DM_Excersize2_Q1

Joey Herrera

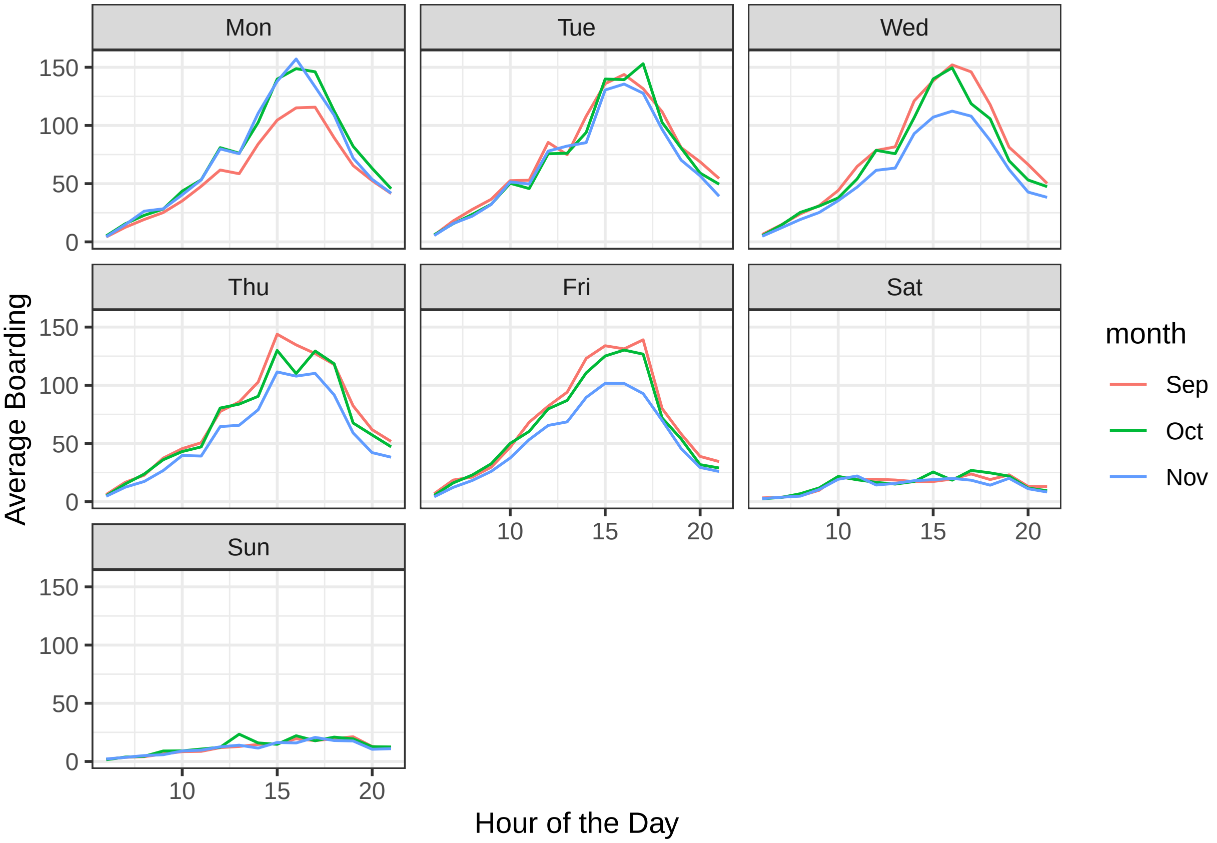
2/10/2021

Data Mining Assignment 2

Question 1: Data Visualization

1A. One panel of line graphs that plots average boardings grouped by hour of the day, day of week, and month. You should facet by day of week. Each facet should include three lines, one for each month, colored differently and with colors labeled with a legend.

Give the figure an informative caption in which you explain what is shown in the figure and address the following questions, citing evidence from the figure. Does the hour of peak boardings change from day to day, or is it broadly similar across days? Why do you think average boardings on Mondays in September look lower, compared to other days and months? Similarly, why do you think average boardings on Weds/Thurs/Fri in November look lower?



Caption: The above facet plots depict the average number of passengers on Capmetro buses at UT by the hour of the day, day of the week, and relevant month of the year.

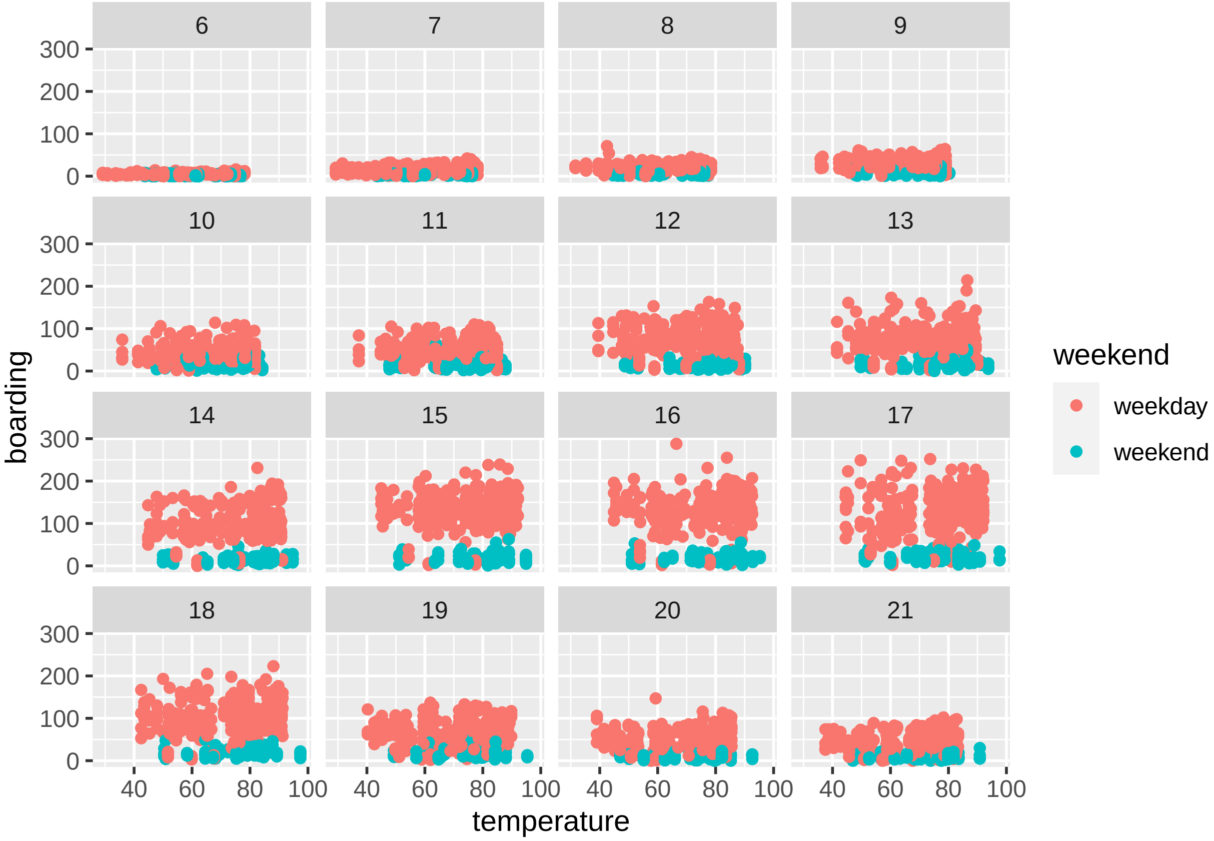
Does the hour of peak boardings change from day to day, or is it broadly similar across days? During peak hours of the day bus boardings are very similar during the week. Average boarding significantly decreases during the weekend since UT students do not have class.

Why do you think average boardings on Mondays in September look lower, compared to other days and months? Compared to other weekdays and months, the average student bus boarding are smaller. This is because of Labor Day, which UT students have off from class. The main reason why students board the bus to campus is to attend class, if you average in Labor Day with the other Monday's in September, the average number of boardings will decrease significantly.

Why do you think average boardings on Weds/Thurs/Fri in November look lower? The average boardings on Wed/Thurs/Fri are also lower because of a holiday break. Typically, UT students recieve Weds/Thurs/Fri off during the week of Thanksgiving.

1B. One panel of scatter plots showing boardings (y) vs. temperature (x) in each 15-minute window, faceted by hour of the day, and with points colored in according to whether it is a weekday or weekend.

Give the figure an informative caption in which you explain what is shown in the figure and answer the following question, citing evidence from the figure. When we hold hour of day and weekend status constant, does temperature seem to have a noticeable effect on the number of UT students riding the bus?



Caption: The facet plots aboce show the temperature's effect on average bus boarding for capmetro at UT for each hour of the day and weekday versus weekend.

When we hold hour of day and weekend status constant, does temperature seem to have a noticeable effect on the number of UT students riding the bus?

The facted plots above illustrate that temperature does not have a noticable effect on the number of UT students riding the bus when holding hour of the day and weekend status constant. There seem to be a couple of outliers on the 16th hour of a weekday (4 pm) around 60 degrees farenehit, but these observations seem insignificant.

Question 2: Saratoga Housing Prices

2A. Return to the data set on house prices in Saratoga, NY that we considered in class. Recall that a starter script here is in saratoga_lm.R. For this data set, you'll run a "horse race" (i.e. a model comparison exercise) between two model classes: linear models and KNN.

Build the best linear model for price that you can. It should clearly outperform the "medium" model that we considered in class. Use any combination of transformations, engineering features, polynomial terms, and interactions that you want; and use any strategy for selecting the model that you want.

Answer: The goal of this question is to compare an optimal KNN with an optimal linear model and the "medium" model we considered in class. I will compare the RMSE of each model to each other where the lower the RMSE the closer to the actual housing price the prediction is. To begin, I standardized my variables to account for the large variation in price. Next, I estimated twenty samples of the medium model and took the average of its RMSE as a measure to compare the KNN and optimal linear model with. The RMSE for the medium model can be found below.

```
## [1] 0.6601405
```

Next, I hand created a linear model using trial and error to put the main effects that gave the model its lowest RMSE. Then, I added a few interactions with the living area variable to other variables that seem dependent on it such as, rooms, bathrooms, and fireplaces. Finally, I took twenty samples of my best linear model to get a standardize RMSE below, which narrowly beats the medium linear model above.

```
## [1] 0.5959369
```

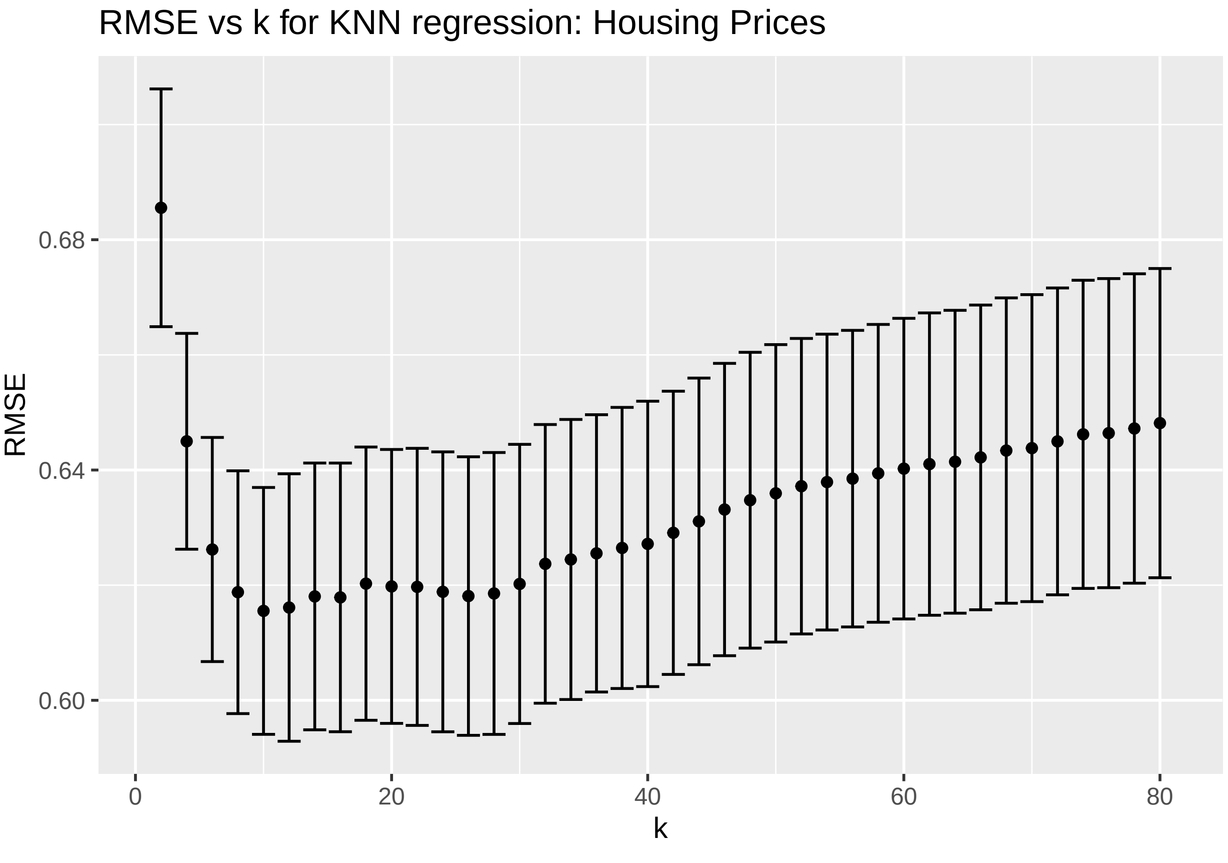
2B. Now build the best K-nearest-neighbor regression model for price that you can. Note: you still need to choose which features should go into a KNN model, but you don't explicitly include interactions or polynomial terms. The method is sufficiently adaptable to find interactions and nonlinearities, if they are there. But do make sure to standardize your variables before applying KNN, or at least do something that accounts for the large differences in scale across the different variables here.

Which model seems to do better at achieving lower out-of-sample mean-squared error? Write a report on your findings as if you were describing your price-modeling strategies for a local taxing authority, who needs to form predicted market values for properties in order to know how much to tax them. Keep the main focus on the conclusions and model performance; any relevant technical details should be put in an appendix.

Answer: To create the KNN model I used the kfold cross validation technique using twenty different folds of data and fourty different k values that start at two and increase by two until reaching eighty.

The KNN technique allowed me to choose the optimal value of K using the figure above. After using k=16 for my predication model I found that the KNN model significantly outperforms the standadized RMSEs for both the best linear model and the medium model. The standardized RMSE for the KNN model can be found below.

```
## [1] 0.6178769
```



In conclusion, as a local tax authority, I recommend using the KNN model to predict market values for properties in order to know how much to tax them.

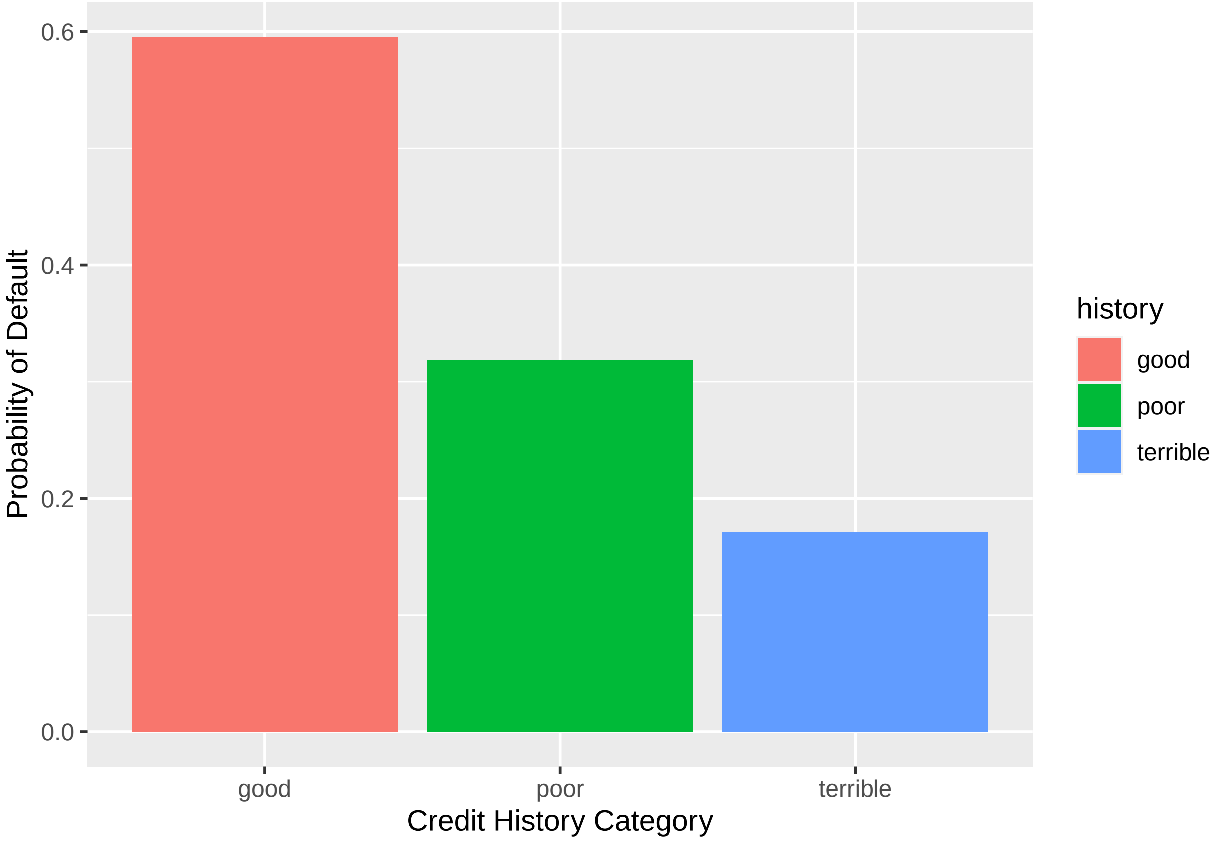
Question 3

Consider the data in german_credit.csv on loan defaults from a German bank. The outcome variable of interest in this data set is default: a 0/1 indicator for whether a loan fell into default at some point before it was paid back to the bank. All other variables are features of the loan or borrower that might, in principle, help the bank predict whether a borrower is likely to default on a loan.

This data was collected in a retrospective, "case-control" design. Defaults are rare, and so the bank sampled a set of loans that had defaulted for inclusion in the study. It then attempted to match each default with similar sets of loans that had not defaulted, including all reasonably close matches in the analysis. This resulted in a substantial oversampling of defaults, relative to a random sample of loans in the bank's overall portfolio.

Of particular interest here is the "credit history" variable (history), in which a borrower's credit rating is classified as "Good", "Poor," or "Terrible." Make a bar plot of default probability by credit history, and build a logistic regression model for predicting default probability, using the variables duration + amount + installment + age + history + purpose + foreign.

What do you notice about the history variable vis-a-vis predicting defaults? What do you think is going on here? In light of what you see here, do you think this data set is appropriate for building a predictive model of defaults, if the purpose of the model is to screen prospective borrowers to classify them into "high" versus "low" probability of default? Why or why not—and if not, would you recommend any changes to the bank's sampling scheme?



Caption: The bar plot illustrates the probability of an individual defaulting on a loan based on their credit history. This is particularly interesting because individuals with good credit history are more likely to default on a loan than people from other credit history categories.

After predicting out-of-sample accuracy for the predictive logisitic model, you can see that there are very few observations where people default in the confusion matrix below. This leads to an out-of-sample accuracy that hovers around 70%.

```
##    yhat
## y     0   1
##   0 128   4
##   1  58   9
```

```
## [1] 0.6884422
```

The null model has a 70% accuracy rating when guessing that no individual defaulted on their loan. Thus, the approximate 1.4% absolute increase in model accuracy with the logisitic model is marginal.

What do you notice about the history variable vis-a-vis predicting defaults? What do you think is going on here? It seems the history variable does a poor job of predicting defaults. I think this is becuase individuals categorized as having poor or terrible credit are less likely to apply for or recieve loans.

In light of what you see here, do you think this data set is appropriate for building a predictive model of defaults, if the purpose of the model is to screen prospective borrowers to classify them into “high” versus “low” probability of default? Why or why not—and if not, would you recommend any changes to the bank’s sampling scheme? I do not think this data is appropriate for building a predictive model of defaults because the accuracy rating of a logistical regression that includes important characteristic variables (duration + amount + installment + age + history + purpose + foreign) only out performs the null model by approximately 1.4%. This could occur because of the small number of individuals who default on loans at the German bank in the dataset. I recommend that German Bank not oversample defaults from their data.

Question 4

Model building: Below is the code, respective confusion matrices, and out-of-sample accuracy for the three different models.

```
#Model building

#Create the initial split for hotels_dev
hotels_dev_split = initial_split(hotels_dev, prop = 0.8)
hotels_dev_train = training(hotels_dev_split)
hotels_dev_test = testing(hotels_dev_split)

log_baseline1 = glm(children ~ market_segment + adults + customer_type + is_repeated_guest, data = hotel

#coef(log_baseline1) %>% round(3)

#Add predictions and calculate the out-of-sample accuracy
phat_baseline1 = predict(log_baseline1, hotels_dev_test)
yhat_baseline1 = ifelse(phat_baseline1 > 0.5, 1, 0)
confusion_matrix_baseline1_out = table(y = hotels_dev_test$children, yhat = yhat_baseline1)

confusion_matrix_baseline1_out #confusion matrix
```

```
##    yhat
## y      0
##   0 8289
##   1  710
```

```
sum(diag(confusion_matrix_baseline1_out))/sum(confusion_matrix_baseline1_out)#out-of-sample accuracy
```

```
## [1] 0.9211023
```

```
#0.9173241
```

At a threshold of 0.5, the small model with few covariates does not predict that any booking will have kids.

```
#Create baseline model 2

hotels_dev_split2 = initial_split(hotels_dev, prop = 0.8)
hotels_dev_train2 = training(hotels_dev_split2)
hotels_dev_test2 = testing(hotels_dev_split2)

log_baseline2 = glm(children ~ . - arrival_date, family = "binomial", data = hotels_dev_train2)

#coef(log_baseline2) %>% round(3)

#Add predictions and calculate the out-of-sample accuracy
phat_baseline2 = predict(log_baseline2, hotels_dev_test2)
yhat_baseline2 = ifelse(phat_baseline2 > 0.5, 1, 0)
confusion_matrix_baseline2_out = table(y = hotels_dev_test2$children, yhat = yhat_baseline2)

confusion_matrix_baseline2_out #confusion matrix
```

```
##    yhat
## y      0    1
##   0 8199   72
##   1  533  195
```

```
sum(diag(confusion_matrix_baseline2_out))/sum(confusion_matrix_baseline2_out)#out-of-sample accuracy
```

```
## [1] 0.9327703
```

```
#0.9329926
```

The large model's confusion matrix (found above) is effective at predicting booking that have children and has an out-of-sample accuracy that is continuously over 90%.

```
#Create the best linear model possible
hotels_dev_step_split = initial_split(hotels_dev, prop = 0.8)
hotels_dev_step_train = training(hotels_dev_step_split)
hotels_dev_step_test = testing(hotels_dev_step_split)

#Create medium baseline model using baseline model 1
lm_medium = lm(children ~ market_segment + adults + customer_type + is_repeated_guest, data = hotels_dev

#Find the best linear model
library(gamlr) #For the Lasso regression

####Use the Lasso regression
## full model
full = glm(children ~ ., data=hotels_val, family=binomial)

#Create numeric feature matrix.
scx = model.matrix(children ~ .-1, data=hotels_val) # do -1 to drop intercept!
scy = hotels_val$children

# Cross validation lasso regression
sccvl = cv.gamlr(scx, scy, nfold=20, family="binomial", verb=F)

#plot(sccvl, bty="n")
```

```
## CV min deviance selection
#scb.min = coef(sccvl, select="min")
#log(sccvl$lambda.min)
#sum(scb.min!=0) # note: this is random!  because of the CV randomness

#scbeta = coef(sccvl)
#scbeta



lm0 = lm(children ~ hotel + lead_time + adults + meal + distribution_channel + is_repeated_guest + marke

#Find out-of-sample accuracy
phat_step = predict(lm0, hotels_dev_step_test)
yhat_step = ifelse(phat_step > 0.5, 1, 0)
confusion_matrix_step_out = table(y = hotels_dev_step_test$children, yhat = yhat_step)

confusion_matrix_step_out #confusion matrix
```

```
##    yhat
## y      0    1
##   0 8170  127
##   1  450  252
```

```
sum(diag(confusion_matrix_step_out))/sum(confusion_matrix_step_out)#out-of-sample accuracy
```

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

```
# 0.9175464
```

To create the best linear model, I used a lasso regression with cross validation to find the which parameters should be in the linear model. The out-of-sample accuracy for this model consistently outperforms the second baseline model by half a percentage point.

Model Validation Step 1: Once you've built your best model and assessed its out-of-sample performance using hotels_dev, now turn to the data in hotels_val. Now you'll validate your model using this entirely fresh subset of the data, i.e. one that wasn't used to fit OR test as part of the model-building stage. (Using a separate "validation" set, completely apart from your training and testing set, is a generally accepted best practice in machine learning.)

Produce an ROC curve for your best model, using the data in hotels_val: that is, plot TPR(t) versus FPR(t) as you vary the classification threshold t.

```
#Model Validation Step 1

#Create the train-test split
#hotels_val_split = initial_split(hotels_val, prop=0.8)
#otels_val_train = training(hotels_val_split)
#hotels_val_test = testing(hotels_val_split)

#Fit the model with the training data
#lm_val = lm(children ~ market_segment + adults + customer_type + is_repeated_guest +
#    market_segment:adults + adults:customer_type + market_segment:is_repeated_guest, data = hotels_val

#coef(lm_val) %>% round(3)

#Add predictions and calculate the out-of-sample accuracy
phat_val = predict(lm0, hotels_val)
yhat_val = ifelse(phat_val > 0.15, 1, 0)
confusion_matrix_val_out = table(y = hotels_val$children, yhat = yhat_val)

confusion_matrix_val_out #confusion matrix
```

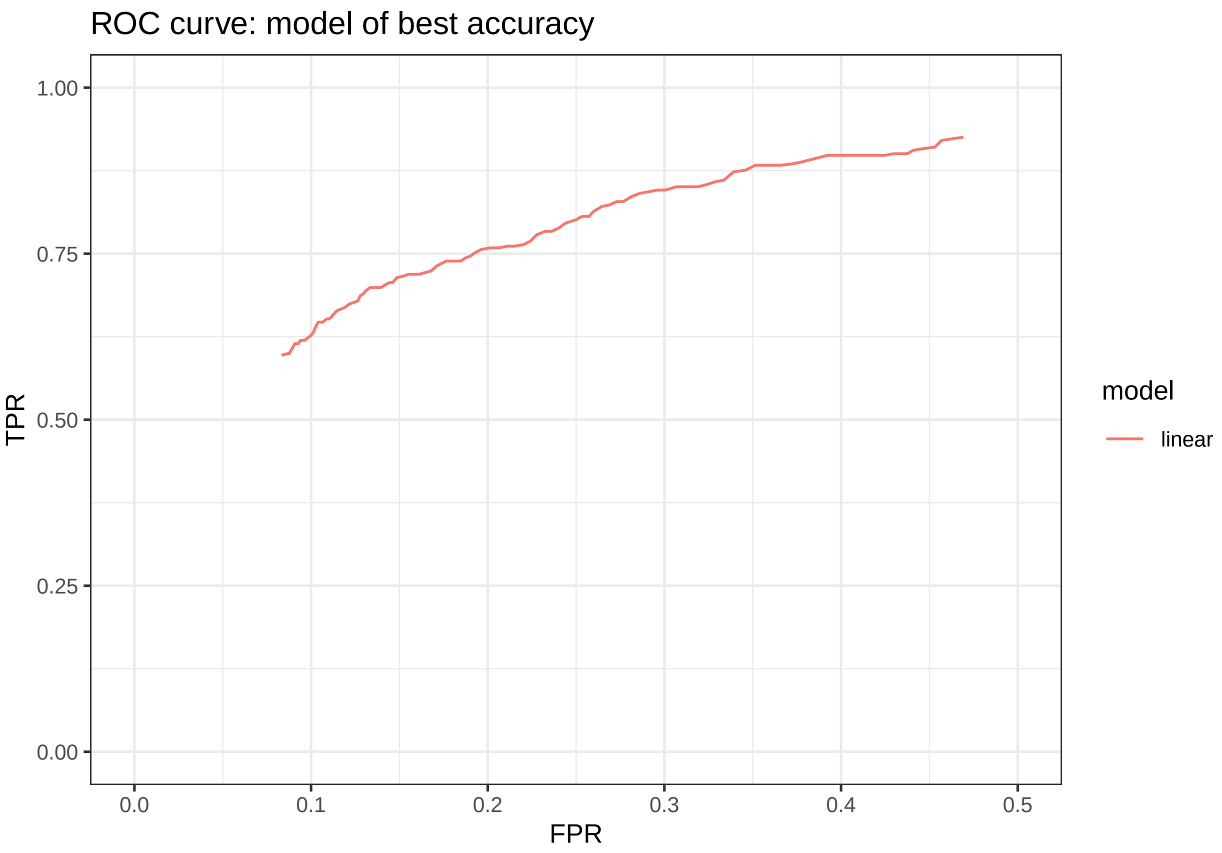
```
##    yhat
## y      0    1
##   0 4214  383
##   1  162  240
```

```
sum(diag(confusion_matrix_val_out))/sum(confusion_matrix_val_out)#out-of-sample accuracy
```

```
## [1] 0.8909782
```

```
# 0.9029029
```

After estimating the accuracy of the predicted logistic model on the hotels_val dataset, we see that the accuracy hovers below 90%.



Caption: The ROC curve depicts the true positive rate and false positive rate for values of threshold t.

Model Validation Step 2: Next, create 20 folds of hotels_val. There are 4,999 bookings in hotels_val, so each fold will have about 250 bookings in it – roughly the number of bookings the hotel might have on a single busy weekend.

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 20 x 5
##    fold_id prop_children prop_pred count_children count_pred
##      <int>         <dbl>     <dbl>          <dbl>      <dbl>
##  1       1         0.096     0.0817             24         11
##  2       2         0.068     0.0868             17         14
##  3       3         0.092     0.0865             23         14
##  4       4         0.108     0.0907             27         14
##  5       5         0.076     0.0853             19         12
##  6       6         0.096     0.101              24         19
##  7       7         0.06      0.0622             15          9
##  8       8         0.08      0.0761             20         13
##  9       9         0.08      0.0954             20         16
## 10      10         0.084     0.0795             21         10
## 11      11         0.072     0.0658             18          7
## 12      12         0.04      0.0722             10          9
## 13      13         0.08      0.0789             20         12
## 14      14         0.092     0.0836             23         13
## 15      15         0.052     0.0739             13         10
## 16      16         0.088     0.0732             22         12
## 17      17         0.076     0.0902             19         16
## 18      18         0.096     0.0709             24          8
## 19      19         0.096     0.0812             24         14
## 20      20         0.0763    0.0729             19          9
```

The table above indicates that this model is inconsistent at predicting the the total number of bookings in a group of 250 bookings. Since the number of bookings with children are small, over or under predicting by ten children could misinform hotel management on how many chicken nuggets they should purchase in the given timeframe.