Linear Regression Question and Answer

Assignment-based Subjective

Q1. From your analysis of the categorical variables from the dataset, what could you infer about their effect on the dependent variable?
Answer:

- The demand of bike is less in the month of spring when compared with other seasons.
- The demand bike increased in the year 2019 when compared with year 2018.
- Month Jun to Sep is the period when bike demand is high. The Month Jan is the lowest demand month.
- Bike demand is less in holidays in comparison to not being holiday.
- The demand of bike is almost similar throughout the weekdays.
- There is no significant change in bike demand with working day and non-working day.
- The bike demand is high when weather is clear and Few clouds however demand is less in case of Light-snow and light rainfall. We do not have any data for Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog , so we cannot derive any conclusion. May be the company is not operating on those days or there is no demand of bike.

Q2. Why is it important to use drop_first=True during dummy variable creation?
Answer:
It **helps in reducing the extra column created during dummy variable creation**. Hence it reduces the correlations created among dummy variables.
Eg: Let's say we have 3 types of values in Categorical column, and we want to create dummy variable for that column. If one variable **is not furnished and semi_furnished, then it is obvious it is unfurnished**. So we do not need 3rd variable to identify the unfurnished.
Hence if we have categorical variable with **n-levels**, then we need to **use n-1** columns to represent the dummy variables.

Value	Indicator Variable	
Furnishing Status	furnished	semi-furnished
furnished	1	0
semi-furnished	0	1
unfurnished	0	0

Q3. Looking at the pair-plot among the numerical variables, which one has the highest correlation with the target variable?

Answer:

Temp has highest positive correlation with target variable cnt.

Q4. How did you validate the assumptions of Linear Regression after building the model on the
training set?

Answer:

The **intercept and coefficient of our model can be calculated as shown below:**

```
#Calculate intercept and coefficient
print(model.intercept_)
print(model.coef_)pred=model.predict(X_test)
predictions = pred.reshape(-1,1)#Calculate root mean squared error to evaluate model
performance
from sklearn.metrics import mean_squared_error
print('MSE : ', mean_squared_error(y_test,predictions)
print('RMSE : ', np.sqrt(mean_squared_error(y_test,predictions)))
```

The performance of the model can be evaluated by finding the **root mean squared error of the model**. **Lesser the RMSE, better the model.**

Q5. Based on the final model, which are the top 3 features contributing significantly towards explaining the demand of the shared bikes?

Answer:

The Top 3 features contributing significantly towards the demands of share bikes are:

1. weathersit_Light_Snow(negative correlation).
2. yr_2019(Positive correlation).
3. temp(Positive correlation).

General Subjective Questions

Q1. Explain the linear regression algorithm in detail.

Answer:

Linear Regression Algorithm is a machine learning algorithm based on supervised learning. Linear regression is a part of regression analysis. Regression analysis is a technique of predictive modeling that helps you to find out the relationship between Input and the target variable.

Regression analysis is used for three types of applications:

1. Finding out the effect of Input variables on Target variable.
2. Finding out the change in Target variable with respect to one or more input variable.
3. To find out upcoming trends.

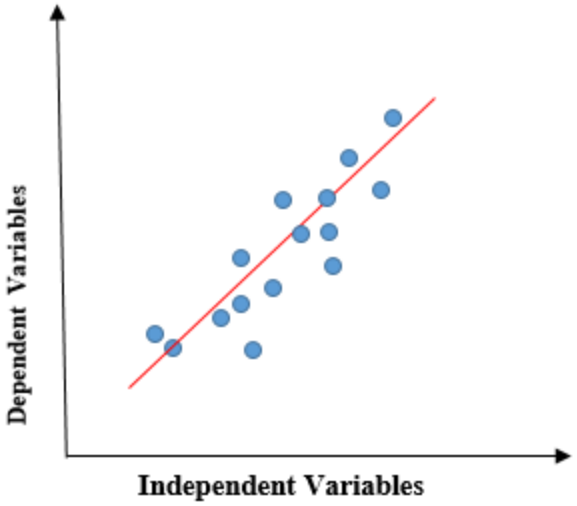
Here are the types of regressions:

Linear Regression: In Regression, we plot a graph between the variables which best fit the given data points. The machine learning model can deliver predictions regarding the data. In naïve words, ***"Regression shows a line or curve that passes through all the data points on a target-predictor graph in such a way that the vertical distance between the data points and the regression line is minimum."*** It is used principally for prediction, forecasting, time series modeling, and determining the causal-effect relationship between variables.

Types of Regression models

1. Linear Regression
2. Multiple Linear Regression
3. Logistic Regression
4. Polynomial Regression

Linear regression is a quiet and simple statistical regression method used for predictive analysis and shows the relationship between the continuous variables. Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis), consequently called linear regression*. If there is a single input variable (x), such linear regression is called* ***simple linear regression****. And if there is more than one input variable, such linear regression is called* ***multiple linear regression*****.** The linear regression model gives a sloped straight line describing the relationship within the variables.



The above graph presents the linear relationship between the dependent variable and independent variables. When the value of x (**independent variable**) increases, the value of y (**dependent variable**) is likewise increasing. The red line is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best.
To calculate best-fit line linear regression uses a traditional slope-intercept form.

$$y = mx+b \quad \Longrightarrow \quad y = a_0+a_1x$$

y= Dependent Variable.
x= Independent Variable.
a0= intercept of the line.
a1 = Linear regression coefficient.

Need of a Linear regression

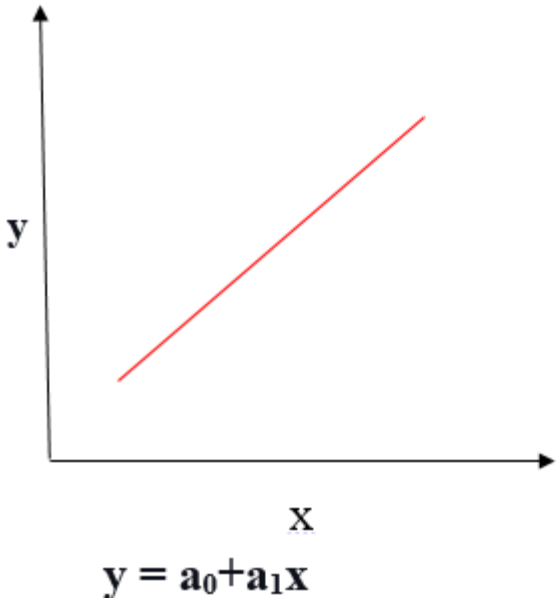
As mentioned above, Linear regression estimates the relationship between a dependent variable and an independent variable. Let's understand this with an easy example:

Let's say we want to estimate the salary of an employee based on year of experience. You have the recent company data, which indicates that the relationship between experience and salary. Here year of experience is an independent variable, and the salary of an employee is a dependent variable, as the salary of an employee is dependent on the experience of an employee. Using this insight, we can predict the future salary of the employee based on current & past information.

A regression line can be a Positive Linear Relationship or a Negative Linear Relationship.

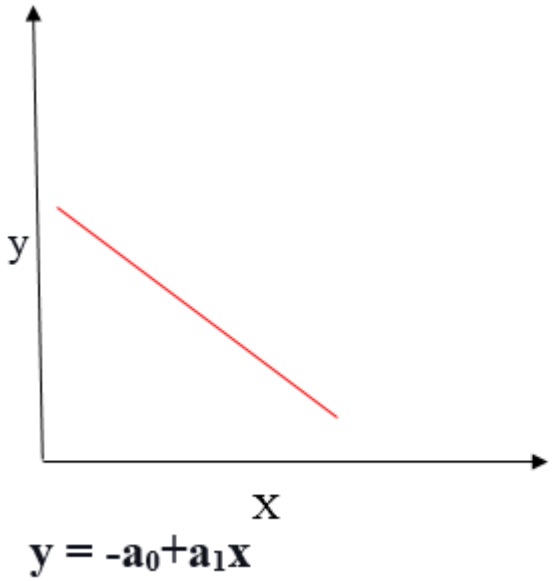
Positive Linear Relationship

If the dependent variable expands on the Y-axis and the independent variable progress on X-axis, then such a relationship is termed a Positive linear relationship.



Negative Linear Relationship

If the dependent variable decreases on the Y-axis and the independent variable increases on the X-axis, such a relationship is called a negative linear relationship.



The goal of the linear regression algorithm is to get the best values for a0 and a1 to find the best fit line. The best fit line should have the least error means the error between predicted values and actual values should be minimized.

Cost function

The cost function helps to figure out the best possible values for a0 and a1, which provides the best fit line for the data points.

Cost function optimizes the regression coefficients or weights and measures how a linear regression model is performing. The cost function is used to find the accuracy of the **mapping**

function that maps the input variable to the output variable. This mapping function is also known as **the Hypothesis function**.

In Linear Regression, **Mean Squared Error (MSE)** cost function is used, which is the average of squared error that occurred between the predicted values and actual values.

By simple linear equation y=mx+b we can calculate MSE as:

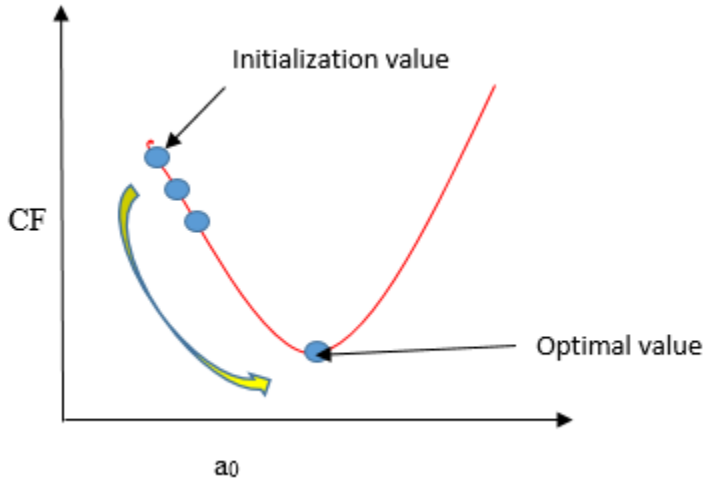
Let's y = actual values, y_i = predicted values

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(y_i - (mx_i + b))^2$$

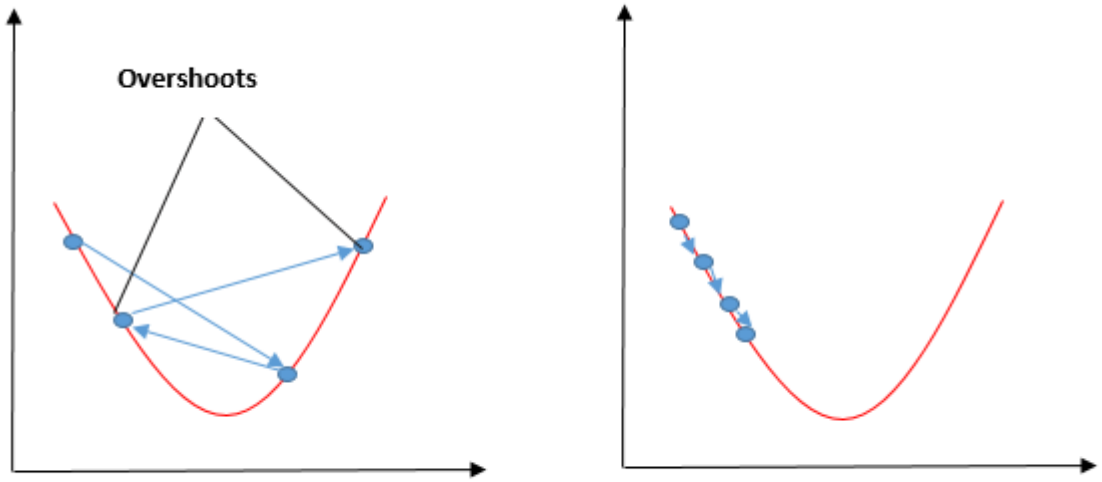
Using the MSE function, we will change the values of a0 and a1 such that the MSE value settles at the minima. Model parameters ***xi, b (a_0,a_1)*** can be manipulated to minimize the cost function. These parameters can be determined using the gradient descent method so that the cost function value is minimum.

Gradient descent

Gradient descent is a method of updating a0 and a1 to minimize the cost function (MSE). A regression model uses gradient descent to update the coefficients of the line (a0, a1 => xi, b) by reducing the cost function by a random selection of coefficient values and then iteratively update the values to reach the minimum cost function.



Imagine a pit in the shape of U. You are standing at the topmost point in the pit, and your objective is to reach the bottom of the pit. There is a treasure, and you can only take a discrete number of steps to reach the bottom. If you decide to take one footstep at a time, you will eventually get to the bottom of the pit but, this will take a longer time. If you choose to take longer steps each time, you may get to sooner but, there is a chance that you could overshoot the bottom of the pit and not near the bottom. In the gradient descent algorithm, the number of steps you take is the learning rate, and this decides how fast the algorithm converges to the minima.



High learning rate

Low learning rate

To update a_0 and a_1, we take gradients from the cost function. To find these gradients, we take partial derivatives for a_0 and a_1.

$$J = \frac{1}{n}\sum_{i=1}^{n}(a_0 + a_1 \cdot x_i - y_i)^2$$

$$\frac{\partial J}{\partial a_0} = \frac{2}{n}\sum_{i=1}^{n}(a_0 + a_1 \cdot x_i - y_i)$$

$$\frac{\partial J}{\partial a_1} = \frac{2}{n}\sum_{i=1}^{n}(a_0 + a_1 \cdot x_i - y_i) \cdot x_i$$

$$\frac{\partial J}{\partial a_0} = \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i)$$

$$\frac{\partial J}{\partial a_1} = \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i) \cdot x_i$$

$$a_0 = a_0 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i)$$

$$a_1 = a_1 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i) \cdot x_i$$

The partial derivates are the gradients, and they are used to update the values of a_0 and a_1. Alpha is the learning rate.

Use case

In this, I will take random numbers for the dependent variable (salary) and an independent variable (experience) and will predict the impact of a year of experience on salary.

Steps to implement Linear regression model

1. **import some required libraries**

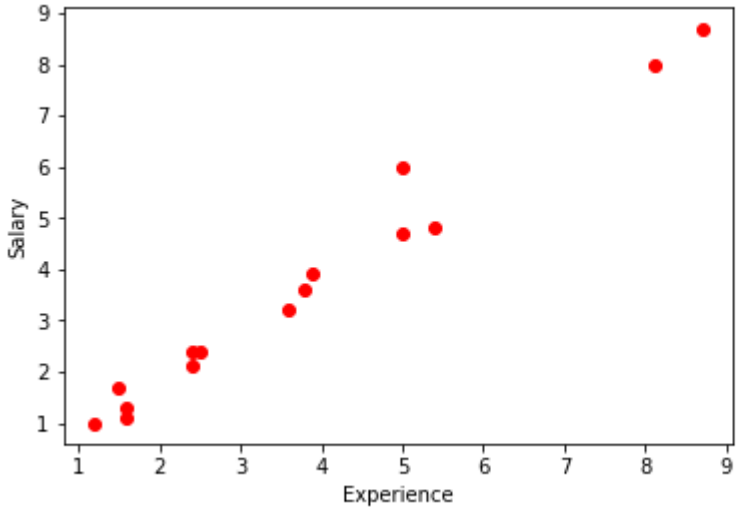
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

2. **Define the dataset**

```
x= np.array([2.4,5.0,1.5,3.8,8.7,3.6,1.2,8.1,2.5,5,1.6,1.6,2.4,3.9,5.4])
y = np.array([2.1,4.7,1.7,3.6,8.7,3.2,1.0,8.0,2.4,6,1.1,1.3,2.4,3.9,4.8])
n = np.size(x)
```

3. **Plot the data points**

```
plt.scatter(experience,salary, color = 'red')
plt.xlabel("Experience")
plt.ylabel("Salary")
plt.show()
```



The main function to calculate values of coefficients

1. Initialize the parameters.
2. Predict the value of a dependent variable by given an independent variable.
3. Calculate the error in prediction for all data points.
4. Calculate partial derivative w.r.t a0 and a1.
5. Calculate the cost for each number and add them.
6. Update the values of a0 and a1.

```
#initialize the parameters
a0 = 0              #intercept
a1 = 0              #Slop
lr = 0.0001          #Learning rate
iterations = 1000      # Number of iterations
error = []           # Error array to calculate cost for each iterations.

for itr in range(iterations):
   error_cost = 0
```

```
  cost_a0 = 0
  cost_a1 = 0
  for i in range(len(experience)):
     y_pred = a0+a1*experience[i]   # predict value for given x
     error_cost = error_cost +(salary[i]-y_pred)**2
     for j in range(len(experience)):
      partial_wrt_a0 = -2 *(salary[j] - (a0 + a1*experience[j]))           #partial derivative w.r.t a0
       partial_wrt_a1 = (-2*experience[j])*(salary[j]-(a0 + a1*experience[j]))   #partial derivative
w.r.t a1
        cost_a0 = cost_a0 + partial_wrt_a0     #calculate cost for each number and add
        cost_a1 = cost_a1 + partial_wrt_a1     #calculate cost for each number and add
     a0 = a0 - lr * cost_a0    #update a0
     a1 = a1 - lr * cost_a1    #update a1
     print(itr,a0,a1)        #Check iteration and updated a0 and a1
  error.append(error_cost)     #Append the data in array
```

```
51 -0.2100587036075669 1.0240594725158565
51 -0.210069416639929 1.0240604165874214
51 -0.2100827665464727 1.0240617279107738
51 -0.21009872814788516 1.024063481908892
51 -0.2101172734497008 1.0240657579772252
51 -0.21013837262662904 1.0240686345668573
51 -0.21016199500166557 1.0240721843016172
51 -0.21018810996167744 1.0240764694231033
51 -0.21021668775506228 1.0240815378378745
51 -0.21024770012426341 1.02408742000484
51 -0.21028112073594665 1.0240941268503343
51 -0.21031692538390484 1.0241016488365344
51 -0.21035509195351762 1.0241099562394875
52 -0.21035743358141284 1.024110693217287
52 -0.21036211816964787 1.0241121510338222
52 -0.2103691481960975 1.0241142983610119
52 -0.2103785269213727 1.024117090524692
52 -0.21039025787243382 1.024120472129565
52 -0.2104043442038269 1.0241243803082456
52 -0.21042078797530397 1.024128748398941
```

At approximate iteration 50- 60, we got the value of a0 and a1.

```
print(a0)
print(a1)
```

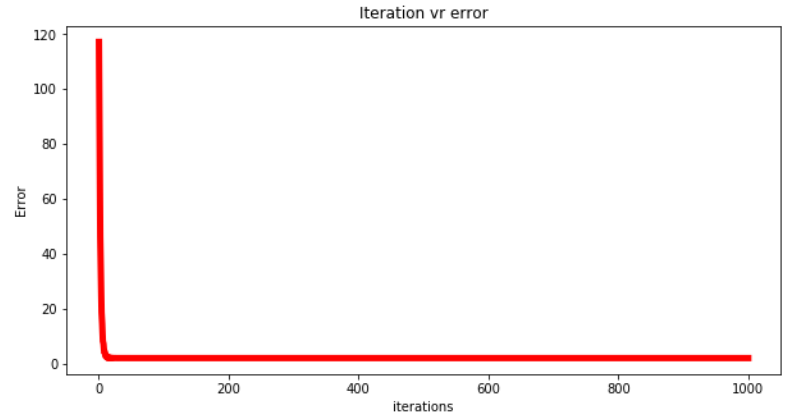
```
-0.21354150071690242
1.0247464287610857
```

4. **Plotting the error for each iteration.**

```
plt.figure(figsize=(10,5))
plt.plot(np.arange(1,len(error)+1),error,color='red',linewidth = 5)
plt.title("Iteration vr error")
```

```
plt.xlabel("iterations")
plt.ylabel("Error")
```

```
Text(0, 0.5, 'Error')
```



Predicting the values.

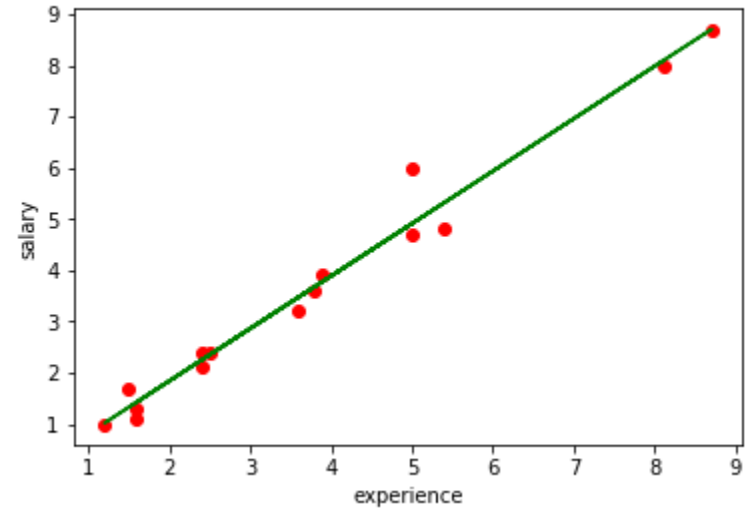
```
pred = a0+a1*experience
print(pred)
```

```
[2.24584993 4.91019064 1.32357814 3.68049493 8.70175243 3.47554564
 1.01615421 8.08690457 2.34832457 4.91019064 1.42605279 1.42605279
 2.24584993 3.78296957 5.32008921]
```

5. Plot the regression line.

```
plt.scatter(experience,salary,color = 'red')
plt.plot(experience,pred, color = 'green')
plt.xlabel("experience")
plt.ylabel("salary")
```

```
Text(0, 0.5, 'salary')
```



6. **Analyze the performance of the model by calculating the mean squared error.**

```
error1 = salary - pred
se = np.sum(error1 ** 2)
mse = se/n
print("mean squared error is", mse)
```

```
mean squared error is 0.12785817711928918
```

7. Use the scikit library to confirm the above steps.

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
experience = experience.reshape(-1,1)
model = LinearRegression()
model.fit(experience,salary)
salary_pred = model.predict(experience)
Mse = mean_squared_error(salary, salary_pred)
print('slop', model.coef_)
print("Intercept", model.intercept_)
print("MSE", Mse)
```

```
slop [1.02474643]
Intercept -0.2135415007169037
MSE 0.1278581771192891
```

Summary

In Regression, we plot a graph between the variables which best fit the given data points. Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis).To calculate best-fit line linear regression uses a traditional slope-intercept form. A regression line can be a Positive Linear Relationship or a Negative Linear Relationship.

The goal of the linear regression algorithm is to get the best values for a0 and a1 to find the best fit line and the best fit line should have the least error. In Linear Regression, **Mean Squared Error (MSE)** cost function is used, which helps to figure out the best possible values for a0 and a1, which provides the best fit line for the data points. Using the MSE function, we will change the values of a0 and a1 such that the MSE value settles at the minima. Gradient descent is a method of updating a0 and a1 to minimize the cost function (MSE)

Q2. Explain the Anscombe's quartet in detail.

Answer:

Anscombe's quartet comprises four datasets that have nearly identical simple statistical properties yet appear very different when graphed. Each dataset consists of eleven (x,y) points. They were constructed in 1973 by the statistician Francis Anscombe to demonstrate both the importance of graphing data before analyzing it and the effect of outliers on statistical properties.

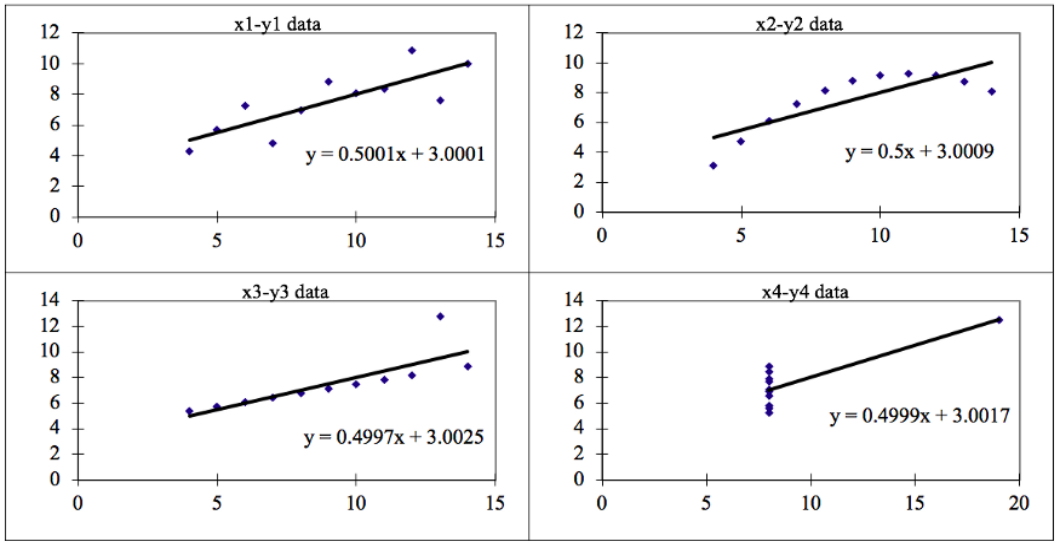
Simple understanding:

Once Francis John "Frank" Anscombe who was a statistician of great repute found 4 sets of 11 data-points in his dream and requested the council as his last wish to plot those points. Those 4 sets of 11 data-points are given below.

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

After that, the council analyzed them using only descriptive statistics and found the mean, standard deviation, and correlation between x and y.

When the models are plotted:



The four datasets can be described as:

1. **Dataset 1:** this **fits** the linear regression model pretty well.
2. **Dataset 2:** this **could not fit** linear regression model on the data quite well as the data is non-linear.
3. **Dataset 3:** shows the **outliers** involved in the dataset which **cannot be handled** by linear regression model
4. **Dataset 4:** shows the **outliers** involved in the dataset which **cannot be handled** by linear regression model

Conclusion:

We have described the four datasets that were intentionally created to describe the importance of data visualization and how any regression algorithm can be fooled by the same. Hence, all the important features in the dataset must be visualized before implementing any machine learning algorithm on them which will help to make a good fit model.

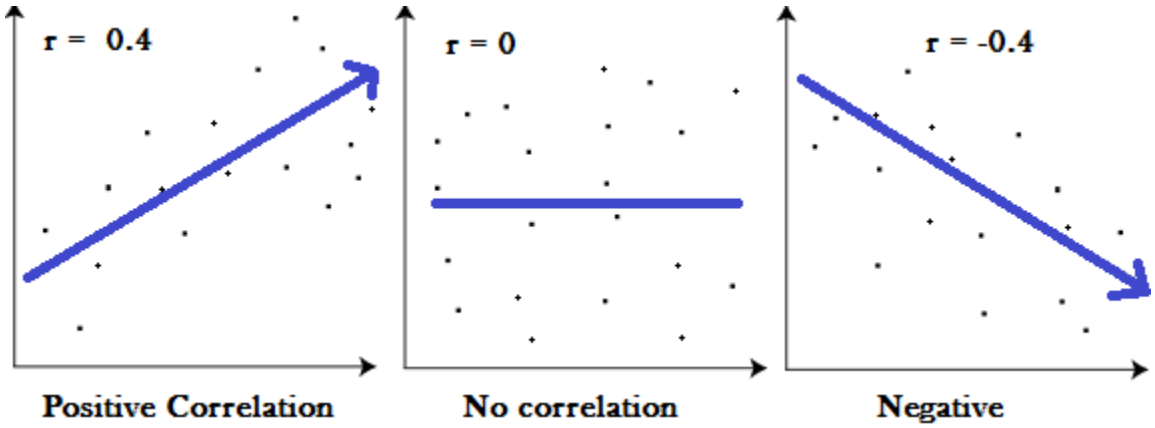
Q 3. What is Pearson's R?

Answer:

Correlation Coefficient Formula: Definition

Correlation coefficient formulas are used to find how strong a relationship is between data. The formulas return a value between -1 and 1, where:

a. 1 indicates a strong positive relationship.
b. -1 indicates a strong negative relationship.
c. A result of zero indicates no relationship at all.



Correlation between sets of data is a measure of how well they are related. The most common measure of correlation in stats is the Pearson Correlation. The full name is the **Pearson Product Moment Correlation (PPMC) or bivariate correlation**. It shows the linear relationship between two sets of data. Two letters are used to represent the Pearson correlation: Greek letter rho (ρ) for a population and the letter "r" for a sample.

Using the formula proposed by Karl Pearson, we can calculate a linear relationship between the two given variables. For example, a child's height increases with his increasing age (different factors affect this biological change). So, we can calculate the relationship between these two variables by obtaining the value of Pearson's Correlation Coefficient r. There are certain requirements for Pearson's Correlation Coefficient:

Scale of measurement should be interval or ratio. Variables should be approximately normally distributed. The association should be linear. There should be no outliers in the data

The formula given is:

$$r = \frac{N\Sigma xy-(\Sigma x)(\Sigma y)}{\sqrt{[N\Sigma x^2-(\Sigma x)^2][N\Sigma y^2-(\Sigma y)^2]}}$$

Where,
N = the number of pairs of scores
Σxy = the sum of the products of paired scores
Σx = the sum of x scores
Σy = the sum of y scores
Σx2 = the sum of squared x scores
Σy2 = the sum of squared y scores

Some steps are needed to be followed:

Step 1: Make a Pearson correlation coefficient table. Make a data chart using the two variables and name them as X and Y. Add three additional columns for the values of XY, X^2, and Y^2. Refer to this table.

Person	Age (X)	Income (Y)	XY	X^2	Y^2
1					
2					
3					
4					

Step 2: Use basic multiplications to complete the table.

Person	Age (X)	Income (Y)	XY	X^2	Y^2
1	20	1500	30000	400	2250000
2	30	3000	90000	900	9000000
3	40	5000	200000	1600	25000000
4	50	7500	375000	2500	56250000

Step 3: Add up all the columns from bottom to top.

Person	Age (X)	Income (Y)	XY	X^2	Y^2
1	20	1500	30000	400	2250000
2	30	3000	90000	900	9000000
3	40	5000	200000	1600	25000000

4	**50**	**7500**	**375000**	**2500**	**56250000**
Total	**140**	**17000**	695000	**5400**	**92500000**

Step 4: Use these values in the formula to obtain the value of r.

r = [4 * 695000 - 140 * 17000] / √{4 * 5400 - (140)^2} {4 * 92500000 - (17000)^2}
= [2780000 - 2380000] / √{21600 - 19600} {370000000 - 289000000}
= 400000 / √{2000} {81000000}
= 400000 / √162000000000
= 400000 / 402492.24
= 0.99

The positive value of Pearson's correlation coefficient implies that if we change either of these variables, there will be a **positive effect** on the other. For example, if we increase the age there will be an increase in the income.

Value of correlation coefficients lie between -1 and +1. The magnitude tells us the strength of the relationship while the sign suggests the direction.

Q4. What is scaling? Why is scaling performed? What is the difference between normalized scaling and standardized scaling?
Answer:
What is scaling:

It is a step of data Pre-Processing which is applied to independent variables to normalize the data within a particular range. It also helps in speeding up the calculations in an algorithm.

Feature scaling in machine learning is one of the most critical steps during the pre-processing of data before creating a machine learning model. Scaling can make a difference between a weak machine learning model and a better one.

The most common techniques of feature scaling are Normalization and Standardization.

Why Scaling Performed:

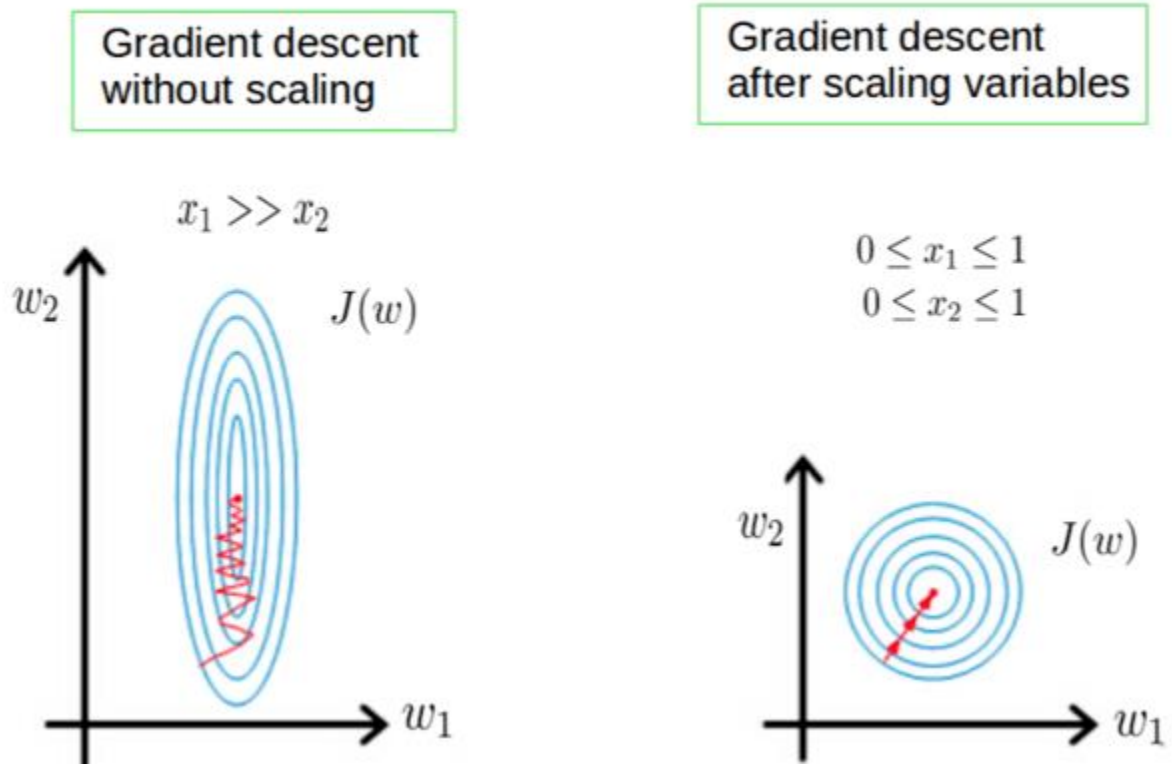
Most of the times, collected data set contains features highly varying in magnitudes, units, and range. If scaling is not done, then algorithm only takes magnitude in account and not units hence incorrect modelling. To solve this issue, we have to do scaling to bring all the variables to the same level of magnitude.

It is important to note that scaling just affects the coefficients and none of the other parameters like t-statistic, F-statistic, p-values, R-squared, etc.

Eg. Suppose we have two features of weight and price, as in the below table. The “Weight” cannot have a meaningful comparison with the “Price.” So, the assumption algorithm makes that since “Weight” > “Price,” thus “Weight,” is more important than “Price.”

Name	Weight	Price
Orange	15	1
Apple	18	3
Banana	12	2
Grape	10	5

So, these more significant number starts playing a more decisive role while training the model. Thus, feature scaling is needed to bring every feature in the same footing without any upfront importance. Interestingly, if we convert the weight to “Kg,” then “Price” becomes dominant. Another reason why feature scaling is applied is that few algorithms like Neural network gradient descent **converge much faster** with feature scaling than without it.



One more reason is **saturation**, like in the case of sigmoid activation in Neural Network, scaling would help not to saturate too fast.

Difference between normalized scaling and standardized scaling

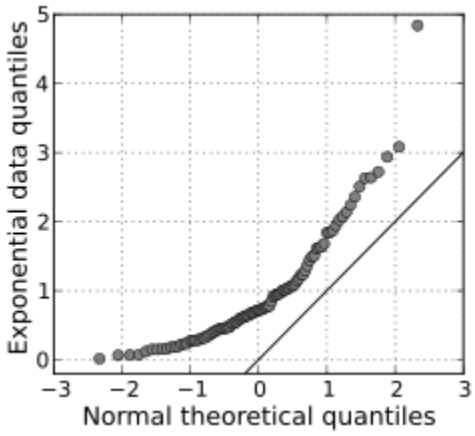
Srno	Advantages	Disadvantages
1	Minimum and maximum value of features are used for scaling	Mean and standard deviation is used for scaling.
2	It is used when features are of different scales	It is used when we want to ensure zero mean and unit standard deviation.
3	Scales values between [0, 1] or [-1, 1].	It is not bounded to a certain range.
4	It is really affected by outliers.	It is much less affected by outliers.
5	It is useful when we don't know about the distribution	It is useful when the feature distribution is Normal or Gaussian.
6	It is a often called as Scaling Normalization	It is a often called as Z-Score Normalization.

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

If there is perfect correlation, then VIF = infinity. This shows a **perfect correlation between two independent variables**. In the case of perfect correlation, **we get R2 =1**, which **lead to 1/(1-R2) infinity**. To solve this problem, we need to drop one of the variables from the dataset which is causing this perfect multicollinearity.
An infinite VIF value indicates that the corresponding variable may be expressed exactly by a linear combination of other variables (which show an infinite VIF as well).

Q6. What is a Q-Q plot? Explain the use and importance of a Q-Q plot in linear regression.
Answer:

Q-Q Plots (Quantile-Quantile plots) are plots of two quantiles against each other. A quantile is a fraction where certain values fall below that quantile. For example, the median is a quantile where 50% of the data fall below that point and 50% lie above it. The purpose of Q Q plots is to find out if two sets of data come from the same distribution.
A 45-degree angle is plotted on the Q Q plot; if the two data sets come from a common distribution, the points will fall on that reference line.
A Q Q plot showing the 45-degree reference line:



If the two distributions being compared are similar, the points in the Q–Q plot will approximately lie on the line y = x. If the distributions are linearly related, the points in the Q–Q plot will approximately lie on a line, but not necessarily on the line y = x. Q–Q plots can also be used as a graphical means of estimating parameters in a location-scale family of distributions.
A Q–Q plot is used to compare the shapes of distributions, providing a graphical view of how properties such as location, scale, and skewness are similar or different in the two distributions.

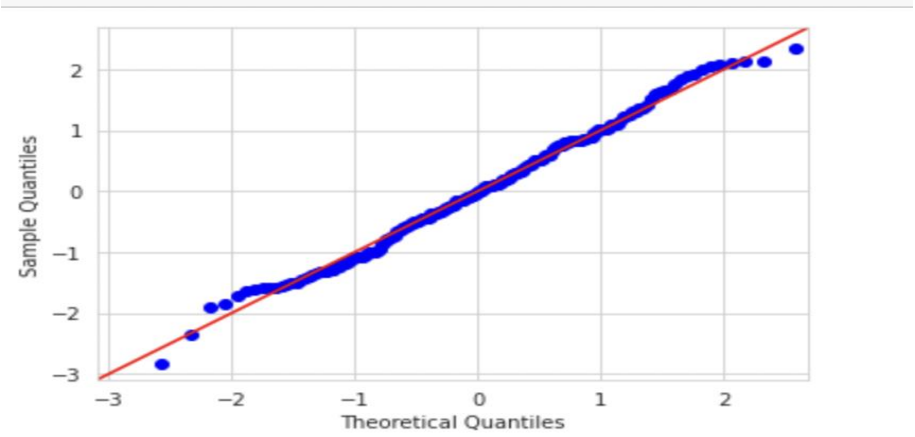
Uses and Importance of QQ plot in Linear Regression:

Help us assess if a set of data possibly came from some theoretical distribution such as a Normal, exponential or Uniform distribution.
Also, it helps to determine if two data sets come from populations with a common distribution.

This helps in a scenario of linear regression when we have training and test data set received separately and then we can confirm using Q-Q plot that both the data sets are from populations with same distributions.

In Q-Q plots, we plot the theoretical Quantile values with the sample Quantile values. Quantiles are obtained by sorting the data. It determines how many values in a distribution are above or below a certain limit.

If the datasets we are comparing are of the same type of distribution type, we would get a roughly straight line. Here is an example of normal distribution.



Here is an example, where we are generating data x from a Gamma distribution with shape = 2 and rate = 1 parameter.

```
# Set seed for reproducibility
set.seed(2017);
# Generate some Gamma distributed data
x <- rgamma(100, shape = 2, rate = 1);
# Sort x values
x <- sort(x);
# Theoretical distribution
x0 <- qgamma(ppoints(length(x)), shape = 2, rate = 1);
plot(x = x0, y = x, xlab = "Theoretical quantiles", ylab = "Observed quantiles");
abline(a = 0, b = 1, col = "red");
```

Above code will output like this: