Problem Set 2

By Jae You (jhy479)

1)Saratoga House prices

```
## [1] 66006.19
```

This is the previous model that was done in class, with an average rmse of 65969.85.

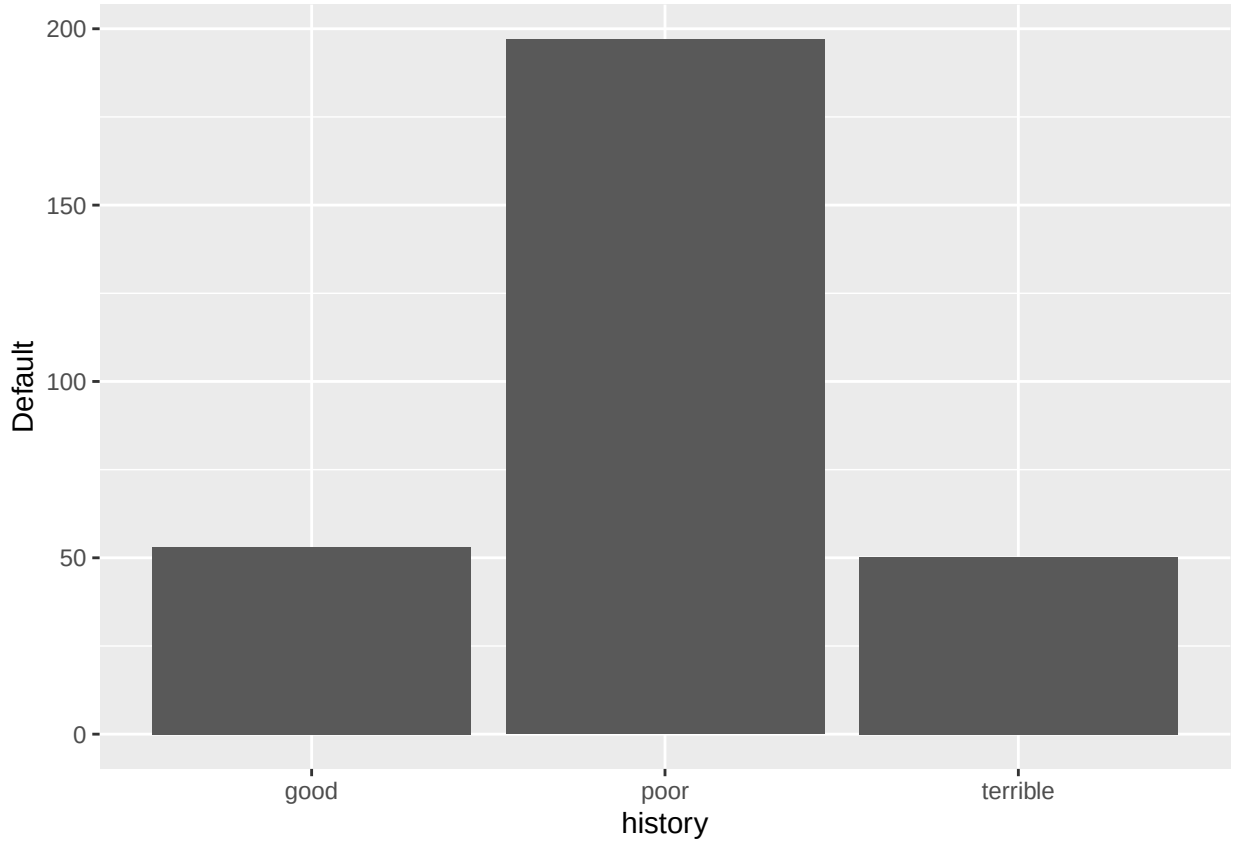
```
## [1] 59810.4
```

This is the new model that outperforms the model done in class, with an average rmse of 59384.4. I removed CentralAir and added the landValue interaction.

```
## [1] 64984.31
```

This is the KNN model that was scaled and done at a lower value of K than usual, K=10. It has an average rmse of 63550.47, which is still higher than the linear model that I produced. Theoretically, the rmse of the KNN model can be drastically lowered by lowering the value of K (as K=1 gives an rmse of 545!). However, this would cause much higher variance in return for eliminating bias, which is not helpful for differentiating between noise and signals for Saratoga house prices. As K=10 is already a low enough number and yet the KNN model is still outperformed by the linear model, I would argue that the linear model does a better job of achieving lower out-of-sample mean-squared errors.

2)Classification and retrospective sampling




```
##    yhat
## y     0   1
##   0 626  74
##   1 139 161
```

```
## [1] 0.626
```

We can see that the majority of defaulters are those with poor history, while borrowers with good and terrible history both tend not to default as much. This may be because individuals with good credit history would want to keep having a good history, while those with terrible history may not be approved of loans which they may consider defaulting. On the other hand, those with poor history may feel as if they can default some more before they hit rock bottom with terrible history.

I believe this model is a step in the right direction for building a predictive model of defaults. However, in order to better know if borrowers may have higher or lower probabilities of defaulting, I would suggest that the bank take into account the duration of the loan and the purpose of the loan, as some borrowers who are looking to default may prefer shorter-term loans since they have no intention to pay it back.

3)Children and hotel reservations

For all the models, I averaged the estimate of out-of-sample RMSE over many different random train/test splits due to random variation.

Baseline 1:

```
## [1] 0.268188
```

Baseline 2:

```
## [1] 0.2329475
```

Baseline 3: I kept getting errors when trying to add more than 1 interaction, as well as any log(variable) or interactions with children. Thus, my best linear model is the following, which slightly outperforms model #2:

```
## [1] 0.2329524
```

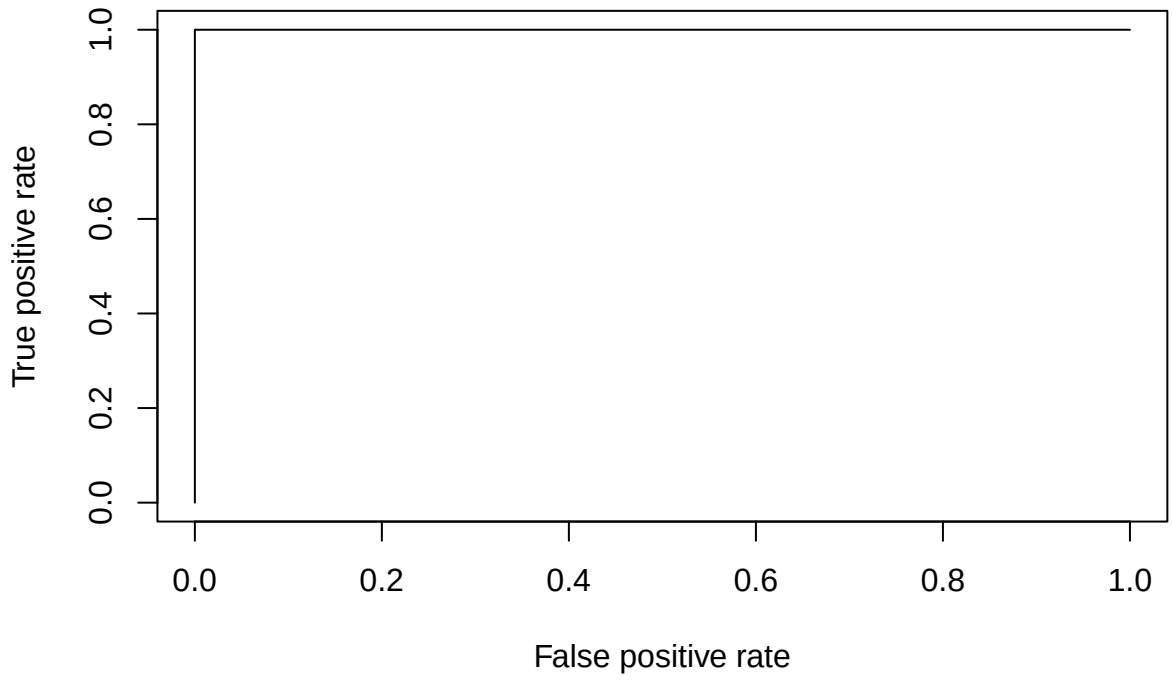
Step 1:

Originally, my code was to validate the linear model that I did in baseline 3. However, I kept having the "factor arrival_date has new levels" error, even though my model excluded arrival_date entirely. I attempted to remove the column itself from the data, but failed to do so. Here is the attempt below:

"hotels_val = hotels_val %>% mutate(fold_id = rep(1:5, length=nrow(hotels_val)) %>% sample)

rmse_cv_hotels_val = foreach(fold = 1:5, .combine='c') %do% { hotels_val_split=initial_split(hotels_val, prop=0.8) hotels_val_train=training(hotels_val_split) hotels_val_test=testing(hotels_val_split) lmval = lm(children ~ . + (stays_in_weekend_nights * stays_in_week_nights) - arrival_date- previous_cancellations, data=hotels_val_train) modelr::rmse(lmval, data=filter(hotels_val, fold_id == fold)) }"

As such, I went straight to logistic regression:




```
##    yhat
```

```
## y      0    1
##   0 3645   46
##   1  217   91
```

From the confusion matrix, we can see that the (out-of-sample) FPR here is 52/(52+3634)= 0.0141, and the TPR here is 106/(207+106)= 0.3387. So the ROC curve should have a point at (0.0141, 0.3387).

Model validation: step 2: I was able to split the data into 20 equal folds. However, I could not use the folds for my training/testing sets because of many errors, most notably " 'x' should be an 'rsplit' object." Here is my code:

"folds20 <- createFolds(hotels_val$children, k=20)

lap <- lapply(folds20, function(ind, dat) dat[ind,], dat = hotels_val) split_up <-unlist(lapply(lap, nrow))"

From here, I tried many things to split/train/test:

->Running initial_split(folds20,prop=0.8) led to the error "subscript out of bounds".

->Running the individual folds themselves (split=initial_split(folds20[["Fold01"]], p=0.8) split=initial_split(lap[["Fold01"]], p=0.8)) led to a multiple formal argument error, with and without the lap variable.

```
##    yhat
## y      0    1
##   0 3621   46
##   1  211  121
```

The (out-of-sample) FPR here is 51/(51+3629)= 0.01386, and the TPR here is 121/(198+121)= 0.3793.