Problem Statement

Objective

This assignment will help you apply linear regression techniques to a real-world business scenario. By working through this exercise, you will gain hands-on experience in using regression analysis to identify key relationships between variables, make data-driven predictions and extract actionable business insights. You will also develop an understanding of how to interpret regression outputs, assess model performance and effectively communicate findings to support strategic decision-making.

Business Value

The growing demand for quick and efficient delivery in the logistics industry calls for the development of systems that can predict delivery times accurately. Porter, an intra-city logistics marketplace, services millions of customers daily, and optimising delivery times is crucial for improving operational efficiency.

The objective is to build a linear regression model that can perform the following:

1. Predict the delivery time for an order based on various input features.
2. Help optimise delivery operations by providing accurate time estimates.
3. Support operational planning and resource management, allowing for more effective use of delivery partners.

Dataset Overview

Context

The Porter Parcel Delivery Times data set provides details about orders placed through Porter. The data has information about the orders, such as item quantity and order amount; information about the store, such as category, distance from customer and market ID; and operational information, such as the number of delivery partners.

Content

The dataset consists of multiple observations, each representing data for a specific order, with the following key attributes:

Column Name	Description
`market_id`	ID of the market where the restaurant is located.
`created_at`	The timestamp when the order was placed.
`actual_delivery_time`	The timestamp when the order was delivered.
`store_primary_category`	The category of the restaurant (e.g., fast food or dine-in).
`order_protocol`	The integer code indicating how the order was placed (e.g., via Porter or call to restaurant).
`total_items`	The total number of items in the order.
`subtotal`	The total price of the order.
`num_distinct_items`	The number of distinct items in the order.
`min_item_price`	The price of the cheapest item in the order.
`max_item_price`	The price of the most expensive item in the order.

`total_onshift_dashers`	The number of delivery partners on duty at the time the order was placed.
`total_busy_dashers`	The number of delivery partners already attending to other tasks.
`total_outstanding_orders`	The number of outstanding orders to be fulfilled at the time the order was placed.
`distance`	The total distance from the restaurant to the customer.

Scoring and Penalty

- **Total marks**: **100** (75 for code notebook and 25 for report)
- **Extension and penalty**: As given in your learner handbooks

Instructions

1. Each learner should attempt this assignment individually.
2. Programming Language: Python
3. You will be provided with the data set and a starter notebook. You have to perform analyses in the starter notebook only. The notebook also contains some subjective questions, which also have to be answered inside the notebook itself.
4. You must not change any headings, subheadings, questions or tasks in your notebook as this can cause problems with grading.
5. For analyses and modelling tasks, you should use only the following libraries: NumPy, Pandas, Matplotlib, Seaborn, Plotly, Statsmodels and Scikit-Learn.
6. The data will have inconsistencies and outliers. Handle them as per your understanding and mention them in your report.
7. You are encouraged to search the web and consult AI tools for conceptual understanding. However, using plagiarized code is strictly prohibited and using purely AI-generated code is strongly discouraged.
8. Submitting plagiarized and purely AI-generated code or reports will result in significant penalties to your scores.

Submission Guidelines

1. To submit your solution, push your submission to GitHub and submit the GitHub link in the submission field.
2. You are required to upload your solution as a **zip file** titled `"LR_Delivery_Time_Prediction_<your_name>.zip"` in a public GitHub repository.
3. The repository should be named after the assignment.
4. The zipped folder that you upload should contain two files:
   (a) an **Interactive Python Notebook** (.ipynb) that contains your code
   (b) a **Report Document** (.pdf) that presents your visualisations, analysis, results, insights, and outcomes.
5. Note that these files should only be generated from the starter files provided to you.
6. Both your Jupyter notebook and report should contain your name and the assignment title.
7. Mention all assumptions made in the report.
8. Your answers to all the tasks mentioned in the starter notebook should be present in the report. Any graphs/plots you generate for analysis should also be attached to the report.

Results Expected from Learners

Present the overall approach of the analysis in a report document. Mention the problem statement and the analysis approach briefly.

In the starter notebook, you will find headings, subheadings and checkpoints stating the tasks you need to perform. The marks associated with each checkpoint will also be mentioned in the notebook. Keep in mind not to edit the cells with marking schemes and questions. You can find a brief description of the tasks below.

1. Load the data

Load the CSV file as a DataFrame

2. Data Preprocessing and Feature Engineering [15 marks]

- 2.1 Fixing Datatypes [5 marks]
- 2.2 Handling Missing Values [5 marks]
- 2.3 Train-Validation Split [5 marks]

3. Exploratory Data Analysis on Training Data [20 Marks]

- 3.1 **Feature Distribution** [7 Marks]
  - i. Distribution of numerical features
  - ii. Distribution of categorical features
  - iii. Distribution of Target feature
- 3.2 **Relationships Between Features** [3 Marks]
  - i. Analyse relationships of features with the target variable
- 3.3 **Correlation Analysis** [5 Marks]
  - i. Plot heatmap of feature correlations
  - ii. Drop columns with weak correlation to the target variable
- 3.4 **Outlier Handling** [5 Marks]
  - i. Visualise potential outliers
  - ii. Handle outliers in all columns

4. Exploratory Data Analysis on Validation Data [Optional]

- 4.1 Feature Distribution
- 4.2 Relationships between different features
- 4.3 Correlation Analysis

5. Model Building [15 Marks]

- 5.1 Perform feature scaling [3 Marks]
- 5.2 Build a Simple Linear Regression Model [5 Marks]
- 5.3 Build the model and fit RFE to select the most important features [7 Marks]

6. Results and Inference [5 Marks]

- 6.1 Perform Residual Analysis [3 Marks]
- 6.2 Perform Coefficient Analysis [2 marks]

7. **Subjective Questions [20 Marks]**

7 .1 Assignment-related subjective questions [8 Marks]

7 .2 Topic-related subjective questions [12 marks]

Evaluation Guidelines

The following rubrics will be used for judging your solutions to the tasks above.

Table 2: Rubrics

Criteria	Meets expectations	Does not meet expectations
Data Preparation	1. Data types are fixed appropriately to match data requirements. 2. Features are engineered effectively to add value to the analysis.	1. Data types are not fixed or are fixed incorrectly. 2. Features are poorly engineered or irrelevant.
Train Test Split	1. Data is split correctly into training and test sets maintaining a ratio of 80:20 or 70:30.	1. Data is split incorrectly or a proper ratio is not maintained.
Exploratory Data Analysis	1. The target variable's distribution and relationships are analysed with relevant plots and explanations. 2. Summary statistics (mean, median, standard deviation, etc.) are calculated and interpreted correctly for all relevant variables. 3. Feature distributions are visualised using appropriate plots, with clear insights provided on trends and anomalies. 4. Outliers are identified accurately and handled appropriately, ensuring no significant loss of information.	1. The distribution analysis is missing or incomplete, with no insights provided. 2. Statistics are incomplete, incorrect, or not interpreted. 3. Visualisations are missing, incorrect, or lack insights. 4. Outlier analysis is missing, incorrect, or poorly implemented.

Continued on next page

Table 2: Rubrics (Continued)

Criteria	Meets expectations	Does not meet expectations
Model Building	1. Target and feature variables are defined correctly. 2. Baseline Linear Regression model is initialised and trained correctly. 3. Feature elimination is performed effectively using RFE. Suitable balance between number of features and resulting performance is maintained. 4. The selected features in RFE are identified. 5. Predictions on test data are generated and evaluated. 6. Model evaluation is performed using appropriate metrics.	1. Target and feature variables are not defined correctly. 2. Baseline Linear Regression model is not initialised and trained correctly. 3. Feature elimination is performed incorrectly. Number of selected features do not explain the resulting performance. 4. The selected features in RFE are not identified. 5. Predictions on test data are not generated and evaluated. 6. Model evaluation is inadequate.
Results and Inference	1. Residual Analysis is implemented correctly and inferences are drawn accurately. 2. Error analysis is performed appropriately explaining model performance.	1. Residual Analysis missing or implemented incorrectly. 2. Error analysis is not performed appropriately.
Presentation and Recommendations	1. Report has a clear structure, is concise, and highlights important results. 2. Recommendations are realistic, actionable, and coherent with the analysis. 3. Assumptions are stated clearly.	1. Report lacks structure, is too long, or does not emphasise key observations. 2. Recommendations are unrealistic, non-actionable, or incoherent with the analysis. 3. Assumptions made, if any, are not stated clearly.

Continued on next page

Table 2: Rubrics (Continued)

Criteria	Meets expectations	Does not meet expectations
Conciseness and Readability of the Code	1. Code is concise, syntactically correct, and uses built-in functions wherever possible. 2. Custom functions are used for repetitive tasks. 3. Code is readable with clear variable names and necessary comments.	1. Long and complex code is used instead of built-in functions. 2. Custom functions are not used for repetitive tasks, leading to code repetition. 3. Code readability is poor due to unclear variables or lack of comments.