Bicycle Ridesharing

Sonya Gomez Enriquez   Mack Gregory   Joshua Guillen

Bryce Joseph-Nelson   Malini Manimaran   Devon Mirsalis

Charles Nguyen   Gabriel Raulet   Albert Stanley   Jordan Stefani

May 23, 2021

1 Introduction

1.1 Meet the Team:

- Devon Mirsalis
- Mack Gregory
- Bryce Joseph-Nelson
- Charles Nguyen
- Gabriel Raulet
- Albert Stanley
- Sonya Gomez
- Malini Manimaran
- Jordan Stefani
- Joshua Guillen

1.2 Project Description:

We plan to use the Seoul Bike Sharing Dataset form the UCI archive.

In our project, we hope to develop a regression model which will allow us to accurately predict how many bikes should be available to be rented out given information about the weather and day. In order to do this, we will first figure out which attributes we will drop from the Seoul Bike Sharing dataset through set of preliminary data analysis. Then we will find and create the best regression model which accurately captures the data. Finally, we will create a website in which a user will be able to enter weather and date information in order to recieve the best prediction of how many bikes will have to be prepared for renting.

2 Literature Review

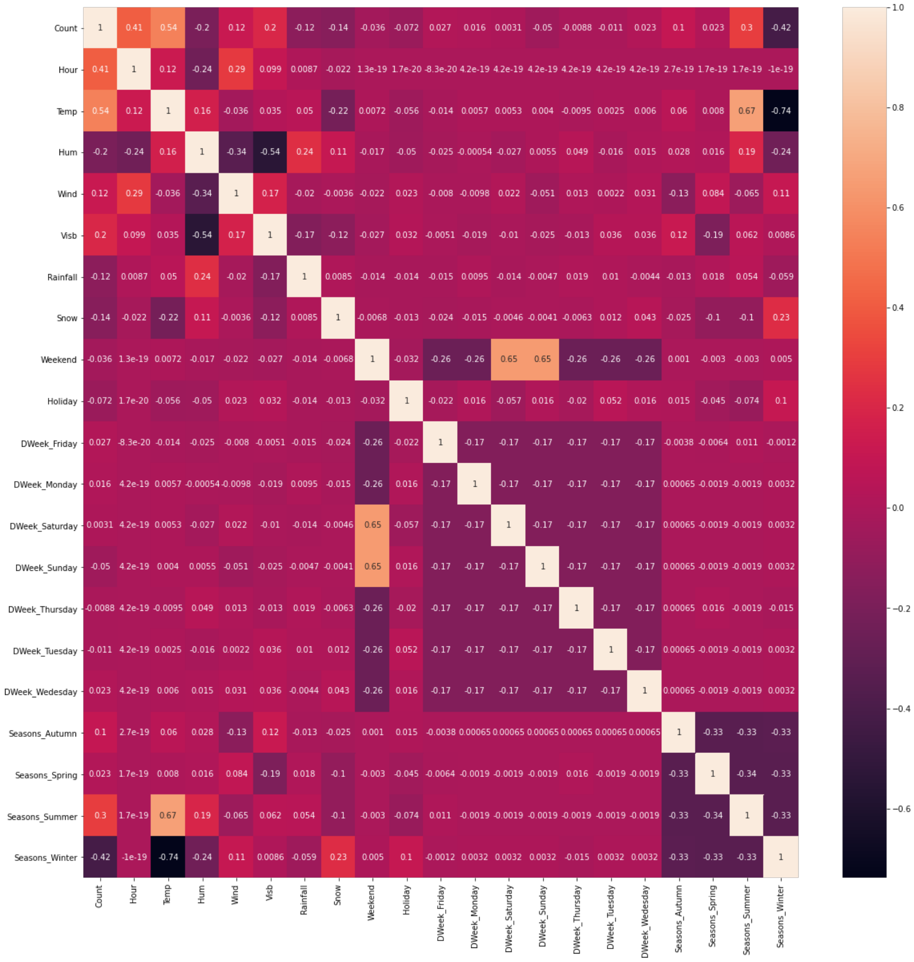
Bike sharing is an up-and-coming transportation method that can be implemented in cities and other regions to offer users the ability to transport themselves quickly where they want to go without the costs and concerns of personal bike ownership as well as helping to reduce the impact of the city on the environment. A problem arises with these systems when the operators need to determine how to best balance the availability of bicycles against monetary losses from bicycle systems with excessive availability. The dataset[**Irvine**] has already been examined in two papers[**datamining**][**rulebased**] that we examined so that we would not merely reproduce their models, but be able to learn from their methodology and add to their conclusions by looking at other types of models.

In ”A rule-based model for Seoul Bike sharing demand prediction using weather data” [**rulebased**], the authors collected the data from several APIs and merged them together to get a more complete view of biking activity. After this paper, the authors shared the dataset that they collected with us through the UC Irvine machine learning datasets portal[**Irvine**]. The authors tried five approaches: CUBIST, regularized random forests, classification and regression trees and K Nearest Neigbors to understand the dataset that they gathered. Of the five methods that they used to attempt to model the dataset, CUBIST showed the most promising results with an R^2 of 0.95 on the testing dataset, as well as lower root mean square errors, mean absolute error, and coefficent of variation than the other four methods presented. In "Using data mining techniques for bike sharing demand prediction in metropolitan city" [**datamining**] the authors used the same dataset with all of its added features to determine what was the smallest subset of the data that they could use and with which models could it be used. In the article, the authors explored linear regression, gradient boosting machines, support vector machines, boosted trees, and extreme gradient boosted trees. They found with all of the variables they were able to achieve an R^2 of 0.92 on the testing dataset using gradient boosted machines and not using data about several of the days of the week as well as snow. The paper found that weather and categorical data helps greatly to ensure model accuracy finding that removing those can reduce the testing R^2 by 0.27 or 0.15.

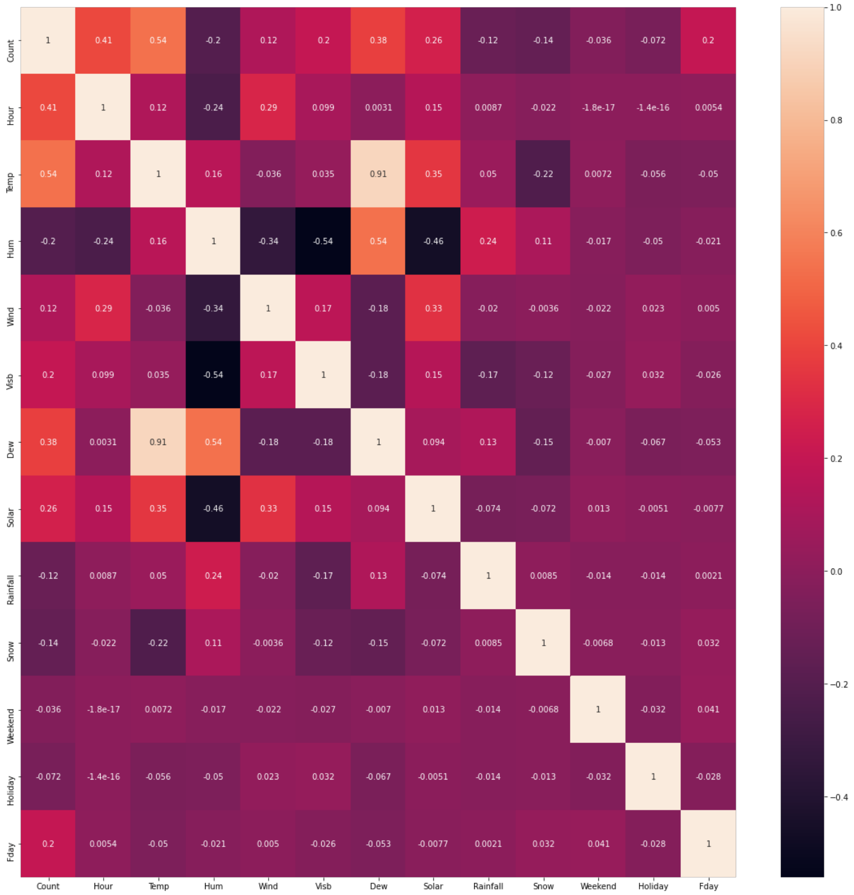
3 Dataset Introduction

For our project, we plan to use the Seoul Bike Sharing Demand dataset found on UCI Machine Learning Repository. Within this multivariate dataset, there are 8760 instances and 14 attributes. These attributes are either integer or string values and there are no missing values. In this section, we will go over a description of our dataset as well as the findings of our initial data analysis.

Our dataset consists of the following 14 attributes: Date, Rented Bike Count, Hour, Temperature, Humidity, Wind Speed, Visibility, Dew Point Temperature, Solar Radiation, Rainfall, Snowfall, Seasons, Holiday and Functioning Day. Out of these 14 attributes, these 4 attributes are categorical: Data, Seasons, Holiday and Functioning Day. As a consequence, during our data analysis, we will convert them into flag variables for ease of analysis. After doing so, we have created a heatmap to show positive and negative correlations between these variables.



From this heatmap, we can see that there are many positively correlated pairs of attributes:summer and temperature, weekend and saturday, weekend and sunday, bike count and temperature. Additionally, there are many negatively correlated pairs of attributes: winter and bike count, winter and temperature, visibility and humidity. Due to these correlations, we decided to combine day attributes into weekday or weekend attributes to create a second heatmap as this will more clearly show which attributes should be dropped due to high levels of correlations.



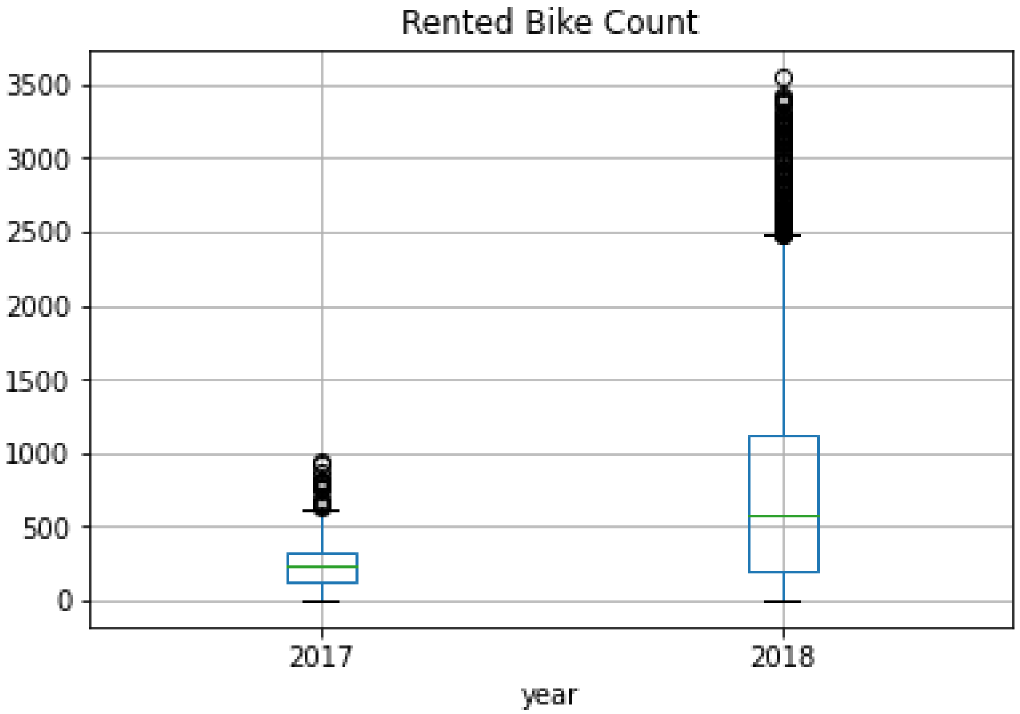
With this new heatmap, we see that temperature and dewpoint have a very high positive correlation of 0.91; therefore, we will drop the Dew Point Temperature attribute from our dataset. In addition, after discussing how we want our model to work, we have decided to drop the Solar Radiation attribute (this will be discussed in the next section).

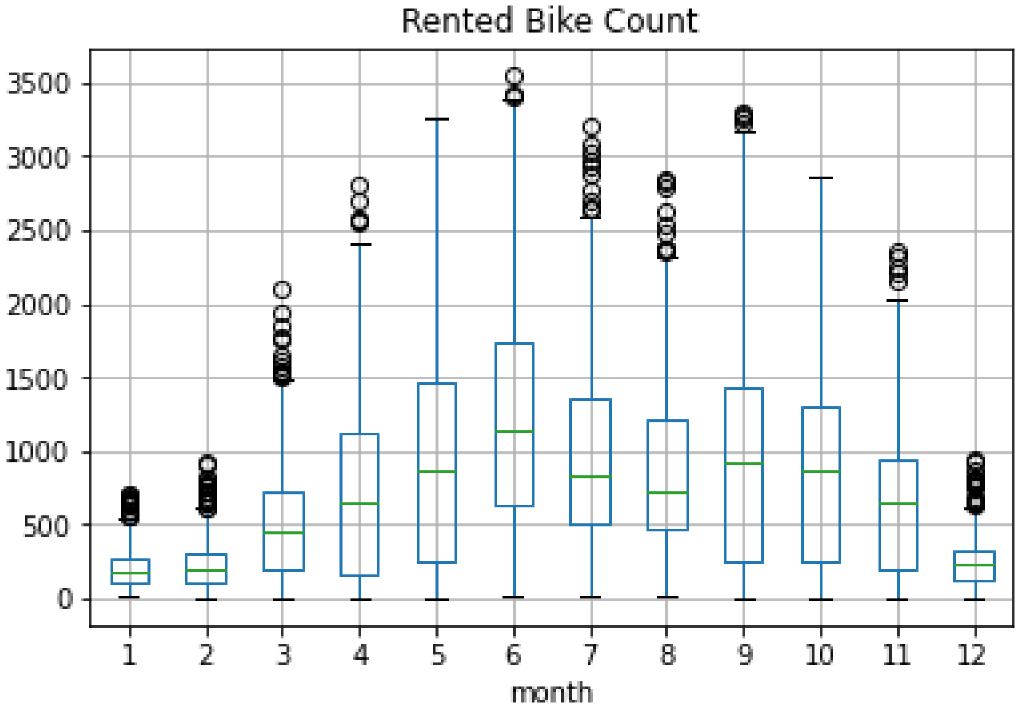
4 Project Goals and Effect on Chosen Attributes

Our goal for this project was to take this dataset to create a model which will most accurately predict the number of bike rentals needed when the user suggests a day. This means that we will create a regression model with the single target attribute of Rented Bike Count. In hopes of creating the most comprehensive model, the user will be able to provide the date, wind speed, temperature and other dependent variables to get a predicted bike count. Consequently, the user will not likely hold data on what Solar Radiation level they would like to predict bike count for. This means that it is in our interest to take this attribute out of our dataset and model since there will be no use for this attribute and we lower the risk of this attribute negatively affecting or skewing our model's prediction accuracy.

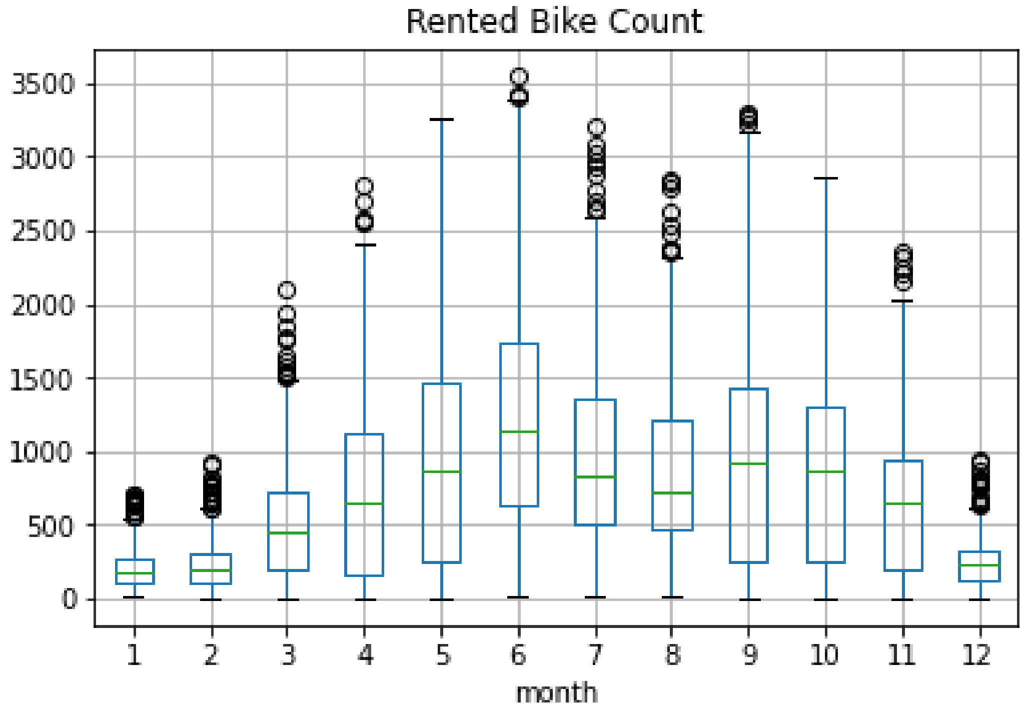
5 Data Exploration

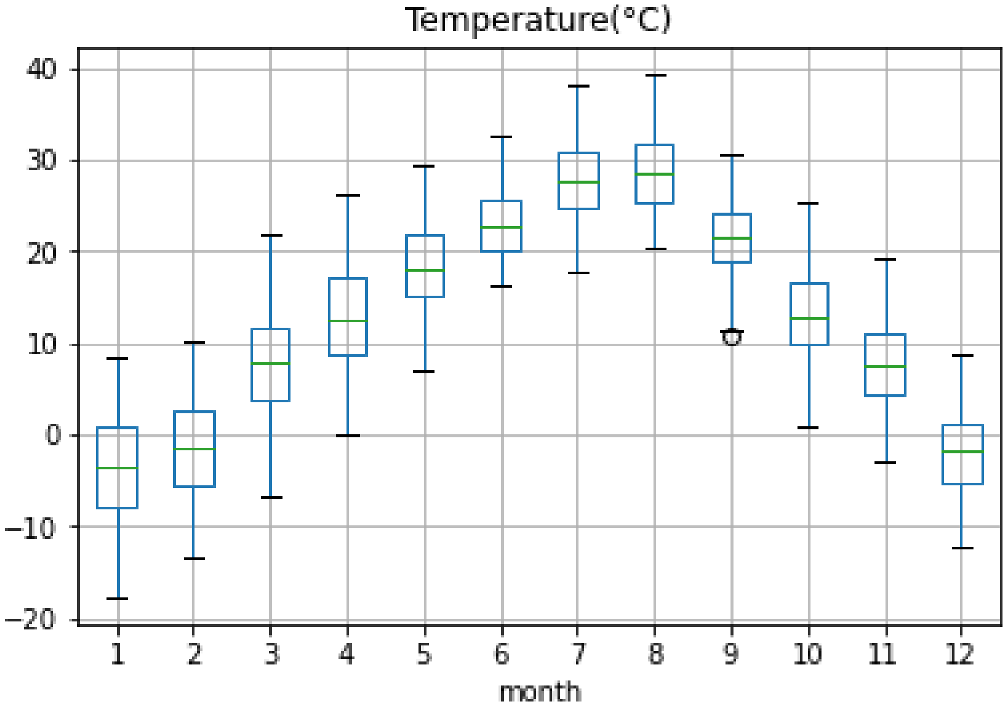
- The largest effect of the data restrictions is seen in out year vs bike count graph. For instance, our data has information from both 2017 and 2018 and the results appear somewhat inconsistent. This data occupies a period of 1 year December 1, 2017 - November 30, 2018. This explains our year box plot that shows the 2017 and 2018 bike rental count. We see that the 2017 bike count average is very low compared to the 2018 bike count because we can only include the month of December in the 2017 data whereas we can include January through November in the 2018 data. This automatically skews the 2018 median as January through November holds the highest averages of bike rent counts and December has the third lowest median for bike rent count as seen by the following graphs.



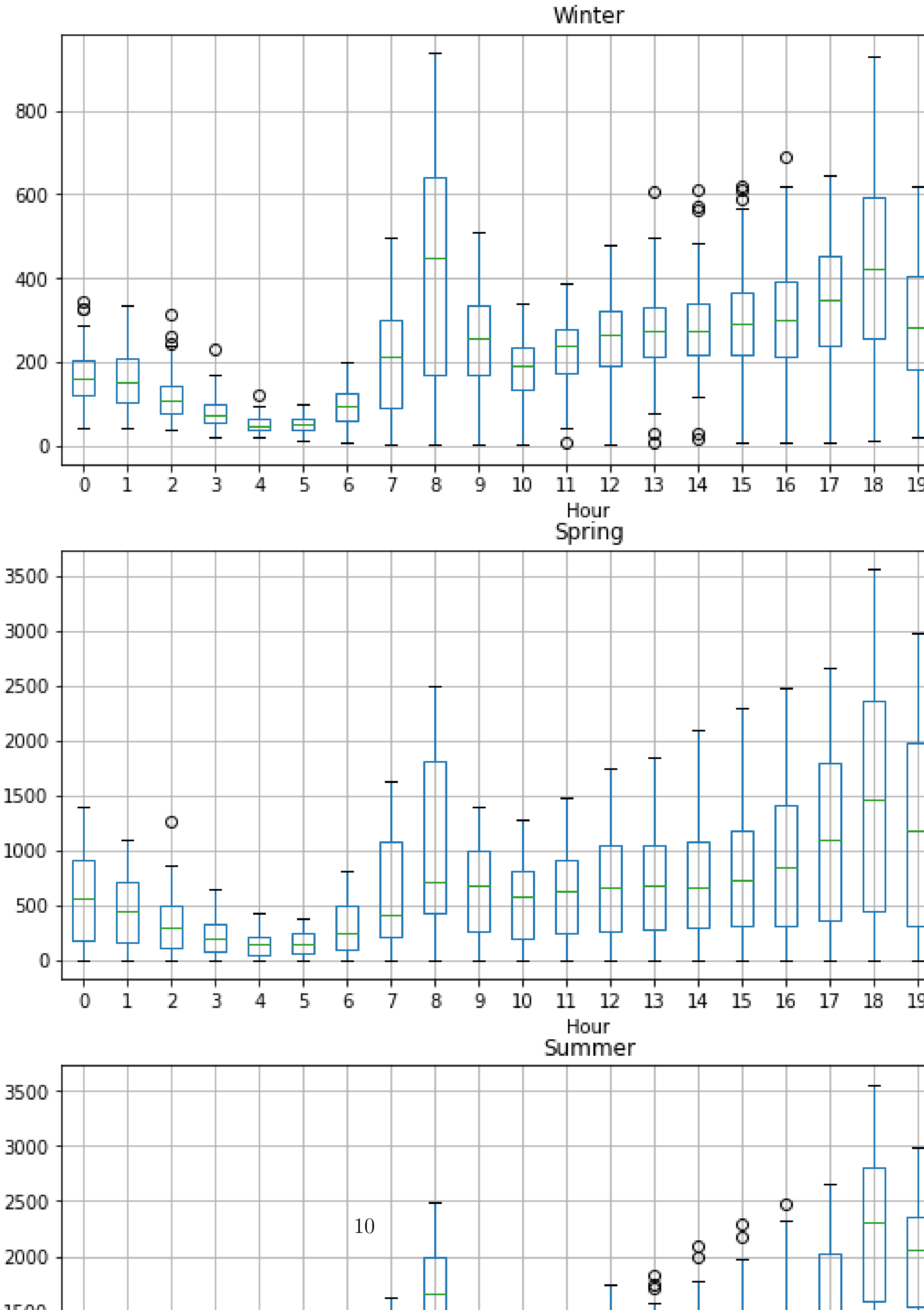


- Looking further into bike rent counts according to the month, we see that the median and interquartile range is highest from May through December; however we do see that there is a small dip in bike rentals from July to August. We attribute this to very high temperatures during the months of July and August deterring people from renting bikes.





- We also see the expected bike counts behavior per season. We expected that winter would have the lowest bike rentals due to the bad weather and summer would have the highest bike rentals due to a combination of good weather and students having time off of school. However spring and fall show reasonably high levels of bike counts, which are closer to the summer pattern compared to the winter patterns.


Winter
800
600
400
200
0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
Hour
Spring
3500
3000
2500
2000
1500
1000
500
0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
Hour
Summer
3500
3000
2500
2000
1500

- When looking at trends within the days, we see that during the weekdays, at 8AM there is a sharp increase in the number of bikes rented that we don't see on the weekends. A possible explanation for this is the use of bikes to travel to work.

  weekend and weekday.pdf weekend and weekday.pdf weekend and weekday.png weekend and weekda

- Here are some other graphs that show interesting patterns within the data.

  holiday.pdf holiday.pdf holiday.png holiday.png holiday.jpg holiday.jpg holiday.mps holiday.mps holi

  of week.pdf of week.pdf of week.png of week.png of week.jpg of week.jpg of week.mps of week.mps o

  .pdf .pdf .png .png .jpg .jpg .mps .mps .jpeg .jpeg .jbig2 .jbig2 .jb2 .jb2 .PDF .PDF .PNG .PNG .J

6 Proposed Solution

To find the most accurate prediction algorithm, we tried four different ones– Linear Regression, SVM(Support Vector Machines), Gradient Boosting, and Random Forest. We compared their accuracy using their R^2 Score. All the algorithms followed the same code structure.

1. Split the dataset 70:30 into a training and testing set.

2. Fit the training dataset onto the required algorithm using the sklearn library in Python. Evaluate with the testing set and take it's R^2 value for future comparison.

3. If applicable, re-fit the model after using grid search to find the best possible hyperparameters to increase model accuracy [R^2 value]. Train model on entire dataset and persist model for deployment on web application.

6.1 Algorithm Results

Model	R^2 value	Grid Search	R^2 value after Grid Search
SVM(Support Vector Machines)	0.799	Yes	0.898
Linear Regression	0.531	No	N/A
Gradient Boosting	0.835	Yes	0.928
Random Forest	0.908	Yes	0.91

6.2 Random Forests

Our most reliable and successful algorithm was the Random Forests algorithm[**Donges2019**]. The Random Forests algorithm works by creating a large number of decision trees and combining them into a "forest". The ending "forest" tends to be reliable since its decision trees don't allow overfitting because they rely on multiple features. The Random Forest algorithm was chosen to be the main algorithm in our web application because on average it had a higher R^2 value than the other 3 algorithms. The Random Forest algorithm's final R^2 value after grid search was 0.91, with its pre-grid search R^2 value being 0.908.

6.3 Gradient Boosting

Our second most successful algorithm, which you can also try on the web application and that should have fairly similar results was the gradient boosting algorithm[**Singh2018**]. Like the Random Forests algorithm, the gradient boosting algorithm works by making decision trees. But unlike the random forests algorithm, this one works by improving on a single tree by updating the weights for a number of iterations. The final R^2 value was 0.928 with it's pre-grid search score being 0.835.

6.4 Linear Regression and SVM

Our Linear Regression model[**StatisticsSolutions**] was the least successful. Linear Regression works by fitting an equation in the form of $Y = b + cx$ if simple or $Y = b + \sum_{i=0}^{n} c_i x^i$

if more complex. The data is fit onto the equation by updating the weights (c_i coefficients and the bias b). Linear Regression has a R^2 value of 0.531– proving to be less accurate than the other algorithms. SVM[**Gandhi2018**] is similar to Linear Regression, but in this case, the algorithm adjusts the hyperplane according to the data points closest to the hyperplane. The SVM algorithm has a R^2 value of 0.799 and 0.898 after grid search.

6.5 User Interface

To create the most optimal web app to support our needs for this project, we looked for one that was Python-based and could easily support proper data visualization along with easy HTML/text implementation for the sections that relied mostly on sharing data and parts of the report. We did not want to expend unnecessary energy on a complex web app. We ended up using "Streamlit" to launch our web app. Streamlit runs on Python code and takes care of most of the frontend, so we only had to code the actual Machine Learning application, the Report section, About Us section, and Data Visualization section.

We created a sidebar on the web app that holds our navigation menu. The Navigation menu has a few options, "Home", "Data Visualization", "Report", and "About Us". The Home page is where our best performing machine learning application will be run. The variables required to run the algorithm will be input in the sidebar and the algorithm results are presented on the left. The code of the algorithm is explained in the Proposed Solution and Results section. The Data Visualization is a very thorough report of our Data Exploration and general visualization of the dataset so we know what to expect. The Report section just showcases our report. The About Us section tells about the team members and all of our roles and contributions.

7 Discussion and Conclusion

In the pursuit of building an accurate model, we did not encounter many problems due to the fact that regression problems are fairly straightforward. Because of this, we were able to experiment with many different algorithms and find out which would most accurately fit our data. We found out that Random Forests was the most accurate and reliable for our purposes. Because supervised learning algorithms were covered over the course of the class, we were able to implement a lot of material from class into our project. Our project allowed us all to understand the class material more in depth. We were able to use algorithms we learned in class (Linear Regression and SVM) while also learning of new algorithms based on topics covered during lectures (Random Forests and Gradient Boosting).

In future versions of the project, we would like to implement a feature where the user can choose their locations and the Web Application fetches the required information and inputs it into the model, then predicts it. We would also like to find a way to include the solar radiation into our model predictions because we had to remove it due to inability to find real time values of it.