In [1]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from scipy.stats import stats
```

In [2]:

```python
data = pd.read_csv("../Feature Engineering project/delhivery_data.csv")
data.head(5)
```

Out[2]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	source_c
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812

5 rows × 24 columns

Basic EDA

In [3]:

```
print("------Column Names-------")
for i in data.columns:
    print(i)
```

```
------Column Names-------
data
trip_creation_time
route_schedule_uuid
route_type
trip_uuid
source_center
source_name
destination_center
destination_name
od_start_time
od_end_time
start_scan_to_end_scan
is_cutoff
cutoff_factor
cutoff_timestamp
actual_distance_to_destination
actual_time
osrm_time
osrm_distance
factor
segment_actual_time
segment_osrm_time
segment_osrm_distance
segment_factor
```

In [4]:

```
data.shape
```

Out[4]:

```
(144867, 24)
```

In [5]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 144867 entries, 0 to 144866
Data columns (total 24 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   data                            144867 non-null  object 
 1   trip_creation_time              144867 non-null  object 
 2   route_schedule_uuid             144867 non-null  object 
 3   route_type                      144867 non-null  object 
 4   trip_uuid                       144867 non-null  object 
 5   source_center                   144867 non-null  object 
 6   source_name                     144574 non-null  object 
 7   destination_center              144867 non-null  object 
 8   destination_name                144606 non-null  object 
 9   od_start_time                   144867 non-null  object 
 10  od_end_time                     144867 non-null  object 
 11  start_scan_to_end_scan          144867 non-null  float64
 12  is_cutoff                       144867 non-null  bool   
 13  cutoff_factor                   144867 non-null  int64  
 14  cutoff_timestamp                144867 non-null  object 
 15  actual_distance_to_destination  144867 non-null  float64
 16  actual_time                     144867 non-null  float64
 17  osrm_time                       144867 non-null  float64
 18  osrm_distance                   144867 non-null  float64
 19  factor                          144867 non-null  float64
 20  segment_actual_time             144867 non-null  float64
 21  segment_osrm_time               144867 non-null  float64
 22  segment_osrm_distance           144867 non-null  float64
 23  segment_factor                  144867 non-null  float64
dtypes: bool(1), float64(10), int64(1), object(12)
memory usage: 25.6+ MB
```

In [6]:

```
print("---------Data type of Columns---------")
for i in data.columns:
    print(i,':', data[i].dtypes)
```

```
---------Data type of Columns---------
data : object
trip_creation_time : object
route_schedule_uuid : object
route_type : object
trip_uuid : object
source_center : object
source_name : object
destination_center : object
destination_name : object
od_start_time : object
od_end_time : object
start_scan_to_end_scan : float64
is_cutoff : bool
cutoff_factor : int64
cutoff_timestamp : object
actual_distance_to_destination : float64
actual_time : float64
osrm_time : float64
osrm_distance : float64
factor : float64
segment_actual_time : float64
segment_osrm_time : float64
segment_osrm_distance : float64
segment_factor : float64
```

In [7]:

```
print("-------------Uniquer values in the Columns------------")
for i in data.columns:
    print(i,':',data[i].nunique())
```

```
-------------Uniquer values in the Columns------------
data : 2
trip_creation_time : 14817
route_schedule_uuid : 1504
route_type : 2
trip_uuid : 14817
source_center : 1508
source_name : 1498
destination_center : 1481
destination_name : 1468
od_start_time : 26369
od_end_time : 26369
start_scan_to_end_scan : 1915
is_cutoff : 2
cutoff_factor : 501
cutoff_timestamp : 93180
actual_distance_to_destination : 144515
actual_time : 3182
osrm_time : 1531
osrm_distance : 138046
factor : 45641
segment_actual_time : 747
segment_osrm_time : 214
segment_osrm_distance : 113799
segment_factor : 5675
```

In [8]:

```
print("-------------Null values in the Columns-------------")
data.isna().sum()
```

```
-------------Null values in the Columns-------------
```

Out[8]:

```
data                                0
trip_creation_time                  0
route_schedule_uuid                 0
route_type                          0
trip_uuid                           0
source_center                       0
source_name                       293
destination_center                  0
destination_name                  261
od_start_time                       0
od_end_time                         0
start_scan_to_end_scan              0
is_cutoff                           0
cutoff_factor                       0
cutoff_timestamp                    0
actual_distance_to_destination      0
actual_time                         0
osrm_time                           0
osrm_distance                       0
factor                              0
segment_actual_time                 0
segment_osrm_time                   0
segment_osrm_distance               0
segment_factor                      0
dtype: int64
```

In [9]:

```
data = data.dropna(how='any')
data = data.reset_index(drop=True)
# data
```

In [10]:
```
print("-------------Null values in the Columns after drooping NA -------------
data.isna().sum()
```

```
-------------Null values in the Columns after drooping NA -------------
```

Out[10]:
```
data                              0
trip_creation_time                0
route_schedule_uuid               0
route_type                        0
trip_uuid                         0
source_center                     0
source_name                       0
destination_center                0
destination_name                  0
od_start_time                     0
od_end_time                       0
start_scan_to_end_scan            0
is_cutoff                         0
cutoff_factor                     0
cutoff_timestamp                  0
actual_distance_to_destination    0
actual_time                       0
osrm_time                         0
osrm_distance                     0
factor                            0
segment_actual_time               0
segment_osrm_time                 0
segment_osrm_distance             0
segment_factor                    0
dtype: int64
```

In [11]:
```
data["trip_creation_time"] = pd.to_datetime(data["trip_creation_time"])
data["od_start_time"] = pd.to_datetime(data["od_start_time"])
data["od_end_time"] = pd.to_datetime(data["od_end_time"])
data["cutoff_timestamp"] = pd.to_datetime(data["cutoff_timestamp"])
print("---------Data types of Columns after changing the data type of column--
for i in data.columns:
    print(i,':', data[i].dtypes)
```

```
---------Data types of Columns after changing the data type of column--------
-
data : object
trip_creation_time : datetime64[ns]
route_schedule_uuid : object
route_type : object
trip_uuid : object
source_center : object
source_name : object
destination_center : object
destination_name : object
od_start_time : datetime64[ns]
od_end_time : datetime64[ns]
start_scan_to_end_scan : float64
is_cutoff : bool
cutoff_factor : int64
cutoff_timestamp : datetime64[ns]
actual_distance_to_destination : float64
actual_time : float64
osrm_time : float64
osrm_distance : float64
factor : float64
segment_actual_time : float64
segment_osrm_time : float64
segment_osrm_distance : float64
segment_factor : float64
```

In [12]:
```
data["trip_creation_time"].dt.month_name().value_counts()
```

Out[12]:
```
September    126932
October       17384
Name: trip_creation_time, dtype: int64
```

In [13]:
```
data["trip_creation_time"].dt.year.value_counts()
```

Out[13]:
```
2018    144316
Name: trip_creation_time, dtype: int64
```

In [14]:

```
data["trip_creation_time"].dt.day_name().value_counts()
```

Out[14]:

```
Wednesday    26634
Thursday     20422
Friday       20177
Saturday     19874
Tuesday      19858
Monday       19540
Sunday       17811
Name: trip_creation_time, dtype: int64
```

In [15]:

```
data.describe(include = "all")
```

Out[15]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	sou
count	144316	144316	144316	144316	144316	
unique	2	14787	1497	2	14787	
top	training	2018-10-01 05:04:55.268931	thanos::sroute:4029a8a2-6c74-4b7e-a6d8-f9e069f...	FTL	trip-153837029526866991	IND
freq	104632	101	1812	99132	101	
first	NaN	2018-09-12 00:00:16.535741	NaN	NaN	NaN	
last	NaN	2018-10-03 23:59:42.701692	NaN	NaN	NaN	
mean	NaN	NaN	NaN	NaN	NaN	
std	NaN	NaN	NaN	NaN	NaN	
min	NaN	NaN	NaN	NaN	NaN	
25%	NaN	NaN	NaN	NaN	NaN	
50%	NaN	NaN	NaN	NaN	NaN	
75%	NaN	NaN	NaN	NaN	NaN	
max	NaN	NaN	NaN	NaN	NaN	

13 rows × 24 columns

2.Merging the rows

Grouping by segment

In [16]:

```
data['segment_key'] = data['trip_uuid'] + data['source_center'] + data['destin

segment_cols = ['segment_actual_time', 'segment_osrm_distance', 'segment_osrm_

for col in segment_cols:
    data[col + '_sum'] = data.groupby('segment_key')[col].cumsum()
data
```

Out[16]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	sou
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND3
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND3
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND3
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND3
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND3
...	...	...	...	...	...	
144311	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND1
144312	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND1
144313	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND1
144314	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND1
144315	training	2018-09-20 16:24:28.436231	thanos::sroute:f0569d2f-4e20-4c31-8542-67b86d5...	Carting	trip-153746066843555182	IND1

144316 rows × 28 columns

Aggregating at segment level

In [17]:

```
create_segment_dict = {

    'data' : 'first',
    'trip_creation_time': 'first',
    'route_schedule_uuid' : 'first',
    'route_type' : 'first',
    'trip_uuid' : 'first',
    'source_center' : 'first',
    'source_name' : 'first',

    'destination_center' : 'last',
    'destination_name' : 'last',

    'od_start_time' : 'first',
    'od_end_time' : 'first',
    'start_scan_to_end_scan' : 'first',


    'actual_distance_to_destination' : 'last',
    'actual_time' : 'last',

    'osrm_time' : 'last',
    'osrm_distance' : 'last',

    'segment_actual_time_sum' : 'last',
    'segment_osrm_distance_sum' : 'last',
    'segment_osrm_time_sum' : 'last',

    }
```

In [18]:

```
segment = data.groupby('segment_key').agg(create_segment_dict).reset_index()
segment = segment.sort_values(by=['segment_key','od_end_time'], ascending=True
```

In [19]:
```
segment.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26222 entries, 0 to 26221
Data columns (total 21 columns):
 #   Column                          Non-Null Count  Dtype         
---  ------                          --------------  -----         
 0   index                           26222 non-null  int64         
 1   segment_key                     26222 non-null  object        
 2   data                            26222 non-null  object        
 3   trip_creation_time              26222 non-null  datetime64[ns]
 4   route_schedule_uuid             26222 non-null  object        
 5   route_type                      26222 non-null  object        
 6   trip_uuid                       26222 non-null  object        
 7   source_center                   26222 non-null  object        
 8   source_name                     26222 non-null  object        
 9   destination_center              26222 non-null  object        
 10  destination_name                26222 non-null  object        
 11  od_start_time                   26222 non-null  datetime64[ns]
 12  od_end_time                     26222 non-null  datetime64[ns]
 13  start_scan_to_end_scan          26222 non-null  float64       
 14  actual_distance_to_destination  26222 non-null  float64       
 15  actual_time                     26222 non-null  float64       
 16  osrm_time                       26222 non-null  float64       
 17  osrm_distance                   26222 non-null  float64       
 18  segment_actual_time_sum         26222 non-null  float64       
 19  segment_osrm_distance_sum       26222 non-null  float64       
 20  segment_osrm_time_sum           26222 non-null  float64       
dtypes: datetime64[ns](3), float64(8), int64(1), object(9)
memory usage: 4.2+ MB
```

3. Feature Engineering:

1. Calculate time taken between od_start_time and od_end_time and keep it as a feature named od_time_diff_hour.

In [20]:
```
segment['od_time_diff_hour'] = (segment['od_end_time'] - segment['od_start_tim
segment['od_time_diff_hour']
```

Out[20]:
```
0        1260.604421
1         999.505379
2          58.832388
3         122.779486
4         834.638929
            ...     
26217      62.115193
26218      91.087797
26219      44.174403
26220     287.474007
26221      66.933565
Name: od_time_diff_hour, Length: 26222, dtype: float64
```

In [21]:
```
segment.head()
```

Out[21]:

	index	segment_key	data	trip_creation_time	route_s
0	0	trip-153671041653548748IND209304AAAIND000000ACB	training	2018-09-12 00:00:16.535741	thanos::sro a2
1	1	trip-153671041653548748IND462022AAAIND209304AAA	training	2018-09-12 00:00:16.535741	thanos::sro a2
2	2	trip-153671042288605164IND561203AABIND562101AAA	training	2018-09-12 00:00:22.886430	thanos::sro bb
3	3	trip-153671042288605164IND572101AAAIND561203AAB	training	2018-09-12 00:00:22.886430	thanos::sro bb
4	4	trip-153671043369099517IND000000ACBIND160002AAC	training	2018-09-12 00:00:33.691250	thanos::sro 76

5 rows × 22 columns

2. Destination Name: Split and extract features out of destination. City-place-code (State)

In [22]:
```
data["source_city"] = data["source_name"].str.split(" ",n=1,expand=True)[0].st
data["source_state"] = data["source_name"].str.split(" ",n=1,expand=True)[1].s

data["destination_city"] = data["destination_name"].str.split(" ",n=1,expand=T
data["destination_state"] = data["destination_name"].str.split(" ",n=1,expand=

data["source_pincode"] = data["source_center"].apply(lambda x : x[3:9] )
data["destination_pincode"] = data["destination_center"].apply(lambda x : x[3:
```

In [23]:
```
data.head()
# data
```

Out[23]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	source_c
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812

5 rows × 34 columns

In [24]:
```
# data['trip_creation_time'] = pd.to_datetime(data['trip_creation_time'])

# Extract features
data['year'] = data['trip_creation_time'].dt.year
data['month'] = data['trip_creation_time'].dt.month
data['day'] = data['trip_creation_time'].dt.day
data['hour'] = data['trip_creation_time'].dt.hour
```

In [25]:
```
data.head()
```

Out[25]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	source_c
0	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812
1	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812
2	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812
3	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812
4	training	2018-09-20 02:35:36.476840	thanos::sroute:eb7bfc78-b351-4c0e-a951-fa3d5c3...	Carting	trip-153741093647649320	IND38812

5 rows × 38 columns

4. In-depth analysis:

1. Grouping and Aggregating at Trip-level

In [26]:
```
create_trip_dict = {

    'data' : 'first',
    'trip_creation_time': 'first',
    'route_schedule_uuid' : 'first',
    'route_type' : 'first',
    'trip_uuid' : 'first',

    'source_center' : 'first',
    'source_name' : 'first',

    'destination_center' : 'last',
    'destination_name' : 'last',

    'start_scan_to_end_scan' : 'sum',
    'od_time_diff_hour' : 'sum',

    'actual_distance_to_destination' : 'sum',
    'actual_time' : 'sum',
    'osrm_time' : 'sum',
    'osrm_distance' : 'sum',

    'segment_actual_time_sum' : 'sum',
    'segment_osrm_distance_sum' : 'sum',
    'segment_osrm_time_sum' : 'sum',

    }
```

In [27]:
```
trip = segment.groupby('trip_uuid').agg(create_trip_dict).reset_index(drop = T
```

In [28]:
```
trip.head()
```

Out[28]:

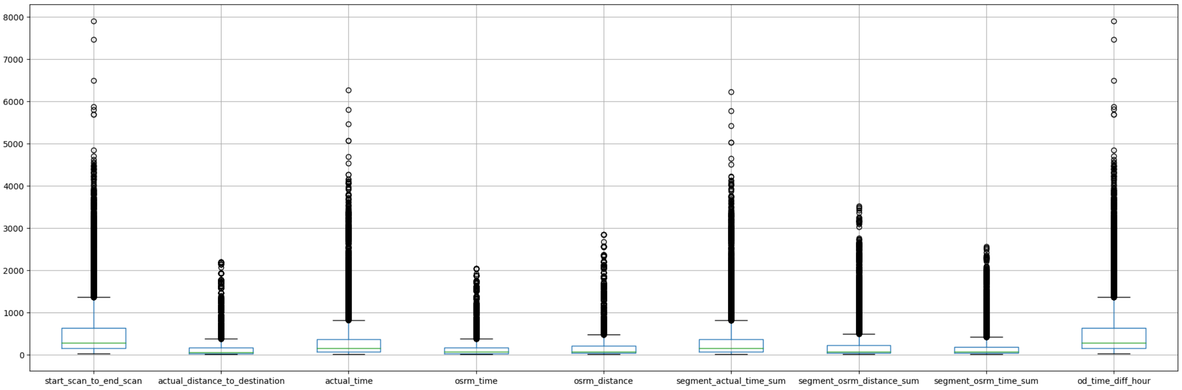
	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	source_c
0	training	2018-09-12 00:00:16.535741	thanos::sroute:d7c989ba-a29b-4a0b-b2f4-288cdc6...	FTL	trip-153671041653548748	IND20930
1	training	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting	trip-153671042288605164	IND56120
2	training	2018-09-12 00:00:33.691250	thanos::sroute:de5e208e-7641-45e6-8100-4d9fb1e...	FTL	trip-153671043369099517	IND00000
3	training	2018-09-12 00:01:00.113710	thanos::sroute:f0176492-a679-4597-8332-bbd1c7f...	Carting	trip-153671046011330457	IND40007
4	training	2018-09-12 00:02:09.740725	thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134...	FTL	trip-153671052974046625	IND58310

2. Outlier Detection & Treatment

```
In [29]: num_cols = ['start_scan_to_end_scan', 'actual_distance_to_destination', 'actua
                     'osrm_distance', 'segment_actual_time_sum', 'segment_osrm_distance
                     'segment_osrm_time_sum', 'od_time_diff_hour']
```

```
In [30]: trip[num_cols].boxplot(figsize=(25,8))
```

```
Out[30]: <Axes: >
```



```
In [31]: # Handle the outliers using the IQR method.
         Q1 = trip[num_cols].quantile(0.25)
         Q3 = trip[num_cols].quantile(0.75)

         IQR = Q3 - Q1
         IQR
```

```
Out[31]: start_scan_to_end_scan            483.000000
         actual_distance_to_destination    140.814159
         actual_time                       300.000000
         osrm_time                         139.000000
         osrm_distance                     175.887300
         segment_actual_time_sum           298.000000
         segment_osrm_distance_sum         183.981750
         segment_osrm_time_sum             154.000000
         od_time_diff_hour                 483.839201
         dtype: float64
```

```
In [32]: trip = trip[~((trip[num_cols] < (Q1 - 1.5 * IQR)) | (trip[num_cols] > (Q3 + 1.
         trip = trip.reset_index(drop=True)
```

```
In [33]: trip[num_cols].boxplot(figsize=(25,8))
```

```
Out[33]: <Axes: >
```

In [34]:
```
trip
```

Out[34]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	sou
0	training	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	Carting	trip-153671042288605164	IND5
1	training	2018-09-12 00:01:00.113710	thanos::sroute:f0176492-a679-4597-8332-bbd1c7f...	Carting	trip-153671046011330457	IND4
2	training	2018-09-12 00:02:09.740725	thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134...	FTL	trip-153671052974046625	IND5
3	training	2018-09-12 00:02:34.161600	thanos::sroute:9bf03170-d0a2-4a3f-aa4d-9aaab3d...	Carting	trip-153671055416136166	IND6
4	training	2018-09-12 00:04:22.011653	thanos::sroute:a97698cc-846e-41a7-916b-88b1741...	Carting	trip-153671066201138152	IND6
...	...	...	...	...	...	
12718	test	2018-10-03 23:55:56.258533	thanos::sroute:8a120994-f577-4491-9e4b-b7e4a14...	Carting	trip-153861095625827784	IND1
12719	test	2018-10-03 23:57:23.863155	thanos::sroute:b30e1ec3-3bfa-4bd2-a7fb-3b75769...	Carting	trip-153861104386292051	IND1
12720	test	2018-10-03 23:57:44.429324	thanos::sroute:5609c268-e436-4e0a-8180-3db4a74...	Carting	trip-153861106442901555	IND2
12721	test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	Carting	trip-153861115439069069	IND6
12722	test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	FTL	trip-153861118270144424	IND5

12723 rows × 18 columns

5. Hypothesis Testing

Perform hypothesis testing / visual analysis

In [35]:

```
# actual_time aggregated value and OSRM time aggregated value.

from scipy.stats import ttest_ind
null_hypothesis = 'mean of actual_time is not higher than mean of osrm_time'
alternative_hypothesis = 'mean of actual_time is higher than mean of osrm_time
sample1 = trip['actual_time']
sample2 = trip['osrm_time']
t_stat, p_value = ttest_ind(sample1, sample2, equal_var=False, alternative='gr
print(t_stat, p_value)

if(p_value < 0.05):
    print('Since, p-value < 0.05, the null hypothesis is rejected')
    print(alternative_hypothesis)
else:
    print('Since p-value > 0.05, we fail to reject null hypothesis')
    print(null_hypothesis)
```

```
64.21934953647681 0.0
Since, p-value < 0.05, the null hypothesis is rejected
mean of actual_time is higher than mean of osrm_time
```

In [49]:

```
# actual_time aggregated value and segment actual time aggregated value.

from scipy.stats import ttest_ind
null_hypothesis = 'mean of actual_time is similar as segment_actual_time'
alternative_hypothesis = 'mean of actual_time is different than mean of segmen
sample1 = trip['actual_time']
sample2 = trip['segment_actual_time_sum']
t_stat, p_value = ttest_ind(sample1, sample2)
print(t_stat, p_value)
if(p_value < 0.05):
 print('Since, p-value < 0.05, the null hypothesis is rejected')
 print(alternative_hypothesis)
else:
 print('Since p-value > 0.05, we fail to reject null hypothesis')
 print(null_hypothesis)
```

```
0.8381648951065266 0.40194597338420224
Since p-value > 0.05, we fail to reject null hypothesis
mean of actual_time is similar as segment_actual_time
```

In [48]:
```
# OSRM distance aggregated value and segment OSRM distance aggregated value.

from scipy.stats import ttest_ind
null_hypothesis = 'mean of osrm_distance is similar as mean of segment_osrm_di
alternative_hypothesis = 'mean of osrm_distance is higher than mean of segment
sample1 = trip['osrm_distance']
sample2 = trip['segment_osrm_distance_sum']
t_stat, p_value = ttest_ind(sample1, sample2, equal_var=False, alternative='gr
print(t_stat, p_value)
if(p_value < 0.05):
 print('Since, p-value < 0.05, the null hypothesis is rejected')
 print(alternative_hypothesis)
else:
 print('Since p-value > 0.05, we fail to reject null hypothesis')
 print(null_hypothesis)
```

```
-5.394101351961479 0.9999999652583499
Since p-value > 0.05, we fail to reject null hypothesis
mean of osrm_distance is similar as mean of segment_osrm_distance
```

In [47]:
```
# OSRM time aggregated value and segment OSRM time aggregated value.

from scipy.stats import ttest_ind
null_hypothesis = 'mean of osrm_time is similar as mean of segment_osrm_distan
alternative_hypothesis = 'mean of osrm_time is higher than mean of segment_osr
sample1 = trip['osrm_time']
sample2 = trip['segment_osrm_distance_sum']
t_stat, p_value = ttest_ind(sample1, sample2, equal_var=False, alternative='gr
print(t_stat, p_value)
if(p_value < 0.05):
    print('Since, p-value < 0.05, the null hypothesis is rejected')
    print(alternative_hypothesis)
else:
    print('Since p-value > 0.05, we fail to reject null hypothesis')
    print(null_hypothesis)
```

```
-18.472775559666545 1.0
Since p-value > 0.05, we fail to reject null hypothesis
mean of osrm_time is similar as mean of segment_osrm_distance
```

In [ ]:

In [ ]:

In [ ]:

3. Perform one-hot encoding on categorical features.

```
In [51]: trip_copy = trip
         trip_copy['route_type'].value_counts()
```

```
Out[51]: Carting    8812
         FTL        3911
         Name: route_type, dtype: int64
```

```
In [52]: trip_copy['route_type'] = trip_copy['route_type'].map({'FTL':0, 'Carting':1})
```

4. Normalize/ Standardize the numerical features using MinMaxScaler or StandardScaler.

```
In [53]: from sklearn.preprocessing import StandardScaler
```

```
In [54]: scaler = StandardScaler()
         scaler.fit(trip_copy[num_cols])
```

```
Out[54]: ▾ StandardScaler
         StandardScaler()
```

```
In [58]: trip_copy[num_cols] = scaler.transform(trip_copy[num_cols])
```

```
In [59]: trip_copy[num_cols]
```

Out[59]:

	start_scan_to_end_scan	actual_distance_to_destination	actual_time	osrm_time	osrm_dist
0	-1.255068	-1.003307	-1.123469	-1.086461	-1.02
1	-1.256293	-1.014092	-1.126828	-1.096592	-1.03
2	-1.246845	-0.992860	-1.115552	-1.077095	-1.01
3	-1.254930	-1.012663	-1.126748	-1.095063	-1.03
4	-1.256323	-1.015647	-1.128227	-1.096974	-1.03
...	...	...	...	...	
12718	-1.253889	-1.006277	-1.125868	-1.087608	-1.02
12719	-1.256905	-1.014412	-1.128347	-1.097165	-1.03
12720	-1.251377	-1.009950	-1.117911	-1.090284	-1.02
12721	-1.252510	-0.991459	-1.118631	-1.065245	-1.01
12722	-1.252419	-1.004675	-1.118191	-1.086461	-1.02

12723 rows × 9 columns

In [61]:
```
trip
```

Out[61]:

	data	trip_creation_time	route_schedule_uuid	route_type	trip_uuid	sour
0	training	2018-09-12 00:00:22.886430	thanos::sroute:3a1b0ab2-bb0b-4c53-8c59-eb2a2c0...	1	trip-153671042288605164	IND5
1	training	2018-09-12 00:01:00.113710	thanos::sroute:f0176492-a679-4597-8332-bbd1c7f...	1	trip-153671046011330457	IND4
2	training	2018-09-12 00:02:09.740725	thanos::sroute:d9f07b12-65e0-4f3b-bec8-df06134...	0	trip-153671052974046625	IND5
3	training	2018-09-12 00:02:34.161600	thanos::sroute:9bf03170-d0a2-4a3f-aa4d-9aaab3d...	1	trip-153671055416136166	IND6
4	training	2018-09-12 00:04:22.011653	thanos::sroute:a97698cc-846e-41a7-916b-88b1741...	1	trip-153671066201138152	IND6
...	...	...	...	...	...	
12718	test	2018-10-03 23:55:56.258533	thanos::sroute:8a120994-f577-4491-9e4b-b7e4a14...	1	trip-153861095625827784	IND1
12719	test	2018-10-03 23:57:23.863155	thanos::sroute:b30e1ec3-3bfa-4bd2-a7fb-3b75769...	1	trip-153861104386292051	IND1
12720	test	2018-10-03 23:57:44.429324	thanos::sroute:5609c268-e436-4e0a-8180-3db4a74...	1	trip-153861106442901555	IND2
12721	test	2018-10-03 23:59:14.390954	thanos::sroute:c5f2ba2c-8486-4940-8af6-d1d2a6a...	1	trip-153861115439069069	IND6
12722	test	2018-10-03 23:59:42.701692	thanos::sroute:412fea14-6d1f-4222-8a5f-a517042...	0	trip-153861118270144424	IND5

12723 rows × 18 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: