EXPERIMENT REPORT

Student Name	Thanchanok Phuawiriyakul
Project Name	AT3 - ML Data Product
Date	9 November 2023
Deliverables	<Phuawiriyakul_Thanchanok-24582239-keras.ipynb> <Keras> <https://github.com/wongwara/flight-streamlit-at3/blob/main/flight-prediction/notebooks/TP_notebooks/Phuawiriyakul_Thanchanok-24582239-keras.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 objective of this project is to create models for the development of a data product aimed at assisting users in the USA in improving their accuracy when estimating local travel airfare. Users will have the capability to input their trip specifics, and the application will provide predictions for their expected flight fares.

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.

1.b. Hypothesis	• The performance of a fare prediction model built with Keras and KerasTuner will give a similar performance, as assessed by the MSE score.
1.c. Experiment Objective	• The key objective of this endeavour is to accurate Airfare Estimation by developing a data product that can accurately estimate local travel airfare for users in the USA, providing reliable fare 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	1. Data preprocessing ○ Merging Datasets: Merge datasets from various airports into a single dataframe 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. 2. Feature selection: Identifying variables that exhibited a correlation with the target variable (total fare) above |0.24| 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
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. • 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.c. Modelling	• Keras: create model with three layers and Compile the model with MSE score
	  ○ The input layer: 12 neurons, Input data shape (12,) and The ReLU activation function
	  ○ The hidden layer: 8 neurons and The ReLU activation function
	  ○ The output layer: 1 neuron
	• KerasTuner: defined function to create keras model and tuner with randomSearch
	  ○ Create a function with tunable hyperparameters for the number of units in a hidden layer.
	  ○ The model uses mean squared error (MSE) as the loss function to optimise for prediction. ()
	  ○ Keras Tuner with random search: finding best hyperparameters.
	  ■ Tuner with the objective is 'val_loss' or mse
	  ■ max_trails with 5

---

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

From the modelling, we evaluate model performance using MSE for both keras and kerasTuner

	MSE score	
	Keras	KerasTuner
Train set	0.49433	0.47212
Test set	0.49395	0.47097

- Both Keras and KerasTuner do not reveal overfitting, as seen by MSE score variance between the train and test sets.
- Keras shows a slight performance compared to KerasTuner.

3.b. Business Impact	• Keras can accurately predict ticket fares, contributing to more informed business decisions for users. • As the correlation score indicates, identifying features with lower impact on the target variable, we select only high correlation for training.
3.c. Encountered Issues	• First, we successfully tried to save the Keras model using joblib in Jupyter Notebook (saving and loading for predictions). However, it does not work when deploying the joblib model on Streamlit. To fix this, we changed to save the model in the ".keras" format for improved compatibility.

---

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 Keras model shows accurate total fare predictions (lower MSE score). • Performance optimization for the Keras model, with increasing the number of epochs getting improved model accuracy.
4.b. Suggestions / Recommendations	• Training the keras model with different features and comparing model performance on MSE. • Adjusting hyperparameters on both keras: adding more hidden layers or changing different parameters to compare performance.