INTRODUCTION

It is a model that predicts the total ride duration of taxi trips in New York City. The primary dataset is released by the NYC Taxi and Limousine Commission, which include pickup time, geo-coordinates, number of passengers, and several other variables.

In [10]:
```
import numpy as np
import pandas as pd
```

In [11]:
```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [12]:
```
sns.set_style('whitegrid')
```

In [13]:
```
df_train = pd.read_csv(r"C:\Users\Meghna\OneDrive\Desktop\Dataset\train.csv", nrows=200000, parse_dates=["pickup_
```

Data Exploration

In [14]:
```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 11 columns):
 #   Column              Non-Null Count   Dtype         
---  ------              --------------   -----         
 0   id                  200000 non-null  object        
 1   vendor_id           200000 non-null  int64         
 2   pickup_datetime     200000 non-null  datetime64[ns]
 3   dropoff_datetime    200000 non-null  object        
 4   passenger_count     200000 non-null  int64         
 5   pickup_longitude    200000 non-null  float64       
 6   pickup_latitude     200000 non-null  float64       
 7   dropoff_longitude   200000 non-null  float64       
 8   dropoff_latitude    200000 non-null  float64       
 9   store_and_fwd_flag  200000 non-null  object        
 10  trip_duration       200000 non-null  int64         
dtypes: datetime64[ns](1), float64(4), int64(3), object(3)
memory usage: 16.8+ MB
```

In [15]:
```
df_train.describe()
```

Out[15]:

	vendor_id	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	trip_duration
count	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000
mean	1.534715	1.664310	-73.973469	40.751078	-73.973452	40.751991	947.400800
std	0.498795	1.312426	0.039509	0.038705	0.037465	0.033634	3115.290628
min	1.000000	0.000000	-77.896019	37.777771	-77.896019	37.777771	1.000000
25%	1.000000	1.000000	-73.991829	40.737412	-73.991318	40.736136	395.000000
50%	2.000000	1.000000	-73.981728	40.754200	-73.979729	40.754566	661.000000
75%	2.000000	2.000000	-73.967268	40.768410	-73.962927	40.769939	1073.000000
max	2.000000	6.000000	-72.809669	51.881084	-72.711395	43.486885	86390.000000

In [16]:
```
df_train.dtypes
```

Out[16]:
```
id                            object
vendor_id                      int64
pickup_datetime       datetime64[ns]
dropoff_datetime              object
passenger_count                int64
pickup_longitude             float64
pickup_latitude              float64
dropoff_longitude            float64
```

```
dropoff_latitude            float64
store_and_fwd_flag           object
trip_duration                 int64
dtype: object
```

NAN or Missing Values:

In [17]:
```
df_train['trip_duration'].isnull().sum()
```

Out[17]:
```
0
```

In [18]:
```
df_train.drop_duplicates(inplace=True)
df_train.shape
```

Out[18]:
```
(200000, 11)
```

In [19]:
```
df_train['passenger_count'].value_counts().reset_index()
```

Out[19]:

	index	passenger_count
0	1	141653
1	2	28873
2	5	10802
3	3	8269
4	6	6523
5	4	3877
6	0	3

In [20]:
```
df_train[df_train['passenger_count']==0].count()
```

Out[20]:
```
id                  3
vendor_id           3
pickup_datetime     3
dropoff_datetime    3
passenger_count     3
pickup_longitude    3
pickup_latitude     3
dropoff_longitude   3
dropoff_latitude    3
store_and_fwd_flag  3
trip_duration       3
dtype: int64
```

In [21]:
```
df_train=df_train[df_train['passenger_count']!=0]
```

In [22]:
```
df_train["pickup_datetime"][0]
```

Out[22]:
```
Timestamp('2016-03-14 17:24:00')
```

In [23]:
```
df_train['pickup_datetime']=pd.to_datetime(df_train['pickup_datetime'])
df_train['dropoff_datetime']=pd.to_datetime(df_train['dropoff_datetime'])
df_train.head()
```

Out[23]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitu
0	id2875421	2	2016-03-14 17:24:00	2016-03-14 17:32:00	1	-73.982155	40.767937	-73.964630	40.7656
1	id2377394	1	2016-12-06	2016-12-06	1	-73.980415	40.738564	-73.999481	40.7311

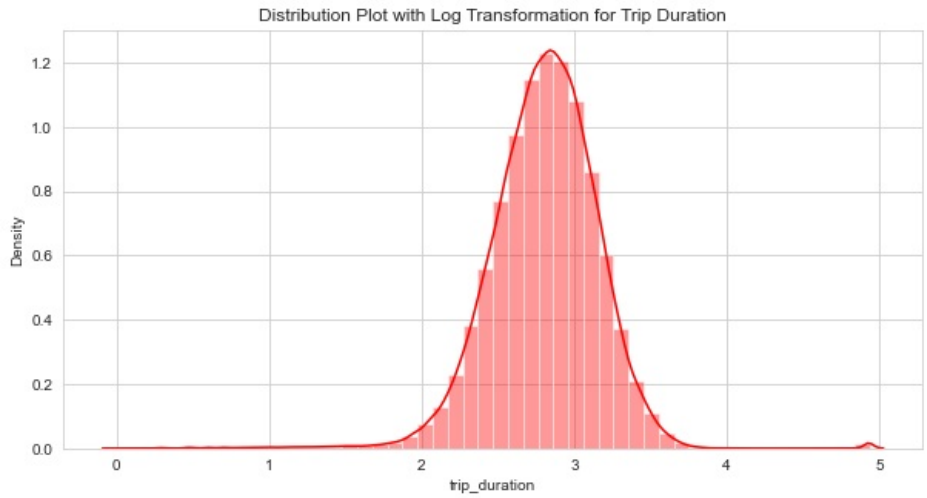
			00:43:00	00:54:00					
2	id3858529	2	2016-01-19 11:35:00	2016-01-19 12:10:00	1	-73.979027	40.763939	-74.005333	40.7100
3	id3504673	2	2016-06-04 19:32:00	2016-06-04 19:39:00	1	-74.010040	40.719971	-74.012268	40.7067
4	id2181028	2	2016-03-26 13:30:00	2016-03-26 13:38:00	1	-73.973053	40.793209	-73.972923	40.7825

Data Preprocessing

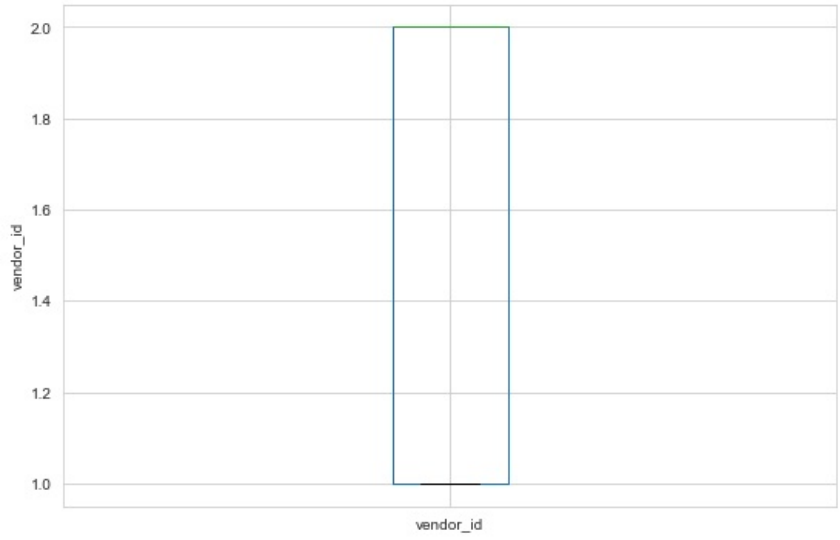
In [24]:
```
plt.figure(figsize=(10, 5))
sns.distplot(np.log10(df_train['trip_duration']), color="red").set(title='Distribution Plot with Log Transformati
```

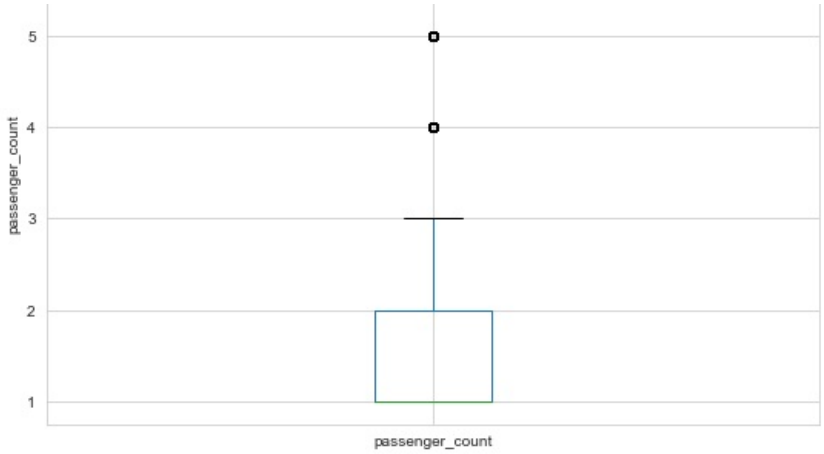
```
C:\Users\Meghna\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a depreca
ted function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-le
vel function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

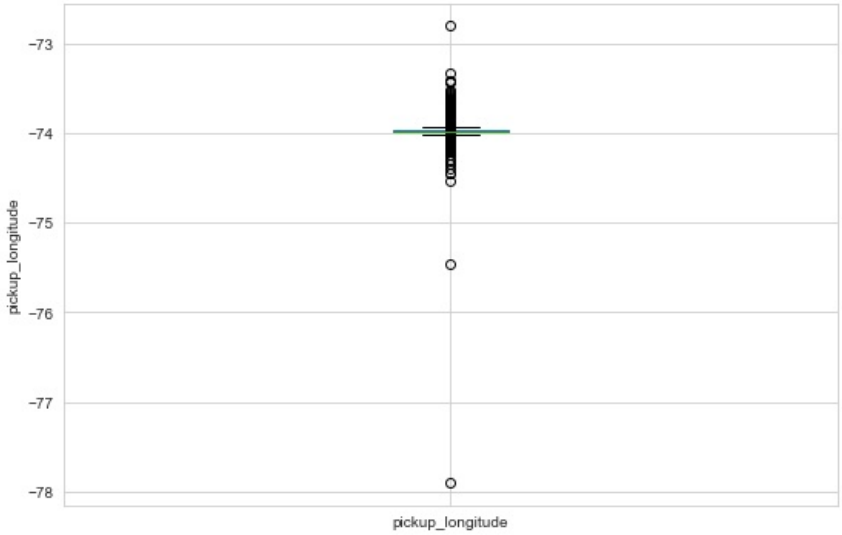
Out[24]:
```
[Text(0.5, 1.0, 'Distribution Plot with Log Transformation for Trip Duration')]
```

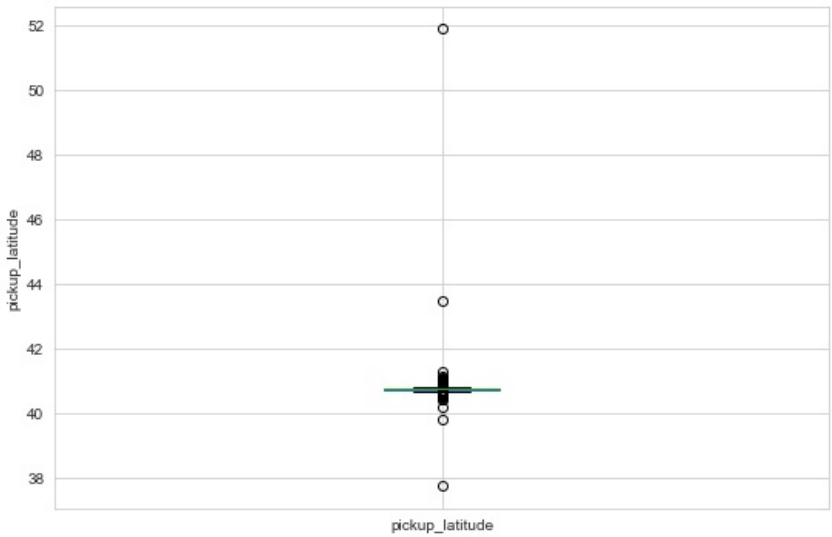


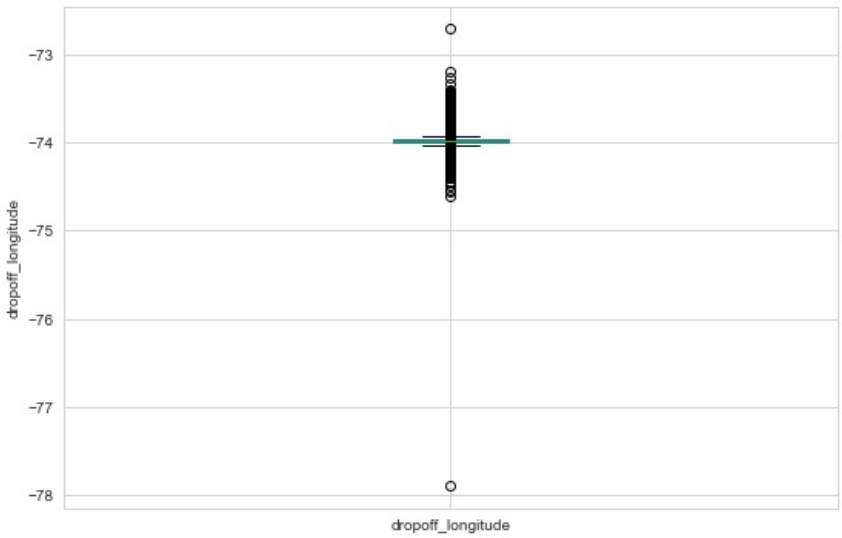
In [25]:
```
for col in df_train.describe().columns:
  fig = plt.figure(figsize=(9, 6))
  ax = fig.gca()
  df_train.boxplot(column = col, ax = ax)
  ax.set_ylabel(col)
plt.title("Box plot for trip_duration")
plt.show()
```

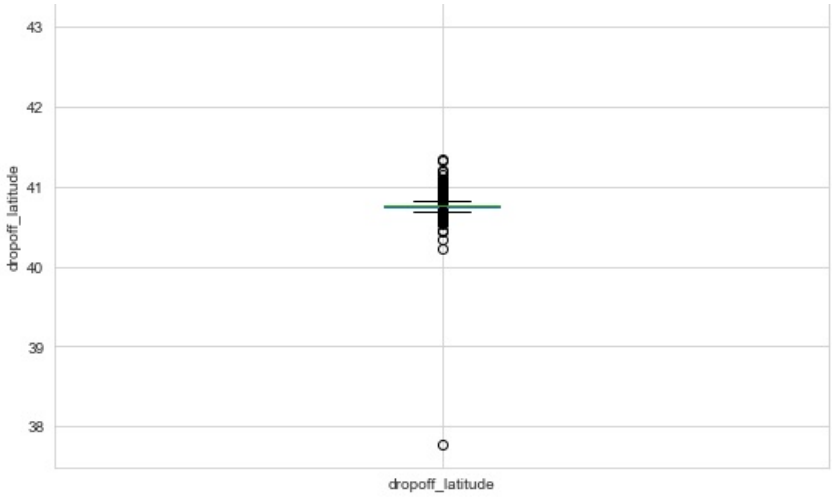


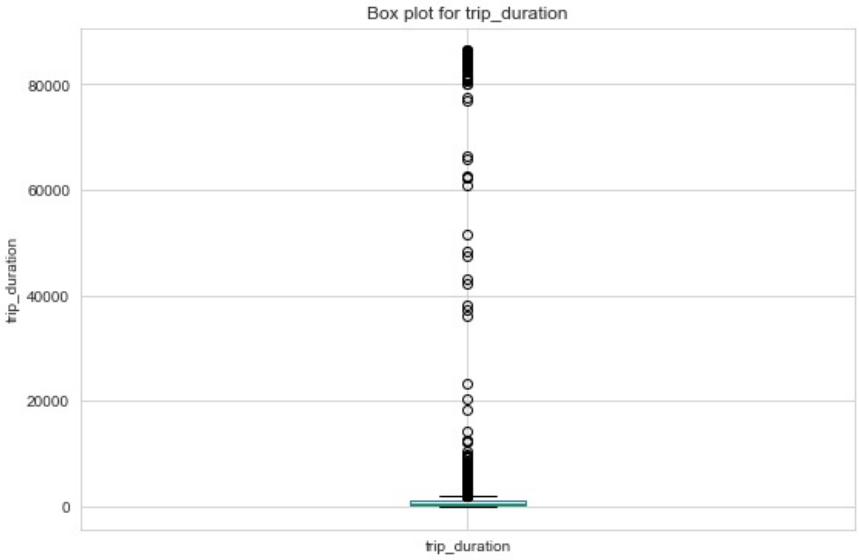

5
4
3
2
1
passenger_count
passenger_count


−73
−74
−75
−76
−77
−78
pickup_longitude
pickup_longitude


52
50
48
46
44
42
40
38
pickup_latitude
pickup_latitude


−73
−74
−75
−76
−77
−78
dropoff_longitude
dropoff_longitude



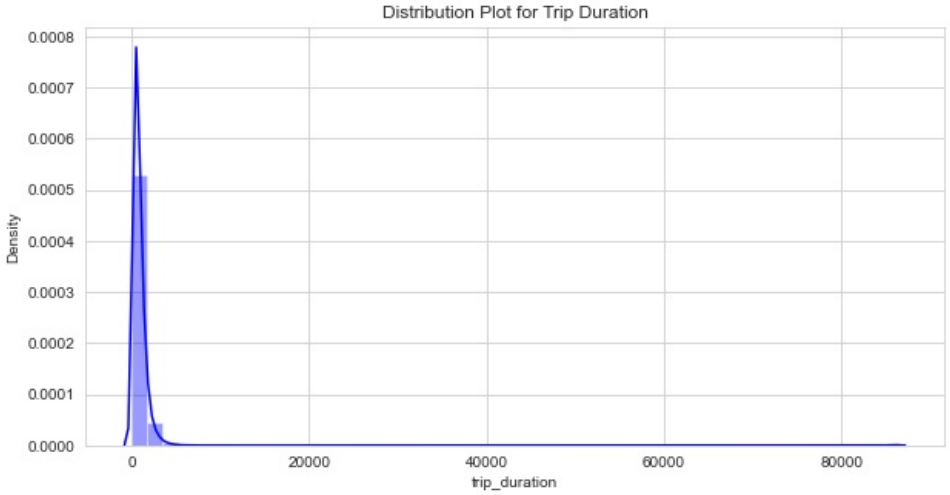


In [26]:

```
plt.figure(figsize=(10,5))
sns.distplot(df_train['trip_duration'],color="b").set(title='Distribution Plot for Trip Duration')
```

```
C:\Users\Meghna\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[26]: [Text(0.5, 1.0, 'Distribution Plot for Trip Duration')]



In [27]:

```
def calculate_trip_duration(pickup,dropoff):
```

```
    return (dropoff-pickup).total_seconds()
```

In [28]:
```
df_train['calculate_trip_duration']=df_train.apply(lambda x: calculate_trip_duration(x['pickup_datetime'],x['drop
```

In [29]:
```
(df_train['calculate_trip_duration']==df_train['trip_duration']).value_counts()
```

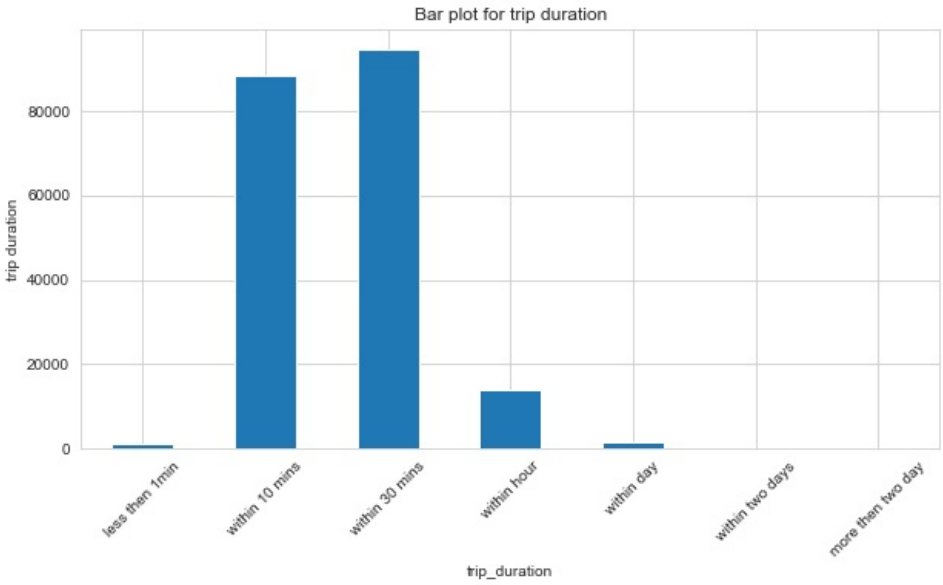
Out[29]:
```
False    196599
True       3398
dtype: int64
```

Here, we see that there the trip duration is consistent with the calculated trip duration. so, this large value are purely an outlier.

In [30]:
```
df_train.drop(['calculate_trip_duration'],axis=1,inplace=True)
```

In [31]:
```
plt.figure(figsize=[10,5])
labels=['less then 1min','within 10 mins','within 30 mins','within hour','within day','within two days','more the
df_train.groupby(pd.cut(df_train['trip_duration'],bins=[0,60,600,1800,3600,86400,86400*2,10000000],labels=labels)
plt.title("Bar plot for trip duration")
plt.ylabel("trip counts")
plt.ylabel("trip duration")
plt.xticks(rotation=45)
```

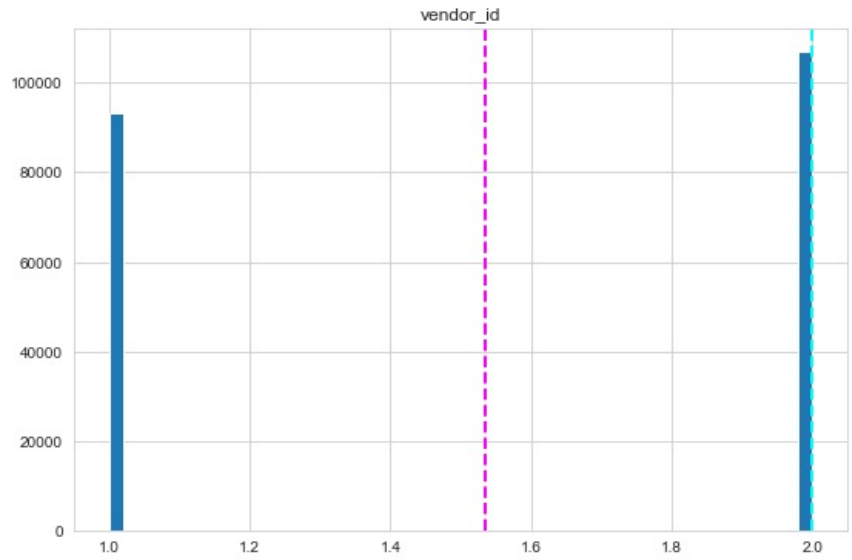
Out[31]:
```
(array([0, 1, 2, 3, 4, 5, 6]),
 [Text(0, 0, 'less then 1min'),
  Text(1, 0, 'within 10 mins'),
  Text(2, 0, 'within 30 mins'),
  Text(3, 0, 'within hour'),
  Text(4, 0, 'within day'),
  Text(5, 0, 'within two days'),
  Text(6, 0, 'more then two day')])
```

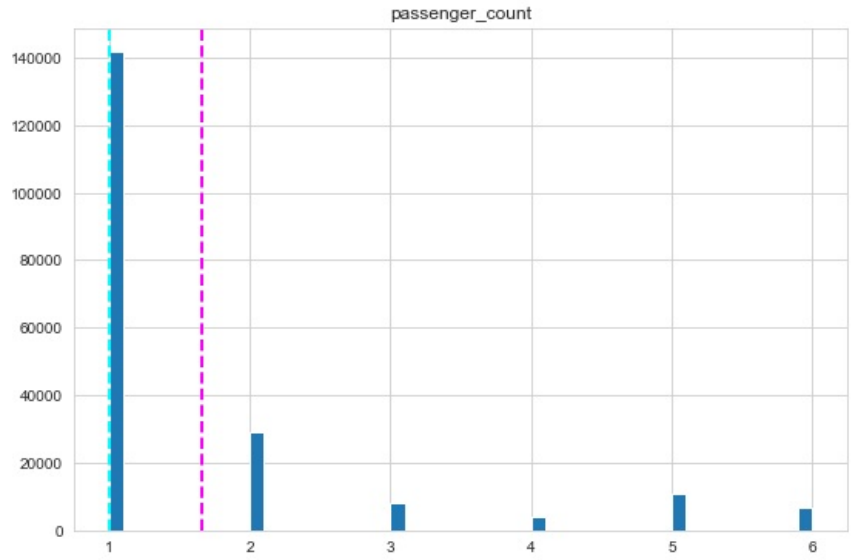


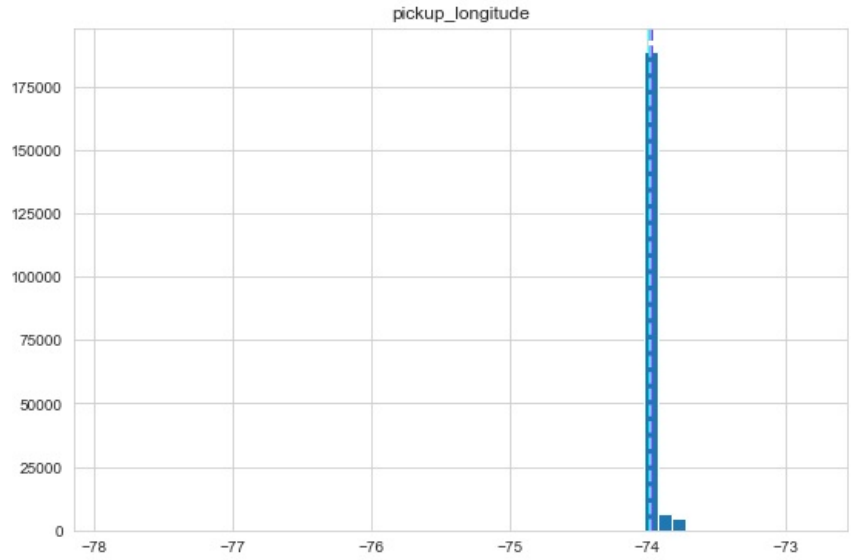
In [32]:
```
numeric_features = df_train.describe().columns
numeric_features
```

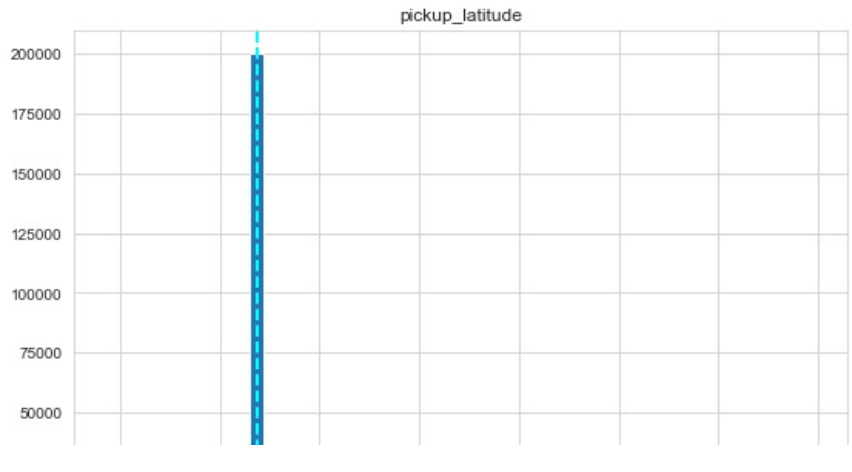
Out[32]:
```
Index(['vendor_id', 'passenger_count', 'pickup_longitude', 'pickup_latitude',
       'dropoff_longitude', 'dropoff_latitude', 'trip_duration'],
      dtype='object')
```

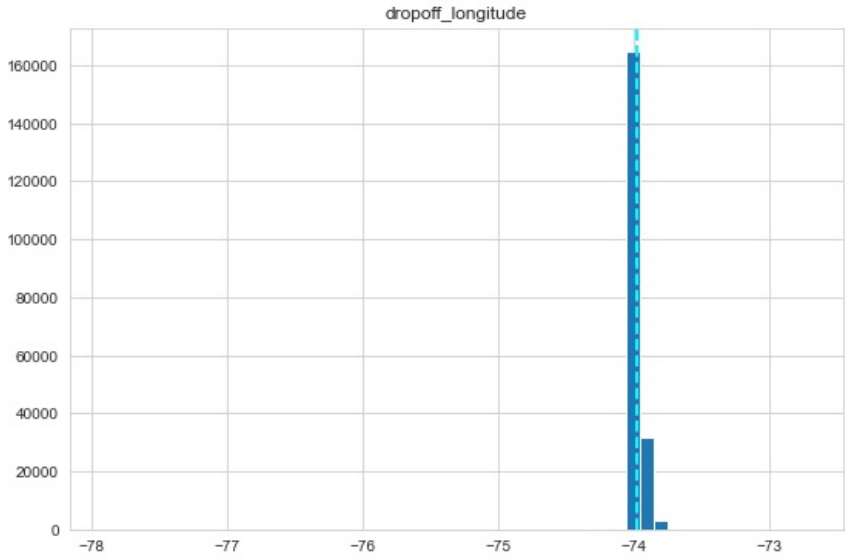
In [33]:
```
for col in numeric_features:
    fig = plt.figure(figsize=(9, 6))
    ax = fig.gca()
    feature = df_train[col]
    feature.hist(bins=50, ax = ax)
    ax.axvline(feature.mean(), color='magenta', linestyle='dashed', linewidth=2)
    ax.axvline(feature.median(), color='cyan', linestyle='dashed', linewidth=2)
    ax.set_title(col)
plt.show()
```

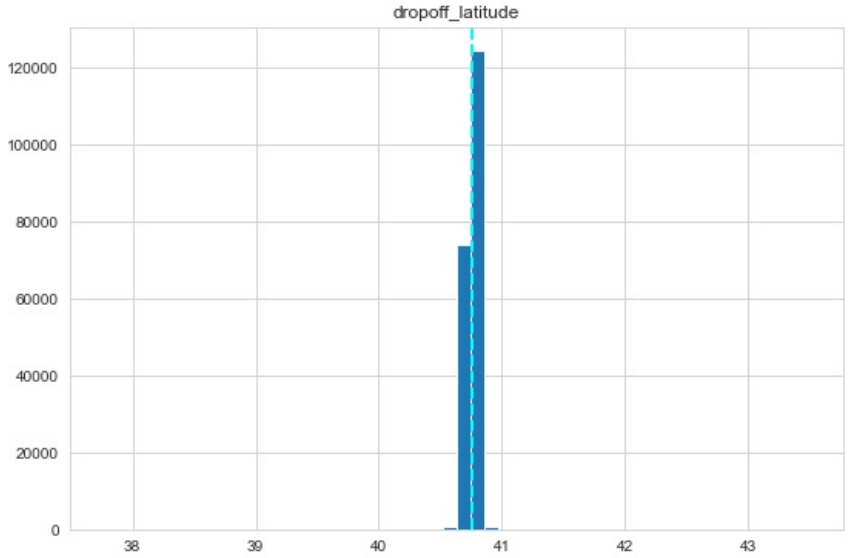

vendor_id
100000
80000
60000
40000
20000
0
1.0
1.2
1.4
1.6
1.8
2.0

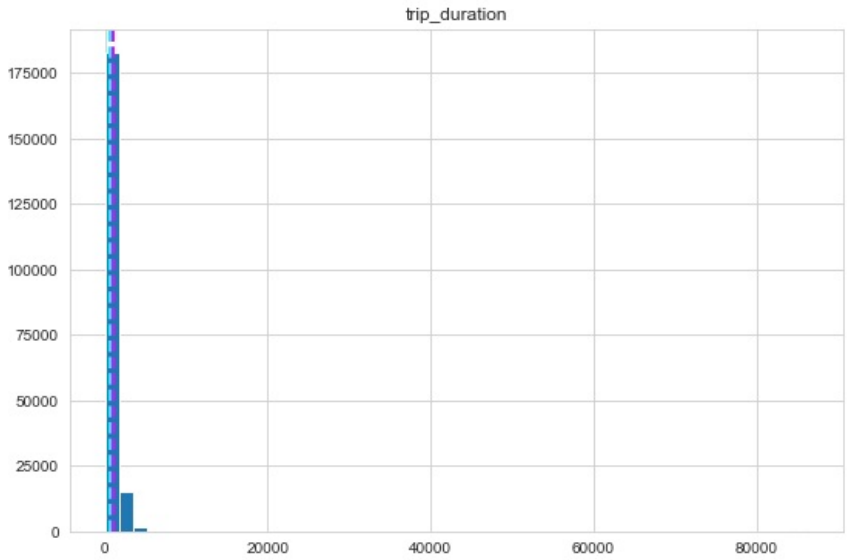

passenger_count
140000
120000
100000
80000
60000
40000
20000
0
1
2
3
4
5
6


pickup_longitude
175000
150000
125000
100000
75000
50000
25000
0
−78
−77
−76
−75
−74
−73


pickup_latitude
200000
175000
150000
125000
100000
75000
50000







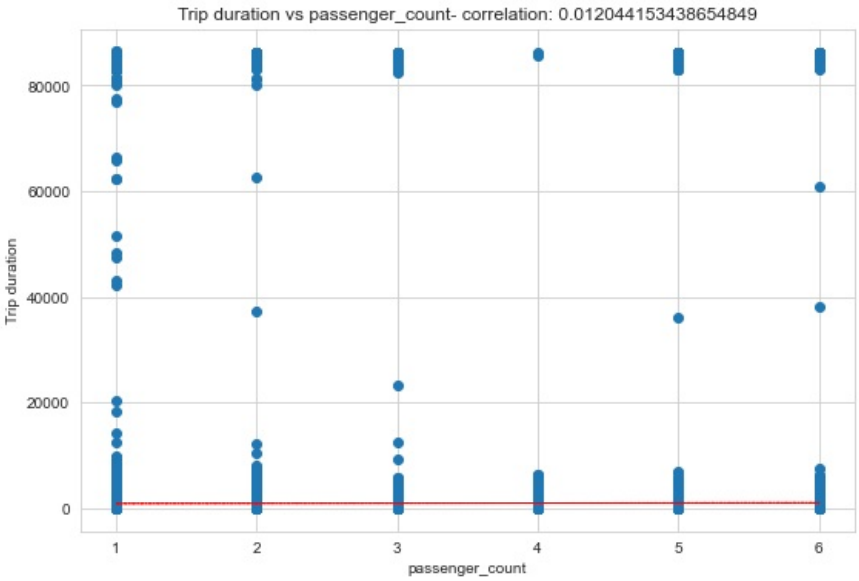
In [34]:
```
df_train.describe().columns
```

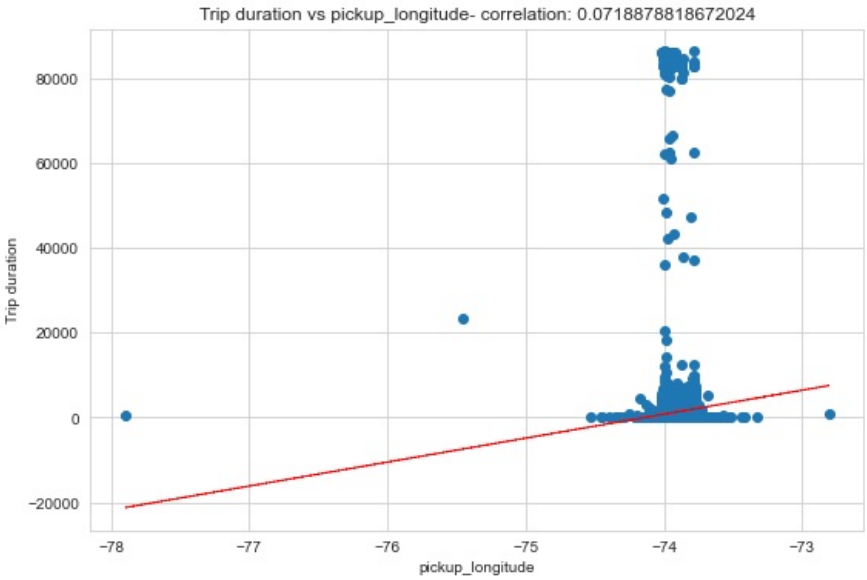
Out[34]:
```
Index(['vendor_id', 'passenger_count', 'pickup_longitude', 'pickup_latitude',
       'dropoff_longitude', 'dropoff_latitude', 'trip_duration'],
```

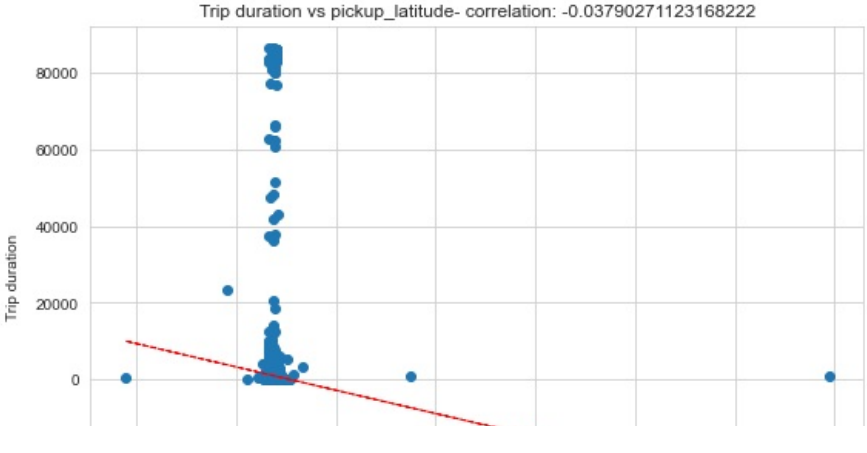
```
      dtype='object')
```

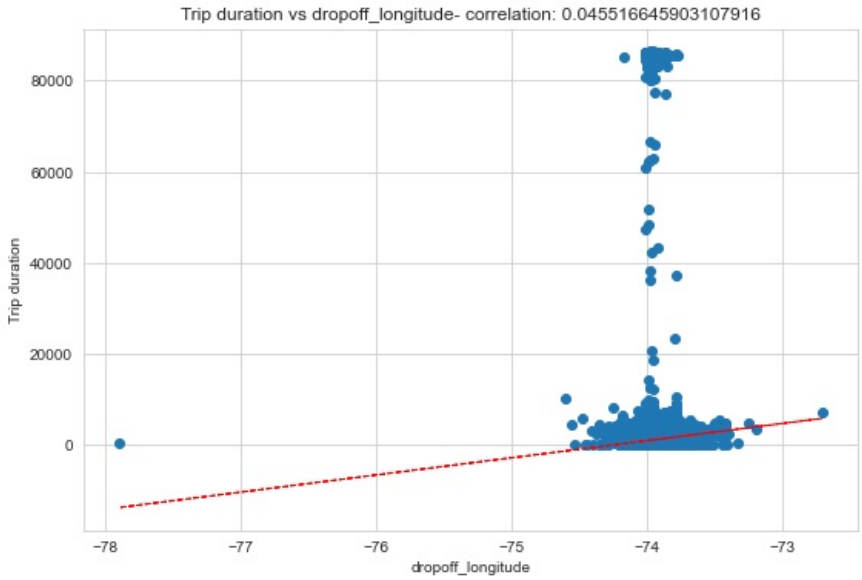
In [35]:

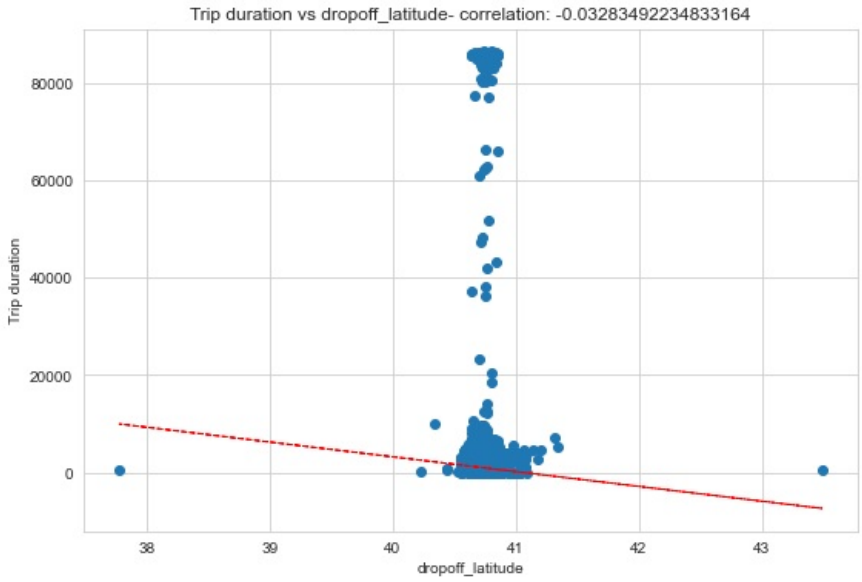
```
for col in numeric_features[1:-1]:
    fig = plt.figure(figsize=(9, 6))
    ax = fig.gca()
    feature = df_train[col]
    label = df_train['trip_duration']
    correlation = feature.corr(label)
    plt.scatter(x=feature, y=label)
    plt.xlabel(col)
    plt.ylabel('Trip duration')
    ax.set_title('Trip duration vs ' + col + '- correlation: ' + str(correlation))
    z = np.polyfit(df_train[col], df_train['trip_duration'], 1)
    y_hat = np.poly1d(z)(df_train[col])

    plt.plot(df_train[col], y_hat, "r--", lw=1)

plt.show()
```









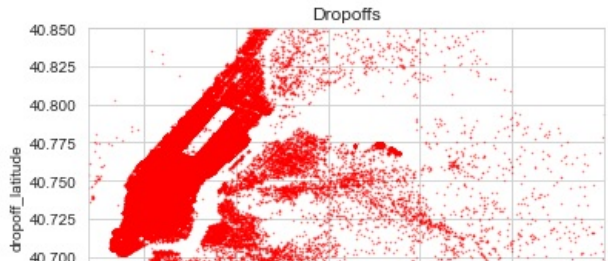


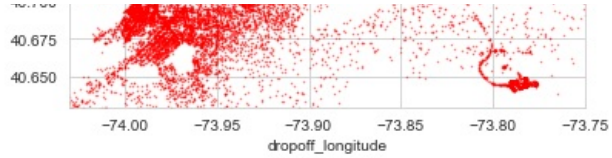
Data Visualization

In [36]:

```
city_long_border = [-74.03, -73.75]
city_lat_border = [40.63,40.85]

df_train.plot(kind='scatter', x='dropoff_longitude',y='dropoff_latitude',
          color='Red',
          s=0.2, alpha =.6)
plt.title('Dropoffs')

plt.ylim(city_lat_border)
plt.xlim(city_long_border)
```

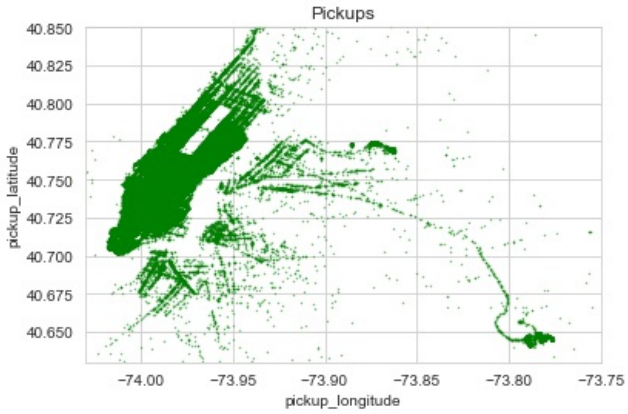
Out[36]: (-74.03, -73.75)





In [37]:
```
city_long_border = [-74.03, -73.75]
city_lat_border = [40.63,40.85]

df_train.plot(kind='scatter', x='pickup_longitude',y='pickup_latitude',
          color='green',
          s=0.2, alpha =.6)
plt.title('Pickups')

plt.ylim(city_lat_border)
plt.xlim(city_long_border)
```

Out[37]: (-74.03, -73.75)



In [38]:
```
def select_within_boundingbox(df, BB):
    return ((df_train["pickup_longitude"] >= BB[0]) & (df_train["pickup_longitude"] <= BB[1]) &
            (df_train["pickup_latitude"] >= BB[2]) & (df_train["pickup_latitude"] <= BB[3]) &
            (df_train["dropoff_longitude"] >= BB[0]) & (df_train["dropoff_longitude"] <= BB[1]) &
            (df_train["dropoff_latitude"] >= BB[2]) & (df_train["dropoff_latitude"] <= BB[3]))
BB = (-74.3, -73.0, 40.6, 41.7)
```

In [39]:
```
!pip install folium
```

```
Requirement already satisfied: folium in c:\users\meghna\anaconda3\lib\site-packages (0.14.0)
Requirement already satisfied: jinja2>=2.9 in c:\users\meghna\anaconda3\lib\site-packages (from folium) (2.11.3)
Requirement already satisfied: requests in c:\users\meghna\anaconda3\lib\site-packages (from folium) (2.25.1)
Requirement already satisfied: numpy in c:\users\meghna\anaconda3\lib\site-packages (from folium) (1.20.1)
Requirement already satisfied: branca>=0.6.0 in c:\users\meghna\anaconda3\lib\site-packages (from folium) (0.6.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\meghna\anaconda3\lib\site-packages (from jinja2>=2.9-
>folium) (1.1.1)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\meghna\anaconda3\lib\site-packages (from requests->f
olium) (4.0.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\meghna\anaconda3\lib\site-packages (from requests->
folium) (2020.12.5)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\meghna\anaconda3\lib\site-packages (from request
s->folium) (1.26.4)
Requirement already satisfied: idna<3,>=2.5 in c:\users\meghna\anaconda3\lib\site-packages (from requests->folium
) (2.10)
```

In [40]:
```
import folium
```

In [41]:
```
nyc = folium.Map(location=[40.730610,-73.935242],zoom_start=12, control_scale=True)
nyc
```

Out[41]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [42]:
```
for i in df_train.index[:100]:
  folium.Marker(location=[df_train['pickup_latitude'][i],df_train['pickup_longitude'][i]],icon=folium.Icon(color=
nyc
```

Out[42]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [43]:
```
for i in df_train.index[:100]:
  folium.Marker(location=[df_train['dropoff_latitude'][i],df_train['dropoff_longitude'][i]],icon=folium.Icon(colo
nyc
```

Out[43]: Make this Notebook Trusted to load map: File -> Trust Notebook

Data Modelling

In [44]:

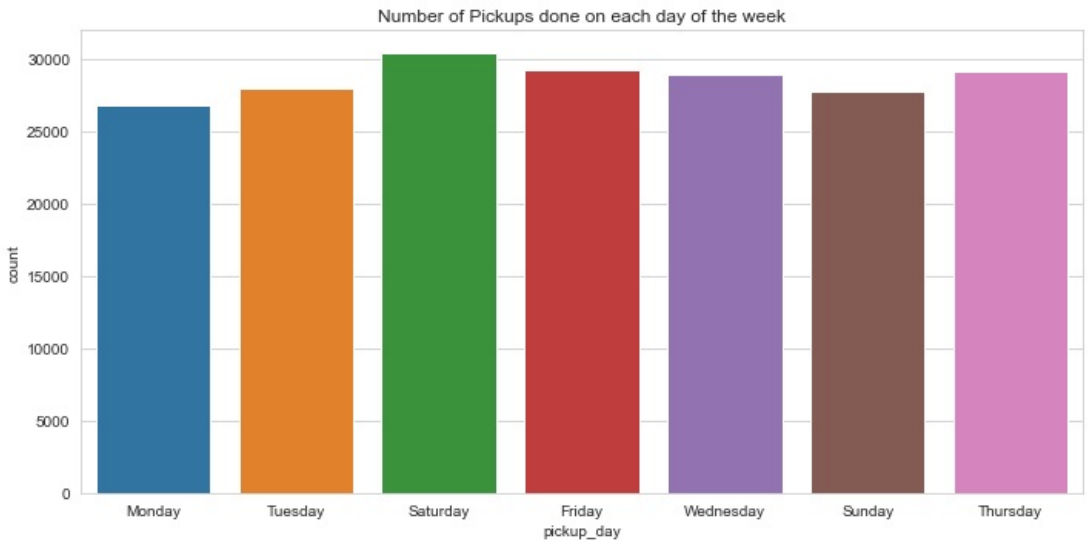
```
df_train['pickup_day']=df_train['pickup_datetime'].dt.day_name()
df_train['dropoff_day']=df_train['dropoff_datetime'].dt.day_name()
df_train.head()
```

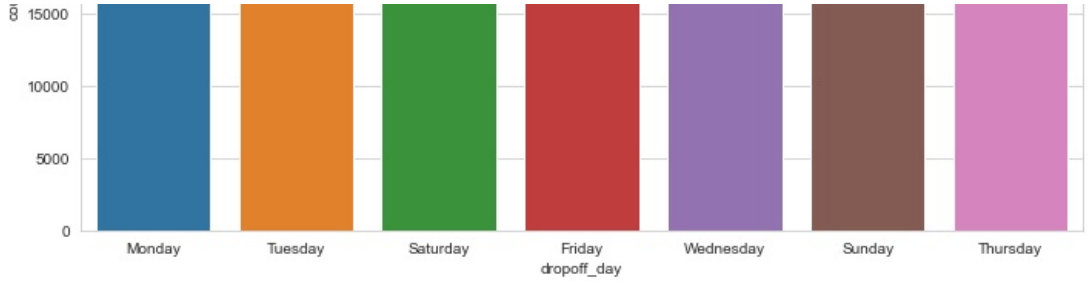
Out[44]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitu
0	id2875421	2	2016-03-14 17:24:00	2016-03-14 17:32:00	1	-73.982155	40.767937	-73.964630	40.7656
1	id2377394	1	2016-12-06 00:43:00	2016-12-06 00:54:00	1	-73.980415	40.738564	-73.999481	40.7311
2	id3858529	2	2016-01-19 11:35:00	2016-01-19 12:10:00	1	-73.979027	40.763939	-74.005333	40.7100
3	id3504673	2	2016-06-04 19:32:00	2016-06-04 19:39:00	1	-74.010040	40.719971	-74.012268	40.7067
4	id2181028	2	2016-03-26 13:30:00	2016-03-26 13:38:00	1	-73.973053	40.793209	-73.972923	40.7825

In [45]:

```
figure,ax=plt.subplots(nrows=2,ncols=1,figsize=(10,10))
sns.countplot(x='pickup_day',data=df_train,ax=ax[0])
ax[0].set_title('Number of Pickups done on each day of the week')
sns.countplot(x='dropoff_day',data=df_train,ax=ax[1])
ax[1].set_title('Number of dropoffs done on each day of the week')

plt.tight_layout()
```



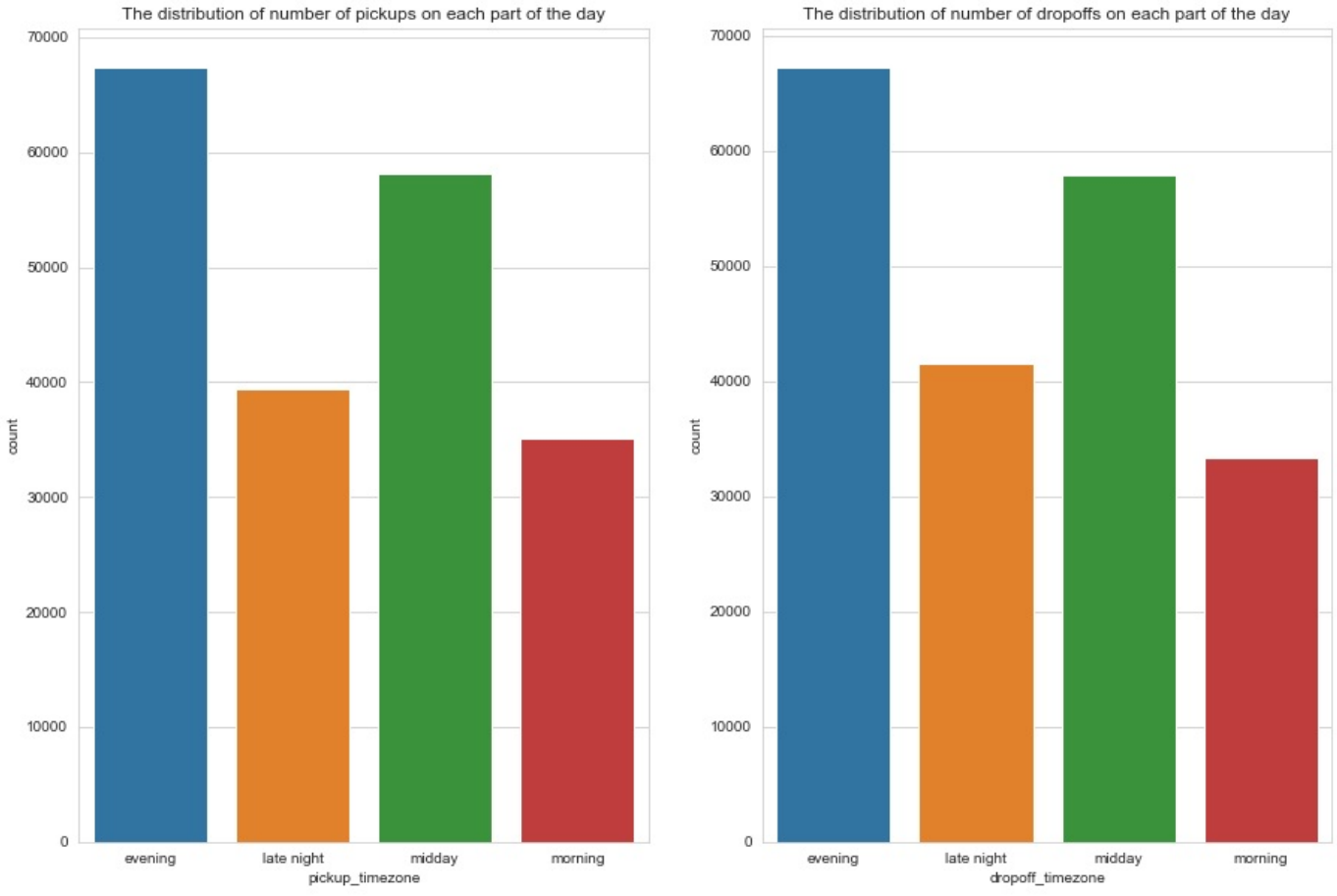


Thus we see most trips were taken on Friday and Monday being the least. The distribution of trip duration with the days of the week is something to look into as well.

In [46]:
```
bins=np.array([0,1800,3600,5400,7200,90000])
df_train['duration_time']=pd.cut(df_train.trip_duration,bins,labels=["< 5", "5-10", "10-15","15-20",">20"])
```

In [47]:
```
import datetime
def timezone(x):
    if x>=datetime.time(4, 0, 1) and x <=datetime.time(10, 0, 0):
        return 'morning'
    elif x>=datetime.time(10, 0, 1) and x <=datetime.time(16, 0, 0):
        return 'midday'
    elif x>=datetime.time(16, 0, 1) and x <=datetime.time(22, 0, 0):
        return 'evening'
    elif x>=datetime.time(22, 0, 1) or x <=datetime.time(4, 0, 0):
        return 'late night'

df_train['pickup_timezone']=df_train['pickup_datetime'].apply(lambda x :timezone(datetime.datetime.strptime(str(x
df_train['dropoff_timezone']=df_train['dropoff_datetime'].apply(lambda x :timezone(datetime.datetime.strptime(str
```

In [48]:
```
figure,ax=plt.subplots(nrows=1,ncols=2,figsize=(15,10))
sns.countplot(x='pickup_timezone',data=df_train,ax=ax[0])
ax[0].set_title('The distribution of number of pickups on each part of the day')
sns.countplot(x='dropoff_timezone',data=df_train,ax=ax[1])
ax[1].set_title('The distribution of number of dropoffs on each part of the day')
plt.show()
```



In [49]:
```
def calc_distance(df):
    pickup = (df['pickup_latitude'], df['pickup_longitude'])
```

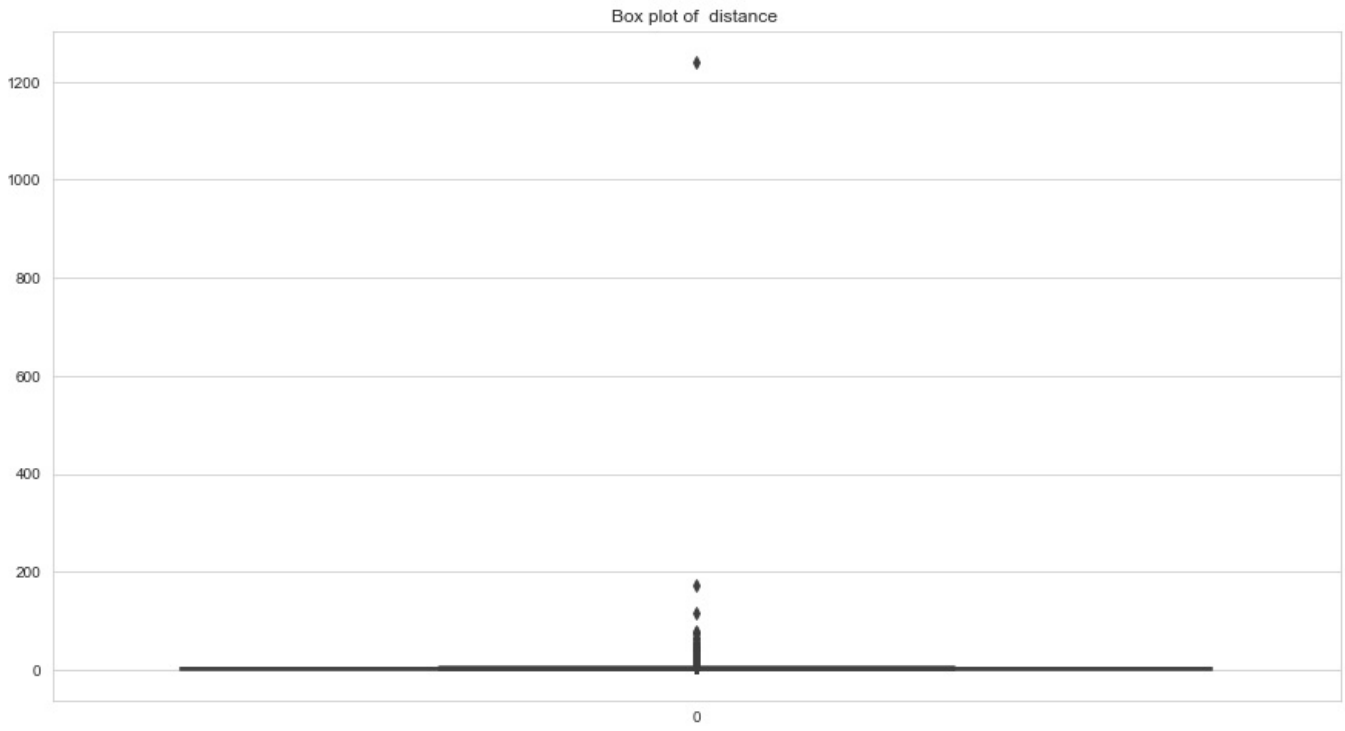
```
    drop = (df['dropoff_latitude'], df['dropoff_longitude'])
    return haversine(pickup, drop)
```

In [50]:
```
from math import radians, sin, cos, sqrt, atan2

def haversine(coord1, coord2):
    # Radius of the Earth in km
    R = 6371.0

    lat1, lon1 = radians(coord1[0]), radians(coord1[1])
    lat2, lon2 = radians(coord2[0]), radians(coord2[1])

    dlat = lat2 - lat1
    dlon = lon2 - lon1

    a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))

    distance = R * c
    return distance
```

In [51]:
```
df_train["distance"] = df_train.apply(lambda x: calc_distance(x),axis=1)
```

Outlier detection using IQR Method

In [52]:
```
plt.figure(figsize=(15,8))
plt.title("Box plot of  distance ")
ax = sns.boxplot(data=df_train['distance'], orient="v")
```



In [53]:
```
percentile_q1 = np.percentile(df_train['distance'],25)
print(percentile_q1)
percentile_q2 = np.percentile(df_train['distance'],50)
print(percentile_q2)
percentile_q3 = np.percentile(df_train['distance'],75)
print(percentile_q3)
```

```
1.2342146012794524
2.0926745318964755
3.8688271065441455
```

In [54]:
```
iqr=percentile_q3 - percentile_q1
lower_limit_outlier=percentile_q1-1.5*iqr
upper_limit_outlier=percentile_q3+1.5*iqr

print("lower limit for outlier  :",lower_limit_outlier)
```

```
print("Upper limit for outlier  :",upper_limit_outlier)
```

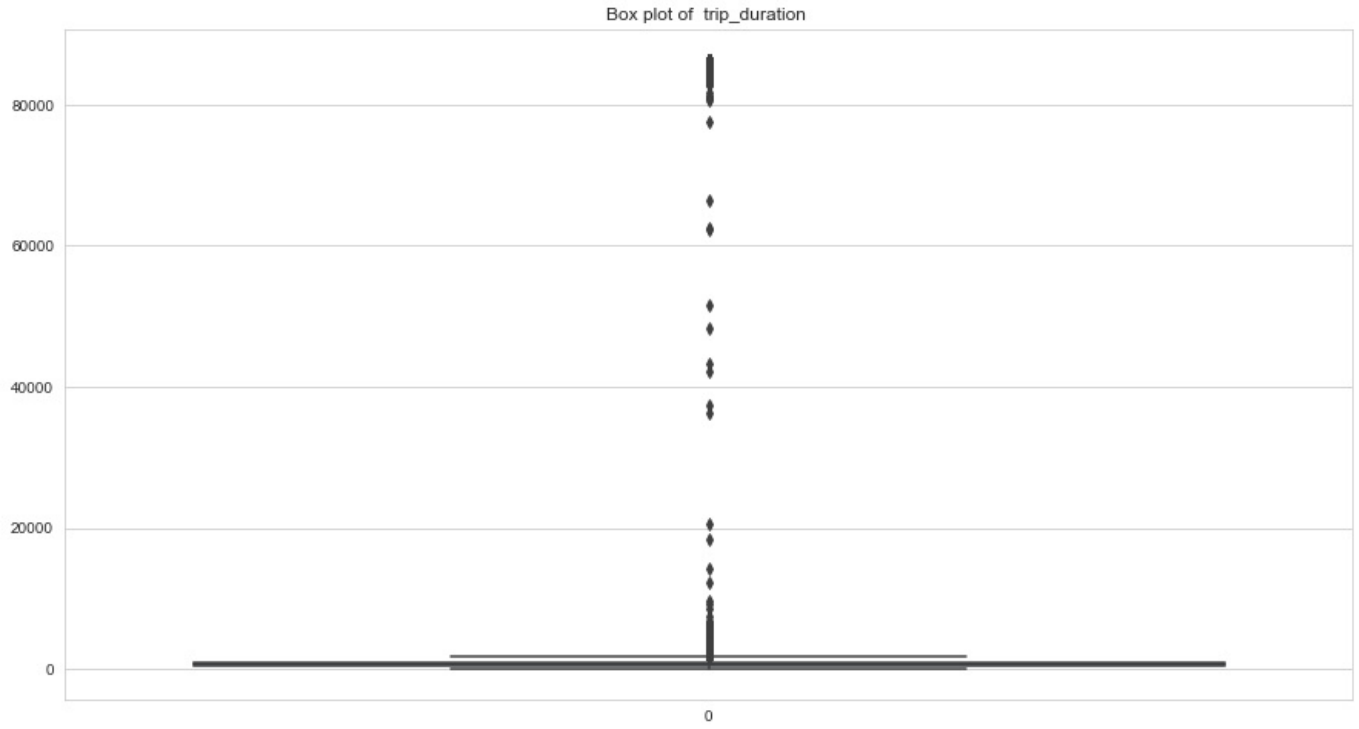
```
lower limit for outlier  : -2.717704156617587
Upper limit for outlier  : 7.820745864441185
```

In [55]:
```
df_train=df_train[df_train['distance']>lower_limit_outlier]
df_train=df_train[df_train['distance']<upper_limit_outlier]
```

In [56]:
```
df_train.shape
```

Out[56]:
```
(180677, 17)
```

In [57]:
```
plt.figure(figsize=(15,8))
plt.title("Box plot of  trip_duration ")
ax = sns.boxplot(data=df_train['trip_duration'], orient="v")
```



In [58]:
```
import numpy as np
```

In [59]:
```
percentile_q1_trip_duration = np.percentile(df_train['trip_duration'],25)
print(percentile_q1_trip_duration)
percentile_q2_trip_duration = np.percentile(df_train['trip_duration'],50)
print(percentile_q2_trip_duration)
percentile_q3_trip_duration = np.percentile(df_train['trip_duration'],75)
print(percentile_q3_trip_duration)
```

```
371.0
604.0
931.0
```

In [60]:
```
iqr=percentile_q3_trip_duration - percentile_q1_trip_duration
lower_limit_outlier_trip_duration=percentile_q1_trip_duration-1.5*iqr
upper_limit_outlier_trip_duration=percentile_q3_trip_duration+1.5*iqr

print("lower limit for outlier  :",lower_limit_outlier_trip_duration)
print("Upper limit for outlier  :",upper_limit_outlier_trip_duration)
```

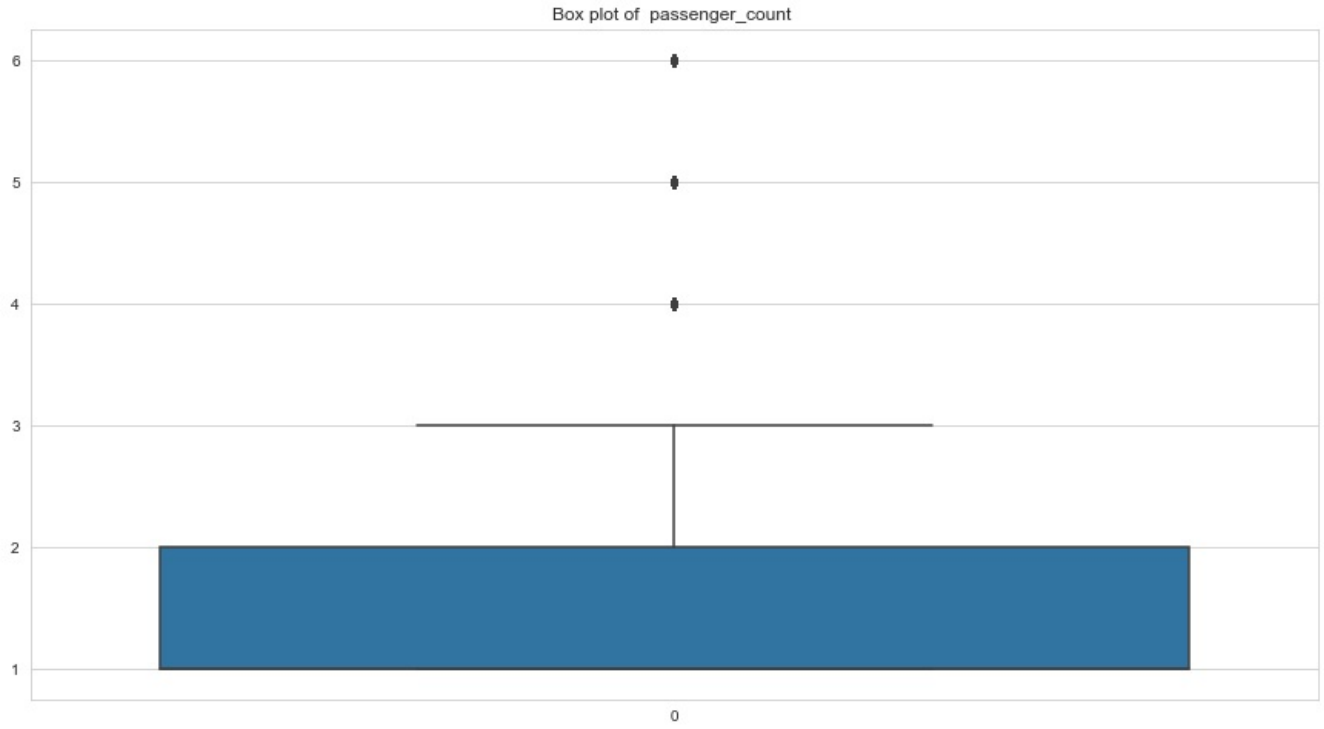
```
lower limit for outlier  : -469.0
Upper limit for outlier  : 1771.0
```

In [61]:
```
df_train=df_train[df_train['trip_duration']>0]
df_train=df_train[df_train['trip_duration']<upper_limit_outlier_trip_duration]
```

In [62]:
```
df_train.shape
```

Out[62]: (175168, 17)

In [63]:
```
plt.figure(figsize=(15,8))
plt.title("Box plot of  passenger_count ")
ax = sns.boxplot(data=df_train['passenger_count'])
```



In [64]:
```
percentile_q1_passenger_count = np.percentile(df_train['passenger_count'],25)
print(percentile_q1_passenger_count)
percentile_q2_passenger_count = np.percentile(df_train['passenger_count'],50)
print(percentile_q2_passenger_count)
percentile_q3_passenger_count = np.percentile(df_train['passenger_count'],75)
print(percentile_q3_passenger_count)
```

```
1.0
1.0
2.0
```

In [65]:
```
iqr=percentile_q3_passenger_count - percentile_q1_passenger_count
lower_limit_outlier_passenger_count=percentile_q1_passenger_count-1.5*iqr
upper_limit_outlier_passenger_count=percentile_q3_passenger_count+1.5*iqr

print("lower limit for outlier  :",lower_limit_outlier_passenger_count)
print("Upper limit for outlier  :",upper_limit_outlier_passenger_count)
```

```
lower limit for outlier  : -0.5
Upper limit for outlier  : 3.5
```

In [66]:
```
df_train=df_train[df_train['passenger_count']>0]
df_train=df_train[df_train['passenger_count']<upper_limit_outlier_passenger_count]
```

In [67]:
```
df_train.shape
```

Out[67]: (156652, 17)

In [68]:
```
df_train["pickup_datetime"] = pd.to_datetime(df_train["pickup_datetime"], format="%Y-%m-%d %H:%M:%S")
df_train['Day']=df_train['pickup_datetime'].dt.day_name()
```

In [69]:
```
df_train["year"] = df_train["pickup_datetime"].apply(lambda x: x.year)
df_train["month"] = df_train["pickup_datetime"].apply(lambda x: x.month)
df_train["day_num"] = df_train["pickup_datetime"].apply(lambda x: x.day)
df_train["hour"] = df_train["pickup_datetime"].apply(lambda x: x.hour)
df_train["minute"] = df_train["pickup_datetime"].apply(lambda x: x.minute)
```

In [70]:
```
df_train.head()
```

Out[70]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitu
0	id2875421	2	2016-03-14 17:24:00	2016-03-14 17:32:00	1	-73.982155	40.767937	-73.964630	40.7656
1	id2377394	1	2016-12-06 00:43:00	2016-12-06 00:54:00	1	-73.980415	40.738564	-73.999481	40.7311
3	id3504673	2	2016-06-04 19:32:00	2016-06-04 19:39:00	1	-74.010040	40.719971	-74.012268	40.7067
4	id2181028	2	2016-03-26 13:30:00	2016-03-26 13:38:00	1	-73.973053	40.793209	-73.972923	40.7825
7	id1324603	2	2016-05-21 07:54:00	2016-05-21 08:20:00	1	-73.969276	40.797779	-73.922470	40.7605

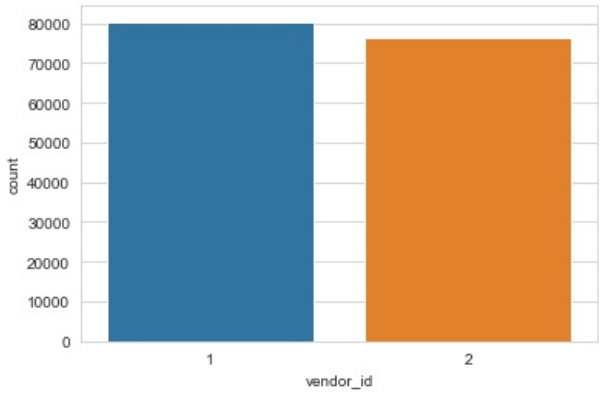
5 rows × 23 columns

In [71]:
```
df_train['trip_duration_hour']=df_train['trip_duration']/3600
df_train['log_distance']=np.log(df_train.distance)
df_train['log_trip_duration']=np.log(df_train.trip_duration_hour)
```

```
C:\Users\Meghna\anaconda3\lib\site-packages\pandas\core\arraylike.py:358: RuntimeWarning: divide by zero encounte
red in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

In [72]:
```
sns.countplot(x='vendor_id',data=df_train)
```

Out[72]:
```
<AxesSubplot:xlabel='vendor_id', ylabel='count'>
```

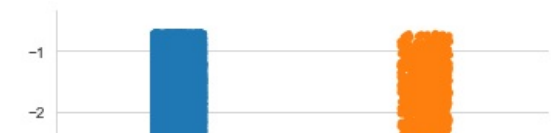


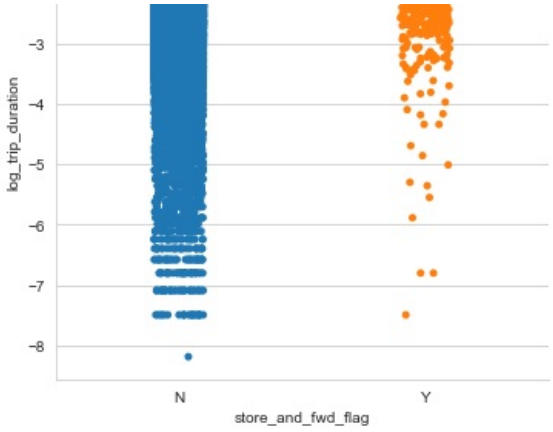
Store and fwd flag

This flag indicates whether the trip record was held in vehicle memory before seding to the vendor because the vehicle did not have a connection to the server - Y store and forward; N=not a store and forward trip.

In [73]:
```
sns.catplot(x="store_and_fwd_flag", y="log_trip_duration",kind="strip",data=df_train)
```

Out[73]:
```
<seaborn.axisgrid.FacetGrid at 0x20eee059160>
```



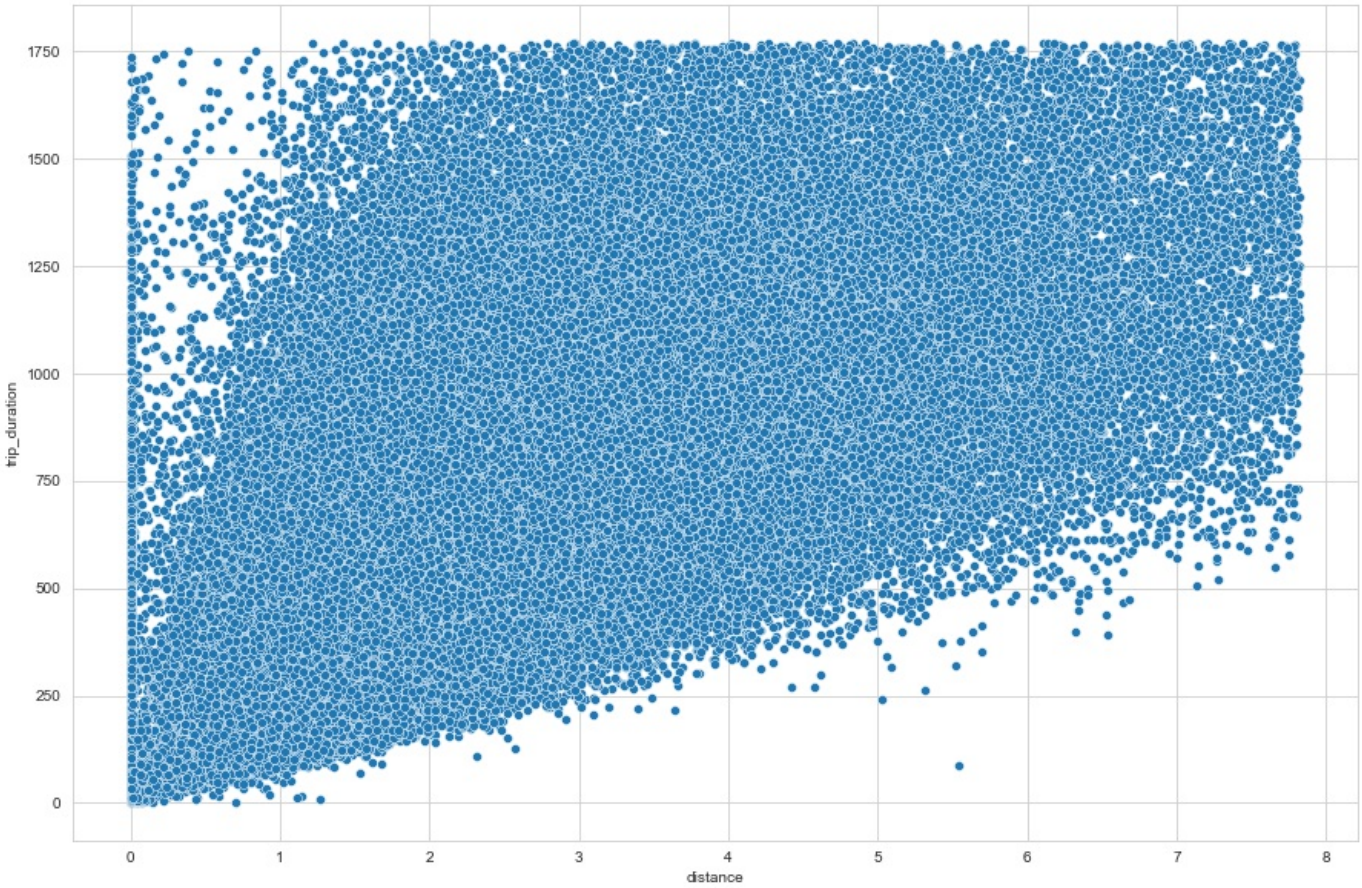


Bivariate Analysis

It is used to find out if there is a relationship between teo sets of values. It usually involves the variables X and Y.

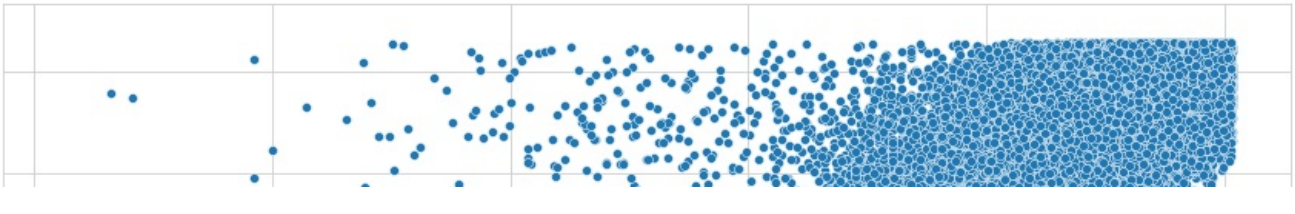
In [74]:
```
fig = plt.figure(figsize=(15, 10))
sns.scatterplot(x='distance',y='trip_duration',data=df_train)
```

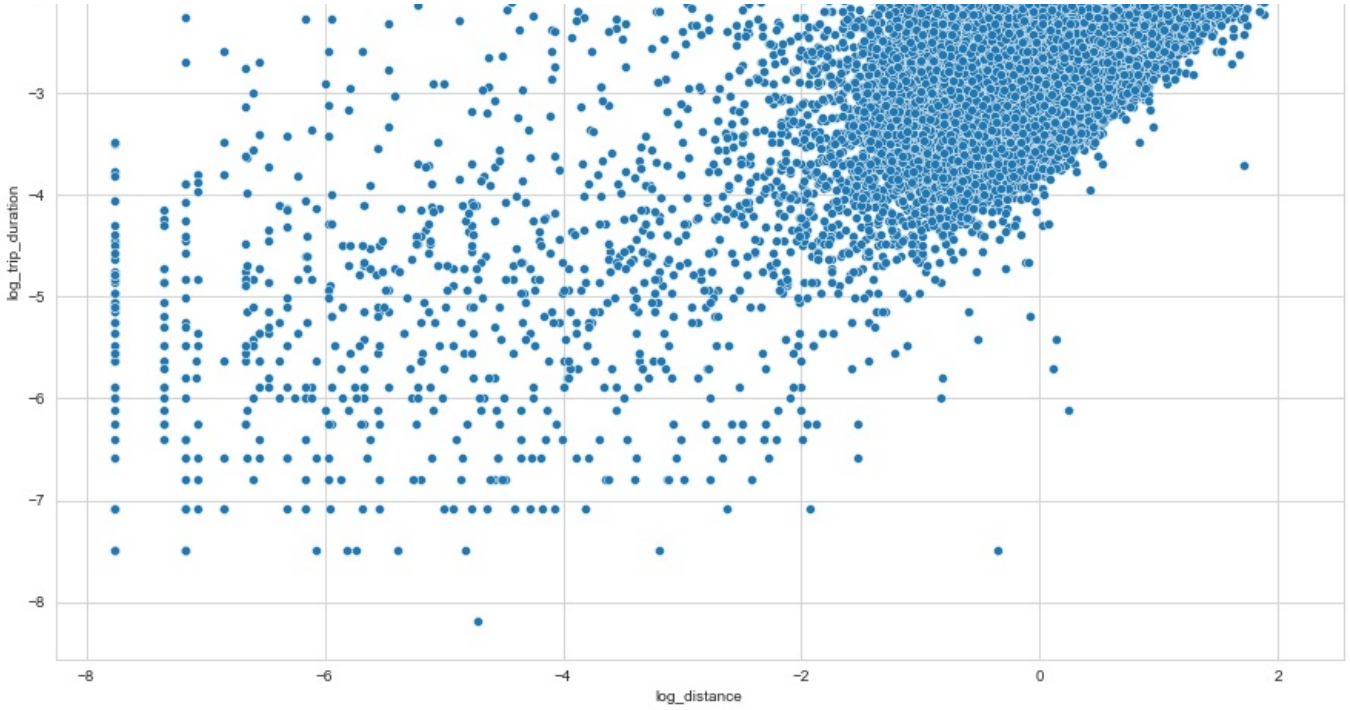
Out[74]: <AxesSubplot:xlabel='distance', ylabel='trip_duration'>



In [75]:
```
fig = plt.figure(figsize=(15, 10))
sns.scatterplot(x='log_distance',y='log_trip_duration',data=df_train)
```

Out[75]: <AxesSubplot:xlabel='log_distance', ylabel='log_trip_duration'>

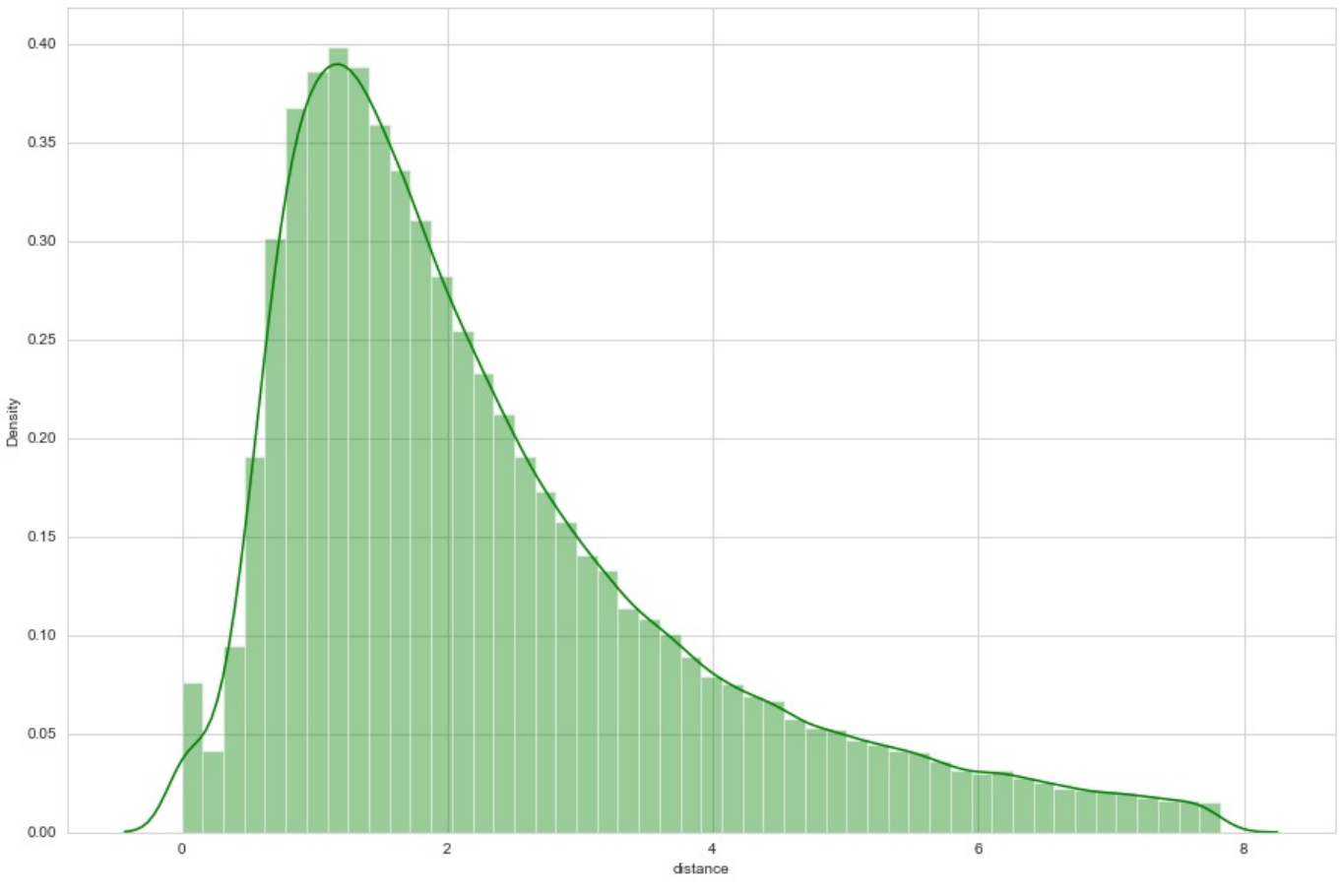


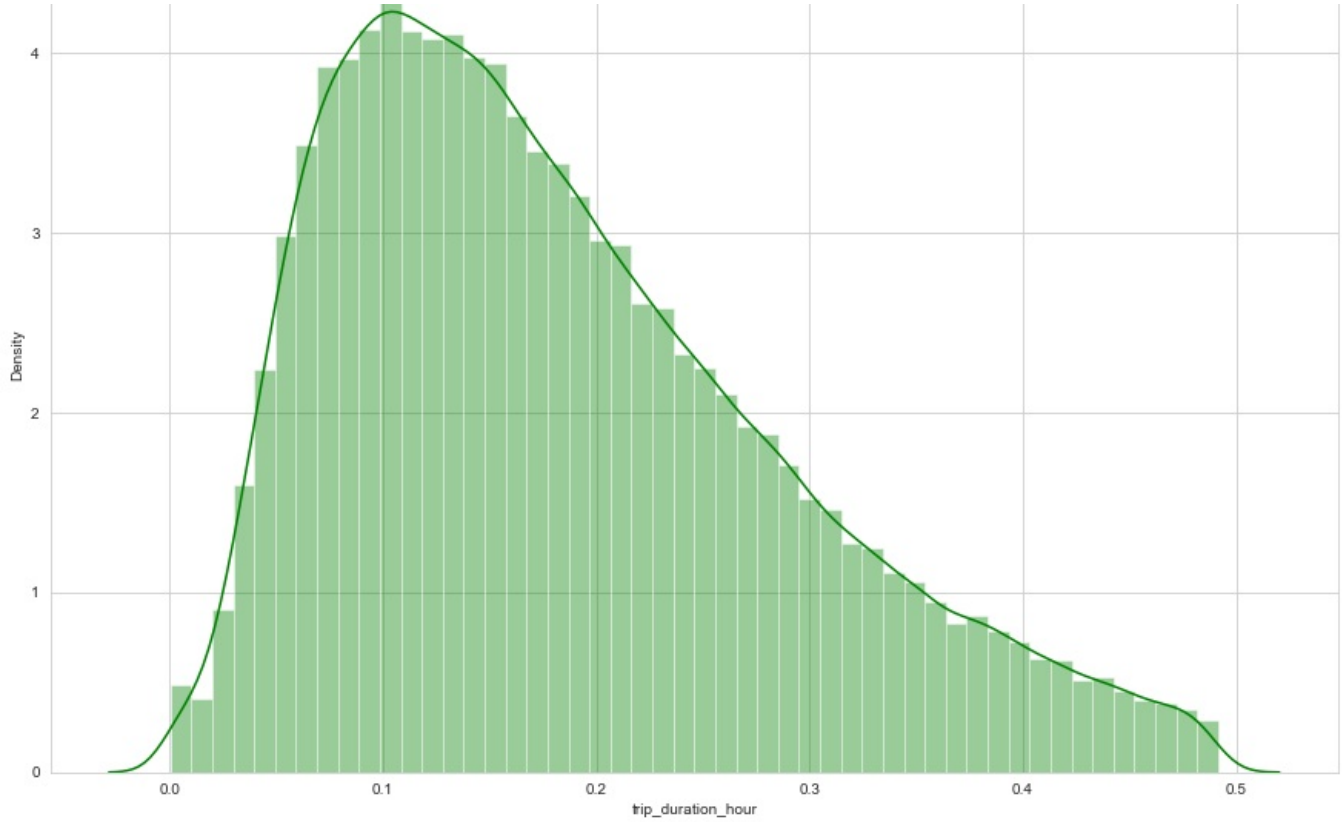


In [76]:

```
sample=['distance','trip_duration_hour']
for i in sample:
  plt.figure(figsize=(15,10))
  sns.distplot(df_train[i],color="g")
```

```
C:\Users\Meghna\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Meghna\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```





In [77]:
```
df_train= pd.get_dummies(df_train, columns=["store_and_fwd_flag", "Day"], prefix=["store_and_fwd_flag",'Day'])
#Feature for the Machine learning models
features=['vendor_id','passenger_count','pickup_longitude','pickup_latitude','dropoff_longitude','dropoff_latitud
          'store_and_fwd_flag_N','store_and_fwd_flag_Y','Day_Friday','Day_Monday','Day_Saturday','Day_Sunday','Da
newdata=['vendor_id','passenger_count','pickup_longitude','pickup_latitude','dropoff_longitude','dropoff_latitude
          'store_and_fwd_flag_N','store_and_fwd_flag_Y','Day_Friday','Day_Monday','Day_Saturday','Day_Sunday','Da
trip_data=df_train[newdata]
df_train.shape
```

Out[77]: (156652, 33)

In [79]:
```
from scipy.stats import zscore
#Train test split
X = df_train[features].apply(zscore)[:100000]
y=df_train['trip_duration_hour'][:100000]
# Importing train_test_split
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=42)
print(X_train.shape,y_train.shape)
print(X_test.shape,y_test.shape)
```

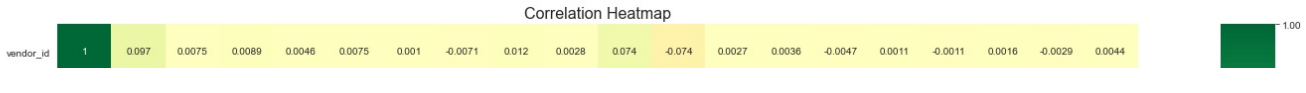
```
(80000, 19) (80000,)
(20000, 19) (20000,)
```

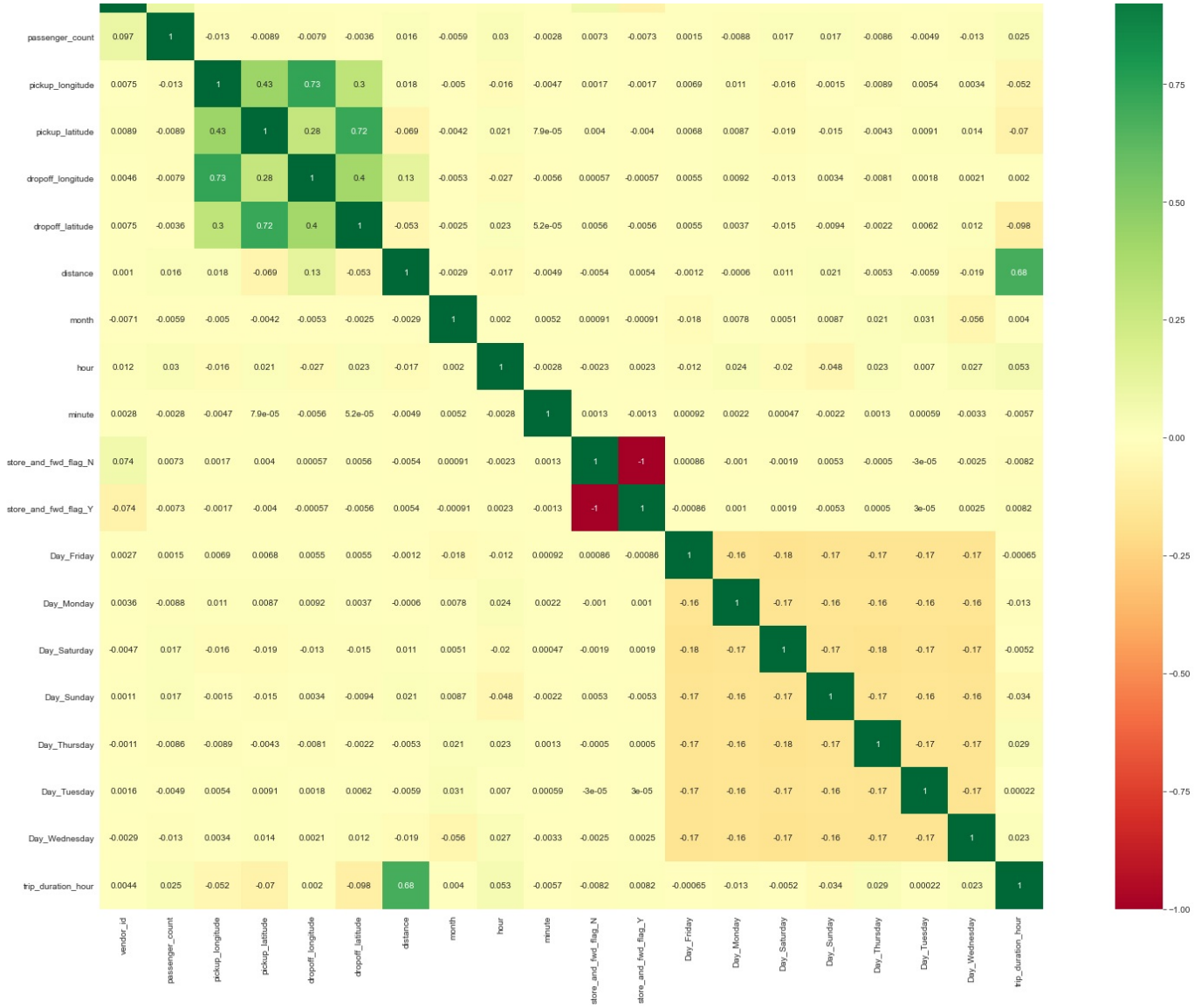
Correlation Analysis

It is a method of statistical evaluation used to study the strength of a relationship between two or more, numerically measured, continuous variables. This analysis is useful to check if there are possible connections between variables.

Heatmap:

In [81]:
```
plt.figure (figsize= (30,20))

sns.heatmap(trip_data.corr(), cmap='RdYlGn', annot=True,vmin=-1, vmax=1, square=True)
plt.title("Correlation Heatmap", fontsize=16)
plt.show()
```





In [82]:

```
from matplotlib import legend
# Function for evaluation metric for regression
def EvaluationMetric(Xt,yt,yp,disp="on"):
  ''' Take the different set of parameter and prints evaluation metrics '''
  MSE=round(mean_squared_error(y_true=yt,y_pred=yp),4)
  RMSE=(np.sqrt(MSE))
  R2=(r2_score(y_true=yt,y_pred=yp))
  Adjusted_R2=(1-(1-r2_score(yt, yp))*((Xt.shape[0]-1)/(Xt.shape[0]-Xt.shape[1]-1)))
  if disp=="on":
    print("MSE :",MSE,"RMSE :", RMSE)
    print("R2 :",R2,"Adjusted R2 :",Adjusted_R2)

  #Plotting Actual and Predicted Values
  plt.figure(figsize=(18,6))
  plt.plot((yp)[:100])
  plt.plot((np.array(yt)[:100]))
  plt.legend(["Predicted","Actual"])
  plt.title('Actual and Predicted Time Duration')



  return (MSE,RMSE,R2,Adjusted_R2)
```

Linear Regression

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependant variable. The variable used to predict the other variable value is called the independent variable.

In [84]:

```
from sklearn.linear_model import LinearRegression

reg = LinearRegression().fit(X_train, y_train)
reg.score(X_train, y_train)
```

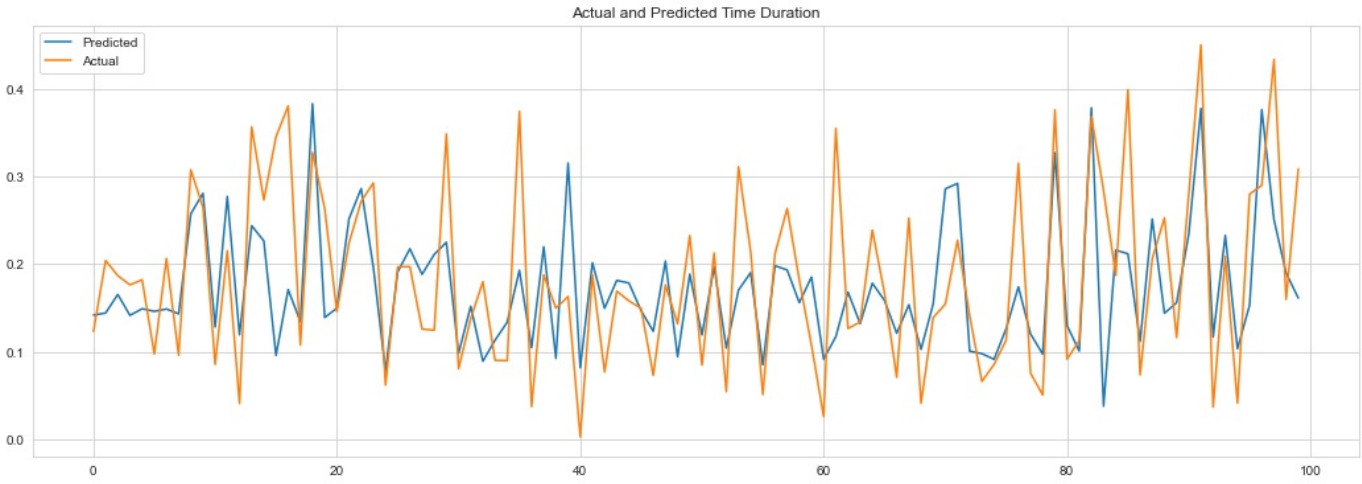
0.4850578716684644

Out[84]: 0.48505787166684644

In [87]:
```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
y_pred_train = reg.predict(X_train)
y_pred_test = reg.predict(X_test)
#Evaluation metrics for Train set
EvaluationMetric(X_train,y_train,y_pred_train)
```

```
MSE : 0.0057 RMSE : 0.0754983443527075
R2 : 0.48505787166684644 Adjusted R2 : 0.4849355423290329
```

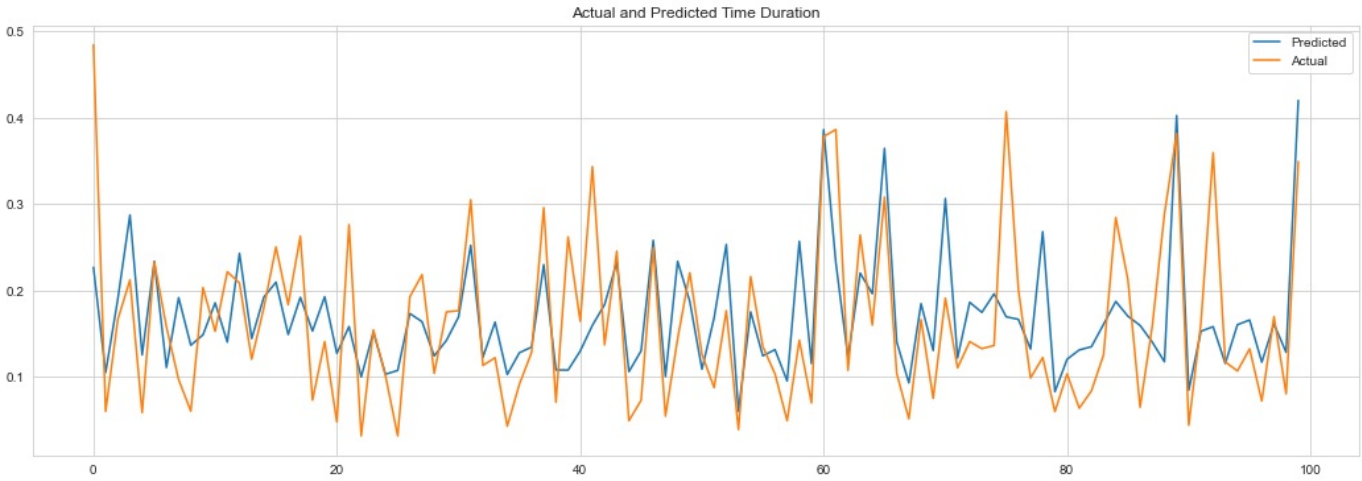
Out[87]: (0.0057, 0.0754983443527075, 0.48505787166684644, 0.4849355423290329)



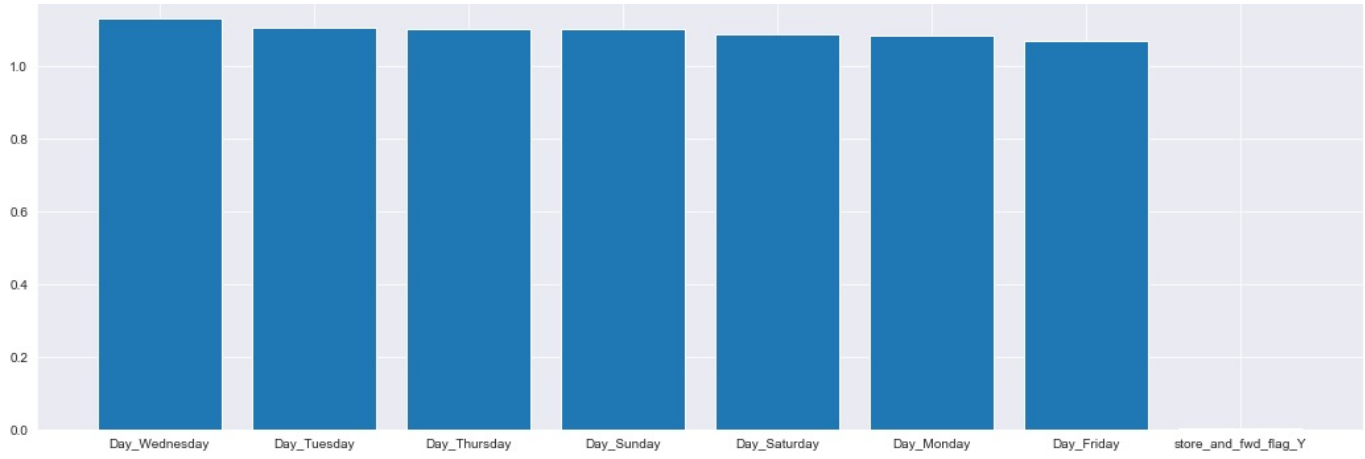
In [88]:
```
EvaluationMetric(X_test,y_test,y_pred_test)
```

```
MSE : 0.0055 RMSE : 0.07416198487095663
R2 : 0.49863317306564425 Adjusted R2 : 0.4981563978047958
```

Out[88]: (0.0055, 0.07416198487095663, 0.49863317306564425, 0.4981563978047958)



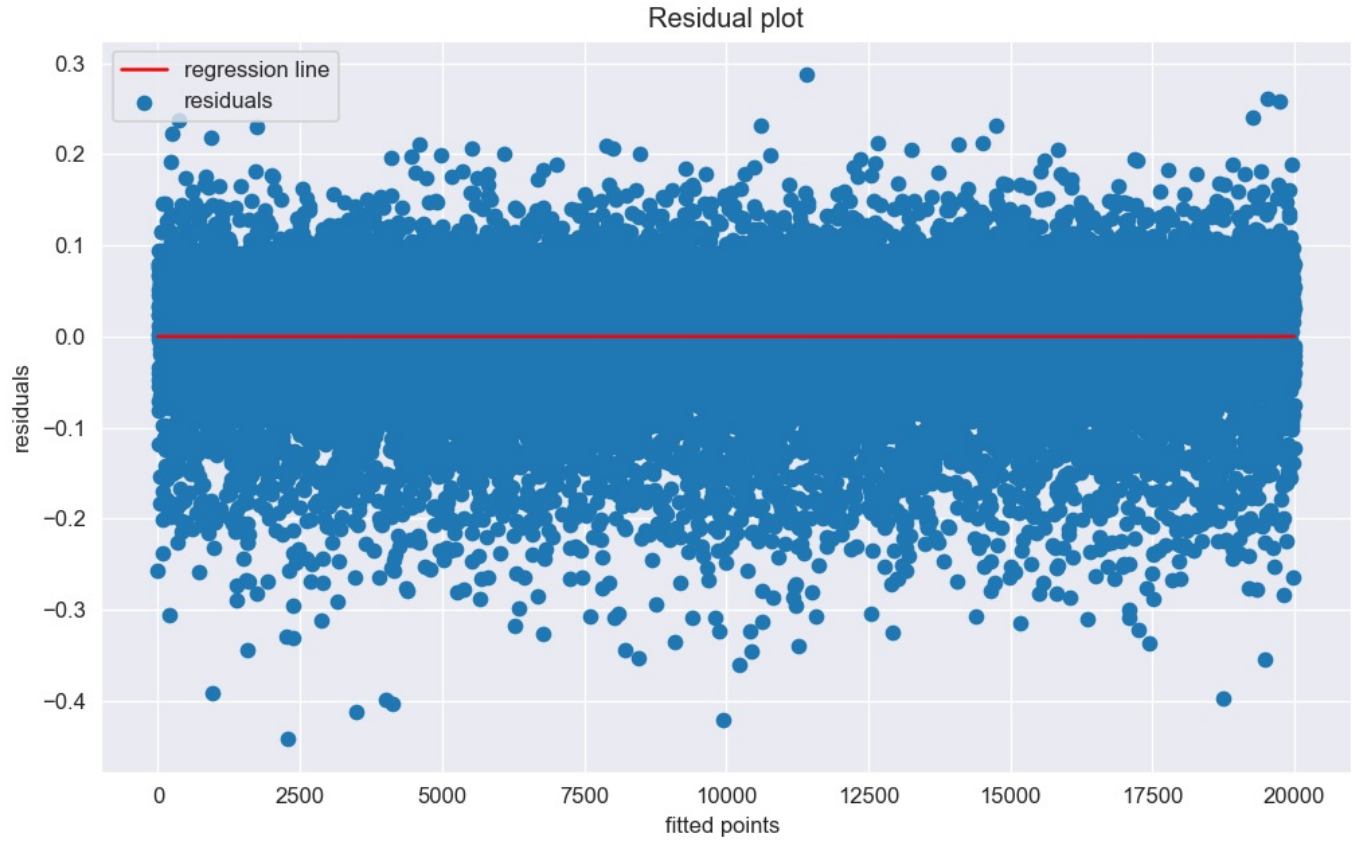
In [90]:
```
plt.figure(figsize=(18,6))

importance=reg.coef_
importance=np.sort(importance)
feature=features
indices=np.argsort(importance)
indices=indices[:10:-1]
#plotting the features and their score in ascending order
sns.set_style("darkgrid")
plt.bar(range(len(indices)),importance[indices])
plt.xticks(range(len(indices)),[feature[i] for i in indices])
plt.show()
```



In [91]:

```
residuals=y_pred_test-y_test

plt.figure(figsize=(10, 6), dpi=120, facecolor='w', edgecolor='b')
f = range(0,len(y_test))
k = [0 for i in range(0,len(y_test))]
plt.scatter( f, residuals, label = 'residuals')
plt.plot( f, k , color = 'red', label = 'regression line' )
plt.xlabel('fitted points ')
plt.ylabel('residuals')
plt.title('Residual plot')
plt.legend()
```

Out[91]: <matplotlib.legend.Legend at 0x20e83f7ae20>



Decision Tree

In [93]:

```
max_depth = [4,6,8,10]

# Minimum number of samples required to split a node
min_samples_split = [10,20,30]

# Minimum number of samples required at each leaf node
min_samples_leaf = [10,16,20]
```

```
# HYperparameter Grid
param_dt = {
              'max_depth' : max_depth,
              'min_samples_split' : min_samples_split,
              'min_samples_leaf' : min_samples_leaf}
```

In [97]:

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import GridSearchCV

dt_model = DecisionTreeRegressor()

# Grid search
dt_grid = GridSearchCV(estimator=dt_model,
                       param_grid = param_dt,
                       cv = 5, verbose=2, scoring='r2')

dt_grid.fit(X_train,y_train)
```

```
Fitting 5 folds for each of 36 candidates, totalling 180 fits
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=10; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=10; total time=   0.3s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=10; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=10; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=10; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=20; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=20; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=20; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=20; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=20; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=30; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=30; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=30; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=30; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=10, min_samples_split=30; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=10; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=10; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=10; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=10; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=10; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=20; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=20; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=20; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=20; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=20; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=30; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=30; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=30; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=30; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=16, min_samples_split=30; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=10; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=10; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=10; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=10; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=10; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=20; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=20; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=20; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=20; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=20; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=30; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=30; total time=   0.2s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=30; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=30; total time=   0.1s
[CV] END max_depth=4, min_samples_leaf=20, min_samples_split=30; total time=   0.1s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=10; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=10; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=10; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=10; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=10; total time=   0.3s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=20; total time=   0.3s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=30; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=30; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=30; total time=   0.3s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=30; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=10, min_samples_split=30; total time=   0.3s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=10; total time=   0.3s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=10; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=10; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=10; total time=   0.2s
```

```
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=10; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=20; total time=   0.3s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=30; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=30; total time=   0.3s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=30; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=30; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=16, min_samples_split=30; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=10; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=10; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=10; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=10; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=10; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=20; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=30; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=30; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=30; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=30; total time=   0.2s
[CV] END max_depth=6, min_samples_leaf=20, min_samples_split=30; total time=   0.2s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=20; total time=   0.4s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=30; total time=   0.4s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=30; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=30; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=30; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=10, min_samples_split=30; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=30; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=30; total time=   0.4s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=30; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=30; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=16, min_samples_split=30; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=10; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=20; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=30; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=30; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=30; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=30; total time=   0.3s
[CV] END max_depth=8, min_samples_leaf=20, min_samples_split=30; total time=   0.3s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=10; total time=   0.5s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=30; total time=   0.4s
```

```
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=10, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=16, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=10; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=20; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=30; total time=   0.4s
[CV] END max_depth=10, min_samples_leaf=20, min_samples_split=30; total time=   0.4s
```

Out[97]:
```
GridSearchCV(cv=5, estimator=DecisionTreeRegressor(),
             param_grid={'max_depth': [4, 6, 8, 10],
                         'min_samples_leaf': [10, 16, 20],
                         'min_samples_split': [10, 20, 30]},
             scoring='r2', verbose=2)
```

In [98]:
```
dt_grid.best_score_
```

Out[98]:
```
0.588909400597721
```

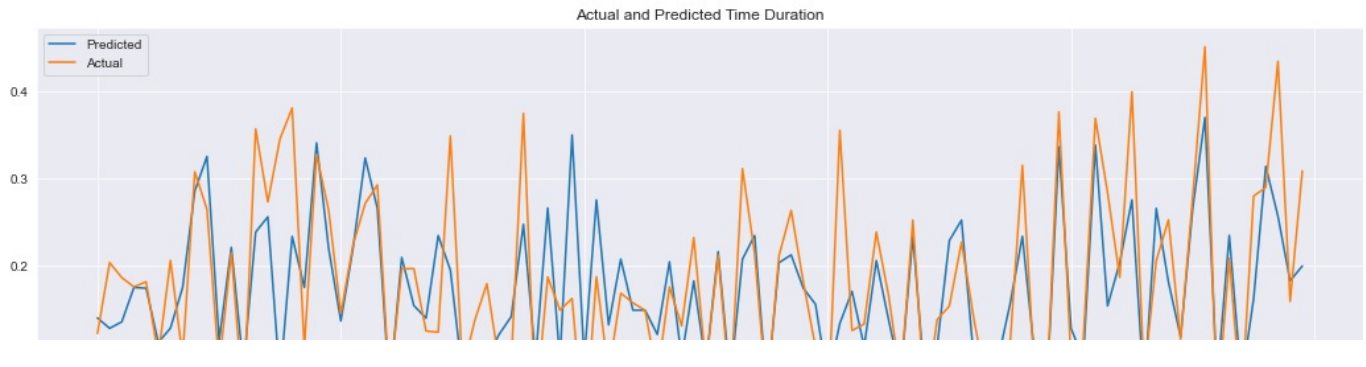
In [99]:
```
dt_grid.best_estimator_
```

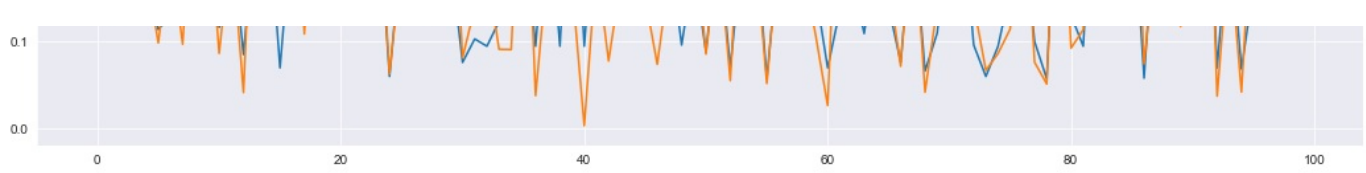
Out[99]:
```
DecisionTreeRegressor(max_depth=10, min_samples_leaf=20, min_samples_split=10)
```

In [100…
```
dt_optimal_model =dt_grid.best_estimator_
y_pred_dt_test=dt_optimal_model.predict(X_test)
y_pred_dt_train=dt_optimal_model.predict(X_train)
y_pred_dt_test=dt_optimal_model.predict(X_test)
y_pred_dt_train=dt_optimal_model.predict(X_train)

EvaluationMetric(X_train,y_train,y_pred_dt_train)
```

```
MSE : 0.0041 RMSE : 0.06403124237432849
R2 : 0.6287682669219997 Adjusted R2 : 0.6286800773379977
```

Out[100…
```
(0.0041, 0.06403124237432849, 0.6287682669219997, 0.6286800773379977)
```

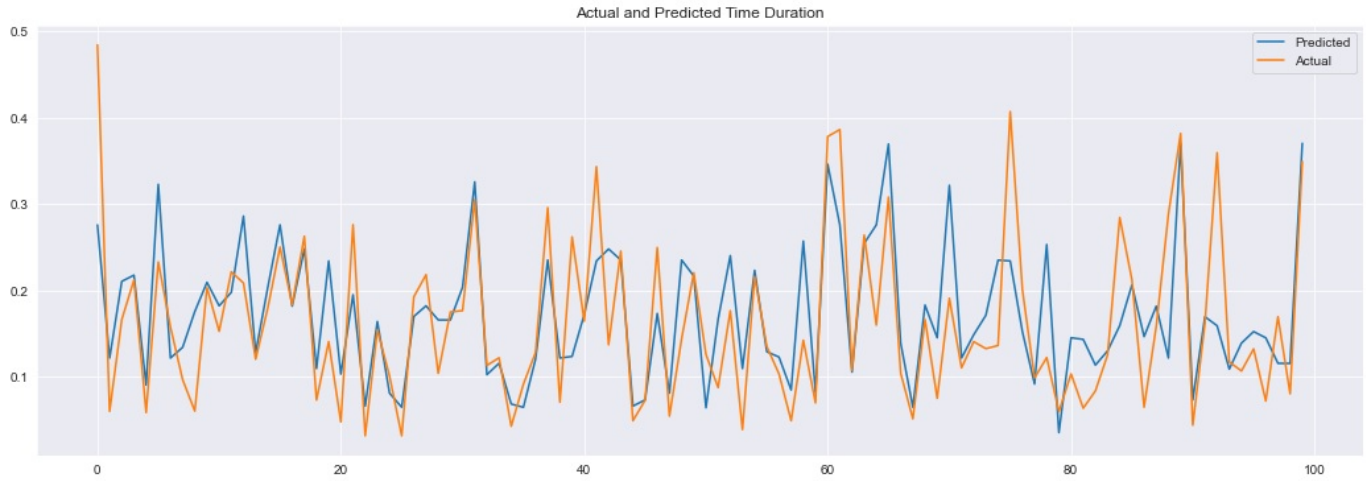




In [101…
```
EvaluationMetric(X_test,y_test,y_pred_dt_test)
```

```
MSE : 0.0044 RMSE : 0.066332495807108
R2 : 0.6064141556700413 Adjusted R2 : 0.606039874837095
```

Out[101… (0.0044, 0.066332495807108, 0.6064141556700413, 0.606039874837095)



In [102…
```
X_train.columns
```

Out[102…
```
Index(['vendor_id', 'passenger_count', 'pickup_longitude', 'pickup_latitude',
       'dropoff_longitude', 'dropoff_latitude', 'distance', 'month', 'hour',
       'minute', 'store_and_fwd_flag_N', 'store_and_fwd_flag_Y', 'Day_Friday',
       'Day_Monday', 'Day_Saturday', 'Day_Sunday', 'Day_Thursday',
       'Day_Tuesday', 'Day_Wednesday'],
      dtype='object')
```

In [103…
```
dt_optimal_model.feature_importances_
```

Out[103…
```
array([9.09773243e-04, 0.00000000e+00, 1.43525182e-02, 8.44322132e-03,
       1.65319218e-02, 5.71145153e-02, 8.09637127e-01, 8.93910345e-04,
       8.35173756e-02, 2.18859759e-03, 0.00000000e+00, 0.00000000e+00,
       1.36891111e-04, 2.27695030e-05, 3.11782537e-04, 3.69228812e-03,
       4.57209202e-04, 0.00000000e+00, 1.79009956e-03])
```

In [104…
```
importances = dt_optimal_model.feature_importances_

importance_dict = {'Feature' : list(X_train.columns),
                   'Feature Importance' : importances}

importance_df = pd.DataFrame(importance_dict)
importance_df.sort_values(by=['Feature Importance'],ascending=False,inplace=True)
importance_df
```

Out[104…

	Feature	Feature Importance
6	distance	0.809637
8	hour	0.083517
5	dropoff_latitude	0.057115
4	dropoff_longitude	0.016532
2	pickup_longitude	0.014353
3	pickup_latitude	0.008443

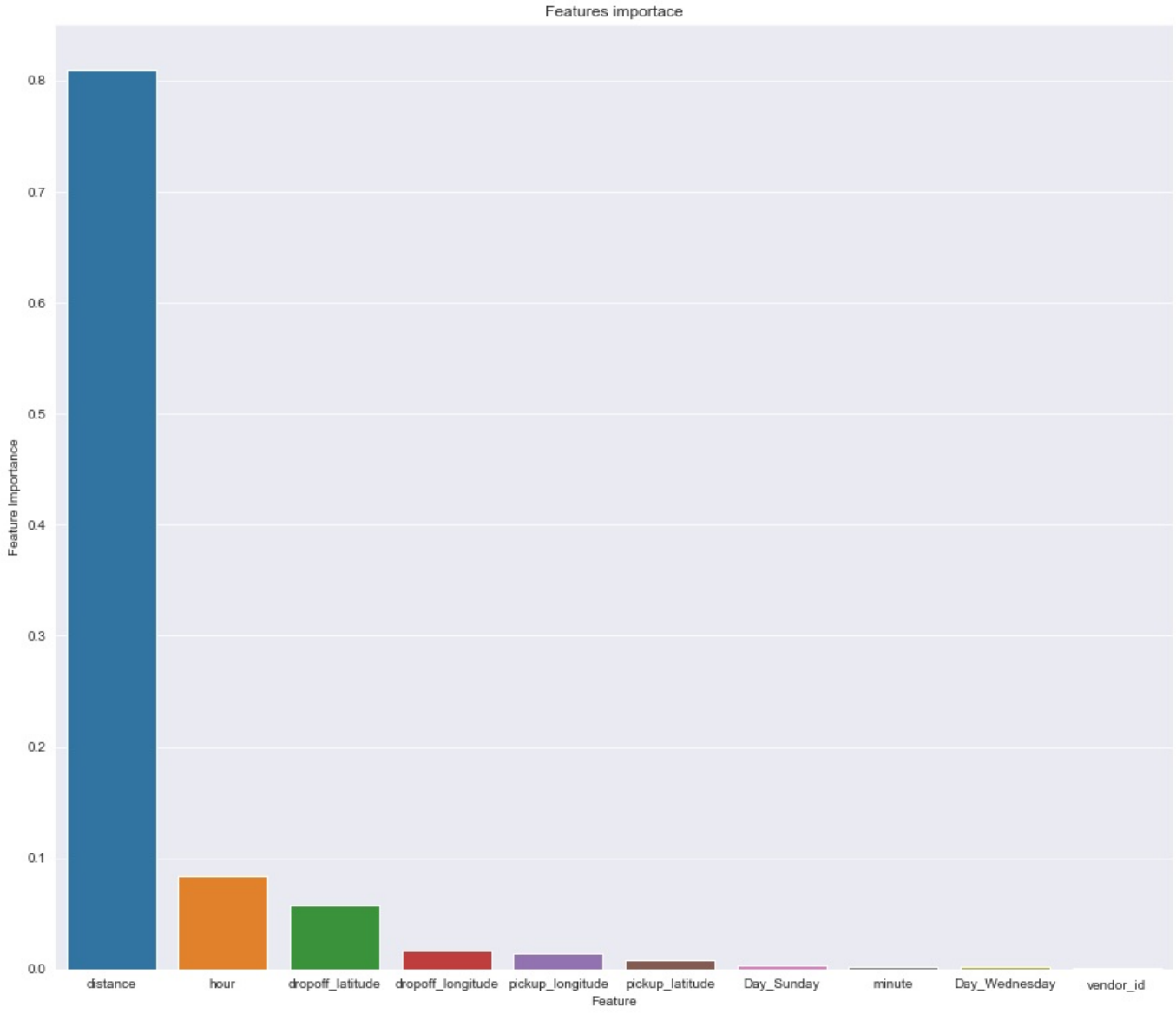
15	Day_Sunday	0.003692
9	minute	0.002189
18	Day_Wednesday	0.001790
0	vendor_id	0.000910
7	month	0.000894
16	Day_Thursday	0.000457
14	Day_Saturday	0.000312
12	Day_Friday	0.000137
13	Day_Monday	0.000023
1	passenger_count	0.000000
10	store_and_fwd_flag_N	0.000000
11	store_and_fwd_flag_Y	0.000000
17	Day_Tuesday	0.000000

In [105…

```
plt.figure(figsize=(15,13))
plt.title('Features importace')
sns.barplot(x='Feature',y="Feature Importance",data=importance_df[:10])
```

Out[105…

```
<AxesSubplot:title={'center':'Features importace'}, xlabel='Feature', ylabel='Feature Importance'>
```



In [109…

```
!pip install graphviz
```

```
Collecting graphviz
  Downloading graphviz-0.20.1-py3-none-any.whl (47 kB)
```

```
Installing collected packages: graphviz
Successfully installed graphviz-0.20.1
```

In [111…

```
from sklearn.tree import DecisionTreeRegressor, export_graphviz
from sklearn import tree
from IPython.display import SVG
from graphviz import Source
from IPython.display import display
```

Lasso Regression

In [115…

```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Lasso

### Cross validation
lasso = Lasso()
parameters = {'alpha': [1e-15,1e-13,1e-10,1e-8,1e-5,1e-4,1e-3,1e-2,1e-1,1,5,10,20,30,40,45,50,55,60,100]}
lasso_regressor = GridSearchCV(lasso, parameters, scoring='r2', cv=5)
lasso_regressor.fit(X_train, y_train)
```

```
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 8.969093141750761,
tolerance: 0.07062055250915078
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 8.901245046797612,
tolerance: 0.07031670361672404
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 9.079678662574736,
tolerance: 0.0704606664548148
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 9.108381238960021,
tolerance: 0.06997086072111833
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 8.934441518823576,
tolerance: 0.0706127930552657
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 8.950939929684353,
tolerance: 0.07062055250915078
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 8.877045539909034,
tolerance: 0.07031670361672404
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 9.056445315975793,
tolerance: 0.0704606664548148
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 9.046568835188054,
tolerance: 0.06997086072111833
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 8.88686307683065, t
olerance: 0.0706127930552657
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 8.950339338687456,
tolerance: 0.07062055250915078
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 8.877527268310985,
tolerance: 0.07031670361672404
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 9.056033059110518,
tolerance: 0.0704606664548148
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 9.046339140515101,
tolerance: 0.06997086072111833
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
```

```
Objective did not converge. You might want to increase the number of iterations. Duality gap: 8.886139845755736,
tolerance: 0.0706127930552657
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 8.950358187009897,
tolerance: 0.07062055250915078
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 8.877546038064906,
tolerance: 0.07031670361672404
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 9.056052559864213,
tolerance: 0.0704606664548148
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 9.046358049482933,
tolerance: 0.06997086072111833
  model = cd_fast.enet_coordinate_descent(
C:\Users\Meghna\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:530: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations. Duality gap: 8.88615902413028, t
olerance: 0.0706127930552657
  model = cd_fast.enet_coordinate_descent(
```

Out[115…
```
GridSearchCV(cv=5, estimator=Lasso(),
             param_grid={'alpha': [1e-15, 1e-13, 1e-10, 1e-08, 1e-05, 0.0001,
                                   0.001, 0.01, 0.1, 1, 5, 10, 20, 30, 40, 45,
                                   50, 55, 60, 100]},
             scoring='r2')
```

In [116…
```
lasso_regressor.score(X_train, y_train)
```

Out[116…
```
0.4850683497833108
```

In [117…
```
print("The best fit alpha value is found out to be :" ,lasso_regressor.best_params_)
print("\nUsing ",lasso_regressor.best_params_, " the negative mean squared error is: ", lasso_regressor.best_scor
```

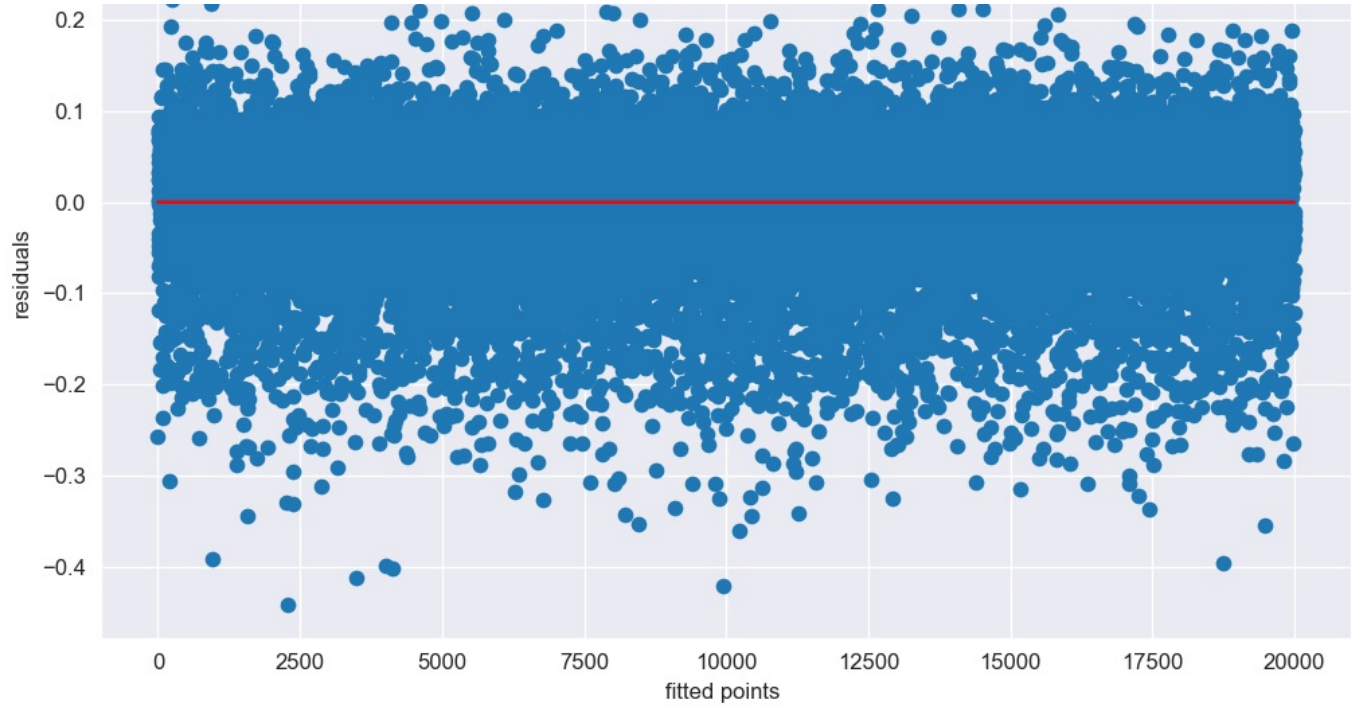
```
The best fit alpha value is found out to be : {'alpha': 1e-05}

Using  {'alpha': 1e-05}  the negative mean squared error is:  0.48467615581453877
```

In [120…
```
from sklearn.linear_model import Ridge


ridge_regressor = Ridge(alpha=1.0)
ridge_regressor.fit(X_train, y_train)
y_pred_ridge_test = ridge_regressor.predict(X_test)
y_pred_ridge_train = ridge_regressor.predict(X_train)
```

In [121…
```
y_pred_ridge_test = ridge_regressor.predict(X_test)
y_pred_ridge_train=ridge_regressor.predict(X_train)
```

In [122…
```
residuals=y_pred_ridge_test-y_test

plt.figure(figsize=(10, 6), dpi=120, facecolor='w', edgecolor='b')
f = range(0,len(y_test))
k = [0 for i in range(0,len(y_test))]
plt.scatter( f, residuals, label = 'residuals')
plt.plot( f, k , color = 'red', label = 'regression line' )
plt.xlabel('fitted points ')
plt.ylabel('residuals')
plt.title('Residual plot')
plt.legend()
```

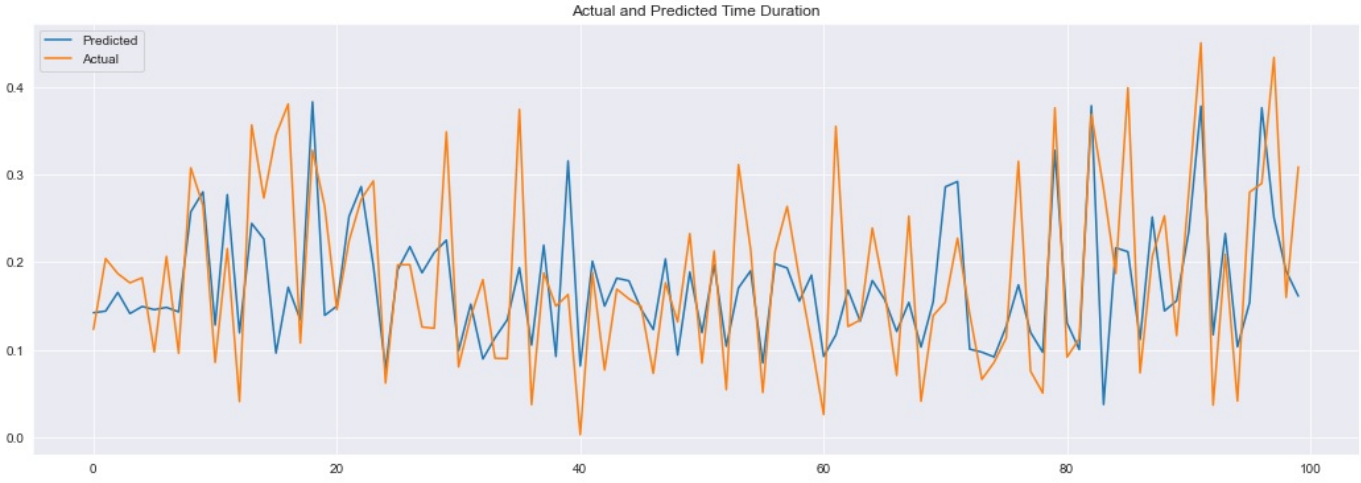
Out[122…
```
<matplotlib.legend.Legend at 0x20e80ec6f10>
```



In [123…
```
EvaluationMetric(X_train,y_train,y_pred_ridge_train)
```

```
MSE : 0.0057 RMSE : 0.0754983443527075
R2 : 0.48506855725471154 Adjusted R2 : 0.4849462304553598
```

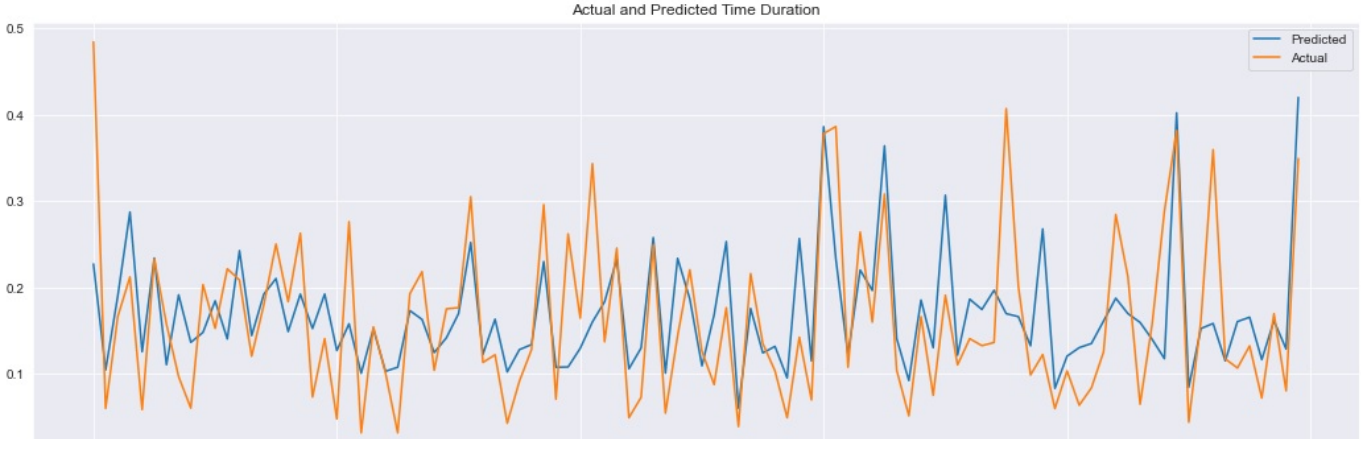
Out[123… (0.0057, 0.0754983443527075, 0.48506855725471154, 0.4849462304553598)



In [124…
```
EvaluationMetric(X_test,y_test,y_pred_ridge_test)
```

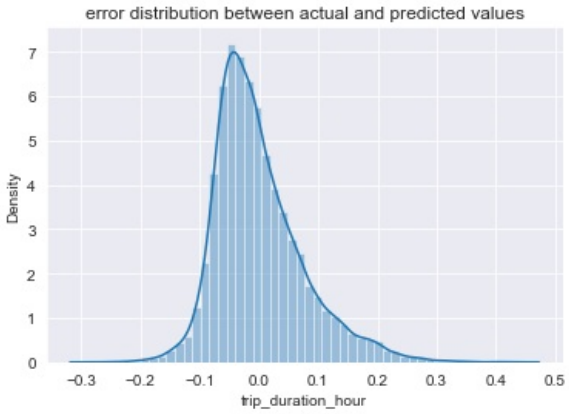
```
MSE : 0.0055 RMSE : 0.07416198487095663
R2 : 0.49864916400465864 Adjusted R2 : 0.49817240395040885
```

Out[124… (0.0055, 0.07416198487095663, 0.49864916400465864, 0.49817240395040885)



In [125…
```
sns.distplot(y_test - y_pred_ridge_test ).set_title("error distribution between actual and predicted values")
plt.show()
```

```
C:\Users\Meghna\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



XGBoost

It is an optimized distributed gradient boosting library designed to be higly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework.

In [126…
```
# Number of trees
n_estimators = [50,100,120]

# Maximum depth of trees
max_depth = [5,7,9]
min_samples_split = [40,50]
#learning_rate=[0.1,0.3,0.5]

# HYperparameter Grid
param_xgb = {'n_estimators' : n_estimators,
             'max_depth' : max_depth,
            'min_samples_split':min_samples_split
             }
```

In [127…
```
import xgboost as xgb
xgb_model = xgb.XGBRegressor()

# Grid search
xgb_grid = GridSearchCV(estimator=xgb_model,param_grid = param_xgb,cv = 3, verbose=2,scoring="r2")

xgb_grid.fit(X_train,y_train)
```

```
Fitting 3 folds for each of 18 candidates, totalling 54 fits
[09:37:37] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboost\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=5, min_samples_split=40, n_estimators=50; total time=   1.0s
[09:37:38] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboost\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=5, min_samples_split=40, n_estimators=50; total time=   1.0s
[09:37:40] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboost\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=5, min_samples_split=40, n_estimators=50; total time=   0.9s
[09:37:41] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
```

```
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=5, min_samples_split=40, n_estimators=100; total time=   1.9s
[09:37:43] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=5, min_samples_split=40, n_estimators=100; total time=   1.5s
[09:37:44] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=5, min_samples_split=40, n_estimators=100; total time=   1.5s
[09:37:46] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=5, min_samples_split=40, n_estimators=120; total time=   1.8s
[09:37:48] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=5, min_samples_split=40, n_estimators=120; total time=   1.9s
[09:37:50] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=5, min_samples_split=40, n_estimators=120; total time=   1.9s
[09:37:52] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=5, min_samples_split=50, n_estimators=50; total time=   0.8s
[09:37:53] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=5, min_samples_split=50, n_estimators=50; total time=   0.8s
[09:37:54] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=5, min_samples_split=50, n_estimators=50; total time=   0.7s
[09:37:54] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=5, min_samples_split=50, n_estimators=100; total time=   1.5s
[09:37:56] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=5, min_samples_split=50, n_estimators=100; total time=   1.5s
[09:37:57] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=5, min_samples_split=50, n_estimators=100; total time=   1.5s
[09:37:59] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=5, min_samples_split=50, n_estimators=120; total time=   1.8s
[09:38:01] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=5, min_samples_split=50, n_estimators=120; total time=   1.7s
[09:38:03] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=5, min_samples_split=50, n_estimators=120; total time=   1.8s
[09:38:05] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=7, min_samples_split=40, n_estimators=50; total time=   1.1s
[09:38:06] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.
```

```
[CV] END .max_depth=7, min_samples_split=40, n_estimators=50; total time=   1.1s
[09:38:07] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=7, min_samples_split=40, n_estimators=50; total time=   1.2s
[09:38:08] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=7, min_samples_split=40, n_estimators=100; total time=   2.3s
[09:38:11] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=7, min_samples_split=40, n_estimators=100; total time=   2.2s
[09:38:13] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=7, min_samples_split=40, n_estimators=100; total time=   2.2s
[09:38:15] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=7, min_samples_split=40, n_estimators=120; total time=   2.6s
[09:38:18] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=7, min_samples_split=40, n_estimators=120; total time=   2.6s
[09:38:21] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=7, min_samples_split=40, n_estimators=120; total time=   2.9s
[09:38:24] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=7, min_samples_split=50, n_estimators=50; total time=   1.2s
[09:38:25] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=7, min_samples_split=50, n_estimators=50; total time=   1.1s
[09:38:26] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=7, min_samples_split=50, n_estimators=50; total time=   1.2s
[09:38:27] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=7, min_samples_split=50, n_estimators=100; total time=   2.2s
[09:38:30] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=7, min_samples_split=50, n_estimators=100; total time=   2.1s
[09:38:32] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=7, min_samples_split=50, n_estimators=100; total time=   2.1s
[09:38:34] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=7, min_samples_split=50, n_estimators=120; total time=   2.6s
[09:38:37] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=7, min_samples_split=50, n_estimators=120; total time=   2.8s
[09:38:40] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=7, min_samples_split=50, n_estimators=120; total time=   2.7s
[09:38:43] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
```

```
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=9, min_samples_split=40, n_estimators=50; total time=   1.4s
[09:38:44] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=9, min_samples_split=40, n_estimators=50; total time=   1.5s
[09:38:46] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=9, min_samples_split=40, n_estimators=50; total time=   1.5s
[09:38:47] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=9, min_samples_split=40, n_estimators=100; total time=   2.9s
[09:38:50] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=9, min_samples_split=40, n_estimators=100; total time=   3.0s
[09:38:53] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=9, min_samples_split=40, n_estimators=100; total time=   3.1s
[09:38:57] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=9, min_samples_split=40, n_estimators=120; total time=   3.6s
[09:39:00] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=9, min_samples_split=40, n_estimators=120; total time=   3.5s
[09:39:04] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=9, min_samples_split=40, n_estimators=120; total time=   3.5s
[09:39:07] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=9, min_samples_split=50, n_estimators=50; total time=   1.4s
[09:39:09] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=9, min_samples_split=50, n_estimators=50; total time=   1.6s
[09:39:11] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END .max_depth=9, min_samples_split=50, n_estimators=50; total time=   1.6s
[09:39:12] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=9, min_samples_split=50, n_estimators=100; total time=   3.0s
[09:39:16] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=9, min_samples_split=50, n_estimators=100; total time=   2.9s
[09:39:19] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=9, min_samples_split=50, n_estimators=100; total time=   2.9s
[09:39:22] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=9, min_samples_split=50, n_estimators=120; total time=   3.6s
[09:39:25] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=9, min_samples_split=50, n_estimators=120; total time=   3.8s
```

```
[09:39:29] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.

[CV] END max_depth=9, min_samples_split=50, n_estimators=120; total time=   3.4s
[09:39:33] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0fdc6d574b9c0d168-1\xgboo
st\xgboost-ci-windows\src\learner.cc:767:
Parameters: { "min_samples_split" } are not used.
```

Out[127…

```
GridSearchCV(cv=3,
             estimator=XGBRegressor(base_score=None, booster=None,
                                    callbacks=None, colsample_bylevel=None,
                                    colsample_bynode=None,
                                    colsample_bytree=None,
                                    early_stopping_rounds=None,
                                    enable_categorical=False, eval_metric=None,
                                    feature_types=None, gamma=None, gpu_id=None,
                                    grow_policy=None, importance_type=None,
                                    interaction_constraints=None,
                                    learning_rate=None, m...
                                    max_cat_threshold=None,
                                    max_cat_to_onehot=None, max_delta_step=None,
                                    max_depth=None, max_leaves=None,
                                    min_child_weight=None, missing=nan,
                                    monotone_constraints=None, n_estimators=100,
                                    n_jobs=None, num_parallel_tree=None,
                                    predictor=None, random_state=None, ...),
             param_grid={'max_depth': [5, 7, 9], 'min_samples_split': [40, 50],
                         'n_estimators': [50, 100, 120]},
             scoring='r2', verbose=2)
```

In [129…

```
xgb_grid.best_score_
```

Out[129… 0.6701216675464057

In [130…

```
xgb_grid.best_params_
```

Out[130… {'max_depth': 7, 'min_samples_split': 40, 'n_estimators': 100}

In [131…

```
xgb_optimal_model =xgb_grid.best_estimator_
```

In [132…

```
y_pred_xgb_test=xgb_optimal_model.predict(X_test)
y_pred_xgb_train=xgb_optimal_model.predict(X_train)
```

In [133…

```
EvaluationMetric(X_train,y_train,y_pred_xgb_train)
```

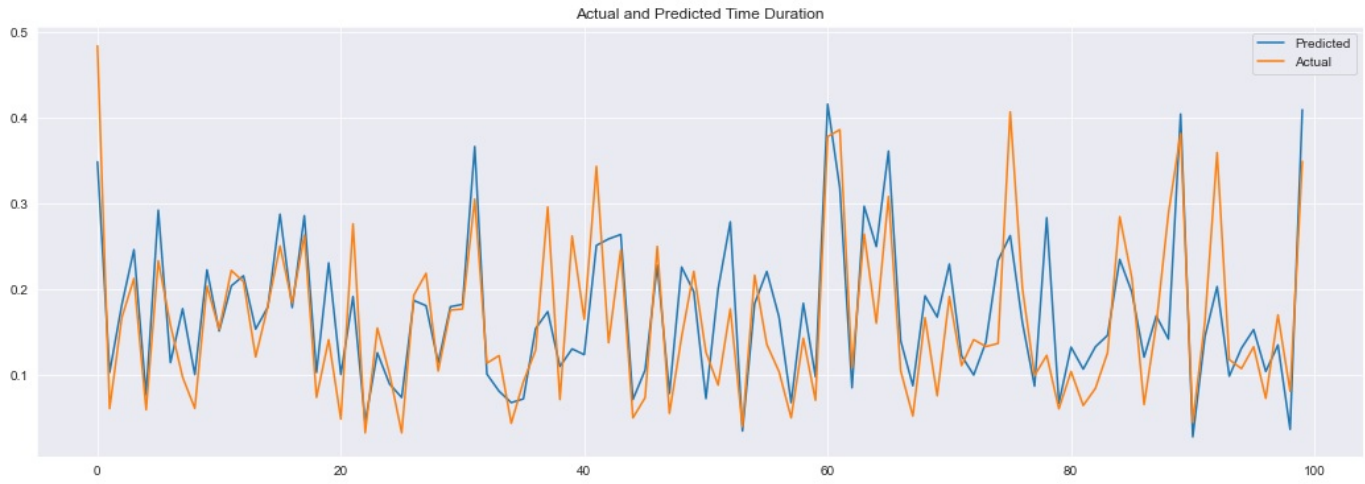
```
MSE : 0.0024 RMSE : 0.04898979485566356
R2 : 0.7783328851838611 Adjusted R2 : 0.7782802260793162
```

Out[133… (0.0024, 0.04898979485566356, 0.7783328851838611, 0.7782802260793162)

In [134…
```
EvaluationMetric(X_test,y_test,y_pred_xgb_test)
```

```
MSE : 0.0035 RMSE : 0.05916079783099616
R2 : 0.6867155158525247 Adjusted R2 : 0.6864175976744065
```

Out[134… (0.0035, 0.05916079783099616, 0.6867155158525247, 0.6864175976744065)



In [135…
```
X_train.columns
```

Out[135…
```
Index(['vendor_id', 'passenger_count', 'pickup_longitude', 'pickup_latitude',
       'dropoff_longitude', 'dropoff_latitude', 'distance', 'month', 'hour',
       'minute', 'store_and_fwd_flag_N', 'store_and_fwd_flag_Y', 'Day_Friday',
       'Day_Monday', 'Day_Saturday', 'Day_Sunday', 'Day_Thursday',
       'Day_Tuesday', 'Day_Wednesday'],
      dtype='object')
```

In [136…
```
xgb_optimal_model.feature_importances_
```

Out[136…
```
array([0.0082792 , 0.00720963, 0.02233937, 0.02442116, 0.02776379,
       0.04846492, 0.5192163 , 0.02155587, 0.08768707, 0.01112846,
       0.01028003, 0.        , 0.01661176, 0.01953007, 0.03899084,
       0.05096768, 0.0324048 , 0.01794679, 0.03520219], dtype=float32)
```

In [137…
```
importances = xgb_optimal_model.feature_importances_

importance_dict = {'Feature' : list(X_train.columns),
                   'Feature Importance' : importances}

importance_df = pd.DataFrame(importance_dict)
```

In [138…
```
importance_df.sort_values(by=['Feature Importance'],ascending=False,inplace=True)
importance_df
```

Out[138…

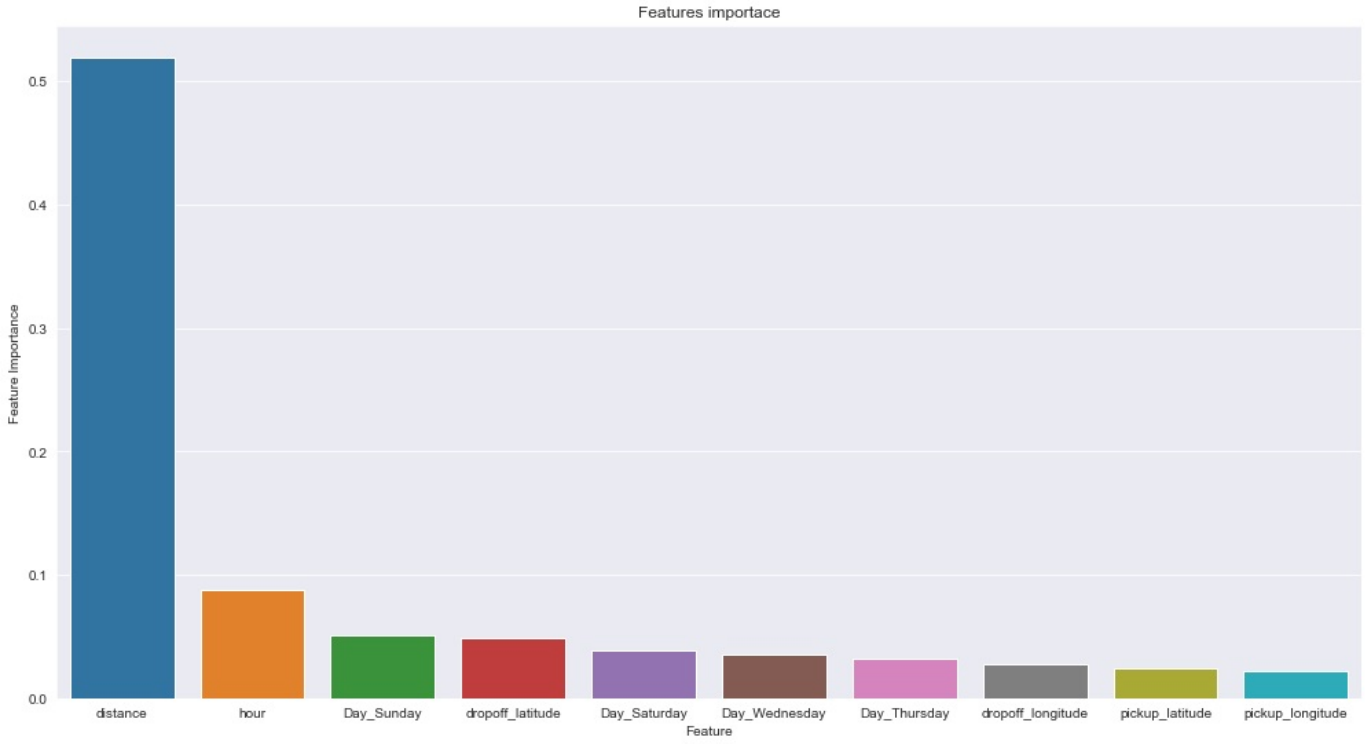
	Feature	Feature Importance
6	distance	0.519216
8	hour	0.087687
15	Day_Sunday	0.050968
5	dropoff_latitude	0.048465
14	Day_Saturday	0.038991
18	Day_Wednesday	0.035202
16	Day_Thursday	0.032405
4	dropoff_longitude	0.027764
3	pickup_latitude	0.024421

2	pickup_longitude	0.022339
7	month	0.021556
13	Day_Monday	0.019530
17	Day_Tuesday	0.017947
12	Day_Friday	0.016612
9	minute	0.011128
10	store_and_fwd_flag_N	0.010280
0	vendor_id	0.008279
1	passenger_count	0.007210
11	store_and_fwd_flag_Y	0.000000

In [139…

```
plt.figure(figsize=(17,9))
plt.title('Features importace')
sns.barplot(x='Feature',y="Feature Importance",data=importance_df[:10])
```

Out[139… <AxesSubplot:title={'center':'Features importace'}, xlabel='Feature', ylabel='Feature Importance'>



GradientBoosting

In [140…

```
# Number of trees
n_estimators = [100,120]

# Maximum depth of trees
max_depth = [5,8,10]

# Minimum number of samples required to split a node
min_samples_split = [50,80]

# Minimum number of samples required at each leaf node
min_samples_leaf = [40,50]


# HYperparameter Grid
param_gb = {'n_estimators' : n_estimators,
            'max_depth' : max_depth,
            'min_samples_split' : min_samples_split,
            'min_samples_leaf' : min_samples_leaf}
```

In [141…

```
# Create an instance of the  GradientBoostingRegressor
from sklearn.ensemble import GradientBoostingRegressor
```

```
gb_model=GradientBoostingRegressor()

# Grid search
gb_grid = GridSearchCV(estimator=gb_model,
                       param_grid = param_gb,
                       cv = 3, verbose=2, scoring='r2')

gb_grid.fit(X_train,y_train)
```

```
Fitting 3 folds for each of 24 candidates, totalling 72 fits
[CV] END max_depth=5, min_samples_leaf=40, min_samples_split=50, n_estimators=100; total time=  24.0s
[CV] END max_depth=5, min_samples_leaf=40, min_samples_split=50, n_estimators=100; total time=  21.8s
[CV] END max_depth=5, min_samples_leaf=40, min_samples_split=50, n_estimators=100; total time=  22.2s
[CV] END max_depth=5, min_samples_leaf=40, min_samples_split=50, n_estimators=120; total time=  26.3s
[CV] END max_depth=5, min_samples_leaf=40, min_samples_split=50, n_estimators=120; total time=  26.1s
[CV] END max_depth=5, min_samples_leaf=40, min_samples_split=50, n_estimators=120; total time=  26.3s
[CV] END max_depth=5, min_samples_leaf=40, min_samples_split=80, n_estimators=100; total time=  22.0s
[CV] END max_depth=5, min_samples_leaf=40, min_samples_split=80, n_estimators=100; total time=  21.8s
[CV] END max_depth=5, min_samples_leaf=40, min_samples_split=80, n_estimators=100; total time=  21.4s
[CV] END max_depth=5, min_samples_leaf=40, min_samples_split=80, n_estimators=120; total time=  26.2s
[CV] END max_depth=5, min_samples_leaf=40, min_samples_split=80, n_estimators=120; total time=  25.0s
[CV] END max_depth=5, min_samples_leaf=40, min_samples_split=80, n_estimators=120; total time=  24.6s
[CV] END max_depth=5, min_samples_leaf=50, min_samples_split=50, n_estimators=100; total time=  20.6s
[CV] END max_depth=5, min_samples_leaf=50, min_samples_split=50, n_estimators=100; total time=  20.8s
[CV] END max_depth=5, min_samples_leaf=50, min_samples_split=50, n_estimators=100; total time=  20.5s
[CV] END max_depth=5, min_samples_leaf=50, min_samples_split=50, n_estimators=120; total time=  25.0s
[CV] END max_depth=5, min_samples_leaf=50, min_samples_split=50, n_estimators=120; total time=  25.7s
[CV] END max_depth=5, min_samples_leaf=50, min_samples_split=50, n_estimators=120; total time=  26.0s
[CV] END max_depth=5, min_samples_leaf=50, min_samples_split=80, n_estimators=100; total time=  21.9s
[CV] END max_depth=5, min_samples_leaf=50, min_samples_split=80, n_estimators=100; total time=  22.2s
[CV] END max_depth=5, min_samples_leaf=50, min_samples_split=80, n_estimators=100; total time=  21.7s
[CV] END max_depth=5, min_samples_leaf=50, min_samples_split=80, n_estimators=120; total time=  26.5s
[CV] END max_depth=5, min_samples_leaf=50, min_samples_split=80, n_estimators=120; total time=  26.1s
[CV] END max_depth=5, min_samples_leaf=50, min_samples_split=80, n_estimators=120; total time=  25.2s
[CV] END max_depth=8, min_samples_leaf=40, min_samples_split=50, n_estimators=100; total time=  33.2s
[CV] END max_depth=8, min_samples_leaf=40, min_samples_split=50, n_estimators=100; total time=  33.6s
[CV] END max_depth=8, min_samples_leaf=40, min_samples_split=50, n_estimators=100; total time=  34.1s
[CV] END max_depth=8, min_samples_leaf=40, min_samples_split=50, n_estimators=120; total time=  40.5s
[CV] END max_depth=8, min_samples_leaf=40, min_samples_split=50, n_estimators=120; total time=  40.3s
[CV] END max_depth=8, min_samples_leaf=40, min_samples_split=50, n_estimators=120; total time= 1.1min
[CV] END max_depth=8, min_samples_leaf=40, min_samples_split=80, n_estimators=100; total time=  32.0s
[CV] END max_depth=8, min_samples_leaf=40, min_samples_split=80, n_estimators=100; total time=  24.3s
[CV] END max_depth=8, min_samples_leaf=40, min_samples_split=80, n_estimators=100; total time=  24.0s
[CV] END max_depth=8, min_samples_leaf=40, min_samples_split=80, n_estimators=120; total time=  30.0s
[CV] END max_depth=8, min_samples_leaf=40, min_samples_split=80, n_estimators=120; total time=  32.8s
[CV] END max_depth=8, min_samples_leaf=40, min_samples_split=80, n_estimators=120; total time=  31.8s
[CV] END max_depth=8, min_samples_leaf=50, min_samples_split=50, n_estimators=100; total time=  26.8s
[CV] END max_depth=8, min_samples_leaf=50, min_samples_split=50, n_estimators=100; total time=  27.2s
[CV] END max_depth=8, min_samples_leaf=50, min_samples_split=50, n_estimators=100; total time=  27.2s
[CV] END max_depth=8, min_samples_leaf=50, min_samples_split=50, n_estimators=120; total time=  32.5s
[CV] END max_depth=8, min_samples_leaf=50, min_samples_split=50, n_estimators=120; total time=  33.0s
[CV] END max_depth=8, min_samples_leaf=50, min_samples_split=50, n_estimators=120; total time=  32.9s
[CV] END max_depth=8, min_samples_leaf=50, min_samples_split=80, n_estimators=100; total time=  29.8s
[CV] END max_depth=8, min_samples_leaf=50, min_samples_split=80, n_estimators=100; total time=  28.1s
[CV] END max_depth=8, min_samples_leaf=50, min_samples_split=80, n_estimators=100; total time=  24.7s
[CV] END max_depth=8, min_samples_leaf=50, min_samples_split=80, n_estimators=120; total time=  29.6s
[CV] END max_depth=8, min_samples_leaf=50, min_samples_split=80, n_estimators=120; total time=  29.1s
[CV] END max_depth=8, min_samples_leaf=50, min_samples_split=80, n_estimators=120; total time=  29.4s
[CV] END max_depth=10, min_samples_leaf=40, min_samples_split=50, n_estimators=100; total time=  31.2s
[CV] END max_depth=10, min_samples_leaf=40, min_samples_split=50, n_estimators=100; total time=  29.5s
[CV] END max_depth=10, min_samples_leaf=40, min_samples_split=50, n_estimators=100; total time=  29.8s
[CV] END max_depth=10, min_samples_leaf=40, min_samples_split=50, n_estimators=120; total time=  35.7s
[CV] END max_depth=10, min_samples_leaf=40, min_samples_split=50, n_estimators=120; total time=  36.7s
[CV] END max_depth=10, min_samples_leaf=40, min_samples_split=50, n_estimators=120; total time=  36.1s
[CV] END max_depth=10, min_samples_leaf=40, min_samples_split=80, n_estimators=100; total time=  29.1s
[CV] END max_depth=10, min_samples_leaf=40, min_samples_split=80, n_estimators=100; total time=  29.7s
[CV] END max_depth=10, min_samples_leaf=40, min_samples_split=80, n_estimators=100; total time=  29.7s
[CV] END max_depth=10, min_samples_leaf=40, min_samples_split=80, n_estimators=120; total time=  35.0s
[CV] END max_depth=10, min_samples_leaf=40, min_samples_split=80, n_estimators=120; total time=  35.5s
[CV] END max_depth=10, min_samples_leaf=40, min_samples_split=80, n_estimators=120; total time=  35.9s
[CV] END max_depth=10, min_samples_leaf=50, min_samples_split=50, n_estimators=100; total time=  29.8s
[CV] END max_depth=10, min_samples_leaf=50, min_samples_split=50, n_estimators=100; total time=  28.9s
[CV] END max_depth=10, min_samples_leaf=50, min_samples_split=50, n_estimators=100; total time=  29.0s
[CV] END max_depth=10, min_samples_leaf=50, min_samples_split=50, n_estimators=120; total time=  35.2s
[CV] END max_depth=10, min_samples_leaf=50, min_samples_split=50, n_estimators=120; total time=  35.2s
[CV] END max_depth=10, min_samples_leaf=50, min_samples_split=50, n_estimators=120; total time=  35.1s
[CV] END max_depth=10, min_samples_leaf=50, min_samples_split=80, n_estimators=100; total time=  29.2s
[CV] END max_depth=10, min_samples_leaf=50, min_samples_split=80, n_estimators=100; total time=  29.2s
[CV] END max_depth=10, min_samples_leaf=50, min_samples_split=80, n_estimators=100; total time=  29.6s
[CV] END max_depth=10, min_samples_leaf=50, min_samples_split=80, n_estimators=120; total time=  36.5s
[CV] END max_depth=10, min_samples_leaf=50, min_samples_split=80, n_estimators=120; total time=  36.2s
[CV] END max_depth=10, min_samples_leaf=50, min_samples_split=80, n_estimators=120; total time=  36.1s
```

Out[141… GridSearchCV(cv=3, estimator=GradientBoostingRegressor(),

```
                param_grid={'max_depth': [5, 8, 10], 'min_samples_leaf': [40, 50],
                            'min_samples_split': [50, 80],
                            'n_estimators': [100, 120]},
                scoring='r2', verbose=2)
```

In [143…
```
gb_grid.best_params_
```

Out[143…
```
{'max_depth': 10,
 'min_samples_leaf': 50,
 'min_samples_split': 50,
 'n_estimators': 120}
```

In [144…
```
gb_grid.best_estimator_
```

Out[144…
```
GradientBoostingRegressor(max_depth=10, min_samples_leaf=50,
                          min_samples_split=50, n_estimators=120)
```

In [145…
```
gb_optimal_model = gb_grid.best_estimator_
```

In [146…
```
y_preds_gb = gb_optimal_model.predict(X_test)
y_pred_gb_train=gb_optimal_model.predict(X_train)
```

In [147…
```
EvaluationMetric(X_train,y_train,y_pred_gb_train)
```

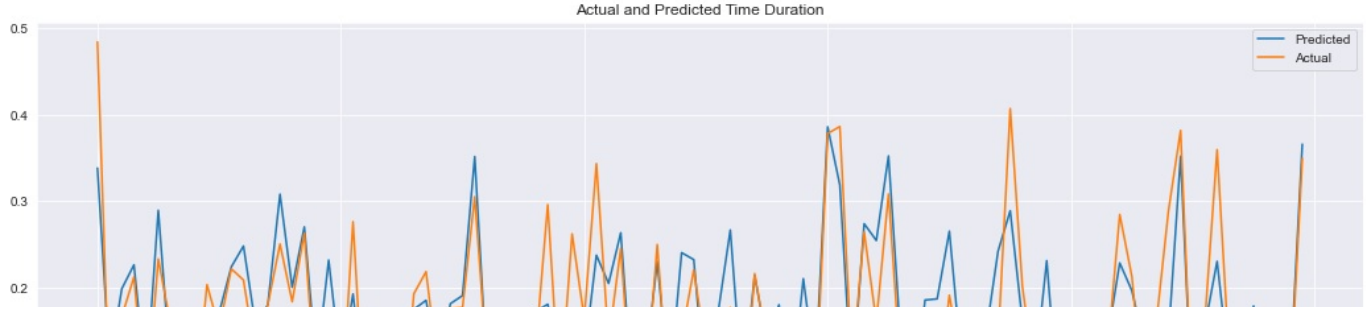
```
MSE : 0.0025 RMSE : 0.05
R2 : 0.7729965302174369 Adjusted R2 : 0.7729426034116621
```

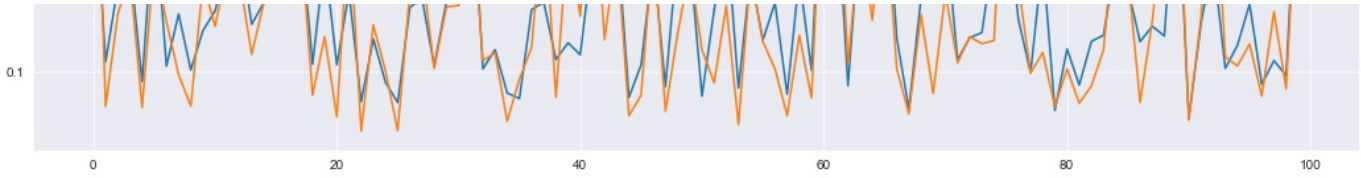
Out[147…
```
(0.0025, 0.05, 0.7729965302174369, 0.7729426034116621)
```

In [148…
```
EvaluationMetric(X_test,y_test,y_preds_gb)
```

```
MSE : 0.0034 RMSE : 0.058309518948453
R2 : 0.694631054812986 Adjusted R2 : 0.6943406639241696
```

Out[148…
```
(0.0034, 0.058309518948453, 0.694631054812986, 0.6943406639241696)
```





In [149…
```
xgb_optimal_model.feature_importances_
```

Out[149…
```
array([0.0082792 , 0.00720963, 0.02233937, 0.02442116, 0.02776379,
       0.04846492, 0.5192163 , 0.02155587, 0.08768707, 0.01112846,
       0.01028003, 0.        , 0.01661176, 0.01953007, 0.03899084,
       0.05096768, 0.0324048 , 0.01794679, 0.03520219], dtype=float32)
```

In [150…
```
importances = gb_optimal_model.feature_importances_

importance_dict = {'Feature' : list(X_train.columns),
                   'Feature Importance' : importances}

importance_df = pd.DataFrame(importance_dict)
```

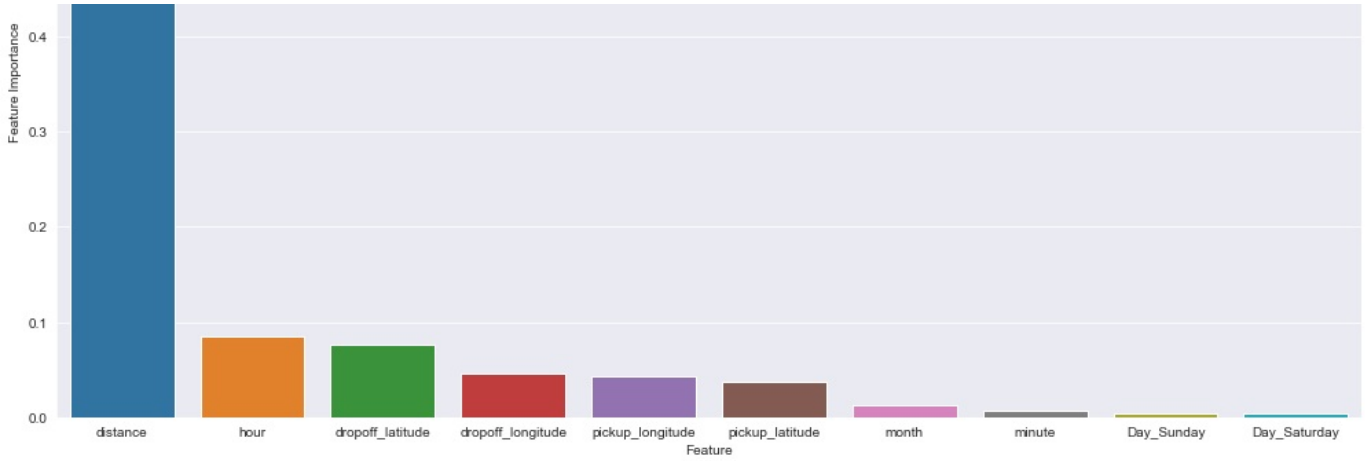
In [151…
```
importance_df.sort_values(by=['Feature Importance'],ascending=False,inplace=True)
importance_df
```

Out[151…

	Feature	Feature Importance
6	distance	0.672464
8	hour	0.085153
5	dropoff_latitude	0.076206
4	dropoff_longitude	0.045665
2	pickup_longitude	0.042593
3	pickup_latitude	0.037061
7	month	0.012195
9	minute	0.007431
15	Day_Sunday	0.004618
14	Day_Saturday	0.004350
18	Day_Wednesday	0.003526
16	Day_Thursday	0.002219
0	vendor_id	0.002059
13	Day_Monday	0.001602
17	Day_Tuesday	0.001351
12	Day_Friday	0.000851
1	passenger_count	0.000655
11	store_and_fwd_flag_Y	0.000000
10	store_and_fwd_flag_N	0.000000

In [152…
```
plt.figure(figsize=(17,9))
plt.title('Top 5 Features')
sns.barplot(x='Feature',y="Feature Importance",data=importance_df[:10])
```

Out[152…
```
<AxesSubplot:title={'center':'Top 5 Features'}, xlabel='Feature', ylabel='Feature Importance'>
```



Light GBM

In [154…

```
!pip install lightgbm
from lightgbm import LGBMRegressor
```

```
Collecting lightgbm
  Downloading lightgbm-4.0.0-py3-none-win_amd64.whl (1.3 MB)
Requirement already satisfied: numpy in c:\users\meghna\anaconda3\lib\site-packages (from lightgbm) (1.20.1)
Requirement already satisfied: scipy in c:\users\meghna\anaconda3\lib\site-packages (from lightgbm) (1.6.2)
Installing collected packages: lightgbm
Successfully installed lightgbm-4.0.0
```

In [155…

```
n_estimator=[5,10,20] # No. of tree
max_depth=[5,7,9] # max depth of tree
min_samples_split=[40,50]
params={"n_estimator":n_estimator,"max_depth":max_depth,"min_samples_split":min_samples_split}
lgb=LGBMRegressor()
gs_lgb=GridSearchCV(lgb,params,cv=3,verbose=2,scoring='r2')
gs_lgb.fit(X_train,y_train)
print(gs_lgb.best_score_)
print(gs_lgb.best_params_)
```

```
Fitting 3 folds for each of 18 candidates, totalling 54 fits
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002274 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=5, min_samples_split=40, n_estimator=5; total time=   0.3s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002122 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=5, min_samples_split=40, n_estimator=5; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
```

```
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001997 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=5, min_samples_split=40, n_estimator=5; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002046 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=5, min_samples_split=40, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002387 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
```

```
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=5, min_samples_split=40, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001784 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=5, min_samples_split=40, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002061 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=5, min_samples_split=40, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002060 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=5, min_samples_split=40, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002160 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=5, min_samples_split=40, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002072 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
```

```
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=5, min_samples_split=50, n_estimator=5; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001831 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=5, min_samples_split=50, n_estimator=5; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001643 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=5, min_samples_split=50, n_estimator=5; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
```

```
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002105 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=5, min_samples_split=50, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002040 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=5, min_samples_split=50, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002302 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
```

```
(num_leaves=31).
[CV] END ..max_depth=5, min_samples_split=50, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002168 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=5, min_samples_split=50, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002211 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=5, min_samples_split=50, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002210 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=5, min_samples_split=50, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002272 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=7, min_samples_split=40, n_estimator=5; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002129 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=7, min_samples_split=40, n_estimator=5; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002314 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=7, min_samples_split=40, n_estimator=5; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001927 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
```

```
(num_leaves=31).
[CV] END ..max_depth=7, min_samples_split=40, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002096 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=7, min_samples_split=40, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001879 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=7, min_samples_split=40, n_estimator=10; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002185 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=7, min_samples_split=40, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001983 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=7, min_samples_split=40, n_estimator=20; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
```

```
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002131 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=7, min_samples_split=40, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002387 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=7, min_samples_split=50, n_estimator=5; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002353 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=7, min_samples_split=50, n_estimator=5; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002197 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=7, min_samples_split=50, n_estimator=5; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002161 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
```

```
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=7, min_samples_split=50, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002151 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=7, min_samples_split=50, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002095 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=7, min_samples_split=50, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002102 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=7, min_samples_split=50, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002072 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=7, min_samples_split=50, n_estimator=20; total time=   0.1s
```

```
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002130 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=7, min_samples_split=50, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002113 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=9, min_samples_split=40, n_estimator=5; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002188 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=9, min_samples_split=40, n_estimator=5; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing col-wise multi-threading, the overhead of testing was 0.003477 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=9, min_samples_split=40, n_estimator=5; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
```

```
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002105 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=9, min_samples_split=40, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002075 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=9, min_samples_split=40, n_estimator=10; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing col-wise multi-threading, the overhead of testing was 0.004634 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=9, min_samples_split=40, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001681 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=9, min_samples_split=40, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001596 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
```

```
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=9, min_samples_split=40, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002149 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=9, min_samples_split=40, n_estimator=20; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001954 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=9, min_samples_split=50, n_estimator=5; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001569 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=9, min_samples_split=50, n_estimator=5; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002062 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ...max_depth=9, min_samples_split=50, n_estimator=5; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
```

```
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002152 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=9, min_samples_split=50, n_estimator=10; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001373 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=9, min_samples_split=50, n_estimator=10; total time=   0.2s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002181 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=9, min_samples_split=50, n_estimator=10; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001736 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183445
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=9, min_samples_split=50, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing col-wise multi-threading, the overhead of testing was 0.004703 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1408
```

```
[LightGBM] [Info] Number of data points in the train set: 53333, number of used features: 19
[LightGBM] [Info] Start training from score 0.183358
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=9, min_samples_split=50, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002235 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 53334, number of used features: 19
[LightGBM] [Info] Start training from score 0.183103
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[CV] END ..max_depth=9, min_samples_split=50, n_estimator=20; total time=   0.1s
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002668 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1408
[LightGBM] [Info] Number of data points in the train set: 80000, number of used features: 19
[LightGBM] [Info] Start training from score 0.183302
0.6592905740259131
{'max_depth': 9, 'min_samples_split': 40, 'n_estimator': 5}
```

In [156…
```
gs_lgb.best_estimator_
```

Out[156… `LGBMRegressor(max_depth=9, min_samples_split=40, n_estimator=5)`

In [157…
```
gs_lgb_opt_model = gs_lgb.best_estimator_
```

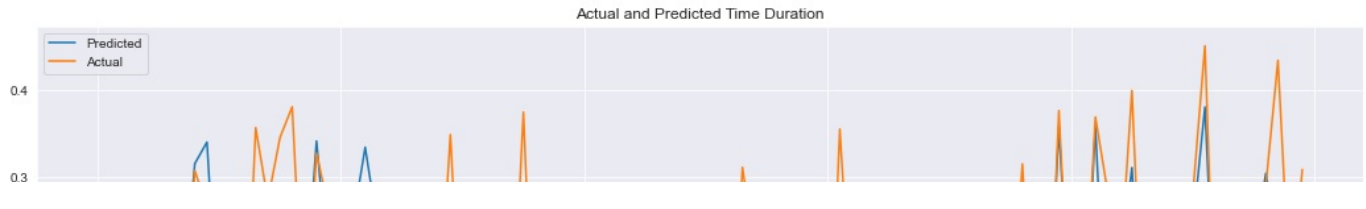
In [158…
```
y_preds_lgb = gs_lgb_opt_model.predict(X_test)
y_pred_lgb_train=gs_lgb_opt_model.predict(X_train)
```

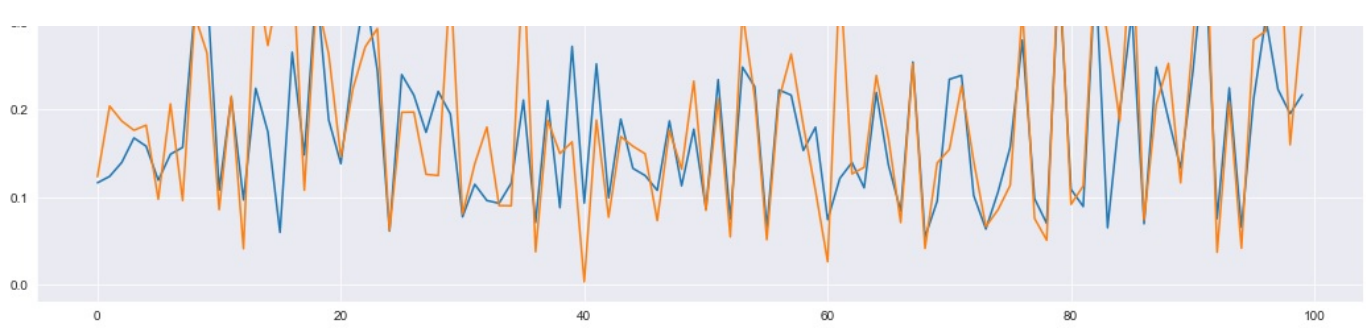
```
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
[LightGBM] [Warning] Unknown parameter: min_samples_split
[LightGBM] [Warning] Unknown parameter: n_estimator
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves.
(num_leaves=31).
```

In [159…
```
EvaluationMetric(X_train,y_train,y_pred_lgb_train)
```

```
MSE : 0.0035 RMSE : 0.05916079783099616
R2 : 0.6814194325467515 Adjusted R2 : 0.6813437507415301
```

Out[159… `(0.0035, 0.05916079783099616, 0.6814194325467515, 0.6813437507415301)`

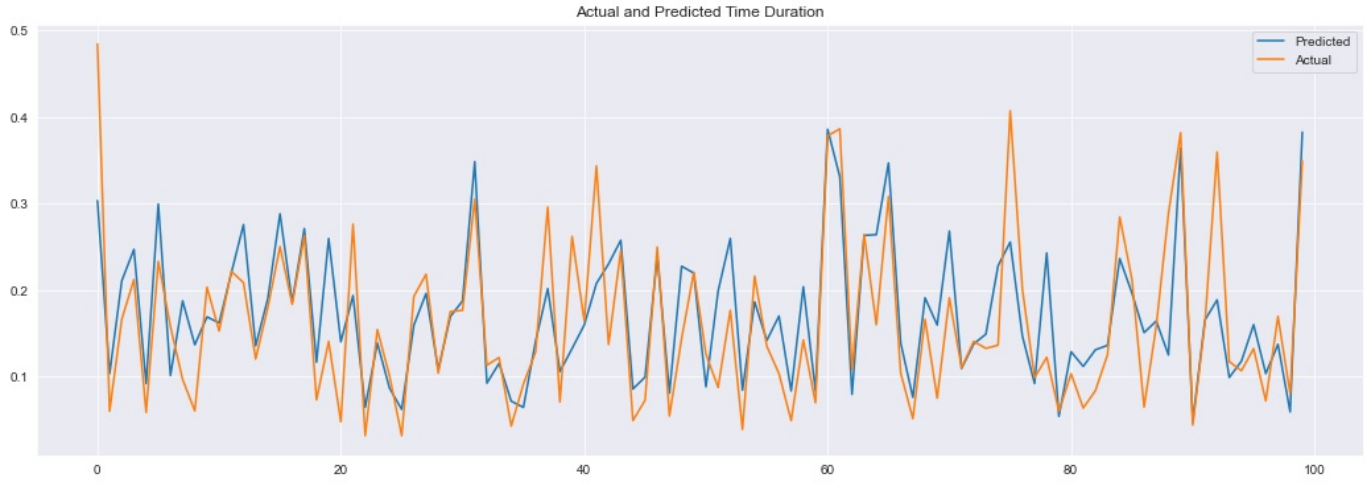




In [160…
```
EvaluationMetric(X_test,y_test,y_preds_lgb)
```

```
MSE : 0.0036 RMSE : 0.06
R2 : 0.6715585072867248 Adjusted R2 : 0.6712461755368975
```

Out[160…
```
(0.0036, 0.06, 0.6715585072867248, 0.6712461755368975)
```



In [161…
```
importances = gs_lgb_opt_model.feature_importances_

importance_dict = {'Feature' : list(X_train.columns),
                   'Feature Importance' : importances}

importance_df = pd.DataFrame(importance_dict)
```

In [162…
```
importance_df.sort_values(by=['Feature Importance'],ascending=False,inplace=True)
importance_df
```

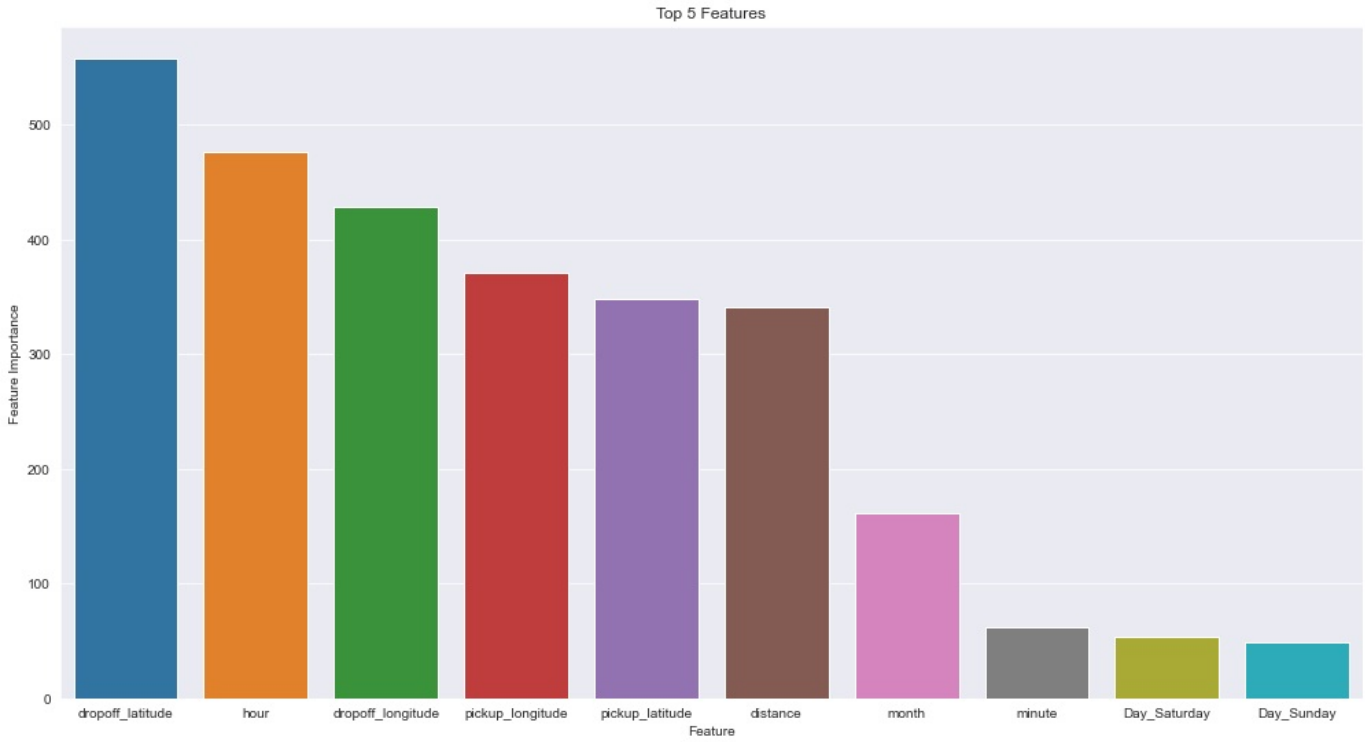
Out[162…

	Feature	Feature Importance
5	dropoff_latitude	558
8	hour	476
4	dropoff_longitude	429
2	pickup_longitude	371
3	pickup_latitude	348
6	distance	341
7	month	162
9	minute	62
14	Day_Saturday	54
15	Day_Sunday	49
18	Day_Wednesday	29
0	vendor_id	28
13	Day_Monday	26
16	Day_Thursday	26
12	Day_Friday	16

17	Day_Tuesday	13
1	passenger_count	12
11	store_and_fwd_flag_Y	0
10	store_and_fwd_flag_N	0

In [163…
```
plt.figure(figsize=(17,9))
plt.title('Top 5 Features')
sns.barplot(x='Feature',y="Feature Importance",data=importance_df[:10])
```

Out[163… <AxesSubplot:title={'center':'Top 5 Features'}, xlabel='Feature', ylabel='Feature Importance'>



In [165…
```
!pip install prettytable
```

```
Collecting prettytable
  Downloading prettytable-3.8.0-py3-none-any.whl (27 kB)
Requirement already satisfied: wcwidth in c:\users\meghna\anaconda3\lib\site-packages (from prettytable) (0.2.5)
Installing collected packages: prettytable
Successfully installed prettytable-3.8.0
```

In [166…
```
from prettytable import PrettyTable
```

In [167…
```
from prettytable import PrettyTable
train = PrettyTable(['SL NO',"MODEL_NAME", "Train MSE", "Train RMSE",'Train R^2','Train Adjusted R^2'])
train.add_row(['1','Linear Regression','0.0055473742826857575', '0.07448069738318619', '0.4975386610902135','0.49
train.add_row(['2','Lasso Regression','0.005544777800761673','0.07446326477372364','0.4977738411443343','0.497654
train.add_row(['3','Ridge Regression ','0.0055447795071197434','0.07446327623144004','0.49777368658851806','0.497
train.add_row(['4','DecisionTree Regressor','0.003933693284929908','0.0627191620234989','0.6437001193563989','0.6
train.add_row(['5','XGBRegressor','0.001963513255289278','0.044311547651704496','0.8221519859766687','0.82210973
train.add_row(['6','GradientBoosting','0.0022675373651197994','0.04761866614175369','0.7946145685424261','0.79456
train.add_row(['7','LightGBM','0.0032980298940813377','0.057428476334318135','0.7012762377478683','0.701205273113
print(train)
```

```
+-------+----------------------+----------------------+---------------------+---------------------+----------
-----------+
| SL NO |       MODEL_NAME     |       Train MSE      |      Train RMSE     |      Train R^2      |  Train Ad
justed R^2 |
+-------+----------------------+----------------------+---------------------+---------------------+----------
-----------+
|   1   |   Linear Regression  | 0.0055473742826857575 | 0.07448069738318619 |  0.4975386610902135 | 0.4974192
9668112017 |
|   2   |    Lasso Regression  |  0.005544777800761673 | 0.07446326477372364 |  0.4977738411443343 |  0.497654
5326044711 |
```

```
|   3   |   Ridge Regression     | 0.0055447795071197434 | 0.07446327623144004  | 0.49777368658851806 |  0.497654
3780119387 |
|   4   | DecisionTree Regressor |  0.003933693284929908 |  0.0627191620234989  |  0.6437001193563989 |  0.643615
4769741506 |
|   5   |      XGBRegressor      |  0.001963513255289278 | 0.044311547651704496 |  0.8221519859766687 |  0.822109
7365109717 |
|   6   |    GradientBoosting    | 0.0022675373651197994 | 0.04761866614175369  |  0.7946145685424261 |  0.794565
7773046455 |
|   7   |        LightGBM        | 0.0032980298940813377 | 0.057428476334318135 |  0.7012762377478683 |  0.701205
2731131748 |
+-------+------------------------+-----------------------+----------------------+---------------------+---------
-----------+
```

In [168…

```
from prettytable import PrettyTable
test = PrettyTable(['SL NO',"MODEL_NAME", "Test MSE", "Test RMSE",'Test R^2','Test Adjusted R^2'])
test.add_row(['1','Linear Regression','0.0054727657060915 76','0.07397814343501448','0.4957919161505717','0.495312
test.add_row(['2','Lasso Regression','0.0054708980396948005','0.07396551926198315','0.49596398499944283','0.49548
test.add_row(['3','Ridge Regression ','0.005470874962507452','0.07396536326218815','0.49596611110990296','0.49548
test.add_row(['4','DecisionTree Regressor','0.004235598377226902','0.0650814749158845','0.609772634819681','0.609
test.add_row(['5','XGBRegressor','0.0031445622329224813','0.056076396397436966','0.7102902292633624','0.710014729
test.add_row(['6','GradientBoosting','0.003121489155427573','0.05587028866425851','0.7124159610810551',' 0.712142
test.add_row(['7','LightGBM','0.003418832582409957',' 0.058470784007142895','0.6850215927460219','0.684722063730
print(test)
```

```
+-------+------------------------+-----------------------+----------------------+---------------------+---------
-----------+
| SL NO |       MODEL_NAME       |        Test MSE       |       Test RMSE      |       Test R^2      |  Test Ad
justed R^2  |
+-------+------------------------+-----------------------+----------------------+---------------------+---------
-----------+
|   1   |   Linear Regression    |  0.005472765706091576 |  0.07397814343501448 |  0.4957919161505717 |  0.49531
2438993758  |
|   2   |    Lasso Regression    | 0.0054708980396948005 |  0.07396551926198315 | 0.49596398499944283 | 0.495484
67147166453 |
|   3   |   Ridge Regression     |  0.005470874962507452 |  0.07396536326218815 | 0.49596611110990296 |  0.49548
67996039515 |
|   4   | DecisionTree Regressor |  0.004235598377226902 |   0.0650814749158845 |  0.609772634819681  |  0.60940
15477356758 |
|   5   |      XGBRegressor      | 0.0031445622329224813 |  0.056076396397436966 |  0.7102902292633624 |  0.71001
47294813806 |
|   6   |    GradientBoosting    |  0.003121489155427573 |  0.05587028866425851  |  0.7124159610810551 |  0.71214
24827657667 |
|   7   |        LightGBM        |  0.003418832582409957 |  0.058470784007142895 |  0.6850215927460219 |  0.68472
20637301148 |
+-------+------------------------+-----------------------+----------------------+---------------------+---------
-----------+
```

We conclude that Gradient Boosting perform well.

Conclusion

Objective:

- We observe that both models show somewhat similar learning rate but with visible differences in error rates.
- Gradient boosting performed very well out of all the models
- XGBoost training curve on the other hand starts quite low and further improves with the increase in the training size and it too plateau towards the end.
- Validation curve seems to show similar trend in both the models i.e. starts very high but improves with the training size with some differences in error rate i.e. XGBoost curve learning is quite fast and more accurate as compared to the RF one.
- Both the models seems to suffer from high variance since the training curve error is very less in both the models.
- The large gap at the end also indicates that the model suffers from quite a low bias i.e. overfitting the training data.
- Also, both the model's still has potential to decrease and converge towards the training curve by the end.

End Notes:

We discussed a variety of machine learning development cycle topics in this project. We found that the exploration of the data and variable analysis is a crucial part of the process and should be carried out for a complete knowledge of the data. While exploring, we also cleansed

the data because some outliers needed to be dealt with before feature engineering. Additionally, we used feature engineering to filter out and collect only the best features—those that are more important and accounted for the majority of the variance in the dataset. Finally, we trained the models using the featureset that produced the best results.

Improvement:

- Add more training instances to improve validation curve in the XGBoost model.
- Increase the regularization for the learning algorithm. This should decrease the variance and increase the bias towards the validation curve.
- Reduce the numbers of features in the training data that we currently use. The algorithm will still fit the training data very well, but due to the decreased number of features, it will build less complex models. This should increase the bias and decrease the variance.