HW5 - Model Comparison

Task 1 - Conceptual Questions

- 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 this allows the model to be trained and tested on different sets of data without having to compromise the size of the data. This allows the random forest model to be fitted on the entire data set through cross-validation.
- Describe the bagged tree algorithm.
  - In the bagged tree algorithm, the data is treated as a population where samples of this data is taken. Each sample could have duplicates or missing values. A tree will be fitted on each sample from the data, then the average of these trees are taken to determine a prediction for the data set.
- What is meant by a general linear model?
  - A general linear model expands on the linear regression model, working well with dependent variables that don't have a normal distribution or response variables that are not continuous.
- 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?
  - When fitting a multiple linear regression model, adding an interaction term will cause the coefficients to drastically change because the variables will now also depend on another variable. This allows the model to show the relationship between the variables on the response as there will be different outcomes with a variable at different levels of another variable.
- Why do we split our data into a training and test set?

- We split our data into a training and test set because we don't want the model to be fitted exactly to the entire data set making it not as good at predicting data it has not seen yet. The model can be fit to the training set, then the test set can be used to determine how well the model does on data it has not seen, allowing for the best model fit to be chosen.

Task 2 - Data Prep

packages and data

```
#packages to library
library(tidyverse)
library(tidymodels)
library(caret)
library(yardstick)
library(glmnet)

#read in data as a tibble
heart_data <- as_tibble(read.csv("data/heart.csv"))
heart_data
```

```
# A tibble: 918 x 12
     Age Sex   ChestPainType RestingBP Cholesterol FastingBS RestingECG MaxHR
   <int> <chr> <chr>              <int>       <int>     <int> <chr>      <int>
 1    40 M     ATA                  140         289         0 Normal       172
 2    49 F     NAP                  160         180         0 Normal       156
 3    37 M     ATA                  130         283         0 ST            98
 4    48 F     ASY                  138         214         0 Normal       108
 5    54 M     NAP                  150         195         0 Normal       122
 6    39 M     NAP                  120         339         0 Normal       170
 7    45 F     ATA                  130         237         0 Normal       170
 8    54 M     ATA                  110         208         0 Normal       142
 9    37 M     ASY                  140         207         0 Normal       130
10    48 F     ATA                  120         284         0 Normal       120
# i 908 more rows
# i 4 more variables: ExerciseAngina <chr>, Oldpeak <dbl>, ST_Slope <chr>,
#   HeartDisease <int>
```

```
#summarize the data
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
```

Heart Disease is a categorical variable showing whether the patient has a heart disease or not. This does make sense because the summary shows either 0 for no heart disease or 1 for having heart disease.

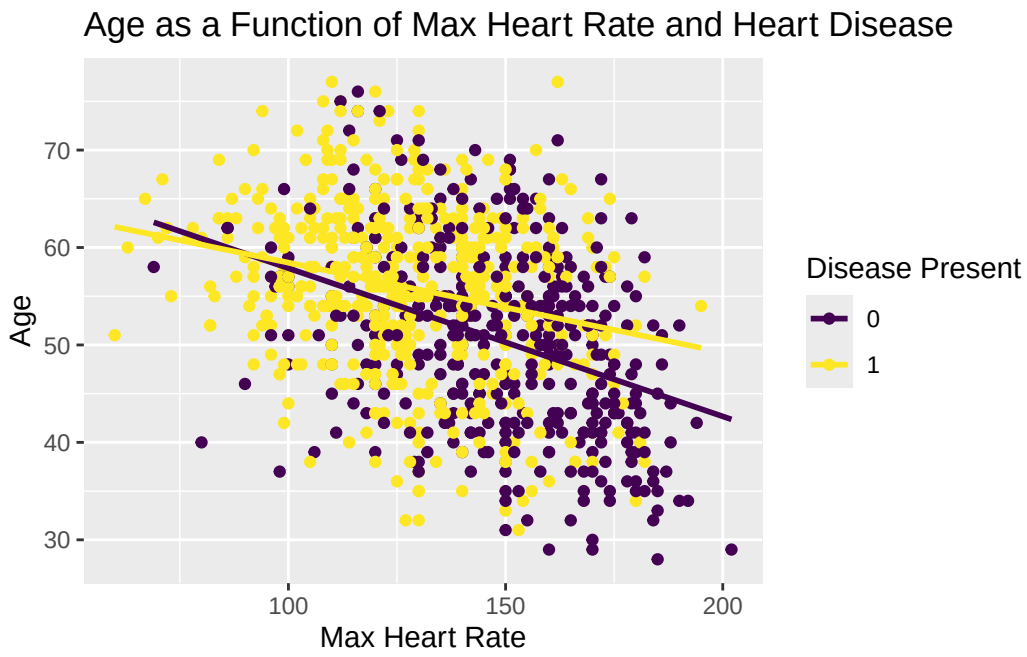
```
#change HeartDisease to categorical and remove some variables
new_heart <- heart_data |>
  mutate(DiseasePresent = as.factor(HeartDisease)) |>
  select(-ST_Slope, -HeartDisease)

new_heart
```

```
# A tibble: 918 x 11
     Age Sex   ChestPainType RestingBP Cholesterol FastingBS RestingECG MaxHR
   <int> <chr> <chr>              <int>       <int>     <int> <chr>      <int>
 1    40 M     ATA                  140         289         0 Normal       172
 2    49 F     NAP                  160         180         0 Normal       156
 3    37 M     ATA                  130         283         0 ST            98
 4    48 F     ASY                  138         214         0 Normal       108
 5    54 M     NAP                  150         195         0 Normal       122
 6    39 M     NAP                  120         339         0 Normal       170
```

```
 7    45 F      ATA                 130          237          0 Normal        170
 8    54 M      ATA                 110          208          0 Normal        142
 9    37 M      ASY                 140          207          0 Normal        130
10    48 F      ATA                 120          284          0 Normal        120
# i 908 more rows
# i 3 more variables: ExerciseAngina <chr>, Oldpeak <dbl>, DiseasePresent <fct>
```

Task 3 - EDA

```
#create scatter plot of data
ggplot(new_heart, aes(x = MaxHR, y = Age, color = DiseasePresent)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Max Heart Rate",
       title = "Age as a Function of Max Heart Rate and Heart Disease",
       color = "Disease Present") +
  scale_color_manual(labels = c("No", "Yes")) +
  scale_color_viridis_d()
```



An interaction model would be the most appropriate for this data because based on the scatter plots, the lines for when the patient has a heart disease and when they do not intersect, showing an interaction.

Task 4 - Testing and Training

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

Task 5 - OLS and LASSO

OLS Model

```
#fit interaction model
ols_mlr <- lm(Age ~ MaxHR*DiseasePresent, data = train)

summary(ols_mlr)
```

```
Call:
lm(formula = Age ~ MaxHR * DiseasePresent, 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 ***
DiseasePresent1      -8.58502    3.83433  -2.239  0.02546 *
MaxHR:DiseasePresent1  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
```

```r
#test ols model on test data
ols_prediction <- predict(ols_mlr, newdata = test)
pred_ols <- test |> mutate(Prediction = ols_prediction)

#calculate rmse
rmse(pred_ols, truth = Age, estimate = Prediction)
```

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

LASSO Model

```r
#set up 10 fold CV
CV_folds <- vfold_cv(train, 10)

#set up LASSO recipe
LASSO_recipe <- recipe(Age ~ MaxHR + DiseasePresent, data = train) |>
  step_dummy(DiseasePresent) |>
  step_normalize(MaxHR) |>
  step_interact(~MaxHR:starts_with("DiseasePresent"))

LASSO_recipe
```

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



-- Inputs

Number of variables by role

outcome:   1
predictor: 2
```

```
-- Operations

* Dummy variables from: DiseasePresent

* Centering and scaling for: MaxHR

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

```
#set up LASSO spec
LASSO_spec <- linear_reg(penalty = tune(), mixture = 1) |>
  set_engine("glmnet")

#create LASSO workflow
LASSO_wkf <- workflow() |>
  add_recipe(LASSO_recipe) |>
  add_model(LASSO_spec)

#set up LASSO grid
LASSO_grid <- LASSO_wkf |>
  tune_grid(resamples = CV_folds, grid = grid_regular(penalty(), levels = 200))

#determine best tunning parameter
lowest_rmse <- LASSO_grid |>
  select_best(metric = "rmse")

#fit training set to model
LASSO_final <- LASSO_wkf |>
  finalize_workflow(lowest_rmse) |>
  fit(train)
tidy(LASSO_final)
```

```
# A tibble: 4 x 3
  term                      estimate penalty
  <chr>                        <dbl>   <dbl>
1 (Intercept)                  52.5   0.0174
2 MaxHR                        -4.22  0.0174
3 DiseasePresent_X1             2.75  0.0174
4 MaxHR_x_DiseasePresent_X1     2.00  0.0174
```

The RMSE calculation between the ols and the LASSO model should be roughly the same because both models are comparing the same two variables and their interaction.

```
#test LASSO model on test data
LASSO_pred <- predict(LASSO_final, new_data = test)
pred_LASSO <- test |> mutate(Prediction = LASSO_pred$.pred)

#calculate rmse
rmse(pred_LASSO, truth = Age, estimate = Prediction)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard        9.09
```

The RMSE for the ols model of 9.100206 is roughly the same as the RMSE for the LASSO model of 9.09553.

The RMSE calculations are roughly the same even though the coefficients for each model is different because the RMSE is showing how far the predictions are from the actual values. The two models have different ways of predicting the values, but still end up with similar accuracy to the actual values.

Task 6 - Logistic Regression

LR Model 1

```
#set up logistic regression recipe
lr1_rec <- recipe(DiseasePresent ~ Age + Sex + MaxHR, data = train) |>
  step_normalize(all_numeric()) |>
  step_dummy(Sex)

#set up logistic regression spec
lr1_spec <- logistic_reg() |>
  set_engine("glm")

#set up logistic regression workflow
lr1_wkf <- workflow() |>
  add_recipe(lr1_rec) |>
  add_model(lr1_spec)
```

```
#fit data to CV folds
lr1_fit <- lr1_wkf |>
  fit_resamples(CV_folds, metrics = metric_set(accuracy, mn_log_loss))
```

LR Model 2

```
#set up logistic regression recipe
lr2_rec <- recipe(DiseasePresent ~ Age + Sex + RestingBP + Cholesterol,
                  data = train) |>
  step_normalize(all_numeric()) |>
  step_dummy(Sex)

#set up logistic regression spec
lr2_spec <- logistic_reg() |>
  set_engine("glm")

#set up logistic regression workflow
lr2_wkf <- workflow() |>
  add_recipe(lr2_rec) |>
  add_model(lr2_spec)

#fit data to CV folds
lr2_fit <- lr2_wkf |>
  fit_resamples(CV_folds, metrics = metric_set(accuracy, mn_log_loss))
```

Compare LR Models

```
#compare metrics for both models
rbind(lr1_fit |> collect_metrics(),
      lr2_fit |> collect_metrics()) |>
  mutate(Model = c("Model 1", "Model 1", "Model 2", "Model 2")) |>
  select(Model, everything())
```

```
# A tibble: 4 x 7
  Model   .metric     .estimator  mean     n std_err .config
  <chr>   <chr>       <chr>      <dbl> <int>   <dbl> <chr>
1 Model 1 accuracy    binary     0.700    10  0.0134 Preprocessor1_Model1
```

```
2 Model 1 mn_log_loss binary     0.567    10  0.0169 Preprocessor1_Model1
3 Model 2 accuracy    binary     0.696    10  0.0207 Preprocessor1_Model1
4 Model 2 mn_log_loss binary     0.588    10  0.0219 Preprocessor1_Model1
```

Model 1 is the best logistic regression model because it has the lowest loss log metric.

Test LR Model on Test set

```
#fit test data
LR_test_fit <- lr1_wkf |>
  fit(test)

#use confusionMatrix() function
conf_mat(test |> mutate(estimate = LR_test_fit |> predict(test) |> pull()),
         DiseasePresent, estimate)
```

```
          Truth
Prediction  0  1
         0 69 20
         1 25 70
```

Sensitivity is the measure of the true positive rate. On the test data, the model was able to correctly identify 70 out of the 90 positives giving a 77.8% sensitivity. This means that the model will be able to identify when a patient has a heart disease 77.8% of the time when the patient actually does have a heart disease.

Specificity is the measure of the true negative rate. The model was able to correctly identify 69 out of the 94 instances of negatives from the test data, giving a 73.4% specificity. This means that the model will be able to identify when the patient does not have a heart disease 73.4% of the time when the patient actually does not have a heart disease.