Homework 5

Hayden Morgan

Task 1: Conceptual Questions

Tyoe each question as part of a bulleted list and answer each question underneath each bullet.

- **Question 1: What is the purpose of using cross-validation when fitting a random forest model?**
  - Random forest models include hyperparameters such as the number of predictors that are sampled at each split. Through the process of tuning, hyperparameter values for random forest models can be estimated and selected for the best overall combination in order to produce the best model (e.g. most accurate model, model that is best at generalizing to unseen data). Cross-validation is an important part of the tuning process because it allows for different combinations of hyperparameters to be applied to different subsets of training data before ultimately sending the best model to the test data. Therefore, cross-validation can help avoid overfitting in the process of tuning random forest models.
- **Question 2: Describe the bagged tree algorithm.**
  - 'Bagging' refers to bootstrap aggregation. 'Bootstrapping' refers to resampling (either from data or a fitted model). In bootstrapping, multiple resamples are taken from an original data sample. Then, for the bagged tree algorithm, tree models are fit to each resample. From there, a prediction can be generated for each tree. At the end, the predictions are combined in some meaningful way to give a final prediction. For example, for continuous numerical responses you can combine predictions through averaging. For classification trees, you could use majority rule to identify the most common prediction from all trees from the bootstrapping step.
- **Question 3: What is meant by a general linear model?**
  - General linear model (GLM) refers to a group of models that can all use a linear equation to express a continuous outcome. For example, both simple and multiple linear regression models are GLMs, as well as ANOVAs. However, for GLMs, normal distribution is assumed, while generalized linear models can handle additional

models such as logistic regression because they don't depend on normal distribution specifically.

- **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?**
  - Adding an interaction effect means that you are investigating if the effect of one variable depends on the value/level of a different variable. In an expression, the interaction term would include two predictor variables multiplied together with the beta term. When the interaction term is NOT included, you can still investigate things like main effects (beta term multiplied by a predictor variable) e.g. if you want to discuss the effects of different predictors–but you would only be able to see if the predictors have main effects, not if they depend on each other. Including either/both main effects and interaction terms can make the model more flexible/complex overall. There is a risk of overfitting the model, though, if an interaction term is included when it is not needed.
- **Question 5: Why do we split our data into a training and test set?**
  - We split data into training and test sets because we want a model that is generalizeable and applicable to data it has yet to see. If you do not reserve part of your data as test data and thus train your model on the entirety of your data set, you have no way of knowing if your model only works really well with the data you've already given it, and not as well with similar data that it hasn't seen. This can be problematic assuming that you are creating a model to work with real-world data: you want to create something that can be applied to future data that is unknown in order to continue making valid predictions within the context of the model. If you use training data for your model and it works really well, but when run on test data your model doesn't perform as well, that's a cue to re-tune parameters/hyperparameters and try to get more accurate output.

Task 2: Data Prep

packages and data

First, create a sub-header titled packages and data and library the tidyverse, tidymodels, caret, and the yardstick package. We will need these our homework. In the code chunk setting, suppress all messages associated with calling these packages.

In the same code chunk, read in your data set as a tibble.

```r
#suppressed messages/warnings in code chunk
library("tidyverse")
library("tidymodels")
library("caret")
library("yardstick")

data <- read.csv("data/heart.csv")

data <- as.tibble(data)

data #proof of data as tibble
```

```
# 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>
```

Question 1

Run and report summary() on your data set.

```r
summary(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
```

What type of variable (in R) is Heart Disease? Categorical or Quantitative?

According to the tibble created in the previous step, Heart Disease is an integer in R, meaning it is quantitative.

Does this make sense? Why or why not.

No, this does not make sense. It is true that Heart Disease includes numbers like in quantitative data, but the numbers are just 0 and 1 and evidenced here:

```
print(data$HeartDisease)
```

```
  [1] 0 1 0 1 0 0 0 0 1 0 0 1 0 1 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 1
 [38] 0 0 0 0 1 0 0 1 0 0 0 0 1 1 1 0 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 1 0 1 0 1 0
 [75] 1 0 1 0 0 1 0 0 1 0 1 1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0
[112] 1 0 0 0 1 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0 1 1 1 0 1 0 0 1 1 1 1 1 0 1 0 0 0
[149] 0 1 0 0 0 0 0 1 1 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 1 0 0
[186] 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 1 0 1 0 0 0 1 1
[223] 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 1 0 0 1 0 0 0 0
[260] 0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1
[297] 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0
[334] 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
[371] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[408] 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 1 0 1 1 0 1 1 1 1 0 1 1 0 0 1 1 1 0 1 1 1 1
[445] 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 0 1 1 1 1 0 1 0 1 1 1 1
[482] 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 1 0 1 1 0 1 1
```

```
[519] 1 1 0 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 0 1 0 1 1 0
[556] 1 0 1 1 1 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 0
[593] 1 1 1 1 1 0 1 1 0 1 1 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 0 0 1 1 1 1 0 0 0 1
[630] 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 1 0 0 1 0 0 0 1 0 1 1 1 1 1
[667] 0 0 0 0 0 1 0 1 1 0 1 0 0 0 1 0 1 0 1 1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0
[704] 0 1 0 1 1 1 1 1 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 0 1 0 0 0 1 1 0 1 1 1 1 0 0
[741] 0 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 1
[778] 0 1 0 0 0 0 0 1 0 1 1 0 0 1 1 1 1 0 0 1 1 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0
[815] 1 0 1 1 1 1 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 1 1 0 1 0 1
[852] 0 1 0 0 1 0 0 1 0 1 1 0 1 1 1 0 1 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 1 0 0 1
[889] 1 1 0 0 0 1 0 1 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 1 1 1 1 1 1 0
```

This indicates that binary form is being used to indicate that Heart Disease is categorical rather than quantitative. This is further confirmed by visiting the website provided in the homework (here) where it is stated that a 1 means heart disease is present and 0 means the patient is normal (no heart disease).

Because of this binary data, treating Heart Disease like integer data and performing quantitative data analysis will not give us the insights desired for the data set. Rather, Heart Disease should be treated as categorical data in order for us to complete predictions and inferences on the data set that make sense in the context of the data.

Question 2

```
new_heart <- data |> #new data set is new_heart
  mutate(HeartDisease = factor(HeartDisease)) |> #changed data type,
  #removed old HeartDisease
  rename(HeartDisease_factor = HeartDisease) |> #renamed HeartDisease
  select(-ST_Slope) #remove ST_Slope

colnames(new_heart) #showing that HeartDisease has been replaced
```

```
 [1] "Age"                "Sex"                "ChestPainType"
 [4] "RestingBP"          "Cholesterol"        "FastingBS"
 [7] "RestingECG"         "MaxHR"              "ExerciseAngina"
[10] "Oldpeak"            "HeartDisease_factor"
```

```
#and there is no more ST_Slope
```

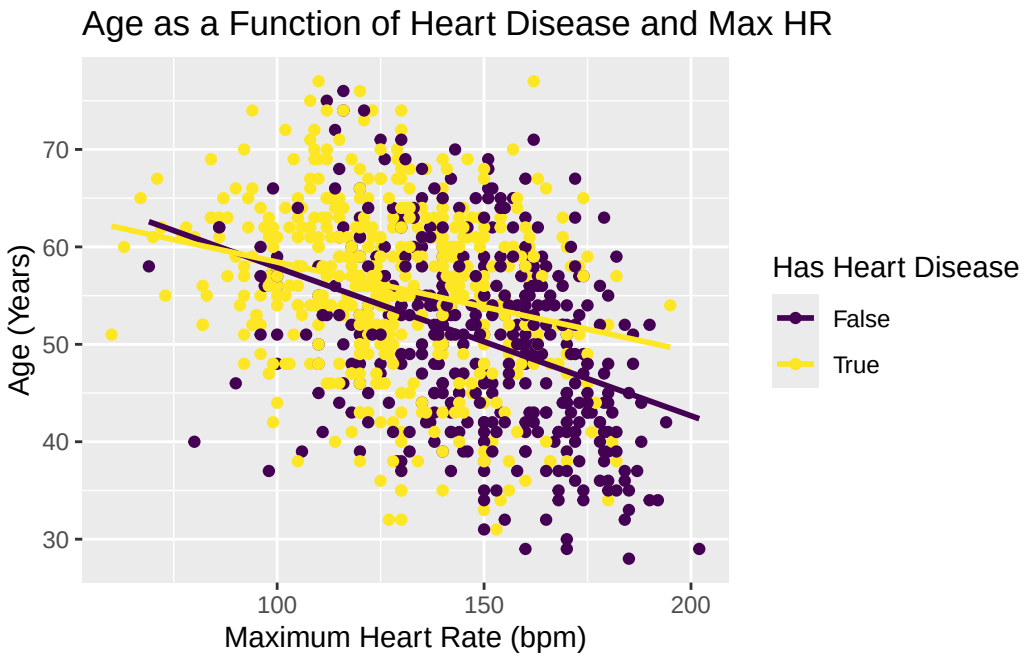
Task 3: EDA

Question 1

We are going to model someone's age (our response variable) as a function of heart disease and their max heart rate. First, create the appropriate scatterplot to visualize this relationship.

```
ggplot(new_heart, aes(x = MaxHR, y = Age, color = HeartDisease_factor))+
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)+ #add a line and remove SE bars
  labs(x = "Maximum Heart Rate (bpm)", y = "Age (Years)", #add labs
  title = "Age as a Function of Heart Disease and Max HR")+
  scale_color_viridis_d(name = "Has Heart Disease", labels = c("False", "True"))
```

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




```
  #colorblind friendly palette
```

Question 2

Based on visual evidence, do you think an interaction model or an additive model is more appropriate? Justify your answer.

Based on visual evidence, I think an interaction model may be more appropriate for this data than an additive model. An interaction model assumes that the effect of one predictor (e.g. maximum HR) depends on the level of a different predictor (e..g having heart disease). In this graph, it is clear that the slope between age and maximum HR for those without heart disease is much more negative than the slope between age and maximum HR for those WITH heart disease, so this indicates that there may be other factors (like max HR and heart disease interacting with each other) at play than just maximum HR and age. If an additive model was indicated instead, I would expect there to be less difference in the slopes of the two heart disease groups.

Task 4: Testing and Training

Split your data into a training and test set.

```
set.seed(101) #seed to 101

new_heart_split <- initial_split(new_heart, prop = 0.8) #80/20 split
train <- training(new_heart_split)
test <- testing(new_heart_split)

train
```

```
# A tibble: 734 x 11
     Age Sex   ChestPainType RestingBP Cholesterol FastingBS RestingECG MaxHR
   <int> <chr> <chr>              <int>       <int>     <int> <chr>       <int>
 1    41 M     ATA                  135         203         0 Normal        132
 2    37 M     NAP                  130         250         0 Normal        187
 3    63 M     NAP                  133           0         0 LVH           120
 4    39 F     NAP                  110         182         0 ST            180
 5    28 M     ATA                  130         132         0 LVH           185
 6    69 M     ASY                  130           0         1 ST            129
 7    53 M     ASY                  120           0         1 Normal        120
 8    58 M     ASY                  128         216         0 LVH           131
 9    53 M     ASY                   80           0         0 Normal        141
10    54 M     TA                   120         171         0 Normal        137
# i 724 more rows
# i 3 more variables: ExerciseAngina <chr>, Oldpeak <dbl>,
```

```
#   HeartDisease_factor <fct>
```

```
test
```

```
# A tibble: 184 x 11
     Age Sex   ChestPainType RestingBP Cholesterol FastingBS RestingECG MaxHR
   <int> <chr> <chr>             <int>       <int>     <int> <chr>      <int>
 1    54 M     ATA                 110         208         0 Normal       142
 2    48 F     ATA                 120         284         0 Normal       120
 3    58 M     ATA                 136         164         0 ST            99
 4    54 F     ATA                 120         273         0 Normal       150
 5    53 F     ATA                 113         468         0 Normal       127
 6    43 F     ATA                 150         186         0 Normal       154
 7    54 F     NAP                 130         294         0 ST           100
 8    52 M     NAP                 140         259         0 ST           170
 9    43 M     ASY                 120         175         0 Normal       120
10    37 M     ASY                 120         223         0 Normal       168
# i 174 more rows
# i 3 more variables: ExerciseAngina <chr>, Oldpeak <dbl>,
#   HeartDisease_factor <fct>
```

Task 5: OLS and LASSO

Question 1

First fit an interaction model (named ols_mlr) with age as your response, and max heart rate + heart disease as your explanatory variables using the training data set using ordinary least squares regression.

```
ols_mlr <- lm(Age ~ MaxHR + HeartDisease_factor + MaxHR * HeartDisease_factor,
              data = train)

summary(ols_mlr) #report summary output
```

```
Call:
lm(formula = Age ~ MaxHR + HeartDisease_factor + MaxHR * HeartDisease_factor,
    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 ***
HeartDisease_factor1        -8.58502    3.83433  -2.239  0.02546 *
MaxHR:HeartDisease_factor1   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
tested <- predict(ols_mlr, newdata = test) #test on testing data set

rmse_vec(test$Age, tested) #this is the RMSE value I want to report
```

```
[1] 9.100206
```

Question 3

```r
folds <- vfold_cv(train, 10) #CV, 10 fold

LASSO_recipe <- recipe(Age ~ MaxHR + HeartDisease_factor, data  = train) |>
  step_normalize(all_numeric_predictors()) |> #standardize predictors
  step_dummy(HeartDisease_factor) |>
  step_interact(~ MaxHR:starts_with("HeartDisease_factor_"))

LASSO_recipe #print recipe
```

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

```
-- Inputs

Number of variables by role

outcome:   1
predictor: 2



-- Operations

* Centering and scaling for: all_numeric_predictors()

* Dummy variables from: HeartDisease_factor

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

Question 4

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

wfl <- workflow() |>
  add_recipe(LASSO_recipe) |>
  add_model(spec)

wfl
```

```
== Workflow ====================================================================
Preprocessor: Recipe
Model: linear_reg()

-- Preprocessor ----------------------------------------------------------------
3 Recipe Steps

* step_normalize()
```

```
* step_dummy()
* step_interact()

-- Model ---------------------------------------------------------------------
Linear Regression Model Specification (regression)

Main Arguments:
  penalty = tune()
  mixture = 1

Computational engine: glmnet
```

```
grid <- wfl |> #set appropriate grid
  tune_grid(resamples = folds,
            grid = grid_regular(penalty(), levels = 10))

final_model <- grid |>
  select_best(metric = "rmse")

final_wfl <- wfl |> #select final model
  finalize_workflow(final_model) |>
  fit(train)

tidy(final_wfl) #report with tidy
```

```
# A tibble: 4 x 3
  term                           estimate      penalty
  <chr>                             <dbl>        <dbl>
1 (Intercept)                       52.5  0.0000000001
2 MaxHR                             -4.26 0.0000000001
3 HeartDisease_factor_X1             2.76 0.0000000001
4 MaxHR_x_HeartDisease_factor_X1     2.05 0.0000000001
```

Question 5

Without looking at the RMSE calculations, would you expect the RMSE calculations to be roughly the same or different? Justify your answer using output from your LASSO model.

Judging by the LASSO model output, I'd expect the RMSE calculations to be roughly the same, given the extremely small penalties for the LASSO model. Because the penalties are so

small, LASSO likely did not shrink the model coefficients much. So, RMSE is probably similar across the models.

Question 6

Now compare the RMSE between your OLS and LASSO model and show that the RMSE calculations were roughly the same.

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

OLS <- rmse_vec(test$Age, tested)

LASSO <- final_wfl |>
  predict(test) |> #eval on test data
  pull() |>
  rmse_vec(truth = test$Age)

OLS
```

```
[1] 9.100206
```

```
LASSO
```

```
[1] 9.09553
```

```
#the RMSEs are are roughly the same, indeed
```

Question 7

Why are the RMSE calculations roughly the same if the coefficients for each model are different?

The LASSO model can shrink coefficients in a model, e.g. if some predictors are irrelevant, etc. Given the small LASSO penalties and the fact that coefficients in LASSO were not near 0, both the LASSO and OLS models likely contain predictors that are actually useful to the context of the data set and data analysis. Therefore, because there was little difference in the structure of the models even after LASSO, this could explain why the RMSE calculations are roughly the same: the LASSO model did not find it necessary to drastically change the anatomy of the model compared to OLS so their outcomes are very similar.

Task 6: Logistic Regression

Question 1

Propose two different logistic regression models with heart disease as our response.

Fit those models on the training set, using repeated CV.

```
set.seed(101)

split_log <- initial_split(new_heart, prop = 0.8)
train_log <- training(split_log)
test_log <- testing(split_log)
folds_log <- vfold_cv(train_log, 10)

log1_rec <- recipe(HeartDisease_factor ~ Age, data = train_log) |>
  step_normalize(all_numeric())
log2_rec <- recipe(HeartDisease_factor ~ Age + MaxHR, data = train_log) |>
  step_normalize(all_numeric())

spec_log <- logistic_reg() |>
  set_engine("glm")

log1_wfl <- workflow() |>
  add_recipe(log1_rec) |>
  add_model(spec_log)
log2_wfl <- workflow() |>
  add_recipe(log2_rec) |>
  add_model(spec_log)

log1_fit <- log1_wfl |>
  fit_resamples(folds_log, metrics = metric_set(accuracy, mn_log_loss))
log2_fit <- log2_wfl |>
  fit_resamples(folds_log, metrics = metric_set(accuracy, mn_log_loss))

rbind(log1_fit |> collect_metrics(),
      log2_fit |> collect_metrics()) |>
  mutate(Model = c("Model1", "Model1", "Model2", "Model2")) |>
  select(Model, everything())
```

```
# A tibble: 4 x 7
  Model  .metric     .estimator  mean     n std_err .config
```

```
  <chr>  <chr>       <chr>      <dbl> <int>   <dbl> <chr>
1 Model1 accuracy    binary     0.630    10  0.0173 Preprocessor1_Model1
2 Model1 mn_log_loss binary     0.645    10  0.0117 Preprocessor1_Model1
3 Model2 accuracy    binary     0.688    10  0.0180 Preprocessor1_Model1
4 Model2 mn_log_loss binary     0.596    10  0.0121 Preprocessor1_Model1
```

Identify your best performing model. Justify why this is your best performing model and provide a basic summary of it.

The best performing model is Model 2. According to the tibble produced, Model 2 has both a larger accuracy and a smaller log loss than Model 1, both of which are favorable in this context. Model 2 is a more complex model than Model 1, including both Age and MaxHR whereas Model 1 just includes Age. In this case, it looks like adding more predictors to explain data trends makes the model better. This is not always true in data analysis scenarios, sometimes a simpler model ends up fitting data better- but not in this case.

Question 2

Lastly, check how well your chosen model does on the test set using the confusionMatrix() function.

```
log2_train_fit <- log2_wfl |>
  fit(train_log)

conf_mat(train_log |>
           mutate(estimate = log2_train_fit |>
                    predict(train_log) |>
                    pull()),
         HeartDisease_factor,
         estimate)
```

```
          Truth
Prediction   0   1
         0 178  93
         1 138 325
```

Question 3

Next, identify the values of sensitivity and specificity, and interpret them in the context of the problem.

Sensitivity is a true positive rate, or the probability of a positive test result that is truly positive. Based on the results of the confusion matrix, the sensitivity of Model 2 is as follows:

number of true positives / (number of true positives + number of false negatives) =

325 / (325 + 93) = 0.7775

In the context of this data set/problem, this means that the logistic regression model Model 2 has a 77.75% chance of positively identifying that heart disease is present when the patient does actually have heart disease.

Specificity is a true negative rate, or the probability of a negative test result that is truly negative. Based on the results of the confusion matrix, the specificity of Model 2 is as follows:

number of true negatives / (number of true negatives + number of false positives) =

178 / (178 + 138) = 0.5633

In the context of this data set/problem, this means that the logistic regression model Model 2 has a 56.33% chance of NOT finding heart disease when the patient, in fact, does not have heart disease.

So, overall, the model is pretty good at identifying if a patient has heart disease when they do, but not as good at identifying if the patient does NOT have heart disease when they don't. The model is more likely to return false negatives than false positives.