

Saudi_Ecommerce Dataset

Supervisor:
Dr. Sameh

Student Name:
Raghad Alshalan
Wejdan Othman
Wafa Othman
Khafuq Abdullah

Content

- 2. CONTENT.....
- 3. DATASET.....
- 4. INTROUDCTION.....
- 5. CONTENT
- 6. DATA TYPE.....
- 7. BENERITS SPARK.....
- 8. ANALYSIS WITH HADOOP
SPARK.....
- 16.FILE USED FOR ANAYSIS.....
- 17.REFERENCE



Data Sets

This dataset for people who are thinking of entering the e-commerce business in Saudi Arabia or just curious about the sector, this data set will come in handy. It's scrapped from Maroof website which is an initiative started by the Ministry of Commerce to help our customers to recognize official stores with legal grounds.

A high-quality eCommerce store dataset delivers extensive firmographic insights for major online retailers as well as niche players. Companies use this data to power competitor analysis and monitor competitor activities.



Introduction

E-commerce data analysis refers to the process of gathering, cleaning, transforming, and modeling data from e-commerce platforms to gain insights and make informed decisions. The goal of analyzing e-commerce data is to understand customer behavior, market trends, and business performance to drive growth and improve decision-making. E-commerce data can come from various sources, such as website analytics, customer demographics, sales data, and marketing campaigns. Then this data is processed and analyzed using various techniques, including statistical analysis, machine learning algorithms, and data visualization tools. In this project, we analyzed your collection of data from the Kaggle program using spark, and we got the best results.



Content

Each row is a Store. Columns are as follows:

- Name_ENG: Name of the store in English
- Name_AR: Name of the store in Arabic
- CR: Commercial Registration number
- Category: Store chosen category within Maroof limits
- Website: Link to the Store main selling channel
- Instagram: Store Instagram Account link
- Twitter: Store Twitter Account link
- Email: Store email address
- Phone Number: Store phone number
- Activity: Status of the business as shown in Maroof
- Rating: Store rating out of 10
- Num_Ratings: Number of ratings the store has

Data Type

It is Structured data. the data conforms to a data model, has a well define structure, follows a consistent order and can be easily accessed and used by a person or a computer program. it was stored in well-defined schemas (Databases). It is generally tabular with column and rows that clearly define its attributes. and can used SQL (Structured Query language) to manage this structured data stored in databases.

eCommerce store data: This dataset provides details about eCommerce vendors, sellers, and platforms, creating a comprehensive database of competitors.



Benefits spark

Speed: Spark is designed to be fast, with in-memory data processing capabilities that can significantly speed up big data analysis tasks compared to traditional MapReduce.

Flexibility: Spark supports multiple data sources and data formats, including structured, semi-structured, and unstructured data.

Scalability: Spark can be easily scaled out to handle large data sets, allowing for quick and efficient processing of big data.

Ease of Use: Spark has a user-friendly API and a variety of libraries, such as Spark SQL, Spark Streaming, MLlib, and GraphX, which make it easier for data scientists and engineers to perform complex data analysis tasks.



Analysis with Hadoop spark

1. Getting start spark (scala).
 2. Import file “Saudi_Ecommerce.csv” to spark (Data Frame) and show it.

```

scala> df.groupBy(col("Name_ENG").rlike(".*")).count.show()
+-----+-----+
|RLIKE(Name_ENG, .*)|count|
+-----+-----+
|      null| 391|
|     true| 5010|
+-----+-----+

scala> df.agg(countDistinct("Name_ENG")).show()
+-----+
|count(Name_ENG)|
+-----+
|      5009|
+-----+


scala> df.groupBy(col("Name_AR").rlike(".*")).count.show()
+-----+-----+
|RLIKE(Name_AR, .*)|count|
+-----+-----+
|     true| 5401|
+-----+-----+


scala> df.agg(countDistinct("Name_AR")).show()
+-----+
|count(Name_AR)|
+-----+
|      5400|
+-----+

```

1. Count valid and missing values of Name_ENG column.
2. Count Unique values of Name_ENG column.
3. Count valid and missing values of Name_AR column.
4. Count Unique values of Name_AR column.

```
C:\ Command Prompt - spark-shell
+-----+
| RLIKE(CR, .*) | count |
+-----+-----+
|      null | 3491 |
|     true  | 1910 |
+-----+-----+


scala> df.agg(countDistinct("CR")).show()
+-----+
| count(CR) |
+-----+
| 1868 |
+-----+


scala> df.select(mean("CR")).show()
+-----+
|      avg(CR) |
+-----+
| 2.5890778503376966E9 |
+-----+


scala> df.select(max("CR")).show()
+-----+
|      max(CR) |
+-----+
| 5.957100556E9 |
+-----+


scala> df.select(min("CR")).show()
+-----+
|      min(CR) |
+-----+
| 1.01001149E9 |
+-----+
```

1. Count valid and missing values of CR column.
2. Count Unique values of CR column.
3. Calculate average, maximum and minimum value of CR column.

```
Command Prompt - spark-shell

scala> df.groupBy(col("Category").rlike(".*")).count.show()
+-----+-----+
|RLIKE(Category, .*)|count|
+-----+-----+
|          true| 5401|
+-----+-----+
r

e
scala> df.agg(countDistinct("Category")).show()
+-----+
|count(Category)|
+-----+
|      15|
+-----+


scala> df.groupBy(col("Website").rlike(".*")).count.show()
+-----+-----+
|RLIKE(Website, .*)|count|
+-----+-----+
|          true| 5401|
+-----+-----+


scala> df.agg(countDistinct("Website")).show()
+-----+
|count(Website)|
+-----+
|      5215|
+-----+


g
scala> df.groupBy(col("Instagram").rlike(".*")).count.show()
+-----+-----+
|RLIKE(Instagram, .*)|count|
+-----+-----+
|        null| 3788|
|       true| 1613|
+-----+-----+


scala> df.agg(countDistinct("Instagram")).show()
+-----+
```

1. Count valid and missing values of Category column.
2. Count Unique values of Category column.
3. Count valid and missing values of Website column.
4. Count Unique values of Website column.
5. Count valid and missing values of Instagram column.
6. Count values of Instagram column.

```
Command Prompt - spark-shell
+-----+
|      null| 3788|
|      true| 1613|
+-----+-----+


scala> df.agg(countDistinct("Instagram")).show()
+-----+
|count(Instagram)|
+-----+
|      1607|
+-----+


scala> df.groupBy(col("Twitter").rlike(".*")).count.show()
+-----+-----+
|RLIKE(Twitter, .*)|count|
+-----+-----+
|      null| 4662|
|      true| 739|
+-----+-----+


scala> df.agg(countDistinct("Twitter")).show()
+-----+
|count(Twitter)|
+-----+
|      738|
+-----+


scala> df.groupBy(col("Email").rlike(".*")).count.show()
+-----+-----+
|RLIKE>Email, .*)|count|
+-----+-----+
|      true| 5401|
+-----+-----+


scala> df.agg(countDistinct("Email")).show()
+-----+
|count>Email)|
+-----+
|      5216|
+-----+
```

1. Count valid and missing values of Twitter column.
2. Count Unique values of Twitter column.
3. Count valid and missing values of email column.
4. Count values of email column.

```
Command Prompt - spark-shell
+-----+
| RLIKE(Phone Number, .*)|count|
+-----+-----+
| true | 5401 |
+-----+-----+


scala> df.agg(countDistinct("Phone Number")).show()
+-----+
| count(Phone Number) |
+-----+
| 5085 |
+-----+


scala> df.groupBy(col("Activity").rlike(".*")).count.show()
+-----+-----+
| RLIKE(Activity, .*)|count|
+-----+-----+
| true | 5401 |
+-----+-----+


scala> df.agg(countDistinct("Activity")).show()
+-----+
| count(Activity) |
+-----+
| 2 |
+-----+


scala> df.groupBy(col("Activity").rlike("Active")).count.show()
+-----+-----+
| RLIKE(Activity, Active)|count|
+-----+-----+
| true | 5401 |
+-----+-----+
```

1. Count valid and missing values of Phone_Number column.
2. Count Unique values of Phone_Number column.
3. Count valid and missing values of Activity column.
4. Count values of Activity column.

```
C:\ Command Prompt - spark-shell

scala> val active = df.where(df("Activity") === "Active").count()
active: Long = 4893

scala> val not_active = df.where(df("Activity") === "Not Active").count()
not_active: Long = 508

scala> df.groupBy(col("Rating").rlike("."*")).count.show()
+-----+----+
|RLIKE(Rating, .*)|count|
+-----+----+
|      true| 5401|
+-----+----+

scala> df.agg(countDistinct("Rating")).show()
+-----+
|count(Rating)|
+-----+
|      53|
+-----+


scala> df.select(mean("Rating")).show()
+-----+
|    avg(Rating)|
+-----+
|1.1772264395482313|
+-----+


scala> df.select(max("Rating")).show()
+-----+
|max(Rating)|
+-----+
|     10.0|
+-----+


scala> df.select(min("Rating")).show()
+-----+
|min(Rating)|
+-----+
|     0.0|
+-----+
```

1. Count stores number that are Active.
2. Count store number that are NOT Active.
3. Count valid and missing values of Rating column.
4. Count Unique values of Rating column.
5. Calculate average, maximum and minimum value of Rating column.

```

Command Prompt - spark-shell
maxCount: Long = 458

scala> val minCount = df.where(df("Rating") === "0").count()
minCount: Long = 4662

scala> df.groupBy(col("Num_Ratings").rlike(".*")).count.show()
+-----+-----+
|RLIKE(Num_Ratings, .*)|count|
+-----+-----+
|           true| 5401|
+-----+-----+

scala> df.select(mean("Num_Ratings")).show()
+-----+
| avg(Num_Ratings)|
+-----+
|0.8579892612479171|
+-----+


scala> df.select(max("Num_Ratings")).show()
+-----+
|max(Num_Ratings)|
+-----+
|      163|
+-----+


scala> df.select(min("Num_Ratings")).show()
+-----+
|min(Num_Ratings)|
+-----+
|      0|
+-----+


scala> val maxCount = df.where(df("Num_Ratings") === "163").count()
maxCount: Long = 1

scala> val maxCount = df.where(df("Num_Ratings") === "0").count()
maxCount: Long = 4662

scala>

```

1. Count number of stores that have maximum Rating.
2. Count number of stores that have minimum Rating.
3. Count valid and missing values of Num_Rating column
4. Calculate average, maximum and minimum value of Num_Rating column.
5. Count number of stores that have maximum Num_Rating.
6. Count number of stores that have minimum Num_Rating.

Data file used for analysis

| A | B | C | D | E | F | G | H | I | J | K |
|-----------------------------------|----------------------|-----------------------|---|---|-----------|---------|---------------------------|--------------|----------|--------|
| Name_Eng | Name_AR | CR | Category | Websit | Instagram | Twitter | Email | Phone Number | Activity | Rating |
| SpotifyBox | سبوتيفي بوكس | ايجار | موقع ايجار | https://www.spotifybox.com/trainingcenter/en/getbox | | | Spotifybox@gmail.com | 531060415 | Active | 0 |
| Woolworths | وول وورثز | بيع وشراء بقالة | موقع ايجار | https://www.woolworths.ae | | | Woolworths1@gmail.com | 53150375 | Active | 2 |
| Shopee | شپی | بيع وشراء بقالة ملابس | موقع ايجار | https://www.shopee.ae | | | Woolworths1@gmail.com | 53150375 | Active | 2 |
| BusinessHub | بريزز هاب | بيع وشراء بقالة ملابس | موقع ايجار | https://www.businesshub.ae | | | Info@businesshub.ae | 531784521 | Active | 0 |
| BusinessHub | بريزز هاب | بيع وشراء بقالة ملابس | موقع ايجار | https://www.businesshub.ae | | | Info@businesshub.ae | 531784521 | Active | 0 |
| VisionaryVines | فيشناري فين | بيع وشراء بقالة ملابس | موقع ايجار | https://www.visionaryvines.ae | | | VisionaryVines1@gmail.com | 53121210 | Active | 0 |
| ForAllAccounts | فور الاقوام | بيع وشراء بقالة ملابس | موقع ايجار | https://www.forallaccounts.ae | | | forallaccounts1@gmail.com | 53120379 | Active | 0 |
| ForAllAccounts | فور الاقوام | بيع وشراء بقالة ملابس | موقع ايجار | https://www.forallaccounts.ae | | | forallaccounts1@gmail.com | 53120379 | Active | 0 |
| UAE Bank for Economic Development | بنك الامارات للتنمية | بيع وشراء بقالة ملابس | موقع ايجار | https://www.uae-bank.ae | | | grahamhughes@gmail.com | 53121210 | Active | 0 |
| GreenFinance | غرن فاننس | بيع وشراء بقالة ملابس | موقع ايجار | https://www.greenfinance.ae | | | greenfinance1@gmail.com | 53105015 | Active | 0 |
| Almarai | المراني | بيع وشراء بقالة ملابس | موقع ايجار | https://www.almarai.ae | | | almarai1@gmail.com | 53120379 | Active | 0 |
| Qatar Store | قرط ستور | بيع وشراء بقالة ملابس | موقع ايجار | https://www.qatarsstore.ae | | | qatarsstore1@gmail.com | 53120379 | Active | 0 |
| Cafes | كافيه | موقع ايجار | https://www.cafes.ae | | | | YTheatre@gmail.com | 53200850 | Active | 0 |
| Drop selling products | دروپ سيلينج بوكليت | موقع ايجار | https://www.drop.com | | | | anthonyc02@gmail.com | 53142516 | Active | 0 |
| DropXPro | دروپ إكس برو | موقع ايجار | https://www.dropxpro.com | | | | drexpro_02@gmail.com | 53200124 | Active | 0 |
| LaptopCity | لابتوب سيتي | موقع ايجار | https://www.laptopcity.ae | | | | laptopcity1@gmail.com | 53174460 | Active | 0 |
| Swiss Army | سويس آرمي | موقع ايجار | https://www.swiss-army.ae | | | | swiss2015@outlook.com | 53109512 | Active | 0 |
| Red Candy | رید كندي | موقع ايجار | https://www.redcandy.ae | | | | RedCandy123@gmail.com | 53149137 | Active | 0 |
| Fair | فير | موقع ايجار | https://www.fair.ae | | | | zourmou2029@gmail.com | 53102144 | Active | 0 |
| Unilever (Uk) Group | لينيلر (أع) جروب | موقع ايجار | https://www.unilever.ae | | | | UkGroup123@gmail.com | 53177211 | Active | 0 |
| Shopping Direct | شوبينج ديركت | موقع ايجار | https://www.shoppingdirect.ae | | | | rober1990075@gmail.com | 53199057 | Active | 0 |
| One Volkswagen AG | ونه فولكس واگن | موقع ايجار | https://www.volkswagen.ae | | | | o.volkswagen@takleem.com | 53142216 | Active | 0 |
| Magical Pen Agency | ماجيكل بىن اجنسى | موقع ايجار | https://www.magicalpen.com | | | | magicalpen191@gmail.com | 53168020 | Active | 0 |
| Bonc Taweeq | بونك تويق | موقع ايجار | https://www.bonctaweeq.ae | | | | bonctaweeq1@gmail.com | 53133218 | Active | 0 |
| Print | پرینت | موقع ايجار | https://www.print.ae | | | | Print123@gmail.com | 53149146 | Active | 0 |
| Pure & Simple | پور و سپل | موقع ايجار | https://www.puresimple.ae | | | | puresimple123@gmail.com | 53121215 | Active | 0 |
| ALFARIDH STORE | الفاريد | موقع ايجار | https://www.alfaridh.ae | | | | rafah123@gmail.com | 53133476 | Active | 0.5 |
| Almarai | المراني | موقع ايجار | https://www.almarai.ae | | | | almarai1@gmail.com | 53120379 | Active | 0.5 |
| Almarai | المراني | موقع ايجار | https://www.almarai.ae | | | | almarai1@gmail.com | 53120379 | Active | 0.5 |

Reference

<https://www.kaggle.com/datasets/abdullahalothman/saudi-e-commerce-data-set?resource=download>

