BIRMINGHAM CITY University

[redacted]

AI and Machine Learning Project
2023–2024

Individual Report

Predictive modelling of Diabetes using the Decision Tree Algorithm

[redacted]

Table of Contents

1 Report Introduction

According to the IDF Diabetes Atlas, over 10.5% of the world's adult population is living with diabetes. It is a significant health problem and has been classed as a global epidemic, being a common cause of death in low and middle-income countries (IDF, 2021). As diabetes causes further complications as it develops, it is important to predict it early or even to prevent its development beforehand. There are many health-related behaviours that affect a person's risk for diabetes, such as diet, weight, physical activity, age, etc. This report explores the health of those with and without diabetes, and the machine learning techniques used to develop a predictive model to help diagnose diabetes, based on the several variables given.

1.1 Dataset identification

Our research team found the 'Diabetes Health Indicators' dataset on Kaggle which contains 21 different health-related characteristics of survey participants, and the presence of diabetes.

The data is available on Kaggle but is originally from the BRFSS survey in 2015. This is an annual telephone survey conducted by the CDC to collect data about the health and health-related risk behaviours of U.S. residents. The 2015 dataset contains just over 253,000 survey responses. Table 1 describes each feature in the dataset, as well as scales if applicable.

Table 1: Features of the 'Diabetes_012' dataset

Attribute	**Description**
Diabetes_012	Presence of diabetes, 0 represents no diabetes, 1 is prediabetes, and 2 for diabetes
HighBP	If the respondent has been diagnosed with high blood pressure
HighChol	If the respondent has been diagnosed with high cholesterol
CholCheck	If the respondent has had a cholesterol check in the last 5 years
BMI	The respondent's Body Mass Index
Smoker	If the respondent has smoked at least 100 cigarettes in their life
Stroke	If the respondent has ever had a stroke
HeartDiseaseorAttack	If the respondent has had CHD or MI
PhysActivity	If the respondent has done any physical activity in the past 30 days – not including job
Fruits	If the respondent consumes fruit once or more per day
Veggies	If the respondent consumes vegetables once or more per day
HvyAlcoholConsump	If the respondent is a heavy drinker – more than 14 drinks per week for men and more than 7 per week for women

AnyHealthcare	If the respondent has any healthcare coverage
NoDocbcCost	If the respondent could not see a doctor because of the cost
GenHlth	General rating of health, scale of 1-5 – 1 being excellent, 5 being poor
MentHlth	How many days has their mental health been poor in the past 30 days, scale of 1-30
PhysHlth	How many days has their physical health been poor in the past 30 days, scale of 1-30
DiffWalk	If the respondent has difficulty walking or climbing stairs
Sex	If the respondent is male or female
Age	13-level age category, - 1 = 18, 9 = 60-64, 13 = 80 or older.
Education	Educational level, scale of 1-6 - 1 = Never attended school or only kindergarten 2 = Grades 1 through 8 (Elementary) 3 = Grades 9 through 11 (Some high school) 4 = Grade 12 or GED (High school graduate) 5 = College 1 year to 3 years (Some college or technical school) 6 = College 4 years or more (College graduate)
Income	Income level, scale of 1-8 - 1 = less than $10,000 5 = less than $35,000 8 = $75,000 or more

1.2 Supervised learning task identification

From the chosen dataset, the question the research team aims to develop a machine learning model that accurately diagnoses an individual based on their health and lifestyle choices. The ground truth our task is based on is “Diabetes_012”. This variable is categorical, with 3 categories, therefore we have been given a classification problem.

2 Exploratory Data Analysis

2.1 Question(s) identification

Table 2 shows a set of questions that this report aims to answer, along with assumptions made before exploration. This provides a foundation for our EDA, knowing which features to explore and help us further understand the dataset.

Table 2: Questions to be answered through EDA

Question no.	Question	Assumption
1	What is the overall prevalence of diabetes in the dataset?	As the IDF states that 10.5% of the population is diagnosed with diabetes, we can expect that the dataset is dominated by respondents who haven't developed diabetes (IDF, 2021).
2	Do males have an increased likelihood of developing diabetes?	Diabetes is more commonly diagnosed in men, at a younger age and lower BMIs. Women however are more susceptible to psychological stress, cardiovascular diseases, etc. (Kautzky-Willer, A. et al., 2016).
3	How does the presence of diabetes differ across different age groups?	Older people are more at risk to developing diabetes. This can be caused by a few factors, such as a lack of physical activity, poor eating, mental health, etc (Mordarska, K. et al., 2017).
4	Does a high BMI increase the likelihood of developing diabetes?	Higher BMI values generally indicate high levels of body fat for a person's age and height (CDC, 2021). The higher a person's BMI, the more at risk they are to health complications such as high blood pressure, heart disease and diabetes.
5	Does heavy alcohol consumption or smoking increase the likelihood of developing diabetes?	Both heavy drinking and smoking can lead to trouble with insulin levels, which is a common factor in developing diabetes.
6	Does high blood pressure and cholesterol increase the likelihood of developing diabetes?	High levels of either blood pressure or cholesterol can put an individual at risk of developing diabetes. Diabetes can also cause high cholesterol itself (Diabetes UK , 2017).
7	How does the prevalence of diabetes differ across different levels of income?	Low-income can lead to struggles with health, obesity, and stress, all of which are factors that increase someone's risk to diabetes (Hsu, C. et al., 2012).
8	Do people who have difficulty walking have a higher risk of developing diabetes?	As physical activity is a common factor in developing diabetes we can assume that people who have difficulty walking, either due to weight or disabilities, would be at higher risk.

2.2 Exploratory Data Analysis process and results

To first understand the data given to us, we can use the head function as shown in Figure 1 to display each feature and examples of the data.

```
df.head(8).T
```

	0	1	2	3	4	5	6	7
Diabetes_012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HighBP	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0
HighChol	1.0	0.0	1.0	0.0	1.0	1.0	0.0	1.0
CholCheck	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0
BMI	40.0	25.0	28.0	27.0	24.0	25.0	30.0	25.0
Smoker	1.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0
Stroke	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HeartDiseaseorAttack	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PhysActivity	0.0	1.0	0.0	1.0	1.0	1.0	0.0	1.0
Fruits	0.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0
Veggies	1.0	0.0	0.0	1.0	1.0	1.0	0.0	1.0
HvyAlcoholConsump	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
AnyHealthcare	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0
NoDocbcCost	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
GenHlth	5.0	3.0	5.0	2.0	2.0	2.0	3.0	3.0
MentHlth	18.0	0.0	30.0	0.0	3.0	0.0	0.0	0.0
PhysHlth	15.0	0.0	30.0	0.0	0.0	2.0	14.0	0.0
DiffWalk	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0
Sex	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
Age	9.0	7.0	9.0	11.0	11.0	10.0	9.0	11.0
Education	4.0	6.0	4.0	3.0	5.0	6.0	6.0	4.0
Income	3.0	1.0	8.0	6.0	4.0	8.0	7.0	4.0

Figure 1: Displayed features of the diabetes dataset

From this we can see that we have a total of 22 features, the majority of which are binary.

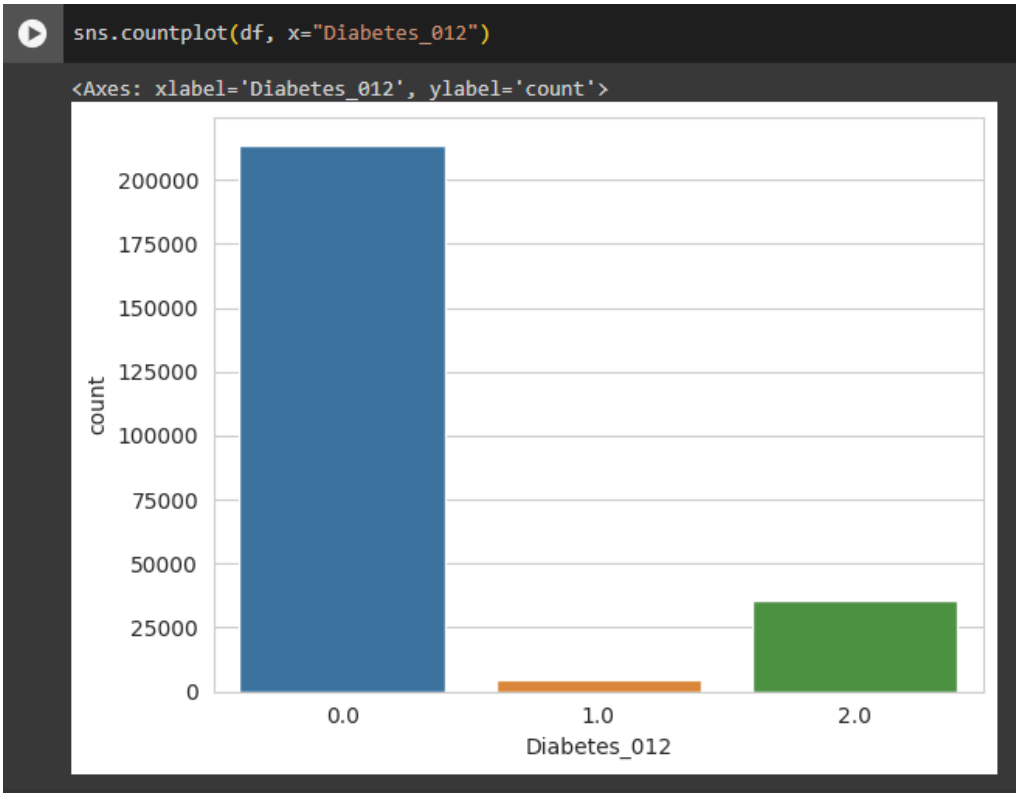


Figure 2: Prevalence of diabetes across respondents, 0 equal to no diabetes, 1 equal to prediabetes, and 2 equals to diabetes

In Figure 2, we can see that there are significantly more respondents with no diabetes. Although this may correlate with the prevalence of diabetes in the real world, it creates an unbalanced dataset. Considering this, a model trained on this data may have difficulty predicting the presence of diabetes as compared to a patient without diabetes, which could introduce inaccuracies in predictions for real use-cases. We have also noticed that there are very few records of individuals with prediabetes, which may produce inaccuracies in our model. We will have to take this into account later in the report during the data processing phase.

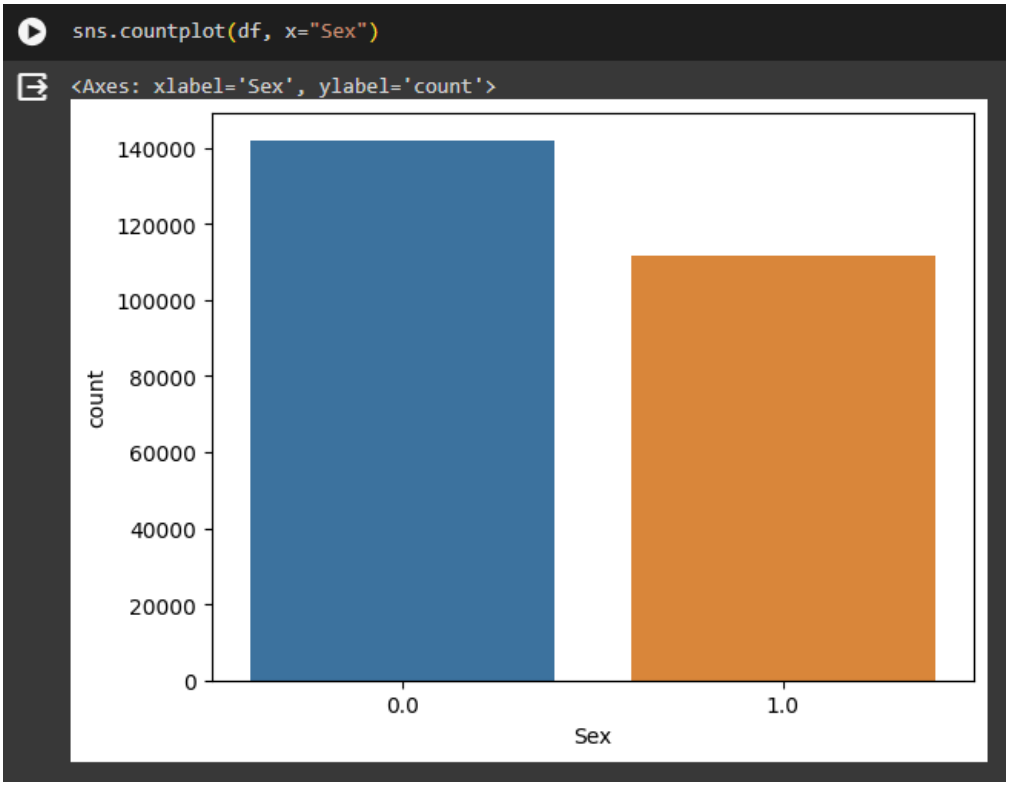


Figure 3: Count of male and female respondents

From the above chart, we can see that the number of males outweigh the number of females in our dataset. This is another imbalance in our dataset that could impact our predictive model, however it's not as significant as the overfitting caused by the imbalance in the 'Diabetes_012' feature. There is no solidifying reason we could find that caused this imbalance, it could simply be due to the population of men and women in the states that the BRFFS survey was conducted in at the time.

```
gender = sns.countplot(df, x="Sex", hue="Diabetes_012")

for p in gender.patches:
   gender.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```

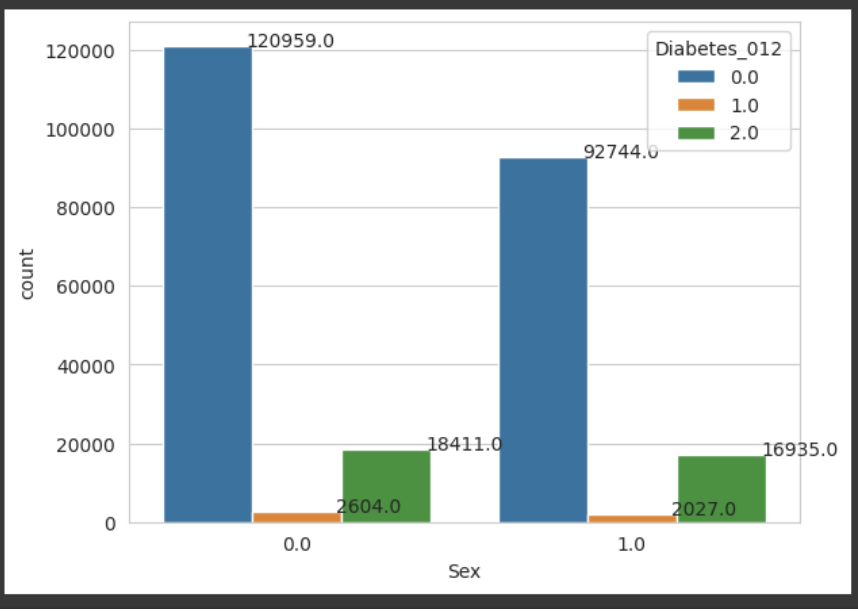
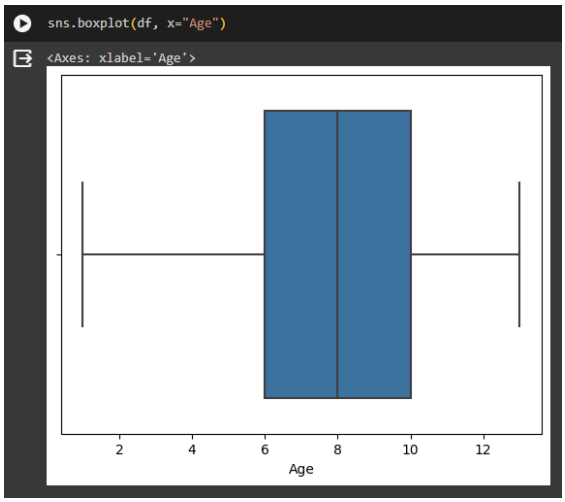


Figure 4: Prevalence of diabetes in male and female respondents

Figure 3 shows that even with the gender imbalance, there is a similar count of female respondents diagnosed with diabetes. From our data, this could mean that females have a higher risk of developing diabetes. This aligns with one of our initial assumptions that women are more at risk to health and mental health complications, which are factors in developing diabetes.



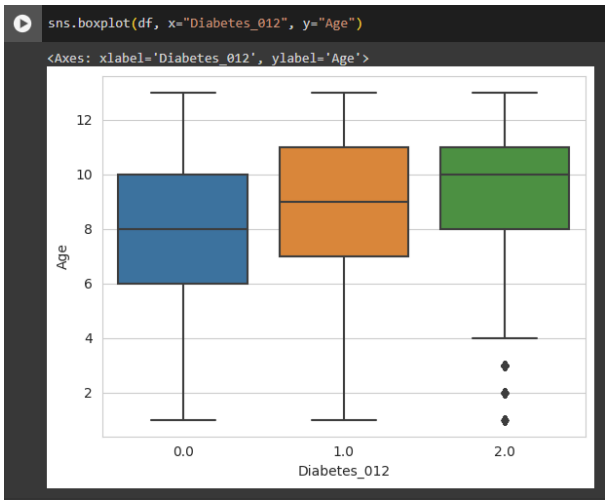


Figure 5 & 6: Distribution of respondent ages

From the boxplot in Figure 5, we can see that most respondents lie within the age groups 6 to 10, ages 45 to 69, with ages 55 to 59 as the median age group. Figure 6 displays the distribution of ages according to their diagnosis, where the average age increases with the presence of diabetes. According to our data the older an individual is, the more likely they are to be diagnosed with diabetes. This confirms that our assumption made before EDA was correct.

Table 3: Calculated age groups (CDC, 2015)

Age Group (AGEG5YR)	Range
1	18 to 24
2	25 to 29
3	30 to 34
4	35 to 39
5	40 to 44
6	45 to 49
7	50 to 54
8	55 to 59
9	60 to 64
10	65 to 69
11	70 to 74
12	75 to 79
13	80 or older

```
sns.histplot(df, x="BMI", hue="Diabetes_012", multiple="stack", bins=25,)
plt.legend(labels=['No diabetes','Prediabetes','Diabetes'])
```

```
<matplotlib.legend.Legend at 0x7f34a9525ba0>
```

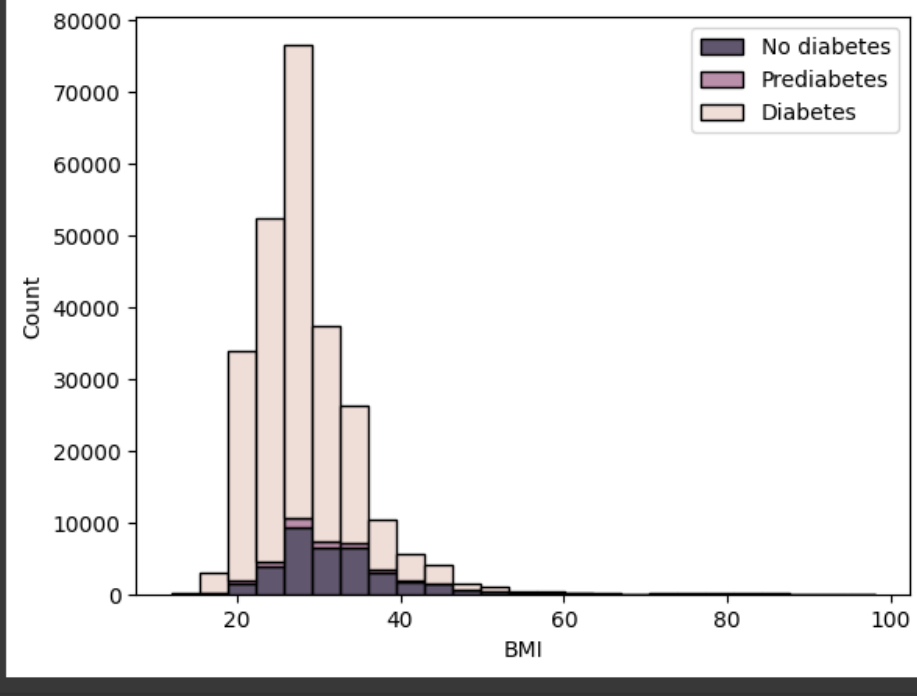
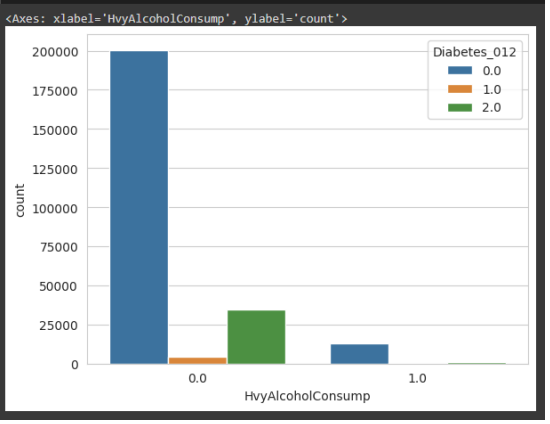


Figure 7: BMI distribution

The distribution of BMI across levels of diabetes are practically identical. According to the CDC in 2015, 39.8% of adults in the US were obese (Hales, C. *et al.,* 2017). Most respondents with any diagnosis of diabetes have a BMI of 23 to 36, meaning that most are overweight. The BMI however is right skewed, with the most extreme value of 98. As we are developing a classification-based model, these extreme values may not significantly affect its performance.

```
sns.countplot(df, x="HvyAlcoholConsump", hue="Diabetes_012")
```

```
<Axes: xlabel='HvyAlcoholConsump', ylabel='count'>
```




```
[28] sns.countplot(df, x="Smoker", hue="Diabetes_012")
```

```
<Axes: xlabel='Smoker', ylabel='count'>
```

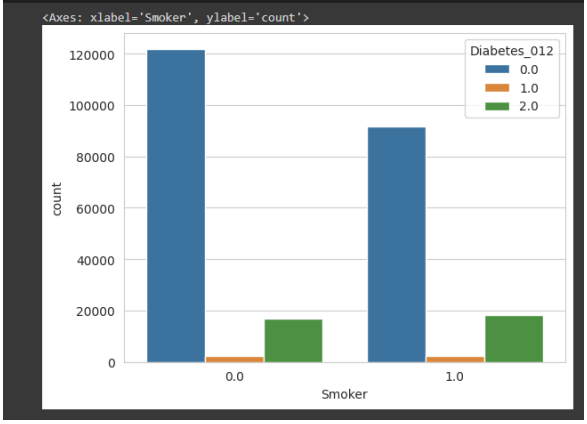
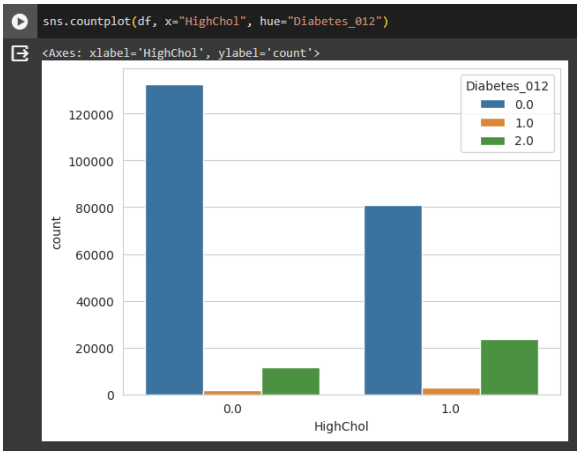


Figure 8 & 9: Relationship between diabetes, smoking, and alcohol consumption

Surprisingly, our data shows that heavy alcohol consumption doesn't have a strong correlation with developing diabetes. Relatively, there are significantly less people with diabetes that drink

heavily than those who aren't diagnosed with diabetes. This could mean that alcohol consumption isn't a very significant factor in developing diabetes.



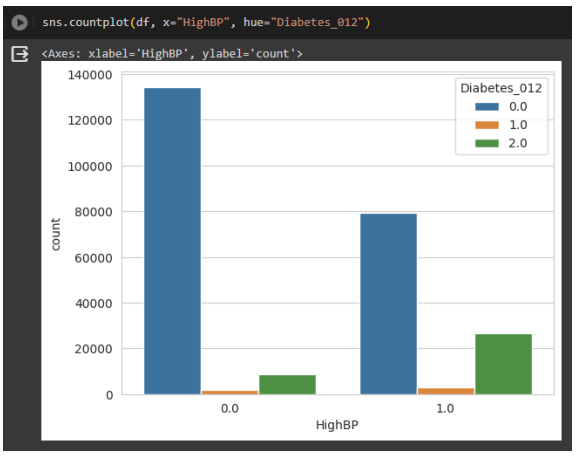


Figure 10 & 11: High cholesterol and High BP count plots

There is a greater ratio of people with high blood pressure and high cholesterol who have been diagnosed with diabetes compared to those who have not. We can assume that these are strong indicators that would be relevant to our prediction model.

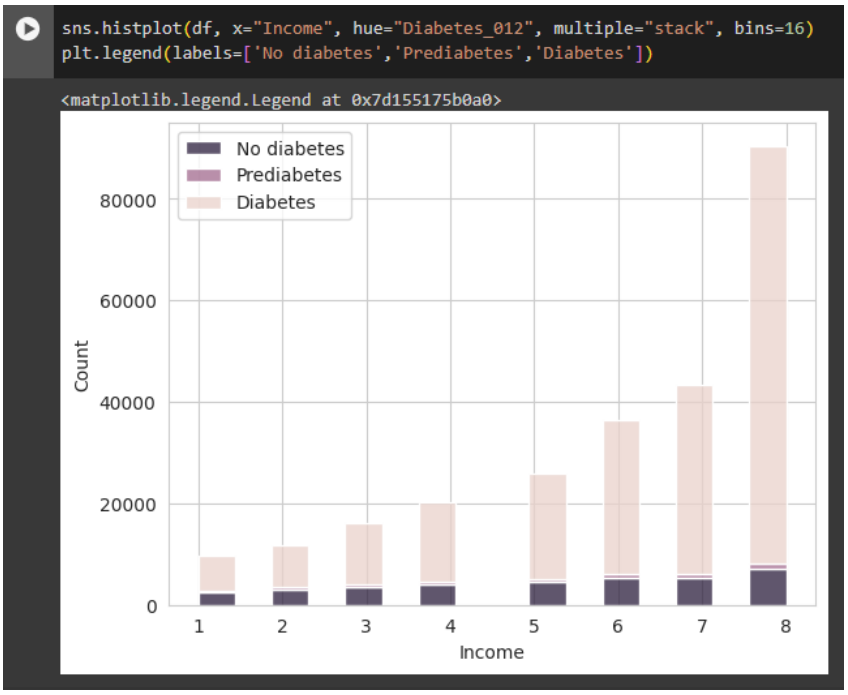


Figure 12: Relationship between income and presence of diabetes

From the histogram shown in Figure 12, we can see that there is a greater prevalence of diabetes amongst respondents in lower income groups. As mentioned before, this can be caused by poorer diets, healthcare costs and general health.

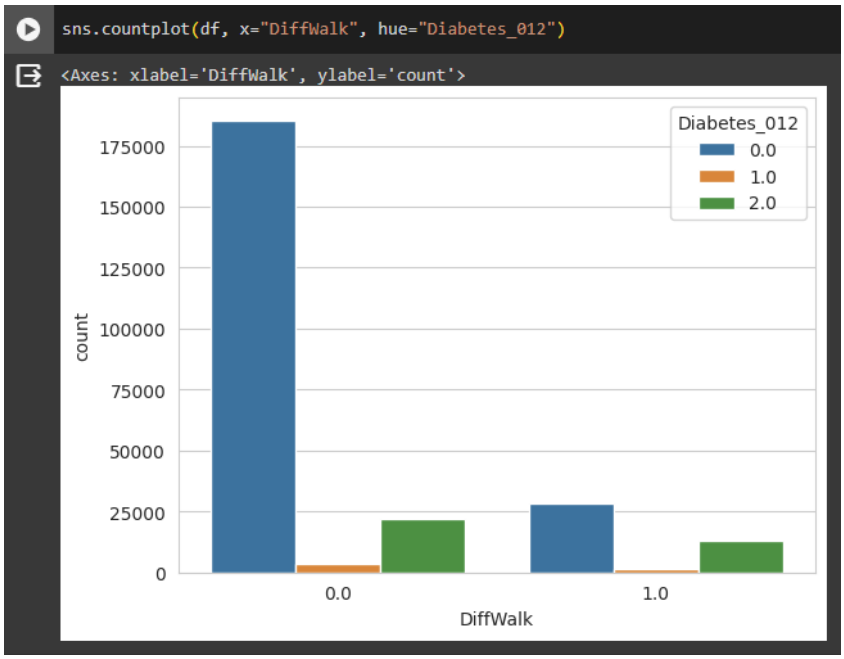


Figure 13: Count plot of 'DiffWalk' variable grouped by diagnosis

From the plot we can see that there is a small percentage of those who have difficulty walking, but of those people a great number of them are diagnosed with diabetes.

2.3 EDA conclusions

From our observations made through EDA, our dataset shows that:

1. There is a significant imbalance in our data, with individuals who aren't diagnosed with diabetes dominating the dataset
2. Even with a slight imbalance of genders, females appear to have a greater likelihood of developing diabetes
3. Older individuals are in fact more at risk of developing diabetes
4. Diabetes is more prevalent in those who are overweight, with higher BMIs
5. Heavy alcohol consumption isn't a prevalent factor in developing diabetes, whereas smoking is
6. People with high cholesterol or blood pressure are more at risk of developing diabetes
7. Low-income populations are more likely to develop diabetes
8. People with difficulty walking are more at risk of developing diabetes

3 Experimental Design

3.1 Identification of your chosen supervised learning algorithm(s)

The supervised learning algorithm chosen for this model is the Decision Tree algorithm. The Decision Tree algorithm works by mapping our features in a tree with nodes, representing decision points, and branches, representing each outcome. This algorithm can be applied to both classification and regression problems but excels in classification tasks as it can handle categorical features more effectively. It's also very efficient in training and prediction making compared to other algorithms. The Decision Tree algorithm is also easily visualised, as seen in Figure 11.

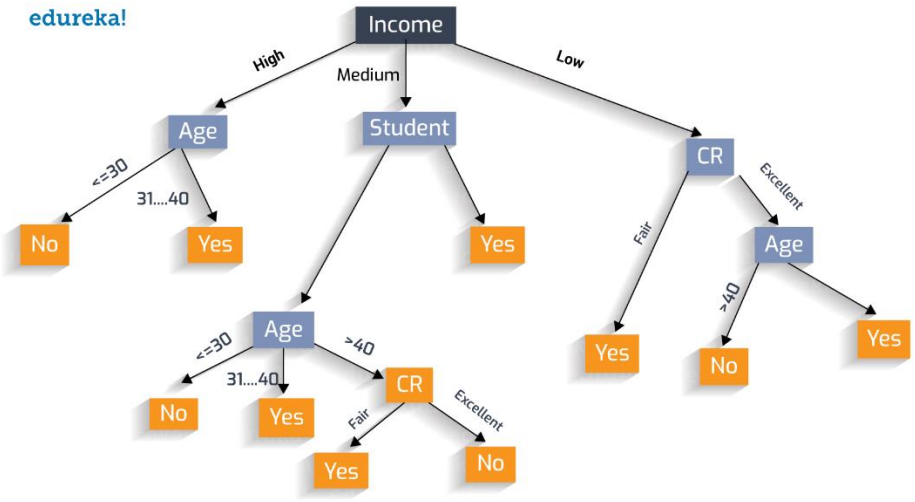


Figure 14: Visualisation of the Decision Tree algorithm (Edureka, 2015)

3.2 Identification of appropriate evaluation techniques

For our model we will be evaluating it on its precision, accuracy, and f1-score. This will provide us with a balanced measure of our model's performance.

F1-score scores are also effective with imbalanced datasets, where predictions may be inaccurate. The metric takes both false positives and false negatives into account, making it more reliable for evaluation. For our target variable, a false positive would be when the model falsely predicts that an individual has diabetes, where the person does not. A false negative would occur when the model falsely predicts that someone does not have diabetes, when they do.

Each of these metrics will be visualised using SciKit Learn's classification report metric. This function also provides accuracy, macro average and weighted average scores, which could also be useful.

3.3 Data cleaning and Pre-processing transformations

To clean our data, we must understand the type of data given to us. First, we will explore the data types of the features in our dataset.

```
df.dtypes
```

```
Diabetes_012              float64
HighBP                    float64
HighChol                  float64
CholCheck                 float64
BMI                       float64
Smoker                    float64
Stroke                    float64
HeartDiseaseorAttack      float64
PhysActivity              float64
Fruits                    float64
Veggies                   float64
HvyAlcoholConsump         float64
AnyHealthcare             float64
NoDocbcCost               float64
GenHlth                   float64
MentHlth                  float64
PhysHlth                  float64
DiffWalk                  float64
Sex                       float64
Age                       float64
Education                 float64
Income                    float64
dtype: object
```

Figure 15: Data types

We can see that all feature data types are float. To make it easier to work with and for cleaner looking plots, we will convert all float types into int.

```
df.astype(int)
```

	Diabetes_012	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke
0	0	1	1	1	40	1	0
1	0	0	0	0	25	1	0
2	0	1	1	1	28	0	0
3	0	1	0	1	27	0	0
4	0	1	1	1	24	0	0
...	...	...	...	...	...	...	...
253675	0	1	1	1	45	0	0
253676	2	1	1	1	18	0	0
253677	0	0	0	1	28	0	0
253678	0	1	0	1	23	0	0
253679	2	1	1	1	25	0	0

253680 rows × 22 columns

Figure 16: Converting floats to int

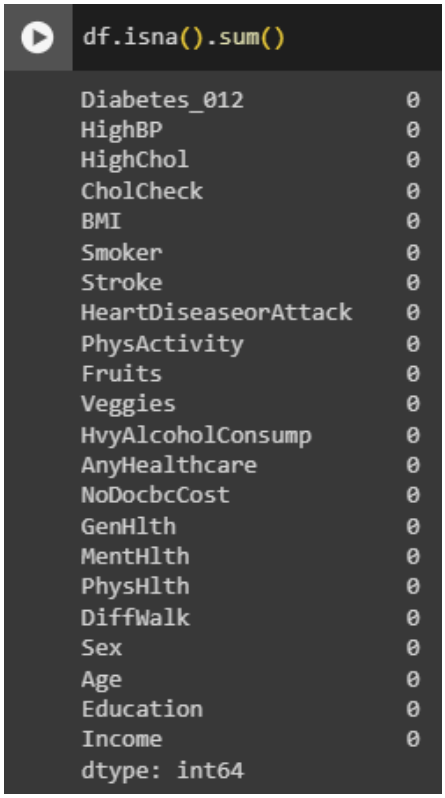


Figure 17: Checking dataset for missing values

Any missing values in a dataset can introduce bias and can result in errors with certain algorithms. Checking for any missing values, we can see that our dataset has already been cleaned.

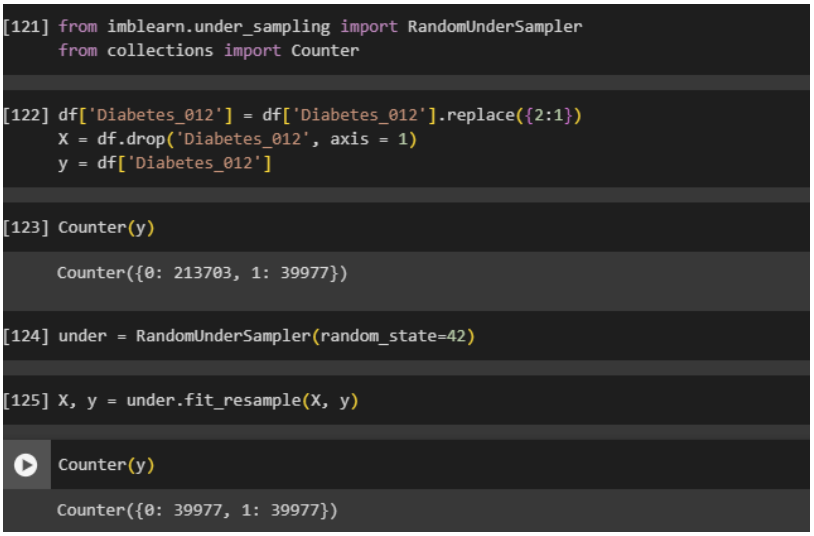


Figure 18: Under sampling diabetes classes

As observed in our EDA our dataset is heavily imbalanced, so we will be under sampling the number of respondents who are classed with no diabetes. Here we have done this by use the RandomUnderSampler function to match the size of our classes. We have also combined the prediabetes and diabetes classes due to the reason described in our EDA.

3.4 Limitations and Options

One limitation previously found was the imbalance in respondents who were and weren't diagnosed with diabetes. This was solved through undersampling, reducing the 'no diabetes' class with a random sample to the same size as the 'diabetes' class.

Another limitation is the number of features in our dataset. As we are using the Decision Tree algorithm, the use of many features may not be beneficial for a model as it can create a tree structure that is too complex. This is an example of overfitted data, where the model becomes too specific and isn't capable of accurately predicting data.

4 Predictive Modelling / Model Development

4.1 Splitting the dataset

Also referred to as a "train-test split", this is a fundamental step in machine learning where a dataset is split into training and testing data. The training data is the "seen" data that will be used to build the model from a chosen algorithm, and the testing data is the "unseen" data that the model will be evaluated on. We will also have a set of validation data which will be used for further evaluation and hyperparameter tuning.

This process helps to prevent data leakage, in which data from outside the training set is used in the training phase. This can result in a biased model that may not perform well in evaluation.

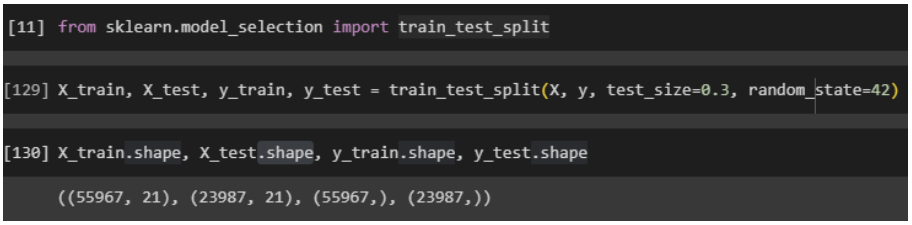


Figure 19: Using 'train_test_split' to split dataset

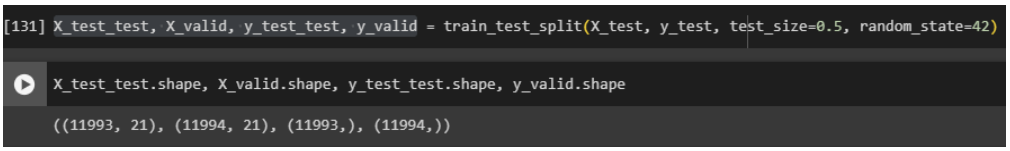


Figure 20: Making second split for validation data

Previously when we balanced our dataset, we created a new dataset 'X', dropping the "Diabetes_012" feature as this is our target variable. The other dataset 'y' consists of the target variable on its own. The train_test_split function is then used to create a split. The ratio that our data is split into won't drastically change our model's performance, but it is usually preferable to have more data for training. For our first split we used a 70:30 split, as seen from the 'test_size' parameter we have set. The random_state parameter essentially acts a seed when randomly splitting our data. When set to 0, the split will be random each time it is executed. If set to 42 in our case, it will be a different split to a random_state of 0 but will remain the same if the value isn't changed. The second split is then made, resulting in a 70:15:15 split of training, testing and validation data.

4.2 The predictive modelling process

We now have a split dataset from our pre-processed data, ready to use for training. We will first create an instance of the support vector machine algorithm using its default parameters and fit a model to our training data. A description of the Decision Tree parameters and its default values can be found in Table 4.

Table 4: Decision Tree parameters and default values (scikit-learn, 2019)

Parameter	Description	Default value
criterion	The function to measure the quality of a split	gini
splitter	The strategy used to choose the split at each node	best
max_depth	The maximum depth of the tree	None
min_samples_split	The minimum number of samples required to split an internal node	2
min_samples_leaf	The minimum number of samples required to be at a leaf node	1
min_weight_fraction_leaf	The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node	0.0
max_features	The number of features to consider when looking for the best split	None
random_state	Controls the randomness of the estimator	None
max_leaf_nodes	Grow a tree with max_leaf_nodes in best-first fashion	None
min_impurity_decrease	A node will be split if this split induces a decrease of the impurity greater than or equal to this value	0.0
class_weight	Weights associated with classes in the form {class_label: weight}	None
ccp_alpha	Complexity parameter used for Minimal Cost-Complexity Pruning	0.0

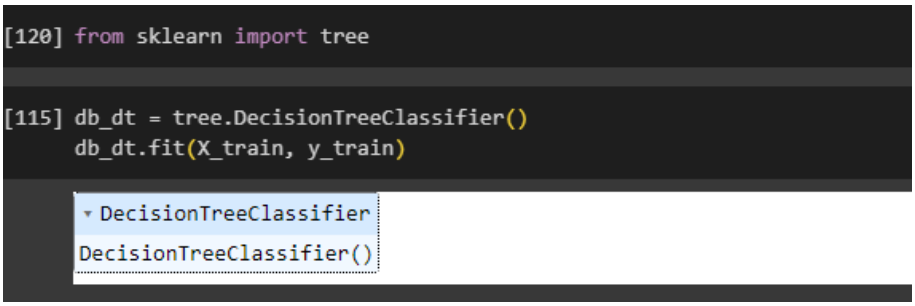


Figure 21: Creating and fitting a Decision Tree model

Here we have created an instance of the Decision Tree algorithm without changing its parameters and is then fitted to our target and features. We then use the predict function to create predictions of our target variable in our test data using the newly fitted model.

4.3 Evaluation results on “seen” data

```
[118] db_pred = db_dt.predict(X_train)
```

Figure 22: Creating predictions from fitted model on “seen” data

Using the classification report, we can look at the precision, recall and f1-score of the results from our model in the form of a text report. From Figure 19, we can see that the model performs well across each metric, meaning the model has fit well to the training data.

```
[140] from sklearn.metrics import classification_report

db_cr = classification_report(y_train, db_pred)
print(db_cr)
```

	precision	recall	f1-score	support
0	0.99	1.00	0.99	27954
1	1.00	0.99	0.99	28013
accuracy			0.99	55967
macro avg	0.99	0.99	0.99	55967
weighted avg	0.99	0.99	0.99	55967

Figure 23: Classification report of predictions on training data

5 Evaluation and further modelling improvements

```
[147] db_pred = db_dt.predict(X_valid)

db_cr = classification_report(y_valid, db_pred)
print(db_cr)
```

	precision	recall	f1-score	support
0	0.64	0.65	0.65	6057
1	0.64	0.63	0.64	5937
accuracy			0.64	11994
macro avg	0.64	0.64	0.64	11994
weighted avg	0.64	0.64	0.64	11994

Figure 24: Classification report of predictions on validation data

The classification report displayed in Figure 19 shows a great difference in performance of our model on the validation data, with predictions much worse than before. This could likely be caused by overfitted data. Overfitting in our data can be caused by noisy data, or a dataset that is too complex for its purpose. This could also be caused by the algorithm's instability. As we have a large set of features, some of them could be redundant to our model. This can be solved through feature selection.

```
df = df.drop(['HvyAlcoholConsump','NoDocbcCost'], axis=1)
```

Figure 25: Dropping redundant features

```
[71] db_pred = db_dt.predict(X_valid)

db_cr = classification_report(y_valid, db_pred)
print(db_cr)
```

	precision	recall	f1-score	support
0	0.65	0.66	0.65	6057
1	0.65	0.64	0.64	5937
accuracy			0.65	11994
macro avg	0.65	0.65	0.65	11994
weighted avg	0.65	0.65	0.65	11994

Figure 26: Evaluation of model after further processing

In Figure 20, we dropped two columns that were irrelevant to our model or that didn't correlate with other features. The classification report shows little to no improvement in the model's performance, with a slight improvement in precision and recall scores.

```
[106] from sklearn.model_selection import GridSearchCV

[115] param_grid = {
          'max_depth': [None, 10, 12, 16, 20, 24],
          'max_features': [10, 14, 18, 20]
      }

[116] grid_search = GridSearchCV(db_dt, param_grid= param_grid, cv=5,
                                 scoring ='neg_mean_squared_error',
                                 return_train_score= True)
```

Figure 27: Evaluation of model after further processing

Here we are attempting hyperparameter tuning, where we are trying to find the optimal set of parameters for our specific model. Using GridSearchCV, we can perform a grid search of parameters we specify. In our case, since the Decision Tree algorithm can become unstable with more complex data, we are limiting the depth of the tree and the number of features it can use, seen in Figure 22.

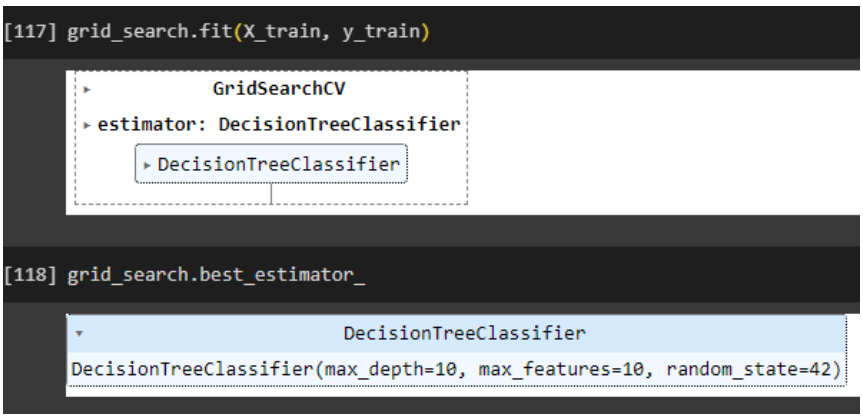


Figure 28: Performing a grid search and finding the best parameters

Figure 23 shows the best parameters found in our grid search. It has given us a maximum tree depth of 10 and a maximum feature number of 10. We can then create a new model from these hyperparameters and see if its performance improves.

```
[119] grid_search.best_params_

     {'max_depth': 10, 'max_features': 10}

[122] best_dt = grid_search.best_estimator_

     best_dt.score(X_valid, y_valid)

     0.725446056361514

[124] db_pred = best_dt.predict(X_valid)

     db_cr = classification_report(y_valid, db_pred)
     print(db_cr)

                  precision    recall  f1-score   support

               0       0.74      0.71      0.72      6057
               1       0.72      0.74      0.73      5937

        accuracy                           0.73     11994
       macro avg       0.73      0.73      0.73     11994
    weighted avg       0.73      0.73      0.73     11994
```

Figure 29: Retrieving hyperparameters from grid search and fitting new model with the new parameters

As seen in the classification report, the performance of the model has greatly improved across all metrics, each increasing by about 0.1 in score. This proves that our initial data was overfitted to the algorithm, as it produced a tree that was too complex.

6 Conclusion

6.1 Summary of results

The initial model performed well on our training set but didn't perform as expected on our testing data, producing low scores across all our evaluation metrics. After considering the limitations of the chosen algorithm, we attempted to apply hyperparameter tuning to our model. By limiting the complexity of the decision tree and the features used, we were able to fit a more accurate model to our data, with an improved f1-score of 0.73. This final model saw a significant increase in performance but is still not accurate enough for its real-world application. A model with better performance would be favourable as inaccurate predictions when diagnosing diabetes could cause implications for patients.

6.2 Reflection on Individual Learning

Individually, this coursework helped in understanding the importance of data-based decision making and the complexity of model development. Throughout the module I've gained more knowledge on EDA and why it's equally important as model development. I've also faced the limitations of a machine learning algorithms and the strategies used to prevent them.

As someone who has recently found a strong fascination for A.I., the module has been enjoyable in developing my skills as a data scientist and engineer, introducing me to various machine learning skills and techniques that can be applied to real-world problems.

I may consider taking on more machine learning tasks in the future, or even data exploration, to further grow my understanding of other machine learning algorithms, as well as the general applications of machine learning in real use-cases.

7 References

CDC (2021) Defining Adult Overweight and Obesity, Centers for Disease Control and Prevention. Available at: https://www.cdc.gov/obesity/basics/adult-defining.html#:~:text=BMI%20is%20a%20person

Kautzky-Willer, A. et al. (2016) Sex and Gender Differences in Risk, Pathophysiology and Complications of Type 2 Diabetes Mellitus, Endocrine Reviews, Volume 37, Issue 3. Available at: https://doi.org/10.1210/er.2015-1137

Mordarska, K. & Godziejewska-Zawada, M. (2017). Diabetes in the elderly. Available at: https://doi.org/10.5114/pm.2017.68589

Diabetes UK (2018) Diabetes and blood pressure, Diabetes UK. Available at: https://www.diabetes.org.uk/guide-to-diabetes/managing-your-diabetes/blood-pressure

Diabetes UK (2017) Cholesterol and diabetes, Diabetes UK. Available at: https://www.diabetes.org.uk/guide-to-diabetes/enjoy-food/eating-with-diabetes/managing-other-medical-conditions/cholesterol-and-diabetes

Hsu, C. et al. (2012). Poverty increases type 2 diabetes incidence and inequality of care despite universal health coverage. Available at: https://doi.org/10.2337/dc11-2052

Hales, C. et al. (2017) 'Prevalence of obesity among adults and youth: United states, 2015-2016 key findings'. Available at: https://www.cdc.gov/nchs/data/databriefs/db288.pdf

CDC (2015) Calculated Variables in the 2015 Data File of the Behavioral Risk Factor Surveillance System. Available at: https://www.cdc.gov/brfss/annual_data/2015/pdf/2015_calculated_variables_version4.pdf

Edureka (2015) Decision Tree | Decision Tree Introduction With Examples. Available at: https://www.edureka.co/blog/decision-trees/

scikit learn (2019) sklearn.tree.DecisionTreeClassifier — scikit-learn 0.22.1 documentation, Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html