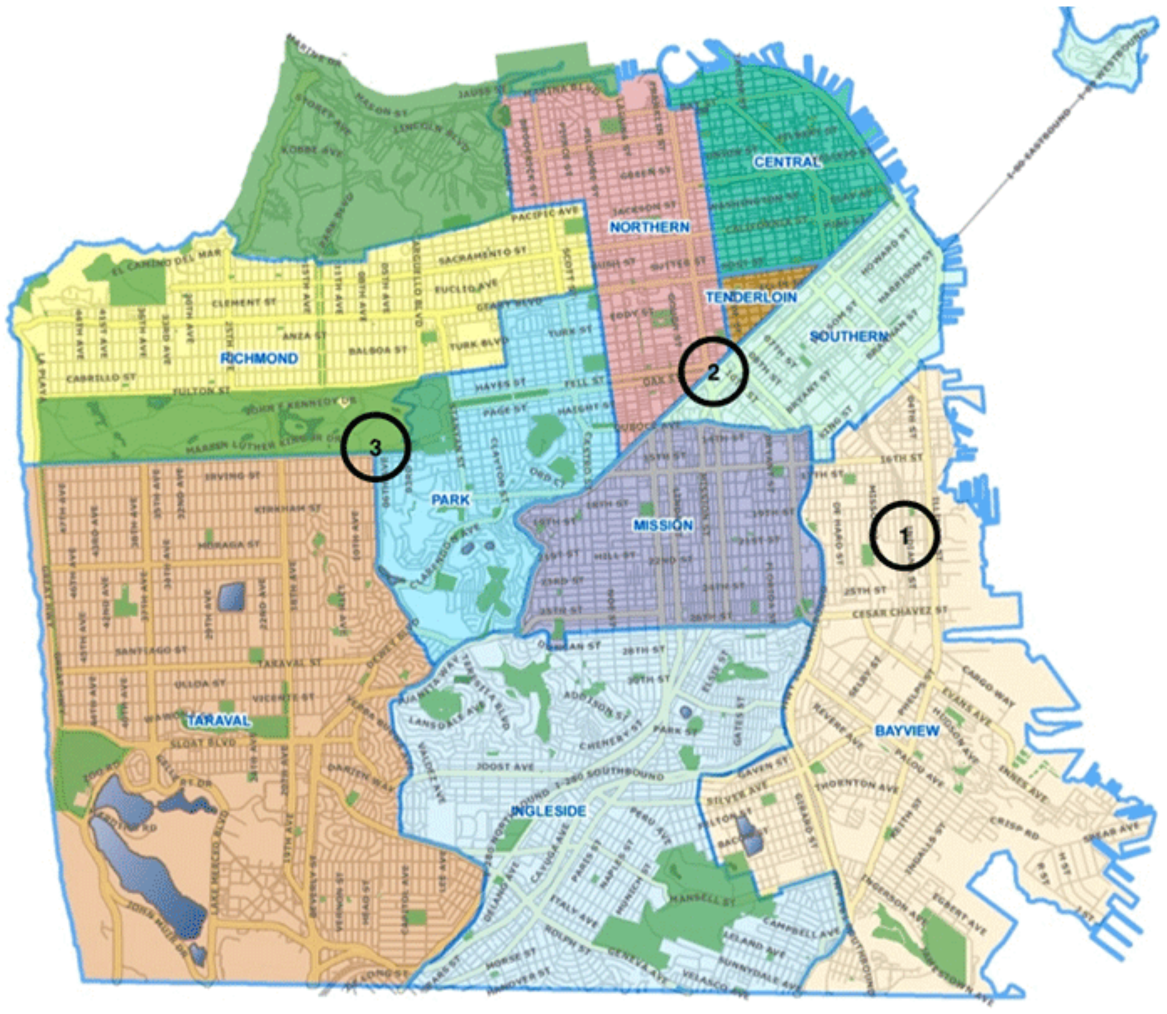
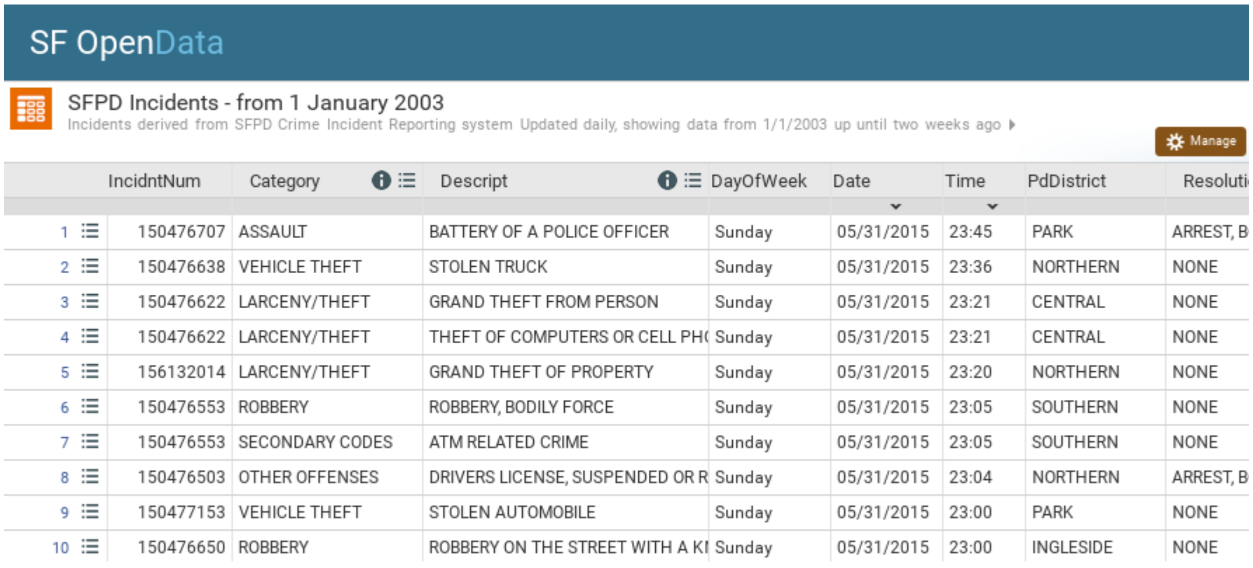
SFPD ♥ Spark

Rares Vernica, Nimish Kulkarni, KC La

DRIVE SMART
Surrey County Council and Surrey Police Working in Partnership
GX10 HFM
Southview Aviation Art



https://data.sfgov.org/Public-Safety/SFPD-Incidents-from-1-January-2003/tmnf-yvry (https://data.sfgov.org/Public-Safety/SFPD-Incidents-from-1-January-2003/tmnf-yvry)



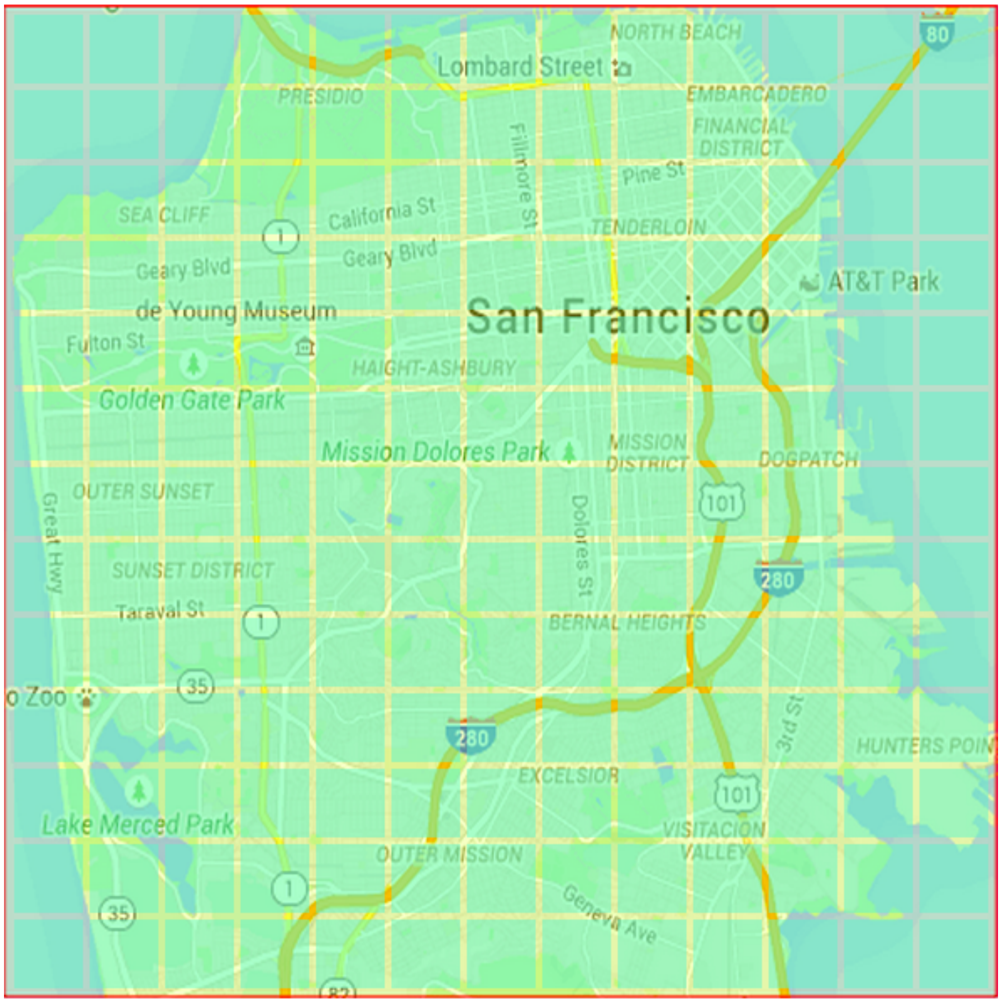
SF OpenData

SFPD Incidents - from 1 January 2003

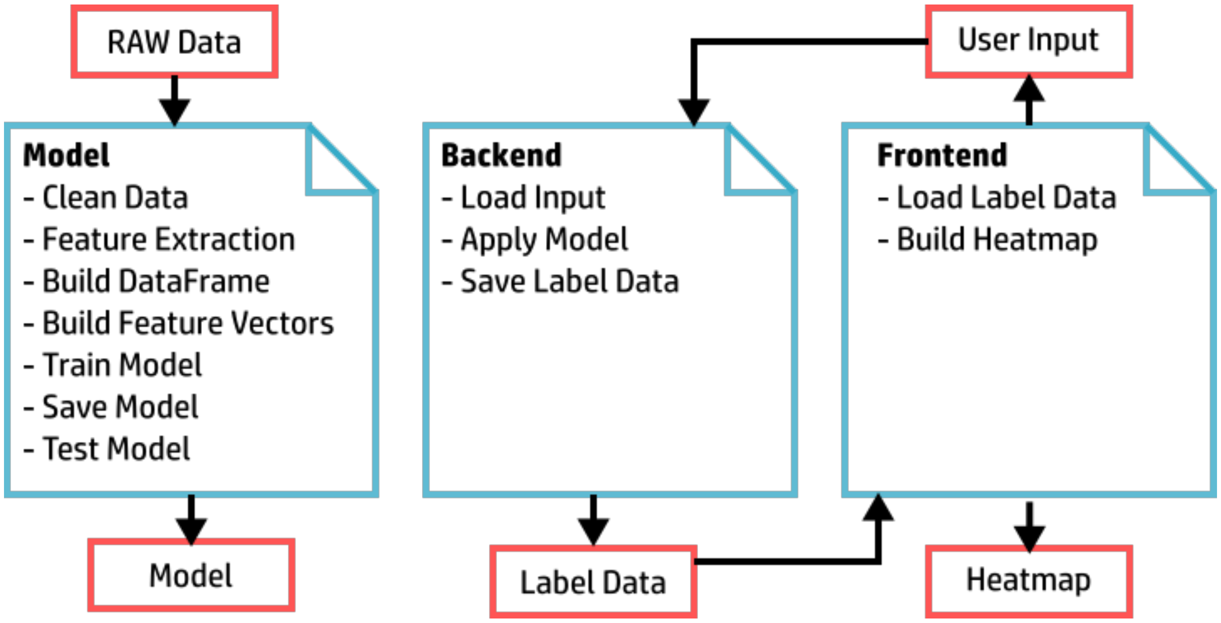
Incidents derived from SFPD Crime Incident Reporting system Updated daily, showing data from 1/1/2003 up until two weeks ago

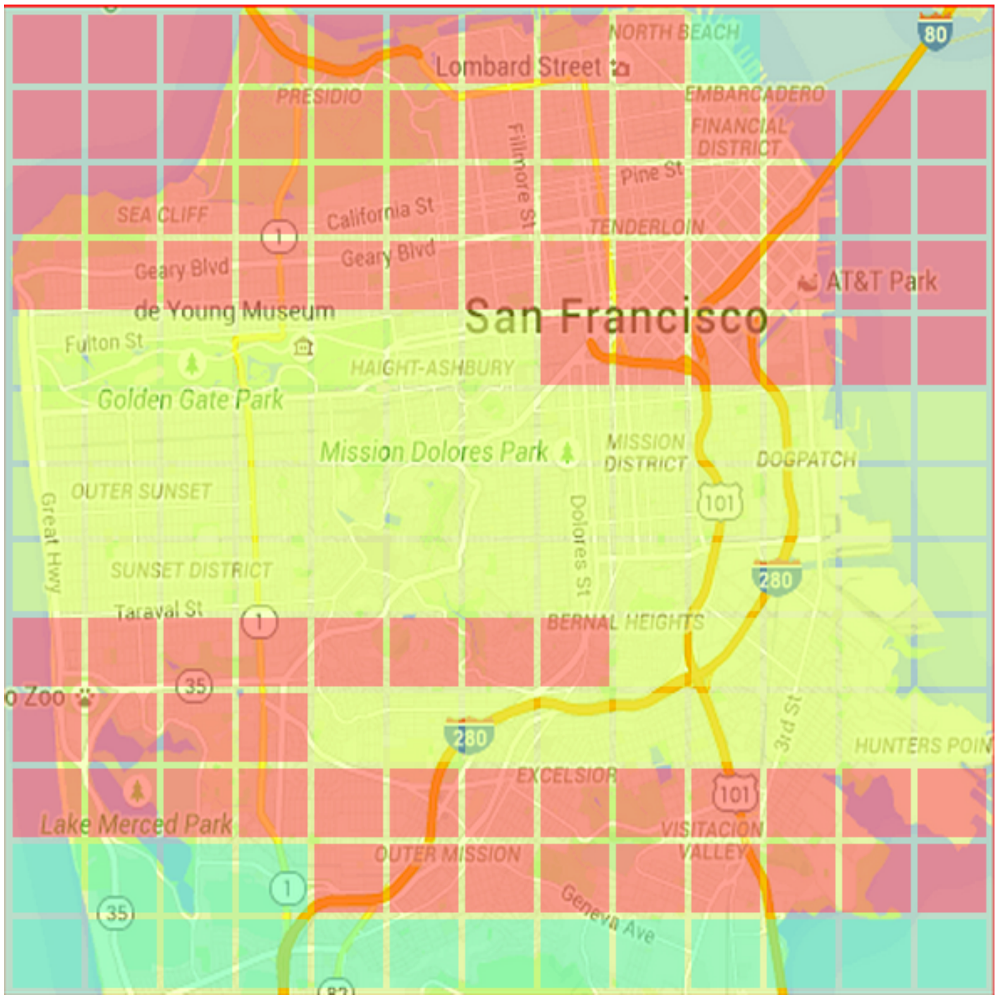
Manage

	IncidntNum	Category	Descript	DayOfWeek	Date	Time	PdDistrict	Resoluti
1	150476707	ASSAULT	BATTERY OF A POLICE OFFICER	Sunday	05/31/2015	23:45	PARK	ARREST, B
2	150476638	VEHICLE THEFT	STOLEN TRUCK	Sunday	05/31/2015	23:36	NORTHERN	NONE
3	150476622	LARCENY/THEFT	GRAND THEFT FROM PERSON	Sunday	05/31/2015	23:21	CENTRAL	NONE
4	150476622	LARCENY/THEFT	THEFT OF COMPUTERS OR CELL PH(	Sunday	05/31/2015	23:21	CENTRAL	NONE
5	156132014	LARCENY/THEFT	GRAND THEFT OF PROPERTY	Sunday	05/31/2015	23:20	NORTHERN	NONE
6	150476553	ROBBERY	ROBBERY, BODILY FORCE	Sunday	05/31/2015	23:05	SOUTHERN	NONE
7	150476553	SECONDARY CODES	ATM RELATED CRIME	Sunday	05/31/2015	23:05	SOUTHERN	NONE
8	150476503	OTHER OFFENSES	DRIVERS LICENSE, SUSPENDED OR R	Sunday	05/31/2015	23:04	NORTHERN	ARREST, B
9	150477153	VEHICLE THEFT	STOLEN AUTOMOBILE	Sunday	05/31/2015	23:00	PARK	NONE
10	150476650	ROBBERY	ROBBERY ON THE STREET WITH A KI	Sunday	05/31/2015	23:00	INGLESIDE	NONE



Data Scientist Workbench


RAW Data
Model
- Clean Data
- Feature Extraction
- Build DataFrame
- Build Feature Vectors
- Train Model
- Save Model
- Test Model
Model
Backend
- Load Input
- Apply Model
- Save Label Data
Label Data
User Input
Frontend
- Load Label Data
- Build Heatmap
Heatmap


NORTH BEACH
Lombard Street
PRESIDIO
EMBARCADERO
FINANCIAL DISTRICT
Fillmore St
Pine St
SEA CLIFF
California St
TENDERLOIN
Geary Blvd
AT&T Park
de Young Museum
San Francisco
Fulton St
HAIGHT-ASHBURY
Golden Gate Park
Mission Dolores Park
MISSION DISTRICT
DOGPATCH
OUTER SUNSET
Great Hwy
Dolores St
SUNSET DISTRICT
Taraval St
BERNAL HEIGHTS
Zoo
3rd St
HUNTERS POIN
EXCELSIOR
Lake Merced Park
VISITACION VALLEY
OUTER MISSION
Geneva Ave

Conclusions

- Address a real problem
- Complete solution
- Configurable region size
- Improve with more data

Load RAW Data

In [1]:
```
val raw = sc.textFile("/resources/rows.csv").cache
println(raw.count)
raw.take(3).foreach(println)
```

```
1769379
IncidntNum,Category,Descript,DayOfWeek,Date,Time,PdDistrict,Resolution,
Address,X,Y,Location,PdId
150470464,OTHER OFFENSES,"DRIVERS LICENSE, SUSPENDED OR REVOKED",Friday
,05/29/2015,23:40,MISSION,"ARREST, BOOKED",19TH ST / SHOTWELL ST,-122.4
15929849548,37.7604330003754,"(37.7604330003754, -122.415929849548)",15
047046465016
150470420,NON-CRIMINAL,"AIDED CASE, MENTAL DISTURBED",Friday,05/29/2015
,23:36,INGLESIDE,NONE,4900 Block of MISSION ST,-122.438695075365,37.719
6920262929,"(37.7196920262929, -122.438695075365)",15047042064020
```

Assign Location to Box

In [3]:
```
// [-122.516, -122.360, 37.700, 37.809]
val minX = -122.516
val maxX = -122.360
val minY = 37.700
val maxY = 37.809
val granularity = .01
val numberBoxes = 15

println((maxX - minX) / granularity)
println((maxY - minY) / granularity)

def toBox(x: Double, y: Double): Int =
    ((y - minY) / granularity).toInt * numberBoxes + ((x - minX) / gran
ularity).toInt
println(toBox(minX, minY), toBox(minX, maxY), toBox(maxX, minY), toBox(
maxX, maxY))
```

```
15.600000000000591
10.899999999999466
(0,150,15,165)
```

Define CSV Regular Expression

In [2]:
```
val otherThanQuote = " [^\"] "
val quotedString = String.format(" \" %s* \" ", otherThanQuote)
val regex = String.format(
    "(?x) "+ // enable comments, ignore white spaces
    ",                         "+ // match a comma
    "(?=                       "+ // start positive look ahead
    "  (                       "+ //   start group 1
    "    %s*                   "+ //     match 'otherThanQuote' zero or
 more times
    "    %s                    "+ //     match 'quotedString'
    "  )*                      "+ //   end group 1 and repeat it zero o
r more times
    "  %s*                     "+ //   match 'otherThanQuote'
    "  $                       "+ // match the end of the string
    ")                         ", // stop positive look ahead
    otherThanQuote, quotedString, otherThanQuote)
```

Split CSV

In [4]:
```
val rawParse = (raw
    .mapPartitionsWithIndex{case (index, iter) => if (index == 0) iter.
drop(1) else iter}
    .map(line => line.split(regex, -1))
    .filter(array => {
        val x = array(9).toFloat
        val y = array(10).toFloat
        if (x >= minX && x <= maxX && y >= minY && y <= maxY) true
        else false
        // array(9) != "-120.5" && array(10) != "90"
    }))
rawParse.cache
println(rawParse.count)
rawParse.take(3).foreach(array => println(array.mkString(";")))
```

```
1765904
150470464;OTHER OFFENSES;"DRIVERS LICENSE, SUSPENDED OR REVOKED";Friday
;05/29/2015;23:40;MISSION;"ARREST, BOOKED";19TH ST / SHOTWELL ST;-122.4
15929849548;37.7604330003754;"(37.7604330003754, -122.415929849548)";15
047046465016
150470420;NON-CRIMINAL;"AIDED CASE, MENTAL DISTURBED";Friday;05/29/2015
;23:36;INGLESIDE;NONE;4900 Block of MISSION ST;-122.438695075365;37.719
6920262929;"(37.7196920262929, -122.438695075365)";15047042064020
150470680;LARCENY/THEFT;GRAND THEFT PICKPOCKET;Friday;05/29/2015;23:30;
NORTHERN;NONE;1200 Block of POLK ST;-122.420326993863;37.7884521578132;
"(37.7884521578132, -122.420326993863)";15047068006113
```

Map Risk Level and Day Period

In [6]:
```scala
def toRiskLevel(count: Long): Int = {
    if (count < 5) 0 // Low
    else if (count < 30) 1 // Medium
    else 2 // High
}

def toDayPeriod(dt: java.util.Date): Int = {
    val time = dt.getHours
    if (time >= 22 || time < 6) 0 // Night
    else if (time >= 6 && time < 12) 1 // Morning
    else if (time >= 12 && time < 18) 2 // Afternoon
    else 3 // Evening
}
```

Build Data Frame

In [7]:
```
val sqlContext = new org.apache.spark.sql.SQLContext(sc)
import sqlContext.implicits._

case class Incident(
    IncidntNum: Int,
    Category: String,
    Descript: String,
    DayOfWeek: String,
    Date: java.sql.Date,
    Timestamp: java.sql.Timestamp,
    PdDistrict: String,
    Resolution: String,
    Address: String,
    X: Double,
    Y: Double,
    PdId: String,
    Box: Int,
    Hour: Int,
    DayPeriod: Int,
    DayOfWeekNum: Int,
    Month: Int)
val formatIn = new java.text.SimpleDateFormat("MM/dd/yyyy HH:mm")
val formatDayNum = new java.text.SimpleDateFormat("u")

val incidentsRDD = rawParse.map(i => {
    val x = i(9).toDouble
    val y = i(10).toDouble
    val dt = formatIn.parse(i(4) + " " + i(5))
    Incident(
    i(0).toInt,
    i(1),
    i(2),
    i(3),
    new java.sql.Date(dt.getDay, dt.getMonth, dt.getYear),
    new java.sql.Timestamp(dt.getTime),
    i(6),
    i(7),
    i(8),
    x,
    y,
    i(12),
    toBox(x, y),
    dt.getHours,
    toDayPeriod(dt),
    formatDayNum.format(dt).toInt,
    dt.getMonth)
}).cache
val incidentsDF = incidentsRDD.toDF()
incidentsDF.registerTempTable("incidents")

println(sqlContext.sql("SELECT MIN(X), MAX(X), MIN(Y), MAX(Y) FROM inci
dents").collect().mkString(","))
println(sqlContext.sql("SELECT MIN(Box), MAX(Box), MIN(Hour), MAX(Hour)
, MIN(DayPeriod), MAX(DayPeriod), MIN(Month), MAX(Month) FROM incidents
").collect().mkString(","))
println(sqlContext.sql("SELECT DayOfWeek, COUNT(*) FROM incidents GROUP
 BY DayOfWeek").collect().mkString(","))
println(sqlContext.sql("SELECT DayOfWeekNum, COUNT(*) FROM incidents GR
OUP BY DayOfWeekNum").collect().mkString(","))
```

```
[-122.51364206429,-122.370193954935,37.7078790224135,37.8086250596257]
[3,161,0,23,0,3,0,11]
[Sunday,233549],[Thursday,252348],[Friday,269319],[Saturday,254263],[Tu
esday,252294],[Wednesday,259950],[Monday,244181]
[1,244181],[2,252294],[3,259950],[4,252348],[5,269319],[6,254263],[7,23
3549]
```

In [60]:
```
sqlContext.sql("SELECT MIN(Timestamp), MAX(Timestamp) FROM incidents").
collect()
```

Out[60]:
```
Array([2003-01-01 00:01:00.0,2015-05-29 23:40:00.0])
```

In [61]:
```
val df = (incidentsDF
    .groupBy($"Box", $"Date")
    .agg(Map("*" -> "count"))).cache
println(df.agg(Map("COUNT(1)" -> "min")).collect.mkString)
println(df.agg(Map("COUNT(1)" -> "max")).collect.mkString)
println(df.agg(Map("COUNT(1)" -> "avg")).collect.mkString)
```

```
[1]
[413]
[14.781768718871636]
```

Aggregate Data at Box Level

In [8]:
```
val boxAgg = sqlContext.sql(
    "SELECT COUNT(*) AS `Count`, Box, Month, DayOfWeekNum, DayPeriod "
+
    "FROM incidents GROUP BY Box, Month, DayOfWeekNum, DayPeriod").cach
e
println(boxAgg.count)
boxAgg.take(5).foreach(a => println(a.mkString(";")))
boxAgg.printSchema
```

```
41936
28;72;8;2;1
30;38;1;7;1
53;111;2;6;0
19;78;9;5;3
10;79;3;4;1
root
 |-- Count: long (nullable = false)
 |-- Box: integer (nullable = false)
 |-- Month: integer (nullable = false)
 |-- DayOfWeekNum: integer (nullable = false)
 |-- DayPeriod: integer (nullable = false)
```

Label Data

In [9]:
```
import org.apache.spark.mllib.linalg.Vectors
import org.apache.spark.mllib.regression.LabeledPoint

val labeled = boxAgg.map(
    i => LabeledPoint(
        toRiskLevel(i.getLong(0)),
        Vectors.dense(i.getInt(1), i.getInt(2), i.getInt(3), i.getInt(4
)))).cache
println(labeled.count)
labeled.take(5).foreach(a => println(a))
```

```
41936
(1.0,[72.0,8.0,2.0,1.0])
(2.0,[38.0,1.0,7.0,1.0])
(2.0,[111.0,2.0,6.0,0.0])
(1.0,[78.0,9.0,5.0,3.0])
(1.0,[79.0,3.0,4.0,1.0])
```

In [64]:
```
labeled.filter(p => p.label == 0.0).count
```

Out[64]:
```
5268
```

Build Model

In [11]:
```
import org.apache.spark.mllib.tree.DecisionTree
import org.apache.spark.mllib.tree.model.DecisionTreeModel
import org.apache.spark.mllib.util.MLUtils

val numClasses = 3
val categoricalFeaturesInfo = Map[Int, Int](1 -> 12, 2 -> 8, 3 -> 4)
val impurity = "gini"
val maxDepth = 5
val maxBins = 32

val model = DecisionTree.trainClassifier(labeled, numClasses, categoric
alFeaturesInfo,
  impurity, maxDepth, maxBins)
model.save(sc, "/resources/model4")
```

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for furth
er details.
```

In [70]:
```
model.predict(Vectors.dense(72, 8, 2, 1))
```

Out[70]:
```
1.0
```

Evaluate Model

In [90]:
```
val splits = labeled.randomSplit(Array(0.7, 0.3), seed=777)
val (trainingData, testData) = (splits(0), splits(1))

val modelEval = DecisionTree.trainClassifier(trainingData, numClasses,
categoricalFeaturesInfo,
  impurity, maxDepth, maxBins)

val labelAndPreds = testData.map { point =>
  val prediction = modelEval.predict(point.features)
  (point.label, prediction)
}
val testErr = labelAndPreds.filter(r => r._1 != r._2).count.toDouble /
testData.count()
println("Test Error = " + testErr)
var sumP:Double = 0
var sumR:Double = 0
for(i <- 0 to 2) {
    var s1:Long = 0
    for(j <- 0 to 2)
        s1 += labelAndPreds.filter(r => r._1 == j.toDouble && r._2 == i
.toDouble).count
    var s2:Long = 0
    for(j <- 0 to 2)
        s2 += labelAndPreds.filter(r => r._1 == i.toDouble && r._2 == j
.toDouble).count
    val t = labelAndPreds.filter(r => r._1 == i.toDouble && r._2 == i.t
oDouble).count.toDouble
    val p = t / s1
    val r = t / s2
    sumP += p
    sumR += r
    println(i + " Precision: " + p + "\tRecall: " + r)
}
println("Average Precision: " + (sumP / 3))
println("Average Recall:    " + (sumR / 3))
```

```
Test Error = 0.3367314431347563
0 Precision: 0.7024793388429752 Recall: 0.32546266751754943
1 Precision: 0.6104910714285714 Recall: 0.5882985588298559
2 Precision: 0.6915646258503402 Recall: 0.8017350157728707
Average Precision: 0.6681783453739621
Average Recall:    0.5718320807067586
```