**Predicting Bike Sharing Rental Demands**

***DATA 1030: Midterm Project Report***

Radhika Mehrotra

Brown University

https://github.com/rads1306/DATA1030-Radhika-Mehrotra

Supervised by Prof. Andras Zsom

## I. Introduction

Stock procurement is often a challenge for businesses since it requires predicting future consumer demand. Demand patterns that are a result of interplay between a multitude of factors make the business problem more complex. Businesses seek to find patterns in consumption as they serve as useful indicators to solve such business problems.

I use Seoul's bike sharing demand dataset to understand how weather conditions, and holidays affect bike rental demand.

**Target variable: Rented Bike Count**

"Rented Bike count", in other words the number of bikes rented in a given hour on a particular date is the target variable. It follows a continuous distribution. To predict the number of bikes rented on a day in the future, we want to understand the quantitative influence that the other features in in the dataset have on the number of bikes rented.

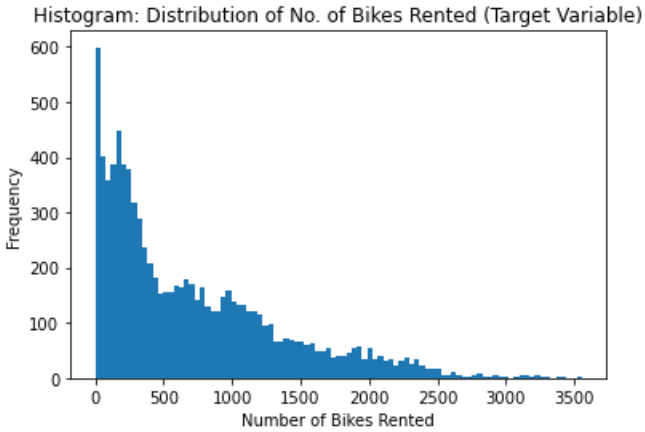



*Figure 1: Histogram depicting the distribution of the number of bikes rented*

I leverage regression tools to predict the number of bikes rented on a given day for a set of weather conditions.

**Understanding the data**

**Number of Rows:** 8760; data available for every hour of the day for 365 days.
**Number of Columns:** 14 columns; each describe the characteristic of the day/hour when the bike rental took place.

**Number of datapoints:** 1,22,640

| Feature Name | Description | Distribution | Data Type | No. of unique Records |
|---|---|---|---|---|
| Date | Date on which the bike was rented. (December 2017 - December 2018) Format- dd/mm/yyyy | Continuous | object | 365 |
| Rented Bike count (Target Variable) | Count of bikes rented at each hour on a given date | Continuous | Integer | 2166 |
| Hour | Hour of the (24 hour clock) | Categorical | Integer | 24 |
| Temperature | Temperature calculated in degree Celsius at a specific hour for a specific date. | Continuous | Float | 564 |
| Humidity | Percentage point level of humidity for a given hour in a given day | Continuous | Float | 90 |
| Windspeed | The speed (m/sec) at which the wind is blowing for a given hour in a given day | Continuous | Float | 65 |
| Visibility | Level of visibility (tens of metres) for a given hour in a given day | Continuous | Integer | 1789 |
| DewPoint Temperature | Temperature to which air must be cooled to become saturated with water vapour. Measured in degree Celsius. | Continuous | Float | 556 |
| Solar radiation | Electromagnetic radiation emitted by the sun (MJ/m2) | Continuous | Float | 345 |
| Rainfall | Amount of rainfall (mm) | Continuous | Float | 61 |
| Snowfall | Amount of snowfall in a given day on a given hour measured in cm | Continuous | Float | |
| Seasons | The four seasons that occur in Seoul | Categorical | Object | 4 (Winter, Spring, Summer, Autumn ) |
| Holiday | Whether the day is a holiday or not. This can mean weekends or national holidays. | Categorical | Object | 2 (Holiday, No Holiday) |
| Functioning Day | Whether the day is a working day for the Bike renting company or not | Categorical | Object | 2 (Yes, No) |

**Literature Review**

- *Sathishkumar V E & Yongyun Cho (2020) A rule-based model for Seoul Bike sharing demand prediction using weather data, European Journal of Remote Sensing, 53:sup1, 166-183, DOI: 10.1080/22797254.2020.1725789*

  The increasing popularity in bike sharing for environmental reasons has led to several studies on bike sharing systems especially in metropolitan cities. These systems aim to predict bike renting demand by combining weather features and geospatial factors. The predictions have enabled researchers to devise more efficient systems of managing bike rental stations, procuring optimal levels of stock and redistributing inventory from one geographic zone to another.

**II. Exploratory Data Analysis**

This section highlights some interesting graphs I encountered during EDA.

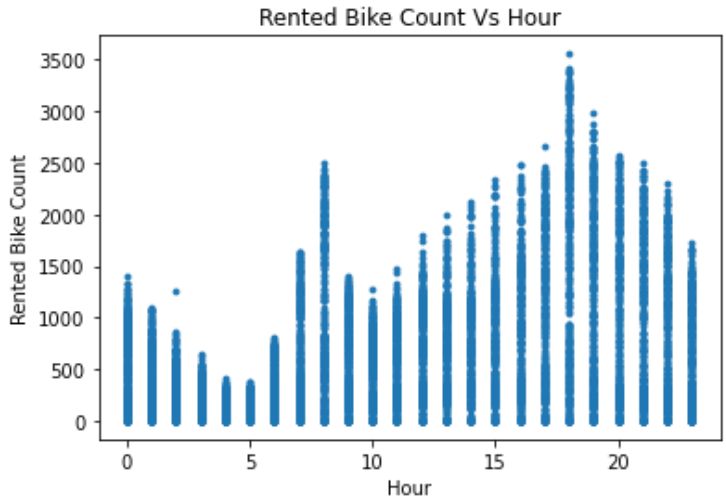



*Figure 2: Number of bikes rented at every hour during the day*

The bimodal distribution in Figure 2 makes it an interesting one.

- **x-axis:** Time (24 hour clock)
- **y-axis**: Number of bikes rented

The scatter plot has two peaks- one at 8am and the other at 6pm. The sudden rise in bike demand at 8am and 6pm each day can probably be attributed to people using the bikes to travel to and from their workplace/school. This distribution shed light on the "peak hours" and can influence the company's daily operations.

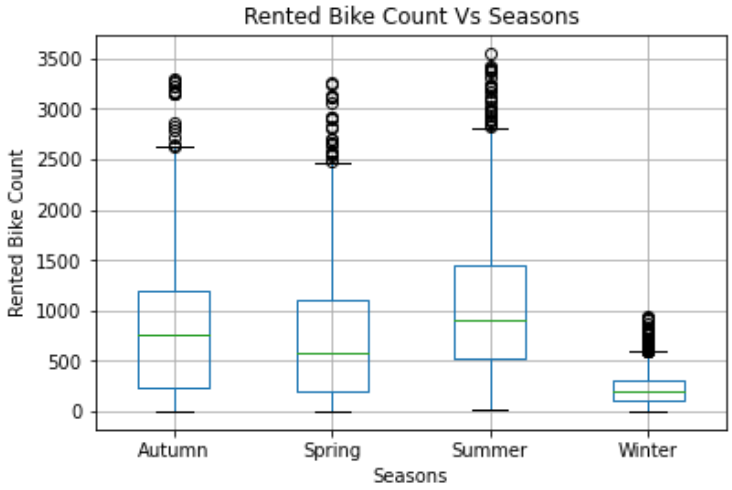



*Figure 3: Number of bikes rented in different seasons*

Figure 3 represents the difference in the distribution of number of bikes rented for different seasons.

- **x-axis:** Seasons in Seoul
- **y-axis**: number of bikes rented

As intuition suggests, in winter the median value of bikes rented is much lower than the median values of the other seasons. Again, such insights will drive stock procurement decisions.

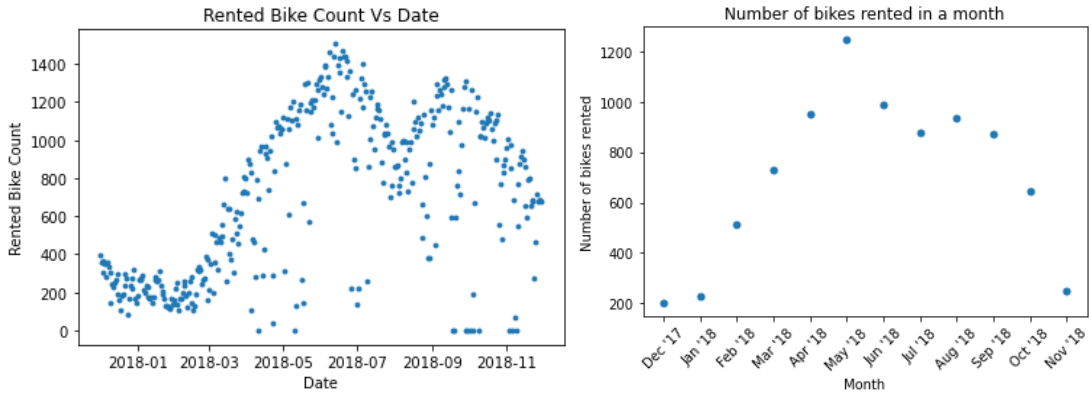



*Figure 4: Number of bikes rented across days/months*

In Figure 4, the plots are essentially showing the same thing; the only difference is the level of granularity at which the EDA was done. The plot on the left focuses on days and the right one at months.

- **x-axis**: Dates (December 2017 - November 2018)
- **y-axis**: Number of bikes rented

Towards the end of the year 2017, the demand for bike renting started growing almost exponentially until July, 2018. It followed a sudden drop in demand that persisted until September, 2018. This could be attributed to some policy change or altered consumption patterns (like a drop in fuel prices). Nevertheless, after a short period of a dry spell for the rental company, the demand for bike renting sees a steep growth again and then a decline. The cyclical fluctuations probe us to dig deeper and ascertain the underlying causes.

## III. Methods

### Data Splitting

Initially my dataset was a time series one because in order to predict the number of bikes rented on a particular day *u*, predictions would be contingent on historical data (days *v,w, z* that occurred prior to *u* ).

However, since I'm predicting the number of bikes rented given certain weather conditions, I dropped the "Date" feature. I use the remaining variables in the feature matrix (X). The target variable (y) is the column "Rented Bike Count".

Dropping the "Date" column from the dataset, makes each data point in every column IID because they all are generated from the same distribution and the way it is generated is not dependent on the value generated preceding it; hence the dataset is IID.

I used the basic train-test-val splitting strategy with 80% of the datapoints as training data, and 10% of the datapoints for each testing and validating the results of the model. Since there are ~8000 points, choosing the 80-20 split seemed reasonable as it leaves a significant number of data points for the test and the validation sets.

### Data Pre-processing

Below is the table that shows the pre-processors I used on the features and the number of feature columns generated post pre-processing:

| Pre-processor | Feature Names | Number of feature columns |
|---|---|---|

| | | |
|---|---|---|
| One hot encoder (categorical and unordered) | Seasons, Holiday, Functioning Day | 8 |
| Standard scalar (continuous and tailed distribution | Hour, Temperature, Humidity, Wind Speed, Visibility, Dew Point Temperature, Solar Radiation, Rainfall, Snowfall. | 10 |

*Figure 5: Data after pre-processing*

**ML Pipeline**

Below is a table summarizing the five different ML algorithms I have used to predict the target variable, the parameters I have tuned in each model and the values used for tuning:

| ML Algorithm | Parameters Tuned | Parameter Values |
|---|---|---|
| Lasso Regression | alpha | 50 values between log (-10) and log (10) |
| Ridge Regression | alpha | 50 values between log (-10) and log (10) |
| Random Forest | max_depth | [1,3,10,30,100] |
| KNN | n_neighbors | [1,3,10,30, 100, 500] |
| | weights | ['distance'] |
| XGBoost | max_depth | [1,10,50,75,100] |
| | learning_rate | [0.01] |
| | n_estimators | [10000] |
| | missing | [np.nan] |
| | colsample_bytree | [0.6] |
| | subsample | [0.5] |

*Figure 6: Summary of ML Models Performance*

**Evaluation Metric**

Since I'm dealing with a regression problem, I have used root mean squared error (RMSE) as my evaluation metric. The model which has the lowest RMSE performs the best relative to the others. The next section focuses on the results of each of these models. The RMSE can be defined as follows:

$$\frac{\sqrt{\sum n = i^{n} (y_{predicted} - y_{observed})^2}}{\sqrt{n}}$$

**Results**

The table below summarises the results of each ML Model and presents a comparative study of the baseline test scores and the test scores derived from the different models.

Here we can see that the best model is the XG Boost model. The lowest test RMSE score (213.535) is derived from the XG Boost model as can be seen in the figure below.

| ML Algorithm | Random State | Baseline Test Score | Best Test Score | Validation Score |
|---|---|---|---|---|

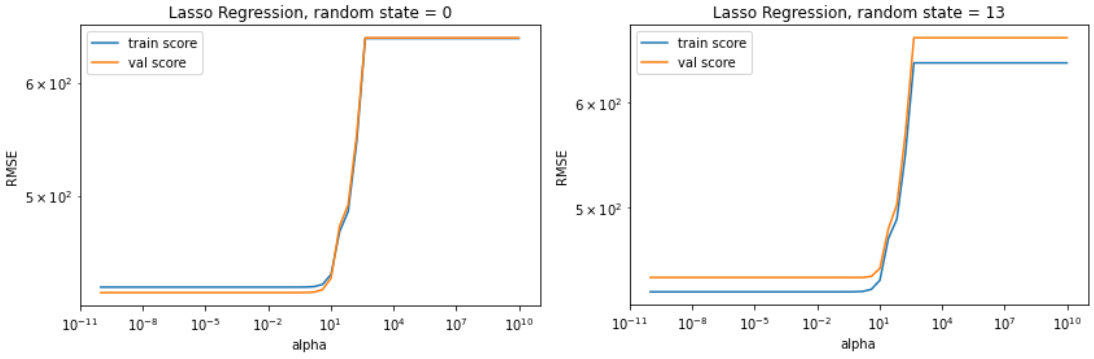
| | | | | |
|---|---|---|---|---|
| Lasso Regression | 0 | 649.643 | 445.329 | 427.667 |
| | 13 | 637.469 | 428.116 | 442.687 |
| | 33 | 642.924 | 438.463 | 421.261 |
| | 86 | 662.748 | 445.970 | 439.398 |
| | 150 | 608.243 | 397.632 | 424.822 |
| | | | | |
| Ridge Regression | 0 | 649.643 | 445.330 | 427.667 |
| | 13 | 637.469 | 428.115 | 442.687 |
| | 33 | 642.924 | 438.685 | 421.261 |
| | 86 | 662.748 | 445.970 | 439.398 |
| | 150 | 608.243 | 397.632 | 424.822 |
| | | | | |
| Random Forest | 0 | 649.643 | 249.441 | 218.112 |
| | 13 | 637.469 | 233.644 | 244.297 |
| | 33 | 642.924 | 235.618 | 227.365 |
| | 86 | 662.748 | 237.143 | 240.234 |
| | 150 | 608.243 | 222.026 | 250.534 |
| | | | | |
| K Nearest Neighbours | 0 | 649.643 | 311.606 | 293.419 |
| | 13 | 637.469 | 285.312 | 295.474 |
| | 33 | 642.924 | 293.798 | 296.525 |
| | 86 | 662.748 | 307.576 | 306.901 |
| | 150 | 608.243 | 281.942 | 299.235 |
| | | | | |
| XG Boost | 0 | 649.643 | 240.493 | 209.426 |
| | 13 | 637.469 | 224.493 | 235.052 |
| | 33 | 642.924 | 224.134 | 220.316 |
| | 86 | 662.748 | 227.117 | 228.816 |
| | 150 | 608.243 | 213.535 | 241.358 |

*Figure 7: Comparison between Model's Best Test Score and Test Baseline Score*

Let us briefly go over each model and its results:

1. **Lasso Regression**

   The figures below summarize the findings of the Lasso Regression model. The close overlap between the train and the validation curves suggests that the model is trained well and it fits well on unseen data. However the test RMSE scores of the Lasso regression as shown in the table above are significantly high and hence this is not the bets model for our data.



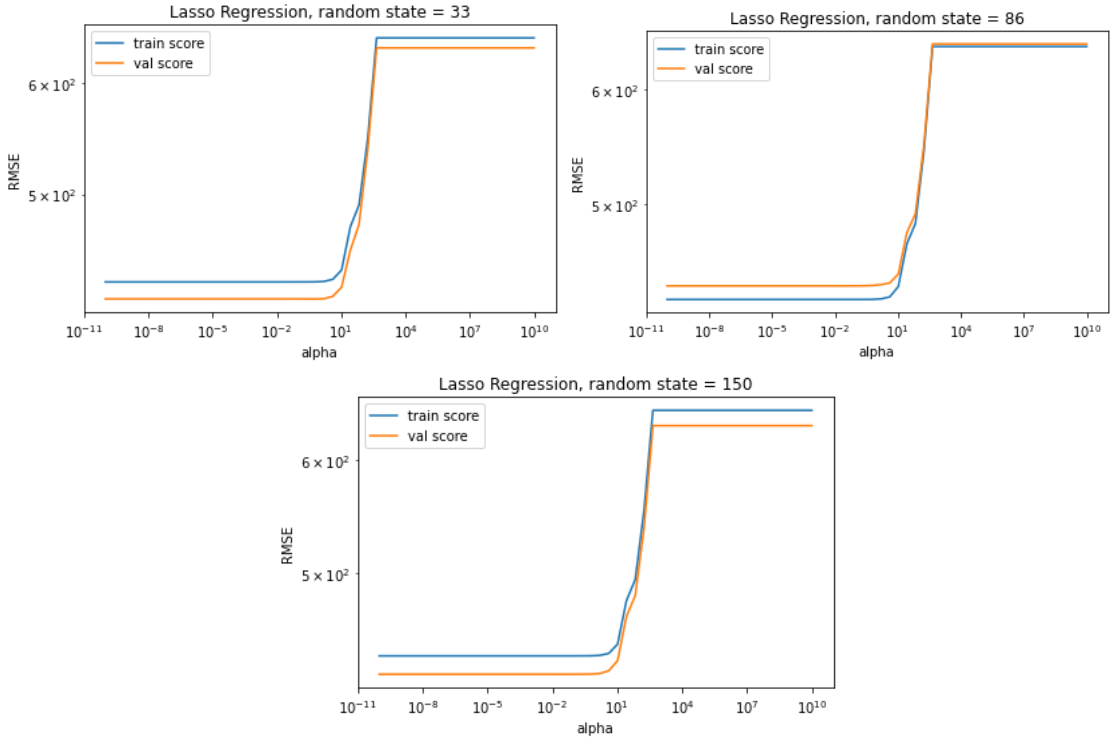



*Figure 8: Lasso Regression*

## 2. Ridge Regression

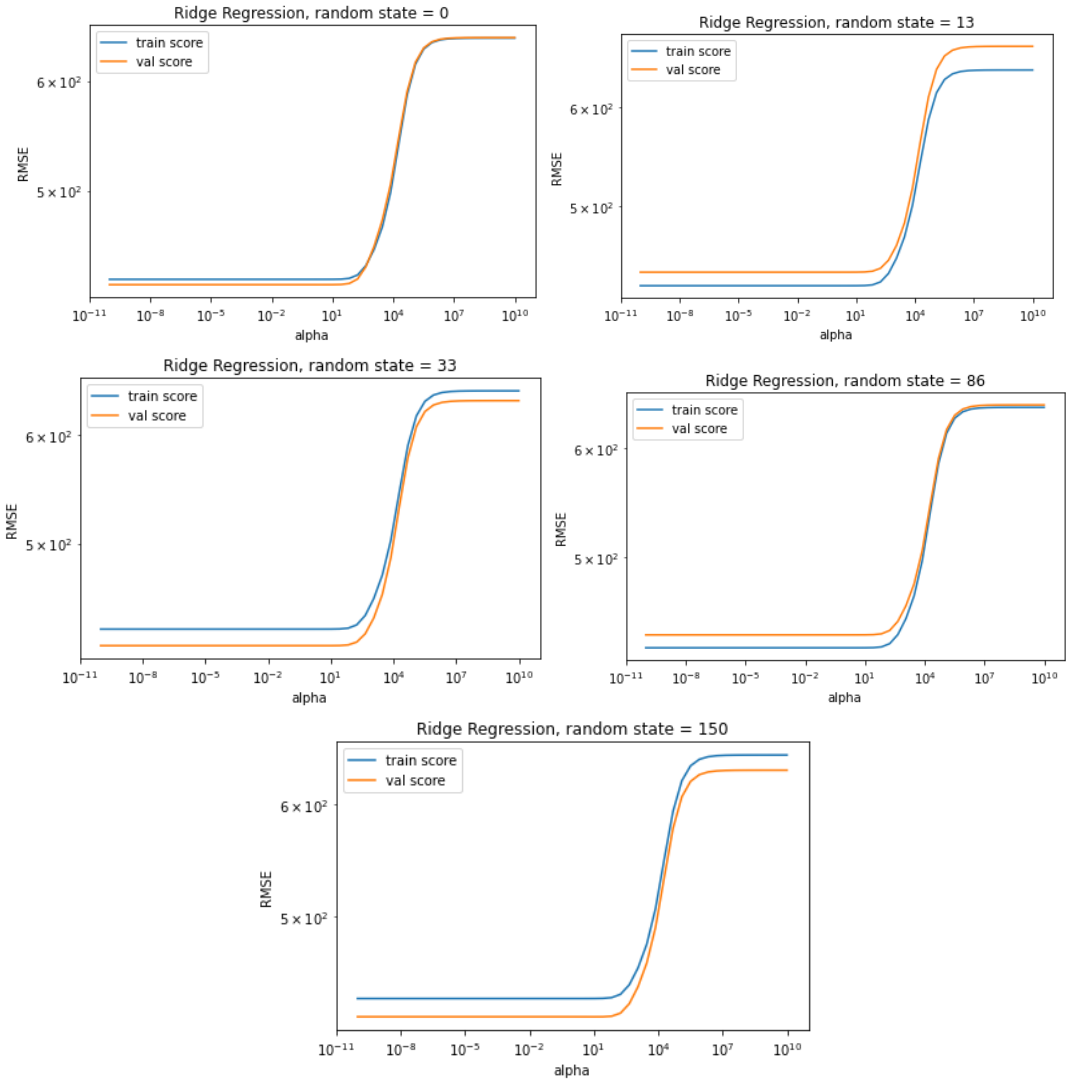
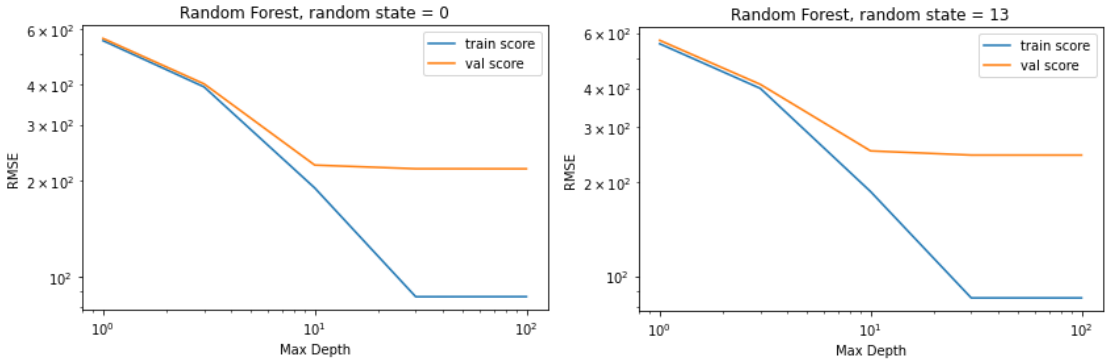



*Figure 9: Ridge Regression*

We have a very similar interpretation for Ridge regression too. While, the model seems to be doing well on unseen data, the test RMSE score is relatively high. Since there are other models that have better scores than Ridge Regression, we don't use this model.

## 3. Random Forest



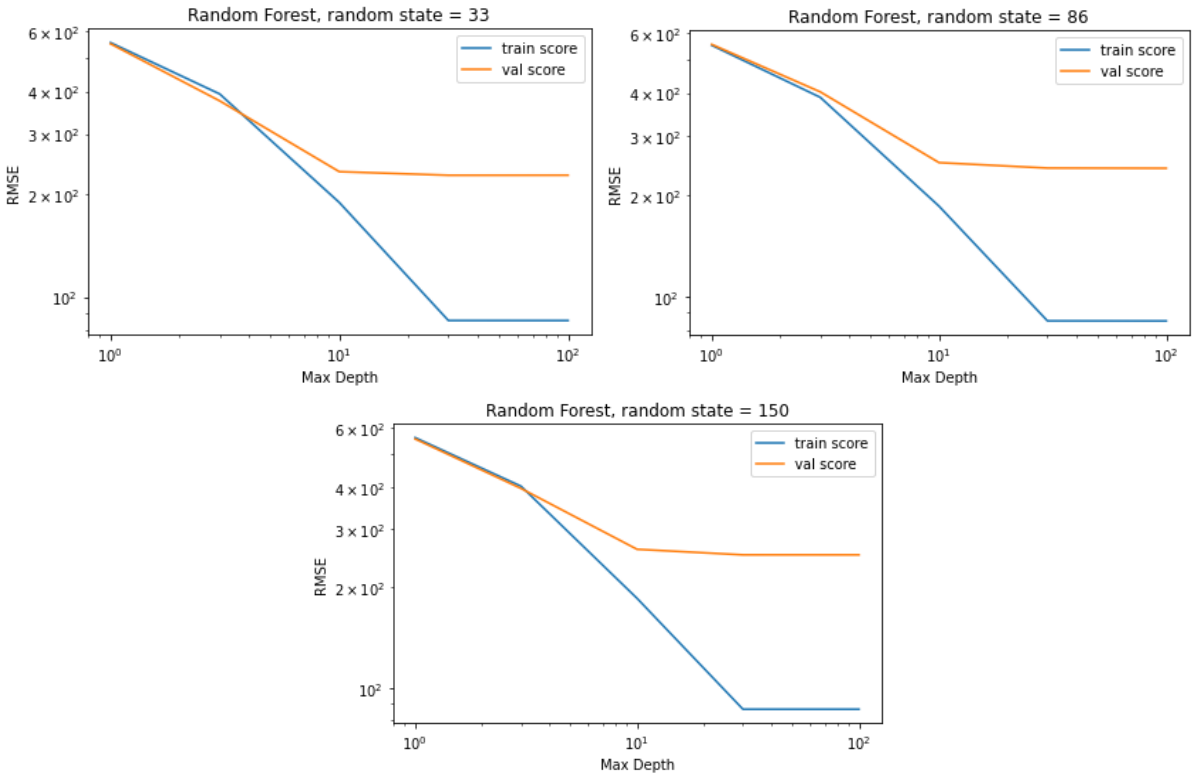



*Figure 10: Random Forest*

In this model, the huge gap between the train and validation scores as the max_depth increases, suggests that the model is overfitting the train data. Thus this is not a great model.

## 4. K Nearest Neighbours

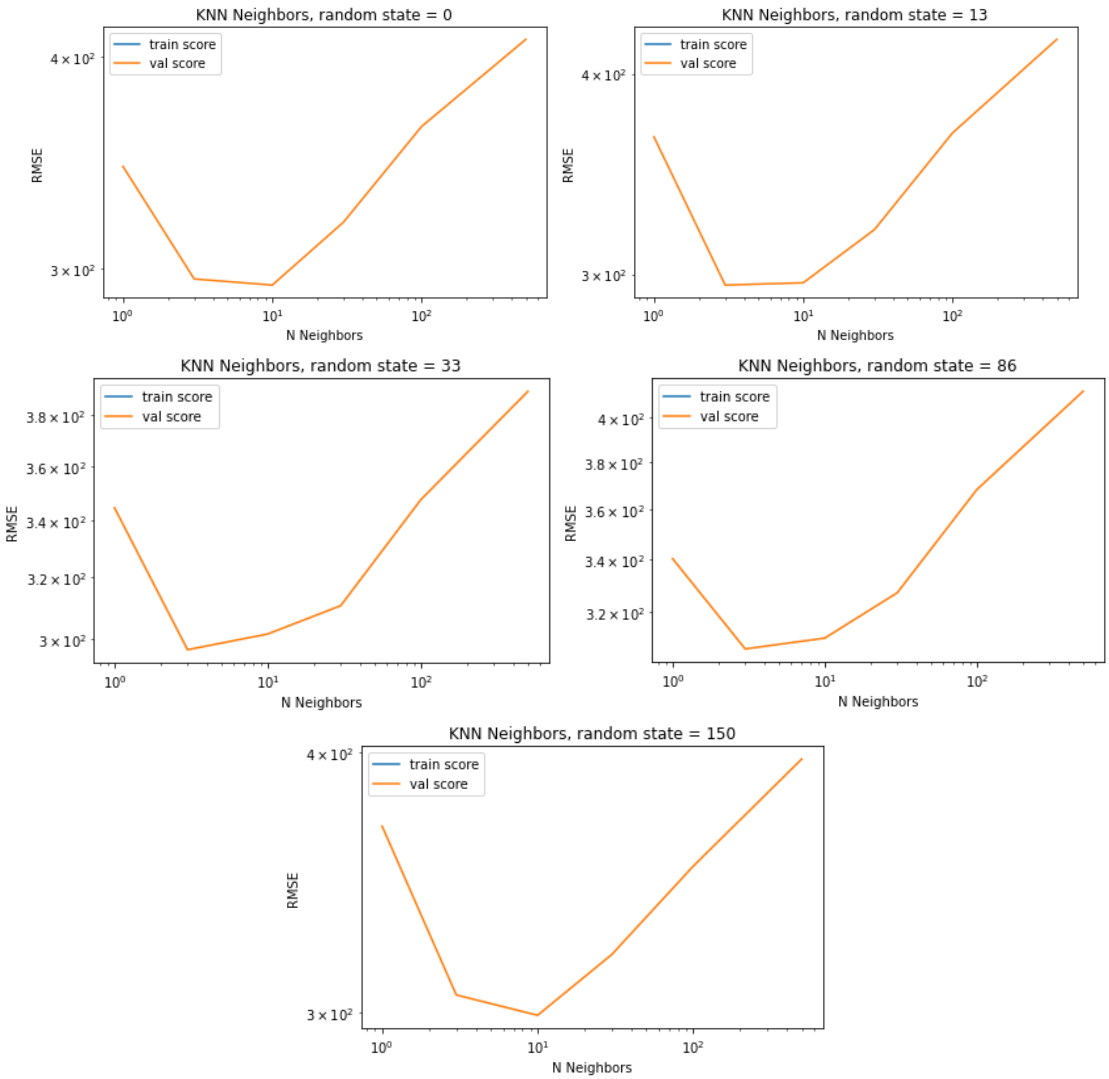



*Figure 11: KNN Regression*

The lack of a train score curve suggests that the data has been overfit so much so that the train RMSE score is 0. Thus this model does not work well with our data.

## 5. XG Boost

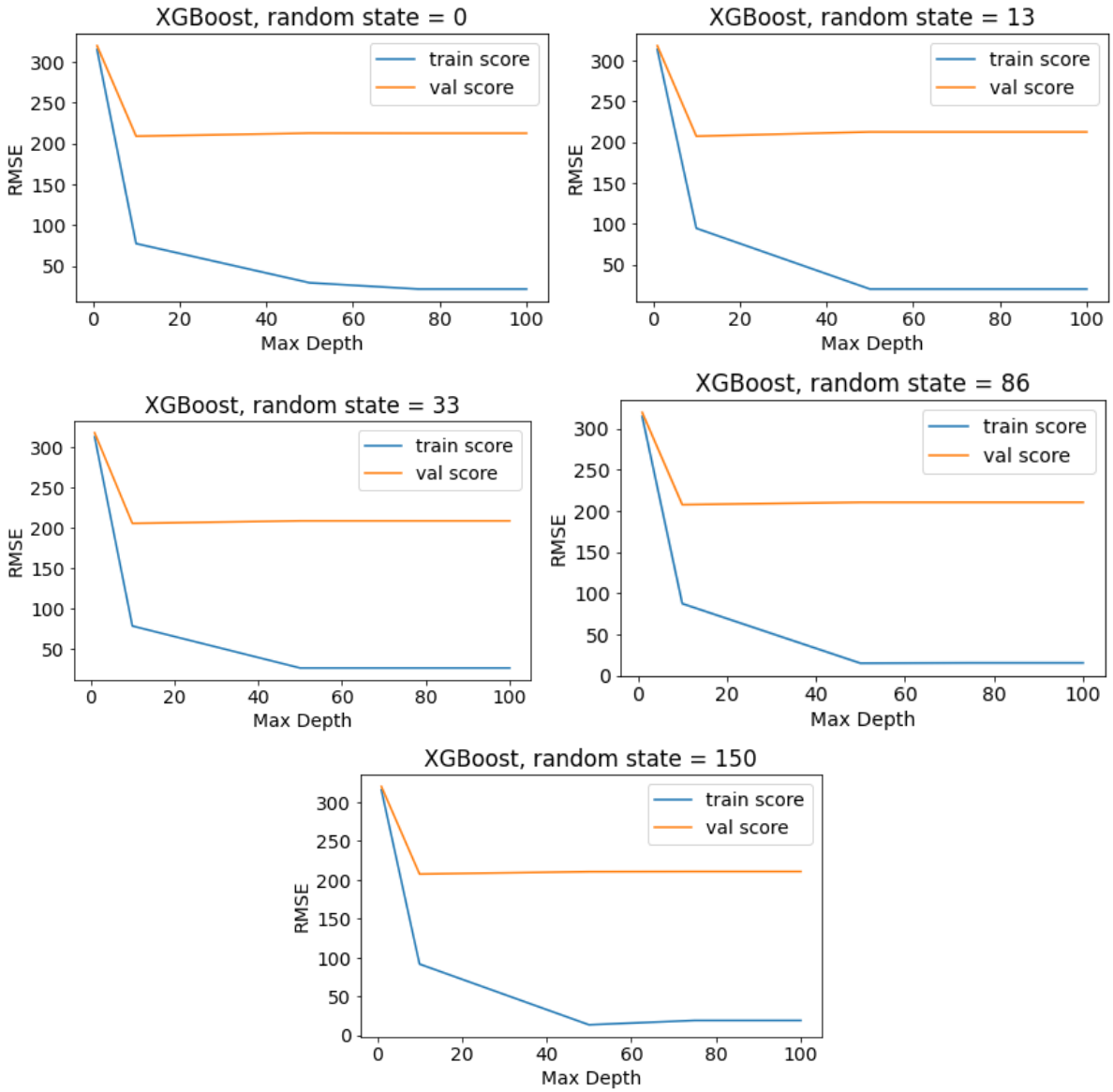



*Figure 12: XGBoost*

While there exists a significant gap between the train and validation score that can be explained by overfitting, since the test RMSE scores are significantly lower than all the other models and the baseline score, this is the best possible model we have.

**Global Feature Importance**

The plots below show the most important features using three different interpretability models:

1. Feature Importance using Importance Score

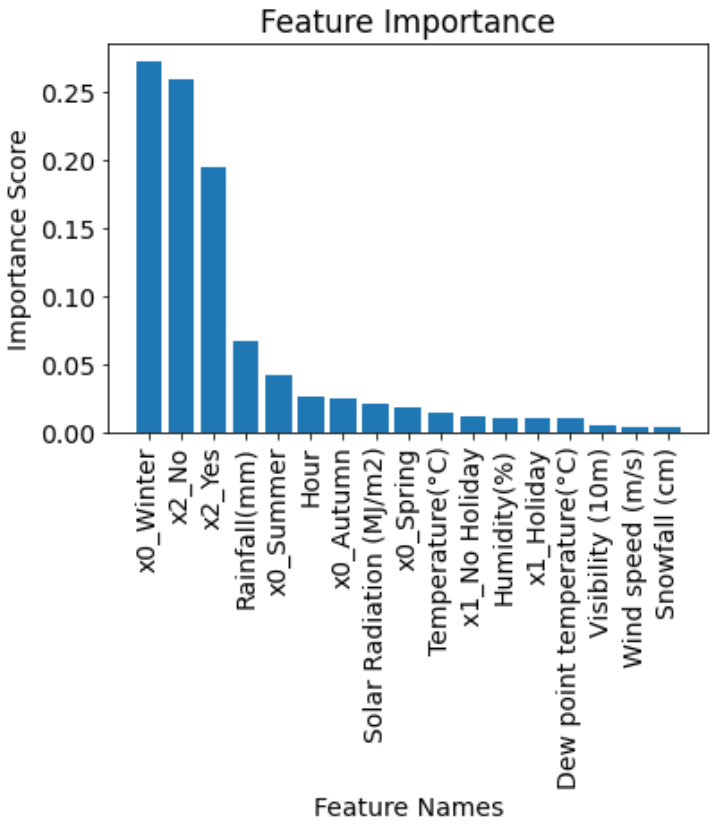
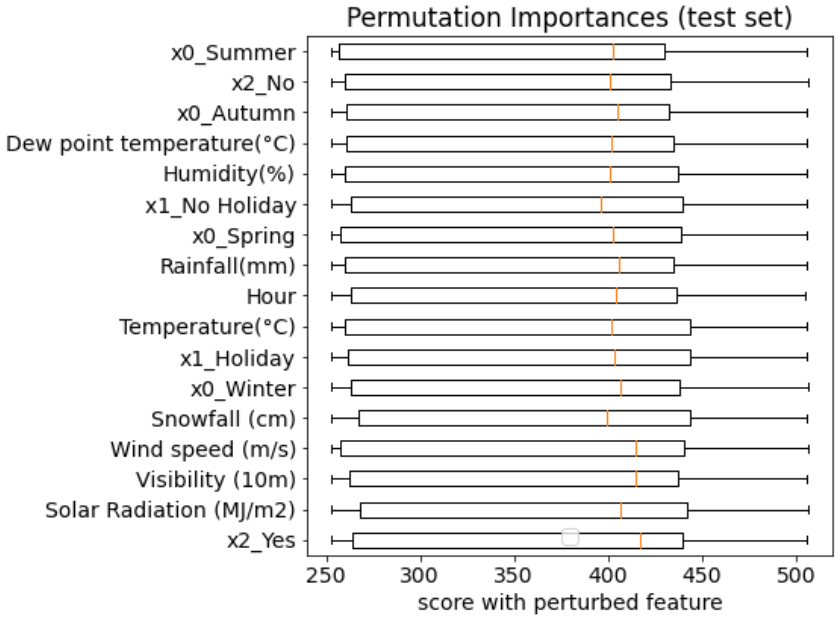



*Figure 13: Feature Importance*

This graph shows the importance of features in hierarchical order in predicting the number of bikes rented. The top 3 features are: Winter Season, No holiday, Yes holiday.

2. Feature Importance using Permutations




*Feature 14: Feature Importance (Permutation)*

This graph shows that almost all the features are equally important for predicting the number of bikes rented.

**SHAP**

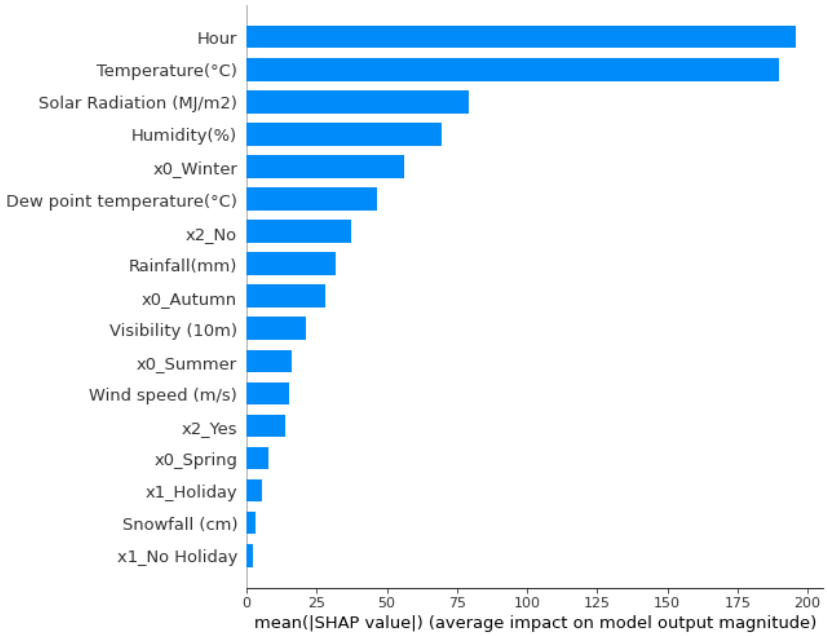



*Figure 15: SHAP*

The figure here explains locally what is the most important feature in predicting bike rental demand.

**Outlook**

Although there are some discrepancies in what is considered the most important feature this is probably explained by the fact that each of these methods are using different algorithms to determine "importance score".

Further, I think if we could use a lower learning rate in the XGBooost model or increasing the early_stopping rounds, would enable us to get better results. I tried that however the score didn't converge.

The model performance can further be improved if we can get more data so that we have a larger dataset over a larger time period. This will allow us to control for any other factors that affect bike rental demands that were specific to the year the data was collected.

Using more global interpretability methods to interpret feature importance and comparing the results will help modellers better understand which features affect bike rental demand more than others.

**References**

*Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.*

*Sathishkumar V E & Yongyun Cho (2020) A rule-based model for Seoul Bike sharing demand prediction using weather data, European Journal of Remote Sensing, 53:sup1, 166-183, DOI: 10.1080/22797254.2020.1725789*