Q1. Business Case: Delhivery - Feature Engineering

About Delhivery

Delhivery is the largest and fastest-growing fully integrated player in India by revenue in Fiscal 2021. They aim to build the operating system for commerce, through a combination of world-class infrastructure, logistics operations of the highest quality, and cutting-edge engineering and technology capabilities.

The Data team builds intelligence and capabilities using this data that helps them to widen the gap between the quality, efficiency, and profitability of their business versus their competitors.

How can you help here?

The company wants to understand and process the data coming out of data engineering pipelines:

• Clean, sanitize and manipulate data to get useful features out of raw fields

• Make sense out of the raw data and help the data science team to build forecasting models on it

Dataset

Dataset Link: delhivery_data.csv

https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/551/original/delhivery_data.csv?1642751181

Column Profiling:

- data - tells whether the data is testing or training data
- trip_creation_time – Timestamp of trip creation
- route_schedule_uuid – Unique Id for a particular route schedule
- route_type – Transportation type
    - FTL – Full Truck Load: FTL shipments get to the destination sooner, as the truck is making no other pickups or drop-offs along the way
    - Carting: Handling system consisting of small vehicles (carts)
- trip_uuid - Unique ID given to a particular trip (A trip may include different source and destination centers)
- source_center - Source ID of trip origin
- source_name - Source Name of trip origin
- destination_cente – Destination ID
- destination_name – Destination Name
- od_start_time – Trip start time
- od_end_time – Trip end time
- start_scan_to_end_scan – Time taken to deliver from source to destination
- is_cutoff – Unknown field
- cutoff_factor – Unknown field
- cutoff_timestamp – Unknown field
- actual_distance_to_destination – Distance in Kms between source and destination warehouse
- actual_time – Actual time taken to complete the delivery (Cumulative)
- osrm_time – An open-source routing engine time calculator which computes the shortest path between points in a given map (Includes usual traffic, distance through major and minor roads) and gives the time (Cumulative)
- osrm_distance – An open-source routing engine which computes the shortest path between points in a given map (Includes usual traffic, distance through major and minor roads) (Cumulative)
- factor – Unknown field
- segment_actual_time – This is a segment time. Time taken by the subset of the package delivery
- segment_osrm_time – This is the OSRM segment time. Time taken by the subset of the package delivery
- segment_osrm_distance – This is the OSRM distance. Distance covered by subset of the package delivery
- segment_factor – Unknown field

Concept Used:

- Feature Creation

- Relationship between Features
- Column Normalization /Column Standardization
- Handling categorical values
- Missing values - Outlier treatment / Types of outliers

How to begin:

Since delivery details of one package are divided into several rows (think of it as connecting flights to reach a particular destination). Now think about how we should treat their fields if we combine these rows? What aggregation would make sense if we merge. What would happen to the numeric fields if we merge the rows?

Hint: *You can use inbuilt functions like groupby and aggregations like sum(), cumsum() to merge some rows based on their 1. Trip_uuid, Source ID and Destination ID 2. Further aggregate on the basis of just Trip_uuid. You can also keep the first and last values for some numeric/categorical fields if aggregating them won't make sense.*

1. **Basic data cleaning and exploration:**
    - Handle missing values in the data.
    - Analyze the structure of the data.
    - Try merging the rows using the hint mentioned above.
2. **Build some features to prepare the data for actual analysis. Extract features from the below fields:**
    - Destination Name: Split and extract features out of destination. City-place-code (State)
    - Source Name: Split and extract features out of destination. City-place-code (State)
    - Trip_creation_time: Extract features like month, year and day etc
3. **In-depth analysis and feature engineering:**
    - Calculate the time taken between od_start_time and od_end_time and keep it as a feature. Drop the original columns, if required
    - Compare the difference between Point a. and start_scan_to_end_scan. Do hypothesis testing/ Visual analysis to check.
    - Do hypothesis testing/ visual analysis between actual_time aggregated value and OSRM time aggregated value *(aggregated values are the values you'll get after merging the rows on the basis of trip_uuid)*
    - Do hypothesis testing/ visual analysis between actual_time aggregated value and segment actual time aggregated value *(aggregated values are the values you'll get after merging the rows on the basis of trip_uuid)*
    - Do hypothesis testing/ visual analysis between osrm distance aggregated value and segment osrm distance aggregated value *(aggregated values are the values you'll get after merging the rows on the basis of trip_uuid)*
    - Do hypothesis testing/ visual analysis between osrm time aggregated value and segment osrm time aggregated value *(aggregated values are the values you'll get after merging the rows on the basis of trip_uuid)*
    - Find outliers in the numerical variables (you might find outliers in almost all the variables), and check it using visual analysis
    - Handle the outliers using the IQR method.
    - Do one-hot encoding of categorical variables (like route_type)
    - Normalize/ Standardize the numerical features using MinMaxScaler or StandardScaler.

Evaluation Criteria (100 Points):

1. Define Problem Statement and perform Exploratory Data Analysis **(10 points)**
    - Definition of problem (as per given problem statement with additional views)
    - Observations on shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), missing value detection, statistical summary.
    - Visual Analysis (distribution plots of all the continuous variable(s), boxplots of all the categorical variables)
    - Insights based on EDA
        - Comments on range of attributes, outliers of various attributes
        - Comments on the distribution of the variables and relationship between them
        - Comments for each univariate and bivariate plot
2. Feature Creation **(10 Points)**
3. Merging of rows and aggregation of fields **(10 Points)**
4. Comparison & Visualization of time and distance fields **(10 Points)**
5. Missing values Treatment & Outlier treatment **(10 Points)**
6. Checking relationship between aggregated fields **(10 Points)**
7. Handling categorical values **(10 Points)**
8. Column Normalization /Column Standardization **(10 Points)**
9. Business Insights **(10 Points)** - Should include patterns observed in the data along with what you can infer from it. Eg:
    - Check from where most orders are coming from (State, Corridor etc)
    - Busiest corridor, avg distance between them, avg time taken
10. Recommendations **(10 Points)** - Actionable items for business. No technical jargon. No complications. Simple action items that everyone can understand.