Name: Robert “Robby” Waxman
JHED: RWAXMAN5
Biomedical Data Science 580.475
Professor Caffo/Professor Winslow

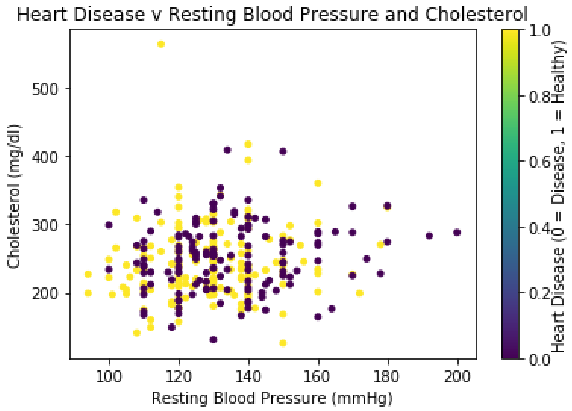
Project on Predicting Heart Disease

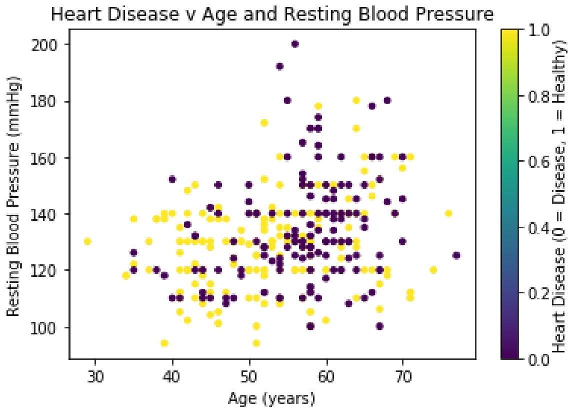
Note: Please check out the attached ipython notebook to see the full code which includes the data frame processing, etc. and is where all the data, plots, etc. I use in this report will be from. Additionally, it includes all my data manipulation etc.

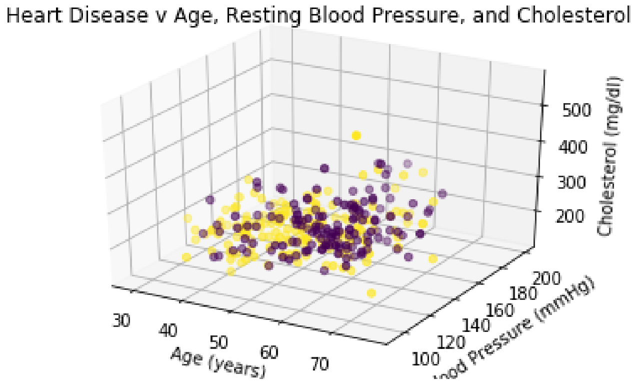
Problem: I am trying to develop and train a model that will predict whether a patient experiencing chest pain has heart disease. The dataset I used for this project comes from UC Irvine’s database (https://archive.ics.uci.edu/ml/datasets/Heart+Disease). This dataset includes patient statistics such as chest pain type, resting electrocardiographic results, sex, age, and many other features, for patients who checked into a hospital for their reported chest pain. Each individual is also tagged as whether having heart disease or not. For this, I chose to predict whether a patient had heart disease by using a logistic regression based on age, sex, chest pain type, resting blood pressure, and cholesterol level.

Basic Scatterplots of Data:

Here, I just ran some preliminary plots to see if the data appeared linearly separable. Unfortunately, it appeared to not, but I thought I could still produce a pretty good model.







Model Fitting:

For this model, I fit a logistic model using scikit learn's LogisticRegression classifier. Since I used the 'lbfgs' solver, I avoided overfitting my model through the built-in regularization that occurs with scikit learning. Furthermore, when using scikit the built-in function also has a default number of iterations (max_iter) which is the maximum number of iterations that the model can take for the solvers to converge. I used the default value of 100 to avoid overfitting as well.

To evaluate my model, I split up my dataset into a train set and test set randomly and then trained my model exclusively on the train set, completely withholding all of the data from the test set. Once the model was trained, I then computed the accuracy of the model on both the train set and test set (0.775 and 0.776 respectively). Since the train and test set both had pretty high and similar accuracies, I can be confident that I did not overfit my train data.

For the logistic model, I can extract the coefficients of the model and so doing this, I got the intercept = 7.597240, and b1 = -0.050866, b2 = -1.631885, b3 = 1.062188, b4 = -0.023025, b5 = -0.005506. These b's correspond to the coefficient associated with age, sex, chest pain, resting bp, and cholesterol respectively. Since this data uses 0 to indicate heart disease and 1 as none, we can for example pull b1 and interpret its meaning. From this, we know a change in 1 year in age corresponds to a change in -0.050866 meaning, meaning the older you get, the more your chances of developing heart disease increase and that is the relative rate. For coefficients for variables like sex (1 = M, 0 = F), we can extract that being a male makes you more likely to be diagnosed with heart disease.

Interpretation:

Using data from the UC Irvine dataset on heart disease, I was able to predict heart disease with approximately 77% accuracy overall for both the training data and test data. This essentially means that I can take the metrics of age, sex, etc. described above for someone who enters a hospital with chest pain and predict with 77% accuracy whether they will or will not have heart disease given these metrics. Some other metrics I investigated for this project were sensitivity and specificity. For the train set, the sensitivity was ~0.75 and specificity was ~0.80, while for the test set, the sensitivity was ~0.69 and the specificity was ~0.85. In both sets, the specificity is significantly higher than sensitivity which means the model is much better at correctly predicting no disease than it is for correctly predicting disease presence. In order to improve model performance, I think it would be a good idea to train and use a larger array of data than just the 5 input variables I included in this project. Furthermore, I think it would also improve the model if we had a significantly larger data set so I could train on much more data and still have data to test my model on.