Observe the effect of change in stock price with change in USD INR exchange

Team Ensemble

EBAC 4 (Mixed Group), Institute of Systems Science, NUS

[Kriti: A0163206N,Muni: A0163382E,Pooja: A0163281J,Pradeep: A0163453H,Sambit: A0163285B]

**Abstract**

This work is part of assignment to apply transfer function on univariate time series of Bombay Stock Exchange data and USD-INR exchange from Year 2006-2016. This is a daily data. Following are the sources of data:

USD-INR Exchange Rate: <https://www.investing.com/currencies/usd-inr-historical-data>

Bombay Stock Exchange: <http://www.bseindia.com/indices/IndexArchiveData.aspx>

Natural Rubber Production: <http://www.cds.edu/research/ru/nrppd/database-on-plantation/rubber/>

<http://ijcttjournal.org/Volume4/issue-5/IJCTT-V4I5P78.pdf>

# Executive Summary

## 

## Data Selection and Preprocessing

Upon analyzing the dataset, we can categorize the data in following segments:

### Days variables: These variables are day of week, weekend, holiday, season. These variables are known in advance for predicting the demand and we feel that there is requirement to create a lag variable for them.

### Weather variables: Variables like temperature, humidity, wind-speed could be segmented in to this category

### Demand variables: Variables like casual, Registered, cnt comes under this category. These variables are known at the end of the day.

Raw daily dataset contains 16 columns time series for 2011 and 2012. It contains total 731 records. Three variables (instant, dteday, and yr: year) could be ignored as they are simple index variable. yr variable simply divides the data in to 2 years.

*R summary output of the various variable is as shown below:*

*'data.frame': 731 obs. of 15 variables:*

*$ season : int 1 1 1 1 1 1 1 1 1 1 ...*

*$ mnth : int 1 1 1 1 1 1 1 1 1 1 ...*

*$ holiday : int 0 0 0 0 0 0 0 0 0 0 ...*

*$ weekday : int 6 0 1 2 3 4 5 6 0 1 ...*

*$ workingday: int 0 0 1 1 1 1 1 0 0 1 ...*

*$ weathersit: int 2 2 1 1 1 1 2 2 1 1 ...*

*$ temp : num 0.344 0.363 0.196 0.2 0.227 ...*

*$ atemp : num 0.364 0.354 0.189 0.212 0.229 ...*

*$ hum : num 0.806 0.696 0.437 0.59 0.437 ...*

*$ windspeed : num 0.16 0.249 0.248 0.16 0.187 ...*

*$ casual : int 331 131 120 108 82 88 148 68 54 41 ...*

*$ registered: int 654 670 1229 1454 1518 1518 1362 891 768 1280 ...*

*$ cnt : int 985 801 1349 1562 1600 1606 1510 959 822 1321 ...*

## Data Partition

Two types of data partitions were done based on model analysis:

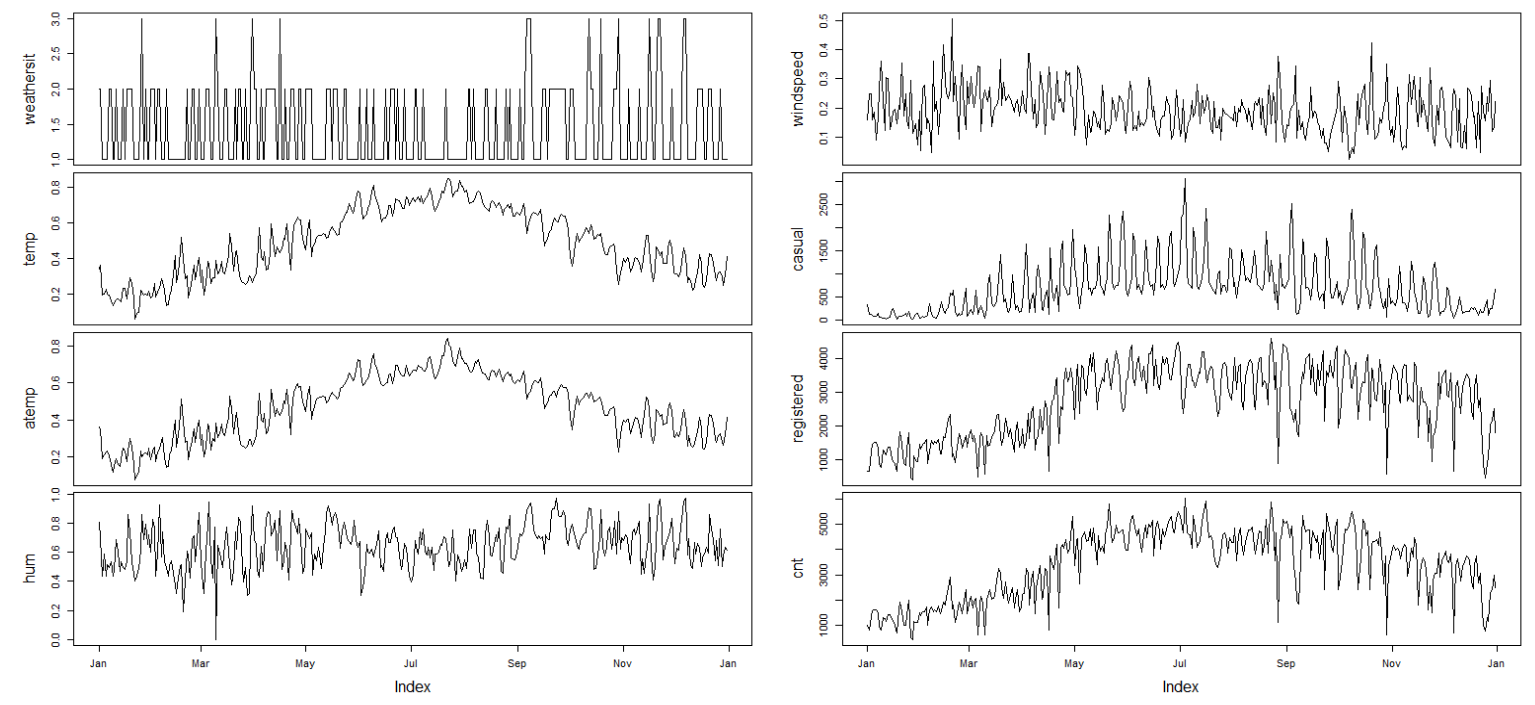
|  |  |  |
| --- | --- | --- |
| **Type** | **Train Set** | **Test Set** |
| 1 (Used in many models) | Year 2011 | Year 2012 |
| 2 (Used in one model) | Year 2011 – Mid 2012 | Mid-Year 2012 – Year 2012 End |

*Please note that we used separate csv for different models as team members used different names and transformations in their respective csv’s. Everyone used the copy of Daily.csv provided as part of assignment.*

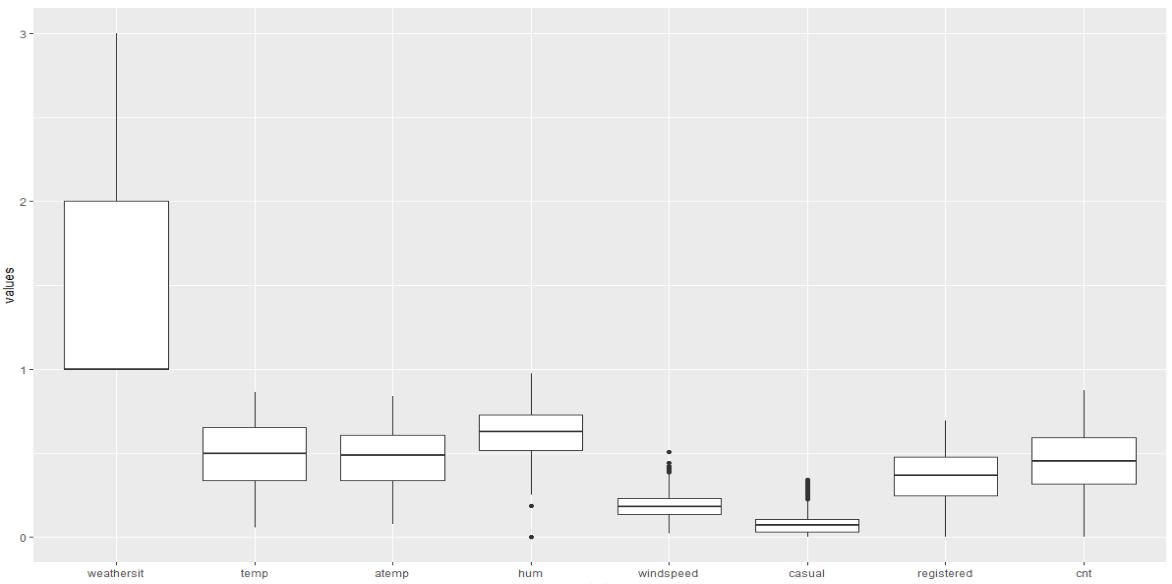
## Feature Engineering

Feature analysis was performed on various variables using Time series line chart and outlier analysis using the Box plots as shown below in Fig (1) and Fig (2)

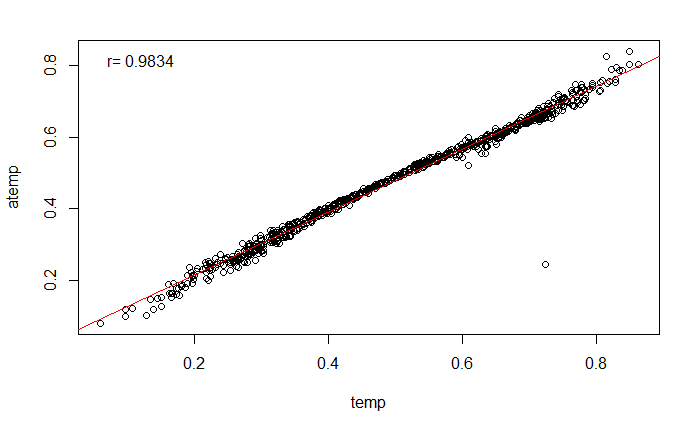
**Fig (1) Time Series Plot**



**Fig (2) Box Plot Chart**



**Fig (3) Correlation between temp and atemp**



Here is the summary of our analysis and action taken:

* No variables found to have missing values
* humidity parameter has one zero value. We replaced this value with past 7 days moving average
* cnt, casual and registered variable have multiple unusual drops. Given the geographic location of Washington D.C. This could be due to harsh weather conditions. A simple google search showed up some instances e.g. in January. This data was imputed with 7 days moving average after trial with 1 to 10 days of moving average.
* We added 7 days moving average for weather parameters as well.
* temp and atemp are highly correlated variable (Fig 3). So we used atemp in our model
* We created a diff variable of temp and atemp by reversing the normalizing process.

as *atemp\*50 - temp\*41*

## Model Building

While the business problem is of demand forecasting for maximizing the profit, it can be approached in multiple ways. We used different kinds of variables (Day, Weather and Demand) in different stages of modeling. This differential treatment also ensures that the predictions from different techniques are less correlated. This will help create a robust predictor as the Ensembling input.

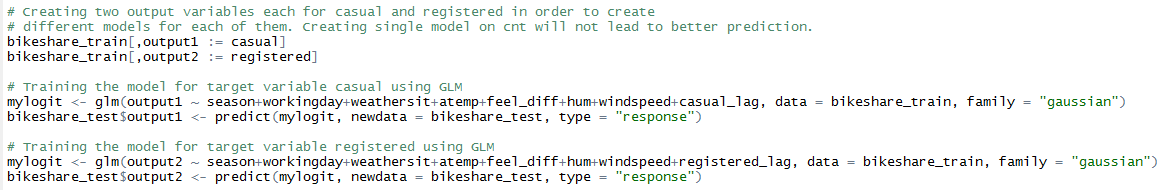
Predictions are evaluated using total profit and RoI for business performance. Six different analytical techniques were used in this project. The common philosophy behind these techniques is that we break the prediction problem in two parts. We assume that there is an underlying demand pattern which does not depend on the Day or Weather variables. Given the nature of industry, it is expected (and observed) to be an increasing pattern with a strong seasonality and yearly effect. We forecast this using univariate time series analysis techniques on the demand variables.

There is a realization that the trend remains same for consecutive years but the demand for recent year is inflated as compared to previous one. Thus, training the demand variable using previous year and testing the same on following years must be accompanied by normalization to balance excessive volatility. Also, different models were created each for different types of demand (casual and registered) as we believe that casual and registered follow different patterns. Finally the output were added and normalized to predict the cnt.

### Generalized Linear Models

As a benchmark, we started with **glm** which is used to fit generalized linear models, specified by giving a symbolic description of the linear predictor and a description of the error distribution. It was quite evident that there is a strong weekly pattern and thus weather related fields were replaced with last 7 days mean to get better prediction. The casual and final count demand was also considered as one of the predictor followed by normalization to balance excessive volatility in 2012.

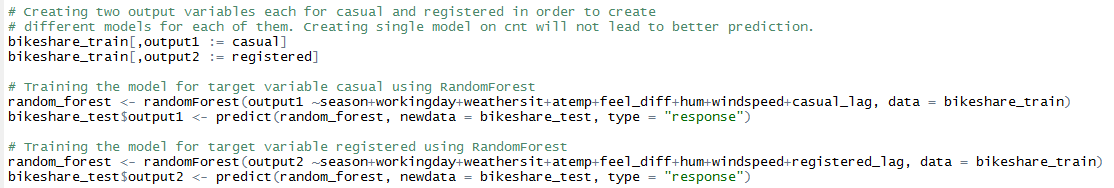
Guided by Gaussian family, code snippet is below:



### Random Forest

**Random forests** combine the predictions of multiple decision trees. The dataset is repeatedly divided into subtrees, guided by the best combination of variables. However, finding the right combination of variables can be difficult. For instance, a decision tree constructed based on a small sample might be not be generalizable to future, large samples. To overcome this, multiple decision trees could be constructed, by randomizing the combination and order of variables used. The aggregated result from these forests of trees would form an ensemble, known as a random forest. Weather related fields were replaced with last 7 days mean to get better prediction. The casual and final count demand was also considered as one of the predictor followed by out normalization to balance excessive volatility in 2012.

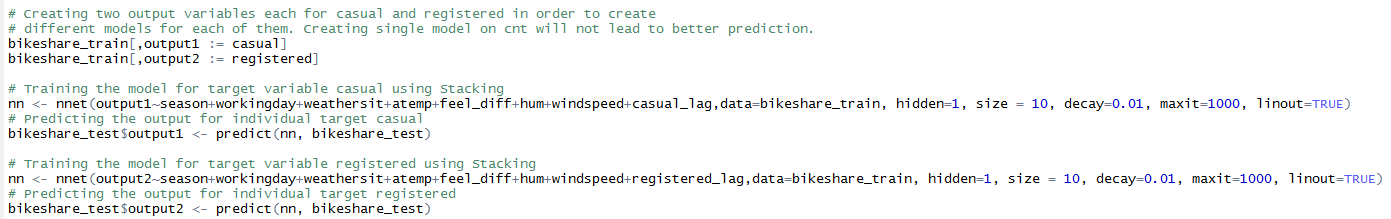
Code snippet is below:

**

### Neural Network

Based on experience, Random forests may require more data but they almost always come up with a pretty robust model. On the other side, neural network algorithms require "relatively" large datasets to work well.

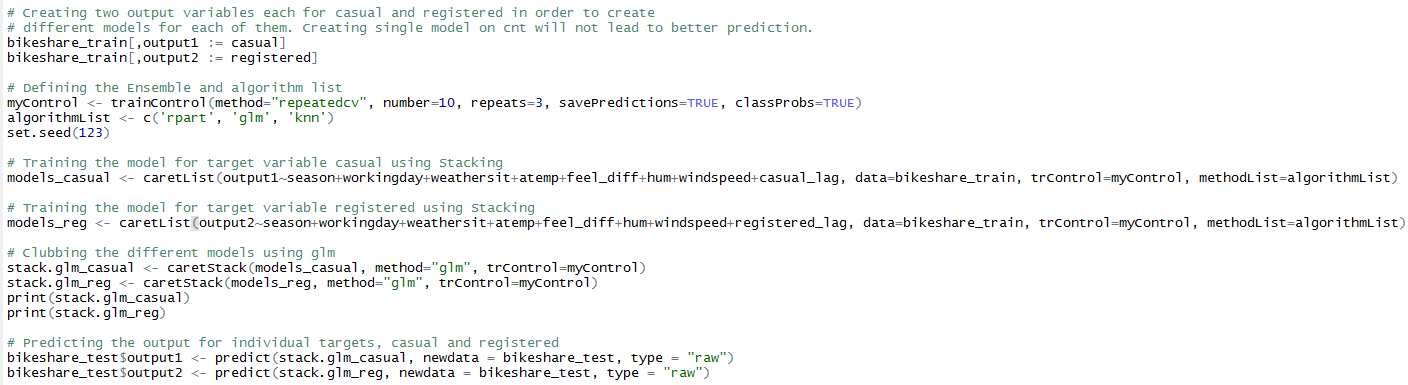
Code snippet is below:

**

### Ensemble Stacking

It can take time to find well performing machine learning algorithms for dataset. This is because of the trial and error nature of applied machine learning. Once someone has a shortlist of accurate models, you can use algorithm tuning to get the most from each algorithm. Another approach that one can use to increase accuracy on dataset is to combine the predictions of multiple different models together. Stacking is a process of building multiple models (typically of differing types) and supervisor model that learns how to best combine the predictions of the primary models. GLM was used as a supervisor model that best combined the predictions from rpart, glm and knn.

Code snippet is below:

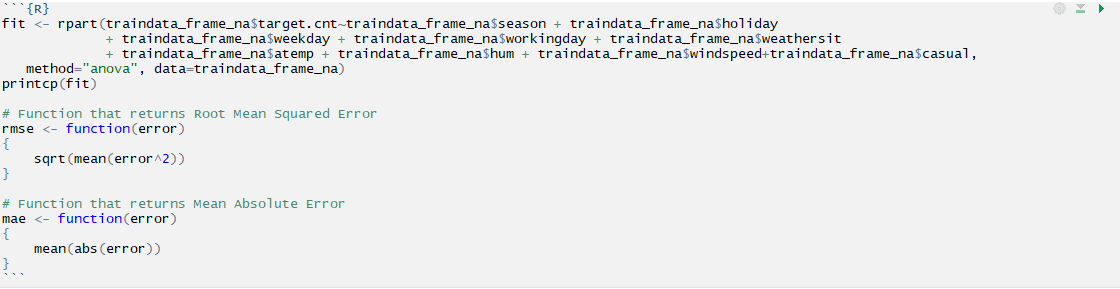


### Decision Trees

Under the Individual category, we also made model using the Decision Tree. We built mainly 2 models:

1. Using weather and demand variables
2. Create the Moving average of 5 days and used 1.5 years as train data

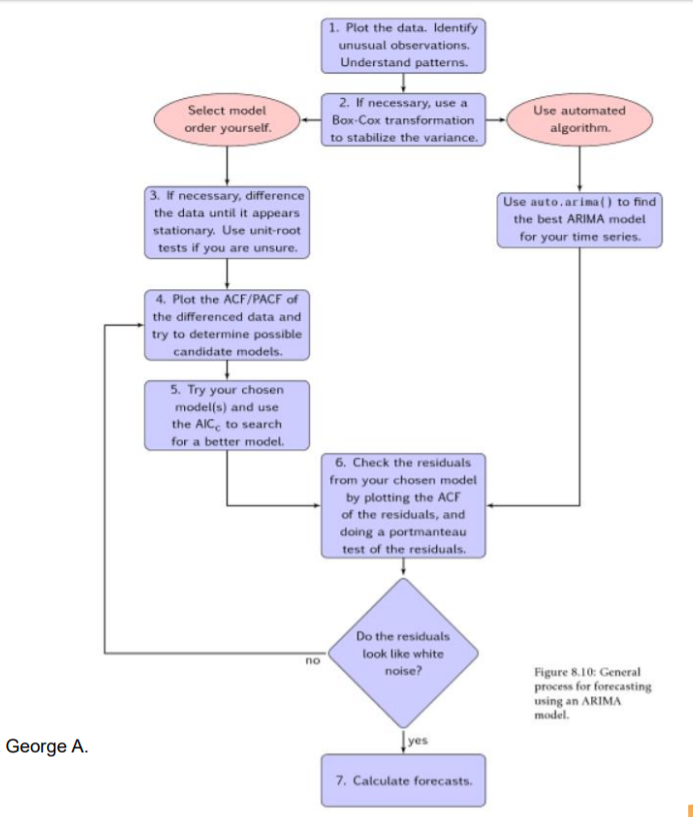
Overall, weather variables were found to be actually used in the construction and business and model performance for the same is mentioned in the respective tables below.



### Time Series Analysis – ARIMA (1,1,1)

We also performed the Univariate Time series analysis using ARIMA models. We took the total count (cnt) variable and prepared model by following the flow chart below:

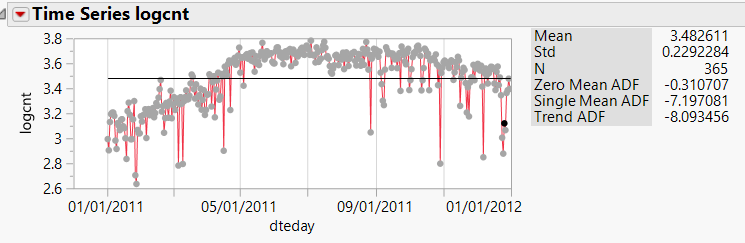
**Flow Chart to perform ARIMA**



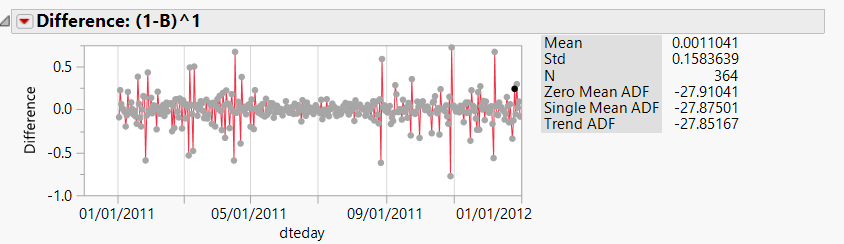
**Observations:**

1. In first iteration, we used the cnt variable as it is and plotted the time-series
2. We used 1 order differencing and applied ***Dickey-Fuller Test***. Data output seems to be stationary
3. We then used the ACF and PACF plot to see the ARIMA order
4. Finally, we ran the ARIMA (1,1,1) model. Residual coefficient plot *(Ljung-Box Q test)* still showed significance at *p-value*
5. Hence, we decided to transform the cnt variable by taking log(cnt) to dampen the effect of seasonality and random effect.
6. We followed the step (1-4) again and found that this time we achieved the White Noise and a good AIC value to consider it as a model.
7. We used the model at Step 6 for the forecasting. Below is the Jmp output of our final model:

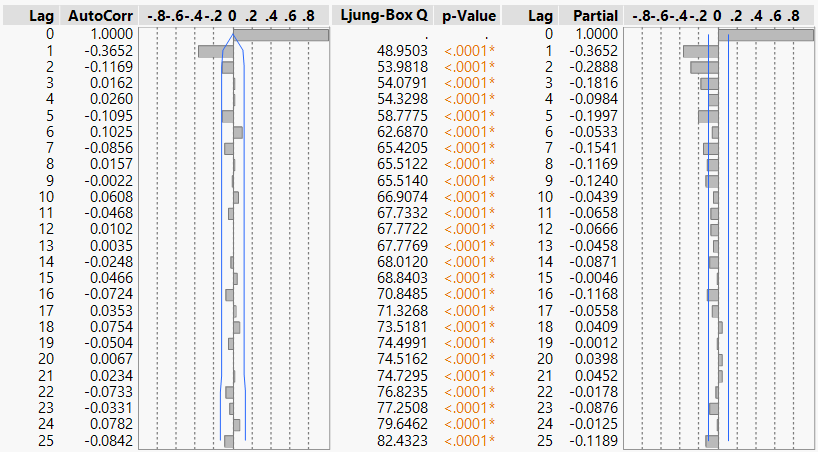
**Time Series plot for logcnt**



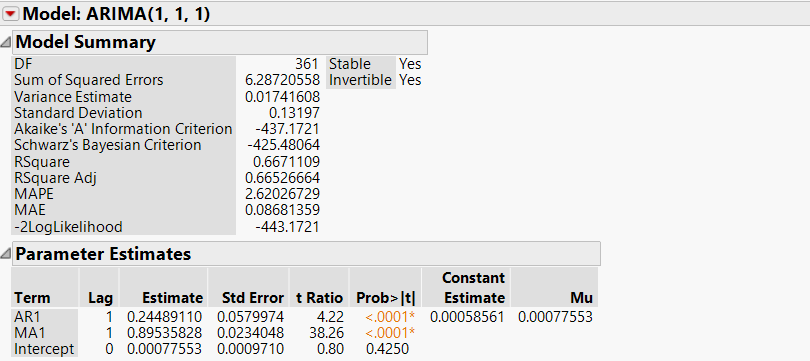
**Difference order 1**



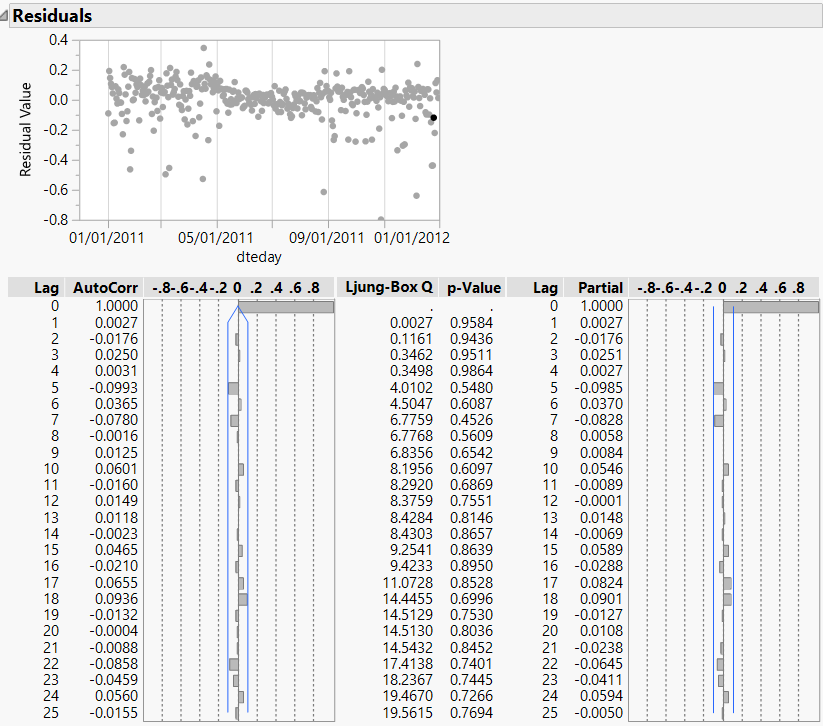
**ACF and PACF Plot**



**ARIMA Model Summary**



**Residual Plot**

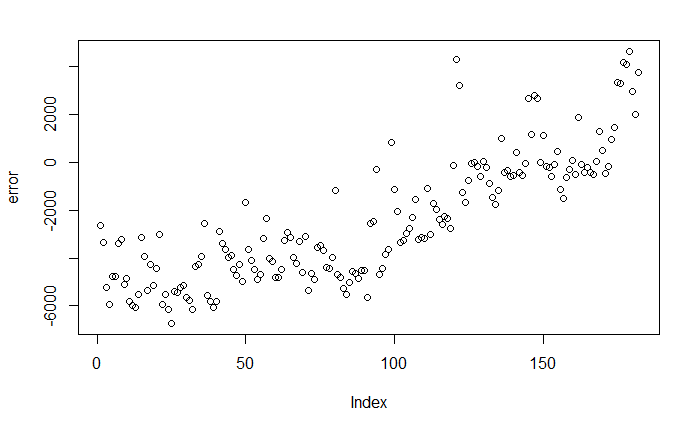
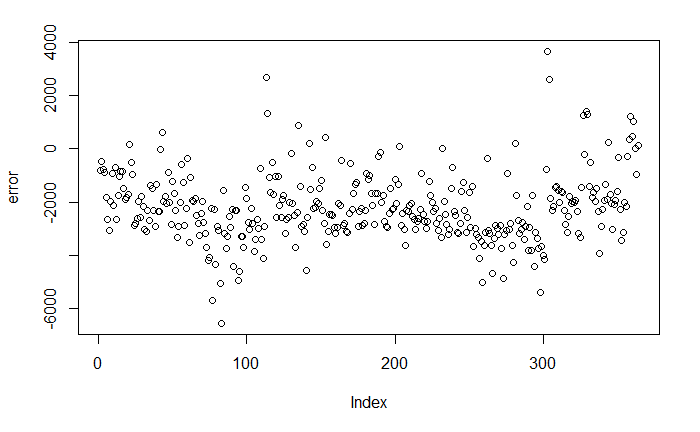


## Model Performance

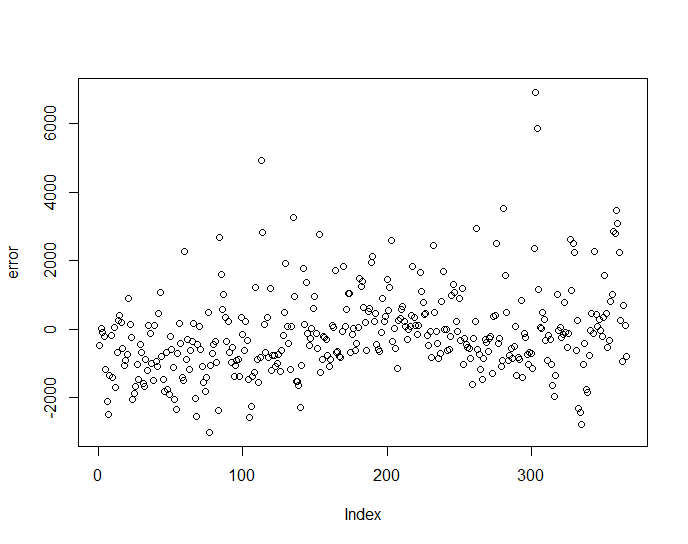
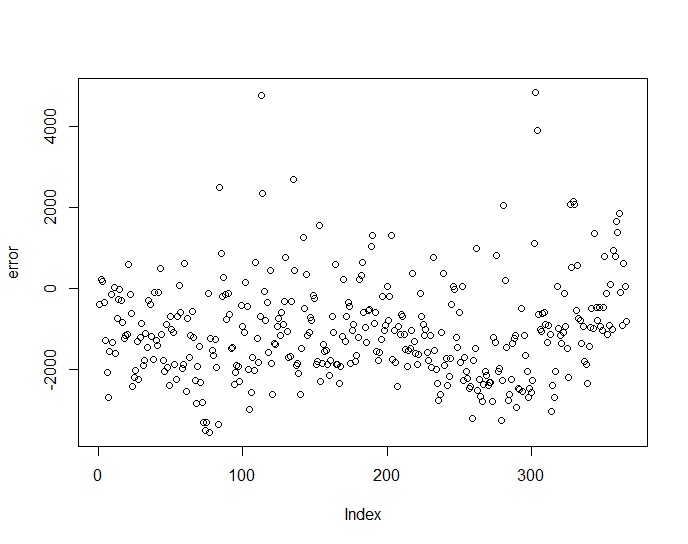
Below table and plot shows the error plot for year 2012:

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Type | RMSE | MAE |
| Naive | Default |  |  |
| Decision Trees | Default Variables | 2549 | 2310 |
| Decision Trees | Weather 5 days moving average+1.5 years train data | 3575 | 3044 |
| ARIMA | 7 day Moving average | 95.35284 | 70.26753 |
| GLM |  | 1275.46 | 934.6744 |
| Random Forest | 7 day Moving average | 1616.523 | 1377.843 |
| NNet | 7 day Moving average | 1877.859 | 1611.204 |
| Ensemble (Stacking) | 7 day Moving average | 1237.905 | 934.270 |

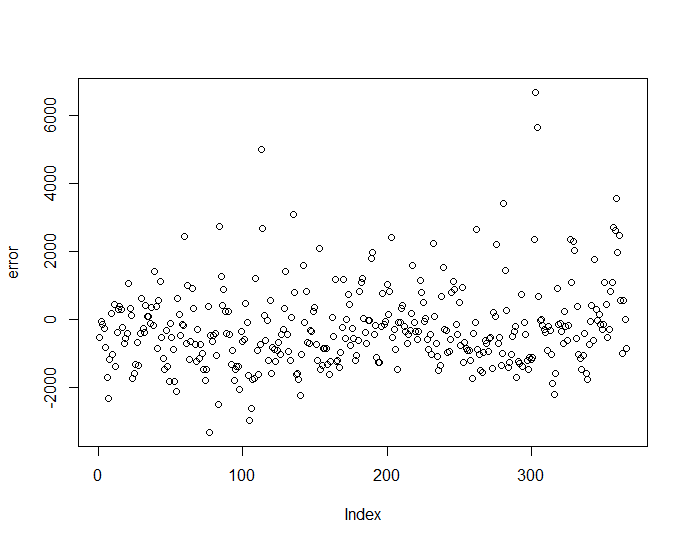
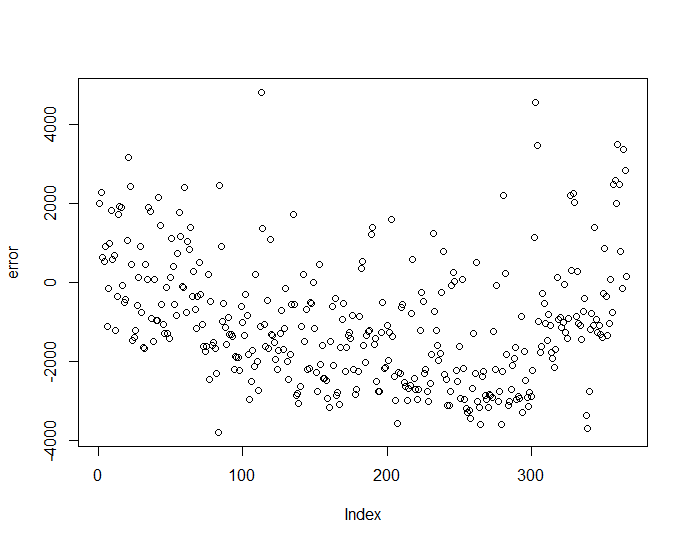
**Fig 4. Error plot (Test Data) for Decision Tree Prediction over time**



**Fig 5 Error Plot GLM Fig 6 Error plot for Random Forest**

**Fig 5 Error Plot NNet Fig 6 Error plot for Ensemble (Stacking)**



## Business Performance

Below shows the return on investment comparisons of different models.

**Table 2. Time Series Model Performance**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Type | Total Spend(,000) | Profit/Loss(,000) | ROI(%) |
| Naïve (Default) | Default | $4092 | $1442 | 35.23 |
| Ensemble (Stacking) | 7 days moving averages for input variables  Target variable: Registered & Casual  GLM+KNN+Rpart | $3919 | **$1582** | 40.0 |
| GLM | 7 days moving averages for input variable  Target variable: registered & Casual | $4007 | $**1560** | 38.93 |
| Random Forest (Bagging) | 7 days moving averages for input variables | $3315 | $**1482** | 44.70 |
| NNet | 7 days moving averages for input variables  Target variable registered and casual | $3111 | $1342 | 43.13 |
| Decision Tree | Default Variables | $2485 | $1187 | 47.76 |
| Decision Tree | Weather 5 days moving average+1.5 years train data | $1327 | $498 (6 months) | 37.5 |
| ARIMA (1,1,1) | Univariate time series analysis over total count | $2340 | $1031 | 44% |

**Business Insights**

* The above Table gives the model profit for 2012 expressed as a $ total and profit expressed as a percentage of total expenditure (total costs for the year).
* First row of the above table gives the profit (total and percentage of expenditure) for the default prediction.
* Our best performing model, Ensemble (stacking) always generted the better predictions in terms of Profit and ROI. We compared it between 3-5 dollars. However, some of our models like decision trees shown above never generated profit better than default.
* We didn’t observe any correlation between our model performance with respect to season or similar factor. However for odd extreme values, model tends to follow irregular pattern
* No. it didn’t show any deviation for test dataset. Our RMSE plot shown below indicates random pattern. Our best performing Ensebmle stacking showed a constant error performance for year 2012
* For our best performing models, 1 year train model performed best in terms of total profit. Below are the numbers when trained for 1 year and 1.5 years:

|  |  |  |  |
| --- | --- | --- | --- |
| Model Type | Training Period | Test Period | Profit (,000) |
| GLM | 1 year | Jul2012-Dec2012 | $847 |
| GLM | 1.5 years | Jul2012-Dec2012 | $579 |
| Ensemble | 1 year | Jul2012-Dec2012 | $862 |
| Ensemble | 1.5 years | Jul2012-Dec2012 | $587 |

* We didn’t perform the data balancing but simply by looking at the 18 months’ data we can see that 2/3rd of the data shows uptrend (Jan-Dec) and 1/3rd of the data (Jul-Dec) is showing downtrend. Hence model may be biased towards uptrend. Thus, the overall profit might go down due to Over-Prediction. We could use give high weightage to Jul-Dec data. Alternatively, suggested approach would be to continuous build the data as mentioned in report at 4pm every day.

## References

Lecture Notes: *ARIMA/Decision Trees/GLM*