Delivery Time Prediction — Regression Report

Objective

The aim is to forecast food delivery times through regression by examining features at the order, restaurant, and dasher levels. This analysis aids in optimizing delivery operations and pinpointing significant factors contributing to delivery delays.

1. Data Overview

The dataset comprises:

Order-level information (items, price, protocol)

Dasher availability (on shift, busy)

Restaurant metadata (category, distance)

Timestamps for order placement and delivery

2. Data Preprocessing & Feature Engineering

Converted timestamps into datetime objects.

Engineered the Time Taken in minutes as the target variable.

Extracted the Hour of the day, Day of the week, and created a binary isWeekend feature.

Encoded categorical features (order_protocol, store_primary_category).

Eliminated irrelevant columns (created_at, actual_delivery_time).

3. Exploratory Data Analysis

Distributions: The target variable exhibits a right skew; distance and subtotal also display skewness.

Categorical Analysis: Weekends, order protocol, and store category have an impact on delivery time.

Heatmap: Distance, total outstanding orders, and total busy dashers show a strong correlation with delivery time.

Outliers: Identified using the IQR method and removed rows with extreme values to enhance model generalization.

4. Train-Test Split

Divided the dataset into:

Train set: 80%

Test set: 20% (test_size=0.2) to assess the model's performance in real-world scenarios.

5. Model Building

Feature Scaling: Implemented StandardScaler on numerical features to ensure fair comparison of coefficients.

Model: Developed using LinearRegression from scikit-learn.

Evaluation Metrics:

R^2 Score

Mean Absolute Error (MAE)

Root Mean Squared Error (RMSE)

Recursive Feature Elimination (RFE)

Conducted RFE to identify the top 8 features affecting delivery time.

Retrained the final model utilizing the selected features for improved interpretability and simplicity.

6. Results & Inference

Residual Analysis

Residuals were predominantly centered around zero (no significant patterns)

A slight skew was noted → potential advantage from log transformation in future models.

Coefficient Analysis

Top positive contributors include distance, total_outstanding_orders, and total_busy_dashers.

Features such as min_item_price and max_item_price exhibited weak correlation and were subsequently removed.

7. Subjective Q&A Highlights

Overfitting versus Underfitting: Overfitting is characterized by high variance, while Underfitting is associated with high bias.

Linear Regression: This method presumes a linear relationship between features and the target variable, and it is optimized by minimizing the Mean Squared Error (MSE).

Residual Plots: These plots are utilized to identify patterns, non-linearity, and issues related to variance.

Dropped Features: The feature total_items was eliminated due to its weak correlation with delivery time.

Conclusion

The model effectively predicts delivery times by leveraging key operational features. There exists an opportunity for further enhancement through the application of regularized models (such as Ridge or Lasso) or non-linear models (including XGBoost and Random Forest).

Subjective Questions [20 marks]

Answer the following questions only in the notebook. Include the visualisations/methodologies/insights/outcomes from all the above steps in your report.

Subjective Questions based on Assignment

Question 1. [2 marks]

Are there any categorical variables in the data? From your analysis of the categorical variables from the dataset, what could you infer about their effect on the dependent variable?

Answer: Yes, there are categorical variables in the dataset. The main ones identified during exploratory analysis are:

Order_protocol store_primary_category iswekend

From the analysis:

Order protocols showed differences in delivery times, likely due to how the orders are processed.

Store categories (e.g., fast food vs. fine dining) influence prep time and thus affect delivery duration.

Weekend orders tend to take longer due to higher demand or limited delivery availability.

These categorical variables, after encoding, contribute meaningfully to predicting delivery time.

Question 2. [1 marks]

What does test_size = 0.2 refer to during splitting the data into training and test sets?

Answer:

Test_size = 0.2 means that 20% of the data will be used as the test set, and the remaining 80% will be used for training the model. It controls the proportion of data reserved for evaluating model performance on unseen data.

Question 3. [1 marks]

Looking at the heatmap, which one has the highest correlation with the target variable?

Answer:

distance with 0.46, is highly correlated

Question 4. [2 marks]

What was your approach to detect the outliers? How did you address them?

Answer:

We used the Interquartile Range (IQR) method to detect outliers. Any data points below Q1 − 1.5 × IQR or above Q3 + 1.5 × IQR were considered outliers and removed from the dataset.

Question 5. [2 marks]

Based on the final model, which are the top 3 features significantly affecting the delivery time?

Answer:

If we take the abosulte values

total_busy_dashers (|−1.173| → Highest impact, negative)

total_outstanding_orders (|1.143| → Second highest, positive)

distance (|0.562| → Third highest, positive) These are the top 3 features significantly affecting the delivery time

General Subjective Questions

Question 6. [3 marks]

Explain the linear regression algorithm in detail

Answer:

Linear Regression is one of the simplest and most widely used algorithms in statistics and machine learning. It is used to model the relationship between a dependent variable (target) and one or more independent variables (features or predictors).

Objective

To model a linear relationship between:

Independent variables (X)

Dependent variable (Y)

The goal is to predict the value of Y given new values of X.

Equation

$Y = b_o + b_1X$

Y = Target (dependent variable)

X = Feature

b_o = Intercept (bias)

b_1 = Coefficients

ϵ = Error Term

The algorithm finds the line that best fits the data, The best-fit line is calculated by minimizing the Residual Sum of Squares (RSS) — the sum of squared differences between the observed and predicted values.

Advantages Simple and easy to implement

Highly interpretable

Efficient with small to moderately sized datasets

Limitations

Assumes linearity — not suitable for nonlinear problems

Sensitive to outliers

Poor performance with multicollinearity

Doesn’t handle categorical variables unless encoded

Question 7. [2 marks]

Explain the difference between simple linear regression and multiple linear regression

Answer:

Difference Between Simple and Multiple Linear Regression

Feature	Simple Linear Regression	Multiple Linear Regression
Number of Independent Variables	1	2 or more
Equation	$Y = b_o + b_1X_1$	$Y = b_o + b_1X_1 + b_2X_2 + ... + b_nX_n$
Visual Representation	Straight line in 2D space	Plane or hyperplane in higher dimensions
Use Case Example	Predicting delivery time using only distance	Predicting delivery time using distance, items, and hour
Complexity	Very simple and easy to interpret	More complex but gives a better model with more variables
Accuracy (usually)	Lower, since it uses limited information	Higher, as it considers more factors

Use Simple Linear Regression when you have only one feature, and Multiple when you want to use many features to make better predictions.

Question 8. [2 marks]

What is the role of the cost function in linear regression, and how is it minimized?

Answer:

Role of the Cost Function in Linear Regression

What is the Cost Function?

The **cost function** measures how well the linear regression model is performing. It calculates the **error** between the predicted values and the actual values.

Common Cost Function: **Mean Squared Error (MSE)**

MSE=1n∑i=1n(yi−$\hat{y}$ i)2MSE=1n∑i=1n(yi−y^i)2

How is it Minimized?

To find the best-fitting line, we minimize the cost function using an optimization algorithm like:

- **Gradient Descent**

Steps:

1. Initialize weights (θ) randomly.
2. Calculate the gradient (partial derivatives of cost w.r.t. each parameter).
3. Update weights in the opposite direction of the gradient.
4. Repeat until convergence (cost becomes minimum).

Update Rule:

θt+1=θt−η·∇J(θt)θt+1=θt−η·∇J(θt)

Goal:

To find the line (model parameters) that minimizes the total error, so that predictions are as close as possible to the actual values.

Question 9. [2 marks]

Explain the difference between overfitting and underfitting.

Answer:

Overfitting happens when a model learns the training data too well, including its noise and outliers. It performs well on training data but poorly on unseen data, meaning it doesn't generalize well.

Underfitting occurs when a model is too simple to capture the underlying patterns in the data. It performs poorly on both training and test sets, indicating that it hasn't learned enough from the data.

Question 10. [3 marks]

How do residual plots help in diagnosing a linear regression model?

Answer:

1. Linearity Check

Plot: Residuals vs. Predicted

What to Look For: Random scatter = good. Curves = model misses non-linear trends.

Solution: Add polynomial terms or transform variables.

2. Heteroscedasticity (Unequal Variance)

Plot: Residuals vs. Predicted

What to Look For: Cone/funnel-shaped spread.

Solution: Transform target (e.g., log) or use weighted regression.

3. Normality of Errors

Plot: Q-Q Plot

What to Look For: Points on a straight line = normally distributed errors.

Solution: Apply transformations or address outliers.

4. Outlier Detection

Plot: Residuals vs. Features

What to Look For: Points beyond ±3 std. deviations.

Solution: Investigate anomalies or use robust models.

5. Autocorrelation (Time-Based Data)

Plot: Residuals vs. Time

What to Look For: Repeating patterns or waves.

Solution: Add time lags or switch to time series models like ARIMA.