Predicting flight ticket prices from *EaseMyTrip.com*

Data Science Capstone Project 2

Adam Reichenbach

Springboard Data Science Career Track, June 2024 Cohort

Context

- ***EaseMyTrip.com*** is a 3rd party OTA (Online Travel Agency) website.

- It searches for flight price options across different airlines and makes recommendations based on the user's preference.

- It's like *Kayak.com*, except the flights are for travel between cities in the country of **India**.

- We are looking to make a machine learning model that can **predict the price of a flight ticket** with given variables.

Possible Stakeholders

- **Airline companies** wanting to forecast revenues and future pricing behavior.

- **Data Scientists** and/or **Product Managers** at the airlines who want to integrate and improve the model.

- **Travel agencies** or **other 3rd party OTAs** (like *Kayak*) who want to offer more competitive pricings and give more personalized recommendations.

- **Passengers** who may want to buy their tickets for a given destination more strategically.

Data source and Information

- The data was already gathered and mostly cleaned for *Kaggle.com* by author Shubham Bathwal
  - Data source: https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction

- Data gathered from *EaseMyTrip.com* were distinct tickets purchased over course of 50 days:
  - February 11th - March 31st, 2022.

- The dataset contains:
  - Over 300,000 rows of data
  - 12 features/columns are part of the original dataset from Kaggle

Data source and Information continued

__Categorical Features__:

- ***Airline***: Airline company the ticket is booked through.

- ***Flight***: An airline-unique flight number that is assigned to the departing flight.

- ***Source City***: The city where the flight is departing from which includes India's top 6 metro cities of:
    - **Delhi, Mumbai, Chennai, Hyderabad, Kolkata, and Bengaluru** (or **Bangalore**).

- ***Destination City***: City the flight arrives at. Contains the same 6 cities as *source city*.

- ***Departure Time***: Derived feature of time bins as to what time of the day the flight leaves. 6 distinct times:
    - **Early morning, morning, afternoon, evening, night, late night**

- ***Arrival Time***: Same derived feature as *departure time*, but for when the plane lands.

__Categorical Features cont'd__:

- ***Class***: Seat quality of the ticket purchased. Only 2 distinct values (**economy** and **business**).

- ***Stops***: Derived categorical feature with the number of stops between the source and destination city.
    - **0, 1, 2+**

__Continuous Features__:

- ***Duration***: The total travel duration in hours between the flight's departure time from the source city and arrival time at the destination city.

- ***Days left***: The amount of days remaining between the time of booking the flight and the day of the flight.

- ***Price***: The target variable of this project. Price of the flight ticket in **Indian Rupees (₹)**

Data Wrangling and Exploratory Data Analysis (EDA)

Data Wrangling

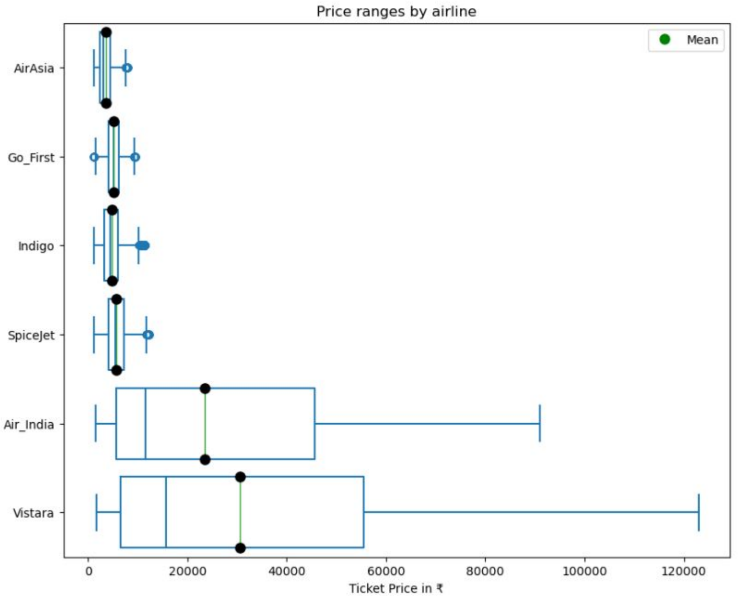
- The data was already pretty cleaned up by the author upon extraction.
- There were not any missing values in any of the fields.
- Only 2 big changes that needed to be made to the dataframe:
    - There were several rows of duplicates that were dropped. It dropped the amount of data to 297,940 observations, so not a huge overall difference from the original dataset size.
    - The categorical feature *flight* has 1,561 unique values. The feature was dropped as it wasn't deemed necessary or relevant for making the final model.

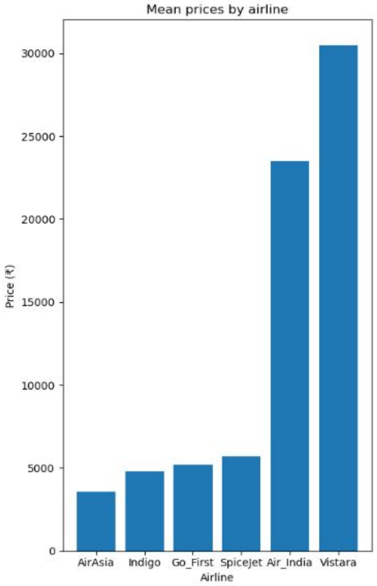
Exploratory Data Analysis

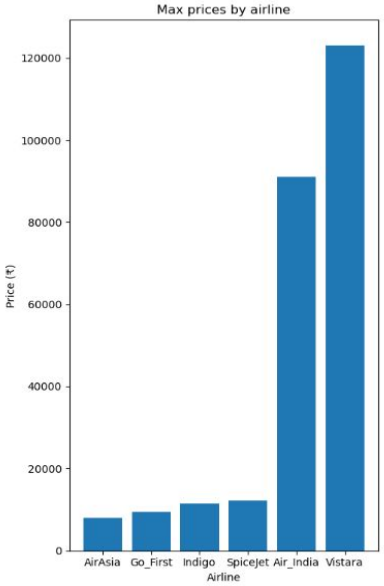
The questions that I chose to explore:

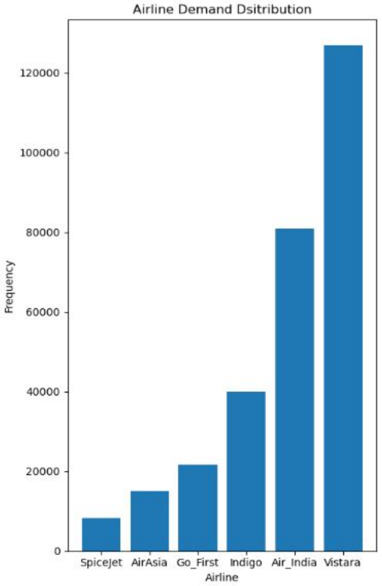
- Does the ticket price vary with **airline**?
- Do the **cities** have any impact?
- How does the ticket price vary between **economy** and **business class**?
- Does **departure** and **arrival time** affect ticket price?
- How are the prices affected when bookings are made **1 or 2 days left** before departure?

EDA - Airlines





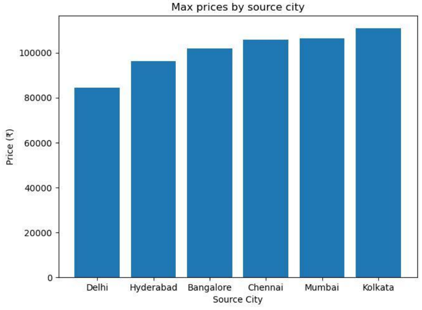


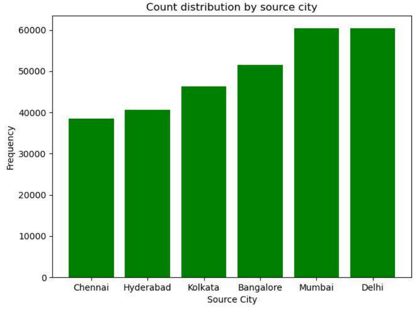


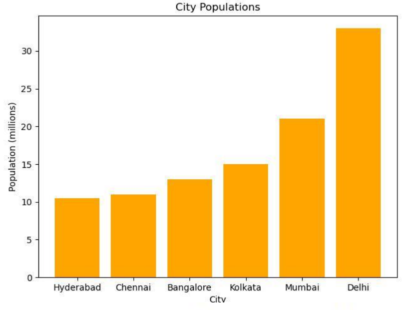
- **Air India** and **Vistara** look to have the largest ranges of ticket prices as well as the highest mean and median prices.
- The remaining 4 airlines all have similar ranges, mean, and median prices.

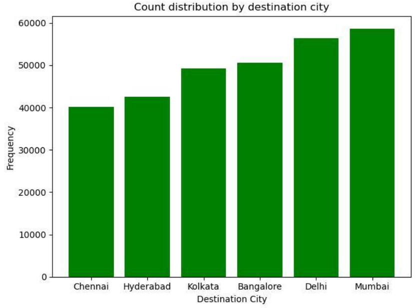
- The wide range of prices is likely due to the demand of each airline where **Air India** and **Vistara** have a clear separation from the rest of the pack.
- The two leading airlines make up **over 71%** of the tickets bought from the gathered data.

EDA - Cities

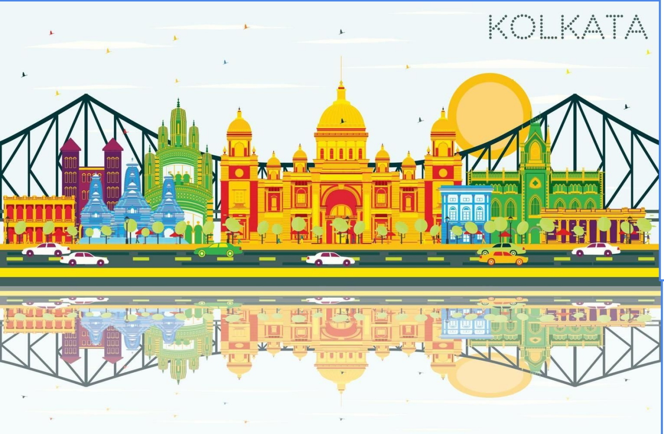








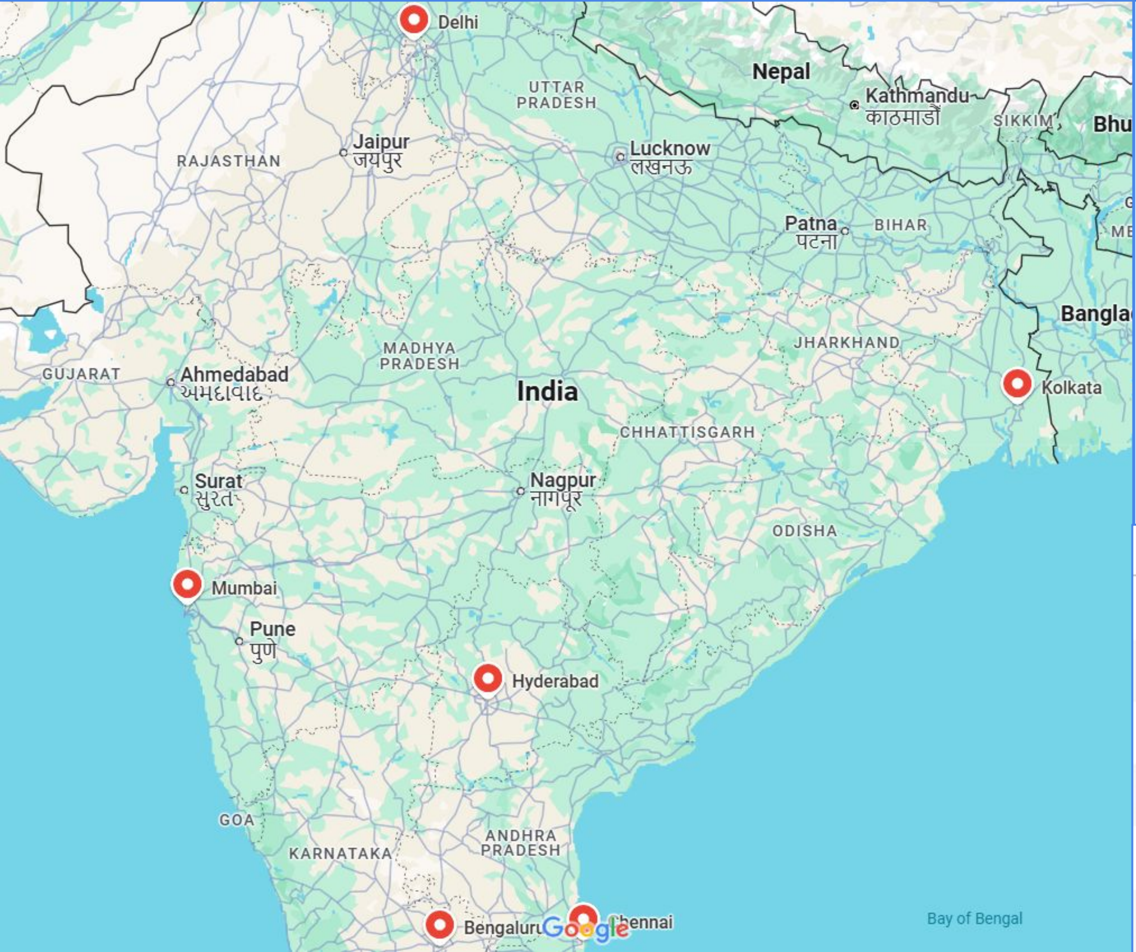
	Population (millions)
Bangalore	13.0
Chennai	11.0
Delhi	33.0
Hyderabad	10.5
Kolkata	15.0
Mumbai	21.0



- **Delhi** and **Mumbai** are the two frontrunners for number of tickets bought by city.
- They are also have the highest and 2nd highest populations respectively, which makes for a logical correlation with the demand count.
- City population does have to seem to almost follow the same trends as ticket demands for both source and destination city.

- **Kolkata** has the highest prices by city as both source and destination while **Delhi** has the lowest.

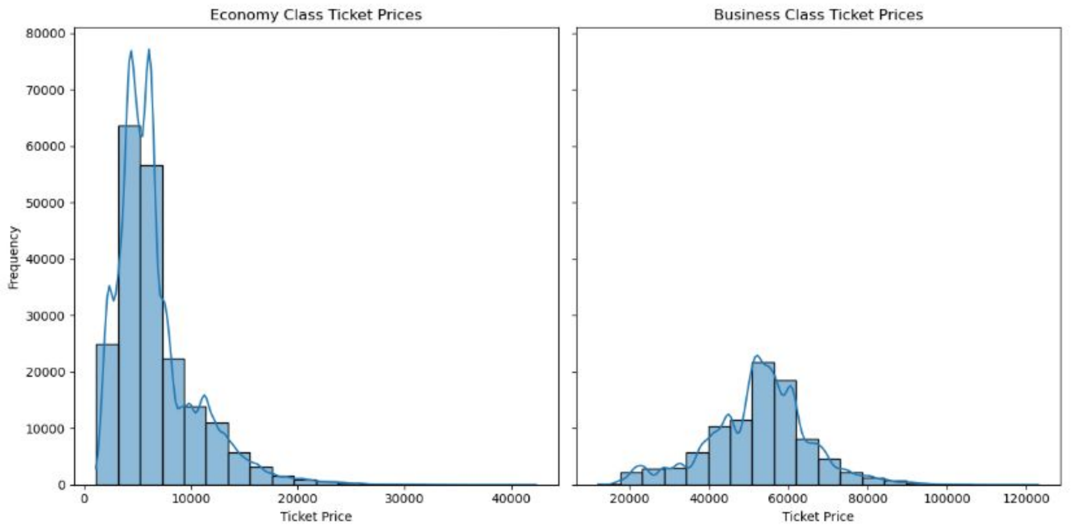
EDA - Cities (continued)

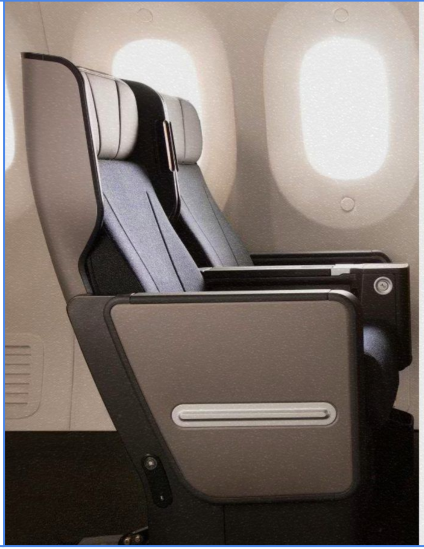


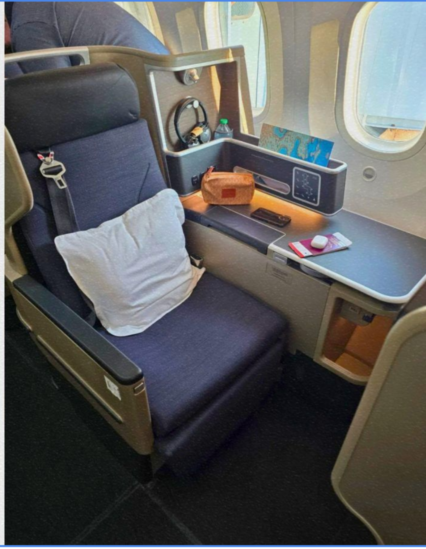
- **Kolkata** is fairly geographically isolated from the rest of the other big metro cities of the country which could explain it’s high prices. Longer **duration** of flight could lead to higher prices due to jet fuel usage and possible layover stops.
    - Geographic isolation doesn’t seem to affect **Delhi** the same way since it has the lowest price by city from the previous slide.
- **Delhi** serves as a major hub for some of the airlines featured, while **Kolkata** doesn’t have any of the listed airlines headquartered there.
- **Delhi’s** role as the capital of the country and high population likely leads to more available flights from it’s airport while lack of an airline HQ likely leads to **Kolkata’s** higher prices.

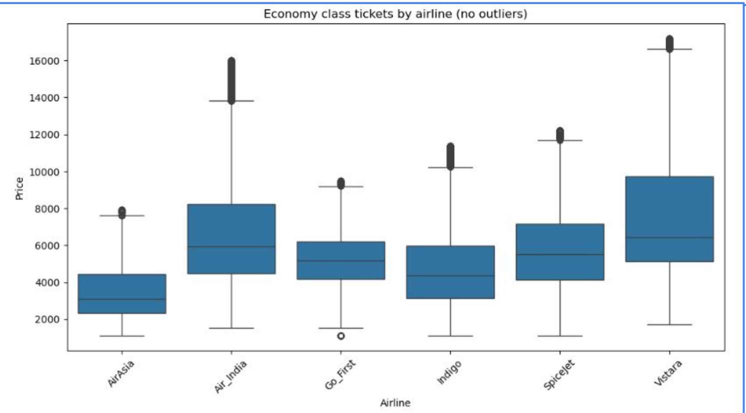
source_city	cheap_dest	cheap_price	exp_dest	exp_price
Bangalore	Chennai	1603	Kolkata	98919
Chennai	Hyderabad	1105	Mumbai	98912
Delhi	Chennai	1998	Kolkata	97337
Hyderabad	Chennai	1543	Bangalore	97767
Kolkata	Hyderabad	2436	Delhi	98543
Mumbai	Chennai	1890	Chennai	98972

EDA - Ticket Class



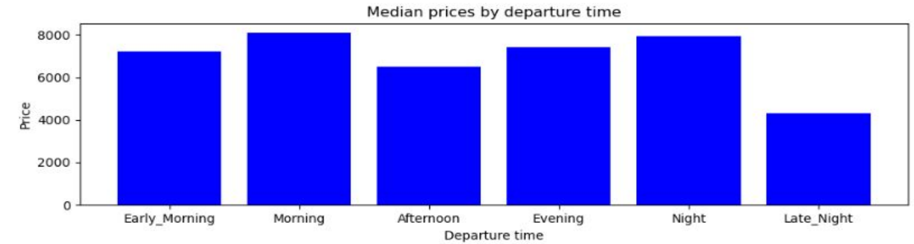


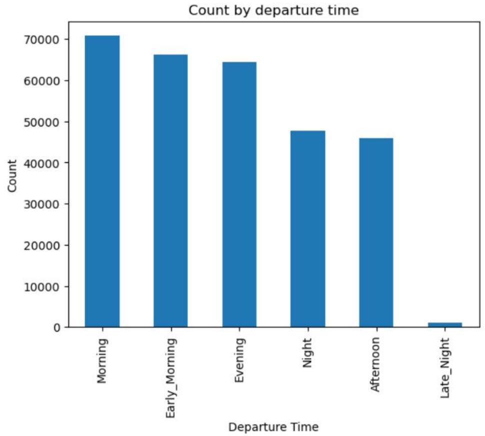




- Economy median price: ₹5780 ($67.28)
- Business median price: ₹53164 ($618.82)
- Class median price difference pct: 819.79%

EDA - Departure and Arrival Time

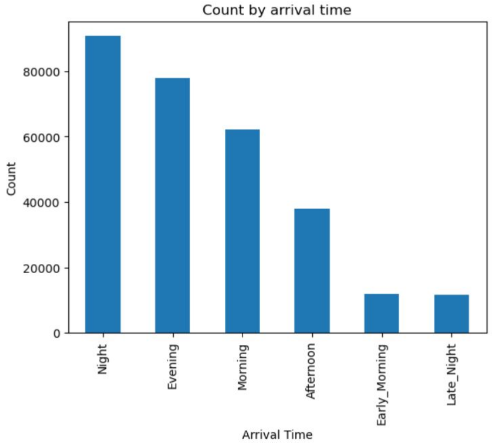




- Demand and price trends seem to match up.

- **Early morning** has a **high** demand/price for **departure** time, but **very low** for **arrival**.

- **Late night** has the lowest demand and prices for both as a departure and arrival time.

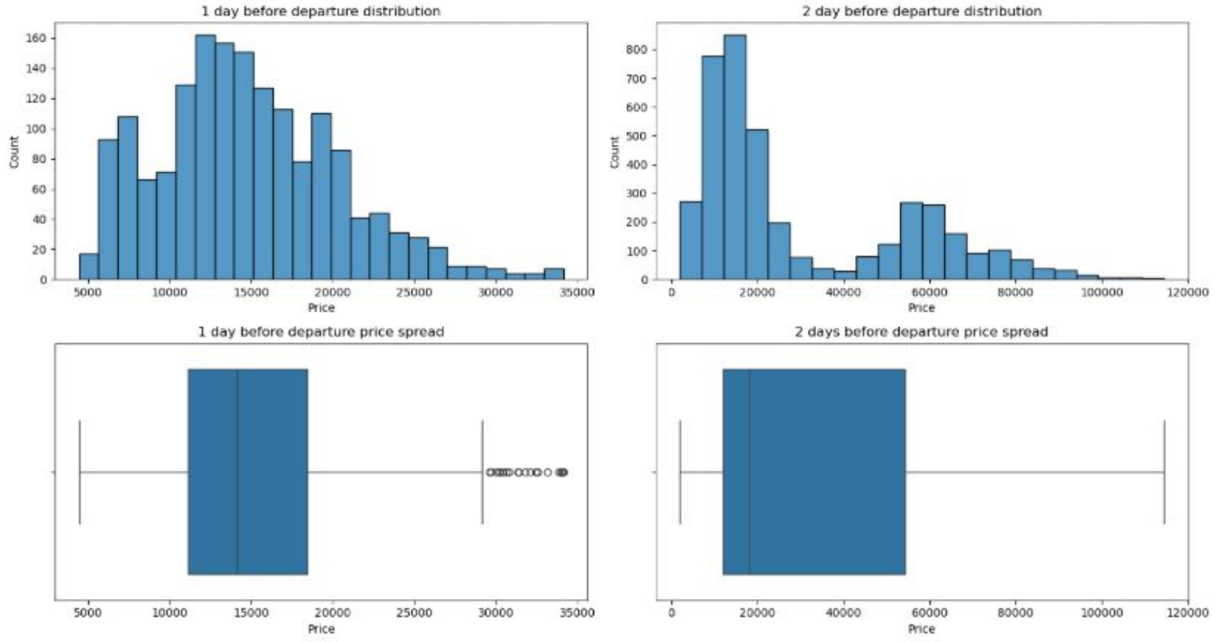


EDA - Days before departure

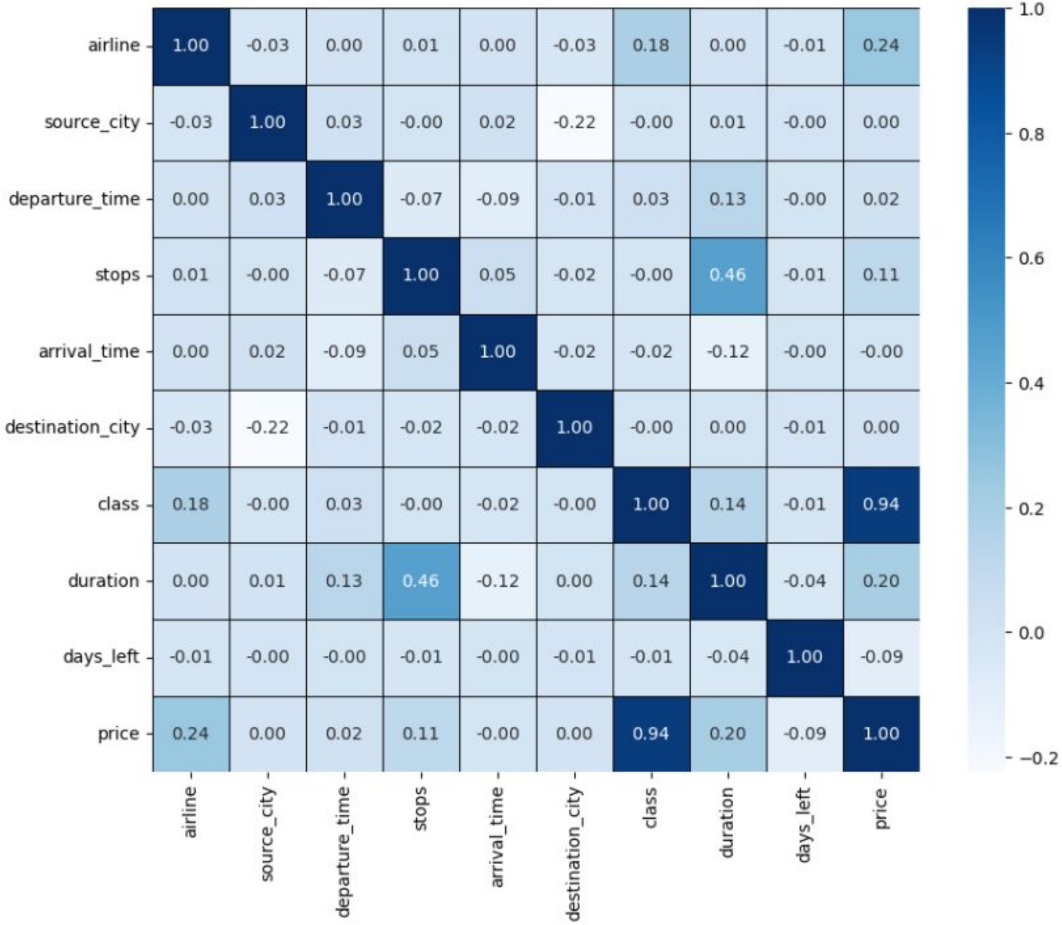
Days Left	Mean Price	Median Price	Max Price
1	₹14760	₹14154	₹34134
2	₹30258	₹18039	₹114523

Price drops from 2 to 1 day before departure:

- Mean price pct Δ: 51.22%
- Median price pct Δ: 21.54%
- Max price pct Δ: 70.19%



Preprocessing



Encoding

Each categorical feature was manually encoded so that the numerical values can fit the models.

Scaling

Scaled the features that aren't ticket prices.

Train-test-split

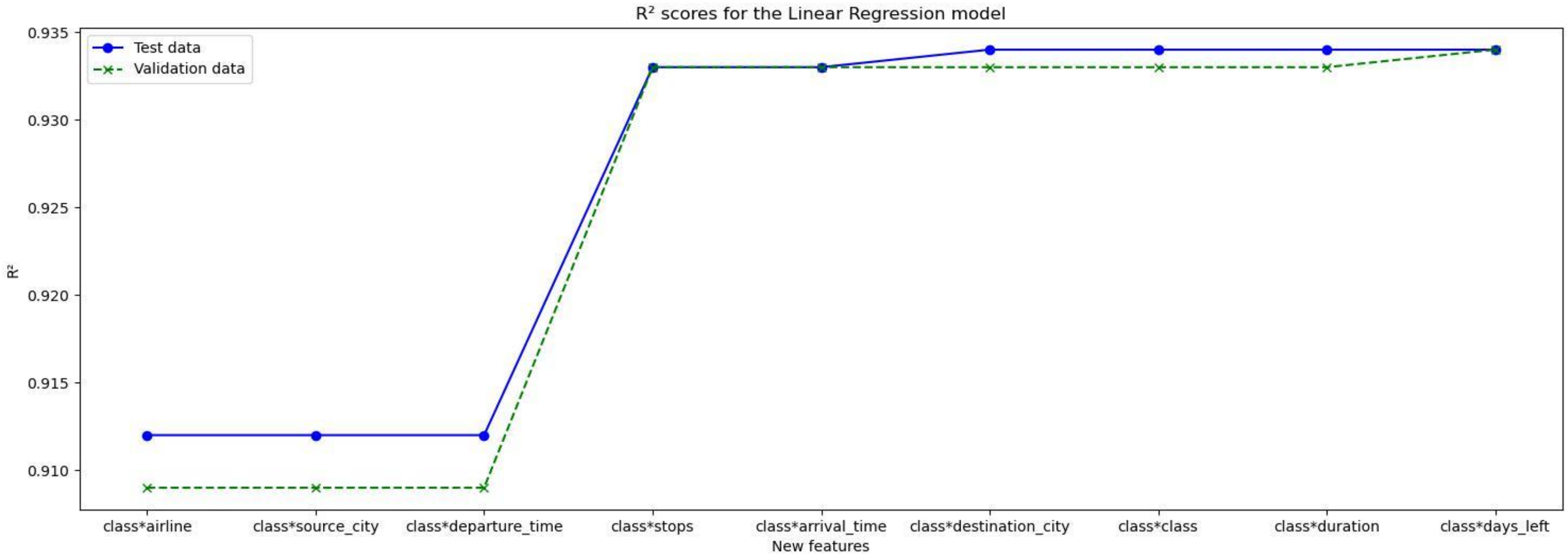
The data was randomly split 80/10/10:

- 80% **training data**
- 10% **testing data**
- 10% **validation data**

Correlation heatmap after encoding the categorical features

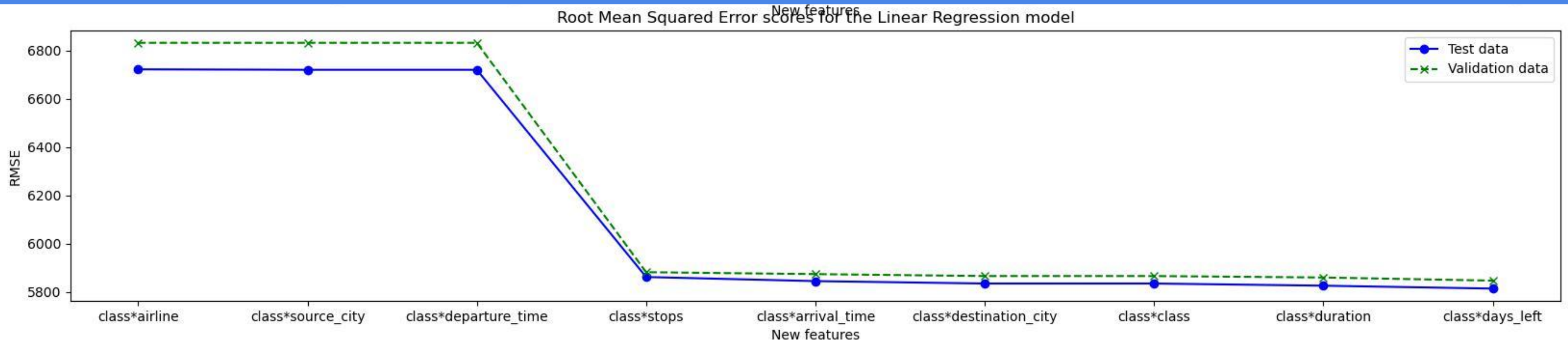
- Ticket class has the strongest correlation with pricing.
- This verifies the class feature from our EDA that the seat class has one of the biggest effects on pricing.
- Class will be the first driving feature to be included with a linear regression model.

Modeling - Linear Regression Metrics - R^2

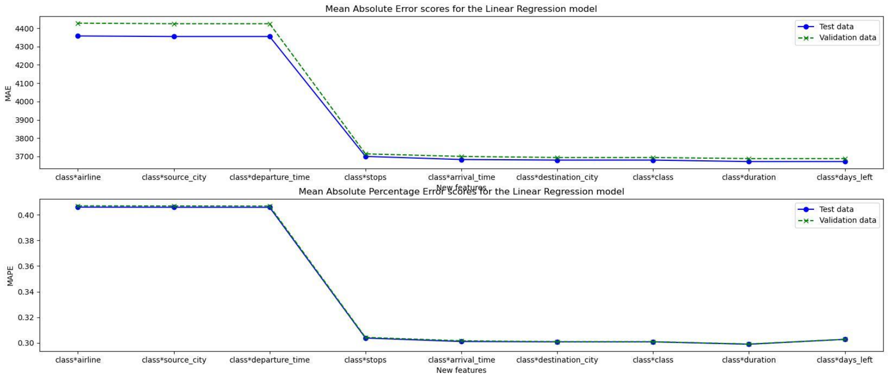


To start, R^2 for class multiplied with **stops**, **arrival_times**, **destination_city**, **duration**, and **days_left** are the front runners.

Modeling - Linear Regression Metrics - RMSE, MAE, MAPE



MAE, MAPE plots



Final metrics for LR

Features	R^2	R^2 val	RMSE	RMSE val	MAE	MAPE	MAE val	MAPE val
class*airline	0.912	0.909	6722.0	6832	4358.0	0.405869	4428.0	0.406919
class*source_city	0.912	0.909	6720.0	6832	4355.0	0.405767	4425.0	0.406792
class*departure_time	0.912	0.909	6720.0	6832	4355.0	0.405754	4425.0	0.406717
class*stops	0.933	0.933	5861.0	5881	3700.0	0.303820	3714.0	0.304333
class*arrival_time	0.933	0.933	5844.0	5873	3683.0	0.301125	3700.0	0.301685
class*destination_city	0.934	0.933	5834.0	5865	3680.0	0.300859	3694.0	0.300963
class*class	0.934	0.933	5834.0	5865	3680.0	0.300859	3694.0	0.300963
class*duration	0.934	0.933	5825.0	5859	3672.0	0.299008	3688.0	0.299105
class*days_left	0.934	0.934	5813.0	5846	3672.0	0.302810	3688.0	0.302829

Modeling - Lasso and Gradient Boosting Regression Models

Lasso

Performing a 5-fold cross-validation on the fitted lasso model yielded an optimal parameter of $\alpha \approx$ **9.9.**

Models Metrics (test vs. validation data)

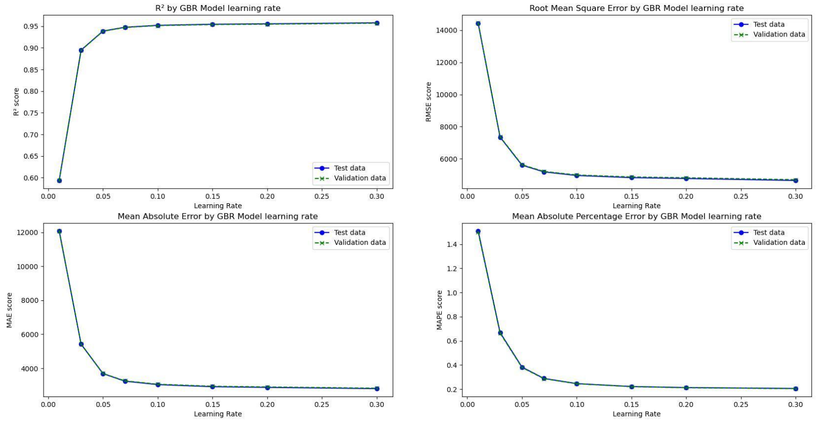
Model	R^2	R^2 val	RMSE	RMSE val	MAE	MAE val	MAPE	MAPE val
Lasso	0.9339	0.9334	5824.0	5853.0	3679.0	3694.0	0.30502	0.30505

Model	R^2	R^2 val	RMSE	RMSE val	MAE	MAE val	MAPE	MAPE val
Gradient Boosting	0.956	0.955	4759.0	4805.0	2860.0	2888.0	0.2108	0.21

Gradient Boosting

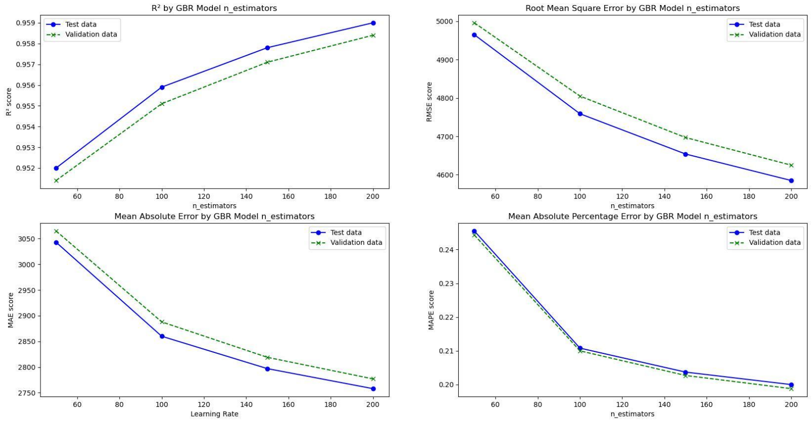
GBR Metrics by learning rate

Chose learning rate of **0.1**



GBR Metrics by n_estimators

Chose **100 n_estimators**



Final Metrics, Model Selection, and Final Thoughts

Final Metrics

Model	R^2	RMSE	MAE	MAPE
Linear (class*duration)	0.934	5825	3672	0.299
Linear (class*days_left)	0.934	5813	3672	0.303
Lasso	0.934	5824	3679	0.305
GBR	0.956	4759	2860	0.211

Model Selection

- The **Gradient Boosting Regressor** model looks to be the best selection since its metrics have clear separation from the other models. Specifically it has quite a bit lower **MAPE** than the rest.

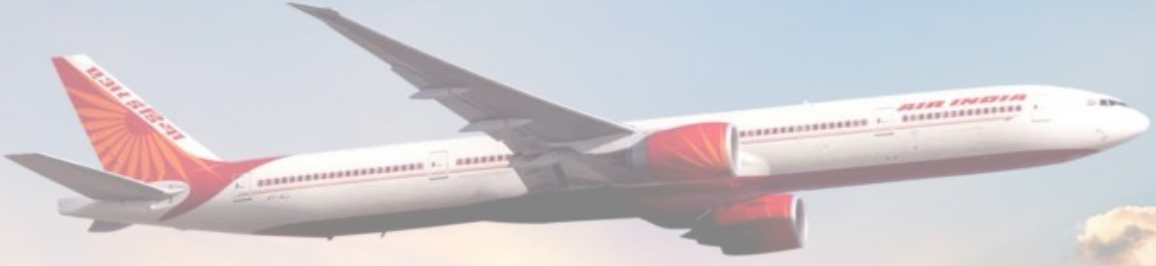
Final Thoughts

- The original data gathered for *Kaggle* was only collected over 50 days over 3 years ago. It also doesn't consider the days of the week for each flight which could affect pricing with weekend travel demand.

- Thorough GridSearchCV for the Lasso and Gradient Boosting model was limited by computational power/time. Had to resort to other methods for tuning the models.

- Even with better metrics, there still is a risk of the models overfitting.

Thank you!



Adam Reichenbach
Email: adre9701@colroado.edu
LinkedIn: https://www.linkedin.com/in/adam-reichenbach-578947185/
Project Github Repo:
https://github.com/DJSydon/Springboard-Capstone-Project-2/tree/main

Special thanks to my SpringBoard Data Science mentor:
Silvia Seceleanu
Founder, ML/AI (ex Block, Udemy, JP Morgan)