STAT 929 Time Series Analysis Project

Deepak Rishi

August 10, 2016

1 Introduction

1.1 What is the problem?

The problem is to forecast the hourly bike sales from the **Bike Sharing Demand** challenge. The dataset descrition as obtained from Kaggle

Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis. Currently, there are over 500 bike-sharing programs around the world.
The data generated by these systems makes them attractive for researchers because the duration of travel, departure location, arrival location, and time elapsed is explicitly recorded. Bike sharing systems therefore function as a sensor network, which can be used for studying mobility in a city. In this competition, participants are asked to combine historical usage patterns with weather data in order to forecast bike rental demand in the Capital Bikeshare program in Washington, D.C.

1.2 Dataset Description

The dataset given is hourly rental data spanning two years. For this competition, the training set is comprised of the first 19 days of each month, while the test set is the 20th to the end of the month. The task is to predict the total count of bikes rented during each hour covered by the test set, using only information available prior to the rental period.

1.2.1 Data Fields

datetime - hourly date + timestamp
season - 1 = spring, 2 = summer, 3 = fall, 4 = winter
holiday - whether the day is considered a holiday
workingday - whether the day is neither a weekend nor holiday
weather - 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 - temperature in Celsius
atemp - "feels like" temperature in Celsius
humidity - relative humidity
windspeed - wind speed
casual - number of non-registered user rentals initiated
registered - number of registered user rentals initiated
count - number of total rentals

2 Techniques to tackle the problem

Section 1 introduced the problem statement . In this section I explore different methods that can help solve the problem.

2.1 Gradient Descent on Kaggle Loss Function

Gradient Descent is an optimization algorithm that iteratively updates the parameters of a model in steps propotional to the negative of the gradient until it converges to a global/local minimum. The advantage of using this approach is that, gradient descent can be applied to a non-convex objective function to find the local minimum.

The following figure shows the application of gradient descent.

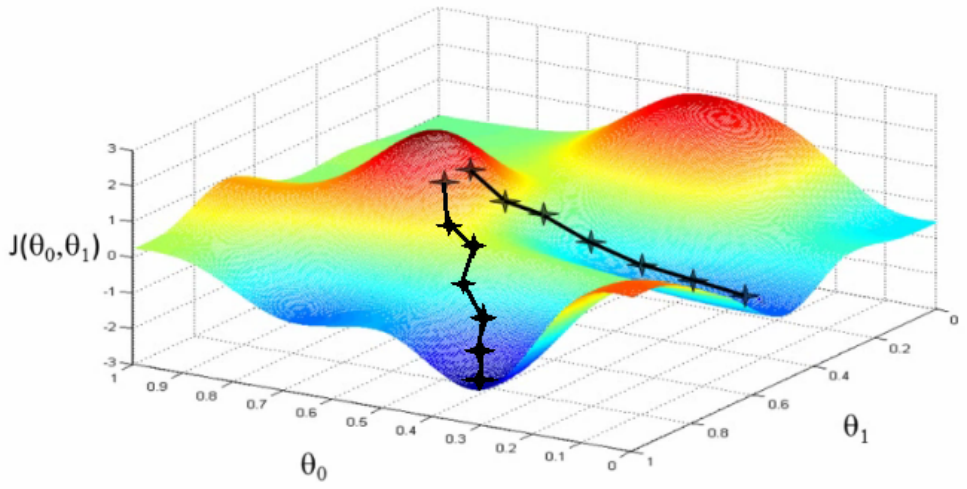


Figure 1: Gradient Descent (Datum-ML-Toolbox)

The Kaggle loss function is Root Mean Squared Logarithmic Error (RMSLE).

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2}$$

Where:

- n is the number of hours in the test set
- p_i is your predicted count
- a_i is the actual count
- $\log(x)$ is the natural logarithm

Figure 2: Kaggle Loss Function

For this project I use the square of the koss function and the gradient update of a parameter is given by

$$\frac{dE}{d\theta_j} = \frac{2}{n}\sum_{i=1}^{n}(log(\theta^T X_i + 1) - log(a_i + 1))\frac{X_{i,j}}{\theta^T X_i + 1}$$
$$\theta_j = \theta_j - \alpha * \frac{dE}{d\theta_j} \tag{1}$$

2.2 Support Vector Regression

Since this is a regression problem, support vector machine regression (Drucker et al., 1996) can be used to predict the bike sales. The optimization problem for Support Vector Machines can be formulated as (Smola and Schölkopf, 2004).

$$min(\frac{1}{||w||^2} + C\sum_{i=i}^{n}(\varepsilon_i + \varepsilon_i^*))$$
$$\text{Subject to}$$
$$y_i - w^T x \leq \epsilon_i + \varepsilon_i \tag{2}$$
$$w^T x - y_i \leq \epsilon_i + \varepsilon_i^*$$

where ϵ_i is the range within which the SVR predictions should lie and ε_i^* & ε_i are the soft margin errors

The purpose of using SVM's for regression is to leverage the power of kernels to transform the data into higher dimensions, which also augments the features of the dataset.

2.3 Gradient Boosted Trees

Gradient Boosted Trees (Friedman, 2000) are an ensemble learning approach to classification and Regression, which use weaks learners of Classification and Regression Trees (CART) (Breiman et al., 1984) to estimate a function.

Gradient Boosted Trees commonly optimize the following loss functions

- Least Squares Loss
$$L(\theta) = \sum_{i=i}^{n}(y - y_{pred})^2 \tag{3}$$
- Losgistic Loss :
$$L(\theta) = \sum_{i=i}^{n}(y_i ln(1 + e^{-y_{i,pred}}) + (1 - y_i ln(1 + e^{y_{i,pred}})) \tag{4}$$
- Exponential loss:
$$L(\theta) = \sum_{i=i}^{n}(e^{y_i * f(x_i)}) \tag{5}$$

  This is also used in Adaboost (Schapire). So, gradient boosted trees with exponential loss is essentially doing Adaboost.

2.4 Random Forests

Random Forests (Breiman, 2001) are an ensemble learning technique which constuct multiple regression trees and then average the results of all to give a prediction. The algorithm first randomly samples (with replacement) a subsets of the same size as the original data to be used for constructing the forest. Each tree in a Random forest

algorithm is constructed by randomply sampling a a subset of features. The most commonly used criteria for selection of a feature is the one which gives the maximum information gain.

The following figure shows random forests in action for classification.

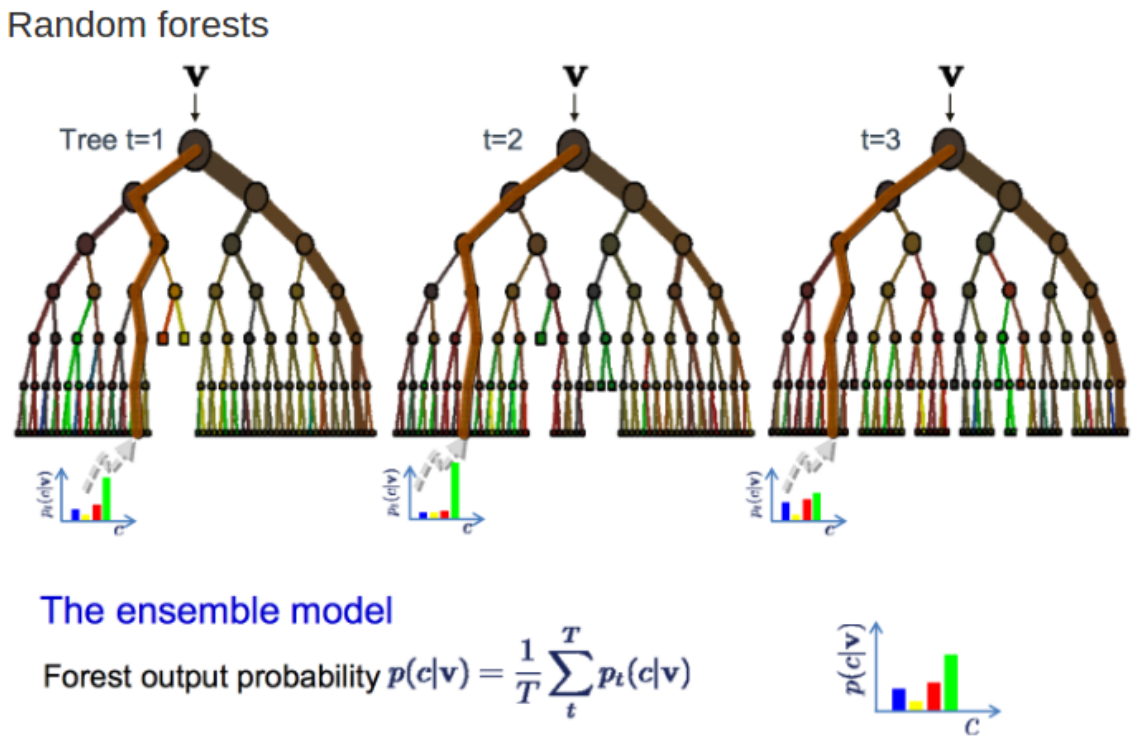


Figure 3: Random Forests

2.5 ARMA/ARIMA/SARIMA models on Residuals

ARMA/ARIMA and SARIMA models were used to forcast the residuals of the sales for each month separately. These residuals would be added back to the best predictor of the forcasted sales.

3 Results

3.1 Data Visulization

As a first step the dataset was visualized in 2D by **t-SNE** (van der Maaten and Hinton, 2008).

The following plot shows the original monthly data reduced to 2D by t-SNE

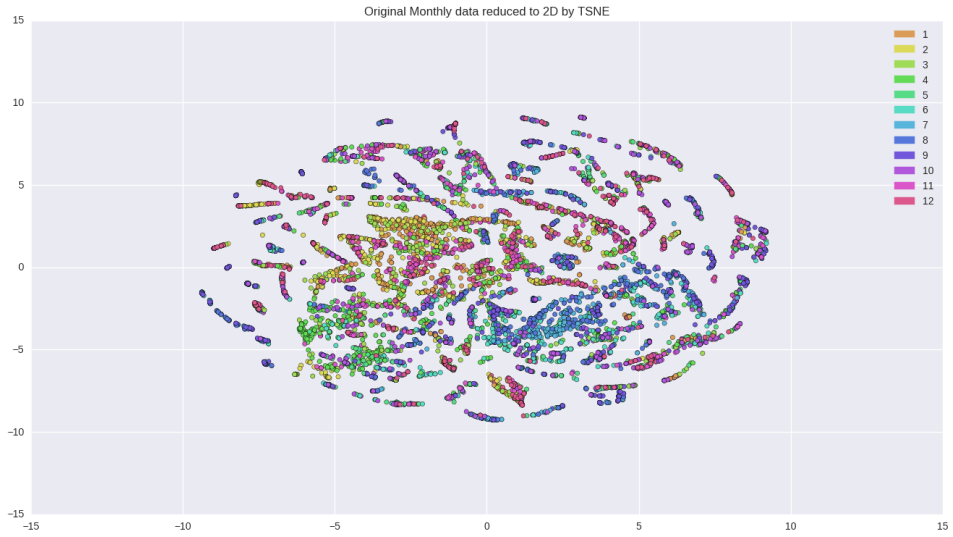


Figure 4: t-SNE 2D

The following plot shows the original monthly data reduced to 3D (the third dimension being the bike sales count) by t-SNE

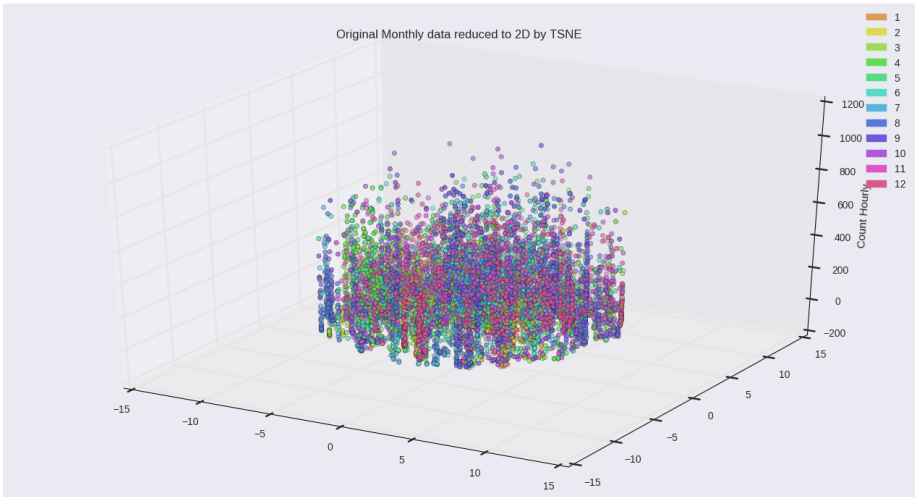
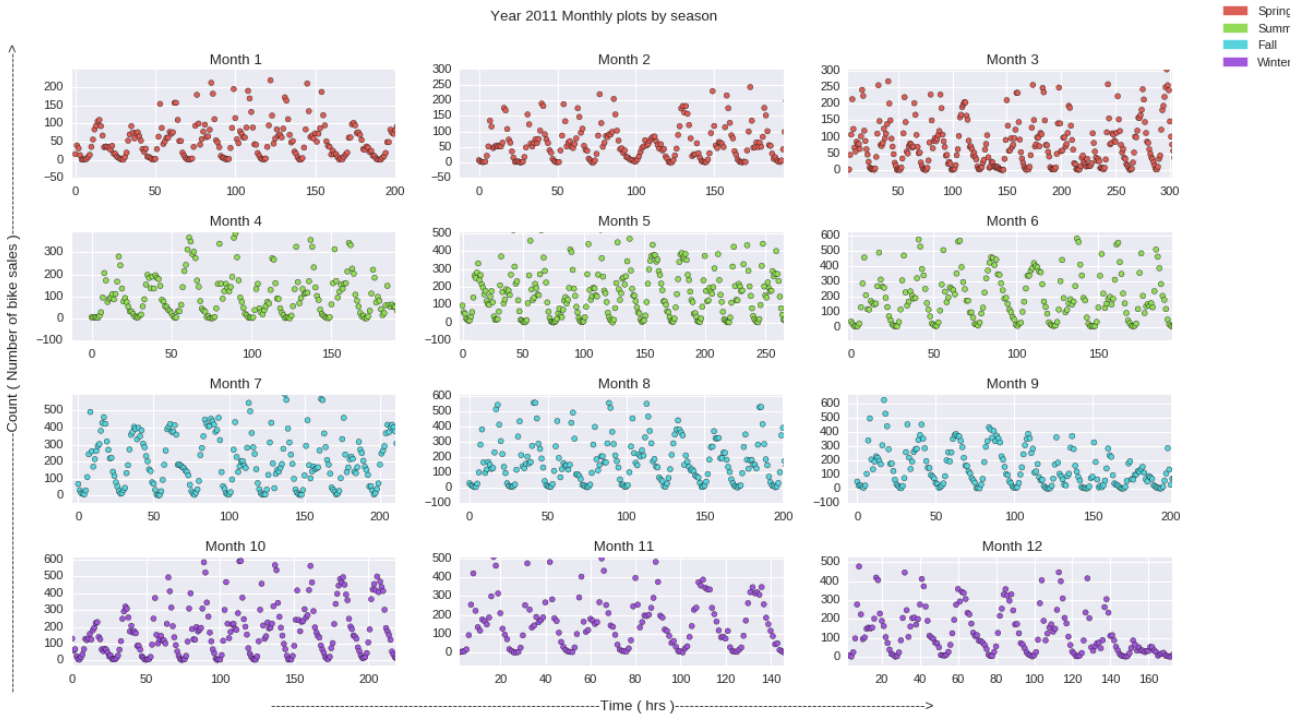


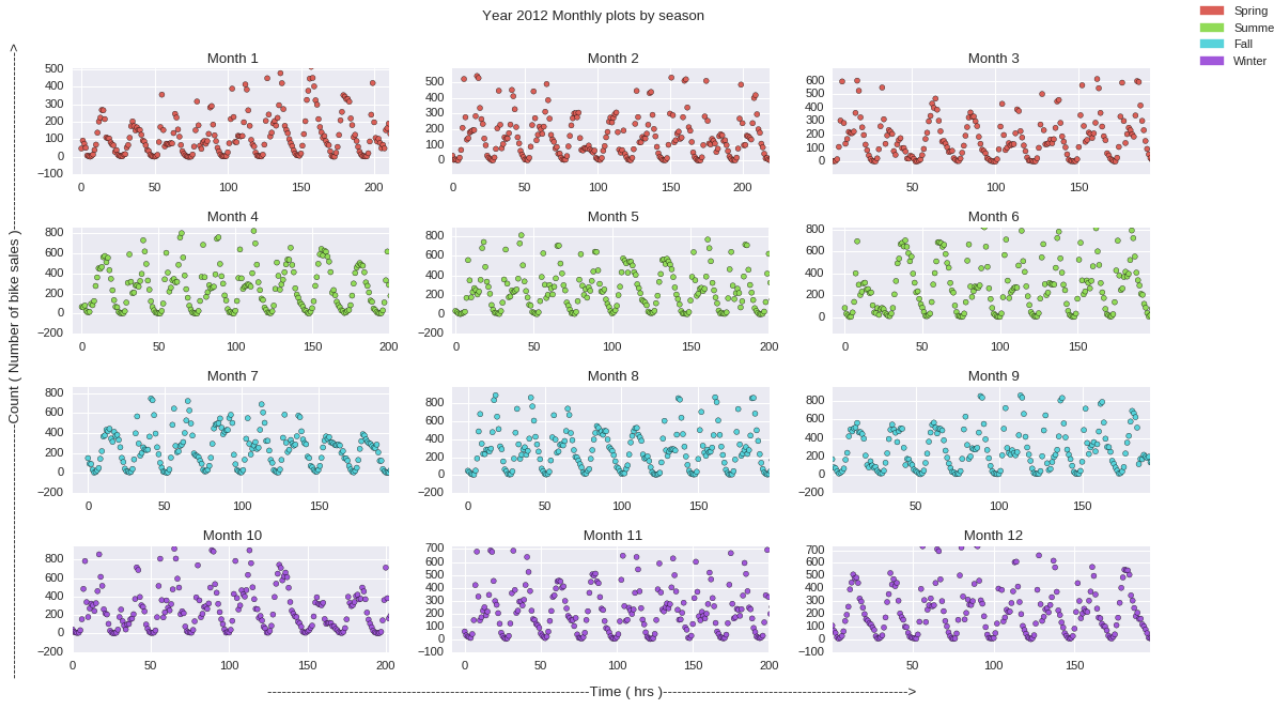
Figure 5: t-SNE 3D

As can be seen in the plots, with no feature engineering, its hard to find any relavant pattern in the dataset.

The following plot shows the montly sales as a time series data.



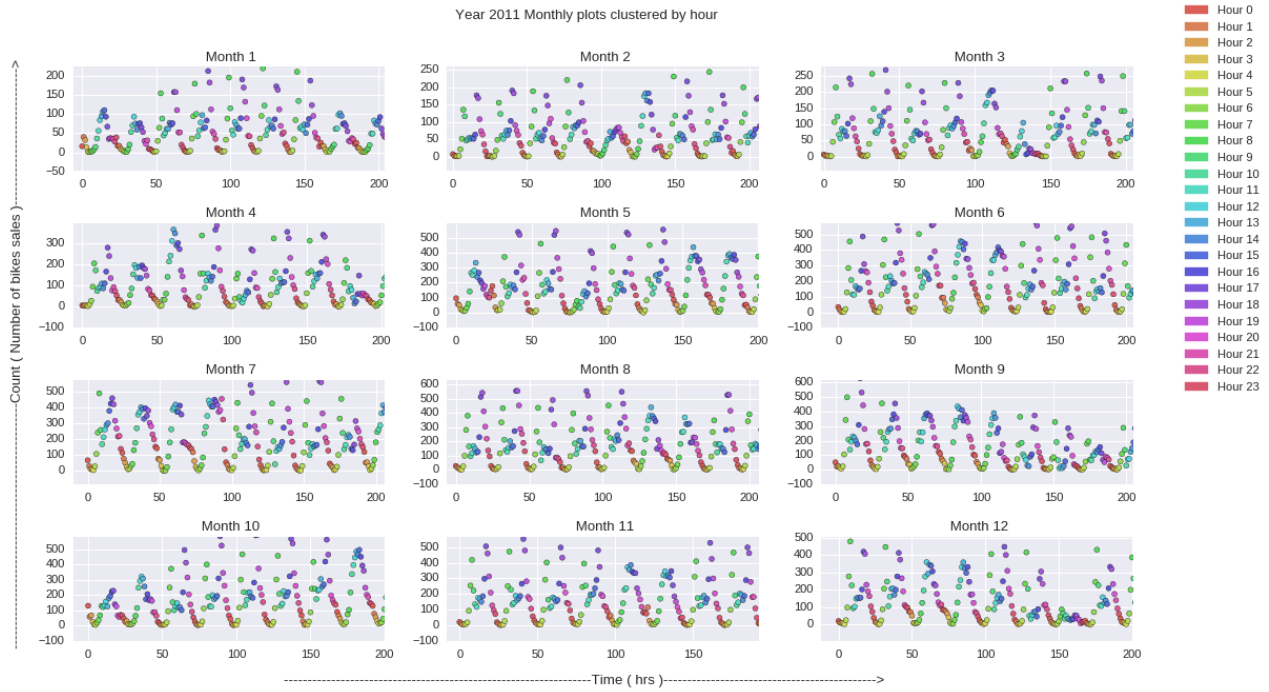
(a) 2011 Monthly dataset as time series by season



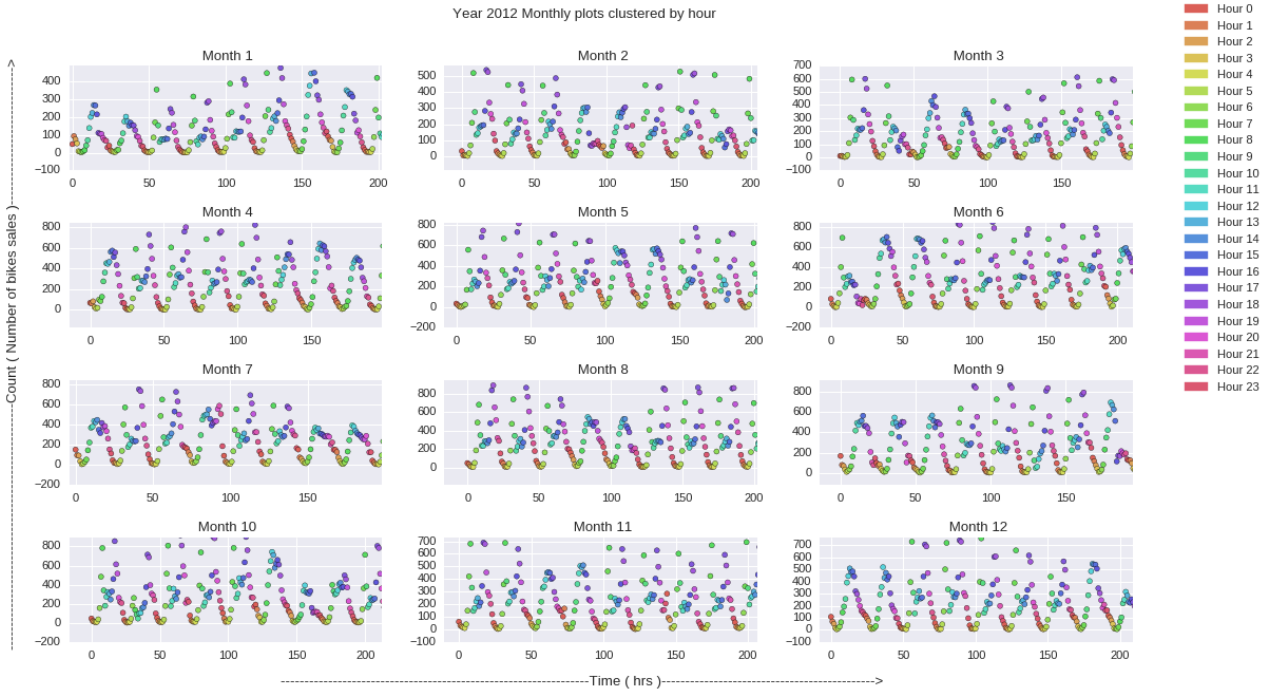
(b) 2012 Monthly dataset as time series by season

Figure 6: Time series representation of the dataset by season

The following plot shows the montly sales grouped by hour as a time series data.



(a) 2011 Monthly dataset as time series grouped by hour



(b) 2012 Monthly dataset as time series grouped by hour

Figure 7: Time series representation grouped by hour

3.2 Gradient Descent with L2 Regularizer on Kaggle Loss Function

As described in 2.1 I used gradient descent on the RMSLE 2 loss function. Before applying the gradient descent algorithm the dataset was preprocessed by augmenting features **Day of the week, Hour at which the sale**

was made, log transform of windspeed and humidity. Also the dataset was preprocessed by converting the categorial features mentioned in 1.2.1 into a One Hot Encoding Vector (Wikipedia)

The following plots shows the results of cross validation with a **L2 Regularizer(on the weights)** on the month December for the years 2011 and 2012 with different learning rates.

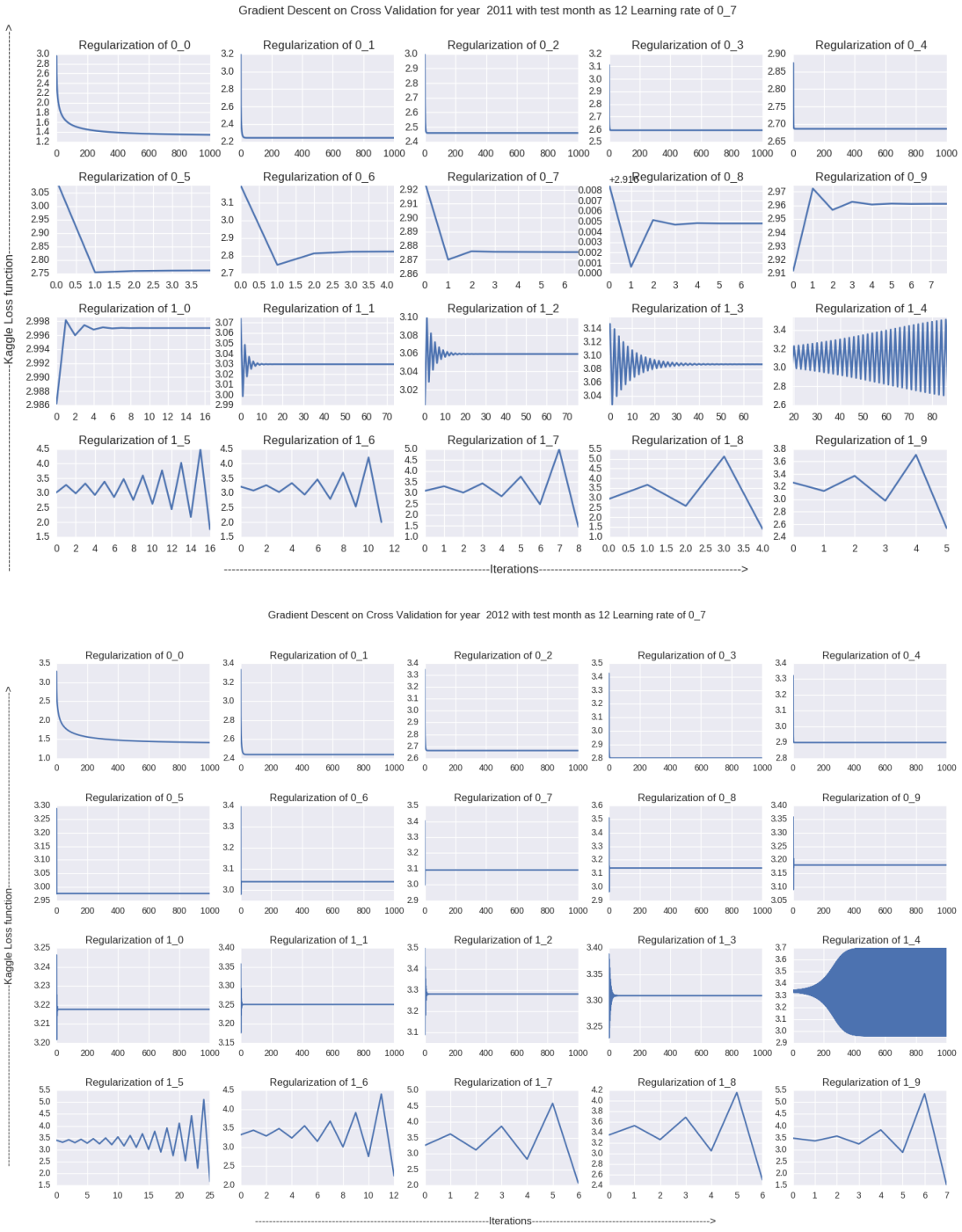


Figure 8: 5000 iterations of gradient descent (learning rate 0.7) cross validation using December as Test Set

Since the loss function started to cycle itself , the learning rate was desreased and the experiment repeated.

Figure 9: 5000 iterations of gradient descent (learning rate 0.4) cross validation using December as Test Set

Figure 8 and Figure 9 show that the minimum loss achieved by the gradient descent with L2 regularization is around **1.01**.

3.3 Support Vector Regression

Here also the dataset was preprocessed by augmenting features **Day of the week and Hour** and by converting the categorial features mentioned in 1.2.1 into a One Hot Encoding Vector (Wikipedia).

For SVR, cross validation and a grid search was performed over the parameters (C and ϵ) mentiontioed in equation 2.

The following plot shows the result of the cross validation and gris search over the parameters.

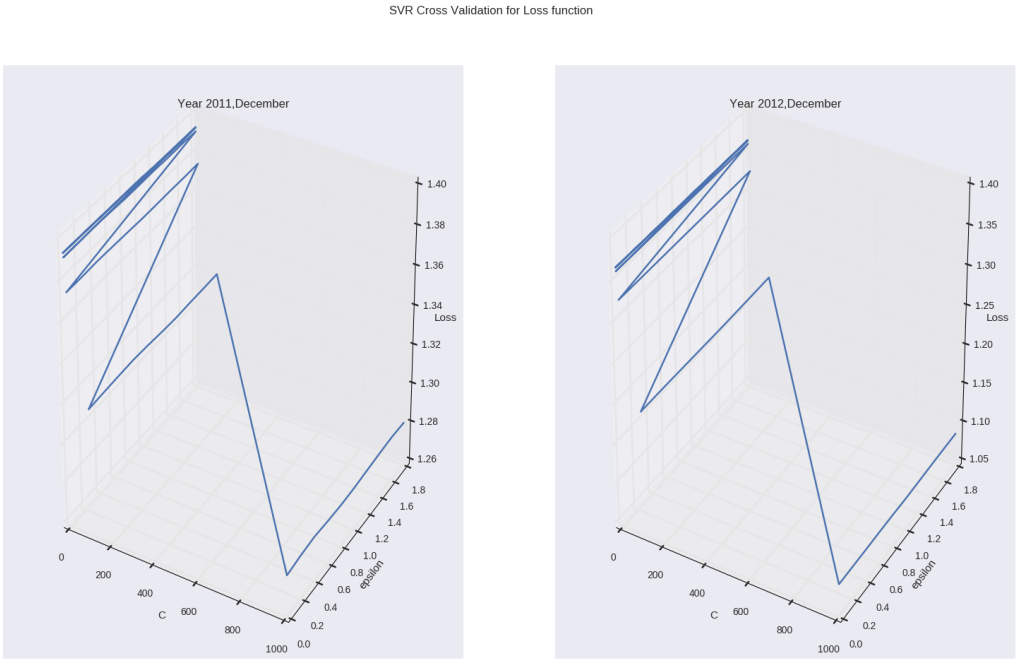


Figure 10: Cross Validation and Grid Search over C and ϵ

The lowest loss achieved by gradient descent is **1.25** with a $C = 1000$ and $\epsilon = 1.8$. Allowing for a large C is essentially overfitting. Hence this model is not acceptable even though the loss trend in Figure 10 is decreasing.

3.4 Gradient Boosted Trees

For this the dataset was preprocessed by only augmenting features **Day of the week and Hour**. The cross validation eror by gradient boosting was **0.525**. The feature importance by Gradient Boosted Trees is shown in Figure 11.

Hour & **Day of the week** are the most important features for regression.

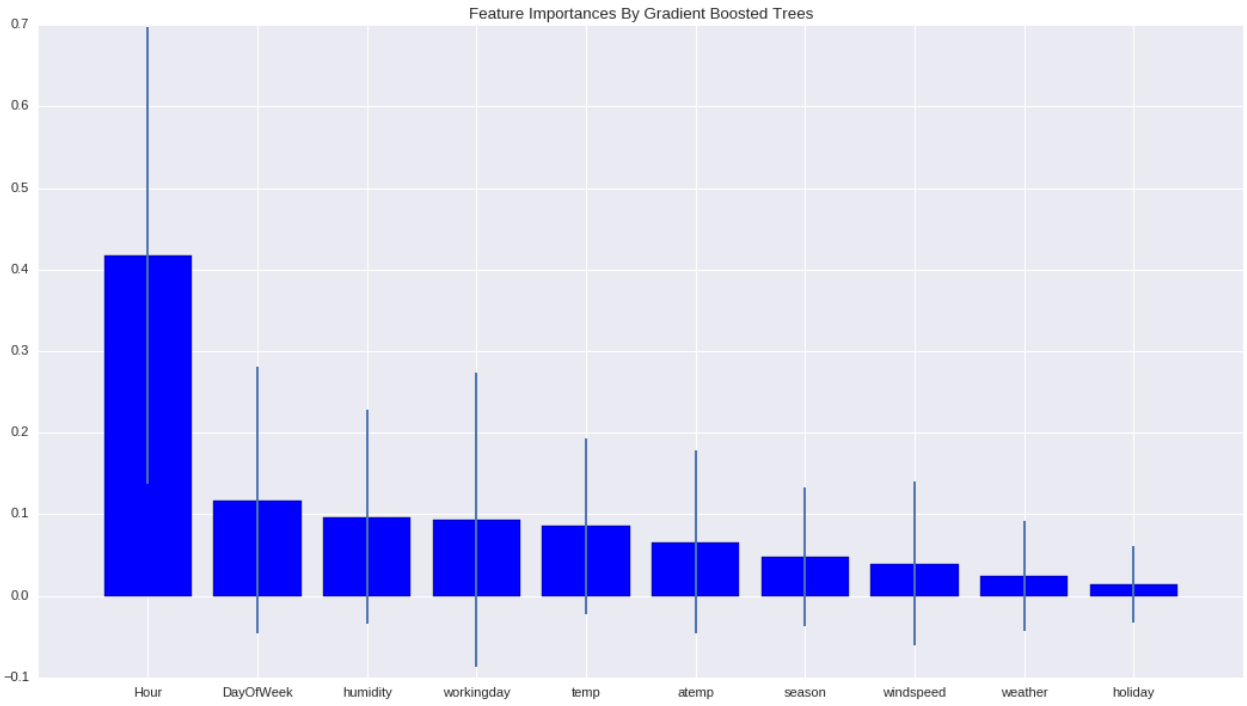


Figure 11: Feature Importance by Gradient Boosting Trees

With an ensemble of 300 trees I got a rank of **2591** on Kaggle Leaderboard.

2587	↓129	Bryan Conklin	0.74344	5	Sun, 11 Jan 2015 23:18:57 (-0.3h)
2588	↓129	Ashutosh Zode	0.74359	6	Tue, 14 Apr 2015 19:13:08 (-68.5d)
2589	↓129	MSDS Go Hoos	0.74487	4	Tue, 25 Nov 2014 20:18:13
2590	↓129	Abann Sunny	0.74577	7	Thu, 11 Dec 2014 02:51:12 (-19.8h)
-		**Deepak Rishi**	**0.74637**	-	**Mon, 08 Aug 2016 20:16:44** Post-Deadline

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

2591	↓129	Darshan Washimkar	0.74647	7	Tue, 16 Dec 2014 00:48:38 (-0.5h)
2592	↓129	UPV_SIE_JordiMagranerGimeno	0.74764	13	Thu, 08 Jan 2015 16:10:04
2593	↓129	ctkrohn	0.74769	1	Mon, 01 Sep 2014 15:05:58

Figure 12: Kaggle Leaderboard Rank by 300 Gradient Boosting Trees

3.5 Ensemble model of Random Forests and Gradient Boosted Trees

For this the dataset was preprocessed by only augmenting features **Day of the week and Hour**.With a large number of trees in the forest, log transforms of **windspeed & humidity** did not offer much help in lowering the loss.

The cross validation eror by random forests of **150** trees and **300** gradient boosted trees was **0.4059**.

With an ensemble of 150 random forest trees and 300 gradient boosted trees I got a rank of **1667** on Kaggle Leaderboard.

1664	↑411	bscheetz	0.49781	23	Thu, 28 May 2015 23:44:36
1665	new	topass	0.49787	1	Fri, 29 May 2015 14:49:45
1666	↓109	Aleksander Ring	0.49796	2	Mon, 25 May 2015 09:11:19 (-99.5d)
-		**Deepak Rishi**	**0.49797**	-	**Mon, 08 Aug 2016 20:40:44** Post-Deadline

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

1667	↓109	HappyPrancer	0.49804	5	Sun, 08 Feb 2015 14:55:41 (-28.6d)
1668	↓109	Nick Kaufman	0.49816	1	Tue, 07 Apr 2015 20:59:39
1669	↓109	Ankit	0.49820	3	Mon, 18 Aug 2014 04:39:55

Figure 13: Kaggle Leaderboard Rank by ensemble of Random Forests of 150 trees and 300 gbt trees

3.6 Random Forests

For this the dataset was preprocessed by only augmenting features **Day of the week and Hour**. The cross validation eror by random forests of **150** trees was **0.4051**

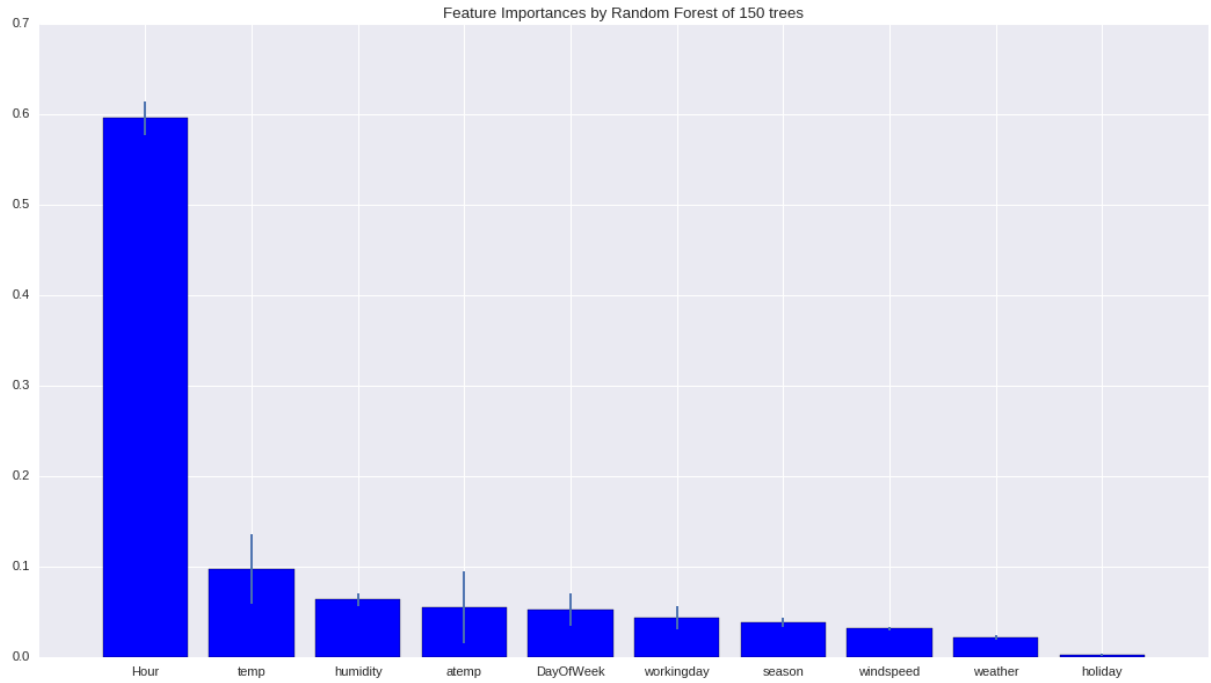


Figure 14: Feature Importance by Random Forests of 150 trees

With an ensemble of 150 trees I got a rank of **902** on Kaggle Leaderboard.

899	↓77	Yanping Yang	0.45612	1	Tue, 12 Aug 2014 22:47:19
900	↓77	Pranjul Yadav	0.45615	15	Mon, 23 Feb 2015 21:07:33 (-41.8h)
901	↓77	Challengers	0.45646	22	Sat, 11 Oct 2014 13:34:09 (-0h)
-		**Deepak Rishi**	**0.45648**	-	**Mon, 08 Aug 2016 20:51:15** Post-Deadline

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

902	↓77	TBarker	0.45649	4	Wed, 11 Feb 2015 22:16:34
903	↓77	Bruce Pan	0.45675	3	Tue, 30 Dec 2014 06:35:28

Figure 15: Kaggle Leaderboard Rank by Random Forests of 150 trees

The following figures show the predicted/forecasted sales by the Random Forest of 150 trees

Figure 16: 2011 Monthy Predicted and Train Sales

Figure 17: 2012 Monthy Predicted and Train Sales

The following figures show the actual sales/predicted sales on the Training Set by the Random Forest of 150 trees.

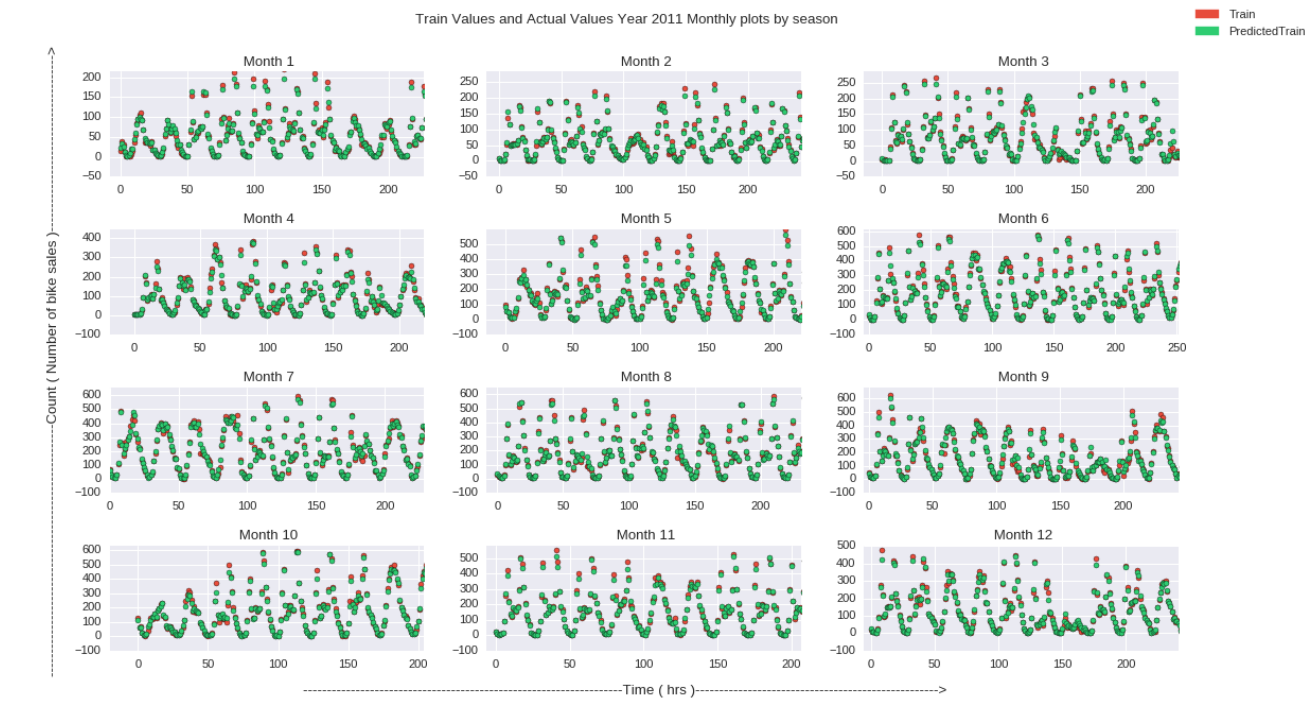


Figure 18: 2011 Monthy Predicted Train sales and Actual Train Sales

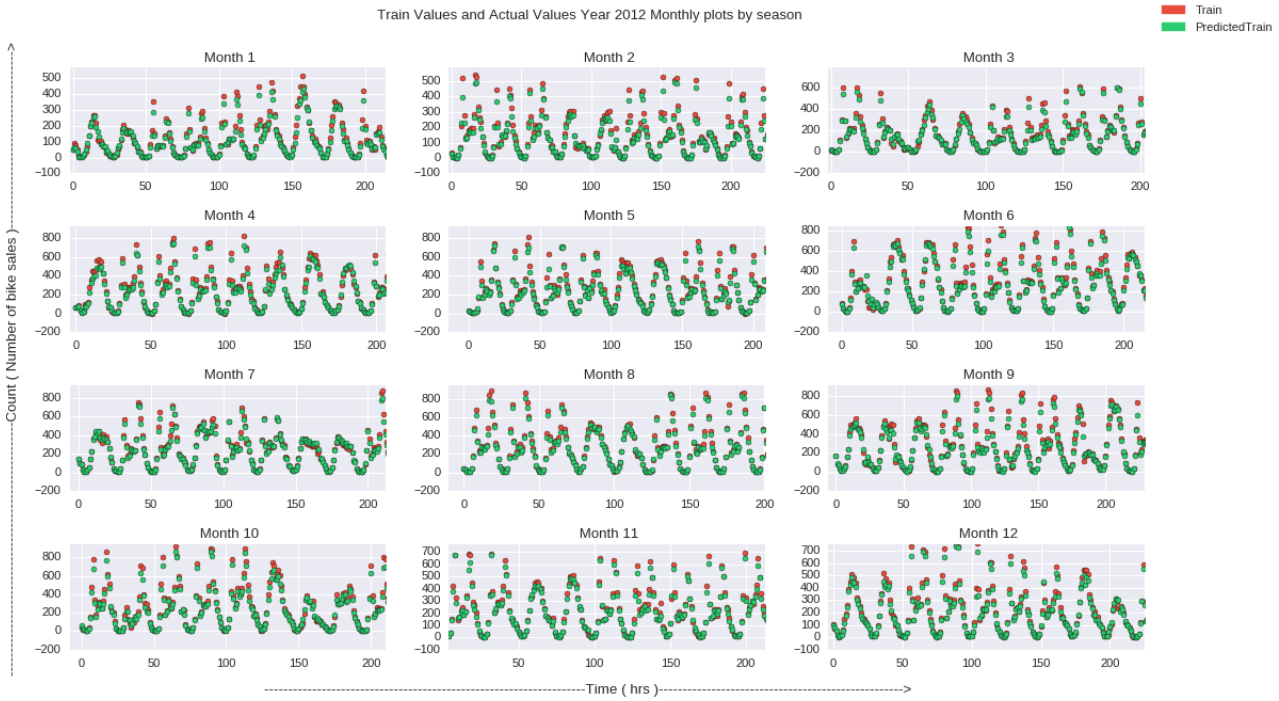


Figure 19: 2012 Monthy Predicted Train sales and Actual Train Sales

The following figures show the residuals by the Random Forest of 150 trees

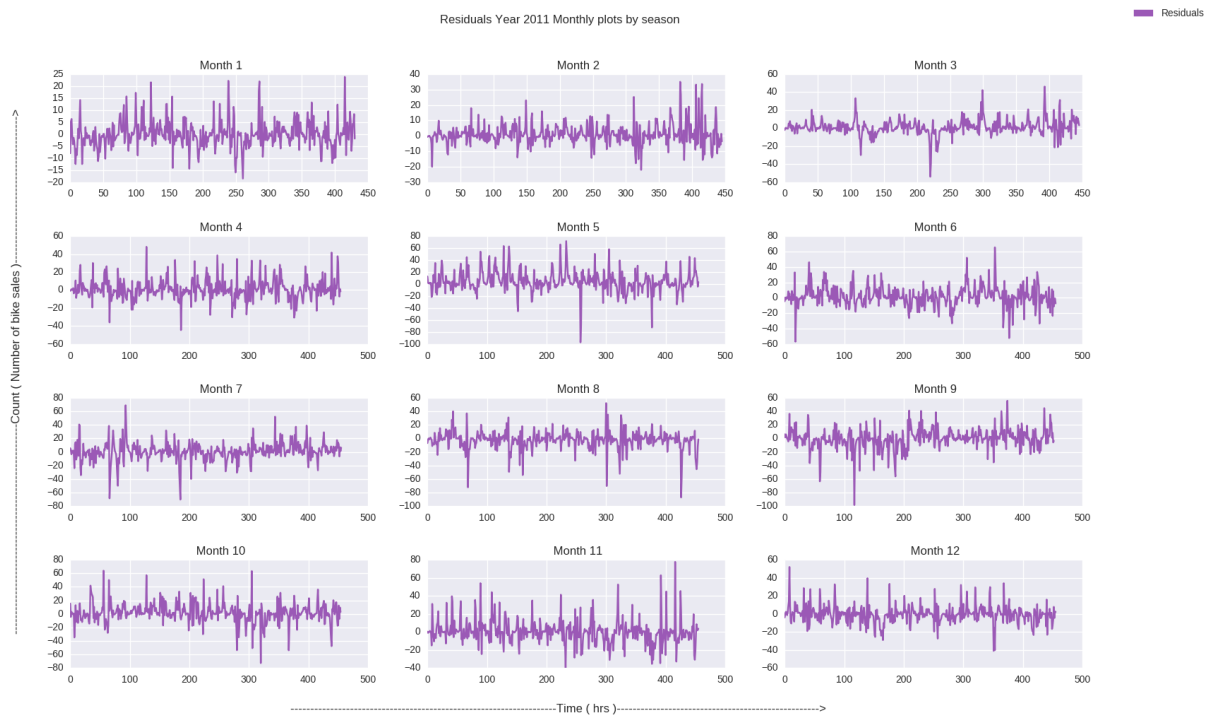


Figure 20: 2011 Monthy Residuals

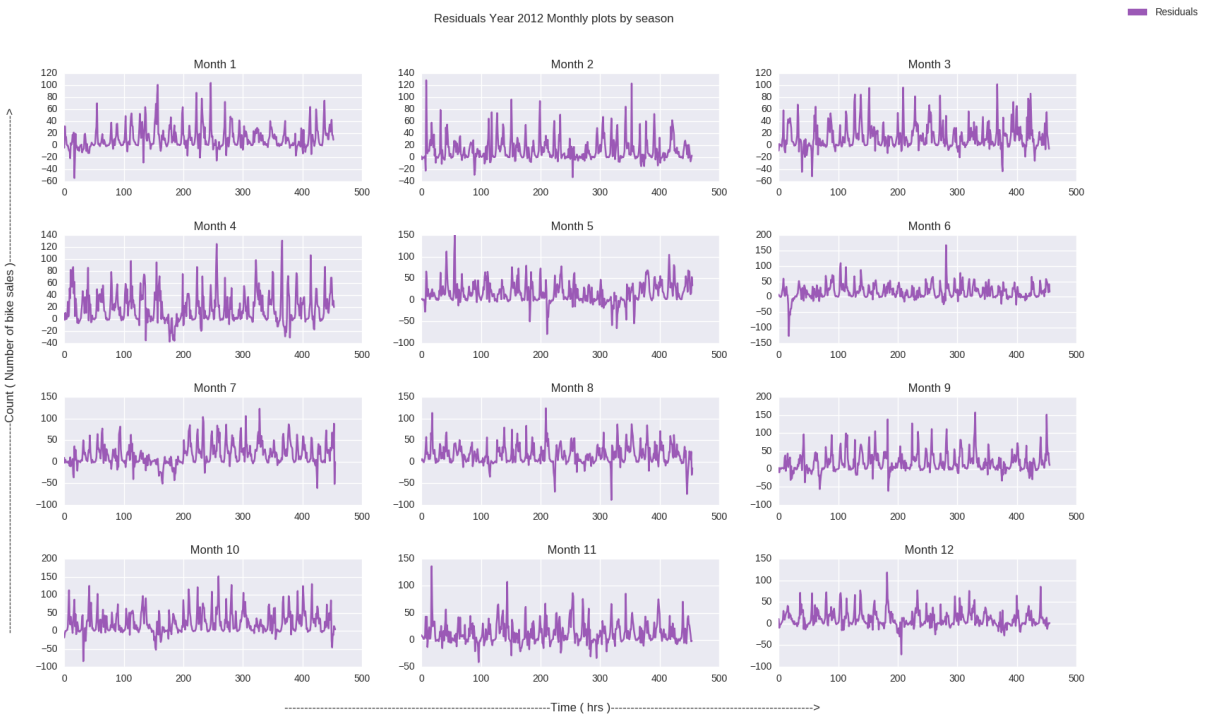


Figure 21: 2012 Monthy Residuals

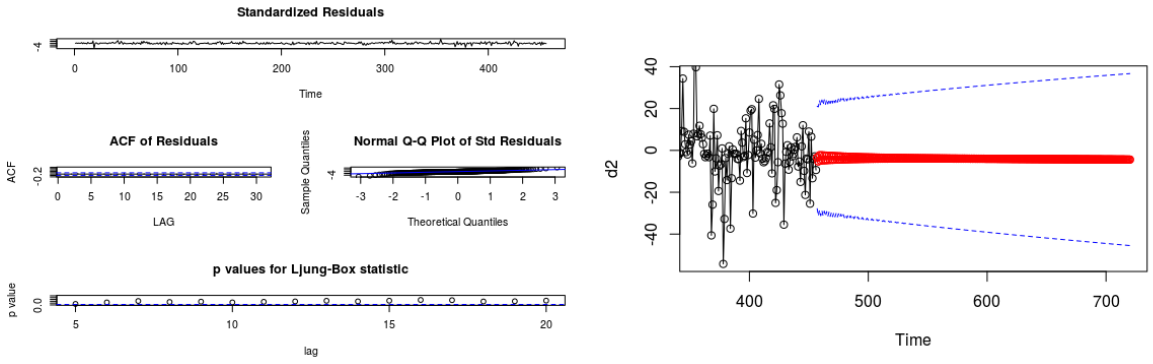
3.7 SARIMA and Random Forests

In this approach I forecast the residuals of of the sales of each month and then add them to the sales predicted by Random Forests.

The residuals for each month are seperately processed and an ARMA/ARIMA/SARIMA model is fitted to each month.

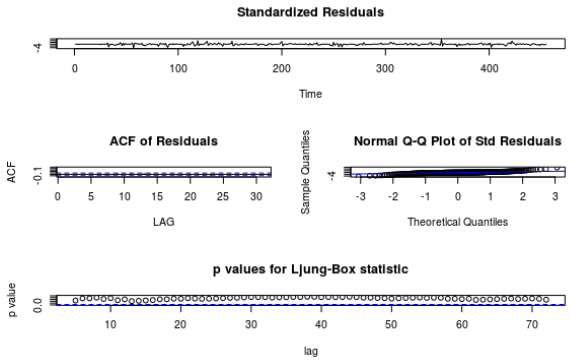
Before fitting the ARMA/ARIMA/SARIMA model the rsiduals for each month are centered by subtracting the mean and later at the time of forcast the mean of the residuals is added back.

The following plots show a few months of data fitted fit a ARMA/ARIMA/SARIMA model

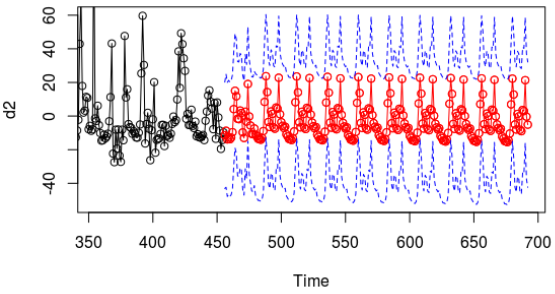


(a) ARIMA (2,1,2) diagnostic on Month 6 YEAR 2011 (b) ARIMA (2,1,2) forecast on Month 6 YEAR 2011

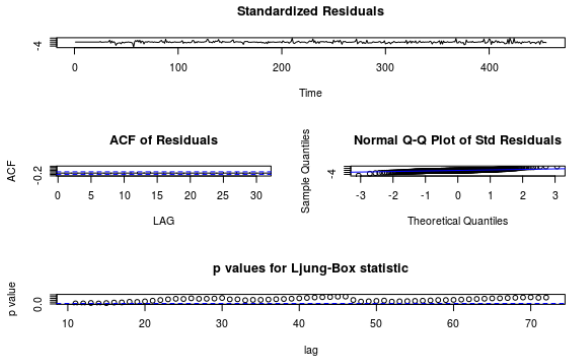
Figure 22: ARIMA models fitted on the residuals.



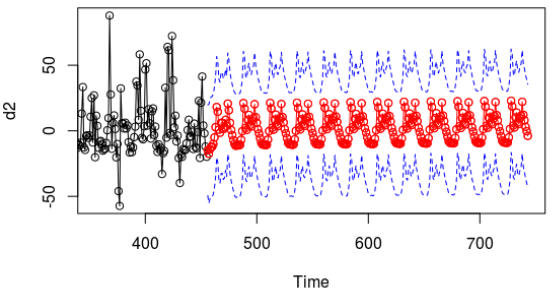
(a) SARIMA (1,0,1)X$(0,1,2)_{24}$ diagnostic on Month 2 YEAR 2012



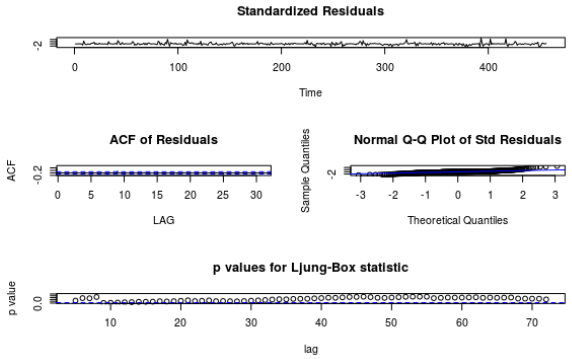
(b) SARIMA (1,0,1)X$(0,1,2)_{24}$ forecast on Month 2 YEAR 2012



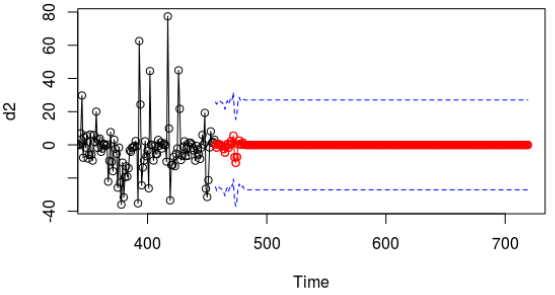
(c) SARIMA (7,0,1)X$(0,1,2)_{24}$ diagnostic on Month 3 YEAR 2012



(d) SARIMA (7,0,1)X$(0,1,2)_{24}$ forecast on Month 3 YEAR 2012



(e) SARIMA (1,0,2)X$(0,0,1)_{24}$ diagnostic on Month 11 YEAR 2011



(f) SARIMA (1,0,2)X$(0,0,1)_{24}$ forecast on Month 11 YEAR 2011

Figure 23: SARIMA models fitted on the residuals.

By adding the forcasted residuals I got a rank of **1869** on the Kaggle leaderboard.

1865	↓107	Sunil	0.51525	11	Mon, 08 Dec 2014 01:45:10 (-0.4h)
1866	↓107	Hamad	0.51548	1	Thu, 27 Nov 2014 05:06:10
1867	↓107	peeps	0.51585	4	Sun, 03 May 2015 18:59:42 (-0.1h)
1868	↓107	Nanun Tr	0.51586	8	Fri, 24 Apr 2015 00:27:59
-		**Deepak Rishi**	**0.51630**	-	**Tue, 09 Aug 2016 19:16:35** Post-Deadline
Post-Deadline Entry If you would have submitted this entry during the competition, you would have been around here on the leaderboard.					
1869	↓106	MikeT	0.51664	6	Mon, 03 Nov 2014 15:20:53
1870	↓106	prabhakaran Arivalagan	0.51679	4	Thu, 01 Jan 2015 03:03:49

Figure 24: Kaggle rank after adding the forecasted residuals

3.7.1 Analysis of SARIMA Models

- ARMA/ARIMA models were fitted to most of the months in the year 2011. The seasonality trend started showing up after from the year 2012.
- For each month an average of about 250 forecasts were made. Thus, for most models, the forecasts converged rapidly to the mean. I think due to that the forcasted residuals did not help much in improving the predicted model.

4 Conclusion

In this project , I explored different methods to forecast a multivariate time series. I started off with gradient descent 2.1 on the RMSLE with L2 regularization. After a grid search and cross validation the lowest loss that could be achieved was 1.01.

Support Vector Regresion 2.2 was tried. Although Gaussian kernel was used transforming the features into a high dimensional space did not help much in this case.

Ensemble methods such as Gradient Boosting Trees 2.3 and Random Forests 2.4 proved to be the best estimators in this case. They were instrumental for doing manual feature engineering. In fact it was observed that increasing the number of trees in the forest lesser feature enginnering was acheving better results.

In the end ARMA/ARIMA/SARIMA 2.5 models were used to forecast the residuals which were later added back. Although promising, due to the fact that we had to forcast a large number of points for each month, the forcast values converged to the mean quickly (especially for the year 2011). The resultant forcasts were not as accuarate as the ones produced by ensemble methods.

References

Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL `http://dx.doi.org/10.1023/A:1010933404324`.

Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.

Datum-ML-Toolbox. Gradient Descent. `http://blog.datumbox.com/tuning-the-learning-rate-in-gradient-descent/`. Accessed: 2016-08-01.

Harris Drucker, Chris J.C. Burges, Linda Kaufman, Chris J. C, Burges* Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines, 1996.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

Kaggle. Bike demand sharing. `https://www.kaggle.com/c/bike-sharing-demand`. Accessed: 2016-08-01.

Robert E. Schapire. Explaining adaboost.

Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August 2004. ISSN 0960-3174. doi: 10.1023/B:STCO.0000035301.49549.88. URL `http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88`.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *The Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

Wikipedia. One Hot encoding. `https://en.wikipedia.org/wiki/One-hot`. Accessed: 2016-08-01.