Andrea Hassler
Madison Volpe
MDML Homework 5

Question One

A. Create one training set and two validation sets in the following manner. Restrict **sqf** to years 2013-2014, randomly shuffle the data, and split it in half. Call one half **train_half**, and the other half **test_half**. Next, restrict **sqf** to just the year 2015, and call this **test_later**. Remove the stop id and year columns from **train_half**, **test_half**, and **test_later**.

```
> str(train_half)
Classes 'tbl_df', 'tbl' and 'data.frame':       29367 obs. of  111 variables:
 $ found.weapon            : Factor w/ 2 levels "0","1": 1 1 1 2 1 2 1 1 1 1 ...
 $ location.housing        : Factor w/ 3 levels "housing","neither",..: 2 2 2 2 2 3 1 1 2 2 .
 $ stopped.bc.bulge        : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 2 1 2 1 1 1 2 ...
 $ stopped.bc.object       : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 2 1 2 1 1 1 1 ...
 $ stopped.bc.desc         : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 2 1 1 1 1 1 ...
 $ stopped.bc.casing       : Factor w/ 2 levels "FALSE","TRUE": 1 1 2 1 1 1 1 1 1 1 ...
 $ stopped.bc.clothing     : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ stopped.bc.drugs        : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ stopped.bc.furtive      : Factor w/ 2 levels "FALSE","TRUE": 2 1 2 1 2 1 2 2 2 2 ...
 $ stopped.bc.lookout      : Factor w/ 2 levels "FALSE","TRUE": 1 2 2 1 1 1 1 1 1 1 ...
 $ stopped.bc.other        : Factor w/ 2 levels "FALSE","TRUE": 1 2 1 1 2 1 1 1 1 1 ...
 $ stopped.bc.violent      : Factor w/ 2 levels "FALSE","TRUE": 1 1 2 1 1 1 1 1 1 2 ...
 $ additional.associating  : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ additional.direction    : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ additional.evasive      : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 2 1 1 2 1 2 ...
 $ additional.highcrime    : Factor w/ 2 levels "FALSE","TRUE": 1 1 2 2 2 2 2 2 1 2 ...
 $ additional.investigation: Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ additional.report       : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
```

```
> str(test_later)
Classes 'tbl_df', 'tbl' and 'data.frame':       6728 obs. of  112 variables:
 $ found.weapon            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ location.housing        : Factor w/ 3 levels "housing","neither",..: 2 2 2 2 1 2 2 2 2 2 ...
 $ stopped.bc.bulge        : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 2 2 1 1 1 1 1 ...
 $ stopped.bc.object       : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ stopped.bc.desc         : Factor w/ 2 levels "FALSE","TRUE": 2 1 2 1 2 2 2 1 1 1 ...
 $ stopped.bc.casing       : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ stopped.bc.clothing     : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ stopped.bc.drugs        : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ stopped.bc.furtive      : Factor w/ 2 levels "FALSE","TRUE": 2 2 1 2 2 1 1 1 1 1 ...
 $ stopped.bc.lookout      : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ stopped.bc.other        : Factor w/ 2 levels "FALSE","TRUE": 2 1 1 1 1 1 1 2 2 2 ...
 $ stopped.bc.violent      : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ additional.associating  : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ additional.direction    : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 2 1 2 1 1 1 1 ...
 $ additional.evasive      : Factor w/ 2 levels "FALSE","TRUE": 2 1 1 1 1 1 1 1 1 1 ...
 $ additional.highcrime    : Factor w/ 2 levels "FALSE","TRUE": 1 2 1 2 2 1 2 1 1 1 ...
 $ additional.investigation: Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
```

```
> str(test_half)
Classes 'tbl_df', 'tbl' and 'data.frame':       29368 obs. of  112 variables:
 $ found.weapon            : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
 $ location.housing        : Factor w/ 3 levels "housing","neither",..: 1 3 1 1 1 2 2 2 2 2 ...
 $ stopped.bc.bulge        : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 2 1 2 2 1 ...
 $ stopped.bc.object       : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ stopped.bc.desc         : Factor w/ 2 levels "FALSE","TRUE": 1 1 2 1 1 1 1 1 1 1 ...
 $ stopped.bc.casing       : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ stopped.bc.clothing     : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 2 1 1 ...
 $ stopped.bc.drugs        : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ stopped.bc.furtive      : Factor w/ 2 levels "FALSE","TRUE": 1 2 1 2 2 2 2 2 2 2 ...
 $ stopped.bc.lookout      : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 2 1 1 1 ...
 $ stopped.bc.other        : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ stopped.bc.violent      : Factor w/ 2 levels "FALSE","TRUE": 2 1 1 1 1 1 2 1 1 1 ...
 $ additional.associating  : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
 $ additional.direction    : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 2 2 2 1 1 ...
 $ additional.evasive      : Factor w/ 2 levels "FALSE","TRUE": 2 1 1 1 1 2 2 1 1 1 ...
```

Andrea Hassler
Madison Volpe
MDML Homework 5

B. Fit a random forest model on **train_half** using the *randomForest* package in R, predicting found.weapon as a function of all features. Use 200 trees, but all other options for the model can be the default options.

```
Call:
 randomForest(formula = found.weapon ~ ., data = train_half, ntree = 200)
               Type of random forest: classification
                     Number of trees: 200
No. of variables tried at each split: 10

        OOB estimate of  error rate: 5.95%
Confusion matrix:
      0   1 class.error
0 27168 193 0.007053836
1  1555 451 0.775174477
```

C. Generate predicted *probabilities* using the model from partB for both **test_half** and **test_later**. Compute the AUC of the model on each test set. Write a paragraph describing and interpreting your results. In particular, discuss the following three questions.

AUC of test_half: **85.25449**
AUC of test_later: **75.73225**

a. Why do you think the AUC on **test_half** is noticeably higher than the AUC on **test_later**?

The AUC on **test_half** is noticeably higher than the AUC on **test_later** because **test_half** comes from the same subset of data as **train_half**, therefore are responses from both 2013 and 2014. However, **test_later** uses responses only from 2015.

b. If you were planning to use this model to guide how officers make stops in the future (e.g., by having officers use the model to compute the probability that an individual suspected of criminal possession of a weapon will have a weapon, and then only making a stop if the model-estimated probability is sufficiently high), would the AUC on **test_half** or **test_later** be a better estimate of performance on unseen data?

The AUC from **test_later** would be a better estimate of performance on unseen data because **test_later** is based off data from a different year. The **test_half** AUC is an inferior choice because our model is overfitting the **test_half**

data.

c. More generally, when evaluating a model using a simple training/validation split approach, should you always do the split by shuffling and splitting randomly?

Splitting the data by a random shuffle and split can be a good way to create your training and testing sets, as it keeps the characteristics of the two sets as similar as possible. This means your training set should produce a model that works very well for the test set. However, when you are dealing with data that may change over time, and you are attempting to create a model to predict on data from the future, it may be more beneficial to train your model on early data and then test on the most recent data. Assuming the most recent data is more similar to future data, this ensures that you are assessing how well your model will perform on future data, not how well it makes predictions on a mix of old data.

Andrea Hassler
Madison Volpe
MDML Homework 5

Question two

A. Clean **all_data**
   - Consult r script

B. Create the sample of data that you will use for prediction as a tibble called **restaurant_data**.
   - Consult r script

C. Perform some feature engineering. We will only create features that could be known *before* a given initial cycle inspection takes place.
   - Consult r script

D. Create a training set of all initial cycle inspections in 2015 and 2016 (**train**), and a testing set of all initial cycle inspections in 2017 (**test**). Fit a standard logistic regression model on the training set, predicting outcome as a function of only cuisine, borough, month, and weekday. Compute the AUC of this model on the **test** dataset.
   - The AUC for the logistic regression model comes out to **61.18.**

```
> summary(log_model)

Call:
glm(formula = outcome ~ cuisine + borough + month + weekday,
    family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8623  -0.5182  -0.4510  -0.3855   2.6968

Coefficients:
                                                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                                                     -1.885873   0.256580  -7.350 1.98e-13 ***
cuisineAmerican                                                 -1.010015   0.239833  -4.211 2.54e-05 ***
cuisineAsian                                                    -0.519862   0.270513  -1.922 0.054635 .
cuisineBagels/Pretzels                                          -0.629670   0.298393  -2.110 0.034841 *
cuisineBakery                                                   -0.722257   0.254813  -2.834 0.004590 **
cuisineBangladeshi                                               0.103704   0.377831   0.274 0.783721
cuisineBarbecue                                                 -0.798047   0.444076  -1.797 0.072320 .
cuisineBottled beverages, including water, sodas, juices, etc.  -1.520379   0.455350  -3.339 0.000841 ***
cuisineCaf\xe9/Coffee/Tea                                       -0.967469   0.249245  -3.882 0.000104 ***
cuisineCaribbean                                                -0.469414   0.253070  -1.855 0.063613 .
cuisineChicken                                                  -0.920947   0.273706  -3.365 0.000766 ***
cuisineChinese                                                  -0.473669   0.241744  -1.959 0.050067 .
cuisineChinese/Japanese                                         -0.141299   0.400329  -0.353 0.724120
```

Andrea Hassler
Madison Volpe
MDML Homework 5

E. Fit a random forest model on **train**, predicting outcome as a function of cuisine, borough, month, weekday, and the four historical features created in Step C. Use 1000 trees, but other settings can have default values. Compute the AUC of this model on the **test** dataset. How does the AUC of the random forest compare with the AUC of the logistic regression model?

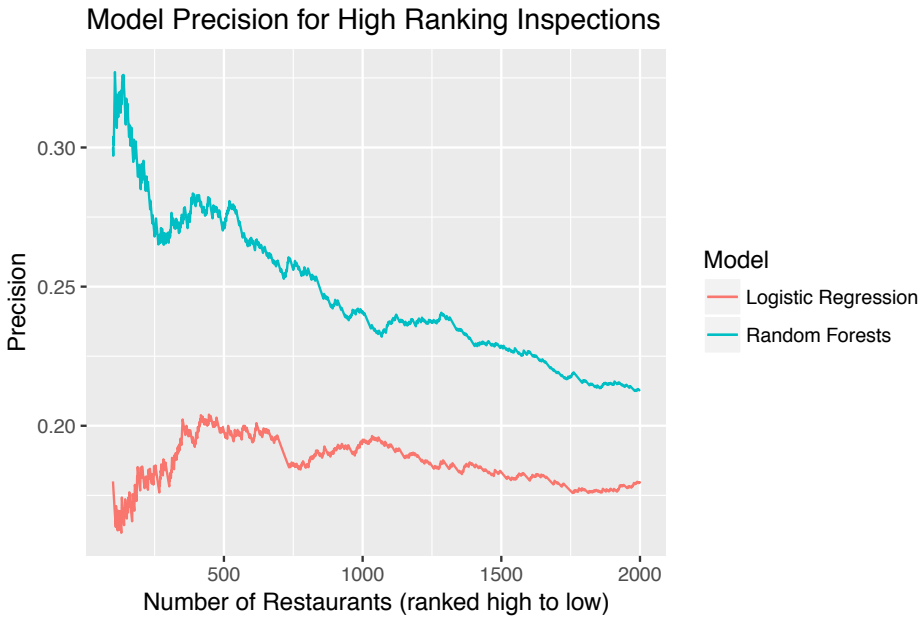
- The AUC for the random forest model comes out to **60.93.** The AUC for the random forest model is slightly smaller than the AUC of the logistic regression model. Both models have a fairly low AUC; neither performs exceptionally well on the test data.

```
> rf_model

Call:
 randomForest(formula = outcome ~ cuisine + borough + month +      weekday + num_previou
s_low_inspections + num_previous_med_inspections +      num_previous_high_inspections +
num_previous_closings, data = train,      ntree = 1000)
               Type of random forest: classification
                     Number of trees: 1000
No. of variables tried at each split: 2

        OOB estimate of  error rate: 10.94%
Confusion matrix:
      0  1  class.error
0 32526 25 0.0007680256
1  3970  7 0.9982398793
```

F. Generate a precision plot that compares the performance of the logistic regression and random forest models on just the highest ranked inspections.



Andrea Hassler
Madison Volpe
MDML Homework 5

G. Write a paragraph or two about your results. Based on the AUCs you computed in steps D) and E), would you choose one model over the other? What about based on the precision plots? How can two models both have fairly low AUC, but one has much higher precision on the highest ranked inspections?

It would be difficult to choose one model over the other based on AUC alone; the AUC was nearly the same and fairly low for both at 61.18 for logistic regression and 60.93 for random forests. However, the models had very different precision measures across the highest ranked inspections. Across the top 100 to 2000 restaurants ranked in descending order of probability of scoring at 28 or above, the precision for the logistic regression model peaked at just over 0.20 for about the top 450 restaurants and then decreased to about 0.10 over the remaining restaurant counts. Across the same restaurant counts, the precision for the random forests model peaked at about 0.35 at about the top 200 restaurants, and then decreased to about 0.25 over the remaining counts. Based on precision, the random forests model would be the chosen model, as it outperforms the logistic regression on the highest ranked inspections.

The high precision measure on the highest ranked inspections reflects that when the random forests model identifies an inspection of having a high probability of a high score, it is much more likely to be correct than when the logistic model identifies a high probability. However, the AUC also takes into account measures of true and false negatives. Both models had many false negatives, so for inspections with low predicted probabilities, the models were both far less accurate in their predictions. This is why both models had a fairly low AUC, but random forests did better on precision for the highest ranked inspections.

H. H. Finally, write a paragraph or two (or more!) about any possible ethical issues involved in using such a predictive model to prioritize restaurant inspections.

As with anything in life, nothing is completely objective meaning that there is always special cases. We can never characterize all similar cases into one box, there will always be exceptions and for this reason using predictive models can cause ethical issues.

The case of using DOHMH data from previous restaurant inspections in a predictive model to prioritize future restaurant inspections is not without its own unique ethical concerns. In this case, ethical concerns can arise from data integrity/clarity, whether our model is correctly specified, and whether the outcome of interest is appropriate for our cause.

Andrea Hassler
Madison Volpe
MDML Homework 5

The DOHMH data is a viable source as it comes from a reputable agency, however this does not mean it is without error. On the NYC open data website, the agency even admits that because the dataset is compiled from several large administrative data systems, that it can contain illogical values as a result from data transfers or data entry errors. Also the agency admits to missing data. "Messy" data is a fact of life when working with almost any large dataset, the data can be cleaned, but we can be losing important information when cleaning the data that will not be reflected in the model.

Likewise, missing data also poses a challenge, imputation methods are possible, but we can never truly be sure that what we are including actually represents what is missing. The biggest problem with the dataset most likely comes from the score variable. Within the exercise, we noticed that some restaurants had different overall scores for the same inspection type on the same date. It is a challenge to accurately know, which score is the most likely score. The agency notes in the data dictionary for the dataset that the score column is updated based on adjudication results, therefore without knowing what the adjudication results are then how do we know which score is the accurate one. Ultimately, the inaccuracy for some observations in the score column can be problematic.

The score column, which corresponds to our outcome of interest, whether or not an inspection results in a score of 28 or higher, is supposed to be objectively measured. The overall score for an inspection is summed based on the points associated with the violations that a restaurant receives during the inspection. Each violation type has its own "base" point. However, inspectors assign additional points based on the extent of the violation, the additional points fall into either least extensive (level 1) to most extensive (level 5) violations. While one can argue that the base points based on the violation type are most likely objective, the inspectors must make the call on how extensive the violation can be, therefore there is some subjectivity when judging the extent of the violation. Overall, the score column and by extent the outcome of interest in this model is not based on entirely objective standards, the human involvement in the process does affect the appropriateness of the outcome of interest.

Ultimately, this does give question to whether or not; the outcome of interest should be based on other factors, such as whether a restaurant is pegged to close or whether a restaurant has a critical health violation. In both cases, subjectivity can also be present, but if a restaurant is going to be closed or if it has a critical health violation, one can argue that the conditions of the restaurant most likely warranted these outcomes. In the case of judging based on the overall score column, we cannot entirely say that a high score is correlated with serious violations, but perhaps an inspector assigned many **general violations**, which are less serious in nature.

The outcome of interest, whether or not an inspection received a score of 28 or

higher, is subject to human bias. As a result, the predictions of our model are also subject to bias. The extent of this bias is not known because we as researchers are not present when an inspector carries out an initial inspection. However, the process of inspection can be improved, which in turn will improve the accuracy and appropriateness of our predictions. One example of improving the inspection process is to send two inspectors instead of one to each restaurant so that the points that a given inspection at a restaurant receives can be less subjective and more standardized.