FINM 32500 Computing for Finance in Python

Project Part 3

Infrastructure and Data Models Description Report

Name
Andy Andikko
Harrison Zhang
Matheus Raka Pradnyatama

The **Data Lake** is the foundational component for storing raw and processed data as pickle objects in separate directories. It offers flexibility by supporting multiple data types (e.g., text, JSON-like, and DataFrames) and provides schema-free storage. Key features include:

- **Raw and Processed Data Storage:** Stores unprocessed datasets and stores data that has undergone transformations for quick retrieval and analysis. The DataLake directory is created via a user_defined_parameter or can be accessed from an existing directory. Each data stored must have a unique name and must have corresponding metadata. This means any update, creation, or deletion will affect the data stored and the metadata.
- **Metadata Management:** Each dataset is associated with rich metadata, including data type, column names, author, timestamps, file size, and modification times, stored in metadata.json for consistent access across components. A different metadata handler is used for textual or JSON files. Any updates to existing data or metadata require user confirmation or a force parameter to be set. store_data checks if a dataset already exists in metadata.
- **Secure Access:** Access to data is protected by the *access_decorator*, which validates an *access_key* against the *secured_access* dictionary. Invalid or missing access keys result in access denial.
- **Retrieval**: The *retrieve_data* method uses the processed flag to locate the dataset in either the raw or processed subdirectory. If the file exists, it is deserialized using pickle and returned.

The **Data Catalog** organizes datasets within the Data Lake into logical categories, enabling efficient data discovery and exploration. It enhances the usability of the platform by:

1. **Catalog Organization**:
   - **Categories**: The catalog organizes datasets into categories, which act as thematic groupings (e.g., Equities, Fixed Income, Macro). Each category is a key in the catalog's main dictionary (*self.categories*).
   - **Datasets Within Categories**: Each category contains a list of datasets represented as dictionaries. These dictionaries store dataset details, including metadata and ownership information.
2. **How Metadata supports Data Discovery**:
   - Each dataset is stored with detailed metadata, such as the dataset name, data source, update frequency, and the user responsible for adding the dataset.
   - **Search Functionality**: Users can search for datasets across all categories using the *search_datasets* method, which matches keywords in names or metadata.
   - **Category Listing**: Users can list all datasets in a specific category using the *list_datasets_in_category* method, which helps them navigate large catalogs.

The **Data Workbench** is the transformation hub, preparing datasets for analysis. It interacts with the Data Lake for persistent storage and includes:

- Transformation Management: Allows users to register and apply custom transformations such as VWAP calculation or sentiment scoring.
- In-Memory Storage: Temporarily stores transformed datasets for quick access.
- Processing Logs: Tracks each transformation for reproducibility and debugging.

The **Quant Data Models** provide domain-specific tools for structured data analysis. Two primary models are implemented. The **IntradayDataModel** analyzes intraday financial data, focusing on stock prices and volumes. It requires specific columns (timestamp, price, volume, and symbol) to validate the data. Key features include calculating technical indicators (e.g., SMA, EMA, RSI), VWAP analysis, and volatility transformations. The model can fetch data directly from Yahoo Finance, apply transformations like rolling volatility or indicator calculations, and export processed data to a DataWorkbench for further use. It also can load saved data. The **NewsDataModel** processes financial news and generates insights using sentiment analysis and named entity recognition (NER). It requires columns like timestamp, headline, and relevance for validation. Key functionalities include cleaning text, predicting sentiment using a pre-trained BERT model, and extracting named entities from news articles. It can analyze articles to compute sentiment scores, extract key entities, and identify top headlines based on sentiment. Like the IntradayDataModel, it integrates with DataWorkbench.

Data Catalog <-- Data Lake <--> Data Workbench <--> Quant Model <--> Quant Analyst

System Flow: The Data Lake stores datasets and metadata, which the Data Catalog categorizes for easy discovery. The Data Workbench retrieves data from the lake, applies transformations, and prepares it for analysis. Processed data flows through Quant Models for domain-specific insights, enabling Quant Analysts to perform backtesting, event studies, and strategy optimization.

1. **Quant Analyst**: Initiates workflows by defining tasks such as data analysis, applying transformations, and retrieving insights.
2. **Data Platform**: The central interface connecting analysts to platform functionalities.
3. **Data Workbench**: Handles operations like data transformations, in-memory storage, and chaining operations.
4. **Data Catalog**: Organizes datasets, manages metadata, and allows efficient search and categorization.
5. **Intraday Data Model**: Provides reusable, parameterized transformation functions like moving averages, VWAP, and volatility.
6. **Market Data Processor**: Applies transformations to data and updates metadata during the processing pipeline.
7. **DataLake**: Stores raw and processed data, acting as the platform's backbone.