

# QR Interview Assessment

## Objective

This assessment is designed to evaluate your problem-solving skills, modeling strategy, and ability to work with time-series financial data. You are required to build a short-term price trend prediction model by implementing the approach outlined in the provided research paper: *Short-Term Stock Price-Trend Prediction Using Meta-Learning*.

## Reference Material

You will be provided with the above research paper along with this assessment. Your task is to implement the trend prediction model described in the paper using OHLC data and a selection of five indicators from those referenced in the paper.

## Data Usage

Use any reliable historical data source compatible with `backtesting.py`. You may select one of the following assets as your modeling instrument:

- SPY (S&P 500 ETF)
- BTCUSD (Bitcoin/USD pair)
- AAPL (Apple Inc.)

Use the same instrument for both modeling and backtesting. Limit your dataset to 1 year of daily OHLCV data. Use technical indicators as features if relevant to your modeling approach. You may use either a local Python environment such as Jupyter or Colab or a local Python environment for data processing and model training.

## Constraints

- Use only daily OHLC (Open, High, Low, Close) data and five technical indicators from the paper.
- Do not use any additional external features or alternative indicators.
- Model training must be done in a local Python environment such as Jupyter or Colab or locally (e.g., Colab, Jupyter).
- The trained model should be saved and loaded into [backtesting.py](#) for backtesting and inference.

## Assessment Tasks

1. Select one instrument (SPY, BTCUSD, or AAPL) and prepare 1 year of daily OHLCV data.
2. Select and compute five indicators from those listed in the research paper.

3. Implement a labeling scheme (e.g., slope-detection) to generate rise/fall labels.
4. Build and train a deep learning model (e.g., CNN or TCN) as outlined in the paper.
5. Save your trained model and load it into a `backtesting.py` backtest script.
6. Implement a simple trading strategy:
  - Buy when the model predicts 'rise'
  - Sell or hold otherwise
7. Run a backtest over the 1-year period and evaluate performance.
8. Submit code, a demo video, and a short report explaining your design decisions and observations.

## Evaluation Criteria

- Implementation of the model logic based on the paper
- Justification for indicator selection
- Soundness of data labeling and model architecture
- Proper use of `backtesting.py` for backtesting inference
- Code organization and reproducibility
- Clarity and depth of written explanation
- Effectiveness and clarity of the demo video

## Deliverables

- Python scripts or notebooks for preprocessing, modeling, and training
- Trained model file (e.g., `.pt` or `.h5`)
- `backtesting.py` backtest script (`main.py`)
- 1–2 page report summarizing your implementation approach, decisions, and findings
- A brief demo video walkthrough (3–5 minutes)

## Important Notes

This assessment is designed to evaluate your reasoning, structure, and ability to interpret academic work. Accuracy is not the primary goal, we are interested in how you approach the problem, explain your decisions, and implement a clean, logical solution. Please use the same instrument (SPY, BTCUSD, or AAPL) for both modeling and backtesting. You may use either `backtesting.py`'s Research Environment or a local setup for the research and training phase. Ensure that your data format is compatible with `backtesting.py` requirements (OHLCV in Pandas DataFrame).