# **PROJECT-INSTANT CARS**

**PROBLEM STATEMENT:**

A US cab service startup (i.e. Instant cabs) wants to meet the demands in an optimum manner and maximize the profit. Thus, they hired you as a data analyst to interpret the available Uber’s data set and find out the beehive customer pickup points & peak hours for meeting the demand in a profitable manner.

**STRATEGY:**

1. Storage -> The UBER data set I copied from local filesystem to HDFS. Where the HDFS is used as storage system. But here I used local file system.

We can do it by using mini hdfs cluster. I will be mentioning the mini hdfs cluster class in end of document.

1. Processing Framework -> Where Spark is used as processing framework and importing all requires libraries
2. Spark SQl -> Spark SQL is used for querying the data set and give results in tabular format and filter some of the records.
3. Spark Mlib -> Spark Mlib is used for using machine learning algorithms on data set for further analysis.
4. Zeppelin -> where it is framework used to visualize the intermediate and final results.

**Code:**

import org.apache.spark.ml.clustering.KMeans

import org.apache.spark.ml.feature.VectorAssembler

import org.apache.spark.{SparkConf, SparkContext}

import org.apache.spark.sql.SQLContext

import org.apache.spark.sql.types.{DoubleType, StringType, StructField, StructType}

val master\_url="local[2]"

val temp\_dir="C:\\Users\\MUKHESH\\OneDrive\\Documents\\Temp\_Dir"

val app\_name="Instant\_cars"

var sc:SparkContext=null

val conf=new SparkConf().setAppName(app\_name).setMaster(master\_url).set("spark.sql.shuffle.partitions","2").set("spark.executor.memory","2g")

sc=SparkContext.getOrCreate(conf)

val spark=new SQLContext(sc)

import spark.implicits.\_

val schema= StructType(Array(StructField("hr",StringType,true),StructField("lat",DoubleType,true),StructField("lon",DoubleType,true),StructField("Base",StringType,true)))

val instant\_cars=spark.read.format("csv").option("header","true").schema(schema).load("D:\\Edureka\\Edureka share folders\\uber-data.csv")

instant\_cars.registerTempTable("instant")

val feature\_col=Array("lat","lon")

val assember= new VectorAssembler().setInputCols(feature\_col).setOutputCol("features")

val tranformed\_data=assember.setHandleInvalid("skip").transform(instant\_cars)

val kmeans=new KMeans().setK(8).setFeaturesCol("features").setPredictionCol("predictions")

val model=kmeans.fit(tranformed\_data)

model.clusterCenters.foreach(println)

val categories=model.transform(tranformed\_data)

categories.registerTempTable("categories")

val count=spark.sql("SELECT hr as hour,predictions,count(predictions) as count from categories group by hr,predictions")

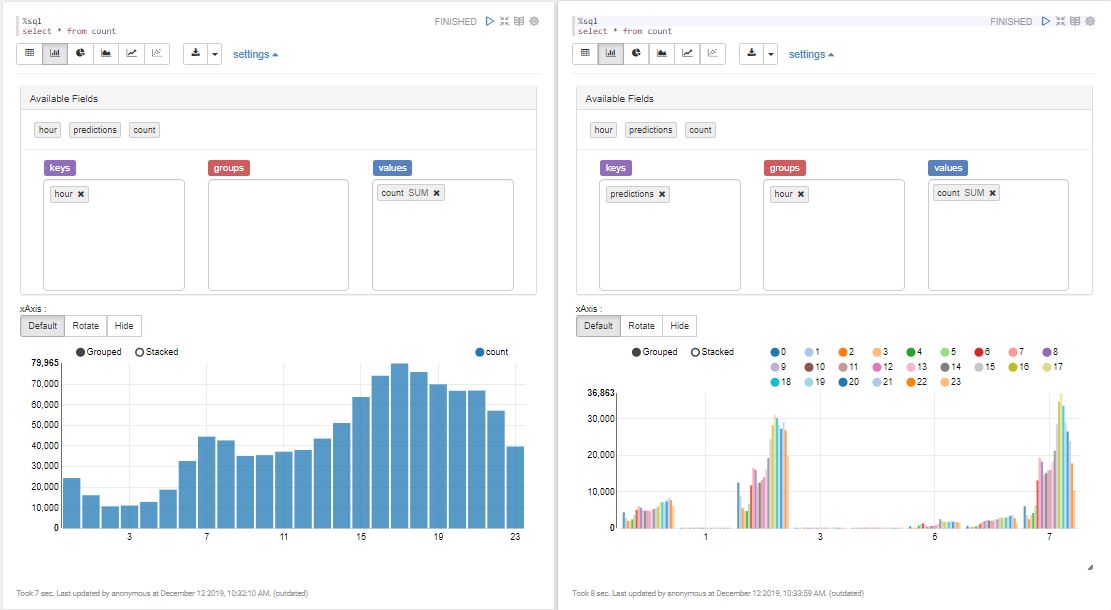
count.registerTempTable("count")

**Some Analysis:**

Where after applying the kmeans clustering with elbow method we got 8 clusters centres which represent the latitude and longitude.

[40.68662777183505,-73.96458362934189] [40.99631659192824,-73.77457814920503] [40.72876468369078,-74.00406323002235] [40.83740506993008,-73.28840874125872] [40.749106666666684,-73.63107980620153] [40.6545255172663,-73.78691747868042] [40.78489695818958,-73.88496691361628] [40.76404321141603,-73.97451451467461]

Analysis says that evening time the demand is higher. We can visualize those things with plots.



And also at which location the rides were concentrated we can get the scatter plot. As we can see that most are contentrated at longitude 74 and latitude spread across 40-41

