********** ASSIGNMENT 14 *********

1. **What is Data Science?List the differences between supervised and unsupervised learning.**

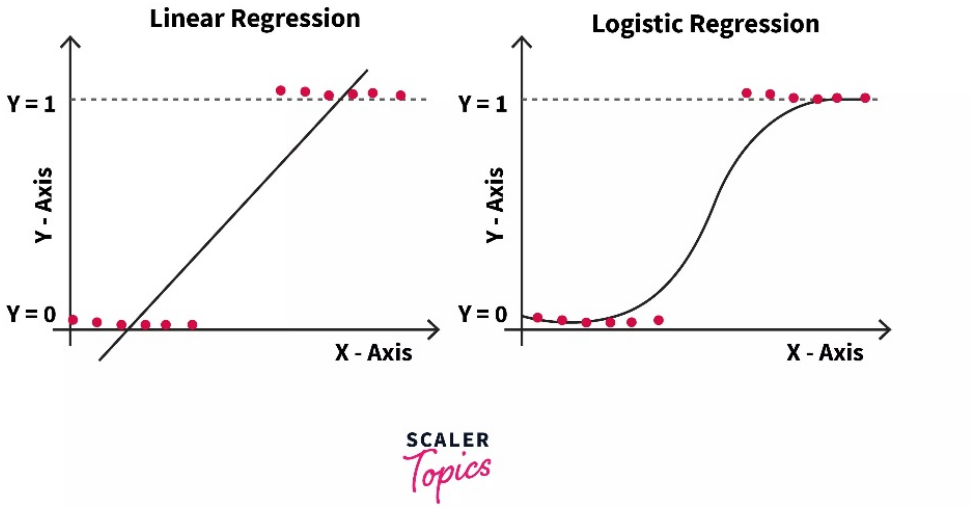
   Data science is the process of extracting insights from data by using scientific methods, processes, algorithms and systems.

Unsupervised Learning	Supervised Learning
The machine is given huge sets of data that are not labelled as inputs to analyse.	The input is in the form of raw data that is labelled.
The machine needs to figure out the output on its own by identifying patterns in the raw data provided to it.	The machine is already fed with the required feature set to classify between inputs (hence the term 'supervised').
Divided into two types of problems – Association (where we want to find a set of rules that describe our data) and Clustering (where we want to find groups in our data).	Divided into two types of problems – Regression (outputs are real values) and Classification (outputs are categories).
K-means for clustering problems and Apriori algorithm for association rule learning problems.	Linear regression for regression problems, Random Forest for classification and regression problems, Support Vector Machines for classification problems.

2. **What is logistic regression?**

   - It is supervised machine learning classification algorithm which is used to predict probabilities of target variable.
   - It is probabilistic ML algorithm.
   - The output is a categorical value meaning simply a direct or discrete value. These might include True/ False, Yes/No, or 0/1. But since, as I said, the Logistic Regression Algorithm is based on Statistics so instead of 0 or 1, it gives a probabilistic answer that lies between 0 and 1.

- In the Logistic Regression Machine Learning, we will get a ‘S’ shaped logistic/sigmoid function . This function is responsible for predicting values between 0 and 1



Logistic regression can be classified as:

1. **binomial:** target variable can have only 2 possible types: “0” or “1” which may represent “win” vs “loss”, “pass” vs “fail”, “dead” vs “alive”, True vs False, Good vs Bad,etc.

2. **multinomial:** target variable can have 3 or more possible types which are not ordered(i.e. types have no quantitative significance) like “disease A” vs “disease B” vs “disease C”.

3. **ordinal:** it deals with target variables with ordered categories. For example, a test score can be categorized as:“very poor”, “poor”, “good”, “very good”. Here, each category can be given a score like 0, 1, 2, 3.

3. How will you deal with the multiclass classification problem using logistic regression?

The output that is given by a Logistic Regression unit is in the range 0 to 1.

We can think of the output to be the probability that it belongs to the positive class. So if output is higher than 0.5, then the example belongs to the positive class else it belongs to the negative class.

So a single logistic regression unit only supports binary classification

To extend logistic regression to multiple classes there are mainly two methods

1. **One vs All**
2. **One vs One**

1) One vs all

Suppose we have 4 classes. We train 4 independent logistic regression units one for each class as positive and other examples as negative.

Thus

For first model, positive class- 1, negative class-(2,3,4)

For second model, positive class-2, negative class-(1,3,4)

For third model, positive class-3, negative class-(1,2,4)

For Fourth model, positive class-4,negative class-(1,2,3)

We then choose the class based on the probability of each of the models. Thus if first model has highest probability then example belongs to the first class.

2) One vs One

In this method, we train a model for each pair of classes

So for 4 classes, we train 6 models

1)Positive class-1, negative class-2

2)Positive class-1, negative class-3

3)Positive class-1, negative class-4

4)Positive class-2, negative class-3

5)Positive class-2, negative class-4

6)Positive class-3, negative class-4

Now we count how many models gave output as 1,2,3 and 4

The one with the maximum count might be the class of the example.

Usually we prefer **One vs All** over **One vs One** since less number of models are to be trained.

Also Logistic Regression units are the building blocks of a Neural Network , so a combination of these logistic regression units support multi class classification.

4. Why is logistic regression very popular/widely used?

Another thing that makes logistic regression so popular is that it's easy to learn and implement. Its ease of use makes it a great learning tool for future data scientists and people in other tech-related college programs.

There are three main reasons for the same.

- Most of the business scenarios has binary dependent variable (like responded or not, defaulted or not, fraud transaction or not etc.)
- Logistic regression is very easy to deploy (because it gives you a simple equation)
- Additionally it gives much finer classification (like 0-1000, if you are convert generated scores in that range), not just few classes (like what classification tree gives). This is the reason why usually logistic regression has better good bad separation power than classification tree etc

5. Why can’t linear regression be used instead of logistic regression for classification?

The reasons why linear regressions cannot be used in case of binary classification are as follows:

Distribution of error terms: The distribution of data in the case of linear and logistic regression is different. Linear regression assumes that error

terms are normally distributed. In the case of binary classification, this assumption does not hold true.

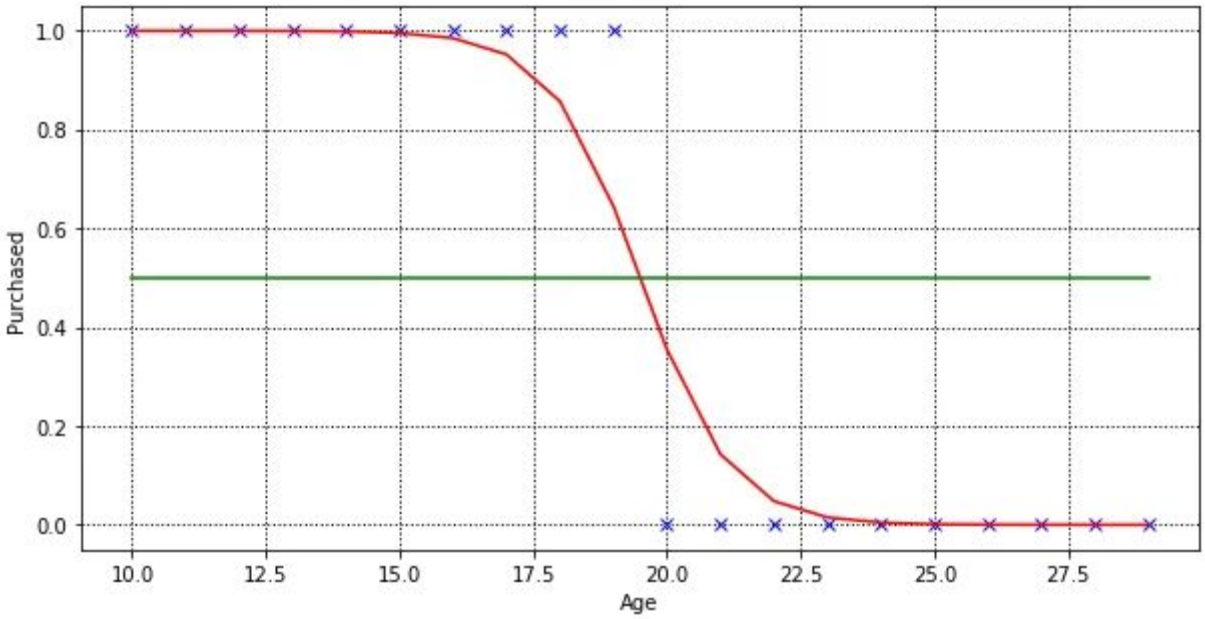
Model output: In linear regression, the output is continuous. In the case of binary classification, an output of a continuous value does not make sense. For binary classification problems, linear regression may predict values that can go beyond 0 and 1. If we want the output in the form of probabilities, which can be mapped to two different classes, then its range should be restricted to 0 and 1. As the logistic regression model can output probabilities with logistic/sigmoid function, it is preferred over linear regression.

Variance of Residual errors: Linear regression assumes that the variance of random errors is constant. This assumption is also violated in the case of logistic regression.

2 reasons why linear regression is not suitable:

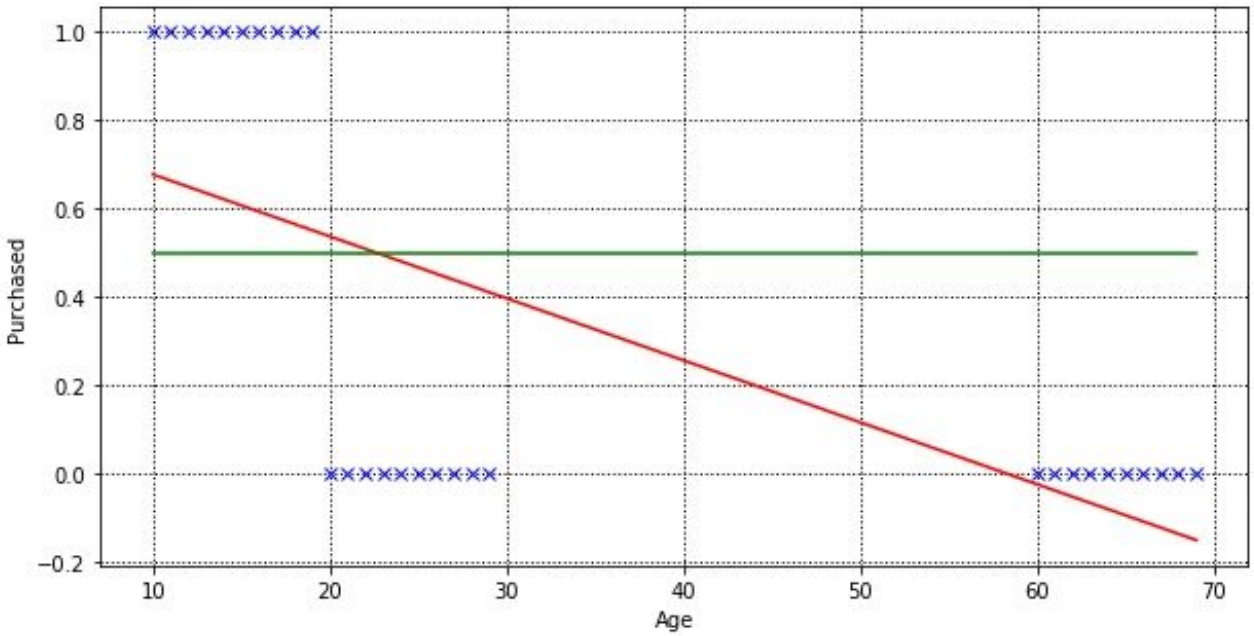
- the predicted value is continuous, not probabilistic
- sensitive to imbalance data when using linear regression for classification

- Problem #1: Predicted value is continuous, not probabilistic
- In a binary classification problem, what we are interested in is the probability of an outcome occurring. Probability is ranged between 0 and 1, where the probability of something certain to happen is 1, and 0 is something unlikely to happen. But in linear regression, we are predicting an absolute number, which can range outside 0 and 1.
- Using our linear regression model, anyone age 30 and greater than has a prediction of negative “purchased” value, which don’t really make sense. But sure, we can limit any value greater than 1 to be 1, and value lower than 0 to be 0. Linear regression can still work, right?
- Yes, it might work, but logistic regression is more suitable for classification task and we want to prove that logistic regression yields better results than linear regression. Let’s see how logistic regression classifies our dataset.



- Now we have 2 models trained on the same dataset, one by linear regression, and another by logistic regression. We can compare both models performance by using root mean squared error (RMSE) and the coefficient of determination (R^2 score).
- +--------------------+--------------------+----------------------+
- | | R2 (higher better) | RMSE (lower better) |
- +--------------------+--------------------+----------------------+
- | Linear regression | 0.7518796992481203 | 0.062030075187969935 |
- | Logistic regression | 0.9404089597242656 | 0.014897760068933596 |
- +--------------------+--------------------+----------------------+
- R^2 is a measure of how closely the observed data points are to the fitted regression line, generally the higher the better. But R^2 alone is not enough, so we look at RMSE as well. RMSE measure how far the observed data points are to the model's predicted values, the lower the better.
- From the metrics, logistic regression performed much better than linear regression in classification tasks. Like how Cassie Kozyrkov quotes it.
- Neural networks may as well be called "yoga networks" — their special power is giving you a very flexible boundary.

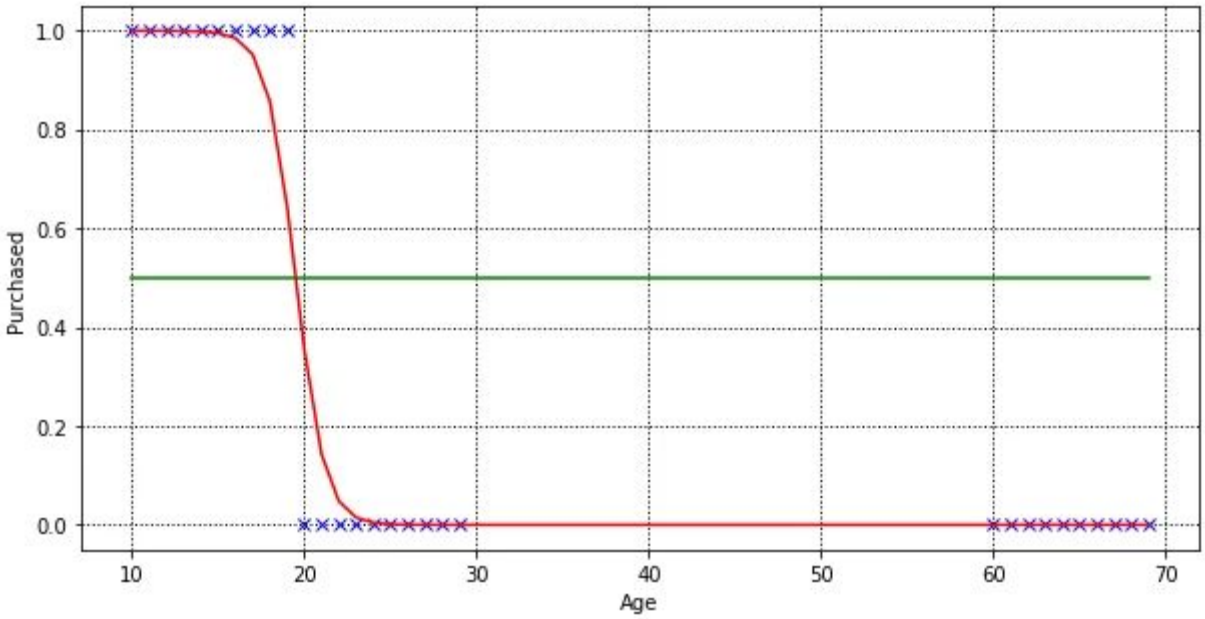
- Problem #2: Sensitive to imbalance data
- Let's add 10 more customers age between 60 to 70, and train our linear regression model, finding the best fit line.



- Our linear regression model manages to fit a new line, but if you look closer, some customers (age 20 to 22) outcome are predicted wrongly.

Age	Predicted Y Value
18	0.56495292
19	0.55091537
20	0.53687781
21	0.52284026
22	0.50880271
23	0.49476516
24	0.48072761
25	0.46669006

- As linear regression tries to fit the regression line by minimising prediction error, in order to minimise the distance of predicted and actual value for customers age between 60 to 70. Let's train a logistic regression model with the same dataset.



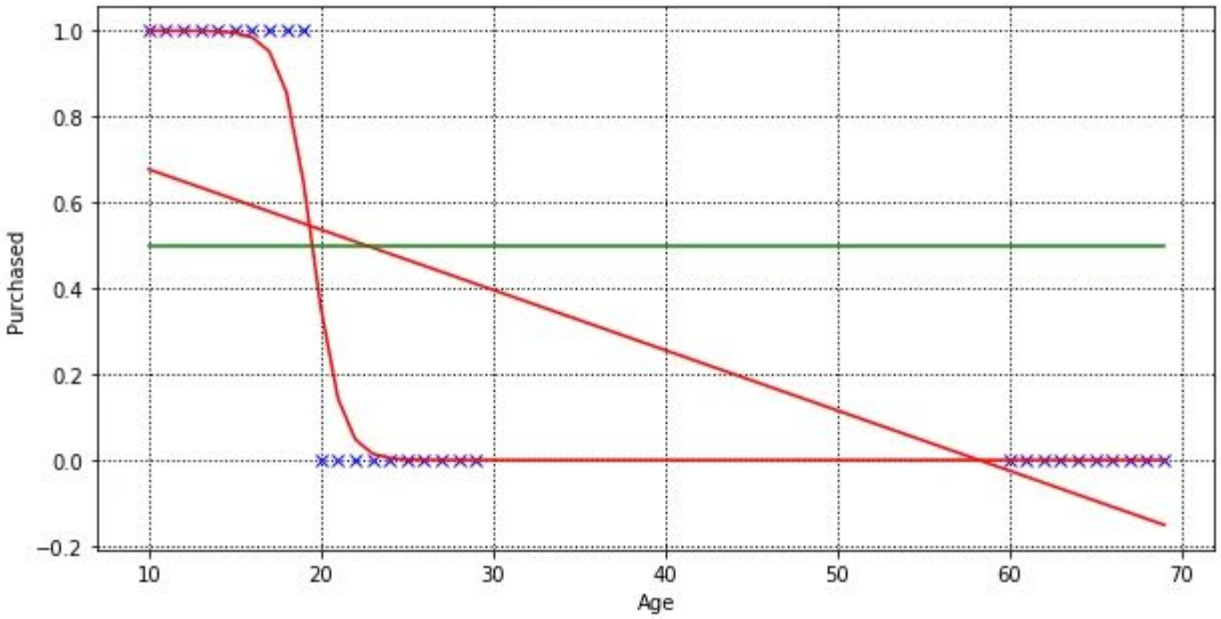
- Yes! In this very simple dataset, logistic regression manages to classify all data points perfectly.

Age	Predicted Y Value
18	0.85713668
19	0.64502441
20	0.35497751
21	0.14286435
22	0.04805457

- Let's compare the R^2 and RMSE again for both models, and you will see that logistic regression does a way better job than linear regression.

	R2 (higher better)	RMSE (lower better)
Linear regression	0.4211265134234073	0.12863855257257611
Logistic regression	0.9553066567250715	0.00993185406109522

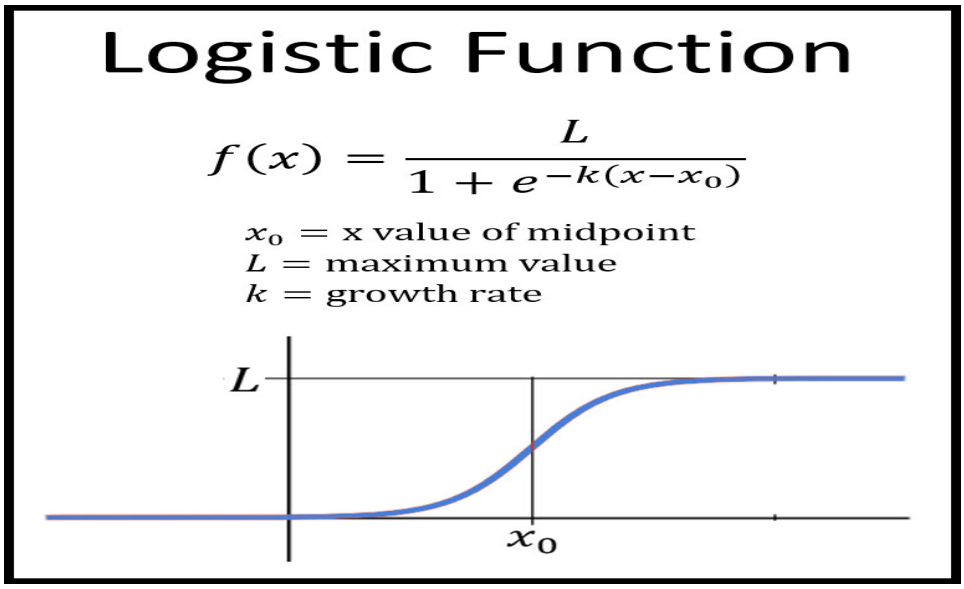
- Conclusion



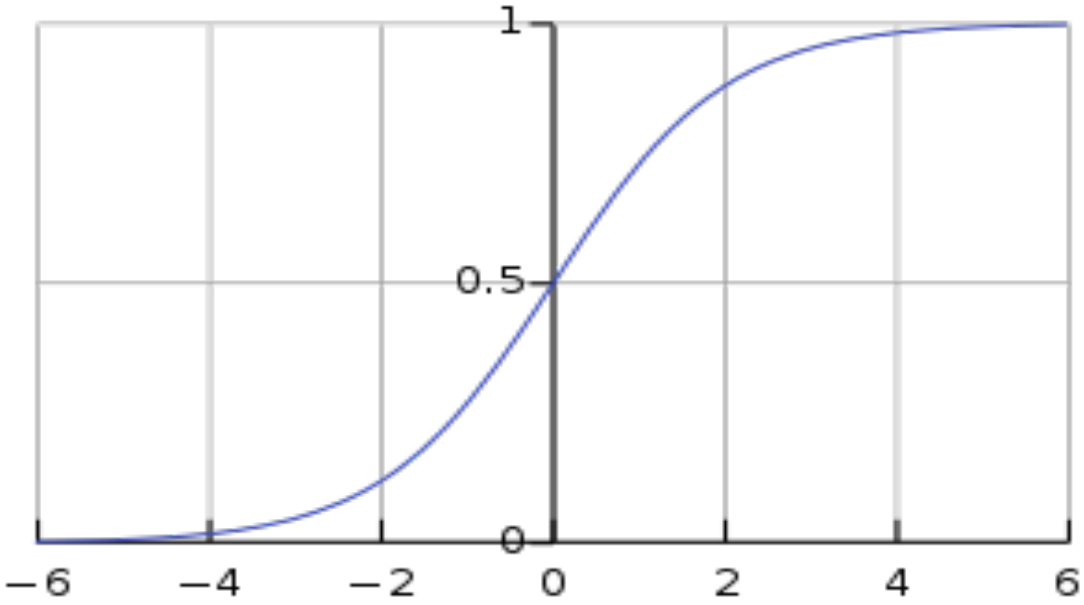
- Linear regression is suitable for predicting output that is continuous value, such as predicting the price of a property. Its prediction output can be any real number, range from negative infinity to infinity. The regression line is generally a straight line.
- Whereas logistic regression is for classification problems, which predicts a probability range between 0 to 1. For example, predict whether a customer will make a purchase or not. The regression line is a sigmoid curve.

6.What is the formula for the logistic regression function?

A logistic function or logistic curve is a common S-shaped curve (sigmoid curve) with equation.



The Logistic Function is sometimes a more realistic growth model than the typical exponential growth models used. Most populations do not grow exponentially without bound. Once the population has grown to reach its environment's maximum capacity, it will level off around the carrying capacity.



The logistic function finds applications in a range of fields,
including biology (especially ecology), biomathematics, chemistry, demo

graphy,economics, geoscience, mathematical,psychology, probability, so ciology, political science, linguistics, statistics, and artificial neural networks.
A generalization of the logistic function is the hyperbolastic function of type. The standard logistic function is sometimes simply called *the sigmoid*. It is also sometimes called the expit, being the inverse of the logit.

7. What are the assumptions of logistic regression?

Basic assumptions that must be met for logistic regression include independence of errors, linearity in the logit for continuous variables, absence of multicollinearity, and lack of strongly influential outliers.
Logistic regression is a highly effective modeling technique that has remained a mainstay in statistics since its development in the 1940s.

Assumption 1— Appropriate Outcome Type

Logistic regression generally works as a classifier, so the type of logistic regression utilized (binary, multinomial, or ordinal) must match the outcome (dependent) variable in the dataset.

By default, logistic regression assumes that the outcome variable is **binary**, where the number of outcomes is two (e.g., Yes/No).

If the dependent variable has three or more outcomes, then **multinomial or ordinal** logistic regression should be used.

How to Check?

We can check this assumption by getting the number of different outcomes in the dependent variable. If we want to use **binary** logistic regression, then there should only be **two** unique outcomes in the outcome variable.

Assumption 2 — Linearity of independent variables and log- odds

One of the critical assumptions of logistic regression is that the relationship between **logit** (**log-odds**) of the outcome and each **continuous** independent variable is *linear.*

The **logit** is the logarithm of the **odds ratio**, where ***p*** = probability of a positive outcome (e.g., survived Titanic sinking)

$$\text{logit}(p) = \log(\frac{p}{1-p})$$

How to Check?

(i) Box-Tidwell Test

The Box-Tidwell test is used to check for linearity between the predictors and the logit. This is done by add transformed interaction terms between the continuous independent variables and their corresponding natural log into the model.

For example, if one of your continuous independent variables is Age, then the interaction term to add as a new variable will be Age * ln**(**Age**).**

As part of the Box-Tidwell test, we filter our dataset to keep just the continuous independent variables.

Note: While R has the carlibrary to perform Box-Tidwell with a single line of code, I could not find any Python package that can do something similar.

If you have more than one continuous variable, you should include the same number of interaction terms in the model.

Generalized Linear Model Regression Results

Dep. Variable:	Survived	No. Observations:	876
Model:	GLM	Df Residuals:	871
Model Family:	Binomial	Df Model:	4
Link Function:	logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-536.19
Date:	Mon, 04 Oct 2021	Deviance:	1072.4
Time:	11:42:53	Pearson chi2:	881.
No. Iterations:	4		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Age	-0.1123	0.058	-1.948	0.051	-0.225	0.001
Fare	0.0785	0.013	6.057	0.000	0.053	0.104
Age:Log_Age	0.0218	0.013	1.640	0.101	-0.004	0.048
Fare:Log_Fare	-0.0119	0.002	-5.251	0.000	-0.016	-0.007
const	-0.3764	0.402	-0.937	0.349	-1.164	0.411

to the logit of the outcome variable and that the assumption is What we need to do is check the statistical significance of the interaction terms (Age: Log_Age and *Fare:* Log_Fare in this case) based on their p-values**.**

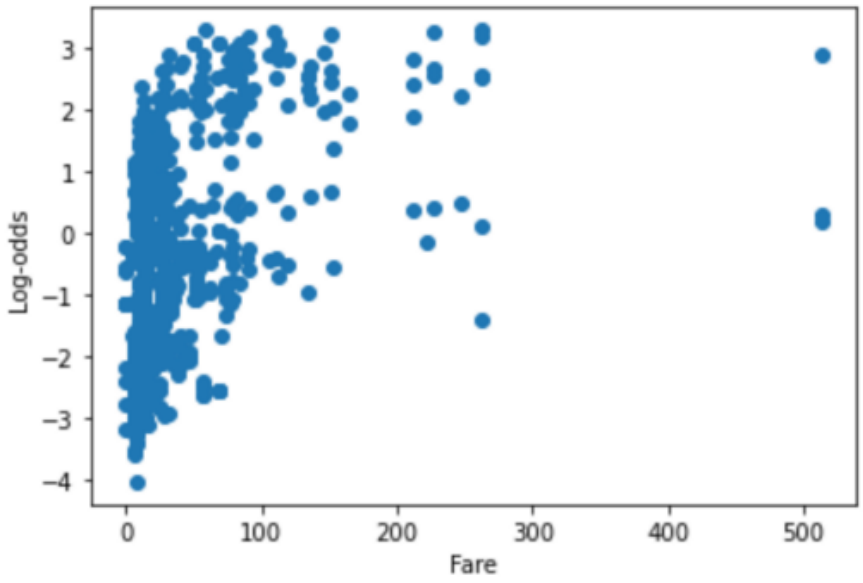
The Age:Log_Age interaction term has a p-value of 0.101 (not statistically significant since p>0.05), implying that the independent variable *Age* is linearly related satisfied.

On the contrary, Fare:Log Fare is statistically significant (i.e., p≤0.05), indicating the presence of non-linearity between *Fare* and the logit.

One solution is to perform transformations by incorporating higher-order polynomial terms to capture the non-linearity (e.g., *Fare²).*

(ii) Visual check

Another way that we can check logit linearity is by visually inspecting the scatter plot between each predictor and the logit values.



The above scatter plot shows a clear non-linear pattern of *Fare* vs. the log-odds, thereby implying that the assumption of logit linearity is **violated**.

Assumption 3— No strongly influential outliers

Logistic regression assumes that there are **no** highly influential outlier data points, as they distort the outcome and accuracy of the model.

Note that not all outliers are influential observations. Rather, outliers have the *potential* to be influential. To assess this assumption, we need to check whether both criteria are satisfied, i.e., influential **and** outlier.

How to Check?

(i) Influence

We can use Cook's Distance to determine the **influence** of a data point, and it is calculated based on its residual and leverage. It summarizes the changes in the regression model when that particular observation is removed.

There are different opinions regarding what cut-off values to use. One standard threshold is **4/N** (where N = number of observations), meaning that observations with **Cook's Distance > 4/N** are deemed as influential.

(ii) Outliers

We use **standardized residuals** to determine whether a data point is an outlier or not. Data points with **absolute standardized residual values greater than 3** represent possible extreme outliers.

(iii) Putting Both Together

We can identify the strongly **influential outlier** data points by finding the top observations based on thresholds defined earlier for Cook's Distance and standardized residuals.

When outliers are detected, they should be treated accordingly, such as removing or transforming them.

Assumption 4 — Absence of Multicollinearity

Multicollinearity corresponds to a situation where the data contain highly correlated independent variables. This is a problem because it **reduces the precision of the estimated coefficients**, which weakens the statistical power of the logistic regression model.

How to Check?

Variance Inflation Factor (VIF) measures the degree of multicollinearity in a set of independent variables.

Mathematically, it is **equal to the ratio of the overall model variance to the variance of** a model that includes only that single independent variable.

The smallest possible value for VIF is 1 (i.e., a complete absence of collinearity). As a rule of thumb, a **VIF value that exceeds 5 or 10** indicates a problematic amount of multicollinearity.

Assumption 5— Independence of observations

The observations must be **independent** of each other, i.e., they should not come from repeated or paired data. This means that each observation is not influenced by or related to the rest of the observations.

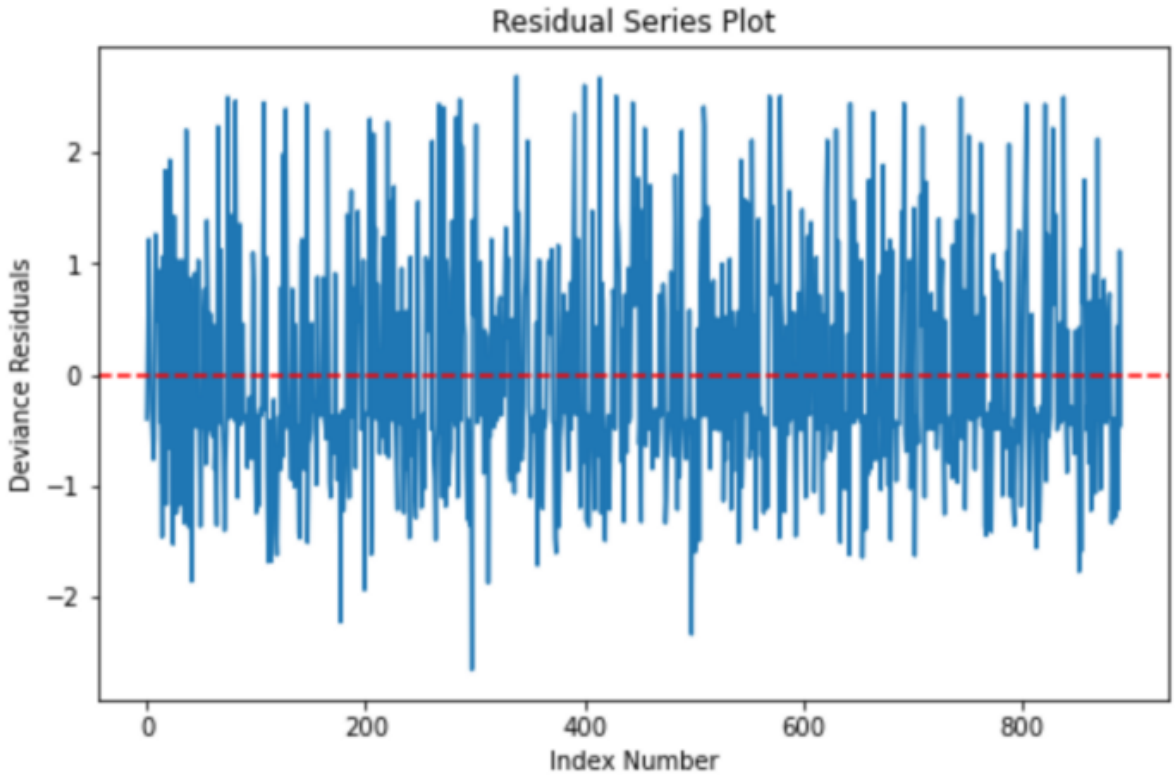
How to Check?

This independence assumption is **automatically** met for our Titanic example dataset since the data consists of individual passenger records.

This assumption would be more of a concern when dealing with time-series data, where the correlation between sequential observations (*auto- correlation*) can be an issue.

Nonetheless, there are still ways to check for the independence of observations for non-time series data. In such cases, the 'time variable' is the order of observations (i.e., index numbers).

In particular, we can create the **Residual Series plot** where we plot the deviance residuals of the logit model against the index numbers of the observations.



Since the residuals in the plot above appear to be randomly scattered around the centerline of **zero**, we can infer (visually) that the assumption is satisfied.

Assumption 6 — Sufficiently large sample size

There should be an adequate number of observations for each independent variable in the dataset to avoid creating an overfit model.

How to Check?

Like Cook's distance, there are numerous opinions on the rule of thumb to determine a 'sufficiently large' quantity.

One rule of thumb is that there should be at least **10 observations with the least frequent outcome** for each independent variable. We can check this by retrieving the *value counts* for each variable.

EXTRA: Comparison with Linear Regression

Although the assumptions for logistic regression differ from linear regression, several assumptions still hold for both techniques.

Differences

- Logistic regression does **not** require a linear relationship between the dependent and independent variables. However, it still needs independent variables to be linearly related to the **log-odds of the outcome.**
- **Homoscedasticity** (constant variance) is required in linear regression but not for logistic regression.
- The error terms (**residuals**) must be **normally** distributed for linear r egression but not required in logistic regression.

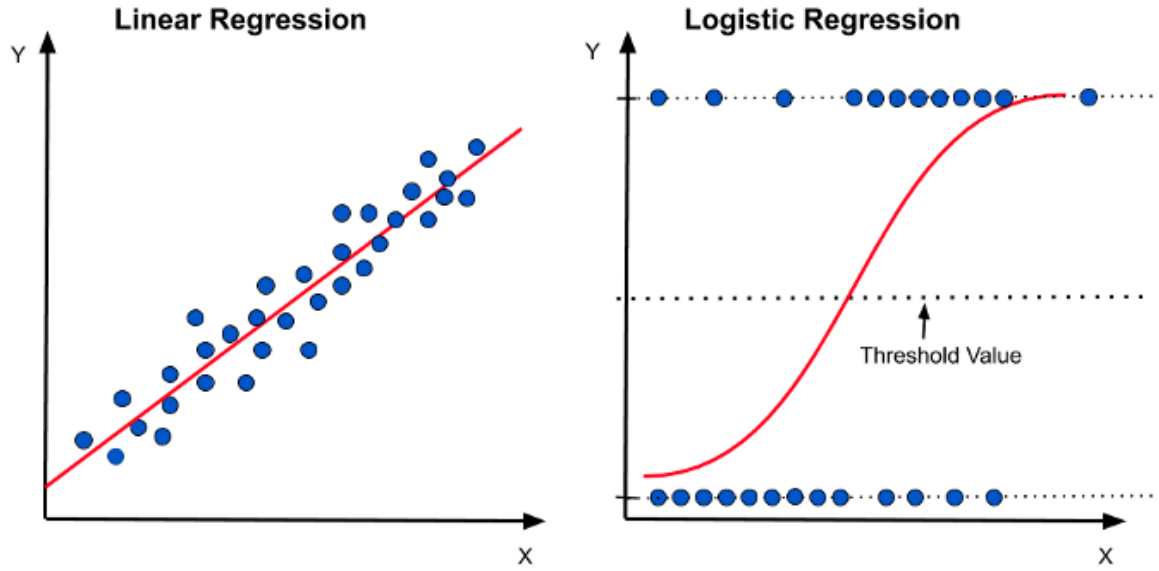
Similarities

- Absence of multicollinearity
- Observations are independent of each other.

8. Why is logistic regression called regression and not classification?

Logistic Regression is one of the basic and popular algorithms to solve a classification problem. It is named 'Logistic Regression' **because its underlying technique is quite the same as Linear Regression**. The term “Logistic” is taken from the Logit function that is used in this method of classification.

There is a strong relationship between linear regression and logistic regression.



Logistic regression is a generalized linear model. And it uses the same basic formula of linear regression.

In linear regression, we predict the output variable Y base on the weighted sum of input variables.

The formula is as follows:

$$\mathbf{Y = b_0 + b_1X_1 + b_2X_2 + ... + b_nX_n}$$

Where,
Y = Dependent Variable (DV)
X_1, X_2, X_n = Independent Variable (IV)
b0 = Y intercept
b_1, b_2, b_n = Coefficient of slope

In linear regression, our main aim is to estimate the values of Y-intercept and weights, minimize the cost function, and predict the output variable Y.

In logistic regression, we perform the exact same thing but with one small addition. We pass the result through a special function known as the **Sigmoid Function** to predict the output Y.

$$\mathbf{Y = Sigmoid\ (b_0 + b_1X_1 + b_2X_2 + ... + b_nX_n)}$$

$$\text{where, Sigmoid} = f(x) = \frac{1}{1 + e^{-Y}}$$

$$\mathbf{Therefore,\ Y = \frac{1}{1 + e^{-(b0 + b1X1 + b2X2 + ... + bnXn)}}}$$

Logistic regression uses the same basic formula as linear regression but it is regressing for the probability of a categorical outcome.

Linear regression gives a continuous value of output y for a given input X. Whereas, logistic regression gives a continuous value of P(Y=1) for a given input X, which is later converted to Y=0 or Y=1 based on a threshold value.

That's the reason, logistic regression has **"Regression"** in its name.

The key takeaways from this are:

1. Both Classification and Regression are Supervised learning methods.
2. In Classification, the task is to predict different class labels.
3. In Regression, the task is to predict a continuous quantity.
4. Both Linear regression and Logistic regression have the same underlying formula.
5. Logistic regression uses the Sigmoid function to predict the output class label for a given input

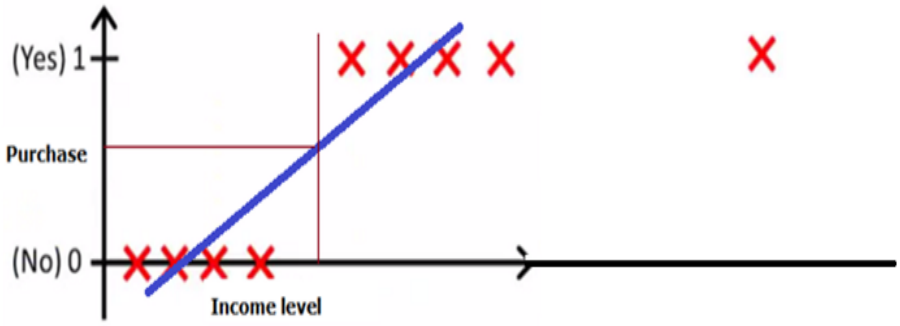
9. Explain the general intuition behind logistic regression.

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression) Logistic Regression is used when the **dependent variable(target) is categorical.**

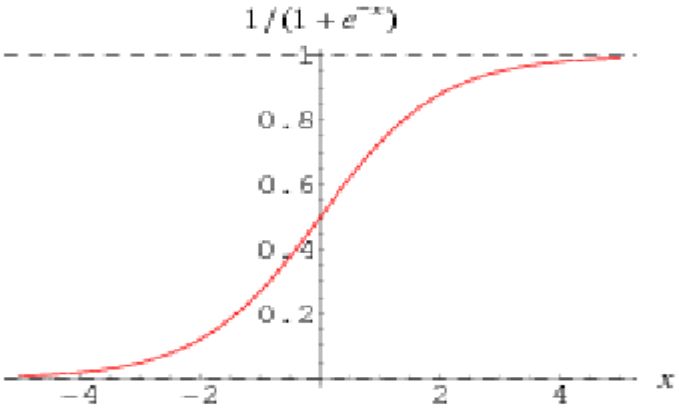
For example,

- To predict whether an email is spam (1) or (0)
- Whether the tumor is malignant (1) or not (0)

Intuition:



In the above two graphs it is clear that a linear line is not the ideal decision surface as it may miss-classify many values , what we ideally need is a decision surface which outputs the decision as 0 or 1/yes or no.



In addition, linear regression outputs values in the entire range of [-∞, ∞],whereas the actual values in this case are bound by 0 and 1.We therefore need a function that can output values between 0 and 1.A sigmoid or a logistic function allows that, and hence the name Logistic Regression.

Mathematical intuition of Logistic Regression

Before we proceed further we will acquaint ourselves with some basic mathematical terms ***Probability*** *and* ***Odds.***

The **probability** that an event will occur is the fraction of times you expect to see that event in many trials. If the probability of an event occurring is Y, then the probability of the event not occurring is 1-Y. **Probabilities** always range between 0 and 1.

The **odds** are defined as the probability that the event will occur divided by the probability that the event will not occur. Unlike **probability**, the odds are not constrained to lie between 0 and 1 but can take any value from zero to infinity.

If the probability of Success is P, then the odds of that event is:

$$odds = \frac{P}{1-P}$$

Y	*1*	*0*
Pr(Y=1)	*P*	*1- P*

**P= Success, 1-P = Failure*

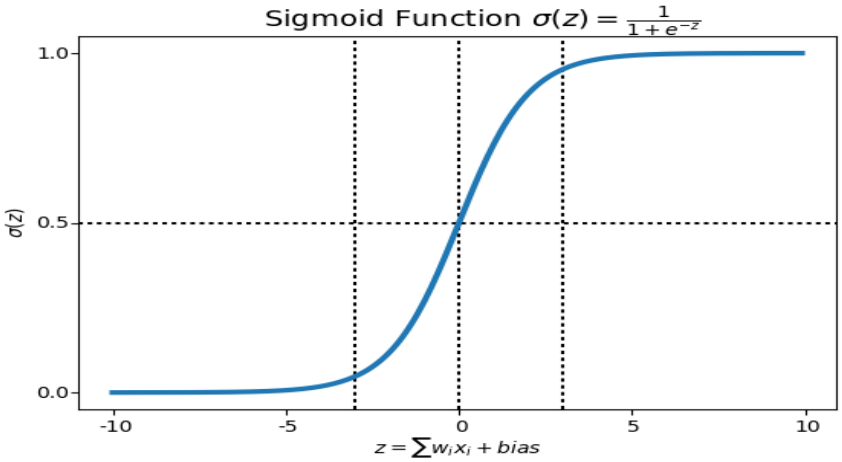
Odds: Success/ Failure.

10. Explain the significance of the sigmoid function.

Sigmoid Function acts as an activation function in machine learning which is used to add non-linearity in a machine learning model, in simple

words it decides which value to pass as output and what not to pass, there are mainly 7 types of Activation Functions which are used in machine learning and deep learning.

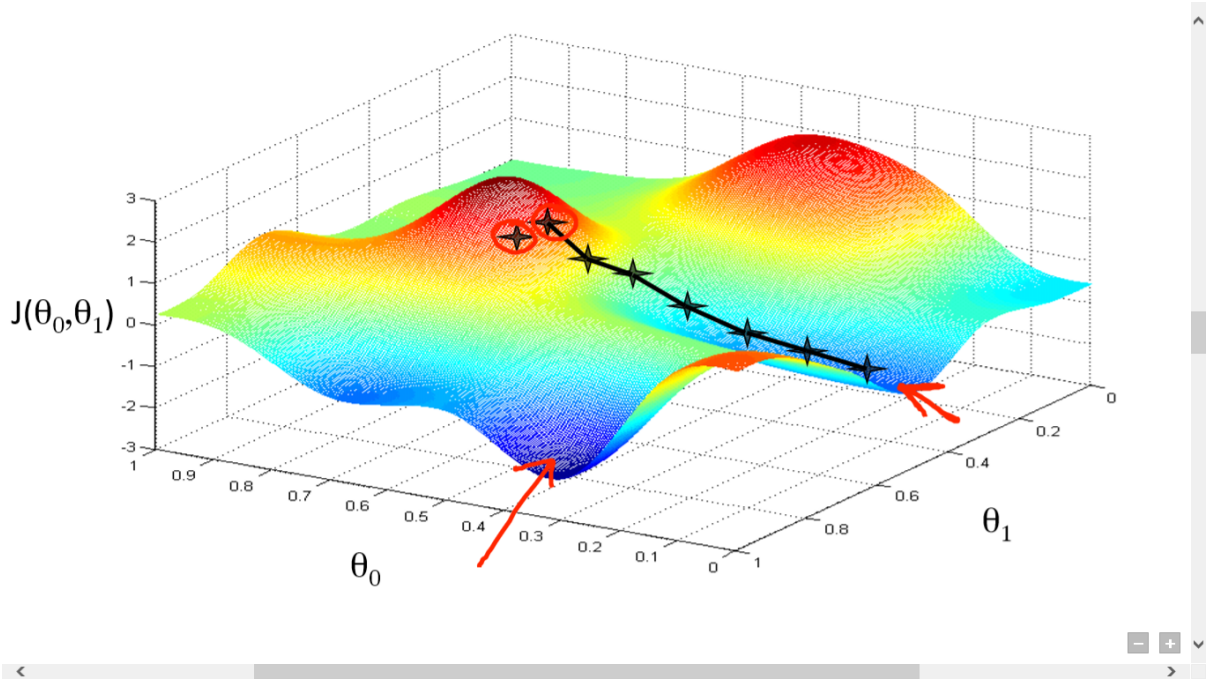
In order to map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.



Sigmoid Function In Logistic Regression Is An Advanced Regression Technique That Can Solve Various Classification Problems. Being A Classification Model, It Is Termed “Regression” Because The Fundamental Techniques Are Similar To Linear Regression.

11. How does Gradient Descent work in Logistic Regression?

We need to minimize the loss to make a good predicting algorithm. To do that, we have the Gradient Descent Algorithm.



Here we have plotted a graph between J()and . Our objective is to find the deepest point (global minimum) of this function. Now the deepest point is where the J()is minimum.

Two things are required to find the deepest point:

- Derivative – to find the direction of the next step.
- (Learning Rate) – magnitude of the next step

The idea is you first select any random point from the function. Then you need to compute the derivative of J()w.r.t. . This will point to the direction of the local minimum. Now multiply that resultant gradient with the Learning Rate. The Learning Rate has no fixed value, and is to be decided based on problems.

Now, you need to subtract the result from to get the new .

This update of should be simultaneously done for every (i).

Do these steps repeatedly until you reach the local or global minimum. By reaching the global minimum, you have achieved the lowest possible loss in your prediction.

Taking derivatives is simple. The major issue is with the Learning Rate. Taking a good learning rate is important and often difficult.

If you take a very small learning rate, each step will be too small, and hence you will take up a lot of time to reach the local minimum.

Now, if you tend to take a huge learning rate value, you will overshoot the minimum and never converge again. There is no specific rule for the perfect learning rate.

You need to tweak it to prepare the best model.

The equation for Gradient Descent is:

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

1. Start with random
2. Loop until convergence:
    1. Compute Gradient
    2. Update
3. Return

12. What are outliers and how can the sigmoid function mitigate the problem of outliers in logistic regression?

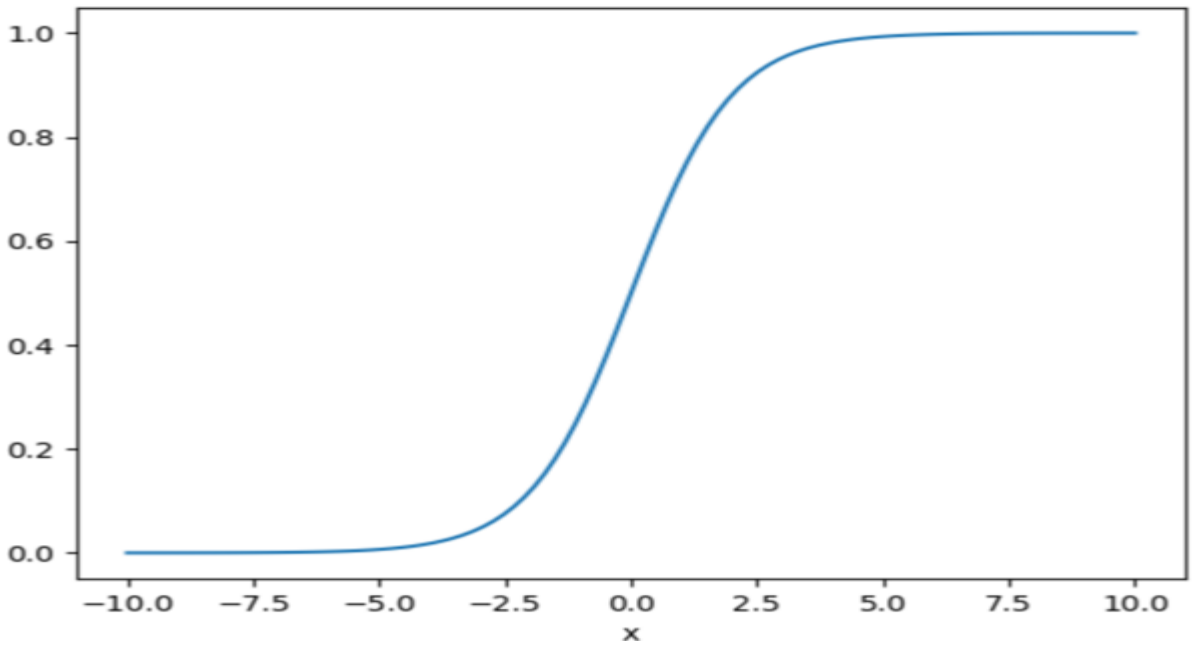
An outlier is a data point that differs significantly from other observations. These outliers can unduly influence the results of the analysis and lead to incorrect inferences. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set.

Sigmoid function:
It is a mathematical function having a characteristic that can take any real value and map it to between 0 to 1 shaped like the letter “S”.

The sigmoid function also called a logistic function.

The formula of the sigmoid function is:

Y = 1 / 1+e -z



So, if the value of z goes to positive infinity then the predicted value of y will become 1 and if it goes to negative infinity then the predicted value of y will become 0. And if the outcome of the sigmoid function is more than 0.5 then we classify that label as class 1 or positive class and if it is less than 0.5 then we can classify it to negative class or label as class 0.

Sigmoid Function acts as an activation function in machine learning which is used to add non-linearity in a machine learning model, in simple words it decides which value to pass as output and what not to pass

13. What are the outputs of the logistic model and the logistic function?

The Logistic model outputs the logits, i.e. log-odds; whereas the Logistic function outputs the probabilities.

Logistic model = $\alpha+1X_1+2X_2+....+kX_k$. Therefore, the output of the Logistic model will be logits.

Logistic function = f(z) = $\mathbf{1/(1+e-(\alpha+1X_1+2X_2+....+kX_k))}$. Therefore, the output of the Logistic function will be the probabilities.

14. Why can't we use Mean Square Error (MSE) as a cost function for logistic regression?

In Logistic Regression, we use the sigmoid function to perform a non-linear transformation to obtain the probabilities. If we square this nonlinear transformation, then it will lead to the problem of non-convexity with local minimums and by using gradient descent in such cases, it is not possible to find the global minimum. As a result, MSE is not suitable for Logistic Regression.

So, in the Logistic Regression algorithm, we used Cross-entropy or log loss as a cost function.

The property of the cost function for Logistic Regression is that:

- The confident wrong predictions are penalized heavily
- The confident right predictions are rewarded less

By optimizing this cost function, convergence is achieved

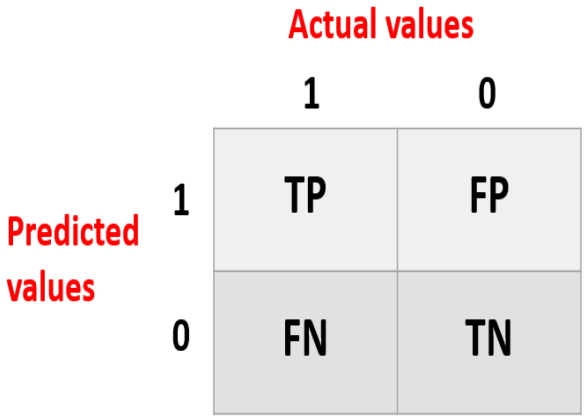
$$\text{Cost}(h\Theta(x), Y(\text{actual})) = -\log(h\Theta(x)) \text{ if } y=1$$

$$-\log(1- h\Theta(x)) \text{ if } y=0$$

15. What is the Confusion Matrix?

Confusion Matrix is the visual representation of the Actual VS Predicted values. It measures the performance of our Machine Learning classification model and looks like a table-like structure.

This is how a Confusion Matrix of a binary classification problem looks like :



Elements of Confusion Matrix

It represents the different combinations of Actual VS Predicted values. Let's define them one by one.

TP: True Positive: The values which were actually positive and were predicted positive.

FP: False Positive: The values which were actually negative but falsely predicted as positive. Also known as Type I Error.

FN: False Negative: The values which were actually positive but falsely predicted as negative. Also known as Type II Error.

TN: True Negative: The values which were actually negative and were predicted negative.

Suppose your confusion matrix is a simple 2 by 2 table, given by:

		Predicted	
		Negative	Positive
Actual	Negative	a	b
	Positive	c	d

Here a is the number of true negatives, and d the number of true positives. b is the number of false positives, and c is the number of false negatives.

- The true positive rate is given by d/(c + d), and is also called the recall. It tells us what proportion of positive cases were correctly identified.
- The false positive rate, or proportion of negative cases (incorrectly) identified as positive, is given by b/(a + b).
- The true negative rate is a/(a + b), and represents the proportion of negative cases that were correctly identified.
- The false negative rate is c/(c + d), and tells us what proportion of positive cases were incorrectly labelled as negative.

The below-given metrics of confusion matrix determine how well our model performs-

1. Accuracy
2. Precision (Positive Prediction Value)
3. Recall (True Positive Rate or Sensitivity)
4. F beta Score

1. ACCURACY:

Accuracy is the number of correctly (True) predicted results out of the total.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Accuracy should be considered when TP and TN are more important and the dataset is balanced because in that case the model will not get biased based on the class distribution. But in real-life classification problem, imbalanced class distribution exists.

For example, we have an imbalanced test data with 900 records of positive class (1) and 100 records of negative class (0). And our model predicted all records as positive (1). In that scenario, TP will be 900 and TN will be 0. Thus, accuracy = (900 + 0) / 1000 = 90%

Therefore, we should consider Precision, Recall and F Score as a better metric to evaluate the model.

2. PRECISION:

Out of the total predicted positive values, how many were actually positive

Precision = TP / (TP + FP)

When precision should be considered?

Taking a use case of Spam Detection, suppose the mail is not spam (0), but the model has predicted it as spam (1) which is FP. In this scenario, one can miss the important mail. So, here we should focus on reducing the FP and must consider precision in this case.

3. RECALL:

Out of the total actual positive values, how many were correctly predicted as positive

Recall= TP / (TP + FN)

When recall should be considered?
In Cancer Detection, suppose if a person is having cancer (1), but it is not predicted (0) by the model which is FN. This could be a disaster. So, in this scenario, we should focus on reducing the FN and must consider recall in this case.

Based on the problem statement, whenever the FP is having a greater impact, go for Precision and whenever the FN is important, go for Recall

4. F beta SCORE
In some use cases, both precision and recall are important. Also, in some use cases even though precision plays an important role or recall plays is important, we should combine both to get the most accurate result.

Selecting beta value

F-1 Score (beta =1)

When FP and FN both are equally important. This allows the model to consider both precision and recall equally using a single score.

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

F-1 Score is the *Harmonic Mean* of precision and recall.

Smaller beta value such as (beta = 0.5).
If the impact of FP is high. This will give more weight to precision than to recall.

Higher beta value such (beta = 2)
If the impact of FN is high. Thus, giving more weight to recall and less to precision.

16. How do you define a classification report?

- A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report as shown below.

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	50
Iris-versicolor	0.77	0.96	0.86	50
Iris-virginica	0.95	0.72	0.82	50
avg / total	0.91	0.89	0.89	150

- The report shows the main classification metrics precision, recall and f1-score on a per-class basis. The metrics are calculated by using true and false positives, true and false negatives. Positive and negative in this case are generic names for the predicted classes. There are four ways to check if the predictions are right or wrong:

    - **TN / True Negative:** when a case was negative and predicted negative
    - **TP / True Positive:** when a case was positive and predicted positive
    - **FN / False Negative:** when a case was positive but predicted negative
    - **FP / False Positive:** when a case was negative but predicted positive

- **Precision – What percent of your predictions were correct?**
  Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as the ratio of true positives to the sum of true and false positives.
  TP – True Positives
  FP – False Positives

  Precision – Accuracy of positive predictions.
  Precision = TP/(TP + FP)

- Recall – What percent of the positive cases did you catch?

Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives.

FN – False Negatives

Recall: Fraction of positives that were correctly identified.
Recall = TP/(TP+FN)

- F1 score – What percent of positive predictions were correct?

  The F_1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally speaking, F_1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F_1 should be used to compare classifier models, not global accuracy.

- F1 Score = 2*(Recall * Precision) / (Recall + Precision)

17. What are the false positives and false negatives?

- A **false positive** is an error in binary classification in which a test result incorrectly indicates the presence of a condition (such as a disease when the disease is not present)
  FP / False Positive: when a case was negative but predicted positive
- A false positive error is a type I error where the test is checking a single condition, and wrongly gives an affirmative (positive) decision.
- a **false negative** is the opposite error, where the test result incorrectly indicates the absence of a condition when it is actually present.
  FN / False Negative: when a case was positive but predicted negative
- A false negative error is a type II error occurring in a test where a single condition is checked for, and the result of the test is erroneous, that the condition is absent.

18. What are the true positive rate (TPR) and false-positive rate (FPR)?

- The false positive rate is calculated as FP/FP+TN,
  Where,
  FP is the number of false positives

TN is the number of true negatives
FP+TN being the total number of negatives.
It's the probability that a false alarm will be raised: that a positive result will be given when the true value is negative.

- The true positive rate (TPR, also called sensitivity) is calculated as TP/TP+FN.
  TP / True Positive: when a case was positive and predicted positive
  FN / False Negative: when a case was positive but predicted negative
- TPR is the probability that an actual positive will test positive.

19. What is the false-positive rate (FPR) and false-negative rate (FNR)?

- The false positive rate is calculated as FP/FP+TN,
  Where,
  FP is the number of false positives
  TN is the number of true negatives
  FP+TN being the total number of negatives.
  It's the probability that a false alarm will be raised: that a positive result will be given when the true value is negative.
- The false negative rate – also called the miss rate – is the probability that a true positive will be missed by the test.
- It's calculated as FN/FN+TP,
  where FN is the number of false negatives
  TP is the number of true positives
  (FN+TP being the total number of positives).

20. What are precision and recall? Explain the importance with examples?

- Precision is defined as the ratio of correctly classified positive samples (True Positive) to a total number of classified positive samples (either correctly or incorrectly).

  Precision = True Positive/True Positive + False Positive
  Precision = TP/TP+FP

The precision of a machine learning model will be low when the value of; TP+FP (denominator) > TP (Numerator)

The precision of the machine learning model will be high when Value of; TP (Numerator) > TP+FP (denominator)

If there is a requirement of classifying all positive as well as Negative samples as Positive, whether they are classified correctly or incorrectly, then use Precision.

- When it comes to components that are being used on a regular or daily basis, precision is of utmost importance. Precision is essential, precision is intricate, and precision is beautiful; more than anything else, precision is necessary.

- When you drive a car, a motorbike, an aeroplane or even a seated lawnmower, your movements are determined by the use of an engine. Engines are engineered for a specific purpose and each and every component plays an important role in ensuring this happens safely. This is especially important when it comes to vehicles carrying passengers.

- The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The ***recall measures the model's ability to detect positive samples***. The higher the recall, the more positive samples detected.

  Recall = True Positive/True Positive + False Negative
  Recall = TP/TP+FN
  TP- True Positive
  FN- False Negative
  Recall of a machine learning model will be low when the value of; TP+FN (denominator) > TP (Numerator)
  Recall of machine learning model will be high when Value of; TP (Numerator) > TP+FN (denominator)

- Unlike Precision, Recall is independent of the number of negative sample classifications. Further, if the model classifies all positive samples as positive, then Recall will be 1.

- Further, on the other end, if our goal is to detect only all positive samples, then use Recall. Here, we should not care how negative samples are correctly or incorrectly classified the samples.

- Recall places a high importance on reducing the number of false negatives, for example positive cases that are misclassified by the model as negatives. For that reason, it is important in mission-critical applications where a false negative could lead to loss of life or millions of dollars in damages. In such applications, it is essential to maximize recall.

21. What is the purpose of the precision-recall curve?

Precision and recall are performance metrics used for pattern recognition and classification in machine learning. These concepts are essential to build a perfect machine learning model which gives more precise and accurate results.

Precision-Recall is a useful measure of success of prediction when the classes are very imbalanced. In information retrieval, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned.

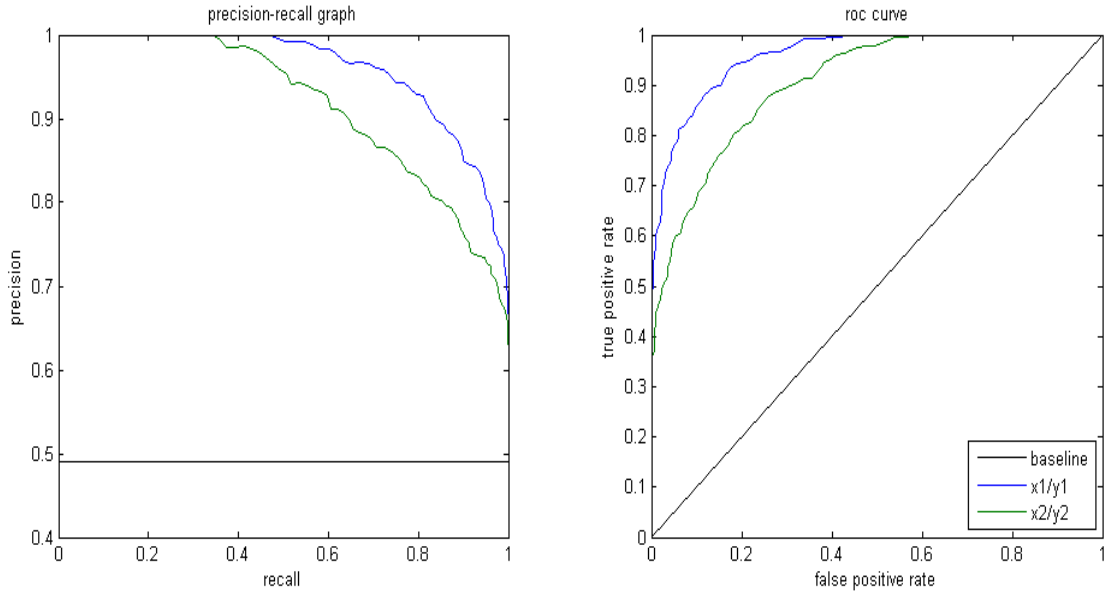
The precision-recall curve shows the trade-off between precision and recall for different threshold. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate. High scores for both show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

A PR curve is simply a graph with Precision values on the y-axis and Recall values on the x-axis. In other words, the PR curve contains TP/(TP+FN) on the y-axis and TP/(TP+FP) on the x-axis.

It is important to note that Precision is also called the Positive Predictive Value (PPV).

Recall is also called Sensitivity, Hit Rate or True Positive Rate (TPR).

The figure below shows a juxtaposition of sample PR and ROC curves.



22. What is the f1 score and Explain its importance?

The F1 score is a popular performance measure for classification and often preferred over, for example, accuracy when data is unbalanced, such as when the quantity of examples belonging to one class significantly outnumbers those found in the other class.

By definition, F1-score is the harmonic mean of precision and recall. It combines precision and recall into a single number using the following **formula**:

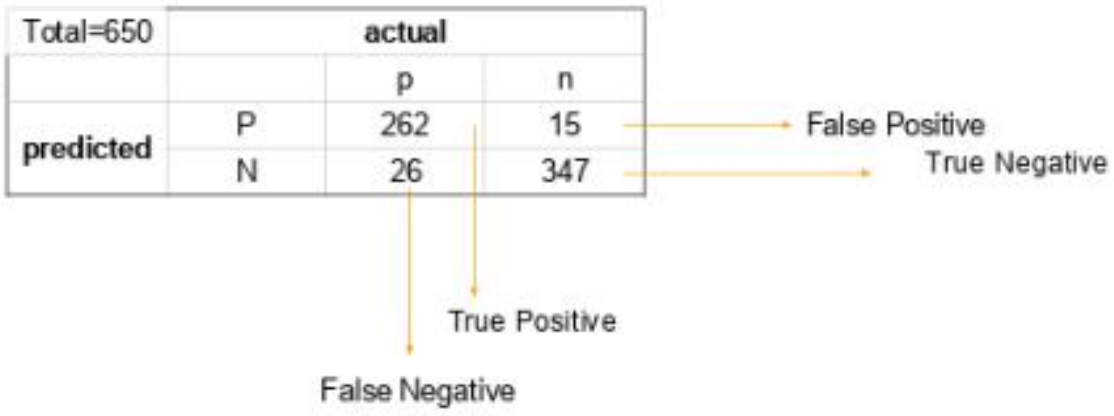
$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

This formula can also be equivalently written as,

$$\text{F1-score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}.$$

Notice that F1-score takes both precision and recall into account, which also means it accounts for both FPs and FNs. The higher the precision and recall, the higher the F1-score. F1-score ranges between 0 and 1. The closer it is to 1, the better the model.

23. Write the equation and calculate the precision and recall rate.



Consider the confusion matrix.
Precision = (TP) / (TP+FP)
= 262 / 277
= 0.94
Recall Rate = (TP) / (TP+FN)
= 262 / 288
= 0.90

24. How can you calculate accuracy using a confusion matrix?

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition: Accuracy = Number of correct predictions Total number of predictions.

To calculate accuracy, use the following formula:
Accuracy = (TP+TN)/(TP+TN+FP+FN)
= (262+347)/(262+15+26+347)= 0.9369=93%
Precision = (TP) / (TP+FP)
= 262 / 277
= 0.94
Recall Rate = (TP) / (TP+FN)

= 262 / 288
= 0.90

- **F1-Score** = (2 * Precision * Recall) / (Precision + Recall)=0.92=92%

25. What is sensitivity?

Sensitivity is a measure of how well a machine learning model can detect positive instances. It is also known as the true positive rate (TPR) or recall. Sensitivity is used to evaluate model performance because it allows us to see how many positive instances the model was able to correctly identify.

Sensitivity = (TP)/(TP + FN)= 0.90

26. What is Specificity?

Specificity itself can be described as the algorithm/model's ability to predict a true negative of each category available. In literature, it is also known simply as the true negative rate
Specificity=TN/(TN+FP)=0.91

27. What is ROC Curve?

A Receiver Operator Characteristic (ROC) curve is a graphical plot used to show the diagnostic ability of binary classifiers. It was first used in signal detection theory but is now used in many other areas such as medicine, radiology, natural hazards and machine learning. In this post I'll show you how a ROC curve is created and how to interpret the ROC curve.

The area under the ROC curve (AUC) results were considered excellent for AUC values between 0.9-1, good for AUC values between 0.8-0.9, fair for AUC values between 0.7-0.8, poor for AUC values between 0.6-0.7 and failed for AUC values between 0.5-0.6.

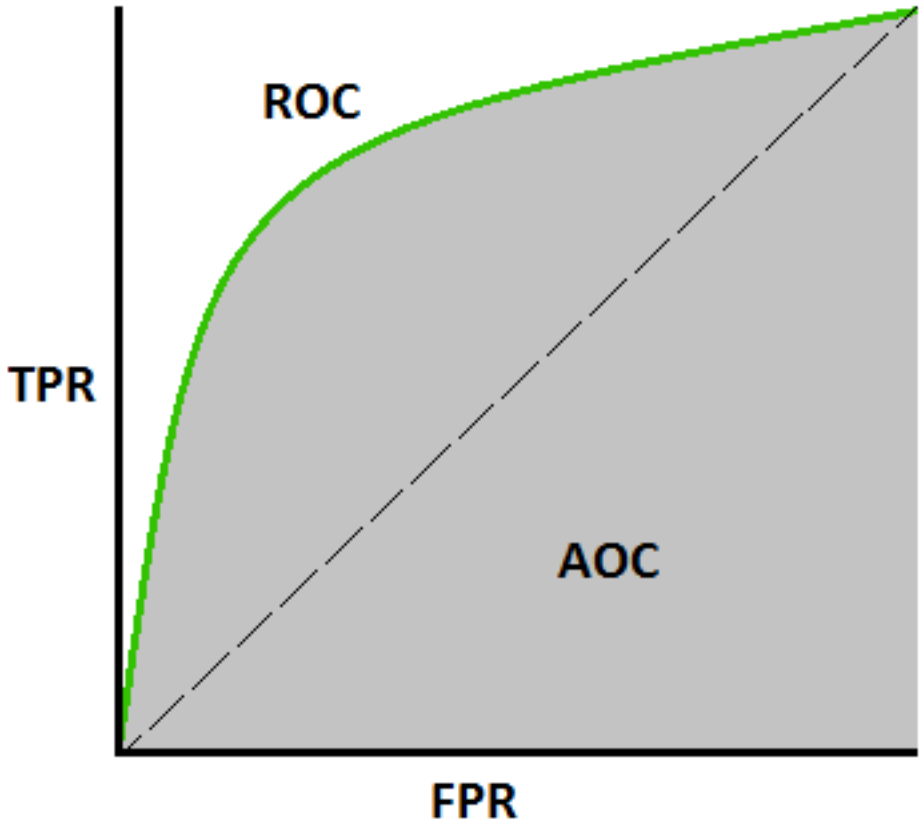
28. What is the importance of the ROC curve?

ROC curves are frequently used to show in a graphical way the connection/trade-off between clinical sensitivity and specificity for every

possible cut-off for a test or a combination of tests. In addition the area under the ROC curve gives an idea about the benefit of using the test(s) in question.

As the area under an ROC curve is a measure of the usefulness of a test in general, where a greater area means a more useful test, the areas under ROC curves are used to compare the usefulness of tests.

The ROC curve **shows you sensitivity and specificity at all possible thresholds**, so if you find a point that represents the right tradeoff, you can choose the threshold that goes with that point on the curve.



29. What are the advantages of ROC Curve?

The ROC curve shows you sensitivity and specificity at all possible thresholds, so if you find a point that represents the right tradeoff, you can choose the threshold that goes with that point on the curve.

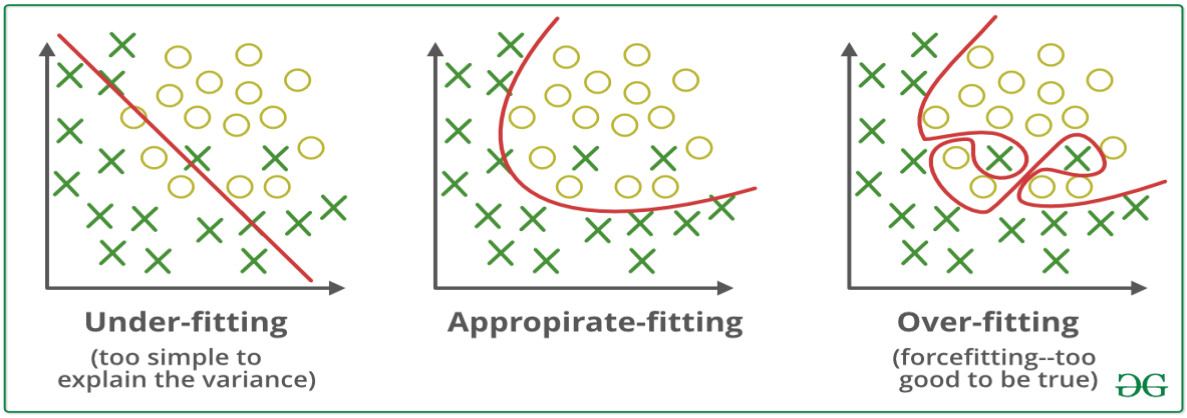
30. What is Overfitting?

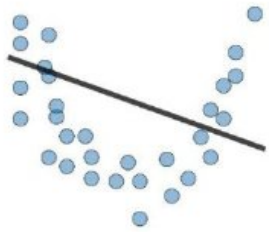
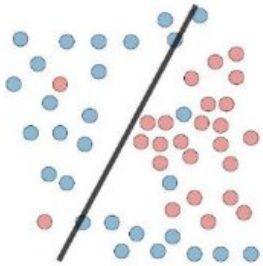
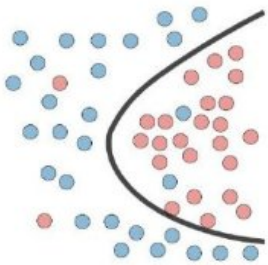
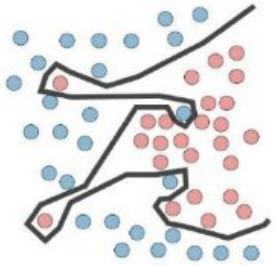
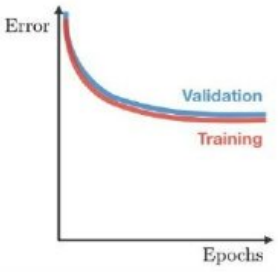
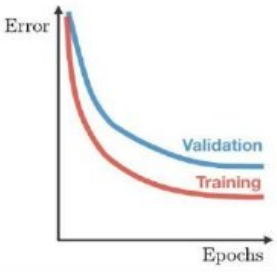
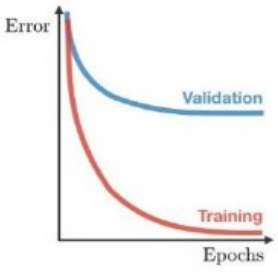
Overfitting occurs when the model cannot generalize and fits too closely to the training dataset instead. It gives good accuracy to training dataset but it wont give accuracy to testing dataset. Overfitting happens due to several reasons, such as: The training data size is too small and does not contain enough data samples to accurately represent all possible input data values.

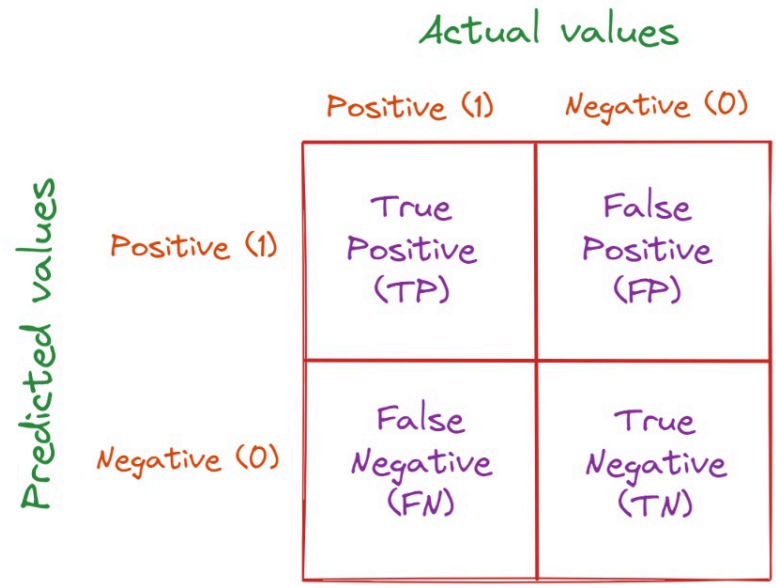
When machine learning algorithms are constructed, they leverage a sample dataset to train the model. However, when the model trains for too long on sample data or when the model is too complex, it can start to learn the "noise," or irrelevant information, within the dataset. When the model memorizes the noise and fits too closely to the training set, the model becomes "overfitted," and it is unable to generalize well to new data. If a model cannot generalize well to new data, then it will not be able to perform the classification or prediction tasks that it was intended for.

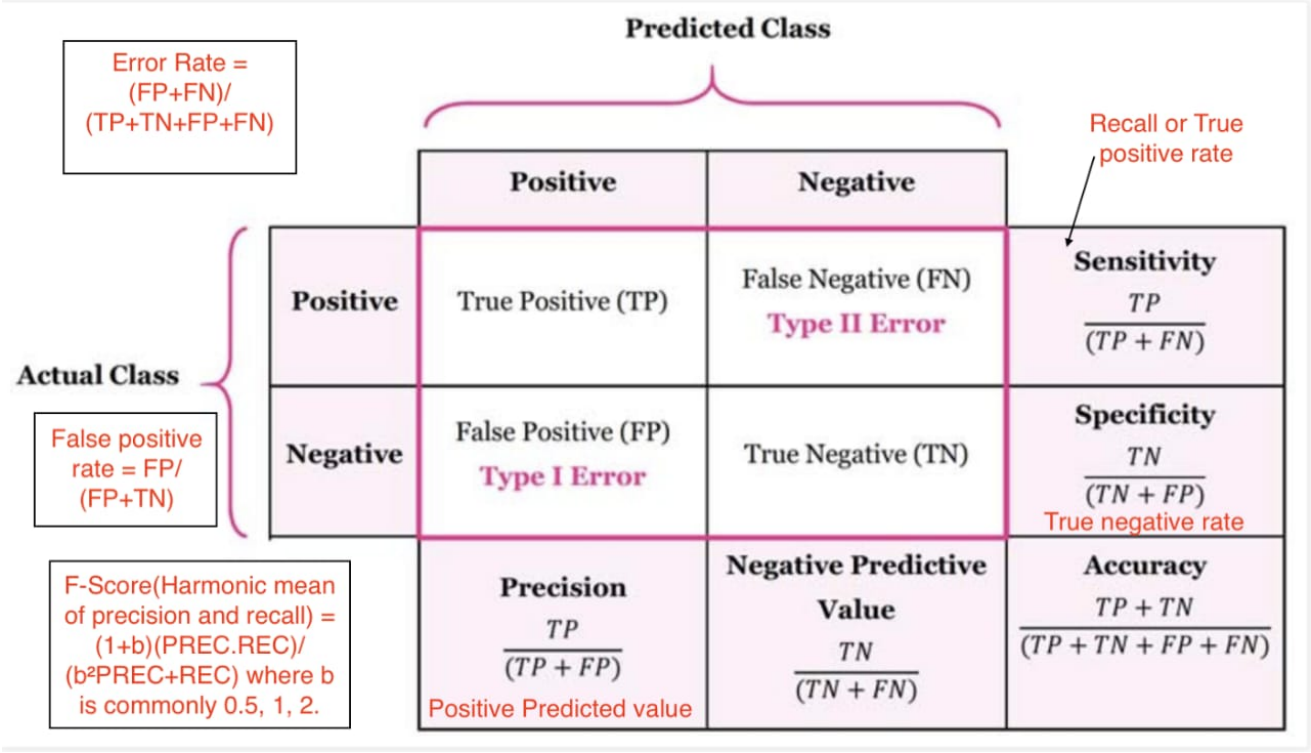
Low error rates and a high variance are good indicators of overfitting. In order to prevent this type of behaviour, part of the training dataset is typically set aside as the "test set" to check for overfitting. If the training

data has a low error rate and the test data has a high error rate, it signals overfitting.



	Underfitting	Just right	Overfitting
Symptoms	- High training error - Training error close to test error - High bias	- Training error slightly lower than test error	- Low training error - Training error much lower than test error - High variance
Regression			
Classification			
Deep learning			
Remedies	- Complexify model - Add more features - Train longer		- Regularize - Get more data





Q.31) What is Underfitting?

A statistical model is said to be underfitted if it can't generalize well with both seen and unseen data.

Underfitting occurs when our machine learning model is not able to capture the underlying trend of the data.

In the case of underfitting, the model is not able to learn enough from the training data, and hence it reduces the accuracy and produces unreliable predictions.

A model which does not perform well on both training and test set. In case of Underfitting, our training set error is high, so it will have high bias and our test set error is also high, so it will have high variance. Underfitted Model — High Bias and High Variance

Reasons for Underfitting:

1. High bias and low variance
2. The size of the training dataset used is not enough.
3. The model is too simple.
4. Training data is not cleaned and also contains noise in it. Techniques to reduce underfitting:
5. Increase model complexity
6. Increase the number of features, performing feature engineering
7. Remove noise from the data.
8. Increase the number of epochs.

Q.32) What is Bias and variance in machine learning?

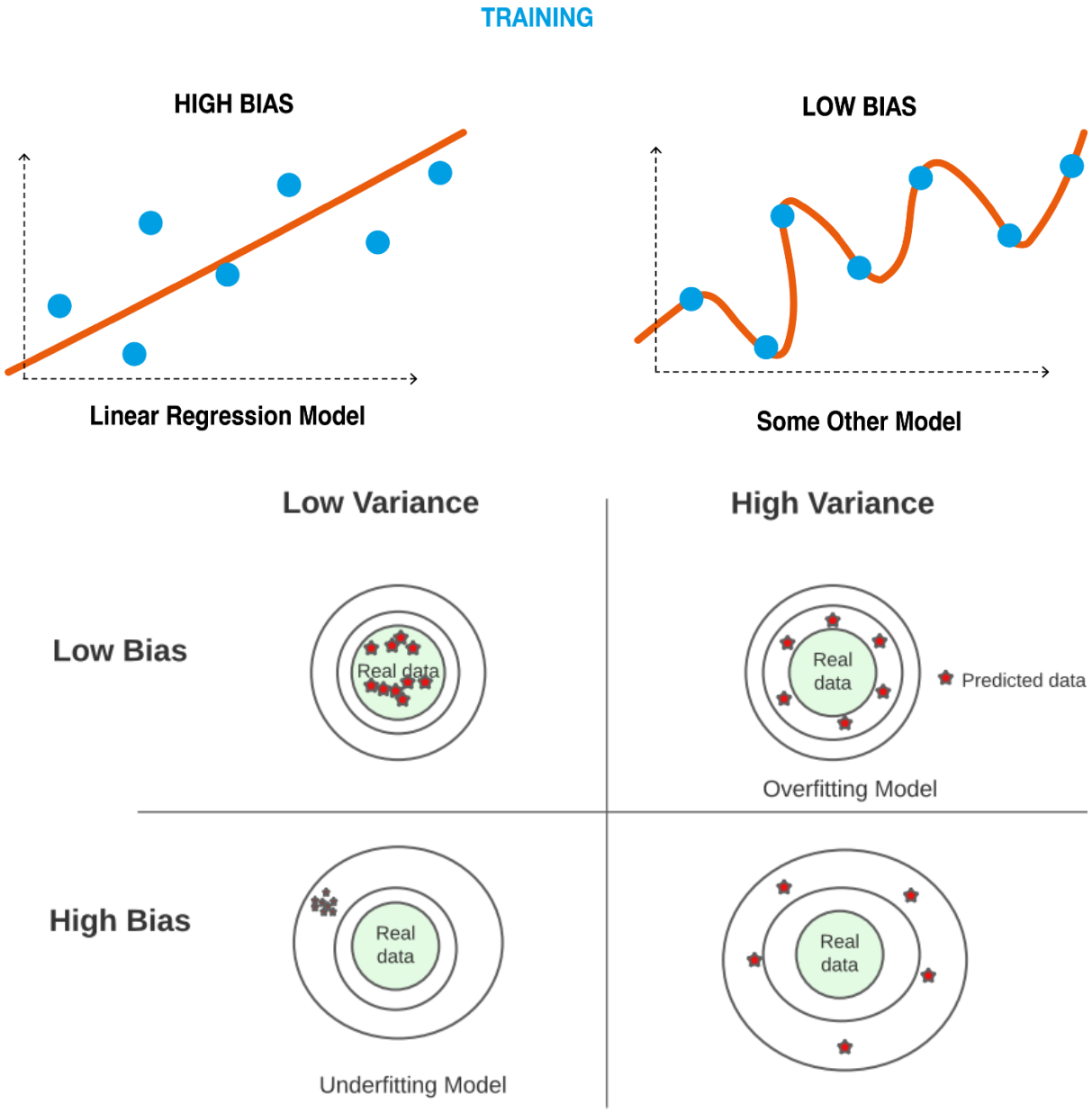
Bias:- Bias is a disproportionate weight in favor of or against a feature.

The bias is known as the difference between the prediction of the values by the ML model and the correct value.

The difference between the average value predicted by our Machine Learning model and the correct target value is known as **Bias**. A model that makes incorrect predictions about a dataset is called a **biased model**. This model oversimplifies the target function to make it easier to learn.

Being high in biasing gives a large error in training as well as testing data. Its recommended that an algorithm should always be low biased to avoid the problem of underfitting.

Bias is a training set error



Variance:- Variance measures how far a set of numbers is spread out from their average value

Variance is a test set error.

The variability of model prediction for a given data point which tells us spread of our data is called the variance of the model.

The amount of variability in the target function in response to a change in the training data is known as **Variance**. When a model takes into consideration the noise and fluctuation in the data, it is said to be of High Variance.

The model with high variance has a very complex fit to the training data and thus is not able to fit accurately on the data which it hasn't seen before. As a result, such models perform very well on training data but has high error rates on test data.

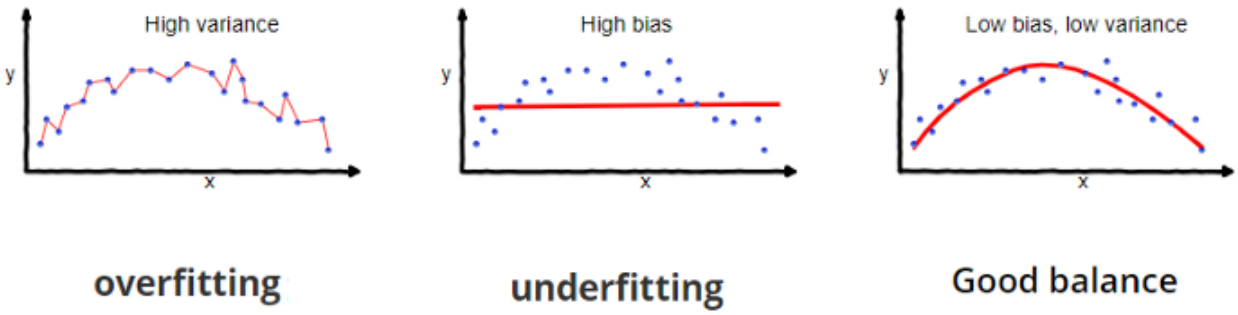
When a model is high on variance, it is then said to as **Overfitting of Data**. Overfitting is fitting the training set accurately via complex curve and high order hypothesis but is not the solution as the error with unseen data is high. While training a data model variance should be kept low.

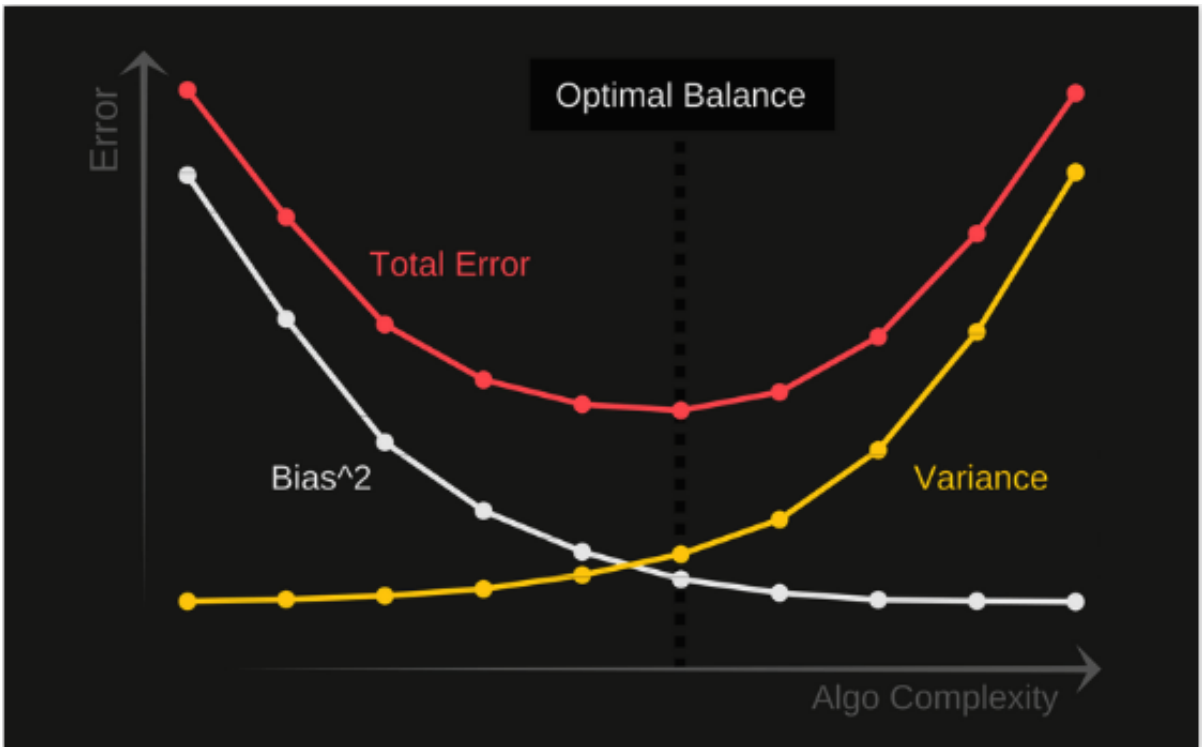
Q.33) What is The trade-off between Bias and Variance:

If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand if our model has large number of parameters then it's going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data. This trade-off in complexity is why there is a tradeoff between bias and variance. An algorithm can't be more complex and less complex at the same time.

Total Error

To build a good model, we need to find a good balance between bias and variance such that it minimizes the total error.





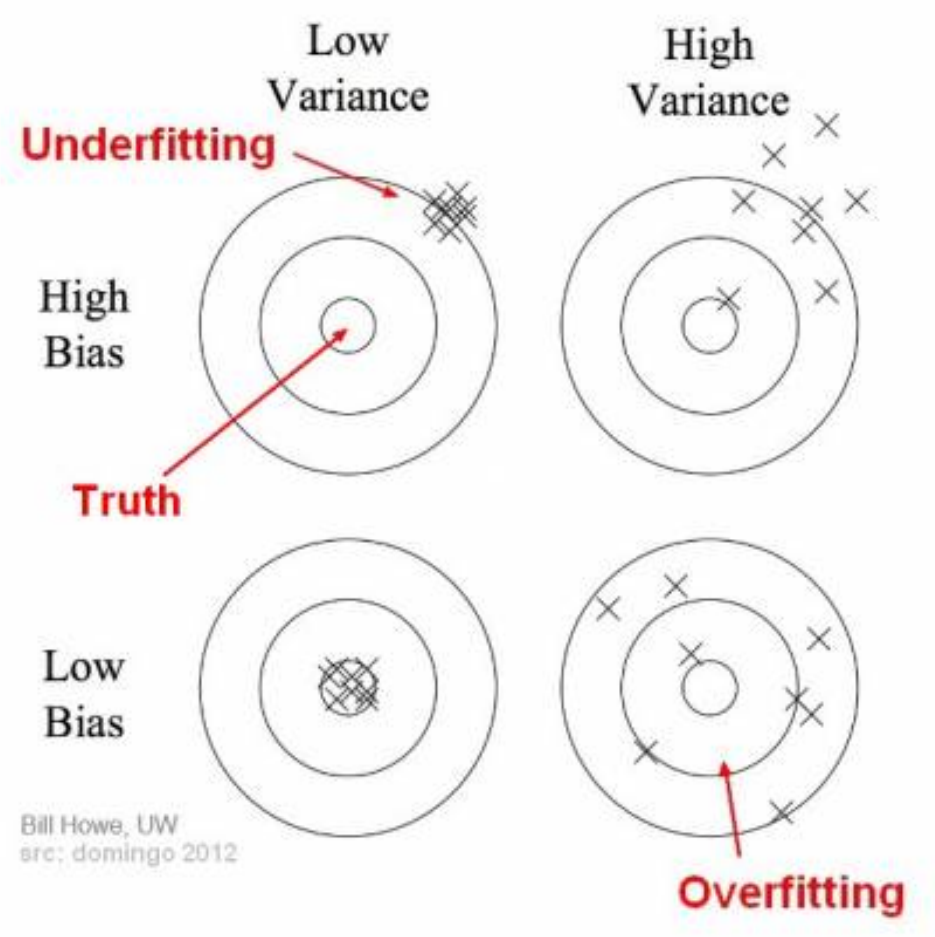
Mathematical Definition

$Err(x)=Bias^2+Variance+Irreducible\ Error$

Irreducible Error refers to the amount of noise in our data. No matter how good the data, it will contain an irreducible error that cannot be removed.

An optimal balance of bias and variance would never overfit or underfit the model. Therefore understanding bias and variance is critical for understanding the behavior of prediction models.

Q.34) Bias and variance using the bulls-eye diagram.



The figure above will give a better understanding of all possible cases when building a machine learning model. To have a clearer idea, the blue dots represent our model predictions. These predictions should be centered in the red area so we can say that we have an accurate predictive model.

The best use case is to have low bias and low variance. The worst case is to have both high Variance and high Bias.

Also, having a low bias and high variance is not ideal but it is still better than having high bias and low variance.

Some of the most common techniques to resolve cases of high Variance would be:

add more training examples. Try smaller sets of features (because you are overfitting). Try dimensionality reduction techniques. Get rid of irrelevant/redundant features. Use Regularization techniques to prevent model overfitting. (increase lambda) In case of high bias we can:

Try to add more features Try to make the model more complicated (add polynomial features) Try to fit the data better. apply regularization techniques (decrease lambda) NB: Lambda is the regularization rate.

If lambda is too high, the model will be simple, we run the risk of underfitting the data. if lambda values too low, your model will be more complex, and you run the risk of overfitting the data. An optimal balance of bias and variance would never overfit or underfit the model.

Therefore understanding bias and variance is critical for understanding the behavior of prediction models.

Q.35) What is L1 AND L2 Regularization?

L1 Regularization:-

L1 Regularization, also called a lasso regression, adds the “absolute value of magnitude” of the coefficient as a penalty term to the loss function. A regression model that uses the L1 regularization technique is called lasso regression.

Again, if lambda is zero, then we'll get back OLS (ordinary least squares) whereas a very large value will make coefficients zero, which means it will become underfit.

L2 Regularization:-

L2 Regularization, also called a ridge regression, adds the “squared magnitude” of the coefficient as the penalty term to the loss function. A model that uses the L2 is called ridge regression. The highlighted part below represents the L2 regularization element.

Here, if lambda is zero then you can imagine we get back OLS. However, if lambda is very large then it will add too much weight and lead to underfitting. This technique works very well to avoid overfitting issues.

L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N} (y_i - \sum_{j=0}^{M} x_{ij} W_j)^2 + \lambda \sum_{j=0}^{M} |W_j|$$

L2 Regularization

$$\text{Cost} = \sum_{i=0}^{N} (y_i - \sum_{j=0}^{M} x_{ij} W_j)^2 + \lambda \sum_{j=0}^{M} W_j^2$$

Loss function

Regularization Term

36. How do you find the best alpha for ridge regression?

The alpha term acts as the control parameter, which determines, how much significance should be given to Xi for the Bi coefficient. If Alpha is close to zero, the Ridge term itself is very small and thus the final error is based on RSS alone. If Alpha is too large, the impact of shrinkage grows and the coefficients B1, B2 ... Bn tends to zero.

Choosing the right value helps the model learn the right features and better generalize the coefficients. One of the methods that help in choosing the right value is Cross-validation.

-- Using an alpha value of 10, the evaluation of the model, the train, and test data indicate better performance on the ridge model than on the linear regression model.

-- An instance of Ridge is created with a value of alpha as 0.1. The alpha value determines how much weight is given to the penalty term. A higher alpha value means that more weight is given to the penalty term, and a lower alpha value means that less weight is given to the penalty term.

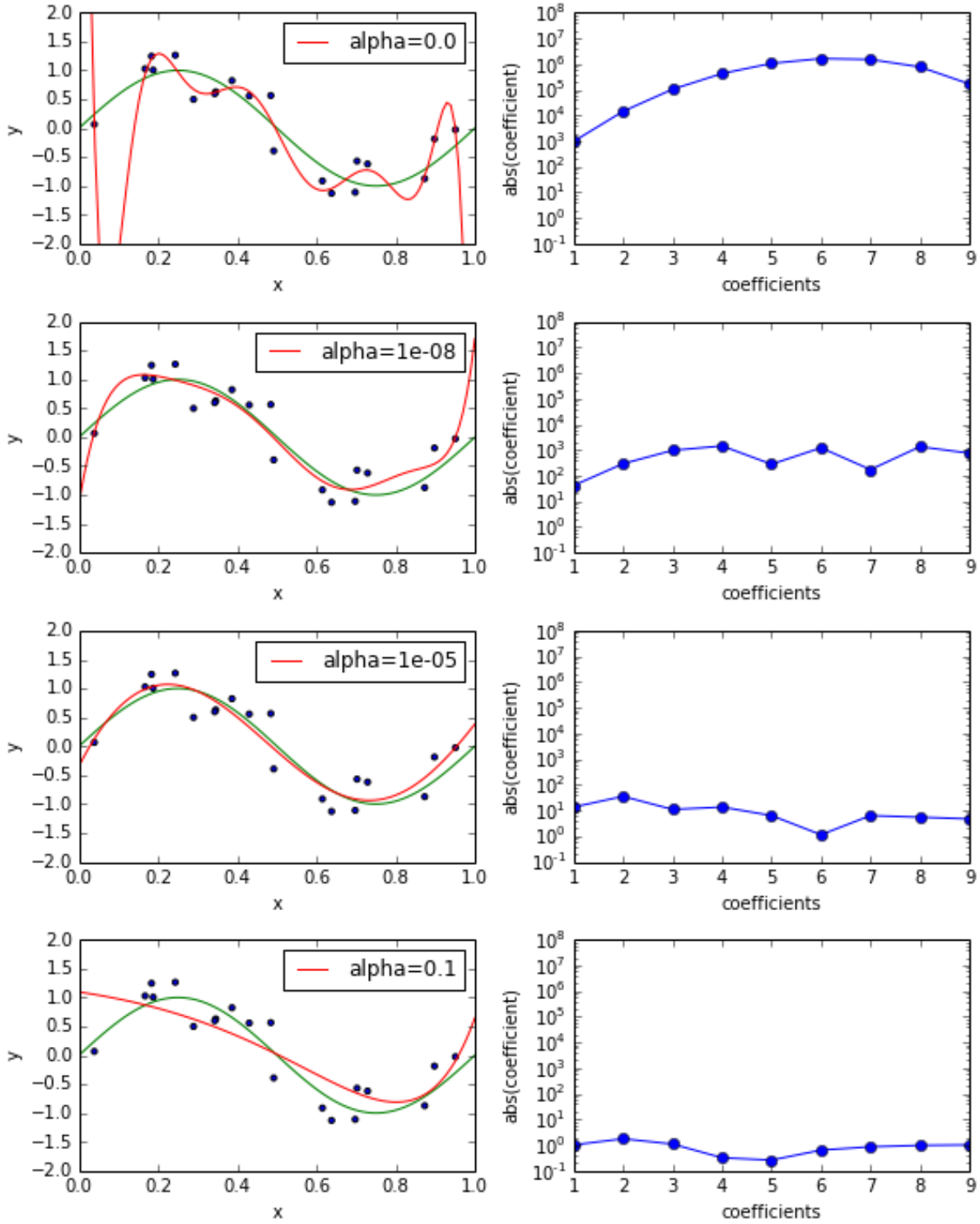
The L2 norm term in ridge regression is weighted by the regularization parameter alpha

So, if the alpha value is 0, it means that it is just an Ordinary Least Squares Regression model. So, the larger is the alpha, the higher is the smoothness constraint.

So, the smaller the value of alpha, the higher would be the magnitude of the coefficients.

I would add an image which would help you visualize how the alpha value influences the fit:

So, the alpha parameter need not be small. But, for a larger alpha, the flexibility of the fit would be very strict.


alpha=0.0
alpha=1e-08
alpha=1e-05
alpha=0.1
x
y
coefficients
abs(coefficient)

37. Does scaling affect logistic regression?

For simple linear/logistic regression (without regularization): no need to scale variables.

For linear/logistic regression with regularization: you need to perform scaling.

-- The performance of logistic regression did not improve with data scaling.

-- Centering/scaling does not affect your statistical inference in regression models — the estimates are adjusted appropriately and the p-values will be the same.

There would be no need for feature scaling for linear or logistic regression if there's no regularization.

However, with regularization, feature scaling in a logistic model is a fundamental step that helps to overcome overfitting and get more accurate results.