Medical patient data analysis with machine learning

Prediction and clustering



Andreas Wendel

EC Utbildning

DS project

2024–10

# Abstract

Machine learning and AI is a very trending subject and the technology is growing exponentially. Being able to utilize Machine learning technologies we could improve our healthcare system and decrease waiting time for our patient.

By utilizing Machin learning we could theoretically predict in advanced, helping our patients and doctors save a lot of time and work.

Using Machin learning we could also cluster data which could unravel new information about lifestyles and habits that could save future individuals

Innehållsförteckning

[Abstract 2](#_Toc181283939)

[1 Purpose 1](#_Toc181283940)

[1.1 Problems 1](#_Toc181283941)

[1.2 Datasetet, ifrån simpelt – Exempel 1](#_Toc181283942)

[2 Models, evaluation and analysis 2](#_Toc181283943)

[2.1 Supervised learning models for predicting 2](#_Toc181283944)

[2.1.1 Logistic Regression 2](#_Toc181283945)

[2.1.2 Random Forest Classifier 3](#_Toc181283946)

[2.1.3 SVC 3](#_Toc181283947)

[2.1.4 XGBoost (Extreme Gradient Boosting) 4](#_Toc181283948)

[2.1.5 Hyperparameter optimering (GridSearchCV) 5](#_Toc181283949)

[2.1.6 Model evaluation 5](#_Toc181283950)

[2.2 Unsupervised learning models for clustering 7](#_Toc181283951)

[2.2.1 Kmeans 7](#_Toc181283952)

[2.2.2 HDBSCAN 8](#_Toc181283953)

[2.2.3 Cluster model evaluation, Silhouette score 8](#_Toc181283954)

[2.3 Analyzing cluster 9](#_Toc181283955)

[2.3.1 Hypothesis test on features 9](#_Toc181283956)

[3 Process 10](#_Toc181283957)

[3.1 Dataset information and EDA 10](#_Toc181283958)

[3.2 Prediction models 16](#_Toc181283959)

[3.2.1 Data imbalance 16](#_Toc181283960)

[3.2.2 Model comparison 18](#_Toc181283961)

[3.2.3 Optimization and feature importance 19](#_Toc181283962)

[3.3 Clustering models 20](#_Toc181283963)

[3.3.1 Model evaluation 20](#_Toc181283964)

[4 Results and discussion 21](#_Toc181283965)

[4.1 Clustering analysis 21](#_Toc181283966)

[4.1.1 Cluster number 26 22](#_Toc181283967)

[4.1.2 Cluster number 45 23](#_Toc181283968)

[4.2 Cluster comments 25](#_Toc181283969)

[4.3 Prediction model 26](#_Toc181283970)

[5 Conclussion 27](#_Toc181283971)

[5.1 Can you predict heart disease 27](#_Toc181283972)

[5.2 What supervised learning model predicts best 27](#_Toc181283973)

[5.3 What model is used for unsupervised learning 27](#_Toc181283974)

[5.4 How has the clusters been assigned to the data 27](#_Toc181283975)

[5.5 Has there been any benefits or any new information through clustering the data 27](#_Toc181283976)

[6 Sources 28](#_Toc181283977)

# Purpose

Attempting to predict a health condition using other feature of information is a difficult task, most solely because there are multiple factors to why an individual may succumb to such conditions, such factors may not even be able to be documented.

Everything you consume knowing and unknowing to the lifestyle and the habits you have and even genetics you inherit may lay cause to the different diseases and conditions that you may acquire throughout your lifetime, but having advanced models even though they may be wrong might just save you by giving you the heads up.

Most conditions are not noticeable before its to late such as Cancer or Cardio vascular but if a model can somehow give you and early warning than you may have just prevented a disaster. The Difficult part comes when a model negatively predicts a positive outcome that could make matters even worse, and such ethics about models may be discussed as well, thought some might argue that it would be the same as if a certified doctor wrongly diagnosed a patient.

Another great part about machine learning is the possibility to find patterns within data. By attempting to find patterns in data we might find a habit or lifestyle that would lead to certain conditions. It might be as simple as groups of individuals who consume unhealthy foods have diabetes. Other similar patterns may be found for other conditions which might help individuals steer clear of certain habits that may cause unwanted conditions. By attempting to cluster our dataset we might find hidden pattern that has yet to be discovered, these hidden patterns may be useful or they may be obvious or they may even be useless. But finding these patterns is a way for us to research ways to decrease the outcome of certain diseases or unhealthy conditions

## Problems

1. Can you predict heart disease
2. What supervised learning model predicts best
3. What model is used for unsupervised learning
4. How has the clusters been assigned to the data
5. Has there been any benefits or any new information through clustering the data

## Datasetet, ifrån simpelt – Exempel

This dataset is taken from Kaggle where the author claims that the dataset is a BRFSS dataset.

The Behavioral Risk Factor Surveillance System (BRFSS) is the nation’s premier system of health-related telephone surveys that collect state data about U.S. residents regarding their health-related risk behaviors, chronic health conditions, and use of preventive services

The author has already preprocessed the dataset from 304 unique features to hand-picked 19 features that relates to lifestyle factors that may contribute to being in risk of CVD.

A full summary of the whole dataset can be found [here](https://eajournals.org/ejcsit/vol11-issue-3-2023/integrated-machine-learning-model-for-comprehensive-heart-disease-risk-assessment-based-on-multi-dimensional-health-factors/).

# Models, evaluation and analysis

## Supervised learning models for predicting

A supervised learning model is a type of machine learning algorithm that is trained using labeled data, which means that the input data is paired with the correct output. The primary purpose of supervised learning is to learn from inputs to outputs using different methods such as assigning weights, finding hyperplane or other methods, so that the model can make accurate predictions or classifications on new, unseen data.

### Logistic Regression

**Logistic regression** is a statistical model used for binary classification tasks, which involves predicting the probability of a binary outcome (for example, yes/no, true/false, 0/1). Despite its name, it’s more of a classification algorithm than a regression one and is widely used in machine learning for tasks such as spam detection, disease diagnosis, and customer segmentation.

It works by applying a **sigmoid function** to a linear combination of input features, converting any real-valued number into a probability between 0 and 1.

During training, the model learns coefficients (weights) that minimize the difference between predicted probabilities and the actual class labels. This process is optimized using a method like **gradient descent**. Predictions are then made by assigning the class based on a probability threshold (usually 0.5).

**Strengths**:

* **Simplicity and Speed**: Logistic regression is computationally efficient and straightforward to implement, making it suitable for large datasets and fast predictions.
* **Feature Importance and Interpretability**: Weights in logistic regression are easy to interpret, allowing insight into how each feature affects the outcome.

**Limitations**:

* **Linearity Assumption**: Logistic regression assumes a linear relationship between input features and the log-odds of the outcome. It struggles with complex data structures or non-linear relationships, where other models may perform better.
* **Sensitive to Outliers/noisy data**: Logistic regression can be sensitive to outliers in the data, which can skew the model. Proper data preprocessing, such as scaling and outlier removal, is necessary Sensitive to imbalanced data, which may require adjustments (e.g., using class weights).

### Random Forest Classifier

A **Random Forest Classifier** is a machine learning algorithm that uses an ensemble of decision trees (se source) to make predictions for classification tasks. In this model, multiple decision trees are trained on random subsets of the data and features. Each tree independently makes a prediction, and the final prediction is determined by the majority vote of all trees, improving accuracy and robustness compared to a single tree.

During training, each tree focuses on different parts of the dataset, and the randomness reduces overfitting. The combination of diverse trees makes Random Forest especially effective for handling complex patterns, yielding high accuracy on a variety of datasets.

**Strengths:**

* **High Accuracy**: Due to the ensemble approach, random forests generally provide better accuracy than single decision trees.
* **Resistant to Overfitting**: The randomness in data and feature sampling helps reduce overfitting, making it generalize well to new data.

**Limitations of Random Forest Classifier:**

* **Computationally Intensive**: Training a large number of trees can be slow and require significant memory, especially with large datasets.
* **May Overfit with Very Noisy Data**: Although resistant, random forests can still be affected by noisy data, especially if there is a very large number of trees.

### SVC

**Support Vector Classifier** (SVC) is a classification method that works by finding the best boundary (hyperplane) to separate data points of different classes in a high-dimensional space. SVC aims to maximize the margin between this hyperplane and the closest data points from each class, known as support vectors. If the data is not linearly separable, SVC uses a “kernel trick” to transform the data into a higher-dimensional space where a separating hyperplane can be found. This makes SVC effective for handling complex data structures and boundaries.

**Strengths:**

* **Effective for High-Dimensional Data**: SVC works well with data that has many features, particularly when the number of dimensions is greater than the number of samples.
* **Robust to Overfitting**: Due to margin maximization, SVC generalizes well to new data, especially when a good regularization parameter is chosen.

**Limitations:**

* **Computational Complexity**: SVC can be slow and memory-intensive, especially with large datasets, due to the quadratic complexity of finding the optimal hyperplane.
* **Not Ideal for Large Datasets**: Training an SVC model on a large dataset can be computationally prohibitive and may not always yield better results than simpler models.
* **Requires Careful Parameter Tuning**: SVC is sensitive to the choice of kernel and regularization parameters, which can significantly impact performance and require tuning.
* **Difficult with Noisy Data**: Outliers and noisy data points can affect the hyperplane, potentially leading to misclassification.

### XGBoost (Extreme Gradient Boosting)

**XGBoost** is a powerful, efficient machine learning algorithm that is based on gradient boosting, a technique that builds an ensemble of decision trees to improve model accuracy. XGBoost builds upon: supervised machine learning, decision trees, ensemble learning, and gradient boosting.

A Gradient Boosting Decision Trees (GBDT) is a decision tree [ensemble learning algorithm](https://en.wikipedia.org/wiki/Ensemble_learning" \t "_blank) similar to random forest. The term “[Gradient boosting](https://developer.nvidia.com/blog/gradient-boosting-decision-trees-xgboost-cuda/)” comes from the idea of “boosting” or improving a single weak model by combining it with a number of other weak models in order to generate a collectively strong model. Gradient boosting sets targeted outcomes for the next model in an effort to minimize errors. Targeted outcomes for each case are based on the gradient of the error (hence the name gradient boosting) with respect to the prediction. With XGBoost, trees are built in parallel, instead of sequentially like GBDT making processing faster when training, this is possible as XGBoost uses the Thousands of cores from the GPU instead of the CPU cores.

More in depth explanation can be found in the source [XGBoost](https://www.nvidia.com/en-us/glossary/xgboost/)

**Strengths:**

* **Highly Accurate**: XGBoost is often one of the top-performing algorithms for structured data, frequently winning machine learning competitions.
* **Speed and Scalability**: XGBoost is optimized for efficiency, leveraging parallel processing and utilizes GPU cores instead of CPU cores.

**Limitations:**

* **Parameter Tuning Required**: To achieve optimal performance, XGBoost requires tuning many hyperparameters, which can be time-consuming.
* **Less Effective for Unstructured Data**: XGBoost is primarily suited for structured/tabular data and is not typically used for tasks like image or text classification.
* **Sensitive to Imbalanced Data**: Without adjustments, XGBoost can struggle with highly imbalanced classes, sometimes requiring techniques like re-weighting or resampling.

### Hyperparameter optimering (GridSearchCV)

Hyperparameter optimization is the process of finding the best combination of hyperparameters to maximize a model’s performance. Unlike model parameters, which are learned during training, hyperparameters are set before training begins and significantly impact the model’s effectiveness. GridSearchCV is a popular technique for hyperparameter tuning, systematically searching through a predefined grid of hyperparameter values to identify the optimal set for a model.

**How GridSearchCV Works:**

1. **Define Hyperparameter Grid**: First, specify a range of values for each hyperparameter in a grid format. For instance, for a Support Vector Classifier (SVC), you might define a grid with different values for C, gamma, and kernel types.
2. **Cross-Validation for Evaluation**: GridSearchCV divides the training data into several folds (or subsets). For each combination of hyperparameters in the grid, the model is trained and validated across all folds, resulting in multiple scores per hyperparameter combination.
3. **Select Best Hyperparameters**: After evaluating all combinations, GridSearchCV selects the combination with the highest average cross-validation score. This set of hyperparameters is considered optimal and is then used to train the final model on the entire training dataset.

### Model evaluation

An Evaluation process is used to determine which model performs the best.

#### Confusion matrix

A confusion matrix is a powerful tool for evaluating the performance of a classification model. It provides a visual representation of how well the model's predictions align with the actual outcomes.

A confusion matrix is typically organized as a square table with rows representing the actual classes and columns representing the predicted classes. For a binary classification problem, the confusion matrix consists of four key components: Tabell 1

* **True Positive (TP)**: The number of instances that were correctly predicted as positive.
* **True Negative (TN)**: The number of instances that were correctly predicted as negative.
* **False Positive (FP)**: The number of instances that were incorrectly predicted as positive. This is also known as a Type I error.
* **False Negative (FN)**: The number of instances that were incorrectly predicted as negative. This is known as a Type II error.​

Using these representations, we can do certain metric calculations such as: **Accuracy, Precision, Recall, F1 Score, Macro Average, Weighted Average.** In depth calculation can be found [here](https://www.kdnuggets.com/2023/01/micro-macro-weighted-averages-f1-score-clearly-explained.html).

Tabell 1

|  |  |  |
| --- | --- | --- |
|  | **Predicted Positive** | **Predicted Negative** |
| **Actual Positive** | True Positive (TP) | True Positive (TP) |
| **Actual Negative** | False Positive (FP) | True Negative (TN) |

#### ROC and AUC

The Receiver Operating Characteristic ([ROC](https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5)) curve and the Area Under the Curve ([AUC](https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5)) are widely used evaluation metrics for binary classification models. They help assess the trade-offs between sensitivity and specificity at various threshold settings.

The ROC curve is a graphical representation that illustrates the performance of a binary classification model as the decision threshold varies. It plots two key metrics:

* **True Positive Rate (TPR)**, also known as **Recall** or **Sensitivity**, which is defined as:
  + ​
  + where TP is the number of true positives, and FN is the number of false negatives.
* **False Positive Rate (FPR)**, which is defined as:
  + where FP is the number of false positives, and TN is the number of true negatives.

The ROC curve is created by plotting the TPR against the FPR at different threshold levels, which means as you change the threshold for classifying an instance as positive, the TPR and FPR change accordingly.

The AUC (Area under the curve) quantifies the overall performance of the model across all thresholds. Specifically, it measures the area under the ROC curve from 0 -1 with different interpretations.

**AUC Interpretation**

* **AUC = 0.5**: The model has no discrimination ability, similar to random guessing.
* **AUC > 0.5 and < 1**: The model has some discrimination ability, with values closer to 1 indicating better performance.
* **AUC = 1**: The model perfectly classifies classes.
* **AUC = 0**: The model perfectly misclassifies classes

## Unsupervised learning models for clustering

Unsupervised learning is a type of machine learning where the model is trained on data without labeled outputs. Instead of learning from a response variable (as in supervised learning), the model seeks to identify patterns, groupings, or structures within the input data.

### Kmeans

**K-Means** is an unsupervised machine learning algorithm that partitions data into K distinct clusters based on feature similarity. With a predetermined amount of *K* clusters, the model randomly distributes *K* centroids and assign datapoints to the closest centroid while calculating the Euclidean distance from each datapoint to assigned centroid. The model then calculated a mean for each centroid and then redistribute them until the centroids stabilize or reach maximum iterations.

K-Means is a straightforward and effective clustering algorithm, but it requires careful selection of the number of clusters and consideration of its assumptions.

**Strengths:**

* **Simplicity:** K-Means is easy to understand and implement, making it a popular choice for clustering tasks**.**
* **Efficiency:** The algorithm is computationally efficient, especially for large datasets, as its time complexity is approximately *,* where *n* is the number of data points, *K* is the number of clusters, and *i* is the number of iterations.
* **Scalability:** K-Means can handle large datasets well, making it suitable for big data applications.
* **Interpretability:** The results are easy to interpret, as each point is assigned to a specific cluster, and the centroids provide a clear representation of each cluster.

**Limitations:**

* **Choosing *K*:** The number of clusters, *K*, must be specified in advance, which can be challenging. Incorrectly choosing K can lead to poor clustering results.
* **Sensitivity to Initialization:** The initial placement of centroids can affect the final clusters. K-Means can converge to local minima, leading to different results based on initialization.
* **Assumption of Spherical Clusters:** K-Means assumes that clusters are spherical and of similar size, which may not be true for all datasets. It can struggle with clusters of varying shapes and densities.
* **Sensitivity to Outliers:** Outliers can disproportionately influence the position of centroids, leading to skewed clusters.
* **Limited to Linear Boundaries:** K-Means creates clusters based on distances and may not capture complex, non-linear relationships between data points.

### HDBSCAN

HDBSCAN is a clustering algorithm developed by Campello, Moulavi, and Sander. It extends DBSCAN (Density-Based Spatial Clustering of Applications with Noise) by converting it into a hierarchical clustering algorithm, and then using a technique to extract a flat clustering based in the stability of clusters. It is particularly effective for clustering data that has varying densities and for identifying clusters of different shapes and sizes. Aswell as handling noisy datapoints in the dataset. HDBSCANgroups points that are closely packed together, marking them as dense regions, while points that lie alone in low-density regions are considered noise.

**Strengths**

* **Robustness**: Can identify clusters of varying shapes and densities.
* **Noise Handling**: Effectively distinguishes noise from clusters.

**Limitations**

* **Parameter Sensitivity**: Performance can be sensitive to the choice of parameters, such as minimum cluster size.
* **Complexity**: The hierarchical approach can lead to higher computational costs compared to simpler clustering algorithms.

### Cluster model evaluation, Silhouette score

The **[Silhouette Score](https://medium.com/@hazallgultekin/what-is-silhouette-score-f428fb39bf9a)** is a metric used to evaluate the quality of clustering results. It measures how similar an object is to its own cluster compared to other clusters, providing insights into the appropriateness of the clustering configuration.

The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is:

To clarify, b is the distance between a sample and the nearest cluster that the sample is not a part of

S has a range from (-1 to 1) where 1 is far away from nearest cluster and -1 means close to nearest cluster which might mean that the datapoint has been assigned wrong.

## Analyzing cluster

Datapoints and features in cluster are analyzed to determent if and what kind of pattern are in a cluster.

### Hypothesis test on features

Hypothesis testing is a statistical method used to make inferences about populations based on sample data. In the context of clustering, hypothesis tests can help evaluate the characteristics of clusters, compare clusters, and determine whether observed differences are statistically significant.

you typically start with two competing hypotheses:

* **Null Hypothesis (​)**: Assumes that there is no significant difference or relationship among the groups being tested. For example, you might state that the mean of a certain feature (e.g., height) in Cluster A is equal to that in Cluster B.
* **Alternative Hypothesis (​)**: Indicates that there is a significant difference or relationship. For example, that the mean height in Cluster A is different from that in Cluster B.

**Choose the Appropriate Test**

The choice of the hypothesis test depends on the type of data and the number of clusters being compared:

* **[ANOVA](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.f_oneway.html) (Analysis of Variance)**: ANOVA also know as F-test is used when comparing means (numerical) across three or more clusters. ANOVA tests if at least one cluster mean is different from the others.
* **[Chi-Squared Test](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2_contingency.html)**: Used for categorical variables to evaluate the association between the cluster membership and the categories.

The p-value indicates the probability of observing the data assuming the null hypothesis is true.

# Process

## Dataset information and EDA

The pure dataset contains 308854 entries with a total of 19 columns or features, 7 features are numerical:

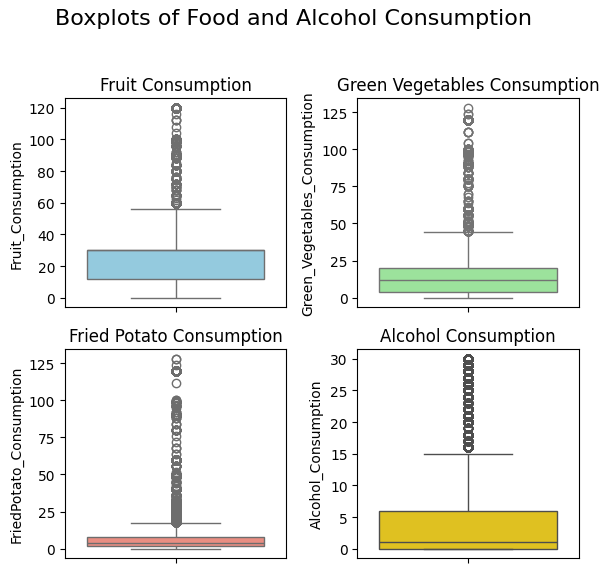
['Height\_(cm)', 'Weight\_(kg)’, 'BMI', 'Alcohol\_Consumption', 'Fruit\_Consumption', 'Green\_Vegetables\_Consumption', 'FriedPotato\_Consumption']

And 12 Categorical:

['General\_Health', 'Checkup', 'Exercise', 'Heart\_Disease', 'Skin\_Cancer', 'Other\_Cancer', 'Depression', 'Diabetes', 'Arthritis', 'Sex', 'Smoking\_History']

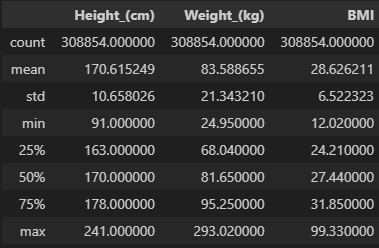
Height, Weight and BMI are self-explanatory but Consumption need further explanation. The consumption values represent consumption per each month.

Notice that all but Alcohol Consumption reach values around 120-130 meaning you have more than 1 daily consumption. Alcohol on the other hand is only calculated 1 consumption per day, meaning 1 unit of alcohol is the same as 5 units per day. See figure 1

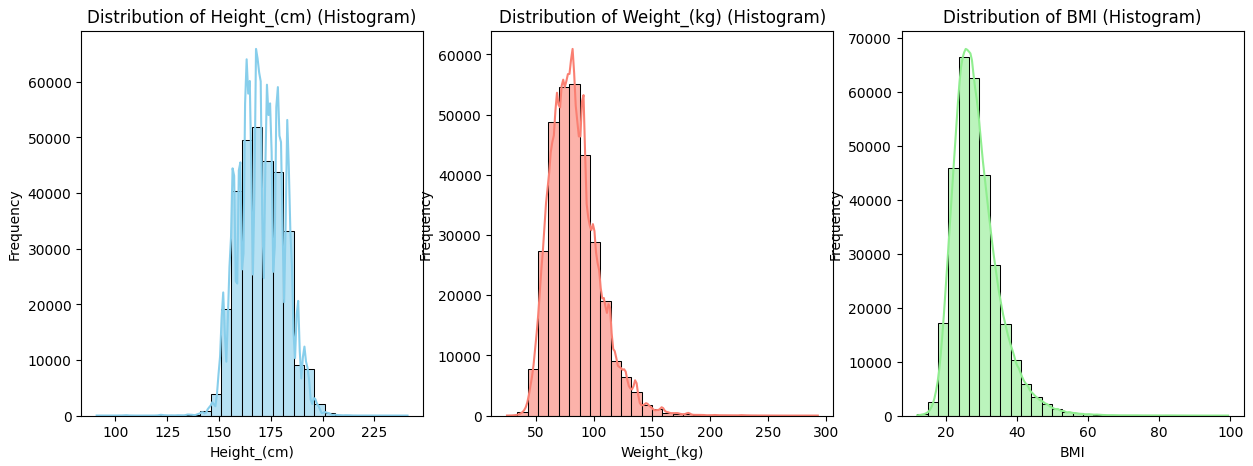
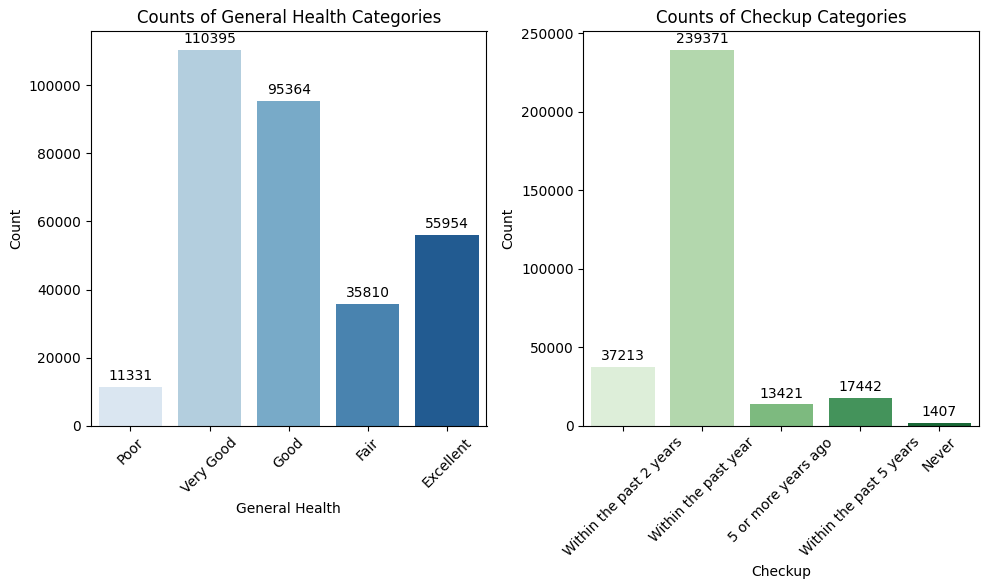


Figur 1

Looking at the numerical data for our patient we can see the distribution of height weight and bmi. Something noteworthy is that BMI is quite high, a mean of 28.6 is considered overweight there is also quite a tail on the upperside of weight and bmi. IT is important though that BMI doesn’t take into account body fat so some of these individuals might be some kind of elite trainer with high muscle mass (weight) and short height which will make the BMI metric increase. See figure 2, 3

The dataset also includes information about health and last checkups, most people do regular checkups here as well as consider their health to be good, very good or excellent. See figure 4

Figur 2



Figur 3

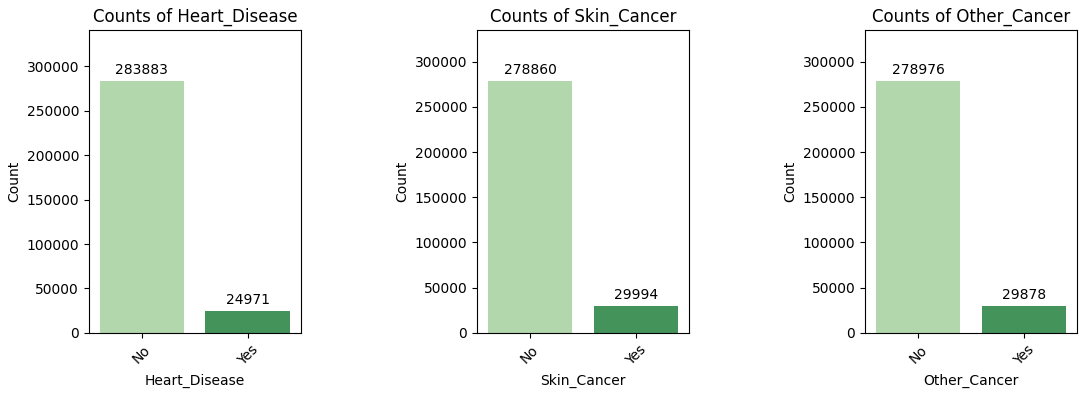
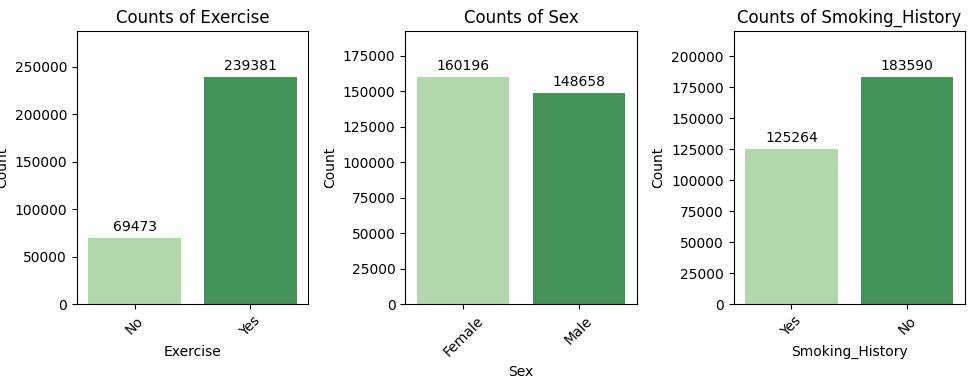
Figur 4

The dataset also includes lifestyle data such as exercise and smoking history as well as health condition such as diseases.

We can see majority of people exercise, but there is also a quite even mix of smoker and none smokers. There is also an even mix of sexes. See figure 5

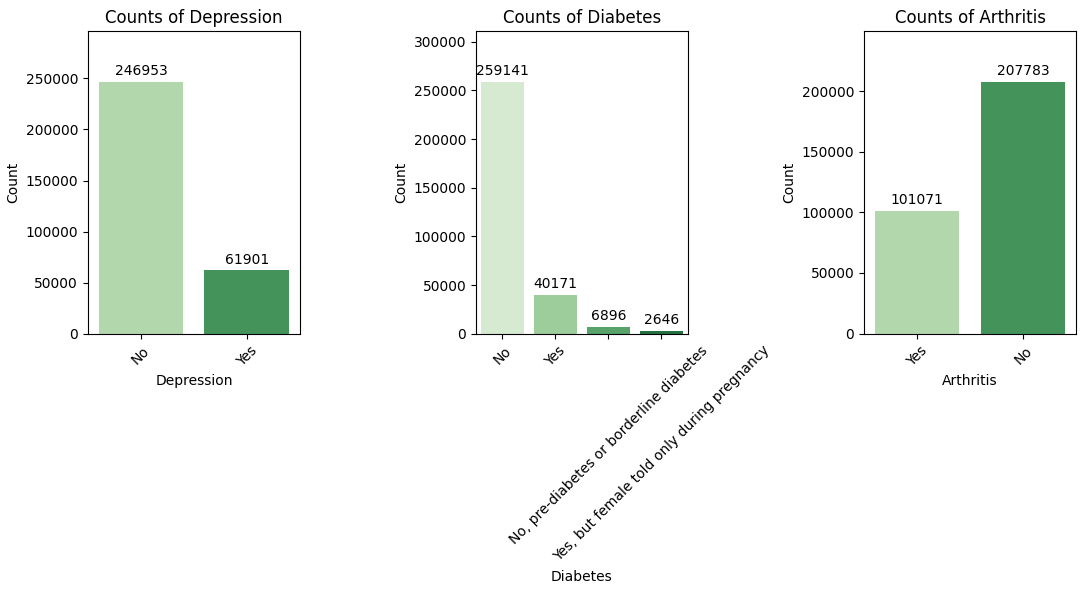
Looking at figure 6, we can see that there is a small part of individuals who are majorly sick with conditions such as cancer and cardio vascular disease.

In figure 7 we can also note that there are some individuals with Depression, Diabetes and larger population with Arthritis.

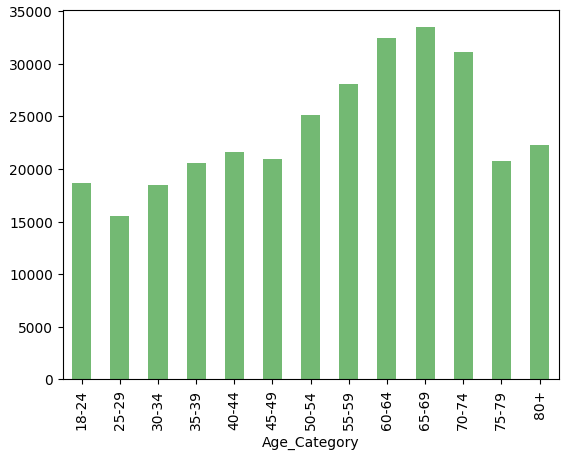


Figur 6

Figur 5

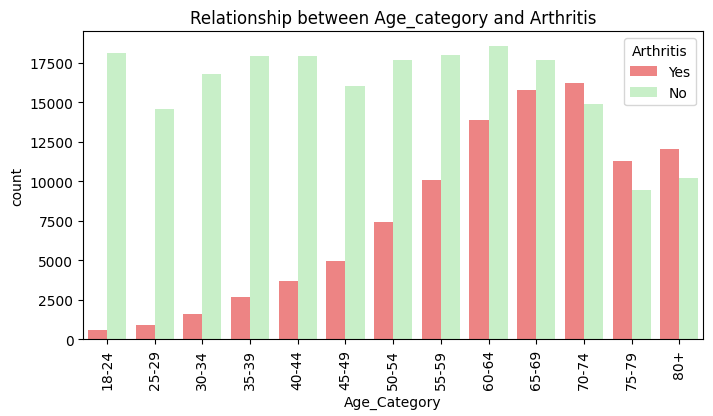
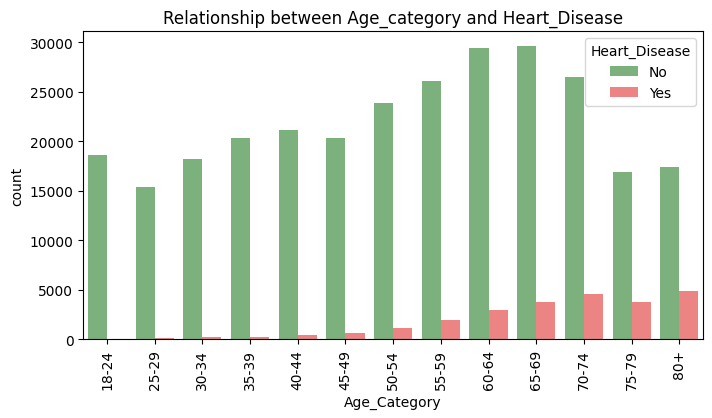
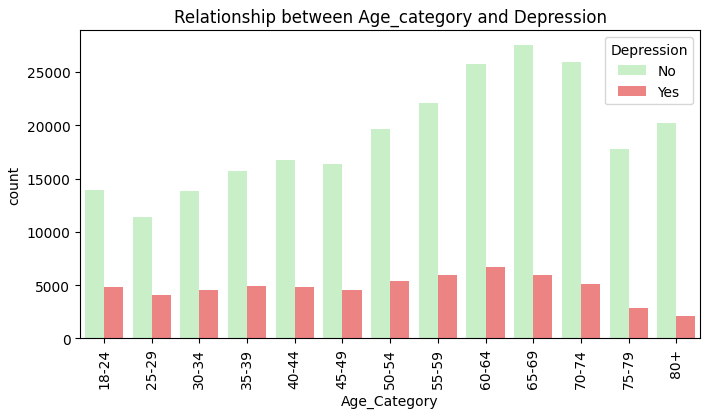
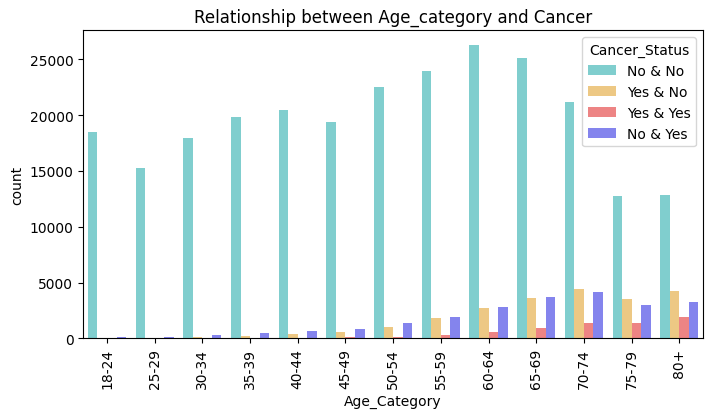
Looking at the age demographic we see age distribution is quite even with a slight skew to mid-old age se figur 8.

Figur 7



Figur 8

By looking at the relationship between certain health conditions and age groups, we can quickly see that certain ages tend to acquire different conditions. Depression (see figure 10) is a condition that seems to say at a certain frequency through out the age groups but falls off in the older ages. On the other side the other conditions such as Cancer, Cardio vascular and Arthritis seems to increase with age, especially with Arthritis. See figure 9, 11, 12



Figur 9

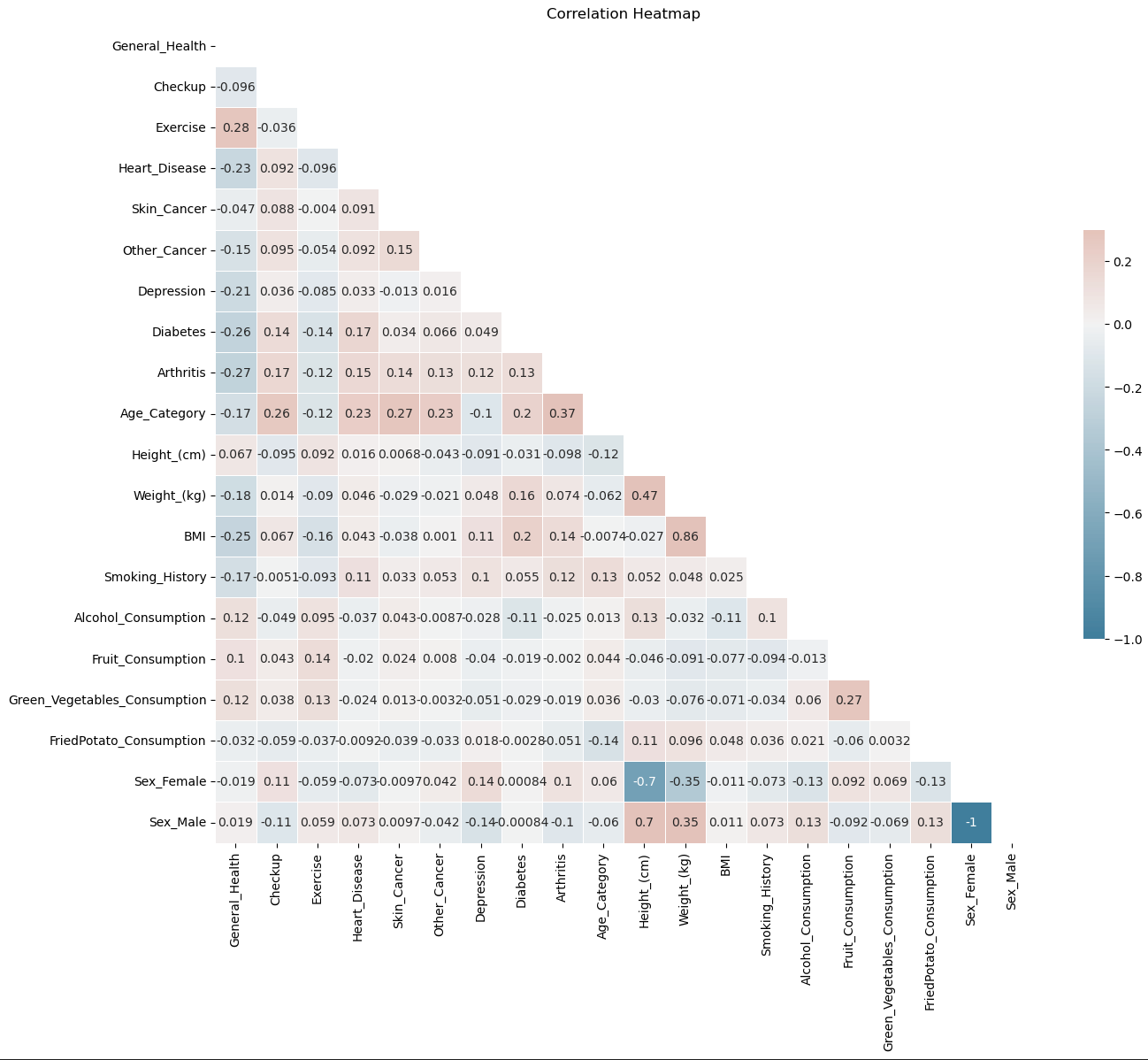
Figur 10

Figur 11

Figur 12

Last, we look at a correlation heatmap where we may see each correlation to each column. Notably as we just discovered the age category has a higher correlation to certain conditions such as Arthritis, Cancer, Cardio vascular disease and even Diabetes, there is also a high a noticeable correlation between age and checkups.

There are 2 noticeably high correlation, the first is height and females and the second is BMI and weight, both should not be surprising as females tend to be shorter than men and BMI is based on weight. See figure 13



Figur 13

## Prediction models

To predict heart disease we first need to consider what we want to predict. This typical problem is called classification where the target are classes such as types of animals “dogs”,”cats” or “horse. In our case we may classify the labels as a risk of heart disease or where p = probability of hearth disease ex. (0 <= 0.3 = low, 0.3 < 0.6 is medium and 0.6 <= 1.0 is high). Another way to classify is simply just to put a yes or no where p = (0 <= 0.5 = no and 0.5 < 1 = yes). Note that as the label when trained are only true or false (0 or 1) we decide to just continue with a yes or no output.

Furthermore multiple models will be trained and tested against each other to find the best performing model.

### Data imbalance

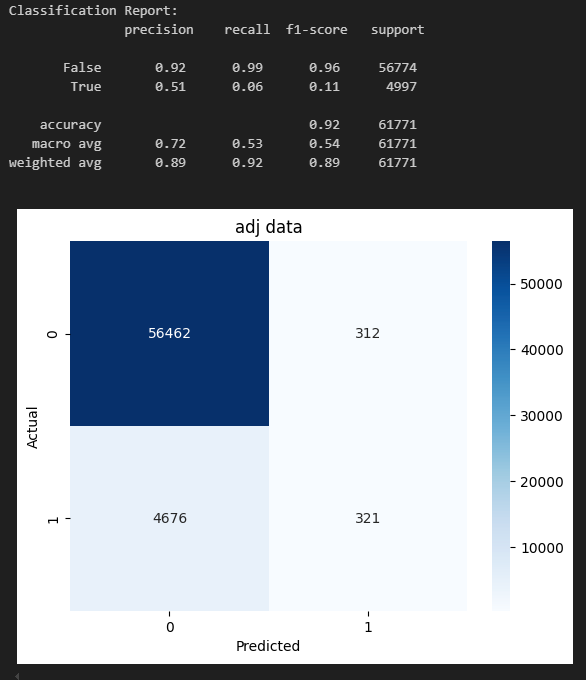
A major problem with the dataset is that most of our label data is unbalance meaning that there are a lot more individuals that does not have heart disease than those that do in our dataset, there is exactly 283883 individuals without heart disease and 24971 with heart disease.

This major imbalanced may cause an overfit for a lot of models as an imbalanced dataset tends to “over” train certain weights in models.

This can be solved by either manually trimming the dataset or adding more synthetic data.

We created 3 datasets.

* Dataset number 1 is named cvd\_adj which is simply the original dataset with a few dummy variable transformations.
* Dataset number 2 is named cvd\_cut which has been created simply buy reducing the amount of data where heart disease was False. There is no exactly 62427 entries with rougly 40% of the data having hearth disease
* Dataset number 3 is named cvd\_smotenc which was created by using syntethic data using on the original dataset using [imblearns](https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTENC.html) smotenc package. This dataset has 567766 entries and has a exactly 50% yes and no in heart disease

We created a simple Logistic regression model for each of the dataset and compared the test results.

On the first model train by the original cvd\_adj data we can se that the model was extremely overfitted to predict false on induviduals.

This is a major drawback as the model failed to identify 4676 patients with hearth disease. Taking a look at the f1-score we see that accuracy is quite high but the metric seems to hide the fact that we miss classified a lot of sick patients, hench forward we will look at macro avg which calculate the arithmetic mean of accuracy of each class instead of total accuracy.

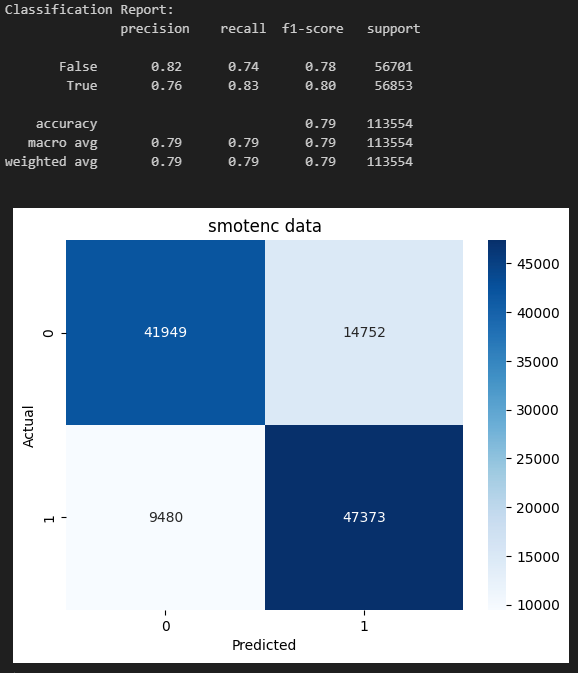
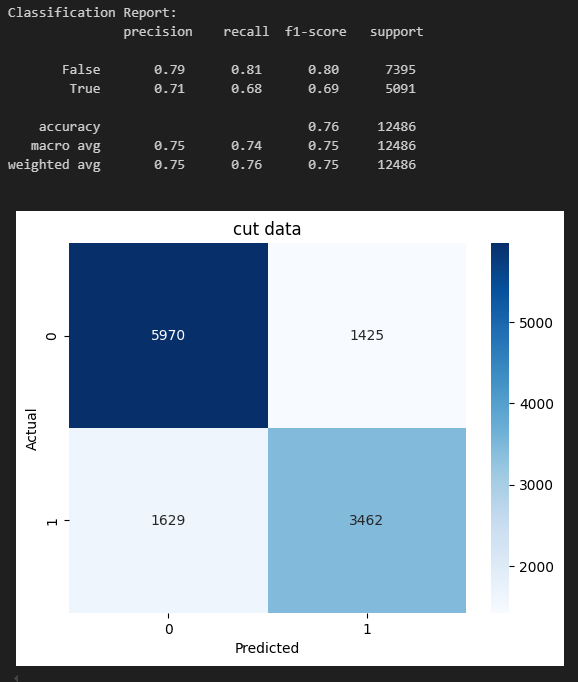
See figure 14

Figur 14

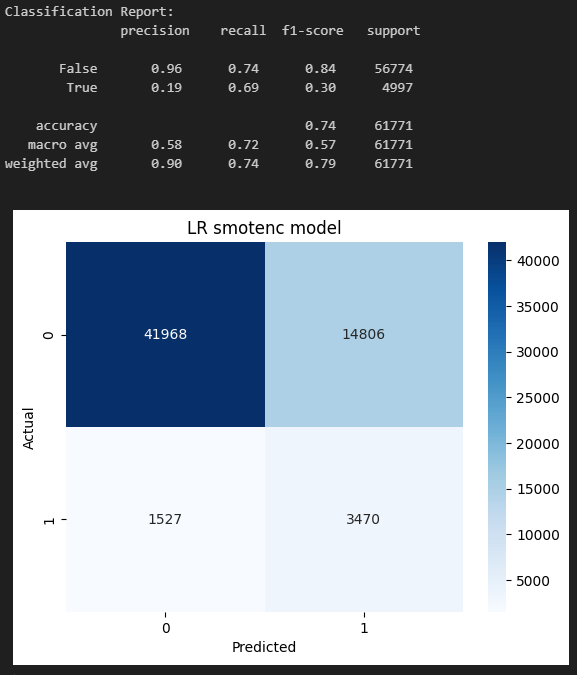
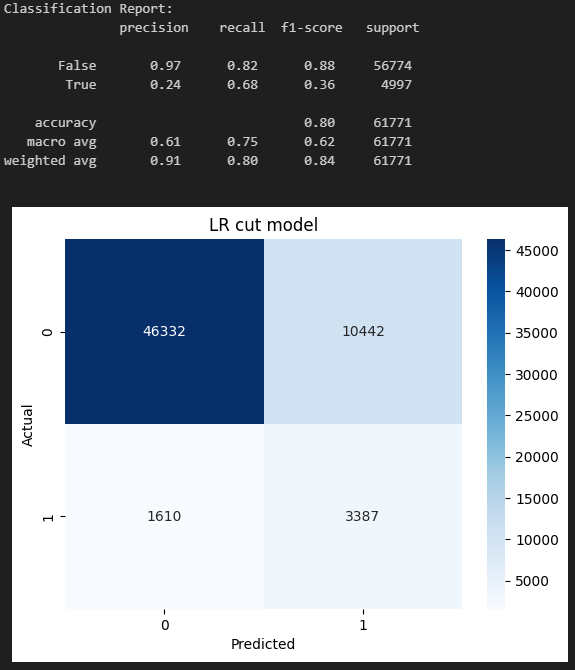
Looking the other results, we can see that both models successfully way more sick patients. see figure 15, 16

The performance increased with a slight edge with the synthetic dataset. We also tested the models on the original unbalanced dataset as within a real-life scenario the distribution of sick patients is also unbalanced, therefore a good model should also perform decent on unbalanced test data.

In figure 17, 18 we can see that the model trained on just the cvd\_cut data performs better on a real-life scenario, therefore we will continue to train the other models on that dataset as well.



Figur 15



Figur 16

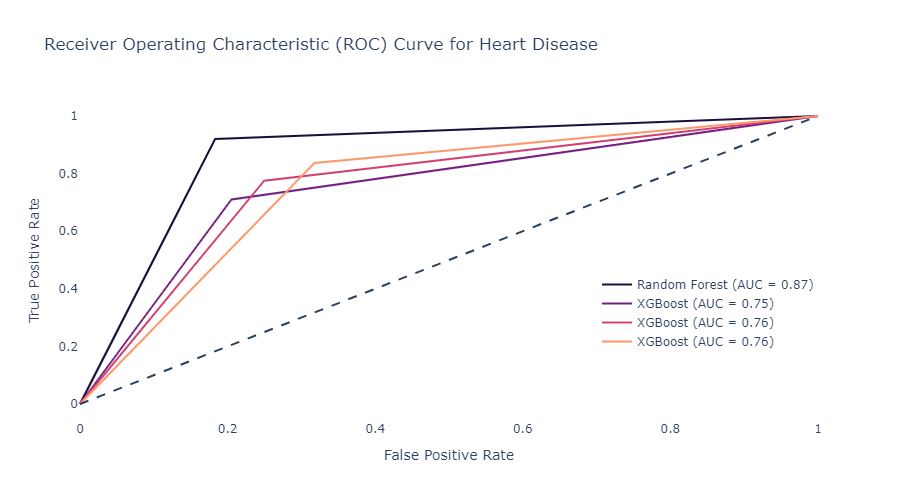
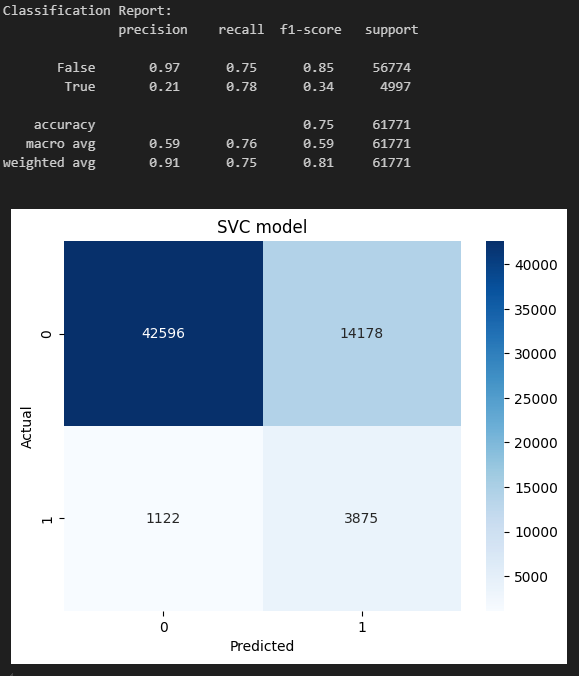
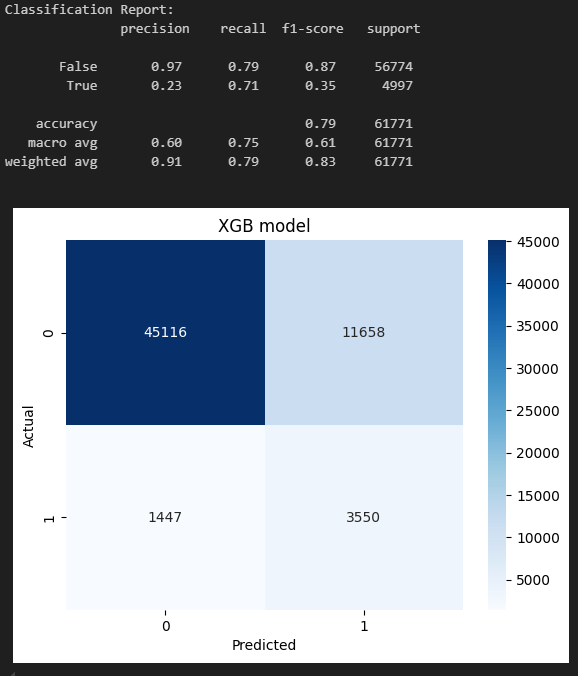
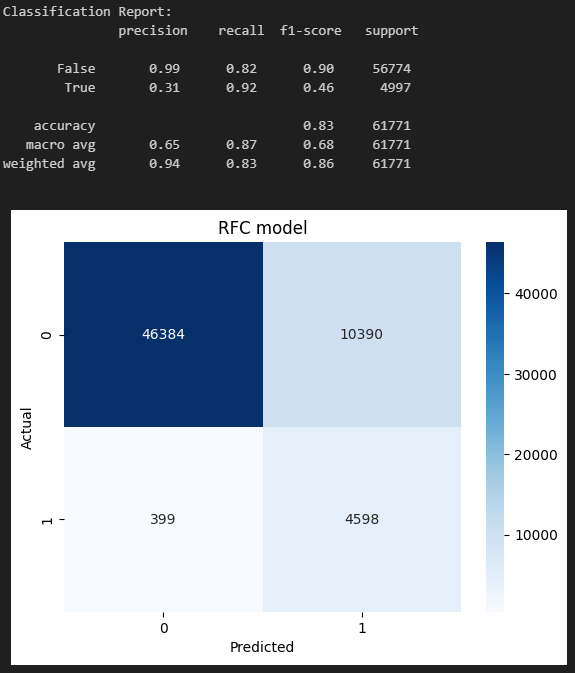
Figur 17

Figur 18

### Model comparison

In the model comparison we have chosen to compare 4 different models. Each model will be trained on the cvd\_cut dataset aswell as be tested on the original data to see how well the model would perform on real-life data. All models are using default hyperparameters from the [scikit-learn](https://scikit-learn.org/1.5/index.html) python package. The 4 chosen models are Logistic regression, Support vector machine, Random Forest Classifier and XGBoost.

From our evaluation we can conclude that the Random Forest model performs the best on all our evaluation metric thus we will continue and try and optimize the model. See figure (19, 20, 21, 22)



Figur 19

Figur 20

Figur 21

Figur 22

### Optimization and feature importance

By trying to optimize we might find hyper parameters that could increase our model accuracy. We will use method called GridSearchCV which will test multiple parameters for us.

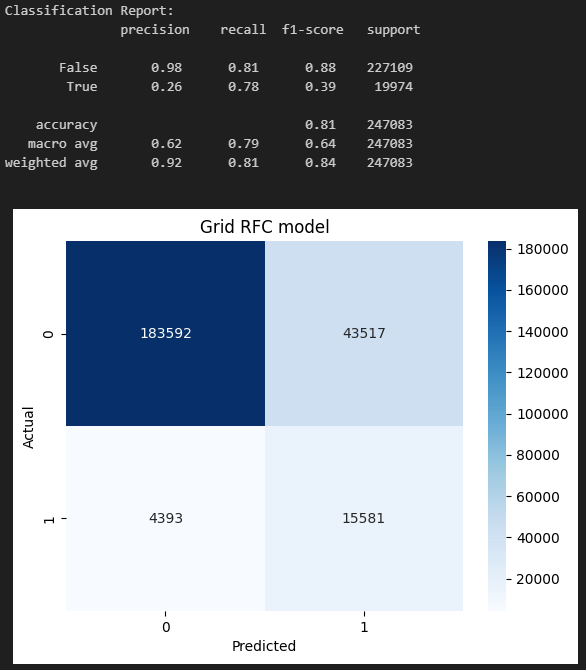
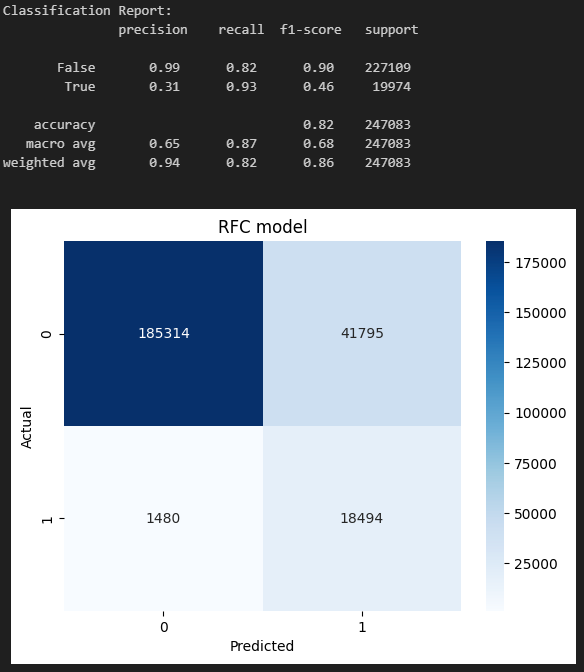
The parameters that were tested are:

* "n\_estimators": [100, 300]
* "max\_depth": [10, 20, None]
* "criterion":["gini", "entropy",]
* "min\_samples\_split": [2, 5]
* "min\_samples\_leaf": [1, 3]
* "max\_features": ["log2", "sqrt"]
* "bootstrap": [True, False]

These parameters will change how the decision trees in the forest will behave and build. Note that increasing parameters will Exponentially increase training time and computer power therefore a limit has to be set. In total the GridSearchCV created 4 folds with 192 candidates each totaling 768 fits.

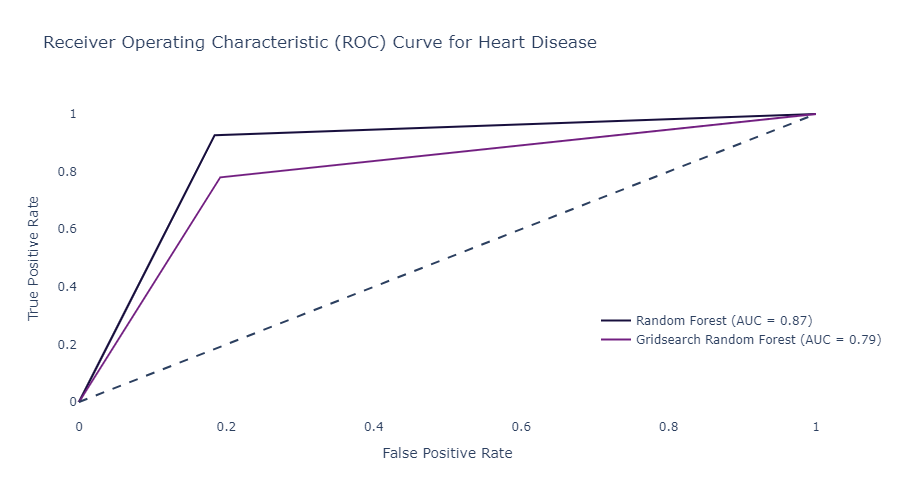
After completing GridSearch the best parameters are:

* 'bootstrap': True
* 'criterion': 'entropy'
* 'max\_depth': 20
* 'max\_features': 'log2'
* 'min\_samples\_leaf': 3
* 'min\_samples\_split': 2
* 'n\_estimators': 300,}

Despite having trying to optimize the parameter for a Random Forest classifier model the original RFC model seems to perform a bit better. This might be the cause of randomness or that the original default hyperparameters are already decently optimal. See figure (23, 24, 25)

Figur 23

Figur 24



Figur 25

## Clustering models

Clustering models are created by using unsupervised machine learning algorithm. This is were algoritms attempt to train weights on data without labels as targets, this is done by trying to find patterns in the dataset.

The dataset that will be used is the original also known as cvd\_adj.

To save time we will only test 2 different models HDBSCAN “Hierarchical Density-Based Spatial Clustering of Applications with Noise” And KMeans, the most common clustering algoritm.

The reason for choosing these too stem from this [research paper](https://link.springer.com/article/10.1007/s44196-024-00568-8) by Abeer Aljohani, Which mentions that in his research the best models for medical classification is kmeans, DBSCAN and Heirarkical.

We decided to combine DBSCAN AND Hierarchical and test HDBSCAN instead.

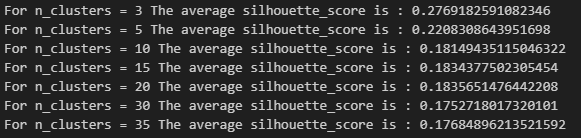
The models will also be trained on the whole dataset

### Model evaluation

A silhouette score was determined to be the way we compare models. While HDBSCAN will find the best *k* number of cluster Kmeans has to be manually set. HDBSCAN was firstly trained and gave a Silhuette score of 0.27 with 52 clusters. After training HDBSCAN we can train Kmeans and use the k=52 as a benchmark, we decided to train multiple kmeans models starting with k = [3,5,10,15,20,30,35] and if there were different results in Silhouette score we we might increase the range of k’s but looking at figure 26 there was a clear indication that the silhouette score was decreasing as k increased for the Kmeans model.

* To be exact the score for HDBSCAN was: 0.27489414220778463
* while the best score for Kmeans was: 0.2769182591082346.

But a decision was made to keep the HDBSCAN as it would be more interesting to analyse more than 3 clusters



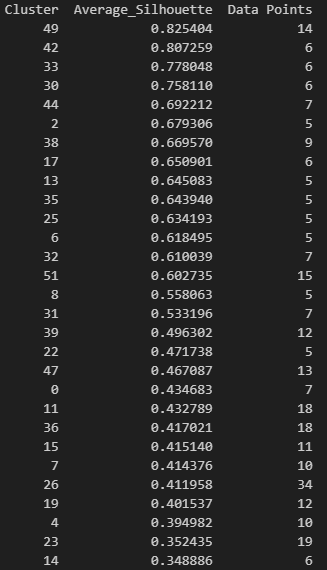
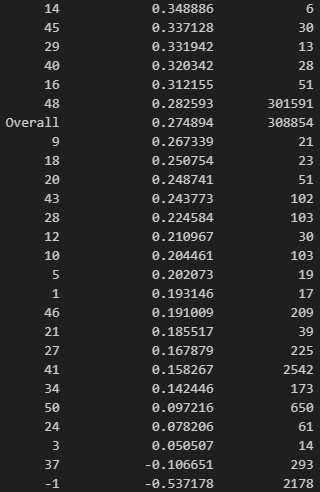
Figur 26

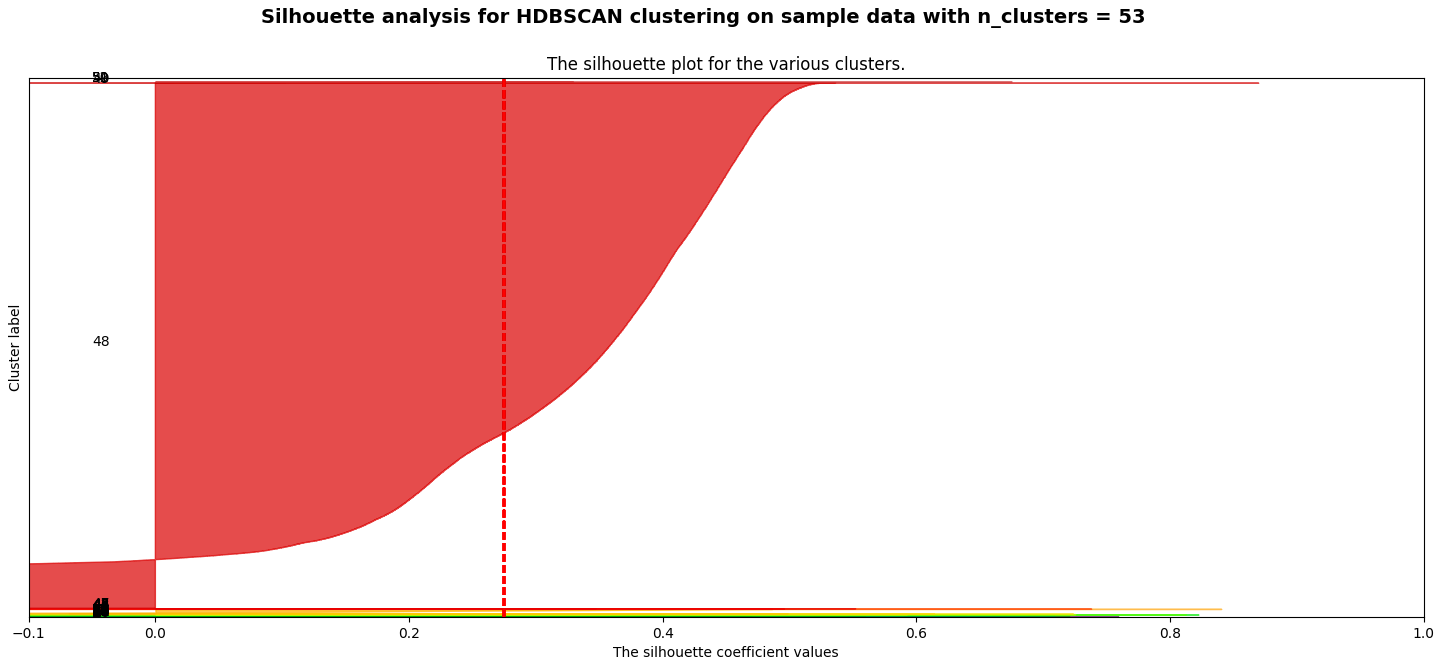
# Results and discussion

## Clustering analysis

Continuing with the HDBSCAN model we can look at a silhouette graph to get a visualization of the cluster distribution. In figure 28 we see a big cluster for cluster number 48. We can also investigate average silhouette score for each cluster in the model, in figure 27 we see that most cluster have a positive score indicating that most datapoints in these cluster do not reach over to other clusters, note that cluster number -1 are filled with datapoints that the model suggest are “noise” and that could not be assigned to a cluster.

Note that a lot of clusters have small amounts of datapoints especially in the high silhouette clusters.





Figur 27

Figur 28

We decided to only include analysis on the top 2 clusters as an analysis on a single cluster takes extensive amount of time. The clusters chosen are clusters with the highest silhouette score but also has more than 20 datapoints. By those criteria’s we end up with cluster number 26 and 45.

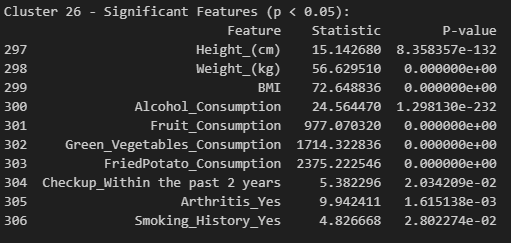
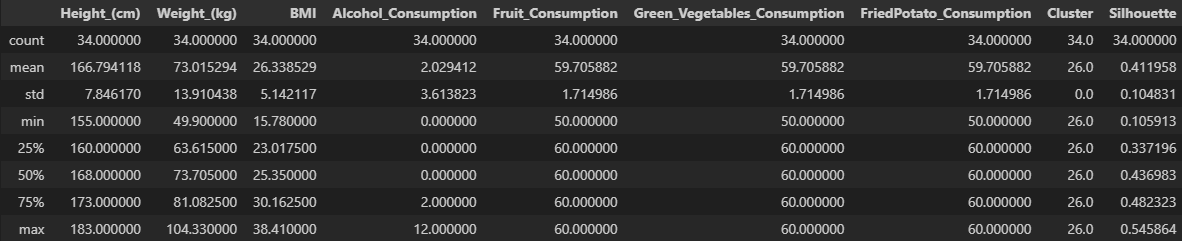
### Cluster number 26

We start by looking at a hypothesis test where we test if the there is a significant difference or relationship among the cluster being tested compared to other clusters.

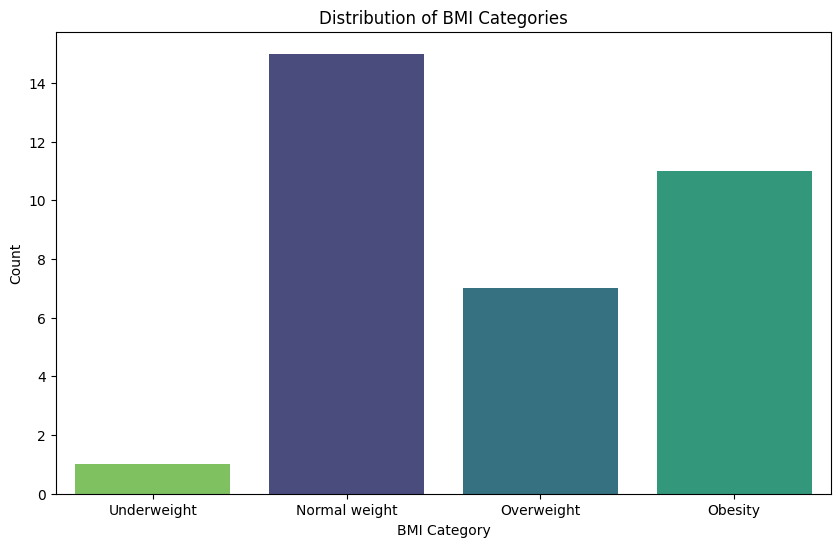
In figure 29 we can note that all p-values are extremely low for these features, note that the Statistic for Green\_vegetables, FriedPotato and Fruit Consumption features are extremely high this indicates that the mean variance for **each** cluster differ a lot compared to **other** clusters. And looking at figure 30 we can see that the STD “Standard deviation” in these features is quite low meaning most if not all individuals here consume high amounts of these foods, note that there is also a lack of alcohol consumption.

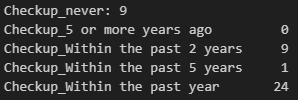
There is also a quite a high BMI here indicating that people might be overweight, looking at figure 32 we see that most people are normal to obese in this cluster.

Figur 29

There is also that argument that most of these people are heavy weight lifter or some other kind of elite athlete, and those individuals tend to have high BMI especially if they are shorter, looking at figure 30 again note that the mean heigh is 166 which is belove average weight and in figure 33 we see that most of these individuals tend to exercise and, on a side, note there is a low percent of smokers here as well.

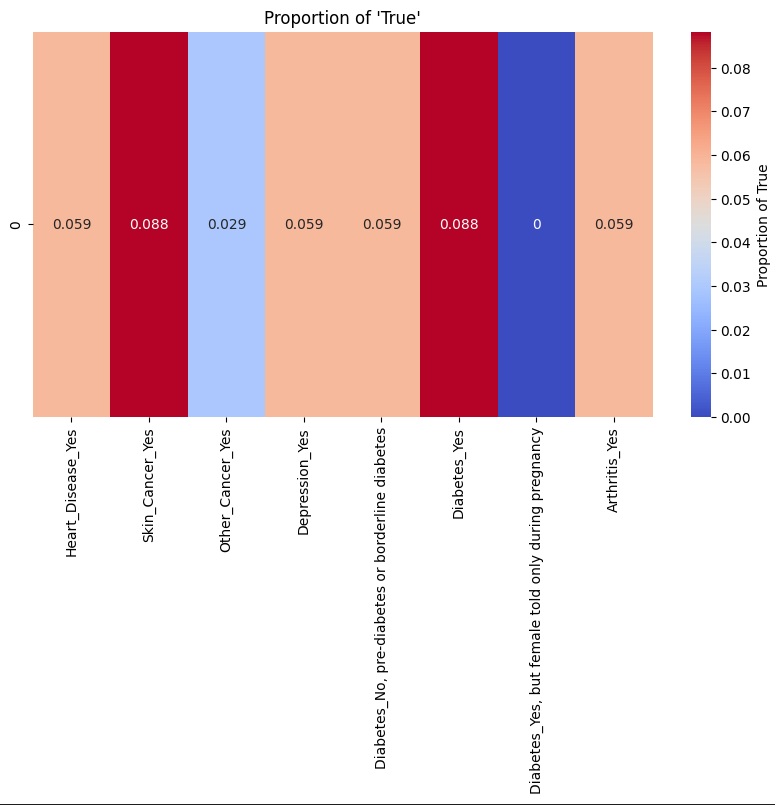
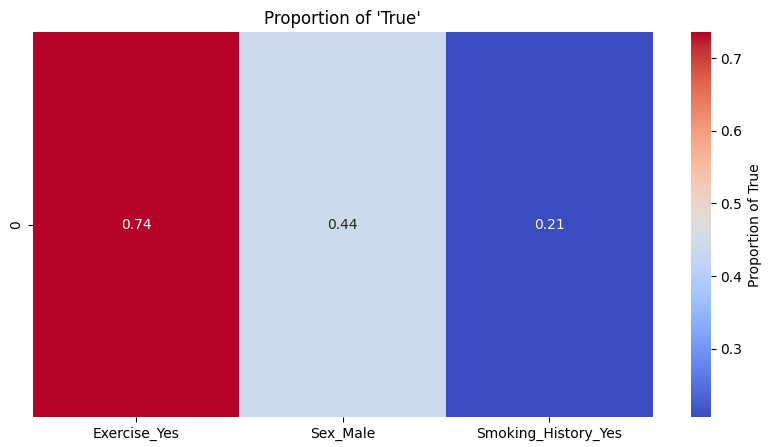
Figur 30



We can argue that in this cluster we have a special profile of individuals who consume a heavy diet but also exercise a lot. In Figure 31 there is also a majority of individuals who do regular checkups and in figure 34 we can see that there is low percentage of sick individuals. We can conclude that this lifestyle seems to keep patient healthy

Figur 31

Figur 32

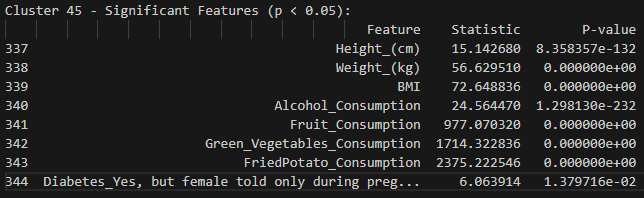


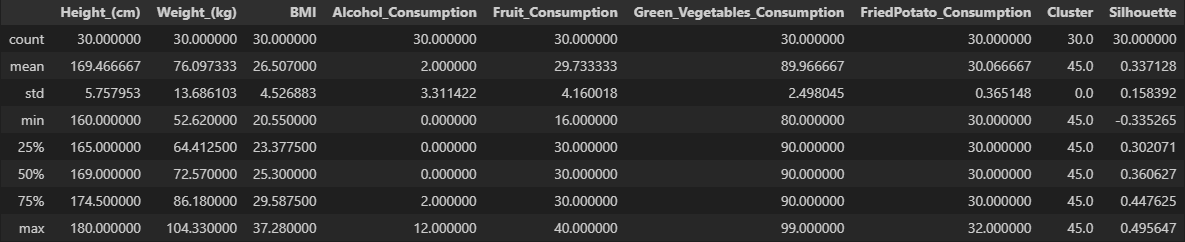
Figur 33

Figur 34

### Cluster number 45

In cluster number 45 we can se that we have fewer categorical significant clusters, the numerical features show the same number as they show the significance for the mean variance between all clusters, see figure 35. This doesn’t tell us as much so we continue to look at figure 36 we can see that we have another dietary habit that focuses less on fruits and Fried foods and have an increased intake of greens, note this group also have almost no alcohol intake and their BMI is also quite high.



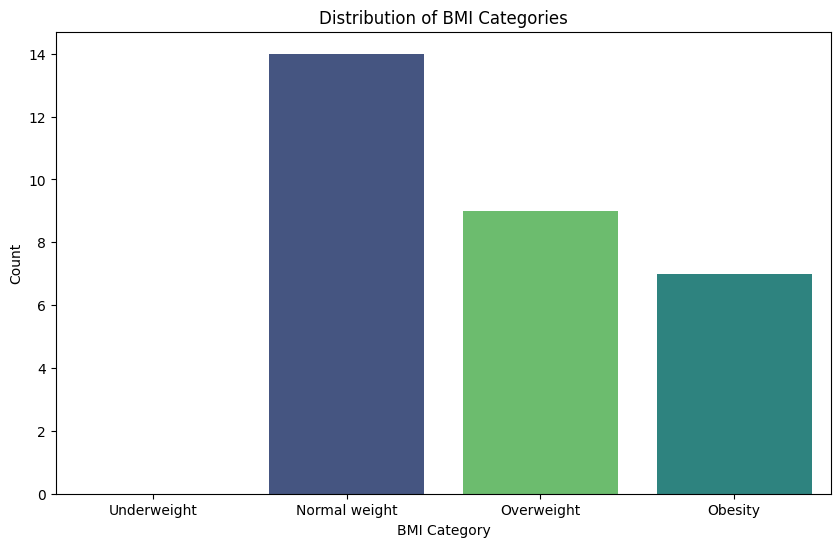


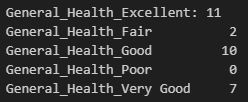
Figur 35

Figur 36

We can get further information by looking at figure 38 and there we can see that BMI is quite heavy but again looking at the exercise data we see that majority people here exercise (see figure 39) but there are also a lot more smokers, we may also note that most individuals here consider themselves quite healthy (see figure 37) and looking at the conditions we would say that they certainly are (se figure 40).

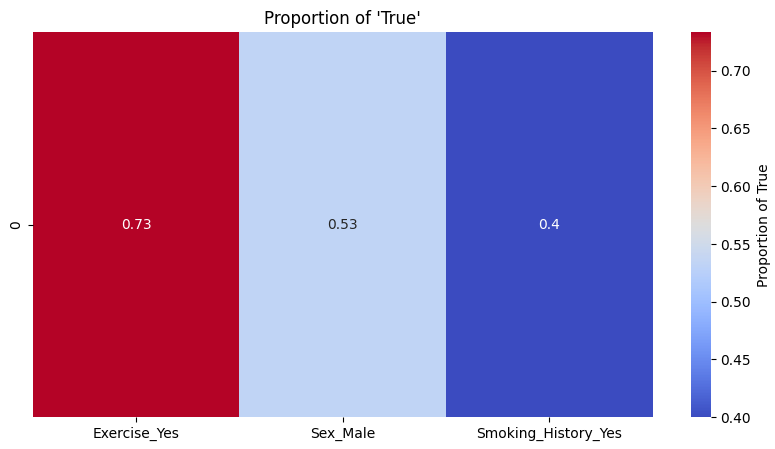
These too clusters look very similar and there might be a reason for that but there is also a big difference which mostly lay in the dietary habits of these people.



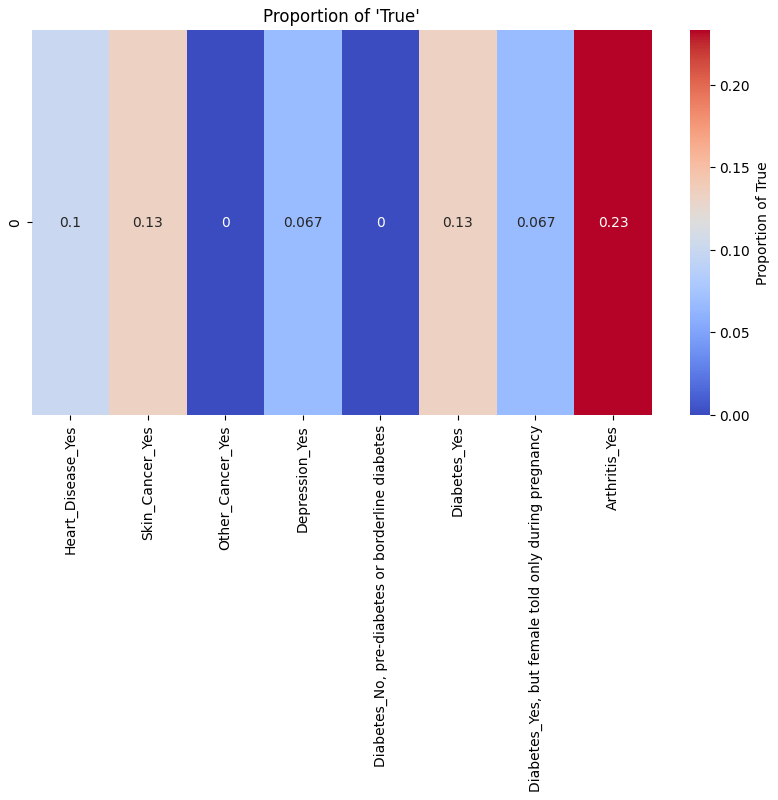


Figur 37

Figur 38



Figur 39



Figur 40

## Cluster comments

There are definitely more similarities between these two clusters than there are differences, but consider that there is quite a lot of clusters there will probably be very similar features. We might also just have data that may be large but not that disperse enough for the model to separate the clusters in a major way. In the ANOVA test we can definitely se that food consumptions tend to be a strong factor between the different clusters and such it lay way for the different eating habits. In real life a lot of people follow different diets almost religiously and all for good reasons staying way from fats and alcohol is usually correlated to being healthier and as we can see in these 2 clusters both these diets have had a healthy effect on these individuals.

With further analysis of more cluster, we might find something more useful but other than dietary habits, and a deeper analysis might uncover even more hidden patterns. But this shallow analysis has uncovered something that we definitely see in the real world which can be an argument that clustering medical data might uncover new beneficial information.

## Prediction model

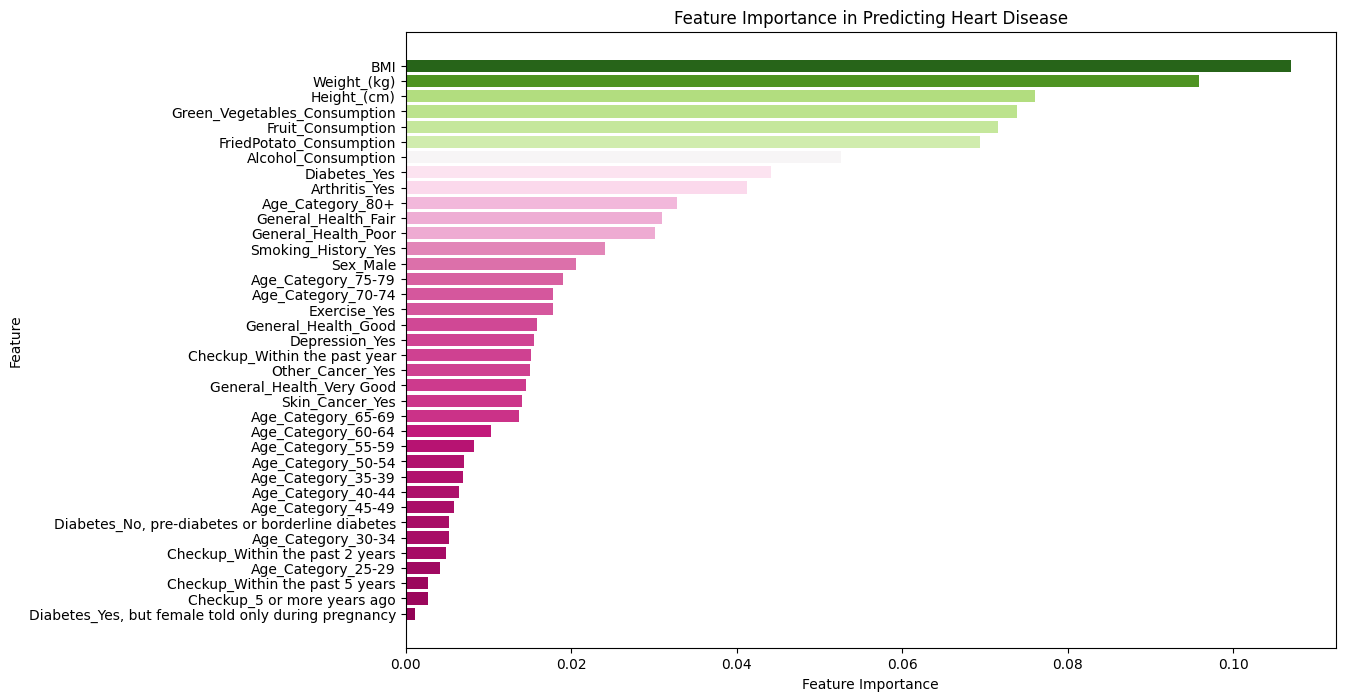
The Supervised model was in fact able to predict a medical condition and that will show potential in further medical research and applications. The difficult part is how we should handle False negatives meaning when a model predicts a healthy condition for someone who is actually sick, those instances might hurt the medical sector more than it helps.

With further research and with greater and better data even if we might not reach 100% accuracy, we may still develop highly accurate models that will help people detect underlying conditions that may have been hidden from the plain eye or even the expertise of a doctor.

In future attempts to predict conditions I would instead predict a risk instead of a yes or no, it would most likely give more information and remove uncertainty when getting a no, being classified with low or medium risk might contribute more than being classified with a yes or no.

There should definitely be taken with a huge grain of salt, a machine learning model will highly likely **not** replace professional and educated doctors but it might instead be used as a tool which can easy the workload and speed up the examination process.

We can also use models for more than prediction, a train model is also able to “rank” feature importance meaning that we can se how a feature is weight to that outcome which can help us draw conclusion to what may cause certain conditions. For example, look at figure 41 we can see that the highest contributing feature is BMI and food consumption (skipping height and weight as they correlate to BMI), this can be interpreted as people with high BMI have a higher probability for getting CVD, followed by food consumption greens would probably be interpreted as having low green consumption can lead to CVD etc.



Figur 41

# Conclussion

## Can you predict heart disease

It is fully possible to predict heart disease using a supervised machine learning model. It is also possible to predict other conditions but it will also depend on the data we have accessible and performance may differ a lot.

If you want to build a good prediction model I would first focus on finding reliable data and instead of predicting yes and no as we did, I would recommend to classify risk instead as that may be give not rule out the condition even if you get a no that could be a false negative.

## What supervised learning model predicts best

In our test Random forest classifier performed the best but the differences were not big enough to rule out other models.

## What model is used for unsupervised learning

In this test we decided to use HDBSCAN but other models might perform better. There is a possibility that other models may cluster medical data better, but it most likely will depend on the data you train on. I would also like to mention that Training a HDBSCAN model take a lot of time and computer power specially compared to KMeans, Kmeans may not be as complex as HDBSCAN but it could do as good as job. The HDBSCAN model took 40 min to train with the whole dataset using 12 threads

## How has the clusters been assigned to the data

Looking at the ANOVA test we can see that a lot of clusters were clustered through food consumption. This will differ depending on data and model. If clusters would instead cluster through conditions we could probably get more interesting cluster but we can’t force the model to cluster conditions as it doesn’t have labels. A solution might be to have as many clusters as conditions but even that might not guarantee the result we want.

In the end cluster algorithm doesn’t know what a medical condition is and will only cluster depending on density or Euclidian dastans. The only thing we can do Is analyze the clusters to find patterns that we be useful or not

## Has there been any benefits or any new information through clustering the data

We did find dietary habits in our cluster which on further analysis seems to had healthy individuals, this can be interpreted as certain diets contribute to less medical conditions, it should also be mention that those clusters had a high number to exercising individuals so it might be interpreted as different diets for elite trainer or athletes and the lack of medical conditions might not be from the food consumptions but instead of their exercising lifestyle.

There are many ways to look and analyze these patterns but some information was extracted from the clusters, useful or not useful.

# Sources

Dataset

<https://eajournals.org/ejcsit/vol11-issue-3-2023/integrated-machine-learning-model-for-comprehensive-heart-disease-risk-assessment-based-on-multi-dimensional-health-factors/>

General

Hands-On\_Machine\_Learning\_with\_Scikit-Learn-Keras-and-TensorFlow-2nd-Edition-Aurelien-Geron.pdf

Logistic regression

<https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148>

random forest classifier

<https://einsteinmed.edu/uploadedfiles/centers/ictr/new/intro-to-random-forest.pdf>

<https://medium.com/@mrmaster907/introduction-random-forest-classification-by-example-6983d95c7b91>

decision tree

<https://www.ibm.com/topics/decision-trees>

support vector machine

<https://medium.com/analytics-vidhya/a-beginners-introduction-to-svm-c641c3ff2769>

xgboost

<https://www.nvidia.com/en-us/glossary/xgboost/>

<https://developer.nvidia.com/blog/gradient-boosting-decision-trees-xgboost-cuda/>

Confusion matrix

<https://www.kdnuggets.com/2023/01/micro-macro-weighted-averages-f1-score-clearly-explained.html>

ROC And AUC

<https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>

HDBSCAN

<https://pberba.github.io/stats/2020/01/17/hdbscan/>

<https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html>

Hypothesis test

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2_contingency.html>

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.f_oneway.html>