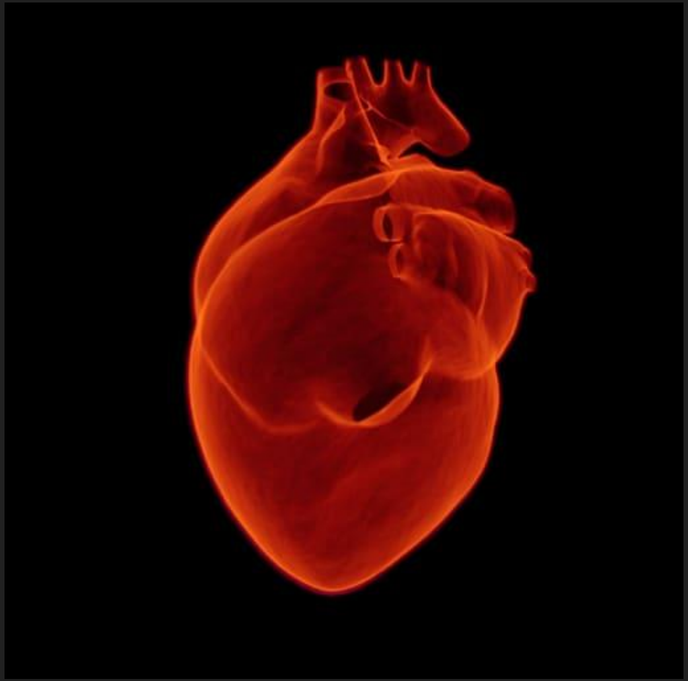
Predicting Heart Disease

Using ML and the UCI Heart Disease Dataset

Impact

- Leading cause of death in the US
- 1 in every 5 deaths (US) is a direct result of the condition
- Someone dies of it every 30-45 seconds
- Costs the country between 2 and 3 billion dollars every year (i.e., healthcare services, medicines, lost productivity due to death, etc.)



Identification

- Identification and prevention of HD is one of the most important topics in the healthcare industry
- Primary prevention models are not accurate enough to be effective.
- Low estimates put their accuracy at just under 60%
- High estimates place them around 80%
- ML models are being used to increase the accuracy of heart disease identification and improve treatment outcomes

Project Goal

- Come up with a supervised learning model that could outperform the current gold standard in primary prevention models
- Use the UCI Heart Disease Dataset
- Predict heart disease with greater than 80% accuracy

Dataset

- UCI Heart Disease Dataset
- Sixteen columns of data and 920 observations
- Four sub-datasets merged to form this dataset, and each sub-dataset corresponds to a different hospital
  - Cleveland
  - Hungary
  - Switzerland
  - VA Long Beach
- Data excluded from modeling process
  - Three predictive features that were missing a third or more of their values
  - About 6% of the patients that were missing 40% or more of their data
- Final dataset had 865 patients and 10 predictive features (five continuous and five binary features)

Continuous

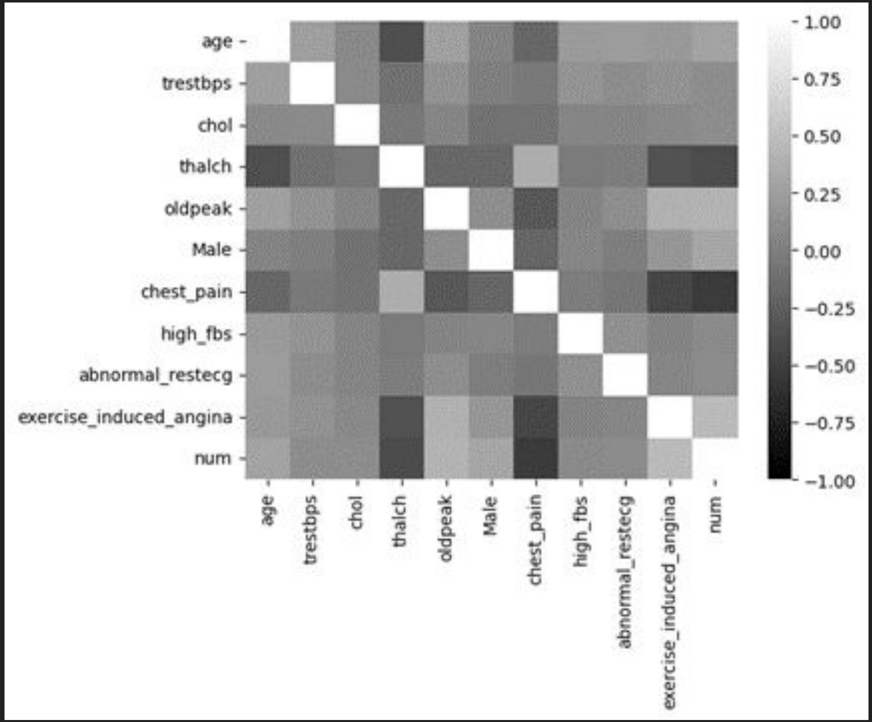
- Age - Age of the patient in years
- Trestbps - Resting blood pressure (in mm Hg) on admission to the hospital
- Chol - Serum cholesterol (in mg/dl)
- Thalch - Maximum heart rate achieved
- Oldpeak - ST depression induced by exercise relative to rest

Binary

- Male - Whether the patient is a male
- Chest-pain - Whether the patient is experiencing chest pain
- High-fbs - Whether fasting blood sugar is above 1200 mg/dl
- Abnormal_restecg - Whether the electrocardiogram results were abnormal
- Exercise_induced_angina - Whether the patient had exercise induced angina

Exploratory Data Analysis

- Project focused on classifying patients as (1) “has heart disease” and (0) “does not have heart disease”
- Bonferroni corrected t-tests of (continuous features) showed significant differences between groups at the .05 level for all five features
- Bonferroni corrected chi-squared tests (binary features) showed significant differences at the .05 level for all but one of the features (high_fbs)
- Thalch and chest pain were the only two features not positively correlated with “has heart disease”



Model Training - Preprocessing

- Data was first split into training (70%) and testing (30%) sets
- Missing values were imputed with either the mean or the median.
- If necessary, features were then scaled

Model Training - Initial Evaluation

- Five different models were evaluated during training:
    1. Logistic Regression
    2. AdaBoost with a Decision Tree estimator
    3. AdaBoost with an SVM Estimator
    4. AdaBoost with a Logistic Regression Estimator
    5. Random Forest
- Hyperparameters were selected using either GridSearchCV or RandomizedSearchCV (n_iter = 180)
- One of each model was selected for the final model selection process
- Hyperparameters for the best models were determined by the highest cross-validation F1 score during training

$$\textbf{F1 Score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}$$

$$\textbf{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

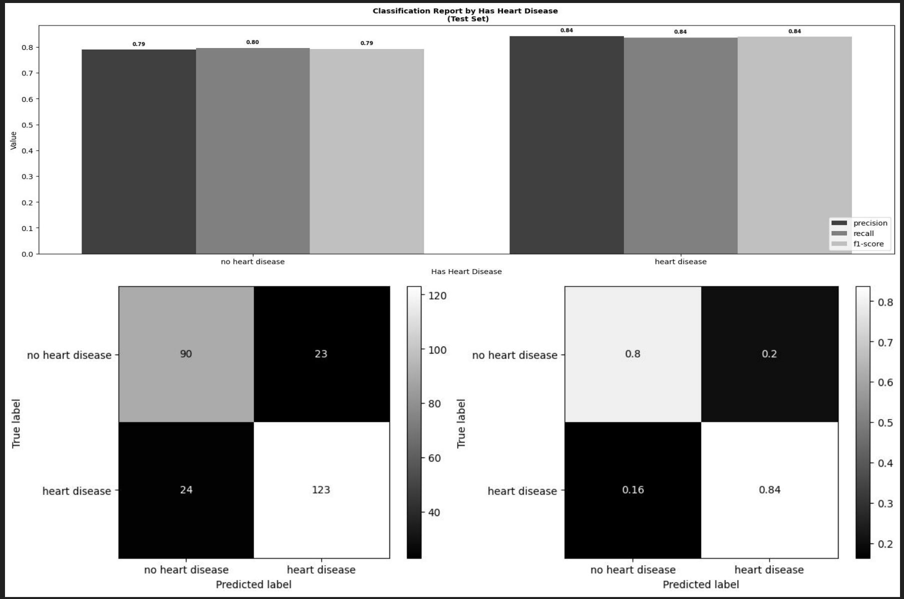
Model Training - Final Selection

- Model with the highest F1 score on the testing set was chosen as the best model
- Adaboost model with Decision Tree estimator (F1 score = 0.84), narrowly outperformed the other Adaboost models
- Final model did not use any scaling and missing values were imputed using the median

```
------------------------------------------------------------------------------------------------
Hyper-parameters
------------------------------------------------------------------------------------------------
adaboostclassifier: AdaBoostClassifier(algorithm='SAMME',
                   estimator=DecisionTreeClassifier(ccp_alpha=0.01,
                                                    criterion='entropy',
                                                    max_depth=1,
                                                    min_impurity_decrease=0.01,
                                                    min_samples_leaf=4,
                                                    min_samples_split=3,
                                                    splitter='random'),
                   learning_rate=0.75, n_estimators=500)

verbose: False

simpleimputer: SimpleImputer()

standardscaler: MinMaxScaler()

adaboostclassifier: AdaBoostClassifier()

simpleimputer__add_indicator: False,
simpleimputer__copy: True,
simpleimputer__fill_value: None,
simpleimputer__keep_empty_features: False,
simpleimputer__missing_values: nan,
simpleimputer__strategy: 'median',
standardscaler__clip: False,
standardscaler__copy: True,
standardscaler__feature_range: (0, 1),
adaboostclassifier__algorithm: 'SAMME',
adaboostclassifier__base_estimator: 'deprecated',
adaboostclassifier__estimator__ccp_alpha: 0.01,
adaboostclassifier__estimator__class_weight: None,
adaboostclassifier__estimator__criterion: 'entropy',
adaboostclassifier__estimator__max_depth: 1,
adaboostclassifier__estimator__max_features: None,
adaboostclassifier__estimator__max_leaf_nodes: None,
adaboostclassifier__estimator__min_impurity_decrease: 0.01,
adaboostclassifier__estimator__min_samples_leaf: 4,
adaboostclassifier__estimator__min_samples_split: 3,
adaboostclassifier__estimator__min_weight_fraction_leaf: 0.0,
adaboostclassifier__estimator__random_state: None,
adaboostclassifier__estimator__splitter: 'random',
adaboostclassifier__estimator: DecisionTreeClassifier(ccp_alpha=0.01, criterion='entropy', max_depth=1,
                       min_impurity_decrease=0.01, min_samples_leaf=4,
                       min_samples_split=3, splitter='random'),
adaboostclassifier__learning_rate: 0.75,
adaboostclassifier__n_estimators: 500,
adaboostclassifier__random_state: None
------------------------------------------------------------------------------------------------
```

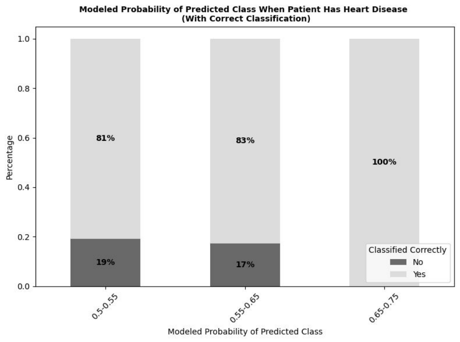
Testing Set - Further Analysis

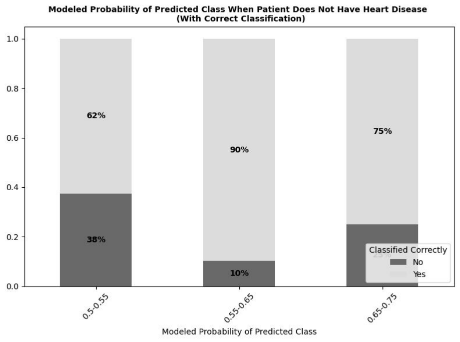
- Model was better at classifying heart disease patients than those without the condition
- Better to make this type of error (Type I) since patients are better off taking precautionary measures with their lifestyle if they are a fringe case

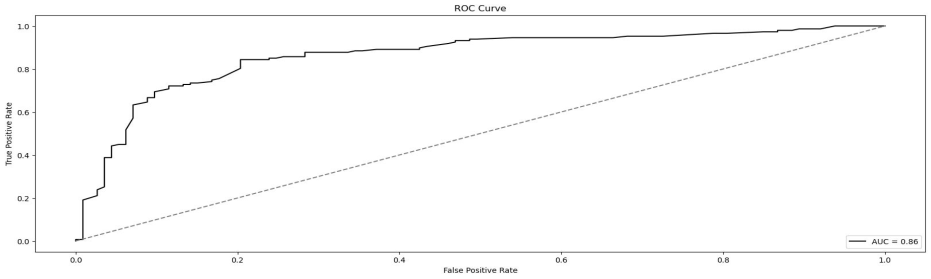


Testing Set - Further Analysis (continued)

- Model tended to perform better when it was more confident in a prediction
- Effect was more pronounced for cases when the patient had heart disease
- Plotting the true positive rate against the false positive rate showed that it was doing a really good job overall (despite differences between classes)

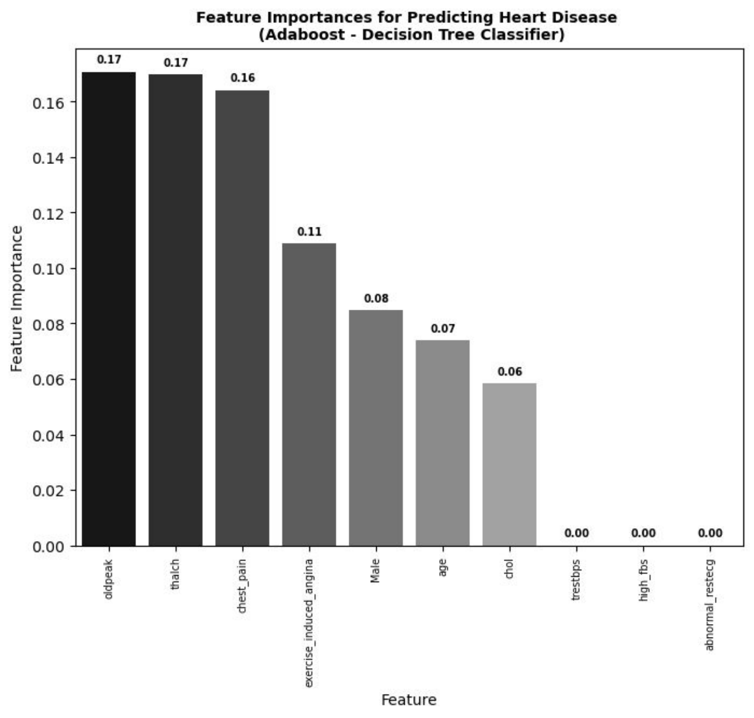






Feature Importances

- Decision Tree estimator gave insight into which features were most important (ordered below):
  1. ST depression induced by exercise relative to rest (oldpeak)
  2. Maximum heart rate achieved (thalch)
  3. Whether or not a patient had chest pain
- Exercise induced angina, sex, age, and cholesterol were also important features



Final Thoughts and Future Research

- Model met initial goal of predicting heart disease with > 80% accuracy
- Was better at identifying cases when patients had heart disease than when they did not
- Ideas for future research:
    1. Add other features related to exercise (2 of the top 4 features were exercise-related)
    2. Incorporate three predictive features that were dropped at the beginning of the project
    3. Add data from other hospitals
    4. Explore whether the predicted probabilities could be used to come up with a risk score

Thank you.