SSE Composite Index Case Study

(Project 2 of 2018 Spring CS282.01 Introduction to Machine Leaning of the School of Information Science and Technology (SIST) of ShanghaiTech University)

Kuang Haofei[1] and Yuan Yijun[1]

[1]School of Information Science and Technology(SIST), ShanghaiTech University

***Abstract*—In this part of project, we need to the Shanghai Composite Index history data to predict the fellow five days maximal and minimal price. Here, we use these original data to generate training dataset and test dataset, and compare the performance with different regression model. Besides, we also discuss some skills to promote the perform of data, like feature selection and cross-validation. Finally, we choose a most suitable model to give prediction of the data from June 4th to 8th.**

I. Introduction

Prediction of the movement of stock market is a long-time attractive topic to researchers from different fields. [1]. And it is also a difficult topic of machine leaning field. Try to use stock market history data to training a model by using machine leaning method is a good ways to promote data process and implement algorithm ability.

In this project, we will focus on Shanghai Composite Index data, and use it's history data to predict the future 5 days maximal values and minimal values. In this report, we will talk about our method of data process and model comparison.

In section II, we will discuss the data process method in detail. In the section III, we will demonstrate our experiment of different model and give the final prediction. And we will summarize our work in the conclusion section.

II. Data Process

In this section, we discuss our methods of data process. We will talk about the detail of how to design a method to generate a dataset with features and labels from original data, and also talk about our method of feature selection and time series split method.

A. Design Dataset

In order to achieving a regression task, we need a dataset to train our models, so the most important thing is get a dataset. In this project, our task is predict future values by using history data. But the original data just contain the whole values of each day. So, we have some criterion to generate features and labels, then build the relationship of feature and label.

The historical datasets of Shanghai composite index are used in this study, each dataset is a series of daily open, max, min and so on, contain 9 items totally. And we use the period from December 19th, 1990 to June 1st, 2018. Here, we use five day as features, and predict the fellow several day maximal value and minimal value. So, it contains 45 features for each sample.

For example, the TABLE I exhibit how to generate features and labels of predict fellow one days. f means the features, and the number 1 to 10 means the date. That means we joining together 5 consecutive days as features, the maximal values of 6th days' data as label. The two part consist a sample. And here is predict fellow two day, the label change to the maximal values of 7th days' data. And use the seem way to generate another dataset. After generating the dataset, we also implement standardization operate of it.

TABLE I: Divide Features and Labels

f_1	...	f_9	......	f_{41}	...	f_{45}	y
	1		......		5		6
	2		......		6		7
	3		......		7		8
	4		......		8		9
	5		......		9		10

B. Feature Selection

In last part, we change original data to a dataset. But each samples contain 45 features, and we just have about six thousand samples. The features is too more for the problem, and it would easy to cause overfitting if we use all features. So, we have to use some methods to remove some unimportant features.

SelectKBest is one of the method of feature selection in scikit-learn library [2], we use it to achieving feature selection. It will select features according to the k highest scores. The process of compute the scores contain 2 steps: a)

1) Computing the cross correlation between each feature and the target:

$$R = \frac{(X_i - mean(X_i)) * (y - mean(y))}{std(X_i) * std(y)} \quad (1)$$

Where X_i is i^{th} feature.

2) It is converted to an F score then to a p-value. Select k highest score features according to p-value.

Here, we do this operate for minimal values prediction and maximal values prediction respectively. And we choose 23 features(about a half of all whole feature).

C. Split Training Data and Test Data using Cross-validation

Because we don't have test set now, we cannot use all samples to train the model. The dataset will be split into training set and test set. Here, we use cross-validation to split it into training set and test set. But there is exist a problem which is the general cross-validation method cannot be used to split time series data.

Fortunately, scikit-learn [2] provide a method called TimeSeriesSplit, and it provides train/test indices to split time series data samples that are observed at fixed time intervals, in train/test sets. In each split, test indices must be higher than before, and thus shuffling in cross validator is inappropriate. The method is based on K-fold cross-validation, and we use 10-fold cross-validation to get the training set and test set here.

III. Experiment

In last section, we talk about how to generate the dataset. And we will discuss how to choose a nice regression model by using the dataset in this section. We compare with several mainstream models and choose the most suitable one to get the final result.

A. Model Comparison

In this project, we implement five models by using scikit-learn [2]:

1) Linear Model:
    a) Linear Regression
    b) Ridge Regression
2) Decision Tree Regression
    a) Random Forest Regression
    b) Extra Forest Regression
3) Support Vector Machine Regression(SVR [3])

We use the data from December 19th, 1990 to May 31st, 2018 training these models after data process and we just use the data to predict fellow one day. The evaluate criterion are R^2 score and *Mean Square Error*(MSE), like Fig. 1 and Fig. 2.

And we found the R^2 score are not much difference between linear model and decision tree model, but the SVR is relative small than each other. And we also could found the MSE of SVR is obviously higher than other models, so we thing SVR is not suit for the problem.

According to results, we decide to use Ridge Regression to predict the final values.

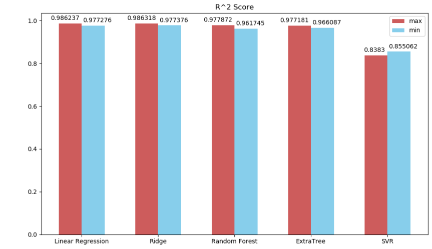

Fig. 1: The R^2 scores of each models, the red part is the result of the dataset which is used to predict maximal values. And the blue part is the result of the dataset which is used to predict minimal values.

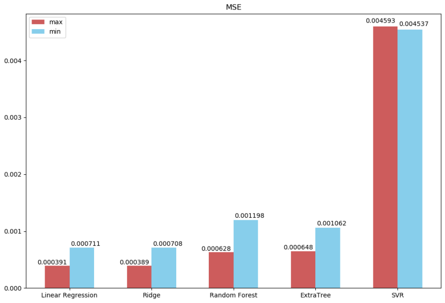

Fig. 2: The *Mean Square Error* of each models, the red part is the result of the dataset which is used to predict maximal values. And the blue part is the result of the dataset which is used to predict minimal values.

B. Predict Final Values

Here, we use Ridge Regression as our model, and training it by using five datasets(predict June 4th, 5th, 6th, 7th and 8th maximal and minimal values respectively). And our Final results are shown as TABLE II

TABLE II: Final prediction of June 4th to June 7th

	4th	5th	6th	7th	8th
maximal values	3090.33	3087.83	3091.59	3092.95	3091.21
minimal values	3043.77	3038.93	3038.05	3043.01	3040.91

IV. Conclusion

In this project, we compare several models for predict future values of Shanghai Composite Index and give our prediction of future 5 days. By our observation, the prediction has some errors from ten to a hundred, it just has a little influence to the model, but it is not accepted in stock market. So, maybe the prediction are useless to stock, but it is a good problem to think about the machine learning with the real word.

References

[1] T. Fu, S. Chen, and C. Wei, "Hong kong stock index forecasting," 2013. [Online]. Available: http://cs229.stanford.edu/proj2013/FuChenWei-HongKongStockIndexForecasting.pdf

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[3] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, “Support vector regression machines,” in Advances in neural information processing systems, 1997, pp. 155–161.