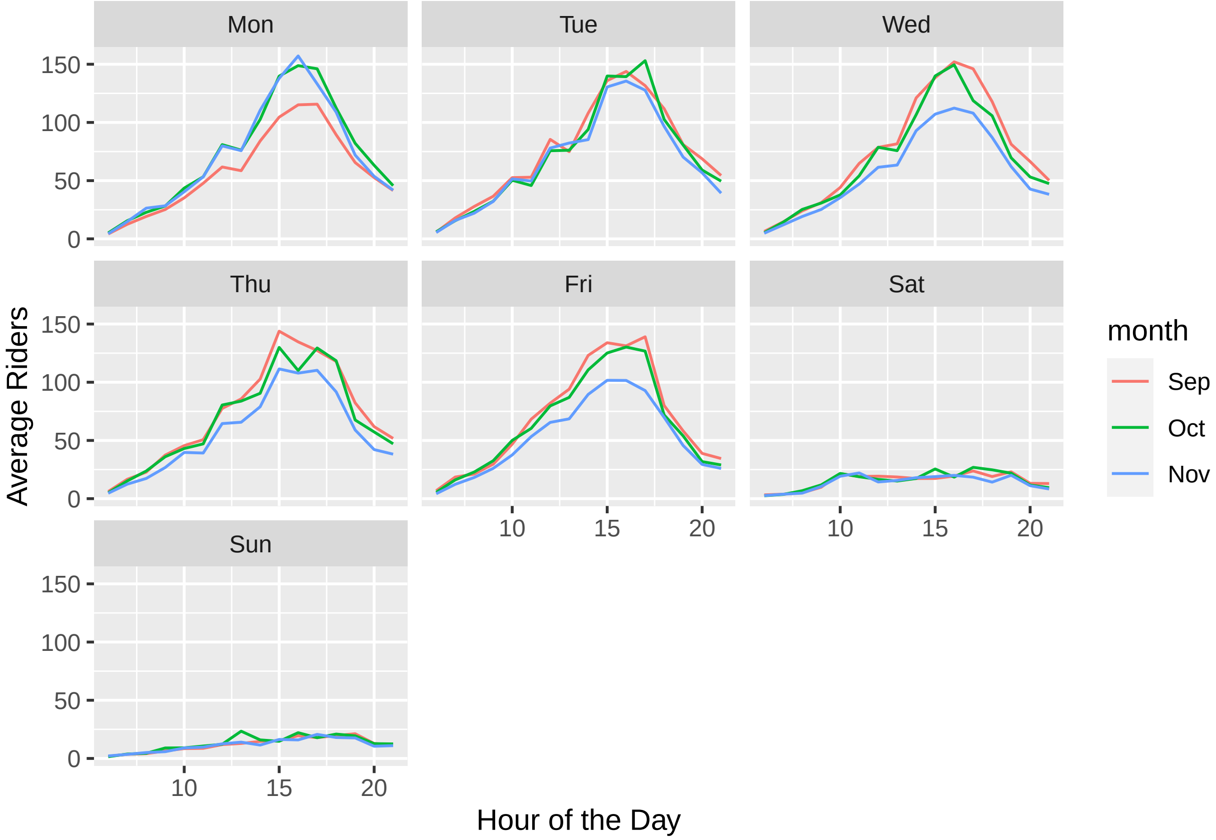
Data Mining Problem Set 2

Hannah Jones

3/12/2021

Problem 1

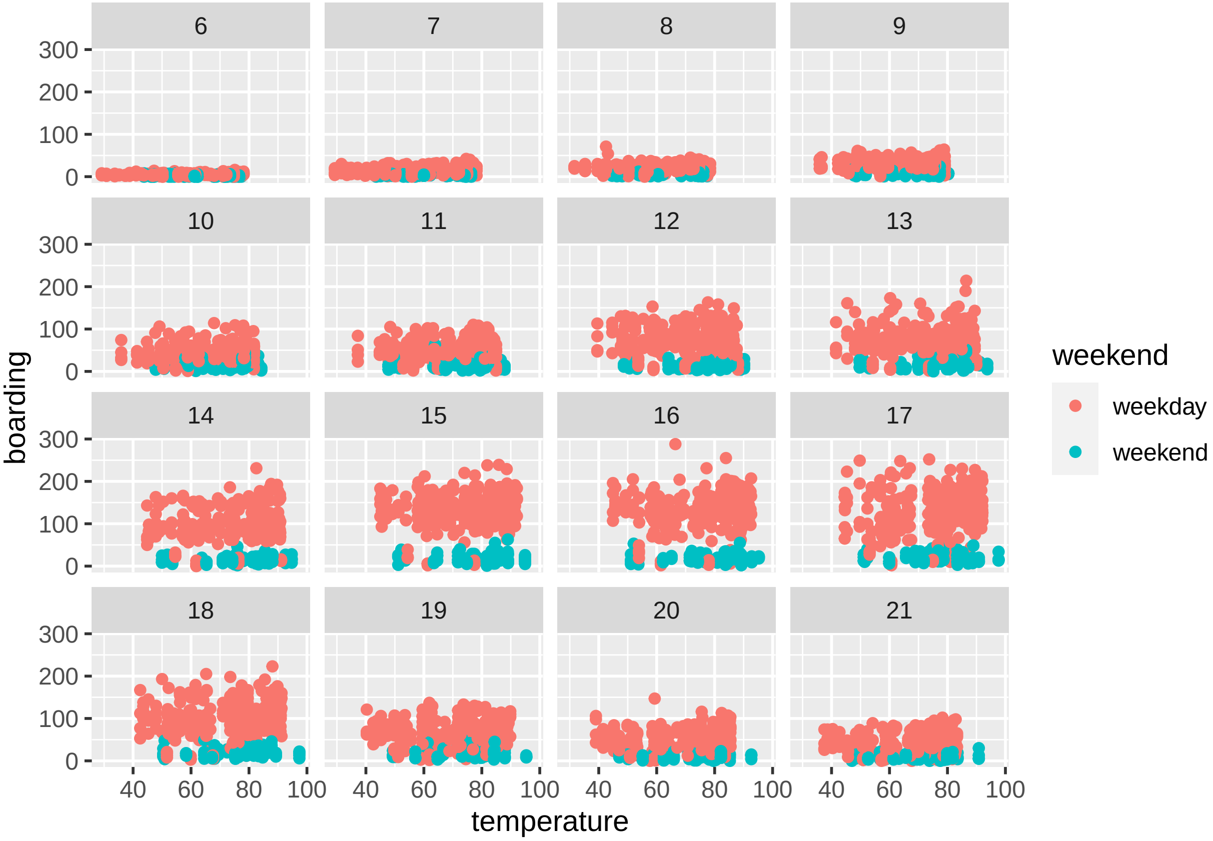
Part 1



The plot above shows average number of riders per hour by day and by month (line color).

As shown in the plot above, on weekdays, no matter which month, ridership generally peaks between 3 and 4pm. The month of September sees fewer average riders on Mondays, likely due to Labor Day weekend weighting down the average as students leave campus. November sees fewer average riders on Wednesday , Thursday and Friday, likely due to Thanksgiving Holidays weighing down the mean as students leave campus. On weekends, there are much fewer riders on average, though it seems Saturday sees a steady stream between 10am and 8pm, while on Sundays, ridership doesnt pick up until midday, but also drops off around 8pm.

Part Two



Plot showing ridership versus temperature by hour of the day

When holding hour of day and weekend status constant, temperature seems to have little effect on ridership. If temperature had an effect, we would see dots of the same color creeping up in riders as temperature increases. However, for both within weekends and within weekdays, ridership seems relatively uniformly distributed across temperatures.

Problem 2

The KNN model and the hand-built model achieved similar out-of-sample mean-squared error, and both beat the medium model. I built two models– one linear model and one using the K-Nearest-Neighbors technique. The hand-built model took into account a variety of home attributes including bedrooms, bathrooms, rooms, living space, lot size, land value, age, location (waterfront), and various interactions of these variables. This process was relatively time and data intensive when trying to choose which variables are significant, and which are not. Then I used the K-Nearest-Neighbors approach which simply looks at a given number of homes that are similar in attributes, and predicts a price for a given house. This technique is less thoughtful, but delivers results close to, and in some cases exceeding, results from the hand-built model when considering out of sample mean squared error.

When assessing home value for taxing purposes, I would suggest using the K-Nearest-Neighbors approach to achieve results comparable to a more human-built model, in much less time. This approach will also succeed in the long term in understanding how different home attributes change in value to buyers. As tastes change, the model will simply capture these changing tastes by relating attributes to home value, rather than require any sort of all-knowing model builder to properly account for these changes.

The mean RMSE values below justify the choice of the KNN model for predicting house value.

```
## [1] 0.6739726
```

The mean RMSE above is for the medium model.

```
#working model

workmod = lm_robust(price~ age + newConstruction + bathrooms+ rooms + waterfront + centralAir + landValu


rmselm_out = foreach(i=1:10, .combine='rbind') %do% {
  saratoga_split = initial_split(saratoga_scale, prop = 0.8)
  saratoga_train = training(saratoga_split)
  saratoga_test = testing(saratoga_split)
    # train the model and calculate RMSE on the test set
  workmod = lm_robust(price~ age + newConstruction + bathrooms+ rooms + waterfront + centralAir + landVa
  this_rmse = modelr::rmse(workmod, saratoga_test)
  }
rmselm_out_mean=mean(rmselm_out)
rmselm_out_mean
```

```
## [1] 0.5933816
```

Above is the mean RMSE for the linear model.

```
#KNN

k_grid = c(2, 4, 6, 8, 10, 15, 20, 25, 30, 35, 40, 45,
           50, 60, 70, 80, 90, 100, 125, 150, 175, 200)
rmse_out = foreach(i=1:10, .combine='rbind') %dopar% {
  saratoga_split = initial_split(saratoga_scale, prop = 0.8)
  saratoga_train = training(saratoga_split)
  saratoga_test = testing(saratoga_split)
  this_rmse = foreach(k = k_grid, .combine='c') %do% {
    # train the model and calculate RMSE on the test set
    knn_model = knnreg(price ~ age + newConstruction + bathrooms + rooms + waterfront + centralAir + lan
    modelr::rmse(knn_model, saratoga_test)
  }
  data.frame(k=k_grid, rmse=this_rmse)
}
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

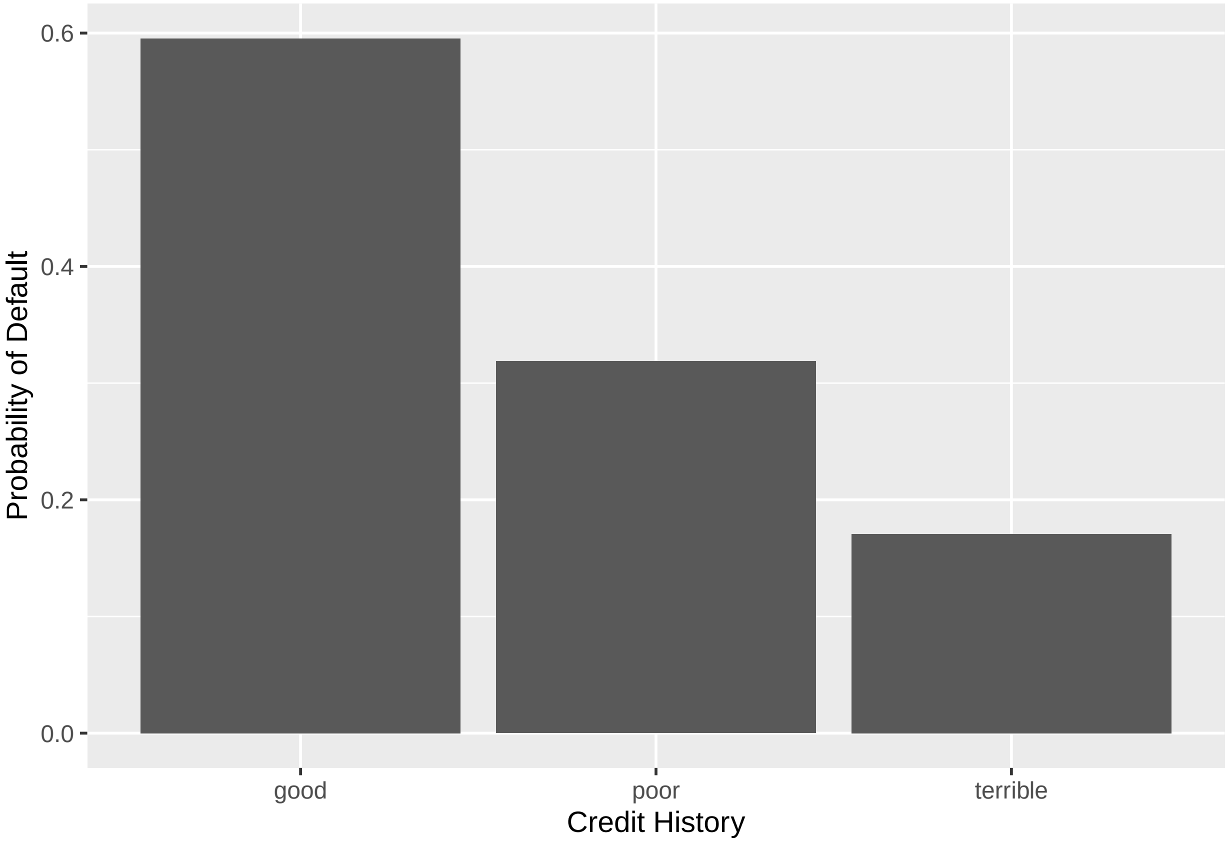
```
rmse_out_knn = arrange(rmse_out, k)

mean_rmse_knn <- rmse_out_knn%>%
  group_by(k)%>%
  summarize(mean = mean(rmse))
min(mean_rmse_knn)
```

```
## [1] 0.6464998
```

Above is the minimum mean RMSE for the KNN model.

## Problem 3



The chart above shows the probability that a lender defaults on their loan based on their Credit History. Contrary to what one might assume, the graph suggests that borrowers with good credit are more likely to default on a loan.

Logit model accuracy:

```
##    yhat
## y     0   1
##   0 109  15
##   1  56  19
```

```
## [1] 0.6432161
```

The logit model predicts whether a borrower will default on their loan based on the loan duration, amount, installments, age of borrower, credit history, purpose and foreign status. As shown by the confusion matrix output and the accuracy score, this model is only successful about 72% of the time.

As discussed above, a borrower's good credit history actually is correlated with a higher probability of default. This could be showing up because of how loan decisions are made. If mostly borrowers with high credit are awarded loans, then they will represent a higher proportion of the loan data and of the default data. It is also possible that good credit holders are over-leveraged due to their good credit, and therefore more likely to default on a loan due to overall credit holdings. It seems there is some selection bias towards good credit in the loan awards in general, and this data suggests perhaps good credit is not alone a good predictor of credit-worthiness.

Based on the chart above and the model accuracy, this model is a poor choice for predicting high vs low probability of default. This model does not do a great job of predicting defaults. The data has over-sampled defaults and has not accounted for the selection bias associated with good credit. This dataset and predictive model would under-predict credit default for bad credit score borrowers and over-predict for high credit score borrowers. The new sample of data will need to have extensive data on the ‘poor’ and ‘terrible’ credit score borrowers.

Problem 4

Baseline 1 This first baseline model model is a logit model, regressed on market segment, adults, customer type, and repeated guest status. The out of sample accuracy is below.

```
##    yhat
## y      0
##   0 8251
##   1  748
```

```
## [1] 0.9168797
```

This model has an accuracy of ~92.3% based on a probability threshold of 0.5. Children rarely are on bookins, so this simple model never predicts a child showing up. This model's accuracy reflects the percent of time when no children are present.

Baseline 2 The next model, baseline 2, predicts children based on all other variables except arrival date, also using a logistic regression. The out of sample accuracy is below:

```
##    yhat
## y      0    1
##   0 8135  116
##   1  481  267
```

```
## [1] 0.9336593
```

The confusion matrix and accuracy rate above show a slightly more sophisticated model, boasting slightly better accuracy. The addition of more covariates improved the model by about 1%.

Baseline 3 For the third baseline model, I use the lasso method to arrive at which variables to use in a predictive model, then plugged them in to a logistic model.

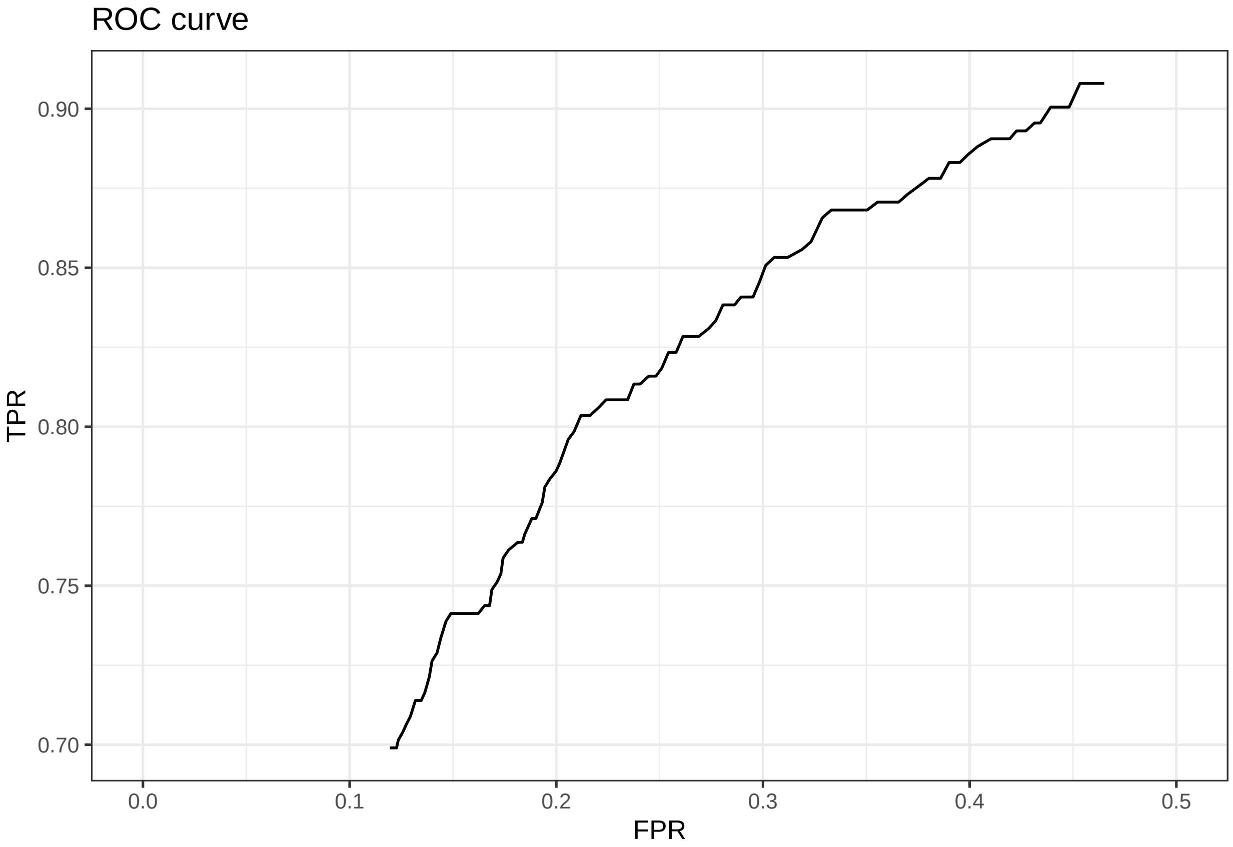
```
##    yhat
## y      0    1
##   0 8140  111
##   1  466  282
```

```
## [1] 0.9358818
```

This model performs the best of the three, with out of sample accuracy of ~94%. The Lasso resulted in the choice variables: hotel, lead_time, adults, meal, market_segment, distribution_channel, is_repeated_guest, previous_bookings_not_canceled, reserved_room_type, booking_changes, customer_type, average_daily_rate, total_of_special_requests, and arrival_date. I will move forward with this model to the validation data.

Validation Step 1

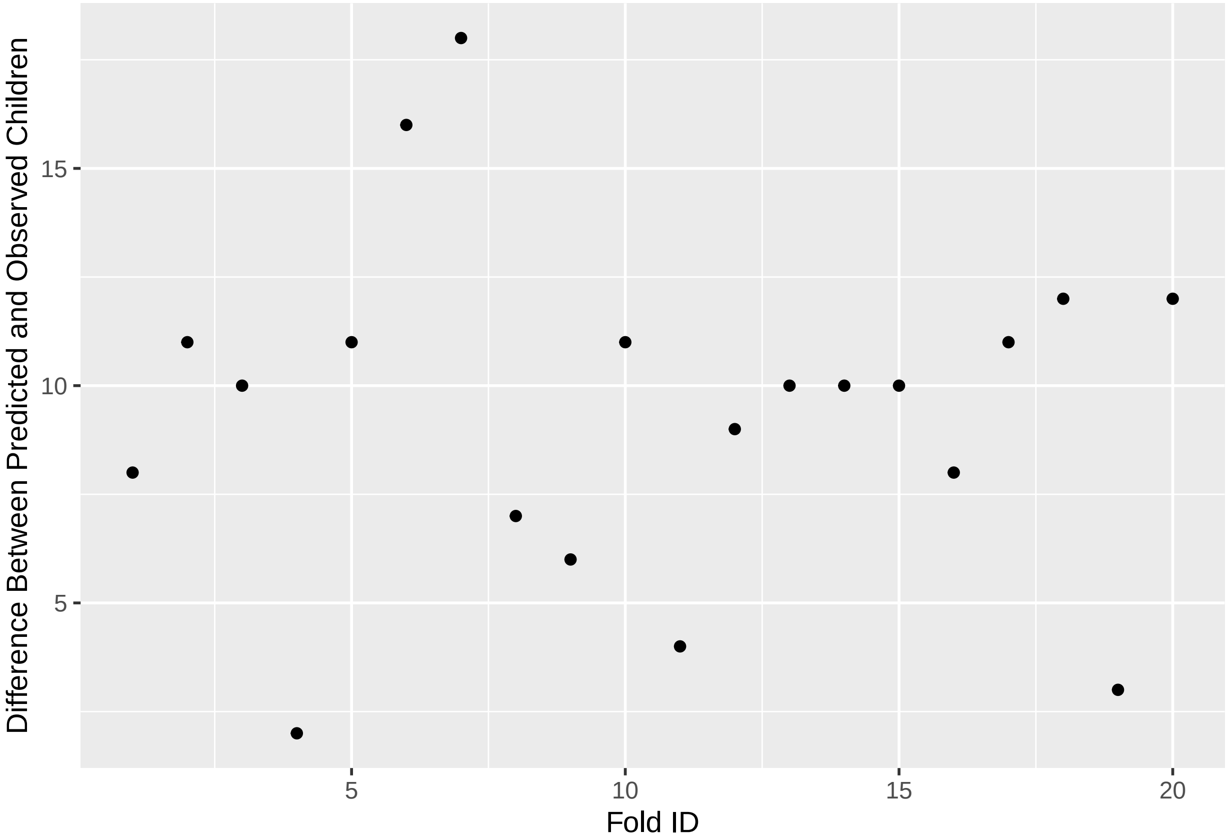
Using the model identified in the previous section, I predicted outcomes of the full validation data set, then looked at the True Positive vs False Positive rates by threshold, as shown in the graph below.



The ROC curve above charts the false positive rate versus the true positive rate for the Lasso model built in the last section.

Validation Step 2

For the final validation step, I used the validation data to fit my best performing model. I then assigned a prediction to each observation before splitting the data into 20 randomly assigned groups. Within each group I calculated the predicted and actual probability of a child, as well as the predicted and actual number of children who did arrive. I took the difference between reality and prediction and charted it below.



The chart and table above shows the difference between prediction and reality for each fold. The difference varies greatly from fold to fold, suggesting that even our best model cannot consistently predict whether a child will be present on a booking across random groups of observations.

Table 1: Children Epectations vs Reality

Fold Number	Predicted Children	Actual Children	Difference
1	12	20	8
2	10	21	11
3	6	16	10
4	11	13	2
5	16	27	11
6	8	24	16
7	13	31	18
8	10	17	7
9	5	11	6
10	8	19	11
11	18	22	4
12	10	19	9
13	10	20	10
14	13	23	10
15	11	21	10
16	10	18	8
17	8	19	11
18	9	21	12
19	17	20	3
20	8	20	12