Report on Financial Data ETL and Analysis Pipeline

Introduction

This report describes a Python-based ETL (Extract, Transform, Load) pipeline designed to process and analyze financial stock data from two sources: the MarketStack API and a Kaggle dataset. The pipeline addresses data inconsistencies, performs feature engineering, and prepares the data for analysis and visualization.

Data Sources

1. **MarketStack API:** Provides real-time and historical stock data via API calls.
2. **Kaggle Dataset:** Offers a broader dataset of world stock prices available as a CSV file through KaggleHub.

Pipeline Stages

1. **Data Collection:**
    - Retrieves stock tickers from a predefined list.
    - Fetches historical data for these tickers from the MarketStack API (or a mock source for testing).
    - Loads the Kaggle dataset using KaggleHub.
2. **Data Transformation:**
    - **Date and Timestamp Normalization:** Standardizes date formats to timezone-aware datetime objects and filters data for a specific year (2025).
    - **Column Name Standardization:** Converts column names to lowercase with underscores for consistency.
    - **Handling Missing Data:** Identifies and addresses missing values, including calculating capital gains.
    - **Data Validation:** Removes rows with negative or invalid financial values.
    - **Feature Engineering:** Calculates daily return and volatility as new features.
    - **Aggregation:** Groups data by ticker and date, calculating aggregate metrics.
    - **Merging:** Combines the processed MarketStack and Kaggle datasets, removing duplicates.
3. **Data Loading:**
    - Loads the final processed data into a MongoDB database for persistent storage and future analysis.

Analysis and Visualization

The pipeline includes functions for generating visualizations such as:

- Daily return distribution
- Stock price vs. volume
- Volatility over time
- Average daily return by stock
- Stock prices over time

Key Features

- **Robust Data Handling:** Addresses common data quality issues like format variations, missing values, and invalid data.
- **Feature Engineering:** Enriches the data with calculated features relevant for financial analysis.

- **Flexibility:** Allows for customization of date ranges, data sources, and analysis parameters.
- **Visualization:** Provides functions to generate insightful visualizations of stock market trends.
- **Data Persistence:** Loads processed data into MongoDB for long-term storage and retrieval.

Potential Use Cases

- **Investment Analysis:** Identify trends, track performance, and assess risk of different stocks.
- **Algorithmic Trading:** Develop and backtest trading strategies using historical data.
- **Market Research:** Understand market dynamics, volatility, and relationships between stocks.
- **Financial Modeling:** Build predictive models for stock prices, returns, and other metrics.

Conclusion

This ETL pipeline offers a comprehensive solution for processing and analyzing financial stock data. By addressing data inconsistencies, adding valuable features, and providing visualization capabilities, it empowers users to gain insights into the stock market and support informed decision-making.