YouTube Analytics: Optimizing Video Performance with PySpark

Team -Sai Swetha Bodlapati, Akshitha Katyaini Reddy Akkireddy, **Advisor** - Dr. Isam Alobaidi

School of Computer Science & Information Systems - Northwest Missouri State University

INTRODUCTION

This YouTube Analytics project provides content creators with data-driven insights to optimize their video publishing strategies and maximize audience engagement. Using PySpark for large-scale data processing, the analysis extracts actionable patterns from YouTube trending video data, covering optimal posting schedules, content categorization, metadata optimization, and engagement metrics. The system processes historical YouTube trending data to identify critical success factors like ideal publishing times, high-performing content categories, optimal tag usage, and title characteristics that correlate with higher viewership and engagement. By translating complex data patterns into clear recommendations, the project empowers content creators to make informed decisions that can significantly improve their YouTube performance metrics.

METHODOLOGY

1. **Data Collection & Loading:** The dataset, containing metadata of trending YouTube videos, is a 138MB CSV file loaded into a Spark DataFrame for large-scale processing. Schema inference is applied to ensure correct data types.
2. **Data Cleaning & Preprocessing:**
   a. **Timestamp Conversion:** publishedAt and trending_date is converted to Timestamp Type for date-based analysis.
   b. **Handling Null Values:** Missing values in view_count, likes, dislikes, and comment_count are replaced with 0.
   c. **Feature Engineering:**
   - days_to_trend is calculated as the difference between trending_date and publishedAt.
   - publish_hour and publish_day are extracted to determine optimal posting times.
   - An engagement score is computed based on likes and comment_count, normalized by view_count.
   - categoryId is mapped to meaningful video category names.
3. **Exploratory Data Analysis (EDA):**
   a. **Best Posting Times:** Identifies the most effective days and hours for video publishing**.**
   b. **Content Performance:** Analyzes popular categories, title lengths, and tag usage for engagement.
   c. **Regional Analysis:** Evaluates country-wise video performance by category and engagement metrics.
   d. **Engagement Patterns:** Studies like-to-view and comment-to-view ratios, along with engagement trends based on metadata.
4. **Data Visualization:**
   Using Matplotlib, bar charts and graphs illustrate:
   - Best days and hours to post
   - Top-performing categories and channels
   - Engagements across different tag counts and title lengths

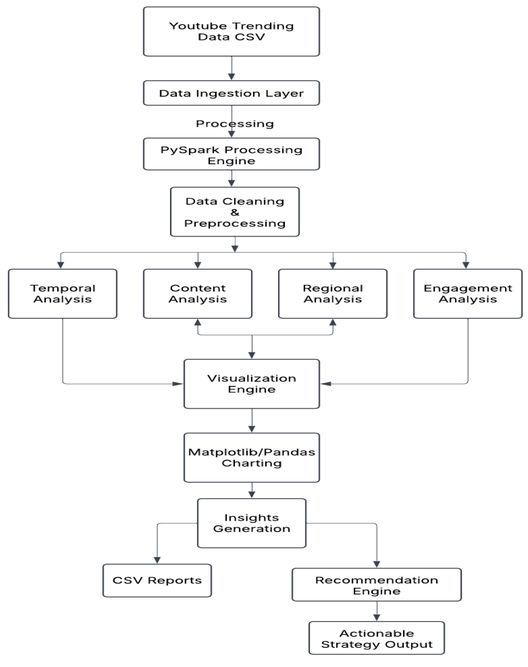
TECHNOLOGY

The code utilizes the following technologies:

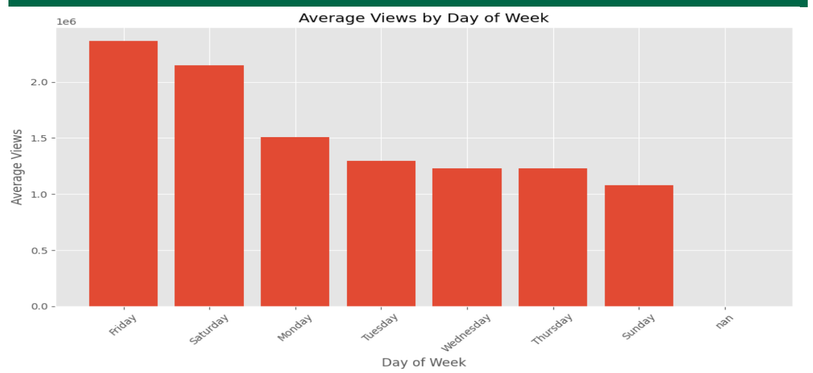
1. Apache Spark (PySpark) – Used for big data processing, data cleaning, transformations, and aggregations.
2. Pandas – Converts Spark DataFrames to Pandas DataFrames for easier analysis and visualization.
3. Matplotlib – Creates visualizations such as bar charts to analyze trends.
4. SQL Functions (PySpark SQL) – Used for data manipulation, filtering, and aggregations.
5. Spark DataFrames API – Performs transformations and actions on large datasets.
6. UDF (User-Defined Functions) – Used to apply custom logic within Spark.
7. Spark SQL Window Functions – Enables advanced aggregations such as ranking, partitioning, and sliding window calculations.
8. File Handling (CSV Processing) – Reads, writes, and processes CSV files.

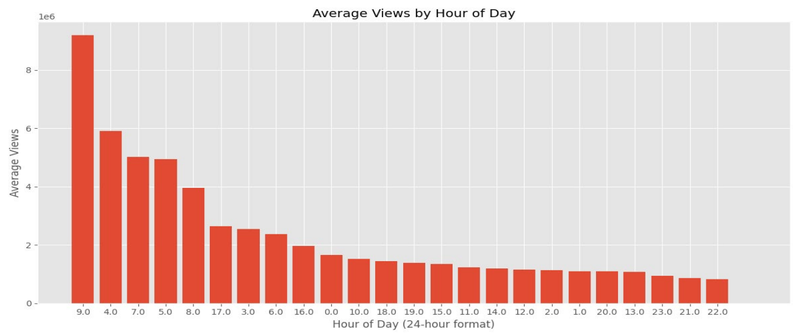
ARCHITECTURE

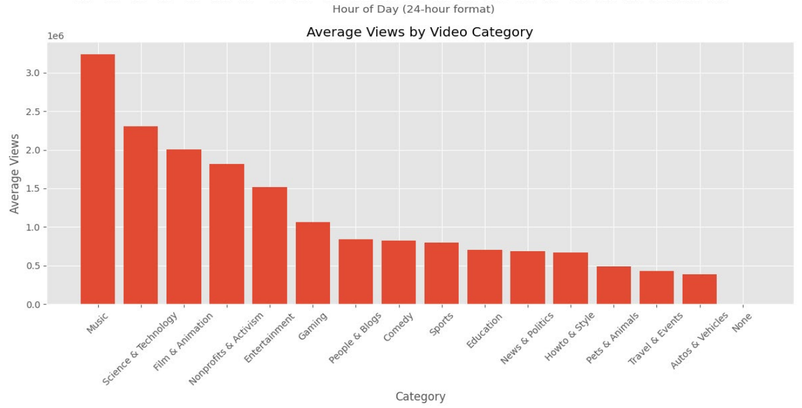
The flowchart represents a **YouTube Trending Video Analysis Pipeline** using **PySpark** for data processing, cleaning, and feature engineering. It covers **temporal, content, regional, and engagement analysis**, with insights visualized using **Matplotlib and Pandas**. The final output includes **CSV reports and strategic recommendations** to help content creators optimize their posting schedules and engagement strategies.



TREND ANALYSIS







CONCLUSION

Effective YouTube strategies involve posting at **optimal times**, focusing on **popular categories**, and tailoring content to **regional preferences**. Using **strategic tags**, well-crafted **titles**, and balanced **content** enhances engagement. Given the vast scale of YouTube data, **big data tools like PySpark** are essential for identifying **complex patterns** and extracting **actionable insights**. By leveraging **data-driven strategies**, creators can optimize their content, align with **viewer behavior**, and maximize **reach, visibility, and engagement**.