Demand Forecasting for City Bike Sharing Systems

Lars Kutschinski

December 2023

1 Introduction

In most larger cities around the world cycling can be a quicker form of transportation than walking and may even be quicker than driving or taking a taxi. Furthermore, bicycles offer a lower-pollution alternative to driving, which can be appealing to cities struggling to contain emissions. Hence the introduction of bike sharing programs, where users are able to rent a bike at one location and drop it off at another, made a significant impact in larger metropoles. These bike sharing programs have been rising in popularity across the globe in the last twenty years and thus they have come to play an important part in the analysis of traffic and environment issues. Discovering the factors that influence bike share demand is essential in understanding the effect that they have on urban travel patterns. One one hand such factors can be related to different weather conditions like temperature, humidity, wind speed and so on. On the other hand, temporal features such as season, week-or working days and month seem to have a significant impact on the daily demand. While the popularity of bike sharing systems has been increasing over the recent years, there are many small scale fluctuations in demand across shorter time periods. Hence time itself is a crucial factor in the analysis of bike demand.

Another problem is that the hourly rental demand can be very unbalanced at different times and places. Bikes must be available in the places that people need them and at the time they are needed in order for the program to be effective. Hence bike sharing administrators need to forecast this demand so that they can efficiently allocate their bikes to their stations.

In this project we are mainly focused on forecasting the hourly bike rental demand. We only focus on the time aspect and neglect the role that location plays in demand. While the inherent time dependence of the hourly demand might call for the usage of time series methods, we explore different models from machine learning for this task. We evaluate and compare these different methods in their prediction power and then analyze the significance of the different factors of time and weather.

This paper begins with an explanatory data analysis of the dataset that we considered. We give a description of the data and illustrate some key findings of different vizualisations of the data. In section 3 we describe the methods that were used for the statistical analysis, namely linear regression, random forests and gradient

boosting. Finally, we present the results in section 4 and compare the performance of our models.

2 Explanatory Data Analysis

The data (*Daily Bike Rental Dataset* 12/19/2013) that we are using for this project was collected in a time range from 2011 to 2012 and contains various seasonal and weather information related to bike share systems. It is a dataset from the UC Irvine Machine Learning repository. The data consists of 17389 instances and 15 features. The variables can be grouped into three different categories: Temporal information, meteorological data and information related to the user count.

Table 1: **Temporal Information**

Variable	**Name**	**Type**	**Definition**
Record index	instant	numerical	trivial variable indexing each entry
Date	dteday	date	year-month-day
Season	season	categorial	1:Winter, 2:Spring, 3:Summer, 4:Fall
Year	year	categorial	0 for 2011 and 1 for 2012
Month	mnth	categorial	months from 1 to 12
Holiday	holiday	categorial	1 if day is holiday and 0 else
Weekday	weekday	categorial	days of the week from 0 to 6
Workday	workingday	categorial	1 if neither weekend nor holiday and 0 else

Table 2: **Meteorological Information**

Variable	**Name**	**Type**	**Definition**
Weather	weathersit	categorial	• 1: Clear, Few clouds, Partly cloudy, 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 Pallets + Thunderstorm + Mist, Snow + Fog
Temperature	temp	numerical	Normalized temperature
Perceived Temperature	atemp	numerical	Normalized felt temperature
Humidity	hum	numerical	Normalized humidity
Wind speed	windspeed	numerical	Normalized windspeed

Table 3: **Demand Information**

Variable	**Name**	**Type**	**Definition**
Casual user count	casual	numerical	Count of hourly casual users
Registered user count	registered	numerical	Count of hourly registered users
Total user count	cnt	numerical	Count of total hourly users

The variation in popularity of bike share systems can be seen in Figure 1. Even though the data contains only information within a two year time frame, we can notice that the average daily demand has significantly increased in the second year of the collected data. Furthermore, there is a noticeable decrease in bike rentals in the winter months, where lower temperatures and bad weather conditions affect the daily demand. Overall, we notice the heavy depends on time, which implies that the rental count behaves like a time series. Hence, ordinary statistical methods will not perform well in the problem of prediction.

A common issue with bike share count data is that there can be missing values on days where the systems are defect. Fortunately, in this data set no missing values have been found.

We notice that there could be high multicollinearities between the features of our data. For example, weather conditions are usually very dependent on variables of time such as season and month. In addition to that, the variables *temp* and *atemp*

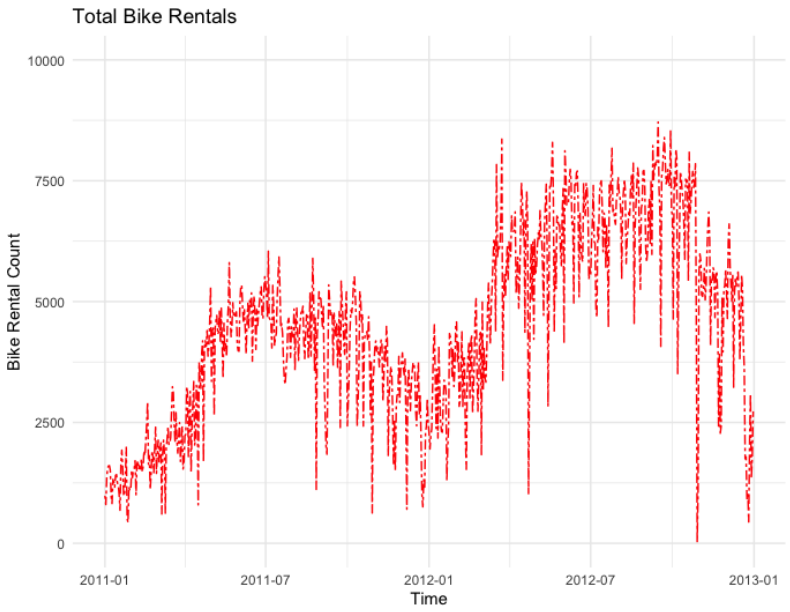


Figure 1: **Total Rentals between the year 2011 and 2012**

are expected to be highly correlated. Unfortunately, most of the features of the data are categorial and thus computing the correlations between such features does not serve any purpose. In Figure 2 we drew a correlation heatmap between the few numerical variables in the dataset. We confirmed that there is a high correlation of 0.99 between *temp* and *atemp*. Thus, dropping one of the features from the prediction models will not significantly affect the prediction accuracy.

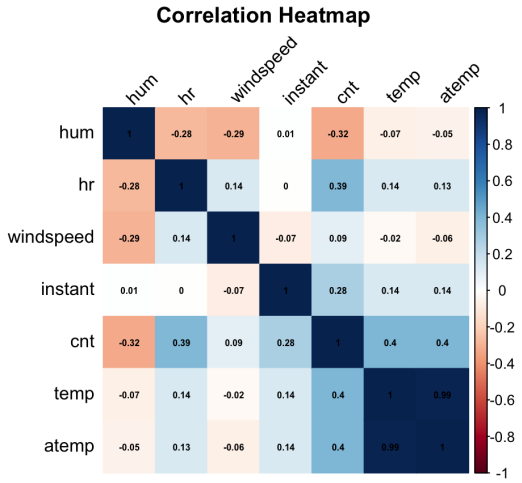


Figure 2: **Correlation Heatmap for numerical features in the dataset**

3 Methods

We have used three different modeling methods for the forecasting problem. The first one is classic linear regression. The reason for this model choice was to test whether

or not such an ordinary model could still perform reasonably well in prediction and, if not, if it could still provide in insights in explanation. Then we trained two machine learning models, namely we used random forests and gradient boosting. These models are highly versatile and easy to implement, and as such are applicable in a wide variety of problems, even on time series data. We restrained from applying time series models as to not overcomplicate the analysis.

Linear Regression
A classical linear model is given by the model equation

$$Y = X\beta + \epsilon, \tag{1}$$

where

- X is the design matrix of the features
- β is the vector of coefficients
- $\epsilon \sim N(0, \sigma^2)$ is a centered error term with unkown variance σ^2
- Y is the response variable

Random Forest
Random forests is an ensemble method for regression and classification tasks that is built on decision trees. It extends on the idea of bootstrap aggregation, or bagging, which is used to reduce the variance of a statistical learning method via averaging. In the context of regression trees, this is done by constructing B unpruned trees from B bootstrap samples and averaging the predictions $\hat{f}$ as displayed in equation 2.

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*,b}(x) \tag{2}$$

One notable drawback of bagging is that the individual trees can be very correlated depending on how strong the predictors are. Random forests address this issue by randomly selecting a subset of the features at each split and thus de-correlating the trees.

Random Forests (RF) is implemented using the *randomForest* package in R. We have chosen the maximum number of tress ntree = 50 and the node size = 5 for increased computational speed. Usually the predictive accuracy decreases when decreasing these factors, but in our case only a negligible amount of accuracy was sacrificed this way.

Gradient Boost
Gradient boosting is an ensemble machine learning technique that iteratively builds a series of decision trees to form a strong predictive model. The core idea of gradient boosting involves constructing new models that predict errors of prior models and then combining these models in a weighted manner. The process starts with a base model that makes an initial prediction, which, in most cases, consists of a shallow

decision tree. The subsequent models are built to predict the differences between the observed values and the predictions from the current group of trees.

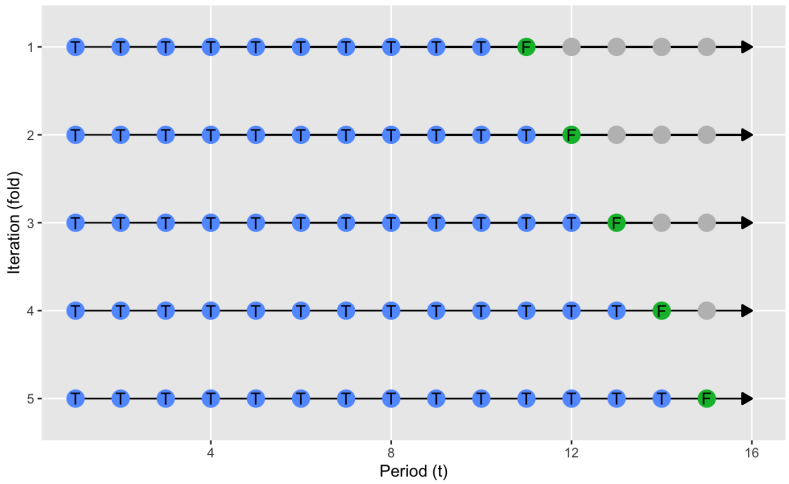
Each new tree is fit on the gradient of the loss function with respect to the predictions. The loss function quantifies how far the current model's predictions are from the target values. By fitting each tree to these gradients, the algorithm focuses on the most challenging cases where the current ensemble performs poorly.

We used the *gbm* package in R for an easy implementation of the gradient boosting method. There are a number of variable parameters to control in the gbm function. We set the target distribution to a Poisson distribution, since this distribution is often used to model count data. Then we set the number of boosting iteration ntrees = 1000 and the tree depth interaction.depth = 4 for faster computation speed. For the other parameters the default settings were used.

Model Evaluation

In order to evaluate and to compare the predictive power of the models we need to consider how to perform validation. If we were to randomly split the data into training and testing data, then we would be using information of the past as well as the future to make predictions on the testing data, which is not possible in real world problems. Thus we need to split the data at a certain time point and use the data before that time point for training and after that for testing. However, we also do not want to make predictions on the whole batch of the testing data, since forecasting very far into the future is not realistic and accurate. Thus the most sensible approach is using time series cross-validation. As illustrated in Figure 3, we split the data at a time point and then first only make a prediction for the next time period. Then in the next step that time period becomes part of the training data and we forecast the next time period. This way we can iteratively go through the testing data and obtain substantiated predictions.

Figure 3: **Time series cross-validation**



In our case we chose to go through 30 iterations of time series cross-validation, which means that we predicted the hourly rental count of the last 30 days of the data.

Each test fold contains 24 observations corresponding to the next 24 hours at a time point.

Now that we have established the validation method we define the error metrics that we used to evaluate the predictions. We used the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) to select the best model among the list of candidate models. After computing these metrics for each 24 hour forecast, we calculated the average (as in equations 3 and 4) in order to obtain error rates for the whole 30 day testing period.

$$\text{MAE} = \frac{1}{30 \cdot 24} \sum_{i=1}^{30} \sum_{j=1}^{24} |y_{i,j} - \hat{y}_{i,j}| \tag{3}$$

and

$$\text{RMSE} = \frac{1}{30 \cdot 24} \sum_{i=1}^{30} \sum_{j=1}^{24} (y_{i,j} - \hat{y}_{i,j})^2 \tag{4}$$

Here $y_{i,j}$ and $\hat{y}_{i,j}$ denote the count and predicted count at day i and hour j of the testing data, respectively.

4 Results

We report the MAE and RMSE to compare the predictive power of the models. These results are displayed in Table 4. We notice that the MSE and RMSE establish the same performance ranking of the three models. As expected, the linear regression model gave the worst predictive performance out of the three models with an MAE of 105 and RMSE of 134. While the other two models generated significantly better results, the Gradient Boost model performed the best with an MAE of 48 and RMSE of 64. Hence we select the Gradient Boost model as the best predictive model.

Table 4: **Comparison of Predictive Models**

Model	MAE	RMSE
Linear Regression	105.18	134.46
Random Forest	59.3	77.68
Gradient Boost	47.7	63.9

In order to evaluate the significance of the different features, we created a feature importance plot that is generated from the random forest model. The *randomForest* package in R contains a function that does this naturally. This function provides a measure of the quality of the splits that each feature makes in the trees of the forest, indicating the importance of each feature. Figure 4 illustrates these results. It seems that the meteorological features corresponding to temperature and humidity are the most significant factors influencing hourly bike share demand. Other than that, the working day variable seems to contribute significantly as well. This suggests that a large portion of the daily bike usage consists of individuals using bikes on their commute to or from work.

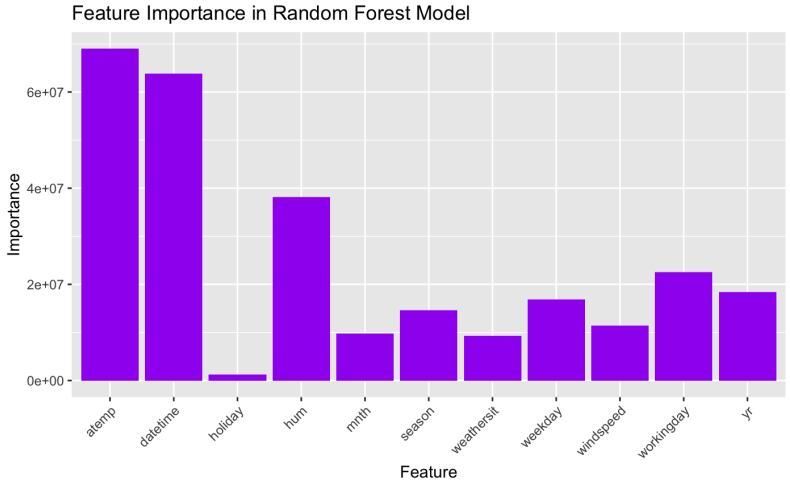


Figure 4: **Importance Plot**

Lastly, we compare the importance plot of the random forest model to the summary of the linear model, in order to evaluate whether the linear model is capable of adequately explaining feature significance. The summary (figure 5) shows that the model does not assign much significance to *workingday*, unlike the random forest importance plot. Instead, *weekday* seems to be more significant. As these two variables are closely related though, this result is not necessarily worse compared to the random forest plot.

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.276e+04  2.883e+04   1.830 0.067292 .
instant      1.423e-01  8.054e-02   1.766 0.077362 .
season       1.947e+01  1.853e+00  10.512  < 2e-16 ***
yr           1.271e+02  4.504e+01   2.823 0.004762 **
mnth         4.112e+00  3.793e+00   1.084 0.278305
hr           7.685e+00  1.652e-01  46.513  < 2e-16 ***
holiday     -2.187e+01  6.698e+00  -3.266 0.001094 **
weekday      1.875e+00  5.415e-01   3.463 0.000535 ***
workingday   4.073e+00  2.398e+00   1.698 0.089441 .
weathersit  -3.499e+00  1.907e+00  -1.835 0.066547 .
temp         7.521e+01  3.709e+01   2.028 0.042584 *
atemp        2.367e+02  4.167e+01   5.679 1.38e-08 ***
hum         -1.973e+02  6.906e+00 -28.562  < 2e-16 ***
windspeed    4.280e+01  9.659e+00   4.431 9.43e-06 ***
datetime    -4.079e-05  2.228e-05  -1.831 0.067146 .
Date                NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 141.9 on 17340 degrees of freedom
Multiple R-squared:  0.3889,    Adjusted R-squared:  0.3884
F-statistic: 788.4 on 14 and 17340 DF,  p-value: < 2.2e-16
```

Figure 5: **Summary of linear Model**

Furthermore, we notice a multiple R-squared value of 0.3889, which suggests that the model is far from providing an optimal fit to the data. The F-statistic value is very

high though. This indicates that the full model with these features fits significantly better than the null model.

5 Conclusion

The focus of this analysis has been on the ability to accurately forecast hourly bike demand for the next 24 hours. We have found models that worked reasonably well on this problem. Thus bike sharing organizations could implement these models to help ensure an adequate number of bikes are available in a 24 hour time frame. Furthermore, we have distinguished important factors that influence the hourly demand significantly. Such features were mostly of meteorological nature, for instance temperature and humidity.

The analysis clearly shows that Gradient Boosting generates the most accurate forecast of demand, as measured by mean absolute error, with an MAE of 48 bikes per hour. The random Forest model also performed similarly with an MAE of 59 bikes per hour. We found that the linear model had an R-squared coefficient of 0.3889, indicating that this model did not yield a great fit for the data. The F-statistic took on a value of 788 though, which suggests that the saturated model with all of the features fits the data much better than the null model.

Future study on this application could concentrate on adding more predictor factors and more recent bike usage data. The models used in this analysis have a major flaw in that they don't take into consideration demand variation across space. In other words, how is the demand for bikes distributed geographically at each hour? The bikeshare administrators could redistribute available bikes from low-demand areas to high-demand areas if they knew not only how many bikes were requested overall but also where in the city they are needed. The prediction would become much more complex if the demand had a geographic component because multiple individual models—one for each geography—or a single model with a multivariate response would be needed. Even though this would be a more complicated issue, it would be of significant value to bikeshare administrators, therefore it would be a good choice for more subsequent research.

References

Daily Bike Rental Dataset (12/19/2013). Accessed: 10/10/2023. URL: `https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset`.