Predictive Police Ticketing: Using Machine Learning to Detect Patterns

Nathan Siefken

2/3/2019

Motivation

Imagine that you are in downtown Los Angeles and you are the waiting for your name to be called at the building department on a Wednesday. Your business project is on hold pending these building permits. It is 12 noon and your number is about to be called, but this took longer than you expected, and your parking meter is about to run out. What do you do? Or maybe you need to run into the Santa Monica grocery store to buy milk for your baby at 8am on a Saturday. You find a parking spot just in front, but it's turns out to be a red curb. The cry of your baby is pulling at the threads of your maternal/paternal instincts. Is it worth a ticket? Are you likely to get a ticket?

I am not encouraging you to break the law. In fact, you have 0 chance (assuming no human police error) of getting a ticket if you follow the rules. However, We will learn that there are about 130,000 tickets issued just this past month and it is likely that some of these people were weighing their options prior to receiving a ticket. Life is full of circumstances, and a person may want to have more information to help aid them in the decisions.

Process for our Prediction Project:

- **Know our Goal:** In this case we want to predict how many parking tickets are being issued at any hour of the day, on any day of the week.
- **Data Wrangling:** Structuring data to create more useful output.
- **Implementing Exploratory Analysis:** Determine if there are any sort of patterns in our data before going into building the models.
- **Creating and Test Baseline Model:** This model is our point of reference. The RMSE of our other models should be better (lower) than this model to prove they are learning.
- **Create and Test Linear Regression Model:** Determine if the dependent variable interacts with the independent variables in a linear fashion. How well does this model predict?
- **Create and Test Regression Tree Model:** This model uses recursive partitioning to separate data in to smaller regions that are similar. Discover if the data interacts in complicated nonlinear ways. Is this our best prediction model?
- **Cross Validation:** Decide if our best model suffers from overfitting and change the cp accordingly.
- **Conclusion:** Compare the results of our models and conclude the usefulness of our best algorithm

Measuring our models:

I use Root Mean Squared Error or RMSE to measure the predictions of my models. If you are familiar with Kaggle, the online community of data scientists and machine learners, they too, use RMSE or MAE as the metric for judging their competitions.

Data Set

This data set is maintained and regularly updated by Kaggle which is acquired from the city of Los Angeles organization page.

Kaggle Data Set: https://www.kaggle.com/cityofLA/los-angeles-parking-citations/home

Data Wrangling:

I started by sub-setting the dates of our data from December 23, 2018 to January 23, 2019. Then, I restructured the values in the date and time columns into a time series friendly format to easily separate out the days of the week and the hours of the day for each observation; then place them into their own column. I then converted the US feet coordinates into the universal Longitude and Latitude coordinates. We take care of our Null values and the following is the structure of our cleaned data set.

```
## 'data.frame':    130298 obs. of  12 variables:
##  $ Ticket.number        : num  4.34e+09 4.34e+09 4.34e+09 4.34e+09 4.34e+09 ...
##  $ Issue.Date           : chr  "2018-12-23" "2018-12-23" "2018-12-23" "2018-12-23" ...
##  $ Issue.time           : chr  "8:30" "8:36" "8:40" "8:41" ...
##  $ Route                : chr  "340R" "340R" "340R" "340R" ...
##  $ Agency               : int  53 53 53 53 53 53 53 55 55 55 ...
##  $ Violation.code       : chr  "80.73.2" "80.56E4+" "80.69B" "80.69B" ...
##  $ Violation.Description: chr  "EXCEED 72HRS-ST" "RED ZONE" "NO PARKING" "NO PARKING" ...
##  $ Fine.amount          : int  68 93 73 73 73 73 68 363 68 73 ...
##  $ Longitude            : num  -118 -118 -118 -118 -118 ...
##  $ Latitude             : num  34.2 34.2 34.2 34.2 34.2 ...
##  $ Weekdays             : chr  "Sunday" "Sunday" "Sunday" "Sunday" ...
##  $ Hour                 : int  8 8 8 8 8 8 8 19 19 19 ...
```

Exploratory Data Analysis:

The goal of our exploratory data analysis or EDA is to find patterns in our data and the variables that are significant in these patterns. Once we understand our data and correctly identified the variables that we will use in our machine learning algorithms we will be able to start the modeling process. A good EDA can be used to support the results of machine learning models.

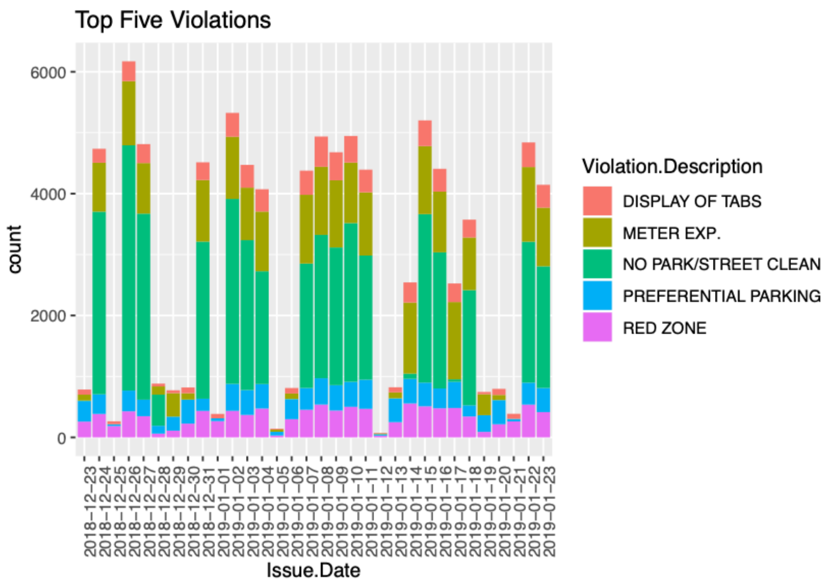
To start, we learned from the structure of our data above, there are a 130,298 observations which are the number of tickets issued. Below I calculated the revenue these tickets generated.

Revenue
9221737

That is $9,221,737 of guaranteed revenue. This is of course assuming that the tickets are paid in full and on time. Otherwise additional fees or penalties may be garnered.

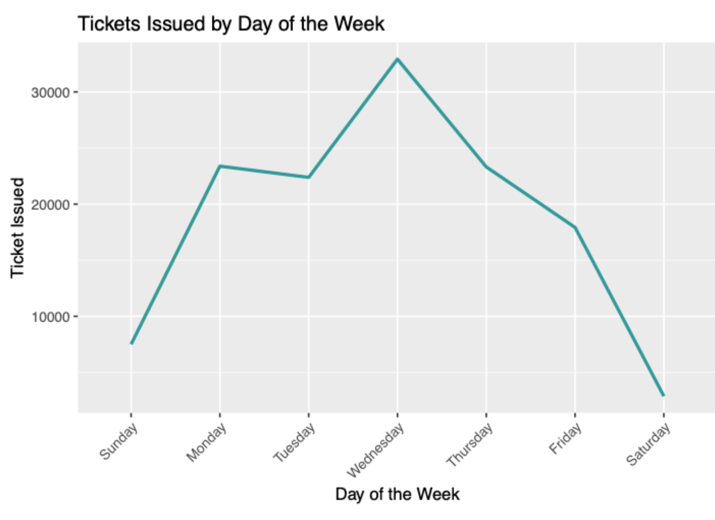
Bar Chart Of Top 5 Violations Issued

And here we see a bar chart that visualizes the distribution of violations that account for the majority of the tickets issued on each day from December 23, 2018 to January 23, 2019.



Line Plot For Total Tickets Issued

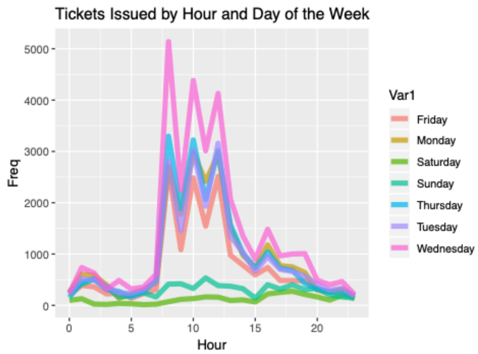
This line chart is the perfect visualization to show the total number of tickets issued for each day of the week. We can easily see the days have high volumes of tickets issued and days which have lower amounts of tickets issued.

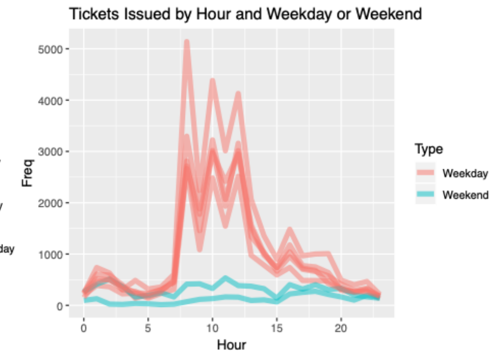


Line Plot For Day Of The Week And Hour Of The Day

To expand on the previous line graph, I included the hours of the day. To do this I charted on the graph to the left, each day of the week by color and the frequency of the tickets issued each hour. Now we can easily see the number of tickets being issued each hour of each day.

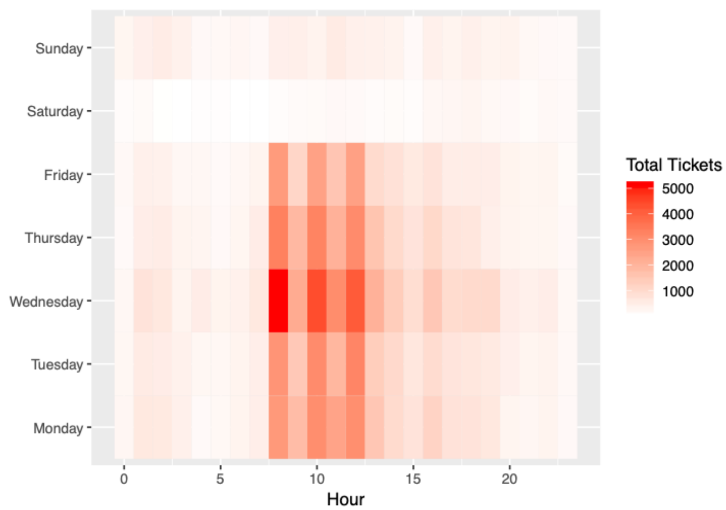
Then on the graph to the right, I charted the weekdays and the weekend by color and the frequency of the tickets issued. This gives a nice comparison between the number of tickets issued on weekdays versus weekends.





Heat Map For Another Interpretation

Let's now plot a heatmap to visualize the data in a different manner. The heatmap is sorted by days of the week and hours of the day. The frequency of tickets issued goes from white (low) to red (high) to give a good understanding of the frequency of our data.



Insights from the Visuals

The EDA illustrates patterns between our variables and leads us to believe that we should be able to predict frequency of police ticketing by the hour of the day and day of the week.

Machine Learning Methods

Before I start the machine learning process I partition the data into a training and test set. Our models will learn from the training data set and then make predictions on the test set, data it has never seen.

We are evaluating our models with the RMSE. Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). In other words, it tells us the difference between our predictions and our values we are predicting. The following is the RSME equation.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i - f_i}{\sigma_i}\right)^2}$$

The following code generates a function that will calculate the RMSE for actual values (true_Frequency) from our test set to their corresponding predictors from our models:

```
RMSE <- function(true_Frequency, predicted_Frequency){
  sqrt(mean((true_Frequency - predicted_Frequency)^2))
}
```

Baseline Model

We will start with our baseline, the most basic prediction model. In statistics this would be $\hat{\mu}$ which is the average for all hours across all days of the week and use this average to predict our ticketing.

```
## [1] 808.3182
```

Now that we have our $\hat{\mu}$ we can determine RMSE for our baseline method.

```
Baseline_rmse <- RMSE(test$Frequency, mu_hat)
Baseline_rmse
```

```
## [1] 656.3121
```

We are getting a RMSE of about 656. Our prediction is on average 656 of citation off the actual amount of citations that are given for each day. This is the case for a couple of reasons, there seems to be days of very high amounts and then days that are very low tickets that are given.

method	RMSE
Baseline	656.3121

Linear Regression

Linear Regression is a global model, where there is a single predictive formula that is used to determine an entire data-space. When the independent variables interact with each other in linear fashion this model works really well.

Now let's run a linear regression model to see if we can improve on our baseline model

$$Y = a + b_1X_1 + b_2X_2 + \varepsilon$$

Y is your prediction, a is the intercept, b is the slope, X is the observed score on the independent variable and ε represents the residuals.

```
## [1] 462.2858
```

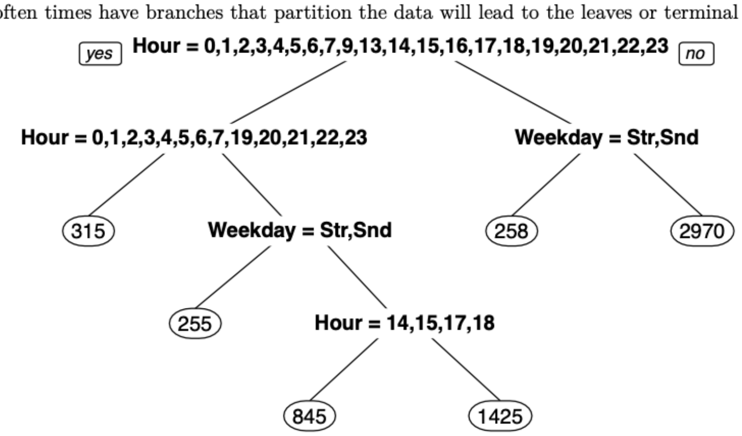
This is an improvement on our baseline model. This model appears to be learning and there is to at least some extent some linear correlation between the variables.

method	RMSE
Baseline	656.3121
Linear Regression Model	462.2858

Regression Tree

Let's see if we can beat the Linear Regression Model with a Regression Tree Model.

Regression Trees use recursive partitioning to separate data in to smaller regions that are similar. This method works really well when the data interacts in complicated nonlinear ways. The tree will start with a root node often times have branches that partition the data will lead to the leaves or terminal nodes.




```
## [1] 242.0429
```

Let's interpret this tree using the two examples from the beginning.

The first person's meter was about to expire at noon. Since, "12" is not listed in the root node, you would go to the first branch on the right. It was on Wednesday, which leads you to the far-right leaf or terminal node, 2970. This means at this time day on this day of the week, our model predicts that there are 2,970 tickets being issued. This is the peak ticketing time, and if this person wants to avoid a ticket they need to get to the meter and put in more money.

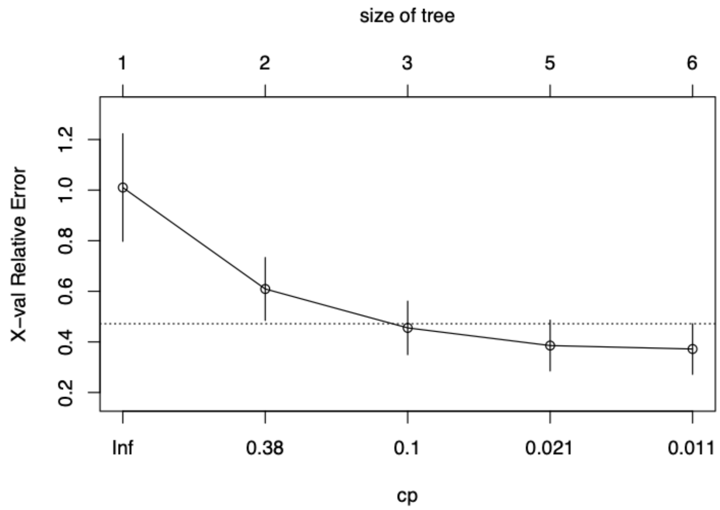
The second person is considering illegally parking at 8am, which is also not in the root node. Again, we will progress to the right side of the tree. It is a Saturday; we would move left leaf from this branch. In this circumstance, there are only 258 tickets being issued at this time for this day of the week. According to our model the likelihood of receiving a ticket are the lowest among all the days and all the times of day, perhaps this would be the time to take the risk.

To take the risk or not, is not for me to decide, I am just putting forth additional information for that person to better understand the risks and help them make their decision.

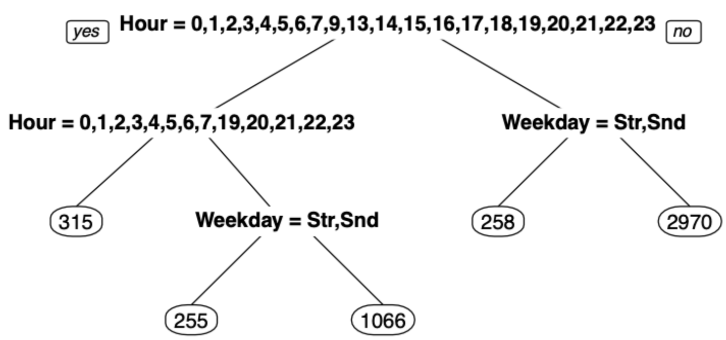
Cross Validation

When using machine learning, it is important to be mindful of overfitting or underfitting models. Cross validations allow for an opportunity to understand our tree better while finding out if our tree is the right size.

We can evaluate our tree by plotting the cp. Our tree has 6 terminal nodes and has a cp of .011. The graph below shows us that the first split gives us the largest improvement and as we continue to split the smaller the improvement.



It appears that we have diminishing returns when we have 3 or 5 terminal nodes. I am going to prune our tree to have 5 leaves and see how it predicts our test set. **Note:** If there is a small difference in the RMSE, this would be a sign of overfitting and pruning would be necessary even though we would have a slightly larger RMSE.




```
## [1] 319.4877
```

There is a big difference in RMSE, indicating that our original tree is the proper size.

Results

method	RMSE
Baseline	656.3121
Linear Regression Model	462.2858
Regression Tree Model	242.0429

The results demonstrate that are models were in fact learning. The Linear Regression model was able to improve from our baseline, verifying there are some linear correlation between the independent variables and the dependent variable. Although our data has some linear correlations, ultimately our data prove to be more complicated and non-linear, this is the reason our tree model out performed our Linear Regression model.

Conclusion:

We have achieved our goal to effectively used machine learning methods to detect patterns of police ticketing in Los Angeles and successfully predicted the amount of tickets being issued on any given hour, on any given day of the week. We can be confident in our results because we can support our results with our exploratory data analysis.

We also showed how machine learning can benefit everyday decisions. As explained in the Regression Tree section, the person at the building department has a high chance to receive a ticket and would be remiss if they did not go to the meter and add money. The second person was at a low risk of getting a ticket and may decide this amount of risk is worth taking.

Shiny application:

I created a interactive map with a shiny application. You can visit it at the URL below.

https://nathans.shinyapps.io/LA_Parking_Violations/