EXPERIMENT REPORT

Student Name	Vishal Raj (14277627)
Project Name	AT3 - Data Product with Machine Learning
Date	9th November, 2023
Deliverables	main_vishal_raj.ipynb train_model_vishal_raj.py best_model-vishal_raj Wide and Deep Neural Network https://github.com/vishalraj247/Flight_Fare_Prediction.git https://github.com/vishalraj247/Flight_Fare_App.git

---

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 business aims to establish a robust flight fare prediction system to inform pricing strategies. Such a system is designed to provide airlines with a tool for better revenue management through dynamic pricing, help travel agencies craft competitive package deals, and assist travellers in budgeting by predicting fare trends. An accurate prediction model could significantly impact revenue streams and customer satisfaction by minimising flight costs' uncertainty.
1.b. Hypothesis	My hypothesis posits that the wide and deep neural network, with its hybrid structure, is superior for the task of predicting flight fares compared to traditional models. This theory is premised on the belief that this architecture will handle both the feature interaction intricacies and the unique hierarchical patterns within the fare data effectively, thus achieving a lower RMSE and MAE on the test set.

1.c. Experiment Objective	The anticipated outcome is achieving precise fare predictions with a lower RMSE and an MAE as possible for the test dataset. These parameters were chosen based on industry standards and previous model performances. The experiment seeks to assess the model's predictive strength, identify any performance gaps, and determine the feasibility of implementing more complex modelling techniques if necessary.

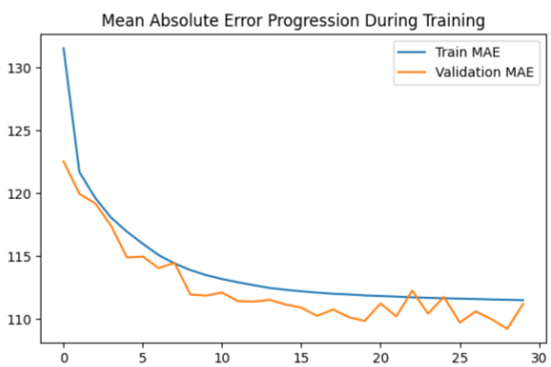
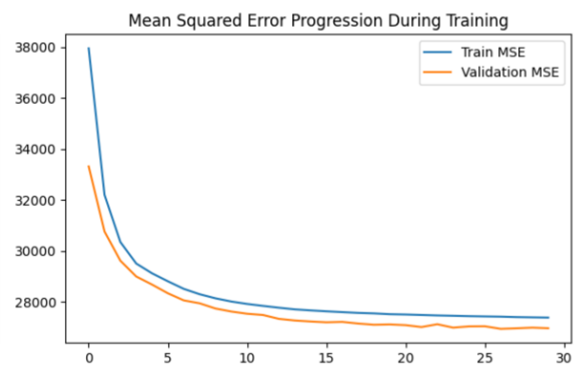
---

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	Data preparation commenced with consolidating multiple zipped data sources into a singular CSV format using the make_dataset.py script, ensuring data integrity for a comprehensive analysis. The data_preprocessor_dl.py script further refined the dataset through custom transformations such as the 'DateFeatureExtractor', which integrated date-related features and weekend flags, highlighting demand-induced fare variations. The dataset had columns with multiple values separated by "||," I used a split and explode function to split the value into numerous rows. Categorical variables, including airport codes and cabin classes, were numerically encoded to preserve the richness of the dataset while accounting for potential new or unknown categories. The 'DataPreprocessor' class played a pivotal role in merging datasets and unfolding complex travel routes into a more analysable form. Our strategy for managing outliers involved normalising fare values to their modal group fares to stabilise the prediction target against extreme fare variations. We employed type optimisation through downcasting to improve computational efficiency during model training. The pre-processing pipeline was augmented with imputation, scaling, and encoding, as well as the integration of context, such as median travel statistics, to aid in the accuracy of fare estimation. The final step involved diligently archiving the processed data, ensuring the predictive system is updated, accurate, and equipped for ongoing model refinement and data assimilation. This level of careful data management is instrumental in supporting the advanced analytics capabilities of our flight fare prediction model.

2.b. Feature Engineering	For feature engineering, we considered temporal features such as day of the week, time of day, is weekend or a weekday, which previous studies suggested could impact fares. We employed embedding layers for categorical features to reduce dimensionality and improve model interpretability. Some features, such as airline-specific codes and names, were excluded to prevent overfitting and ensure the model's applicability across various input features which are later to be provided through the Streamlit app to predict.
2.c. Modelling	The model architecture consisted of a hybrid approach combining a 'wide' linear model with a 'deep' neural network to capitalise on both raw input features and high-level abstractions. The wide component processed seven direct input features. In contrast, the deep component utilised a multi-layer perceptron with dense layers of 256 and 128 neurons, respectively, using ReLU activation functions and a dropout rate of 0.2 to prevent overfitting. The categorical features—‘startingAirport’, ‘destinationAirport’, and ‘segmentsCabinCode’—were embedded into dense representations, with their dimensions carefully calculated to avoid overly complex models. Each categorical feature was limited to a maximum embedding size of 50 to ensure model simplicity and generalisation. I executed the model training using custom data pre-processing methods that segmented data into appropriate wide and deep inputs, including converting boolean columns to integers for TensorFlow processing. The training set was further split into validation and test sets, with the latter reserved for final model evaluation. Model compilation leveraged an Adam optimiser with an optional warmup step schedule to improve convergence. The learning rate was initially set low (0.01) and adjusted dynamically during training. During training, I employed several strategies to enhance performance and prevent overfitting, including early stopping with the patience of three epochs and restoration of the best weights, learning rate reduction on a plateau, and model checkpointing to preserve the best-performing model iteration. The ‘train_model’ function encapsulated the training process, which involved batching (32768) and callbacks for early stopping, reduced learning rate on a plateau, and model checkpointing. I configured the callbacks to monitor validation loss and adjust the learning rate or stop training based on performance, effectively balancing model fit and computational efficiency. This comprehensive training regimen aimed to develop a robust predictive model focusing on the accuracy of fare predictions, as measured by the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics, both of which were crucial considerations for the model's performance evaluation. I decided against a full grid search for hyperparameter tuning due to time and resource constraints but noted this as an area for potential future exploration.

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	1st Experiment: The model's technical performance, with an RMSE of 177.048 and MAE of 120.444 on the test dataset, suggests that it is capable but still has room for improvement, especially in handling outliers and rare events that cause fare spikes. 2nd Experiment: The model's technical performance, with an RMSE of 164.08 and MAE of 110.55 on the test dataset, suggested that it improved a little bit after cleaning the features more properly with duplicates and selecting better parameters. 
3.b. Business Impact	From a business standpoint, the model's current performance can provide value in enhancing pricing strategies, although the relatively higher RMSE may lead to less competitive pricing in some scenarios. The model's predictions can be integrated into current systems with caution, using a phased approach to monitor and improve prediction accuracy with ongoing data collection.
3.c. Encountered Issues	The main issues encountered revolved around computational efficiency and managing the extensive requirements of deep learning models. One strategy adopted was prioritising the depth of feature learning over extensive hyperparameter tuning. We also acknowledged the need to enhance our data pre-processing in future experiments to handle anomalies more effectively. To reduce the model training time, batch size was also increased significantly, keeping in mind the memory of the system on which the model was trained.

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 key learning from this experiment is the potential viability of the wide and deep model for flight fare prediction, with certain limitations in its current form. Continuous model refinement and incorporation of new data are critical for improving the model's accuracy and reliability.
4.b. Suggestions / Recommendations	Future recommendations include increasing the diversity and volume of data to train more generalized models, implementing more sophisticated hyperparameter tuning when resources allow, and conducting ablation studies to understand feature contributions. If the model stability is confirmed, we suggest deploying it in a controlled production environment with real-time monitoring and performance tracking systems.