Bike-Sharing Assignment based Question and Answers

Assignment-based Subjective Questions

Question 1. From your analysis of the categorical variables from the dataset, what could you infer about their effect on the dependent variable?

Answer :

Following are the major categorical variables along with their respective impact on the rented bikes count :

- **season :** One-third (more than 30%) of the bookings were done in fall season, followed by Summer and Winter. This can serve as a **good predictor** for our analysis.

- **holiday :** Very low number of bookings during a holiday. We have a very high bias here and thus it **may not be a good predictor** variable.

- **yr :** During yr 1, we have more number of bookings (almost twice) than yr 0, indicating a clear year on year rise in the rental count. Since our analysis will be based only on two years (2018 and 2019), they are **important predictors**.

- **mnth :** Distribution of rental counts across months show a clear pattern and a crucial factor here could also be the underlying season & weather conditions. This indeed can serve as a **good predictor**.

  For e.g : *observation shows the during fall season (September-October), when humidity is relatively high, there is an optimum increase in the rental count.*

- **weekday :** No clear pattern or impact observed here as all the days show a similar spread. This **is not a good predictor**

- **workingday :** There isn't a major impact on rental count based on different workingday categories. This is **not a good predictor.**

- **weathersit :** Majority of the bookings took place during a clear/partly cloudy weather with a median at around 5000 followed by misty/cloudy weather, with median 4000. Rental count is least during light snow/rain/storm like weather conditions and exactly nil for extreme weather (Heavy Rain/Ice Pallets/Thunderstorm/ Snow). Hence weathersit can serve as a **good predictor**.

. **Question 2**. Why is it important to use **drop_first = True** during dummy variable creation?

Answer :

While working with large number of categorical variables, it becomes complex to handle and classify each of them for the machine to understand. Hence, with an assumption that all the categorical variables are mutually exclusive (do not occur at the same time), we use the technique of Dummy variables, wherein each variable is assigned a binary value (0 and 1).

The logic behind it : When a variable has "1", it is the applicable value while all the other variables are expected to have "0" (since mutually exclusive).

For e.g : Assuming a categorical variable named **status** has 4 values : Pending, In progress, Completed and Stuck/Blocked.

Here, if we want to create dummy variable for this, then number of dummies = 4 :

Pending → 0 or 1
In progress → 0 or 1
Completed → 0 or 1
Stuck/Blocked →0 or 1

Clearly, as each row will have one of the above status values, all the dummies can't be simultaneously 0 or 1. Also, exactly one of the above dummies can be 1 for a given row.

Hence, assuming Pending, In progress and Completed as 0, the only possibility is the status is Stuck/Blocked for a given person. Same shall apply to each of the other three dummies.

Since three dummies here suffice to indicate the fourth (when combined), we must drop the fourth dummy to prevent multicollinearity.

So, our final dataset will have 3 dummies for Status variable : Pending, In progress and Completed

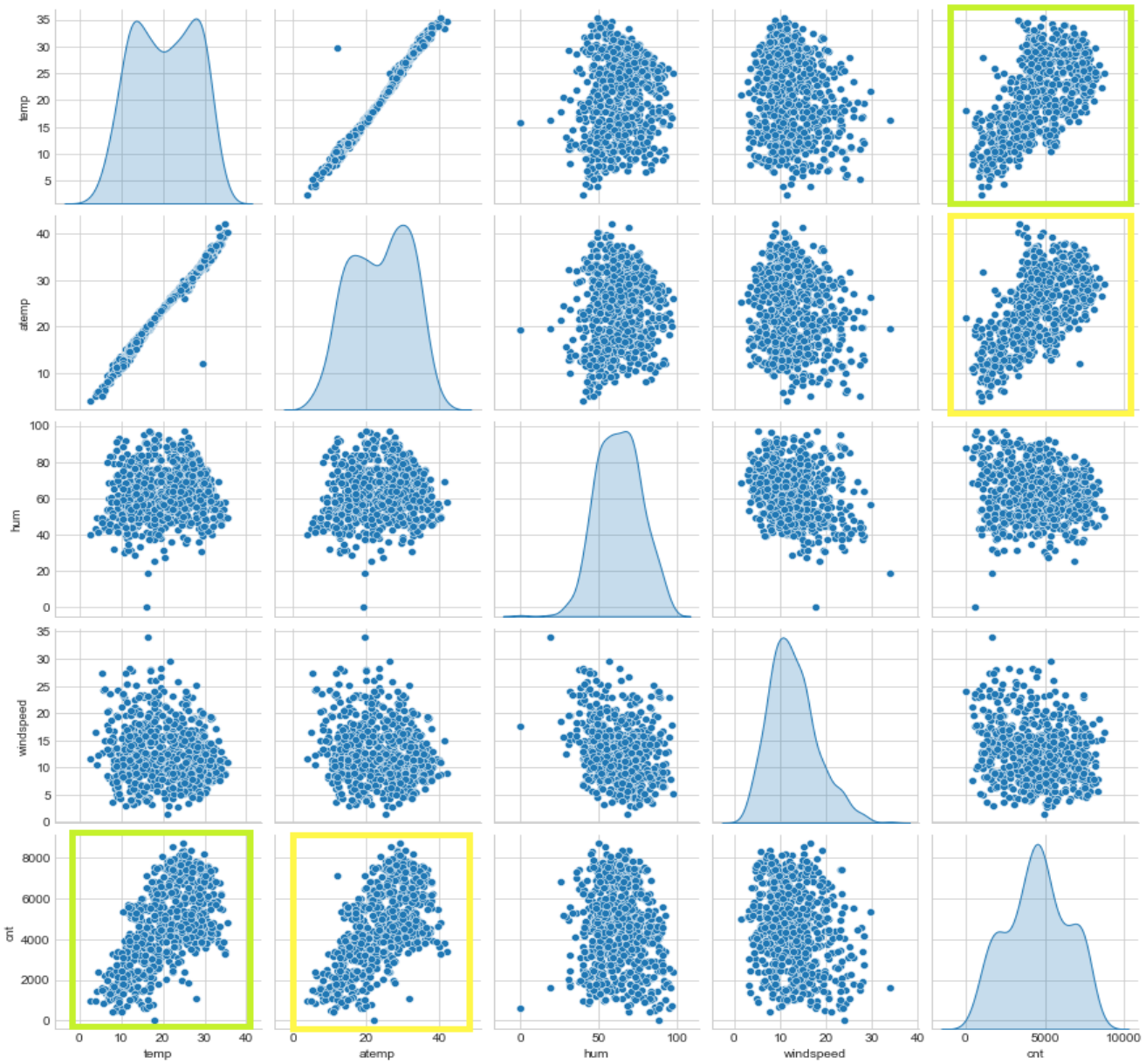
As an inference we can say for each given categorical variable in a dataset having n different categories, the number of required dummies is n-1. In order to achieve this we make use of → drip_first=True argument that comes along with pd.get_dummies function offered by Pandas.

Question 3. Looking at the pair-plot among the numerical variables, which one has the highest correlation with the target variable?

Answer :

Following variable(s) have the highest correlation with the target variable (pair-plot attached for reference) :

atemp has the highest correlation followed by temp (both are highly positively correlated with each other as well as target variable cnt)



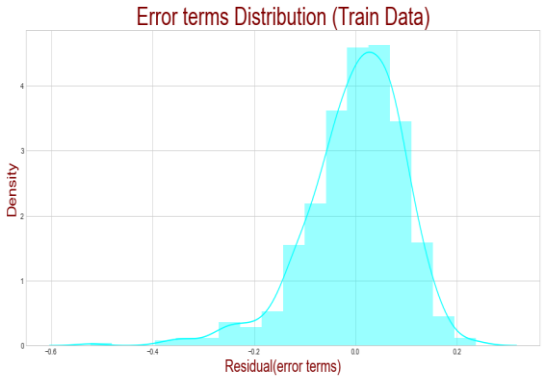
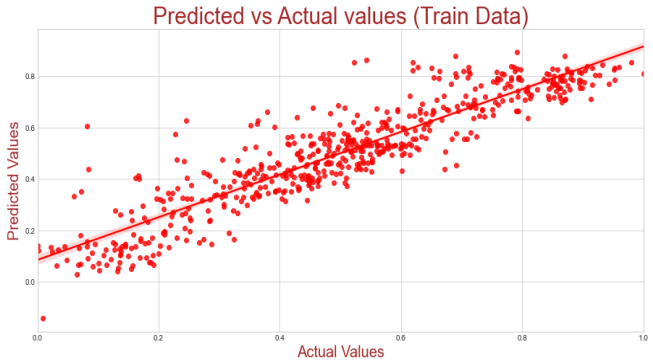
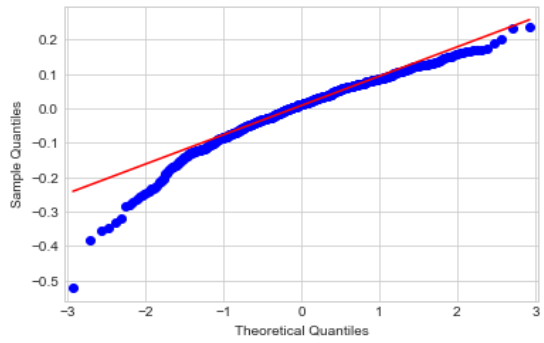
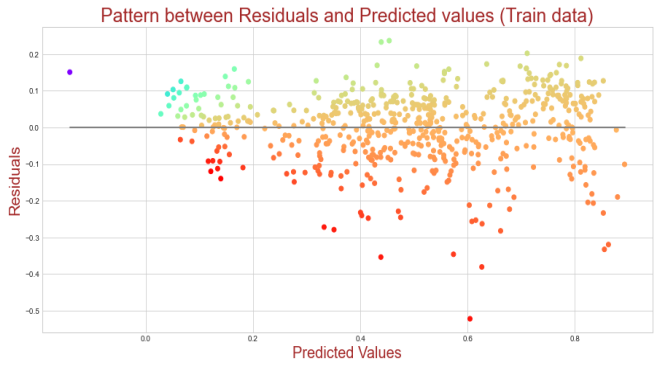
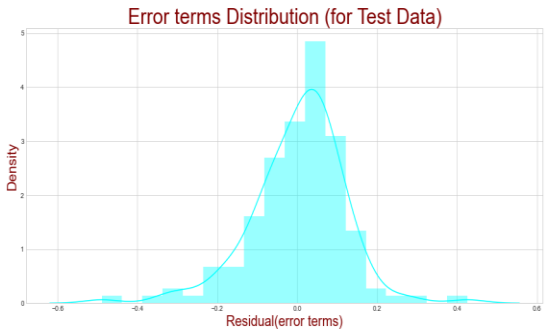
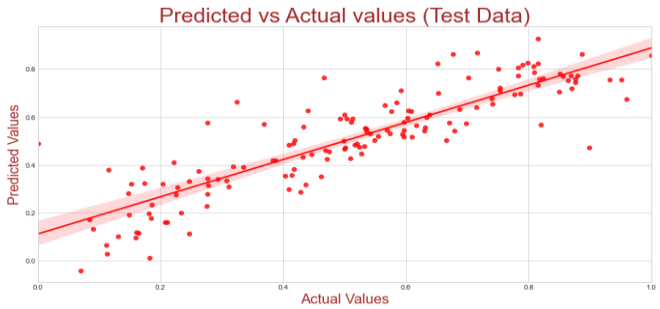
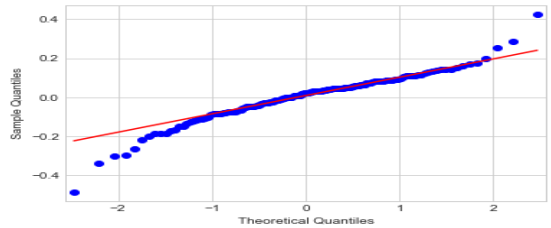
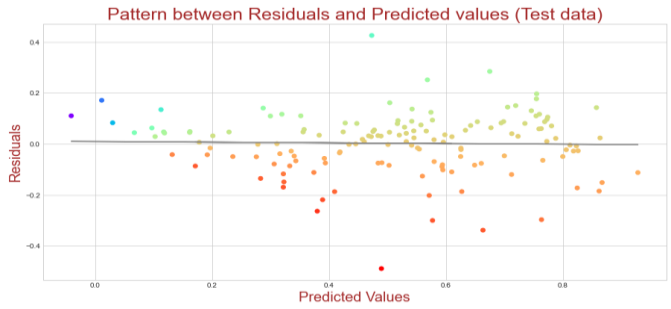
Question 4. How did you validate the assumptions of Linear Regression after building the model on the training set?

Answer : Following are the assumptions for a linear regression model :

1. **Normality :** Error terms are normally distributed (around mean = 0) i.e, for any fixed value of X, Y is normally distributed.
    - Verified using a distplot and QQ-plot *(refer the table in next page)*
2. **Linearity:** The relationship between X and the mean of Y is linear.
    - Verified using a pairplot (reflecting relation between numerical variables and target variable "cnt")
    - Temp and atemp show a positive linear relationship with target variable cnt
3. **No or little multicollinearity :** Observations are independent and do not impact each other (no or least correlation) . There are various various to measure multicollinearity and we have used VIF *(Variation Inflation Factor)* to measure the same :
    - VIF below 5 is acceptable while that below 2 is considered favorable.
    - Final model has **VIF below 2.25 for each** variable *(as shown in the table)*
    - Values having higher VIF values have already been dropped as part of recursive model building.
4. **Homoscedasticity :** The variance of residual is the same for any value of X. (please refer the table in next slide)
5. **No pattern or auto-correlation :** Error terms(residuals) are independent of predicted value and aren't auto-correlated. (please refer the below table)

	Predictors	VIF_Value
0	atemp	2.21
1	yr	1.91
2	weathersit_Mist/Cloudy	1.46
3	season_spring	1.23
4	mnth_9	1.15
5	mnth_10	1.13
6	weathersit_Light-Snow/Rain/Storm	1.04

- **Durbin-Watson statistic value** from our final model : 1.8 → implies slight Autocorrelation (2 represents no autocorrelation)

DATASET	NORMALITY	HOMOSCEDASTICITY & INDEPENDENCE
Train		
		
Test		
		

Question 5. Based on the final model, which are the top 3 features contributing significantly towards explaining the demand of the shared bikes?

Answer :

The equation for the final model obtained is as follows :

cnt = 0.2312 + (atemp x 0.3958) + (yr x 0.2543) - (season_spring x 0.1645) + (mnth_9 x 0.0914) + (mnth_10 x 0.0810) - (weathersit_Light-Snow/Rain/Storm x 0.2732) - (weathersit_Mist/Cloudy x 0.0820)

From the coefficients (sorted order), the top three significant ones are :

1. **Atemp : coeff value is** 0.3958
2. **Yr : coeff value is** 0.2543
3. **Wethersit_Light-Snow/Rain/Storm : coeff value is** -0.2732

General Subjective Questions

Question 1 : Explain the linear regression algorithm in detail.

Answer : Linear Regression is a type of supervised learning algorithm. It is used to achieve a finite numerical value of y (dependent variable) by identifying coefficients of one or more predictor variables x (independent variables), given that x and y follow a linear relationship.

It learns the dependence of y from x from the past data. The equation of a simple one degree **Linear equation is of the following form :**

$$y = mx + c$$

where,

m : slope (can either be positive or negative)

c : intercept (where the value of x becomes zero)

Regression mainly refers to **backwards tracking or learning from past data**. The line that gets drawn using the linear regression model is called the **Regression Line**. Th regression model must find the **best fit line** (one with least total vertical distance from every individual data point) that **best explains variance in the dataset**.

It is represented as : $y_{true} - y_{line}$, which are called as residuals. However, due to data points present on both sides of the regression line, we can't get a better indication of the best fit line (positive and negative datapoints are likely to cancel out each other). Hence , we take the sum of the squared values of the residuals. This is termed as RSS (Residual Sum of Squares)

We thus define the **Best Fit Linear Regression** Line as the line which **best explains the variance of y(dependent) against variance of x(independent).** It is used to predict y for a given set of x values having **least RSS value**.

Question 2. Explain the Anscombe's quartet in detail.

Answer : Anscombe's quarter shows that multiple data sets with similar statistical properties can still vary vastly when graphed. This makes a regression model fit all the datasets, however, once plotted they show a different picture. This is mainly why it is used as a caution measure when learning linear regression. It also warns about the danger of outliers in dataset.

Defined as : *A set of four data sets which appear similar in statistical/descriptive analysis, but have very different distribution when plotted as scatter plots*

Let us take a sample dataset and understand it in more details …

Dataset source : https://www.geeksforgeeks.org/anscombes-quartet/

Code used :

For reading the dataset and printings stats (including Pearson's R) for each x and y :

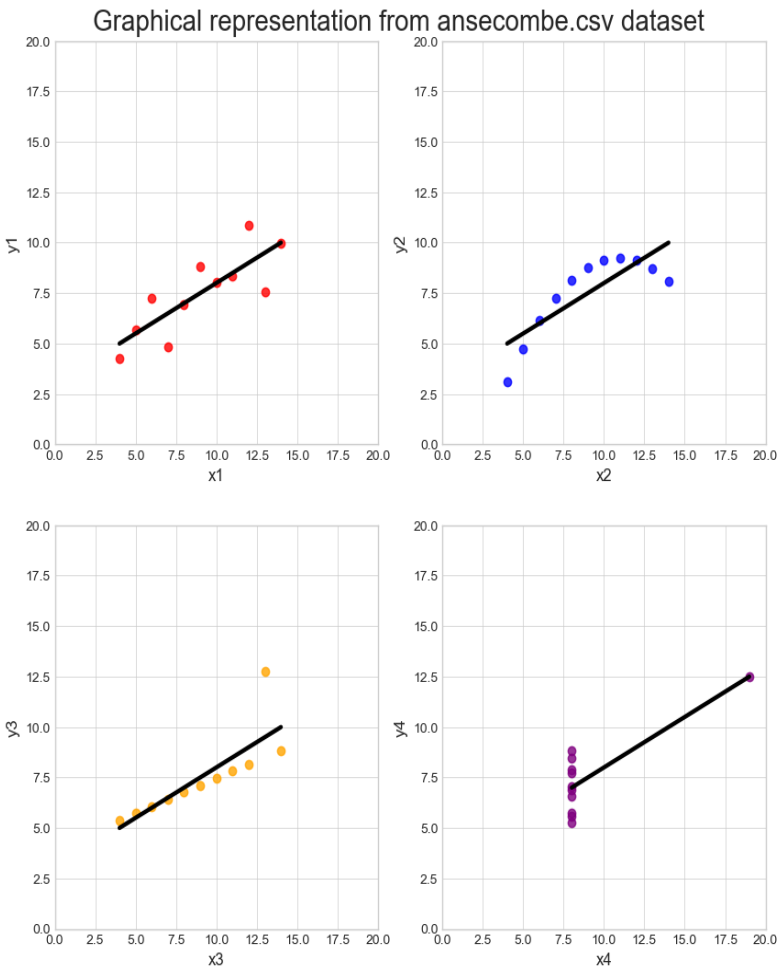
```
import pandas as pd
from scipy.stats import pearsonr
anscombe = pd.read_csv('anscombe.csv')

stats =  anscombe.describe().transpose().drop([ 'count','min', '25%', '50%', '75%', 'max'], axis=1)
x_stats = stats.iloc[:4, :].reset_index().drop('index', axis=1 ).rename(columns = {'mean' : 'xmean', 'std' : 'xstd'})
y_stats = stats.iloc[:4, :].reset_index().drop('index', axis=1 ).rename(columns = {'mean' : 'ymean', 'std' : 'ystd'})

stats = pd.concat([x_stats, y_stats], axis =1 , ignore_index=False)

pearson_r = []
for i in range (4, anscombe.shape[1]):
    pearson_r.append(pearsonr(anscombe.iloc[:, i-4], anscombe.iloc[: ,i])[0])

stats['corr(x,y)'] = pearson_r

#printing the stats
stats
```

	xmean	xstd	ymean	ystd	corr(x,y)
0	9.0	3.316625	9.0	3.316625	0.816421
1	9.0	3.316625	9.0	3.316625	0.816237
2	9.0	3.316625	9.0	3.316625	0.816287
3	9.0	3.316625	9.0	3.316625	0.816521

For Plots :

```
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

plt.style.use('seaborn-whitegrid')

fig,((ax1, ax2),(ax3, ax4)) = plt.subplots(2,2, figsize = (10,10))
plt.subplots_adjust(top = 0.95)
plt.suptitle('Graphical representation from ansecombe.csv dataset', size=20)

p1 = sns.regplot('x1', 'y1', data = anscombe, ax = ax1, ci = None, color ="red", line_kws = dict(color = 'black'))
p2 = sns.regplot('x2', 'y2', data = anscombe, ax = ax2, ci = None, color ="blue", line_kws = dict(color = 'black'))
p3 = sns.regplot('x3', 'y3', data = anscombe, ax = ax3, ci = None, color ="orange", line_kws = dict(color = 'black'))
p4 = sns.regplot('x4', 'y4', data = anscombe, ax = ax4, ci = None, color ="purple", line_kws = dict(color = 'black'))

p1.set_xlim(0,20)
p2.set_xlim(0,20)
p3.set_xlim(0,20)
p4.set_xlim(0,20)

p1.set_ylim(0,20)
p2.set_ylim(0,20)
p3.set_ylim(0,20)
p4.set_ylim(0,20)

plt.show()
```

Plots :



It is evident from these 4 scatter plots that they completely vary in distribution from each other, however, the regression line is the exact same for all.

- In fig [0][0], the regression line is indeed the best fit line among all.
- In fig [0][1], the data itself is non linear and hence shows a polynomial curve of order 2 or more. Here the regression line is thus over simplifying.
- In fig [1][0] and [1][1], there is a presence of outlier which makes the regression line identical with the above two. However, outliers are not considered usually for the general trend. They are thus incorrect and misleading.

Question 3 : What is Pearson's R?

Answer :

Correlation is the measure of how well the variables in a dataset or between sets of data are related. The most common measure in stats is the Pearson Correlation. It's full name is Pearson Product Moment Correlation (PPMC).

Person's R (also known as correlation coefficient) is the measure that reflects the linear relationship between two variables.

It can be in the range : -1 to 1

0 : no linear relationship between x and y (0 correlation)
1 : y is a positive linear function of x (strong positive correlation)
-1 : y is a negative linear function of x (strong negative correlation)

Let us consider 2 variables with same number of data points n and each point is defined by i where i is an integer from 1 to n.

Covariance of x and y is the sum of product of each value of x from its mean and each value of y from its mean.

Question 4. What is scaling? Why is scaling performed? What is the difference between normalized scaling and standardized scaling?

Answer :

Let us assume we have n variables as : x1, x2, x3, ... xn and y is another variable that is a function of x1 to xn. If we now want to create a model that finds coefficients for each of the variables in order to get y for different values of x, the coefficients for each of these x will be a reflection of :

1. Magnitude of x
2. Impact of x

The impact of x because it is a function of y and magnitude of x, as a unit change in x (considering it is numerical) will hardly have any impact on y, keeping it's coefficient low.

Also, the best fit linear equation is obtained by minimizing the cost function using the Gradient Descent algorithm and it goes over several iterations at a specified step (learning rate). It needs to kept low in order to avoid any overshooting.

When the variables are of different scales, the ones with a large range, gradient descent takes longer and still may not be able to give the optimal cost function. Hence, it is crucial to scale the numerical values to a smaller range for accurate and quicker gradient descent operation.

There are 2 types of scalers :

1. Minmax scalar or normalizer
2. Standard scalar or standardizer

Minmax scalar (normalizer) :

Here the minimum value of a variable becomes 0 while maximum becomes 1 and every other value is kept between 0 and 1.

Formula used : $(x - x_{min})/x_{max} - x_{min}$

Standard scalar (standardizer) :

Here, the mean is calculated first followed by each value, which is represented as the difference from it's respective mean divided by it's standard deviation.

Formula used : $(x - x_{mean})/sd(x)$

It usually puts the values between -1 and 1, but if there are outliers, it can go upto -3 to 3 (depending on outlier's range : 3 std deviations away upto 99% confidence interval).

In such cases, the Min Max scalar is likely to suffer due to outliers as it keeps the min and max values fixed irrespective of outliers.

Also, Min max scalar does not change the picture of a variable, it simply shrinks the range in order to make it easier to understand the variable scale, whereas Standard scalar has each point's difference from the mean. Here the entire original picture can only be seen after removing the scaling (i.e, transforming it back to origin scale).

Question 5. You might have observed that sometimes the value of VIF is infinite. Why does this happen?

Answer :

Vif (Variance inflation factor : measure of amount of multicollinearity) for a predictor variable is given by equation :

$$\frac{1}{1 - R^2}$$

where**, R^2 is the value of Explained variance/Total variance** , for that variable on other predictor variables.

There can be cases where R^2 can be 1, in case the predictor has a high linear relationship with another variable. In such a case, **VIF becomes infinity from the above formula**, (with denominator → 0).

This indicates that a particular variable has extreme linear relationship with another predictor variable.

Question 6. What is a Q-Q plot? Explain the use and importance of a Q-Q plot in linear regression.

Answer :

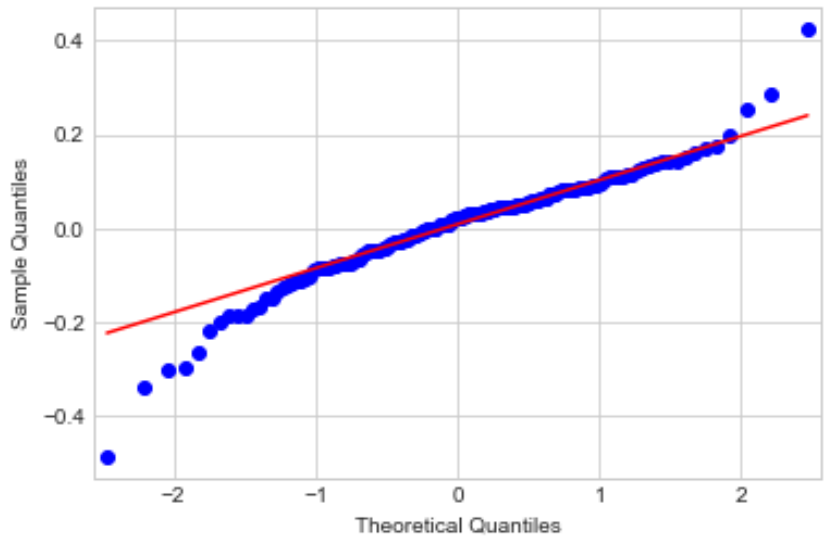
Q-Q plot (Quantile to Quantile plot) is used to compare the quantiles or percentiles between two given set of data points. It is used in linear regression to compare the residuals against the normal distribution.

In Q-Q plot, one plot if a normal curve while the other is the residual curve (difference between actual and predicted y values from the model).

In order to compare, the quantile from one curve needs to be drawn as a linear line (usually normal curve), and then the residuals are plotted (as scattered plots)

If normality is true, we must see that the plots are well line up on the top of the line (masking it)

If normality isn't true (presence of skewness), then the plotted lines are likely to shift away from the normality line.



The red line shows up normal distribution by default (it is used by applying argument 'line' set to 'q')

The **qqplot** function is offered by **statsmodel.api** which needs to be imported separately in order to use qqplot().