# Midpoint Report: Heart Disease G2

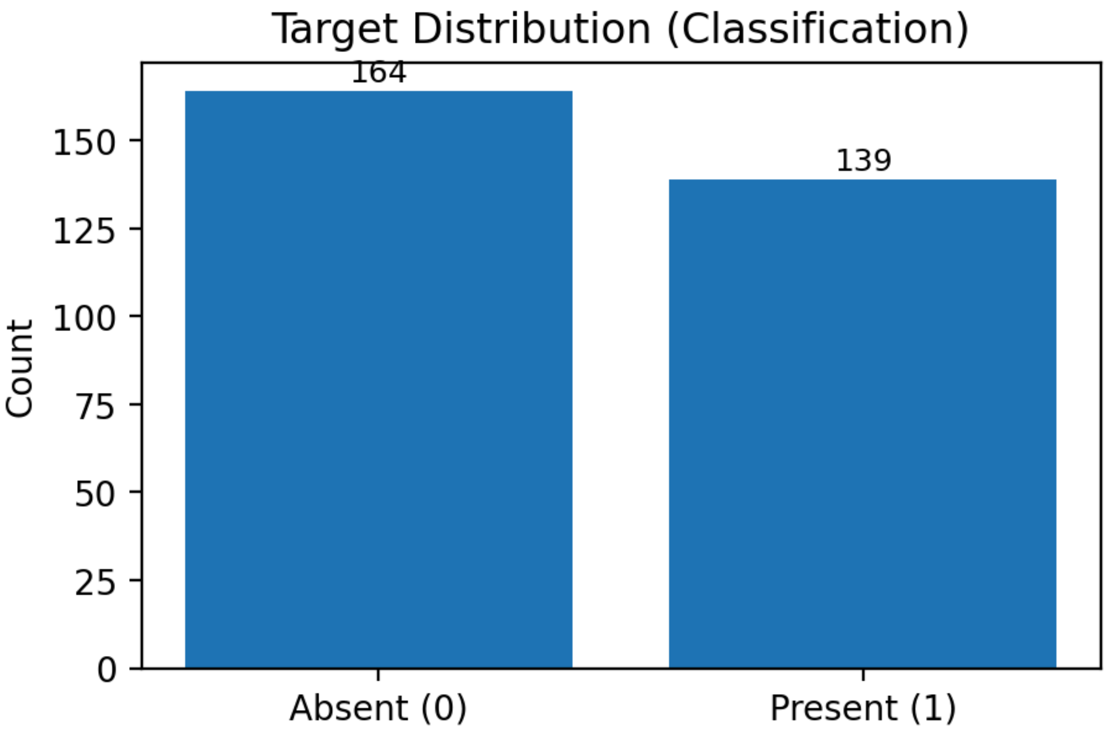
**Github Link:** https://github.com/MBELSABAH/Cs-4120-Project.git

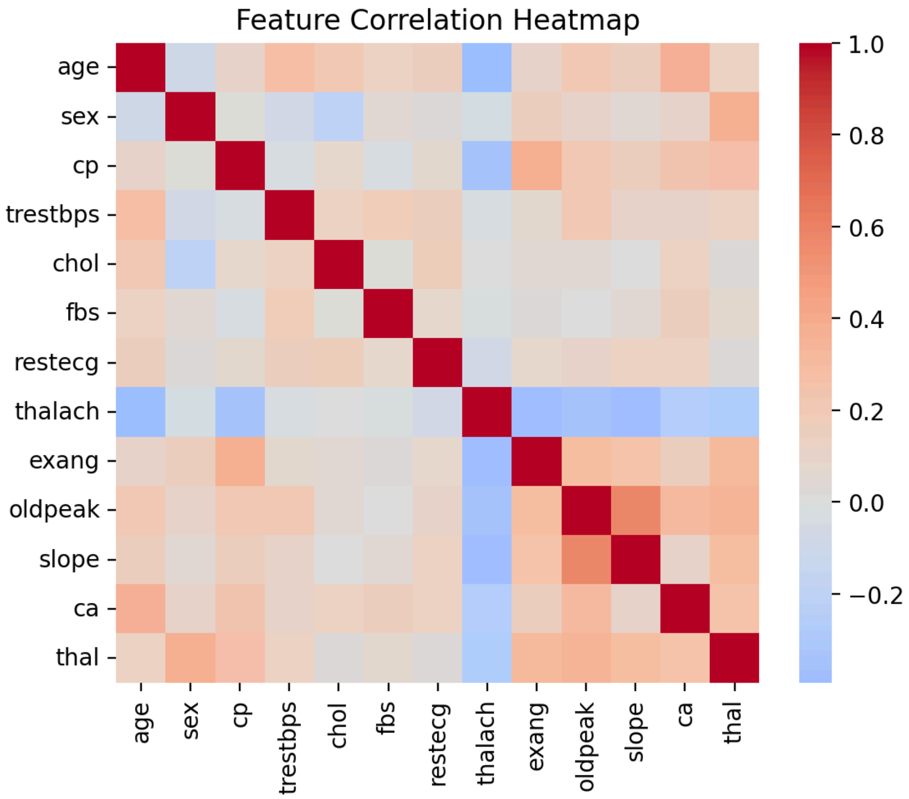
**Updated Dataset Description & Cleaning:**

- UCI Cleveland Heart Disease (303 rows, 14 attributes).
- target = 1 if disease present (num>0), else 0; regression target = thalach (bpm).
- ca (4 rows) & thal (2 rows) imputed by mode; standardization & one-hot via sklearn Pipeline.

**Exploratory Data Analysis (EDA):**

- Slight class imbalance (~55% positive).
- thalach is inversely related to age; the old peak tends to increase with exang.





**Split & Baselines:**

- 60/20/20 train/val/test split, random_state = 42; stratified for classification.
- **Classification baselines:** Logistic Regression, Decision Tree (MLflow tracked).
- **Regression baselines:** Linear Regression, Decision Tree Regressor (MLflow tracked).

**Plots:** 1) target distribution 2) correlation heatmap
3) confusion matrix of the best classification baseline on TEST
4) residuals vs predicted of best regression baseline on TEST

## Results and discussion

We selected the best models for each task based solely on their highest validation performance. For classification, we selected the model with the highest F1 on the validation set; for regression, we selected the model with the lowest validation RMSE. Afterwards, we retrained the models on train & validation together and evaluated it once on the Test set. We are avoiding tuning directly on the test data to not skew the accuracy %.

The heart-disease label is slightly unbalanced (around 55% positive). Accuracy is decent if one class is neglected. F1 averages well across both classes, so it negatively affects any model that performs well on one class but poorly on the others. This is beneficial for medical-style problems where booleans are crucial.

**Table 1 - Classification Metrics**

| Model | Val Accuracy | Val F1 | Test Accuracy | Test F1 |
|---|---|---|---|---|
| Logistic Regression | 0.819672131147541 | 0.817910447761194 | 0.8688524590163930 | 0.8688172043010750 |
| Decision Tree Classifier | 0.7213114754098360 | 0.7213114754098360 | 0.7213114754098360 | 0.7210115684691960 |

**Interpretation:** Logistic regression performs better and is above the majority baseline of 55%, meaning it learned a signal, not class proportions. With standardized numerical features and one-hot categorical features, a linear decision boundary fits this dataset very well. The Confusion Matrix shows balanced true positives/negatives; therefore, any errors are borderline cases. The untuned decision tree underperforms due to the small dataset, and any depth/leaf choices have sensitive repercussions. The label is approximately 55% positive, so F1 macro is safer than accuracy, as we do not want to ignore any classes.

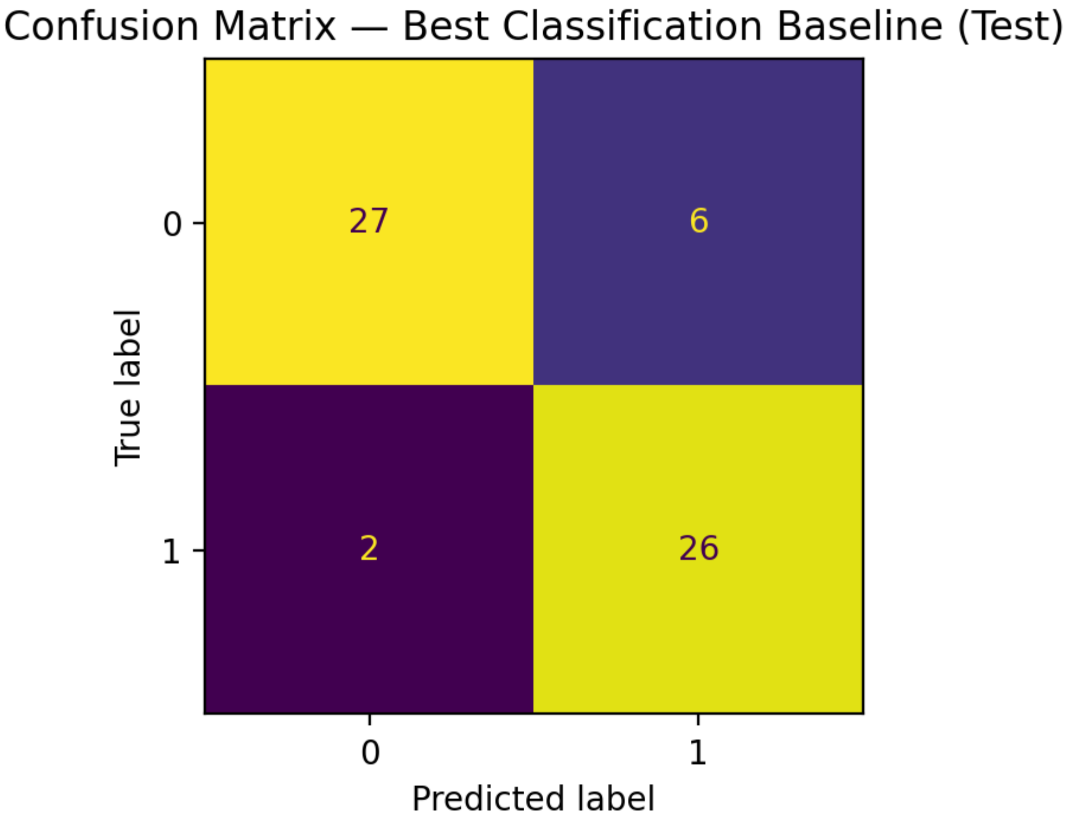
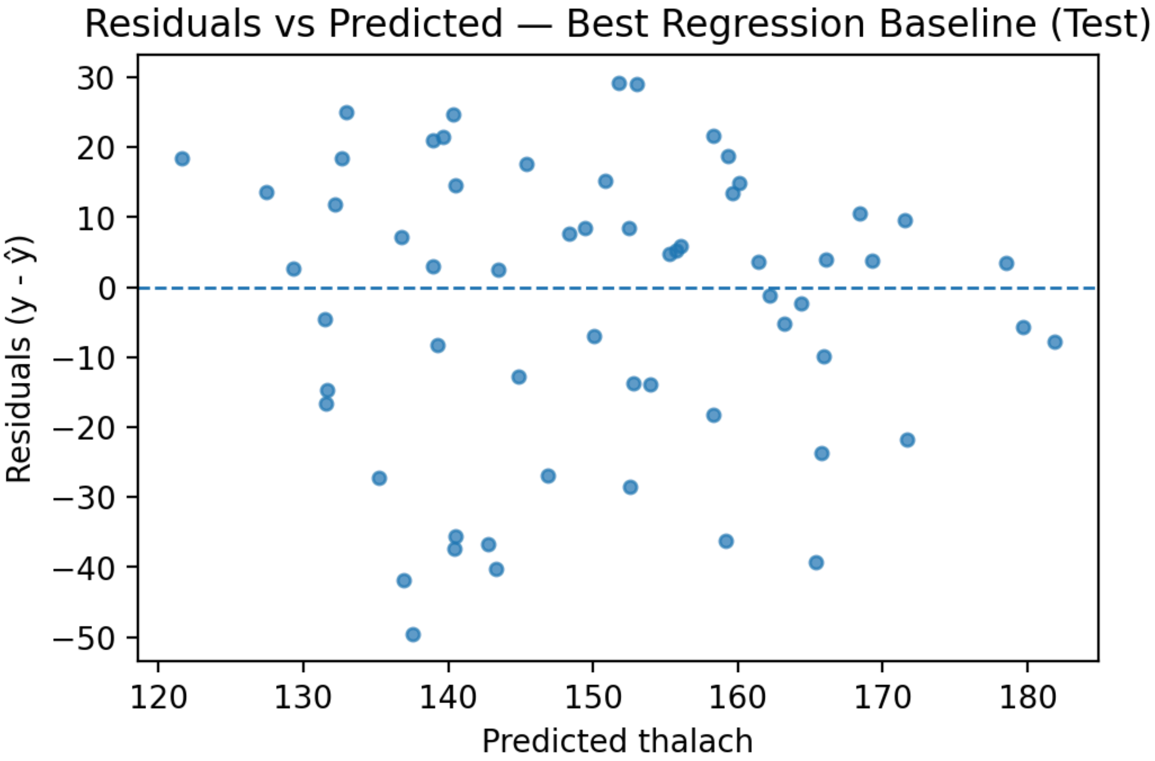


**Table 2 - Regression Metrics**

| Model | Val MAE | Val RMSE | Test MAE | Test RMSE |
|---|---|---|---|---|
| Linear Regression | 12.551189588622600 | 15.468049095241700 | 16.563649350332500 | 20.213762190259800 |
| DecisionTree Regressor | 16.37704918032790 | 22.01154689523910 | 21.59016393442620 | 27.1869665012664 |

Linear regression performed better because dominant trends are somewhat linear (thalach is inversely proportional to age). The residuals vs Predicted plot is mostly centered around zero with mild structure, showing that a few simple interactions could reduce errors. Trees struggle here because of the small sample size, making depth/leaf choices very sensitive and prone to over/under fitting.



The Exploratory Data Analysis and boxplots showed a slight class imbalance, that thalach is inversely related to age, and oldpeak tends to be higher when exang=1. We believe this works in favor of Logistic Regression due to these somewhat linear trends. These small dataset sizes limit model complexity, and categorical levels like thal/ca can be very rare, and any feature interactions are not yet incorporated into the baseline.

**Neural Network Plan**

We will use compact multi-layer perceptrons (MLPs) for tabular data on both tasks. Preprocessing matches the baselines by doing the following: using standardized numeric features, one-hot encode categoricals, keeping the same 60/20/20 split with a fixed seed, and using the same non-linear interactions without overfitting the 303-row dataset. The classification MLP uses hidden layers 64,32,16 with ReLU, BatchNorm, and an around 0.2 dropout, ending in a sigmoid. The regression MLP uses 64,32 with the same setup. We plan to train with Adam, batch 64, early stopping, and a reduce-on-plateau scheduler. We will select the best models based on validation performance and retrain them on Train+Val, and then report the results on the Test set.