Stream Mining Algorithms for Sensor Data Classification

Yue Dong
yuedong029@uottawa.ca

Philippe Paradis
philippe.paradis@carleton.ca

April 16, 2015

Contents

Acknowledgements

Abstract

1 Introduction

What relationships do we expect to see in the data?

- Variation in bike rentals based on hour of the day
- Different bike rental patterns on the weekends versus regular weekdays
- Different bike rental patterns on holidays versus non-holidays
- Variation in bike rentals based on temperature
- Variation in bike rentals based on weather condition
- No idea about effects of humidity
- Variation in bike rentals based on wind speed

What kind of relationships do we expect to see in the data?

- On regular weekdays, spike in traffic during morning and afternoon rush hours
- Typically, higher bike rental volumes during the day and lower during the night
- In other words, highly non-linear relationship between bike rentals and datetime
- Roughly linear variation between bike rentals and temperature (although very high summer temperatures could lead to reduced bike rental volumes)
-

- What could possibly explains the missing values that we see? There is clearly a pattern in the missing values. Most of those happen at the same time at night. Also, the bike count values tend to change and be much smaller around missing values. Why is that?

2 Dataset & Data exploration

2.1 Dataset

The dataset we used is the Bike Sharing Dataset from the UCI repository. According to the dataset description, "this 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." The summary of this dataset is as Table 1 shows. More information of the dataset can be found at http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset.

Number of instances:	17389 (hourly), 731 (daily)
Number of Attributes:	16
Attribute Characteristics:	Categorical, Integer, Date/time, Decimal
Attributes Information:	
Attribute name	**Description**
instant	record index
dteday	date
season	season (1:springer, 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
weekday	day of the week
workingday	1: if day is neither weekend nor holiday; 0: otherwise.
weathersit	1: Clear; 2: Mist; 3: Light Snow, Light Rain; 4: Heavy Rain, Ice Pallets, Snow + Fog;
temp	Normalized temperature in Celsius
atemp	Normalized feeling temperature in Celsius
windspeed	Normalized wind speed
casual	count of casual users
registered	count of registered users
cnt	count of total rental bikes including both casual and registered

Table 1: summary of the bike sharing dataset

2.2 Data Visualization

Data visualization is a way to gain some insight about the dataset we are investigating. We first plotted an scatterplot matrix on bike sharing daily dataset to explore the correlations between the attributes.

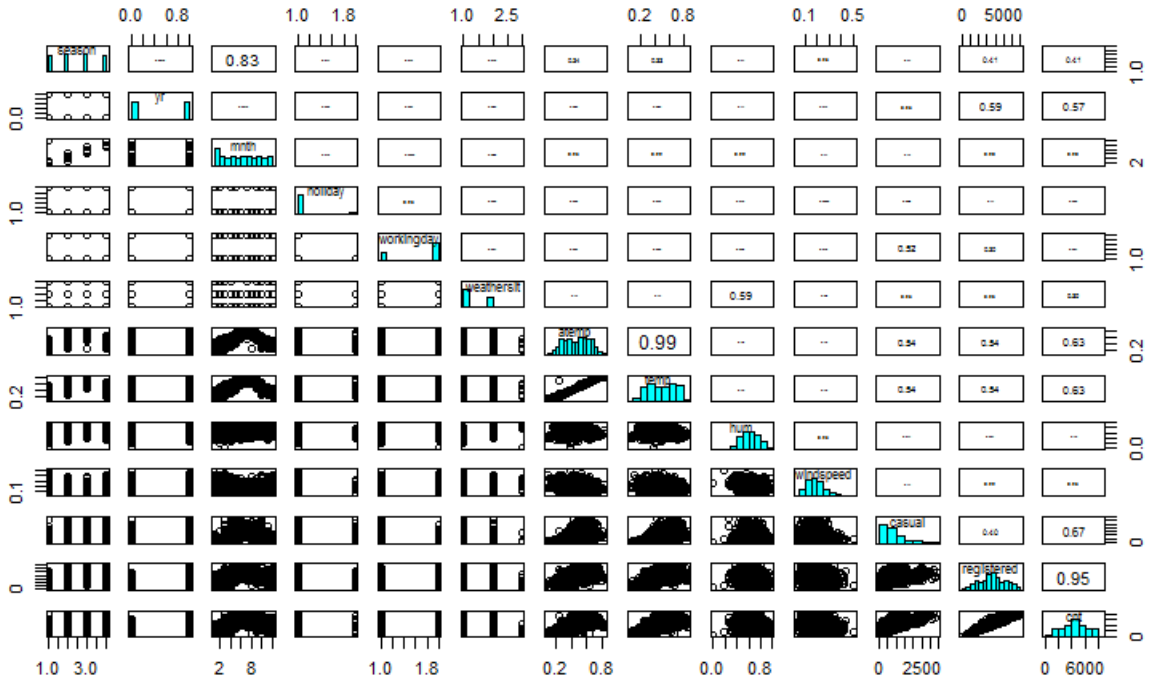


Figure 1: scatterplot matrix of the bike sharing (daily) dataset

As we can see from Figure 1, there are 0.99 correlation between temp and atemp, 0.95 correlation between registered and cnt, and 0.83 correlation between season and mnth.

High correlation between two variables means that as one variable rises or falls, the other variable rises or falls as well. Since we don't want attributes with high correlation in our dataset, we can pick one attribute from each pair with high correlation.

On the other side, attributes with high correlations to cnt can be good predictors for the bike rental counts. From the last column of Figure 1, we obtained a list of attributes with high correlations to cnt in decreasing order:

	registered	casual	temp/atemp	yr	season
correlation to cnt	0.95	0.67	0.63	0.57	0.41

Table 2: correlations to cnt in decreasing order

To further explore the correlations between variables, we plotted a heat map with hierarchical clustering.

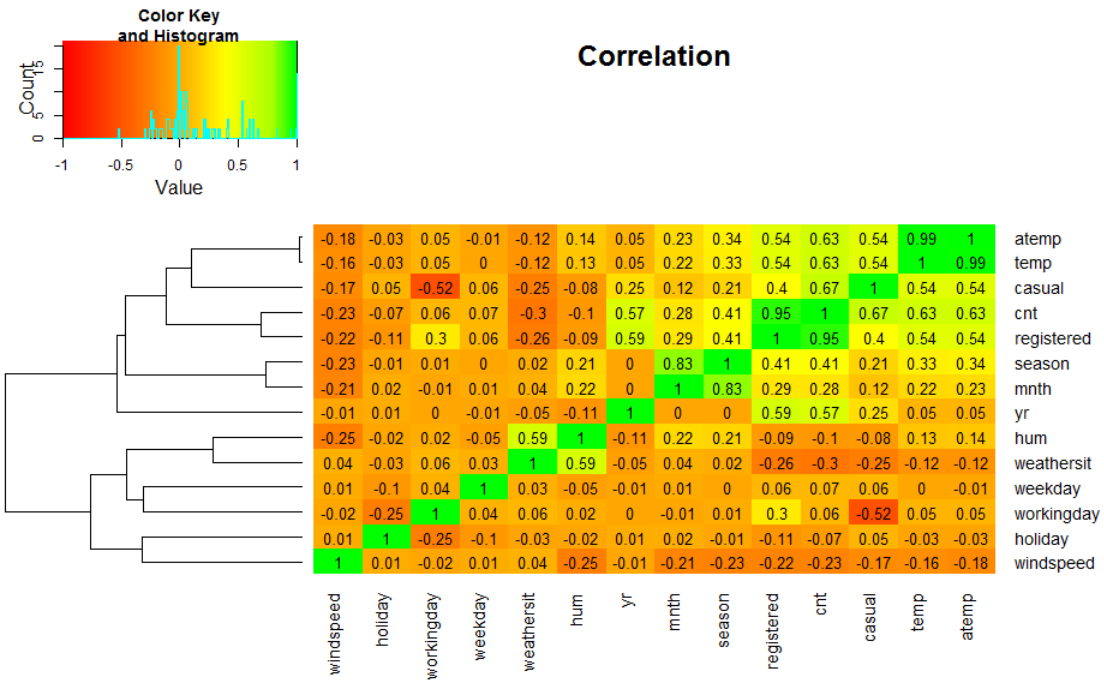


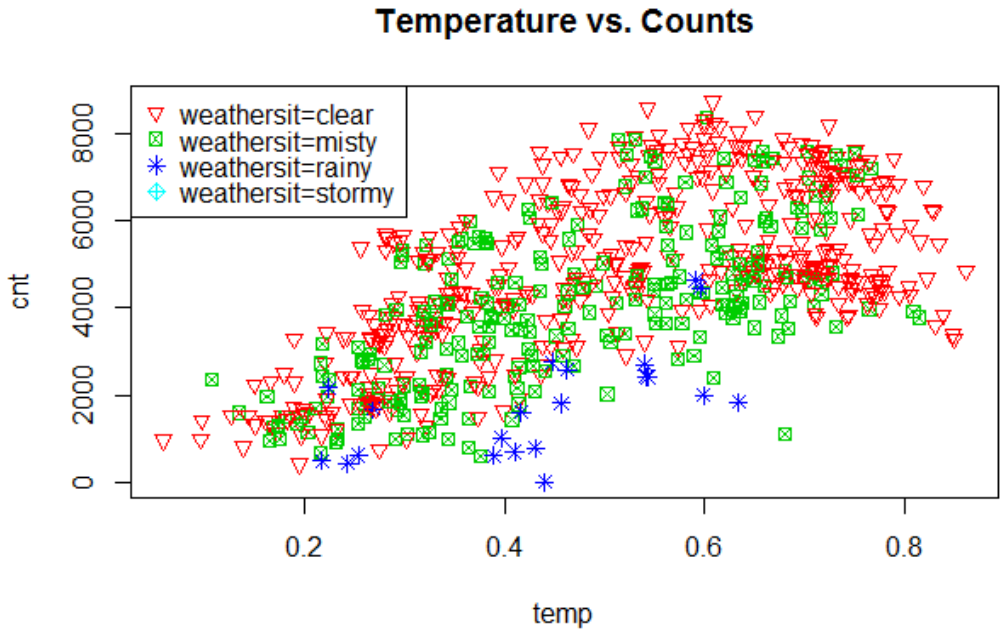
Figure 2: Heat map of the bike sharing (daily) dataset

From Figure 2, we found that workingday has a negative correlation with casual users (-0.52) which corresponds to the intuition that more casual users go out biking when it is not workingday.

Moreover, the hierarchical clustering grouped atemp and temp, cnt and registered, season and month, hum and weathersit, weekday and workingday together in the lowest level. If we want two clusters from the hierarchical clustering, we will get atemp, temp, casual, cnt, registered, season, mnth, yr as one cluster; and hum, weathersit, weekday, workingday, holiday, windspeed as the other cluster.

These correlations give us an insight about the dataset. Based on the above observations, we decided to further investigate on the following relationships: 1) temp/weathersit to cnt, 2) workingday to cnt, 3) hr/workingday to cnt, 4)season/mnth to cnt, 5) registered to casual.

Explore 1. Temp/Weathersit to Cnt Most likely, temperature and weather influence the bike demand. We thus plotted temp against daily cnt with weather information:



The rental bike counts increases when the temperature increases. With the same temperature, the better the weather is, the more bike rental counts are.

To compare the effects of feeling temperature and temperature on bike rental demand, we plotted atemp against the average cnt and temp against the average cnt:

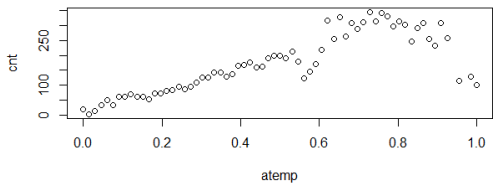


Figure 3: atemp against average cnt

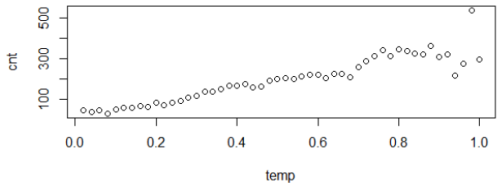
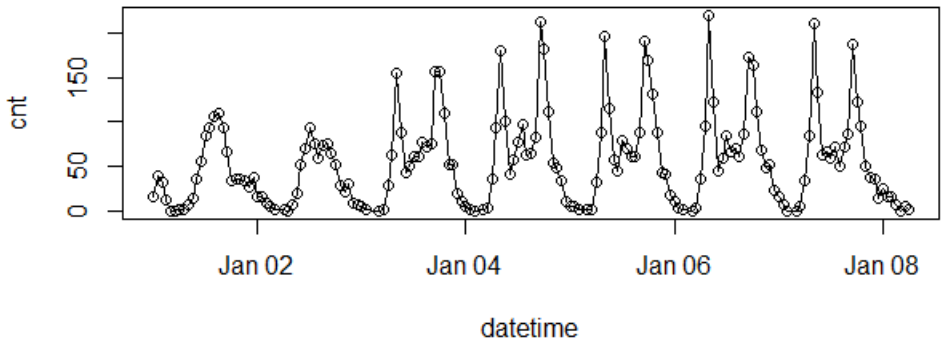


Figure 4: temp against average cnt

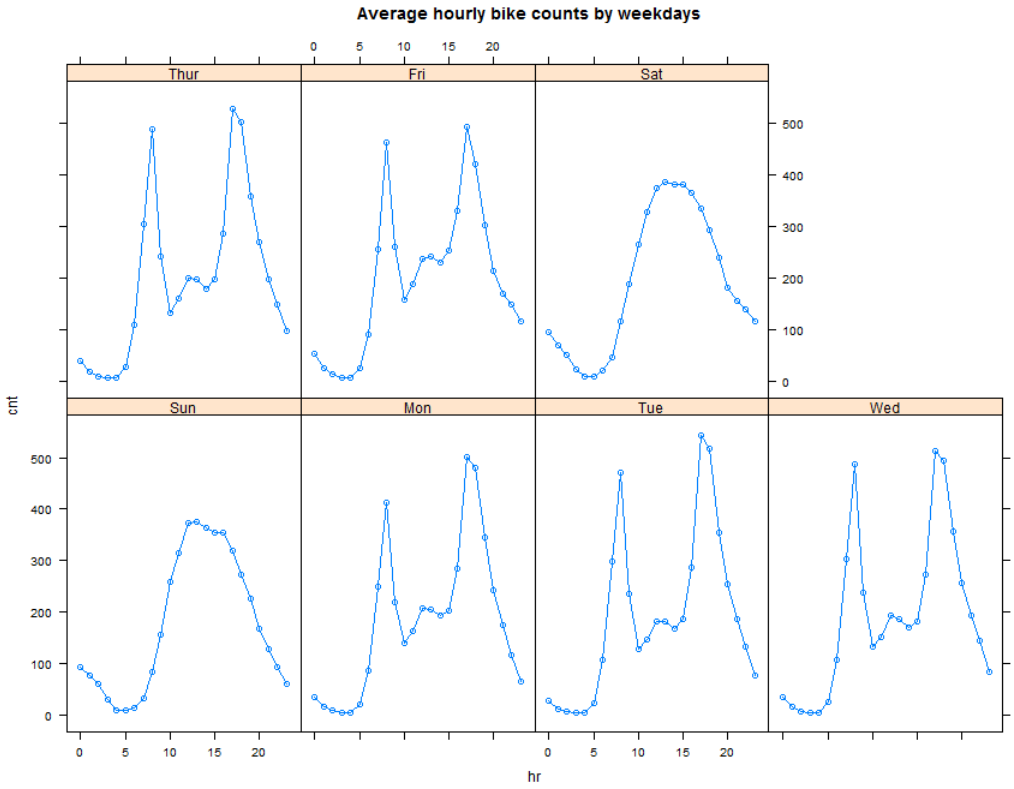
We can confirm the high linear correlation between temp and cnt from Figure 4 since the plot is most linear. On the other hand, we can see from Figure 3 that the bike rental demand is the highest at atemp= 0.7, it start to decrease when atemp exceeds 0.7.

Explore 2. Weekdays to Cnt Most likely, the bike rental will show a different pattern on weekdays and weekends. part of the people who rent bike on weekdays probably use it for commute to work, people who rent bikes on weekends most likely are for leisure.

First, we plotted the counts on a typical week:



We see that the bike demand is different on weekdays and weekends in this week. To check if it is true generally, we plotted the average hourly counts conditional on day of the week from the whole dataset.



The following is the plot of hourly counts conditional on day of the week AND if it's a holiday.

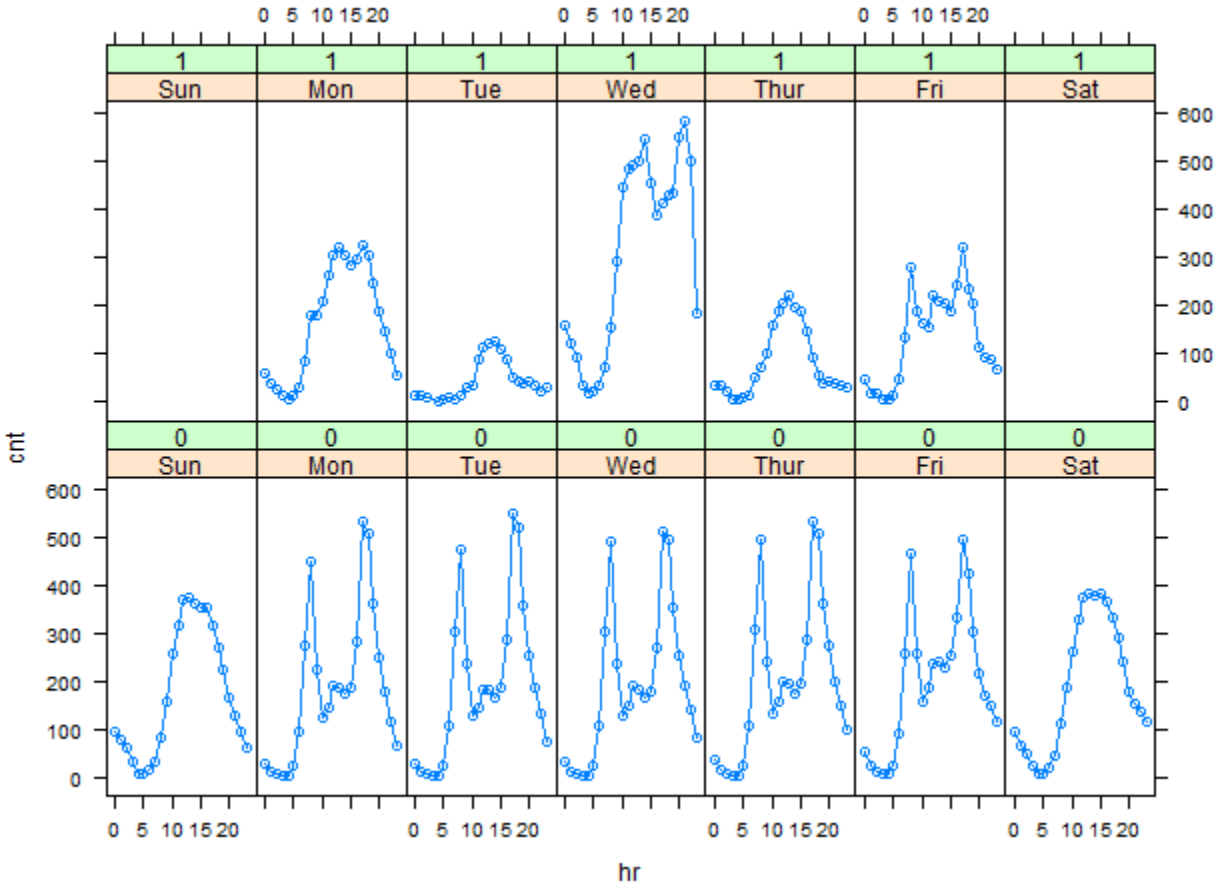
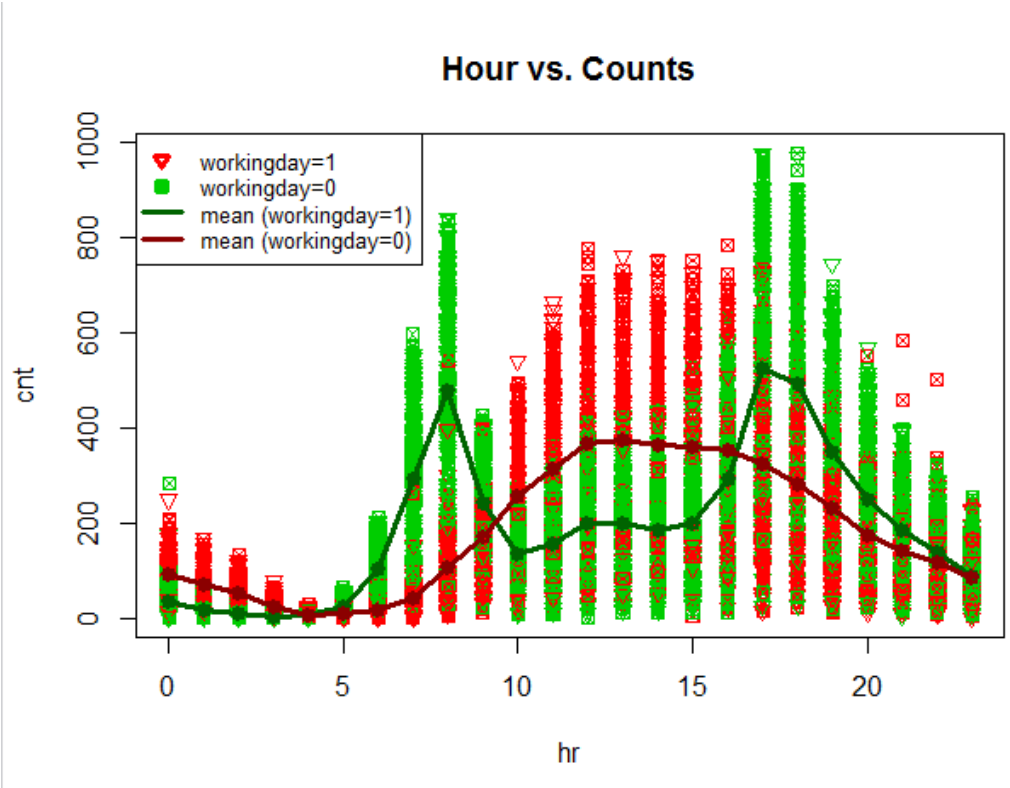


Figure 5: Hourly bike rental on different weekdays and holidays (1 indicates holiday, 0 indicates non-holiday)

From above figure, we can see that whenever there is a holiday, the bike rental shows very different pattern than usual weekdays. Sometimes the pattern is like a bike rental in weekends (Monday in the above figure).

Explore 3. Most certain the hour/workingday of the day also correlates with Bike demand. We can plot:



As we can see from the above figure, the bike rentals are concentrating from 7am to 7pm. Moreover, the working day and non-working day show a different pattern on bike rental. On workingday, the peak rental occurs at 7-8am, and 5-6pm which is the time people go to work and get off work. On non-workingday, the peak rentals occur from 10am to 4pm, which is the time families do outdoor activity. From this, a bold guess is that more registered people rent on workingdays, and more casual users rent on non-working days. We confirmed this by the following conditional plot.

Explore 4. Finally let's see if usage varies depending on the month season. We first drew a boxplot.

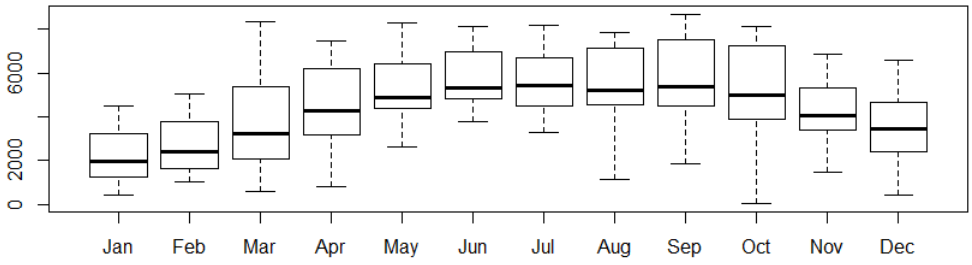


Figure 6: Boxplot of counts on different months

We can see that demand in January is the lowest and it peaks in the summer. To explore more, we drew a coplot of hourly bike counts conditioned on month.

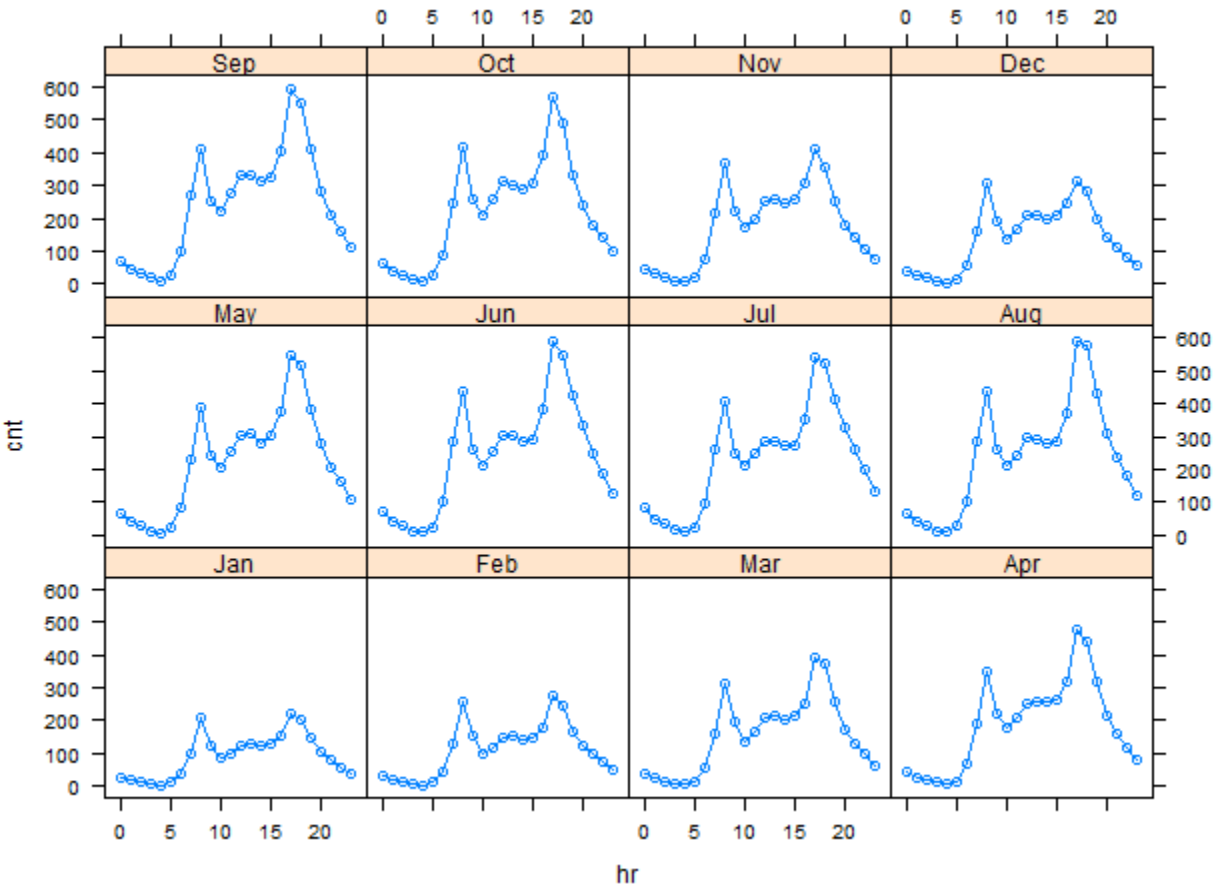


Figure 7: Average hourly bike rental on different month)

The bike rental peaks in the month from May to October. However, the pattern of hourly rental are similar with the peak is bigger in these months. It seems that during the month from May to October, more people rent bikes for commuting to work. For example, compare October and November in the above graph, the rental are very similar except during rush hours.

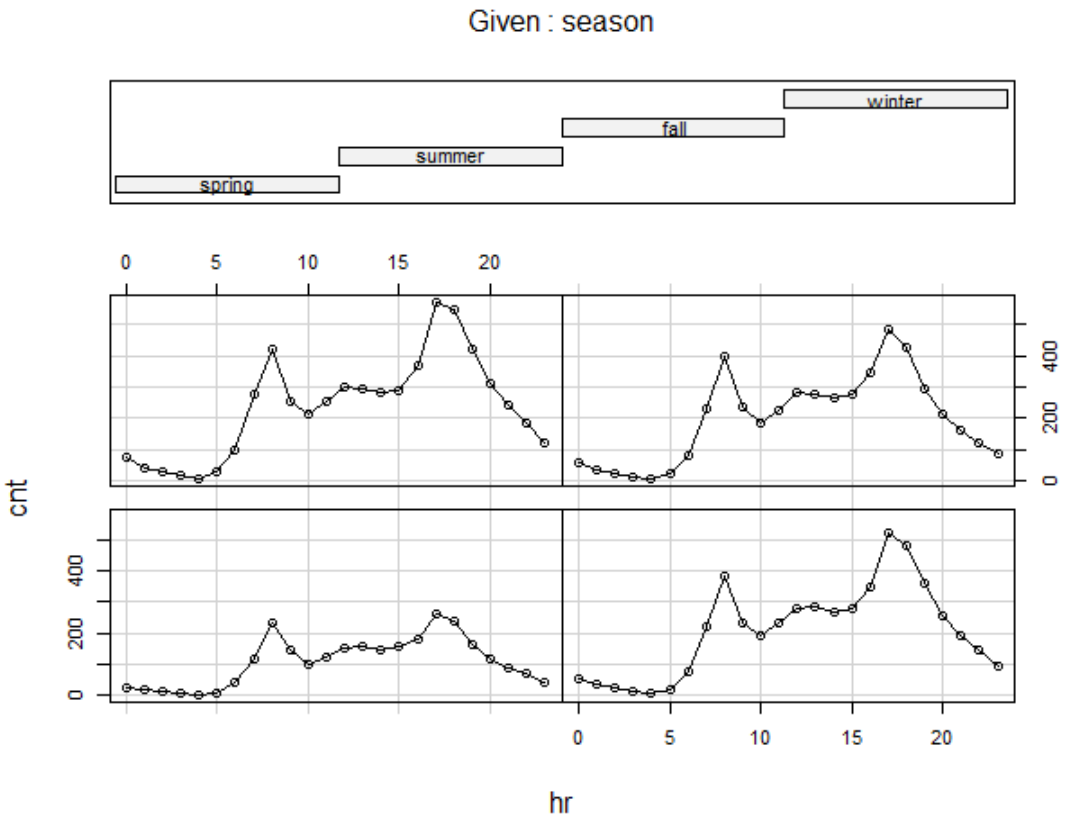


Figure 8: Coplot of hourly counts conditional on season

Let's see if different season behave differently by a parallel coordinate plot. The following are the plots from Ggobi for different seasons.

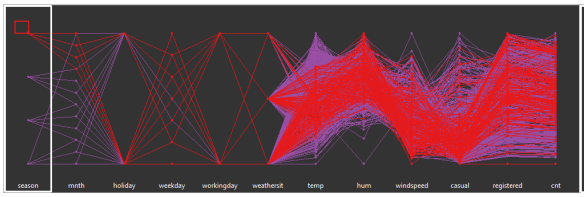


Figure 9: Spring

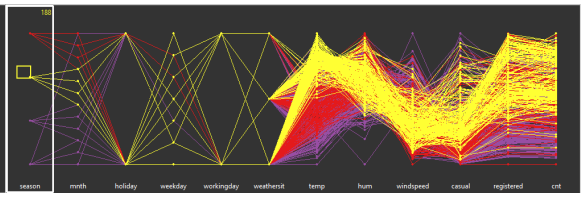


Figure 10: Summer

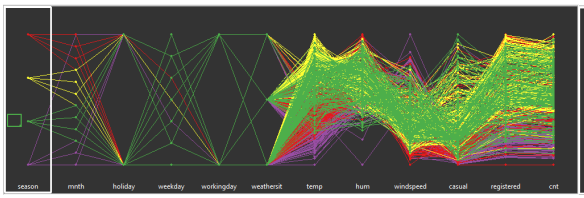


Figure 11: Fall

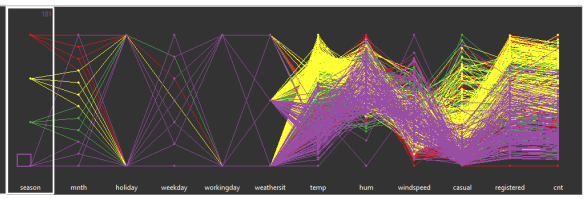


Figure 12: Winter

From the above plots, we can see that casual/registered users' demand are different in different seasons. The pattern is summarize in the following table.

Season	Spring	Summer	Fall	Winter
Registered	median, high	median, high	median, high	low, median
Casual	low	low, median, high	low, median	low

It seems that month will give more accurate information than season in the prediction. Since these two attributes are highly correlated, we will keep month in our experiment.

Explore 5. Registered and casual users' demand Vs. workingday

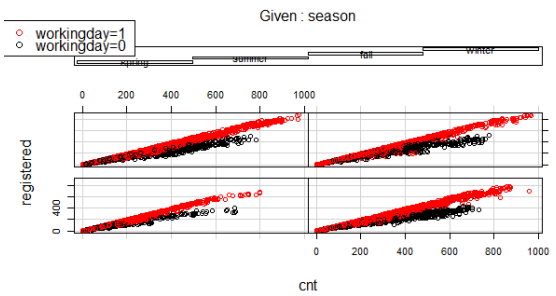


Figure 13: registered-cnt|season

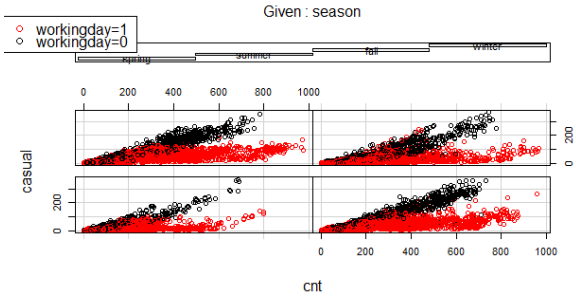


Figure 14: casual-cnt|season

From Figure 13 and Figure 14, we can see that Registered users and casual users behave differently on workingday. More registered users rent bike on workingdays, and more casual users rent bike on non-working days.

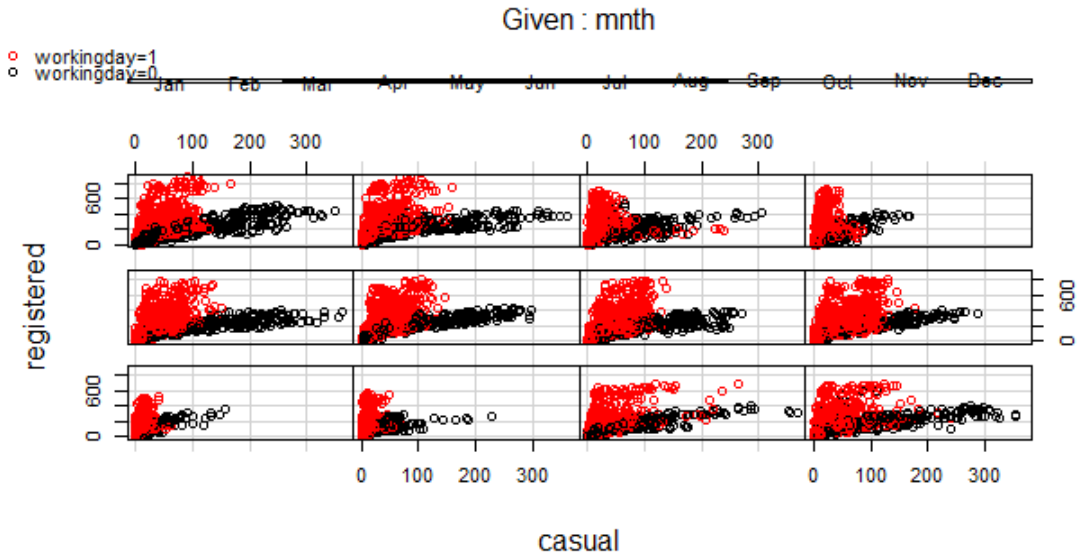


Figure 15: Coplot of registered-casual| mnth colored on workingday

In the above graph, on the non-working day, the registered and casual almost have a linear correlation(black dots). However, on workingday, there is no strong linear correlation between the two (red dots). This suggests us that we should build two separated model for workingday and non-workingday.

To find more, we plotted casual and registered for everyday as appendix A shows. Here are some interesting findings for casual users:

1. In winter, less casual users go out on weekdays, but similar amount of users go out during weekend in winter and summer.

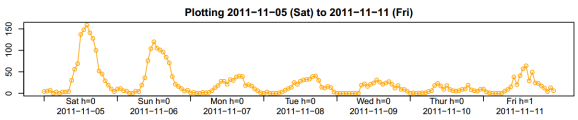


Figure 16: casual typical winter

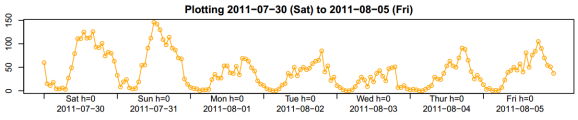


Figure 17: casual typical summer

2. The week when school started is atypical than usually summer weeks. This indicates that a large percent of casual users might be student.

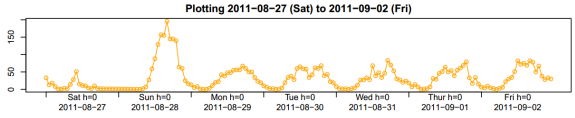


Figure 18: casual week before school

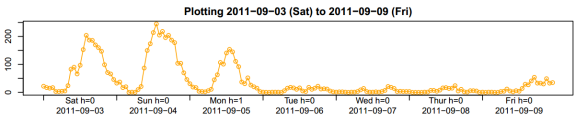


Figure 19: casual school start

3. Christmas is atypical than regular winter weeks.

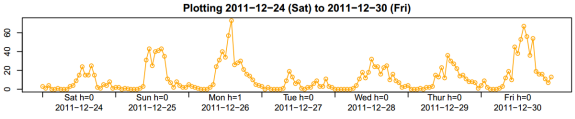


Figure 20: casual chritmas 2011

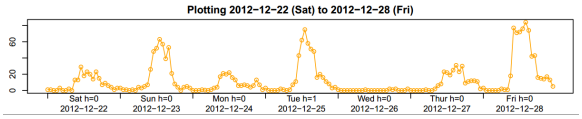


Figure 21: casual christmas 2012

4. Typical winter and summer for registered users don't change that much, except more registered users go bike during the weekends in summer.

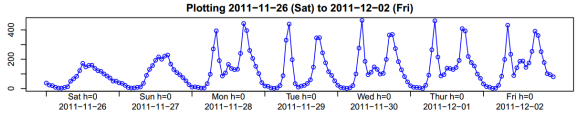


Figure 22: registered typical winter

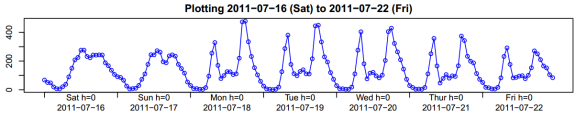


Figure 23: registered typical summer

5. Christmas is atypical than regular winter weeks.

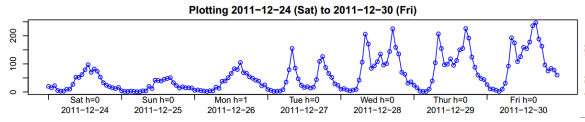


Figure 24: registered chritmas 2011

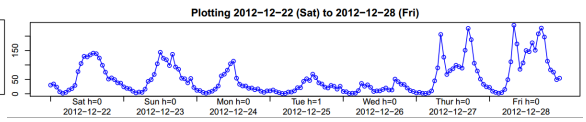


Figure 25: registered christmas 2012

Preliminary Conclusion. Based on the above observations, we identify that temp/atemp, yr, season/mnth, and workingday as the main predictors for the daily dataset, and we will investigate the relation of hr with hourly dataset.

In short we have identified strong correlation between counts and with these predictors: 1. Temperature 2. Hour of the Day 3. Working Day 4. Month of the Year.

2.3 Association Rule Mining

To investigate more about the dataset, we did an association rule mining. The bike sharing dataset contains a mixture of categorical and numeric attributes and therefore need some preparation before using apriori algorithm on it. Here are the steps for our experiment in association rule mining:

1. We fist removed the interrelate features: instant, dteday.
2. Then we mapped the seven remaining continuous attributes (temp, atemp, hum, windspeed, registered, casual, and cnt) to ordinal attributes by building suitable categories.
3. Coerce the data set to transactions as a binary incidence matrix.
4. run Apriori algorithm with support = 0.01, confidence = 0.3.

Ming the rules , and sort the rules by lift with "cnt=high" in the rhs. we obtained the following top 5 rules:

```
lhs                     rhs           support confidence     lift
1  {weekday=Sat,
casual=high}      => {cnt=high} 0.03283174          1 3.768041
2  {casual=high,
registered=high}  => {cnt=high} 0.02872777          1 3.768041
3  {hum=medium,
casual=high}      => {cnt=high} 0.03283174          1 3.768041
4  {mnth=Sep,
registered=high}  => {cnt=high} 0.03419973          1 3.768041
5  {mnth=Jun,
registered=high}  => {cnt=high} 0.03009576          1 3.768041
```

The first rule tells us that when it is Saturday and casual counts is high, there will be a large chance that cnt is high.

sort the rules by lift with "registered=high" in the rhs. we obtained the following top 5 rules:

```
lhs                rhs                   support confidence     lift
1  {weekday=Fri,
cnt=high}      => {registered=high} 0.03967168          1 3.673367
2  {weekday=Wed,
cnt=high}      => {registered=high} 0.03693570          1 3.673367
3  {weekday=Thur,
casual=medium} => {registered=high} 0.01231190          1 3.673367
4  {weekday=Thur,
cnt=high}      => {registered=high} 0.04377565          1 3.673367
5  {casual=low,
cnt=high}      => {registered=high} 0.12859097          1 3.673367
```

These five rules in general tells us that when it is weekdays, registered is high.

sort the rules by lift with "casual=high" in the rhs. we obtained the following top 5 rules:

```
lhs                   rhs              support confidence     lift
1  {season=summer,
weekday=Sat,
cnt=high}         => {casual=high} 0.01504788          1 16.61364
2  {season=summer,
weekday=Sat,
workingday=0,
cnt=high}         => {casual=high} 0.01504788          1 16.61364
3  {season=summer,
weekday=Sat,
atemp=medium,
cnt=high}         => {casual=high} 0.01094391          1 16.61364
4  {season=summer,
yr=2012,
weekday=Sat,
cnt=high}         => {casual=high} 0.01504788          1 16.61364
5  {season=summer,
weekday=Sat,
weathersit=clear,
cnt=high}         => {casual=high} 0.01231190          1 16.61364
```

Here, we can see that when it is summer weekends and cnt is high, there will be a very large chance that casual is high.

Rule 4 attracted our attention since it indicates that yr=2012 can help to predict high casual bike rental demand. We therefore plotted casual counts on yr 2011 and 2012 to compare the difference.

The association rule mining didn't give us too much more information about the dataset. But it confirms that Casual and Registered users behave differently. In the

summer weekends, there are more bike rentals from casual bikers. In weekdays, there are more bike rentals from registered bikers.

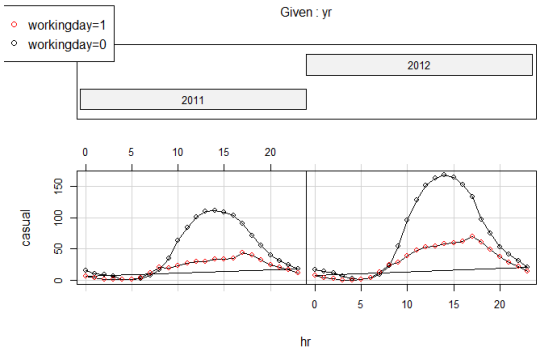


Figure 26: Coplot of casual-hr| yr colored on workingday

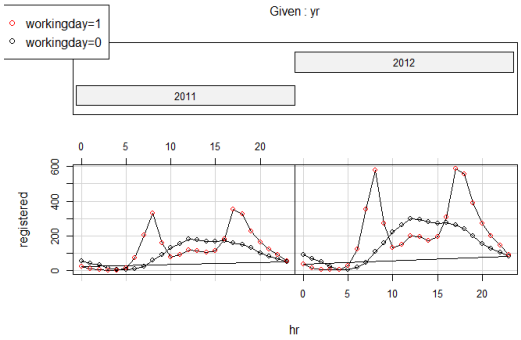


Figure 27: Coplot of registered-hr| yr colored on workingday

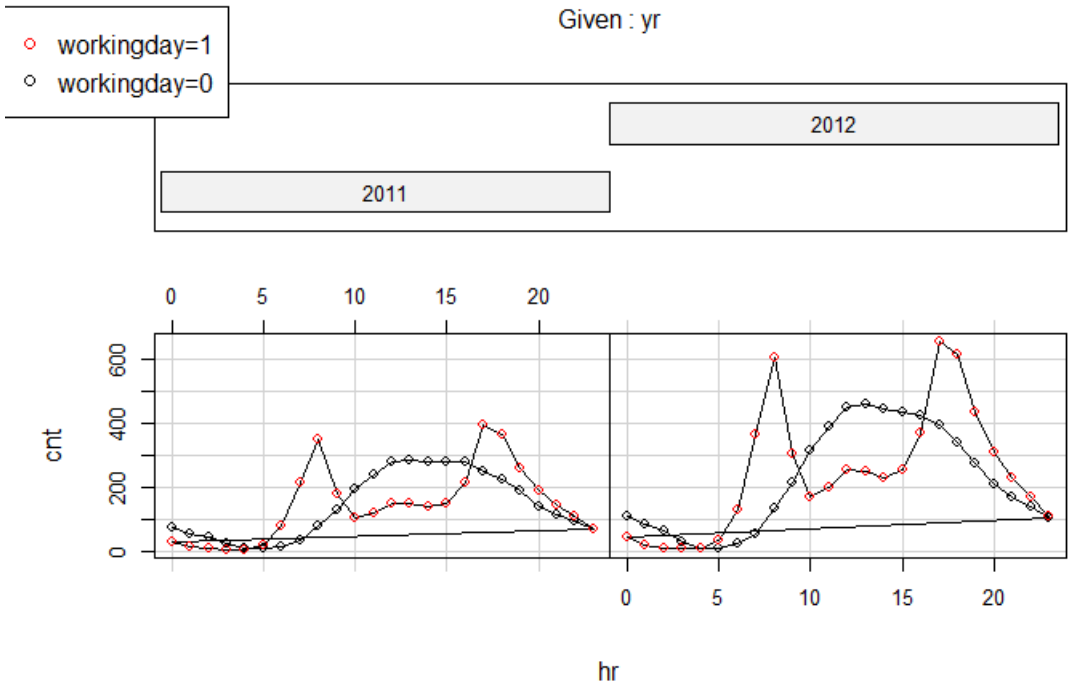


Figure 28: Coplot of cnt-hr| yr colored on workingday

From above figures, the bike rental demand from registered users and casual users all increase from 2011 to 2012 without changing on underlying pattern of the rental. Therefore, yr is also a strong predictor in bike sharing dataset.

3 Unsupervised Learning: Data Preprocessing

3.1 Dimension Reduction: PCA

3.2 Data Reduction: Clustering Kmeans + Davies-Bouldi Index

The bike sharing dataset has 17365 data instances for the hourly bike sharing dataset. Instead of dimension reduction to reduce the number of attributes, we can use data reduction to reduce the number of cases. In this section, we consider to use clustering methods to cluster together similar cases. By doing so, we can have a reduced representation in volume but produces the same or similar analytical results.

3.2.1 Data transformation

The k-means clustering uses Euclidean distance to calculate the similarity between instances, therefore the attributes of the input should be continuous numerical values. The bike sharing dataset has a mixture of categorical and numerical values, we thus need some data transformation before running k-means algorithm.

Instead of using all information to clustering the data, we considered only cluster the daily bike rental patterns. We thus feed the cnt to the k-means to group the days with similar cnt pattern together. We thus transformed the cnt in hourly dataset into days by aggregate data into days. Each day has 24 counts from hour 0 to hour 23. The following is the first 6 instances of the matrix for k-means:

```
head(bike.24hourscnt)
    hr0 hr1 hr2 hr3 hr4 hr5 hr6 hr7 hr8 hr9 hr10 hr11 hr12 hr13 hr14 hr15
1  16  40  32  13   1   1   2   3   8  14   36   56   84   94  106  110
2  17  17   9   6   3  NA   2   1   8  20   53   70   93   75   59   74
3   5   2  NA  NA   1   3  30  64 154  88   44   51   61   61   77   72
4   5   2   1  NA   2   4  36  94 179 100   42   57   78   97   63   65
5   6   6   2  NA   2   3  33  88 195 115   57   46   79   71   62   62
6  11   4   2  NA   1   4  36  95 219 122   45   59   84   67   70   62
  hr16 hr17 hr18 hr19 hr20 hr21 hr22 hr23
1   93   67   35   37   36   34   28   39
2   76   65   53   30   22   31    9    8
3   76  157  157  110   52   52   20   12
4   83  212  182  112   54   48   35   11
5   89  190  169  132   89   43   42   19
6   86  172  163  112   69   48   52   23
```

The goal is to use k-means and DBI to find a proper way to cluster the dataset. For example, as the following graph shows.

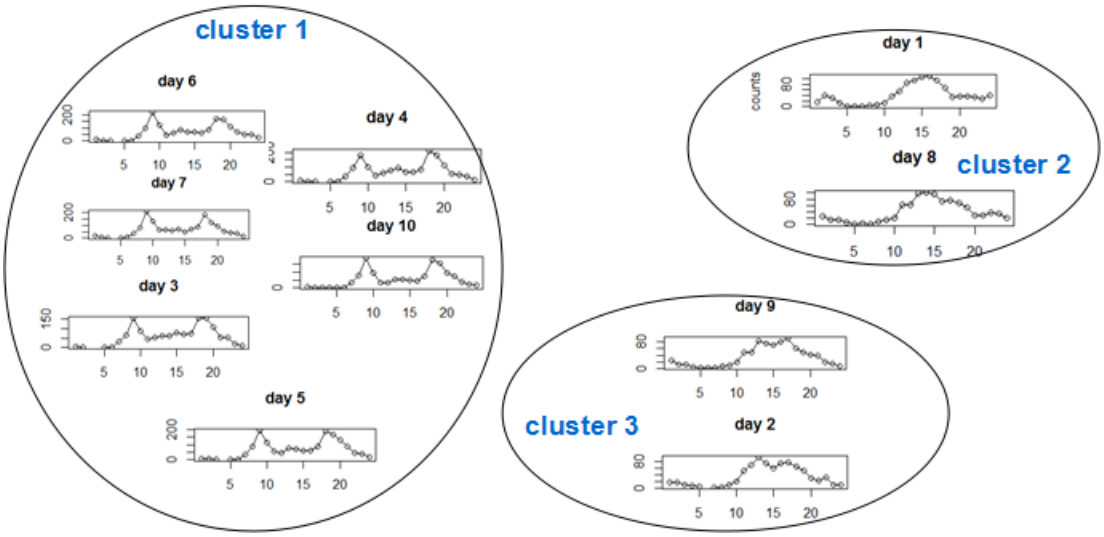


Figure 29: One example of k-means goal

3.2.2 Fill in missing values

As we can see from last section, the bike sharing dataset has lots of missing values. For example, the bike counts on the second day at hour 5 is missing. A check shows that there are 76 days with missing values, 14 days with more than 1 missing values.

We can fill up the missing values by three ways with the "zoo" package:

1. fill the missing value by last
2. fill the missing value by linear
3. fill the missing value by cubic spline

We remove the data of hour(3,5,6,10,11,15,16) on day 1 and day 10, and fill them with the three filling methods. Then we computed the Root Mean Square Error.

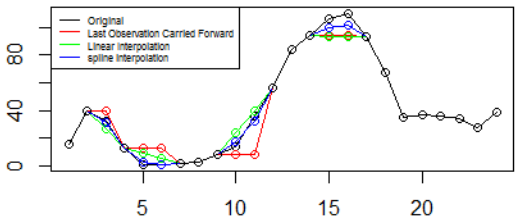


Figure 30: 3 filling methods on day 1, rmse(locf)=8.093207, rmse(linear)=5.273207, rmse(spline)=2.306192

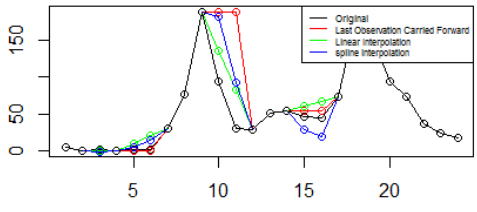


Figure 31: 3 filling methods on day 10, rmse(locf)=37.4316, rmse(linear)=15.07681, rmse(spline)=22.98093

From the rmse on workingday and non-working days, we found linear or spline interpolation fits our dataset better. We chose a combination of linear and spline

interpolation to fill the whole dataset, we first used spline to fill the dataset, then the rest miss values are filled by linear interpolation.

3.2.3 k-means

After we filled up the missing values, we performed k-means from 2 to 9 clusters. The following is the plot of these clusters with the first two principle components as the axis.

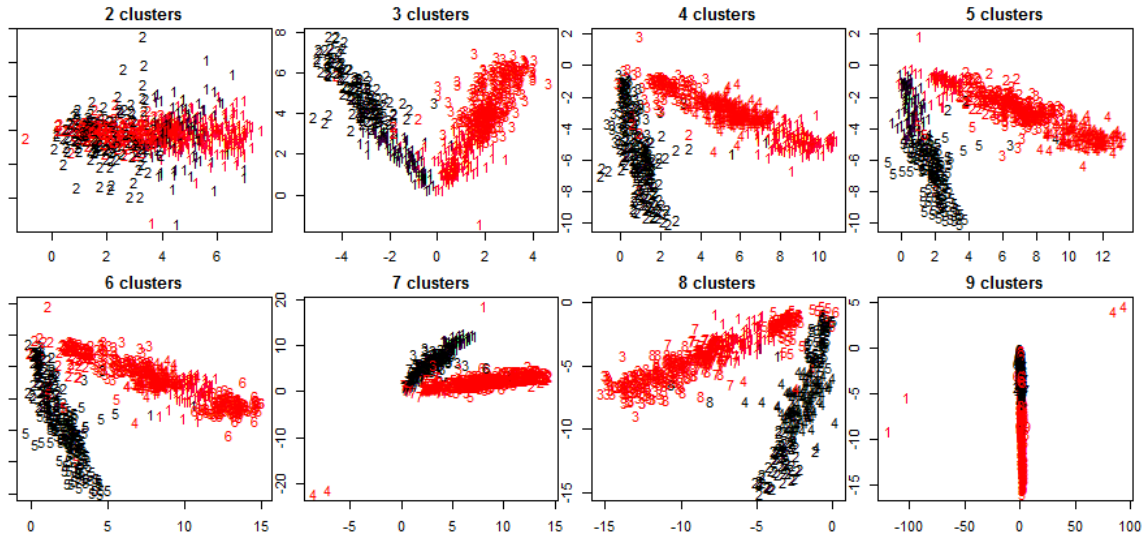


Figure 32: 2 to 15 clusters with DBI with color=red on workingday and black on non-workingday

We can see that workingday give a good separation in 3,4,5,6,7,8 clusters. To investigate what's other attributes the clustering used to separate the data. we print out the average temperature on each cluster.

Let's use k=4 as an example:

```
Cluster #1
Working days: 246 (100%)
Total days: 246
Cluster #2
Working days: 107 (53.8%)
Total days: 199
Cluster #3
Working days: 4 (2.84%)
Total days: 141
Cluster #4
Working days: 143 (98.6%)
Total days: 145
```

```
Cluster #1
Avg temperature: 0.4974
Total days: 246
Cluster #2
Avg temperature: 0.3197
Total days: 199
Cluster #3
Avg temperature: 0.5463
Total days: 141
Cluster #4
Avg temperature: 0.5776
Total days: 145
```

As we can see, cluster 1 and 4 are mainly composed of workingdays: 100% in cluster 1, 98.6% in cluster 4; cluster 3 are mainly composed of non-working days (97.26%). Cluster 2 is a mixture of working and non-workingday, But the temperature in cluster 2 is obviously lower than other clusters(most likely to be in the winter). Therefore, workingday and temp are good attributes for this clustering separation.

A drawing on average counts on each cluster confirmed our finding:

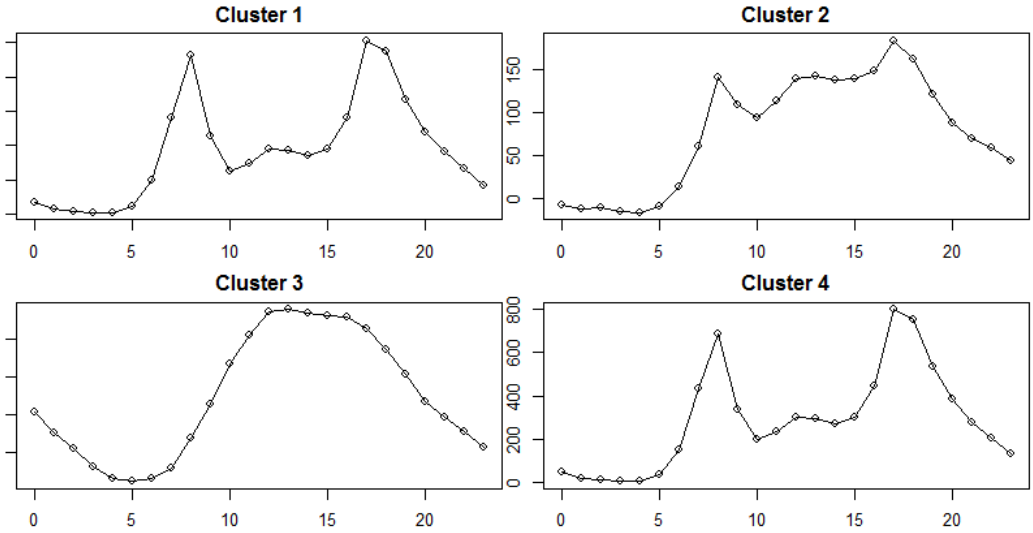


Figure 33: Average counts plot on each cluster

The following are the findings:

1. cluster 1 and cluster 4 are having the pattern of workingdays. However, the counts in cluster 1 is lower than cluster 4 which indicates cluster 1 is representative for lower temp, and 4 for higher temp. item counts in cluster 2 look like a mixture of working and non-working days with low counts. It is more likely in the winter.

2. cluster 3 are high counts with non-working day pattern.

3.2.4 DBI

To determine how many clusters we should choose for the k-means, we used Davies-Bouldin index method. After 8 runs, we find the following graph is representative with the averaged minimum DBI=8.

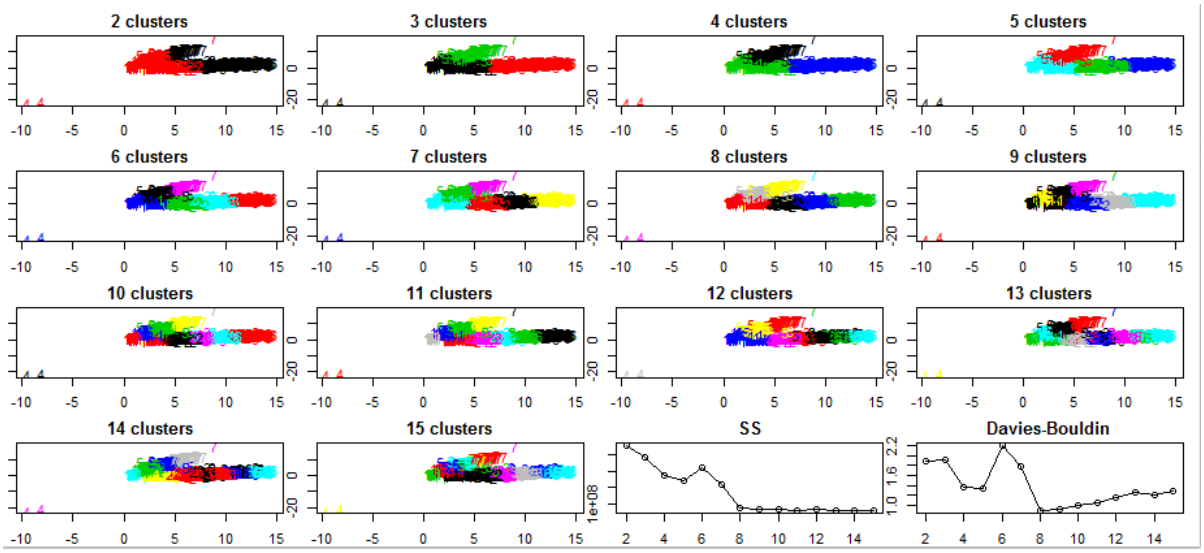


Figure 34: 2 to 15 clusters with DBI

3.3 Data Reduction: Outlier Detection

Another way to reduce the data is getting rid of extreme values. By doing so, we can find a better representation of the data. In this section, we consider to use Interquartile Range ("A filter for detecting outliers and extreme values based on interquartile ranges.") in WEKA and lofactor in R to detect outliers.

We first tried Interquartile Range in WEKA. However, this "smart" algorithm gave us 21 outliers which are all holidays:

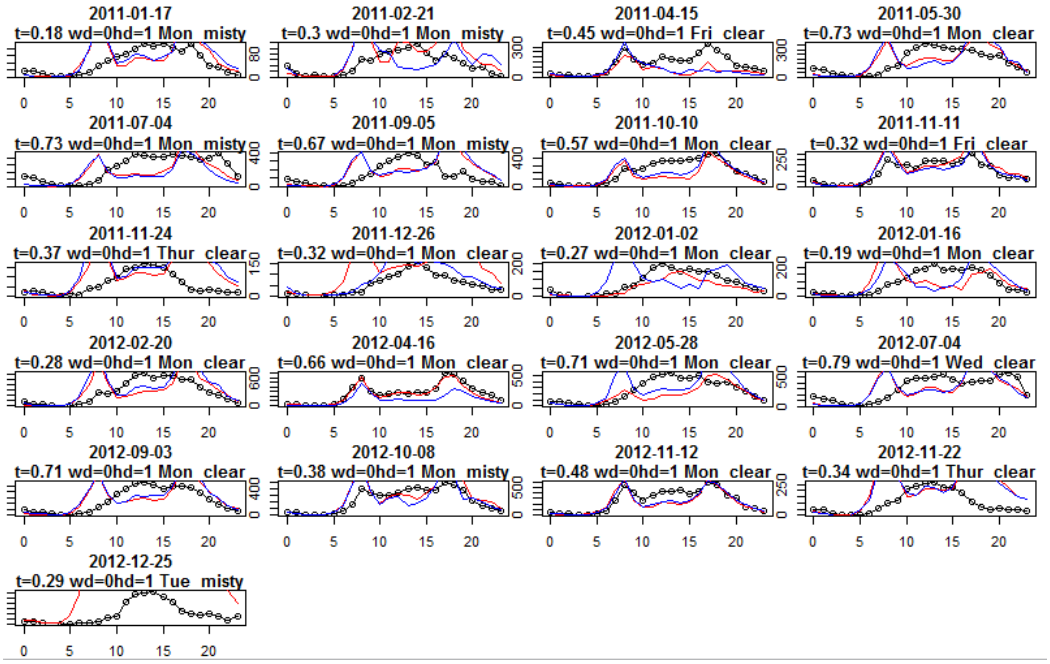


Figure 35: outliers detected by WEKA (black line is the count in this day, red is 7 days ago, black is 7 days after)

We can see that the black lines are very different than red lines (7 days ago) and blue lines (7 days after) in all 21 days. This is because all of these 21 holidays are happening during the weekdays. In holiday, the pattern is the non-workingday. And these holidays usually happen in weekdays.

We decided to remove the holiday information in the dataset, and ran Interquartile Range filter again on the dataset. Then we obtained the following days as outliers:

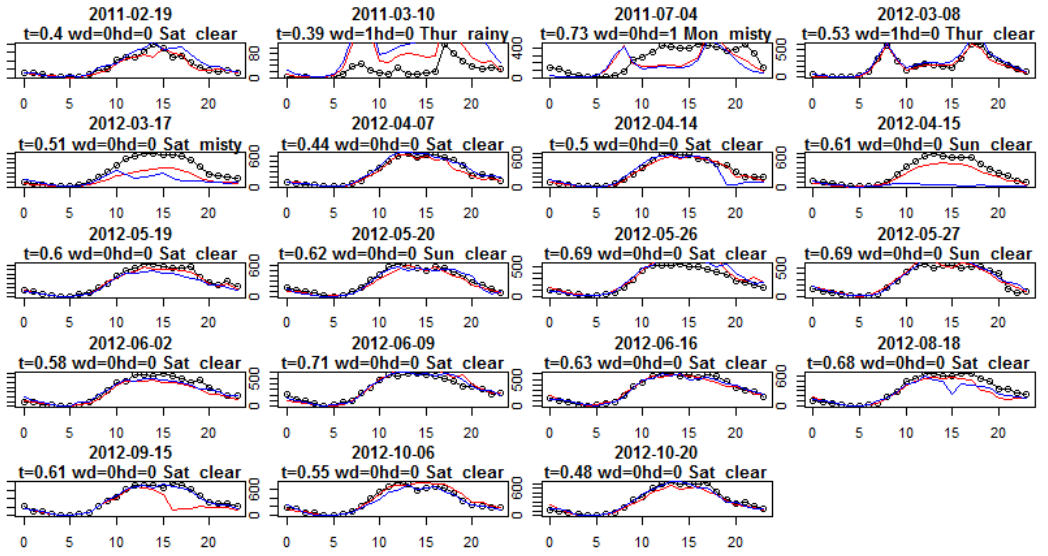
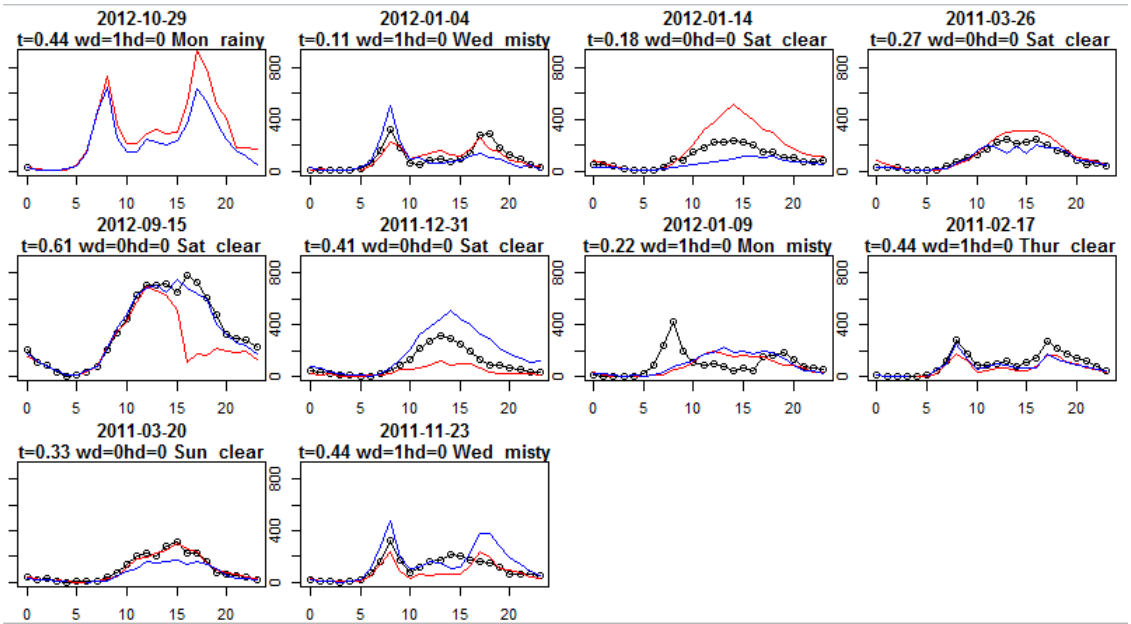


Figure 36: outliers detected by WEKA without holiday information

It seems that only on 2011-03-10 and 2011-07-04, the black lines are different than red/blue lines.



Notice that on 2012-10-29, there is a big storm based on Gama's paper about event detection [1], the detector in R caught this. We use the events detected in Gama's paper as a baseline comparison here.

Table 4 Detected Events after verification phase by background knowledge

Date	Event	Impact
29-10-2012	**Sandy**	5
30-10-2012	**Sandy**	5
19-10-2012	**Storm**	5
04-07-2012	**Washington DC fireworks**	5
23-11-2012	**Black Friday**	4
24-12-2012	**Christmas day**	4
08-10-2012	**Columbus memorial celebration**	4
27-05-2012	**Memorial day**	4
22-11-2012	**Annual Thanksgiving day**	3
12-11-2012	**Veterans day**	2
16-04-2012	**Tax day**	1
23-03-2012	National cherry blossom festival	5
18-09-2012	Heavy rain	5
18-07-2012	Severe thunderstorm	5
01-06-2012	Tornado	4
04-12-2012	Warm weather floods	4
13-05-2012	Bike DC	3
11-02-2012	Cupid Undie run 2012	3
23-01-2012	March for life	3
29-09-2012	Green festival Washington DC	3
25-11-2012	The coldest morning of the season	2
07-10-2012	Unseasonably cool weather	2
07-04-2012	D.C. United vs. Seattle Sounders FC	2
26-05-2012	D.C. United vs. NE revolution	2
21-05-2012	Occasional showers and storms	2
15-09-2012	United vs. NE revolution	2
11-10-2012	D.C. Baseball v.s Tigers	2
12-10-2012	Hockey Capitals vs. NJ devils	2
29-01-2012	Occupy DC	1
19-05-2012	Survive DC 2012	1

Bold items are verified detected events (Events with z-score ≥ 2) and non-bold items are those events that their z-score $\prec 2$. The numbers in third column is the impact rate (from 0 to 5) given by a human domain specialist for that date indicating the impact of event

Figure 37: red: events found by detector 1, yellow: events found by detector 2, green: events found by detector 3

As we can see from the above table, our outlier detectors do help find some events which influence the bike rental. We can eliminate these days found by the three

outlier detectors. However, we are also removing points which might be useful since there might be a lot of false alarms on the days detectors found.

A better way to approach is separate data into day of events or not and train models separately on these two datasets. When a new data instance come, we first let the detector decide if it is event day, and pass it to correct model.

4 Experimental Design

5 Experiments and Results

6 Supervised Learning

The most obvious supervised learning task is to predict `cnt`, which is a regression problem, since `cnt` is a continuous variable. This has practical applications, as any company managing a bicycle sharing system might be interested in having a predictive models that helps ensure the supply of bicycles is always appropriate.

6.1 Kaggle Competition

The UCI Bike Sharing Dataset is featured as a Kaggle problem called *Bike Sharing Demand*[1]. We decided to participate in the competition, so that we could compare the performance of our method with others.

The competition started on 28 May 2014 and ends on 29 May 2015. As of 13 April 2015, approximately 2900 people or team have participated in this competition, submitting more than 26000 entries.

The Bike Sharing dataset provided by Kaggle is essentially the same as UCI's – there are only minor differences in the format, such as the date and hour being specified by only one column named `datetime` and non-normalized units used for temperature, humidity and wind speed.

The data must be split into a training set and a testing set, where the training set is comprised of the first 19 days of every month and the testing set is comprised of the 20th day until the end of the month.

The objective of the competition is to train a classifier on the training data and to predict the testing data's `cnt` field with the lowest possible root-mean-squared-logarithmic error (RMSLE).

Users can submit their predictions online and the RMSLE will be calculated

Note that the testing data on Kaggle excludes the `cnt` field. As such, users must submit their predictions only to discover its RMSLE. However, the full data is present in the UCI's dataset and so, we can compute the RMSLE of our predictions by ourselves.

[1] www.kaggle.com/c/bike-sharing-demand

7 Data preparation

WILL MOST LIKELY NEED TO MOVE THIS SOMEWHERE ELSE AND JUST REFER TO IT HERE.

The relationship between `atemp` and `cnt` is non-linear, as can be seen from GRAPHREF. However, there seems to be a peak temperature for bike rentals at around PEAKTEMP. Hence, it would make sense to define a new variable corresponding to the (absolute value) difference between the current temperature and this "ideal temperature". Let `atempdiff` be defined as:

```
atempdiff <- abs(atemp - PEAKTEMP)
```

Then, the relationship between `atemptdiff` and `cnt` is roughly linear as we can see from OTHERGRAPH.

IDEM with `hum`, except the "ideal humidity" is at around 0.2.

8 Data Imputation

We observed that some rows are missing in the hourly Bike Sharing datasets. In total, 165 rows are missing, which is almost 7 days worth of data. We decided that it would be wise to impute the missing data, since it could help build a better regression model. Moreover, having no missing data make the R programming significantly easier since the dataset structure would become more regular.

We decided to use the `zoo` package to handle data imputation. The numeric variables were imputed using linear interpolation. We estimated that this was sufficient for most variables such as temperature, humidity and windspeed. However, for `casual`, `registered` and `cnt`, this did not always work. In most cases, only 1 or 2 observations were missing in a 24-hour period, in which case the linear interpolation was appropriate. However, there are 8 days with a large number of missing observations (between 6 and 23 missing observations per 24-hour period). We decided to investigate visualize the results of our imputation method for those days (see Figure **??**).

Figure 38: Imputation of missing observations for the 8 days with the highest number of missing observations. The **black** curve represents the actual values of `cnt` from the dataset. The **red** curve are the observations that were imputed using linear interpolation. The **green** curve represents the average value of `cnt` at the same hour on the same day of the week over the same month (the **green** curve therefore provides a rough idea of what a typical day looks like and helps us decide if the imputation is acceptable or not).

labelfig:badrows

As you can see, in most cases, linear interpolation does a poor job. Indeed, there are too many missing values and the pattern they cover is expected to be highly

non-linear, except for the dates 2011-02-22 and 2011-08-28, in which the missing observations are all at night, where the pattern of bikes rental is flat.

At this point, we decided to flag the 6 days above with poor imputation results as "bad rows". This allows us to exclude those days completely in the training phase of supervised learning.

8.1 The two alternative problems on Kaggle

Unfortunately, the problem of modeling and predicting bike rental demand as stated on Kaggle as two different interpretations.

9 Conclusion and Further Work

References

[1] 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.

¿¿¿¿¿¿¿ Stashed changes