SEPTEMBER 22, 2023

EXCHANGE RATE RECOMMENDATION SYSTEM

REQUIREMENT SPECIFICATIONS

Author: Hira Arif

Table of Contents

1 Introduction:

The Currency Exchange Rate Prediction System is a comprehensive software project designed to provide users with a user-friendly web interface to analyze historical currency exchange rate data, make predictions for future exchange rates, and offer recommendations for buying and selling currencies. This system combines data extraction, preprocessing, model training, prediction, recommendation, data visualization, caching, and logging modules to offer a holistic solution for currency traders and analysts.

2 Purpose:

The primary purpose of the Currency Exchange Rate Prediction System is to assist currency traders, analysts, and enthusiasts in making informed decisions related to currency exchange rates. This system aims to provide historical data analysis, predictive modeling, and actionable recommendations to optimize currency trading strategies. Additionally, it promotes data integrity and system efficiency through modular design and extensibility.

3 Scope:

The Currency Exchange Rate Prediction System encompasses several key modules and functionalities:

I. Web Interface: A user-friendly web application that allows users to select base and comparison currencies, choose a historical analysis time, and specify the prediction horizon.
II. Data Extraction: Integration with various data sources, including APIs, CSV files, databases, and web scraping, to gather historical exchange rate data.
III. Data Preprocessing: Ensuring data quality by handling missing values, outliers, and inconsistencies. This module prepares the data for model training.
IV. Model Training and Evaluation: The system supports various predictive models for currency exchange rate forecasting. It evaluates model performance using metrics like accuracy, precision, recall, F1-score, and mean squared error. It triggers retraining if performance degrades significantly.
V. Recommendation System: Utilizing the trained model to provide predictions for the next 30 days and identifying optimal periods for buying and selling specific currencies.
VI. Data Visualization: Enhancing the user experience with data visualization tools, allowing users to explore historical and predicted exchange rate trends using various chart formats.
VII. Data Caching and Logging: Saving crucial information, including trained model paths, system logs, recommendation results, currency selections, analysis types, and timestamps, for efficient retrieval and auditing. Cached data is valid for one day to ensure accuracy.

4 Overview:

The Currency Exchange Rate Prediction System aims to empower users with actionable insights for currency trading by combining data acquisition, preprocessing, modeling, and visualization. Users can access the system via a web interface, select their preferences, and receive predictions and recommendations based on historical data analysis. The system is designed for extensibility, making it easy to incorporate additional data sources, models, or visualization options.

4.1 Product Functions:

The key functions of the Currency Exchange Rate Prediction System include:

- User-friendly web interface for currency analysis.
- Flexible data extraction from APIs, CSV files, databases, and web scraping.
- Data preprocessing to ensure data quality and consistency.
- Model training and evaluation for accurate currency exchange rate predictions.
- Recommendations for optimal buying and selling periods.
- Data visualization for historical and predicted trends.
- Efficient data caching and logging for system optimization and auditing.

5 Functional Requirements per Module:

5.1 Web Interface

FR01. User Interface and Navigation

Req. No.	Functional Requirements
FR01-01	The web interface shall provide a user-friendly design.
FR01-02	Users shall be able to access the web interface via a standard web browser.
FR01-03	The interface shall have a responsive design for various screen sizes and devices.

FR02. Currency List Display:

Req. No.	Functional Requirements
FR02-01	The web interface must display a comprehensive list of available currency units.
FR02-02	The system should present each currency with its code and symbol for clear identification.

FR03. Base Currency Selection

Req. No.	Functional Requirements
FR03-01	Users should be able to select a base currency from the provided list for currency rate analysis.

FR04. Quote Currency Selection

Req. No.	Functional Requirements
FR04-01	Users must be able to select a currency for comparison with the base currency.

FR05. Analysis Time Period Selection

Req. No.	Functional Requirements
FR05-01	The system should offer predefined options for users to select the analysis time (e.g., 1 year, 2 years, specific months).
FR05-02	Custom time period selection should also be supported, allowing users to specify a start and end date.

FR06. Prediction Duration Selection

Req. No.	Functional Requirements
FR06-01	The system must provide the users with the ability to specify the number of days for which they want currency exchange rate predictions.
FR06-02	The system should validate the prediction duration input to prevent errors.

FR07. Input Validation

Req. No.	Functional Requirements
FR07-01	User inputs for base currency, comparison currency, and time periods should be validated to ensure they correspond to valid options.
FR07-02	The system should display Informative error messages for invalid inputs.

FR08. Data Retrieval Trigger

Req. No.	Functional Requirements
FR08-01	The system should initiate data extraction and prediction processes based on user selections of currencies and time periods.
FR08-02	The system should trigger data retrieval automatically upon user input.

FR09. Historical Data Display

Req. No.	Functional Requirements
FR09-01	Historical exchange rate data for the selected currencies and time period should be displayed in a clear and organized manner.

FR09-02	Users should have the ability to view historical data as charts for better analysis.

FR010. Prediction Results Display

Req. No.	Functional Requirements
FR10-01	The system should provide users with the predicted exchange rates for the chosen duration.
FR10-02	Predicted rates should be displayed in an easily understandable format, such as tables or charts.

FR011. Visualization Options

Req. No.	Functional Requirements
FR11-01	Users must have the option to select the type of visualizations (e.g., line charts, bar charts) for data presentation.

FR012. Responsive Design

Req. No.	Functional Requirements
FR12-01	It should ensure that the web interface is responsive and adapts to various devices and screen sizes.
FR12-02	It should provide an optimal user experience on desktops, tablets, and smartphones.

FR013. User Help

Req. No.	Functional Requirements
FR13-01	The system should offer online help resources.

5.2 Data Extraction

FR01. Data Source Selection

Req. No.	Functional Requirements
FR01-01	Users should be able to choose the source of exchange rate data (e.g., API, CSV, database, web scraping).

FR01-02	The system should provide clear options for each data source.

FR02. API Integration

Req. No.	Functional Requirements
FR02-01	The system should be able to retrieve historical exchange rate data from an API.
FR02-02	It should support multiple API endpoints or sources if needed.

FR03. CSV Data Import

Req. No.	Functional Requirements
FR03-01	The system will allow users to upload CSV files containing exchange rate data.
FR03-02	The system will validate the format of uploaded CSV files to ensure compatibility.

FR04. Database Integration

Req. No.	Functional Requirements
FR04-01	The system should enable users to connect to databases to retrieve historical exchange rate data.
FR04-02	The system should support various database management systems (e.g., MySQL, PostgreSQL).

FR05. Web Scraping Capabilities

Req. No.	Functional Requirements
FR05-01	The system should implement web scraping functionality to extract exchange rate data from websites when necessary.
FR05-02	The system must ensure that web scraping is configurable and adaptable to different websites.

FR06. Parameterized Data Duration

Req. No.	Functional Requirements

FR06-01	It must allow users to specify the duration of historical data retrieval (e.g., years=1 by default, years=10).
FR06-02	It must provide flexibility in defining the data duration.

FR07. Data Format Customization

Req. No.	**Functional Requirements**
FR07-01	The system should Include the option to customize the format of retrieved data to meet the requirements of different machine learning models.
FR07-02	The system must Allow users to override default data formatting methods.

FR08. Decoupled Data Input Stream

Req. No.	**Functional Requirements**
FR08-01	The system should ensure that the data input stream is decoupled from the rest of the system for flexibility in adding new data sources.
FR08-02	The system should use design patterns like the Strategy pattern for data source selection.

FR09. Data Validation

Req. No.	**Functional Requirements**
FR09-01	The system should perform data validation tasks to check for missing values, outliers, and data inconsistencies.
FR09-02	The system should implement rules for handling invalid or incomplete data.

FR010. Data Cleaning

Req. No.	**Functional Requirements**
FR10-01	The system should cleanse the data by removing duplicates and irrelevant information.
FR10-02	The system should handle cases of inconsistent or incomplete data.

FR011. Data Transformation

Req. No.	Functional Requirements
FR011-01	The system should provide functionality for transforming data as needed (e.g., feature scaling, encoding categorical data) before training machine learning models.
FR11-02	The system should allow for customization of data transformation methods.

FR012. Data Retrieval Frequency

Req. No.	Functional Requirements
FR12-01	The system should implement a mechanism to check if the selected parameters for data retrieval are already saved in the database for the same day.
FR12-02	The system should avoid redundant data retrieval for the same date and parameters.

FR013. Data Caching

Req. No.	Functional Requirements
FR13-01	The system should cache retrieved data to improve performance and reduce the load on data sources.
FR13-02	The system should define a caching policy to manage the cache effectively.

FR014. Concurrency Handling

Req. No.	Functional Requirements
FR14-01	The system should ensure that data retrieval processes are handled efficiently and can handle multiple user requests simultaneously.
FR14-02	The system should implement concurrency control mechanisms if necessary.

FR015. Data Source Authentication

Req. No.	Functional Requirements
FR15-01	The system should support authentication mechanisms for accessing data sources, such as API keys or database credentials.
FR15-02	The system should also encrypt and securely store authentication information.

FR016. Rate Limiting Handling

Req. No.	Functional Requirements
FR16-01	The system should manage rate limiting imposed by external data sources, including handling API rate limits.
FR16-02	The system should also implement mechanisms for retrying failed requests within rate limits.

FR017. Data Source Availability Monitoring

Req. No.	Functional Requirements
FR17-01	The system should monitor the availability of data sources and provide alerts in case of downtime or issues.
FR17-02	The system should also implement automated recovery or fallback mechanisms when data sources are unavailable.

FR018. Data Source Documentation

Req. No.	Functional Requirements
FR18-01	The system should provide documentation or references for the selected data source(s), including API documentation or CSV format specifications.

FR019. Error Logging

Req. No.	Functional Requirements
FR19-01	The system should log errors and exceptions that occur during data extraction processes for debugging and monitoring purposes, including timestamps and detailed error messages.

FR020. Data Retrieval Performance Metrics

Req. No.	Functional Requirements
FR20-01	The system should track and report the performance of data retrieval operations, including response times and success rates.

FR021. Data Source Switching

Req. No.	Functional Requirements
FR21-01	The system should allow users to switch between different data sources easily.
FR21-02	The system should also handle the transition seamlessly without data loss.

FR022. Data Source Versioning

Req. No.	Functional Requirements
FR22-01	The system should support different versions of data sources and ensure backward compatibility when upgrading to newer versions of those sources.

FR023. Data Source Selection History

Req. No.	Functional Requirements
FR23-01	The system should maintain a history of data source selections and parameters for auditing and analysis purposes.

FR024. Data Source Validation Rules

Req. No.	Functional Requirements
FR24-01	The system should define and manage validation rules for each data source to ensure data integrity and reliability.

FR025. Data Source Connection Pooling

Req. No.	Functional Requirements
FR25-01	The system must Implement connection pooling for database connections to optimize resource usage.

FR026. Data Source Scalability

Req. No.	Functional Requirements
FR26-01	The system should design a data extraction module to scale horizontally to accommodate increasing data volumes.

FR027. Data Source Rate Limit Monitoring

Req. No.	Functional Requirements
FR27-01	The system should continuously monitor data source rate limits and adjust data retrieval strategies accordingly.

FR028. Data Source Data Backup

Req. No.	Functional Requirements
FR28-01	The system must implement regular data backups to prevent data loss in case of data source failures or issues.

FR029. Data Source Data Ownership

Req. No.	Functional Requirements
FR29-01	The system should clearly define ownership and copyright information for data retrieved from external sources and communicate this to users.

5.3 Data Preprocessing

FR01. Data Validation

Req. No.	Functional Requirements
FR01-01	The module must validate incoming data to ensure it meets quality standards.
FR01-02	It should check for missing values, outliers, and data inconsistencies.

FR02. Data Source Flexibility

Req. No.	Functional Requirements
FR02-01	The module should be able to handle data from various sources, including APIs, CSV files, databases, and web scraping.
FR02-02	It should allow for easy integration of new data sources.

FR03. Data Cleaning

Req. No.	Functional Requirements
FR03-01	It should remove duplicate records from the dataset.

FR03-02	It should implement methods to handle noisy data, such as smoothing techniques or filtering.

FR04. Data Transformation

Req. No.	Functional Requirements
FR04-01	The system should provide options for data transformation, including feature scaling and normalization.
FR04-02	The system should also allow for encoding categorical data into numerical format.

FR05. Handling Missing Values

Req. No.	Functional Requirements
FR05-01	The system should implement strategies for handling missing data, such as imputation or removal of incomplete records.

FR06. Outlier Detection

Req. No.	Functional Requirements
FR06-01	The system should identify outliers in the data using statistical methods or machine learning techniques.
FR06-02	The system should also provide the ability to handle outliers, either by removing them or transforming them.

FR07. Data Normalization

Req. No.	Functional Requirements
FR07-01	The system should normalize numerical features to ensure they have a consistent scale.

FR08. Data Quality Metrics

Req. No.	Functional Requirements
FR08-01	The system must Calculate and report data quality metrics such as data completeness, accuracy, and consistency.

FR09. Data Preprocessing Efficiency

Req. No.	Functional Requirements

FR09-01	The system should optimize preprocessing routines for efficiency, especially for handling large datasets
FR09-02	The system should minimize redundant computations.

FR010. Handling Data Skewness

Req. No.	**Functional Requirements**
FR10-01	The system should address data skewness issues, such as log transformations or power transformations, and offer options for choosing the appropriate skewness correction method.

FR011. Data Preprocessing Error Handling

Req. No.	**Functional Requirements**
FR11-01	It should implement error handling mechanisms for unexpected data conditions.
FR11-02	It must Provide clear error messages and debugging information.

FR012. Data Preprocessing Parallelization

Req. No.	**Functional Requirements**
FR12-01	It must Support parallel processing for preprocessing tasks when applicable.
FR12-02	It should Utilize multi-core systems for faster data preprocessing.

FR013. Handling Multi-modal Data Integration

Req. No.	**Functional Requirements**
FR13-01	It should Provide capabilities to integrate and preprocess data from multiple sources or modalities.
FR13-02	The system must Ensure data compatibility and consistency across modalities.

FR014. Handling Data Inheritance

Req. No.	**Functional Requirements**
FR14-01	It should Support inheritance of preprocessing settings and configurations between related datasets.
FR14-02	It must Streamline the preprocessing process for similar datasets.

5.4 Model Training and Evaluation

FR01. Training Algorithm

Req. No.	Functional Requirements
FR01-01	The module must implement a machine learning algorithm for model training.
FR01-02	The algorithm should support various types of models (e.g., regression, classification) based on project needs.

FR02. Training Data Split

Req. No.	Functional Requirements
FR02-01	The system should divide the dataset into training and validation sets for model training.
FR01-02	The split ratio should be adjustable and documented.

FR03. Hyperparameter Tuning

Req. No.	Functional Requirements
FR03-01	There should be an option to tune hyperparameters of the machine learning model for optimization.

FR04. Cross-Validation

Req. No.	Functional Requirements
FR04-01	The system should Implement k-fold cross-validation to assess model performance.
FR04-02	The number of folds should be configurable.

FR05. Model Persistence

Req. No.	Functional Requirements
FR05-01	The module must save the trained model for later use.
FR05-02	Model storage should allow for remote and local storage options.

FR06. Model Import

Req. No.	Functional Requirements
FR06-01	The system should support importing pre-trained models from remote machines.

FR06-02	It must Ensure compatibility with model formats (e.g., TensorFlow, Py Torch).

FR07. Model Evaluation Metrics

Req. No.	Functional Requirements
FR07-01	It should Implement evaluation metrics such as accuracy, precision, recall, F1-score for classification models.
FR07-02	The model should include mean squared error (MSE) for regression models.

FR08. Performance Thresholds

Req. No.	Functional Requirements
FR08-01	The system must define performance thresholds for model evaluation (e.g., minimum accuracy).
FR08-02	The module should trigger retraining if performance falls below specified thresholds.

FR09. Confusion Matrix Generation

Req. No.	Functional Requirements
FR09-01	It must generate confusion matrices for classification tasks to analyze model performance in detail.
FR09-02	The system should allow users to access and visualize these matrices.

FR010. Model Training Logs

Req. No.	Functional Requirements
FR10-01	It must log training progress, including training loss and evaluation metrics at each epoch.
FR10-02	It must Store logs for future reference and analysis.

FR011. Batch Training

Req. No.	Functional Requirements
FR11-01	It must implement batch training to handle large datasets efficiently.

FR012. Feature Selection

Req. No.	Functional Requirements

FR12-01	It must Allow users to select and customize the features used for model training.

FR013. Parallel Processing

Req. No.	**Functional Requirements**
FR13-01	It should enable parallel processing for faster model training on multi-core systems.
FR13-02	It must optimize resource utilization.

FR014. Training Visualization

Req. No.	**Functional Requirements**
FR14-01	It must display training progress and model performance visually through plots and graphs.

FR015. Model Assembling

Req. No.	**Functional Requirements**
FR15-01	It must support model assembling techniques for improved predictions.
FR15-02	The system must allow users to combine multiple models and define ensemble strategies.

FR016. GPU Acceleration

Req. No.	**Functional Requirements**
FR16-01	It must Utilize GPU acceleration for faster model training when available.

5.5 Recommendation System

FR01. Prediction Generation

Req. No.	**Functional Requirements**
FR01-01	The system shall generate currency exchange rate predictions for the next 30 days.
FR01-02	Predictions should consider historical data, user-selected currencies, and analysis time period.

FR017. Best Buying and Selling Periods

Req. No.	**Functional Requirements**

FR02-01	The system shall identify and display the optimal periods for buying and selling the specific currency based on predicted results.

FR018. Trend Analysis

Req. No.	Functional Requirements
FR03-01	The system shall provide trend analysis, showing historical and predicted trends for the selected currency.
FR03-02	Trend information should be available in graphical format by default.

FR019. Prediction Confidence Levels

Req. No.	Functional Requirements
FR04-01	The system shall provide confidence levels or intervals for its predictions to indicate the level of uncertainty.

FR020. Performance Metrics

Req. No.	Functional Requirements
FR05-01	The system shall calculate and display performance metrics for the prediction model, such as Mean Absolute Error (MAE) or Root Mean Square Error (RMSE).

FR021. User Education

Req. No.	Functional Requirements
FR06-01	The system shall provide educational resources to help users understand the significance of the recommendations and trends provided.

5.6 Data Visualization

FR01. Historical Data Visualization

Req. No.	Functional Requirements
FR01-01	The system must provide visualizations of historical exchange rate data in various timeframes (e.g., daily, weekly, monthly).
FR01-02	Users should be able to select specific currencies and date ranges for historical data visualization.

FR02. Predicted Data Visualization

Req. No.	Functional Requirements
FR02-01	The system must generate visualizations for predicted exchange rate trends for the next 30 days.
FR02-02	Users should have the option to view predicted data in different chart formats.

FR03. Customization of Charts

Req. No.	Functional Requirements
FR03-01	Users should have the ability to customize chart appearance (e.g., colors, labels, legends).
FR03-02	Customization options should be intuitive and user-friendly.

FR04. Zoom and Pan Functionality

Req. No.	Functional Requirements
FR04-01	Charts should support zooming in on specific time periods and planning to explore data in detail.
FR04-02	Zoom and pan should be smooth and responsive.

FR05. Cross-Device Compatibility

Req. No.	Functional Requirements
FR05-01	The data visualization module should be accessible and responsive on various devices (e.g., desktop, tablet, mobile).
FR05-02	The system must ensure compatibility with different screen sizes and resolutions.

FR06. Interactive Tooltip

Req. No.	Functional Requirements
FR06-01	Charts should provide interactive tooltips that display data points' values on hover.
FR06-02	Tooltips should include relevant information and be customizable.

FR07. Technical Indicators

Req. No.	Functional Requirements
FR07-01	The system should support the overlay of technical indicators (e.g., moving averages, Bollinger Bands) on charts.

FR07-02	Users can configure and customize these indicators.

FR08. Responsive Legends

Req. No.	Functional Requirements
FR08-01	Legends accompanying charts should dynamically adjust to the number of plotted elements for clarity.
FR08-02	Legends should not overcrowd the chart area.

5.7 Data Caching and Logging

FR01. Save Trained Model Path

Req. No.	Functional Requirements
FR01-01	The system must save the file path or location of the trained machine learning model which allows for easy retrieval and loading of the model for subsequent predictions.

FR02. Logging System Integration

Req. No.	Functional Requirements
FR02-01	The module should integrate with a logging system to record system events, errors, and actions taken.
FR02-02	Logging should include timestamps, severity levels, and descriptions.

FR03. Log Recommendation Results

Req. No.	Functional Requirements
FR03-01	The system should record the results of currency trading recommendations in the system's log.
FR03-02	It must Include details like recommended actions (buy/sell), currency pairs, and prediction confidence levels.

FR04. Cache Prediction Results

Req. No.	Functional Requirements
FR04-01	The system should implement a caching mechanism to store currency prediction results.
FR04-02	Cached results should be accessible for a predefined duration (e.g., 1 day) for quicker retrieval.

FR05. Save Base and Quote Currency

Req. No.	Functional Requirements
FR05-01	The system should store the base and quote currencies selected by the user for each prediction.

FR06. Cache Analysis Type

Req. No.	Functional Requirements
FR06-01	The system should cache the type of analysis performed (e.g., 1 year, 2 years) for each prediction. This information is crucial for reproducing and analyzing past predictions.

FR07. Store Datetime Information

Req. No.	Functional Requirements
FR07-01	The system should record the date and time when each prediction was made and saved in the cache for reference and auditing purposes.

FR08. Data Storage Efficiency

Req. No.	Functional Requirements
FR08-01	The system should optimize the storage of cached data to minimize system resource usage.
FR08-02	It must the system should optimize the storage of cached data to minimize system resource usage.

FR09. Event Logging

Req. No.	Functional Requirements
FR09-01	The system should implement a comprehensive event logging system to capture system events, errors, and actions.
FR09-02	It should Include event timestamps, severity levels, and descriptions in the logs.

6 Non-Functional Requirements

NFR01. Modularity and Extensibility

Req. No.	Non-Functional Requirements
NFR01-01	The system should be decomposed into modules with no more than 2000 lines of code per module, promoting maintainability and extensibility.

NFR01-02	The system must support the addition of new modules or plugins without affecting the core architecture. The time to integrate into a new module should not exceed 2 weeks.

NFR02. Response Time

Req. No.	**Non-Functional Requirements**
NFR02-01	The web interface must respond to user interactions within 300 milliseconds on average, ensuring a smooth user experience.
NFR02-02	The average time to retrieve historical data from external sources should not exceed 500 milliseconds per request.

NFR03. Scalability

Req. No.	**Non-Functional Requirements**
NFR03-01	The system should be able to handle at least 1000 concurrent users without degradation in performance, maintaining a response time under 1 second.

NFR04. Reliability

Req. No.	**Non-Functional Requirements**
NFR04-01	The system should have an uptime of at least 99.9%, with planned maintenance windows communicated in advance.
NFR04-02	System errors and exceptions should be logged with a rate of no more than 1 error per hour.

NFR05. Data Storage Efficiency

Req. No.	**Non-Functional Requirements**
NFR05-01	The system should optimize data storage to limit redundancy, ensuring that the database size doesn't grow by more than 10% per month.

NFR06. Code Maintainability

Req. No.	**Non-Functional Requirements**
NFR06-01	At least 80% of the codebase should be adequately documented with inline comments, promoting code understandability and maintainability.

NFR07. Error Recovery

Req. No.	Non-Functional Requirements
NFR07-01	In case of system failures or external service disruptions, the system should gracefully degrade functionality rather than crashing, ensuring uninterrupted user access.
NFR07-02	Aim for a system recovery time of no more than 15 minutes in the event of a critical failure, minimizing downtime.

NFR08. Generalizability

Req. No.	Non-Functional Requirements
NFR08-01	The system should run on multiple platforms, including Windows, Linux, and macOS, without platform-specific code modifications.
NFR08-02	Ensure compatibility with various data sources (CSV, databases, web scraping) without significant code changes.

NFR09. Resource Utilization

Req. No.	Non-Functional Requirements
NFR09-01	The system must comply with data privacy regulations, such as GDPR or HIPAA, with documented processes for data handling and consent management.

NFR10. Concurrent API Requests

Req. No.	Non-Functional Requirements
NFR10-01	The system should be able to handle a minimum of 100 API requests per second for data extraction without a significant increase in response time.

7 Requirement Statistics:

Modules	7
Functional Requirements	99
Non – Functional Requirements	10