Forecasting Financial Data via Machine Learning Techniques and Time-Series Models

Sijia Huo

Supervisor: Eun Yi Chung

May 2017

Contents

Chapter 1

Introduction

1.1 Motivation

The concept of Machine Learning (or Statistical Learning) was stated by Arthur Samuel in 1959 and refers to "statistical aspects of automated extraction and identification of regularities (structure) in data sets."[1] Along with the arrival of big-data era, the machine learning techniques have been applied more and more into various realms including financial econometrics. Even though according to the theory of efficient capital markets, "in an efficient market, the results from test of return predictability should not reject the null hypothesis of no predictability."[2], some data scientists claim that by adding independent variables which are related to the social sentiments, the accuracy of the forecasting can be improved dramatically.[3] Therefore, our project aim to provide an empirical assessment of this methodology and find different ways that could help to improve the accuracy of forecasting. Even though it is clear that the stock data depends on so many factors that can hardly be predicted accurately, we wish that by adding more independent variables into the time series/machine learning models, the model can provide us more information about the market trend and investment direction.

1.2 the Outline and the Ultimate Goal of the Project

Our project can be divided into three parts. The first two parts discuss the accuracy of statistical models/ machine learning models with only historical stock data as independent variables. To be specific, the first part will discuss the accuracy of different linear and non-linear models that work to predict the future stock prices. The second part will discuss the Markov Models that will not give the exact prediction of prices but will give the trend of stock prices in the future (go higher or lower). We will both discuss the Markov Model and the Hidden Markov Model.The third part discusses whether the accuracy of the prediction can be increased and how large the increasing can be by adding more variables into the model. We plan to add data related to social sentiments from Google and Twitter into our original models to see whether the percentage of accuracy improves. We could also add data from other sources to see which kind of data influences the results most.

For this term paper, we'll only discuss the first part of the outline.We will discuss in which way the "unpredictable" stock price can be better predicted with the historical stock data and the statistical/ machine learning models.

1.2.1 Times-Series Forecasting Techniques

Among the numerous linear and non-linear time-series forecasting models, we pick up five of them to make a further study. Besides, since the distribution of the stock prices is not normal, plus, the volatility of the stock prices clusters according to our observation, we'll also discuss which type of GARCH model can captures this kind of volatility best.

ARIMA Model

The ARIMA model serves as the most popular time-series model to be used in econometrics. The formula for the model can be denoted as $X_t - \alpha_1 X_{t-1} - \ldots - \alpha_p X_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}$.[4] To be specific, the model is a integration of Auto-regressive Model(AR) and Moving Average Model(MA). The model indicates that the target variable for forecasting - X_t depends on the historical data $X_{t-1} \ldots X_{t-p}$ along with the lag-error terms $\varepsilon_t \ldots \varepsilon_{t-q}$. The α and β terms are coefficients. Also, since normally, the continuously compounded (cc) returns displays a mean that is always around zero,we may assume that the model is stationary or week stationary - the previous fluctuation of the stock price will not make a permanent influence on the future stock price.

GARCH Models

Before fitting any GARCH Models, we first need to test the distribution of the data set to guarantee that the data satisfies the requirements of volatility models which were taught in ECON490: (1) The process of the returns is nearly "mds". (2) The volatility is clustering and persistent. So there's a period of high volatility and then a period of low volatility. (3) The autocorrelation is weak (almost zero). (4) The distribution has heavier tail than the normal distribution.

There're multiple types of GARCH Models including the symmetric GARCH model and the asymmetric EGARCH and GJR-GARCH models. The difference between the two types is that the previous one makes no differences between positive positive errors and the negative errors whereas the later two differentiates between positive errors and negative errors and indicates that the sign of the errors makes an influence on the variance of the predictor. Since according to the graphs of cc-return, we can detect a difference of variance between signs, we suspect here that asymmetric model may have a better performance. And the suspection is the motivation for the model comparison.

The formula of the three volatility models are listed below[5]:

- GARCH:
  - $$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2$$
  - $y_t = x_t b + \epsilon_t$
  - The error ϵ_t follows a normal distribution with mean of zero and variance of σ_t^2.
- EGARCH:
  - $g(Z_t) = \theta Z_t + \lambda(|Z_t| - E(|Z_t|))$
  - $$ln\sigma_t^2 = \omega + \sum_{k=1}^{q} \beta_k g(Z_{t-k}) + \sum_{k=1}^{p} \alpha_k ln\sigma_{t-k}^2$$
- GJR GARCH
  - $\sigma_t^2 = K + \delta_{t-1}^2 + \alpha\epsilon_{t-1}^2 - \phi\epsilon_{t-1}^2 I_{t-1}$
  - $I_{t-1} = 0$ if $\epsilon_{t-1} \geq 0$
  - $I_{t-1} = 1$ if $\epsilon_{t-1} \leq 0$

Simple Exponential Smoothing Model

This is one of the simplest model for prediction. The formula for the model is [6]

$$\hat{y}_{T+h|T} = \frac{1}{T}\sum_{t=1}^{T} y_t$$

. The model uses the average of the previous given data as the predicted data for the future. Even though the model is simple, it could perform very well when the data is martingale and when we only devote historical data as the independent variables. This feature has been verified in the later chapter.

Holt-Winters seasonal method

Since the stock data is actually a time-series data with frequency of 5 (five working days per week), we also raise the doubt of whether the stock data follows the seasonal pattern. Therefore, we apply the Holt-Winters Model here, which could be regarded as an upgraded version of the exponential smoothing model. The formula of the model is : $\hat{y}_{t+h|t} = l_t + hb_t + s_{t-m+h_m}$.[7] Here the l_t is similar to the average of the historical data in the exponential smoothing model. In addition, variable b_t denotes the trend of the data and the variable s_{t-m+h_m} is a seasonal dummy variable. Since we assume that the cc-return follows a martingale process,in other words, there's no clear trend among the data, we ignore the hb_t part of the model.

Neural Networks (feed-forward)

This is one of our non-linear machine-learning models. The trick for this model is to add hidden layers between the independent variables and the dependent variable. The layers are a couple of newly created variables which can be transferred into signals by logistic activation functions. We denote the newly created hidden variables as z_j where $(1 \leq j \leq q)$ and

$$z_j = \sum_{i=1}^{p} w_{ij} x_i$$

. Then, we transfer each z_j into a signal via the function $L(z) = \frac{exp(z)}{1+exp(z)}$. Finally, we put the newly created signals and our original independent variables into a non-linear activation function f to give the final prediction of the dependent variable. During the parameter estimation for the model, we iterates the procedure many times to get the optimal estimation. The whole procedure can be expressed by the formula [8]

$$y = f(a + \sum_{i=1}^{p} \phi_i x_i + \sum_{j=1}^{q} \theta_j L(\alpha_j + \sum_{i \to j} w_{ij} x_j))$$

Support Vector Machines (for regression)

This is the second non-linear machine learning model for our prediction. The formula for the estimation is [9]

$$f(x_k, w) = \sum_{i=1}^{D} w_i \phi_i(x_k) + b$$

where the each $w_i \phi_i(x_k)$ denotes one dimension of the "feature"s of the input variables x. We wish to find the proper ws to minimize the risk function

$$R(w) = \frac{1}{N} \sum_{i=1}^{N} |y_i - f(x_i, w)|_\epsilon + \frac{1}{2} ||w||^2$$

. Where y_i corresponds to the observing values for each dimension of input variables . There are multiple ways to build the loss function. Here we choose ϵ-insensitive loss function, which controls the magnitude of errors for our model-building. The formula for this is $|y_i - f(x_i, w)|_\epsilon = 0 \;\; (if \;\; |y_i - f(x_i, w)| < \epsilon)$; $|y_i - f(x_i, w)|_\epsilon = |y_i - f(x_i, w)| \;\; (if \;\; |y_i - f(x_i, w)| \geq \epsilon)$. To find the optimal estimators and coefficients, we take advantage of the kernel function $K(x, y) = exp(-\gamma ||x - y||^2), \gamma > 0$ which is also called "Gaussian radial basis function". We use this kernel function to minimize another formula that Vapnik raised in 1995 that

$$f(x, \alpha, \alpha') = \sum_{i=1}^{D} (\alpha'_i - \alpha_i) K(x, x_i) + b$$

where α_i and α'_i are Lagrange multipliers to extract the best "features" as predictors.

1.2.2 Variable Selection

The variable selection part is the most important part for our project. However, as mentioned before, the first two parts of the project restrict to only applying historical stock data as predictors. To be specific, for the $ARIMA(p, q)$ Model, we applied 75 percent of the total historical data for model-fitting, use the computer to auto-select the best p and q for the model. For the $GARCHFamily$ models, we simply take the one day-ahead data as the predictors. That's to say, we set p and q as both 1 when doing the model comparison. For the Exponential Smoothing Model and the Holt-Winters seasonal method, we applied the earliest 75 percent of the historical data to fit the model and to find the seasonal/average parameters. For the two machine learning models, we use the previous five-day data for each target prediction as the variables.

1.2.3 Accuracy Assessment

For all the models except the GARCH-Family Models, we use the following three index as the indicators of the accuracy:

- Root-Mean-Square Error (RMSE)

  $RMSE = \sqrt{E((\hat{y_t} - y_t)^2)}$

- Mean-Absolute Error (MAE)

  $MAE = E(|(\hat{y_t} - y_t)|)$

- Normalised-Mean-Square Error (NRMSE)

  $NRMSE = \frac{RMSE}{SD(y_t)}$

For the GARCH-Family Models, since they only predict the volatilises of the future data rather than the exact values, we cannot calculate forecasting errors like other models. However, there are other ways to assess the models. Here, we will use AIC, BIC, and ML as such index. For the AIC and BIC, we will pick us the model with the smallest value. Normally, BIC has a larger penalty for the number of estimators than AIC and is asymptotic, it is more reliable. For the ML (maximum likelihood), we will choose the one with the largest value.

Chapter 2

Code Explanation

2.1 Data Source and Data Cleaning

We collected data from the Yahoo Finance. Since we are both interested in the predictability of the daily continuously compounded (cc) returns and that of the monthly cc-returns, we separately collect these two parts of the data.

For the daily stock price, we used the "getSymbols" function from the "quantmod" package in R. We picked up the time period from "2010-05-12" to "2017-05-12" for our project. For the monthly stock price, we directly download the data from the Yahoo Finance webpage. Since the sample size of the monthly return data is significantly smaller than that of the daily data, we choose a longer time period for our study: from June 1990 to May 2017.

We choose two companies for our analysis : Microsoft and McDonalds to avoid bias. In the future, we may expand our study to more companies in other realms.

Our current study focus on the adjusted close price of the stock, therefore, we only picked up this part from the data set. The next step is to change the adjusted close price into continuously compounded returns. The formula for this transition is $r_t = ln(\frac{P_t}{P_{t-1}})$.

Before fitting models, we split the dataset into two parts. We take the first 75 percent of data as training data and the last 25 percent of the data as testing data. We fit models with the training data and compare the predicted value with the testing data to assess the effectiveness of the models.

2.2 Parameter Selection

2.2.1 ARIMA

We take the advantage of function $auto.arima()$ to choose the best value of p and q for the model. To increase the effectiveness of the model, we add a Fourier term to capture the potential periodicity. However, since this kind of periodicity is trivial, we also fit the same model with no Fourier terms. Then, we take use the $forecast()$ function in the same package to make predictions.

2.2.2 GARCH Family

For the garch family models, we apply the $ugarchspec()$ function to specify which kind of GARCH model we want to fit. We set the arma order to $(0, 0)$ here because of the weak auto-correlation in the original data and also because this can simplify our model-fitting procedure. We set the garch order to $(1, 1)$ because as mentioned before, we only use the one-day ahead data as predictors for the simplicity.

2.2.3 Holt-Winters

We use the function $HoltWinters()$ to fit both the Simple Exponential Smoothing Model and the Holt-Winters Seasonal Model. For the former one, we set the parameters γ and β to 0 whereas for the later one, we only set β to 0 to avoid adding trends to the models.

2.2.4 Neural Networks

We use the function $nnet()$ to fit the model and make predictions. We add only one layer to our neural network with six hidden nodes. We set "skip" term to TRUE to add skip-layer connections from input to output. We iterates this fitting procedure for 10^4 times to get the optimal results. Also, we make a penalty on the large weights w to avoid over-fitting. We do this by setting the $decay$ parameter to 10^{-2}. We finally wish that the outputs are in linear unit rather the logistic unit, so we set the parameter $linout$ to TRUE.

2.2.5 Support Vector Machines

As mentioned in the earlier chapter, we use the ϵ-insensitive loss function to restrict the magnitude of errors, so we set $type$ parameter to $"eps-regression"$. We use Gaussian radial basis function as our kernel function and set γ to 10^{-2} for the function. Also, we set $cost$ to $10^{4.5}$ to avoid over-fitting.

Chapter 3

Results

3.1 Tables

Table 3.1: P-values for Jarque-Bera Tests

Monthly MSFT	Daily MSFT	Monthly MCD	Daily MCD
3e-15	<2e-16	6e-15	<2e-16

Table 3.2: GARCH Family Index for MSFT Daily Return

	AIC	BIC	ML
GARCH	-5.6097	-5.5948	3931
E-GARCH	-5.7912	-5.7687	4060
GJR-GARCH	-5.7856	-5.7632	4056

Table 3.3: GARCH Family Index for MSFT Monthly Return

	AIC	BIC	ML
GARCH	-1.9440	-1.8876	247
E-GARCH	-1.9496	-1.8650	249.7
GJR-GARCH	-1.9557	-1.8712	250.5

Table 3.4: MSFT Daily Return Forecast Accuracy

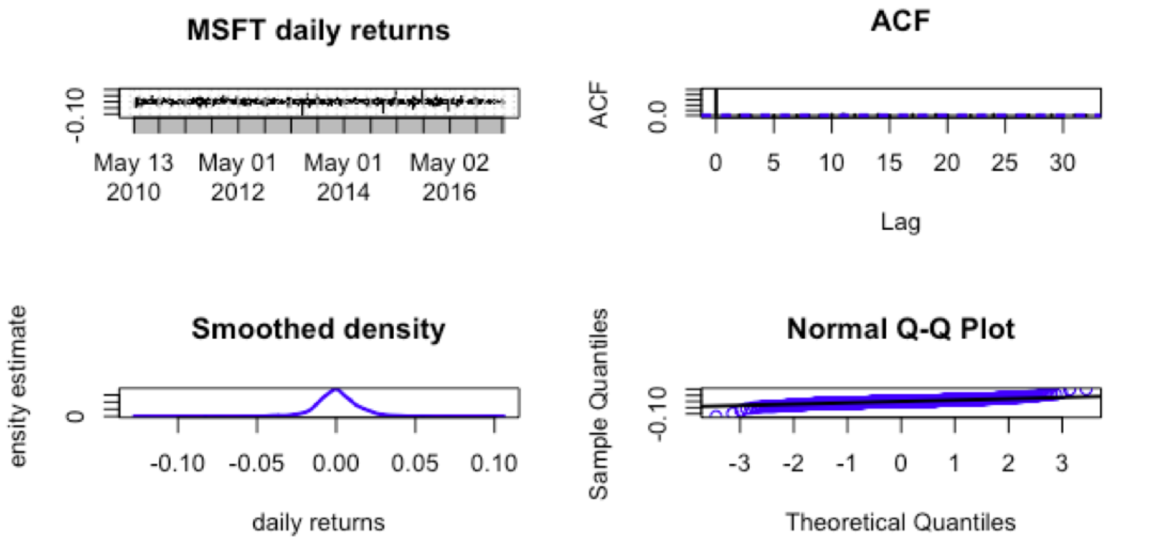
	SVM	NNET	Exponential Smoothing	Holt-Winters	ARIMA-Fourier	ARIMA
RMSE	0.01831	0.01526	0.01289	0.01531	0.01292	0.01284
MAE	0.012166	0.010826	0.008911	0.011526	0.008848	0.008786
NRMSE	120.4	100.3	100.3	119.1	100.5	99.9

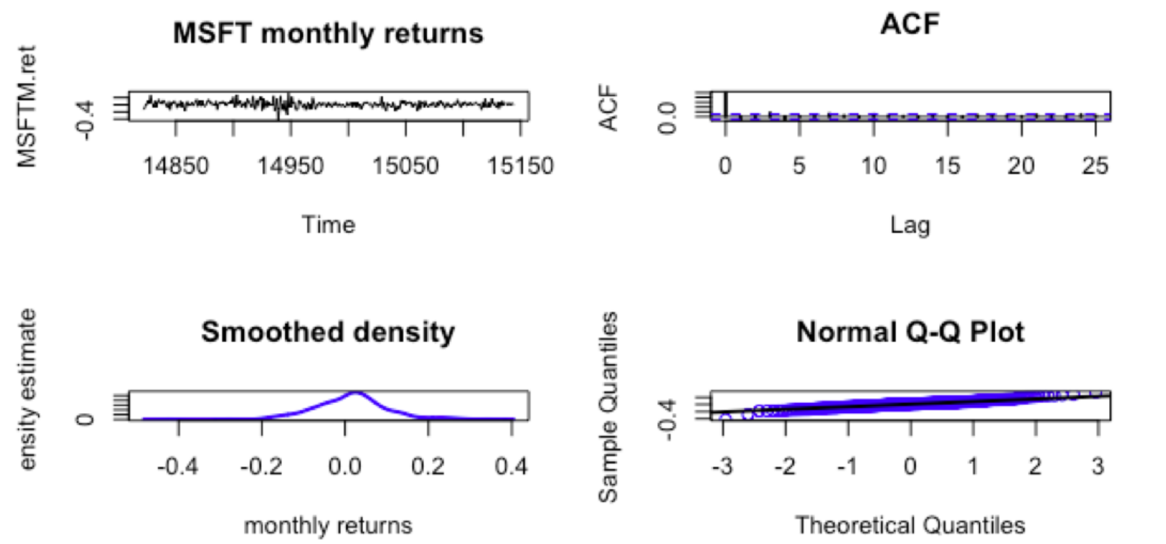
Table 3.5: MCD Daily Return Forecast Accuracy

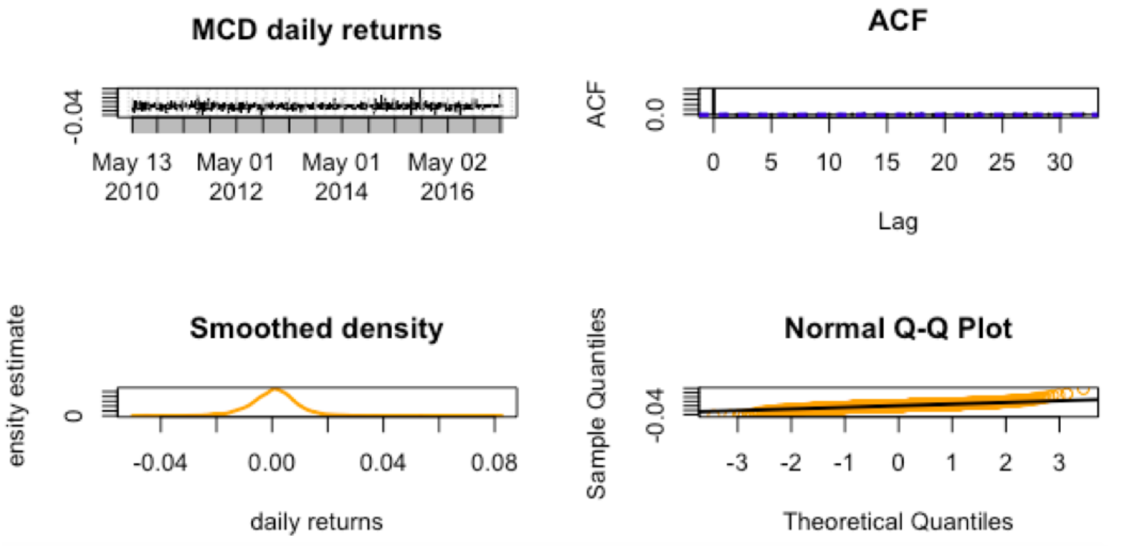
	SVM	NNET	Exponential Smoothing	Holt-Winters	ARIMA-Fourier	ARIMA
RMSE	0.015388	0.009685	0.009021	0.010264	0.009055	0.009018
MAE	0.008286	0.006754	0.006212	0.007250	0.006294	0.006237
NRMSE	158.9	100.0	100.0	113.7	100.3	99.9

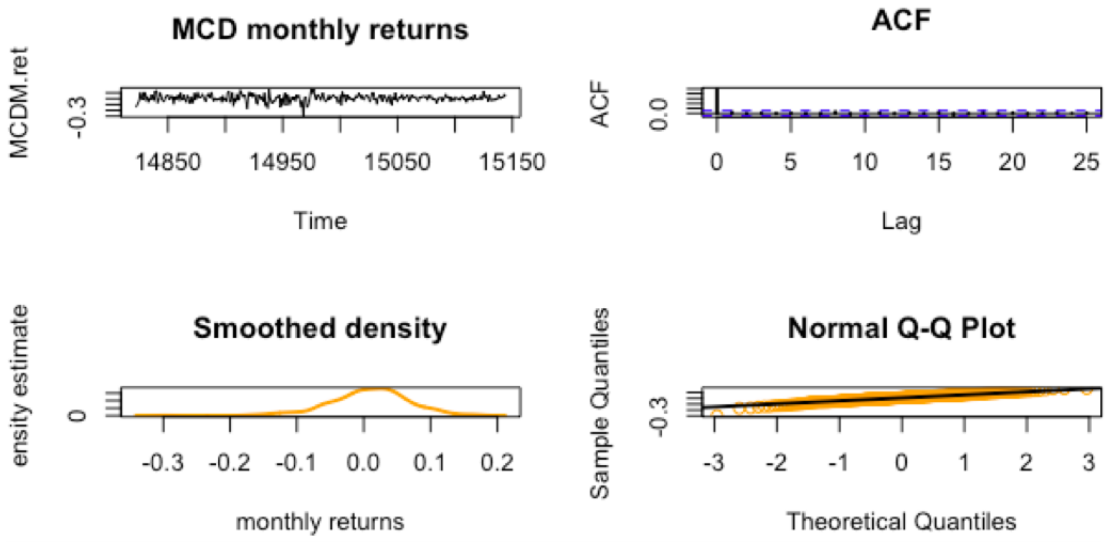
3.2 Graphs

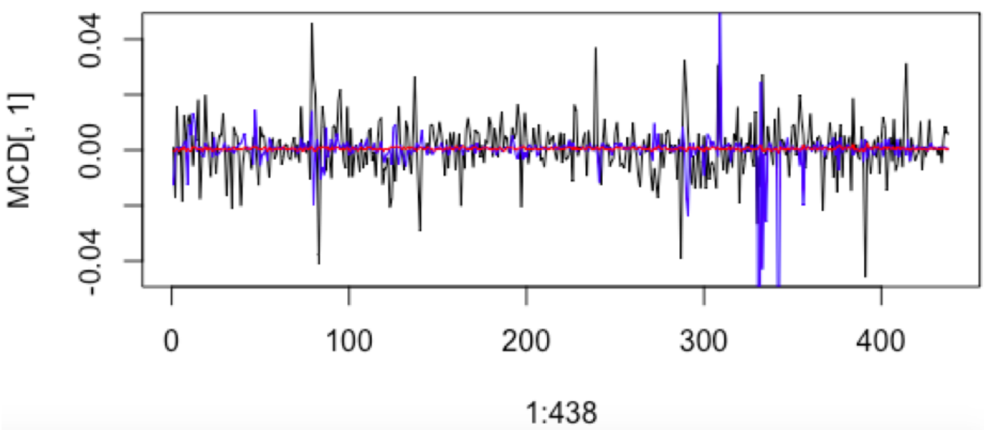
The first four graphs display the distribution of cc-return data. The fifth and seventh graph is the forecasting VS actual data via non-linear machine-learning models. The fifth is for McDonald's and the seventh is for Microsoft. The sixth and the eighth graph are the comparison for linear models. The former one is for McDonald's and the later one is for Microsoft.

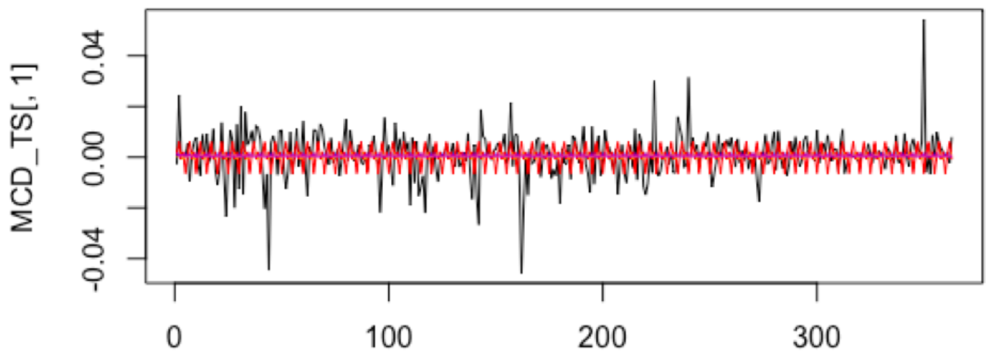


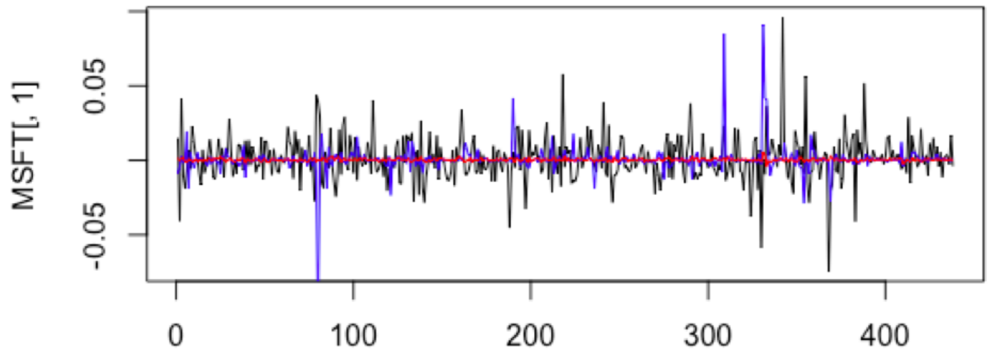


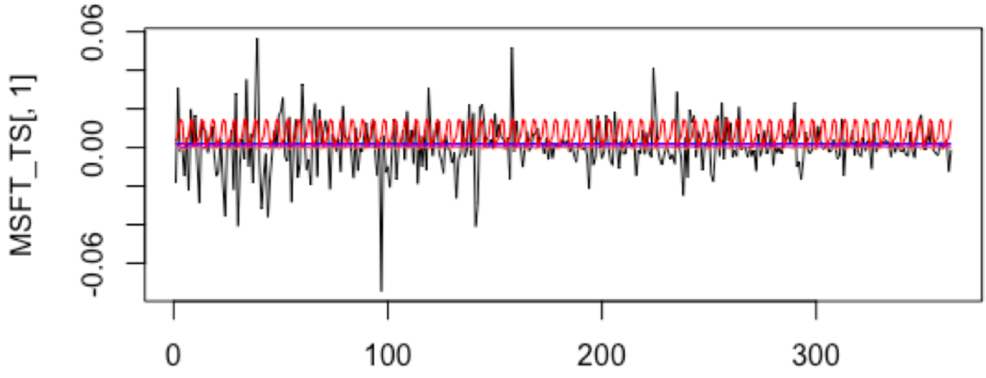

MCD daily returns
-0.04
May 13 2010
May 01 2012
May 01 2014
May 02 2016
ACF
ACF
0.0
0
5
10
15
20
25
30
Lag
Smoothed density
ensity estimate
0
-0.04
0.00
0.04
0.08
daily returns
Normal Q-Q Plot
Sample Quantiles
-0.04
-3
-2
-1
0
1
2
3
Theoretical Quantiles


MCD monthly returns
MCDM.ret
-0.3
14850
14950
15050
15150
Time
ACF
ACF
0.0
0
5
10
15
20
25
Lag
Smoothed density
ensity estimate
0
-0.3
-0.2
-0.1
0.0
0.1
0.2
monthly returns
Normal Q-Q Plot
Sample Quantiles
-0.3
-3
-2
-1
0
1
2
3
Theoretical Quantiles


MCD[, 1]
0.04
0.00
-0.04
0
100
200
300
400
1:438


MCD_TS[, 1]
-0.04
0.00
0.04
0
100
200
300


MSFT[, 1]
-0.05
0.05
0
100
200
300
400


MSFT_TS[, 1]
-0.06
0.00
0.06
0
100
200
300

Chapter 4

Conclusions and Future Works

4.1 Conclusions

According to the tables and graphs provided, we could conclude that the simple ARIMA model with no Fourier Terms performs best among all models. However, the model restricts its predictors to only historical data. The exponential smoothing model also performs quite well even though it is the simplest among others. However, it also restricts its predictors. We think that the neural network model also has a good performance. And it has the advantage of adding more variables into the model for the prediction. Therefore, in the future, we will work more on this non-linear model to see whether it can have an impressive performance after adding more variables into the model. Besides, we can also conclude that the cc-return data embeds no periodicity.

For the GARCH-Family models, we can first claim that none of the data we collected follows the normal distribution. So it qualifies the requirements of GARCH-family models. For the daily return data, the E-GARCH model works better, but for the monthly return data, the three GARCH-family Models work equivalently well.

4.2 Future Works

Like what is mentioned in the first chapter, the next step for our project is to fit Markov models for the cc-return data. After that, we will try to add variables including social sentiments to our model to see how much the prediction can be improved.

Bibliography

[1]Maasoumi, Esfandiar and Marcelo C. Medeiros. "The Link between Statistical Learning Theory and Econometrics: Applications in Economics, Finance, and Marketing." Econometric Reviews, vol. 29, no. 5/6, Sep-Dec2010, pp. 470-475. EBSCOhost, doi:10.1080/07474938.2010.481544.

[2]Milionis, Alexandros E. "Efficient Capital Markets: A Statistical Definition and Comments." Statistics Probability Letters, vol. 77, no. 6, 15 Mar. 2007, pp. 607-613. EBSCOhost, doi:10.1016/j.spl.2006.09.007.

[3]ARIAS, MARTA, et al. "Forecasting with Twitter Data." ACM Transactions on Intelligent Systems Technology, vol. 5, no. 1, Mar. 2014, pp. 1-24. EBSCOhost, doi:10.1145/2542182.2542190.

[4]Mladenović, Jelena, et al. "Modelling and Prognosis of the Export of the Republic of Serbia by Using Seasonal Holt-Winters and Arima Method." ["MODELIRANJE I PROGNOZIRANJE IZVOZA REPUBLIKE SRBIJE PRIMENOM SEZONSKOG HOLT-WINTERSOVOG I ARIMA METODA"]. Economic Themes, vol. 54, no. 2, Apr. 2016, pp. 233-260. EBSCOhost, search.ebscohost.com/login.aspx?direct=truedb=bthAN=116851007site=ehost-live.

[5]SARANGI, PRASANT, et al. "Forecasting Capital Market Volatility: A Comparison among GARCH Families of Models." Finance India, vol. 29, no. 2, June 2015, pp. 495-513. EBSCOhost, search.ebscohost.com/login.aspx?direct=truedb=bthAN=111493203site=ehost-live.

[6]Maria, Fat Codruta and Dezsi Eva. "Exchange-Rates Forecasting: Exponential Smoothing Techniques and Arima Models." Annals of the University of Oradea, Economic Science Series, vol. 20, no. 1, June 2011, pp. 499-508. EBSCOhost, search.ebscohost.com/login.aspx?direct=truedb=bthAN= live.

[7]Chatfield, C. "The Holt-Winters Forecasting Procedure." Journal of the Royal Statistical Society: Series C (Applied Statistics), vol. 27, no. 3, Nov. 1978, p. 264. EBSCOhost, search.ebscohost.com/login.aspx?direct=truedb=bthAN=5926721site=ehost-live.

[8]Bahrammirzaee, Arash. "A Comparative Survey of Artificial Intelligence Applications in Finance: Artificial Neural Networks, Expert System and Hybrid Intelligent Systems." Neural Computing Applications, vol. 19, no. 8, Nov. 2010, pp. 1165-1195. EBSCOhost, doi:10.1007/s00521-010-0362-z.

[9]Shen, Chuanhe, et al. "A Reduced Pricing Model for Mezzanine Financing Based on Options and Support Vector Machines." Cluster Computing, vol. 19, no. 4, Dec. 2016, pp. 2169-2177. EBSCOhost, doi:10.1007/s10586-016-0671-9.