Heart Failure Prediction Project

# Introduction

The purpose of this project is to explore and analyze heart failure data to develop and evaluate machine learning models capable of delivering accurate predictions.

The first section of the project will focus on data preparation and exploratory data analysis (EDA), starting with a concise explanation of the dataset and its variables. The second section will involve fitting three different machine learning models — Logistic Regression, SVM, Random Forest, and Decision Tree Boosting — and optimizing them to identify variants capable of delivering the most accurate predictions.

Finally, all models will be compared to determine which type performs best, not only in terms of prediction accuracy but also in interpretability and overall stability.

# Exploratory Data Analysis (EDA)

## Data description

The dataset analyzed in this project focuses on heart failure, specifically on determining whether a patient has heart disease based on various demographic characteristics and biomarkers. The dataset, originally titled *Heart Failure Prediction*, was sourced from *Kaggle*[[1]](#footnote-1).

According to *World Health Organization*, the cardiovascular diseases (CVDs) are the leading cause of death globally, representing 32% of all global deaths in 2019 – nearly 85% of those were due to a heart attack, or a stroke. There are many different variants of cardiovascular diseases, however nearly all of them are manageable if detected early. There are many proven behavioral (i.e.: unhealthy diet) and environmental (i.e.: air pollution) risk factors, which effects influence biomarkers, such as blood pressure. Those immediate risk factors can be measured in healthcare facilities, and thus it is possible to predict whether certain individual is in danger of developing a heart disease[[2]](#footnote-2).

There have been many attempts to model risks of developing a heart disease, with the notable one being the sex-specified algorithm - *Framingham Risk Score*. Its’ basic form is used to estimate the 10-year risk of developing a CVDs[[3]](#footnote-3).

## Data structure

This dataset was created by combining different datasets already available independently but not combined before – thus resulting in 918 observations of one outcome column (heart disease) and 11 predictors. To achieve a didactic purpose, the **original dataset has been modified to slightly imbalance the data** – the positive class was randomly undersampled.

The variables included in the dataset are as follows:

* *HeartDisease* [binary] – outcome column,
* *Age* [years],
* *Sex* [M/F],
* *RestingBP* [mm/Hg] – resting blood pressure,
  + High values suggest hypertension, a significant risk factor for heart disease.
* *Cholesterol* [mm/dl] - serum cholesterol,
  + Elevated cholesterol, particularly LDL ("bad" cholesterol), is associated with atherosclerosis and heart disease.
* *MaxHR* [beats/min] - maximum heart rate achieved,
  + A lower-than-expected maximum heart rate may indicate impaired cardiac function.
* *FastingBS* [binary, if > 120 mg/dl: 1] - fasting blood sugar,
  + A fasting blood sugar level above 120 mg/dl is indicative of hyperglycemia, a marker of diabetes or insulin resistance, which increases cardiovascular risk.
* *ChestPainType* [TA, ATA, NAP, ASY] - chest pain type,
  + TA (Typical Angina): Chest pain triggered by exertion, relieved by rest,
  + ATA (Atypical Angina): Less predictable or associated with other factors,
  + NAP (Non-Anginal Pain): Chest pain unrelated to heart disease,
  + ASY (Asymptomatic): No chest pain, common in silent ischemia[[4]](#footnote-4).
* *ExerciseAngina* [Y/N] - exercise-induced angina,
* *RestingECG* [Normal, ST, LVH] - resting electrocardiogram results,
  + Normal: No abnormalities detected,
  + ST: Indicates ST-T wave abnormalities, a sign of ischemia or previous heart attack,
  + LVH: Suggests left ventricular hypertrophy, a thickened heart muscle wall often due to chronic hypertension.
* *Oldpeak* - ST depression on the ECG during stress tests,
  + Indicates the severity of ischemia; higher values suggest worse outcomes.
* *ST\_Slope* [Up, Flat, Down] - the slope of the peak exercise ST segment.
  + Flat and Down indicates ischemia.

## Data analysis

The exploratory data analysis (EDA) began with a review of **summary statistics (Table 1)** to gain an initial understanding of the dataset's overall structure and detect any potential anomalies or inconsistencies. This provided valuable insights into central tendencies, variability, and possible irregularities within both numerical and categorical features.

Table Summary Statistics for numeric variables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Age | RestingBP | Cholesterol | MaxHR | Oldpeak |
| **count** | 522.000000 | 522.000000 | 522.000000 | 522.000000 | 522.000000 |
| **mean** | 51.660920 | 131.197318 | 216.691571 | 143.622605 | 0.591762 |
| **std** | 9.497245 | 17.357076 | 88.244897 | 25.364167 | 0.899450 |
| **min** | 28.000000 | 80.000000 | 0.000000 | 63.000000 | -1.500000 |
| **25%** | 44.000000 | 120.000000 | 195.000000 | 126.250000 | 0.000000 |
| **50%** | 52.000000 | 130.000000 | 226.000000 | 145.000000 | 0.000000 |
| **75%** | 58.000000 | 140.000000 | 264.000000 | 162.000000 | 1.000000 |
| **max** | 76.000000 | 200.000000 | 564.000000 | 202.000000 | 5.000000 |

**A graph of a box plot

Description automatically generated** **For numerical variables, distributions were assessed to identify outliers, and appropriate corrections were applied when necessary to mitigate their impact.** This ensured the data remained representative and minimized distortions that could affect model performance. In the end, **only *cholesterol* was adjusted**, as observations at 0 must be a numerical mistake, and values above 500 are highly improbable. Observations at 0 were replaced using KNN Imputation later during data preparation process – for now, they are treated as NAN values), and observations over 500 were omitted, thus reducing the number of overall observations.

Figure Box plot of Cholesterol in training set after adjustments.

Additionally, to explore variable relationships, correlation analyses were conducted. **Linear relationships were assessed through *Pearsons’ correlation (Table 2)*, while *Spearman’s rho (Table 3)* was used to detect monotonic relationship between numerical features and the response variable.** This dual approach helped capture associations both among exogenous variables, and between exogenous and endogenous variables.

Table Correlations among predictors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Age | RestingBP | Cholesterol | MaxHR | Oldpeak |
| **Age** | 1.000000 | 0.235697 | 0.132970 | -0.391675 | 0.249109 |
| **RestingBP** | 0.235697 | 1.000000 | 0.125848 | -0.109799 | 0.190216 |
| **Cholesterol** | 0.132970 | 0.125848 | 1.000000 | -0.041538 | 0.112503 |
| **MaxHR** | -0.391675 | -0.109799 | -0.041538 | 1.000000 | -0.123762 |
| **Oldpeak** | 0.249109 | 0.190216 | 0.112503 | -0.123762 | 1.000000 |

**All correlations presented above seem to fall within an acceptable range, ensuring that the collinearity does not occur.** As for *Spearman’s rho* (Table 3), *Cholesterol* and *RestingBP* appear to have low correlation, however it may not imply them to be not significant in a machine learning model – it will be checked during model creation.

|  |  |
| --- | --- |
|  | Spearman’s rho |
| **Age** | 0.238759 |
| **RestingBP** | 0.098001 |
| **Cholesterol** | 0.061974 |
| **MaxHR** | -0.319722 |
| **Oldpeak** | 0.346148 |

Table Correlations between numerical predictors and the response variable

**For categorical variables, *Cramer’s V (Table 4)* was employed to measure the strength of associations with the response variable**, highlighting potential dependencies and valuable predictive features. Additionally, the proportions of each class were examined to uncover any rare or underrepresented categories (assumed <1%) that could introduce bias or instability in the modeling. Most of these variables seem to have some predictive power, but a possible exception is *RestingECG*.

Table Associations between categorical predictors and the response variable

|  |  |
| --- | --- |
|  | Cramer’s V |
| **Sex** | 0.241612 |
| **ChestPainType** | 0.442648 |
| **FastingBS** | 0.261378 |
| **RestingECG** | 0.056554 |
| **ExerciseAngina** | 0.479162 |
| **ST\_Slope** | 0.517812 |

# Methodology

## Overview

The data has been split into a training and testing set using a 70/30 ratio, where 70% of the data is allocated to training and 30% to testing. As previously highlighted during the *Exploratory Data Analysis (EDA)* phase, cholesterol values recorded as 0 were initially treated as missing (NAN). **For the modeling phase, these missing values have been inputted using *K-Nearest Neighbors (KNN) imputation*.** This process was conducted separately for each dataset, with the KNN imputer trained solely on the training set to prevent data leakage.

## Chosen models

After splitting, the summary statistics of the training and testing sets were compared to ensure they are not significantly different, reducing the risk of sampling bias. Fortunately, no major discrepancies were observed. However, the class distribution within the sets reveals a class imbalance around 22% (for the positive class). This imbalance may introduce challenges during model evaluation and performance assessment. To address this, we will employ *Diagnostic Odds Ratio (DOR)* for cross-validation during threshold tuning.

The following models will be tested, as outlined in the introduction:

1. **Logistic Regression** – A simple yet effective linear model used for binary classification problems. It estimates the probability of a given input belonging to the positive class by applying the logistic (sigmoid) function to a linear combination of the input features.
2. **Random Forest** – An ensemble learning method that constructs multiple decision trees during training. The model outputs the class that is the majority prediction of the individual trees, enhancing accuracy and reducing the risk of overfitting.
3. **Decision Tree Boosting** – We will use ADA Boost (Adaptive Boosting) for decision tree boosting. ADA Boost builds decision trees sequentially, with each tree correcting errors made by the previous one. This iterative process focuses on misclassified instances to improve overall model performance.
4. **Support Vector Machine (SVM)** – A powerful algorithm that identifies a hyperplane in a high-dimensional space to separate classes. SVMs aim to maximize the margin between the closest data points (support vectors) and the hyperplane, leading to robust classification results, even for non-linearly separable data.

## Chosen approaches

**Initially, we will test these models on the basic dataset without applying weighting, undersampling, or oversampling.** This will provide a baseline performance metric to compare against more sophisticated methods. Observation weighting involves assigning different weights to individual instances during model training, allowing the model to give more importance to underrepresented classes, effectively compensating for imbalance.

**Next, we will present the results after applying observation weighting to mitigate the class imbalance. Following this, undersampling techniques will be introduced using *NearMiss*.** *NearMiss* is a family of undersampling methods that reduce the size of the majority class by selecting samples based on their distance to minority class instances. It has three main variants:

* *NearMiss-1* – Selects samples of the majority class for which average distances to the k **closest** instances of the minority class is smallest.
* *NearMiss-2* – It selects samples of the majority class for which average distances to the k **farthest** instances of the minority class is smallest.
* *NearMiss-3* – A hybrid approach that selects a given number of majority class instances for each minority class instance, balancing proximity and distribution.

**For oversampling, we will utilize *SMOTENC* (Synthetic Minority Over-sampling Technique for Nominal and Continuous variables).** *SMOTENC* is an extension of the *SMOTE* algorithm designed to handle datasets containing both categorical and continuous features. It generates synthetic samples by interpolating between existing minority class instances, preserving the distribution of continuous features while intelligently managing categorical variables to avoid introducing unrealistic data points.

By sequentially evaluating the models under these different conditions, we aim to identify the most effective strategy for addressing class imbalance and improving model performance.

**Additionally, we will perform threshold tuning to further address class imbalance and improve model sensitivity.** Threshold tuning works by adjusting the decision threshold at which a model classifies an instance as positive or negative, effectively shifting the balance between sensitivity (true positive rate) and specificity (true negative rate) – here we use *Diagnostic Odds Ratio Scorer*. By manually lowering the required threshold below 0.5, we increase the likelihood of predicting the positive class, which helps counteract the underrepresentation of positive cases in the dataset. This is especially important when higher sensitivity is desired, as in this case. The cross-validation process (cv=10) ensures the robustness of the selected threshold by evaluating performance across multiple folds, reducing the risk of overfitting.

## Tuned hyperparameters

Hyperparameter tuning is a crucial step in optimizing the performance of machine learning models by identifying the most effective combination of model parameters. This process involves systematically searching for the best configuration to enhance accuracy, reduce overfitting, and ensure generalization to new data. For the models we are using – *Logistic Regression*, *Support Vector Machine (SVM)*, *Random Forest*, and *Decision Tree Boosting (ADA Boost)* – hyperparameter tuning plays a key role in maximizing performance.

**Tuning of *Decision Tree Boosting* was carried out using the *grid search* approach**, which systematically evaluates combinations of these hyperparameters using 10-fold cross-validation (cv=10), optimizing for the average precision score to identify the best-performing model configuration. **The tuning process of *SVM and Random Forest* uses *randomized search* with 50 iterations to explore a wide range of hyperparameter combinations efficiently.** This approach speeds up the search compared to exhaustive grid search while still ensuring robust performance evaluation through 10-fold cross-validation (cv=10).

**Logistic Regression, unlike other models, does not have traditional hyperparameters** that require extensive tuning. The model relies on the assumption of a linear relationship between the features and the log-odds of the target variable. This simplicity makes logistic regression an attractive option for baseline models and situations where interpretability and efficiency are paramount.

**For Decision Tree Boosting we adjust three hyperparameters:**

* The *n\_estimators* parameter which determines the number of boosting rounds (iterations). Higher values generally lead to better performance but can increase the risk of overfitting if the model becomes too complex.
* The *learning\_rate* which controls the contribution of each tree, effectively scaling the weight updates. Lower learning rates often require more boosting iterations but can result in smoother and more generalizable models.
* The estimator parameter which specifies the base learner used in boosting, which in this case is a *DecisionTreeClassifier*. The max\_depth of 1 creates shallow trees (decision stumps), and the criterion ('gini' or 'entropy') defines how the decision trees are split.

**For SVM we adjust for various hyperparameters, depending on chosen kernel (Table 5):**

Table Hyperparameters of SVM

|  |  |  |
| --- | --- | --- |
| Kernel Type | Hyperparameter | Description |
| Linear | C | Regularization strength controlling the trade-off between margin size and misclassification. |
| Polynomial | C | - |
| Polynomial | Degree | Degree of the polynomial kernel. Higher degrees allow for more complex decision boundaries but can lead to overfitting. |
| Polynomial | Coef0 | Controls the influence of high-degree versus low-degree terms. |
| RBF | C | - |
| RBF | Gamma | Determines the influence of each training example. Lower values create broader decision boundaries; higher values create more complex ones. |

**For Random Forest we adjust four hyperparameters:**

* The *n\_estimators* parameter which defines the number of trees in the forest. A larger number of trees typically improves performance but increases computational cost.
* The criterion which determines the function used to measure the quality of splits, with gini focusing on minimizing impurity and entropy emphasizing information gain.
* The *max\_features* parameter which controls the number of features to consider when looking for the best split, acting as a regularization mechanism to prevent overfitting.
* The *max\_depth* parameter restricts the maximum depth of the trees, ensuring the model does not grow excessively complex, which can lead to overfitting.

# Results

To evaluate the performance of our models, *Diagnostic Odds Ratio (DOR)* has been employedas the primary metric. The DOR is a single indicator of test performance that combines sensitivity and specificity into one value. It is calculated as:

This metric reflects the odds of a positive result being correctly diagnosed relative to the odds of an incorrect positive result. **A higher DOR indicates better model performance, as it signifies a strong distinction between the positive and negative classes.** DOR is particularly useful when dealing with imbalanced datasets, as it provides a balanced perspective on model performance without being overly influenced by class prevalence.

**The results for training (Table 6) and testing (Table 7) datasets are presented below:**

Table DOR for training dataset (the highest value is marked in red)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Regular | Weighted | Undersampled | Oversampled |
| **Logistic Regression** | 18.685714 | 16.05 | 28.171429 | 16.7375 |
| **Random Forest** | 80.452381 | 39.487179 | 20.264706 | 46.764706 |
| **Boosting** | 13.082707 | 39.487179 | 7.5625 | 17.28 |
| **SVM** | 23.221154 | 18.631579 | 18.208333 | 41.166667 |

Table DOR for testing dataset (the highest value is marked in red)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Regular | Weighted | Undersampled | Oversampled |
| **Logistic Regression** | 45.241935 | 36.417778 | 42.47619 | 36.186667 |
| **Random Forest** | 149.390244 | 128.472222 | 124.586667 | 113.775 |
| **Boosting** | 36.698413 | 22.511905 | 31.952381 | 28.952381 |
| **SVM** | 101.454545 | 96.0 | 60.330827 | 33.066667 |

According to DOR, the best model would be a *Regular (without weighting or under-/oversampling) Random Forest*, for which the metrics are as follow:

* **Accuracy:** 0.89,
* **Sensitivity:** 0.91,
* **Specificity:** 0.89,
* **Precision:** 0.69.

Compared to, for example, Logistic Regression, Random Forest has no easily interpretable parameters, and thus various tools need to be employed to understand the model’s results. Firstly, an *MDI (Mean Decrease in Impurity – Figure 2)* method has been applied, where y-axis shows the mean decrease in impurity, which indicates how much each feature contributes to reducing impurity (or increasing homogeneity) during model training. Here, *ST\_Slope* seems to contribute most to decreasing impurity, however, the larger bar and wider confidence interval suggest variability in the importance of this feature across different trees. Values below zero suggest adding heterogeneity, or simply, noise rather than A graph with blue and black lines

Description automatically generatedpredictive power.

Figure MDI plot for Regular Random Forest

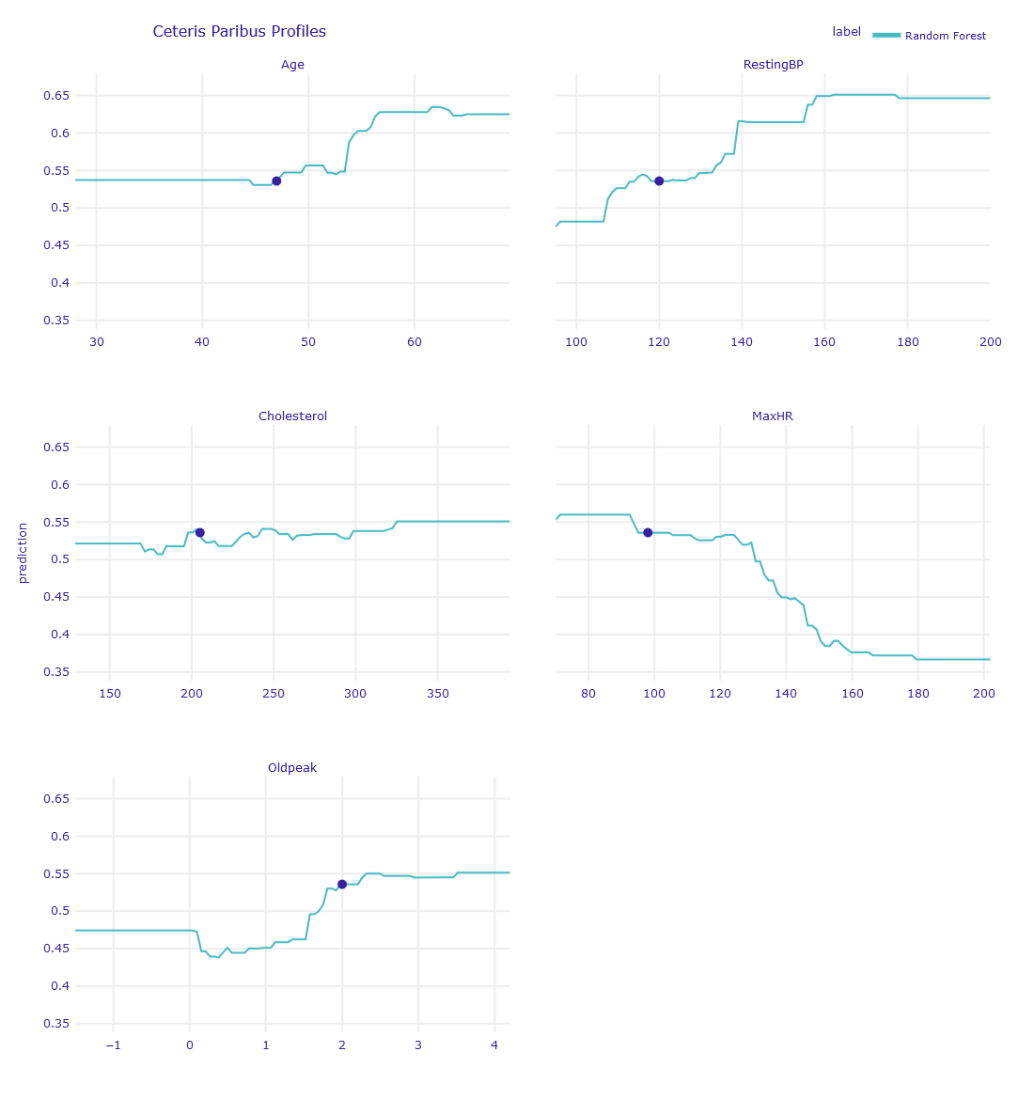
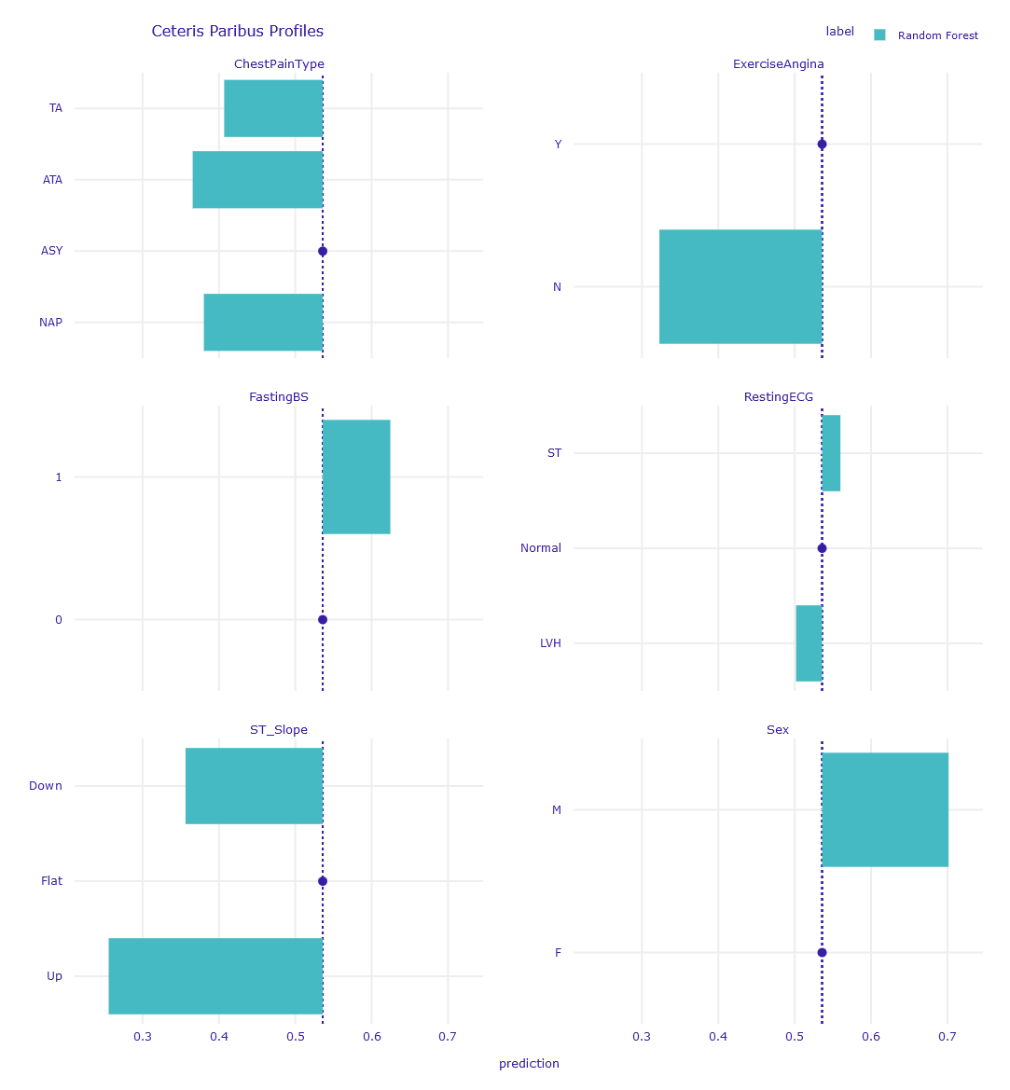
Later, *Ceteris-Paribus (CP) Profiles* have been created with a random “average” observation used as a default. Ceteris-Paribus (CP) Profiles plot how a model's predictions change as a single feature varies, while keeping all other features constant (hence "ceteris paribus" – Latin for "all else equal"). It can be noticed that, for example, higher heart rate decreases the probability of having a heart disease **for the chosen patient** (Figure 3).

Figure CP Profiles for continuous variables

Figure CP Profiles for categorical variables

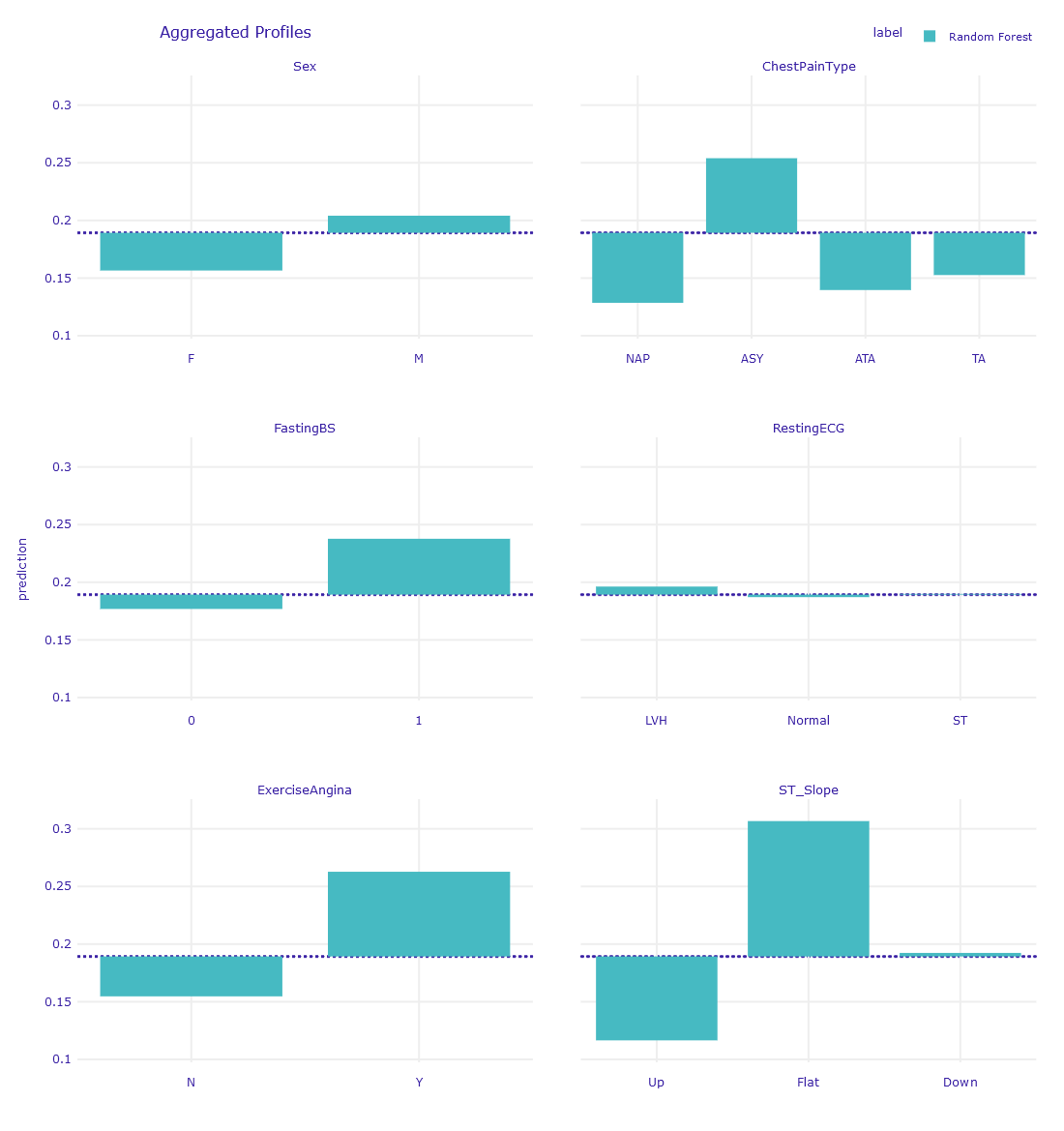
Besides CP Profiles, *Partial Dependence (PD) Profiles* have also been created, which show the average effect of one or more features on a model’s predictions by marginalizing them over all other features. To explain it easily, we use the average of a set of individual CP profiles – because we have a bigger number of observations, only average line is shown on the figures. It can be noticed that, for example, higher heart rate decreases the probability of having a heart disease **on average** (Figure 5).

Figure PD Profiles for categorical variables

Figure PD Profiles for continuous variables

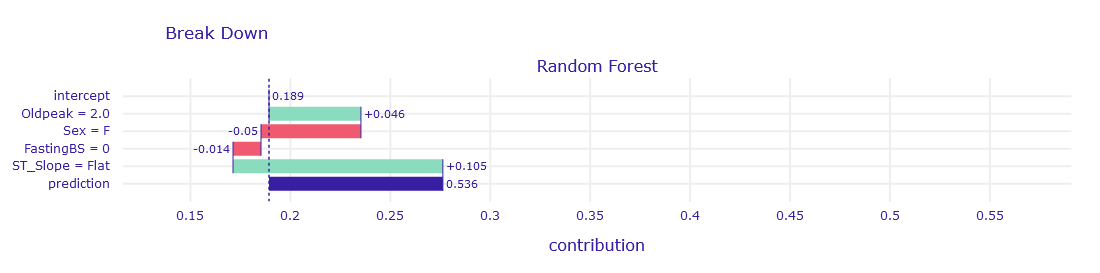
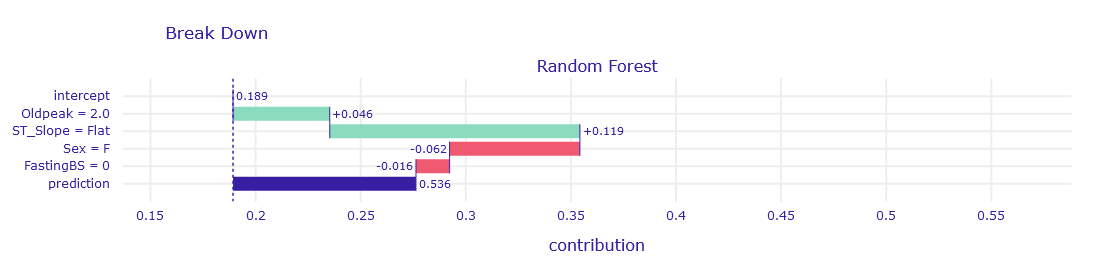
Another useful tool that has been employed is a *Break-down Plot*, which is a visual tool used to explain individual model predictions by decomposing them into contributions from each feature. **They show how much each feature pushes the prediction higher or lower relative to the average prediction (or baseline).** Break-down plots are similar to *CP Profiles* in a way that they are interpreted locally – we assume some basic value of a feature and look how it influences prediction. Additionally, it is worth noting that the order in which features are presented influences the interpretation of results, as each feature’s impact is calculated step-by-step.

Figure Break-down plot (Order 2)

Figure Break-down plot (Order 1)

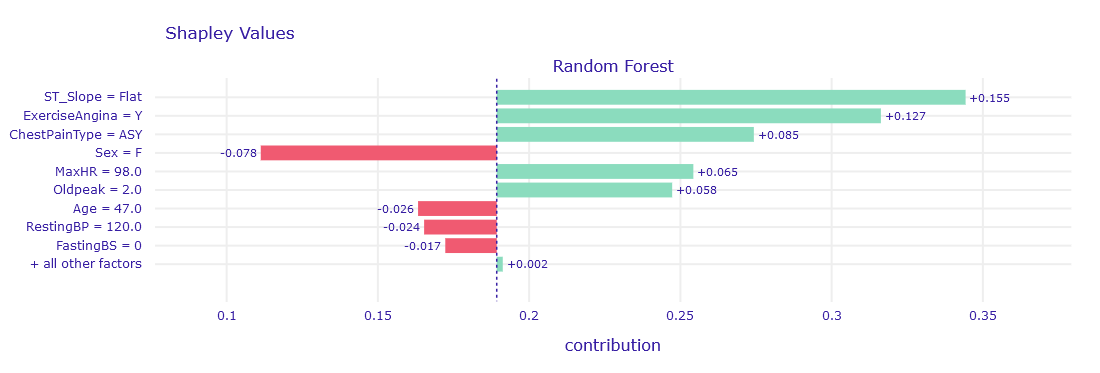
*Shapley Values* are a natural consequence of *BD plots*, as their one significant problem is the sequential approach and different results based on ordering. To remove the influence of the ordering of the variables, a mean value of the attributions can be calculated – such things are called Shapley Values. **Thus, we can notice that on average *ST\_Slope (Flat)* contributes the most to increasing the prediction value, and *Sex (Female)* contributes the most to decreasing the prediction value.** Such observation is in line with our previous findings.

Figure Shapley Values

# Conclusions

**After thorough evaluation, we selected *Regular Random Forest* as the final model** due to its superior *Diagnostic Odds Ratio (DOR)* on both the training and testing sets. This result indicated that the model performed well without requiring correction for class imbalance through weighting, undersampling, or oversampling. The data suggested that allowing the model to naturally handle the imbalance yielded better predictive performance.

**A key factor in the Random Forest’s success was the feature *ST\_Slope***, which represents the slope of the ST segment in an electrocardiogram (ECG). This feature is clinically significant as it reflects ischemic changes in the heart, serving as a crucial indicator of cardiac conditions. Analysis through techniques like *Shapley Values* revealed that *ST\_Slope* had a strong influence on predictions and played a pivotal role in reducing variability across individual trees, as evidenced by its high *Mean Decrease in Impurity* (MDI).

***Sex* also emerged as a contributing factor to the model’s predictions**, highlighted by Shapley values. However, its impact on reducing impurity (*MDI*) was not as pronounced as *ST\_Slope*. This suggests that while sex influenced individual predictions, it was not as critical in overall model stability and performance as *ST\_Slope*.

**In conclusion, the Random Forest model demonstrated that addressing class imbalance through complex resampling strategies was unnecessary.** Instead, the model effectively leveraged key features like *ST\_Slope* and *Sex* to drive accurate and reliable predictions, making it the optimal choice for this task.

1. *Heart Failure Prediction Dataset*, Kaggle, <https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction?select=heart.csv> [06.12.2024] [↑](#footnote-ref-1)
2. *Cardiovascular Diseases (CVDs)*, WHO, <https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)> [06.12.2024] [↑](#footnote-ref-2)
3. *Framingham Risk Score*, Wikipedia, <https://en.wikipedia.org/wiki/Framingham_Risk_Score> [06.12.2024] [↑](#footnote-ref-3)
4. Ischemia here refers to cardiac ischemia, which is an insufficient blood flow to the heart muscle, often leading to chest pain (angina) or a heart attack. [↑](#footnote-ref-4)