# Heart Disease Predictions

**GA Final Project - Justin Moss** 

## Objective

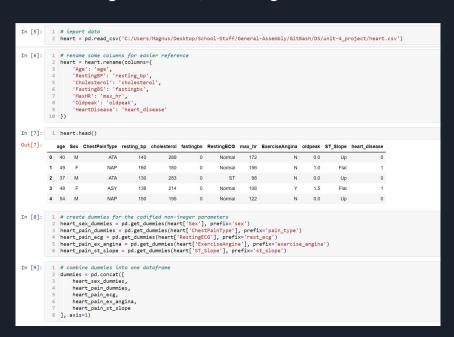
Based on the observed parameters, predict if a given set of characteristics puts a patient at risk for heart disease

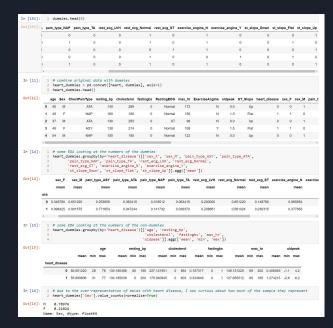
#### Data Dictionary

- Age: age of the patient [years]
- Sex: sex of the patient [M: Male, F: Female] / [0: Female, 1: Male]
- ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
- RestingBP: resting blood pressure [mm Hg]
- Cholesterol: serum cholesterol [mm/dl]
- FastingBS: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]
- Resting ECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
- MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
- ExerciseAngina: exercise-induced angina [Y: Yes, N: No]
- Oldpeak: oldpeak = ST [Numeric value measured in depression]
- ST\_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
- Heart Disease: output class [1: heart disease, 0: Normal]

## First Steps:

Loading in the data, renaming columns for ease of use, and some EDA.





#### **EDA Observations:**

- 55% of samples have heart\_disease
- Age Range is from 28 to 77, with no significant separation in average age
- 90% of heart disease cases are men despite making up only 79% of the samples
- No significant separation in average resting Resting BP
- Heart disease cases typically have lower cholesterol
- Heart disease cases typically have lower max heart rates
- 33% of heart disease cases have fasting blood sugar over 120 mg/dl
- 77% of heart disease cases have ASY pain type (asymptomatic ie: no pain)
- 56% of heart disease cases have normal resting ECG
- 62% of heart disease cases have Exercise Angina
- Higher oldpeak positively associated with higher risk of heart disease
- Flat ST Slope for 75% of heart disease cases

#### Moving forward:

Since we are predicting a binary value, we will use a Logistic Regression. We'll use all of the parameters as our features.

Fitting with all of the data gave an accuracy of ~87%

```
# for whatever reason, I had to increase max iterations to make the code work
    logreg = LogisticRegression(max_iter=1000)
    feature cols = ['age', 'resting bp', 'cholesterol', 'fastingbs', 'max hr', 'oldpeak',
                    'sex F', 'sex M', 'pain type ASY', 'pain type ATA', 'pain type NAP',
                    'pain_type_TA', 'rest_ecg_LVH', 'rest_ecg_Normal', 'rest_ecg_ST',
                    'exercise_angina_N', 'exercise_angina_Y', 'st_slope_Down', 'st_slope_Flat',
                    'st slope Up'
11
12
    X = heart dummies[feature cols]
    v = heart dummies['heart disease']
15
16 # First I'll fit all of the data and see what results we get
17 logreg.fit(X,v)
18 pred = logreg.predict(X)
19 logreg.score(X,y)*100
87.25490196078431
```

### More False Negatives or Positives?

As this is something as serious as a heart attack, I would rather a potential patient received a false positive and sought care instead of a false negative. There will always be a tradeoff there. (more on that later)

```
# Lets stack the predcitions and their probabilities against the actual data
    heart_dummies['predict_prob'] = logreg.predict_proba(X)[:, 1]
   heart_dummies['predict'] = logreg.predict(X)
 4 heart_dummies[['heart_disease', 'predict', 'predict_prob']].head(10)
   heart_disease predict predict_prob
                         0.839502
                         0.030736
                         0.011728
                         0.061855
 1 false_neg = heart_dummies[heart_dummies['heart_disease'] > heart_dummies['predict']]
 2 false_neg.shape
(48, 28)
 1 false_pos = heart_dummies[heart_dummies['heart_disease'] < heart_dummies['predict']]</pre>
2 false_pos.shape
(69, 28)
1 f'Percent false readings: {((false_neg.shape[0]+false_pos.shape[0])/heart_dummies.shape[0])*100} %
'Percent false readings: 12.745098039215685 %'
```

#### Train-Test-Split?

Our score falls slightly with a train-test-split, which of course means we're more generalized now, but we can do better.

```
# Going with a test-train-split:
logreg1 = LogisticRegression(max_iter=1000)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)

logreg1.fit(X_train, y_train);

# scoring the train-test-split model
y_pred = logreg1.predict(X_test)
logreg1.score(X_test, y_test)

0.8652173913043478
```

#### Confusion Matrixing, Altering Threshold

Using a confusion matrix, we will change the threshold for a positive indication of heart disease to '.3' for the probability. We now have significantly fewer false scores.

```
# Build a confusion matrix to send more potential false negative to coming back as positive
logit_simple = linear_model.LogisticRegression(max_iter=1000, C=1e9).fit(X_train, y_train)
logit_pred_proba = logit_simple.predict_proba(X_test)[:,1]
tn, fp, fn, tp = metrics.confusion_matrix(y_true=y_test, y_pred=logit_pred_proba > .3).ravel()
(tn, fp, fn, tp)

# percent of false readings
(fp+fn)/len(y_test)*100

12.173913043478262

# percent of 'true' readings
(tn+tp)/len(y_test)*100

87.82608695652175

87.82608695652175
```

#### Am I at risk?

I made some assumptions as I've never heard of oldpeak or ST Slope before this project, but I should be fine for now.

```
1 me = [[31, 80, 160, 0, 190, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1]]
2 print(logit_simple.predict(me))
3 print(logit_simple.predict_proba(me))
[0]
[[0.75145737 0.24854263]]
```

#### Conclusions:

Based on our findings with the confusion matrix possibly creating more false positives, I am personally not presently at risk for heart disease.

In the future, we could expand the number of observations and take in more health data such as:

- Is the patient a smoker?
- Do they exercise regularly?
- Body-mass Index
- etc.

Additionally, I could clean up the dummies to bring it closer to a binary. In particular:

- Sex could be 0=male 1=female
- Pain types could have three columns for the actual pains, with asymptomatic being all 0s
- same with resting ecg, st slope, exercise angina

## Questions?

Data sourced from:

https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction