Model Based Signal Analysis Project

Oil Price Estimation

Detailed Analysis and Estmation of Oil Prices(Time Series Data) using different models involving MLE and Bayesian Estimating methods

Group: 04
Members:
Subash J - S20190020253
Prudhvi Raj Chitta - S20190020207
Venkata Siva Tanari - S20190020255

1 Project Description

The markets are undoubtedly one of the most fascinating things one can get involved in. They can be considered an embodiment of the entire world. By observing the markets you learn everything about people. The main problem that tends to appear with markets, is that they are often unpredictable. The mathematical models might all show that the value of a certain commodity will go up and then something unpredictable happens and everything changes. Thus, it becomes obvious that the markets are extremely prone to external influence and factors. Nevertheless, we are going to attempt, by using various time series estimators, to predict the price of oil in various time stamps.

2 Description of the Data

The data set includes basic fields required in signal such as Date, Open, High, Low, Close,Volume and OpenInt. Prices have been adjusted for dividends and splits. Volume is the total trading volume on that particular day of oil stock

2.1 Data Set Information:

The data set contains a total of 7 columns (Date, Open, High, Low, Close, Volume and OpenInt) with 4661 data points. (There were no missing values in the data set). All attributes except Date are numerical attributes.

Attribute Information:

No.	Attribute	Description
1	Date	Date range from 1999 to 2017
2	Open	Opening price on that day.
3	High	Maximum price on that day
4	Low	Minimum price on that day
5	Close	Closing price on that day
6	Volume	Trading volume of Oil stock

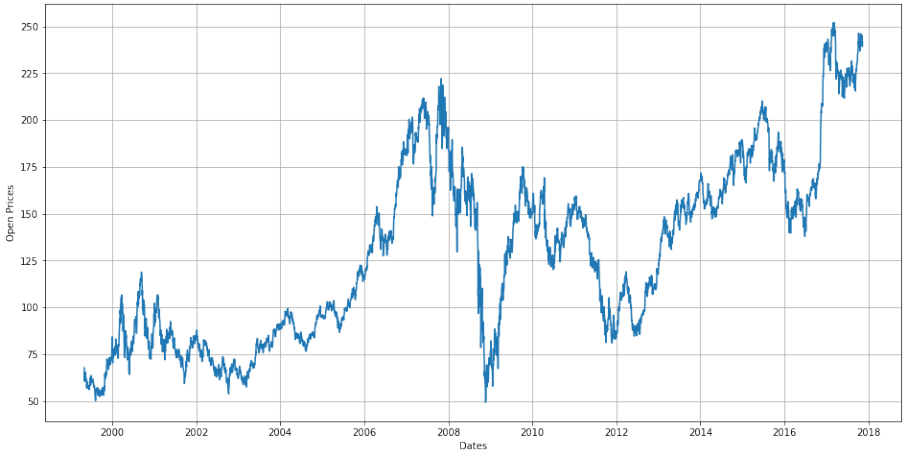
3 Methodology

3.1 Visualizing and Pre-processing the Data set

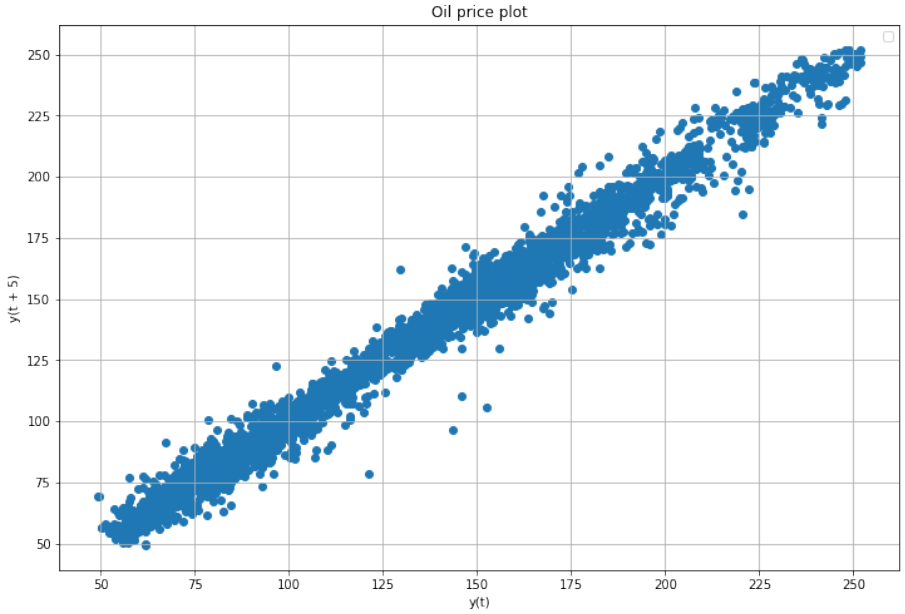
- The Data set is read and the descriptive statistics of the numerical attributes of the data set is calculated using describe().

	Open	High	Low	Close	Volume	OpenInt
count	4661.000000	4661.000000	4661.000000	4661.000000	4.661000e+03	4661.0
mean	128.617729	130.219184	127.028745	128.650169	6.257749e+06	0.0
std	47.634295	47.868451	47.391684	47.624879	7.243749e+06	0.0
min	49.252000	49.743000	43.241000	47.429000	0.000000e+00	0.0
25%	85.705000	87.093000	84.423000	85.771000	2.807258e+06	0.0
50%	128.950000	130.390000	126.810000	128.490000	4.144442e+06	0.0
75%	161.310000	163.120000	159.590000	161.330000	6.714921e+06	0.0
max	252.000000	253.420000	249.810000	251.180000	1.253138e+08	0.0

- The signal for open prices is plotted for visualization of trends.



- Auto correlation plot between present price and price after 5 days

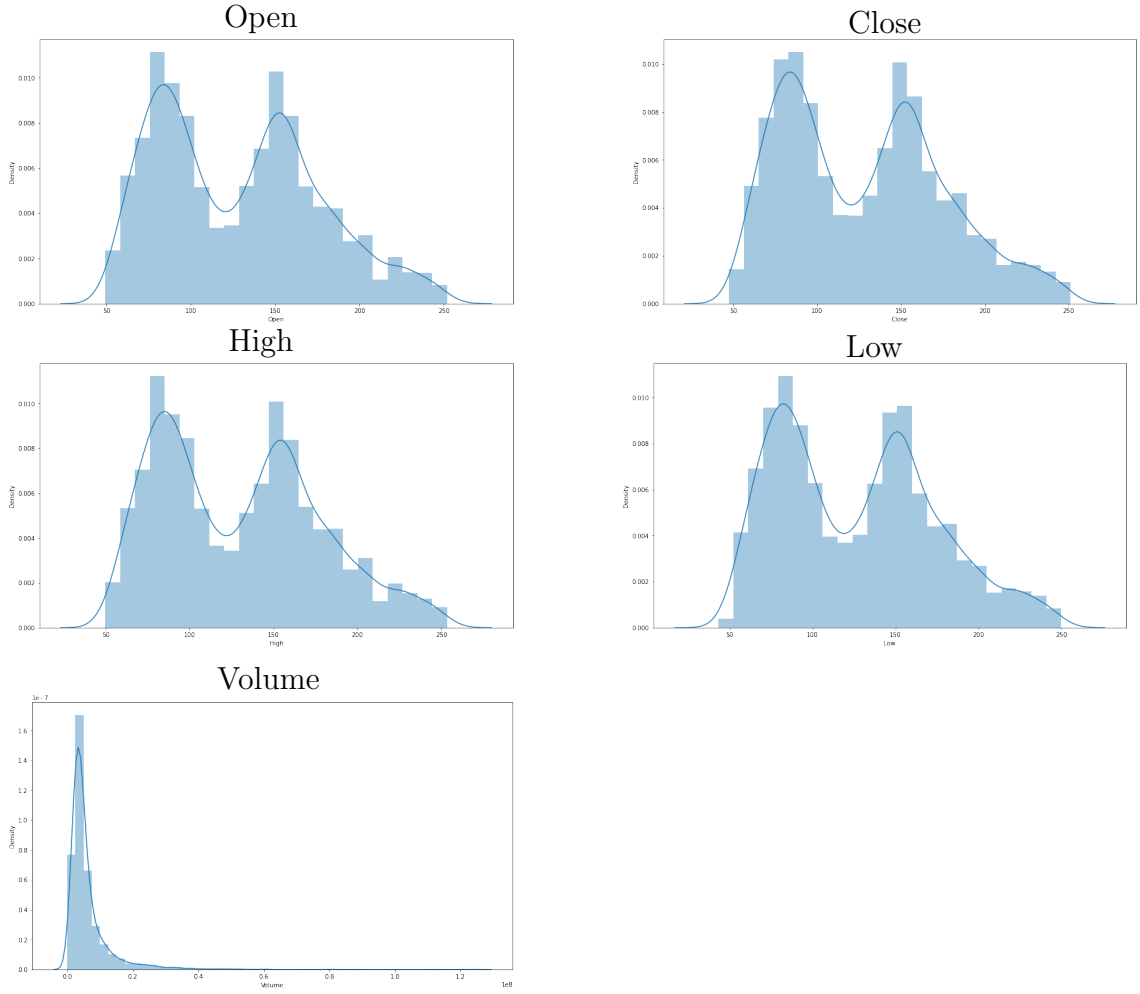


- Correlation plot is plotted for different time splits

	t	t+1	t+5	t+10	t+30
t	1.000000	0.994216	0.988282	0.956678	0.998288
t+1	0.994216	1.000000	0.992890	0.963463	0.992929
t+5	0.988282	0.992890	1.000000	0.971829	0.987001
t+10	0.956678	0.963463	0.971829	1.000000	0.954921
t+30	0.998288	0.992929	0.987001	0.954921	1.000000

- In the data pre-processing step the OpenInt column is dropped since it is not a redundant attribute. The data is checked for missing values. (No missing values).

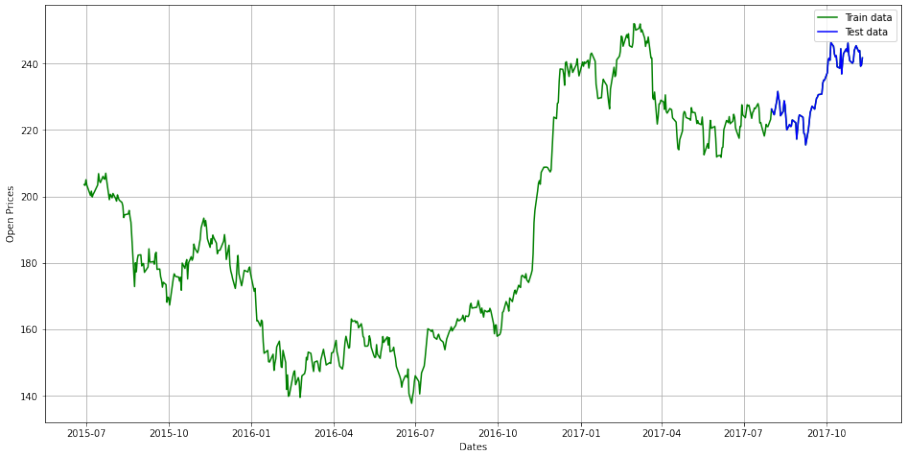
3.2 EDA



4 Model Building

- Since it is a time series data, to forecast the observation at (t+1) based on the historical data of previous time spots recorded for the same observation. So the model we are interested are AR, MA, ARMA, ARIMA. Also these models are compared to LSTM and GRU in the latter stage.

- The data is split into Training set and Testing set by keeping last 70 data points for testing purpose



4.1 AR Model

In a autoregression model, we forecast the variable of interest using a linear combination of past values of the variable. The term autoregression indicates that it is a regression of the variable against itself.
Thus, an autoregressive model of order of p can be written as

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t$$

where $\varepsilon_t$is white noise. This is like a multiple regression but with lagged values of y_t as predictors. We refer to this as an AR(p) model, an autoregressive model of order p .

4.2 MA Model

Rather than using past values of the forecast variable in a regression, a moving average model uses past forecast errors in a regression-like model.
Thus, an Moving average model of order of q can be written as

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q},$$

where ε_t is white noise. We refer to this as an MA(q) model, a moving average model of order q. Of course, we do not observe the values of ε_t, so it is not really a regression in the usual sense.

4.3 ARMA Model

autoregression and a moving average model, we obtain a non-seasonal ARMA model. The full model can be written as

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

where $\varepsilon_t$is white noise. The "predictors" on the right hand side include both lagged values of y_t and lagged errors. We call this an ARMA(p,d,q) model. .

4.4 ARIMA Model

If we combine differencing with autoregression and a moving average model, we obtain a non-seasonal ARIMA model. ARIMA is an acronym for AutoRegressive Integrated Moving Average (in this context, "integration" is the reverse of differencing).
The full model can be written as

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

where y'_t is the differenced series (it may have been differenced more than once). The "predictors" on the right hand side include both lagged values of y_t and lagged errors. We call this an ARIMA(p,d,q) model.

4.5 Estimation and order selection

4.5.1 Maximum likelihood estimation

Once the model order has been identified (i.e., the values of p,d and q), we need to estimate the parameters c $\theta_1, \ldots, \theta_q$ When R estimates the ARIMA model, it uses maximum likelihood estimation (MLE). This technique finds the values of the parameters which maximise the probability of obtaining the data that we have observed. For ARIMA models, MLE is similar to the least squares estimates that would be obtained by minimising

$$\sum_{t=1}^{T} \varepsilon_t^2$$

4.5.2 Information Criteria

Akaike's Information Criterion (AIC), which was useful in selecting predictors for regression, is also useful for determining the order of an ARIMA model. It can be written as

$$AIC = -2\log(L) + 2(p + q + k + 1),$$

where L is the likelihood of the data, k=1 if c0 and k=0 if c=0. Note that the last term in parentheses is the number of parameters in the model (including σ^2, the variance of the residuals).
For ARIMA models, the corrected AIC can be written as

$$AICc = AIC + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2},$$

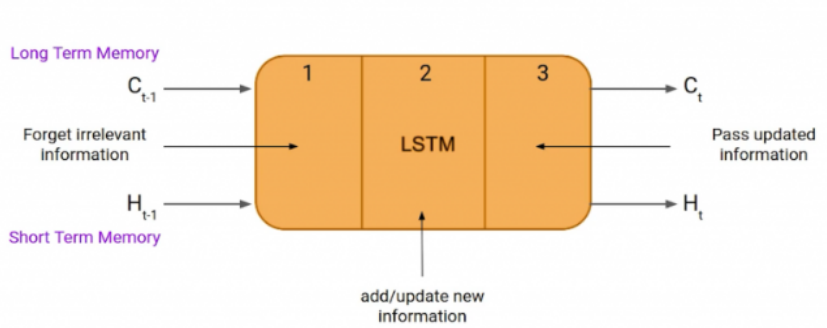
and the Bayesian Information Criterion can be written as

$$BIC = AIC + [\log(T) - 2](p + q + k + 1).$$

Good models are obtained by minimising the AIC, AICc or BIC. Our preference is to use the AICc.

4.6 LSTM Model

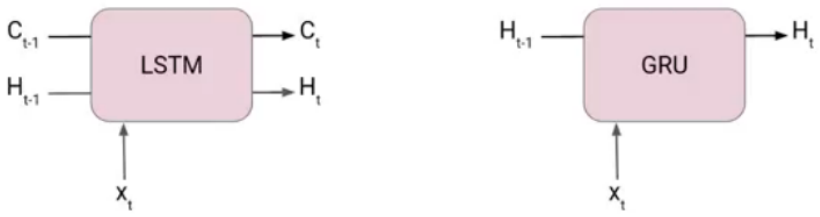
At a high-level LSTM works very much like an RNN cell. Here is the internal functioning of the LSTM network. The LSTM consists of three parts, as shown in the image below and each part performs an individual function.



The first part chooses whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten. In the second part, the cell tries to learn new information from the input to this cell. At last, in the third part, the cell passes the updated information from the current timestamp to the next timestamp

4.7 GRU Model

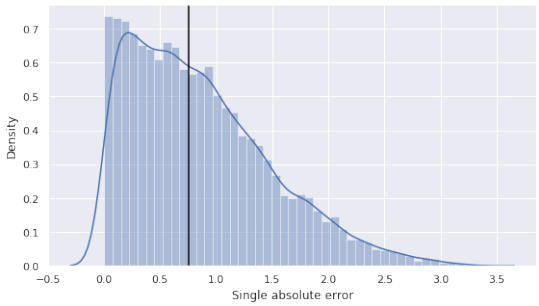
GRUs are very similar to Long Short Term Memory(LSTM). Just like LSTM, GRU uses gates to control the flow of information. They are relatively new as compared to LSTM. This is the reason they offer some improvement over LSTM and have simpler architecture.

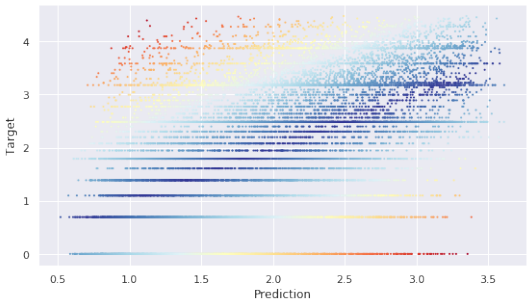


Another Interesting thing about GRU is that, unlike LSTM, it does not have a separate cell state (Ct). It only has a hidden state(Ht). Due to the simpler architecture, GRUs are faster to train.

4.8 Hyperparameter-Search class(Bayesian)

This is a class for hyperparameter search that uses Bayesian Optimization and Gaussian Process Regression to find optimal hyperparameters. I decided to use this method as the computation of the score for one catfamily model may be expensive. In this case bayesian optimization could be a plus. As this optimization methods takes some time as well you should try random search as well as this may be faster.

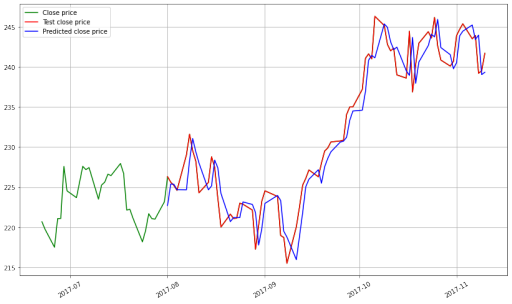




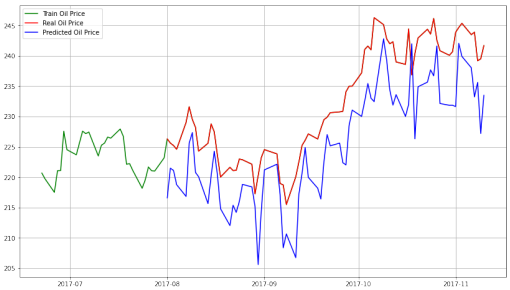
5 Simulations and Analysis

5.1 Simulations

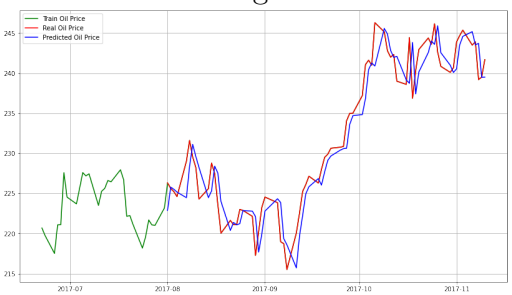
Estimation using AR Model



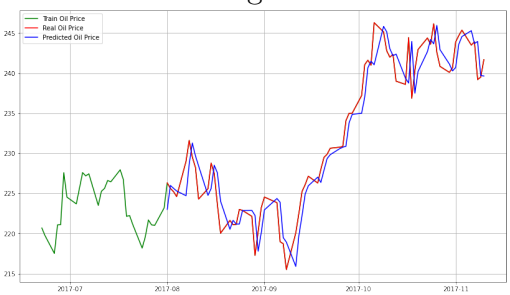
Estimation using MA Model



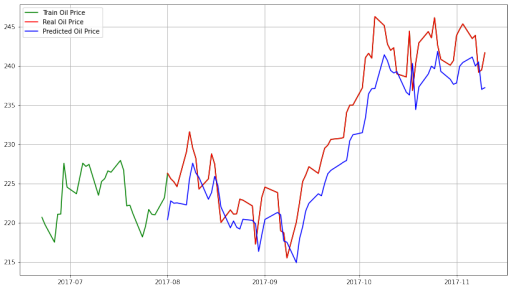
Estimation using ARMA Model



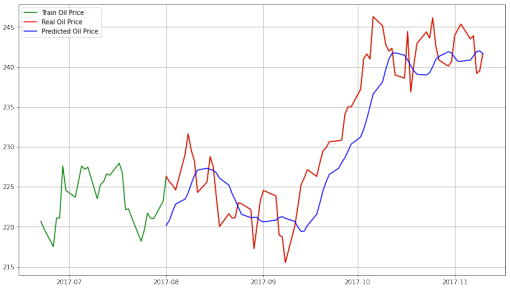
Estimation using ARIMA Model



Estimation using LSTM Model

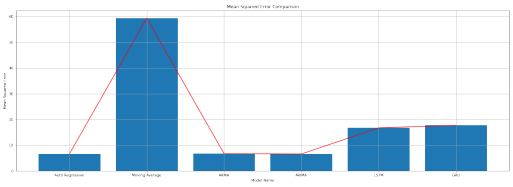


Estimation using GRU Model

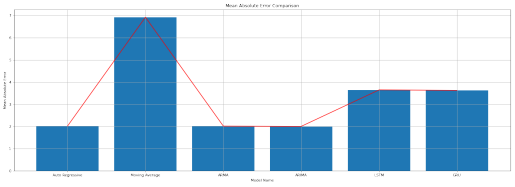


5.2 Analysis

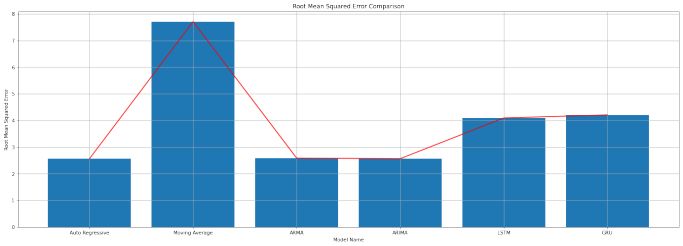
MSE(Mean Square error)



MAE(Mean absolute error)



RMSE(Root Mean Square error)



ARMA model parameters

ARMA Model Results

Dep. Variable:	y	**No. Observations:**	4660
Model:	ARMA(1, 8)	**Log Likelihood**	-11363.786
Method:	css-mle	**S.D. of innovations**	2.770
Date:	Sat, 30 Apr 2022	**AIC**	22749.572
Time:	06:12:31	**BIC**	22820.487
Sample:	0	**HQIC**	22774.516

	coef	std err	z	P>|z|	[0.025	0.975]
const	137.5026	35.938	3.826	0.000	67.065	207.941
ar.L1.y	0.9992	0.001	1668.542	0.000	0.998	1.000
ma.L1.y	-0.0975	0.015	-6.609	0.000	-0.126	-0.069
ma.L2.y	-0.0130	0.015	-0.877	0.381	-0.042	0.016
ma.L3.y	-0.0021	0.015	-0.139	0.889	-0.031	0.027
ma.L4.y	-0.0127	0.015	-0.851	0.395	-0.042	0.017
ma.L5.y	-0.0257	0.015	-1.749	0.080	-0.055	0.003
ma.L6.y	-0.0154	0.014	-1.066	0.287	-0.044	0.013
ma.L7.y	0.0118	0.015	0.805	0.421	-0.017	0.040
ma.L8.y	-0.0087	0.014	-0.612	0.540	-0.037	0.019

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.0008	+0.0000j	1.0008	0.0000
MA.1	-1.6870	-0.0000j	1.6870	-0.5000
MA.2	-1.1109	-1.2646j	1.6832	-0.3647
MA.3	-1.1109	+1.2646j	1.6832	0.3647
MA.4	0.3548	-1.7677j	1.8029	-0.2185
MA.5	0.3548	+1.7677j	1.8029	0.2185
MA.6	1.6459	-0.0000j	1.6459	-0.0000
MA.7	1.4509	-1.5391j	2.1152	-0.1297
MA.8	1.4509	+1.5391j	2.1152	0.1297

ARIMA model parameters

ARIMA Model Results

Dep. Variable:	D.y	**No. Observations:**	4659
Model:	ARIMA(1, 1, 1)	**Log Likelihood**	-11362.019
Method:	css-mle	**S.D. of innovations**	2.773
Date:	Sat, 30 Apr 2022	**AIC**	22732.039
Time:	06:13:21	**BIC**	22757.825
Sample:	1	**HQIC**	22741.109

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0371	0.036	1.035	0.301	-0.033	0.107
ar.L1.D.y	0.1754	0.187	0.936	0.349	-0.192	0.543
ma.L1.D.y	-0.2734	0.184	-1.490	0.136	-0.633	0.086

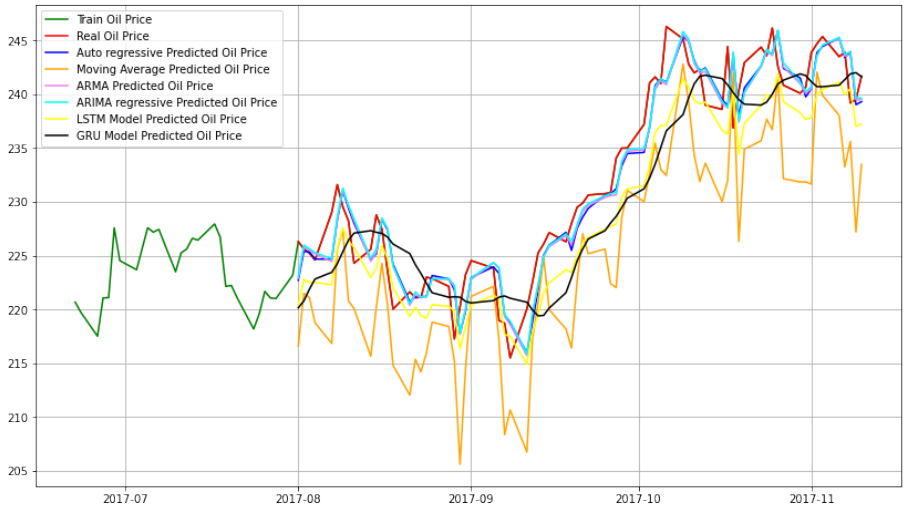
Roots

	Real	Imaginary	Modulus	Frequency
AR.1	5.7016	+0.0000j	5.7016	0.0000
MA.1	3.6579	+0.0000j	3.6579	0.0000

Comparison of all models used

	ModelName	MSE	MAE	RMSE
0	Auto Regressive	6.585071	2.020316	2.566139
1	Moving Average	59.365881	6.924000	7.704926
2	ARMA	6.676974	2.016496	2.583984
3	ARIMA	6.585688	2.004152	2.566260
4	LSTM	16.745473	3.647587	4.092123
5	GRU	17.703629	3.627497	4.207568

6 Conclusion



We can observe that the values of RMSE,MSE and MAE are least for AR, ARMA and ARIMA model. Since the error rate is less we can choose these a the best model from the comparisons.Since we have taken a time series model, we estimate the value based on the previous 30 days value which we trained.Auto ARIMA model was carried to represent the order that we should take as params to get an effective model. Bayesian Information Criterion and Maximum Likelihood Estimation is used in the ARIMA model to select the best parameters for our model.

7 References

1. `https://www.researchgate.net/publication/225877219_The_Cramer-Rao_Bound_for_Continuous-Time_Autoregressive_Parameter_Estimation_with_Irregular_Sampling`

2. http://cas.et.tudelft.nl/ leus/papers/icassp09$_2$.*pdf*