Alotta Cyclists

Prediction of Bike Sharing Demands Report

Devroop Banerjee
Computer Science, Math
University of Victoria
Victoria, Canada
indroneil20@gmail.com

Yi Xing Hu
Computer Science
University of Victoria
Victoria, Canada
simonhyx@gmail.com

David Toole
Software Engineering
University of Victoria
Victoria, Canada
static@uvic.ca

Hunter Watson
Software Engineering
University of Victoria
Victoria, Canada
hwatson@uvic.ca

***Abstract*— Bike sharing is blooming in the recent years for several reasons such as its low cost, simplistic installation, and eco-friendliness. With an increase in bike rental demands, it is necessary to ensure sufficient storage for bikes along with its availability to both registered and casual users at times of high demand. Our project aims to predict the demand of bike sharing for a particular day and time, given the date and weather conditions. Since the aim of this project is to predict a continuous variable, namely the number of bike rentals, the primary method used is regression. Additional methods such as clustering, and decision trees are also used to obtain more accurate predictions.**

***Keyworks*— Bike-Sharing, Data Mining, Kagel, Linear Regression.**

I. INTRODUCTION

The primary objective of this project is to efficiently and accurately predict the total number of bike rentals for a particular day and time, given the date and weather conditions.

The bike sharing system is simple; users can rent a bike at any kiosk and return it at a different one. Bike sharing users are split into 2 groups. They can either be classified as registered or non-registered users (referred to as *casual* users). Registered users, as the name suggests, are users that are registered in the system, and would use the bike sharing system on a regular basis whereas casual users are ones who are not registered in the system and use the system as per their needs basis.

In our project, we use the bike-sharing dataset provided by Kaggle [5]. The dataset contains the date and weather information needed to predict the demand of bike-sharing within a city. Bike-sharing can be used to extrapolate mobility in a city and this concept can be extended to anticipate the distribution of people based on when and where users are traveling. This information can be used by businesses and organizations to determine popular areas within a city.

II. RELATED WORK

The dataset chosen for this project is the subject of a Kaggle competition. Many of the people who have worked on this subject had made their findings available. The leaderboard for this competition contains, as of 7th March 2018, 3251 entries, representing 3251 attempts to predict bike share use on the basis of weather, season, day of the week, hour of the day, and numerous over variables.

Among these is Brandon Harris, who is placed 946th in the Bike Share Kaggle competition at time of writing using an R library called 'party' to build conditional inference trees. He also discussed his choices about handling of the dataset, converting dates to days of the week, and dividing the day into 4 six-hour partitions instead of twenty-four-hour long partitions [1].

Aditya Sharma shared his own top fifth-percentile scoring R code through his GitHub account [2] and discussed it on a post on Analytics Vidhya [3]. In this

project, 3 different models were used: decision trees, conditional inference trees and a random forest. Out of these

three, the random forest seems to deliver the best results. There were a series of analyses done prior to the process of building the model. For each feature in the data set, a histogram was generated and used to visualize user preferences. It was observed that temperature is an important factor in the number of bike rentals [3].

Giot and Cherrier compared the performance of numerous regression analysis models in their somewhat exhaustive article on the subject, predicting bike-share demand up to one day ahead [4]. They used ridge regression aka Tikhonov regularization, AdaBoost regression, support vector regression, random forest regression, and gradient tree boosting regression in the prediction of bike rental use between 1 and 24 hours in advance. They found ridge regression to be the most accurate model, followed closely by AdaBoost regression. The other 3 regression models were not as promising, often providing inaccurate predictions.

III. DATA DESCRIPTION

For our project, we are using a dataset provided by Hadi Fanaee Tork from the company Capital Bikeshare. The dataset can be found on Kaggle [5]. The data was collected from Washington, D.C. As this data was used for a Kaggle competition, we are provided a training dataset as well as a testing dataset. The training set contains data from the first 19 days of each month, and the remaining days are used as the testing data. In the training data, there are 9 attributes and 3 labels for each data point. The attributes contain the date and weather information. The labels are the number of casual users, the number of registered users, and the total number of bike rentals. The total number of bike rentals is the sum of the number of casual and registered users. The testing dataset exclusively contains the attributes and not the labels. Without the labels, we are unable to measure the performance of the model. Therefore, we have split the training dataset into 2 sets: one used for training our model, and the other for testing. For the purpose of our project, we are only interested in predicting the total number of bike rentals.

The description of each attributes and labels are listed in the following table:

Header	Type Information	Description
Datetime	datetime	Hourly date with timestamp
Season	int (1-4)	Integers mapped to the four seasons
Holiday	int (0-1)	Whether the day is a holiday
Workingday	int (0-1)	Whether the day is neither a weekend nor holiday
Weather	int (1-4)	Weather generalization where 1 is clear and 4 is heavy rain.
Temp	float	Temperature in Celsius
Atemp	float	“Feels like” temperature in Celsius
Humidity	int	relative humidity
Windspeed	float	Wind speed
Casual	int	Number of non-registered user rentals initiated
Registered	int	Number of registered user rentals initiated
Count	int	Total number of rentals

Table 1. Data fields in Kaggle Bike Sharing Demand dataset

Kaggle also provides a test data set of the remaining days of the month, from the 20th onwards, of each month for the purpose of scoring. However, as the count, casual, and registered fields are missing from this data, we cannot use it for our own purposes. Instead we train our data on the first 8 of every 10 days in the Kaggle data training set, and validate on the 9th and 10th day of each ten days i.e:- training on the 1st-8th and 11th-18th of January 2011, while validating on the 9th, 10th, and 19th of January, as well as the 1st of February.

IV. Proposed Project

The purpose of this project is to predict the number of bike rentals for a particular day given the date and weather conditions, more specifically, the count field. We will be using the dataset provided by Kaggle to train and test our model.

V. Current Completed Tasks

A. Data analysis

The data contains both discrete and continuous attributes. The discrete attributes contain season (4 seasons), holiday (true or false), workday (true or false), and weather (4 types of weather). The continuous data contains the date, the measured temperature, the temperature that it feels like, humidity, and wind speed. We have decided to not include the discrete attribute *season*, and continuous attribute *date* from our model. Our argument is that the number of bike rentals depends heavily on the current environmental circumstances, and the time of the year is simply a classification of the same at a certain point in time. We are aware that the temperature and weather depend on the season. However, any observed effects of season on the number of bike rentals is due to the effects of temperature and weather of that season.

An approach to test the hypothesis mentioned above, is to train our model using only data from 3 seasons (such as summer, fall, winter), test the data on the remaining one, and compare the performances of training the model on a random split of the data. If the bike rental count truly has no dependency on the time of year, then the model is expected to perform equally well on all sets and produce the same results on every random split regardless of the time of the year.

B. Model Design and Implementation

The purpose of our algorithm is to be able to accurately predict the number of bike rentals for a particular day. Since we are predicting continuous data, a regression model would be the most suitable choice.

R_squared value was the main method for model evaluation. For some of our models such as regression trees and neural nets, there are hyper-parameters we needed to choose. To tune the hyper-parameter of our model, we did a 64-16-20 split of the data into training, validation and testing sets, respectively. Any model that did not require hyper-parameter tuning, sufficed with a 20-80 split into testing and training sets, respectively.

Our first, but naive, attempt was to use linear regression. The R_squared value for linear regression was 0.32 and had an RMSE of 146.6 on the testing data. Then, we attempted kernel ridge regression. Using kernel ridge regression with polynomial kernel with default hyper-parameters, the R_squared value on the testing data was 0.54. In attempts to improve the performance, we started tuning the hyper-parameters associated with this model. After testing the gamma and alpha values, we realised that they did not contribute to minimizing the error value. The degree of the polynomial had a significant influence on the model's performance. The default degree of the polynomial was 3. To tune the hyper-parameter of our model, we did a 64-16-20 split of the data into training, validation and testing sets, respectively.

C. Kernel Ridge Polynomial Model

The R_squared score was plotted against the degree of the polynomial shown in figure 1. The kernel ridge regression model, provided by sklearn, ran to produce runtime errors and yielded inaccurate results upon exceeding a polynomial degree of 3. Therefore, no further investigations were conducted using polynomial kernels. Other kernel functions were tested, and the R_squared values were 0.21, and 0 for rbf, and sigmoid, respectively.

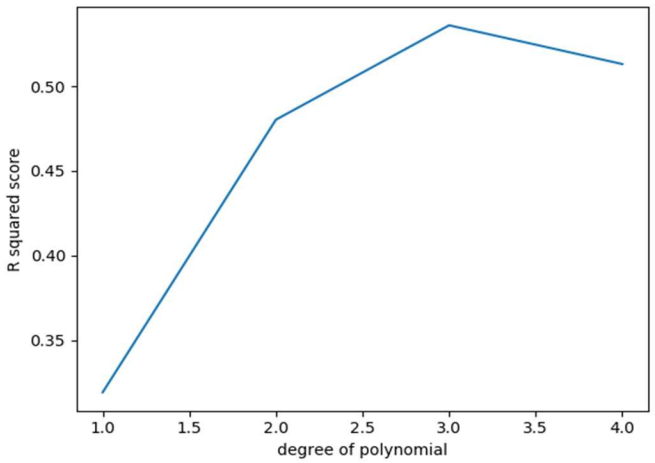

Fig. 1. Performance of Kernel Ridge Polynomial

D. MLP Regression Model

The next model we attempted was MLP regression. With max_iter set to 100000, and all other parameters set to default, the relative performance of each activation function was tested. It was observed that the R_squared value of relu was 0.45, logistic was 0.54, identity was 0.32, and tanh was 0.61. Since tanh performed significantly better than all the other activation functions, we decided to use tanh as the activation function for this model.

There were two hyper-parameters we had to tune, the number of hidden layer, and the size of the layer. We generated 9 plots where the number of hidden layers ranged from 1 to 9. For each plot, we plotted the R-squared value against the number of neurons at each layer. It was observed that the overall performance got better as the number hidden layers increased. After the number of hidden layers got to 6, there was no significant increase in the performance. Comparing each graph, we saw that the performance against the number of neurons was unpredictable. For our final model, we selected the model with a hidden layer of size 8 with 100 neurons in each layer. The reason we chose the model with 8 layers was because its performance appears to be less susceptible to changes in the number of neurons. For this model, the R-squared value and RMSE were 0.61 and 115.5 respectively.

E. Regression Trees

Afterwards, we attempted regression trees. In order to find the optimal height of the regression tree, R_squared was plotted against the maximum height of the regression tree. In figure 2 we see that the performance of the regression tree model has the best performance when the maximum height is set to 10. Therefore, our final regression tree model had a maximum height of 10. The R_squared value of the regression tree on the testing set was 0.79, with an RMSE of 85.2.

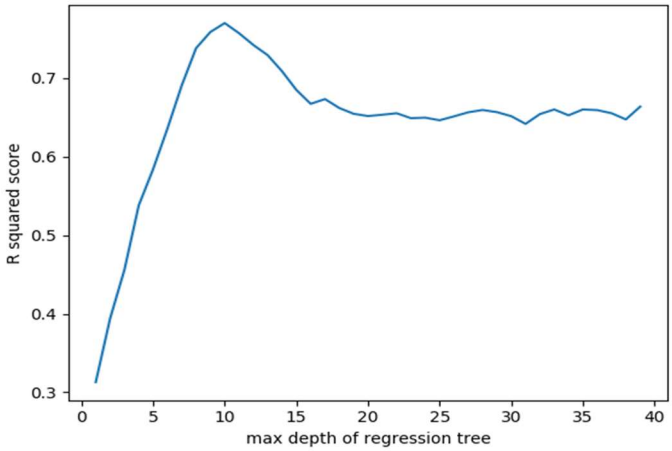

Fig. 2. Performance of Regression Tree

F. Regression Trees with adaBoost

To improve the performance of regression trees, we used adaBoost on regression trees. The R_squared value was plotted against the maximum height of the regression tree in Figure 3. From the figure, it is evident that the R_squared value showed no significant improvement when the maximum height was greater than 8. Therefore, we decided to use 8 as the maximum height of each regression tree. The model had a similar performance on the testing data as it did with the validation set. The R_squared value was recorded to be 0.83 on the testing set, with an RMSE of 77.1.

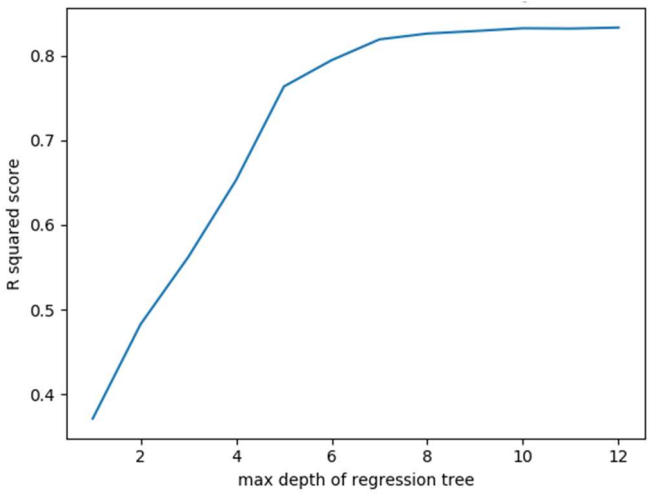

Fig. 3. Performance of adaBoost over all days

VI. SUMMARY

Out of all the models, regression trees with adaBoost has the best result, which was expected. Since our data contains both continuous and discrete attributes, a regression tree model is best suited since it is efficient at handling both discrete and continuous data simultaneously.

Problems arise when there are discrete attributes in the data. The kernel ridge regression with a polynomial kernel tries fitting a continuous function over an n-dimensional space. The following table summarizes the results obtained from all the models that we have used throughout the duration of our project:

Model	R squared value	RMSE	Parameters
Linear Regression	0.32	146.6	None
Kernel ridge regression (polynomial kernel)	0.54	122.5	Degree of polynomial: 3
Neural net MLP regression	0.61	115.5	Layers: 8 Neurons: 100
Regression Trees	0.79	85.2	Max height: 10
AdaBoost on regression trees	0.83	77.1	Number of estimators: 100 Max height of tree: 8

Table 2. Summary of the models used

VII. Future Work

One of the issues faced by our team was the amount of time it took for the various models to run. Python certainly made it easier to code and test our program but it was rather slow. In the future, we intend to alter our code to test on numerous other machine learning and/or numerical analysis software and applications such as R or SAS. We hope to obtain better results in terms of performance and accuracy.

Apart from applying our model to other software, we also intend to apply our model to other modes of transportation to gain a deeper understanding of the population density of a city. This would require additional datasets which is readily available on Kagel. These cases might give rise to several, unforeseen results which we will delve deeper into, to further our understanding of machine learning.

VIII. Ethical Implications

One of the biggest game changers in history and politics was the invention of the atom bomb, which started off with the discovery of nuclear fission and the potential to harness all the energy around us. By no means are we comparing our project to the discovery of nuclear fusion. However, it must be agreed upon that every scientific discovery/invention could be used for malicious purposes.

A statistical record of the daily usage of bikes (and other modes of transportation) in a certain city might actually have a positive outcome on the city's mindset towards global warming. If people get a better understanding of the number of bikes used on a daily basis and how it affects the CO2 emissions of that area, a trend might catch on, eventually leading to an eco-friendlier and healthier society. The same can be shown for the daily usage of other modes of transportation in order to inform the citizens about how much damage they are causing to the environment.

On a darker note, this program might turn out to be extremely useful to terrorists and/or extremists. This program essentially tells the user the population density of a city at different points of time in a day. It is not our intention to harm anyone, but this could raise some serious concerns if made publicly available.

IX. Timeline

Our proposed timeline for the project is visualized in the following Gantt chart:

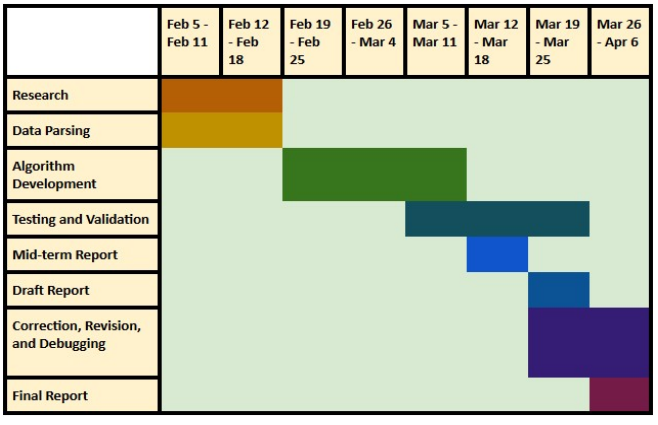

Fig. 4. Proposed timeline for project

X. TASK DISTRIBUTION

The preliminary breakdown of the responsibilities of each team member was as follows:

Name	Task
Devroop Banerjee	• Leading, Brainstorming • Research • Communication Management • Document Review
Yi Xing Hu	• Data Parsing • Algorithm Development • Model Design and Implementation • Testing and Validation
David Toole	• Research • Algorithm Development
Hunter Watson	• Document Review • Research • Algorithm Validation and Testing

Table 3. Proposed breakdown of tasks by group members

XI. REFERENCES

[1] B. Harris, "A simple model for Kaggle Bike Sharing", http://brandonharris.io/kaggle-bike-sharing/, accessed March 9, 2018.

[2] A. Sharma, "Bike Sharing Demand, Kaggle", https://github.com/adityashrm21/Bike-Sharing-Demand-Kaggle/blob/master/Bike_Sharing_Demand.R, accessed March 9, 2018.

[3] A. Sharma, "Kaggle Bike Sharing Demand Prediction – How I got in top 5 percentile of participants?", https://www.analyticsvidhya.com/blog/2015/06/solution-kaggle-competition-bike-sharing-demand/, accessed March 9, 2018

[4] R. Giot and R. Cherrier, "Predicting bikeshare system usage up to one day ahead", published in Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2014 IEEE Symposium on, http://ieeexplore.ieee.org/document/7009473/, accessed March 9, 2018.

[5] Kaggle, "Bike Sharing Demand", https://www.kaggle.com/c/bike-sharing-demand, accessed March 9, 2018.