Benchmark Model

In [35]:

```
#importing libraries

%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime as dt
import warnings
warnings.filterwarnings("ignore")
```

In [36]:

```
#Reafing data
data = pd.read_csv('nyc_taxi_trip_duration.csv')

# converting strings to datetime features
data['pickup_datetime'] = pd.to_datetime(data.pickup_datetime)
data['dropoff_datetime'] = pd.to_datetime(data.dropoff_datetime)

#Making two new columns
data['day_of_week'] = data['pickup_datetime'].dt.weekday
data['hour_of_day'] = data['pickup_datetime'].dt.hour

#Removing outliers
data=data[data["trip_duration"]<2000]
data = data.loc[(data.pickup_latitude > 40.6) & (data.pickup_latitude < 40.9)]
data = data.loc[(data.dropoff_latitude>40.6) & (data.dropoff_latitude < 40.9)]
data = data.loc[(data.dropoff_longitude > -74.05) & (data.dropoff_longitude < -73.7)]
data = data.loc[(data.pickup_longitude > -74.05) & (data.pickup_longitude < -73.7)]
data.drop(["id","pickup_datetime","dropoff_datetime","pickup_longitude","pickup_latitude","dropoff_longitude","dropoff_latitude",
data.head()
```

Out[36]:

	vendor_id	passenger_count	trip_duration	day_of_week	hour_of_day
0	2	1	400	0	16
1	1	2	1100	4	23
2	2	2	1635	6	17
3	2	6	1141	1	9
4	1	1	848	2	6

In [14]:

```
data.shape
```

Out[14]:

```
(685677, 5)
```

In [163]:

```
data.head()
```

Out[163]:

	vendor_id	passenger_count	trip_duration	day_of_week	hour_of_day
0	2	1	400	0	16
1	1	2	1100	4	23
2	2	2	1635	6	17
3	2	6	1141	1	9
4	1	1	848	2	6

In [164]:

```
np.sum(pd.isnull(data))
```

Out[164]:

```
vendor_id          0
passenger_count    0
trip_duration      0
day_of_week        0
hour_of_day        0
dtype: int64
```

In [15]:

```
data.head()
```

Out[15]:

	vendor_id	passenger_count	trip_duration	day_of_week	hour_of_day
0	2	1	400	0	16
1	1	2	1100	4	23
2	2	2	1635	6	17
3	2	6	1141	1	9
4	1	1	848	2	6

In [37]:

```
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 [167]:

```
train.head()
```

Out[167]:

	vendor_id	passenger_count	trip_duration	day_of_week	hour_of_day
71517	2	1	1199	2	8
139536	2	2	1152	3	18
4526	2	5	423	1	20
625848	2	2	758	1	18
410258	2	2	515	5	3

In [38]:

```
test['simple_mean'] = train['trip_duration'].mean()
train['simple_mean'] = train['trip_duration'].mean()
```

In [39]:

```
from sklearn.metrics import mean_absolute_error as MAE

simple_mean_error = MAE(test['trip_duration'] , test['simple_mean'])
simple_mean_error2 = MAE(train['trip_duration'] , train['simple_mean'])
simple_mean_error
```

Out[39]:

```
351.75273416135633
```

In [40]:

```
simple_mean_error2
```

Out[40]:

```
351.26564067758295
```

In [46]:

```
#Mean trip_duration with respect to vendor_id
passenger_count_type = pd.pivot_table(train, values='trip_duration', index = ['vendor_id'], aggfunc=np.mean)
passenger_count_type
```

Out[46]:

vendor_id	trip_duration
1	718.647054
2	723.436769

In [23]:

```
# initializing new column to zero
test['vendor_type_mean'] = 0

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

Out[23]:

```
514258    723.436769
728708    718.647054
186490    723.436769
97215     723.436769
183307    718.647054
             ...
275683    723.436769
389094    723.436769
140385    723.436769
713848    718.647054
129793    718.647054
Name: vendor_type_mean, Length: 68380, dtype: float64
```

In [24]:

```
vendor_type_error = MAE(test['trip_duration'] , test['vendor_type_mean'] )
vendor_type_error
```

Out[24]:

```
351.74532034817554
```

In [25]:

```
#Mean trip_duration with respect to passenger_count
passenger_count_type = pd.pivot_table(train, values='trip_duration', index = ['passenger_count'], aggfunc=np.mean)
passenger_count_type
```

Out[25]:

passenger_count	trip_duration
0	213.576923
1	715.643458
2	739.278536
3	739.656598
4	745.854947
5	724.996737
6	721.017223
9	560.000000

In [175]:

```
# initializing new column to zero
test['passenger_count_type_mean'] = 0
# For every unique entry
for i in train['passenger_count'].unique():
  # Assign the mean value corresponding to unique entry
  test['passenger_count_type_mean'][test['passenger_count'] == int(i)] = train['trip_duration'][train['passenger_count'] == int(i
passenger_count_type_error = MAE(test['trip_duration'] , test['passenger_count_type_mean'] )
passenger_count_type_error
```

Out[175]:

```
351.6379135240336
```

In [26]:

```
#Mean trip_duration with respect to day_of_week
day_of_week_type = pd.pivot_table(train, values='trip_duration', index = ['day_of_week'], aggfunc=np.mean)
day_of_week_type
```

Out[26]:

day_of_week	trip_duration
0	695.401126
1	734.629776
2	747.448019
3	749.358113
4	739.553838
5	702.639391
6	674.321097

In [27]:

```
# initializing new column to zero
test['day_of_week_type_mean'] = 0
# For every unique entry
for i in train['day_of_week'].unique():
  # Assign the mean value corresponding to unique entry
  test['day_of_week_type_mean'][test['day_of_week'] == int(i)] = train['trip_duration'][train['day_of_week'] == int(i)].mean()
day_of_week_type_error = MAE(test['trip_duration'] , test['day_of_week_type_mean'] )
day_of_week_type_error
```

Out[27]:

```
350.7151856225268
```

In [43]:

```
#Mean trip_duration with respect to hour_of_day
hour_of_day_type = pd.pivot_table(train, values='trip_duration', index = ['hour_of_day'], aggfunc=np.mean)
hour_of_day_type
```

Out[43]:

hour_of_day	trip_duration
0	721.184614
1	693.080736
2	668.994869
3	674.059729
4	692.889246
5	647.553284
6	563.043401
7	640.550479
8	721.685790
9	736.213329
10	738.193306
11	755.721452
12	755.070818
13	755.468455
14	758.750505
15	745.719218
16	725.472184
17	732.917201
18	735.694209
19	710.497657
20	701.490276
21	712.296099
22	736.784072
23	737.016873

In [44]:

```
# initializing new column to zero
test['hour_of_day_type_mean'] = 0
# For every unique entry
for i in train['hour_of_day'].unique():
  # Assign the mean value corresponding to unique entry
  test['hour_of_day_type_mean'][test['hour_of_day'] == int(i)] = train['trip_duration'][train['hour_of_day'] == int(i)].mean()
hour_of_day_type_error = MAE(test['trip_duration'] , test['hour_of_day_type_mean'] )
hour_of_day_type_error
```

Out[44]:

```
349.949874273212
```

In [30]:

```
#Mean trip_duration with respect to hour_of_day
hour_of_day_type = pd.pivot_table(train, values='trip_duration', index = ['hour_of_day'], aggfunc=np.mean)
hour_of_day_type
```

Out[30]:

hour_of_day	trip_duration
0	721.184614
1	693.080736
2	668.994869
3	674.059729
4	692.889246
5	647.553284
6	563.043401
7	640.550479
8	721.685790
9	736.213329
10	738.193306
11	755.721452
12	755.070818
13	755.468455
14	758.750505
15	745.719218
16	725.472184
17	732.917201
18	735.694209
19	710.497657
20	701.490276
21	712.296099
22	736.784072
23	737.016873

In [45]:

```
# initializing new column to zero
test['hour_of_day_type_mean'] = 0
# For every unique entry
for i in train['hour_of_day'].unique():
  # Assign the mean value corresponding to unique entry
  test['hour_of_day_type_mean'][test['hour_of_day'] == int(i)] = train['trip_duration'][train['hour_of_day'] == int(i)].mean()
hour_of_day_type_error = MAE(test['trip_duration'] , test['hour_of_day_type_mean'] )
hour_of_day_type_error
```

Out[45]:

```
349.949874273212
```

In [ ]:

```

```