Walmart M5 Sales Forecasting

STAT 654-STATISTICAL COMPUTING WITH R & PYTHON-FINAL REPORT-TEAM 10

1. NAVEEN VELUCHAMY - 529005677
2. NAREN VIVEKANANDAN - 729007317
3. SHYIAM SUNDHAR NARAYANAN - 930001873
4. MOULIK PALANI - 729007246
5. PARTHASARATHY RAMAMOORTHY - 529005616

1. ABSTRACT

Demand plays an important role in the functioning of every business. It helps an organization reduce risks involved in business activities and make important business decisions. Demand forecasting provides insight into the capital investment and expansion decisions of an organization. It is a structured process that involves anticipating the demand (stochastic or imperfectly predictable) for the product and services of an organization in future under a set of uncontrollable and competitive factors.

Walmart is an American multinational retail corporation that operates a chain of hypermarkets, discount department stores, and grocery stores. The need to make decisions based on limited history is one of the biggest challenges of modeling retail data. If Christmas comes but once a year, so does the chance to see how strategic decisions impacted the bottom line. We are given sales data for over 5 years from 2011-01-29 to 2016-06-19. The objective is to predict the point sales for the next 28 days.

Time series forecasting is widely used in many areas such as economics, inventory systems, statistics, etc. There are many forecasting models ranging from basic models such as simple moving average and linear regression to more advanced models such as autoregressive integrated moving average (ARIMA) and Recurrent Neural Networks. These models analyze historical data in order to provide estimates of the future.

Keywords: Demand Forecasting, Walmart, predict point sales, Time series forecasting, Moving Average, ARIMA, Recurrent Neural Networks.

2. INTRODUCTION

We will use hierarchical sales data from Walmart, the world's largest company by revenue. The dataset involves the unit sales of 3,049 products, classified in 3 product categories (Hobbies, Foods, and Household) and 7 product departments (Foods_1, Foods_2, Foods_3, Hobbies_1, Hobbies_2, HouseHold_1, HouseHold_2) in which the above-mentioned categories are disaggregated. A total of ten stores, located in three States (CA, TX, and WI) where the products are sold. The historical data range from 2011-01-29 to 2016-06-19

Link to Dataset – click here

2.1 OBJECTIVE

The objective of this project is to predict sales data for the next 28 days into the future. We then assess the model accuracy by means of an error measure or scoring function.

2.2 DELIVERABLES

- Develop **Demand forecasting models** to assess the **next 28 days** of demand
- Evaluate the model using a scoring function like **Root Mean Squared Error (RMSE)** to assess the quality of the prediction.

3. EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis helps us comprehend the dataset which gives us visibility about its behavior. Here, we initially see how the demand for different categories like Hobbies, Household and Foods are among all the stores in different states combined *(Fig 3.1)*. We can observe the demand among the Food products being the most. Then, the same gross sales data is plotted in the form of a time series *(Fig 3.2)* where we can see the upward trend over time. This upward trend can be attributed to the ever-increasing population and the growing need. In the consecutive graph *(Fig 3.3)*, we have plotted rolling sales across all stores in the dataset. The sales curve has a "linear oscillation" trend at the macroscopic level. The sales oscillate like a sine wave about a mean value, but this mean value has a linear trend which is upward. This shows the sales oscillating at a higher and higher level every few months.

This trend is like that of the business cycle, where economies have short-term oscillatory fluctuations whereas grow linearly in the long run. Such small-scale trends at the level of stores might be adding up to decide trends we see at the macroeconomic level.

The *(Fig 3.4)* plot compares the sales distribution for each store in the dataset. The stores in California have the highest variance in sales, which could indicate that some places in California grow significantly faster than others. The Wisconsin and Texas sales are quite consistent among themselves, without much variance which might indicate that development might be more uniform in these states. The California stores seem to have the highest overall mean sales.

Due to computational factors, we have decided to proceed with the top 50 selling products in each category (Hobbies, Household, Foods). However, the function is built in such a way that any dataset can be accommodated regardless of its size. The final output of each model is a data frame that consists of 150 products and their point forecasts for the next 28 days.

---

4. TIME SERIES FORECASTING

Time series forecasting is an important technique because there are so many prediction problems that involve a time component. Here we will carry out three models that are considered highly effective.

4.1 MOVING AVERAGE

Moving Average is often used to model univariate time series. The output variable depends linearly on the current and various past values of a stochastic term.

A moving average of order m can be written as:

$$\widehat{T_t} = \frac{1}{m} \sum_{j=-n}^{n} \quad y_{t+j}$$

where m=2n+1. The estimate of the trend-cycle at time t is obtained by averaging values of the time series within n periods of t. Observations that are close in time to the prediction are also likely to be close in value. Thus, the averaging will eliminate some of the randomness in the data, giving a smooth trend-cycle component. We call this an m-MA, a moving average of order m. The *Fig 4.1.1* shows a 3-month moving average procedure.

Fig 4.1.2 displays a snippet of the point sales forecast for the next 28 days for the 150 products as discussed before. Several iterations were performed in order the optimal step size for the entire dataset. The step size which yields the lowest RMSE is chosen and the MA model is implemented from there on.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (P_i - O_i)^2}{n}}$$

The efficiency of the code is tested, and it turned out to be 4.08 seconds and the RMSE is 11.17.

4.2 AUTO REGRESSIVE INTEGRATED MOVING AVERAGE

An autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model. It is widely applied in statistics, econometrics and time series analysis. These models are fitted to time series data to either understand the data better or to predict future points in the series (forecasting). ARIMA models are used in situations where the data is non-stationary. For such cases, the initial differencing step (the "integrated" part of the model) can be done one or more times to eliminate the non-stationarity.

The ARIMA model is a combination of two models. The first is non-stationary:

$$Y_t = (1 - L)^d X_t$$

while the second is wide-sense stationary:

$$(1 - \sum_{i=1}^{p} (\emptyset_i) L^i) Y_t = (1 + \sum_{i=1}^{q} (\theta_i) L^i) \epsilon_t$$

Now forecasts can be made for Y_t by using a generalization of the method of autoregressive forecasting.

Differencing in statistics is a kind of transformation applied to time-series data in order to make it stationary. A stationary time series' property is one which will not depend on the time at which the series is observed.

To difference the data, the difference between consecutive observations is calculated. Mathematically, this is shown as:

$$y'_t = y_t - y_{t-1}$$

Differencing will remove the changes in the level of a time series, and thus eliminate trend and seasonality and eventually stabilize the mean of the time series. Sometimes it is necessary to difference the data a second time to get a stationary time series, which is known as second order differencing:

$$\begin{aligned} y^*_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= (y_t - 2y_{t-1} + y_{t-2}) \end{aligned}$$

Seasonal differencing is another method of differencing data, which computes the difference between an observation and the corresponding observation in the previous season e.g. a year. This is shown as:

$$y'_t = y_t - y_{t-m} \quad where\ m = duration\ of\ season$$

The differenced data can be now used for the estimation of an ARIMA model

AR component

$$AR(1)\ Y_T = A_1 * Y_{T-1}$$

$$AR(2)\ Y_T = A_1 * Y_{T-1} + A_2 * Y_{T-2}$$
$$AR(3)\ Y_T = A_1 * Y_{T-1} + A_2 * Y_{T-2} + A_3 * Y_{T-3}$$

MA component

$$MA(1)\ E_T = B_1 * E_{T-1}$$
$$MA(2)\ E_T = B_1 * E_{T-1} + B_2 * E_{T-2}$$
$$MA(3)\ E_T = B_1 * E_{T-1} + B_2 * E_{T-2} + B_3 * E_{T-3}$$

In order to implement ARIMA, we need three parameters **P,D,Q.** These parameters are different for each product depending on its previous demand. Hence, the function **pm.auto_arima** computes the optimal P,D,Q parameters for a product. This step is repeated for all products and their point forecasts are determined (*Fig 4.2.1*)

The efficiency of the code is tested, and it turned out to be 4005 seconds and the RMSE is 11.91.

4.3 LONG SHORT-TERM MEMORY (LSTM)

Long short-term memory (LSTM) is a type of Recurrent Neural Network (RNN) architecture used in the field of deep learning. While neural networks are usually feedforward, LSTM has feedback connections. It can process single data points (such as images) and also an entire sequence of data (such as speech or video). The LSTM recurrent neural network can learn long sequences of observations and is a perfect match for time series forecasting.

A LSTM unit consists of a cell, an input gate, an output gate and a forget gate. The cell remembers values over a set of arbitrary time intervals. Three gates regulate the flow of information into and out of the cell. LSTM networks are well-suited to processing, classifying, and making forecasts based on time series data, since there can be presence of lags of unknown duration between important events in a time series. LSTMs deal with the vanishing gradient problem that might be encountered when training traditional RNNs.

Normalizing - LSTMs are sensitive to the scale of the Input data. It is good practice is to rescale the data to the range of 0-to-1, which is called normalizing.

The input data (X) should be provided with a specific array structure in the form of: [samples, time steps, features]. 2 LSTM layers were added with ‘relu’ activation with optimizer and loss function as ‘Adam’ and ‘mse’. Relu ignores the negative neurons completely which introduces nonlinearity in the model. Adam Optimizer sets the learning rate for each parameter individually at each iteration which makes the model to learn better. This might lead to potential overfitting of data, in order to avoid it, dropout layer is added.

The data to be tested is normalized, reshaped to LSTM standards and then predicted and stored in a csv file along with their product id’s.

The point sales demand forecast using LSTM for the next 28 days can be seen in Fig 4.3.1.The efficiency of the code is tested, and it turned out to be 72.96 seconds and the RMSE is 2.23.

5. CONCLUSION

The comparison of the three methods was done using the Root Mean Squared Error. Below were the results obtained after running the three methods:

S.NO	Method	RMSE	Running Time (s)
1	Moving Average	11.17484	4.08
2	ARIMA	11.91526	4005
3	LSTM	2.23134	72.96

The RMSE comparison between the three methods can be seen in Fig.5.1 and the running time comparison can be seen in Fig.5.2.
The ARIMA model is one of the most suitable Time series models researched over time, but since it looks for the optimal parameters for each product it is computationally intensive.

But in terms of RMSE, LSTM seems to perform the best among the three methods.

Some observations that can be deduced from our work:

- Different states have different mean and variance of sales, reveals the variations in the distribution of development in these states.
- Most sales have a linearly trended sine wave shape, which is like that of the macroeconomic business cycle.

6. FUTURE SCOPE OF THE PROJECT

- We could use other forecasting techniques like exponential smoothing, Naïve approach etc.
- Do three separate forecasts for 3 states since there will be obvious operating differences /consumer trends between the three states
- Reduce running time for complex methods such as ARIMA.
- The dataset used for the project was only for Walmart. Though different, these analyses could be performed for any similar retail stores/consumer goods stores such as Target, HEB etc.

7. REFERENCES

- https://www.kaggle.com/c/m5-forecasting-accuracy
- https://www.sciencedirect.com/science/article/pii/S1018363918307554
- https://en.wikipedia.org/wiki/Signal_processing#Statistical
- https://en.wikipedia.org/wiki/Long_short-term_memory
- https://link.springer.com/article/10.1007/s11600-019-00330-1
- https://otexts.com/fpp2/moving-averages.html
- https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average#Forecasts_using_ARIMA_models

APPENDIX

LIST OF FIGURES

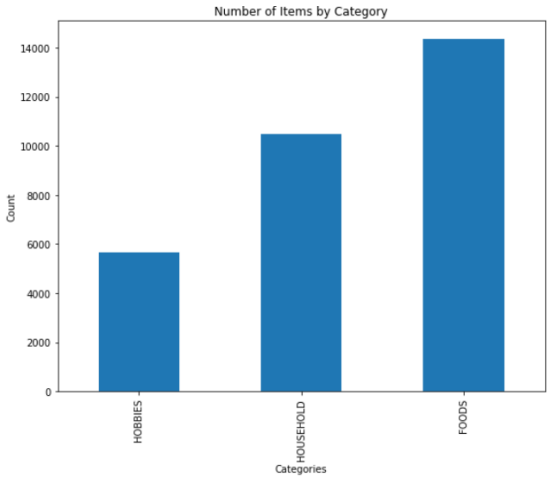

Figure 3.1. Number of items by each category

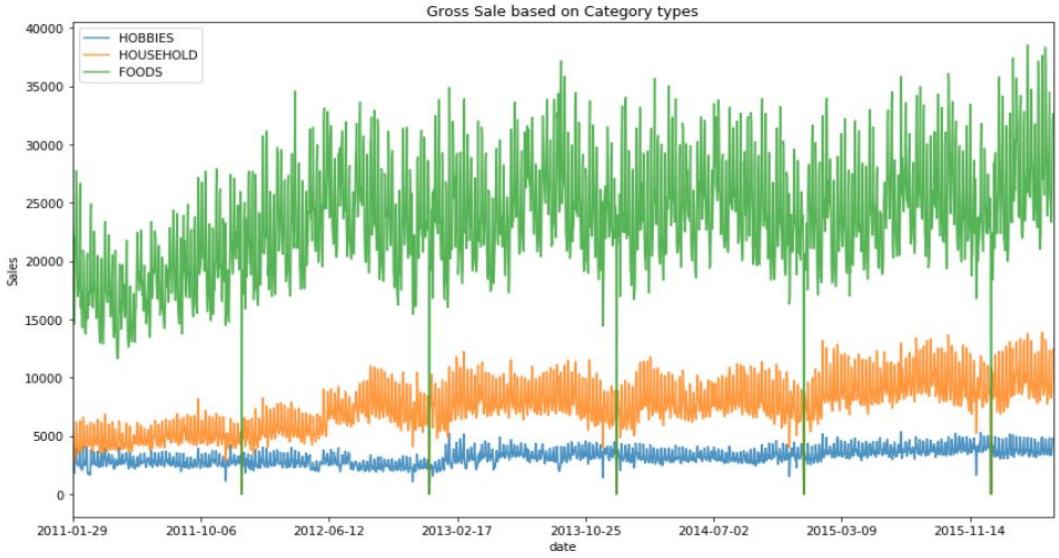

Figure 3.2. Gross Sale based on Category types

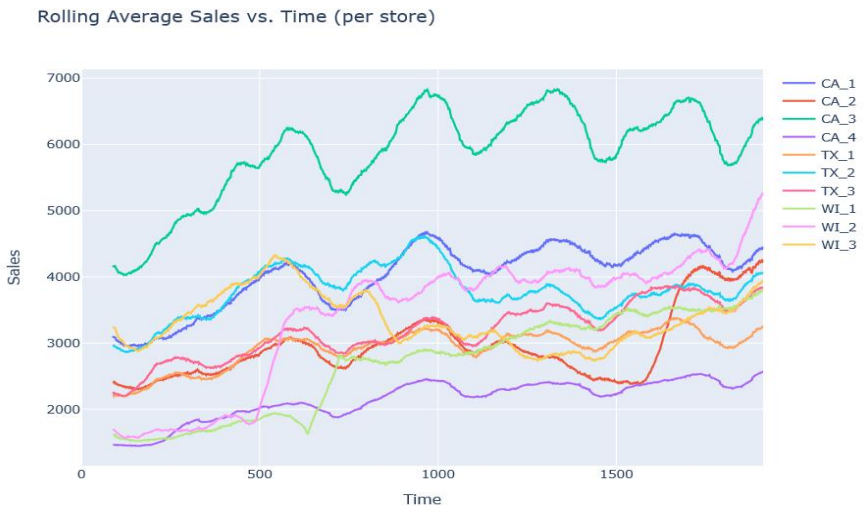

Figure 3.3. Rolling Average Sales vs Time (per store)

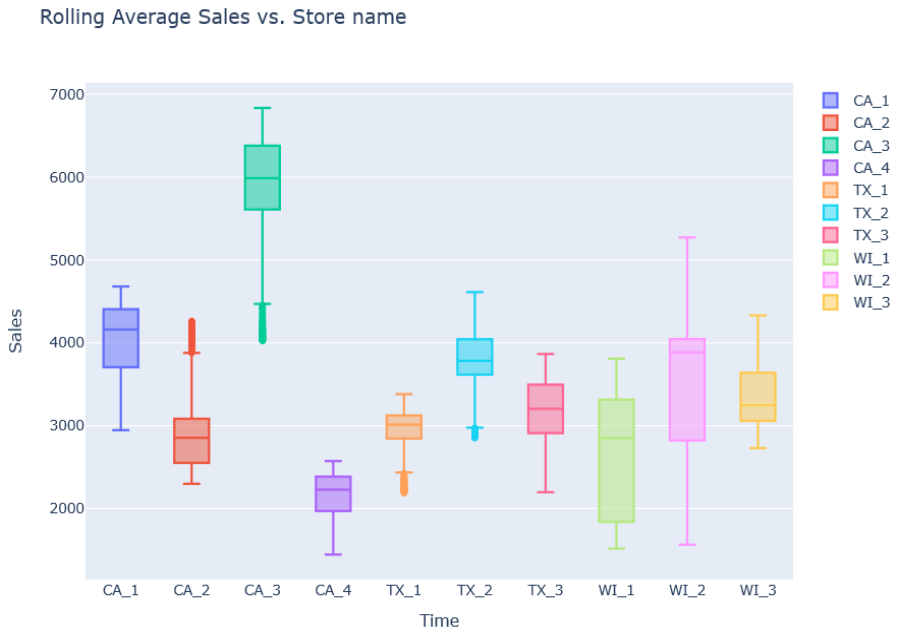

Figure 3.4. Rolling Average Sales vs Store name

MONTH	ACTUAL SHED SALES	THREE-MONTH MOVING AVERAGE
January	10	
February	12	
March	13	
April	16	(10 + 12 + 13)/3 = 11.67
May	19	(12 + 13 + 16)/3 = 13.67
June	23	(13 + 16 + 19)/3 = 16.00
July	26	(16 + 19 + 23)/3 = 19.33
August	30	(19 + 23 + 26)/3 = 22.67
September	28	(23 + 26 + 30)/3 = 26.33
October	18	(26 + 30 + 28)/3 = 28.00
November	16	(30 + 28 + 18)/3 = 25.33
December	14	(28 + 18 + 16)/3 = 20.67
January	—	(18 + 16 + 14)/3 = 16.00

Figure 4.1.1 Example for moving average for shed sales

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_10	...	d_19	d_20	d_21	d_22	d_23	d_24	d_25	d_26	d_27	d_28
6324	11	11	11	17	12	6	9	11	15	21	...	25	25	26	25	25	29	26	24	28	26
339	2	2	2	2	3	3	6	7	7	10	...	3	4	3	3	3	4	4	4	4	5
362	10	10	12	11	9	11	9	12	13	16	...	5	5	6	5	7	8	11	12	14	14
6344	16	17	17	19	16	16	12	12	10	9	...	19	18	17	19	21	21	14	15	15	17
6460	4	4	4	6	7	7	6	7	5	5	...	3	3	4	6	5	5	7	8	9	10
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
25374	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
28455	16	14	13	14	14	16	17	17	17	18	...	9	9	9	8	8	8	9	11	12	11
7167	16	14	14	15	14	13	13	14	13	14	...	11	13	13	14	13	13	13	13	13	13
18965	13	13	11	12	13	13	16	17	18	18	...	16	15	15	16	14	14	14	14	15	16
15890	4	5	5	5	5	4	5	6	6	4	...	5	5	5	5	4	7	7	6	5	6

150 rows × 28 columns

Figure 4.1.2. Moving Average prediction

	d_1886	d_1887	d_1888	d_1889	d_1890	d_1891	d_1892	d_1893	d_1894	d_1895	...	d_1904	d_1905	d_1906	d_1907	d_1908	d_1909	d_1910	d_1
6324	31.0	23.0	20.0	19.0	22.0	22.0	21.0	21.0	21.0	21.0	...	21.0	22.0	22.0	22.0	22.0	22.0	22.0	2
339	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	...	4.0	4.0	4.0	4.0	4.0	4.0	4.0	
362	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	...	8.0	8.0	8.0	8.0	8.0	8.0	8.0	
6344	14.0	14.0	14.0	14.0	14.0	14.0	14.0	14.0	14.0	14.0	...	14.0	14.0	14.0	14.0	14.0	14.0	14.0	1
6460	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	...	6.0	6.0	6.0	6.0	6.0	6.0	6.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
25374	1.0	1.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	...	2.0	2.0	2.0	2.0	2.0	2.0	2.0	
28455	17.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	...	16.0	16.0	16.0	16.0	16.0	16.0	16.0	1
7167	18.0	15.0	13.0	13.0	15.0	18.0	19.0	17.0	15.0	13.0	...	16.0	18.0	19.0	17.0	15.0	13.0	14.0	1
18965	12.0	12.0	13.0	13.0	13.0	13.0	13.0	13.0	13.0	13.0	...	13.0	13.0	13.0	13.0	13.0	13.0	13.0	1
15890	5.0	4.0	5.0	5.0	6.0	6.0	5.0	5.0	4.0	5.0	...	5.0	6.0	5.0	5.0	5.0	5.0	5.0	

150 rows × 28 columns

Figure 4.2.1 ARIMA prediction

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	...	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28
0	9.0	11.0	13.0	14.0	15.0	15.0	15.0	15.0	15.0	15.0	...	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
1	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	...	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0
2	7.0	8.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0	...	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0
3	9.0	12.0	14.0	15.0	16.0	16.0	17.0	17.0	17.0	17.0	...	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
4	6.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	...	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
145	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	...	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0
146	9.0	11.0	12.0	13.0	14.0	14.0	14.0	14.0	14.0	14.0	...	14.0	14.0	14.0	14.0	14.0	14.0	14.0	14.0	14.0	14.0
147	8.0	10.0	12.0	12.0	13.0	13.0	13.0	13.0	13.0	13.0	...	13.0	13.0	13.0	13.0	13.0	13.0	13.0	13.0	13.0	13.0
148	7.0	9.0	10.0	10.0	11.0	11.0	11.0	11.0	11.0	11.0	...	11.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
149	6.0	6.0	6.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	...	7.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0	6.0

150 rows × 28 columns

Figure 4.3.1 LSTM prediction

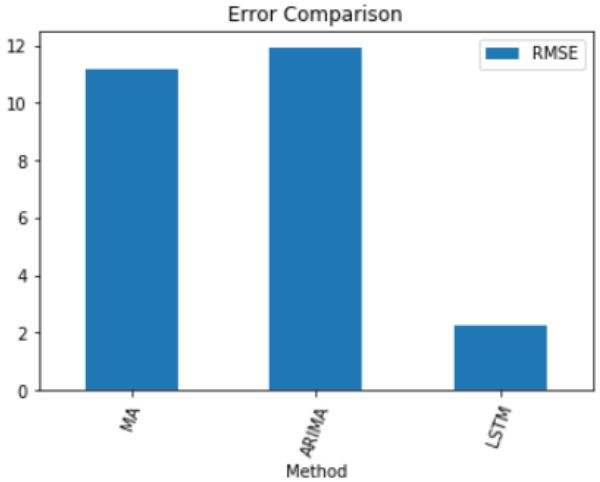

Figure 5.1. RMSE comparison for each method

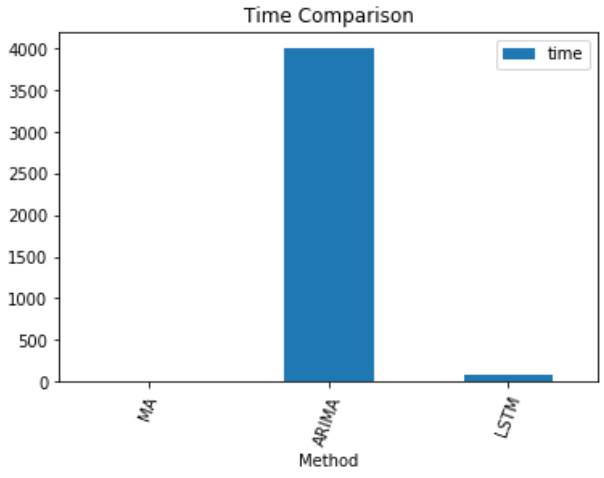

Figure 5.2. Time comparison for each method