Flavien Moise, Tenzin Nargee, Charlie Payne, Duncan Tanner

04.23.2024

Professor Johnson

DS 3001

CHD Project Findings

Summary: A one-paragraph description of the question, methods, and results (about 350 words).

Our project builds predictive algorithms that analyze the likelihood of a person developing coronary heart disease (CHD) within ten years, measured by the variable TenYearCHD. We use three tools to make correlation predictions between 15 independent variables and TenYearCHD, including *k* nearest neighbor, linear regression, and decision trees. The variables analyzed include things such as sex, age, and education, as well as things like heart rate, systolic blood pressure measure, and BMI. The methods suggested that sysBP, BMI, and glucose were the best predictors of TenYearCHD overall. That being said, our results were not entirely conclusive. Issues arose with the linear regression analysis, for example, due to the nature of the variable types, which led us to approach the problem using a logistic regression analysis for a slightly better analysis. The results of our *k* nearest neighbor method also suggested limited predictive power for given variables due to the complex nature of the data set and their correlations.

Data:

As the chart below defines, we analyzed 15 key variables measured in 3,179 individuals for their predictive power in estimating the 10-year risk of coronary heart disease (listed last).

Variable Name	**Definition**
age	Age at the time of medical examination in years
BMI	Body Mass Index, weight (kg)/height (m)2
BPMeds	Use of Anti-hypertensive medication at the exam
cigsPerDay	Number of cigarettes smoked each day
currentSmoker	Current cigarette smoking at the time of examinations
diabetes	Diabetic according to criteria of the first exam treated
diaBP	Diastolic blood pressure (mmHg)
education	A categorical variable of the participants' education, with the levels: Some high school (1), high school/GED (2), some college/vocational school (3), college (4)
glucose	Blood glucose level (mg/dL)
heartRate	Heart rate (beats/minute)
prevalentHyp	Prevalent Hypertensive. The subject was defined as hypertensive if treated
prevalentStroke	Prevalent Stroke (0 = free of disease)
sex	The recorded sex of the observations with 1 denoting a participant coded as male
sysBP	Systolic Blood Pressure (mmHg)
totChol	Total cholesterol (mg/dL)
TenYearCHD	The 10-year risk of coronary heart disease (CHD), the target variable

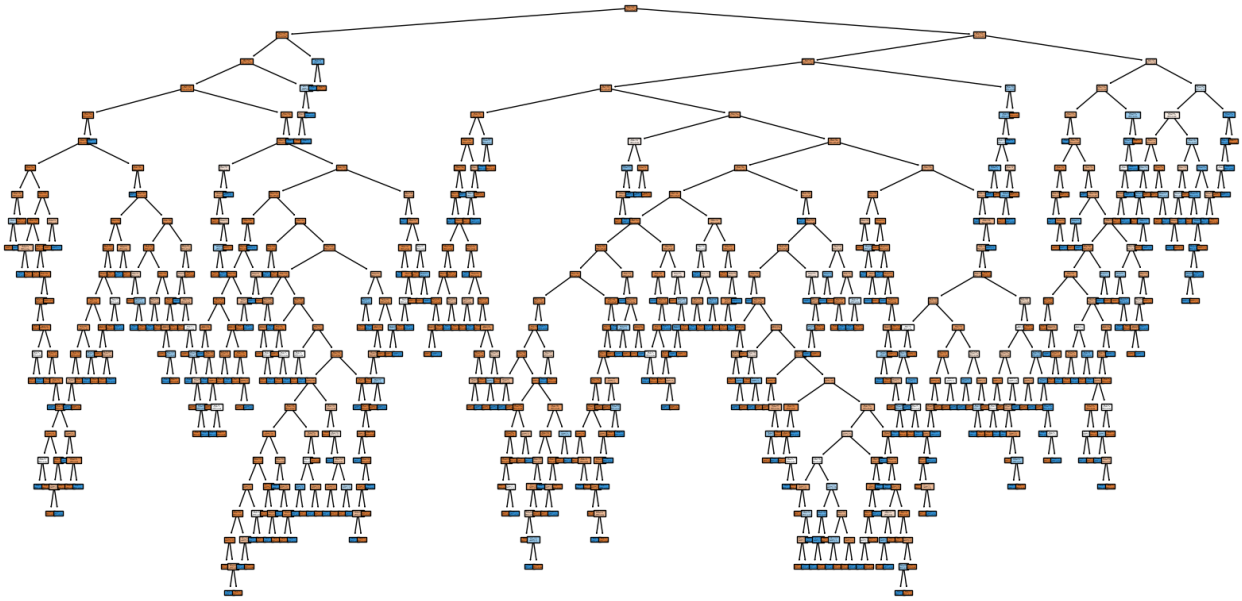
The first step in dealing with the data was to address missing values. To do so, we used print(df.isnull().sum()) to identify which variables held missing values. We found that the variables education, cigsPerDay, BPMeds, totChol, BMI, heartRate, and glucose all had null values, which we had to clean using the fillna function to replace missing values. With the decision tree regression, we replaced missings with median results using SimpleImputer and then scaled the features using StandardScaler. In the case of *k* nearest neighbor, we used KNN imputer and standard scaler to address missings, shifting the r^2 significantly.

With linear regression, which required the most preparation, we started with addressing missings in the train data, which we replaced using medians for almost all features. For the variable "glucose," we replaced missings using a linear regression approach due to the number of missings (285).

To prepare the data for linear regression after cleaning the training data, we created box plots of each feature to visualize distributions and outliers and aid us going forward. When addressing missing values, we followed the same steps for the test data as we did for the training data. We then used OneHotEncoder and "df_change" to convert categorical variables like "education" into a numerical format that allows the model to run more effectively. Using OneHotEncoder required us to split the data into categorical and numerical values before applying the algorithm. Afterward, we combined encoded categorical and numerical columns. Finally, we split the data into X_train_encoded, X_test_econded, y_train, and y_test, where y was the outcome variable "TenYearCHD" and X were the features used to predict y. Now, we can apply the "LinearRegression()" model. We followed similar, shorter steps for splitting the train and test data for the decision tree and kNN.

Results:

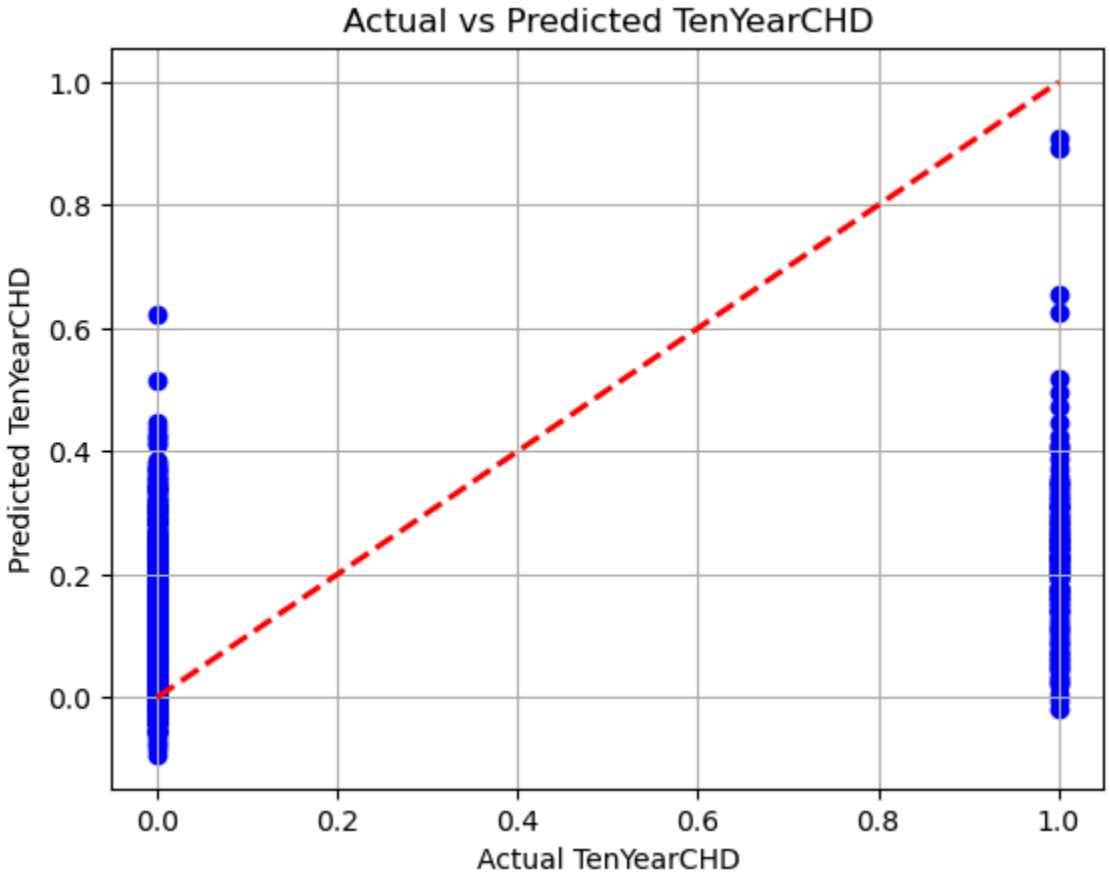
The first algorithm we used to predict the outcome variable was the decision tree, which was pretty straightforward. After replacing missing values and scaling the data, we used "DecisionTreeClassifier" to create and plot the decision tree below.



We then used feature importance scores to rank which features were most likely to predict coronary heart disease, which are calculated based on how much each feature reduces Gini impurity. Features with the highest importance scores (more likely to be predictive of the outcome variable

("TenYearCHD") were "sysBP" (0.146), glucose ("0.142"), "BMI" (0.137), "totChol" (0.122), and "age" (0.118). The least essential features were "BPMeds" (0.014), "diabetes" (0.005), and "prevalentStroke" (0.003).

The second algorithm we used was linear (and eventually logistic) regression. After cleaning and preparing the data, we applied the algorithm, giving us the following result.

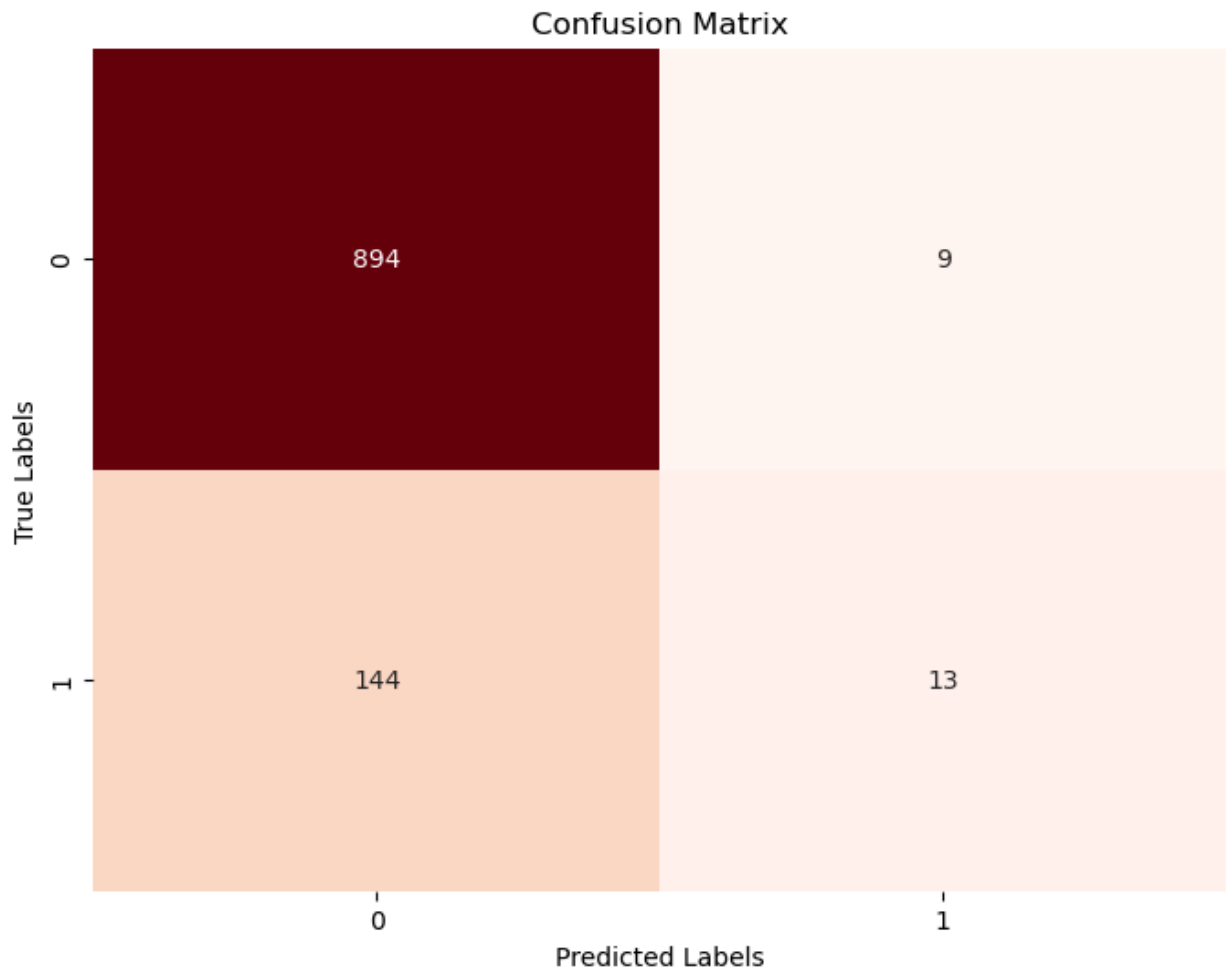


The most positive coefficients (i.e., stronger positive relationship between that feature and the target variable) were those of "prevalentStroke" (0.0929), "BPMeds_1.0" (0.0349), and "sex_1" (0.0308). ("BPMeds_1.0" and "sex_1" were renamed by OneHotEncoder as mentioned previously, where "BPMeds_1.0" represents *true* for the use of anti-hypertensive medication and "sex_1" represents

male). For this regression, the mean squared error was 0.115, the root mean squared error was 0.339, and the r-squared was 0.0887. From these coefficients (which were almost entirely opposite to the results from the decision tree) and the graph, we can see that linear regression does not do a great job at predicting the outcome of "TenYearCHD," especially since it is binary. Due to this, we turned to logistic regression, which is better suited for binary classifications. Logistic regression provided us with the following classification report and confusion matrix:

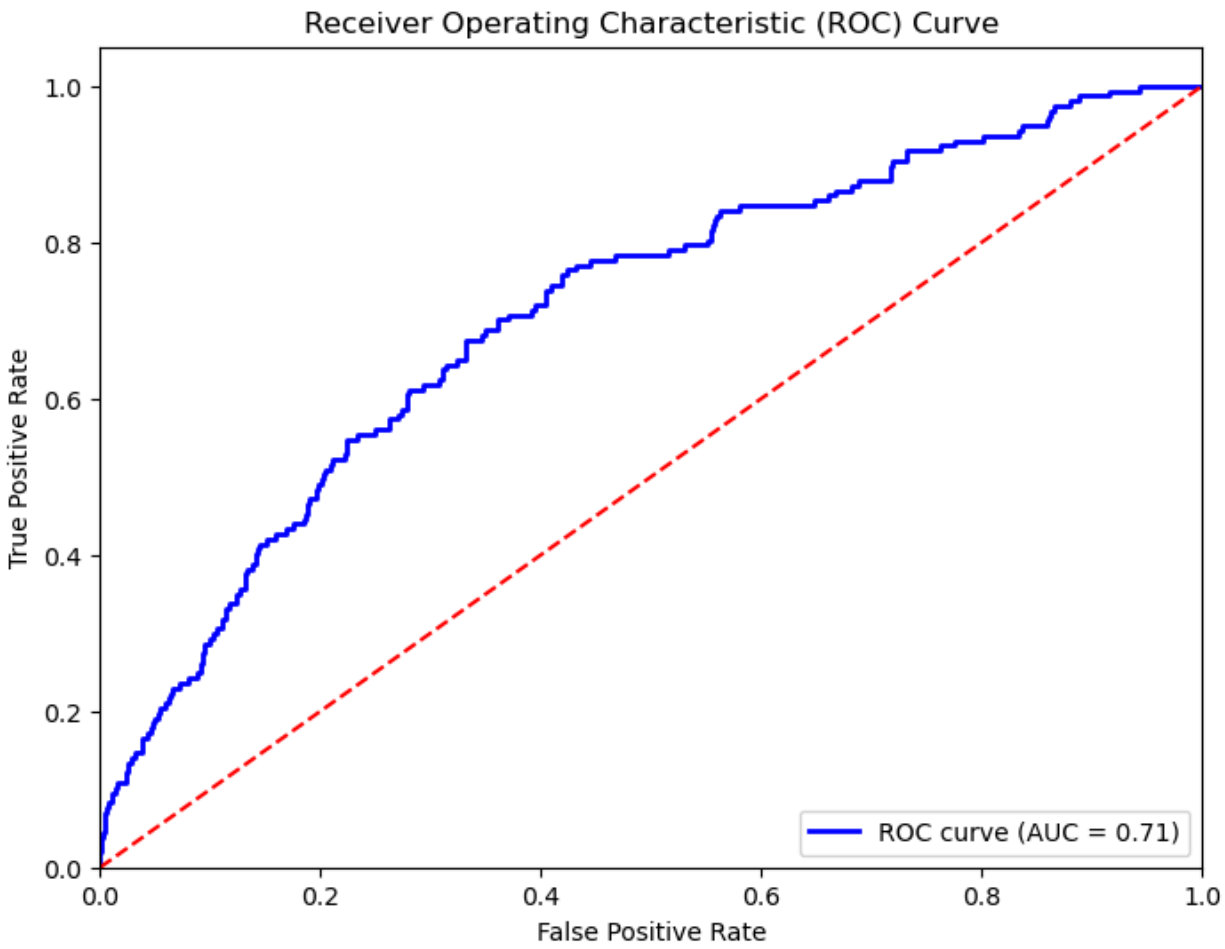
	precision	recall	f-1 score	support
0	0.86	0.99	0.92	903
1	0.59	0.08	0.15	157
accuracy			0.86	1060
macro avg	0.73	0.54	0.53	1060
weighted avg	0.82	0.86	0.81	1060

It is essential to notice the difference in results between classes 0 and 1 here. For example, the recall for class 0 (0.99) is extremely high, indicating that the model correctly classified most instances of not having CHD. On the other hand, recall for class 1 (0.08) is extremely low, indicating the model missed many positive CHD cases. Precision numbers for 0 (0.86) suggest that when the model predicts negative outcomes, it is correct 86% of the time. Precision numbers for 1 (0.59), however, indicate that when the model predicts a positive result, it is only correct 59% of the time. F-1 scores, as well, are skewed in favor of 0. These discrepancies can be observed more plainly in the confusion matrix.



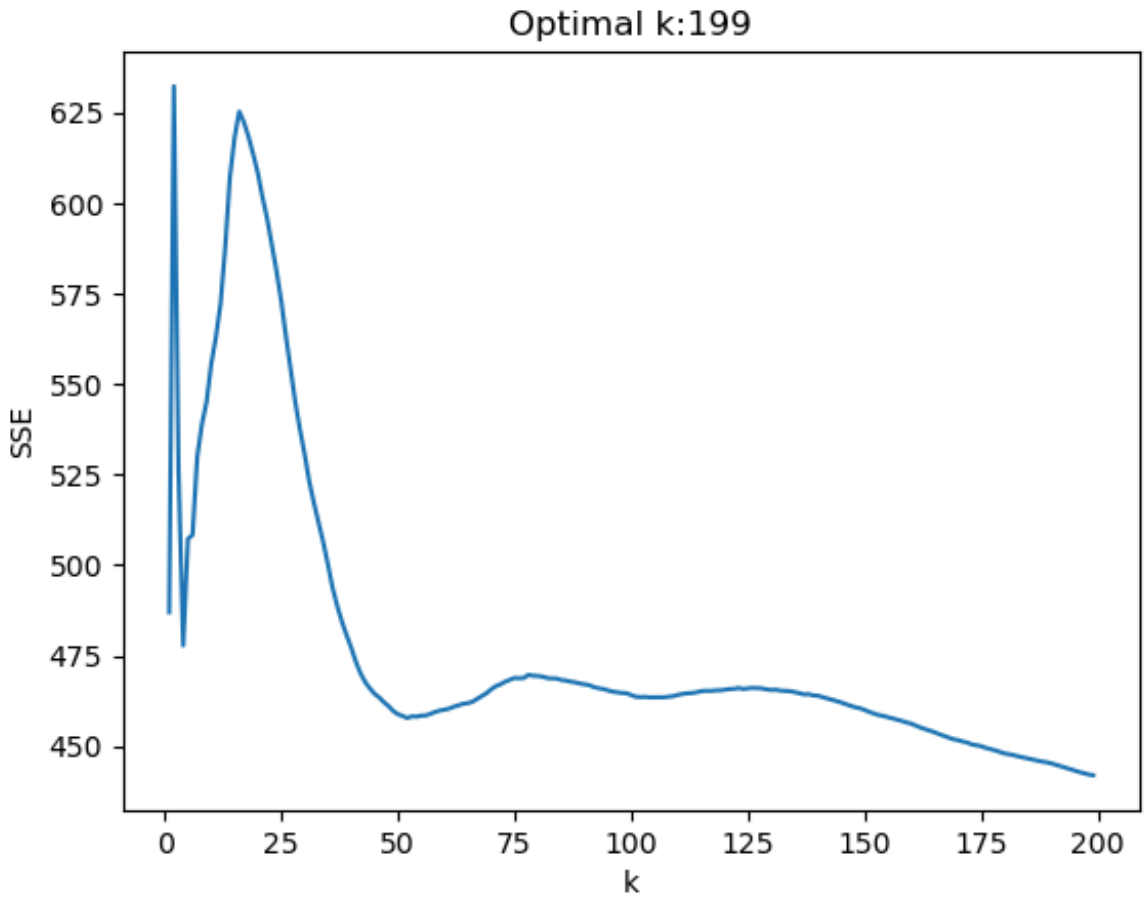
The confusion matrix is quite simple: it shows 894 instances of true negatives (the model correctly predicted the lack of CHD) and 9 cases of false positives (the model predicted the presence of CHD when there were none). For all of the instances of CHD existing, the model did much less well, with 144 false negatives and 13 true positives. Overall, the logistic regression model is much better at predicting the lack of CHD than it is predicting the presence of CHD, which, due to the imbalanced nature of the data set (many more cases of 0 than 1), skews the overall accuracy of the model to 0.86, which is relatively high.

We see this in the visualization of the Receiver Operating Characteristic (ROC) Curve.



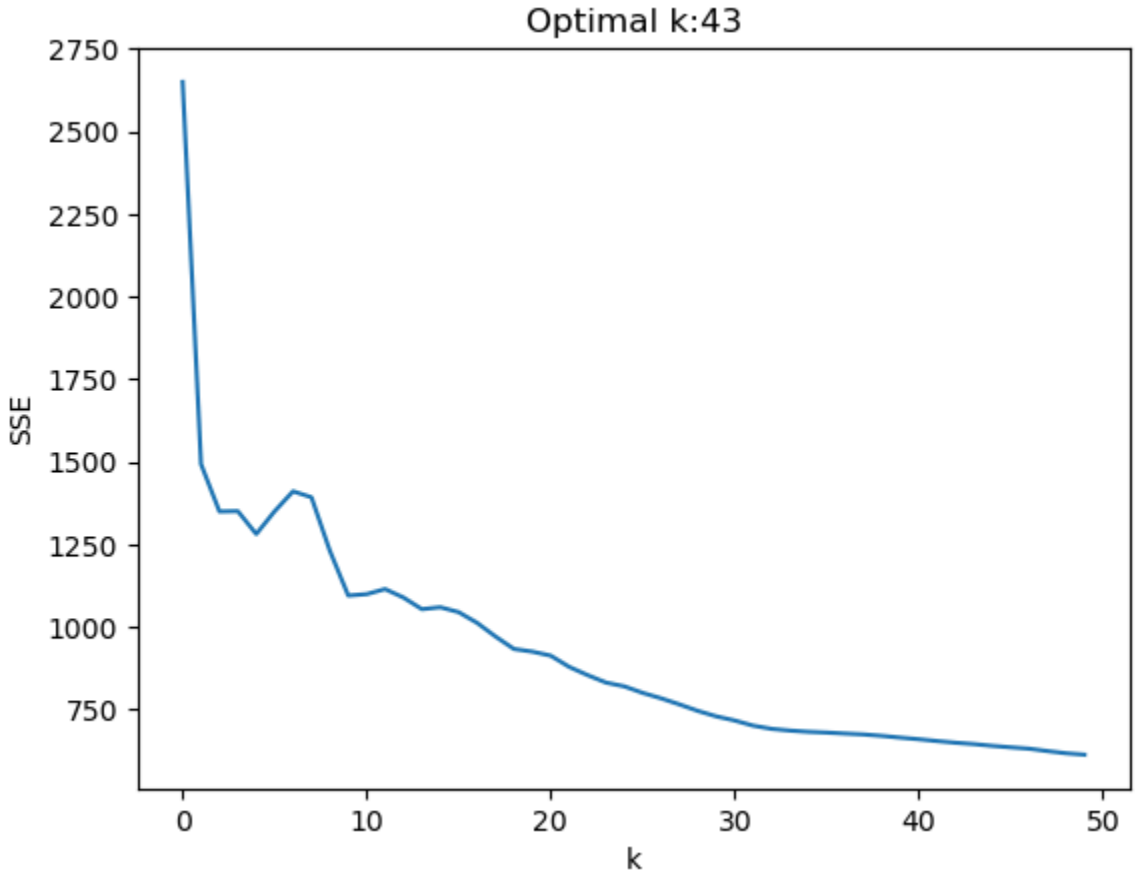
The ROC curve represents the actual positive rate (sensitivity) against the false positive rate (1 - specificity) at various threshold settings. It shows the trade-off between sensitivity and specificity. A curve that hugs the upper left corner indicates a better-performing model. The AUC score quantifies the model's overall performance across all threshold settings. It ranges from 0 to 1, where a score closer to 1 indicates better discrimination power of the model (i.e., the model can distinguish between positive and negative cases more effectively). An AUC of .5 means it would do no better than randomly picking. That said, an AUC of .71 is considered good, though there is room for improvement. Overall, logistic regression is more helpful in predicting CHD than linear regression.

Our last approach was kNN using sklearn, with an upper limit of 200 neighbors (k) to consider. Considering all features, we achieved an optimal k of 199 and an r-squared of -0.07. This indicated that the model's performance was incredibly poor and did not fit the data well, suggesting our original approach was unsuitable for accurately predicting TenYearCHD.



Clearly, we needed to make changes. Instead, we attempted to mix and match a few features to include in the analysis, which we first attempted manually and then automated using "itertools" and "tqdm" to iterate over all combinations of features. We then trained the kNN model for each value of k to find the new optimal k value and the best combination of features, which we found were "sex,"

"currentSmoker," "prevalentStroke," "prevalentHyp," and "diabetes." This left us with an optimal k-value of 43 and an r-squared of 0.0366.



<u>Conclusion</u>:

Our project aimed to develop predictive algorithms for estimating the target variable "TenYearCHD," which measured the 10-year risk of developing coronary heart disease (CHD). We

used k nearest neighbor (kNN), decision trees, and linear and logistic regression. We sought to identify key variables and their strength in predicting CHD.

Our analysis began with meticulously cleaning the data set, handling missing values, and preparing variables for modeling. When it came to implementing the algorithms, we faced a few challenges and limitations, especially in predicting positive CHD outcomes. Most notably, we had to pivot from linear to logistic regression due to the nature of the target variable and its binary. Our logistic regression model showed a high accuracy of 0.86, but due to the imbalanced nature of the data set, this accuracy is likely skewed. The logistic regression successfully predicted true negatives (recall score of 0.99 for class 0) but was poor at predicting true positives (recall score of 0.08 for class 1). Our logistic regression model yielded an AUC score of 0.71, which we could still improve. Still, it is significantly better than our linear regression model, which yielded an r-squared of 0.0887.

Another limitation we faced was the original performance of the kNN algorithm, which yielded an r-squared of -0.07. To address these limitations, we iterated through every combination of feature selections, which improved the r-squared to 0.036, which also could be improved.

Still, our analysis yielded some promising results. Our decision tree suggested the most predictive features (based on how much Gini impurity decreases) were “sysBP,” “glucose,” “BMI,” “totChol,” and “age.” The final iteration of our kNN model, however, suggested a different set of predictive features: “sex,” “currentSmoker,” “prevalentStroke,” “prevalentHyp,” and “diabetes.” Interestingly, 2 of the top 3 variables with the highest positive correlation in our original linear regression model (“sex” and “prevalentsStroke”) overlapped with the results from kNN.

Ultimately, our results were not incredibly conclusive, and there is room for improvement and additional work. In the future, addressing the imbalanced data set would be helpful. It might require more different sampling techniques like oversampling and undersampling or incorporating more features in the data set that might be predictive. There is also room for using more advanced modeling techniques, such as ensemble methods, which combine multiple models simultaneously, or neural networks to achieve more predictive capabilities.