In [1]:

```python
import pandas as pd  #pandas for using dataframe and reading csv
import numpy as np   #numpy for vector operations and basic maths
#import simplejson    #getting JSON in simplified format
import urllib       #for url stuff
#import gmaps      #for using google maps to visulalize places on maps
import re           #for processing regular expressions
import datetime      #for datetime operations
import calendar      #for calendar for datetime operations
import time         #to get the system time
import scipy        #for other dependancies
from sklearn.cluster import KMeans # for doing K-means clustering
from haversine import haversine # for calculating haversine distance
import math         #for basic maths operations
import seaborn as sns #for making plots
import matplotlib.pyplot as plt # for plotting
import os  # for os commands
#from scipy.misc import imread, imresize, imsave  # for plots
import chart_studio.plotly as py
import plotly.graph_objs as go
import plotly
from bokeh.palettes import Spectral4
from bokeh.plotting import figure, output_notebook, show
from IPython.display import HTML
from matplotlib.pyplot import *
from matplotlib import cm
from matplotlib import animation
import io
import base64
output_notebook()
plotly.offline.init_notebook_mode() # run at the start of every ipython notebook
```

(https://bokeh.pydata.org)BokehJS 1.3.4 successfully loaded.

In [2]:

```python
from sklearn.metrics import r2_score
from scipy.stats.stats import pearsonr
from sklearn.metrics import mean_squared_error as mse
import lightgbm as lgb
```

In [3]:

```
train_fr_1 = pd.read_csv('fastest_routes_train_part_1.csv')
train_fr_2 = pd.read_csv('fastest_routes_train_part_2.csv')
train_fr = pd.concat([train_fr_1, train_fr_2])
train_fr.head()
```

Out[3]:

	id	starting_street	end_street	total_distance	total_travel_time	number_of_steps
0	id2875421	Columbus Circle	East 65th Street	2009.1	164.9	5
1	id2377394	2nd Avenue	Washington Square West	2513.2	332.0	6
2	id3504673	Greenwich Street	Broadway	1779.4	235.8	4
3	id2181028	Broadway	West 81st Street	1614.9	140.1	5
4	id0801584	Lexington Avenue	West 31st Street	1393.5	189.4	5

In [4]:

```
train_fr.shape
```

Out[4]:

```
(1458643, 12)
```

In [6]:

```
train_df = pd.read_csv('train.csv')
train_df.head()
```

Out[6]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitu
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.9821
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.9804
2	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.9790
3	id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.0100
4	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.9730

In [7]:

```
train_df.shape
```

Out[7]:

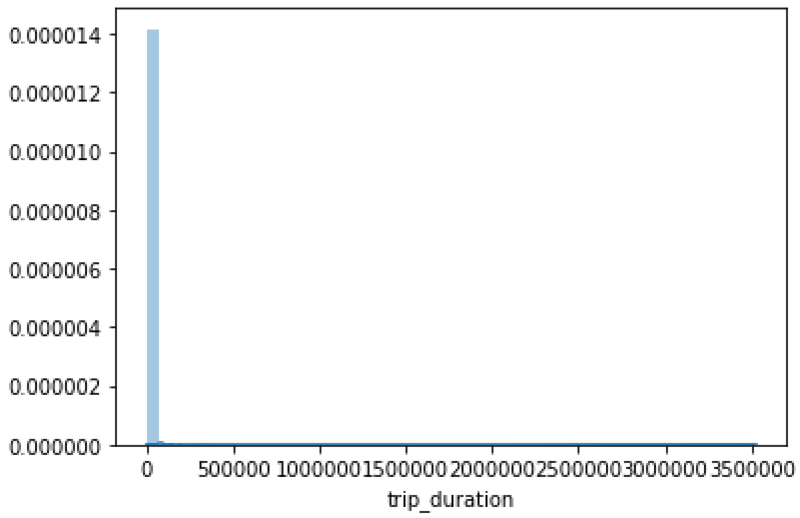
```
(1458644, 11)
```

In [8]:

```
sns.distplot(train_df['trip_duration'])
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1eecfdf7cc8>
```

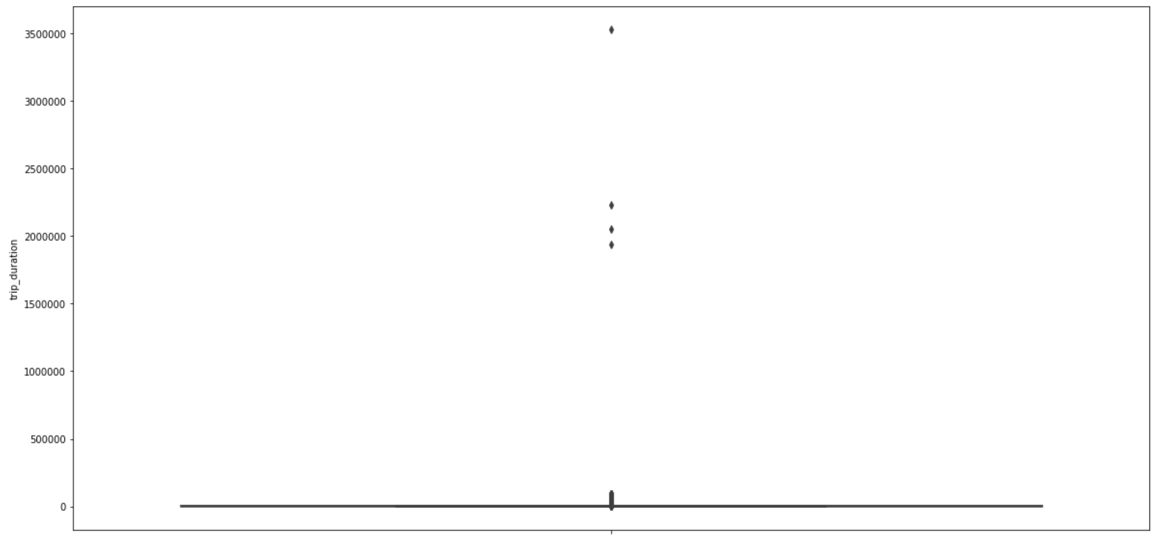


In [9]:

```
plt.figure(figsize = (20,10))
sns.boxplot(data = train_df, y='trip_duration')
```

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ef06198e88>
```



In [10]:

```
np.mean(train_df['trip_duration'])
```

Out[10]:

```
959.4922729603659
```

In [11]:

```
np.max(train_df['trip_duration'])/3600
```

Out[11]:

```
979.5227777777778
```

In [12]:

```
np.mean(train_df[train_df['trip_duration'] <= 3600*3]['trip_duration'])
```

Out[12]:

```
836.8008674028446
```

In [13]:

```
train_df = train_df[train_df['trip_duration'] <= 3600*3]
```

In [15]:

```
print(train_df['pickup_latitude'].max())
print(train_df['pickup_latitude'].min())
print(train_df['pickup_longitude'].max())
print(train_df['pickup_longitude'].min())
```

```
51.88108444213867
34.359695434570305
-61.33552932739258
-121.93334197998048
```

In [16]:

```
df = train_df.loc[(train_df.pickup_latitude > 40.4772) & (train_df.pickup_latitude < 45.0153)]
df = df.loc[(df.dropoff_latitude>40.4772) & (df.dropoff_latitude < 45.0153)]
df = df.loc[(df.dropoff_longitude > -79.7624) & (df.dropoff_longitude < -71.7517)]
df = df.loc[(df.pickup_longitude > -79.7624) & (df.pickup_longitude < -71.7517)]
```

In [17]:

```
df['trip_duration'].mean()
```

Out[17]:

```
836.7714700022108
```

In [18]:

```
df.head()
```

Out[18]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitu
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.9821
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.9804
2	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.9790
3	id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.0100
4	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.9730

In [19]:

```
df.shape
```

Out[19]:

```
(1456474, 11)
```

In [20]:

```
train_fr_new = train_fr[['id', 'total_distance', 'total_travel_time']]
```

In [21]:

```
df = pd.merge(df, train_fr_new, on = 'id', how = 'left')
```

In [22]:

```
df = df.dropna(subset=['total_distance'])
```

In [23]:

```
df = df[df['total_distance'] != 0]
```

In [25]:

```
df.shape
```

Out[25]:

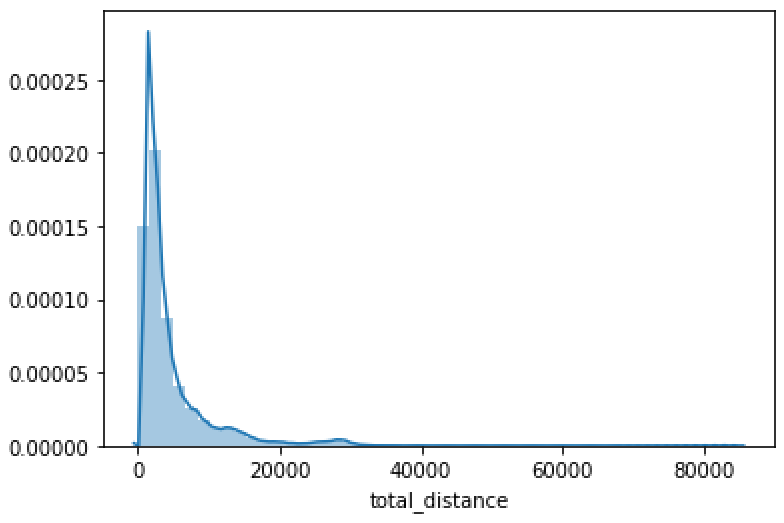
```
(1450497, 13)
```

In [24]:

```
sns.distplot(df['total_distance'])
```

Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ee9a5be248>
```



In [26]:

```
df[df['trip_duration'] == df['trip_duration'].max()]
```

Out[26]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_l
1017225	id3661441	2	2016-06-27 21:01:09	2016-06-28 00:00:00	5	-7

In [27]:

```
df[df['total_distance'] == df['total_distance'].max()]
```

Out[27]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_l
1407369	id1390461	2	2016-06-20 11:27:37	2016-06-20 12:59:20	1	-7

In [28]:

```
#Very simple dataset with only 3 features.
df1 = df[['vendor_id','passenger_count','trip_duration','total_distance']]
```

In [29]:

```
df1 = pd.get_dummies(df1,columns=['vendor_id'])
```

In [30]:

```
y = df1['trip_duration']
X = df1.drop('trip_duration',axis = 1)
```

In [31]:

```
from sklearn.model_selection import train_test_split
```

In [32]:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
```

In [33]:

```
from sklearn.linear_model import LinearRegression
```

In [34]:

```
reg = LinearRegression().fit(X_train.total_distance.values.reshape(-1, 1), y_train)
```

In [35]:

```
print(reg.intercept_)
print(reg.coef_)
```

```
390.50495784922003
[0.09645902]
```

In [36]:

```
y_pred = reg.predict(X_test.total_distance.values.reshape(-1, 1))
```

In [37]:

```
r2_score(y_test, y_pred)
```

Out[37]:

```
0.60313440149766
```

In [38]:

```
np.sqrt(mse(y_test,y_pred))
```

Out[38]:

```
414.8816493534373
```

In [39]:

```
import xgboost as xgb
```

In [40]:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
Xtr, Xv, ytr, yv = train_test_split(X_train, y_train, test_size=0.2)
```

In [47]:

```
dtrain = xgb.DMatrix(Xtr, label=ytr)
dvalid = xgb.DMatrix(Xv, label=yv)
dtest = xgb.DMatrix(X_test)
watchlist = [(dtrain, 'train'), (dvalid, 'valid')]

xgb_pars = {'min_child_weight': 50, 'eta': 0.1, 'colsample_bytree': 0.3, 'max_depth': 10,
            'subsample': 0.8, 'lambda': 1., 'nthread': -1, 'booster' : 'gbtree', 'silent': 1,
            'eval_metric': 'rmse', 'objective': 'reg:linear'}

# You could try to train with more epoch
model = xgb.train(xgb_pars, dtrain, 50, watchlist, early_stopping_rounds=2,
                  maximize=False, verbose_eval=1)
print('Modeling RMSLE %.5f' % model.best_score)
```

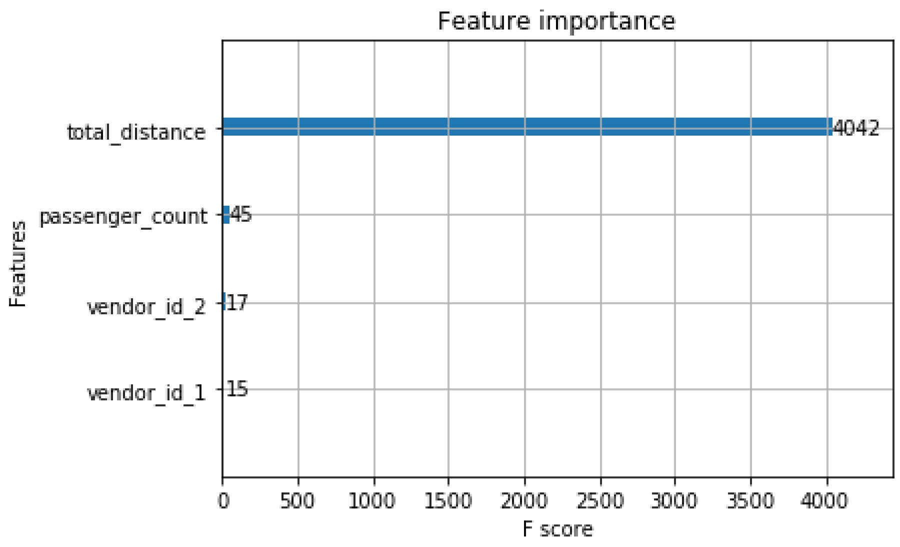
```
[0]	train-rmse:999.50867	valid-rmse:1001.66260
Multiple eval metrics have been passed: 'valid-rmse' will be used for early stoppi
ng.

Will train until valid-rmse hasn't improved in 2 rounds.
[1]	train-rmse:944.01953	valid-rmse:946.32098
[2]	train-rmse:896.51898	valid-rmse:898.96985
[3]	train-rmse:856.15619	valid-rmse:858.73456
[4]	train-rmse:821.97101	valid-rmse:824.67407
[5]	train-rmse:793.23712	valid-rmse:796.04657
[6]	train-rmse:734.16864	valid-rmse:737.00885
[7]	train-rmse:713.10889	valid-rmse:716.04218
[8]	train-rmse:695.55310	valid-rmse:698.57105
[9]	train-rmse:681.02258	valid-rmse:684.11157
[10]	train-rmse:669.04413	valid-rmse:672.19617
[11]	train-rmse:659.17200	valid-rmse:662.37640
[12]	train-rmse:617.55011	valid-rmse:620.81342
[13]	train-rmse:610.52869	valid-rmse:613.83350
[14]	train-rmse:604.81232	valid-rmse:608.15051
[15]	train-rmse:600.09467	valid-rmse:603.46350
[16]	train-rmse:596.28003	valid-rmse:599.67175
[17]	train-rmse:593.14343	valid-rmse:596.55621
[18]	train-rmse:590.60449	valid-rmse:594.03418
[19]	train-rmse:588.55072	valid-rmse:591.99365
[20]	train-rmse:586.87573	valid-rmse:590.33252
[21]	train-rmse:585.51379	valid-rmse:588.97986
[22]	train-rmse:584.41174	valid-rmse:587.88446
[23]	train-rmse:583.51331	valid-rmse:586.99249
[24]	train-rmse:582.78265	valid-rmse:586.26770
[25]	train-rmse:582.19159	valid-rmse:585.68225
[26]	train-rmse:581.69525	valid-rmse:585.19098
[27]	train-rmse:550.89972	valid-rmse:554.44995
[28]	train-rmse:550.55591	valid-rmse:554.10767
[29]	train-rmse:524.27533	valid-rmse:527.90832
[30]	train-rmse:524.04681	valid-rmse:527.68109
[31]	train-rmse:523.86273	valid-rmse:527.49884
[32]	train-rmse:523.71179	valid-rmse:527.34949
[33]	train-rmse:523.59216	valid-rmse:527.23029
[34]	train-rmse:501.35709	valid-rmse:505.05048
[35]	train-rmse:501.27445	valid-rmse:504.96942
[36]	train-rmse:501.20740	valid-rmse:504.90262
[37]	train-rmse:482.43521	valid-rmse:486.23086
[38]	train-rmse:482.38962	valid-rmse:486.18433
[39]	train-rmse:482.35080	valid-rmse:486.14645
[40]	train-rmse:482.31934	valid-rmse:486.11591
[41]	train-rmse:482.28976	valid-rmse:486.08664
[42]	train-rmse:482.27060	valid-rmse:486.06607
[43]	train-rmse:482.25430	valid-rmse:486.05057
[44]	train-rmse:466.53388	valid-rmse:470.37842
[45]	train-rmse:466.52292	valid-rmse:470.36719
[46]	train-rmse:466.51205	valid-rmse:470.35687
[47]	train-rmse:453.38126	valid-rmse:457.28717
[48]	train-rmse:453.37488	valid-rmse:457.28213
[49]	train-rmse:453.36990	valid-rmse:457.27716
Modeling RMSLE 457.27716
```

In [48]:

```
xgb.plot_importance(model)
```

Out[48]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ee9a6cf1c8>
```



In [49]:

```
y_pred = model.predict(dtest)
```

In [50]:

```
print(y_pred.shape == y_test.shape)
```

```
True
```

In [51]:

```
r2_score(y_test, y_pred)
```

Out[51]:

```
0.5210643821337254
```

In [52]:

```
np.sqrt(mse(y_test,y_pred))
```

Out[52]:

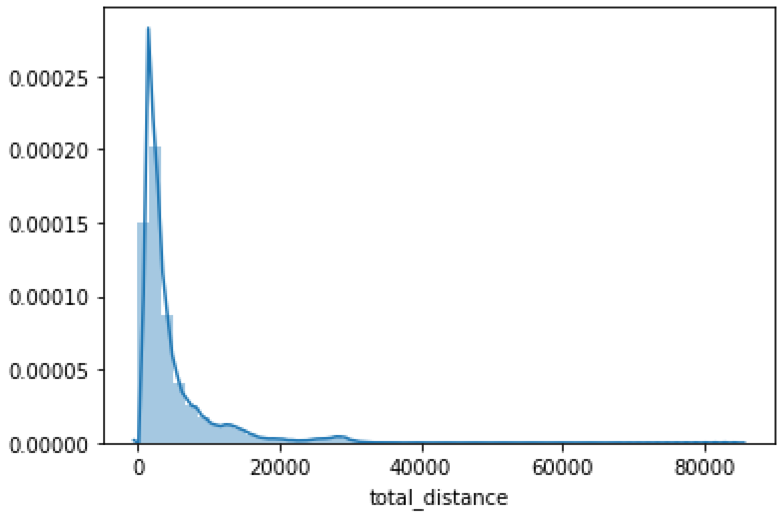
```
457.6701339507657
```

In [53]:

```
#The data is very skewed, what if I take the log transformation for it?
sns.distplot(X.total_distance)
```

Out[53]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ee9a727208>
```



In [394]:

```
df2 = df1.copy()
```

In [395]:

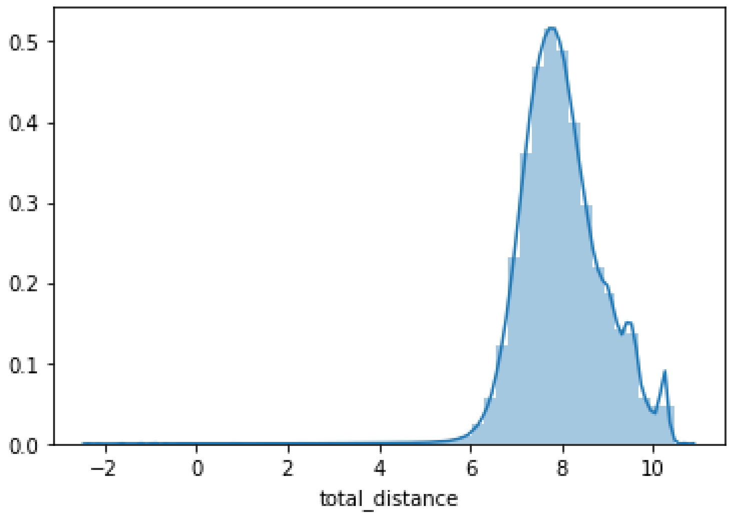
```
df2['total_distance'] = np.log(df2['total_distance'])
df2['trip_duration'] = np.log(df2['trip_duration'])
```

In [396]:

```
sns.distplot(df2['total_distance'])
```

Out[396]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2497010d448>
```



In [397]:

```
df2.head()
```

Out[397]:

	passenger_count	trip_duration	total_distance	vendor_id_1	vendor_id_2
0	1	6.120297	7.605442	0	1
1	1	6.496775	7.829312	1	0
2	1	7.661056	9.311163	0	1
3	1	6.061457	7.484032	0	1
4	1	6.075346	7.387028	0	1

In [398]:

```
y = df2['trip_duration']
X = df2.drop('trip_duration',axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
```

In [399]:

```
reg = LinearRegression().fit(X_train.total_distance.values.reshape(-1, 1), y_train)
```

In [400]:

```
print(reg.intercept_)
print(reg.coef_)
```

```
1.1511701415762072
[0.66308148]
```

In [401]:

```
y_pred = reg.predict(X_test.total_distance.values.reshape(-1, 1))
y_pred = np.exp(y_pred)
y_test = np.exp(y_test)
r2_score(y_test, y_pred)
```

Out[401]:

```
0.6210462024301806
```

In [402]:

```
np.sqrt(mse(y_test,y_pred))
```

Out[402]:

```
401.18238961113883
```

In [413]:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
Xtr, Xv, ytr, yv = train_test_split(X_train, y_train, test_size=0.2)
```

In [414]:

```
dtrain = xgb.DMatrix(Xtr, label=ytr)
dvalid = xgb.DMatrix(Xv, label=yv)
dtest = xgb.DMatrix(X_test)
watchlist = [(dtrain, 'train'), (dvalid, 'valid')]

xgb_pars = {'min_child_weight': 50, 'eta': 0.2, 'colsample_bytree': 0.3, 'max_depth': 10,
            'subsample': 0.8, 'lambda': 1., 'nthread': -1, 'booster' : 'gbtree', 'silent': 1,
            'eval_metric': 'rmse', 'objective': 'reg:linear'}

# You could try to train with more epoch
model = xgb.train(xgb_pars, dtrain, 50, watchlist, early_stopping_rounds=2,
                  maximize=False, verbose_eval=1)
print('Modeling RMSLE %.5f' % model.best_score)
```

```
[0]	train-rmse:4.82752	valid-rmse:4.82709
Multiple eval metrics have been passed: 'valid-rmse' will be used for early stoppi
ng.

Will train until valid-rmse hasn't improved in 2 rounds.
[1]	train-rmse:3.88906	valid-rmse:3.88861
[2]	train-rmse:3.14456	valid-rmse:3.14419
[3]	train-rmse:2.55678	valid-rmse:2.55643
[4]	train-rmse:2.09576	valid-rmse:2.09549
[5]	train-rmse:1.73772	valid-rmse:1.73752
[6]	train-rmse:1.46337	valid-rmse:1.46324
[7]	train-rmse:1.25666	valid-rmse:1.25664
[8]	train-rmse:1.10417	valid-rmse:1.10424
[9]	train-rmse:0.99450	valid-rmse:0.99467
[10]	train-rmse:0.83810	valid-rmse:0.83820
[11]	train-rmse:0.77978	valid-rmse:0.77998
[12]	train-rmse:0.74007	valid-rmse:0.74035
[13]	train-rmse:0.71350	valid-rmse:0.71384
[14]	train-rmse:0.69593	valid-rmse:0.69633
[15]	train-rmse:0.68448	valid-rmse:0.68492
[16]	train-rmse:0.67706	valid-rmse:0.67754
[17]	train-rmse:0.60235	valid-rmse:0.60281
[18]	train-rmse:0.59891	valid-rmse:0.59940
[19]	train-rmse:0.54674	valid-rmse:0.54721
[20]	train-rmse:0.51052	valid-rmse:0.51104
[21]	train-rmse:0.50946	valid-rmse:0.50999
[22]	train-rmse:0.48514	valid-rmse:0.48577
[23]	train-rmse:0.48468	valid-rmse:0.48531
[24]	train-rmse:0.46850	valid-rmse:0.46920
[25]	train-rmse:0.46831	valid-rmse:0.46901
[26]	train-rmse:0.46818	valid-rmse:0.46889
[27]	train-rmse:0.46809	valid-rmse:0.46880
[28]	train-rmse:0.46803	valid-rmse:0.46875
[29]	train-rmse:0.46800	valid-rmse:0.46871
[30]	train-rmse:0.46797	valid-rmse:0.46869
[31]	train-rmse:0.46796	valid-rmse:0.46868
[32]	train-rmse:0.46795	valid-rmse:0.46867
[33]	train-rmse:0.46795	valid-rmse:0.46867
[34]	train-rmse:0.46794	valid-rmse:0.46866
[35]	train-rmse:0.45750	valid-rmse:0.45830
[36]	train-rmse:0.45064	valid-rmse:0.45151
[37]	train-rmse:0.45064	valid-rmse:0.45150
[38]	train-rmse:0.45064	valid-rmse:0.45150
[39]	train-rmse:0.45064	valid-rmse:0.45150
Stopping. Best iteration:
[37]	train-rmse:0.45064	valid-rmse:0.45150

Modeling RMSLE 0.45150
```

In [415]:

```
y_pred = model.predict(dtest)
y_test = np.exp(y_test)
y_pred = np.exp(y_pred)
r2_score(y_test, y_pred)
```

Out[415]:

```
0.5659807276759494
```

In [416]:

```
np.sqrt(mse(y_test,y_pred))
```

Out[416]:

```
430.23334860540683
```

In [377]:

```
df.head()
```

Out[377]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitu
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.9821
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.9804
2	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.9790
3	id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.0100
4	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.9730

In [54]:

```
train_data = df.copy()
train_data['pickup_datetime'] = pd.to_datetime(train_data.pickup_datetime)
train_data.loc[:, 'pick_month'] = train_data['pickup_datetime'].dt.month
train_data.loc[:, 'hour'] = train_data['pickup_datetime'].dt.hour
train_data.loc[:, 'week_of_year'] = train_data['pickup_datetime'].dt.weekofyear
train_data.loc[:, 'day_of_year'] = train_data['pickup_datetime'].dt.dayofyear
train_data.loc[:, 'day_of_week'] = train_data['pickup_datetime'].dt.dayofweek
```

In [55]:

```
train_data.head()
```

Out[55]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitu
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.9821
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.9804
2	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.9790
3	id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.0100
4	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.9730

In [425]:

```
df3 = train_data[['pickup_longitude','dropoff_longitude','vendor_id','passenger_count','trip_dur
ation','total_distance']]
```

In [459]:

```
y = df3['trip_duration']
X = df3.drop('trip_duration',axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
Xtr, Xv, ytr, yv = train_test_split(X_train, y_train, test_size=0.2)
```

In [460]:

```
dtrain = xgb.DMatrix(Xtr, label=ytr)
dvalid = xgb.DMatrix(Xv, label=yv)
dtest = xgb.DMatrix(X_test)
watchlist = [(dtrain, 'train'), (dvalid, 'valid')]

xgb_pars = {'min_child_weight': 50, 'eta': 0.03, 'colsample_bytree': 0.3, 'max_depth': 10,
            'subsample': 0.8, 'lambda': 1., 'nthread': -1, 'booster' : 'gbtree', 'silent': 1,
            'eval_metric': 'rmse', 'objective': 'reg:linear'}

# You could try to train with more epoch
model = xgb.train(xgb_pars, dtrain, 150, watchlist, early_stopping_rounds=2,
                  maximize=False, verbose_eval=1)
print('Modeling RMSLE %.5f' % model.best_score)
```

```
[0]     train-rmse:1033.53943   valid-rmse:1033.33704
Multiple eval metrics have been passed: 'valid-rmse' will be used for early stoppi
ng.

Will train until valid-rmse hasn't improved in 2 rounds.
[1]     train-rmse:1007.05750   valid-rmse:1006.87079
[2]     train-rmse:986.58221    valid-rmse:986.39410
[3]     train-rmse:969.15137    valid-rmse:968.97174
[4]     train-rmse:952.45483    valid-rmse:952.28345
[5]     train-rmse:934.14502    valid-rmse:933.97900
[6]     train-rmse:911.11523    valid-rmse:910.95721
[7]     train-rmse:896.33740    valid-rmse:896.18811
[8]     train-rmse:882.19605    valid-rmse:882.05652
[9]     train-rmse:868.67450    valid-rmse:868.54218
[10]    train-rmse:854.27661    valid-rmse:854.17450
[11]    train-rmse:839.65082    valid-rmse:839.55621
[12]    train-rmse:827.83771    valid-rmse:827.74622
[13]    train-rmse:816.55518    valid-rmse:816.47266
[14]    train-rmse:797.75421    valid-rmse:797.68420
[15]    train-rmse:785.26068    valid-rmse:785.19739
[16]    train-rmse:775.35974    valid-rmse:775.30548
[17]    train-rmse:764.51233    valid-rmse:764.48810
[18]    train-rmse:755.51111    valid-rmse:755.49054
[19]    train-rmse:745.56177    valid-rmse:745.56665
[20]    train-rmse:736.07782    valid-rmse:736.10413
[21]    train-rmse:728.29126    valid-rmse:728.32416
[22]    train-rmse:719.62000    valid-rmse:719.68256
[23]    train-rmse:710.40082    valid-rmse:710.47034
[24]    train-rmse:702.46277    valid-rmse:702.56201
[25]    train-rmse:696.07617    valid-rmse:696.18243
[26]    train-rmse:690.00482    valid-rmse:690.11804
[27]    train-rmse:682.11334    valid-rmse:682.24048
[28]    train-rmse:674.62366    valid-rmse:674.76648
[29]    train-rmse:669.42487    valid-rmse:669.57074
[30]    train-rmse:662.54126    valid-rmse:662.70758
[31]    train-rmse:657.84393    valid-rmse:658.01422
[32]    train-rmse:645.26746    valid-rmse:645.45380
[33]    train-rmse:641.00018    valid-rmse:641.19147
[34]    train-rmse:636.95636    valid-rmse:637.15436
[35]    train-rmse:631.97308    valid-rmse:632.19879
[36]    train-rmse:627.25513    valid-rmse:627.50446
[37]    train-rmse:623.82861    valid-rmse:624.08215
[38]    train-rmse:620.58124    valid-rmse:620.83832
[39]    train-rmse:609.58008    valid-rmse:609.84906
[40]    train-rmse:606.62561    valid-rmse:606.90008
[41]    train-rmse:602.10376    valid-rmse:602.39526
[42]    train-rmse:598.46100    valid-rmse:598.77783
[43]    train-rmse:595.01849    valid-rmse:595.36462
[44]    train-rmse:585.13025    valid-rmse:585.48810
[45]    train-rmse:581.28589    valid-rmse:581.66437
[46]    train-rmse:578.29321    valid-rmse:578.70374
[47]    train-rmse:575.46582    valid-rmse:575.90472
[48]    train-rmse:573.54486    valid-rmse:573.98773
[49]    train-rmse:571.73590    valid-rmse:572.18005
[50]    train-rmse:570.02106    valid-rmse:570.46924
[51]    train-rmse:561.24719    valid-rmse:561.70502
[52]    train-rmse:559.70020    valid-rmse:560.16016
[53]    train-rmse:557.52264    valid-rmse:558.01324
[54]    train-rmse:556.15356    valid-rmse:556.64691
[55]    train-rmse:554.18292    valid-rmse:554.69623
[56]    train-rmse:551.50983    valid-rmse:552.05182
```

```
[57]	train-rmse:550.35181	valid-rmse:550.89581
[58]	train-rmse:542.51404	valid-rmse:543.08203
[59]	train-rmse:540.87707	valid-rmse:541.47089
[60]	train-rmse:533.47675	valid-rmse:534.08875
[61]	train-rmse:532.53992	valid-rmse:533.15320
[62]	train-rmse:525.52692	valid-rmse:526.15546
[63]	train-rmse:524.18713	valid-rmse:524.83972
[64]	train-rmse:517.56592	valid-rmse:518.23804
[65]	train-rmse:516.80353	valid-rmse:517.47790
[66]	train-rmse:510.53372	valid-rmse:511.22864
[67]	train-rmse:504.57028	valid-rmse:505.28583
[68]	train-rmse:498.87863	valid-rmse:499.61563
[69]	train-rmse:498.26328	valid-rmse:499.00180
[70]	train-rmse:497.68433	valid-rmse:498.42221
[71]	train-rmse:497.13705	valid-rmse:497.87515
[72]	train-rmse:491.82178	valid-rmse:492.58267
[73]	train-rmse:490.46088	valid-rmse:491.24863
[74]	train-rmse:489.18497	valid-rmse:489.99258
[75]	train-rmse:488.74805	valid-rmse:489.55609
[76]	train-rmse:487.55420	valid-rmse:488.38562
[77]	train-rmse:482.72299	valid-rmse:483.57312
[78]	train-rmse:482.35669	valid-rmse:483.20660
[79]	train-rmse:481.70035	valid-rmse:482.57910
[80]	train-rmse:477.17795	valid-rmse:478.07950
[81]	train-rmse:476.86816	valid-rmse:477.76914
[82]	train-rmse:472.58173	valid-rmse:473.50180
[83]	train-rmse:471.68948	valid-rmse:472.63974
[84]	train-rmse:471.42786	valid-rmse:472.37759
[85]	train-rmse:470.60773	valid-rmse:471.58359
[86]	train-rmse:469.83411	valid-rmse:470.83432
[87]	train-rmse:469.61334	valid-rmse:470.61331
[88]	train-rmse:469.40601	valid-rmse:470.40677
[89]	train-rmse:465.51755	valid-rmse:466.53980
[90]	train-rmse:464.85535	valid-rmse:465.90308
[91]	train-rmse:464.45517	valid-rmse:465.52841
[92]	train-rmse:464.07147	valid-rmse:465.17127
[93]	train-rmse:460.47586	valid-rmse:461.60211
[94]	train-rmse:460.32929	valid-rmse:461.45529
[95]	train-rmse:460.18988	valid-rmse:461.31522
[96]	train-rmse:459.87097	valid-rmse:461.01914
[97]	train-rmse:459.33115	valid-rmse:460.50769
[98]	train-rmse:459.21799	valid-rmse:460.39343
[99]	train-rmse:459.10837	valid-rmse:460.28421
[100]	train-rmse:458.61093	valid-rmse:459.81094
[101]	train-rmse:458.14002	valid-rmse:459.36176
[102]	train-rmse:457.69672	valid-rmse:458.94571
[103]	train-rmse:457.61185	valid-rmse:458.86072
[104]	train-rmse:457.53125	valid-rmse:458.77994
[105]	train-rmse:457.27786	valid-rmse:458.55078
[106]	train-rmse:454.06958	valid-rmse:455.36911
[107]	train-rmse:453.69754	valid-rmse:455.01880
[108]	train-rmse:453.63452	valid-rmse:454.95456
[109]	train-rmse:453.57501	valid-rmse:454.89477
[110]	train-rmse:453.51987	valid-rmse:454.83853
[111]	train-rmse:450.50470	valid-rmse:451.84424
[112]	train-rmse:450.18616	valid-rmse:451.55316
[113]	train-rmse:449.88562	valid-rmse:451.27103
[114]	train-rmse:449.84088	valid-rmse:451.22693
[115]	train-rmse:449.55832	valid-rmse:450.96832
[116]	train-rmse:449.29538	valid-rmse:450.72345
[117]	train-rmse:449.25812	valid-rmse:450.68558
```

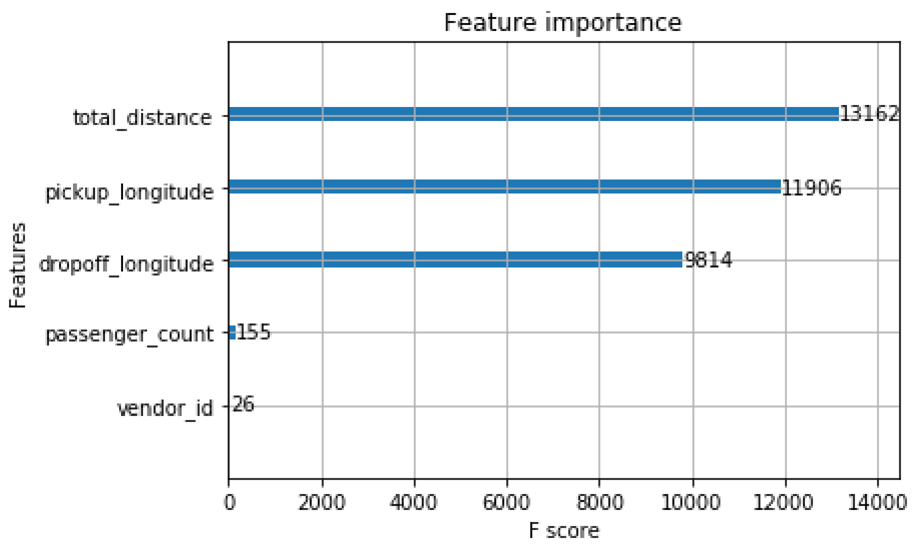
```
[118]	train-rmse:449.22433	valid-rmse:450.65146
[119]	train-rmse:449.04413	valid-rmse:450.49466
[120]	train-rmse:449.01337	valid-rmse:450.46375
[121]	train-rmse:448.84589	valid-rmse:450.31174
[122]	train-rmse:448.68527	valid-rmse:450.17734
[123]	train-rmse:448.44324	valid-rmse:449.95847
[124]	train-rmse:448.41815	valid-rmse:449.93320
[125]	train-rmse:448.39481	valid-rmse:449.90982
[126]	train-rmse:448.24750	valid-rmse:449.78436
[127]	train-rmse:445.55811	valid-rmse:447.11386
[128]	train-rmse:445.42810	valid-rmse:447.00226
[129]	train-rmse:445.40985	valid-rmse:446.98288
[130]	train-rmse:445.39224	valid-rmse:446.96542
[131]	train-rmse:445.27249	valid-rmse:446.86466
[132]	train-rmse:445.25766	valid-rmse:446.84988
[133]	train-rmse:445.24301	valid-rmse:446.83481
[134]	train-rmse:445.22797	valid-rmse:446.82059
[135]	train-rmse:445.03259	valid-rmse:446.64594
[136]	train-rmse:442.51556	valid-rmse:444.15012
[137]	train-rmse:442.50412	valid-rmse:444.13843
[138]	train-rmse:440.12964	valid-rmse:441.78677
[139]	train-rmse:439.97345	valid-rmse:441.65106
[140]	train-rmse:439.82407	valid-rmse:441.51999
[141]	train-rmse:437.59549	valid-rmse:439.31174
[142]	train-rmse:437.46930	valid-rmse:439.20569
[143]	train-rmse:437.35455	valid-rmse:439.10284
[144]	train-rmse:437.34714	valid-rmse:439.09485
[145]	train-rmse:437.24973	valid-rmse:439.02640
[146]	train-rmse:437.24292	valid-rmse:439.01977
[147]	train-rmse:437.23712	valid-rmse:439.01318
[148]	train-rmse:437.14810	valid-rmse:438.94934
[149]	train-rmse:437.14221	valid-rmse:438.94327
Modeling RMSLE 438.94327
```

In [461]:

```
xgb.plot_importance(model)
```

Out[461]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x249736482c8>
```



In [462]:

```
y_pred = model.predict(dtest)
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.5425089081711917
RMSE is 441.8027011064124
```

In [423]:

```
train_data.columns
```

Out[423]:

```
Index(['id', 'vendor_id', 'pickup_datetime', 'dropoff_datetime',
       'passenger_count', 'pickup_longitude', 'pickup_latitude',
       'dropoff_longitude', 'dropoff_latitude', 'store_and_fwd_flag',
       'trip_duration', 'total_distance', 'total_travel_time', 'pick_month',
       'hour', 'week_of_year', 'day_of_year', 'day_of_week'],
      dtype='object')
```

In [56]:

```
df4 = train_data[['vendor_id','passenger_count','trip_duration','total_distance','day_of_week',
'hour']]
```

In [463]:

```
df4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1444660 entries, 0 to 1450294
Data columns (total 6 columns):
vendor_id          1444660 non-null int64
passenger_count    1444660 non-null int64
trip_duration      1444660 non-null int64
total_distance     1444660 non-null float64
day_of_week        1444660 non-null int64
hour               1444660 non-null int64
dtypes: float64(1), int64(5)
memory usage: 77.2 MB
```

In [427]:

```
y = df4['trip_duration']
X = df4.drop('trip_duration',axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
Xtr, Xv, ytr, yv = train_test_split(X_train, y_train, test_size=0.2)
```

In [431]:

```
dtrain = xgb.DMatrix(Xtr, label=ytr)
dvalid = xgb.DMatrix(Xv, label=yv)
dtest = xgb.DMatrix(X_test)
watchlist = [(dtrain, 'train'), (dvalid, 'valid')]

xgb_pars = {'min_child_weight': 50, 'eta': 0.03, 'colsample_bytree': 0.3, 'max_depth': 10,
            'subsample': 0.8, 'lambda': 1., 'nthread': -1, 'booster' : 'gbtree', 'silent': 1,
            'eval_metric': 'rmse', 'objective': 'reg:linear'}

# You could try to train with more epoch
model = xgb.train(xgb_pars, dtrain, 150, watchlist, early_stopping_rounds=2,
                  maximize=False, verbose_eval=1)
print('Modeling RMSLE %.5f' % model.best_score)
```

```
[0]	train-rmse:1035.86633	valid-rmse:1035.28223
Multiple eval metrics have been passed: 'valid-rmse' will be used for early stoppi
ng.

Will train until valid-rmse hasn't improved in 2 rounds.
[1]	train-rmse:1016.94488	valid-rmse:1016.37116
[2]	train-rmse:998.98340	valid-rmse:998.41461
[3]	train-rmse:973.64490	valid-rmse:973.04370
[4]	train-rmse:956.96289	valid-rmse:956.36883
[5]	train-rmse:941.07147	valid-rmse:940.48517
[6]	train-rmse:925.70227	valid-rmse:925.12811
[7]	train-rmse:902.92407	valid-rmse:902.31690
[8]	train-rmse:888.83124	valid-rmse:888.23010
[9]	train-rmse:875.36218	valid-rmse:874.76807
[10]	train-rmse:862.54987	valid-rmse:861.96210
[11]	train-rmse:850.34003	valid-rmse:849.76178
[12]	train-rmse:830.24622	valid-rmse:829.63928
[13]	train-rmse:818.95581	valid-rmse:818.35370
[14]	train-rmse:808.03894	valid-rmse:807.45001
[15]	train-rmse:797.83032	valid-rmse:797.24676
[16]	train-rmse:788.03290	valid-rmse:787.45398
[17]	train-rmse:778.74249	valid-rmse:778.16986
[18]	train-rmse:761.25604	valid-rmse:760.65881
[19]	train-rmse:752.75641	valid-rmse:752.16577
[20]	train-rmse:744.66260	valid-rmse:744.08069
[21]	train-rmse:736.91193	valid-rmse:736.33478
[22]	train-rmse:729.60523	valid-rmse:729.03674
[23]	train-rmse:722.66480	valid-rmse:722.10230
[24]	train-rmse:716.07721	valid-rmse:715.52270
[25]	train-rmse:709.74823	valid-rmse:709.19836
[26]	train-rmse:703.74884	valid-rmse:703.20599
[27]	train-rmse:698.11438	valid-rmse:697.57996
[28]	train-rmse:692.76929	valid-rmse:692.24262
[29]	train-rmse:678.55994	valid-rmse:678.01544
[30]	train-rmse:673.70355	valid-rmse:673.16644
[31]	train-rmse:660.26178	valid-rmse:659.71222
[32]	train-rmse:655.58496	valid-rmse:655.04535
[33]	train-rmse:642.84222	valid-rmse:642.29633
[34]	train-rmse:638.74481	valid-rmse:638.20477
[35]	train-rmse:634.91711	valid-rmse:634.38544
[36]	train-rmse:631.30395	valid-rmse:630.77881
[37]	train-rmse:619.52094	valid-rmse:618.99518
[38]	train-rmse:608.23041	valid-rmse:607.70935
[39]	train-rmse:604.82532	valid-rmse:604.31024
[40]	train-rmse:601.78668	valid-rmse:601.27765
[41]	train-rmse:598.97778	valid-rmse:598.47650
[42]	train-rmse:596.32080	valid-rmse:595.82696
[43]	train-rmse:593.81006	valid-rmse:593.32300
[44]	train-rmse:591.18109	valid-rmse:590.70020
[45]	train-rmse:588.94983	valid-rmse:588.47668
[46]	train-rmse:586.83972	valid-rmse:586.37219
[47]	train-rmse:584.83649	valid-rmse:584.37616
[48]	train-rmse:574.97632	valid-rmse:574.52508
[49]	train-rmse:565.53308	valid-rmse:565.09186
[50]	train-rmse:563.73120	valid-rmse:563.29645
[51]	train-rmse:561.81903	valid-rmse:561.38855
[52]	train-rmse:560.20898	valid-rmse:559.78467
[53]	train-rmse:558.75476	valid-rmse:558.33673
[54]	train-rmse:549.98328	valid-rmse:549.57495
[55]	train-rmse:548.67200	valid-rmse:548.26977
[56]	train-rmse:547.44080	valid-rmse:547.04431
```

```
[57]	train-rmse:546.20526	valid-rmse:545.81457
[58]	train-rmse:544.82996	valid-rmse:544.44183
[59]	train-rmse:543.79218	valid-rmse:543.41046
[60]	train-rmse:542.55157	valid-rmse:542.17377
[61]	train-rmse:534.44421	valid-rmse:534.08221
[62]	train-rmse:533.30701	valid-rmse:532.94727
[63]	train-rmse:532.47504	valid-rmse:532.12079
[64]	train-rmse:531.45007	valid-rmse:531.09949
[65]	train-rmse:530.64221	valid-rmse:530.29553
[66]	train-rmse:529.71741	valid-rmse:529.37280
[67]	train-rmse:528.84308	valid-rmse:528.49994
[68]	train-rmse:528.02478	valid-rmse:527.68457
[69]	train-rmse:527.38007	valid-rmse:527.04156
[70]	train-rmse:519.83746	valid-rmse:519.52032
[71]	train-rmse:519.25171	valid-rmse:518.93726
[72]	train-rmse:518.55432	valid-rmse:518.24182
[73]	train-rmse:518.09106	valid-rmse:517.78278
[74]	train-rmse:517.65277	valid-rmse:517.34906
[75]	train-rmse:517.18262	valid-rmse:516.88013
[76]	train-rmse:516.79639	valid-rmse:516.49860
[77]	train-rmse:516.24066	valid-rmse:515.94434
[78]	train-rmse:509.18002	valid-rmse:508.90424
[79]	train-rmse:508.84616	valid-rmse:508.57428
[80]	train-rmse:508.34787	valid-rmse:508.07724
[81]	train-rmse:501.62524	valid-rmse:501.38123
[82]	train-rmse:501.16483	valid-rmse:500.92239
[83]	train-rmse:500.90381	valid-rmse:500.66473
[84]	train-rmse:494.50952	valid-rmse:494.29623
[85]	train-rmse:494.27411	valid-rmse:494.06464
[86]	train-rmse:494.05447	valid-rmse:493.84769
[87]	train-rmse:493.78207	valid-rmse:493.57736
[88]	train-rmse:487.71152	valid-rmse:487.53400
[89]	train-rmse:487.33270	valid-rmse:487.15601
[90]	train-rmse:487.15747	valid-rmse:486.98267
[91]	train-rmse:486.99060	valid-rmse:486.81744
[92]	train-rmse:486.83118	valid-rmse:486.66171
[93]	train-rmse:486.50592	valid-rmse:486.33646
[94]	train-rmse:480.73352	valid-rmse:480.60187
[95]	train-rmse:480.54001	valid-rmse:480.40961
[96]	train-rmse:480.41363	valid-rmse:480.28506
[97]	train-rmse:480.29770	valid-rmse:480.17184
[98]	train-rmse:474.81631	valid-rmse:474.73032
[99]	train-rmse:474.65283	valid-rmse:474.56833
[100]	train-rmse:474.55597	valid-rmse:474.47269
[101]	train-rmse:474.46347	valid-rmse:474.38251
[102]	train-rmse:474.37720	valid-rmse:474.29987
[103]	train-rmse:469.19217	valid-rmse:469.15097
[104]	train-rmse:469.05597	valid-rmse:469.01541
[105]	train-rmse:468.98169	valid-rmse:468.94183
[106]	train-rmse:468.71796	valid-rmse:468.67740
[107]	train-rmse:468.65338	valid-rmse:468.61447
[108]	train-rmse:463.71670	valid-rmse:463.71585
[109]	train-rmse:459.03040	valid-rmse:459.06528
[110]	train-rmse:454.55725	valid-rmse:454.63061
[111]	train-rmse:454.29892	valid-rmse:454.37106
[112]	train-rmse:454.25055	valid-rmse:454.32358
[113]	train-rmse:454.20380	valid-rmse:454.27847
[114]	train-rmse:454.09970	valid-rmse:454.17502
[115]	train-rmse:454.05884	valid-rmse:454.13678
[116]	train-rmse:454.02084	valid-rmse:454.09982
[117]	train-rmse:453.92777	valid-rmse:454.00736
```

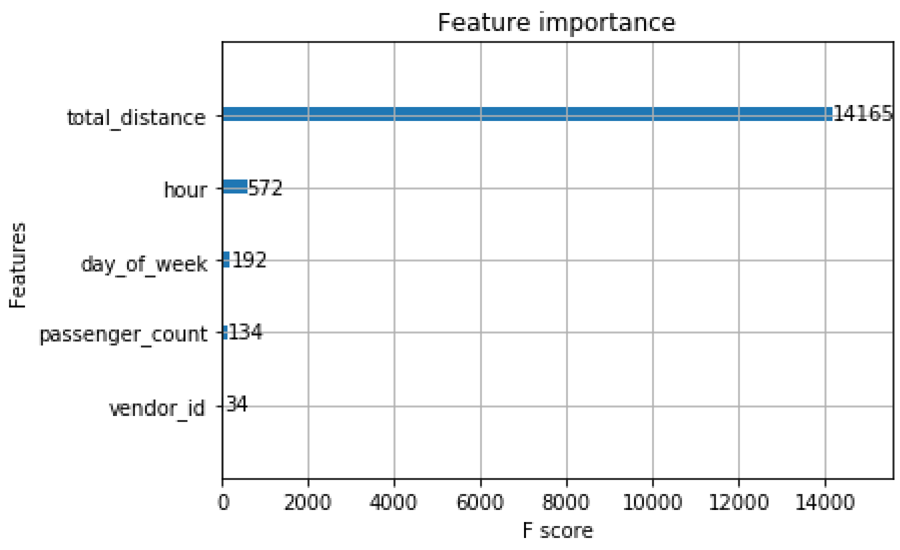
```
[118]	train-rmse:449.70895	valid-rmse:449.82782
[119]	train-rmse:449.67432	valid-rmse:449.79453
[120]	train-rmse:445.65573	valid-rmse:445.81335
[121]	train-rmse:445.62558	valid-rmse:445.78403
[122]	train-rmse:445.59607	valid-rmse:445.75647
[123]	train-rmse:445.57123	valid-rmse:445.73264
[124]	train-rmse:445.49081	valid-rmse:445.65259
[125]	train-rmse:445.41702	valid-rmse:445.57935
[126]	train-rmse:445.39343	valid-rmse:445.55679
[127]	train-rmse:445.16675	valid-rmse:445.32776
[128]	train-rmse:445.14581	valid-rmse:445.30820
[129]	train-rmse:445.07901	valid-rmse:445.24136
[130]	train-rmse:445.01608	valid-rmse:445.17804
[131]	train-rmse:444.99832	valid-rmse:445.16156
[132]	train-rmse:441.18692	valid-rmse:441.38840
[133]	train-rmse:441.12824	valid-rmse:441.32950
[134]	train-rmse:441.07434	valid-rmse:441.27521
[135]	train-rmse:441.06134	valid-rmse:441.26370
[136]	train-rmse:440.84814	valid-rmse:441.04782
[137]	train-rmse:440.79773	valid-rmse:440.99731
[138]	train-rmse:440.59860	valid-rmse:440.79453
[139]	train-rmse:440.58826	valid-rmse:440.78571
[140]	train-rmse:440.57950	valid-rmse:440.77795
[141]	train-rmse:440.39197	valid-rmse:440.58710
[142]	train-rmse:440.38284	valid-rmse:440.57990
[143]	train-rmse:440.37549	valid-rmse:440.57358
[144]	train-rmse:440.33191	valid-rmse:440.52850
[145]	train-rmse:440.32330	valid-rmse:440.52042
[146]	train-rmse:436.68085	valid-rmse:436.92673
[147]	train-rmse:436.63867	valid-rmse:436.88501
[148]	train-rmse:436.63165	valid-rmse:436.87756
[149]	train-rmse:433.18781	valid-rmse:433.47409
Modeling RMSLE 433.47409
```

In [432]:

```
xgb.plot_importance(model)
```

Out[432]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x24971c32788>
```



In [433]:

```
y_pred = model.predict(dtest)
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.5541557633859144
RMSE is 436.7434000976652
```

In [434]:

```
df5 = pd.get_dummies(df4,columns=['day_of_week','hour'])
```

In [435]:

```
y = df5['trip_duration']
X = df5.drop('trip_duration',axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
Xtr, Xv, ytr, yv = train_test_split(X_train, y_train, test_size=0.2)
```

In [437]:

```
dtrain = xgb.DMatrix(Xtr, label=ytr)
dvalid = xgb.DMatrix(Xv, label=yv)
dtest = xgb.DMatrix(X_test)
watchlist = [(dtrain, 'train'), (dvalid, 'valid')]

xgb_pars = {'min_child_weight': 50, 'eta': 0.03, 'colsample_bytree': 0.3, 'max_depth': 10,
            'subsample': 0.8, 'lambda': 1., 'nthread': -1, 'booster' : 'gbtree', 'silent': 1,
            'eval_metric': 'rmse', 'objective': 'reg:linear'}

# You could try to train with more epoch
model = xgb.train(xgb_pars, dtrain, 150, watchlist, early_stopping_rounds=2,
                  maximize=False, verbose_eval=1)
print('Modeling RMSLE %.5f' % model.best_score)
```

```
[0]	train-rmse:1036.98572	valid-rmse:1034.80566
Multiple eval metrics have been passed: 'valid-rmse' will be used for early stoppi
ng.

Will train until valid-rmse hasn't improved in 2 rounds.
[1]	train-rmse:1018.15753	valid-rmse:1015.98218
[2]	train-rmse:1000.11365	valid-rmse:997.94470
[3]	train-rmse:982.76794	valid-rmse:980.61096
[4]	train-rmse:966.23090	valid-rmse:964.08282
[5]	train-rmse:941.87555	valid-rmse:939.72162
[6]	train-rmse:926.58929	valid-rmse:924.44721
[7]	train-rmse:911.96954	valid-rmse:909.84119
[8]	train-rmse:897.98401	valid-rmse:895.86414
[9]	train-rmse:875.88831	valid-rmse:873.74420
[10]	train-rmse:862.99445	valid-rmse:860.86383
[11]	train-rmse:850.66827	valid-rmse:848.55145
[12]	train-rmse:838.88995	valid-rmse:836.79669
[13]	train-rmse:827.67548	valid-rmse:825.59992
[14]	train-rmse:816.96600	valid-rmse:814.90045
[15]	train-rmse:797.83276	valid-rmse:795.75903
[16]	train-rmse:779.22119	valid-rmse:777.14520
[17]	train-rmse:769.77734	valid-rmse:767.71313
[18]	train-rmse:752.28217	valid-rmse:750.19171
[19]	train-rmse:743.56067	valid-rmse:741.48315
[20]	train-rmse:727.09534	valid-rmse:725.01355
[21]	train-rmse:719.12048	valid-rmse:717.05420
[22]	train-rmse:711.54919	valid-rmse:709.49634
[23]	train-rmse:696.26379	valid-rmse:694.20306
[24]	train-rmse:689.35773	valid-rmse:687.30762
[25]	train-rmse:682.75738	valid-rmse:680.72247
[26]	train-rmse:676.52289	valid-rmse:674.50092
[27]	train-rmse:670.58252	valid-rmse:668.57751
[28]	train-rmse:664.94482	valid-rmse:662.95416
[29]	train-rmse:659.59113	valid-rmse:657.61200
[30]	train-rmse:654.48676	valid-rmse:652.51984
[31]	train-rmse:649.60260	valid-rmse:647.65015
[32]	train-rmse:645.02972	valid-rmse:643.09058
[33]	train-rmse:640.65967	valid-rmse:638.73419
[34]	train-rmse:636.56970	valid-rmse:634.66046
[35]	train-rmse:632.64221	valid-rmse:630.74493
[36]	train-rmse:628.92706	valid-rmse:627.04791
[37]	train-rmse:625.40582	valid-rmse:623.54639
[38]	train-rmse:622.03748	valid-rmse:620.19879
[39]	train-rmse:618.91480	valid-rmse:617.09399
[40]	train-rmse:615.93225	valid-rmse:614.12274
[41]	train-rmse:613.10138	valid-rmse:611.30988
[42]	train-rmse:610.44049	valid-rmse:608.66687
[43]	train-rmse:607.88886	valid-rmse:606.12775
[44]	train-rmse:605.51367	valid-rmse:603.76477
[45]	train-rmse:594.21509	valid-rmse:592.45947
[46]	train-rmse:592.01172	valid-rmse:590.27075
[47]	train-rmse:589.94806	valid-rmse:588.21997
[48]	train-rmse:587.95081	valid-rmse:586.23321
[49]	train-rmse:577.45416	valid-rmse:575.71826
[50]	train-rmse:567.36481	valid-rmse:565.61920
[51]	train-rmse:557.63489	valid-rmse:555.89386
[52]	train-rmse:555.98932	valid-rmse:554.26367
[53]	train-rmse:554.46930	valid-rmse:552.75806
[54]	train-rmse:553.00433	valid-rmse:551.30823
[55]	train-rmse:551.64770	valid-rmse:549.96411
[56]	train-rmse:550.31366	valid-rmse:548.64368
```

```
[57]    train-rmse:549.09241    valid-rmse:547.43762
[58]    train-rmse:547.94586    valid-rmse:546.30279
[59]    train-rmse:538.94055    valid-rmse:537.29437
[60]    train-rmse:530.36633    valid-rmse:528.71960
[61]    train-rmse:529.35193    valid-rmse:527.71765
[62]    train-rmse:521.18689    valid-rmse:519.57379
[63]    train-rmse:520.23669    valid-rmse:518.63489
[64]    train-rmse:519.30933    valid-rmse:517.72150
[65]    train-rmse:518.49170    valid-rmse:516.91321
[66]    train-rmse:517.71783    valid-rmse:516.15021
[67]    train-rmse:517.00201    valid-rmse:515.44183
[68]    train-rmse:516.24365    valid-rmse:514.69226
[69]    train-rmse:515.58655    valid-rmse:514.04462
[70]    train-rmse:514.97900    valid-rmse:513.44611
[71]    train-rmse:507.42505    valid-rmse:505.91177
[72]    train-rmse:506.84668    valid-rmse:505.34448
[73]    train-rmse:506.26721    valid-rmse:504.76898
[74]    train-rmse:499.13074    valid-rmse:497.64203
[75]    train-rmse:498.62164    valid-rmse:497.14215
[76]    train-rmse:491.67047    valid-rmse:490.20276
[77]    train-rmse:491.19125    valid-rmse:489.73358
[78]    train-rmse:490.73035    valid-rmse:489.27890
[79]    train-rmse:490.32394    valid-rmse:488.87918
[80]    train-rmse:489.95276    valid-rmse:488.51956
[81]    train-rmse:489.59933    valid-rmse:488.17520
[82]    train-rmse:489.25793    valid-rmse:487.84134
[83]    train-rmse:488.94428    valid-rmse:487.53662
[84]    train-rmse:488.62488    valid-rmse:487.22430
[85]    train-rmse:488.30411    valid-rmse:486.91034
[86]    train-rmse:481.87909    valid-rmse:480.49292
[87]    train-rmse:481.59152    valid-rmse:480.21695
[88]    train-rmse:475.40704    valid-rmse:474.05173
[89]    train-rmse:469.64987    valid-rmse:468.30926
[90]    train-rmse:464.01071    valid-rmse:462.68411
[91]    train-rmse:463.75299    valid-rmse:462.43072
[92]    train-rmse:463.53152    valid-rmse:462.21417
[93]    train-rmse:463.31219    valid-rmse:462.00378
[94]    train-rmse:463.09558    valid-rmse:461.79144
[95]    train-rmse:462.86935    valid-rmse:461.57284
[96]    train-rmse:462.67719    valid-rmse:461.38312
[97]    train-rmse:457.30032    valid-rmse:456.01291
[98]    train-rmse:457.12115    valid-rmse:455.83911
[99]    train-rmse:456.90543    valid-rmse:455.62802
[100]   train-rmse:451.76541    valid-rmse:450.50873
[101]   train-rmse:451.59784    valid-rmse:450.34576
[102]   train-rmse:446.68759    valid-rmse:445.45004
[103]   train-rmse:446.52664    valid-rmse:445.29092
[104]   train-rmse:441.93570    valid-rmse:440.72223
[105]   train-rmse:441.81134    valid-rmse:440.60373
[106]   train-rmse:437.38031    valid-rmse:436.21243
[107]   train-rmse:437.24576    valid-rmse:436.08463
[108]   train-rmse:437.12335    valid-rmse:435.96683
[109]   train-rmse:436.96957    valid-rmse:435.81726
[110]   train-rmse:436.84366    valid-rmse:435.70047
[111]   train-rmse:436.72507    valid-rmse:435.58804
[112]   train-rmse:436.62054    valid-rmse:435.48489
[113]   train-rmse:436.49805    valid-rmse:435.36252
[114]   train-rmse:436.35278    valid-rmse:435.21869
[115]   train-rmse:436.27625    valid-rmse:435.14600
[116]   train-rmse:432.12561    valid-rmse:431.00855
[117]   train-rmse:432.03714    valid-rmse:430.92206
```

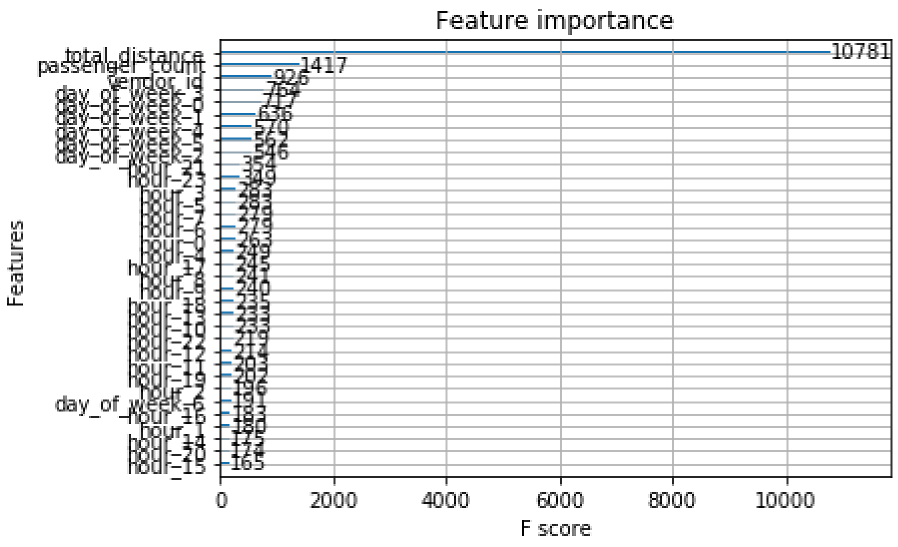
```
[118]	train-rmse:431.94333	valid-rmse:430.83203
[119]	train-rmse:427.96637	valid-rmse:426.87320
[120]	train-rmse:427.87387	valid-rmse:426.78067
[121]	train-rmse:427.78085	valid-rmse:426.69089
[122]	train-rmse:427.71637	valid-rmse:426.63129
[123]	train-rmse:423.98178	valid-rmse:422.91763
[124]	train-rmse:423.91629	valid-rmse:422.85660
[125]	train-rmse:423.82696	valid-rmse:422.77457
[126]	train-rmse:420.25760	valid-rmse:419.22659
[127]	train-rmse:420.20093	valid-rmse:419.17450
[128]	train-rmse:420.13797	valid-rmse:419.11462
[129]	train-rmse:420.06799	valid-rmse:419.04852
[130]	train-rmse:419.97345	valid-rmse:418.95636
[131]	train-rmse:419.89630	valid-rmse:418.88171
[132]	train-rmse:419.84030	valid-rmse:418.82871
[133]	train-rmse:419.76999	valid-rmse:418.76370
[134]	train-rmse:416.33459	valid-rmse:415.34772
[135]	train-rmse:416.28375	valid-rmse:415.29883
[136]	train-rmse:412.97433	valid-rmse:412.02148
[137]	train-rmse:412.90466	valid-rmse:411.95630
[138]	train-rmse:412.83810	valid-rmse:411.89224
[139]	train-rmse:412.77832	valid-rmse:411.83566
[140]	train-rmse:412.72235	valid-rmse:411.77976
[141]	train-rmse:409.66214	valid-rmse:408.74506
[142]	train-rmse:409.60449	valid-rmse:408.68988
[143]	train-rmse:406.68036	valid-rmse:405.78586
[144]	train-rmse:403.82123	valid-rmse:402.95804
[145]	train-rmse:403.76001	valid-rmse:402.89929
[146]	train-rmse:403.71225	valid-rmse:402.85577
[147]	train-rmse:403.66904	valid-rmse:402.81531
[148]	train-rmse:403.61408	valid-rmse:402.76065
[149]	train-rmse:400.94742	valid-rmse:400.12427
Modeling RMSLE 400.12427
```

In [439]:

```
xgb.plot_importance(model)
```

Out[439]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2497234ba48>
```



In [438]:

```
y_pred = model.predict(dtest)
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.6199615509943914
RMSE is 400.60676775504254
```

In [441]:

```
df4.head()
```

Out[441]:

	vendor_id	passenger_count	trip_duration	total_distance	day_of_week	hour
0	2	1	455	2009.1	0	17
1	1	1	663	2513.2	6	0
2	2	1	2124	11060.8	1	11
3	2	1	429	1779.4	2	19
4	2	1	435	1614.9	5	13

In [57]:

```
df6 = df4.copy()
```

In [58]:

```
df6['sin_hour'] = np.sin(2*np.pi*df6.hour/24)
df6['cos_hour'] = np.cos(2*np.pi*df6.hour/24)
df6['sin_week'] = np.sin(2*np.pi*df6.day_of_week/6)
df6['cos_week'] = np.cos(2*np.pi*df6.day_of_week/6)
```

In [59]:

```
df6 = df6.drop(['hour','day_of_week'],axis = 1)
```

In [60]:

```
df6.head()
```

Out[60]:

	vendor_id	passenger_count	trip_duration	total_distance	sin_hour	cos_hour	sin_we
0	2	1	455	2009.1	-0.965926	-0.258819	0.000000e+
1	1	1	663	2513.2	0.000000	1.000000	-2.44929
2	2	1	2124	11060.8	0.258819	-0.965926	8.660254e-
3	2	1	429	1779.4	-0.965926	0.258819	8.660254e-
4	2	1	435	1614.9	-0.258819	-0.965926	-8.66025

In [61]:

```
y = df6['trip_duration']
X = df6.drop('trip_duration',axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
Xtr, Xv, ytr, yv = train_test_split(X_train, y_train, test_size=0.2)
```

In [62]:

```
dtrain = xgb.DMatrix(Xtr, label=ytr)
dvalid = xgb.DMatrix(Xv, label=yv)
dtest = xgb.DMatrix(X_test)
watchlist = [(dtrain, 'train'), (dvalid, 'valid')]

xgb_pars = {'min_child_weight': 50, 'eta': 0.03, 'colsample_bytree': 0.3, 'max_depth': 10,
            'subsample': 0.8, 'lambda': 1., 'nthread': -1, 'booster' : 'gbtree', 'silent': 1,
            'eval_metric': 'rmse', 'objective': 'reg:linear'}

# You could try to train with more epoch
model = xgb.train(xgb_pars, dtrain, 150, watchlist, early_stopping_rounds=2,
                  maximize=False, verbose_eval=1)
print('Modeling RMSLE %.5f' % model.best_score)
```

```
[0]     train-rmse:1046.10913   valid-rmse:1047.00903
Multiple eval metrics have been passed: 'valid-rmse' will be used for early stoppi
ng.

Will train until valid-rmse hasn't improved in 2 rounds.
[1]     train-rmse:1019.13019   valid-rmse:1019.99744
[2]     train-rmse:1000.81171   valid-rmse:1001.69513
[3]     train-rmse:975.28699    valid-rmse:976.13098
[4]     train-rmse:958.40723    valid-rmse:959.26685
[5]     train-rmse:942.19562    valid-rmse:943.07214
[6]     train-rmse:926.72394    valid-rmse:927.61401
[7]     train-rmse:903.87640    valid-rmse:904.73938
[8]     train-rmse:889.60785    valid-rmse:890.48212
[9]     train-rmse:875.95886    valid-rmse:876.84686
[10]    train-rmse:862.78345    valid-rmse:863.68420
[11]    train-rmse:850.33179    valid-rmse:851.24731
[12]    train-rmse:829.62616    valid-rmse:830.49634
[13]    train-rmse:818.02423    valid-rmse:818.90863
[14]    train-rmse:807.09045    valid-rmse:807.98529
[15]    train-rmse:796.48822    valid-rmse:797.39716
[16]    train-rmse:786.54651    valid-rmse:787.46796
[17]    train-rmse:777.07165    valid-rmse:778.00073
[18]    train-rmse:759.64325    valid-rmse:760.53796
[19]    train-rmse:742.87024    valid-rmse:743.73352
[20]    train-rmse:734.33752    valid-rmse:735.21466
[21]    train-rmse:726.38525    valid-rmse:727.27173
[22]    train-rmse:718.76483    valid-rmse:719.65936
[23]    train-rmse:703.28137    valid-rmse:704.14197
[24]    train-rmse:696.36292    valid-rmse:697.23163
[25]    train-rmse:689.78839    valid-rmse:690.66467
[26]    train-rmse:683.55054    valid-rmse:684.43097
[27]    train-rmse:677.50226    valid-rmse:678.39185
[28]    train-rmse:663.09558    valid-rmse:663.94275
[29]    train-rmse:657.57587    valid-rmse:658.42999
[30]    train-rmse:644.53149    valid-rmse:645.35962
[31]    train-rmse:639.58673    valid-rmse:640.42053
[32]    train-rmse:634.90839    valid-rmse:635.74573
[33]    train-rmse:622.87622    valid-rmse:623.68756
[34]    train-rmse:618.60718    valid-rmse:619.42352
[35]    train-rmse:614.47339    valid-rmse:615.29584
[36]    train-rmse:603.16681    valid-rmse:603.98120
[37]    train-rmse:592.31543    valid-rmse:593.12103
[38]    train-rmse:582.02185    valid-rmse:582.79993
[39]    train-rmse:578.59723    valid-rmse:579.38074
[40]    train-rmse:575.40002    valid-rmse:576.18475
[41]    train-rmse:565.82587    valid-rmse:566.58881
[42]    train-rmse:556.37238    valid-rmse:557.11993
[43]    train-rmse:553.58447    valid-rmse:554.33630
[44]    train-rmse:550.96179    valid-rmse:551.71808
[45]    train-rmse:542.14954    valid-rmse:542.88946
[46]    train-rmse:533.98120    valid-rmse:534.70868
[47]    train-rmse:531.50964    valid-rmse:532.24170
[48]    train-rmse:529.31219    valid-rmse:530.04932
[49]    train-rmse:521.58350    valid-rmse:522.32391
[50]    train-rmse:519.53644    valid-rmse:520.27972
[51]    train-rmse:517.60602    valid-rmse:518.35437
[52]    train-rmse:515.73480    valid-rmse:516.48798
[53]    train-rmse:507.89227    valid-rmse:508.63538
[54]    train-rmse:506.33569    valid-rmse:507.08160
[55]    train-rmse:499.67114    valid-rmse:500.40253
[56]    train-rmse:493.31833    valid-rmse:494.03705
```

```
[57]	train-rmse:487.26215	valid-rmse:487.96323
[58]	train-rmse:485.76740	valid-rmse:486.47321
[59]	train-rmse:484.55219	valid-rmse:485.26068
[60]	train-rmse:483.29938	valid-rmse:484.01086
[61]	train-rmse:482.21896	valid-rmse:482.93131
[62]	train-rmse:481.19708	valid-rmse:481.91034
[63]	train-rmse:475.66721	valid-rmse:476.36691
[64]	train-rmse:474.59207	valid-rmse:475.29639
[65]	train-rmse:473.72885	valid-rmse:474.43503
[66]	train-rmse:472.92001	valid-rmse:473.62753
[67]	train-rmse:471.94727	valid-rmse:472.65817
[68]	train-rmse:471.01059	valid-rmse:471.72201
[69]	train-rmse:470.14444	valid-rmse:470.85953
[70]	train-rmse:465.08023	valid-rmse:465.79254
[71]	train-rmse:464.26660	valid-rmse:464.97897
[72]	train-rmse:463.66031	valid-rmse:464.37512
[73]	train-rmse:463.09158	valid-rmse:463.80655
[74]	train-rmse:458.35004	valid-rmse:459.06488
[75]	train-rmse:453.84607	valid-rmse:454.54584
[76]	train-rmse:453.37772	valid-rmse:454.07849
[77]	train-rmse:452.88052	valid-rmse:453.58322
[78]	train-rmse:452.44522	valid-rmse:453.14810
[79]	train-rmse:452.05191	valid-rmse:452.75449
[80]	train-rmse:451.54010	valid-rmse:452.24643
[81]	train-rmse:447.25540	valid-rmse:447.97318
[82]	train-rmse:443.01831	valid-rmse:443.73059
[83]	train-rmse:442.56158	valid-rmse:443.27689
[84]	train-rmse:438.54776	valid-rmse:439.26764
[85]	train-rmse:438.12598	valid-rmse:438.84903
[86]	train-rmse:434.44443	valid-rmse:435.16788
[87]	train-rmse:431.04977	valid-rmse:431.76559
[88]	train-rmse:427.72052	valid-rmse:428.44430
[89]	train-rmse:427.34592	valid-rmse:428.07336
[90]	train-rmse:427.12253	valid-rmse:427.85037
[91]	train-rmse:426.78186	valid-rmse:427.51279
[92]	train-rmse:426.47763	valid-rmse:427.20966
[93]	train-rmse:426.19064	valid-rmse:426.92349
[94]	train-rmse:426.01535	valid-rmse:426.74832
[95]	train-rmse:425.60864	valid-rmse:426.34570
[96]	train-rmse:425.22488	valid-rmse:425.96457
[97]	train-rmse:425.03391	valid-rmse:425.77396
[98]	train-rmse:424.89121	valid-rmse:425.63178
[99]	train-rmse:424.71835	valid-rmse:425.45932
[100]	train-rmse:424.59415	valid-rmse:425.33609
[101]	train-rmse:424.26404	valid-rmse:425.00858
[102]	train-rmse:421.05389	valid-rmse:421.81696
[103]	train-rmse:420.93601	valid-rmse:421.69778
[104]	train-rmse:420.83688	valid-rmse:421.59921
[105]	train-rmse:420.74271	valid-rmse:421.50552
[106]	train-rmse:417.92001	valid-rmse:418.67191
[107]	train-rmse:417.75095	valid-rmse:418.50296
[108]	train-rmse:415.00272	valid-rmse:415.76297
[109]	train-rmse:414.77432	valid-rmse:415.53665
[110]	train-rmse:414.54041	valid-rmse:415.30170
[111]	train-rmse:414.43796	valid-rmse:415.19885
[112]	train-rmse:414.37222	valid-rmse:415.13400
[113]	train-rmse:414.11148	valid-rmse:414.87564
[114]	train-rmse:414.05176	valid-rmse:414.81641
[115]	train-rmse:413.97827	valid-rmse:414.74322
[116]	train-rmse:413.85428	valid-rmse:414.61951
[117]	train-rmse:413.80679	valid-rmse:414.57230
```

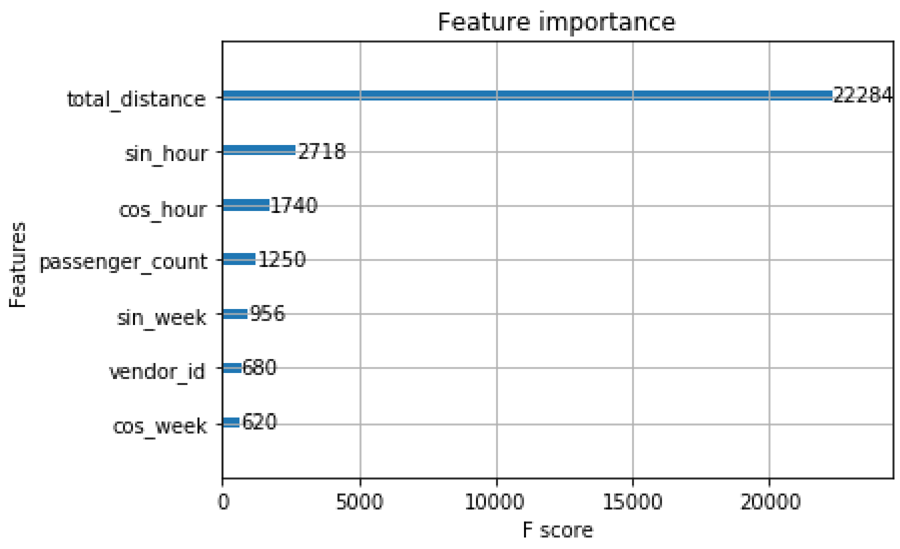
```
[118]	train-rmse:413.66486	valid-rmse:414.43182
[119]	train-rmse:413.60516	valid-rmse:414.37274
[120]	train-rmse:413.38217	valid-rmse:414.15213
[121]	train-rmse:413.34375	valid-rmse:414.11417
[122]	train-rmse:410.62112	valid-rmse:411.39966
[123]	train-rmse:410.45236	valid-rmse:411.22949
[124]	train-rmse:410.33386	valid-rmse:411.11255
[125]	train-rmse:407.85080	valid-rmse:408.64145
[126]	train-rmse:407.82004	valid-rmse:408.61008
[127]	train-rmse:407.77695	valid-rmse:408.56897
[128]	train-rmse:407.68362	valid-rmse:408.47458
[129]	train-rmse:407.57513	valid-rmse:408.36774
[130]	train-rmse:407.54837	valid-rmse:408.34210
[131]	train-rmse:405.09418	valid-rmse:405.90234
[132]	train-rmse:402.94562	valid-rmse:403.75928
[133]	train-rmse:402.75284	valid-rmse:403.56876
[134]	train-rmse:402.71896	valid-rmse:403.53442
[135]	train-rmse:400.67413	valid-rmse:401.49823
[136]	train-rmse:398.75803	valid-rmse:399.58527
[137]	train-rmse:398.67673	valid-rmse:399.50372
[138]	train-rmse:398.65732	valid-rmse:399.48502
[139]	train-rmse:398.64102	valid-rmse:399.46875
[140]	train-rmse:398.62415	valid-rmse:399.45227
[141]	train-rmse:398.48709	valid-rmse:399.31366
[142]	train-rmse:398.30911	valid-rmse:399.13812
[143]	train-rmse:395.84302	valid-rmse:396.69174
[144]	train-rmse:394.11722	valid-rmse:394.96777
[145]	train-rmse:393.95032	valid-rmse:394.80399
[146]	train-rmse:392.13742	valid-rmse:393.01099
[147]	train-rmse:392.06833	valid-rmse:392.94171
[148]	train-rmse:391.95032	valid-rmse:392.82205
[149]	train-rmse:391.93872	valid-rmse:392.81116
Modeling RMSLE 392.81116
```

In [63]:

```
xgb.plot_importance(model)
```

Out[63]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ee9a6de988>
```



In [64]:

```
y_pred = model.predict(dtest)
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.6439666406062137
RMSE is 391.4199296759177
```

In [65]:

```
train_data.columns
```

Out[65]:

```
Index(['id', 'vendor_id', 'pickup_datetime', 'dropoff_datetime',
       'passenger_count', 'pickup_longitude', 'pickup_latitude',
       'dropoff_longitude', 'dropoff_latitude', 'store_and_fwd_flag',
       'trip_duration', 'total_distance', 'total_travel_time', 'pick_month',
       'hour', 'week_of_year', 'day_of_year', 'day_of_week'],
      dtype='object')
```

In [66]:

```
clus = train_data[['pickup_latitude','pickup_longitude']]
```

In [67]:

```
n_clusters = np.arange(1, 10)
sse = []
for n in n_clusters:
    kmeans = KMeans(n_clusters=n)
    kmeans.fit(clus)
    sse.append(kmeans.inertia_)
```

In [68]:

```
sse = pd.DataFrame(sse)
```

In [69]:

```
sse = sse.reset_index()
```

In [70]:

```
sse.columns = ['number of clusters','sse']
```

In [71]:

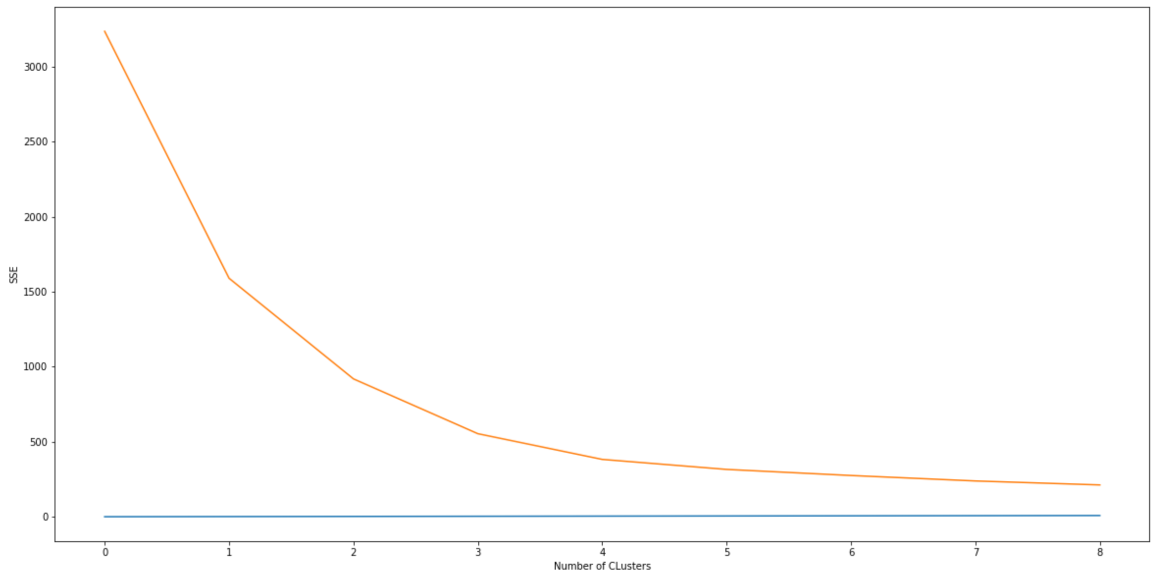
```
sse.to_csv('new_elbow_method.csv')
```

In [73]:

```
plt.figure(figsize = (20,10))
plt.plot(sse)
plt.xlabel('Number of CLusters')
plt.ylabel('SSE')
```

Out[73]:

```
Text(0, 0.5, 'SSE')
```



In [81]:

```
kmeans = KMeans(n_clusters=7)
```

In [82]:

```
kmeans.fit(clus)
```

Out[82]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=7, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

In [83]:

```
centroids = kmeans.cluster_centers_
```

In [84]:

```
centroids
```

Out[84]:

```
array([[ 40.79440476, -73.96697864],
       [ 40.74163794, -73.99223351],
       [ 40.64667703, -73.78472208],
       [ 40.76979509, -73.87088567],
       [ 40.77273968, -73.95533111],
       [ 40.71596576, -73.99742455],
       [ 40.76045088, -73.97931196]])
```

In [85]:

```
centroid = centroids.tolist()
```

In [86]:

```
centroid
```

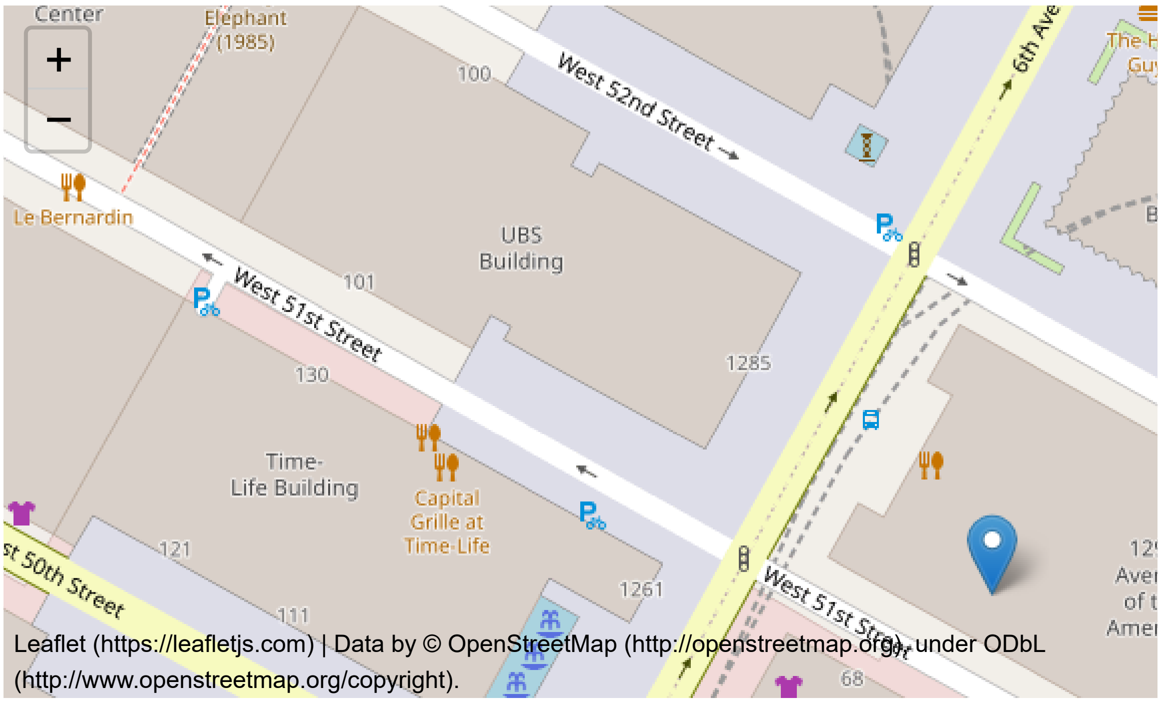
Out[86]:

```
[[40.79440475809989, -73.96697863905337],
 [40.741637941089415, -73.99223350528618],
 [40.64667702945011, -73.78472207788793],
 [40.76979509309178, -73.8708856709822],
 [40.77273967602175, -73.95533110543452],
 [40.71596576089606, -73.99742454966292],
 [40.76045087888758, -73.97931195630005]]
```

In [126]:

```
import folium
# Plotting the centroids on google map using Folium library.
kmeanmap = folium.Map(location=[40.76045087888758, -73.97931195630005], zoom_start = 25)
for point in range(0, len(centroid)):
    folium.Marker(centroid[point], popup = centroid[point]).add_to(kmeanmap)
kmeanmap
```

Out[126]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [90]:

```
train_data['pickup_center'] = kmeans.labels_
```

In [104]:

```
train_data['dropout_center'] = kmeans.predict(train_data[['dropoff_latitude','dropoff_longitude'
]])
```

In [105]:

```
train_data.head()
```

Out[105]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitu
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.9821
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.9804
2	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.9790
3	id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.0100
4	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.9730

In [106]:

```
df7 = df6.copy()
```

In [107]:

```
df7['pickup_center'] = train_data['pickup_center']
df7['dropout_center'] = train_data['dropout_center']
df7.head()
```

Out[107]:

	vendor_id	passenger_count	trip_duration	total_distance	sin_hour	cos_hour	sin_we
0	2	1	455	2009.1	-0.965926	-0.258819	0.000000e+
1	1	1	663	2513.2	0.000000	1.000000	-2.44929
2	2	1	2124	11060.8	0.258819	-0.965926	8.660254e-
3	2	1	429	1779.4	-0.965926	0.258819	8.660254e-
4	2	1	435	1614.9	-0.258819	-0.965926	-8.66025

In [108]:

```
df7 = pd.get_dummies(df7,columns=['pickup_center','dropout_center'])
```

In [109]:

```
y = df7['trip_duration']
X = df7.drop('trip_duration',axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
Xtr, Xv, ytr, yv = train_test_split(X_train, y_train, test_size=0.2)
```

In [110]:

```
df7.head()
```

Out[110]:

	vendor_id	passenger_count	trip_duration	total_distance	sin_hour	cos_hour	sin_we
0	2	1	455	2009.1	-0.965926	-0.258819	0.000000e+
1	1	1	663	2513.2	0.000000	1.000000	-2.44929
2	2	1	2124	11060.8	0.258819	-0.965926	8.660254e
3	2	1	429	1779.4	-0.965926	0.258819	8.660254e
4	2	1	435	1614.9	-0.258819	-0.965926	-8.66025

5 rows × 22 columns

In [102]:

```
reg = LinearRegression()
```

In [113]:

```
reg.fit(X_train.values, y_train)
```

Out[113]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [114]:

```
y_pred = reg.predict(X_test.values)
```

In [116]:

```
reg.intercept_
```

Out[116]:

```
193.75090331404692
```

In [117]:

```
reg.coef_
```

Out[117]:

```
array([-1.54419739e+00,  3.71574231e+00,  1.07547968e-01, -8.86889796e+01,
       -1.45481397e+02,  1.82059808e+01, -6.87053026e+01,  3.60964134e+01,
        8.48644054e+01, -3.32528580e+02,  2.17972208e+01,  2.97867840e+01,
        9.17280399e+01,  6.82557169e+01, -5.15365314e+01,  7.42703580e+01,
       -2.37287714e+02, -1.83734326e+01,  1.80346653e+01,  1.05430908e+02,
        1.09461747e+02])
```

In [115]:

```
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.6514977905992905
RMSE is 387.53389199428193
```

In [118]:

```
dtrain = xgb.DMatrix(Xtr, label=ytr)
dvalid = xgb.DMatrix(Xv, label=yv)
dtest = xgb.DMatrix(X_test)
watchlist = [(dtrain, 'train'), (dvalid, 'valid')]

xgb_pars = {'min_child_weight': 50, 'eta': 0.03, 'colsample_bytree': 0.3, 'max_depth': 10,
            'subsample': 0.8, 'lambda': 1., 'nthread': -1, 'booster' : 'gbtree', 'silent': 1,
            'eval_metric': 'rmse', 'objective': 'reg:linear'}

# You could try to train with more epoch
model = xgb.train(xgb_pars, dtrain, 150, watchlist, early_stopping_rounds=2,
                  maximize=False, verbose_eval=1)
print('Modeling RMSLE %.5f' % model.best_score)
```

```
[0]     train-rmse:1038.42688   valid-rmse:1036.40881
Multiple eval metrics have been passed: 'valid-rmse' will be used for early stoppi
ng.

Will train until valid-rmse hasn't improved in 2 rounds.
[1]     train-rmse:1011.68738   valid-rmse:1009.67444
[2]     train-rmse:992.42188    valid-rmse:990.37305
[3]     train-rmse:966.56073    valid-rmse:964.51611
[4]     train-rmse:948.32068    valid-rmse:946.24091
[5]     train-rmse:924.12579    valid-rmse:922.05786
[6]     train-rmse:906.08783    valid-rmse:904.01959
[7]     train-rmse:890.06091    valid-rmse:887.97809
[8]     train-rmse:873.66931    valid-rmse:871.60187
[9]     train-rmse:858.75531    valid-rmse:856.68762
[10]    train-rmse:843.69708    valid-rmse:841.61951
[11]    train-rmse:827.95178    valid-rmse:825.90448
[12]    train-rmse:815.08850    valid-rmse:813.04724
[13]    train-rmse:801.46759    valid-rmse:799.40656
[14]    train-rmse:788.74780    valid-rmse:786.64154
[15]    train-rmse:769.89783    valid-rmse:767.81830
[16]    train-rmse:758.71338    valid-rmse:756.63117
[17]    train-rmse:745.74323    valid-rmse:743.69080
[18]    train-rmse:729.23669    valid-rmse:727.18646
[19]    train-rmse:713.47186    valid-rmse:711.42981
[20]    train-rmse:703.24847    valid-rmse:701.19763
[21]    train-rmse:693.64551    valid-rmse:691.61041
[22]    train-rmse:685.06274    valid-rmse:683.00677
[23]    train-rmse:670.79663    valid-rmse:668.75256
[24]    train-rmse:662.14020    valid-rmse:660.10883
[25]    train-rmse:654.57751    valid-rmse:652.53882
[26]    train-rmse:641.45697    valid-rmse:639.45404
[27]    train-rmse:634.19019    valid-rmse:632.18256
[28]    train-rmse:627.20508    valid-rmse:625.19324
[29]    train-rmse:614.44244    valid-rmse:612.45062
[30]    train-rmse:602.89001    valid-rmse:600.91943
[31]    train-rmse:596.97662    valid-rmse:595.01697
[32]    train-rmse:591.70294    valid-rmse:589.75470
[33]    train-rmse:581.07989    valid-rmse:579.16180
[34]    train-rmse:575.72199    valid-rmse:573.81732
[35]    train-rmse:570.46130    valid-rmse:568.55273
[36]    train-rmse:564.79779    valid-rmse:562.92389
[37]    train-rmse:560.31738    valid-rmse:558.45886
[38]    train-rmse:550.78809    valid-rmse:548.92902
[39]    train-rmse:541.41058    valid-rmse:539.59582
[40]    train-rmse:537.68121    valid-rmse:535.86713
[41]    train-rmse:532.65137    valid-rmse:530.86188
[42]    train-rmse:529.04974    valid-rmse:527.26910
[43]    train-rmse:520.56445    valid-rmse:518.81360
[44]    train-rmse:517.28284    valid-rmse:515.53784
[45]    train-rmse:509.73920    valid-rmse:508.01697
[46]    train-rmse:506.24963    valid-rmse:504.52902
[47]    train-rmse:502.51031    valid-rmse:500.82153
[48]    train-rmse:495.50723    valid-rmse:493.84860
[49]    train-rmse:491.78934    valid-rmse:490.15115
[50]    train-rmse:485.30869    valid-rmse:483.69788
[51]    train-rmse:482.57263    valid-rmse:480.98309
[52]    train-rmse:479.69571    valid-rmse:478.11020
[53]    train-rmse:477.36063    valid-rmse:475.77499
[54]    train-rmse:474.44076    valid-rmse:472.87436
[55]    train-rmse:468.60730    valid-rmse:467.06747
[56]    train-rmse:466.78717    valid-rmse:465.24884
```

```
[57]	train-rmse:464.28622	valid-rmse:462.76044
[58]	train-rmse:462.44501	valid-rmse:460.91635
[59]	train-rmse:460.42792	valid-rmse:458.89111
[60]	train-rmse:458.29489	valid-rmse:456.76901
[61]	train-rmse:456.82999	valid-rmse:455.31576
[62]	train-rmse:455.39499	valid-rmse:453.90222
[63]	train-rmse:454.12500	valid-rmse:452.64111
[64]	train-rmse:452.82489	valid-rmse:451.35230
[65]	train-rmse:447.96945	valid-rmse:446.55222
[66]	train-rmse:446.26843	valid-rmse:444.87387
[67]	train-rmse:444.59317	valid-rmse:443.23471
[68]	train-rmse:440.18063	valid-rmse:438.85144
[69]	train-rmse:438.52698	valid-rmse:437.22443
[70]	train-rmse:436.87534	valid-rmse:435.60233
[71]	train-rmse:435.79614	valid-rmse:434.54190
[72]	train-rmse:434.55939	valid-rmse:433.33032
[73]	train-rmse:433.10852	valid-rmse:431.87927
[74]	train-rmse:432.15832	valid-rmse:430.93774
[75]	train-rmse:431.30847	valid-rmse:430.09168
[76]	train-rmse:429.72662	valid-rmse:428.53085
[77]	train-rmse:428.92996	valid-rmse:427.75754
[78]	train-rmse:428.20010	valid-rmse:427.03201
[79]	train-rmse:423.72330	valid-rmse:422.61215
[80]	train-rmse:422.69867	valid-rmse:421.58533
[81]	train-rmse:422.17389	valid-rmse:421.06564
[82]	train-rmse:420.91992	valid-rmse:419.82596
[83]	train-rmse:420.13440	valid-rmse:419.05481
[84]	train-rmse:419.44592	valid-rmse:418.37866
[85]	train-rmse:418.68152	valid-rmse:417.61426
[86]	train-rmse:418.28955	valid-rmse:417.23379
[87]	train-rmse:417.04150	valid-rmse:416.00879
[88]	train-rmse:416.45862	valid-rmse:415.44367
[89]	train-rmse:415.79877	valid-rmse:414.79242
[90]	train-rmse:415.02374	valid-rmse:414.02310
[91]	train-rmse:414.40659	valid-rmse:413.43033
[92]	train-rmse:413.89502	valid-rmse:412.91190
[93]	train-rmse:412.84842	valid-rmse:411.87967
[94]	train-rmse:412.06961	valid-rmse:411.11502
[95]	train-rmse:409.03964	valid-rmse:408.12604
[96]	train-rmse:408.28574	valid-rmse:407.38678
[97]	train-rmse:408.00714	valid-rmse:407.11594
[98]	train-rmse:407.64151	valid-rmse:406.75909
[99]	train-rmse:407.22415	valid-rmse:406.34531
[100]	train-rmse:406.88263	valid-rmse:406.01605
[101]	train-rmse:404.16998	valid-rmse:403.34586
[102]	train-rmse:403.76590	valid-rmse:402.95462
[103]	train-rmse:401.06891	valid-rmse:400.31052
[104]	train-rmse:400.75992	valid-rmse:400.00977
[105]	train-rmse:400.03204	valid-rmse:399.28754
[106]	train-rmse:399.79364	valid-rmse:399.05713
[107]	train-rmse:397.35803	valid-rmse:396.66846
[108]	train-rmse:397.12176	valid-rmse:396.44135
[109]	train-rmse:394.16626	valid-rmse:393.54865
[110]	train-rmse:391.79285	valid-rmse:391.21906
[111]	train-rmse:391.38443	valid-rmse:390.82684
[112]	train-rmse:391.14828	valid-rmse:390.60025
[113]	train-rmse:388.99582	valid-rmse:388.49628
[114]	train-rmse:388.68674	valid-rmse:388.19778
[115]	train-rmse:386.71939	valid-rmse:386.27924
[116]	train-rmse:386.48044	valid-rmse:386.05319
[117]	train-rmse:386.16806	valid-rmse:385.73395
```

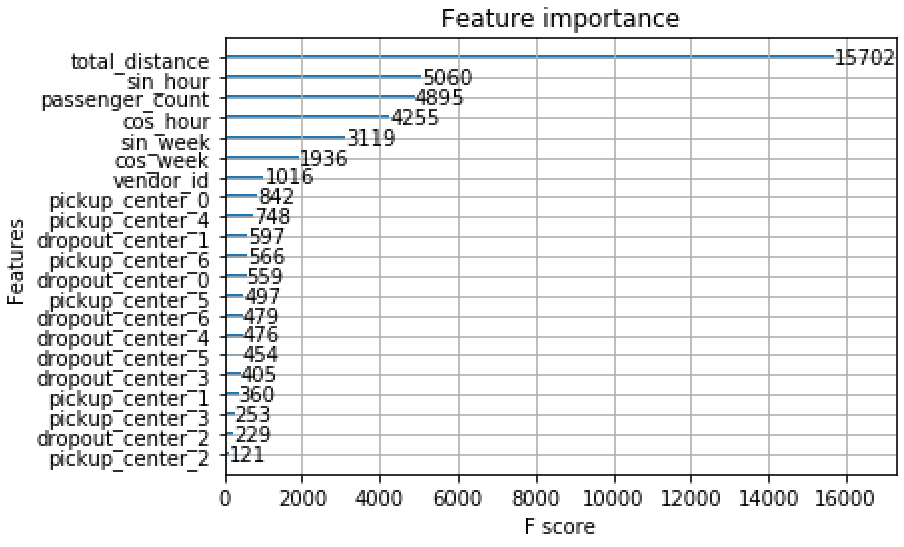
```
[118]	train-rmse:383.76944	valid-rmse:383.40292
[119]	train-rmse:383.56720	valid-rmse:383.20538
[120]	train-rmse:383.14334	valid-rmse:382.78903
[121]	train-rmse:382.95291	valid-rmse:382.60879
[122]	train-rmse:382.76489	valid-rmse:382.41156
[123]	train-rmse:382.60681	valid-rmse:382.25320
[124]	train-rmse:380.88849	valid-rmse:380.58307
[125]	train-rmse:380.34003	valid-rmse:380.03946
[126]	train-rmse:378.66339	valid-rmse:378.41727
[127]	train-rmse:378.28192	valid-rmse:378.05432
[128]	train-rmse:378.13751	valid-rmse:377.91361
[129]	train-rmse:377.98123	valid-rmse:377.76248
[130]	train-rmse:377.65646	valid-rmse:377.44098
[131]	train-rmse:377.35571	valid-rmse:377.15369
[132]	train-rmse:375.49322	valid-rmse:375.34766
[133]	train-rmse:375.31287	valid-rmse:375.17587
[134]	train-rmse:373.52042	valid-rmse:373.42926
[135]	train-rmse:373.20380	valid-rmse:373.12558
[136]	train-rmse:373.02862	valid-rmse:372.96271
[137]	train-rmse:371.32703	valid-rmse:371.32721
[138]	train-rmse:371.26721	valid-rmse:371.27411
[139]	train-rmse:371.17291	valid-rmse:371.18103
[140]	train-rmse:371.05981	valid-rmse:371.06909
[141]	train-rmse:369.77176	valid-rmse:369.81915
[142]	train-rmse:369.56045	valid-rmse:369.59204
[143]	train-rmse:369.23300	valid-rmse:369.28287
[144]	train-rmse:369.04437	valid-rmse:369.10297
[145]	train-rmse:368.85181	valid-rmse:368.91440
[146]	train-rmse:367.68304	valid-rmse:367.77948
[147]	train-rmse:367.31961	valid-rmse:367.42773
[148]	train-rmse:367.07672	valid-rmse:367.19849
[149]	train-rmse:366.95654	valid-rmse:367.08902
Modeling RMSLE 367.08902
```

In [120]:

```
xgb.plot_importance(model)
```

Out[120]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1eebc82db88>
```



In [119]:

```
y_pred = model.predict(dtest)
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.6899016948001107
RMSE is 365.5582742656132
```

In [131]:

```
def haversine_(lat1, lng1, lat2, lng2):
    """function to calculate haversine distance between two co-ordinates"""
    lat1, lng1, lat2, lng2 = map(np.radians, (lat1, lng1, lat2, lng2))
    AVG_EARTH_RADIUS = 6371  # in km
    lat = lat2 - lat1
    lng = lng2 - lng1
    d = np.sin(lat * 0.5) ** 2 + np.cos(lat1) * np.cos(lat2) * np.sin(lng * 0.5) ** 2
    h = 2 * AVG_EARTH_RADIUS * np.arcsin(np.sqrt(d))
    return(h)

def manhattan_distance_pd(lat1, lng1, lat2, lng2):
    """function to calculate manhatten distance between pick_drop"""
    a = haversine_(lat1, lng1, lat1, lng2)
    b = haversine_(lat1, lng1, lat2, lng1)
    return a + b
```

In [132]:

```
train_data.loc[:,'hvsine_pick_drop'] = haversine_(train_data['pickup_latitude'].values, train_da
ta['pickup_longitude'].values, train_data['dropoff_latitude'].values, train_data['dropoff_longit
ude'].values)
train_data.loc[:,'manhtn_pick_drop'] = manhattan_distance_pd(train_data['pickup_latitude'].value
s, train_data['pickup_longitude'].values, train_data['dropoff_latitude'].values, train_data['dro
poff_longitude'].values)
```

In [135]:

```
train_data[['total_distance','hvsine_pick_drop','manhtn_pick_drop']].head(10)
```

Out[135]:

	total_distance	hvsine_pick_drop	manhtn_pick_drop
0	2009.1	1.498521	1.735433
1	2513.2	1.805507	2.430506
2	11060.8	6.385098	8.203575
3	1779.4	1.485498	1.661331
4	1614.9	1.188588	1.199457
5	1393.5	1.098942	1.554180
6	1705.1	1.326279	1.873902
7	10642.0	5.714981	8.078684
8	1310.7	1.310353	1.774804
9	6786.8	5.121162	5.754187

In [136]:

```
df8 = df7.copy()
```

In [137]:

```
df8['hvsine_pick_drop'] = train_data['hvsine_pick_drop']
df8['manhtn_pick_drop'] = train_data['manhtn_pick_drop']
df8.head()
```

Out[137]:

	vendor_id	passenger_count	trip_duration	total_distance	sin_hour	cos_hour	sin_we
0	2	1	455	2009.1	-0.965926	-0.258819	0.000000e+
1	1	1	663	2513.2	0.000000	1.000000	-2.44929
2	2	1	2124	11060.8	0.258819	-0.965926	8.660254e
3	2	1	429	1779.4	-0.965926	0.258819	8.660254e
4	2	1	435	1614.9	-0.258819	-0.965926	-8.66025

5 rows × 24 columns

In [138]:

```
y = df8['trip_duration']
X = df8.drop('trip_duration',axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
Xtr, Xv, ytr, yv = train_test_split(X_train, y_train, test_size=0.2)
```

In [139]:

```
reg = LinearRegression()
reg.fit(X_train.values,y_train)
y_pred = reg.predict(X_test.values)
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.6479566348613686
RMSE is 389.3763257320753
```

In [140]:

```
dtrain = xgb.DMatrix(Xtr, label=ytr)
dvalid = xgb.DMatrix(Xv, label=yv)
dtest = xgb.DMatrix(X_test)
watchlist = [(dtrain, 'train'), (dvalid, 'valid')]

xgb_pars = {'min_child_weight': 50, 'eta': 0.03, 'colsample_bytree': 0.3, 'max_depth': 10,
            'subsample': 0.8, 'lambda': 1., 'nthread': -1, 'booster' : 'gbtree', 'silent': 1,
            'eval_metric': 'rmse', 'objective': 'reg:linear'}

# You could try to train with more epoch
model = xgb.train(xgb_pars, dtrain, 150, watchlist, early_stopping_rounds=2,
                  maximize=False, verbose_eval=1)
print('Modeling RMSLE %.5f' % model.best_score)
```

```
[0]	train-rmse:1038.55615	valid-rmse:1037.90344
Multiple eval metrics have been passed: 'valid-rmse' will be used for early stoppi
ng.

Will train until valid-rmse hasn't improved in 2 rounds.
[1]	train-rmse:1011.78876	valid-rmse:1011.17023
[2]	train-rmse:986.02704	valid-rmse:985.45856
[3]	train-rmse:960.32019	valid-rmse:959.80664
[4]	train-rmse:936.28882	valid-rmse:935.80115
[5]	train-rmse:912.45587	valid-rmse:912.00018
[6]	train-rmse:894.39374	valid-rmse:893.95496
[7]	train-rmse:872.64880	valid-rmse:872.25098
[8]	train-rmse:856.17523	valid-rmse:855.80707
[9]	train-rmse:835.80176	valid-rmse:835.48352
[10]	train-rmse:816.06903	valid-rmse:815.77814
[11]	train-rmse:800.39331	valid-rmse:800.17371
[12]	train-rmse:781.90283	valid-rmse:781.72223
[13]	train-rmse:768.20886	valid-rmse:768.01459
[14]	train-rmse:751.09839	valid-rmse:750.93842
[15]	train-rmse:733.44080	valid-rmse:733.31103
[16]	train-rmse:717.36157	valid-rmse:717.26337
[17]	train-rmse:702.05914	valid-rmse:701.99408
[18]	train-rmse:687.04498	valid-rmse:687.02673
[19]	train-rmse:672.78241	valid-rmse:672.79974
[20]	train-rmse:662.27295	valid-rmse:662.29852
[21]	train-rmse:652.54437	valid-rmse:652.58954
[22]	train-rmse:643.57660	valid-rmse:643.60980
[23]	train-rmse:630.87347	valid-rmse:630.93768
[24]	train-rmse:618.83929	valid-rmse:618.93848
[25]	train-rmse:610.94800	valid-rmse:611.04340
[26]	train-rmse:599.46942	valid-rmse:599.59815
[27]	train-rmse:587.74957	valid-rmse:587.91571
[28]	train-rmse:577.50531	valid-rmse:577.70374
[29]	train-rmse:566.56512	valid-rmse:566.80609
[30]	train-rmse:556.75354	valid-rmse:557.03491
[31]	train-rmse:547.35327	valid-rmse:547.68677
[32]	train-rmse:541.60156	valid-rmse:541.93951
[33]	train-rmse:532.89215	valid-rmse:533.26276
[34]	train-rmse:524.35626	valid-rmse:524.76843
[35]	train-rmse:515.88971	valid-rmse:516.33051
[36]	train-rmse:510.19400	valid-rmse:510.65497
[37]	train-rmse:505.25250	valid-rmse:505.70975
[38]	train-rmse:498.01642	valid-rmse:498.52152
[39]	train-rmse:490.78021	valid-rmse:491.33310
[40]	train-rmse:486.82025	valid-rmse:487.37146
[41]	train-rmse:480.48068	valid-rmse:481.07883
[42]	train-rmse:474.63699	valid-rmse:475.27246
[43]	train-rmse:468.30203	valid-rmse:468.97580
[44]	train-rmse:464.90170	valid-rmse:465.57184
[45]	train-rmse:459.49963	valid-rmse:460.20236
[46]	train-rmse:456.01334	valid-rmse:456.71951
[47]	train-rmse:452.41248	valid-rmse:453.14645
[48]	train-rmse:447.41959	valid-rmse:448.19522
[49]	train-rmse:442.59555	valid-rmse:443.44229
[50]	train-rmse:438.17923	valid-rmse:439.04669
[51]	train-rmse:434.07828	valid-rmse:434.98666
[52]	train-rmse:431.25797	valid-rmse:432.15515
[53]	train-rmse:428.97186	valid-rmse:429.86496
[54]	train-rmse:426.50095	valid-rmse:427.41403
[55]	train-rmse:422.74536	valid-rmse:423.69995
[56]	train-rmse:419.15265	valid-rmse:420.14450
```

```
[57]    train-rmse:416.04245    valid-rmse:417.06399
[58]    train-rmse:414.24762    valid-rmse:415.27191
[59]    train-rmse:412.25168    valid-rmse:413.29294
[60]    train-rmse:409.39050    valid-rmse:410.46902
[61]    train-rmse:406.44666    valid-rmse:407.55447
[62]    train-rmse:404.72095    valid-rmse:405.83496
[63]    train-rmse:403.44122    valid-rmse:404.55170
[64]    train-rmse:401.03091    valid-rmse:402.17276
[65]    train-rmse:398.32980    valid-rmse:399.52338
[66]    train-rmse:396.66315    valid-rmse:397.86804
[67]    train-rmse:393.80939    valid-rmse:395.05139
[68]    train-rmse:391.49756    valid-rmse:392.78366
[69]    train-rmse:389.91971    valid-rmse:391.20047
[70]    train-rmse:387.35944    valid-rmse:388.67877
[71]    train-rmse:384.96777    valid-rmse:386.32166
[72]    train-rmse:383.61905    valid-rmse:384.97492
[73]    train-rmse:381.89688    valid-rmse:383.27496
[74]    train-rmse:380.25916    valid-rmse:381.67450
[75]    train-rmse:378.74719    valid-rmse:380.19336
[76]    train-rmse:377.55841    valid-rmse:379.01837
[77]    train-rmse:376.68692    valid-rmse:378.15585
[78]    train-rmse:375.28146    valid-rmse:376.78516
[79]    train-rmse:372.93802    valid-rmse:374.49783
[80]    train-rmse:372.28281    valid-rmse:373.83582
[81]    train-rmse:371.09662    valid-rmse:372.67673
[82]    train-rmse:369.95459    valid-rmse:371.55951
[83]    train-rmse:368.32974    valid-rmse:369.96542
[84]    train-rmse:367.67084    valid-rmse:369.30936
[85]    train-rmse:366.54776    valid-rmse:368.23672
[86]    train-rmse:366.08566    valid-rmse:367.77878
[87]    train-rmse:364.63028    valid-rmse:366.35358
[88]    train-rmse:363.03677    valid-rmse:364.80844
[89]    train-rmse:362.60931    valid-rmse:364.37491
[90]    train-rmse:362.12393    valid-rmse:363.88986
[91]    train-rmse:360.72067    valid-rmse:362.52652
[92]    train-rmse:359.96570    valid-rmse:361.79510
[93]    train-rmse:359.56024    valid-rmse:361.39545
[94]    train-rmse:358.87244    valid-rmse:360.73047
[95]    train-rmse:358.11197    valid-rmse:359.98758
[96]    train-rmse:357.76071    valid-rmse:359.64169
[97]    train-rmse:357.55911    valid-rmse:359.43973
[98]    train-rmse:357.17273    valid-rmse:359.05646
[99]    train-rmse:356.92947    valid-rmse:358.81165
[100]   train-rmse:356.37521    valid-rmse:358.28738
[101]   train-rmse:355.73608    valid-rmse:357.67291
[102]   train-rmse:354.81534    valid-rmse:356.79034
[103]   train-rmse:354.15491    valid-rmse:356.15344
[104]   train-rmse:353.23566    valid-rmse:355.26532
[105]   train-rmse:352.11307    valid-rmse:354.16330
[106]   train-rmse:351.61871    valid-rmse:353.70801
[107]   train-rmse:350.95148    valid-rmse:353.08520
[108]   train-rmse:350.77078    valid-rmse:352.89865
[109]   train-rmse:349.62500    valid-rmse:351.79678
[110]   train-rmse:348.99448    valid-rmse:351.19699
[111]   train-rmse:348.39368    valid-rmse:350.64581
[112]   train-rmse:347.85525    valid-rmse:350.15704
[113]   train-rmse:347.44006    valid-rmse:349.77438
[114]   train-rmse:347.33414    valid-rmse:349.66458
[115]   train-rmse:346.92334    valid-rmse:349.28226
[116]   train-rmse:346.66480    valid-rmse:349.03159
[117]   train-rmse:346.35989    valid-rmse:348.74890
```

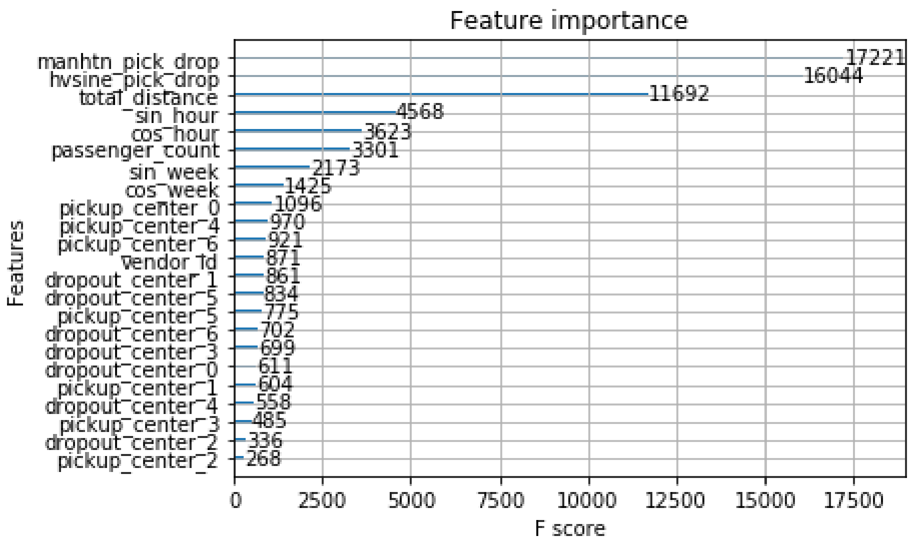
```
[118]	train-rmse:345.39279	valid-rmse:347.83545
[119]	train-rmse:345.29611	valid-rmse:347.74222
[120]	train-rmse:345.06747	valid-rmse:347.51755
[121]	train-rmse:344.99561	valid-rmse:347.44388
[122]	train-rmse:344.72754	valid-rmse:347.20651
[123]	train-rmse:344.65265	valid-rmse:347.13117
[124]	train-rmse:344.31027	valid-rmse:346.81583
[125]	train-rmse:344.18707	valid-rmse:346.69653
[126]	train-rmse:343.93393	valid-rmse:346.47528
[127]	train-rmse:343.66748	valid-rmse:346.22269
[128]	train-rmse:343.58536	valid-rmse:346.13858
[129]	train-rmse:343.37424	valid-rmse:345.95523
[130]	train-rmse:343.14352	valid-rmse:345.74124
[131]	train-rmse:342.87140	valid-rmse:345.50217
[132]	train-rmse:342.24973	valid-rmse:344.90985
[133]	train-rmse:342.13049	valid-rmse:344.79187
[134]	train-rmse:341.77908	valid-rmse:344.46939
[135]	train-rmse:341.47009	valid-rmse:344.16501
[136]	train-rmse:341.38388	valid-rmse:344.07816
[137]	train-rmse:340.91415	valid-rmse:343.64337
[138]	train-rmse:340.87067	valid-rmse:343.60141
[139]	train-rmse:340.82873	valid-rmse:343.56122
[140]	train-rmse:340.56659	valid-rmse:343.33252
[141]	train-rmse:340.32849	valid-rmse:343.12854
[142]	train-rmse:340.18631	valid-rmse:343.00668
[143]	train-rmse:339.60715	valid-rmse:342.47189
[144]	train-rmse:339.45102	valid-rmse:342.32117
[145]	train-rmse:339.10165	valid-rmse:342.00885
[146]	train-rmse:338.98145	valid-rmse:341.91577
[147]	train-rmse:338.56833	valid-rmse:341.53851
[148]	train-rmse:338.03766	valid-rmse:341.05121
[149]	train-rmse:337.70490	valid-rmse:340.74863
Modeling RMSLE 340.74863
```

In [141]:

```
xgb.plot_importance(model)
```

Out[141]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1eec055e308>
```



In [142]:

```
y_pred = model.predict(dtest)
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.7275870603601551
RMSE is 342.51952036342726
```

In [161]:

```
train_data.loc[:, 'center_latitude'] = (train_data['pickup_latitude'].values + train_data['dropo
ff_latitude'].values) / 2
train_data.loc[:, 'center_longitude'] = (train_data['pickup_longitude'].values + train_data['dro
poff_longitude'].values) / 2
```

In [163]:

```
import math
def bearing_array(lat1, lng1, lat2, lng2):
    """ function was taken from beluga's notebook as this function works on array
    while my function used to work on individual elements and was noticably slow"""
    AVG_EARTH_RADIUS = 6371  # in km
    lng_delta_rad = np.radians(lng2 - lng1)
    lat1, lng1, lat2, lng2 = map(np.radians, (lat1, lng1, lat2, lng2))
    y = np.sin(lng_delta_rad) * np.cos(lat2)
    x = np.cos(lat1) * np.sin(lat2) - np.sin(lat1) * np.cos(lat2) * np.cos(lng_delta_rad)
    return np.degrees(np.arctan2(y, x))
```

In [164]:

```
train_data.loc[:,'bearing'] = bearing_array(train_data['pickup_latitude'].values, train_data['pi
ckup_longitude'].values, train_data['dropoff_latitude'].values, train_data['dropoff_longitude'].
values)
```

In [165]:

```
train_data.head()
```

Out[165]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitu
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.9821
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.9804
2	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.9790
3	id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.0100
4	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.9730

5 rows × 25 columns

In [170]:

```
df = train_data.copy()
```

In [171]:

```
y = df['trip_duration']
X = df.drop('trip_duration',axis = 1)
train_df,test_df,train_y,test_y = train_test_split(X,y,test_size=0.2)
```

In [175]:

```
df9 = df8.copy()
```

In [177]:

```
df9['bearing'] = df['bearing']
```

In [180]:

```
y = df9['trip_duration']
X = df9.drop('trip_duration',axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
Xtr, Xv, ytr, yv = train_test_split(X_train, y_train, test_size=0.2)
```

In [181]:

```
reg = LinearRegression()
reg.fit(X_train.values,y_train)
y_pred = reg.predict(X_test.values)
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.6499246049916059
RMSE is 390.5318314085945
```

In [182]:

```
dtrain = xgb.DMatrix(Xtr, label=ytr)
dvalid = xgb.DMatrix(Xv, label=yv)
dtest = xgb.DMatrix(X_test)
watchlist = [(dtrain, 'train'), (dvalid, 'valid')]

xgb_pars = {'min_child_weight': 50, 'eta': 0.03, 'colsample_bytree': 0.3, 'max_depth': 10,
            'subsample': 0.8, 'lambda': 1., 'nthread': -1, 'booster' : 'gbtree', 'silent': 1,
            'eval_metric': 'rmse', 'objective': 'reg:linear'}

# You could try to train with more epoch
model = xgb.train(xgb_pars, dtrain, 150, watchlist, early_stopping_rounds=2,
                  maximize=False, verbose_eval=1)
print('Modeling RMSLE %.5f' % model.best_score)
```

```
[0]	train-rmse:1037.32996	valid-rmse:1036.99341
Multiple eval metrics have been passed: 'valid-rmse' will be used for early stoppi
ng.

Will train until valid-rmse hasn't improved in 2 rounds.
[1]	train-rmse:1010.48492	valid-rmse:1010.18927
[2]	train-rmse:984.57172	valid-rmse:984.33179
[3]	train-rmse:959.19299	valid-rmse:958.98852
[4]	train-rmse:935.16748	valid-rmse:934.99536
[5]	train-rmse:911.31049	valid-rmse:911.21326
[6]	train-rmse:889.08203	valid-rmse:889.03272
[7]	train-rmse:867.01221	valid-rmse:867.00732
[8]	train-rmse:849.36554	valid-rmse:849.42426
[9]	train-rmse:829.16998	valid-rmse:829.28113
[10]	train-rmse:809.61139	valid-rmse:809.75958
[11]	train-rmse:793.85095	valid-rmse:794.06555
[12]	train-rmse:775.48529	valid-rmse:775.76520
[13]	train-rmse:761.60791	valid-rmse:761.95105
[14]	train-rmse:744.34467	valid-rmse:744.72315
[15]	train-rmse:726.74152	valid-rmse:727.21698
[16]	train-rmse:710.51831	valid-rmse:711.07373
[17]	train-rmse:695.30926	valid-rmse:695.91028
[18]	train-rmse:680.44678	valid-rmse:681.11200
[19]	train-rmse:666.26154	valid-rmse:666.97766
[20]	train-rmse:655.71808	valid-rmse:656.48926
[21]	train-rmse:642.50238	valid-rmse:643.32037
[22]	train-rmse:632.35437	valid-rmse:633.23676
[23]	train-rmse:619.69312	valid-rmse:620.65399
[24]	train-rmse:607.90686	valid-rmse:608.92853
[25]	train-rmse:598.72906	valid-rmse:599.83423
[26]	train-rmse:587.48279	valid-rmse:588.65894
[27]	train-rmse:576.01605	valid-rmse:577.25818
[28]	train-rmse:565.65137	valid-rmse:566.96100
[29]	train-rmse:555.38355	valid-rmse:556.75787
[30]	train-rmse:545.76880	valid-rmse:547.23138
[31]	train-rmse:536.64490	valid-rmse:538.15424
[32]	train-rmse:529.86529	valid-rmse:531.43518
[33]	train-rmse:520.49512	valid-rmse:522.15741
[34]	train-rmse:512.04700	valid-rmse:513.79150
[35]	train-rmse:503.46326	valid-rmse:505.31009
[36]	train-rmse:495.37619	valid-rmse:497.28262
[37]	train-rmse:489.68027	valid-rmse:491.64697
[38]	train-rmse:482.69250	valid-rmse:484.71197
[39]	train-rmse:475.80881	valid-rmse:477.90448
[40]	train-rmse:471.50577	valid-rmse:473.66107
[41]	train-rmse:465.50983	valid-rmse:467.73175
[42]	train-rmse:459.95953	valid-rmse:462.23114
[43]	train-rmse:453.90802	valid-rmse:456.23535
[44]	train-rmse:449.93469	valid-rmse:452.31576
[45]	train-rmse:444.83844	valid-rmse:447.27432
[46]	train-rmse:439.69980	valid-rmse:442.19986
[47]	train-rmse:434.43689	valid-rmse:437.00131
[48]	train-rmse:429.89072	valid-rmse:432.54056
[49]	train-rmse:425.44748	valid-rmse:428.17319
[50]	train-rmse:421.44354	valid-rmse:424.22369
[51]	train-rmse:417.66006	valid-rmse:420.50006
[52]	train-rmse:414.68021	valid-rmse:417.56308
[53]	train-rmse:412.40964	valid-rmse:415.32111
[54]	train-rmse:409.96176	valid-rmse:412.90002
[55]	train-rmse:406.39203	valid-rmse:409.40225
[56]	train-rmse:403.13416	valid-rmse:406.20209
```

```
[57]	train-rmse:399.98566	valid-rmse:403.13013
[58]	train-rmse:397.66821	valid-rmse:400.84738
[59]	train-rmse:395.35675	valid-rmse:398.56705
[60]	train-rmse:392.49826	valid-rmse:395.77328
[61]	train-rmse:389.91971	valid-rmse:393.24725
[62]	train-rmse:388.06683	valid-rmse:391.42197
[63]	train-rmse:386.56616	valid-rmse:389.94766
[64]	train-rmse:384.44037	valid-rmse:387.87302
[65]	train-rmse:382.12787	valid-rmse:385.63605
[66]	train-rmse:380.27209	valid-rmse:383.79748
[67]	train-rmse:377.43500	valid-rmse:381.03729
[68]	train-rmse:375.46432	valid-rmse:379.12170
[69]	train-rmse:372.78482	valid-rmse:376.48822
[70]	train-rmse:370.27445	valid-rmse:374.03848
[71]	train-rmse:367.86322	valid-rmse:371.70126
[72]	train-rmse:366.27527	valid-rmse:370.16242
[73]	train-rmse:364.82754	valid-rmse:368.75400
[74]	train-rmse:363.43381	valid-rmse:367.39349
[75]	train-rmse:361.94904	valid-rmse:365.95248
[76]	train-rmse:360.79627	valid-rmse:364.83536
[77]	train-rmse:359.82239	valid-rmse:363.88382
[78]	train-rmse:358.51721	valid-rmse:362.64218
[79]	train-rmse:356.50986	valid-rmse:360.70007
[80]	train-rmse:355.53803	valid-rmse:359.75738
[81]	train-rmse:354.28629	valid-rmse:358.57394
[82]	train-rmse:353.28555	valid-rmse:357.61505
[83]	train-rmse:351.88919	valid-rmse:356.27328
[84]	train-rmse:351.08765	valid-rmse:355.48511
[85]	train-rmse:350.00619	valid-rmse:354.45578
[86]	train-rmse:348.82321	valid-rmse:353.32999
[87]	train-rmse:347.59775	valid-rmse:352.15466
[88]	train-rmse:346.28622	valid-rmse:350.89340
[89]	train-rmse:345.86368	valid-rmse:350.47641
[90]	train-rmse:345.22278	valid-rmse:349.86560
[91]	train-rmse:344.36108	valid-rmse:349.05814
[92]	train-rmse:343.79846	valid-rmse:348.52295
[93]	train-rmse:343.44635	valid-rmse:348.17871
[94]	train-rmse:342.65970	valid-rmse:347.44611
[95]	train-rmse:342.02634	valid-rmse:346.85931
[96]	train-rmse:341.29871	valid-rmse:346.18195
[97]	train-rmse:340.83615	valid-rmse:345.74795
[98]	train-rmse:340.34119	valid-rmse:345.26123
[99]	train-rmse:339.94803	valid-rmse:344.90561
[100]	train-rmse:339.33536	valid-rmse:344.34500
[101]	train-rmse:338.89816	valid-rmse:343.94064
[102]	train-rmse:338.12888	valid-rmse:343.20917
[103]	train-rmse:337.67365	valid-rmse:342.78220
[104]	train-rmse:336.92593	valid-rmse:342.06830
[105]	train-rmse:335.82568	valid-rmse:341.02335
[106]	train-rmse:335.45456	valid-rmse:340.69116
[107]	train-rmse:334.94049	valid-rmse:340.22363
[108]	train-rmse:334.76498	valid-rmse:340.04672
[109]	train-rmse:334.26657	valid-rmse:339.60507
[110]	train-rmse:333.83765	valid-rmse:339.21222
[111]	train-rmse:333.27069	valid-rmse:338.69522
[112]	train-rmse:332.87466	valid-rmse:338.35361
[113]	train-rmse:332.56787	valid-rmse:338.08722
[114]	train-rmse:332.45853	valid-rmse:337.98016
[115]	train-rmse:332.18317	valid-rmse:337.73395
[116]	train-rmse:331.91779	valid-rmse:337.47919
[117]	train-rmse:331.54199	valid-rmse:337.15265
```

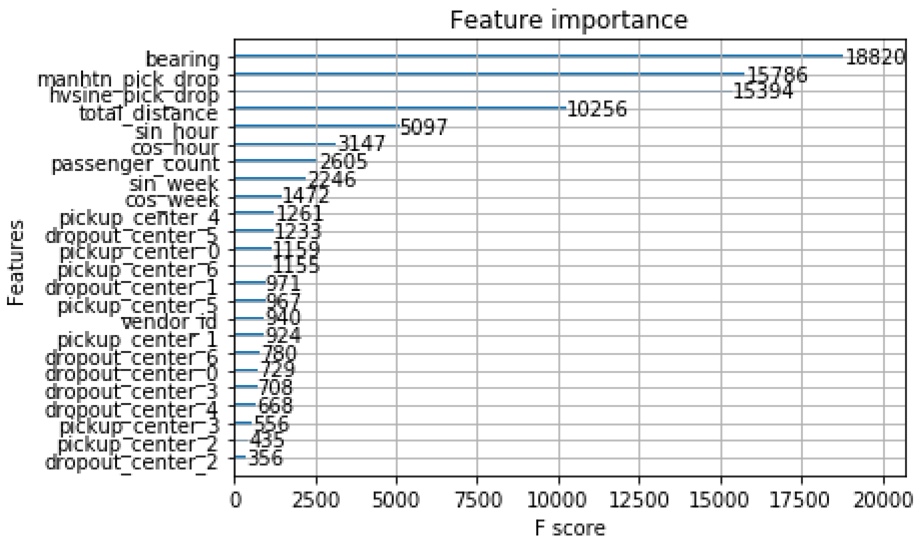
```
[118]	train-rmse:330.70468	valid-rmse:336.37424
[119]	train-rmse:330.60245	valid-rmse:336.26874
[120]	train-rmse:330.39731	valid-rmse:336.08426
[121]	train-rmse:330.31485	valid-rmse:336.00497
[122]	train-rmse:330.11664	valid-rmse:335.83856
[123]	train-rmse:330.04877	valid-rmse:335.77023
[124]	train-rmse:329.80664	valid-rmse:335.56186
[125]	train-rmse:329.63785	valid-rmse:335.41782
[126]	train-rmse:329.45493	valid-rmse:335.25903
[127]	train-rmse:329.08130	valid-rmse:334.89059
[128]	train-rmse:328.91315	valid-rmse:334.74448
[129]	train-rmse:328.72430	valid-rmse:334.58502
[130]	train-rmse:328.59384	valid-rmse:334.46793
[131]	train-rmse:328.40167	valid-rmse:334.31644
[132]	train-rmse:327.68808	valid-rmse:333.64941
[133]	train-rmse:327.41034	valid-rmse:333.41028
[134]	train-rmse:327.21237	valid-rmse:333.24402
[135]	train-rmse:326.85879	valid-rmse:332.89533
[136]	train-rmse:326.57007	valid-rmse:332.65665
[137]	train-rmse:326.02347	valid-rmse:332.15860
[138]	train-rmse:325.90314	valid-rmse:332.06842
[139]	train-rmse:325.79898	valid-rmse:331.98883
[140]	train-rmse:325.58768	valid-rmse:331.82001
[141]	train-rmse:325.33929	valid-rmse:331.60437
[142]	train-rmse:325.26611	valid-rmse:331.54468
[143]	train-rmse:324.74304	valid-rmse:331.06912
[144]	train-rmse:324.57770	valid-rmse:330.91305
[145]	train-rmse:324.23312	valid-rmse:330.61221
[146]	train-rmse:324.15451	valid-rmse:330.55557
[147]	train-rmse:323.67108	valid-rmse:330.11435
[148]	train-rmse:323.36258	valid-rmse:329.84839
[149]	train-rmse:322.90808	valid-rmse:329.42865
Modeling RMSLE 329.42865
```

In [183]:

```
xgb.plot_importance(model)
```

Out[183]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1eec3022bc8>
```



In [184]:

```
y_pred = model.predict(dtest)
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.7478454581973049
RMSE is 331.44314545159654
```

In [186]:

```
df10 = df9.copy()
```

In [190]:

```
df10['pickup_latitude'] = df['pickup_latitude']
df10['pickup_longitude'] = df['pickup_longitude']
df10['dropoff_latitude'] = df['dropoff_latitude']
df10['dropoff_longitude'] = df['dropoff_longitude']
```

In [191]:

```
y = df10['trip_duration']
X = df10.drop('trip_duration',axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
Xtr, Xv, ytr, yv = train_test_split(X_train, y_train, test_size=0.2)
```

In [192]:

```
reg = LinearRegression()
reg.fit(X_train.values,y_train)
y_pred = reg.predict(X_test.values)
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.654126322977458
RMSE is 385.67490776437387
```

In [193]:

```
dtrain = xgb.DMatrix(Xtr, label=ytr)
dvalid = xgb.DMatrix(Xv, label=yv)
dtest = xgb.DMatrix(X_test)
watchlist = [(dtrain, 'train'), (dvalid, 'valid')]

xgb_pars = {'min_child_weight': 50, 'eta': 0.03, 'colsample_bytree': 0.3, 'max_depth': 10,
            'subsample': 0.8, 'lambda': 1., 'nthread': -1, 'booster' : 'gbtree', 'silent': 1,
            'eval_metric': 'rmse', 'objective': 'reg:linear'}

# You could try to train with more epoch
model = xgb.train(xgb_pars, dtrain, 150, watchlist, early_stopping_rounds=2,
                  maximize=False, verbose_eval=1)
print('Modeling RMSLE %.5f' % model.best_score)
```

```
[0]	train-rmse:1038.75476	valid-rmse:1036.70752
Multiple eval metrics have been passed: 'valid-rmse' will be used for early stoppi
ng.

Will train until valid-rmse hasn't improved in 2 rounds.
[1]	train-rmse:1011.65723	valid-rmse:1009.67273
[2]	train-rmse:985.85620	valid-rmse:983.90381
[3]	train-rmse:960.46655	valid-rmse:958.56921
[4]	train-rmse:936.35895	valid-rmse:934.51324
[5]	train-rmse:912.29535	valid-rmse:910.50000
[6]	train-rmse:889.77100	valid-rmse:888.00256
[7]	train-rmse:869.90772	valid-rmse:868.18133
[8]	train-rmse:852.05182	valid-rmse:850.31506
[9]	train-rmse:831.69080	valid-rmse:829.99213
[10]	train-rmse:811.61993	valid-rmse:809.96429
[11]	train-rmse:795.75507	valid-rmse:794.10541
[12]	train-rmse:777.42188	valid-rmse:775.82336
[13]	train-rmse:759.50031	valid-rmse:757.93738
[14]	train-rmse:742.27783	valid-rmse:740.75299
[15]	train-rmse:724.64105	valid-rmse:723.16516
[16]	train-rmse:708.54297	valid-rmse:707.12793
[17]	train-rmse:693.10358	valid-rmse:691.75848
[18]	train-rmse:678.17444	valid-rmse:676.90008
[19]	train-rmse:663.77460	valid-rmse:662.56140
[20]	train-rmse:652.73047	valid-rmse:651.53967
[21]	train-rmse:639.21710	valid-rmse:638.09497
[22]	train-rmse:628.90704	valid-rmse:627.81079
[23]	train-rmse:616.20740	valid-rmse:615.19879
[24]	train-rmse:604.36823	valid-rmse:603.42279
[25]	train-rmse:594.14319	valid-rmse:593.25201
[26]	train-rmse:582.96942	valid-rmse:582.16065
[27]	train-rmse:571.16876	valid-rmse:570.41583
[28]	train-rmse:560.73907	valid-rmse:560.09192
[29]	train-rmse:551.63055	valid-rmse:551.06177
[30]	train-rmse:541.99493	valid-rmse:541.52460
[31]	train-rmse:532.89490	valid-rmse:532.47498
[32]	train-rmse:525.83179	valid-rmse:525.45923
[33]	train-rmse:516.38049	valid-rmse:516.09729
[34]	train-rmse:507.98300	valid-rmse:507.79297
[35]	train-rmse:499.12976	valid-rmse:499.03177
[36]	train-rmse:490.96661	valid-rmse:490.93906
[37]	train-rmse:484.45703	valid-rmse:484.53067
[38]	train-rmse:477.43991	valid-rmse:477.58411
[39]	train-rmse:470.62588	valid-rmse:470.86343
[40]	train-rmse:464.31396	valid-rmse:464.63873
[41]	train-rmse:458.35791	valid-rmse:458.77362
[42]	train-rmse:452.96564	valid-rmse:453.44757
[43]	train-rmse:447.30063	valid-rmse:447.88635
[44]	train-rmse:442.09540	valid-rmse:442.77985
[45]	train-rmse:437.16452	valid-rmse:437.91830
[46]	train-rmse:431.95398	valid-rmse:432.77445
[47]	train-rmse:426.67017	valid-rmse:427.57614
[48]	train-rmse:421.59573	valid-rmse:422.58081
[49]	train-rmse:417.18378	valid-rmse:418.27994
[50]	train-rmse:413.25671	valid-rmse:414.42981
[51]	train-rmse:409.51379	valid-rmse:410.79364
[52]	train-rmse:405.60055	valid-rmse:406.94125
[53]	train-rmse:402.81335	valid-rmse:404.21701
[54]	train-rmse:399.91476	valid-rmse:401.38400
[55]	train-rmse:396.54614	valid-rmse:398.10480
[56]	train-rmse:393.44900	valid-rmse:395.07358
```

```
[57]	train-rmse:390.38672	valid-rmse:392.10736
[58]	train-rmse:387.73093	valid-rmse:389.51752
[59]	train-rmse:385.33914	valid-rmse:387.18454
[60]	train-rmse:382.67209	valid-rmse:384.61514
[61]	train-rmse:380.10770	valid-rmse:382.12546
[62]	train-rmse:377.87711	valid-rmse:379.96347
[63]	train-rmse:376.15817	valid-rmse:378.30252
[64]	train-rmse:374.18594	valid-rmse:376.40320
[65]	train-rmse:372.09186	valid-rmse:374.40305
[66]	train-rmse:370.13293	valid-rmse:372.46420
[67]	train-rmse:367.27576	valid-rmse:369.69647
[68]	train-rmse:365.60150	valid-rmse:368.10092
[69]	train-rmse:362.93912	valid-rmse:365.51764
[70]	train-rmse:360.41141	valid-rmse:363.09656
[71]	train-rmse:358.09613	valid-rmse:360.86145
[72]	train-rmse:356.55807	valid-rmse:359.38419
[73]	train-rmse:355.37811	valid-rmse:358.26852
[74]	train-rmse:353.81644	valid-rmse:356.76123
[75]	train-rmse:352.38312	valid-rmse:355.40250
[76]	train-rmse:350.57236	valid-rmse:353.68561
[77]	train-rmse:349.52637	valid-rmse:352.66949
[78]	train-rmse:348.30981	valid-rmse:351.52353
[79]	train-rmse:346.92719	valid-rmse:350.21701
[80]	train-rmse:345.88300	valid-rmse:349.21362
[81]	train-rmse:344.63605	valid-rmse:348.04712
[82]	train-rmse:343.07712	valid-rmse:346.57190
[83]	train-rmse:341.50577	valid-rmse:345.09311
[84]	train-rmse:340.63217	valid-rmse:344.23535
[85]	train-rmse:339.68665	valid-rmse:343.36310
[86]	train-rmse:338.54956	valid-rmse:342.29901
[87]	train-rmse:337.15607	valid-rmse:340.96625
[88]	train-rmse:335.72873	valid-rmse:339.60992
[89]	train-rmse:335.21100	valid-rmse:339.13617
[90]	train-rmse:334.52606	valid-rmse:338.50897
[91]	train-rmse:333.70056	valid-rmse:337.76294
[92]	train-rmse:333.11987	valid-rmse:337.23721
[93]	train-rmse:332.46011	valid-rmse:336.62894
[94]	train-rmse:331.67505	valid-rmse:335.91153
[95]	train-rmse:331.08145	valid-rmse:335.38000
[96]	train-rmse:330.32785	valid-rmse:334.70450
[97]	train-rmse:329.90152	valid-rmse:334.33481
[98]	train-rmse:329.40134	valid-rmse:333.83948
[99]	train-rmse:328.71063	valid-rmse:333.19031
[100]	train-rmse:328.17117	valid-rmse:332.72195
[101]	train-rmse:327.37033	valid-rmse:331.97153
[102]	train-rmse:327.00494	valid-rmse:331.66199
[103]	train-rmse:326.59296	valid-rmse:331.29764
[104]	train-rmse:325.78348	valid-rmse:330.55664
[105]	train-rmse:324.66251	valid-rmse:329.50760
[106]	train-rmse:324.31934	valid-rmse:329.22983
[107]	train-rmse:323.82284	valid-rmse:328.79727
[108]	train-rmse:323.64856	valid-rmse:328.63303
[109]	train-rmse:323.16794	valid-rmse:328.22839
[110]	train-rmse:322.72775	valid-rmse:327.84302
[111]	train-rmse:322.19397	valid-rmse:327.36780
[112]	train-rmse:321.61108	valid-rmse:326.83630
[113]	train-rmse:321.29721	valid-rmse:326.58389
[114]	train-rmse:321.04904	valid-rmse:326.37463
[115]	train-rmse:320.78571	valid-rmse:326.15817
[116]	train-rmse:320.43399	valid-rmse:325.82959
[117]	train-rmse:320.14182	valid-rmse:325.58188
```

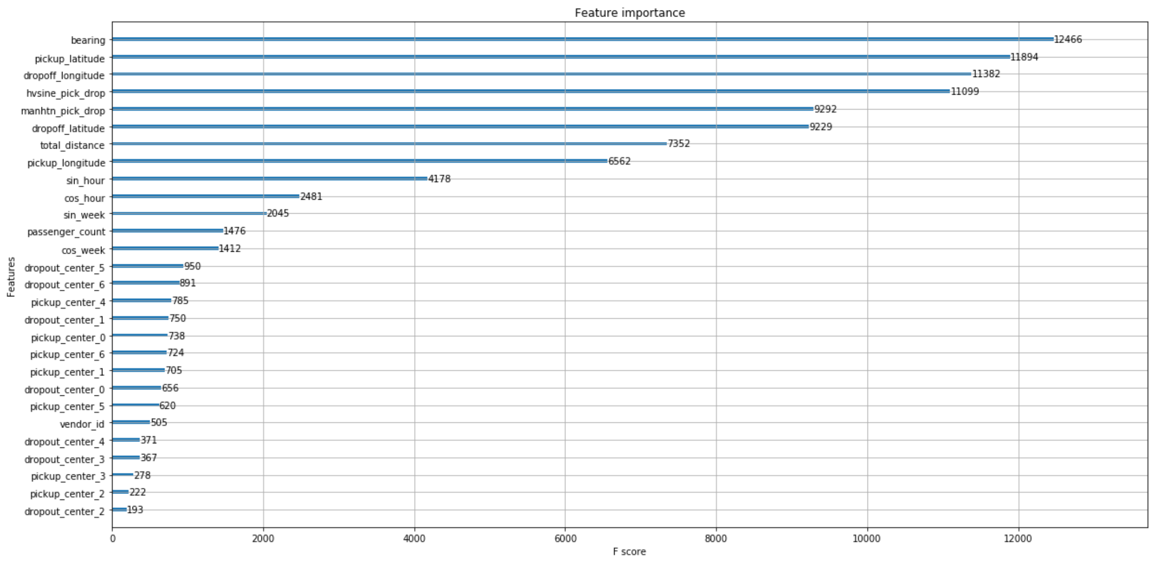
```
[118]	train-rmse:319.81406	valid-rmse:325.32111
[119]	train-rmse:319.69830	valid-rmse:325.23624
[120]	train-rmse:319.44449	valid-rmse:325.02365
[121]	train-rmse:319.24356	valid-rmse:324.86813
[122]	train-rmse:319.08658	valid-rmse:324.74869
[123]	train-rmse:318.98218	valid-rmse:324.67600
[124]	train-rmse:318.70074	valid-rmse:324.45447
[125]	train-rmse:318.43616	valid-rmse:324.23657
[126]	train-rmse:318.05322	valid-rmse:323.92239
[127]	train-rmse:317.52985	valid-rmse:323.43161
[128]	train-rmse:316.89084	valid-rmse:322.83875
[129]	train-rmse:316.64606	valid-rmse:322.65155
[130]	train-rmse:316.49277	valid-rmse:322.54114
[131]	train-rmse:316.15512	valid-rmse:322.27496
[132]	train-rmse:315.71530	valid-rmse:321.90530
[133]	train-rmse:315.42340	valid-rmse:321.67017
[134]	train-rmse:315.16724	valid-rmse:321.47739
[135]	train-rmse:314.47830	valid-rmse:320.84482
[136]	train-rmse:314.21609	valid-rmse:320.63452
[137]	train-rmse:313.61606	valid-rmse:320.09936
[138]	train-rmse:313.51016	valid-rmse:320.02817
[139]	train-rmse:313.36096	valid-rmse:319.91480
[140]	train-rmse:313.20377	valid-rmse:319.80524
[141]	train-rmse:313.04568	valid-rmse:319.68564
[142]	train-rmse:312.93494	valid-rmse:319.60440
[143]	train-rmse:312.42737	valid-rmse:319.16089
[144]	train-rmse:312.30661	valid-rmse:319.07455
[145]	train-rmse:312.02911	valid-rmse:318.84549
[146]	train-rmse:311.90619	valid-rmse:318.76596
[147]	train-rmse:311.45825	valid-rmse:318.36609
[148]	train-rmse:311.21484	valid-rmse:318.17093
[149]	train-rmse:310.89899	valid-rmse:317.89127
Modeling RMSLE 317.89127
```

In [198]:

```
fig, ax = plt.subplots(figsize=(20, 10))
xgb.plot_importance(model,ax = ax)
```

Out[198]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1eec768ebc8>
```



In [199]:

```
y_pred = model.predict(dtest)
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.7670794738896357
RMSE is 316.4947497773846
```

In [215]:

```
df11 = df9.copy()
```

In [218]:

```
df11['pickup_latitude'] = df['pickup_latitude']
df11['pickup_longitude'] = df['pickup_longitude']
df11['dropoff_latitude'] = df['dropoff_latitude']
df11['dropoff_longitude'] = df['dropoff_longitude']
```

In [219]:

```
y = df11['trip_duration']
X = df11.drop('trip_duration',axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
Xtr, Xv, ytr, yv = train_test_split(X_train, y_train, test_size=0.2)
```

In [220]:

```
coords = np.vstack((Xtr[['pickup_latitude', 'pickup_longitude']].values,
                    Xtr[['dropoff_latitude', 'dropoff_longitude']].values,
                    Xv[['pickup_latitude', 'pickup_longitude']].values,
                    Xv[['dropoff_latitude', 'dropoff_longitude']].values,
                    X_test[['pickup_latitude', 'pickup_longitude']].values,
                    X_test[['dropoff_latitude', 'dropoff_longitude']].values))
```

In [221]:

```
pca = PCA().fit(coords)
```

In [222]:

```
Xtr.loc[:,'pickup_pca0'] = pca.transform(Xtr[['pickup_latitude', 'pickup_longitude']])[:, 0]
Xtr.loc[:,'pickup_pca1'] = pca.transform(Xtr[['pickup_latitude', 'pickup_longitude']])[:, 1]
Xtr.loc[:,'dropoff_pca0'] = pca.transform(Xtr[['dropoff_latitude', 'dropoff_longitude']])[:, 0]
Xtr.loc[:,'dropoff_pca1'] = pca.transform(Xtr[['dropoff_latitude', 'dropoff_longitude']])[:, 1]
Xv.loc[:,'pickup_pca0'] = pca.transform(Xv[['pickup_latitude', 'pickup_longitude']])[:, 0]
Xv.loc[:,'pickup_pca1'] = pca.transform(Xv[['pickup_latitude', 'pickup_longitude']])[:, 1]
Xv.loc[:,'dropoff_pca0'] = pca.transform(Xv[['dropoff_latitude', 'dropoff_longitude']])[:, 0]
Xv.loc[:,'dropoff_pca1'] = pca.transform(Xv[['dropoff_latitude', 'dropoff_longitude']])[:, 1]
X_test.loc[:,'pickup_pca0'] = pca.transform(X_test[['pickup_latitude', 'pickup_longitude']])[:,
0]
X_test.loc[:,'pickup_pca1'] = pca.transform(X_test[['pickup_latitude', 'pickup_longitude']])[:,
1]
X_test.loc[:,'dropoff_pca0'] = pca.transform(X_test[['dropoff_latitude', 'dropoff_longitude']])
[:, 0]
X_test.loc[:,'dropoff_pca1'] = pca.transform(X_test[['dropoff_latitude', 'dropoff_longitude']])
[:, 1]
```

```
C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:376: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:494: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:376: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:494: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:376: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:494: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:376: SettingWithC
```

```
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:494: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:376: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:494: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:376: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:494: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:376: SettingWithC
opyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:494: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:376: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:494: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:376: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:494: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:376: SettingWithC
opyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:494: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:376: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:494: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:376: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Lin\Anaconda3\lib\site-packages\pandas\core\indexing.py:494: SettingWithC
opyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy
```

In [228]:

```
dtrain = xgb.DMatrix(Xtr, label=ytr)
dvalid = xgb.DMatrix(Xv, label=yv)
dtest = xgb.DMatrix(X_test)
watchlist = [(dtrain, 'train'), (dvalid, 'valid')]

xgb_pars = {'min_child_weight': 50, 'eta': 0.03, 'colsample_bytree': 0.3, 'max_depth': 10,
            'subsample': 0.8, 'lambda': 1., 'nthread': -1, 'booster' : 'gbtree', 'silent': 1,
            'eval_metric': 'rmse', 'objective': 'reg:linear'}

# You could try to train with more epoch
model = xgb.train(xgb_pars, dtrain, 150, watchlist, early_stopping_rounds=2,
                  maximize=False, verbose_eval=1)
print('Modeling RMSLE %.5f' % model.best_score)
```

```
[0]	train-rmse:1037.54065	valid-rmse:1039.28894
Multiple eval metrics have been passed: 'valid-rmse' will be used for early stoppi
ng.

Will train until valid-rmse hasn't improved in 2 rounds.
[1]	train-rmse:1010.50214	valid-rmse:1012.28870
[2]	train-rmse:984.54218	valid-rmse:986.38403
[3]	train-rmse:959.16711	valid-rmse:961.05054
[4]	train-rmse:934.92657	valid-rmse:936.85113
[5]	train-rmse:910.87390	valid-rmse:912.84058
[6]	train-rmse:888.32941	valid-rmse:890.34851
[7]	train-rmse:868.47766	valid-rmse:870.52350
[8]	train-rmse:848.49731	valid-rmse:850.60303
[9]	train-rmse:827.96545	valid-rmse:830.12579
[10]	train-rmse:807.83447	valid-rmse:810.03461
[11]	train-rmse:788.11615	valid-rmse:790.35773
[12]	train-rmse:769.91309	valid-rmse:772.21735
[13]	train-rmse:751.98376	valid-rmse:754.34088
[14]	train-rmse:734.84503	valid-rmse:737.26898
[15]	train-rmse:717.35175	valid-rmse:719.82263
[16]	train-rmse:701.41589	valid-rmse:703.95551
[17]	train-rmse:685.95453	valid-rmse:688.55566
[18]	train-rmse:671.09552	valid-rmse:673.75940
[19]	train-rmse:656.79804	valid-rmse:659.51495
[20]	train-rmse:642.53601	valid-rmse:645.32019
[21]	train-rmse:629.28583	valid-rmse:632.14142
[22]	train-rmse:617.68616	valid-rmse:620.61188
[23]	train-rmse:605.25702	valid-rmse:608.25214
[24]	train-rmse:594.85370	valid-rmse:597.91065
[25]	train-rmse:584.77417	valid-rmse:587.91443
[26]	train-rmse:573.81415	valid-rmse:577.01459
[27]	train-rmse:562.22467	valid-rmse:565.48883
[28]	train-rmse:552.20312	valid-rmse:555.52741
[29]	train-rmse:543.21222	valid-rmse:546.59741
[30]	train-rmse:533.77991	valid-rmse:537.24823
[31]	train-rmse:524.78485	valid-rmse:528.33081
[32]	train-rmse:517.04755	valid-rmse:520.66431
[33]	train-rmse:507.90179	valid-rmse:511.58173
[34]	train-rmse:499.80338	valid-rmse:503.55978
[35]	train-rmse:491.17764	valid-rmse:495.02914
[36]	train-rmse:483.01068	valid-rmse:486.93991
[37]	train-rmse:476.63797	valid-rmse:480.67148
[38]	train-rmse:469.77155	valid-rmse:473.88040
[39]	train-rmse:463.08160	valid-rmse:467.25284
[40]	train-rmse:456.90771	valid-rmse:461.15445
[41]	train-rmse:450.90427	valid-rmse:455.23163
[42]	train-rmse:445.50088	valid-rmse:449.90713
[43]	train-rmse:439.85800	valid-rmse:444.36282
[44]	train-rmse:434.75568	valid-rmse:439.34189
[45]	train-rmse:429.76172	valid-rmse:434.43176
[46]	train-rmse:423.90561	valid-rmse:428.64023
[47]	train-rmse:418.75793	valid-rmse:423.58240
[48]	train-rmse:413.81799	valid-rmse:418.72043
[49]	train-rmse:409.56064	valid-rmse:414.55051
[50]	train-rmse:405.00250	valid-rmse:410.06598
[51]	train-rmse:401.23023	valid-rmse:406.38345
[52]	train-rmse:397.78397	valid-rmse:403.00708
[53]	train-rmse:394.80972	valid-rmse:400.08817
[54]	train-rmse:391.99069	valid-rmse:397.31192
[55]	train-rmse:388.72125	valid-rmse:394.10886
[56]	train-rmse:385.55878	valid-rmse:391.00937
```

```
[57]	train-rmse:382.51434	valid-rmse:388.04462
[58]	train-rmse:379.99036	valid-rmse:385.58038
[59]	train-rmse:377.56821	valid-rmse:383.21902
[60]	train-rmse:374.92844	valid-rmse:380.64856
[61]	train-rmse:372.13367	valid-rmse:377.92468
[62]	train-rmse:369.96912	valid-rmse:375.82400
[63]	train-rmse:367.79392	valid-rmse:373.72229
[64]	train-rmse:365.93219	valid-rmse:371.93326
[65]	train-rmse:363.90167	valid-rmse:369.96933
[66]	train-rmse:361.50720	valid-rmse:367.63763
[67]	train-rmse:358.86716	valid-rmse:365.07828
[68]	train-rmse:357.20160	valid-rmse:363.48215
[69]	train-rmse:354.73398	valid-rmse:361.08734
[70]	train-rmse:352.36679	valid-rmse:358.79956
[71]	train-rmse:350.67456	valid-rmse:357.18216
[72]	train-rmse:349.08969	valid-rmse:355.67670
[73]	train-rmse:347.98630	valid-rmse:354.64600
[74]	train-rmse:346.31693	valid-rmse:353.04538
[75]	train-rmse:344.77780	valid-rmse:351.57758
[76]	train-rmse:343.09128	valid-rmse:349.96472
[77]	train-rmse:342.02316	valid-rmse:348.92346
[78]	train-rmse:340.82196	valid-rmse:347.78461
[79]	train-rmse:339.51517	valid-rmse:346.54678
[80]	train-rmse:338.47369	valid-rmse:345.55444
[81]	train-rmse:337.32565	valid-rmse:344.47418
[82]	train-rmse:335.87030	valid-rmse:343.08817
[83]	train-rmse:334.38962	valid-rmse:341.66876
[84]	train-rmse:333.17590	valid-rmse:340.51456
[85]	train-rmse:332.25238	valid-rmse:339.65820
[86]	train-rmse:331.20581	valid-rmse:338.67969
[87]	train-rmse:329.86142	valid-rmse:337.39563
[88]	train-rmse:328.52661	valid-rmse:336.12894
[89]	train-rmse:328.01038	valid-rmse:335.65463
[90]	train-rmse:327.36548	valid-rmse:335.04968
[91]	train-rmse:326.59665	valid-rmse:334.34714
[92]	train-rmse:326.04950	valid-rmse:333.84689
[93]	train-rmse:325.36423	valid-rmse:333.22574
[94]	train-rmse:324.80624	valid-rmse:332.70493
[95]	train-rmse:324.26880	valid-rmse:332.22150
[96]	train-rmse:323.62997	valid-rmse:331.64130
[97]	train-rmse:323.22772	valid-rmse:331.27066
[98]	train-rmse:322.59460	valid-rmse:330.71759
[99]	train-rmse:321.74957	valid-rmse:329.92660
[100]	train-rmse:321.20883	valid-rmse:329.44321
[101]	train-rmse:320.38293	valid-rmse:328.67151
[102]	train-rmse:320.03058	valid-rmse:328.35223
[103]	train-rmse:319.64703	valid-rmse:328.00983
[104]	train-rmse:318.96039	valid-rmse:327.37857
[105]	train-rmse:317.73538	valid-rmse:326.22775
[106]	train-rmse:317.33826	valid-rmse:325.90299
[107]	train-rmse:316.91235	valid-rmse:325.54334
[108]	train-rmse:316.65738	valid-rmse:325.32355
[109]	train-rmse:316.18723	valid-rmse:324.91931
[110]	train-rmse:315.83676	valid-rmse:324.60608
[111]	train-rmse:315.36859	valid-rmse:324.19428
[112]	train-rmse:314.84219	valid-rmse:323.71939
[113]	train-rmse:314.55972	valid-rmse:323.47748
[114]	train-rmse:314.31763	valid-rmse:323.27087
[115]	train-rmse:314.14340	valid-rmse:323.12442
[116]	train-rmse:313.52039	valid-rmse:322.56677
[117]	train-rmse:312.89731	valid-rmse:321.99622
```

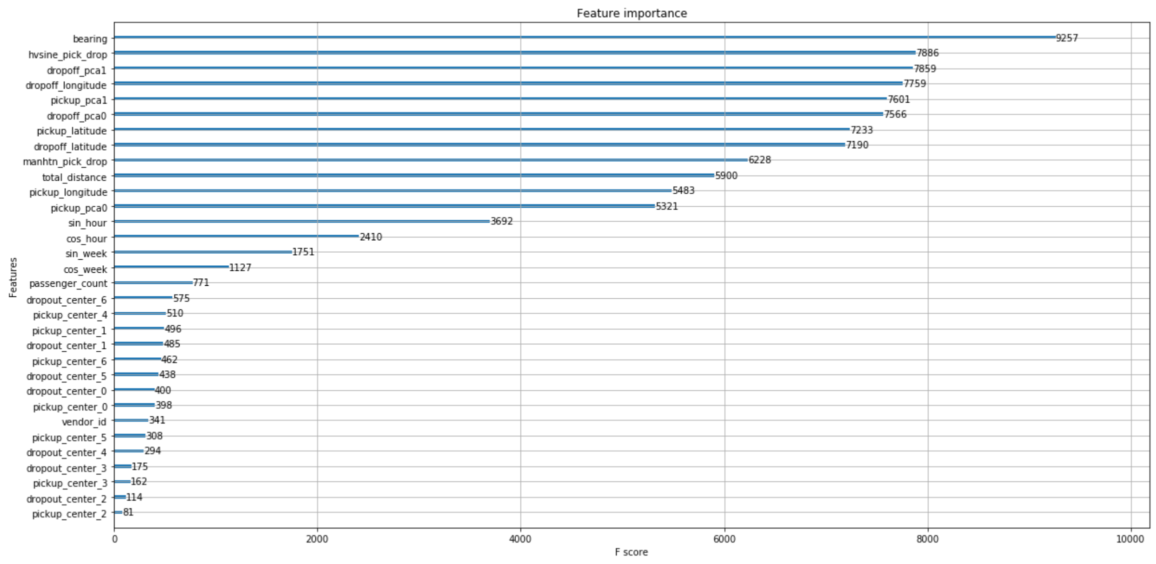
```
[118]	train-rmse:312.58347	valid-rmse:321.73767
[119]	train-rmse:312.38776	valid-rmse:321.57556
[120]	train-rmse:312.06940	valid-rmse:321.30774
[121]	train-rmse:311.91537	valid-rmse:321.18887
[122]	train-rmse:311.66336	valid-rmse:320.99310
[123]	train-rmse:311.52808	valid-rmse:320.88650
[124]	train-rmse:311.39066	valid-rmse:320.77859
[125]	train-rmse:311.18030	valid-rmse:320.60535
[126]	train-rmse:310.96091	valid-rmse:320.43405
[127]	train-rmse:310.49600	valid-rmse:320.01254
[128]	train-rmse:309.89713	valid-rmse:319.46338
[129]	train-rmse:309.70203	valid-rmse:319.31030
[130]	train-rmse:309.54028	valid-rmse:319.18915
[131]	train-rmse:309.30154	valid-rmse:319.00131
[132]	train-rmse:308.96371	valid-rmse:318.70874
[133]	train-rmse:308.72632	valid-rmse:318.51929
[134]	train-rmse:308.52081	valid-rmse:318.37125
[135]	train-rmse:308.04340	valid-rmse:317.94574
[136]	train-rmse:307.80704	valid-rmse:317.75470
[137]	train-rmse:307.28043	valid-rmse:317.28568
[138]	train-rmse:307.12637	valid-rmse:317.17413
[139]	train-rmse:306.92975	valid-rmse:317.01633
[140]	train-rmse:306.69858	valid-rmse:316.83469
[141]	train-rmse:306.52762	valid-rmse:316.68448
[142]	train-rmse:306.09384	valid-rmse:316.30545
[143]	train-rmse:305.67007	valid-rmse:315.92731
[144]	train-rmse:305.57465	valid-rmse:315.86313
[145]	train-rmse:305.37845	valid-rmse:315.70117
[146]	train-rmse:305.27686	valid-rmse:315.63895
[147]	train-rmse:304.87448	valid-rmse:315.28320
[148]	train-rmse:304.70679	valid-rmse:315.15433
[149]	train-rmse:304.34888	valid-rmse:314.85162
Modeling RMSLE 314.85162
```

In [229]:

```
fig, ax = plt.subplots(figsize=(20, 10))
xgb.plot_importance(model,ax = ax)
```

Out[229]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1eee71fa148>
```



In [230]:

```
y_pred = model.predict(dtest)
print('R^2 is',r2_score(y_test, y_pred))
print('RMSE is',np.sqrt(mse(y_test,y_pred)))
```

```
R^2 is 0.7709231800031054
RMSE is 314.2683229097263
```

In [235]:

```
df.columns
```

Out[235]:

```
Index(['id', 'vendor_id', 'pickup_datetime', 'dropoff_datetime',
       'passenger_count', 'pickup_longitude', 'pickup_latitude',
       'dropoff_longitude', 'dropoff_latitude', 'store_and_fwd_flag',
       'trip_duration', 'total_distance', 'total_travel_time', 'pick_month',
       'hour', 'week_of_year', 'day_of_year', 'day_of_week', 'pickup_center',
       'dropout_center', 'hvsine_pick_drop', 'manhtn_pick_drop',
       'center_latitude', 'center_longitude', 'bearing'],
      dtype='object')
```

In [237]:

```
df10.head()
```

Out[237]:

	vendor_id	passenger_count	trip_duration	total_distance	sin_hour	cos_hour	sin_we
0	2	1	455	2009.1	-0.965926	-0.258819	0.000000e+
1	1	1	663	2513.2	0.000000	1.000000	-2.44929
2	2	1	2124	11060.8	0.258819	-0.965926	8.660254e-
3	2	1	429	1779.4	-0.965926	0.258819	8.660254e-
4	2	1	435	1614.9	-0.258819	-0.965926	-8.66025

5 rows × 29 columns

In [239]:

```
train_data = pd.get_dummies(train_data,columns=['pickup_center','dropout_center'])
```

In [240]:

```
train_data.head()
```

Out[240]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitu
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.9821
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.9804
2	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.9790
3	id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.0100
4	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.9730

5 rows × 37 columns

In [242]:

```
train_data['sin_hour'] = df10['sin_hour']
train_data['cos_hour'] = df10['cos_hour']
train_data['sin_week'] = df10['sin_week']
train_data['cos_week'] = df10['cos_week']
```

In [245]:

```
train_data.shape
```

Out[245]:

```
(1450497, 41)
```

In [246]:

```
train_data.to_csv('train_data424.csv')
```

In [ ]:

```

```