1. What is Regression?

Regression in statistics is the process of predicting a Label(or Dependent Variable) based on the features(Independent Variables) at hand.

Regression is used for time series modelling and finding the causal effect relationship between the variables and forecasting. For example, the relationship between the stock prices of the company and various factors like customer reputation and company annual performance etc. can be studied using regression.

Regression analysis is an important tool for analysing and modelling data. Here, we fit a curve/line to the data points,

in such a manner that the differences between the distance of the actual data points from the plotted curve/line is minimum. The topic will be explained in detail in the coming sections.

2. What is Linear Regression?

Linear Regression

Linear Regression is one of the most fundamental and widely known Machine Learning Algorithms which people start with. Building blocks of a Linear Regression Model are:

- Discreet/continuous independent variables
- A best-fit regression line
- Continuous dependent variable. i.e., A Linear Regression model predicts the dependent variable using a regression line based on the independent variables. The equation of the Linear Regression is:

  `Y=a+b*X + e`

  Where, a is the intercept, b is the slope of the line, and e is the error term. The equation above is used to predict the value of the target variable based on the given predictor variable(s).

3. When to use Linear Regression? Explain the equation of a straight line.

Linear regression is a way to explain the relationship between a dependent variable and one or more explanatory variables using a straight line. It is a special case of regression analysis.

Simple Linear Regression

Simple Linear regression is a method for predicting a **quantitative response** using a **single feature** ("input variable"). The mathematical equation is:

$y = \beta_0 + \beta_1 x$

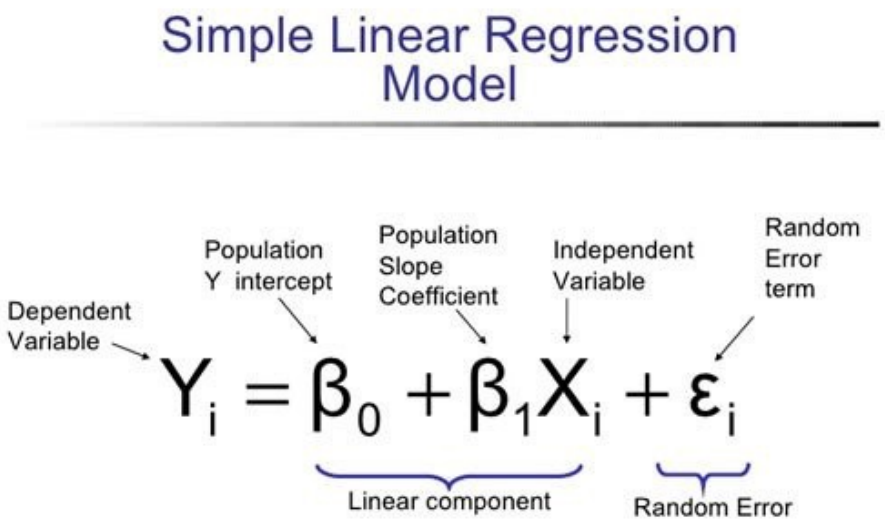
What do terms represent?

- y is the response or the target variable
- x is the feature
- β_1 is the coefficient of x
- β_0 is the intercept

β_0 and β_1 are the **model coefficients**. To create a model, we must "learn" the values of these coefficients. And once we have the value of these coefficients, we can use the model to predict the Sales!

Estimating ("Learning") Model Coefficients

The coefficients are estimated using the **least-squares criterion**, i.e., the best fit line has to be calculated that minimizes the **sum of squared residuals** (or "sum of squared errors").



Mathematically, a linear regression is defined by this equation:

$$y = bx + a + \varepsilon$$

4. What kind of plots will you use to showcase the relationship amongst the columns?

Scatter plots are used to determine the relationship between two variables.

Bar graphs are used:

To make comparisons between variables To visualize any trend in the data, i.e., they show the dependence of one variable on another Estimate values of a variable

Countplot This is a seaborn-specific function which is used to plot the count or frequency distribution of each unique observation in the categorical variable.

It is similar to a histogram over a categorical rather than quantitative variable.

Joint Plot Jointplot is seaborn library specific and can be used to quickly visualize and analyze the relationship between two variables

and describe their individual distributions on the same plot.

PairPlot We can use scatter plot to plot the relationship between two variables. But what if the dataset has more than two variables (which is quite often the case), it can be a tedious task to visualize the relationship between each variable with the other variables.

The seaborn pairplot function does the same thing for us and in just one line of code. It is used to plot multiple pairwise bivariate (two variable) distribution in a dataset. It creates a matrix and plots the relationship for each pair of columns. It also draws a univariate distribution for each variable on the diagonal axes.

Box plot Box plots are widely used in data visualization. Box plots, also known as box and whisker plots are used to visualize variations and compare different categories in a given set of data. It doesn't display the distribution in detail but is useful in detecting whether a distribution is skewed and detect outliers in the data. In a box and whisker plot:

the box spans the interquartile range a vertical line inside the box represents the median two lines outside the box, the whiskers, extending to the highest and the lowest observations represent the possible outliers in the data

5. How is the best fit line chosen?

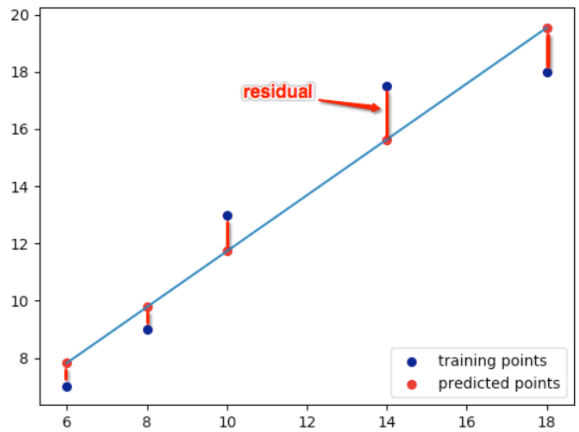
The general equation of a straight line is:

$$y = mx + b$$

It means that if we have the value of m and b, we can predict all the values of y for corresponding x. During construction of a Linear Regression Model, the computer tries to calculate the values of m and b to get a straight line. But the question is:

How Do you Know this is the best fit line?

The best fit line is obtained by minimizing the *residual*. Residual is the distance between the actual Y and the predicted Y, as shown below:



Mathematically, Residual is:

$$r = y - (mx + b)$$

Hence, the sum of the square of residuals is:

$$r_i = y_i - (mx_i + b) \quad \text{(Residual for one point)}$$

$$\sum_{i=1}^{n} r_i = \sum_{i=1}^{n} (y_i - (mx_i + b)) \quad \text{(Sum of residuals)}$$

$$R(x) = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} (y_i - (mx_i + b))^2 \quad \text{(Sum of squares of residuals)}$$

As we can see that the residual is both a function of m and b, so differentiating partially with respect to m and b will give us:

$$\frac{\partial R}{\partial m} = \sum_{i=0}^{n} 2x_i (b + mx_i - y_i)$$

$$\frac{\partial R}{\partial b} = \sum_{i=0}^{n} 2 (b + mx_i - y_i)$$

For getting the best fit line, residual should be minimum. The minima of a function occurs where the derivative=0. So, equating our corresponding derivatives to 0, we get:

$$\sum_{i=0}^{n} 2x_i (b + mx_i - y_i) = 0$$

$$\sum_{i=0}^{n} 2 (b + mx_i - y_i) = 0$$

–

$$\sum_{i=0}^{n} 2x_i b + 2mx_i^2 - 2y_i x_i = 0$$

$$\sum_{i=0}^{n} 2b + 2mx_i - 2y_i = 0$$

–

$$\sum_{i=0}^{n} 2x_i b + \sum_{i=0}^{n} 2mx_i^2 - \sum_{i=0}^{n} 2y_i x_i = 0$$

$$\sum_{i=0}^{n} 2b + \sum_{i=0}^{n} 2mx_i - \sum_{i=0}^{n} 2y_i = 0 \quad \text{(Break up the summations)}$$

–

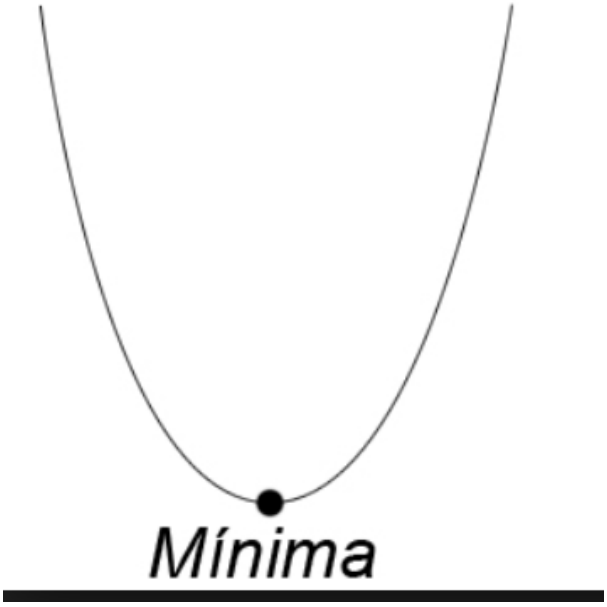
$$\sum_{i=0}^{n} x_i b + \sum_{i=0}^{n} mx_i^2 - \sum_{i=0}^{n} y_i x_i = 0$$

$$\sum_{i=0}^{n} b + \sum_{i=0}^{n} mx_i - \sum_{i=0}^{n} y_i = 0 \quad \text{(diving both sides by 2)}$$

This same equation can be written in matrix form as:

$$\begin{bmatrix} \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 \\ n & \sum_{i=0}^{n} x_i \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} y_i x_i \\ \sum_{i=0}^{n} y_i \end{bmatrix}$$

Ideally, if we'd have an equation of one dependent and one independent variable the minima will look as follows:



But as the residual's minima is dependent on two variables m and b, it becomes a *Paraboloid* and the appropriate m and b are calculated using *Gradient Descent* as shown below:

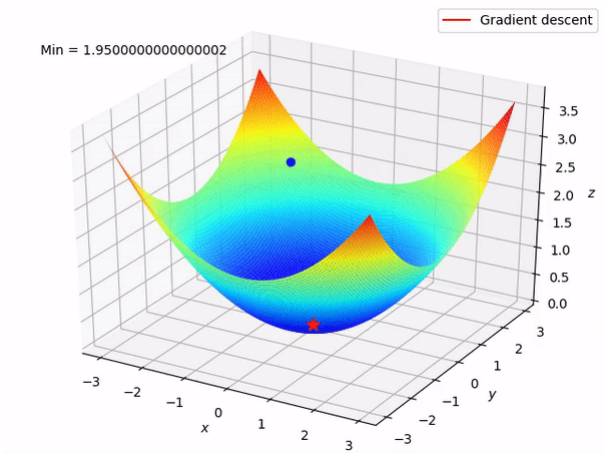


Photo:Google

6. What is gradient descent, and why is it used? Explain the maths.

As we know Error = Actual - prediction

How to control above error

you cant change the value of y and x but we can change the B0 and B1 .

decrease or minimum error by adusting B0 and B1.

Find the value of B0 and B1 then again change B0 and B1 to where error is minimum.

The process of changing the values of B0 and B1 and trying to minimum the error is called Gradient Descent.

Need to find the best value of B0 and B1 where I will get minimum error so we will find the best value of B0 and B1 where the error is minimum

Here we are taking partial derivative Bo and B1 upto such a point it almost becomes zero after that it will not change so that entire process is called GD

Partial derivative comes to lowest point

cost fn is there and to find the optimal value for that decrease the at lowest point.

GD decrese the error at lowest point.

if 10000 records GD will pass the entrie data set then it takes longer time to find the value of B0 and B1.

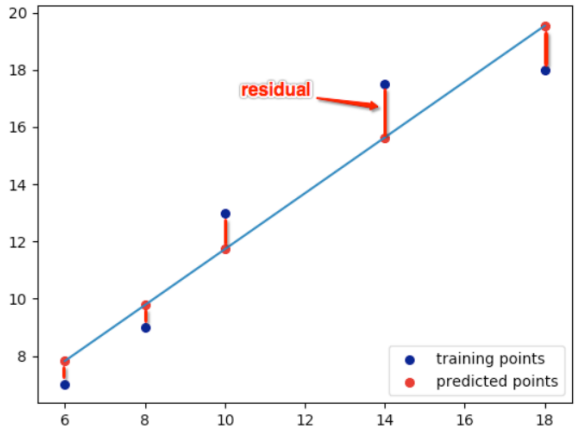
SCG is we do the batches of data. like if 10000 we divide into 10 batches

we divide into 10 parts ie 10 batches instead of passing entire dataset here we will pass the subset of ie batch .

1st subset or batch find B0 and B1 whatever it has learn it will pass to next batch i.e 2nd batch this process si continued

so divide data into batch so its faster also SGD doing parllel computing.

Residual is the distance between the actual Y and the predicted Y, as shown below:



Mathematically, Residual is:

$$r = y - (mx + b)$$

Hence, the sum of the square of residuals is:

$$r_i = y_i - (mx_i + b) \quad \text{(Residual for one point)}$$

$$\sum_{i=1}^{n} r_i = \sum_{i=1}^{n} (y_i - (mx_i + b)) \quad \text{(Sum of residuals)}$$

$$R(x) = \sum_{i}^{n} r_i^2 = \sum^{n} (y_i - (mx_i + b))^2 \quad \text{(Sum of squares of residuals)}$$

As we can see that the residual is both a function of m and b, so differentiating partially with respect to m and b will give us:

$$\frac{\partial R}{\partial m} = \sum_{i=0}^{n} 2x_i (b + mx_i - y_i)$$

$$\frac{\partial R}{\partial b} = \sum_{i=0}^{n} 2 (b + mx_i - y_i)$$

For getting the best fit line, residual should be minimum. The minima of a function occurs where the derivative=0. So, equating our corresponding derivatives to 0, we get:

$$\sum_{i=0}^{n} 2x_i (b + mx_i - y_i) = 0$$

$$\sum_{i=0}^{n} 2 (b + mx_i - y_i) = 0$$

–

$$\sum_{i=0}^{n} 2x_i b + 2mx_i^2 - 2y_i x_i = 0$$

$$\sum_{i=0}^{n} 2b + 2mx_i - 2y_i = 0$$

–

$$\sum_{i=0}^{n} 2x_i b + \sum_{i=0}^{n} 2mx_i^2 - \sum_{i=0}^{n} 2y_i x_i = 0$$

$$\sum_{i=0}^{n} 2b + \sum_{i=0}^{n} 2mx_i - \sum_{i=0}^{n} 2y_i = 0 \quad \text{(Break up the summations)}$$

–

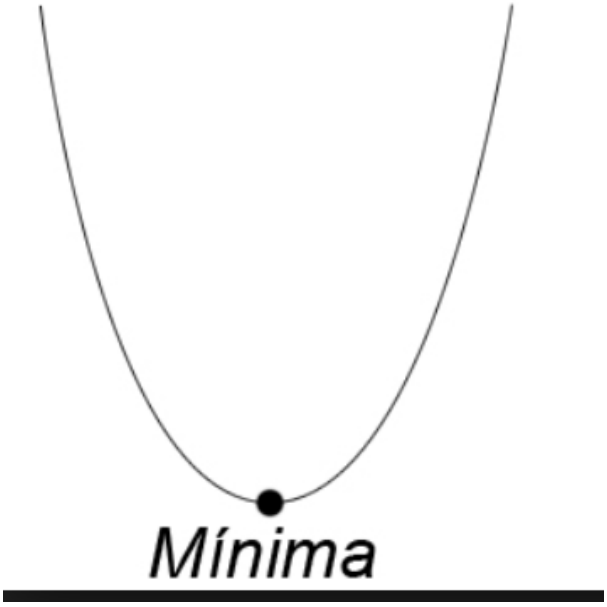
$$\sum_{i=0}^{n} x_i b + \sum_{i=0}^{n} mx_i^2 - \sum_{i=0}^{n} y_i x_i = 0$$

$$\sum_{i=0}^{n} b + \sum_{i=0}^{n} mx_i - \sum_{i=0}^{n} y_i = 0 \quad \text{(diving both sides by 2)}$$

This same equation can be written in matrix form as:

$$\begin{bmatrix} \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 \\ n & \sum_{i=0}^{n} x_i \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} y_i x_i \\ \sum_{i=0}^{n} y_i \end{bmatrix}$$

Ideally, if we'd have an equation of one dependent and one independent variable the minima will look as follows:



But as the residual's minima is dependent on two variables m and b, it becomes a *Paraboloid* and the appropriate m and b are calculated using *Gradient Descent* as shown below:

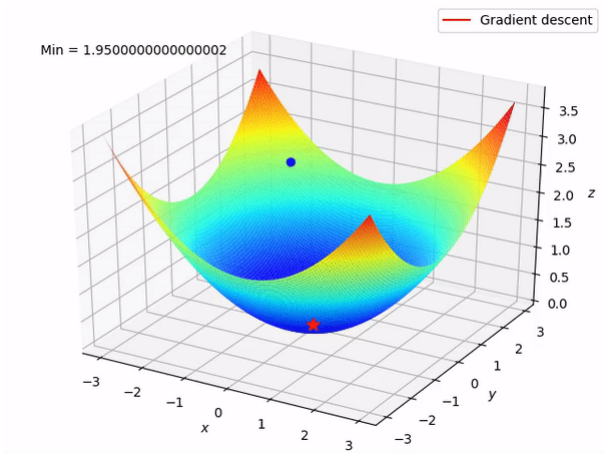


Photo:Google

7. What are residuals?

In [ ]:
```
Residual(e) refers to the difference between observed value(y) vs predicted v


Residual = Observed value - predicted value
e = y - ŷ
```

8. What is correlation?

Correlation quantifies the direction and strength of the relationship between two numeric variables.

scatterplot is the best place to start. A scatterplot (or scatter diagram) is a graph of the paired (x, y)

```
sample data with a horizontal x-axis and a vertical y-axis.

Each individual (x, y) pair is plotted as a single point.
```

Correlation and regression analysis are related in the sense that both deal with relationships among variables. The correlation coefficient is a measure of linear association between two variables. Values of the correlation coefficient are always between -1 and +1. A correlation coefficient of +1 indicates that two variables are perfectly related in a positive linear sense, a correlation coefficient of -1 indicates that two variables are perfectly related in a negative linear sense, and a correlation coefficient of 0 indicates that there is no linear relationship between the two variables. For simple linear regression, the sample correlation coefficient is the square root of the coefficient of determination, with the sign of the correlation coefficient being the same as the sign of b1, the coefficient of x1 in the estimated regression equation.

Neither regression nor correlation analyses can be interpreted as establishing cause-and-effect relationships. They can indicate only how or to what extent variables are associated with each other. The correlation coefficient measures only the degree of linear association between two variables. Any conclusions about a cause-and-effect relationship must be based on the judgment of the analyst.

Correlation coefficient The degree of association is measured by a correlation coefficient, denoted by r. It is sometimes called Pearson's correlation coefficient after its originator and is a measure of linear association. If a curved line is needed to express the relationship, other and more complicated measures of the correlation must be used.

The correlation coefficient is measured on a scale that varies from + 1 through 0 to - 1. Complete correlation between two variables is expressed by either + 1 or -1. When one variable increases as the other increases the correlation is positive; when one decreases as the other increases it is negative. Complete absence of correlation is represented by 0. Figure 11.1 gives some graphical representations of correlation.

Correlation and Regression are the two analysis based on multivariate distribution. A multivariate distribution is described as a distribution of multiple variables.

Correlation is described as the analysis which lets us know the association or the absence of the relationship between two variables ‘x’ and ‘y’. On the other end,

Regression analysis, predicts the value of the dependent variable based on the known value of the independent variable, assuming that average mathematical relationship between two or more variables.

Key Differences Between Correlation and Regression The points given below, explains the difference between correlation and regression in detail:

A statistical measure which determines the co-relationship or association of two quantities is known as Correlation. Regression describes how an independent variable is numerically related to the dependent variable. Correlation is used to represent the linear relationship between two variables. On the contrary, regression is used to fit the best line and estimate one variable on the basis of another variable.

In correlation, there is no difference between dependent and independent variables i.e. correlation between x and y is similar to y and x. Conversely, the regression of y on x is different from x on y. Correlation indicates the strength of association between variables. As opposed to, regression reflects the impact of the unit change in the independent variable on the dependent variable. Correlation aims at finding a numerical value that expresses the relationship between variables. Unlike regression whose goal is to predict values of the random variable on the basis of the values of fixed variable.

Basis	Correlation	Regression
Meaning	A statistical measure that defines co-relationship or association of two variables.	Describes how an independent variable is associated with the dependent variable.
Dependent and Independent variables	No difference	Both variables are different.
Usage	To describe a linear relationship between two variables.	To fit the best line and estimate one variable based on another variable.
Objective	To find a value expressing the relationship between variables.	To estimate values of a random variable based on the values of a fixed variable.

9. What is multicollinearity?

Definition: The purpose of executing a Linear Regression is to predict the value of a dependent variable based on certain independent variables.

So, when we perform a Linear Regression, we want our dataset to have variables which are independent i.e., we should not be able to define an independent variable with the help of another independent variable because now in our model we have two variables which can be defined based on a certain set of independent variables which defeats the entire purpose.

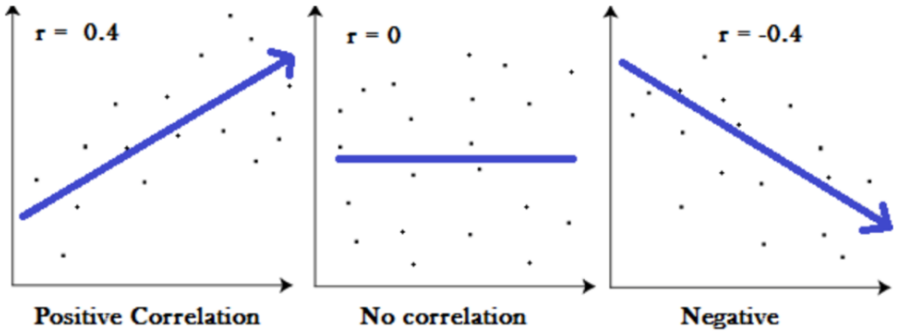
- Multi-collinearity is the statistical term to represent this type of a relation amongst the independent variable- when the independent variables are not so independent😉.
- We can define multi-collinearity as the situation where the independent variables (or the predictors) have strong correlation amongst themselves.

The results will be between -1 and 1. You will very rarely see 0, -1 or 1. You'll get a number somewhere in between those values. The closer the value of r gets to zero, the greater the variation the data points are around the line of best fit.

High correlation: .5 to 1.0 or -0.5 to 1.0

Medium correlation: .3 to .5 or -0.3 to .5

Low correlation: .1 to .3 or -0.1 to -0.3

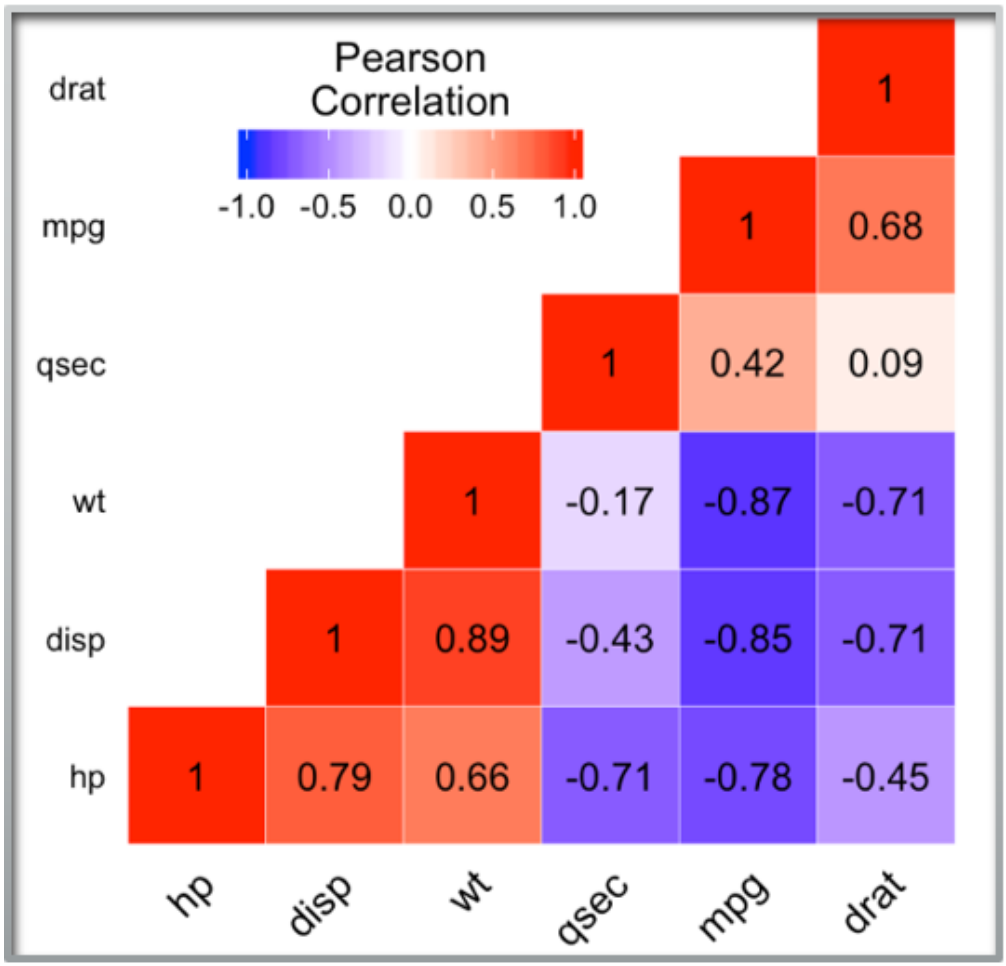


10. How to detect multicollinearity?

Detection

- **Correlation Matrices and Plots:** for correlation between all the X variables.

```
This plot shows the extent of correlation between the independe
nt variable. Generally, a correlation greater than 0.9 or less th
an -0.9 is to be avoided.
```



- **Variance Inflation Factor:** Regression of one X variable against other X variables.

$$VIF = \frac{1}{(1 - Rsquared)}$$

The VIF factor, if greater than 10 shows extreme correlation between the variables and then we need to take care of the correlation.

11. What are the remedies for multicollinearity?

Remedies for Multicollinearity

- **Do Nothing:** If the Correlation is not that extreme, we can ignore it. If the correlated variables are not used in solving our business question, they can be ignored.
- **Remove One Variable**: Like in dummy variable trap
- **Combine the correlated variables:** Like creating a seniority score based on Age and Years of experience
- Principal Component Analysis

12. What is the R-Squared Statistics?

R^2 statistics

The R-squared statistic provides a measure of fit. It takes the form of a proportion—the proportion of variance explained—and so it always takes on a value between 0 and 1. In simple words, it represents how much of our data is being explained by our model. For example, R^2 statistic = 0.75, it says that our model fits 75 % of the total data set. Similarly, if it is 0, it means none of the data points is being explained and a value of 1 represents 100% data explanation. Mathematically R^2 statistic is calculated as :

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

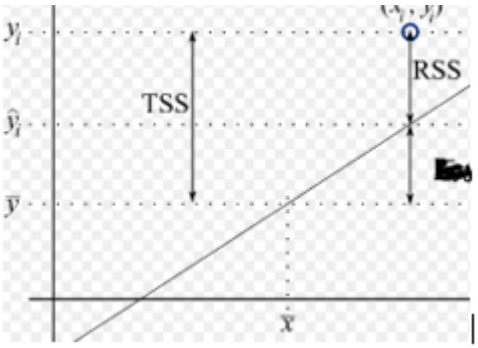
Where RSS: is the Residual Sum of squares and is given as :

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

RSS is the residual(error) term we have been talking about so far. And, TSS: is the Total sum of squares and given as :

$$\text{TSS} = \sum (y_i - \bar{y})^2$$

TSS is calculated when we consider the line passing through the mean value of y, to be the best fit line. Just like RSS, we calculate the error term when the best fit line is the line passing through the mean value of y and we get the value of TSS.

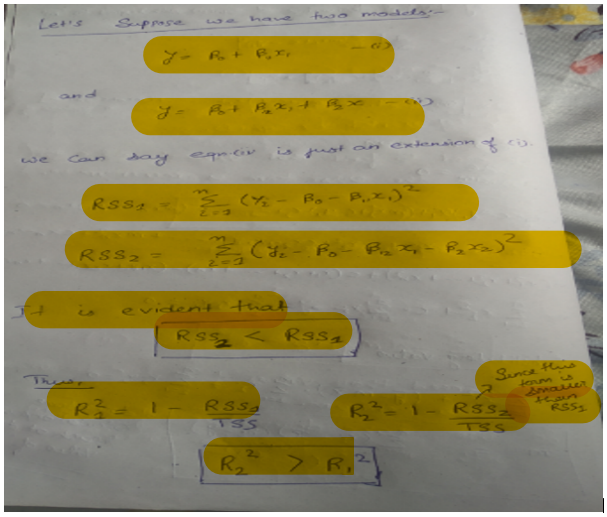


The closer the value of R2 is to 1 the better the model fits our data. If R2 comes below 0(which is a possibility) that means the model is so bad that it is performing even worse than the average best fit line.

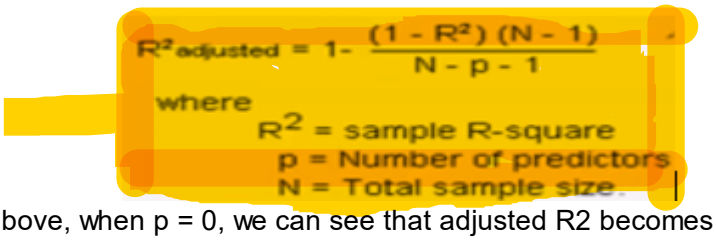
13. What is an adjusted R-Squared Statistics?

Adjusted R^2 statistics

As we increase the number of independent variables in our equation, the R2 increases as well. But that doesn't mean that the new independent variables have any correlation with the output variable. In other words, even with the addition of new features in our model, it is not necessary that our model will yield better results but R2 value will increase. To rectify this problem, we use Adjusted R2 value which penalises excessive use of such features which do not correlate with the output data. Let's understand this with an example:



We can see that R2 always increases with an increase in the number of independent variables. Thus, it doesn't give a better picture and so we need Adjusted R2 value to keep this in check. Mathematically, it is calculated as:



In the equation above, when p = 0, we can see that adjusted R2 becomes equal to R2. Thus, adjusted R2 will always be less than or equal to R2, and it penalises the excess of independent variables which do not affect the dependent variable.

14. Why do we use adj R-squared?

R-squared is the fraction of total variation in the response that can be attributed to the variation in the predictors.

R-squared-adjusted adds a penalty to R-squared for model complexity.

R-squared will always increase if a new predictor is added to a model. Model selection based on large R-squared could result in selecting an overfit model.

But, as predictors are added to a model R-squared-adjusted will increase to a maximum and the decrease. it starts to decrease when the additional predictor's ability is not worth the additional model complexity. (Which might actually be overfitting.) Model selection based on large R-squared-adjusted should not result in an overfit model

15. Why adj R-squared decreases when we use incompetent variables?

R2 shows the linear relationship between the independent variables and the dependent variable. It is defined as 1−SSESSTO which is the sum of squared errors divided by the total sum of squares. SSTO=SSE+SSR which are the total error and total sum of the regression squares. As independent variables are added SSR will continue to rise (and since SSTO is fixed) SSE will go down and R2 will continually rise irrespective of how valuable the variables you added are.

The Adjusted R2 is attempting to account for statistical shrinkage. Models with tons of predictors tend to perform better in sample than when tested out of sample. The adjusted R2 "penalizes" you for adding the extra predictor variables that don't improve the existing model. It can be helpful in model selection. Adjusted R2 will equal R2 for one predictor variable. As you add variables, it will be smaller than R2.

R^2 explains the proportion of the variation in your dependent variable (Y) explained by your independent variables (X) for a linear regression model.

While adjusted R^2 says the proportion of the variation in your dependent variable (Y) explained by more than 1 independent variables (X) for a linear regression model.

Adjusted R^2 is the better model when you compare models that have a different amount of variables.

The logic behind it is, that R^2 always increases when the number of variables increases. Meaning that even if you add a useless variable to you model, your R^2 will still increase. To balance that out, you should always compare models with different number of independent variables with adjusted R^2.

Adjusted R^2 only increases if the new variable improves the model more than would be expected by chance.

16. How to interpret a Linear Regression model?

Interpreting the model

How do we interpret the coefficient for spends on TV ad (β_1)?

- A "unit" increase in spends on a TV ad is **associated with** a 0.047537 "unit" increase in Sales.
- Or, an additional \$1,000 on TV ads is **translated to** an increase in sales by 47.53 Dollars.

As an increase in TV ad expenditure is associated with a **decrease** in sales, β_1 would be **negative**.

17. What is the difference between fit, fit_transform and predict methods?

he model is fit on the training dataset by calling the fit() function and passing in the training dataset.

Finally, we can evaluate the model by first using it to make predictions on the training dataset by calling predict() and then comparing the predictions to the expected class labels and calculating the accuracy.

We apply fit on the training dataset and use the transform method on both - the training dataset and the test dataset.

Here the basic difference between .fit() & .fit_transform():

.fit(): is use in the Supervised learning having two object/parameter(x,y) to fit model and make model to run, where we know that what we are going to predict

.fit_transform(): is use in Unsupervised Learning having one object/parameter(x), where we don't know, what we are going to predict.

hese methods are used to center/feature scale the given data. It basically helps to normalize the data within a particular range

For this, we use Z-score method.

$Z = (x - \mu)/\sigma$

We do this on the training set of data.

1.Fit(): Method calculates the parameters μ and σ and saves them as internal objects.

2.Transform(): Method using these calculated parameters apply the transformation to a particular dataset.

3.Fit_transform(): joins the fit() and transform() method for transformation of dataset.

fit() => predict() is almost used for all classifiers in SKLearn (Knn, SVC, Logistic Reg, NaiveBayes … etc)

fit() => transform() or fit_transform() used for Scalers, and NLP Vectorizers'

fit: when you want to train your model without any pre-processing on the data transform: when you want to do pre-processing on the data using one of the functions from sklearn.preprocessing fit_transform(): It's same as calling fit() and then transform() - a shortcut

18. How do you plot the least squared line?

Plotting the Least Squares Line

In [26]:
```
# create a DataFrame with the minimum and maximum values of TV
X_new = pd.DataFrame({'TV': [data.TV.min(), data.TV.max()]})
X_new.head()
```

Out[26]:

	TV
0	0.7
1	296.4

In [27]:
```
# make predictions for those x values and store them
preds = lm.predict(X_new)
preds
```

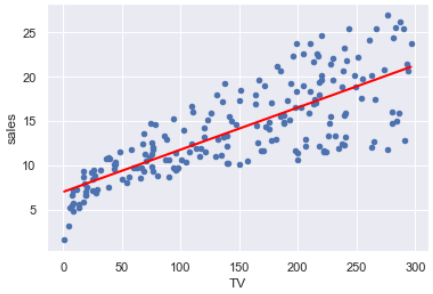
Out[27]:
```
array([ 7.0658692 , 21.12245377])
```

In [28]:
```
# first, plot the observed data
data.plot(kind='scatter', x='TV', y='sales')

# then, plot the least squares line
plt.plot(X_new, preds, c='red', linewidth=2)
```

```
'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have preceden
ce in case its length matches with 'x' & 'y'.  Please use a 2-D array with a single row if you really want to specify the
same RGB or RGBA value for all points.
```

Out[28]:
```
[<matplotlib.lines.Line2D at 0x269b2214b38>]
```



19. What are Bias and Variance? What is Bias Variance Trade-off?

==========

Linear Regression

Linear regression describes the relationship between a dependent variable and an independent variable. Think of this as an X variable and a Y variable.

What are some assumptions of a linear regression model?

The relationship between the two variables is linear.

All variables are multivariate normal.

There isn't much multicollinearity among the dependent variables. A good way to test this is variance inflation.

There is little to no autocorrelation in the data. If you are unfamiliar with autocorrelation, autocorrelation occurs when the residuals of the variables in the model are not independent of each other.

There is homoscedasticity. That is, the size of the error does not vary by the sizes of the independent variables. The error does not increase substantially if your variables get larger or smaller.

The residuals of the linear regression should be normally distributed around a mean of 0.

What are some approaches to solving this problem?

We want to find a line that best captures the essence of the data provided. Why? In the context of machine learning, we want to find a line that predicts the values of Y based upon values of X. The two variables to measure the effectiveness of your model are bias and variance.

Bias- the error or difference between points provided and points captured on your line in your training set

Variance- error from sensitivity to small fluctuations in the training set

Underfit Model

We can find a line that captures a general direction but does not capture many of the points. Since it does not capture many of the points well, it has a high bias or high error. Since it does not grasp many of the points in the graph, it also has a low variance. This would be called an underfit model.

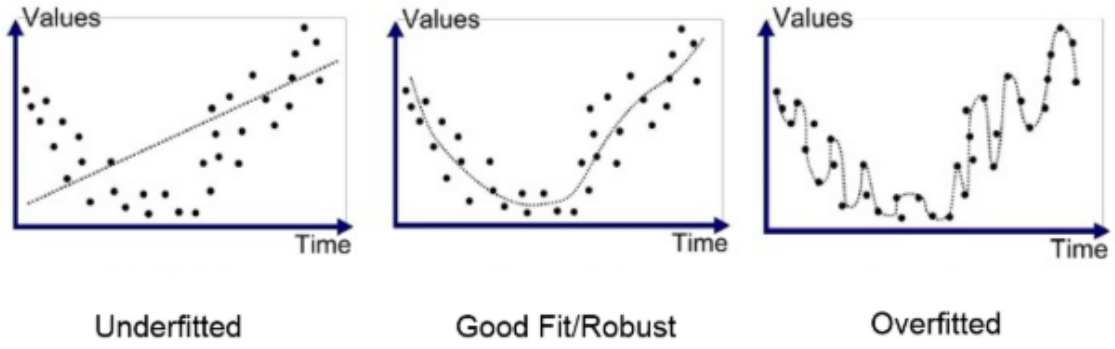
Good Fit Model

We can find a line that captures the general direction of the points but might not capture every point in the graph. This would be a good fit model.

Overfit Model

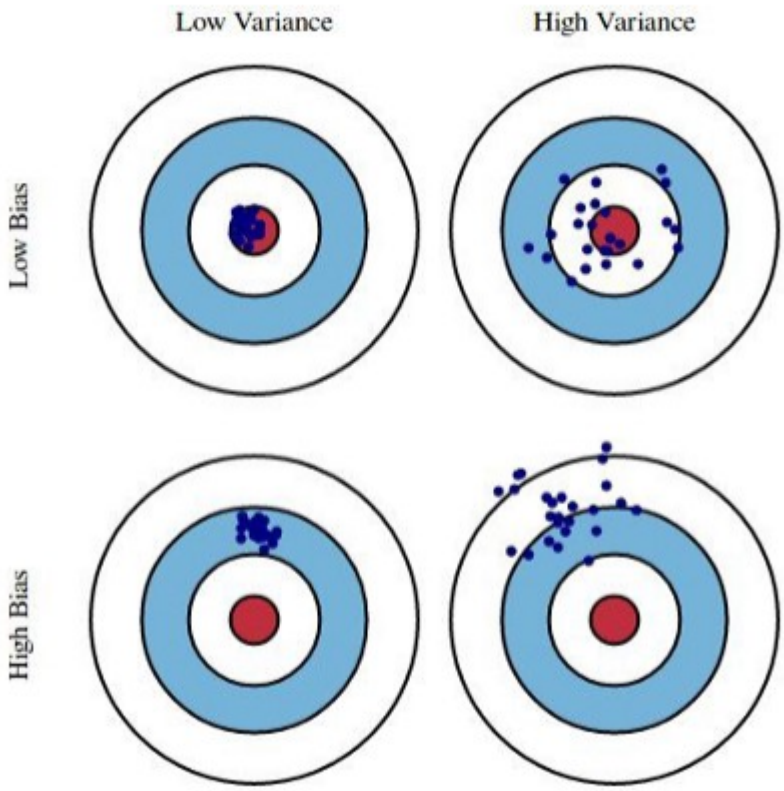
Lastly, we can find a line that captures every single point in the graph. This would be called an overfit model.

Below are good images to describe the differences in models.



Bias-Variance Trade-off

This situation above describes the bias-variance tradeoff. Bias measures the error between what the model captures and what the available data is showing. Variance is the error from sensitivity to small fluctuations in the available data. That is, a model with high variance effectively captures random noise in the available data rather than the intended outputs. We ideally want to find a line that has low bias and low variance.



Why would a line with high variance be a bad thing?

After all, wouldn't a line that captures every point be the ideal line? Sadly This line captures the peculiar nuances of the sample data well but it will n necessarily perform as well on unseen data. These nuances of the sample data are the outliers and other unique characteristics of the sample data, characteristics that might not be very true on unseen data.

What are some solutions to prevent overfitting on sample data?

While there are a number of solutions to prevent or reduce overfitting on sample data, in this article I will talk about Lasso, Ridge, and Elastic Net.

Reducing Overfit with regularization

Lasso, Ridge, and Elastic Net models are forms of regularized linear techniques found in General Linear Models.

Ridge

Ridge assigns a penalty that is the squared magnitude of the coefficients to the loss function multiplied by lambda. As Lasso does, ridge also adds a penalty to coefficients the model overemphasizes. The value of lambda also plays a key role in how much weight you assign to the penalty for the coefficients. The larger your value of lambda, the more likely your coefficients get closer and closer to zero. Unlike lasso, the ridge model will not shrink these coefficients to zero.

Ridge Formula: Sum of Error + Sum of the squares of coefficients

$$L = \sum(\hat{Y}i - Yi)^2 + \lambda \sum \beta^2$$

What are some limitations of Ridge?

Ridge does not eliminate coefficients in your model even if the variables are irrelevant. This can be negative if you have more features than observations.

Elastic Net

Elastic Net combines characteristics of both lasso and ridge. Elastic Net reduces the impact of different features while not eliminating all of the features.

The formula as you can see below is the sum of the lasso and ridge formulas.

Elastic Net Formula: Ridge + Lasso

$$L = \sum(\hat{Y}i - Yi)^2 + \lambda \sum \beta^2 + \lambda \sum |\beta|$$

To conclude, Lasso, Ridge, and Elastic Net are excellent methods to improve the performance of your linear model. This includes if you are running a neural network, a collection of linear models. Lasso will eliminate many features, and reduce overfitting in your linear model. Ridge will reduce the impact of features that are not important in predicting your y values. Elastic Net combines feature elimination from Lasso and feature coefficient reduction from the Ridge model to improve your model's predictions.

$$L = \sum(\hat{Y}i - Yi)^2 + \lambda \sum |\beta|$$

Looking at the formula, Lasso adds a penalty equal to the absolute value of the magnitude of the coefficients multiplied by lambda. The value of lambda also plays a key role in how much weight you assign to the penalty for the coefficients. This penalty reduces the value of many coefficients to zero, all of which are eliminated.

What is the significance of adding a penalty to coefficients in lasso?

Lasso adds a penalty to coefficients the model overemphasizes. This reduces the degree of overfitting that occurs within the model.

What are some limitations of the Lasso model?

Lasso does not work well with multicollinearity. If you are unfamiliar, multicollinearity occurs when some of the dependent variables are correlated with each other. Why? Lasso might randomly choose one of the multicollinear variables without understanding the context. Such an action might eliminate relevant independent variables.

Why is regularization important?

Regularization favors simpler models to more complex models to prevent your model from overfitting to the data. How so? They address the following concerns within a model: variance-bias tradeoff, multicollinearity, sparse data handling(i.e. the situation where there are more observations than features), feature selection, and an easier interpretation of the output.

Lasso

Lasso stands for Least Absolute Shrinkage Selector Operator. Lasso assigns a penalty to the coefficients in the linear model using the formula below and eliminates variables with coefficients that zero. This is called shrinkage or the process where data values are shrunk to a central point such as a mean.

Lasso Formula: Lasso = Sum of Error + Sum of the absolute value of coefficients

$$L = \sum(\hat{Y}i - Yi)^2 + \lambda \sum |\beta|$$

20. What is the null and alternate hypothesis?

Hypothesis testing is Closely related to confidence intervals. We start with a **null hypothesis** and an **alternate hypothesis** (that is opposite to the null). Then, we check whether the data **rejects the null hypothesis** or **fails to reject the null hypothesis**.

("Failing to reject" the null hypothesis does not mean "accepting" the null hypothesis. The alternative hypothesis might indeed be true, but that we just don't have enough data to prove that.)

The conventional hypothesis test is as follows:

- **Null hypothesis:** No relationship exists between TV advertisements and Sales (and hence β_1 equals zero).
- **Alternative hypothesis:** There exists a relationship between TV advertisements and Sales (and hence, β_1 is not equal to zero).

How do we test this? We reject the null hypothesis (and thus believe the alternative hypothesis) if the 95% confidence interval **does not include zero**. The **p-value** represents the probability of the coefficient actually being zero.

21. What is multiple linear regression?

Multiple Linear Regression

we'll include multiple features and create a model to see the relationship between those features and the label column. This is called **Multiple Linear Regression**.

$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$

Each x represents a different feature, and each feature has its own coefficient. In this case:

$y = \beta_0 + \beta_1 \times TV + \beta_2 \times Radio + \beta_3 \times Newspaper$

Let's use Statsmodels to estimate these coefficients

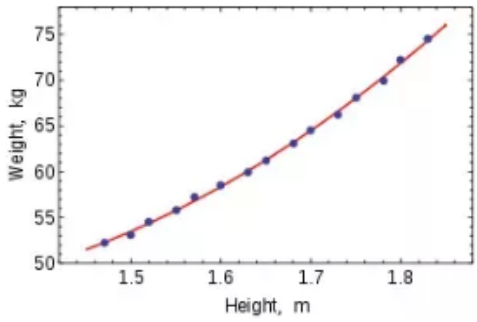
22. What is the OLS method? Derive the formulae used in the OLS method.

Ordinary least squares regression is a statistical method that produces the one straight line that minimizes the total squared error.

Ordinary Least Squares. It is the best unbiased linear estimator for the statistics of the linear regression. There are other methods as well, which sometimes work as good as OLS, but they have different drawbacks.

The core of OLS lies in minimizing the sum of (y - bX)^2, whereas y and b are vectors and X is a matrix of regressors. Note that both Y and X are known variables (measured observations, given data, whatever you want to call them, they are known to you and that's my point). b is the unknown variable and by minimizing this sum of squares, you are minimizing the errors of the model: y = bX + erro

In statistics, ordinary least squares (OLS) or linear least squares is a method for estimating the unknown parameters in a linear regression model. This method minimizes the sum of squared vertical distances between the observed responses in the dataset and the responses predicted by the linear approximation.



Ols estimates refer to ordinary least square method for estimating the coefficient of the indepedent variables in your model. OLS is the most common method used for estimating a regression equation because its estimates are BLUE.

B - Best

L - linear

U - unbiased

E - estimate

And they are blue because OLS gives the minimum variance of Beta.

After we get the OLS estimates, commonly known as beta/s. We then want to do a hypothesis test. There are various methods for that, depending on the model at hand.

T test is carried out when indiviual beta testing is to be done. It follows standard normal distribution.

F test is carried out when multiple beta testing is to be done. It follows chi square distribution.

An F-test is any statistical test in which the sampling distribution of test statistic has an F-distribution when the null hypothesis is true. Similarly, any statistical test that uses the F distribution can be called F test.

We can use F-test for different purpose.

F-test of the equality of two variances: It is used to compare the variances of two quantitative variables. Example: H0:σ21=σ22,H1:σ2≠σ22 ANOVA: We use ANOVA to compare more than two means. F-test in ANOVA is used to assess whether the expected values of a quantitative variable within several pre-defined groups differ from each other. Example: Medical trial compares four treatments. The ANOVA F-test can be used to assess whether any of the treatments is on average superior, or inferior, to the others versus the null hypothesis that all four treatments yield the same mean response. Regression problems: We do t-test linear model for same data.different for individual coefficient significance in regression. We can use F-test for overall significance of the model. It helps to compare the fit of Example: H0:β1=β2=β3……=βk=0,H1: At least one coefficient is significant.

Ordinary Least Squares (OLS) is a general method for deciding what parameter estimates provide the ‘best’ solution. “Least squares” refers to what you want to minimize: the sum of squared prediction errors (SSE) The ‘best’ or optimal model is often defined as the one that minimizes sum of squared prediction errors (see note at end for more information).

Multiple regression is one of many statistical techniques for which parameter estimates can be obtained based on OLS method. Other statistics (such as the sample mean M) can also found using OLS methods.

For simple models, the same parameter estimates that work best can also be obtained using Maximum Likelihood (ML) estimation methods (in other words, OLS and ML are two of the most common methods of paramer estimation, there are many others).

For complicated models, such as Structural Equation Models, OLS methods do not work; instead, we have to use ML or other estimation methods.

This question is a great classic question that you see in a linear models class.

Assume

Yi=XTi β+ϵi,

For i=1,...,n , where Xi is a vector of independent variables, β is an unknown parameter and ϵi are iid distributed according to a mean zero distribution with finite variance.

Notice I have already made some assumptions here: The observations are independent, the errors have mean zero and finite variance. Some of these assumptions can be relaxed.

For an unbiased estimate of β you just need E(Yi)=XTiβ .

For the OLS estimator to be the best linear unbiased estimator you just need ϵi to all have the same distribution.

For your t- and f-tests to be valid you need for the errors ϵi to be normal with the same variance for all subjects.

If ϵi are not normal but still all have the same distribution, then you can use the central limit theorem to approximate test statistic distributions.

There are several ways to derive OLS; but at least one way - essentially makes only these assumptions

model is linear coefs should minimize squared error There's no "features have gaussian distribution" or "residuals have gaussian distribution" or anything of the sort baked into this.

You don't need any of those assumptions to derive OLS.

F-Test is based on F- distribution and is used to compare the variance of the two independent samples.this is also used in the context of analysis of variance for judging the significance of more then two samlpe.

T-TEST is based on t-distribution and is considered an appropriate test for judging the significance of a sample mean or for judging the significance of difference between the means of two samples in case of small sample when population variance in unknown.

23. What is the p-value? How does it help in feature selection?

TV and Radio have positive p-values, whereas Newspaper has a negative one. Hence, we can reject the null hypothesis for TV and Radio that there is no relation between those features and Sales, but we fail to reject the null hypothesis for Newspaper that there is no relationship between newspaper spends and sales. The expenses on bot TV and Radio ads arepositively associated with Sales, whereas the expense on newspaper ad is slightly negatively associated with the Sales. This model has a higher value of R-squared (0.897) than the previous model, which means that this model explains more variance and provides a better fit to the data than a model that only includes the TV.

24. How to handle categorical values in the data?

The variable is considered categorical, when it describes some qualitative property and there is a limited number of option for its values. Here are some examples: Weather conditions: "sun", "rain", "overcast", "snow", etc. Car manufacturers: "Toyota", "Ford", "Mercedes", etc. Exam

grades: “A”, “B”, “C”, “D” and “F” Example 3 can be classified as ordinal variable meaning you can order its values in a meaningful way: “A” is better than “B” and so on. In other words, ordinal variable allows you to compare its values with each other. On the other hand, we cannot order (or compare) values from examples 1 and 2 in a similar way. Such variables are called nominal. Encoding Categorical Data Many machine learning algorithms cannot handle categorical variables as they require all inputs to be numerical. For this reason, we need to find a way for transforming values of categorical variable to numbers. The general idea behind the most popular methods for encoding categorical data is to come up with a mapping of each category to numbers.

Fig. 1. Example of black-box (well, orange in our case) magic for encoding categorical variable In the Figure 1 the mapping is completely random. However, we want to give to our machine learning algorithm a better sense of our data in order to make more accurate results. Thus, let’s discuss some more smart approaches than random encoding.

1. Encoding Ordinal Data Ordinal data is a piece of cake to encode. Let’s recall the example 3 from the above (exam grades). We can keep the same relationship between different grades if we assign larger numbers to better grades. One of the possible mappings for this case is: “A” → 5; “B” → 4; “C” → 3; “D” → 2; “F” → 1. Note, you can also assign smaller numbers for better grades and it basically doesn’t matter, as long as you are consistent with the encodings. Snippet 1. Jupyter Notebook Example of Ordinal Encoding
2. Encoding Nominal Data Nominal data is more challenging to encode. We will discuss several methods in the order of increasing complexity, so feel free to stop whenever you feel like :) 2.1 One-Hot Encoding One-Hot encoding is one of the most popular methods for encoding categorical data. For each unique category it creates a separate column with binary values (0 or 1). In each of these new columns there is a value of 1 if the corresponding category is present in this row and 0 otherwise. Too convoluted, right? Let’s jump to example! Snippet 2. Jupyter Notebook Example of One-Hot Encoding These method has several disadvantages for machine learning algorithms: Data matrix of dummy variables is sparse (much more zeros than ones) Categorical features with high cardinality (number of unique categories) can blow your PC memory (or at least slow down your code), because for each category a separate column will be created. 2.2 Label Encoding This method maps a single categorical column to just one numeric column which is a great advantage in comparison to One-Hot encoding. In label encoding every distinctive category is mapped to some arbitrary number. Thus, this approach does not preserve any relationship between values in categorical variable. The simplest way to implement it is to enumerate all unique categories and use these numbers as encoded values. Snippet 3. Jupyter Notebook Example of Label Encoding (Manual Way) You can also use sklearn.preprocessing package or pandas factorize method to achieve the same encoding. The only difference can be seen when there are NaN values presented in the data: Aforementioned example with “Manual Way” will preserve NaN values in the encoded data. Pandas factorize method will encode NaN values as -1. sklearn.preprocessing package will break when encounter a NaN value. Approaches 1 and 2 give you a chance to fix NaN values later e.g. via imputing them, assigning it to a separate category (e.g. “missing”) or simply dropping it. Snippet 4. Jupyter Notebook Example of Label Encoding (Using sklearn package) Snippet 5. Jupyter Notebook Example of Label Encoding (Using pandas.factorize method) Notice that the numbers assigned to different categories in Snippet 4 and Snippet 5 are slightly different. But conceptually they are the same, because as we mentioned above, the goal of label encoding is to map every distinctive category to some arbitrary number. 2.3 Target Encoding Numbers obtained by

label encoding can mislead machine learning model. Let's take a look at the Snippet 5. The encodings have values from 0 to 2 (remember, this assignment was random). There is a trivial relationship between them: 2>1>0. But what does it mean for Mercedes to be "greater" than Ford? Well, you may argue that Mercedes is on average more expensive than Ford, but what if we had "apple" and "orange" categories instead? Right, this makes no sense. Thus, we want to encode our categories in a more meaningful for machine learning model way. As we cannot generate additional information from the thin air (ohhh, right, besides frequency encoding…), we will need a target (or response) variable that needs to be predicted. Let's say we have a dataset describing various neighborhoods across US and our goal is to predict an average household income. One of the features in the dataset is State (which is a categorical variable). And our target variable is numeric. The general idea behind target encoding is that categories which correspond to larger values of target variable will be encoded with larger numbers. Wow, there were 3 paragraphs of text without examples… Let's fix it! Snippet 6. Jupyter Notebook Example of Target Encoding As you can see in Snippet 6, after examining the data it looks like households in California (CA) have the highest average household income and in Texas (TX) it is the lowest among the states presented in our dataset. As you can see from the encodings, they indicate similar relationship: CA category gets the highest encoding value, TX gets the lowest, and NY gets the intermediate one

his involves two steps:

Integer Encoding One-Hot Encoding

1. Integer Encoding As a first step, each unique category value is assigned an integer value.

For example, "red" is 1, "green" is 2, and "blue" is 3.

This is called a label encoding or an integer encoding and is easily reversible.

For some variables, this may be enough.

The integer values have a natural ordered relationship between each other and machine learning algorithms may be able to understand and harness this relationship.

For example, ordinal variables like the "place" example above would be a good example where a label encoding would be sufficient.

2. One-Hot Encoding For categorical variables where no such ordinal relationship exists, the integer encoding is not enough.

In fact, using this encoding and allowing the model to assume a natural ordering between categories may result in poor performance or unexpected results (predictions halfway between categories).

In this case, a one-hot encoding can be applied to the integer representation. This is where the integer encoded variable is removed and a new binary variable is added for each unique integer value.

In the "color" variable example, there are 3 categories and therefore 3 binary variables are needed. A "1" value is placed in the binary variable for the color and "0" values for the other colors.

For example:

red, green, blue 1, 0, 0 0, 1, 0 0, 0, 1 1 2 3 4 red, green, blue 1, 0, 0 0, 1, 0 0, 0, 1 The binary variables are often called “dummy variables” in other fields, such as statistics.

25. What is regularization, and why do we need it?

Regularization is used to avoid overfitting of the model on training data.

Lasso(L1) and Ridge(L2) are the commonly used regularization techniques

Ridge regression don’t zero out coefficients, it penalizes the square of coefficients

In Lasso, some of the coefficients can be zero, means that it does variable selection as well. Lasso penalizes coefficients unlike ridge (which penalizes squares of coefficients)

Both Ridge and Lasso are combined to get a hybrid regularization technique called as elastic net.

26. Explain Ridge Regression.

In [ ]:

27. Explain Lasso Regression.

Type *Markdown* and LaTeX: α^2

28. Explain Elastic Net.

L1 is good for sparsity, when there are many inputs and you believe that only a few of them are meaningful. L2 is good at dealing with correlated inputs. If two inputs are perfectly correlated L2 put half of the weight (beta) on each input, while L1 would pick one randomly (so less stable). One can use a combination of L1, L2 to get a balance of both, also known as Elastic Net.

Elastic net regularization method includes both LASSO (L1) and Ridge (L2) regularization methods.

Overfitting : The core idea behind machine learning algorithms is to build models that can find the generalised trends within the data. However, if no measures are taken, sometimes the models tend to rote learn the data instead of learning the patterns. During such cases, although the model fits well to the training data (model yields accurate results when evaluated on training data), however, it evaluated poorly on the test data. This is called overfit.

Regularization is used to prevent overfitting the model to training data. This is achieved by slightly perturbing (adding noise) the objective function of the model before optimizing it (optimising a model means to find the model parameters **w*** such that the argmin /argmax of the objective function is found- in other words, it is to find the global optima of the objective function) . In **L1 Regularisation,** a noise of magnitude *lambda* **.|w*|** is added while in **L2 Regularisation,** noise of magnitude *lambda*.**|w*|. |w*|** is added. where **|w*|** is the magnitude of the optimal parameter vector.

In **Elastic Net Regularization,** a linear sum of both noises are added. Hence, the objective function would then be

$$argmin_{\omega^*}(f(\omega^*) + \lambda_1|\omega^*| + \lambda_2|\omega^*|^2)$$

Note that L1 and L2 regularizations are special cases of Elastic Net regularization.

Practically, I think the biggest reasons for regularization are 1) to avoid overfitting by not generating high coefficients for predictors that are sparse. 2) to stabilize the estimates especially when there's collinearity in the data.

1) is inherent in the regularization framework. Since there are two forces pulling each other in the objective function, if there's no meaningful loss reduction, the increased penalty from the regularization term wouldn't improve the overall objective function. This is a great property since a lot of noise would be automatically filtered out from the model.

To give you an example for 2), if you have two predictors that have same values, if you just run a regression algorithm on it since the data matrix is singular, your beta coefficients will be Inf if you try to do a straight matrix inversion. But if you add a very small regularization lambda to it, you will get stable beta coefficients with the coefficient values evenly divided between the equivalent two variables.

For the difference between L1 and L2, the following graph demonstrates why people bother to have L1 since L2 has such an elegant analytical solution and is so computationally straightforward. Regularized regression can also be represented as a constrained regression problem (since they are Lagrangian equivalent). In Graph (a), the black square represents the feasible region of of the L1 regularization while graph (b) represents the feasible region for L2 regularization. The contours in the plots represent different loss values (for the unconstrained regression model). The feasible point that minimizes the loss is more likely to happen on the coordinates on graph (a) than on graph (b) since graph (a) is more **angular.** This effect amplifies when your number of coefficients increases, i.e. from 2 to 200.

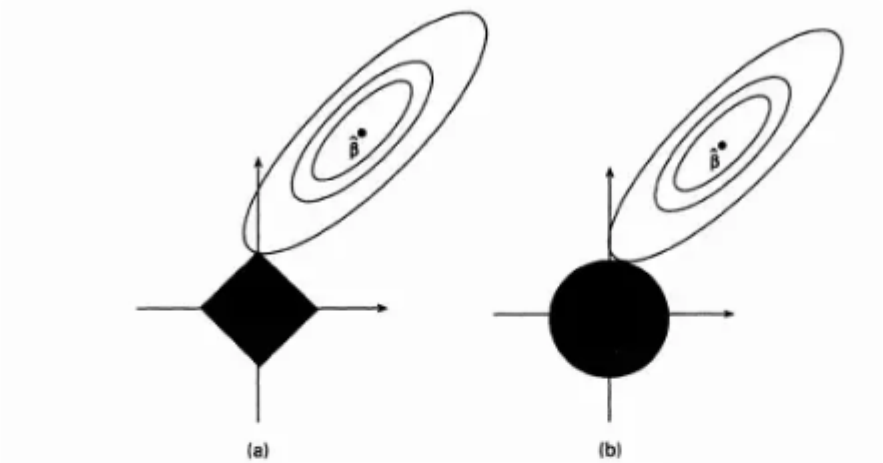


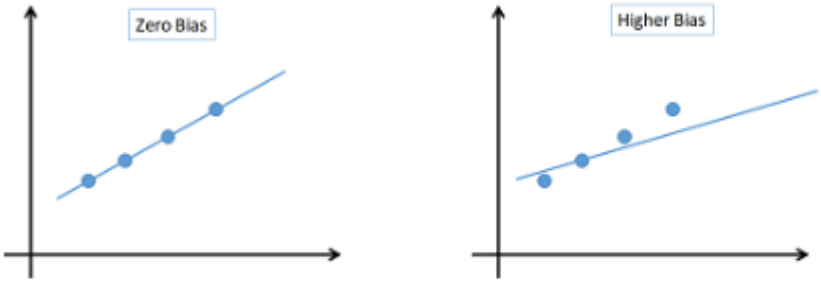
Fig. 2. Estimation picture for (a) the lasso and (b) ridge regression

The implication of this is that the L1 regularization gives you **sparse** estimates. Namely, in a high dimensional space, you got mostly zeros and a small number of non-zero coefficients. This is huge since it incorporates variable selection to the modeling problem. In addition, if you have to score a large sample with your model, you can have a lot of computational savings since you don't have to compute features(predictors) whose coefficient is 0. I personally think L1 regularization is one of the most beautiful things in machine learning and convex optimization. It is indeed widely used in bioinformatics and large scale machine learning for companies like Facebook, Yahoo, Google and Microsoft.

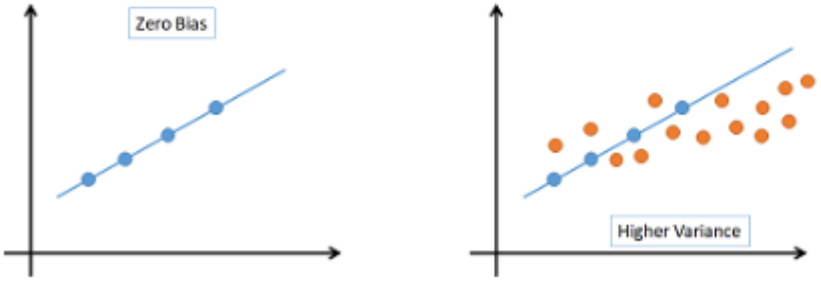
==========

What is Bias and Variance in the context of Regularization?

Have a look at this image below.

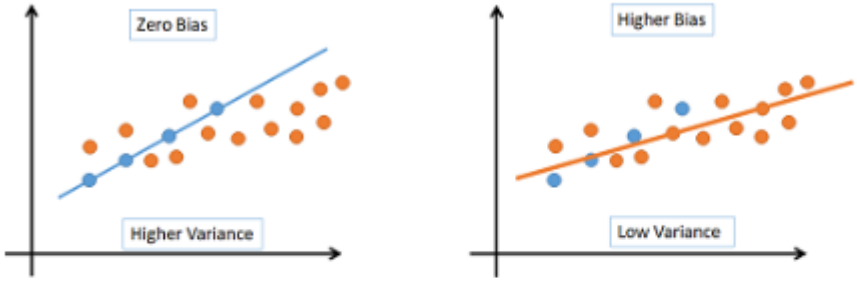


If these four points are your training data and your linear regression model fits it perfectly, we can say that we have zero bias for our regression model. Whereas the second plot has the higher bias. But this is just the training data. Look at the following plot where we have the test data or the data that our model has not seen before.



In this case, our model will have high prediction errors and is considered as the one with high variance of predictions.

To summarise Bias and Variance in this context, look at the following plot.

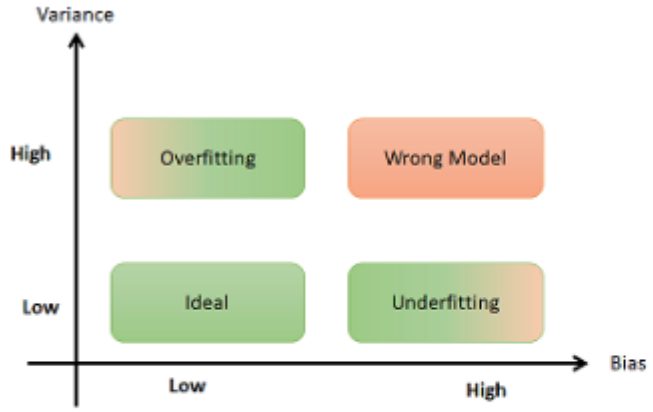


In the first plot, we have zero bias as our model perfectly fits the data during training, but has high variance while predicting the new data.

Whereas the second model, though appears to have higher bias and less accurate during training, it performs pretty well while predicting the unseen data.

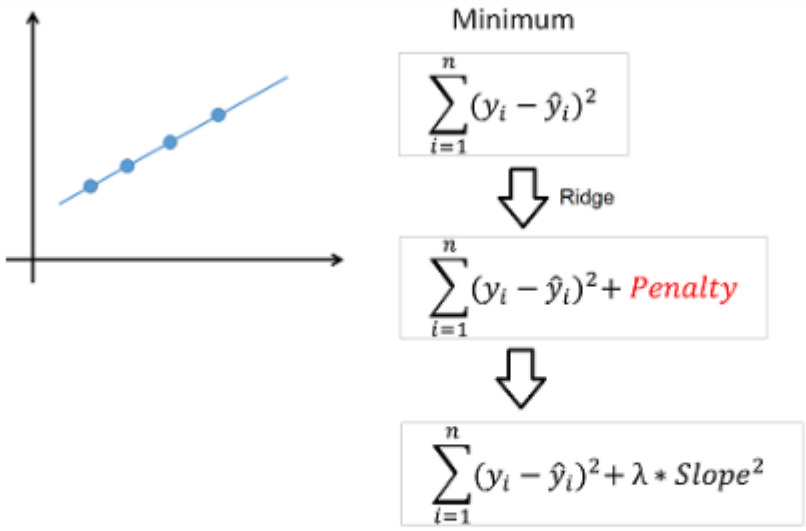
This trade-off between bias and variance during training to get better results for unseen predictions, is nothing but the **Bias-Variance Tradeoff**. :)

When we have very low bias during training and high variance during testing, we say that our model has Over fitting.

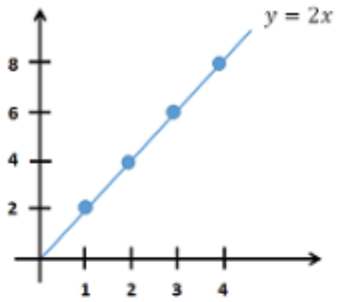


What is Ridge Regression?

Ridge Regression helps us avoid the over fitting by adding the bias in the form of the penalty. This penalty is added to the Cost/Loss/Error function of the regression which as we know is given by the Ordinary Least Squares or OLS or sum of the squared errors of predictions. Ridge adds the penalty as follows,



Now because of this the earlier line of best fit is no more the same. The penalty increases the value of the loss function. As can be seen from the example below, the earlier equation of the line using OLS was y =2*x and had an OLS value as 0. But now with the Ridge penalty and penalty parameter as 1 for lambda, the value of loss function has increased to 4.



$Slope = 2 \quad \lambda = 1$

OLS

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$0$$

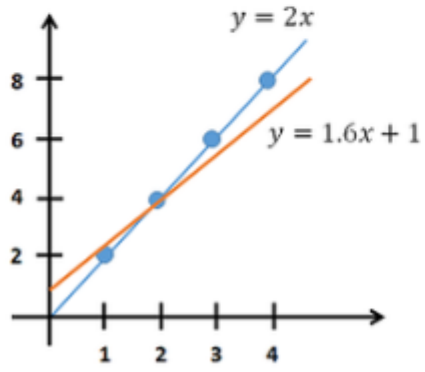
Ridge

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda * Slope^2$$

$$0 + 1 * 2^2$$

$$4$$

Now our objective is to minimise this new loss function and get a model that gives us the loss value less than 4. Have a look at the diagram below. The new line has indeed got a loss value less than 4.



$Slope = 1.6 \quad \lambda = 1$

$Intercept = 1$

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda * Slope^2$$

x	y	$\hat{y}$	$(y - \hat{y})^2$
1	2	2.6	0.36
2	4	4.2	0.04
3	6	5.8	0.04
4	8	7.4	0.36
Sum of Squared Differences			**0.80**

$$Penalty = \lambda * Slope^2 = 1 * 1.6^2 = 2.56$$

$$0.8 + 2.56$$

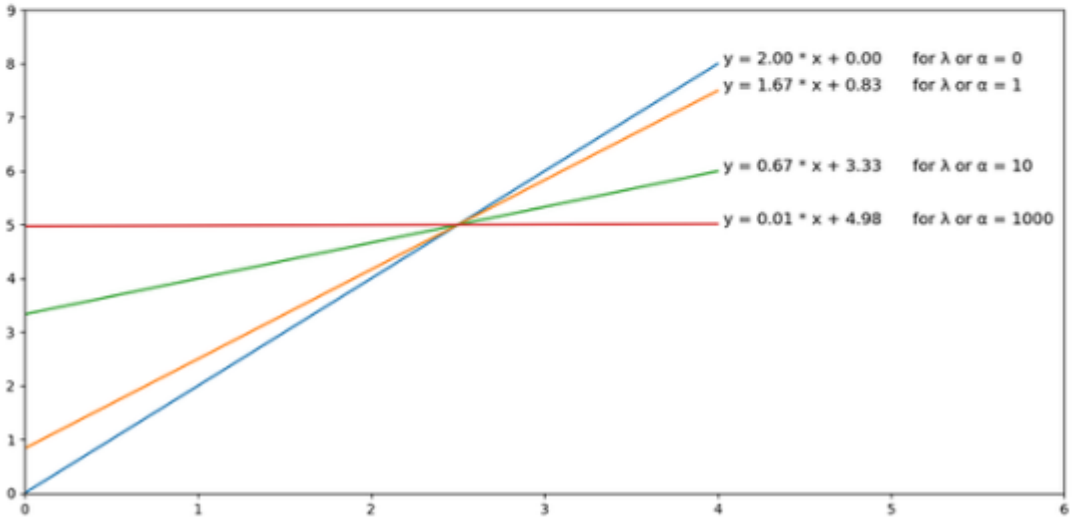
$$3.36 < 4$$

Hence the new best fit line is

y = 1.6 * x + 1 instead of y = 2 * x

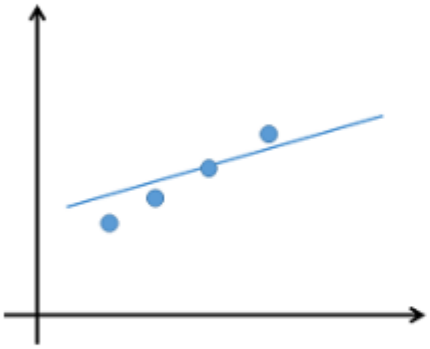
But because of this the line is now tilted or has the additional bias. Also, the weight or coefficient of X is now reduced. That also means there is less dependency or weight for independent variable X.

So as the value of lambda or Alpha, as it is called in sklearn of Python, increases, the value of the coefficient or weight of X decreases. You can see the evidence of it from the following plot I created using matplotlib.



What is Lasso Regression?

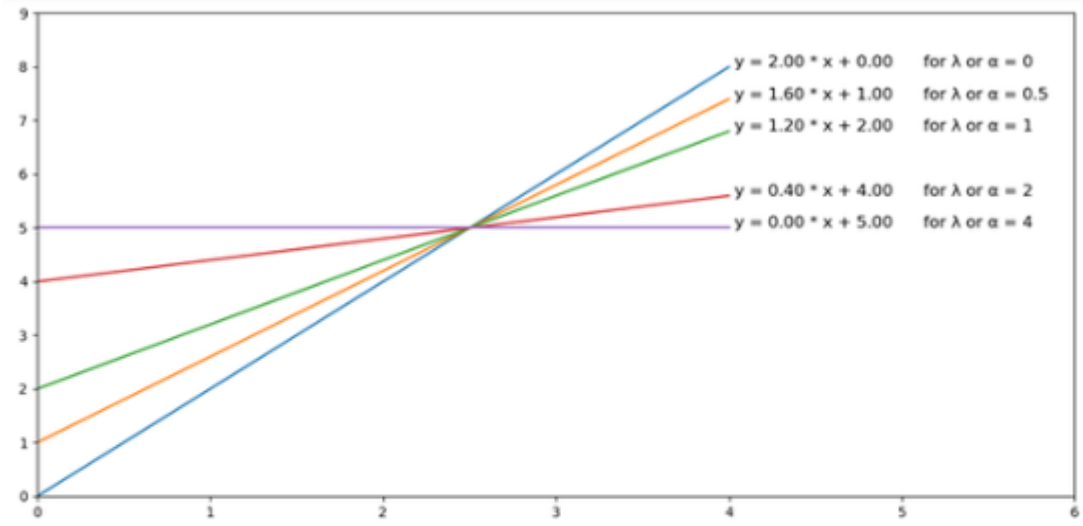
Well, Lasso is exactly same as Ridge, in the sense that it also adds penalty. But instead of the squared slope/coefficient/weight, it adds the absolute value of the slope/weight as the penalty to OLS loss function.



Minimum

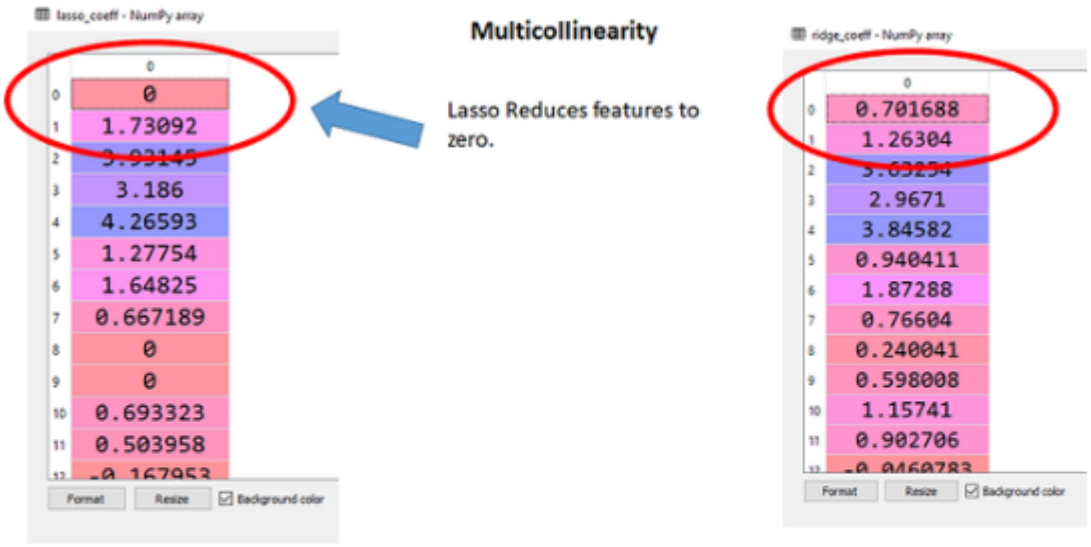
$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda * |Slope|$$

However, because of this and some of the explanations that others have given regarding the L1-Norm of the vector which results in a diamond shape, Lasso can reduce the features very quickly. See the plot below for various values of penalty parameter Lambda or Alpha. See how quickly Lasso made the coefficient of X to Zero whereas Ridge could reduce it to near zero with large values of Alpha. But Ridge was unable to make it zero even with Lambda as 100 or even 1000.



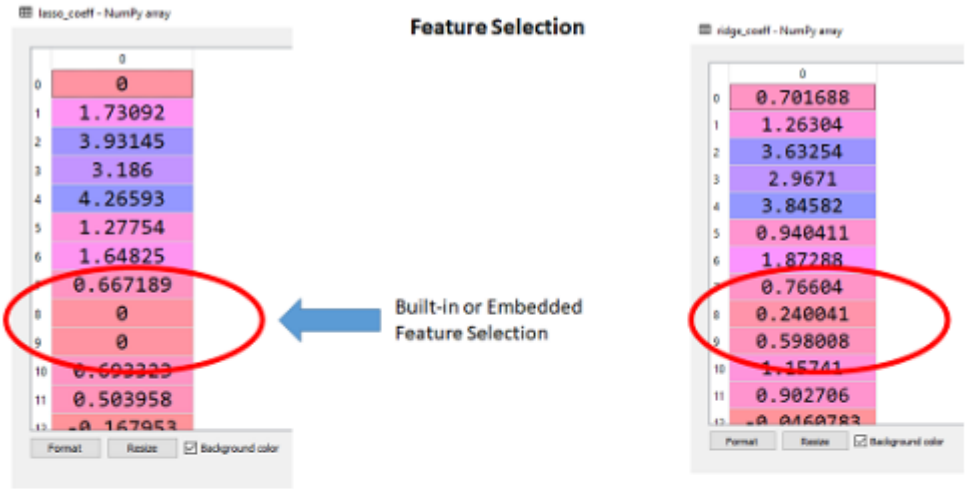
Effect of Lasso and Ridge on Multicollinearity and Feature Selection

Lastly, I ran an experiment that had 15 features. There was multicollinearity present for two of these features, while some of the features had very less impact on the dependent variable Y. See diagram below for Multicollinearity.



As you can see, Lasso made one of the features with multicollinearity to Zero while Ridge could only reduce its value. So another important point to note is Lasso can help us get rid of Multicollinearity from the input data.

Similarly, Lasso has reduced the features 8,9 to zero. That means, they are not part of the linear regression equation to predict the value of Y. This basically leads to feature selection using embedded feature selection by feature removal. See the diagram below,



Hope that answers all the questions with evidence.

However, I would like to add one more Regularization method of Elastic Net which is a combination of the two. It combines both the penalties as follows,

Ridge $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda * Slope^2$ Lasso $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda * |Slope|$

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda_1 * Slope^2 + \lambda_2 * |Slope|$$

29. Why do we do a train test split?

The Purpose of Train/Test Sets

Why do we use train and test sets?

More training data = better model

More testing data = better accuracy on testing results

Creating a train and test split of your dataset is one method to quickly evaluate the performance of an algorithm on your problem.

The training dataset is used to prepare a model, to train it.

We pretend the test dataset is new data where the output values are withheld from the algorithm.

We gather predictions from the trained model on the inputs from the test dataset and compare them to the withheld output values of the test set.

Comparing the predictions and withheld outputs on the test dataset allows us to compute a performance measure for the model on the test dataset. This is an estimate of the skill of the algorithm trained on the problem when making predictions on unseen data.

Training set is to build your model

Cross validation is to fine tune the model parameters on the unknown instances.

Test set is to check for accuracy for the model.

The simplest method that we can use to evaluate the performance of a machine learning algorithm is to use different training and testing datasets.

We can take our original dataset, split it into two parts. Train the algorithm on the first part, make predictions on the second part and evaluate the predictions against the expected results.

The size of the split can depend on the size and specifics of your dataset, although it is common to use 67% of the data for training and the remaining 33% for testing.

This algorithm evaluation technique is very fast. It is ideal for large datasets (millions of records) where there is strong evidence that both splits of the data are representative of the underlying problem. Because of the speed, it is useful to use this approach when the algorithm you are investigating is slow to train.

A downside of this technique is that it can have a high variance. This means that differences in the training and test dataset can result in meaningful differences in the estimate of accuracy.

30. What is polynomial regression? When to use it?

The goal of polynomial regression is to model a non-linear relationship between the independent

and dependent variables (technically, between the independent variable and the conditional mean of the dependent variable).

Polynomial Regression

For understanding Polynomial Regression, let's first understand a polynomial. Merriam-webster defines a polynomial as: "*A mathematical expression of one or more algebraic terms each of which consists of a constant multiplied by one or more variables raised to a non-negative integral power (such as a + bx + cx^2)*". Simply said, poly means many. So, a polynomial is an aggregation of many monomials(or Variables). A simple polynomial equation can be written as:

$$y = a + bx + cx^2 + \ldots + nx^n + \ldots$$

So, Polynomial Regression can be defined as a mechanism to predict a *dependent variable* based on the polynomial relationship with the *independent variable*.

In the equation,

$$y = a + bx + cx^2 + \ldots + nx^n + \ldots$$

the maximum power of 'x' is called the degree of the polynomial equation. For example, if the degree is 1, the equation becomes

$$y = a + bx$$

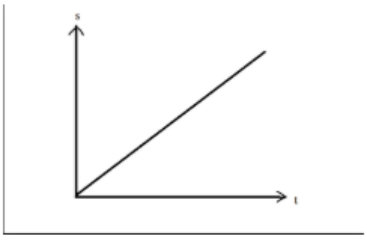
which is a simple linear equation. if the degree is 2, the equation becomes

$$y = a + bx + cx^2$$

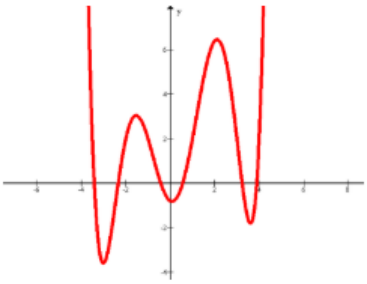
which is a quadratic equation and so on.

When to use Polynomial Regression?

Many times we may face a requirement where we have to do a regression, but when we plot a graph between a dependent and independent variables, the graph doesn't turn out to be a linear one. A linear graph typically looks like:



But what if the relationship looks like:



It means that the relationship between X and Y can't be described Linearly. Then comes the time to use the Polynomial Regression.

We can generalize the matrix obtained above (for Linear Regression) for an equation of n coefficients(in y=mx+b, m and b are the coefficients) as follows:

$$\begin{bmatrix} n & \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 & \cdots & \sum_{i=0}^{n} x_i^m \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 & \sum_{i=0}^{n} x_i^3 & \cdots & \sum_{i=0}^{n} x_i^{(m+1)} \\ \sum_{i=0}^{n} x_i^2 & \sum_{i=0}^{n} x_i^3 & \sum_{i=0}^{n} x_i^4 & \cdots & \sum_{i=0}^{n} x_i^{(m+2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n} x_i^m & \sum_{i=0}^{n} x_i^{(m+1)} & \sum_{i=0}^{n} x_i^{(m+2)} & \cdots & \sum_{i=0}^{n} x_i^{2m} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} y_i \\ \sum_{i=0}^{n} x_i y_i \\ \sum_{i=0}^{n} x_i^2 y_i \\ \vdots \\ \sum_{i=0}^{n} x_i^m y_i \end{bmatrix}$$

Where m is the _degree_(maximum power of x) of the polynomial and n is the number of observation points. The above matrix results in the general formula for Polynomial Regression. Earlier, we were able to visualize the calculation of minima because the graph was in three dimensions. But as there are n number of coefficients, it's not possible to create an (n+1) dimension graph here.

Polynomial Regression is a form of linear regression in which the relationship between the independent variable x and dependent variable y is modeled as an *nth* degree polynomial. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted E(y |x)

Why Polynomial Regression:

- There are some relationships that a researcher will hypothesize is curvilinear. Clearly, such type of cases will include a polynomial term.
- Inspection of residuals. If we try to fit a linear model to curved data, a scatter plot of residuals (Y axis) on the predictor (X axis) will have patches of many positive residuals in the middle. Hence in such situation it is not appropriate.
- An assumption in usual multiple linear regression analysis is that all the independent variables are independent. In polynomial regression model, this assumption is not satisfied.

Uses of Polynomial Regression:
These are basically used to define or describe non-linear phenomenon such as:

- Growth rate of tissues.
- Progression of disease epidemics
- Distribution of carbon isotopes in lake sediments

The basic goal of regression analysis is to model the expected value of a dependent variable y in terms of the value of an independent variable x. In simple regression, we used following equation –

```
y = a + bx + e
```

Here y is dependent variable, a is y intercept, b is the slope and e is the error rate.

In many cases, this linear model will not work out For example if we analyzing the production of chemical synthesis in terms of temperature at which the synthesis take place in such cases we use quadratic model

```
y = a + b1x + b2^2 + e
```

Here y is dependent variable on x, a is y intercept and e is the error rate.

In general, we can model it for nth value.

```
y = a + b1x + b2x^2 +....+ bnx^n
```

Since regression function is linear in terms of unknown variables, hence these models are linear from the point of estimation.

Hence through Least Square technique, let's compute the response value that is y.

What is so important about Polynomial regression?

Polynomial regressions are often the most difficult regressions.

This is niche skill set and is extremely rare to find people with in-depth knowledge of the creation of these regressions

So what exactly is Polynomial regression is all about?

In this regression, the relationship between **dependent** and the **independent variable** is modeled such that the dependent variable Y is an nth degree function of independent variable Y.

The polynomial regression fits into a non-linear relationship between the value of X and the value of Y.

The Polynomial regression is also called as multiple linear regression models.

The Polynomial regression model has been an important source for the development of regression analysis.

Advantages of using Polynomial Regression:

- Broad range of function can be fit under it.
- Polynomial basically fits wide range of curvature.
- Polynomial provides the best approximation of the relationship between dependent and independent variable.

Disadvantages of using Polynomial Regression

- These are too sensitive to the outliers.
- The presence of one or two outliers in the data can seriously affect the results of a nonlinear analysis.
- In addition there are unfortunately fewer model validation tools for the detection of outliers in nonlinear regression than there are for linear regression.

So when was Polynomial regression got into existence?

It is modeled based on the method of least squares on condition of Gauss Markov theorem. The method was published in 1805 by Legendre and 1809 by Gauss. The first **Polynomial regression** model came into being in1815 when Gergonne presented it in one of his papers. It is a very common method in scientific study and research.

Let us example Polynomial regression model with the help of an example:

Formula and Example:

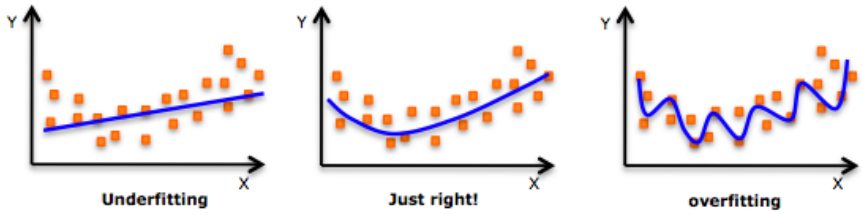
The formula, in this case, is modeled as -

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n + \varepsilon.$$

Where y is the dependent variable and the betas are the coefficient for different nth powers of the independent variable x starting from 0 to n. The calculation is often done in a **matrix** form as shown below -

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ 1 & x_3 & x_3^2 & \dots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix},$$

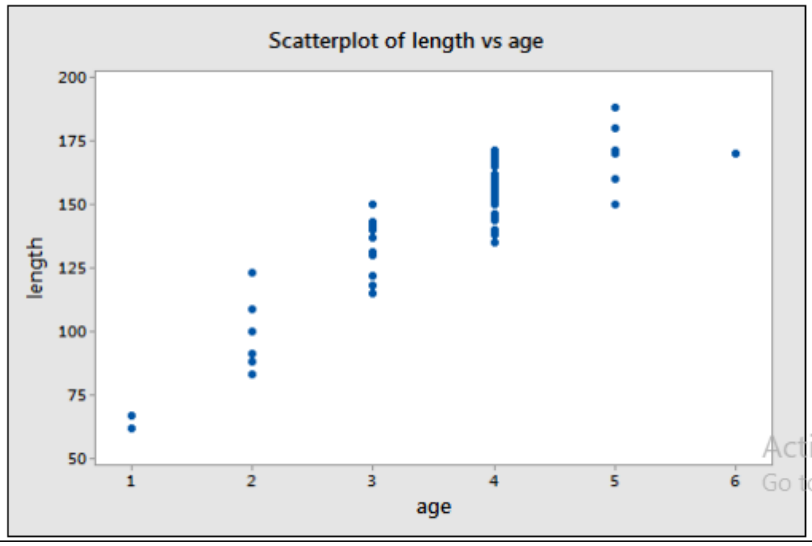
This is due to the high amount of data and correlation among each data type. The **matrix** is always invertible as they follow the statistical rule of m < n and thus become Vandermonde matrix. While it might be tempting to fit the curve and decrease error, it is often required to analyze whether fitting all the points makes sense logically and avoid overfitting. This is a **highly important step** as **Polynomial Regression** despite all its benefit is still only a statistical tool and requires human logic and intelligence to decide on right and wrong. Thus, while analytics and regression are great tools to help make decision-making, they are not complete decision makers. An example for overfitting may be seen below -



It is also advised to keep the order of the polynomial as low as possible to avoid unnecessary complexities. There are two ways of doing a **Polynomial regression** one is forward selection procedure where we keep on increasing the degree of polynomial till the t-test for the highest order is insignificant. The other process is called backward selection procedure where the highest order polynomial is deleted till the t-test for the higher order polynomial is significant.

An example might be an impact of the increase in temperature on the process of chemical synthesis. Such process is often used by chemical scientists to determine optimum temperature for the chemical synthesis to come into being. Another example might be the relation between the lengths of a bluegill fish compared to its age. Where dependent variable is Y in mm and the dependent variable is X in years.

The marine biologists were primarily interested in knowing how the bluegill fish grows with age and were wanting to determine a correlation between them. The data was collected in the scatter plot given bellow -



A polynomial regression is an application of multiple linear regression - the distinction is only the names. For example, whether you write

$$y = \beta_0 + \beta_1 x + \beta_2 x^2$$

or

$$y = \beta_0 + \beta_1 x + \beta_2 z$$

where $z = x^2$ you have the same model. The 'linear' in 'linear regression' means the function you estimate is linear **in the coefficients**.

Polynomial regression is stating the fact that there is polynomial relationship between predictors and response variable regardless of the number of features you have .

Multiple linear regression is a case of linear regression where the the relationship between response and predictor variables in linear, for example plane, line etc.

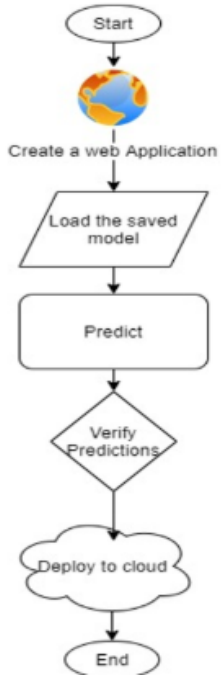
Polynomial relationship is not the same as linear relationship.

when you add degrees to your features in your model, you essentially turn a linear model into a spline surface or curve which then is said to be polynomial in nature.

31. Explain the steps for GCP deployment.

Cloud Deployment (Google Cloud Platform)

Once the training is completed, we need to expose the trained model as an API for the user to consume it. For prediction, the saved model is loaded first and then the predictions are made using it. If the web app works fine, the same app is deployed to the cloud platform. The application flow for cloud deployment looks like:



32. What difficulties did you face in cloud deployment?

Create a file ‘app.yaml’ and put ‘runtime: python37’ in that file.

Create a ‘requirements.txt’ file by opening the command prompt/anaconda prompt, navigate to the project folder and enter the command ‘pip freeze > requirements.txt’. It is recommended to use separate environments for different projects.

Your python application file should be called ‘main.py’. It is a GCP specific requirement.

Open command prompt window, navigate to the project folder and enter the command gcloud init to initialise the gcloud context.

In [ ]: