**Zeppelin** Notebook

Search your Notebooks

anonymous

# Flight Data

default

FINISHED

```
%sh

wget http://stat-computing.org/dataexpo/2009/2007.csv.bz2 -O /tmp/flights_2007.csv.bz2
```

FINISHED

```
%sh
wget http://stat-computing.org/dataexpo/2009/2008.csv.bz2 -O /tmp/flights_2008.csv.bz2
```

FINISHED

```
%sh
wget https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/2007.csv.gz -O /tmp/weather_2007.csv.gz
```

FINISHED

```
%sh
wget https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/2008.csv.gz -O /tmp/weather_2008.csv.gz
echo "downloaded"
```

FINISHED

```
%dep

z.reset()
z.load("joda-time:joda-time:2.9.1")
```

FINISHED

```
%sh
```

```
hadoop fs -rm -r -f /tmp/airflightdelays
hadoop fs -mkdir /tmp/airflightdelays

hadoop fs -put /tmp/flights_200*.bz2 /tmp/airflightdelays/
hadoop fs -put /tmp/weather_200*.bz2 /tmp/airflightdelays/
```

```
17/02/03 00:56:48 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java
classes where applicable
17/02/03 00:56:48 INFO Configuration.deprecation: io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
17/02/03 00:56:48 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interva
l = 0 minutes.
Deleted /tmp/airflightdelays
17/02/03 00:56:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java
classes where applicable
17/02/03 00:56:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java
classes where applicable
17/02/03 00:56:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java
classes where applicable
put: `/tmp/weather_200*.bz2': No such file or directory
17/02/03 00:56:55 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java
classes where applicable
Found 2 items
-rw-r--r--   1 root wheel    115.6 M 2017-02-03 00:56 /tmp/airflightdelays/flights_2007.csv.bz2
-rw-r--r--   1 root wheel    108.5 M 2017-02-03 00:56 /tmp/airflightdelays/flights_2008.csv.bz2
```

FINISHED

```
%spark

import org.apache.spark.rdd._
```

```
import org.apache.spark.rdd._
```

FINISHED

```
import scala.collection.JavaConverters._
import au.com.bytecode.opencsv.CSVReader
```

```
import scala.collection.JavaConverters._
import au.com.bytecode.opencsv.CSVReader
```

FINISHED

```
import java.io._
import org.joda.time._
import org.joda.time.format._
import org.joda.time.format.DateTimeFormat
import org.joda.time.DateTime
import org.joda.time.Days
```

```
import java.io._
import org.joda.time._
import org.joda.time.format._
import org.joda.time.format.DateTimeFormat
import org.joda.time.DateTime
import org.joda.time.Days
```

FINISHED

```
%spark

case class DelayRec(year: String,
                    month: String,
                    dayOfMonth: String,
                    dayOfWeek: String,
                    crsDepTime: String,
                    depDelay: String,
                    origin: String,
                    distance: String,
                    cancelled: String) {

    val holidays = List("01/01/2007", "01/15/2007", "02/19/2007", "05/28/2007", "06/07/2007", "07/04/2007",
      "09/03/2007", "10/08/2007" ,"11/11/2007", "11/22/2007", "12/25/2007",
      "01/01/2008", "01/21/2008", "02/18/2008", "05/22/2008", "05/26/2008", "07/04/2008",
      "09/01/2008", "10/13/2008" ,"11/11/2008", "11/27/2008", "12/25/2008")

    def gen_features: (String, Array[Double]) = {
      val values = Array(
        depDelay.toDouble,
        month.toDouble,
        dayOfMonth.toDouble,
        dayOfWeek.toDouble,
        get_hour(crsDepTime).toDouble,
        distance.toDouble,
        days_from_nearest_holiday(year.toInt, month.toInt, dayOfMonth.toInt)
```

```
   )
   new Tuple2(to_date(year.toInt, month.toInt, dayOfMonth.toInt), values)
  }

  def get_hour(depTime: String) : String = "%04d".format(depTime.toInt).take(2)
  def to_date(year: Int, month: Int, day: Int) = "%04d%02d%02d".format(year, month, day)

  def days_from_nearest_holiday(year:Int, month:Int, day:Int): Int = {
   val sampleDate = new org.joda.time.DateTime(year, month, day, 0, 0)

   holidays.foldLeft(3000) { (r, c) =>
    val holiday = org.joda.time.format.DateTimeFormat.forPattern("MM/dd/yyyy").parseDateTime(c)
    val distance = Math.abs(org.joda.time.Days.daysBetween(holiday, sampleDate).getDays)
    math.min(r, distance)
   }
  }
 }

// function to do a preprocessing step for a given file
def prepFlightDelays(infile: String): RDD[DelayRec] = {
  val data = sc.textFile(infile)

  data.map { line =>
   val reader = new CSVReader(new StringReader(line))
   reader.readAll().asScala.toList.map(rec => DelayRec(rec(0),rec(1),rec(2),rec(3),rec(5),rec(15),rec(16),rec(18),rec
(21)))
  }.map(list => list(0))
  .filter(rec => rec.year != "Year")
  .filter(rec => rec.cancelled == "0")
  .filter(rec => rec.origin == "ORD")
}

val data_2007tmp = prepFlightDelays("/tmp/airflightsdelays/flights_2007.csv.bz2")
val data_2007 = data_2007tmp.map(rec => rec.gen_features._2)
val data_2008 = prepFlightDelays("/tmp/airflightsdelays/flights_2008.csv.bz2").map(rec => rec.gen_features._2)

data_2007tmp.toDF().registerTempTable("data_2007tmp")

data_2007.take(5).map(x => x mkString ",").foreach(println)
```

```
defined class DelayRec
prepFlightDelays: (infile: String)org.apache.spark.rdd.RDD[DelayRec]
data_2007tmp: org.apache.spark.rdd.RDD[DelayRec] = MapPartitionsRDD[71] at filter at <console>:57
data_2007: org.apache.spark.rdd.RDD[Array[Double]] = MapPartitionsRDD[72] at map at <console>:52
data_2008: org.apache.spark.rdd.RDD[Array[Double]] = MapPartitionsRDD[80] at map at <console>:50
warning: there was one deprecation warning; re-run with -deprecation for details
-8.0,1.0,25.0,4.0,11.0,719.0,10.0
41.0,1.0,28.0,7.0,15.0,925.0,13.0
45.0,1.0,29.0,1.0,20.0,316.0,14.0
-9.0,1.0,17.0,3.0,19.0,719.0,2.0
180.0,1.0,12.0,5.0,17.0,316.0,3.0
```

```
%sql

select * from data_2007tmp limit 10
```

FINISHED

| year | month | dayOfMonth | dayOfWeek | crsDepTime | depDelay |
|---|---|---|---|---|---|
| 2,007 | 1 | 25 | 4 | 1,100 | -8 |
| 2,007 | 1 | 28 | 7 | 1,500 | 41 |
| 2,007 | 1 | 29 | 1 | 2,000 | 45 |
| 2,007 | 1 | 17 | 3 | 1,900 | -9 |
| 2,007 | 1 | 12 | 5 | 1,745 | 180 |
| 2,007 | 1 | 12 | 5 | 930 | 29 |
| 2,007 | 1 | 26 | 5 | 2,000 | 35 |
| 2,007 | 1 | 2 | 2 | 1,325 | -1 |
| 2,007 | 1 | 28 | 7 | 1,600 | 9 |