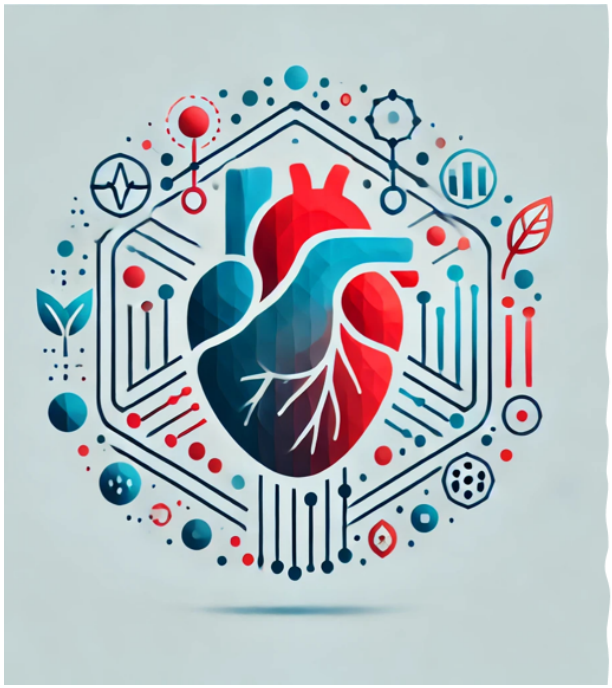

ChromAlyze

Your solution to understand your risk of developing coronary heart disease (CHD)

Project Documentation

Team Members:

- Joëlle Höchle
- Leonie Isele
- Ramon Winkler

Table of Contents

1 Project Overview

1.1 Project Description

ChromAlyze is a web-based application, allowing users to input their health metrics to calculate the risk of coronary heart disease. The program will return a personalized report including recommendations for health and lifestyle improvements. It is possible to access it from any device with a browser, like a laptop, tablet or smartphone. The Application is accessible from everywhere (when running on a server).

The projects aim is to target citizens, with the goal to enable them to understand their individual risk for coronary heart disease. Our solution intends to prevent disease by raising awareness in users to improve their health and lifestyle. Prevention is a key factor in reducing the long-term rising health care costs and deaths related to coronary heart disease.

1.2 Background

Coronary heart disease (CHD), also called ischemic heart disease or coronary artery disease is the global leading cause of death (WHO 2021). CHD is caused by fatty deposits in the coronary arteries, which supply the heart muscle with oxygen. The accumulating fat deposits ultimately block the blood flow, resulting in muscle damage. CHD is a multifactorial disease, including exogenous and endogenous factors like smoking, high blood pressure, high cholesterol and obesity.

2 Data

2.1 Data source

The Framingham Heart Study (FHS) is a long-term prospective study, which started in 1948, with 5209 men and women. Over the time of three generations, the study grew significantly, resulting in combining data from nearly 15'000 participants. The study focuses on different aspects of cardiovascular disease.

A subset of the Framingham dataset, from the 1960s, was used to pretrain our machine learning model. The data set has 4240 entries.

- **male**: gender of the participant (binary feature)
  - 0: female
  - 1: male
- **age**: age or the participant (continuous feature)
- **education**: categorical feature with different levels of education:
  - 1: Some high school (0 to 11 years)
  - 2: high school/GED
  - 3: some college/ vocational school
  - 4: college (BS, BA) degree or more
- **currentSmoker**: smoker (binary feature)
- **cigsPerDay**: number of cigarettes smoked per day (continuous feature)
- **BPMeds**: Use of anti-hypertensive medication at the examination process (binary feature)
  - 0: no anti-hypertensive medication at the examination process
  - 1: anti-hypertensive medication at the examination process
- **prevalentStroke**: (binary feature)
  - 0: free of disease
  - 1: prevalent disease
- **prevalentHyp**: (binary feature)
  - 0: free of disease
  - 1: prevalent disease
- **diabetes**: (binary feature)
  - 0: not a diabetic
  - 1: diabetic
- **totChol**: serum total cholesterol in mg/dL (continuous feature)
- **sysBP**: systolic blood pressure in mmHg (continuous feature)
- **diaBP**: diastolic blood pressure in mmHg (continuous feature)
- **BMI**: weight in kilograms/height meters squared (continuous feature)

- **heartrate**: in beats per minute (continuous feature)
- **glucose**: casual serum glucose in mg/dL (continous feature)
- **TenYearCHD**: The 10-year risk of coronary heart disease (binary target)

2.2 Data preprocessing

The goal of preprocessing is to have a complete and consistent data set. The first step is to address the missing values, which are imputed by sklearn's iterative imputer. The imputer predicts missing values based on the available data inputs. Second, the data was statistically analyzed. Focusing on the outliers, there are some alarmingly high values, but after evaluation they were determined to be realistic. Therefore, no outliers were removed from the dataset. In addition, the distribution of the data was taken into account by checking whether the features have a normal distribution.

After cleaning, the data is split into a training and a test data set using the "train_test_split" function from sklearn. To include all features equally, the "StandardScaler" is used. This function standardizes the features by removing the mean and scaling to unit variance.

To address the imbalance of the binary target (15.2% positive cases), different sampling techniques were used on the training data. The functions are from the imblearn library:

- Undersampling of the majority class
- Minority class oversampling
- Synthetic Minority Oversampling Technique - Smote

Each training set is stored in a separate data frame as preparation for the machine learning model.

2.3 Synthetic Data

Introducing synthetic data requires a very deep understanding of the relationships between the characteristics and the outcome. It is possible to create and enter new data based on the odds ratio or hazard ratio with some random imputation and probability with the baseline risk. In this example, only the association with the target is considered. As soon as other correlations are brought into the equation, it becomes more and more complicated.

The correlations largely reflect how well the data was interpreted and understood by the programmer. Due to the complexity and falsification of the real data set, we have not included the synthetic data in the application.

3 Machine Learning

3.1 Preparation

To keep the code lean, reduce repetitive code and make global changes easier, functions have been developed. We implemented a function “run_model()” allowing us to run machine learning models, including the decision logic for the sampling mode and the output of the models performance, in one line of code. The performance metrics of each model is automatically stored in a dataframe to simplify the model selection process.

The second function, called “plot_roc_curves()”, allows to plot all Receiver Operating Characteristics Curves (ROC) for each model combined in one plot.

To further contribute towards lean code we established a “model_optimizer()” function. This function is using RandomizedSearchCV to find a combination of hyperparameters, aiming to improve the individual base classifiers performance. The algorithm uses k-fold cross validation, where the evaluation of the parameters is calculated with different test sets. The input is a parameter grid, in which the algorithm searches for the best scoring combination.

The following classifier algorithms have been tested:

- Logistic Regression
- Naive Bayes (GaussianNB)
- Decision Tree
- Random Forest
- K-nearest neighbour (KNN)
- Extreme Gradient Boost
- Support Vector Machines
- Multilayer Perceptron

To keep the code lucid, the functions are executed in separate code cells. After using the standard parameters of the machine learning algorithm as a baseline, the “model_optimizer()” is executed to improve the performance.

3.2 Feature exclusion

The study originated in the United States. Education affects several aspects of health, including access to health insurance, where a person lives, and individual wealth. This factor and its effects cannot be easily applied to different countries. In addition, our analysis showed that the correlation with the risk of coronary heart disease was not high. Therefore, this characteristic was removed from the user input.

3.3 Model selection

With the previously created performance dataset and an analysis of the metrics, the most appropriate model for the research question can be determined. But which metric is the most relevant?

Due to the class imbalance of the target variable, relying on the accuracy of a model would have been misleading. To find the most appropriate model, a weighted score was applied to each model's performance metrics.

Since we are interested in early detection of coronary heart disease risk, it is particularly important to consider true positives. Therefore, recall was given the highest weight.

```
weights = {
   "Recall": 0.3,
   "F1 Score": 0.25,
   "ROC AUC": 0.2,
   "Precision": 0.15,
   "Accuracy": 0.1
}
```

Based on the weighted score, a logistic regression model was selected:

Table 1 Receiver Operating Characteristics (ROC)

Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC (%)	Weighted Score (%)
67	25	63	35	71	53

Coronary heart disease has a very complex set of influences on why someone develops it.

Because coronary heart disease is a multifactorial disease, the implemented traits are important risk factors, but there are many more, and research is discovering new risk factors all the time. This means that our dataset is still missing other potentially strongly correlated features. As a result, classification is possible, but when looking at the metrics, the performance is moderate.

4 Web application

We built our web application using the flask library in python. The logic for each endpoint was implemented in the app code, rendering the individual html file. Using different libraries, it starts with a library check to ensure that the whole code can be executed without errors. Because of safety issues it is also important to run it on a virtual environment.

At the start it loads the scaler and the previously prepared model. To improve traceability, a log file is created that stores all entries, including the machine learning algorithm, parameters, form entries, and the generated report.

In its current state, the application does not meet the legal requirements for public access. Therefore, it is accessible by running the application locally on a private computer.

The web site provides terms and conditions, a privacy policy and a contact form. All but the contact form is pro forma, they would consider intensive research regarding legal and ethical considerations. The information from the contact form is stored in a csv file, to which it would be possible to add some automatic email or task generation based on the content.

4.1 Risk prediction

The web application loads the pretrained model to predict a citizen's risk. By using a categorization function the risk is assigned to one of three categories of risk scores:

- Low risk [0…0.25]
- Elevated risk]0.25 … 0.6[
- High risk [0.6 … 1]

4.2 Personalized Health Report Generation

To create an individual health recommendation, a combination of manual reporting and LLM is used. The manual reporting introduces a split of the entry data into individual risk categories. Each feature follows its own logic. As an example, for BMI, the official classification in underweight, normal, overweight and obese has been used. Based on these classifications, individual sentences will be added to the report. To extract the key points of this text, an LLM was implemented; the open source LLM (zephyr-7b-beta). This process can be deactivated with a check box.

5 Conclusion

Version 1.0 of ChromAlyze was successfully implemented. We developed a web application for personalized health risk assessment using a machine learning model. The modular architecture allows the integration of additional features, new data or different algorithms, to enhance the accuracy of the prediction. Future versions will focus on user feedback and expanding the dataset to include a wider population and increase the scope of risk factors responsible for coronary heart disease.

6 Discussion

ChromAlyze was built on older data of the Framingham dataset, based in the 1960s. Its performance must be validated with new data, including data from other countries. This will be the crucial next step prior to a public launch. In this validation process, not only the performance, but also the accuracy of the categorization process must be reviewed with new data.

Launching an app classified as software as a medical device, will require to meet legal and regulatory requirements. These requirements have yet to be established.

7 References

Python libraries:

- Scikit – machine learning
- Imblearn – sampling methods
- Pandas – data frame management
- Flask - API
- Numpy – array calculations
- Markupsafe – safety of server
- Haystack – pipeline for LLM
- Waitress – lightweight WSGI Server

LLM:

- Hugging face, zephyr-7b-beta

Dataset:

- Framingham dataset from https://github.com/GauravPadawe/Framingham-Heart-Study/blob/master/framingham.csv
- Study conductor: https://www.framinghamheartstudy.org/

Knowledge Sources:

- Body mass index – BMI. World Health Organization. https://who-sandbox.squiz.cloud/en/health-topics/disease-prevention/nutrition/a-healthy-lifestyle/body-mass-index-bmi. Accessed 17 November 2024
- 07 August 2024. The top 10 causes of death. World Health Organization. https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death. Accessed 08 November 2024
- 10 January 2024. Coronary Artery Disease – Coronary Heart Disease. American Heart Association. 100 Years Bold Hearts. https://www.heart.org/en/health-topics/consumer-healthcare/what-is-cardiovascular-disease/coronary-artery-disease. Accessed 08 November 2024
- 20 December 2023. What Is Coronary Heart Disease. National Heart, Lung, and Blood Institute. https://www.nhlbi.nih.gov/health/coronary-heart-disease. Accessed 08 November 2024
- 17 January 2024. Coronary heart disease. NHS website for England. https://www.nhs.uk/conditions/coronary-heart-disease/. Accessed 08 November 2024
- Cardiovascular Disease (10-year risk). Framingham Heart Study. Three Generations of Dedication. https://www.framinghamheartstudy.org/fhs-risk-functions/cardiovascular-disease-10-year-risk/. Accessed 10 November 2024

Image Sources:

- ChromAlyze project symbol: created with OpenAI (2024), DALL-E (Version 3)

Usage of artificial intelligence: Artificial intelligence like ChatGPT, DALL-E, Claude, Elevenlabs was utilized to assist with multiple parts of the project. DALL-E was used for the project logo, Claude for code assistance and correction, DeepL correction of spelling, grammar and syntax. ChatGPT was utilized for explanation and research (with attention to potential AI errors and fact checking with other resources) and assistance with the general draft and structure of the project. Elevenlabs provided the AI for Text to Voice Transition.