Residência em Engenharia e Ciência de Dados

Processamento de Dados em Larga Escala

{ SPARK }

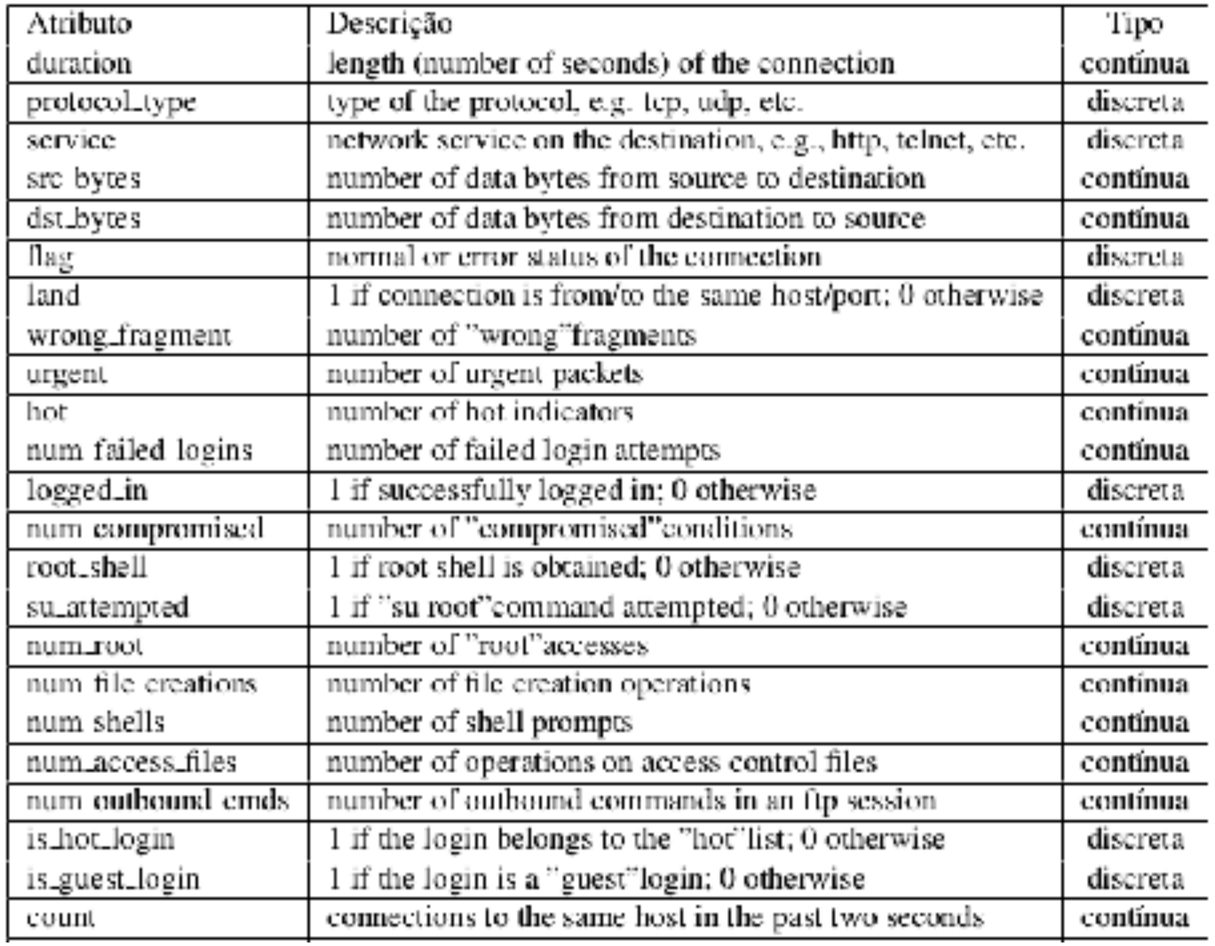
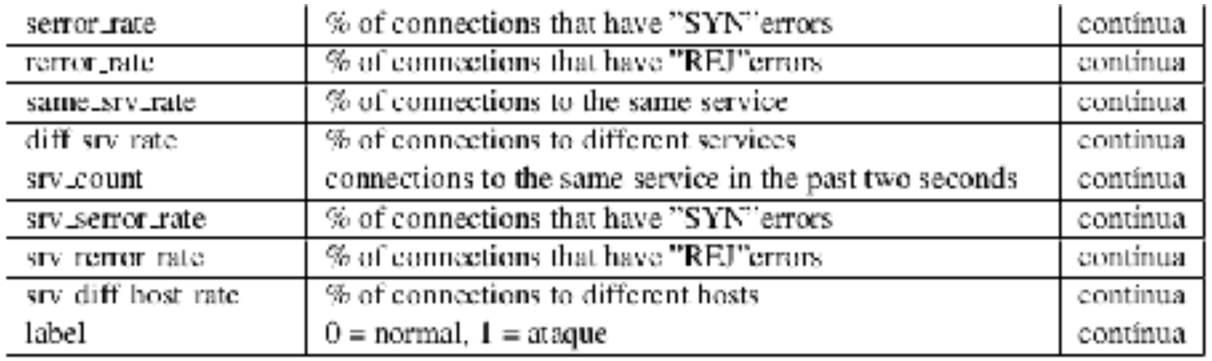
KDD Cup 99 | PT7 Web

Prof. Jairson B. Rodrigues
jbr@cin.ufpe.br

{ exemplo aplicado / KDDCup 99 }

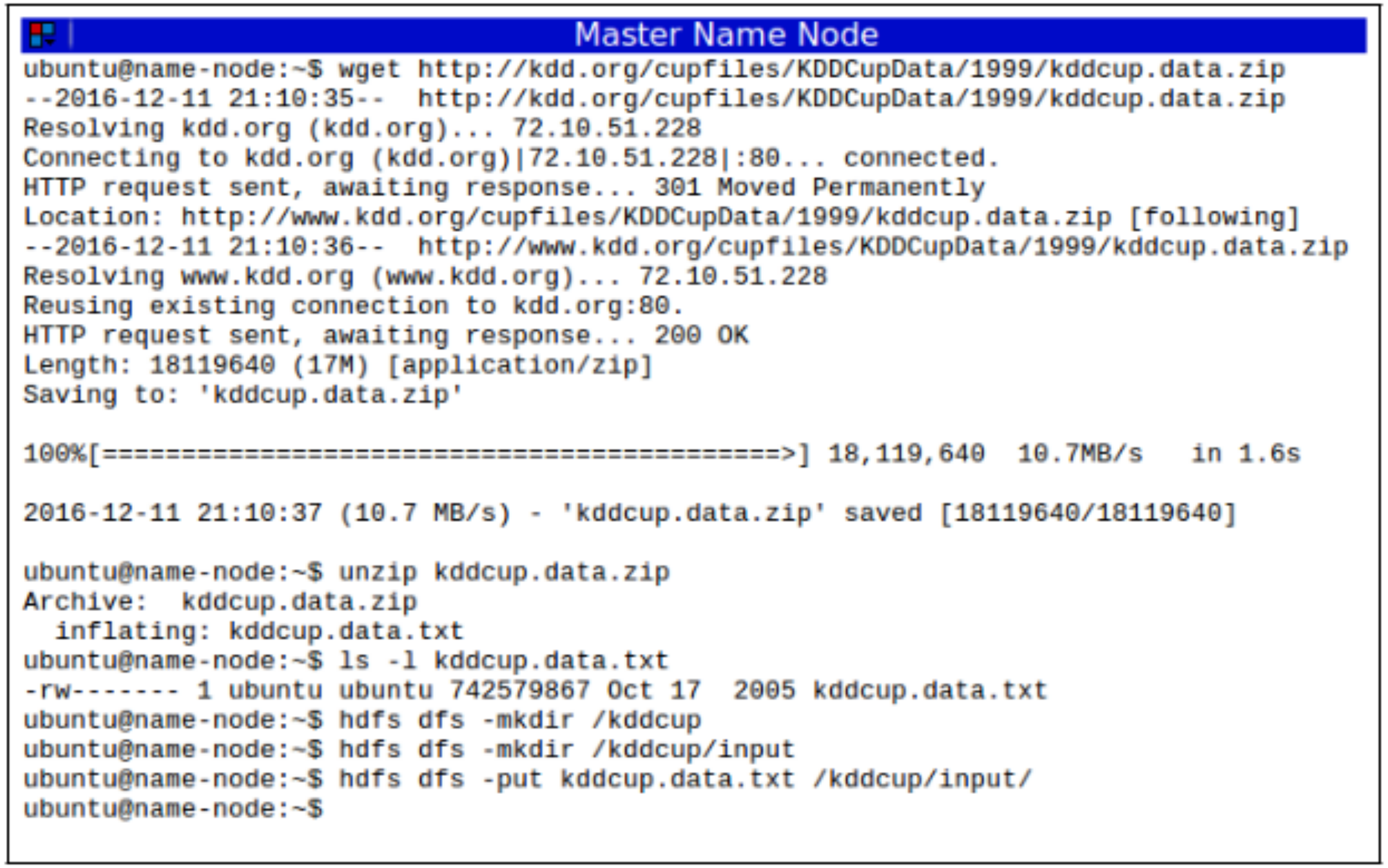
- Detecção de padrões de ataque em conexões de rede a partir da análise de log conexões disponibilizada pelo MIT Lincoln Labs (KDD Cup 99)
  - https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
- 4.898.431 de observações
- 41 características (sete categóricas, demais contínuas)
- Classe 0 / normal = 972781, 0,19859%,
- Classe 1 / ataque = 3925650, 0,80141%
- 709.18 Megabytes

{ atributos }

Atributo	Descrição	Tipo
duration	length (number of seconds) of the connection	continua
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discreta
service	network service on the destination, e.g., http, telnet, etc.	discreta
src bytes	number of data bytes from source to destination	continua
dst_bytes	number of data bytes from destination to source	continua
flag	normal or error status of the connection	discreta
land	1 if connection is from/to the same host/port; 0 otherwise	discreta
wrong_fragment	number of "wrong"fragments	continua
urgent	number of urgent packets	continua
hot	number of hot indicators	continua
num failed logins	number of failed login attempts	continua
logged_in	1 if successfully logged in; 0 otherwise	discreta
num compromised	number of "compromised"conditions	continua
root_shell	1 if root shell is obtained; 0 otherwise	discreta
su_attempted	1 if "su root"command attempted; 0 otherwise	discreta
num_root	number of "root"accesses	continua
num file creations	number of file creation operations	continua
num shells	number of shell prompts	continua
num_access_files	number of operations on access control files	continua
num outbound cmds	number of outbound commands in an ftp session	continua
is_hot_login	1 if the login belongs to the "hot"list; 0 otherwise	discreta
is_guest_login	1 if the login is a "guest"login; 0 otherwise	discreta
count	connections to the same host in the past two seconds	continua
serror_rate	% of connections that have "SYN"errors	continua
rerror_rate	% of connections that have "REJ"errors	continua
same_srv_rate	% of connections to the same service	continua
diff srv rate	% of connections to different services	continua
srv_count	connections to the same service in the past two seconds	continua
srv_serror_rate	% of connections that have "SYN"errors	continua
srv rerror rate	% of connections that have "REJ"errors	continua
srv diff host rate	% of connections to different hosts	continua
label	0 = normal, 1 = ataque	continua

{ distribuição dos dados }



```
ubuntu@name-node:~$ wget http://kdd.org/cupfiles/KDDCupData/1999/kddcup.data.zip
--2016-12-11 21:10:35--  http://kdd.org/cupfiles/KDDCupData/1999/kddcup.data.zip
Resolving kdd.org (kdd.org)... 72.10.51.228
Connecting to kdd.org (kdd.org)|72.10.51.228|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.kdd.org/cupfiles/KDDCupData/1999/kddcup.data.zip [following]
--2016-12-11 21:10:36--  http://www.kdd.org/cupfiles/KDDCupData/1999/kddcup.data.zip
Resolving www.kdd.org (www.kdd.org)... 72.10.51.228
Reusing existing connection to kdd.org:80.
HTTP request sent, awaiting response... 200 OK
Length: 18119640 (17M) [application/zip]
Saving to: 'kddcup.data.zip'

100%[===========================================>] 18,119,640  10.7MB/s   in 1.6s

2016-12-11 21:10:37 (10.7 MB/s) - 'kddcup.data.zip' saved [18119640/18119640]

ubuntu@name-node:~$ unzip kddcup.data.zip
Archive:  kddcup.data.zip
  inflating: kddcup.data.txt
ubuntu@name-node:~$ ls -l kddcup.data.txt
-rw------- 1 ubuntu ubuntu 742579867 Oct 17  2005 kddcup.data.txt
ubuntu@name-node:~$ hdfs dfs -mkdir /kddcup
ubuntu@name-node:~$ hdfs dfs -mkdir /kddcup/input
ubuntu@name-node:~$ hdfs dfs -put kddcup.data.txt /kddcup/input/
ubuntu@name-node:~$
```

{ simplificação pedagógica :-) }

- Todos os tipos de ataque foram classificados como classe = 1
- Padrões que não correspondem a ataques, classe = 0
- Variáveis categóricas foram submetidas a
  - StringIndexer: converte variáveis categóricas para índices
  - OneHotEncoder: converte índices vetores binários
  - VectorAssembler: agrega todas as características em um único vetor esparso

{ vetores esparsos vs vetores densos }

- Vetor denso:

  [1, 0, 66, 29.4, 0, 0, 0, 0, 0, 0, -9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

- Vetor esparso:

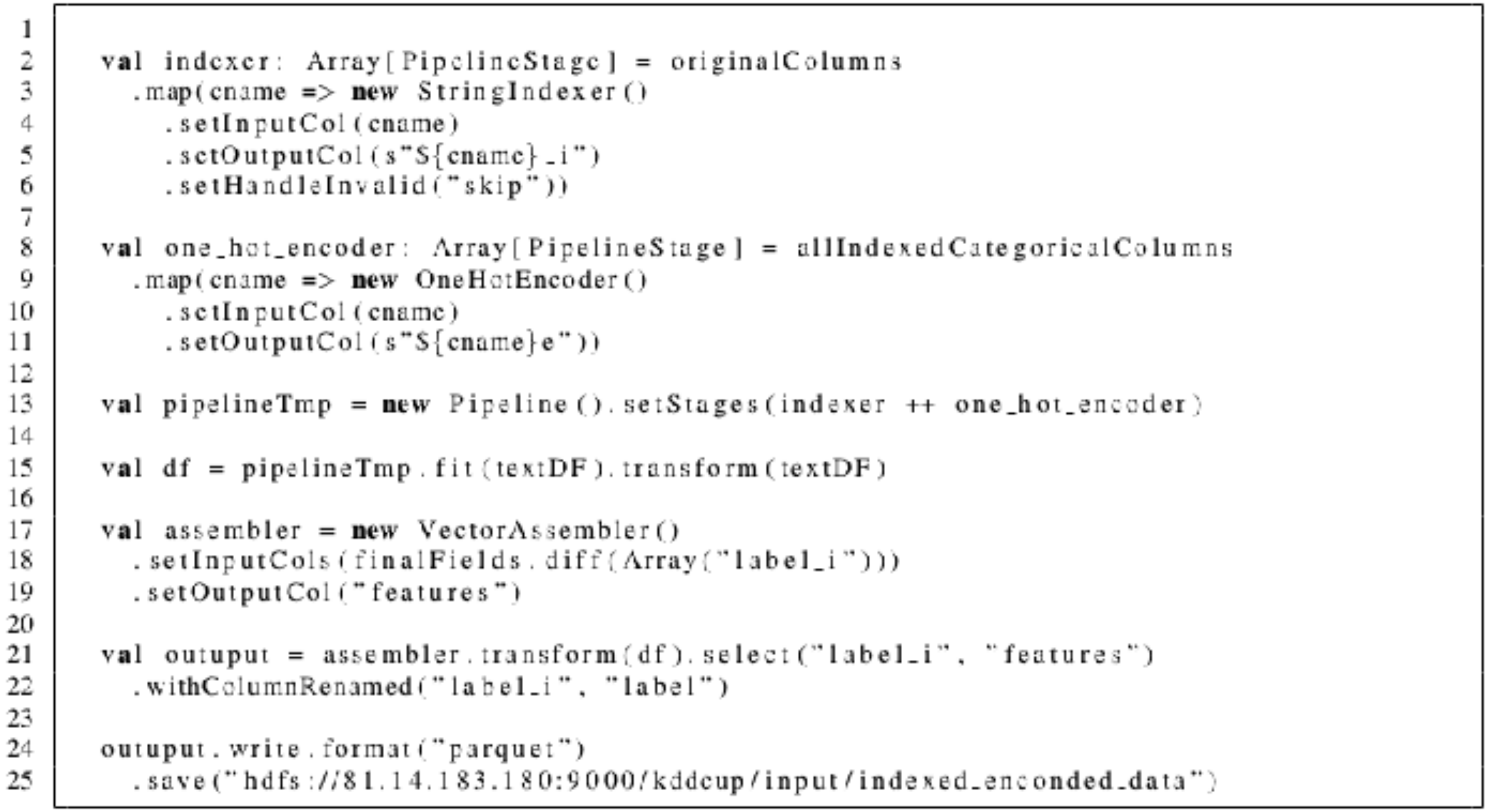
  (26, [0, 2, 3, 10], [1, 66, 29.4, -9])

Valores

Tamanho

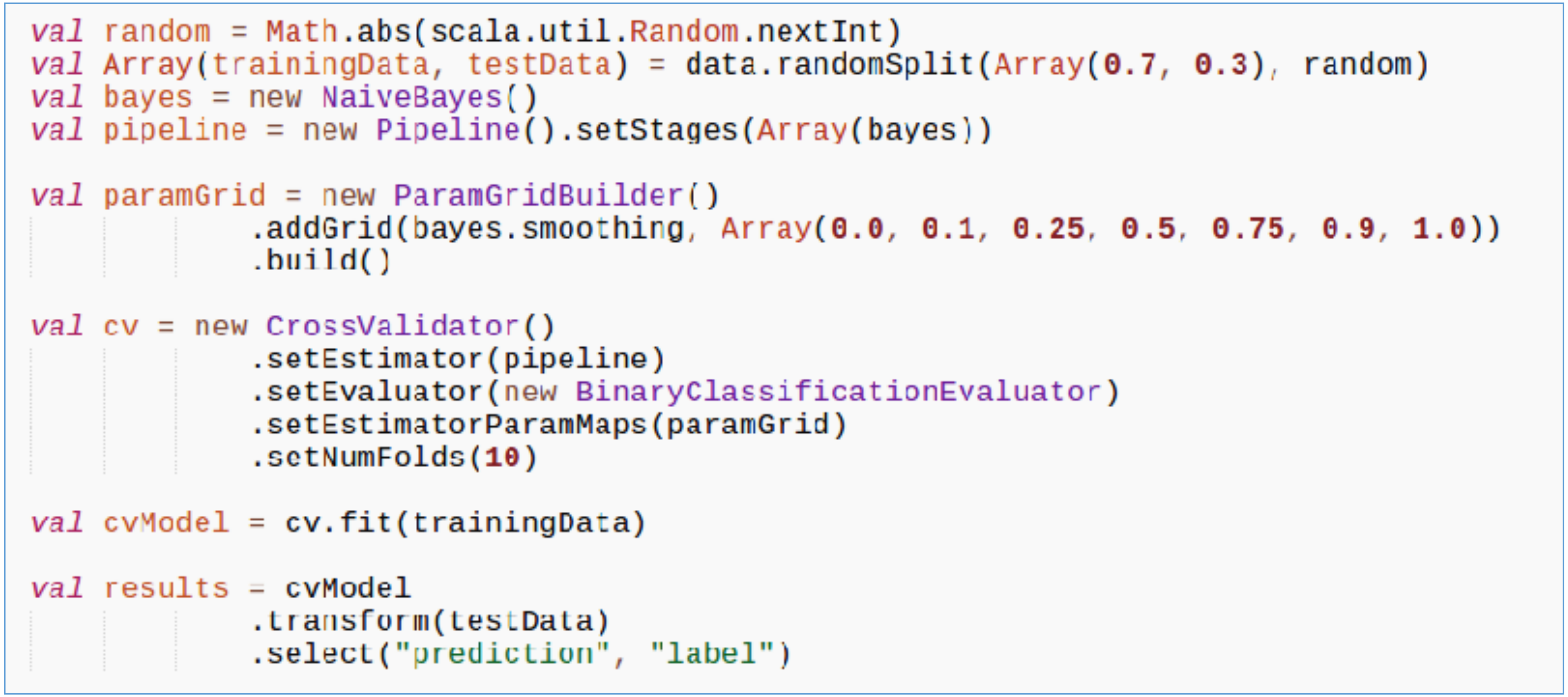
Índices não nulos

{ tratamento de dados categóricos }



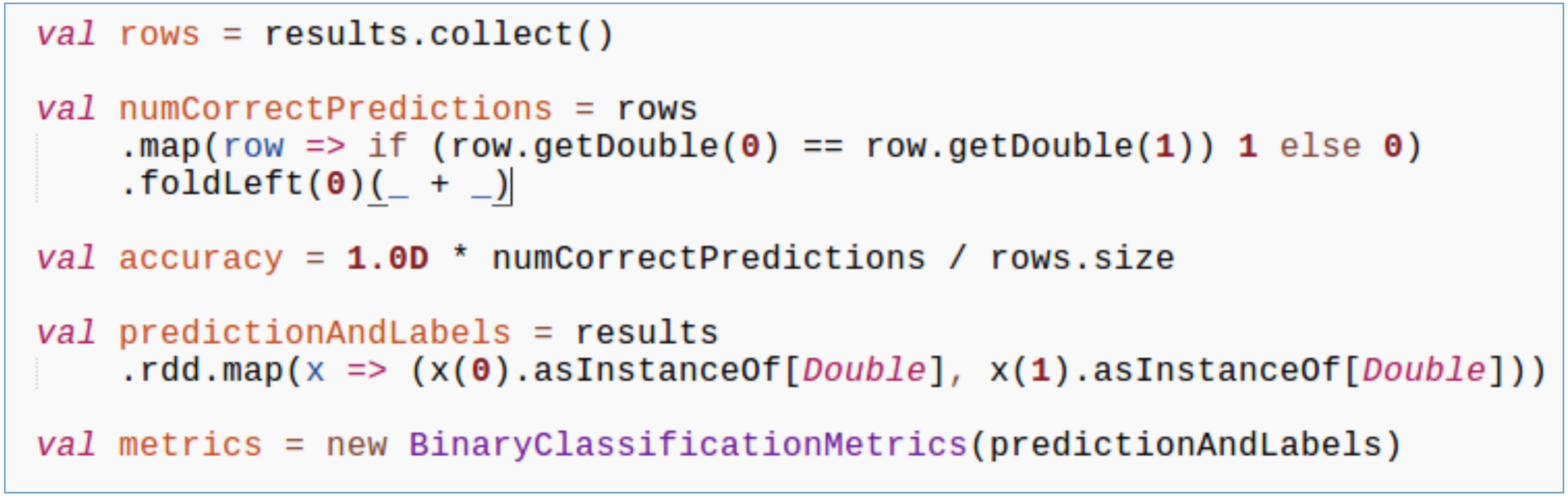
```scala

val indexer: Array[PipelineStage] = originalColumns
  .map(cname => new StringIndexer()
    .setInputCol(cname)
    .setOutputCol(s"${cname}_i")
    .setHandleInvalid("skip"))

val one_hot_encoder: Array[PipelineStage] = allIndexedCategoricalColumns
  .map(cname => new OneHotEncoder()
    .setInputCol(cname)
    .setOutputCol(s"${cname}e"))

val pipelineTmp = new Pipeline().setStages(indexer ++ one_hot_encoder)

val df = pipelineTmp.fit(textDF).transform(textDF)

val assembler = new VectorAssembler()
  .setInputCols(finalFields.diff(Array("label_i")))
  .setOutputCol("features")

val outuput = assembler.transform(df).select("label_i", "features")
  .withColumnRenamed("label_i", "label")

outuput.write.format("parquet")
  .save("hdfs://81.14.183.180:9000/kddcup/input/indexed_enconded_data")
```

{ classificador naïve bayes }



```scala
val random = Math.abs(scala.util.Random.nextInt)
val Array(trainingData, testData) = data.randomSplit(Array(0.7, 0.3), random)
val bayes = new NaiveBayes()
val pipeline = new Pipeline().setStages(Array(bayes))

val paramGrid = new ParamGridBuilder()
			.addGrid(bayes.smoothing, Array(0.0, 0.1, 0.25, 0.5, 0.75, 0.9, 1.0))
			.build()

val cv = new CrossValidator()
			.setEstimator(pipeline)
			.setEvaluator(new BinaryClassificationEvaluator)
			.setEstimatorParamMaps(paramGrid)
			.setNumFolds(10)

val cvModel = cv.fit(trainingData)

val results = cvModel
			.transform(testData)
			.select("prediction", "label")
```

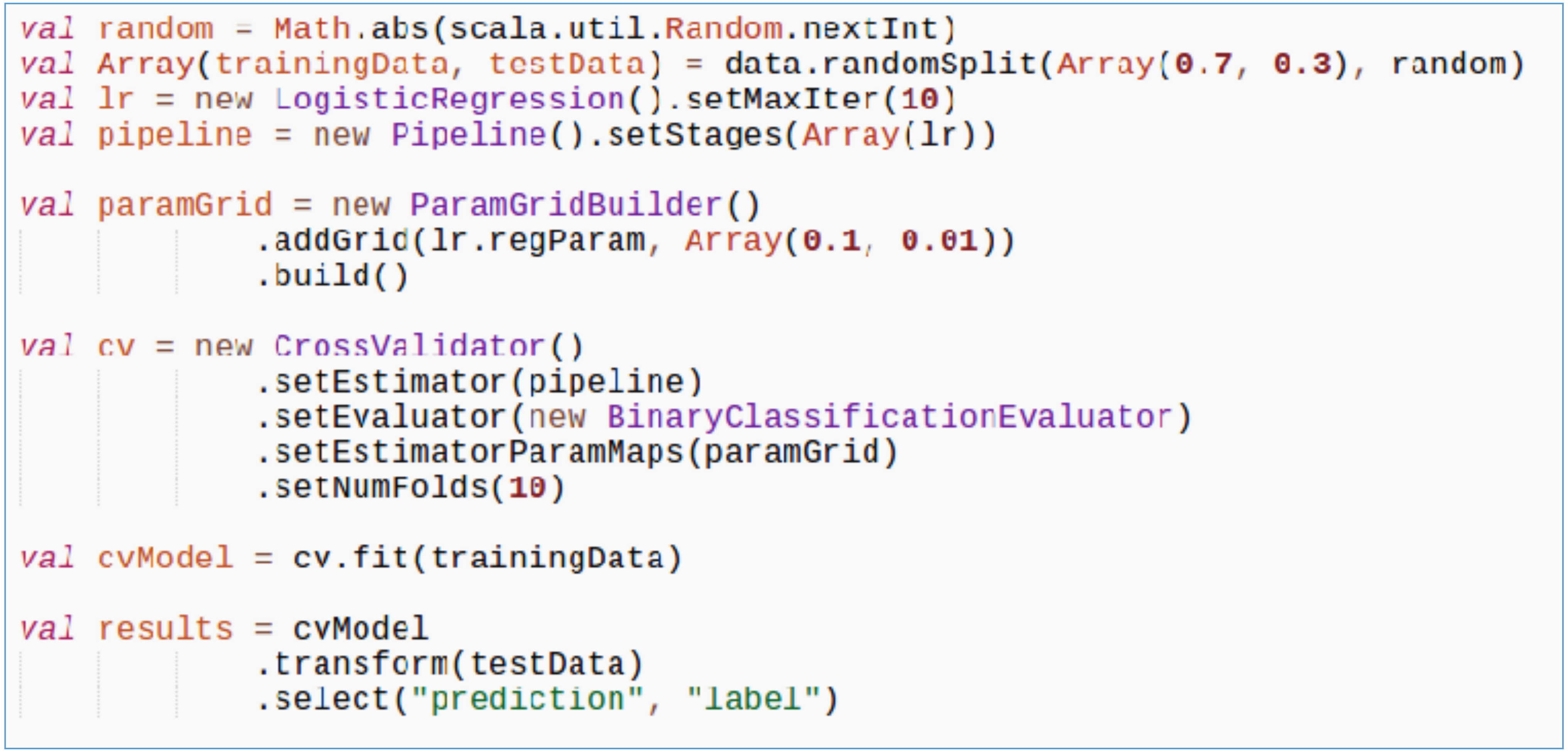
{ classificador naïve bayes }



```
val rows = results.collect()

val numCorrectPredictions = rows
    .map(row => if (row.getDouble(0) == row.getDouble(1)) 1 else 0)
    .foldLeft(0)(_ + _)

val accuracy = 1.0D * numCorrectPredictions / rows.size

val predictionAndLabels = results
    .rdd.map(x => (x(0).asInstanceOf[Double], x(1).asInstanceOf[Double]))

val metrics = new BinaryClassificationMetrics(predictionAndLabels)
```

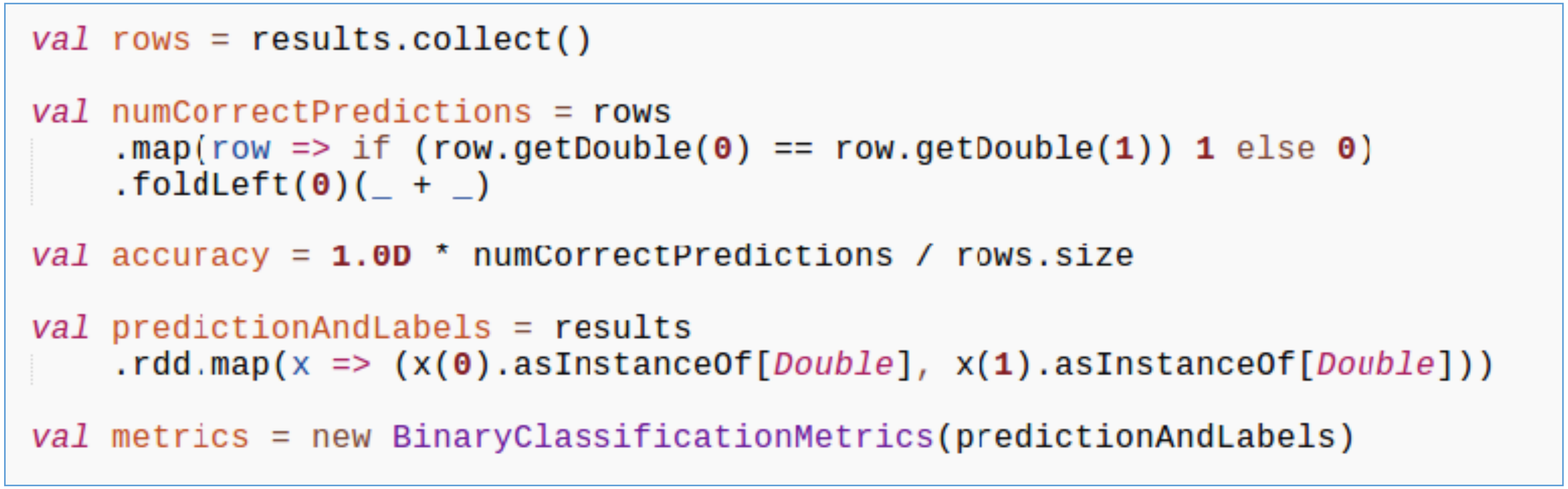
Acurácia: de 96%,
Área sob a curva Precision-Recall: 0.917
Área sob a curva ROC: 0.902

{ regressão logística }



```
val random = Math.abs(scala.util.Random.nextInt)
val Array(trainingData, testData) = data.randomSplit(Array(0.7, 0.3), random)
val lr = new LogisticRegression().setMaxIter(10)
val pipeline = new Pipeline().setStages(Array(lr))

val paramGrid = new ParamGridBuilder()
            .addGrid(lr.regParam, Array(0.1, 0.01))
            .build()

val cv = new CrossValidator()
            .setEstimator(pipeline)
            .setEvaluator(new BinaryClassificationEvaluator)
            .setEstimatorParamMaps(paramGrid)
            .setNumFolds(10)

val cvModel = cv.fit(trainingData)

val results = cvModel
            .transform(testData)
            .select("prediction", "label")
```

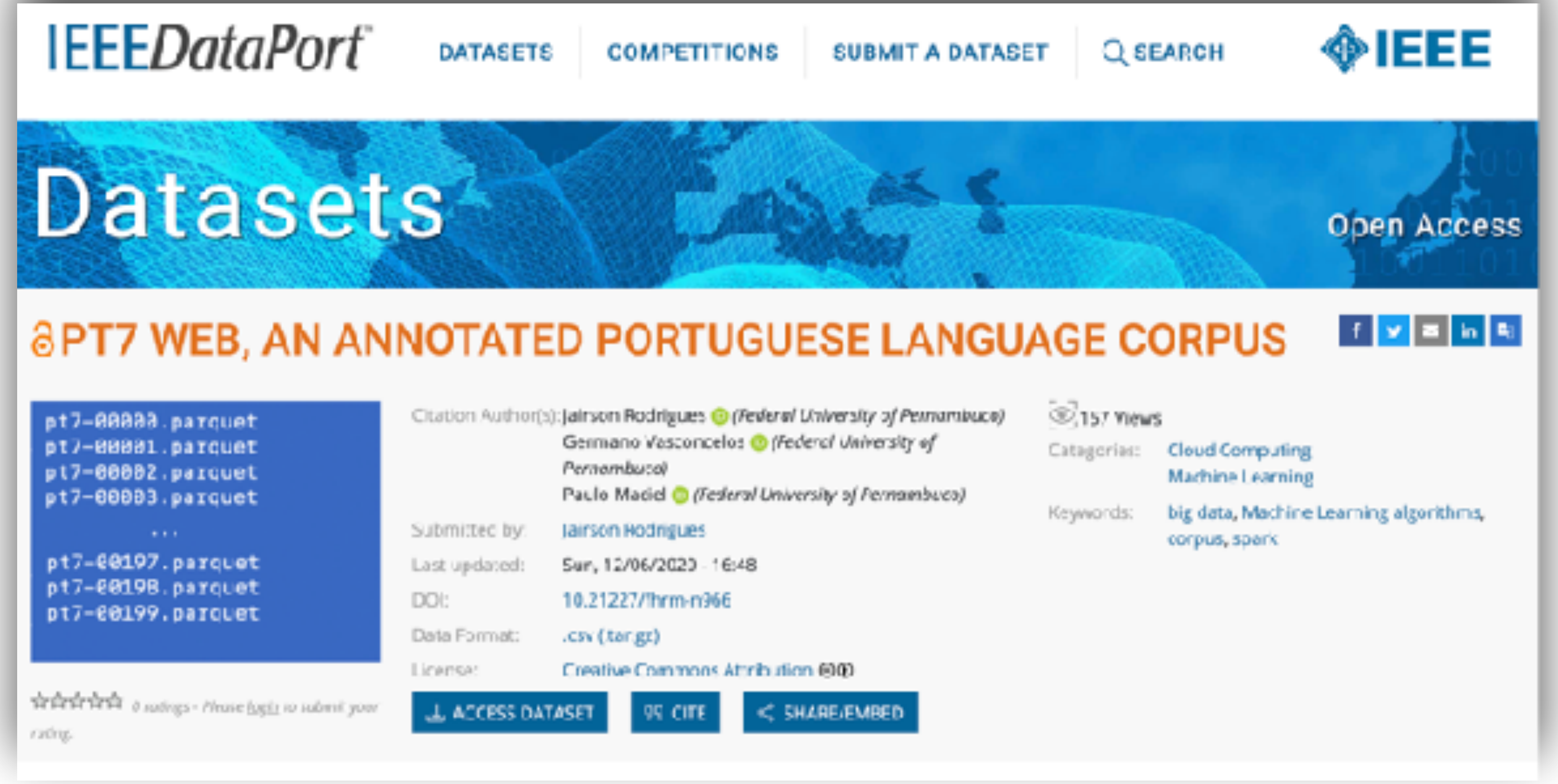
{ regressão logística }



```
val rows = results.collect()

val numCorrectPredictions = rows
    .map(row => if (row.getDouble(0) == row.getDouble(1)) 1 else 0)
    .foldLeft(0)(_ + _)

val accuracy = 1.0D * numCorrectPredictions / rows.size

val predictionAndLabels = results
    .rdd.map(x => (x(0).asInstanceOf[Double], x(1).asInstanceOf[Double]))

val metrics = new BinaryClassificationMetrics(predictionAndLabels)
```

Acurácia: de 97,5%,
Área sob a curva Precision-Recall: 0.947
Área sob a curva ROC: 0.945

{ prática pt7 web corpus }



PT7 WEB, AN ANNOTATED PORTUGUESE LANGUAGE CORPUS

```
pt7-00000.parquet
pt7-00001.parquet
pt7-00002.parquet
pt7-00003.parquet
        ...
pt7-00197.parquet
pt7-00198.parquet
pt7-00199.parquet
```

0 ratings - Please login to submit your rating.

Citation Author(s):	Jairson Rodrigues *(Federal University of Pernambuco)* Germano Vasconcelos *(Federal University of Pernambuco)* Paulo Maciel *(Federal University of Pernambuco)*
Submitted by:	Jairson Rodrigues
Last updated:	Sun, 12/06/2020 - 16:48
DOI:	10.21227/fhrm-n966
Data Format:	.csv (tar.gz)
License:	Creative Commons Attribution

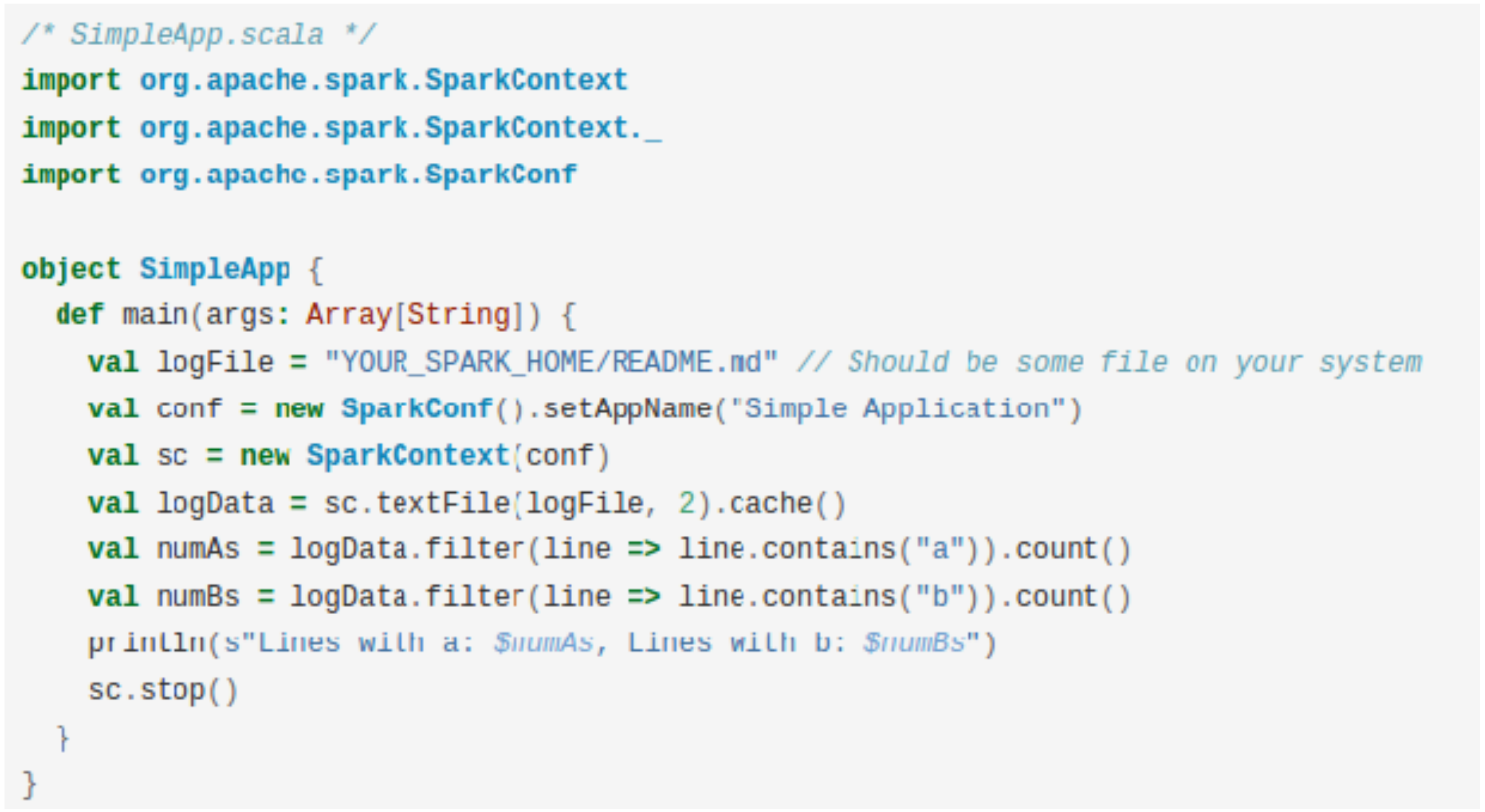
157 Views

Categories:	Cloud Computing Machine Learning
Keywords:	big data, Machine Learning algorithms, corpus, spark

https://dx.doi.org/10.21227/fhrm-n966

{ aplicações auto-contidas }

- Utilizadas para submeter jobs em spark para o cluster



```scala
/* SimpleApp.scala */
import org.apache.spark.SparkContext
import org.apache.spark.SparkContext._
import org.apache.spark.SparkConf

object SimpleApp {
  def main(args: Array[String]) {
    val logFile = "YOUR_SPARK_HOME/README.md" // Should be some file on your system
    val conf = new SparkConf().setAppName("Simple Application")
    val sc = new SparkContext(conf)
    val logData = sc.textFile(logFile, 2).cache()
    val numAs = logData.filter(line => line.contains("a")).count()
    val numBs = logData.filter(line => line.contains("b")).count()
    println(s"Lines with a: $numAs, Lines with b: $numBs")
    sc.stop()
  }
}
```

{ aplicações auto-contidas - compilação }

- Em Scala...
- Estrutura do projeto

```
./simple.sbt
./src
./src/main
./src/main/scala
./src/main/scala/SimpleApp.scala
```

Compilação

```
sbt package
...
[info] Packaging {..}/{..}/target/scala-2.11/simple-project_2.11-1.0.jar
```

{ aplicações auto-contidas - dependências }

- Arquivo de dependências (simple.sbt)

```
name := "Simple Project"
version := "1.0"
scalaVersion := "2.11.7"
libraryDependencies += "org.apache.spark" %% "spark-core" % "2.1.0"
```

{ aplicações auto-contidas - execução }

- SPARK-SUBMIT (localhost)

```
YOUR_SPARK_HOME/bin/spark-submit \
  --class "SimpleApp" \
  --master local[*] \
  <path>/target/scala-2.11/simple-project_2.11-1.0.jar
```

- SPARK-SUBMIT (utilizando o cluster em modo YARN)

```
YOUR_SPARK_HOME/bin/spark-submit \
--class "SimpleApp" \
--master yarn \
<path>/target/scala-2.11/kdd-cup_2.11-1.0.jar
```

- SPARK-SUBMIT (utilizando o cluster em modo Standalone)

```
YOUR_SPARK_HOME/bin/spark-submit \
--class "SimpleApp" \
--master spark://spark-node1:7077 \
<path>/target/scala-2.11/kdd-cup_2.11-1.0.jar
```

Maiores informações em: http://spark.apache.org/docs/latest/submitting-applications.html

{ spark-submit }

- KDDCupETL
- KDDCupRL
- KDDCupNaiveBayes

- No **Namenode**

```
spark-submit --class "<nome_do_objeto>" \
--master yarn \
--deploy-mode <modo> \
--conf spark.serializer=org.apache.spark.serializer.KryoSerializer \
/home/ubuntu/mllib/target/scala-2.11/kddcup_2.11-1.0.jar
```

- cluster
- client

{ spark-shell }

```
$ spark-shell
   --master spark://spark-node1:7077
   --num-executors 2
   --driver-memory 6G
   --executor-memory 4G
   --executor-cores 4
   -i KDD-ETL.scala

   … OU KDD-NB.scala
   … OU KDD-LR.scala
```

{ para onde ir agora? }

- Machine Learning Library Guide
- Extração de Características
- Questões de Otimização
- Métricas de Avaliação