whole project

Ben Christensen, Amelia Ryan, Cecelia Kaufmann, Emma Nguyen and Caedmon Kollmer-Dorsey

4/21/2022

Final Project: Stat 253 - Statistical Machine Learning

May 9th, 2022

```
knitr::opts_chunk$set(echo = TRUE, eval = TRUE, warning = FALSE, message = FALSE, tidy = TRUE)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(readr)
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.1.2
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(astsa)
library(splines)
library(tidymodels)
```

```
## Warning: package 'tidymodels' was built under R version 4.1.2

## Registered S3 method overwritten by 'tune':
##   method                   from
##   required_pkgs.model_spec parsnip

## -- Attaching packages --------------------------------------- tidymodels 0.1.4 --

## v broom        0.7.9      v rsample      0.1.1
## v dials        0.0.10     v tibble       3.1.4
## v infer        1.0.0      v tidyr        1.1.4
## v modeldata    0.1.1      v tune         0.1.6
## v parsnip      0.1.7      v workflows    0.2.4
## v purrr        0.3.4      v workflowsets 0.1.0
## v recipes      0.1.17     v yardstick    0.0.9

## Warning: package 'dials' was built under R version 4.1.2

## Warning: package 'infer' was built under R version 4.1.2

## Warning: package 'modeldata' was built under R version 4.1.2

## Warning: package 'parsnip' was built under R version 4.1.2

## Warning: package 'recipes' was built under R version 4.1.2

## Warning: package 'rsample' was built under R version 4.1.2

## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'tune' was built under R version 4.1.2

## Warning: package 'workflows' was built under R version 4.1.2

## Warning: package 'workflowsets' was built under R version 4.1.2

## Warning: package 'yardstick' was built under R version 4.1.2

## -- Conflicts ------------------------------------------ tidymodels_conflicts() --
## x purrr::discard()  masks scales::discard()
## x dplyr::filter()   masks stats::filter()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Use tidymodels_prefer() to resolve common conflicts.
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(vip)
```

```
## Warning: package 'vip' was built under R version 4.1.3
```

```
##
## Attaching package: 'vip'
```

```
## The following object is masked from 'package:utils':
##
##     vi
```

```
library(probably)
```

```
## Warning: package 'probably' was built under R version 4.1.2
```

```
##
## Attaching package: 'probably'
```

```
## The following objects are masked from 'package:base':
##
##     as.factor, as.ordered
```

```
tidymodels_prefer()
conflicted::conflict_prefer("vi", "vip")
```

```
## [conflicted] Will prefer vip::vi over any other package
```

```
health1 <- read.csv("heart_failure_clinical_records_dataset.csv")
```

```
set.seed(321)

data_split <- initial_split(health1, prop = 0.75,strata = creatinine_phosphokinase) #Create Train/Test
health_train <- training(data_split) # Fit model to this
health_test  <- testing(data_split)
health_cv13 <- vfold_cv(health_train, v = 13, strata = creatinine_phosphokinase)
```

Data context

Clearly describe what the cases in the final clean dataset represent.

In our clean data set, a case is a person and specific variables in their medical records linked to a cardiovascular disease. These variables can be used to model and predict an instance of heart failure or (as the variable shows) a death event.

Broadly describe the variables used in your analyses.

The variables in this dataset are ones that can be used to predict heart failure, a common and deadly cardiovascular disease. Variables such as sex of participant and certain variables like smoking, diabetes, high blood pressure, anemia, and death event are all categorical (and binary) variables in this data set. There are also variables measuring measuring creatinine and ejection fraction (which is the amount of blood leaving

the heart during each contraction). There are variables measuring platelet count, creatinine, and sodium serum in the blood.

Who collected the data? When, why, and how? Answer as much of this as the available information allows.

The data contains the medical records of 299 patients with criteria that would make them at risk for heart failure. The data set was collected at the Faisalabad Institute of Cardiology and at the Allied Hospital in Punjab, Pakistan between April and December 2015.

Research question

Research question(s)/motivation for the regression task; make clear the outcome variable and its units.

For the regression task, we were trying to figure out what predictors had the greatest impact on CPK (creatinine phosphokinase). Our question was: What predictor had the greatest impact on CPK?

Research question(s)/motivation for the classification task; make clear the outcome variable and its possible categories.

For the classification task, we were trying to predict whether a patient would die of heart failure and what predictors best predicted this. The possible categories for the outcome variable were whether or not the patient died of heart failure (0 or 1).

Research question(s)/motivation for the unsupervised learning task?

Our research question for unsupervised learning was to see what the most common indicator of heart failure caused the death of a patient.

Regression

Methods

Describe the models used.

We used Ordinary Least Squares (OLS) and LASSO to build our model, which was to find what was greatest indicator of CPK levels.

Describe what you did to evaluate models.

We used different processing and the workflow to step over the predictors from CPK onwards to predict what had the greatest impact on CPK.

Describe the goals / purpose of the methods used in the overall context of your research investigations.

The purpose of the methods used was the predict the greatest impact on CPK. This was done with LASSO because it helps us to predict variable importance and OLS is a better way to predict, so we used that afterwards.

```
# model spec
lm_spec <-
    linear_reg() %>%
    set_engine(engine = 'lm') %>%
    set_mode('regression')
```

Indicate how you estimated quantitative evaluation metrics.

```
## recipe and wf
life_rec<-recipe(creatinine_phosphokinase ~ ., data = health_train) %>%
  step_nzv(all_numeric_predictors()) %>%
  step_corr(all_numeric_predictors()) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_dummy(all_nominal_predictors())

training_prep <- life_rec %>%
  prep() %>%
  juice()

lm_model_wf <- workflow() %>%
  add_recipe(life_rec) %>%
  add_model(lm_spec)

lm_fit_train <- lm_model_wf %>%
  fit(data=health_train)

training_prep %>%
  select(creatinine_phosphokinase) %>%
  bind_cols( predict(lm_fit_train, health_train)) %>%
  metrics(estimate = .pred, truth = creatinine_phosphokinase)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard     981.
## 2 rsq     standard       0.101
## 3 mae     standard     568.
```

```
lm_fit_train %>%
  tidy()
```

```
## # A tibble: 13 x 5
##    term                estimate std.error statistic  p.value
##    <chr>                  <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)          624.         67.6   9.23    2.92e-17
##  2 age                 -145.         70.9  -2.05    4.17e- 2
##  3 anaemia             -221.         69.4  -3.18    1.68e- 3
##  4 diabetes               7.09       69.1   0.103   9.18e- 1
##  5 ejection_fraction    -11.4        74.0  -0.154   8.78e- 1
##  6 high_blood_pressure  -45.7        70.1  -0.652   5.15e- 1
##  7 platelets             55.6        70.3   0.791   4.30e- 1
##  8 serum_creatinine       0.328      72.2   0.00454 9.96e- 1
##  9 serum_sodium         103.         71.3   1.45    1.50e- 1
## 10 sex                  128.         78.4   1.63    1.05e- 1
## 11 smoking             -130.         76.7  -1.70    9.07e- 2
## 12 time                 -26.7        81.6  -0.327   7.44e- 1
## 13 DEATH_EVENT          144.         86.1   1.67    9.70e- 2
```

```
lm_fit_cv <- fit_resamples(lm_model_wf, resamples =health_cv13, metrics = metric_set(rmse, mae, rsq))
```

```
lm_fit_cv %>%
  collect_metrics()
```

```
## # A tibble: 3 x 6
##   .metric .estimator     mean     n  std_err .config
##   <chr>   <chr>         <dbl> <int>    <dbl> <chr>
## 1 mae     standard   607.        13  54.8    Preprocessor1_Model1
## 2 rmse    standard   921.        13 136.     Preprocessor1_Model1
## 3 rsq     standard     0.0895    13   0.0251 Preprocessor1_Model1
```

```
lm_fit_test <- last_fit(lm_model_wf,
         split = data_split)

lm_fit_test %>%
  collect_metrics()
```

```
## # A tibble: 2 x 4
##   .metric .estimator    .estimate .config
##   <chr>   <chr>             <dbl> <chr>
## 1 rmse    standard   806.         Preprocessor1_Model1
## 2 rsq     standard     0.00000760 Preprocessor1_Model1
```

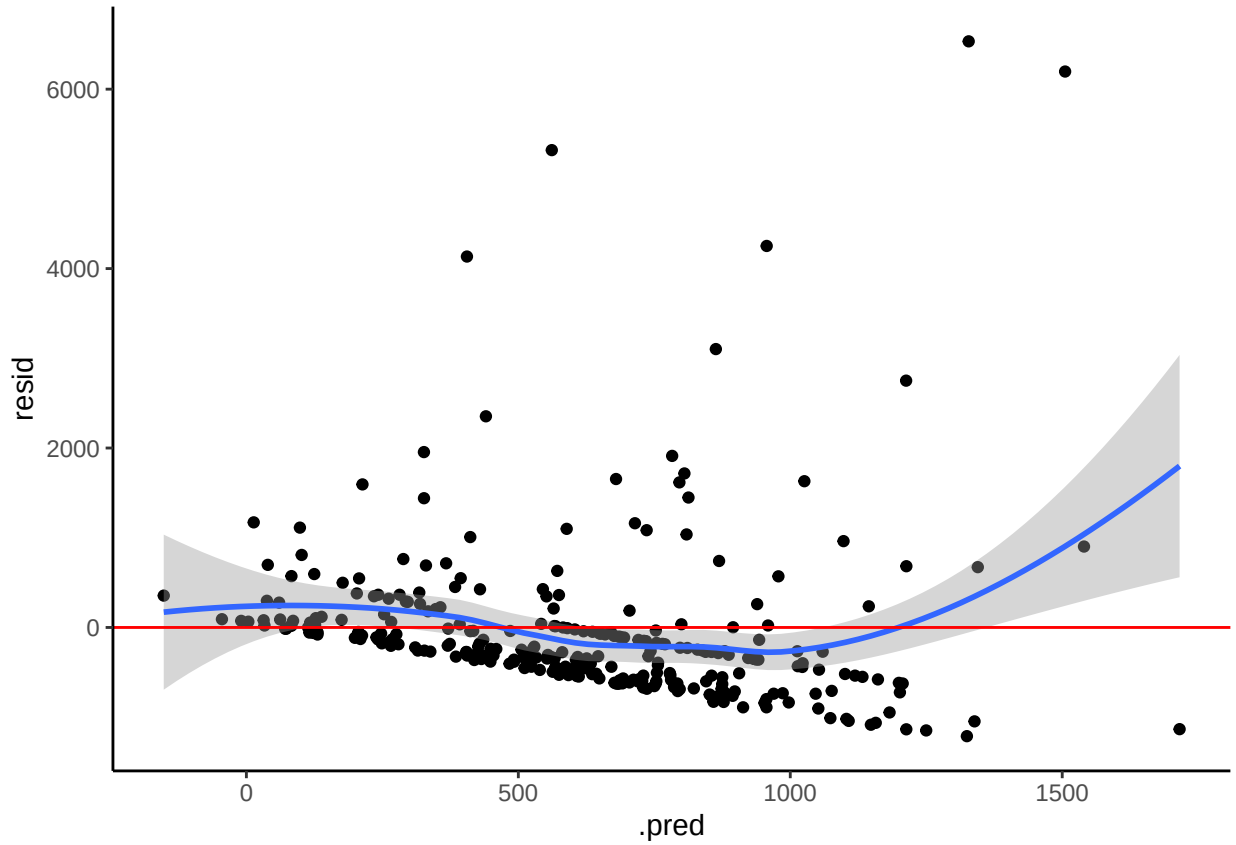
```
mod_ols <- fit_resamples(lm_model_wf,
      resamples = health_cv13,
      metrics = metric_set(rmse, rsq, mae)
      )  %>%
      collect_metrics(summarize = TRUE)
```

Results

Summarize your final model and justify your model choice (see below for ways to justify your choice).

The models show us that there is probably not a linear relationship between CPK and the predictors because the rsq is so low and the mae is so high. On average, we expect an error of 1 in our predictions, and considering that the highest CPK is 9 and the lowest is 3, we expect our predictions to be off by a lot. In additon, an rsq of 0.08 tells us that barely any of the CPK can be explained by our predictors(8%). At the very least, linear regression does not accurately represent the relationship between our predictors and CPK, and at worst, there isn't a relationship between our predictors and CPK.

```
mod_new <- lm_fit_train %>%
    predict(new_data = health1) %>%
    bind_cols(health1) %>%
    mutate(resid = creatinine_phosphokinase - .pred)

ggplot(mod_new, aes(x = .pred, y = resid)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = "red") +
    theme_classic()
```



Our analysis is more focused on interpretability than predictive accuracy as we are interested in which of our predictor variables have the greatest impact on a person's CPK (creatinine phosphokinase) levels rather than, for example, trying to predict a person's CPK levels based on other health indicators. High CPK levels usually indicate that there has been injury or stress to muscle tissue, the heart, or the brain. We want to see which health indicators have the strongest influence on damage to the body. We do not care to predict the CPK levels themselves as that does not carry much meaning for us.

We then used splines to break this data down further and in essence smooth the data by breaking it into "buckets" to help us fit the data better.

```
gam_rec <- recipe(creatinine_phosphokinase ~ high_blood_pressure + age + anaemia + platelets + sex + se

gam_spline_rec <- gam_rec %>%
   step_ns(high_blood_pressure, deg_free = 1) %>%
     step_ns(age, deg_free = 1) %>%
     step_ns(anaemia, deg_free = 1) %>%
     step_ns(platelets, deg_free = 1) %>%
     step_ns(sex, deg_free = 1) %>%
     step_ns(serum_sodium, deg_free = 1) %>%
     step_ns(ejection_fraction, deg_free = 1)

gam_spline_wf <- workflow() %>%
    add_model(lm_spec) %>%
    add_recipe(gam_spline_rec)
```

```
# fit model
gam_spline_fit_train <- gam_spline_wf %>%
  fit( data = health_train)

gam_spline_fit_train %>%
  tidy()
```

```
## # A tibble: 8 x 5
##   term                      estimate std.error statistic p.value
##   <chr>                        <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)                  475.       446.     1.06  0.289
## 2 high_blood_pressure_ns_1     -79.3      181.    -0.439 0.661
## 3 age_ns_1                    -576.       394.    -1.46  0.145
## 4 anaemia_ns_1                -502.       173.    -2.91  0.00403
## 5 platelets_ns_1               340.       653.     0.520 0.603
## 6 sex_ns_1                     164.       188.     0.875 0.382
## 7 serum_sodium_ns_1            837.       700.     1.20  0.233
## 8 ejection_fraction_ns_1      -373.       483.    -0.773 0.440
```

```
training_prep %>%
  select(creatinine_phosphokinase) %>%
  bind_cols( predict(gam_spline_fit_train, health_train) ) %>%
  metrics(estimate = .pred, truth = creatinine_phosphokinase)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard   1000.
## 2 rsq     standard      0.0667
## 3 mae     standard    567.
```

```
mod_gam_spline <- gam_spline_wf %>%
  fit_resamples(resamples = health_cv13,
                metrics = metric_set(rmse, mae, rsq))

mod_gam_spline %>%
  collect_metrics()
```

```
## # A tibble: 3 x 6
##   .metric .estimator     mean     n  std_err .config
##   <chr>   <chr>         <dbl> <int>    <dbl> <chr>
## 1 mae     standard   582.        13  56.1    Preprocessor1_Model1
## 2 rmse    standard   885.        13 143.     Preprocessor1_Model1
## 3 rsq     standard     0.0867    13   0.0243 Preprocessor1_Model1
```

Conclusions

Interpret you final model (show plots of estimated non-linear functions, or slope coefficients) for important predictors, and provide some general interpretations of what you learn from these. Interpret evaluation metric(s) for the final model in context with units. Does the model show an acceptable amount of error?

Summarization should show evidence of acknowledging the data context in thinking about the sensibility of these results.

We found that age, sex, sodium and platelets are the biggest predictors of CPK. This makes sense because sodium and platelets affect the amount of oxygen in the bloodstream, and would affect someone's CPK. Additionally, it makes sense that age and sex are important predictors because the older someone is, the more likely they are to have heart failure and women are more likely to experience heart failure than men. At the moment, the methods that we've applied to reach a consensus on which variables are most important makes sense because when we used LASSO, the algorithm penalized all of the variables that do not add much to the model, leaving us with the ones that do. We were a little surprised at first that sex was a significant predictor because it seemed at first to be exogenous to whether or not someone experienced heart failure. In looking at our evaulation metrics, the r-squared of 0.53 means the model is explaining 53% of the variance in our model. This makes sense in context our goal within our analysis is to predict death event, so using regression methods, whether they be OLS, nonlinear models or LASSO are not explaining the relationship between all of the quantitative variables we are analyzing as they are all are more related to death event than each other.

Classification

Methods

The two different methods that we used to answer our classification research question were logistic regression and decision trees.

We estimated all of our quantitative evaluation metrics using cross validation and the out of bag error.

To evaluate the models, we used cross validation and the out of bag error. For cross validation, we randomly split a data set into two groups that are called folds, given by k. For each fold, the data is divided randomly into a training set and test set, k-1 folds. The training data is used to train the model to make predictions about the data and the test set is used to see how well the model works on data it has not seen before, allowing us to test its accuracy. The resulting metrics from cross validation are averaged over the number of folds to get a result that encapsulates all the folds and tests that were done. For the out of bag error, the algorithm uses bootstrapping which is where the algorithm randomly resamples the dataset to generate the model. For the out of bag error, the algorithm randomly leaves out some cases out of the resampling and uses these left out cases to test the model and determine the accuracy of the decision trees.

Our data set includes data on different health factors that influence the probability of a person dying from heart failure. It makes logical sense that the classification models that we are creating for this section of the project will generate the most valuable results from the data set and will be the most important in regards to the overall context of our research investigations.

Decision trees

```
health1$DEATH_EVENT <- as.factor(health1$DEATH_EVENT)

rf_spec <- rand_forest() %>%
  set_engine(engine = 'ranger') %>%
  set_args( floor(sqrt(13)),
            trees = 1000,
            min_n = 2,
            probability = FALSE,
            importance = 'impurity') %>%
  set_mode('classification')

data_rec <- recipe(DEATH_EVENT ~ ., data = health1)
```

```
data_wf <- workflow() %>%
  add_model(rf_spec ) %>%
  add_recipe(data_rec)

data_fit <- fit(data_wf, data = health1)
```

Logistic Regression

```
health1 <- health1 %>%
mutate(DEATH_EVENT = relevel(factor(DEATH_EVENT), ref= '0'))

health_cv13 <- vfold_cv(health1, v = 13)

logistic_spec <- logistic_reg() %>%
  set_engine('glm') %>%
  set_mode('classification')

logistic_rec <- recipe(DEATH_EVENT ~ ., data = health1) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_dummy(all_nominal_predictors())

log_wf <- workflow() %>%
  add_recipe(logistic_rec)%>%
  add_model(logistic_spec)

log_fit <- fit(log_wf, data = health1)


log_fit %>% tidy() %>%
  mutate(OR.conf.low = exp(estimate- 1.96*std.error),OR.conf.high = exp(estimate + 1.96*std.error)) %>%
  mutate(OR = exp(estimate))
```

```
## # A tibble: 13 x 8
##    term     estimate std.error statistic  p.value OR.conf.low OR.conf.high    OR
##    <chr>       <dbl>     <dbl>     <dbl>    <dbl>       <dbl>        <dbl> <dbl>
##  1 (Interc~ -1.33        0.200   -6.65   3.03e-11       0.179        0.392 0.264
##  2 age       0.564       0.188    3.00   2.69e- 3       1.22         2.54  1.76
##  3 anaemia  -0.00371     0.179   -0.0207 9.83e- 1       0.702        1.41  0.996
##  4 creatin~  0.216       0.173    1.25   2.12e- 1       0.884        1.74  1.24
##  5 diabetes  0.0717      0.174    0.413  6.79e- 1       0.765        1.51  1.07
##  6 ejectio~ -0.907       0.193   -4.69   2.67e- 6       0.276        0.589 0.404
##  7 high_bl~ -0.0491      0.172   -0.286  7.75e- 1       0.680        1.33  0.952
##  8 platele~ -0.117       0.185   -0.635  5.25e- 1       0.619        1.28  0.889
##  9 serum_c~  0.689       0.188    3.67   2.42e- 4       1.38         2.88  1.99
## 10 serum_s~ -0.296       0.175   -1.69   9.19e- 2       0.528        1.05  0.744
## 11 sex      -0.255       0.198   -1.29   1.97e- 1       0.526        1.14  0.775
## 12 smoking  -0.00631     0.193   -0.0327 9.74e- 1       0.681        1.45  0.994
## 13 time     -1.63        0.234   -6.98   2.92e-12       0.123        0.309 0.195
```

Predictions on the Logistic Model

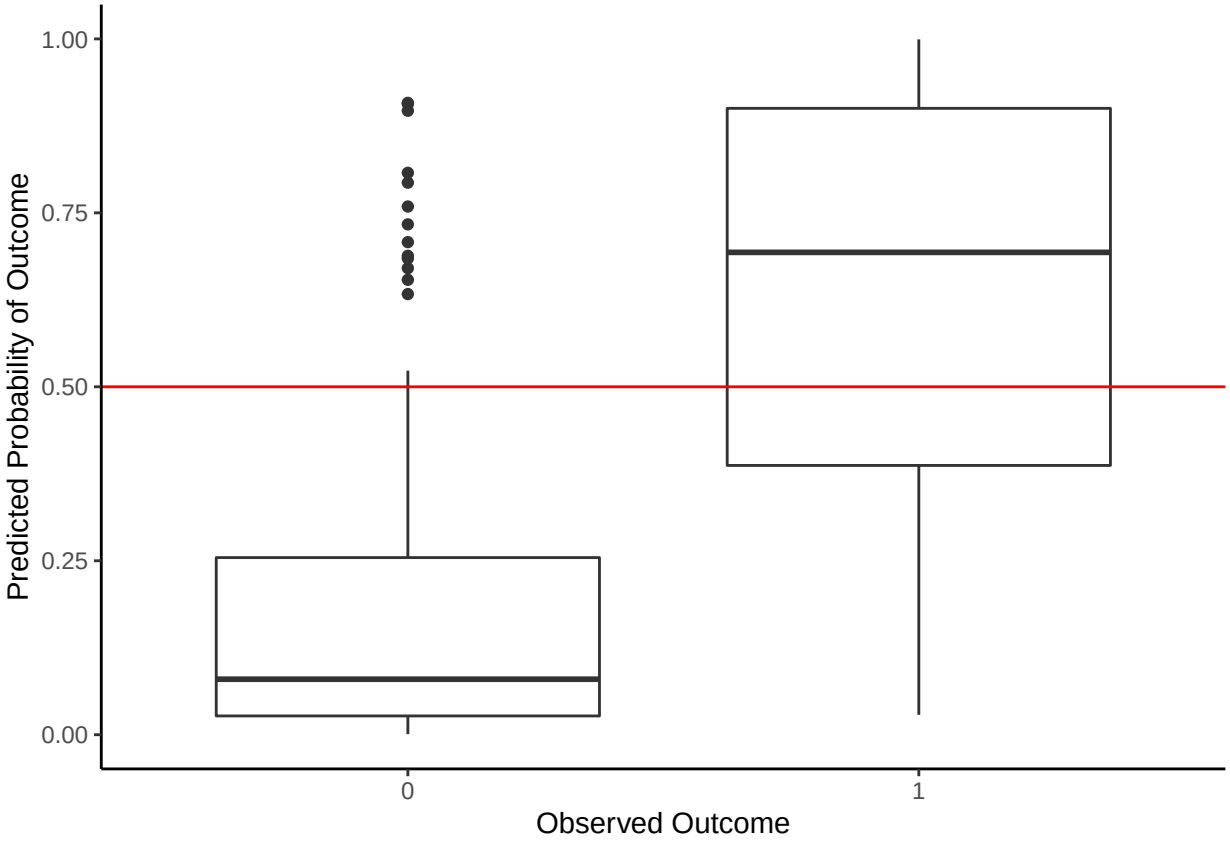
```
predict(log_fit, new_data = health1, type = "prob")
```

```
## # A tibble: 299 x 2
##     .pred_0 .pred_1
##       <dbl>   <dbl>
##  1 0.0207     0.979
##  2 0.0933     0.907
##  3 0.0430     0.957
##  4 0.100      0.900
##  5 0.00461    0.995
##  6 0.0530     0.947
##  7 0.0334     0.967
##  8 0.670      0.330
##  9 0.611      0.389
## 10 0.000647   0.999
## # ... with 289 more rows
```

```
predict(log_fit, new_data = health1, type = "class")
```

```
## # A tibble: 299 x 1
##    .pred_class
##    <fct>
##  1 1
##  2 1
##  3 1
##  4 1
##  5 1
##  6 1
##  7 1
##  8 0
##  9 0
## 10 1
## # ... with 289 more rows
```

```
logistic_output <-  health1 %>%
  bind_cols(predict(log_fit, new_data = health1, type = 'prob'))

logistic_output <- logistic_output %>%
  mutate(.pred_class = make_two_class_pred(`.pred_0`, levels(DEATH_EVENT), threshold = .5))

logistic_output %>%
  ggplot(aes(x = DEATH_EVENT, y = .pred_1)) +
  geom_boxplot() +
  geom_hline(yintercept = 0.5, color='red') +
  labs(y = 'Predicted Probability of Outcome', x = 'Observed Outcome') +
  theme_classic()
```



Results

Summarize your final model and justify your model choice (see below for ways to justify your choice).Compare the different classification models tried in light of evaluation metrics, variable importance, and data context. Display evaluation metrics for different models in a clean, organized way. This display should include both the estimated metric as well as its standard deviation. (This won't be available from OOB error estimation. If using OOB, don't worry about reporting the SD.)Broadly summarize conclusions from looking at these evaluation metrics and their measures of uncertainty.

In our final model we predicted the probability of someone dying using logistic regression. We chose decision trees over logistic regression because it has a higher likelihood it will correctly predict death at an accuracy rate of 85.61%; compared to our logistic regression model which has an accuracy of 82.27% on predicting death event. Since the sensitivity for the logistic regression is low, it is worse at predicting whether or not they will die, which is another reason why we've chosen the decision trees. Although the ROC AUC curve for logistic regression is 88%, it is still not the same as the actual prediction of the death event, which is lower than logistic regression.For the decision trees, the predictors that had the biggest impact on the final result of predicting death event are ejection fraction, cpk, age and serum creatinine. From looking at our evaulation metrics we can conclude that

OOB

```
rf_OOB_output <- function(fit_model, model_label, truth){
    tibble(
          .pred_class = fit_model %>% extract_fit_engine() %>% pluck('predictions'), #OOB predictions
          class = truth,
          label = model_label
```

```
      )
}


rf_OOB_output(data_fit, sqrt(13), health1 %>% pull(DEATH_EVENT))
```

```
## # A tibble: 299 x 3
##    .pred_class class label
##    <fct>       <fct> <dbl>
##  1 1           1      3.61
##  2 1           1      3.61
##  3 1           1      3.61
##  4 1           1      3.61
##  5 1           1      3.61
##  6 1           1      3.61
##  7 1           1      3.61
##  8 1           1      3.61
##  9 1           1      3.61
## 10 1           1      3.61
## # ... with 289 more rows
```

```
data_rf_OOB_output <- bind_rows(
    rf_OOB_output(data_fit, sqrt(13), health1 %>% pull(DEATH_EVENT)))

data_rf_OOB_output %>%
    group_by(label) %>%
    accuracy(truth = class, estimate = .pred_class)
```

```
## # A tibble: 1 x 4
##   label .metric  .estimator .estimate
##   <dbl> <chr>    <chr>          <dbl>
## 1  3.61 accuracy binary         0.849
```

```
rf_OOB_output(data_fit,12, health1 %>% pull(DEATH_EVENT)) %>%
    conf_mat(truth = class, estimate= .pred_class)
```

```
##           Truth
## Prediction   0   1
##          0 185  27
##          1  18  69
```

logistic regression cross validation

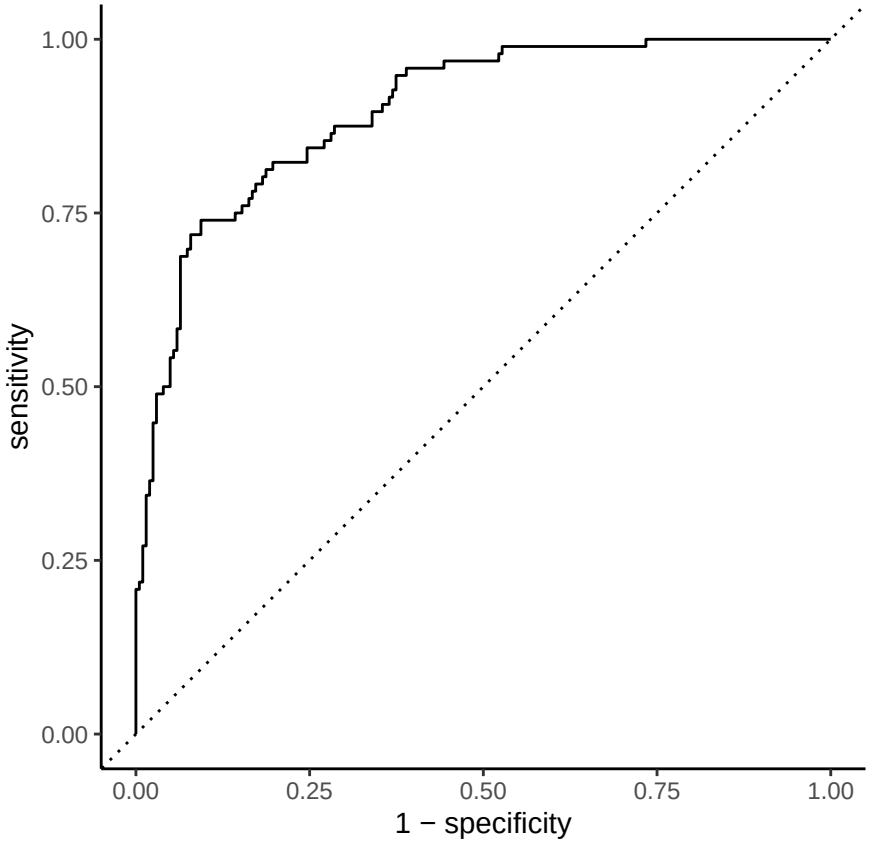
```
logistic_output %>%
  conf_mat(truth = DEATH_EVENT, estimate = .pred_class)
```

```
##           Truth
## Prediction   0   1
##          0 187  27
##          1  16  69
```

```
log_metrics <- metric_set(sens, yardstick::spec, accuracy)

logistic_output %>%
  log_metrics(estimate = .pred_class, truth = DEATH_EVENT, event_level = "second")
```

```
## # A tibble: 3 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 sens     binary         0.719
## 2 spec     binary         0.921
## 3 accuracy binary         0.856
```

```
logistic_roc <- logistic_output %>%
    roc_curve(DEATH_EVENT, .pred_1, event_level = "second")

autoplot(logistic_roc) + theme_classic()
```




```
log_cv_fit <- fit_resamples(
    log_wf,
    resamples = health_cv13,
    metrics = metric_set(sens, yardstick::spec, accuracy, roc_auc),
    control = control_resamples(save_pred = TRUE, event_level = 'second'))

collect_metrics(log_cv_fit)
```

```
## # A tibble: 4 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.819    13  0.0230 Preprocessor1_Model1
## 2 roc_auc  binary     0.874    13  0.0181 Preprocessor1_Model1
## 3 sens     binary     0.683    13  0.0315 Preprocessor1_Model1
## 4 spec     binary     0.882    13  0.0278 Preprocessor1_Model1
```

Conclusions

Interpret evaluation metric(s) for the final model in context. Does the model show an acceptable amount of error? If using OOB error estimation, display the test (OOB) confusion matrix, and use it to interpret the strengths and weaknesses of the final model.Summarization should show evidence of acknowledging the data context in thinking about the sensibility of these results.

The model does not show an acceptable amount of error because if was told that the prediction of a death event (or not dying) was 85% accurate, it is quite low. In our confusion matrix, there were 187 people whose that were predicted to not happen and did not happen, 26 people whose deaths we predicted to not happen but happened, 17 people who we predicted to die and did not die and 70 people who we predicted to die who died. Ideally, the model would have as close to 100% accuracy to predict a death event and we would prefer the model to predict that someone was going to die and they don't die versus the opposite. Our ROC AUC, our model is further evidence that our model does not explain the relationship between death event and our selected variables. Given the context of our data, our results are not sensible and our model should not be used to predict death event of these heart failure patients, since we have too much error to be predicting death.

Unsupervised learning clustering

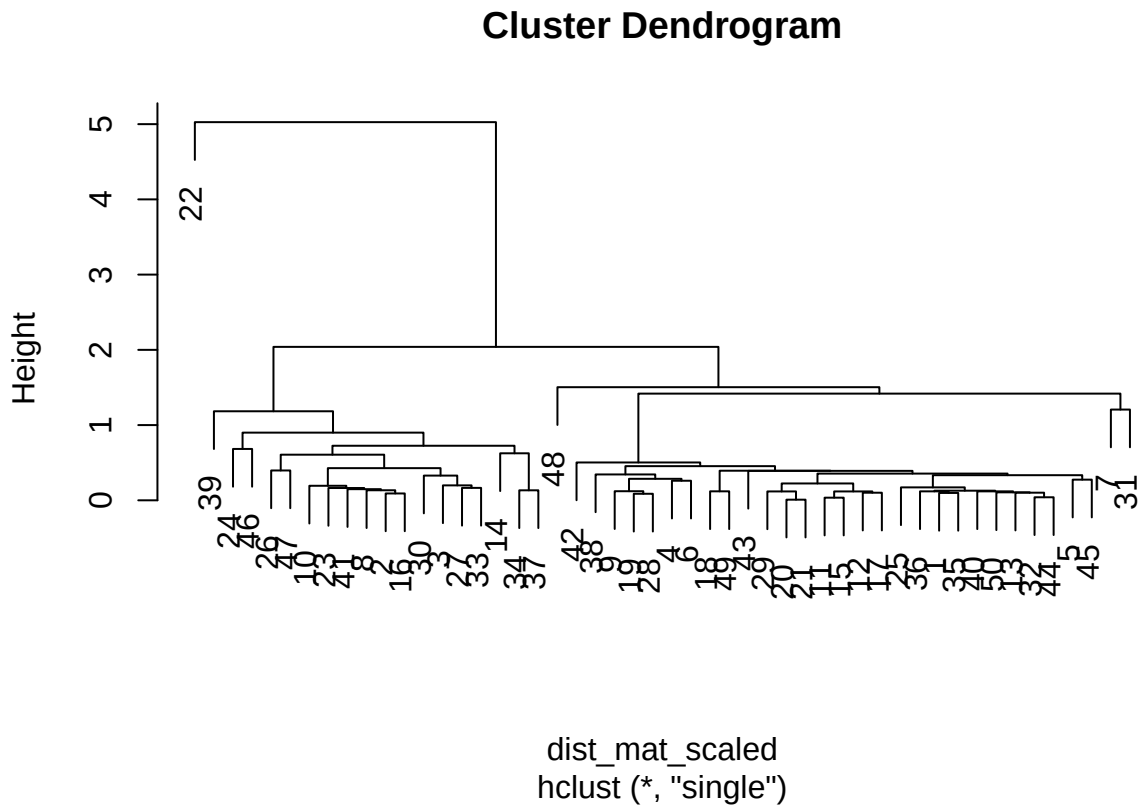
Choose one method for clustering. Justify the choice of features included in a distance measure based on the research goals. Justify the choice of k and summarize resulting clusters. Interpret the clusters qualitatively. Evaluate clusters quantitatively (kmeans: within cluster sum of squares, pam: silhouette, hclust: height of cut on dendrogram). If appropriate, show visuals to justify your choices.Summarize what information you gain from the clustering in context (tell a story)

In our consideration of what variables impacted the chances of a person dying from heart failure, we decided to use un supervised learning to see what the most common indicator of heart failure caused a death event. The method that we chose for clustering was hierarchical clustering. We picked this method of clustering over k-means and principle component analysis because we didn't know what the right number of k-clusters would be that would explain how to cluster our cases. Additionally, since our data set contains both meaningful binary and quantitative variables, it did not make sense to do dimension reduction on the data. Hierarchical clustering made the most sense because without guessing the number of clusters we would need, the distances between each cluster would tell us how to cluster DEATH_EVENT by a patient's cpk or platelet levels. We chose two of the quantitative variables that we used for other parts of our analysis, cpk and platelets and grouped whether or not a patient died, which resulted in 4 clusters. We used single linkage clustering to form our clusters because it made the most sentence to draw a connection between the clusters if we were using the shortest distance between points. The heights of the clusters from the resulting dendrogram indicate the distance between the clusters. The first set of clusters are 1 unit away from each other, showing they contain very similar information. The next set of clusters at a height of 2, shows the four clusters, based on death event and whether a patient met the threshold for cpk or platelets. The clusters at the height of 5 show whether a person died from heart failure. From these clusters we can group what patients died from heart failure if they had platelets or cpk.

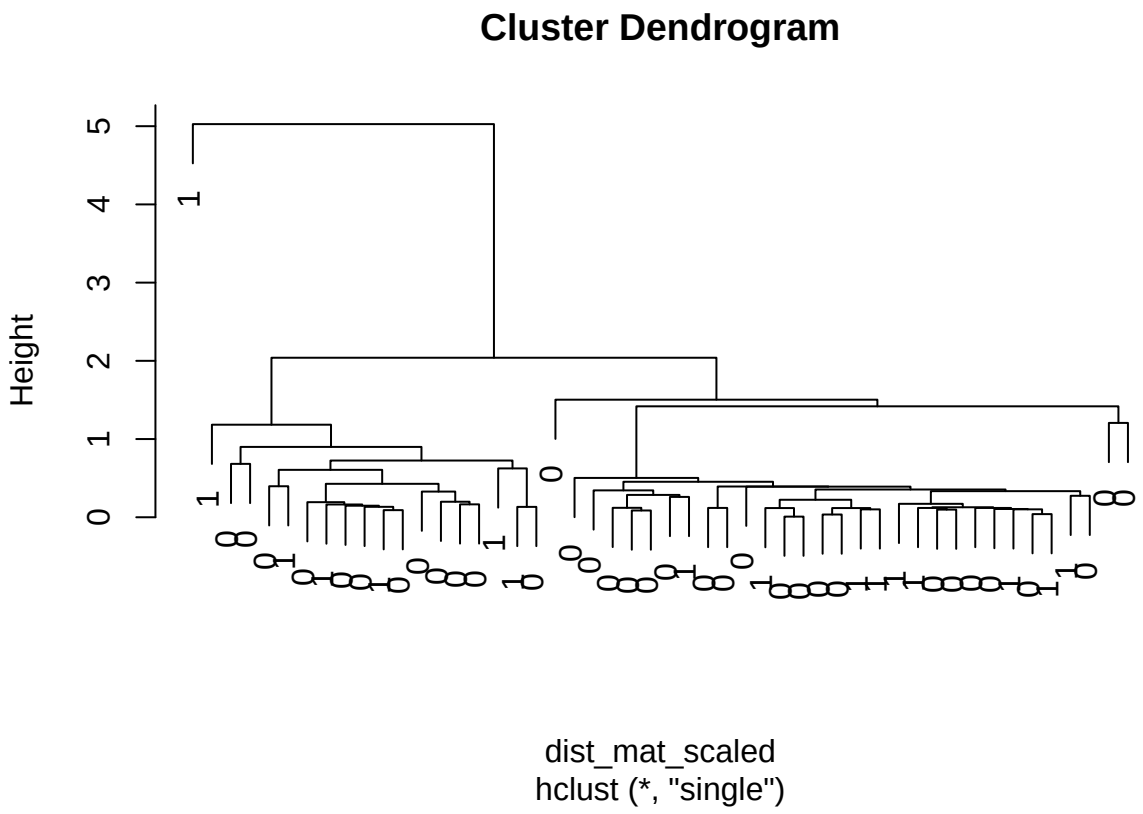
```
set.seed(253)
health2 <- health1 %>%
    slice_sample(n = 50)

health2_sub <- health2 %>%
    select(creatinine_phosphokinase, DEATH_EVENT, platelets)

health2_sub$DEATH_EVENT=as.numeric(health2_sub$DEATH_EVENT)
summary(health2_sub)
```

```
##  creatinine_phosphokinase  DEATH_EVENT     platelets
##  Min.   :  59.0           Min.   :1.00   Min.   : 25100
##  1st Qu.: 148.2           1st Qu.:1.00   1st Qu.:212250
##  Median : 312.0           Median :1.00   Median :262679
##  Mean   : 641.5           Mean   :1.38   Mean   :252558
##  3rd Qu.: 582.0           3rd Qu.:2.00   3rd Qu.:310750
##  Max.   :7702.0           Max.   :2.00   Max.   :418000
```

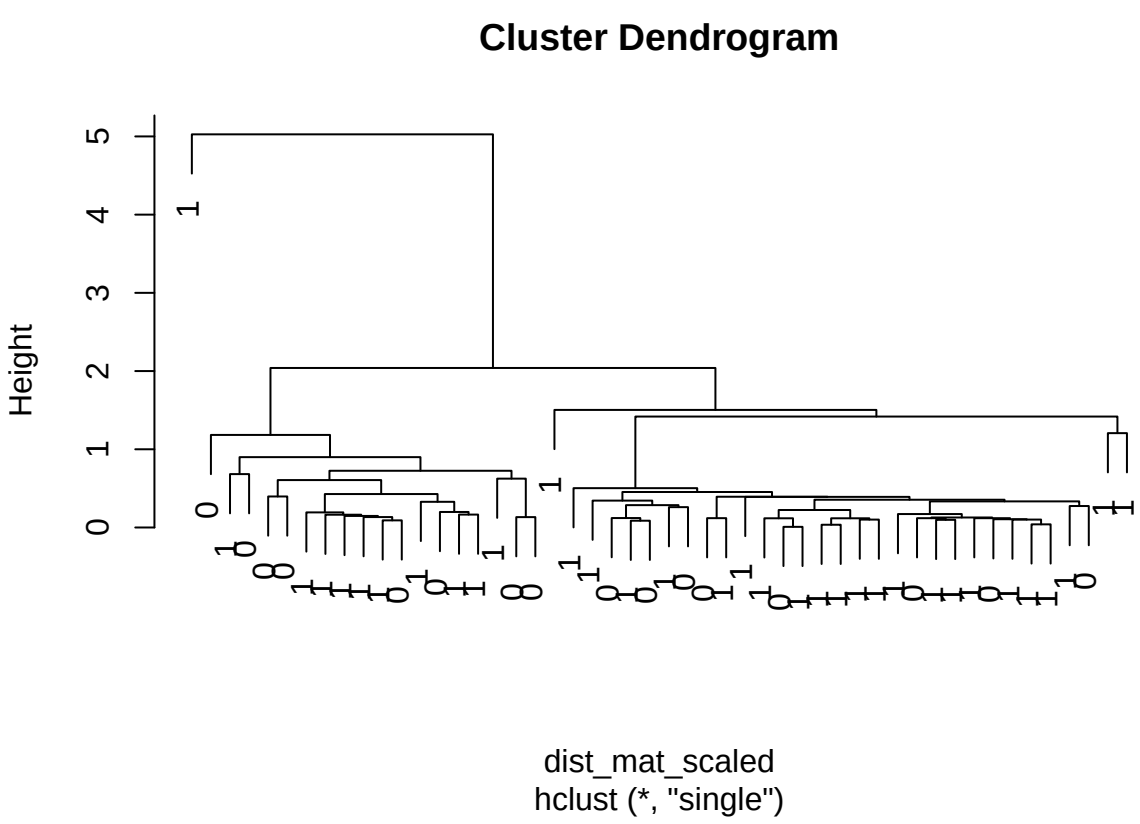
```
dist_mat_scaled <- dist(scale(health2_sub))

hc_single <- hclust(dist_mat_scaled, method = "single")

plot(hc_single)
```



```
plot(hc_single, labels = health2$high_blood_pressure)
```




```
plot(hc_single, labels = health2$sex)
```



We also made two more clusters that labeled the clusters on whether or not they had high blood pressure and sex of the patient. In the dendrogram that is labeled by blood pressure, the patient that died had high blood pressure. Most of the patients that survived did not have high blood pressure although a few of them did. So from this we can conclude that high blood pressure is correlated with having high blood pressure. In the dendrogram that is labeled by sex, there is no connection to which sex was more likely to die from heart failure.