MoMo Data Analysis Project

Authors

- **Samuel Ihimbazwe** - s.ihimbazwe@alustudent.com
- **Jesse Walusansa** - j.walusansa@alustudent.com
- **Andrew Ogayo** - a.ogayo@alustudent.com
- **Ntare Allan** - a.ntare@alustudent.com

Project Overview

The MoMo Data Analysis Project is a comprehensive initiative aimed at analyzing Mobile Money (MoMo) data to uncover patterns, trends, and insights. This project serves as a tool for understanding user behavior, transaction patterns, and identifying potential areas for service improvement. By leveraging data analytics and visualization techniques, the project provides actionable insights to stakeholders in the mobile money ecosystem.

Features

Data Analysis

- **Transaction Volume Analysis**: Identify the volume of transactions over specific periods.
- **User Behavior Patterns**: Understand how users interact with the platform.
- **Time-Based Transaction Analysis**: Assess peak and off-peak transaction times.
- **Geographic Distribution of Transactions**: Analyze where transactions occur most frequently.
- **Transaction Value Analysis**: Examine the monetary value of transactions.

Visualization

- **Interactive Dashboards**: Real-time representation of data trends and patterns.
- **Time Series Plots**: Track transaction trends over time.
- **Geographic Heat Maps**: Visualize transaction density in different regions.
- **Transaction Flow Diagrams**: Illustrate the flow of transactions between users.
- **Statistical Distribution Charts**: Display the distribution of transaction amounts and frequencies.

Statistical Analysis

- **Descriptive Statistics**: Summarize and describe transaction data.
- **Correlation Analysis**: Identify relationships between transaction attributes.
- **Trend Analysis**: Discover patterns in transaction data over time.
- **Predictive Modeling**: Forecast transaction trends using machine learning.
- **Anomaly Detection**: Detect irregularities or potential fraudulent activities in transactions.

Technologies Used

Programming Languages

- Python 3.x
- SQL for data querying

Data Analysis Libraries

- pyhon
- flask

Development Tools

- Git for version control
- VS Code for a development environment

Project Contributions

Backend Development

- **Jesse Walusansa** and **Ntare Allan** focused on the backend aspects of the project.
    - Developed APIs for seamless integration of data.
    - Extracted data from SMS and integrated it into the analytics pipeline.
    - Designed database schemas and implemented data storage solutions.
    - Ensured accurate parsing and processing of transaction data.

Frontend Development

- **Samuel Ihimbazwe** and **Andrew Ogayo** led the frontend development.
    - Designed and implemented interactive dashboards.
    - Created layouts and user-friendly interfaces for data visualization.
    - Developed components to display transaction details effectively.
    - Focused on responsiveness and usability across devices.

Documentation

- **Jesse Walusansa** authored the README file and overall project documentation.
    - Created comprehensive guides for installation and usage.
    - Documented project features, structure, and contributions.
    - Maintained clarity and thoroughness for ease of understanding.

Project Workflow

1. **Data Collection and Integration**:
    - SMS data was extracted using custom parsers.
    - APIs were developed to integrate data from multiple sources into the database.
2. **Data Analysis**:
    - Raw data was cleaned and transformed into a usable format.
    - Analytical scripts were executed to derive meaningful insights.
3. **Visualization**:
    - Dashboards were created to represent data visually.
4. **Testing and Validation**:
    - Unit tests ensured the accuracy of APIs and data pipelines.
    - Visualizations were validated against sample data for consistency.
5. **Documentation and Deployment**:
    - A detailed README was created for project onboarding.
    - The project was prepared for deployment on both local and cloud environments.

Project Structure

```
momo-data-analysis/
├── api.py              # Backend API logic
├── app.js              # Frontend application logic
├── charts.js           # Handles chart visualizations
├── data.js             # Manages data for frontend purposes
├── index.html          # Main HTML file for the project
├── modified_sms_v2.xml # Dataset or configuration file
├── momo.py             # Backend or processing script for MoMo analysis
├── styles.css          # CSS file for styling
├── Assets/             # Additional resources like images or icons
```

Challenges Faced

API Integration

- Merging data from multiple APIs required careful handling of discrepancies in data formats and inconsistencies.

Transaction Display

- Ensuring that transactions were properly displayed in the dashboard involved extensive debugging and testing.

Collaboration

- Coordinating efforts between frontend and backend teams required effective communication and version control.

Data Cleaning

- Cleaning and transforming raw data to ensure it was ready for analysis posed significant challenges.

Visualization Consistency

- Achieving consistency across different visualization tools was complex and required iterative refinement.

Usage Instructions

Prerequisites

- Python 3.x
- pip (Python package installer)
- Git

Installation

Clone the repository: git clone https://github.com/yourusername/momo-data-analysis.git
cd momo-data-analysis

1. Create a virtual environment:

   ```
   python -m venv venv
   source venv/bin/activate  # On Windows: venv\Scripts\activate
   ```
2. Install required packages:

   ```
   pip install -r requirements.txt
   ```

Running the Project

1. Navigate to the project directory.
2. Activate the virtual environment.

Run Jupyter notebooks for analysis:

```
jupyter notebook
```

3.

Contributing

1. Fork the repository.
2. Create a new branch (`git checkout -b feature/AmazingFeature`).
3. Commit your changes (`git commit -m 'Add some AmazingFeature'`).
4. Push to the branch (`git push origin feature/AmazingFeature`).
5. Open a Pull Request.

Acknowledgments

- **ALU**: For providing the opportunity to work on this project.
- **Team Members**: For their dedication and collaborative spirit.
- **Open Source Community**: For the tools and libraries that made this project possible.