BIKE SHARING DEMAND PREDICTION

Data Science Pro
AlmaBetter, Bangalore.

Tanay Tupe

BIKE SHARING DEMAND PREDICTION (SEOUL)

For this Regression project we we are going to use " SEOUL Bike sharing Data Set ". This data set has around "8760" observations/listings in it with 14 columns in it's raw form. This data set is a mix between categorical and numeric values. These observations also have various features like Date of each observation, hourly stamp, bikes rented in hour from "00:00" to "23:00" few natural factors affecting hourly rents like temperature, wind speed, solar radiation, humidity, visibility and more.

ABSTRACT

A bike rental or bike hire business rents out bicycles for short periods of time, usually for a few hours. Most rentals are provided by bike shops as a sideline to their main businesses of sales and service, but shops specialize in rentals.

Renting bike for short time business grew massively in recent years and have recorded a huge demand in bikes rents and bookings. This enormous number of renting creates enormous amount of data, data which can be studied and used to predict renting patterns, make meaning full Business Decisions, improve hourly supply, control traffic on application on peak hours, apply strong marketing strategies, improve service for users and ultimately grow business.

PROBLEM STATEMENT

Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time.

Eventually, providing the city with a stable supply of rental bikes becomes a major concern. The crucial part is the **prediction of bike count required at each hour** for the stable supply of rental bikes.

- **Date**
  Date recorded for a particular observation.
  (Date-Time)
- **Rent Bike Count**
  Hourly bikes rented record.
  (continuous) ***(Target variable)***

- **Hour**
  24-Hours record for each day. values from “00:00” to “23:00”.
  (Discrete)

- **Temperature**
  Temperature recorded hourly in Celsius.
  (continuous)

- **Humidity**
  Water content in atmosphere in percentage.
  (Discrete)

- **Wind speed**
  Velocity of wind recorded in meter per second.
  (continuous)

- **Visibility**
  The degree of clearness of the atmosphere.
  (continuous)

- **Dew point temperature**
  Temperature below which the water vapour will condense into liquid water recorded hourly.
  (continuous)

- **Solar Radiation**
  Electromagnetic radiation emitted by the sun. In watt per square metre.
  (continuous)

- **Rainfall**
  Hourly record of water poured in “mm”.
  (continuous)

- **Snowfall**
  Hourly record of snow poured in “cm”.
  (continuous)

- **Seasons**
  Each of the four divisions of the year.
  (Categorical)

- **Holiday**
  Record of working day and non-working day
  (Categorical)

- **Functioning Day**
  No Func(Non Functional Hours), Fun (Functional hours).
  (Categorical)

INTRODUCTION

Bike rental is a Business that operates online which primarily rents bike for short duration for locals and tourists. Data set is recorded from Seoul, South Korea. The platform is accessible via mobile app. Bikes can be rented from various bike stands across city, mainly near bus, train or metro station. As with car rental, bicycle rental shops primarily serve people who do not have access to a vehicle, typically travelers and particularly tourists.

Our goal here is to predict hourly bike rents from "Bike Rental (Seoul)" data set using EDA techniques, implementing different regression models which will present a basic insight of our data and be able to predict bike required in hourly basis. Also, we will be answering some questions and visualize few trends using Numpy, pandas, matplotlib and seaborne.

STEPS INVOLVED/MODULES

1. Importing necessary libraries, Cleaning and understanding data.

- Importing required packages for data manipulation.(Numpy, Pandas, matplotlib and seaborne)
- Mounting our notebook to drive and Importing **" SEOUL Bike sharing Data Set "** data set into notebook. For this analysis we are using google colaboratory.
- Finding shape of our data set to get number of rows and columns. In this data set we have "8760" columns and '14' rows.(From unmodified/raw data)

- Analyzing some basic information of data set (using .info function). Also, we have three different data types (int64, object, float64) occupying around '958.2+ KB' of memory.
- We encounter "0" Null values In this set. Though it's highly unlikely to encounter such data sets. Here it works in our favour.
- Next, we Checked for duplicate records/observations for termination. But surprisingly, this data set had no duplicate observations.
- We also checked for number of unique values for each column in order to identify categorical features of this data set.
- Using describe function to check the min/max values, mean, standard deviation and spread of numeric columns.
- Now, we are done with cleaning, and our data is now ready for analysis.

2. Exploratory Data Analysis.

Exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods.

We began our EDA with univariate analysis. Where checks have been performed for type of distribution, skewness and outliers.

Considering outliers, we encountered outliers in few important features like "Solar Radiation" and "rainfall" even our target/dependent variable. We can clearly see in fig 2.1 that some variables have a lot of outliers.

Still, we will be keeping them in our analysis and check how models fits. Reason we are considering outliers is because few important variables like "Rainfall", "Snowfall" have a ton of them, and deleting them may result in information loss from data set.

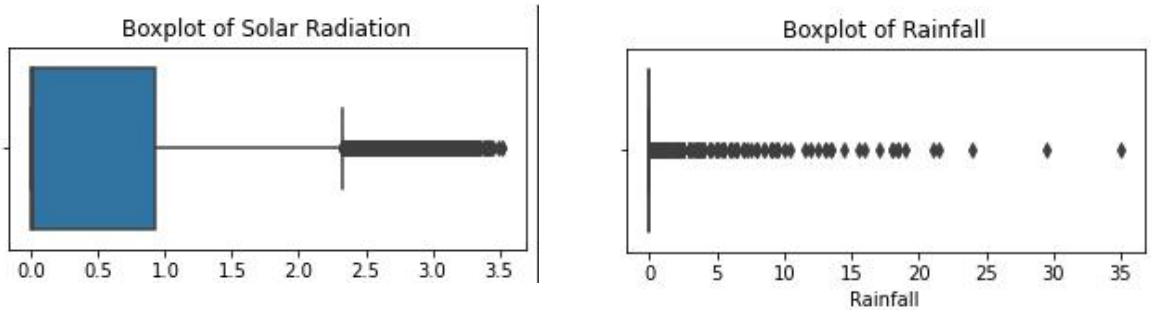


Figure 2.1

Analyzing distributions of variables.

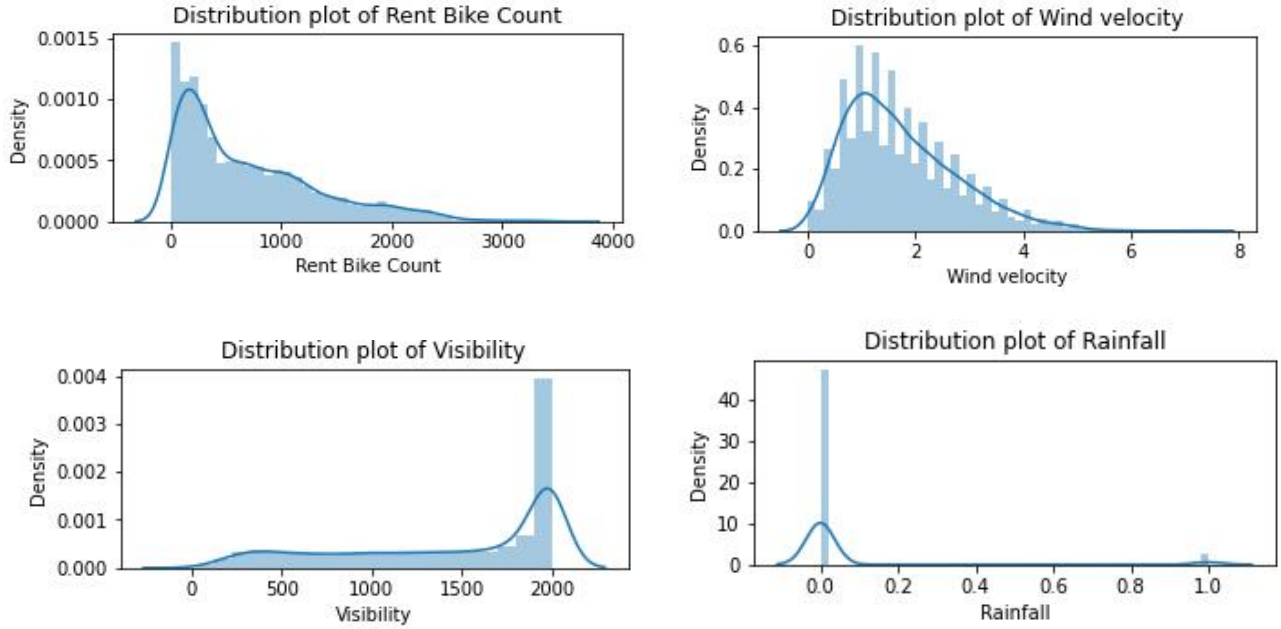
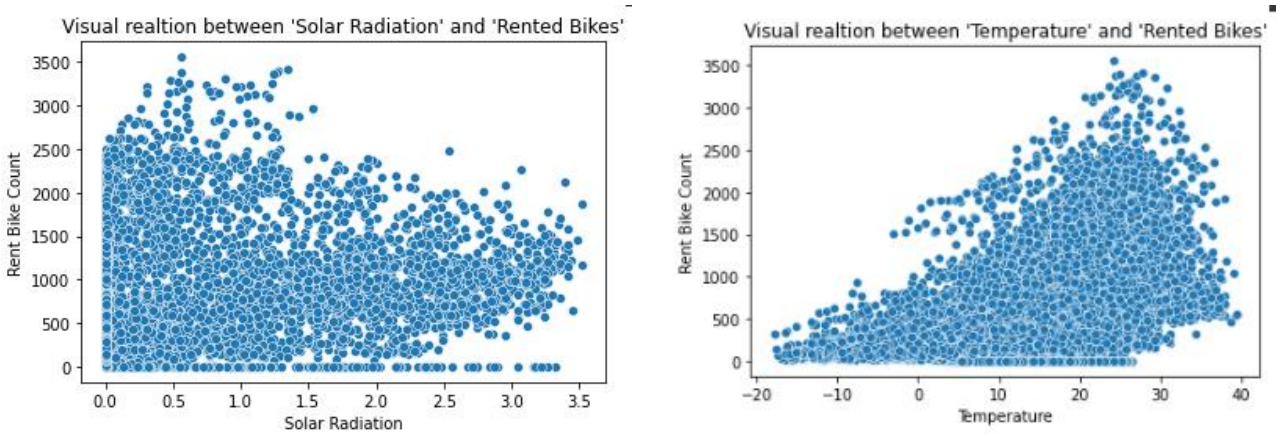


Figure 2.2

Here in figure 2.2, we can witness most of the distribution are not normal, skewness can be found in every other variable. Initial guess here is " Tree based models will work better than linear regression ".

Bivariate analysis. Checking for linear relationship between dependent and independent variables.

Figure 2.3



In above figure 2.3 we can again observe that, there ain't enough linear relation between dependent and independent variables. As thought Linear Regression will be an issue on this data set. But again, our main focus will be on Tree based models for this analysis.

We also performed check for multicollinearity using “Variance Inflation Factor”. Here we can clearly see "Temperature" and "Dew.P Temperature" have a little high "VIF". "VIF" can be a problem considering Linear Regression, but Tree based models will not be affected with multicollinearity.

Top features with high "VIF"

Temperature	33.3
Dew.P Temperature	17.1
Visibility	9.0

Using descriptive statistics we found that, bikes are mostly rented on "8:00 AM" and "5:00 PM" to "10:00 PM". Most people who rents bike might be working professional and office staff with no personal transport, as the peak rent time matches with office timings.

We also observed a significant drop in rentals during winter seasons. Reason being, temperature averages to "-2 Deg Celsius" in winters with records as low as "-17 Deg Celsius". Whereas in summers temperatures are quite favourable averaging to "26.5 Deg Celsius".

3. Preparation for prediction model

- **Vectorizing continuous variables.**

  Solar Radiation : continuous feature categorized to "0" and "1". Split point on mean of data.

  Rainfall : continuous feature categorized to "0" and "1". Split point on mean of data.

  Snowfall : continuous feature categorized to "0" and "1". Split point on mean of data.

- **Vectorizing categorical variables.**
  Holiday : Mapping “No Holiday” to “0” and “Holiday” to “1”.
  Functioning Day : Mapping “No” to “0” and “Yes” to “1”.

- **Finally Vectorizing using Dummies**.
  Seasons : Divided into “Seasons_Autumn”, “Seasons_Spring”, “Seasons_Summer”, “Seasons_Winter”

Finally we scaled our feature variables and split data into independent and dependent set and Splitting data into Train and Test sets. Now we are ready to pass this set to desired regressors.

4. Implementing regression models

- **Implementing Linear Regressor**

  The reason we are passing data to a Linear regressor is just to prove that these type of regression models won’t actually fit very closely to target variable in data set like these. As it was evident enough during analysis. Still, let’s pass data to regressor and verify using evaluation metrics. We will use Cross-Validation for Linear regressor. And, instead of normal Linear Regression, we'll be using Regularized Regression.

Evaluation Metrics for Lasso Regression

Best Alfa	0.1
Train RMSE	423.3
Test RMSE	431.6
Train R-sq	0.56
Test R-sq	0.55

- **Implementing Decision Tree Regressor**

  As decision tree regressor was our initial guess, let's see how this regressor performs. Decision tree is a flowchart-like structure and splits the data on Entropy or Information gain, which is a complete different approach unlike mapping a function in Linear regression. Let's pass data to regressor and verify using evaluation metrics.

Evaluation Metrics for Lasso Regression

Train RMSE	0.0
Test RMSE	320.18
Train R-sq	1.0
Test R-sq	0.75

- **Implementing Random Forest**

  As we saw in previous slide, Decision Tree did a better job then Linear regression. But using single tree approach was not very beneficial. Reason being, the Decision Tree over-fitted to data way too much on training set. We can play and experiment with different hyper parameters or we could consider Random Forest to deal with overfitting.

Evaluation Metrics for Random Forest

Train RMSE	84.04
Test RMSE	235.67
Train R-sq	0.98
Test R-sq	0.86

- **Implementing Extreme-Gradient Boosting**

  XGBoosting is an implementation of gradient boosting trees algorithm. We are trying to close the variance gap by using another variation of tree based Ensemble technique. Let's check the metrics and compare all the metrics from previous algorithms.

Evaluation Metrics for Random Forest

Train RMSE	245.24
Test RMSE	260.53
Train R-sq	0.85
Test R-sq	0.83

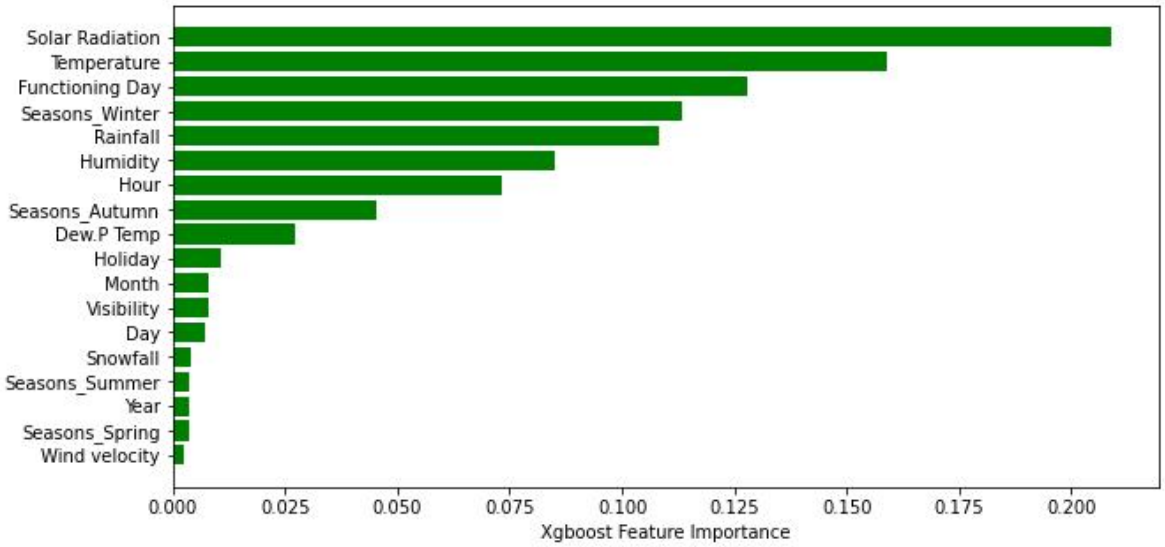
As evident as it is from all regression types tested, we finally come to conclusion that XGBoosting works very well to fit train and test set closely (Low variance) also having a good amount of R-sq value.

Evaluation metrics Comparison table

	Train RMSE	**Test RMSE**	**Train R-sq**	**Test R-sq**
Lin. R	423.3	431.6	0.56	0.55
Dec.Tree	0.0	320.18	1.0	0.75
Ran. F	84.04	235.67	0.98	0.86
XGB	245.24	260.53	0.85	0.83

- **Feature Importance**

  Feature Importance refers to techniques that calculate a score for all the input features for a given model. The scores simply represent the "importance" of each feature. A higher score means that the specific feature will have a larger effect on the model that is being used to predict a certain variable. Below we have a bar-plot representing feature importance in sorted order.



As we can see her, our top features which were important for predicting target variable are "Solar Radiation", "Temperature", "Functioning Day" and "seasons".

CONCLUSION

Decision tree regressor was our initial guess, which is a complete different approach unlike mapping a function in Linear regression. It did a better job then Linear regression. But using single tree approach was not very beneficial. Because Decision Tree over-fitted to data way too much on training set, resulting in high variance.

To overcome this variance we could have experimented with hyper parameters or go with Ensemble techniques, and chose to go with Random forest models. But the model still had a hint of variance.

To close this variance gap even more and generalize our model we used XGBoost regressor. where we finally had promising metrics which were generalized and had minimal variance. Here model had lowest variance and best R-square value for train and test set.

Most important feature according to XGBoost model are “Solar Radiation”, “Temperature”, “Functioning Day” , “Season (winter)” and “Rainfall”.