[20181321] [Kang Min Hyeok]

IE30301-Final Report

Knowledge of rental bike and sharing economy

With the recent activation of the sharing economy, the rental business is expanding in various fields. Vehicles such as automobiles and electric scooters are also part of the sharing economy, and sharing platforms are increasing worldwide. In particular, this project deals with the bicycle sharing system. People rent bicycles (casual way, resisted way). And it can be assumed that this rental figure is affected by season, date, weather, temperature, etc.

Problem Defining - rental bike and sharing economy

Demand forecasting is essential for efficient operation of the sharing economy platform. This is because, if the demand can be predicted, the number of bicycles placed at each bicycle rental site or per day can be efficiently adjusted.
So I try to predict the bicycle rental demand with the features of the given data.

My hypothesis

When will be count of total rental bikes('cnt') more?		
Time(hour)		
Commute time	>	**Working time**
Season		
Summer	>	**Winter**
Holiday vs Working day (similar – SUN, SAT vs others)		
Working day	>	**Holiday**
Weather		
Clear	>	**Rain or Cloudy**
Temperature or Feeling temperature		
higher	>	**Lower**
Humidity		
Lower	>	**Higher**
windspeed		
Slower	>	**Faster**

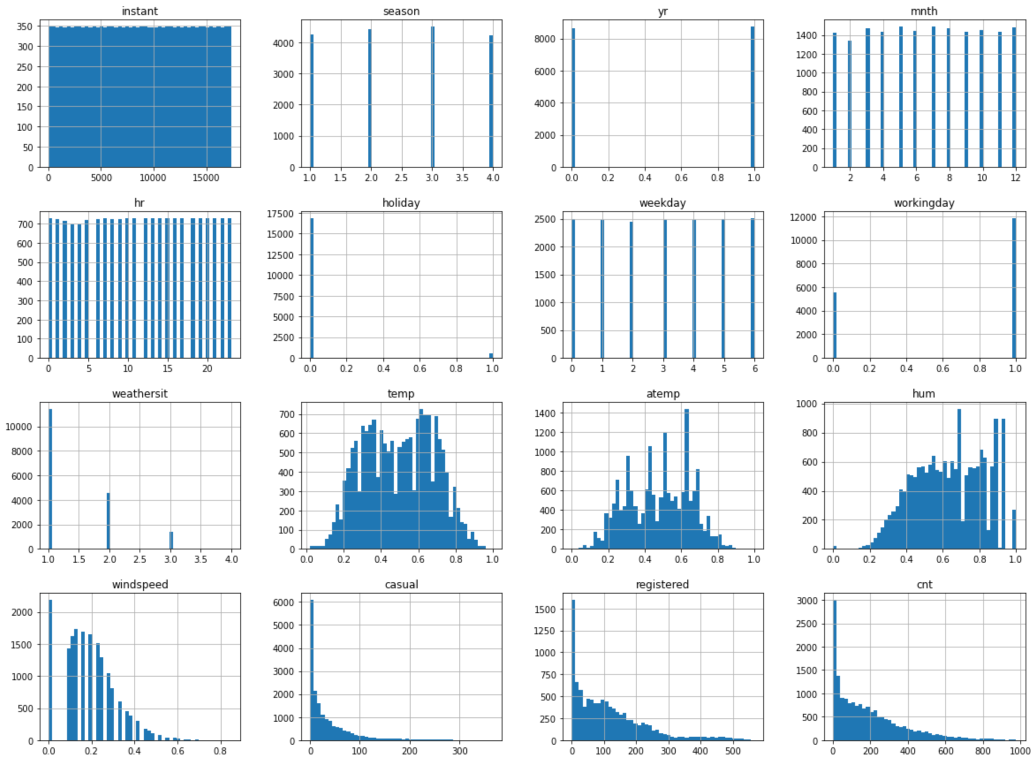
Time: Bicycles are meant to be a faster means of transport than walking. There will be more demand for bicycle rentals, especially during time-saving commute times.
Season: In winter it is windy and cold, so it is not suitable for cycling. So, rather, more people will rent bicycles in summer.
Holiday vs Working day: If the purpose of renting a bicycle is to move quickly, working days with busy commuting times will have more demand for bicycle rentals than holidays.
Weather: Rainy or cloudy days are not suitable for cycling. So on sunny days more people will rent bikes.
Temperature or Feeling temperature: The higher the temperature, the more people will rent a bike to get around cooler.
Humidity: The higher the humidity, the less sweat you will get while riding the bike. So the lower the humidity, the more people will be able to ride the bike comfortably. In other words, more people will rent bikes when the humidity is low.
Windspeed: The stronger the wind, the less suitable it is to ride a bicycle. This is because it is difficult to ride the bicycle stably when the wind is strong. So when the wind is slower, many people will rent bikes.
***Casual and Registered:** Looking at the given raw data, count of casual users and count of registered users cannot be a feature used to predict count of total rental bikes. Because 'cnt' = 'casual' + 'registered'. If casual features and registered features are used for 'cnt' prediction, it is cheating. Considering the actual situation, knowing the count of casual users and the count of registered users at the time means that you can know the count of total rental bikes by just adding the two values immediately. In conclusion, casual features and registered features will not be used for prediction of 'cnt' in my project, but rather, I will compare the counts of 'casual' and 'registered' users over time.

(1) Exploratory data analysis

* **TRAING_SIZE:** 80%
* **TEST_SIZE:** 20%
* **RANDOM_SEED:** 0
* **Features from given data**: dteday, season, yr, mnth, hr, holiday, weekday, workingday, weathersit,

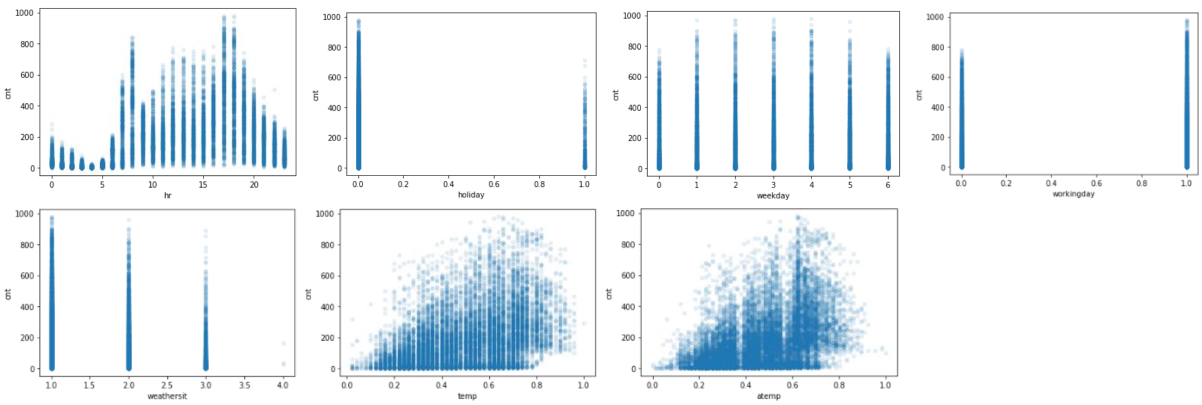
temp, atemp, hum, windspeed, casual, registered
* **Target from given data**: cnt

- Plot Histogram
Data: whole raw data



We can look at the data distribution of all features with a plot histogram.

- Scatter-Plot for indicating features
Data: training set
y-axis: cnt
x-axis: dteday, season, yr, mnth, hr, holiday, weekday, workingday, weathersit, temp, atemp, hum, windspeed, casual, registered

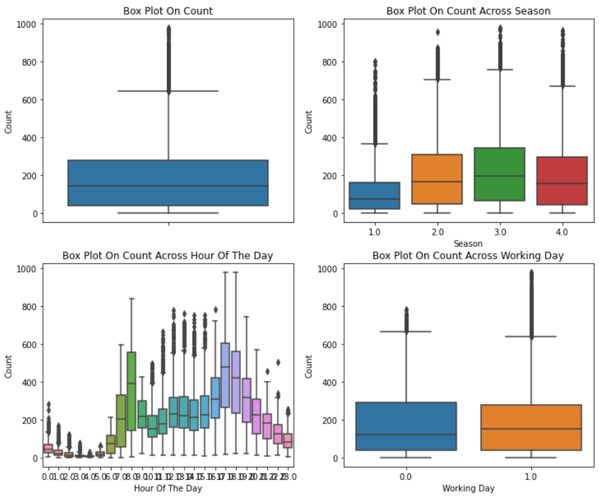


I drew a scatter-plot with 'cnt', the target we want to predict, and each feature on the y-axis and x-axis, respectively. Through this, you can examine the relationship between each feature and 'cnt'. The following is an analysis of features that can derive meaningful results.

hr: There are few bicycle rental users in the early hours of 0-6 o'clock, and the most rental users are at 7-8 o'clock and 17-19 hrs, when commuting.
holiday: Because it is not a holiday, there are more rental users than when it is a holiday.
weekday: There are more rental users on weekdays than on weekends.
workingday: There are more rental users than on days that are not working days.
weathersit: The clearer the weather, the more rental users.
temp & atemp: The higher the temperature, the more users generally rent bikes.

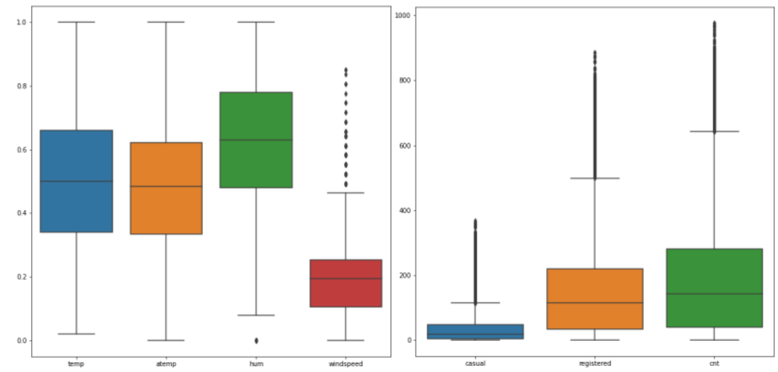
- Box-Plot for indicating features

Data: training set



cnt outliers for features in the training set

We can observe outliers through box plots and check the distribution of data. Let's first look at the 'cnt' outliers for features in the training set. First, there are many outliers in the 'cnt-only' box plot. If you check other box plots, this outlier is common on the 'workingday', and it occurs a lot between 10 and 16 o'clock. Afterwards, we plan to remove these outliers in data preprocessing.

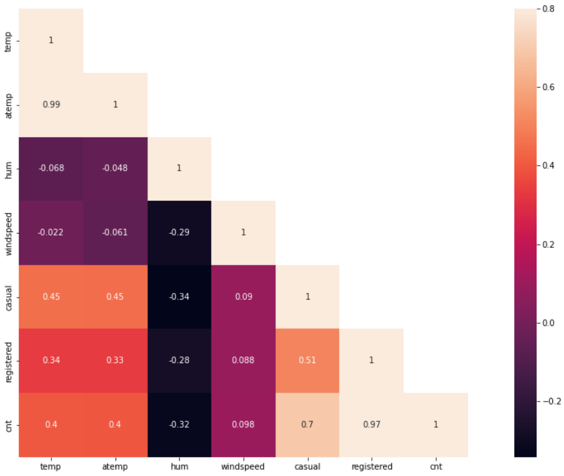


numeric features box plot

If you look at the box plot of numeric features this time, you can see that the already given raw data is normalized and the data distribution is even. However, in the case of 'windspeed', there are outliers.

- Correlation Analysis

Data: training set

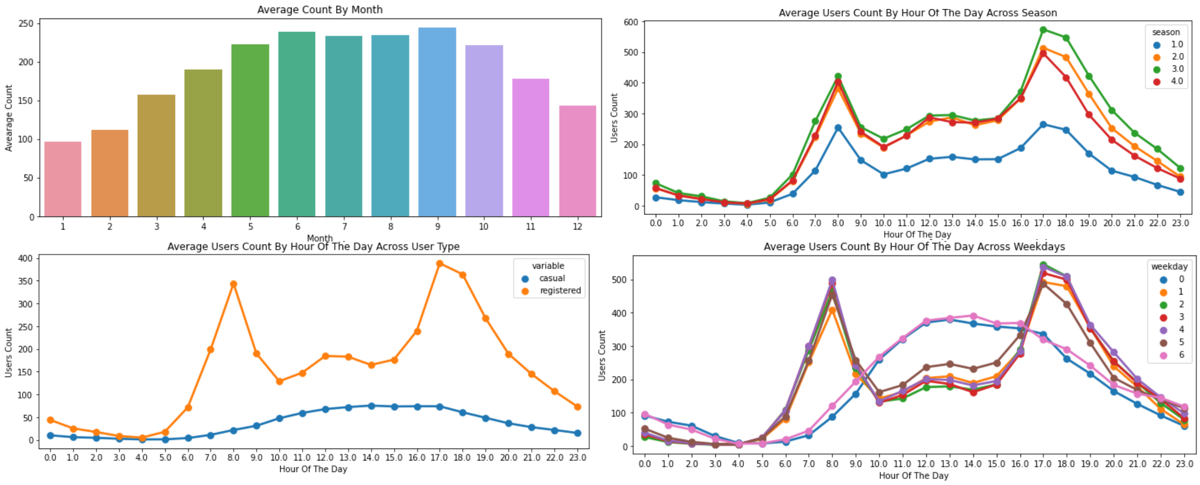


Correlation analysis is a technique to analyze the linear relationship between two variables measured as continuous variables. Therefore, correlation analysis is performed based on the continuous features 'temp', 'atemp', 'hum', 'windspeed', 'casual', 'registered' and 'cnt'. This analysis method indicates whether an increase in one variable linearly increases or decreases the other variable. Here we can see the correlation coefficient, which indicates the degree of linear relationship between two variables.

Looking at the correlation matrix, first, the correlation coefficient between 'registered' and 'cnt' is very high at 0.97, and ‘casual’ and ‘cnt’ are also very high at 0.7. This is because 'casual' + 'registered' = 'cnt', **<u>so 'casual' and 'registered' features will be excluded</u> from the data to be included in the later model.** **If you use these two features to predict cnt, that's because it's actually cheating.**

And looking at the correlation between ‘cnt’ and other features, it can be seen that 'cnt' shows a significant correlation with 'temp', 'atemp', and 'hum', and 'temp' and 'atemp' have a positive correlation. , 'hum' has a negative correlation.

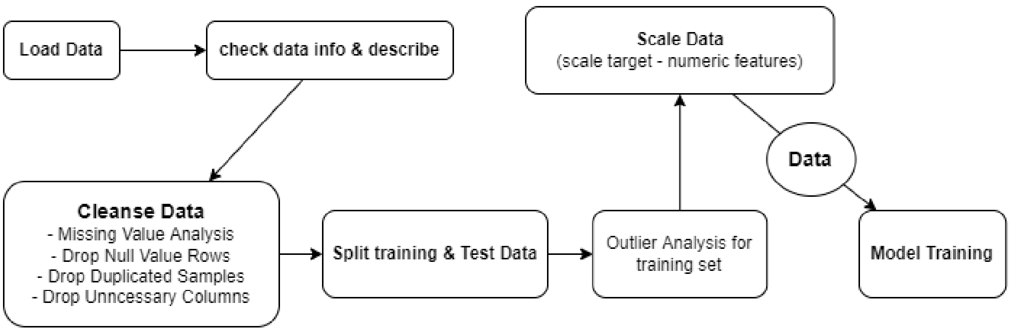
<u>- Plot Analysis</u>



First, the first plot shows **"average count by month".** Generally, it shows more 'cnt' in May-September, which is the hottest season of the year. And the second plot shows **"Average Users count by hour of the day across seasons"**. Looking at this plot, it can be seen that the demand for bicycle rentals is generally low in spring, and the demand for bicycles is high in the rest of the season. And the third plot shows **"Average Users count by hour of the day across user type"**. 'registered' users are generally more numerous than 'casual' users, and in particular, they tend to rent bicycles more at 7-8 o'clock and 17-18 hrs. The last plot shows **"Average Users count by hour of the day across weekdays"**. If you look at this plot, you can see that the pattern from Monday-Friday and the pattern from Saturday-Sunday are very different. On Saturdays and Sundays, I rent a lot of bicycles between 10 and 17 o'clock rather than commuting time.

(2) Preprocessing

<u>Flow chart</u>



Load Data

raw data file: regression_project.csv

Check data info & describe

	instant	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp
count	17376.000000	17376.000000	17376.000000	17376.000000	17376.000000	17376.000000	17376.000000	17376.000000	17376.000000	17376.000000
mean	8690.026876	2.501784	0.502532	6.538214	11.543854	0.028775	3.004431	0.682666	1.425357	0.496994
std	5017.010872	1.106890	0.500008	3.438710	6.915858	0.167179	2.005871	0.465452	0.639388	0.192540
min	1.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.020000
25%	4345.750000	2.000000	0.000000	4.000000	6.000000	0.000000	1.000000	0.000000	1.000000	0.340000
50%	8689.500000	3.000000	1.000000	7.000000	12.000000	0.000000	3.000000	1.000000	1.000000	0.500000
75%	13034.250000	3.000000	1.000000	10.000000	18.000000	0.000000	5.000000	1.000000	2.000000	0.660000
max	17379.000000	4.000000	1.000000	12.000000	23.000000	1.000000	6.000000	1.000000	4.000000	1.000000

	atemp	hum	windspeed	casual	registered	cnt
count	17376.000000	17376.000000	17376.000000	17376.000000	9999.000000	17376.000000
mean	0.475791	0.627337	0.190020	35.666034	116.692569	189.422364
std	0.171822	0.192913	0.122341	49.287257	110.576429	181.384969
min	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	0.333300	0.480000	0.104500	4.000000	27.000000	40.000000
50%	0.484800	0.630000	0.194000	17.000000	91.000000	142.000000
75%	0.621200	0.780000	0.253700	48.000000	169.000000	281.000000
max	1.000000	1.000000	0.850700	367.000000	567.000000	977.000000

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17382 entries, 0 to 17381
Data columns (total 17 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   instant     17376 non-null  float64
 1   dteday      17376 non-null  object
 2   season      17376 non-null  float64
 3   yr          17376 non-null  float64
 4   mnth        17376 non-null  float64
 5   hr          17376 non-null  float64
 6   holiday     17376 non-null  float64
 7   weekday     17376 non-null  float64
 8   workingday  17376 non-null  float64
 9   weathersit  17376 non-null  float64
 10  temp        17376 non-null  float64
 11  atemp       17376 non-null  float64
 12  hum         17376 non-null  float64
 13  windspeed   17376 non-null  float64
 14  casual      17376 non-null  float64
 15  registered  9999 non-null   float64
 16  cnt         17376 non-null  float64
dtypes: float64(16), object(1)
memory usage: 2.3+ MB
```

Looking at data info and description for data preprocessing, first, the count of registered features is 9999, unlike other features. It is necessary to check this part, and then if you look at the RangeIndex, there are 17382 indexes, and the non-null count of the actual features is 17376. That is, since 6 rows are rows with null values, they also need to be dropped.

Cleanse Data

1) Missing Value Analysis

Looking at the data info above, the number of registered feature data is 9999, which is less than other data. That is, the data is missing. However, since we know the number of casual users and the total count, subtract the number of casual users from the total count to get the number of registered users. We can fill in the missing data this way.

```
df['registered'] = df.apply(lambda x: x['cnt'] - x['casual'], axis = 1)
df['registered'].tail()
```

```
17377    48.0
17378    37.0
17379    13.0
17380    13.0
17381    13.0
Name: registered, dtype: float64
```

Through the above process, we can now check that the missing values of registered feature are filled in.

2) Drop null value rows

100	#######	1	0	1	8	0	3	1	1	0.2	0.1818
102	#######	1	0	1	10	0	3	1	1	0.22	0.197
103	#######	1	0	1	11	0	3	1	1	0.26	0.2273
104	#######	1	0	1	12	0	3	1	1	0.26	0.2273

Above table is raw data table, we should check null value column and drop.

```
df.shape
```

(17382, 17)

```
# Drop missing rows
df = df.dropna()
```

```
df.shape
```

(17376, 17)

3) Drop Duplicated Samples

```
df.shape
```

(17376, 17)

```
df.tail()
```

	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	tem
17377	17378.0	2012-12-31	1.0	1.0	12.0	22.0	0.0	1.0	1.0	1.0	0.2
17378	17379.0	2012-12-31	1.0	1.0	12.0	23.0	0.0	1.0	1.0	1.0	0.2
17379	1.0	2011-01-01	1.0	0.0	1.0	0.0	0.0	6.0	0.0	1.0	0.2
17380	1.0	2011-01-01	1.0	0.0	1.0	0.0	0.0	6.0	0.0	1.0	0.2
17381	1.0	2011-01-01	1.0	0.0	1.0	0.0	0.0	6.0	0.0	1.0	0.2

```
df = df.drop_duplicates()
df.shape
```

(17373, 17)

```
df.tail()
```

	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	tem
17374	17375.0	2012-12-31	1.0	1.0	12.0	19.0	0.0	1.0	1.0	2.0	0.2
17375	17376.0	2012-12-31	1.0	1.0	12.0	20.0	0.0	1.0	1.0	2.0	0.2
17376	17377.0	2012-12-31	1.0	1.0	12.0	21.0	0.0	1.0	1.0	1.0	0.2
17377	17378.0	2012-12-31	1.0	1.0	12.0	22.0	0.0	1.0	1.0	1.0	0.2
17378	17379.0	2012-12-31	1.0	1.0	12.0	23.0	0.0	1.0	1.0	1.0	0.2

In this process, duplicate samples are removed. As shown in the figure above, the overlapping part disappeared and three rows were dropped.

4) Drop unnecessary columns

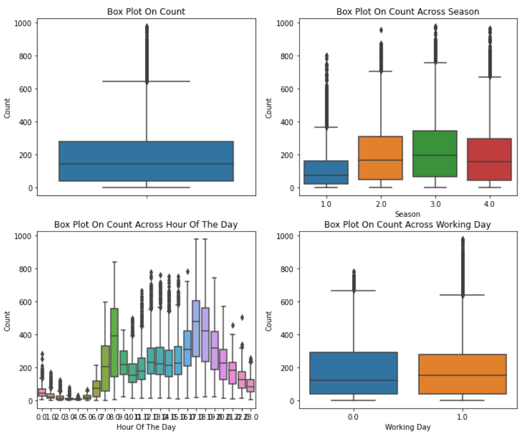
```
# Drop columns
df = df.drop(columns=['instant', 'dteday', 'yr'], axis=1)
```

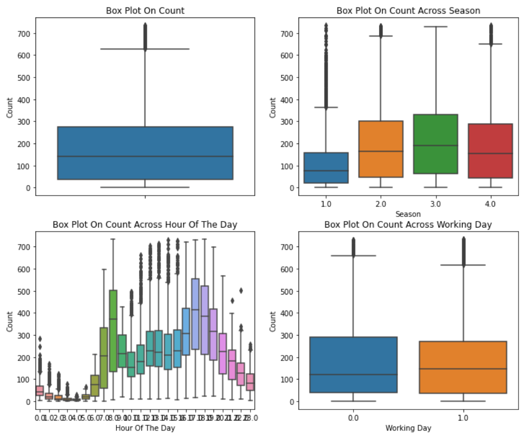
In my assumption, the 'dteday' feature and 'yr' feature are not used for model training and analysis. The 'instant' feature is unnecessary because the index number is provided in the pandas dataframe, and the 'dteday' feature only shows the date and does not contain a specific meaning like 'weekday' or 'holiday', so it was judged to be an unnecessary column. , 'yr' is also a dropped column for the same reason.

- Split Training & Test Data

To divide the training set and the test set, sklearn's train_test_split is used, and Random_state is 0. The proportion of the test set is 20%, the total number of training samples is 13898, and the number of test samples is 3475.

- Outlier Analysis






```
Training set shape: (13898, 14)
```

(13702, 14)

For outlier analysis in EDA, we looked at the distribution of data using box plots. Therefore, in this process, data preprocessing was performed to remove outliers.

- Scale Data

Scaling target features(numeric): ‘temp’, ‘atemp’, ‘hum’, ‘windspeed’
Data scaling was performed on numeric features. First, we separated categorical and numeric variables from the training set, scaled the numeric variables, and then put them back together. At this time, scaling was performed using sklearn's standardScaler.

(3) Model train & test

- Simple Linear Regression

X features: ‘temp’, ‘atemp’, ‘hum’, ‘windspeed’
In simple linear regression, as the result of Correlation Analysis of EDA, regression is performed on features that have a linear relationship with 'cnt'. According to the correlation matrix, 'temp' and 'atemp' have a positive correlation with 'cnt' and 'hum' have a negative correlation, so this model was used to confirm this. The model was trained using sklearn's LinearRegression.

- Multiple Linear Regression

X features: ‘season’, ‘mnth’, ‘hr’, ‘holiday’, ‘weekday’, ‘workingday’, ‘weathersit’, ‘temp’, ‘atemp’, ‘hum’, ‘windspeed’
In multiple linear regression, all preprocessed features are used. As explained above, 'casual' and 'registered' are not used for model training. The model was trained using sklearn's LinearRegression.

- Random Forest Regression

X features: ‘season’, ‘mnth’, ‘hr’, ‘holiday’, ‘weekday’, ‘workingday’, ‘weathersit’, ‘temp’, ‘atemp’, ‘hum’, ‘windspeed’
parameter tuning-GridSearchCV: n_estimator, max_depth
GridSearchCV of sklearn is used to tune n_estimator and max_depth, which are parameters necessary for learning sklearn's Random Forest Regression model. GridSearchCV finds the optimal parameter value by sequentially applying the specified parameters based on cross-validation. The following are the best parameters derived through GridSearchCV.

Parameter	Compared Range	Best value	Best score	k-fold
n_estimators	100~500	500	0.85055	3
max_depth	5~20	20		

- Gradient Boosting Regression

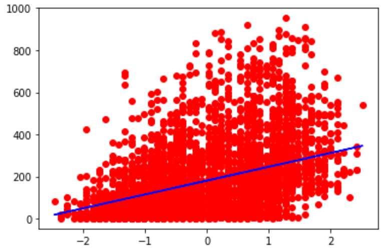
X features: ‘season’, ‘mnth’, ‘hr’, ‘holiday’, ‘weekday’, ‘workingday’, ‘weathersit’, ‘temp’, ‘atemp’, ‘hum’, ‘windspeed’
parameter tuning-GridSearchCV: loss, learning_rate, n_estimator
GridSearchCV of sklearn is used to tune loss, learning_rate and n_estimator, which are parameters necessary for learning sklearn's Gradient Boosting Regression model. GridSearchCV finds the optimal parameter value by sequentially applying the specified parameters based on cross-validation. The following are the best parameters derived through GridSearchCV.

Parameter	Compared Range	Best value	Best score	k-fold
n_estimators	100~500	500	0.83827	3
learning_rate	0.0001~0.1	0.1		
loss	'squared_error', 'absolute_error', 'huber', 'quantile'	'squared_error'		

(4) Result

- Simple Linear Regression

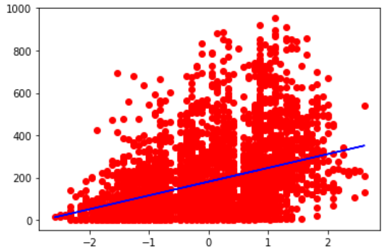
1) 'temp'



$$y = 180.83842 + 65.64906x_{temp}$$

```
coefficient:  [65.64905837]
intercept:  180.83841774923368
score:  0.15567717994474162
MAE: 122.4870
MSE: 26976.8923
RMSE: 164.2464
```

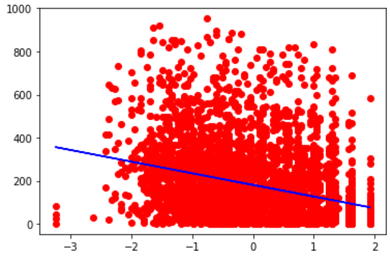
2) 'atemp'



$$y = 180.83842 + 65.04547x_{atemp}$$

```
coefficient:  [65.04546973]
intercept:  180.83841774923368
score:  0.15282769467572588
MAE: 122.6763
MSE: 27023.0725
RMSE: 164.3870
```

3) 'hum'



$$y = 180.83842 - 53.94041x_{hum}$$

```
coefficient:  [-53.94041403]
intercept:  180.83841774923368
score:  0.10509849684457395
MAE: 128.4986
MSE: 29182.8700
RMSE: 170.8299
```

- Multiple Linear Regression

feature: 'season', 'mnth', 'hr', 'holiday', 'weekday', 'workingday', 'weathersit', 'temp', 'atemp', 'hum', 'windspeed'

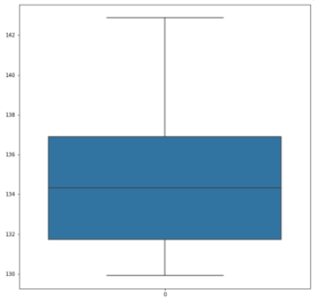
$$y = 53.45121 + 17.04548x_{season} - 0.00447968x_{mnth} + 7.17823x_{hr} - 20.58881x_{holiday} + 2.21848x_{weekday} - 5.94139x_{workingday} + 0.41417x_{weathersit} + 11.40458x_{temp} + 40.0815x_{atemp} - 39.99060x_{hum} + 3.62976x_{windspeed}$$

Internal evaluation

> without cross-validation

```
R2 Score: 0.3439352719441101, RMSE: 134.7688
```

> with cross-validation (10-fold)



```
RMSE CV Scores:
[137.05244353 137.65922877 134.42153501 142.87681944
136.40252996 130.78020622 134.25408078 133.89737514
129.91739533 131.00689729],
Mean: 134.8269,
Std: 3.6971
```

External evaluation

```
R2 Score: 0.3247586481326067, RMSE: 147.8072
```

- Random Forest Regression

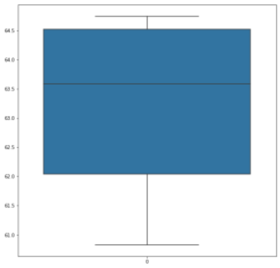
feature: 'season', 'mnth', 'hr', 'holiday', 'weekday', 'workingday', 'weathersit', 'temp', 'atemp', 'hum', 'windspeed'

Internal evaluation

> without cross-validation

```
R2 Score: 0.9770606715467265, RMSE: 25.2003
```

> with cross-validation (10-fold)



```
RMSE CV Scores:
[64.63745725 62.8440545  63.76287154 64.80916363
64.47035348 61.94338465 64.52942696 61.92461187
60.62517353 62.55777868],
Mean: 63.2104,
Std: 1.3685
```

External evaluation

```
R2 Score: 0.8525768486352798, RMSE: 69.0636
```

- Gradient Boosting Regression

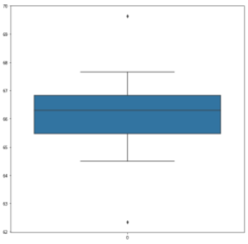
feature: 'season', 'mnth', 'hr', 'holiday','weekday','workingday','weathersit', 'temp', 'atemp', 'hum', 'windspeed'

Internal evaluation

> without cross-validation

```
R2 Score: 0.8597231802717618
MAE: 43.5740, MSE: 3883.4521, RMSE: 62.3174
```

> with cross-validation (10-fold)



```
RMSE CV Scores:
[66.94439161 66.49176595 66.42062635 69.69999084
66.15597694 65.59905713 67.66657117 65.41949735
62.34946746 64.44402967]
Mean: 66.1191
Std: 1.8413
```

External evaluation

```
R2 Score: 0.8301537481626672, RMSE: 74.1300
```

Regression Model	Internal Evaluation				External Evaluation	
	R2 Score	RMSE	RMSE CV(10) Score		R2 Score	RMSE
			Mean	Std		
Multiple Linear Regression	0.34394	134.7688	134.8269	3.6971	0.32476	147.8072
Random Forest Regression	**0.97706**	**25.2003**	**63.2104**	**1.3685**	**0.85257**	**69.0636**
Gradient Boosting Regression	0.85972	62.3174	66.1191	1.8413	0.83015	74.13

(5) Discussion & Conclusion

When three regression models (multiple linear regression, random forest regression, gradient boosting regression) were used, the one with the highest performance was Random Forest regression.

1) Insights from results

- First, looking at the results obtained through simple linear regression, it can be seen that 'temp' and 'atemp' have a positive linear relationship with 'cnt', and 'hum' has a negative linear relationship with

'cnt'. Through this, it can be inferred that the higher the temperature, the higher the number of bicycle rental users, and the lower the humidity, the smaller the number of bicycle rental users.
- Looking at the coeficients obtained through multiple linear regression, like simple linear regression, 'temp' and 'temp' have positive influence and 'hum' have negative influence. This makes the conclusion in simple linear regression more reliable. And looking at the other coefficients, it can be seen that 'holiday', 'season', 'hr', and 'workingday' show a strong correlation with bicycle rental users.

2) Insight to discover the differences between each model

It is difficult to know how each feature affected the most performing Random Forest Regression, but when looking at the given initial raw data and EDA results, the number of bicycle rental users is clearly distinguished by season, holiday, and time. For this reason, due to the nature of the decision tree type model that is predicted separately by node, the branch of the data became clear, and a higher-performance regression model could be created. In the case of multiple linear regression, there are many features, and it can be seen that insignificant features generate noise, resulting in low performance. If feature extraction such as PCA is applied or feature selection is applied, it will show better predictive power. In addition, it can be seen that Gradient Boosting Regression shows good predictive performance, which is suitable for bicycle rental data with clear data distinction according to the value of feature because the predictive model is formed with branches by the decision tree like random forest regression. In conclusion, it can be determined that the decision tree model is the most appropriate model for the prediction of bicycle rental users.

3) Hypothesis validity and improvement

When will be count of total rental bikes('cnt') more?			Validity
Time(hour)			**Valid**
Commute time	**>**	**Working time**	
Season			**Valid**
Summer	**>**	**Winter**	
Holiday vs Working day (similar – SUN, SAT vs others)			**Valid**
Working day	**>**	**Holiday**	
Weather			**Valid**
Clear	**>**	**Rain or Cloudy**	
Temperature or Feeling temperature			**Valid**
higher	**>**	**Lower**	
Humidity			**Valid**
Lower	**>**	**Higher**	
windspeed			**Invalid**
Slower	**>**	**Faster**	

The table on the left is an initial hypothesis. First, according to the Plot Analysis results from EDA, it was confirmed that 'commuting time (7-8, 17-18)' had more rental users than 'working time'. Similarly, according to the Plot Analysis results from EDA, more people rent bicycles in 'summer' than 'winter'. And according to Box plot analysis and plot analysis in EDA, more people rent bicycles on 'working day' than on 'holiday'. Looking at the multiple linear regression coefficients, 'weathersit' has a low coefficient. Since 'weathersit' is closer to 0, the weather is clearer, so the low coefficient of 'weathersit' means that more rental users occur when the weather is sunny. Looking at the results of correlation analysis of EDA and the results of linear regression, it can be seen that bicycle rental users increase as 'temp' increases. Conversely, it can be seen that the lower the 'hum', the more bicycle rental users. However, in the case of 'windspeed', according to correlation analysis, it can be confirmed that there is little correlation with the number of bicycle rental users.
We can make better hypotheses and lead to more accurate analysis with the results obtained from this project. Through our regression model, we can obtain quantitative linear relationships or branching nodes, as well as positive or negative correlations between features and 'cnt'. For example, in the initial hypothesis, it was simply predicted that rental users would increase if the temperature was high. Because this project looked closely at the data through EDA and created a regression model using various models, the hypothesis could be improved.

Reference

V. (2017, April 23). *EDA & Ensemble Model (Top 10 Percentile)*. Kaggle. https://www.kaggle.com/code/viveksrinivasan/eda-ensemble-model-top-10-percentile