Analyzing Concert Data to Predict Ticket Price Markups

Evan Paul
SF-DAT-20

Background

- Buyers of concert tickets are able to re-sell them on StubHub.com, often at a markup compared to face values
- The price one is willing to pay for a ticket on StubHub is influenced by many factors
  - Is the show sold out?
  - How popular is the artist?
  - How soon is the show?
- Can we use features to predict the price markup of a concert ticket on StubHub?

Hypothesis

Variables such as the number of days until a show, whether the show is sold out or not, and artist popularity can be used to predict the price markup of concert tickets on StubHub.com.

Data

Data Sources

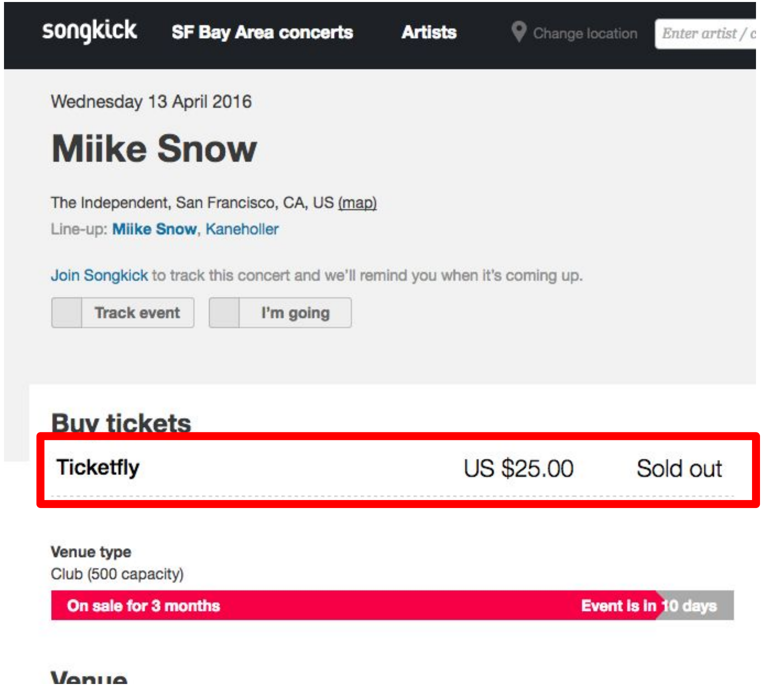
- Data was gathered from 3 primary sources:
  - **StubHub.com API**
    - Event details and ticket prices
  - **Webpage scrapes of SongKick.com**
    - Ticket Face values and whether shows were sold out or not
  - **EchoNest.com API**
    - Artist metadata and popularity data

StubHub API Data

- Artist
- Date of show
- # of days until show *(from 3/13/16)*
- Lowest available StubHub ticket price
- Venue name
- City

Data Gathered from Scraping SongKick.com

- Ticket Vendor
  - E.g., Ticketmaster, TicketFly, EventBrite, etc.
- Ticket Face Value
- Whether the show is sold out or not *(as of 3/13/16)*

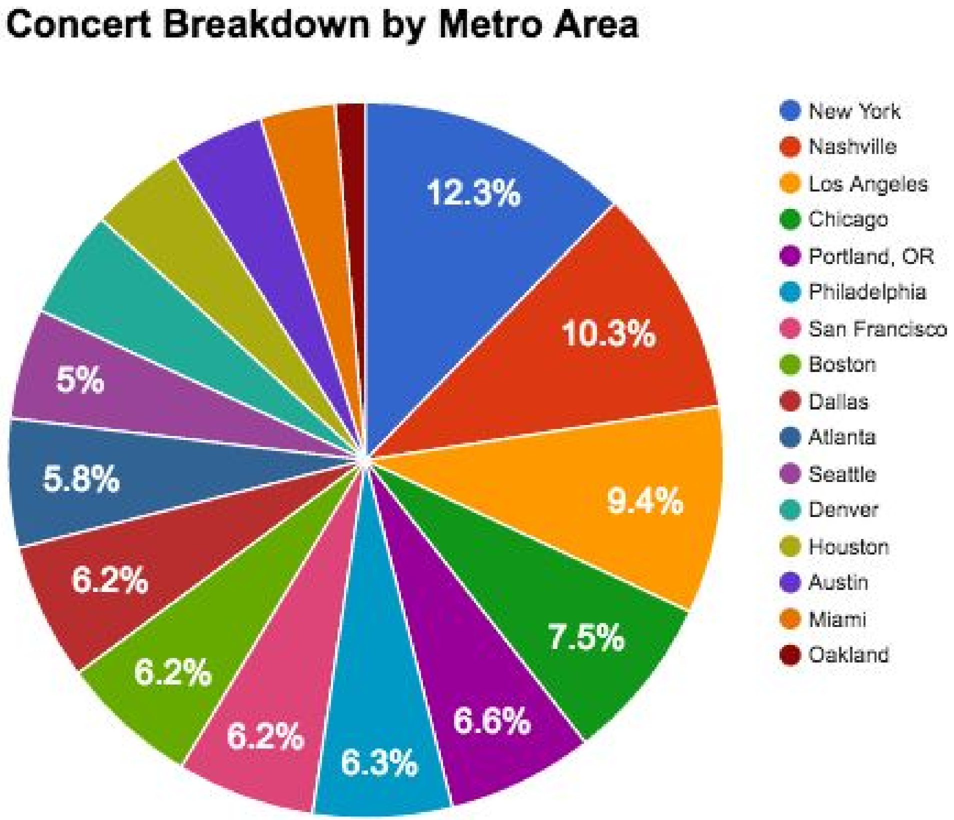


Artist Data from EchoNest

- Artist “Familiarity” score
  - Measures how well known an artist is (cont. values between 0 and 1
- Artist “Discovery” score
  - Measures the current “discovery” level of an artist (cont. values between 0 and 1)
  - I.e., artist who is relatively unknown but is currently getting many plays gets a high score
- Artist “hotttnesss” score
  - Measures how much people are sharing an artist currently (cont. values between 0 and 1)
- Number of blogs published recently about artist
- Number of news articles published recently
- Number of reviews published recently
- How many years an artist has been active

Data Collection

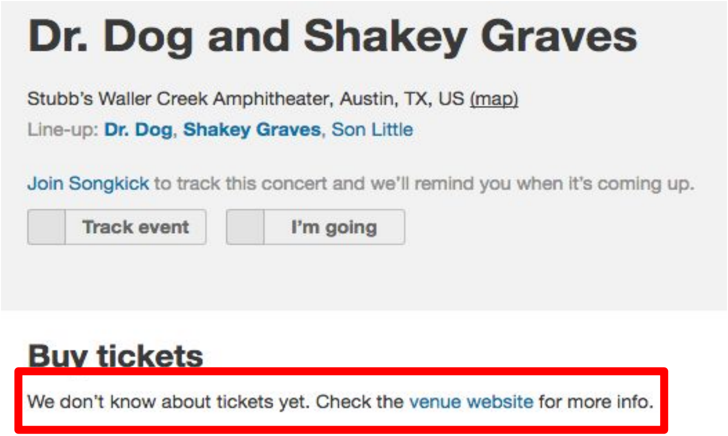
- Collected data for concerts from 16 metropolitan areas in USA
- Resulted in **3,126** concerts total
- All data was collected on March 13th, 2016



Data Limitations and Challenges

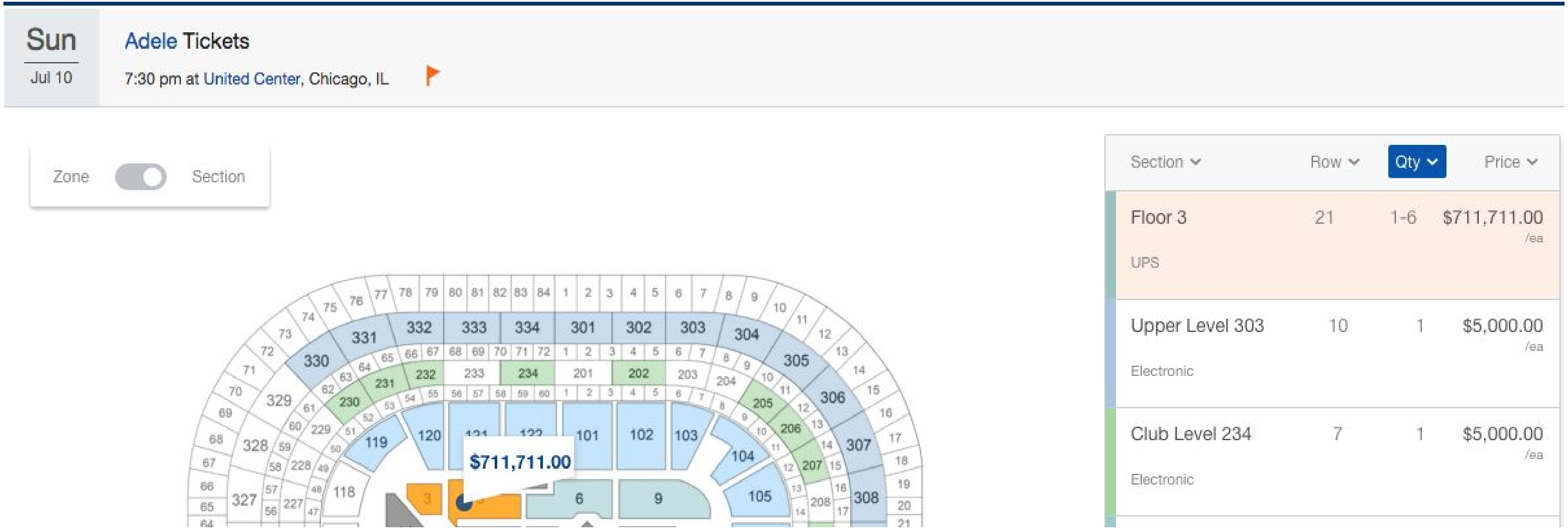
Data Limitations and Challenges

- SongKick did not have complete data for every show
  - 3,126 events total
  - SongKick webpages only had valid ticket info for 1,436 of them



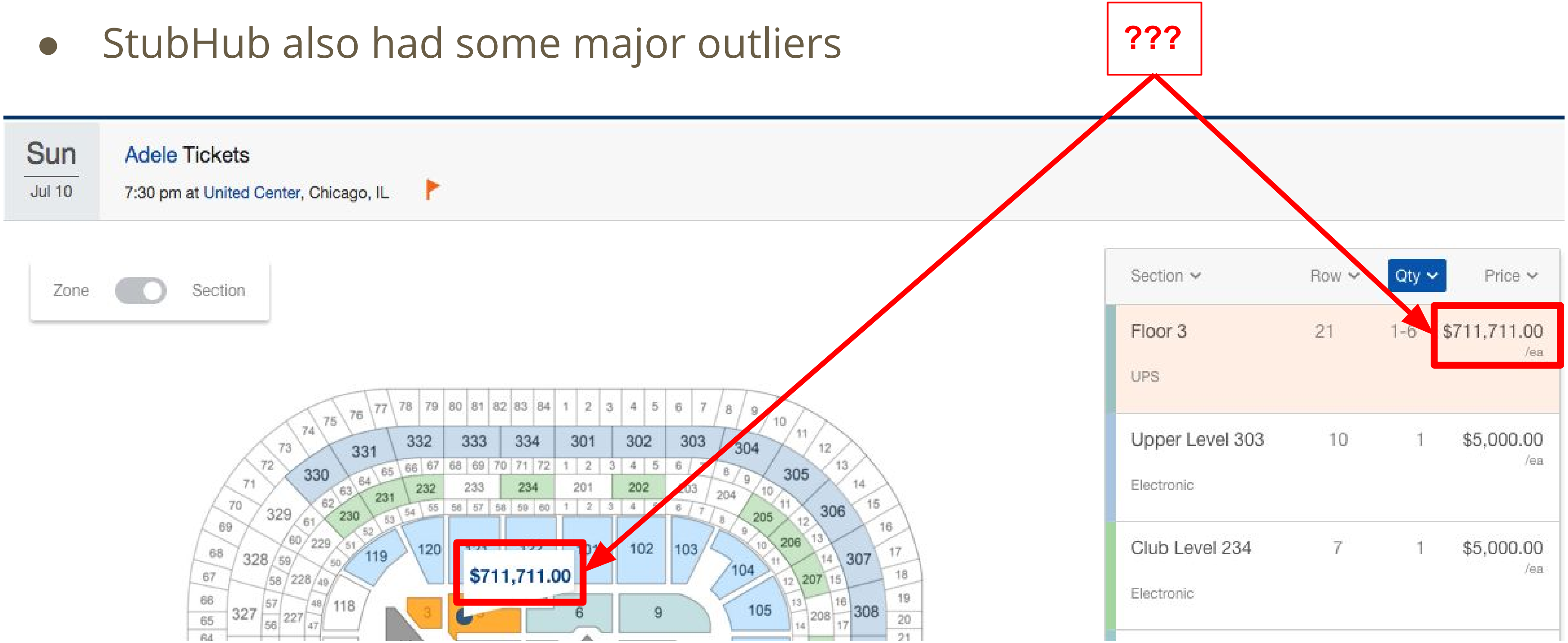
Data Challenges

- StubHub also had some major outliers



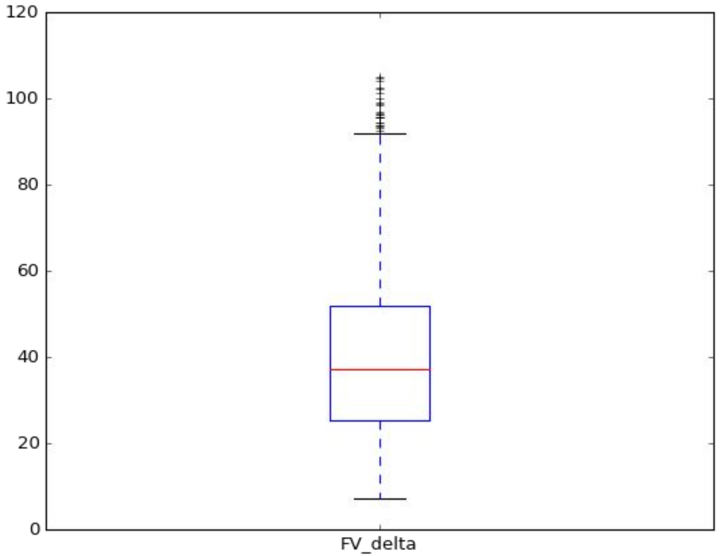
Data Challenges

- StubHub also had some major outliers



Cleaned Data

- After removing outliers and bad data, we were left with **1,192** valid concerts with the following markup characteristics:
- Mean ticket markup: **$40.87**
- Standard Deviation: **20.7**
- Min markup: **$7.26**
  - Charlie Puth @ Theatre of Living Arts, Philadelphia, PA
- Max markup: **$104.90**
  - Robert Plant @ The Moody Theater, Austin, TX



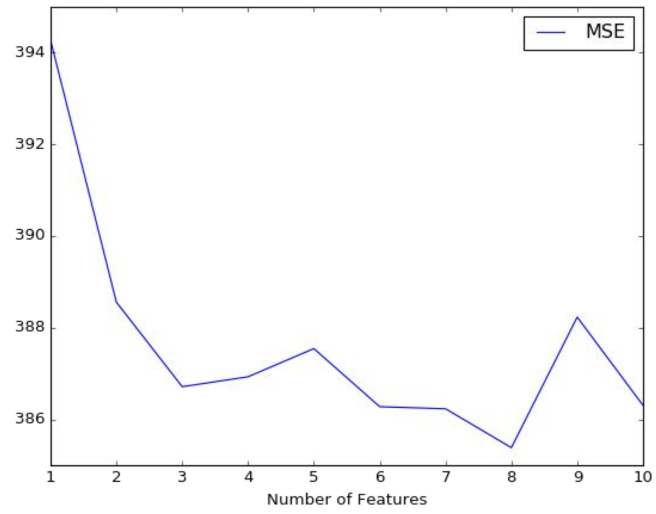
Let’s try to predict ticket markup

Random Forest Regressor

- Advantages of Random Forest:
  - One of the best models for prediction
  - Don’t need to standardize/scale data
  - Only need to tune 1 parameter: max # of features to consider at each split
- Used the following variables to attempt to predict ticket markup:
  - **'face_value', 'sold_out', 'days_to_show', 'num_blogs', 'num_news', 'num_reviews', 'discovery', 'familiarity', 'hotttnesss', 'num_years_active'**
- Used validation (train_test_split) as opposed to cross validation to save computation time
  - With ~1,200 observations and only 10 features, we don’t need CV

Random Forest Regressor

- Tuned model: found best results with 8 max_features and 5,000 trees
- Model with these ideal parameters gave an MSE of ~386.65 and OOB score of 0.11
    - MSE of 386.65 means our prediction was off by about **$19.66** on average
- Strangely, Boosting gave us a higher MSE, even after tuning (~403.15, or $20.08 error)




```
# Check feature importances
sorted(zip(RF.feature_importances_,X.columns.values))
```

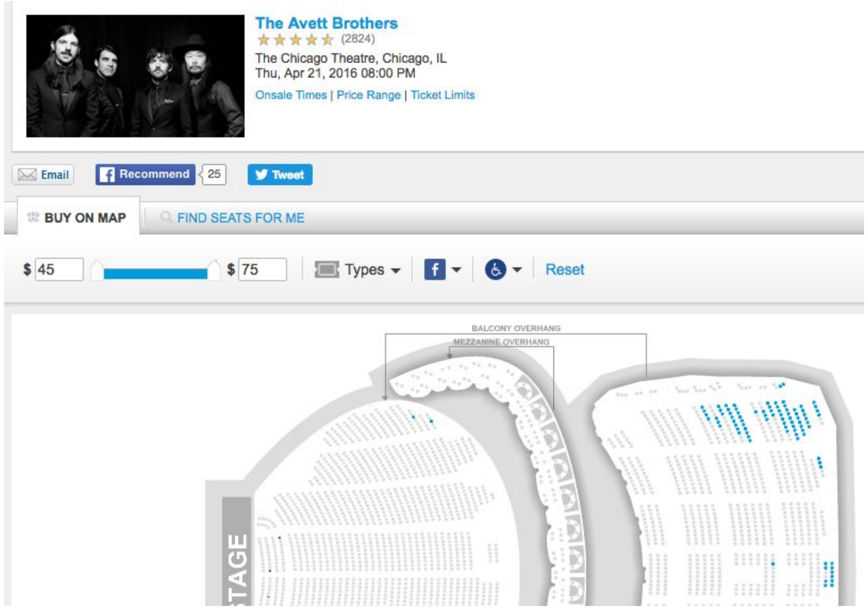
```
[(0.019122638683807328, 'sold_out'),
 (0.076386896156620757, 'num_reviews'),
 (0.088009621941953622, 'familiarity'),
 (0.097252680838548961, 'discovery'),
 (0.098578076025170588, 'num_news'),
 (0.10871390046650277, 'num_blogs'),
 (0.1169508361476027, 'hotttnesss'),
 (0.12461576593001537, 'face_value'),
 (0.13091751663667109, 'days_to_show'),
 (0.13945206717351824, 'num_years_active')]
```

Sample Predictions with our RF Model

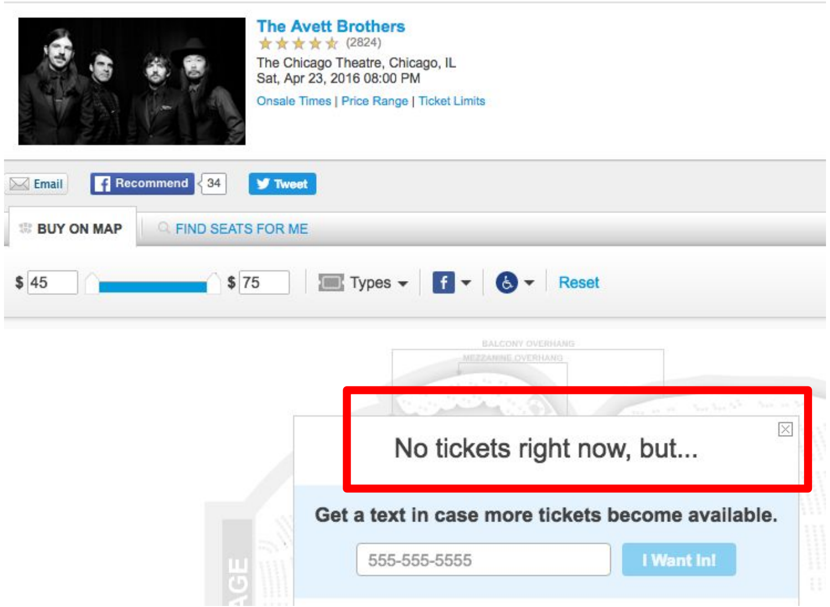
- Avett Brothers @ Chicago Theatre in Chicago, IL on 4/21/2016
  - Ticket Face value: $45.00
  - Minimum StubHub Price: $82.74
  - Actual Markup: $37.74
  - Model's predicted markup: **$36.78** (off by $0.96 - pretty good!)
- Avett Brothers @ Chicago Theatre in Chicago, IL on 4/23/2016
  - Ticket Face value: $45.00
  - Minimum StubHub Price: $120.50
  - Actual Markup: $75.50
  - Model's predicted markup: **$37.01** (off by $38.49 - eh...)
- **Wait! These are 2 predictions for the same artist only 2 days apart! What gives?**

One of the shows is sold out!

April 21st Show - <u>not sold out</u>
(StubHub markup=$37.74)



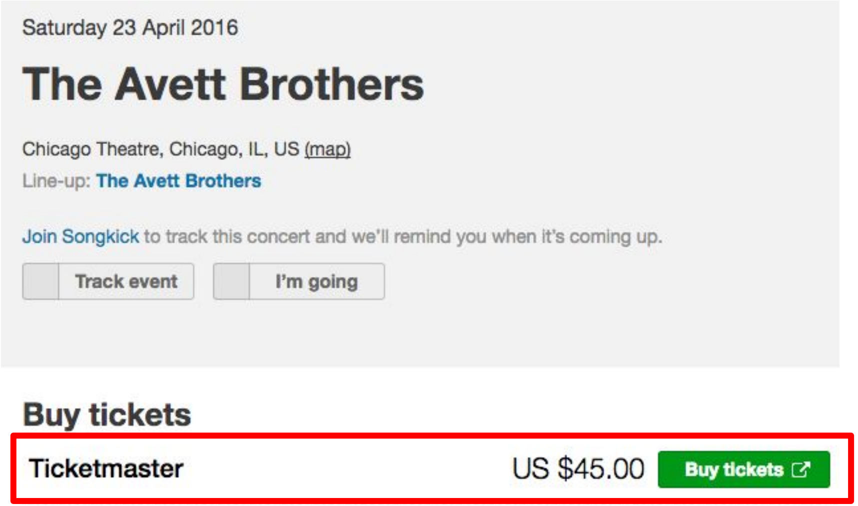
April 23rd Show - <u>sold out!</u>
(StubHub markup=$75.50)



But our data did not capture this...

	date	artist	venue	sold_out
77	2016-06-19T20:00:00-0500	The Avett Brothers	ACL Live at The Moody Theater	1
203	2016-04-22T20:00:00-0500	The Avett Brothers	Chicago Theatre	0
204	2016-04-23T20:00:00-0500	The Avett Brothers	Chicago Theatre	0
214	2016-04-21T19:00:00-0500	The Avett Brothers	Chicago Theatre	0

- **SongKick does not have accurate sold_out status**



Re-run Prediction with correct sold_out value

- Let’s try re-running our prediction algorithm with the correct sold_out value for this show
- Knowing event was sold out, our RF model predicts a markup of **$45.50**
  - We predicted a markup of $37.01 with 0 sold_out value
  - Actual markup: $75.50
  - Not an amazing improvement, but still better
- Lack of correct sold_out values from SongKick may explain why sold_out was an insignificant feature in prediction model

MSE was high with RF Regressor/Boosting.
Can we do better?

Let's try turning this into a classification problem...

- Random Forest didn't allow us to predict ticket prices very precisely
- But maybe we can predict the range that a markup is in
- Let's create buckets for different markup ranges:
    - Bucket 1: $0 - $25
        - *293 observations*
    - Bucket 2: $25 - $37
        - *299 observations*
    - Bucket 3: $37-$52
        - *303 observations*
    - Bucket 4: >$52
        - *297 observations*

```
TicketData['FV_delta_bucket'] = 4

mask_1 = (TicketData['FV_delta'] <= 25)
mask_2 = ((TicketData['FV_delta'] > 25) &
          (TicketData['FV_delta'] <= 37))
mask_3 = ((TicketData['FV_delta'] > 37) &
          (TicketData['FV_delta'] <= 52))

TicketData.loc[mask_1,'FV_delta_bucket'] = 1
TicketData.loc[mask_2,'FV_delta_bucket'] = 2
TicketData.loc[mask_3,'FV_delta_bucket'] = 3

TicketData[['FV_delta','FV_delta_bucket']].head(10)
```

	FV_delta	FV_delta_bucket
0	50.04	3
1	26.99	2
2	16.91	1
3	35.32	2
4	32.37	2

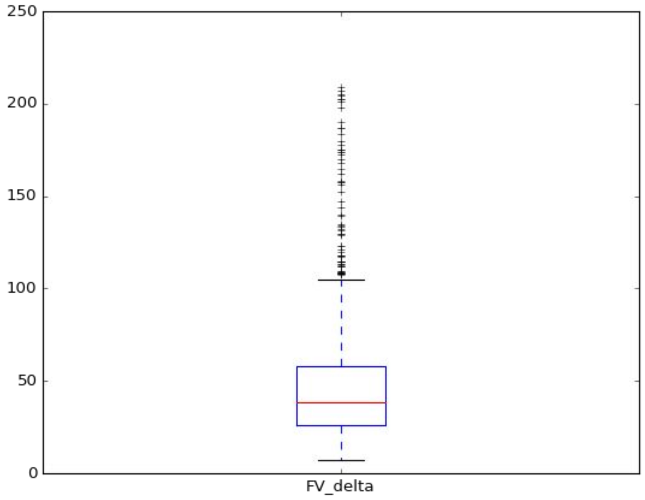
Random Forest Classifier

- RandomForestClassifier yielded best results with max_features value of 4
  - OOB score was ~0.34 and we made correct predictions **~38.3%** of the time
- Let's try Boosting with Tuning
- Ideal parameters for GradientBoostingClassifier:
  - Learning rate: **0.05**
  - Number of trees: **4,000**
  - Max depth: **4**
- This Boosting algorithm allowed us to predict the markup range for concerts in our test set with **41.6% accuracy**
  - Better than nothing, but still not great

Can we interpret anything using the data?

Let’s Try Linear Regression

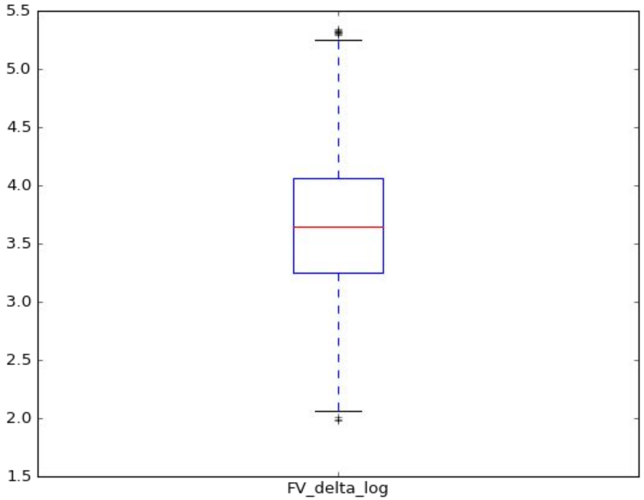
- Linear regression is prone to outliers, so let’s make sure our data isn’t too skewed
- Box plot of raw markup values:



- **Let’s take the logs of our data**

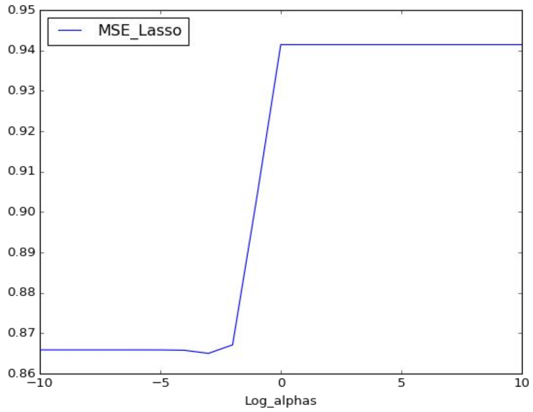
Let's Try Linear Regression

- Looks much better



Lasso Regression to Find Best Variables

- First scaled the data
- Then found ideal alpha for Lasso: **a = -3**
- Then checked the Lasso coefficients
- Then checked correlation matrix
    - hotttnesss was highly correlated with all variables except sold_out (-0.01)
- Decided to use **hotttnesss** and **sold_out** for regression




```
# Find feature coefficients using Lasso regre
lm = linear_model.Lasso(alpha=10**(-3))
lm.fit(X_lasso, y_lasso)
sorted(zip(lm.coef_, X_lasso.columns))
```

```
[(-0.33055393604368116, 'familiarity'),
 (-0.30831655082782616, 'discovery'),
 (-0.061658698486772807, 'num_blogs_log'),
 (-0.0010283055307501879, 'num_reviews_log'),
 (0.022689061491567599, 'num_news_log'),
 (0.057162538340669429, 'face_value_log'),
 (0.06264697551684871, 'days_to_show_log'),
 (0.12417420023515652, 'sold_out'),
 (0.22574460169029595, 'num_years_active'),
 (0.35455412807878184, 'hotttnesss')]
```

Running Linear Regression

- Ran Linear Regression using hotttnesss and sold_out values to predict the logarithm of ticket price markups
- Used the Statsmodel python package to get p-values, R^2, and coefficients:
- R^2 is low (**~0.01**)
- But coefficient P-values are significant!
    - **0.046 and 0.001**
- Model may not capture much variability, but results are significant

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.011
Model:                            OLS   Adj. R-squared:                  0.010
Method:                 Least Squares   F-statistic:                     7.048
Date:                Sun, 03 Apr 2016   Prob (F-statistic):           0.000904
Time:                        17:49:50   Log-Likelihood:                -1157.8
No. Observations:                1260   AIC:                             2322.
Df Residuals:                    1257   BIC:                             2337.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept      3.4353      0.103     33.349      0.000         3.233     3.637
X[0]           0.3267      0.164      1.996      0.046         0.006     0.648
X[1]           0.2531      0.079      3.201      0.001         0.098     0.408
==============================================================================
Omnibus:                        0.582   Durbin-Watson:                   1.660
Prob(Omnibus):                  0.748   Jarque-Bera (JB):                0.478
Skew:                           0.030   Prob(JB):                        0.787
Kurtosis:                       3.074   Cond. No.                         13.3
==============================================================================
```

Interpreting Linear Regression Results

- Hotttnesss coefficient is 0.3267
  - “hotttnesss” = how much people are currently talking about/sharing artist
- Sold_out coefficient is 0.2531
- <u>Interpretation:</u> holding all other variables fixed…
  - For every increase of 0.1 in EchoNest’s hotttness metric, the StubHub ticket price markup increases by **~3.3%***
  - If a show sells out, the StubHub ticket price markup increases by **~25%***

*The prediction values were the <u>logarithms</u> of ticket markups, so we interpret coefficients as % increases rather than absolute increases

Limitations of this Analysis

Data Limitations

- Only looked at **minimum** StubHub ticket price to compute markup
    - Future studies might look at differing price levels - e.g., VIP sections vs. GA
- Only looked at 16 U.S. metros, so conclusions are limited to concerts in those cities
    - Future studies might look at wider concert data
- Dropped data from 1,690 concerts since SongKick didn't have valid ticket info for them
    - Future studies might try to get more official face values directly from ticket vendors
- SongKick did not always give us correct sold_out values
    - Only had ~80 out of 1,200 shows marked as "sold out"

Model Limitations

- Interpretation from Linear Regression is based on the assumption that the data is linear
  - This may not be true - low R^2 value suggests that linear model doesn't capture much variability
- Did not include some variables that may explain additional variability
  - Metro area for concert
  - Day of the week of show

Questions?