Forecasting stock prices using Chronos for use in trading.

Overview :

Developed a simple framework to forecast stock prices using historical data. Derive insights from the forecast and use it to generate actions namely ‘buy’, ‘sell’ or ‘hold’. Evaluation is performed with the given framework and compared against the blotter.

Task : Generate trading signals using a transformer model and historical data.

Solution space :

- Solution 1:
  - Process the data to generate trading signals by looking into the future and create a dataset.
  - Train a transformer model to take different input signals and create a classification model that produces an output signal which is ‘buy’, ‘sell’ or ‘hold’.
  - Use the model in real time when sufficient data becomes available and trade based on the signals produced
- Solution 2 :
  - Pose the problem as a time series problem instead of a classification problem.
  - Train a transformer model to take historic input signals and output forecasts of the signals.
  - Use the forecasted data to generate action signals namely ‘buy’, ‘sell’ or ‘hold’
  - Use the model in real time to make decision while trading.

I chose to proceed with solution 2, as it provides more human flexibility and control to modify existing trading model to generate actions based on real time events without having to retrain the model.

I used the framework of Chronos[1], for forecasting. Chronos treats the forecasting problem as a regular classification problem by an LLM by quantizing different values of the time series and tokenizing them. The problem becomes to predict the next token as in natural language.

Steps :

1) Preprocessed to data for training chronos model
2) Feature extraction, selected small subset of features to train the chronos model including the closing price which is regarded as the price of the stock
3) Trained the model with the dataset created in the previous step. The model is periodically checkpointed and the best model checkpoint is used for final evaluation.
4) Generated forecasted data for each and every data point based on fixed context size of historical data and stored it in a csv file.
    a) For the first 100(#history) data points where no historical data is available, the action is hold, as the model does not have enough data to generate a reliable signal.
    b) The price is forecasted for the next n days, we check the maximum percentage change and the minimum percentage change and store it.
    c) The data is stored in a csv file.
5) Generated actions from the forecasted data.
    a) The data from the previous step is loaded and appended with the main data.
    b) If the maximum percentage change in the next n days is positive beyond a certain threshold, 'buy' action is triggered and is subject to other conditions (same as the trade blotter)
    c) If the minimum percentage change in the next n days is negative lower than a certain threshold, 'sell' action is triggered.
    d) Otherwise hold action is triggered and nothing happens.
6) Use the actions to trade the stock at a given point of time.

Evaluation :
Models :

- original pretrained chronos model which was trained as a generic forecasting model that has a very good zero shot performance.
- Chronos model trained on the given dataset.
- Signal generated with RSI value.

Evaluation is done on the same environment provided, only the trading action is generated differently for each model.

Surprisingly the original pretrained model performs better than the model trained specifically on the given dataset. I believe that the reason is that the original model is trained on very large number of time series datasets which includes all kinds of forecasting problems making the model robust for a variety of different problems. Having experimented with different settings to generate the signal, the best outcome in terms of cumulative reward is always obtained for the pretrained model, followed by the model trained from scratch and then the trading blotter provided in the notebook.

	Different configurations (max_pct, min_pct)			
	(0.008,-0.003)	(0.001,-0.001)	(0.005,-0.005)	(0.01,-0.01)
pretrained	0	-112.55	0	0
trained_scratch	-2.2006	-22037	-22037	0

The above table depicts performance in terms of cumulative reward of pretrained model and model trained from scratch on this specific dataset for various configurations. We observe that the pretrained model outperforms the model trained on scratch. It could be different if trained on more data.

- The trading blotter has a cumulative reward of -12231 for the given data. If configurations are adjusted correctly, both the models discussed above perform better than the blotter.

References

[1] Chronos: Learning the Language of Time Series, https://arxiv.org/pdf/2403.07815