Porter: Neural Networks Regression Business Case

❖ Topic: Neural Networks in Delivery Time Prediction
❖ Duration: 1 week

Why this case study?

From the company’s perspective:

- Porter, India's largest marketplace for intra-city logistics, is revolutionizing the delivery sector with technology-driven solutions.
- This case focuses on leveraging neural networks to accurately predict delivery times, a critical aspect of customer satisfaction in logistics.
- With a dataset encompassing various aspects of orders and deliveries, Porter aims to refine its delivery time estimations.
- Analyzing this dataset can provide significant insights into delivery dynamics, efficiency bottlenecks, and optimization opportunities.
- The insights obtained can enhance Porter's operational efficiency, ensuring timely deliveries and improving driver-partner allocation.

From the learner’s perspective:

- This case presents an opportunity to engage with real-world logistics data, understanding the nuances of delivery time prediction.
- Neural Networks, a powerful tool in regression analysis, are crucial in predicting continuous outcomes like delivery times.
- Participants will develop competencies in data preprocessing, neural network architecture, and model evaluation, key skills in the data science field.
- The case encourages developing practical solutions, reinforcing the ability to implement data-driven strategies in the logistics sector.

Dataset Explanation: Porter Data

Each row in this dataset corresponds to a unique delivery, and each column represents a feature critical in predicting delivery times. The dataset 'Porter Data' includes the following variables:

1. market_id: An integer ID indicating the market area of the restaurant.
2. created_at: Timestamp of when the order was placed.
3. actual_delivery_time: Timestamp of when the order was delivered.
4. store_primary_category: Category classification of the restaurant.
5. order_protocol: Numeric code representing the mode of order placement (e.g., through Porter, direct call, pre-booking, third-party platform).
6. total_items_subtotal: A combined feature detailing the total number of items in the order and the final price of the order before taxes and fees.
7. num_distinct_items: Count of different items in the order.
8. min_item_price: Price of the least expensive item in the order.
9. max_item_price: Price of the most expensive item in the order.
10. total_onshift_partners: Number of delivery partners on duty when the order was placed.
11. total_busy_partners: Number of delivery partners busy with other tasks at the order placement time.
12. total_outstanding_orders: Total count of orders pending at the time.

Note: The target variable for this study, 'estimated_delivery_time', is to be derived from the difference between 'created_at' and 'actual_delivery_time'.

What is Expected?

As a data scientist at Porter, your task is to analyze the dataset to accurately predict delivery times for different orders. Your primary goal is to build a regression model using neural networks, evaluate its performance, and provide insights for optimizing delivery operations.

Submission Process:

Once you’ve completed the case study...

- Compile your findings and the entire process in a Jupyter Notebook.

- In the notebook, ensure that you:
    - Display the Python code for all your analysis, model building, and evaluation.
    - Include visualizations such as plots, scatter diagrams, and error distributions that support your findings.
    - Provide valuable insights derived from the data analysis, and suggest actionable recommendations for Porter to improve their delivery time predictions.
- Convert your Jupyter Notebook into a PDF (Save as PDF using the Chrome browser’s Print command).

- Upload the PDF on the designated platform as per the submission guidelines.

Note: Once submitted, you won't be able to edit your submission.

General Guidelines:

- Approach this as a real-world scenario, akin to the challenges faced by data scientists in the logistics industry.
- It's normal to face difficulties:
    - Revisit the problem statement to keep your analysis aligned with the objectives.
    - Break down complex tasks into smaller segments.
    - Search online, consult documentation, and solve problems as they arise—problem-solving is key in data science.
    - Collaborate with peers in discussion forums for different perspectives and solutions.
    - Review course materials or external resources for better understanding.
    - If you're stuck or need clarification on the problem statement, don't hesitate to reach out to your instructor or mentor.

What does 'good' look like?

1. Define the Problem Statement and perform Exploratory Data Analysis

	Hint	**Approach**
a. Definition of problem	Begin with a clear objective. Why is predicting delivery times crucial for Porter?	The goal is to estimate delivery times accurately using various attributes to enhance customer satisfaction and optimize logistic operations.
b. Observations on Data	Thoroughly understand the dataset's structure. Observe the shape of the data, types of all attributes, detection of missing values, and statistical summary.	Utilize functions like data.info(), data.describe(), and data.shape in Python. Identify numeric vs. categorical attributes and convert data types if necessary.
c. Univariate Analysis	Analyze individual variables (distribution plots of continuous variables, bar plots/count plots of categorical variables).	For continuous variables, use histograms or density plots; for categorical variables, use countplots. This helps in understanding the distribution of individual variables.
d. Bivariate Analysis	Explore relationships between two variables (scatter plots for continuous vs. continuous, box plots for categorical vs. continuous).	For example, use scatter plots to analyze the relationship between the number of items and delivery time.
e. Illustrate the insights based on EDA	Each graph and table should offer an insight.	Note any surprising distributions, correlations, or unusual behaviors seen in the bivariate analysis. Comment on the range of attributes and identify outliers using box plots and IQR. Decide if outliers should be managed or retained based on

		the business context.
f. Comments on Distributions:	For continuous variables, comment on skewness.	For relationships, note positive or negative correlations, clusters, or other patterns. Each univariate and bivariate plot should be accompanied by a 2-3 line comment or observation. For example, "Scatter plot of order time vs. delivery time shows that orders placed during peak hours tend to have longer delivery times."

2. Data Preprocessing

	Hint	Approach
a. Duplicate value check	Identify and handle duplicate entries.	It's useful to check for duplicates based on a subset of features, as complete rows might not be identical, but a subset of attributes could have repeated patterns.
b. Missing value treatment	Crucial for reliable model training.	a. Identify columns with missing values. b. Choose the best strategy for each column: imputation using central tendencies, deletion, or advanced methods based on the variable's type and importance. c. Focus on smart imputation strategies to retain the integrity of the data.
c. Outlier treatment	Outliers can significantly affect model predictions.	a. Use graphical tools to visualize and detect outliers. b. Choose an appropriate

		technique to handle outliers: capping, transformation, or removal, based on logical reasoning.
d. Feature engineering	Enhance predictive power through creative feature modification.	a. Create binary flags for certain conditions in relevant attributes. b. Extract meaningful time-related features from datetime fields, like the hour of the day or day of the week. c. Derive location-based trends by extracting regions from address fields. d. Convert textual data into numerical values or categories for improved model utility.
e. Data preparation for modeling	Make data suitable for neural network modeling.	Scale features appropriately, considering the data distribution and model sensitivity. Apply suitable encoding techniques for categorical variables:
f. Identify normal vs skewed distributions and understand why.	Identify normal vs skewed distributions and understand why.	For continuous variables, comment on the skewness. For relationships, comment on positive or negative correlations, clusters, or other patterns noticed.

3. Model building

	Hint	**Approach**
a. Build the Neural Network Regression Model	Start by initializing a neural network model. Consider the complexity of your data to determine the number of layers and neurons.	Prepare the data for training and validation. Split your data into training and testing sets. Fit the model on the training data. Use a sequential model

		with an appropriate loss function for regression, like Mean Squared Error (MSE).
b. Hyperparameter Tuning	Experiment with different hyperparameters like learning rate, batch size, and the number of epochs.	Utilize Keras Tuner or GridSearchCV for systematic tuning. Define a grid of hyperparameters to explore. Test various combinations to find the best performing model.
c. Handling Overfitting and Underfitting	Monitor overfitting by comparing training and validation loss. Implement dropout or regularization if overfitting is observed..	Implement strategies like dropout, regularization, or adjusting the model's complexity to prevent overfitting or underfitting..
d. Activation Functions and Optimizers:	Coefficients provide insight into the importance and relationship of predictors with the outcome.	Adjust the complexity of the network or the amount of training data if underfitting is an issue.
e. Activation Functions and Optimizers:	Choose activation functions that match your problem type.	ReLU is a common choice for hidden layers, while a linear function might be used for the output layer in regression. Select an optimizer based on your model's requirements. Adam is a good starting point due to its adaptive learning rate capabilities.

4. Results Interpretation & Stakeholder Presentation

	Hint	**Approach**
a. Understand the Business Context	Immerse yourself in Porter's objectives and challenges.	Understand how reducing delivery times impacts customer satisfaction and operational efficiency. Frame your findings to align

		with Porter's key performance indicators (KPIs), like on-time delivery rates and customer satisfaction scores.
b. Interpreting Model Coefficients	While neural networks are complex, focus on the overall performance of the model	Interpret the results in terms of how different factors (like time of day, order size) impact delivery times . Discuss how the model's predictions align with business expectations and operational realities.
c. Visual Representations	Use graphs to illustrate the model's performance, such as loss curves over training epochs or scatter plots comparing predicted vs. actual delivery times.	Visualize key findings or trends that emerged from the analysis to make the insights more accessible to stakeholders.
d. Trade-off Analysis	Discuss the balance between model complexity and interpretability, or between prediction accuracy and computational efficiency.	Highlight any trade-offs made during feature selection, model architecture design, or hyperparameter tuning.
e. Recommendations	Provide actionable strategies for Porter, like optimizing delivery routes or adjusting delivery partner allocation based on order patterns.	Back up these recommendations with evidence from your analysis, showing how certain changes can lead to improved delivery times.
f. Feedback Loop	Propose methods for continuous monitoring and updating of the model.	Discuss how changing order patterns or expanding services might require model adjustments. Suggest a framework for incorporating new data and feedback into the model for ongoing improvement.

Questionnaire (Answers should present in the text editor along with insights):

1. Defining the problem statements and where can this and modifications of this be used?
2. List 3 functions the pandas datetime provides with one line explanation.
3. Short note on datetime, timedelta, time span (period)
4. Why do we need to check for outliers in our data?
5. Name 3 outlier removal methods?
6. What classical machine learning methods can we use for this problem?
7. Why is scaling required for neural networks?
8. Briefly explain your choice of optimizer.
9. Which activation function did you use and why?
10. Why does a neural network perform well on a large dataset?