FINISHED

```
%dep

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

```
DepInterpreter(%dep) deprecated. Remove dependencies and repositories through GUI interpreter
 menu instead.
DepInterpreter(%dep) deprecated. Load dependency through GUI interpreter menu instead.
res0: org.apache.zeppelin.dep.Dependency = org.apache.zeppelin.dep.Dependency@7f3b2b7
```

Took 8 sec. Last updated by anonymous at February 02 2017, 8:01:30 PM.

FINISHED

```
// Imports
import org.apache.spark.sql.functions._
import org.joda.time.format.DateTimeFormat
```

```
import org.apache.spark.sql.functions._
import org.joda.time.format.DateTimeFormat
```

Took 1 sec. Last updated by anonymous at February 02 2017, 9:07:35 PM.

Machine Learning Zeppelin

Machine Learning

Untitled Untitled Untitled Untitled Untitled Untitled Untitled FINISHED

default

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

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

wget ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/2007.csv.gz -O /tmp/weather_2007.csv

wget ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/2008.csv.gz -O /tmp/weather_2008.csv
```

```
 38150K .......... .......... .......... .......... .......... 34% 6.73M 30s
 38200K .......... .......... .......... .......... .......... 34% 5.71M 30s
 38250K .......... .......... .......... .......... .......... 34% 2.99M 30s
 38300K .......... .......... .......... .......... .......... 34% 1.58M 30s
 38350K .......... .......... .......... .......... .......... 34% 1.79M 30s
 38400K .......... .......... .......... .......... .......... 34% 10.1M 30s
 38450K .......... .......... .......... .......... .......... 34% 9.24M 30s
 38500K .......... .......... .......... .......... .......... 34% 7.99M 30s
 38550K .......... .......... .......... .......... .......... 34% 2.24M 30s
 38600K .......... .......... .......... .......... .......... 34% 1.21M 30s
 38650K .......... .......... .......... .......... .......... 34% 3.65M 30s
 38700K .......... .......... .......... .......... .......... 34% 6.61M 30s
 38750K .......... .......... .......... .......... .......... 34% 3.38M 30s
 38800K .......... .......... .......... .......... .......... 34% 8.83M 30s
 38850K .......... .......... .......... .......... .......... 35% 24.1M 30s
 38900K .......... .......... .......... .......... .......... 35% 1.32M 30s
 38950K .......... .......... .......... .......... .......... 35% 2.77M 30s
 39000K .......... .......... .......... .......... .......... 35% 4.16M 30s
```

Took 1 min 0 sec. Last updated by anonymous at February 02 2017, 7:29:37 PM.

FINISHED

```
%spark

import org.apache.spark.rdd._
import scala.collection.JavaConverters._
import au.com.bytecode.opencsv.CSVReader

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 org.apache.spark.rdd._
import scala.collection.JavaConverters._
import au.com.bytecode.opencsv.CSVReader
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
```

Took 4 sec. Last updated by anonymous at February 02 2017, 8:39:16 PM.

FINISHED

```
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",
    "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)
```

Machine Learning Zeppelin

Machine Learning

```
    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").parseDateT
        val distance = Math.abs(org.joda.time.Days.daysBetween(holiday, sampleDate).getDays)
        math.min(r, distance)
      }
    }
  }
```

```
defined class DelayRec
```

Took 1 sec. Last updated by anonymous at February 02 2017, 8:50:02 PM.

FINISHED

```
// 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),r
    }.map(list => list(0))
    .filter(rec => rec.year != "Year")
    .filter(rec => rec.cancelled == "0")
    .filter(rec => rec.origin == "ORD")
}

val data_2007tmp = prepFlightDelays("/Users/joannariascos/Downloads/DataFiles/2007.csv")
val data_2007 = data_2007tmp.map(rec => rec.gen_features._2)
val data_2008 = prepFlightDelays("/Users/joannariascos/Downloads/DataFiles/2008.csv").map(rec
data_2007tmp.toDF().registerTempTable("data_2007tmp")

data_2007.take(5).map(_.mkString(",")).foreach(println)
```

Machine Learning

default

```
prepFlightDelays: (infile: String)org.apache.spark.rdd.RDD[DelayRec]
data_2007tmp: org.apache.spark.rdd.RDD[DelayRec] = MapPartitionsRDD[168] at filter at <console
>:58
data_2007: org.apache.spark.rdd.RDD[Array[Double]] = MapPartitionsRDD[169] at map at <console
>:52
data_2008: org.apache.spark.rdd.RDD[Array[Double]] = MapPartitionsRDD[177] 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
```

Took 4 sec. Last updated by anonymous at February 02 2017, 9:06:07 PM.

READY