**FINAL REPORT**

1. **Summary**

The objective of the project was to test whether the market gains of major companies within the Nasdaq 500 index could be outperformed using machine learning-based return predictions. Weekly return predictions were made, and profits were calculated based on the buy-sell transactions of the top 5 stocks with the strongest predictions for each week. The stocks that were no longer among the top 5 strongest predictions in the following week were sold, while the position weights of the remaining stocks in the top 5 were increased. Additionally, a 5% take-profit condition was implemented. Optimization was performed based on the ROI and Sharpe ratios calculated from the transactions at the end of the year. Additionally, metrics such as RMSE and MAE were calculated for the weekly predictions.

1. **ETL Process (PySpark with PostgreSQL)**

The dataset was obtained using the yfinance library by downloading the following stock data for the period 2012–2024: ['AAPL', 'NVDA', 'MSFT', 'AMZN', 'META', 'ADBE', 'TSLA', 'FFIE', 'ASTI', 'ALLR']. The dataset included the fields: 'Date', 'High', 'Low', 'Open', 'Close', 'Adj Close', and 'Volume'.

The data was transformed into a PySpark DataFrame, and the multi-index structure was removed. The column names were converted to lowercase, and any spaces were removed. Stock names were added as a column. Missing values in the Date column were identified and removed, and the Date column was converted to a datetime format. As a result, a multi-stock DataFrame was created. At this step, no additional processes were deemed necessary, as the EDA (Exploratory Data Analysis) phase was not yet performed.

During the data loading process into PostgreSQL, a table schema was created. If a table with the specified name already existed in the database, it was updated. Otherwise, a new table was created and the data was loaded into the database.

1. **Exploratory Data Analysis (EDA) & Visualization**

**3.1 Data Retrieval and Basic Statistics**

The daily fetched data was read from PostgreSQL, and basic statistical metrics were calculated to understand the dataset's characteristics.

**3.2 Candlestick Chart**

Using the price data, candlestick charts were created to visualize stock price movements over time, providing a clear view of opening, closing, high, and low prices.

**3.3 Distribution and Time-Series Analysis**

The distributions and time trends of key metrics such as **returns**, **price differences**, **volume**, and **volatility** were analyzed.

Time-series plots were generated to explore the behavior of these metrics over the analyzed period.

**3.4 Technical Indicators**

Popular technical indicators commonly used in the market, such as moving averages, Relative Strength Index (RSI), MACD, and Bollinger Bands, were calculated.

These indicators were examined to gain insights into stock price movements and trend analysis.

**3.5 Focus on Price Movements and Trend Tracking**

Particular attention was paid to how the calculated metrics and technical indicators correlated with stock price movements and the detection of market trends. This analysis provided valuable insights for feature engineering in the modeling phase.

The analysis revealed that making price predictions using daily data would not yield realistic results. It was concluded that, especially without incorporating additional data, the model would struggle to capture short-term fluctuations effectively.

To improve the **Signal-to-Noise Ratio**, the idea of transitioning from daily to weekly predictions was explored. This approach was deemed the most logical due to the following reasons:

* **Reducing Noise**: Aggregating data weekly helped mitigate abrupt changes and outliers.
* **Maintaining Prediction Precision**: Weekly predictions retained a reasonable prediction interval without sacrificing significant detail.

As a result, the decision was made to transform the data into weekly intervals and focus on predicting **weekly returns**. This adjustment aimed to balance noise reduction and model accuracy effectively.

**4. Modelling**

**4.1 Experiments with Daily Data**

In the modeling phase, experiments were conducted using daily data. Both **Gradient Boosted Decision Tree (GBDT)** models and **LSTM** models were tested for regression and classification tasks.

**4.2 Observations**

* **Price Prediction (Regression Tasks)**:
  + Regardless of the model used, predictions closely aligned with the previous day's values, resulting in overfitting.
  + The models optimized to minimize error metrics such as RMSE and MAE, but this led to predictions failing to capture meaningful changes in price trends.
* **Classification Tasks (Positive/Negative Close)**:
  + Accuracy scores ranged between **50% and 53%**, indicating that the models struggled to differentiate between positive and negative closures effectively.
  + When reframed as a signal-based classification problem (e.g., return > 2: buy, return < -2: sell), the dominant class (**hold**) overwhelmingly comprised most of the predictions, rendering the model impractical for actionable insights.

**4.3 Conclusion on Daily Data**

The results highlighted significant limitations in using daily data for both regression and classification tasks. These findings further reinforced the decision to transition to **weekly return predictions**, where models could potentially perform better by focusing on aggregated trends and reducing noise.

**4.2 Experiments with Weekly Data**

**4.2.1 Target and Models**

With weekly data, the target was defined as the return of the following week. Various GBDT models, including CatBoost, XGBoost, and LightGBM, were utilized for predictions.

**4.2.2 Initial Optimization**

* Metrics-Based Optimization:
  + Models were optimized using metrics such as RMSE, MAE, and MAPE.
  + While the results showed a slight improvement compared to daily data, they were deemed insufficient for practical use.

**4.2.3 Backtesting and New Strategy**

* A new strategy was developed based on backtesting results:
  + Predictions were made each week, and the top 5 stocks with the strongest predictions were selected for purchase.
  + Positions were either maintained or terminated in the following week based on whether the stocks remained in the top 5 predictions.
* Evaluation Metrics:
  + Optimization was shifted to focus on ROI and Sharpe Ratio derived from backtesting results.
  + While the model was trained on the training dataset, ROI and Sharpe values were calculated on the test dataset to validate its performance.

**4.2.4 Hyperparameter Tuning and Feature Selection**

* Optuna was used for:
  + Hyperparameter tuning to improve model performance.
  + Feature selection to identify the most impactful features.

**4.2.5 Final Model Training**

The training process was divided into two groups:

1. Group 1: Models were optimized to maximize the Sharpe Ratio.
2. Group 2: Models were optimized to maximize ROI.

Each group included one model from CatBoost, LightGBM, and XGBoost.

**4.2.6 Ensemble Learning**

* A stacking ensemble technique was applied as the final step.
* The outputs of the models from both groups were combined and passed through a Random Forest model for final predictions.

This approach allowed the ensemble model to leverage the strengths of individual models while balancing the objectives of maximizing Sharpe and ROI.

**5. Conclusion**

When the validation and prediction datasets were evaluated, the ensemble model achieved a **Sharpe ratio of 1.07** and a **ROI of 47%** for the year 2024. In comparison, the benchmark index, **^IXIC (Nasdaq 500 Composite)**, had a **Sharpe ratio of 1.73** and a **ROI of 29%**.

**Key Observations:**

* Although the results are less favorable in terms of **reward-to-risk** (Sharpe ratio), the **ROI** achieved by the model is nearly double that of the benchmark.
* The model's performance varied across the test, validation, and prediction periods, indicating challenges in **generalizing market trends and movements** effectively.
* Despite these limitations, the results are promising, especially considering that the 2024 dataset, and particularly the unseen 2023 data, posed significant challenges for the model.

**Overall Assessment:**

While the model has room for improvement in terms of risk management and generalization, the significant ROI demonstrates its potential in leveraging machine learning for stock market predictions. With further refinements, such as incorporating additional features or external datasets, the model could achieve even more competitive results.