

# Configuration Manual for Time Series Forecasting: Gold Price Prediction

## Overview

This configuration manual serves as a step-by-step guide to reproduce the experiments described in the project. It includes environment setup, dataset preparation, execution instructions, and tool-specific configurations.

## System Requirements

- Operating System: Windows 10/11, macOS, or Linux
- Python Version: 3.8+
- Hardware Requirements:
  - Minimum 8GB RAM (16GB recommended for LSTM training)
  - GPU (optional but recommended for LSTM models)

## Environment Setup

### 1. Install Python and Dependencies

Ensure Python 3.8+ is installed on your system. You can download it from [python.org](https://python.org). Install the required Python packages by running:

```
'''
```

```
pip install -r requirements.txt
```

```
'''
```

Dependencies include pandas, numpy, scikit-learn, xgboost, tensorflow, and matplotlib.

### 2. Create a Virtual Environment

To isolate dependencies and avoid conflicts:

```
'''
```

```
python -m venv gold_env
source gold_env/bin/activate # On Linux/Mac
gold_env\Scripts\activate   # On Windows
pip install -r requirements.txt
```

```
'''
```

## Directory Structure

Organise the project files as follows:

```
...
GOLD
├── README.md      # Documentation
├── requirements.txt # Dependencies
├── src
│   └── forecasting.ipynb # Source code (Jupyter Notebook)
├── data           # Dataset(s)
│   ├── Gold_1h.csv # Original dataset
│   └── crudeOil.csv # Auxiliary dataset
├── figures        # Output plots
│   ├── xgboost_fold_1.png
│   └── lstm_fold_1.png
...
```

## Dataset Configuration

Ensure the following datasets are available in the `/data` directory:

### 1. Gold Price Dataset:

- File Name: `Gold\_1h.csv`
- Columns: Timestamp, Open, High, Low, Close, Volume (after preprocessing)
- Format: CSV, hourly intervals

### 2. Crude Oil Dataset:

- File Name: `crudeOil.csv`
- Columns: Timestamp, Close, Volume (after preprocessing)
- Format: CSV, hourly intervals

## Execution Instructions

### 1. Running the Jupyter Notebook

Navigate to the `src` directory and launch the Jupyter Notebook:

```
'''
```

```
cd src
jupyter notebook forecasting.ipynb
'''
```

Execute each cell sequentially to perform preprocessing, feature engineering, model training, and evaluation.

### 2. Training Models

XGBoost:

- Add lagged features and rolling statistics via the notebook.
- Train the model using 5-fold TimeSeriesSplit.

LSTM:

- Reshape the dataset into 3D tensors for sequential learning.
- Train the model with the specified architecture.

## Output Configuration

Visualisations and metrics are stored as:

- Actual vs Predicted Plots: `/figures` directory as `.png` files. (Can be stored as .pdf also for better clarity )
- Error Distribution: Visualised for XGBoost.
- Rolling Mean Analysis: Plots for XGBoost in original dataset experiment.

## Troubleshooting

Common Issues:

- Ensure all dependencies are installed using `pip install -r requirements.txt`.
- Verify dataset headers match the expected format.

GPU Acceleration:

- Install TensorFlow with GPU support: (For reducing training time while LSTM training)

```
'''
```

```
pip install tensorflow-gpu
'''
```

## Contact and Support

For additional inquiries, reach out to:

- **Email:** x22234926@student.ncirl.ie
- **GitHub:** <https://github.com/abhisheksuga/price-forecasting/>