**DELIVERY TIME PREDICTION: LINEAR REGRESSION MODEL**

****

**Prepared by: Tarun Mondal**

**Table of Contents**

[Objective](#_n2yopppgmbr5) 3

[Methodology](#_4al1ey1rmq13) 3

[End-to-end process with solution architecture](#_8n8seygnqb6h) 3

[Importing libraries in Python](#_j4er0b6hqjd3) 3

[Data exploration](#_nq6j5jgfuivr) 4

[Data preprocessing](#_5bdlpe9xqeme) 5

[Exploratory data analysis (EDA)](#_4j1he0cm1jgd) 6

[Model development](#_xxsr8bt48lqc) 8

[Time Taken](#_pak2qmbqkyoh) 10

[Challenges faced](#_n66ncrpgyp64) 10

[Complexity level](#_dhl7r1wlxkiy) 10

# Objective

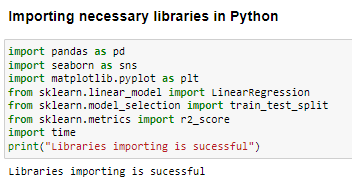
The objective of this project is to predict the Delivery Time based on Sorting Time.

# Methodology

The main purpose of this project is to predict delivery time using sorting time based on a fictional dataset. The dataset contains 21 observations and 2 variables. A machine learning approach has been adopted for predicting Delivery time. The target variable in this project is ‘numerical (continuous)’, suggesting the selection of a regression model for predicting the target variable (Delivery Time). Due to the presence of a single feature (independent variable) (‘Sorting Time’), a simple linear regression model has been developed for predicting ‘Delivery Time’ using Python programming language in Jupyter Notebook IDE.

# End-to-end process with solution architecture

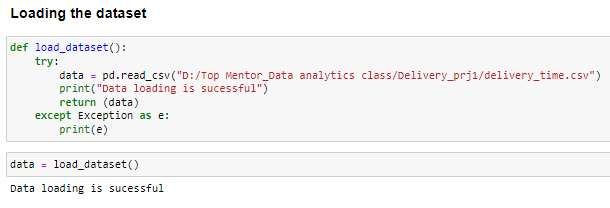
## Importing libraries in Python

****

**Figure 1: Importing libraries in Python**

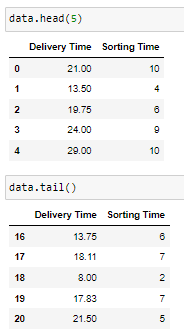
Pandas library has been imported into Python for data loading and manipulation, whereas Seaborn and Matplotlib libraries have been used for data visualisations. For developing machine learning model (Linear regression), ‘LinearRegression’ module has been imported from scikit-learn framework, whereas the evaluation metric (R2-score) has been imported from ‘sklearn.metrics module.

## Data exploration

****

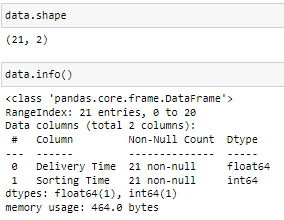
**Figure 2: Loading the dataset**

The dataset has been loaded in Python (Jupyter Notebook Environment) using the ‘read’ function from pandas library. Try-catch block has been used for better handling of errors.

****

**Figure 3: Head and tail of the dataset**

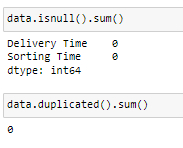
**Figure 3** shows the head and tail of the ‘delivery\_time’ dataset, which has been calculated using the ‘head ()’ and ‘tail ()’ functions from pandas library.

****

**Figure 4: Shape and info of the dataset**

The shape and info of the dataset revealed that the dataset contains 21 valid observations and 2 variables (**Refer to Figure 4**).

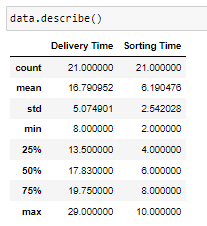
## Data preprocessing

****

**Figure 5: Data preprocessing**

Missing values in the dataset have been checked using ‘isnull (). sum ()’ function, from which it can be observed that the dataset contains 0 missing values for both variables. This indicates that data errors are absent in the dataset, thus, data cleaning steps like dropping null values or filling the null values are not required in this project. Additionally, duplicate values have been checked and no duplicate values have been identified.

## Exploratory data analysis (EDA)

****

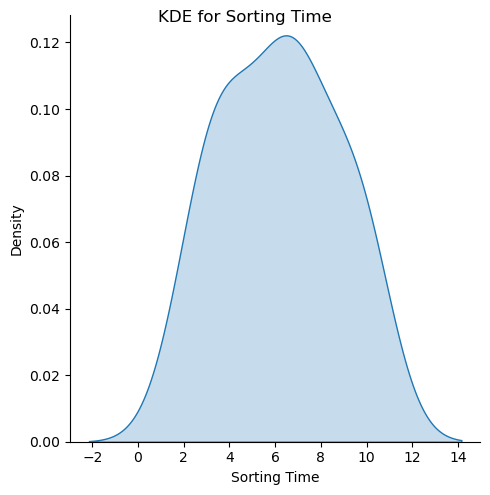
**Figure 6: Summary statistics**

The summary statistics show the central tendencies and variability in the dataset, from which it can be observed that the mean value of delivery time is 16.79 with a standard deviation of 5.07, indicating a substantial variation in delivery time. On the other hand, mean value of Sorting Time is 6.19 with a standard deviation of 2.54, which is considerably low.

****

**Figure 7: Distribution plot for ‘Delivery Time’**

**Figure 7** shows the distribution plot (kernel density plot) of ‘Delivery Time’ from which it can be observed that distribution of ‘Delivery Time’ has followed a normal distribution (bell-shaped curve).

****

**Figure 8: Distribution plot for Sorting Time**

**Figure 8** shows the distribution plot (kernel density plot) of ‘Sorting Time’ from which it can be observed that distribution of ‘Sorting Time’ has followed a normal distribution (bell-shaped curve).

****

**Figure 9: Association between delivery time and sorting time**

The scatter plot between delivery time and sorting time revealed an approximately positive relationship between these two factors, where delivery time has increased with the increase in sorting time (**Refer to Figure 9**).

## Model development

****

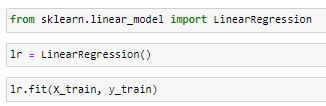
**Figure 10: Feature and target**

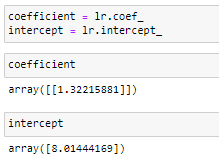
Feature and target variable have been selected and reshaped to (-1,1) to convert them into vertical arrays.

****

**Figure 11: train-test splitting**

70-30% splitting has been performed by using the train\_test\_split function from ‘scikit-learn’ framework, through which 70% of data has been used to train the model and remaining 30% of data has been used to test the model performance. A random state of 1234 has been considered to maintain the same output across multiple runs.

****

****

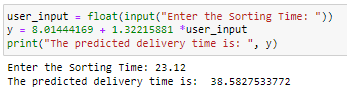
**Figure 12: Model architecture**

The coefficient and intercept obtained from the simple linear regression model are 1.32215 and 8.01444. Thus, the equation of the regression line is: Delivery Time = 8.01444 + (1.32215 \* Sorting Time).

****

**Figure 13: R-square**

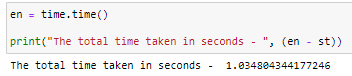
The R-square of the regression model is 0.6317, indicating that within the model, feature (Sorting Time) has explained almost 63.17% variability in the target variable (Delivery Time).

****

**Figure 14: Prediction on a sample test case**

For a sorting time of 23.12, the predicted delivery time is 38.58 (Refer to Figure 14). Using this model, users can estimate the delivery time based on a given sorting time, allowing them to make informed decisions regarding delivery schedules.

# Time Taken

****

**Figure 15: Time taken**

The total time elapsed for the complete execution of the code is 1.03 seconds, indicating high speed of the system in estimating delivery time.

# Challenges faced

Challenges have been faced in this project in identifying the appropriate data-splitting ratio.

# Complexity level

The dataset is very small and contains only 2 variables with no data errors (such as missing values, duplicate values and outliers), reflecting the project is very simple. However, due to the robustness of the model and application of a proper Machine Learning flow, this project can also apply to large datasets.