# Yelp Review Data Analytics Using Big Data Technologies

## Basic Details

**Title:** Yelp Review Data Analytics Using Big Data Technologies

**Duration:** 2 Months

**Team Size:** 7

**Team Members**

* Karan Anpat
* Mayuri Kakde
* Yogeshwar Patil
* Tushar Nandurkar
* Santosh Satti Reddy
* Sanjana Dalvi
* Mrunmai Jawalekar

## Problem Statement

Analyze the Yelp dataset to identify trends in customer reviews, business performance, and user engagement across locations and categories, aiming to provide actionable insights through data exploration only.

## Project Objectives

1. Analyze the distribution of star ratings across businesses and categories.
2. Track review volume and rating trends over time.
3. Identify top-performing businesses based on ratings and number of reviews.
4. Compare customer engagement across different cities and business types.
5. Explore patterns in review behavior, such as review length and timing.

## Dataset Description

The Yelp dataset contains a wealth of information, including:

* Business Information:
* Details such as business names, categories (e.g., restaurants, gyms, salons), location (city, state, latitude, longitude), operating hours, price range, and other attributes like Wi‑Fi availability or parking options.
* User Data:
* User profiles with information such as user ID, number of reviews written, average rating given, number of friends, and useful/funny/cool votes received — helping to identify influential customers.
* Reviews and Ratings:
* Millions of user-generated reviews including star ratings (1–5), written feedback, timestamps, and sentiment-rich content that reflect customer satisfaction and experience.
* Check-ins:
* Time-stamped records of when users check in at businesses, offering insights into peak hours and customer visit patterns.
* Tips:
* Short user-generated suggestions or comments that highlight specific aspects of businesses, often more concise than full reviews.

## Tools & Technologies Used

* **Amazon S3 (Simple Storage Service):** Used for storing raw and processed Yelp datasets in a scalable, durable cloud environment.
* **Amazon EMR (Elastic MapReduce):** Managed cluster platform used to process large-scale data using distributed computing frameworks like PySpark.
* **PySpark (on EMR):** Used for distributed data processing, transformation, and building machine learning pipelines on large-scale data.
* **Jupyter Notebook:** For running PySpark interactively, visualizing results, and building exploratory analytics and ML models.
* **AWS Glue:** For serverless data integration, ETL (Extract, Transform, Load), and data cataloging to prepare data for analytics.
* **Amazon Athena:** An interactive query service used to analyze data directly in S3 using standard SQL via ODBC/JDBC connections.
* **ODBC/JDBC Connections:** To enable integration between Athena and external tools like BI dashboards or Python notebooks.
* **Power BI:** Business intelligence for creating interactive dashboards. Connected to Athena via ODBC/JDBC.
* **Git & GitHub:** Version control and collaboration; codebase, notebooks, scripts, and documentation maintained in a GitHub repository.
* **GitHub Actions (YAML + Python):** CI/CD automation to upload scripts to S3, trigger Glue jobs & crawlers, and orchestrate the pipeline.
* **Terraform / CloudFormation:** Infrastructure as Code to provision S3, Glue, IAM, and related resources.

## Methodology

A structured big data analytics pipeline was implemented to transform raw Yelp datasets into a unified master dataset and generate insights through EDA and visualization.

Step 1: Data Conversion (JSON to CSV)

* Original Yelp data was provided in JSON format for five datasets — user, review, business, tip, and check-in. Converted datasets into CSV format using Python to enable efficient loading into Amazon S3 and compatibility with PySpark.

Step 2: Initial EDA on Individual Datasets

* Performed exploratory analysis on all five datasets to understand schema, identify missing values, and evaluate relevance. Selected user, review, and business datasets for further analysis.

Step 3: Feature Selection

* Extracted relevant columns to reduce complexity and focus on meaningful attributes:
* User Dataset: user\_id, name, review\_count, fans
* Review Dataset: review\_id, user\_id, business\_id, text, stars, cool, funny, useful, date
* Business Dataset: business\_id, b\_name, state, city, categories, super\_category

Step 4: Data Preprocessing & Transformation (PySpark on Amazon EMR)

* Removed null values and duplicates. Extracted year and month from review dates. Transformed categories into higher-level super\_category groupings. Joined datasets on user\_id and business\_id to consolidate information.

Step 5: Creation of Master Table

* Combined cleaned datasets into a single master table with final schema:

['business\_id', 'user\_id', 'name', 'cool', 'review\_id', 'funny', 'stars', 'useful', 'city', 'review\_count', 'fans', 'b\_name', 'state', 'categories', 'super\_category', 'year', 'month', 'sentiment\_stars']

Step 6: Final EDA on Master Dataset

* Identified most active and influential users.
* Ranked top-rated businesses by city and category.
* Analyzed review trends over time and extracted keyword patterns and review sentiment.
* Assessed distribution of ratings and engagement metrics (funny, cool, useful).

## Workflow Breakdown – GitHub Actions (deploy\_glue.yml)

* **Checkout Code:** Uses actions/checkout@v3 to check out the repository so that the ETL script and related files are available on the runner.
* **Configure AWS Credentials:** Uses aws-actions/configure-aws-credentials@v4 to set up AWS CLI credentials. Reads the AWS Access Key ID, Secret Access Key, and Session Token from GitHub Secrets.
* **Upload ETL Scripts to S3:** Synchronizes the local scripts/ directory to an S3 bucket using aws s3 sync. Copies updated ETL code to S3 for the Glue job.
* **Start AWS Glue Job:** Triggers the Glue job (yelpdata) and captures the JOB\_RUN\_ID output.
* **Wait for Glue Job to Finish:** Polls job status. Proceeds on SUCCEEDED; fails on FAILED/STOPPED/TIMEOUT.
* **Start Glue Crawler:** Runs the Glue crawler (yelpcrawlers) to update the Data Catalog.
* **Wait for Crawler to Finish:** Polls until crawler state is READY, indicating cataloging is complete.

## Terraform

This AWS data pipeline ingests public Yelp datasets (business, review, user) from S3, processes them in AWS Glue, and writes cleaned results back to S3. The infrastructure is provisioned with Terraform and consists of:

* S3 Bucket
* AWS Glue Resources (database, job, crawler)
* IAM Role

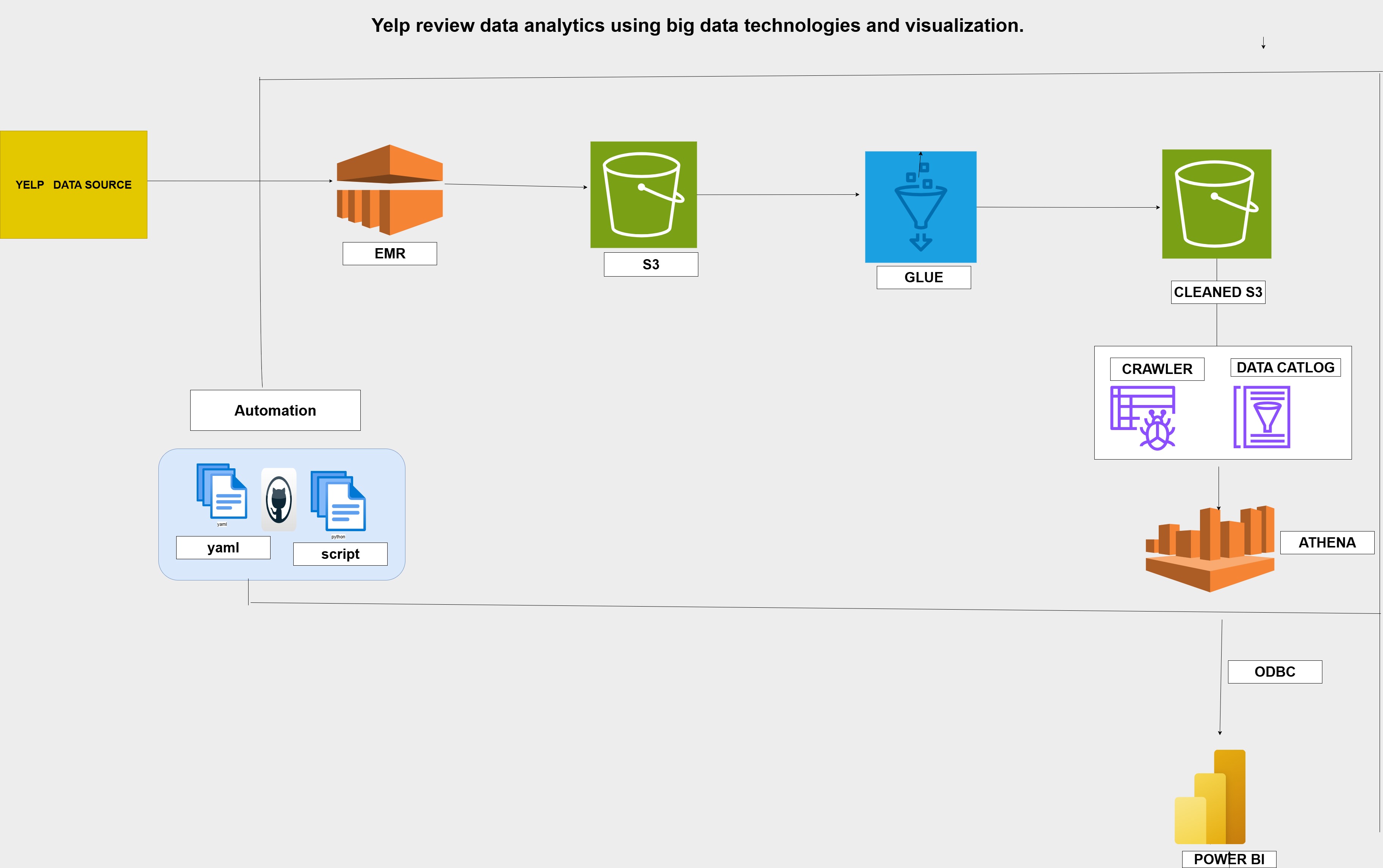
### Terraform Workflow Breakdown

* **Terraform Apply:** Runs terraform apply (via CI/CD) to create S3 bucket, Glue database, Glue job, and Glue crawler.
* **Upload Glue Script:** Uploads the ETL Python script (glue\_job.py) to S3 under /scripts/.
* **Start Glue Job:** Starts the Glue job via AWS CLI or Console; reads the script from S3 and provisions a Spark cluster.
* **Glue Processing:** AWS Glue reads Parquet data, performs joins and transformations (e.g., renaming, category mapping), and writes cleaned CSV output to S3.
* **Glue Crawler:** After the job succeeds, the Glue crawler scans the CSV output folder, infers schema, and creates a table in the Glue Data Catalog for Athena.

## Results & Outcomes

* Processed millions of reviews using distributed PySpark on AWS EMR.
* Automated ETL and infrastructure provisioning reduced manual work by ~80%.
* Delivered Power BI dashboards visualizing trends in ratings, reviews, and engagement across locations and categories.
* Enabled serverless, query-on-S3 analytics through Athena with ODBC/JDBC connectivity.

## Architecture Diagram



**Dashboard**

