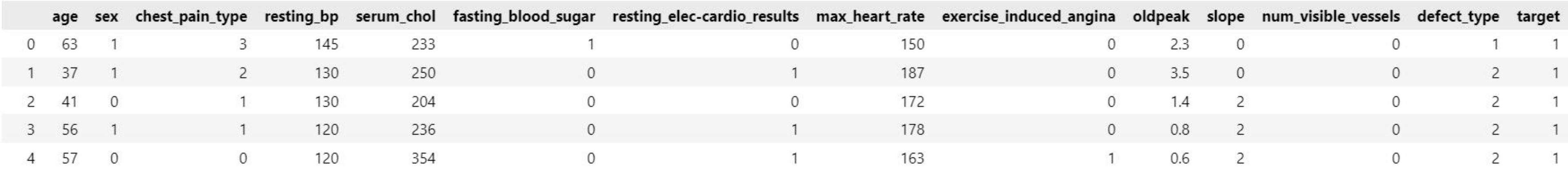
Random Forest

By Emily Atkinson, Vanessa Gleason, Anthony Rondos, and Eduard Stalmakov

Introduction

- Algorithm: Random Forest
- Data set: “Heart Disease Dataset”

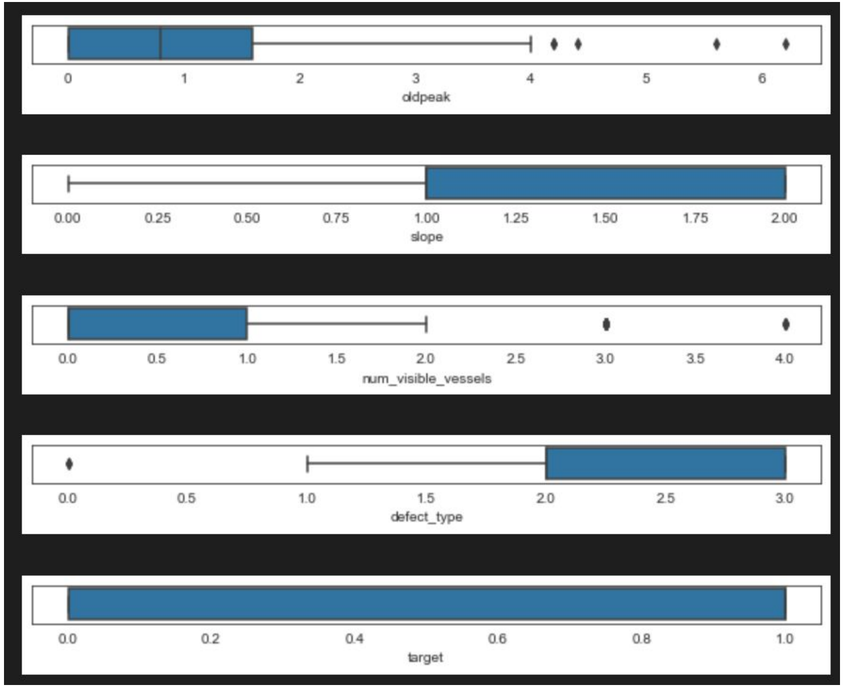


	age	sex	chest_pain_type	resting_bp	serum_chol	fasting_blood_sugar	resting_elec-cardio_results	max_heart_rate	exercise_induced_angina	oldpeak	slope	num_visible_vessels	defect_type	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

- Can we tune the algorithm to accurately predict whether or not someone has heart disease?
- Models used:
  - Random Forest
  - k Nearest-Neighbors

Data Processing

- Two requirements for cleaning:
  - Remove or impute missing values
  - Decide whether or not to drop outliers
- We didn’t have any missing values to remove
- We printed out box plots to decide whether or not to remove the outliers
- Based on the results, we decided to leave the outliers



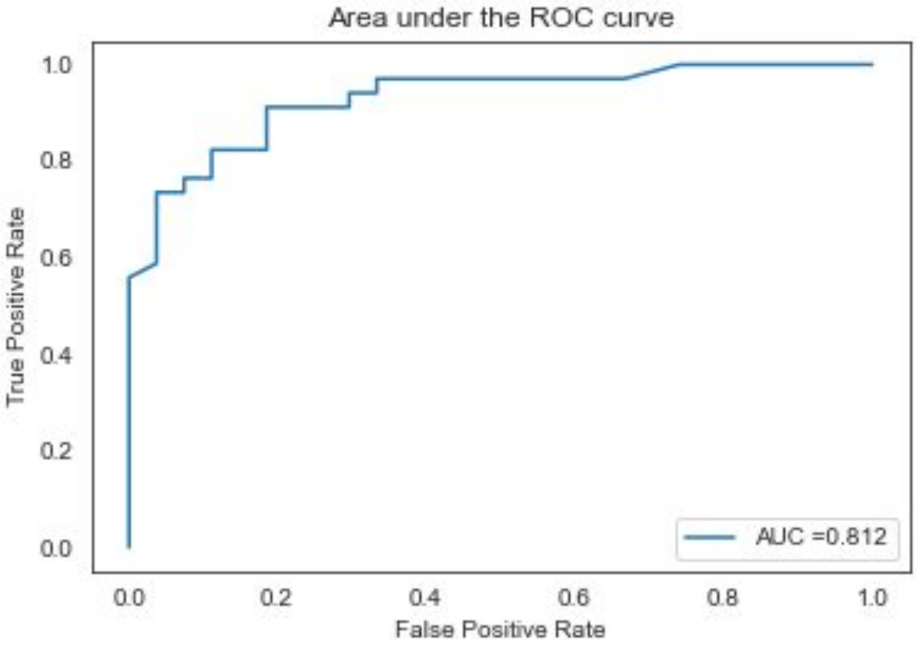
Train Test Split

- Our Y column is ‘Target,’ which indicates whether or not the individual has heart disease
- We ran with a 20% test group
- Initially, we started with the default parameters

Initial Results

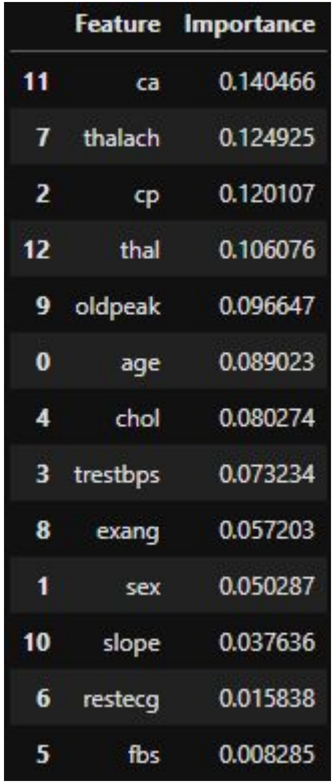
Average of 3 Trials:

- Model Accuracy = 84.2%
- MCC = .633
- AUC Score = .812



Feature Engineering

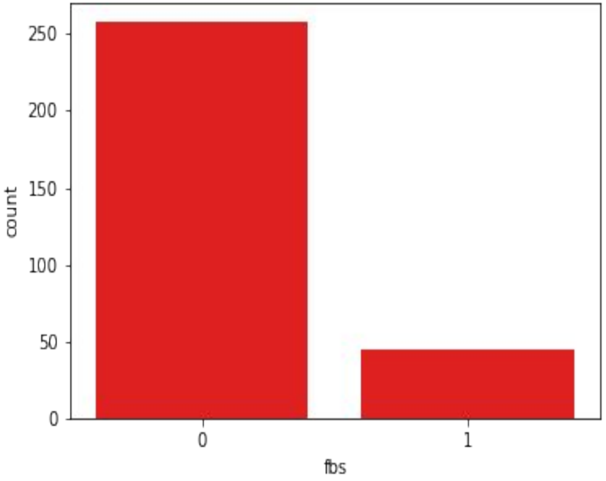
The Fasting Blood Sugar (fbs) column had a particularly low impact on the model.

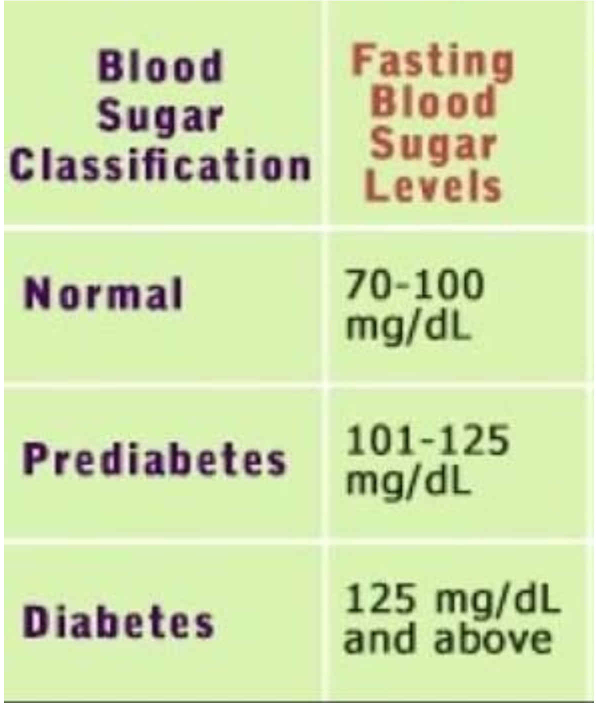


	Feature	Importance
11	ca	0.140466
7	thalach	0.124925
2	cp	0.120107
12	thal	0.106076
9	oldpeak	0.096647
0	age	0.089023
4	chol	0.080274
3	trestbps	0.073234
8	exang	0.057203
1	sex	0.050287
10	slope	0.037636
6	restecg	0.015838
5	fbs	0.008285

Fasting Blood Sugar

In the data set, Fasting Blood Sugar was *1 if > 120 mg/dl, 0 otherwise*.





Blood Sugar Classification	Fasting Blood Sugar Levels
Normal	70-100 mg/dL
Prediabetes	101-125 mg/dL
Diabetes	125 mg/dL and above

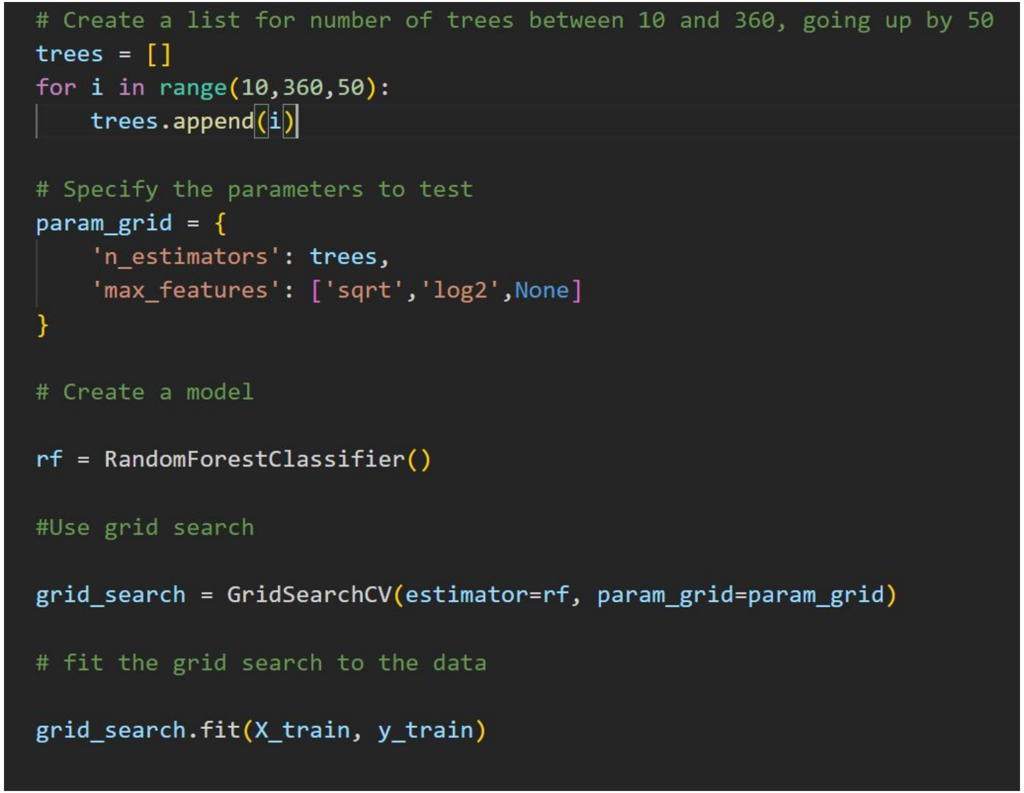
Fasting Blood Sugar Removal Results

Average Model Accuracy of 3 Trials:

-With fbs: 84.15%

-Without fbs: 84.7%

Hyperparameters Tuning



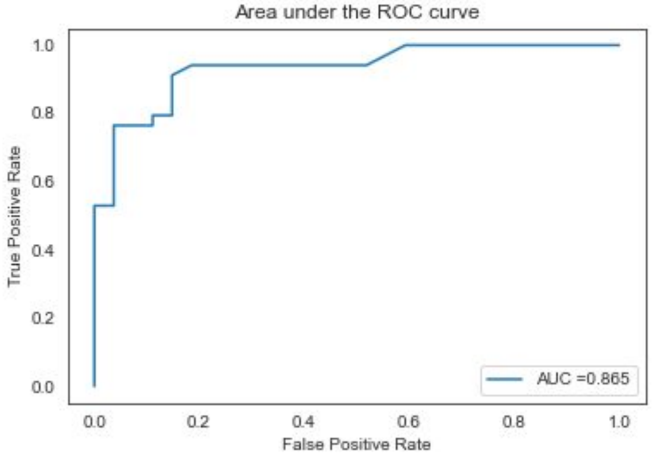
```
# Create a list for number of trees between 10 and 360, going up by 50
trees = []
for i in range(10,360,50):
    trees.append(i)

# Specify the parameters to test
param_grid = {
    'n_estimators': trees,
    'max_features': ['sqrt','log2',None]
}

# Create a model

rf = RandomForestClassifier()

#Use grid search

grid_search = GridSearchCV(estimator=rf, param_grid=param_grid)

# fit the grid search to the data

grid_search.fit(X_train, y_train)
```

- Optimal number of trees is 260
- The best max_feature is square root

Model Evaluation

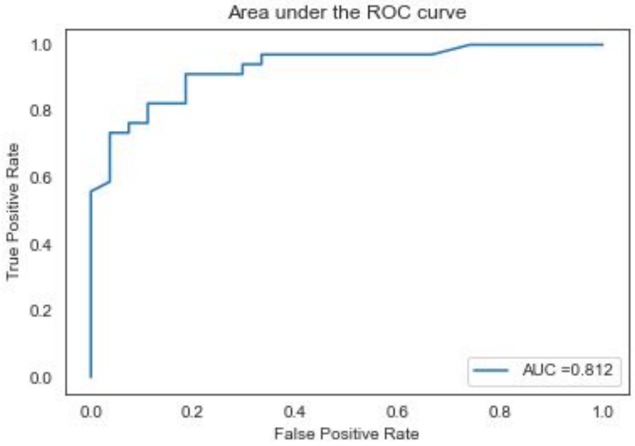
After Tuning and Feature Engineering

- Model Accuracy = 86.8%
- MCC = .734
- AUC Score = .865



Using Default Parameters

- Model Accuracy = 84.2%
- MCC = .633
- AUC Score = .812



Increased Performance Examples

To increase model accuracy, increase n_estimators

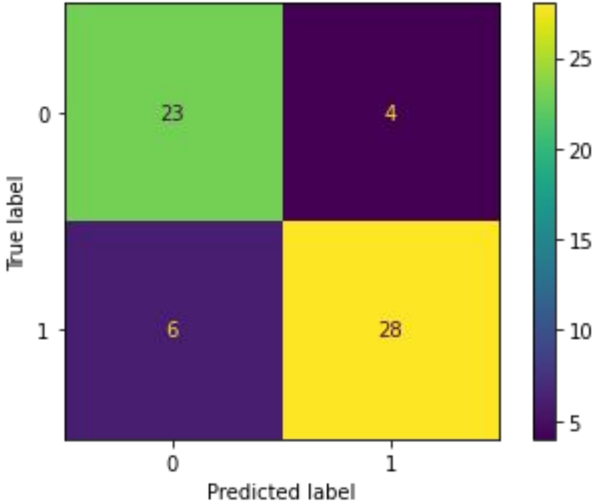
Average Model Accuracy of 3 Trials:

- 260 n_estimators: 86.89% model accuracy
  - 8.2% outcomes were false negatives
- 1000 n_estimators: 88.52% model accuracy
  - 6.5% outcomes were false negatives

Base Model Performance Assessment

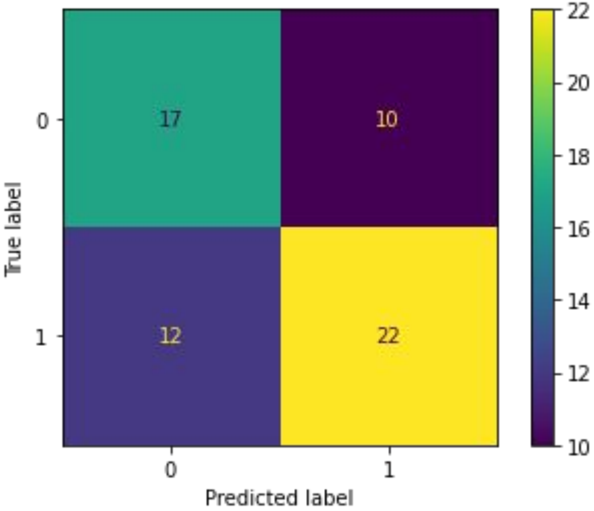
Random Forest ✓

- Model Accuracy = 84.2%
- MCC = .633
- AUC Score = .812



K-Nearest Neighbor (kNN) ✗

- Model Accuracy = 63.9%
- MCC = .275
- AUC Score = .638



Sources

Heart Disease Dataset

https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

Random Forest Presentation

https://github.com/lovelyleiva/Random-Forest-Research-Product