Homework 2

Charles Hanks

01/18/2023

Directions:

Please turn in **both** a knitted HTML file *and* your Rmd file on WISE.

Good luck!

Setup (1pt)

Change the author of this RMD file to be yourself and modify the below code so that you can successfully load the 'wine.rds' data file from your own computer.

```
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE)
knitr::opts_knit$set(root.dir = "/Users/charleshanks/Desktop/MSDS/SPRING_23/ML")
library(tidyverse)
library(caret)
library(fastDummies)
wine = read_rds("wine.rds")
```

Feature Engineering (3pts)

1. Modify the below code and create a total of 10 features (including points).
2. Make sure that you remove all rows with a missing value.
3. Make sure that log(price) and your features are the only columns that remain in the wino dataframe.

Note: each item in a factor variable counts as one feature. I.e., each variety of wine counts as one feature, even though they are all in the same column within the dataframe.

```
vin = wine %>%
      mutate(lprice = log(price),
             critic = fct_lump(taster_name,3),
             before_y2k = year < 2000,
             winery = fct_lump(winery,4),
             estate = designation == "Estate") %>%
            select(lprice, points, critic, before_y2k, winery, estate) %>%
      drop_na(.)


vin = dummy_cols(vin, remove_selected_columns = T) %>%
select(-winery_Other, -critic_Other) %>%
rename_all(funs(tolower(.))) %>%
rename_all(funs(str_replace_all(., "-", "_"))) %>%
rename_all(funs(str_replace_all(., " ", "_")))
```

Caret (5pts)

1. Use the Caret library to partition the wino dataframe into an 80/20 split.
2. Then run a linear regression with bootstrap resampling.
3. Report RMSE when your model is run on the test partition of the data.

Hint: control <- trainControl(method=“boot”, number=5)

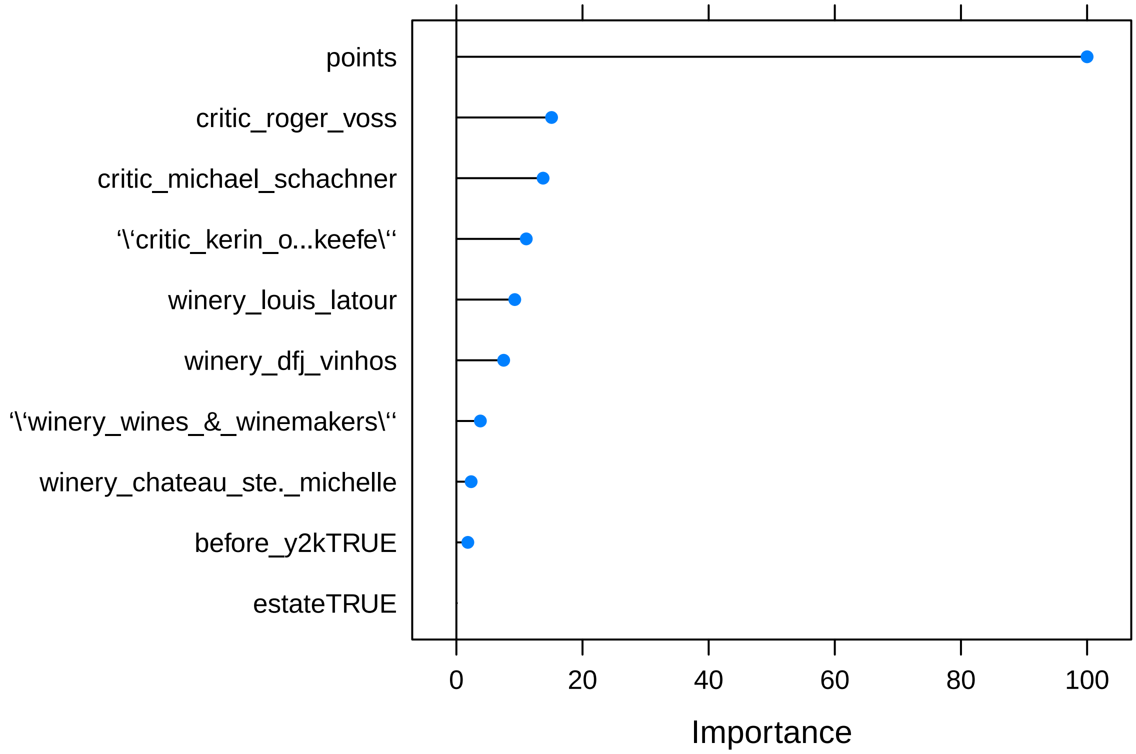
```
vin_index = createDataPartition(vin$lprice, p  = 0.8, list = FALSE)

vin_train <- vin[vin_index, ]
vin_test <- vin[-vin_index, ]

control <- trainControl(method="boot", number=5)

modelo <- train(lprice ~ .,
             data = vin_train,
             method = "lm",
             trControl = control)

vin_pred <- predict(modelo, vin_test)

postResample(pred = vin_pred, obs=vin_test$lprice)
```

```
##      RMSE  Rsquared       MAE
## 0.5044658 0.4144334 0.3955093
```

Variable selection (1pt)

Graph the importance of your 10 features.

```
importance <- varImp(modelo, scale=TRUE)
plot(importance)
```



(2pts)

Explain how the bootstrap method in train control you used differs from cross validation (see the link to feat.engineering in the slides).

Answer: The bootstrap resampling method in train control randomly picks observations with replacement. The size of this bootstrap sample will be the size of the original training data set. This means that an observation from the training data could be used 0, 1, or more than 1 times in training the model. Whereas the k-fold cross validation method samples without replacement, meaning that each observation is used exactly once.

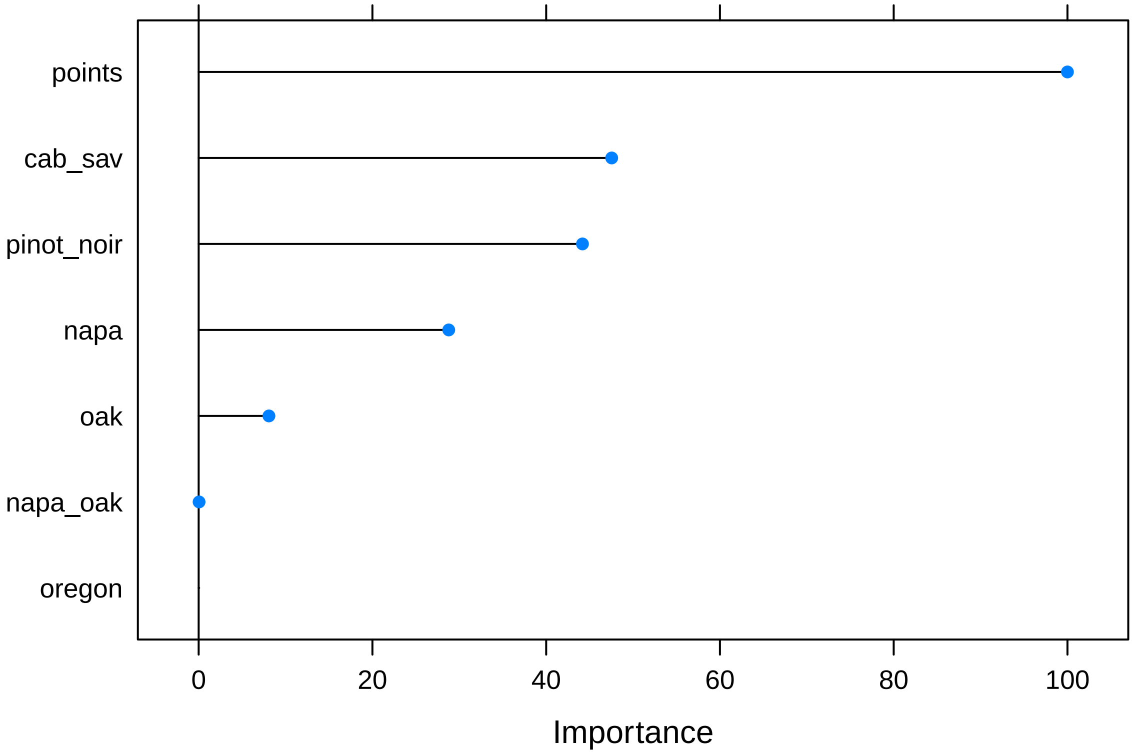
Bonus (3pts)

1. Execute ‘set.seed(504)’ prior to running your (training/test) data partition
2. Generate an RMSE on the test data of < 0.47 (1pt), < 0.46 (2pts), or < 0.45 (3pts)

```
set.seed(504)
vin2 = wine %>%
      mutate(lprice = log(price),
             country = fct_lump(country,2),
             oregon = as.integer(str_detect(province, "Oregon")),
             cab_sav = as.integer(variety == "Cabernet Sauvignon"),
             pinot_noir = as.integer(variety == "Pinot Noir"),
             napa = as.integer(str_detect(region_1, "[Nn]apa")),
             oak = as.integer(str_detect(description, "[Oo]ak")),
             champ = as.integer(str_detect(province, "Champagne")),
             napa_oak = napa*oak) %>%
            drop_na(.) %>%
            select(lprice, points,country,oregon,cab_sav,pinot_noir,champ,napa,oak,napa_oak) %>%
            dummy_cols(remove_selected_columns = T) %>%
              select(-country_Other)

vin2_index = createDataPartition(vin2$lprice, p  = 0.8, list = FALSE)

vin2_train <- vin2[vin2_index, ]
vin2_test <- vin2[-vin2_index, ]

control <- trainControl(method="boot", number=5)

modelo2 <- train(lprice ~ .,
             data = vin2_train,
             method = "lm",
             trControl = control)

vin2_pred <- predict(modelo2, vin2_test)

postResample(pred = vin2_pred, obs=vin2_test$lprice)
```

```
##      RMSE  Rsquared       MAE
## 0.4213662 0.3845748 0.3278375
```

```
#Rstudio throws warning: prediction from a rank deficient fit may be misleading.
#Perhaps 2 predictors highly correlated?

importance <- varImp(modelo2, scale=TRUE)
plot(importance)
```


points
cab_sav
pinot_noir
napa
oak
napa_oak
oregon
0
20
40
60
80
100
Importance