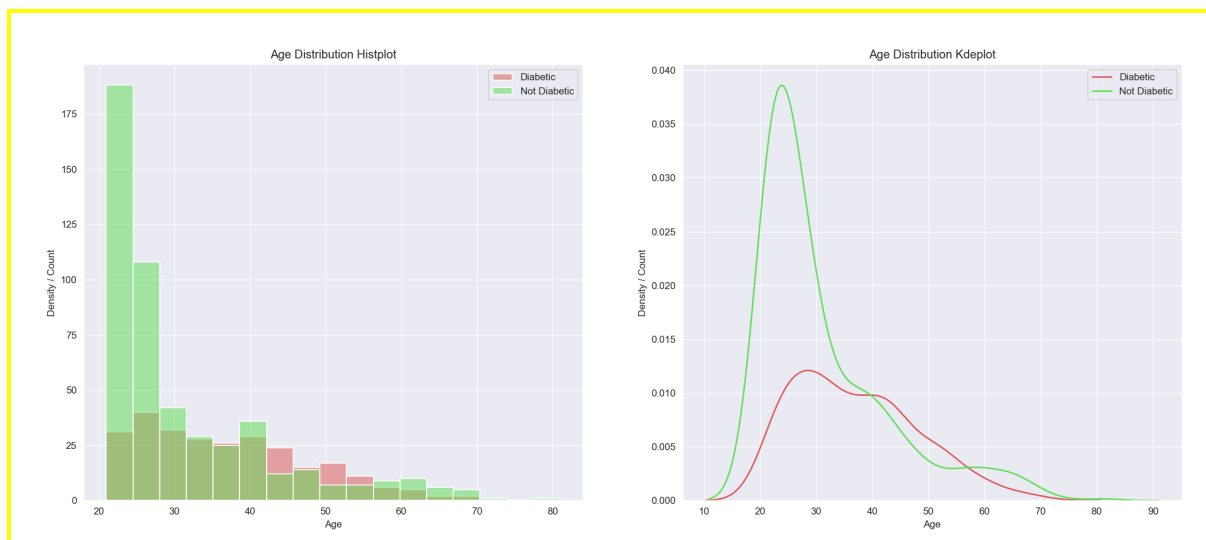
Skin thickness has a general skew to the left (positive distribution), the distribution of data between diabetic and not diabetic is very similar.



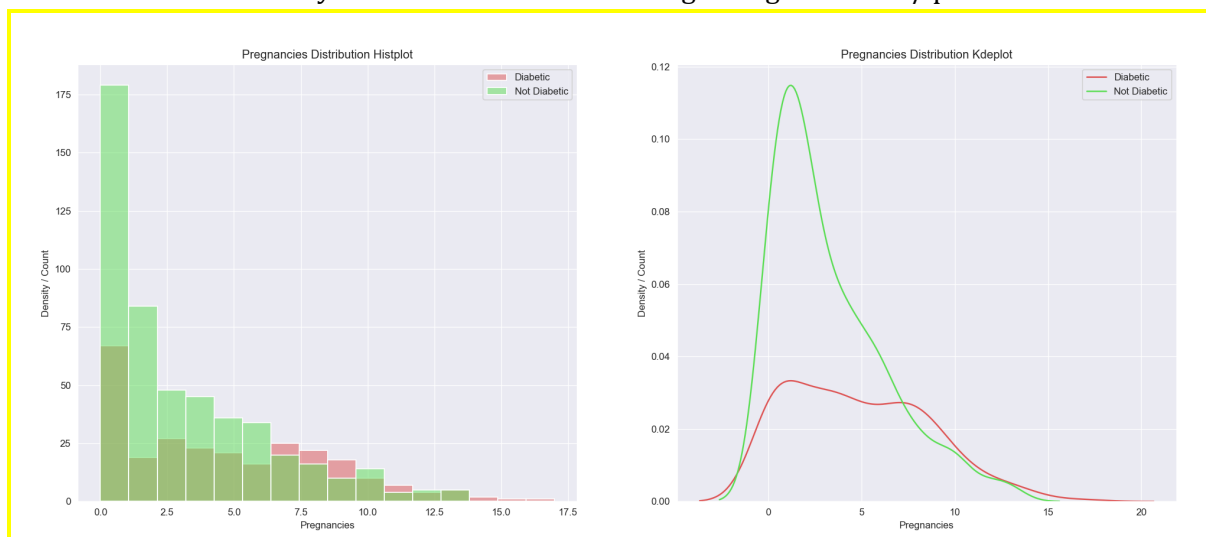
The distribution between diabetic and not diabetic for blood pressure again seems very similar. This does not rule it out as a factor. The data does not appear to be skewed to either the left or the right.



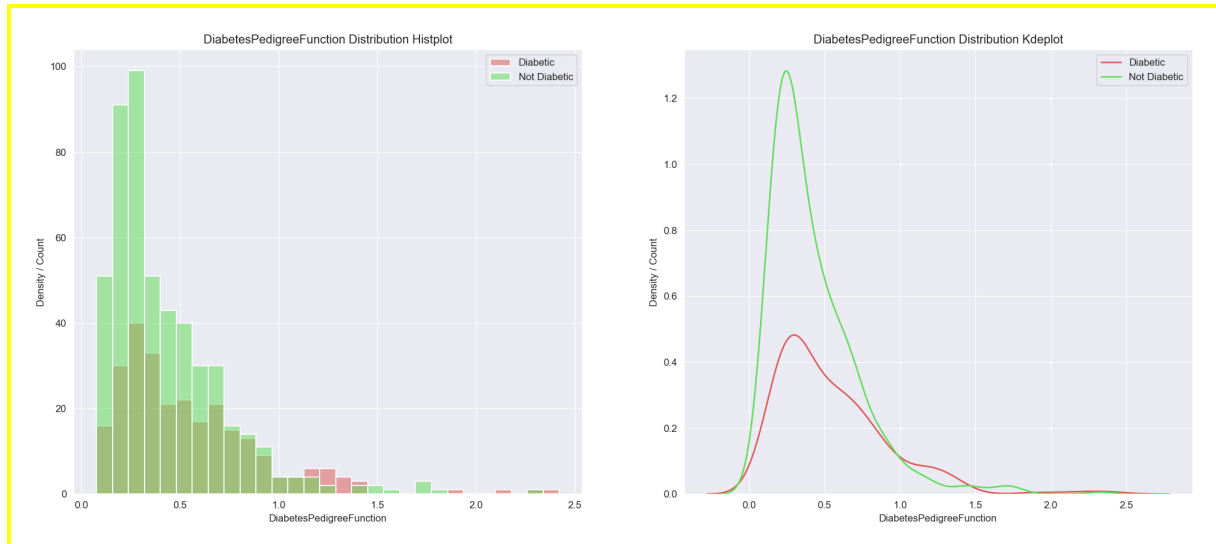
The distributions for both of these plots are very similar indicating that this is not a highly correlated variable, this is also slightly skewed to the left (positively).



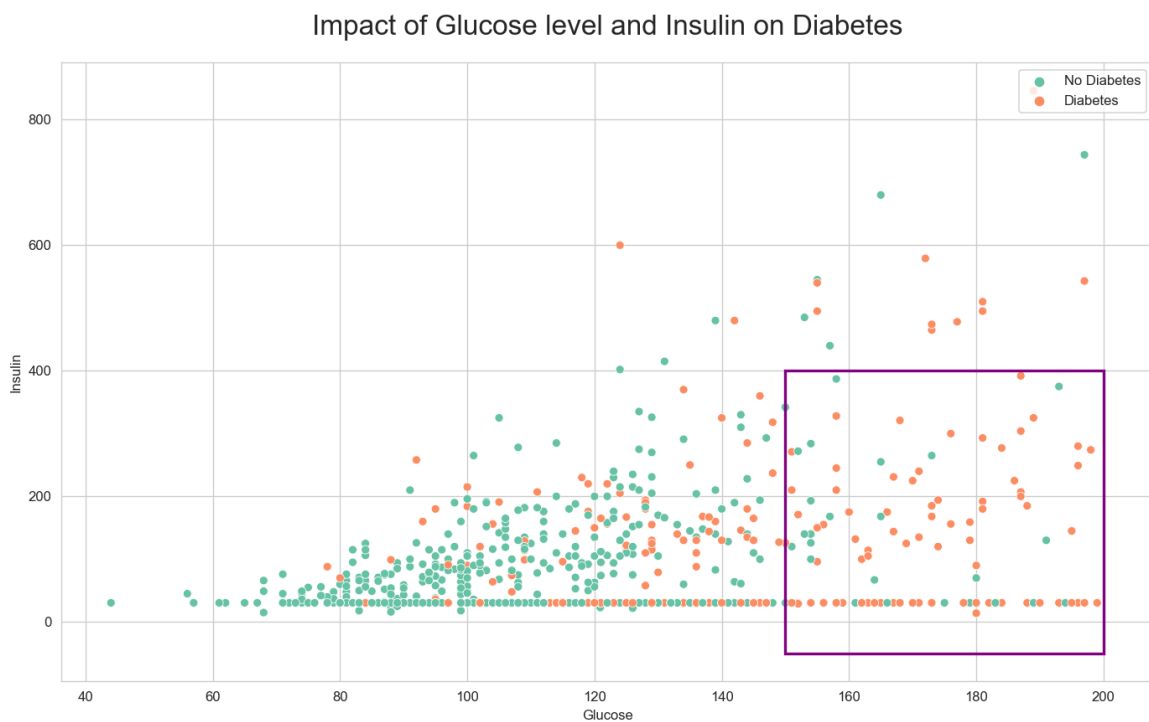
The distributions are very similar for both outcomes again. Age has a left/ positive skew.



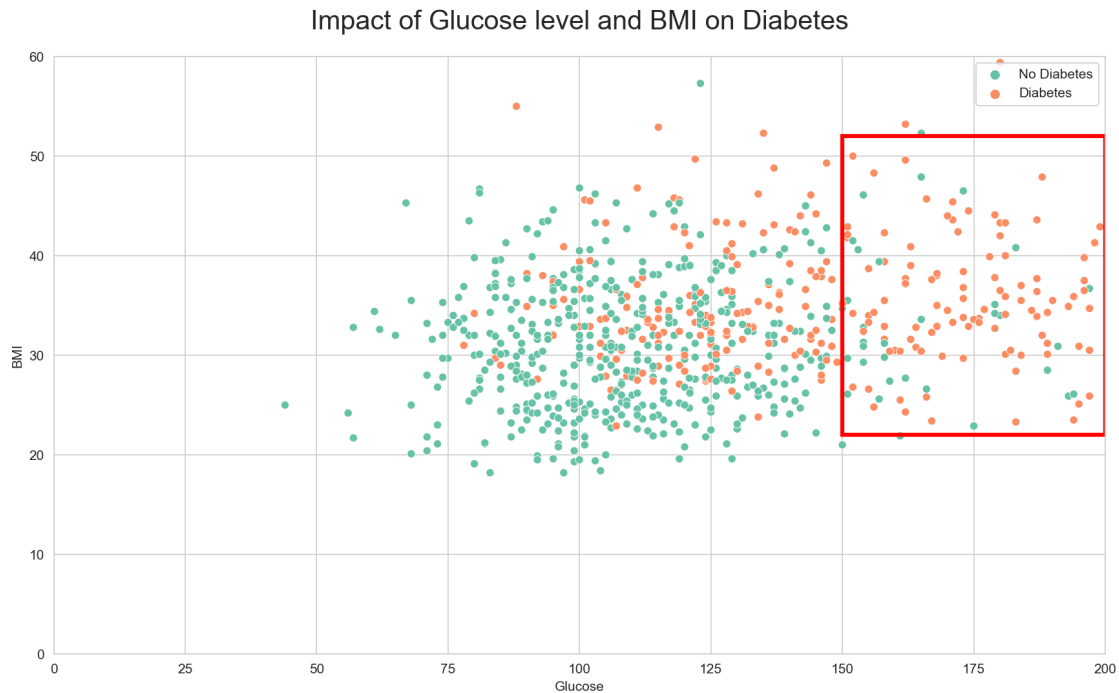
Again, the distributions are very similar for both outcomes of pregnancy. Age has a left/ positive skew.



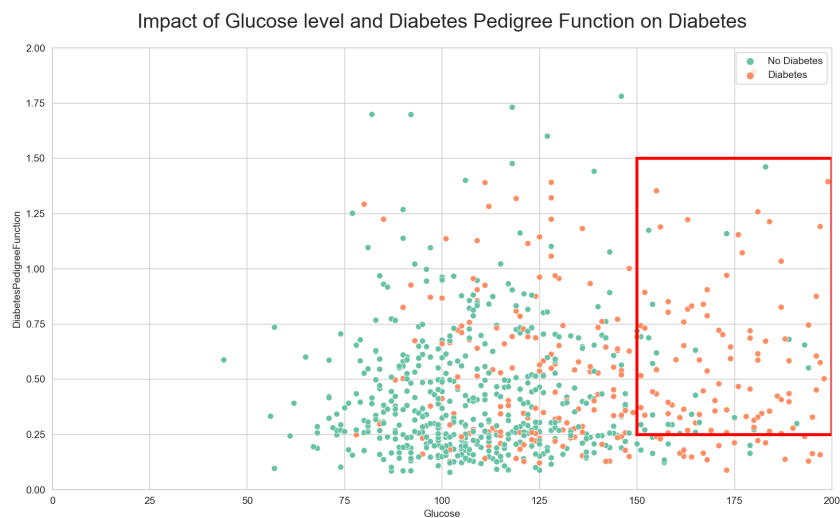
Again, the non diabetic vs diabetic data for diabetes Pedigree function is positively skewed. Overall, it can be said that the Pima diabetes dataset/ variables are heavily skewed.



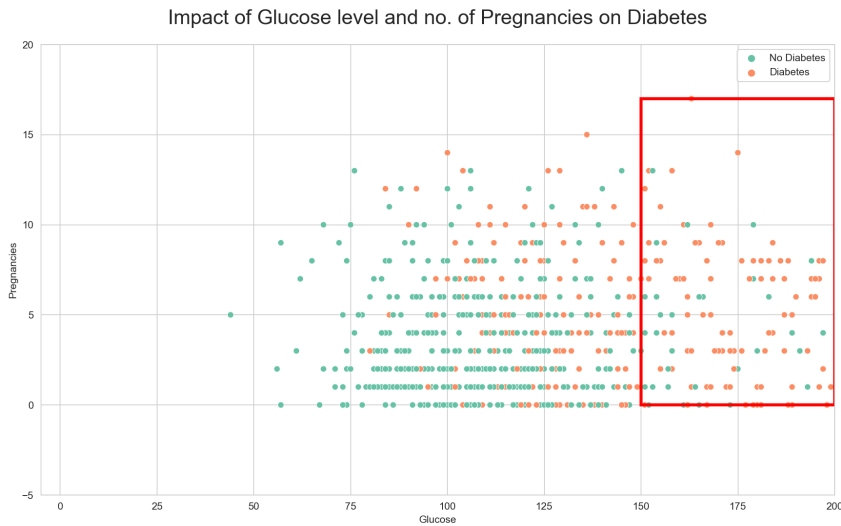
There is a cluster of non diabetics whose insulin remains lower at around 50-200. However as the insulin and glucose variables move away from the cluster to the right, there tends to be more diabetics than non diabetics. The insulin in these cases is either extremely low or extremely high. Hyperinsulinemia could be a reason behind this, this is when the amount of insulin in the blood is beyond what would be considered healthy. Hyperinsulinemia is most commonly associated with diabetes type 2, on its own it is not considered diabetes.



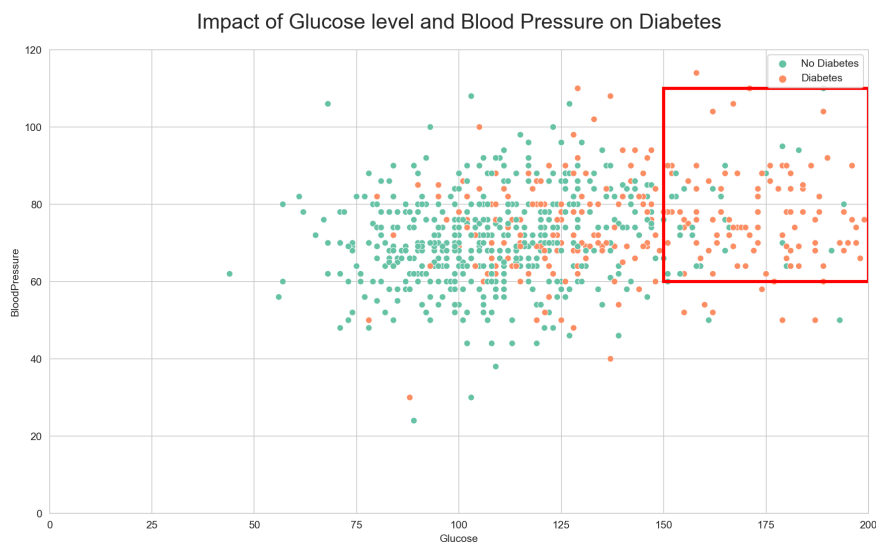
BMI does not seem to affect the likelihood of diabetes with an impact. There is a cluster of both non diabetics and diabetics within the BMI ranges of 25-45 and plasma glucose of 75-140. However, no one below a BMI of 23 seems to have diabetes. Likewise, it seems like those that do not have diabetes above a BMI of around 48 are outliers. The red box demonstrates where the majority cluster of diabetics are alone from non diabetics, this seems to rely heavily on glucose. However, from an observational standpoint, BMI still plays a smaller role.



This plot is very consistent with Glucose being the dominant variable for outcome. It shows the same scatter as you would expect for that situation, this is highlighted by the red box. The distribution for diabetes pedigree function does not seem impactful. There seems to be an even distribution for non diabetics and diabetics for the diabetes pedigree function.



The red box highlights the primary cluster of diabetics without as many non diabetics. This cluster seems to again rely heavily on Glucose. The cluster of non diabetics with glucose between 75-125 seem to have a range of no. times pregnant, it does not seem that pregnancy plays a large role in the likelihood of diabetes in this case.



Within the red box there is a group of majoritively diabetics, this sits outside of the mostly non diabetic cluster outside of the box to the left. It seems that blood pressure does not impact the

likelihood of diabetes as most of the examples for both diabetics and nondiabetics lie within the same diastolic blood pressure range of 50-90. Glucose seems to be the dominant independent variable in this case for diabetes.



Within the red box there is a cluster of persons aged 20-early 30's whose blood glucose lies between 70-145, the vast majority of this group are non diabetic. However, it can also be observed in the box that as age increases while remaining in the plasma glucose range of 70-145 the risk of developing diabetes seems to increase. It can also be noted that outside of the red box from 155-200 plasma glucose that the risk for diabetes is highly reliant on the glucose variable, not age.

Even though there was visualisation of scatter plots within the pair plot, visualising the effect of glucose and other independent variables was important for the outcome variable. In order to properly analyse patterns, a larger scale of the scatter plot was necessary. Scatter plots are known to be useful for studying the correlation against two variables, this is why I have chosen to utilise them as a visualisation technique. Scatterplots are effective at displaying results between two continuous and one categorical variable.

PROBLEM MODELLING

Description of and justification for the AI technique chosen. This will involve specifying the technique(s) chosen and describing why the chosen technique(s) is suitable for the topic.

The model I have chosen to use for this data set is Random Forests (RF). I have chosen to use a supervised classification model as I have a categorical target variable of predicting whether someone has diabetes or not. I have also chosen this model as my data set is imbalanced and Random Forest works well with imbalanced data sets because it is a strong modelling technique that is much more solid than a single decision tree as it is using many (Huh, 2023). Since many trees are being aggregated, the possibility of overfitting and miscalculating because of bias is limited and useful results are produced (Huh, 2023). Another reason why RF is a good choice for imbalanced data sets is that RF uses a combination of ensemble learning and sampling techniques which in turn grow trees on a more balanced set of data because the majority class is being downsampled (Huh, 2023). Random Forest is also known to work well with both categorical and continuous data, the Pima diabetes dataset contains both.

The random forests model works as a collection of many decision trees where predictions are made by each tree on samples of data (Lindner, 2017). Random Forest utilises the method of bootstrapped samples from the training data and random feature selection to create these trees (Misra & Li, 2020). It is important to note that Random Forest can be utilised as a package of either classification or regression trees (Misra & Li, 2020). Results of individual decision trees are averaged which aids in reducing the overfitting of data (Misra & Li, 2020). Selection of the best solution is then classified through a vote, and will be deemed the final prediction result (Misra & Li, 2020). Despite RF fitting the criteria of suitability to my dataset, I have also explored other datasets that appear to be suitable on paper. Other models I explored included (but is not limited to) Catboost, K-nearest neighbours, Logistic regression, Decision trees, and more. Another reason for my choice of RF over other models can be exemplified through these visualisations as one can see, no other model performed as well as the random forest classifier.

Before selecting this model, I ran my data through multiple models without any cleaning or tuning to see which one handled the data best. The image below depicts some of these results. I decided that I would use random forests because it had the best accuracy score. Even though logistic regression would have been a good alternative as it works well with categorical 0 and 1, I decided that due to the accuracy I would go with Random Forest.

Raw data results:

```

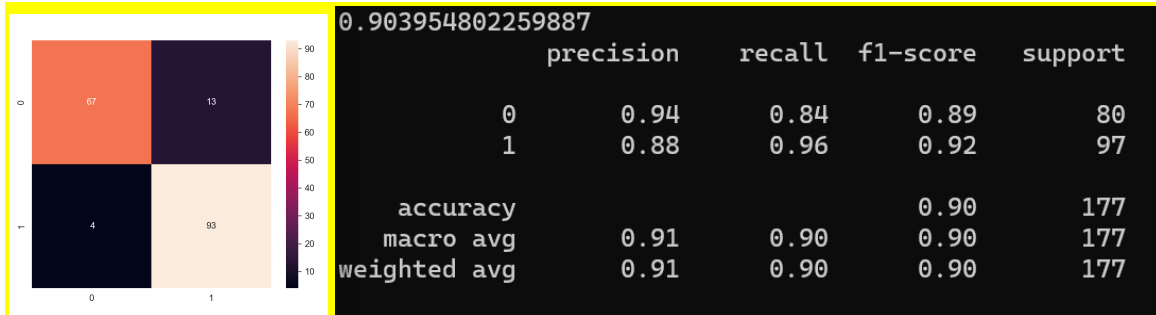
Train Accuracy of Logistic Regression 77.68729641693811
Accuracy (test) score of Logistic Regression 79.22077922077922
Accuracy (test) score of Logistic Regression 79.22077922077922
Train Accuracy of KNN 79.96742671009773
Accuracy (test) score of KNN 70.77922077922078
Accuracy (test) score of KNN 70.77922077922078
Train Accuracy of naive-bayes 76.0586319218241
Accuracy (test) score of naive-bayes 75.32467532467533
Accuracy (test) score of naive-bayes 75.32467532467533
Train Accuracy of Random Forest 99.8371335504886
Accuracy (test) score of Random Forest 81.81818181818183
Accuracy (test) score of Random Forest 81.81818181818183

```

Random forest classification model after cleaning and tuning

-highest

1.



2.

```

0.903954802259887
precision    recall  f1-score   support

0           0.93      0.85      0.89         80
1           0.88      0.95      0.92         97

accuracy          0.90         177
macro avg         0.91      0.90      0.90         177
weighted avg      0.91      0.90      0.90         177

Test Accuracy : 0.903954802259887

```

3.

```

0.9209039548022598
precision    recall  f1-score   support

0           0.97      0.85      0.91         80
1           0.89      0.98      0.93         97

accuracy          0.92         177
macro avg         0.93      0.91      0.92         177
weighted avg      0.93      0.92      0.92         177

Test Accuracy : 0.9209039548022598

```

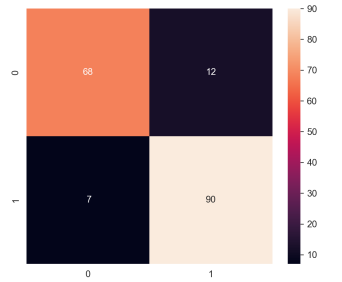
Another example of random forest:

0.9209039548022598

	precision	recall	f1-score	support
0	0.99	0.84	0.91	80
1	0.88	0.99	0.93	97
accuracy			0.92	177
macro avg	0.93	0.91	0.92	177
weighted avg	0.93	0.92	0.92	177

Test Accuracy : 0.9209039548022598

More random forest



0.8926553672316384

	precision	recall	f1-score	support
0	0.91	0.85	0.88	80
1	0.88	0.93	0.90	97
accuracy			0.89	177
macro avg	0.89	0.89	0.89	177
weighted avg	0.89	0.89	0.89	177

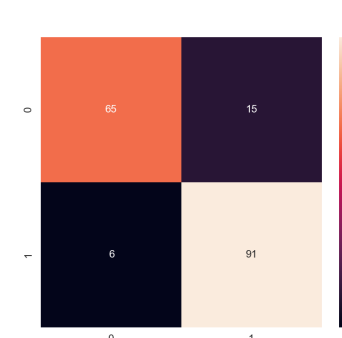
Test Accuracy : 0.8926553672316384

Catboost

-3rd best

-good number of true positive and true negatives but could still be improved

-I do not know enough about catboost to use the model and I thought that Random Forest would be a better fit because it is known to work well with categorical and continuous data.



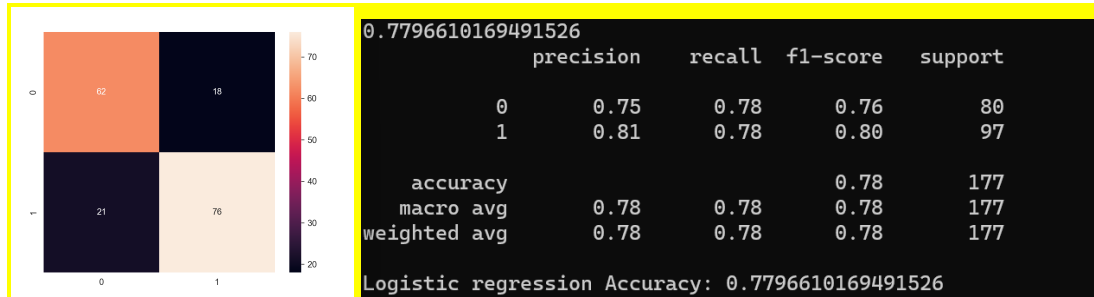
0.8813559322033898
Test Accuracy Catboost : 0.8813559322033898

Confusion Matrix
[[65 15]
[6 91]]

	precision	recall	f1-score	support
0	0.92	0.81	0.86	80
1	0.86	0.94	0.90	97
accuracy			0.88	177
macro avg	0.89	0.88	0.88	177
weighted avg	0.88	0.88	0.88	177

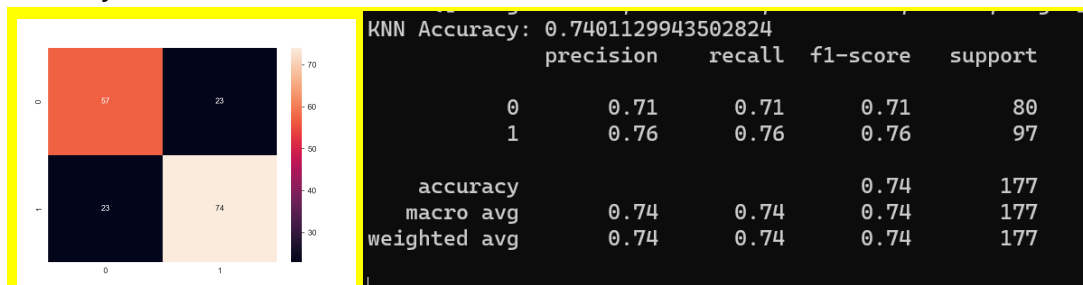
Logistic regression

-lower score



KNN

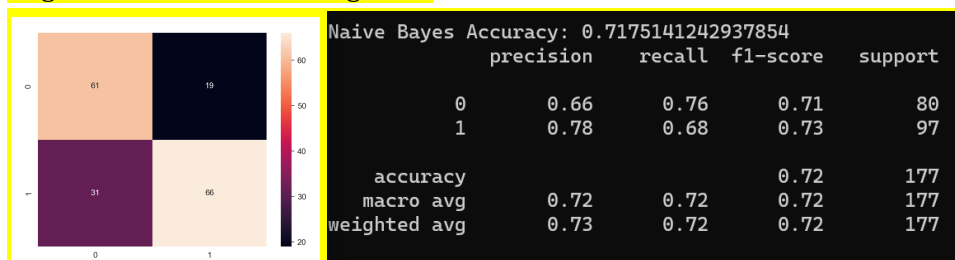
KNN had the lowest true positive. This is the algorithm that I thought I would use originally, however, it did not perform the way I wanted it to. Though it is a very versatile model, due to the accuracy scores I had to rule it out.



Naive bayes (lowest accuracy)

-most even distribution of true negatives and true positives

-highest number of false negatives



Implementation and evaluation of the chosen technique. This will involve preparing the data in an appropriate fashion for the model, constructing a model to solve the problem and evaluating the performance of the model over the chosen data set.

Before preparing my model I was able to have a look over the dataset and identify that there were some 0 values for variables such as Skin thickness, glucose, insulin, BMI, blood pressure and pregnancies.

Chart depicting the number of zero values:

Variable :	No. of zero values
Glucose	5
Blood Pressure	35
Skin Thickness	227
Insulin	384
BMI	11

Although it may be possible to have 0 insulin with type 2 diabetes, many of the 0 insulin values were associated with no diabetes outcome, thus, I had to replace the 0 values with the median (skewed). Insulin also had the highest number of 0 values, notably exceeding the number of diabetics in the dataset itself. I also replaced skin thickness 0 values with median as it is a skewed variable, along with BMI as well. I decided to replace Glucose and blood pressure with the mean as they seemed evenly distributed with fewer outliers than the other values. This follows the rule of using mean for balanced data and median for skewed data(Kumar, 2023). I decided to replace these missing values as missing values can reduce the statistical power of the model, reduce the representation of the dataset, and lead to invalid conclusions(Htoon, 2021). There were some 0 values for pregnancy, however, this would be seen as a reliable value as it is possible to have 0 pregnancies.

I have decided to not normalise my data, this is not necessary for tree-based models and would significantly impact my accuracy(Arya, n.d.). Since each node in a random forest does not compare feature values, it is splitting a sorted list that requires absolute values for branching(Arya, n.d.). The random forest algorithms basis is partitioning the data to eventuate a prediction, thus, no normalisation / feature scaling is required(Arya, n.d.). For example, when feature scaling, my accuracy decreased to 0.84 percent.

I did, however, decide to increase the number of trees, this is because most of my data is under correlated. The under correlated variables are easier to model as there are fewer dimensions involved.

I did notice some outliers in my dataset for example a BMI of 57 and glucose level of 196. Even though some readings have suggested that RF is robust to outliers (Srivastava, 2022), I have found otherwise. When leaving the outliers in my data, again, the accuracy significantly decreased to around 0.83 percent. I decided to remove the bottom 25 percent and top 25 percent of outliers for my data with an IQR of 1.5. I felt it was important to remove some of these extreme outliers as these were extreme values that lay far from the average that could be due to measurement variability or simply experimental error (Srivastava, 2022). I also removed these as most of my data was heavily skewed.

Random forest is able to calculate feature importance as the decrease in node impurity is weighted by the probability of reaching that node (R, 2023). The node probability can be calculated by the number of samples that reach the node, divided by the total number of samples. The higher the value the more important the feature (R, 2023). This is likewise to decision trees (R, 2023). However, I still tested if adding selectKbest and chi2 to perform feature selection would impact my models performance. My findings were that the selectKbest and chi2 didn't do anything that impacted my performance so I chose not to include these in my data cleaning procedures.

```
Number of Outcome 0: 500
Number of Outcome 1: 268
```

I have decided to account for over sampling as my data set is imbalanced. I have experimented using the Synthetic minority Over Sampling Technique (SMOTE) and RandomOverSampler. SMOTE is able to synthesise new examples for the minority class, which in this case, outcome 1 (having diabetes) (Brownlee, 2021). The minority class only has 268 whereas outcome 0 has 500 examples. Using SMOTE significantly decreased my accuracy, however, RandomOverSampling propped up my accuracy. RandomOverSampling randomly selects examples from the minority class and replicates them (Brownlee, 2021).

```
Number of Outcome 0 after SMOTE: 435
Number of Outcome 1 after SMOTE: 435
0.8160919540229885
      precision    recall  f1-score   support

     0       0.87     0.78     0.82         94
     1       0.77     0.86     0.81         80

   accuracy                0.82        174
  macro avg       0.82     0.82     0.82        174
 weighted avg       0.82     0.82     0.82        174

Test Accuracy : 0.8160919540229885
```

```

Number of Outcome 0 after oversampling: 435
Number of Outcome 1 after oversampling: 435
0.9080459770114943

```

	precision	recall	f1-score	support
0	0.98	0.85	0.91	94
1	0.85	0.97	0.91	80
accuracy			0.91	174
macro avg	0.91	0.91	0.91	174
weighted avg	0.92	0.91	0.91	174

```

Test Accuracy : 0.9080459770114943

```

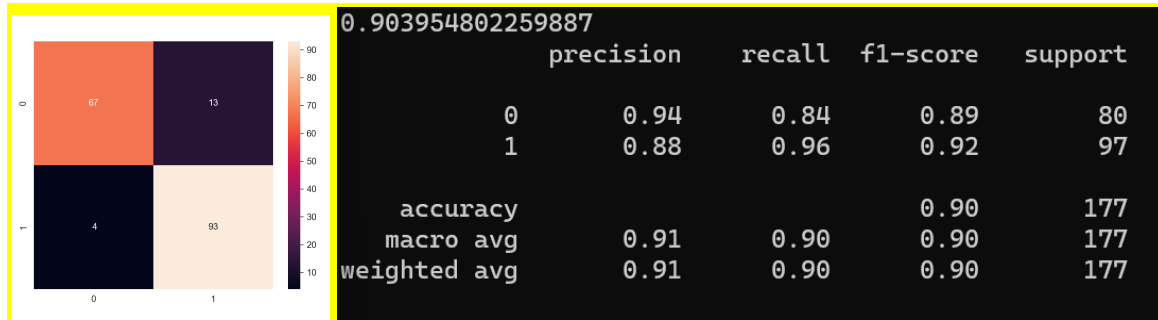
The last preprocessing technique I implemented was using an imputer. It is important to note that Random Forest is unable to handle nan or missing values. The imputer I have chosen to handle these values is the KNNimputer. The KNN imputer handles missing values by predicting them as a mean from the missing values nearest neighbours. That is, the imputer is utilising similar examples to make this prediction. The reason why I have implemented this imputer is to handle missing values in my software implementation, for example, if the user does not know the input for glucose it will be a nan value.

Upon review, I am happy with the model I have chosen. In the future, I think I would like to experiment more with hyperparameters to improve my model or even try to ensemble models together. It would be easier if random forests could deal with missing values, however, this is something that another model could be used for, meaning, I would not have to use an imputer. Though I believe that my model choice was a good fit to my dataset, I believe that the accuracy could be further improved.

From the example I have attached, you can see that 87 cases of the absence of diabetes were correctly identified and four were incorrectly identified. On the other hand, 93 true cases of diabetes were identified and 11 cases were falsely identified.

Breaking down my model, the precision was relatively high for detecting the absence of diabetes in comparison to the case of diabetes being present, this may be due to my random sampling method and the replication and recycling of examples. Further exploration of other oversampling techniques could improve the precision for diabetic prediction. It would be nice to see a more even precision. This could also be due to the way I have prepared my data. I would like to try to implement another KNNImputer to use inside the model next time as it may prove more accurate than just replacing 0's with median or mean.

Random Forest model:



SOFTWARE IMPLEMENTATION

SOFTWARE IMPLEMENTATION EXPLAINED

I have implemented my system using my saved Random Forest model along with my saved imputer. The reason why I have implemented an imputer is to account for nan values that the user may input. I did not want the user to input 0 values if they did not know the value because this would throw off the prediction. The imputer allows the user to enter "na" as an answer, a predicted value is then used as a replacement, the KNNImputer uses the mean of the neighbours to predict these values. I have also added an option to replace 0 values with mean or median depending on the variable.

Upon opening the software the user will be greeted with the main menu where they can choose to navigate to either: testing for diabetes, instructions, or to exit. If the user chooses to look at the instructions they can then return back to the main menu. The instructions and the testing function both include the metrics for each variable, I felt that this was important as it can be very unclear and confusing without these.

The software is designed to calculate the users BMI using weight and height ($\text{bmi} = \text{weight} / ((\text{height}/100) ** 2)$), this is to make the software more intuitive for the user and to base the software on known values. The software will also calculate the diabetes pedigree function for the user by the number of family members * 0.1/age of the individual. Again, I have implemented this feature to aid in the user experience, some people may not know what a diabetes pedigree function is.

```
bmi = weight / ((height/100) ** 2)
```

```
diabetes_pedigree_function = (family_members * 0.1) / age
```

The test can still be run with nan inputs but the instructions menu clearly explains that missing variables could affect/decrease the accuracy of the test. This is why the software is meant to be used in a clinical setting.

The user is able to input values and receive feedback at the end as to whether the patient has diabetes or not. The user will then be asked if they want to save the data (this will be saved as a csv file to the computer). The data will then clear and the user will be returned to the main menu, they can either exit or continue entering data. The reason I have included the option to save the data as a csv file is because I think that this would be important in a clinical setting for notes and patient history.

USER GUIDE FOR SOFTWARE IMPLEMENTATION:

Instructions:

Please note that this program is intended for the calculation of diabetes probability in Females above 21 years of age in a clinical setting.")

Instructions:

1. You will be asked to input the following variables please input integer variables as a number without measurement details.
2. Pregnancies (number of pregnancies as an integer).
3. Glucose level (concentration of plasma glucose, 2hr oral glucose tolerance test).
4. Blood pressure (diastolic blood pressure, mm Hg).
5. Skin thickness (mm).
6. Insulin level (2-hr serum (U/ml).
7. Weight (Kg).
8. Height (cm).
9. Number of family members with diabetes (calculate diabetes pedigree function, these must be BLOOD related family members.
10. Age (years).
11. The program will calculate the BMI and diabetes pedigree function.")
12. If you do not know the answer, please enter NA or 0- note that missing variables will decrease the accuracy of your test"
13. Please enter all values as numbers only, if you are unsure, enter na or 0.
14. NA/na will predict the missing variable based on similar examples and 0 will predict as a general mean or median.
15. Based on the provided information, the program will predict if you have diabetes or not.
16. You will be asked if you want to save the data, this will be saved as patient_data.csv"
17. The data will then clear and you will return to the main menu

References

1. Admin. (n.d.). *Types of diabetes*. Diabetes UK.
<https://www.diabetes.org.uk/diabetes-the-basics/types-of-diabetes>
2. Arya, N. (n.d.). *Does the Random Forest Algorithm Need Normalization?* - KDnuggets. KDnuggets.
<https://www.kdnuggets.com/2022/07/random-forest-algorithm-need-normalization.html>
3. Brownlee, J. (2021a). Random Oversampling and Undersampling for Imbalanced Classification. *MachineLearningMastery.com*.
<https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/>
4. Brownlee, J. (2021b). SMOTE for Imbalanced Classification with Python. *MachineLearningMastery.com*.
<https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification>
5. Dabral, S. (2021, December 15). What is an Outlier? How to handle and remove them? Algorithms that are affected by outliers. *Medium*.
<https://medium.com/analytics-vidhya/what-is-an-outliers-how-to-detect-and-remove-them-which-algorithm-are-sensitive-towards-outliers-2d501993d59>
6. Htoon, K. S. (2021, December 15). A Guide To KNN Imputation - Kyaw Saw Htoon - Medium. *Medium*.
<https://medium.com/@kyawsawhtoon/a-guide-to-knn-imputation-95e2dc496e>
7. Huh, K. (2023, May 22). Surviving in a Random Forest with Imbalanced Datasets. *Medium*.
<https://medium.com/sfu-csmp/surviving-in-a-random-forest-with-imbalanced-datasets-b98b963d52eb#:~:text=Random%20forest%20is%20an%20ideal,penalizes%20misclassifying%20the%20minority%20class.>
8. JDRF UK. (2021, October 15). *What causes type 1 diabetes?* - JDRF, the type 1 diabetes charity. JDRF, the Type 1 Diabetes Charity.
<https://jdrf.org.uk/information-support/about-type-1-diabetes/causes-of-type-1-diabetes/#:~:text=While%2090%20per%20cent%20of,with%20type%201%20diabetes%20risk.>
9. Kharroubi, A., & Darwish, H. M. (2015). Diabetes mellitus: The epidemic of the century. *World Journal of Diabetes*, 6(6), 850.
<https://doi.org/10.4239/wjd.v6.i6.850>
10. Kumar, A. (2023, March 26). *Python - Replace Missing Values with Mean, Median & Mode - Data Analytics*. Data Analytics.
<https://vitalflux.com/pandas-impute-missing-values-mean-median-mode/#:~:text=Mean%20imputation%20is%20often%20used,to%20outliers%20than%20the%20mean.>

11. Lindner, C. (2017). Automated Image Interpretation Using Statistical Shape Models. In *Elsevier eBooks* (pp. 3–32).
<https://doi.org/10.1016/b978-0-12-810493-4.00002-x>
12. Misra, S., & Li, H. (2020). Noninvasive fracture characterization based on the classification of sonic wave travel times. In *Elsevier eBooks* (pp. 243–287).
<https://doi.org/10.1016/b978-0-12-817736-5.00009-0>
13. *National Institute of Diabetes and Digestive and Kidney Diseases*. (2021, July 19). National Institutes of Health (NIH).
<https://www.nih.gov/about-nih/what-we-do/nih-almanac/national-institute-diabetes-digestive-kidney-diseases-niddk>
14. R, S. E. (2023). Understand Random Forest Algorithms With Examples (Updated 2023). *Analytics Vidhya*.
<https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/#:~:text=One%20of%20the%20most%20important,for%20classification%20and%20regression%20tasks>.
15. Srivastava, A. (2022, January 7). Let's Talk about Random Forests! - Analytics Vidhya - Medium. *Medium*.
<https://medium.com/analytics-vidhya/lets-talk-about-random-forests-524ae1138d8b#:~:text=Random%20forests%20are%20robust%20to,output%20of%20multiple%20decision%20trees>.
16. *What is Diabetes?* (2023, April 24). Centers for Disease Control and Prevention.
<https://www.cdc.gov/diabetes/basics/diabetes.html#:~:text=With%20diabetes%20C%20your%20body%20doesn,vision%20loss%20and%20kidney%20disease>.
17. *What Is Diabetes?* (2023). *National Institute of Diabetes and Digestive and Kidney Diseases*.
<https://www.niddk.nih.gov/health-information/diabetes/overview/what-is-diabetes>

References used for coding help:

1. Deepmalviya. (2022). 87% Accuracy using Random Forest. *Kaggle*.
<https://www.kaggle.com/code/deepmalviya7/87-accuracy-using-random-forest>
2. *Pima Indians Diabetes Database*. (2016, October 6). *Kaggle*.
<https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database>
3. R, S. E. (2023). Understand Random Forest Algorithms With Examples (Updated 2023). *Analytics Vidhya*.
<https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>
4. Simplilearn. (2023). Random Forest Algorithm. *Simplilearn.com*.
<https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm#:~:text=Step%201%3A%20Select%20random%20samples,as%20the%20final%20prediction%20result>.

5. *sklearn.impute.KNNImputer*. (n.d.). Scikit-learn.
<https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html>