Final Report

Group 13

2021/12/16

Introduction

Bike-Sharing systems are currently widely introduced in urban cities to solve the "last mile" problem, improve the link between other modes of transportation. Bike-sharing systems facilitate the use of public transportation, enhance traffic troubles, as well as minimize greenhouse gas emissions. However, the availability and accessibility of sharing bikes could be a problem since the demand and supply of bikes are not stable.

This project aims to use machine learning and data-mining-based algorithms to predict the demand for rental bikes in Seoul city at each hour in order to provide a stable bike supply. The project is based on the dataset downloaded at UC Irvine Machine Learning Repository. The dataset contains the hourly public rental number of Seoul bikes with date and weather information (Temperature, Humidity, Wind speed, Visibility, Dew point, Solar radiation, Snowfall, Rainfall) for one year from December 2017 to November 2018, 365 days. The number of rental bikes rented at every hour is determined from the bike rental history data collected from the Seoul Public Data Park website of South Korea.[1]

As all modes of transportation depend primarily on the weather conditions, including cycling, we would assume that the corresponding climate conditions have effects on the total number of rental bikes rented at each hour. And we would also assume that date parameters such as weekdays may enhance the performance of the prediction model. Thus, weather details and date parameters would be the covariates of the model.

Methodology

Linear Regression

Linear Regression Model (LM) is the simplest method to analyze the relationship of outcome Y and predictor X. The model is defined as below.

$$Y = \beta_0 + X^T \beta \tag{1}$$

In the regression, after we input the data of X and Y, the coefficients β_0 and β could be estimated by solving the equation.

Ridge

Ridge regression estimates coefficients of multiple linear regression with the aim for lower variance. By minimizing the shrinkage penalty, $\lambda \Sigma_{j=1}^{k} |\hat{\beta_j^2}|$, along with RSS, unnecessary estimated coefficients could be shrunk.

The coefficient λ controls the magnitude of the penalty. As it increase, coefficients tend towards 0, leading to low variance and low bias.

LASSO

Lasso regression is a type of linear regression similar to Ridge, minimizing shrinkage penalty $\lambda\Sigma_{j=1}^{k}|\hat{\beta}_j|$, but performs selection in addition to shrinkage.

As the coefficient λ increase, coefficients of LASSO model could be shurnk to absolute zero, resulting in removing variables.

Random Forest

Random forest generates a large number of bootstrapped decision trees, which performs binary splits on data at decision nodes, which are defined with a predictors' levels. At each node, features are selected from a random sub sample of all features in the dataset. The feature that could split the dataset in a way that minimizes RSS for the model is selected. This procedure increases the speed of model building as well as the variety in the use of features.

Out of Bag(OOB) score is used to validate random forest models. OOB samples are left-outs of each bootstrapping samples, used as a testset. OOB score is computed as the number of correctly predictions, while OOB error rate is the number of wrong classifications of OOBs.

Variables in the model is assessed by variable importance, which is penalized for large OOB error rate.

Bagging Forest

Bagging forest is similar to Random Forest, generating bootstrapped decision trees and using features that minimizes model RSS as the node to perform binary split. A fundamental difference between bagging forest and random forest is that at each node, all features in the dataset are compared against each other.

Therefore bagging forest is slower in model building speed and more restricted in the use of features in splitting. On the other hand, it could bring higher predictive accuracy, having better performance in minimizing MSE.

Out of Bag(OOB) score is also used to validate bagging forest models. Variables in the model is assessed by variable importance, which is penalized for large OOB error rate as well.

Support Vector Machine

Support Vector Machine(SVM) was designed for classification, and later the idea was generalized to regression. The optimization idea in SVM classification is to increase the amount of separation between two classes. However, in SVM regression, we would like to form a "band" around the true regression function that contains most of the points. With this purpose, the loss function is defined as below.

$$\mu(x) = \beta_0 + x^T\beta \tag{2}$$

If the point (X,Y) = (x,y) is such that $|y - \mu(x)| \leq \epsilon$, then the loss is taken to be zero; if $|y - \mu(x)| \geq \epsilon$, then the loss is taken to be $|y - \mu(x)| - \epsilon$. The main goal is to find a $\mu(x)$ such that most points with the loss taken to be zero.

Neural Network

Neural networks (NNs), also known as artificial neural networks (ANNs), are computing systems inspired by the way biological neural networks in the human brain process information. Building Neural networks is a widely adopted method in machine learning. The tasks to which neural networks are applied tend to fall within the following broad categories including regression analysis, classification, and data processing.

Neural networks provide the current best solutions in solving many computer science problems such as image recognition, speech recognition, and natural language processing.

In this project, we are going to solve a regression problem and make a prediction using neural networks. Instead of applying Convolutional neural network or Recurrent neural network, we adopt the most basic one: *Feedforward Neural Network* (FFNN). Feedforward neural network was the first type of artificial neural network devised. Different from recurrent neural networks, the connections between nodes do not form a circle. It has only one direction from input layer formed by input nodes, through hidden layers and to out the output layer formed by output nodes. Specifically, we will adopt *Multilayer Perceptron* (MLP). A multilayer perceptron consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. There are 8 hidden layers in our model.

The toolkit we used for building the neural network is *Keras*. Keras is a deep learning API (application programming interface) written in Python, running on top of the machine learning platform *TensorFlow*. The activation function we adopted are ReLU and Linear. Rectified Linear Unit (ReLU) is an activation function that defines the positive part of its arguement. It can be expressed as:

$$f(x) = x^+ = \max(0, x)$$

Linear is an activation function that simply returns the the argument itself. We also add regularizers that apply to a L1 and L2 penalty on the layer's output. L1 penalty is the penalty term in L1 regression or Lasso regression. L2 penalty is the penalty term in L2 regression or Ridge regression. Regularizers are used to avoid model overfitting and improve the robustness and generality.

Dataset description

Data description

The raw data is from the UCI machine learning repository which is a collection of 8,760 renting records from 2017 to 2018. There are 14 variables in the dataset. The table shows the features of each variable. The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information.

Exploratory data analysis

The data set included in the UCI we used has been processed, so there is no missing data. What we have done is to extract day, month, year from date, and convert date to the day of week. The final rental bike data is partitioned into two namely, training set for building the regression and testing set for assessing the model performance. In a random way, 20% of the 8760 records were selected as the test set and the other 80% as the data of the training set. The dimensions of training and testing set is shown in Table 1.

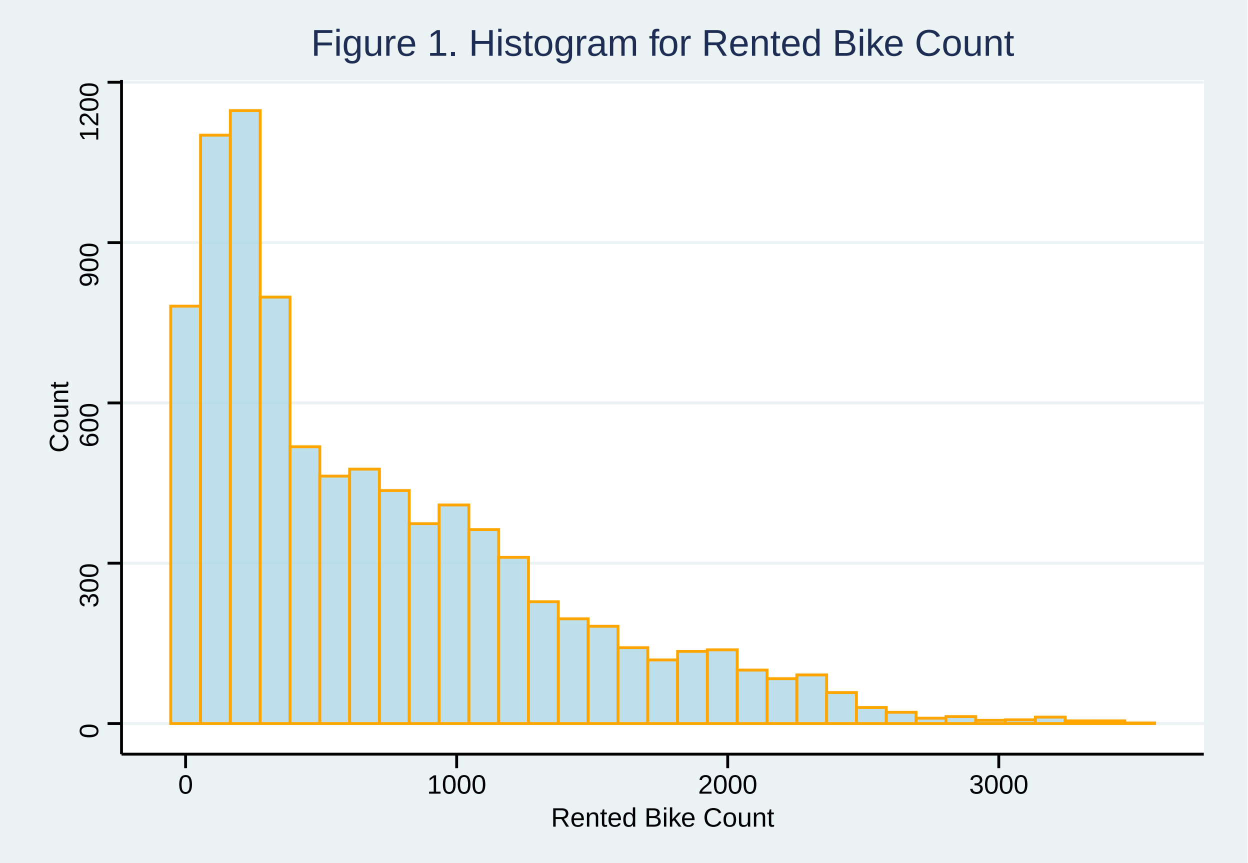
Table 1. Training and testing dataset

Dataset	Number of observations
Training set	7008 and 17 variables
Testing set	1752 and 17 variables

Figure 1 shows the total number of rented bikes for the entire period. It illustrates that the rental bike count is highly variable at each hour. As can be seen, there is a long tail in the data distribution.

Figure 2 displays an average number of the day throughout the week. It is shown that the count distribution follows identical trends over the weekdays and different patterns over the weekends. Figure 3 shows that the average count is high at each hour in the summer and low in the winter. The count is quite similar during autumn and spring. The count abruptly increases in hours 8 and 18. This is because the hours from 8 AM and 6 PM is regarded as the peak hours, during which the usage of the rental bike is high in Seoul.

As shown in Figure 4, the corplot function creates a graph of a correlation matrix, coloring the regions according to the value correlation coefficients. A correlation value of 1 is considered as a total positive correlation, -1 is considered total negative correlation, and if 0 no correlation exists between the variable. Positive correlations are notable between count and hour, temp, wind, visb, dew and solar. There is a negative correlation between count and humidity, rain and snow respectively. These correlation values imply that the weather variables are related to the rental bike count at each hour.



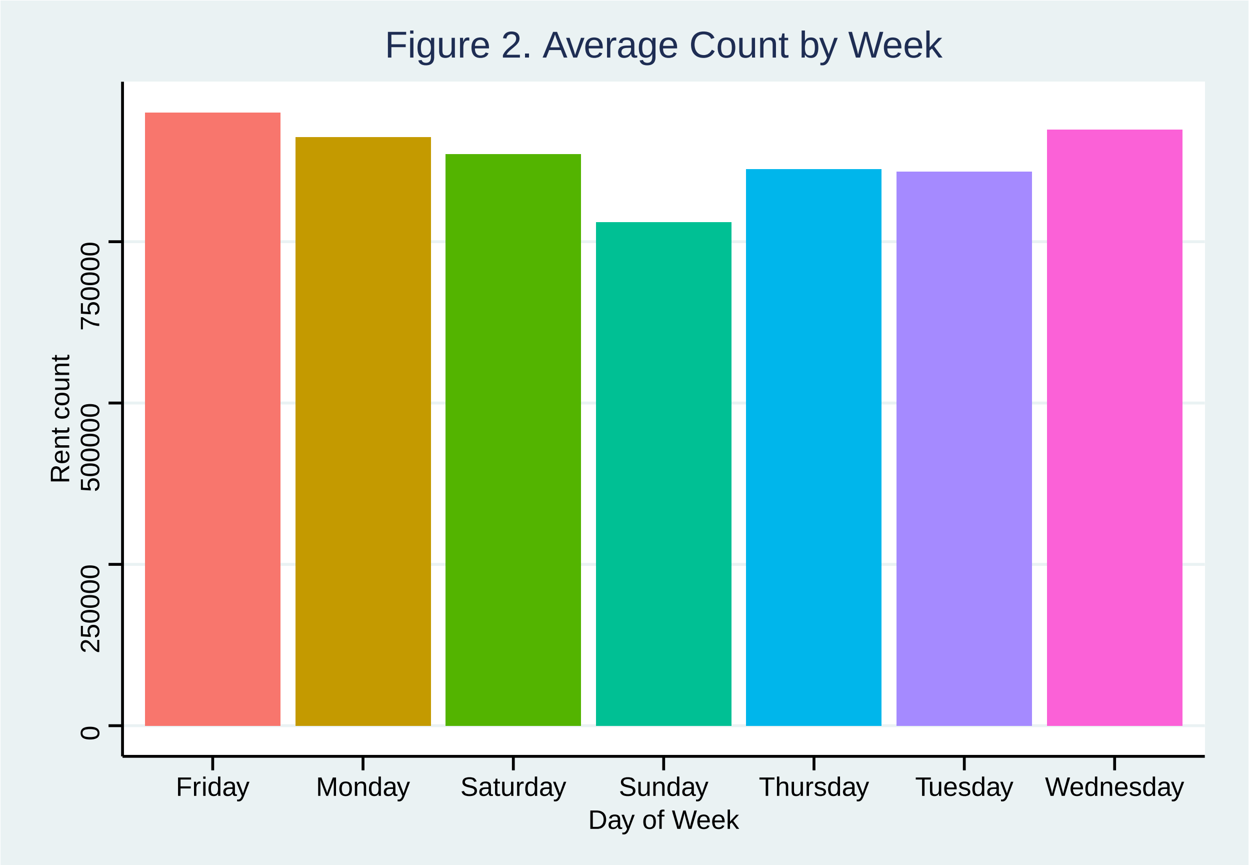


Figure 2. Average Count by Week

Figure 3. Average Count vs Hour grouped by Seasons

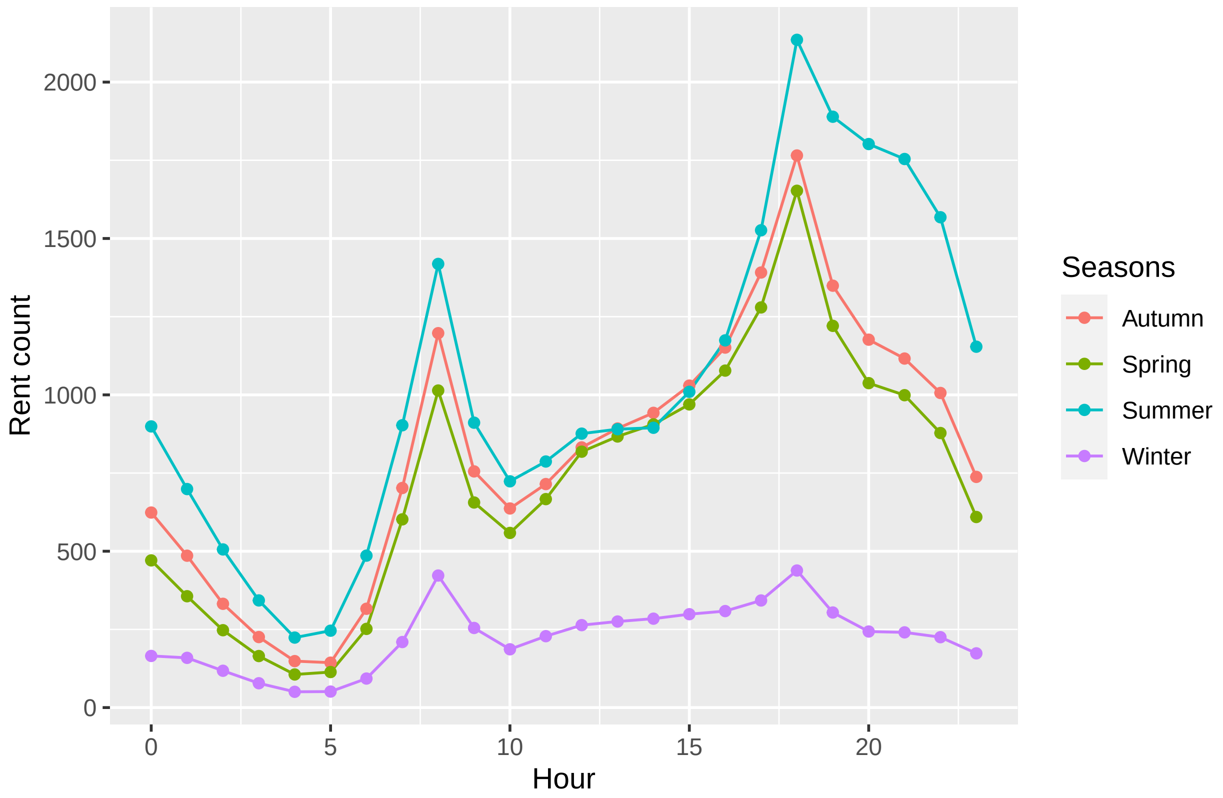
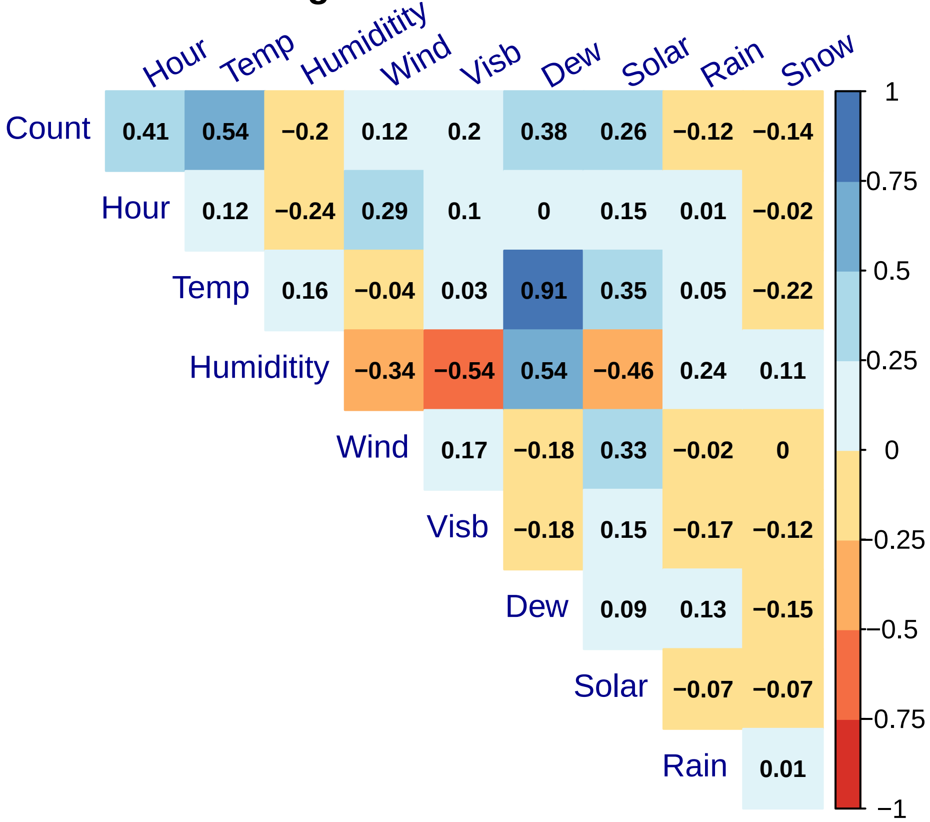


Figure 4. Correlation Plot



	Hour	Temp	Humiditity	Wind	Visb	Dew	Solar	Rain	Snow
Count	0.41	0.54	−0.2	0.12	0.2	0.38	0.26	−0.12	−0.14
Hour		0.12	−0.24	0.29	0.1	0	0.15	0.01	−0.02
Temp			0.16	−0.04	0.03	0.91	0.35	0.05	−0.22
Humiditity				−0.34	−0.54	0.54	−0.46	0.24	0.11
Wind					0.17	−0.18	0.33	−0.02	0
Visb						−0.18	0.15	−0.17	−0.12
Dew							0.09	0.13	−0.15
Solar								−0.07	−0.07
Rain									0.01

Evaluation index

The performance assessment index used here is Root Mean Squared Error (RMSE), which is the standard sample deviation between the observed and the predicted values of the residuals. RMSE measures the fluctuation of variance regarding different models. The better model is the model with lower RMSE. RMSE is defined by the equation as below.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{n}} \tag{3}$$

Model development

It is essential to tuning parameters in order to find the optimal model with relatively low loss or error values.

When building neural networks, we have tried different size of the hidden layers. Greater number of nodes in the layer indicates more parameters would be used for training and model fitting. We believe that it is more likely to fit a model with more parameters. Hence we add the hidden layers from 5, the starting model, to 8, the final optimal model. We also try different number of nodes ranging from 1 to 512 in different layers. There are more nodes in the first few hidden layers, and the nodes number keep reducing until the size of output (which is 1 since we only need to make a prediction of the number of rented bike).

However, large number of parameters may also bring the problem: overfit. To solve this problem, we constraint the complexity of the model. No more than 8 hidden layers and no more than 512 number of nodes in each layers. Typically, we adopt 256 or 64 nodes per layer. Futhermore, we introduce regularizers that add L1 or L2 penalty to the model during training base on the magnitude of the activation. We also try to randomly dropout a portion (i.e. 20%) of the input in specific layer while training. But later we find that the performance of these models are not as good as the one without dropping input.

Result

Method	Training	Testing
LM	407.0000	417.0000
Ridge	427.1653	437.7671
LASSO	427.3688	437.4730
RF	224.6018	222.9583
BagF	184.1149	182.7463
SVM	226.5303	221.5075
NN	120.0056	133.2593

Discussion

Ridge and Lasso are used when n is not much larger than p, there can be a lot of variability in the fit which can result in either overfitting and very poor predictive ability. Our trainset has 7008 observations of 17 variables, which implies we are in the case n»p. Therefore, we don't have the problem that Ridge and Lasso intended to solve. Furthermore, variable selection (like in Ridge and Lasso) is useful when some predictors are not significant or predictors are highly correlated. In our dataset, most predictors are highly significant, correlation between them is moderate and VIFs aren't exaggerate. Thus, other circumstances that could make variable selection useful don't hold here. This explains why Lasso and Ridge regression didn't perform well in our project.

Both bagging forest and random forest shows that Hour and Functioning.Day is significantly more important than other variables in predicting rental count. Bagging forest results in smaller RMSE than random forest, suggesting that since number of variables is not excessive, increasing variety during feature selection is not needed for this dataset.

Citation

E, S., Park, J., & Cho, Y. (2020). Using data mining techniques for bike sharing demand prediction in Metropolitan City. Computer Communications, 153, 353–366. https://doi.org/10.1016/j.comcom.2020.02.007