

Assignment 3

ML Data Product

10-11-2023

Group 12

Student Last Name	Student First Name	Student ID	Group Allocation
Shah	Rushab	14351985	Student A
Lim	Simon	24661225	Student B
Siarivita	Joanne	24714673	Student C
Amin	Naeer	14203071	Student D

Github	Project Repo: https://github.com/naeer/at3_advanced_ml
--------	---

36120 - Advanced Machine Learning Application
Master of Data Science and Innovation
University of Technology of Sydney

Table of Contents

1. Executive Summary	2
2. Business Understanding	3
a. Business Use Cases	3
b. Key Objectives	3
3. Data Understanding	4
4. Data Preparation	10
5. Modelling	12
a. Minimum Fare	12
b. Median Fare	12
c. Mean Fare	13
d. Modal Fare	14
6. Evaluation	17
a. Evaluation Metrics	17
b. Results and Analysis	17
c. Business Impact and Benefits.	18
d. Data Privacy and Ethical Concerns	19
7. Deployment	20
a. Model Serving	20
b. Web App	21
8. Collaboration	22
a. Individual Contributions	22
b. Group Dynamic	23
c. Ways of Working Together	23
d. Issues Faced	24
9. Conclusion	25
10. References	26
11. Appendix	27



1. Executive Summary

Airlines record millions of flight information in order to guide competitive pricing, plan new strategies for marketing and optimise flight schedules and revenue streams. Furthermore, flight information such as flight fare and flight duration can be helpful for travellers to estimate costs for different dates and routes and manage their schedules. In this regard, our data scientists team has been requested to build a product that helps users in the USA to better estimate their local travel airfare. The main objective of the project is to build an application that can predict the expected flight fare for different airport trips. In particular, this project split the travel airfare into 4 different fares, including minimum, median, modal and mean of airfare. Together, these metrics aim to provide a comprehensive overview of potential fares to users. The minimum fare highlights the best case, while the median and mean demonstrate more typical costs users can expect. Also, the mode shows the most common price point among all options. Since this project aims at building user-friendly applications for users, a minimum of features were used as predictor variables, including airport departure and destination, date, days from flight, cabin type, time category and day of week. The performance of models is mainly based on RMSE scores, measuring the average difference between predicted values and actual values. The project conducts four different experiments (one experiment per a data scientist) to predict median, minimum, modal and mean of travel airfare. Prior to training models, the process of data preparation, feature engineering, EDA, data transformation and data pipeline have been performed. Models for each airfare were successfully established with a moderate performance. To enable model's access to users, Docker and Gradio were used for model deployment, enabling users to use the models as a application.





2. Business Understanding

a. Business Use Cases

This project focuses on developing a machine learning model to predict airfare prices for flights within the United States. The ability to estimate ticket prices is extremely valuable in the travel industry across various business use cases.

Travellers can leverage the app when planning vacations to estimate costs for different dates and routes, enabling selection of lower-priced options. Airlines use fare data to guide competitive pricing decisions and inventory management. Online travel agencies can also integrate the prediction API to recommend optimal booking dates and times to users.

However, accurately forecasting flight fares is challenging due to complex factors like demand fluctuations, variable airline and flight routes, competitive pricing strategies, and more. Traditional rules-based approaches fail to account for these dynamics. Machine learning algorithms can unlock deeper insights from historical fare data to uncover patterns linking trip details to ticket prices.

b. Key Objectives

The primary goal of this project is to develop a machine learning model to accurately predict airfares for non-stop flights based on trip parameters like origin, destination, travel dates, and cabin class. Since the model does not forecast fares for specific airlines, we utilize multiple statistical metrics to provide robust fare estimates:

- Minimum Fare: To represent the cheapest available fare across airlines
- Median Fare: To indicate the middle value, reducing the impact of outliers
- Mean Fare: To reflect the average fare across airlines
- Modal Fare: To capture the most frequently occurring fare

Together, these metrics aim to provide a comprehensive overview of potential fares to users. The minimum fare highlights the best case, while the median and mean demonstrate more typical costs users can expect. The mode shows the most common price point among all options. By blending multiple indicators, we mitigate the lack of airline-specific pricing and deliver actionable fare estimates that users can trust for budgeting and planning.



3. Data Understanding

The dataset used in this project contains important information related to flight bookings, including details such as flight date, starting and destination airports, and total fare. The data was sourced from Expedia and provided a comprehensive view of flight bookings.

The dataset consisted of a lot of diverse features. Some notable ones are listed below:

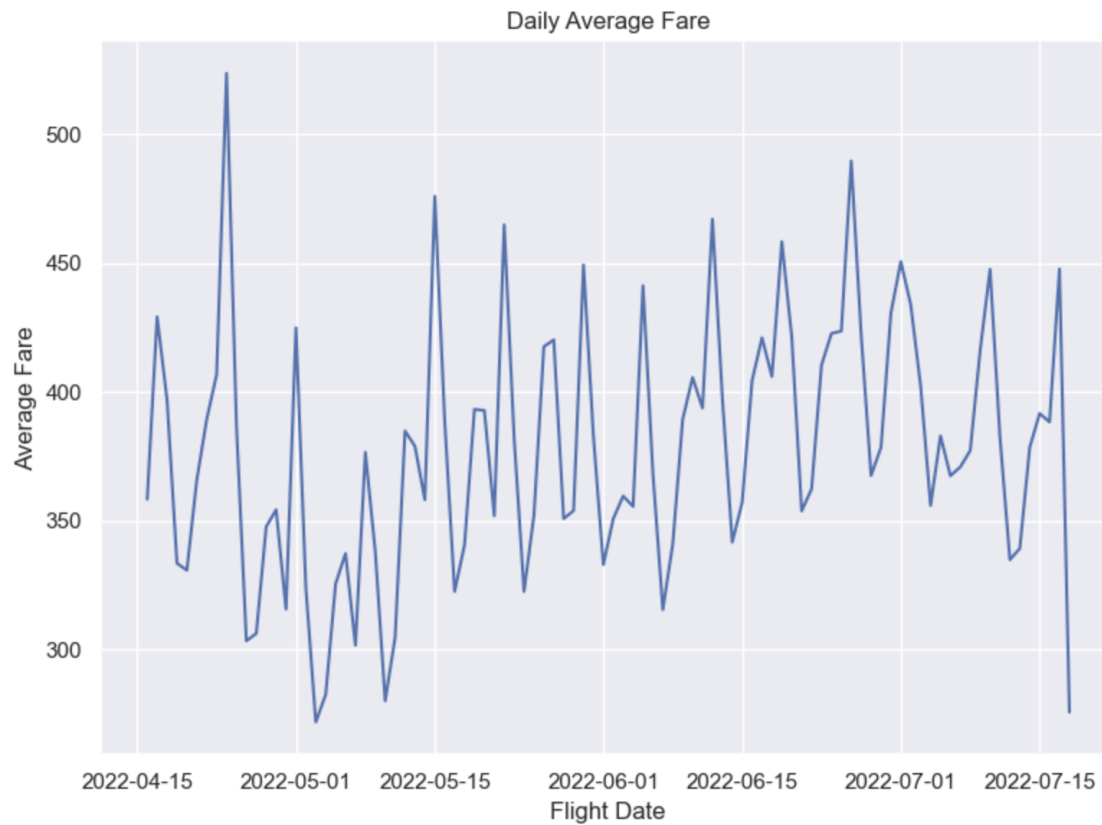
- Flight details such as flightDate (date of the flight), startingAirport (Airport from which the flight is starting), destinationAirport (Final destination of the flight).
- Fare Information such as totalFare (the total fare of the flight including taxes).
- Time Information such as segmentsDepartureRaw (departure time of each segment of the flight in ISO 8601 format)
- Airline and flight segment details such as segmentsAirlineName (name of the airlines for each segment) and segmentsDistance (distance travelled for each segment).

While the dataset provided very detailed information about the flight bookings, not all features were used in the modelling process, which was tailored to accommodate the end user's limited input. The end user is expected to provide specific details like the departure and destination airports, departure date, departure time, and cabin type. As a result, only these user-provided features and the features derived from them were used in the data modelling phase.

The full data dictionary of the dataset used in this project can be found in the [Appendix](#).

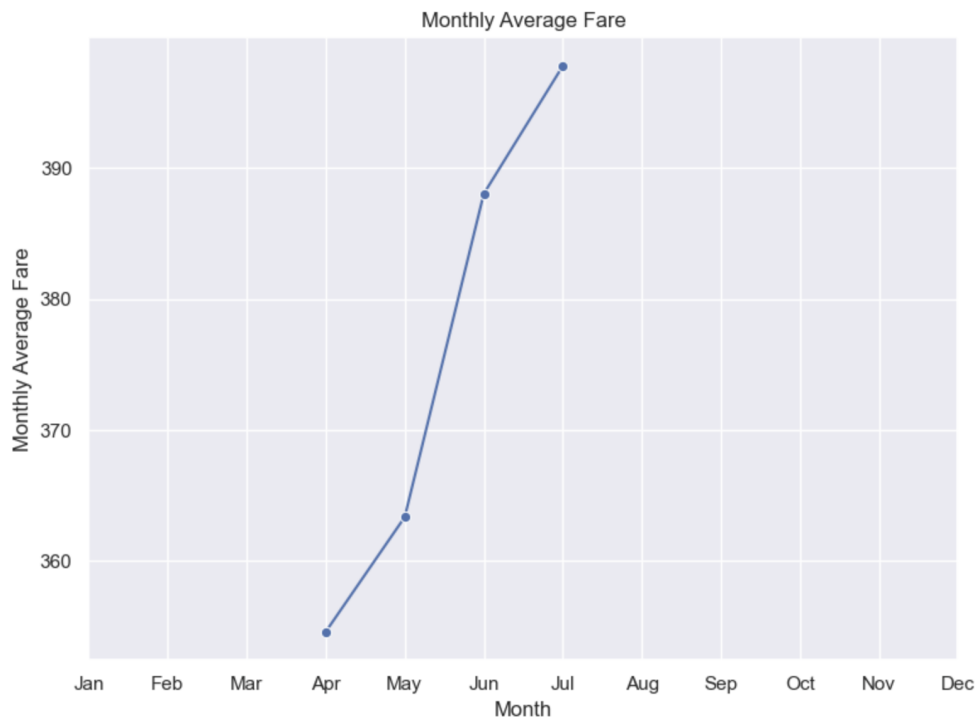
Exploratory Data Analysis

Exploratory data analysis was carried out on the dataset to find useful insights.

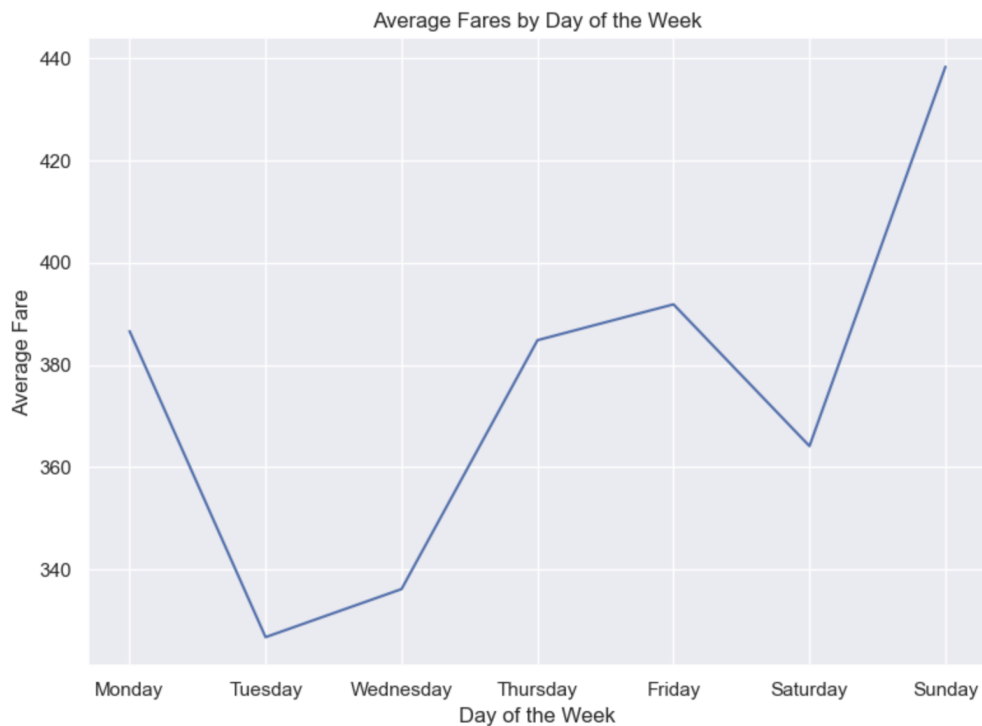


As can be seen from the above plot, the average airfare had a daily trend, i.e., for certain days the fare went down and for others, it went up. However, as the dataset only had data from 16th April 2022 to 18th July 2022, it did not show any seasonality or holiday trends.

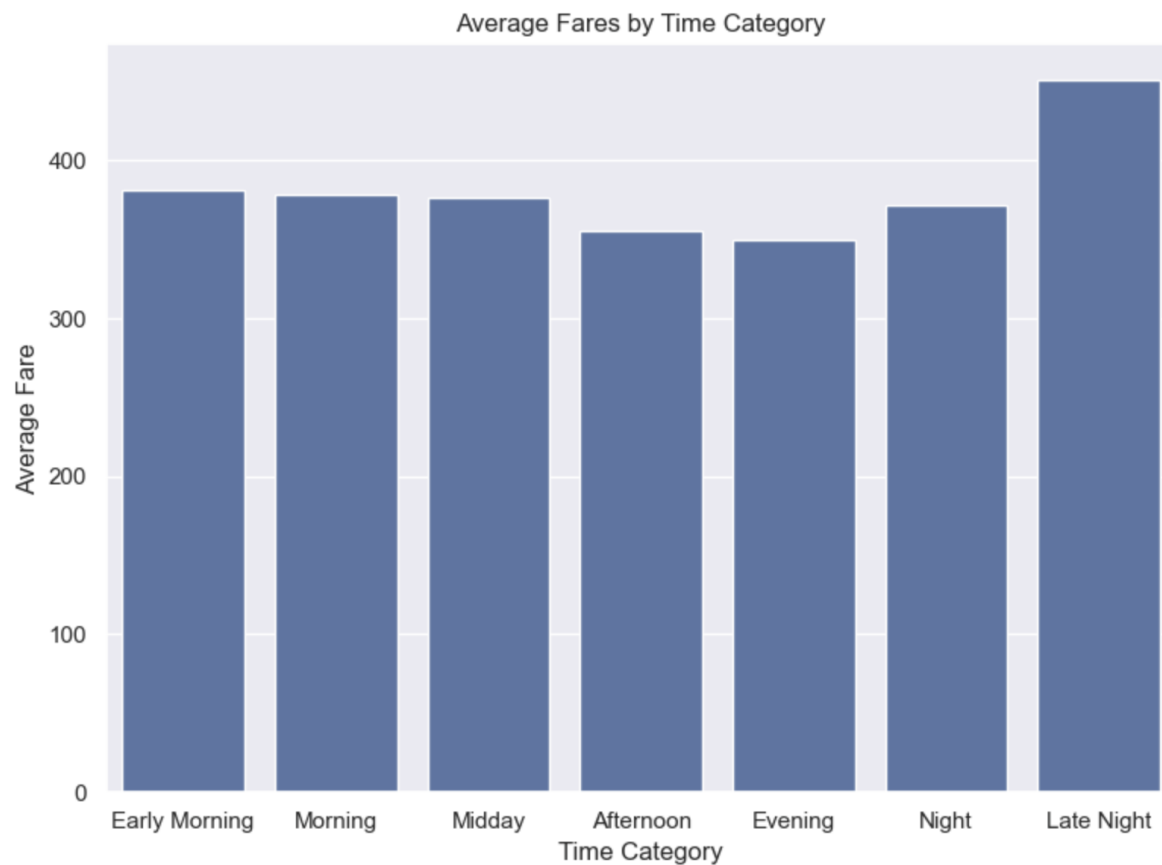
The monthly average airfare had an upward trend from April 2022 to July 2022, as shown by the plot on the next page.



In terms of average airfare on each day of the week, Tuesdays and Wednesdays had the cheapest airfares, whereas Sundays had the most expensive ones. This is shown by the following plot:

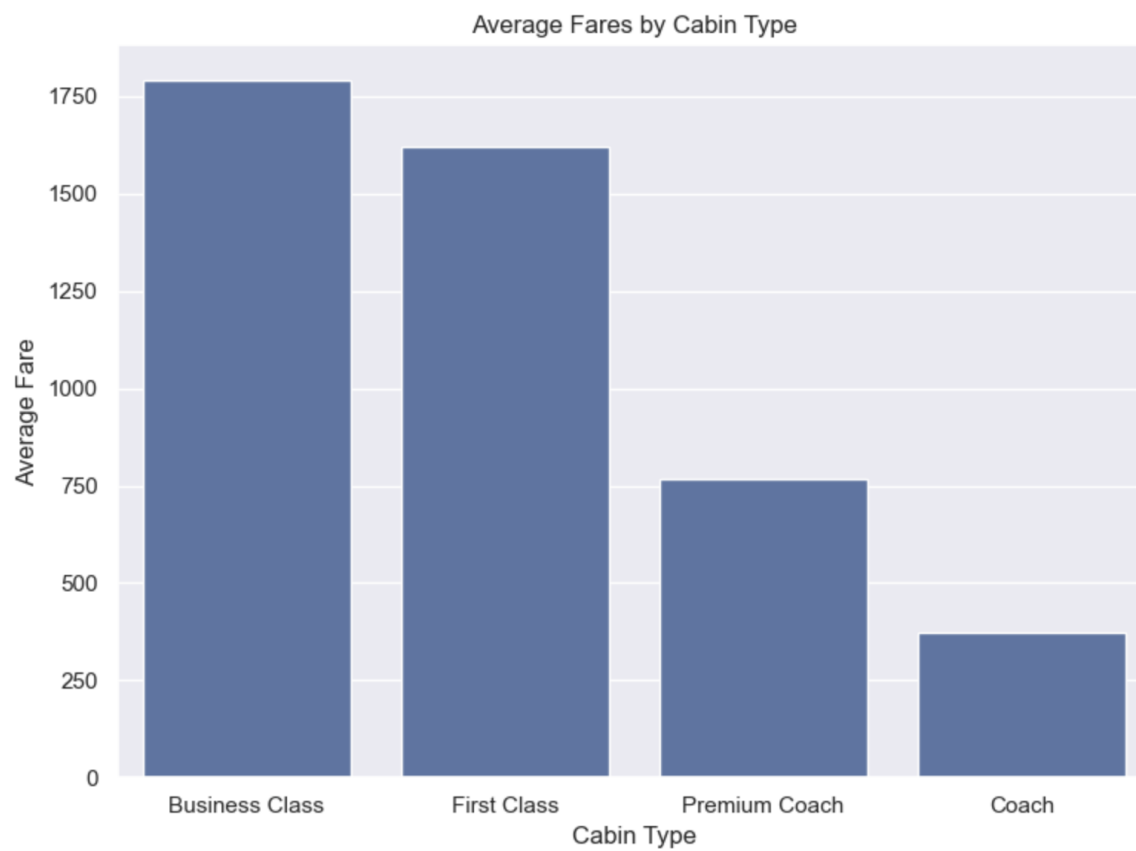


The following plot shows the average airfares by each time category.

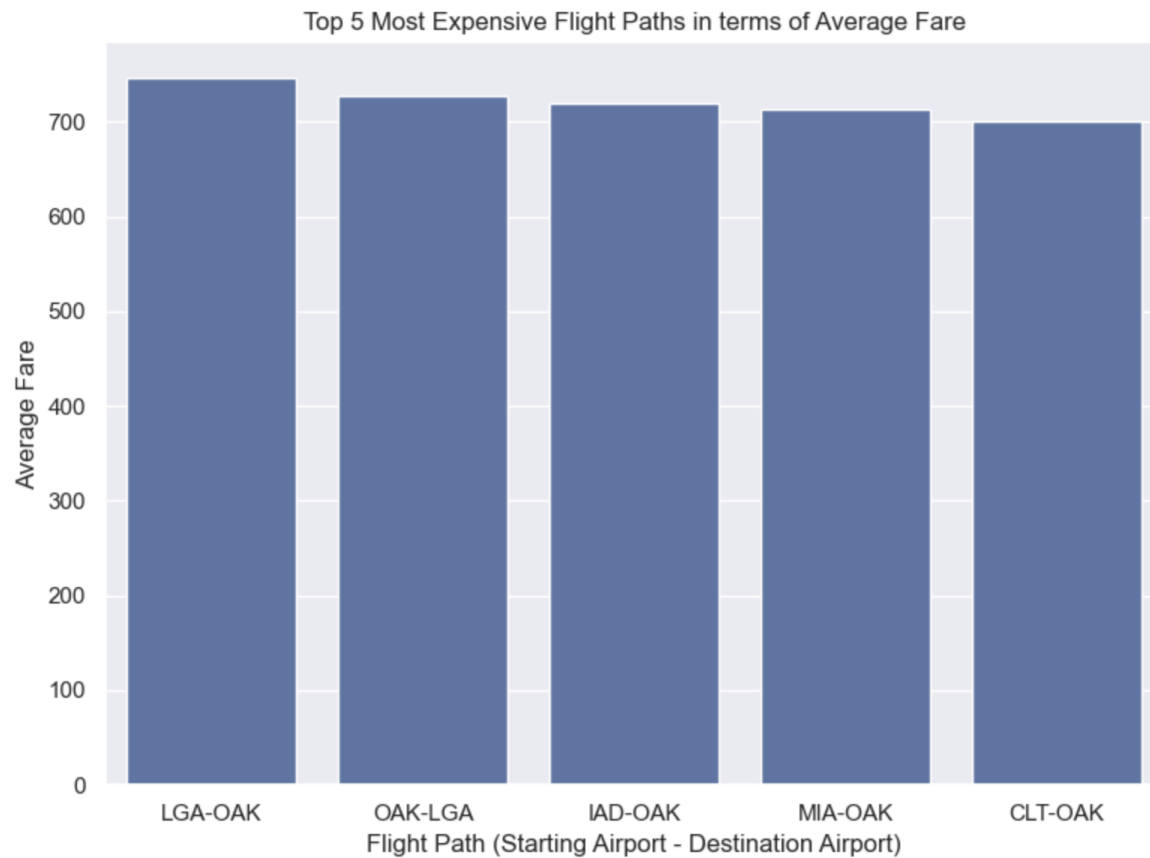


As can be seen from the plot, air ticket prices were the highest during late night (11pm - 2am) and the cheapest during afternoon (2pm - 5pm) and evening (5pm - 8pm).

The plot on the next page shows the average fares by cabin type.



As expected, Business Class and First Class had the highest ticket prices, whereas Coach had the cheapest ones.



The above plot shows the top 5 most expensive flight paths in terms of average fare. As expected, these flights were the ones with the longest travel distance as they were all cross-country trips. For example, LGA (LaGuardia Airport) to OAK (Oakland International Airport) was a east coast to west coast trip.



4. Data Preparation

In the process of preparing our dataset for modelling, we took several deliberate steps to refine our data and create meaningful features. Here is a summary of what was done:

1. Filtering Non Stop Flights:

We began by filtering our dataset to focus exclusively on non-stop flights, thereby excluding any flights with multiple stopovers. This step was crucial to ensure consistency and accuracy in our analysis of direct flight patterns.

2. Converting Departure Times:

The raw departure times were converted into a more informative feature, categorizing the departure into distinct time categories. These categories are based on the hour of the day and provide a clear framework for segmenting flights into time slots that reflect typical daily routines.

The time categories were defined as follows:


Hour Range	Time Category
5:00 - 8:00	Early Morning
8:00 - 11:00	Morning
11:00 - 14:00	Midday
14:00 - 17:00	Afternoon
17:00 - 20:00	Evening
20:00 - 23:00	Night
23:00 - 2:00	Late Night

It's important to note that flights up to 2:00 AM were considered to be part of the previous day, effectively categorising them as 'Late Night' flights.

3. Lead Time Calculation:

We calculated the difference between the search date and the departure date as 'days from the flight'. This represents the lead time or the time before booking, which is a significant factor in flight pricing strategies and demand forecasting.

4. Summary Metrics:



We derived summary metrics for the data, which included the minimum, mean, median, and mode for the flight fares. These metrics were computed with respect to input features such as airports, cabin type, and the newly created datetime features. These summary statistics are instrumental in understanding the central tendencies and dispersion of flight prices.

5. Data Splitting for Model Training:

For the purpose of model training and testing, we split the datasets based on the date 6-17-2022. The data from before this date was used to create a training set, while the data following this date formed the testing set. This resulted in a training set that comprised approximately 80% of the total data. This consistent split ratio was maintained across all modelling experiments to ensure comparability and reliability of the results.

These steps contributed to creating robust datasets, ready for various modelling experiments.



5. Modelling

In the evaluation of three machine learning algorithms—CatBoost, XGBoost, and Random Forest—applied to a specific task, CatBoost emerges as the top performer with an R2 of 0.69, indicating a superior fit to the data. It also exhibits lower RMSE and competitive MAPE values compared to XGBoost and Random Forest. While XGBoost closely follows CatBoost in performance metrics, Random Forest lags behind with lower R2 and higher RMSE and MAPE values. Notably, all three algorithms share similar MAE values. The results suggest that CatBoost is the most suitable model for the task at hand, but ongoing optimization efforts may further enhance overall performance, particularly in reducing RMSE and MAPE. Stakeholders should consider these findings when making decisions related to model selection and tuning.

The methodology applied to model each summary metric has been detailed below:

a. Minimum Fare

Incorporating the features detailed in Section 3.0 Data Preparation, a group aggregation was conducted on the dataframe to calculate the minimum value of the total fare column within each group. This process yielded a new dataframe, encapsulating the minimum total fare for each group, which subsequently served as the basis for training a random forest regressor. This strategic approach not only leverages relevant data preparation techniques but also ensures that the model is trained on a representative subset, enhancing its capacity to provide accurate and meaningful predictions for non-stop flight airfares.

b. Median Fare

Initially, an automated machine learning library was applied on the dataset, using the features outlined in Section 3.0 Data Preparation. However, the extended training times for most regression models rendered this method impractical. Preliminary results indicated that algorithms such as linear regression and decision trees were unsuitable for our dataset. Consequently, alternative algorithms were evaluated, yielding the following performance metrics:

Algorithm	R2	RMSE	MAPE	MAE
Catboost	0.69	89.46	62.53	0.21
XGBoost	0.68	90.37	62.90	0.21

Random Forest	0.62	98.67	64.68	0.21
---------------	------	-------	-------	------

It is noteworthy that the Random Forest algorithm required approximately 20 hours to finalise training, whereas XGBoost and Catboost were significantly more efficient, completing the process in under one minute. Catboost performed the best, slightly outperforming XGBoost, however the goodness of fit and errors were very similar.

The hypothesis that distance and time might also affect flight prices led to the inclusion of median distance and duration as additional features. The impact of these variables was tested, and the results are as follows:

Algorithm	R2	RMSE	MAPE	MAE
Catboost with extra features	0.68	90.40	62.51	0.21

Although the coefficient of determination and error metrics were very close, they did not enhance the model's performance. Therefore, the preceding Catboost model—using fewer features—was deemed optimal. In light of the deployment constraints and the requirement for a model not tailored to any specific airline, it proved challenging to identify additional predictive features.

c. Mean Fare

To predict the average of total fares, two kinds of XGBoost Regressor models and two kinds of LightGBM Regressor models were employed, two with the default model and the other two with tuned hyperparameters.

The rationale behind choosing XGBoost Regressor model is that it is one of the optimal machine learning algorithms for regression analysis that shows very strong predictive performance for a complex (non-parametric) and a large dataset. As the model performance is crucial in regression analysis, XGBoost is optimal in predicting average airfares. Furthermore, its unique ability to handle non-linearity relationships between features and identify important features on the target variable makes more benefits beyond other algorithms.

Similarly, LightGBM (LGBM) was chosen as another algorithm to predict the average airport fare. LGBM is fast and accurate and also optimal in handling a non-parametric and large dataset. Furthermore, it grows trees in a leaf-wise manner, which allows the algorithm to focus on the leaves that contribute the most to the reduction in loss, resulting in reduced overfitting and faster convergence.

Pre-processing step was outlined in the Data Preparation section but one more feature was added into the mean fare prediction. Day of week was extracted (i.e., Monday to Friday) from the departure date column and was used as a feature. Therefore, there were 7 features in total that were used as predictor variables.

Features: 'StartingAirport', 'DestinationAirport', 'Cabin_type', 'time_category', 'day_of_week', 'date', 'days_from_flight'

Target: Mean airfare

Algorithm	R2	RMSE	MSE	MAE
LGBM Default	0.61	102.03	10416.47	70.43
LGBM Tuned	0.60	103.15	10639.69	71.84
XGBoost Default	0.61	102.65	10537.39	70.66
XGBoost Tuned	0.60	103.60	10733	71.33


Default models showed slightly a stronger performance than tuned hyperparameter models. Given that other important features, such as time and distance, were not included in feature inputs, the performance of the models was poorer than it was expected.

Furthermore, there was a lack of searching optimal hyperparameters using 'RandomSearchCV'. Since features were transformed using 'Pipeline' and 'ColumnTransformer', it was unable to use RandomSearchCV to find optimal hyperparameters for LGBM and XGBoost models. Instead, I searched for each option of hyperparameter and put an effort into including the optimal hyperparameters for models. Nevertheless, the tuned hyperparameter models for both LGBM and XGBoost showed slightly poorer performance than default models.

Overall, to improve the performance of models next time, more relevant features to the target variable would be essential, such as time and distance of flight.

d. Modal Fare

XGBoost was selected for modelling the data to predict the most occurring airfares, i.e., the modal fare. It was selected because of its remarkable capabilities in being able to deal with complex and big datasets. XGBoost is a tree ensemble learning method that uses weak learners, specifically shallow trees, in a sequential manner to make predictions. This allows each tree to



learn from its predecessor, making it an algorithm with a high predictive power and achieving better performance than other ensemble learning methods such as Random Forest.

XGBoost was perfect for this modelling due its ability in capturing complex relationships between features and identifying the most important ones. It was chosen over other boosting techniques such as Gradient Boost as it has regularisation parameters that helps reduce overfitting to the data.

Apart from the preprocessing steps mentioned in the [Data Preparation](#) section, an additional feature called day_name was created and used in the modelling of the data. The day_name feature had the day names of the week (e.g. Monday or Tuesday) to capture the upward and downward trend of the ticket prices depending on the day of the week.

Hyperparameters used for this modelling:

- n_estimator or weak learners: 200
- learning_rate: 0.1
- max_depth: 7
- min_child_weight: 3
- gamma: 3
- subsample: 0.9
- colsample_bytree: 0.9

n_estimator of greater than 200 was not used in the modelling as a value greater than 200 resulted in overfitting to the training data


A learning rate of less than 0.1 was not used as the loss function was decreasing much slower.

A max_depth value of more than 7 was not used as it led to overfitting to the training data. On the other hand, a value lower than 7 meant that the model was not learning complex relationships between the features.

A min_child_weight value of 3 was used to reduce overfitting as it would not let the algorithm create partitions with too few samples.

A gamma value of 5 was also used to prevent overfitting as a higher value means it would require significant loss reduction to make another partition from the leaf node.

A subsample value of 0.9 and colsample_bytree value of 0.9 was used to limit the number of features and observations used to create each tree. This was also done to reduce overfitting.



Results obtained from this modelling:

Data split	MAE	RMSE	R-squared
Train	60.60	92.45	0.77
Test	70.89	101.08	0.62

The table above reveals a notable disparity between the test and train performance metrics. Specifically, the test MAE score (70.89) and test RMSE score (101.08) were considerably higher than their respective training scores (60.60 and 92.45), suggesting that the model was overfitting to the training data. Given the typical range of air ticket prices, which range from the hundreds to the lower thousands, the MAE scores seem to be considerably high, suggesting that the model was not doing so well.

This is further reinforced by the test R-squared score of only 0.62, which means that the model was able to only explain 62% of the variance in the test data, leaving room for improvement. Nevertheless, the imposition of feature constraints, driven by limited user inputs, deemed this level of model performance acceptable for the current context. Perhaps, the feature constraints can be relaxed in the future to improve the modelling of the data by acquiring newer features with higher predictive power.



6. Evaluation

a. Evaluation Metrics

The evaluation of the machine learning models involved four key metrics: R2 (Coefficient of Determination), RMSE (Root Mean Squared Error), MAPE (Mean Absolute Percentage Error), and MAE (Mean Absolute Error). R2 is a measure of how well the model predicts the variability in the data, with a higher value indicating a better fit. RMSE quantifies the average magnitude of the model's prediction errors, providing a comprehensive assessment of prediction accuracy. MAPE measures the average percentage difference between the predicted and actual values, offering insights into the model's performance relative to the scale of the data. Finally, MAE represents the average magnitude of errors without considering their direction, providing a straightforward measure of model accuracy. The choice of these metrics reflects a comprehensive evaluation strategy, considering aspects of fit, accuracy, percentage errors, and absolute errors, which are all crucial in understanding and optimising the models for the specific project goals. These metrics collectively enable a nuanced assessment of model performance, informing decisions on model selection and refinement.

b. Results and Analysis

The median, minimum, mean and modal of airport fares models were trained, predicted and their performance were assessed based on RMSE and R2 metrics. To accurately predict airport fares, the performance of models was the most important part in our project. Hence, we all tried to use fast and accurate algorithms that can handle large and complex nature of features. XGBoost was the most common algorithm that was used in most of the experiments and consequently outperformed most of the models with other algorithms. Nevertheless, the performance of models we obtained for mean, median, minimum and modal airfares were poorer than expected due to insufficient optimal number of features that were used in modelling.


RMSE of our models was a range from 90 to 103 and r^2 obtained was a range from 0.60 to 0.72. There could be a slight performance gap between models of airfare, as we prepared and transformed features in different ways.

The best **RMSE** scores for each part of airfare were as follow :

Median Fare: 89.46

Mean Fare: 102.03

Modal Fare: 92.45



On the other hand, there may be few ways to improve the performance of our models. First, the project aimed at building user-friendly applications that can help people to estimate their travel fares. Hence, a minimum information of flight was used as features but if more relevant and influential features such as distance and time of flight are included as features, the performance of models will significantly increase. Moreover, if we use more optimal hyperparameters for our models using search tools, such as 'GridSearchCV' or 'RandomSearchCV', the performance of models can be improved. Lastly, subsampling methods, such as training subsampling and feature subsamplings can be considered to increase the performance of models.

c. Business Impact and Benefits.

The final model, despite relying on limited user features, emerges as a practical and impactful tool for businesses, particularly due to its simplicity in required user inputs. This simplicity enhances its accessibility and usability for a wide range of users. Notably, the model's performance, particularly the robust CatBoost model for median fares, holds significant implications for both travellers and airlines. Travellers would be able to leverage the final app to estimate costs for different dates and routes, facilitating effective budgeting and travel planning. The accuracy of the model would empower users to make well-informed decisions on optimal booking dates, potentially leading to cost savings and improved travel experience.

Moreover, the benefits extend to airlines, providing them with a strategic advantage in the competitive market. The fare predictions generated by the final app offer valuable insights for airlines to refine and guide their competitive pricing strategies. This capability would allow airlines to have a competitive edge over rivals. The inclusion of multiple statistical measures, including minimum, mean, median, and modal fares, in the final app addresses the challenge of limited airline-specific pricing data. This comprehensive approach offers users a well-rounded set of actionable fare estimates, covering various scenarios from best-case to typical and most common fare values.

The final app's ability to provide trustworthy estimates is a key factor in mitigating data limitations. Users can rely on the estimates for budgeting and planning, fostering trust in the tool's predictive capabilities. The user-friendly interface further enhances its real-world applicability, catering to diverse user needs, from budget-conscious travellers to business travel coordinators and travel agencies.



d. Data Privacy and Ethical Concerns

This section provides a thorough examination of the data privacy implications associated with the project, emphasizing considerations related to data collection, usage, and deployment, with a specific focus on sensitive information. Notably, the dataset utilized in this project excludes any personal information, resulting in the absence of inherent data privacy risks. The project demonstrates a commitment to ethical principles, proactively addressing privacy concerns throughout its lifecycle.

The careful execution of the project underscores its dedication to maintaining the highest ethical standards. The combination of the dataset's lack of personal information and proactive privacy measures showcases a commitment to safeguarding privacy. However, ethical challenges arise from the economic implications of accurate fare predictions, particularly in dynamic pricing strategies, necessitating a focus on preventing pricing discrimination and ensuring equal access.

A comprehensive approach is essential, integrating robust data protection measures, transparent model development, unbiased algorithms, clear user consent mechanisms, and adherence to regulations. Moreover, the project addresses the environmental impact of resource-intensive model training, highlighting the importance of sustainability in technological development.

A multifaceted strategy that incorporates ethical considerations and data protection measures is imperative. This approach ensures the project addresses the complex ethical and data privacy implications, emphasizing transparency, fairness, regulatory compliance, and environmental responsibility.





7. Deployment

a. Model Serving

The product was deployed as a containerized Gradio web application, utilising inputs such as origin and destination airports, cabin type, date and time category to yield four predictive fare estimates corresponding to each summary metric. Gradio boasts an ease of setup that is particularly advantageous; it can run directly from a notebook, which streamlines the process for users familiar with notebook environments. While it offers much of the same functionality as Streamlit, Gradio is specifically optimised for machine learning models, making it the preferred choice for developers looking to quickly deploy and share their models.


On the flip side, Gradio is relatively new compared to Streamlit and thus lacks some customization options, such as a dedicated date picker. To circumvent this limitation, we implemented separate inputs for day, month, and year, which, although not as elegant as a single date picker, effectively allows for precise date selection within the application's user interface.

To enhance real-world applicability, deployment was refined to accept only extant flight routes, and the user interface was streamlined for ease and speed in obtaining fare predictions. Date input validation enforces day entries between 1 and 31 and year entries between 2020 and 2030. However, given the model's training on data from the same years, it may not effectively discern seasonal trends.

The model's reliability is limited by its training dataset, which spans only two months, potentially undermining its predictive accuracy for periods beyond this duration. Consequently, it may not fully encapsulate annual or seasonal fluctuations. Moreover, the lack of airline-specific differentiation means the model provides general averages rather than precise, airline-specific fare predictions; thus, it functions as an indicative tool rather than a booking engine. It is important to note the difference.

Another challenge is the search date parameter's reliance on the proximity to the flight date, rendering predictions for past dates invalid. While the model presupposes daily flight availability across all time categories, its accuracy is susceptible to volatility in fuel costs, unforeseen events such as natural disasters or conflicts, and economic changes like tax adjustments.

For future deployment efforts, it is recommended to extend the training dataset to cover a full annual cycle, allowing for the capture of comprehensive temporal trends. Additionally, incorporating airline-specific data could significantly refine fare estimates. To account for dynamic



factors affecting flight prices, it might be prudent to integrate real-time data feeds or establish periodic model retraining schedules to maintain prediction accuracy.

b. Web App

The application takes inputs such as origin and destination airports, cabin type, date and time category to yield four predictive fare estimates corresponding to each summary metric. These metrics collectively offer a comprehensive view of the expected price range.

The instructions for deploying the application have been provided below:

1. Clone the main branch for the github repo: https://github.com/naeer/at3_advanced_ml.git
2. In a terminal, navigate to the root folder containing the repo and build the docker image by running the command: `docker build -t flight-fare-app .`
3. Launch the application with the command: `docker run -p 7860:7860 flight-fare-app`
4. The application will be available on the URL: <http://localhost:7860>
5. Fill in the input form and hit submit!

The application is designed to cater to a variety of users who require quick and reliable fare estimates for air travel. Some examples include:

- Budget-conscious travellers can use the application to explore different flight options and find the best deals that align with their travel dates and preferences.
- Business travel coordinators and corporate finance teams may find the application invaluable for forecasting travel expenses and managing budgets.
- Travel agencies and independent travel consultants can leverage the tool to provide clients with immediate and comprehensive fare estimates, enhancing their service offering and efficiency.

The main limitations concerning the web application pertain to the models and training data - they have been discussed in more detail in 7a. Model Serving.





8. Collaboration


a. Individual Contributions

Rushab contributed significantly to the team's efforts by initiating the use of summary metrics, which shaped the entire project strategy. He came up with the features in the data preparation notebook, ensuring a consistent methodology was employed by all team members from the outset. His focus was on constructing a model with the median fare as the target variable, and he authored the respective section along with those on business understanding, data preparation, and deployment sections. Moreover, Rushab crafted the Dockerfile and built most of the Gradio application - the other team members only had to specify the model location and feature names to integrate their models.

Naeer played an important role in this project by facilitating group discussions from the start and agreeing on the project's objectives with his team members. In subsequent discussions, he along with his team members came to a careful selection of features that aligned with the user's limited inputs to ensure the app's simplicity. His primary focus was on building a model for the modal fare prediction. As a result, he built a data processing and modelling pipeline using scikit-learn to model the data using XGBoost. During the preprocessing phase, he introduced additional key features such as `day_name` to the overall data modelling project. After completion of modelling the data, he seamlessly integrated the model with the front-end Gradio app, ensuring that the modal fare prediction was being displayed in the app. In addition to his data modelling responsibilities, he performed exploratory data analysis to gain useful insights from the data to aid in the modelling phase. Furthermore, he wrote the data understanding, exploratory data analysis, modal fare modelling, and business impact and benefits sections of the report. Additionally, he contributed to the readme file of the project.

Joanne's role was focused in constructing a model with the minimum fare as the target variable. Her contributions extended to authoring key sections of the report, such as modeling, data privacy, ethical concerns, and considerations. In addition to her specialized areas, Joanne demonstrated versatility by addressing issues faced during the project and filling in gaps in other sections.

Simon contributed to the modelling part, building features and writing some major parts of reports. His part in modelling was to predict the mean airfare. He used common features and extra features to train models and obtain the optimal performance of models. He tested different algorithms and hyperparameters to obtain the best model performance. There was a major issue when transforming a predicted model to a data product using Gradio. An error regarding some



columns data type kept raising but issues were successfully fixed after changing approaches to building features. His parts in writing reports include executive summary, modelling and evaluation.

b. Group Dynamic

Rushab and Naeer played pivotal roles in ensuring effective teamwork and collaboration within the group. As leaders, they excelled in facilitating group discussions, fostering open communication, and guiding the team towards consensus. Leveraging Microsoft Teams, the team's primary communication platform, they enabled real-time collaboration, conducted a few virtual meetings, and seamless coordination through all the project stages. Additionally, they ensured tasks were delegated efficiently and progress was tracked through on MS teams as well as github. Regular check-ins and updates, along with a proactive approach to conflict resolution, helped maintain a cohesive and productive team dynamic.


Joanne, a valued member of the team, brings dedication and enthusiasm, despite being relatively less experienced. Her primary focus has been on training the random forest regressor model, specifically with an emphasis on computing minimum fares. Throughout her journey, Joanne has demonstrated a commendable commitment to learning and growth, benefiting significantly from the support and guidance provided by her colleagues. Notably, Naeer has played a pivotal role in assisting Joanne in overcoming challenges and resolving issues, contributing to her ongoing professional development within the team.

Simon was a team member of the group, who concentrated on back-up and details of the project. He regularly checked details and tried to communicate with members if there were any issues or questions. His role was to check the progress and manage and find any mutual issue in the project. As communication was a main key to obtain the best results for all models, it was important to check all members' codes and ensure we are all on the same track.

c. Ways of Working Together

The group effectively managed their collaborative efforts through a combination of methodologies and tools. Communication was maintained on Microsoft Teams, fostering an open line of continuous discussion among the four team members. Initial and subsequent meetings were conducted to kickstart and progress the project, primarily taking the form of asynchronous chat discussions.

To ensure parallel development and individual contributions, the team employed a centralized GitHub repository with each member having a dedicated branch for model training. Decision-making was consensus-driven, with all team members reaching agreement before



implementing any decisions. Notably, Microsoft Teams and GitHub were the primary tools utilized for communication, collaboration, and version control.

This approach facilitated an agile and streamlined project management process, ensuring efficient progress tracking and cohesive decision-making throughout the project's lifecycle.

d. Issues Faced

Throughout the project, the team encountered some challenges, primarily revolving around data preparation for model training. Notably, Rushab initiated the development of summary metrics, a pivotal step that guided the team's subsequent actions. Naeer played a crucial role in addressing data issues faced by less experienced team members, demonstrating his expertise and leadership by creating a central repository pivotal to the project's success.

The team's response to these challenges was marked by open communication. Regular discussions on Microsoft Teams facilitated collaborative problem-solving as models were trained and the project progressed. Naeer's repository became a focal point for streamlined collaboration, further aiding in overcoming data-related obstacles.

Reflecting on these challenges, the team identified the importance of continuous communication and shared initiatives to tackle issues head-on. As a lesson learned, the team recommends implementing regular team meetings specifically dedicated to addressing challenges faced during the project. These meetings would provide a platform for collective problem-solving, knowledge sharing, and skill development, ultimately improving the efficiency of future group collaborations.

The team's ability to overcome challenges was rooted in effective communication, shared initiatives, and the expertise of individual team members. The project's success serves as a testament to the importance of addressing issues collaboratively and highlights the value of ongoing improvement through regular team interactions.





9. Conclusion

In conclusion, the project focused on developing machine learning models to predict airfare prices for non-stop flights within the United States, catering to both travellers and the airline industry. Extensive data preparation, feature engineering, and model evaluation processes were undertaken, leading to the selection of CatBoost as the top-performing algorithm. The project addressed the complexities of airfare estimation by considering minimum, median, mean, and modal fares, providing users with a comprehensive overview of potential costs. Despite achieving moderate success in model performance, there is room for improvement, particularly in incorporating additional relevant features such as flight distance and time. The deployment of the models as a Gradio web application enhances accessibility, offering a user-friendly interface for estimating airfare costs based on minimal input parameters. Ethical considerations and data privacy concerns were proactively addressed throughout the project, ensuring compliance with ethical principles. Overall, the project's success lies in its impactful and practical tool for estimating airfare prices, contributing to better planning and decision making for both travellers and airlines.





10. References

- *Sklearn.impute.SimpleImputer*. scikit. (n.d.).
<https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html>
- *Target encoder*. Target Encoder - Category Encoders 2.6.3 documentation. (n.d.).
https://contrib.scikit-learn.org/category_encoders/targetencoder.html
- Ray, S. (2023, September 25). *8 ways to improve accuracy of machine learning models (updated 2023)*. Analytics Vidhya.
<https://www.analyticsvidhya.com/blog/2015/12/improve-machine-learning-results/#:~:text=There%20are%20several%20ways%20to,bagging%2C%20boosting%2C%20and%20stacking.>



11. Appendix

Data Dictionary of the dataset used in this project:

Field	Description
flightDate	The date (YYYY-MM-DD) of the flight.
searchDate	The date (YYYY-MM-DD) on which this entry was taken from Expedia.
startingAirport	Three-character IATA airport code for the initial location.
destinationAirport	Three-character IATA airport code for the arrival location.
departureTimeRaw	String containing the departure time (ISO 8601 format: YYYY-MM-DDThh:mm:ss.000±[hh]:00) for the flight.
cabinType	String containing the cabin type of the flight (e.g. coach, premium coach, business).
totalFare	The price of the ticket (in USD) including taxes and other fees.