FORECASTING DEMAND OF BIKE SHARING SYSTEMS

Aniruddha Rajshekar ARAJS@SEAS.UPENN.EDU
Yongbo Qian YONGBO@SEAS.UPENN.EDU
Funmilade Lesi LESI@SEAS.UPENN.EDU

Abstract

This project seeks to use bike data to help planners assess and improve bike sharing as a viable transportation investment. In doing so, this report helps to inform a larger policy dialogue about bicycle sharing locally and nationally. The data includes information of bike sharing systems and weather pattern. The research used Extra Trees and Random Forest methods to predict the demand and help economics and business planners on different needs in various locations.

1. Introduction

Forecasting demand is a crucial issue for driving efficient operational management plans. The increasing availability of new demand signals and large amounts of data are helping define robust and efficient techniques that can infer the stochastic dependency between past and future based on data. Over the years, there has been an increase in different mode of transport sharing systems around the world. One strong example is the increase in bike sharing systems in different cities. There are more than fifty cities in United States have already initiated Public Bike System, and Philadelphia will have a bicycle program run by Bicycle Transit Systems in April 2015. However, many of the bike-sharing programs are suffering financially because there is a lack of understanding of the demand of bike. A lot of projects over-estimate demand for bike sharing services or cannot show investors correct estimate of demand in order to get access to funds. In order to solve this problem and promote investment in bike sharing systems, this project aims to provide the appropriate forecast on bike sharing demand using tree-based ensemble learning methods.

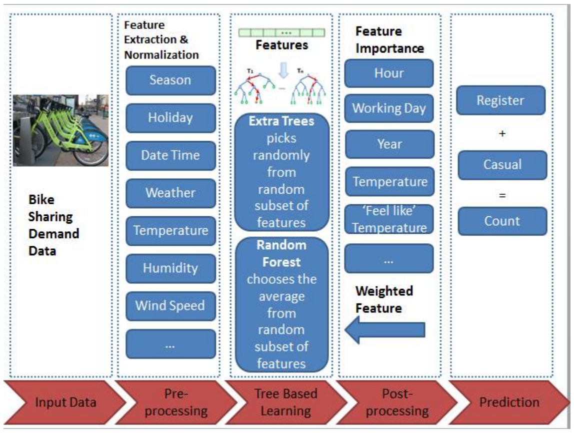


Figure 1. Having extracted features from the raw data and normalization, our approach implemented *tree-based ensemble learning* model to predict bike sharing demand

2. Related Work

There has been an explosion of machine learning methods for forecasting transport demand in the last few years and a few attempts to use machine learning in demand forecasting problems in transport have been reported in the literature. Guedelha and Seixas studied the demand for parking space in shopping centers. Using data from shopping centers spread around Brazil, the developed neural estimator proved to be superior to the estimates obtained from econometric techniques traditionally applied to this problem [1]. Shmueli tested neural processing as a tool for transport planning [2]. One of the case studies involved the behavioral forecasting of urban car trips, comparing trip patterns for men and women in Israel. The aim was to determine the connection between such trip patterns and the key socio-economic and demographic variables. The problem involves complex data, highly dimensional and in large scale, for which the neural and random forest methodology has been applied successfully in physical sciences and cognitive modeling, but it is relatively new to social sciences. The author concluded that the neural or random forest modeling was effective, particularly when expressed in terms of spatial relationships.

University of Pennsylvania, CIS 419/519 Course Project.

3. Project Goals and Metrics

It is the interest of transport planners to understand the demand for a particular transport system to better plan for future capacity. Most transport planners use a four step trip generation model for trip generation, trip distribution, mode choice and route assignment. However, this paper will use data collected from different bike sharing systems in over 30 countries to model different demands based on different features in the data. The data contains the following features; date, season, holiday, weather, temperature, humidity, windspeed and windspeed. To forecast demand, the following must be done: The application of linear and non-linear methods to analyze the data, a creation of a model to predict the demand and a metric based comparison of the predicted result which includes root mean square error, accuracy and waggle competition leaderboard score. [3].

4. Pre-Processing

Pre-analysis was done to understand the importance of each feature. Humidity was plotted against total count in four different weather conditions. The four different weather categories were; Clear, heavy rain/snow, light snow/rain and misty/cloudy. As seen in the plot below, there were very large demand during clear weather and at humidity between 31 to 60.

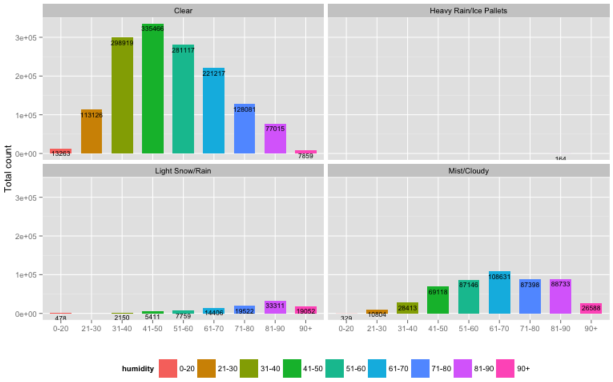


Figure 2. Plot of Humidity against different weather condition

Humidity was plotted against total count in four different season. The four different seasons were; summer, spring, winter, fall. As seen in the plot below, there were very large demand for humidity range of 31 to 60. However, the different seasons do not show any difference in demand.

After the plots above, A feature plot of humidity against registered and non-registered users was done. There is a total count of demand that is a combination of registered and non-registered users in our data set, by breaking up the total demand and predicting based on different demand segments, a more accurate predictive model can be created. With this pre-conceived analysis, we made plots of different features against registered and non-registered users.

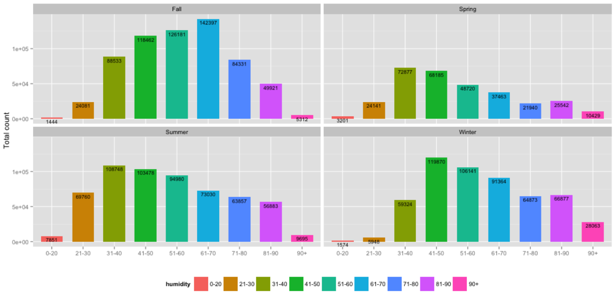


Figure 3. Plot of Humidity against different seasonal condition

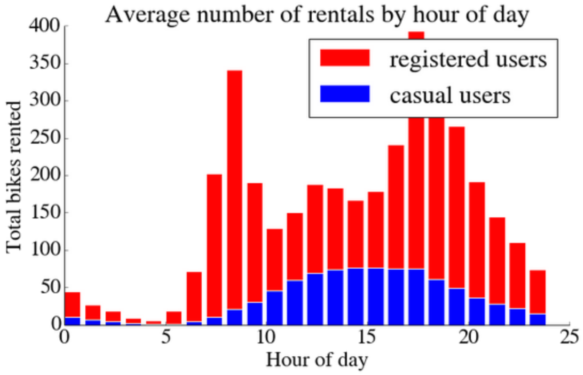


Figure 4. Plot of Hours in the day against total counts of registered and non-registered users

Other analysis were done in this same form above, plots of season and weather against windspeed were graphed, however; this plots weren't as visually informative as the two shown above. Information from the pre-analysis was used in the machine learning algorithm to create a more general model.

5. Analysis

Tree based algorithms were used because the data contains parameters like weather, temp, atemp, humidity, windspeed which can be split using a Decision Tree. These were: DecisionTrees, Random Forest, ExtraTrees Regressor, Gradient Boosting Regressors, and K Nearest Neighbors.
Out of all the tree based algorithm, extra tree regressor and random forest were found to have the best predictive power and result for the data selected. Using ensemble algorithms usually gives the best results for such types of problems.
The two algorithms that gave the least RMSLE were Ran-

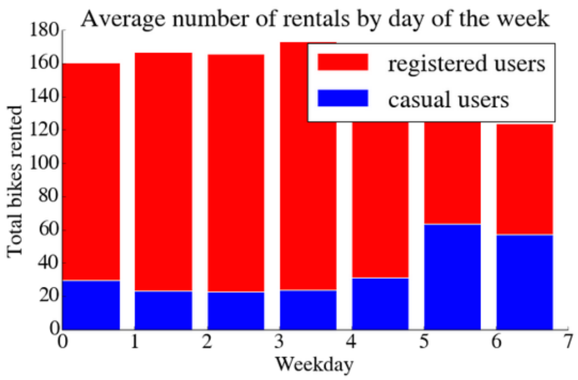


Figure 5. Plot of Weekday against total counts of registered and non-registered users

dom Forests and ExtraTrees Regressors. We used Python 2.7, numpy 1.8.1, scipy, scikit_learn 0.14.1, pandas 0.14.0. as tools to analyze the data.

5.1. ExtraTrees Regressor

ExtraTrees (or Extremely Randomized Trees) is an ensemble method consist of averaging predictions of an ensemble of trees built in a randomized fashion. Each tree is grown by selecting at each node a number K of random splits (random choice of variable x_i', and random choice of threshold t) and keeping among these the one which maximizes the score. These trees are grown until all sub-samples at all leaves are either pure in terms of outputs or contains less than n_{min} learning samples. If the parameters are set respectively to $K = 1$ and $n_{min} = 2$, these trees are totally random (in the sense that their structure does not depend on the outputs in the learning sample) and they perfectly fit the learning sample. Thus, in this case the Extra-Trees method can be viewed as an approach to sample in a neutral way from the set of all possible trees perfectly fitting the learning sample [4].

The main properties of ExtraTrees that are useful for problems of this nature are:

Universal approximation/consistency: It carries over directly from single-tree based methods.

Robustness to outliers: While the linear regression is strongly and globally affected by the outliers, the Extra-Trees model is only marginally affected, and only very locally in the regions where the outliers are located [5].

Computational complexity : The complexity is essentially proportional to $M * K * N * \log N$, which may be better than that of single trees if the number n of input variables is very large compared to K. Note that accuracy always increases monotonically with M, but in a problem dependent way; typical values of M are in the range $[10; 100]$.

Robustness w.r.t. irrelevant/redundant inputs: The tree growing remains robust to irrelevant variables as long as K is sufficiently large with respect to n.

Very low variance: Compared to standard tree-based regression the variance of Extra-Trees is negligible

Extensions/generalizations: Those extensions that were discussed for standard trees all carry over to Extra-Trees, with minor adaptations [5]

5.2. Random Forest Regressor

Random Forest Another method implemented in this application is Random Forests. Similar to Extra trees method, it also relies on generating random vectors to control the growth of each tree in the ensemble. Intuitively, it represents a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of finding a nature balance between variance and bias.

Random forests for regression are formed by growing trees depending on a random vector θ. A simple procedure is described as below: The training set is independently drawn from the distribution of the random for the k^{th} tree, a random vector θ_k is generated, independent of the past random vectors θ_1, ... ,$\theta_k - 1$ but with the same distribution; and a tree is grown using the training set and θ_k , resulting in a predictor h(x,θ_k) where x is an input vector. After a large number of trees are generated, The final predictor is formed by taking the average over k of the trees h(x,θ_k) [7].

Since there are many input variables in this bike sharing demand prediction problem, with each only containing a small amount of information, combining trees grown using random features can produce better accuracy. Random Forest regressor can also run efficiently on this fairly large data set with relatively robust to outliers and noise. In addition, as the number of trees increases, the testing performance does not decrease due to overfitting. That is why this method is also favorable for this application [6].

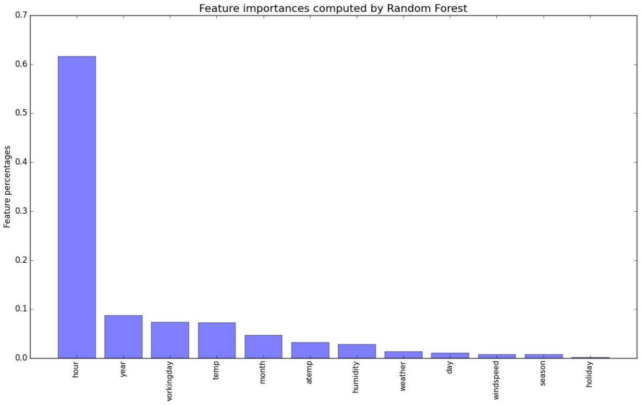
6. Feature Importance and Initial Result

By computing internal estimates of variable importance, the mechanism of these tree-based ensemble learning algorithms as well as the characteristics of the original dataset could be better understood. Table 1 ranks each feature from

Extra Trees Regressor	Random Forest Regressor
Hour, 0.5737	Hour, 0.6102
Working Day, 0.0863	Year, 0.0867
Year, 0.0815	Working Day, 0.0727
”Feel like” Temp, 0.0634	Temperature, 0.0714
Temperature, 0.0587	Month, 0.0484
Humidity, 0.0360	”Feel like” Temp, 0.0343
Season, 0.0277	Humidity, 0.0301
Month, 0.0274	Weather, 0.0150
Weather, 0.0201	Day, 0.0121
Day, 0.0111	Wind Speed, 0.0088
Wind Speed, 0.0102	Season, 0.0075
Holiday, 0.0037	Holiday, 0.0030

Table 1: Feature Importance

Figure 6. Random Forest Feature Importance



most to least important with their importance fractions for each classifier. For both classifiers, hour, working day and year turn out to be more important, while holiday is the least important. Two corresponding plots were also generated for visualization purpose.

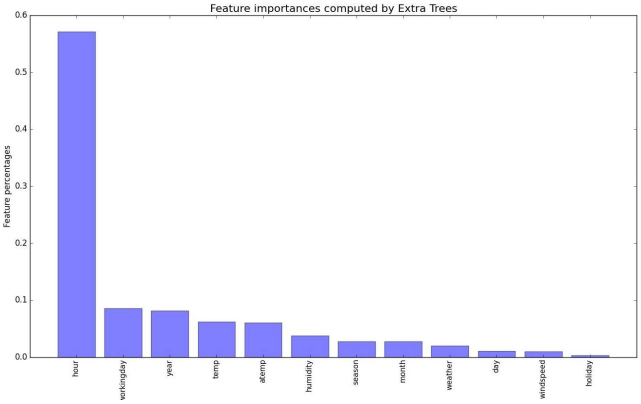
We submitted our initial result for these two classifiers to Kaggle competition. The root mean square log error for Extra Trees regressor is 0.47124, which ranks 334 out of 1276 (at the submission time); and Random Forests Regressor has an error of 0.48177 which ranks 400.

7. Post-Processing and Final Result

7.1. Initial Unsuccessful Attempts

After going through papers which used neural processing to predict demand, instead of looking and analyzing our data accurately before implementing a predictive algorithm, assumptions were made that neural net might be a good predictive algorithm for our data. However; after running our data through the neural net algorithm, we got a predictive model with low accuracy and low score on the Kaggle leader board, ranking bottom 10%.

Figure 7. Extra Trees Feature Importance



Weighing features was also an approach investigated in the paper. After seeing pre-processing and the feature importance results that showed that some features were more important than others, weights were assigned to features, with higher weights assigned to features of more important and lower weights to features of lower importance. However, after standardization, there was no effect of the weighted approach to the accuracy of the predictive algorithm.

7.2. Standardization

After getting a reasonable result, a further analysis and preparation of the data was required to get a better predictive model. The first approach was to normalize the data. Database normalization can essentially be defined as the practice of optimizing table structures. Optimization is accomplished as a result of a thorough investigation of the various pieces of data that will be stored within the database, in particular concentrating upon how this data is interrelated. An analysis of this data and its corresponding relationships is advantageous because it can result both in a substantial improvement in decreasing the chance that the database integrity could be compromised due to tedious maintenance procedures [7]. We improved our accuracy by 0.5 and our position on the Kaggle leader board by 100.

7.3. GridSearch and Cross validation

Since injecting the right kind of randomness is important to make both methods accurate regressors, a grid search algorithm has been ran in order to optimize the model parameters.The GridSearchCV function provided by Sckitlearn was used to exhaustively generate candidates from a grid of parameter values. 5 fold cross validation was used here. Parameters obtained were:

Metrics	Extra Trees	Random Forest
RMS Log Error Train	0.148	0.201
RMS Log Error Test	0.339	0.352
Training Accuracy	99.16%	98.51%
Test Accuracy	94.92%	94.38%
Kaggle result	0.46711	0.48079

Table 2: Initial prediction result

Metrics	Casual User	Registered User	Combined Model
RMSLE Train	0.228	0.148	0.112
RMSLE Test	0.540	0.341	0.272
Training Accuracy	98.88%	99.18%	99.61%
Test Accuracy	90.99%	94.80%	96.25%

Table 3: Final prediction result

(*n_jobs=1, n_estimators=1200, min_samples_leaf=2, min_samples_split=2, max_features = 'auto', oob_score=True*)
Using these optimized parameters and previous step of standardization, the result improved a lot.

7.4. Comparison

The promising result indicates both methods are an effective tool in predicting this particular dataset. However, Extra Trees method yields to a higher prediction accuracy. This makes sense theoretically because Extra Trees implement more randomized trees by randomizing an additional the top-down splitting in the tree learner. With bigger trees been grown, Extra Trees generalizes better and slightly improves the prediction result.

7.5. Predicting based on registered and non-registered users

The predicting model was built on the total number of demand, however; the total demand had two split, registered and non-registered users. Registered or not is another factor distinguishes people's bike renting pattern. Therefore, a better predictive model will be the one that predict separately based on the number of registered users and non-registered users, and add them together. This approach increased our accuracy by 0.1 and our position on the Kaggle leader board by 150. We finished within the top 20% of the competition. Using extra trees as the final predictive model, table 3 shows the new prediction result. It is clear that combining the models of casual and registered users, the result is much better than using the single prediction model.

8. Conclusion and Future Work

In this paper two different tree-based ensemble learning approaches were investigated on the dataset of Bike Sharing Demand Prediction. After the pre-processing procedure, the Extra Trees method shows better performance than the Random Forest method. Using Extra Trees Repressor as our prediction model, the results were also improved by implementing some post-processing techniques. A timeline of the scores of Kaggle competition submission was shown below (lower score represents better result). For the future work, the result could be further improved by combining random features with boosting. Same approaches should also be tested in some other transportation demand prediction systems to become more generalized.

First Stage	Second Stage (with Standardization)	Third Stage(with Casual/Registered Split)
0.47124	0.46711	0.45459

Table 4: Kaggle RMSLE submission timeline for Extra-Trees

9. Citations and References

[1] K. C. P. Guedelha and J. M. Seixas, A Neural Estimation of the Parking Space Needed in Shopping Centers, The 3rd World Multiconference on Systemics, Cybernetics and Informatics SCI99 and The 5th International Conference on Information Systems, Analysis and Synthesis ISAS?99, Orlando, FL, 1999.
[2] D. Shmueli, Applications of neural networks in transport planning, Progress in Planning, 50 (3), 1998, 141-204.
[3] Assessing the predictive capability of randomized tree-based ensembles in streamflow modeling S. Galelli and A. Castelletti Singapore-Delft Water Alliance, National University of Singapore 2 Engineering Drive 2, 117577, Singapore.
[4] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano Piazza L. da Vinci, 32, 20133 Milano, Italy.
[5] Centre for Water Research, University of Western Australia, Crawley, Western Australia, Australia
[6] TRAVEL DEMAND FORECASTING FOR URBAN TRANSPORTATION PLANNING by Arun Chatterjee and Mohan M. Venigalla
[7] RANDOM FORESTS, Leo Breiman, Statistics Department, University of California, Berkeley, CA 94720 (http://www.stat.berkeley.edu/ breiman/randomforest2001.pdf)