When Is The Best Time To Buy Concert Tickets on Stubhub?

A machine learning project by Daniel Stein, Devon D'Apuzzo, Jackson Middleton, and Maryssa Sklaroff

Final Report
June 8, 2016

Data Collection:

A large portion of the time spent on this project was in data collection. We built a scraper using Node.JS to access the Stubhub API periodically. Node was an excellent framework to use for this since it's easy to schedule asynchronous calls over periods of time. Once we were able to write the code to collect one observation, we could simply set a recursive timeout at a desired period.

The Stubhub API returned a JSON document with all of the parameters that we requested, such as pricing breakdown by section, percentiles of pricing, quantity of each type of ticket available, etc. Then, we saved the JSON document in a mongoDB database hosted for free by mlab.

The scraper worked exactly as intended, but we ran into some issues with hosting the program. The Stubhub API blocked both AWS and Heroku IP addresses and returned a 403 error. Initially, we ran it on a desktop computer at one of our group members home. This worked well for the most part, except there were a few times that the computer crashed or updated, interrupting our scraper without us knowing. Because of this we have a few small periods of missing data. After this happened a multiple times, we bought a Raspberry Pi computer and installed Ubuntu on it. Once this was set up, we experienced zero interruptions in data collection.

The git repository for this scraper is: https://github.com/djdapz/eecs349_project . index.js contains the main scraper file. dataFormatter.js contains the data processing file. This retrieves all of the JSON documents from that database, extracts the attributes that we decided to use, and converts them to .csv files for us to use. This was a critical piece of the project since, at the end of the data collection, we had collected over 15,000 observations. The dataFormatter processes them all in a matter of seconds!

Data Analysis:

We looked at a range of attributes surrounding Stubhub ticket sales to a number of music festivals. The festivals we looked at were Outside Lands, Lollapalooza, Spring Awakening and Governor's Ball. The attributes we collected were the time the ticket was bought (in relation to days until the festival), the type of ticket (General Admission or VIP, single day or 3-day), the minimum ticket price listed, and the number of tickets available. Since we collected data every hour, we originally looked at a data point every hour. However, we found that the ticket price did not change substantially enough every hour to help us, so we instead restructured the data to analyze attributes every six hours.

After analyzing the data we determined that ticket buying patterns were an important attribute. In order to incorporate the trends surrounding each attribute from the raw data, we created three tracking attributes. These three attributes were the difference in ticket price from one data point previously (6 hours), five data points previously (30 hours), and ten data points previously (60 hours). We also looked at the change in number of tickets for these three time stamps.

We decided to combine the data from all the festivals for every type of ticket so that we could predict prices of tickets in general on Stubhub, instead of for one specific ticket type of a specific event. Using multiple festivals also helped us train and test our data. Some of the festivals took place while we were collecting our data and some still have not yet occurred. Therefore we can look at the complete data sets from the festivals that occurred while we were

collecting our data and use what we learn from that data to predict whether or not it is a good time to buy.

When classifying our data we looked at two possible outcomes “good time to buy” and “bad time to buy.” We originally defined “good time to buy” as a time when the price of the ticket would not be any lower in the future. This does not mean the minimum ticket price overall (which would only provide us with one “good time to buy” output), but instead anytime that the ticket price will not be any lower in the *future.* These good times to buy are represented by the local minima in a graph of ticket price versus time until the festival. However, after working with the data we found that this was not going to be a good indicator of whether or not it was a good time to buy because it was difficult to specify in our data especially on the incomplete data sets. Instead we looked at each individual festival and ticket type and applied a simple linear function that served as a threshold value to determine whether or not it was a good or bad time to buy.

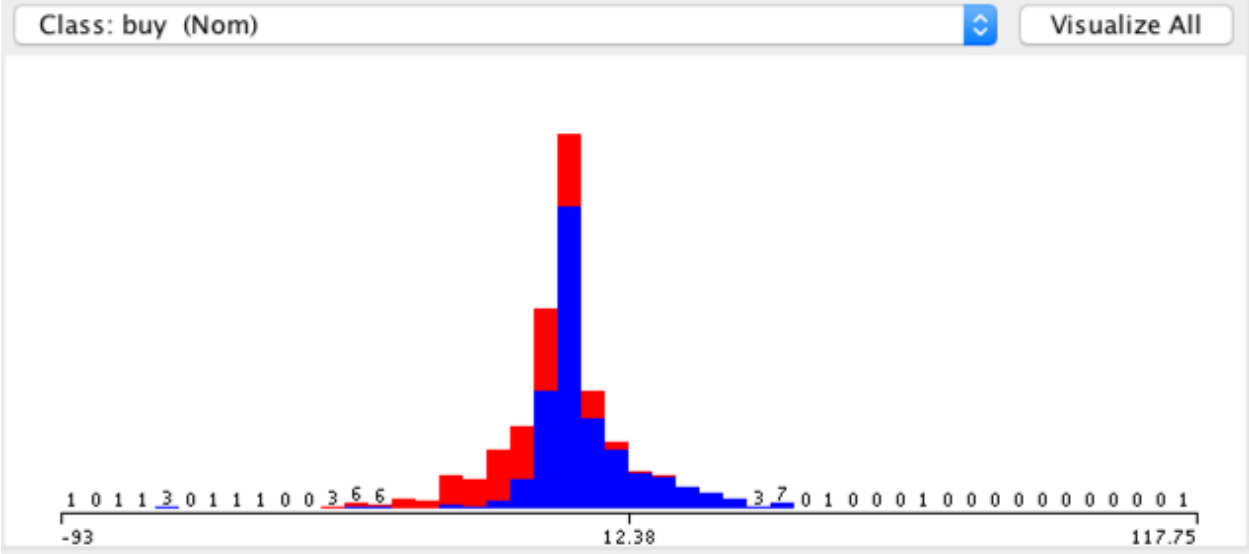
Machine Learning Results:

To implement the actual machine learning we first used a few simple predictors. We just used the previous four data points and then we predicted one week out. We then looked at the actual prices one week out to see which of the predictors would be good to use going forward. We continued to test on small sets of data like this, slightly altering the classifiers we used until we determined the classifiers that were the most accurate. We then used these classifiers on bigger and bigger sets of data. Finally we used the best classifiers on all of our data and applied this learning to predict whether or not it was a good time to buy.

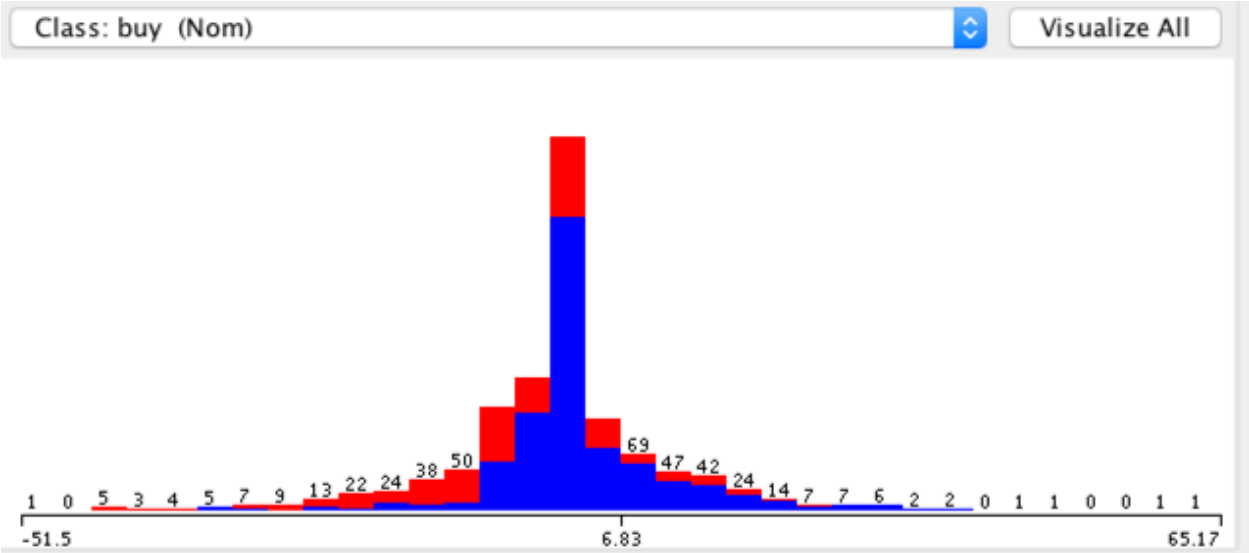
In the end of data collection and consolidation (into six hour increments), we had a total of 1,597 data points (across all festivals and all ticket types). We then split that data into a training set of 1,314 data points and validation set of 283 data points. We randomly pulled the data points for the validation set from all festivals and ticket types. This meant our validation set was roughly 15% of our total data set. We used a range of classifiers, but had the best results with the following:

Algorithm	Training Set Accuracy	10-Fold Cross-Validation Accuracy	Validation Set Accuracy
ZeroR	65.30%	65.30%	63.96%
Random Forest	100%	96.50%	90.11%
BFTree	98.17%	91.32%	86.93%
J48	96.65%	93.15%	88.30%

We also were able to identify the most important attributes by looking mostly at the BFTree. The BFTree identified the general admission ticket price 60 hours previously as the most important attribute in determining current ticket price. Next best attributes were the quantity of general admission tickets available and the ticket price 30 hours before. This shows that the ticket buying trends were very important to include.



60 Hour General Admission Price Difference (Red = Buy, Blue = Don't Buy)



30 Hour General Admission Price Difference (Red = Buy, Blue = Don't Buy)

Future Steps:

The next steps to take in this project would be to get more data. Ideally, we would like to have data ranging from the onset of ticket sales up to the start of the festival for each festival we looked at. We started collecting data at different points in the ticket selling process for each festival and two of the three festivals we looked at have not happened yet so we could not have data up until the end of ticket sales. It would be a good next step to start collecting data the day tickets went on sale up until the day of the event. Beyond looking at these specific festivals we would want to look at tickets for other events. Stubhub sells tickets to a range of concerts, sports, and other ticketed events. It would be important to see if all ticket prices acted the same way as the festivals we looked at. If they did we could look at the data as a whole to increase the accuracy of our current work. If not, we could create another attribute that filtered by event type.

In this project, Devon worked most on data collection, and all of us worked together to organize the data, run the classifiers, and write up the final report.