DECODING BIG DATA: SPARK INSIGHTS FOR MUSIC ANALYSIS AND SMART CITY TRAFFIC

Francisco Batista 20221847 | Joel Mendes 20221825 | Lourenço Martins 20222053 | Tomé Santos 20221948 | Vicente Miranda 2022184

Introduction

In a world where **data is the new gold**, the ability to distill insights from massive datasets is a game-changer. This report harnesses the power of Apache Spark on Databricks to showcase how big data technologies can unlock transformative value through two compelling use cases: **Spotify Music Analysis** and **Traffic Prediction for Smart Cities**.

Imagine diving into the intricate universe of music streaming data to uncover the pulse of global listening trends. In the **Spotify Music Analysis**, we process and analyze extensive dataset to identify different collaboration between artists, popularity of songs distributed over time and group musics based on similar features. Spark's unparalleled ability to handle large-scale data enables us to transform thousands of records into insights that highlight how data can drive decisions in the music industry.

On the other hand, envision a city where traffic flows with seamless precision, guided by predictive intelligence. In our **Traffic Prediction for Smart Cities**, we deploy machine learning models on Spark to anticipate congestion and optimize traffic flow. This application of Spark's distributed computing and ML capabilities shows how data can be harnessed to create smarter, more efficient urban environments, reducing gridlock and improving the quality of life.

Through these case studies, we demonstrate how Spark excels at managing the 4 V's of Big Data — Volume, Velocity, Variety, and Veracity. This report is more than an exploration of data: it's a blueprint for how big data, processed with Spark, can power innovation, efficiency, and smarter decision-making.

Objectives

Traffic prediction for Smart Cities

In an era where urbanization is accelerating, traffic congestion remains one of the most pressing challenges for modern cities. Efficiently managing traffic flow is crucial for reducing delays, improving air quality, and enhancing the overall quality of life for citizens. The objective of the Traffic Prediction for Smart Cities analysis is to harness the power of Apache Spark to process large-scale traffic data and build predictive models that forecast congestion patterns.

This project aims to:

- Ingest and Analyze Traffic Data:

Utilize Spark's distributed data processing capabilities to efficiently handle large datasets containing traffic flow, road conditions, and temporal patterns.

- **Predict Traffic Congestion:**

Develop machine learning models to predict traffic bottlenecks based on historical data and real-time information. These predictions help anticipate and mitigate congestion before it occurs.

- **Enable Smarter Decision-Making:**

Provide city planners and transportation authorities with actionable insights to optimize traffic light timings, plan road expansions, and manage peak-hour traffic dynamically.

- **Scalability and Real-World Application:**

Ensure that the solution is scalable, capable of handling data streams from multiple sensors across an entire city, and adaptable to the complexities of urban traffic systems.

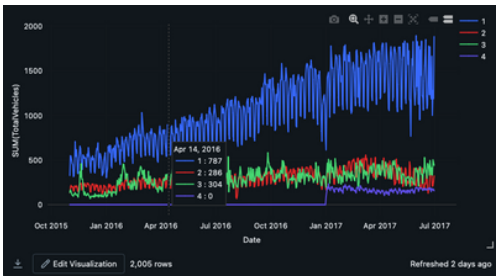


Image 1: Time-Series per Junction

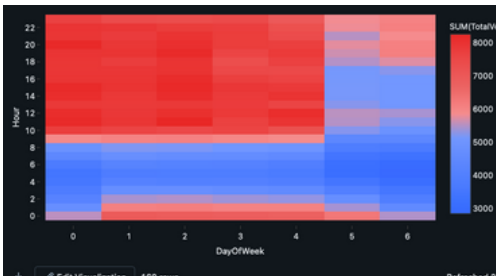


Image 2: Days vs Hour Traffic

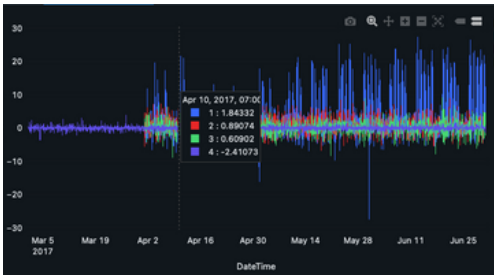


Image 3: Residuals

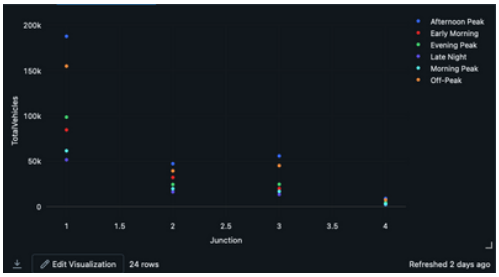


Image 4: Peak Hours Analysis

Music Analysis and Playlist Creation

Music streaming platforms like Spotify have transformed the way we listen to and discover music. With millions of tracks available, Spotify can leverage data to massively improve user experience.
We will begin by analysing the popularity of songs throughout the years, the collaborations between artists and group songs based on similar characteristics in order to create playlist.
Through data-driven insights, this project reveals the power of Spotify's data in shaping trends and enhancing music discovery.

This project aims to:

- Analyze Music Data
- Analyse Collaboration between Artists
- Create Playlist based on similarities between songs

Utilize Spark's distributed data processing capabilities to efficiently handle the scalability of this large datasets containing continuous addition of new music.

- **General Analysis:**

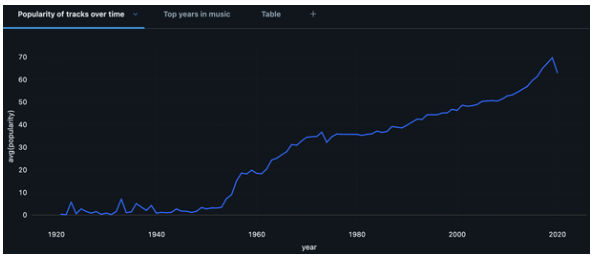
We did some overall analysis of the dataset features such as the average popularity of songs throught the years in order to see the most popular years. This has to be taken in context of course, because Spotify only exists since 2006 so songs after that are more likely to be popular in the given platform.

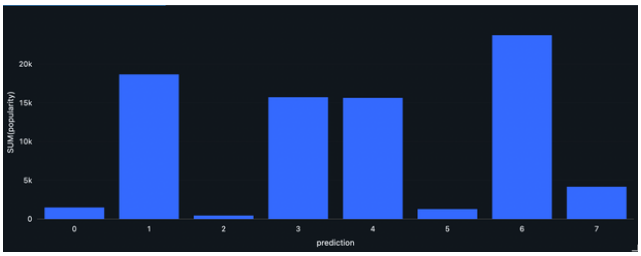
- **Playlist Creation:**

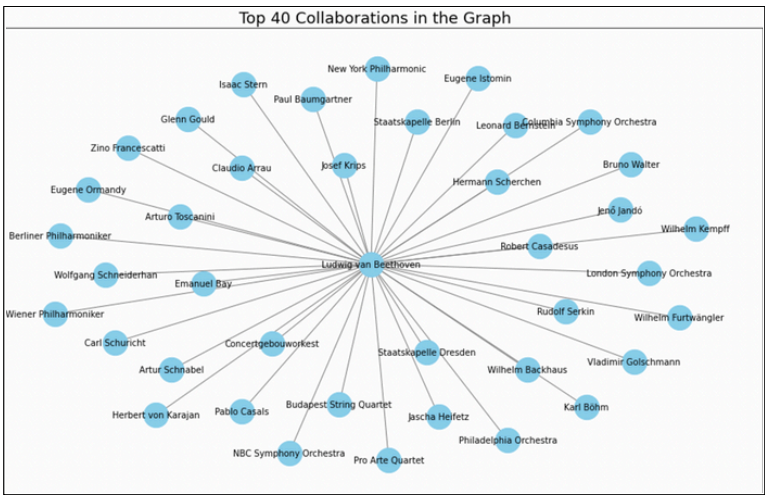
To create personalised playlists based on music data, we employed clustering techniques to group songs with similar characteristics. We started by preprocessing the relevant features from Spotify's music dataset, then used a K-means clustering algorithm to organize the songs into distinct clusters. This method ensures that the playlists are not only based on popularity but also on intrinsic musical qualities, offering a more personalized listening experience.

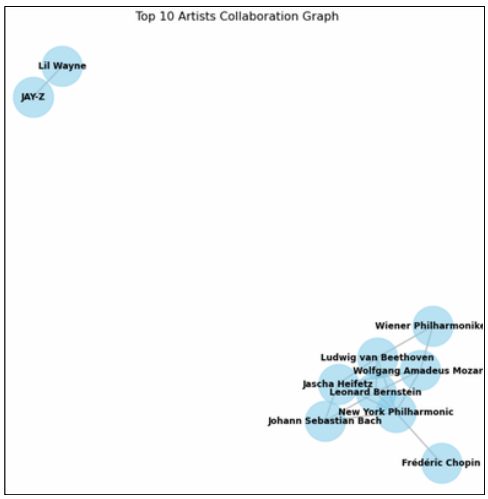
- **Collaboration Analysis:**

We observed the artists who had more collaborations and interpret them visually. Delving deeper into one specific artist (Beethoven) plotting his top40 collaborations. Also looking into the most common collaborations between artists we decided to plot the top10 noticing a clear division between 2 different musical eras.









Methodology

Our approach leverages Apache Spark on Databricks to ensure scalability and efficiency, making it big data-proof. Key steps include:

- **Data Acquisition and Preprocessing**

Traffic data was ingested and stored in Parquet format for optimized retrieval and processing. Using RDD transformations and SparkSQL, the data was cleaned, partitioned, and cached, ensuring seamless parallel processing for large datasets.

Spotify data was ingested as a Spark Dataframe with a predefined StructType.

- **Feature Engineering and Modeling**

Time-based features and lag features were engineered to capture traffic patterns. Predictive models were developed using Spark MLlib Pipelines, optimized for distributed execution to handle vast data volumes effectively.

For clustering music analysis we created a new feature that identifies musics with lyrics in order to better segment our suggested playlists and also to reduce the data size.

- **Real-Time Streaming**

Spark Streaming was implemented to ingest and analyze live traffic data, providing real-time predictions for immediate decision-making and congestion management.

This methodology ensures that every step — from ingestion to modeling — is scalable, efficient, and ready for big data environments.

- **Node Base Analysis**

We incorporated GraphFrames to analyze relationships within the Spotify dataset. By representing artists as nodes and their collaboratios as edges, we constructed a graph model of the data. This allowed us to apply graph-based algorithms, such as PageRank, to identify influential entities based on their connections.

Results and Value Proposition

Our traffic prediction model shows high accuracy across multiple junctions, demonstrating its effectiveness for smart city traffic management:

Why It Works:

1. **Scalable:** Built with Apache Spark, handling large datasets efficiently.
2. **Real-Time:** Spark Streaming provides live traffic predictions.
3. **Optimized Storage:** Parquet format ensures fast data access.

Value Proposition:

1. **Reduced Congestion:** Dynamic traffic optimization.
2. **Cost Efficiency:** Lower fuel and infrastructure costs.
3. **Enhanced Mobility:** Improved urban traffic flow.

Performance Metrics:

Junction 1: RMSE: 0.0688 | R^2: 0.90
Junction 2: RMSE: 0.0951 | R^2: 0.73
Junction 3: RMSE: 0.0625 | R^2: 0.92
Junction 4: RMSE: 0.0504 | R^2: 0.94

RSME: Lower the Better
R^2: Higher the better

Conclusion

This project showcases the transformative potential of Apache Spark to drive innovation through big data. We didn't just analyze data — we unlocked insights that redefine how we understand music and urban mobility.

In the Spotify Music Analysis, we explored and identified trends and patterns in song features. By leveraging graph analysis with GraphFrames, we uncovered connections between artists and through Clustering Analysis we created playlists based on the characteristics of songs. These insights provide a analysis on artist collaboration and musical trends over the years.

For Traffic Prediction in Smart Cities, our model delivers real-time, scalable predictions that empower cities to anticipate congestion and optimize traffic flow dynamically. Using Spark Streaming, MLlib, and Parquet storage, our solution is designed to handle massive datasets seamlessly, offering a blueprint for **efficient urban management**.

Together, these analyses highlight **Spark's ability to transform complex**, high-volume data into actionable insights. This isn't just about handling data — it's about harnessing it to **create smarter systems, better decisions, and a more connected, efficient future.** Big data powers progress, and Spark is the engine driving it forward.

Citations

Kapturov, A. (n.d.). Spotify Data from PySpark Course [Data set]. Kaggle. https://www.kaggle.com/datasets/kapturovalexander/spotify-data-from-pyspark-course

Soriano, F. (2021). Traffic Prediction Dataset [Data set]. Kaggle. https://www.kaggle.com/datasets/fedesoriano/traffic-prediction-dataset

"The use of real-time Big Data frameworks like Apache Spark enables cities to predict and manage traffic flow dynamically, significantly reducing congestion and improving urban mobility." – Zhang et al., 2020

"Graph-based Big Data analysis provides a powerful tool to visualize and explore collaborative patterns within the music industry, revealing insights that traditional analytics cannot." – Torres & Silva, 2021