*Wiki Trend Analytics using Cloud*

Rajendra Kumar   
kumar13@iu.eduData Architect  
  
Mukund Komati  
mkomati@iu.edu  
Data Engineer  
Ajay Reddy Gajulapally  
agajulap@iu.edu  
BI Engineer

*Abstract*— This project focuses on building a scalable ETL pipeline and interactive dashboards to analyze large-scale Wikimedia datasets. Our solution integrates datasets such as pageviews, media usage, and editorial activity to uncover trends in user behavior and improve decision-making for Wikipedia stakeholders.

Keywords— Wikimedia Analytics, ETL Pipeline, Data Visualization, Big Data, Cloud Computing

# Introduction

The Wikimedia Foundation hosts vast datasets detailing user interactions, media consumption, and edits across its projects. These datasets, while publicly available, remain underutilized due to their complexity and scale. This project addresses the need for a scalable analytics framework to derive actionable insights, such as pageview trends, media demand, and geographic editing activity, aiding in better content strategy and user engagement.

# Dataset

The Commons Impact Metrics dataset offers valuable insights into the impact of community contributions to Wikimedia Commons. It focuses primarily on media files uploaded by and categories belonging to GLAM actors (galleries, libraries, archives, and museums). This dataset aggregates monthly data on GLAM-related categories and media files, using pageviews as the primary metric. The data is available through the data lake, API, and public dumps, making it accessible for various research purposes1.

For the Commons Impact Metrics, useful columns include category name, subcategory count, file count, and pageviews for category metrics. For media file metrics, important columns are file name, uploader, upload date, usage count, and pageviews. This dataset is particularly valuable for understanding the reach and impact of cultural heritage content on Wikimedia platforms1.

The Unique Devices dataset estimates the number of unique devices accessing Wikimedia projects. It provides data at both daily and monthly granularity, covering all Wikimedia projects. This dataset uses a privacy-preserving methodology to count devices rather than individual users. Key columns in this dataset include project name, country (for country-specific data), date, and unique devices count. This information is crucial for understanding readership trends across different Wikimedia projects and geographical regions.

The Mediacounts dataset offers information on how often media files from upload.wikimedia.org are transferred to users. It provides daily data on both original and transcoded file transfers, including referer information. Important columns in this dataset include the base name of the media file, total response size, original file transfers, transcoded file transfers (for audio, image, and movie), and referer types (internal, external, unknown). This dataset is particularly useful for analyzing the popularity and usage patterns of multimedia content across Wikimedia projects.

Lastly, the Geoeditors dataset contains information about the geographic distribution of Wikipedia editors. It provides monthly data covering all Wikipedia language versions, offering editor counts by country. Key columns in this dataset include wiki database name, country code, year-month, editor activity level (categorized as 5-99 edits or 100+ edits), and editor count2. This dataset is invaluable for understanding the global distribution of Wikipedia contributors and identifying trends in editor engagement across different regions.

These four datasets, when used together, provide a comprehensive view of Wikimedia projects' usage, contributions, and impact. They offer insights into content popularity, user engagement, geographic distribution of both readers and contributors, and the impact of cultural heritage contributions. Researchers and community members can leverage these datasets to gain a deeper understanding of Wikimedia projects' global reach and influence

# Technical Approach

Our project follows a structured technical workflow to process large-scale Wikimedia datasets and deliver insightful visualizations.

Our technical approach starts with gathering data from sources such as Commons Impact Metrics, Geo Editors, Media Counts, and Unique Devices. Using a web scraper built with Beautiful Soup, the data is extracted and stored in HDFS. A Python ETL process is then used to clean, transform, and load the data into a Hive staging database. From there, the data flows into a DataMart layer, enabling seamless access through tools like Hue for query execution and Jupyter Notebook for advanced analysis. Finally, we generate insightful visualizations, including charts for metrics like page views, editor locations, media categories, and device usage trends.

The process involves the following key components:

A diagram of data processing

Description automatically generated

1. **Data Sources**:

The datasets used include:

* + *Common Impact Metrics*: Analyzing pageview counts by category.
  + *Geo Editors*: Tracking editorial activity based on geographic regions.
  + *Media Counts*: Measuring media file access frequency.
  + *Unique Devices*: Understanding device-specific access trends.

1. **Hadoop File System (HDFS)**:

All raw datasets are ingested into the Hadoop File System (HDFS) for distributed storage and scalable processing.

A computer screen shot of white text

Description automatically generated

1. **Main ETL Process**:

An ETL (Extract, Transform, Load) pipeline processes data in HDFS:

* + **Extract**: Pulls raw data from sources into HDFS.
  + **Transform**: Cleans and aggregates the data using PySpark, ensuring consistency and accuracy.
  + **Load**: Loads the transformed data into a staging area in Hive.

1. **Hive Database (Staging)**:

The cleaned and structured data is stored in Hive tables, enabling efficient querying and integration with downstream layers.

1. **DataMart Layer**:

A DataMart layer extracts relevant subsets of data from Hive for focused analysis. This intermediate layer ensures that insights are generated quickly without overwhelming the system.

1. **Visualization Tools**:

Jupyter Notebooks using PyHive are used to create exploratory visualizations and develop charts for specific datasets for now.

# Methodology

1. **Cloud Infrastructure Setup**: Configured a scalable cloud environment using Jetstream2. Set up for Apache Hadoop, Tez and Hive to ensure efficient data storage and processing.

A screenshot of a computer

Description automatically generated

1. **Data Extraction**: Automated data retrieval using Python scripts to fetch raw datasets from Wikimedia Analytics.
2. **Data Transformation**: Preprocessed data using PySpark, handling missing values, correcting anomalies, and aggregating metrics for efficient analysis.

A screenshot of a computer

Description automatically generated

1. **Data Storage**: Stored transformed data in Hive tables for structured and queryable access, enabling SQL-like queries for trend analysis.
2. **Data Visualization**: We will generate more user insights and develop interactive dashboards using Streamlit and also integrate with Hive to allow exploration of trends like pageviews, media usage, and geo-edits.

# Preliminary Results

## Anayysis on Datasets

1. **Monthly Editor Activity by Language**:

We analyzed editor activity for the top 10 non-English languages on Wikimedia. The data revealed that Spanish and French have the highest editor activity, consistently exceeding 8,000 editors per month. Languages like Korean and Hebrew showed significantly lower editor engagement, highlighting potential areas for community growth.

**A graph of different colored lines

Description automatically generated**

1. **Editor Activity by Country**:

The United States leads with over 25,000 active editors monthly, followed by Germany and Japan. Developing nations like India also exhibit steady growth in editor engagement, indicating expanding contributions from diverse regions.

**A graph of different colored lines

Description automatically generated**

1. **Mobile-to-Desktop Usage by Project Family**:

Wikimedia projects like Wikifunctions and Wikivoyage exhibit a higher mobile-to-desktop ratio, indicating their popularity on mobile devices. In contrast, projects like Wikinews and Wikimedia Foundation see more desktop-based usage.

**A graph of a graph of a company

Description automatically generated with medium confidence**

1. **Proportion of Mobiles vs. Desktops**:

Mobile devices account for 43.1% of total traffic, while desktops dominate at 56.9%. This emphasizes the need for optimized experiences across both platforms.

**A pie chart with text overlay

Description automatically generated**

1. **Total Devices by Project Family**:

Wiktionary leads with the highest number of devices accessing the platform, followed by Wikisource and Wikibooks. These trends reflect user preference for specific knowledge categories. This does not include Wikipedia, since it has 17.5 B Unique Devices in its project family and to get better analysis overall, exclusion of this project family is considered.

**A graph with orange bars

Description automatically generated**

1. **Monthly Total Devices Across All Project Families**:

Device usage across all Wikimedia projects saw consistent growth throughout the year, peaking at over 2.2 billion devices in October 2024.

**A graph with blue lines

Description automatically generated**

1. **Key Lessons from Tools and Analysis**:

The integration of Python, Pandas, and visualization librariesprovided detailed insights. Hive facilitated efficient querying for large datasets. PySpark's distributed processing enabled scalable data transformation.

# Road Map

* + **Week 5 (Nov 18 – Nov 22):** Analyze Media Counts and Commons Impact Metrics datasets.
  + **Week 6 (Nov 23 – Nov 30)**: Develop visualizations and connect Hive to tools.
  + **Week 7 (Dec 1 – Dec 5)**: Test and finalize the ETL pipeline, dashboards, and report.

# Obstacles and Backup Plans

**Data Volume**:

Obstacle: Handling large-scale datasets, which include monthly and daily updates, posed challenges in performing aggregated analysis over the entire year due to memory and processing limitations. The daily data size is more than 2 GB. So, if we need to analyze yearly data, then we need to have higher memory space to handle such a large volume of data. We limited our dataset to only 4 key metrics. These four tables are in itself requiring more space. So, we are using sampling to progress with our project.

Backup Plan: To mitigate this, we implemented data sampling techniques, focusing on representative subsets of the data. Additionally, we designed the ETL pipeline to aggregate data during the transformation stage, minimizing the need for full dataset analysis. If further scalability issues arise, we plan to leverage additional Hadoop nodes to distribute the processing load.

**Ports Issue**:

Obstacle: Configuring ports required for Hive connectivity was a challenge due to security constraints on the Jetstream2 cloud environment. This initially blocked us from directly querying Hive.

Backup Plan: As a temporary solution, we integrated PyHive with Jupyter Notebooks to access and analyze Hive data programmatically. We are also exploring secure SSH tunneling and alternative access configurations to overcome these port restrictions permanently.

**System Resource Utilization**

Obstacle: Running resource-intensive operations using MapReduce on Jetstream2 caused delays and system crashes.

Backup Plan: Switched to Apache Tez for faster and more efficient processing, reducing intermediate overhead. Scheduled heavy tasks during low-usage periods and prioritized critical jobs using Hadoop’s resource manager.

**Visualizations**:

Obstacle: Creating interactive visualizations with large data required optimizing query performance and handling frequent access to specific datasets, which could lead to latency.

Backup Plan: We will implement caching mechanisms for frequently accessed datasets, reducing the load on Hive queries. Additionally, we can use key metrics in the DataMart layer to improve response times. For more dynamic needs, we plan to incorporate incremental updates to reduce full reloads of data into visualization tools.

# Expected Deliverables

Build a scalable pipeline to process Wikimedia datasets using Hadoop, PySpark, and Hive. Extract metrics like Unique Devices, assessing edits by country, Total Transfers of the images per category and page view counts. Create interactive visualizations using available tools. Provide a detailed project report and user guide. Showcase the dashboards, and key findings from the datasets.

##### References

Below are some of the wikimedia dataset links and wikipedia pages from which we got the information about datasets and initiated us to perform analysis.

1. [Wikimedia Downloads: Analytics](https://dumps.wikimedia.org/other/analytics/)

[2] [Analytics: Mediacounts](https://dumps.wikimedia.org/other/mediacounts/readme.html)

[3] [Data Platform/Data Lake/Traffic/Mediacounts - Wikitech](https://wikitech.wikimedia.org/wiki/Data_Platform/Data_Lake/Traffic/Mediacounts)

[4] [Analytics: Unique Devices](https://dumps.wikimedia.org/other/unique_devices/readme.html)

[5] [Research:Unique devices - Meta](https://meta.wikimedia.org/wiki/Research:Unique_devices)

[6] [Analytics: Commons Impact Metrics](https://dumps.wikimedia.org/other/commons_impact_metrics/readme.html)

[7] [Commons Impact Metrics/Data Model - Wikitech](https://wikitech.wikimedia.org/wiki/Commons_Impact_Metrics/Data_Model)

[8] [Commons Impact Metrics - Wikitech](https://wikitech.wikimedia.org/wiki/Commons_Impact_Metrics)

[9] [Data Platform/Data Lake/Edits/Geoeditors - Wikitech](https://wikitech.wikimedia.org/wiki/Data_Platform/Data_Lake/Edits/Geoeditors)

[10] [Data Platform/Data Lake/Edits/Geoeditors/Public - Wikitech](https://wikitech.wikimedia.org/wiki/Data_Platform/Data_Lake/Edits/Geoeditors/Public)