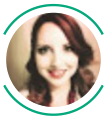
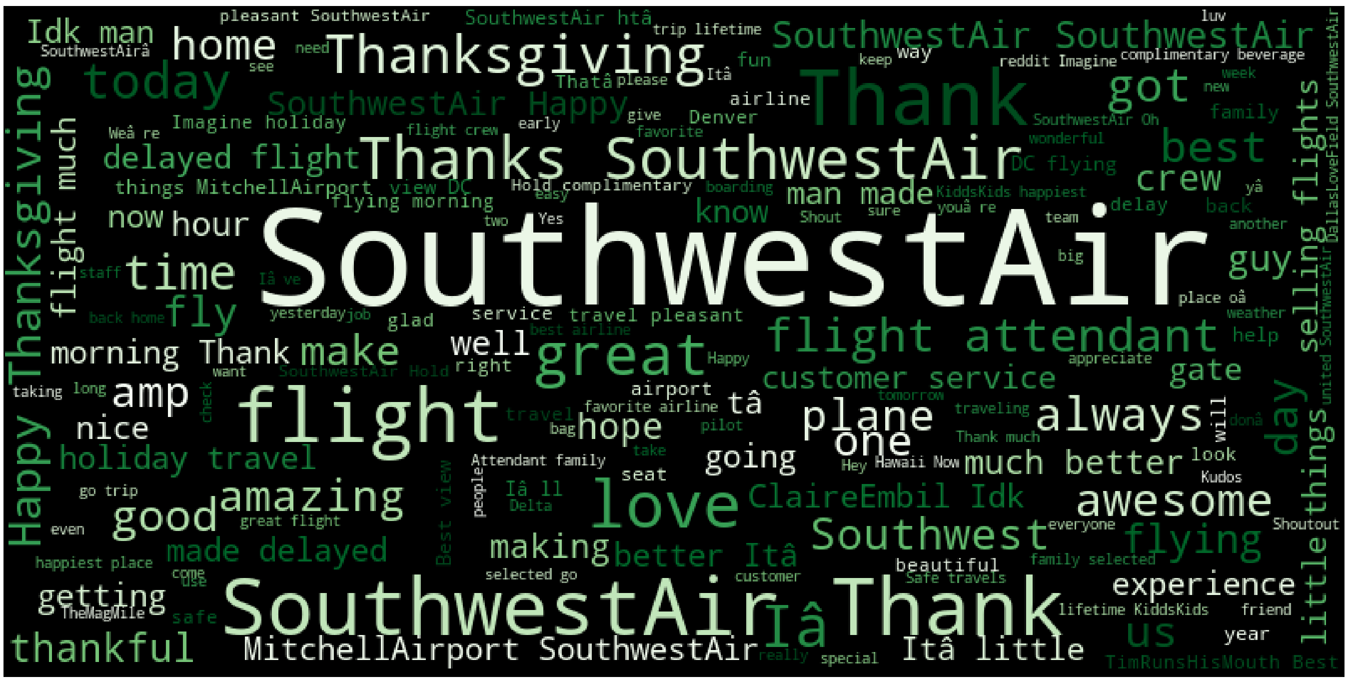
Airline revenue prediction



Lydia Guarino Follow
Dec 14 · 9 min read

A tale of heartbreak, discovery, and how we almost beat Wall Street's revenue consensus.



Hey, look, that NLP tweet analysis WAS good for something.

Background

This project was authored by **Alex Hoganson**, **Dale Pedzinski** and **Lydia Guarino** as part of an introductory ML class at the University of Texas under the guidance of professor **Alex Dimakis**. Our datasets and analysis can be found on **data.world** and **github**. We link to our relevant code and datasets throughout this post.

Introduction

The **US Bureau of Transportation Statistics** makes a wealth of aviation data available for public download. Our team was inspired by the richness of these flight statistics and became curious about the application of using publicly available data to predict something valuable about individual airline carriers. We worked from the following question:

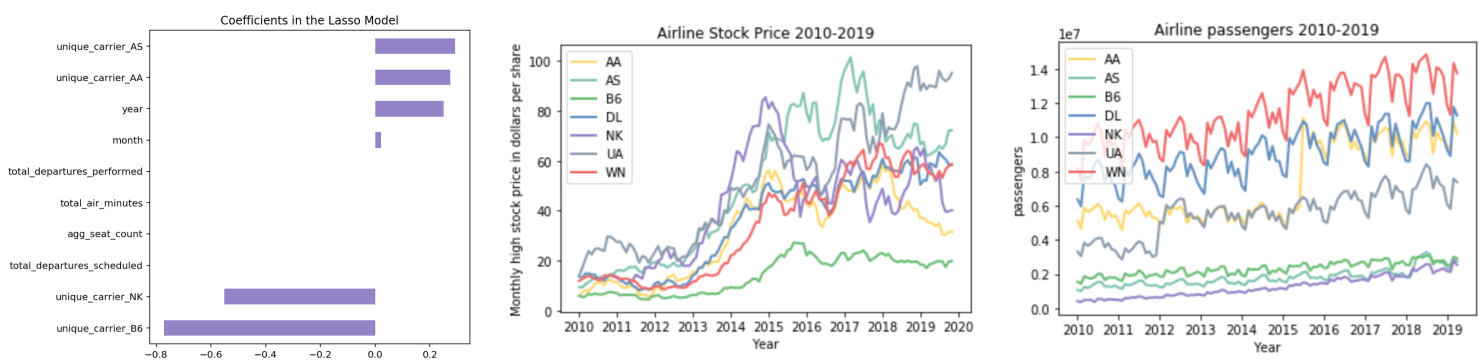
> *Can we create a Machine Learning model to predict airline stock price or revenue using public aviation data?*

The Project

To answer this question, we worked through 4 experiments. 2 were flops and 2 were wins. Our best performing experiment produced an average percent error of just 2.5% when estimating quarterly revenues, rivaling Wall Street's revenue estimate consensus. Further refinements that applied the WS projections as a feature dropped the average percent error even further, to 0.95%. Below, we walk through the specifics of each of these experiments.

Experiment 1 — a flop: Linear Regression stock price prediction

We began with flight statistics for 7 major airlines spanning 2010–2019. Our hypothesis was that flight statistics such as passenger volume could be used as a proxy for demand. We added fuel expenditure stats as a proxy for monthly operating expenses and finally compiled monthly stock data for each airline to use in a supervised linear regression model.



Feature coefficient analysis showed that there was little correlation between our features and stock price.

While this exercise was valuable in exploring tools, practicing data manipulation, and leveling up matplotlib skills, it was quickly evident that there was insufficient correlation between our features and stock price. Flight statistics, it seems, were not a good fit for stock price prediction, which has deep roots in human psychology.

Experiment 2— a flop: Sentiment analysis

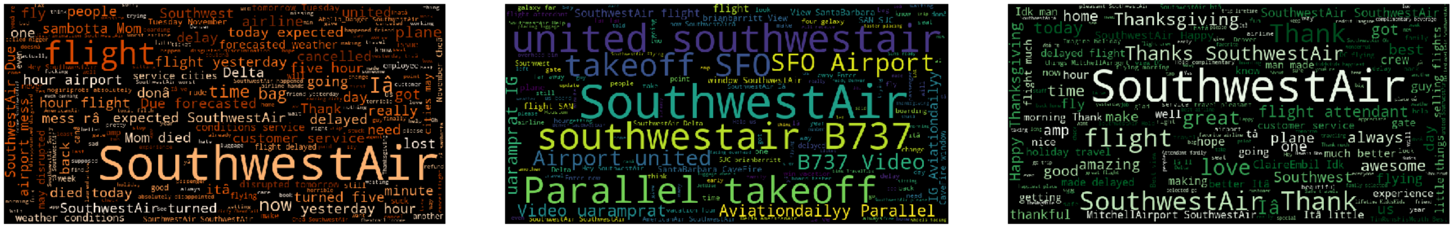
Working from the idea that stock price might have better correlations with market sentiment, we decided to explore whether we could predict stock inflection from an NLP sentiment analysis of tweets referencing a particular airline. Some initial research surfaced support that sentiment analysis performed on tweet corpora had been used in several academic studies to predict stock behavior. To focus our research, we chose to dive deep on just one airline — Southwest Airlines. We made this selection based on their middle-of-the-pack stock price and also because they hold the largest US fleet of Boeing 737 Max Jets. We were eager to see if negative sentiment about the safety of these jets had any impact on stock price.

This experiment seemed promising. After a search uncovered a fully labeled sentiment dataset containing tweets about airlines from February 2015 to use as a validation dataset, we signed up for a twitter developer account (waited a day for approval), and used the Tweepy api to obtain recent tweets about Southwest Airlines.

	Total Requests PER MONTH ⓘ	**Month-to-month** PRICE PER MONTH ⓘ
Paid		
	Up to 100	$99.00
	Up to 250	$224.00
	Up to 500	$399.00
	Up to 1000	$774.00
	Up to 2,500	$1,899.00

The problem: Twitter is notoriously stingy with what data they make available without a hefty price tag (we're talking hundreds of dollars per month). Although the tooling was easy to work with, we were ultimately only able to obtain tweets from the previous 10 days, totaling around 7000 tweets. This was plenty for performing sentiment analysis, however, 10 days was far too small a time series to build a stock price prediction model against.

We decided to use the data we obtained to test out the AWS Comprehend API and to generate some sweet sentiment word cloud graphics for this blog post.



Negative, neutral, and positive sentiment extracted from Southwest Airline tweets. The "parallel takeoff" terms came from a heavily recirculated video of a United E-175 and Southwest B737 parallel takeoff from San Francisco International Airport.

Experiment 3 — a win: Linear Column Regression revenue prediction

It seemed that stock prediction was perhaps outside our grasp given our available data, delivery timeline, and relative naiveté about the complexities of the market. We crossed stock prediction off our list and moved on to exploring *revenue prediction.*

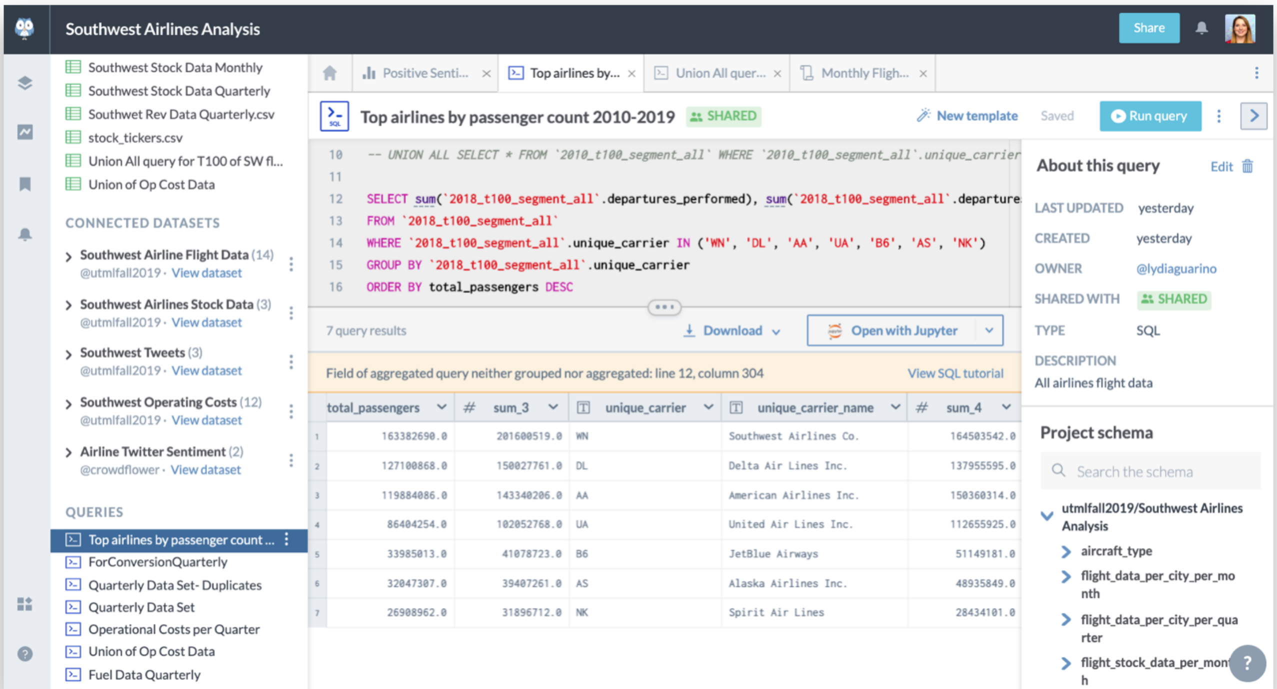
Here's the problem with revenue: It's only reported quarterly. While monthly data was still giving us a reasonably sized dataset, rolling that data up to the quarter level was going to cut the dataset down to, well, *a quarter* of its' initial size. To overcome this, we decided to use a much larger dataset that included a breakdown by carrier segment. A segment essentially represents a single type of flight: a departure location, a destination location, and a specific aircraft type.

T-100 Segment (All Carriers)	This table combines domestic and international T-100 segment data reported by U.S. and foreign air carriers, and contains non-stop segment data by aircraft type and service class for transported passengers, freight and mail, available capacity, scheduled departures, departures performed, aircraft hours, and load factor. For a uniform end date for the combined databases, the last 3 months U.S. carrier domestic data released in T-100 Domestic Segment (U.S. Carriers Only) are not included. Flights with both origin and destination in a foreign country are not included.
	Table Profile Carrier Release Status Download

T-100 Segment (All Carriers) dataset

Using segment data produced a massive dataset. We decided to continue with our previous plan to dive deep on Southwest Airlines. Even with a filtered dataset containing only Southwest segments, we were able to obtain 300,117 rows of data with 21 columns of features.

A major asset to this phase of the project was using data.world as our centralized project space. We had 3 people compiling data, executing queries, hopping in and out of Jupyter notebooks and all working remotely. Data.world proved an excellent centralized place to keep track of our data and analysis and provided some native data inspection tools for this project. Our project and datasets have been made public on the platform and can be accessed with a free account.



Disclaimer: Lydia Guarino works for data.world. All use of the platform for this project was conducted solely for academic purposes.

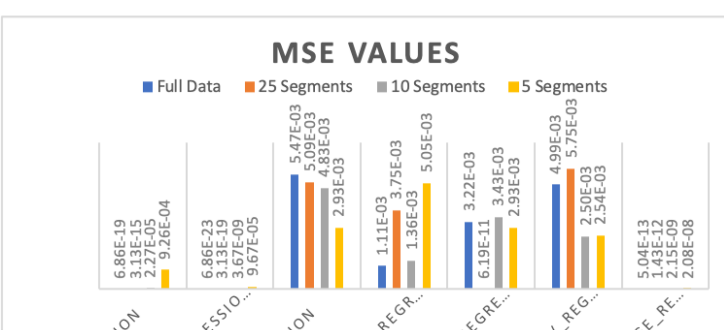
Once we had our carrier segment data and our project space set up, we began to search for additional data to enrich a wider financial picture. While it was relatively straight-forward to obtain fuel costs, operational costs, and stock data — the challenge became combining the data for maximum impact on predicting quarterly revenue. Each of these datasets were provided at a different time scale. Some were daily, some monthly, and some quarterly.

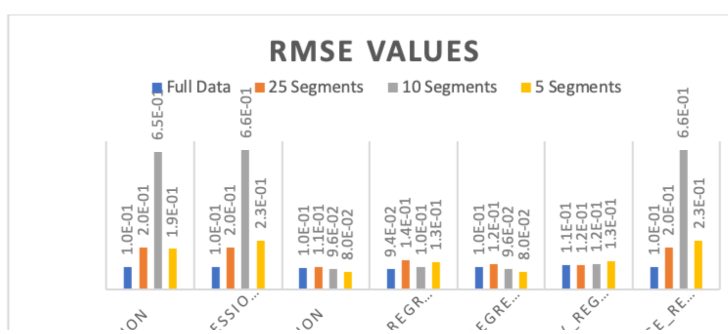
After some significant feature engineering, we decided to organize the data as a column regression structure, rather than the traditional row regression format. This meant we had a wide dataset with *thousands* of columns and only 40 rows representing each quarter of a 10yr period. This essentially gave us a row representation of the entire network of flights over a given quarter with revenue as a label. From this structure, we were able to ascertain which *segments* had the greatest correlation with company revenue. Perhaps unsurprisingly, this process identified several of the airline's "mega stations" in Dallas Love Field (DAL), Houston Hobby(HOU), and Las Vegas McCarran(LAS) as having the highest correlation with revenue prediction. Once we identified these segments, we decided to key in on this correlation and filter the number of features from the full network down to the top 5, 10 and 25 passenger flown segments per quarter. Which reduced the included features from 9126 to 60.

	year	quarter	fuel_price	stock_price	operational_expense	HOU-DAL-Boeing 737-300_flights_flown	HOU-DAL-Boeing 737-300_passengers_carried	DAL-HOU-Boeing 737-300_flights_flown	DAL-HOU-Boeing 737-300_passengers_carried
0	2010	1	78.81	2.591	-2575561	1149	107530	1077	97114
1	2010	2	77.82	2.628	-2805548	1230	119403	1032	101422
2	2010	3	76.06	2.555	-2836702	1310	126379	990	95550
3	2010	4	85.16	2.668	-2898066	1144	118670	1004	101133
4	2011	1	94.07	2.607	-2988795	907	90669	903	89313
5	2011	2	102.02	2.549	-3400693	0	0	1015	104610
6	2011	3	89.49	2.335	-3345870	0	0	1004	99796
7	2011	4	94.02	2.245	-3267209	1102	113212	985	102076
8	2012	1	102.98	2.309	-3285329	1145	114125	1041	105028
9	2012	2	93.29	2.263	-4156226	1205	130146	1082	118338
10	2012	3	92.17	2.322	-4257722	1012	105182	1112	114087

Sample of Experiment 3 data

We ran our data through eight popular linear regression models: *Ridge, RidgeCV, Lasso, LassoCV, LassoLars, ElasticNet, ElasticNetCV, and BayesianRidge* - comparing the results. We tuned alphas for each model and evaluated the predictions for RMSE, MSE, and average percent error. The outcome was a reasonably solid prediction model, with ElasticNet and Lasso performing the best. Lasso and ElasticNet down selected features with the most impactful coefficients to use in the estimates. This eliminated some of the noise and improved the RMSE and MSE values. ElasticNet achieved an RMSE of **0.0959** and an average percent error of **6.4%**. Lasso achieved an RMSE of **0.0955** and an average percent error of **5.7%.**





MSE and RMSE results for each model type and top segment splits by passenger count for Experiment 3

Experiment 4 — the winner: Linear Regression revenue prediction using the neighborhood comparable homes hypothesis

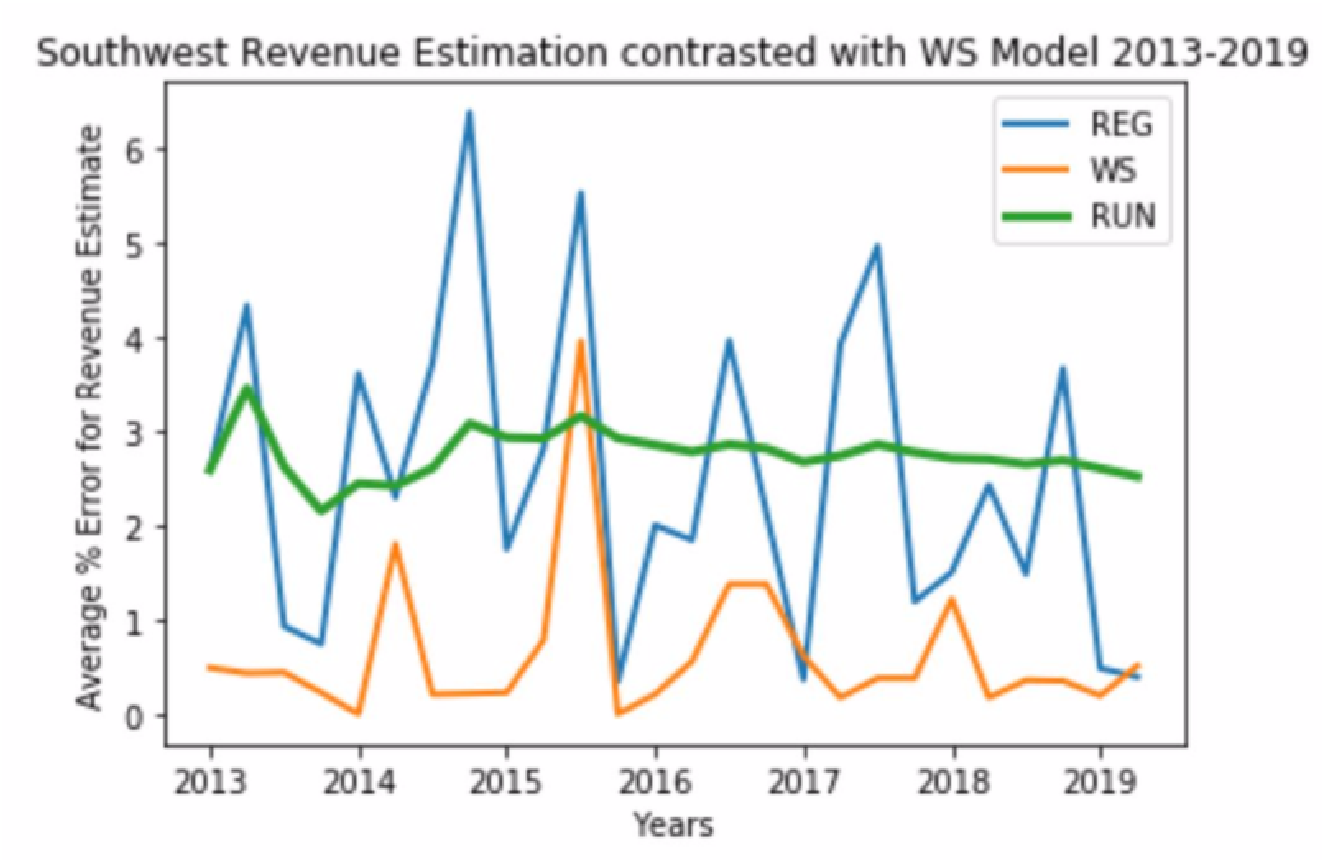
The results from Experiment 3 were promising. They certainly showed that our dataset had enough correlation with revenue to make another iteration worth the exploration. For this variation, we went back to our flight segment dataset and re-evaluated it through the lens of an individual segment's bearing on revenue. We formed a hypothesis that the model could be thought of as a variation of average neighborhood home price estimation algorithms.

We reformatted the dataset into 37,896 rows of quarterly segment data and performed the following algorithm:

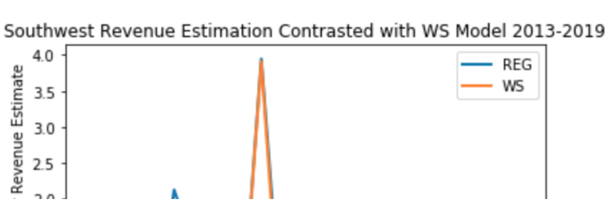
1. For each year/quarter pair
    a. Create test and training Sets
        i. Test set = all flight segments in designated year/quarter Pair
        ii. Training Set = All Flight Segments not in Test Set
    b. Preprocess Data
        i. Apply log operation on skewed numerical features(flights scheduled)
        ii. One-hot encode Enumerated Data(city pair, quarter)
    c. Fit the training set with the best possible RMSE
        i. Use Ridge and Lasso models, and attempt to optimize alpha
        ii. Cache the Lasso models most relevant features for later
    d. Predict the test set, calculate average % error and RMSE
        i. This will be the average prediction for all the flight segment data for that quarter

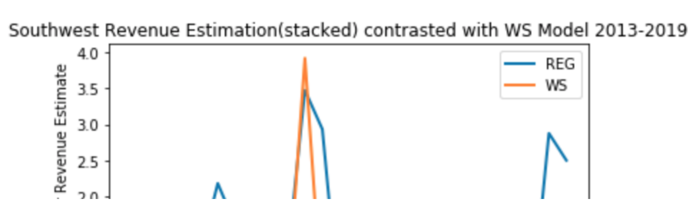
The best performing regression type was Ridge with Alpha 2 after hyperparameter tuning; this model produced an average test error of **2.5%** over the Q1 2013 to Q2 2019 interval. It is important to note that we chose to exclude values from 2010 to 2012 in this evaluation, after determining that they produced less reliable results due to high confidence predictions that missed residual effects of the Great Recession and significant disparities in inflation. This model surfaced *departures performed*, *seat count*, and certain *city pairs* like FLL-IAD as important features. Additionally, it highlighted the 737 Max Jet as an important feature in Q4 of 2018.

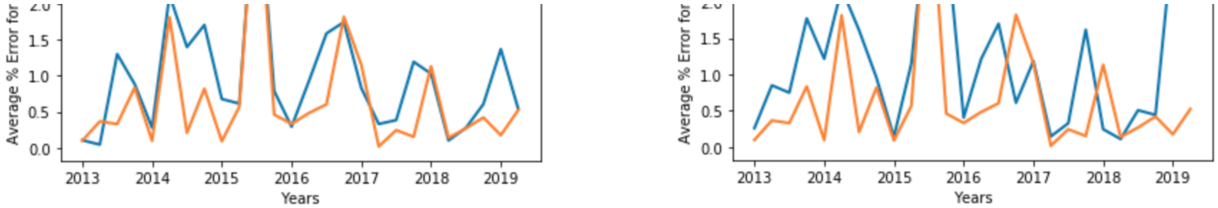
To further evaluate *how well* this model was performing, we compared it to the average percent error of quarterly Southwest Airlines revenue as projected by the Wall Street's revenue consensus. The WS model has an astounding average percent error of just **0.617%**. The graph below shows how we did against the WS estimate as well as the running error of our model over time.



We were happy with these results, but wanted to see if we could refine it even further. We applied the WS projections as an additional feature in our model and were able to produce an average percent error of **0.95%** (seen in the graphs below). Our model even outperformed the WS projections for some time periods (Q4 2017 and Q1 2018), seen here in places where the blue line crosses beneath the orange line. Unfortunately the graph shows that our model is pulling away from the WS projections for most time periods, but was still impressive, considering we were working with publicly available data with only a few weeks of data engineering.







You can read about possible influences for the inaccuracy spike in 2015, here.

Conclusions and Future Work

> *Can we create a Machine Learning model to predict airline stock price or revenue using public aviation data?*

After 4 experiments, we can safely say that predicting stock price from flight statistics is a losing battle, Twitter is stingy, we CAN accurately predict revenue from flight, stock, and expenditure data, and that there is room for many further explorations in this space.

We're cooking up ideas to try to beat WS's revenue predictions with a 1D Convolutional Neural Network approach to using monthly flight and financial data as a time series to predict revenue behavior.

We thank our professor, Alex Dimakis, for his support. Our code and datasets are linked to throughout this post, but you can see the full project and repo on **data.world** and **github**.

Thanks to Dale Pedzinski (hide) and Alex Hoganson.

Machine Learning