Regression Benchmark

In [1]:
```
#importing libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Importing Dataset

In [2]:
```
data = pd.read_csv('nyc_taxi_trip_duration.csv')
```

In [3]:
```
data.shape
```

Out[3]:
```
(729322, 11)
```

In [4]:
```
data.head()
```

Out[4]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration
0	id1080784	2	2016-02-29 16:40:21	2016-02-29 16:47:01	1	-73.953918	40.778873	-73.963875	40.771164	N	400
1	id0889885	1	2016-03-11 23:35:37	2016-03-11 23:53:57	2	-73.988312	40.731743	-73.994751	40.694931	N	1100
2	id0857912	2	2016-02-21 17:59:33	2016-02-21 18:26:48	2	-73.997314	40.721458	-73.948029	40.774918	N	1635
3	id3744273	2	2016-01-05 09:44:31	2016-01-05 10:03:32	6	-73.961670	40.759720	-73.956779	40.780628	N	1141
4	id0232939	1	2016-02-17 06:42:23	2016-02-17 06:56:31	1	-74.017120	40.708469	-73.988182	40.740631	N	848

In [5]:
```
data.isnull().sum()
```

Out[5]:
```
id                  0
vendor_id           0
pickup_datetime     0
dropoff_datetime    0
passenger_count     0
pickup_longitude    0
pickup_latitude     0
dropoff_longitude   0
dropoff_latitude    0
store_and_fwd_flag  0
trip_duration       0
dtype: int64
```

In [6]:
```
# creating an instance(date) of DatetimeIndex class using "pickup_datetime"
date_pick = pd.DatetimeIndex(data['pickup_datetime'])
# creating an instance(date) of DatetimeIndex class using "pickup_datetime"
date_drop = pd.DatetimeIndex(data['dropoff_datetime'])
```

In [7]:
```
# extracting new columns from "pick datetime"

# last day of year when pickup was done
data['doy_pick'] = date_pick.dayofyear

# week of year when pickup was done
data['woy_pick'] = date_pick.weekofyear

# month of year when pickup was done
data['moy_pick'] = date_pick.month

# day of week when pickup was done
data['dow_pick'] = date_pick.dayofweek
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/344715810.py:7: FutureWarning: weekofyear and week have been deprecated, please use DatetimeIndex.isocalendar().week instead, which retur
ns a Series.  To exactly reproduce the behavior of week and weekofyear and return an Index, you may call pd.Int64Index(idx.isocalendar().week)
  data['woy_pick'] = date_pick.weekofyear
```

In [8]:
```
# extracting new columns from "dropoff datetime"

# last day of year dropoff was done
data['doy_drop'] = date_drop.dayofyear

# week of year when dropoff was done
data['woy_drop'] = date_drop.weekofyear

# month of year when dropoff was done
data['moy_drop'] = date_drop.month

# day of week when dropoff was done
data['dow_drop'] = date_drop.dayofweek
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2963120698.py:7: FutureWarning: weekofyear and week have been deprecated, please use DatetimeIndex.isocalendar().week instead, which retu
rns a Series.  To exactly reproduce the behavior of week and weekofyear and return an Index, you may call pd.Int64Index(idx.isocalendar().week)
  data['woy_drop'] = date_drop.weekofyear
```

Shuffling and Creating Train and Test Set

In [9]:
```
def UVA_outlier(data, var):

    # calculating descriptives of variable
    quant25 = data[var].quantile(0.25)
    quant75 = data[var].quantile(0.75)
    IQR = quant75 - quant25
    med = data[var].median()
    whis_low = quant25-(1.5*IQR)
    whis_high = quant75+(1.5*IQR)

    ls = data.index[(data[var] < whis_low) | (data[var] > whis_high)]

    return ls
```

In [10]:
```
def remove(df,ls):
    ls = sorted(set(ls))
    df = df.drop(ls)
    return df
```

In [11]:
```
# import pdb
index_list1 = []
```

```
# for j in data.drop(['id','vendor_id','pickup_datetime','dropoff_datetime','store_and_fwd_flag'], axis=1).columns:
for j in ['trip_duration','pickup_longitude','dropoff_longitude','pickup_latitude','dropoff_latitude']:
# for j in data.columns:
#     pdb.set_trace()
    for i in [j]:
        index_list1.extend(UVA_outlier(data,i))
        data_cleaned = remove(data,index_list1)
        index_list1.clear()
```

In [12]:
```
data = data_cleaned
```

In [13]:
```
data.head()
```

Out[13]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration	doy_pick	woy_pick	moy_pick	dow
0	id1080784	2	2016-02-29 16:40:21	2016-02-29 16:47:01	1	-73.953918	40.778873	-73.963875	40.771164	N	400	60	9	2	
1	id0889885	1	2016-03-11 23:35:37	2016-03-11 23:53:57	2	-73.988312	40.731743	-73.994751	40.694931	N	1100	71	10	3	
2	id0857912	2	2016-02-21 17:59:33	2016-02-21 18:26:48	2	-73.997314	40.721458	-73.948029	40.774918	N	1635	52	7	2	
3	id3744273	2	2016-01-05 09:44:31	2016-01-05 10:03:32	6	-73.961670	40.759720	-73.956779	40.780628	N	1141	5	1	1	
4	id0232939	1	2016-02-17 06:42:23	2016-02-17 06:56:31	1	-74.017120	40.708469	-73.988182	40.740631	N	848	48	7	2	

In [14]:
```
from sklearn.utils import shuffle

# Shuffling the Dataset
data = shuffle(data, random_state = 42)

#creating 4 divisions
div = int(data.shape[0]/4)

# 3 parts to train set and 1 part to test set
train = data.loc[:3*div+1,:]
test = data.loc[3*div+1:]
```

In [15]:
```
train.head()
```

Out[15]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration	doy_pick	woy_pick	moy_pick
22461	id3544967	2	2016-01-28 16:17:40	2016-01-28 16:43:30	1	-73.990288	40.756130	-73.997154	40.722305	N	1550	28	4	1
606882	id0689371	2	2016-02-04 22:50:50	2016-02-04 23:24:14	1	-73.867104	40.767715	-73.999352	40.760342	N	2004	35	5	2
509380	id3480861	2	2016-05-12 12:13:04	2016-05-12 12:39:55	2	-73.994865	40.725689	-73.972260	40.759197	N	1611	133	19	5
105494	id1455448	2	2016-06-30 20:02:49	2016-06-30 20:04:20	2	-73.989311	40.753269	-73.990791	40.750496	N	91	182	26	6
54890	id3407284	2	2016-06-15 23:56:40	2016-06-16 00:08:50	1	-73.996964	40.720181	-73.979126	40.750042	N	730	167	24	6

In [16]:
```
test.head()
```

Out[16]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration	doy_pick	woy_pick	moy_pick
519808	id0683339	1	2016-02-17 12:16:21	2016-02-17 12:27:40	2	-73.997002	40.762810	-73.985909	40.752457	N	679	48	7	2
142513	id3393904	1	2016-01-09 23:19:02	2016-01-09 23:33:36	2	-73.986290	40.761925	-73.949242	40.771610	N	874	9	1	1
169629	id1033730	1	2016-02-17 20:09:13	2016-02-17 20:30:55	1	-73.980392	40.754139	-73.947739	40.771877	N	1302	48	7	2
246123	id1140859	2	2016-01-29 09:46:03	2016-01-29 10:05:10	1	-73.974907	40.751991	-74.002029	40.706989	N	1147	29	4	1
652889	id0258836	2	2016-04-14 10:47:17	2016-04-14 11:36:10	5	-73.873268	40.774132	-73.953018	40.772541	N	2933	105	15	4

Simple Mean (mean of trip duration)

In [17]:
```
# storing simple mean in a new column in the test set as "simple_mean"
train['simple_mean'] = train['trip_duration'].mean()
test['simple_mean'] = test['trip_duration'].mean()
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/255938964.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['simple_mean'] = train['trip_duration'].mean()
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/255938964.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['simple_mean'] = test['trip_duration'].mean()
```

In [18]:
```
#calculating mean absolute error
from sklearn.metrics import mean_absolute_error as MAE

simple_train_mean_error = MAE(train['trip_duration'] , train['simple_mean'])
simple_train_mean_error
```

Out[18]:
```
583.5166309121103
```

In [19]:
```
simple_test_mean_error = MAE(test['trip_duration'] , test['simple_mean'])
simple_test_mean_error
```

Out[19]: 567.0411896837451

Mean Trip duration with respect to vendors

In [20]:
```
vendor_type = pd.pivot_table(train, values='trip_duration', index = ['vendor_id'], aggfunc=np.mean)
vendor_type
```

Out[20]:

vendor_id	trip_duration
1	795.738098
2	1010.981462

In [21]:
```
# initializing new column to zero
train['vendor_type_mean'] = 0

# For every unique entry in vendor id
for i in train['vendor_id'].unique():
  # Assign the mean value corresponding to unique entry
  train['vendor_type_mean'][train['vendor_id'] == str(i)] = train['trip_duration'][train['vendor_id'] == str(i)].mean()
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2223692779.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['vendor_type_mean'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2223692779.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['vendor_type_mean'][train['vendor_id'] == str(i)] = train['trip_duration'][train['vendor_id'] == str(i)].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
```

In [22]:
```
test['vendor_type_mean'] = 0

# For every unique entry in vendor id
for i in test['vendor_id'].unique():
  # Assign the mean value corresponding to unique entry
  test['vendor_type_mean'][test['vendor_id'] == str(i)] = test['trip_duration'][test['vendor_id'] == str(i)].mean()
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/1302875188.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['vendor_type_mean'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/1302875188.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['vendor_type_mean'][test['vendor_id'] == str(i)] = test['trip_duration'][test['vendor_id'] == str(i)].mean()
```

In [23]:
```
#calculating mean absolute error
train_vtype_error = MAE(train['trip_duration'] , train['vendor_type_mean'] )
train_vtype_error
```

Out[23]: 911.0425272415664

In [24]:
```
test_vtype_error = MAE(test['trip_duration'] , test['vendor_type_mean'] )
test_vtype_error
```

Out[24]: 900.715975161932

Mean Trip duration with respect to month of year - pick up

In [25]:
```
Trip_month = pd.pivot_table(train, values='trip_duration', index = ['moy_pick'], aggfunc=np.mean)
Trip_month
```

Out[25]:

moy_pick	trip_duration
1	892.196075
2	856.480523
3	892.053777
4	921.990727
5	947.885673
6	955.332183

In [26]:
```
# initializing new column to zero
train['Trip_month_mean'] = 0
test['Trip_month_mean'] = 0

# For every unique entry in Outlet_Identifier
for i in train['moy_pick'].unique():
  # Assign the mean value corresponding to unique entry
  train['Trip_month_mean'][train['moy_pick'] == i] = train['trip_duration'][train['moy_pick'] == i].mean()


# For every unique entry in Outlet_Identifier
for i in test['moy_pick'].unique():
  # Assign the mean value corresponding to unique entry
  test['Trip_month_mean'][test['moy_pick'] == i] = test['trip_duration'][test['moy_pick'] == i].mean()
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/1125606743.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Trip_month_mean'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/1125606743.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Trip_month_mean'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/1125606743.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Trip_month_mean'][train['moy_pick'] == i] = train['trip_duration'][train['moy_pick'] == i].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/1125606743.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Trip_month_mean'][train['moy_pick'] == i] = train['trip_duration'][train['moy_pick'] == i].mean()
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/1125606743.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Trip_month_mean'][test['moy_pick'] == i] = test['trip_duration'][test['moy_pick'] == i].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/1125606743.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Trip_month_mean'][test['moy_pick'] == i] = test['trip_duration'][test['moy_pick'] == i].mean()
```

In [27]:
```
train_month_error = MAE(train['trip_duration'] , train['Trip_month_mean'] )
train_month_error
```

Out[27]:
```
582.7813977341748
```

In [28]:
```
#calculating mean absolute error
test_month_error = MAE(test['trip_duration'] , test['Trip_month_mean'] )
test_month_error
```

Out[28]:
```
566.3088101447672
```

Mean trip duration with respect to day of the week -pick

In [29]:
```
Trip_dow = pd.pivot_table(train, values='trip_duration', index = ['dow_pick'], aggfunc=np.mean)
Trip_dow

# initializing new column to zero
test['Trip_dow_mean'] = 0
train['Trip_dow_mean'] = 0

# For every unique entry in Outlet_Identifier
for i in train['dow_pick'].unique():
  # Assign the mean value corresponding to unique entry
  train['Trip_dow_mean'][train['dow_pick'] == i] = train['trip_duration'][train['dow_pick'] == i].mean()

# For every unique entry in Outlet_Identifier
for i in test['dow_pick'].unique():
  # Assign the mean value corresponding to unique entry
  test['Trip_dow_mean'][test['dow_pick'] == i] = test['trip_duration'][test['dow_pick'] == i].mean()
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/4165244768.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Trip_dow_mean'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/4165244768.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Trip_dow_mean'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/4165244768.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Trip_dow_mean'][train['dow_pick'] == i] = train['trip_duration'][train['dow_pick'] == i].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/4165244768.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Trip_dow_mean'][train['dow_pick'] == i] = train['trip_duration'][train['dow_pick'] == i].mean()
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/4165244768.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Trip_dow_mean'][test['dow_pick'] == i] = test['trip_duration'][test['dow_pick'] == i].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/4165244768.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Trip_dow_mean'][test['dow_pick'] == i] = test['trip_duration'][test['dow_pick'] == i].mean()
```

In [30]:
```
#calculating mean absolute error
train_dow_error = MAE(train['trip_duration'] , train['Trip_dow_mean'] )
train_dow_error
```

Out[30]:
```
581.9665110196446
```

In [31]:
```
#calculating mean absolute error
test_dow_error = MAE(test['trip_duration'] , test['Trip_dow_mean'] )
test_dow_error
```

Out[31]:
```
565.4238077668508
```

In [32]:
```
Trip_dow
```

Out[32]:

dow_pick	trip_duration
0	855.583961
1	939.529755
2	928.676655
3	959.628694
4	950.116374
5	884.212638
6	845.927257

Mean trip duration with respect to both vendors and month of year -pick

In [33]:
```
combo = pd.pivot_table(train, values = 'trip_duration', index = ['vendor_id','moy_pick'], aggfunc = np.mean)
combo
```

Out[33]:

vendor_id	moy_pick	trip_duration
1	1	793.135969
	2	750.130578
	3	766.890106
	4	795.459520
	5	831.346326
	6	838.007191
2	1	976.394012
	2	947.725381
	3	999.445695
	4	1032.667396
	5	1050.218117
	6	1058.761908

In [34]:
```
# Initiating new empty column
train['Super_mean'] = 0
test['Super_mean'] = 0

# Assigning variables to strings ( to shorten code length)
s2 = 'vendor_id'
s1 = 'moy_pick'

# For every Unique Value in s1
for i in train[s1].unique():
  # For every Unique Value in s2
  for j in train[s2].unique():
    # Calculate and Assign mean to new column, corresponding to both unique values of s1 and s2 simultaneously
    train['Super_mean'][(train[s1] == i) & (train[s2]==str(j))] = train['trip_duration'][(train[s1] == i) & (train[s2]==str(j))].mean()


# For every Unique Value in s1
for i in test[s1].unique():
  # For every Unique Value in s2
  for j in test[s2].unique():
    # Calculate and Assign mean to new column, corresponding to both unique values of s1 and s2 simultaneously
    test['Super_mean'][(test[s1] == i) & (test[s2]==str(j))] = test['trip_duration'][(test[s1] == i) & (test[s2]==str(j))].mean()
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3604637461.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Super_mean'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3604637461.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Super_mean'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3604637461.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Super_mean'][(train[s1] == i) & (train[s2]==str(j))] = train['trip_duration'][(train[s1] == i) & (train[s2]==str(j))].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3604637461.py:22: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Super_mean'][(test[s1] == i) & (test[s2]==str(j))] = test['trip_duration'][(test[s1] == i) & (test[s2]==str(j))].mean()
```

In [35]:
```
#calculating mean absolute error
train_smean_error = MAE(train['trip_duration'] , train['Super_mean'] )
train_smean_error
```

Out[35]:
```
911.0425272415664
```

In [36]:
```
#calculating mean absolute error
test_smean_error = MAE(test['trip_duration'] , test['Super_mean'] )
test_smean_error
```

Out[36]:
```
900.715975161932
```

Mean trip duration with respect to both vendors and day of week - pick

In [37]:
```
combo2 = pd.pivot_table(train, values = 'trip_duration', index = ['vendor_id','dow_pick'], aggfunc = np.mean)
```

```
combo2
```

Out[37]:

vendor_id	dow_pick	trip_duration
1	0	759.987205
	1	863.412794
	2	837.181823
	3	839.635341
	4	818.657167
	5	734.879315
	6	705.973554
2	0	937.868509
	1	1004.566555
	2	1009.253392
	3	1065.028488
	4	1064.615221
	5	1013.197161
	6	965.720997

In [38]:

```
# Initiating new empty column
test['Super_mean2'] = 0
train['Super_mean2'] = 0

# Assigning variables to strings ( to shorten code length)
s2 = 'vendor_id'
s1 = 'dow_pick'

# For every Unique Value in s1
for i in train[s1].unique():
  # For every Unique Value in s2
  for j in train[s2].unique():
    # Calculate and Assign mean to new column, corresponding to both unique values of s1 and s2 simultaneously
    train['Super_mean2'][(train[s1] == i) & (train[s2]==str(j))] = train['trip_duration'][(train[s1] == i) & (train[s2]==str(j))].mean()


# For every Unique Value in s1
for i in test[s1].unique():
  # For every Unique Value in s2
  for j in test[s2].unique():
    # Calculate and Assign mean to new column, corresponding to both unique values of s1 and s2 simultaneously
    test['Super_mean2'][(test[s1] == i) & (test[s2]==str(j))] = test['trip_duration'][(test[s1] == i) & (test[s2]==str(j))].mean()
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2622238579.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Super_mean2'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2622238579.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Super_mean2'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2622238579.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Super_mean2'][(train[s1] == i) & (train[s2]==str(j))] = train['trip_duration'][(train[s1] == i) & (train[s2]==str(j))].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2622238579.py:22: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Super_mean2'][(test[s1] == i) & (test[s2]==str(j))] = test['trip_duration'][(test[s1] == i) & (test[s2]==str(j))].mean()
```

In [39]:

```
train_smean_error2 = MAE(train['trip_duration'] , train['Super_mean2'] )
train_smean_error2
```

Out[39]:

```
911.0425272415664
```

In [40]:

```
#calculating mean absolute error
test_smean_error2 = MAE(test['trip_duration'] , test['Super_mean2'] )
test_smean_error2
```

Out[40]:

```
900.715975161932
```

Mean Trip duration with respect to month of year - drop off

In [41]:

```
Trip_month_drop = pd.pivot_table(train, values='trip_duration', index = ['moy_drop'], aggfunc=np.mean)
Trip_month_drop
```

Out[41]:

moy_drop	trip_duration
1	887.265796
2	856.383882
3	892.838883
4	921.098969
5	950.823898
6	953.016083
7	9996.900000

In [42]:

```
# initializing new column to zero
train['Trip_month_drop_mean'] = 0
```

```
test['Trip_month_drop_mean'] = 0

# For every unique entry in Outlet_Identifier
for i in train['moy_drop'].unique():
  # Assign the mean value corresponding to unique entry
  train['Trip_month_drop_mean'][train['moy_drop'] == i] = train['trip_duration'][train['moy_drop'] == i].mean()

for i in test['moy_drop'].unique():
  # Assign the mean value corresponding to unique entry
  test['Trip_month_drop_mean'][test['moy_drop'] == i] = test['trip_duration'][test['moy_drop'] == i].mean()
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3961951233.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Trip_month_drop_mean'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3961951233.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Trip_month_drop_mean'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3961951233.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Trip_month_drop_mean'][train['moy_drop'] == i] = train['trip_duration'][train['moy_drop'] == i].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3961951233.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Trip_month_drop_mean'][train['moy_drop'] == i] = train['trip_duration'][train['moy_drop'] == i].mean()
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3961951233.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Trip_month_drop_mean'][test['moy_drop'] == i] = test['trip_duration'][test['moy_drop'] == i].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3961951233.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Trip_month_drop_mean'][test['moy_drop'] == i] = test['trip_duration'][test['moy_drop'] == i].mean()
```

In [43]:

```
#calculating mean absolute error
train_dom_error = MAE(train['trip_duration'] , train['Trip_month_drop_mean'] )
train_dom_error
```

Out[43]:

```
582.9433928358521
```

In [44]:

```
#calculating mean absolute error
test_dom_error = MAE(test['trip_duration'] , test['Trip_month_drop_mean'] )
test_dom_error
```

Out[44]:

```
566.5147948874875
```

Mean trip duration with respect to day of the week -drop

In [45]:

```
Trip_dow_drop = pd.pivot_table(train, values='trip_duration', index = ['dow_drop'], aggfunc=np.mean)
Trip_dow_drop

# initializing new column to zero
train['Trip_dow_drop_mean'] = 0
test['Trip_dow_drop_mean'] = 0

# For every unique entry in Outlet_Identifier
for i in train['dow_drop'].unique():
  # Assign the mean value corresponding to unique entry
  train['Trip_dow_drop_mean'][train['dow_drop'] == i] = train['trip_duration'][train['dow_drop'] == i].mean()


for i in test['dow_drop'].unique():
  # Assign the mean value corresponding to unique entry
  test['Trip_dow_drop_mean'][test['dow_drop'] == i] = test['trip_duration'][test['dow_drop'] == i].mean()
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2296294940.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Trip_dow_drop_mean'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2296294940.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Trip_dow_drop_mean'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2296294940.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Trip_dow_drop_mean'][train['dow_drop'] == i] = train['trip_duration'][train['dow_drop'] == i].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2296294940.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Trip_dow_drop_mean'][train['dow_drop'] == i] = train['trip_duration'][train['dow_drop'] == i].mean()
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2296294940.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Trip_dow_drop_mean'][test['dow_drop'] == i] = test['trip_duration'][test['dow_drop'] == i].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/2296294940.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Trip_dow_drop_mean'][test['dow_drop'] == i] = test['trip_duration'][test['dow_drop'] == i].mean()
```

In [46]:
```
#calculating mean absolute error
train_dow_error2 = MAE(train['trip_duration'] , train['Trip_dow_drop_mean'] )
train_dow_error2
```

Out[46]: 582.4710676885143

In [47]:
```
#calculating mean absolute error
test_dow_error2 = MAE(test['trip_duration'] , test['Trip_dow_drop_mean'] )
test_dow_error2
```

Out[47]: 565.7320091081157

In [48]:
```
Trip_dow_drop
```

Out[48]:

dow_drop	trip_duration
0	903.280962
1	897.420137
2	958.669669
3	933.672868
4	933.116908
5	866.358782
6	881.024046

Mean trip duration with respect to both month of year pick and drop

In [49]:
```
combo3 = pd.pivot_table(train, values = 'trip_duration', index = ['moy_pick','moy_drop'], aggfunc = np.mean)
combo3
```

Out[49]:

moy_pick	moy_drop	trip_duration
1	1	887.265796
	2	28588.533333
2	2	851.662766
	3	24439.666667
3	3	888.331738
	4	10892.428571
4	4	917.325177
	5	10956.069767
5	5	946.102682
	6	11102.687500
6	6	951.128562
	7	9996.900000

In [50]:
```
# Initiating new empty column
train['Super_mean3'] = 0
test['Super_mean3'] = 0

# Assigning variables to strings ( to shorten code length)
s2 = 'moy_pick'
s1 = 'moy_drop'

# For every Unique Value in s1
for i in train[s1].unique():
  # For every Unique Value in s2
  for j in train[s2].unique():
    # Calculate and Assign mean to new column, corresponding to both unique values of s1 and s2 simultaneously
    train['Super_mean3'][(train[s1] == i) & (train[s2]==str(j))] = train['trip_duration'][(train[s1] == i) & (train[s2]==str(j))].mean()


# For every Unique Value in s1
for i in test[s1].unique():
  # For every Unique Value in s2
  for j in test[s2].unique():
    # Calculate and Assign mean to new column, corresponding to both unique values of s1 and s2 simultaneously
    test['Super_mean3'][(test[s1] == i) & (test[s2]==str(j))] = test['trip_duration'][(test[s1] == i) & (test[s2]==str(j))].mean()
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3119986238.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Super_mean3'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3119986238.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Super_mean3'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3119986238.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Super_mean3'][(train[s1] == i) & (train[s2]==str(j))] = train['trip_duration'][(train[s1] == i) & (train[s2]==str(j))].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3119986238.py:22: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Super_mean3'][(test[s1] == i) & (test[s2]==str(j))] = test['trip_duration'][(test[s1] == i) & (test[s2]==str(j))].mean()
```

In [51]:
```
#calculating mean absolute error
train_smean3_error = MAE(train['trip_duration'] , train['Super_mean3'] )
train_smean3_error
```

Out[51]:
```
911.0425272415664
```

In [52]:
```
#calculating mean absolute error
test_smean3_error = MAE(test['trip_duration'] , test['Super_mean3'] )
test_smean3_error
```

Out[52]:
```
900.715975161932
```

Mean trip duration with respect to both vendors and day of week -drop

In [53]:
```
combo4 = pd.pivot_table(train, values = 'trip_duration', index = ['moy_pick','dow_pick'], aggfunc = np.mean)
combo4
```

Out[53]:

		trip_duration
moy_pick	**dow_pick**	
1	**0**	837.341783
	1	1085.736575
	2	854.120155
	3	877.132434
	4	905.715195
	5	877.117457
	6	813.200017
2	**0**	845.120140
	1	867.978668
	2	850.235674
	3	884.497101
	4	886.758887
	5	843.207743
	6	814.930504
3	**0**	839.021106
	1	888.227921
	2	897.553445
	3	951.164336
	4	961.919057
	5	847.371224
	6	833.097357
4	**0**	840.797440
	1	930.571239
	2	942.378691
	3	981.675223
	4	961.700926
	5	904.906535
	6	874.147401
5	**0**	856.903733
	1	963.195140
	2	990.546884
	3	1061.791107
	4	987.113586
	5	904.947750
	6	881.177396
6	**0**	918.172631
	1	916.620877
	2	1022.514737
	3	995.217011
	4	1007.154110
	5	928.214875
	6	853.529350

In [54]:
```
# Initiating new empty column
train['Super_mean4'] = 0
test['Super_mean4'] = 0

# Assigning variables to strings ( to shorten code length)
s2 = 'moy_pick'
s1 = 'dow_pick'

# For every Unique Value in s1
for i in train[s1].unique():
  # For every Unique Value in s2
  for j in train[s2].unique():
    # Calculate and Assign mean to new column, corresponding to both unique values of s1 and s2 simultaneously
    train['Super_mean4'][(train[s1] == i) & (train[s2]==str(j))] = train['trip_duration'][(train[s1] == i) & (train[s2]==str(j))].mean()


for i in test[s1].unique():
  # For every Unique Value in s2
  for j in test[s2].unique():
    # Calculate and Assign mean to new column, corresponding to both unique values of s1 and s2 simultaneously
    test['Super_mean4'][(test[s1] == i) & (test[s2]==str(j))] = test['trip_duration'][(test[s1] == i) & (test[s2]==str(j))].mean()
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/1093559382.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Super_mean4'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/1093559382.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Super_mean4'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/1093559382.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Super_mean4'][(train[s1] == i) & (train[s2]==str(j))] = train['trip_duration'][(train[s1] == i) & (train[s2]==str(j))].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/1093559382.py:21: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Super_mean4'][(test[s1] == i) & (test[s2]==str(j))] = test['trip_duration'][(test[s1] == i) & (test[s2]==str(j))].mean()
```

In [55]:
```
#calculating mean absolute error
train_smean_error4 = MAE(train['trip_duration'] , train['Super_mean4'] )
train_smean_error4
```

Out[55]:
```
911.0425272415664
```

In [56]:
```
#calculating mean absolute error
test_smean_error4 = MAE(test['trip_duration'] , test['Super_mean4'] )
test_smean_error4
```

Out[56]:
```
900.715975161932
```

Mean trip duration with respect to vendors, day of week, month of year - pick

In [57]:
```
combo_mega = pd.pivot_table(train, values = 'trip_duration', index = ['vendor_id','moy_pick','dow_pick'], aggfunc = np.mean)
combo_mega
```

Out[57]:

			trip_duration
vendor_id	**moy_pick**	**dow_pick**	
1	**1**	**0**	734.556954
		1	1172.413275
		2	769.508708
		3	782.635528
		4	756.438118
...	**...**	**...**	...
2	**6**	**2**	1118.584774
		3	1095.508307
		4	1128.006176
		5	1063.287740
		6	943.424473

84 rows × 1 columns

In [58]:
```
# Initiating new empty column
train['Super_mean5'] = 0
test['Super_mean5'] = 0


# Assigning variables to strings ( to shorten code length)
s1 = 'vendor_id'
s2 = 'dow_pick'
s3 = 'moy_pick'

# For every Unique Value in s1
for i in train[s1].unique():
  # For every Unique Value in s2
  for j in train[s2].unique():
        # For every Unique Value in s3
        for k in train[s3].unique():
            # Calculate and Assign mean to new column, corresponding to both unique values of s1 and s2 simultaneously
            train['Super_mean5'][(train[s1] == i) & (train[s2]==str(j)) & (train[s3] == k)] = train['trip_duration'][(train[s1] == i) & (train[s2]==str(j)) & (train[s3] == k)].mean()


# For every Unique Value in s1
for i in test[s1].unique():
  # For every Unique Value in s2
  for j in test[s2].unique():
        # For every Unique Value in s3
        for k in test[s3].unique():
            # Calculate and Assign mean to new column, corresponding to both unique values of s1 and s2 simultaneously
            test['Super_mean5'][(test[s1] == i) & (test[s2]==str(j)) & (test[s3] == k)] = test['trip_duration'][(test[s1] == i) & (test[s2]==str(j)) & (test[s3] == k)].mean()
```

```
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3034671049.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Super_mean5'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3034671049.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Super_mean5'] = 0
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3034671049.py:18: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train['Super_mean5'][(train[s1] == i) & (train[s2]==str(j)) & (train[s3] == k)] = train['trip_duration'][(train[s1] == i) & (train[s2]==str(j)) & (train[s3] == k)].mean()
C:\Users\vempa\anaconda3\lib\site-packages\pandas\core\generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
C:\Users\vempa\AppData\Local\Temp/ipykernel_6676/3034671049.py:28: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test['Super_mean5'][(test[s1] == i) & (test[s2]==str(j)) & (test[s3] == k)] = test['trip_duration'][(test[s1] == i) & (test[s2]==str(j)) & (test[s3] == k)].mean()
```

In [59]:
```
#calculating mean absolute error
train_smean_error5 = MAE(train['trip_duration'] , train['Super_mean5'] )
train_smean_error5
```

Out[59]:
```
911.0425272415664
```

In [60]:
```
#calculating mean absolute error
test_smean_error5 = MAE(test['trip_duration'] , test['Super_mean5'] )
test_smean_error5
```

Out[60]:
```
900.715975161932
```

In [62]:
```
##### After various combinations the Least possible train and test errors were:
test_error_list = [simple_test_mean_error, test_vtype_error, test_month_error, test_dow_error, test_smean_error,
test_smean_error2,test_dom_error,test_dow_error2,test_smean3_error,test_smean_error4,test_smean_error5]

train_error_list = [simple_train_mean_error,train_vtype_error,train_month_error,train_dow_error,train_smean_error,
train_smean_error2,train_dom_error,train_dow_error2,train_smean3_error,train_smean_error4,train_smean_error5]

print("Min Train error =", min(train_error_list))
print("Corresponding Min Test error =",min(test_error_list))
```

```
Min Train error = 581.9665110196446
Corresponding Min Test error = 565.4238077668508
```

In [ ]:

In [ ]:

In [ ]: