Bike sharing Demand Forecasting

DATA MINING II PROJECT

NITISH GHOSAL | MAGGIE LEDBETTER | ADRIAN VALLES

Problem Background

Bike sharing systems are new generation of traditional bike rentals where whole process from membership, rental and return has become automatic. Through these systems, users can easily rent a bike from a particular location and return it at another location. Currently, there are about over 500 bike-sharing programs around the world which is composed of over five hundred thousand bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues.

Apart from interesting real-world applications of bike sharing systems, the characteristics of data being generated by these systems make them attractive for the research. Opposed to other transport services such as bus or subway, the duration of travel, departure and arrival position is explicitly recorded in these systems. This feature turns bike sharing system into a virtual sensor network that can be used for sensing mobility in the city.

About Capital Bikeshare

Capital Bikeshare is metro DC's bikeshare system, with more than 3,700 bikes available at 440 stations across five jurisdictions: Washington, DC; Arlington, VA; Alexandria, VA; Montgomery County, MD; and Fairfax County, VA. Capital Bikeshare provides residents and visitors with a convenient, fun and affordable transportation option for getting from Point A to Point B.

Capital Bikeshare, like other bikeshare systems, consists of a fleet of specially designed, sturdy and durable bikes that are locked into a network of docking stations throughout the region. The bikes can be unlocked from any station and returned to any station in the system, making them ideal for one-way trips. People use bikeshare to commute to work or school, run errands, get to appointments or social engagements and more.

Capital Bikeshare is available for use 24 hours a day, 7 days a week, 365 days a year. Riders have access to a bike at any station across the system.

Problem Definition

- **Regression**
  Through this project we aim to combine historical usage patterns with weather data in order to forecast bike rental demand in the Capital Bikeshare program in Washington, D.C. on any given day.
- **Time series analysis**
  Given the daily bike demand for 2011 and 2012, we will forecast the 2013 bike demand through an univariate time series analysis using a seasonality factor.

Data Description

The dataset contains the hourly and daily count of rental bikes between years 2011 and 2012 in Capital bikeshare system with the corresponding weather and seasonal information. Both hour.csv and day.csv have the following fields, except hr which is not available in day.csv

instant: record index
dteday : date
season : season (1: spring, 2: summer, 3: fall, 4: winter)
yr : year (0: 2011, 1:2012)
mnth : month (1 to 12)
hr : hour (0 to 23)
holiday : whether day is holiday or not (extracted from [Web Link])
weekday : day of the week
workingday : if day is neither weekend nor holiday is 1, otherwise is 0
weathersit :
1: Clear, Few clouds, Partly cloudy, Partly cloudy
2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
temp : Normalized temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-8, t_max=+39 (only in hourly scale)
atemp: Normalized feeling temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-16, t_max=+50 (only in hourly scale)

hum: Normalized humidity. The values are divided to 100 (max)
windspeed: Normalized wind speed. The values are divided to 67 (max)
casual: count of casual users
registered: count of registered users
cnt: count of total rental bikes including both casual and registered

Methodology

For demand forecasting of bikes per hour, we have built models in R using various machine learning algorithms such as GLM, GAM, decision trees, bagging, random forest, boosting and boosting and compared the results to come up with the best performing model.

We split the dataset into training and test samples in the 80:20 ratio, built models on training dataset, and measured mean square error (MSE) and mean squared prediction error (MSPE) on training and testing dataset, respectively, to compare model performances. We then conducted a similar analysis in SPSS Modeler to compare overall findings.

Finally, we conducted time series analysis in SAS to predict daily bike demand for 2013.

Exploratory data analysis

The starting point of our exploratory data analysis involved understanding the dataset. The dataset is composed of 17379 observations and 17 variables where each observation computes the count of rental bikes in one hour. 8 variables are categorical while the rest are continuous variables. **Table 1** shows a summary statistic of the continuous variables.

	temp	atemp	hum	windspeed	registered	casual	cnt
Mean	0.5	0.48	0.63	0.19	153.79	35.68	189.46
SD	0.19	0.17	0.19	0.12	151.36	49.31	181.39
Median	0.5	0.48	0.63	0.19	115	17	142
Min	0.02	0	0	0	0	0	1
Max	1	1	1	0.85	886	367	977
N_Obs	17379	17379	17379	17379	17379	17379	17379
N_NA	0	0	0	0	0	0	0

Table 1- Summary statistics of the continuous variables

In order to show the distribution of the main categorical variables we decided to plot bar plots of each categorical variable to understand how the levels within each

level behave. **Figure 1** below shows this, where the horizontal line in each plot represents the average number of observations for each variable.

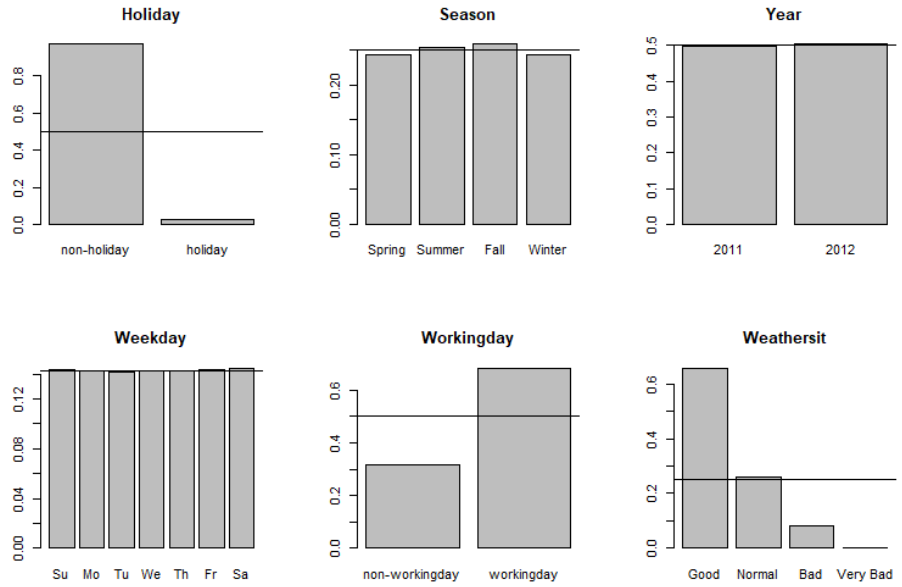


Figure 1 – Bar plots of the categorical variables

Once we understood the general distribution of the variables, we took a deeper look into the response variable of our study, "cnt", which computes the count of rental bikes at each hour. **Figure 2** shows the histogram of this variable. It is interesting to see how the number of observations decrease exponentially with respect to the "cnt" variable.

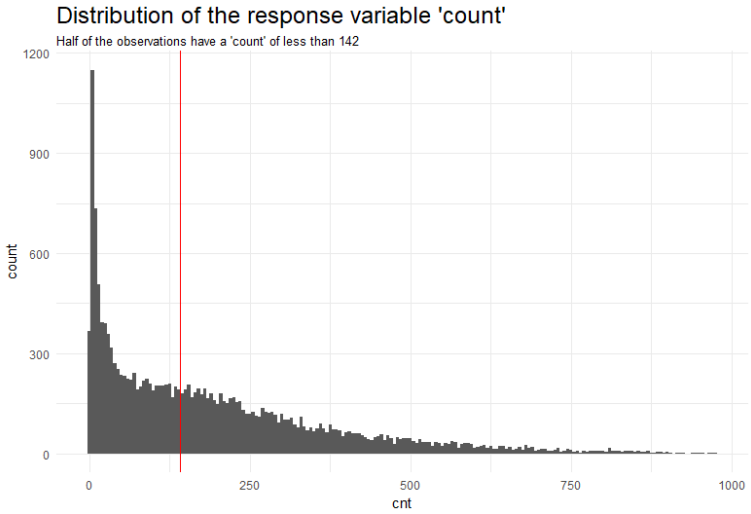


Figure 2- Histogram of the response variable "cnt"

With a better understanding of the of the variable we wanted to predict, we took a look at some of visualizations of the predictor variables. **Figure 3** shows the count of bike rentals by hour and season. Logically, bike rentals are more popular during the day from 8a.m to 8p.m and during the Fall and Summer.

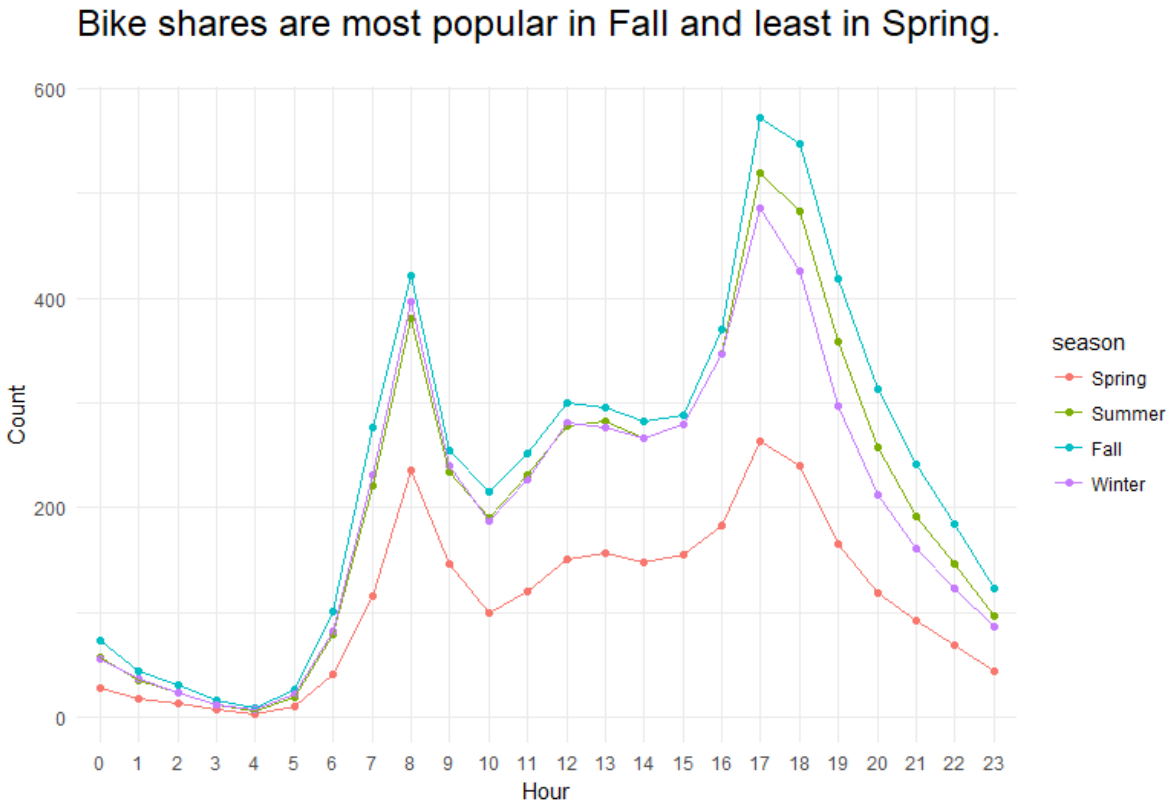


Figure 3- Bike rentals by hour and season

We might be able to explain the popularity of the bike rentals during Summer and Fall because of the good weather associated to these season, but, does the weather really influence the number of bike rentals? As **Figure 4** shows, bike rentals are more popular with good weather conditions. By good weather conditions we understand Clear, Few clouds, Partly cloudy or Partly cloudy. Further description for the type of weather conditions can be found in the data description section under the weathersit variable.

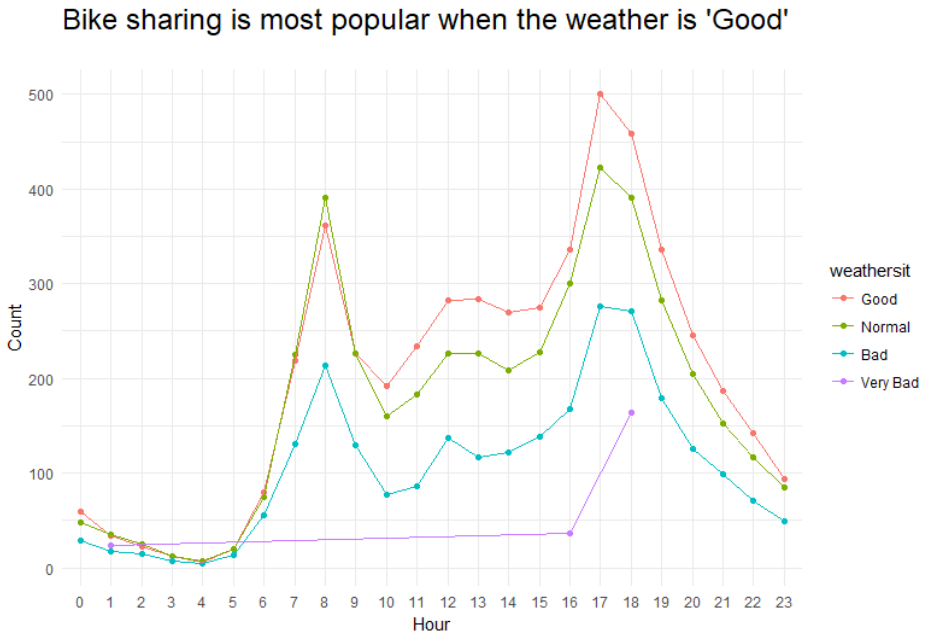


Figure 4- Bike rentals by hour and weather

However, to explain the popularity of the bike rentals during the Fall, we should demonstrate that Fall is associated with good weather conditions. By taking into consideration to the "weathersit" variable, we can see, as shown in **Figure 5,** that the Fall is the season with the highest probability of good weather. Therefore, now we understand why the Fall is the season with the highest number of bike rentals.

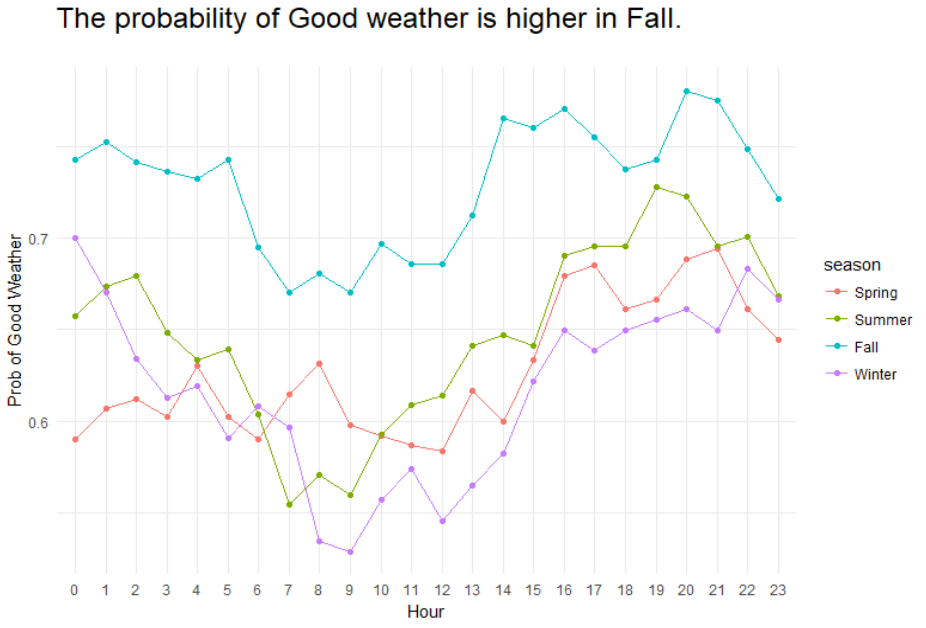


Figure 5-Probability of "Good" weather by hour and season

Now that we have seen how the number of rental bikes is influenced by the weather and therefore by the seasons, let's take a look at the distribution of bike rentals by day of the week with **Figure 6.**

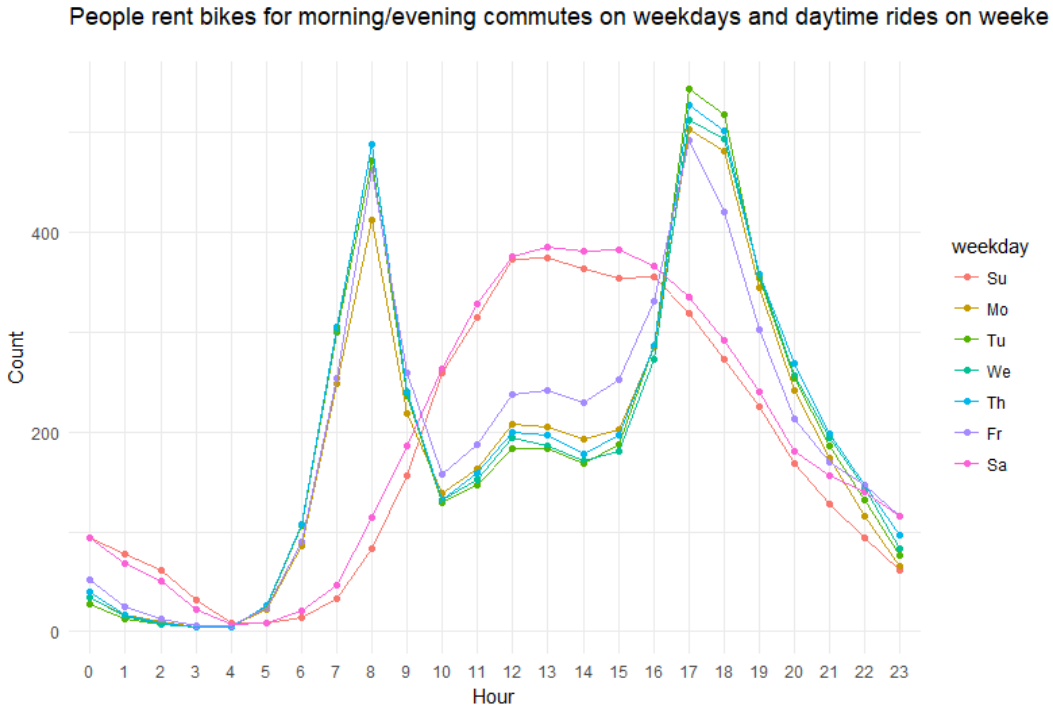


Figure 6-Count of bike rentals by hour and day of the week

Bike rentals are more popular for morning/evening commutes on weekdays and daytime rides during the weekends. The reason behind this could be that people renting the bikes during weekdays use them to get to work, given that the peaks are found at 8a.m and 5p.m, which are popular times to enter and leave work. On the other side, it seems that during the weekends, the rentals are more likely made by city tourists given the time of the day.

In order to better understand if people rent bikes to get to work, we have decided to divide the study of bike demand between casual and registered users. Given that people that rent bikes to get to work are more likely to be registered users in order to benefit from discounts, we can see in **Figure 7** how registered users rent the bikes during commute times. On the other side, the bike demand for casual users is not aligned with commute times as the people renting these bikes are usually city visitors. We can therefore see how bike demand during weekdays is more likely to be by registered users during commute times, and demand during weekends is usually by casual city visitors during the day.

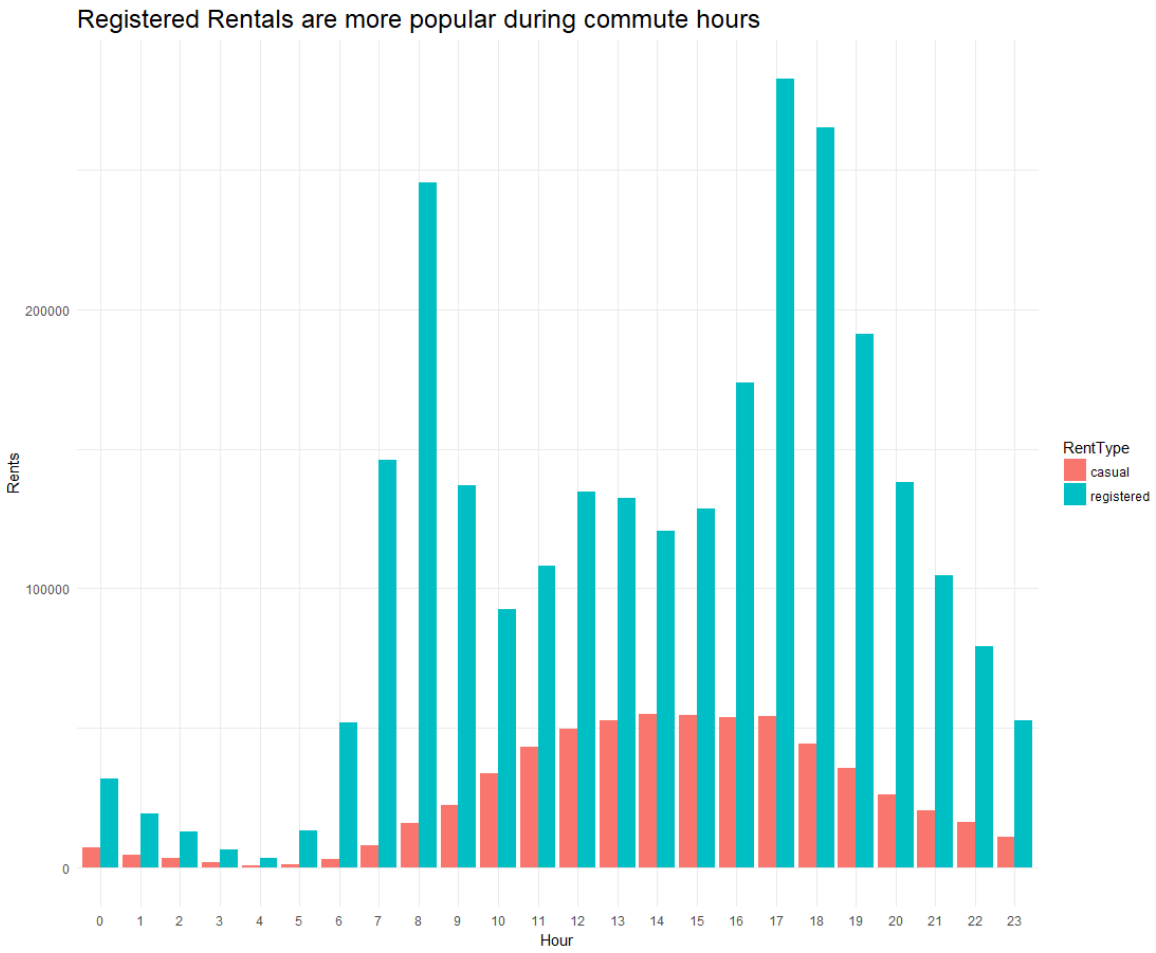


Figure 7- Bike rentals by temperature and by workingday type

DATA PREPARATION FOR MODELING

Before starting with the modeling stage, we needed to make some adjustments to the data in order to simplify the interpretation of the models. For this, we have first decided to group some variables:

- The 'hour' variable has been grouped into 4 levels: "Night"," morning", "afternoon" and "evening".
- The 'weekday' variable has been converted into a 0,1 binary variable depending if the observation is a "weekend" or a "weekday" respectively.

Similarly, we decided to get rid of certain variables that are not meaningful for our regression study. The following variables were deleted for the modeling stage:

- 'instant': It is just the record index of the observations

- ‘dtday’: as the date doesn't seem to have any impact on prediction given that we already have ‘year’, ’month’, ’weekday’, and ‘hour’ variables…
- ‘temp’: as we already have ‘atemp’ which is same but normalized variable
- ‘casual’ and ‘registered’ because their sum is just our response ‘cnt’ variable
- ‘month’: as we have 12 levels that we were going to group…However, we already have a season variable, which is just the variable month grouped.

Finally, we split the data into 80% for training purposes and 20% for testing purposes.

REGRESSION ANALYSIS-MODELING

Before we dug into the regression analysis, we needed to briefly understand the correlation matrix between the continuous variables. As shown in **Figure 8**, the temperature variable is positively correlated with the number of bike rentals, while humidity looks to be negatively correlated with it.

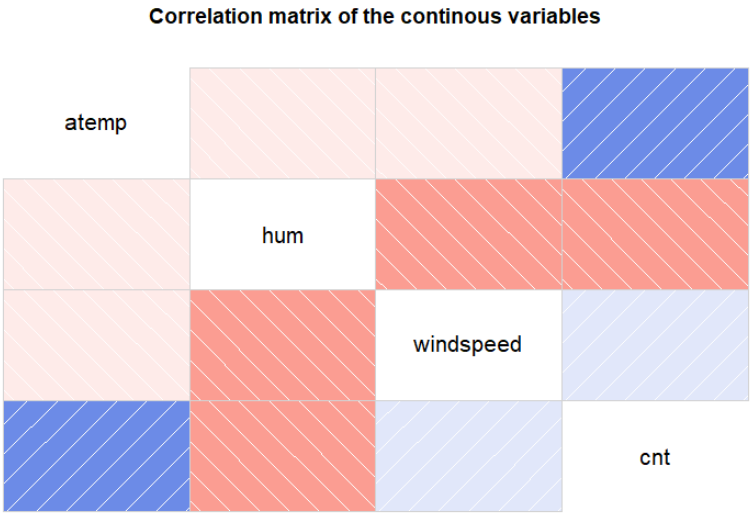


Figure 8-Correlation scores of the variables

Generalized Linear Regression model (LASSO)

To start our regression analysis, we started the study with a generalized linear regression model. We used LASSO for our variable selection. **Figure 9** shows the behavior of the coefficients with respect to lambda.

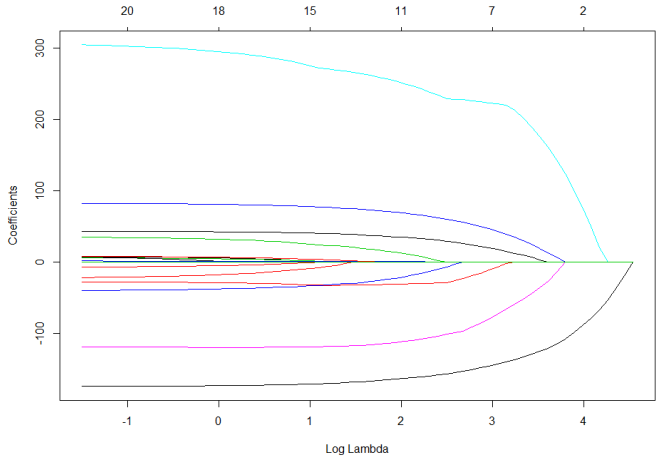


Figure 9 - LASSO variable selection

We then needed to select the optimal tuning parameter lambda. To do this, we performed cross-validation. We chose the λ parameter that is within 1 standard error of the smallest cv error. We chose this λ parameter instead of the one that gives the smallest mean-squared error in order to get a simpler model and avoid overfitting. **Figure 10** shows the MSE associated with each λ parameter where the dotted line represents the λ parameter that is within 1 standard error of the smallest cv error.

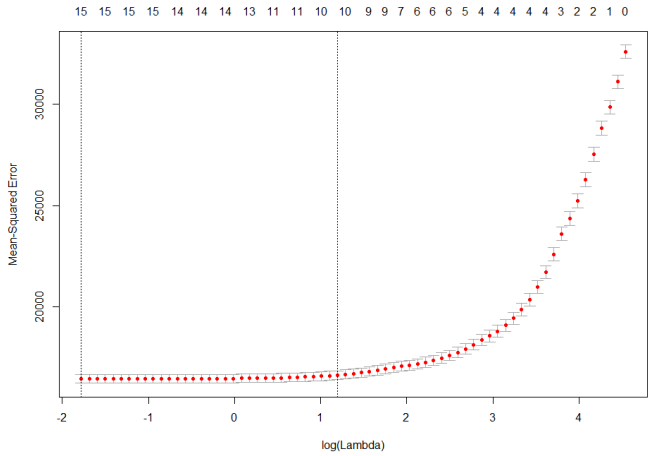


Figure 10 – Tuning parameter selection

Given this tuning parameter, the error values associated with this GLM model are an **in sample mean squared error** equal to 16673.05 and an **out of sample mean squared error** equal to 17713.25.

Regression Tree

We wanted to start our analysis with a large tree, so we could prune it and get an optimal one. Therefore, we decided to choose the default cp value, which is equal to 0.001, to make sure that the tree could be pruned. A large tree is seen in **Figure 11.**

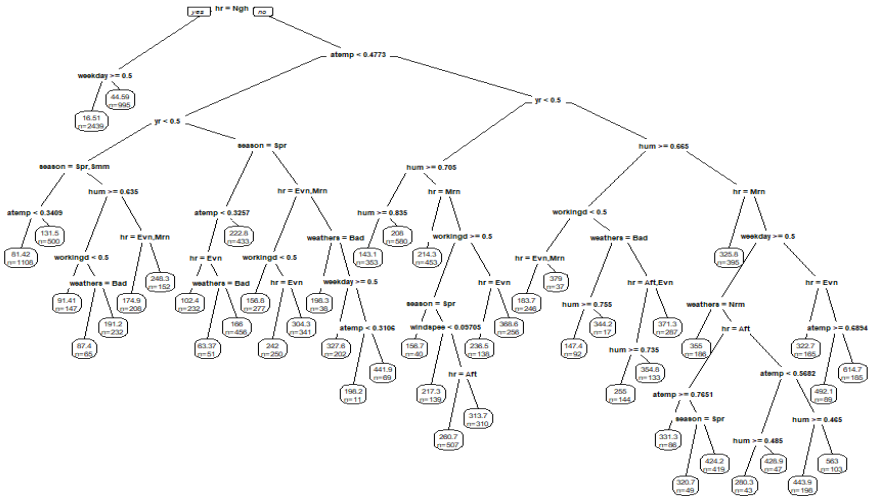
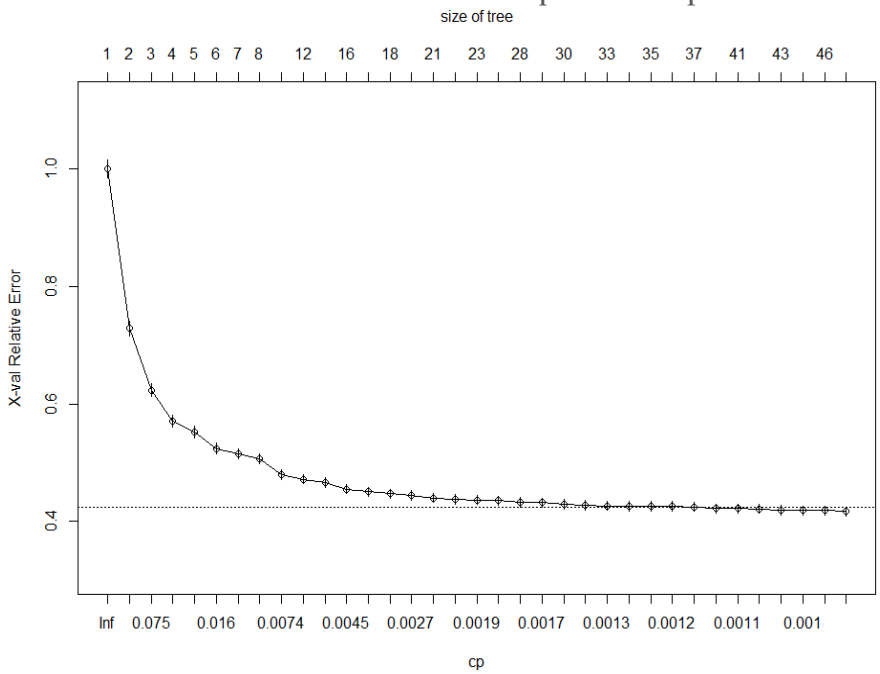


Figure 11– Large regression tree for bike prediction demand

We wanted to prune this tree because it might lead to overfitting and therefore a high out of sample error. Shown below, in **figure 12,** there is a plot and a table of the relative error associated with each potential cp value.



	CP	nsplit	rel error	xerror	xstd
1	0.271596	0	1	1.0003	0.015601
2	0.105339	1	0.7284	0.72871	0.012883
3	0.052846	2	0.62306	0.62336	0.011241
4	0.024134	3	0.57022	0.57047	0.009849
5	0.023405	4	0.54608	0.55202	0.009456
6	0.011608	5	0.52268	0.52424	0.008932
7	0.009282	6	0.51107	0.5151	0.008705
8	0.008462	7	0.50179	0.50683	0.008595
9	0.006496	10	0.4764	0.48054	0.008453
10	0.005294	11	0.46991	0.47216	0.008398

Figure 12- cp score vs relative error

Ideally, we should select the highest cp-value corresponding to a relative error below the dotted line in **figure 12**, which represents one s.d above the lowest relative error. However, we can see that after 10 splits the relative error decreases very slow. Consequently, we decided to select the cp value that corresponds with 10 splits in order to avoid complex models. This pruned tree is shown in **Figure 13.**

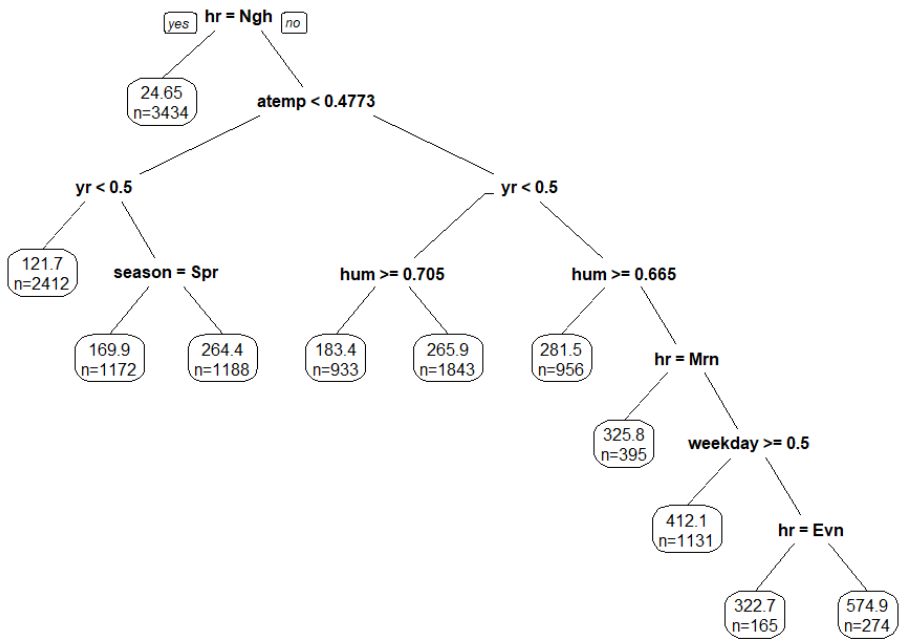


Figure 13-Pruned Tree

This pruned tree has a **in-sample** mean squared error equal to 15534.63 and an **out of sample** mean squared error equal to 16778.2, showing a better performance than the GLM.

Baggine

While bagging employs the idea of bootstrap, its purpose is not to study bias and standard errors of estimates. Instead, the goal of Bagging is to improve prediction accuracy. It fits a tree for each bootstrap sample, and then aggregate the predicted values from all these different trees.

In order to understand how many trees we needed to run to minimize the mean squared error, we plotted number of trees versus error. **Figure 14,** shows the mean squared error by number of trees.

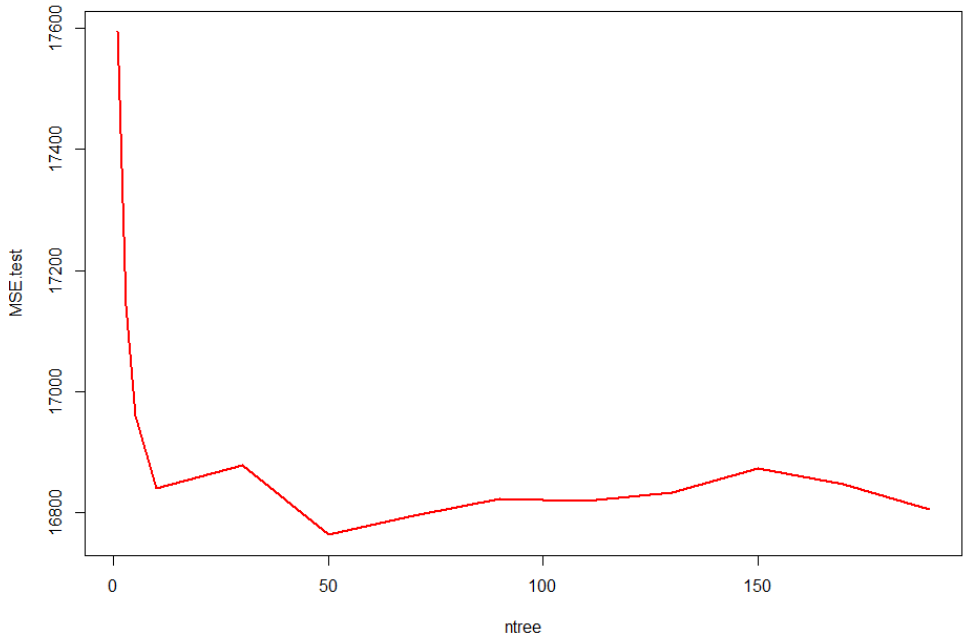


Figure 14- Number of trees by out of sample mean squared error

Therefore, it seems that 50 trees should give the best results. The values associated with bagging 50 trees are an **in sample** mean squared error equal to 15819.39 and an **out of sample** mean square error equal to 16817.87.

Random Forest

Random forest is an extension of Bagging, but it makes significant improvement in terms of prediction. The idea of random forests is to randomly select mm out of pp predictors as candidate variables for each split in each tree. **Figure 15** below shows the importance of each variable in this random forest model given 500 tress. The variable *hr* is the most selected variable followed by *year* and *hum*.

	%IncMSE	IncNodePurity
season	4801.795	26740162
yr	5250.528	30657827
hr	21321.73	129034550
holiday	92.64688	1033935
weekday	1929.792	6476476
workingday	2258.915	7235098
weathersit	1533.161	10066895
atemp	10650.2	75583877
hum	5040.047	46919164
windspeed	974.6945	21509380

Figure 15-Importance of each variable in the random forest

Figure 16 below shows the out of sample mean squared error associated with the potential number of trees that we can use to run the model.

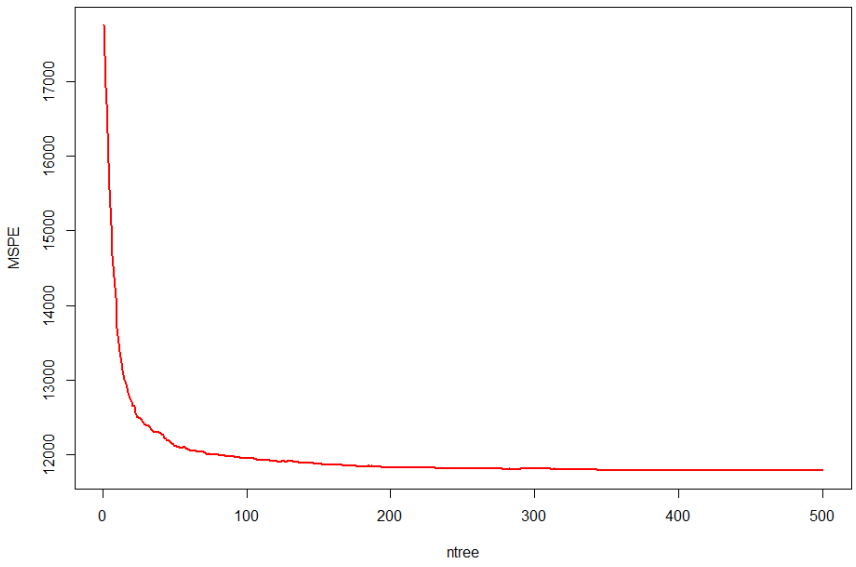


Figure 16-MSPE by number of trees

Given that the MSPE decreases as long as the number of trees increase, we ran 500 trees in order to reduce the MSPE.

The error values associated with random forest are an **in sample** mean squared error equal to 6695.373 and an **out of sample** mean square error equal to 12781.75

While predicting errors improve significantly with respect to previous models (around a 60% in MSE and 25% in MSPE), we also have to say that random forest loses interpretability with respect to linear regression and regression tree, as there is not an explicit form of the model that we can write down.

Boosting

Boosting builds a number of small trees, and each time, the response is the residual from last tree. It is a sequential procedure. **Figure 17** shows the out of sample mean squared error associated with each of the potential number of trees that we can use to run the model.

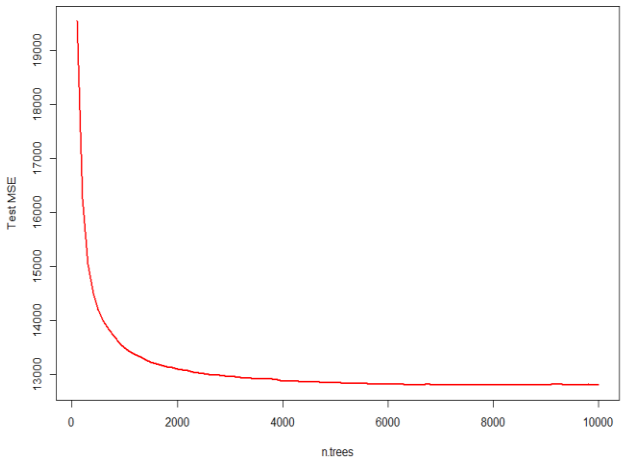


Figure 17- MSPE by number of trees

Given that the higher the number of trees the lower the MSPE, we chose 10000 trees for boosting. Therefore, as it happened in random forest, we can calculate the relative influence of each variable in the final model given 10000 iterations. Variable *hr* is again the most influential variable as it is shown in **figure 18,**

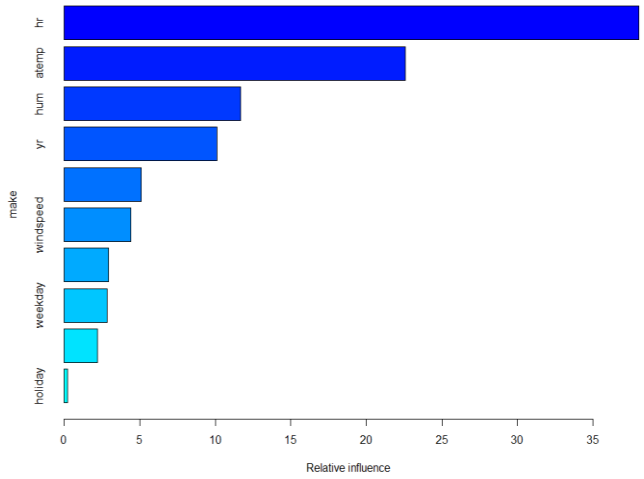


Figure 18-Relative influence of each of the variables for boosting

The error values associated with boosting are an **in sample** mean squared error equal to 9895.225 and an **out of sample** mean square error equal to 12797.27.

While the in-sample error is a little higher than Random forest, the out of sample error, which is the one that we care about, is basically the same than random forest. We can conclude from this that both Random forest and boosting are equally useful in predicting power, although Random forest might be less computational intensive.

Generalize additive model (GAM)

Before fitting a general additive model to the data, we needed to understand the type of data that we are dealing with. Season, year, hr, holiday, weekday, workingday and weathersit are not continuous variables and therefore don't need to be modelled as a smoothing parameter, as opposed to all the other variables in our initial analysis. From an initial analysis, we get the following summary of the model as shown in **figure 19**.

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	202.772	4.161	48.728	< 2e-16	***
seasonSummer	29.365	3.936	7.461	9.09E-14	***
seasonFall	7.934	4.944	1.605	0.108525	
seasonWinter	67.587	3.503	19.294	< 2e-16	***
yr	83.032	2.189	37.937	< 2e-16	***
hrEvening	-46.74	3.205	-14.584	< 2e-16	***
hrMorning	-45.392	3.372	-13.46	< 2e-16	***
hrNight	-218.123	3.596	-60.66	< 2e-16	***
holiday	-15.995	4.366	-3.663	0.00025	***
weekday	-6.771	2.563	-2.642	0.008252	**
workingday	9.224	2.414	3.821	0.000133	***
weathersitNorm	-7.529	2.659	-2.831	0.004645	**
weathersitBad	-48.431	4.625	-10.471	< 2e-16	***
weathersitVery	15.241	73.261	0.208	0.835204	
	edf	Ref.df	F	p-value	
s(atemp)	6.969	7.904	147.743	< 2e-16	***
s(hum)	6.467	7.541	37.713	< 2e-16	***
s(windspeed)	3.028	3.794	6.662	4.80E-05	***

Figure 19- Summary GAM model

From the summary above, given that the variables that were modeled as smoothing parameters have high edf values, we can be certain that they should be modeled as smoothing parameters. On the other hand, given that the non-significant variables are just a single category of the entire variable, we can just leave the variable in the model unsmoothed.

Therefore, the error values associated with this GAM model are an **in sample** mean squared error equal to 15993.74 and an **out of sample** mean square error equal to 16950.79. Even if they don't report a good performance compared to Random forest and bagging, it still improves the results from the GLM model.

MODEL COMPARISON

Below, in **Figure 20**, we can see the performance for all the models evaluated.

	mean squared error	
	In sample	Out of sample
Linear Regression	16673.05	17766.12
Regression tree	15534.63	16778.2
Bagging	15819.39	16817.87
Random forest	6695.373	12781.75
Boosting	9895.225	12797.27
GAM	15993.74	16950.79

Figure 20-Model comparison

From the table above, we can conclude that both random forest and bagging are the models that predict bike demand with better accuracy. Given that boosting is more computational intensive than random forest, random forest is more desired. In case we are concerned about interpretability, the regression tree is a good option even though if it has less predicting power.

SPSS MODELER COMPARISON

We completed a similar regression analysis using SPSS Modeler to test whether our conclusions are the same. We began by using the auto-numeric modeler using MSE and relative error as ranking criteria to determine which models ranked best in performance.

Models included in this analysis were:

- CART
- Bagging
- Boosting
- Random Forest
- CHAID Regression Tree
- Neural Net
- GLM
- GAM
- Logistic Regression

Overall, our conclusion remains that Random Forest and Boosting are the most accurate models, and that hour and temperature are the two most important variables.

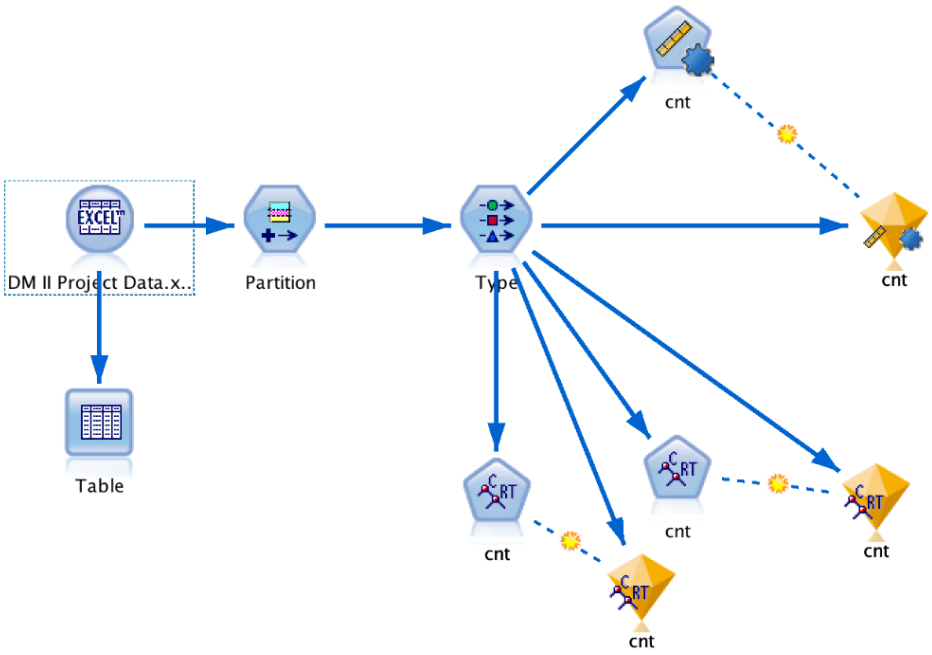


Figure 21 – SPSS Modeler Stream

Similar to modeling done in R, results revealed that Random Forest and Boosting were the optimal models, as seen in **Figure 22.**

File Generate View Preview

Model Graph Summary Settings Annotations

Sort by: Relative error Ascending Descending Delete Unused Models View: Testing set

Use?	Graph	Model	Build Time (mins)	Correlation	No. Fields Used	Relative Error
☑		XGBoost Tree 1	< 1	0.784	10	0.389
☑		Random Trees 1	< 1	0.777	10	0.398
☑		Neural Net 1	< 1	0.773	10	0.403
☑		Tree-AS 1	< 1	0.745	9	0.445
☑		CHAID 1	< 1	0.718	8	0.485
☑		C&R Tree 1	< 1	0.71	10	0.496
☑		Linear 1	< 1	0.7	8	0.510

Figure 22 – SPSS Model Ranking on Relative Error

Next, we specifically looked at CART modeling to determine if SPSS Modeler would output similar large and pruned trees, and if relative importance of variables was similar. We found that the trees themselves were somewhat different, as seen in **Figure 23**, but the rank order of variable importance was similar, with hour,

temperature, and year rising to the top, as seen in **Figure 24.** However, one major difference of the methods is that in SPSS Modeler, the humidity variable was much less important to the model than in R.

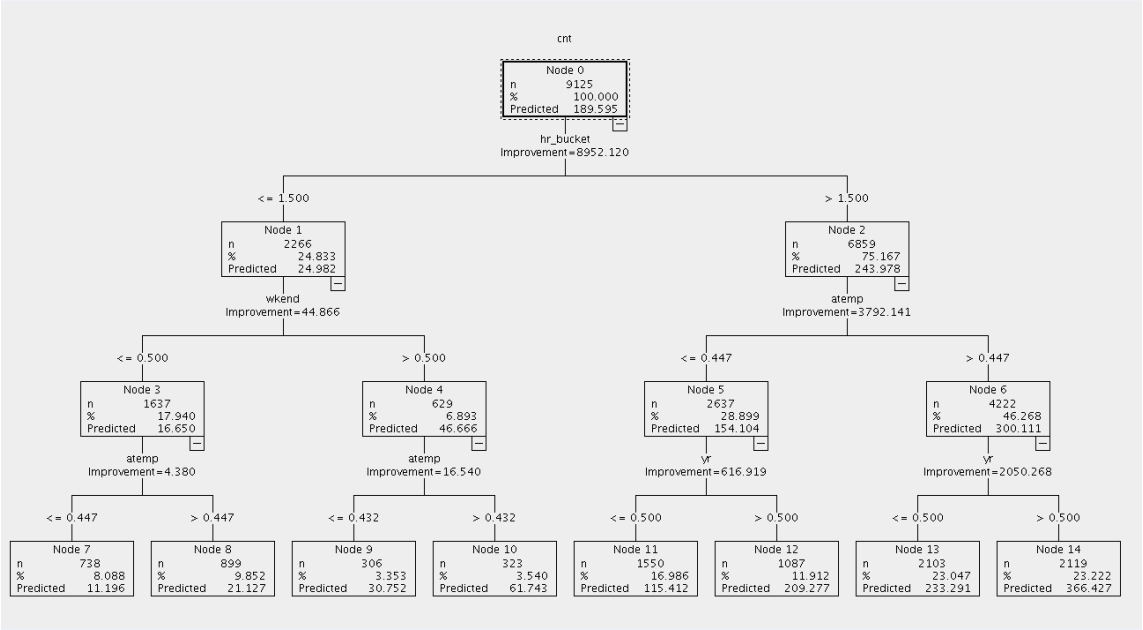


Figure 23 – SPSS Modeler Pruned Tree

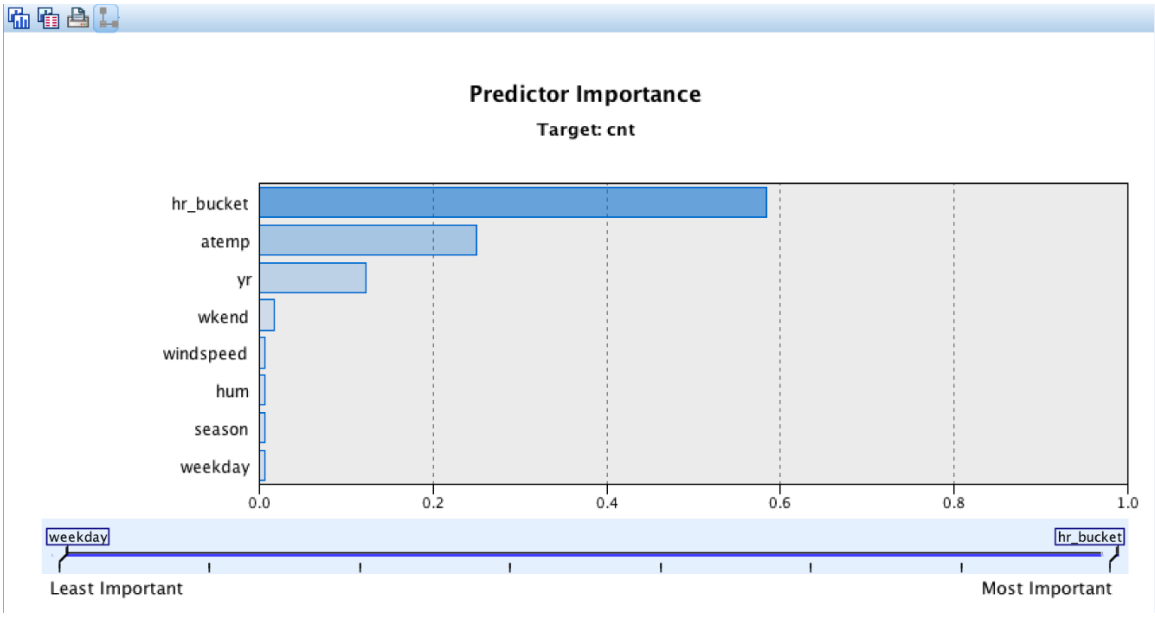


Figure 24 – SPSS Modeler Variable Importance

TIME SERIES ANALYSIS (SAS)

Given that we have the bike demand data for the last 2 years, we have decided to do a univariate time series analysis that will allow us to forecast the bike demand for next year (2013). For this analysis, we have also decided to aggregate the hourly data into a daily basis, in order to stabilize the forecasts. **Figure 25** shows the daily demand for bikes for the last 2 years (top left) as well as the ACF and PACF of the observations.

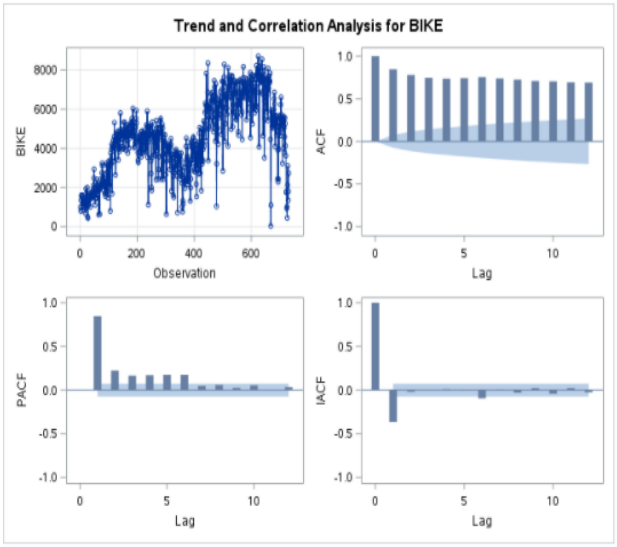


Figure 25 -Time series analysis

It seems pretty obvious that this data isn't stationary given the non-constant mean. Also, given that the ACF fails to decay, we are going to take first order difference with respect to 365 days ago, given that the data follows a seasonal pattern given that we have the bike demand for 2011 and 2012. The summary for this first order difference can be seen below in **Figure 26.**

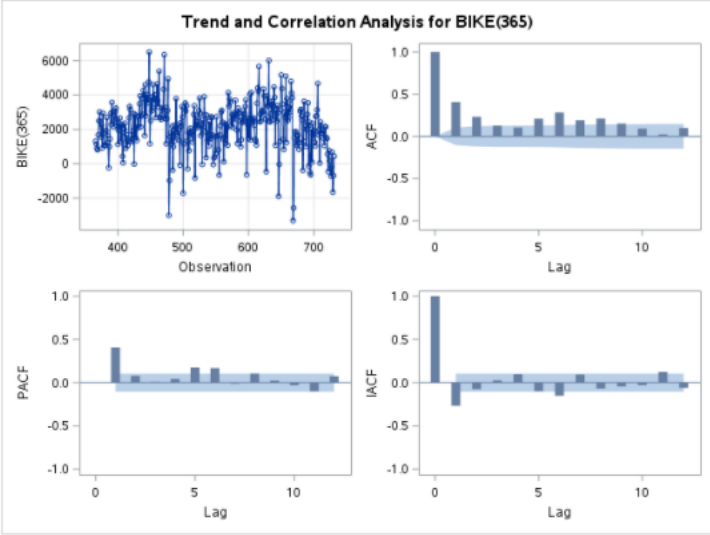


Figure 26 -Analysis of the 1sr order difference

Even if it has improved and the data is more stationary, we will take second order difference to make sure that we have zero mean and no trend.

Figure 27 below shows the analysis for the second order difference, which is stationary as given by the Dickey Fuller test (p-values for zero mean, single mean and no trend are lower than 0.05). We can therefore conclude that the data is finally stationary.

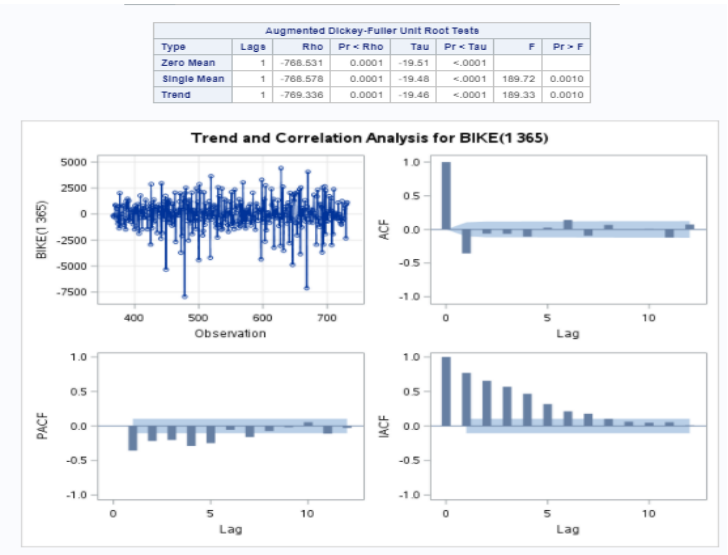


Figure 27 -Analysis for the 2nd order difference

Given that the ACF is truncated after lag equal to 1 and the PACF could be either truncated after 1 or slowly decaying, our guess is that the process can be modeled with an ARIMA (1,2,1) or an ARIMA(0,2,1).

After comparing the model diagnostics of both models, ARIMA (1,2,1) looks like it fits the data better. The model diagnostics for this model is shown in **Figure 28** .

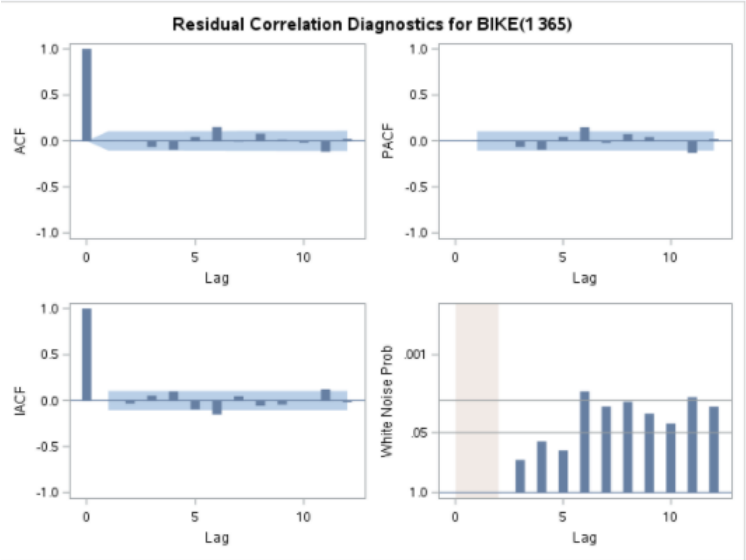


Figure 28 - Analysis for ARIMA (1,2,1)

Given that the ACF and the PACF of the model ARIMA (1,2,1) performs like white noise, we can conclude that the model is appropriate. Moreover, the estimate

parameters are significant as shown in **Figure 25** below. The estimate for the MA is 0.91 and the estimate for the AR parameter is 0.26.

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MA1,1	0.91406	0.02607	35.06	<.0001	1
AR1,1	0.25544	0.05987	4.27	<.0001	1

Figure 25- Model estimates for ARIMA (1,2,1)

Given this model we can get a forecast for the demand of next year (2013) by day as well as a 95% Confidence interval for each day. This is shown in **figure 29.** It is interesting to see how the seasonal patter affects the data as bike demand is forecasted to be higher during Summer and Fall.

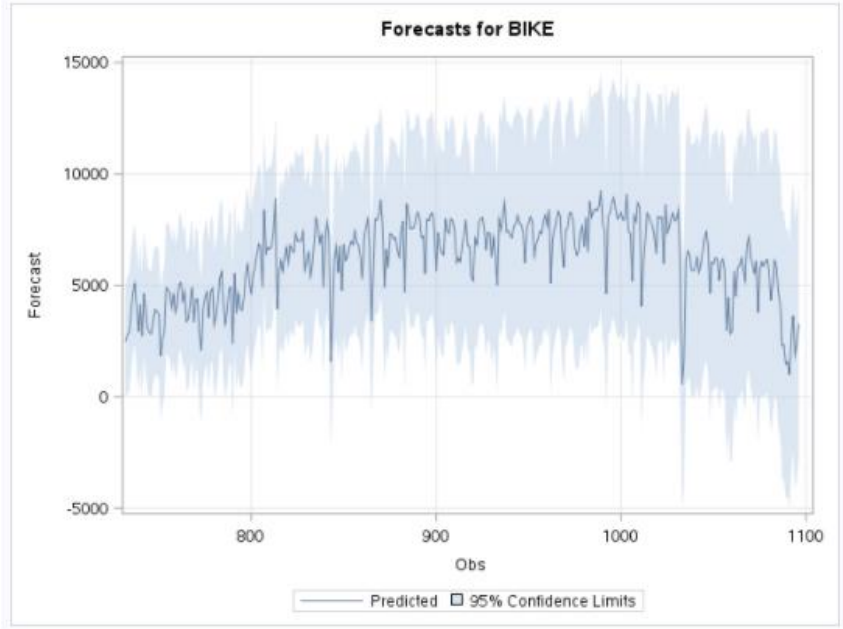


Figure 29 - Bike demand forecast for year 2013

REFERENCES

Fanaee-T, Hadi, and Gama, Joao, 'Event labeling combining ensemble detectors and background knowledge', Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg, [Web Link].

Data Original Source: http://capitalbikeshare.com/system-data
Weather Information: http://www.freemeteo.com
Holiday Schedule: http://dchr.dc.gov/page/holiday-schedule