Predicting Ticket Resale

by

Christopher B. Pamieta

Dr. Piotr Szczurek, Advisor

Masters project submitted in partial fulfillment of the

requirements for the Master of Science in Data Science degree

in the College of Arts and Sciences of

Lewis University

**Abstract**— Pinpointing the optimal time to buy or sell music festival tickets offers buyers the best deals and sellers the highest sale prices, respectively. Predicting the best time to buy or sell music festival tickets can lead a seller to making a profit, or help a buyer find an affordable deal to attend a concert. By using Ebay API to gather data, this paper looks into different machine learning models to predict if the auction will sell at the current listed price and if a buyer should buy today or wait. Random forest produced the most accurate results with predicting if an auction will sell or not.

**Index Terms**—Machine learning, Random forest, Neural Network, Ticket Prediction

————————————————

—————————— ◆ ——————————

# Introduction

More than ever before websites are advertising how they are able to predict the optimal time to buy plane tickets or forecasting the price of a car. Following this trend, this report describes a similar application which allows users to see predictions on music festival ticket prices. Being able to predict trends is not only valuable for corporations, but also the everyday consumer. Moreover, this web application provides two features for the users, ability to determine if they will be able to sell their tickets at their desired price, and predicting a price trend for the tickets. For the web application, its primary focus will on a proof of concept and only taking into account a single music festival, Coachella. One uniqueness of this music festival is that it is essentially two festivals in one because it spans over two weekends. Essentially, both weekends have the same artists playing and same ticket price so there is no real distinction between the two other then the date.

The advantage of focusing exclusively on music festivals instead of single artists concerts is that festivals contain a large number of performers so the exposure and popularity of these festivals make ticket prices fluctuate significantly more than a single artist concert ticket. In the end, the choice of this data-set came down to the available data at the time.

# Background

The ticket resale market can be seen as a simplistic version of the stock market, which has outside forces influencing the prices with lineup announcements being comparable to stock earning reports. Furthermore, if a stock has a bad quarter, the price will likely go down, which is similar with music festivals, bad artist lineups will create an influx of ticket resales. Unlike the stock market, festival ticket price trends tend to follow the same trend regardless of the festival. Subsequently, the three common factors that are known to influence ticket prices are weather, lineup, and popularity . As a time series forecasting problem, the goal is to predict future outcomes using the outcomes from the past.

In [12], the authors are utilizing a LSTM recurrent neural networks to predict stock prices. The benefit of using a RNN is that it takes into consideration the past data instead of relying solely on present data. This is imperative in making stock predictions since a key factor of what direction a stock may take is the support and resistance level. For instance, if the stock is around a support level, then there is a chance it will go up while at a resistance it may go down. Without knowing the past prices, these levels would be unknown, therefore increases the odds of buying a stock at record high prices. This type of RNN would likely work better with festival ticket prices Vs stock prices since there are less outside forces dictating the trends of festival tickets, like rising gold prices putting downward pressure on the stock market. The one drawback to a RNN is that it is limited in how far back past data can be assessed. The reason for this is that over time, “vanishing gradient, and exploding gradients” occurs as more data is sent into the model. Essentially, what it learned in the past gets lost as time goes by. Lowering the learning rate or using a relu activation can prevent this in a neural network, but for a Recurrent neural network, implementing a Long Short Term Memory is able to solve that issue. LSTM is a more complex version of a RNN in that it has the ability to determine which information should be forgotten or saved , so the data can be viewed for a longer period.

[11], predicting stock trends can be a powerful tool if the model is accurate. The main difference in [12] and [11] is that mire complex feature extraction was performed for random forest model. Using known stock indicators, they calculated “Relative Strength Index”, “Moving Average Convergence Divergence” and other indicators while only using the open/close price and the high/low price of the stock for the RNN. The main functionality of random forest is the decision trees that are generated from the data. Furthermore, a decision tree works by using the best determined features as a root node and then splitting to create two new nodes and continuing until no additional splitting can occur. However, the downside in using a decision tree that it easily overfits the training data and does not accommodate and function well when new data is sent in. Random forest fixes the issue of overfeeding by creating multiple decision trees in which each tree sees a different set of data. Additionally, when more trees that are created, it leads to a higher accuracy, but at a cost of computer power. In the end, the authors were able to achieve a 85-95 percent accuracy in predicting stock trends.

Research suggests that Google search trends are a strong indicator of consumer feelings. [2] discovered a strong correlation between Google search hits and how much money the team made in ticket sales. This research data can be useful in determining the success of music festival ticket sales as well.

# Methodology

The core of a ticket predicting website is the machine learning algorithm that is used to predict if a ticket will sell and if a buyer should buy now or wait. The first part of the web application is obtaining data. Part of the website utilizes several ticket resale APIs that give the ability to query the website for data. Websites that don't provide an API require the use of a scrapper. Subsequently, this is able to extract data from the website by simulating actions a user would take. From this, it’s able to screen capture the text or extract the data from the HTML tags. Google trend was also utilized to extract data.

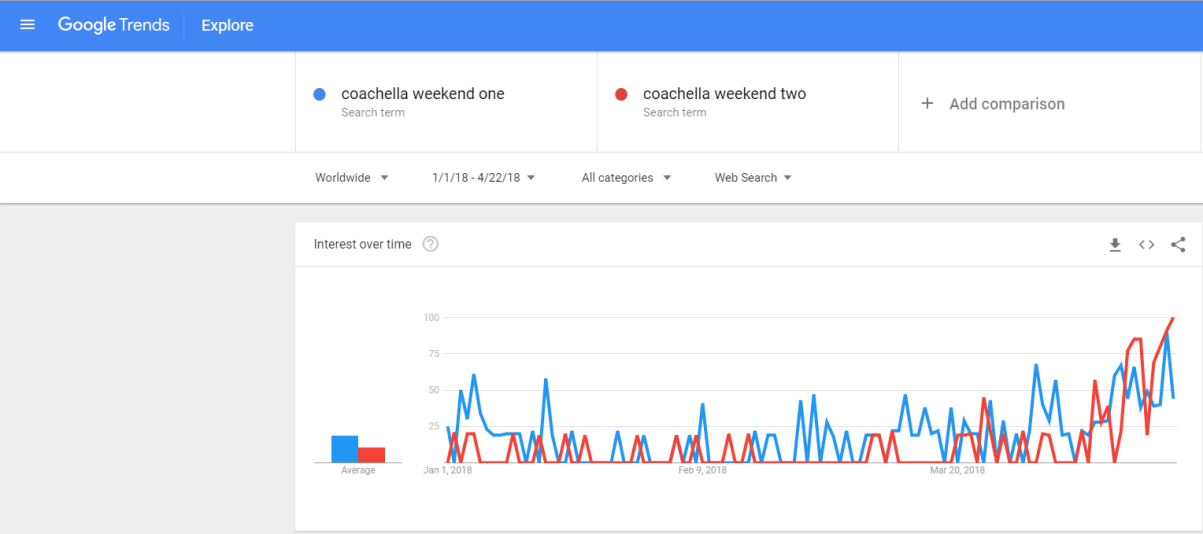


Fig. 1. Trend results from Google trend.

For this, two data sets were extracted from Google trend, first was search ranking of the phrase “Coachella weekend one” while the other one was “Coachella weekend two”. The goal was to identify if a spike in the search phrase correlated to a spike in prices.

The only data source used for proof of concept was eBay because it provided a simple to use API which contained three months of past data. Additionally, using PHP and sample code eBay provided, a list of 3200 ticket for Coachella were obtained with the title, description, final price, shipping cost, if the auction sold or not, auction start time, auction end time, link, and subtitle. Furthermore, once all of the data cleaning took place, the number of tickets dropped to 2634.

The model building phase included 3 models and the first model implemented was a neural network. From the article [10], they wrote about the importance of identifying outside forces when predicting sales of the product. When the models were built weather data was taken into account. With the inclusion of external data, they were able to achieve 23.9% improvement in forecasting with the neural network. From that, the results of Google trend were added in the data set. The next model was a random forest. In [11] it was discussed that transforming the problem into a classification problem would minimize forecasting errors which influenced the risk an investor had in investing in stocks. In addition, a new feature called “Price Range” was created to treat ticket price prediction as a classification problem and this was completed by having each range span twenty five dollars. In the end, both the neural network and random forest took in the features listed in Table1.

Table 1. Sample data sent into Neural network

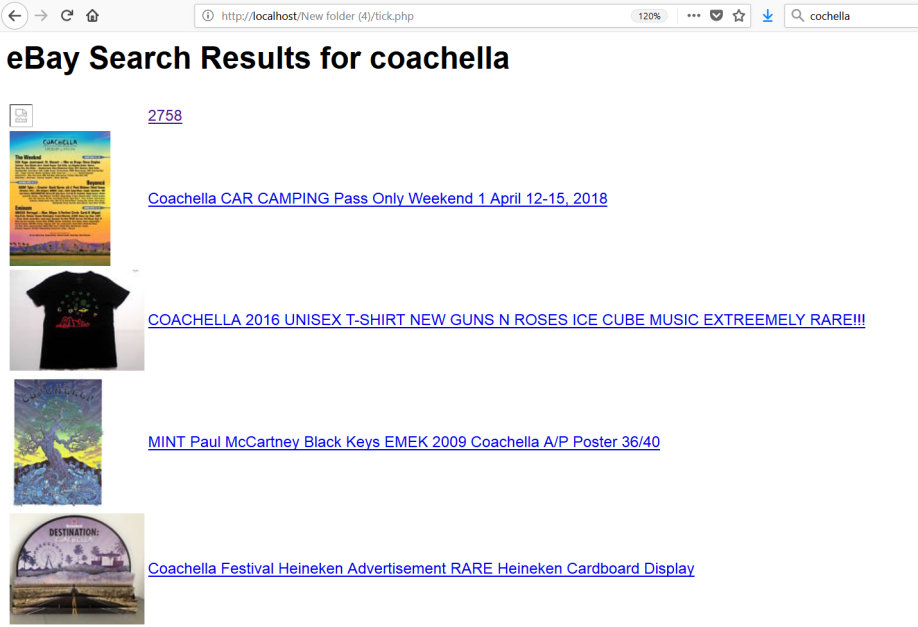
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # of ticket | CAR CAMPING PASS | VIP | VIP Parking | weekend | days till | PriceRange | ticketPrice | trend | avg unsold price | avg sold price | Shuttle Passes | sold |
| 1 | 0 | 0 | 0 | 2 | 2 | 44 | 443.5 | 41 | 532 | 488 | 1 | 1 |

The final model that was built was a LSTM Recurrent neural network. For this model, only the average price was fed into the model for weekend one.

Implementation

The implementation phase of the project had four parts which included backend, data cleaning, building, and frontend. First, was the backend phase, which is the PHP code that is used to access the eBay API to collect the data.

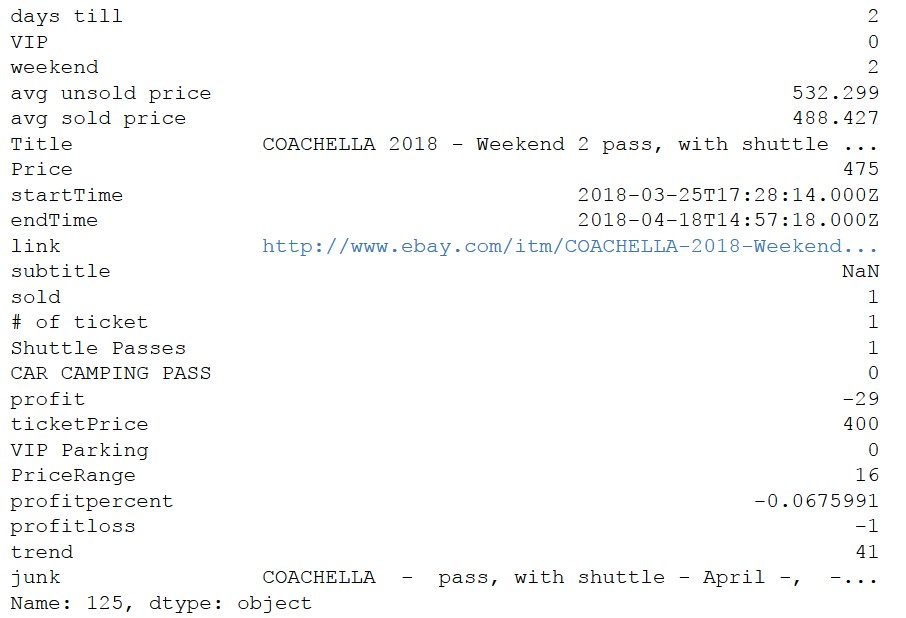
Image 1. Results from php call to Ebay.



The key settings for this API was to return all results that were above $100 and were in the event ticket category. However, the initial data collection phase took a more manual approach because eBay API only allows a max of 100 items to be returned in one call. In order to get the other items, there would need to be a manual update to the “pageNumber” criteria in the api. Secondly, the next phase was the data cleaning phase which included a major research component. The goal was to obtain as much information regarding the music festival and identify all the options available for buying tickets. Based on research there appeared to be two ticket versions, general and VIP, where VIP provided extra perks such as gourmet food and drink options. Subsequently, there were other perks that could be purchased for an additional fee, including a shuttle pass, Car Camping Pass, or VIP Parking. These perks did not directly relate to the festival itself, but added to the convenience factor which many value. Using the title of the auction and what was learned about the event , new features were created out of each option. Another important feature was the end date of the auction. This feature was converted from a date to days till the event. As a result, there were more desired features created by removing the date of the event and focusing on how many days are left till the music festival begins. This allowed for a more flexible model.

Next, it was crucial to calculate if the tickets were sold at a profit or not, which ultimately was useful information for the users of the web application in determining if its worth selling or just go. The way profit was calculated was that the price of the ticket was dynamic while all the other add-ons like camping pass had a static price. With that, all the add-ons where subtracted from the total price. Finally, the last element included interpreting the data and calculating the average price of sold and unsold VIP and standard tickets on a daily base. This in turn gave a clearer vision on how the prices rose and fell in the four month span.

Table 2. Sample data after data cleaning and feature engineering.



The final step in the data cleaning phase was removing the junk data. This was easy to do since all valid data had either a weekend value of 1 or 2. Everything else was dropped from the list.

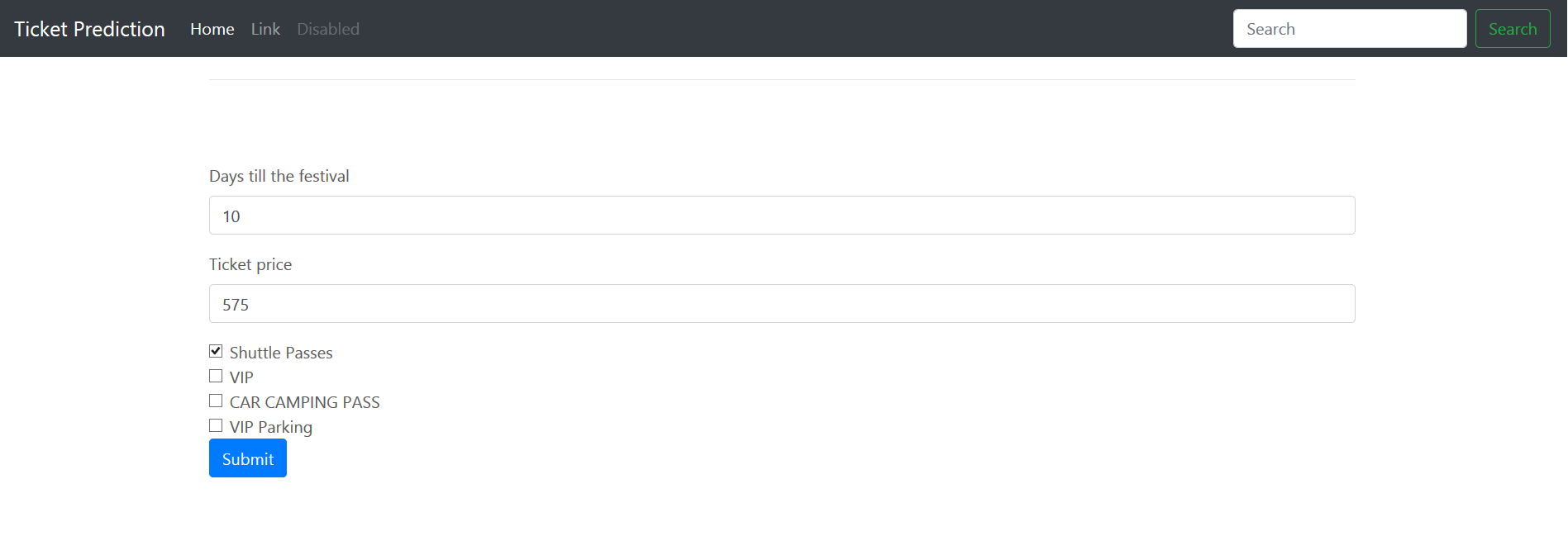
he third phase was comprised of building the models, both Random forest, a Neural network and a LSTM recurrent neural networks using sklearn and keras packages. The neural network incorporated two layers a relu activation with 100 neurons and the output layer having a sigmoid activation. Both layers had a L2 regulariser of 0.01. From all the testing, this provided the best results without overfitting the data. Results from this and other models can be seen in Table 3

For the LSTM RNN, three total layers included in the model. The first layer was the LSTM with 1000 neurons and an activation of tanh. The output layer had a linear activation. In between each layer, drop out at a value of 0.2 was added to prevent overfitting.

With the random forest a few settings were adjusted to allow for better performance, which included the grid search. This enabled a pass in a list of estimators and minimal sample leaf to search for the best combination. The list for the number of trees included 10,100,500,1000 and the list for min sample leaf: 1,2,3,5,10. In the end it, it chose a value of 10 for n\_estimators and a value of one for min\_samples\_leaf, meaning that only ten decision trees were generated while only needing one sample data to create a split in the tree.

The final phase was the frontend phase which involved the HTML website that is used to display the results for the models.

Image 2. Frontend page for ticket prediction

****

This was accomplished with a simple HTML form that allowed users to input data on the ticket and submit the request. After, a PHP script then formatted it and created all call out to the python script. Then the python script loaded up the saved, trained model to run the date through and spit out the prediction. Finally, the python script then sent the response to the PHP script to display on the website.

results

First, it was important to create a time series graph displaying average daily price so that any visible price trends could be identified. There appeared to be a downward trend for week two prices while week one price stayed relatively flat until the final days of sale.

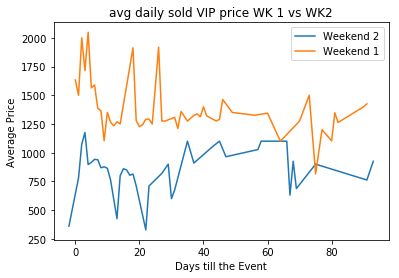
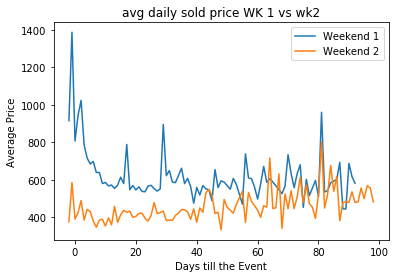


Fig.2. Average daily price for weekend 1 and weekend 1. Fig. 3. Average daily VIP price for weekend 1 and weekend 1

The first model that was built was a neural network. Using the Neural network to predict if a ticket would sell or not returned a loss of 4.57 with an accuracy of 71%. These were the most accurate results that were obtained using the neural network and trying a variety of layers, activation, and regularizations.

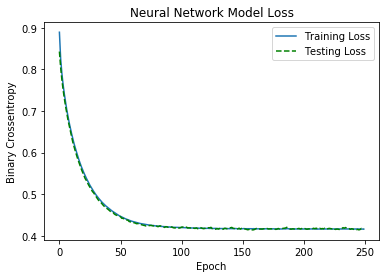


Fig. 4. Neural network Model loss

Table 3. Neural network results in predicting if the ticket sold or not.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | LOSS | ACC | Val Loss | Val Acc |
| relu activation with 100 neurons,regularizer: l2 at 0.01, |output layer: sigmoid activation,regularizer: l2 0.01 | 0.4161 | 0.8614 | 0.417 | 0.8254 |
| relu activation with10 neurons,regularizer l2 at 0.01, | Output: sigmoid activation,regularizer l2 at 0.01 | 0.4176 | 0.8548 | 0.4156 | 0.8311 |
| relu activation with 50 neurons,regularizer:l2 at 0.01| dropout 0.2 | Output :sigmoid activation regularizer: l2 at 0.01 | 0.4204 | 0.8524 | 0.4171 | 0.8368 |
| relu activation with 50 neurons, regularizer: l2 at 0.001| output: sigmoid activation regularizer: l2 at 0.001 | 0.3019 | 0.8832 | 0.3285 | 0.8691 |
| relu activation with 50 neurons, regularizer: l2 at 0.1| Output sigmoid activation regularizer: l2 at 0.1 | 0.5835 | 0.719 | 0.5855 | 0.7078 |
| relu activation with 50 neurons | dropout at 0.2 | Output: sigmoid activation | 0.2715 | 0.8818 | 0.3023 | 0.8672 |
| relu activation with 50 neurons, regularizer: l1 at 0.01| Output: sigmoid activation regularizer: l1 with 0.01 | 0.4987 | 0.8168 | 0.4798 | 0.7951 |
| relu activation with 50 neurons, regularizer: l2 at 0.01| output sigmoid activation , regularizer:l2 at 0.01 | 0.417 | 0.8543 | 0.4136 | 0.8349 |
| relu activation with 50 neurons, regularizer: l2 at0.01| output: relu activation regularizer: l2 at 0.01 | 0.365 | 0.7456 | 0.4943 | 0.723 |
| sigmoid activation with 50 neurons, regularizer: l2 at 0.01|Output: sigmoid activation regularizer: l2 with 0.01 | 0.5185 | 0.7537 | 0.5112 | 0.7476 |
| relu activation with 50 neurons regularizer: l2 at 0.01 | 1.2389 | 0.9948 | 0.9948 | 0.6679 |
| sigmoid activation with 50 neurons regularizer l2 at 0.01 | 0.4305 | 0.8026 | 0.4204 | 0.7951 |

The Random forest model was the following model which provided significantly better results with an accuracy of 87%. Using feature ranking and the following features played the biggest role in determining if the ticket sold or not, including Avg unsold price, Avg sold price, ticket price, and days till the event.

Decision Tree Results

Table 4. Confusion Matrix

|  |  |
| --- | --- |
| 104 | 50 |
| 15 | 358 |

Table 5. Classification Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.87 | 0.68 | 0.76 | 154 |
| 1 | 0.88 | 0.96 | 0.95 | 373 |
| avg/total | 0.88 | 0.88 | 0.87 | 527 |

The last model built was the LSTM. For this model, a different question was being asked. The goal was to be able to predict the ticket price for the next day. At first, it seemed that this model performed very well, with the prediction matching close to the actual results, but later discovering that the model was overfitting the data, with a higher loss of the testing data. Few changes to the model were made, but the model still was overfitting the data. In the end a loss of 0.0458 on the training and 0.0571 on the test data was achieved.

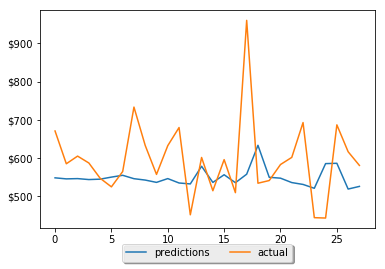
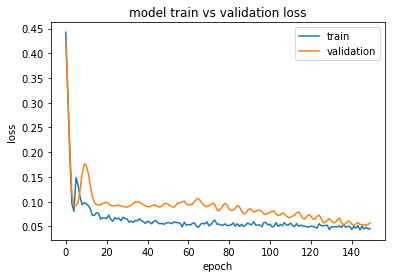


Fig. 5. LSTM RNN Model loss Fig. 6. Price Prediction

When comparing the two models and inputting test data it appeared that there was a discrepancy in the output data. In the first case, the neural network returned a predicted value of ‘not sold’, while the random forest predicted a ‘sold’ value. Only when the price was lowered did neural net return a ‘sold’ value. On the other hand, for the ‘not sold’ data, both models were able to return a predicted value of ‘not sold’, but they both did not perform as well with ‘unsold data’. A potential explanation for could be the lack of unsold data set when compared to sold. With no way of getting more data for now, creating a fake unsold listing using the average unsold price might be able to solve this issue. In the end, this was not fully implemented. For the LSTM model, it looked like it did a good job in predicting future ticket prices, but in the end that was not the case. Possibly getting a larger time frame and adding new features might be able to solve those issues

Conclusion

With the popularity of music festivals increasing, determining festival ticket resale value and trends will help buyers secure the best deals and/or maximize sellers’ profits. When comparing both neural network and random forest, the random forest provided the best results for the manual data that was entered. The major benefit of the models is that it allows the users the ability to identify what the maximum and minimum ticket prices are. This is more efficient because the seller will know if a price is just way too high to sell tickets for. Future advancements to the website include adding a buy now or wait suggestion box, with a predicted price trend for the buyer. One feature that was not taken into consideration was the percent profit. This feature would be valuable for training the model with data from multiple different festivals. In this case, the ticket price feature would be dropped since different events would have a different base ticket price.

References

[1] J. Suher, “Forecasting Event Ticket Sales.” *Penn Libraries*, 2018. [Online]. Available: [repository.upenn.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1048&context=wharton\_research\_scholars](http://repository.upenn.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1048&context=wharton_research_scholars)

[2] N. Silver, “How Google Searches Can Predict Hockey Ticket Sales.” *FiveThirtyEight*, FiveThirtyEight, 2015, [Online]. Available: fivethirtyeight.com/features/how-google-searches-can-predict-hockey-ticket-sales/.

[3] S. Axon, “SeatGeek Predicts the Best Time to Buy Concert and Sports Tickets.” *Mashable*, Mashable, 2010, [Online]. Available: mashable.com/2010/03/30/seatgeek-predicts-the-best-time-to-buy-concert-and-sports-tickets/#dTRPA8D89EqQ.

[4] S. Heinz, “A Simple Deep Learning Model for Stock Price Prediction Using TensorFlow.” Medium, ML Review. 2017, [Online]. Available:

medium.com/mlreview/a-simple-deep-learning-model-for-stock-price-prediction-using-tensorflow-30505541d877.

[5] J . Grossman, “Predicting EBay Auction Sales with Machine Learning.” Jaygrossman, 2013, [Online]. Available:

jaygrossman.com/post/2013/06/10/Predicting-eBay-Auction-Sales-with-Machine-Learning.aspx.

[6] I. Raykhel, and D. Ventura. “Real-Time Automatic Price Prediction for EBay Online Trading.” Real-Time Automatic Price Prediction for EBay Online Trading, 2009, pp. 1–6., [Online]. Available:

[www.google.com/url?q=https://aaai.org/ocs/index.php/IAAI/IAAI09/paper/download/264/1029&sa=U&ved=0ahUKEwikysazt4\_aAhUL1lMKHeVbC0cQFggJMAI&client=internal-uds-cse&cx=016314354884912110518:gwmynp16xuu&usg=AOvVaw2HBCT\_jbnXfyjaJyq07lGX](https://aaai.org/ocs/index.php/IAAI/IAAI09/paper/download/264/1029).

[7] R. Ren,. “Prediction of Airline Ticket Price.” Prediction of Airline Ticket Price, 2015, [Online]. Available:

cs229.stanford.edu/proj2015/211\_report.pdf.

[8] R. Paranjpe, and D. Nicholson. “A Novel Method for Predicting the End-Price of EBay Auctions.” Semanticscholar, 2013, [Online]. Available: pdfs.semanticscholar.org/7193/7bde07556da6b145ac3528eaa85b725bc60d.pdf.

[9] M.Hoekstra, “Hotel Booking Prediction by Means of Fuzzy Logic Prediction.” Beta.vu, 2010, [Online]. Available:

beta.vu.nl/nl/Images/werkstuk-hoekstra\_tcm235-182528.pdf.

[10] M. Riemer, “Correcting Forecasts with Multifactor Neural Attention.” Proceedings, [Online]. Available:

proceedings.mlr.press/v48/riemer16.pdf.

[11] L. Khaidem,. “Predicting the Direction of Stock Market Prices Using Random Forest.” *Applied Mathematical Finance*, 2016, [Online]. Available: arxiv.org/pdf/1605.00003.pdf.

[12] M. Roondiwala,. “Predicting Stock Prices Using LSTM.” *International Journal of Science and Research*, vol. 6, no. 4, 2017, [Online]. Available :www.ijsr.net/archive/v6i4/ART20172755.pdf.

[13] owygs156. “Machine Learning Stock Prediction with LSTM and Keras.” *Machine Learning Applications*. 2018, [Online]. Available :intelligentonlinetools.com/blog/2018/01/19/machine-learning-stock-prediction-lstm-keras/.

[14] P.Srivastava,. “Essentials of Deep Learning : Introduction to Long Short Term Memory.” *Analytics Vidhya*, 2017, [Online]. Available: www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/.

[15] Getting Started with the Finding API: Finding Items by Keywords, 20019,[Sourcecode].

http://developer.ebay.com/DevZone/finding/HowTo/GettingStarted\_PHP\_NV\_XML/GettingStarted\_PHP\_NV\_XML.html

Appendix

All source code is located the following github link.

https://github.com/cpamieta/547-202