Final Report

Ethan Lee | Michael Considine

Our Project

Air travel is the quickest and most efficient way to travel long distances and millions of people a year buy plane tickets to travel for leisure, work, or emergencies. As avid travellers ourselves, we often experience a feeling of uncertainty buying tickets, not knowing if that purchase was a good deal or not. Our model is an attempt to resolve this issue. By using past ticket prices and information, we can predict the price of a ticket, and compare that to the actual price offered to determine if a ticket is a good deal.

Data Collection

In order to create our dataset, we wrote Python scripts to crawl the Southwest Airlines site and to find the prices of flights between our 100 airports on July 1st, 2018. The script used various default settings: it searched for flights that would arrive and depart at any time during the day, finding prices for a single adult passenger with no senior discount and no extra baggage. The script returned data in JSON format for each flight: the cost, the arrival and departure times, the number of RapidRewards points accrued, whether the flight was sold out, and the three-letter codes of the arrival and departure airports. However, we quickly realized that the number of accrual points was related directly to the cost of the ticket, so we removed this attribute. We then parsed this JSON data and stored it in a Mongo database, which we finally exported as a CSV file. These scripts took several hours to run, but returned to us information about roughly 47,000 flights. We combined the flight data with several manually-collected attributes: the populations of each relevant city, the sizes of each airport (in acres), the average number of yearly passengers per airport, and the average number of yearly sunny days in each relevant city. This gave us a dataset with 14 attributes and roughly 47,000 instances, which we would later feed through Weka.

Feature Selection

Our dataset consisted of about 47,000 different flights scheduled for the start of July. We included 13 attributes (excluding ticket cost) to predict the cost of a ticket:

- Whether or not the flight was sold out
- The departure and arrival times of the flight
- The average number of yearly passengers through both departure and arrival cities
- The size of both departure and arrival cities
- Airport codes of the both departure and arrival cities
- Airport populations of both departure and arrival cities
- Average numbers of yearly sunny days of both the departure and arrival cities

Models Tested

Given that we wanted our model to predict the cost of flight tickets (a numerical value), we decided to start by testing linear regression models. We achieved our highest correlation score of 0.7577 through this model, but we decided to test other methods as well (best model for each learner type shown in the table below), although none of the other models were as effective as our linear regression model.

Model Tested	**Correlation score**
Additive Regression (10-fold)	0.3440
Locally Weighted Learning (66 split)	0.3008
Linear Regression (8-fold)	0.7577
Nearest Neighbor (100-fold)	0.6824
ZeroR	0.0

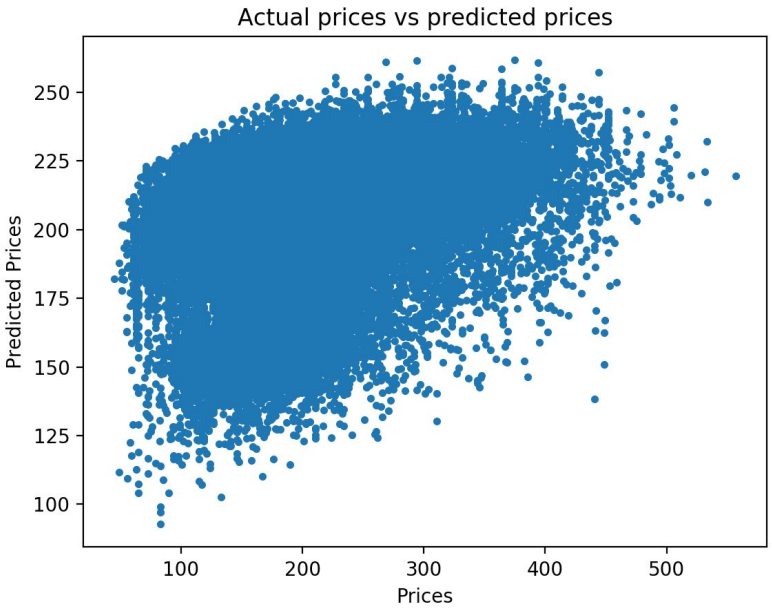
Our best results from our linear regression model gave us a mean absolute error of 35.15, meaning that on average, our prediction was $35 above or below the actual price. Because prices generally fell between the $150-$350 price range, with an average cost of $206.76, we think that in the scope of this project our model achieved the desired results even with a 17% average error. Our goal was to find if a ticket price was fair or not, and a difference of $35 is small enough for our model to be useful in the real world.

We initially expected (and our experimentation confirmed) that linear regression would be the best learner for this dataset. This is because regression is generally suited to numerical predictions, and since it assigns weights to each attribute, it was able to focus more on the most important attributes while assigning less significant weights to the less relevant attributes. Even though our dataset had categorical variables like the departure and arrival airports, Weka automatically converts these to continuous values by weighting the combinations of the presence or absence of these values.

Naive Linear Regression Implementation

After our initial theory that linear regression would be the best model for this type of data, we decided to attempt to implement our own linear regression model (although with very minimal success). Our biggest problem was figuring out how to represent our categorical variables (destination and departure airports). The method that we decided to use was to represent our categorical feature as a combination of the continuous features that correspond to it. The way we did this was by representing each airport code as the sum of the normalized features that it affect:

population, airport size, annual passengers, and days sunny. This combination was fairly arbitrary, and is probably a large cause of this bad distribution.



As can be seen from the image, this model is extremely inaccurate. After running the model on separate training and test data, we achieved a mean squared error of 4380. We believe that this high error is due to our attempts to represent our categorical features as arbitrary combinations of the features that it corresponds to.

Conclusions & Future Work

In future iterations of this project, we would work to more closely determine which sets of attributes were most effective. We tried several different combinations of attributes during our testing, but none granted better results than simply using all 14. We believe that with more experimentation we could find a more effective grouping of features and thus would be able to create a more efficient and accurate model. However, within the scope of this project, we believe that we achieved positive and useful results.

Contributions

Mike was responsible for the web crawler implementation/data collection and the website, while Ethan worked on testing various models and combinations of attributes in addition to implementing his own machine learning model. Both members contributed to the final report.