EXPERIMENT REPORT

Student Name	Thirada Tiamklang – Student C
Project Name	TensorFlow's Decision Forests (TF-DF) algorithm
Date	09 November 2023
Deliverables	<Tiamklang_Thirada-14337188-tfdf.ipynb> <Gradient Boosted Trees> <Random forest model> https://github.com/wongwara/flight-streamlit-at3/blob/main/flight-prediction/notebooks/TT_notebooks/tensor_flow.ipynb

1. EXPERIMENT BACKGROUND

1.a. Business Objective

The project aiming to build a data product that empowers users in the USA to make accurate estimates of their local travel airfare and deploy it on Streamlit application. Travellers struggle with predicting flight costs due to factors like dynamic pricing, fare classes, and seasonal variations, making it challenging to plan and budget trips effectively.

Stakeholder	**Requirement**
Users	Users seek accurate, personalised airfare estimates that help them plan and optimise their travel while staying within their budget.
Travel Platforms/ Agencies	Travel platforms offer travel planning tools to attract and retain users, increasing engagement and revenue.
Airlines	Airlines may benefit from more informed travellers who can make cost-effective booking decisions.

Table 1: Project's stakeholder and requirements

1.b. Hypothesis

With the business objective of building a data product that accurately predicts the total fare, the hypothesis for this experiment is that different models and architectures of TensorFlow's Decision Forests (TF-DF) will result in varying MSE scores.

1.c. Experiment Objective

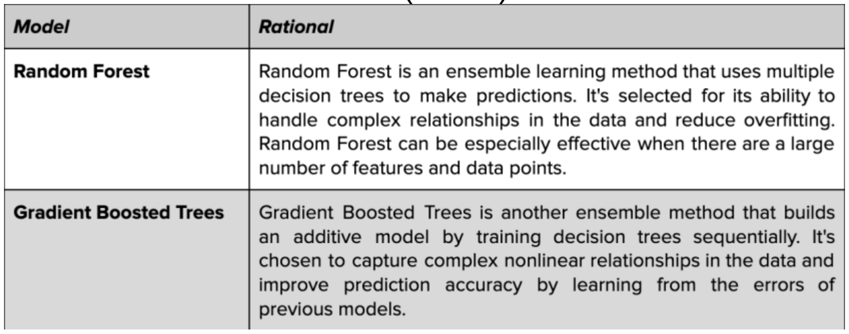
The primary objective of this experiment is to achieve accurate airfare estimation for local travel in the USA. This will be accomplished by developing a data product that employs the **TensorFlow Decision Forests (TF-DF) algorithm** to train the data and provide reliable fare predictions.

The anticipated outcome is the identification of the best model with the lowest MSE score, which will be deployed for making predictions in the Streamlit application.

2. EXPERIMENT DETAILS	
2.a. Data Preparation	**1. Merging Datasets:** Merge datasets from various airports into a single dataframe for exhaustive analysis. **2. Converting Travel Duration:** Convert travel duration from a string format into seconds for consistency and easy analysis. **3. Segment Splitting:** The segment columns are split into lists. **4. Total Segment Duration:** Calculate the total duration of segments and create a new column named "totalDuration (segment)." **5. Travel Layover:** A new column, "travelLayover," represents the difference between travel duration and the total segment duration. **6. Transit Airport Code:** Extract and store the airport code of transit locations in a new column named "transitAirportcode." **7. All Airport Codes:** We collect all airport codes, including departure, arrival, and transit, and store them in a new column labelled "All airport." **8. Date Transformation:** The date column serachDate and flightDate is transformed into separate columns for day, month, and year. **9. Departure Time Extraction:** Extract the departure time in Unix epoch seconds format and separate it into individual columns for hours, minutes, and seconds. **10. Handling Null Values:** To address missing data, fill null values with the mean of the respective columns. **11. Feature selection:** The selected features, characterised by both high correlation with the target variable (above |0.24|) and low collinearity (remained below 0.7.), encompass *'totalTravelDistance,' 'isNonStop,' 'isBasicEconomy,' 'startingAirport,' 'destinationAirport,' 'segmentsCabinCode,' 'flightDate_day,' 'flightDate_month,' 'flightDate_year,' 'DepartTime_hour,' 'DepartTime_minute,' 'DepartTime_second,' and 'totalFare.'*
2.b. Feature Engineering	**1. feature engineering for all experiments in this project:** **1.1 Factorising Segment Columns:** The segment columns were factorised to enhance their usability in modelling. **1.2 Label Encoding 'SegmentCabinCode':** Applied label encoding to the 'segmentCabinCode' column, converting categorical data into a numerical format for analysis. **1.3 Encoding Categorical Columns:** Other categorical columns were also encoded using label encoding, ensuring consistency in handling categorical data. **1.4 Scaling Numeric Columns:** Numeric columns were scaled using Standard Scaler, making them compatible with the modelling process. **1.5 Data splitting:** The dataset was divided into training, testing, and validation sets with an 80:20 ratio. It is important to note that all the experiments used the same dataset after this split, allowing for a fair comparison of the best model later **2. feature engineering for TensorFlow Decision Forests (TF-DF)** **2.1 Convert to tensor flow datasets:** With batch size=100, using \`from_tensor_slices\` to convert NumPy arrays into tensors

2.c. Modelling

Algorithm: TensorFlow Decision Forests (TF-DF) with Keras



Model	***Rational***
Random Forest	Random Forest is an ensemble learning method that uses multiple decision trees to make predictions. It's selected for its ability to handle complex relationships in the data and reduce overfitting. Random Forest can be especially effective when there are a large number of features and data points.
Gradient Boosted Trees	Gradient Boosted Trees is another ensemble method that builds an additive model by training decision trees sequentially. It's chosen to capture complex nonlinear relationships in the data and improve prediction accuracy by learning from the errors of previous models.

Model fitting: train_dataset, epochs=1, validation_data=validation_dataset
Model compile: metrics=["mean_squared_error"]
Model Architecture:

- RandomForestModel(task=tfdf.keras.Task.REGRESSION)
- RandomForestModel(task=tfdf.keras.Task.REGRESSION, num_trees=100, max_depth=10)
- GradientBoostedTreesModel(task=tfdf.keras.Task.REGRESSION)

Model Evaluation: Evaluate train and test dataset and output loss as `MSE score`

3. EXPERIMENT RESULTS

3.a. Technical Performance

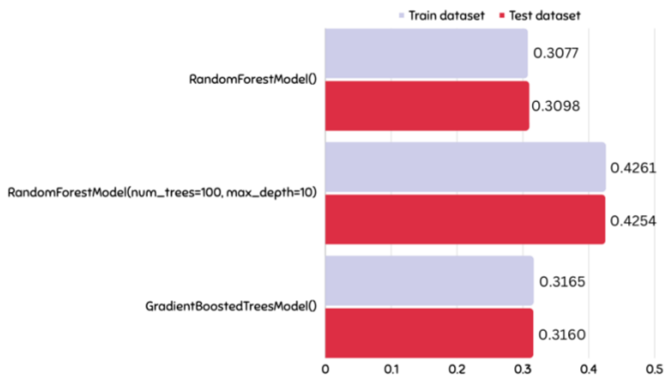


Figure 1: The bar chart of MSE score for TensorFlow Decision Forest (TF-DF)

The bar chart above represents the MSE score results on the training and testing sets for each model. It can be seen that the Random Forest model, with default hyperparameters, provided the lowest MSE score of 0.3098, which we will define as the best model among these three.

3.b. Business Impact

The development of a data product for estimating local airfare with MSE scores0.3098 has several significant business implications and benefits that show that this estimate is accurate.

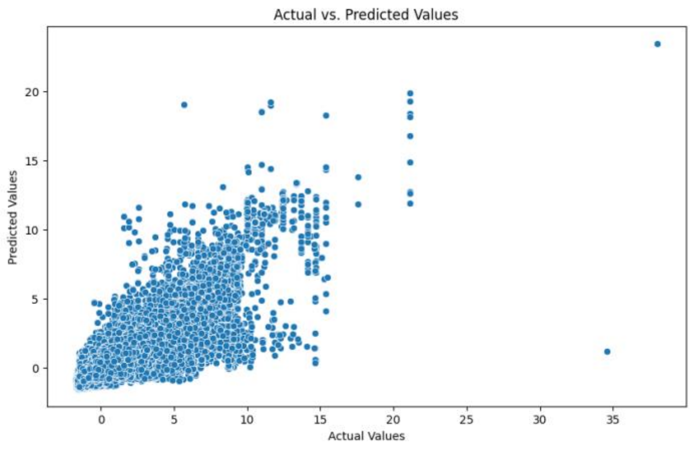
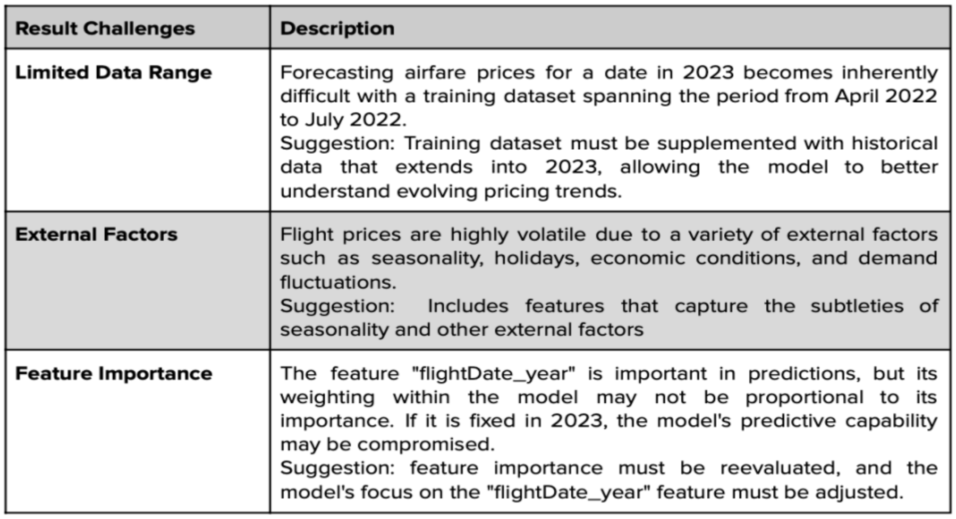


Figure 2: Model prediction comparison for Random Forest with defualt

The model has a significant impact and benefit on users by providing accurate and reliable airfare estimates for their travel planning needs. These advantages include cost savings, time savings, and increased customer satisfaction, which can lead to increased customer retention and loyalty. The model helps to address the identified challenges in budget planning and fare analysis, while also leveraging opportunities for better-informed travel decisions.

3.c. Encountered Issues



Result Challenges	Description
Limited Data Range	Forecasting airfare prices for a date in 2023 becomes inherently difficult with a training dataset spanning the period from April 2022 to July 2022. Suggestion: Training dataset must be supplemented with historical data that extends into 2023, allowing the model to better understand evolving pricing trends.
External Factors	Flight prices are highly volatile due to a variety of external factors such as seasonality, holidays, economic conditions, and demand fluctuations. Suggestion: Includes features that capture the subtleties of seasonality and other external factors
Feature Importance	The feature "flightDate_year" is important in predictions, but its weighting within the model may not be proportional to its importance. If it is fixed in 2023, the model's predictive capability may be compromised. Suggestion: feature importance must be reevaluated, and the model's focus on the "flightDate_year" feature must be adjusted.

4. FUTURE EXPERIMENT

4.a. Key Learning

Reflecting on the experiment results, the Random Forest model with default settings achieved the lowest MSE score of 0.3098, making it the best model among the three. This high accuracy has several benefits, including cost savings, time efficiency, and increased customer satisfaction. It addresses budget planning challenges and improves fare analysis, emphasizing the importance of our current approach and potential further experimentation to enhance the model.

4.b. Suggestions / Recommendations

The best model from this experiment will be deployed on Streamlit application to predict the total airfare along with other experiments. However, there are some suggestions that can be work in the future:

1. Feature Enhancement: Improve features, such as detailed airport data and user preferences.
2. Hyperparameter Tuning: Optimize model settings for better accuracy.
3. Ensemble Approach: Combine models for higher prediction quality.
4. User Interface: Enhance Streamlit for a user-friendly experience.
5. Regular Updates: Keep models current with changing conditions.
6. Feedback Loop: Gather user insights for continuous improvement.
7. Market Expansion: Consider extending services globally.
8. Airlines Collaboration: Partner with airlines for real-time deals.