Model Comparison

Trever Yoder

Task 1: Conceptual Questions

Question 1:

- **What is the purpose of using cross-validation when fitting a random forest model?**

  Cross-validation is used to estimate how well the random forest will perform on unseen data by repeatedly splitting the data into "folds," training on some folds and validating on the other(s). This helps guard against overfitting and provides a more reliable measure of out-of-sample predictive accuracy.

Question 2:

- **Describe the bagged tree algorithm.**

  Bagged trees build many decision trees on different bootstrap samples of the training data and then average their predictions. By aggregating across models trained on varied subsets, bagging reduces variance and improves stability compared to a single decision tree.

Question 3:

- **What is meant by a general linear model?**

  A general linear model expresses a continuous response variable as a linear combination of predictors where the errors are assumed to be normally distributed with constant variance. It encompasses methods like ANOVA, ANCOVA, and multiple linear regression.

Question 4:

- **When fitting a multiple linear regression model, what does adding an interaction term do? That is, what does it allow the model to do differently as compared to when it is not included in the model?**

  An interaction term allows the effect of one predictor on the response to change depending on the level of the other predictor. Without interactions the model assumes additive effects, but with interactions, it can capture non-additive relationships, where the slope for one variable varies by the value of another.

Question 5:

- **Why do we split our data into a training and test set?**

  Splitting data ensures that we can train the model on one subset (training set) and then evaluate its performance on completely unseen data (test set). This provides an unbiased assessment of how well the model generalizes to new observations and helps detect overfitting.

Task 2: Data Prep

Packages and Data

```
library(tidyverse)
library(tidymodels)
library(caret)
library(yardstick)
library(glmnet)

#Read in data and print summary
Heart_data <- read_csv(
  "heart.csv")
summary(Heart_data)
```

```
      Age            Sex            ChestPainType        RestingBP
 Min.   :28.00   Length:918         Length:918         Min.   :  0.0
 1st Qu.:47.00   Class :character   Class :character   1st Qu.:120.0
 Median :54.00   Mode  :character   Mode  :character   Median :130.0
 Mean   :53.51                                         Mean   :132.4
 3rd Qu.:60.00                                         3rd Qu.:140.0
```

```
 Max.   :77.00                                          Max.   :200.0
  Cholesterol      FastingBS       RestingECG            MaxHR
 Min.   :  0.0   Min.   :0.0000   Length:918         Min.   : 60.0
 1st Qu.:173.2   1st Qu.:0.0000   Class :character   1st Qu.:120.0
 Median :223.0   Median :0.0000   Mode  :character   Median :138.0
 Mean   :198.8   Mean   :0.2331                      Mean   :136.8
 3rd Qu.:267.0   3rd Qu.:0.0000                      3rd Qu.:156.0
 Max.   :603.0   Max.   :1.0000                      Max.   :202.0
 ExerciseAngina        Oldpeak          ST_Slope          HeartDisease
 Length:918         Min.   :-2.6000   Length:918         Min.   :0.0000
 Class :character   1st Qu.: 0.0000   Class :character   1st Qu.:0.0000
 Mode  :character   Median : 0.6000   Mode  :character   Median :1.0000
                    Mean   : 0.8874                      Mean   :0.5534
                    3rd Qu.: 1.5000                      3rd Qu.:1.0000
                    Max.   : 6.2000                      Max.   :1.0000
```

Question 1: Variable Type

Heart Disease is treated as a quantitative variable which does not make sense. Conceptually, Heart Disease is a categorical variable since 0 means no heart disease and 1 means presence of heart disease. A decimal like 0.5 wouldn't fall into either category. A person either has or does not have heart disease (1 or 0).

Question 2: Create New Data

```
#Create new heart data
new_heart <- Heart_data %>%
  mutate(Heart_Disease = factor(HeartDisease, levels = c(0, 1), labels = c("No", "Yes"))) %>%
  select(-ST_Slope, -HeartDisease)
```

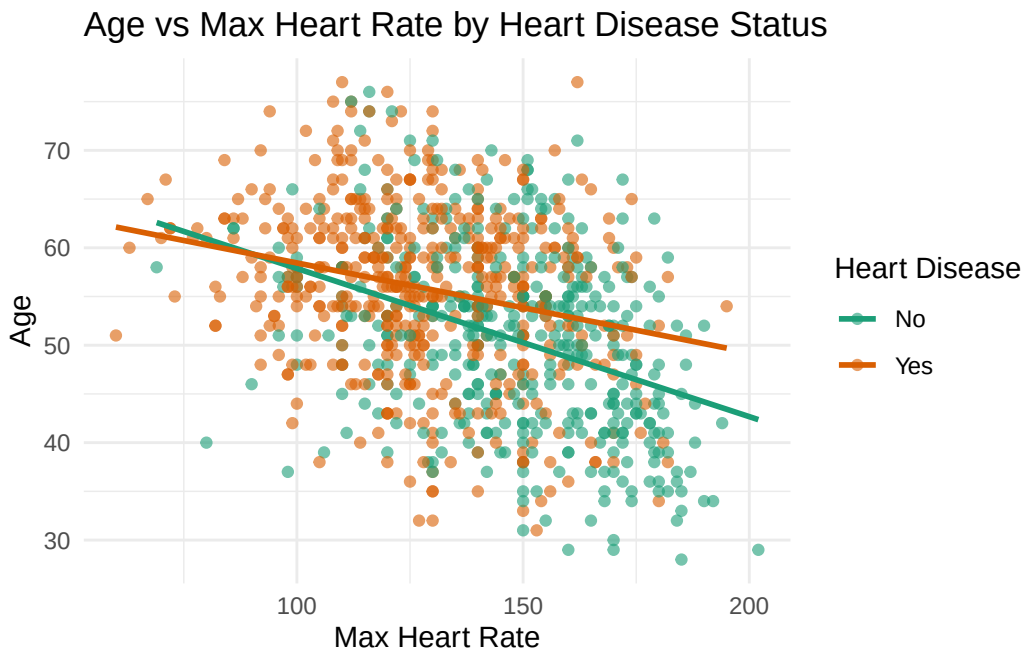
Task 3: EDA

Question 1: Plot to check for interaction

```
#plot age vs max heart rate by disease status
ggplot(new_heart, aes(x = MaxHR, y = Age, color = Heart_Disease)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE) +
```

```
  scale_color_brewer(palette = "Dark2") +
  labs(
    title = "Age vs Max Heart Rate by Heart Disease Status",
    x = "Max Heart Rate",
    y = "Age",
    color = "Heart Disease"
  ) +
  theme_minimal()
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Question 2: Conclusion based on visual evidence

We can see that the slopes are different. An additive model would assume their slopes to be the same. Therefore, to account for different slopes, we need to use an interaction model.

Task 4: Testing and Training

```r
#set seed for repeat ability and split data into train and test data
set.seed(101)
split <- initial_split(new_heart, prop = 0.8)
train <- training(split)
test <- testing(split)
```

Task 5: OLS and LASSO

Question 1: Interaction model

```r
#specify model
ols_mlr <- lm(Age ~ MaxHR * Heart_Disease, data = train)
summary(ols_mlr)
```

```
Call:
lm(formula = Age ~ MaxHR * Heart_Disease, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-22.7703  -5.7966   0.4516   5.7772  20.6378

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)            75.58896    3.07510  24.581  < 2e-16 ***
MaxHR                  -0.16992    0.02064  -8.233 8.43e-16 ***
Heart_DiseaseYes       -8.58502    3.83433  -2.239  0.02546 *
MaxHR:Heart_DiseaseYes  0.08343    0.02716   3.072  0.00221 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.478 on 730 degrees of freedom
Multiple R-squared:  0.1839,    Adjusted R-squared:  0.1806
F-statistic: 54.84 on 3 and 730 DF,  p-value: < 2.2e-16
```

Question 2: RMSE

```r
#calculate RMSE
predictions <- predict(ols_mlr, newdata = test)
residuals <- test$Age - predictions
OLS_rmse <- sqrt(mean(residuals^2))
OLS_rmse
```

```
[1] 9.100206
```

Question 3: Performance

```r
#Create Recipe
LASSO_recipe <- recipe(Age ~ MaxHR + Heart_Disease, data = train) %>%
  step_dummy(Heart_Disease) %>%
  step_normalize(all_predictors()) %>%
  step_interact( ~ MaxHR:starts_with("Heart_Disease_") )
LASSO_recipe
```

```
-- Recipe ----------------------------------------------------------------------


-- Inputs

Number of variables by role

outcome:   1
predictor: 2


-- Operations

* Dummy variables from: Heart_Disease

* Centering and scaling for: all_predictors()

* Interactions with: MaxHR:starts_with("Heart_Disease_")
```

Question 4: Spec, Grid, Results

```
#Create specs and folds
LASSO_spec <- linear_reg(penalty = tune(), mixture = 1) %>%
  set_engine("glmnet")
cv_folds <- vfold_cv(train, v = 10)

#Create workflow
LASSO_wf <- workflow() %>%
  add_model(LASSO_spec) %>%
  add_recipe(LASSO_recipe)

#create grid
lasso_grid <- grid_regular(penalty(), levels = 200)

#tune
LASSO_results <- tune_grid(
  LASSO_wf,
  resamples = cv_folds,
  grid = lasso_grid,
  metrics = metric_set(rmse)
)

#print best lasso
best_LASSO <- select_best(LASSO_results, metric = "rmse")
best_LASSO
```

```
# A tibble: 1 x 2
       penalty .config
         <dbl> <chr>
1 0.0000000001 Preprocessor1_Model001
```

```
#setup and report results from final model
final_lasso_wf <- finalize_workflow(LASSO_wf, best_LASSO)
final_lasso_fit <- fit(final_lasso_wf, data = train)
tidy(final_lasso_fit)
```

```
# A tibble: 4 x 3
  term                      estimate      penalty
  <chr>                        <dbl>        <dbl>
```

```
1 (Intercept)                 54.0  0.0000000001
2 MaxHR                       -3.08 0.0000000001
3 Heart_Disease_Yes            1.36 0.0000000001
4 MaxHR_x_Heart_Disease_Yes    1.03 0.0000000001
```

Question 5: RMSE

I would expect the RMSE to be very similar since the LASSO applied very little penalty (1e-10) and the same variables and interaction were used. The scale is different (ie our intercept is only 54) since we normalized the variables, but that won't change the RMSE.

```
# Get LASSO predictions
lasso_preds <- predict(final_lasso_fit, new_data = test) %>%
  bind_cols(test)

# Compute RMSE
rmse_lasso <- rmse(lasso_preds, truth = Age, estimate = .pred)
rmse_lasso
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard        9.10
```

Question 6: Compare RMSE

We can see that both models have similar RMSE.

```
rmse_lasso
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard        9.10
```

```
OLS_rmse
```

```
[1] 9.100206
```

Question 7: Why are they same even though we have different coefficient?

As stated, we standardized our variables in the LASSO model, which makes the coefficients look different. However, we use the same variables and interactions, so they are making very similar predictions. Thus, our RMSE is nearly the same.

Task 6: Logistic Regression

Question 1: Fit 2 models and Identify Best models

```
#specify control
ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3, classProbs = TRUE, sum

# Model 1
logit_model1 <- train(
  Heart_Disease ~ MaxHR + Age + Sex + Cholesterol,
  data = train,
  method = "glm",
  family = "binomial",
  trControl = ctrl,
  metric = "ROC"
)

# Model 2
logit_model2 <- train(
  Heart_Disease ~ Age + MaxHR + Cholesterol + RestingBP + Sex + ChestPainType,
  data = train,
  method = "glm",
  family = "binomial",
  trControl = ctrl,
  metric = "ROC"
)

#ID best model
logit_model1$results
```

```
  parameter       ROC      Sens      Spec      ROCSD    SensSD     SpecSD
1      none 0.7712107 0.5982191 0.7950832 0.06730173 0.1019016 0.07298769
```

```
logit_model2$results
```

```
  parameter       ROC      Sens      Spec      ROCSD     SensSD     SpecSD
1      none 0.8484478 0.7205981 0.8358691 0.05724802 0.09621062 0.06046176
```

```
#provide summary of best model
summary(logit_model2$finalModel)
```

```
Call:
NULL

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -0.403073   1.182132  -0.341 0.733125
Age               0.038505   0.011482   3.354 0.000798 ***
MaxHR            -0.014720   0.004465  -3.297 0.000977 ***
Cholesterol      -0.002843   0.000937  -3.034 0.002416 **
RestingBP         0.008601   0.005329   1.614 0.106532
SexM              1.346203   0.251134   5.360 8.30e-08 ***
ChestPainTypeATA -2.562901   0.288670  -8.878  < 2e-16 ***
ChestPainTypeNAP -1.901788   0.230068  -8.266  < 2e-16 ***
ChestPainTypeTA  -1.621879   0.398296  -4.072 4.66e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1003.32  on 733  degrees of freedom
Residual deviance:  677.45  on 725  degrees of freedom
AIC: 695.45

Number of Fisher Scoring iterations: 5
```

Model 2 is our best model with an ROC of .848 (vs .771). The sensitivy and specificity standard deviation is also lower among folds for model 2, which indicates consistency. Model 2 also has higher sensitivity and specificity, which I will define in question 3.

Question 2: Check on Test data with confusion matrix

```
#Use trained model on test data
logit_preds <- predict(logit_model2, newdata = test)
conf_matrix <- confusionMatrix(logit_preds, test$Heart_Disease)
conf_matrix
```

```
Confusion Matrix and Statistics

          Reference
Prediction No Yes
       No  72  20
       Yes 22  70

               Accuracy : 0.7717
                 95% CI : (0.7042, 0.8303)
    No Information Rate : 0.5109
    P-Value [Acc > NIR] : 2.899e-13

                  Kappa : 0.5435

 Mcnemar's Test P-Value : 0.8774

            Sensitivity : 0.7660
            Specificity : 0.7778
         Pos Pred Value : 0.7826
         Neg Pred Value : 0.7609
             Prevalence : 0.5109
         Detection Rate : 0.3913
   Detection Prevalence : 0.5000
      Balanced Accuracy : 0.7719

       'Positive' Class : No
```

Question 3: ID Values of Sensitivity etc.

```
conf_matrix$byClass["Sensitivity"]
```

```
Sensitivity
  0.7659574
```

```
conf_matrix$byClass["Specificity"]
```

```
Specificity
  0.7777778
```

Sensitivity: Proportion of people with heart disease that our model correctly identifies

- Our Model can correctly identify 76.6% of people with heart disease (23.4% false negative)

Specificity: proportion of people without heart disease that our model correctly identifies

- Our model can correctly identify 77.8% of people without heart disease (22.2% false positive)