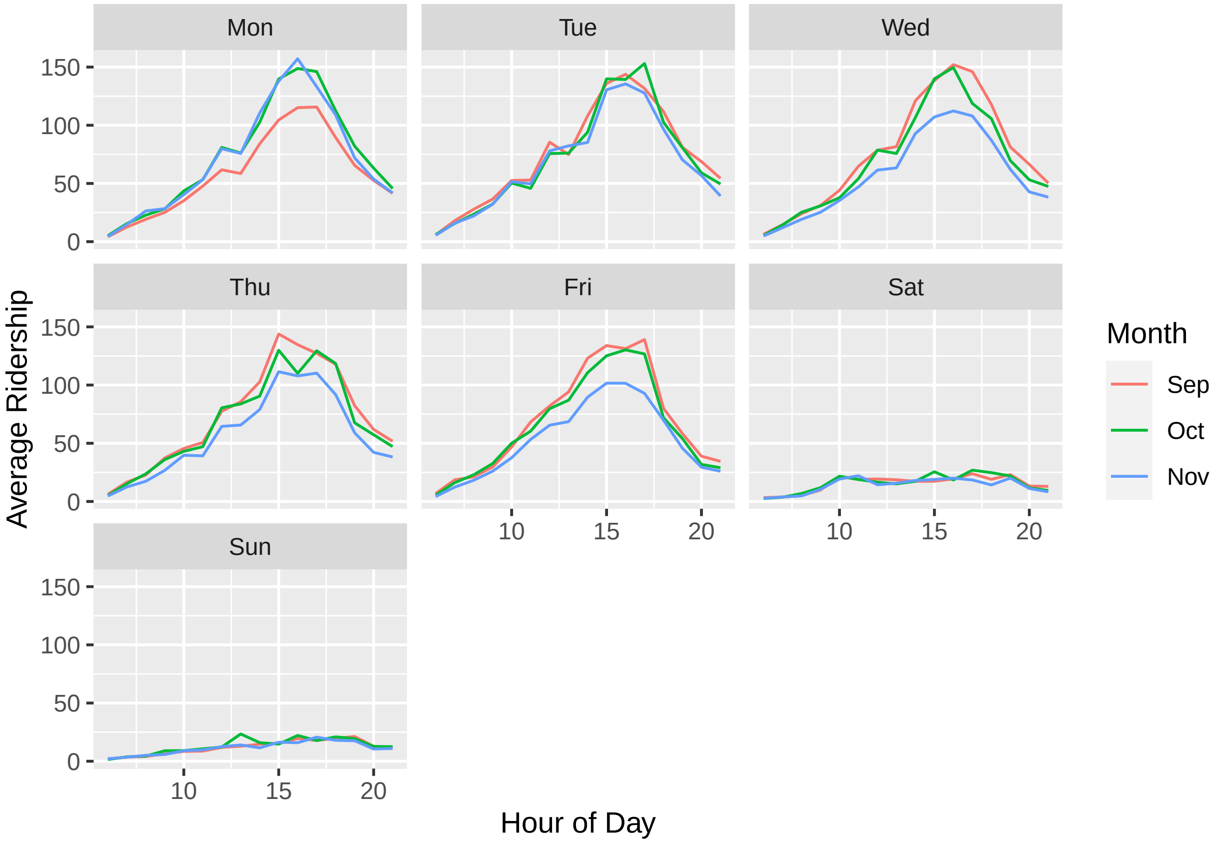
Exercises #2 for Data Mining and Statistical Learning

Gaetano Dona-Jehan and Jordan Fox

Problem 1: Capital Metro Ridership Rates

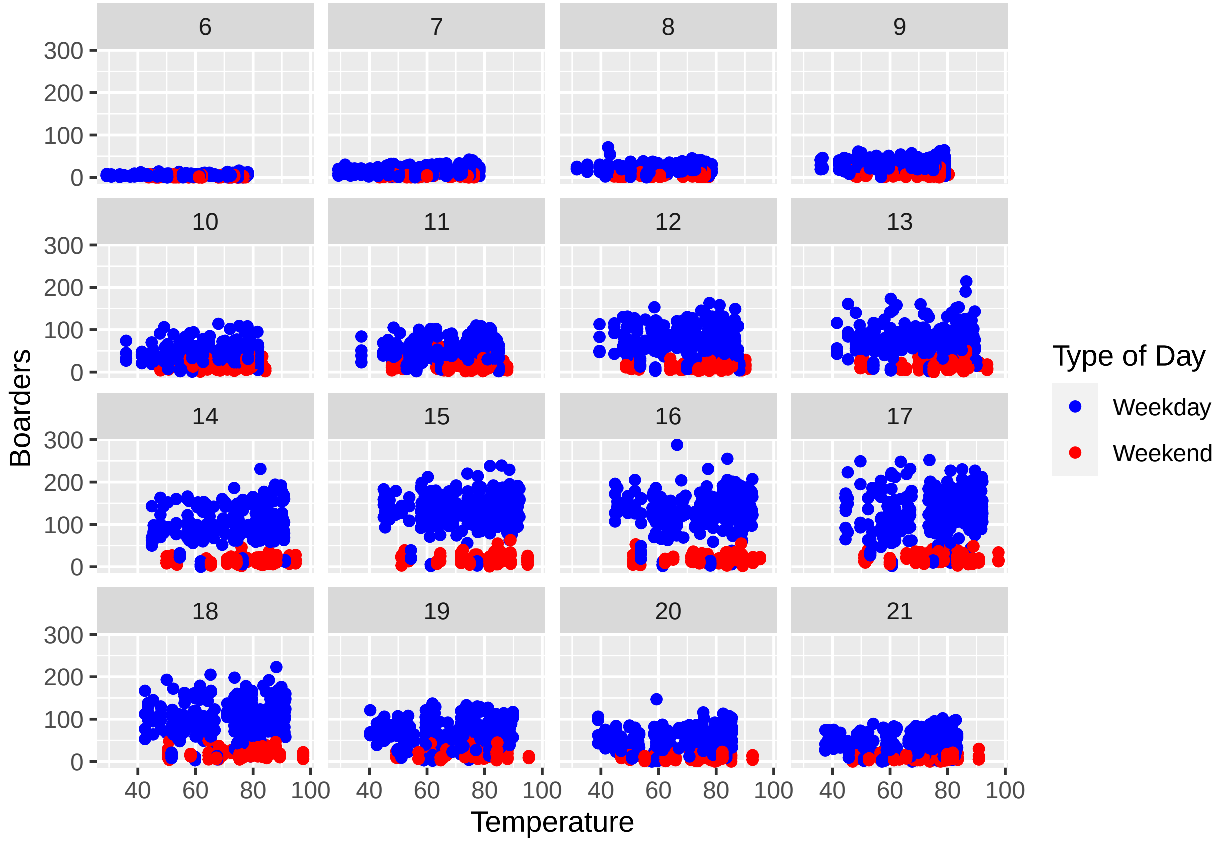
We're asked to make two faceted plots using data on bus ridership collected by Capital Metro. First, we create a table of line graphs showing the average ridership by hour of the day, faceted by day of the week. Then, we're asked to create a panel of scatter plots of the average ridership per 15-minute interval by hour of the day and average temperature.

Average Ridership by Hour of Day by Month of Year



Above we've plotted the average ridership by hour of the day for September, October, and November, faceted by day of the week. We can see that utilization tends to peak around 17:00 (5:00PM) on weekdays, signifying the end of the work day for many commuters. For weekends, ridership is so low that it isn't clear where the peak is. One interesting feature is that November has the lowest ridership for Wednesdays, Thursdays, and Fridays, which we attribute to the Thanksgiving Holidays. Additionally, September has the lowest average ridership on Mondays. This could be because the first Monday in September is Labor Day, and for many students and public workers this represents a state holiday.

Average Ridership By Temperature by 15-minute interval Per Hour



Above, we plot the average ridership as a function of temperature using 15-minute intervals, with the data points colored by weekend. Controlling for hour of day and weekend, it's not obvious that temperature has an effect on the number of students using the bus; there are some slight upticks in ridership around 90 degrees Fahrenheit in the later afternoon, but we have a hard time attributing this to anything beyond a coincidence. It could be that students are more likely to use the bus in the afternoon on hot days, but we can't be certain.

Report on Model Selection for Predicting Housing Prices in Saratoga Springs

We've been tasked with designing a predictive model that will help the city assess property values. To do so, we use a data set on almost 2,000 homes within the city limits. Variables include price of the home, the number of rooms, the age of the home, lot size, land value, sewage type, and air conditioning (heat/ central air). Additionally, we also have data on the percentage of college students living nearby, and whether the home is built in a newly-constructed subdivision. To build the best model for predicting house prices given house characteristics, we used three approaches.

First, we regressed price onto all variables to test which variables had a statistically significant effect. From here, we dropped variables whose effects were statistically insignificant from our specification.The second approach was to use a step function, allowing for interactions between the variables. We then estimated the out-of-sample RMSE over several train/test splits, which tells us how well our model performed on data that was not originally used to build the model. Lastly, we used an approach known as K-Nearest Neighbors, which estimates house prices by taking averages across observations.

The output of our models can be found in the appendix of this write-up. Here, we focus on the root mean squared error (RMSE) of our models:

RMSE: Step Function

```
## [1] 63322.26
```

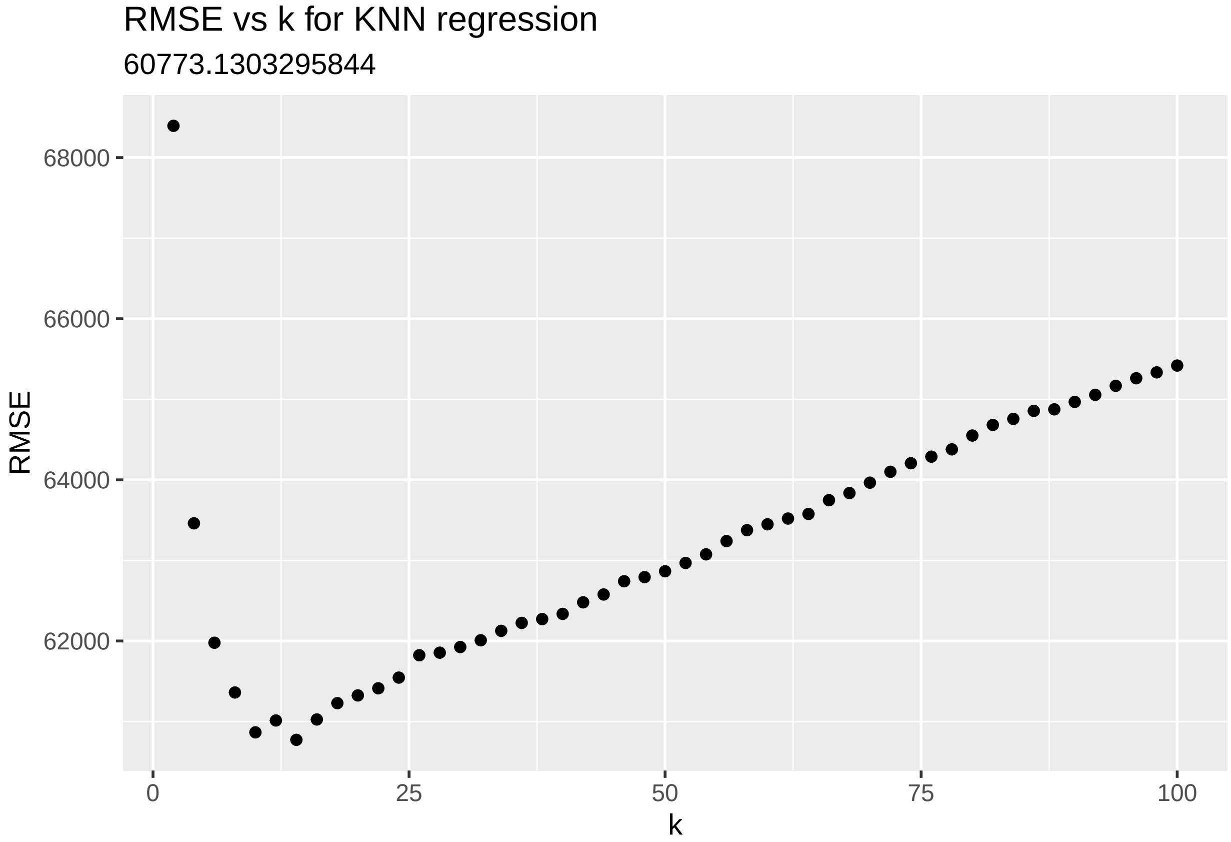
RMSE: Hand-Selected Model

```
## [1] 58715.57
```

Above are the RMSEs for step function and hand-selection approaches. These values give us an idea of how wrong our models were when it came to predicting house prices. We can see that the model that we built manually by inspecting the statistical significance of the variables in the "full" model outperforms the step function model considerably, which we found surprising.

RMSE for KNN Over a Range of K

Now we turn our attention to the performance of the KNN approach. To evaluate its performance, we plot the RMSE over a range of values for k. This will clue us in to the optimal value of k to use for prediction.



Above, we see the RMSE plotted against various values of k. The lowest RMSE across this range of K was just above 60,500, which was significantly higher than the RMSE of our manually built model. Values of K beyond 13 or so begin to be increasingly incorrect.

In conclusion, the model which performed best was our hand-selected linear model, which used all of the available variables with the exception of sewer, fuel, heating type, number of fireplaces, and the percent of college students living in the proximity. As a result, we recommend that appraisals not take these into consideration when trying to determine the tax value of a property. These may cause the estimates for a given house to be biased, resulting in either too little or too much of a tax being levied on the property.

Appendix

Here, we present the outputs from our model. This will be of particular use to appraisers who would like to know how a particular characteristic being present in a home might affect its property value.

Output From Hand-Selected Model

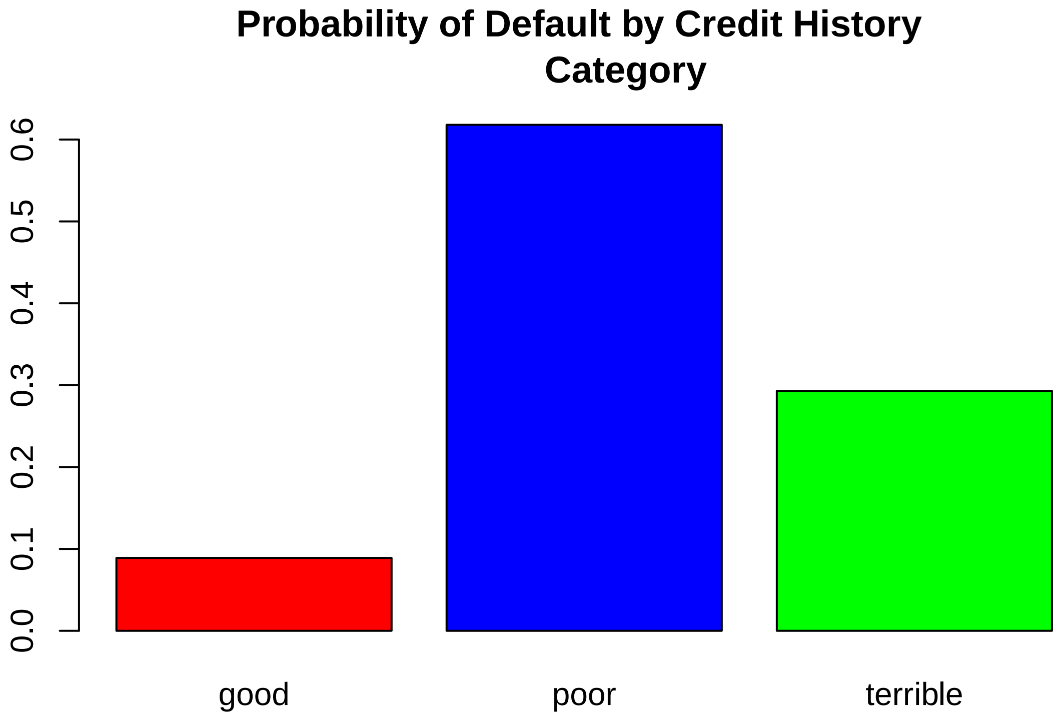
	Dependent variable:
	price
lotSize	8,334.667***
	(2,234.446)
age	−146.684**
	(61.184)
landValue	0.961***
	(0.055)
livingArea	71.942***
	(5.134)
bedrooms	−10,021.040***
	(2,883.729)
bathrooms	22,327.990***
	(3,774.235)
rooms	3,317.544***
	(1,098.625)
waterfrontNo	−107,700.600***
	(19,114.960)
newConstructionNo	49,638.780***
	(8,117.619)
centralAirNo	−12,053.560***
	(3,723.436)
Constant	85,643.090***
	(22,258.170)
Observations	1,382
R^2	0.642
Adjusted R^2	0.639
Residual Std. Error	59,695.930 (df = 1371)
F Statistic	245.954*** (df = 10; 1371)
Note:	*p<0.1; **p<0.05; ***p<0.01

We can see that an additional unit in lotSize increases the price of a home by about $7,400. The age of a home is inversely related to its value, with each year depreciating the value of the home by about $160 dollars. Bedrooms also had a negative impact on price; we believe this could be due to the fact that as the number of bedrooms in a home increases, the amount of open living space decreases. Not surprisingly, the number of *rooms* is positively related to the price, with an additional room being associated with a higher price by about $2,900. A waterfront property is expected to be about $118,000 more valuable than a home not on the waterfront. Additionally, not having central air is associated with a price that is about $11,000 lower on average.

Problem 3: Modeling the Probability of Default

We're asked to build a predictive model for an individual defaulting on a loan, given a range of characteristics that have been recorded for that individual. In particular, we want to focus on the variable "history", which quantifies an individual's history using a range of categories: good, poor, and terrible. The purpose of this problem is to assess how this variable performs in the context of the predictive model.

First, we build a bar plot of default probability by credit history:



We can see that about 10% of the defaults in the data set come from people with a "good" credit history. About 60% of the defaults are attributable to people with a "poor" rating for credit history, and about 30% are attributable to people with a "terrible" credit history. It appears that people with a rating of "poor" are over-represented in the data, and "good" ratings are under- represented.

Next, we're asked to build a predictive model of the probability that an individual will default using the duration of the loan, the amount of the loan, the installment plan, the age of the applicant, their history, and the purpose of the loan as independent variables. These are presented and discussed on the next page.

Results of Predictive Model for Probability of Default

	Dependent variable:
	Default
duration	0.025^{***}
	(0.008)
amount	0.0001^{***}
	(0.00004)
installment	0.222^{***}
	(0.076)
age	-0.020^{***}
	(0.007)
historypoor	-1.108^{***}
	(0.247)
historyterrible	-1.885^{***}
	(0.282)
purposeedu	0.725^{*}
	(0.371)
purposegoods/repair	0.105
	(0.257)
purposenewcar	0.854^{***}
	(0.277)
purposeusedcar	-0.796^{**}
	(0.360)
foreigngerman	-1.265^{**}
	(0.577)
Constant	−0.708
	(0.473)
Observations	1,000
Log Likelihood	−534.977
Akaike Inf. Crit.	1,093.954
Note:	*p<0.1; **p<0.05; ***p<0.01

Regarding the coefficients on the “history” categories, we can see that the coefficient on “poor” implies a greater probability of defaulting than being in the “terrible” category, which is a somewhat unexpected result. This is likely due to the fact that people with poor credit histories are over- represented in this sample. It might be more helpful to have an indicator variable coded as “good” or “bad”, as it’s not clear how much of a difference there is between people with poor or terrible credit histories; the average difference between someone with a good and poor rating might be much greater than the average poor rating when compared to a terrible one.

Question 4: Predicting Children for Hotel Reservations

We were tasked with building three models to predict whether a reservation will have a child as a member of the party. The first being a baseline logistic regression which only considers the market segment, number of adults, customer type, and whether the party booking the reservation is a repeated guest. The second model, which we call our big model, is a logistic regression which considers all of the variables available for making predictions. Our third model, which we build ourselves, uses many of the variables available, but also includes interactions between them. Additionally, we construct an indicator variable, *weekend*, equal to 1 if the reservation is for a weekend.

To assess the out-of-sample performance, we'll partition *hotels_dev.txt* into training and testing sets. Then, we'll use each model to predict each outcome for each observation in the testing set, and evaluate its performance using a confusion matrix. By summing the diagonals and dividing this by the total number of observations, we can get an idea of how accurate our model is. The outputs for our baseline and best models are presented at the end of this section to improve readability. Here, we again focus on the predictive accuracy of the models instead of the coefficients in the output.

Out-Of-Sample Performance of Three Models

Baseline Model Accuracy

```
## [1] 0.918
```

Big Model Accuracy

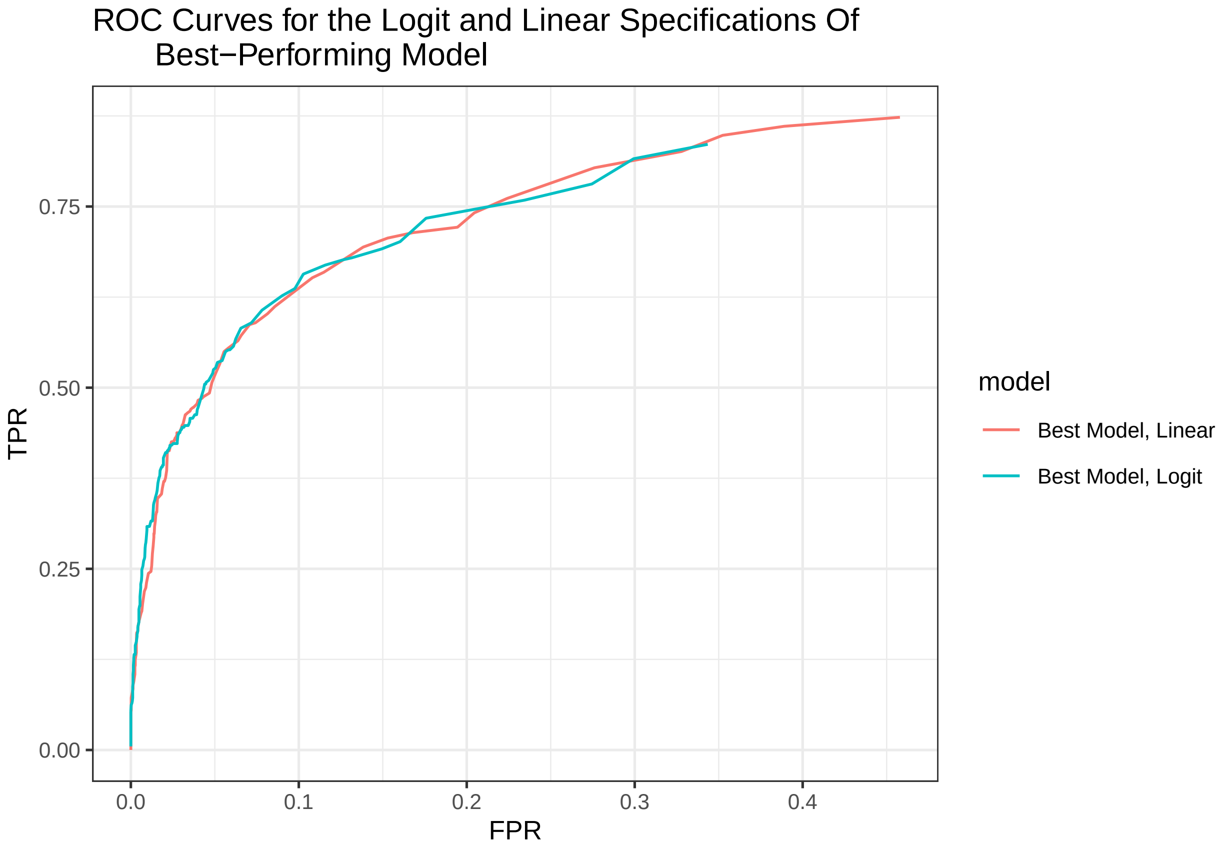
```
## [1] 0.935
```

Best Model Accuracy

```
## [1] 0.937
```

Above, we have the out-of-sample accuracy for our baseline, big, and best models respectively. Our best model slightly outperforms both the baseline and the big model. We attribute this to the fact that we used the "date of arrival" variable to create a weekend variable. We figured that this would be a useful variable for prediction because children are far less likely to be traveling with their parents on weekdays, when they are expected to be at school in the morning. We also included an interaction between hotel and lead time, because we figured that parents who are planning a vacation with children might be more likely to schedule farther in advance, particularly if the hotel is a resort.

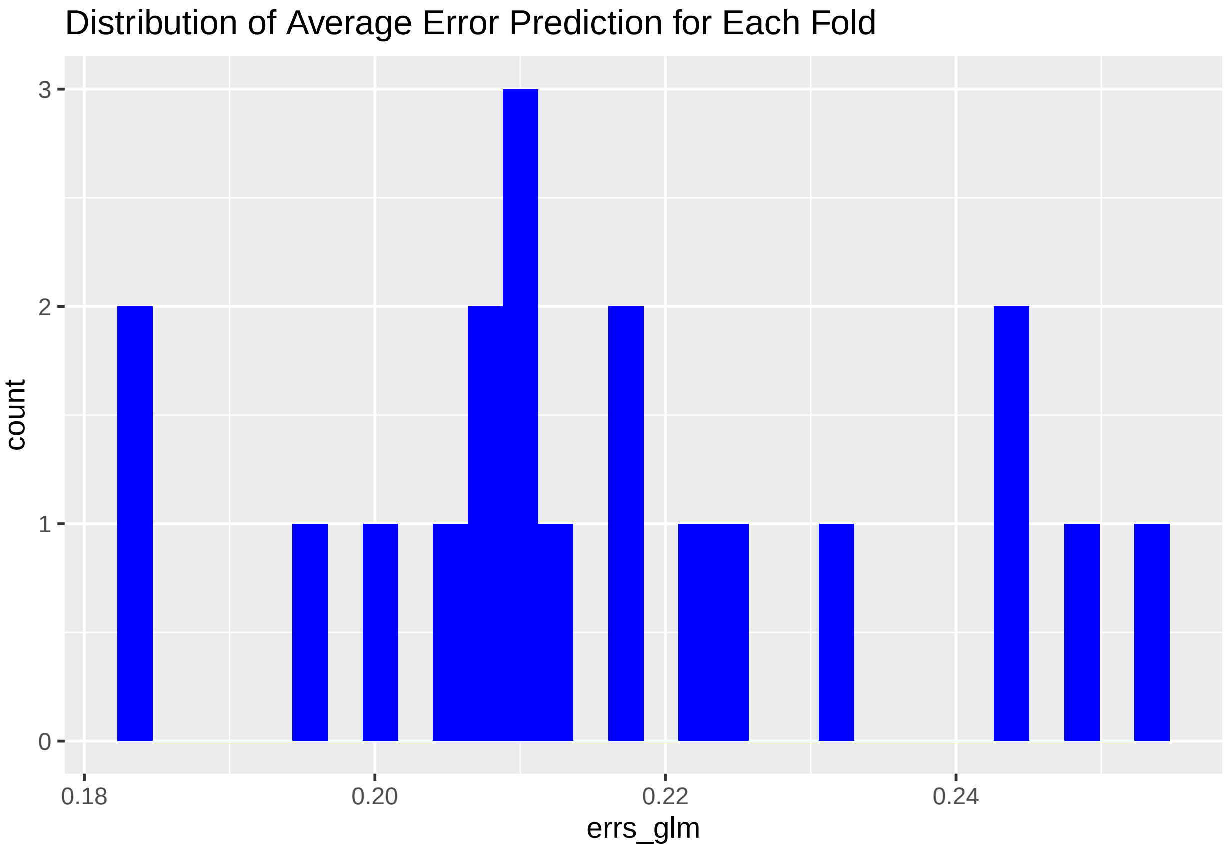
ROC Curves for Highest-Performing Model



Here, we have the ROC curves for our best model, in logit and linear forms. It's not clear which performs better, so we have decided to include both. We can see that the true positive rate can get to about 60% before the false positive rate gets above 10%. In fact, the false positive rate never gets above 50% over the range of threshold values tested.

Next, we use K-fold cross validation to assess our model performance. We partition the data randomly into 20 folds, with about 250 observations in each. This allows us to simulate a busy weekend for a hotel and gives us an opportunity to test the model using observations from outside of the sample that was used to build it. Below is a histogram of the average error per fold.

K-Fold Cross Validation: Average Error Per Fold



We can see these fall in a range between .18 and .30 for all twenty folds, indicating that our predictions are between 70% and 88% accurate. This represents a 6% to 25% decrease in accuracy, and was a wider range than what we would have expected given our performance on the testing set at the beginning of the problem. This could be a result of unobserved differences in the observations between hotels_dev.txt and hotels_val.txt.