MAFS5140 Statistical Methods in Quantitative Finance
Assignment Final Report

YangShihao 20660753
The Hong Kong University of Science and Technology

1 Investment objective & Trading strategy

Nowadays, many people are keen on applying machine learning methods or deep learning method in quantitative finance. The stock price prediction are proved to be valid in many cases. When it comes to cryptocurrency which has more volatility and less influence factors, we used the model which has good performance in quantitative trading of stocks to forecast cryptocurrencies price. That is deep learning and momentum models. Unlike the stock market, cryptocurrency trading has more volatility and risk, which means that the opening conditions, clearance conditions, etc. should be taken into account strictly in the model. At the same time, the transaction cost of cryptocurrencies trading is much smaller than that in stock trading, so high-frequency trading is more suitable for cryptocurrencies. All in all, our goal is to carefully adjust positions and maximize sharpe ratio on the basis of accurate forecasting.

1.1 Prediction Model

Before the model is built, the data is briefly cleaned, for example, the unnecessary features are deleted, the bar data are normalized. The time series prediction problem is a problem, it is not like the general regression model. The input variable predicted by a time series is a set of chronological sequences of numbers. It is both continuous and random, so it is more difficult to build comparing to regression model. To take advantage of the memory capabilities of the RNN model, we used the LSTM model, the following figure1 shows a diagram of the LSTM architecture.

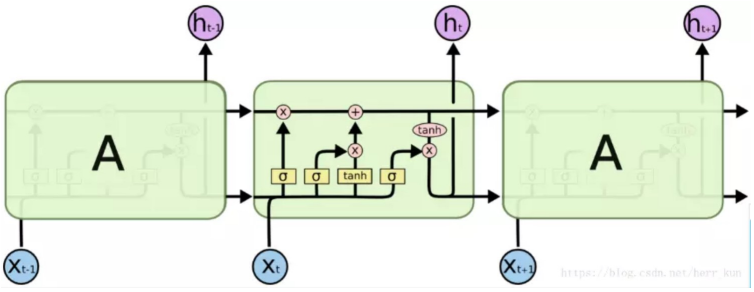

Figure 1: LSTM principle

figure2 shows our LSTM model architecture.

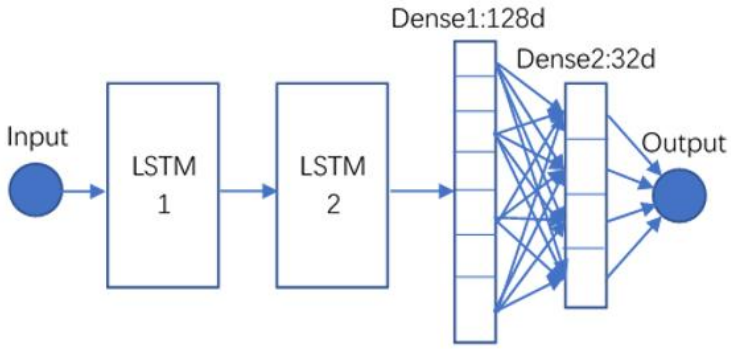

Figure 2: LSTM Structure

About model parameters. We selected the bar_length = 30, and we use average price in the next 30 minutes as label, also the first 22 minutes of minute-level data as input to do LSTM training. Dataset conditions and parameters are as follows:

Table 1: Dataset Description

Dataset	
Training dataset	(671998,22,7)
Test dataset	(74666,22,7)

Table 2: Parameter Description

Parameter	value
Imput shape	[512,22,7]
Output shape	[128, 128, 32, 1]
Dropout	0.2
Loss function	MSE
optimizer	Adam
batch_size	512
epoch	30

After obtaining the forecast results, we have a around judgment on the future price trend of cryptocurrencies, and then we have an advanced momentum strategy.

1.2 Main Trading Strategy

First set bar_length = 30 which means we trade every 30 minutes. The bar data is stored in data_list waiting to be used. At trading time, export historical data, and for certain cryptocurrency it needs to be calculated

separately. Then we get 22 minutes of cryptocurrency data. After normalized processing and training using the LSTM model, we obtain average price prediction in next 30 minutes. Furthermore, we calculate the ratio of forecast price and the current minute price, as the standard K for buying and selling. At the same time, the rise or fall of the currency is recorded in the history. The specific purchase rules are as follows:

Table 3: threshold

Threshold	Action
K>1.1	buy(50%cash_balance)
K<0.95	sell(100%)
K<0.98	sell(80%)

At the same time, in order to reduce the ris, we have adopted a strict momentum opening strategy, when the position is 0, it is necessary to be increasing in last three consecutive 30 minutes before choosing to buy some cryptocurrencies, otherwise we will not buy. To be clearly,

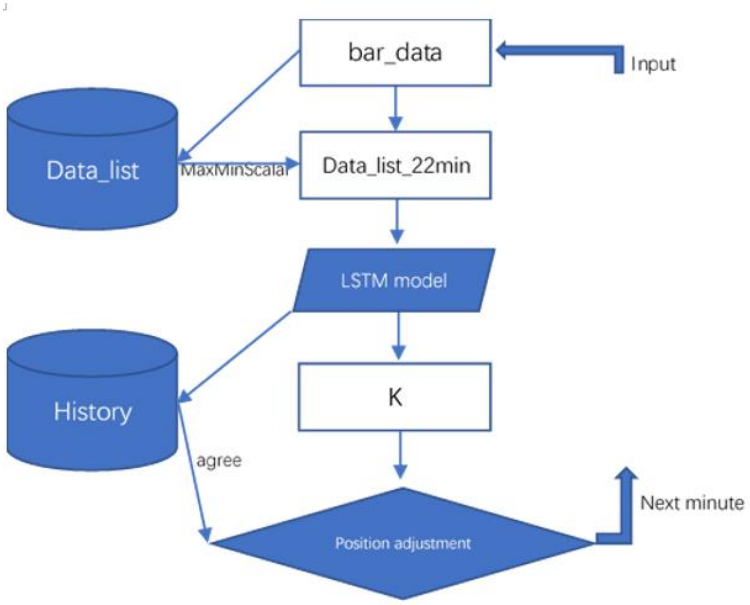

Figure 3: total strategy flow

2 Strategy Analysis

2.1 Algorithm Comparison

We compares serval machine learning methods: LinearRegression(LR),long short-term memory (LSTM), RandomForest(RF), support vector machine (SVM), and XGBoost to test which one performs the best in predicting the cryptocurrency trend. It’s good news for us that we try lots of algorithm to train this dataset

not only for finding the best structure but also for getting the internal knowledge of the data.

2.1.1 LinearRegression(LR)

It is easy to train a LinearRegression model using python. But this time we use statsmodels.api and here is the summary:

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0004	0.000	-1.187	0.235	-0.001	0.000
open	-0.0393	0.021	-1.841	0.066	-0.081	0.003
high	0.1253	0.024	5.194	0.000	0.078	0.173
low	0.1080	0.022	4.819	0.000	0.064	0.152
close	0.8066	0.022	35.937	0.000	0.763	0.851
volume	0.0840	0.055	1.532	0.126	-0.023	0.191
quote asset volume	-0.1130	0.064	-1.757	0.079	-0.239	0.013
number of trades	0.0044	0.002	2.340	0.019	0.001	0.008
taker buy base asset volume	0.0768	0.085	0.900	0.368	-0.090	0.244
taker buy quote asset volume	-0.0701	0.100	-0.699	0.485	-0.267	0.127

Figure 4: Summary

It is clearly that P-value of the last serval factor is very high which means that we are very sure that its coefficient is 0, that is, the importance of these factor is low. The t value of the attribution “close” reaches 35.937, indicating that the factor has a large influence on the label.

2.1.2 RandomForest(RF)

To train the model using RandomForest, we focus on the advantages of Tree model. To reduce the running time we use n_estimators=5, max_depth=20 in our training. Then got the feature importance.

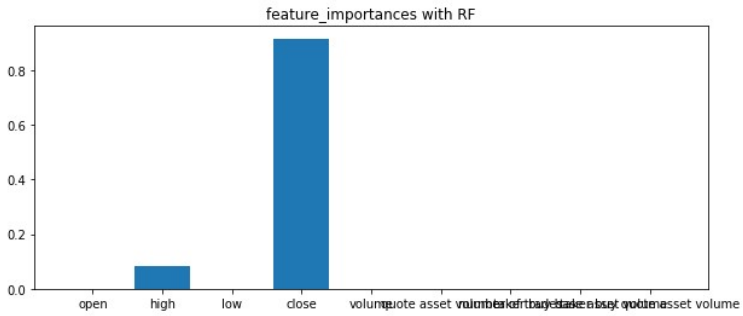

Figure 5: Feature_importance for RandomForest(RF)

2.1.3 XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. Because of the tree model, we can also get the feature importance.

It’s interesting that xgboost gives high assessment to the low price in one minute. I think this may result from the regularization term in the loss function. Yes we can forever believe xgboost.

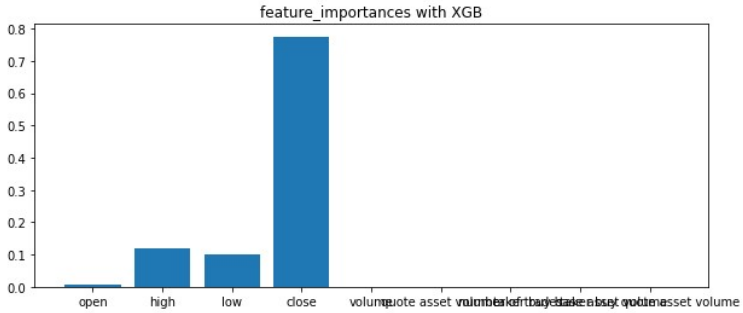

Figure 6: Feature_importance for XGBoost

2.1.4 LSTM

According to the last three model which has no preference for time series data, we selected the feature by the result of our finding. The structure is introduced in the last chapter. In the test dataset we plot the close price like this:

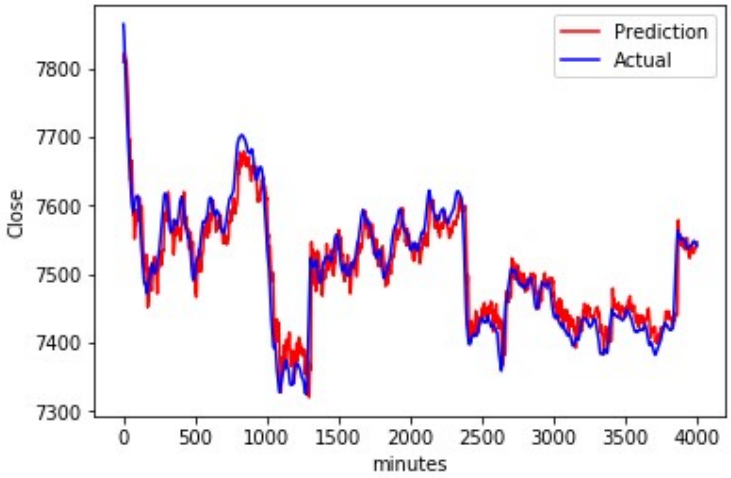

Figure 7: Test outcome

Remind that our LSTM model predict the average price in future 30min, Although there are still some differences, trend is almost similar. Here is the results of machine learning model.

Table 4: threshold

Model	loss
LSTM	0.02991
XGBoost	0.04024
RandomForest	0.04366
svm(kernel='rbf')	0.07223
svm(kernel='linear')	0.37824
svm(kernel='poly')	0.48119

So LSTM performed better than the other algorithm. So we finally chose LSTM. And there are many other excellent method to predict the cryptocurrencies price like reinforcement learning, Generative adversarial network, and also GRU. However the prediction is only a part in our strategy, so LSTM is enough to deal with the problem. Then the problem is transformed to a classification problem. For these four cryptocurrency, here is the output:

	Accuracy	Precision	Recall	F1
BTCUSDT	0.547	0.562	0.351	0.432
ETHUSDT	0.545	0.475	0.134	0.209
LTCUSDT	0.559	0.551	0.128	0.209
XRPUSDT	0.530	0.477	0.136	0.213

Figure 8: Accuracy

2.2 Further analysis

2.2.1 Data Analysis

In the last part, we discuss the LinearRegression. But in that case, we use the label "next minute close price". So it is possible that the feature has good correlation with the label because they only have One minute difference. So we change the label to "the mean of next 10 minute close price", "the mean of next 30 minute close price" and "the mean of next 60 minute close price". First the covariance matrix is figure9. Also there are summary table for the three models.

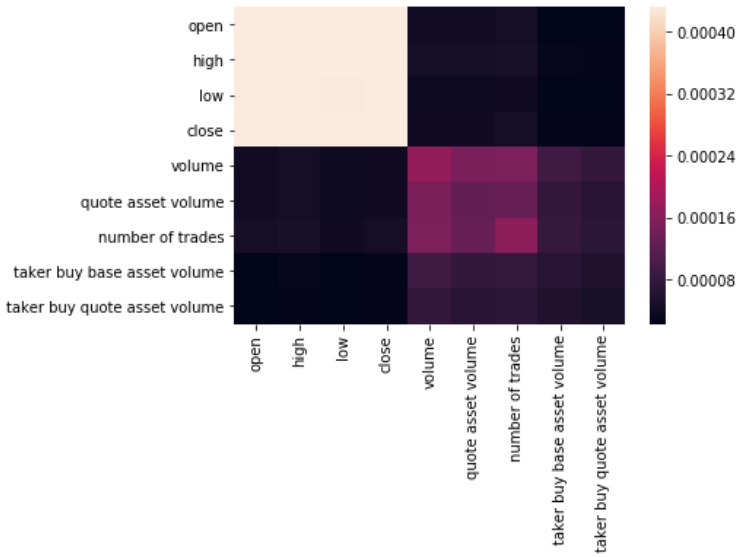

Figure 9: Covariance matrix

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0008	0.001	1.146	0.252	-0.001	0.002
open	0.0791	0.039	2.014	0.044	0.002	0.156
high	-0.0450	0.044	-1.012	0.311	-0.132	0.042
low	0.1295	0.041	3.136	0.002	0.049	0.210
close	0.8348	0.041	20.183	0.000	0.754	0.916
volume	0.1243	0.101	1.230	0.219	-0.074	0.322
quote asset volume	-0.1493	0.118	-1.260	0.208	-0.382	0.083
number of trades	0.0034	0.003	0.991	0.322	-0.003	0.010
taker buy base asset volume	0.2870	0.157	1.826	0.068	-0.021	0.595
taker buy quote asset volume	-0.3172	0.185	-1.717	0.086	-0.679	0.045

Figure 10: Regression summary for "the mean of next 10 minute close price"

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0028	0.001	2.705	0.007	0.001	0.005
open	0.1085	0.062	1.749	0.080	-0.013	0.230
high	-0.1840	0.070	-2.623	0.009	-0.321	-0.047
low	0.1341	0.065	2.059	0.039	0.006	0.262
close	0.9362	0.065	14.346	0.000	0.808	1.064
volume	0.5117	0.159	3.210	0.001	0.199	0.824
quote asset volume	-0.5809	0.187	-3.107	0.002	-0.947	-0.214
number of trades	-0.0063	0.005	-1.160	0.246	-0.017	0.004
taker buy base asset volume	0.0901	0.248	0.363	0.716	-0.396	0.576
taker buy quote asset volume	-0.1110	0.292	-0.381	0.703	-0.682	0.460

Figure 11: Regression summary for "the mean of next 30 minute close price"

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0059	0.001	4.152	0.000	0.003	0.009
open	0.1261	0.085	1.483	0.138	-0.041	0.293
high	-0.2976	0.096	-3.094	0.002	-0.486	-0.109
low	0.1560	0.089	1.746	0.081	-0.019	0.331
close	1.0047	0.089	11.228	0.000	0.829	1.180
volume	0.9864	0.219	4.513	0.000	0.558	1.415
quote asset volume	-1.1362	0.256	-4.432	0.000	-1.639	-0.634
number of trades	-0.0066	0.007	-0.895	0.371	-0.021	0.008
taker buy base asset volume	-0.0155	0.340	-0.046	0.964	-0.682	0.651
taker buy quote asset volume	0.0071	0.400	0.018	0.986	-0.776	0.791

Figure 12: Regression summary for "the mean of next 60 minute close price"

With the time span grows, R square decrease which means the relation reduces between feature and label. When we predict the next minute close price, there are three feature('high','low','close') with p value equals to 0. When we predict the mean of next 10 minute close price, 'close' is the only one with p value equals to 0. When we predict the mean of next 60 minute close price, 'close"volume'and 'quato asset volume' have p value equal to 0. That's to say, when time span grows, the volume related feature may have more importance.

2.2.2 Opening condition

Opening condition is part of the momentum strategy, which means if the forecast outcome tells us that we should buy, meanwhile the position is just 0, then it is necessary to check whether there is a continuous growth in last w bars. If so, we decide to buy, if not, then we continue to keep 0 positions. The same threshold w is also used when the forecast outcome tells us clear all the position. For different w, the total return is different, here is the result coming from the backtest:

Table 5: w selection

w	total return
0	0.02146
1	0.03132
2	0.04096
3	0.0397
4	-0.00072
5	-0.00443

Finaly w = 2 is selected to be our best parameter which has good performence when facing with the volitality.

3 Summary

This paper first introduces the strategic objectives and strategy architecture, introduces the details of the LSTM model and the training process. The second part shows the advantages of analyzing and comparing various machine learning models and processing time series data in the LSTM model. Finally, the highlights of the importance analysis and strategy of using linear regression model are added, and the rules of opening conditions are added, and the course project is completed on the basis of the above results.