```
#Benchmark Model
%matplotlib inline
import numpy as np
import pandas as pd
from datetime import timedelta
import datetime as dt
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
from sklearn.neighbors import KNeighborsRegressor as KNN
from sklearn.metrics import mean_squared_error as mse

df = pd.read_csv('nyc_taxi_trip_duration Dataset.csv')
df['pickup_datetime'] = pd.to_datetime(df.pickup_datetime)
df['dropoff_datetime'] = pd.to_datetime(df.dropoff_datetime)

df['day_of_week'] = df['pickup_datetime'].dt.weekday
df['hour_of_day'] = df['pickup_datetime'].dt.hour

df=df[df["trip_duration"]<2000]
df = df.loc[(df.pickup_latitude > 40.6) & (df.pickup_latitude < 40.9)]
df = df.loc[(df.dropoff_latitude>40.6) & (df.dropoff_latitude < 40.9)]
df = df.loc[(df.dropoff_longitude > -74.05) & (df.dropoff_longitude <
-73.7)]
df = df.loc[(df.pickup_longitude > -74.05) & (df.pickup_longitude < -
73.7)]
df.drop(["id","pickup_datetime","dropoff_datetime","pickup_longitude",
"pickup_latitude","dropoff_longitude","dropoff_latitude","store_and_fw
d_flag"],axis=1,inplace=True)
df.head()




from sklearn.utils import shuffle

# Shuffling the Dataset
data = shuffle(df, 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:]

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

```
#calculating mean squared error
from sklearn.metrics import mean_absolute_error as MAE

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

```
351.75273416135633
```

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

```
           trip_duration
vendor_id
1              718.647054
2              723.436769
```

```
# 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']
```

```
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
```

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

```
351.74532034817554
```

```
#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
```

```
# 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)].mean()
passenger_count_type_error = MAE(test['trip_duration'] ,
test['passenger_count_type_mean'] )
passenger_count_type_error
```

```
351.6379135240336
```

```
#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
```

```
# 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
```

```
350.7151856225268
```

```
#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
```

```
# 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
```

```
349.949874273212
```