CS4641 Project

Brian Zhang, Sean Peng, Felix Wang, Max Xu, Alex Wang

Introduction and Background

Airlines use complex strategies and algorithms to price their tickets dynamically, based on a number of hidden variables, to maximize profit. As ticket buyers, we want to know when a ticket will be at its lowest price. In this paper, we will use ML models to predict future price changes. Some potential datasets we will use include the MachineHack Flight Fare dataset, which was used by published papers like [3]. Other datasets include flight data curated from the US Department of Transportation, and ticket data curated from Expedia shared on Kaggle. These datasets include features that were used by other works [1], [2], such as departure/arrival time, number of intermediate stops, and number of days until departure.

Problem Definition

For a specific plane ticket on a particular day in the future, we want to predict the ticket prices for every day up to and including the day of the flight. We will use regression, neural network, and tree based models to predict future prices. Our project is unique because we are predicting the prices for every day leading up to the flight, which is more granular than previous literature.

Some potential datasets we will use include:

- The MachineHack Flight Fare dataset, which was used by published papers like [3]
- Flight data curated from the US Department of Transportation
- Ticket data curated from Expedia shared on Kaggle

These datasets include features that were used by other works [1], [2], such as departure/arrival time, number of intermediate stops, and number of days until departure.

Potential Datasets

- Kaggle flight prices dataset
- MachineHack dataset
- US Department of Transportation

Methods

Preprocessing

- Data Cleaning
  - Our data set started out as 30 GB. In order to use it, we had to clean it by removing unnecessary feature. Columns like 'Arrival Time' and 'Departure Time' were removed because they were not relevant to our model. We also removed rows that corresponding to ticket prices of different classes, as we wanted our model to be consistent. After cleaning, we were left with a dataset of 1.8 GB which is much easier to work with.
  - And to make the dataset more suitable for later use in a regression algorithm, we added the feature of days to takeoff and performed one-hot encoding of the airline
- Data sampling
  - We wanted to make sure our model was learnable first and to create a linear model so we sampled only flights going from ATL to BOS.

Models

- Linear Regression (Implemented for Midterm)
  - We can use regression to predict prices as we expect prices to increase over time. We chose this model since it fits our problem description perfecly. We need a curve that minimizes the errors, so that the model can accurately predict plane ticket prices.
- Random Forest
  - We chose this model because random forests also can take in non-linear relationships, but they are much more interpretable than NN. Random forests also have hyperparameters we can tune to get the best model, like number of estimators.
- NN
  - We can use NN to predict non-linear relationships between time and flight prices. We used three hidden layers with 30, 30, and 25 neurons respectively, the adam optimizer, ReLU activation, and a max iteration of 2000.

Results and Discussion

After building our three predictive models, we evaluate their performance based on the mean squared error, root mean squared error, mean absolute error. These metrics will demonstrate the ability of the

model to understand the relationship between time and flight prices.

Linear Regression Results

We evaluated the performance of the linear and polynomial models predicting the price of a ticket up to 50 days before the flight date. Both models performed resonably well, with the degree 4 polynomial model perfoming better than the linear model. On average, it predicts within ~$23 of the actual ticket price on the test set. However, the polynomial model performs worse on the test set than the train set on all its metrics, which is the opposite of the linear model, suggesting that there may be overfitting.

To improve performance, we may need to prevent overfitting by adding regularization, perhaps by adding an L2 regularizer or lowering the degree of the polynomial model. A visualization of the models and the actual lowest prices is shown in the graph below.

Linear Model

	MSE	RMSE	MAE
Train	1203.06	34.69	28.13
Test	1006.16	31.72	27.45

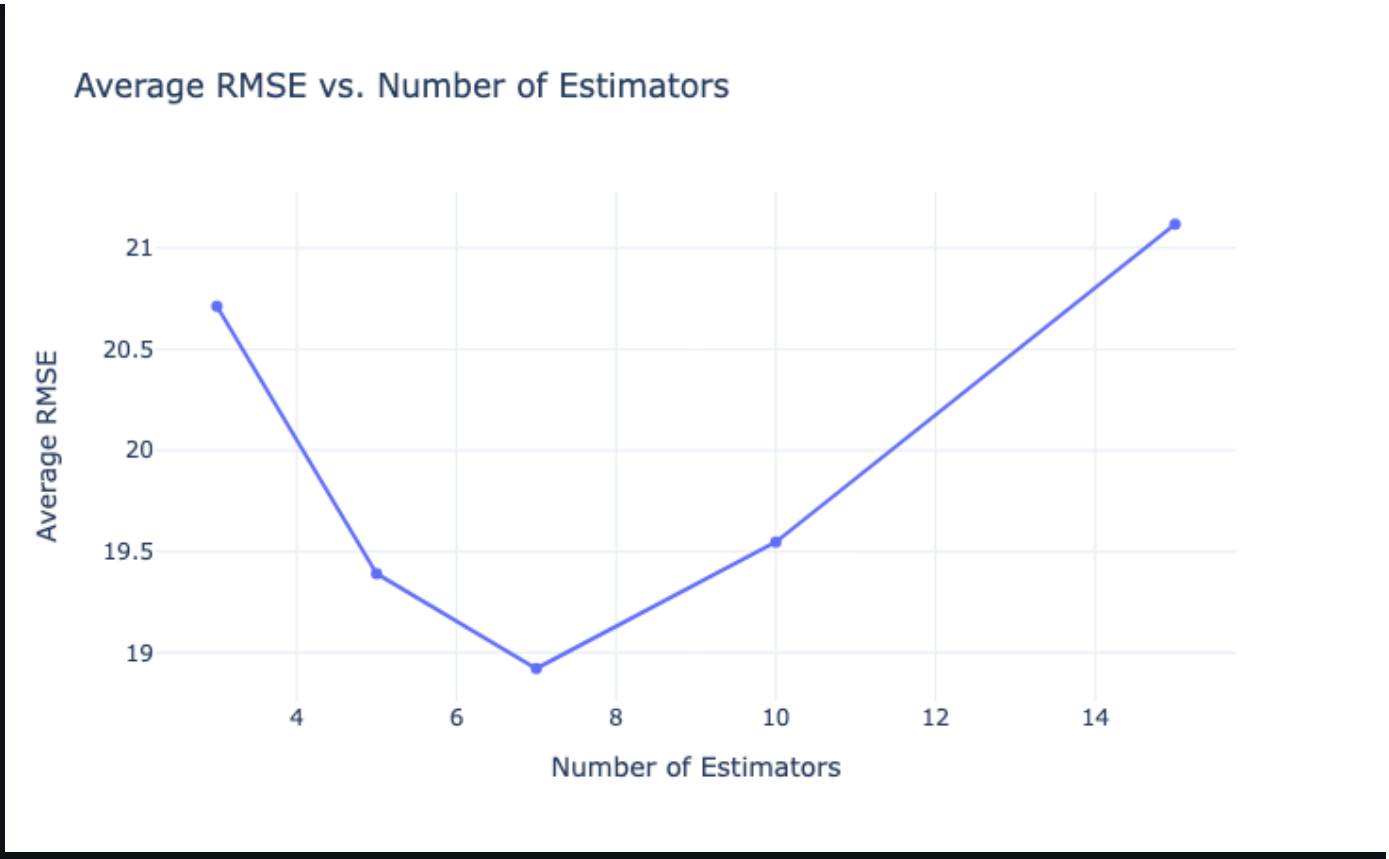
Polynomial Model (degree 4)

	MSE	RMSE	MAE
Train	267.79	16.36	13.24
Test	714.45	26.73	23.13

Linear Regression Model

Random Forest Results

We try to predict the price of a ticket leading up to the flight date using a random forest model. We used grid search and 10-fold cross validation to tune the hyperparameteres for the number of estimators and the max depth. The values we tested for max depth are 2, 5, 7, and 9, and the number of estimators are 3, 5, 7, 10, and 15. We found that the best performing model uses 7 estimators with a max depth of 7. The graph below shows that at 7 estimators we reach a minimum RMSE. Anything above that overfits the data and increases the error on the validation set.



GridSearch for no. estimators

The following table and graph shows the metrics and predictions of our best random forest model. On the training set, the metrics are much better than that of the linear models, but on the test set the results are comparable to that of the polynomial model. This suggests that there is still some overfitting. This can also be seen in the graph of the predictions. The line representing the predictions seems to be capturing too much of the noise of the training data, suggesting that pruning or regularization may be helpful.

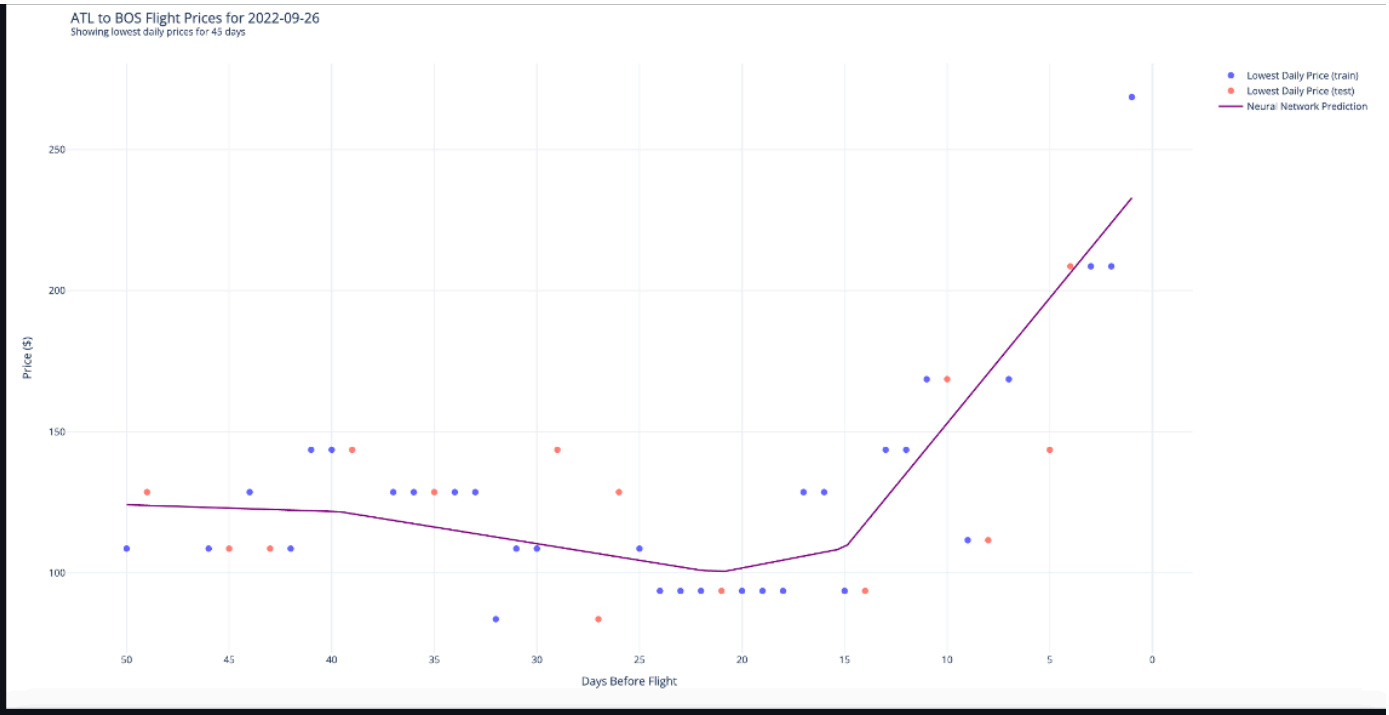
	MSE	RMSE	MAE
Train	111.25	10.55	6.34
Test	935.97	30.59	23.27

Random Forest Model

Neural Network Results

While we still faced some degree of overfitting in our neural network regressor, it was far less evident than the random forest regressor, with still relatively low error metrics in both the training and testing sets. We can see this visually in the graph as although the model prediction line follows the training set to a certain degree, it still predicts the testing data relatively well. To improve performance and decrease overfitting, we could experiment with using a more comprehensive dataset with more context and more flights than just ATL to BOS, as well as running experiments to determine the optimal amount of hidden layers and neurons per layer.

	MSE	RMSE	MAE
Train	321.77	17.94	14.91
Test	756.42	27.50	22.15



Neural Network Regressor

References

[1] K. Tziridis, Th. Kalampokas, G. A. Papakostas, and K. I. Diamantaras, "Airfare prices prediction using machine learning techniques," IEEE Xplore, Aug. 01, 2017. https://ieeexplore.ieee.org/document/8081365

[2] R. Ren, Y. Yang, and S. Yuan, "Prediction of Airline Ticket Price." Available: https://cs229.stanford.edu/proj2015/211_report.pdf

[3] M. Tuli, L. Singh, S. Tripathi, and N. Malik, "Prediction of Flight Fares Using Machine Learning," 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Jan. 2023, doi: https://doi.org/10.1109/confluence56041.2023.10048801

Gantt Chart

Link: https://docs.google.com/spreadsheets/d/1C99KDB_KTDGdgwayHkGORPX8w1IvCShse4sxoEilC3o/edit?gid=0#gid=0

Contribution Table

Link(Second Tab):
https://docs.google.com/spreadsheets/d/1C99KDB_KTDGdgwayHkGORPX8w1IvCShse4sxoEilC3o/edit?gid=0#gid=0

Project Slides

Below are our project slides:

Link:
https://docs.google.com/presentation/d/1_3RfbHeupxhppUFCOvPx13TWm5PZZk_DmBdBzAaHOXc/edit?usp=sharing