558 Homework 5

Susan Hajmohammad

Task 1

- Question 1: What is the purpose of using cross-validation when fitting a random forest model? The purpose of using cross validation when fitting a random forest model is to rotate through the data partitions so each one has a turn testing the model. That way we can see how well the random forest model performs on new data multiple times.
- Question 2: Describe the bagged tree algorithm. Bagged tree algorithm is bootstrapping samples then aggregating. We would make some new datasets using the bootstrapping method (with replacement, non-parametric), then create a full tree on each new dataset. We then average the results of the trees and in theory the averaged results are more reliable than just making one tree.
- Question 3: What is meant by a general linear model? A general linear model is a regression model where the model is generally Y= intercept + betas*x's + an error term. You can have SLR, MLR and ANOVA models too.
- 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 looks at how two variables affect the response together. When the model doesn't include an interaction term, the model is just looking at how the variables affect the response independently.
- Question 5: Why do we split our data into a training and test set? That way we have a chunk of data that we didn't train the model on, so we can see how it does with predicting new data it hasn't seen yet. If we just used all the data to train the model we wouldn't have data to test it with!

Task 2

Packages and Data

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

heart_data <- read_csv("heart.csv")
```

Question 1

```r
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  
```

- a) What type of variable (in R) is Heart Disease? Categorical or Quantitative? Heart disease appears to be quantitative.

- b)Does this make sense? Why or why not. This doesn't really make sense since Heart Disease is supposed to be a binary response like True or False.

Question 2

```
new_heart <- heart_data %>%
  mutate(heart_disease = as.factor(HeartDisease))%>%
  select(-HeartDisease, -ST_Slope)

summary(new_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        heart_disease
 Length:918         Min.   :-2.6000   0:410        
 Class :character   1st Qu.: 0.0000   1:508        
 Mode  :character   Median : 0.6000                
                    Mean   : 0.8874                
                    3rd Qu.: 1.5000                
                    Max.   : 6.2000                
```

Task 3

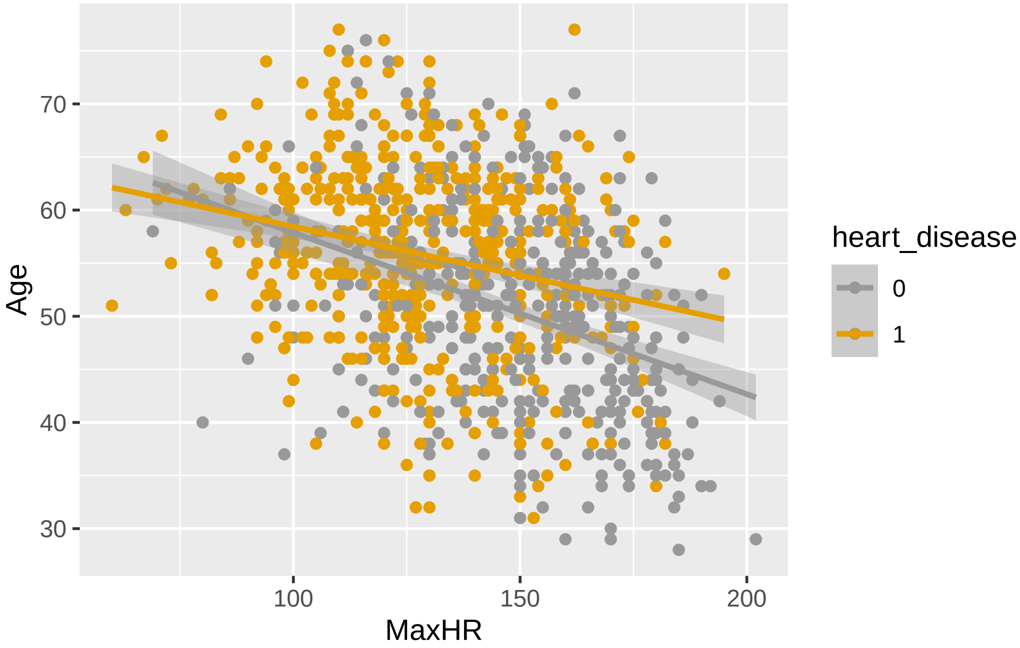
Question 1

```
#colorblind friendly scatterplot for age as function of heart disease
#palette from cookbook-r.com
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "

p <- ggplot(data = new_heart, mapping = aes(x = MaxHR, y = Age, color = heart_disease))

p+ geom_point() + geom_smooth(method = "lm") +  scale_colour_manual(values=cbPalette)
```

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



Question 2

Based on the graph visually, I think there is evidence for interaction because the two lines aren't parrallel and cross each other.

Task 4

Split data into training and test set:

```
set.seed(101)
new_heart_split <- initial_split(new_heart, prop = 0.8)

test <- testing(new_heart_split)

train <- training(new_heart_split)
```

Task 5

Question 1

```
# fit interaction model named ols_mlr

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_disease1       -8.58502    3.83433  -2.239  0.02546 *  
MaxHR:heart_disease1  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

```
test_model <- predict(ols_mlr, newdata = test)

# calculation for RMSE
sqrt(mean((test$Age - test_model)^2))
```

```
[1] 9.100206
```

Question 3

```
#LASSO recipe

LASSO_recipe <- recipe(Age ~ MaxHR + heart_disease, data = train) %>%
  step_dummy(heart_disease) %>%
  step_normalize(all_numeric_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_numeric_predictors()

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

Question 4

```r
#model spec
lasso_spec <- linear_reg(penalty = tune(), mixture = 1) |>
  set_engine("glmnet") |>
  set_mode("regression")

#tuning grid
lambda_grid <- grid_regular(penalty(), levels = 30)

#lasso workflow
lasso_wkf <- workflow() |>
  add_recipe(LASSO_recipe) |>
  add_model(lasso_spec)

#Cv folds
set.seed(101)
cv_splits <- vfold_cv(train, v = 10)

#tune model on grid

lasso_fit <- lasso_wkf |>
  tune_grid(
    resamples = cv_splits,
    grid      = lambda_grid,
    metrics   = metric_set(rmse))
```

```
Warning: package 'glmnet' was built under R version 4.4.3
```

```r
#selecting best penalty
lowest_rmse <- lasso_fit |>
  select_best(metric =  "rmse")
```

```
#fit lasso on all training data
final_lasso <- lasso_wkf |>
  finalize_workflow(lowest_rmse) |>
  fit(data = train)

#final coefficients
tidy(final_lasso)
```

```
# 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_X1               1.36 0.0000000001
4 MaxHR_x_heart_disease_X1       1.03 0.0000000001
```

Question 5

Without even looking, I'd expect them to be roughly the same because the penalty is almost 0 (above). So the LASSO barely shrank the coefficients from their original values in the OLS, i think the test data RMSE will be almost the same for both. ### Question 6

```
ols_rmse <- rmse_vec(
  truth    = test$Age,
  estimate = predict(ols_mlr, newdata = test)
)
ols_rmse
```

```
[1] 9.100206
```

```
lasso_rmse <- rmse_vec(
  truth    = test$Age,
  estimate = predict(final_lasso, new_data = test)$.pred
)
lasso_rmse
```

```
[1] 9.095981
```

Question 7

Because the cross validation penalty is almost 0. That means the shrinkage is doing almost nothing to the lasso coefficients.

Task 6

Question 1

```
set.seed(101)

# recode & split
heart_data <- heart_data %>%
  mutate(HeartDisease = factor(HeartDisease))
heart_split <- initial_split(heart_data, prop = 0.8)
heart_train <- training(heart_split)
heart_test  <- testing(heart_split)

# 10-fold CV on training set
heart_CV_folds <- vfold_cv(heart_train, v = 10)

#  Recipes for  models
# model1 Age + Sex
LR1_rec <- recipe(HeartDisease ~ Age + Sex, data = heart_train) %>%
  step_normalize(Age) %>%
  step_dummy(Sex)

# model2 Age + Sex + ChestPainType + RestingBP + RestingECG + MaxHR + ExerciseAngina
LR2_rec <- recipe(HeartDisease ~ Age + Sex + ChestPainType + RestingBP + RestingECG + MaxHR +
                  data = heart_train) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_dummy(all_nominal_predictors())

#  Specify logistic regression
LR_spec <- logistic_reg() %>% set_engine("glm")

# workflows
LR1_wkf <- workflow() %>% add_recipe(LR1_rec) %>% add_model(LR_spec)
LR2_wkf <- workflow() %>% add_recipe(LR2_rec) %>% add_model(LR_spec)

#fit with cv folds
```

```
LR1_res <- LR1_wkf %>% fit_resamples(resamples = heart_CV_folds,
                                     metrics   = metric_set(accuracy, mn_log_loss))
LR2_res <- LR2_wkf %>% fit_resamples(resamples = heart_CV_folds,
                                     metrics   = metric_set(accuracy, mn_log_loss))

cv_compare <- bind_rows(
  LR1_res %>% collect_metrics() %>% mutate(Model = "Model1"),
  LR2_res %>% collect_metrics() %>% mutate(Model = "Model2")
) %>%
  select(Model, .metric, mean, std_err)

cv_compare
```

```
# A tibble: 4 x 4
  Model  .metric      mean std_err
  <chr>  <chr>       <dbl>   <dbl>
1 Model1 accuracy    0.673  0.0165
2 Model1 mn_log_loss 0.602  0.0179
3 Model2 accuracy    0.789  0.0130
4 Model2 mn_log_loss 0.452  0.0148
```

```
# final fit
final_wkf <- LR2_wkf %>% fit(data = heart_train)

# confusion matrix on test
test_preds <- predict(final_wkf, heart_test) %>%
  bind_cols(heart_test)

test_cm <- conf_mat(test_preds, truth = HeartDisease, estimate = .pred_class)
test_cm
```

```
          Truth
Prediction  0  1
         0 73 18
         1 21 72
```

```
# extract sensitivity & specificity
test_cm %>% summary()
```

```
# A tibble: 13 x 3
```

```
   .metric              .estimator .estimate
   <chr>                <chr>          <dbl>
 1 accuracy             binary         0.788
 2 kap                  binary         0.576
 3 sens                 binary         0.777
 4 spec                 binary         0.8  
 5 ppv                  binary         0.802
 6 npv                  binary         0.774
 7 mcc                  binary         0.576
 8 j_index              binary         0.577
 9 bal_accuracy         binary         0.788
10 detection_prevalence binary         0.495
11 precision            binary         0.802
12 recall               binary         0.777
13 f_meas               binary         0.789
```

The model is about 81% accurate for patients. Sensisitive, 71% with HD were correctly classified. Spec, 89% without HD were correct as well.