

# **Project 2 – CHD**

**Author(s):**

**Gabriel Jackson**

**Naad Kundu**

**Thao Nguyen**

**Kayla Nguyen**

**Halbert Nguyen**

**Ben Willoughby**

**Omar Zeineddine**

**DS 3001: Foundations of Machine Learning**

**April 16, 2024**

## Summary

The main focus for this project was to build predictive algorithms to determine the likelihood of a person developing coronary heart disease. We utilized various demographic variables such as sex, age, education, current smoker status, cigarettes per day, etc. We first cleaned our data by updating any NaN values. We needed to clean several variables, including education, glucose, BPMeds, totChol, BMI, cigsPerDay, and heart rate, in terms of NaNs. We then saved these cleaned datasets into updated CSV files. After cleaning the data, we used Linear Regression, K-Nearest Neighbors Classifiers and Decision Tree Classifiers to predict the likelihood of developing coronary heart disease. To create these models, we used the provided data to build our training model and to test our model. For our linear models, we used the SKLearn library. Our linear regression model results were interesting, particularly due to our dummy education variable. We decided to compare two linear models: one with all variables and one excluding the education variable. We found that the adjusted  $R^2$  values for the model with all variables were higher than those for the model without the education variable, indicating that the model with all variables had more explanatory power for predicting CHD. Our decision tree model also yielded interesting results—it had an 85% accuracy rating. However, it did end up having a lot of false positives, incorrectly predicting CHD in 143 cases where the person did not have CHD. The importance of education in predicting CHD suggests potential socio-economic factors influencing heart health, warranting further investigation into how these predictors interact. For our last model, we used a K-Nearest Neighbors Classifier. The goal was to classify the data based on the distance to their neighbors. It would find a most common classification among their neighbors, and the model was run with anywhere between 1-150 neighbors. Our results showed that the ideal number of neighbors was 6, giving an accuracy of about 85%. Our next steps will focus on refining the models by integrating additional variables and employing more advanced machine

learning techniques to enhance predictive accuracy and reduce false positives. This continued refinement will aim to minimize overfitting while exploring the impact of unexamined variables.

### **Data**

In terms of the data, two CSV files were provided—one for training our model and one to test the model. The variable we are trying to predict is ‘10YearCHD’, which represents the 10-year risk of coronary heart disease (as a 0 or 1). Initially, we had to clean the training dataset. For our education variable, we replaced all NAs with 0s, in order to keep its value numerical. For our glucose variable, we filled all NAs with 85, which is the human average. This was done to keep the data from being skewed for our models. For our BPMeds variable, we replaced all NAs with 0 to indicate that they are not being taken. For our totChol (total cholesterol), BMI (body mass index), cigsPerDay, and heartRate variables, we decided to drop all the NaNs. We then saved these cleaned datasets into their respective updated CSV files to prepare them to be used by both our linear regression model and our decision tree. To begin our linear regression models, we first updated our education variable. One big challenge we encountered was with our education variable. The education variable was categorical, but it appeared numerical. We proceeded to change the entries into text, so that we could easily create dummy variables. We replaced its values from 0-5 to ‘Unknown education’, ‘Some high school’, ‘High school/GED’, ‘Some college’, ‘College’, respectively. Overall, our dataset contains multiple dummy variables. For many of the variables, the representation is intuitive, as 1 is true and 0 is false. But for our sex variable, 0 represents female and 1 represents a male. In our exploration, we proceeded to construct two linear models: one without an intercept using the variables sex, age, our education dummy variable, and currentSmoker, cigsPerDay, BPMeds, prevalentStroke, and prevalentHyp, and one without the education variable due to the speculation that it did not have as much impact on predicting risk than the others did. To make a

decision tree, we used the sklearn library for the decision trees, utilizing the same cleaned data files as we did for our linear models.

## **Results**

In order to predict the likelihood of a person developing coronary heart disease, three predictive models were considered using training and test data from the Framingham Heart Study. This data includes the following variables: sex, age, currently a smoker, cigarettes per day, use of blood pressure medication, stroke prevalence, prevalent hypertension, diabetic, total cholesterol, systolic and diastolic blood pressure, body mass index, heart rate, glucose, and the 10 year risk of coronary heart disease. Overall, the three predictive models used were K-Nearest Neighbors Classifier, Decision Trees Classifier, and linear regression.

The first predictive algorithm used was linear regression. The linear regression model used the training data to create an equation where each variable and its associated weight are used to make a prediction of whether or not a person is likely to develop coronary heart disease within the next 10 years. The code for the linear regression model can be found under 'linear\_regression.ipynb'. This code generated a table for each variable and their respective weights (under 'coefficients'), which can be seen in Figure 1.

	Variables	Coefficient
0	sex	0.064109
1	age	0.006864
2	College	-0.582501
3	High school/GED	-0.613281
4	Some college	-0.587104
5	Some high school	-0.585679
6	Unknown education	-0.625217
7	currentSmoker	-0.011914
8	cigsPerDay	0.002694
9	BPMeds	0.069935
10	prevalentStroke	0.129553
11	prevalentHyp	0.020255
12	diabetes	0.048580
13	totChol	0.000289
14	sysBP	0.002250
15	diaBP	-0.001345
16	BMI	-0.000788
17	heartRate	0.000179
18	glucose	0.001228

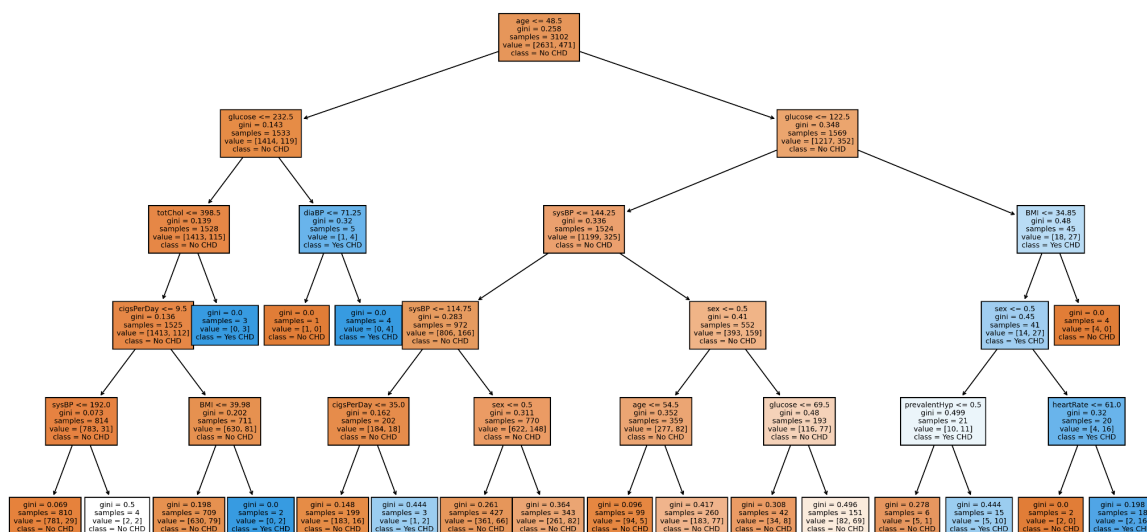
**Figure 1: The linear regression table of each variable and each variable's coefficient.**

As can be seen from Figure 1, there were some variables that impacted the risk of developing coronary heart disease the most: prevalence for stroke had a coefficient of roughly 13%, taking blood pressure medication had a coefficient of 6.9%, and being male had a coefficient of 6.4%. The higher magnitude coefficients show which of the variables have a greater impact in developing CHD in the next 10 years. Additionally, it seems that the education variables ('High School/GED', 'College', 'Some college', 'Some high school', and 'Unknown Education') all had a coefficient of roughly -0.6 with some variation among the coefficients. One theory as to why this happened could be how every person fits one of the categories for education which causes the model to shift the risk of any person by roughly -0.6%. This high coefficient is likely to reflect how each person is relatively safe against developing CHD before considering the other variables. The linear regression model had an adjusted  $R^2$  value of roughly 0.094, which normally means that the model does not fit the data well. One explanation for this poor  $R^2$  value is that the regression line can never fit the data perfectly when the dependent variable is binary and the

regressors are continuous. This means that the  $R^2$  value, which is typically used for measuring the explanatory power or ‘usefulness’ of this model, loses its interpretation. Furthermore, under linear probability models such as this, the regression line will not fit the data perfectly if the dependent variable is binary and the regressors are continuous.

The next predictive model used was K-Nearest Neighbor Classification. The code for this model can be found under the ‘knn.ipynb’ Jupyter Notebook file. The KNN model was used to classify the data based on their distance to a certain number of their neighbors. It will then classify data based on the most common classification among these neighbors. To determine the correct number of neighbors to use, the KNN model was run from 1 to 150 neighbors, where the ideal numbers of neighbors can be selected based on that model’s accuracy. From these iterations, the ideal number of neighbors was found to be 6, which gave the highest accuracy score of roughly 85%. The max-min normalized data found an optimized number of neighbors of 11, which also had an accuracy of about 85% due to the fact that the predictive outputs consisted entirely of 0’s.

The final predictive model used was Decision Tree Classification, which presented a series of decisions in the form of a tree of nodes that predict if a person is likely to develop coronary heart disease (‘Yes CHD’) or not (‘No CHD’). The code used to create this decision tree can be found under the ‘decision\_trees.ipynb’ Jupyter Notebook file. The training and test data were used to create a 5-layer Decision Tree Classifier from SKLearn, which can be seen in Figure 2.



## Figure 2: The Decision Classification Tree

The decision tree will make a decision at each node, which after making a series of decisions in the tree leads to a terminal node that will make a prediction on whether or not a person is likely to develop coronary heart disease. The terminal nodes that end in a 'Yes CHD' prediction are colored from a range of white to blue depending on how many samples were used to make this terminal decision. Additionally, the terminal nodes that end in a 'No CHD' prediction are colored from a range of white to orange, depending on how many samples were used to make this terminal decision.

Along with creating this decision tree diagram, the decision tree was tested with the testing data to produce a confusion matrix that determined accuracy, specificity, sensitivity, and MCC score of the tree. The confusion matrix showed 85% accuracy, 0.90% specificity, 98.76% sensitivity, and a MCC score of 10.68%. This means that with a specificity rate of 0.90% there was a high false positive rate and that with a sensitivity of 98.76% that it was rather successful at detecting actual coronary heart disease cases. Therefore, the low specificity and high sensitivity leads to the model favoring 'Yes CHD' strongly. This can be seen in its low false negative rate and its high false positive rate. Additionally, the MCC score of 10.68% showed that it was not good at balancing errors made within the tree. However, it did have an accuracy of roughly 85% in the testing data, which demonstrates its ability to correctly classify a person as likely to develop CHD. While this decision classification tree was correct 85% of the time, it can only account for up to 5 nodes (5 decisions in the tree). Using a higher number of layers is possible, but makes the diagram harder to interpret (Figure 2). Therefore, the researchers chose to use 5 layers to balance readability with performance. This comes at a cost, where roughly 5 decisions can be made to determine if a person is likely or not to develop coronary heart disease (despite there being more than 5 variables). While the tree does not consider all variables (at each decision node), it does indicate which variables are most likely to determine developing CHD, such as a higher age, glucose level, etc. It's also fairly

straight-forward to understand the effects of these variables as the conditions for each node are fairly interpretable.

While each model provides some amount of predictive ability for the test data, there are tradeoffs to each model. Classifying models such as KNN or decision trees produce definitive positive or negative predictions with a constraint on the number of layers or groups to prevent overfitting the data. In the context of health services, this is a powerful tool as the recommendations from classifying models can help doctors reach a CHD diagnosis and actionable plan to treat a patient. The main problem with implementing this sort of model in the real world is the consequences of a prediction being wrong, but since the inaccuracies we encountered were mostly false positives, misdiagnosing someone that will not develop coronary heart disease seems overly cautious rather than dangerous. Linear regression on the other hand was useful because it gave predictions in terms of probability or likelihood of developing coronary heart disease. These predictions are useful for identifying people who are most at risk, but not as actionable as the previous models. For example, how would you choose to dedicate a treatment to two patients who both have a 50% chance of developing coronary heart disease? Statistically, one patient has the opportunity of being successfully treated, but doctors would likely avoid leaving that decision up to chance. As previously mentioned, it's also hard to tune a linear regression model for a binary dependent variable because predicted values are probabilities while observations are positive or negative results. Overall, the predictions generated by classification models such as KNN and decision trees seem more actionable in the context of health data while linear regression predictions seem better at highlighting groups that are most at risk.

### **Conclusion**

Overall, our project embarked on an ambitious task to predict the likelihood of developing coronary heart disease (CHD) using a subset of data from the Framingham Heart Study. Through the



methods of linear regression models, decision trees, and KNN, we explored various factors that could increase risk factors in developing CHD, such as sex, age, education, smoking habits, and medical history. Our approach required us to focus on the fundamentals: meticulous data cleaning, thoughtful variable transformation, and strategic model selection in order to adhere with the project's objectives of experimenting with machine learning models while having fun.

One of the project's significant findings was the varied impact of education on the risk of CHD, revealed through comparing models with and without the education variable. This result highlights the interesting relationship between socio economic factors and health, suggesting that education level can serve as a precursor to activities that impact heart health. Additionally, the decision tree model further complemented our analysis with an 85% accuracy rate; however, with a good amount of false positives. This goes to show the complications of predicting binary values (yes/no) in data regarding health. It is important to note that while our decision tree model did generate a higher number of false positives than ideal, this is often a challenge in medical prediction models, particularly when prioritizing sensitivity. In future iterations, adjusting the decision threshold and exploring cost-sensitive learning could better balance the trade-off between sensitivity and specificity, reducing false positives without significantly compromising the ability to detect actual cases of CHD.

Our project revealed that while linear regression can model the trajectory of CHD development, limitations arise as depicted by the low  $R^2$  values. This limitation most likely stems from the binary nature of the dependent variable (risk of CHD), as compared to the continuous variables used, complicating model fitness and accuracy. Additionally to further improve the predictive accuracy of our decision tree model, we could consider implementing a more complex decision tree with more levels. This approach allows for a more in depth analysis of the variables and their interactions, reducing the amount of false positives and improving the overall model performance. Moreover, the complexity and abundance of features in our models made it challenging to effectively pinpoint which variables impact CHD risk. As a

result, these issues hindered our ability to draw definitive conclusions from the predictive models.

As for further exploration, we could incorporate other machine learning techniques such as clustering, in hopes of enhancing accuracy and reducing false positives. Exploring the relationship between other variables in the dataset can reveal other potentially unexpected predictors of CHD. Another way we can further elaborate on this data is to account the model for change in risk factors over time. This would provide us a different and dynamic point of view on CHD prediction. Moreover, doing further analysis to include more information such as genetic information, dietary habits, or even psychological markers can add depth to our models. These multidimensional analyses allow for an intricate platter of factors influencing CHD, presenting deeper insights and more detailed predictions.

In conclusion, although our project may be not perfect, it gives us a glimpse into the world of health data science. Going from the raw data to cleaned data to predictive models is complicated and tedious, but the experience gained from this project is what makes it possible for us to adopt a thirst for knowledge, hopefully understanding more of how we can decrease the risk factors of CHD. As technology progresses and machine learning models become even smarter, we are getting closer to a future where predictive models can provide personalized insights on improving health, increasing human lifespan.

## **Appendix**

No additional plots or tables were useful to include in this write-up, however the source code can be found within the '.ipynb' files in the project's repository's root at:

[https://github.com/thaonguyyen/project\\_chd/](https://github.com/thaonguyyen/project_chd/)