Capstone Final Report: India Airfare Prediction

Problem Statement

There are approximately 100,000 flights daily worldwide. Airfare is notoriously dynamic and fluctuates frequently due to a variety of factors such as seasonality, demand, destinations, and competitor pricing. The unpredictability makes it challenging for budget travelers to determine the best time to book a flight, often resulting in overpaying. The objective was to build a predictive model to highlight cost-driving factors and help raise customer satisfaction through saving money, while also optimizing pricing strategies, improving user retention by providing accurate fare predictions and reducing customer churn by offering timely and reliable recommendations. So, what factors could a future traveler focus on in order to minimize costs on a domestic trip in India? Could some of these factors be a universal focus for worldwide travelers?

This project focuses on a dataset containing flight features specific to India, as well as factors universally relevant to travelers worldwide. By leveraging this dataset, we aim to develop an airfare prediction model that identifies the key drivers of price fluctuations. Such a model has the potential to make travel more accessible and affordable, saving millions of travelers money by enabling informed booking decisions. For travel platforms and apps, integrating accurate airfare prediction technology not only enhances user experience but also drives customer loyalty and engagement, reinforcing its value as a competitive advantage in the travel industry.

Data Wrangling

The pre-cleaned dataset came from Kaggle.com. The flight dataset contained approximately 445k data points along 13 columns. Although the data had come clean, there were still alterations to be made to the dataset in order to explore and make use of the data in the way that was necessary to this project. First we split the Date_of_journey column into year, month, and day columns (eventually got rid of the year column since the data only occurs over a three month period in the same year). To eliminate duplicates, entries were compared based on flight_code, Destination, Fare, Arrival, Duration_in_hours, and several other columns, and any duplicate records were removed.

Once the data was cleaned up until that point it was time to start looking at some of the characteristics of the dataset as a whole and doing some preliminary exploratory data analysis. At first glance, it was noted that the dataset was categorically dense. There were few to little continuous features in the dataset, with Fare being one of them. Once this was discovered, more steps were taken to conduct further analysis on the values of each feature and their counts. The categorical features in the dataset seemed to have anywhere between three and seven values per category. This will be important when we create dummy variables in future steps of the project. Along with looking at the counted values of the categorical features, there was some aggregation of the categorical features with the mean of the column Fare. In this part of the preliminary exploration it was discovered that Ahmedabad was the most expensive destination and the longest flight in hours. At this point there is a baseline understanding of the data, and it is very important to notice that there are many outliers in our dataset due to the natural variability in airfare; this can be visualized in Figure 1, which shows the distribution of the Fare column in rupees. The figure shows bimodal distribution with right skew. Due to this discovery, the focus became the median as the main central tendency metric of all further analysis.

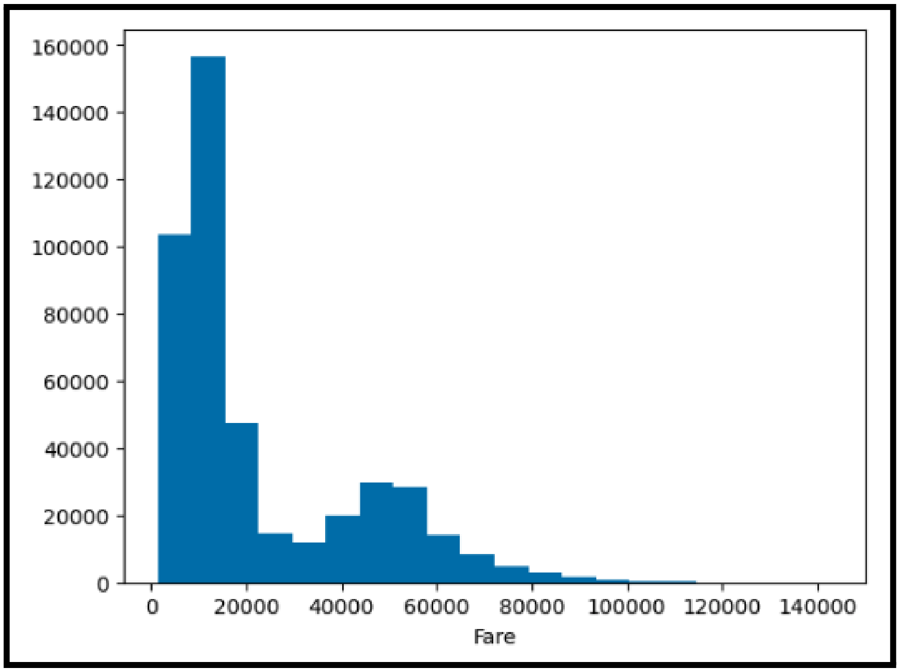


Figure 1: Bimodal distribution of flight Fare for Indian domestic flights conveyed in rupees

Exploratory Data Analysis

Data overview and cleaning

The exploratory data analysis (EDA) process aimed to uncover patterns, detect anomalies, and understand key trends in the airfare dataset to prepare it for predictive modeling. Due to preliminary EDA in the previous data wrangling step it was noted that the median would be the primary metric to use during the analysis portion due to the many outliers in our dataset. The median of the Fare column was 13362 rupees, which was much lower than the mean which was about 22920 rupees. The median is more accurate in this case as the measure of central tendency because it is robust to outliers. The median Fare was used as the comparison feature for all other features going forward.

First, EDA was performed on the numerical/continuous features to convey their relationship, if any, with airfare. After conducting a seaborn pairplot using scatter plots it was understood that there was no correlation between any of the numerical data, Duration_in_hours, Day, and Days_left. Therefore the focus was shifted to categorical features against Fare. Next, EDA was performed on the Class column amongst different Airlines against Fare. This conveyed that only two airlines, Vistara and Air India, had availability in premium seats such as Premium Economy, Business, and First class. According to sundaygaurdianlive.com, Vistara and Air India share the same parent company (Tata Group). These airlines both offer a luxurious experience which explains their higher airfare and why they may offer more availability in the premium seats. Due to this, it was decided that all the values that were not considered economy be combined into one value, Premium. This now conveyed that the most expensive flights were those in the Premium classes, shown in Figure 2. Then, the destination and source columns were combined to create a route column, a minimal form of dimension reduction. A Booking_category feature was also created to replace the Days_left, which is a feature that conveys the amount of days left from booking until departure. This feature has three values "Last-minute", "Short-notice", and "Planned" and is used in future analysis. The last column that was created was Part of the month, conveying what time of the month the flight would take place, the values were Beginning, Middle, and End. This was the last form of data cleaning that was done to this dataset.

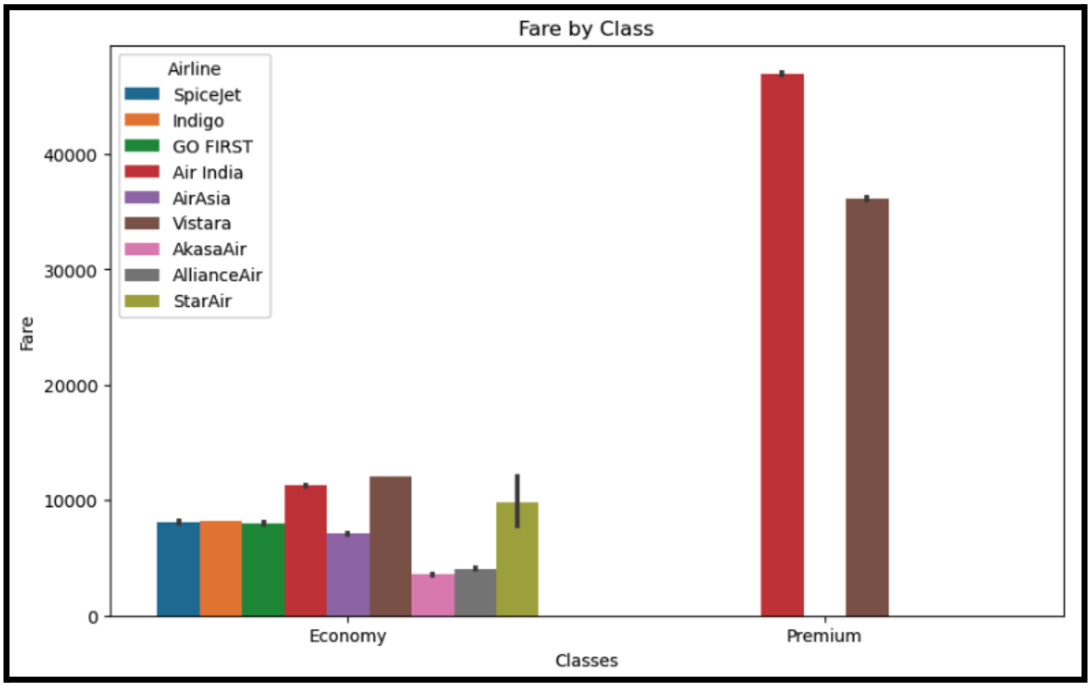
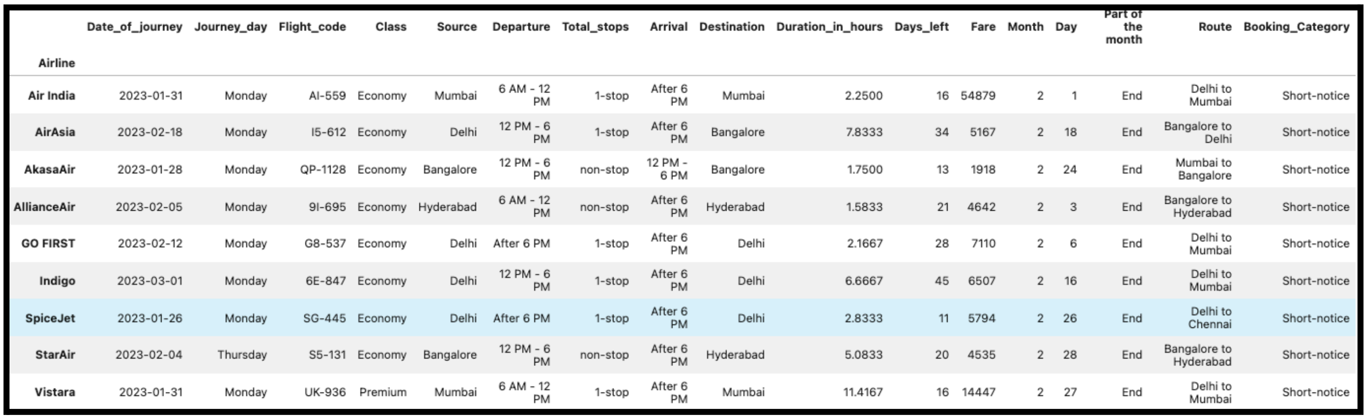


Figure 2: Bar plot created using seaborn, conveys classes across different airlines and how those flights compare in fare

Feature Analysis

After the creation of the new columns and conversion of those numerical categories into more useful categorical features for the model that was built, an in depth EDA was performed to better understand the information the data had to convey. In this part of the EDA there is a first glance at our new and improved dataset, Table 1 conveys the highest frequency values for each category organized by Airline. At first glance the highest frequency of the data grouped by airline conveys a load of important information. Such that the most common day of the week to fly is Mondays for almost every airline, this is most likely because it is the beginning of the work week and people may be more hesitant to fly on this day because they do not want to use another PTO day. The most frequent Class to fly in is economy because it is the most common and cheapest class to buy a seat in; although this is not true for Vistara which is a luxury airline and their most frequent tickets are in the premium class. It is also clear that the highest frequency for almost all airline arrival flights are after 6 PM with one after noon, which can be due to the fact that getting to destinations takes about a whole day of travel especially if there are any connecting flights. On the other hand, it is easy to tell that the most popular times for departure flights are between noon and midnight, probably because people may want to sleep in before a flight takes off or maybe pack up and check out of the hotel if ending a vacation or alternatively may wait to leave until after a day of work for their vacation. The most common number of stops is one stop for most airlines with a few nonstop flights for the smaller airlines which may have limited availability in destinations. Duration in flights will always vary depending on how many layovers there are, how long layovers are, or how far a distance they may be traveling. Akasa has the lowest fare in terms of frequency and AirIndia has the highest fare. We have limited range on the month column, the only whole month we have is February, therefore it makes sense that February is the most frequent month in all the airlines. The most time of the month to fly is the end of the month for all airlines, most likely due to end of the month paychecks, work and vacation cycles, etc.The most popular routes by airline are from Delhi to Mumbai, these are some of the most populated cities in India therefore they are hotspots for flights.

Table 1: Highest frequency values amongst each feature in the dataset compared amongst airlines



	Date_of_journey	Journey_day	Flight_code	Class	Source	Departure	Total_stops	Arrival	Destination	Duration_in_hours	Days_left	Fare	Month	Day	Part of the month	Route	Booking_Category
Airline																	
Air India	2023-01-31	Monday	AI-559	Economy	Mumbai	6 AM - 12 PM	1-stop	After 6 PM	Mumbai	2.2500	16	54879	2	1	End	Delhi to Mumbai	Short-notice
AirAsia	2023-02-18	Monday	I5-612	Economy	Delhi	12 PM - 6 PM	1-stop	After 6 PM	Bangalore	7.8333	34	5167	2	18	End	Bangalore to Delhi	Short-notice
AkasaAir	2023-01-28	Monday	QP-1128	Economy	Bangalore	12 PM - 6 PM	non-stop	12 PM - 6 PM	Bangalore	1.7500	13	1918	2	24	End	Mumbai to Bangalore	Short-notice
AllianceAir	2023-02-05	Monday	9I-695	Economy	Hyderabad	6 AM - 12 PM	non-stop	After 6 PM	Hyderabad	1.5833	21	4642	2	3	End	Bangalore to Hyderabad	Short-notice
GO FIRST	2023-02-12	Monday	G8-537	Economy	Delhi	After 6 PM	1-stop	After 6 PM	Delhi	2.1667	28	7110	2	6	End	Delhi to Mumbai	Short-notice
Indigo	2023-03-01	Monday	6E-847	Economy	Delhi	12 PM - 6 PM	1-stop	After 6 PM	Delhi	6.6667	45	6507	2	16	End	Delhi to Mumbai	Short-notice
SpiceJet	2023-01-26	Monday	SG-445	Economy	Delhi	After 6 PM	1-stop	After 6 PM	Delhi	2.8333	11	5794	2	26	End	Delhi to Chennai	Short-notice
StarAir	2023-02-04	Thursday	S5-131	Economy	Bangalore	12 PM - 6 PM	non-stop	After 6 PM	Hyderabad	5.0833	20	4535	2	28	End	Bangalore to Hyderabad	Short-notice
Vistara	2023-01-31	Monday	UK-936	Premium	Mumbai	6 AM - 12 PM	1-stop	After 6 PM	Mumbai	11.4167	16	14447	2	27	End	Delhi to Mumbai	Short-notice

Once we identified which features of the dataset and fare values appeared to have the greatest and least impact, we used the median to explore how these features interact and specifically influence Fare. First, the median fare for each route was calculated and sorted from most to least expensive. The most expensive route was Ahmedabad to Mumbai, with a median fare of 18,712 rupees, while the cheapest route was Bangalore to Delhi, with a median fare of 10,338 rupees. This aligns with our frequency table, which shows that AirAsia—a budget airline—frequently operates flights on the Bangalore-to-Delhi route, making it an economical choice. AirAsia could be worth noting as a key player in the budget airline sector for further analysis. Next, we examined the relationship between Fare and booking category using median values. The analysis revealed that last-minute flights were the most expensive, while planned flights were the cheapest. As shown in Table 1, short-notice bookings were the most common category, likely driven by business-related travel (e.g., conferences, meetings), revised vacation plans, and emergencies. Using this same method, it was discovered that the most expensive day to travel was Sunday and the cheapest Thursday. This is possibly because people are trying to come home before the work week starts therefore a higher price can try to persuade them to fly another day with more available seats at a cheaper price. The cheapest time to fly is any time before 6 AM for departure and arrival with a median price for both at about 9800 rupees, but the most expensive departure time is 6 AM to Noon with about 14200 rupees and about the same price for arrival flights after 6 PM. These times are probably due to sleeping convenience and work schedule. Flights with one or more stops are more expensive than non-stop flights, because multiple planes means more jet fuel and more employees to pay therefore it costs more. The end of the month is more expensive to travel compared to the beginning of the month, possibly because it is also in high demand because of business travel, payday travel, holiday travel, etc.. To conclude this analysis, the median values revealed key insights about how route, booking category, travel day, time of departure, and flight type influence airfare. These findings provide a foundation for further exploration of the data through visualizations, which will offer a clearer understanding of these patterns and trends.

Trend Analysis and Visualizations

To better understand the relationships and distributions within the dataset, we visualized key features to identify patterns and trends. The violin plot below in Figure 3 illustrates the distribution of fares across different flight classes, revealing a significant disparity in ticket pricing between Economy and Premium classes. Premium fares exhibit a wider range and higher median, while Economy fares remain more concentrated around lower values. This suggests a stark difference in pricing structure, which may further influence passenger choices and airline profitability strategies. Figure 4 compares

Airlines and their median Fare. It can be concluded from this bar plot that Vistara and Air India have the most expensive airfare. This is due to the luxury travel experience and more availability of premium tickets. Akasa Air, Alliance Air, Star Air, and Air Asia have the lowest airfare prices, these are most likely budget airlines and may have limited destinations, stops, etc. Using this knowledge, key routes that were identified earlier in the EDA phase were compared amongst airlines using median Fare, this analysis also conveyed that the most expensive flights were on Vistara and AirIndia as well as the cheapest flights were on Akasa Air. These two airlines could be key features in influencing our model.

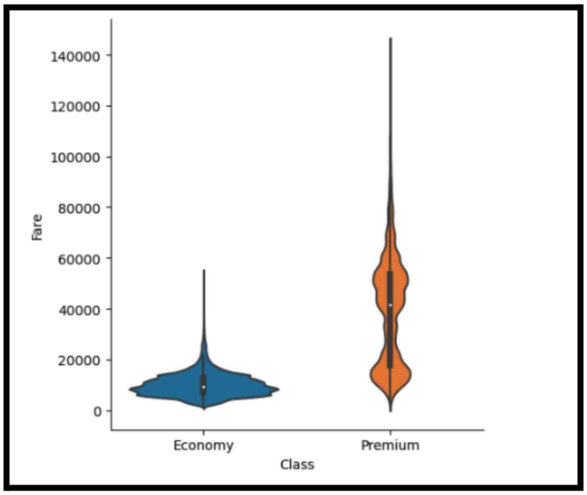


Figure 3: Seaborn violin plot of flight classes against airfare, conveying a large difference in price and variability between the two class categories

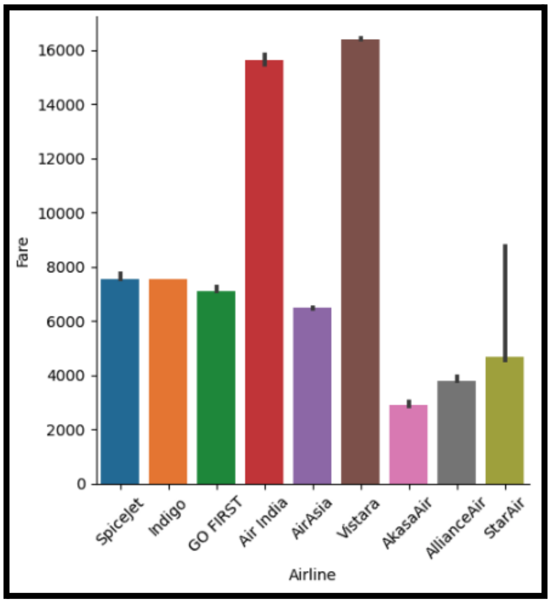


Figure 4: Seaborn barplot uses median fare to compare airlines against one another, portraying Vistara and Air India as the most expensive airlines and Akasa Air as the cheapest airline

The EDA process uncovered key relationships and patterns in the airfare dataset. Features such as Class, Route, Airline, Journey_day, Departure, Arrival, and Total_stops were identified as significant drivers of Fare. Notably, last-minute bookings were found to have significantly higher fares than planned ones. The Fare distribution exhibited right-skewness with a few extreme outliers, reflecting variability influenced by factors like seasonality, demand peaks, booking timing, and flight types (e.g., nonstop vs. connecting flights). Newly created features, such as Route and Part_of_the_month, alongside existing ones like Airline, Journey_day, and Booking_Category, show strong potential for predictive modeling. The next steps will focus on leveraging these insights through feature engineering and preparing the data for modeling.

Preprocessing

Previously, the data was checked through the performance of preliminary assessments of data quality. It was determined that predicting the Fare was our primary aim. Records were thrown away with missing price data, but not before making the most of the other available data to look for any patterns between the airlines. None were found and decided to treat all airlines equally; the year label didn't seem to be particularly useful, as well as numerical features. We converted those numerical features into useful categorical features and then created dummy variables. This is where machine learning models will start to be built. Before even starting with learning a machine learning model, however, it was considered how useful the median value is as a predictor. The first model is a baseline performance comparator for any subsequent model. Then, build up the process of efficiently and robustly creating and assessing models against it.

Encoding

The shape of the data is now (445366, 18) with 5 more columns than it originated with. The columns that the data now has are:

Date_of_journey	object
Journey_day	object
Airline	object
Flight_code	object
Class	object
Source	object
Departure	object
Total_stops	object
Arrival	object
Destination	object
Duration_in_hours	float64
Days_left	int64
Fare	int64
Month	int64
Day	int64
Part of the month	object
Route	object
Booking_Category	object

With this knowledge, the object features would be extracted and then converted to encoded variables using the pandas pd.get_dummies() function. The object columns Source, Destination, Month, and Day were excluded from the creation of encoded variables because they were substituted with the Part of the month and Route columns. It was also decided to use the drop_first argument to drop one of the encoded variables for each categorical feature to avoid multicollinearity, which could cause issues down the line for the linear regression model. It also minimized the amount of dimensions that the model dealt with later. Once we encoded the chosen categorical features, the shape of the new database was (445366, 68).

Train/Test Split

To ensure the reliability and generalizability of our airfare prediction model, the dataset was divided into training and testing sets using a 70/30 split. This means that 70% of the data was allocated for training the model, allowing it to learn patterns and relationships within the dataset, while the remaining 30% was reserved for testing its performance on unseen data. By using this split, we can evaluate how well the model predicts airfares on data it has not encountered before, simulating real-world scenarios. This approach helps prevent overfitting and ensures that the model performs effectively in production, providing accurate airfare predictions for future travelers. The X variable in this function will include all the encoded variables, which means going forward there will be no need for scaling since all the features are binary values. The target variable is the Fare column in the updated dataset including all categorical and numerical features. This methodology not only ensures a robust evaluation of the model's predictive accuracy but also establishes a solid foundation for creating a reliable and scalable airfare prediction system that can adapt to diverse scenarios in real-world applications.

Preliminary Modeling

Before delving into the detailed modeling and prediction phase of the project, a dummy regressor was employed to establish a baseline performance metric using the median value. The median value prior to using the dummy regressor is 13379 rupees. After when the dummy regressor was applied the value of the regressor was the exact same 13379 rupees. The median predicted by the dummy regressor was exactly the same as the true median of the target values, this indicates that the baseline model is effectively capturing the central tendency of the data. To assess how closely the dummy regressor matches or explains the actual values, evaluation metrics such as R-squared, Mean Absolute Error (MAE), and Mean Square Error (MSE) were used. Using the dummy regressor median value, the r-squared value calculated on the training data and test data was (-0.2195, -0.2172), which is not good! It suggests that the model is performing very poorly and actually makes predictions worse than simply using the median value of the target variable for all predictions. R-squared is negative because the dummy regressor was not capturing any useful relationships in the data. It was essentially performing worse than the baseline mean prediction, which is why the R-squared value is negative. It suggests that the data might be extremely noisy, or the relationships between the features and target variable are too complex for such a simple model. It also indicates that your model might need more sophisticated features or transformations to improve performance. The next evaluated metric was MAE. It is arguably the most intuitive of all the metrics, this essentially states, on average, you might expect to be off by around (14289, 14273) rupees if the ticket price was an estimation based on an average of known values. Lastly, the mean square error was calculated as a performance metric, the value it gave on the training data was (506460343, 508088827) rupees, which is extremely high. The MSE is very high this is probably due to the outliers in the data, we will have to rely on more robust metrics for this model such as MAE. Since the data has many large values, the root mean squared error was calculated on the train and test set which was (22505, 22541); the results of the RMSE still convey a large amount of error. One good insight that can be concluded from these metrics is that the train and test set metric values are very similar. This means the model generalizes well. The model is performing consistently on both the training data and the test data. This indicates that the model has learned the underlying patterns in the data, rather than memorizing or overfitting to the specific training set. In conclusion, while the dummy regressor model provides a useful baseline, the evaluation metrics highlight the limitations of such a simplistic approach. These results underscore the complexity of the data and indicate that more advanced models, with additional features and hyperparameter tuning, will be necessary to capture the underlying patterns and improve predictive accuracy. Moving forward, a more sophisticated model will be crucial to better understand the relationships within the data and enhance performance.

Modeling

Linear Regression Model

Linear regression is often used as a starting point in predictive modeling because of its simplicity, interpretability, and efficiency. If the results are unsatisfactory, more complex models will be explored. Using the data from the train/test split function from earlier, the linear regression model, with no specified hyperparameters, was fit and predicted. Using this very simple model, performance metrics were applied, starting with R-squared which was (0.5595, 0.5593) for the train and test set. The simple linear regression model explains about 56% of the variance on the train and test set. This suggests our model is effective. The MAE for this model was approximately (9424, 9431) rupees. Using this model, then, on average you'd expect to estimate airfare within 9430 rupees or so of the real price. This is not the best prediction, maybe with more hyperparameter tuning or even a different model we can get a better prediction. To try and improve the model, the hyperparameter K was tuned to be K=15 and the simple imputer used the median as its metric of central tendency, to evaluate if a slightly larger K value would make the model perform better. This resulted in larger error values with R-squared equal to 0.54. To make the process more efficient in finding the best hyperparameters, specifically K, a GridSearch Cross Validation was performed. Grid search was chosen due to the evaluation of all hyperparameter combinations in a deterministic way, it is more reproducible. Running the same grid search multiple times on the same data will yield the exact same results. Grid search determined that the best K value for this model was 68, which is all the dimensions of the model. Figure 5 conveys the best k value determined by the grid search cross validation (CV) method using the R-squared cross validation score. The CV score conveys that with all 68 features the model only explains 56% of the variation. Using this information given, the linear regression coefficients were given for the most contributional features. These results suggest that the premium class is the biggest positive feature, meaning it increases the price. This makes intuitive sense and is consistent with what we saw during the EDA work. Also, you see the Routes from Kolkata have a strong positive influence as well. Non-stop flights are negatively associated with airfare, meaning it contributes to making fare decrease. This seems accurate. People will pay less for non-stop flights; people want the least amount of travel time possible to get to their destination faster. Overall, this model has decent performance. There are other models that are more well-suited for complex datasets with more hyperparameter tuning abilities.

Gradient Boosting Regressor

The data was then tested on a different type of model, Gradient Boosting Regressor. It may be more optimal for the dataset because it is better at handling complex datasets such as this one, 68 dimensions, and it is robust to outliers, which this dataset is abundant in. The pipeline that was created consisted of the SimpleImputer using the median and the Gradient Boosting Regressor model. With 15 cross validations, the mean cross validation R-squared of the test set values was 0.60. This means that this model can explain 60% of the variation from the data. Although this model did better than the simple linear regression model from before, it can do even better with some hyperparameter tuning. This time around Random Search Cross Validation was performed, due to the search space being very large. Random search is generally more efficient because it samples hyperparameters randomly and has a better chance of finding near-optimal solutions in less time. A large focus of this project was to reduce computational time with the model and cross validation because there are so many dimensions in the dataset. Once the best hyperparameters were found through the random search they were then applied to the gradient boosting regressor with a cross validation fold of three. The mean r-squared value of these three folds was 0.66, meaning that 66% of the variance can be explained by the model. The MAE for this model is 8151 for the train set and 8176 for the test set, meaning that on average, the model's predictions deviate from the actual Fare value by approximately 8160 rupees. Figure 6 below conveys the most important and most contributional features of the gradient boosting regressor model. Those features were similar to those of the simple linear regression model. The features are: Class_Premium, Total_stops_non-stop, Airline_Vistara, Booking_Category_Planned, etc. In conclusion, the Gradient Boosting Regressor demonstrated a significant improvement over the previous model, effectively handling the dataset's complexity and variance, and highlighting key features that contribute most to predicting airfare. With further fine-tuning, this model holds great potential for even better performance and deeper insights.

Histogram Gradient Boosting Regressor

After further research it was decided to also try the histogram gradient boosting regressor. It can greatly reduce computation time and works well with high dimensional categorical datasets. Using the same pipeline as the other two models, SimpleImputer with the strategy argument set to median and then instantiating the model. The mean of the three-fold cross validation R-squared results, the basic model, with no hyperparameter tuning, was 0.64, which is better than the gradient boosting regressor simple model. With the best hyperparameters calculated using the Random Search CV, the HGBR model did only slightly better with a mean R-squared score of 0.65. This is slightly lower than the previous GBR model, although it does not mean that this is the optimal metric for this model. The objective is to get the best prediction, therefore the MAE metric was prioritized. In this model, the MAE was 7514 for the train set and 7558 for the test set. This model out-predicted the previous GBR model, therefore the HGBR model is the selected model for this project. Figure 5 below is the Shap plot of the HGB regressor model which portrays that the most impactful features to the model are: Class_Premium (positive, increasing price), Total_stops_Non-stop (negative, decreasing price), Route_Delhi_to_Kolkata, Total_stops_2+-stop.

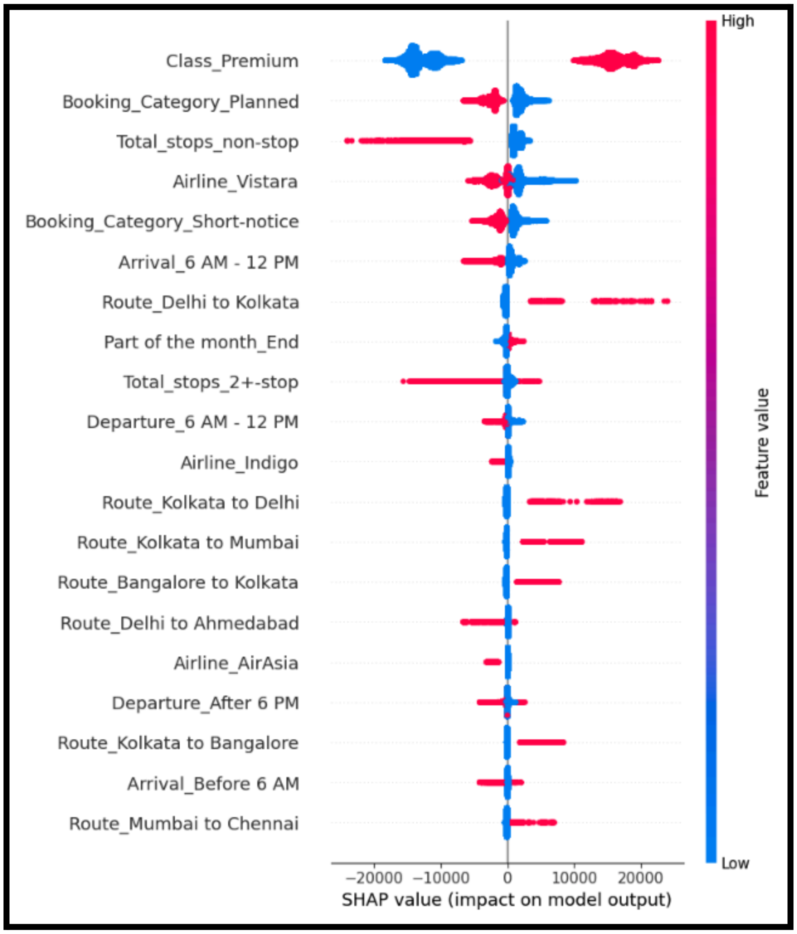


Figure 5: Shap plot conveying the most important features (both positive and negative impact) on the HGBR model

Final Model Selection

The final model that was selected was the Histogram Gradient Boosting Regressor, it had the best predictive power for airfare. Figure 6 conveys an overall good correlation between actual and predicted values. Most points cluster around the red dashed line meaning the model is reasonably effective but not perfect. There is some scatter around the red line especially as fare values increase. This indicates that the prediction errors tend to grow for higher fare values and that the model struggles more with these. For fares above 80000 rupees the model seems to underpredict. There are a few noticeable outliers where predicted fares deviate from the actual fares; this can be due to noise in the data and model limitations.

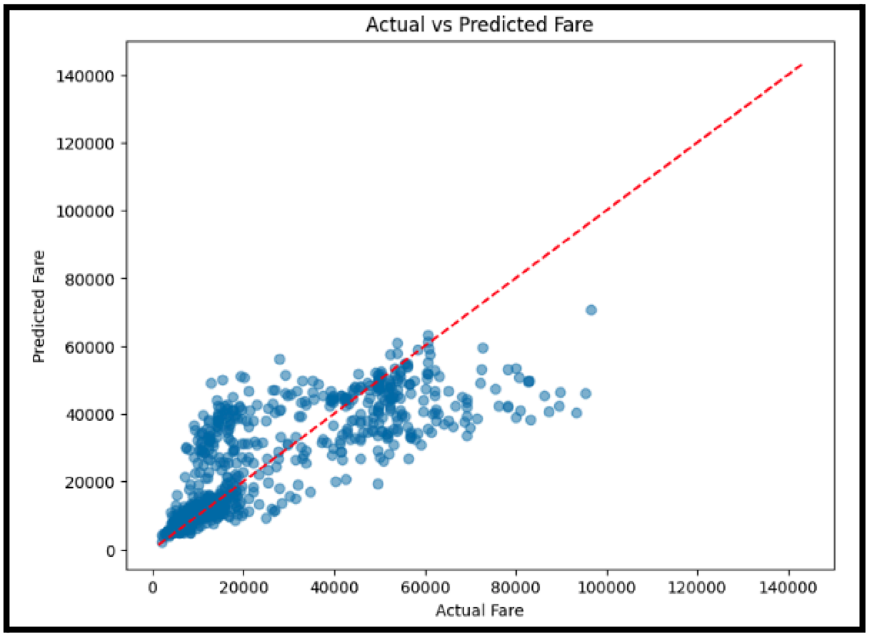


Figure 6: Scatter plot of Actual Fare vs Predicted Fares from HGBR model

Conclusion

The airfare predictor model using Histogram Gradient Boosting Regressor, achieves an MAE of about plus or minus 7520 rupees, it means fare predictions will deviate by that amount, affecting pricing accuracy. Reducing MAE ensures customers are shown accurate prices, improving trust and booking rates. 65% of fare variance is explained by the model, showing its ability to capture pricing trends and market behavior. This enables better competitive pricing strategies. Identifying top features like "Class_Premium" or "Airline_Vistara" helps target premium customers or optimize deals for high-performing routes, boosting revenue. As well as, consistent train/test metrics (e.g., MAE, R-squared) indicate a robust model, minimizing the risk of pricing errors during high-demand periods like holidays. With better fare accuracy, users are more likely to return to the platform for future bookings, increasing Customer Lifetime Value (CLV). Companies such as Hopper, Google Flights, and Expedia can predict airfare drops for specific routes but the model has a relatively high MAE and a moderate R-squared value which could cause customers to lose trust in the platform resulting in fewer bookings due to perceived unreliability or poor optimization of demand-supply dynamics. There are some areas that could be tuned in order to improve the model such as more feature engineering (specifically using an OrdinalEncoder), investigating whether important features are missing or underrepresented, experimenting with other models such as XGBoost or LightGBM, and/or maybe exploring log transform to reduce the impact of large values and improve the fit for higher ranges. In conclusion, while the Histogram Gradient Boosting Regressor provides a solid foundation for airfare prediction with reasonable accuracy and actionable insights, further enhancements through advanced feature engineering, alternative models, and targeted preprocessing could unlock even greater predictive power, enabling businesses to deliver more reliable pricing strategies and enhance customer trust.