**Final Project Written Component**

**Empire Z Hourly Sale Predictor**

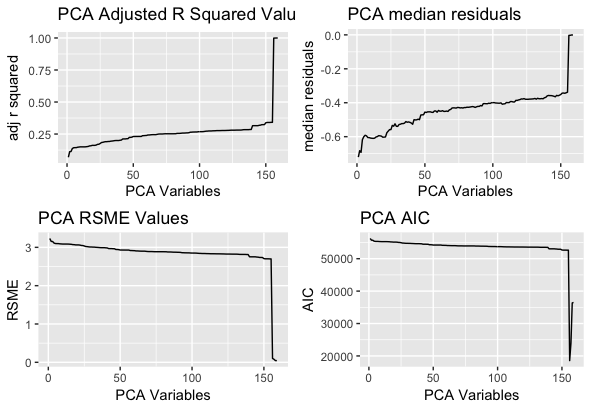
By Sam Robinson

**Project Overview**: The goal of my final project was to take sale information from the game Empire Z (which I am the product manager for) and create a predictive model that can tell us hour by hour how much money we will make based off the sales and events that we schedule using historical data.

**Hypothesis**: Based on previous sale data, sale values and ARMA variables we can create a model that can inform us on the value of future sales before we run them.

**Data Collection:** The data collection is a python script using proprietary functionality called the “Brief System” which is designed for writing one off scripts to help support the game. The specific brief that I wrote took data from multiple dbs and aggregated it into a single data point for each hour the game was live since December of 2015. The data I collected was aggregate sales data, number of players online, number of sessions from users and what users bought with our hard currency called “Gold”. This gives us 10848 instances of single hour data. We then create 24,48,72 and 168 previous hour auto regressive variables so that we can predict what happened today based off yesterday, 2 days ago, 3 days ago and 1 week ago. I believe that sales are very hourly dependent so it would not make sense to look at 12 hours ago. It may make sense to look at things like what is being bought with gold over those time periods but realistically they don’t happen soon enough to be able to react to the data. All of the data collection is proprietary information that I personally wrote for Ember Entertainment, code snippets aren’t currently available for review as it is very closely tied to server technology for the game.

**Results**:



Final adjusted rsquared value: 0.999822258186455

Final RSME Value: 0.0446491888481165

Final median residuals-0.000717021551504995

Best AIC Value: -36505.010890315

Run Time - 0:34:41.63  
  
I’m really happy with both the final adjust rsquared value and the median residual. This makes me extremely confident in using this model to attempt to predict sales that will be run three days out. This would allow us to look at our data on Friday and try to maximize profits over the weekend by running what sales and events we want to run through this model.

The model takes a little bit on the long side to create but not so much so that we couldn’t automate the process of recreating the model every day before work hours begin. I’m satisfied with the results and hope to reproduce this in a way that we can utilize it as an automated process to predict future value as soon as our sales and events are scheduled.

**Pain Points**:

During the creation of this script I attempted to use a genetic algorithm to do feature selection. While I was happy with the direction the genetic algorithm was heading, it was ultimately computationally too expensive to get any data in a reasonable amount of time. At one point I was estimating that it would take 45 days to finish running the r script without even knowing what the output would be. I wanted to do some work to improve run time by optimizing the number of features to select but it proved to be too much work up front given the time constraints of the project due date. I plan on going back and writing a more customized genetic algorithm to optimize on both the features to select and the number of features to select. This would ideally speed up the script by not trying to make a linear regression model with 1200 variables in it. I would also want to be more efficient at selecting features that are consistently important to our model and saving them out to specifically make as a jump off point for future generations. I don’t think the genetic algorithm function I was using was as efficient at this as something customized would have been.

**Continuation of the project:**

One of the major short comings of my submission is that I don’t do any testing for over fitting and I don’t have test and training data. It is just taking what currently exists and fitting data to it very specifically. Ideally, during my model optimization, I would compare the train set to a test set to make sure that my model remains equally accurate on both sets of data. Not doing this means that my model will be very prone to overfitting and may perform poorly when given new data.