```
[yw2983@login-1-1 ~]$ spark-shell --executor-memory 5g --conf spark.driver.args=
"BDAD_Project/2016YellowTaxiData BDAD_Project/CleanWeatherData BDAD_Project/Clea
nTrafficData BDAD_Project/linkinfo.csv" -i code_drop.scala
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 1.6.0
      /_/

Using Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_152)
Type in expressions to have them evaluated.
Type :help for more information.
Spark context available as sc (master = yarn-client, app id = application_152807
7494936_9501).
18/08/02 01:34:55 WARN metastore.ObjectStore: Version information not found in m
etastore. hive.metastore.schema.verification is not enabled so recording the sch
ema version 1.1.0
18/08/02 01:34:55 WARN metastore.ObjectStore: Failed to get database default, re
turning NoSuchObjectException
SQL context available as sqlContext.
Loading code_drop.scala...
args: Array[String] = Array(BDAD_Project/2016YellowTaxiData, BDAD_Project/CleanW
eatherData, BDAD_Project/CleanTrafficData, BDAD_Project/linkinfo.csv)
warning: there were 1 deprecation warning(s); re-run with -deprecation for detai
ls
import org.apache.spark.mllib.tree.GradientBoostedTrees
import org.apache.spark.mllib.tree.configuration.BoostingStrategy
import org.apache.spark.mllib.tree.model.GradientBoostedTreesModel
import org.apache.spark.mllib.util.MLUtils
import org.apache.spark.sql.SQLContext
import org.apache.spark.mllib.linalg.Vectors
import org.apache.spark.mllib.regression.LabeledPoint
import org.apache.spark.mllib.feature.Normalizer
import org.apache.spark.mllib.regression.LinearRegressionModel
import org.apache.spark.mllib.regression.LinearRegressionWithSGD
args: Array[String] = Array(BDAD_Project/2016YellowTaxiData, BDAD_Project/CleanW
eatherData, BDAD_Project/CleanTrafficData, BDAD_Project/linkinfo.csv)
TaxiDataFileName: String = BDAD_Project/2016YellowTaxiData
linkinfoFileName: String = BDAD_Project/linkinfo.csv
cleanWeatherDataFileName: String = BDAD_Project/CleanWeatherData
```

```
cleanTrafficDataFileName: String = BDAD_Project/CleanTrafficData
sqlContext: org.apache.spark.sql.SQLContext = org.apache.spark.sql.SQLContext@19
7d38a
must_train_lr_model_flag: Boolean = false
must_train_gbr_model_flag: Boolean = false
import sqlContext._
import sqlContext.implicits._
csv: org.apache.spark.rdd.RDD[String] = BDAD_Project/2016YellowTaxiData MapParti
tionsRDD[1] at textFile at <console>:49
headerAndRows: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[2] at
map at <console>:51
header: Array[String] = Array(VendorID, tpep_pickup_datetime, tpep_dropoff_datet
ime, passenger_count, trip_distance, pickup_longitude, pickup_latitude, Ratecode
ID, store_and_fwd_flag, dropoff_longitude, dropoff_latitude, payment_type, fare_
amount, extra, mta_tax, tip_amount, tolls_amount, improvement_surcharge, total_a
mount)
data: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[3] at filter at
 <console>:55
tupleData: org.apache.spark.rdd.RDD[(Int, String, String, Int, Double, Double, D
ouble, Int, String, Double, Double, Int, Double, Double, Double, Double, Double,
 Double, Double)] = MapPartitionsRDD[5] at map at <console>:57
df: org.apache.spark.sql.DataFrame = [VendorID: int, tpep_pickup_datetime: strin
g, tpep_dropoff_datetime: string, passenger_count: int, trip_distance: double, p
ickup_longitude: double, pickup_latitude: double, RatecodeID: int, store_and_fwd
_flag: string, dropoff_longitude: double, dropoff_latitude: double, payment_type
: int, fare_amount: double, extra: double, mta_tax: double, tip_amount: double,
tolls_amount: double, improvement_surcharge: double, total_amount: double]
cleanDF: org.apache.spark.sql.DataFrame = [VendorID: int, tpep_pickup_datetime:
string, tpep_dropoff_datetime: string, passenger_count: int, trip_distance: doub
le, pickup_longitude: double, pickup_latitude: double, RatecodeID: int, store_an
d_fwd_flag: string, dropoff_longitude: double, dropoff_latitude: double, payment
_type: int, fare_amount: double, extra: double, mta_tax: double, tip_amount: dou
ble, tolls_amount: double, improvement_surcharge: double, total_amount: double]
selectedCleanDF: org.apache.spark.sql.DataFrame = [VendorID: int, tpep_pickup_da
tetime: string, tpep_dropoff_datetime: string, passenger_count: int, trip_distan
ce: double, pickup_longitude: double, pickup_latitude: double, dropoff_longitude
: double, dropoff_latitude: double]
computeDist: (x1: (Double, Double), x2: (Double, Double))Double
computeMidPoint: (x1: (Double, Double), x2: (Double, Double))(Double, Double)
computeNearestLinkId: (x: (Double, Double), linkArr: Array[(Int, (Double, Double
))])Int
computeDuration: (PUtimeStamp: String, DOtimeStamp: String)Int
```

```
linkCsv: org.apache.spark.rdd.RDD[String] = BDAD_Project/linkinfo.csv MapPartitionsRDD[8] at textFile at <console>:49
head: String = linkId,linkPoints,EncodedPolyLine,EncodedPolyLineLvls,Transcom_id,Borough,linkName,Owner
linkRDD: org.apache.spark.rdd.RDD[(Int, (Double, Double))] = MapPartitionsRDD[16] at map at <console>:55
linkArr: Array[(Int, (Double, Double))] = Array((4616337,(40.75025,-74.005941)), (4616325,(40.734225,-74.0104105)), (4616324,(40.75604,-74.003481)), (4616338,(40.765855200000004,-73.9982605)), (4616323,(40.768905000000004,-73.9958855)), (4616279,(40.63835,-74.20326)), (4616280,(40.621785,-74.17743)), (4616281,(40.52544,-74.24831499999999)), (4456502,(40.693735000000004,-74.010165)), (4616344,(40.7106857,-74.0145705)), (4616345,(40.70624,-74.016005)), (4456501,(40.692675,-74.009800005)), (4616246,(40.8253356,-73.8642755)), (4616260,(40.82544555,-73.864471)), (4456479,(40.79893525,-73.92338050000001)), (4456478,(40.7794358,-73.924701)), (4616342,(40.70855525,-73.99823)), (4616257,(40.69616025,-73.9969955)), (4616339,(40.695235499999995,-73.9979355)), (4616340,(40.7006005,-73.987836)), (46...
taxiRDD: org.apache.spark.rdd.RDD[(Int, String, String, Int, Double, Double, Double, Double, Double)] = MapPartitionsRDD[20] at map at <console>:65
linkedTaxiRDD: org.apache.spark.rdd.RDD[(Int, String, String, Int, Double, Double, Double, Double, Int, Int)] = MapPartitionsRDD[21] at map at <console>:83
labeledTaxiRDD: org.apache.spark.rdd.RDD[(Int, String, String, Int, Double, Double, Double, Double, Int, Int, Int)] = MapPartitionsRDD[23] at filter at <console>:87
labeledTaxiDF: org.apache.spark.sql.DataFrame = [VendorID: int, tpep_pickup_datetime: string, tpep_dropoff_datetime: string, passenger_count: int, trip_distance: double, pickup_longitude: double, pickup_latitude: double, dropoff_longitude: double, dropoff_latitude: double, Pickup_LinkID: int, Dropoff_LinkID: int, Duration: int]
computeWeatherJoinID: (timeStamp: String)String
roundTimeToFive: (timeStamp: String)String
computeTrafficJoinID: (timeStamp: String)String
dirExists: (hdfsDirectory: String)Boolean
finalTaxiRDD: org.apache.spark.rdd.RDD[(String, Int, Double, Double, Double, String, String, String, Int)] = MapPartitionsRDD[25] at map at <console>:93
cleanTrafficData: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[28] at map at <console>:49
finalTrafficRDD: org.apache.spark.rdd.RDD[(String, String)] = MapPartitionsRDD[29] at map at <console>:53
joinPUtrafficData: org.apache.spark.rdd.RDD[(String, ((String, Int, Double, Double, Double, String, String, String, Int), String))] = MapPartitionsRDD[33] at join at <console>:103
joinPUtrafficRDD: org.apache.spark.rdd.RDD[(String, Int, Double, Double, Double,
```

```
joinPUtrafficRDD: org.apache.spark.rdd.RDD[(String, Int, Double, Double, Double,
 String, String, Int, String)] = MapPartitionsRDD[34] at map at <console>:105
joinDOtrafficData: org.apache.spark.rdd.RDD[(String, ((String, Int, Double, Doub
le, Double, String, String, Int, String), String))] = MapPartitionsRDD[38] at jo
in at <console>:107
joinDOtrafficRDD: org.apache.spark.rdd.RDD[(String, Int, Double, Double, Double,
 String, Int, String, String)] = MapPartitionsRDD[39] at map at <console>:109
cleanWeatherData: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[43]
 at filter at <console>:49
WeatherColumns: Array[String] = Array(STATION, DATE, HOURLYVISIBILITY, HOURLYPre
cip, HOURLYWindSpeed)
cleanWeatherRDD: org.apache.spark.rdd.RDD[(String, (Double, Double, Double))] =
MapPartitionsRDD[47] at map at <console>:53
weatherLookupMap: scala.collection.immutable.Map[String,(Double, Double, Double)
] = Map(4/22/16 13:51 -> (8.0,0.0,7.0), 3/10/16 3:51 -> (10.0,0.0,8.0), 2/29/16
16:51 -> (10.0,0.0,9.0), 1/30/16 18:51 -> (10.0,0.0,13.0), 5/22/16 15:51 -> (10.
0,0.0,3.0), 4/10/16 2:51 -> (10.0,0.0,-1.0), 2/26/16 18:51 -> (10.0,0.0,8.0), 2/
3/16 10:51 -> (10.0,0.0,6.0), 3/15/16 12:51 -> (10.0,0.0,0.0), 3/18/16 16:51 ->
(10.0,0.0,6.0), 2/15/16 23:44 -> (2.0,0.07,3.0), 3/5/16 2:51 -> (10.0,0.0,7.0),
2/8/16 12:28 -> (6.0,0.01,10.0), 1/31/16 8:51 -> (10.0,0.0,8.0), 1/20/16 19:51 -
> (10.0,0.0,5.0), 1/18/16 13:51 -> (10.0,0.0,10.0), 1/23/16 9:51 -> (0.25,0.17,2
3.0), 6/21/16 18:51 -> (8.0,0.0,0.0), 4/1/16 18:51 -> (10.0,0.01,3.0), 4/7/16 6:
51 -> (10.0,0.0,13.0), 6/25/16 5:51 -> (8.0,0.0,5.0), 5/20/16 9:51 -> (10.0,0...
lookupWeather: (date: String, map: Map[String,(Double, Double, Double)])(Double,
 Double, Double)
TaxiTrafficWeatherData: org.apache.spark.rdd.RDD[(String, Int, Double, Double, D
ouble, Int, String, String, (Double, Double, Double))] = MapPartitionsRDD[48] at
 map at <console>:123
TaxiTrafficWeatherRDD: org.apache.spark.rdd.RDD[(String, Int, Double, Double, Do
uble, Int, Double, Double, Double, Double, Double)] = MapPartitionsRDD[49] at ma
p at <console>:125
saveTTWRDD: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[50] at map at <c
onsole>:127
TaxiTrafficWeatherDF: org.apache.spark.sql.DataFrame = [Pickup_Datetime: string,
 Passanger_Count: int, Trip_Distance: double, Pickup_Longitude: double, Pickup_L
atitude: double, Trip_Duration: int, Pickup_Location_Traffic_Speed: double, Drop
off_Location_Traffic_Speed: double, Visibility: double, Precipitation: double, W
ind_Speed: double]
TrainDF: org.apache.spark.sql.DataFrame = [Passanger_Count: int, Trip_Distance:
double, Pickup_Location_Traffic_Speed: double, Dropoff_Location_Traffic_Speed: d
ouble, Visibility: double, Precipitation: double, Wind_Speed: double, Trip_Durat
ion: int]
```

```
labeledPointRDD_lr: org.apache.spark.rdd.RDD[org.apache.spark.mllib.regression.L
abeledPoint] = MapPartitionsRDD[55] at map at <console>:131
train_lr_model: (modelFileName: String, flag: Boolean)org.apache.spark.mllib.reg
ression.LinearRegressionModel
lr_model: org.apache.spark.mllib.regression.LinearRegressionModel = org.apache.s
park.mllib.regression.LinearRegressionModel: intercept = 0.0, numFeatures = 2
labelsAndPreds_lr: org.apache.spark.rdd.RDD[(Double, Double)] = MapPartitionsRDD
[63] at map at <console>:141
labeledPointRDD_gbr: org.apache.spark.rdd.RDD[org.apache.spark.mllib.regression.
LabeledPoint] = MapPartitionsRDD[69] at repartition at <console>:131
train_gbr_model: (modelFileName: String, flag: Boolean)org.apache.spark.mllib.tr
ee.model.GradientBoostedTreesModel
model: org.apache.spark.mllib.tree.model.GradientBoostedTreesModel =
TreeEnsembleModel regressor with 3 trees

labelsAndPreds_gbr: org.apache.spark.rdd.RDD[(Double, Double)] = MapPartitionsRD
D[80] at map at <console>:141
computeRMSLE: (labelsAndPreds: org.apache.spark.rdd.RDD[(Double, Double)])Double
LabelAndPreds: org.apache.spark.rdd.RDD[(Double, Double, Double)] = MapPartition
sRDD[81] at map at <console>:149
labelAndPredsDF: org.apache.spark.sql.DataFrame = [Real Label: double, GBR Predi
ct Label: double, LR Predict Label: double]
a: Double = 0.7077772347025818
b: Double = 1.8860337017967405E-4
```

Final Trained Gradient Boosted Trees Model

1. Tree 0:

```
gradient boosted regression model:
tree weights: 1.0 0.1 0.1
TreeEnsembleModel regressor with 3 trees

  Tree 0:
    If (feature 0 <= 2101.0)
     If (feature 0 <= 773.0)
      If (feature 0 <= 420.0)
       If (feature 0 <= 266.0)
        If (feature 0 <= 164.0)
         Predict: 115.80328257459068
        Else (feature 0 > 164.0)
         Predict: 217.53368108453333
       Else (feature 0 > 266.0)
        If (feature 0 <= 350.0)
         Predict: 308.32323877699747
        Else (feature 0 > 350.0)
         Predict: 383.38775609306816
      Else (feature 0 > 420.0)
       If (feature 0 <= 582.0)
        If (feature 0 <= 504.0)
         Predict: 462.4095005552242
        Else (feature 0 > 504.0)
         Predict: 543.3749280017119
       Else (feature 0 > 582.0)
        If (feature 0 <= 665.0)
         Predict: 623.3571345033337
        Else (feature 0 > 665.0)
         Predict: 718.746214448774
     Else (feature 0 > 773.0)
      If (feature 0 <= 1239.0)
       If (feature 0 <= 1033.0)
        If (feature 0 <= 870.0)
         Predict: 825.4154428768626
        Else (feature 0 > 870.0)
         Predict: 946.6972294519978
       Else (feature 0 > 1033.0)
        If (feature 0 <= 1131.0)
         Predict: 1081.7396885170594
        Else (feature 0 > 1131.0)
         Predict: 1181.9194283057734
      Else (feature 0 > 1239.0)
```

```
   If (feature 0 <= 1623.0)
    If (feature 0 <= 1405.0)
     Predict: 1321.0211562127117
    Else (feature 0 > 1405.0)
     Predict: 1503.3198727183287
   Else (feature 0 > 1623.0)
    If (feature 2 <= 11.56)
     Predict: 1807.135709273185
    Else (feature 2 > 11.56)
     Predict: 1880.3036696516972
Else (feature 0 > 2101.0)
 If (feature 2 <= 2.69)
  If (feature 4 <= 6.21)
   If (feature 3 <= 22.99)
    If (feature 2 <= 1.63)
     Predict: 47334.879975874544
    Else (feature 2 > 1.63)
     Predict: 6179.514184397163
   Else (feature 3 > 22.99)
    If (feature 2 <= 2.08)
     Predict: 8189.500225784602
    Else (feature 2 > 2.08)
     Predict: 2669.4454843557864
  Else (feature 4 > 6.21)
   If (feature 2 <= 1.35)
    If (feature 4 <= 21.75)
     Predict: 70958.04959276828
    Else (feature 4 > 21.75)
     Predict: 84207.80105255966
   Else (feature 2 > 1.35)
    If (feature 2 <= 1.63)
     Predict: 26835.886989462153
    Else (feature 2 > 1.63)
     Predict: 68243.45247199317
 Else (feature 2 > 2.69)
  If (feature 2 <= 4.04)
   If (feature 1 <= 2.0)
    If (feature 4 <= 41.01)
     Predict: 3974.6016789014993
    Else (feature 4 > 41.01)
     Predict: 17710.25901042898
   Else (feature 1 > 2.0)
    If (feature 7 <= -1.0)
```

```
   Predict: 82729.15756813838
  Else (feature 7 > -1.0)
   Predict: 6234.970747892701
Else (feature 2 > 4.04)
 If (feature 2 <= 5.09)
  If (feature 4 <= 47.85)
   Predict: 4274.785409995188
  Else (feature 4 > 47.85)
   Predict: 56892.658333704014
 Else (feature 2 > 5.09)
  If (feature 1 <= 5.0)
   Predict: 2945.9197095210775
  Else (feature 1 > 5.0)
   Predict: 4415.096349933305
```

2. Tree 1:

```
Tree 1:
  If (feature 1 <= 2.0)
   If (feature 0 <= 2126.0)
    If (feature 0 <= 1644.0)
     If (feature 0 <= 122.0)
      If (feature 3 <= 3.73)
       Predict: 4.605666996414682
      Else (feature 3 > 3.73)
       Predict: -61.35492684110071
     Else (feature 0 > 122.0)
      If (feature 0 <= 1412.0)
       Predict: 2.473622942765018
      Else (feature 0 > 1412.0)
       Predict: -22.388898366965808
    Else (feature 0 > 1644.0)
     If (feature 2 <= 2.79)
      If (feature 5 <= 9.0)
       Predict: -1547.7537201945172
      Else (feature 5 > 9.0)
       Predict: -474.4254658614791
     Else (feature 2 > 2.79)
      If (feature 7 <= -1.0)
       Predict: -182.48477012463965
      Else (feature 7 > -1.0)
       Predict: -13.004592910794827
   Else (feature 0 > 2126.0)
    If (feature 2 <= 2.79)
     If (feature 3 <= 8.7)
      If (feature 2 <= 2.1)
       Predict: -29411.614759994605
      Else (feature 2 > 2.1)
       Predict: -113023.16524112491
     Else (feature 3 > 8.7)
      If (feature 3 <= 11.18)
       Predict: 29676.57384074846
      Else (feature 3 > 11.18)
       Predict: -10039.897377102267
    Else (feature 2 > 2.79)
     If (feature 2 <= 3.18)
      If (feature 3 <= 45.98)
       Predict: 861.2361276698035
      Else (feature 3 > 45.98)
       Predict: 104977.78475107397
```

```
   Else (feature 2 > 3.18)
    If (feature 7 <= 9.0)
     Predict: -195.8363990339416
    Else (feature 7 > 9.0)
     Predict: 1968.7858935515628
Else (feature 1 > 2.0)
 If (feature 0 <= 2126.0)
  If (feature 0 <= 1412.0)
   If (feature 0 <= 197.0)
    If (feature 0 <= 173.0)
     Predict: -12.176402874464552
    Else (feature 0 > 173.0)
     Predict: -64.7006684566424
   Else (feature 0 > 197.0)
    If (feature 0 <= 268.0)
     Predict: 25.39113075416679
    Else (feature 0 > 268.0)
     Predict: -1.3692832604431808
  Else (feature 0 > 1412.0)
   If (feature 3 <= 11.81)
    If (feature 2 <= 1.29)
     Predict: -2826.348168516667
    Else (feature 2 > 1.29)
     Predict: -206.48436690810746
   Else (feature 3 > 11.81)
    If (feature 2 <= 2.52)
     Predict: -245.63045701724738
    Else (feature 2 > 2.52)
     Predict: -19.228977926387945
 Else (feature 0 > 2126.0)
  If (feature 2 <= 2.52)
   If (feature 2 <= 1.39)
    If (feature 4 <= 11.18)
     Predict: 29256.74904043108
    Else (feature 4 > 11.18)
     Predict: 1020.0192479673601
   Else (feature 2 > 1.39)
    If (feature 2 <= 1.69)
     Predict: 113497.91045605353
    Else (feature 2 > 1.69)
     Predict: 23828.12592847087
  Else (feature 2 > 2.52)
   If (feature 2 <= 2.79)
    If (feature 3 <= 42.25)
     Predict: -379.8379767151863
    Else (feature 3 > 42.25)
     Predict: 2712.760389642381
```

3. Tree 2:

```
Tree 2:
  If (feature 1 <= 2.0)
   If (feature 0 <= 2137.0)
    If (feature 0 <= 1655.0)
     If (feature 0 <= 119.0)
      If (feature 2 <= 0.79)
       Predict: -32.78821823549921
      Else (feature 2 > 0.79)
       Predict: -167.28101424744517
     Else (feature 0 > 119.0)
      If (feature 0 <= 167.0)
       Predict: 39.47175149326943
      Else (feature 0 > 167.0)
       Predict: -0.5952465339800738
    Else (feature 0 > 1655.0)
     If (feature 2 <= 2.79)
      If (feature 3 <= 3.73)
       Predict: -5819.86988104627
      Else (feature 3 > 3.73)
       Predict: -755.5157993640438
     Else (feature 2 > 2.79)
      If (feature 4 <= 26.72)
       Predict: 27.097444615271293
      Else (feature 4 > 26.72)
       Predict: -79.84561389664596
   Else (feature 0 > 2137.0)
    If (feature 2 <= 1.8)
     If (feature 2 <= 1.45)
      If (feature 7 <= 3.0)
       Predict: 4950.1066318748335
      Else (feature 7 > 3.0)
       Predict: -12205.751165614429
     Else (feature 2 > 1.45)
      If (feature 4 <= 42.87)
       Predict: -41785.58829444251
      Else (feature 4 > 42.87)
       Predict: 41976.06848063964
    Else (feature 2 > 1.8)
     If (feature 3 <= 4.97)
      If (feature 2 <= 2.79)
       Predict: -68221.37415086041
      Else (feature 2 > 2.79)
       Predict: -19.060597493782435
```

```
   Else (feature 3 > 4.97)
    If (feature 2 <= 2.3)
     Predict: 10879.592672334777
    Else (feature 2 > 2.3)
     Predict: -61.885357001298296
Else (feature 1 > 2.0)
 If (feature 0 <= 2137.0)
  If (feature 0 <= 1655.0)
   If (feature 0 <= 224.0)
    If (feature 0 <= 167.0)
     Predict: 1.2327410043167732
    Else (feature 0 > 167.0)
     Predict: -46.075508690651716
   Else (feature 0 > 224.0)
    If (feature 0 <= 268.0)
     Predict: 50.13886773003391
    Else (feature 0 > 268.0)
     Predict: -2.545069040246212
  Else (feature 0 > 1655.0)
   If (feature 2 <= 2.79)
    If (feature 7 <= 3.0)
     Predict: -395.7338036011728
    Else (feature 7 > 3.0)
     Predict: -1810.3047008669935
   Else (feature 2 > 2.79)
    If (feature 3 <= 14.91)
     Predict: -195.79443953117718
    Else (feature 3 > 14.91)
     Predict: -7.428571959542901
 Else (feature 0 > 2137.0)
  If (feature 2 <= 2.5)
   If (feature 2 <= 1.39)
    If (feature 4 <= 9.94)
     Predict: 23598.227704838253
    Else (feature 4 > 9.94)
     Predict: 868.7011937626667
   Else (feature 2 > 1.39)
    If (feature 2 <= 1.69)
     Predict: 90911.32490154149
    Else (feature 2 > 1.69)
     Predict: 19537.355746470454
  Else (feature 2 > 2.5)
   If (feature 2 <= 2.79)
    If (feature 4 <= 49.09)
     Predict: 985.1163681947523
    Else (feature 4 > 49.09)
     Predict: -1761.8516021941564
```

4. Real label and Predict label of GBR model and LR model

```
linear regression model's RMSLE: 0.34451972214011983
gradient boosted regression model's RMSLE: 0.15093662004590216
10 sample of real label and predict label:
+----------+-----------------+------------------+
|Real Label|GBR Predict Label|  LR Predict Label|
+----------+-----------------+------------------+
|     273.0| 308.511076417876|193.22331709616398|
|     536.0|543.5627656425903| 379.3691824710314|
|     669.0|718.9340520896525|473.50330950209354|
|    2044.0|1797.850688592441|1446.6980256763425|
|     684.0|718.9340520896525| 484.1201189053284|
|     435.0|462.5973381961027|307.88361386885737|
|    1748.0|1797.850688592441| 1237.196579051365|
|    1442.0|1501.021458228234|1020.6165830334767|
|    1026.0|946.8850670928763| 726.1801783579926|
|     911.0|946.3057942219289| 644.7858868968134|
+----------+-----------------+------------------+
```

5. Where to find the input data

2016YellowTaxiData is on hdfs:///user/yw2983/BDAD_Project/2016YellowTaxiData
If you have trouble to access it please contact yw2983@nyu.edu

linkinfo.scv is on hdfs:///user/yw2983/BDAD_Project/linkinfo.scv
If you have trouble to access it please contact yw2983@nyu.edu

CleanWeatherData is on hdfs:///user/yw2983/BDAD_Project/CleanWeatherData
If you have trouble to access it please contact biqi.lin@nyu.edu

CleanTrafficData is on hdfs:///user/yw2983/BDAD_Project/CleanTrafficData
If you have trouble to access it please contact sy1144@nyu.edu