# Bike Sharing Company Analysis

## Introduction

The new generation of traditional bike rentals is Bike-sharing systems where the entire process has become automatic for membership, return, and rental. It is easy for a user to use these as, through these systems from a position and while returning, the bike can be placed at another location. At present, around the world, there are over 500 bike-sharing programs that are composed of over 500 thousand bicycles. Now, due to their important role in traffic, environmental, and health issues, there exist a great interest in these systems.

The features of data being generated by these systems make them attractive for the research, along with significant real-world applications of bike-sharing systems. In contrast to other transport services like the duration of travel, bus or subway, arrival and departure position, which is explicitly recorded in systems. This unique feature turns a bike-sharing system into a virtual sensor network that can be used for sensing mobility in the city. Therefore, it is supposed that most of significant events in the town could be detected via monitoring these data.

## About Data Set:

The process Bike-sharing rental is highly correlated to the environmental and seasonal settings. Rental behaviors can be affected by precipitation, weather conditions, season, day of the week, an hour of the day, etc. The core data set is concerned with the historical log of two-years 2011 and 2012, from Capital Bikeshare system, Washington D.C., USA, that is publicly available in http://capitalbikeshare.com/system-data.

**Data Collection**:

The dataset we are utilizing is from the UCI AI storehouse. The informational index compilers used data somewhat from a chronicled log of two years, i.e., 2011 and 2012 from Capital Bikeshare System, Washington D.C. They have totaled the information on two, day by day, and hourly premise and afterward extricated alongside the expansion of relating climate and occasional data.

Our dataset is a CSV document with data from 17,379 hours more than 731 days with 16 attributes for every hour. The features are:

1. Record index
2. Date
3. Season (1:spring, 2:summer, 3:fall, 4:winter)
4. Year (0: 2011, 1:2012)
5. Month (1 to 12)
6. Hour (0 to 23)
7. Holiday: whether that day is holiday or not
8. Weekday: day of the week
9. Working-day: if a day is neither weekend nor holiday, value is 1. Otherwise 0
10. Weather situation:
    - Clear, Few clouds, Partly cloudy, Partly cloudy.
    - Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist

- Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
- Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog

11. The normalized temperature in Celsius. Values are divided into 41 (max)

12. The normalized feeling temperature in Celsius. Values are divided into 50 (max)

13. Normalized humidity. The values are divided into 100 (max)

14. Normalized wind speed. The values are divided into 67 (max)

15. Count of casual users

16. Count of registered users

17. Count of total rental bikes including both casual and registered

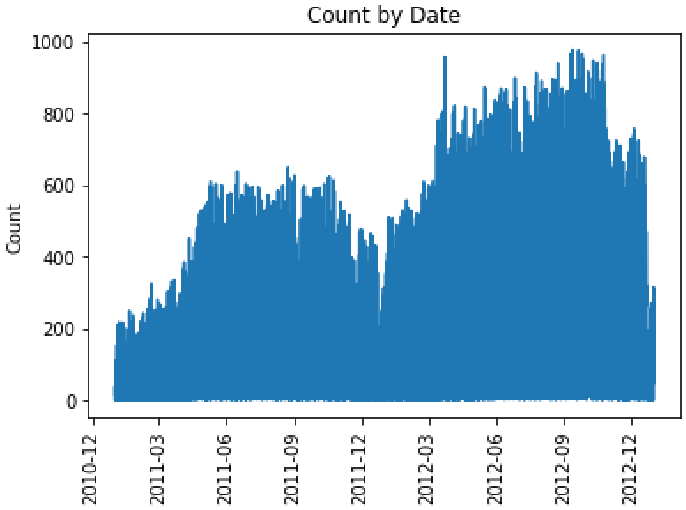
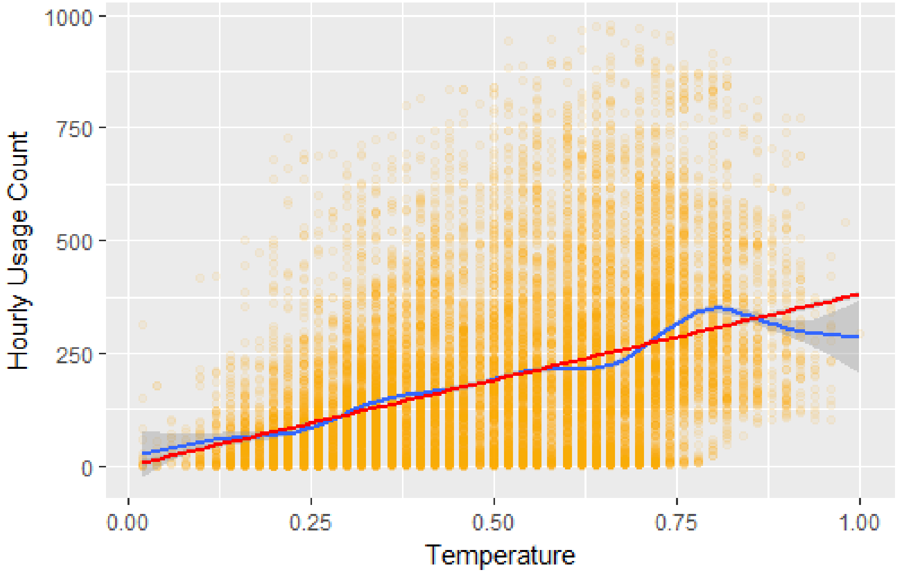


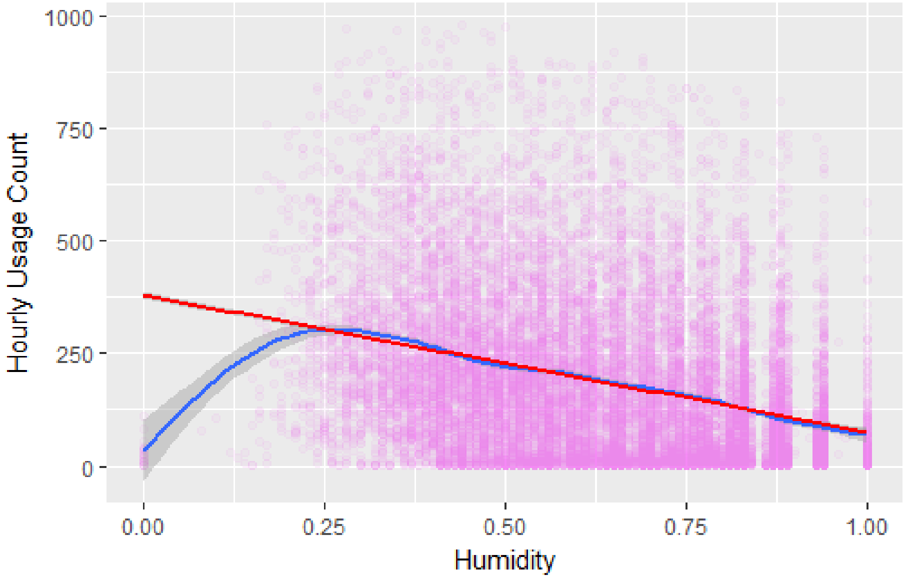
Fig: Usage Count Vs. Usage Count by date

## Exploratory Data Analysis

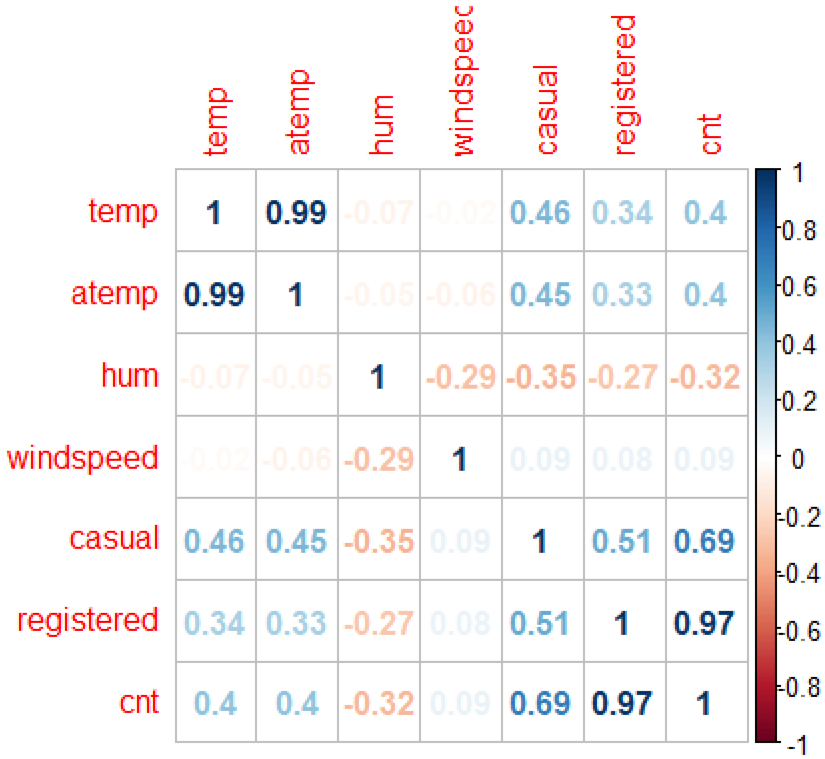
1. A positive correlation between the temperature – to – usage and adjusted temperature - to – usage, which means the users rent the bike mostly in moderate conditions rather than the extreme temperatures.



2. A negative correlation between the humidity and the usage rate. This means high humidity is inversely proportional to the user riding a bicycle.



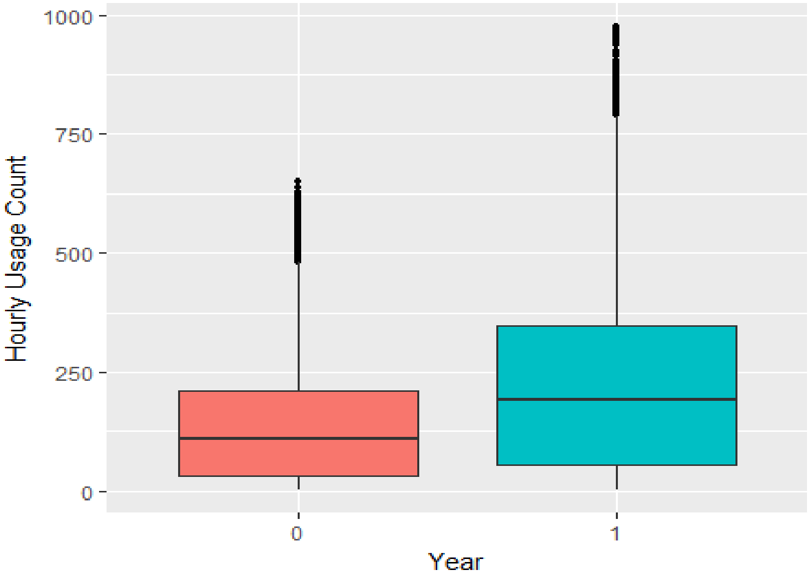
3. There are brighter days than overcast and rainy days, which is known by the weather situation data.
4. The wind speed doesn't show a significant effect on the usage as the correlation between them is least.



|  | temp | atemp | hum | windspeec | casual | registered | cnt |
|---|---|---|---|---|---|---|---|
| temp | 1 | 0.99 | -0.07 | -0.02 | 0.46 | 0.34 | 0.4 |
| atemp | 0.99 | 1 | -0.05 | -0.06 | 0.45 | 0.33 | 0.4 |
| hum | -0.07 | -0.05 | 1 | -0.29 | -0.35 | -0.27 | -0.32 |
| windspeed | -0.02 | -0.06 | -0.29 | 1 | 0.09 | 0.08 | 0.09 |
| casual | 0.46 | 0.45 | -0.35 | 0.09 | 1 | 0.51 | 0.69 |
| registered | 0.34 | 0.33 | -0.27 | 0.08 | 0.51 | 1 | 0.97 |
| cnt | 0.4 | 0.4 | -0.32 | 0.09 | 0.69 | 0.97 | 1 |

Fig: Correlation Matrix

5. Looking at the year variable, the usage count has been increased from 2011 to 2012.



**Literature Review:**

Bikeshare has been rapidly growing. The review encompasses recent research from Asia, North America, Europe, and Australia. The emphasis is on published papers of relevance, included in the synthesis of bike-share literature published by the journal. Relevance papers are collected using Google scholar databases, using the search terms 'Bicycle sharing', 'Bike Sharing', 'Public Bike', etc. This paper reviews research across the range of bike-share topics, including documented history and growth, user preferences, usage patterns, demographics etc. The current limitations in bikeshare knowledge are highlighted in the review, and it is particularly evident in areas of rebalancing and impacts on bike-share. The usage can vary dramatically between BSP's in different cities, but they generally exhibit similar daily usage profiles. The usage peaks between 7 am – 9 am and 4 pm – 6 pm, while weekend usage is strongest during mid-day. The demographics have become a common focus of attention for bike-share operators and researchers. The issues examined here are gender, education status, ethnicity, income, and underlying population averages. Much of this analysis revealed that users tend to be of higher average income and education status.

**Models Used to Predict the Data:**

1. Ridge Regression
2. Random Forest Regression
3. Support Vector Machine

**Data Preprocessing**:

The variables such as instant (indicating the record index as row numbers), today(date), casual and registered(indicating the number of casual and registered users) which sum up to give the target variable count are excluded, as they are not appropriate to include in modeling.

For each algorithm, data is divided as 60% for training, 20% for validation, and 20% for testing. The models were trained using a train set, tuned using a validation set, and results were generated using the test set.

**Ridge Regression Model:**

Ridge regression is a data analysis technique that is used for analyzing multiple regression data that suffers from multicollinearity. It is used as a baseline model for performance comparison with other complex models. We have tried a simple linear regression model without a regularization term that resulted in overfitting on train data. In order to address the problem of high variance, L2 regularization then included in the cost function, which resulted in a better generalization of the validation data set. The regularization parameter α is used to control the amount of regularization, which helps in maintaining a good bias-variance tradeoff in the model.

The below plot determines that the cost function reaches global optimum, and parameter update has fallen in the first 4000 iterations as it is a relatively simple model.

Cost as no. of iterations function: Test RMSE: 187.274 custom model is finally compared with a model from the sklearn library in Test RMSE: 187.278 python and observed to have similar performance but resulted in a better way than latter.

**Random Forest Model:**

Random forest is a model that is a combination of many decision trees. Here we use random sampling of training data points when building trees. It is designed using sklearn library in python. Initially, we made N bootstraps from both training and testing data, and the bootstrap is randomly chosen as a subset of the whole training dataset. Each decision tree returns the best result from the obtained. Here, any two decision trees should be independent. Hence, the algorithm with generalization ability is provided and avoids overfitting. For testing, each testing bootstrap is put in a forest and resulted in output generated by each decision tree. Then, we use the average of all the outputs obtained as a result.

In this algorithm, there are 3 hyperparameters that affect regression accuracy:

N_estimators: The algorithm has a high risk of overfitting when the no. of trees is too small. With increment of hyper parameter, the algorithm becomes more stable, but time consuming.

The Max_Features: We randomly select a subset of feature and choose one from them, that can best divide data into two parts when we split a node in decision tree. The hyperparameter restricts the no of randomly chosen feature when splitting each node. The hyperparameter should be less than no. of whole features and far less, to make the algorithm more accurate.

The Max_Depth: If the maximum length is too small, there is a high chance of increase in error. The judge function is used to measure quality of each split and different functions will result in different accuracies on algorithm, with in small range.

When each bootstrap is formed, the rest of the data is Out of Bag data(OOB), from which the error is calculated for certain feature x of the data, naming OOB1. Then, some random noise is added into same OOB data and do the same error calculation on same feature, naming it OOB2.

Importance = Σ(errOOB2-errOOB1)/N.

If the error changes dramatically, after adding noise, it means this factor plays a prominent role in regression and the more it varies, the more important it would be.

## Support Vector Regression:

Support vector machine is a regression method, with all main features that characterize algorithm. Here, support vector with linear kernel is implemented using SMO algorithm. It aims to find optimal values of lagranagian multipliers ensuring KKT conditions not violated. The main 4 hyperparameters C, Epsilon, tolerance and Number of iterations are mainly focused to optimize the model performance.

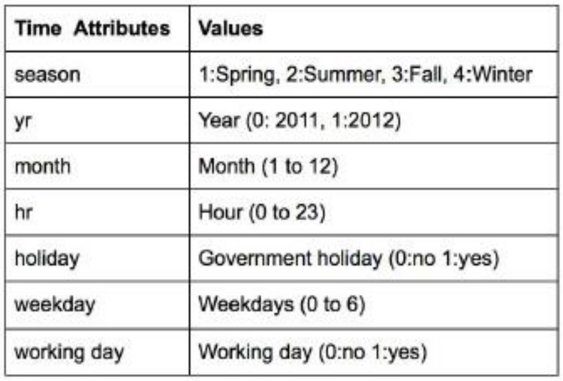
The 'C' hyperparameter in the model, shows the tradeoff between model complexity and degree which deviates larger than epsilon, tolerated in optimization formulization. When 'C' value is set to 500 ( a large value), we noticed a high error rate, which is because of the objective function that aims to minimize empirical risk, without regard to model complexity. When 'C' value is set to 1 (a small value) it has improved the performance of model, by allowing model to make some limited mistakes.

Epsilon is used to control the width of insensitive zone, to fit the training data. It affected the no. of support vectors to construct regression function. Higher the epsilon value, lower the support vectors to construct regression function. On the other hand, higher values resulted in 'flat' estimates.

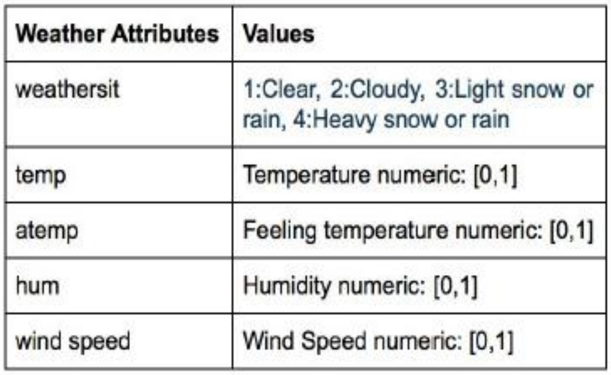
By decreasing the value of 'tolerance', the RMSE on validation set decreased until certain threshold. We increased value of no. of iterations starting with 100, till algorithm converged and reached global optimum.

## Data Visualization and Results:

The data taken from UCI Machine Learning Repository has 17389 instances with 17 features containing hourly count for bikes rented between years 2011- 2012 in Washington DC. The variables instant, casual, registered sum up to give target variable count, are excluded as they are not appropriate in modelling. The other numeric variables are normalized with min-max normalization as (t-t_min)/(t_max-t_min). ‘cnt’, the target variable is integer type with max=997, min=1, mean=189.4, having high variance.



| Time Attributes | Values |
|---|---|
| season | 1:Spring, 2:Summer, 3:Fall, 4:Winter |
| yr | Year (0: 2011, 1:2012) |
| month | Month (1 to 12) |
| hr | Hour (0 to 23) |
| holiday | Government holiday (0:no 1:yes) |
| weekday | Weekdays (0 to 6) |
| working day | Working day (0:no 1:yes) |



| Weather Attributes | Values |
|---|---|
| weathersit | 1:Clear, 2:Cloudy, 3:Light snow or rain, 4:Heavy snow or rain |
| temp | Temperature numeric: [0,1] |
| atemp | Feeling temperature numeric: [0,1] |
| hum | Humidity numeric: [0,1] |
| wind speed | Wind Speed numeric: [0,1] |

**Splitting data and tuning Hyperparameter:**

The distribution of data is done as 60% for training, 20% for validation and 20% for testing. Train set is used for training the data and tuned with validation set. Results are generated with test set. The loops are used to tune SVM model by tuning hyperparameter one by one at a time keeping other hyperparameters fixed initially till we get some decent values and later turned by focusing on associations between different ones, to find best combination of hyperparameters.

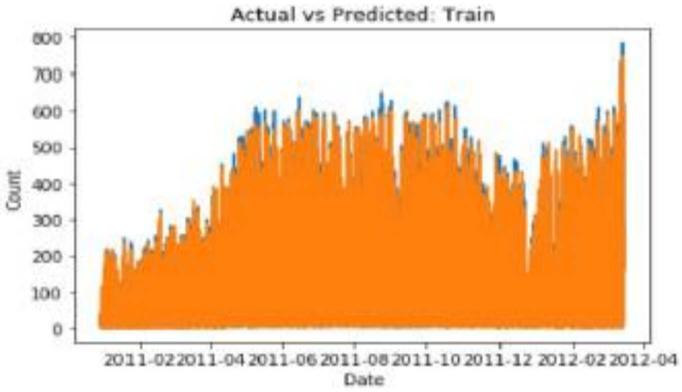
**Comparison of Performances of Model:**

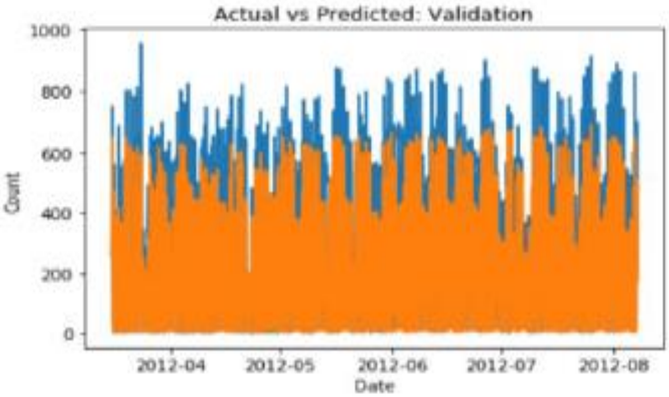
| **Model** | **RMSE in Training** | **RMSE in Validation** | **RMSE in Testing** |
|---|---|---|---|
| Ridge Regression | 107.4178 | 184.7357 | 187.2781 |
| Random Forest | 12.722 | 100.0381 | 113.5600 |
| SVM Regression | 109.7181 | 198.2046 | 197.6296 |

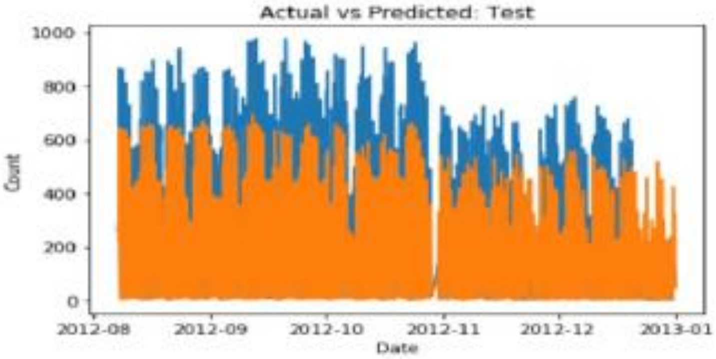
**Conclusion:**

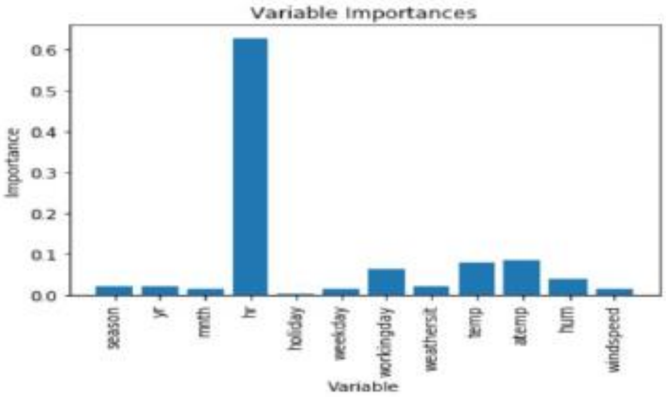
The Random Forest Algorithm has smallest RMSE error for training, testing and validation. Therefore, this is the best model a company can use for future predictions.

First three plots determine predicted values(orange) and true values(blue) of target variable count of the target variable count on y-axis which is plotted against 'date' variable on x-axis for random forest, showing the importance of hour followed by 'atemp'.




Actual vs Predicted: Validation


Actual vs Predicted: Test


Variable Importances

## Bibliography:

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

2. http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset

3. https://www.researchgate.net/publication/275517751_Bikeshare_A_Review_of_Recent_Literature
4. Segal,Mark R,'Machine Learning Benchmarks and Random Forest Regression':UC San Francisco,2004-04-14
5. Improvements to SMO Algorithm for SVM Regression