UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Big Data Management

Project Phase 1: Landing Zone

DigiScan360

Authors
Narmina Mahmudova
narmina.mahmudova@estudiantat.upc.edu
MD Kamrul Islam
md.kamrul.islam@estudiantat.upc.edu

Supervisors
Alberto Abelló
Sergi Nadal

April 7, 2024

Table of Contents

1 Introduction **2**

2 Data Sources **2**

2.1 E-commerce Data 2

2.2 Social Media Data 2

2.3 Expert Reviews 2

3 Methodology **2**

3.1 Data Collection 2

3.2 Expert Review Collection 3

3.3 Twitter Synthetic Data 3

3.4 Facebook Synthetic Data 3

3.5 Landing Zone 3

3.5.1 Choosing Azure Blob Storage 4

3.5.2 Choice of Microsoft Fabric and Data Factory 4

3.5.3 Lakehouse Deployment 4

3.5.4 Temporal Landing Zone Deployment 4

3.5.5 Persistent Loading 5

3.6 Persistent Landing 6

4 Future Work **6**

References **7**

1 Introduction

The rise of digital technology and online platforms has increased the amount and variety of data that can be used to stay competitive. However, the traditional methods of market analysis, characterized by manual data collection and lengthy analysis periods, are increasingly becoming inadequate. These methods are not only time-consuming but also prone to inaccuracies, making it difficult for companies to make informed strategic decisions in a timely manner. In response to these challenges, DigiScan360, a state-of-the-art platform designed to revolutionize the way headphone companies in Spain conduct competitive analysis and strategic planning. Leveraging advanced data processing technologies, DigiScan360 offers a comprehensive solution for navigating the complexities of the digital ecosystem. By integrating diverse data sources, including social media, expert review websites, and e-commerce platforms, the platform ensures a holistic view of the market.

2 Data Sources

In the process of data collection for our analysis, we employed a methodology that involves both the extraction of existing data and the synthetic generation of data sets. We have collected data from five sources, which are classified into three primary categories: e-commerce, social media engagement data, and expert product reviews. Each category provides a holistic understanding of market dynamics, consumer behavior, and the positioning of products within the competitive landscape.

2.1 E-commerce Data

Data Sources: Amazon and MediaMarkt
DigiScan360 extracted detailed product information, including product specs, reviews, ratings, and even pricing info, from a couple of the big e-commerce sites out there. This strategic aggregation of e-commerce data is instrumental in facilitating a nuanced understanding of sales dynamics, consumer sentiment, and competitive market positioning.

2.2 Social Media Data

Data Sources: Facebook and Twitter
In response to the challenges posed by limited access to Facebook and Twitter's paid APIs, we have innovated by creating synthetic data. This method involves mimicking real-world data that would typically be obtained through these APIs, thereby capturing key metrics such as likes, followers, and shares. The use of social media data is crucial for evaluating brand engagement, gathering consumer feedback, and spotting trends in the market. This approach allows for a comprehensive analysis of social media impact, even in the absence of direct access to full API data streams, ensuring that essential insights into consumer behavior and market dynamics are not overlooked.

2.3 Expert Reviews

Data Source: CNET
CNET is recognized as a leading authority in providing reviews for technology products.(*CNET* 2024) DigiScan360 enhances its analysis by incorporating expert reviews from CNET, specifically focusing on headphones. This addition of professional evaluations to consumer reviews adds a valuable layer of depth to the analysis. The insights gained from these expert reviews are vital for benchmarking products, increasing credibility, and obtaining insights into innovation. This strategy ensures a well-rounded understanding of product performance and market standing, informed by both consumer experiences and professional assessments.

3 Methodology

3.1 Data Collection

For the purpose of enriching our analysis with expert insights, DigiScan360 incorporates a comprehensive data collection process focusing on social media data (Facebook and Twitter) and expert reviews from CNET. Given the restrictions and costs associated with accessing social media APIs, we opted for synthetic generation of data that mirrors the kind of data these APIs would typically provide. This section details the methodology behind collecting expert reviews from CNET.

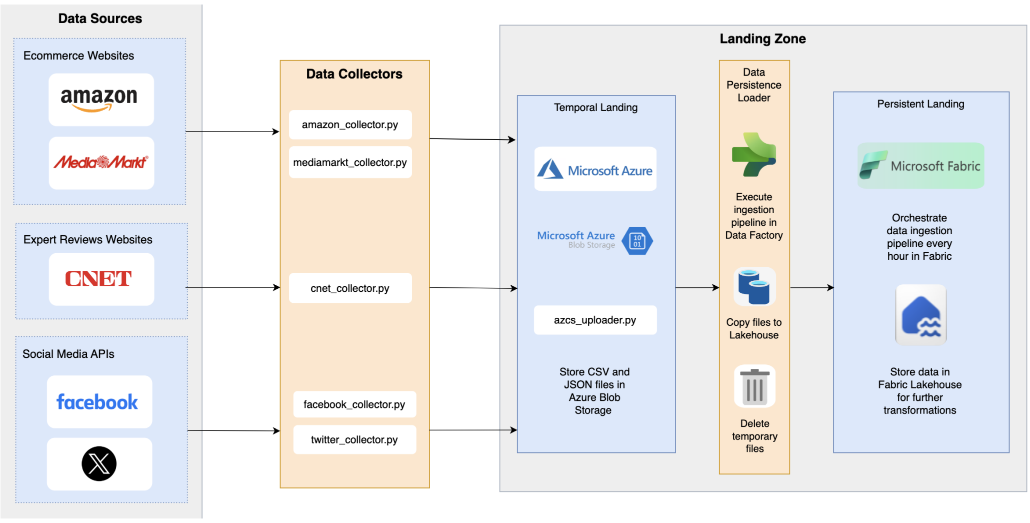


Figure 1: Data management backbone (Phase 1) of DigiScan360

3.2 Expert Review Collection

To collect data from CNET, we developed and utilized a Python script named cnet_collectors.py. This script was specifically engineered to navigate CNET's website and systematically extract information relevant to our study. It begins by initiating a logging system to monitor the process and record any potential issues encountered during its operation. The script then proceeds to access selected CNET web pages, where it identifies and retrieves product listings. Key pieces of information such as product names, descriptions, expert reviews, pros, cons, and professional ratings are extracted. Notably, the get_rating function within the script is designed to pinpoint and extract numerical ratings given by CNET experts, providing crucial insights into each product's market reception and quality. The collected data is carefully organized and saved in a structured format into the expert_reviews.json file for further analysis. Throughout this process, the script includes pauses between operations to ensure ethical scraping practices are adhered to, thus avoiding server overload and complying with web usage policies.

3.3 Twitter Synthetic Data

Our synthetic data generation process was guided by a comprehensive study of the Twitter API v2 documentation, which informed the structure and content of our dataset. By understanding the typical data returned by the Twitter API, we could simulate realistic and relevant data fields (Twitter 2021). To augment the real-world applicability of our dataset, we intentionally introduced variability and ”noise” into the data. This noise took the form of random misspellings, inclusion of off-topic hashtags, and the simulation of varied engagement metrics (e.g., likes, shares). Such deliberate perturbations are designed to mirror the unpredictable nature of actual Twitter data, thereby presenting realistic challenges for data pre-processing and analysis.

3.4 Facebook Synthetic Data

We thoroughly analyzed the Facebook Graph API documentation to structure our dataset authentically (Meta for Developers 2024). By understanding typical data patterns, we replicated pertinent attributes. To make our dataset more practical, we intentionally introduced variability, including null and inappropriate values, mirroring genuine Facebook data's unpredictability. These intentional disturbances create realistic challenges for data preprocessing and analysis.

3.5 Landing Zone

Regarding our modeing approach and technology, we chose to use the following tools in differnt parts of our data management flow:

1. **Temporal Landing:** Microsoft Azure Bolb Storage
2. **Persistent Loading:** Data Factory
3. **Persistent Landing:** Microsoft Fabric Lakehouse

3.5.1 Choosing Azure Blob Storage

We selected Microsoft Azure Blob Storage for its reliability, scalability, and ease of integration with Microsoft Fabric. Azure Blob Storage is ideal for holding a large mix of data temporarily, thanks to its ability to store unstructured data efficiently and cost-effectively. This choice aligns with our project's needs for diverse data handling and integrates smoothly with Microsoft Fabric, facilitating seamless data transfer to subsequent stages.

3.5.2 Choice of Microsoft Fabric and Data Factory

In the next stages of our project, we selected Microsoft Fabric for its extensive ecosystem and integrated services, including data lake, data engineering, and data integration, all conveniently accessible in a single platform. Microsoft Fabric stands out by offering a seamless experience across data warehousing, engineering (featuring Lakehouses and Notebooks), integration (through pipelines and dataflows), real-time analytics, and Power BI, all unified under OneLake without the need for additional setup. This choice was motivated by Microsoft Fabric's holistic approach, contrasting with Hadoop by providing a more complete solution with lakehouse capabilities ensuring ACID compliance via Delta lake, financial advantages, and the provision of a free trial, making it a cost-effective alternative. While Hadoop is a proven choice, the allure of Microsoft Fabric's modern features and its promise of simplification in complex data ingestion and transformation workflows prompted our decision.

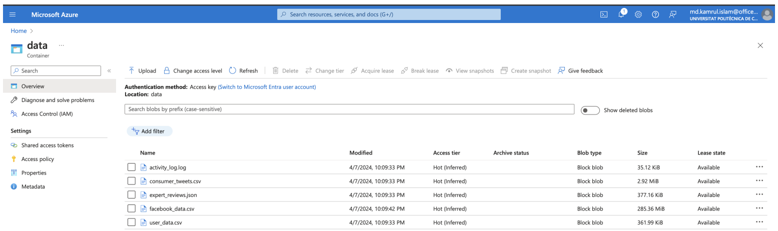
For data persistence and loading, we opted for Data Factory, a fully managed, serverless data integration service known for its user-friendly interface and wide range of connectivity options (Microsoft 2024). This choice is aimed at streamlining the creation, scheduling, and orchestration of data workflows, enabling efficient data movement and transformation from our temporary storage solutions to the lakehouse, thus aligning with our goal of optimized data pipeline management and orchestration.

3.5.3 Lakehouse Deployment

We adopted a lakehouse architecture for our persistent data storage, leveraging Delta Lake to enable ACID transactions and schema enforcement on read (Microsoft 2023). This approach combines the benefits of data lakes and data warehouses, supported by Azure Data Lake Storage, which offers scalability, disaster recovery, and flexibility in data format storage. The lakehouse model was chosen for its ability to maintain high data integrity and support complex analytical queries.

3.5.4 Temporal Landing Zone Deployment

The deployment of a temporary storage area is crucial for managing the diverse data collected in our project. This section explains our choice of Microsoft Azure Blob Storage and the method used to automate data uploads. Following the collection and local storage of data into pertinent files, we aimed to transfer these files to a temporary cloud storage platform to benefit from improved accessibility and scalability. For this task, we deployed a script named `azcs_uploader.py`, designed specifically to upload our compiled data to Microsoft Azure Blob Storage, our chosen temporary storage facility. This script initiates by pinpointing the files designated for upload, including JSONs, CSVs, and log files, as listed in `resources.txt`. Utilizing the Azure Storage Blob client library, the script then establishes a connection with Azure Blob Storage and proceeds to upload each identified file into a pre-set container dubbed ”data” within the Blob Storage framework. A key feature of this script is its ability to overwrite existing files of the same name, while also implementing a versioning system to ensure the storage repository remains both current and orderly. This temporary staging area serves as a holding zone for the files, where they remain until the orchestrator schedules them for processing, thus streamlining the data workflow.



Authentication method: Access key (Switch to Microsoft Entra user account)
Location: data

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
activity_log.log	4/7/2024, 10:09:33 PM	Hot (Inferred)		Block blob	35.12 KiB	Available
consumer_tweets.csv	4/7/2024, 10:09:33 PM	Hot (Inferred)		Block blob	2.92 MiB	Available
expert_reviews.json	4/7/2024, 10:09:33 PM	Hot (Inferred)		Block blob	377.16 KiB	Available
facebook_data.csv	4/7/2024, 10:09:42 PM	Hot (Inferred)		Block blob	285.36 MiB	Available
user_data.csv	4/7/2024, 10:09:33 PM	Hot (Inferred)		Block blob	361.99 KiB	Available

Figure 2: Social Media and CNET data stored in Azure Bolb Storage

3.5.5 Persistent Loading

Persistent loading is a critical phase in our data management strategy, enabling the transition of data from temporary to permanent storage, ensuring its availability for analysis and processing. The transition of data from the temporal landing to the persistent landing is facilitated through a meticulously designed data ingestion pipeline, utilizing the Data Factory tool provided by Microsoft Fabric. This pipeline plays a crucial role in integrating newly acquired data with the pre-existing datasets within our data lake, employing a sequence of ”Copy Data” and ”Delete Data” activities. These operations efficiently transfer data from Azure Blob Storage to the Fabric Lakehouse, subsequently removing it from the temporary storage to maintain data integrity and organization. The pipeline architecture is engineered to process three distinct data streams—expert reviews, Twitter, and Facebook data—in parallel. This process initiates with the transfer of data from
`consumer_tweets`, followed by the deletion of this file from Azure Blob Storage. Similar operations are conducted for `user_data` and Facebook data, ensuring a consistent and streamlined data flow. Additionally, activity logs and expert reviews are managed concurrently, emphasizing the pipeline's capacity for efficient data handling.

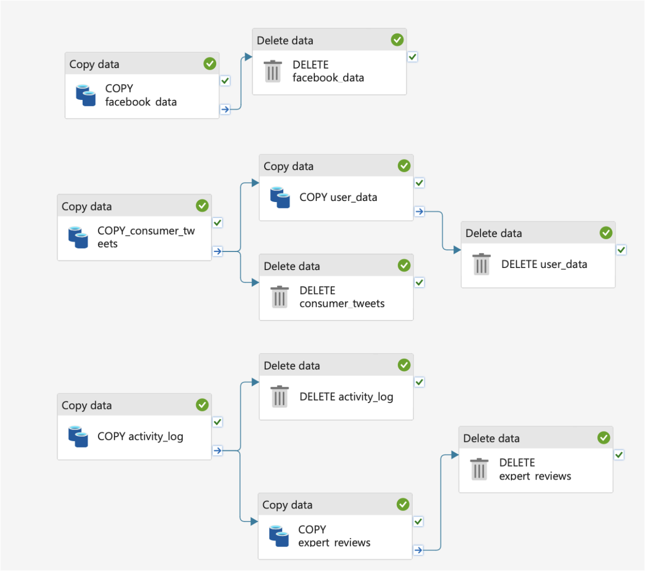


Figure 3: Process diagram of data ingestion pipeline

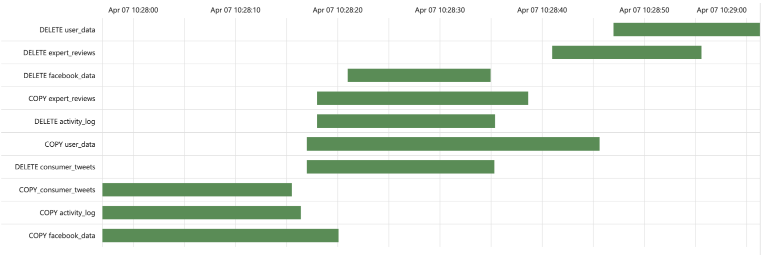


Figure 4: Gantt Chart of data ingestion pipeline

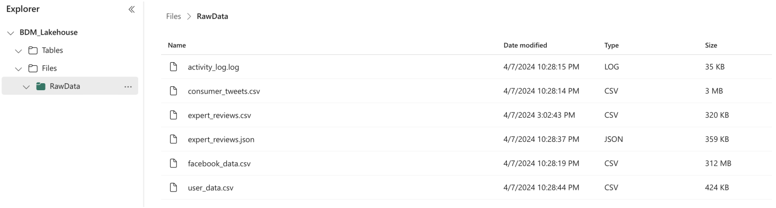
Validation of this pipeline within Fabric confirmed its effectiveness, as indicated by green tick marks denoting the successful completion of each activity. A Gantt chart, detailing the time allocation for each task, revealed insights into the operational dynamics of the pipeline:

- The `user_data` loading took the most time and deleting `facebook_data` took the least amount of time.
- The entire process is optimized to conclude within approximately 1 minutes, showcasing the efficiency of our data persistence strategy.

This approach not only ensures the seamless integration of diverse data sources into our lakehouse but also establishes a robust foundation for subsequent data analysis and insight generation.

3.6 Persistent Landing

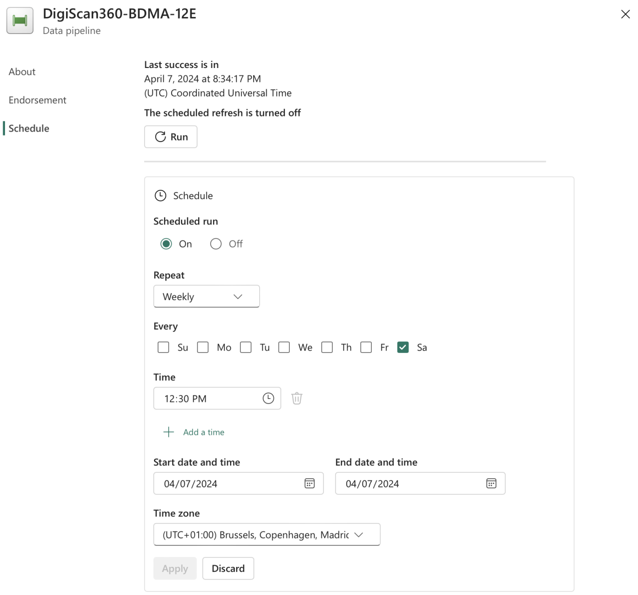
For our persistent landing, we chose to use the lakehouse architecture with Delta Lake as the unified table format, since it provides an analytical store that combines the file storage flexibility of a data lake with the SQL-based query capabilities of a data warehouse. After executing the data ingestion pipeline, our data lakehouse stores the following files from Facebook, Twitter and CNET:



Name	Date modified	Type	Size
activity_log.log	4/7/2024 10:28:15 PM	LOG	35 KB
consumer_tweets.csv	4/7/2024 10:28:14 PM	CSV	3 MB
expert_reviews.csv	4/7/2024 3:02:43 PM	CSV	320 KB
expert_reviews.json	4/7/2024 10:28:37 PM	JSON	359 KB
facebook_data.csv	4/7/2024 10:28:19 PM	CSV	312 MB
user_data.csv	4/7/2024 10:28:44 PM	CSV	424 KB

Figure 5: Current view of files in our lakehouse

In later stages, these files will be transformed to tables inside the Lakehouse. Another feature of Fabric is that it allows orchestration of the data pipelines designed in the Data Factory tool. Hence, we have scheduled our pipeline to run after every week (Saturday) and refresh the data lake:



DigiScan360-BDMA-12E
Data pipeline

Last success is in
April 7, 2024 at 8:34:17 PM
(UTC) Coordinated Universal Time
The scheduled refresh is turned off

Schedule

Scheduled run
● On ○ Off

Repeat
Weekly

Every
□ Su □ Mo □ Tu □ We □ Th □ Fr ☑ Sa

Time
12:30 PM

Start date and time
04/07/2024

End date and time
04/07/2024

Time zone
(UTC+01:00) Brussels, Copenhagen, Madric

Figure 6: Schedule of data pipeline orchestration

4 Future Work

Upon the successful ingestion of data into our lakehouse architecture, our project will advance to a subsequent phase within our data management strategy. This next phase is dedicated to the homogenization of data according to a canonical data model within the formatted zone. Following the standardization process, the data will move into the exploitation zone. In this phase, data undergoes further refinement and structuring, preparing it for detailed analysis and consumption. The primary aim here is to support the development of detailed dashboards and analytical tools that are closely aligned with the strategic goals of DigiScan360. By equipping manufacturers with detailed, actionable insights, we intend to empower them to innovate more effectively, devise strategic plans, and secure a competitive advantage within the industry.

References

CNET (2024). Accessed: 2024-04-07. URL: https://www.cnet.com.

Meta for Developers (2024). *Graph API*. URL: https://developers.facebook.com/docs/graph-api/reference/page/ (visited on 04/07/2024).

Microsoft (2023). *Introduction to Data Engineering - Building a Lakehouse Architecture*. URL: https://learn.microsoft.com/en-us/fabric/data-engineering/tutorial-lakehouse-introduction (visited on 04/07/2024).

— (2024). *Azure Data Factory*. Accessed: 2024-04-07. URL: https://azure.microsoft.com/en-us/products/data-factory.

Twitter (2021). *Twitter API v2 Data Dictionary*. URL: https://developer.twitter.com/en/docs/twitter-api/data-dictionary/introduction (visited on 04/07/2024).