

EXPERIMENT REPORT

Student Name	Group 3
Project Name	AT3 - Data Product with Machine Learning
Date	14 th November, 2023
Deliverables	Wide and Deep Neural Network

1. EXPERIMENT BACKGROUND

<p>1.a. Business Objective</p>	<p>The successful implementation of a flight fare prediction ML model can contribute to achieving a combination of these business objectives, depending on the priorities and strategies of the airline or travel-related business. The model's ability to analyze complex data and provide actionable insights contributes to strategic decision-making in the highly dynamic and competitive airline industry.</p> <p>1. Optimizing Revenue: Objective: Maximize revenue by setting optimal ticket prices. How the Model Helps: The ML model can analyze historical data, market trends, and various factors influencing ticket prices to recommend pricing strategies that optimize revenue. This might involve dynamic pricing adjustments based on demand, seasonality, and competitor pricing.</p> <p>2. Dynamic Pricing Strategies: Objective: Implement dynamic pricing strategies to respond to market changes. How the Model Helps: Analyze real-time data to adjust prices dynamically. This ensures that the pricing strategy aligns with current market conditions, demand fluctuations, and other relevant factors.</p> <p>3. Forecasting Demand: Objective: Anticipate and meet demand for different routes and times.</p>
	<p>How the Model Helps: Predict future demand based on historical patterns, allowing airlines to optimize seat inventory and adjust pricing to match expected demand. This helps avoid overbooking or underutilization of resources.</p> <p>4. Revenue Management: Objective: Efficiently manage capacity and revenue across different flights. How the Model Helps: The ML model aids in optimizing the allocation of seats on flights, considering factors like seasonality, holidays, and special events. This ensures that capacity is managed effectively to maximize revenue.</p> <p>5. Operational Efficiency: Objective: Enhance operational efficiency and resource utilization. How the Model Helps: By accurately predicting demand and adjusting pricing accordingly, airlines can optimize their operations, allocate resources efficiently, and reduce waste.</p>

1.b. Hypothesis

When formulating hypotheses for a wide and deep neural network flight prediction ML model, I'm essentially making educated guesses or assumptions about how certain features and model components might impact the model's performance.

Hypothesis 1: The Wide Component Captures Linear Relationships

Justification: The wide component of the model is designed to capture linear relationships between features. I hypothesize that factors such as day of the week, time of day, and seasonality have a linear impact on flight fares.

Hypothesis 2: The Deep Component Captures Non-linear and Complex Patterns

Justification: The deep component of the model can capture non-linear and complex relationships. I hypothesize that the deep neural network will learn intricate patterns in the data that go beyond simple linear correlations, such as the interaction between departure city, airline, and historical pricing trends.

Hypothesis 3: Feature Crosses Improve Model Performance

Justification: Creating feature crosses (interactions between features) in the wide component can help capture dependencies between different factors. I hypothesize that feature crosses, such as combining departure city and airline, will enhance the model's ability to predict flight fares accurately.

Hypothesis 4: Categorical and Numerical Features Contribute Complementarily

Justification: The wide and deep model is designed to handle both categorical and numerical features. I hypothesize that the combination of information from categorical features (e.g., airline, departure city) and numerical features (e.g., distance, time) will contribute complementarily to the model's predictive power.

Hypothesis 5: The Model Adapts to Dynamic Pricing Changes

Justification: I hypothesize that the model will effectively adjust to changes in market conditions, allowing for accurate predictions in a dynamic pricing environment.

Hypothesis 6: Model Performance Improves with Sufficient Training Data

Justification: Neural networks often benefit from a large amount of training data. I hypothesize that as the amount of historical pricing and relevant feature data increases, the model's performance in predicting flight fares will improve.

By testing and analyzing the model's performance against these hypotheses, we can refine and improve the model for more accurate flight fare predictions.

1.c. Experiment Objective

The experiment objective for my model involves setting clear goals and expectations for evaluating the model's performance and capabilities.

1. Assess the model's ability to accurately predict flight fares.
 2. Identify the most influential features in predicting flight fares.
 3. Compare the performance of the wide and deep components separately. Evaluate if the deep component improves prediction accuracy over a purely wide (linear) model.
 4. Compare the model's performance with and without feature crosses. This helps determine if the interactions between features significantly enhance predictive capabilities.
 5. Assess how well the model generalizes to new, unseen data.
 6. Evaluate the computational efficiency of training and inference.
-

2. EXPERIMENT DETAILS

2.a. Data Preparation

The `make_dataset.py` script was used to consolidate multiple zipped data sources into a single CSV format, ensuring data integrity for a thorough analysis. The `data_preprocessor_dl.py` script refined the dataset further with custom transformations like the 'DateFeatureExtractor,' which combined date-related features and weekend flags to highlight demand-induced fare variations. Columns with multiple values were separated by "||," in the dataset. To divide the value into multiple rows, I used the `split` and `explode` functions.

To preserve the richness of the dataset while accounting for potential new or unknown categories, categorical variables such as airport codes and cabin classes were numerically encoded. The 'DataPreprocessor' class was critical in merging datasets and unravelling complex travel routes into a more readable format. Our approach to dealing with outliers to stabilise the prediction target against extreme fare variations, fare values were normalised to their modal group fares.

To improve computational efficiency during model training, we used type optimisation via downcasting. To improve the accuracy of fare estimation, the pre-processing pipeline was enhanced with imputation, scaling, and encoding, as well as the incorporation of context, such as median travel statistics.

The final step was to meticulously archive the processed data, ensuring that the predictive system is always up to date, accurate, and ready for ongoing model refinement and data assimilation. This level of data management is critical in supporting our flight fare prediction model's advanced analytics capabilities.

<p>2.b. Feature Engineering</p>	<p>We considered temporal features such as day of the week, time of day, whether it is a weekend or a weekday for feature engineering, which previous studies suggested could impact fares. To reduce dimensionality and improve model interpretability, we used embedding layers for categorical features. Some features, such as airline-specific codes and names, were excluded to avoid overfitting and ensure the model's applicability across various input features that will be provided later for prediction via the Streamlit app.</p>
---------------------------------	--

<p>2.c. Modelling</p>	<p>To capitalise on both raw input features and high-level abstractions, the model architecture used a hybrid approach that combined a 'wide' linear model with a 'deep' neural network. The broad component handled seven direct input features. To avoid overfitting, the deep component used a multi-layer perceptron with dense layers of 256 and 128 neurones, respectively, with ReLU activation functions and a dropout rate of 0.2.</p> <p>The categorical features—'startingAirport,' 'destinationAirport,' and 'segmentsCabinCode'—were embedded into dense representations, with their dimensions calculated carefully to avoid overly complex models. To ensure model simplicity and generalisation, each categorical feature was limited to a maximum embedding size of 50. I used custom data pre-processing methods to segment data into appropriate wide and deep inputs, including converting boolean columns to integers for TensorFlow processing. The training set was divided further into validation and test sets, with the latter being reserved for final model evaluation. To improve convergence, model compilation used an Adam optimiser with an optional warmup step schedule. The learning rate was initially set low (0.01) and dynamically adjusted during training.</p> <p>Several strategies were used during training to improve performance and avoid overfitting, including early stopping with three epochs of patience and restoration of the best weights, learning rate reduction on a plateau, and model checkpointing to preserve the best-performing model iteration.</p> <p>The training process was encapsulated in the 'train_model' function, which included batching (32768) and callbacks for early stopping, reduced learning rate on a plateau, and model checkpointing. I set up the callbacks to monitor validation loss and adjust or stop training based on performance, effectively balancing model fit and computational efficiency.</p> <p>This extensive training regimen aimed to develop a robust predictive model with a focus on fare prediction accuracy as measured by the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics, both of which were critical considerations for the model's performance evaluation. Due to time and resource constraints, I decided against a full grid search for hyperparameter tuning, but I did note this as an area for potential future exploration.</p>
-----------------------	---

3. EXPERIMENT RESULTS

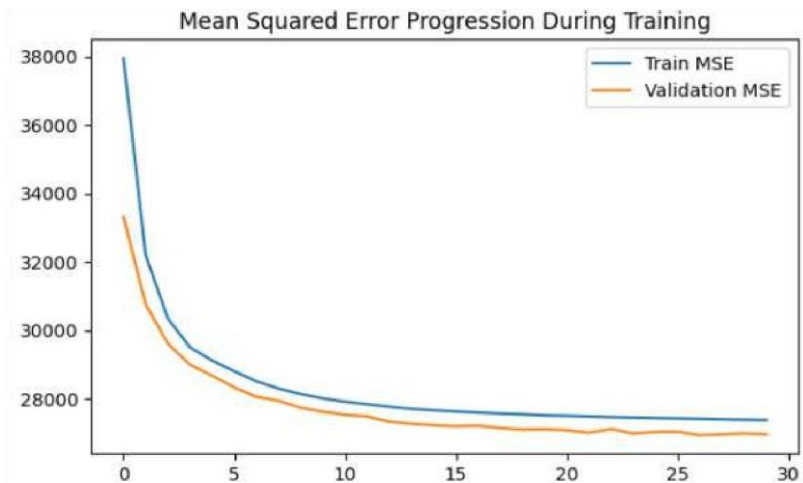
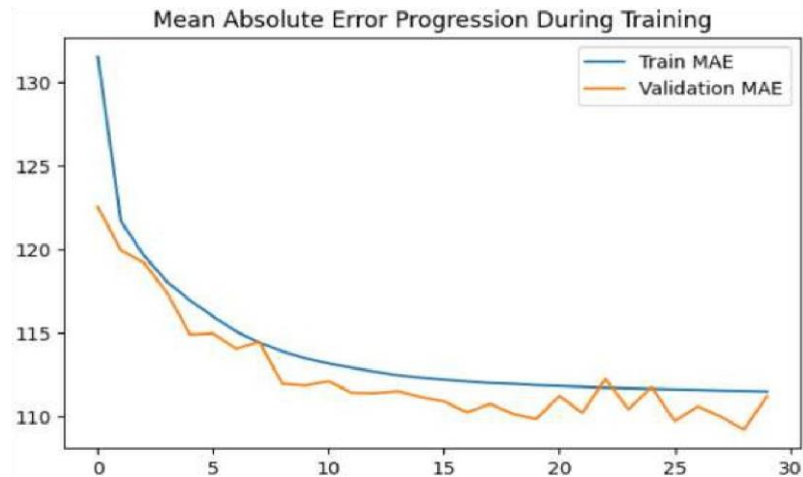
3.a. Technical Performance

1st Experiment:

The model's technical performance on the test dataset, with an RMSE of **177.048** and MAE of **120.444**, suggests that it is capable but still has room for improvement, particularly in dealing with outliers and rare events that cause fare spikes.

2nd Experiment:

The model's technical performance on the test dataset, with an RMSE of **164.08** and MAE of **110.55**, suggested that it improved slightly after cleaning the features more thoroughly for duplicates and selecting better parameters.



3.b. Business Impact	<p>It is important to note that the specific business impact will be determined by the model's effectiveness, data quality, and how well the organization integrates the model into its decision-making processes. By predicting optimal pricing strategies based on historical data, market trends, and real-time factors, the model can help to maximize revenue. This leads to more informed decisions about how to adjust ticket prices to meet demand. A wide and deep neural network flight prediction ML model must be monitored, refined, and aligned with broader business strategies on a regular basis to maximize its positive impact.</p>
-----------------------------	--

3.c. Encountered Issues	<p>The main problems encountered were with computational efficiency and managing the extensive requirements of deep learning models. One approach was to prioritize feature learning depth over extensive hyperparameter tuning. We also recognized the need to improve our data pre-processing in future experiments to better handle anomalies. To reduce model training time, the batch size was increased significantly while keeping the memory of the system on which the model was trained in mind.</p> <p>While wide and deep neural networks are powerful models, there can be several challenges or issues when developing our model using this architecture. Some common issues include:</p> <p>Feature Engineering Challenges:</p> <p>Issue: Identifying and creating relevant features, especially for categorical variables, can be challenging. Incorrect feature representations may hinder model performance.</p> <p>Mitigation: Invest time in thorough feature engineering, considering domain knowledge and exploring different encoding techniques. Experiment with feature crosses to capture interactions between variables.</p> <p>Hyperparameter Tuning:</p> <p>Issue: Selecting appropriate hyperparameters for both the wide and deep components can be challenging. Poor choices may result in suboptimal model performance.</p> <p>Mitigation: Conduct systematic hyperparameter tuning using techniques like grid search or random search. Consider leveraging automated hyperparameter optimization tools.</p> <p>Computational Complexity:</p> <p>Issue: Training wide and deep neural networks can be computationally intensive, especially with large datasets and complex architectures. This may pose challenges for resource-constrained environments.</p> <p>Mitigation: Optimize the model architecture, use hardware accelerators (GPUs or TPUs), and explore distributed training techniques to improve computational efficiency.</p> <p>Addressing these issues requires a combination of careful model design, data preparation, and ongoing monitoring and refinement. Regular evaluation and adaptation of the model based on real-world performance are essential for building a robust and effective wide and deep neural network flight prediction ML model.</p>
--------------------------------	---

4. FUTURE EXPERIMENT

4.a. Key Learning

The combination of a wide (linear) component and a deep (non-linear) component allows the model to capture both simple and complex patterns. Striking the right balance between these components is crucial for optimal performance. Thorough feature engineering is essential. Categorical feature encoding, creation of feature crosses, and careful consideration of temporal aspects contribute significantly to the model's ability to understand and predict flight fares accurately. Proper hyperparameter tuning is essential for model performance. Systematic exploration of hyperparameter space and leveraging tools for automated hyperparameter optimization contribute to finding optimal configurations. Monitoring key performance indicators (KPIs) related to revenue, customer satisfaction, and operational efficiency helps gauge the model's effectiveness.

These key learnings highlight the multidimensional nature of deploying a wide and deep neural network flight prediction ML model. From technical considerations to business impact and ethical considerations, successful implementation requires a holistic and iterative approach.

4.b. Suggestions Recommendations

/ Future recommendations include expanding the diversity and volume of data used to train more generalized models, implementing more sophisticated hyperparameter tuning when resources allow, and conducting ablation studies to better understand feature contributions. We recommend deploying the model in a controlled production environment with real-time monitoring and performance tracking systems if its stability is confirmed.