PROJECT GROUP 2

SEOUL BIKE SHARING DEMAND PREDICTION

Group Members: Nikita Pai, Jay Soman, Parikshit Dumbhare

Link to dataset: https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand

ABSTRACT

Seoul city in South Korea has a rental bike sharing program. The common public can pick up and drop the rental bike in many different bike stands. The dataset contains the rental bike count and many other variables that could affect the demand such as temperature, humidity, visibility, etc. In this paper, our group has attempted to predict the demand for rental bikes reliably using the features given in the dataset. We have used two models to do regression, one being Linear Regression and the other being regression using Neural Networks. Below we present our findings as well as exploratory data analysis we used to get deeper insight into the data.

INTRODUCTION

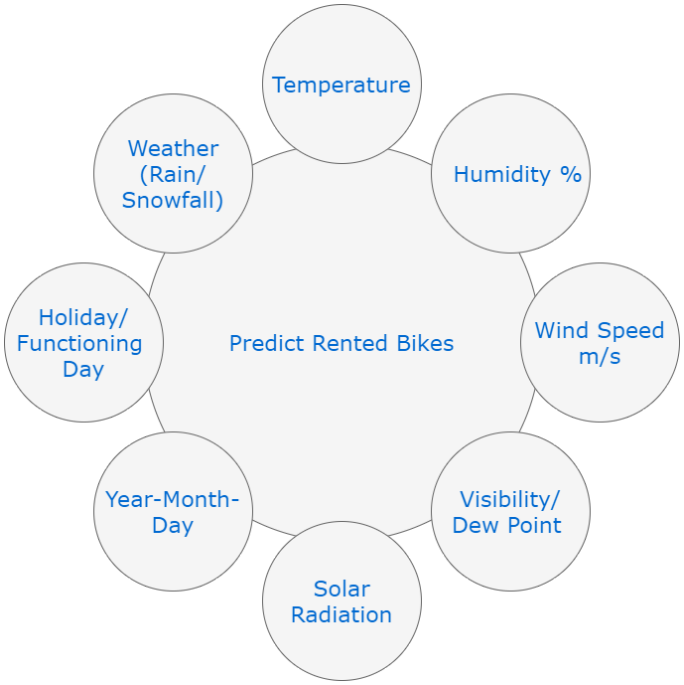
Bike sharing is a method of renting bicycles that is automated using a network of kiosks located across a city. People can hire a bike from a single pick-up site and combine it with their needs. The customer returns the bike to a kiosk of their choice after using the bike. Such rental programs are becoming increasingly popular in highly dense cities thanks to automated systems and the smartphone age we live in. Individuals purchase a membership and then use the system to hire a bike from one location and return it to another within a certain amount of time. Seoul's bike-sharing demand comprises one of the world's 500 bike-sharing schemes. The system administrator for such a program would ideally like to predict the demand to make sure the city doesn't rent less bikes than it requires. As demand for bike rentals is a complicated metric to predict, it is dependent on a number of factors, including weather, humidity, temperature, and age group.

The aim of this project was to predict the number of rented bikes using different techniques learnt in the course. As a baseline model, we have chosen Linear Regression and we have chosen Non-Linear Regression using Neural Networks as the more favourable "advanced" model. To start with, we did exploratory data analysis on the dataset to gain an insight into the dataset. Then the data was cleaned and moulded to make it favorable for feeding into the models. In the end of the report we discuss our findings and conclude as to which model performed better and what is the future work possible.

DATASET DESCRIPTION

Originally our dataset consisted of 14 Features (columns) and 8760 instances (rows). There are 11 real valued features and 3 categorical variables. We preprocessed the data by dropping certain columns and shaped the same according to our needs. We made the date vector into 3 feature vectors using one hot encoding into month, year and day. We have assigned data type to the variables which are given as follows:

Rented Bike Count	int64
Hour	category
Temperature(°C)	float64
Humidity(%)	int64
Wind speed (m/s)	float64
Visibility (10m)	int64
Dew point temperature(°C)	float64
Solar Radiation (MJ/m2)	float64
Rainfall(mm)	float64
Snowfall (cm)	float64
Seasons	category
Holiday	category
Functioning Day	category
Year	category
Month	category
Day	category
DayofWeek	category



The data consists of 365 days and an instance for each hour of the day, hence 24 instances for a day. The dataset does not contain any missing values or NaN values.

EXPLORATORY DATA ANALYSIS

Exploratory data analysis is required to gain insight into the data. Preliminary studies on data were conducted using summary statistics and graphical representations such as heat maps, bar plots, and other similar tools to find patterns, anomalies, test hypotheses, and see the distributions of the data and features.

The following are the results of the EDA performed:

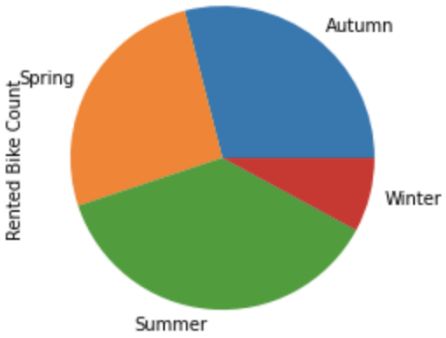


Figure 1

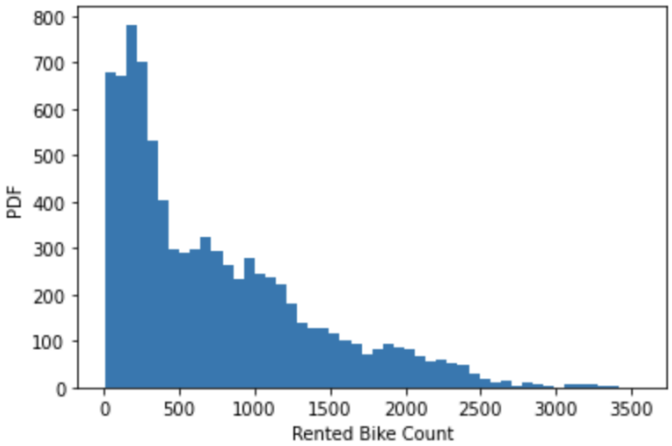


Figure 2

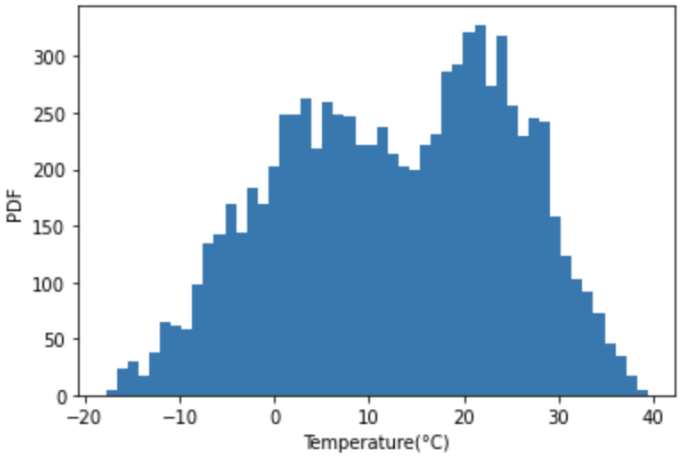


Figure 3

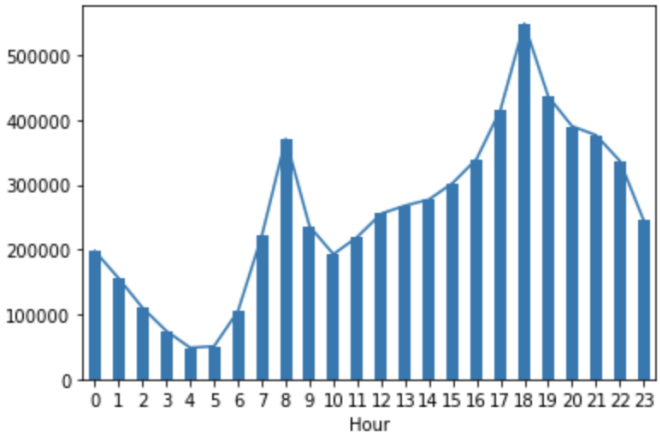
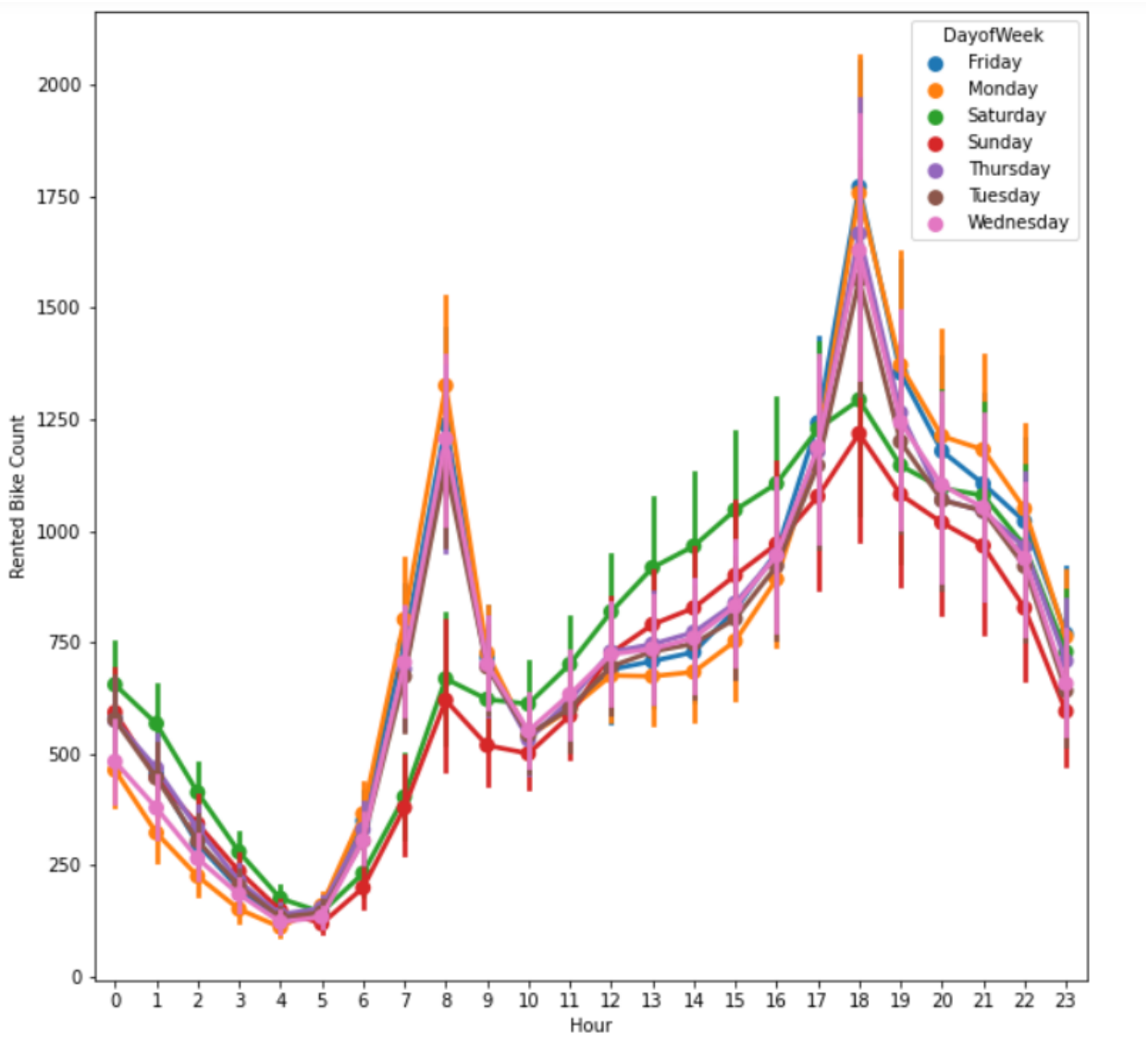
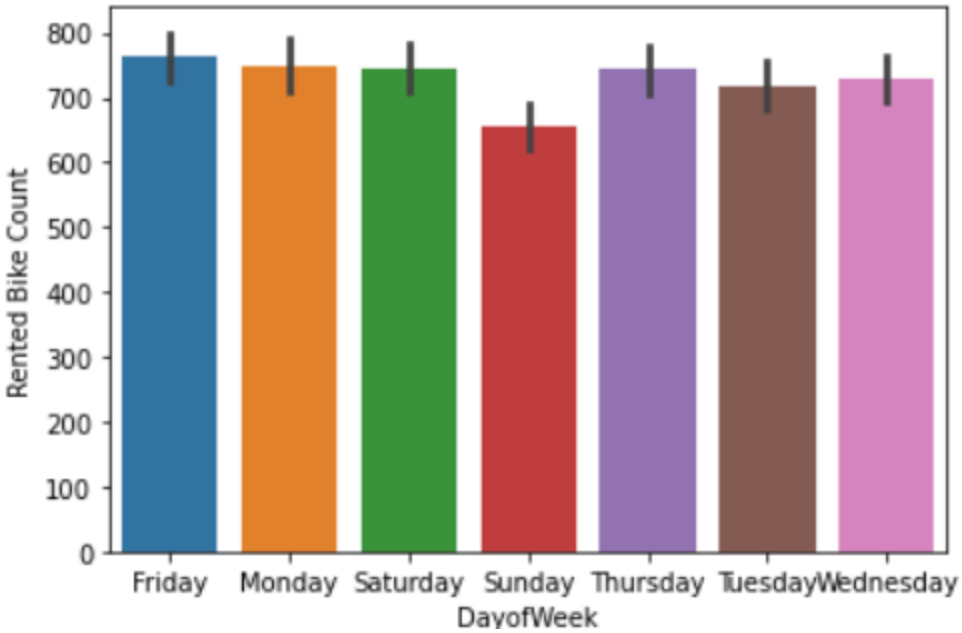


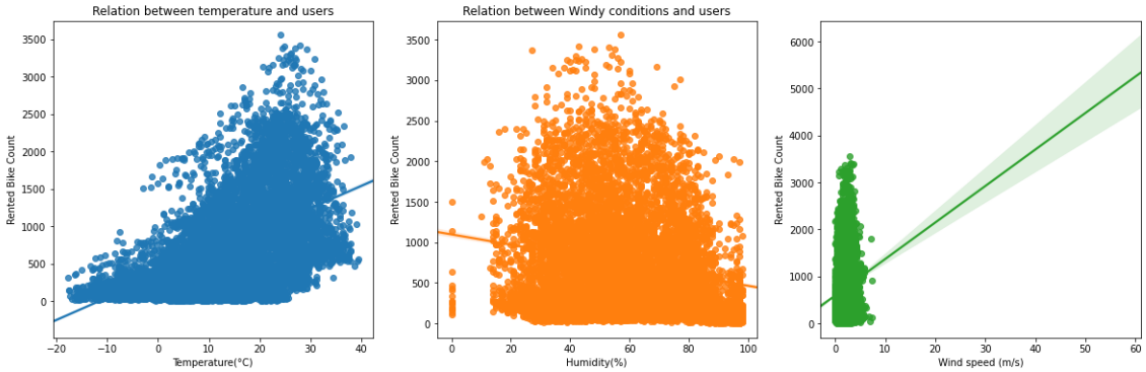
Figure 4

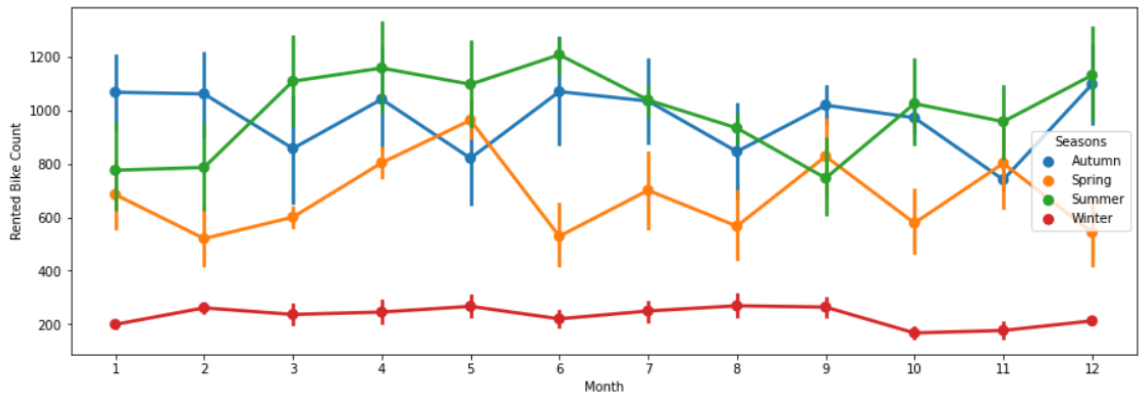
- FIgure 1 : Season wise distribution of the rented bike count
- Figure 2: Probability Distribution Function(PDF) of the Rented bike count
- Figure 3: PDF of Temperature(C)
- Figure 4: Hour wise distribution of rented bike count



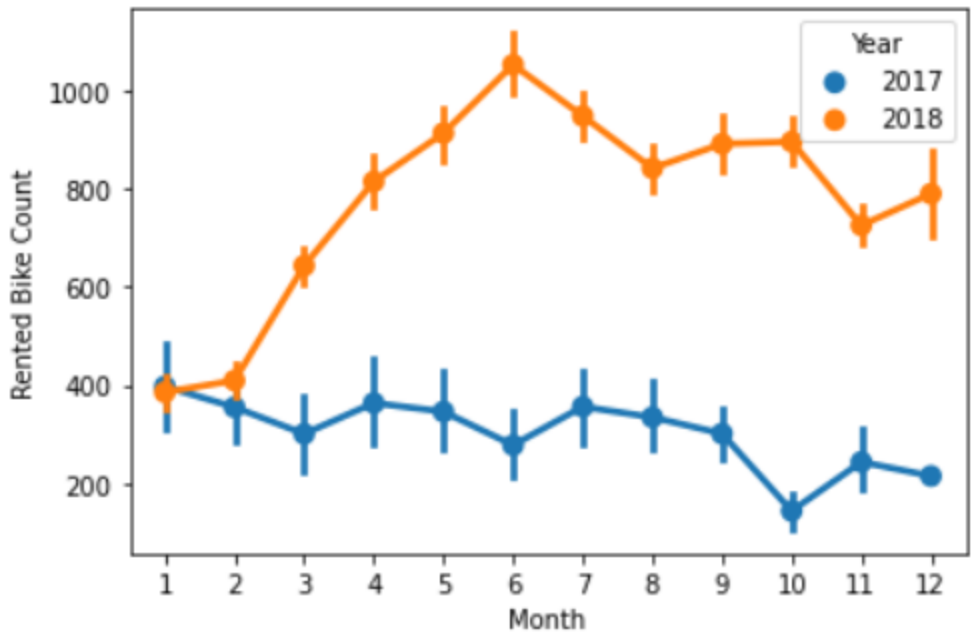
Plot above shows that the users are highest on weekdays during working hours. It might show that most bike users are students/working professionals who frequently use the bike during daytime to reach/leave a destination. Meanwhile on weekend, the upsurge between 10 am to 5pm indicate that people might like travelling for outings on weekend







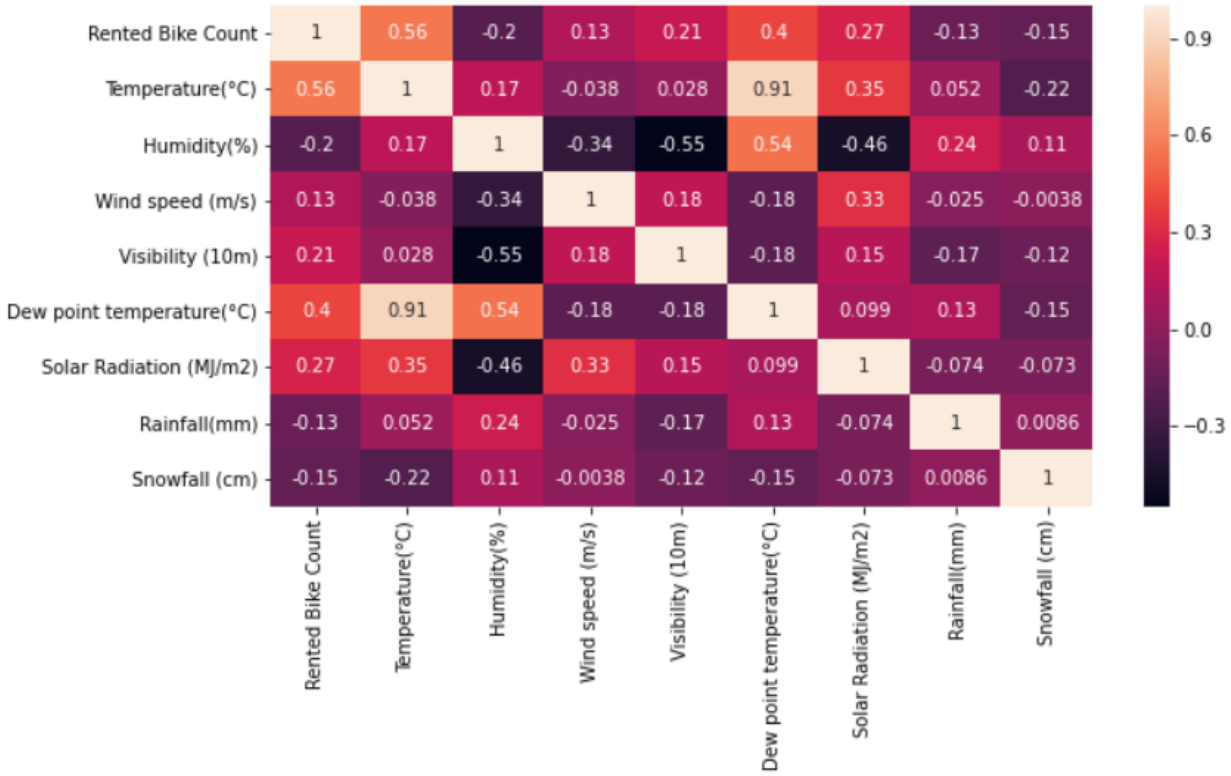
Summer, Autumn followed by Spring is the most active season for biking. People don't prefer biking during the Winter season, which can significantly affect the number of bikes that need to be present.



Year 2018 has been a much busier year than 2017 for reasons we do not know.

CORRELATION

Temperature is shown as the most correlated feature amongst all the features. Other factors don't show that much correlation towards outcome. We don’t see any factor that highly resembles one another except the Dew point temperature with temperature which is expected. Rainfall and Snowfall show the least contribution in this dataset. Due point, Rainfall, Snowfall can be eliminated when the dataset is of high volume. This finding is surprising as one would expect there to be a significant correlation between bike demand and weather conditions such as rainfall. Hence we have set the threshold rate between +-(0.9 and 0.2) to the target variable. Anything beyond and/or below that can be considered insignificant.



MISSING VALUES/ELIMINATION

There are no missing values in the dataset. However, if the models have to run with the updated dataset in the near future, the following two methods can be implemented.

1. Imputing - Filling out the numerical value with mean value for factors that contribute significantly(Temperature)
2. Elimination - Elimination of rows that doesn't hold much information

ONE HOT ENCODING

Categorical variables cannot be passed into regression models directly, hence we've passed all the categorical variables in binary form through one_hot_encoding function.

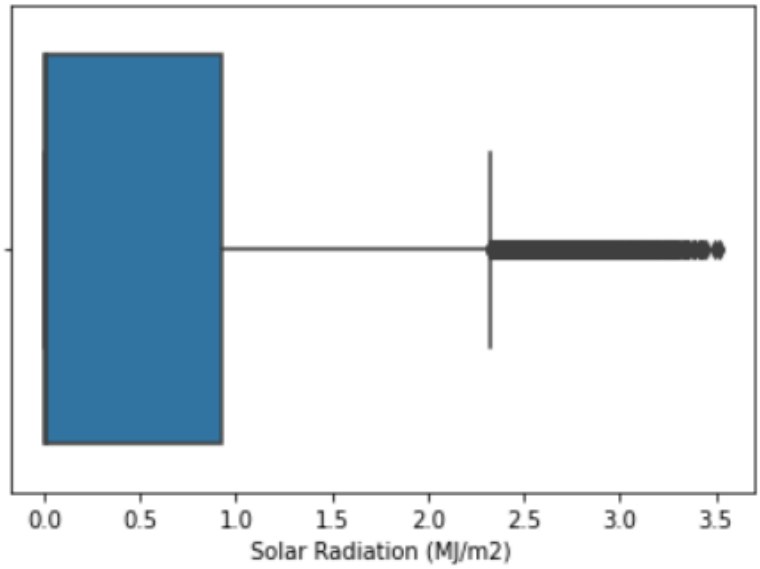
SCALING/NORMALIZATION

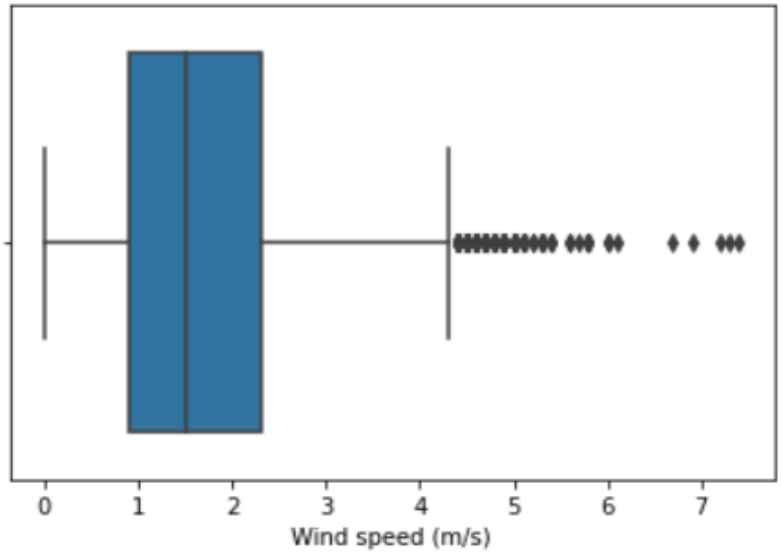
Shifting all the values and rescaling so that the variables range between 0 and 1

OUTLIERS ELIMINATION

Solar Radiation and Wind Speed showed outliers that could have affected the accuracy of the model.

The shape of our dataset after removal of outliers and added one_hot notation is (7638, 91)





The above are box plots of wind speed and solar radiation variables.

IMPLEMENTATION OF MODELS

Linear Regression:

As a baseline, we used linear regression to predict the rented bike count. Linear regression is a method to fit a straight line onto the variable we are trying to predict. The model takes in two matrices X and Y, where X is a matrix of features and Y is the feature we are trying to predict. The output is predicted values as well as linear coefficients that the model learns. It is a batch based learning algorithm. The metric used to measure the accuracy of the model is Root mean squared error(RMSE). The error used here is the vertical distance between the predicted value

and the actual value. Lower the RMSE, better the fit of the model. Since the target variable is not a linear curve, a linear regression model is expected to perform poorly on the dataset.
The steps taken to perform linear regression:

1. Load dataset
2. Split date feature into 3 features - day, month and year
3. Remove the feature “functioning day” from the dataset as all instances have 0 values
4. Use one hot encoding to make seasons, holiday, hour, day, month, year into binary features
5. Create X as the feature matrix and Y as the target variable
6. Feed into the model
7. Observe output RMSE

If the closed form solution does not exist, the model would automatically run SGD as an alternative. The model also has optional inputs to use regularisation (Lasso). Regularisation is a technique to make sure that the model does not overfit by introducing an artificial error term using a lambda value. The linear regression model used 25% of the data as a test set and 75% as a training set.

Neural Network:

As our target variable follows a non-linear curve, it makes sense for neural networks to implement a non-linear curve to fit. There are a number of methods to deal with non-linear data. Our approach here is to use Neural Network in the form of regression to implement on the dataset.

We implement Neural networks using Keras and Tensorflow. The implementation of model has been done adjusting the following parameter:

1. Sequential API has been chosen over Functional API in Keras to create model layer by layer
2. Adam has been chosen as the optimization function over tanh and sigmoid because:
    - It computes Adaptive learning rate instead of constant learning rate
    - Computationally efficient
    - Little memory requirement leading to better performance
    - Well suited for large dataset affected by a number of factor
3. The final layer will be “linear” instead of “softmax” since it’s a regression model

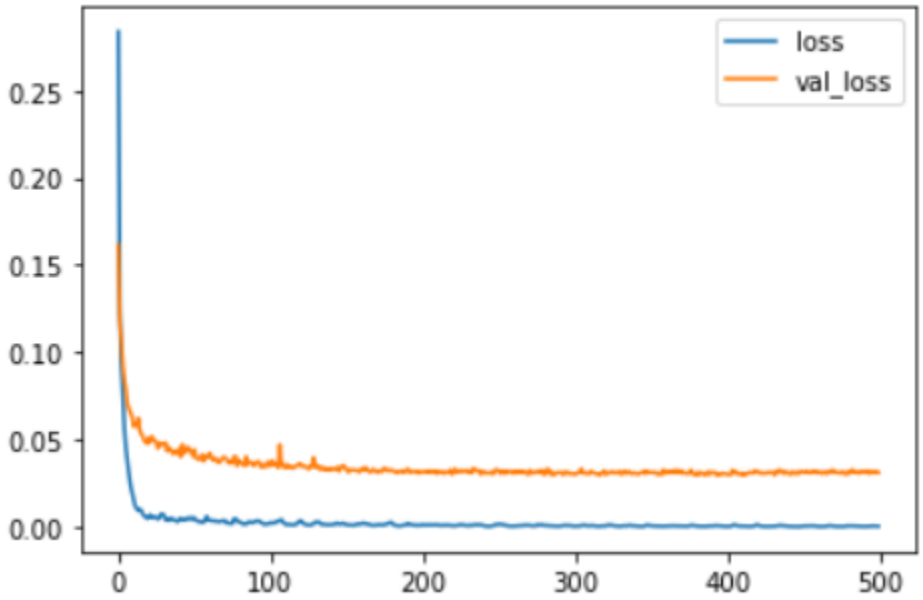
As it can be seen from the image below, the variance captures more than 96% which implies a pretty accurate model. Along with the the mean_squared error is pretty low as well that justifies the model's performance

```
In [384]: pred = model.predict(X_test)

In [385]: mean_squared_error(y_test, pred)
Out[385]: 0.03625542197235573

In [386]: mean_absolute_error(y_test, pred)
Out[386]: 0.11306966033599986

In [387]: explained_variance_score(y_test, pred)
Out[387]: 0.963178495423104
```

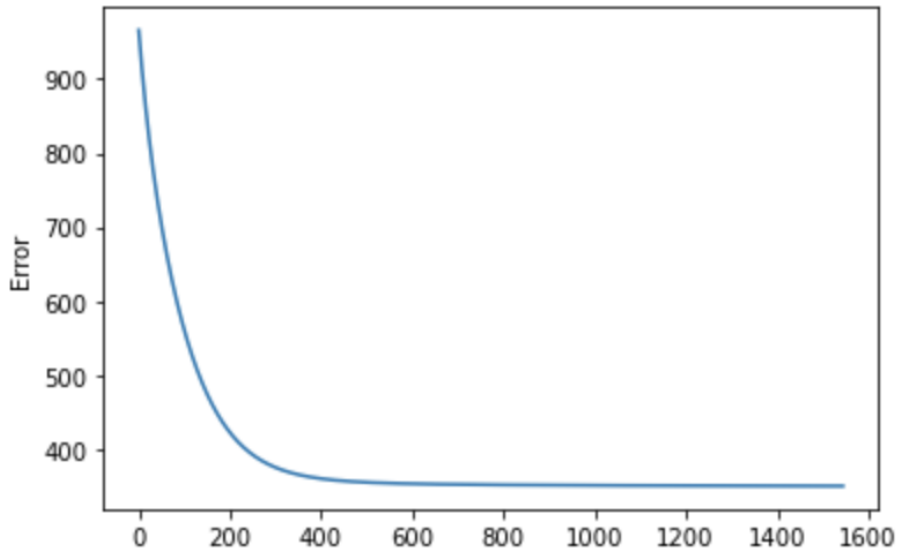


As the validation data loss is not curving up and is constant throughout the line, it means that there is no case of overfitting that would affect the performance

And finally, the test data vs. predicted shows the following line indication the line has been fitted perfectly to inculcate all the data.

RESULTS

We found that the baseline linear regression performed poorly as expected. Since our target curve is highly non-linear, a straight line fit does not make sense. We ran Stochastic Gradient Descent as well as Gradient Descent. A closed form solution for this dataset does nor exist as it is not a full rank matrix, which implies that X^t*X does not have an inverted solution. RMSE observed using SGD was 366 whereas RMSE using GD was 364 which is expected as both would converge to the same theta eventually. What was surprising to us was that SGD took a lot longer to compute, compared to GD. The reason we have concluded is that our SGD function is not well optimised and needs tuning. Hence GD is preferred here for testing purposes.




```
error rmse :  364.24564360118325
```

Our results using the neural networks model were much more encouraging as seen below. And we conclude that it is a better model to use for this dataset than linear regression. This is a figure of predicted points vs actual points and we were able to get a good prediction of the target variable. We were able to achieve a RMSE of 0.36 and an explained variance 96%.