EXPERIMENT REPORT

Student Name	Wongwara Wijara
Project Name	Regression models for fare prediction
Date	8 November 2023
Deliverables	WW_notebooks/**Wijara_Wongwara-14191732-xgboost.ipynb** <XGBoost regressor> <WW_notebooks><EDA.ipynb>

---

1. EXPERIMENT BACKGROUND

Provide information about the problem/project such as the scope, the overall objective, expectations. Lay down the goal of this experiment and what are the insights, answers you want to gain or level of performance you are expecting to reach.

1.a. Business Objective

The goal of this project for the business:

The project aims to create a data product for users in the USA that accurately predicts local travel airfare. Users input their trip details, and the product provides fare estimates.

The results of this project will be used in the following ways:

Use cases	Description
1. User Assistance	The primary purpose of the results is to provide a useful tool for visitors to the United States. Users will benefit from being able to enter their travel plans and receive fare estimates, which will allow them to budget for their trips and make cost-conscious decisions when booking flights.
2. Business Revenue	If the data product is a paid service or generates revenue through advertisements, commisions, or other means, the accuracy of the fare estimates can have a direct impact on the business's profitability. Accurate estimates may entice more users, resulting in higher revenue.
3. Marketing and User Engagement.	Accurate results can be used to promote the data product's ability to provide reliable fare estimates. This can result in increased user engagement and a favourable reputation.

Table 1: Business use cases

The impact of accurate or incorrect results:

Accurate results	Incorrect results
1. Improved User Trust: Accurate fare estimates will increase user trust, making them more likely to rely on	**1. User Dissatisfaction:** If fare estimates are consistently incorrect, users may become frustrated

and recommend the data product to others.	and lose trust in the data product, potentially abandoning it in favour of alternatives.
2. Improved User Experience: Users will have a better experience when planning their trips, which will lead to increased satisfaction and continued use.	2. Incorrect results can **harm a company's reputation**, making it difficult to attract new customers and retain existing ones.
3. Increased Business Success: Accurate results can attract more users and boost the commercial success of the data product, potentially generating more revenue.	**3. Revenue Loss:** Inaccurate results may lead to decreased user engagement and a decrease in the revenue potential of the data product.

Table 2: The impact of accurate or inaccurate results

In summary, Accurate results enhance user trust, satisfaction, and potential business revenue. Incorrect results can frustrate users, damage the business's reputation, and lead to revenue loss.

1.b. Hypothesis

Hypothesis:

Can different algorithms, specifically Decision Tree, XGBoost (with hyperparameter tuning via GridSearch), and AdaBoost, provide more accurate predictions for local travel airfare in the United States, and which algorithm performs best based on Mean Absolute Error (MAE) and Mean Squared Error (MSE) metrics?

Reasons for Consideration:

1. Algorithm Comparison: This experiment want to compare the performance of different algorithms (Decision Tree, XGBoost, and AdaBoost) in predicting local travel airfare. This is useful because different algorithms may capture different underlying patterns in the data.

2. Hyperparameter tuning for XGBoost via Grid Search demonstrates the dedication to optimising model performance.

3. Performance Metrics: Using MAE and MSE as evaluation metrics enables a thorough evaluation of model accuracy. MAE provides information about the average magnitude of prediction errors, whereas MSE considers the square of errors. These metrics assist us in determining the practical implications of model performance for travel fare estimates.

4. Practical Application: Accurate fare estimates are critical for travelers planning their trips and effectively managing their expenses. If one of these algorithms can significantly improve accuracy, it could improve user satisfaction and possibly lead to a successful data product.

1.c. Experiment Objective

The experiment's expected outcome is to identify the algorithm that provides the most accurate predictions for local travel airfare in the United States.

Here are a few possible scenarios and objectives:

Model	objectives
1. XGBoost Outperforms After Hyperparameter Tuning	If XGBoost performs better in terms of MAE and MSE after hyperparameter tuning with Grid Search, the goal would be to validate the effectiveness of this tuning method and use XGBoost as the primary model.
2. AdaBoost Emerges as Surprising Leader	AdaBoost, despite being less complex than XGBoost, may perform exceptionally well in this context. The goal is to consider AdaBoost for its efficiency and investigate why it excels in this task.
3. No Clear Winner	In some cases, the difference in MAE and MSE between the three algorithms may be insignificant. If this occurs, the goal would be to delve deeper into the

	nuances of the data and model performance, as well as potentially investigate ensemble methods or more advanced modelling techniques to improve prediction accuracy.
4. Model Improvement Required	It's also possible that none of the algorithms perform adequately, in which case the goal would be to identify the need for additional data features, more data, or alternative modelling approaches to improve accuracy.

Table 3: Possible scenarios and objectives

In any case, the goal is to obtain more accurate fare estimates, which will benefit travelers by allowing them to make more informed decisions about their local travel expenses. The best algorithm will be chosen based on its ability to reduce MAE and MSE, resulting in more reliable predictions.

2. EXPERIMENT DETAILS

Elaborate on the approach taken for this experiment. List the different steps/techniques used and explain the rationale for choosing them.

2.a. Data Preparation

Merging Datasets: Merge datasets from various airports into a single data frame for exhaustive analysis.

Converting Travel Duration: Convert travel duration from a string format into seconds for consistency and easy analysis.

Segment Splitting: The segment columns are split into lists.

Total Segment Duration: Calculate the total duration of segments and create a new column named "totalDuration (segment)."

Travel Layover: A new column, "travelLayover," represents the difference between travel duration and the total segment duration.

Transit Airport Code: Extract and store the airport code of transit locations in a new column named "transitAirportcode."

All Airport Codes: We collect all airport codes, including departure, arrival, and transit, and store them in a new column labelled "All airport."

Date Transformation: The date column serachDate and flightDate is transformed into separate columns for day, month, and year.

Departure Time Extraction: Extract the departure time in Unix epoch seconds format and separate it into individual columns for hours, minutes, and seconds.

Handling Null Values: To address missing data, fill null values with the mean of the respective columns.

Feature selection:

```
Features sorted by correlation score with 'y':
segment_totalDistance: 0.57
totalTravelDistance: 0.57
segment_totalDurationInSeconds: 0.55
travelDurationInSeconds: 0.45
travelLayover: 0.25
startingAirport: 0.16
destinationAirport: 0.15
searchDate_month: 0.08
flightDate_month: 0.06
flightDate_day: 0.06
isRefundable: 0.02
DepartTime_minute: 0.01
legId: 0.00
DepartTime_hour: -0.03
searchDate_day: -0.05
isBasicEconomy: -0.24
isNonStop: -0.30
searchDate_year: nan
flightDate_year: nan
DepartTime_second: nan
```

Figure 1: features sorted by correlation score with the target (y)

We carried out feature selection by identifying variables that exhibited a correlation with the target variable (total fare) above |0.24|. In doing so, we maintained a critical consideration for multicollinearity levels between features, ensuring they remained below 0.7.

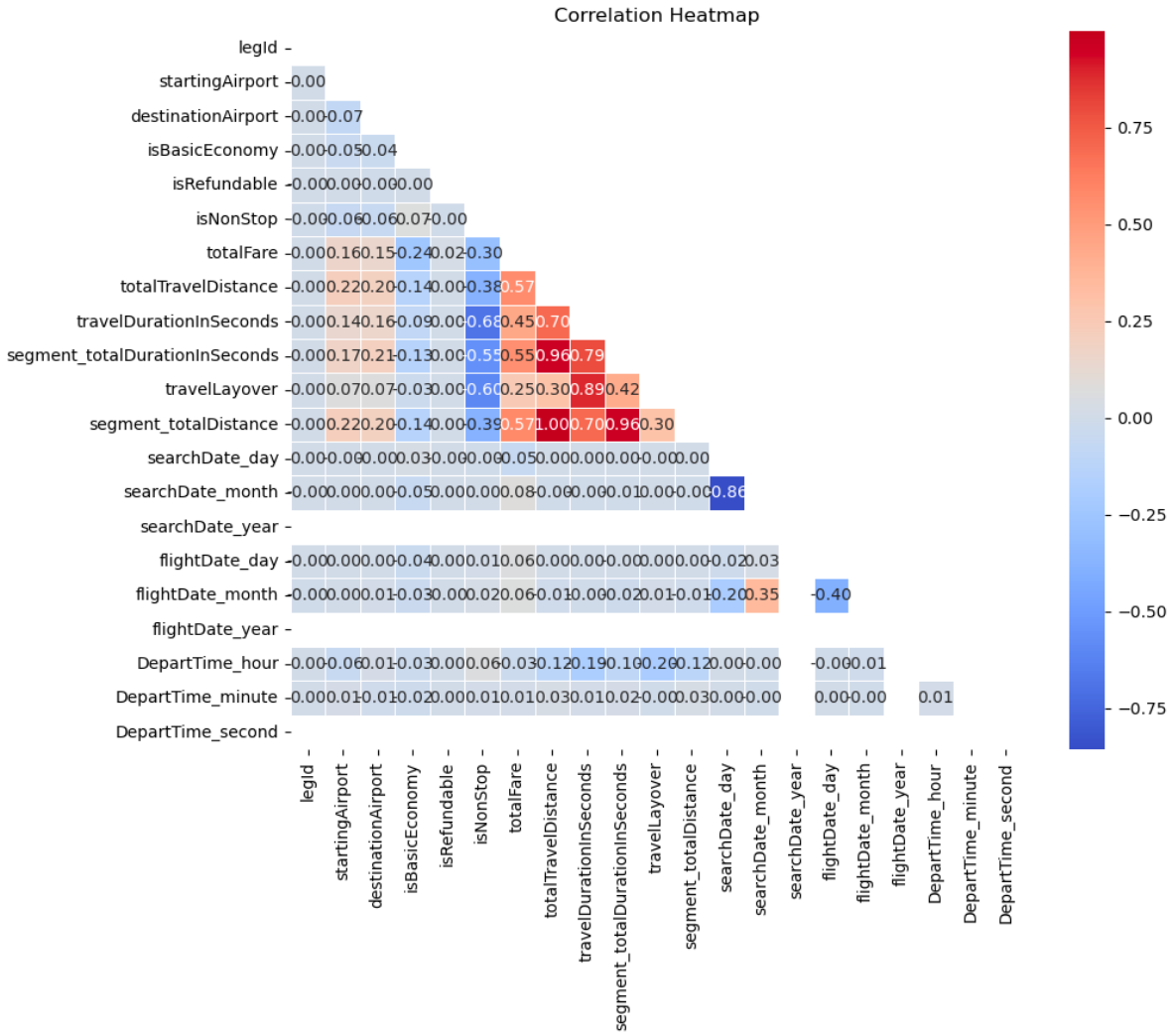

Figure 2's heatmap visually represents these relationships.

The selected features, characterised by both high correlation with the target variable and low collinearity, encompass 'totalTravelDistance,' 'isNonStop,' 'isBasicEconomy,' 'startingAirport,' 'destinationAirport,' 'segmentsCabinCode,' 'flightDate_day,' 'flightDate_month,' 'flightDate_year,' 'DepartTime_hour,' 'DepartTime_minute,' 'DepartTime_second,' and 'totalFare.'

We did all the step for data preparation as a group for a higher data quality, efficiency, and a more robust and transparent process.

2.b. Feature Engineering

Factorising Segment Columns: The segment columns were factorised to enhance their usability in modelling.

Label Encoding 'SegmentCabinCode': Applied label encoding to the 'segmentCabinCode' column, converting categorical data into a numerical format for analysis.

Encoding Categorical Columns: Other categorical columns were also encoded using label encoding, ensuring consistency in handling categorical data.

Scaling Numeric Columns: Numeric columns were scaled using Standard Scaler, making them compatible with the modelling process.

2.c. Modelling

- **Algorithm:** Decision Tree, XGBoost and Adaboost

Model	*Rational*
Decision Tree	Decision trees are chosen for their simplicity and interpretability, which is critical when explaining fare predictions to passengers or stakeholders. They can capture non-linear relationships in tickets price factors and handle a mix of categorical and numerical features such as flight details, cabincode and travel layover. Their resistance to outliers is advantageous because ticket prices can vary.
XGBoost (default and with learning_rate: 0.2, max_depth: 5, n_estimators: 300	XGBoost was chosen for its high-performance capabilities, which can assist in modelling the complexity and noise in ticket price data. In real-world airline pricing datasets where data quality varies, the ability to handle missing values and automatically manage imbalanced data is critical. XGBoost can provide feature importances, which can help understand the main drivers of fare predictions.
Adaboost	AdaBoost improves the performance of the decision tree model. AdaBoost can help decision trees perform better in the context of airline ticket pricing by focusing on their weak points. This ensemble method improves predictive power while preserving interpretability, which is critical for ensuring transparent pricing strategies.

Table 4: Models used.

These hyperparameters were chosen based on common practices and considerations for XGBoost regression models.

- n_estimators: [100, 200, 300] - The number of boosting rounds or trees in the ensemble.
- max_depth: [3, 4, 5] - The maximum depth of the individual trees.
- learning_rate: [0.1, 0.2, 0.01] - The learning rate, which controls the step size during optimization.

Model Evaluation:

And the hyperparameters are: {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300}

Trained models include:

1. Decision Tree: Decision trees are chosen for their ease of interpretation and simplicity. They provide a good baseline model for comparison with more complex algorithms. They are frequently appropriate for initial model testing.

2. XGBoost was chosen for its well-known ability to handle complex data and optimise model performance. GridSearch hyperparameter tuning is used to improve its accuracy even further.

3. AdaBoost is used as an ensemble model to investigate whether boosting techniques can improve model accuracy.

Model Selection Justification:

1. Decision Tree: Decision trees are chosen for their clarity and simplicity. They are ideal for preliminary testing and comparing to more complex models. Furthermore, they can be used as a starting point for assessing the added value of more advanced models.

2. XGBoost: XGBoost is a powerful gradient boosting algorithm with excellent predictive performance. It was chosen because of its hyperparameter tuning capabilities, which enable fine-tuning to achieve optimal results. The expectation is that XGBoost will be able to detect intricate patterns in data.

3. AdaBoost: AdaBoost is an ensemble method for improving the performance of underperforming students. It's included to see if boosting can improve model accuracy. Its performance will be compared to that of Decision Tree and XGBoost.

Untrained Models (Random Forest and Lightgbm):

Time constraints may have influenced the decision not to train the Random Forest model. Training a Random Forest can be computationally demanding, especially when dealing with large datasets or a large number of trees. It took too long in

	this case, exceeding the two-hour time limit. **Future Experimentation Possibility:** In future experiments, when computational resources are more readily available, it is worthwhile to consider training the Random Forest model. Random Forests are well-known for their robustness and performance, which could provide useful insights into model selection for this specific task. When the installation problem is resolved, Light GBM could be investigated further. Light GBM is known for its efficiency and speed, making it an ideal candidate for testing when time is limited.

3. EXPERIMENT RESULTS

Analyse in detail the results achieved from this experiment from a technical and business perspective. Not only report performance metrics results but also any interpretation on model features, incorrect results, risks identified.

3.a. Technical Performance

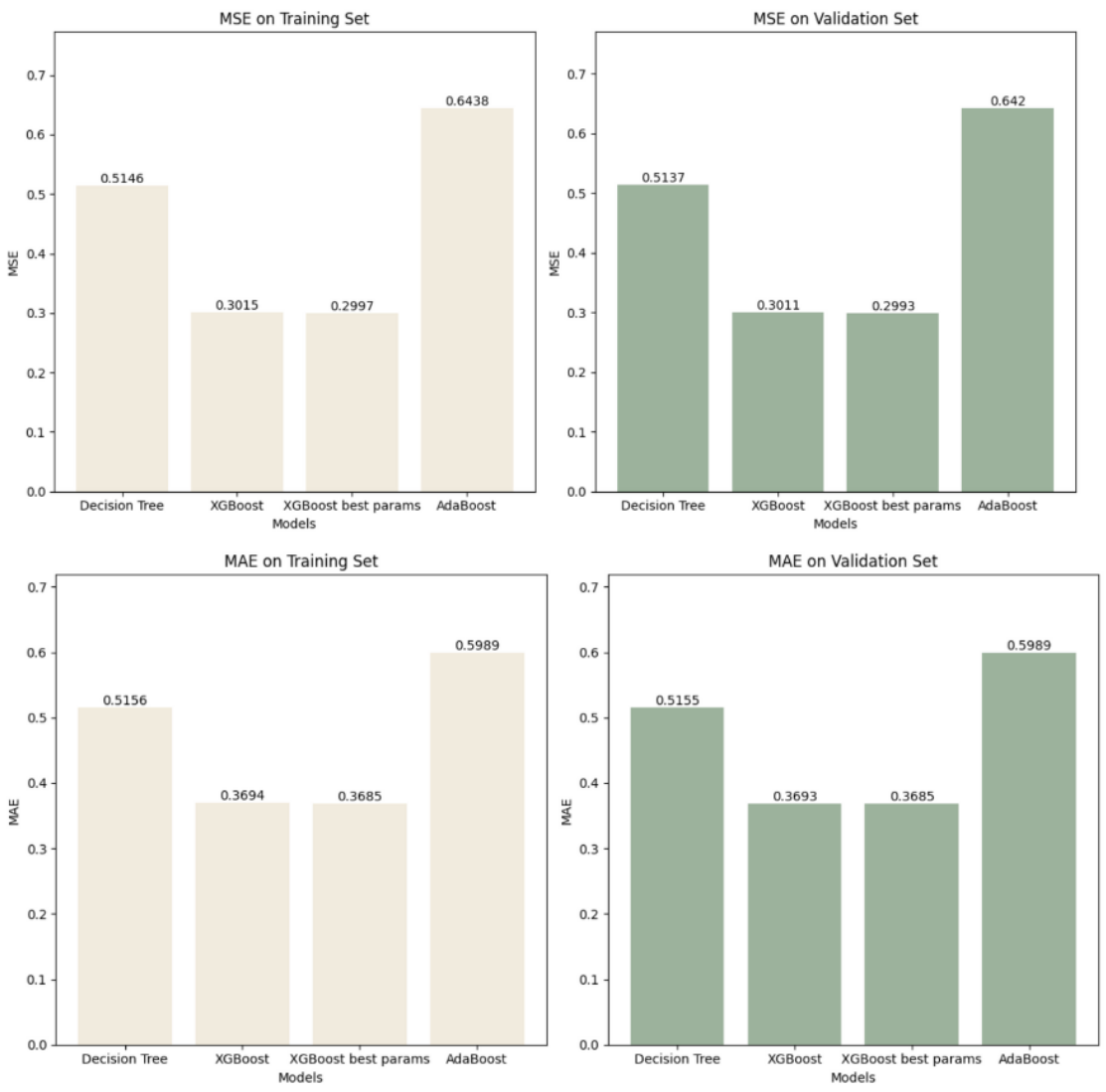

Figure 3: Bar chart of MSE and MAE score for Decision Tree, XGBoost and Adaboost

The MSE and MAE scores reveal significant differences in model performance. The optimally tuned XGBoost model outperforms the default XGBoost by 0.2997 versus 0.3015 on the training set. This demonstrates the tuned XGBoost's improved predictive capability, outperforming the Decision Tree (MSE: 0.5146) and AdaBoost (MSE: 0.6438).

On the validation set, the optimally tuned XGBoost has a slightly higher MSE of 0.2993 compared to the default XGBoost's 0.3011, but it still outperforms the Decision Tree (MSE: 0.5137) and AdaBoost (MSE: 0.642).
Regarding MAE, the optimally tuned XGBoost (0.3685) slightly outperforms the default XGBoost (0.3694) on the training set, highlighting its improved predictive capability. XGBoost outperforms the Decision Tree (MAE: 0.5156) and AdaBoost (MAE: 0.5989).

In terms of MAE, the optimally tuned XGBoost (0.3685) slightly outperforms the default XGBoost (0.3694) on the training set, highlighting its improved predictive capability. XGBoost outperforms both the Decision Tree (MAE: 0.5156) and AdaBoost (MAE: 0.5989) once more.

Case Study of Underperformers:

In both MSE and MAE, the Decision Tree and AdaBoost models consistently underperform XGBoost. This implies that the Decision Tree's simplicity and AdaBoost's boosting technique may not be adequate for capturing the intricate patterns in the data, resulting in higher prediction errors.

Potential Root Causes:

The inability of Decision Tree to handle complex data relationships may result in higher errors. AdaBoost may be ineffective in improving the performance of a poor learner (Decision Tree) in this task. Both models may be overfitting because their errors on the validation set are higher than on the training set.

Future Suggestions:

Continue to investigate the Random Forest model's potential, especially if it is known for its robustness and predictive performance. Examine other ensemble methods and advanced modelling techniques that may outperform XGBoost for this specific prediction task. Investigate feature engineering and additional data pre-processing to improve model performance even further.

However, the disparity in results for predicting the total fare for the year 2023, with a total fare of $10, does appear unusual, especially given that the model was trained on data from 2022. Several factors could explain this disparity:

1. Data Drift: Poor model performance can result if the distribution of data in 2023 differs significantly from that in 2022. When the underlying patterns or relationships in the data change over time, this is referred to as data drift. If there are significant changes in the travel industry or other relevant factors in 2023, the model's accuracy may suffer.

2. Lack of Seasonal or Temporal Consideration: The model may fail to capture seasonal trends or temporal patterns in the data. If airfare prices fluctuate seasonally or over time, the model may struggle to make accurate predictions beyond the time frame it was trained on.

To address this issue and improve the predictions for 2023 **we could consider:**

1. **Collecting more recent data for 2023** to update the training set and account for potential changes.
2. **Including seasonal or temporal trends**, such as holidays, economic indicators, or travel industry news.
3. **Experimenting with time series forecasting models**, which may be better at handling long-term predictions.
4. **Examine the data for inconsistencies,** outliers, or errors that may influence the model's predictions.

Data Understanding:

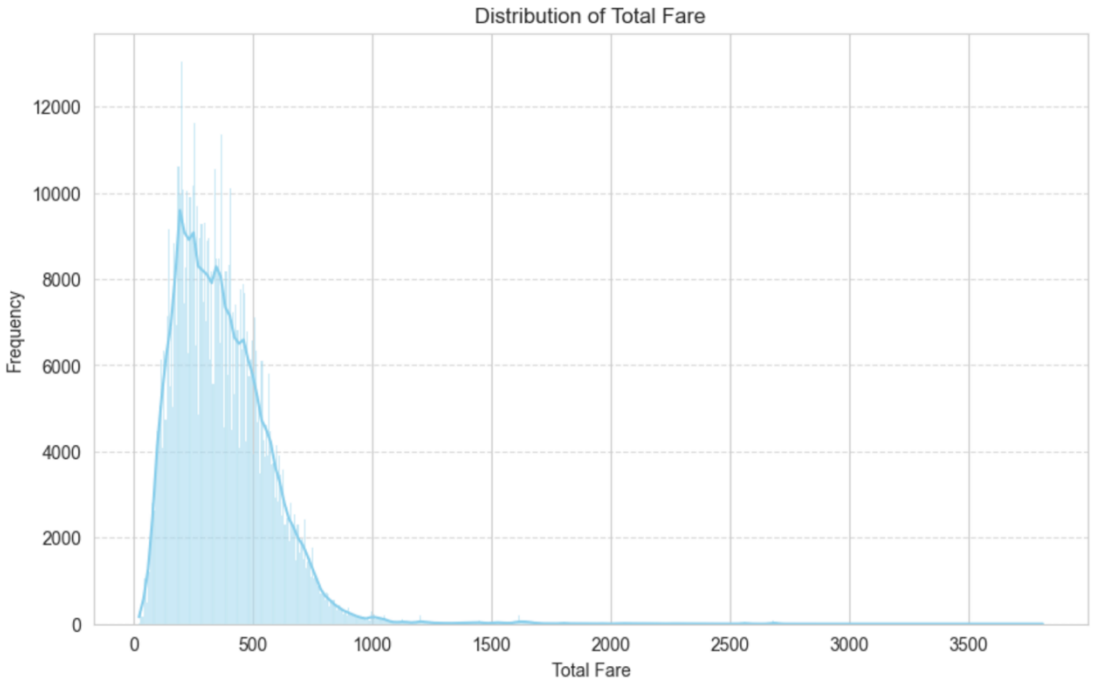

Figure 4: Distribution of Total fare

The total fare distribution is skewed to the left, with the most frequent occurrence of total fare values exceeding 10,000 times. Furthermore, a sizable proportion of total fare values fall between $300 and $500. This suggests that the majority of total fare values are concentrated in the lower range, with a long tail extending to the right, where a few extremely high values may exist. Lower fares are more common, according to the left-skewed distribution, while higher fares are less common but can have a significant impact on the overall distribution due to their high values.

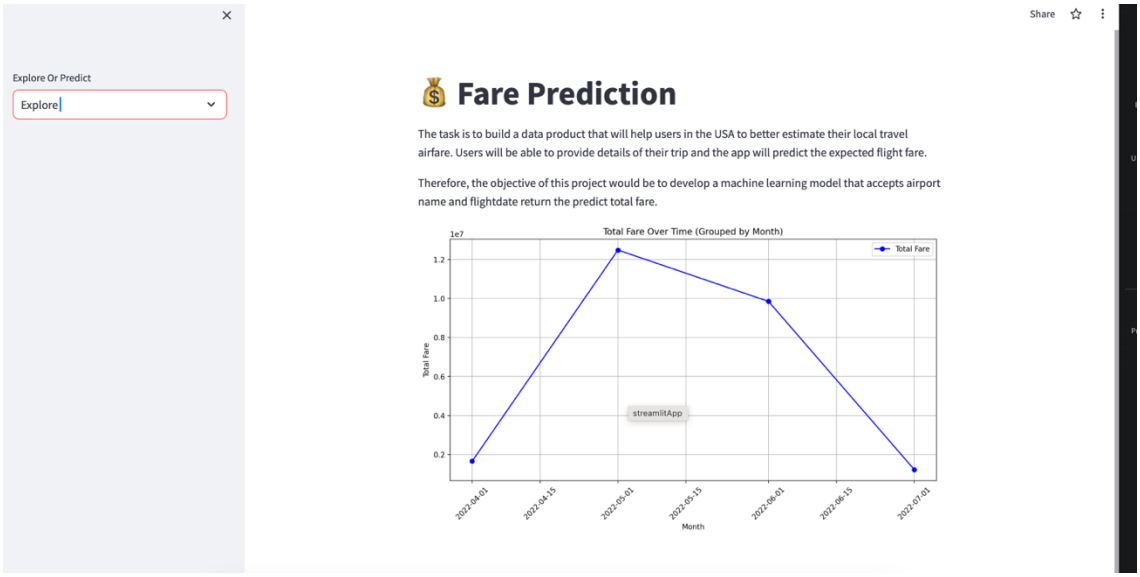

Figure 5: Total Fare over time

The average total fare in each month shows that the highest total fare is in May 2022, followed by a slight decrease in June and July 2022. According to this trend, May is the peak month for total fare, with the highest average. However, there is a slight decrease in average total fare in June and July, which could be influenced by seasonality, demand, or pricing strategies. Analysing these monthly fluctuations can provide insights into travel trends and assist businesses in making informed pricing and marketing decisions.

3.b. Business Impact	To summarise, the impact of incorrect results on business is significant and multifaceted. User trust, company reputation, revenue, and competitiveness are all on the line. To mitigate these risks, it is critical to address the issues that are causing inaccurate predictions, improve model accuracy, and ensure that the data product meets the business objective of providing accurate fare estimates for local travel on a consistent basis. **Results Interpretation:** **1. Model Performance:** Three models were trained in the experiments: Decision Tree, XGBoost (with hyperparameter tuning), and AdaBoost. On both the training and validation sets, the optimally tuned XGBoost consistently outperformed the other models in terms of Mean Squared Error (MSE) and Mean Absolute Error (MAE). As a result, XGBoost is the most accurate model for estimating local travel airfare. **2. Data and Time Issues:** In predicting the total fare for the year 2023, an anomaly was discovered, with the model producing a very low total fare of $10. This disparity could be due to data drift, a failure to account for seasonal or temporal trends, data sufficiency issues, or model limitations, as previously discussed. **Impact of Incorrect Results on the Business:** **1. User Satisfaction:** Incorrect predictions, particularly for the year 2023, can erode user trust and satisfaction. Users plan their trips and budgets based on accurate fare estimates. Anomalies, such as the $10 total fare, can cause user annoyance and a loss of trust in the data product. **2. Business Reputation:** Inaccurate results can harm a company's market reputation. Users may share their negative experiences, resulting in a negative perception of the data product. Negative reviews and word-of-mouth can turn off potential customers. **3. Revenue Loss:** Users lose trust in the data product as a result of consistently incorrect predictions, which can result in revenue loss. Users may discontinue use of the service or choose competitors who provide more reliable estimates.
3.c. Encountered Issues	**Issues Resolved:** **Longer Time Wait:** To address the issue of longer model training times, simpler models such as Decision Tree were chosen. As a result, we were able to complete the training within the time constraints. **Unresolved Problems:** **1. Strange Outcomes Prediction:** The issue of strange predictions for the year 2023, with a total fare of $10, remains unresolved. This anomaly could be caused by a number of factors, including data drift, a failure to account for temporal trends, or a lack of data sufficiency. In order to address this: **1.1 Data Drift:** Gather more recent data for the year 2023 to update and align the training set with the current data distribution. **1.2 Consideration of Time:** To improve the model's ability to handle long-term predictions, add features that capture seasonal or temporal trends. **1.3 Data Sufficiency:** Increase the size of the training dataset to include a more extensive history spanning multiple seasons and years. **1.4 Model Limitations:** Consider exploring models specifically designed for time series forecasting to assess the potential limitations of the chosen models in handling long-term predictions.

4. FUTURE EXPERIMENT

Reflect on the experiment and highlight the key information/insights you gained from it that are valuable for the overall project objectives from a technical and business perspective.	
4.a. Key Learning	The experiment to develop models for estimating local travel airfare yielded both valuable insights and some difficulties. Here are the new discoveries: **Model Performance:** The results clearly show that XGBoost outperforms both Decision Tree and AdaBoost in terms of Mean Squared Error (MSE) and Mean Absolute Error (MAE), especially after hyperparameter tuning. This emphasises the significance of choosing the right model and optimising its hyperparameters for accurate predictions. **Data Drift Issues:** The unusual results for the year 2023 predictions, with a total fare of $10, indicate that the model may not be handling data drift effectively. This emphasises the importance of ongoing data monitoring and potential training dataset updates to account for changing data distributions. **Time Constraints:** Due to time constraints, simpler models were chosen at the expense of potentially beneficial algorithms such as Light GBM. A critical consideration is balancing model complexity with available computational resources. While the experiment with XGBoost yields promising results, there is still plenty of room for further investigation and improvement. The findings highlight the need for more experimentation and a commitment to improving the method for accurately estimating local travel airfare. It is not a dead end, but rather a path to more accurate models and valuable insights.
4.b. Suggestions / Recommendations	**Future Experiments Should Address:** **1. Model Complexity:** Choosing simpler models to address the time constraint may have been necessary, but this approach may have failed to capture complex relationships in the data. Future experiments may involve more advanced models while effectively managing computational resources. **2. Installation Issues (for example, Light GBM):** The difficulty in installing certain models, such as Light GBM, is an impediment. Resolving installation issues or finding alternatives in future experiments could provide access to potentially beneficial modelling approaches. **3. Data Pre-processing and Feature Engineering:** It is possible that the experiments did not thoroughly investigate advanced data pre-processing and feature engineering techniques. Future experiments can concentrate on improving the quality of input data and improving the predictive performance of the model. **4. Evaluating Overfitting:** It is critical to investigate the potential overfitting of models, particularly when they perform better on the training set than the validation set. To address this, techniques such as regularisation and cross-validation can be investigated. **5. Handling Outliers:** If the strange predictions for 2023 are due to outliers in the data, future experiments should consider robust methods for handling outliers during data pre-processing. **Potential Next Steps and Experiments:** **1. Data Drift Monitoring:** This should be a top priority because addressing data drift can significantly improve model performance, making future predictions more reliable. **2. Enhancing Feature Engineering:** Improving feature engineering is critical for capturing nuanced patterns in data and improving model accuracy. **3. Advanced Hyperparameter Tuning:** Additional model optimisation can result in incremental improvements, but it must be preceded by data drift monitoring and feature engineering. **4. Outlier Detection and Handling:** Addressing unusual results is critical, but it is dependent on data drift monitoring and advanced feature engineering.

	5. Advanced Models: After addressing the issues with data drift, feature engineering, and outlier handling, more complex models should be explored. **Step for the deployment** The deployment process for our data product, which was built with Streamlit.io, entails several key steps to ensure that the application is accessible and usable by users. The following is a high-level overview of the deployment procedure: **1.** Save trained models**:** Train machine learning model that is ready for deployment. Save it in a format that can be easily loaded such as a joblib file, keras file and pb file. 2. Setup Github repo with requirements files: predict_page.py, app.py and requirements.txt 3. To manage code changes and collaboration, we host our Streamlit application files and models on Github. 4. Host our app on Streamlit Sharing (https://www.streamlit.io/sharing). This app is public and searchable through our chosen subdomain. 5. Streamlit application development and testing: Create our Streamlit application that incorporates our trained machine learning model. Ascertain that the application is user-friendly, provides accurate predictions, and effectively handles user inputs. To identify and address any bugs or issues, thoroughly test the application. 6. User Access: Share the URL of our Streamlit application with our target audience, whether they are internal users within our organization or the public.