Model-Comparison

Task 1: Conceptual Questions

Question 1

Along with ensuring the robustness of the final model, cross-validation is used to select the best combination of hyperparameters for a random forest model. There are many hyperparameters that need to be tuned for a random forest model, including the number of trees, maximum depth, and the number of features that are randomly selected at each split.

Question 2

A bagged (bootstrap aggregated) tree model uses the concept of bootstrap sampling and aggregation to improve prediction over a single classification / regression tree by reducing variance. Bootstrap sampling uses random sampling with replacement to mimic the wider population. These are then aggregated as an ensemble model to produce a final fit that has a lower variance compared to a single tree fit.

Question 3

A general linear model is a generalization of linear regression. It consists of three components: a distribution for modeling the outcome variable, a linear predictor, and a link function that "links" the response variable to the linear function of the parameters.

Question 4

Adding an interaction term to a MLR model allows the model to capture the effect of two predictors together, rather than just their individual additive effects.
This allows the effect of one predictor to depend on the value of another.

Question 5

We split our data into a training and test set to assess model robustness and generalization. This allows the model to be evaluated on unseen data using relevant performance metrics.

Task 2: Data Prep

packages and data

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

heart <- read_csv("data/heart.csv")
```

Question 1

```r
summary(heart)
```

```
      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
```

a

HeartDisease is a quantitative variable

b

This does not make much sense, as one can only either have heart disease or not have it.

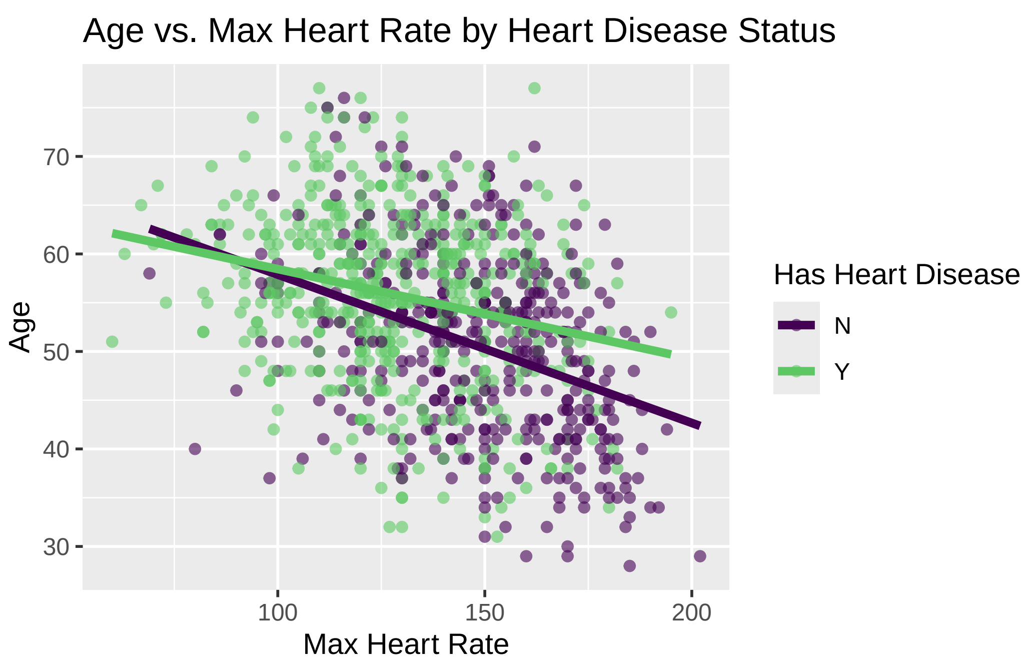
Question 2

```r
# change HeartDisease to factor with levels Y and N and remove unneeded cols
new_heart <- heart %>%
  mutate(HasHeartDisease = factor(HeartDisease, labels = c("N", "Y"))) %>%
  select(-HeartDisease, -ST_Slope)
```

Task 3: EDA

Question 1

```r
# plot age vs max heart rate by HeartDisease status and include fit lines
ggplot(new_heart, aes(x = MaxHR, y = Age, color = HasHeartDisease)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE, size = 1.5) +
  scale_color_viridis_d(end = 0.75) +
  labs(title = "Age vs. Max Heart Rate by Heart Disease Status",
       x = "Max Heart Rate",
       y = "Age",
       color = "Has Heart Disease")
```



Question 2

Based on the plot above, an interaction model is more appropriate. This is because the effect of Max Heart Rate on Age appears to depend on Heart Disease status, as can be seen by the differently sloped lines. As the relationship is not consistent across groups, an interaction term is likely to produce a better model.

Task 4: Testing and Training

Question 1

```
set.seed(101)

# split data into train and test sets
data_split <- initial_split(new_heart, prop = 0.8)
train <- training(data_split)
test <- testing(data_split)
```

Task 5: OLS and LASSO

Question 1

```r
# define ols spec
ols_spec <- linear_reg() %>%
  set_engine("lm")

# define ols model (with interaction) and transform HasHeartDisease to a dummy variable
ols_recipe <- recipe(Age ~ MaxHR + HasHeartDisease, data = train) %>%
  step_dummy(HasHeartDisease) %>%
  step_interact(terms = ~ MaxHR:starts_with("HasHeartDisease"))

# setup workflow
ols_workflow <- workflow() %>%
  add_model(ols_spec) %>%
  add_recipe(ols_recipe)

# fit
ols_mlr <- fit(ols_workflow, data = train)

# produce summary output
extract_fit_engine(ols_mlr) %>% summary()
```

```
Call:
stats::lm(formula = ..y ~ ., data = data)

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 ***
HasHeartDisease_Y         -8.58502    3.83433  -2.239  0.02546 *
MaxHR_x_HasHeartDisease_Y  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

```r
# predict using ols model and pull rmse
ols_final_rmse <- ols_mlr %>%
  predict(test) %>%
  pull() %>%
  rmse_vec(truth = test$Age) %>%
  tibble(rmse = .)

ols_final_rmse
```

```
# A tibble: 1 x 1
   rmse
  <dbl>
1  9.10
```

Question 3

```r
# define LASSO model (with interaction) and transform HasHeartDisease to a dummy variable
LASSO_recipe <- recipe(Age ~ MaxHR + HasHeartDisease, data = train) %>%
  step_dummy(HasHeartDisease) %>%
  step_normalize(all_numeric(), -all_outcomes()) %>%
  step_interact(~ MaxHR:starts_with("HasHeartDisease_"))

LASSO_recipe
```

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



-- Inputs
```

```
Number of variables by role

outcome:   1
predictor: 2



-- Operations

* Dummy variables from: HasHeartDisease

* Centering and scaling for: all_numeric() -all_outcomes()

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

Question 4

```r
# define spec and tune LASSO penalty hyperparam
LASSO_spec <- linear_reg(penalty = tune(), mixture = 1) %>%
  set_engine("glmnet")

# define workflow
LASSO_workflow <- workflow() %>%
  add_recipe(LASSO_recipe) %>%
  add_model(LASSO_spec)

# define 10-fold cv
LASSO_cv_folds <- vfold_cv(train, 10)

# setup LASSO grid with 200 levels of penalty hyperparam
LASSO_grid <- LASSO_workflow %>%
  tune_grid(resamples = LASSO_cv_folds,
            grid = grid_regular(penalty(), levels = 200))

# select best model according to rmse
lowest_rmse <- LASSO_grid %>%
  select_best(metric = "rmse")

# finalize model with tuned hyperparam and fit on train set
LASSO_final <- LASSO_workflow %>%
  finalize_workflow(lowest_rmse) %>%
  fit(train)

tidy(LASSO_final)
```

```
# A tibble: 4 x 3
  term                        estimate penalty
  <chr>                          <dbl>   <dbl>
1 (Intercept)                    54.0   0.0392
2 MaxHR                          -3.07  0.0392
3 HasHeartDisease_Y               1.35  0.0392
4 MaxHR_x_HasHeartDisease_Y       1.01  0.0392
```

Question 5

I would expect the OLS and LASSO models to have roughly the same RMSE. Since none of the estimates are close to zero, the LASSO model likely did not shrink any terms by a large amount. Since there's only a few predictors and an interaction term, the OLS model probably captures most of the variance without overfitting. LASSO regularization prevents overfitting by shrinking unnecessary predictors, but with such a small and well-specified model, this isn't likely to make a huge difference.

Question 6

```
# use test set to evaluate rmse and show LASSO rmse
LASSO_workflow %>%
  finalize_workflow(lowest_rmse) %>%
  last_fit(data_split) %>%
  collect_metrics() %>%
  filter(.metric == "rmse") %>%
  select(rmse = .estimate)
```

```
# A tibble: 1 x 1
   rmse
  <dbl>
1  9.09
```

```
# show ols rmse
ols_final_rmse
```

```
# A tibble: 1 x 1
   rmse
  <dbl>
1  9.10
```

Question 7

The RMSE values are roughly the same since both models capture the same underlying relationship between the predictors and the response. Even though LASSO shrinks the coefficients by a bit, it doesn't change the predictive behavior of the overall model too much. Since the important predictors are still there in both models, their predictions on test data are very similar, leading to a similar RMSE (even with different coefficient estimates).

Task 6: Logistic Regression

Question 1

```r
# define train control using repeated cv
control <- trainControl(method = "repeatedcv",
                        number = 10,
                        repeats = 5)

# LR_1 is the first logistic regression model (simpler)
LR_1 <- train(HasHeartDisease ~ Age + Sex + ChestPainType + RestingBP + RestingECG,
              data = train,
              method = "glm",
              family = "binomial",
              trControl = control)

# LR_2 is the second logistic regression model (two additional predictors)
LR_2 <- train(HasHeartDisease ~ Age + Sex + ChestPainType + RestingBP + RestingECG
              + MaxHR + ExerciseAngina,
              data = train,
              method = "glm",
              family = "binomial",
              trControl = control)

# show results for LR_1 and LR_2
LR_1$results
```

```
  parameter  Accuracy     Kappa AccuracySD    KappaSD
1      none 0.7815548 0.5514582 0.04335092 0.08939872
```

```
LR_2$results
```

```
  parameter  Accuracy     Kappa AccuracySD    KappaSD
1      none 0.7913208 0.5712624 0.04501077 0.09168508
```

```
# show the summary for the final model (LR_2)
summary(LR_2$finalModel)
```

```

Call:
NULL

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)    
(Intercept)      -1.280424   1.333515  -0.960   0.3370    
Age               0.038614   0.012303   3.139   0.0017 ** 
SexM              1.360006   0.261931   5.192 2.08e-07 ***
ChestPainTypeATA -2.132805   0.298477  -7.146 8.96e-13 ***
ChestPainTypeNAP -1.639813   0.238856  -6.865 6.64e-12 ***
ChestPainTypeTA  -1.162777   0.407187  -2.856   0.0043 ** 
RestingBP         0.002389   0.005468   0.437   0.6622    
RestingECGNormal -0.143833   0.264415  -0.544   0.5865    
RestingECGST     -0.075508   0.327142  -0.231   0.8175    
MaxHR            -0.011072   0.004606  -2.404   0.0162 *  
ExerciseAnginaY   1.519939   0.223682   6.795 1.08e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1003.3  on 733  degrees of freedom
Residual deviance:  637.9  on 723  degrees of freedom
AIC: 659.9

Number of Fisher Scoring iterations: 5
```

LR_2 is the best model, as it has the highest predictive accuracy.

Question 2

```
# predict using test set
preds <- predict(LR_2, newdata = test)
# make confusion matrix by comparing predicted response to ground truth
cm <- confusionMatrix(preds, test$HasHeartDisease, positive = "Y")

cm
```

```
Confusion Matrix and Statistics

          Reference
Prediction  N  Y
         N 73 18
         Y 21 72

               Accuracy : 0.788
                 95% CI : (0.7218, 0.8447)
    No Information Rate : 0.5109
    P-Value [Acc > NIR] : 7.333e-15

                  Kappa : 0.5762

 Mcnemar's Test P-Value : 0.7488

            Sensitivity : 0.8000
            Specificity : 0.7766
         Pos Pred Value : 0.7742
         Neg Pred Value : 0.8022
             Prevalence : 0.4891
         Detection Rate : 0.3913
   Detection Prevalence : 0.5054
      Balanced Accuracy : 0.7883

       'Positive' Class : Y
```

Question 3

```r
# show sensitivity and specificity
cm$byClass["Sensitivity"]
```

```
Sensitivity 
        0.8 
```

```r
cm$byClass["Specificity"]
```

```
Specificity 
  0.7765957 
```

The model correctly identified 80% of patients who actually do have heart disease. This is a fairly high true positive rate, which reduces the chance of false negatives. The model correctly identified 77.7% of patients who do not have heart disease. So, the model does fairly well at avoiding false positives.