

README: Nifty50 Forecasting

Barbieri Claudio, Delera Giacomo, Hauser Cyril, Howe Alessandro

R Code

The implemented models are:

- `linear_model+Trading.R` – contains the linear regression model.
- `dynamic_model+Trading.R` – contains the VAR model on OHLC data.

Both scripts require `Nifty50_daily.csv` as the input dataset. This dataset is already available in the folder, but note that it is a processed version of the original dataset.

R Auxiliary Functions

The script `read_data.R` takes the original dataset as input and generates an RDATA file (`lists_of_datasets` necessary to run `plot_data.R` and `build_data.R`). The original dataset has been excluded from the folder on account of the 5 GB size of the file.

The LSTM model implemented in Python requires the following input files:

- `nifty_lagged_indicator.csv`
- `common_daily_SPY_filtered_lag_tech_ind.csv`
- `common_daily_nifty_filtered_lag_tech_ind.csv`

For simplicity, these files were generated using R.

The first file is used by the "standard" LSTM model and can be created using `linear_model+Trading.R`.

The other two files are generated using `integrate_spy.R`, with `nifty_lagged_indicator.csv` and `daily_SPY.csv` as input.

All required datasets are already available in the folder, so there is no need to regenerate them.

Python Code

`LSTM.ipynb` is a Jupyter notebook where we developed the Long Short-Term Memory (LSTM) model.

As mentioned in the previous section, it requires the following files: `nifty_lagged_indicator.csv`, `common_daily_SPY_filtered_lag_tech_ind.csv`, and `common_daily_nifty_filtered_lag_tech_ind.csv`. To ensure the code runs without modifications to file paths, we recommend using Google Drive. To execute the notebook, upload it along with the three required datasets into the same folder in your Drive and then run the code.

`GARCHINMEAN.ipynb` is a Jupyter notebook where we developed the GARCH model. The code requires `Nifty_50.csv`. The same suggestions of the previous notebook hold.