Assignment 6

Aditya Srivastava (2201013)
Deepanika Gupta (2201062)
Kartik Saini (2201103)

1 Q1: Core Functional Modules in the Business Logic Layer

The Business Logic Layer (BLL) of Analytica contains several core functional modules that mediate between the React/TypeScript presentation layer and the Django data access layer. These modules encapsulate the core business logic required for analyzing tweets and delivering insights to users.

1.1 Tweet Analysis Module

This module handles the processing of tweets through the three different analysis models (emotion, sentiment, and toxicity):

- **Responsibilities:** Processes tweet text through the RoBERTa models, classifies results, and aggregates analysis data for visualization.
- **Interaction with Presentation Layer:** Provides processed analysis results that are displayed in the React frontend's visualization components. The frontend displays charts showing the distribution of emotions, sentiment categories, and toxicity levels.

1.2 Tweet Fetching Module

This module manages the real-time retrieval of tweets from Twitter using Selenium-based scraping:

- **Responsibilities:** Handles Twitter scraping operations based on user input (hashtag or username), manages the number of tweets to fetch, and preprocesses tweets before analysis.

- **Interaction with Presentation Layer:** Receives search parameters from the user interface form, providing feedback on the fetching process through loading indicators and status messages in the UI.

1.3 Leaderboard Module

This module manages the ranking and categorization of tweets for the leaderboard feature:

- **Responsibilities:** Ranks tweets based on various criteria (highest emotion scores, most positive/negative sentiment, toxicity levels), sorts and filters tweets, and determines which tweets appear on the leaderboard.
- **Interaction with Presentation Layer:** Provides sorted and categorized tweet data to the leaderboard UI component, which displays tweets in ranked order with relevant analysis metrics and interactive elements.

1.4 User History Module

This module handles the storage and retrieval of analysis history for logged-in users:

- **Responsibilities:** Records analysis parameters and results for authenticated users, manages history storage, and provides access to past analyses.
- **Interaction with Presentation Layer:** Supplies historical data to the history view in the React frontend, allowing users to browse, filter, and reload past analyses.

1.5 Live Wall Module

This module manages the real-time display of analyzed tweets from the database:

- **Responsibilities:** Retrieves recently analyzed tweets from the database, applies filters or categories if needed, and prepares them for display.
- **Interaction with Presentation Layer:** Provides a continuous stream of analyzed tweets to the Live Wall component in the UI, which displays tweets with their analysis results in an auto-updating feed.

2 Q2: Description of Business Logic Implementation

2.1 A) Implementation of Business Rules

In Analytica, business rules are implemented across different modules to ensure consistent operation and accurate analysis:

2.1.1 Tweet Analysis Rules

- **Classification Thresholds:** Rules define how tweets are classified into emotion categories, sentiment types, and toxicity levels based on confidence scores from the RoBERTa models. For example, a tweet is classified as "toxic" only if its toxicity score exceeds a predefined threshold.
- **Aggregation Rules:** Rules determine how individual tweet analyses are aggregated for visualization. For example, the dominant emotion across a set of tweets is calculated based on the highest average emotion score.

2.1.2 Tweet Fetching Rules

- **Rate Limiting:** Rules ensure the application respects Twitter's scraping limitations, fetching betweeen 1 to 100 tweets at a time.
- **Content Filtering:** Rules determine which tweets are included or excluded from analysis, such as filtering out tweets without meaningful text.
- **Duplicate Handling:** Business rules define how to handle duplicate tweets or similar content, ensuring the analysis is not skewed by repetitive data.

2.1.3 Leaderboard Rules

- **Category Definition:** Rules define what constitutes each leaderboard category, such as the criteria for "most optimistic" or "most toxic" tweets.
- **Refresh Policies:** Business rules govern when and how the leaderboard data is refreshed to ensure timeliness while managing computational resources.

2.1.4 User History Rules

- **Storage Limits:** Rules define what information is stored in user history and for how long. For example, storing detailed analysis results for the most recent 20 searches while keeping only summary data for older searches.
- **Access Control:** Business rules determine who can access historical data and under what conditions, ensuring that users can only access their own history.
- **History Categorization:** Rules govern how history entries are categorized and tagged for easy retrieval and filtering by users.

2.1.5 Live Wall Rules

- **Display Eligibility:** Business rules determine which tweets appear on the Live Wall, possibly filtering out inappropriate content or prioritizing tweets with interesting analysis results.
- **Update Frequency:** Business rules define how often the Live Wall refreshes with new content from the database.

2.2 B) Validation Logic Implementation

Analytica implements several layers of validation logic to ensure data integrity and consistency:

2.2.1 User Input Validation

- **Frontend Validation:** The React/TypeScript frontend implements form validation for user inputs:
  - Hashtag validation to ensure proper formatting (e.g., starts with #)
  - Username validation to ensure valid Twitter usernames
  - Tweet count validation to ensure it's within reasonable limits (e.g., 1-100)
- **Backend Validation:** The Django backend performs additional validation before processing:
  - Validates all required parameters are present

  - Ensures parameters are within acceptable ranges
  - Verifies the analysis type is supported (emotion, sentiment, toxicity)
  - Checks authentication for user history access

2.2.2 Tweet Content Validation

- **Content Verification:** Validates that tweet content is appropriate for analysis:
  - Checks tweet length is sufficient for meaningful analysis
  - Verifies tweet is not primarily media without substantial text
  - Ensures tweet language is supported by the analysis models
- **Model Input Validation:** Prepares and validates tweet text before feeding it to the RoBERTa models:
  - Removes URLs, special characters, and other elements that might interfere with analysis
  - Checks for and handles encoding issues in tweet text

2.2.3 Analysis Result Validation

- **Output Verification:** Validates model outputs to ensure they are within expected ranges:
  - Verifies emotion scores sum to approximately 1.0
  - Ensures sentiment and toxicity probabilities are between 0 and 1
  - Checks for anomalous results that might indicate model errors

2.3 C) Data Transformation Implementation

Analytica implements comprehensive data transformation processes to convert data between the database format and the UI presentation format:

2.3.1 Tweet Data Transformation

- **Scraping to Analysis:** Raw tweets scraped from Twitter are transformed into a structured format suitable for analysis:
  - Extracting relevant fields (text, author, timestamp)

  - Normalizing text content (removing URLs, special characters, etc.)
  - Converting Twitter-specific formatting to plain text
- **Analysis to Storage:** Analysis results are transformed into a format optimized for database storage:
  - Serializing complex model outputs into storable formats
  - Organizing related data into structured documents

2.3.2 Visualization Data Transformation

- **Aggregation Transforms:** Raw analysis data is transformed into formats suitable for visualization:
  - Converting raw emotion scores into percentage distributions
  - Transforming sentiment classifications into chart-friendly data structures
  - Creating time-series data from multiple analyses
- **Chart-Specific Formatting:** Data is formatted specifically for different visualization components:
  - Color-coding data points based on sentiment or emotion values
  - Creating comparison datasets for before/after or trend analysis

2.3.3 Leaderboard Data Transformation

- **Ranking Transforms:** Analysis results are transformed into ranked and categorized data:
  - Computing composite scores based on multiple metrics
  - Applying normalization to ensure fair comparison across different metrics
  - Creating category-specific data structures for leaderboard display
- **UI Presentation Transforms:** Leaderboard data is transformed for optimal UI presentation:
  - Formatting timestamps into human-readable formats
  - Generating summary statistics and highlights for each category

2.3.4 User History Transformation

- **Storage Optimization:** Full analysis results are transformed into space-efficient formats for long-term storage:
  - Extracting key metrics and summaries from detailed results
  - Compressing detailed results when necessary
  - Creating indexed and searchable structures for efficient retrieval
- **History View Transforms:** Stored history data is transformed into user-friendly formats for the history view:
  - Generating timeline visualizations of past analyses
  - Creating searchable and filterable history entries
  - Producing comparison views between historical analyses

2.3.5 API Data Transformation

- **Request Transforms:** User requests from the frontend are transformed into backend-compatible formats:
  - Converting form data into structured API requests
  - Handling pagination and filtering parameters
- **Response Transforms:** Backend data is transformed into frontend-friendly JSON structures:
  - Structuring hierarchical data for easy consumption by React components