Centro de Informática - Cin – UFPE

Residência em Engenharia e Ciência de Dados

Disciplina: Processamento de Dados em Larga Escala

Residente: Liviany Reis Rodrigues

Projeto Final da disciplina de Processamento de dados em Larga Escala

[ETL / treinamento / teste sobre Corpus PT7 multiclasse]

Contextualização:

O PT7 Web (https://ieee-dataport.org/open-access/pt7-web-annotated-portuguese-language-corpus) é um Corpus anotado em língua portuguesa construído a partir de amostras coletadas de setembro de 2018 a março de 2020 de sete países de língua portuguesa: Angola, Brasil, Portugal, Cabo Verde, Guiné-Bissau, Macau e Moçambique. Os registros foram filtrados do Common Crawl — um conjunto de dados em escala de petabytes de domínio público de páginas da Web em vários idiomas, misturados em instantâneos temporais da Web, disponíveis mensalmente [1]. As páginas brasileiras foram rotuladas como classe positiva e as demais como classe negativa (português não brasileiro). O conjunto de dados totalizou 249,74 GB de texto HTML bruto relacionado a 16.346.693 páginas da web exclusivas. Os dados foram pré-processados para produzir vetores de distribuição de palavras de alta dimensionalidade (2 elevado a 18 = 262.144 características) como entrada para as fases de treinamento e teste. Uma demonstração do uso desses dados pode ser verificada em um projeto fracionário de dois níveis para investigar o desempenho do cluster no Spark.

Será utilizado um extrato reduzido do PT7 Web, equivalente 17014 páginas ~ 0.1% do Corpus original. Foram disponibilizados cinco arquivos (pt7-raw.zip), separados pelo domínio de nível superior de cada país (.br, .pt, .mo, .gw, .mz, .ao e .pt).

Os dados se encontram rotulados como 1:pt_BR e 0:pt_OTHERS e estão disponível no formato a seguir, onde:

• label - rótulo

• url - endereço original completo da página

• digest - uma função de bash do conteúdo da página

• raw - os dados brutos do texto da página após limpeza de tags HTML

A tarefa preliminar do projeto consiste em realizar o processo de ETL sobre os dados brutos, transformando o conteúdo de cada página web em um vetor esparso de características no formato exigido pelo Spark. A base deve separar os dados em novos

rótulos, de acordo com cada país, formando uma base rotulada multiclasse.

Modelo de Machine Learning : Random Forest

Random Forest: *Random* significa aleatório, e denota o comportamento do algoritmo ao selecionar subconjuntos de *features* e montar mini árvores de decisão. *Forest* significa floresta, já que são geradas várias árvores de decisão. Basicamente, o algoritmo possui 4 passos:

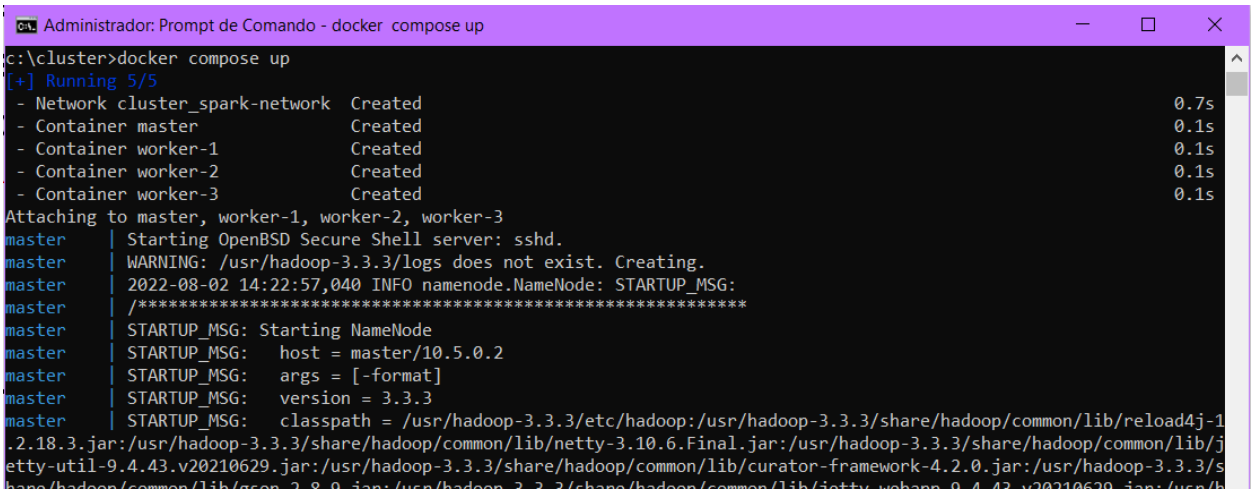
1. Seleção aleatória de algumas features
2. Seleção da feature mais adequada para a posição de nó raiz
3. Geração dos nós filhos
4. Repete os passos acima até que se atinja a quantidade de árvores desejada

Depois que o modelo é gerado, as previsões são feitas a partir de “votações”. Cada mini árvore toma uma decisão a partir dos dados apresentados. A decisão mais votada é a resposta do algoritmo.

Executando o Projeto

Etapa 01: Processamento ETL

1. Instale o Docker Desktop na sua máquina de acordo com seu sistema Operacional
2. Baixe o arquivo cluster.zip disponível no link: https://abrir.link/GdgwQ
3. Copie a pasta **cluster** descompactada para c:/ e execute via prompt de comando como administrador o seguinte comando:
    a. Entrar na pasta cluster:
        i. cd c:/cluster
    b. docker compose up

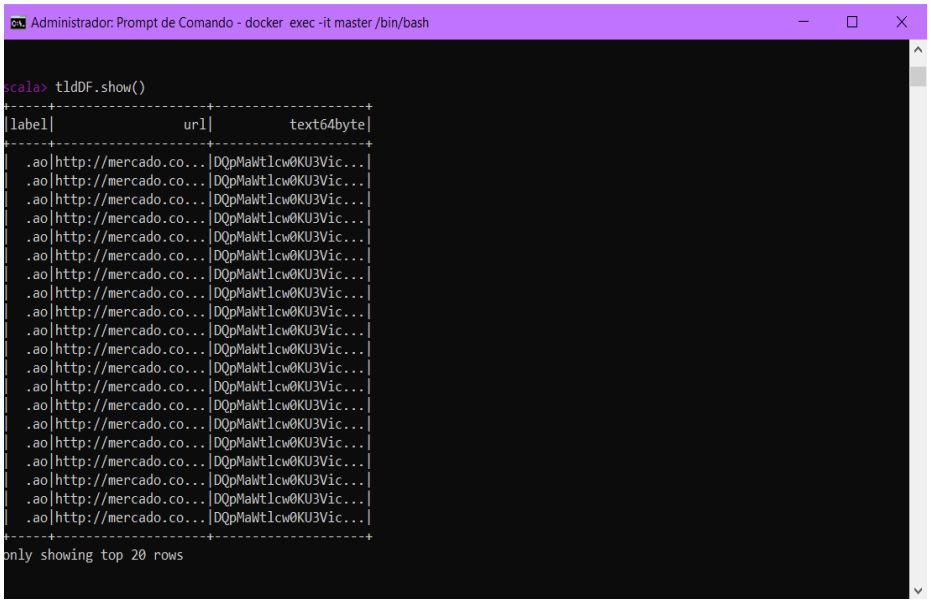


```
c:\cluster>docker compose up
[+] Running 5/5
 - Network cluster_spark-network  Created                                                                 0.7s
 - Container master               Created                                                                 0.1s
 - Container worker-1             Created                                                                 0.1s
 - Container worker-2             Created                                                                 0.1s
 - Container worker-3             Created                                                                 0.1s
Attaching to master, worker-1, worker-2, worker-3
master    | Starting OpenBSD Secure Shell server: sshd.
master    | WARNING: /usr/hadoop-3.3.3/logs does not exist. Creating.
master    | 2022-08-02 14:22:57,040 INFO namenode.NameNode: STARTUP_MSG:
master    | /************************************************************
master    | STARTUP_MSG: Starting NameNode
master    | STARTUP_MSG:   host = master/10.5.0.2
master    | STARTUP_MSG:   args = [-format]
master    | STARTUP_MSG:   version = 3.3.3
master    | STARTUP_MSG:   classpath = /usr/hadoop-3.3.3/etc/hadoop:/usr/hadoop-3.3.3/share/hadoop/common/lib/reload4j-1
.2.18.3.jar:/usr/hadoop-3.3.3/share/hadoop/common/lib/netty-3.10.6.Final.jar:/usr/hadoop-3.3.3/share/hadoop/common/lib/j
etty-util-9.4.43.v20210629.jar:/usr/hadoop-3.3.3/share/hadoop/common/lib/curator-framework-4.2.0.jar:/usr/hadoop-3.3.3/s
```

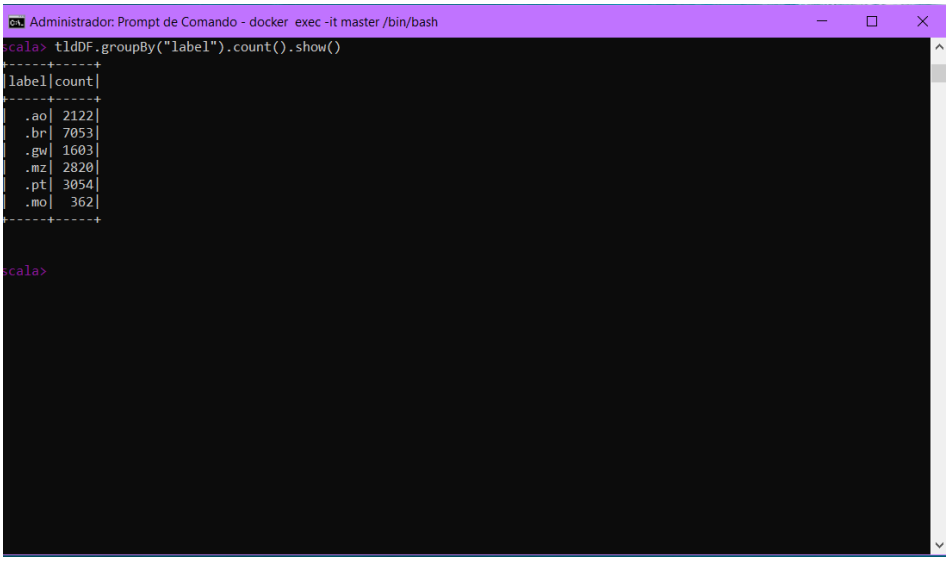
Nesse momento seu cluster já está montado.

5. Baixe o arquivo pt7-raw.zip disponível no link: https://abrir.link/GdgwQ
6. Copie a pasta descompactada para c:/cluster/user_data/pt7-raw
7. Copie os arquivos do PT7 para o HDFS:
    1. Abra outra aba do prompt de comando, mas, atenção, não feche a aba que está em execução, e execute o seguinte comando
    2. Entre na pasta do cluster:
        i. cd c:/cluster
    3. Execute: docker exec -it master/bin/bash
    4. hadoop fs -mkdir -p /bigdata/
    5. hadoop fs -put /user_data/pt7-raw hdfs://master:8020/bigdata/
10. Baixe e copie o arquivo **labels-pt7-raw.scala** para a pasta c:/cluster/user_data e execute no cmd:
    i. spark-shell --master spark://master:7077 -i /user_data/labels-pt7-raw.scala

Como resultado, será obtido um dataframe conforme imagens a seguir.



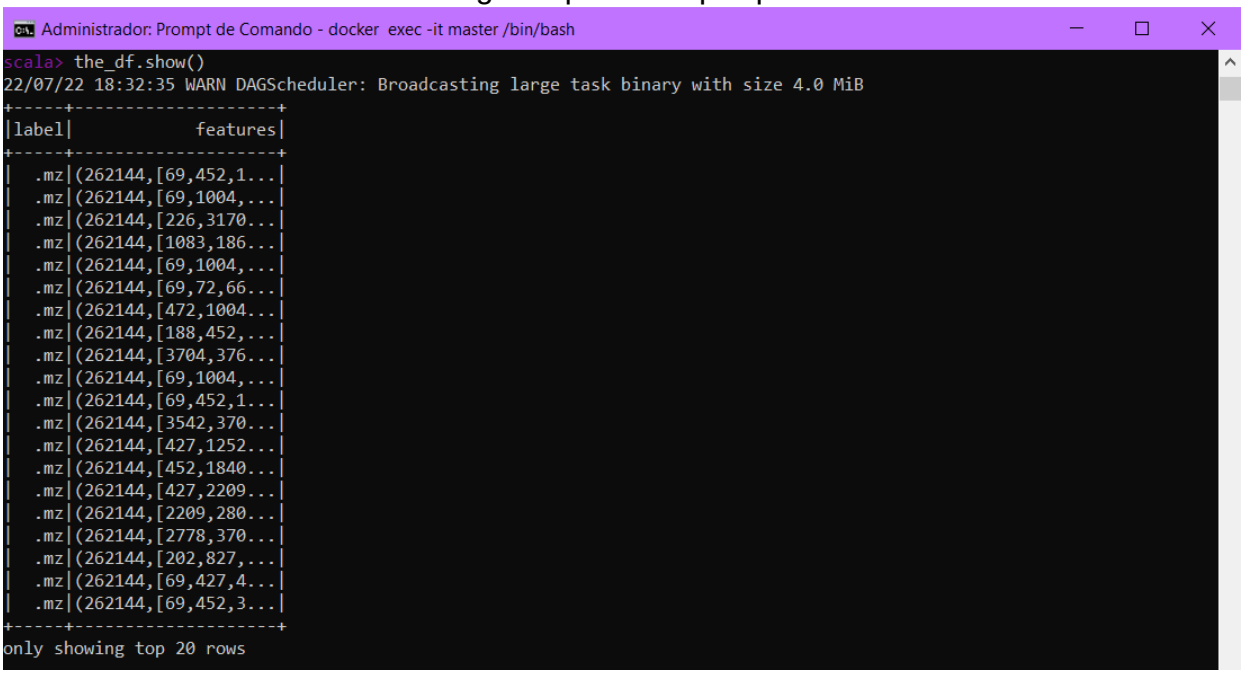
```
scala> tldDF.show()
+-----+--------------------+--------------------+
|label|                 url|          text64byte|
+-----+--------------------+--------------------+
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
|  .ao|http://mercado.co...|DQpMaWtlcw0KU3Vic...|
+-----+--------------------+--------------------+
only showing top 20 rows
```



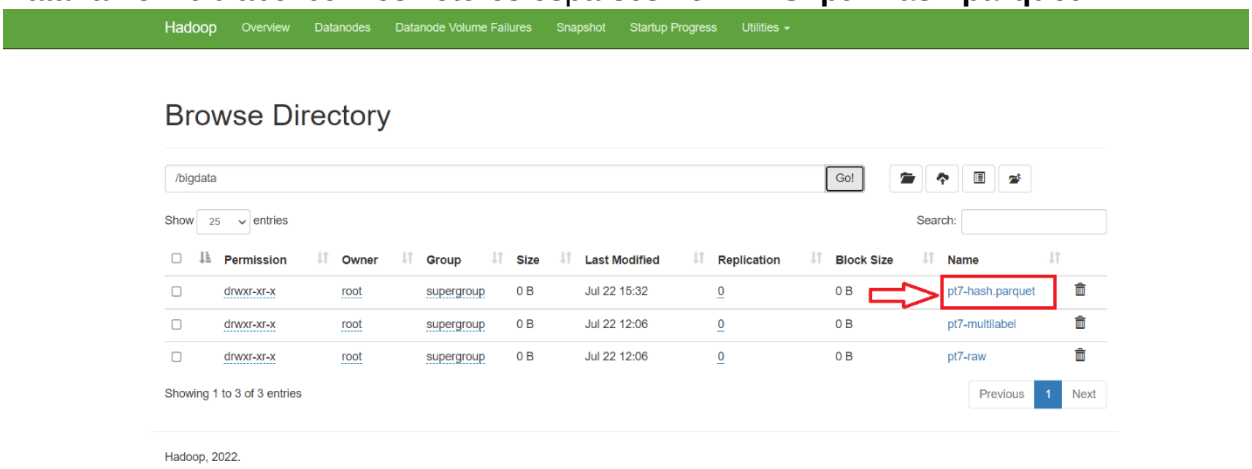
9. Baixe e copie o arquivo **etl-pt7.scala** para a pasta c:/cluster/user_data e execute no cmd:
    i. spark-shell --master spark://master:7077 -i /user_data/etl-pt7.scala

Como resultado, será obtido um dataframe conforme imagens a seguir. Neste ponto, o dataframe multilabel com os vetores esparsos será gravado no seu HDFS

no caminho hdfs://master:8020/bigdata/pt7-hash.parquet



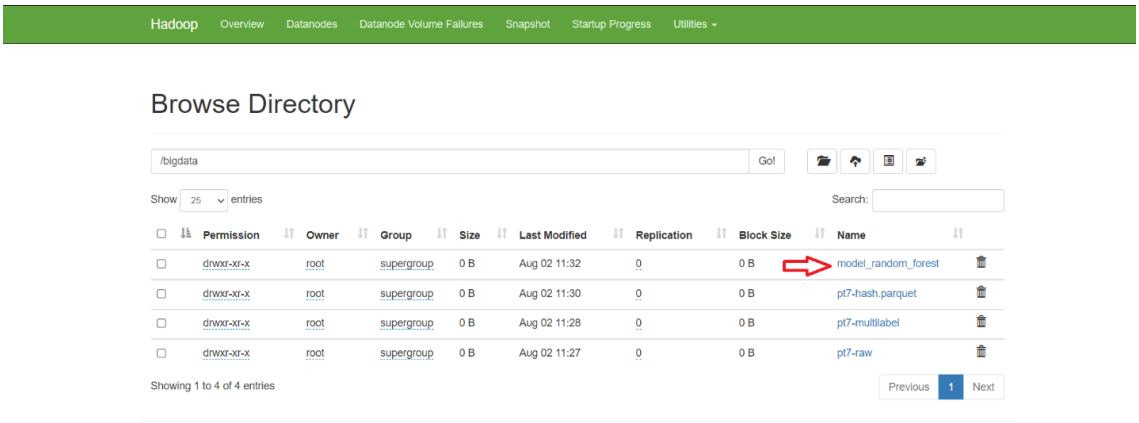
Dataframe multilabel com os vetores esparsos no HDFS: **pt7-hash.parquet**



Etapa 02: Treinar e Testar o Modelo Random Forest:

11. Baixe e copie o arquivo **random_forest_pt7web.scala** para a pasta c:/cluster/user_data e execute no cmd:
    a. spark-shell --master spark://master:7077 -i /user_data/random_forest_pt7web.scala

No final o resultado do modelo estará salvo no seu HDFS no caminho hdfs://master:8020/bigdata/modelo_random_forest, e as métricas serão salvas no user_data no arquivo metrics.txt



Saída do Modelo:

```
+----------+--------------------+
|prediction|            features|
+----------+--------------------+
|       3.0|(262144,[11,618,6...|
|       3.0|(262144,[37,452,6...|
|       3.0|(262144,[38,202,4...|
+----------+--------------------+
only showing top 3 rows
```

Métricas

F1-Score por label:

F1-Score(0.0) = 0.7781472139417565

F1-Score(1.0) = 0.4874077842788095

F1-Score(2.0) = 0.8260493292946776

Precision:

Precision(0.0) = 0.6380558428128231

Precision(1.0) = 0.9835728952772074

Precision(2.0) = 0.9953076120959332

Recall por label:

Recall(0.0) = 0.9970618480975466

Recall(1.0) = 0.3239770037199865

Recall(2.0) = 0.705991124260355

Sumário de métricas

Acurácia = 0.7603411513859275

weightedFMeasure = 0.7397598243283833

weightedPrecision = 0.8414480284406481

weightedRecall = 0.7603411513859274