Course Project Presentation

Foundations of machine learning - CS725

Submitted by:
ANAND NAMDEV(163050068)
PARTH LATHIYA(163050095)
AWISHA MAKWANA(163050079)

Stock Price Prediction

Problem Definition and objective

- Stock market imp for both Industry & Investor
- The best way for the company to raise fund & expand by issuing shares which customers buy
- To predict the stock price and compare the closing price
- Compare and analyze different models
- Tune parameters to achieve maximum accuracy

Motivation

- Stock prices : one of the time series data
- Time series one of the most unpredictable data series domain
- Output value independent of input feature
- Time is the only feature in such problems
- Learn and study ML and Statistical Models

Methods

- Statistical Model
  - ARIMA

- Machine learning Model
  - Artificial Neural Networks (ANN)
  - Support Vector Classification (SVC)

Data Description

- Nikkei-225 official stock of Japan
- Data consists of daily opening and closing price
- Data Sets types used
    - Original data (type-I)
    - Logarithmic data (type-II)
    - Moving average logarithmic data (type-III)
    - Exponential moving average logarithmic data (type-IV)
    - Decomposed data (type-V)

ARIMA

- Auto Regressive Integrated Moving Average
- Consists of 3 parts : AR | I | MA
- AR : Previous p-days regression
- I : how many times data has been differenced
- (p>=1) ⇨ AR model & (q>=1) ⇨ MA model
- i=0 ⇨ ARMA model , No differencing over the data

$$Y_t = \mu + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q}$$

Y(t) : predicted value

ϕ : previous p-days data coefficients

θ : previous q-days error coefficients

Data & Requirement & Fixes

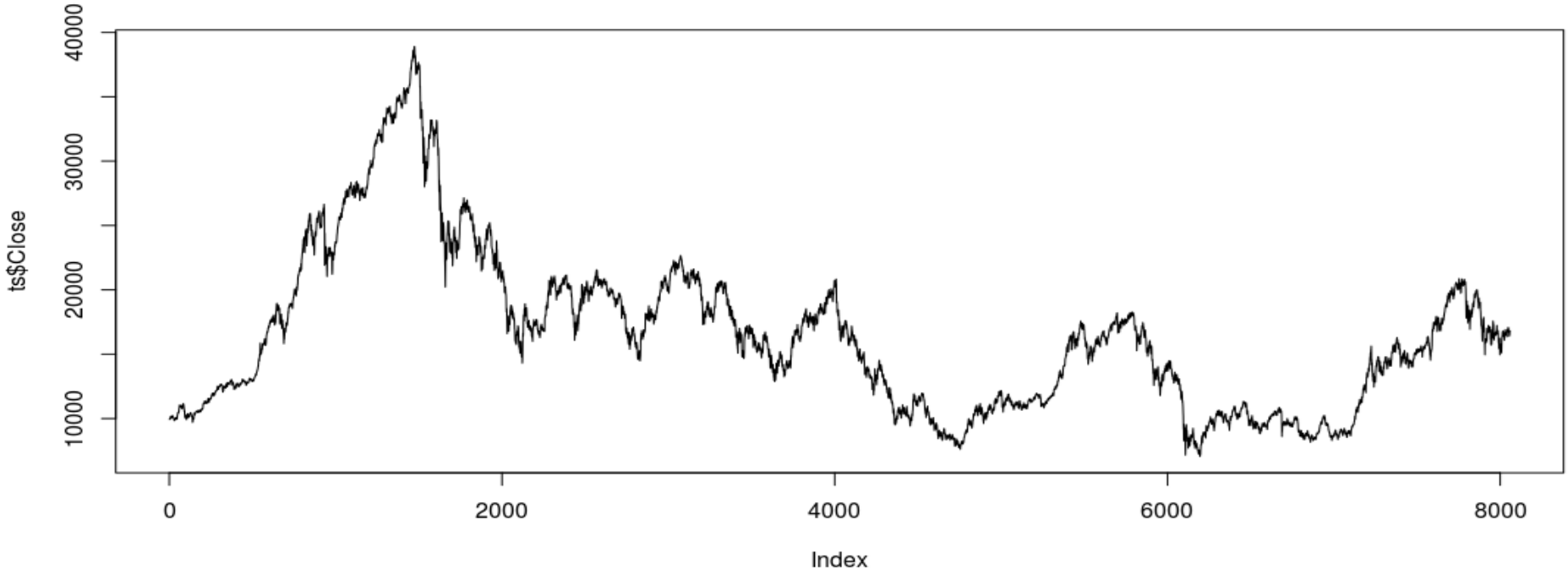
- Experiment performed on Nikkei-225 Japan Stock
- ARIMA gives best performance for stationary data
- If data possess non-zero mean & nonzero variance then data : non-stationary
- Make data stationary by : a) differencing b) decomposing
- Differencing ⇨ i!=0 : only for ARIMA
- Applied Dickey-Fuller test on each to check if data is stationary or non-stationary

Mean & Variance across different scaled data

Mean and Variance across different dataset

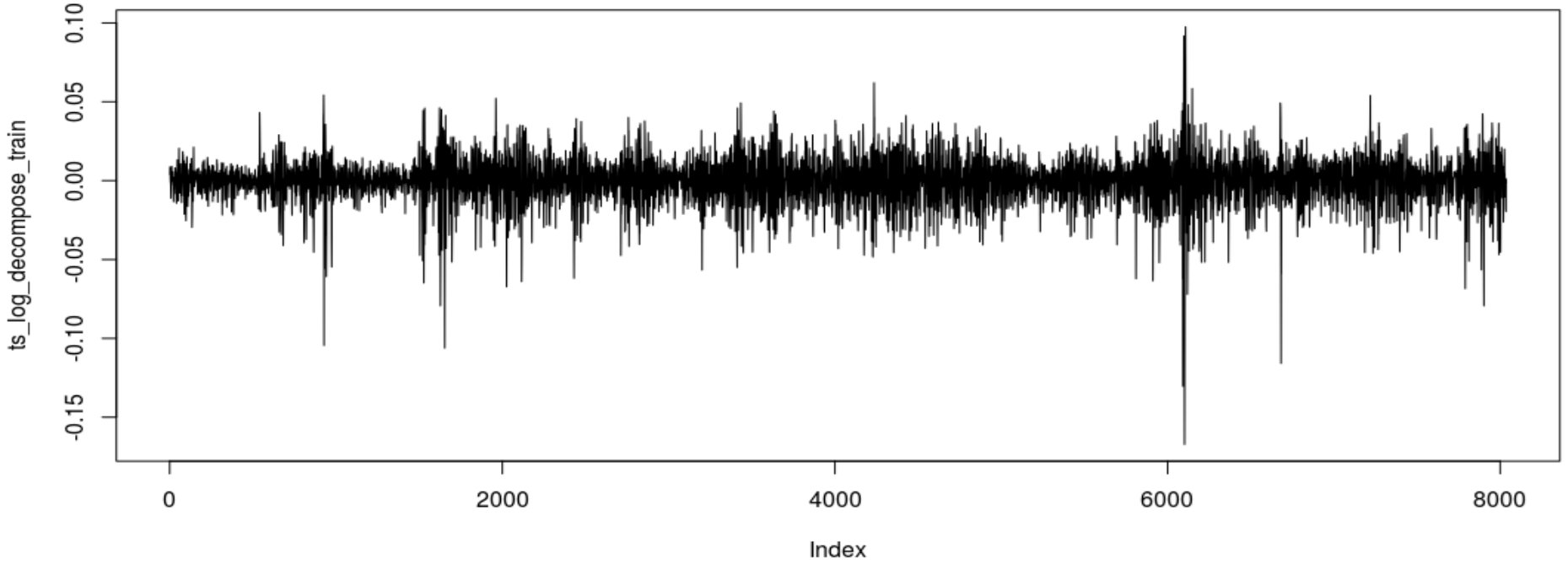
SNo.	TYPE	MEAN	VARIANCE
1	Original	1.649584	38618370
2	Logarithm	9.645	0.1292
3	Moving Average	0.0002911	0.0005400021
4	Exponentiating	0.001053	0.0015610
5	Decomposing	$-1.55e^{-06}$	0.0002067

Original Data : mean = 16495 ; variance = 38618370

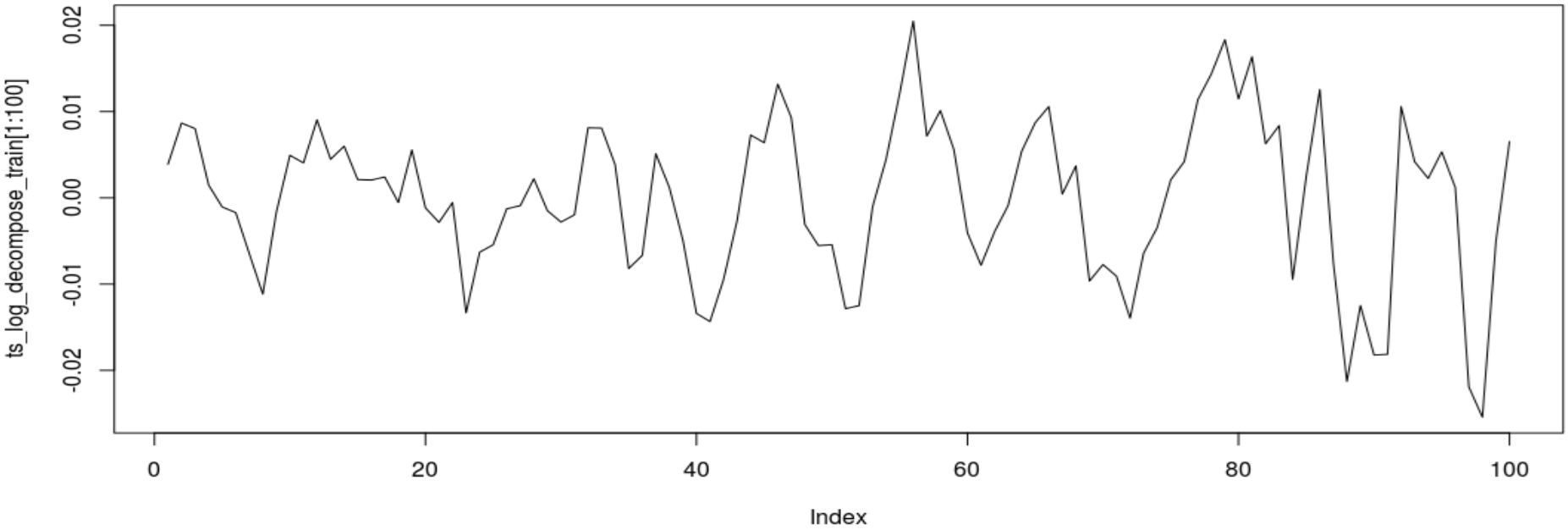


Decomposed Data : mean = 1.55e-06 ; varaince : 0.0002067

Complete data : 8050 rows



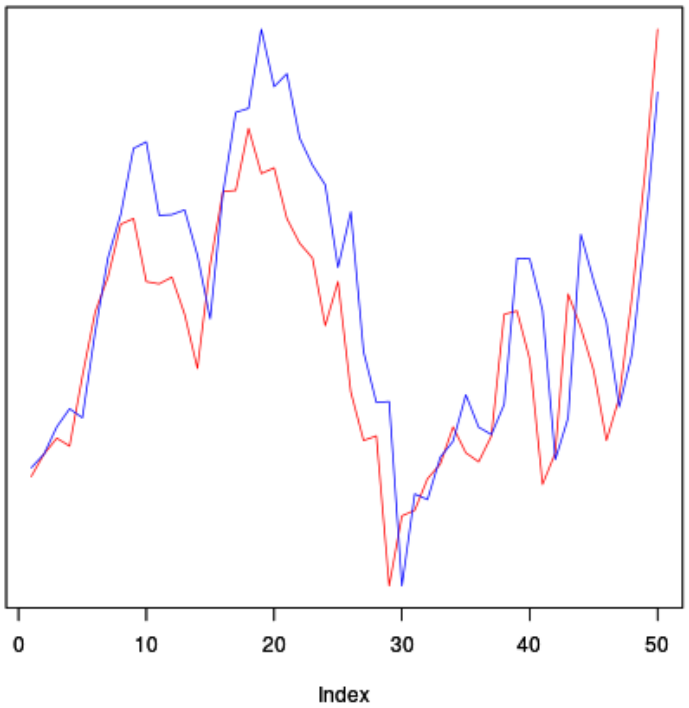
Decomposed data : Shorter data : 50 rows



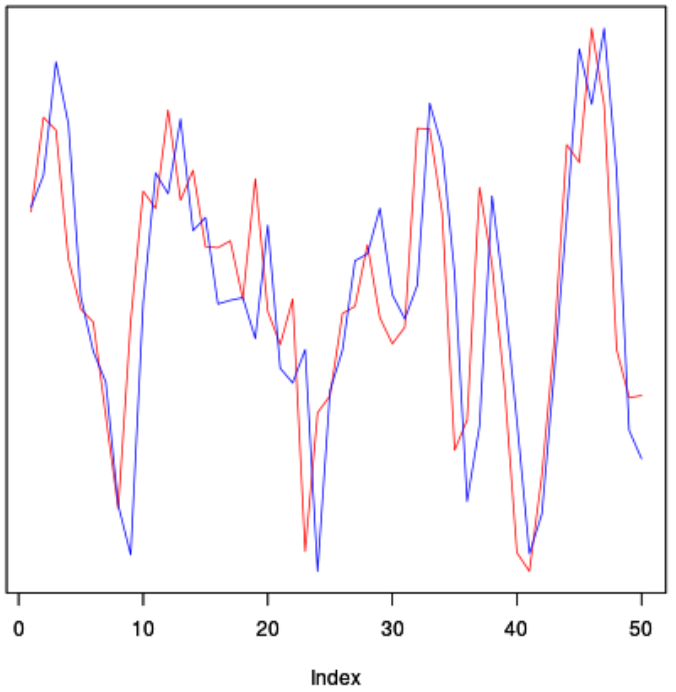
Training

- For each type of data, found out the correct value of p&q by tuning the parameters over the accuracy AIC/BIC value
- Also tuned I in case of ARIMA, for ARMA I=0
  - Original data (p=0;i=1;q=2) | (p=0;i=0;q=2)
  - Logarithmic data (p=0;i=1;q=0) | (p=1;i=0;q=0)
  - Moving average logarithmic data (p=2;i=1;q=4) | (p=4;i=0;q=2)
  - Exponential moving average log data (p=1;i=1;q=1) | (p=0;i=0;q=2)
  - Decomposed data (p=2;i=1;q=2) | (p=3;i=0;q=4)

Type - I data



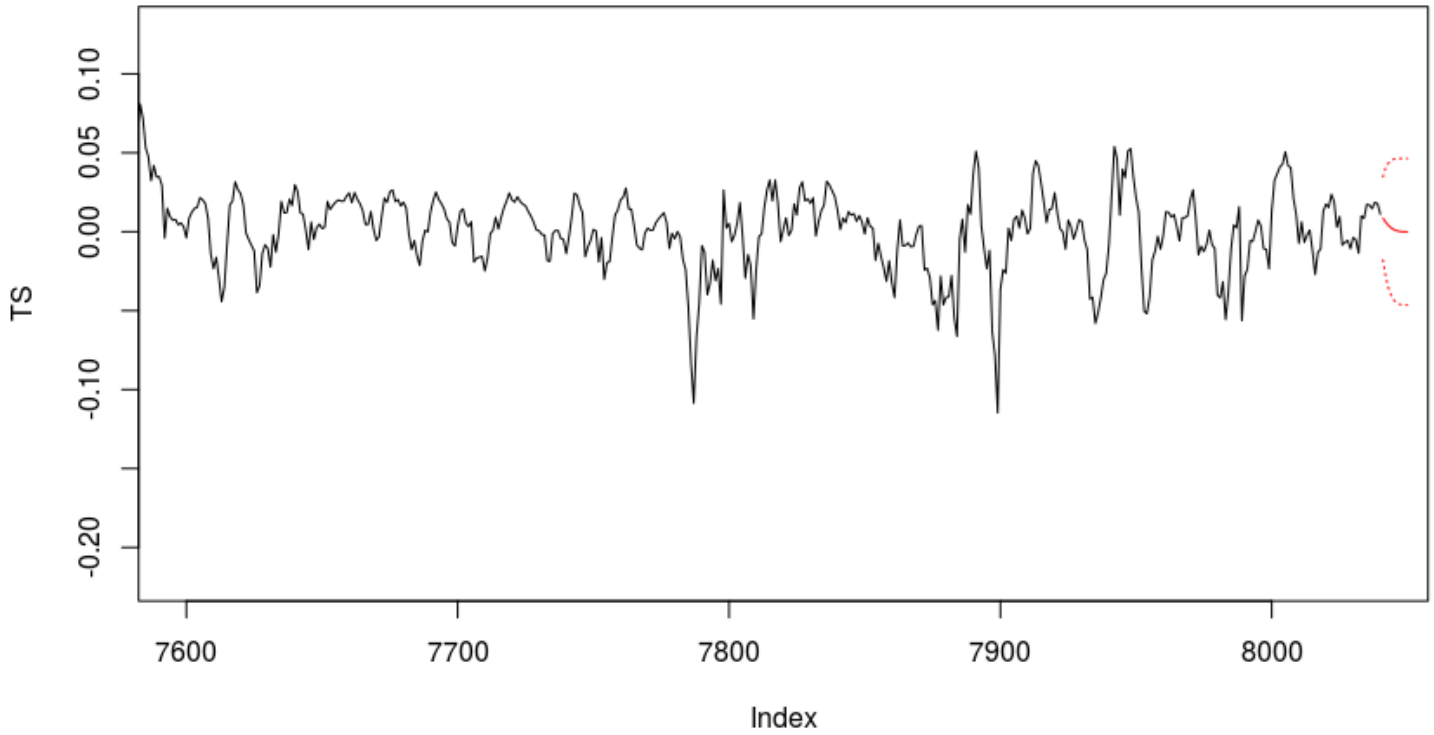
Type - V data



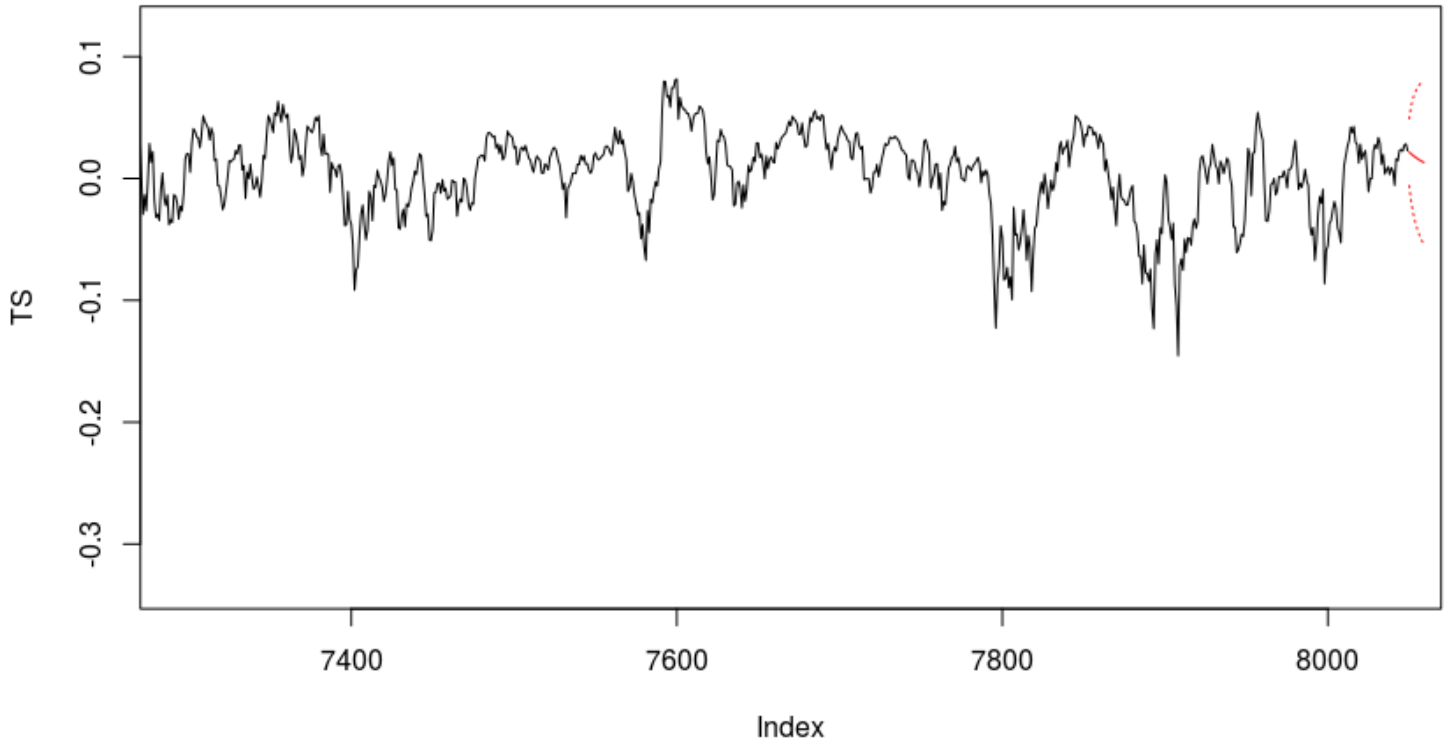
Prediction

- Consists of predicting 10-days ahead values
- Did prediction for each of the 5 models
- Analysed that original data failed to correctly predict ahead data because of high Non-stationarity (for both ARIMA/ARMA)
- Provided upper(95%) and lower(80%) bound for prediction
- Prediction is done for ARIMA & ARMA(i=0)

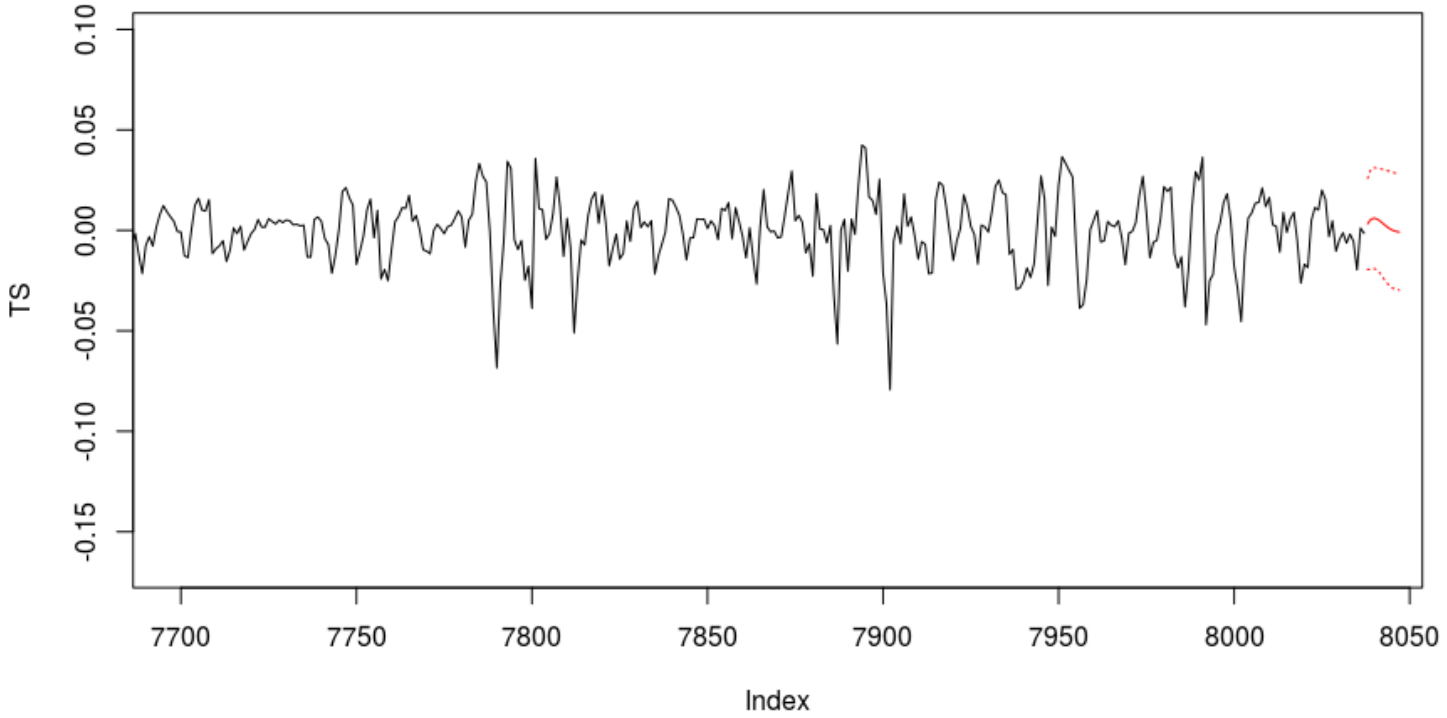
Data : Moving Average - 10day ahead



Data : Moving Average - 10 days ahead



Decomposed Data :



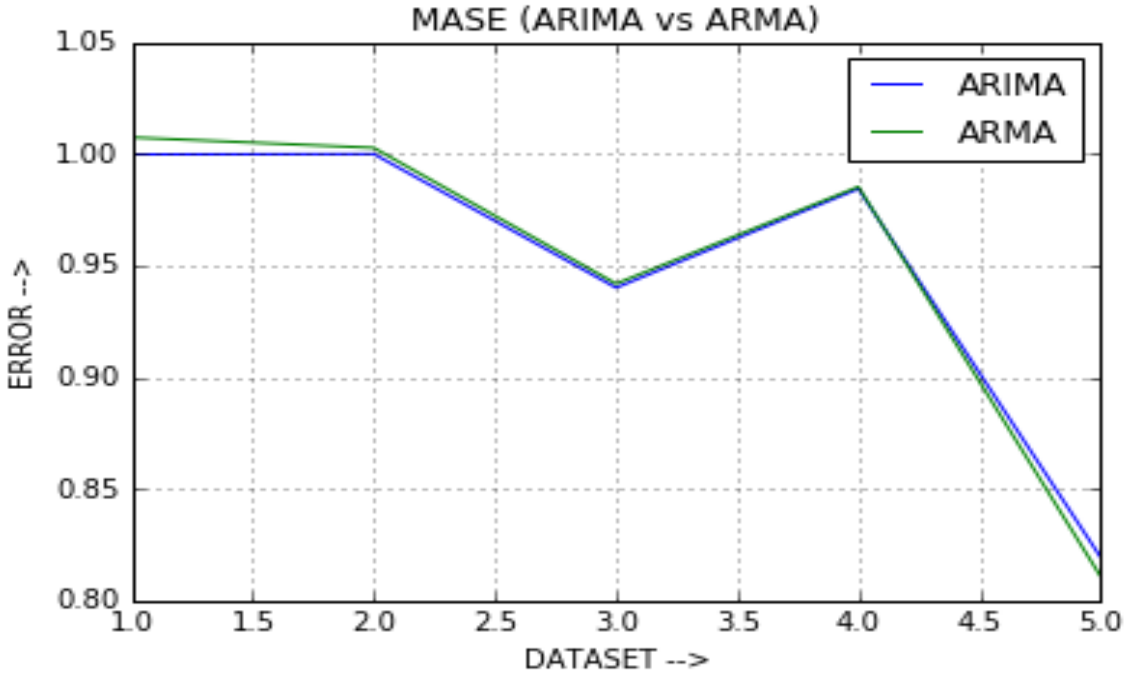
Data Type - Error :

MASE: To compare forecast accuracy across series of different scales.

$$q_j = \frac{e_j}{\frac{1}{T-m}\sum_{t=2}^{T}|y_t - y_{t-m}|}$$

S. No	ARIMA (I!=0)	ARMA(i=0)
1	0.99968	1.0073
2	0.99987	1.00263
3	0.94	0.9418
4	0.9843	0.9852
5	0.8193	0.8107

Error Comparison of ARIma & ARMA :

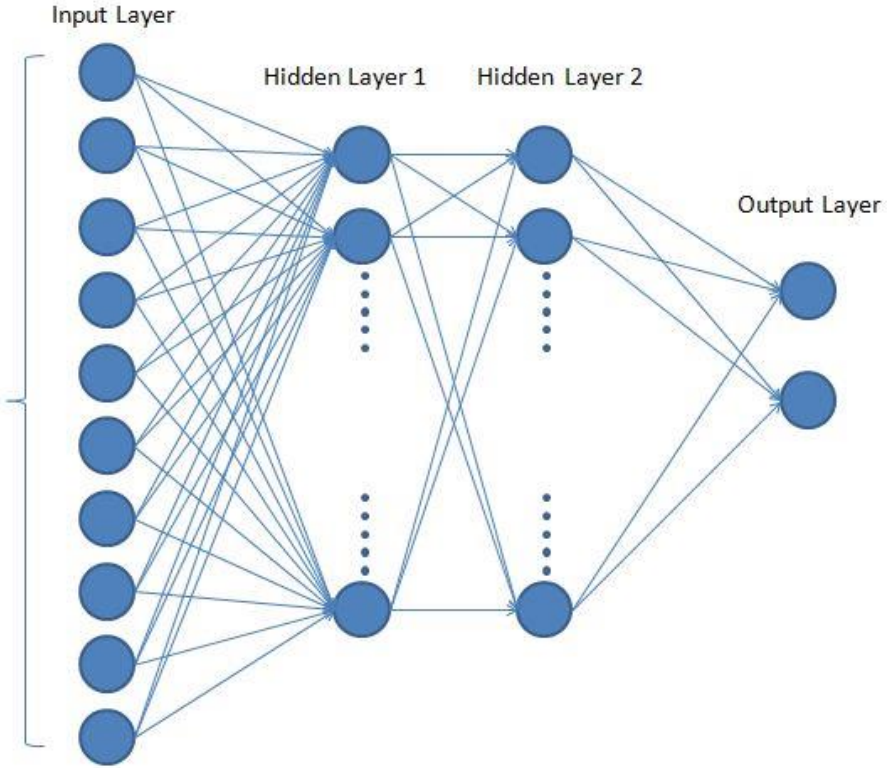


ANN

- Artificial Neural Network
- Machine Learning Model
- Relatively crude electronic models based on neural structure of the brain.
- Input Layer, hidden layers, output layer
- An ANN architecture with two hidden layers is as follows

ANN ARCHITECTURE

Processed Input



Data PrePROCESSING

- Experiment performed on Nikkie-225 Japan Stock
- The data has 4 columns opening amount, closing amount, high and low.
- 10 formulations for input preprocessing.
- 10 neurons in input layer computed as follows

S No.	NAME	FORMULA
1	Simple 10-day moving average	$\frac{C_t + C_{t-1} \cdots + C_{t-10}}{10}$
2	Weighted 10-day moving average	$\frac{((n) \times C_t + (n-1)C_{t-1} \cdots + C_{10})}{(n-(n-1) \cdots +1)}$
3	Momentum	$C_t - C_{t-n}$
4	Stochastic K%	$\frac{(C_t - LL_{t-n})}{(HH_{t-n} - LL_{t-n})} \times 100$
	Stochastic D%	$\sum_{i=0}^{n-1} K_{t-i}\%$
6	RSI (Relative Strength Index)	$100 - \frac{100}{1 + (\sum_{i=0}^{n-1} Up_{t-i}/n)/(\sum Dw_{t-i}/n)}$
7	MACD (moving average convergence divergence)	$MACD(n)_{t-1} + 2/n+1 \times (DIFF_t - MACD(n)_{t-1})$
8	Larry William's R%	$\frac{H_n - C_t}{H_n - L_n} \times 100$
9	A/D (Accumulation/Distribution) Oscillator	$\frac{H_t - C_{t-1}}{H_t - L_t}$
10	CCI (Commodity Channel Index)	$\frac{M_t - SM_t}{0.015D_t}$

HYPERPARAMETER TUNING

- The number of neurons(n) in the hidden layer, value of learning rate(lr) and number of iterations(epochs) are ANN model hyper parameters that must be efficiently determined.
- Eight levels of n, 10 levels of learning rate and ten levels of epochs were tested in the hyper parameter tuning.

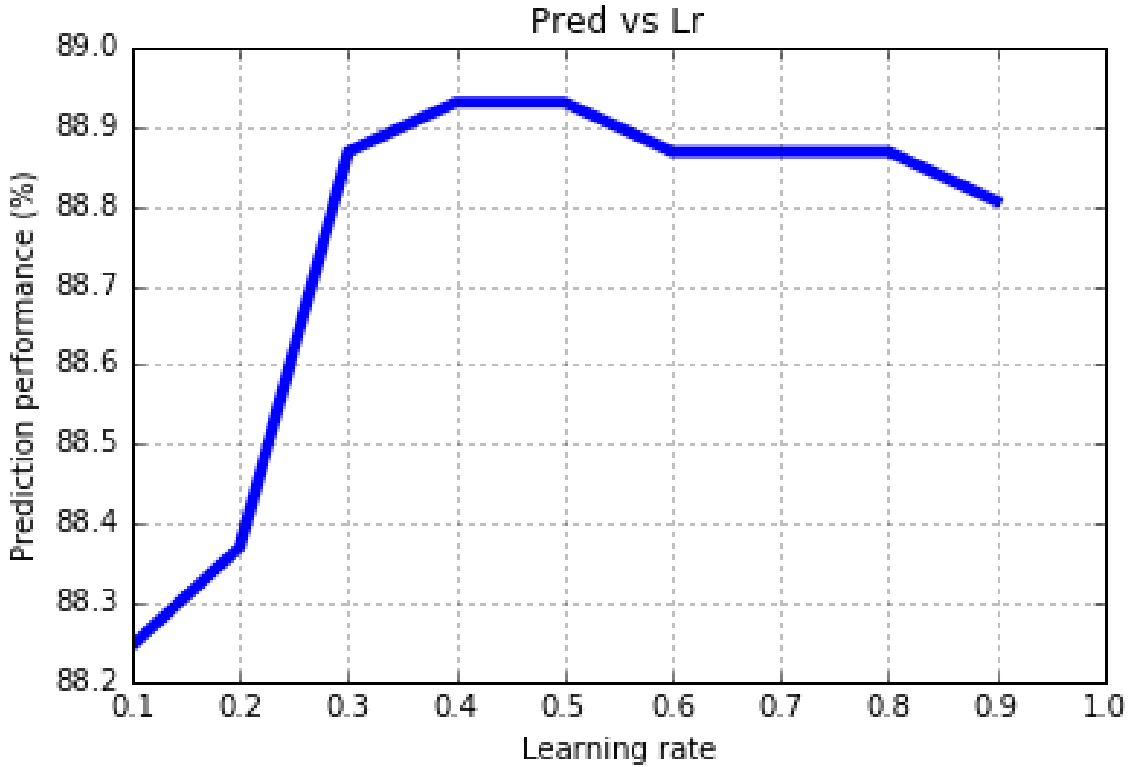
Parameter	Levels
Number of neurons in hidden layer	10,15,20.....,40
Value of learning rate	0.1,0.2,.....,0.9
Number of iterations(epochs)	10,20,.....,100

Observed Accuracies

- Some observations while tuning:

Learning Rate	Epochs	No. of neurons	Training	Testing
0.1	10	30	0.9113	0.8818
0.1	10	12	0.9134	0.8868
0.1	50	30	0.9141	0.8849
0.9	20	28	0.9146	0.8905

Prediction performance vs learning rate



SUPPORT VECTOR CLASSIFICATION

- constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space
- Good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class
- In general the larger the margin the lower the generalization error of the classifier.
- Kernel functions adopted in experiments:
  - Polynomial
  - Radial basis functions(RBF)

Parameter setting experiment

- Different levels of the degree of polynomial function (d), gamma constant of radial basis function (c) and regularization parameter (c) were tested in the parameter setting experiments. The SVM parameters and their levels are summarized in below table.

Parameters	Levels (polynomial)	Levels (radial basis)
Gamma in kernel function (c)	0, 0.1, 0.2, … ,5.0	0, 0.1, 0.2, … ,5.0
Regularization parameter (c)	1,10,100	1,10,100

Observation

- The data sets were applied to the SVM models with three different parameter combinations and the results are given in above table.

No	kernel function	d	γ	C	Training	Testing	Average
1	RBF	-	2.5	100	0.9125	0.9038	0.9081
2	RBF	-	5.0	100	0.9125	0.9001	0.9063
3	RBF	-	3.1	100	0.9125	0.9041	0.9083
4	Linear	-	-	100	0.9036	0.8992	0.9014
5	Polynomial	1	3.5	100	0.9003	0.8982	0.8992
6	Polynomial	1	0.3	100	0.9033	0.9032	0.9032
7	Polynomial	1	0.5	100	0.9064	0.9002	0.9033

Advantages & Disadvantages of ARIMA, ANN & SVC

- ARIMA gives one of the most accurate prediction results based on previous data
- However, the accuracy is often achieved by varying error for each coefficients.
- ANN can fit almost fit any kind of nonlinearity in data by varying tuning parameters.
- However, to fit such a nonlinearity neural networks take huge time in training and testing.
- Consists of regularization parameter to avoid overfitting
- SVC suffers from the problem of choice of kernel function while tuning parameters and also the testing time.

THANK YOU