ISM 6357 Business Intelligence & Analytics Tools & Programming Spring 2019
Instructor: Dr. Weaver Cloud Project – Azure ML Studio May 30, 2019

Predicting Presence of Heart Disease using Machine Learning

By: Jules Caruso, Moe Jobarteh, Angelina Kiman, Nhi Phan & Charles Tay

Problem Description

According to the World Health organization, coronary heart disease is the undisputed global cause of death (2018). Based on the most recent global cause of death release from the World Health Organization; 10 million deaths were due to heart disease in 2016. Heart disease has remained the leading cause of death in the world for the past 18 years. The goal of this experiment is to use the Heart Disease dataset from the Machine Learning repository accessed through driven data to train models for heart disease prediction. We are trying to identify one or a combination of models that would be able to predict heart disease and help save lives.

Dataset

We will be building the heart disease classification models using the heart disease data from the Cleveland Clinic. The dataset contains data from 270 people with datapoints recorded for each of individual across 13 attributes or features pertaining to heart disease. Below are the features of the dataset and the parameters that define them.

1.	age	: age in years
2.	sex	: 1=male, 0= Female
3.	chest_pain_type	: chest pain type (4 types)
4.	blood_pressure	: resting blood pressure mm/hg
5.	cholesterol	: serum cholesterol in mg/Dl
6.	blood_sugar	: fasting blood sugar > 120 mg/Dl (1=true, 2=false)
7.	rest_ecg	: resting electrocardiographic results
8.	max_heart_rate	: maximum heart rate achieved
9.	exercise_induced_angina	: 1=yes, 0=no
10.	st_depresion	: ST depression induced by exercise relative to rest
11.	st_slope	: Slope of Peak exercise ST segment
12.	num_major_vessels	: colored by fluoroscopy (0-3)
13.	thalassemia	: 3=normal, 6= fixed defect, 7=reversible defect

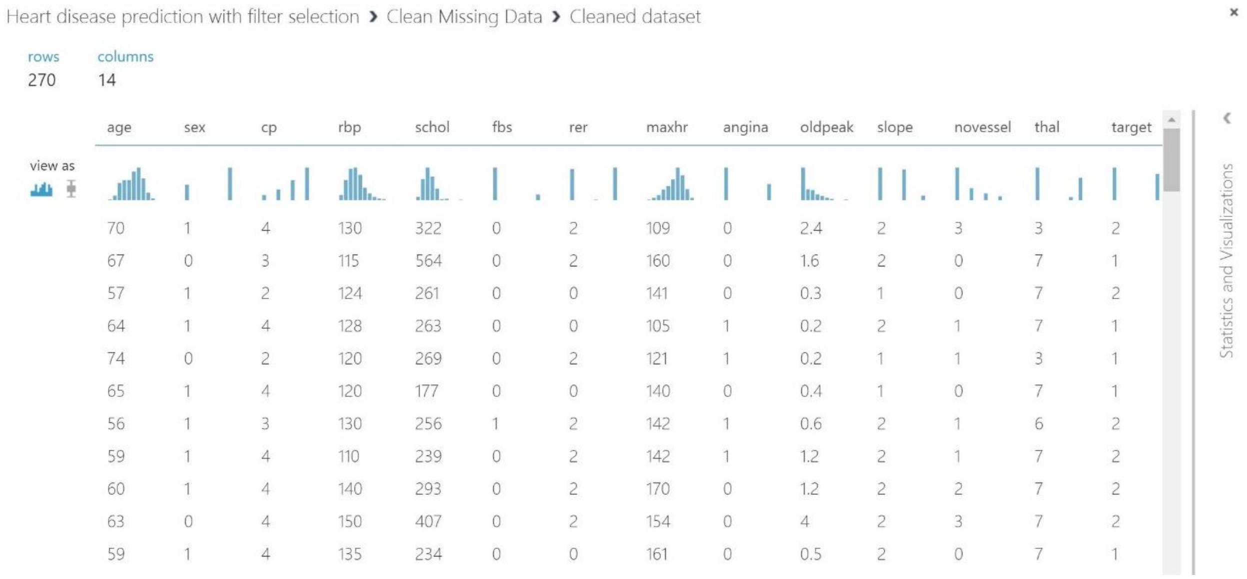
The Experiment – Background:

For this experiment, the dataset will be running through five different algorithms, all of which would be analyzed for accuracy, precision and recall; a recommendation of most predictively potent model(s) for this particular case would be provided based on ensuing results. The five algorithms we will using to predict heart disease are: Two Class Boosted Decision Tree, Two Class Decision Forest, Two Class Naïve Bayes Model, Support Vector Machine and Two Class Logistic Regression.

Cleaning the Data

Before running our algorithms, we needed to make sure the data was clean and free of missing values. After cleaning the data, a snapshot statistical summary of datapoints was obtained. The purpose of this exercise is to give us a standard range for applicable features. For example, with age, we wanted to look at the age distribution of this dataset as it correlates to the real world because heart disease is not just a problem for older adults. Based on our data summary we could tell that individuals in our data set had minimum age of 29 and maximum age of 77 with mean age of 54 and std deviation of 7.5 years. We found the age distribution as plausible and representative of the real world. We applied this same logic to the "sex" feature. With 0 being female and 1 being male, a mean of .67 tells us that the data is more skewed towards men than women. These considerations have been noted and will be made available to the beneficiaries of this experiment. Shown below are graphics of the cleaned data and its statistical summary.

Cleaned Data Set

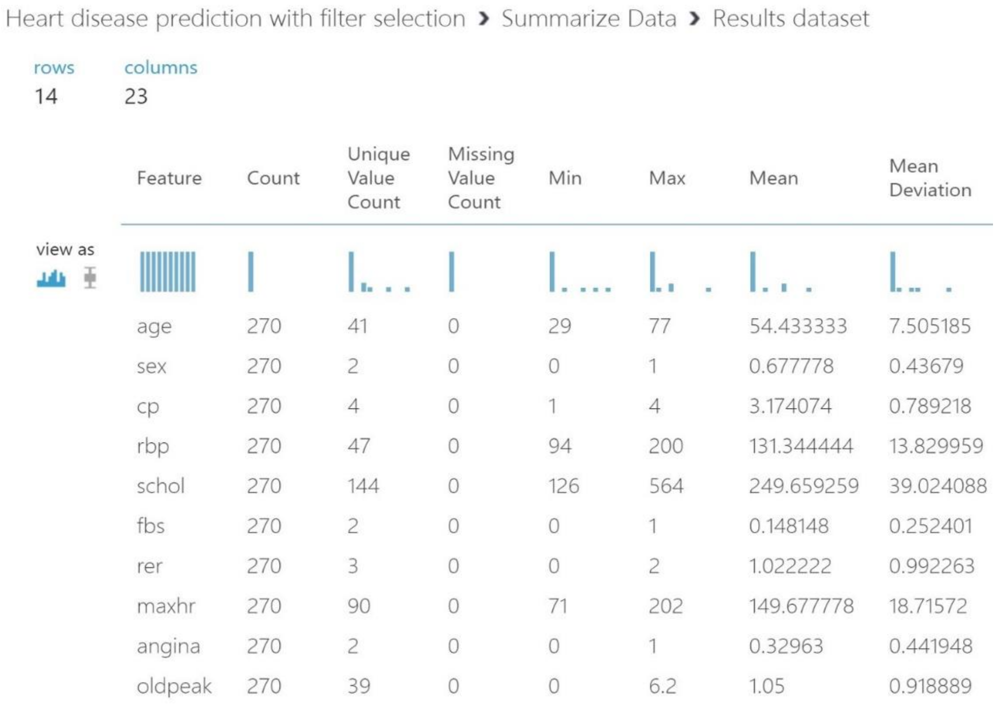


Heart disease prediction with filter selection › Clean Missing Data › Cleaned dataset

rows 270 columns 14

age	sex	cp	rbp	schol	fbs	rer	maxhr	angina	oldpeak	slope	novessel	thal	target
70	1	4	130	322	0	2	109	0	2.4	2	3	3	2
67	0	3	115	564	0	2	160	0	1.6	2	0	7	1
57	1	2	124	261	0	0	141	0	0.3	1	0	7	2
64	1	4	128	263	0	0	105	1	0.2	2	1	7	1
74	0	2	120	269	0	2	121	1	0.2	1	1	3	1
65	1	4	120	177	0	0	140	0	0.4	1	0	7	1
56	1	3	130	256	1	2	142	1	0.6	2	1	6	2
59	1	4	110	239	0	2	142	1	1.2	2	1	7	2
60	1	4	140	293	0	2	170	0	1.2	2	2	7	2
63	0	4	150	407	0	2	154	0	4	2	3	7	2
59	1	4	135	234	0	0	161	0	0.5	2	0	7	1

Data Summary



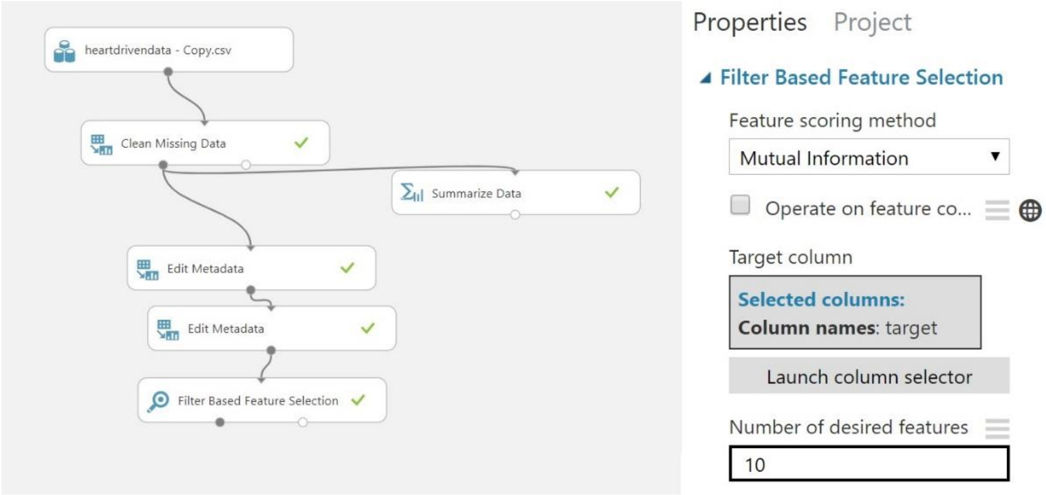
Heart disease prediction with filter selection › Summarize Data › Results dataset

rows 14
columns 23

Feature	Count	Unique Value Count	Missing Value Count	Min	Max	Mean	Mean Deviation
age	270	41	0	29	77	54.433333	7.505185
sex	270	2	0	0	1	0.677778	0.43679
cp	270	4	0	1	4	3.174074	0.789218
rbp	270	47	0	94	200	131.344444	13.829959
schol	270	144	0	126	564	249.659259	39.024088
fbs	270	2	0	0	1	0.148148	0.252401
rer	270	3	0	0	2	1.022222	0.992263
maxhr	270	90	0	71	202	149.677778	18.71572
angina	270	2	0	0	1	0.32963	0.441948
oldpeak	270	39	0	0	6.2	1.05	0.918889

Feature Selection

For this experiment we chose to focus on the 10 most important features within our dataset. It is our goal to prioritize the features that are most linked to heart disease. We want to know which of the list of features tell us the most about the dependent variable; the "target"; identifying presence of heart disease or not. The importance of this feature filter for us is that it translates to statistical significance. We want to load our algorithms with significant data in order to make impactful deductions and increase the potency of our models.



Filter Based Feature Selection

Feature scoring method
Mutual Information

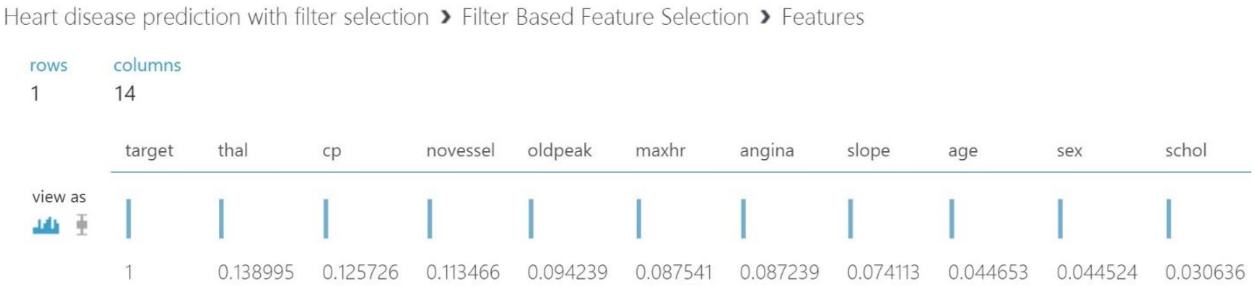
☐ Operate on feature co...

Target column
Selected columns:
Column names: target

Launch column selector

Number of desired features
10

We selected mutual information as feature scoring method because we are dealing with a lack of numeric uniformity in how features are quantified. Mutual Information measures how much information the presence or absence of any feature contributes to making the correct classification decision, in this case; is heart disease present or not ("target" column). When we ran the filter feature selection the results were as follows:



Heart disease prediction with filter selection › Filter Based Feature Selection › Features

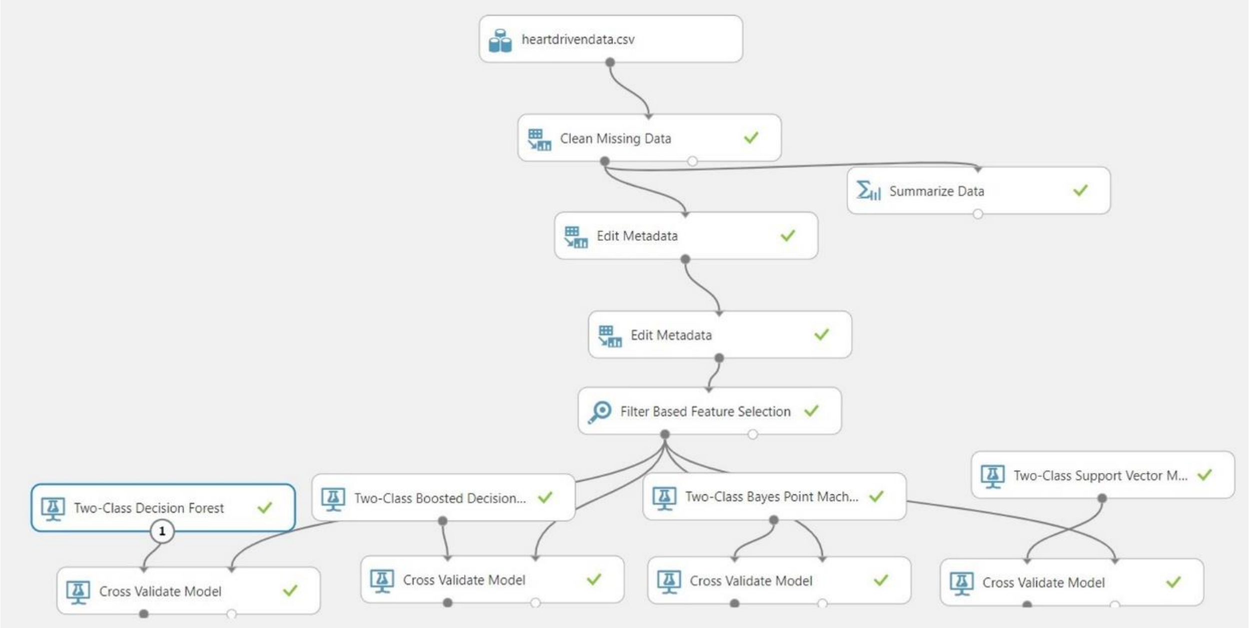
rows 1 columns 14

	target	thal	cp	novessel	oldpeak	maxhr	angina	slope	age	sex	schol
view as	1	0.138995	0.125726	0.113466	0.094239	0.087541	0.087239	0.074113	0.044653	0.044524	0.030636

Thalassemia at about 14% was our most correlated feature to heart disease with Cholesterol levels being the least out of the ten features selected at about 3%. The rest of the features; which are fasting blood sugar, resting ecg and resting blood pressure all had a correlation factor below 2% when it comes to predicting the predicting the presence of heart disease.

Cross Validation Technique

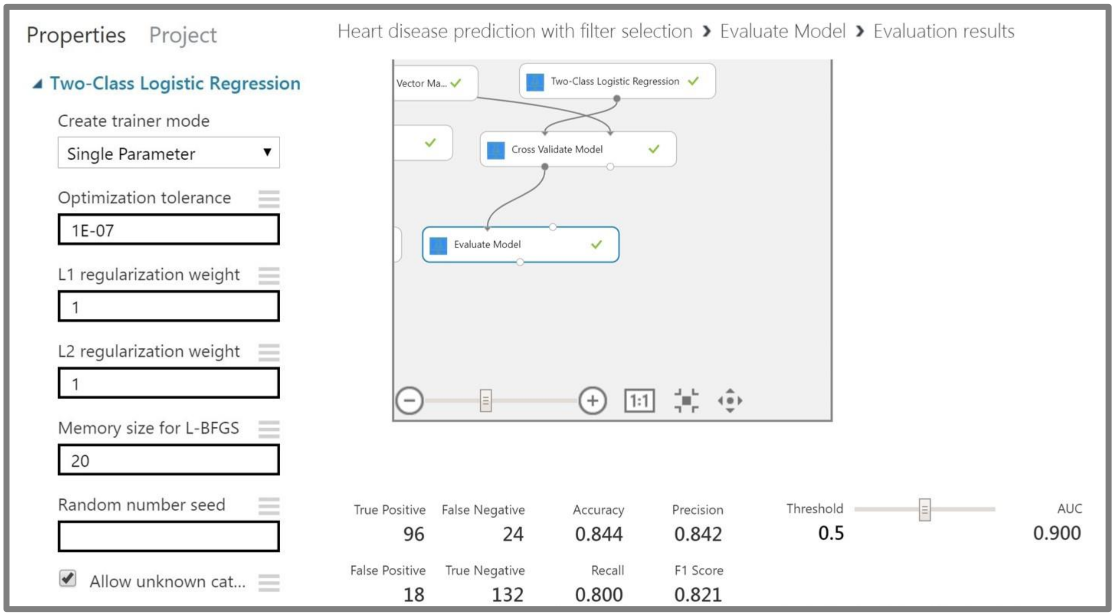
Since our goal of this experiment is predictability of heart disease given a combination of features, we chose to not individually score and train the models but rather cross validate them. Cross Validation, we believe is ideal for this experiment because it helps mitigate the problem of overfitting (a situation whereby the models are trained to learn about the actual dataset so closely the models "memorize" and mimic the original dataset). Overfitting would be catastrophic to our efforts to solve a global problem that is heart disease. We cross- validated all five models to ensure that our models do not learn about our data to "memorize" it but rather to "understand" the data so they can predict results given new information. We believe that with cross validation technique, the predictive power of our models has been optimized, giving us more confidence about the resulting accuracy and precision that we obtain. Below is a graphic showing cross-validation of the models. Since we have a small dataset and we could end up getting sufficiently big differences in quality or different optimal parameters between folds. We choose k=10, as a general rule; empirically k=10 tends to yield test error estimates that suffer neither from excessively high bias nor high variance.



Models and Model Evaluations

Two Class Logistic Regression:

The first Machine Learning algorithm we ran was the two-class logistic regression because it is the go-to method for predicting binary classification outcomes. Our ultimate goal is to classify people under two categories of "heart disease not present" or "heart disease present" with both classes represented by a "1" or "2" respectively. Below are the properties that we ran the model with and the ensuing evaluation results.


Properties
Project
Heart disease prediction with filter selection › Evaluate Model › Evaluation results
Vector Ma... ✓
Two-Class Logistic Regression ✓
Cross Validate Model ✓
Evaluate Model ✓
1:1

Two-Class Logistic Regression

Create trainer mode: Single Parameter

Optimization tolerance: 1E-07

L1 regularization weight: 1

L2 regularization weight: 1

Memory size for L-BFGS: 20

Random number seed:

☑ Allow unknown cat...

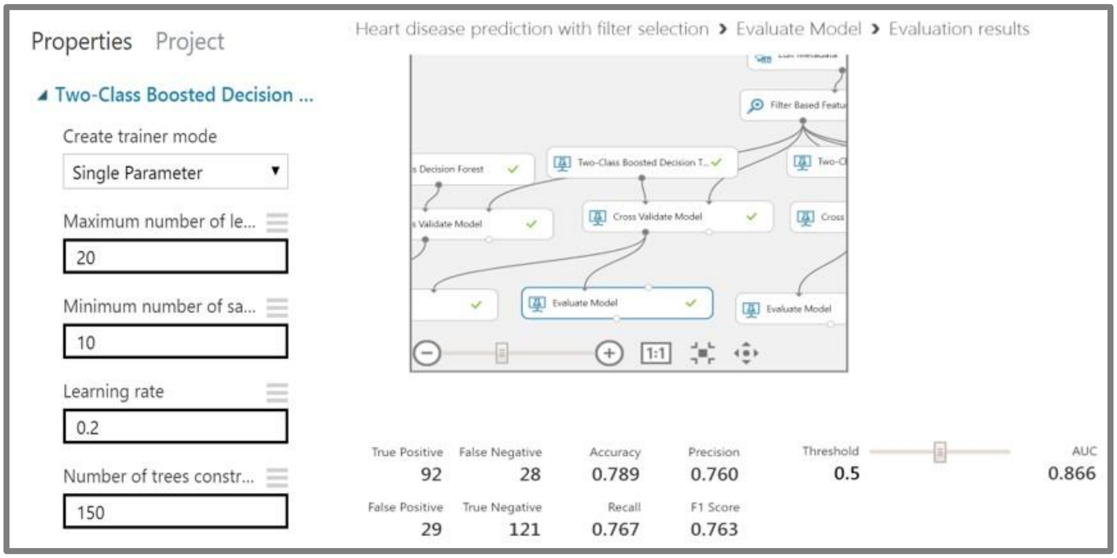
True Positive	False Negative	Accuracy	Precision	Threshold	AUC
96	24	0.844	0.842	0.5	0.900
False Positive	**True Negative**	**Recall**	**F1 Score**		
18	132	0.800	0.821		

We knew the independent parameter that we are testing against (target column) so we chose single parameter logistic regression and left memory size at 20 because the data set is not large enough to require longer training for optimization.

Overall this model had an accuracy and precision level of about 84% with a recall level of 80%. Overall this model will be able to predict heart disease with an accuracy of 84%. This is considered significant for us and viable for future application.

Two Class Boosted Decision Tree:

The decision tree model predicts the value of a target variable based on several input variables. Since we have a target variable we are trying to predict, we chose this algorithm as our second stop. We ran it with 100 trees and then 150 trees at a learning rate of 0.2 with a single parameter as shown below. Note, 50 extra trees on the second iteration had negligible effect on the results. By increasing the number of trees, the algorithm took a few seconds longer to run. In principle this helps reduce variance and increase accuracy. Given the size of the data set the results were not as impressive as the logistic regression model as shown below:



Two-Class Boosted Decision ...

Create trainer mode: Single Parameter

Maximum number of le...: 20

Minimum number of sa...: 10

Learning rate: 0.2

Number of trees constr...: 150

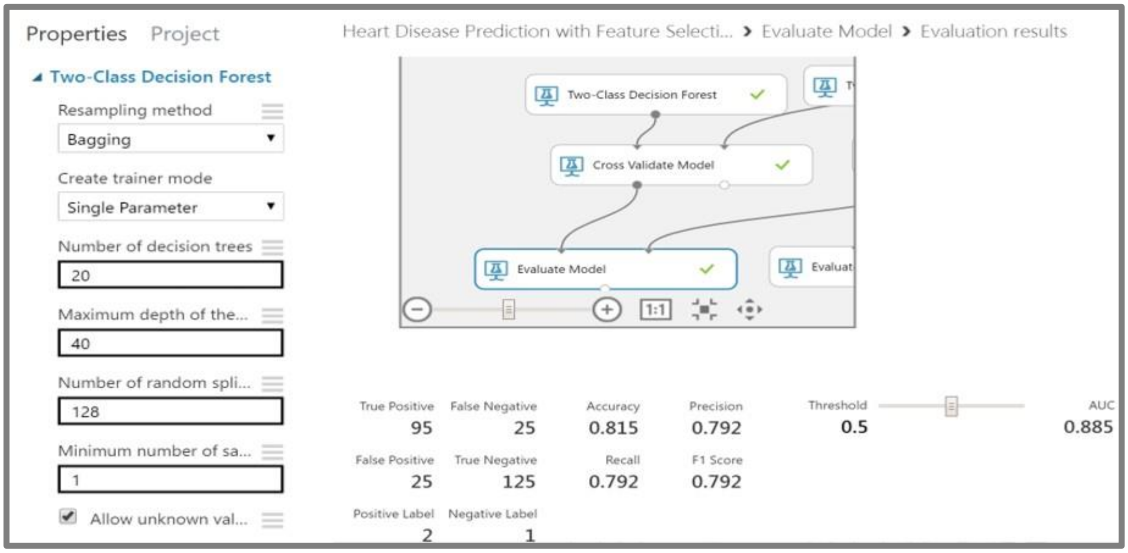
True Positive	False Negative	Accuracy	Precision	Threshold	AUC
92	28	0.789	0.760	0.5	0.866
False Positive	**True Negative**	**Recall**	**F1 Score**		
29	121	0.767	0.763		

This model had an accuracy score of 78.9% and Precision 76% with a recall of 76%. These numbers are satisfactory but not impressive by our standards. Whether the boosted decision is recommended for future use with this dataset will depend on the results of remaining three models.

Two Class Decision Forest:

The third algorithm we ran was a two-class decision forest. By using a decision forest which is, in essence an aggregation of multiple decision trees. We hope to see a better accuracy compared to the boosted decision tree. To further solidify the model accuracy, we increased the number to trees to twenty (20) and chose to run the algorithm with the "Bagging" method.

Bagging allows for each tree to be grown from a new sample randomly generated from the original dataset. Bagging enables the model to accommodate variables with wide ranges and varied distributions. The properties of the algorithm and its performance were as follows:


Properties
Project
Heart Disease Prediction with Feature Selecti... > Evaluate Model > Evaluation results
Two-Class Decision Forest
Cross Validate Model
Evaluate Model
Evaluat
1:1

Two-Class Decision Forest

Resampling method: Bagging

Create trainer mode: Single Parameter

Number of decision trees: 20

Maximum depth of the...: 40

Number of random spli...: 128

Minimum number of sa...: 1

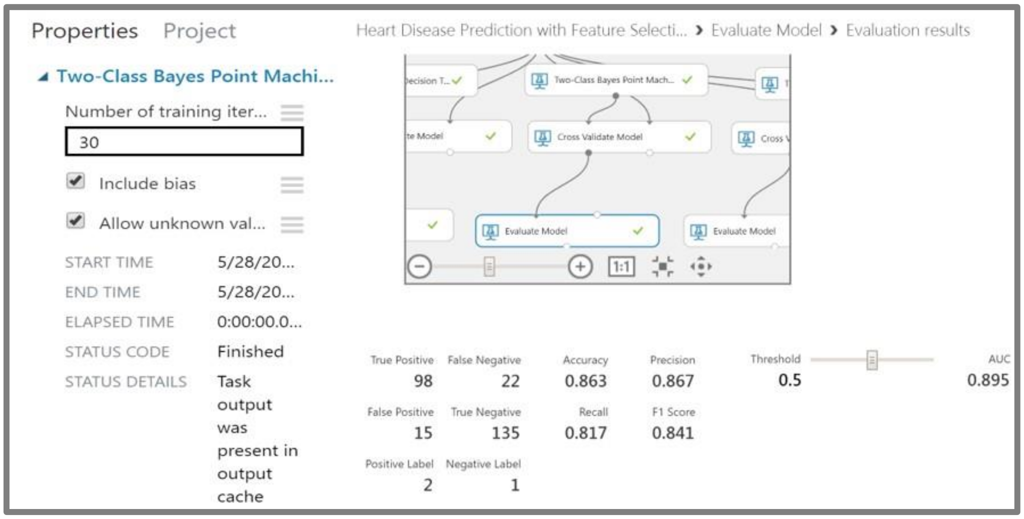
☑ Allow unknown val...

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
95	25	0.815	0.792	0.5	0.885
False Positive	**True Negative**	**Recall**	**F1 Score**		
25	125	0.792	0.792		
Positive Label	**Negative Label**				
2	1				

The decision forest had an accuracy score of 81.5% and Precision 79.2% with a recall of 79.2%. The numbers from the decision forest are higher than the decision tree as we envisioned, on average, accuracy, precision and recall all increased by 3%. Given the size of the heart dataset, we saw minimal increase in accuracy and a longer training time for the algorithm when we increased the number of trees.

Two Class Naïve Bayes Model:

We chose to run the two class Naïve Bayes model because we believe the independence assumption holds in terms of our features. While all the futures could contribute to heart disease they do not directly cause or depend on each other e.g age or sex doesn't lead to high cholesterol and vice-versa. This is just one example among 15 others that could be drawn from the dataset. We ran the Naïve Bayes algorithm with the default 30 iterations. We found 30 iterations to be enough because of the rather small dataset and efficient model training time. Through our feature selection we recorded that Thalassemia had the highest correlation factor to heart disease at about 14%. Since the correlation of the features was low with regards to the target we did not deem it necessary to change model properties. Shown below are the properties of the Naïve Bayes model and resulting performance.



Properties Project

Heart Disease Prediction with Feature Selecti... › Evaluate Model › Evaluation results

Two-Class Bayes Point Machi...

Number of training iter... 30

☑ Include bias

☑ Allow unknown val...

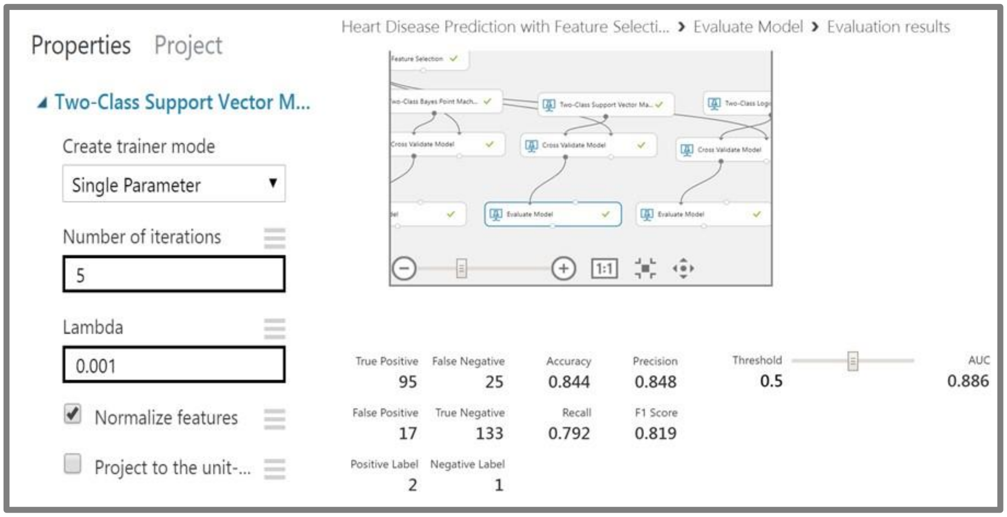
START TIME 5/28/20...
END TIME 5/28/20...
ELAPSED TIME 0:00:00.0...
STATUS CODE Finished
STATUS DETAILS Task output was present in output cache

True Positive 98
False Negative 22
Accuracy 0.863
Precision 0.867
Threshold 0.5
AUC 0.895
False Positive 15
True Negative 135
Recall 0.817
F1 Score 0.841
Positive Label 2
Negative Label 1

The two-class Naïve Bayes algorithm resulted in 86% accuracy and about 87% precision. Higher than any other model that we trained and evaluated. Along with the Logistic Regression model, both scored a recall of about 80%. The Naïve Bayes model was the best performing algorithm in the experiment.

Two-Class Locally Deep Support Vector Machine:

We chose to run the Support Vector Machine (SVM) model because we wanted to see how this model would perform compared to decision trees, decision forest and logical regression models. SVM is capable of doing both classification and regression but tends to take a longer time to run. SVMs excel at identifying complex boundaries, we hoped that this algorithm would set better boundaries within the parameters of the dataset and predict directly what factors lead to heart disease. To further optimize the boundary setting power of the SVM, we ran the algorithm with 5 iterations. The model took longer to run but resulted in better accuracy than all the decision tree and regression models. Shown below are the SVM algorithm properties and performance.



Properties Project

Heart Disease Prediction with Feature Selecti... › Evaluate Model › Evaluation results

Two-Class Support Vector M...

Create trainer mode: Single Parameter ▾

Number of iterations 5

Lambda 0.001

☑ Normalize features

☐ Project to the unit-...

True Positive 95
False Negative 25
Accuracy 0.844
Precision 0.848
Threshold 0.5
AUC 0.886
False Positive 17
True Negative 133
Recall 0.792
F1 Score 0.819
Positive Label 2
Negative Label 1

The Support Vector Machine was the second most accurate algorithm in the experiment behind Naïve Bayes model.

Model Analysis and Recommendations

The 'Add Rows' module was used to combine all the accuracy rates of the predictions of all the models we used in the project. It appends a set of rows from an input dataset to the end of another dataset. The results in the 'Evaluate Model' Module was connected to the 'Add Rows' module to create a column-row result with all the accuracy rates from the models

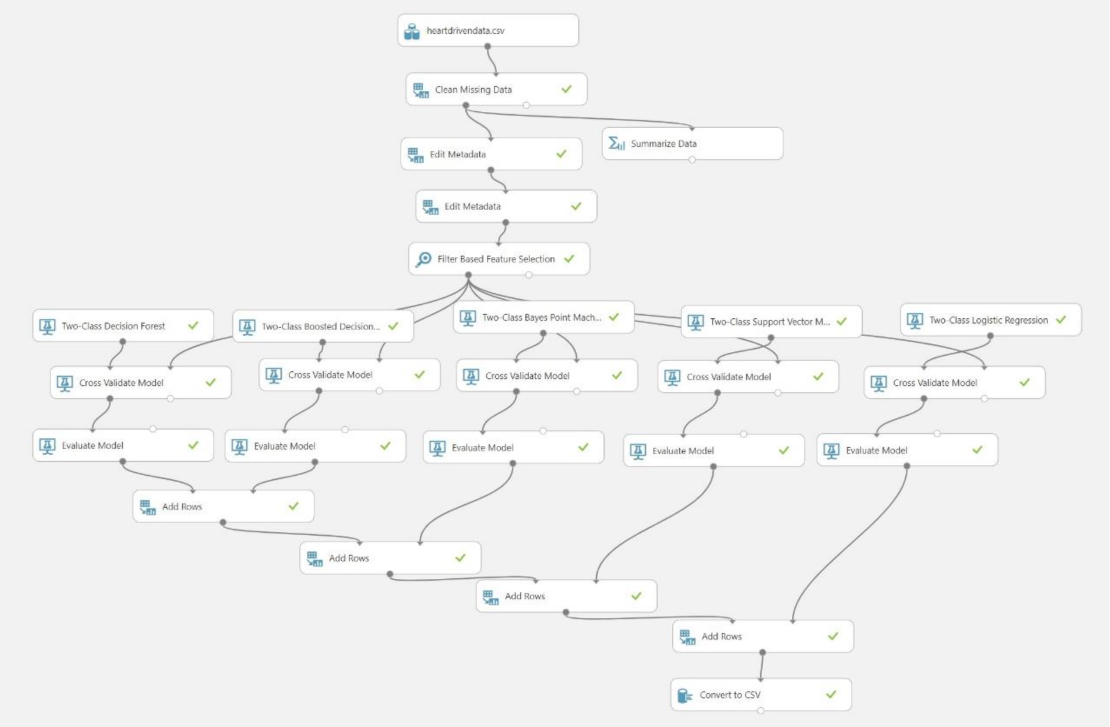
Below is table summarizing the performance of the 5 models in terms of accuracy, precision recall and f-1 score.

Model (Two-Class)	**ACCURACY**	**PRECISON**	**RECALL**	**F1 SCORE**
Logistic Regression	0.844	0.842	0.800	0.821
Decision Tree (Boosted)	0.789	0.760	0.767	0.763
Decision Forest	0.815	0.792	0.792	0.792
Naive Bayes	0.863	0.867	0.817	0.841
Support Vector Machine	0.844	0.848	0.792	0.819

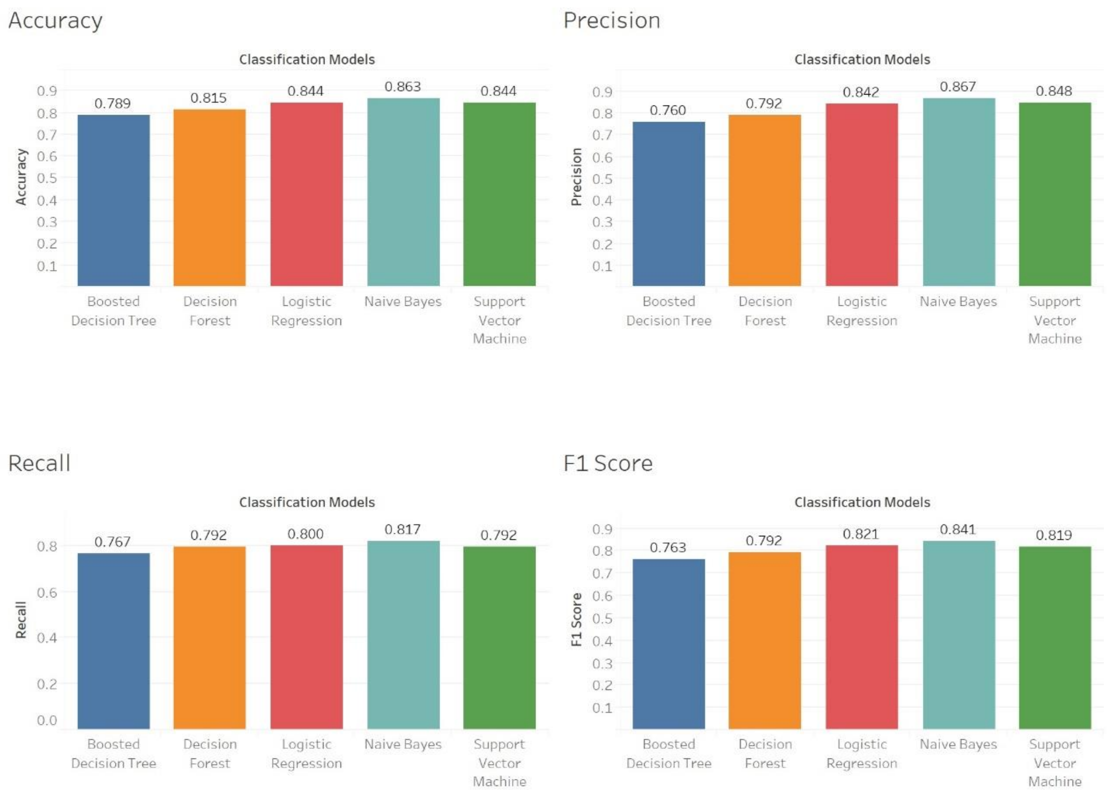
For this experiment we have chosen to evaluate the performance of our models based on their f-1 score. We believe that the F-1 score, which is a weighted average between precision and recall is more useful than accuracy, precision or recall if individually used when analyzing model performance.

Our dataset is unevenly distributed. We are dealing with identifying heart disease in people, the problem that we are trying to solve directly involves human life. The cost of having false positives and false negatives in this experiment needs to be mitigated as much as possible. The nature of this real world problem we are trying to solve makes it almost impossible to solely just look at model accuracy. For this reason, we believe the F1 score gives us a better judgement of model performance. Based on this premise, we believe that the Naïve Bayes model with an F1 score of 0.819 performed the best in our experiment, followed by Support Vector machine with a score of 0.891. We would like to recommend these two models for this data because the other remaining models all fall under the .800 mark or 80th percentile in terms of F1 Score. All models scored relatively high but given the size of the dataset, we are choosing to recommend the two most proficient models which are Naïve Bayes and Support Vector Machine.

Appendix I – Screenshot of Entire Model



Appendix II – Dashboard of Performances from our Azure ML Algorithms


Accuracy
Classification Models
Accuracy
Boosted Decision Tree
Decision Forest
Logistic Regression
Naive Bayes
Support Vector Machine
Precision
Classification Models
Precision
Boosted Decision Tree
Decision Forest
Logistic Regression
Naive Bayes
Support Vector Machine
Recall
Classification Models
Recall
Boosted Decision Tree
Decision Forest
Logistic Regression
Naive Bayes
Support Vector Machine
F1 Score
Classification Models
F1 Score
Boosted Decision Tree
Decision Forest
Logistic Regression
Naive Bayes
Support Vector Machine