Bike Sharing Demand

Shouvik Sachdeva, Srijan R. Shetty, Pratik Somani, Sai Krishna

kaggle

Problem Statement

Combine historical usage patterns with weather data in order to forecast bike rental demand in the Capital Bikeshare program in Washington, D.C.

Motivation

- Bike Sharing systems function as a sensor network
- Duration, departure, arrival recorded
- Useful for studying mobility in a city

Dataset

- This dataset was provided by Hadi Fanaee Tork using data from Capital Bikeshare.
- Hosted by UCI Machine Learning Repository
- Training Data: 10866 observations of 12 variables
- Test Data: 6493 observations of 9 variables

Dataset

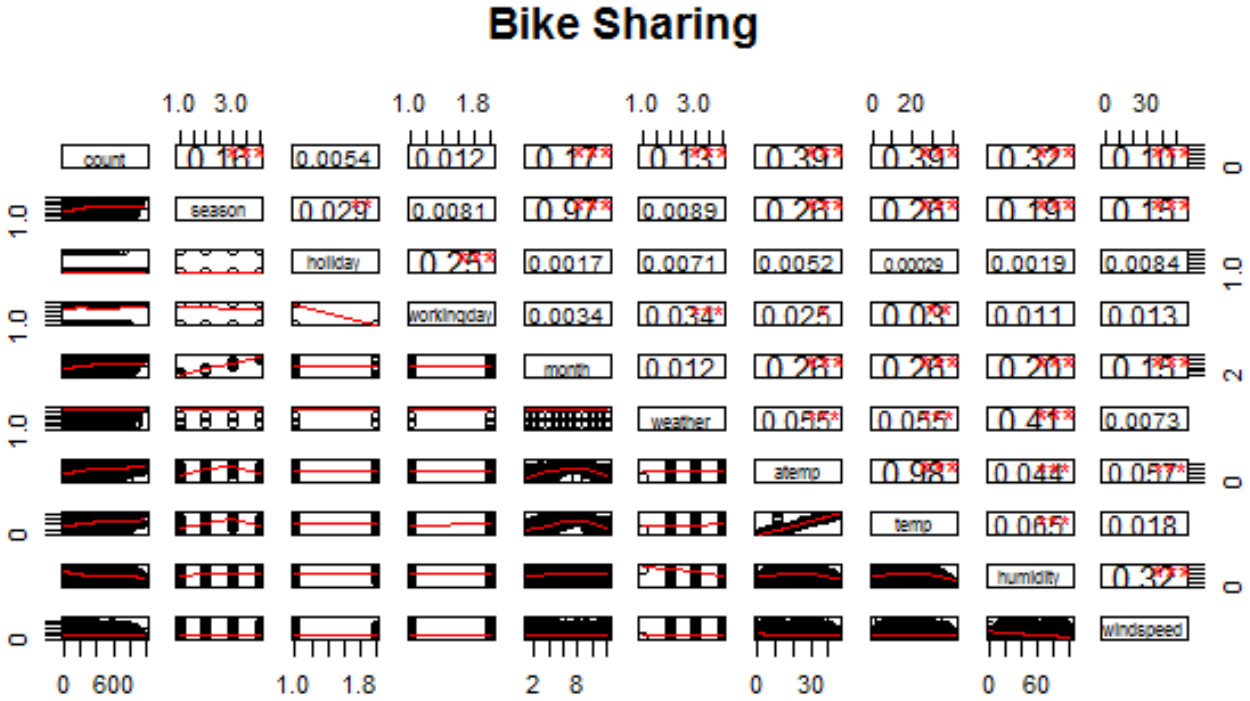
Datetime	Date and Time Format
Season	Integer: 1(Spring), 2(Summer), 3(Fall), 4(Winter)
Holiday	Boolean: 1(Holiday)
Working Day	Boolean: 1(Working Day)
Weather	Integer: 1(Clear), 2(Mist), 3(Snow), 4(Heavy Rain)
Temp	Decimal: In degree Celsius

Dataset

Atemp	Decimal: Apparent temperature in degree Celsius
Humidity	Integer: Relative humidity percentage
Windspeed	Decimal: Speed of air
Casual	Integer: # of non-registered bike shares for the hour
Registered	Integer: # of registered bike shares for the hour
Count	Integer: Total # of bike shares for the hour

Feature Engineering

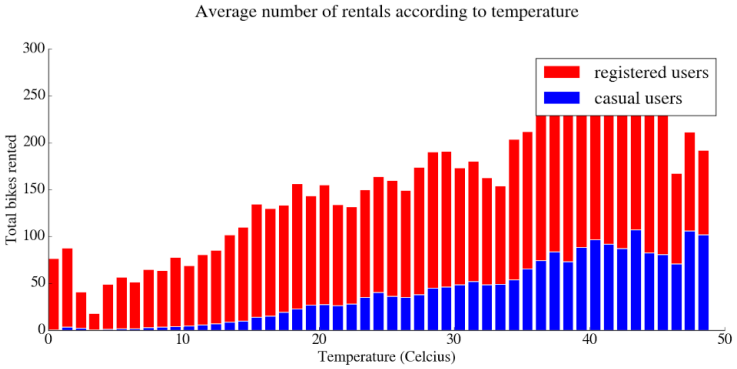
- Convert season, holiday, working day and weather into factors
- Convert date-time into time stamps (split day and hour)
- Convert hours to factors
- Day of the week
- Year
- Remove extraneous features using Correlation Matrix

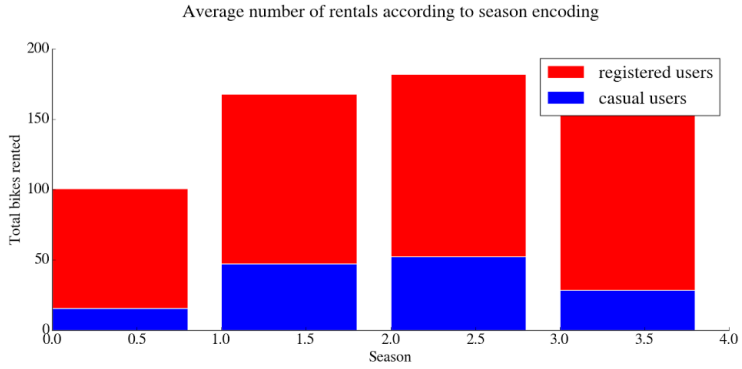


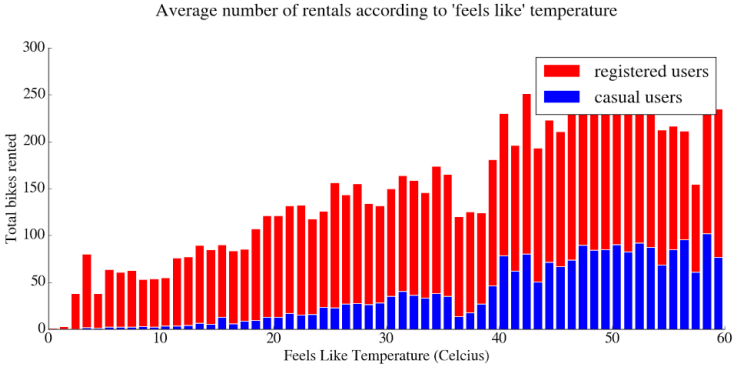
Correlation Matrix

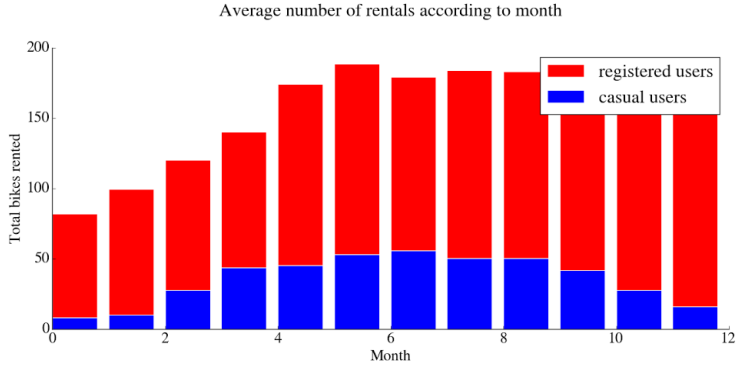
Data Analysis

- We analysed the trends of the demand w.r.t each feature
- ‘temp’ and ‘atemp’ were highly correlated with a correlation coefficient greater than 0.98.
- ‘month’ and ‘season’ also had a high correlation coefficient of 0.97

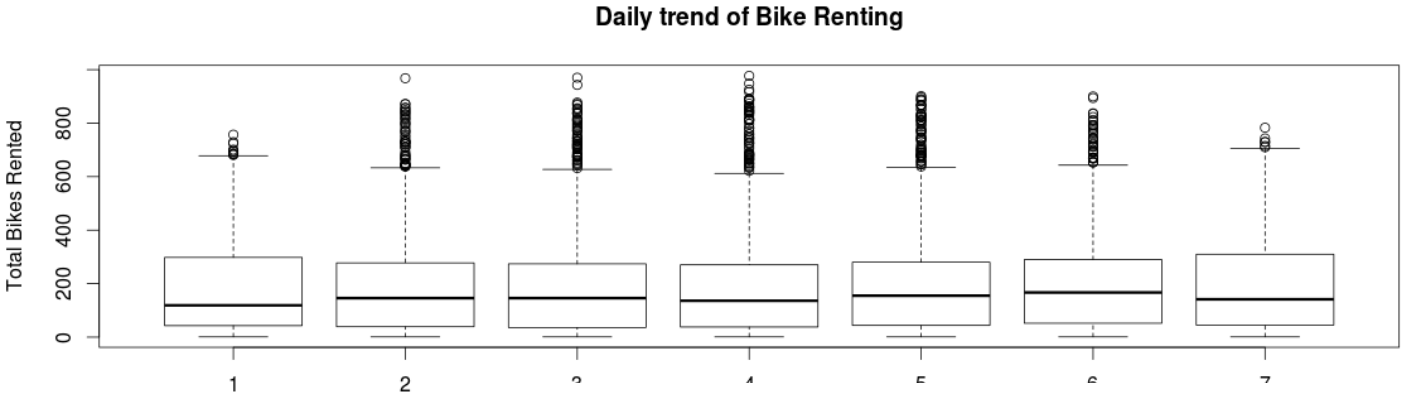


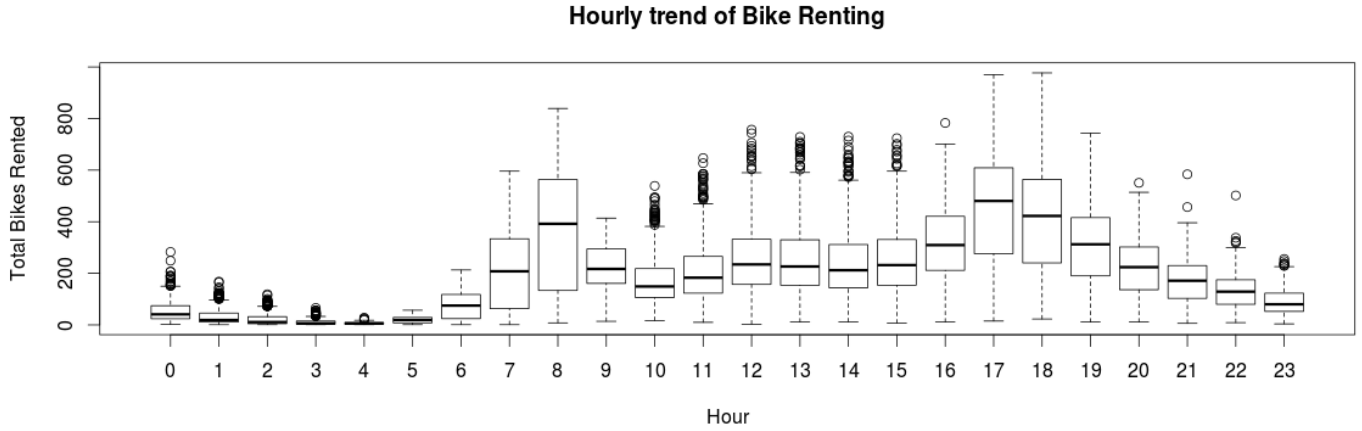




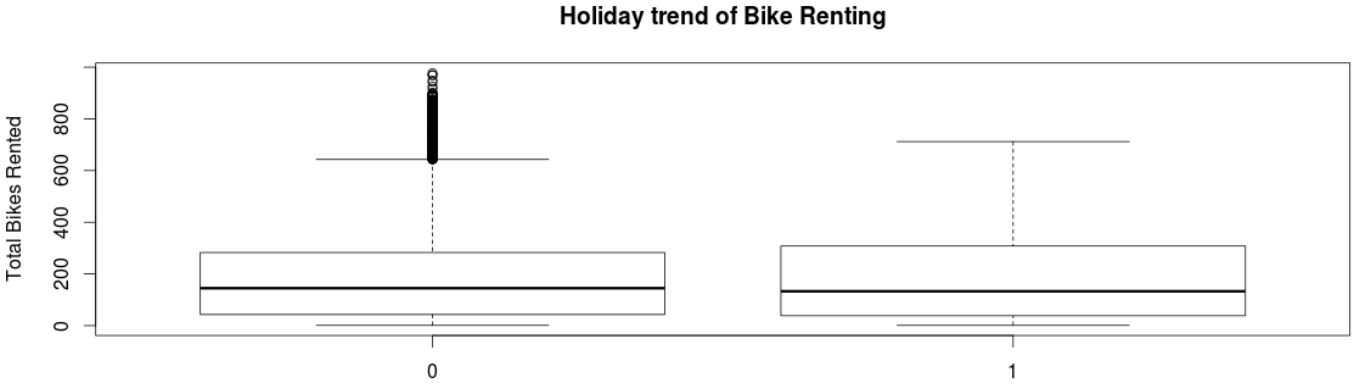


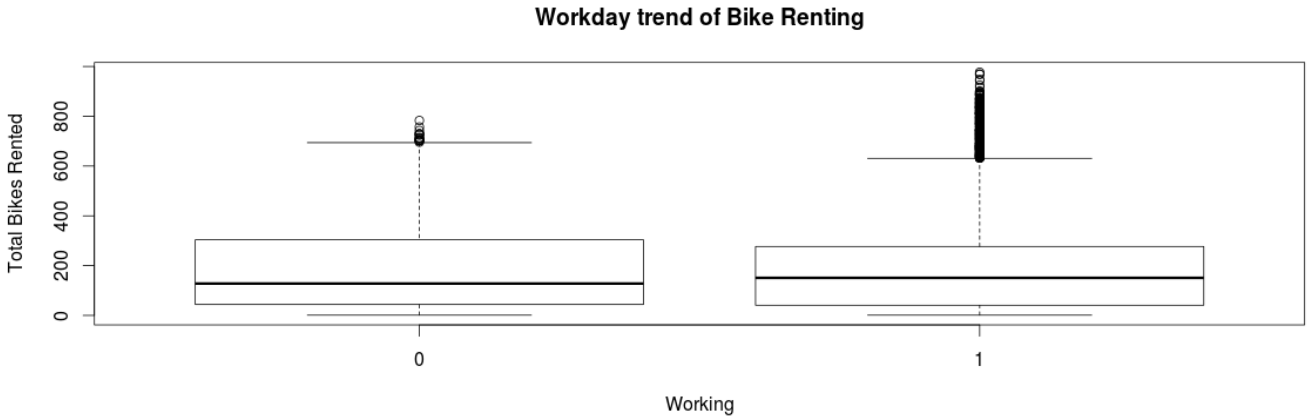
temp vs atemp, season vs month



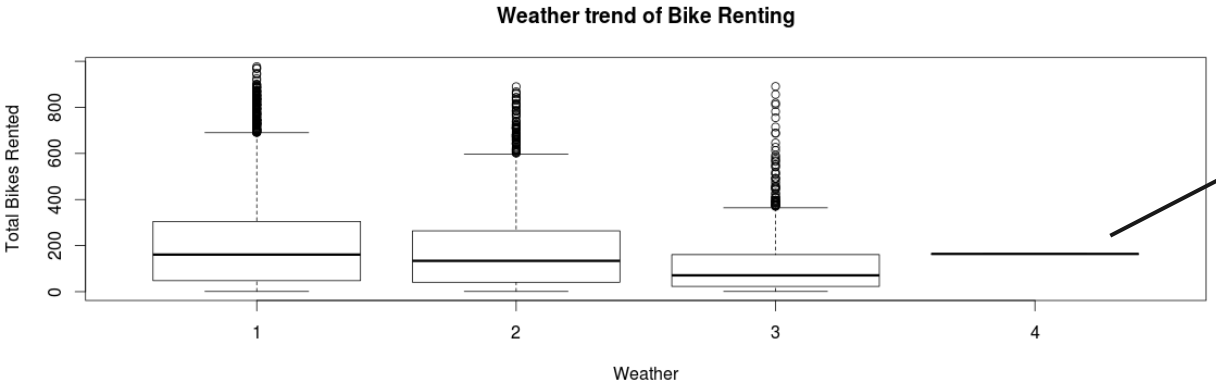


Daily and Hourly trend

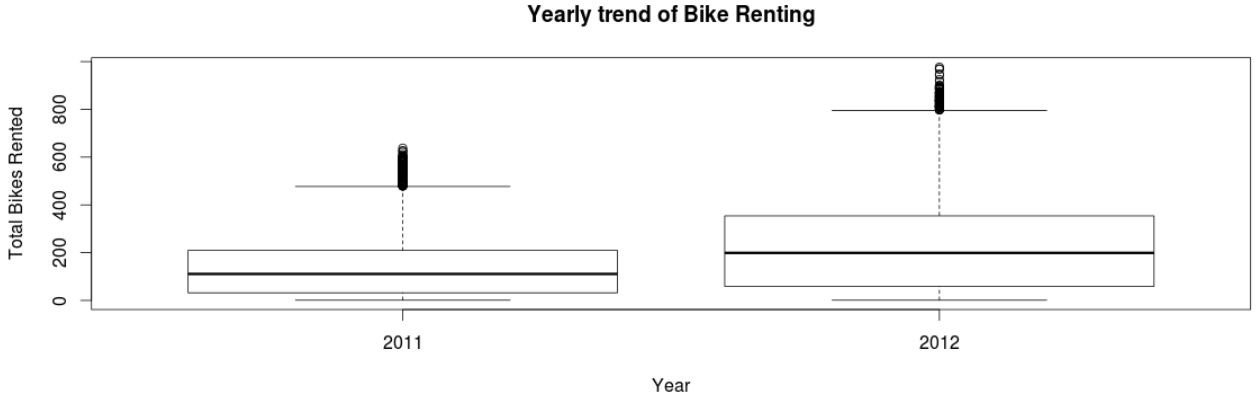




Holiday and Workday trend



The value of ‘4’ seemed to be an outlier but experimental results didn’t bolster our hypothesis



Weather and Yearly trend

Methods

- SVM
- GBM
- Random Forest
- Extra Tree Regressor
- Neural Net
- Poisson Regression
- Linear Combination of GBM and RF
- Discriminating Linear Combination

Salient Points

- Given the fact that most of the features are factors, it made much more sense to use Decision Trees as the base regressor.
- Also the number of dimensions was low ~ 10, hence the problem was tractable even without dimensionality reduction.

Approach

kaggle

Standard Approaches

Idea

- We tried the following standard approaches:
  - SVM with different kernels
  - Neural networks
  - Poisson Regression

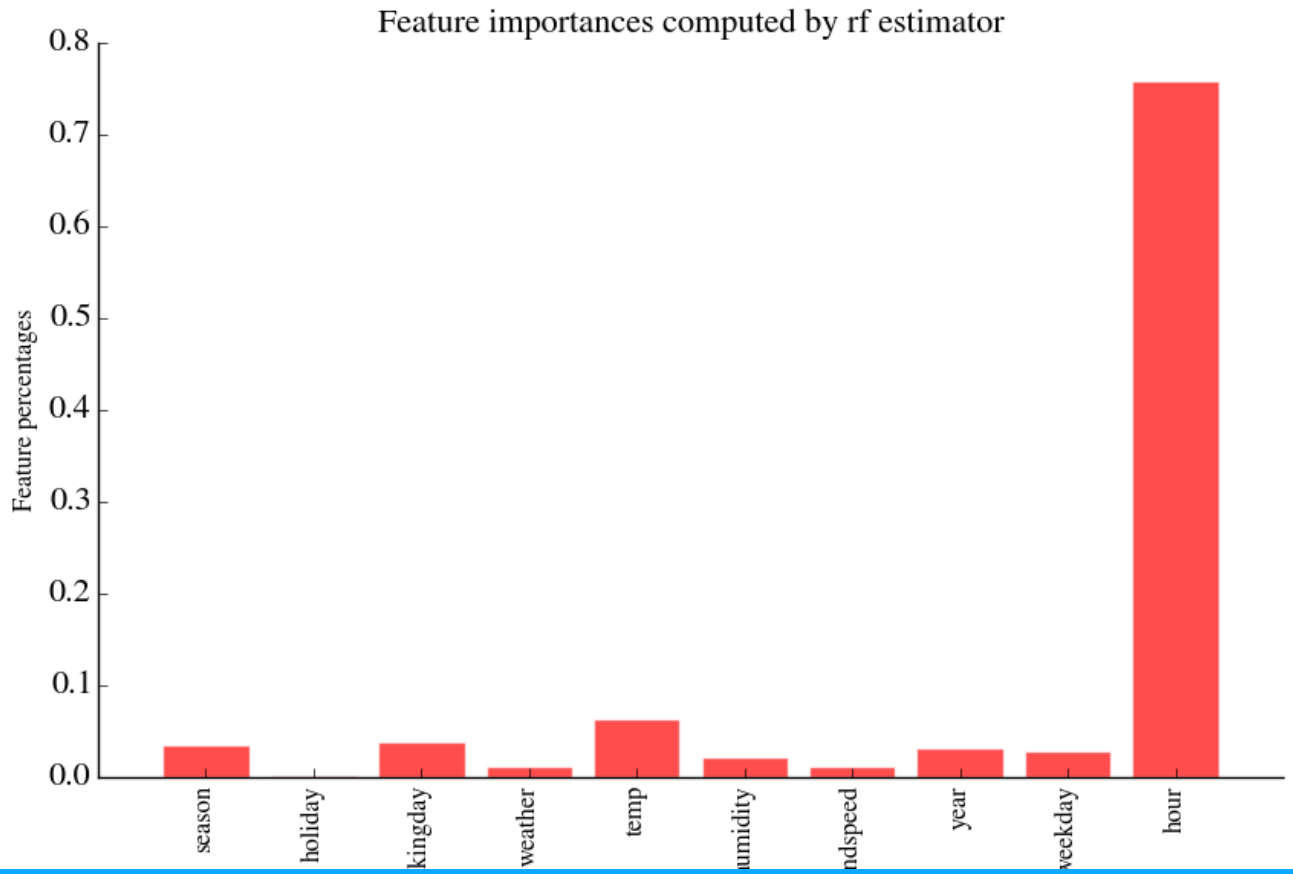
Random Forests

Idea

- *Create multiple weak learners for different subsets of the training set and combine them to form a strong learner*.
- Considering the abysmal performance of neuralnets and the fact that the problem is not amenable to them, we chose Random Forests.
- We were able to reduce to error to 0.39 using Random Forests.

Parameter Tuning

- We use scikit-learn's GridSearchCV to do an exhaustive search on the following parameters of GBM.
  - n_estimators
  - max_features
- The optimum parameters obtained were (100, 'auto')

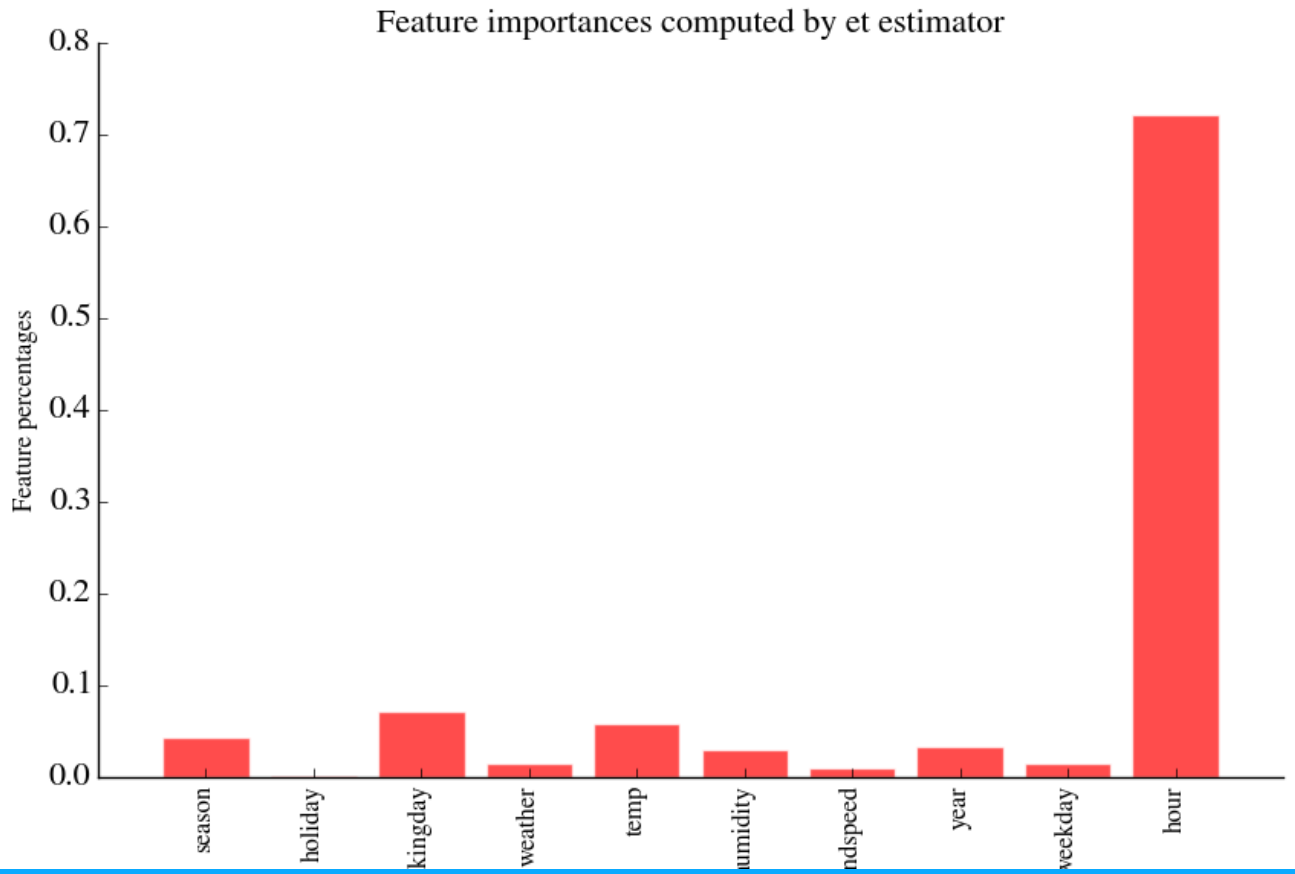


Feature Importance – Random Forest

Extra Trees Regressor

Idea

- A variant of Random Forests with random splits
- Increases randomness by choosing random splits.
- Didn’t fare much differently than Random Forests.



Feature Importance – Extra Trees Regressor

Gradient Boosting

Central Idea

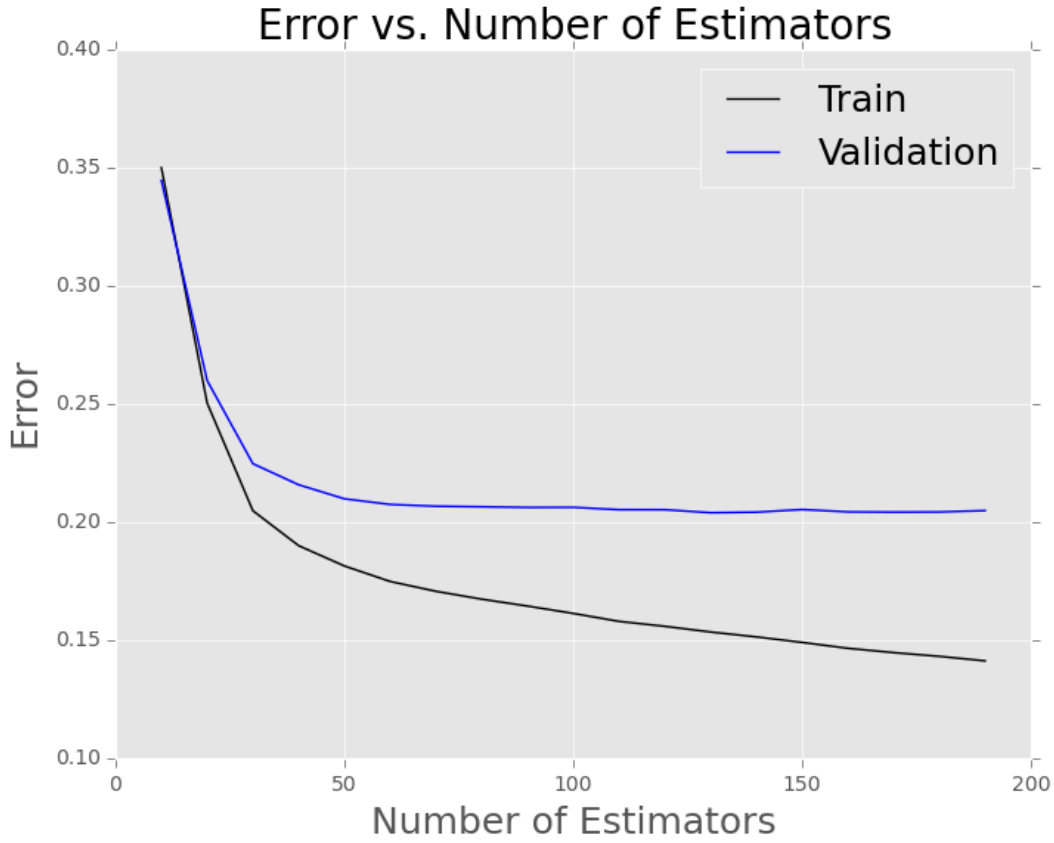
- Iteratively create a strong learner by using weak learners acting on error residuals.
- Gradient Boost and AdaBoost have been successfully applied to a number of problems and that motivated us to use them for this problem.
- Gradient Boost proved to be the best prediction model with an error of 0.375 on kaggle.

Parameter Tuning

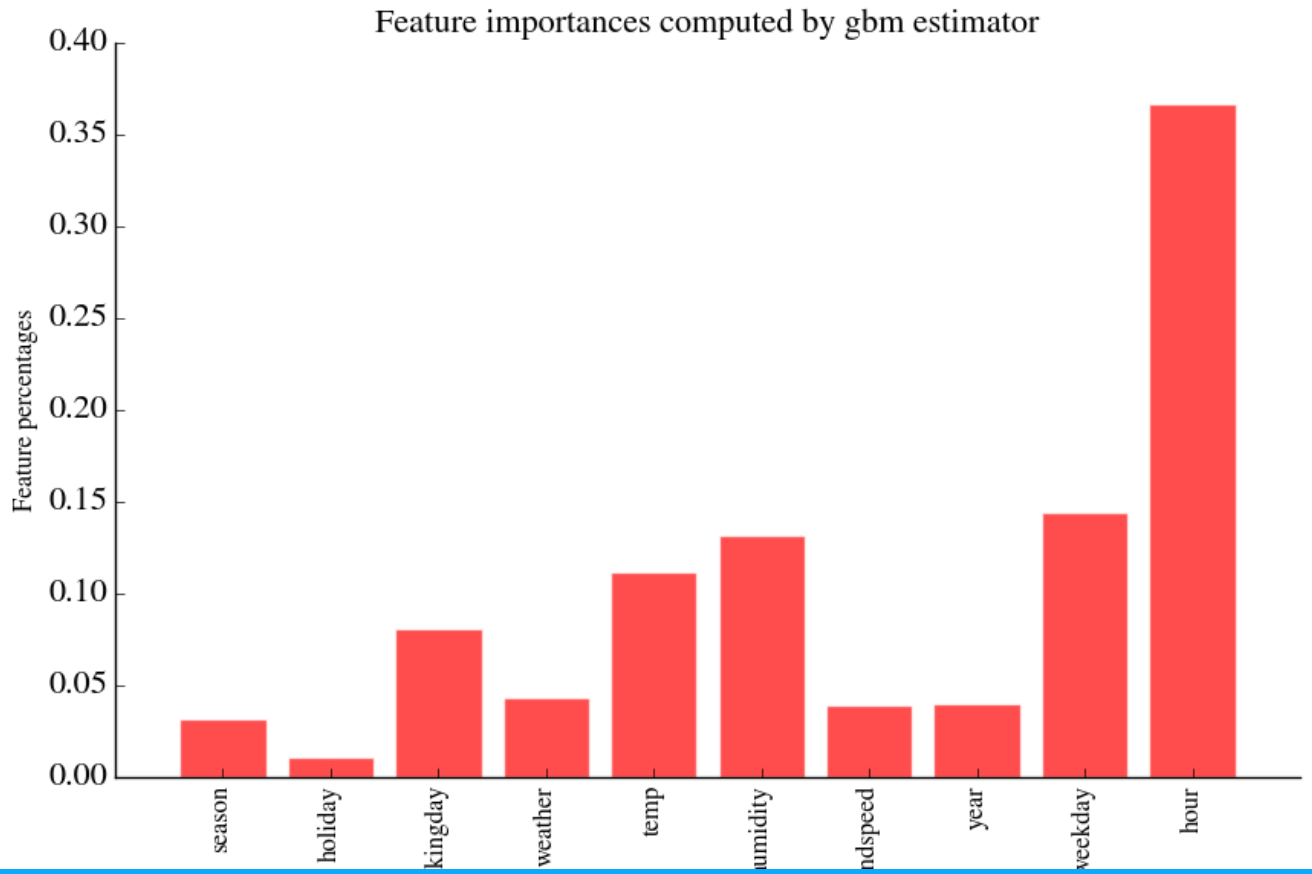
- We use scikit-learn's GridSearchCV to do an exhaustive search on the following parameters of GBM.
  - learning_rate
  - max_depth
  - min_samples_leaf
- The optimum parameters obtained (while keeping the no of estimators at 500) were (0.4, 3, 15)

Parameter Tuning

- Using the optimum values of learning_rate, max_depth and min_samples_leaf, we a curve for error vs no of estimators on a validation set to ascertain the optimum number of estimators.
- Our experiments found 85 estimators to be optimum.



Error vs No. of estimators



Feature Importance – Gradient Boosting

Variants

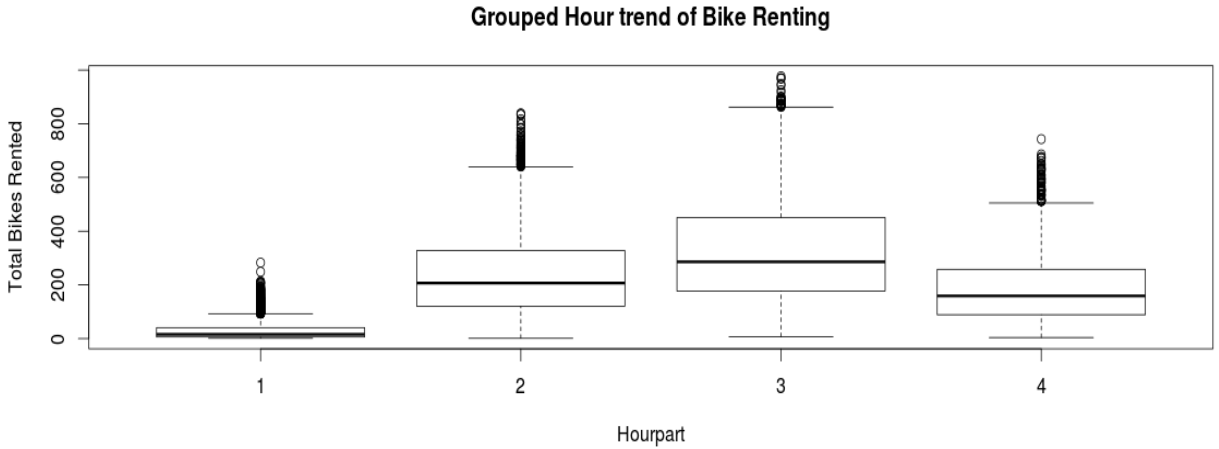
Linear Combinations

- Random Forest scored 0.397 on Kaggle while GBM scored 0.373 on Kaggle.
- We decided to use a linear combination of these two models weighted according to their performance on a validation set.
- RF got a weight of 0.4 while GBM got a weight of 0.6 in our experiments.

Discriminating Linear Combination

- We create separate GBM and RF estimators for casual and registered users.
- The results were comparable to that of the model without this distinction and hence we did not pursue this idea further.

Hour Slice Model



We grouped hours into 4 buckets: “Morning”, “Afternoon”, “Evening” and “Night”. The resulting model scored 0.72.

Results

Method	**Accuracy**
SVM	0.43
Random Forest	0.39
Neural Net	0.46
Gradient Boost Machine	0.37
Linear Discriminator	0.37
Extra Tree Regressor	0.39
Poisson Regression	0.42
Linear Combination of GBM and RF	**0.36**

Questions?