National University of Singapore
DSA3101 - Data Science in Practice
(Department of Statistics and Data Science)

Assignment 1 - September 2021

Resource Optimisation in Bike Sharing System within Seoul, South Korea (Analyses and Recommendations)

by demand prediction, predictive maintenance and location analysis.

Team 15
Bong Jo Yee, Choong Meng Zhun,
Leo Feng Yi Fei, Niveditha Nerella,
Oh Jian Hui, Tessa Liew Lee Yi,
Wu Chenhong

Contributions

Bong Jo Yee	Feature Engineering, Clustering Analysis, Location Analysis, Presentation
Choong Meng Zhun	Random Forest, Cost Analysis
Leo Feng Yi Fei	Feature Engineering, Random Forest
Niveditha Nerella	Feature Engineering, Tableau Dashboard, Location Analysis, Presentation
Oh Jian Hui	Data Cleaning, EDA, Predictive Maintenance, Presentation
Tessa Liew Lee Yi	Linear Regression Analysis, Random Forest, Cost Analysis
Wu Chenhong	Factor Analysis, Presentation

Table of Contents

Section 1: Introduction

1.1 Bike Sharing in Seoul, South Korea

Bike sharing refers to a shared transport service in which bicycles owned by a private or public company are made available for the use of individuals on a short term basis for a price. While the first bike sharing system started as early as 1965 in the Netherlands (Runde Sache, 2011), Seoul has picked up the trend relatively recently, by providing their first public bike rental service in April 2000. Known as "Ddareungi" in South Korea, a proper bike sharing system was only set up in October 2015. Under this system, the citizens were incentivised for choosing to cycle when transferring between public transports as they could accumulate mileage (Lee, 2015).

As such, the popularity of bike sharing in Seoul has grown tremendously over the past 6 years. The payment, insurance, road and fee systems have been iteratively improved to support the growing service usages (SMG, 2016). Currently, bikes are parked at specific unmanned-dock stations (a.k.a rental offices) throughout Seoul. Users can conveniently rent a bike through their mobile phone or rental card. Despite the growing popularity, ddareungi has been experiencing losses, due to the inefficient systems (Ko, 2021). While there are countless other features to be improved, we will focus on the system of distributing the bike supplies in this project.

1.2 Objective of the Project

As data analysts of the biggest bike sharing company in Seoul, we aim to improve the company's profit through understanding consumer usage patterns. Our main objective is to *"Optimise bike supply resources to reduce operational costs by location and seasonal analysis, demand prediction and predictive maintenance"*.

In other words, we aim to provide data-based recommendations regarding the number of bikes to be transported to each location, to maximise revenue while reducing the cost of doing so. The suggestions are made one week in advance, to allow a sufficient time for the planning of logistics. This recommendation system should be sustainable, where only occasional minor updates are required.

1.3 Assumptions and Constraints

For the purpose of this project, we will make the following assumptions.

1. The company owns a total of 35,000 bikes, placed in either their main depot or a docking station. This figure is estimated from Ko (2021)'s report.
2. Company workers will conduct daily checks on the bikes at each docking station to retrieve faulty or potentially faulty bikes at around 4am, when the usage is the lowest.
3. Functioning bikes from the depot will replace the bikes retrieved for maintenance.
4. The bikes placed at each station should be minimised, to a point where there are still enough bikes for the users of the day.
5. The cost of placing bikes in the docking stations is mainly contributed by:
    a. Damages to the bikes by passersby and the weather.
    b. Transportation of bikes from the depot to the stations.

1.4 Collection of Data

Our main data set, Seoul Bike Sharing Demand Data Set (2020), is retrieved from the UCI Machine Learning Repository. The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information. The data ranges for 1 year, from December 2017 to November 2018.

To complement our main dataset, we also found 2 other datasets from Seoul Open Data Plaza (Usage Information By Public Bicycle Rental Office In Seoul (Monthly), 2020). One for the locations of bike docks and another is the rental counts at each of those docks. Locations of bike docks dataset contains the code for each bike docks and their exact location in latitude and longitude between January 2017 to December 2018. Rental counts by dock dataset consists of the bike docks' code and their respective rental and return count for all the months between January 2017 and May 2019.

As the last 2 datasets are to complement and support the analysis of our main bike sharing dataset, the date range we will focus on for the project would be December 2017 to November 2018 inclusive.

1.5 Workflow

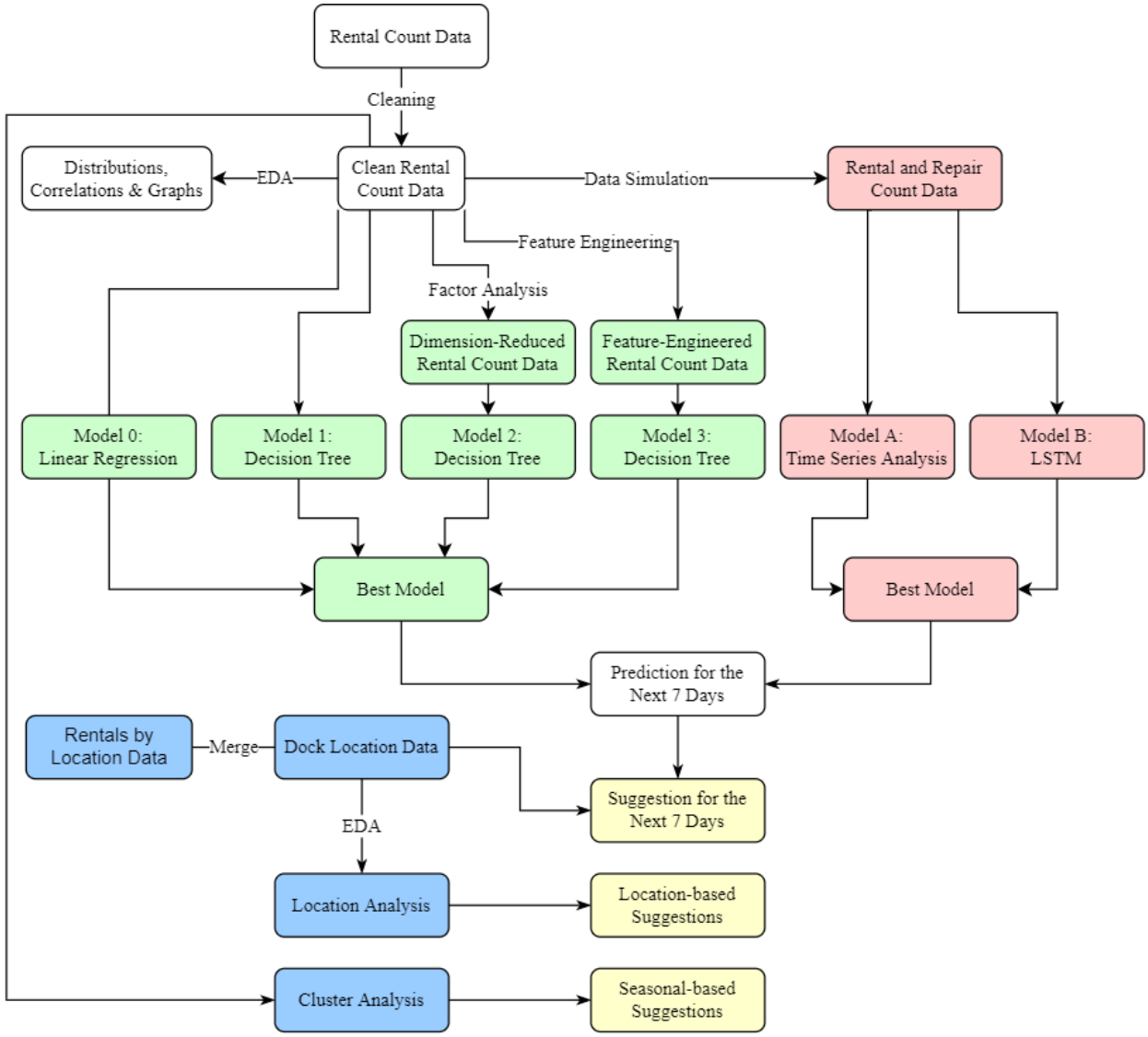

Figure 1.1: Project Workflow.

There are three main parts to our project workflow, indicated in three different colours. The cells in green represent the workflow for demand prediction, the ones in red represent predictive maintenance while the ones in blue represent location and cluster analysis. The three yellow cells in the bottom right are the business recommendations we will be proposing.

Section 2: Analyses & Models

2.1 Data Cleaning and Exploratory Data Analysis

Three datasets are available to us, (1) rental counts and weather, (2) dock locations and (3) rental counts by docks.

2.1.1 Data Cleaning

Rental count and weather data is relatively clean, with no missing or erroneous values. The only step required is to standardise the data type and measurement units. We then aggregate the observations by day instead of the original by hour; because we only perform one bike re-supply per day, hence hourly data is not actionable on. Dock locations and rental counts by docks are taken from multiple csv or xlsx files. The challenge is to combine the datasets together and deal with the Korean characters in the files.

```
[bike_clean.csv]
Rows: 365
Columns: 12
$ date            <date> 2017-12-01, 2017-12-02, 2017-12-~
$ season          <fct> Winter, Winter, Winter, Winter, W~
$ holiday         <fct> No Holiday, No Holiday, No Holida~
$ snowfall        <dbl> 0, 0, 0, 0, 0, 86, 104, 0, 0, 325~
$ rainfall        <dbl> 0.0, 0.0, 4.0, 0.1, 0.0, 1.3, 0.0~
$ rent_count      <int> 9539, 8523, 7222, 8729, 8307, 666~
$ temperature     <dbl> -1.10, 2.70, 4.35, -0.25, -3.80, ~
$ humidity        <dbl> 37.5, 55.5, 84.5, 43.5, 34.5, 76.~
$ wind_speed      <dbl> 1.40, 1.60, 1.60, 3.60, 0.00, 0.5~
$ visibility      <dbl> 20000, 14410, 3380, 19130, 20000,~
$ dewpoint_temp   <dbl> -17.40, -5.55, 2.65, -12.95, -17.~
$ solar_radiation <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
```

```
[location_jan17_may19_clean.csv]
Rows: 32,778
Columns: 4
$ code    <chr> "108", "503", "504", "505",~
$ month   <date> 2017-01-01, 2017-01-01, 20~
$ rentals <dbl> 246, 246, 232, 302, 72, 169~
$ returns <dbl> 198, 224, 261, 313, 77, 199~

[office_info_clean.csv]
Rows: 4,621
Columns: 7
$ code         <int> 301, 302, 303, 304, 30~
$ latitude     <dbl> 37.57579, 37.57595, 37~
$ longitude    <dbl> 126.9715, 126.9741, 12~
$ install_date <date> 2015-10-07, 2015-10-0~
$ LCD          <int> 16, 12, 8, NA, 16, 19,~
$ QR           <int> NA, NA, NA, 7, NA, NA,~
$ type         <chr> "LCD", "LCD", "LCD", "~
```

Figure 2.1: Summary of cleaned datasets.

2.1.2 Exploratory Data Analysis

In demand prediction, the target variable is ‘rent_count’ which specifies the total bikes being rented in a day. We have also engineered some additional date-time related variables.

We first use hierarchical clustering on the variables to investigate the similarity of all the variables. As shown in Figure 2.2, week, day_of_year and month are highly similar as expected, while temperature and dewoint_temp are also quite similar. The other variables represent relatively different information. Rent count is close to the two temperature variables indicating

that temperature plays a huge part in explaining the variation in rent_count. The same can be seen in Figure 2.3, the correlation of rent_count with the features.

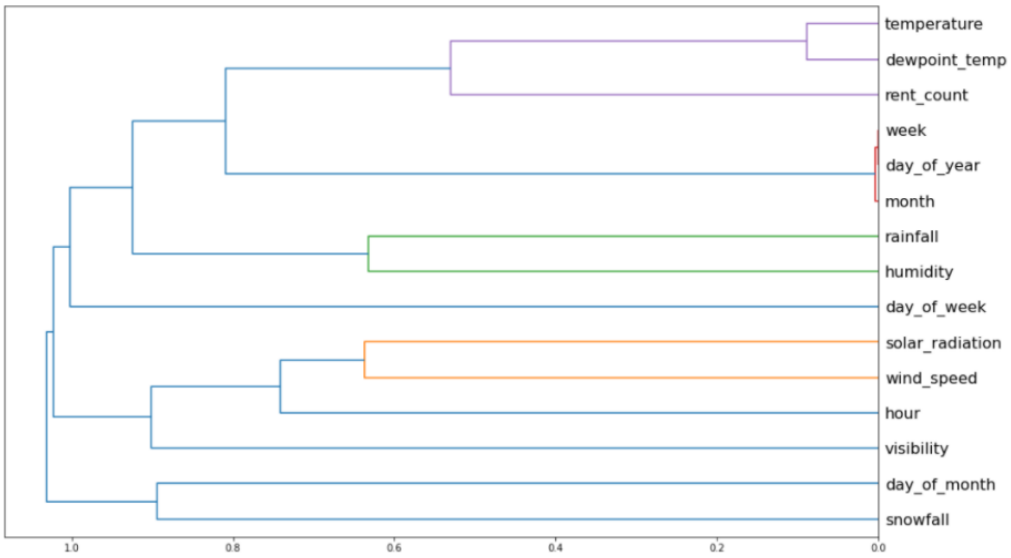

Figure 2.2: Dendrogram from hierarchical clustering on the variables.

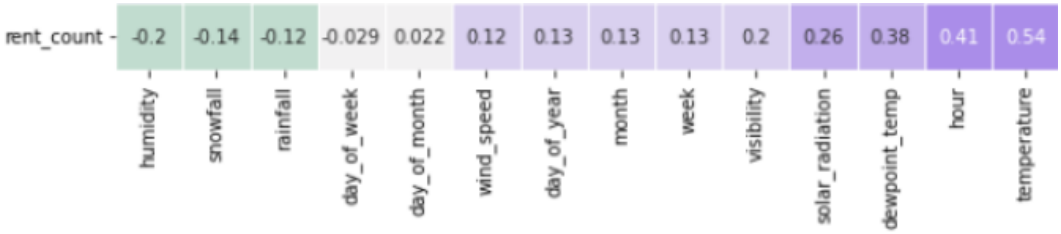

Figure 2.3: Correlation of features with rent_count.

In Figure 2.4, we have plotted rent_count against the three most correlated and useful features: temperature, solar_radiation and humidity in (a), (b) and (c) respectively. Rent_count shows a very clear non-linear increasing trend with temperature. In (b) there are a lot of points on the boundary but an increasing trend can still be seen. In (c) rent_count varies negatively with non-constant variance with humidity. In (d) and (e), the rent_count of two categorical features were compared against the categorical levels. There is a clear distinction on the rent_count distribution in Winter. Rent_count is generally highest in Summer. Holiday rent_count has a different distribution as well.

Figure 2.5 shows the variation of smoothed rent_count with day of the year, which will be useful for later parts of the project.

Other EDA are omitted to only publish relevant findings. Full EDA is attached in the codes.

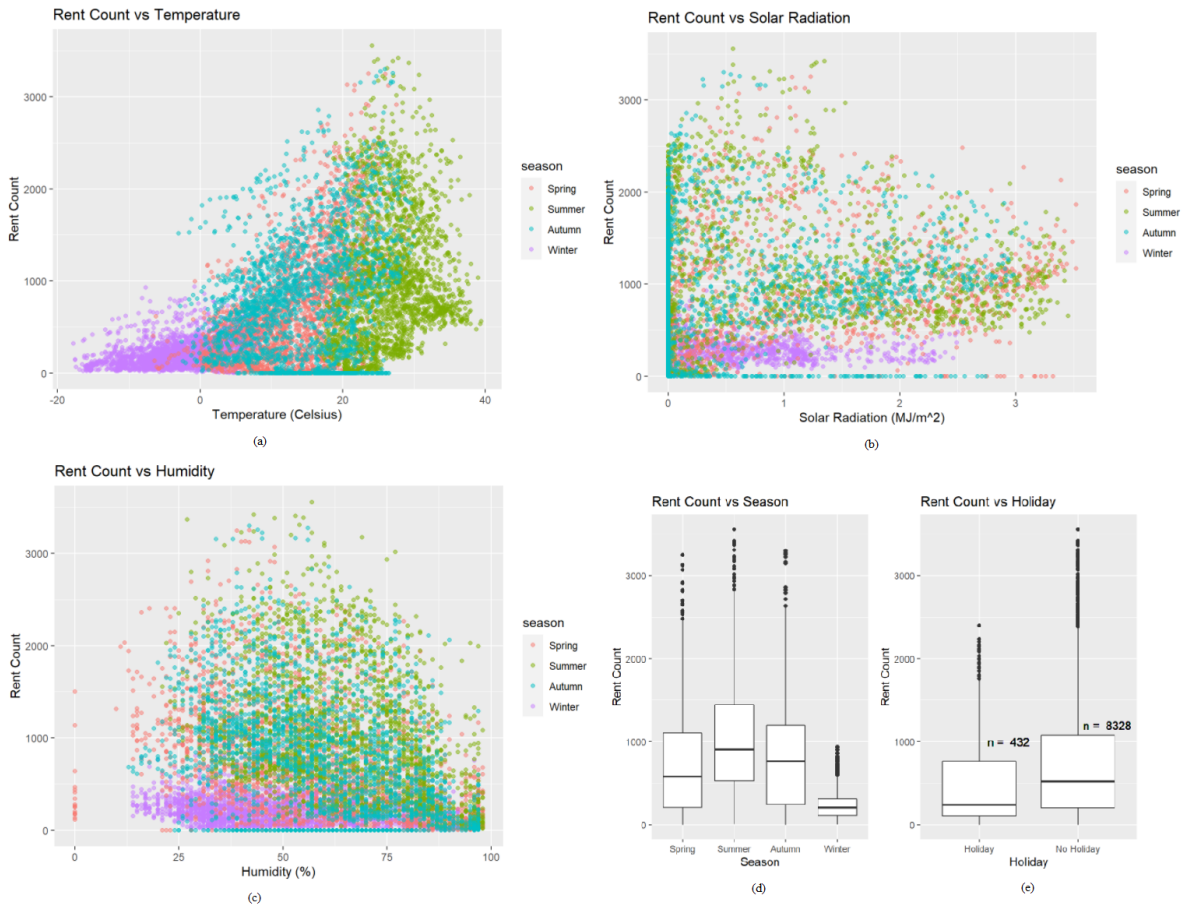

Figure 2.4: Plots of rent_count against several features.

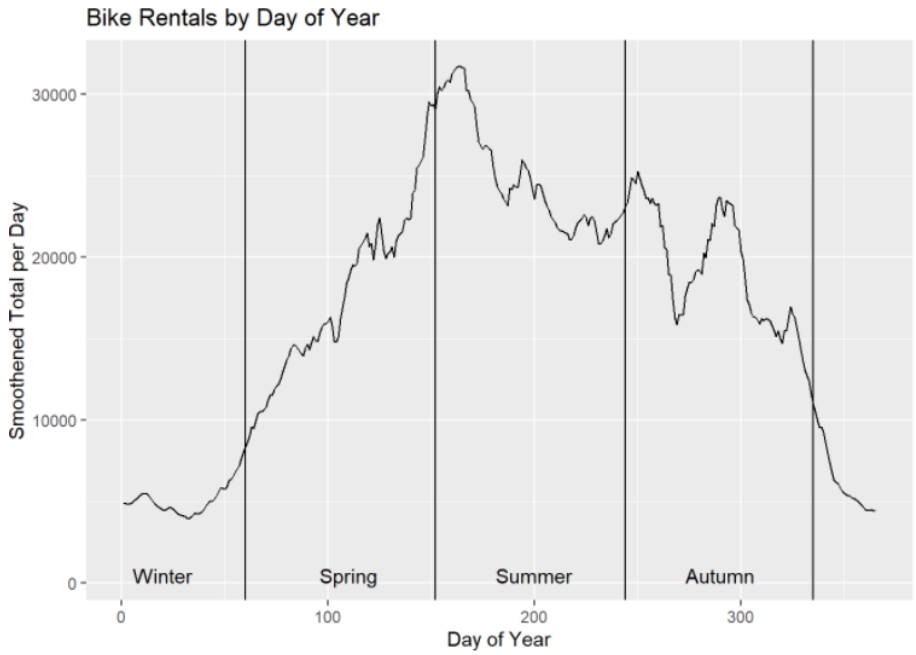

Figure 2.5: Rent count against day of the year.

2.2 Feature Engineering

The input to machine learning algorithms comprises features which are in the form of structured columns in a dataset. The features in the dataset directly influence the predictive models that we have used for forecasting the 'rent_count' results. Hence, we use feature engineering to transform the raw data into features that can better represent the underlying problem statement.

We explore feature engineering in the following 3 ways:

2.2.1 Datetime Engineering

After cleaning the main bike dataset, there are 15 features. We chose the <date> and <datetime> features in order to create 7 new granular features. The date-time stamp contains a lot of information that can be difficult for data visualisation to take advantage of (Brownlee, 2014). Hence, we decompose a date-time into constituent parts that may allow more insightful visualisations for the dashboard in the later section. We use some of the date time features in the model too.

We begin by converting the columns to DateTime format. Then we create the following new features – month, week, day_of_month, day_of_year, day_of_week, dayName_of_week and month_name. After datetime feature engineering there are 22 features in the dataset.

```
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Unnamed: 0       8760 non-null   int64
 1   datetime         8760 non-null   object
 2   date             8760 non-null   object
 3   hour             8760 non-null   int64
 4   season           8760 non-null   object
 5   holiday          8760 non-null   object
 6   open             8760 non-null   object
 7   rent_count       8760 non-null   int64
 8   temperature      8760 non-null   float64
 9   humidity         8760 non-null   int64
 10  wind_speed       8760 non-null   float64
 11  visibility       8760 non-null   int64
 12  dewpoint_temp    8760 non-null   float64
 13  solar_radiation  8760 non-null   float64
 14  rainfall         8760 non-null   float64
 15  snowfall         8760 non-null   int64
dtypes: float64(5), int64(6), object(5)
```

Figure 2.6: Before Feature Engineering

```
Data columns (total 23 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Unnamed: 0       8760 non-null   int64
 1   datetime         8760 non-null   datetime64[ns]
 2   date             8760 non-null   datetime64[ns]
 3   hour             8760 non-null   int64
 4   season           8760 non-null   object
 5   holiday          8760 non-null   object
 6   open             8760 non-null   object
 7   rent_count       8760 non-null   int64
 8   temperature      8760 non-null   float64
 9   humidity         8760 non-null   int64
 10  wind_speed       8760 non-null   float64
 11  visibility       8760 non-null   int64
 12  dewpoint_temp    8760 non-null   float64
 13  solar_radiation  8760 non-null   float64
 14  rainfall         8760 non-null   float64
 15  snowfall         8760 non-null   int64
 16  month            8760 non-null   int64
 17  week             8760 non-null   UInt32
 18  day_of_month     8760 non-null   int64
 19  day_of_year      8760 non-null   int64
 20  day_of_week      8760 non-null   int64
 21  dayName_of_week  8760 non-null   object
 22  month_name       8760 non-null   object
dtypes: UInt32(1), datetime64[ns](2), float64(5), int64(10), object(5)
```

Figure 2.7: After Feature Engineering

By doing this, we create more factors that are related to the datetime and may influence the output 'rent_count' to a higher degree than just the original date. Thus, we can make more insightful conclusions about the data.

2.2.2 Automated Feature Engineering

An open-sourced Python framework Featuretools is used to perform automated feature engineering. This tool is designed to fast-forward the feature generation process from the existing features by automating the process with Deep Feature Synthesis (DFS) algorithm. In essence, the algorithm follows relationships in the data to a base field, then sequentially applies mathematical functions along that path to create the final feature. (Manna, 2020) In other words, it will perform aggregation functions on numerical variables, grouped by the specified categorical variables of our choice.

We have chosen five categorical variables - <month>, <day>, <day_off_week>, <season> and <holiday>, as shown in Figure 2.8. As a result, we ended up with 298 features, shown in Figure 2.9, with 284 being the newly generated features from aggregation function – MAX, MIN, MEAN, SUM, STD(stands for standard deviation), SKEW(stands for skewness), NUM UNIQUE and COUNT.

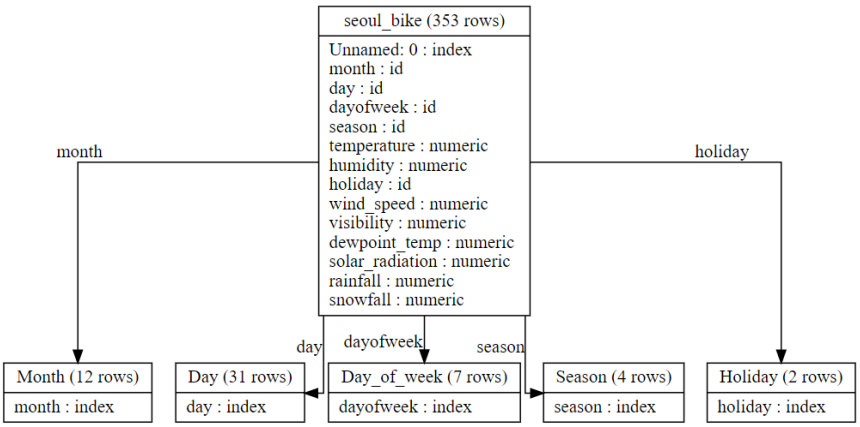

Figure 2.8: Overview of Feature Engineering

```
Index(['month', 'day', 'dayofweek', 'season', 'temperature', 'humidity',
       'holiday', 'wind_speed', 'visibility', 'dewpoint_temp',
       ...
       'Holiday.STD(seoul_bike.visibility)',
       'Holiday.STD(seoul_bike.wind_speed)',
       'Holiday.SUM(seoul_bike.dewpoint_temp)',
       'Holiday.SUM(seoul_bike.humidity)', 'Holiday.SUM(seoul_bike.rainfall)',
       'Holiday.SUM(seoul_bike.snowfall)',
       'Holiday.SUM(seoul_bike.solar_radiation)',
       'Holiday.SUM(seoul_bike.temperature)',
       'Holiday.SUM(seoul_bike.visibility)',
       'Holiday.SUM(seoul_bike.wind_speed)'],
      dtype='object', length=298)
```

Figure 2.9: Columns After Automated Feature Engineering

One of the benefits of using Featuretools is that the features generated can be easily explained by the feature name.

Take an example <Month.MEAN(seoul_bike.temperature)>. This column means that it is the mean of the daily temperature in Seoul for each month. Figure 2.10 indicates the detailed operations performed to generate the corresponding feature. After grouping by <month> variable, the mean function is called on the <temperature> variable and the resulting values are joined back to the original data set.

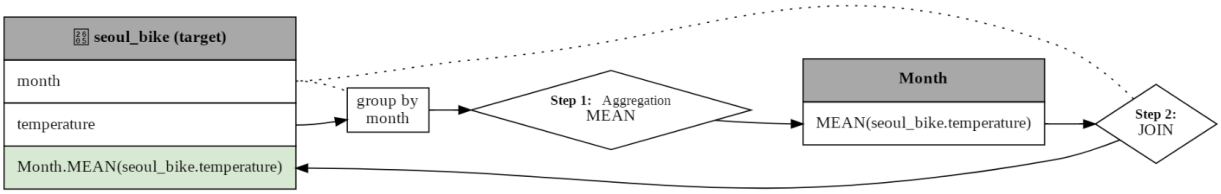

Figure 2.10: Detail of <Month.MEAN(seoul_bike.temperature)> variable

While we have created many features, we will need more experimentations to see whether they are useful. The important things are that we can automate the feature generating process and provide newly created features that might be useful in improving the accuracy of the model we are creating.

2.2.3 Factor Analysis (FA)

Factor analysis is a way to compact the dataset with many variables into just a few variables, also known as dimension reduction. Principal Component Analysis (PCA) is a technique under FA where it attempts to find uncorrelated linear combinations of the variables in the data, capturing the variability in the data while at the same time generating meaningful axes. The assumptions in PCA are: there must be linearity in the data set, and the variables exhibit relationships among themselves.

To run PCA on the main bike dataset, we have to first remove all the columns containing qualitative features. Then we perform 2 tests - Bartlett's Sphericity and Kaiser-Meyer-Olkin (KMO) on the remaining dataset to ensure PCA is indeed suitable. Doing the Bartlett's Sphericity Test gives a probability value of $0.0 < 0.05$, and KMO Test yields a test statistics, Measures Sampling Adequacy (MSA), of $0.527 > 0.5$, both results indicate that we can continue with PCA on the dataset.

With the assurance that PCA will be beneficial to us, we continue by standardizing the dataset, followed up by fitting this new standardized dataset into the PCA function. To determine how many principal components (PCs) to retain, we will need the scree plot as shown in Figure 2.11 as well as the variance explained graph in Figure 2.12.

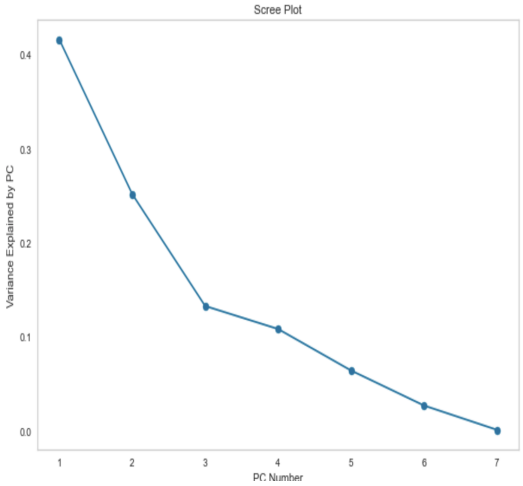

Figure 2.11: Scree Plot for PCA

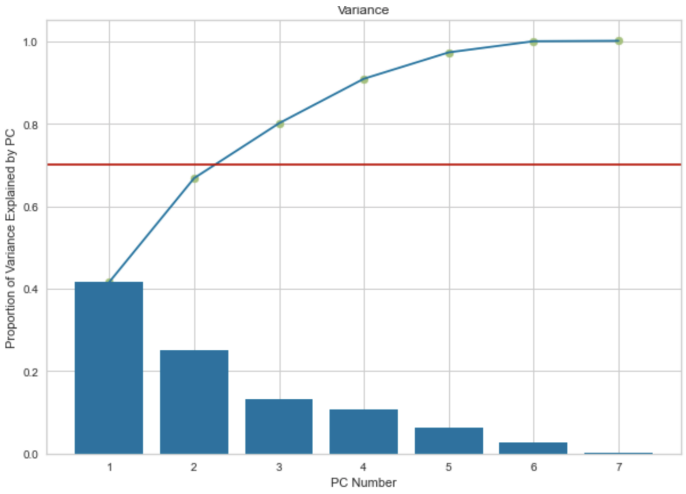

Figure 2.12: Variance Explained by PCs

The Elbow Method suggests using 3 PCs since 3 PCs can explain more than 70% of the variance of current data - with reference to Figure 2.12, when PC number is 3, the variance explained is 0.801. However, by calculating the Cronbach's Alpha values for the PCs shown in Figure 2.13 gave - 0.880, 0.683, and 0.367 respectively. The last value is significantly lesser than the acceptable minimum value of 0.7, which means it is unreliable to use PC3 to explain a significant proportion of variance of future samples of data. Cronbach's Alpha value for PC2 is slightly less than 0.7, we will still accept it as Pallant (2001) states that Cronbach's Alpha value above 0.6 is also considered high reliability and acceptable index (Nunnally and Bernstein, 1994). Therefore we will remove PC3 from the selection, leaving 2 PC columns as shown in Figure 2.14. These 2 columns will be relabelled as 'heat_effect' and 'rain_effect' and to be used in Decision Tree model 2 in the later section of the report.

	PC1	PC2	PC3
temperature	0.947734		
rainfall		0.816227	
wind_speed			0.893213
visibility			0.544678
solar_radiation	0.828107		
dewpoint_temp	0.879032		
humidity		0.84833	

Figure 2.13: Rotation Matrix for 3 PCs

	PC1	PC2
temperature	0.94991	
rainfall		0.684807
wind_speed		
visibility		
solar_radiation	0.826479	
dewpoint_temp	0.881503	
humidity		0.869279

Figure 2.14: Rotation Matrix for 2 PCs

2.3 Clustering Analysis

Applying the data mining techniques, including data clustering, on raw data for evaluation and comprehension purposes will help organisations in various domains to obtain and retrieve useful information or patterns from the data. (Ardavan, 2020) In this project, K-Means Clustering is chosen and performed.

To determine the optimal choice of the number of clusters (k), we explored the Elbow Method and the Silhouette Method. Figure 2.15 showed the total Within Cluster Sum of Squares (WCSS) against different numbers of clusters, while Figure 2.16 showed the Average Silhouette Width across different clusters.

Although the Silhouette Score for k = 3 is the highest, we have decided to go for k = 4 as suggested by KneeLocator, a Python library used to determine the elbow in Figure 2.15 because we believe it is more interpretable to analyze this data set with four clusters, instead of three.

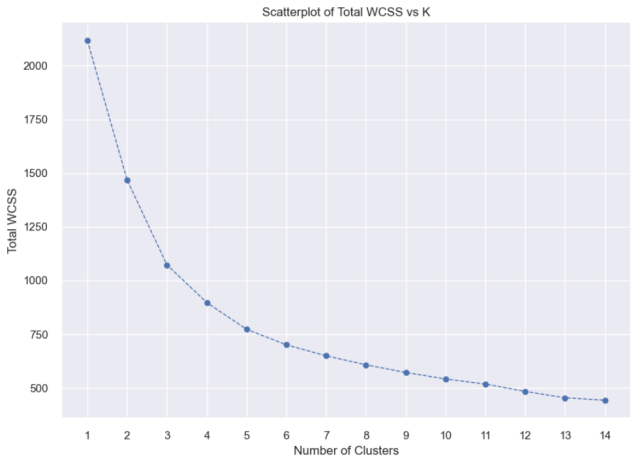

Figure 2.15: Total WCSS vs Number of Clusters

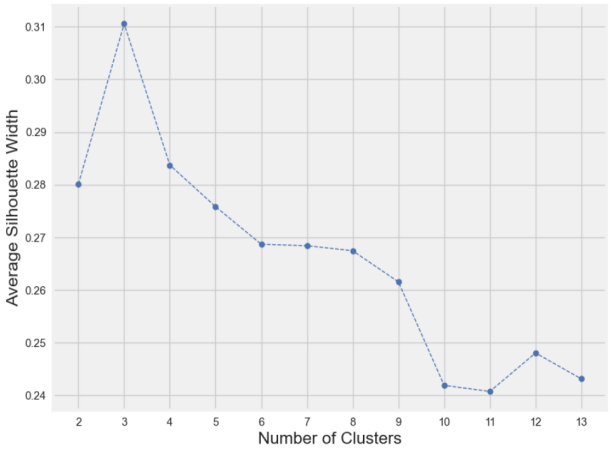

Figure 2.16: Average Silhouette Width

After performing the K-Means Clustering algorithm with k = 4, we obtained four clusters, where each coincidentally matches with each of the four seasons. Hence, we decided to name the clusters with season names – Spring, Summer, Autumn, and Winter.

Figure 2.17 shows the summary of some important features of each cluster. The mean of all variables is different among the four clusters, which is a good sign. We can see that the bike rental count is the highest in Summer, while the lowest is Winter. This might be due to the average daily temperature of each cluster, which shows similar trends. Moreover, it is also interesting to point out that even though cluster Summer and Winter both have high wind speed, the low temperature in Winter might be the contributing factor that deters the public from riding bikes in the cold weather.

Cluster:	Spring	Summer	Autumn	Winter
daily rent count:	moderate	very high	high	low
temperature:	moderate	very high	high	low
wind speed:	low	high	moderate	very high
visibility:	low	moderate	very high	high
solar radiation:	moderate	very high	high	low
humidity:	very high	moderate	high	low

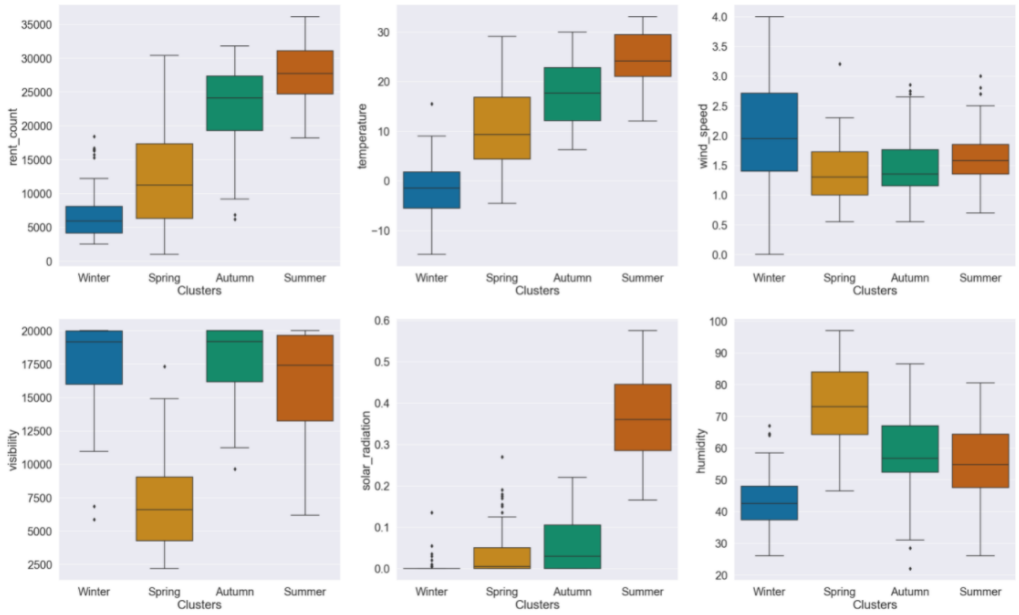

Figure 2.17: Summary of the 4 Clusters

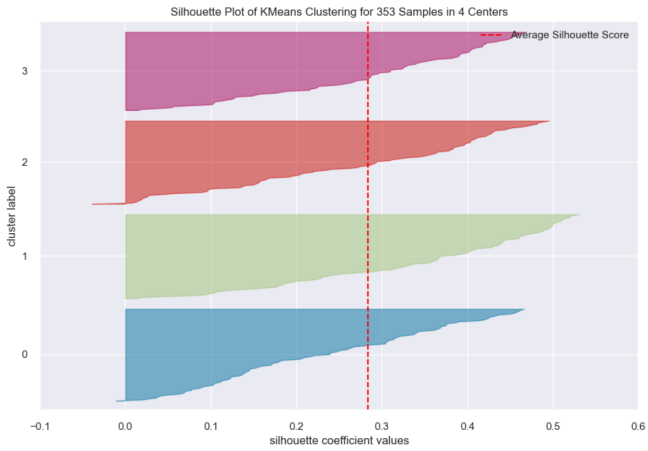

Figure 2.18: Silhouette Plot of K-Means Clustering with 4 Clusters
(Label: Cluster 0 - Spring, Cluster 1 - Winter, Cluster 2 - Summer, Cluster 3 - Autumn)

One important thing to note is that the average Silhouette Score for the clustering analysis of this data set is 0.28, as shown in Figure 2.18. Although it is considered relatively low, it is still useful to perform clustering analysis on this data because we can have an overview of the overall trends of features and discover any important variables that contributed to the increase in rental counts.

2.4 Demand Prediction

For all the models that we have presented in Section 2.4, 80% of our available data is used in 5-fold cross-validation (except Linear Regression) to tune and train the models. The remaining 20% of the available data is used as a final test set to evaluate the performance of each model.

The final metric for comparison is root mean square error (RMSE). For unbiased estimates, this metric is an estimate to the variance of our estimations.

2.4.1 Model 0: Linear Regression

Linear regression is a basic statistical tool used in prediction and inference analysis. We want to determine whether the predictors (weather conditions) have any significant effect on the daily bike rent counts in Seoul and for forecasting. We first fit a linear regression model as the baseline model for its simplicity and versatility.

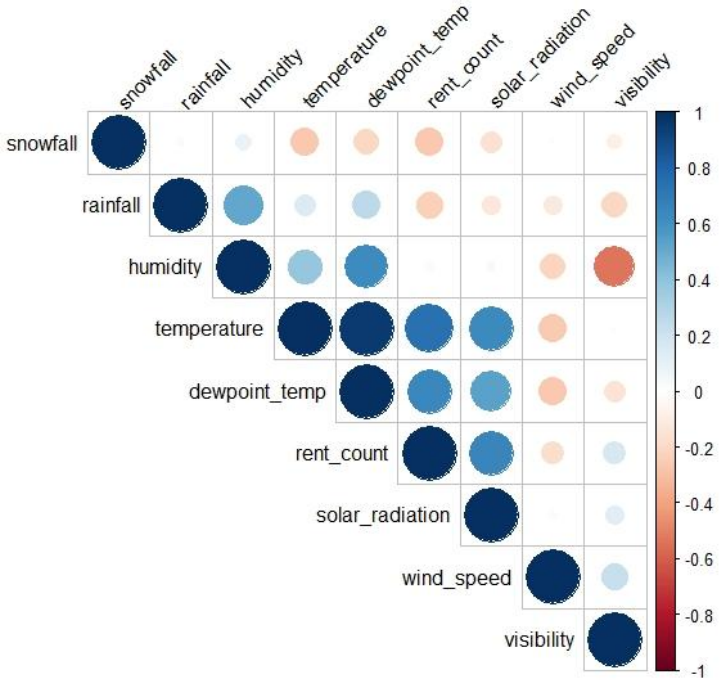

Figure 2.19: Correlation Plot for Bike Dataset.

Before fitting the model, we can see from Figure 2.19 that some of the predictors are highly correlated to each other. This gives an indication that not all the variables are required in our model. However, this is inconclusive by just looking at their individual correlations so we fit a full model as our initial model.

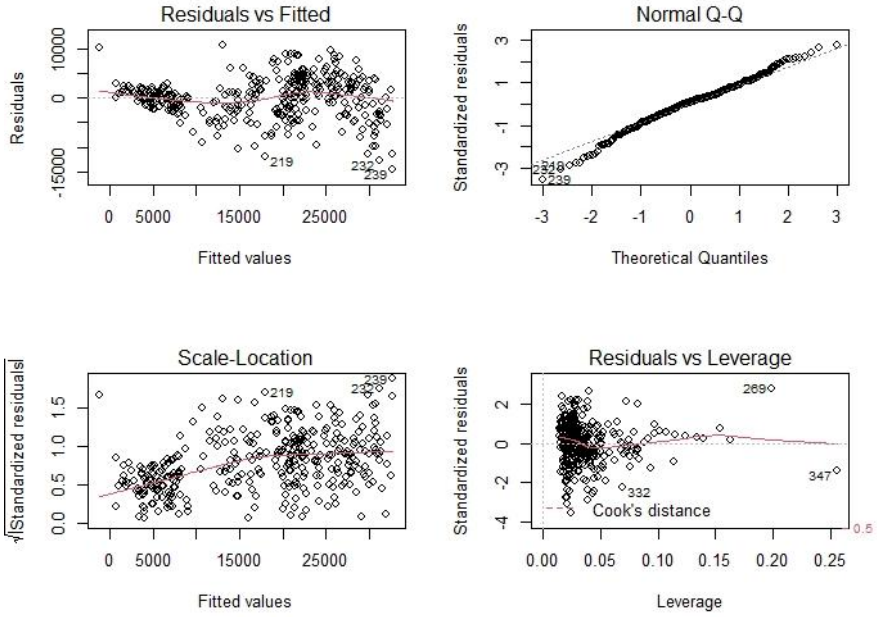

Figure 2.20: Summary Table & ANOVA Table for the Initial Regression Model.

Initial model observation:

See Figure 2.21, although the Adjusted R-squared value is relatively high, we can see that there are many insignificant variables. On top of that, checking the goodness of fit using the adjusted R-squared value is unreliable because multicollinearity may exist. From the residual analysis, we can see that the linearity assumption and constant variance assumption have been violated. Therefore, we perform transformation on the response and fit a new model. Also, we add interaction terms and use mixed stepwise selection for variable selection.

```
Call:
lm(formula = rent_count ~ ., data = bike_train)

Residuals:
   Min     1Q Median     3Q    Max
-13820  -2228    214   2522  10684

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        2.155e+04  4.903e+03   4.395 1.59e-05 ***
seasonSpring      -7.003e+03  8.664e+02  -8.082 2.09e-14 ***
seasonSummer      -7.523e+03  1.070e+03  -7.033 1.64e-11 ***
seasonWinter      -1.170e+04  1.062e+03 -11.021  < 2e-16 ***
holidayNo Holiday  2.634e+03  1.203e+03   2.189   0.0295 *
snowfall          -5.249e+00  2.815e+00  -1.865   0.0633 .
rainfall          -1.676e+02  2.451e+01  -6.835 5.38e-11 ***
temperature        2.086e+02  1.837e+02   1.135   0.2573
humidity          -7.068e+01  5.294e+01  -1.335   0.1830
wind_speed        -4.752e+02  4.124e+02  -1.152   0.2502
visibility        -1.320e-02  6.030e-02  -0.219   0.8269
dewpoint_temp      1.064e+02  1.875e+02   0.568   0.5707
solar_radiation    2.534e+04  2.739e+03   9.251  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4140 on 272 degrees of freedom
Multiple R-squared:  0.8348,    Adjusted R-squared:  0.8275
F-statistic: 114.6 on 12 and 272 DF,  p-value: < 2.2e-16

Analysis of Variance Table

Response: rent_count
                Df     Sum Sq    Mean Sq  F value    Pr(>F)
season           3 1.6366e+10 5455220533 318.2163 < 2.2e-16 ***
holiday          1 7.0168e+07   70168181   4.0931  0.044037 *
snowfall         1 2.2236e+08  222363753  12.9710  0.000376 ***
rainfall         1 3.3502e+09 3350227785 195.4269 < 2.2e-16 ***
temperature      1 1.6150e+09 1614974751  94.2054 < 2.2e-16 ***
humidity         1 4.5747e+08  457470833  26.6854 4.636e-07 ***
wind_speed       1 8.5773e+06    8577251   0.5003  0.479960
visibility       1 5.5545e+06    5554461   0.3240  0.569680
dewpoint_temp    1 3.0377e+06    3037721   0.1772  0.674125
solar_radiation  1 1.4670e+09 1466993860  85.5733 < 2.2e-16 ***
Residuals      272 4.6629e+09   17143121
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 2.21: Residual Analysis Plots for the Initial Regression Model.

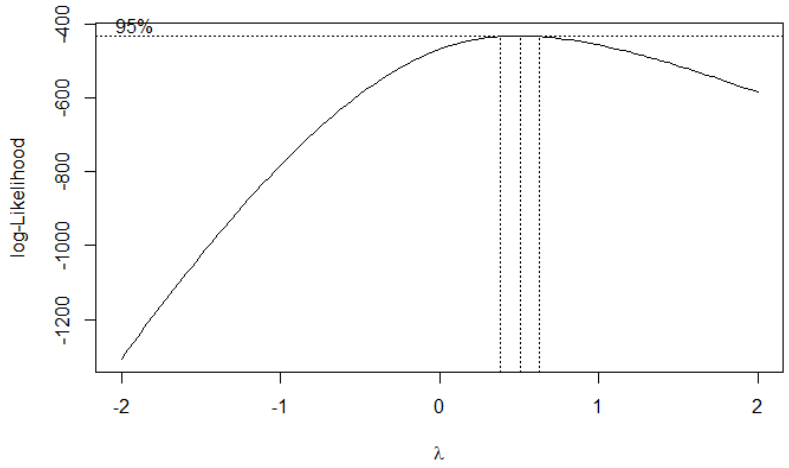

Figure 2.22: Plot of Box Cox Method for Transformation of the Response (rent_count) Variable.

After variable selection and the appropriate transformation of the response variable, the final fitted linear regression model is given by:

$$log(rent\ count) = 10.3 + 0.00996\ humidity + 2.12\ solar\ radiation - 0.0206\ rainfall - 0.000579\ snowfall - 0.482\ I_{Spring} - 0.292\ I_{Summer} - 0.894\ I_{Winter} + 0.290\ I_{No\ Holiday} - 0.0754\ wind\ speed + 0.0332\ dewpoint\ temp - 0.113\ solar\ radiation\ *\ dewpoint\ temp$$

```
Call:
lm(formula = log(rent_count) ~ humidity + solar_radiation + rainfall +
    snowfall + season + holiday + wind_speed + dewpoint_temp +
    solar_radiation:dewpoint_temp, data = bike_train)

Residuals:
     Min       1Q   Median       3Q      Max
-1.41262 -0.13093  0.02611  0.17857  1.10272

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                   10.2603248  0.1413452  72.591  < 2e-16 ***
humidity                      -0.0099584  0.0020631  -4.827 2.31e-06 ***
solar_radiation                2.2050380  0.2850135   7.737 1.98e-13 ***
rainfall                      -0.0206367  0.0017577 -11.741  < 2e-16 ***
snowfall                      -0.0005792  0.0001994  -2.905 0.003976 **
seasonSpring                  -0.4823827  0.0604718  -7.977 4.15e-14 ***
seasonSummer                  -0.2922391  0.0793748  -3.682 0.000279 ***
seasonWinter                  -0.8935149  0.0736727 -12.128  < 2e-16 ***
holidayNo Holiday              0.2902772  0.0847522   3.425 0.000709 ***
wind_speed                    -0.0753821  0.0277784  -2.714 0.007077 **
dewpoint_temp                  0.0332074  0.0039366   8.435 1.94e-15 ***
solar_radiation:dewpoint_temp -0.1128587  0.0183951  -6.135 2.98e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2931 on 273 degrees of freedom
Multiple R-squared:  0.8653,    Adjusted R-squared:  0.8599
F-statistic: 159.5 on 11 and 273 DF,  p-value: < 2.2e-16

Analysis of Variance Table

Response: log(rent_count)
                              Df Sum Sq Mean Sq  F value    Pr(>F)
humidity                       1  0.000   0.000   0.0048  0.945038
solar_radiation                1 60.552  60.552 705.0371 < 2.2e-16 ***
rainfall                       1  9.495   9.495 110.5514 < 2.2e-16 ***
snowfall                       1 10.414  10.414 121.2606 < 2.2e-16 ***
season                         3 59.700  19.900 231.7080 < 2.2e-16 ***
holiday                        1  0.625   0.625   7.2830  0.007395 **
wind_speed                     1  1.528   1.528  17.7861 3.365e-05 ***
dewpoint_temp                  1  5.103   5.103  59.4229 2.375e-13 ***
solar_radiation:dewpoint_temp  1  3.233   3.233  37.6414 2.977e-09 ***
Residuals                    273 23.446   0.086
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 2.23: Summary Table & ANOVA Table of the Final Regression Model

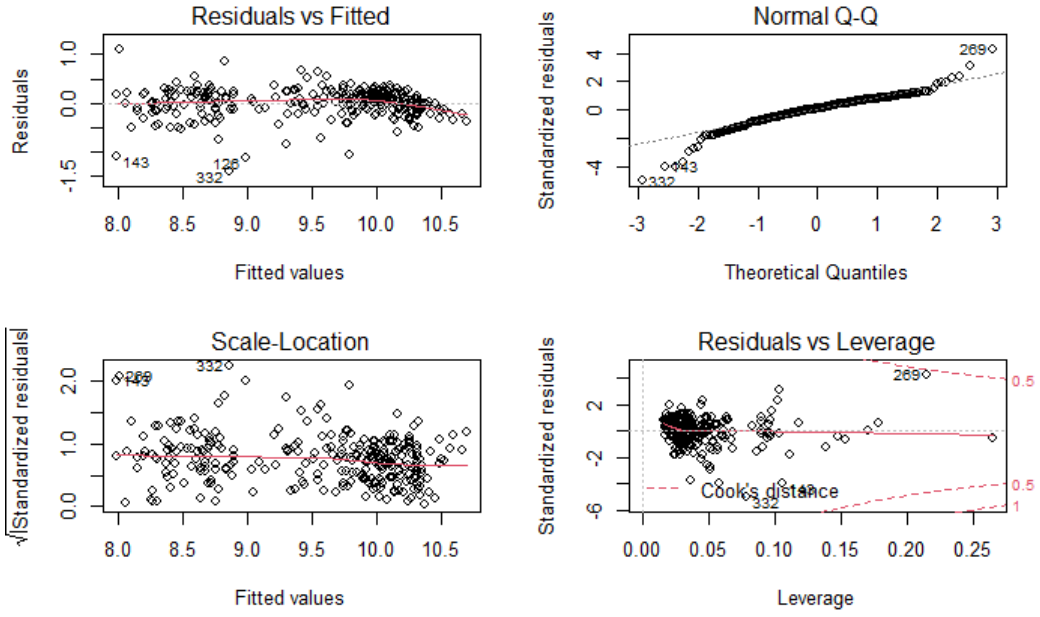

Figure 2.24: Final Regression Model Residual Analysis Plots.

Comparing the initial regression model and the final regression model, we can see that the adjusted R-squared value has increased moderately. Other than that, we can see that all the variables are highly significant in the final regression model. However, according to Figure 2.24,

we can see that there are a few outliers (eg. points 126, 269 and 332) and a few unusual high leverage points (eg. points 143, 269 and 332), which has significant influence on our regression model and thus renders our regression model less appropriate for our data. Other than that, the Root Mean Square Error (RMSE) of our regression model is 4426.039 and the Mean Absolute Percentage Error (MAPE) of our final fitted regression model is approximately 22.25%. This will be taken as a baseline metric.

2.4.2 Model 1: Random Forest on Full Data

The next naive model would be creating a decision tree using all the variables as input. In our model, we use random forests instead of a decision tree. In this random forest model, we are creating several decision trees to estimate the output instead of just one decision tree. This means that the model is not influenced solely on important features of just one decision tree and allows the model to generalise over the whole dataset with reduced variance. The output could be estimated by averaging or majority voting based on the output of all the decision trees.

Looking at one of the decision trees in our random forest, to keep the decision tree readable, we set the maximum depth of the decision tree to 3.

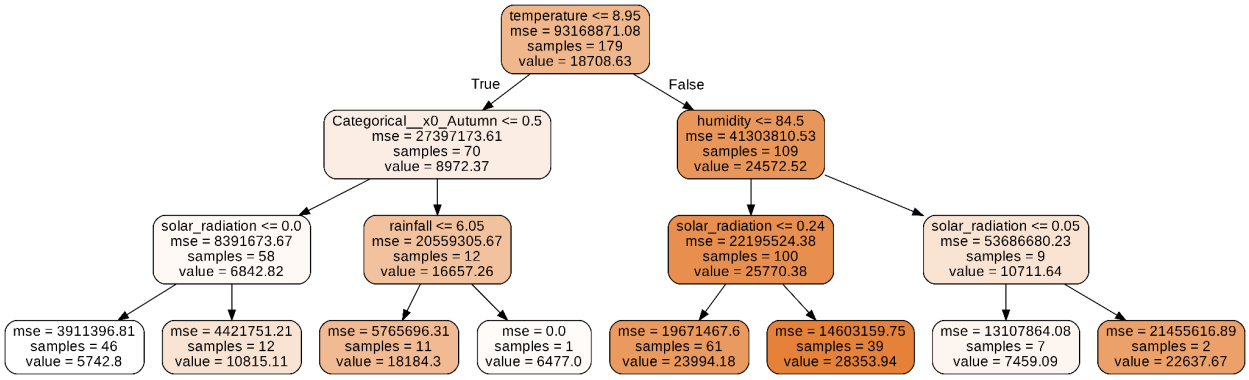

Figure 2.25: Initial Model Tree Diagram

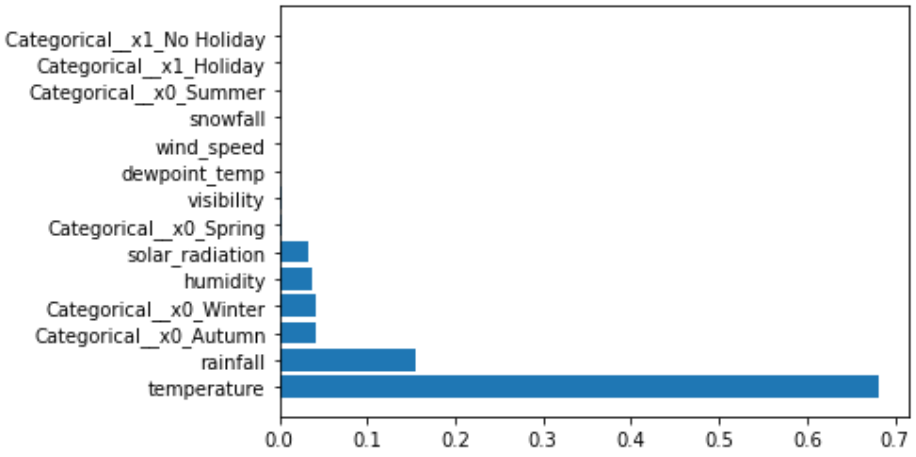

Figure 2.26: Initial Model Feature Importance

In the first model, we have an explained variance score of 0.835, indicating that the model could be a good fit for the data. We also checked for its cross validation score and got an explained variance of 0.823, which shows that the model may not be overfitted. The mean absolute percentage error for this model is 34.4%. To compare our models, we use the metric root mean squared error, in this model, the root mean squared error is 4253.675.

Checking the importance of the variables, in Figure 2.26, we can see that temperature has the highest feature importance out of the other variables in predicting rental counts.

However, there could be a better model that gives us a better root mean squared error depending on how the model parameters are changed, hence, we randomly tested parameters and graded them using this metric. We got the final model from this testing and one of the decision trees of the model is shown in Figure 2.27.

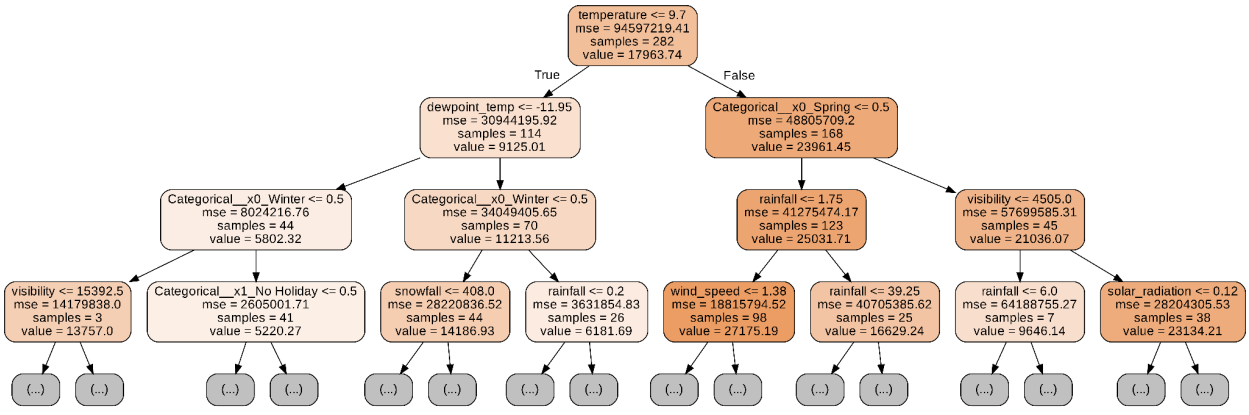

Figure 2.27: Tuned Model Tree Diagram

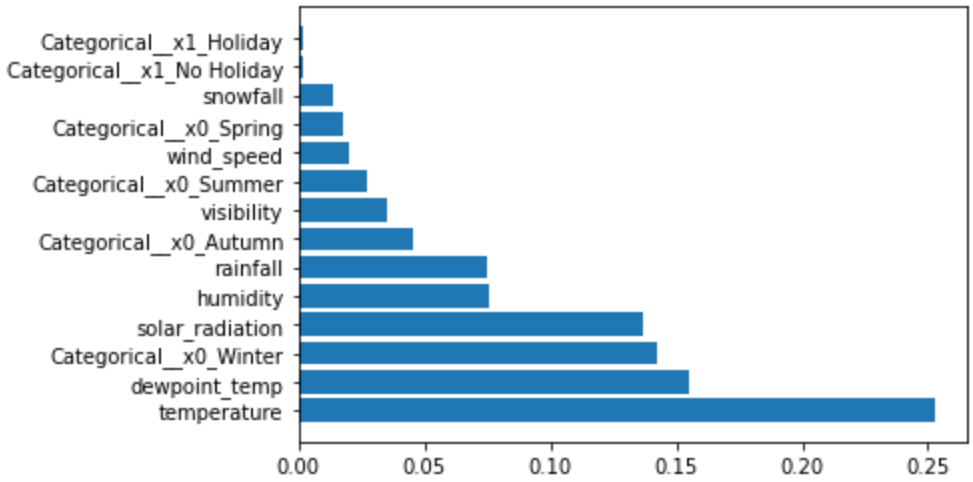

Figure 2.28: Tuned Model Feature Importance

The decision tree of our recommended model has a depth of 10, hence only 3 depths are shown. The explained variance score of our tuned model is 0.901 and its cross validation score got us an explained variance score of 0.888. The mean absolute percentage error for this tuned model is 31.6%. In our metrics used to calculate the suitability of our model, the root mean squared error of this model is 3283.973, which is lower compared to the initial model.

Also, in this model, even though temperature has lesser feature importance, the other variables now have more impact on the output. To see how our variables affect the output, we also created a plot using SHAP values shown in Figure 2.29. SHAP values are a measure of how a feature contributes to the prediction of the model output based on the actual value of the feature.

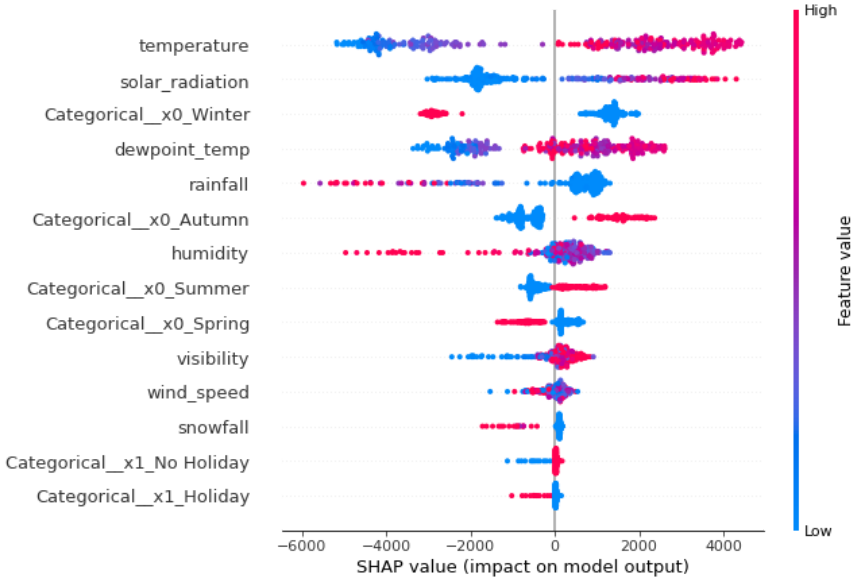

Figure 2.29: Tuned Model SHAP Diagram

The y labels correspond to the features sorted based on their importance, whereas the x labels correspond to how each feature affects the output based on its own value. For example, for the rainfall feature, it might be safe to say that there might be some negative correlation between rainfall and rental counts. As rainfall increases, more people are less likely to rent bikes.

2.4.3 Model 2: Random Forest on Principle Components Data

In this Decision Tree model, the variables "heat_effect" and "rain_effect" correspond to PC1 and PC2 respectively which are obtained from the PCA. This is to take into account the existence of multicollinearity in the data and also to reduce the dimensionality for easy interpretation.

The naive Random Forest model has a mean absolute percentage test error of approximately 30.7%. We perform feature importance to decrease the impurity of the split and remove redundant variables.

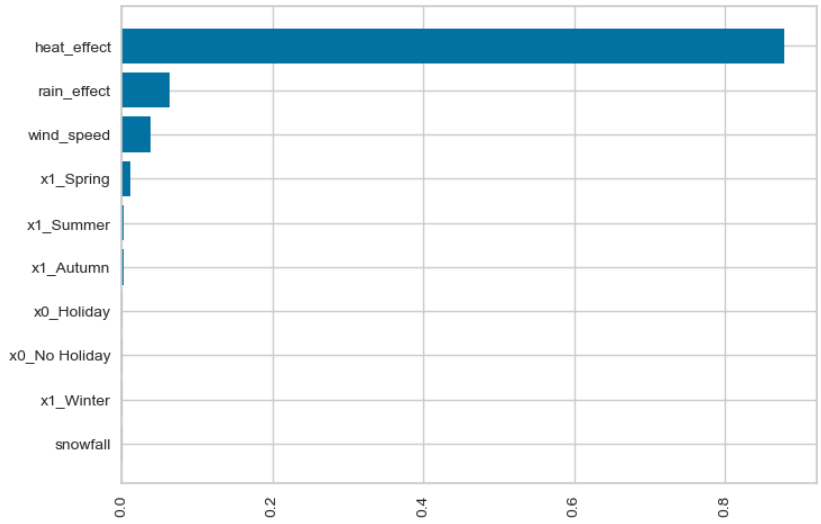

Figure 2.30: Random Forest Feature Importance for initial Model.

From Figure 2.30, we can see that only heat_effect, rain_effect and wind_speed are relatively important features. We remove the other variables from our Random Forest model and see that our absolute percentage validation error is 30.37%. We then perform hyperparameter tuning to further improve our Random Forest model. For hyperparameter tuning, we perform many iterations of the entire K-Fold CV process using GridSearchCV, using the bootstrap method and MSE criterion.

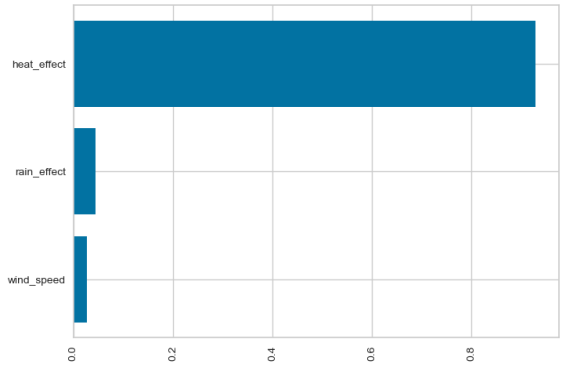

Figure 2.31: Random Forest Feature Importance for final Model.

After hyperparameter tuning, the best random forest model has a maximum depth of 7 and a minimum sample leaf of 2. The absolute training percentage error of 30.6%.

After using this model on our test data, the final test set error percentage obtained is 38.1% and the RMSE is approximately 7537 which are both relatively high. Hence, the model is not suitable to predict the daily bike rent counts in Seoul for the following week.

2.4.4 Model 3: Random Forest on Feature Engineered Data

From feature engineered data as mentioned in Section 2.2, we obtained a total of 300 features. Our goal is to find out how many and what features to use to build the model. By using the randomForest package in R, we first sorted our features according to increasing node purity (i.e., IncNodePurity). . IncNodePurity is a measurement of the total increase in node purities from splitting on the variable, averaged over all trees. For a regression tree, node purity is measured by the residual sum of squares. We then use rfcv which is the cross validation function defined in randomForest package to calculate the mean of squared residuals of the models that top N features with highest IncNodePurity were used. By plotting out the relationship between numbers of features and the mean of squared residuals we want to find an optimal number that has lower error.

Number_of_Features <dbl>	Mean_of_Squared_Residual <dbl>
0	17708.32
1	17751.40
2	679642.63
3	2848229.32
5	759463.11
8	1700052.26
12	1848173.88
17	2128876.02
26	2977717.30
39	4100528.46

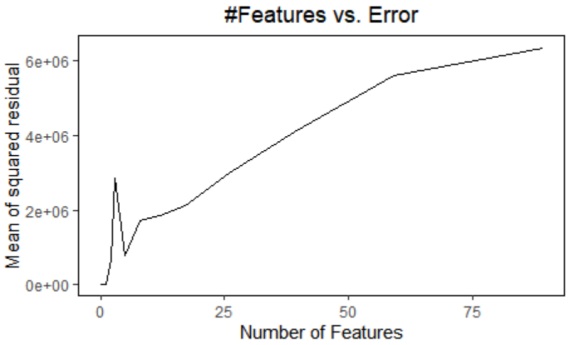

Figure 2.32: Table and Plot of Feature number vs mean of squared residual.

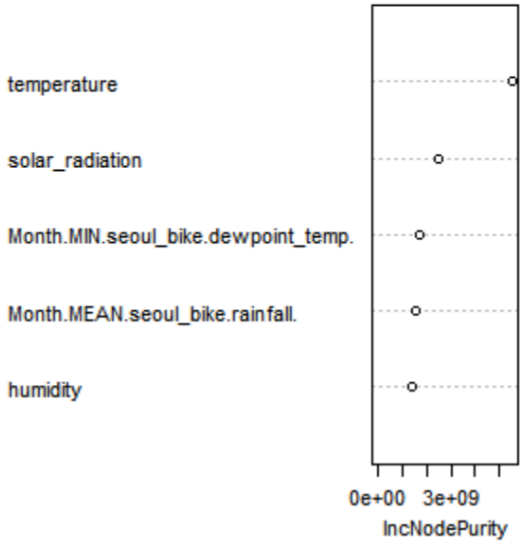

Figure 2.33: Top 5 features ranked by increasing node purity.

We have decided to select the top five features despite the errors of selecting one or two features are lower. This is because selecting one or two features will diminish too much information. Due to the lack of instances in our train set, it is risky to predict our result with no more than two features. Figure 2.33 shows the features: temperature, solar radiation, month.min.seoul_bike.temperature, month.mean.seoul_bike.rainfall and humidity.

The random forest model built by these five features gives a RMSE of 3173 and variance explained of 89.2%. The hyperparameters and summary is shown below.

```
Call:
 randomForest(formula = rent_count ~ ., data = train, importance =
TRUE,      ntree = 100)
               Type of random forest: regression
                     Number of trees: 100
No. of variables tried at each split: 1

          Mean of squared residuals: 10473306
                    % Var explained: 89.21
[1] "temperature"
[2] "solar_radiation"
[3] "Month.MIN.seoul_bike.dewpoint_temp."
[4] "Month.MEAN.seoul_bike.rainfall."
[5] "humidity"
[1] "mean absolute percentage error: 3173.31828800149"
```

Figure 2.34: Summary table of the random forest tree with 5 features.

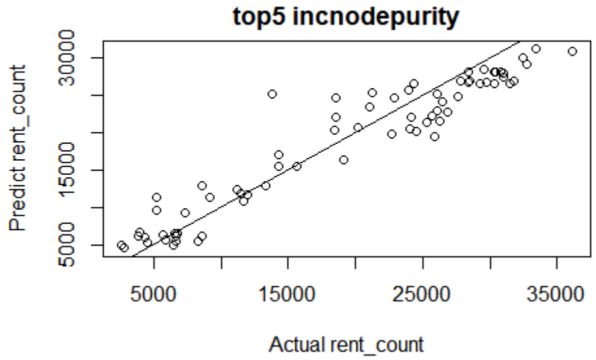

Figure 2.35: Plot of actual rent count vs predicted rent count.

This model has improved the root mean squared test error to 3173 (mean absolute percentage test error to 19.5%, which is 11.2% better than our previous best model). Hence this model will be used in the decision making process in Section 3.

2.5 Predictive Maintenance

To optimise the logistic process and data-driven recommendations in Section 3, the ability to predict the number of bikes that require maintenance is crucial.

2.5.1 Data Simulation

While predictive maintenance is a common application of AI today, the datasets are most of the time, not open to the public due to sensitive information about a company's machinery. According to online forums, predictive maintenance datasets for the public are mostly simulated according to the creators' domain knowledge. Since there is no complete predictive maintenance dataset related to bikes or similar equipment available online, we will simulate our own data.

Justification of simulating our own data:

1. In practice, it is very easy to collect the data of "number of bikes that require maintenance". Should our recommendation workflow be accepted by the company, the company can start collecting this information daily to refine our model.
2. The simulation is based on researching the effects of weather on bike-related equipment online in Fredrick (n.d.) and Pazdan (2020).

The data simulated has the following features:

1. Depends on weather variables and rent_count for up to a week before.
2. Mainly follows the trend of rent_count.
3. Has Gaussian noise of standard deviation 10.

Figure 2.36 shows the simulated data, the points are the simulated data for repair counts, while the lines are at a 10% scale of the Rent Count or Total Number of Bikes.

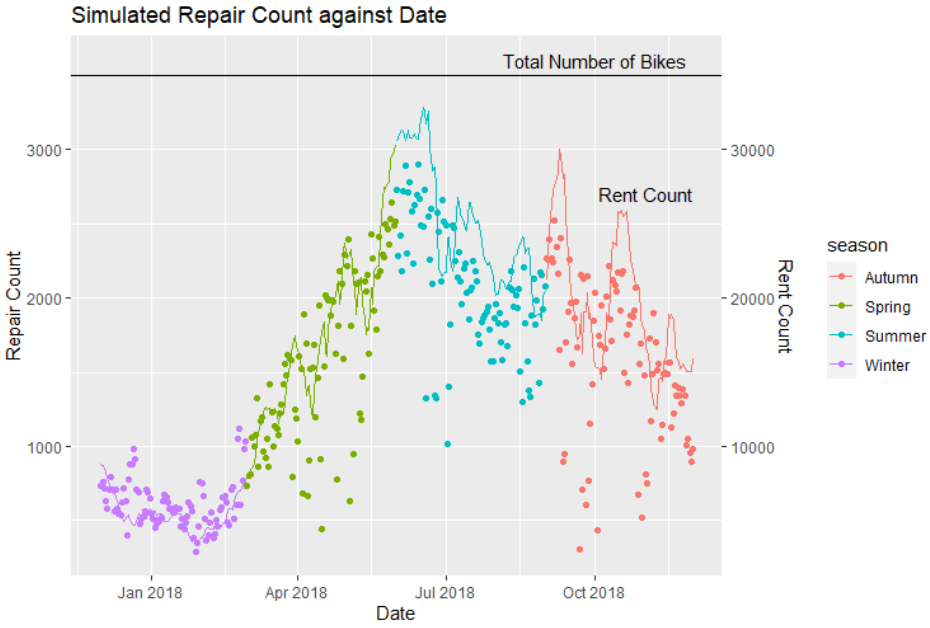

Figure 2.36: Simulated data.

2.5.2 Model A: Autoregression Model

$$X_t = m_t + s_t + Y_t$$

A Classical Time Series Decomposition model is assumed for the repair count (known as rep_count from this point onwards). X_t is rep_count random variable, m_t is the trend component, s_t is the seasonal component and Y_t is a random noise variable of mean zero.

Seasonal Component: A period of 7 days is best used to describe the data, as compared to 14 or 30 days. A moving average filter with window size of 7 is first used to smooth the data, then the residuals were used to estimate the seasonal components at each of the 7 time points.

Trend Component: A linear model of order 3 is used to estimate the trend, after subtracting the seasonal component off. 74.2% of the total variation is explained by this model, and the residual plots show slight violation of the constant variance assumption at high predicted rent_count, hence the model might provide poorer predictions in this region.

Noise Modelling: After subtracting the trend and seasonal component from the actual data, we expect the residuals belong to the noise random variable. The sample autocovariance of the

residuals is observed to be not independent and identically distributed. Using the Yule-Walker algorithm, we determined the autoregressive model that minimises the Corrected Akaike Information Criterion (AICC). Autoregressive model of order 10 is suggested. Hence we model the noise by the AR(10) model.

The final RMSE on the test set is 700.5, which is far from ideal. Hence we take this model as our baseline and try to improve the results using another model.

2.5.3 Model B: Long Short Term Memory (LSTM)

LSTM is a neural net framework that takes into account data from previous timepoints. It is preferred over Recurrent Neural Network (RNN) because (1) LSTM accounts for longer term dependencies and (2) Gonfalonieri (2019) and Peters (2020) have both suggested the use of LSTM for predictive maintenance.

To evaluate our model, we used a stacked validation split method which is more suitable for time-dependent models as compared to the conventional cross validation split. Three validation sets of size 50 and one test set of size 50 were created.

Below are the results of feature selection:

Changes	Validation MSE 1	Validation MSE 2	Validation MSE 3
Full	NA	NA	318462
w/o dewpoint_temp	399901	289980	256236
w/o temperature	378970	324943	202046
w/o temperature, wind_speed	380308	235905	120903
w/o temperature, wind_speed, solar_radiation	204352	372550	206873

Figure 2.37: Feature selection performance

With this, we finalise our features to be snowfall, rainfall, rent_count, humidity, visibility, dewoint_temp, season and holiday. We then further tune our model by adjusting the number of layers, number of nodes, number of iterations (epochs) and optimiser.

In Figure 2.38, the validation loss achieves its elbow value at a low number of epochs, at around 5. This is due to the limited data that are available to us. With more data, the number of epochs can be increased to fine tune the model further.

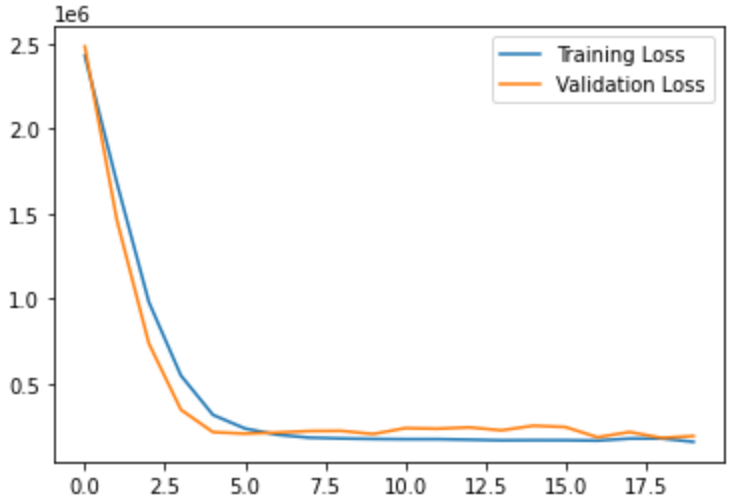

Figure 2.38: Mean Squared Error loss against number of epochs.

The final hyperparameters are:

Layers	1. Dense: 16 nodes, identity activation 2. LSTM: 50 nodes 3. Dense: 16 nodes, identity activation 4. Dense: 1 node, relu activation
Optimiser	Adaptive Momentum
Number of Iterations	5

Table 2.1: Hyperparameters for LSTM

The final RMSE on the test set, using the model tuned from the training and validation set is 352.7 (mean absolute percentage error is 24.9%), which halves the error from the autoregressive model. Hence, we will use this model for the decision making process.

2.6 Location Analysis

Upon data exploration we notice that the bike rental numbers differ from location to location. Certain rental offices have a consistently higher number of rentals while some follow the opposite trend. Hence, along with factors like weather, time, and day; it can be suggested that 'location' of the rental offices or docks is another influencing factor for determining the rental counts.

Depending on the popularity of the location, businesses decide how the bicycles are efficiently distributed amongst the rental offices. More bikes are in stock at those offices with higher rent

count. However, as the demand for bike rental increases, unavailability and uneven distribution of bikes becomes a critical problem for the bike sharing program, as this could undermine the users' experience and waste resources.

The optimal locations of the bike services can be used to drive an increase in profit, efficiency and maximise accessibility of the bikes to the consumers (Revelle et al., 1970). So, selecting appropriate locations for bike rental offices is a significant contributor to the accomplishment of the bike sharing system.

In order to explore how the location of rental offices determine the rent count fluctuation, we created Tableau dashboards to visualise the bike rental demands geographically. The results from the dashboard can provide us a better understanding of Seoul's cycling network, evaluate the current system in place and provide insights on how to enhance the existing bike sharing services to make it beneficial for more people.

The Tableau dashboard can be accessed at shorturl.at/bhtKU.

2.6.1 Location Density Dashboard

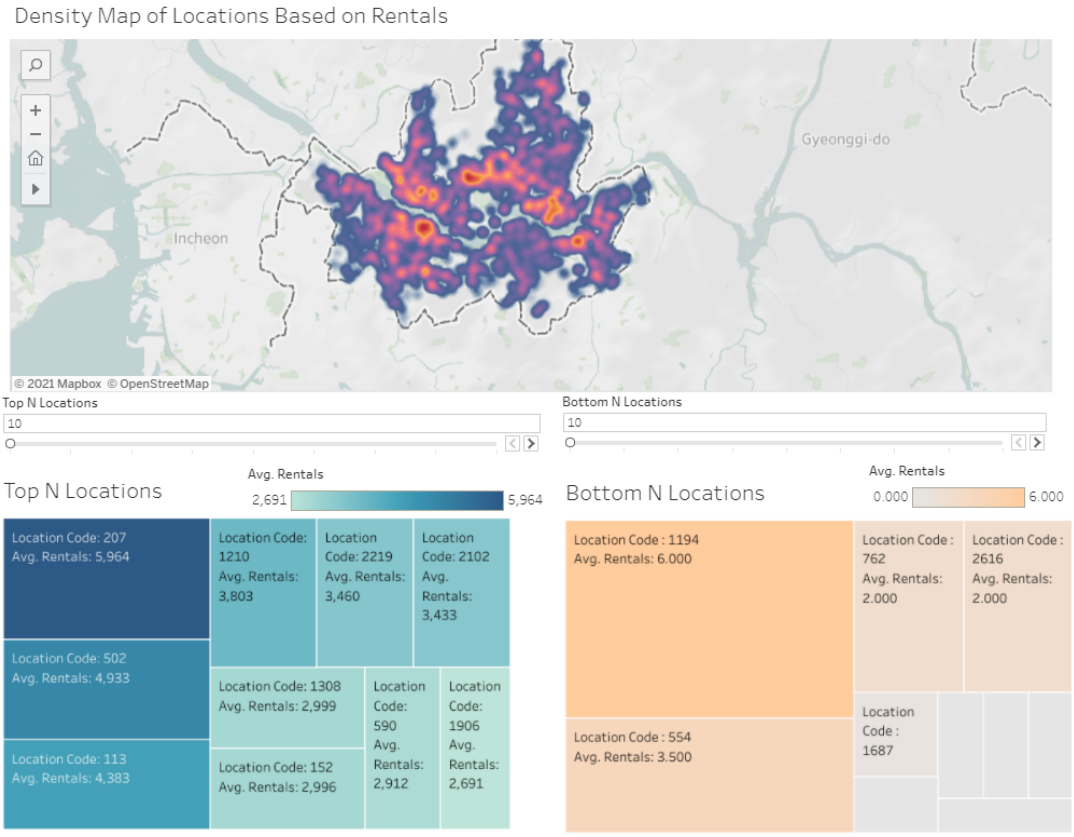

Figure 2.39: Tableau Dashboard 1 Screenshot

This dashboard shows the density map of bike rentals across different locations in Seoul. The locations marked red have the highest concentration of bike rentals. It also allows users to customize the number of offices/stations to be shown in Top N Locations and Bottom N Locations.

The top three popular offices/stations with most rentals in Seoul are the following:

Location Code 207: Located in front of Yeouinaru Station Exit 1. The station is situated near Yeouido Hangang Park, which is one of the most easily accessible parks along the Han River, especially famous for bicycle rental. Publics cycle along the riverside to experience the beauty of Han River with its many majestic bridges overhead and beautiful seasonal flowers.

Location Code 502: Located in front of Exit 1 of Ttukseomwon Station, near Ttukseom Hangang Park. Another cycling place alongside the majestic Hangang River Seoul.

Location Code 113: Located in front of Hongik University Station Exit 2, nearby Hongik University. The place is famous for being near to Hongdae, known for its urban arts and indie music culture, local shops, clubs, and entertainment.

The bottom stations with less rentals are mostly located on the outskirts of Seoul, which might explain why there are less people renting bikes from these stations.

2.6.2 Major Attractions and Business Districts in Seoul Dashboards

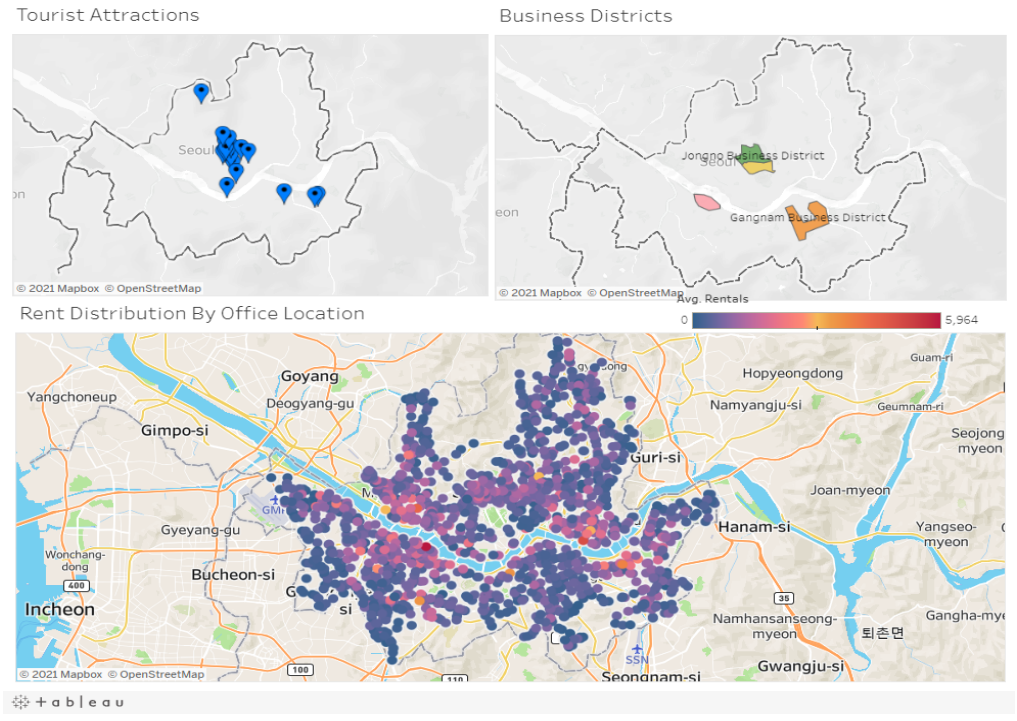

Figure 2.40: Tableau Dashboard 2 Screenshot

The top left map is that of the major tourist attractions in Seoul. These attractions in central Seoul can be located on the rent distribution map and we can see that these areas are brighter coloured, implying higher rental counts. The main reason is that Seoul offers a tourist package called 'Seoul Bike Tour', which gives tourists the option to pedal their way through some of these must-visit attractions that the city has to offer. This package gained quite some popularity amongst tourists as one could visit landmarks where large tour buses or public transportation couldn't take them. Furthermore, biking provided an escape from the heavy road traffic.

The business districts are also regions with high rental counts especially 'Yeouido Business District' and 'Gangnam Business District' which are located closer to the Han river. These business districts are home to a variety of attractions for locals too with popular shopping malls and entertainment areas.

For example, 'Gangnam Business District' is famous for its high-end shopping areas with flagship stores and premium shops of the most luxurious global fashion brands. Gangnam also houses several hi-tech and media agencies and another popular name for it is the 'Beverly Hills of Seoul' (Partners, 2020). This explains the higher rental counts in the area.

2.6.3 Land Use and Bicycle Paths in Seoul Dashboard

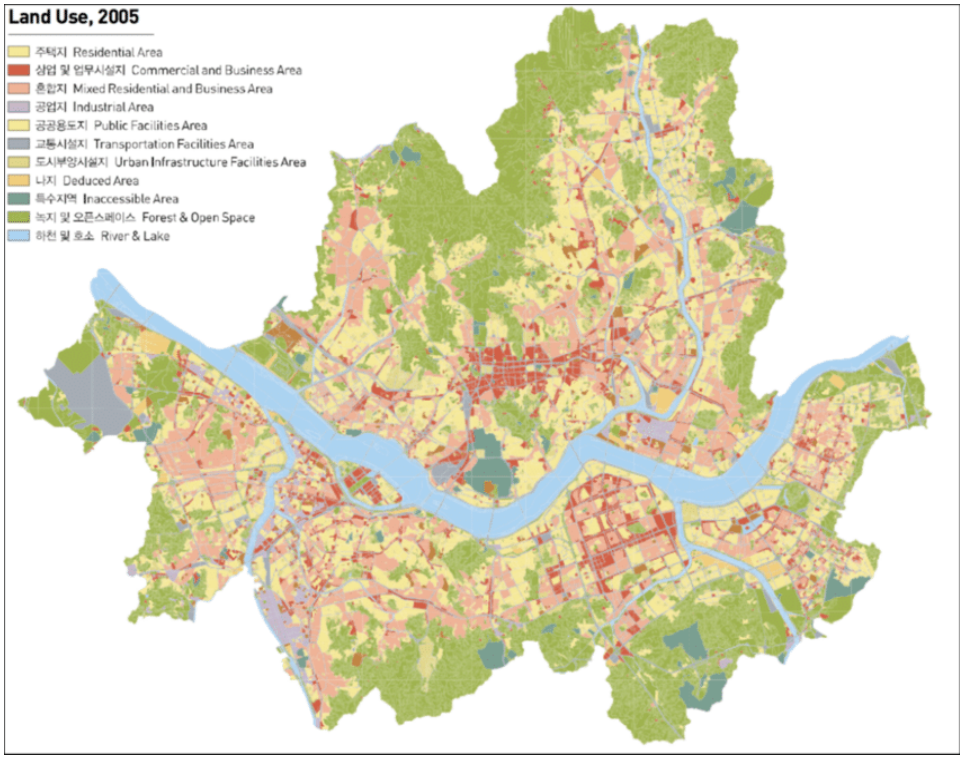

Figure 2.41: Seoul Land Use

Figure 2.41 shows the Land Uses of Seoul. According to the map, red coloured areas show Commercial and Business zones, while the yellow coloured areas denote the Residential zones.

Figure 2.42 shows the quality of bicycle paths in Seoul. The ranking system of the map is based on whether or not bike lanes and paths are separate from cars and pedestrians. The highest quality trails are indicated in blue, while the worst quality road segments are displayed in orange or red (Savannah & Karakoc, 2017).

In this dashboard we try to compare the Seoul Land Use maps and Bike Paths to the rental counts. Upon comparing Figures 2.41 and 2.42, we notice that Seoul's cycling tracks mostly lead to recreation destinations, not commercial and business areas. Despite this, the rental counts are high near some business districts too but the bike paths are not low-stress(bike-friendly) roads.

As a result, we conclude that rentals are higher in many yellow coloured areas which are the residential areas. Furthermore the rentals are also higher along the blue cycling paths which are safe paths dedicated solely for cyclers.

Figure 2.42: Seoul Bicycle Paths

From Figure 2.42 we can observe that most of the blue cycling paths tend to be clustered around the Han River. The Han River (Hangang) is one of the main outdoor attractions in Seoul no matter what the season is, it offers visitors a ton to do and the easiest way to get around is by bike. There are about 12 Han River Parks to visit (CRICCHIO, 2020). The Han has convenient bike rental shops located at the various subway stations around it. We can see that our rent count

distribution is consistent with this, as most of the locations clustered around the Han River have the highest bike rent counts.

2.6.4 Seasonal Variations Dashboard

The bar chart on the left in Figure 2.43 shows how bike rent counts vary with the temperature in Seoul. The colors are marked according to the four seasons, with orange as Spring, red as Summer, blue as Autumn, and green as Winter. In this dashboard we explore the effect of seasons on peak hour rental variations.

Cycling is more sensitive to weather conditions compared with other modes of transport. (Sabir, 2011, as cited in Pazdan, 2020) It suggests that any change in weather conditions may have a significant effect on the bike rental by the public

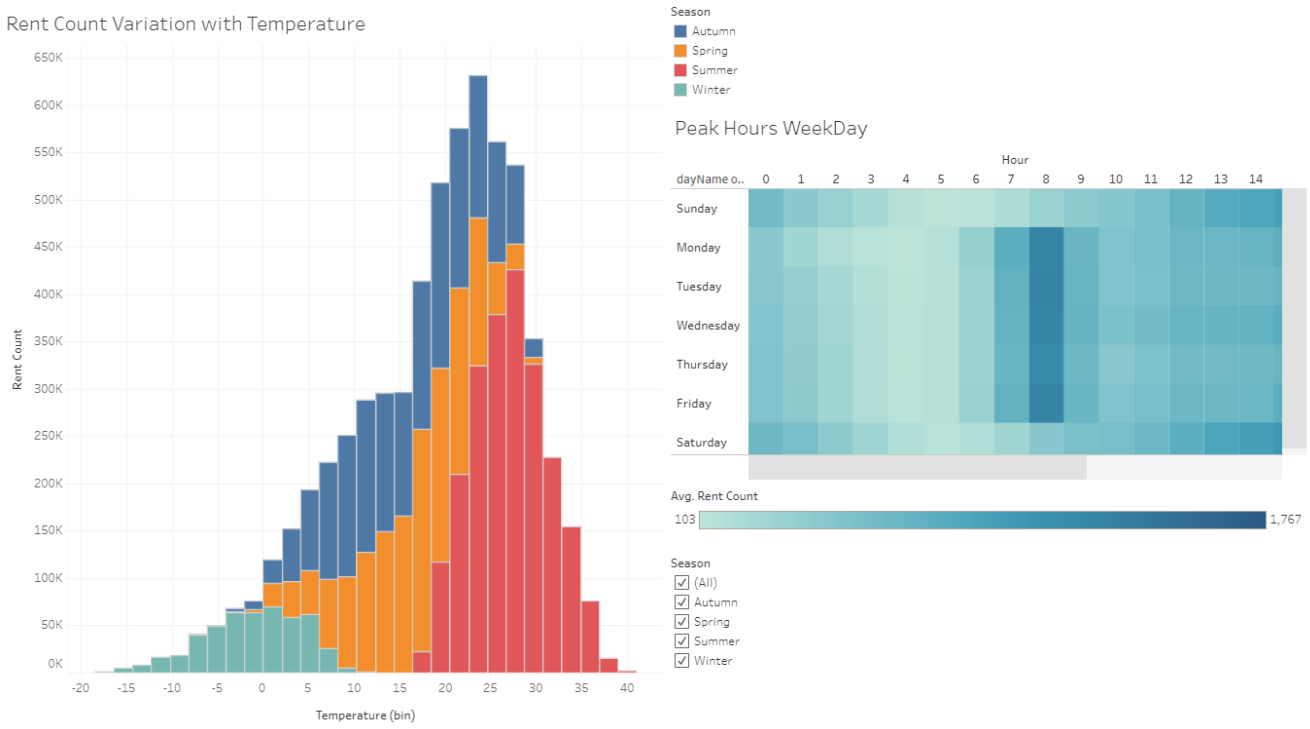

Figure 2.43: Dashboard 3 Screenshot

.

Referring to Figure 2.43, we can infer that higher temperature leads to more bike rental counts, with an all time peak observed when the temperature is around 23 °C.

Winter in Seoul normally associates with freezing, cloudless, clear, and dry weather. January, being the coldest month, has a mean temperature around -2.4 °C with the lowest almost dropping below -15 °C. Research has shown that during winter, layers of ice can build-up and alter the basic layouts of pathways within the built environment. This affects the cycle-ability of a local area, and presents challenges to delivering good, functional cycle networks. (Chapman &

Larsson, 2021) Consequently, there's a sharp drop in rental counts when the temperature dropped below zero celsius in the winter season.

The graph on the right of the dashboard indicates how the rent counts vary throughout the day in a week for each season. The peak hours during weekdays are similar for all seasons, that is during 7-10 am in the morning and 5-8 pm in the evening. There isn't any specific peak hour for weekends since the crowd is more spread out in the afternoon. The only difference is that during winter, the crowds start to gather as early as noon and end around 5pm, while that of other three seasons is normally around 3-10 pm. We can also observe that the maximum number of daily rental counts drop by 70-80% in winter, as compared to the other seasons.

Section 3: Results & Recommendations

3.1 Model Manual

The following manual is a step-by-step guideline on how to use our models to distribute bikes across the country in a cost efficient manner.

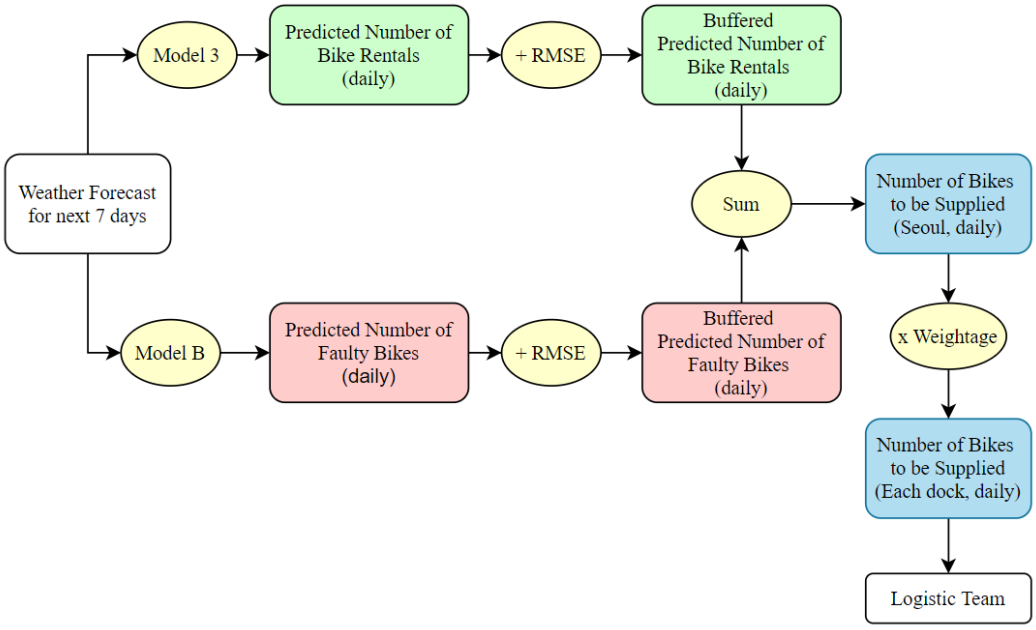

Figure 3.1: Business Decision Flowchart

In this project, we have developed two final models to guide the bike distribution process. We use Model 3 (Random Forest on Feature Engineered Data) to predict the bike demand for the upcoming week, and allow a buffer of 3236 (RMSE on test set). Then, we use Model B (LSTM) to predict the number of bikes that need maintenance for the upcoming week, and allow a buffer of 353 (RMSE on test set). With this, we know the number of bikes which have to be transported across Seoul.

The reason we chose to take the RMSE as buffer, is that if the data is Gaussian and estimation is unbiased, the RMSE is approximately the standard error of estimation. Then by taking RMSE as the buffer, it is approximately a 70% confidence interval for the mean. As we do not want to oversupply due to the costs of maintenance and transport, 70% is a reasonable level of significance. Although there might be some violations to the above statistical assumptions, when this solution is applied in practice, it is able to achieve our target of cost saving, see Section 3.2.

To estimate the bikes to be required at each dock station, we calculate the weightage for each dock station, which will be explained in Section 3.1.1. We split the bike supply according to this weightage to all the dock stations in Seoul. Then we have the figure of bikes required everyday for the upcoming week at each dock station.

This information will be passed to the logistics team to plan their routes and transportations.

3.1.1 Monthly Rental Weightage/Percentage of Each Dock

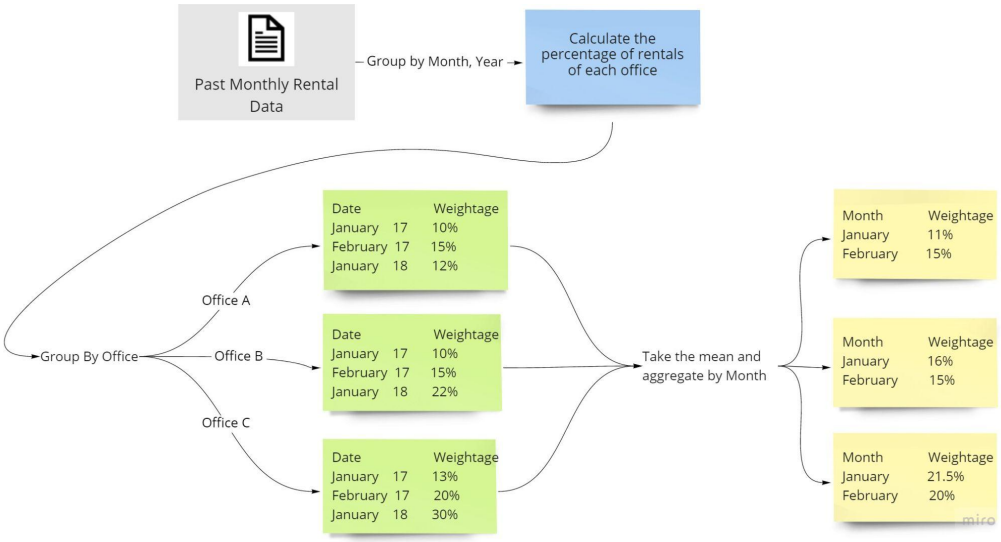

Figure 3.2: The Summary Chart of Obtaining Monthly Weightage

There's a total of 1,566 bike rental docks in Seoul. Our team has come out with a suggestion on how the company can distribute the bikes among different offices, that is according to the proportion of rental count of each office out of Seoul's total in the past month.

Referring to Figure 3.2, to obtain the weightage, we first need to prepare the past monthly rental counts, with the office details. For each office, calculate the percentage of rentals it accounted for in that month of the year. With all the monthly data available, perform aggregation by taking the average of the rental weightage on the monthly level. By doing so, we will have all the monthly weightage for all bike rental offices in Seoul, ready to be used.

3.2 Demonstration

In this example, we used the forecasted weather conditions to predict the bike rental and repair counts for the following week. Using the models, we obtained the predicted counts and added the respective buffers as shown in Figure 3. For this demonstration, we will use the last 7 days provided in the dataset. The sum of the predicted counts and buffer will be distributed to the different bike docks based on their percentage of total rentals for that month in previous data.

Day	Predicted Repair Count	Predicted Rent Count	Actual Repairs Count	Actual Rent Count	Total Actual Count	Total Predicted + Buffer
1	1515	7927	1382	6477	7859	13031
2	1618	8716	1338	11212	12550	13923
3	1318	10775	1007	17162	18169	15682
4	1318	11004	1054	16282	17336	15911
5	725	10948	954	16524	17478	15262
6	741	10866	893	16423	17316	15196
7	842	10401	982	16297	17279	14832

Table 3.1: Bike Counts for the last 7 days

Although our models tend to underestimate the actual count, we will always add in the buffer to account for random error. Also, the actual rental count does not take into account different people using the same bike after either of them have finished using the bike. Hence, in reality, the number of bikes needed to be supplied may be in fact lower than the actual rent count.
But, more data may be used to refine our model as we only used 352 data points to train our model.

For the dock station corresponding to the code 113, the rental percentage (weightage) is approximately 0.623%. The number of bikes to be assigned to this location each day is as follows:

Day	1	2	3	4	5	6	7
Approximate Bike Supply	82	87	98	100	96	95	93

Table 3.2: Dock 113 Daily Bike Supply

Here are some assumptions for us to compare our models and other strategies:

- The monthly maintenance of a bike while it is outside costs USD 120 (adjusted to USD 100 according to cost of living), which is reduced to USD 50 because the company has an in-house team for repairs. The repair count corresponds to bikes that have to be sent back to the workshop for repairs.
- Since Seoul has a bike rental service, every person subscribes for a daily pass for 2 hours which costs 2000 KRW or USD 1.71.

- We will assume we use the strategies stated later to predict the last 7 days to compare the predicted and actual values.
- The formula we will use to calculate profit would be: Total Revenue - Total Cost. Total revenue is calculated by multiplying the number of bikes actually rented with the rental fees; Total cost is calculated by multiplying the number of bikes that are still available for service with the daily repair costs for each bike that is out in the field.

For comparison with our models, we will compare them with two other strategies, (1) naively sending 35,000 bikes to the respective dock stations based on percentage of rentals everyday and (2) setting the mean of rental counts of the last 7 days as the amount of bikes to be sent.

- The actual revenue for the week would be USD 171,644.67.
- Using our model, with the bikes we predicted and distributed, the predicted cost would be USD 160,378.33. The profit from this would be a net gain of USD 11,266.34.
- If we use the method of sending all 35,000 bikes everyday, the predicted repair count will increase, hence the predicted cost would be USD 380,553.33. The profit from this would be a net loss of USD 208,908.66.
- If we use the strategy of setting the mean of actual rental counts from the last 7 days as the daily amount of bikes to distribute, the amount of bikes to distribute per day is 14340 bikes. Comparing the bikes distributed with the actual rent count, there are some days where the bikes distributed are lower than the actual rent count, losing revenue. Hence, the revenue gained from this strategy yields USD 152,855.19 while the predicted cost would be USD 154,616.67. The profit from this would be a net loss of USD 1,761.48.

Comparing all three strategies, we can see that only the strategy of using our model will give us a net gain in profit. Hence, using our model may be a better method compared to other strategies available as we attempt to maximise revenue while minimising costs that results in a better profit margin.

3.3 Location, Daily and Seasonal Recommendations

3.3.1 Location Recommendation

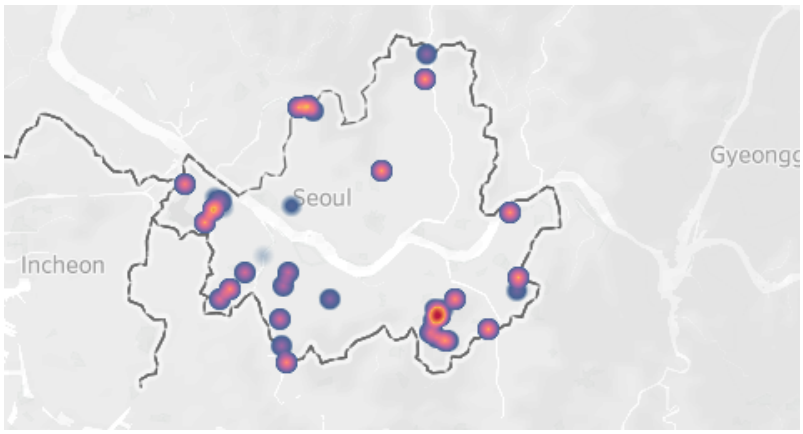

Figure 3.3: Top 50 Rental Office Locations

The locations marked in the above figure are the bottom 50 bike rental locations for which the average rental lies between 0 and 73. From the map, we see that some of these low rental offices are geographically located quite close to one another. These rental offices are situated far away from the bustling core of Seoul and mostly lie in the outskirts. The cost incurred for running these locations may be more than the profit made through bike rentals. Thus, it might be more profitable for the company to close or merge some of these bike rental offices and thereby save the operational cost of running them.

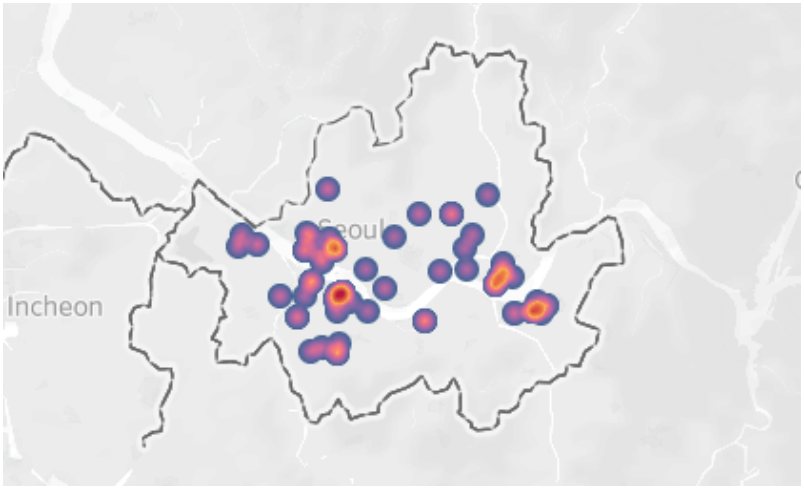

Figure 3.4: Bottom 50 Rental Office Locations

The above map shows the locations of the top 50 bike rental offices with average rental counts ranging from 1,756 to 5,964. Most of these offices seem to cluster around the Han River and its

tributaries, specially at those spots that have a subway station near them, while some are located in residential areas. Since these locations have such high average rental counts, it's vital for the company to ensure that the rental docks at these locations are stocked with a sufficient number of bikes at all times. It would also be profitable for the company to open more offices along the river. Some promotions that can be run by the company to increase rentals are -

- Offer a 'Monthly bike rental passes' as the Seoul Bike system offers benefits in connection with Seoul's mass transportation system, including buses and subways. If a bus or subway is taken within 30 minutes of returning a Seoul Bike or vice-versa, users can receive a "mileage benefit". Adding on to this, users with this pass can receive more perks and discounts. (SMG, 2016)

- Rent-1hr-get-1hr-free promotion or even couple bike rental offers, near popular hangout locations such as parks along the Han River in order to boost the rental numbers.

3.3.2 Daily/Hourly Recommendations

Figure 3.5 below shows the average distribution of recounts per hour throughout the year, irrespective of the season. The average number of rentals and returns are very scarce between 00:00–5:00 AM. Hence, this would be the perfect time for night rebalancing operations to meet the optimal initial number of bicycles at every station for the next day's operations.

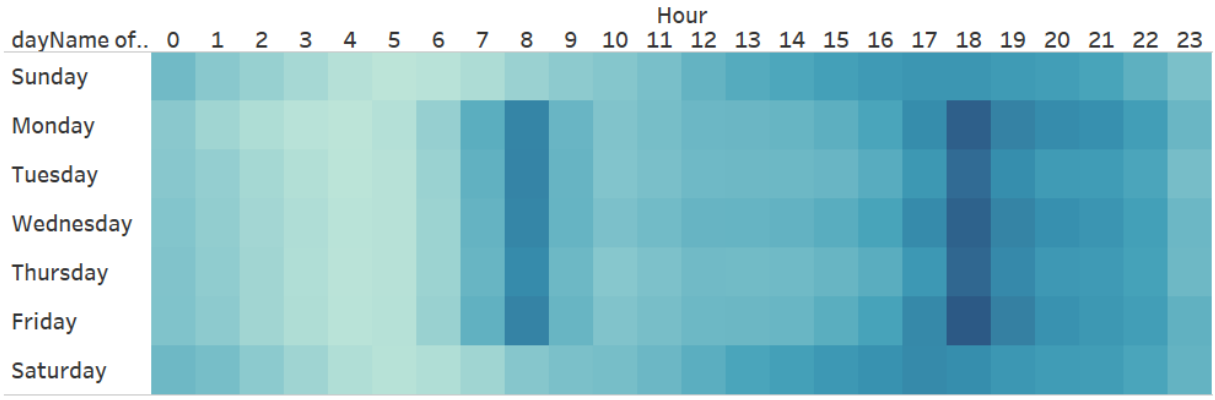

Figure 3.5: Hourly Distribution of Avg. Rental Counts

We can also infer from the visualisation that 7-8am and 5-9pm are peak hours on weekdays while the weekend crowd is much more distributed and spread out amongst the 2-11pm timeslots. Hence, having a promotional hour with discounted rental rates from 4-5:30pm on weekdays would be a good deal to disperse the crowds during the peak hour and increase overall rental counts. While the weekend promotions could include night-cycling discounts.

3.3.3 Seasonal Recommendations

From both clustering analysis and seasonal variations dashboard, we know that Summer has the most bike rental counts, while Winter has the least. This might be due to the fact that challenges faced when cycling in winter are manifold. Snow, ice, and rain can turn road surfaces into slippery slides.

During the peak Summer season, implementing a dynamic pricing strategy, by charging higher prices during peak hours and reducing the prices during non-peak hours might help in attracting more people while at the same time maximizing the revenue for the company.

To promote biking in winter, the company should increase awareness about winter cycling. Ensuring the public is aware of the risks but also the benefits associated, such that they are able to make informed decisions on the suitability of this form of transport for them. Training on winter cycling and more frequent bike maintenance should be provided. Besides that, maintaining the safety and quality of the equipment, routes and facilities are also important considerations and measures that should be taken by the company (Chapman & Larsson, 2021).

3.4 Limitations

In this project, we gained access to only 1 year's worth of data, dated 2018. The percentage error is 19.5% for demand prediction and 24.9% for predictive maintenance. While the error is still at an acceptable range, it could be well-improved using the data from 2019 to 2021.

Another limitation is the simulation of the number of bikes needing repair. In order to solve this limitation, the logistic team can easily record the number of bikes retrieved for maintenance over every day. This data can be added to our dataset to fine-tune our model regularly.

3.5 Conclusion

In this project, we proposed a method to predict the number of bikes needed in each location for every day in the upcoming 7 days. The model takes into account the number of bikes that will be rented out and the number of bikes that have to be replaced for maintenance. The numbers are buffered according to the RMSE. Using this method, we are able to generate USD11,266.34 for the last 7 days in our dataset, while 2 other possible strategies fail to generate profit.

We have recognised that location of the bike rental offices plays an active role on the bike rental demands and have analysed the location preferences of Seoul residents. We identified the most and least popular locations using the tableau dashboard and have proposed methods to increase revenue by closing down some of the rental offices. The impacts of seasonal and weather

parameters were also studied to understand the pattern of bike ridership in Seoul. The insights we gathered through this project would contribute significantly to future planning of bike-sharing systems in Seoul, and with the necessary and sufficient data it can be extended to other locations too.

References

Ardavan A., Shamsul S. & Mehdi H. (2020). The Systematic Review of K-Means Clustering Algorithm. In 2020 The 9th International Conference on Networks, Communication and Computing (ICNCC 2020). Association for Computing Machinery, New York, NY, USA, 13–18. DOI:https://doi.org/10.1145/3447654.3447657

Brownlee, J. (2014, September 26). Discover feature engineering, how to engineer features and how to get good at it. Machine Learning Mastery. https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/

Bill's bike and repair. (n.d.). Cost of Bike Tune-ups, Service & Repair. Bill's Bike and Run. Retrieved September 12, 2021, from https://www.billsbikeandrun.com/about/service-repair-pg63.htm

C. Revelle, D. Marks, J.C. Liebman. An analysis of private and public sector location models Manag. Sci., 16 (11) (1970), pp. 692-707

Chapman, D. & Larsson, A. (2021) Practical urban planning for win https://github.com/pablo14/shap-valuester cycling; lessons from a Swedish pilot study, Journal of Transport & Health, Volume 21, 2021, 101060, ISSN 2214-1405, https://doi.org/10.1016/j.jth.2021.101060.

CRICCHIO, M. (2020, November 10). How to rent bikes in Seoul: Bike ride on the Han river or just get from a to B. The Soul of Seoul. https://thesoulofseoul.net/2020/11/10/how-to-rent-bike-in-seoul/

Fredrick, H (n.d.). How Can the Weather Affect the Rate of Metal Rusting? https://www.hunker.com/12583637/how-can-the-weather-affect-the-rate-of-metal-rusting

Gonfalonieri, A. (2019). How to Implement Machine Learning For Predictive Maintenance. Towards Data Science. https://towardsdatascience.com/how-to-implement-machine-learning-for-predictive-maintenance-4633cdbe4860

Ko, J.T. (2021). Seoul's bike rental service hits over 3 million users, yet losses continue. The Korean Herald. http://www.koreaherald.com/view.php?ud=20210726000484

Lee, K (2015). Seoul Bike 'Ttareungi' will be in full operation starting tomorrow. CPBC News. http://www.cpbc.co.kr/CMS/news/view_body.php?cid=597923&path=201510

Manna, S. (2020). Deep Feature Synthesis / Introduction to Automated Feature Engineering. Medium. https://medium.com/@souvikmanna251/deep-feature-synthesis-introduction-to-feature-engineering-49438cdc143e

Molnar, C. (2021, September 14). 8.6 SHAP (SHapley Additive exPlanations) | Interpretable Machine Learning. Christoph Molnar. https://christophm.github.io/interpretable-ml-book/shap.html

Nunnally, J.C. and Bernstein, I.R. (1994), Psychometric theory, Ed. ke-3, McGraw-Hill, New York

Pallant, J. (2001), SPSS survival manual - a step by step guide to data analysis using SPSS for windows (version 10), Buckingham Open University Press.

Pazdan, S. (2020). The impact of weather on bicycle risk exposure. Archives of Transport. 56. 89-105. 10.5604/01.3001.0014.5629.

Partners, P. &. (2020, August 25). Top 3 major business districts (CBD) in Seoul, Korea. Pearson & Partners. https://pearsonkorea.com/insights/Top-3-Main-CBD-in-Seoul-Korea-%E2%80%93-Bringing-Best-Business-Practices-to-the-Table/

Peters, K. (2020). LSTM for predictive maintenance of turbofan engines. Towards Data Science. https://towardsdatascience.com/lstm-for-predictive-maintenance-of-turbofan-engines-f8c7791353f3

Runde Sache (2011). Readers Digest Deutschland (in German). 06/11: 74–75.

Savannah, E., & Karakoc, M. (2017). Seoul Cycle: Making Seoul a Bike-Friendly Destination. University of Delaware. https://cpb-us-w2.wpmucdn.com/sites.udel.edu/dist/4/10696/files/2018/01/2-20ymhsx.pdf

Seoul Bike Sharing Demand Data Set. (2020, March 1). Retrieved from UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand

Seoul Metropolitan Government, SMG (2016). Expanded Operation of Seoul Bike “Ddareungi”. http://english.seoul.go.kr/expanded-operation-seoul-bike-ddareungi/

Sharma, A. (2020, May 12). Decision Tree vs. Random Forest - Which Algorithm Should you Use? Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/

Singh, A. (2019, December 9). Feature engineering techniques for time series data. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2019/12/6-powerful-feature-engineering-techniques-time-series/

SMG. (2016, December 14). Ring ring! Enjoy riding through Seoul on a Seoul bike! -. Official Website of the. https://english.seoul.go.kr/ring-ring-enjoy-riding-seoul-seoul-bike/

Usage Information By Public Bicycle Rental Office In Seoul (Monthly). (2020, July 23). Retrieved from Seoul Open Data Plaza: http://data.seoul.go.kr/dataList/OA-15249/F/1/datasetView.do#

Yang, L. (2017, May 19). Gangnam business district in Seoul, South Korea. seouloffice. https://yka96110.wixsite.com/seouloffice/single-post/2017/05/19/major-business-districts-in-the-capital-area-of-south-korea-gbd

서울자전거 따릉이 - 무인대여시스템. (n.d.). Seoul Bike Rental Service. Retrieved September 14, 2021, from https://www.bikeseoul.com/info/infoCoupon.do