Final Report

Group 1: Jiawen Chen, Brooke Felsheim, Elena Kharitonova, Xinjie Qian, and Jairui Tang

4/29/2022

Introduction

As human populations are rising across the world, so is the proportion of people that live in urban areas. Estimates from the *UN World Urbanization Prospects* indicate that over 4.2 billion people (55% of the global population) currently live in urban areas, and by 2050, an additional 2.5 billion people (68% of the global population) could be living in urban areas[1]. More people living in urban areas calls for more space-, cost-, and energy-efficient systems of transportation as an alternative to cars. One such promising transportation alternative is the implementation of bicycle sharing programs.

Bicycle sharing programs are transportation schemes that allow individuals to rent bicycles on a short-term basis for either a set rate or for free. Most bicycle sharing programs have many computer-controlled bicycle rack "hubs" dispersed across a city that keep bikes locked and release them for use when a user enters the appropriate information/payment from a station or an app (Figure 1). A user can then ride the bike and return it to any other bicycle hub that is part of the same program. Many cities across the world have begun implementing bicycle sharing programs, including Chapel Hill, which has a Tar Heel Bikes sharing system[2]. Systems like these provide convenient, inexpensive, and eco-friendly transportation options for individuals residing in a city.

Figure 1: A 'hub' of bicycles belonging to the Santander Cycles system in London. SOPA Images/Lightrocket via Getty Images

Successful implementations of bike sharing programs depend on proper management of these systems. It is important for a bike sharing program to provide a stable supply of rental bikes to its population so its users feel that they can rely on the system for their transportation needs. The analysis of bike sharing data allows for a better understanding of the demand of rental bikes in a city, which, in turn, can help inform a city about how to provide appropriate supplies of rental bikes for its population.

[1] United Nations, Department of Economic and Social Affairs, Population Division (2018). World Urbanization Prospects: The 2018 Revision, Online Edition.

[2] https://move.unc.edu/bike/bikeshare/

For our project, we were interested in predicting the number of bikes rented within a given bike sharing system given information about weather, time of day, and date. We were also interested in assessing the most important variables for predicting bike rental counts. To answer these questions, we fit and evaluated a negative binomial generalized mixed model, a conditional inference tree, and a random forest model, using data from three publicly available bike sharing demand datasets.

The first dataset we use is a London bike sharing demand dataset downloaded from Kaggle[3] and provided by Transport for London[4]. This dataset contains hourly bike rental count observations over two years, from Jan 04 2015 - Jan 03 2017. The first full consecutive year of data was used as the training set in the analysis, and the second full consecutive year of data was held out as a test set in the analysis.

The second dataset we use is a Seoul bike sharing demand dataset downloaded from the UCI Machine Learning Repository[5] and provided by the Seoul Metropolitan Government[6]. This dataset contains hourly bike rental counts over one year, from Dec 1 2017 - Nov 30 2018. This was used as an independent test set in the analysis.

The third dataset we use is a Washington, D.C. bike sharing demand dataset downloaded from Kaggle[7] and provided by Capital Bikeshare[8]. This dataset contains hourly bike rental counts over two years, from Jan 01 2011 - Dec 31 2012. This was used as an independent test set in the analysis.

Each dataset contained hourly observations of bike rental count data. To simplify our analysis, we chunked the hourly data into three time blocks: [0:00 - 8:00), [8:00 - 16:00), and [16:00 - 24:00). Additionally, because temperature and humidity can be correlated with time of day, we chose to use the maximum and minimum daily temperature and humidity measurements for each 8-hour data point.

We created an R package named `bikeSharing` that includes methods for training and evaluating the negative binomial glmm and random forest models, as well as the processed source data from all three sets.

The below code can be used to install the package and load the package library. The zipped package source data, `bikeSharing_1.0.0.tar.gz` can be found in the Github repository for our project[9].

```
if(!require("bikeSharing", quietly = TRUE))
  install.packages("package/bikeSharing_1.0.0.tar.gz", repos = NULL)
library(bikeSharing)
```

Once the package is loaded, the bike sharing data from all three sets becomes easily accessible through the variable names `london`, `seoul`, and `dc`. To directly load them into the R environment, one can simply run:

All three datasets contain bike rental count data as well as 11 additional weather-, time-, and date-related variables that were used as predictors in our models:

- Hour chunk (00:00 - 8:00, 8:00-16:00, 16:00-24:00)
- Weekend status (Yes/No)
- Holiday status (Yes/No)
- Season (Winter, Spring, Summer, Autumn)
- Minimum daily temperature (C)
- Maximum daily temperature (C)
- Minimum daily humidity (%)
- Maximum daily humidity (%)
- Wind speed (m/s)
- Presence of any rain or snow (Yes/No)
- Date (mm-dd)

[3] https://www.kaggle.com/datasets/hmavrodiev/london-bike-sharing-dataset
[4] https://cycling.data.tfl.gov.uk
[5] https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand
[6] https://data.seoul.go.kr
[7] https://www.kaggle.com/datasets/marklvl/bike-sharing-dataset
[8] https://ride.capitalbikeshare.com/system-data
[9] https://github.com/brookefelsheim/bios735-group1

The way that these variables are used within our models will be further described in the Methods section. For all of the analyses performed, the `london` dataset was divided into training and testing sets, where the training set contained all "Year 1" data (Jan 04 2015 - Jan 03 2016), and the testing set contained all "Year 2" data (Jan 04 2016 - Jan 03 2017).

```
london_train <- london[london$Year == "Year 1",]
london_test <- london[london$Year == "Year 2",]
```

Methods

Negative Binomial Generalized Linear Mixed Model

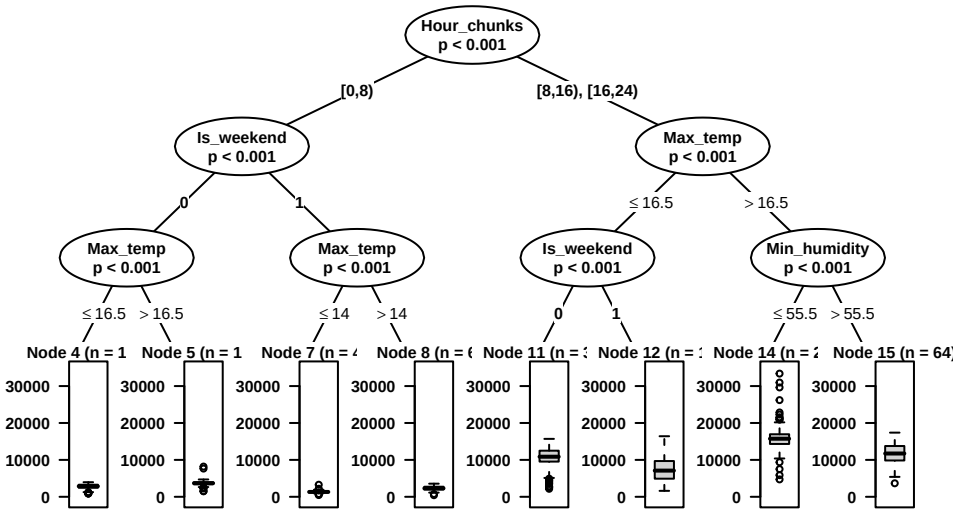
```
str(glmm_fit)
```

```
## List of 7
##  $ beta     : num [1:14] 8.353 1.534 1.415 -0.337 -0.393 ...
##  $ s2gamma  : num 0.0296
##  $ theta    : num 18.4
##  $ eps      : num 5.15e-05
##  $ qfunction: num -9520
##  $ day_ranef: num [1:365] 0.0653 -0.398 -0.5165 -0.2612 -0.0374 ...
##  $ iter     : num 23
```

Machine learning models

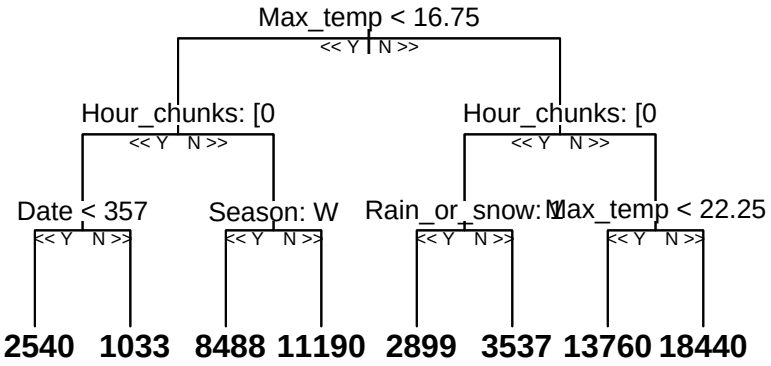
Next, we applied two machine learning methods to predict the bike count. The first model we used is a conditional inference tree. In traditional partitioning, all possible splits are investigated to find the best split, which results in overfitting and selection. The conditional inference tree embeds the partition step with a permutation test, thereby enabling this method to be robust to covariates of different scales. Furthermore, it is capable of stopping when no significant correlation exists between the covariates and the response (Hothorn, Hornik, and Zeileis 2006). We emplyed the ctree funciton in the partykit to fit the conditional inference tree.

We visualized the three layer conditional inference tree in order to verify the relationship between bike count and variables. It is apparent that people tend to rent fewer bikes between midnight and 8 am. Maximum temperatures and humidity also affect bike rental rates. The splits that are employed in the conditional reference tree provide well reasoned explanations of the data structure, which means high accuracy predictions are made.



A random forest model was then applied to predict the number of bikes that would be rented. Random forest is composed of many decision trees as opposed to a single tree, resulting in a more accurate result. Incorporating the randomness allows random forest to protect against overfitting and can be applied more effectively to other data sets. The disadvantage of the random forest is its computational complexity. To fit the random forest, we created a method `train_random_forest()` within our `bikeSharing` R package that leverages the the `train()` function within the `caret` R package. The optimal tuning parameter mtry was determined using a 5-fold cross validation. We visualized a random selected tree.

```
rf_fit <- train_random_forest(data = london_train)
```



Random forest sample trees share similar splits with conditional inference trees. The estimated bike rental count is much higher in the hour chunks that are not 0-8am. Additionally, the estimated count is high when the temperature is more than 22.25 degrees.

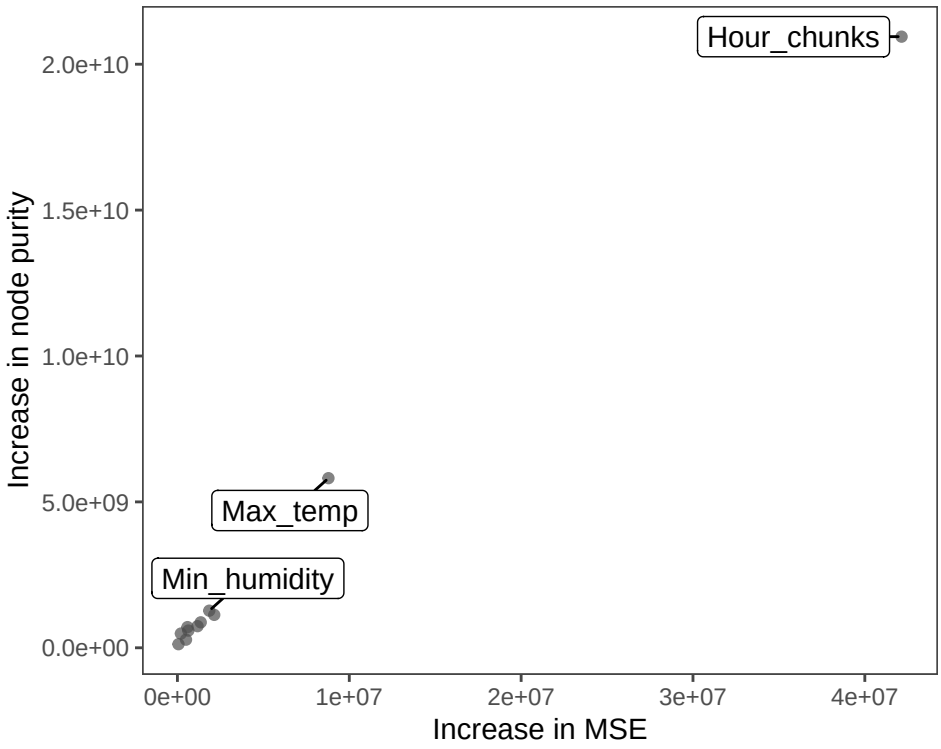
Variable Importance in machine learning models

To investigate which variables affect the prediction most, we calculate the importance of the variables in the conditional inference tree and the random forest. In conditional inference tree, we calculated the mean decrease in accuracy when deleting a variable.

Variable	Mean decrease in accuracy
Hour_chunks	42292743.475
Max_temp	7305786.627
Is_weekend	2819028.041
Rain_or_snow	1622451.011
Min_humidity	1014266.239
Season	905987.474
Max_humidity	184868.121
Wind_speed	35027.139
Is_holiday	25815.318
Date	3737.395

In random forest, we calculate the increase in MSE when deleting a variable. The code to do this was implemented as a function `plot_rf_importance` in the `bikeSharing` R package.

```
plot_rf_importance(london_train)
```



The top important variables of the conditional inference tree and random forest are similar. The hour chunk is the most important variable in both models. Additionally, the maximum temperature has a significant impact on the model prediction. Additional details of the variables are discussed in the discussion section. We further compare the performance of random forest and conditional inference tree using the second year bike renting data in London. Conditional inference tree results in a $R^2 = 0.81$ and random forest has $R^2 = 0.91$. Due the similarity in these two models, we selected random forest tree in the further comparison.

Results

```
glmm_model_fit(glmm_fit, london_train, scale_to_reference_mean = "no",
               reference = london)
```

```
##       RMSE      MAE        R2
## 1 1886.267 1291.106 0.8831618
```

```
glmm_model_fit(glmm_fit, london_test, scale_to_reference_mean = "no",
               reference = london)
```

```
##       RMSE      MAE        R2
## 1 2491.293 1647.064 0.8142036
```

```
glmm_model_fit(glmm_fit, dc, scale_to_reference_mean = "yes",
               reference = london)
```

```
##      RMSE     MAE       R2
## 1 845.741 605.217 0.521788
```

```
glmm_model_fit(glmm_fit, seoul, scale_to_reference_mean = "yes",
               reference = london)
```

```
##       RMSE      MAE        R2
## 1 3519.413 2719.021 0.4999935
```

```
rf_model_fit(rf_fit, london_train, scale_to_reference_mean = "no",
             reference = london)
```

```
##       RMSE      MAE        R2
## 1 752.2336 471.5412 0.9822717
```

```
rf_model_fit(rf_fit, london_test, scale_to_reference_mean = "no",
             reference = london)
```

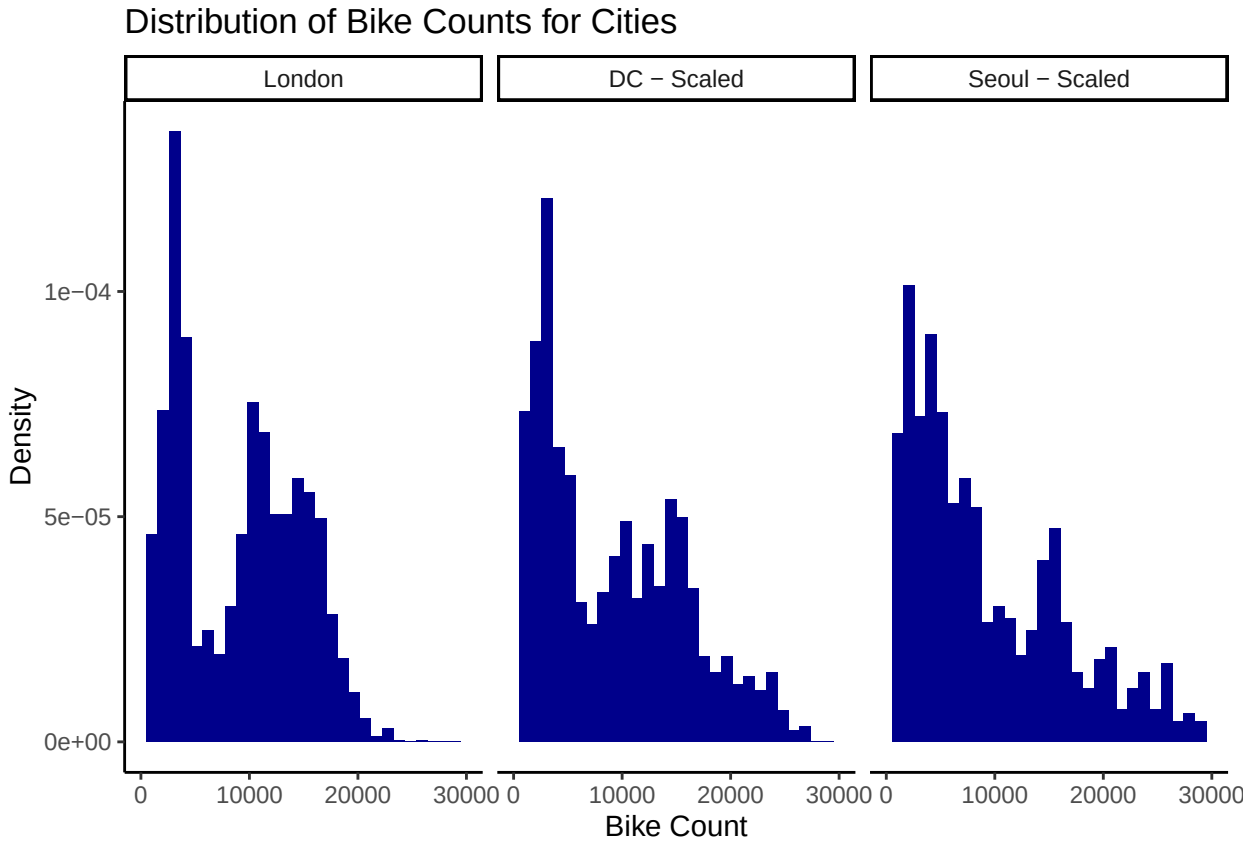
```
##      RMSE     MAE        R2
## 1 1789.24 1191.32 0.9084533
```

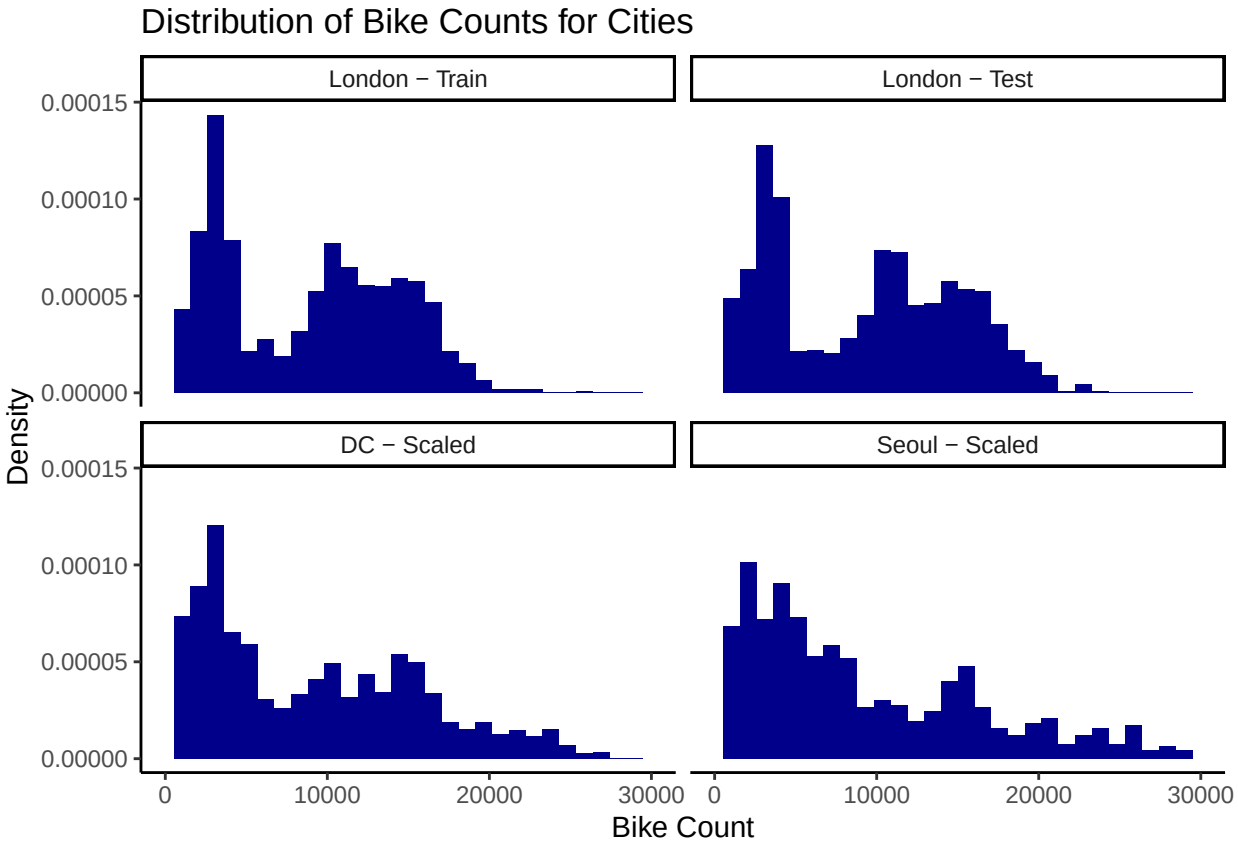
```
rf_model_fit(rf_fit, dc, scale_to_reference_mean = "yes",
             reference = london)
```

```
##       RMSE      MAE        R2
## 1 664.5474 492.4647 0.6737107
```

```
rf_model_fit(rf_fit, seoul, scale_to_reference_mean = "yes",
             reference = london)
```

```
##       RMSE      MAE        R2
## 1 3163.613 2479.835 0.5658297
```






```
## [1] 9286.037
```

```
## [1] 8913.796
```

Discussion

Reference

Hothorn T, Hornik K, Zeileis A (2006). "Unbiased Recursive Partitioning: A Conditional Inference Framework." Journal of Computational and Graphical Statistics, 15(3), 651–674. doi:10.1198/106186006X133933.