Predicting bike sharing demand in Washington, D.C.

Kelly Cercy, Madison Hassler, Emma Kortemeier
Furman University, CSC 272-Dr. Treu

Introduction

Bike sharing started in Europe in the 1960's but has since grown in popularity here in the United States due to its convenience, affordability, and sustainability. According to the Bikesharing World Map, there are now 553 bike share programs in operation worldwide and another 193 in planning or under construction including one now in Greenville, SC. Capital Bikeshare launched in Washington, D.C in August 2008 and currently operates 2500 bicycles at over 300 locations across D.C and the surrounding areas. Bike sharing operates from a system of self-service bike stations operating 24/7, 365 days a year--bikers can rent a bike, ride to their destination, and return the bike to any docking location near their destination! As such, datasets from bikeshares have been popular among researchers because the times and dates are explicitly recorded and are often used for sensing mobility in the city.

This project attempts to predict bike rental demand for Capital Bikeshare in Washington, D.C. based on usage patterns from 2011-2012 combined with weather data. We used classification learning and numerical estimation data mining techniques to understand what attributes had the most significant impact on the scale bike rentals daily and hourly, and predict bike rentals based on the values of these attributes. From our investigations we were able to learn that time of day and feels like temperature are the most predictive attributes and the year also is very predictive.

Dataset Description

The two datasets used in this project were constructed by UCI's Center for Machine Learning and Intelligent System and obtained from the Machine Learning Repository website at http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset. The data contains a count of rental bikes (both daily and hourly) between the years of 2011-2012 from the Capital Bikeshare program in Washington, D.C from http://www.capitalbikeshare.com/system-data and has been combined with weather data to produce the data set we will be using.

The first dataset on daily bike rentals contained 731 instances with 16 attributes. Our second data set on hourly bike rentals contained 17379 instances with 17 attributes due to the addition of Hour. We modified the format of the data slightly, but all information given by the original dataset was retained in the final version used for mining and analysis. Attributes with a description and corresponding values are presented below with an asterisk marking Hour which is only included in the hourly data set:

Attribute	Values
Instance	1 to 17389/731 for each instance
Date	month/day/year
Month	1 to 12 (for each month of the year, 1:January)
Year	0:2011, 1:2012
Hour*	0 to 23 (for each hour of the day, 0:midnight)
Season	1:spring, 2:summer, 3:fall, 4:winter
Holiday	0:not holiday, 1:holiday
Day of Week	0 to 6 (for each day of the week, 0:Sunday)
Working Day	0:not a weekend or holiday, 1:weekend or holiday
Weather	1:clear, few clouds, 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 pellets+thunderstorm+mist, snow+fog
Temperature	normalized temperature in celsius
Feels Like Temperature	normalized feels like temperature in celsius
Humidity	normalized humidity
Wind Speed	normalized wind speed
Casual	number of non-registered users who rent a bike
Registered	number of registered users who rent a bike
Count	number of total rentals (casual and registered)

Data Preparation

We took several steps to prepare the data for mining an analysis. We first converted our .csv file to an .arff file in Weka. As many of our data fields were numeric with a large number of possible values, we needed to discretize our data. Originally, we used Weka to perform an equal-length discretization of all the attributes. However, we felt as though Weka's discretization was not appropriate as each attribute was being put into the same number of bins and attributes such as season or holiday were either ordinal or binary, so we discretized each attribute in our .csv file by using if statements in Excel before saving as a new .arff file in Weka. Specifically to discretize count in our daily data we discretized it into three bins, low [0,3744), medium [3744,5314) and high [5314, 8714]. To discretize the count in our hourly data we discretized it into four bins one [0,98), two [98,189), three [189,321), and four [321,977]. We first tried discretizing the Hourly count by dividing it into five bins of equal length but since over 80% of the data fell into the first

bin we could not obtain any reasonable results so we tried to create five bins with equal frequency but to maintain some weight on the more frequent lower instances we combined the first two bins thus arriving at our four bins given. The hour attribute was also discretized into four categories early morning (12am-6am), morning (6am-12pm), afternoon(12pm-6pm) and evening (6pm-12am). Of the remaining attributes, Feels Like Temperature, Humidity, and Windspeed were discretized into ten bins since they were normalized data for both datasets.

Next, we deleted the attribute Date because each "/" disrupted the data mining process in Weka and we were also able to see the individual effects of each attribute through Year and Month. We also deleted Instance to avoid overfitting the data. Then we deleted the Day of the Week and Working Day attributes to avoid redundancy with Holiday, the Casual and Registered attributes to avoid redundancy with Count, and the Temperature attribute to avoid redundancy with Feels Like Temperature (and since we are talking about human behavior, the "real feel" temperature seemed more pertinent). As each of these attributes are dependent on the other, we felt it was unnecessary to keep both, for fear of skewing the data.

Once the final datasets were loaded into Weka, we used Weka to randomize the data and split it 80%/20% into a training set for creating the model and testing set for each dataset to test the resulting model's accuracy.

Data Analysis

We looked to analyze our bike sharing dataset in two ways: first, to predict what the range of bike demand will be for a particular day given the above attributes, and second, to estimate the exact bike demand for each day.

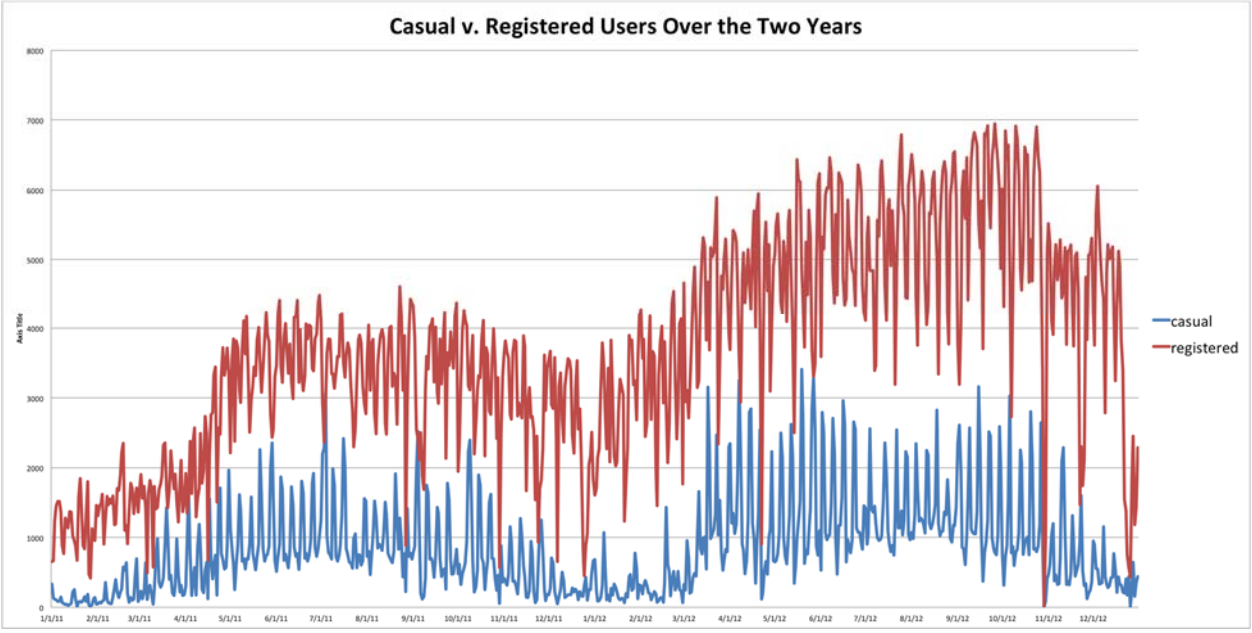
We explored our first question using classification learning algorithms in Weka, focusing specifically on J48 and using OneR at first to better understand the datasets. OneR is used to create a model from the one attribute rule that gives the highest accuracy, and J48 is an algorithm that creates a pruned decision tree. OneR gave us an initial rule that the dataset split on to give us an idea of what our datasets' important attributes are. J48 gave us a more sophisticated look at how each of the datasets further classified each test instance. For the daily dataset in order to create a more readable tree we decreased our confidence factor to 20%. For the hourly dataset since there were so many instances, to create a tree with less than 100 leaves we increased the minimum number of objects to 125. Each of these actions decreased our accuracy, but it was worthwhile in order to create a model that was usable.

We then looked to estimate bike demand using numerical estimation. We chose to use both the Linear Regression algorithm and M5P algorithms on our hourly and daily

datasets. To do this we took the discretized hourly and daily datasets and changed the class attribute, Count, back to its non-discretized values in each, in order to have a numerical class attribute. The Linear Regression algorithm creates a linear function model using all of the given attributes-computing a line of best fit for the multiple variables, while the M5P algorithm generates a regression tree, a decision tree with equations on the leaves which calculate values of the output attribute. We chose to generate a pruned tree with M5P.

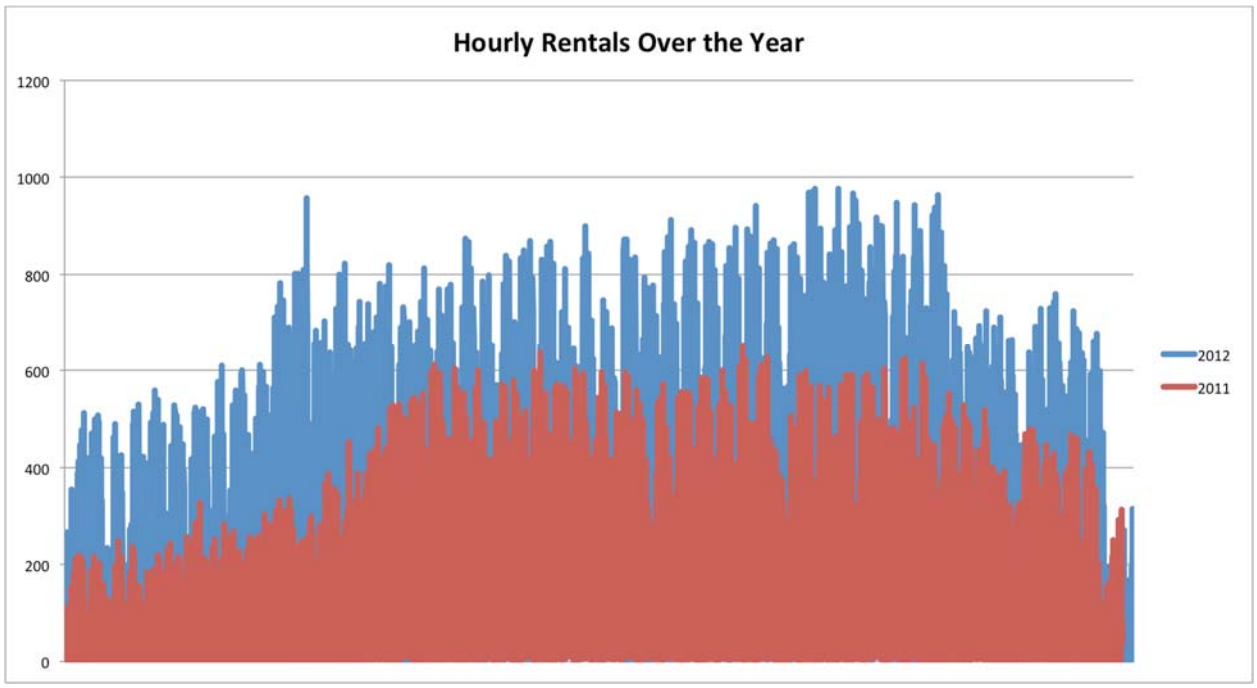
Results

First, in order to get to know our datasets a little better, we created some visualizations of different attributes. Here we have the Casual User and Registered User counts for each day during the two year period.



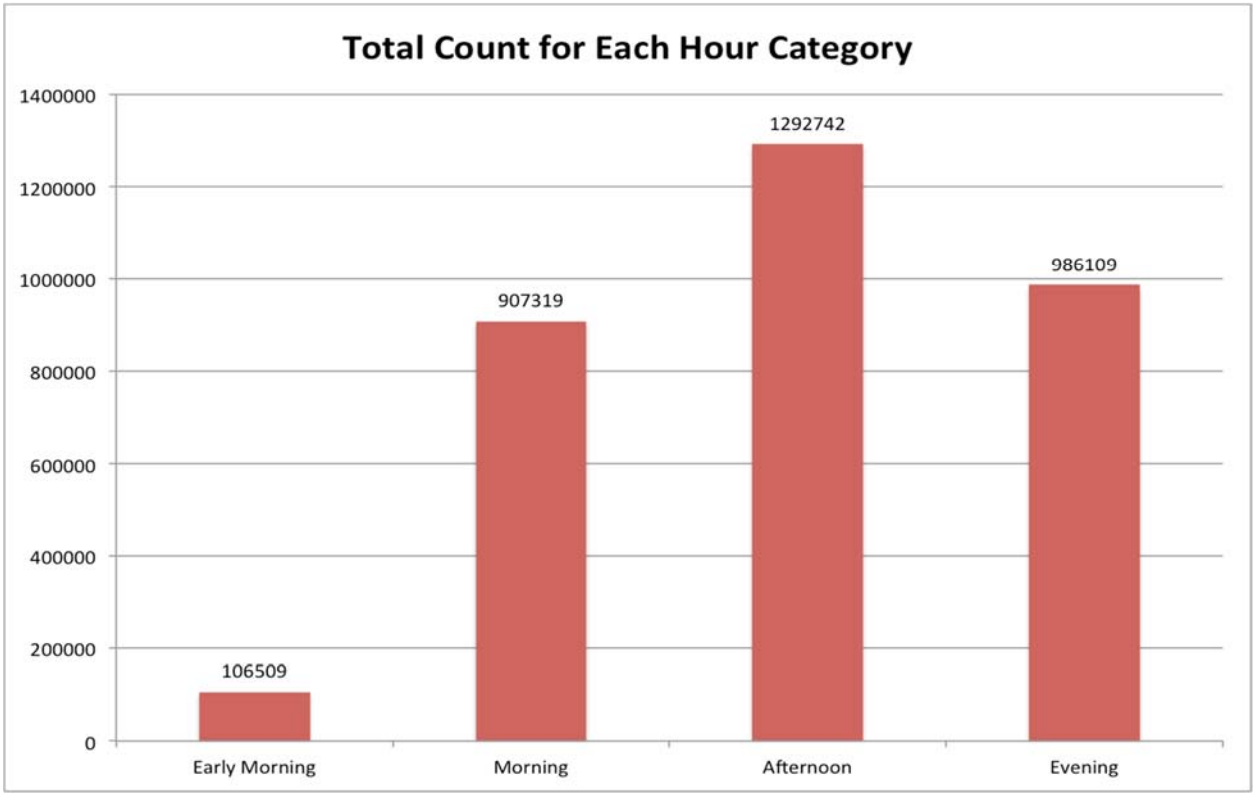
From this graph, it can be seen that for the majority of the time, the Registered User count (the red line) is higher than the Casual User count. Based on this, in order to increase their profit, the company might consider making this bike sharing system more approachable for people who are not already a part of the system. Based on the above chart, we can also conclude that both Casual and Registered User count increased throughout the second year. This is good news for the company since this means that they were able to attract more customers.

In our quest to learn more about our data, we also created a visual of the Hourly dataset. We plotted the Hourly count (Registered and Casual counts combined) for every hour for each of the two years.



This visual confirms what the first chart suggested. It is clear that the Count for the second year (2012) was consistently higher than the Count for the first year (2011). This tells us that the company did a better job marketing the bike sharing program and its popularity grew from 2011 to 2012. We can also see that each year followed a similar trend with lower rates of usage at the very beginning of the year and the very end of the year (the colder months).

Here is another visual from the hourly dataset that depicts the total count for each hour category (Early Morning, Morning, Afternoon, and Evening).



We can see that there are significantly less rentals during the Early Morning compared to each of the other categories. We can also see that the Afternoon is the most popular time for bike rentals.

When we applied OneR to the hourly dataset we obtained the following rule with 35.328% accuracy:

Hour: Early Morning → one

Morning → two

Afternoon → four

Evening → four

This gave us an initial idea that hour was one of the most influential attributes for the hourly dataset. This is also backed up by the strong variances seen in the total counts for each hour category in our previous visual.

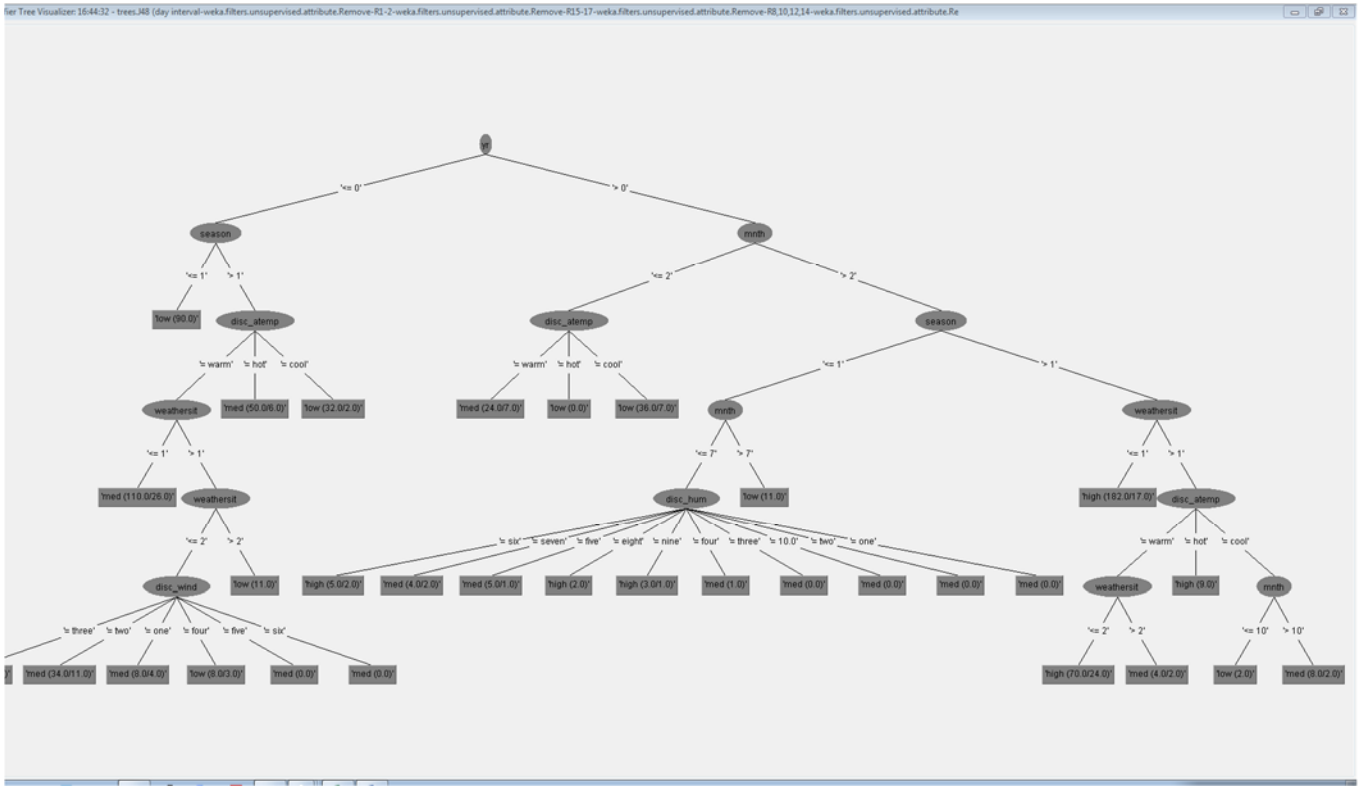
Similarly, when we applied OneR to the Daily dataset we obtained the following rule with 55.212% accuracy:

Year:

< 0.5 → med

≥ 0.5 → high

For the Daily dataset the J48 tree produced with the holdout method mentioned earlier and a coverage of 20 percent is shown below.



This tree predicted the Count (low, medium, or high) with 77.39% accuracy. Note that J48 split on year first, meaning it is the most influential attribute. This is logical since this is also the attribute that OneR picked for the rule. The first two line graphs also present a visual of the marked differences between the first and second year, so this is right in line with what we expected to see.

When we ran J48 on the hourly data set we could not get a tree that was small enough to fit on one screen without drastically reducing the accuracy so instead we aimed for a tree with less than 100 hundred leaves. We obtained a final tree with 81 leaves and 59.1484% accuracy (to see the full tree in text format see the appendix). Our initial intuition that hour was an influential attribute obtained from our OneR analysis was backed up by J48 since it also chose hour as the first attribute to split on. The attribute most frequently split on after that was feels like temperature and then the weather attribute. Some of the interesting rules pulled from the tree include

If Hour=Early Morning → Count=one

If Hour=Evening and Feels Like Temp=eight or nine → Count=four

If Hour= Evening and Feels Like Temp=ten → Count=three

The first rule intuitively makes sense because there are likely not many people riding bikes between the hours of 12am and 6am. Again this is backed up by our bar graph visualization. The following two rules tell us that people are riding bikes a lot when the weather is warm (eight or nine) but if it is too hot (ten) then there are fewer people riding. Whenever the tree split on year 2011 would always lead to a lower predicted count than 2012. This backs up what we saw in our visual that in 2012 there were far more bikes rented than in 2011.

We first ran the Linear Regression algorithm on the training sets for daily and hourly bike rentals to generate a linear regression function with any applicable attributes, and tested the model in Weka on the test set of instances. The correlation results are shown below and the regression formulas can be seen in the appendix.

	Daily	Hourly
Correlation coefficient	0.3942	0.6952
Mean absolute error	885.2003	123.1274
root mean squared error	1217.3361	165.758
relative absolute error (%)	25.1805	70.3647
root relative squared error (%)	34.2001	71.2605
total number of instances	146	3476

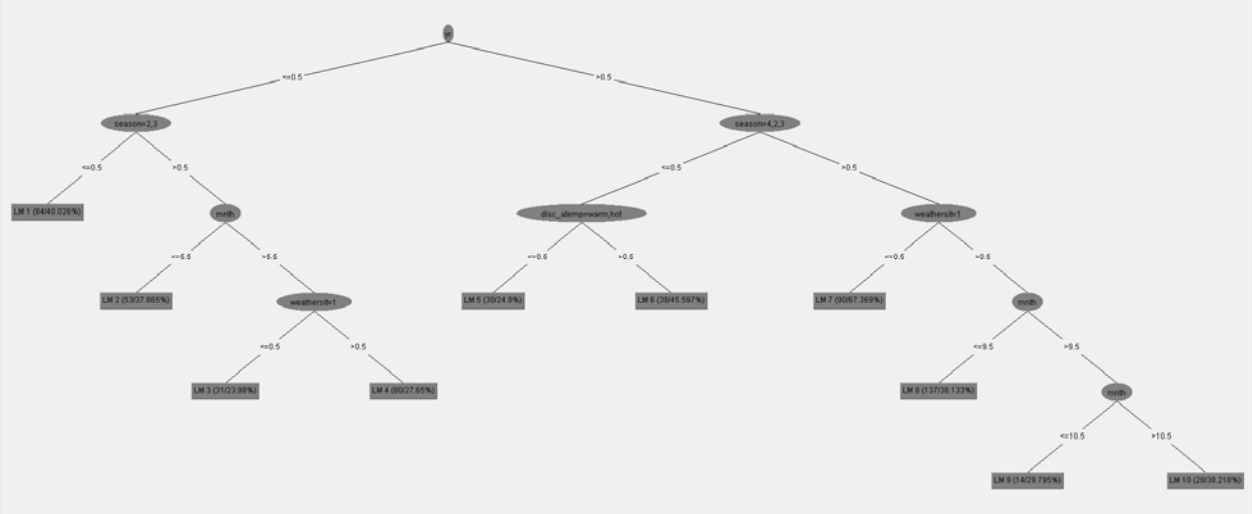
The linear regression function created using the daily dataset had a significantly higher accuracy, but a lower correlation coefficient. We can see from both functions that the demand for bike sharing is dependent on many attributes, and cannot be accurately determined by one, or even a few attributes. However, examining the coefficients assigned to the attributes does reveal which attributes of the date and weather seem to have the most and least effect on rank. In the daily dataset, we see that Year has by far the largest impact, followed by Season, as Month has the least impact. In the hourly data set, Hourly has the largest impact, followed by Year, as Month again has the least impact.

We then ran the MP5 algorithm on both the training and hourly the training set of instances to generate the regression tree, and tested this model in Weka on the test set of instances. The correlation results and regression trees for both daily and hourly datasets are displayed below.

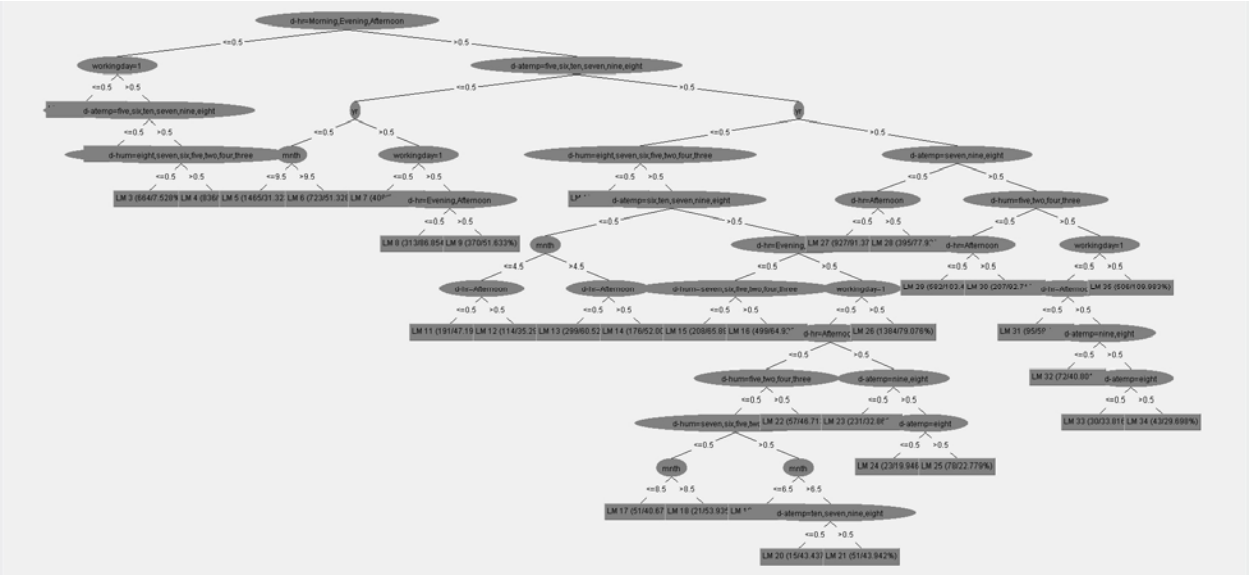
	Daily	Hourly

Correlation coefficient	0.2502	0.6489
Mean absolute error	1388.3653	116.2941
root mean squared error	1589.8513	170.7413
relative absolute error (%)	39.4935	66.4596
root relative squared error (%)	44.6657	73.4029
total number of instances	146	3476

Daily:



Hourly:



Overall there was not a strong linear correlation between our attributes (as we can see through the low correlation coefficient), but this intuitively makes sense because count is a very specific variable and small differences such as 18 versus 19 bikes rented in the grand scheme really do not matter. Regardless though our regression analysis does

give some valuable information and it further backs up our previous results obtained from J48 and OneR.

Conclusions

Overall we have seen that in our hourly dataset that the time of day is the most predictive attribute of the number of bike rentals over the span of our dataset. Also from both datasets we could see that the feels like temperature was also highly predictive. In addition we have seen conclusive evidence that 2012 was a much more successful year than 2011. We also had much more success predicting ranges of rentals rather than a specific number of rentals. In general the results of our models reflect our intuition. People want to be outside when the weather is nicer and at hours of the day when it is light outside.

Appendix

Hourly J48 Tree:

```
d-hr = Early Morning: one (3453.0/165.0)
d-hr = Morning
|   d-atemp = three: one (516.0/310.0)
|   d-atemp = four
|   |   yr <= 0: one (164.0/106.0)
|   |   yr > 0: two (335.0/228.0)
|   d-atemp = five
|   |   yr <= 0: two (197.0/121.0)
|   |   yr > 0: four (373.0/242.0)
|   d-atemp = two: one (104.0/47.0)
|   d-atemp = one: one (8.0/3.0)
|   d-atemp = six
|   |   d-weathersit = Partly Cloudy/Clear: four (340.0/228.0)
|   |   d-weathersit = Cloudy/Misty: four (253.0/170.0)
|   |   d-weathersit = Light Precipitation: one (103.0/61.0)
|   |   d-weathersit = Heavy Precipitation: two (0.0)
|   d-atemp = seven
|   |   yr <= 0: two (395.0/243.0)
|   |   yr > 0: four (448.0/239.0)
|   d-atemp = eight: two (197.0/129.0)
|   d-atemp = nine: two (45.0/26.0)
|   d-atemp = ten: two (1.0)
d-hr = Afternoon
|   d-weathersit = Partly Cloudy/Clear
|   |   d-atemp = three: two (112.0/52.0)
|   |   d-atemp = four: three (234.0/136.0)
|   |   d-atemp = five: four (386.0/224.0)
|   |   d-atemp = two: two (16.0/8.0)
|   |   d-atemp = one: two (2.0)
|   |   d-atemp = six: four (244.0/106.0)
|   |   d-atemp = seven: four (779.0/308.0)
|   |   d-atemp = eight: four (407.0/193.0)
|   |   d-atemp = nine: three (158.0/103.0)
|   |   d-atemp = ten: two (12.0/7.0)
|   d-weathersit = Cloudy/Misty
|   |   d-atemp = three: three (64.0/35.0)
|   |   d-atemp = four: three (124.0/72.0)
|   |   d-atemp = five: three (154.0/96.0)
|   |   d-atemp = two: two (17.0/8.0)
|   |   d-atemp = one: three (0.0)
|   |   d-atemp = six: three (134.0/75.0)
|   |   d-atemp = seven: four (246.0/131.0)
|   |   d-atemp = eight: four (81.0/45.0)
|   |   d-atemp = nine: three (13.0/6.0)
|   |   d-atemp = ten: three (0.0)
|   d-weathersit = Light Precipitation: one (302.0/184.0)
|   d-weathersit = Heavy Precipitation: four (0.0)
d-hr = Evening
|   d-atemp = three: one (305.0/133.0)
|   d-atemp = four
```

```
|   |   yr <= 0: one (194.0/102.0)
|   |   yr > 0: two (361.0/220.0)
|   d-atemp = five
|   |   d-weathersit = Partly Cloudy/Clear: two (393.0/246.0)
|   |   d-weathersit = Cloudy/Misty: two (133.0/89.0)
|   |   d-weathersit = Light Precipitation: one (65.0/31.0)
|   |   d-weathersit = Heavy Precipitation: two (0.0)
|   d-atemp = two: one (53.0/11.0)
|   d-atemp = one: one (5.0/2.0)
|   d-atemp = six
|   |   d-weathersit = Partly Cloudy/Clear
|   |   |   yr <= 0: two (134.0/84.0)
|   |   |   yr > 0: four (234.0/150.0)
|   |   d-weathersit = Cloudy/Misty: three (183.0/120.0)
|   |   d-weathersit = Light Precipitation: one (94.0/50.0)
|   |   d-weathersit = Heavy Precipitation: three (0.0)
|   d-atemp = seven
|   |   d-hum = nine: two (65.0/33.0)
|   |   d-hum = eight: three (214.0/128.0)
|   |   d-hum = ten: four (0.0)
|   |   d-hum = seven: four (215.0/129.0)
|   |   d-hum = six: four (227.0/124.0)
|   |   d-hum = five: four (156.0/62.0)
|   |   d-hum = four: four (92.0/21.0)
|   |   d-hum = three: four (41.0/8.0)
|   |   d-hum = two: four (0.0)
|   |   d-hum = one: four (0.0)
|   d-atemp = eight: four (263.0/123.0)
|   d-atemp = nine: four (58.0/27.0)
|   d-atemp = ten: three (1.0)
```

Daily Regression formula:

```
cnt =
          1336.5623 * season=4,2,3 +
           273.3869 * season=2,3 +
           439.1347 * season=3 +
          2091.8138 * yr +
            43.4198 * mnth +
          1165.3339 * weathersit=2,1 +
           558.9076 * weathersit=1 +
          1174.9562 * disc_atemp=warm,hot +
          -512.4092 * disc_atemp=hot +
           650.0731 * disc_hum=nine,eight,one,two,five,six,four,seven,three +
           268.9908 * disc_hum=two,five,six,four,seven,three +
           354.2029 * disc_wind=three,six,one,two +
           144.7483 * disc_wind=one,two +
         -1497.6504
```

Hourly Regression Formula:

```
cnt =
           42.317  * season=4,2,3 +
          -12.6876 * season=2,3 +
```

-32.8656 * season=3 +
 82.0024 * yr +
  2.9493 * mnth +
159.3296 * d-hr=Morning,Evening,Afternoon +
  5.1398 * d-hr=Evening,Afternoon +
 37.4049 * d-hr=Afternoon +
 20.4447 * holiday=0 +
 40.5138 * weathersit=2,1 +
  6.7178 * weathersit=1 +
 11.4481 * d-atemp=three,four,five,six,ten,seven,nine,eight +
 12.2288 * d-atemp=four,five,six,ten,seven,nine,eight +
 29.7447 * d-atemp=five,six,ten,seven,nine,eight +
 31.6974 * d-atemp=six,ten,seven,nine,eight +
-17.9143 * d-atemp=ten,seven,nine,eight +
 77.4475 * d-atemp=seven,nine,eight +
-25.3515 * d-atemp=nine,eight +
 37.805 * d-atemp=eight +
  6.8567 * d-hum=eight,seven,six,five,two,four,three +
 14.5319 * d-hum=seven,six,five,two,four,three +
  5.6726 * d-hum=six,five,two,four,three +
  9.7702 * d-hum=five,two,four,three +
 45.0211 * d-hum=two,four,three +
-31.8885 * d-hum=four,three +
 26.0222 * d-hum=three +
-52.8514 * d-windspeed=nine,seven,two,six,three,four,five +
 14.7921 * d-windspeed=seven,two,six,three,four,five +
 44.1719 * d-windspeed=two,six,three,four,five +
-20.62   * d-windspeed=six,three,four,five +
 20.6334 * d-windspeed=three,four,five +
-184.12