ORIE 4740 report

Predicting Bike Rentals

Anthony Niznik / Lishan Ong / Alexis Rouge Carrassat

Data source: https://www.kaggle.com/c/bike-sharing-demand

ABSTRACT

This paper aims to predict the hourly number of bikes rented in Washington D.C.'s Capital bike share using temporal and weather factors. We use a dataset with 10,886 rows, consisting of hourly entries for the first 19 days of each month from 01/01/2011 to 12/19/2012. Models used can be categorized into linear regressions, non-linear regressions with smoothing splines, and regression trees. Throughout our analysis, we increased the number of predictors and the complexity of our models.

Our results show that increasing the number of predictors resulted in improved predictive accuracy, and the gradient boosting algorithm yielded the best training and test set performance. The hour, number of days since the start of the bike share program, and the day of week are the most important predictors. Specifically, the after work (4:30 PM to 6:30PM) and morning (before work, 6:30AM to 8AM) hours are the busiest, while night hours are the least busy. Also, the number of bikes rented has been increasing since the start of the program, thus it is likely that demand will continue to rise. Moving forward, these findings may be useful for Washington D.C. governors in resource allocation and capacity planning, while other predictors such as bike station location can be added in future.

INTRODUCTION

1. Dataset

The dataset we are working on was found on Kaggle and describe bike rentals for a Capital bike share program in Washington D.C. There are 10886 rows in the dataset, each entry being an hour of the day (from 01/01/2011, at midnight to 12/19/2012, at 11 pm). The training set comprises of the first 19 days of each month. The overall goal is to predict, hour by hour, how many bikes are being rented.

In the original data set, we are given data fields concerning time and weather that will help us predict the behavior of bike users in the city (Table 1). There are no missing values in this dataset.

Table 1 Description of variables in original dataset

Data Fields	**Data Type**	**Predictor Type**	**Description**
Date and time	Datetime	Temporal	Hourly date and time stamp
Season	Ordinal	Temporal	1: Spring, 2: Summer, 3: Fall, 4: Winter (Spring: January to March, and so on)
Holiday	Binary	Temporal	Whether the day is considered a holiday
Working day	Binary	Temporal	Whether the day is neither a weekend nor holiday
Weather	Ordinal	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 pallets + thunderstorm + mist, Snow + fog
Temp	Numeric	Weather	Actual temperature in °C
Atemp	Numeric	Weather	“Feels-like” temperature in °C
Humidity	Numeric	Weather	Relative humidity
Wind speed	Numeric	Weather	Wind speed
Casual	Numeric	-	Number of non-registered user rentals initiated
Registered	Numeric	-	Number of registered user rentals initiated
Count	**Numeric**	**-**	**Output variable: Number of total bikes rented.** This is equal to the number of casual users + the number of registered users.

2. Data cleaning and adding features

When exploring the dataset, we realized that some data were inaccurate. Indeed, the feature “season” did not match the actual seasons (E.g. “summer” is from April to June). In order to improve our analysis, we created new features:

- ✓ “hour”: indicating the hour of the day. Can be used as binary or ordinal variable.
- ✓ Time of day predictors --- “night”, “morning”, “afternoon”, “afterwork”, “evening”: binary variables indicating at what time of day the bikes are being rented
  - ○ “night” – 11pm to 5am
  - ○ “morning” – 6am to 10am
  - ○ “afternoon” – 11am to 3pm
  - ○ “afterwork” – 4pm to 7pm
  - ○ “evening” – 8pm to 10pm
- ✓ “winter”, “spring”, “summer”, “autumn”: 4 binary variables indicating the true seasons.
  - ○ “winter” – December to February
  - ○ “spring” – March to May
  - ○ “summer” – June to August
  - ○ “autumn” – September to November
- ✓ “dayofweek”: from 0 to 6, 0 being Monday. Can be used as binary or ordinal variable.
- ✓ “nice_weather”, “cloudy”, “rain_snow”, “very_bad_weather”: binary variables indicating the type of the weather for each hour of the days (we used the original feature “weather”, which takes values from 1 to 4, 1 being “Nice_weather” and 4 being very_bad_weather”)
- ✓ “timediff’: stands for time difference, which is a running integer that keeps track of the number of days passed since 01/01/2011, the start of the bike share program.

Variables “registered” and “casual” (which sum to “count”) were not used at all, as we are only interested in predicting the total number of bikes rented.

3. Data summary and visualization

We first obtained summary statistics for selected numeric and categorical variables (Table 2 and Table 3) for the training data.

Table 2 Summary statistics for selected numeric variables

	Temp	**Atemp**	**Humidity**	**Wind Speed**	**Count**
Min	0.82	0.76	0.00	0.000	1.0
Mean	20.23	23.66	61.89	12.799	191.6
Median	20.50	24.24	62.00	12.998	145.0
Max	41.00	45.45	100.00	56.997	977.0

Table 3 Counts of selected categorical variables

Holiday	**Working Day**	**Weather**
0: 10575 1: 311	0: 3474 1: 7412	1: 7192 2: 2834 3: 859 4: 1

We examined the relationship between bike rentals and other predictors (Figure 1).

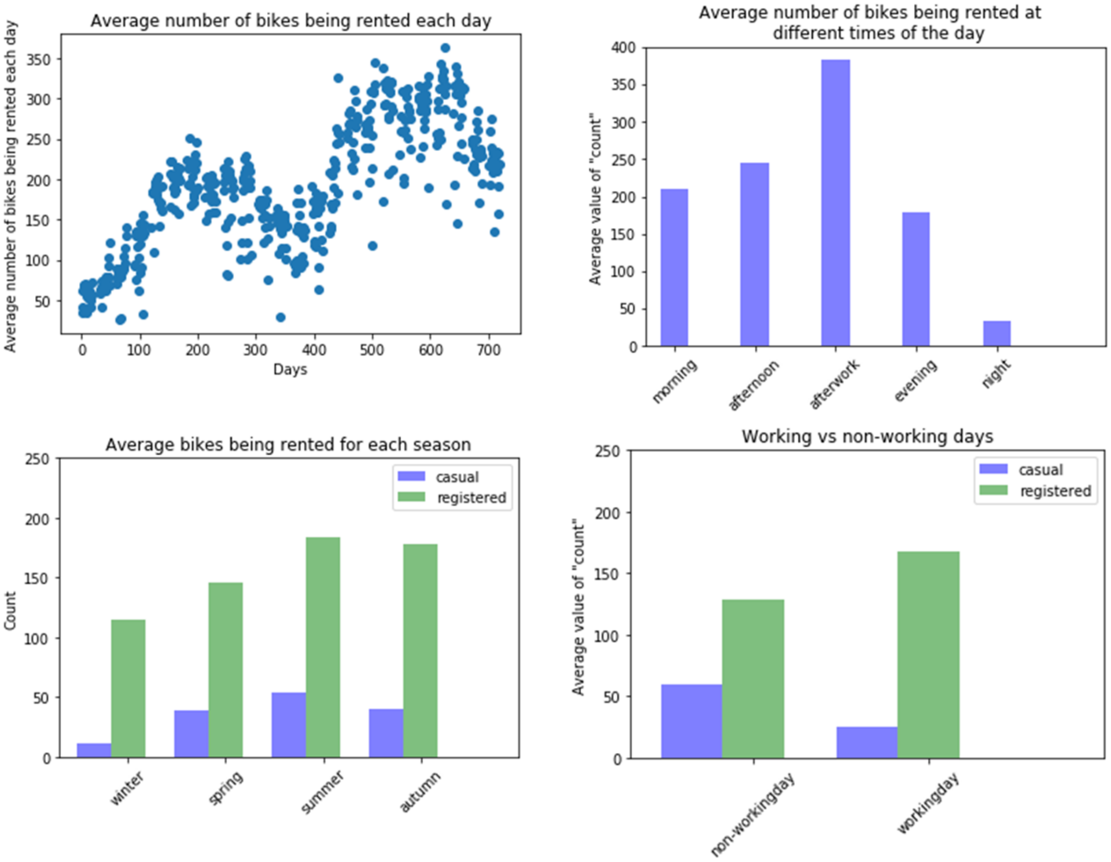


Figure 1 Visualization of the number of bikes rented with respect to other variables

From the top-left panel, we see that there is an overall increasing trend in the average number of bikes rented across two years. From the top-right panel, we see that the average number of bikes rented is highest after work and lowest at night. From the bottom-left panel, there are more registered users than casual users, and there are unsurprisingly more bike rentals in the summer than in other seasons. Lastly, casual users rent more bikes on non-working days, whereas registered users rent more bikes on working days.

BASELINE MODEL

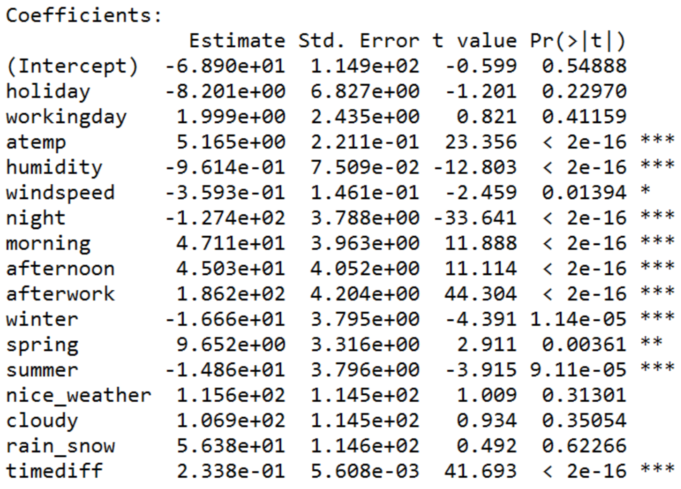
Before building any predictive models, we applied an algorithm that predicts the mean number of bikes rented as a baseline model, which is 191.57 bikes. The cross-validation mean-squared error (MSE) is 32821.0. By comparing the results of the subsequent models to that of the baseline model, we can see if the subsequent models do better than the baseline.

LINEAR REGRESSION WITH 16 PREDICTORS

1. Ordinary least squares regression

We first built a regression model using the following 16 features (Figure 2). We used “atemp” instead of “temp” because we felt that the feels-like temperature would be more relevant than the actual temperature. Since both variables have a high correlation of 0.985, it would make sense to only include one, to reduce multicollinearity.

Figure 2 shows the coefficients obtained from the linear regression model. Features “atemp”, “humidity”, “night”, “morning”, “afternoon”, “afterwork", “winter”, “summer” and “timediff” have very low p-values, showing they have a strong influence on the variable count. Variables “holiday”, “humidity”, “wind speed”, “night”, “winter” and “summer” are negatively associated with bike rentals. “Afterwork” has the most positive coefficient (186), followed by “nice weather” (116) and “cloudy” (107). “Night” has the most negative coefficient (-127), followed by “winter” (-17) and “summer” (-15). “Holiday”, "working day" and "weather" are not significant. The R^2 value and cross-validation error for this model is 0.602 and 13103.4 respectively. The test MSE is much lower than that of the baseline (32820.99).

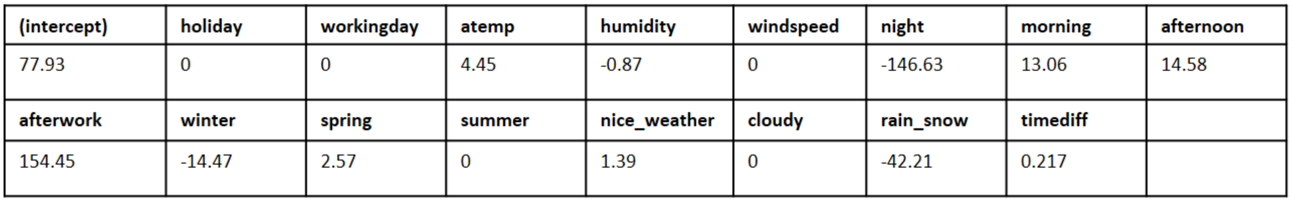


```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -6.890e+01  1.149e+02  -0.599  0.54888    
holiday      -8.201e+00  6.827e+00  -1.201  0.22970    
workingday    1.999e+00  2.435e+00   0.821  0.41159    
atemp         5.165e+00  2.211e-01  23.356  < 2e-16 ***
humidity     -9.614e-01  7.509e-02 -12.803  < 2e-16 ***
windspeed    -3.593e-01  1.461e-01  -2.459  0.01394 *  
night        -1.274e+02  3.788e+00 -33.641  < 2e-16 ***
morning       4.711e+01  3.963e+00  11.888  < 2e-16 ***
afternoon     4.503e+01  4.052e+00  11.114  < 2e-16 ***
afterwork     1.862e+02  4.204e+00  44.304  < 2e-16 ***
winter       -1.666e+01  3.795e+00  -4.391 1.14e-05 ***
spring        9.652e+00  3.316e+00   2.911  0.00361 ** 
summer       -1.486e+01  3.796e+00  -3.915 9.11e-05 ***
nice_weather  1.156e+02  1.145e+02   1.009  0.31301    
cloudy        1.069e+02  1.145e+02   0.934  0.35054    
rain_snow     5.638e+01  1.146e+02   0.492  0.62266    
timediff      2.338e-01  5.608e-03  41.693  < 2e-16 ***
```

Figure 2 Coefficients from linear regression

2. Regularized regression: LASSO and Ridge regression

For greater interpretability and to allow us to focus on a subset of features, we performed LASSO and ridge regression. LASSO is a feature selection method that forces some features to 0, while ridge regression shrinks the coefficients. LASSO selected 12 out of 16 features (Figure 3). The 4 unselected features are mostly those which are insignificant in the unregularized regression.



(intercept)	**holiday**	**workingday**	**atemp**	**humidity**	**windspeed**	**night**	**morning**	**afternoon**
77.93	0	0	4.45	-0.87	0	-146.63	13.06	14.58
afterwork	**winter**	**spring**	**summer**	**nice_weather**	**cloudy**	**rain_snow**	**timediff**	
154.45	-14.47	2.57	0	1.39	0	-42.21	0.217	

Figure 3 LASSO coefficients

Both LASSO and ridge obtained similar training and test error as compared to the unregularized regression. We recommend using the LASSO model as it achieves similar results with fewer predictors.

LINEAR REGRESSION WITH 41 PREDICTORS

Next, we attempted another linear regression with greater granularity in the predictors, where “workingday” was replaced with “dayofweek” and the time of day predictors (“morning”, “afternoon”, “afterwork”, and “night”) were replaced with “hour”. This resulted in a total of 41 predictors when categorical variables were converted to dummies.

1. Ordinary least squares regression

The most positive coefficient is “hour5pm” (388), followed by “hour6pm” (354) and “hour8am” (314), coinciding with the afterwork and morning hours. The most negative coefficients are “hour4am” and “hour3am” (-40), followed by “hour2am” (-28), coinciding with the morning hours. The R^2 value and 10-fold cross-validation error for this expanded model is 0.691 and 10216.6 respectively, showing an improvement compared to the 16-predictor regression in both in-sample and out-of-sample error.

2. Regularized regression: LASSO and Ridge regression

37 predictors out of 41 are selected with LASSO. It is probably not a coincidence that the most positive and negative coefficients of the LASSO coincide with those of the unregularized regression (Table 4). “Hour2pm”, “dayofweek1” (Tuesday), “dayofweek2” (Wednesday) and “cloudy” are 0 in the LASSO model. The last 3 unselected variables are not significant in the unregularized regression, which might explain why these are not selected. The R^2 is 0.689, close to that of the unregularized regression. The tuning parameter that gives the lowest cross-validation error is 1, with a cross-validation error of 10256.9. This cross-validation error is slightly higher than that of the unregularized regression. This is expected because the LASSO regression, through reducing variance, introduces bias due to the bias-variance tradeoff.

Ridge regression shrinks the coefficients of the unregularized regression but does not force any to 0. The ridge regression model’s R^2 value is 0.691, close to that of the unregularized regression. The tuning parameter of 0.5 that gives the best cross-validation error of 10219.8. The coefficients are similar to that of the LASSO (Table 4).

Table 4 Comparing most positive and negative coefficients of different regressions

Unregularized regression		**LASSO regression**		**Ridge regression**	
Predictor	**Coefficient**	**Predictor**	**Coefficient**	**Predictor**	**Coefficient**
hour5pm	388	hour5pm	229	hour5pm	224
hour6pm	354	hour6pm	195	hour6pm	190
hour8am	314	hour8am	157	hour8am	150
hour2am	-28	hour2am	-175	hour2am	-191
hour3am	-40	hour3am	-187	hour3am	-202
hour4am	-40	hour4am	-187	hour4am	-203

We conclude that using more predictors both leads to a higher R^2 and lower cross-validation error, hence adding more predictors leads to better predictions rather than overfitting.

NON-LINEAR METHODS: SMOOTHING SPLINES

So far, we have used linear regression models. While they are simple to construct, they rely on the linear assumption. Here, we used general additive models with smoothing splines because they allow to us to model a non-linear relationship between selected variables with the outcome, which can potentially make more accurate predictions. Smoothing splines were also chosen because of their continuous and smoothness properties, and the constraint that they are linear at the boundary.

First, we made four pairwise comparisons between the linear model with the 16 features (as in the unregularized regression) with 4 other models which had “atemp”, “humidity”, “windspeed” and “timediff” as splines with 3 degrees of freedom. Using a significance level of 0.001, the ANOVA tests shows that using a non-linear model for “atemp” and “humidity” is needed, while using a non-linear model for “windspeed’ and “timediff’ is not needed. From Figure 4, even when a smoothing spline is used for “windspeed” and ‘timediff” respectively, the splines appear close to linear.

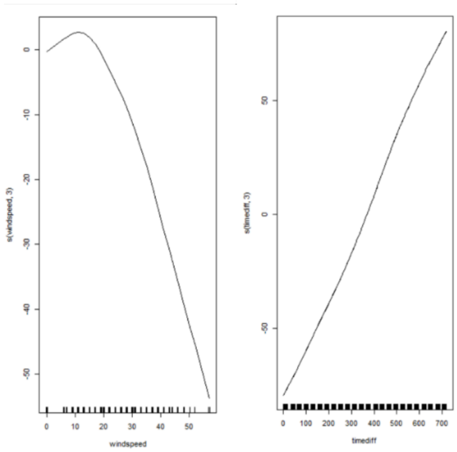


Figure 4 Smoothing spline models for “windspeed” (left) and “timediff” (right)

Therefore, keeping “windspeed” and “timediff’ linear, we performed ANOVA tests for various generative additive models which kept all 16 features to be linear except for “atemp” and “humidity”. Bike rental peaks at around 35°C and 30% humidity (Figure 5). Using the same degrees of freedom for “atemp” and “humidity”, we compared models with 3, 4, 5 and 6 degrees of freedom. The ANOVA test shows that we should choose the model where “atemp” and “humidity” have 5 degrees of freedom. This chosen general additive model has a p-value of < 2.2e^-16 when compared to the fully linear model, with a R^2 and cross-validation MSE of 0.609 and 12831.4 respectively.

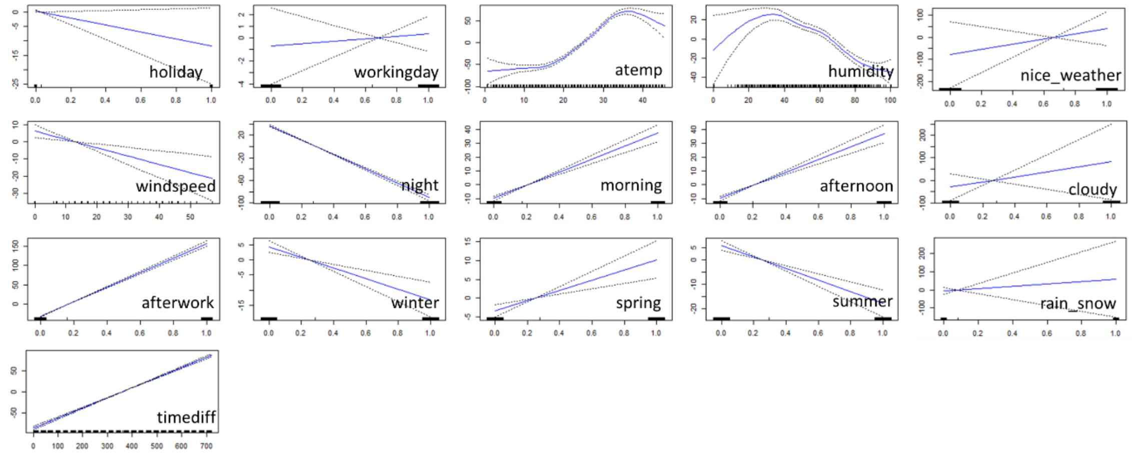


Figure 5 Plots for general additive models with smoothing splines

REGRESSION TREES

So far, we have used additive models (linear and non-linear). Moving forward, we decided to use a tree-based approach which partitions the feature space into non-overlapping rectangular regions, and then predicts the outcome to be the average of the points in the region. Conceptually, they are simple to interpret and can handle categorical variables well.

1. Single Tree

We first decided to use a basic tree since it is the easiest to interpret and to see which variables the decision tree decides to split on. To prevent overfitting, we built a full tree and pruned it to an optimal size. From cross-validation, the best tree is one that has 10 terminal nodes. The same 41 predictors were used, except that hour was used as an ordinal variable. We then plotted a pruned decision tree with 10 terminal nodes to see which variables are most important to split on (Figure 6).

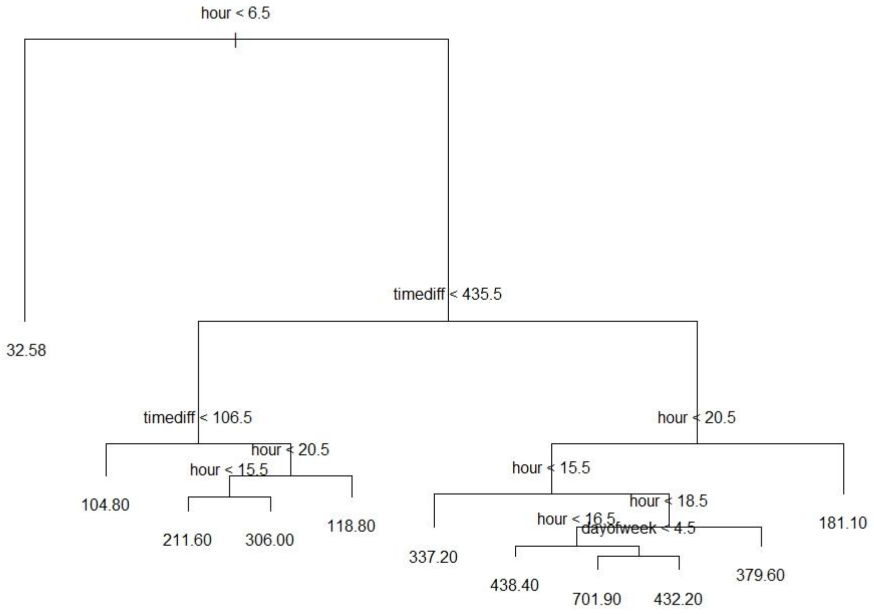


Figure 6 Decision tree plot for a single pruned tree

We see that the first node that is split on is whether the hour is 6:30 AM or earlier, where the number of bike rentals after this time are higher than the ones before. The next node that is split on is “timediff” and we can see that after about 1 year into the program, the number of rentals is higher generally. This increase in bike rentals may be due to greater exposure and availability of more bikes.

The following nodes that are split on are “hour” and “dayofweek”. To get a sense of the greatest demand for bikes, we go through the nodes that get to the highest amount of bike rentals (701.90). We see that after over a year of the bike rental program, and if the time was after 4:30 PM but before 6:30 PM during a work week (Monday through Friday), there was the highest number of bike rentals. This makes sense because it means most people rented a bike right after work to get back home. These results tally with the regression where “afterwork” has the most positive coefficient.

We found the R^2 on the training set yields 0.663 and that the MSE is 11067.9, which is better than the linear regression methods with 16 predictors but worse than that with 41 predictors.

2. Bagging

Next, we tried bagging, which reduces the variance by averaging many trees which makes our results more accurate but at the cost of less interpretability. We split the data set into 2 parts: 80% for training and 20% for testing. We see that the MSE starts to stabilize at 100 trees (Figure 7). The R^2 is 0.949 and MSE is 1645.2 for 100 trees. This is a drastic improvement from the single tree ---- the R^2 is very close to 1 and MSE is now reduced by 7 times.

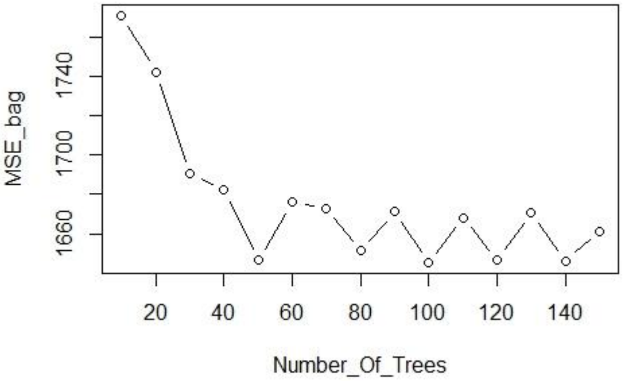


Figure 7 Bagging: MSE vs the number of trees

3. Random Forest

We then went on to use random forest because it decorrelates trees by random feature selection. This is important because averaging uncorrelated trees will further reduce our overall variance and lead to even better results. When we used random forest with 100 trees and tried a different number of variables to select, we see that MSE is minimized when the number of variables used is 7 (Figure 8).

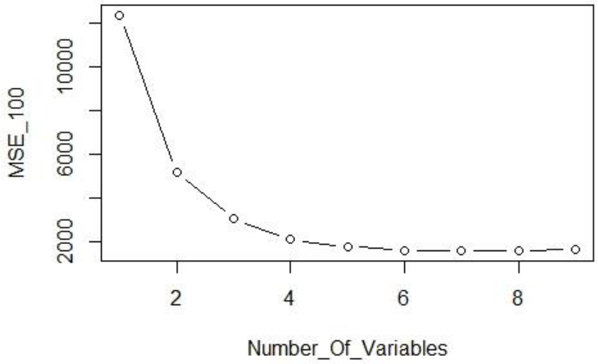


Figure 8 Random forest: MSE vs the number of variables with 100 trees

The R^2 is 0.951 and MSE is 1598.7 when 7 variables are used. Unsurprisingly, the random forest performs better than bagging because of the decorrelation between trees. From the variance importance plot (Figure 9), we see that the hour of the day, the day of the week, and the days since the bike rental was put in place are very important in the model. These results match the insights from the single decision tree plot we created earlier (Figure 6).

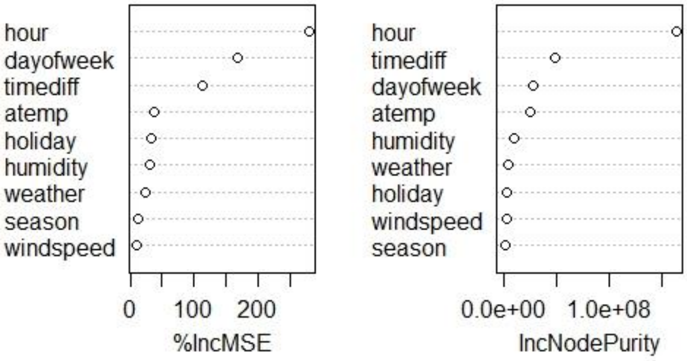


Figure 9 Random forest: Variable importance plot

4. Gradient Boosting

Our last model is gradient boosting. Gradient boosting is effective as a boosting procedure because the learning procedure ensembles weak learners iteratively to fit the model better, which are maximally correlated with the negative gradient of the loss function. Using 1000 trees with 85 leaves each, and the full set of predictors, we obtained the variable importance plot (Figure 10). Here, the full set of predictors refer to the 41 predictors used in the regression, as well as actual temperature (“temp”), and the less granular predictors “workingday” and the time of day (e.g. “morning”). Similar to the random forest, “hour”, “timediff” and “dayofweek” are the top three most important features.

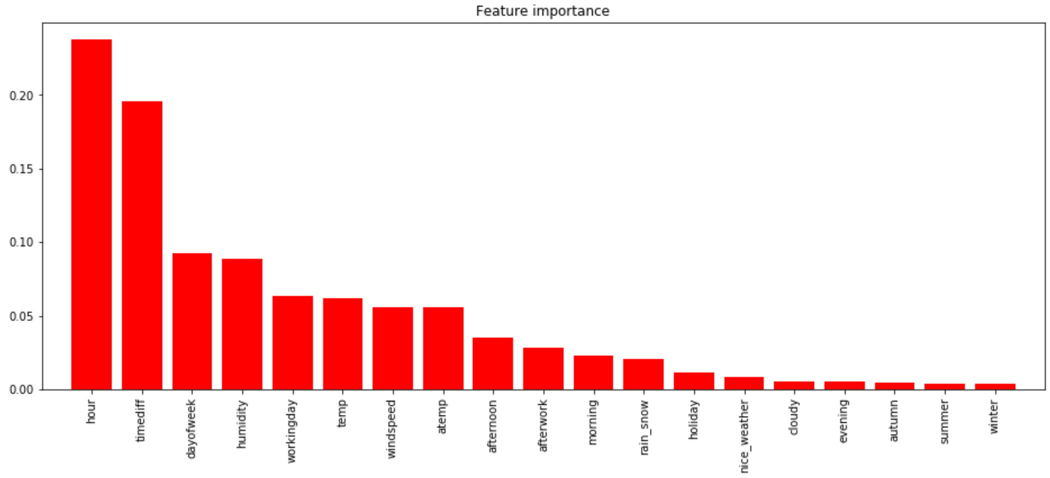


Figure 10 Gradient boosting: Variable importance plot

Out of all models, gradient boosting gives the best R^2 value of 0.958 and test MSE of 1404.2. Figure 11 shows that predicted number of bikes rented is very close to the actual figures.

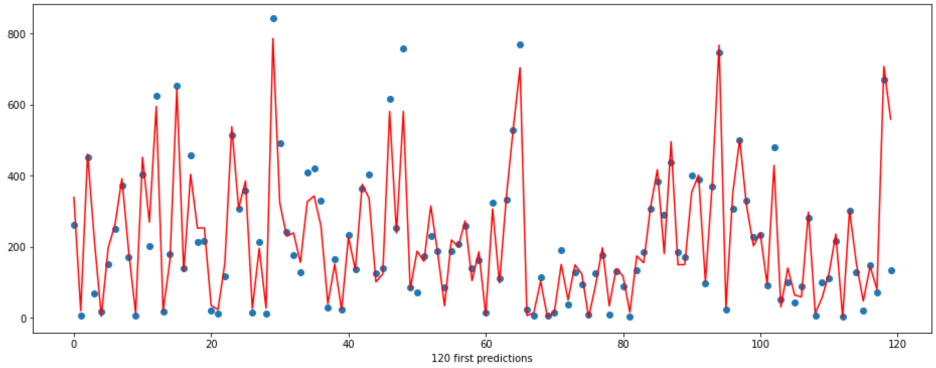


Figure 11 Predicted (lines) vs actual (dots) number of bike rentals for first 120 rows

MODEL EVALUATION

This paper aims to predict hourly bike rentals in Washington D.C.'s Capital bike share program. The predictors used were categorized into two types: temporal (e.g. hour, season) and weather factors (e.g. temperature, wind speed), and models built were categorized into linear regressions, non-linear splines and regression trees. Throughout the paper, we proceeded from simple models and predictors to more complex ones, with increasing predictive accuracy (Table 5). First, we started with the less granular 16 predictors, which included “workingday” (binary variable) and time of day variables (“morning”, “afternoon” etc). These linear and non-linear models performed quite poorly, with a test MSE of about 13000. Regularization increased MSE, while non-linear regression with smoothing splines reduced the MSE slightly due to the non-linear relationship between “atemp” and “humidity” with the number of bikes rented.

Upon replacing “workingday” and time of day predictors with more granular ones like “dayofweek” (6 dummy variables) and “hour” (23 dummy variables or one ordinal variable), the MSE of the regressions and single decision tree dropped to 10200. This showed that adding more predictors helped to fit the model better with better performance in both training and test data, thus we did not have to worry about overfitting. Regularization is preferred over unregularized regression as similar results were obtained with fewer predictors.

Performing bagging and random forests by averaging 100 trees subsequently demonstrated a drastic improvement where MSE dropped by about 6 times to 1600 (as compared to a single tree). This is because bagging and random forests reduce variance by taking averages, while random forests performed better than bagging by decorrelating trees. Bagging and random forests performed extremely well on the training set, with R^2 values of close to 1.

Finally, our gradient boosting model demonstrated the best performance. Unlike random forests where at each iteration, the tree is trained independently from previous trees, gradient boosting iteratively adds trees so that the next tree is trained to improve the already trained ensemble. Boosting has been widely known to be a quick and effective algorithm that performs well in Kaggle competitions. Our gradient boosting used a full set of predictors (including both granular and less granular predictors). Doing so resulted in the lowest estimated test error of 1404, which shows that we were not overfitting.

Table 5 Summary of all models on bike rentals dataset

	R^2 on Training Set	**Estimated Test Error (Mean Squared Error)**
Baseline model (Predict mean)		32821.0
Starting with 16 predictors		
Ordinary least squares regression	0.602	13103.4
LASSO regression	0.601	13106.5
Ridge regression	0.601	13106.3
General additive model with splines	0.609	12831.4
Starting with 41 predictors		
Ordinary least squares regression	0.691	10216.6
LASSO regression	0.689	10256.9
Ridge regression	0.691	10219.8
Single Tree	0.663	11067.9
Bagging	0.949	1645.2
Random Forest	0.951	1598.7
With full set of predictors		
Gradient Boosting	0.958	1404.2

CONCLUSION

In determining the number of bikes rented, the gradient boosting and random forest models both suggest that hour, number of days since the start of the bike share program, and day of week are the most important. Specifically, the after work (4:30 PM to 6:30PM) and morning (before work, 6:30AM to 8AM) hours are the busiest, while night hours are the least busy. Also, the number of bikes rented has been increasing since the start of the program, thus it is likely that demand will continue to rise. We believe that these findings may be useful to the Washington D.C. governors in resource allocation and capacity planning. Moving forward, the bike share program can look beyond temporal and weather data, and include location data so that resource allocation can be improved by specific bike stations.