



Fashion Recommender on Amazon

#Team 5:

Yichuan Zhang – zhang.yichu@husky.neu.edu

Houze Liu – liu.hou@husky.neu.edu



Project Goals

01

Practice in Scala

Code in Scala as much as possible

02

Process Data from Amazon Commercial API

Utilize Amazon API and XML parsing

03

Rank the Trend

Get most popular features that prevail in fashion trend

04

Compute fast on Spark

Utilize MapReduce

As a Buyer

● A Keyword

● Top 10 Colors

● Top 10 Brands

● Prices: High, Medium to Low

System Output

Our system tells you most popular colors, brands and how expensive are products on Amazon.

We help you make better shopping choices on Amazon website.

User Case1

As a Seller

● Keywords
(As many as you want)

System Output

- Popular Colors
- Popular Brands
- Prices of your Competitors'

Our system allows you build your Amazon products database according to your interests.
We help you make better marketing strategy, including pricing and customer-targeting

User Case2

Restrictions on Amazon API

Number of API calls

limited to 8,640 requests per day. once this limit is reached, will be limited to one request every ten seconds

01

Number of item pages

Only allow at most 10 pages of items, 10 items per page. Usually pages after 6th are blank

02

Url timeout

Amazon API plants timestamps in Url which will expire in 15 minutes .

03

Frequency of API calls

Amazon server may refuse API calls if Amazon thinks request is temporary overload

04



Methodology



Preprocessing

Generating Url with Amazon account signature



Acquiring

Calling Amazon Product Advertising API



Parsing

XML Format into Case Class



Feature Extracting

Get "Color", "Brand", "Price" and "Item Url" from all items



Mapping & Reducing

Using Apache Spark

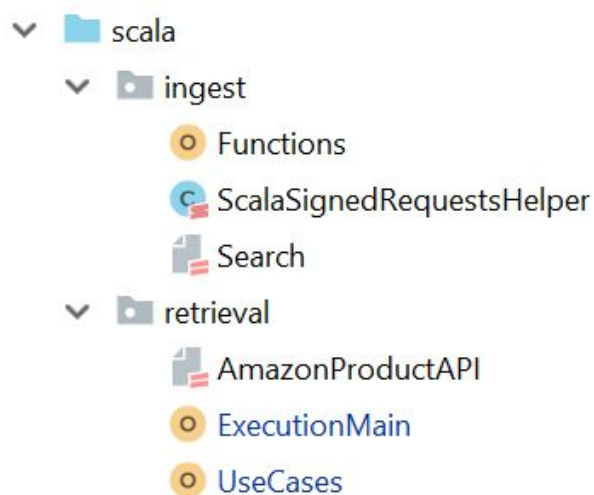


Visualization

Using Apache Zeppelin

Programming

- **Functions**
- **ScalaSignedRequestsHelper**
- **Search**
- **AmazonProductAPI**
- **ExecutionMain**
- **UseCase**
- **Unit Test**



Unit Test



Total 20 Unit Tests.....



69% Functions covered.....

- ✓ Test Results
 - ✓ UseCasesSpec
 - > ✓ Top_K_Colors
 - > ✓ Top_K_Brands
 - > ✓ Top_k_p()
 - > ✓ getLocalColors
 - > ✓ getLocalBrands
 - > ✓ getLocalPricesDouble
 - > ✓ getLocalPricesString
 - > ✓ vectorizedListOfDouble

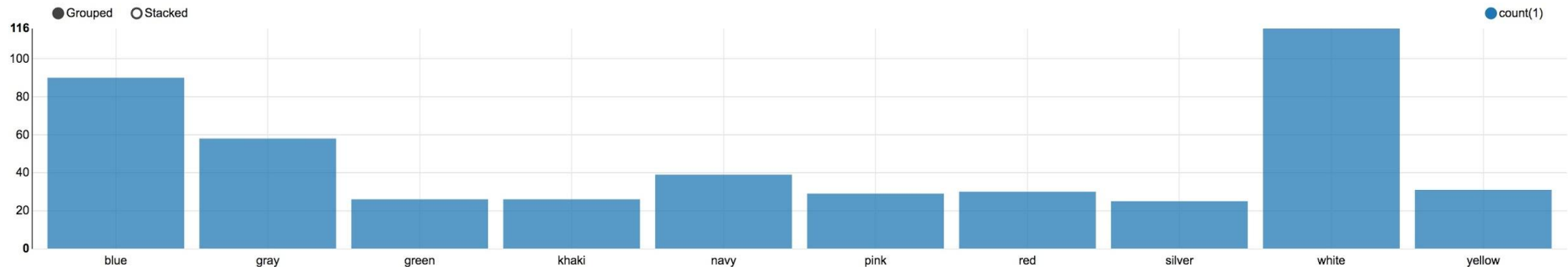
- ✓ Test Results
 - ✓ FunctionsSpec
 - > ✓ transfer
 - > ✓ futureToFutureTry
 - > ✓ itemToAttribute
 - > ✓ sortResultDescending
 - > ✓ sortResultDescending on RDD
 - > ✓ safeStringToDouble
 - > ✓ mergeAndSort
 - ✓ ScalaSignedRequestsHelperSpec
 - > ✓ sign
 - > ✓ hmac
 - > ✓ timestamp
 - > ✓ canonicalize
 - > ✓ percentEncodeRfc3986

Color



settings ▾

- ignored black



Took 1 sec. Last updated by anonymous at April 18 2018, 3:52:21 PM.

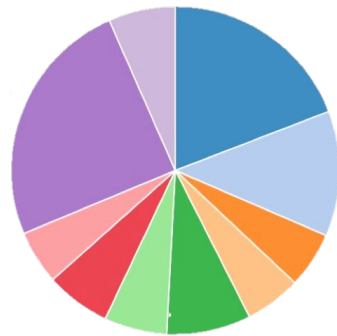
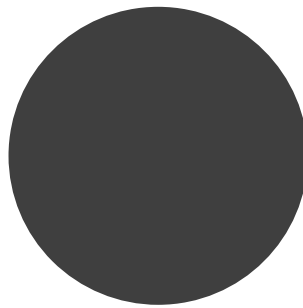
Black is the most Popular color !



Black, White and Blue are top-3 most popular colors. If you are going to sell products on Amazon, that is what you can refer to

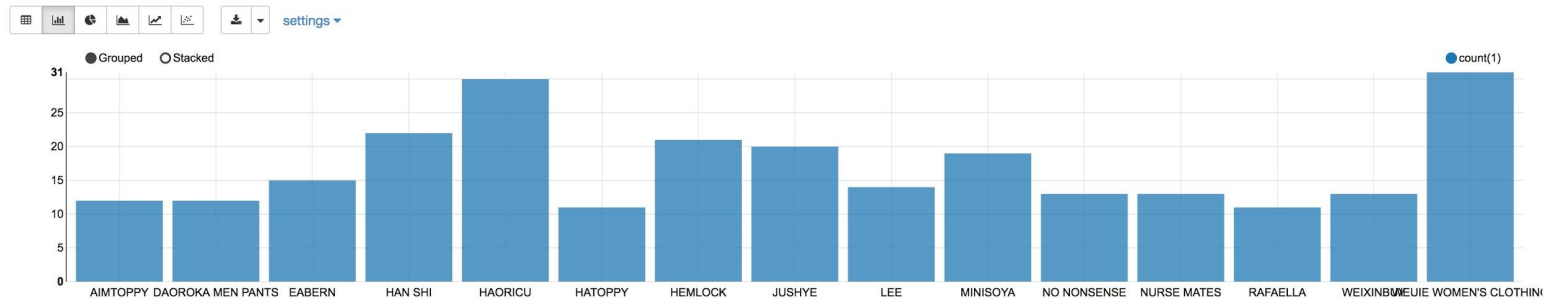
Black

Nearly as popular as other colors combined

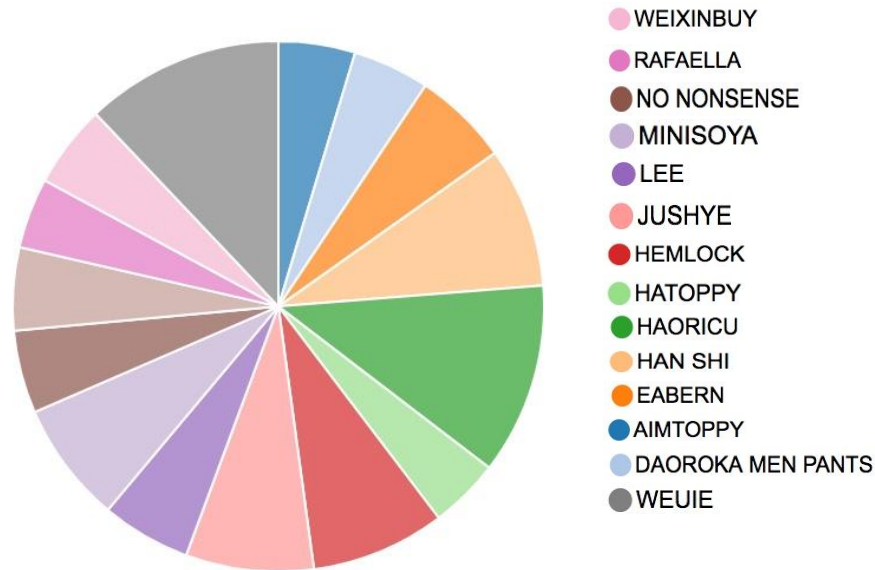




Brand

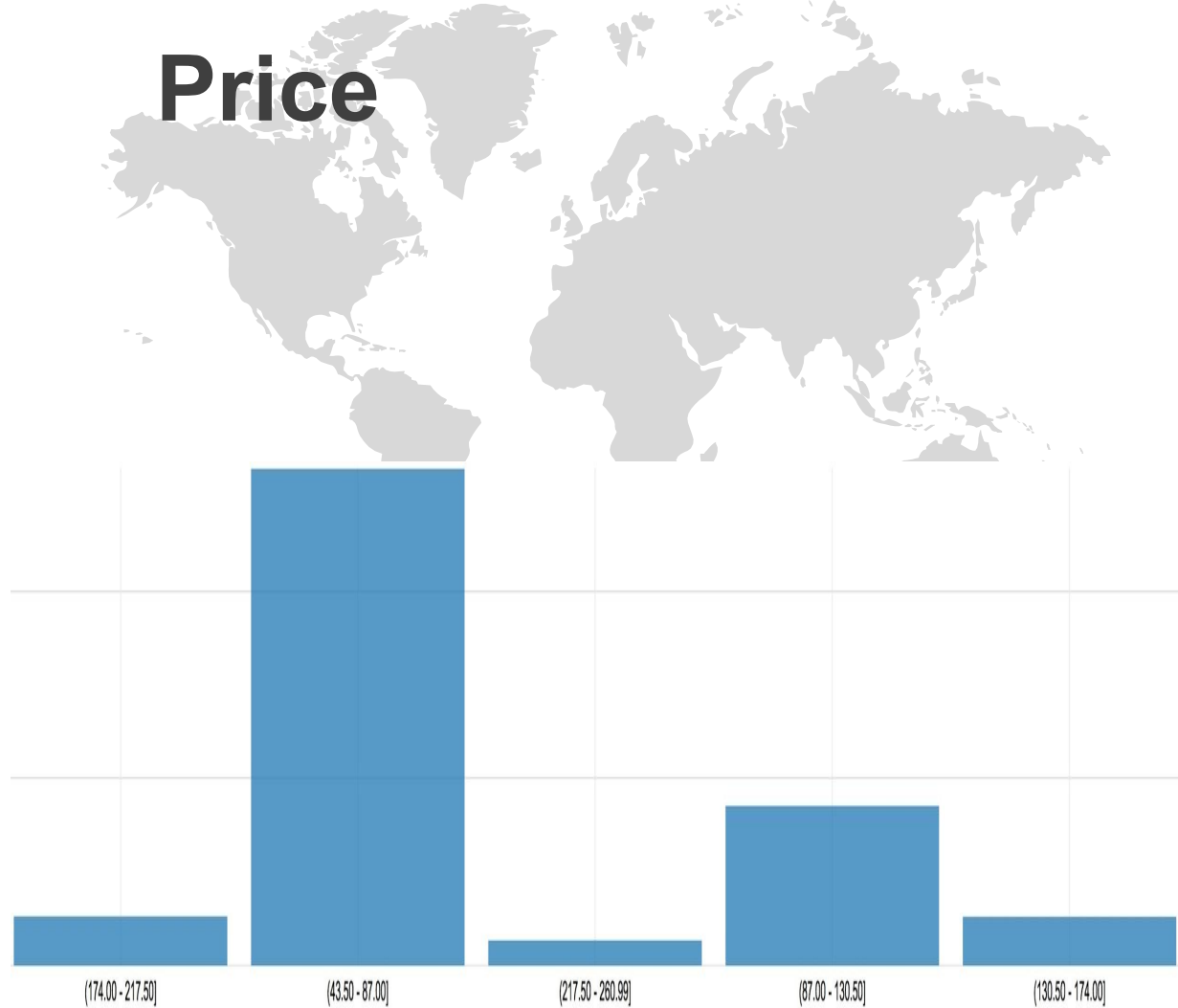


Took 1 sec. Last updated by anonymous at April 18 2018, 12:47:50 PM. (outdated)



Price

If you are selling products here, you can adopt low-cost marketing strategies



Acceptance Criteria

Verify Popularity

The most popular elements should frequently appear in Amazon Trend web page. ex: In 'clothes' catalog, 'black' clothes in the page should beats other color clothes in percentage, if 'black' tops popularity in our system. This goal should reached precision of 70%

[illegible]

Acceptance Criteria

Verify Popularity

Randomly select 10 type of clothing in Amazon Fashion, examine if black is the most popular(one page of items per type, total 300+ items)

Trousers ☒

Hats ☒

T-Shirt ☐

Shoes ☒

Sweaters ☒

Socks ☒

Skirts ☒

Coats ☒

Vests ☒

Scarves ☐



Criteria: 70%

Prediction Reached 8/10 = 80%

Acceptance Criteria

Verify Top Match

The result based on our system outputs should match items on Amazon's pages: 2 items for every page

Our system


output combinations



all items on one page

Amazon

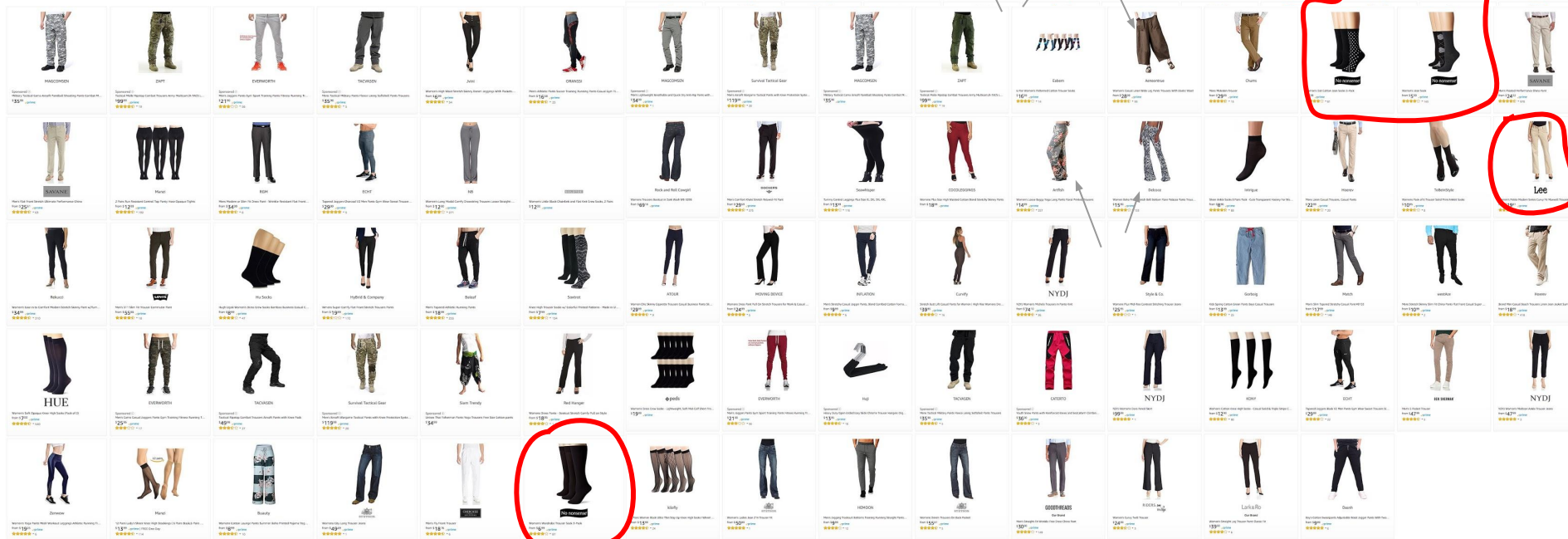
At least 2 results can be found on the page!

we tell you what kinds of products you should sell, instead of listing out searching result! 

Acceptance Criteria

Verify Top Match

8 items matched in first 3 pages



Screenshot of Program Running



User case 1: KNN trained on prices

Use case 2: After 20 keywords as input

```
java.io.IOException: Server returned HTTP response code: 503 for URL: http://webservices.amazon.com/onca/xml?AWSAccessKeyId=AKIAJVADVVC5WAOOQHA&AssociateTag=scalapr
java.io.IOException: Server returned HTTP response code: 503 for URL: http://webservices.amazon.com/onca/xml?AWSAccessKeyId=AKIAJVADVVC5WAOOQHA&AssociateTag=scalapr
java.io.IOException: Server returned HTTP response code: 503 for URL: http://webservices.amazon.com/onca/xml?AWSAccessKeyId=AKIAJVADVVC5WAOOQHA&AssociateTag=scalapr
java.io.IOException: Server returned HTTP response code: 503 for URL: http://webservices.amazon.com/onca/xml?AWSAccessKeyId=AKIAJVADVVC5WAOOQHA&AssociateTag=scalapr
java.io.IOException: Server returned HTTP response code: 503 for URL: http://webservices.amazon.com/onca/xml?AWSAccessKeyId=AKIAJVADVVC5WAOOQHA&AssociateTag=scalapr
81/82 accomplished
82/82 accomplished
(black,13998)
(white,5447)
(blue,3343)
(gray,1795)
(red,1652)
(pink,1442)
(navy,1387)
(green,1009)
(yellow,781)
(grey,684)
(HAORICU,773)
(HEMLOCK,640)
(HANES,546)
(HAN SHI,529)
(THE CHILDREN'S PLACE,465)
(JUSHYE,426)
(WEUIE WOMEN'S CLOTHING,367)
(ALO YOGA,360)
(GODDESSVAN,356)
(AIMTOPPY,342)
List(20.99, 19.99, 79.95, 13.99, 114.00, 24.99, 34.90, 22.99, 36.16, 46.00, 14.99, 16.85, 18.99, 49.00, 7.90, 10.59, 10.99, 10.45, 23.99, 15.99, 5.99, 26.99, 38.99,
Process finished with exit code 0
Very Cheap,List(((6.17 - 7.05],1308), ((5.29 - 6.17],1205), ((7.93 - 8.81],920), ((3.52 - 4.41],764), ((4.41 - 5.29],696), ((7.05 - 7.93],681), ((2.64 - 3.52],435), ((0.88 - 1.76],420), ((1.76 - 2.64],283), ((0.00 - 0.88],71)))
Very High,List(((29.56 - 70.10],2921), ((70.10 - 110.65],502), ((110.65 - 151.19],293), ((151.19 - 191.73],112), ((232.28 - 272.82],65), ((272.82 - 313.36],63), ((191.73 - 232.28],63), ((313.36 - 353.90],17), (other,15), ((394.45 - 434.99],
Medium,List(((14.90 - 15.21],727), ((15.84 - 16.15],495), ((16.77 - 17.09],400), ((16.46 - 16.77],189), ((15.53 - 15.84],75), ((14.59 - 14.90],48), ((17.40 - 17.71],33), ((17.09 - 17.40],30), ((16.15 - 16.46],18), ((15.21 - 15.53],9)))
High,List(((18.92 - 20.04],1117), ((24.52 - 25.64],443), ((17.80 - 18.92],438), ((20.04 - 21.16],354), ((22.28 - 23.40],335), ((23.40 - 24.52],220), ((26.76 - 27.88],201), ((21.16 - 22.28],180), ((25.64 - 26.76],174), ((27.88 - 29.00],173))
Low,List(((9.44 - 10.00],1246), ((12.81 - 13.37],728), ((8.88 - 9.44],682), ((10.56 - 11.12],673), ((11.69 - 12.25],511), ((13.93 - 14.49],408), ((10.00 - 10.56],396), ((11.12 - 11.69],275), ((12.25 - 12.81],208), ((13.37 - 13.93],153)))
```


Code Repository



<https://github.com/YichuanZhang/CSYE7200-SPRING2018-TEAM5>

YichuanZhang / CSYE7200-SPRING2018-TEAM5

Unwatch 1

Star 0

Fork 1

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Project Repository for CSYE7200

Scala 100.0%

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



HouHouHouHouHouHou add zeppelinNotebook and finalPresentation

Latest commit 614ae77 2 minutes ago

ZeppelinNotebook

add zeppelinNotebook and finalPresentation

2 minutes ago

database

add zeppelinNotebook and finalPresentation

2 minutes ago

src

add zeppelinNotebook and finalPresentation

2 minutes ago

test/resources

update

8 days ago

.gitignore

Update .gitignore

23 days ago

Final presentation for CSYE7200 team5 sprin...

add zeppelinNotebook and finalPresentation

2 minutes ago

Planning presentation for CSYE7200 team5 s...

update slide

27 days ago

README.md

Update README.md

a month ago

brand.txt

add zeppelinNotebook and finalPresentation

2 minutes ago

build.sbt

add unit test

2 days ago

color.txt

add zeppelinNotebook and finalPresentation

2 minutes ago

item.txt

add zeppelinNotebook and finalPresentation

2 minutes ago

price.txt

add zeppelinNotebook and finalPresentation

2 minutes ago



Link

Reference



- Course Repo
- Amazon API:
<https://docs.aws.amazon.com/AWSECommerceService/latest/DG/Welcome.html>
- Amazon Account constraints:
<https://docs.aws.amazon.com/AWSECommerceService/latest/DG/TroubleshootingApplications.html>
- How to generate signed Url:
<https://github.com/ebayopensource/turmeric-runtime/blob/master/examples/turmeric-example-item-search/ItemSearchConsumers/src/org/ebayopensource/turmeric/demo/consumer/SignedRequestHelper.java>
- Spark:
<https://spark.apache.org/docs/latest/quick-start.html>





THANK YOU