Abstract: This paper is an analysis of a data set from a bike sharing program to determine if a correlation exists between weather and day (e.g. weekday vs. weekend and holiday), as independent variables, and daily ridership.

Background

The data comes from two years' worth of records in the Capital Bikeshare program in Washington, D.C. This program is setup much like Zipcar in that a renter picks up a bike from a location in the city, uses it for a duration, and drops it at the same or another location.

This set is composed of a daily ridership total and several independent attributes describing the day and weather conditions for the first two years of the program.

The dependent variable ("ridersup") is derived by taking the median of values in the "cnt" attribute. Any day with ridership greater than the median is labeled "UP" while all others are labeled "NOTUP".

Initial Analysis

A rich description of each day is given by attributes "holiday", "weekday", and "workingday". Holiday and workingday contain simple true/false values represented by the normal 1/0 values. Weekday contains the range 0..6 with the corresponding days being Sunday..Saturday.

Similar to day, weather is composed of the attributes "weathersit", "temp", "atemp", "hum", and "windspeed". A description of values for the weathersit attribute are in Table 1. A description of the remaining attributes is in Table 2.

Label	Textual Description
1	Clear
2	Mist and/or clouds
3	Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
4	Heavy Rain + Ice Pallets + Thunderstorm + Mist,

	Snow + Fog

Table 1.

Attribute	Value Description[1]
temp	Normalized temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-8, t_max=+39
atemp	Normalized feeling temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-16, t_max=+50
hum	Normalized humidity. The values are divided to 100 (max)
windspeed	Normalized wind speed. The values are divided to 67 (max)

Table 2.

Basic analysis shows some interesting features. From Figure 1, it is clear that the general trend in ridership is up, but consistent dips appear. The consistency of the rise and fall suggests related conditions.

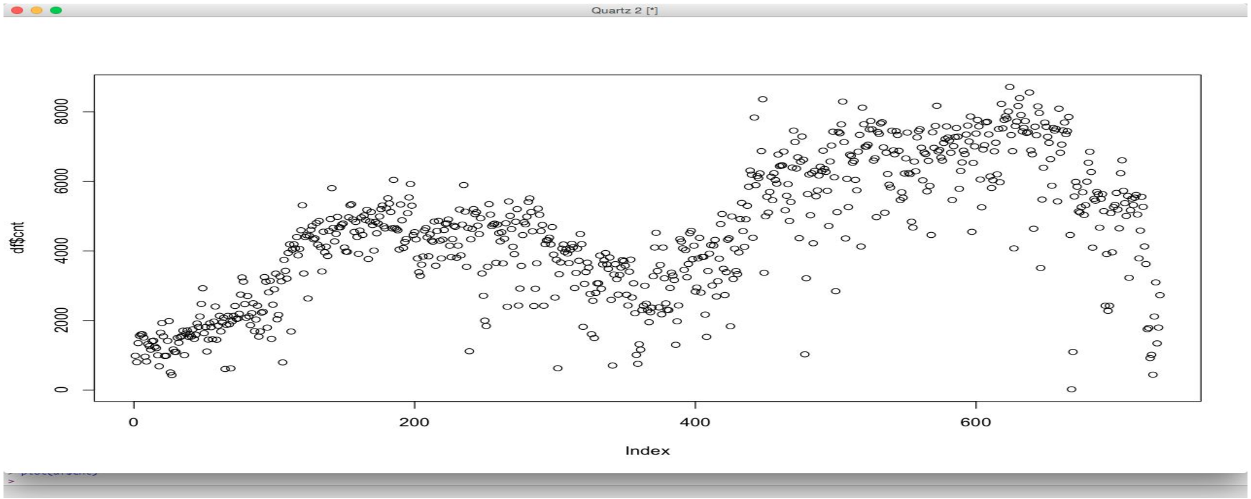

Figure 1. Daily Ridership Plot

The histogram of the dependent variable (Figure 2) shows slightly more days with increased ridership than not. This result is not so surprising since the variable was created by taking a median.

[1] http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset#

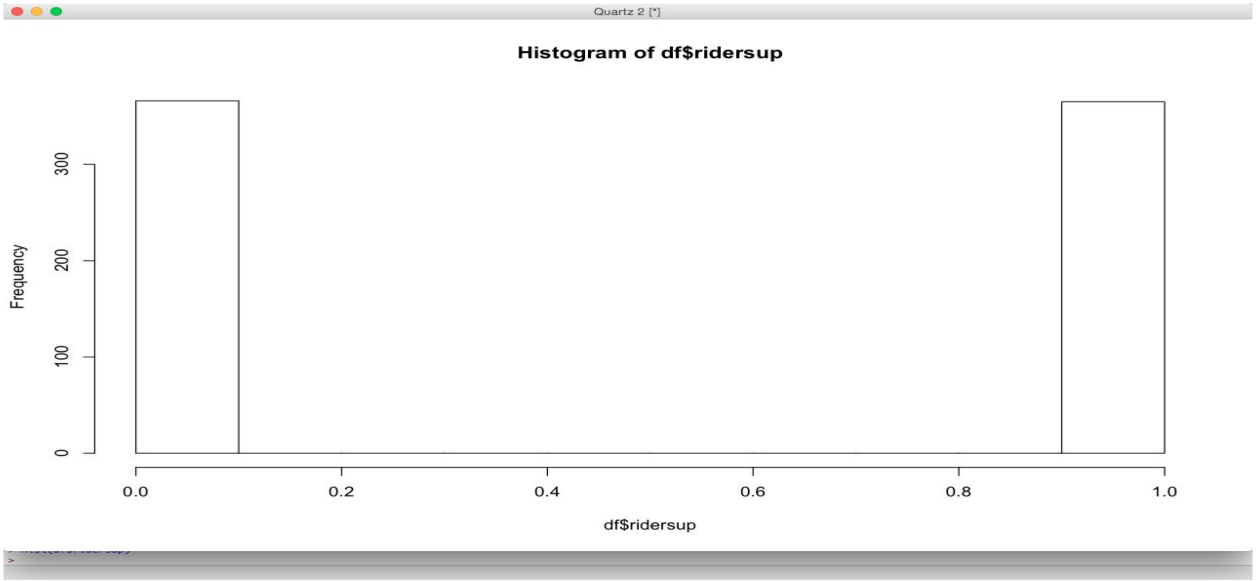

Figure 2. Riders Up/Not Up Histogram

Analysis Using Support Vector Machines (SVM)

The first step with this stage of analysis was to remove all attributes not related to weather or day. The removed attributes are in Table 3.

Attribute	Description
instant	Row number
dteday	Calendar date
season	1..4 => Winter..Fall
yr	1..2 => First and second year of the program
mnth	1..12 => January..December
casual	Unregistered users
registered	Registered users

Table 3.

While a danger exists in removing attributes, i.e. the missing attributes may hide correlations stronger than the correlations under examination, they do not hinder my analysis for the following reasons.

Attribute “instant” is meta-information extraneous to the actual ridership data: it is simply a row index. Attributes “dteday”, “season”, and “mnth” may show some correlation with the ridership fluctuations. However, they are important only in that weather patterns change with date ranges and seasons.

Attributes “casual” and “registered”, when added together, make the daily “cnt”. This breakdown of daily ridership is worth further analysis. However, that analysis is outside the scope of this report.

I used a 2-fold cross-validation - a fold for training and one for testing - due to the small number of records in the data universe. Data was separated into folds by a simple even/odd scheme on the row number. I did not want to simply split the data set down the middle due to the records going in order from onset through year 2 because the initial implementation in Year 1 may have had problems, not recorded in the data set, that affected ridership. For that reason, I decided to interleave data from each year.

I ran the linear, polynomial, and radial kernels for several values of their free parameters. The values for the free parameters are below in Table 4. Cost ranged from a large to a small margin. Degree ranged from low dimensional feature space to a higher one. Gamma ranged from a large to small radius.

Kernel	Cost	Degree	Gamma
linear	1..1000	n/a	n/a
polynomial	1..1000	1..3	n/a
radial	1..1000	n/a	0.1..0.9

Table 4.

Initially I ran each kernel with a set of values for the free parameters using the training set. Afterward, I ran the same kernels with the same set of values for free parameters using the testing set. From each run, I got a confusion matrix, stored the (1,2) and (2,1) values, added them to get an error total, and stored the free parameter values.

After getting back the results of each run, I compared the training and testing error totals. The testing run with the lowest error total became the winner for each kernel. Then I compared the kernel totals to decide on the top 2. The linear and radial kernels had the lowest errors with 0. The best runs for each kernel are in Table 5.

Kernel	TP	FN	FP	TN	Error	Cost	Gamma

linear	187	0	0	179	0	1000	n/a
radial	187	0	0	179	0	1000	0.1

Table 5.

It should be noted that the polynomial kernel had only 1 error. More than likely it is due to a strong correlation in the data.

As you can see from Table 5, both kernels display well-balanced confusion matrices: FN and FP are identical. Looking back to the plot in Figure 1, there is a marked slough off toward the right end that would get missed in a larger data set using the linear kernel with a narrow margin (1000). It seems like the narrow margin, but wide radius, radial kernel would perform better given a data set beyond a certain size.

The confidence intervals are in Table 6.

Kernel	Upper Bound	Lower Bound
linear	0	0
radial	0	0

Table 6.

As you can see, the error intervals are the same. Given this information, I would stay with the radial kernel for the reason I stated earlier in this document.