SEIS763 Final Project Report - Seoul Bike Rental

Using Machine Learning Techniques to Predict Bike Rentals in Seoul

Matt Adeola†
Graduate Programs in Software
University of St. Thomas
St. Paul, MN, USA
adeo9806@stthomas.edu

Shivali Dalmia
Graduate Programs in Software
University of St. Thomas
St. Paul, MN, USA
dalm3066@stthomas.edu

Sukaina Ali
Graduate Programs in Software
University of St. Thomas
St. Paul, MN, USA
sali@stthomas.edu

Sushant Khullar
Graduate Programs in Software
University of St. Thomas
St. Paul, MN, USA
skhullar@stthomas.edu

ABSTRACT

Rental bike sharing systems have been introduced in many urban areas to improve mobility and comfort, it also contributes to a greener environment. With the recent popularity of the system, it is important to make the rental bikes available on time to lessen the waiting time of people. The purpose of this study was to create an analytical model, using machine learning methods and techniques, which would closely predict the number of bikes rented out in the city of Seoul, South Korea. To provide a deeper and more thoughtful understanding of the predicted numbers, the study also observed predicted bikes rented based on various features within the data, like quarters (four quarters within the year), seasons, holidays, temperature, rush hour, etc. As the target variable was continuous, regression models like, linear regression, polynomial regression (with ridge, lasso, elastic net), decision trees, and random forests were implemented. The Root Mean Square Error (RMSE - square root of the variance of the residuals) metric was used as a measure of how accurately the model predicts the response. It is the most important criterion for fit, if the main purpose of the model is prediction. Lower values of RMSE indicate better fit. By comparing the RMSE values of different models pre and post feature elimination, the random forest regression model on raw clean dataset was selected as the best performing model.

INTRODUCTION TO THE DATASET

The Seoul bike rental dataset, downloaded from a public data portal for all Seoul citizens [3], contained 8,760 instances with fourteen features. The raw data provided a total number of bikes which were rented out in the city of Seoul, South Korea, on an hourly basis each day for 12 months. The data spanned from December 1, 2017, through November 30, 2018. The dataset contained nine continuous attributes including the target variable - Rented Bike Count (*Dependent Variable*), Temperature(C), Humidity (%), Wind Speed(m/s), Visibility, Dew point temperature(C), Solar Radiation (MJ/m2), Rainfall(mm) and Snowfall(cm)). It also included four categorical variables - Hour, Seasons, Holiday, Functioning Day. The only series data was Date which was formatted in dd/mm/yyyy format.

HYPOTHESES QUESTIONS

In addition to the overall prediction for the number of bikes rented, the study also put forth several hypotheses and questions which would help in further defining the patterns of bike rentals based on various factors. Below are statements and questions that they were explored:

1. Summer season has the most bike rentals.
2. Rush hour has the most bike rentals.
3. No Holiday days have the most bike rentals.
4. Number of bikes rented in each quarter.
5. Extreme temperature [6] led to low bike rentals.
6. Rainy days with low visibility led to low bike rentals.
7. A snowy day with low visibility led to low least bike rentals.

DATA EXPLORATION

During the data exploration phase, no missing values were found within the dataset for any categorical or continuous features. The study also checked for class-imbalance - to assess if machine learning techniques like oversampling using random selection, Synthetic Augmentation techniques such as Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic (ADASYN) or under sampling, should be applied to the dataset. The dataset was equally distributed for the most of the features so the oversampling techniques were not implemented.

A total of 6.2million bikes were rented out between December 2017 and November 2018. There were 8,760 total instances in the dataset, however when filtering out non-functional days, the total number of instances went down to 8,465. This included 353 days of the twelve months captured in the dataset.

A five-number summary, including mean, standard deviation, minimum, maximum, and quartiles, was used to provide a quick snapshot for all the continuous variables in the dataset. The data showed that the average number of bikes rented on “no holiday” (715) days were significantly higher compared to those rented on a “holiday” (500). The maximum number of

bikes were rented out at 1800hrs followed by 1900hrs and 1700hrs (evening rush hour). 0000hrs to 0600hrs had the least number of bikes rented out. For different quarters of the year, quarter2 and quarter3 had the maximum bike rentals followed by quarter4 and quarter1. All the top 30 instances of bikes rented out belonged in the 1800hrs group. For seasons, summer had the highest number of bikes rented out followed by autumn and spring. Number of bikes rented in winter was significantly lower compared to other seasons. Exploration was also conducted using various weather-related features such as temperature, humidity, wind speed, solar radiation, rainfall, and snowfall. The average number of bikes rented out gradually increases with the observed temperature (in degrees Celsius). However, the number starts to decline when the temperature goes above 34C. For these continuous variables, new categories were created for data exploration purposes. The average number of bikes rented during calm wind instances were higher compared to breezy and windy instances. Also, a significantly higher number of bikes rented out when the visibility was excellent compared to good and low - which garnered the least number of bike rentals. For the Functioning Day feature, only days marked as No had no bike rentals.

DATA TRANSFORMATION

The information gathered during the data exploration phase was used to clean and transform the raw dataset. A new binary feature - "Rush Hour'', was created using the "hour" variable. Rush hour was defined as those between 0600hrs and 0800hrs, and 1700hrs to 1900hrs. It was noted that the average number of bikes rented out during rush hours (957) were higher compared to non-rush hours (620). The hour feature had a count of hours from 0000hrs to 2300hrs. For ease of implementing one hot encoding, this feature was also used to create "Request Time Slot ''. Which included six-hour intervals, thereby including four time slots. The third time slot (1300hrs - 1800hrs) had the highest number of bikes rented out followed by the fourth slot (1900hrs - 2400hrs), second time slot (0600hrs - 1200hrs) and first time slot (0000hrs - 0500hrs).

As there were no bike rentals on those days marked as "No" within the Functioning Day variables, these instances were deleted from any further analysis. The Hour and Date variables were also deleted as they had been captured in new variables that were created (e.g., Request Time Slot, Rush Hour, Month). The Holiday feature and the newly created Rush Hour variable were transformed into binary format using label encoder technique. A "1" represented true and a "0" represented false in each column. For ease of applying machine learning methods and techniques, the target variable, Rented Bike Count, was moved to be the last column of the table.

Before performing any predictions, the dataset was broken into two data frames, X, and y, where X included all columns except the target variable and y contained only the dependent variable, Rented Bike Count.

ONE HOT ENCODING

One hot encoding was performed on the categorical variables, including Seasons, Quarter, Request Time Slot. Extra variable created by one-hot encoding for each variable was removed.

SPLITTING AND NORMALIZING

Once one-hot encoding was performed, the data was ready for splitting and normalizing to a standard scale. Using the scikit learn's [1] *train_test_split*, the data was split into seventy percent training and thirty percent test data. The *X_train*, *X_test*, *y_train*, *y_test* variables were created, where *X_train* and *y_train* would be used to train machine learning models and check for error rates, and *X_test* and *y_test* would be used to test the performance of each model.

MACHINE LEARNING MODELS (Raw Dataset)

As Rented Bike Count was a continuous variable, various regression methods like, linear regression and polynomial regression (with ridge, lasso, elastic net), in addition to decision trees and random forests were implemented to make predictions. These models were applied to the transformed data set; however, no feature elimination was performed at this point. These methods explored significant relationships between the dependent variable (Rented Bikes Count) and other features within the dataset.

1. Linear Regression

The first model evaluated was linear regression. *X_train* and *y_train* were used to train the model. Linear regression produced a straight best fit line to explore relationships between the target variable and independent variables. To predict *y_pred* and calculate the Root Mean Squared Error (RMSE), *y_pred* was compared to *y_test* using *X_test* as new instances. Once applied, the RMSE for the linear regression model was calculated as 431.56.

2. Polynomial Regression

The second model evaluated was polynomial regression. *X_train* and *y_train* were used to train the model. Polynomial regression produces a curved best fit line to explore relationships between the target variable and independent variables. The degree for the polynomial features were set at 2. To predict *y_pred* and calculate the RMSE, *y_pred* was compared

to *y_test* using *X_test* as new instances. Once applied, the RMSE for the polynomial regression model was calculated as 362.95.

3. Ridge Regularized Regression

The next model evaluated was regularized regression with ridge. *X_train* and *y_train* were used to train the model. The degree for the polynomial features were set at 2. To predict *y_pred* and calculate the RMSE, *y_pred* was compared to *y_test* using *X_test* as new instances. Once applied, the RMSE for the regression model using ridge was calculated as 359.23.

4. Lasso Regularized Regression

The next model evaluated regularized regression with lasso. *X_train* and *y_train* were used to train the model. The degree for the polynomial features were set at 2. To predict *y_pred* and calculate the RMSE, *y_pred* was compared to *y_test* using *X_test* as new instances. Once applied, the RMSE for the regression model using lasso was calculated as 360.20.

5. Elastic Net Regularized Regression

The next model evaluated regularized regression with elastic net. *X_train* and *y_train* were used to train the model. The l1_ratio was set to 1 and the degree for the polynomial features were set at 2. To predict *y_pred* and calculate the RMSE, *y_pred* was compared to *y_test* using *X_test* as new instances. Once applied, the RMSE for the regression model using elastic net was calculated as 360.20.

6. Decision Tree Regression

The next model evaluated was a decision tree regression model. *X_train* and *y_train* were used to train the model. To predict *y_pred* and calculate the RMSE, *y_pred* was compared to *y_test* using *X_test* as new instances. Although, for each run the RMSE value differs by 1-3 units, still the best RMSE for the decision tree regression model was calculated as 443.95.

7. Random Forest Regression

The next model evaluated was a random forest regression model. *X_train* and *y_train* were used to train the model. In order to determine the best estimator value for n_estimators. RMSE plot was created using different values of n_estimators, starting with minimum value of 5 and maximum value of 150, incremented by 10. The curve was nearly flat for n_estimators value between 90 to 140. Therefore, *the n_estimator* that was selected was 135, for minimum stable RMSE value. To predict *y_pred* and calculate the RMSE, *y_pred* was compared to *y_test* using *X_test* as new instances. Once applied, the RMSE for the random forest regression model was calculated as 324.52.

8. Ensemble Model

Using the models mentioned above from 1 - 7, an ensemble model was implemented on the data. The RMSE for the ensemble model was 338.91.

SUMMARY OF RMSE's ON RAW DATASET

The summary of RMSE's of all implemented models is explained in Table1.

Table1.RMSE Summary of all regression models.

Model Name	RMSE
Multiple Regression	431.56
Polynomial Regression	362.95
Ridge Regularized Regression Model	359.23
Lasso Regularized Regression Model	360.20
Elastic Net Regularized Regression Model	360.20
Decision Tree Regression model	443.95
Random Forest Regression model	324.52
Ensemble Model	338.91

FEATURE ELIMINATION

The feature elimination technique[5] was employed in the study to remove weakest features based on their p-values or their significance in the prediction model. The study used backward elimination method to eliminate features with p-value greater than 0.05. The least significant features were removed from the transformed dataset.

MODELS POST FEATURE ELIMINATION

After feature elimination, the previous steps of train-test splitting, normalization and standardization, and implementation of regression models were conducted again. New test and train variables were created. Using the scikit learn *train_test_split* (70-30 split), the data was split in to *X_train_sig*, *X_test_sig*, *y_train_sig*, *y_test_sig*, where *X_train_sig* and *y_train_sig* would be used to train machine learning models and check for error rates, and *X_test_sig* and *y_test_sig* would be used to test the performance of each model.

1. Linear Regression

The first post feature elimination model to be evaluated was linear regression. *X_train_sig* and *y_train_sig* were used to train the model. To predict *y_pred* and calculate the RMSE, *y_pred* was compared to *y_test_sig* using *X_test_sig* as new instances. Once applied, the RMSE for the linear regression model was calculated as 431.44. This RMSE was very similar compared to the pre-feature elimination linear regression model at 431.44.

2. Polynomial Regression

The post feature elimination polynomial regression model used *X_train_sig* and *y_train_sig* to train the model. The degree for the polynomial features were set at 2. To predict *y_pred* and calculate the RMSE, *y_pred* was compared to *y_test_sig* using *X_test_sig* as new instances. Once applied, the RMSE for the post-feature elimination polynomial regression model was

372.24. This RMSE was higher compared to the pre-feature elimination polynomial regression model at 372.24.

3. Ridge Regularized Regression

The post feature elimination regularized regression with ridge used *X_train_sig* and *y_train_sig* to train the model. The degree for the polynomial features were set at 2. To predict *y_pred* and calculate the RMSE, *y_pred* was compared to *y_test_sig* using *X_test_sig* as new instances. The RMSE for the model was 369.04 which was higher compared to the pre-feature elimination regularized regression using ridge at 369.04.

4. Lasso Regularized Regression

The post feature elimination regularized regression with lasso used *X_train_sig* and *y_train_sig* to train the model. *X_train* and *y_train* were used to train the model. The degree for the polynomial features were set at 2. To predict *y_pred* and calculate the RMSE, *y_pred* was compared to *y_test_sig* using *X_test_sig* as new instances. Once applied, the RMSE for the regression model using lasso was 369.76, which was higher compared to the pre-feature elimination regularized regression using lasso at 369.76.

5. Elastic-Net Regularized Regression

The post feature elimination regularized regression with elastic net used *X_train_sig* and *y_train_sig* to train the model. The *l1_ratio* was set to 1 and the degree for the polynomial features were set at 2. To predict *y_pred* and calculate the RMSE, *y_pred* was compared to *y_test_sig* using *X_test_sig* as new instances. Once applied, the RMSE for the regression model using elastic net was 369.76, which was higher compared to the pre-feature elimination regularized regression using elastic net at 369.76.

6. Decision Tree Regression

The post feature elimination decision tree regression used *X_train_sig* and *y_train_sig* to train the model. To predict *y_pred* and calculate the RMSE, *y_pred* was compared to *y_test_sig* using *X_test_sig* as new instances. Once applied, the RMSE for the decision tree model was 443.97, which was higher compared to the pre-feature elimination model at 442.09.

7. Random Forest Regression

The post feature elimination random forest regression model used *X_train_sig* and *y_train_sig* to train the model. In order to determine the best estimator value for n_estimators. RMSE plot was created using different values of n_estimators, starting with minimum value of 5 and maximum value of 150, incremented by 10. To predict *y_pred* and calculate the RMSE, *y_pred* was compared to *y_test_sig* using *X_test_sig* as new instances. Once applied, the RMSE for the random forest regression model was 342.11, which was higher compared to the pre-feature elimination model at 341.01.

8. Ensemble Model

Using the models mentioned above from 1 - 7, an ensemble model was implemented on the post feature eliminated data. The RMSE for the ensemble model was 348.64.

SUMMARY OF RMSE's POST FEATURE ELIMINATION

The summary of RMSE's of all implemented models on feature eliminated dataset is explained in Table2.

Table2. RMSE Summary of all regression models.

Model Name	RMSE
Multiple Regression	431.44
Polynomial Regression	372.24
Ridge Regularized Regression Model	369.04
Lasso Regularized Regression Model	369.76
Elastic Net Regularized Regression Model	369.76
Decision Tree Regression model	442.09
Random Forest Regression model	342.01
Ensemble Model	348.64

FEATURE EXTRACTION

To reduce the dimensionality of data and extract important features from the dataset principal component analysis was done on the dataset which was obtained post feature elimination.

1. Principal Component Analysis (PCA).

The study also performed feature extraction using PCA machine learning technique[7]. Using this technique, the study strives to create thirteen new features which can help improve the prediction capabilities for the model. The study set *n_components* were set at 13 for prediction. New *X_train_pc* and *X_test_pc* were created using the *X_train_sig* and *X_test_sig*, respectively. Once created, linear regression model was fit to *X_train_pc* and *y_train_sig*. The RMSE for this linear regression model was 432.02.

2. Kernel Principal Component Analysis.

The study also used kernel PCA[4] (using rbf kernel) feature extraction technique and applied linear regression to the newly created features. The study set *n_components* were set at 14 for prediction. New *X_train_pc* and *X_test_pc* were created using the *X_train_sig* and *X_test_sig*, respectively. Once created, linear regression model was fit to *X_train_pc* and *y_train_sig*. The RMSE for the linear regression model was 444.71.

BEST MODEL SELECTION

After comparing RMSE from various regression models; pre and post feature elimination, linear regression after feature extraction using PCA, and Kernel RBF PCA. The lowest RMSE value of 324.52 was observed with random forest regression

model on the raw cleaned dataset (pre feature elimination). This model was further implemented to create final predictions on the number of bikes rented, as well as to explain certain hypothesis questions.

EXPLANATION TO HYPOTHESES

To test initial hypotheses, the study created a new data frame which included predicted target variable (Predicted Rented Bike Count) using the best model listed above and combining it with *X_test* data frame which was used to test the performance of the final model. This new dataset was called *X_test_f* and included temperature, humidity, wind speed, visibility, dew point temperature, solar radiation, rainfall, snowfall, holiday, rush hour, seasons, quarter, request time slot and predicted rented bike count features.

1. Summer season had the most bike rentals.

To test the hypothesis, the study used the *X_test_f* data frame. For the total of bikes rented by season, summer had the most number of bikes rented (672,222), followed by autumn (541,052), spring (492,617) and winter (156,173).

2. Rush hours had the most bike rentals.

The rush hours are defined as early morning from 0600 hours to 0800 hours and evening 1700 to 1900 hours .To test the hypothesis, the study used the *X_test_f* dataframe. The predicted data showed the average number of bikes rented per hour during rush hour was 979. This number was significantly higher compared to the average number of bikes rented per hour for non-rush hours (619).

3. No Holiday days had the most bike rentals.

To test the hypothesis, the study used the *X_test_f* dataframe. The predicted average number of bikes rented on no holiday's days (713) was higher than the number rented on holidays (611).

4. Number of bikes rented in each quarter.

The study used the *X_test_f* data frame to answer the question. Second quarter of the year had the most total number of bikes rented (631457), closely followed by Q3 (627237), Q4 (383614) and Q1 (219,755).

5. Extreme temperature led to low bike rentals.

To test the hypothesis, the study used the *X_test_f* dataframe. Extreme temperatures were set at any reading below -9 and above 35 degrees Celsius. The predicted average number of bikes rented per hour on the days with extreme temperatures (423) was significantly lower when compared to the average number rented per hour on non-extreme temperature days (709).

6. Rainy days with low visibility led to low bike rentals.

To test the hypothesis, the study used the *X_test_f* dataframe. Low visibility is defined as visibility levels below 4000m, and rainy days are defined as measured rainfall of 5mm or more. Rainy, low visibility days had a predicted average number of bikes rented per hour at 72. This number was over 600 bike rentals below other days (711).

7. A snowy day with low visibility led to low least bike rentals.

To test the hypothesis, the study used the *X_test_f* data frame. Low visibility is defined as visibility levels below 4000m, and snowy days are defined as snowfall of at least an inch. Snowy, low visibility days had a predicted average number of bikes rented per hour at 126. This number was over 550 bike rentals below other days (709).

VARIABLE DEFINITION

X_train - Train split (70%) of independent variables performed using scikit learn. This split was used to train machine learning models.

X_test - Test split (30%) of independent variables performed using scikit learn. This split was used to test machine learning models' performance.

y_train - Train split (70%) of dependent variable performed using scikit learn. This split was used to train machine learning models.

y_test - Test split (30%) of dependent variable performed using scikit learn. This split was used to test machine learning models' performance.

y_pred - Predicted dependent variable (Rented Bike Count) using a specific machine learning model. This variable was reused for predicting Bike Rented Count for all machine learning models used.

X_train_sig - Train split (70%) of independent variables performed using scikit learn after feature elimination using backward elimination technique. This split was used to train machine learning models post feature elimination.

X_test_sig - Test split (30%) of independent variables performed using scikit learn after feature elimination using backward elimination technique. This split was used to test machine learning models' performance post feature elimination.

y_train_sig - Train split (70%) of dependent variable performed using scikit learn after feature elimination using backward elimination technique. This split was used to train machine learning models post feature elimination.

y_test_sig - Test split (30%) of dependent variable performed using scikit learn after feature elimination using backward elimination technique. This split was used to test machine learning models' performance post feature elimination.

X_train_pc - Train split (70%) of independent variables performed using scikit learn after feature extraction using PCA. This split was used to train machine learning models post feature extraction.

X_test_pc - Test split (30%) of independent variables performed using scikit learn after feature extraction using PCA. This split was used to train machine learning models post feature extraction.

X_test_f - Dataframe created after using the best fit machine learning model to predict the number of bikes rented. This dataframe was also used for hypothesis testing.

CONCLUSION

This study started with exploring certain features of the rental bike dataset in order to identify how these features such as season, temperature, wind speed, rainfall etc., influenced rented bike count for a particular kind of day. For better understanding of data, new features like 'Rush Hour', 'Requests Time Slot' and 'Quarter' were created. A thorough data exploration was conducted using various plots to understand the distribution of Rented bike count with respect to other independent features. To derive a conclusion for the hypotheses questions, regression models like polynomial regression, regularized regression, decision tree regression, random forest regression, and ensemble models were implemented in two phases - pre feature elimination and post, in addition to feature extraction using PCA techniques. Finally, the random forest model before feature elimination was selected as the best performing model based on the lowest root mean square value at 325.95. Once the best fit model was applied, various plots were generated on the final dataset with predicted rented bike count to answer hypotheses questions.

So altogether, after testing the data set, people in Seoul, South Korea mostly rented bikes during no holiday and rush hour time. From this we conclude that bikes are rented for daily commute. Also, weather conditions are very important and based on the hypotheses question, summer and spring had the most bike rentals. Mainly people rent bikes when the temperature is favorable. These conclusions can be used for planning purposes to add more bike stations to the city in the future and also to make current stations have adequate bikes to support the demand of rental bikes.

ACKNOWLEDGMENTS

We would like to express our sincere thanks to Professor Manjeet Rege for his guidance and useful insights throughout our case study.

REFERENCES

[1]Scikit Learn user guide. https://scikit-learn.org/stable/user_guide.html.

[2] Seoul weather information. https://en.climate-data.org/asia/south-korea/seoul/seoul-718563/

[3] Public data portal for Seoul citizens. http://data.seoul.go.kr

[4] Kernel PCA https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.KernelPCA.html

[5] Feature Selection https://scikit-learn.org/stable/modules/feature_selection.html

[6] Seoul Bike using weather data https://www.tandfonline.com/doi/pdf/10.1080/22797254.2020.1725789

[7] Feature Extraction using PCA https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html