Assignment 3  
ML Data Product

short line

Adv MLA 11/23

Group 15

|  |  |  |  |
| --- | --- | --- | --- |
| Student Last Name | Student First Name | Student ID | Group Allocation |
| Vimalasri | Chanthru | 25048240 | Student A |
| Gupta | Tarun | 24607855 | Student B |
| Chen | Zhicong | 12636736 | Student C |
| Shrestha | Nipesh | 24605646 | Student D |

|  |  |
| --- | --- |
| Github | Project + Streamlit Repo: https://github.com/ChanthruV/data\_product\_with\_ml |

36120 - Advanced Machine Learning Application

Master of Data Science and Innovation

University of Technology of Sydney

Table of Contents

[**1. Executive Summary 2**](#_xvtyf97iwmd0)

[**2. Business Understanding 3**](#_if9iwijz2pg3)

[a. Business Use Cases 3](#_bnjfuh5ewogw)

[**3. Data Understanding 4**](#_yg03y3w5hig9)

[**4. Data Preparation 5**](#_tj77rrmixzhi)

[**5. Modeling 6**](#_6vwh14nc44ga)

[a. Approach 1 6](#_uv8sn0ik6017)

[b. Approach 2 6](#_wtwxuxemqtvc)

[c. Approach 3 6](#_nrl8g3ri789x)

[**6. Evaluation 8**](#_mngiyx5tet6w)

[a. Evaluation Metrics 8](#_qx20m6hrk8uo)

[b. Results and Analysis 8](#_25v57zsj5m7g)

[c. Business Impact and Benefits 8](#_hkob76wu4d6q)

[d. Data Privacy and Ethical Concerns 9](#_uf1z6gbsejg6)

[**7. Deployment 10**](#_8rqux13i2o01)

[**8. Collaboration 11**](#_9hr11g79asdx)

[a. Individual Contributions 11](#_5fycwkjr7bjp)

[b. Group Dynamic 11](#_4tp33skt3su4)

[c. Ways of Working Together 11](#_t9y1jnesjjm)

[d. Issues Faced 12](#_oeeyog5w6nd)

[**9. Conclusion 13**](#_usxxc83ipwqu)

[**10. References 14**](#_ma3y8ytjvp1y)

# 

# Executive Summary

**Overview**

Our project developed a data product to estimate US airfares, aiming to make travel planning easier. We created a straightforward app that predicts flight costs based on user input, using machine learning to provide more accurate estimates.

**Problem Statement and Context**

We noticed that travelers often struggle with finding current and specific fare information, which can make budgeting for trips difficult. Our app attempts to address this by using updated data to give users a better idea of what they might pay for their flights. It allows users to customize their inputs to deliver more accurate estimations of fares for their trips.

**Achieved Outcomes and Results**

We've built a Streamlit app that uses four machine learning models to estimate airfares. It's easy to use and gives quick predictions. This shows how machine learning can be applied to help with everyday tasks like planning a trip.

In short, our project offers a practical tool that helps predict the cost of flying, which could make it easier for travelers to budget and plan their journeys.

short dash

# Business Understanding

a**. Business Use Cases**

The project serves the travel and aviation industry by providing accurate fare predictions, essential for airlines, travel agents, and customers. It addresses the need for reliable pricing forecasts for airline ticketing—a tool that assists in making informed decisions about pricing strategies, inventory management, and purchasing tickets.

The impetus for the project came from the complex and ever-changing nature of airline pricing. Prices fluctuate due to various factors, including seasonality, demand shifts, and competitive pricing. Machine learning algorithms are particularly suited for this context as they can analyze historical pricing data to forecast future fares. The objective was to minimize the unpredictability of ticket pricing, aiding travelers in planning and businesses in maximizing revenue.

**b. Key Objectives**

The project had several clear goals:

1. To employ machine learning techniques (e.g., XGBoost, Random Forest etc.) to develop dependable models for predicting airline ticket prices.

2. To provide travelers with accurate fare predictions to inform their booking decisions.

3. To assist airlines and travel companies in refining inventory management and pricing strategies in response to forecasted fare changes.

**Stakeholders and Requirements:**

The stakeholders include passengers seeking cost-effective travel options, and travel agencies and airlines needing to predict fare fluctuations for better inventory and pricing decisions. Passengers require transparency and reliable information on fare trends, while businesses need predictive insights for strategic planning.

**Project Approach:**

To address these requirements, the project leverages machine learning algorithms using historical flight data. This approach allows for detailed fare predictions. For airlines and travel agencies, this translates into the ability to meet demand and adjust pricing strategies proactively. Meanwhile, travelers benefit from making more informed decisions on when to book flights, leading to savings and improved travel experiences. Overall, the project aims to bridge the gap between the need for pricing insights among stakeholders and enable smarter decision-making through machine learning.

short dash

# Data Understanding

**Insights into the Dataset**

The dataset from Expedia is an extensive collection of flight data points that include price, dates, airports, and flight characteristics. The data is structured to reflect the variety of information a potential traveler would encounter when searching for flights, making it a comprehensive source for price prediction.

**Data Sources and Collection Methods**

Data was sourced directly from Expedia's listings, ensuring a high level of relevance as it reflects actual market offerings. It was made available via a google drive where a zip folder was downloaded.

**Data Limitations**

Whilst data was thorough and organised by airport, the various zip files created additional resistance in sourcing and cleaning the data. Zip file functions in python were used to overcome this but could have been averted through the availability of a more collated data source. Some data was missing e.g., 'totalTravelDistance' field, which is critical for fare estimation. Also, the high dimensionality due to numerous categorical variables like airport codes and airline names presented challenges for modeling. These challenges were investigated and dealt as part of data preparation.

**Variables/Features Significance**

Key variables include:

* legId: Useful for data tracking but not for modeling.
* flightDate and searchDate: Essential for capturing seasonal fare variations.
* startingAirport and destinationAirport: Critical for understanding route-specific price dynamics.
* fareBasisCode: Could indicate the fare class and its pricing structure.
* travelDuration and elapsedDays: Time-related factors that could influence pricing.
* isBasicEconomy, isRefundable, isNonStop: Service-related attributes that are likely to affect fare.
* totalFare: The target variable for our prediction model.
* segments...: Detailed itinerary information that could help in fine-tuning the fare prediction but are complex to integrate directly into the models.

**Exploratory Data Analysis**

During the EDA process, we conducted a number of checks including:

* A range check for dates to understand the timeframe covered, to be potentially used for time series analysis.
* A data type assessment to plan necessary transformations.
* A search for missing values to strategize imputation methods.
* An evaluation of unique values in categorical variables to prepare for encoding by factoring in cardinality.

short dash

# Data Preparation

short dash

**Step 1: Unzipping and Loading Data**

Objective: Streamline the process of getting data ready for use.

* Located the source of zip files and prepared a destination for the extracted CSVs.
* Wrote a script to automatically unzip files and compile the data into a single location for easy access.

**Step 2: Combining Data into One DataFrame**

Objective: Create a single view of all data points for analysis.

* Initialized a list to collect individual DataFrames created from each CSV.
* Combined these DataFrames into one master DataFrame for a unified dataset.

**Step 3: Cleaning the Combined Dataset**

Objective: Ensure data quality and relevance for modeling.

* Removed columns unrelated to the fare estimation to declutter the dataset. Focused on what inputs would be easiest for the user, what features were most vital to improve model performance, and what features could also be calculated if not available so they could be used within our Streamlit app.
* Filled in missing 'totalTravelDistance' values with the average to avoid data gaps.
* Deleted any duplicate entries to prevent data redundancy.

**Step 4: Converting and Encoding Features**

Objective: Transform data into a machine-learning-friendly format.

* Changed the 'travelDuration' from text to seconds for numerical consistency.
* Applied hashing to turn complex categorical data into a simpler numeric array, facilitating easier processing by algorithms.

**Step 5: Normalizing the Data**

Objective: Balance all features on a common scale to aid in model accuracy.

* Used StandardScaler to normalize features, negating the influence of outlier values.

**Step 6: Splitting the Data for Training and Validation**

Objective: Reserve a portion of data to test the model's predictions.

* Divided the dataset into training (80%) and validation (20%) sets to ensure the model can be assessed fairly.

**Step 7: Establishing a Baseline for Comparison**

Objective: Set a simple standard to measure the improvement of machine learning models.

* Determined the most common fare value in the training set to use as a constant prediction for initial comparison.

**Step 8: Saving Preprocessing Artifacts**

Objective: Document and save the steps and tools used for preprocessing for future consistency.

* Kept a record of which columns were removed and stored this list for reference.
* Saved the configured StandardScaler to apply the same transformation to new data in the future.

The focus in this portion of code was on maintaining data integrity, ensuring simplicity in the modeling process, and setting up robust standards for model evaluation.

# Modeling

* Describe the machine learning algorithms used for modeling.
* Discuss the rationale behind selecting these algorithms.
* Explain the parameter tuning and model selection process.

Instructions: Describe the machine learning algorithms used for modeling, providing a rationale for their selection based on the project goals. Explain the process of parameter tuning and model selection. Include details about the algorithms' implementation and any considerations made during the modeling phase.

## Approach 1

* Describe the specific details of the first model used, including the algorithm and its key hyperparameters.
* Discuss any preprocessing or feature engineering specific to this model.
* Explain the training process and any techniques used to handle imbalanced data.

Instructions: Provide a detailed description of the first model used, including the algorithm name, its key hyperparameters, and any specific preprocessing or feature engineering steps taken for this model. Explain the training process, including how imbalanced data was handled if applicable.

## Approach 2

* Repeat the same structure as for Approach 1, but provide details for the second model used.

## Approach 3

* Repeat the same structure as for Approach 1, but provide details for the second model used.

Instructions: Provide a detailed description of the second model used, following the same structure as Model 1. Provide the algorithm name, key hyperparameters, and any specific preprocessing or feature engineering steps taken for this model.

short dash

# Evaluation

## Evaluation Metrics

* Describe the evaluation metrics used to assess the models' performance.
* Explain why these metrics were chosen and how they relate to the project goals.

Instructions: Describe the evaluation metrics used to assess the models' performance, including the specific metrics chosen and their relevance to the project goals.

## Results and Analysis

* Present the results of the model evaluation, including accuracy, precision, recall, F1-score, etc.
* Analyze and compare the performance of each model.
* Discuss the key insights gained during the experimentation phases.

Instructions: Present the results of the model evaluation, including accuracy, precision, recall, F1-score, or any other relevant metrics. Analyze and compare the performance of each model, highlighting the key insights gained during the experimentation phases. Discuss the implications of these insights on the project's goals and potential areas for further improvement.

## Business Impact and Benefits

* Assess the impact and benefits of the final model on the business use cases.
* Discuss how the model contributes to solving the identified challenges or exploiting opportunities.
* Quantify the improvements achieved and the potential value generated.

Instructions: Assess and discuss the impact and benefits of the final model on the identified business use cases. Explain how the model contributes to solving the identified challenges or exploiting opportunities. Quantify the improvements achieved and discuss the potential value generated by the model.

## Data Privacy and Ethical Concerns

* Assess the data privacy implications of the project.
* Discuss any ethical concerns related to data collection, usage, or model deployment.
* Address steps taken to ensure data privacy and ethical considerations.

Instructions: Assess the data privacy implications of the project, considering any sensitive information or privacy concerns related to data collection, usage, or model deployment. Discuss any ethical concerns and considerations. Address the steps taken to ensure data privacy and mitigate ethical concerns.

short dash

# Deployment

## Streamlit

Creating the Streamlit app involved designing a user-friendly interface that allows users to input their travel details and receive estimated airfares. Here's how the app was constructed and the logic behind it:

**User Interface Components**

* Title: Introduced the app with a title "Local Travel Airfare Estimator" using Streamlit's st.title function.
* Origin Airport Dropdown: Implemented a dropdown menu using st.selectbox, populated with a list of starting airports provided by the get\_starting\_airports function.
* Destination Airport Dropdown: Similar to the origin, another dropdown for destination airports enables users to select their end point.
* Flight Date Input: A date input control allows users to select their flight date through st.date\_input.
* Fare Type Checkboxes: Checkboxes for basic economy, refundable tickets, and non-stop flights capture user preferences using st.checkbox.
* Cabin Type Dropdown: Another dropdown lets users choose the cabin type, ranging from coach to first class.
* Airline Dropdown: Allows users to select an airline from a list returned by the get\_airline\_names function.

**Back-End Processing**

Upon clicking the "Estimate Fare" button, the following occurs:

* Capture Current Date: The current date is obtained to factor in the booking lead time.
* Preprocess User Inputs: The function preprocess\_user\_input takes the details provided by the user and prepares the data for the prediction model. This includes:
  + Converting booleans to integers for model readability.
  + Transforming dates to UNIX timestamps to handle them as continuous variables.
  + Encoding categorical data like airports and airlines using a FeatureHasher for dimensionality reduction.
  + Fetching average travel durations and distances for the selected routes from a pre-processed dataset, which aids in providing contextual predictions.
* Model Prediction: Each trained model's filename is listed in model\_filenames. For each model, the app:
  + Loads the trained model using joblib.
  + Passes the preprocessed user input to the model's predict function.
  + Collects the fare predictions from each model.

**User Output**

* Predicted Fares: The app displays a simple list of predicted fares from each model, formatted to two decimal places to represent currency.

**Summary**

The Streamlit app serves as an interactive front end to a set of machine learning models. Users can specify their travel preferences, and the app processes these inputs, applies the trained models, and outputs fare estimates. The app makes the machine learning models accessible to non-technical users, encapsulating complex data transformations and predictions within a clean interface.

## Model Serving

**Deployment Process:**

* Selected the best machine learning model based on performance metrics.
* Utilized joblib to serialize the trained model, saving its state for later use.
* Prepared a deployment environment, like a cloud server, ensuring it has the necessary resources.
* Deployed the serialized model onto the server, ready to be called by the application.

**Integration Considerations/ Challenges:**

* Ensured that the model could be integrated seamlessly with the Streamlit web app.
  + This was probably one of the **greatest challenges** for us due to the group nature of this assessment. For this to work well and seamlessly, we all needed to ensure we had the same inputs for our respective models when training, so this aligned with expected inputs for the Streamlit app. In instances where this was achieved, the size of the dataset made certain models computationally inefficient too so there was a lot of trial and error involved.
* Established a reliable data pipeline for the model to receive and process input data.
  + Imputation of certain values that the user shouldn’t need to input (e.g. travelDistance) posed a challenge and required creative yet simple solutions such as deriving the mean for a given origin and destination airport.
* Implemented error handling and logging to address any runtime issues that may arise.
  + Introduced a ‘Predicting Fares…’ loader to give users a greater sense of security and a better indication of processing times.

## Web App

**Purpose and Functionalities:**

The web application acts as a user-friendly interface for the fare estimation model, allowing users to input travel details and receive fare predictions.

It includes dropdowns for airport selection, date pickers for flight dates, checkboxes for flight preferences, and a prediction button to initiate the fare estimation.

**Setup and Launch Instructions:**

* Ensure all dependencies, such as Streamlit and joblib, are installed in the deployment environment.
* Load the web application files onto the server, including app.py and features.py.
* Start the application using Streamlit, which will host the app on a local port that can be accessed via a web browser.

**Potential Users and Use Cases:**

* Travelers looking to estimate airfares for budgeting purposes.
* Airlines and travel agencies to provide quick fare estimates to their clients.
* Benefits include real-time fare estimates, personalized travel options, and the convenience of planning from anywhere.

**Commercialization Potential:**

* The application could be integrated into travel booking websites as a value-added service.
* Subscription-based access for frequent users or travel agencies.
* Data collected from user interactions could be utilized for market analysis and strategic planning.

**Current Limitations and Improvements:**

* The current version might not handle extremely high user loads, necessitating scaling solutions.
* Integration with live fare databases could enhance accuracy.
* User feedback mechanisms could guide iterative improvements and feature additions.
* Training on larger datasets would be useful in improving model accuracy.

The app.py and features.py files outline the backbone of the web application, detailing the workflow from user input to fare prediction. They highlight the use of preprocessing for user inputs, model loading, and prediction execution, all facilitated through an interactive Streamlit interface. The setup is designed to be straightforward, ensuring that users can get fare estimates with minimal clicks and waiting time.

short dash

# Collaboration

## Individual Contributions

|  |  |
| --- | --- |
| Member | Contribution |
| Chanthru Vimalasri | * Data Preparation Notebook (consolidated all pre-processing steps, creating a base for us to create our models from) * XGBoost Model (Hyperparameter Tuning) * Streamlit App – wrote code for app.py and features.py. * Final Report Writing |
| Tarun Gupta | * Random Forest Model * Light GBM Model * Final Report Writing |
| Nipesh Shreshta | * WideDeepModel * Linear Regression * Final Report Writing |
| Zhicong Chen | * ARIMA Time Series Model * Final Report Writing |

## Group Dynamic

Group 15 established a healthy group dynamic. Communication was established early, first via email and then WhatsApp to streamline discussion about the project. Base data cleaning code was created and shared and to optimize time and resources, we individually constructed our machine learning models in line with our interests and overall goals. We then re-convened to discuss the integration of the Streamlit app and writing of the final written report. Questions were relayed via WhatsApp and group members did their best to answer these doubts.

## Ways of Working Together

Our team follows agile project management methodologies to effectively oversee our projects. While it's challenging to conduct regular in-person meetings due to scheduling constraints, we maintain constant communication through online collaboration tools. This enables us to engage in real-time discussions on project-related issues and ideas, ensuring that information flows smoothly and rapidly among team members.

In terms of tracking project progress and assigning tasks, we have a well-structured approach. Although we don't rely on specific project management tools, our team members are diligent about understanding and completing their tasks. This level of commitment ensures that everyone is clear about their responsibilities and objectives. Additionally, we utilize GitHub as a code version control and collaboration platform, simplifying the sharing and management of code among team members.

When it comes to teamwork, our core principles include open communication and collaboration. We had weekly check ins. Every team member was encouraged to contribute ideas, make suggestions, and participate in the decision-making process. This transparent communication and collaborative approach strengthen team cohesion and enhance work efficiency, allowing us to achieve our project's common goals effectively.

## Issues Faced

* Identify any challenges, issues, or obstacles encountered during the project.
* Describe how these challenges were addressed and resolved within the team.
* Discuss any lessons learned or recommendations for future group collaborations.

Instructions: Identify and discuss any challenges, issues, or obstacles that the team encountered during the project. Describe how these challenges were addressed and resolved within the team, including any strategies or actions taken to overcome them. Reflect on the lessons learned and provide recommendations for improving future group collaborations.

* Large dataset
* adjusting code to align for streamlit
* miscommunication
* streamlit app – issues understanding how to write code and integrate implementation

short dash

# Conclusion

The Local Travel Airfare Estimator project aimed to provide a reliable estimate of flight costs, and it has done so effectively, demonstrating the practical use of machine learning models in everyday decision-making tools. We've built a web application that simplifies the complexity behind airfare predictions into a user-friendly interface, empowering users to estimate travel costs with ease.

**Project Goals and Stakeholder Satisfaction:**

The web app has met its goal to offer a straightforward tool for airfare estimation, aligning with user expectations for simplicity and efficiency. We've provided stakeholders with a functional product that aligns with their needs for accessible travel information.

**Reflection on Success:**

While we set out to create a helpful tool, and by many measures, we succeeded, the true value of the project lies in its potential for growth and improvement.

**Recommendations for Future Development:**

Future enhancements could include:

* Updating the data feeding the predictive models to reflect current trends.
* Expanding the app's capacity to handle more simultaneous users.
* Refining the models with new data to improve prediction accuracy.
* Gathering user feedback for iterative development, ensuring the tool remains aligned with user needs.

Ultimately, the project has laid down a solid groundwork for a potentially more robust system in the future, and with continued attention and development, it can become an even more valuable resource for travelers and industry professionals alike.

short dash