Usman Zafar

ID : st20194302

Programming for Data Analysis : CIS7031_S2_20

Seoul South Korea Rental Bike Sharing

TABLE OF CONTENT :

1. Introduction
2. Objective
3. Data preparation and exploration
4. Data Visualization
5. Analysis and Interpretation of Evaluation Results
6. Comparison of all models
7. Critical review and analysis of techniques used
8. Reflection upon the project experience
9. Conclusion

1. Introduction:

Seoul is a modern city with an ever increasing population. In order to combat the issue of congestion in traffic and to make intercity commuting more efficient we would be analyzing the past figures of ‘Seoul Bike Rental’.

2. Objective of Report:

The objective of the report is to identify and analyze the factors which significantly impact bike usage in Seoul city and accordingly make the most suitable machine learning model to accurately predict the demand of bikes in the future.

3. Data Preparation and Exploration:

The data taken has been imported from UCI Machine Learning Repository. The data set contains information on the number of bikes rented against various weather indicators. We have the rental bikes data of one year from 01 December 2017 till 30 November 2018. Data contains 14 attributes(variables) and has 7260 rows.All the data mining and analyses would be done in Python.

Data Cleaning:

We have only changed the format of the date to “yyyy-mm-dd”.

```
bikes['Date']=pd.to_datetime(bikes['Date'])
```

Furthermore, we have 13 dates in which bike rentals were closed and it's classified as ‘Non-Functioning Day’ in the data set. We need to remove them as well.

After careful examination of the data, no null values and no missing values were found in the dataset.

Data Exploration:

The fields in the dataset include ID, date, hour, temperature, humidity, wind speed, visibility, dew point, solar radiation, rainfall, snowfall, season, is-holiday, is-functioning day and bikes rented.

Data Codes Explanation:

```
bikes = pd.concat(
map(pd.read_csv, ['train.csv', 'test.csv']), ignore_index=True)
```

This code will merge the two CSV files that are ‘training’ and ‘testing’ datasets.

```
bikes
ax2 = bikes.plot.scatter(x='Temperature',
                      y='Bikes_Rented',
                       colormap='viridis',figsize=(15,10))
```

It is used for showing the data frame

```
bikes["Season"].value_counts().
```

This code counts each season instance ensuring that data is balanced.

```
bikes.describe()
```

This gives us all the statistical information about our dataset.

```
%matplotlib inline
import matplotlib.pyplot as plt
bikes.hist(bins=50, figsize=(20,15))
plt.show()
```

This shows us the visualization of the attributes in the dataset.

```
bikes=bikes[~(bikes['IsFunctioningDay'] == 'No')]
bikes=bikes.drop("IsFunctioningDay",axis=1)
bikes=bikes.drop("Id",axis=1)
```

We are removing **'ID'** from the data frame because it does not have statistical significance in making our prediction model.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
bikes_num_std=scaler.fit_transform(bikes_num)
```

StandardScaler removes the mean and scales each feature/variable to unit variance. This operation is performed feature-wise in an independent way.

```
bikes_num_df=pd.DataFrame(bikes_num_std,columns=['Hour',
'Temperature(°C)', 'Humidity(%)',
       'Wind speed (m/s)', 'Visibility (10m)', 'Dew point
temperature(°C)',
       'Solar Radiation (MJ/m2)', 'Rainfall(mm)', 'Snowfall
(cm)','IsHoliday'],dtype='float64')
```

This code loads the column names into the data frame.

```
from sklearn.preprocessing import OneHotEncoder

cat_encoder = OneHotEncoder()
bikes_cat_OH = cat_encoder.fit_transform(bikes_cat)
bikes_cat_OH.toarray()
```

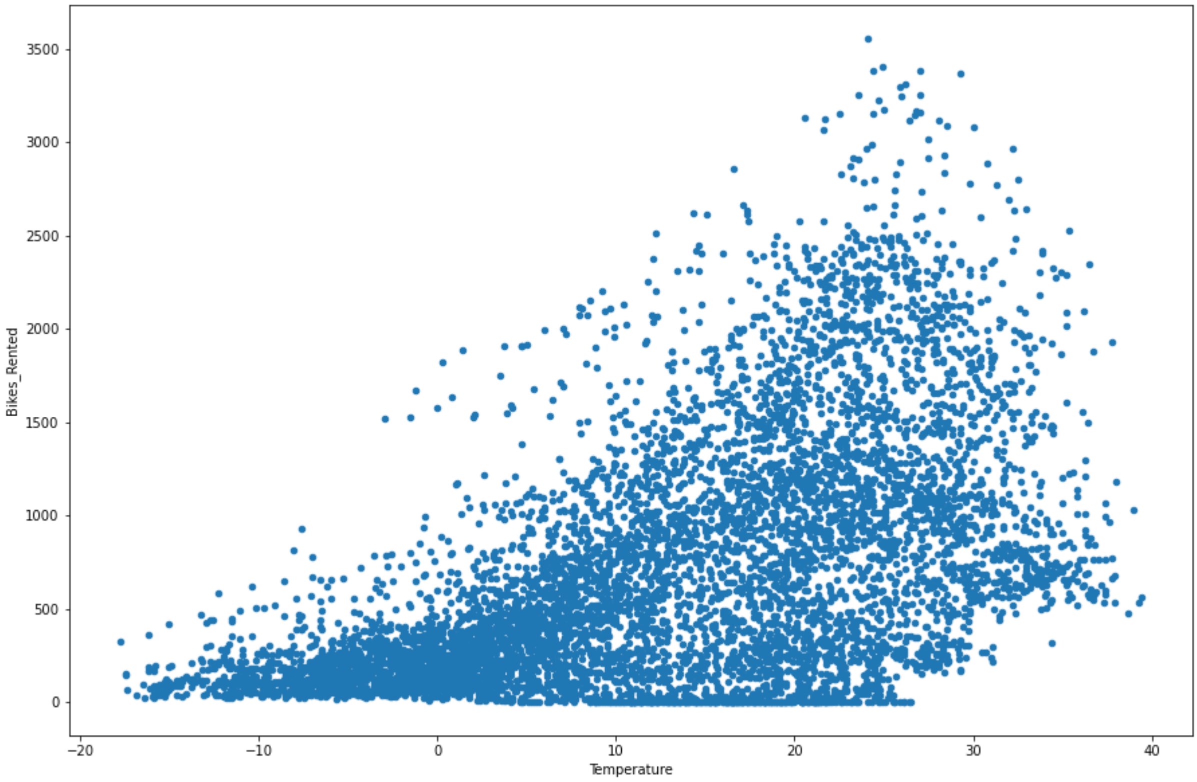
```
array([[0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       ...,
       [1., 0., 0., 0.],
       [1., 0., 0., 0.],
       [1., 0., 0., 0.]])
```

We need to convert all our data into numerical form so that machine learning algorithms can be implemented efficiently.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
```

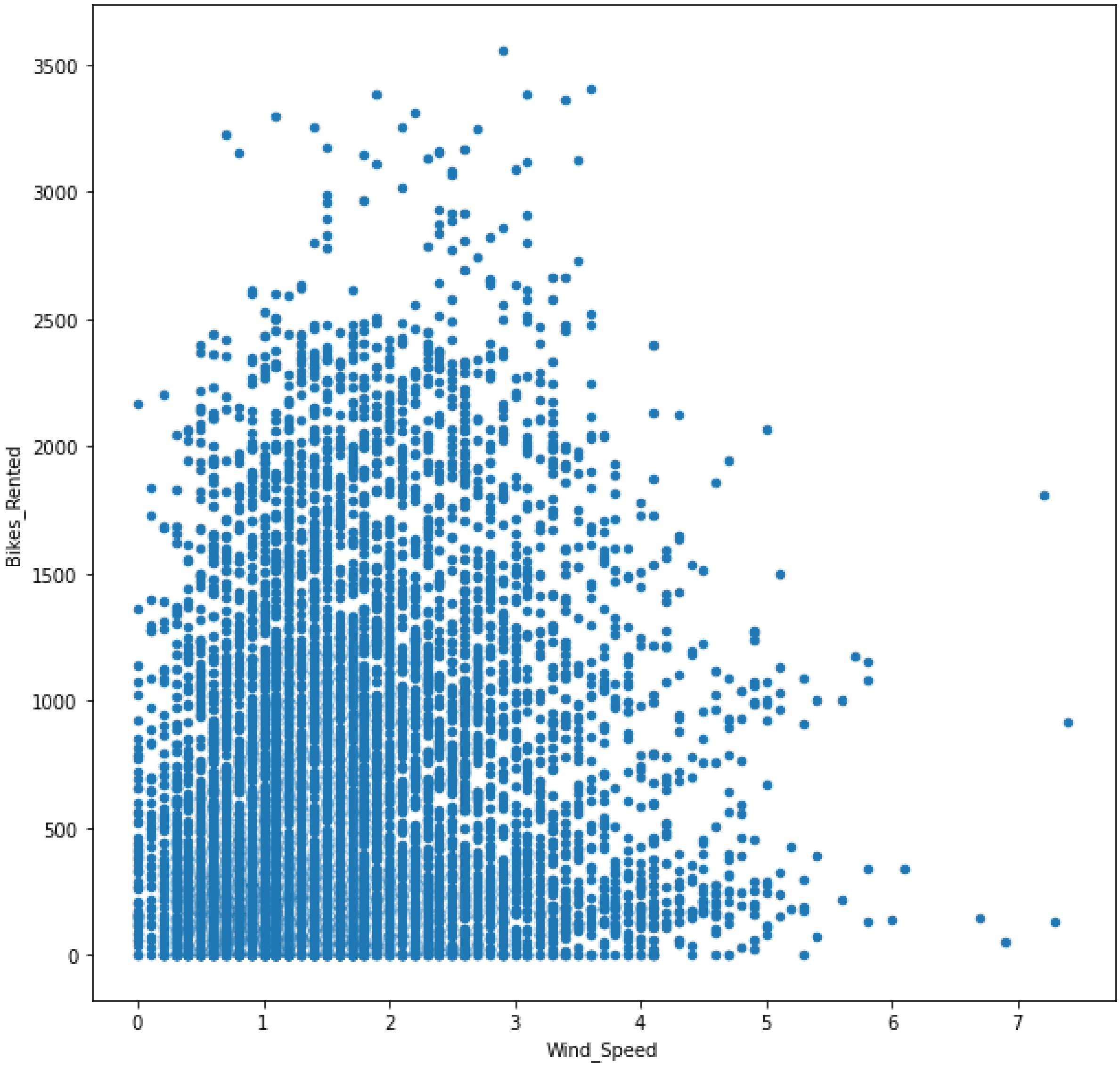
We would be splitting our data into two parts. To check the reliability of our model, we need to split our dataset into training and testing datasets. In this way we can make our model using ‘training’ data while to check the accuracy of our model we can use the ‘testing’ data.

4. Data Visualization:



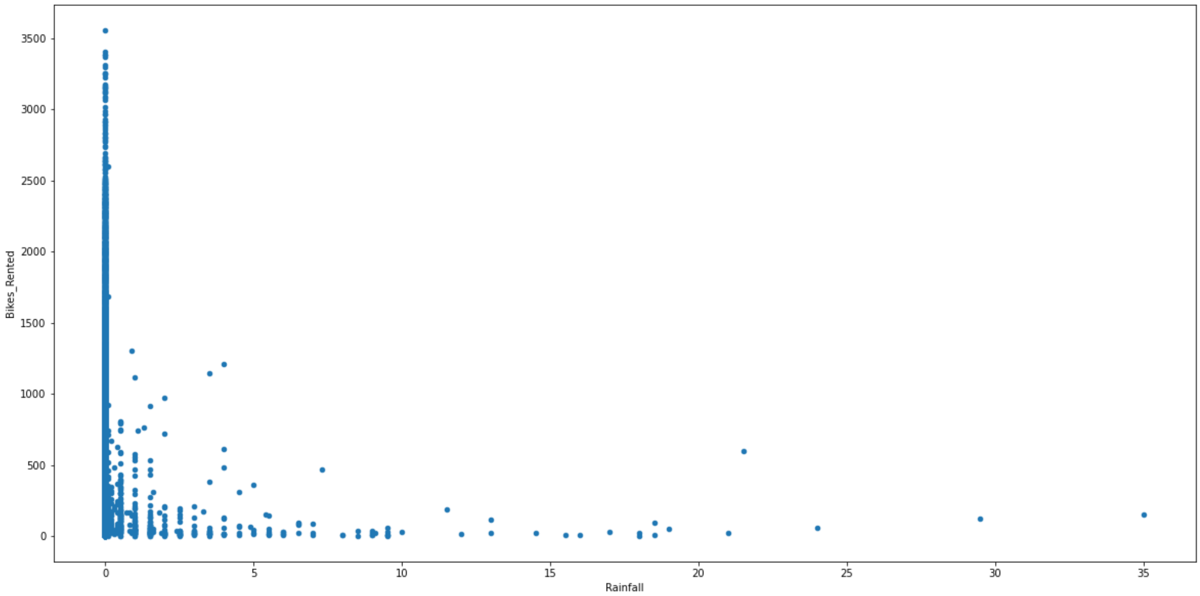
(Figure 1)

Figure 1 shows the spread of bikes rented at various temperatures. We can see that the number of bike users is directly proportional to warmer temperature while bike users start decreasing as temperature falls below zero degrees. The highest density of the spread was when temperature was between 0 degrees and 10 degrees . While between temperatures of 10 degrees to 20 degrees the number of bikers significantly increased. There are few outliers as well which might be users who needed the bike in case of some emergency.



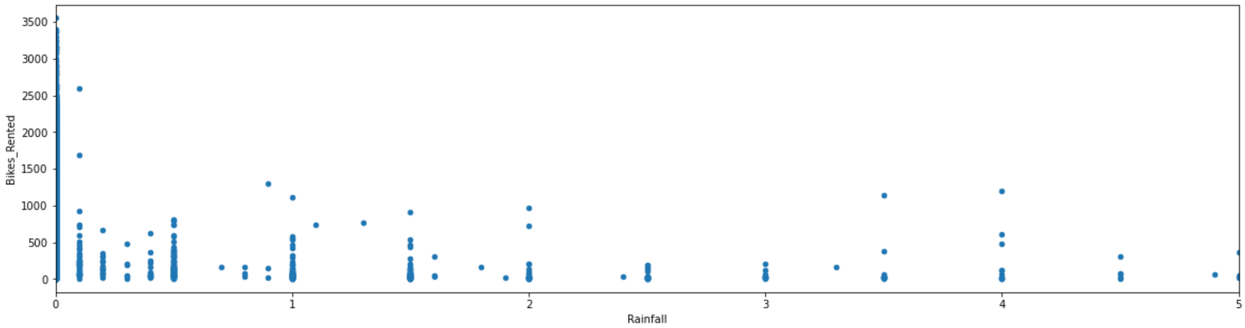
(Figure 2)

Figure 2 shows the relationship between bike users and wind speed. More people use bikes when wind speeds are between 0 m/s to 3 m/s which shows biker's preference is to rent bikes when wind is less.



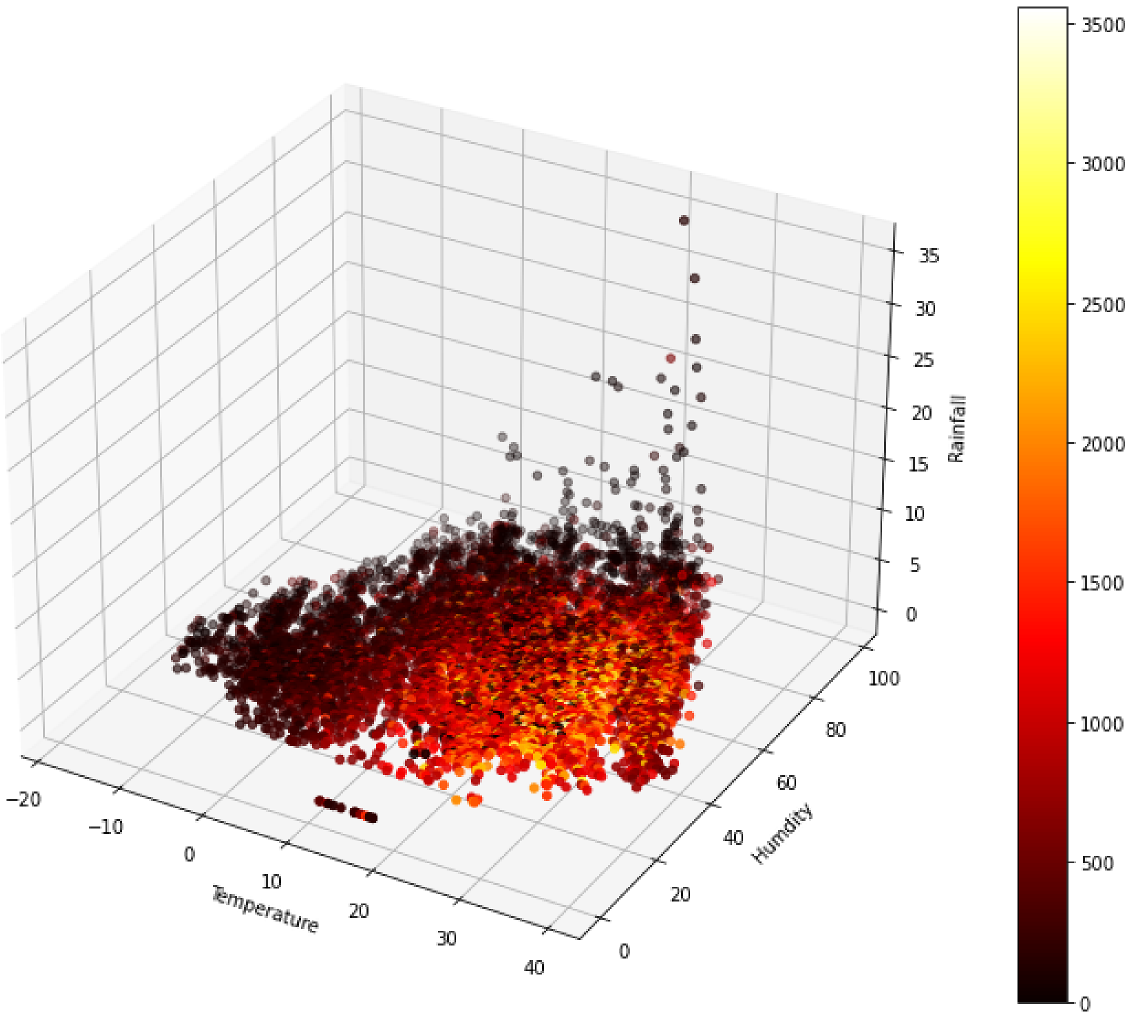
(Figure 3)

Figure 3 clearly shows that the majority of the people only rent a bike when it is not raining while only a minority of people use it during rain.



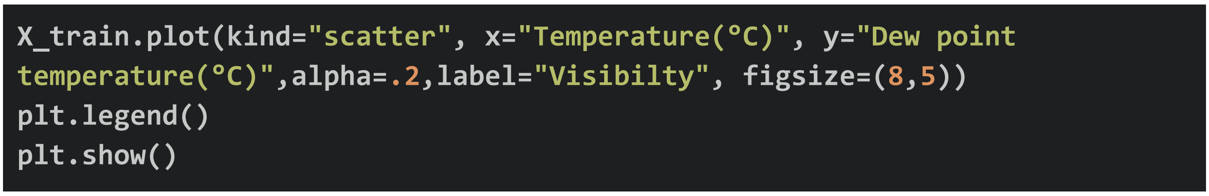
(Figure 4)

Figure 4 shows a closeup of figure 3, we can see here that when there was slight drizzling people still used bikes but as the rainfall became heavier the amount of bikes rented decreased proportionately.



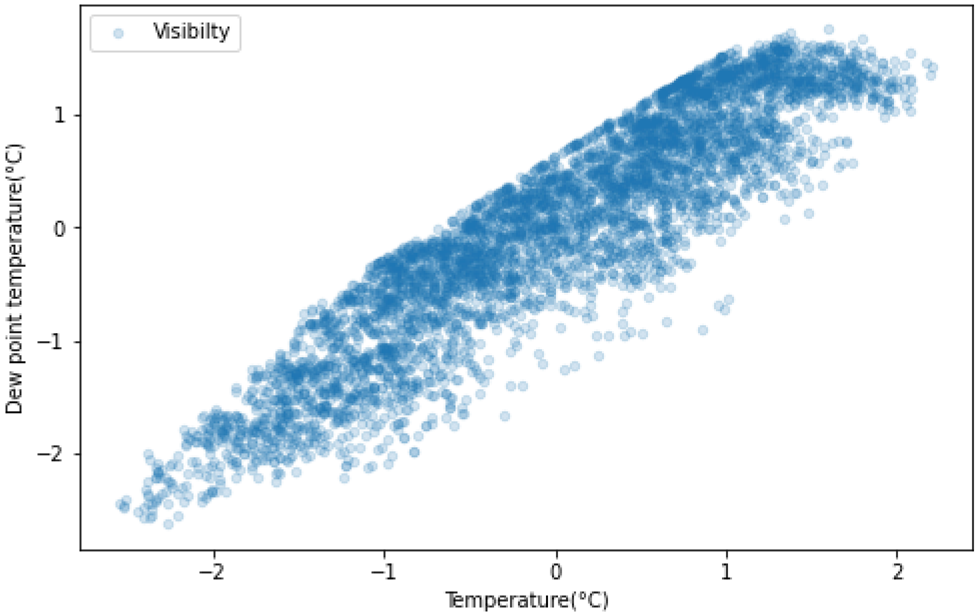
(Figure 5)

Figure 5 shows a 3D graph having three parameters namely temperature, humidity and rainfall. It can be seen that most bikes were rented when the temperature was warmer between 0℃ to 30℃, had no rainfall and had humidity between 40 to 80. While there were comparatively very few bike users when temperature was low with moderate humidity levels and no rainfall. There are a few outliers here as well which are mostly because of the bikes used in rainfall.



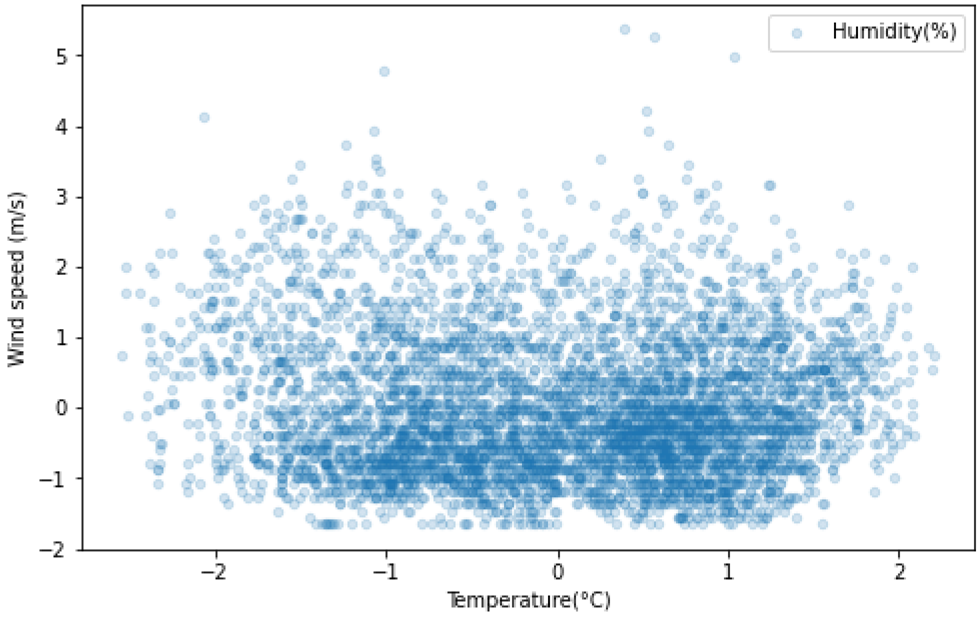
```
X_train.plot(kind="scatter", x="Temperature(°C)", y="Dew point
temperature(°C)",alpha=.2,label="Visibilty", figsize=(8,5))
plt.legend()
plt.show()
```

This code is used for making the graph. We have used scatter plot graphs because it enables us to understand the trends and patterns in our data.



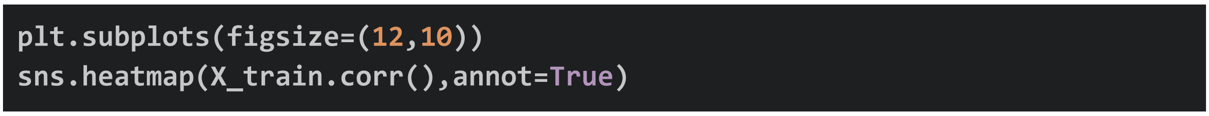
(Figure 6)

Figure 6 shows bike users data when comparing them against temperature and dew points. When there is less due point in the atmosphere, clear visibility and warmer weather that is when most people prefer to use bikes.The bikes rented decreases as temperature starts falling below zero degrees and due to poor visibility.



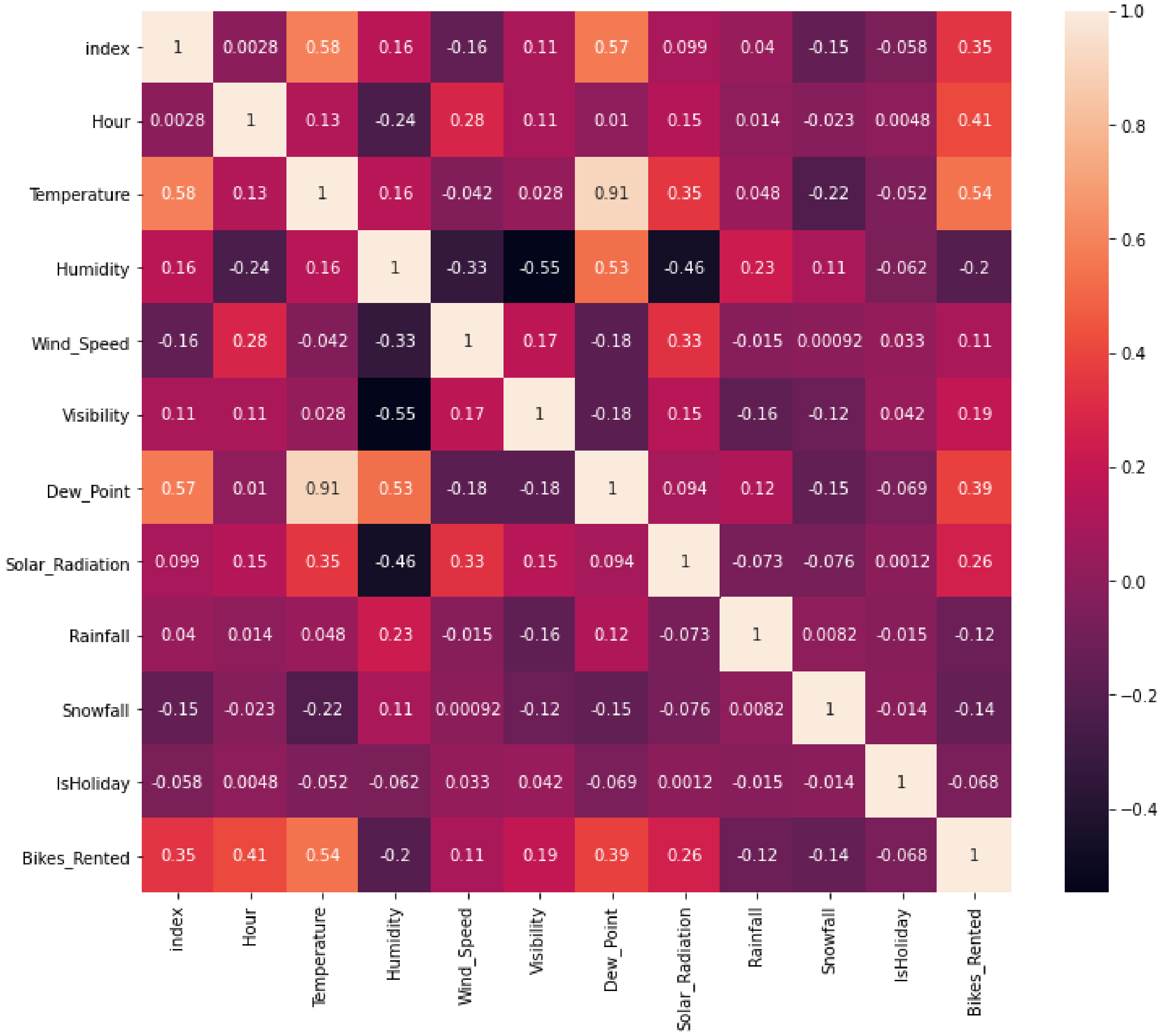
(Figure 7)

Figure 7 shows bike user data which clearly depicts that most people rent bikes at warmer temperatures and low wind speeds, that is when temperature is between -1 (℃) and 1 (℃) while the wind speed is zero.



```
plt.subplots(figsize=(12,10))
sns.heatmap(X_train.corr(),annot=True)
```

This code enables us to show a seaborn heatmap of correlations in X_train. It is an effective way to show the relationship between multiple variables in a dataset which is illustrated in **figure 8.**



(Figure 8)

Figure 8 shows the correlation heatmap between different variables in the dataset. We can see that bikes rented has a strong direct relation with temperature and hour while an inverse relation with humidity, rainfall, snowfall and is-holiday .Furthermore, temperature has a strong positive relationship with dew points and summer season(more bike users). On the other hand, temperature has a negative relation with rainfall and snowfall(hence low users of bikes).Finding the correlation between multiple variables in a data is widely used to identify patterns and the impact that each variable has on the whole project. Heatmaps are an effective way to project large amounts of data and drive relationships between variables. (Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D., 1998).

```
(X_train.isnull().sum() + X_train.isna().sum()).sum()
```

As we have already seen, this data is really nice with no null/NaN values.

5. Analysis and Interpretation of Evaluation Results:

Various machine learning models have been used in this research project to predict the demand of bikes according to various weather indicators. We have analyzed the data according to these six models:

a) Linear Regression
b) Support Vector Machine (SVM)
c) Decision Tree
d) Random Forest
e) Multiple Linear Regression
f) Extreme Gradient Boosting (XGBoost)

In order to increase the efficiency of these models we would be doing hyper parameter tuning as well. The criteria we have used to measure the suitability and accuracy of the models is by calculating the coefficient of determination(R^2), mean absolute error(MAE) and root mean squared error(RMSE).

a) Linear Regression:

The first modeling technique we have used is the simple linear regression predictive model. After running the model, we found that the coefficient of determination (R^2) is 0.485 ,mean absolute error is 339.773 and root mean squared error is 462.587

```
y_pred = lm.predict(X_test)
```

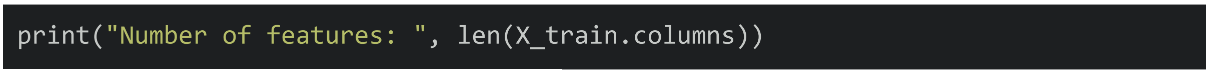
This makes prediction of counts of X_test.

```
r2 = sklearn.metrics.r2_score(y_test, y_pred)
mae = sklearn.metrics.mean_absolute_error(y_test,y_pred)
rmse = np.sqrt(sklearn.metrics.mean_squared_error(y_test,y_pred))
print("R2 Score:", r2)
print("Mean Absolute Error:", mae)
print("Root Mean Squared Error",rmse)
```

Evaluating the model on the test set.

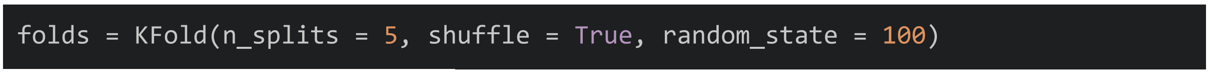
To further improve we have hyper tuned this model as well. After applying hyper parameter tuning and running the model we found the model to be much more improved and the values of coefficient of determination (R^2) to be 0.999 ,mean absolute error was 2.064 and root mean squared error was 2.549

Steps used in hyper parameter tuning:



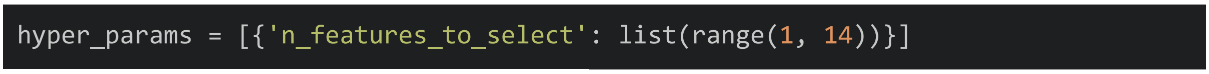
```
print("Number of features: ", len(X_train.columns))
```

This gives us the number of features in X_train.



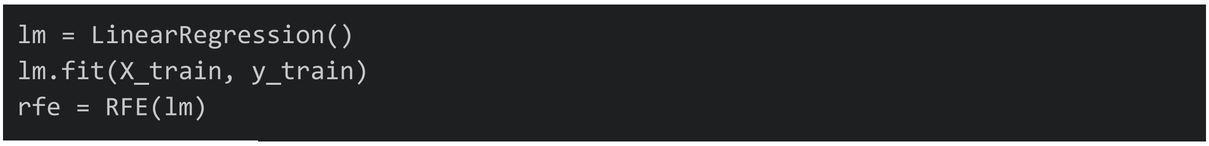
```
folds = KFold(n_splits = 5, shuffle = True, random_state = 100)
```

Creation of a cross validation scheme.



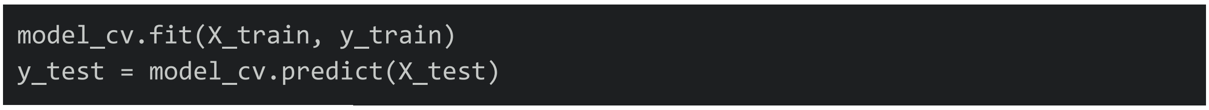
```
hyper_params = [{'n_features_to_select': list(range(1, 14))}]
```

Specifying the range of hyper parameters to tune.



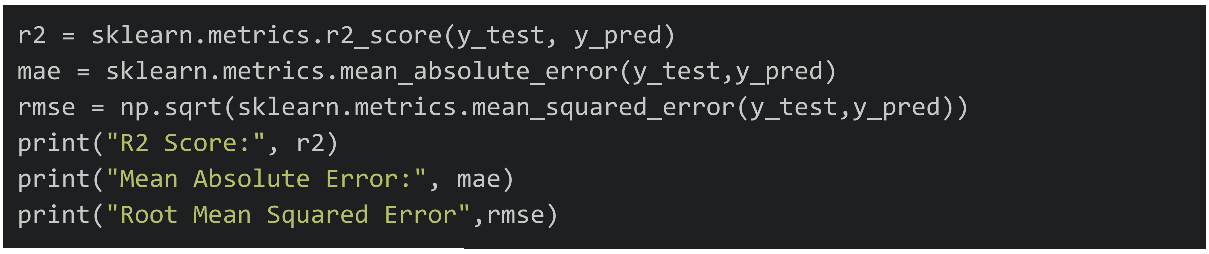
```
lm = LinearRegression()
lm.fit(X_train, y_train)
rfe = RFE(lm)
```

Specifying the model.



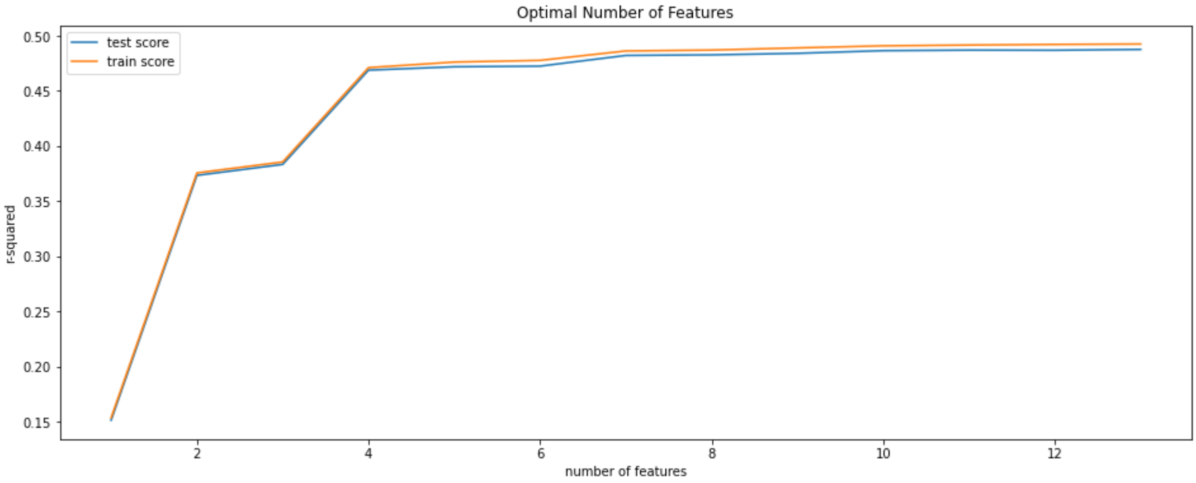
```
model_cv.fit(X_train, y_train)
y_test = model_cv.predict(X_test)
```

Fitting the Grid search model.



```
r2 = sklearn.metrics.r2_score(y_test, y_pred)
mae = sklearn.metrics.mean_absolute_error(y_test,y_pred)
rmse = np.sqrt(sklearn.metrics.mean_squared_error(y_test,y_pred))
print("R2 Score:", r2)
print("Mean Absolute Error:", mae)
print("Root Mean Squared Error",rmse)
```

Evaluation of the model on the test set.

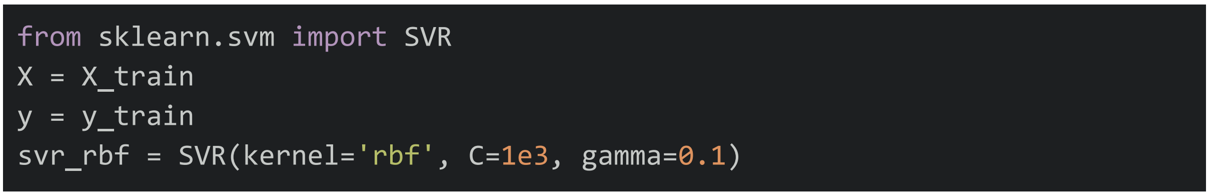


(Figure 9)

Figure 9 shows the optimal number of features needed for correct prediction of bikes in this project. It shows that 8 to 10 features are enough to predict properly in the model and any additional increase in features after that won't make any significant change. Here we can see that in figure 9 that from '1 to 2', '2 to 3' and '3 to 4' features there is a significant increase. After that the percentage increase has decreased significantly.

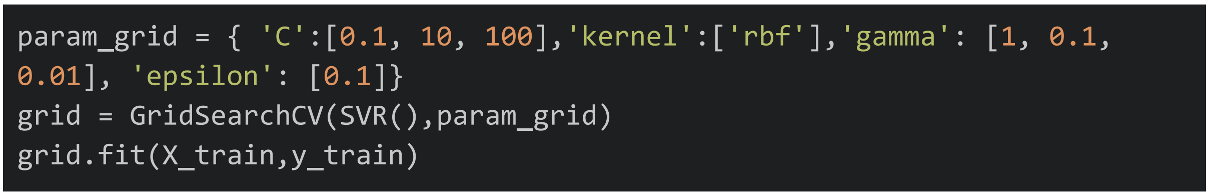
b) Support Vector Machine (SVM):

Next we have used the SVM model on our data set. Applying this model to our dataset, we found the values of coefficient of determination (R²) to be 0.714 ,mean absolute error was 196.390 and root mean squared error was 243.342 .For hyper tuning(optimization) we have used the 'Grid Search Cross Validation Technique'.

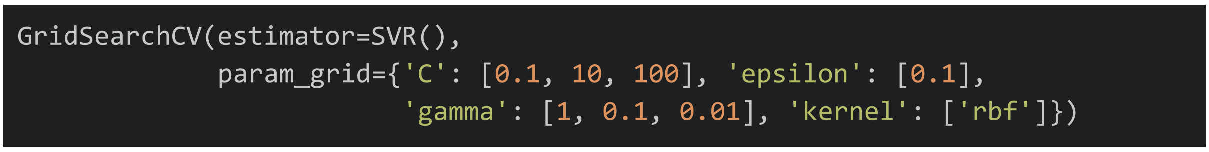


```
from sklearn.svm import SVR
X = X_train
y = y_train
svr_rbf = SVR(kernel='rbf', C=1e3, gamma=0.1)
```

Support Vector Machine Regression.



```
param_grid = { 'C':[0.1, 10, 100],'kernel':['rbf'],'gamma': [1, 0.1,
0.01], 'epsilon': [0.1]}
grid = GridSearchCV(SVR(),param_grid)
grid.fit(X_train,y_train)
```

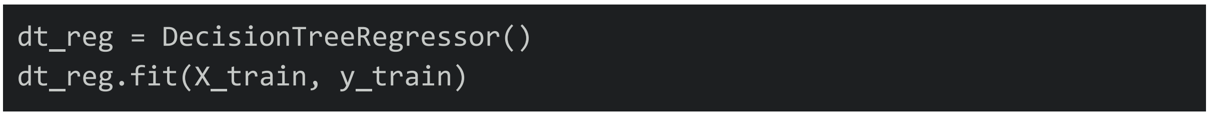


```
GridSearchCV(estimator=SVR(),
             param_grid={'C': [0.1, 10, 100], 'epsilon': [0.1],
                         'gamma': [1, 0.1, 0.01], 'kernel': ['rbf']})
```

Using 'Grid Search Cross Validation Technique' for hyper tuning.

c) Decision Tree:

We are using the decision tree regressor model for prediction here.
Applying this model to our dataset, we found the values of coefficient of determination (R²) to be -0.053 ,mean absolute error was 18.539 and root mean squared error was 467.584
Since this model did not fit our data set properly so there is no need to hyper tune it as it would still be inaccurate to make predictions based on this.



```
dt_reg = DecisionTreeRegressor()
dt_reg.fit(X_train, y_train)
```

```
dt_prediction = dt_reg.predict(X_test)
```

d) Random Forest:

We have used a random forest regressor model here. Applying this model to our dataset, we found the values of coefficient of determination (R²) to be 0.613 ,mean absolute error was 274.97 and root mean squared error was 367.16

We would be hyper tuning our parameters using grid search cross validation technique to optimize our model. After hyper tuning our model we found significant improvements in the values where the values of coefficient of determination (R²) to be 0.628 ,mean absolute error was 267.386 and root mean squared error was 352.988

```
from sklearn.ensemble import RandomForestRegressor
forest_reg = RandomForestRegressor(verbose=1,n_estimators=100,
random_state=42)
forest_reg.fit(X_train,y_train)
```

```
{'n_estimators': [3, 10, 30], 'max_features': [2, 4, 6, 8]},
```

Trying 12 (3×4) combinations of hyperparameters.

```
{'bootstrap': [False], 'n_estimators': [3, 10], 'max_features': [2, 3,
4]},]
```

Now trying 6 (2×3) combinations with bootstrap set as False.

```
grid_search = GridSearchCV(forest_reg, param_grid, cv=5,
                           scoring='neg_mean_squared_error',
                           return_train_score=True)
grid_search.fit(X_train, y_train)
y_test = grid_search.predict(X_test)
```

Training across 5 folds, that's a total of (12+6)*5=90 rounds of training.

```
r2 = sklearn.metrics.r2_score(y_test, y_pred)
mae = sklearn.metrics.mean_absolute_error(y_test,y_pred)
rmse = np.sqrt(sklearn.metrics.mean_squared_error(y_test,y_pred))
print("R2 Score:", r2)
print("Mean Absolute Error:", mae)
print("Root Mean Squared Error",rmse)
```

Evaluation of the model on the test set.

e) Multiple Linear Regression:

After implementing this model, we found the values of coefficient of determination (R²) to be 0.485 ,mean absolute error was 339.773 and root mean squared error was 462.587
To further improve the accuracy of our model, we hypertuned it using grid search cross validation technique. This improved the accuracy of our model and now the measurement for coefficient of determination (R²) came out to be 0.485, mean absolute error was 339.773 and root mean squared error was 462.587

```
n_features_optimal = 10
lm = LinearRegression()
lm.fit(X_train, y_train)
rfe = RFE(lm)
rfe = rfe.fit(X_train, y_train)
```

Model for linear regression.

```
y_pred = lm.predict(X_test)
r2 = sklearn.metrics.r2_score(y_test, y_pred)
mae = sklearn.metrics.mean_absolute_error(y_test,y_pred)
rmse = np.sqrt(sklearn.metrics.mean_squared_error(y_test,y_pred))
print("R2 Score:", r2)
print("Mean Absolute Error:", mae)
print("Root Mean Squared Error",rmse)
```

Predicting counts of X_test.

Steps in HyperParameter Tuning:

```
hyper_params = [{'n_features_to_select': list(range(2, 12))}]
```

Specifying the range of hyperparameters.

```
model_cv = GridSearchCV(estimator = rfe,
                        param_grid = hyper_params,
                        scoring= 'r2',
                        cv = folds,
                        verbose = 1,
                        return_train_score=True)
```

Setting up the GridSearchCV().

```
model_cv.fit(X_train, y_train)
y_pred=model_cv.predict(X_test)
acc = sklearn.metrics.r2_score(y_test,y_pred)
```

```
print(acc)
```

Fitting the model.

f) Extreme Gradient Boosting (XGBoost):

After implementing this model, we found the values of coefficient of determination (R²) to be 0.756 ,mean absolute error was 188.924 and root mean squared error was 317.831

```
algo = XGBRegressor()
params = {  "max_depth" : [1, 5, 10, 60, 70, 80, 90, 100],
            "gamma"     : [0.5, 1, 1.1, 1.2, 1.5]    }

grid   = GridSearchCV(algo, params, scoring='r2')
grid.fit(X_train, y_train)
```

```
y_pred=grid.predict(X_test)
r2 = sklearn.metrics.r2_score(y_test, y_pred)
mae = sklearn.metrics.mean_absolute_error(y_test,y_pred)
rmse = np.sqrt(sklearn.metrics.mean_squared_error(y_test,y_pred))
print("R2 Score:", r2)
print("Mean Absolute Error:", mae)
print("Root Mean Squared Error",rmse)
```

6. Comparison of all the models:

Before Hyper Parameter Tuning **After Hyper Parameter Tuning**

Models	**Coefficient of determination (R²)**	**Mean Absolute Error**	**Root Mean Squared Error**	**Coefficient of determination (R²)**	**Mean Absolute Error**	**Root Mean Squared Error**
Linear Regression	0.485	339.773	462.587	0.999	2.064	2.549
Support Vector Machine	0.714	196.390	243.342	0.714	196.390	243.342
Decision Tree	-0.053	18.539	467.584	-0.053	18.539	467.584
Random Forest	0.613	274.971	367.161	0.628	267.386	352.988
Multiple Linear Regression	0.485	339.773	462.587	0.485	339.773	462.587
XGBoost	0.756	188.924	317.831	0.756	188.924	317.831

We would be ranking the models on the basis of high coefficient of determination, secondly on the basis of low MAE(mean absolute error) and lastly low RMSE(root mean squared error).

From this table, we can see that the best model for making predictions about rental bikes is the 'XGBoost' model due to its low value of errors followed by the 'Support Vector Machine(SVM)' model. 'Decision Tree' is the worst model amongst all the models as it is under-fitting having a negative coefficient of determination value(R²). 'Linear Regression' model is overfitting, showing mean absolute error(MAE) of only 2.064 and root mean squared(RMSE) error of 2.549. 'Random Forest' has better results but it still overfits the training set. Lastly, 'Multiple Linear Regression' models can be used but it is showing high error numbers so it's not too accurate in making predictions.

ANOVA TESTING:

ANOVA or analysis for variance is a statistical analysis technique that is used to check the significance of variables and check the relationships between dependent and independent variables. **9(St, L. and Wold, S., 1989).** We would be using SPSS software for doing this ANOVA tests. We can do one, two or three way ANOVA tests .The interpretations from the table provide information about relationship existences between variables, identifying significant factors and interactions.

Tests of Between-Subjects Effects

- **Dependent Variable: Bikes Rented**

Source	Type III Sum of Squares	Degrees of Freedom	Mean Square	F - calculated	Sig.
				Low number is bad	**Less than 0.05**
Season	155379799.299	3	51793266.433	170.520	.00004
Is_Holiday	180250.002	1	180250.002	.593	.441
Is_Functional_Day	54832428.678	1	54832428.678	180.526	.000003
Season * Is_Holiday	1086445.677	3	362148.559	1.192	.311
Season * Is_Functional_Day	849925.285	1	849925.285	2.798	.094
Is_Holiday * Is_Functional_Day	59911.153	1	59911.153	.197	.657
Error	2201788939.984				
Total	6615315306.000	7260			

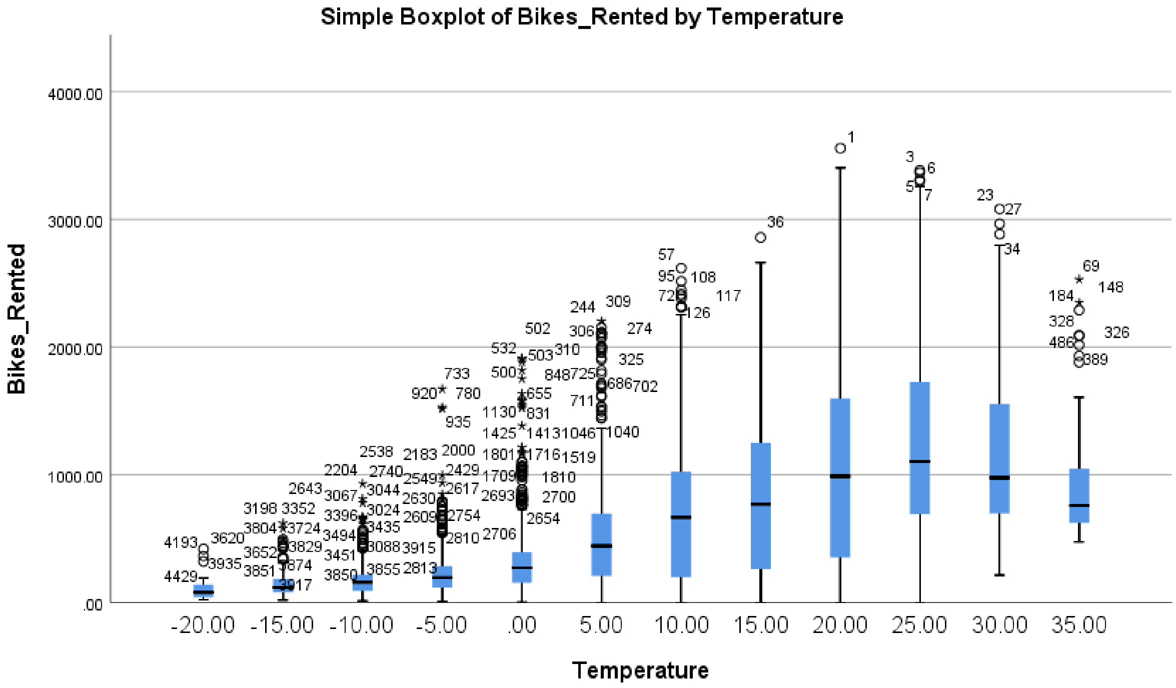
Corrected Total	3022992414.890	7259			

The lowest significance out of all three factors was the ‘is holiday’. Whereas all other variables lied within a 95% confidence level.

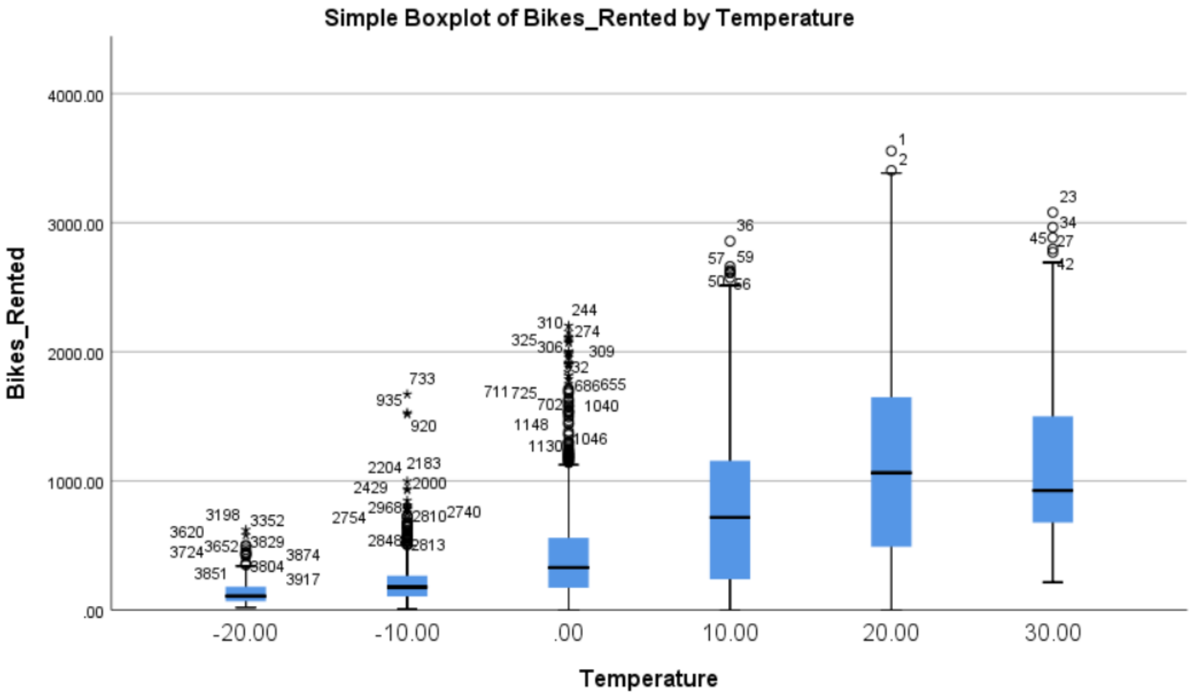
The highest contribution was led if the day was functional or not (180.53), showing the highest significance (0.00003). Intersections between all independent parameters did not verify any significance by other means they don’t have an effect on each other which is true that’s why all intersections F values were very low.

- **Testing bikes rented against the climate conditions:**

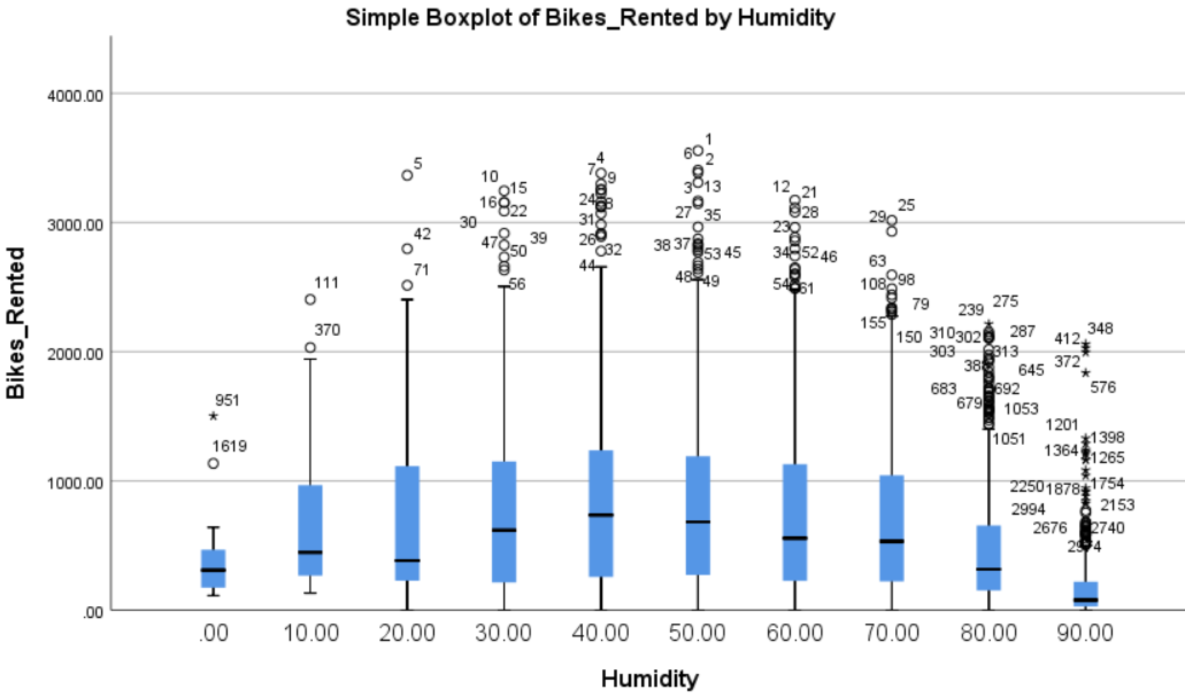
1. **Temperature.**
2. **Humidity.**
3. **Wind Speed.**

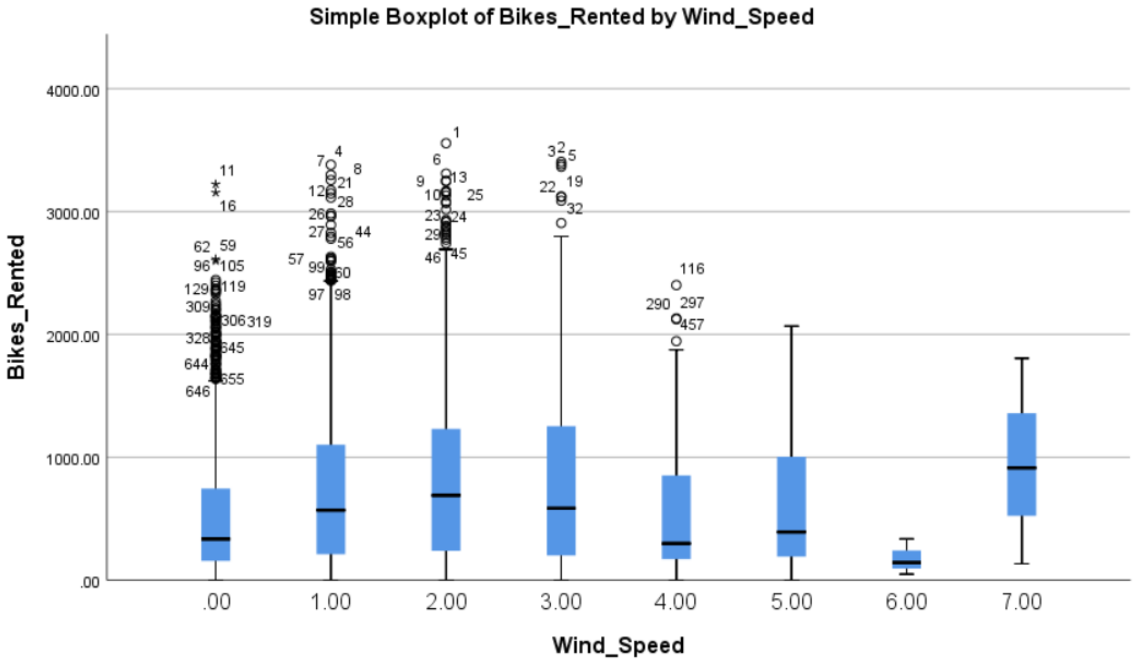


Rounding to the upper box plot to 10s as then it can be understood more clearly.



A lot of outliers were present when the temperature lied below zero, because I am assuming that the measuring device was sending inaccurate data. Some outlier days were present depending on the people’s emergency needs to leave the house. The maximum median of bikes rented lied when the temperature was 20 degrees.





Tests of Between-Subjects Effects

- **Dependent Variable: Bikes Rented**

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Rainfall	1770045.403	45	39334.342	.102	1.000
Snowfall	3991175.884	50	79823.518	.206	1.000
Rainfall * Snowfall	2330197.415	15	155346.494	.401	.979
Error	2767497149.081				

Total	6615315306.000				

a. R Squared = .085 (Adjusted R Squared = .070)

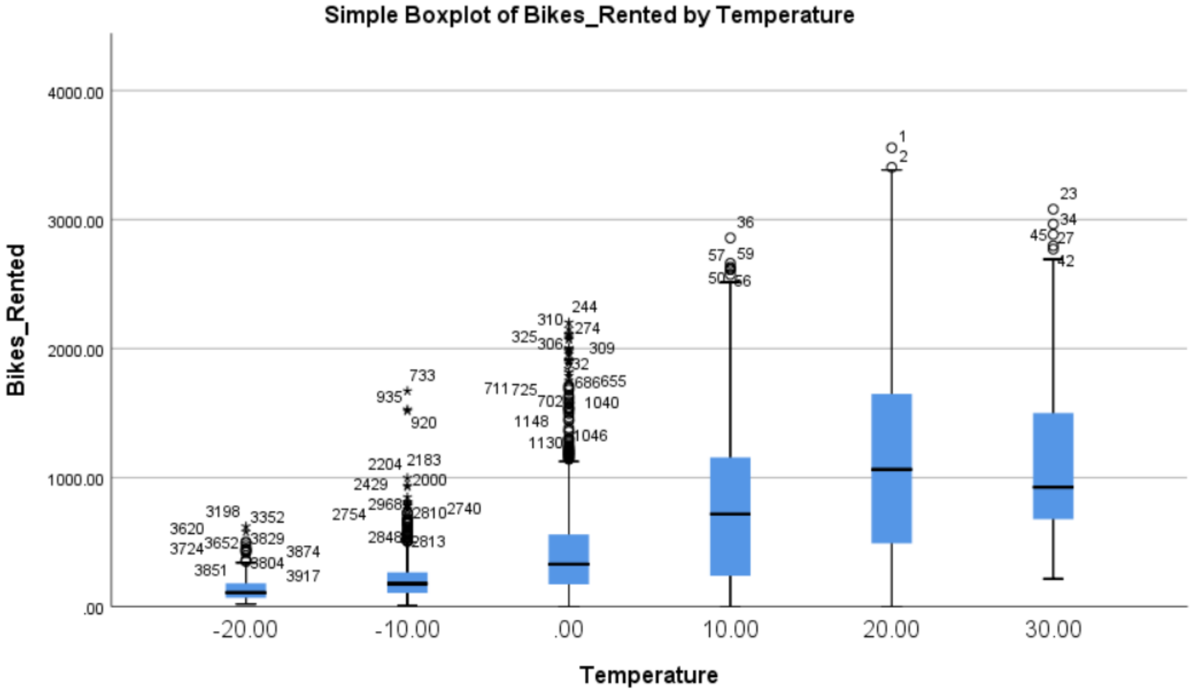
ANOVA test verified the data, that means the values are random, with no correlation and no significance on results of bikes rented as it needs to be below 0.05

Correlations

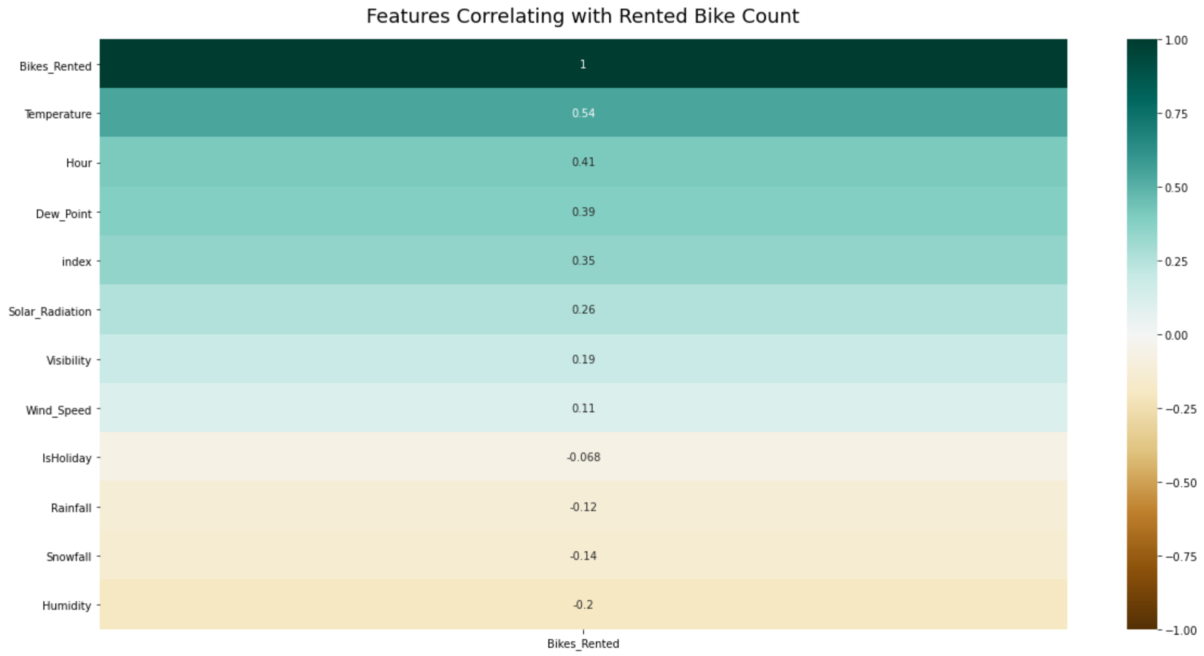
			Bikes Rented	Temperature	Humidity	Windspeed
Spearman's Correlation	Bikes Rented	Correlation Coefficient	1.000	.549**	-.215**	.128**
	Temperature	Correlation Coefficient	.549**	1.000	.156**	-.009
	Humidity	Correlation Coefficient	-.215**	.156**	1.000	-.337**
	Windspeed	Correlation Coefficient	.128**	-.009	-.337**	1.000
		N	7260	7260	7260	7260

**. Correlation is significant at the 0.01 level (2-tailed).

Here, we have used Spearman’s correlation to show the relationships between variables. Bikes on bikes rented would obviously have a correlation of 1 as they perfectly correlate, the highest correlation with the bikes rented is with temperature(0.549) followed by wind speed with a very low correlation of 0.128 and then with humidity bikes has a negative correlation. Similarly the table is also showing temperature, humidity and wind speed correlation with other variables where temperature has the least correlation with wind speed (-0.009), humidity has least with wind speed(-0.337) and wind speed has least with humidity(-0.337).



The highest variance was found when the temperature was 20, and it has been shown that the situation optimises when the temperature is 20 degrees, the following Interquartile Range plot (IQR) could be further expanded into details from 18 degrees to 25 in order to specify a more accurate result.



From the above figure, it is evident that bikes rented has the highest correlation with the temperature which is 0.54. Thus a change in temperature has a huge impact on the demand of the bike. This is followed by hour(0.41), dew point(0.39), solar radiations(0.26), visibility(0.19), wind speed(0.11), is-holiday(-0.068), rainfall(-0.12), snowfall(-0.14) and humidity. The least correlation which bike has is with humidity which is -0.2

7. Critical review and analysis of techniques used:

a) Linear Regression:

It is a statistical regression model which is used to find linear relationships between dependent and independent variables through the use of a linear line(Neter, J., Wasserman, W. and Kutner, M.H., 1989). Linear regression model is shown by the following equation:

$$Y = a + bX$$

Here X is the independent variable whereas Y is the dependent variable. ‘b’ is the slope of the line and ‘a’ is the intercept. Moreover, linear regression models are easy to understand, interpret and can be trained easily on every machine as compared to other models. Regression line is used for predicting patterns and the nearer the value is to the regression line the better.
Furthermore, root mean squared error(RMSE) and mean absolute error(MAE) can be used for testing correlations but it is preferable to use MSE as it gives weightage to outliers as well.
This model is easy to understand on a modular level due to its linearity. It is also easy to train the data set on the basis of which we would be making predictions. But this model has an inherent fault of overfitting the data which has happened here and thus it is not a suitable model to use.

HYPER PARAMETER TUNING (Linear Regression):

Hyper parameters tuning means adjusting the variables which are used to design a machine learning model in order to make it more efficient (Yang, L. and Shami, A., 2020). Hyper parameter Tuning is done to increase the efficiency of the model and eventually gives us better results. We are using the ‘Grid Search Cross Validation’ technique for optimization of results.

Grid search uses a technique where it searches the best possible combination of all the parameters available to increase the accuracy and prediction score of the model. Cross validation is a technique used to effectively split the data into training and testing datasets. (Abas, M.A.H., Ismail, N., Ali, N.A., Tajuddin, S. and Tahir, N.M., 2020). We would be combining both these two techniques to find the best available parameters for optimization of our model.

b) Support Vector Machine (SVM):

Support Vector Machines are algorithms that learn from its experience of analyzing the data by labeling it and eventually predicting future values based on this analysis (Noble, W.S., 2006).
We have used Grid Search Cross Validation technique to hyper-tune our model further to improve our model results.
This model is best suited for high dimensional spaces and where classes are separate. It has a useful feature inside it called kernel which enables us to pick a function that is not necessarily linear in nature which has allowed it to give relatively good predictions in our data set.

c) Decision Tree:

Decision tree is a machine learning algorithm technique that is used to separate dataset into subsets. It's a predictive modeling statistical technique that learns from observing the data and then coming up with its own conclusions regarding that data about its future .It is very useful in extracting patterns, trends and making efficient predictions from a database. (Myles, A.J., Feudale, R.N., Liu, Y., Woody, N.A. and Brown, S.D., 2004).
This model is simple, automatic, and does not require too much pre-processing of data. One of the biggest problems with this model is that it overfits the data easily, has issues with accurately predicting from out of sample data and is very sensitive to change in data. This is what has happened here as well hence the poor performance of the decision tree model over here.

d) Random Forest:

Random forest is an ensemble learning method in which we make multiple decision trees and combine these predictions to come up with an average prediction which would represent an accurate model (Biau, G. and Scornet, E., 2016).
Random forest is very good with data containing missing fields. It is very versatile as it can be used for classification and regression tasks. The reason for this model being not too accurate is that a small change in data like the outliers had a large change on the random forest optimization. Due to outliers in our data this model has not performed as good as we thought but still it is considered relatively better then other models.

e) Multiple Linear Regression

Multiple linear regression model is used to predict the relationship between two or more independent variables and a single dependent variable. This model is an extension of a

simple linear regression model but has more than one explanatory variable (Tranmer, M. and Elliot, M., 2008).
This model allows for multiple variables to be checked in one model. This model has a huge issue of overfitting the data and the linear nature of relationship between independent and dependent variables which causes it to predict in-accurately which is exactly what has happened here.

e) Extreme Gradient Boosting (XGBoost):

It is a machine learning algorithm based on decision trees which makes predictions about a variable by merging estimates of other models. We have used the XGBoost regressor model here (Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y. and Cho, H., 2015).
This model is highly flexible, is faster, uses parallel processing, tackles the missing items well and after each iteration we can run cross validation which makes this model one of the best and most reliable models. That is what happened here and hence its giving the highest coefficient of determination values and the lowest error as compared to other models.

8. Reflection upon the project experience:

The whole project was a great learning experience for me. It enabled me to further my python skill and learn some new libraries in the process. Usage of six models to make predictions also helped me alot in learning new things as I encountered many issues while running the model so with every error I learnt a new thing. While processing the data I faced many challenges as well which taught me many new skills on its way. Moreover I also learnt to hyper tune the model which was a completely new thing for me. I also learnt how to make different types of graphs and how to effectively utilize visualizations to communicate the message across the user. Lastly, I also utilized my statistics skills and further polished them in the usage of SPSS software to do ANOVA testing and finding correlation between variables which was helpful in coming to a final conclusion towards the dataset.

9. Conclusion:

The whole data could be mathematically and statistically interpreted which is what we did here using a machine learning algorithm. We made predictions on bike usage in Seoul by users considering other variable factors and came up with the results as to which model is best for making future predictions .The best model to use here is Extreme Gradient Boosting model(XGBoost), followed by Support Vector Machine(SVM) model, then Random forest, then Multiple Linear Regression, after that Linear Regression model and lastly Decision Tree which was the worst model to use. The basis of our decision was based on a higher value of coefficient of determination(R^2), lower value of mean absolute error(MAE) and low root mean squared error(RMSE).

The variable which had the highest impact and correlation on bikes rented was temperature while the variable which had the least impact was humidity.

REFERENCES:

1. Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D., 1998. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, *95*(25), pp.14863-14868.

2. Neter, John, William Wasserman, and Michael H. Kutner. "Applied linear regression models." (1989).

3. Yang, L. and Shami, A., 2020. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, pp.295-316

4. Abas, M.A.H., Ismail, N., Ali, N.A., Tajuddin, S. and Tahir, N.M., 2020. Agarwood oil quality classification using support vector classifier and grid search cross validation hyperparameter tuning. *Int. J*, *8*.

5. Noble, W.S., 2006. What is a support vector machine?. *Nature biotechnology*, *24*(12), pp.1565-1567.

6. Myles, A.J., Feudale, R.N., Liu, Y., Woody, N.A. and Brown, S.D., 2004. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *18*(6), pp.275-285.

7. Biau, G. and Scornet, E., 2016. A random forest guided tour. *Test*, *25*(2), pp.197-227.

8. Tranmer, M. and Elliot, M., 2008. Multiple linear regression. *The Cathie Marsh Centre for Census and Survey Research (CCSR)*, *5*(5), pp.1-5.

9. St, L. and Wold, S., 1989. Analysis of variance (ANOVA). *Chemometrics and intelligent laboratory systems*, *6*(4), pp.259-272.

10. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y. and Cho, H., 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2*, *1*(4).