Kristianstad University
SE-291 88 Kristianstad
+46 44-250 30 00
www.hkr.se

Big Data Analytics project report
Autumn Semester 2024
Faculty of Natural Sciences
Department of Computer Science

Big Data Analytics

Amazon Reviews Analysis iOS vs Android users, the eternal feud!

Sam El Saati & Mohamad Alloosh

Content

Introduction

This project was undertaken to fulfill the requirements of applying big data analytical techniques to real-world datasets, as specified in the course. The aim was to develop a framework for analyzing and deriving meaningful insights from a large dataset while addressing key learning outcomes such as data preprocessing, analytics, and visualization. Initially, we explored a dataset that was not raw and lacked the necessary characteristics to meet the project objectives.

We subsequently shifted to the Amazon Reviews dataset, which better aligned with the project's scope. Among its many categories, we selected "Cell Phones and Accessories," a subset comprising approximately 9GB of data, as it was both substantial in size and relevant to our analytical goals. This category allowed us to explore user reviews, ratings, and behavioral patterns for Android and iOS users, enabling us to create actionable insights and fulfill the analytical and representational requirements outlined in the project.

Dataset Overview

Dataset Description

The dataset "Cell_Phones_and_Accessories" was sourced from Amazon Reviews Dataset and contains information about user reviews, ratings, and purchase details. The dataset's total size is approximately 9GB, making it ideal for big data analytics.

Key attributes of the dataset:

- **User Information**: Includes anonymized user IDs.
- **Review Details**: Contains text reviews, ratings (1-5 stars), and timestamps.
- **Product Details**: Includes product IDs.
- **Meta Information**: Includes helpfulness ratings and verified purchase indicators.

This dataset was selected due to its relevance to the project's objectives and its potential to generate meaningful insights through advanced analytical techniques.

Method

The first step in this project was setting up the Hadoop environment to manage and process the large dataset efficiently. This involved downloading and installing Hadoop, which proved to be a meticulous process. Configuring environment variables and ensuring compatibility with Java installations posed challenges, as mismatched versions often led to runtime errors. Once Hadoop was operational, HDFS (Hadoop Distributed File System) was initialized to provide distributed storage for the dataset.

After setting up Hadoop, the next step involved installing PySpark within an Anaconda virtual environment. This setup was critical for leveraging Spark's capabilities for data processing and analysis. Configuring the Spark session posed challenges due to the limited resources of the laptop being used, requiring extensive research and fine-tuning of Spark configurations to ensure compatibility and optimize performance.

```
# Step 1: Initialize Spark Session
spark = SparkSession.builder \
    .appName("CellPhonesAccessoriesAnalysis") \
    .config("spark.network.timeout", "1000s") \
    .config("spark.executor.heartbeatInterval", "900s") \
    .config("spark.driver.memory", "6g") \
    .config("spark.executor.memory", "7g") \
    .config("spark.executor.memoryOverhead", "2g") \
    .config("spark.hadoop.fs.defaultFS", "hdfs://localhost:9000") \
    .getOrCreate()
```

Figure 1- Configuring the Spark session

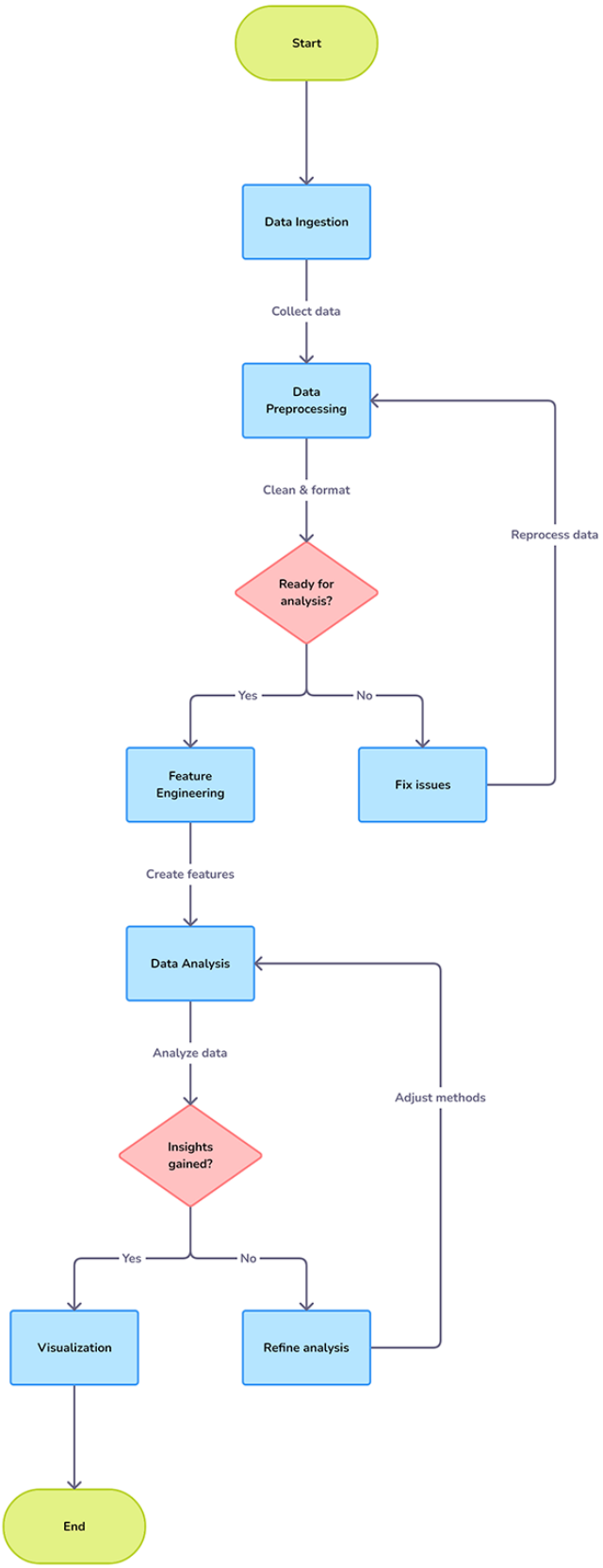
The raw JSON dataset was uploaded into HDFS, ensuring secure and scalable access for preprocessing and analysis.

```
# Step 2: Define HDFS file path
file_path = "hdfs://localhost:9000/project/Cell_Phones_and_Accessories.jsonl"

# Step 3: Load JSONL file into Spark DataFrame
df_spark = spark.read.json(file_path)
```

Figure 2- Uploading JSON dataset to HDFS

Workflow Diagram



Preprocessing and Workflow

[1] Data Conversion & Feature Selection

a. Converted the raw JSON data to CSV format for easier manipulation and analysis.
b. Selected relevant features such as ratings and review text, etc..

```
# Define the output path for the CSV
output_path = "hdfs://localhost:9000/project/Cell_Phones_and_Accessories_selected.csv"

# Select only the required features
required_columns = ["asin", "rating", "text", "timestamp", "verified_purchase", "title"]
df_selected = df_spark.select(*required_columns)

# Save the selected columns to a CSV file
df_selected.write.csv(output_path, header=True, mode="overwrite")
```

Figure 3- Converting to CSV and Feature Selection

	asin	rating	text	timestamp	verified_purchase	title
0	B08L6L3X1S	4.0	I bought this bc I thought it had the nice white background. Turns out it's clear & since my phone is blue it doesn't look anything like this. If I had known that I would have purchased something else. It works ok.	1612044451196	true	No white background! It's clear!
1	B079BPGF6C	5.0	Perfect. How pissed am I that I recently paid $20 for 1 Fitbit cable and promptly lost the damned thing? Extremely pissed! I keep the spare in my medicine bag so hopefully I won't lose it and my grandson can't get to it and try to use it as a belt or a dog leash or any of the other nutty things he's been using the other one for.	1534443517349	true	Awesome! Great price! Works well!

Figure 4- Showing first rows of the dataset

[2] **Data Cleaning**:

a. Addressed missing values by applying imputation techniques.
b. Checked for outliers and removed irrelevant timestamps.
c. Eliminated invalid timestamps.

```
Number of rows with missing values:
+----+------+-----+---------+-----------------+-----+
|asin|rating| text|timestamp|verified_purchase|title|
+----+------+-----+---------+-----------------+-----+
|   0|     0|22275|        0|                0|    0|
+----+------+-----+---------+-----------------+-----+

Number of missing values after dropping rows with missing values:
+----+------+----+---------+-----------------+-----+
|asin|rating|text|timestamp|verified_purchase|title|
+----+------+----+---------+-----------------+-----+
|   0|     0|   0|        0|                0|    0|
+----+------+----+---------+-----------------+-----+

Total number of rows after dropping missing values: 20790670
```

[3] **Data Transformation**:

a. Tokenized review text and removed stop words, special characters, and unnecessary symbols [4].

```
# Tokenize and remove stopwords
tokenizer = Tokenizer(inputCol="text", outputCol="text_tokens")
df_tokenized = tokenizer.transform(df_cleaned)
remover = StopWordsRemover(inputCol="text_tokens", outputCol="text_cleaned")
df_cleaned = remover.transform(df_tokenized)
```

Figure 5- Tokenized review text

[4] **Feature Engineering**:

Created new columns for analysis, including:

```
+--------------------+--------------------+--------------------+------+---------+--------+
|               title|                text|        text_cleaned|rating|sentiment|category|
+--------------------+--------------------+--------------------+------+---------+--------+
|Worked but took a...|overall very happ...|[overall, happy, ...|   5.0| positive|     iOS|
|Works Great with ...|this item works g...|[item, works, gre...|   5.0| positive|     iOS|
|A bit complicated...|a bit complicated...|[bit, complicated...|   2.0| negative|     iOS|
|            One Star|fell apart right ...|[fell, apart, rig...|   1.0| negative|     iOS|
```

a. **Sentiment labels**: Classified as positive, neutral, or negative based on rating scores.
b. **Platform Category**: Categorized reviews into "Android" or "iOS" using keywords in titles and text body, accounting for possible typos.

These preprocessing steps laid the foundation for subsequent analysis, ensuring data quality and relevance.

[5] **Data Analysis**:

a. Conducted sentiment analysis using a rule-based approach by mapping review ratings to sentiment labels.
b. Performed user behavior analysis by aggregating data at the platform level, examining differences between Android and iOS users.
c. Investigated correlations of ratings over years.

[6] **Visualization**:

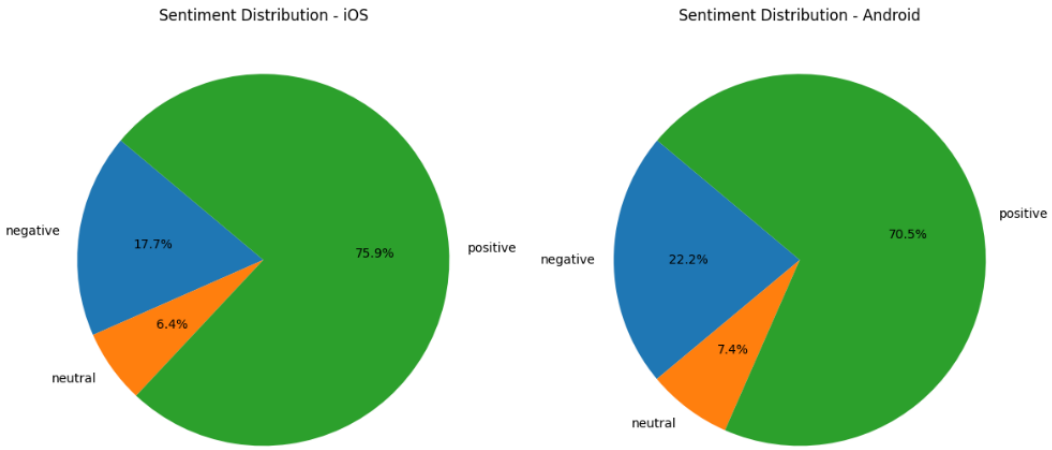


Figure 6- Sentiment distribution per category

a. Created compelling visuals such as word clouds, sentiment distribution graphs, and time-series trends to illustrate findings effectively.
b. Used tools like Matplotlib, Pandas, and WordCloud for comprehensive graphical representation.

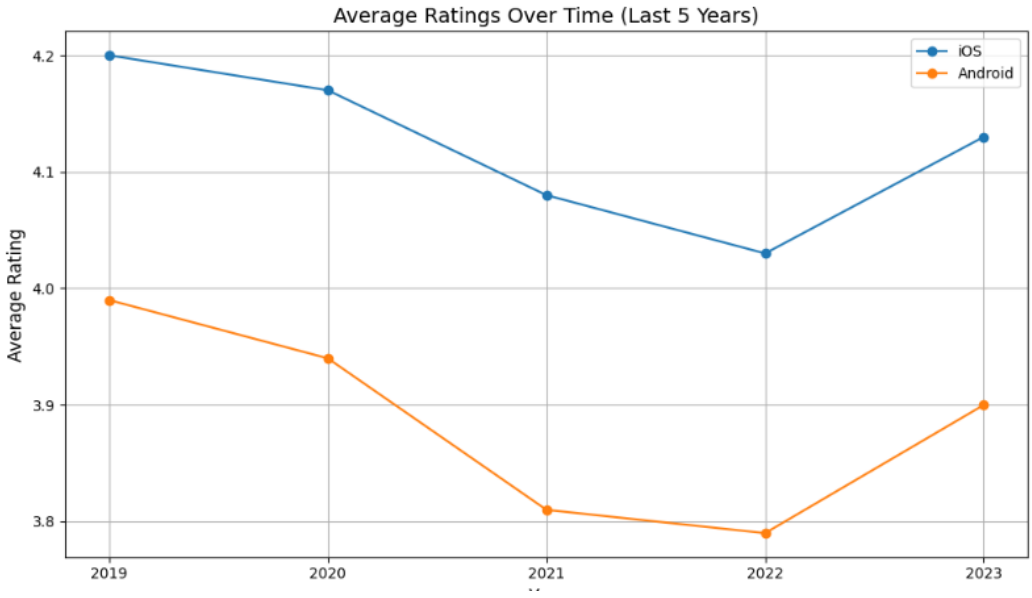


Figure 7- Average Ratings Over Time

Results Representation and Visualization

Figure 8- Most Common words in iOS reviews

iOS Users: The most prominent words, such as "phone," "case," "screen," and "iPhone," highlight a strong focus on protective accessories. Keywords like "Apple," "great," and "fit" suggest a preoccupation with aesthetics and brand loyalty. The emphasis on "perfect" indicates high expectations for quality.

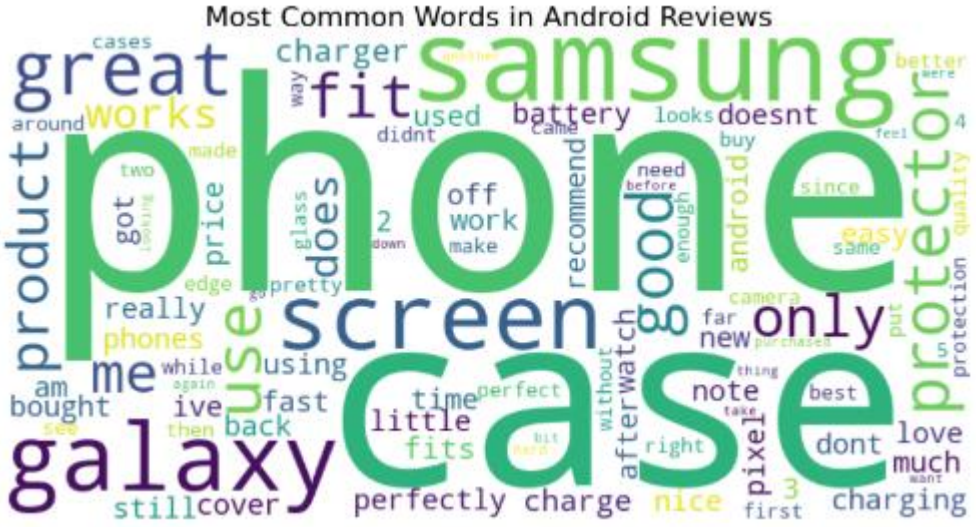


Figure 9- Most Common words in in Android reviews

Android Users: The dominant words include "phone," "case," "Samsung," and "Galaxy," revealing a similar interest in protection but also in specific

brands. Words like "battery," "work," and "fit" underscore practical concerns about usability and durability, while "great" and "recommend" show a mix of satisfaction and practicality.

These word clouds reflect the differing priorities of Android and iOS users, with the former focusing on functionality and the latter on brand identity and precision.

Challenges

During the project, several challenges were encountered:

1. **Data Volume**: Managing a dataset of approximately 9GB in size presented significant memory and processing challenges. Using distributed tools like Hadoop and PySpark mitigated some issues but required optimization.
2. **Data Quality**: The raw data contained inconsistencies, such as missing values, and irrelevant fields. Significant effort went into cleaning and transforming the dataset.
3. **Feature Engineering**: Identifying relevant keywords for categorizing Android and iOS users required extensive text preprocessing and accounting for typos and linguistic variability.
4. **Visualization**: Generating meaningful and visually appealing insights demanded careful selection of visualization techniques, particularly for word clouds and trend graphs.
5. **System Crashes**: Frequent crashes during intensive data processing required resource reallocation and optimization of temporary storage settings in Spark. Analyzing text with context to classify sentiments using Vader and TextBlob were not possible for lack of resources. We experimented a lot but couldn't make them work with our limited resources.

Personal Reflections

Sam El Saati:

This project has been a transformative experience, providing deep insights into the world of big data analytics. Working with tools like Hadoop and PySpark helped me develop technical skills essential for managing large datasets efficiently. One of my key contributions was in designing the feature engineering pipeline, where I focused on deriving rating-based sentiment labels and platform categories. Despite the challenges of handling system crashes and optimizing temporary storage, I learned the value of perseverance and resource management. The experience also enhanced my ability to visualize data effectively, bridging technical findings with impactful communication. I now feel more prepared to tackle real-world data challenges confidently.

Mohamad Alloosh:

This project helped me improve my data analysis and problem-solving skills. I had a great chance to learn using Hadoop and Spark with two of the best teachers. I also focused on preparing the data by cleaning it and making sure it was ready to use. Fixing the messy data was a tricky but satisfying challenge. I also helped a lot with making visualizations, like word clouds and sentiment graphs, to show the data in a simple way. These tasks made me better at using technical tools and working as a team to solve problems step by step. Now, I feel more confident working with big datasets and sharing results clearly in the future.

Security Measures

To ensure data integrity and security, the following measures were implemented:

- Enabled HDFS audit logs to track data access and modifications, as described in the project report [6].

  if not defined HADOOP_SECURITY_LOGGER (

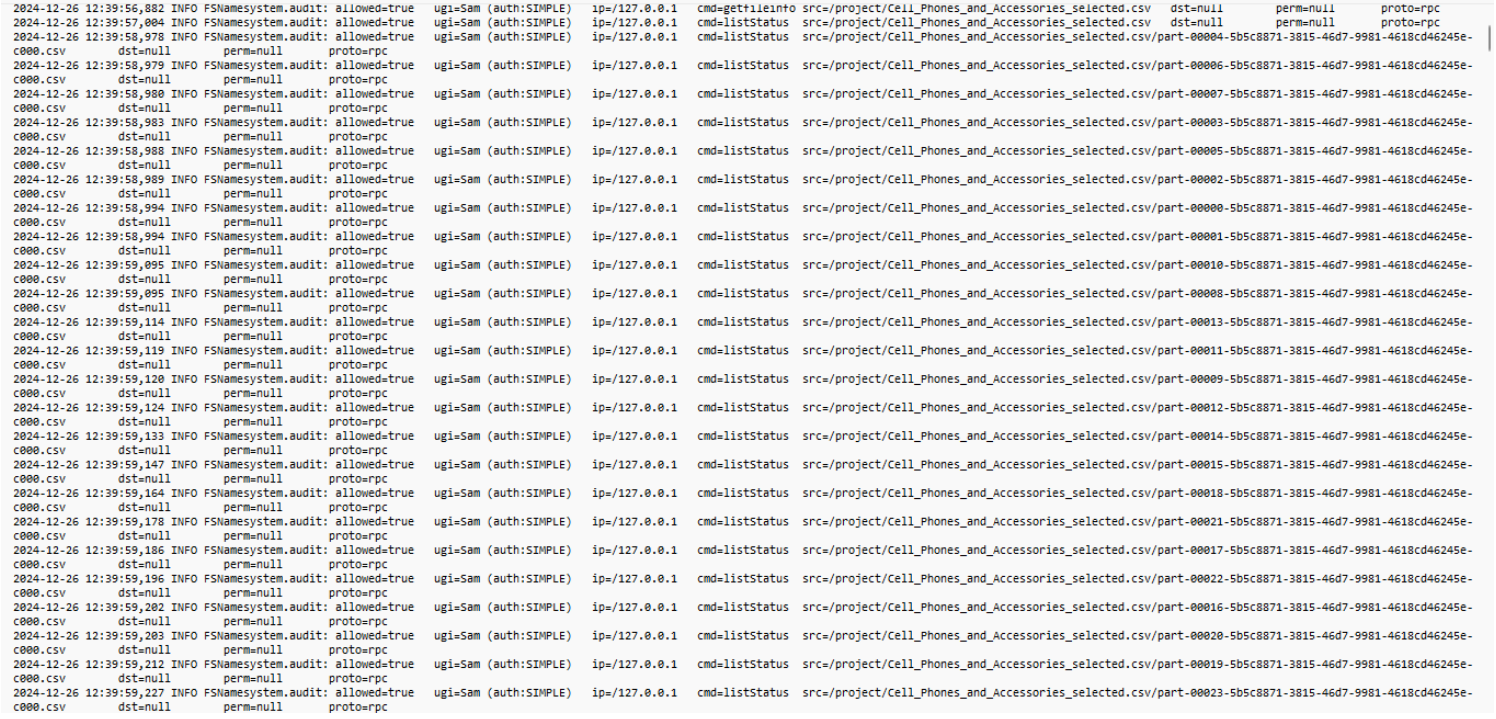
  set HADOOP_SECURITY_LOGGER=INFO,RFAS

)

  if not defined HDFS_AUDIT_LOGGER (

  set HDFS_AUDIT_LOGGER=INFO,RFAAUDIT

)



Conclusion

This project has demonstrated the power of big data analytics in uncovering insights from large and complex datasets. By focusing on the "Cell Phones and Accessories" subset of the Amazon Reviews dataset, we successfully identified user behaviors and preferences across Android and iOS platforms. The analysis revealed distinct patterns, such as Android users prioritizing functionality and battery life, while iOS users emphasized aesthetics and brand alignment. These findings not only highlight user priorities but also offer actionable insights for businesses aiming to improve product offerings and customer engagement.

The integration of tools like Hadoop and PySpark enabled us to manage and analyze a 9GB dataset efficiently, overcoming challenges related to data volume and quality. Visualizations, including word clouds and sentiment distributions, provided clear and impactful representations of the findings. Despite encountering system limitations and resource constraints, the project delivered meaningful results, showcasing the potential of data-driven decision-making in real-world applications.

This experience underscores the importance of robust preprocessing, thoughtful feature engineering, and effective visualization in deriving insights from big data. It has been a valuable learning journey, equipping us with technical and analytical skills essential for tackling similar challenges in the future.

References

[1] The Apache Software Foundation. (n.d.). Apache Hadoop: Setting up a Single Node Cluster. Retrieved from https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html

[2] Pandas Development Team. (2023). Pandas User Guide. Retrieved from https://pandas.pydata.org/docs/user_guide/index.html

[3] Matplotlib Developers. (2023). Pyplot Tutorial. Retrieved from https://matplotlib.org/stable/tutorials/introductory/pyplot.html

[4] Apache Spark. (n.d.). Quick Start Guide. Retrieved from https://spark.apache.org/docs/latest/quick-start.html

[5] Hugging Face. (n.d.). Amazon Reviews Dataset. Retrieved from https://huggingface.co/datasets/McAuley-Lab/Amazon-Reviews-2023

[6] Kiasar Mian & Lasse Poulsen. (2021). Big Data Analytics Report. Retrieved from big_data_project_report_Kia_Lasse-2