Foreign Exchange Rate Prediction

Suraj Mallikarjuna Devatha
sdevath@ncsu.edu
North Carolina State University

Sruthi Vandhana Thirunavukkarasu
sthirun2@ncsu.edu
North Carolina State University

Akhil Gangarpu Sudhakar
agangar@ncsu.edu
North Carolina State University

1 Background:

In the dynamic global economy, for any future investment, the better accuracy and correctness of Foreign exchange value is important. The Foreign exchange market is facing unprecedented growth over the last few years. Hence the accurate prediction of foreign exchange is very crucial for global investing and international businesses. It is very important to predict the forex rate, as it will help investors make better decisions to minimize risks and gain more returns. In modern time series forecasting, the foreign exchange prediction is one of the important and demanding applications. The forex rates are noisy, chaotic and non-stationary. This characteristic shows there is no clear past behavior employs in the data and there is no connection that we can explain clearly from the future data from that of the past.One assumption that we can make is that the historic data have all the behaviors incorporated in it. Hence the historic data helps to predict the future values better. However we cannot say how good this prediction is.

In this project we plan to design and implement ARIMA time series model that can predict future rates and its trends. We also implemented ARIMA with XGBoost methodology using the same dataset to evaluate which model is more accurate.

1.1 Literature Survey:

The project works on designing and implementing the ARIMA time series model along with certain enhancements in the method using XGBoost and the paper we referred to has taken the same kind of approach.

Deka et al. (2019) applied ARIMA methodology to forecast the consumer price index and an exchange rate of Turkish Lira and Turkish Inflation rate. They used ARIMA(3,1,3), ARIMA(3,1,1), ARIMA(4,1,1) and ARIMA(1,1,4) and compared the result of each model for exchange and inflation rate. The results revealed that ARIMA (3,1,3) and ARIMA (1,1,4) are providing the most effective forecasting in terms of the exchange rate and inflation rate, respectively. The success rate of the ARIMA model is very good and being evaluated for each and every parameter cases.

Babu et al.(2015) attempts to examine the performance of ARIMA, Neural Network and Fuzzy neuron models in forecasting the currencies traded in Indian foreign exchange markets for predictability of exchange rates of Rupee against USD, GBP, Euro and Yen. As a result, compared to non linear models such as Neural Network or Fuzzy Neuron model, the ARIMA model predicts a better exchange rate.

Islam et al.(2021) focuses the research on making forecasting applications and analyzing the exchange rate of USD against rupiah based on time series data or temporal datasets from the Investing.com site using machine learning methods, namely Extreme Gradient Boosting (XGBoost) and RMSE error value is calculated before creating new data set and after applying the boosting method.. As a result, The RMSE value is getting smaller when the model is used for forecasting the new dataset. So it can be said that the XGBoost method is suitable for forecasting, especially in the case of this exchange rate.

2 Proposed Method

Foreign Exchange rates are volatile and dynamic in nature whereas accurate forecasting of the future rates are highly challenging. The foreign exchange rate depends on various factors. Many people are trying to figure out the pattern of the forecasting rate. Despite the rates of the forex are dynamic, it is not random. It can be analysed by the series of Time Series data. The model we tried to implement is ARIMA and XGBOOST. The Forex rate prediction of USD to INR is impelemented using this model.

2.1 ARIMA

The AutoRegressive Integrated Moving Average (ARIMA) model is one of the most extensively used time series forecasting methodologies. Forecasts in an AutoRegressive model correspond to a linear combination of the variable's historical values. Forecasts in a Moving Average model are a linear accumulation of previous forecast errors. The ARIMA models, in effect, combine these two methodologies. Differentiating (Integrating) the time series, i.e. evaluating the time series of the differences instead of the original, may be a necessary step because they require the time series to remain stationary.

The parameters of ARIMA are defined as follows:

- p: The number of lagged observations included in the model, also known the lag order.
- d: The number of times that the raw observations are differenced, also known the degree of
- q: The size of the moving average window, also known as the order of moving average

2.2 XG Boost

Boosting is a sequential technique which works on the principle of an ensemble. It combines a set of weak learners and delivers improved prediction accuracy. XGBoost is an open-source software library which provides a regular gradient boosting framework for different programming languages.It aims to provide a Scalable, Portable and Distributed Gradient Boosting Library. XG-Boost can be added on top of any model to give better performance.

3 Plans and Experiment

The figure 1 shows the steps involved in the project implementation.

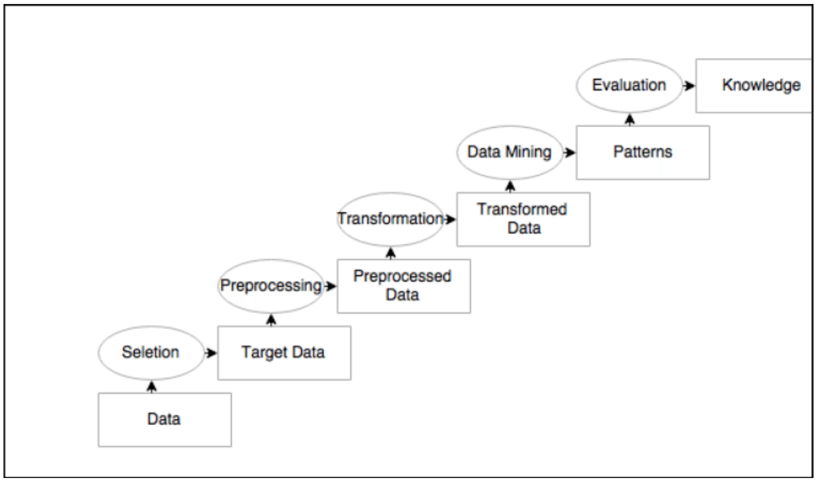
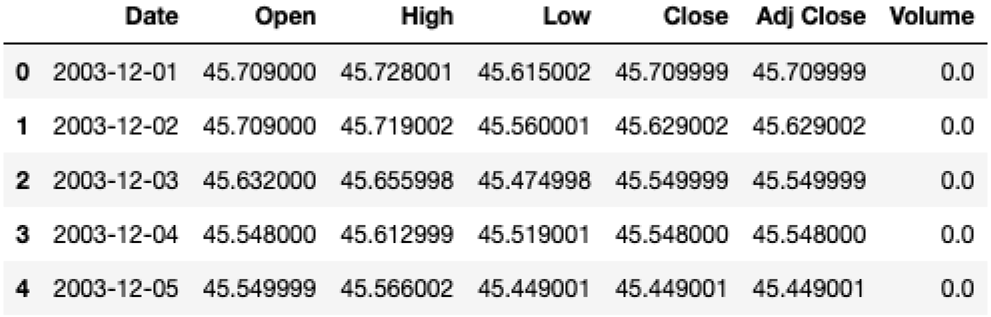


Figure 1: Steps involved in the project

3.1 Dataset

Dataset: The dataset is obtained from Yahoo Finance. The data includes 4661 observations of daily ask price of USD INR between the dates of 1st December, 2003 and 13th October, 2021. The dataset contains the following columns:

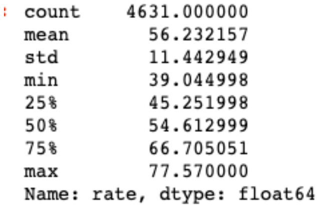
- Date: The trading session date
- Open: The first rate offered when the trading session starts.
- High: The highest rate during the trading session
- Low: The lowest rate during the trading session
- Close: The last rate offered when the trading session ends.
- Adjusted Close: Yahoo finances adjusts the closing prices considering the stock splits and dividends. This might not be applicable for currency exchanges and will be equivalent to the closing price.



	Date	Open	High	Low	Close	Adj Close	Volume
0	2003-12-01	45.709000	45.728001	45.615002	45.709999	45.709999	0.0
1	2003-12-02	45.709000	45.719002	45.560001	45.629002	45.629002	0.0
2	2003-12-03	45.632000	45.655998	45.474998	45.549999	45.549999	0.0
3	2003-12-04	45.548000	45.612999	45.519001	45.548000	45.548000	0.0
4	2003-12-05	45.549999	45.566002	45.449001	45.449001	45.449001	0.0

Figure 2: Dataset

This OHLC (Open, High, Low, Close) data is useful as it can determine the increasing or decreasing momentum of the rate. When the close and open are close to each other, it indicates that the momentum is weak. When the open and close are far from each other, it indicates strong momentum. The high and low show the full price range of the period, useful in assessing volatility.



```
count    4631.000000
mean       56.232157
std        11.442949
min        39.044998
25%        45.251998
50%        54.612999
75%        66.705051
max        77.570000
Name: rate, dtype: float64
```

Figure 3: Dataset Description

3.2 Hypothesis

While going through the project these are certain hypothesis come across to solve to get the desired behavior of the problem statement

1. How the forex rate changes with respect to time?
2. How the forex rate data gets distributed?
3. How to handle the missing or erroneous data?
4. How can we attempt to predict the forex rate for future?

3.3 Data Preprocessing

The data can have missing or irrelevant values and can cause problems during our analysis if this is not handled. Data cleaning is performed by removing the missing value using data frames from the panda's Python library. Also only relevant features are selected for our time series analysis and the remaining are ignored. Hypothesis three is being answered here. The missing and null value attributes are being processed in such a way that it is removed for training and testing as the dataset chosen have a lot of data that is sufficient for processing, removing would not be a problem in this case.

3.4 Data Visualisation:

The description of our dataset is shown in Figure 3.

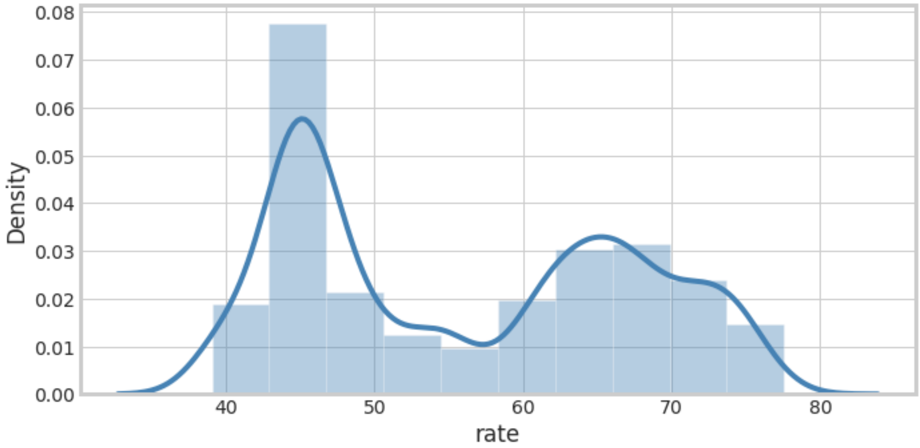


Figure 4: Dataset Distribution

Data distribution can be plotted using the distplot method from the Seaborn library. The distplot represents the univariate distribution of data i.e. data distribution of a variable against the density distribution. The data distribution of the dataset is shown in Figure 4. Hypothesis two is being answered here, the Figure 4 represent how the data is being distributed by examining the rate and the density.

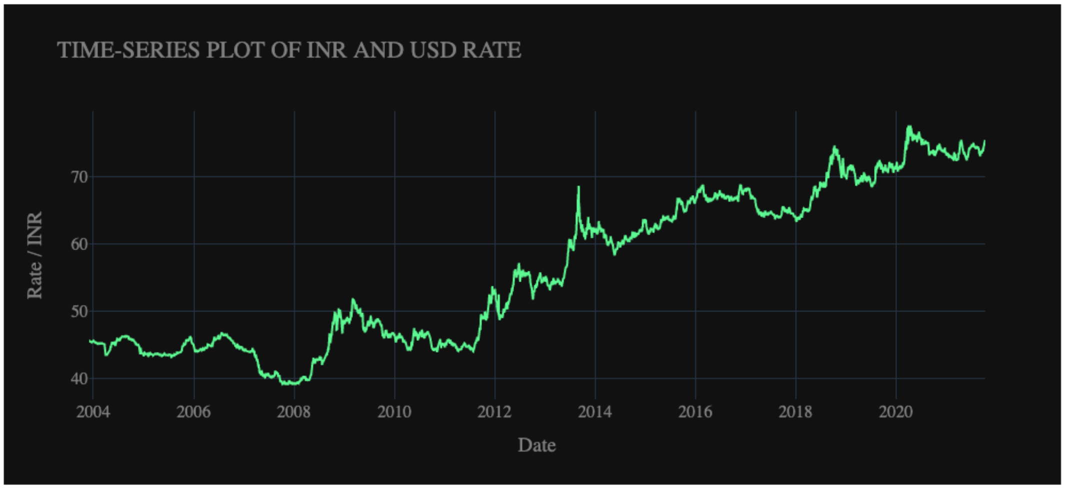


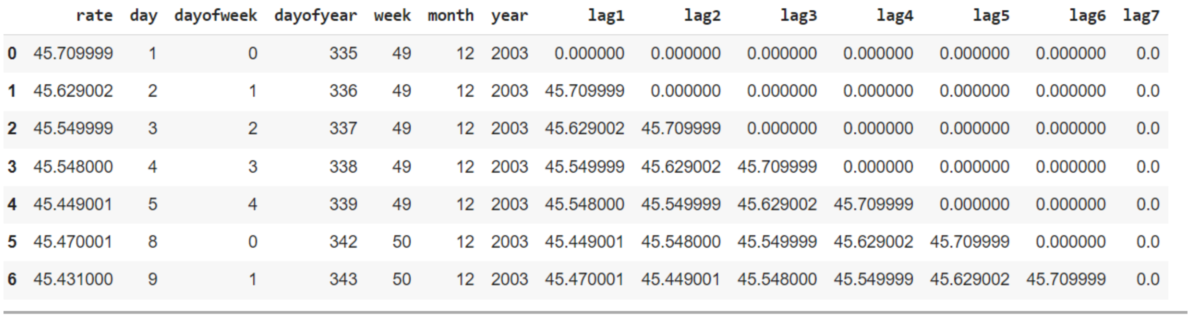
Figure 5: Time Series Graph

Time series data is a collection of observations recorded over even intervals of time and are later ordered chronologically. The time period at which the data was collected is called the time series frequency.To understand the patterns and behavior of the dataset, a time series graph can be plotted with observed values on the y-axis against an increment of time on the x-axis. How the forex rate changes with respect to time that is the hypothesis one is being answered from out time series graph in the figure 5. where we can see there is a slight curve down in 2008 which represent the market down happened. This is giving the clear data of the rate of INR and USD rate from 2004 to 2020.

The plotly Python library can be used to plot a time series graph as shown in Figure 5. The x-axis represents the time frame and the y-axis represents the Indian Rupee rate compared to US Dollar.

3.5 Data Transformation

Data Transformation is a step in data mining to convert the dataset from one format into another format to help in the analysis process. In this step, the dataset is converted into time series data, and further can be decomposed to daily, weekly, monthly or yearly intervals. The intervals helps us to analyse the pattern and how the rate is changing with respect to many factors in the economy. Figure 6 shows the Auto correlation and Partial Auto Correlation of the data.



	rate	day	dayofweek	dayofyear	week	month	year	lag1	lag2	lag3	lag4	lag5	lag6	lag7
0	45.709999	1	0	335	49	12	2003	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
1	45.629002	2	1	336	49	12	2003	45.709999	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
2	45.549999	3	2	337	49	12	2003	45.629002	45.709999	0.000000	0.000000	0.000000	0.000000	0.0
3	45.548000	4	3	338	49	12	2003	45.549999	45.629002	45.709999	0.000000	0.000000	0.000000	0.0
4	45.449001	5	4	339	49	12	2003	45.548000	45.549999	45.629002	45.709999	0.000000	0.000000	0.0
5	45.470001	8	0	342	50	12	2003	45.449001	45.548000	45.549999	45.629002	45.709999	0.000000	0.0
6	45.431000	9	1	343	50	12	2003	45.470001	45.449001	45.548000	45.549999	45.629002	45.709999	0.0

Figure 6: XG-Boost Feature Engineering

For applying XGBoost the data have to go to feature engineering and few data transformation have to be done. The day, days of week, days of year, month, year, week and lag values are being generated from the given dataset and it is considered as the new dataset and fed inside the XGBoost for further training and modelling.

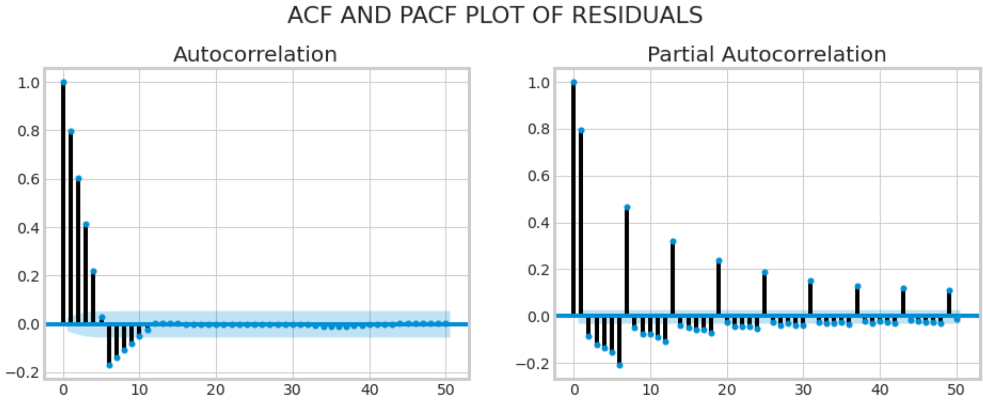
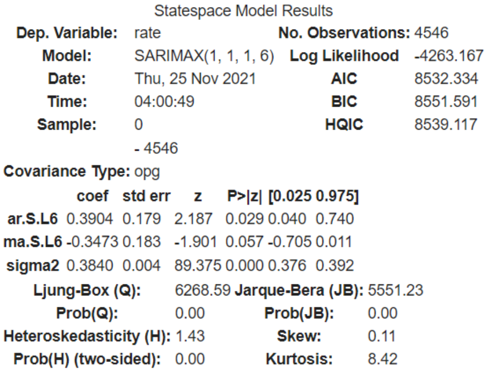


Figure 7: Auto-correlation graph

3.6 Modelling and Training

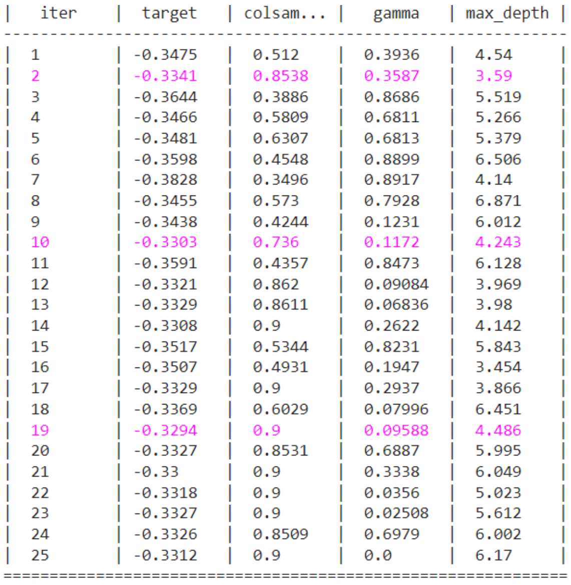
First model in the implementation is ARIMA with the parameter (1,1,6) this will take the initial preprocessed dataset and split the data into 90-10 ratio and model it into training and test dataset. The training data is fed and the process is being shown in the figure 8. As it is clear from figure 8 description that there are 4546 number of observations after preprocessing and being trained by the ARIMA model. ARIMA model always takes the time series data and process in such a way that it gives the proper output.



```
                    Statespace Model Results
 Dep. Variable:   rate                  No. Observations: 4546
      Model:      SARIMAX(1, 1, 1, 6)   Log Likelihood    -4263.167
       Date:      Thu, 25 Nov 2021            AIC         8532.334
       Time:      04:00:49                    BIC         8551.591
     Sample:      0                          HQIC         8539.117
                  - 4546
Covariance Type: opg
             coef   std err    z     P>|z| [0.025 0.975]
 ar.S.L6  0.3904  0.179   2.187   0.029 0.040  0.740
ma.S.L6 -0.3473 0.183   -1.901  0.057 -0.705 0.011
 sigma2  0.3840  0.004   89.375 0.000 0.376  0.392
      Ljung-Box (Q):        6268.59 Jarque-Bera (JB): 5551.23
          Prob(Q):          0.00          Prob(JB):    0.00
Heteroskedasticity (H): 1.43              Skew:        0.11
  Prob(H) (two-sided):  0.00          Kurtosis:        8.42
```

Figure 8: ARIMA Summary

The next model is XGBOOST along with ARIMA, which is a boosting technique which can be used with any other algorithm to improve the accuracy of the model of the time series model with large dataset. To feed the data inside the XGBoost we need to feature engineer the given dataset. From the preprocessing stage, the feature engineered data is fed into the XGBoost model. XGBoost goes into the Bayesian data optimization phase which is described in the Figure 9. The max depth parameter from the Bayesian optimization is taken as a parameter to train the dataset and the model uses 90-10 ratio for training and testing data.



```
|   iter    |  target   | colsam... |   gamma   | max_depth |
-------------------------------------------------------------
|  1        | -0.3475   |  0.512    |  0.3936   |  4.54     |
|  2        | -0.3341   |  0.8538   |  0.3587   |  3.59     |
|  3        | -0.3644   |  0.3886   |  0.8686   |  5.519    |
|  4        | -0.3466   |  0.5809   |  0.6811   |  5.266    |
|  5        | -0.3481   |  0.6307   |  0.6813   |  5.379    |
|  6        | -0.3598   |  0.4548   |  0.8899   |  6.506    |
|  7        | -0.3828   |  0.3496   |  0.8917   |  4.14     |
|  8        | -0.3455   |  0.573    |  0.7928   |  6.871    |
|  9        | -0.3438   |  0.4244   |  0.1231   |  6.012    |
|  10       | -0.3303   |  0.736    |  0.1172   |  4.243    |
|  11       | -0.3591   |  0.4357   |  0.8473   |  6.128    |
|  12       | -0.3321   |  0.862    |  0.09084  |  3.969    |
|  13       | -0.3329   |  0.8611   |  0.06836  |  3.98     |
|  14       | -0.3308   |  0.9      |  0.2622   |  4.142    |
|  15       | -0.3517   |  0.5344   |  0.8231   |  5.843    |
|  16       | -0.3507   |  0.4931   |  0.1947   |  3.454    |
|  17       | -0.3329   |  0.9      |  0.2937   |  3.866    |
|  18       | -0.3369   |  0.6029   |  0.07996  |  6.451    |
|  19       | -0.3294   |  0.9      |  0.09588  |  4.486    |
|  20       | -0.3327   |  0.8531   |  0.6887   |  5.995    |
|  21       | -0.33     |  0.9      |  0.3338   |  6.049    |
|  22       | -0.3318   |  0.9      |  0.0356   |  5.023    |
|  23       | -0.3327   |  0.9      |  0.02508  |  5.612    |
|  24       | -0.3326   |  0.8509   |  0.6979   |  6.002    |
|  25       | -0.3312   |  0.9      |  0.0      |  6.17     |
=============================================================
```

Figure 9: XG Boost Bayesian Optimization

4 Results

As a first step, we did data processing on our dataset and fixed the issues and made it ready to be used for model construction. Next we constructed an ARIMA model and XGBoost model and tried to predict the values for forex. This result was plotted against the actual values to do a comparison. The results are plotted in Figure 10 and 11. To answer the last hypothesis, ARIMA and XGBoost are used to predict the forex rate.

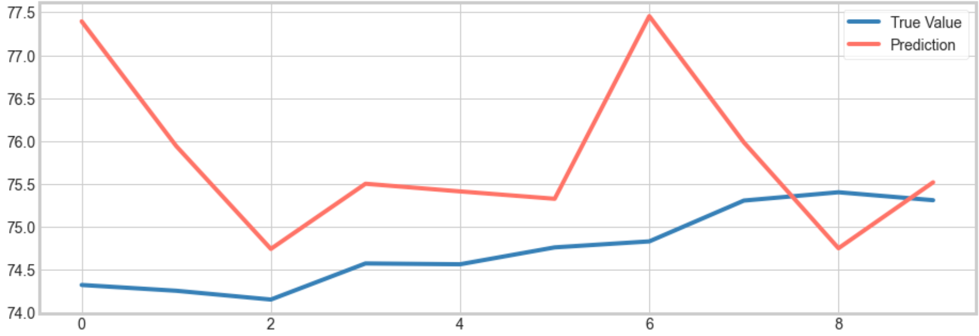


Figure 10: ARIMA Prediction

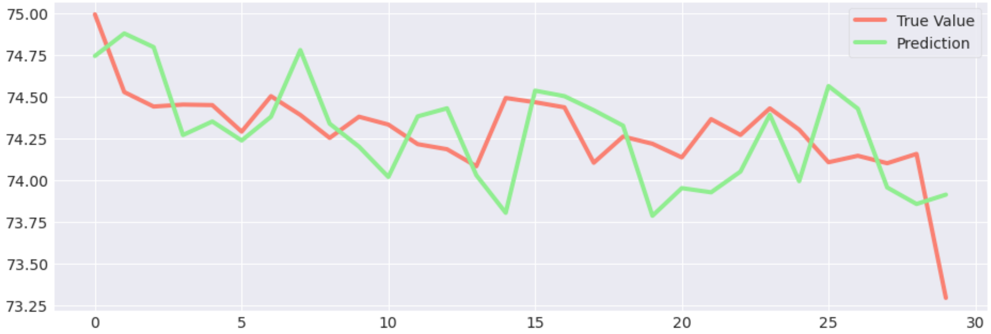


Figure 11: XG-Boost Prediction

We calculated error rates for our model as below:
For ARIMA we got
Mean Absolute Error: 1.185
Mean Squared Error: 2.236
Root Mean Squared Error: 1.495

For XG-Boost we got
Mean Absolute Error: 0.24943084716796876
Mean Squared Error: 0.08950113691389561
Root Mean Squared Error: 0.29916740616901366

5 Conclusion

Predicting the Forex exchange value for the Indian market is a crucial problem given the dynamic nature of the markets and the unprecedented growth India is seeing lately in its economy.

In the initial step, we cleaned the data to remove missing values and processed the data to select relevant features for the time series analysis. We explored various models that could be employed for our problem statement.

We implemented the ARIMA model and XG-Boost for the Forex Prediction project and obtained the results. Though both the models worked well, we found from the prediction result, that XG-Boost performs well and is better than the ARIMA model.

Github link: `https://github.ncsu.edu/sdevath/engr-ALDA-fall2021-P20`

6 References

1. Abraham Deka, Nil Gunsel Resatoglu,(2019) Forecasting Foreign Exchange Rate and Consumer Price Index with Arima Model: The Case of Turkey, International Journal of Scientific Research and Management (IJSRM) ,Volume 07 Issue 07

2. Babu AS, Reddy SK (2015) Exchange Rate Forecasting using ARIMA, Neural Network and Fuzzy Neuron, Journal of Stock Forex Trading, DOI: 10.4172/2168-9458.1000155

3. S F N Islam, A Sholahuddin, and A S Abdullah (2021) Extreme gradient boosting (XGBoost) method in making forecasting application and analysis of USD exchange rates against rupiah, Journal of physics : conference series, DOI: 10.1088/1742-6596/1722/1/012016

4. H. P. S.D Weerathunga, A. T. P. Silva (2018) DRNN-ARIMA Approach to Short-term Trend Forecasting in Forex Market, IEEE, DOI: 10.1109/ICTER.2018.8615580

5. H. Talebi, W. Hoang and M. L. Gavrilova, "Multi-scale foreign exchange rates ensemble for classification of trends in forex market", Procedia Computer Science, vol. 29, pp. 2065-2075, 2014