Sabancı University-Data Analytics Master Program Term Project
Duruhan Yalçın

# Stock Price Prediction for Income Generation using RNN and Machine Learning Models

## Abstract

This research project aims to create predictive models for generating income through stock trading, with a focus on the NASDAQ Composite. The approach employs a blend of Recurrent Neural Networks (RNNs), XGBoost, custom loss functions, and meticulous data preprocessing techniques.

## Introduction

Stock price prediction has traditionally focused on predicting various metrics like close prices and candle colors. This project goes a step further by optimizing predictive models to facilitate profitable trading while minimizing risk.

## Methodology

### Data Preprocessing

1. **Data Collection**:

    - The study uses historical NASDAQ Composite data, enriched with technical indicators like SMA, EMA, MACD, RSI, and external indices from other stock markets, cryptocurrencies, and commodities.

    **Technical Indicators**

    To capture market psychology and trading patterns, various technical indicators were also added:

    - **Trend Indicators:** SMA (Simple Moving Average), MACD (Moving Average Convergence Divergence)
    - **Momentum Indicators:** RSI (Relative Strength Index), Momentum, Stochastic Oscillator
    - **Volatility Indicators:** Bollinger Bands, ATR (Average True Range)
    - **Volume Indicator:** OBV (On-Balance Volume)

    These indicators provide insights into market direction, volatility, momentum, and trading volume, which are essential for accurate stock price prediction.

    **External Indices, Cryptocurrencies, and Commodities**

    To provide a more holistic view of the market conditions and trends, the following external indices, cryptocurrencies, and commodities were integrated into the dataset:

    - **Stock Indices:** ^DJI, ^GSPC, ^FTSE, ^GDAXI, ^FCHI, ^N225, ^HSI, ^AXJO, ^KS11, ^BSESN, ^BVSP
    - **Commodities:** GC=F (Gold Futures), SI=F (Silver Futures), CL=F (Crude Oil WTI Futures), NG=F (Natural Gas Futures), ZC=F (Corn Futures), ZW=F (Wheat Futures)
    - **Cryptocurrencies:** BTC-USD (Bitcoin against the US Dollar)
    - **Volatility Index:** ^VIX (CBOE Volatility Index)

    These features offer a broad economic context that may influence the NASDAQ Composite, enhancing the model's predictive power.

2. **Time Features**:
    - Day of the week and week of the year were added to capture temporal patterns.
    - Cycle encoding was used on these time features to maintain their cyclical nature.

   **Advantages of Cycle Encoding**

   Cycle encoding captures the cyclical nature of certain features like days of the week or weeks of the year, allowing the model to better understand periodic fluctuations and improve its predictive capability.

3. **Data Cleaning**:
    - Features with heavy tails were log-transformed. Negative values were offset by adding the absolute minimum of the feature distribution to each data point before log transformation.
    - Outliers beyond 1.5 IQR from the median were suppressed.
4. **Data Normalization**:
    - A MinMax scaler was applied to the features.
    - Correlated features were dropped using the Feature Engine library.
    - These steps were conducted within a pipeline.

**Model Architecture**

1. **RNN Models**:
    - SimpleRNN, LSTM, and GRU were used.
    - A custom function was developed for hyperparameter tuning, taking multiple inputs including layer count, neuron count, batch size, activation and loss functions.
2. **Machine Learning Models**:
    - XGBoost was used, and hyperparameters were tuned using Optuna.
3. **Custom Loss Functions**:
    - Two custom loss functions were designed for percentage change prediction, one to maximize profit and another to minimize loss.

**Evaluation Metrics**

RMSE was used for close price prediction, and F1 Score was used for candle color prediction. Custom loss functions were employed for percentage change prediction.

**Results**

1. **Close Price**: RNN models outperformed XGBoost but did not beat naive predictions.
2. **Candle Color**: XGBoost was more effective but not significantly better than naive models.
3. **Percentage Change**: The LSTM model trained with the custom loss function aimed at maximizing profit was the most profitable in simulated trading scenarios.

### RNN Model Performance

The table below presents the trading performance results for the RNN models:

| Explanation | Cumulative Profit | Compounded Profit | Nr of Loss Days | Nr of Profit Days |
|---|---|---|---|---|
| SimpleRNN Without Custom Loss Function | -236.20 | -91.06 | 238 | 962 |
| LSTM Without Custom Loss Function | -382.08 | -97.93 | 236 | 964 |
| GRU Without Custom Loss Function | -274.42 | -93.98 | 275 | 955 |
| SimpleRNN With Custom Loss Function | -282.24 | -94.46 | 297 | 933 |
| LSTM With Custom Loss Function | -371.71 | -97.73 | 236 | 964 |
| GRU With Custom Loss Function | -377.81 | -97.87 | 264 | 936 |
| SimpleRNN With Custom Loss Function, Technical Indicators and Indices | 146.54 | 314.24 | 331 | 899 |
| LSTM With Custom Loss Function, Technical Indicators and Indices | 453.85 | 9020.32 | 20 | 1210 |
| GRU With Custom Loss Function, Technical Indicators and Indices | 175.85 | 473.72 | 91 | 1139 |

### XGBoost Model Performance

The table below outlines the trading performance results for the XGBoost models:

| Explanation | Cumulative Profit | Compounded Profit | Nr of Loss Days | Nr of Profit Days |
|---|---|---|---|---|
| Only technical indicators | 105.54 | 181.30 | 85 | 1115 |
| Technical indicators and Other indices | 151.42 | 338.41 | 196 | 1004 |
| Technical indicators, Other indices and Feature engine | 160.95 | 384.18 | 171 | 1029 |
| Only technical indicators with custom loss | 159.34 | 382.80 | 75 | 1125 |
| Technical indicators and Other indices with custom loss | 155.25 | 356.11 | 192 | 1008 |
| Technical indicators, Other indices and Feature engine with custom loss | 161.84 | 389.58 | 154 | 1046 |
| Only technical indicators with custom loss (maximize) | 142.65 | 308.85 | 70 | 1130 |
| Technical indicators and Other indices with custom loss(maximize) | 194.61 | 578.39 | 128 | 1072 |
| Technical indicators, Other indices and Feature engine with custom loss(maximize) | 163.20 | 396.29 | 152 | 1048 |

### Trading Profit/Loss Evaluation

A function was created to evaluate the actual trading performance based on the model's predictions. XGBoost models using custom loss functions showed significant profitability, particularly the model optimized for profit maximization.

### Discussion

The project shows the potential of custom loss functions and machine learning models like XGBoost in achieving the goal of profitable trading.

### Conclusion

While RNN models did not perform as expected for trading, XGBoost models trained with custom loss functions generated significant profits, highlighting the effectiveness of specialized machine learning techniques and loss functions.

### Future Work

1. **Generalizability**: To test the model's robustness across different stocks or indices.
2. **Time Frame**: To evaluate the model's performance on weekly or monthly data.
3. **Optimization**: Further tuning of the custom loss functions and other hyperparameters.