**Experiment No. 7**

**Title:** Development of a Dashboard and Reporting Tool Based on Real-Time Social Media Data Using Power BI

**Aim**

To develop an interactive dashboard and reporting tool using **Power BI** to analyze and visualize real-time social media data, helping users gain insights into engagement trends, audience demographics, and content performance.

**Software Requirements**

* **Power BI Desktop** (for developing dashboards and reports)
* **Power BI Service** (for online sharing and collaboration)
* **Data Connectors** (for real-time social media data extraction, e.g., APIs for Twitter, Facebook, Instagram, YouTube)
* **Microsoft Excel** (optional, for data preprocessing)
* **SQL Server** (optional, for storing large datasets)
* **Python/R** (optional, for advanced data processing and visualization)

**Theory**

**3.1 Introduction to Social Media Analytics**

Social media analytics involves collecting, analysing, and interpreting data from platforms like Twitter, Facebook, and Instagram to measure engagement, audience behaviour, and trends. Real-time analytics is crucial for businesses to make data-driven decisions.

**3.2 Role of Power BI in Real-Time Analytics**

**Power BI** is a powerful business intelligence tool that enables users to create dynamic dashboards and reports using real-time data streams. It integrates with multiple data sources, allowing live data updates and interactive visualizations.

**3.3 Key Features of Power BI for Social Media Analysis**

* **Data Connectivity:** Connects with APIs for real-time data retrieval.
* **Data Transformation:** Cleans and processes data using Power Query.
* **Data Modelling:** Establishes relationships between datasets for better insights.
* **Visualizations:** Provides various charts, graphs, and reports.
* **AI-Driven Insights:** Uses machine learning for sentiment analysis.

**3.4 Dataset Description**

The dataset used for this project is a **GitHub dataset of top companies repositories**, containing various attributes related to repository activity and metadata. The key columns in the dataset include:

* **Creation Date:** The date when the repository was created.
* **Description:** A brief summary of the repository's purpose.
* **Fork:** Indicates whether the repository is a fork of another project.
* **Has Downloads:** Boolean value specifying if downloads are enabled.
* **Has Issues:** Boolean value indicating if issue tracking is enabled.
* **Has Projects:** Boolean value indicating if the repository has projects enabled.
* **Issues:** The total number of issues reported in the repository.
* **Languages:** The programming languages used in the repository.
* **Last Update:** The last modification date of the repository.
* **License:** The type of open-source license applied to the repository.
* **Owner:** The organization or individual managing the repository.
* **Repository:** The name of the repository.
* **Stars:** The total number of stars (likes) received by the repository.
* **Tags:** Keywords or labels associated with the repository.
* **URL:** The web link to the repository.
* **Watchers:** The number of users watching the repository for updates.

This dataset helps analyse repository popularity, trends in programming languages, licensing preferences, and engagement metrics. By visualizing this data in Power BI, users can gain insights into the open-source ecosystem and corporate contributions to GitHub.

**Steps to Develop the Dashboard**

**Step 1: Data Collection**

* Use **Power BI Data Connectors** or **Python scripts** to fetch real-time data via social media APIs (e.g., Twitter API, Facebook Graph API).
* Store fetched data in **SQL Server**, **Excel**, or directly in Power BI.

**Step 2: Data Cleaning and Transformation**

* Remove duplicate entries and handle missing values.
* Transform raw text into structured data (e.g., extracting hashtags, mentions).
* Use **Power Query Editor** to preprocess and structure data.

**Step 3: Data Modeling**

* Create relationships between different datasets (e.g., posts, likes, comments, shares).
* Define calculated columns and measures using **DAX (Data Analysis Expressions)**.

**Step 4: Creating Visualizations**

* Build interactive reports using **bar charts, line graphs, pie charts, heatmaps, and word clouds**.
* Use slicers and filters to enable dynamic user interactions.
* Implement sentiment analysis using AI-powered visuals.

**Step 5: Real-Time Data Updates**

* Use **Power BI Streaming Datasets** to update visuals in real-time.
* Set up automatic data refresh intervals in **Power BI Service**.

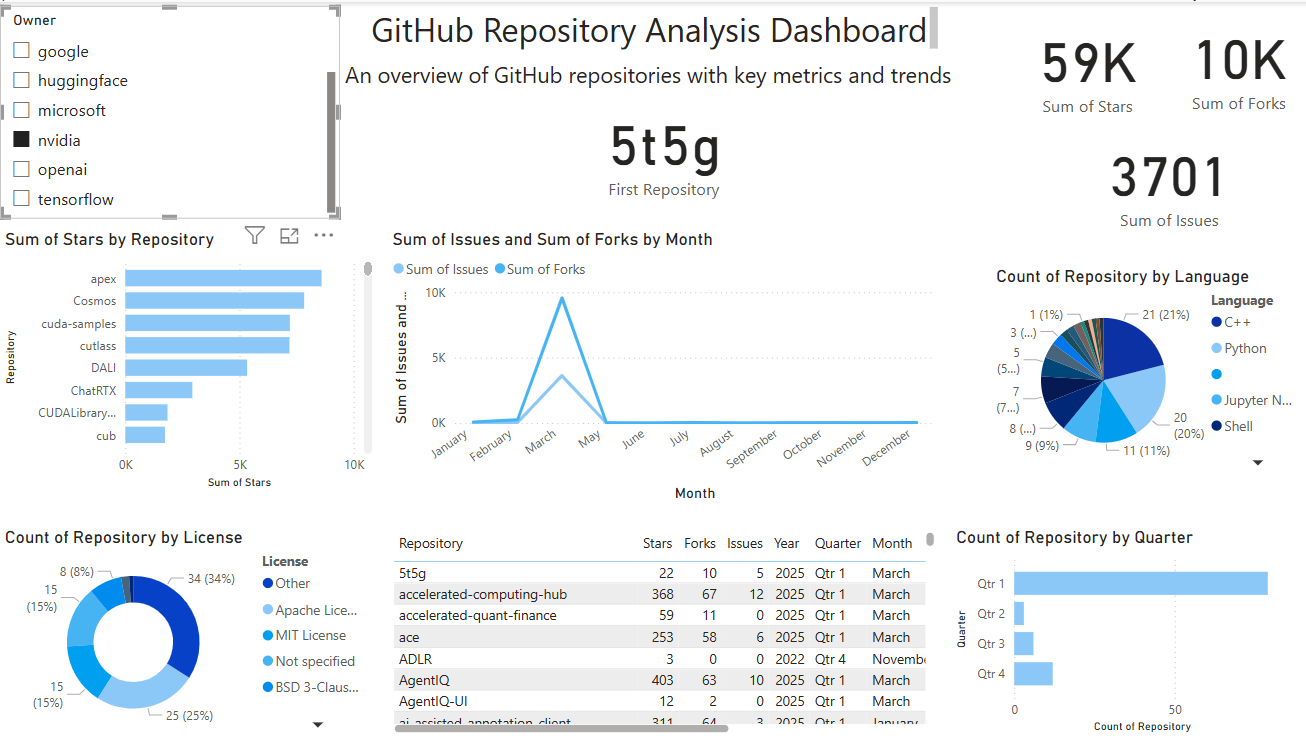
**Step 6: Publishing and Sharing**

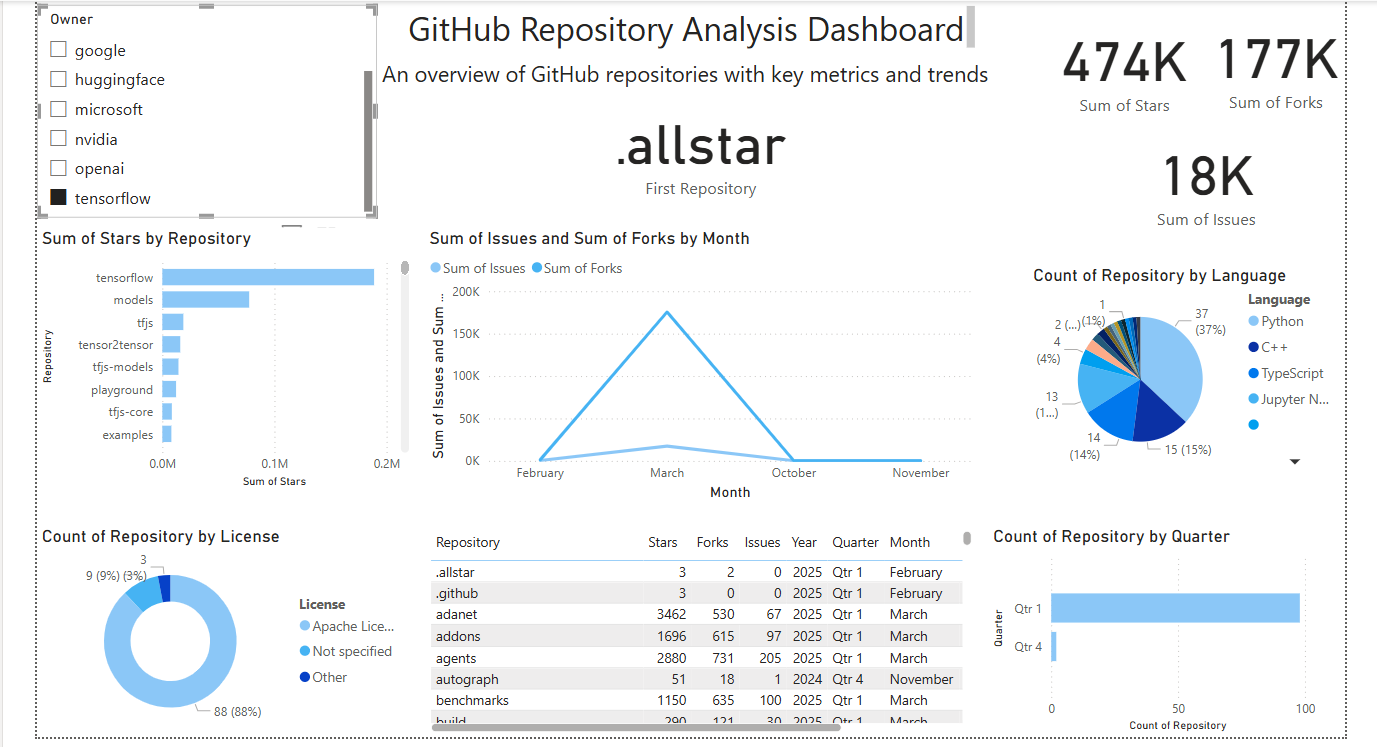
* Publish the dashboard to **Power BI Service** for online access.
* Embed reports in web applications or share them with stakeholders.
* Configure **row-level security (RLS)** to manage user access.

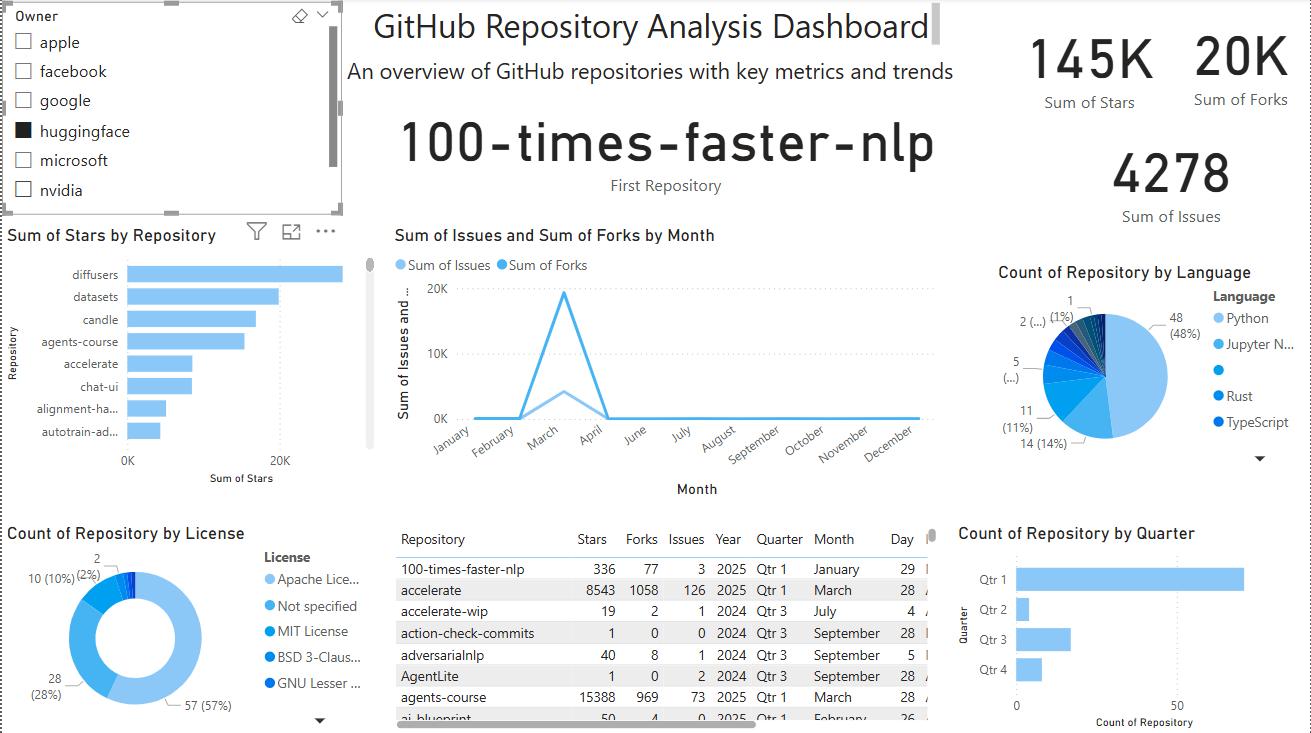
**Output**

**The final output is a fully interactive Power BI dashboard that provides real-time insights into GitHub repository performance. Key dashboard elements include:**

* **Repository Growth Metrics: Number of stars, forks, and watchers over time.**
* **Programming Language Distribution: Breakdown of languages used across repositories.**
* **Top Repositories: Most popular repositories based on stars, forks, and activity.**
* **License Distribution: Analysis of common open-source licenses used.**
* **Recent Updates: Last modification date trends to track active repositories.**







**Conclusion**

The developed Power BI dashboard successfully enables real-time monitoring and analysis of social media data. By leveraging Power BI's data integration, transformation, and visualization capabilities, businesses can make informed decisions, optimize social media strategies, and improve audience engagement. The project demonstrates how real-time analytics enhances digital marketing effectiveness and provides actionable insights for brands and content creators.