Statistical/Machine Learning Methods

Zach Ginder

Task 1: Conceptual Questions

- 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 that it allows us to get a better sense of how well each random forest model performs on unseen data. We want to first tune the tree to determine the number of parameters that will be randomly utilized for each split of the tree model. By staying within the training data set to do this and using cross validation, it allows us to leave the test set untouched and leave that for comparison to other model types and/or a final model test. The more often we utilize the test set we run the risk of over fitting to our data, which cross validation helps prevent.
- Question 2: Describe the bagged tree algorithm.
  - A bagged tree algorithm is a type of ensemble tree method. First you split the data set into a training and test set. In order to tune the model paramteres we will use k-fold cross validation within the training data set. For each tuning parameter value we perform B bootstrap resamples within each of the k-folds, fitting a tree to each resample, and aggregating the predictions within each fold. For each tuning parameter value we determing the average prediction accuracy or some other model metric to determine the optimal tuning parameters. The best tuning parameters are then utilized to fit a bagged tree utilizing just the training data set. We then can compare our tunned bagged tree model to other models based on prediction on the test data set and utilizing some sort of metric.
- Question 3: What is meant by a general linear model?
  - A general linear model allows for more flexibility than standard linear regression by allowing the errors of the response variable to be non-normal. One example of its utilization is when fitting logistic regression models with the logit link function, which allows us to model a response variable that is binary.

- 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?
  - By adding an interaction term it allows for the effect of one variable to depend on the value of another. An interaction term normally effects the slope term of another variable whereas the slope of one term depends on the value of the other term.
- Question 5: Why do we split our data into a training and test set?
  - We split our data into a training and test set because if we want our model to do well for prediction this means we want it to predict well for observations it has yet to see. By fitting our model to the training set, we can then use the test set to make predictions and judge the effectiveness of our model with our specified metric.

Task 2: Data Prep

Packages and Data

```
#Library these packages
library(tidyverse)
library(tidymodels)
library(caret)
library(yardstick)

#Read in heart.csv as tibble
heart<-read_csv("heart.csv")
```

Question 1

```
#Run and report summary()
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.) What type of variable (in R) is Heart Disease? Categorical or Quantitative?

- Heart Disease is currently quantitative in this tibble.

b.) Does this make sense? Why or why not.

- This does not make sense as the HeartDisease variable is a binary variable (0 or 1) indicating whether heart disease is present or not. Therefore, the variable HeartDisease should be a factor variable.

Question 2

```r
#Update the Heart tibble
new_heart<-heart|>
  mutate(HeartDiseaseIndicator=as.factor(HeartDisease))|>
  select(-ST_Slope,-HeartDisease)
```

Task 3: EDA

Question 1

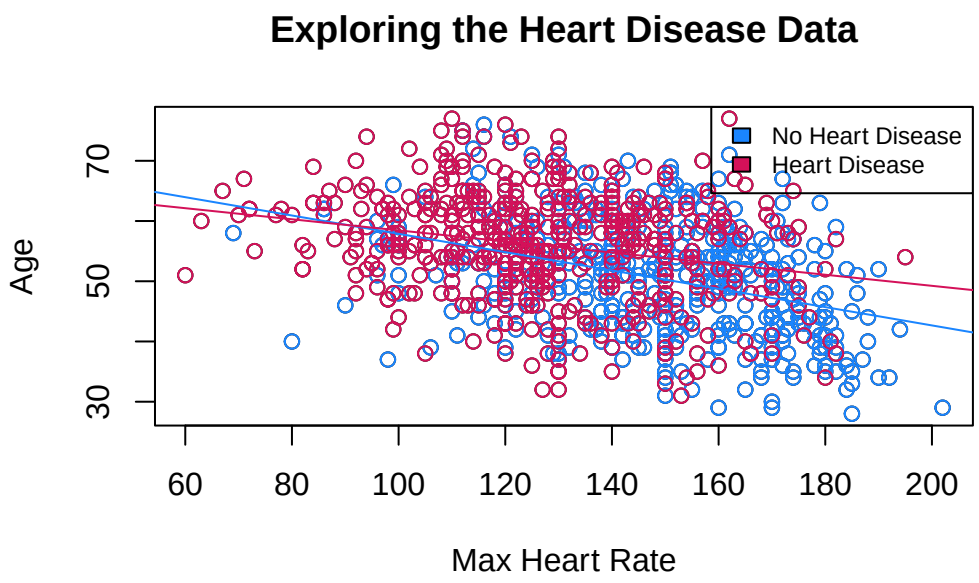
```r
#Create scatterplot for MaxHR vs Age with color being the HeartDiseaseIndicator
plot(x=new_heart$MaxHR,y=new_heart$Age,xlab="Max Heart Rate",ylab="Age")

#Add colored points based on the heart disease indicator variable
points(x=new_heart[new_heart$HeartDiseaseIndicator==0,]$MaxHR,
       y=new_heart[new_heart$HeartDiseaseIndicator==0,]$Age,col="#1A85FF")
points(x=new_heart[new_heart$HeartDiseaseIndicator==1,]$MaxHR,
       y=new_heart[new_heart$HeartDiseaseIndicator==1,]$Age,col="#D41159")

#Add linear regression lines for MaxHR vs Age for each level of the
#HeartDiseaseIndicator variable
abline(lm(Age~MaxHR,data=new_heart[new_heart$HeartDiseaseIndicator==0,]),
          col="#1A85FF")
abline(lm(Age~MaxHR,data=new_heart[new_heart$HeartDiseaseIndicator==1,]),
          col="#D41159")

#Add a legend
legend("topright",legend=c("No Heart Disease", "Heart Disease"),
       fill=c("#1A85FF","#D41159"),cex=.75)

#Add a title
title("Exploring the Heart Disease Data")
```



Question 2

Based on the scatter plot we would say that an interaction model would be appropriate. We determine this by looking at the slopes of the lines for heart disease and no heart disease. Since the lines for heart disease and no heart disease are not parallel it suggests that the effects of max heart rate on age is dependent on heart disease status. This suggests that an interaction model is more appropriate than an additive model.

Task 4: Testing and Training

```
#Set seed for reproducibility
set.seed(101)

#Split dataset into training and test sets
heart_split<-initial_split(new_heart,prop=0.8)
train<-training(heart_split)
test<-testing(heart_split)
```

Task 5: OLS and LASSO

Question 1

```
#Fit an OLS model
ols_mlr<-lm(Age~MaxHR*HeartDiseaseIndicator,data=new_heart)
summary(ols_mlr)
```

```
Call:
lm(formula = Age ~ MaxHR * HeartDiseaseIndicator, data = new_heart)

Residuals:
     Min       1Q   Median       3Q      Max
-23.9600  -5.8521   0.3956   5.9879  24.2622

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                 73.08380    2.73877  26.685  < 2e-16 ***
MaxHR                       -0.15209    0.01826  -8.328 2.98e-16 ***
```

```
HeartDiseaseIndicator1       -5.43219    3.46324  -1.569   0.1171
MaxHR:HeartDiseaseIndicator1  0.06003    0.02450   2.450   0.0145 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.601 on 914 degrees of freedom
Multiple R-squared:  0.1712,    Adjusted R-squared:  0.1685
F-statistic: 62.95 on 3 and 914 DF,  p-value: < 2.2e-16
```

Question 2

```
#Calculate test RMSE
sqrt(mean((test$Age-predict(ols_mlr,newdata = test))^2))
```

```
[1] 9.053599
```

Question 3

```
#Create folds
heart_CV_folds<-vfold_cv(train,10)

#Create a LASSO recipe
LASSO_recipe<-recipe(Age~MaxHR+HeartDiseaseIndicator,data=train)|>
  step_normalize(MaxHR)|>
  step_dummy(HeartDiseaseIndicator)|>
  step_interact(~MaxHR:starts_with("HeartDiseaseIndicator"))
LASSO_recipe
```

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

```
-- Inputs
```

```
Number of variables by role
```

```
outcome:   1
predictor: 2



-- Operations

* Centering and scaling for: MaxHR

* Dummy variables from: HeartDiseaseIndicator

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

Question 4

```
#Set spec
LASSO_spec <- linear_reg(penalty = tune(), mixture = 1) |>
  set_engine("glmnet")
#Create workflow
LASSO_wkf <- workflow() |>
  add_recipe(LASSO_recipe) |>
  add_model(LASSO_spec)

#Create tuning grid for LASSO
LASSO_grid <- LASSO_wkf |>
  tune_grid(resamples = heart_CV_folds,
            grid = grid_regular(penalty(), levels = 200))

#Final tuning parameter with lowest rmse
lowest_rmse <- LASSO_grid |>
  select_best(metric = "rmse")

#fit it to the entire training set to see the model fit
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 HeartDiseaseIndicator_X1                2.75  0.0174
4 MaxHR_x_HeartDiseaseIndicator_X1        2.00  0.0174
```

Question 5

I would expect the RMSE values to be fairly close as the penalty for the "best" LASSO model is close to zero. A LASSO penalty of zero is essentially an ordinary linear regression model therefore the RMSE of this LASSO model should be similar to the RMSE of the ordinary least squares "best" model.

Question 6

```
#Calculate test RMSE
OLS_RMSE<-sqrt(mean((test$Age-predict(ols_mlr,newdata = test))^2))

#Calculate test RMSE
LASSO_RMSE<-LASSO_final |>
  predict(test) |>
  pull() |>
  rmse_vec(truth = test$Age)

cat("The RMSE of the OLS Model is",OLS_RMSE,"\n")
```

```
The RMSE of the OLS Model is 9.053599
```

```
cat("The RMSE of the LASSO Model is",LASSO_RMSE,"\n")
```

```
The RMSE of the LASSO Model is 9.091133
```

Question 7

The RMSE calculations for the models are different because in LASSO the penalty term shrinks some of the less important predictors' coefficients towards zero. As we saw the penalty term is small so while the coefficients might be slightly shrunk, the LASSO model is actually close to the OLS model due to the small penalty term and therefore the RMSEs are very similar.

Task 6: Logistic Regression

Question 1

```
#LR Model 1
LR1_rec<-recipe(HeartDiseaseIndicator~Age+MaxHR,data=train)|>
  step_normalize(all_numeric(), -HeartDiseaseIndicator)
LR_spec<-logistic_reg()|>
  set_engine("glm")
LR1_wkf<-workflow()|>
  add_recipe(LR1_rec)|>
  add_model(LR_spec)
LR1_fit<-LR1_wkf|>
  fit_resamples(heart_CV_folds,metrics=metric_set(accuracy,mn_log_loss))


#LR Model 2
LR2_rec<-recipe(HeartDiseaseIndicator~Age+MaxHR+Cholesterol+Sex,data=train)|>
  step_normalize(all_numeric(), -HeartDiseaseIndicator)
LR2_wkf<-workflow()|>
  add_recipe(LR2_rec)|>
  add_model(LR_spec)
LR2_fit<-LR2_wkf|>
  fit_resamples(heart_CV_folds,metrics=metric_set(accuracy,mn_log_loss))
rbind(LR1_fit|>collect_metrics(),
      LR2_fit|>collect_metrics())
```

```
# A tibble: 4 x 6
  .metric     .estimator  mean     n std_err .config
  <chr>       <chr>      <dbl> <int>   <dbl> <chr>
1 accuracy    binary     0.688    10  0.0180 Preprocessor1_Model1
2 mn_log_loss binary     0.596    10  0.0121 Preprocessor1_Model1
3 accuracy    binary     0.707    10  0.0102 Preprocessor1_Model1
4 mn_log_loss binary     0.562    10  0.0178 Preprocessor1_Model1
```

The best performing model is the second model corresponding to the model with predictors Age, MaxHR, Cholesterol, and Sex when compared to the first model with only predictors of Age and MaxHR. By looking at the metrics when the model was fit on the training set using repeated CV, the accuracy was higher for model 2, and the log loss was lower for model 2. These are both associated with better models (higher accuracy and lower log loss).

Question 2

```
LR2_fit<- LR2_wkf |>
 fit(train)
conf_mat(test |> mutate(estimate = LR2_fit |> predict(test) |> pull()),
 HeartDiseaseIndicator,
 estimate)
```

```
          Truth
Prediction  0  1
         0 66 18
         1 28 72
```

Question 3

The sensitivity refers to when this model correctly identified cases that have heart disease.

$$SensitivityRate = \frac{72}{184} = 0.3913$$

The specificity refers to when this model correctly identified cases that do not have heart disease.

$$SpecificityRate = \frac{66}{184} = 0.3587$$