Jagdsh Chand   
Report: LR Delivery Time Estimation

Include your visualisations, analysis, results, insights, and outcomes. Explain your methodology and approach to the tasks. Add your conclusions to the sections.

## Loading the data

“porter\_data\_1.csv” was loaded into the pandas dataframe and data types of the individual columns was printed.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Data Preprocessing and Feature Engineering

### Fixing the Datatypes

* + 1. **Date and time fields were fixed**

A screenshot of a computer program

AI-generated content may be incorrect.

* + 1. **Convert categorical fields to appropriate data types**

The datatype “category” was used for the the categorical columns

['market\_id', 'store\_primary\_category', 'order\_protocol']

A screenshot of a computer program

AI-generated content may be incorrect.

### Feature Engineering

* + 1. **Calculate the time taken using the features** **`actual\_delivery\_time` and `created\_at`**

I populated the time\_taken column in minutes for the `actual\_delivery\_time` and `created\_at`

A screenshot of a computer program

AI-generated content may be incorrect.

* + 1. **Extract the hour at which the order was placed and which day of the week it was. Drop the unnecessary columns.**

1. **I used the lambda function to calculate the isWeekend column**
2. **I dropped unnecessary columns** 'created\_at', 'actual\_delivery\_time', 'day\_of\_week', 'delivery\_time\_timedelta'

A screenshot of a computer program

AI-generated content may be incorrect.

### Creating training and validation sets

* + 1. **Define target and input features**

I defined the y as time\_taken since it is what we are predicting, and the features as variables\_all

A screenshot of a computer program

AI-generated content may be incorrect.

* + 1. **Split the data into training and test sets**

I split the data into 70% training and 30% test data with the respective rows **123043** and **52734**

A screenshot of a computer program

AI-generated content may be incorrect.

## Exploratory Data Analysis on Training Data

### Feature Distributions

Defined the categorical ('market\_id', 'store\_primary\_category', 'order\_protocol', 'isWeekend') and numerical columns.

A screen shot of a computer

AI-generated content may be incorrect.

* + 1. **Plot distributions for numerical columns in the training set.**

Following numerical columns has some notable distributions:

'total\_items': mostly less than 100, one outlier above 400

'subtotal': mostly less than 5000, few outliers

'num\_distinct\_items': 2.5 items distinct stands out, however it varies a lot

'min\_item\_price': many below 1000 and a huge outlier

'max\_item\_price': many still less than 1000

'total\_onshift\_dashers': peak in 20-25 and decreases linearly to 175

'total\_busy\_dashers': it is similar to onshift dashers, decreases linearly to 140

'total\_outstanding\_orders': more less than 50

'distance': 20-40 wins the chart

'time\_taken': less than 70 , with many in 50

'hour\_of\_day': as the time increases less orders, peaking midnight

A group of blue and white graphs

AI-generated content may be incorrect.

* + 1. **Check the distribution of categorical features**

'market\_id': market 2, 4,1 are the top three

'store\_primary\_category': three category (5, 48, 68) makes more than 8000 times

'order\_protocol': alternative order protocol (1,3,5) are doing good.

'isWeekend': many in weekdays than in weekendsA group of graphs with numbers

AI-generated content may be incorrect.

* + 1. **Visualise the distribution of the target variable to understand its spread and any skewness**

'time\_taken': less than 70 , with peaking b/w 40-50.

A graph of a distribution of goods

AI-generated content may be incorrect.

**Conclude with results**

This analysis of the of the delivery time . Below are the key findings:

### Majority of transactions involve a relatively small quantity of items.

### People tend to order a small variety of items, even if the total quantity of items (from total\_items) can be higher. The small secondary peak might indicate some bulk or diverse orders

### Most deliveries are completed within a relatively consistent timeframe, indicating efficiency. The slight skew or minor peak might suggest some faster deliveries.

### Hour of the day distribution strongly suggests typical dinner and late night as the busiest periods for the service. The small early morning peak could be related to breakfast orders.

### 

### Relationships Between Features

* + 1. **Scatter plots for important numerical and categorical features to observe how they relate to `time\_taken`**

I used scatter plot iterated every variable to plot the scatter plot

n\_variables\_cols = len(variables\_all)  
n\_cols = 3  
n\_rows = (n\_variables\_cols + n\_cols - 1) // n\_cols  
plt.figure(figsize=(n\_cols \* 6, n\_rows \* 5))  
  
for i, var in enumerate(variables\_all):  
 plt.subplot(n\_rows, n\_cols, i + 1)  
 sns.scatterplot(data=porter\_train, x=var, y='time\_taken')  
 plt.title(f'Scatter plot b/w time\_taken of {var}')  
  
plt.tight\_layout()  
plt.show()

A graph with different colored bars

AI-generated content may be incorrect.

* No much pattern is reconized in store\_primary\_category, order\_protocol, num\_distinct\_items, hour\_of\_day, is\_weekend since these are categorical, histogram shows more useful info
* As **subtotal** increases the time\_taken also increase and there is minimum time\_taken (30min) above total of 10000
* Similar to subtotal **total****\_outstanding\_orders** increases time\_taken also increase and there is minimum time\_taken (30min) above 150 total\_outstanding\_orders
* As distance increase the time\_taken also increases
* **Total\_onshit\_dasher** and **total\_busy\_dashers** does not bring any major pattern.
* Time\_taken is less in mornings 7-14 and increases especially after midnight 2-4 AM

A graph with blue dots

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

### Correlation Analysis

* + 1. **Heatmap to display correlation**

A screenshot of a graph

AI-generated content may be incorrect.

Heat map correlation with numerical columns:

A screenshot of a graph

AI-generated content may be incorrect.

* + 1. Drop the columns with target variables

time\_taken 1.000000

distance 0.459712

subtotal 0.412878

total\_outstanding\_orders 0.381642

hour\_of\_day 0.344002

num\_distinct\_items 0.313384

max\_item\_price 0.254671

total\_items 0.219104

total\_busy\_dashers 0.202562

total\_onshift\_dashers 0.166812

order\_protocol 0.137906

isWeekend 0.133896

market\_id 0.075735

store\_primary\_category 0.027475

min\_item\_price 0.022281

dtype: float64

based on the above information dropping the following columns which has weak correlations: ['market\_id', 'store\_primary\_category', 'min\_item\_price']

* 1. Handling the Outliers
     1. Visualizing the potential outliers
     2. time\_taken, num\_distinct\_items, subtotal have visible outliers
     3. total\_items, max\_item\_price way out of values which are very clear outlier

A screenshot of a graph

AI-generated content may be incorrect.

* + 1. handling Outliers

Based on the above info, have take the following condition to eliminate the outlier

(porter\_train['distance'] < 60) &  
 (porter\_train['total\_items'] < 25) &  
 (porter\_train['max\_item\_price'] < 5000) &  
 (porter\_train['subtotal'] < 15000) &  
 (porter\_train['total\_outstanding\_orders'] < 250) &  
 (porter\_train['num\_distinct\_items'] < 10) &  
 (porter\_train['time\_taken'] < 90)

A screenshot of a graph

AI-generated content may be incorrect.

**\*\*\*Reduced the outlier but not completely removed. I use StandardScalar since there are outliers\*\*\***

## Optional

## **Model Building**

### Feature Scaling

I used StandardScaler for scaling numerical columns for training and testing data.

scaler\_X\_train = StandardScaler()  
porter\_train[numerical\_cols] = scaler\_X\_train.fit\_transform(porter\_train[numerical\_cols])  
original\_feature\_means = pd.Series(scaler\_X\_train.mean\_, index=porter\_train[numerical\_cols].columns)  
original\_feature\_stds = pd.Series(scaler\_X\_train.scale\_, index=porter\_train[numerical\_cols].columns)  
porter\_train.head()

A screenshot of a computer

AI-generated content may be incorrect.

* 1. Build a linear regression model

I added the constants and run OLS model with summary

A screenshot of a computer

AI-generated content may be incorrect.

* r-squared is 87.1 which is good
* F-Sta is 7.461e+04 which is high also good
* All the variables have P>|t| as 0.000 which means all the variable contribute to the effect in non randomness manner.
* Certain varaibles has positive and negative coeefficients

Train the model and predict:

# Train the model using the training data  
from sklearn.linear\_model import LinearRegression  
  
model = LinearRegression()  
model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

Evaluation metrics:

* R2 between test data and predicted data: 0.8737992135500233
* Mean absolute error between test data and predicted data: 0.2551228125215107

A graph of error terms

AI-generated content may be incorrect.

* Errors meaned near zero,
* Little stretched to the right

A graph with blue dots

AI-generated content may be incorrect.

* It is scattered around the fitted line and little fan in and fan out model
* It converges the top values
  1. **Build the model and fit RFE to select the most important features**

By using RFE, I try to find a subset of 9 features that are considered the most impactful predictors of time\_taken according to the LinearRegression model's internal metrics

# Importing RFE and LinearRegression  
from sklearn.feature\_selection import RFE  
from sklearn.linear\_model import LinearRegression  
  
# Running RFE with the output number of the variable equal to 10  
lm = LinearRegression()  
lm.fit(X\_train, y\_train)  
  
rfe = RFE(lm, n\_features\_to\_select=9)  
rfe = rfe.fit(X\_train, y\_train)  
print(list(zip(X\_train.columns, rfe.support\_, rfe.ranking\_)))  
print(X\_train.columns[rfe.support\_])

**Finally I get the following 9 important features and running the OLS :**

['distance', 'subtotal', 'total\_outstanding\_orders', 'hour\_of\_day', 'num\_distinct\_items', 'total\_busy\_dashers', 'total\_onshift\_dashers', 'order\_protocol', 'isWeekend']

**A screenshot of a computer

AI-generated content may be incorrect.**

* **Dependent Variable**: time\_taken: Confirms that the model is attempting to predict delivery time.
* **R-squared: 0.855:**Very high R-squared value meaning approximately 85.5% of the variance in delivery time can be explained by the 9 independent variables included in your model. This indicates a strong predictive capability.
* **Adjusted R-squared**: 0.855: Identical to the R-squared, which is expected with a large number of observations and a well-chosen set of predictors. It further confirms the strength of the model, indicating that the features are contributing meaningfully.
* **F-statistic:** 7.985e+04 / Prob (F-statistic): 0.00: Large F-statistic and the p-value of 0.00 (effectively zero) mean that the model is highly statistically significant.

**VIF of the independent variable to confirm the multicollinearity:**

**A screenshot of a computer

AI-generated content may be incorrect.**

* **total\_onshift\_dashers has impact on other variable it is confirmed when it is removed from the model calculation and the r2 reduces to 71.4% from 85.5 %**
* **further when total\_busy\_dashers is removed r2 score reduces further to 52.5 % from 85.5% so these two variables are important**

#### 

#### **Removing total\_onshift\_dashers:**

A screenshot of a computer

AI-generated content may be incorrect.

#### **Removing total\_busy\_dashers:**

A screenshot of a computer

AI-generated content may be incorrect.

* 1. **Final Model**

Final model was selected with following variables

selected\_variables = ['distance', 'subtotal', 'total\_outstanding\_orders', 'hour\_of\_day','num\_distinct\_items', 'max\_item\_price', 'total\_items',  
'total\_busy\_dashers', 'total\_onshift\_dashers']

A screenshot of a computer screen

AI-generated content may be incorrect.

**Interpretation of Coefficients (coef )**.

1. **total\_outstanding\_orders (Coef: 1.9854):** **Most impactful positive predictor.** A one standard deviation increase in outstanding orders leads to nearly a 2 standard deviation increase in delivery time. Strong evidence of system congestion.
2. **total\_onshift\_dashers (Coef: -1.3866):** **Most impactful negative predictor.** A one standard deviation increase in on-shift dashers leads to a 1.39 standard deviation *decrease* in delivery time. Crucial for operational efficiency.
3. **total\_busy\_dashers (Coef: -0.4997):** Very strong negative impact. A one standard deviation increase in busy dashers reduces delivery time by 0.5 standard deviations.
4. **distance (Coef: 0.4556):** Significant positive impact. A one standard deviation increase in distance adds 0.46 standard deviations to delivery time.
5. **hour\_of\_day (Coef: -0.2388):** Noticeably larger negative impact than previously seen (-0.1016). This could indicate that transforming hour\_of\_day or other correlated variables has clarified its linear effect, where later hours correspond to faster delivery times.
6. **subtotal (Coef: 0.2450):** Significant positive impact. Larger subtotals associated with longer times.
7. **num\_distinct\_items (Coef: 0.0981):** Positive impact. More distinct items slightly increase time.
8. **max\_item\_price (Coef: 0.0515):** Positive impact. Higher priced items slightly increase time.
9. **total\_items (Coef: -0.0154):** Smallest absolute impact. More total items slightly *decrease* delivery time.
10. **Performance Residual Analysis**
    1. **Performance Residual Analysis**
       1. **Error Terms**

**A graph with blue lines

AI-generated content may be incorrect.**

Inference from Residuals Histogram (Error Terms)

* The distribution is cantered very close to zero, which is ideal for regression models, indicating that the model is not systematically over or under-predicting.
* It is nearly bell curve, however it extends to the positive side.the mean error is nearly zero, however little to the negative side.
* **Right skew** (positive skew) in the residuals with the tail extending further to the right (positive errors) than to the left (negative errors). This may tends that model may under-predict more often
  + 1. **Actual vs Predicted**

**A graph with blue dots

AI-generated content may be incorrect.**

Inference from Plotting y\_test and y\_pred (Residuals vs. Predicted Values ) -

* It clearly says most of the variables
* The residuals are not randomly scattered around the horizontal line.
* The graph is not as scatted when compared to ideal scenario.
* 'total\_onshift\_dashers'. This is a tarde-off that we are gonna take for prediction
* Heteroscedasticity is present, meaning the variance of the errors is not constant across all levels of the predicted or actual values.
  + 1. **Q\_Q Plot**

Inference from Q-Q (Quantile-Quantile) plot

* The residuals are not perfectly normally distributed, particularly in the tails. The presence of larger positive residuals (under-predictions) is a concern and aligns with the skew observed in the error distribution plot.
* non Normality of residuals at the start and end of the line.

**A graph with a line and a red line

AI-generated content may be incorrect.**

* + 1. **Variance Inflation Factor (VIF)**
* From the Variance Inflation Factor (VIF) it is very clear that "total\_onshift\_dashers" has more multicollinearity once we remove the r2 comes to 71.4% and once we remove "total\_busy\_dashers" it comes around 52.5%
* These two fields total\_onshift\_dashers, total\_busy\_dashers has high multi collinearity and influences other variables as well.
  1. **Perform coefficient Analysis**
     1. Perform coefficient analysis to find how changes in features affect the target. Also, the features were scaled, so interpret the scaled and unscaled coefficients to understand the impact of feature changes on delivery time.

1. # Compare the scaled vs unscaled features used in the final model  
     
   scaled\_coefs = lm\_final.params.drop('const')  
   scaled\_intercept = lm\_final.params['const']  
     
   unscaled\_feature\_coefs = {}  
   sum\_for\_unscaled\_intercept = 0  
     
   for feature, scaled\_b in scaled\_coefs.items():  
    unscaled\_b = scaled\_b / original\_feature\_stds[feature]  
    unscaled\_feature\_coefs[feature] = unscaled\_b  
    sum\_for\_unscaled\_intercept += (unscaled\_b \* original\_feature\_means[feature])  
    print(f"Coef for {feature} Scaled: {scaled\_b:.2f} Unscaled: {unscaled\_b:.2f}")

| Feature | Standardized Coefficient | Impact Direction |
| --- | --- | --- |
| total\_outstanding\_orders | +1.98 | Positive |
| total\_onshift\_dashers | -1.39 | Negative |
| total\_busy\_dashers | -0.50 | Negative |
| distance | +0.46 | Positive |
| subtotal | +0.25 | Positive |
| hour\_of\_day | -0.24 | Negative |
| num\_distinct\_items | +0.10 | Positive |
| max\_item\_price | +0.05 | Positive |
| total\_items | -0.02 | Negative |

A screen shot of a computer

AI-generated content may be incorrect.

**6.3.1.** Analyze the effect of a unit change in a feature, say 'total\_items'

* For every 1 km increase in distance, time\_taken is expected to increase by approximately 0.46 minutes
* For every 1 price increase in subtotal, time\_taken is expected to increase by approximately 0.24 minutes
* For every 1 order increase in total\_outstanding\_orders, time\_taken is expected to increase by approximately 1.99 minutes
* For every 1 hour increase in hour\_of\_day, time\_taken is expected to decrease by approximately -0.24 minutes
* For every 1 item increase in num\_distinct\_items, time\_taken is expected to increase by approximately 0.10 minutes
* For every 1 price increase in max\_item\_price, time\_taken is expected to increase by approximately 0.05 minutes
* For every 1 item increase in total\_items, time\_taken is expected to decrease by approximately -0.02 minutes
* For every 1 dasher increase in total\_busy\_dashers, time\_taken is expected to decrease by approximately -0.50 minutes
* For every 1 dasher increase in total\_onshift\_dashers, time\_taken is expected to decrease by approximately -1.39 minutes

A black screen with white text

AI-generated content may be incorrect.

#### Subjective Questions based on Assignment

**Question 1: 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?**

* The categrorical columns are 'market\_id', 'store\_primary\_category', 'order\_protocol', 'isWeekend' has detectable effect however weak in linear correlation with time\_taken. Since they have low correlation with time\_taken -0.076, 0.027, -0.14, 0.13 respectively. And in the Recursive Feature Elimination (RFE) the numerical column have more contribution on the time\_taken than any other categorical variables.
* P > |t| values of all the categorical values are 0.000 meaning that non-random effect on the model

**Question 2: What does `test\_size = 0.2` refer to during splitting the data into training and test sets?**

test\_size =0.2 means to split 20% of the total data as test data. And keep other 80% as training data.

**Question 3:Looking at the heatmap, which one has the highest correlation with the target variable?**

- "total\_onshift\_dashers" and "total\_busy\_dashers" => 0.94

- "total\_onshift\_dashers" and "total\_outstanding\_orders" => 0.94

- "total\_outstanding\_orders" and "total\_busy\_dashers" => 0.93

- "num\_distinct\_items" and "total\_items" => 0.74

**Question 4:What was your approach to detect the outliers? How did you address them?**

> I used boxplot to identify the outliers and I removed the values which are way out of line

> I arrived an optimal threshold in accepting few outliers on individual features, so that I can use StandardScaler

```

('distance' < 60)

('total\_items' < 25)

('max\_item\_price' < 5000)

('subtotal' < 15000)

('total\_outstanding\_orders' < 250)

('num\_distinct\_items' < 10)

('time\_taken' < 90)

```

**Question 5: Based on the final model, which are the top 3 features significantly affecting the delivery time?**

> The top three significantly affecting delivery time in the decreasing order, first being the most significant

- total\_outstanding\_orders -> positive correlation

- total\_onshift\_dashers -> negative correlation

- total\_busy\_dashers -> negative correlation

**Question 6:Explain the linear regression algorithm in detail**

> It is supervised predictive analysis model which establishes between one scaler (dependant) variable and one or more independant variables.

> Since it is supervised learning you label the training data and feed the model

> It is a fundamental and most used model for predicting is a continuous variable, e.g. scores of a student, time\_taken in delivery.

> It attempts to draw a "best-fit" straight line through the data points.

> There are two main types:

- Simple Linear Regression (SLR): Involves only one independent variable.

- Multiple Linear Regression (MLR): Involves two or more independent variables

> Linear regression guarantes interpolation then extrapolation

> There are few assumptions of OLS Linear Regression that must be satisfied

- Linear relationship between X and Y

- Error terms are normally distributed (not X, Y)

- Error terms are independent of each other

- Error terms have constant variance (homoscedasticity)

**Question 7:Explain the difference between simple linear regression and multiple linear regression**

> Simple Linear Regression (SLR): Involves only one independent variable while

> the equation for SLR is `Y=Beta0 + Beta1X + constant` since there is only one independe variable, only one X with coefficient

> Main purpose of using this to model is to fit a straight-line relationship between two continuous variables and predict the target based on changes in that single feature.

> Use Cases: When only one factor significantly influences the outcome, or for initial exploratory analysis of pairwise relationships

> Multiple Linear Regression (MLR): Involves two or more independent variables

> MLR equat is similar and goes like `Y=Beta0 + Beta1X1 + Beta2X2 + Beta3X3 + .... + BetanXn + constant` where X1 ... Xn are the multiple independent variables

> Main purpose of using this to model the linear relationship between a dependent variable and several predictors, allowing for a more comprehensive and often more accurate prediction by considering multiple influencing factors simultaneously.

> use cases: predicting house prices based on square footage, number of bedrooms, number of bathrooms, and location. Delivery time prediction model is a classic example of MLR.

**Question 8:What is the role of the cost function in linear regression, and how is it minimized?**

> The main goal of cost function is to predict the errors between the values predicted and the actual vales. This is created to evaluate the model efficiency. And the process involves minimizing the cost function

> Ways to minimize the cost function (RSS)

- Differentiation: e.g Ordinary Least Squares (OLS)

- Gradient Descent Appxoach: e.g R2

> Find the optimized Beta0, Beta1 and use it in the susequent prediction

> Residual Sum of Squares (RSS) will be minimal for the best fit line, and Total sum of squares (TSS) will be higher

> R2 = 1 - (RSS / TSS)

**Question 9:Explain the difference between overfitting and underfitting.**

> When we hear Model may ‘overfit’ means that Model fits the train set ‘too well’, doesn’t generalize and does not predict well in unknow dataset.

> It might give right data in the training dataset, while in test data it might not perform well.

> It essentially memorizes the training data instaed of generalizing

> Common cause : Complex model, noisy, insufficient data set, making it too much perfect.

> it is just opposite of overfit, Underfitting means the model is too simple to capture relationships in the training data. It fails to learn from the data adequately.

> When if we have the right data in the training dataset, the model does understand the varies varibles and contribute

> Common cause : Insufficient features, model is simple, Over-regularization

**Question 10:How do residual plots help in diagnosing a linear regression model?**

> Residual plots are used for diagnosing linear regression models. They visually represent the errors (residuals) of the model, allowing to check whether the underlying assumptions of Ordinary Least Squares (OLS) regression hold true. Residuals vs. Predicted Values Plot

> A random scatter of points around the horizontal line at zero

> Common Problems (Patterns) to help understand the model

- Curved/Non-linear Pattern (e.g., U-shape, inverted U-shape): Indicates that the relationship between the variables is not linear, and a linear model is likely missing an important curve or trend. The model might be biased

- Funnel/Cone Shape: Indicates heteroscedasticity The spread of the residuals either increases or decreases as the predicted values change.