Slide 1

Good evening - the name's Albert, and tonight I'll be taking you on a trip into the dark, mysterious world of the online ticket arbitrage market.

DUH DUH

Slide 2

This journey started months ago, in the cold damp basement of my parent's house, when I was broke, unemployed and had a lot of free time. I was interested in exploring ways to develop passive income, and many individuals online seemed to have had some good success with taking advantage of market inefficiencies: Dropshipping on Amazon, flipping cryptocurrency across hemispheres and yes - reselling tickets on Stubhub.

In other words … arbitrage

Slide 3

I bought my first pair of tickets to a Beyonce and Jay-Z concert. Intuition told me that those were a surefire high-value resell. I ended up reselling them for a net gain of…

Slide 4

...$-200

Slide 5

Frustrated, my short experiment with arbitrage ended. But it stayed on my mind

Slide 6

Fast forward 6 months, and I've blossomed into a ripe data scientist. So when the final capstone came up, I knew what I wanted to tackle:

The ticket resale market

Aside from my personal vendetta though, the ticket resale market had a few things going for it that made it an appealing market:

> The first was that it was fully digital. I could analyze, purchase and resell without leaving my seat - which meant it was well suited for automation - a necessity when thinking about passive income flows

Slide 7

> Secondly, there were only a few big players in the space (Ticketmaster for concert sales and Stubhub for the resale market) - meaning the market was reasonably siloed and the information was centralized

Slide 8

> Thirdly, both of the sites had robust APIs which would allow me to access a large amount of data with relative ease

Slide 9

> And finally, although the resale market is saturated with a lot of small players, there just simply isn't enough money to be made for big companies to invest in ways to milk the market. Which means it's a prime opportunity for someone to gain a marginal advantage over other resellers using some predictive magic.

Slide 10

All that was left to do was to turn my unjustified anger into justified science

In approaching this problem, there was only really 1 question I had to solve:

> Can you create a predictive algorithm that could determine which events would have the greatest resale opportunity on the resale market

Slide 11

To further formulate this into a machine learning problem, I had to think about some metrics that could quantitatively allow me to represent 2 important concepts in investing: return and risk - and then to track them over time

For return rate, I calculated this as the percent difference between the original, wholesale price (as listed on Ticketmaster) and the mean of all listing prices on resale market (as listed on Stubhub)

For risk, I calculated this as the DOD decrease in quantity of tickets listed. The higher the expected change in ticket quantities listed, the greater the opportunity for sale and the lower the risk of not being able to sell the ticket.

Both metrics are imperfect and have limitations.

The return rate metric is calculated based on tickets left in the market, not on tickets sold, so it is not a guarantee of going market rate. We can think of problematic instances, for example towards the beginning of an event resell when prices fluctuate wildly as the market finds its selling point, or towards the end of the selling period as the event is about to start and only highly priced tickets remain.

Similarly, change in ticket listing quantity is not indicative of ticket sales. Individuals may be deciding to not sell their tickets, or are perhaps pull the listings down to give them away or sell to friends at discounted rates.

In multiplying the 2 together however, we can manage some of these indicators' limitations.

When avg. price differential is high, it is more likely that a high change in ticket quantity is indicative of sales, as resellers have less incentive to pull their listings. When expected change in ticket quantity is high, it is likely that a high price differential is more indicative of a properly priced market as many of these tickets get sold.

This final unified metric will help us in identifying investment opportunities from ticketed events

Slide 12

So for the rest of the presentation I'm going to jump into the nitty gritty details of the project. I'm gonna take the opportunity quickly here to caveat all of this by saying that what I did was beyond a simple data based analysis. It was more of a rough prototype attempt for building a product, and it's going to have a number of flaws, as many first drafts do

So first will be the data architecture, and discussing how I pulled and stored data from both Ticketmaster and Stubhub to make it usable for my analysis.

The next piece will be taking a look at some of the features, and pulling out some interesting initial insights.

And finally, I'll move on to the modeling, and give you guys a summary of the ensemble model I built as well as some initial results

Slide 13

First up - data architecture

Slide 14

I conceptualized my two databases as 2 buckets:

First as the static data - which was the Ticketmaster bucket and it included things such as event name, location and start date, as well as performing artists and their respective music categories. It also included ticketing information, such as ticket sale start date and the wholesale minimum and maximum ticket prices for any given event.

The second was the time dependent data - which was the Stubhub bucket. And this included data related to listing price and ticket quantity

Slide 15

First, using Ticketmasters API I refined my set of observations to only music events in the US, and after some cleaning…

Slide 16

… dumped the data into a table in a local SQL Database

Next came the tricky part. How to match up our event data and wholesale prices from Ticketmaster, with the time dependent price and ticket quantities on Stubhub's resale market.

Stubhub's data was labeled differently than Ticketmaster's, and it was organized on a listings level - not an event level.

Slide 17

So first I passed all Zip Codes of events in our local Ticketmaster database to the Stubhub API, similarly limiting my search to music events...

Slide 18

… and then pulled the data, organized by listing and stubhub event id, into a local pandas dataframe.

Slide 19

I was then able to call my Ticketmaster Database, and left join the Stubhub listings data with our Ticketmaster event IDs using shared Zip Code and Start Date

Slide 20

We now had a dataframe of all matching Stubhub event IDs, along with our time dependent variables for each resale listing for that event, including price and ticket quantity.

Slide 21

Next, I dumped the data into a new table in my local SQL Database and continued to pull data twice a day for the next 5 days...

Slide 22

… adding a datetime variable at each pull to track the date of the values

Slide 23

Once I was ready to compile data for my analysis, I grouped my Stubhub listings by event id and datetime…

Slide 24

And joined it to my static Ticketmaster variables using the Ticketmaster Event ID…

Slide 25

As a final piece of additional information, I also joined US Census data to add city populations to my dataset

I now had my final dataset for analysis

Slide 26

It's important to note that my data has some clear limitations. For one, all the Stubhub pulls I did were manual.

So although I did do them twice a day, they weren't pulled at the same time every single day, and eventually I ended up skipping a beat on my 6th day, which limited our dataset.

All the pulls were also done on my local machine, and in some instances either my server would fail or the connection to the API would give out, leading to duplicates and strange values

Slide 27

Secondly, because I joined across 2 databases our joined data was inevitably imperfect.

This limited my analysis to an event level rather than a more granular, and undoubtedly more informative, zone pricing level. Unfortunately there was just no reliable way to join the 2 databases' information along pricing tier.

And even then some of the events also did not join well along zip code and event start date. I was able to catch some troublemakers (such as Las Vegas), but I was not able to review the entire dataset manually and it's possible this was responsible for some strange values as well.

Slide 28

BUT… in spite of all that, we were still able to pull out a data set of 4700 music events over 10 instances of time (between 07/29 - 08/03)

Slide 29

Next up, we did some feature exploration and engineering

Slide 30

First we had to clean up some of those funky values in our metrics of interest that I had mentioned. As you can see here, there are some extreme outliers at either end of the distributions.

The values for price differential go as high as 40,000%, and the change in ticket quantity goes as low as -20,000%.

Slide 31

Looking at the distribution in 1% increments however, we see the extreme outliers skewing our distributions are few and far between.

I removed the top and bottom 1.5% of values for both metrics, as well as an additional 5% of values at the low range of our change in ticket quantity metric - which effectively removed all values indicating a greater than 100% decrease in ticket quantity.

As a quick aside, you may be asking where our tertiary, unified metric is. In this section, decisions on outlier removal were made based on our risk and return metrics to allow for a cleaner and more nuanced analysis.

Slide 32

After outlier removal our value distributions look much cleaner

Slide 33

There are certainly still some large values here, but I wanted to be conservative in removing extreme values, as the extremities can often be very informative for our models

Slide 34

Next, I explored the metrics across time. I engineered 2 features: the first is a datetime grouping of my 10 listing pulls, segmented into halfdays.

This would become the unit of analysis for the upcoming time series models.

As you can see, we're a little limited in how much data we have. However, it's good to see that even in this limited time frame the combination of our 2 features helps smooth out some of the extremes in our return and risk metrics, as predicted.

Slide 35

The second feature is actually a pair of features - that is the days from the sale start day, and the days to the event start date.

When plotting these against our two metrics, we see virtually no correlation. The R-score is close to 0 here.

Slide 36

At first glance this might seem concerning if we're trying to run a time series. In actuality it's not really anything to be worried about, and might even be a positive sign.

Time series models rely on stationarity, and the lack of a trend in the direction of time means our data is already way closer to stationarity.

Slide 37

The 3rd and final step in our exploratory journey were the static variables. I decided for presentation purposes to make a Word Cloud, where size is indicative of our models predicted feature importance relative to our unified metric.

I ran significance tests on all features to determine which attributes were most valuable, and then kept those in the analysis.

Here you see which features have the greatest predictive power:

- Max wholesale price, by and large, had the greatest correlation to a higher unified metric. The greater the initial price, the greater the opportunity - which means to a degree artists, venues and promoters ARE pricing in demand - just not completely
- City population was second most important - with large cities having the greatest resale opportunities far and beyond
- And third on the list were how many upcoming events the artists performing had scheduled, with more events in the future decreasing the potential resale value of that event,

Slide 38

I also ran the feature importance word cloud using engineered categorical only variables.

This is a slightly more important visualization, as it informed with how we ended up weighting our ensemble model - and I'll go into that in detail later

Here you see the similar trends I pointed out earlier with one new standout - presales

This insight was probably the most interesting to me, because it turns out concerts that have presales of their tickets have LOWER resale values! I was a little confused about this, but when you think about it deeper the logic actually makes sense.

Presale tickets are typically gated with membership codes or fan offers, which the casual buyer would not have access to. In this way, the seller ensures that the most ardent fans are getting the tickets on a first priority basis, which appears to actually calm down the secondary resale market.

Marketing strategy coming through in the data! Super cool!

Slide 39

And finally, we'll address the modeling

Slide 40

ARIMA models are some of the most effective models in time series forecasting. However they are quite sensitive to external variables. And as we saw - there are a number of those static variables that have quite a high impact on our unified metric values. In order to maximize the predictive power of my model, I would have to find a way to limit the variance of the other external variables ideally without losing their predictive impact as well.

To resolve this, I chose a 2-layer ensemble model for our prediction algorithm.

Slide 41

The first layer is a supervised learning classifier, optimized for recall, which filters out events that do not break a predetermined unified metric threshold (in this case we chose 500 - this was the combination of a 50% change of sale + a 10% return on investment).

Slide 42

After some testing and fine tuning, we chose a Gradient Boosting Classifier with 20 estimators - we found that as estimators decreased our Recall - in other words the ability to identify positives as positives - increased, without affecting our ability to weed out negatives too much

So what this first layer allows us to do is to (1) utilize the combined predictive power of our static variables, and (2) provide a feature importances layer of analysis to inform how we weight our ARIMA models

Slide 43

The second layer is a set of 54 pretrained ARIMA models. Each model's parameters are specified and trained for each of our significant categorical variables' features, and then provided weights. Our initial idea was to weight each model by feature importance, as mentioned earlier, but we opted to weight using AIC values as our scale. The weakest model's AIC score was set to a 1, and all other values were scaled accordingly.

The AIC values represent an ARIMA model's goodness of fit, and we felt that this was a more accurate representation of how much noise was removed by narrowing the model down to that specific feature. In other words, the lower the AIC, the cleaner the time series data and the better the fit.

Ultimately though, when comparing AIC and feature importance, we found there was a general trend that the lower the AIC (in other words the better the fit) the greater the feature importance value was - indicating that both ways of looking at the data aim at the same thing - in other words how much the removal of a features impact narrows and stabilizes the data

Slide 44

So for example, Presales would have 2 trained models - 1 using all the data with no presales, the other using all the data with presales.

Each of these models would have an associated weight.

Slide 45

What it would look like in action is:

1) Our first layer processes our data, and filters out uninteresting events

Slide 46

2) The remaining observations are passed on to our second layer.

Slide 47

Each event is then filtered into 5 different models based on each feature's category value, and each model produces its own forecast.

Slide 48

Each model is then weighted…

Slide 49

And then aggregated into a final forecast. What this structure allows us to do is narrow our ARIMA model training sets and eliminate the noise from external variables, while still taking full advantage of the features available.

Slide 50

Unfortunately, our time series model does not forecast very well. There are most likely 2 reasons why this is the case:

1) Insufficient Data - in exploring readings on the topic, this type of forecast behavior was common in instances where there simply was not a lot of data. Standard model recommendations put 30 samples as the minimum - we had 10

Slide 51

2) Seasonal Trends - many of the PACF charts showed huge discrepancies around lags 15 and onward, which roughly aligns with about late Thursday through Friday. There is some indication here that there is a weekly seasonal trend that we're not capturing. Unfortunately without sufficient data it's difficult to confirm or deny it

There's certainly a good modeling foundation here, but we would have to go back to the board to make some adjustments to our data architecture in order to properly train and test the model

Slide 52

But for fun, I'll give you guys a look at some of the top events from the training set, seeing as they're a little curious. I think it's very obvious all of us have been sleeping on folk music.

Slide 53

Ok, so to wrap it up what have we learned today:

> Data architecture is so important! This was my first time working with a self-made dataset during this program, and real world data is so much messier than our pretty kaggle sets. Setting up the data flows and warehouse structures requires you to account for as many of the data's uses as possible ahead of time, and an improper foundation can ruin your analysis (as it did in this case)

Slide 54

> The data reflects the basics of supply and demand - Events in big cities (high demand), starring artists with few upcoming events (low supply) typically have the greatest opportunity to turn a profit

Slide 55

> As I mentioned earlier, artist and venues utilize presales to limit resale and ensure fans have access to the concert tickets. The data shows that presales do in fact decrease activity in the resale market - so getting your hands on some presale tickets isn't always a surefire strategy

Slide 56

> And I think the last one speaks for itself

Slide 57

Where do we go next?

To the Cloud! Setting up AWS storage containers and live servers to pull and update our databases real time would be HUGE in tackling a lot of our data related limitations, which would ultimately improving our reliability of prediction, and our ability to react to it

Slide 58

More data would also mean being able to spot seasonal trends, and refining our ARIMA models. In the 5 day sample period alone, there seemed to me to be a fairly clear indication of differences in weekend and weekday pricing. With more data, we'd be able to fit more accurate models with longer tail seasonal activity.

Slide 59

Fitting all those ARIMA models manually was a great learning experience, but it's also incredibly inefficient. There are a few packages out there that will fit a time series ARIMA model with incredible accuracy. With live AWS servers, models could be fit real time with the most up to date data, and with a larger data warehouse we could refine our ARIMA models even further. Rather than narrowing our models by 1 feature, why not narrow it down to every sample that shares 80% of an events features - more similar data allows for better models

Slide 60

One thing that this model did not account for was the real time changes in the perception of the artist. This is undoubtedly a huge missing factor for our predictive capability, and being able to stream live Twitter sentiment analysis into our models would be huge