Predicting stock returns using news headline and sentiment

Author: Haotian Zhang hz2708

1. Method

My methodology is composed of 4 steps.

- I clean the data by using Robust Scaler standardization and filling NA with median.
- I introduce the features, (for news sentiment, I use aggregated mean group by RP_ENTITY_ID and date) whereas we set up our two targets: 1. monthly_return (binary) 2. monthly_return_F1 (binary).
- I define the setup of the three machine learning methods we employ, namely random forest, XGBoost and GRU.
- Finally, evaluate by accuracy score and confusion matrix.

2. Model

2.1 Tree Based Method

2.1.1 Model specification for Random forest

As first model, I use random forests introduced by Ho (1995) and expanded by Breiman (2001), with the following parameters:

- Number of decision trees in the forest = 1000 (part A); 1500 (part B)
- Maximum depth of each tree = 8 (part A); 7 (part b)
- For every split, we select m := ⌊√p⌋ features randomly from the p = 40 features in the data, see Pedregosa et al. (2011). I refer to (Krauss et al., 2017, Subsubsection 4.3.3) and (Fischer & Krauss, 2018, Section 3.4) for further details regarding random forests.

2.1.2 Model specification for XGBoost

- I use XGBoost which is frequently used in Kaggle, with the following parameters:
- Number of decision trees in the forest = 170 (part A); 380 (part B)
- Maximum depth of each tree = 3 (part A); 3 (part B)

2.2 Deep Learning model

2.2.1 Model specification for GRU

Introduced by Cho, et al. in 2014, GRU (Gated Recurrent Unit) aims to solve the vanishing gradient problem which comes with a standard recurrent neural network. GRU can also be considered as a variation on the LSTM. LSTM is another popular recurrent neural network introduced by Schmidhuber & Hochreiter (1997). In my parameter setting, GRU performs well in fitting training and test set, so I choose GRU. I created a model with 50, 25, 10 cells of GRU, followed by a dropout layer of 0.1 and then a dense layer of 1 output nodes with tanh activation function.

- Loss function: mean_squared_error
- Optimizer: RMSProp (with the keras default learning rate of 0.001)
- Batch size: 512
- Early stopping: patience of 20 epochs, monitoring the validation loss
- Validation_data: test set.

3. Results

Some interesting findings have been included in the notebook. Besides:

1. My empirical results show that the model performance based on 80-20 train_test_split set is as following:

TABLE 1: Prediction Results on test sets

	Random Forest	XGBoost	GRU
Prediction Accuracy for Current Month Return	65.40%	68.10%	58.10%
Prediction Accuracy for Next Month Return	60.80%	59.30%	60.20%

Note: numbers are accuracy scores based on test set (2017/05-2020/12), I change labels into binary by predicting whether stock goes up or down at that month because basically we are more concerned about which stocks to buy or short and the accuracy can be better measured.

The prediction results seems reasonable since all model achieves about 60% accuracy in the test set. The XGBoost performs well in current month return prediction.

2. EVENT_SENTIMENT_SCORE and some sentiment measurements (like BEE) are important in the tree-based prediction since both RF and XGBoost classifiers share these similar important features.

3. I suppose Structured data may be suitable for simple machine learning models like random forest, but the GRU is not fully tuned so there is some potential improvement to do.

4. Further evaluations may be applied in future. My initial idea to to set up a long-short portfolio (long 5 top stocks, short 5 bottom stocks) and measure the Sharpe Ratio compared to some benchmark(S&P500 or equal weight portfolio).