PROJECT 2: Walmart Stores Forecasting

For this project we used the historical sales data for 45 Walmart stores that was provided to us. The aim was to build a model that can predict the Weekly Sales per store per department.

The kind of problem could be posed as a time series problem and with the data having temporal dependency we can extract a lot of information from the past data , deduce some pattern and use that to make future forecast.

I started off by doing some exploratory data analysis to get a first-hand feel of how the data looks like and what pre-processing would be helpful.

Following are the steps that I followed:

1. Since this was a time series data , I started by looking at the series and tried to figure out the seasonality and trend in the data. Careful analysis showed the data has seasonality with a period of 52 weeks(1 year) and very minimal trend.

2. I created ACF and PACF plots to see how the series is dependent on its past values and how can I deduce parameters for the ARIMA model.

3. I planned to use different models at different stages of walk forward validation since for t <6 we didn't had 2 full seasons of data which kind of restricted us in terms of the model selection.

4. The modelling was performed on every combination of store and department.

5. The dataset was prepared in a way that the missing value for any store, department combination was imputed by a zero as our best guess.

6. For t between 1 and 6 we use 3 models, One using **SNAIVE** , other using **SNAIVE with SVD** decomposition and last was **TSLM with SVD** decomposition. Snaive basically made the prediction for a week by taking its value from the last year of that same week.

7. We also performed linear modelling by using **TSLM** which fits an **linear model** to time series taking into account the trend and the seasonality component. This seems to work really well for fold<6.

8. The reason of using SVD decomposition was that it helped in smoothening out the data and performed better with the snaive algorithm.

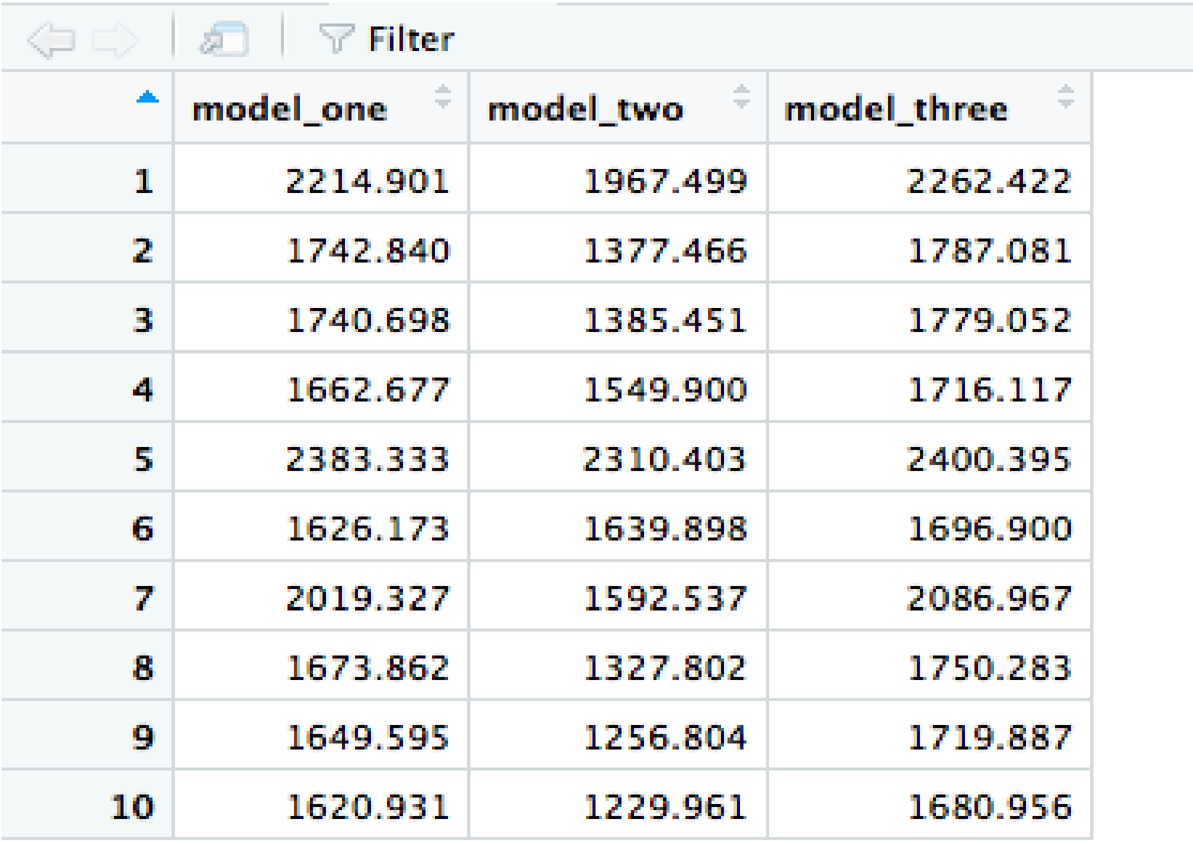
9. For t >6 we used snaive with and without SVD but along with that we used **STLF with SVD decomposition along with ARIMA.** This seemed to work really well since the model had data of at least 2 seasons and was well able to make forecast as the data can be seasonally adjusted by performing differencing. Again SVD helped in smoothening the series and ARIMA model

helped using Auto regression and Moving averages with the differencing performed.

10. STLF works by applying forecasting methods like **ETS** and **ARIMA** on the **seasonally adjusted data**, in our case we use ARIMA , we tried with ETS but that didn’t gave much accuracy.

11. ARIMA uses 2 models an AR models which models an observation with its lag value and MR model which models the residual with its lag residuals.

Model Evaluation Results:



	model_one	model_two	model_three
1	2214.901	1967.499	2262.422
2	1742.840	1377.466	1787.081
3	1740.698	1385.451	1779.052
4	1662.677	1549.900	1716.117
5	2383.333	2310.403	2400.395
6	1626.173	1639.898	1696.900
7	2019.327	1592.537	2086.967
8	1673.862	1327.802	1750.283
9	1649.595	1256.804	1719.887
10	1620.931	1229.961	1680.956

Average for 10 folds:

model_one = 1833.434
model_two =1563.772
model_three = 1888.006

Running Time:

The script completed its full run on all 10 folds in **36.92372 mins**.

System Specifications:

Model Name:	MacBook Air
Model Identifier:	MacBookAir7,2
Processor Name:	Intel Core i5
Processor Speed:	1.6 GHz
Number of Processors:	1
Total Number of Cores:	2
L2 Cache (per Core):	256 KB
L3 Cache:	3 MB
Memory:	8 GB

Acknowledgements:

We have had many discussions on piazza about the approach, the instructors helped in sharing the approach and the code framework for this project. We also looked at many solutions of the winner code and many discussion thread on Kaggle where people had posted their suggestions.