Bike Renting

Prity Vijay

29 April 2019

**Chapter 1**

**Introduction**

* 1. **Problem Statement**

Bike sharing systems are increasingly deployed in many cities worldwide. These systems provide a prominent solution for the first/last-mile problem with their cost-effectiveness and eco-friendliness. However, the use of bikes among stations is often spatiotemporally imbalanced, causing many problems in daily operations (e.g., rebalancing challenges). Thus, knowing how the system demand evolves in advance helps improve the preparedness of operational schemes. As the demands for bikes rent is increasing therefore manually detecting number of bike availability is next to impossible. And hence system needs to be automated with a utilization of machine learning algorithm.

* 1. **Data**

The problem discussed above is a regression problem. Our objective is to predict bike rental count on daily basis, based on the environmental and seasonal settings. Given below is a sample of the data set that we are using to predict the bike rent count.

**Table 1.1**: Bike Rent Sample Data (Columns: 1-8)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| instant | Dteday | season | yr | mnth | holiday | Weekday | workingday |
| 1 | 01-01-2011 | 1 | 0 | 1 | 0 | 6 | 0 |
| 2 | 02-01-2011 | 1 | 0 | 1 | 0 | 0 | 0 |
| 3 | 03-01-2011 | 1 | 0 | 1 | 0 | 1 | 1 |
| 4 | 04-01-2011 | 1 | 0 | 1 | 0 | 2 | 1 |
| 5 | 05-01-2011 | 1 | 0 | 1 | 0 | 3 | 1 |

**Table 1.2**: Bike Rent Sample Data (Columns: 9-16)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| weathersit | Temp | Atemp | Hum | windspeed | casual | registered | cnt |
| 2 | 0.344167 | 0.363625 | 0.805833 | 0.160446 | 331 | 654 | 985 |
| 2 | 0.363478 | 0.353739 | 0.696087 | 0.248539 | 131 | 670 | 801 |
| 1 | 0.196364 | 0.189405 | 0.437273 | 0.248309 | 120 | 1229 | 1349 |
| 1 | 0.200000 | 0.212122 | 0.590435 | 0.160296 | 108 | 1454 | 1562 |
| 1 | 0.226957 | 0.229270 | 0.436957 | 0.186900 | 82 | 1518 | 1600 |

As we can see above table 1.1 1nd 1.2 is showing top 5 rows of the dependant and independent variable.

Let us have a name of all the columns individually. ’cnt’ is a dependant variable for which we have to develop a model for prediction of bike count.

'instant', 'dteday', 'season', 'yr', 'mnth', 'holiday', 'weekday', 'workingday', 'weathersit', 'temp', 'atemp', 'hum', 'windspeed', 'casual', 'registered', 'cnt'

* 1. **Column Description:**
* **instant:** Record index
* **dteday:** Date of bike count for rent
* **season:** Season (1:spring, 2:summer, 3:fall, 4:winter)
* **yr**: Year (0: 2011, 1:2012)
* **mnth:** Month (1 to 12)
* **holiday:** 1: Holiday, 0: Not-Holiday
* **weekday:** Day of the week
* **workingday:** If day is (neither weekend nor holiday) working day :1,otherwise:0
* **weathersit:** Weather situation during renting bike

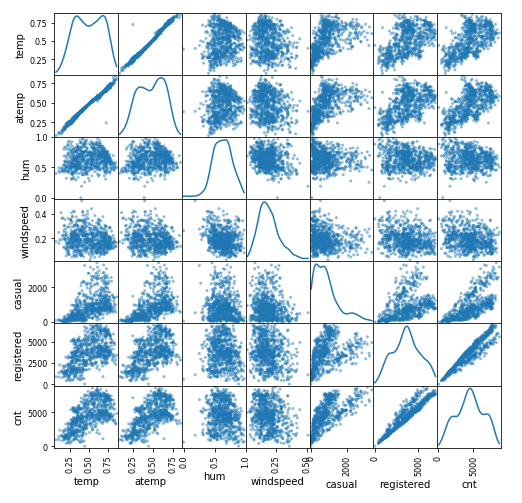
1: Clear, Few clouds, Partly cloudy, Partly cloudy

2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist

3: Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds

* **temp:** Normalized temperature in Celsius.
* **atemp:** Normalized feeling temperature in Celsius.
* **hum:** Normalized humidity. The values are divided to 100 (max)
* **windspeed:** Normalized wind speed. The values are divided to 67 (max)
* **casual:** count of casual users
* **registered:** count of registered users
* **cnt:** count of total rental bikes including both casual and registered

Let us visualize continuous variable of a sample dataset in Fig. 1.1 In a next chapter we will further analyze the data in more detail.



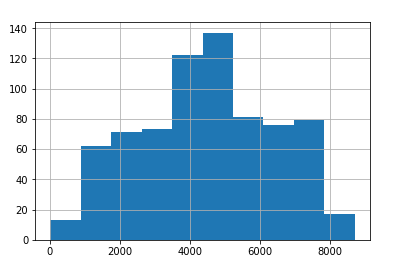
**Figure 1.1** Scatter matrix of continuous variable of a Bike Rent Dataset

**Python Code:**

pd.scatter\_matrix(data.loc[:,col],figsize=(8,8),diagonal='kde')

**Chapter 2**

**Analysis**

****

**Figure 2.1:** Distribution of dependant variable (cnt)

As we can see above our dependant variable is normally distributed. We have altogether 731 rows and 16 columns. Let us try to observe some trend among the variables.

|  |  |
| --- | --- |
|  |  |
|  | |

**Figure 2.1**: Scatter plot to check co linearity of cnt with temp

As we know cnt is our target variable our first analysis is to check how cnt and temp is related. In fig. 2.1 we can clearly notice that temp and cnt is positively correlated with each other. We can also observe increase pattern of bike count when the weather situation is clear and temp is high. The second graph of fig. 2.1 blue line is for clear weather, orange line is for mist and cloudy weather and green line is for light rain and scattered rain.

**Python Code:**

g = sns.lmplot(x="temp", y="cnt", hue="weathersit", data=data)

g = sns.lmplot(x="temp", y="cnt", hue="weathersit", data=data)

g = sns.lmplot(x="temp", y="cnt", data=data,hue="holiday")

|  |  |
| --- | --- |
|  |  |
|  | |

**Figure 2.2**: Scatter Plot to check dependency of bike count with weekdays

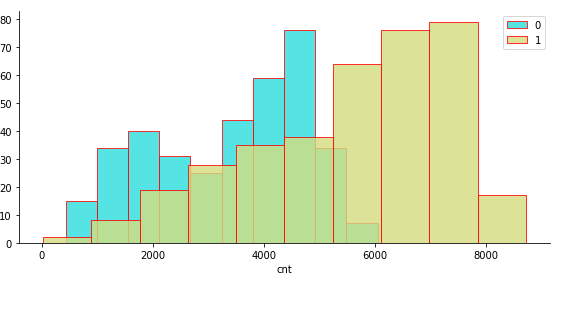
Our next analysis is to check how count is dependent on the days of week. In above figure we can observe clearly that casual user rented bike more on weekends while registered user rented bike more on weekday. In the last plot blue line denotes no holiday while orange line denotes holiday so we can clearly say that count for bike rent decreases on holiday while increases on working days.

**Python Code:**

g = sns.lmplot(x="weekday", y="casual", data=data)

g = sns.lmplot(x="weekday", y="registered", data=data)

g = sns.lmplot(x="weekday", y="cnt", data=data,hue='holiday')



**Figure 2.3**: Number of bike Rented per year(2010-2011)

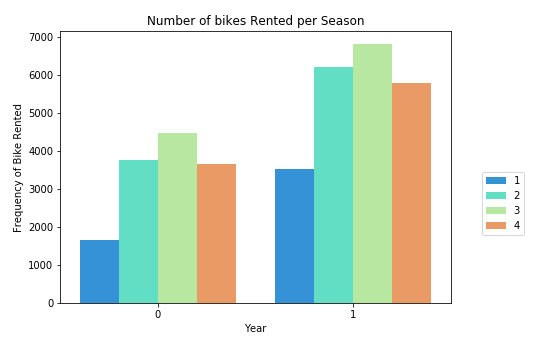
**Python Code:**

#Analysis shows more bike was rented on year 2011

g = sns.FacetGrid(df, hue='yr', palette='rainbow',size=4,aspect=2)

g = g.map(plt.hist,'cnt',alpha=0.8, edgecolor='r')

plt.legend()

****

**Figure 2.4** : Number of bike Rented per year(2010-2011) on each Season

**Python Code:**

plt.figure(figsize=(7,5))

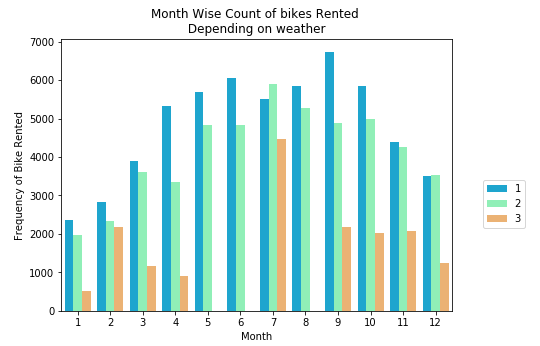
sns.barplot('yr','cnt',hue='season', data=df,palette='rainbow', ci=None)

plt.legend(loc='upper right',bbox\_to\_anchor=(1.2,0.5))

plt.xlabel('Year')

plt.ylabel('Frequency of Bike Rented')

plt.title('Number of bikes Rented per Season')

****

**Figure 2.5:** Number of bike Rented per month on each weather situation

**Python Code:**

plt.figure(figsize=(7,5))

sns.barplot('mnth','cnt',hue='weathersit', data=df,palette='rainbow', ci=None)

plt.legend(loc='upper right',bbox\_to\_anchor=(1.2,0.5))

plt.xlabel('Month')

plt.ylabel('Frequency of Bike Rented')

plt.title('Month Wise Count of bikes Rented \n Depending on weather ')

From the above histogram fig: 2.3 it is clear that with the increase in time demands for bike in rent are also increasing. Therefore more number of counts is observed in the year 2011.In the fig 2.4 it can be observed that demands for bike was more on summer and fall. From fig 2.5 clear weather is more favourable for bike demands between June to September month.

**Conclusion based on above analysis:**

* Demand of bike for rent is increasing with the increase in time.
* Bikes are rented more when the weather is clear.
* Bikes are rented more between June to September.
* Counts of bike for rent increases with the increase in temperature.
* Casual user rented bike more on weekends.
* Registered user rented bike more on weekdays
* Count of bike for rent is more on the working days and less on holidays

Therefore with the help of above visualization it clear that the number of count for bike rent depends mainly on temperature, weather condition and weathers the day was working or holiday. Trend of user demands for bike is also different, as registered user demands bike on the weekdays while casual user demands bike more on Saturday and Sunday. In the next chapter we will move forward for data pre-processing.

**Chapter 3**

**Data Pre-processing**

**3.1 Data Pre Processing:**

Data is growing day-by-day. Extracting knowledge from it is a real threat otherwise it is nothing but garbage. Data, when loaded into the database from various source, emerge as a messy dataset. These datasets are of no use because extracting valuable information from it is very tough. Therefore data pre-processing is the first and mandatory step for any data scientist before mining [2]. According to Lour, data scientists spend 50% - 80% of their valuable time and effort in data collection and preparing disorderly digital data, before it can be explored for useful nuggets.

**3.1.1 Missing Value Analysis:**

In the first step of data pre-processing we look into the dataset to collect missing values in different variables. Missing value should be recognized and replace (mean, median or knn imputation) or drop it because in the presence of missing values, a machine learning algorithm doesn’t perform well. Bike renting dataset does not contain any missing values therefore we dropped further analysis related to it.

|  |  |
| --- | --- |
| Instant | 0 |
| Dteday | 0 |
| Season | 0 |
| Yr | 0 |
| Mnth | 0 |
| Holiday | 0 |
| weekday | 0 |
| workingday | 0 |
| weathersit | 0 |
| Temp | 0 |
| Atemp | 0 |
| Hum | 0 |
| windspeed | 0 |
| Casual | 0 |
| registered | 0 |
| Cnt | 0 |

**Table 3.1**: Missing value analysis detail

**3.1.2 Outlier Detection**

Outliers are extreme values in an observation of dataset that falls beyond the range from other records that are more or less similar to each other [1]. Outliers can occur in a dataset due to some experiment error variation in measurement etc. Outliers in a dataset should be taken care of before application of machine learning algorithm to the dataset in order to get an appropriate result because machine learning algorithm is sensitive towards the distribution of the dataset. A massive amount of outliers in a dataset can cause poor and inaccurate models with longer training times by misguiding machine learning algorithms. In a dataset, outliers can be part of records during the collection, processing or analyzing records. Machine learning algorithm like linear and logistic regression is very much affected in its training process with the presence of outliers. Human error, instruments error, experimental errors, data processing errors, etc. are some of the causes of outliers. Outliers can be broadly classified into two types' univariate and multivariate.

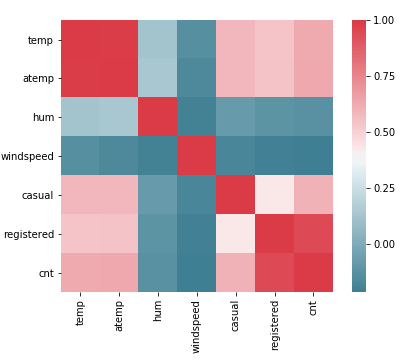
Univariate outliers occur when extreme values are searched on one feature of the dataset. Box plots fall under the category of a univariate method. It is one of the simplest and popular techniques for the detection of outliers. Box plot describes a feature of the dataset, by using statistics calculation such as lower quartiles, upper quartiles, and median. Box plot uses the format of the box for specification of data distribution. Box plot, known as Tukey's method, introduced in 1977 is a very important visualization tool for the detection of outliers by displaying univariate feature as lower quartile, upper quartile, lower extreme, upper extreme and median of the records present in the dataset. IQR is an interquartile range, which is a distance between Q1 and Q3 quartiles. Inner fences are the place at 1.5 IQR below Q1 and above Q3 distance. Outer fences are at the place of 3 IQR below Q1 and above Q3. Features detected between the inner and outer fences are the possible outliers and values beyond outer and inner fences can be identified as extreme outliers. Here we are using box plot tukey method to analyze outliers in the continuous variables. In fig 3.1 it is noticeable that only casual, wind speed and humid is consist of outliers which is further replaced by mean value of that particular variable.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  | |

**Figure 3.1**: Box Plot of different continuous variables for detecting outliers

**3.1.3 Feature selection**

It is not realistic to perform a complex data analytics on a massive dataset. Data reduction is a technique to achieve a smaller dataset. Prior to any machine learning modelling we should measure the significance of each independent variable of our dataset. It can happen that many variables are not performing important role during model building. These types of variable should be recognized and dropped before any predictive analysis. Let us look in our dataset for insignificant independent variable. We have constructed heat map to measured the co-relation of numeric behaviour.



**Figure 3.2**: Heat map plot

**Python Code:**

#Set the width and hieght of the plot

f, ax = plt.subplots(figsize=(7, 5))

#Generate correlation matrix

corr = df\_corr.corr()

#Plot using seaborn library

sns.heatmap(corr, mask=np.zeros\_like(corr, dtype=np.bool),

cmap=sns.diverging\_palette(220, 10, as\_cmap=True),

square=True, ax=ax)

From the fig. 3.2 it is clearly seen that temp and atemp has a strong co-relation, which is not acceptable in case of dependant variables. Therefore, one variable among them should be removed as they pass redundant information to the model. It is observed that dependant variable cnt is equals to the sum of the casual + registered columns. Therefore this should better to remove to avoid overfitting. Another reason to avoid those columns is that these will not be available for the future data. date column is well-represented by the other date-related columns season, yr, mnth, and weekday. We will discard it. row index column instant is a useless column to us. Therefore in this step of feature selection we will remove those columns.

**df = df.drop(['instant', 'dteday', 'atemp','casual','registered'], axis=1)**

As a result we are left with final 11 columns for modelling which has shown below. In the next chapter we will build a model for predicting bike rent count with the help of ml algorithms.

'season', 'yr', 'mnth', 'holiday', 'weekday', 'workingday', 'weathersit', 'temp', 'hum', 'windspeed', 'cnt'

**Chapter 4**

**Data Modeling**

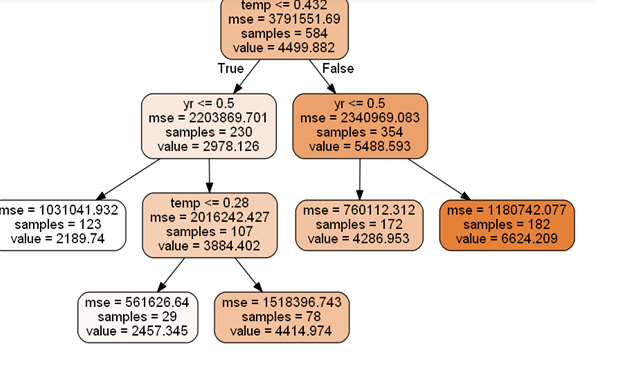
**4.1.1 Model Selection**

From the previous step of analysis it is clear that cnt which is our dependant variable is a continuous variable and therefore this is a regression problem. To solve regression problem we have to select machine learning regression algorithm. To predict the bike rent count here we are using decision tree and multiple linear regression and decision tree. Let us first try decision tree on our dataset and check the result.

**4.1 Decision Tree**

By using the default values of decision tree doesn’t provide satisfactory result and therefore we tried pruning trees. Our decision tree was built based on following criteria.

DecisionTreeRegressor (criterion='mse', max\_depth=5, max\_features=None, max\_leaf\_nodes =5, min\_impurity\_decrease=0.0, min\_impurity\_split=None, min\_samples\_leaf=10, min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0, presort=False, random\_state=None, splitter='best')

****

**Fig 4.1:** Decision Tree Regression Plot for Bike Renting

**4.1.1 Model Evaluation**

After building a model we have to evaluate it to check the performance of a model. There are numerous error matrixes for the evaluation of regression model like MAE, MAPE and MSE. We are using MAPE here because it is easy to understand as it gives a percentage of error during prediction.

**MAPE:**

**Error : 22.5**

**Accuracy : 77.5**

**4.1.2 Multiple Linear Regression**

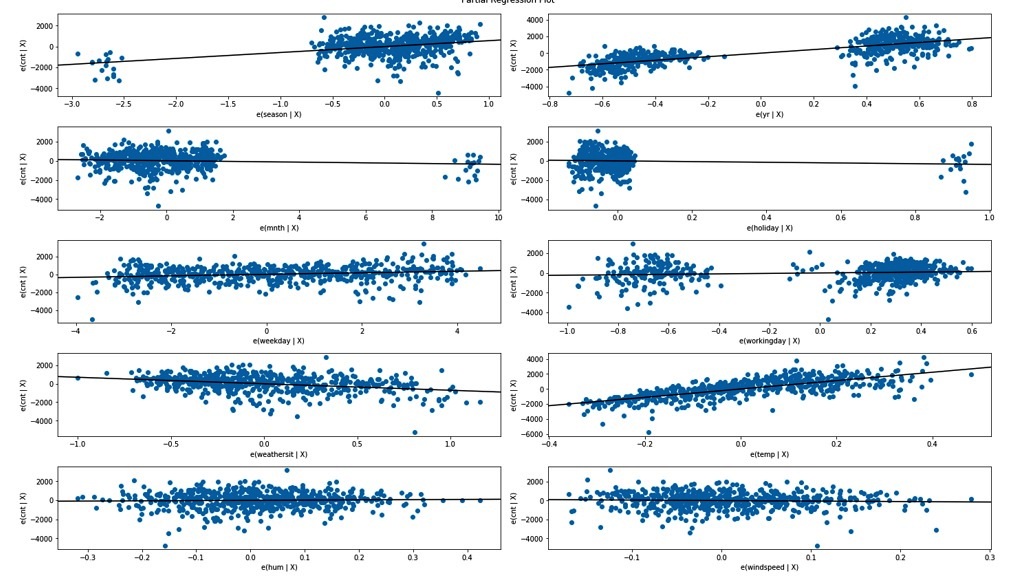
|  |  |  |  |
| --- | --- | --- | --- |
| OLS Regression Results | | | |
| **Dep. Variable:** | cnt | **R-squared:** | 0.967 |
| **Model:** | OLS | **Adj. R-squared:** | 0.966 |
| **Method:** | Least Squares | **F-statistic:** | 1681. |
| **Date:** | Sun, 28 Apr 2019 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 16:19:05 | **Log-Likelihood:** | -4793.7 |
| **No. Observations:** | 584 | **AIC:** | 9607. |
| **Df Residuals:** | 574 | **BIC:** | 9651. |
| **Df Model:** | 10 |  |  |
| **Covariance Type:** | nonrobust |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Coef** | **std err** | **t** | **P>|t|** | **[0.025** | **0.975]** |
| **season** | 568.2739 | 63.089 | 9.007 | 0.000 | 444.360 | 692.188 |
| **Yr** | 2147.7342 | 73.167 | 29.354 | 0.000 | 2004.027 | 2291.442 |
| **Mnth** | -37.0161 | 19.807 | -1.869 | 0.062 | -75.919 | 1.886 |
| **holiday** | -365.4268 | 234.102 | -1.561 | 0.119 | -825.227 | 94.374 |
| **weekday** | 85.5326 | 18.088 | 4.729 | 0.000 | 50.006 | 121.059 |
| **workingday** | 229.5165 | 80.513 | 2.851 | 0.005 | 71.380 | 387.653 |
| **weathersit** | -705.6275 | 91.100 | -7.746 | 0.000 | -884.557 | -526.698 |
| **Temp** | 5563.8419 | 218.197 | 25.499 | 0.000 | 5135.281 | 5992.403 |
| **Hum** | 249.7116 | 298.335 | 0.837 | 0.403 | -336.251 | 835.674 |
| **windspeed** | -517.1135 | 451.031 | -1.147 | 0.252 | -1402.986 | 368.759 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Omnibus:** | 78.902 | **Durbin-Watson:** | 1.850 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 174.467 |
| **Skew:** | -0.743 | **Prob(JB):** | 1.30e-38 |
| **Kurtosis:** | 5.227 | **Cond. No.** | 103. |

*Adjusted R-squared* value, we can 96% of the data can be explaines using our multiple

linear regression model. This is impressive, but there can also be a chance of overfitting but looking at the *F-statistic* and combined p-value we can reject the null hypothesis that target variable does not depend on any of the predictor variables. Looking at the significance values of some of the predictor we can see that there is some insignificant variable let us remove them and check the result.



**Fig 4.2: Partial Regression Plot**

**4.1.2.1 Model Evaluation**

Mean absolute percentage error**(**MAPE**) is used here for model evaluation.**

**Error – 21%**

**Accuracy-79%**

*Although* **Adjusted** *R-squared* value is 96% error rate is high therefore we can remove insignificant variable in order to tune the model. OLS regression result after removing insignificant variable.

|  |  |  |  |
| --- | --- | --- | --- |
| OLS Regression Results | | | |
| **Dep. Variable:** | cnt | **R-squared:** | 0.964 |
| **Model:** | OLS | **Adj. R-squared:** | 0.964 |
| **Method:** | Least Squares | **F-statistic:** | 2214. |
| **Date:** | Sun, 28 Apr 2019 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 16:45:40 | **Log-Likelihood:** | -4819.9 |
| **No. Observations:** | 584 | **AIC:** | 9654. |
| **Df Residuals:** | 577 | **BIC:** | 9684. |
| **Df Model:** | 7 |  |  |
| **Covariance Type:** | nonrobust |  |  |
|  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **coef** | **std err** | **t** | **P>|t|** | **[0.025** | **0.975]** |
| **season** | 576.9837 | 65.692 | 8.783 | 0.000 | 447.960 | 706.007 |
| **yr** | 2173.0764 | 75.454 | 28.800 | 0.000 | 2024.878 | 2321.275 |
| **mnth** | -42.5215 | 20.532 | -2.071 | 0.039 | -82.849 | -2.194 |
| **weekday** | 83.2736 | 18.303 | 4.550 | 0.000 | 47.324 | 119.223 |
| **workingday** | 249.0526 | 80.701 | 3.086 | 0.002 | 90.549 | 407.556 |
| **weathersit** | -636.1377 | 58.553 | -10.864 | 0.000 | -751.140 | -521.136 |
| **temp** | 5461.1050 | 200.374 | 27.255 | 0.000 | 5067.553 | 5854.657 |

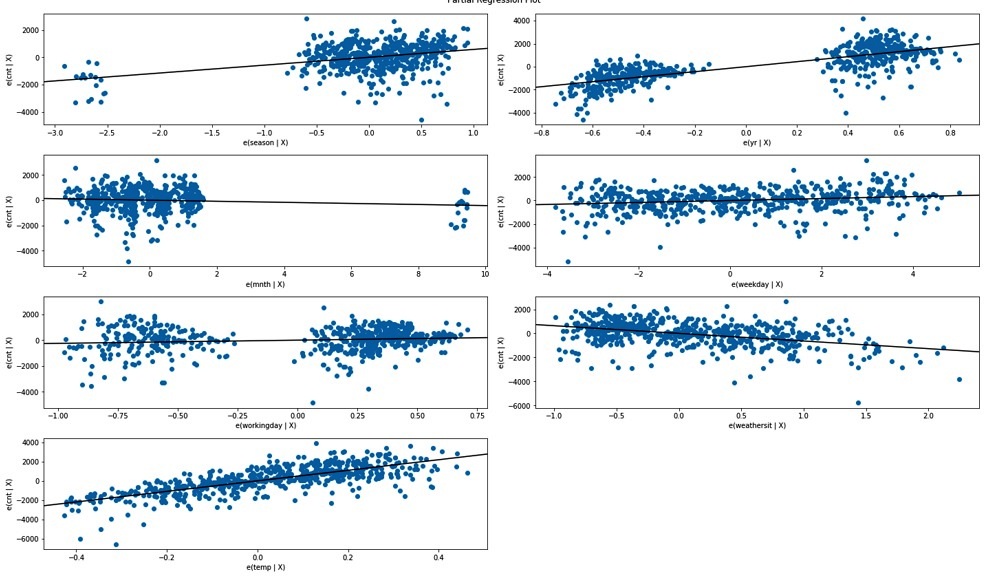
|  |  |  |  |
| --- | --- | --- | --- |
| **Omnibus:** | 91.517 | **Durbin-Watson:** | 1.916 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 214.230 |
| **Skew:** | -0.830 | **Prob(JB):** | 3.02e-47 |
| **Kurtosis:** | 5.459 | **Cond. No.** | 44.3 |

Mean absolute percentage error**(**MAPE**) is used here for model evaluation.**

**As we can seen that Adjusted** *R-squared* value is 96% which is same as the previous model but there is huge different in the error and accuracy percentage

**Error – 17%**

**Accuracy – 83%**



**Fig 4.3:** Partial Regression Plot

**Conclusion**

All the three model seems to be good and selecting a model is again very tedious job. Our first model (using decision tree) is showing 77% accuracy whereas our third model is showing 83% accuracy with a very high*R-squared* value which is 96%. Obviously our third model (using Multiple linear regression) seems to be very powerful with high accuracy , high R-square and less error percentage with a slight doubt of overfitting. To best of my knowledge I have seen all the prospectives, I have compare predicted values with that of actual values still the r-square value is same. Therefore by rejecting the null hypothesis that target variable does not depend on any of the predictor variables. I can conclude that independant variables can very strongly explain the dependant variable and can freeze our last model as a best among all three.

**References:**

1. Jingke Xi, Outlier detection algorithm in data mining, 2008 Second International Symposium on Intelligent Information Technology Application, December 2008.
2. Prity Vijay, Bright Keswani,ICCS 2018, Enhanced Approach to Attain Competent Big Data Pre-Processing