Flight Price Prediction

24 March 2024

Aditya Shah
Aryan Chaudhary
Jhomar Gharbarran
Rehan Nasir

Table of Contents

Description of Dataset

Overview

The dataset consists of data from the website "Ease My Trip" with flight booking options for India's top 6 metro cities from February 11th to March 31st, 2022. It provides flight options that a user could book with features to categorize the data such as the airline, flight code, source and departure cities and times, amount of stops, seat class, total flight time, the number of days booked before the trip and the price of the flight.

There are 300,261 instances and 11 features in the dataset.

Features

There are a total of 11 features provided in the dataset:

Feature	Description	Type	Values
airline	The name of the airline company for a flight.	Nominal	"Air_India", "AirAsia", "GO_FIRST", "Indigo", "SpiceJet", or "Vistara"
flight	The flight code for a flight.	Nominal	"SG-8709", "I5-764", etc.
source_city	The city where the flight takes off.	Nominal	"Bangalore", "Chennai", "Delhi", "Hyderabad", "Kolkata", or "Mumbai"
departure_time	The time of day when the flight takes off from the source city.	Nominal	"Early_Morning", "Morning", "Afternoon", "Evening", "Night" or "Late_Night"
stops	The amount of layovers the flight has between the source and destination cities.	Nominal	"zero", "one", "two_or_more"
arrival_time	The time of day when the flight arrives at the destination city.	Nominal	"Early_Morning", "Morning", "Afternoon", "Evening", "Night" or "Late_Night"
destination_city	The city where the flight will land.	Nominal	"Bangalore", "Chennai", "Delhi", "Hyderabad", "Kolkata", or "Mumbai"
class	The seat class of a flight.	Nominal	"Economy" or "Business"
duration	The flight's total duration to travel from the source to the destination city.	Numeric	0.83-49.83 hours
days_left	The amount of days the flight was booked before the trip.	Numeric	1-49 days
price	The price of the flight.	Numeric	(INR) 1105 - 123071

Look at the Big Picture

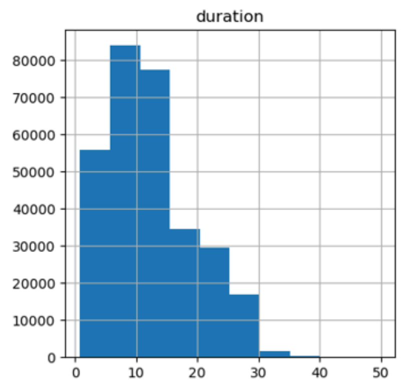
Framing the Problem

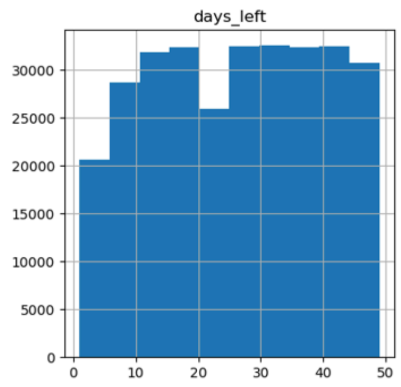
- Supervised learning: the dataset contains training examples that are labelled.
    - Airline, flight, city, class, etc.
- Regression Task: the goal is to predict flight prices, which are numerical values.
    - Flight prices will be displayed in Indian Rupees (INR).
- Batch Learning: the dataset contains examples from February 11th, 2022 - March 31st, 2022. The dataset is not receiving any new data, meaning adjusting for it is unnecessary.

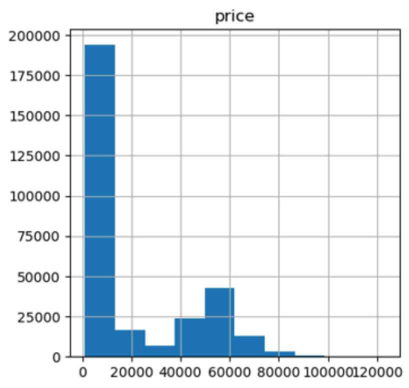
Task: Use predictions to inform the user when prices are the cheapest for any given flight.

Exploratory Data Analysis

Histogram





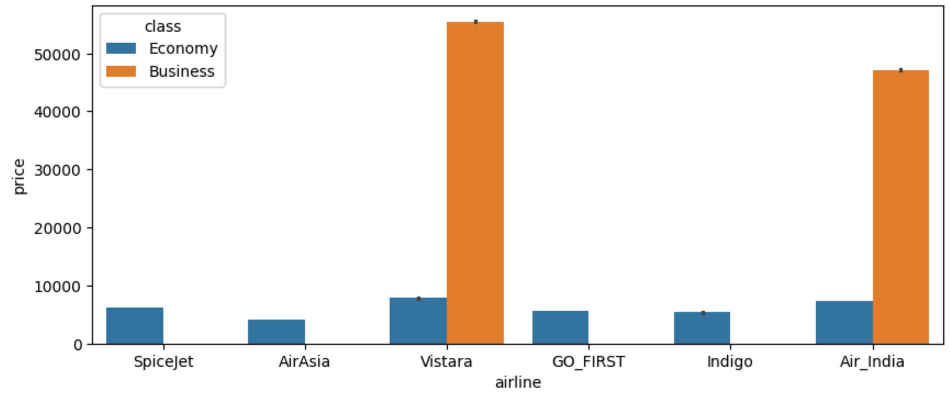


The first graph is right-skewed and shows that over 80,000 flights have a duration of 10 hours, with the inverse being around 1,000 flights having a duration of 30 hours.

The middle graph displays flight tickets by how many days left you have to purchase them. The graph seems to have a uniform distribution, With most flights having 10-50 days left. There are 2 outliers with 20,000 flights having 0 days left and 25,000 flights with 20 days left.

The right graph is also right-skewed and groups the number of flight tickets by their price. The most common price seems to be around ~10,000 INR, with over 190,000 flight tickets costing that much. The lowest with 1,000 flights costing around 80,000 INR.

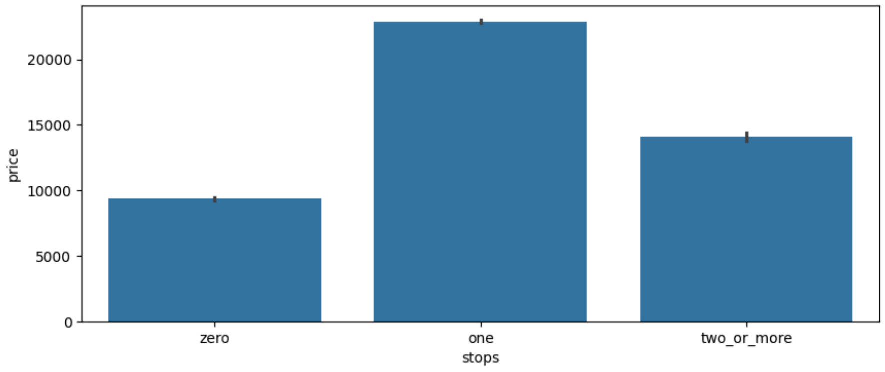
Airline Price Comparison



This bar graph compares the average price (INR) of each airline. Tickets for the economy class for all airlines are under 10,000 INR, with AirAsia being the cheapest around ~4,000 INR and Vistara the most expensive, at ~9,000 INR.

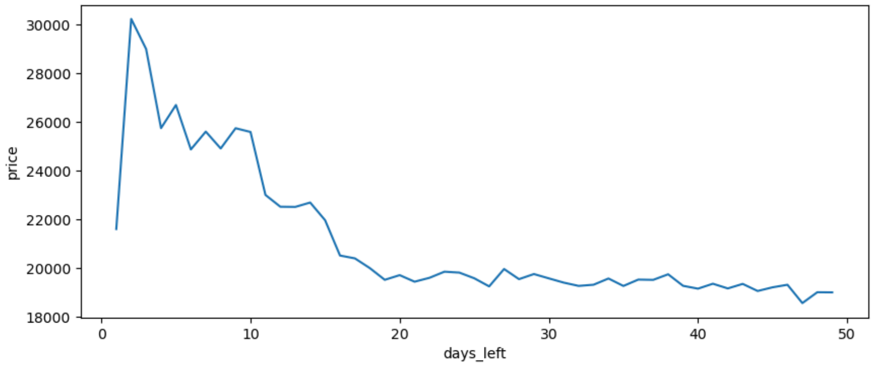
Vistara and Air India are the only airlines with business class and are about 5x more expensive than the economy class. Between the 2 airlines, Vistara's average business-class ticket is the most expensive, averaging over 50,000 INR.

Average price comparison based on the number of stops



Direct flights on average is the cheapest, being around ~10,000 INR. Indirect flights with one-stop are the most expensive, costing more than 20,000 INR. Indirect flights with 2 or more stops average around ~12,000 INR.

Ticket prices based on the date purchased

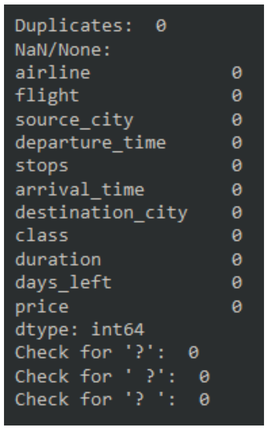


This graph indicates that tickets purchased further in advance are significantly cheaper than tickets bought sooner to the day of departure. Prices start hiking after 20 days before departure.

Data Cleaning and Preprocessing

Data Cleaning

The dataset we used contains no duplicate rows nor does it contain any missing values.

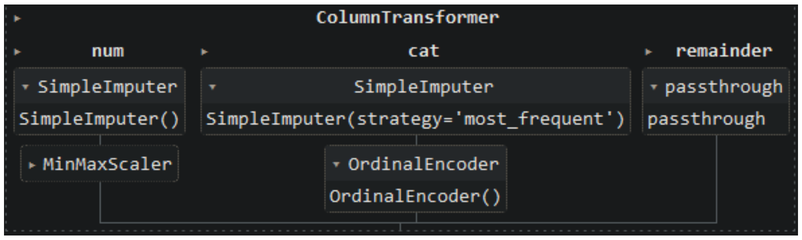


```
Duplicates:  0
NaN/None:
airline            0
flight             0
source_city        0
departure_time     0
stops              0
arrival_time       0
destination_city   0
class              0
duration           0
days_left          0
price              0
dtype: int64
Check for '?':  0
Check for ' ?':  0
Check for '? ':  0
```

The dataset did contain a redundant column that labelled each row with a number. The column affected all of our visual data, so we decided to remove it completely.

Preprocessing

For feature scaling, MinMaxScaler was the more optimal choice because none of the histogram visuals followed a Gaussian distribution.

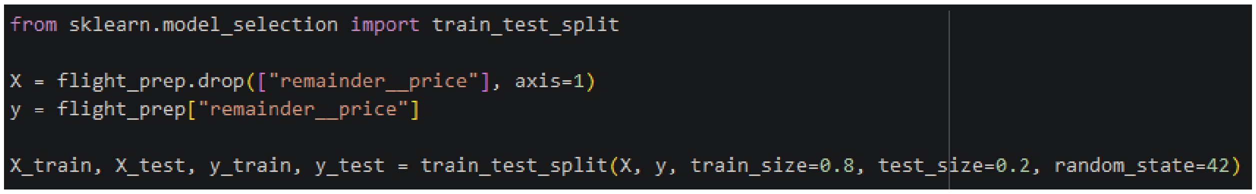


1. Two columns were created, one for numerical data and one for categorical data.
    a. The column for numerical data consists of the “duration” and “day_left” features.
    b. The column for categorical data consists of “airline”, “flight”, ‘source_city”, “departure_time”, “stops”, “arrival_time”, “destination_city” and “class” features.
    c. Price was removed from the numerical column because that is what we are trying to predict.
2. Created a pipeline for both columns. For the numerical column, the “mean” strategy was used to fill in missing values via SimpleImputer, while the “most frequent” strategy was used for the categorical column. The data was scaled for the numerical pipeline with MinMaxScaler() and encoded with OrdinalEncoder() for the categorical pipeline.

Machine Learning Models

Test Set

The dataset was split 20% for testing and 80% for testing.



```
from sklearn.model_selection import train_test_split

X = flight_prep.drop(["remainder__price"], axis=1)
y = flight_prep["remainder__price"]

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, test_size=0.2, random_state=42)
```

Training Set

The 3 models we used were:

- Linear Regression
- Elastic Net
- Random Forest Regressor (with hyperparameter max_depth = 8)

Performance Metrics

For each model, we evaluated the correlation coefficient, mean absolute error, mean squared error, and root mean squared error.

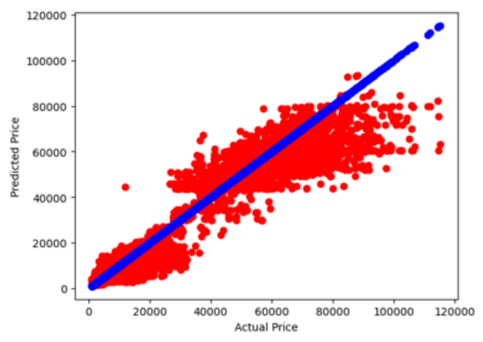
Model	R^2 Score	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Linear Regression	0.9045789774846467	4619.351827705692	49187845.632480934	7013.404710444203
Elastic Net	0.5030942976137227	13174.492205149032	256146081.2154158	16004.564386930868
Random Forest	0.9613951493107497	2550.8790021788823	19900116.204081617	4460.9546292337045

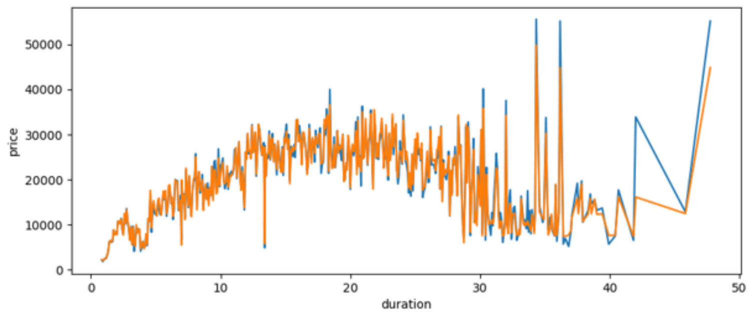
From the metrics, the elastic net had the highest errors and lowest correlation coefficient out of all the models, followed by linear regression. The random forest model had the lowest errors and the strongest positive correlation of about 0.96.

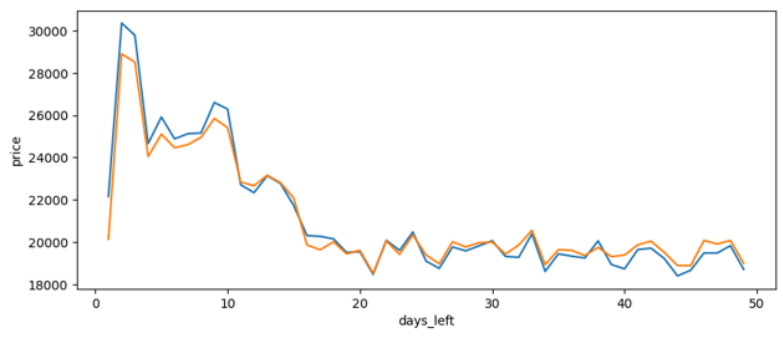
Based on these evaluations, the elastic net model performed the worst with linear regression following it, allowing us to conclude that the random forest regressor model was the best.

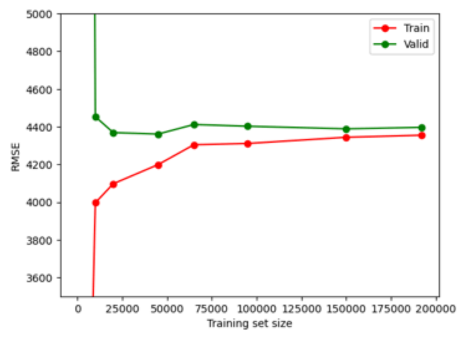
Findings

Given that the random forest regressor was our best model based on the metrics mentioned previously, we could now investigate how our model's predictions compare to the actual values in our dataset with graphs.









The red dots and yellow lines are our model and the blue dots and lines are the actual values. The top left graph compares the actual price of each ticket to what our model predicted. We can see that for each ticket, many of them are predicted to be lower or higher than the actual price. It also seems to underestimate the prices of the most expensive tickets and plateaus in the 40000-80000 INR range.

The top right and bottom left graphs show how close our model is in line with the actual data in the days left and the duration of the flight. It is close at some points but not exactly. We can also see severe underestimations when there are less than 10 days left or when the duration is more than 40 hours.

The bottom right graph shows the learning curve for our model. We can see that the error is high and that the two lines almost converge but there is still a gap between them. This implies that there is overfitting or the variance is high. By lowering the max depth hyperparameter in our model, we could get the lines to converge more but at the cost of more error and lower accuracy.

Limitations and Next Steps

Limitations

As our dataset was extremely large, having over 300,000 instances, graphing the model or trying other algorithms without certain hyperparameters to limit the training would take from 20 minutes up to more than an hour to complete. An example would be the neural network algorithm or support vector regression that we ultimately gave up on as it took over an hour to complete.

Next Steps

Based on our findings there is a lot more room for fine-tuning our model to better predict the ticket price. For example, we could have used grid search or randomized search to find better hyperparameters to use, as we had the problem of getting a low R^2 score where linear regression would score better or severe overfitting if we increased the depth too much despite getting extremely high accuracy (such as an R^2 of 0.99). Another option would be to explore other algorithms that could fit better with our data.

There was also the option of cleaning our dataset a bit more. The feature with the flight number did not appear to have any importance for predicting the price of a ticket and could have been removed to reduce the complexity of our model. It is also possible that we could have removed some of the outliers to lower our error or increase our R^2 score. Flights randomly spiking in price due to unknown real-life factors that had been included in our data are usually unpredictable, so it could have been beneficial for our model.