Big Data - Machine Learning Project and Beyond: Stock Price Prediction and XRP Ledger Volume Forecasting

Alexandre Amalric
XRPL Commons - Ripple

January 2025

Abstract

This paper documents a two-phase machine learning exploration, starting with a stock price prediction task from a curated project list and extending to a real-world application: forecasting transaction volumes on the XRP Ledger. Initially, the project focused on predicting Apple Inc.'s stock price (AAPL) using XGBoost, leveraging lagged variables, temporal features, and hyperparameter optimization via Optuna. The model demonstrated robust performance, providing actionable insights into market trends and highlighting the effectiveness of advanced time-series techniques.

Encouraged by the results, the approach was extended to a more complex and data-intensive problem provided by my employer at Ripple: predicting transaction volumes on the XRP Ledger. This second project introduced significant Big Data challenges, requiring the processing of a 40GB `ledger.db` file and an 8.2TB `transaction.db` file. SQL queries were employed to filter, aggregate, and extract relevant features from these massive datasets, reducing the data size while retaining critical information. Advanced models, including RNNs, GRUs, LSTMs, and Transformers, were implemented to capture intricate patterns in blockchain transaction activity.

The research emphasized extensive data preparation, incorporating on-chain features, external market indicators, and social media sentiment to enhance model accuracy. Key insights included the discovery and implications of a negative R^2, which occurs when model predictions perform worse than a constant mean predictor. This finding underscores the challenges of handling high volatility, dynamic temporal dependencies, and anomaly detection in blockchain ecosystems. By linking machine learning to Big Data methodologies, this paper demonstrates the scalability and adaptability of predictive frameworks to real-world applications. The resulting framework offers significant benefits to validators, investors, and network operators within the XRPL ecosystem, bridging theoretical knowledge and practical solutions in the dynamic domains of finance and blockchain.

1 Introduction

1.1 Background and Motivation

This project began with a prescribed machine learning task: predicting stock prices using advanced time-series models. Stock price prediction is a widely studied problem due to its real-world relevance and the availability of extensive historical data. Using historical data for Apple Inc. (AAPL), I implemented a predictive framework with XGBoost, applying rigorous feature engineering and hyperparameter tuning to improve accuracy. The results demonstrated strong performance and validated the effectiveness of the chosen methods.

Inspired by the success of this task, I decided to apply these techniques to a real-world challenge posed by my employer at Ripple: forecasting transaction volumes on the XRP Ledger. Unlike stock price prediction, blockchain transaction forecasting involves heterogeneous data types, high volatility, and complex temporal patterns, making it an excellent testbed for advanced machine learning methods.

1.2 Significance of the Study

Accurate transaction volume forecasting is crucial for blockchain stakeholders, enabling better resource allocation, anomaly detection, and financial decision-making. By comparing stock price prediction and blockchain forecasting, this study demonstrates the adaptability of time-series methods and provides a scalable framework for real-world applications. The findings contribute to enhanced operational efficiency and risk management within the XRPL ecosystem.

2 Opinion

The stock price prediction task provided a valuable introduction to machine learning and predictive modeling, as the XGBoost model achieved strong performance with a low RMSE and meaningful insights into temporal market trends. Initially, I was surprised by how well the model handled the volatility inherent in stock price data, which motivated me to explore whether the same techniques could be applied to a more dynamic and complex problem: forecasting transaction volumes on the XRP Ledger.

Transitioning to XRP Ledger forecasting introduced a new layer of complexity, not only in terms of the domain but also in terms of the scale and nature of the data. Blockchain transaction data is inherently dynamic and influenced by external factors such as market sentiment, regulatory news, and airdrop events. To tackle these challenges, I had to integrate heterogeneous datasets, including on-chain transaction data, market indicators, and social media sentiment. A significant technical hurdle was managing the massive datasets provided by Ripple: a 40GB `ledger.db` file and an 8.2TB `transaction.db` file. These Big Data challenges required learning and applying SQL to filter and aggregate the data effectively.

Extracting relevant information from the `ledger.db` dataset involved writing optimized SQL queries to reduce its size while preserving essential features. For example, I aggregated daily transaction volumes by counting the number of closed ledgers per day, reducing the dataset from 40GB to a manageable 10GB. Despite these efforts, I could not process the full 8.2TB `transaction.db` dataset due to computational limitations, which highlighted the importance of scalable infrastructure for Big Data analysis.

This experience deepened my understanding of the unique challenges posed by Big Data in machine learning projects. Working with such large-scale datasets required careful planning to manage storage, memory, and computational resources efficiently. It also underscored the need for robust preprocessing pipelines to transform raw data into usable features while avoiding bottlenecks in data handling.

Another challenge was addressing the occurrence of a negative R^2, which indicated that the model's predictions were less accurate than using the mean value as a constant predictor. This finding highlighted the limitations of certain models in capturing the high volatility and sudden transaction spikes common in blockchain data. It also emphasized the importance of choosing appropriate evaluation metrics and interpreting them in the context of the problem.

To overcome these challenges, I explored advanced models like LSTMs and Transformers, which are better suited for capturing long-term dependencies in sequential data. LSTMs provided a structured way to manage temporal dependencies through their gating mechanisms, while Transformers introduced the flexibility of self-attention, allowing the model to weigh the importance of different time steps dynamically. These techniques offered valuable insights into seasonal patterns and provided a more robust framework for forecasting transaction volumes.

Despite these advancements, the project revealed the limitations of existing data and infrastructure. Missing auxiliary data, such as real-time regulatory news or more granular sentiment metrics, limited the model's ability to generalize effectively. This underscored the critical role of feature engineering and the need for integrating diverse data sources in future work.

The dual challenges of handling Big Data and designing effective machine learning models have significantly expanded my technical skills and problem-solving abilities. From managing SQL queries and preprocessing pipelines to experimenting with advanced architectures, this project bridged the gap between theoretical concepts and practical applications. It has also inspired me to delve further into hybrid models that combine the strengths of tree-based methods and deep learning architectures, as well as explore distributed computing frameworks for large-scale data processing. This journey has reinforced my passion for addressing complex, real-world challenges through machine learning and data-driven solutions.

3 Implementation Description

3.1 Stock Price Prediction Using XGBoost

The stock price prediction task included the following steps:

- **Data Preprocessing**: Historical stock data from Yahoo Finance was preprocessed to create lagged features (e.g., 1-day to 12-day lags) and temporal variables (e.g., day of the week, month).
- **Model Optimization**: Hyperparameters for XGBoost were tuned using Optuna, optimizing RMSE as the evaluation metric. The best parameters were used to train the final model.

- **Evaluation and Visualization**: Model performance was evaluated using RMSE, MAE, and residual analysis. Confidence intervals were calculated to assess prediction uncertainty.

3.2 XRP Ledger Transaction Volume Forecasting

The XRP Ledger forecasting task required addressing significant challenges related to data extraction, processing, and modeling. The following steps outline the approach:

- **Data Collection and Aggregation**: The data for this task was extracted from two main sources:
  - **Ledger.db (40GB)**: This SQLite database contains information about validated ledgers on the XRP Ledger. Each ledger represents a set of transactions validated approximately every 3-4 seconds, with around 100 transactions per ledger.
  - **Transaction.db (8.2TB)**: This database contains detailed information on individual transactions. Due to its massive size, handling and processing this data was infeasible within the scope of this project.

  To reduce the size of the dataset and focus on relevant features, SQL queries were used to filter and aggregate data. For instance, the `ledger.db` dataset was reduced from 40GB to 10GB by:

  - Selecting ledgers between August 2017 and November 2019.
  - Aggregating daily transaction counts by counting the number of closed ledgers per day, which provides a reliable proxy for daily transaction volume.

  I then failed to enrich the dataset with external features. I did not manage to add historical XRP price, trading volume, and social media sentiment metrics obtained from the CoinGecko API.

- **Feature Engineering**: Feature engineering played a critical role in improving the accuracy of the forecasting models. Key steps included:
  - Creating **lagged features** to capture autocorrelation in transaction volumes. For example, features like `volume_t-1`, `volume_t-7`, and `volume_t-30` were generated to capture daily, weekly, and monthly patterns.
  - Encoding **temporal indicators** such as day of the week, month, and quarter to model seasonality and periodic trends.
  - Adding **external data features**, including market sentiment scores and token-specific transaction types, to capture the influence of external events on transaction volumes.

- **Data Normalization and Splitting**: To prepare the data for machine learning models:
  - **Normalization**: Min-max scaling was applied to scale transaction volumes between 0 and 1, ensuring consistency in model training:

    $$x' = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

  - **Data Splitting**: The data was split into training (70%), validation (15%), and testing (15%) sets to evaluate model performance.

- **Model Selection and Training**: Advanced time-series models were trained to capture the complex temporal dependencies in transaction volumes:
  - **GRUs and LSTMs**: These recurrent neural networks were employed to capture medium- and long-term dependencies in the data. Their gating mechanisms allowed the models to retain or discard information dynamically, improving their ability to handle sequential data.
  - **Transformers**: Leveraging self-attention, Transformers provided the ability to weigh the importance of different time steps dynamically. This made them particularly effective for long-sequence forecasting.

  Hyperparameters for these models were optimized using techniques like grid search and Optuna, which minimized RMSE and ensured the models were fine-tuned for the dataset.

- **Evaluation and Error Analysis**: The models were evaluated using RMSE, MAE, and R^2. A significant finding was the occurrence of a negative R^2, where model predictions performed worse than using the mean value as a constant predictor. This highlighted the challenges of handling high volatility and complex temporal dependencies in blockchain data. Residual analysis was conducted to identify patterns in prediction errors, and confidence intervals were calculated to assess prediction uncertainty.

4 Conclusion

This project illustrates the adaptability of machine learning methods across diverse use cases, from stock price prediction to blockchain transaction forecasting. The initial task provided a foundation in time-series modeling, while the XRP Ledger forecasting extended these techniques to a more dynamic and complex domain. Key insights include the challenges of handling heterogeneous Big Data, addressing anomalies, and interpreting a negative R^2. These findings underscore the importance of rigorous evaluation and iterative model improvement. Future work will explore hybrid architectures and enhanced feature engineering to further improve forecasting accuracy.

5 Bibliography

- Analytics Vidhya, "Machine Learning Projects for Big Data," `https://www.analyticsvidhya.com/blog/2024/12/machine-learning-projects/`.
- Tianqi Chen and Carlos Guestrin, *XGBoost: A Scalable Tree Boosting System*, 2016. Available at: `https://xgboost.readthedocs.io/`
- Yahoo Finance, `https://finance.yahoo.com`.
- Optuna Documentation, `https://optuna.org`.
- S. Elsworth and S. Güttel, *Time Series Forecasting Using LSTM Networks: A Symbolic Approach*, arXiv preprint arXiv:2003.05672, 2020. Available at: `https://arxiv.org/abs/2003.05672`
- D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, *DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks*, International Journal of Forecasting, vol. 36, no. 3, pp. 1181–1191, 2020. DOI: `https://doi.org/10.1016/j.ijforecast.2019.07.001`
- Ripple, "XRP Ledger Developer Portal," `https://xrpl.org`.