**1. Explain the terms Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning?**

Artificial Intelligence (AI) is the domain of producing intelligent machines. ML refers to systems that can assimilate from experience (training data) and Deep Learning (DL) states to systems that learn from experience on large data sets. ML can be considered as a subset of AI. Deep Learning (DL) is ML but useful to large data sets. The figure below roughly encapsulates the relation between AI, ML, and DL:

In summary, DL is a subset of ML & both were the subsets of AI.

Additional Information: ASR (Automatic Speech Recognition) & NLP (Natural Language Processing) fall under AI and overlay with ML & DL as ML is often utilized for NLP and ASR tasks.

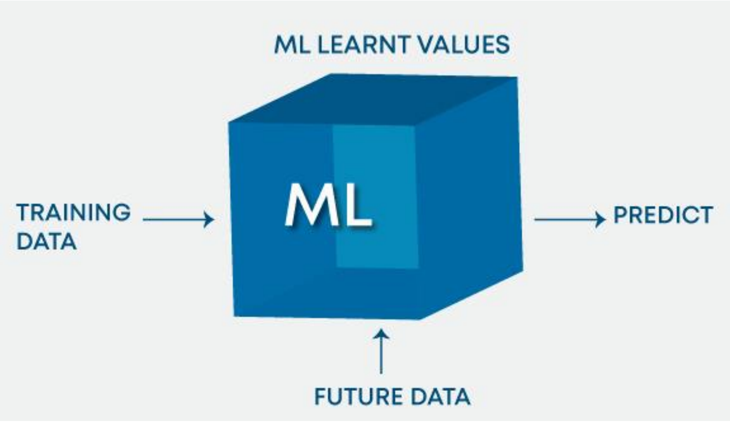
**2. What are the different types of Learning/ Training models in ML?**

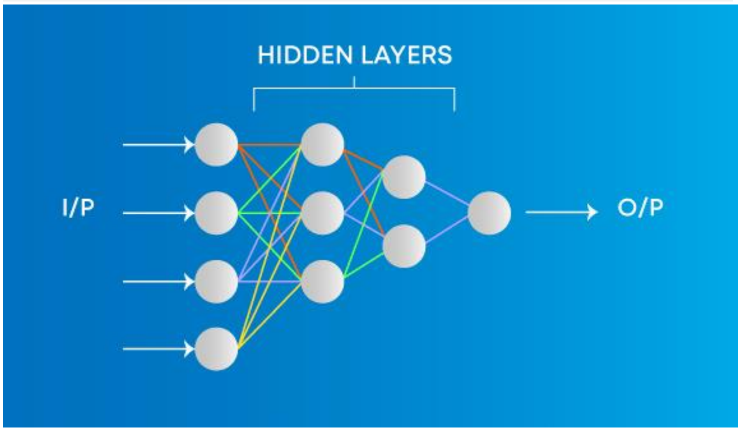
***A. Supervised learning*:**
The machine learns using labelled data. The model is trained on an existing data set before it starts making decisions with the new data.
*The target variable is continuous:* Linear Regression, polynomial Regression, and quadratic Regression.
*The target variable is categorical:* Logistic regression, Naive Bayes, KNN, SVM, Decision Tree, Gradient Boosting, ADA boosting, Bagging, Random forest etc.

***B. Unsupervised learning:***
The machine is trained on unlabelled data and without any proper guidance. It automatically infers patterns and relationships in the data by creating clusters. The model learns through observations and deduced structures in the data.
Principal component Analysis, Factor analysis, Singular Value Decomposition etc.

***C. Reinforcement Learning:***
The model learns through a trial and error method. This kind of learning involves an agent that will interact with the environment to create actions and then discover errors or rewards of that action.

**3. What is the difference between deep learning and machine learning?**





Machine Learning involves algorithms that learn from patterns of data and then apply it to decision making. Deep Learning, on the other hand, is able to learn through processing data on its own and is quite similar to the human brain where it identifies something, analyse it, and makes a decision.
The key differences are as follows:

- The manner in which data is presented to the system.
- Machine learning algorithms always require structured data and deep learning networks rely on layers of artificial neural networks.

**4. What is the main key difference between supervised and unsupervised machine learning?**

| **Supervised learning** | **Unsupervised learning** |
|---|---|
| The supervised learning technique needs labelled data to train the model. For example, to solve a classification problem (a supervised learning task), you need to have label data to train the model and to classify the data into your labelled groups. | Unsupervised learning does not need any labelled dataset. This is the main key difference between supervised learning and unsupervised learning. |

**5. How do you select important variables while working on a data set?**

There are various means to select important variables from a data set that include the following:

- Identify and discard correlated variables before finalizing on important variables
- The variables could be selected based on 'p' values from Linear Regression
- Forward, Backward, and Stepwise selection
- Lasso Regression
- Random Forest and plot variable chart
- Top features can be selected based on information gain for the available set of features.

**6. There are many machine learning algorithms till now. If given a data set, how can one determine which algorithm to be used for that?**

Machine Learning algorithm to be used purely depends on the type of data in a given dataset. If data is linear then, we use linear regression. If data shows non-linearity then, the bagging algorithm would do better. If the data is to be analyzed/interpreted for some business purposes then we can use decision trees or SVM. If the dataset consists of images, videos, audios then, neural networks would be helpful to get the solution accurately.

So, there is no certain metric to decide which algorithm to be used for a given situation or a data set. We need to explore the data using EDA (Exploratory Data Analysis) and understand the purpose of using the dataset to come up with the best fit algorithm. So, it is important to study all the algorithms in detail.

**7. How are covariance and correlation different from one another?**

| **Covariance** | **Correlation** |
|---|---|
| Covariance measures how two variables are related to each other and how one would vary with respect to changes in the other variable. If the value is positive it means there is a direct relationship between the variables and one would increase or decrease with an increase or decrease in the base variable respectively, given that all other conditions remain constant. | Correlation quantifies the relationship between two random variables and has only three specific values, i.e., 1, 0, and -1. |

1 denotes a positive relationship, -1 denotes a negative relationship, and 0 denotes that the two variables are independent of each other.

**8. State the differences between causality and correlation?**

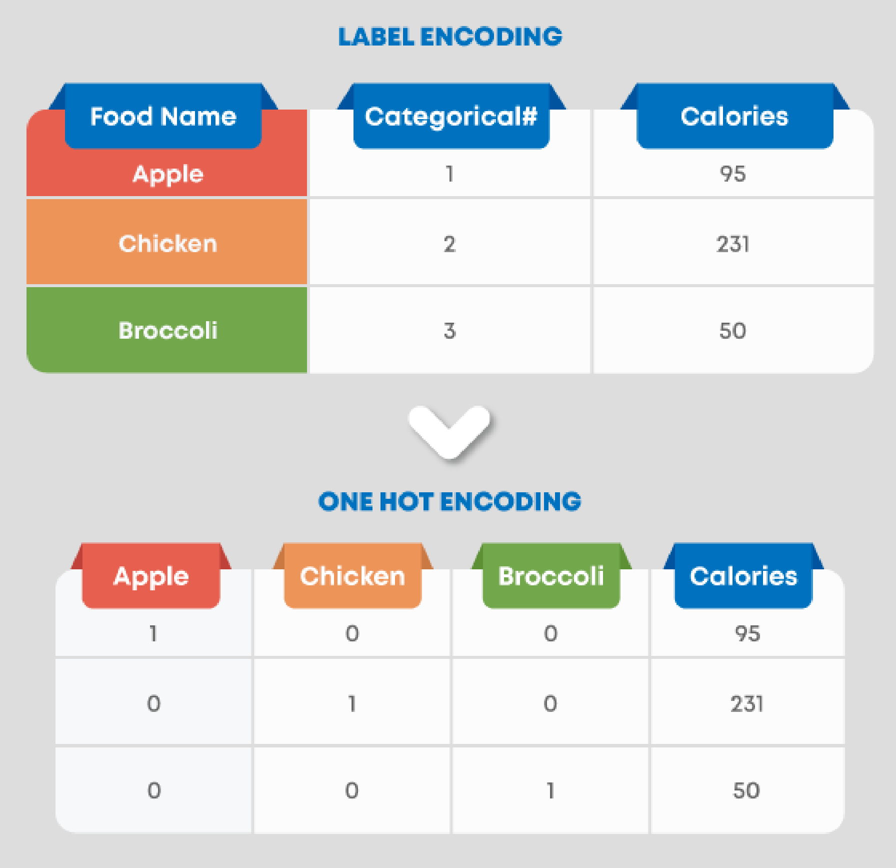
Causality applies to situations where one action, say X, causes an outcome, say Y, whereas Correlation is just relating one action (X) to another action(Y) but X does not necessarily cause Y.

**9. We look at machine learning software almost all the time. How do we apply Machine Learning to Hardware?**

We have to build ML algorithms in System Verilog which is a Hardware development Language and then program it onto an FPGA to apply Machine Learning to hardware.

**10. Explain One-hot encoding and Label Encoding. How do they affect the dimensionality of the given dataset?**

One-hot encoding is the representation of categorical variables as binary vectors. Label Encoding is converting labels/words into numeric form. Using one-hot encoding increases the dimensionality of the data set. Label encoding doesn’t affect the dimensionality of the data set. One-hot encoding creates a new variable for each level in the variable whereas, in Label encoding, the levels of a variable get encoded as 1 and 0.



LABEL ENCODING

| Food Name | Categorical# | Calories |
|---|---|---|
| Apple | 1 | 95 |
| Chicken | 2 | 231 |
| Broccoli | 3 | 50 |

ONE HOT ENCODING

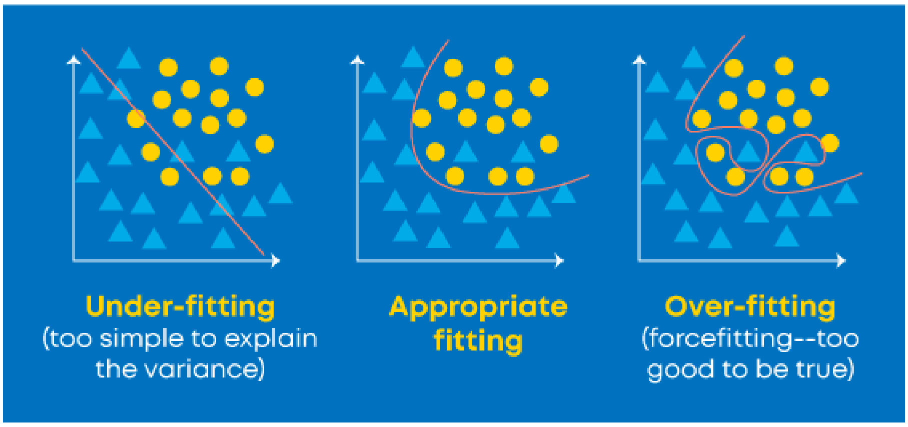
| Apple | Chicken | Broccoli | Calories |
|---|---|---|---|
| 1 | 0 | 0 | 95 |
| 0 | 1 | 0 | 231 |
| 0 | 0 | 1 | 50 |

**Deep Learning Interview Questions**

Deep Learning is a part of machine learning that works with neural networks. It involves a hierarchical structure of networks that set up a process to help machines learn the human logic behind any action. We have compiled a list of the frequently asked deep learning interview questions to help you prepare.

**11. When does regularization come into play in Machine Learning?**

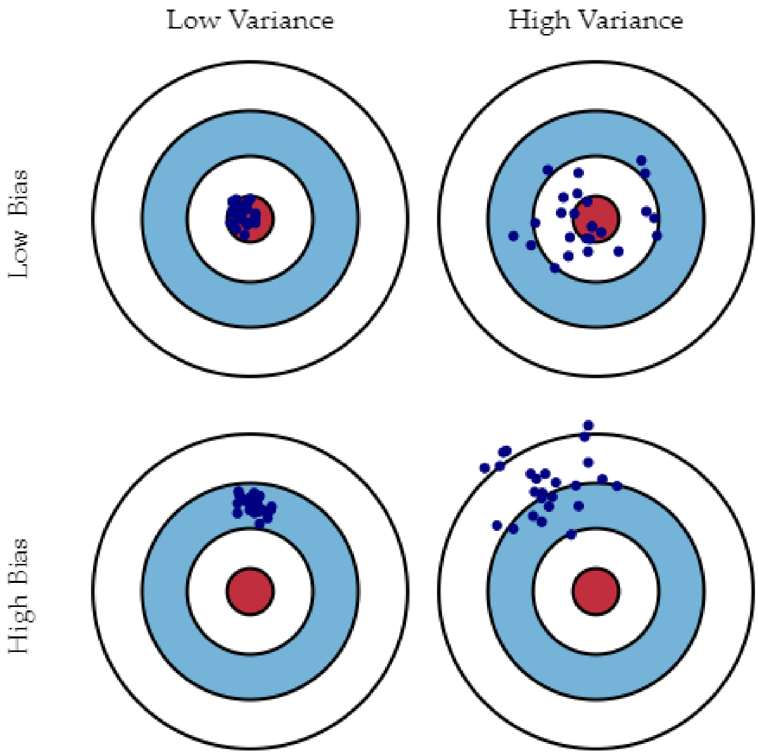
At times when the model begins to underfit or overfit, regularization becomes necessary. It is a regression that diverts or regularizes the coefficient estimates towards zero. It reduces flexibility and discourages learning in a model to avoid the risk of overfitting. The model complexity is reduced and it becomes better at predicting.


Under-fitting
(too simple to explain the variance)
Appropriate fitting
Over-fitting
(forcefitting--too good to be true)

**12. What is Bias, Variance and what do you mean by Bias-Variance Tradeoff?**

Both are errors in Machine Learning Algorithms. When the algorithm has limited flexibility to deduce the correct observation from the dataset, it results in bias. On the other hand, variance occurs when the model is extremely sensitive to small fluctuations.

If one adds more features while building a model, it will add more complexity and we will lose bias but gain some variance. In order to maintain the optimal amount of error, we perform a tradeoff between bias and variance based on the needs of a business.



Bias stands for the error because of the erroneous or overly simplistic assumptions in the learning algorithm . This assumption can lead to the model underfitting the data, making it hard for it to have high predictive accuracy and for you to generalize your knowledge from the training set to the test set.

Variance is also an error because of  too much complexity in the learning algorithm. This can be the reason for the algorithm being highly sensitive to high degrees of variation in training data, which can lead your model to overfit the data. Carrying too much noise from the training data for your model to be very useful for your test data.

The bias-variance decomposition essentially decomposes the learning error from any algorithm by adding the bias, the variance and a bit of irreducible error due to noise in the underlying dataset. Essentially, if you make the model more complex and add more variables, you'll lose bias but gain some variance — in order to get the optimally reduced amount of error, you'll have to trade off bias and variance. You don't want either high bias or high variance in your model.

**13. How can we relate standard deviation and variance?**

*Standard deviation* refers to the spread of your data from the mean. *Variance* is the average degree to which each point differs from the mean i.e. the average of all data points. We can relate Standard deviation and Variance because it is the square root of Variance.

**14. A data set is given to you and it has missing values which spread along 1 standard deviation from the mean. How much of the data would remain untouched?**

It is given that the data is spread across mean that is the data is spread across an average. So, we can presume that it is a normal distribution. In a normal distribution, about 68% of data lies in 1 standard deviation from averages like mean, mode or median. That means about 32% of the data remains uninfluenced by missing values.

**15. Is a high variance in data good or bad?**

Higher variance directly means that the data spread is big and the feature has a variety of data. Usually, high variance in a feature is seen as not so good quality.

**16. If your dataset is suffering from high variance, how would you handle it?**

For datasets with high variance, we could use the bagging algorithm to handle it. Bagging algorithm splits the data into subgroups with sampling replicated from random data. After the data is split, random data is used to create rules using a training algorithm. Then we use polling technique to combine all the predicted outcomes of the model.

**17. A data set is given to you about utilities fraud detection. You have built aclassifier model and achieved a performance score of 98.5%. Is this a goodmodel? If yes, justify. If not, what can you do about it?**

Data set about utilities fraud detection is not balanced enough i.e. imbalanced. In such a data set, accuracy score cannot be the measure of performance as it may only be predict the majority class label correctly but in this case our point of interest is to predict the minority label. But often minorities are treated as noise and ignored. So, there is a high probability of misclassification of the minority label as compared to the majority label. For evaluating the model performance in case of imbalanced data sets, we should use Sensitivity (True Positive rate) or Specificity (True Negative rate) to determine class label wise performance of the classification model. If the minority class label's performance is not so good, we could do the following:

- We can use under sampling or over sampling to balance the data.
- We can change the prediction threshold value.
- We can assign weights to labels such that the minority class labels get larger weights.
- We could detect anomalies.

**18. Explain the handling of missing or corrupted values in the given dataset.**

An easy way to handle missing values or corrupted values is to drop the corresponding rows or columns. If there are too many rows or columns to drop then we consider replacing the missing or corrupted values with some new value.

Identifying missing values and dropping the rows or columns can be done by using IsNull() and dropna( ) functions in Pandas. Also, the Fillna() function in Pandas replaces the incorrect values with the placeholder value.

**19. What is Time series?**

A Time series is a sequence of numerical data points in successive order. It tracks the movement of the chosen data points, over a specified period of time and records the data points at regular intervals. Time series doesn't require any minimum or maximum time input. Analysts often use Time series to examine data according to their specific requirement.

**20. What is a Box-Cox transformation?**

Box-Cox transformation is a power transform which transforms non-normal dependent variables into normal variables as normality is the most common assumption made while using many statistical techniques. It has a lambda parameter which when set to 0 implies that this transform is equivalent to log-transform. It is used for variance stabilization and also to normalize the distribution.

**21. What is the difference between stochastic gradient descent (SGD) and gradient descent (GD)?**

Gradient Descent and Stochastic Gradient Descent are the algorithms that find the set of parameters that will minimize a loss function.
The difference is that in Gradient Descend, all training samples are evaluated for each set of parameters. While in Stochastic Gradient Descent only one training sample is evaluated for the set of parameters identified.

**22. What is the exploding gradient problem while using the back propagation technique?**

When large error gradients accumulate and result in large changes in the neural network weights during training, it is called the exploding gradient problem. The values of weights can become so large as to overflow and result in NaN values. This makes the model unstable and the learning of the model to stall just like the vanishing gradient problem. This is one of the most commonly asked interview questions on machine learning.

### 23. Can you mention some advantages and disadvantages of decision trees?

The advantages of decision trees are that they are easier to interpret, are nonparametric and hence robust to outliers, and have relatively few parameters to tune. On the other hand, the disadvantage is that they are prone to overfitting.

### 24. Explain the differences between Random Forest and Gradient Boosting machines.

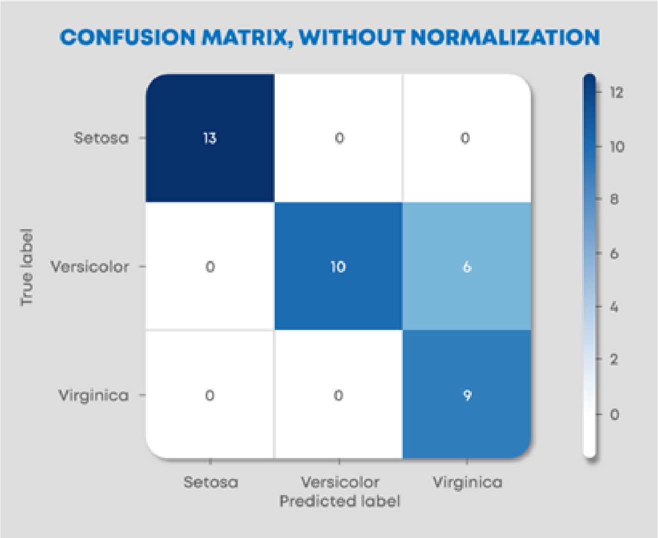
| Random Forests | Gradient Boosting |
|---|---|
| Random forests are a significant number of decision trees pooled using averages or majority rules at the end. | Gradient boosting machines also combine decision trees but at the beginning of the process, unlike Random forests. |
| The random forest creates each tree independent of the others while gradient boosting develops one tree at a time. | Gradient boosting yields better outcomes than random forests if parameters are carefully tuned but it’s not a good option if the data set contains a lot of outliers/anomalies/noise as it can result in overfitting of the model. |
| Random forests perform well for multiclass object detection. | Gradient Boosting performs well when there is data which is not balanced such as in real-time risk assessment. |

### 25. What is a confusion matrix and why do you need it?

Confusion matrix (also called the error matrix) is a table that is frequently used to illustrate the performance of a classification model i.e. classifier on a set of test data for which the true values are well-known.

It allows us to visualize the performance of an algorithm/model. It allows us to easily identify the confusion between different classes. It is used as a performance measure of a model/algorithm.

A confusion matrix is known as a summary of predictions on a classification model. The number of right and wrong predictions were summarized with count values and broken down by each class label. It gives us information about the errors made through the classifier and also the types of errors made by a classifier.



**26. What’s a Fourier transform?**

Fourier Transform is a mathematical technique that transforms any function of time to a function of frequency. Fourier transform is closely related to Fourier series. It takes any time-based pattern for input and calculates the overall cycle offset, rotation speed and strength for all possible cycles. Fourier transform is best applied to waveforms since it has functions of time and space. Once a Fourier transform applied on a waveform, it gets decomposed into a sinusoid.

**27. What do you mean by Associative Rule Mining (ARM)?**

Associative Rule Mining is one of the techniques to discover patterns in data like features (dimensions) which occur together and features (dimensions) which are correlated. It is mostly used in Market-based Analysis to find how frequently an itemset occurs in a transaction. Association rules have to satisfy minimum support and minimum confidence at the very same time. Association rule generation generally comprised of two different steps:

- “A min support threshold is given to obtain all frequent item-sets in a database.”
- “A min confidence constraint is given to these frequent item-sets in order to form the association rules.”

Support is a measure of how often the “item set” appears in the data set and Confidence is a measure of how often a particular rule has been found to be true.

**28. What is Marginalisation? Explain the process.**

Marginalisation is summing the probability of a random variable X given joint probability distribution of X with other variables. It is an application of the law of total probability.

$P(X=x) = \sum_{Y} P(X=x,Y)$

Given the joint probability P(X=x,Y), we can use marginalization to find P(X=x). So, it is to find distribution of one random variable by exhausting cases on other random variables.

**29. Explain the phrase "Curse of Dimensionality".**

The Curse of Dimensionality refers to the situation when your data has too many features.

The phrase is used to express the difficulty of using brute force or grid search to optimize a function with too many inputs.

It can also refer to several other issues like:

- If we have more features than observations, we have a risk of overfitting the model.
- When we have too many features, observations become harder to cluster. Too many dimensions cause every observation in the dataset to appear equidistant from all others and no meaningful clusters can be formed.

Dimensionality reduction techniques like PCA come to the rescue in such cases.

**30. What is the Principle Component Analysis?**

The idea here is to reduce the dimensionality of the data set by reducing the number of variables that are correlated with each other. Although the variation needs to be retained to the maximum extent.

The variables are transformed into a new set of variables that are known as Principal Components'. These PCs are the eigenvectors of a covariance matrix and therefore are orthogonal.

**31. Why is rotation of components so important in Principle Component Analysis (PCA)?**

Rotation in PCA is very important as it maximizes the separation within the variance obtained by all the components because of which interpretation of components would become easier. If the components are not rotated, then we need extended components to describe variance of the components.

**32. What are outliers? Mention three methods to deal with outliers.**

A data point that is considerably distant from the other similar data points is known as an outlier. They may occur due to experimental errors or variability in measurement. They are problematic and can mislead a training process, which eventually results in longer training time, inaccurate models, and poor results.

The three methods to deal with outliers are:
**Univariate method** – looks for data points having extreme values on a single variable
**Multivariate method** – looks for unusual combinations on all the variables
**Minkowski error** – reduces the contribution of potential outliers in the training process

**33. What is the difference between regularization and normalisation?**

| **Normalisation** | **Regularisation** |
|---|---|

| | |
|---|---|
| Normalisation adjusts the data; . If your data is on very different scales (especially low to high), you would want to normalise the data. Alter each column to have compatible basic statistics. This can be helpful to make sure there is no loss of accuracy. One of the goals of model training is to identify the signal and ignore the noise if the model is given free rein to minimize error, there is a possibility of suffering from overfitting. | Regularisation adjusts the prediction function. Regularization imposes some control on this by providing simpler fitting functions over complex ones. |

**34. Explain the difference between Normalization and Standardization.**

Normalization and Standardization are the two very popular methods used for feature scaling.

| **Normalisation** | **Standardization** |
|---|---|
| Normalization refers to re-scaling the values to fit into a range of [0,1]. Normalization is useful when all parameters need to have an identical positive scale however the outliers from the data set are lost. | Standardization refers to re-scaling data to have a mean of 0 and a standard deviation of 1 (Unit variance) |

**35. List the most popular distribution curves along with scenarios where you will use them in an algorithm.**

The most popular distribution curves are as follows- Bernoulli Distribution, Uniform Distribution, Binomial Distribution, Normal Distribution, Poisson Distribution, and Exponential Distribution.
Each of these distribution curves is used in various scenarios.

Bernoulli Distribution can be used to check if a team will win a championship or not, a newborn child is either male or female, you either pass an exam or not, etc.

***Uniform distribution*** is a probability distribution that has a constant probability. Rolling a single dice is one example because it has a fixed number of outcomes.

***Binomial distribution*** is a probability with only two possible outcomes, the prefix 'bi' means two or twice. An example of this would be a coin toss. The outcome will either be heads or tails.

***Normal distribution*** describes how the values of a variable are distributed. It is typically a symmetric distribution where most of the observations cluster around the central peak. The values further away from the mean taper off equally in both directions. An example would be the height of students in a classroom.

***Poisson distribution*** helps predict the probability of certain events happening when you know how often that event has occurred. It can be used by businessmen to make forecasts about the number of customers on certain days and allows them to adjust supply according to the demand.

***Exponential distribution*** is concerned with the amount of time until a specific event occurs. For example, how long a car battery would last, in months.

**36. How do we check the normality of a data set or a feature?**

Visually, we can check it using plots. There is a list of Normality checks, they are as follow:

- Shapiro-Wilk W Test
- Anderson-Darling Test
- Martinez-Iglewicz Test
- Kolmogorov-Smirnov Test
- D’Agostino Skewness Test

**37. What is Linear Regression?**

Linear Function can be defined as a Mathematical function on a 2D plane as, Y =Mx +C, where Y is a dependent variable and X is Independent Variable, C is Intercept and M is slope and same can be expressed as Y is a Function of X or Y = F(x).

At any given value of X, one can compute the value of Y, using the equation of Line. This relation between Y and X, with a degree of the polynomial as 1 is called Linear Regression.

In Predictive Modeling, LR is represented as Y = Bo + B1x1 + B2x2
The value of B1 and B2 determines the strength of the correlation between features and the dependent variable.

Example: Stock Value in $ = Intercept + (+/-B1)*(Opening value of Stock) + (+/-B2)*(Previous Day Highest value of Stock)

**38. Differentiate between regression and classification.**

Regression and classification are categorized under the same umbrella of supervised machine learning. The main difference between them is that the output variable in the regression is numerical (or continuous) while that for classification is categorical (or discrete).

Example: To predict the definite Temperature of a place is Regression problem whereas predicting whether the day will be Sunny cloudy or there will be rain is a case of classification.

**39. What is target imbalance? How do we fix it? A scenario where you have performed target imbalance on data. Which metrics and algorithms do you find suitable to input this data onto?**

If you have categorical variables as the target when you cluster them together or perform a frequency count on them if there are certain categories which are more in number as compared to others by a very significant number. This is known as the target imbalance.

Example: Target column – 0,0,0,1,0,2,0,0,1,1 [0s: 60%, 1: 30%, 2:10%] 0 are in majority. To fix this, we can perform up-sampling or down-sampling. Before fixing this problem let’s assume that the performance metrics used was confusion metrics. After fixing this problem we can shift the metric system to AUC: ROC. Since we added/deleted data [up sampling or downsampling], we can go ahead with a stricter algorithm like SVM, Gradient boosting or ADA boosting.

**40. List all assumptions for data to be met before starting with linear regression.**

Before starting linear regression, the assumptions to be met are as follow:

- Linear relationship
- Multivariate normality
- No or little multicollinearity
- No auto-correlation
- Homoscedasticity

**41. When does the linear regression line stop rotating or finds an optimal spot where it is fitted on data?**

A place where the highest RSquared value is found, is the place where the line comes to rest. RSquared represents the amount of variance captured by the virtual linear regression line with respect to the total variance captured by the dataset.

**42. Why is logistic regression a type of classification technique and not a regression? Name the function it is derived from?**

Since the target column is categorical, it uses linear regression to create an odd function that is wrapped with a log function to use regression as a classifier. Hence, it is a type of classification technique and not a regression. It is derived from cost function.

**43. What could be the issue when the beta value for a certain variable varies way too much in each subset when regression is run on different subsets of the given dataset?**

Variations in the beta values in every subset implies that the dataset is heterogeneous. To overcome this problem, we can use a different model for each of the clustered subsets of the dataset or use a non-parametric model such as decision trees.

**44. What does the term Variance Inflation Factor mean?**

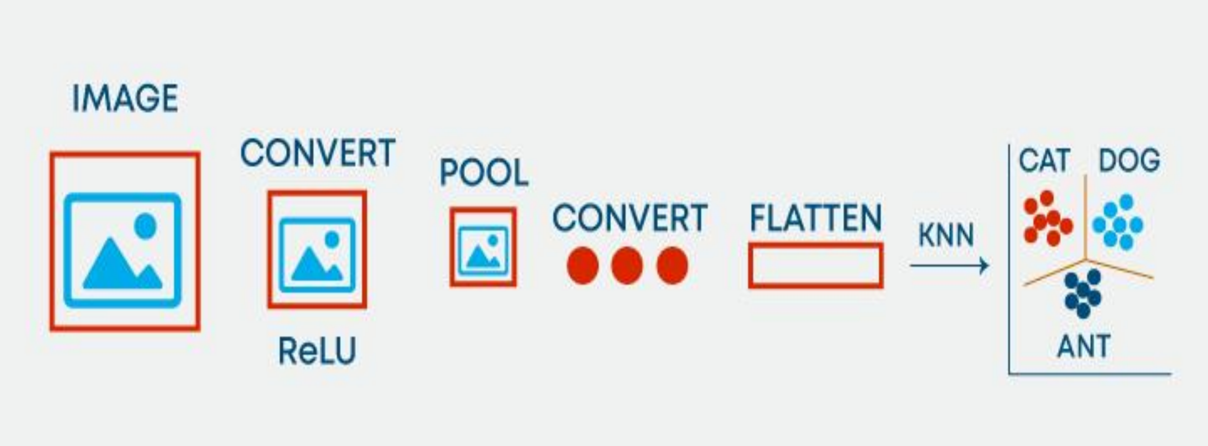
Variation Inflation Factor (VIF) is the ratio of the variance of the model to the variance of the model with only one independent variable. VIF gives the estimate of the volume of multicollinearity in a set of many regression variables.

VIF = Variance of model with one independent variable

**45. Which machine learning algorithm is known as the lazy learner and why is it called so?**

KNN is a Machine Learning algorithm known as a lazy learner. K-NN is a lazy learner because it doesn’t learn any machine learnt values or variables from the training data but dynamically calculates distance every time it wants to classify, hence memorising the training dataset instead.

**46. Is it possible to use KNN for image processing?**



Yes, it is possible to use KNN for image processing. It can be done by converting the 3-dimensional image into a single-dimensional vector and using the same as input to KNN.

**47. Differentiate between K-Means and KNN algorithms?**

| **KNN algorithms** | **K-Means** |
|---|---|
| KNN algorithms is Supervised Learning where-as K-Means is Unsupervised Learning. With KNN, we predict the label of the unidentified element based on its nearest neighbour and further extend this approach for solving classification/regression-based problems. | K-Means is Unsupervised Learning, where we don't have any Labels present, in other words, no Target Variables and thus we try to cluster the data based upon their coord |

**48. How does the SVM algorithm deal with self-learning?**

SVM has a learning rate and expansion rate which takes care of this. The learning rate compensates or penalises the hyperplanes for making all the wrong moves and expansion rate deals with finding the maximum separation area between classes.

**49. What are Kernels in SVM? List popular kernels used in SVM along with a scenario of their applications.**

The function of the kernel is to take data as input and transform it into the required form. A few popular Kernels used in SVM are as follows: RBF, Linear, Sigmoid, Polynomial, Hyperbolic, Laplace, etc.

**50. What is Kernel Trick in an SVM Algorithm?**

Kernel Trick is a mathematical function which when applied on data points, can find the region of classification between two different classes. Based on the choice of function, be it linear or radial, which purely depends upon the distribution of data, one can build a classifier.

**51. What are ensemble models? Explain how ensemble techniques yield better learning as compared to traditional classification ML algorithms.**

An ensemble is a group of models that are used together for prediction both in classification and regression classes. Ensemble learning helps improve ML results because it combines several models. By doing so, it allows for a better predictive performance compared to a single model.
They are superior to individual models as they reduce variance, average out biases, and have lesser chances of overfitting.

**52. What are overfitting and underfitting? Why does the decision tree algorithm suffer often with overfitting problems?**

Overfitting is a statistical model or machine learning algorithm which captures the noise of the data. Underfitting is a model or machine learning algorithm which does not fit the data well enough and occurs if the model or algorithm shows low variance but high bias.

In decision trees, overfitting occurs when the tree is designed to perfectly fit all samples in the training data set. This results in branches with strict rules or sparse data and affects the accuracy when predicting samples that aren’t part of the training set.

**53. What is OOB error and how does it occur?**

For each bootstrap sample, there is one-third of data that was not used in the creation of the tree, i.e., it was out of the sample. This data is referred to as out of bag data. In order to get an unbiased measure of the accuracy of the model over test data, out of bag error is used. The out of bag data is passed for each tree is passed through that tree and the outputs are aggregated to give out of bag error. This percentage error is quite effective in estimating the error in the testing set and does not require further cross-validation.

**54. Why boosting is a more stable algorithm as compared to other ensemble algorithms?**

Boosting focuses on errors found in previous iterations until they become obsolete. Whereas in bagging there is no corrective loop. This is why boosting is a more stable algorithm compared to other ensemble algorithms.

**55. How do you handle outliers in the data?**

Outlier is an observation in the data set that is far away from other observations in the data set. We can discover outliers using tools and functions like box plot, scatter plot, Z-Score, IQR score etc. and then handle them based on the visualization we have got. To handle outliers, we can cap at some threshold, use transformations to reduce skewness of the data and remove outliers if they are anomalies or errors.

**56. List popular cross validation techniques.**

There are mainly six types of cross validation techniques. They are as follow:

- K fold
- Stratified k fold

- Leave one out
- Bootstrapping
- Random search cv
- Grid search cv

**57. Is it possible to test for the probability of improving model accuracy without cross-validation techniques? If yes, please explain.**

Yes, it is possible to test for the probability of improving model accuracy without cross-validation techniques. We can do so by running the ML model for say **n** number of iterations, recording the accuracy. Plot all the accuracies and remove the 5% of low probability values. Measure the left [low] cut off and right [high] cut off. With the remaining 95% confidence, we can say that the model can go as low or as high [as mentioned within cut off points].

**58. Name a popular dimensionality reduction algorithm.**

Popular dimensionality reduction algorithms are Principal Component Analysis and Factor Analysis.
Principal Component Analysis creates one or more index variables from a larger set of measured variables. Factor Analysis is a model of the measurement of a latent variable. This latent variable cannot be measured with a single variable and is seen through a relationship it causes in a set of **y** variables.

**59. How can we use a dataset without the target variable into supervised learning algorithms?**

Input the data set into a clustering algorithm, generate optimal clusters, label the cluster numbers as the new target variable. Now, the dataset has independent and target variables present. This ensures that the dataset is ready to be used in supervised learning algorithms.

**60. List all types of popular recommendation systems? Name and explain two personalized recommendation systems along with their ease of implementation.**

Popularity based recommendation, content-based recommendation, user-based collaborative filter, and item-based recommendation are the popular types of recommendation systems. Personalised Recommendation systems are- Content-based recommendation, user-based collaborative filter, and item-based recommendation. User-based collaborative filter and item-based recommendations are more personalised. Ease to maintain: Similarity matrix can be maintained easily with Item-based recommendation.

**61. How do we deal with sparsity issues in recommendation systems? How do we measure its effectiveness? Explain.**

Singular value decomposition can be used to generate the prediction matrix. RMSE is the measure that helps us understand how close the prediction matrix is to the original matrix.

**62. Name and define techniques used to find similarities in the recommendation system.**

Pearson correlation and Cosine correlation are techniques used to find similarities in recommendation systems.

### 63. State the limitations of Fixed Basis Function.

Linear separability in feature space doesn't imply linear separability in input space. So, Inputs are non-linearly transformed using vectors of basic functions with increased dimensionality. Limitations of Fixed basis functions are:

- Non-Linear transformations cannot remove overlap between two classes but they can increase overlap.
- Often it is not clear which basis functions are the best fit for a given task. So, learning the basic functions can be useful over using fixed basis functions.
- If we want to use only fixed ones, we can use a lot of them and let the model figure out the best fit but that would lead to overfitting the model thereby making it unstable.

### 64. Define and explain the concept of Inductive Bias with some examples.

Inductive Bias is a set of assumptions that humans use to predict outputs given inputs that the learning algorithm has not encountered yet. When we are trying to learn Y from X and the hypothesis space for Y is infinite, we need to reduce the scope by our beliefs/assumptions about the hypothesis space which is also called inductive bias. Through these assumptions, we constrain our hypothesis space and also get the capability to incrementally test and improve on the data using hyper-parameters. Examples:

1. We assume that Y varies linearly with X while applying Linear regression.
2. We assume that there exists a hyperplane separating negative and positive examples.

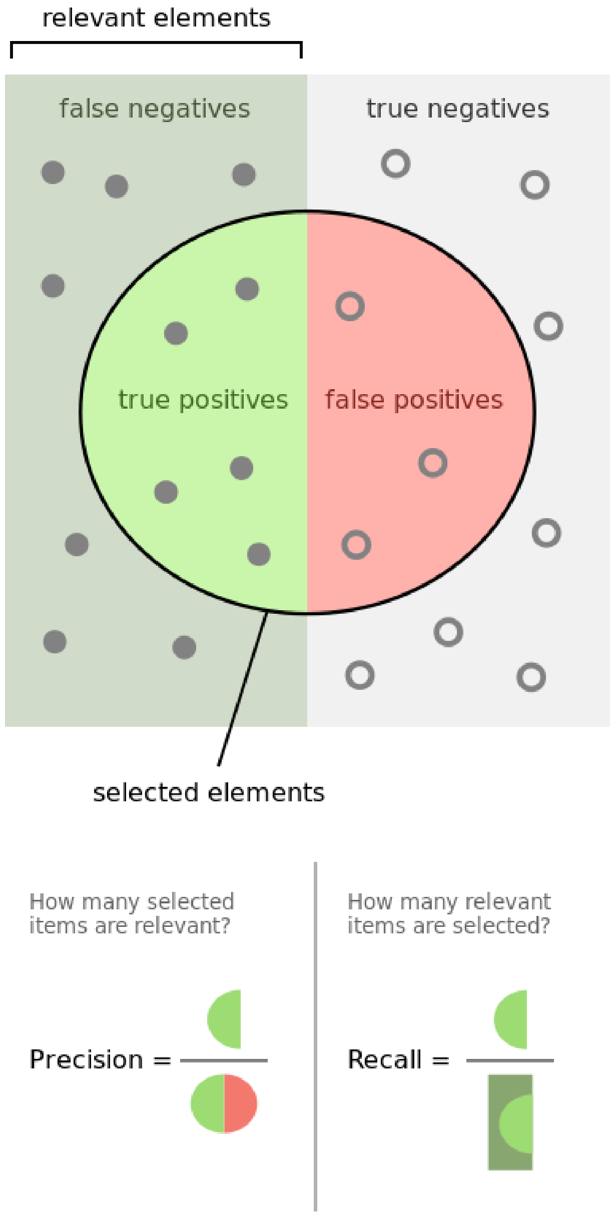
### 65. Explain the term instance-based learning.

Instance Based Learning is a set of procedures for regression and classification which produce a class label prediction based on resemblance to its nearest neighbors in the training data set. These algorithms just collects all the data and get an answer when required or queried. In simple words they are a set of procedures for solving new problems based on the solutions of already solved problems in the past which are similar to the current problem.

### 66. Keeping train and test split criteria in mind, is it good to perform scaling before the split or after the split?

Scaling should be done post-train and test split ideally. If the data is closely packed, then scaling post or pre-split should not make much difference.

### 67. Define precision, recall and F1 Score?



The metric used to access the performance of the classification model is Confusion Metric. Confusion Metric can be further interpreted with the following terms:-

**True Positives (TP)** – These are the correctly predicted positive values. It implies that the value of the actual class is yes and the value of the predicted class is also yes.

**True Negatives (TN)** – These are the correctly predicted negative values. It implies that the value of the actual class is no and the value of the predicted class is also no.

**False positives and false negatives**, these values occur when your actual class contradicts with the predicted class.

**Now,**
**Recall,** also known as Sensitivity is the ratio of true positive rate (TP), to all observations in

actual class – yes
Recall = TP/(TP+FN)

**Precision** is the ratio of positive predictive value, which measures the amount of accurate positives model predicted viz a viz number of positives it claims.
Precision = TP/(TP+FP)

**Accuracy** is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.
Accuracy = (TP+TN)/(TP+FP+FN+TN)

**F1 Score** is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have a similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

**68. Plot validation score and training score with data set size on the x-axis and another plot with model complexity on the x-axis.**

For high bias in the models, the performance of the model on the validation data set is similar to the performance on the training data set. For high variance in the models, the performance of the model on the validation set is worse than the performance on the training set.

**69. What is Bayes' Theorem? State at least 1 use case with respect to the machine learning context?**

Bayes' Theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. For example, if cancer is related to age, then, using Bayes' theorem, a person's age can be used to more accurately assess the probability that they have cancer than can be done without the knowledge of the person's age.
Chain rule for Bayesian probability can be used to predict the likelihood of the next word in the sentence.

**70. What is Naive Bayes? Why is it Naive?**

Naive Bayes classifiers are a series of classification algorithms that are based on the Bayes theorem. This family of algorithm shares a common principle which treats every pair of features independently while being classified.

Naive Bayes is considered Naive because the attributes in it (for the class) is independent of others in the same class. This lack of dependence between two attributes of the same class creates the quality of naiveness.

71. Explain how a Naive Bayes Classifier works.

Naive Bayes classifiers are a family of algorithms which are derived from the Bayes theorem of probability. It works on the fundamental assumption that every set of two features that is being classified is independent of each other and every feature makes an equal and independent contribution to the outcome.

**72. What do the terms prior probability and marginal likelihood in context of Naive Bayes theorem mean?**

Prior probability is the percentage of dependent binary variables in the data set. If you are given a dataset and dependent variable is either 1 or 0 and percentage of 1 is 65% and percentage of 0 is 35%. Then, the probability that any new input for that variable of being 1 would be 65%.

Marginal likelihood is the denominator of the Bayes equation and it makes sure that the posterior probability is valid by making its area 1.

**73. Explain the difference between Lasso and Ridge?**

Lasso(L1) and Ridge(L2) are the regularization techniques where we penalize the coefficients to find the optimum solution. In ridge, the penalty function is defined by the sum of the squares of the coefficients and for the Lasso, we penalize the sum of the absolute values of the coefficients. Another type of regularization method is ElasticNet, it is a hybrid penalizing function of both lasso and ridge.

**74. What's the difference between probability and likelihood?**

Probability is the measure of the likelihood that an event will occur that is, what is the certainty that a specific event will occur? Where-as a likelihood function is a function of parameters within the parameter space that describes the probability of obtaining the observed data. So the fundamental difference is, Probability attaches to possible results; likelihood attaches to hypotheses.

**75. Why would you Prune your tree?**

In the context of data science or AIML, pruning refers to the process of reducing redundant branches of a decision tree. Decision Trees are prone to overfitting, pruning the tree helps to reduce the size and minimizes the chances of overfitting. Pruning involves turning branches of a decision tree into leaf nodes and removing the leaf nodes from the original branch. It serves as a tool to perform the tradeoff.

**76. Model accuracy or Model performance? Which one will you prefer and why?**

This is a trick question, one should first get a clear idea, what is Model Performance? If Performance means speed, then it depends upon the nature of the application, any application related to the real-time scenario will need high speed as an important feature. Example: The best of Search Results will lose its virtue if the Query results do not appear fast.

If Performance is hinted at Why Accuracy is not the most important virtue – For any imbalanced data set, more than Accuracy, it will be an F1 score than will explain the business case and in case data is imbalanced, then Precision and Recall will be more important than rest.

**77. List the advantages and limitations of the Temporal Difference Learning Method.**

Temporal Difference Learning Method is a mix of Monte Carlo method and Dynamic programming method. Some of the advantages of this method include:

- It can learn in every step online or offline.
- It can learn from a sequence which is not complete as well.
- It can work in continuous environments.
- It has lower variance compared to MC method and is more efficient than MC method.

*Limitations of TD method are:*

- It is a biased estimation.
- It is more sensitive to initialization.

**78. How would you handle an imbalanced dataset?**

Sampling Techniques can help with an imbalanced dataset. There are two ways to perform sampling, Under Sample or Over Sampling.

In Under Sampling, we reduce the size of the majority class to match minority class thus help by improving performance w.r.t storage and run-time execution, but it potentially discards useful information.

For Over Sampling, we upsample the Minority class and thus solve the problem of information loss, however, we get into the trouble of having Overfitting.

There are other techniques as well –
**Cluster-Based Over Sampling** – In this case, the K-means clustering algorithm is independently applied to minority and majority class instances. This is to identify clusters in the dataset. Subsequently, each cluster is oversampled such that all clusters of the same class have an equal number of instances and all classes have the same size

**Synthetic Minority Over-sampling Technique (SMOTE) –** A subset of data is taken from the minority class as an example and then new synthetic similar instances are created which are then added to the original dataset. This technique is good for Numerical data points.

**79. Mention some of the EDA Techniques?**

Exploratory Data Analysis (EDA) helps analysts to understand the data better and forms the foundation of better models.

**Visualization**

- Univariate visualization
- Bivariate visualization
- Multivariate visualization

**Missing Value Treatment** – Replace missing values with Either Mean/Median

**Outlier Detection** – Use Boxplot to identify the distribution of Outliers, then Apply IQR to set the boundary for IQR

**Transformation** – Based on the distribution, apply a transformation on the features

**Scaling the Dataset** – Apply MinMax, Standard Scaler or Z Score Scaling mechanism to scale the data.

**Feature Engineering** – Need of the domain, and SME knowledge helps Analyst find derivative fields which can fetch more information about the nature of the data

**Dimensionality reduction** — Helps in reducing the volume of data without losing much information

**80. Mention why feature engineering is important in model building and list out some of the techniques used for feature engineering.**

Algorithms necessitate features with some specific characteristics to work appropriately. The data is initially in a raw form. You need to extract features from this data before supplying it to the algorithm. This process is called feature engineering. When you have relevant features, the complexity of the algorithms reduces. Then, even if a non-ideal algorithm is used, results come out to be accurate.

Feature engineering primarily has two goals:

- Prepare the suitable input data set to be compatible with the machine learning algorithm constraints.
- Enhance the performance of machine learning models.

Some of the techniques used for feature engineering include Imputation, Binning, Outliers Handling, Log transform, grouping operations, One-Hot encoding, Feature split, Scaling, Extracting date.

**81. Differentiate between Statistical Modeling and Machine Learning?**

Machine learning models are about making accurate predictions about the situations, like Foot Fall in restaurants, Stock-Price, etc. where-as, Statistical models are designed for inference about the relationships between variables, as What drives the sales in a restaurant, is it food or Ambience.

**82. Differentiate between Boosting and Bagging?**

Bagging and Boosting are variants of Ensemble Techniques.

**Bootstrap Aggregation or bagging** is a method that is used to reduce the variance for algorithms having very high variance. Decision trees are a particular family of classifiers which are susceptible to having high bias.

Decision trees have a lot of sensitiveness to the type of data they are trained on. Hence generalization of results is often much more complex to achieve in them despite very high fine-tuning. The results vary greatly if the training data is changed in decision trees.

Hence bagging is utilised where multiple decision trees are made which are trained on samples of the original data and the final result is the average of all these individual models.

**Boosting** is the process of using an n-weak classifier system for prediction such that every weak classifier compensates for the weaknesses of its classifiers. By weak classifier, we imply a classifier which performs poorly on a given data set.

It’s evident that boosting is not an algorithm rather it’s a process. Weak classifiers used are generally logistic regression, shallow decision trees etc.

There are many algorithms which make use of boosting processes but two of them are mainly used: Adaboost and Gradient Boosting and XGBoost.

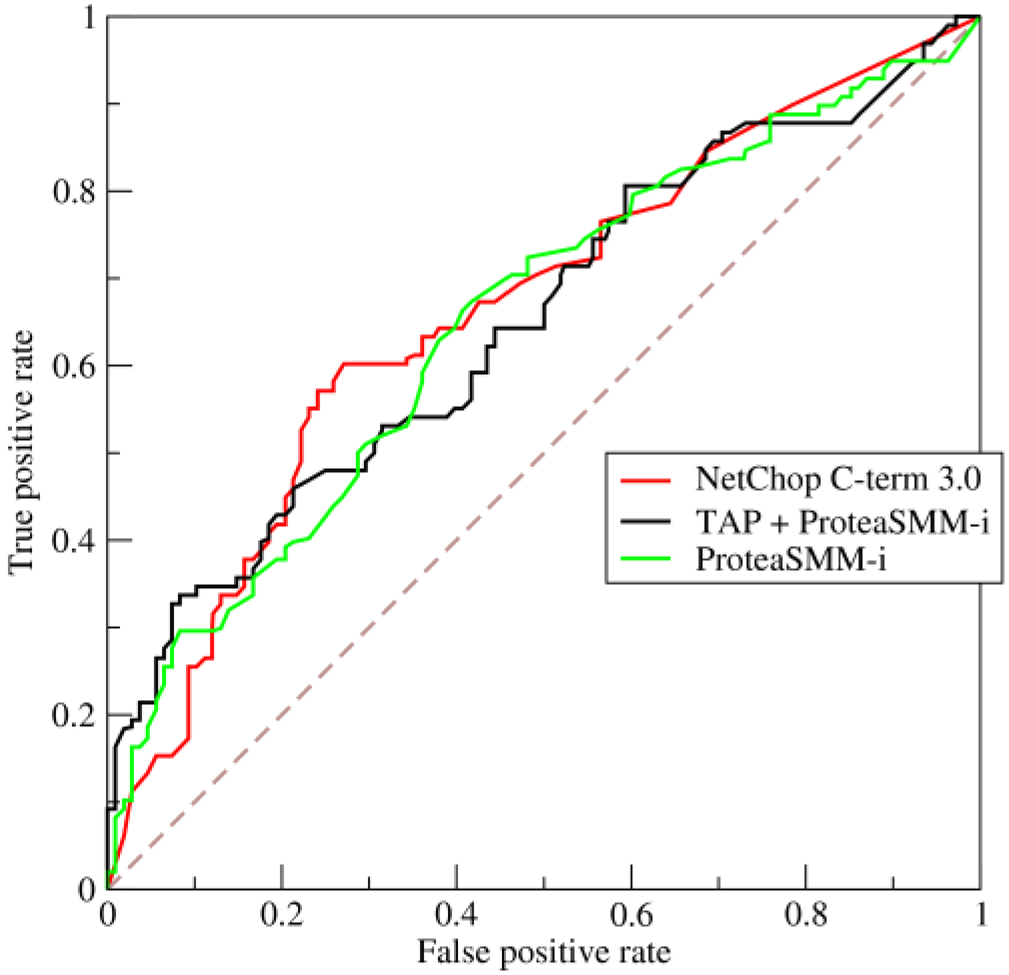
**83. What is the significance of Gamma and Regularization in SVM?**

The gamma defines influence. Low values meaning ‘far’ and high values meaning ‘close’. If gamma is too large, the radius of the area of influence of the support vectors only includes the support vector itself and no amount of regularization with C will be able to prevent overfitting. If gamma is very small, the model is too constrained and cannot capture the complexity of the data.

The regularization parameter (lambda) serves as a degree of importance that is given to miss-classifications. This can be used to draw the tradeoff with OverFitting.

**84. Define ROC curve work**

The graphical representation of the contrast between true positive rates and the false positive rate at various thresholds is known as the ROC curve. It is used as a proxy for the trade-off between true positives vs the false positives.



**85. What is the difference between a generative and discriminative model?**

A generative model learns the different categories of data. On the other hand, a discriminative model will only learn the distinctions between different categories of data. Discriminative models perform much better than the generative models when it comes to classification tasks.

**86. What are hyperparameters and how are they different from parameters?**

A parameter is a variable that is internal to the model and whose value is estimated from the training data. They are often saved as part of the learned model. Examples include weights, biases etc.

A hyperparameter is a variable that is external to the model whose value cannot be estimated from the data. They are often used to estimate model parameters. The choice of parameters is sensitive to implementation. Examples include learning rate, hidden layers etc.

**87. What is shattering a set of points? Explain VC dimension.**

In order to shatter a given configuration of points, a classifier must be able to, for all possible assignments of positive and negative for the points, perfectly partition the plane such that positive points are separated from negative points. For a configuration of ***n*** points, there are $\boldsymbol{2^n}$ possible assignments of positive or negative.

When choosing a classifier, we need to consider the type of data to be classified and this can be known by VC dimension of a classifier. It is defined as cardinality of the largest set of points that the classification algorithm i.e. the classifier can shatter. In order to have a VC dimension of *at* least **n**, a classifier must be able to shatter a single given configuration of **n** points.

**88. What are some differences between a linked list and an array?**

Arrays and Linked lists are both used to store linear data of similar types. However, there are a few difference between them.

| **Array** | **Linked List** |
|---|---|
| Elements are well-indexed, making specific element accessing easier | Elements need to be accessed in a cumulative manner |
| Operations (insertion, deletion) are faster in array | Linked list takes linear time, making operations a bit slower |
| Arrays are of fixed size | Linked lists are dynamic and flexible |
| Memory is assigned during compile time in an array | Memory is allocated during execution or runtime in Linked list. |
| Elements are stored consecutively in arrays. | Elements are stored randomly in Linked list |
| Memory utilization is inefficient in the array | Memory utilization is efficient in the linked list. |

**89. What is the meshgrid () method and the contourf () method? State some usesof both.**

The meshgrid( ) function in numpy takes two arguments as input : range of x-values in the grid, range of y-values in the grid whereas meshgrid needs to be built before the contourf( ) function in matplotlib is used which takes in many inputs : x-values, y-values, fitting curve (contour line) to be plotted in grid, colours etc.

Meshgrid () function is used to create a grid using 1-D arrays of x-axis inputs and y-axis inputs to represent the matrix indexing. Contourf () is used to draw filled contours using the given x-axis inputs, y-axis inputs, contour line, colours etc.

**90. Describe a hash table.**

Hashing is a technique for identifying unique objects from a group of similar objects. Hash functions are large keys converted into small keys in hashing techniques. The values of hash functions are stored in data structures which are known hash table.

**91. List the advantages and disadvantages of using Neural Networks.**

Advantages:

We can store information on the entire network instead of storing it in a database. It has the ability to work and give a good accuracy even with inadequate information. A neural network has parallel processing ability and distributed memory.

Disadvantages:

Neural Networks requires processors which are capable of parallel processing. It’s unexplained functioning of the network is also quite an issue as it reduces the trust in the network in some situations like when we have to show the problem we noticed to the network. Duration of the network is mostly unknown. We can only know that the training is finished by looking at the error value but it doesn’t give us optimal results.

**92. You have to train a 12GB dataset using a neural network with a machine which has only 3GB RAM. How would you go about it?**

We can use NumPy arrays to solve this issue. Load all the data into an array. In NumPy, arrays have a property to map the complete dataset without loading it completely in memory. We can pass the index of the array, dividing data into batches, to get the data required and then pass the data into the neural networks. But be careful about keeping the batch size normal.

**93. Write a simple code to binarize data.**

Conversion of data into binary values on the basis of certain threshold is known as binarizing of data. Values below the threshold are set to 0 and those above the threshold are set to 1 which is useful for feature engineering.

Code:

```
from sklearn.preprocessing import Binarizer
import pandas
import numpy
names_list = ['Alaska', 'Pratyush', 'Pierce', 'Sandra', 'Soundarya', 'Meredith', 'Richard', 'Jackson',
'Tom',’Joe’]
data_frame = pandas.read_csv(url, names=names_list)
array = dataframe.values
# Splitting the array into input and output
A = array [: 0:7]
B = array [:7]
binarizer = Binarizer(threshold=0.0). fit(X)
binaryA = binarizer.transform(A)
numpy.set_printoptions(precision=5)
print (binaryA [0:7:])
```

**Machine Learning Using Python Interview Questions**

**94. What is an Array?**

The array is defined as a collection of similar items, stored in a contiguous manner. Arrays is an intuitive concept as the need to group similar objects together arises in our day to day lives. Arrays satisfy the same need. How are they stored in the memory? Arrays consume blocks of data, where each element in the array consumes one unit of memory. The size of the unit depends on the type of data being used. For example, if the data type of elements of the array is int, then 4 bytes of data will be used to store each element. For character data type, 1 byte will be used. This is implementation specific, and the above units may change from computer to computer.

Example:

fruits = [‘apple’, banana’, pineapple’]

In the above case, fruits is a list that comprises of three fruits. To access them individually, we use their indexes. Python and C are 0- indexed languages, that is, the first index is 0. MATLAB on the contrary starts from 1, and thus is a 1-indexed language.

**95. What are the advantages and disadvantages of using an Array?**

1. Advantages:

- Random access is enabled
- Saves memory
- Cache friendly
- Predictable compile timing
- Helps in re-usability of code
- Disadvantages:

Addition and deletion of records is time consuming even though we get the element of interest immediately through random access. This is due to the fact that the elements need to be reordered after insertion or deletion.

If contiguous blocks of memory are not available in the memory, then there is an overhead on the CPU to search for the most optimal contiguous location available for the requirement.

Now that we know what arrays are, we shall understand them in detail by solving some interview questions. Before that, let us see the functions that Python as a language provides for arrays, also known as, lists.

append() – Adds an element at the end of the list
copy() – returns a copy of a list.
reverse() – reverses the elements of the list
sort() – sorts the elements in ascending order by default.

**96. What is Lists in Python?**

Lists is an effective data structure provided in python. There are various functionalities associated with the same. Let us consider the scenario where we want to copy a list to another list. If the same operation had to be done in C programming language, we would have to write our own function to implement the same.

On the contrary, Python provides us with a function called copy. We can copy a list to another just by calling the copy function.

```
new_list = old_list.copy()
```

We need to be careful while using the function. copy() is a shallow copy function, that is, it only stores the references of the original list in the new list. If the given argument is a compound data structure like a list then python creates another object of the same type (in this case, a new list) but for everything inside old list, only their reference is copied. Essentially, the new list consists of references to the elements of the older list.

Hence, upon changing the original list, the new list values also change. This can be dangerous in many applications. Therefore, Python provides us with another functionality called as deepcopy. Intuitively, we may consider that deepcopy() would follow the same paradigm, and the only difference would be that for each element we will recursively call deepcopy. Practically, this is not the case.

deepcopy() preserves the graphical structure of the original compound data. Let us understand this better with the help of an example:

```
import copy.deepcopy
a = [1,2]
b = [a,a] # there's only 1 object a
c = deepcopy(b)

# check the result by executing these lines
c[0] is a # return False, a new object a' is created
c[0] is c[1] # return True, c is [a',a'] not [a',a'']
```

This is the tricky part, during the process of deepcopy() a hashtable implemented as a dictionary in python is used to map: old_object reference onto new_object reference.

Therefore, this prevents unnecessary duplicates and thus preserves the structure of the copied compound data structure. Thus, in this case, c[0] is not equal to a, as internally their addresses are different.

Normal copy

```
a = [[1, 2, 3], [4, 5, 6]]
b = list(a)
a
Output:[[1, 2, 3], [4, 5, 6]]
b
Output: [[1, 2, 3], [4, 5, 6]]
a[0][1] = 10
a
Output: [[1, 10, 3], [4, 5, 6]]
b   # b changes too -> Not a deepcopy.
Output: [[1, 10, 3], [4, 5, 6]]
```

Deep copy

```
import copy
```

```
b = copy.deepcopy(a)
a
Output: [[1, 10, 3], [4, 5, 6]]
b
Output: [[1, 10, 3], [4, 5, 6]]
a[0][1] = 9
a
Output: [[1, 9, 3], [4, 5, 6]]
b    # b doesn't change -> Deep Copy
Output: [[1, 10, 3], [4, 5, 6]]
```

Now that we have understood the concept of lists, let us solve interview questions to get better exposure on the same.

**97. Given an array of integers where each element represents the max number of steps that can be made forward from that element. The task is to find the minimum number of jumps to reach the end of the array (starting from the first element). If an element is 0, then cannot move through that element.**

Solution: This problem is famously called as end of array problem. We want to determine the minimum number of jumps required in order to reach the end. The element in the array represents the maximum number of jumps that, that particular element can take.

Let us understand how to approach the problem initially.

We need to reach the end. Therefore, let us have a count that tells us how near we are to the end. Consider the array A=[1,2,3,1,1]

In the above example we can go from
> 2 - >3 - > 1 - > 1 - 4 jumps
1 - > 2 - > 1 - > 1 - 3 jumps
1 - > 2 - > 3 - > 1 - 3 jumps

Hence, we have a fair idea of the problem. Let us come up with a logic for the same.

Let us start from the end and move backwards as that makes more sense intuitionally. We will use variables right and prev_r denoting previous right to keep track of the jumps.

Initially, right = prev_r = the last but one element. We consider the distance of an element to the end, and the number of jumps possible by that element. Therefore, if the sum of the number of jumps possible and the distance is greater than the previous element, then we will discard the previous element and use the second element’s value to jump. Try it out using a pen and paper first. The logic will seem very straight forward to implement. Later, implement it on your own and then verify with the result.

```
def min_jmp(arr):


   n = len(arr)
   right = prev_r = n-1
```

```
    count = 0


    # We start from rightmost index and travesre array to find the leftmost index
    # from which we can reach index 'right'
    while True:
        for j in (range(prev_r-1,-1,-1)):
            if j + arr[j] >= prev_r:
                right = j


        if prev_r != right:
            prev_r = right
        else:
            break


        count += 1


    return count if right == 0 else -1


# Enter the elements separated by a space
arr = list(map(int, input().split()))
print(min_jmp(n, arr))
```

**98. Given a string S consisting only ‘a’s and ‘b’s, print the last index of the ‘b’ present in it.**

When we have are given a string of a’s and b’s, we can immediately find out the first location of a character occurring. Therefore, to find the last occurrence of a character, we reverse the string and find the first occurrence, which is equivalent to the last occurrence in the original string.

Here, we are given input as a string. Therefore, we begin by splitting the characters element wise using the function split. Later, we reverse the array, find the first occurrence position value, and get the index by finding the value len – position -1, where position is the index value.

```
def split(word):
    return [(char) for char in word]

a = input()
a= split(a)
a_rev = a[::-1]
pos = -1
for i in range(len(a_rev)):
    if a_rev[i] == ‘b’:
        pos = len(a_rev)- i -1
        print(pos)
        break
    else:
        continue
if pos==-1:
    print(-1)
```

**99. Rotate the elements of an array by d positions to the left. Let us initially look at an example.**

```
A = [1,2,3,4,5]
A <<2
[3,4,5,1,2]
A<<3
[4,5,1,2,3]
```

There exists a pattern here, that is, the first d elements are being interchanged with last n-d +1 elements. Therefore we can just swap the elements. Correct? What if the size of the array is huge, say 10000 elements. There are chances of memory error, run-time error etc. Therefore, we do it more carefully. We rotate the elements one by one in order to prevent the above errors, in case of large arrays.

```
# Rotate all the elements left by 1 position
def rot_left_once ( arr):
n = len( arr)
    tmp = arr [0]
    for i in range ( n-1): #[0,n-2]
        arr[i] = arr[i + 1]
arr[n-1] = tmp

# Use the above function to repeat the process for d times.
def rot_left (arr, d):
    n = len (arr)
    for i in range (d):
        rot_left_once ( arr, n)

arr = list( map( int, input().split()))
rot =int( input())
leftRotate ( arr, rot)

for i in range( len(arr)):
    print( arr[i], end=' ')
```

**100. Water Trapping Problem**

Given an array arr[] of N non-negative integers which represents the height of blocks at index I, where the width of each block is 1. Compute how much water can be trapped in between blocks after raining.

```
#  Structure is like below:

# | |

# |_|

# answer is we can trap two units of water.
```

Solution: We are given an array, where each element denotes the height of the block. One unit of height is equal to one unit of water, given there exists space between the 2 elements to store it. Therefore, we need to find out all such pairs that exist which can store water. We need to take care of the possible cases:

- There should be no overlap of water saved
- Water should not overflow

Therefore, let us find start with the extreme elements, and move towards the centre.

```
n = int(input())
arr = [int(i) for i in input().split()]
left, right = [arr[0]], [0] * n
# left =[arr[0]]
#right = [ 0 0 0 0…0] n terms
right[n-1] = arr[-1] # right most element

# we use two arrays left[ ] and right[ ], which keep track of elements greater than all
# elements the order of traversal respectively.

for elem in arr[1 : ]:
   left.append(max(left[-1], elem) )
for i in range( len( arr)-2, -1, -1):
   right[i] = max( arr[i] , right[i+1] )
water = 0
# once we have the arrays left, and right, we can find the water capacity between these arrays.

for i in range( 1, n - 1):
   add_water = min( left[i - 1], right[i]) - arr[i]
   if add_water > 0:
      water += add_water
print(water)
```

**101. Explain Eigenvectors and Eigenvalues.**

**Ans.** Linear transformations are helpful to understand using eigenvectors. They find their prime usage in the creation of covariance and correlation matrices in data science.

Simply put, eigenvectors are directional entities along which linear transformation features like compression, flip etc. can be applied.

Eigenvalues are the magnitude of the linear transformation features along each direction of an Eigenvector.

**102. How would you define the number of clusters in a clustering algorithm?**

**Ans.** The number of clusters can be determined by finding the silhouette score. Often we aim to get some inferences from data using clustering techniques so that we can have a broader picture of a number of classes being represented by the data. In this case, the silhouette score helps us determine the number of cluster centres to cluster our data along.

Another technique that can be used is the elbow method.

**103. What are the performance metrics that can be used to estimate the efficiency of a linear regression model?**

**Ans.** The performance metric that is used in this case is:

- Mean Squared Error
- $R^2$ score
- Adjusted $R^2$ score
- Mean Absolute score

**104. What is the default method of splitting in decision trees?**

The default method of splitting in decision trees is the Gini Index. Gini Index is the measure of impurity of a particular node.

This can be changed by making changes to classifier parameters.

**105. How is p-value useful?**

**Ans.** The p-value gives the probability of the null hypothesis is true. It gives us the statistical significance of our results. In other words, p-value determines the confidence of a model in a particular output.

**106. Can logistic regression be used for classes more than 2?**

**Ans.** No, logistic regression cannot be used for classes more than 2 as it is a binary classifier. For multi-class classification algorithms like Decision Trees, Naïve Bayes' Classifiers are better suited.

**107. What are the hyperparameters of a logistic regression model?**

**Ans.** Classifier penalty, classifier solver and classifier C are the trainable hyperparameters of a Logistic Regression Classifier. These can be specified exclusively with values in Grid Search to hyper tune a Logistic Classifier.

**108. Name a few hyper-parameters of decision trees?**

**Ans.** The most important features which one can tune in decision trees are:

- Splitting criteria
- Min_leaves
- Min_samples
- Max_depth

**109. How to deal with multicollinearity?**

**Ans.** Multi collinearity can be dealt with by the following steps:

- Remove highly correlated predictors from the model.
- Use Partial Least Squares Regression (PLS) or Principal Components Analysis

**110. What is Heteroscedasticity?**

**Ans.** It is a situation in which the variance of a variable is unequal across the range of values of the predictor variable.

It should be avoided in regression as it introduces unnecessary variance.

**111. Is ARIMA model a good fit for every time series problem?**

**Ans.** No, ARIMA model is not suitable for every type of time series problem. There are situations where ARMA model and others also come in handy.

ARIMA is best when different standard temporal structures require to be captured for time series data.

**112. How do you deal with the class imbalance in a classification problem?**

**Ans.** Class imbalance can be dealt with in the following ways:

- Using class weights
- Using Sampling
- Using SMOTE
- Choosing loss functions like Focal Loss

**113. What is the role of cross-validation?**

**Ans.** Cross-validation is a technique which is used to increase the performance of a machine learning algorithm, where the machine is fed sampled data out of the same data for a few times. The sampling is done so that the dataset is broken into small parts of the equal number of rows, and a random part is chosen as the test set, while all other parts are chosen as train sets.

**114. What is a voting model?**

**Ans.** A voting model is an ensemble model which combines several classifiers but to produce the final result, in case of a classification-based model, takes into account, the classification of a certain data point of all the models and picks the most vouched/voted/generated option from all the given classes in the target column.

**115. How to deal with very few data samples? Is it possible to make a model out of it?**

**Ans.** If very few data samples are there, we can make use of oversampling to produce new data points. In this way, we can have new data points.

**116. What are the hyperparameters of an SVM?**

**Ans.** The gamma value, c value and the type of kernel are the hyperparameters of an SVM model.

**117. What is Pandas Profiling?**

**Ans.** Pandas profiling is a step to find the effective number of usable data. It gives us the statistics of NULL values and the usable values and thus makes variable selection and data selection for building models in the preprocessing phase very effective.

**118. What impact does correlation have on PCA?**

**Ans.** If data is correlated PCA does not work well. Because of the correlation of variables the effective variance of variables decreases. Hence correlated data when used for PCA does not work well.

**119. How is PCA different from LDA?**

**Ans.** PCA is unsupervised. LDA is unsupervised.

PCA takes into consideration the variance. LDA takes into account the distribution of classes.

**120. What distance metrics can be used in KNN?**

**Ans.** Following distance metrics can be used in KNN.

- Manhattan
- Minkowski
- Tanimoto
- Jaccard
- Mahalanobis

**121. Which metrics can be used to measure correlation of categorical data?**

**Ans.** Chi square test can be used for doing so. It gives the measure of correlation between categorical predictors.

**122. Which algorithm can be used in value imputation in both categorical and continuous categories of data?**

**Ans.** KNN is the only algorithm that can be used for imputation of both categorical and continuous variables.

**123. When should ridge regression be preferred over lasso?**

**Ans.** We should use ridge regression when we want to use all predictors and not remove any as it reduces the coefficient values but does not nullify them.

**124. Which algorithms can be used for important variable selection?**

**Ans.** Random Forest, Xgboost and plot variable importance charts can be used for variable selection.

**125. What ensemble technique is used by Random forests?**

**Ans.** Bagging is the technique used by Random Forests. Random forests are a collection of trees which work on sampled data from the original dataset with the final prediction being a voted average of all trees.

**126. What ensemble technique is used by gradient boosting trees?**

**Ans.** Boosting is the technique used by GBM.

**127. If we have a high bias error what does it mean? How to treat it?**

**Ans.** High bias error means that that model we are using is ignoring all the important trends in the model and the model is underfitting.

To reduce underfitting:

- We need to increase the complexity of the model
- Number of features need to be increased

Sometimes it also gives the impression that the data is noisy. Hence noise from data should be removed so that most important signals are found by the model to make effective predictions.

Increasing the number of epochs results in increasing the duration of training of the model. It's helpful in reducing the error.

**128. Which type of sampling is better for a classification model and why?**

**Ans.** Stratified sampling is better in case of classification problems because it takes into account the balance of classes in train and test sets. The proportion of classes is maintained and hence the model performs better. In case of random sampling of data, the data is divided into two parts without taking into consideration the balance classes in the train and test sets. Hence some classes might be present only in tarin sets or validation sets. Hence the results of the resulting model are poor in this case.

**129. What is a good metric for measuring the level of multicollinearity?**

**Ans.** VIF or 1/tolerance is a good measure of measuring multicollinearity in models. VIF is the percentage of the variance of a predictor which remains unaffected by other predictors. So higher the VIF value, greater is the multicollinearity amongst the predictors.

A **rule of thumb** for interpreting the variance inflation factor:

- 1 = not correlated.
- Between 1 and 5 = moderately correlated.
- Greater than 5 = highly correlated.

**130. When can be a categorical value treated as a continuous variable and what effect does it have when done so?**

**Ans.** A categorical predictor can be treated as a continuous one when the nature of data points it represents is ordinal. If the predictor variable is having ordinal data then it can be treated as continuous and its inclusion in the model increases the performance of the model.

**131. What is the role of maximum likelihood in logistic regression.**

**Ans.** Maximum likelihood equation helps in estimation of most probable values of the estimator's predictor variable coefficients which produces results which are the most likely or most probable and are quite close to the truth values.

**132. Which distance do we measure in the case of KNN?**

**Ans.** The hamming distance is measured in case of KNN for the determination of nearest neighbours. Kmeans uses euclidean distance.

**133. What is a pipeline?**

**Ans.** A pipeline is a sophisticated way of writing software such that each intended action while building a model can be serialized and the process calls the individual functions for the individual tasks. The tasks are carried out in sequence for a given sequence of data points and the entire process can be run onto n threads by use of composite estimators in scikit learn.

**134. Which sampling technique is most suitable when working with time-series data?**

**Ans.** We can use a custom iterative sampling such that we continuously add samples to the train set. We only should keep in mind that the sample used for validation should be added to the next train sets and a new sample is used for validation.

**135. What are the benefits of pruning?**

**Ans.** Pruning helps in the following:

- Reduces overfitting
- Shortens the size of the tree
- Reduces complexity of the model
- Increases bias

**136. What is normal distribution?**

**Ans.** The distribution having the below properties is called normal distribution.

- The mean, mode and median are all equal.
- The curve is symmetric at the center (i.e. around the mean, μ).

- Exactly half of the values are to the left of center and exactly half the values are to the right.
- The total area under the curve is 1.

**137. What is the 68 per cent rule in normal distribution?**

**Ans.** The normal distribution is a bell-shaped curve. Most of the data points are around the median. Hence approximately 68 per cent of the data is around the median. Since there is no skewness and its bell-shaped.

**138. What is a chi-square test?**

**Ans.** A chi-square determines if a sample data matches a population.

A chi-square test for independence compares two variables in a contingency table to see if they are related.

A very small chi-square test statistics implies observed data fits the expected data extremely well.

**139. What is a random variable?**

**Ans.** A Random Variable is a set of possible values from a random experiment. Example: Tossing a coin: we could get Heads or Tails. Rolling of a dice: we get 6 values

**140. What is the degree of freedom?**

**Ans.** It is the number of independent values or quantities which can be assigned to a statistical distribution. It is used in Hypothesis testing and chi-square test.

**141. Which kind of recommendation system is used by amazon to recommend similar items?**

**Ans.** Amazon uses a collaborative filtering algorithm for the recommendation of similar items. It's a user to user similarity based mapping of user likeness and susceptibility to buy.

**142. What is a false positive?**

**Ans.** It is a test result which wrongly indicates that a particular condition or attribute is present.

Example – "Stress testing, a routine diagnostic tool used in detecting heart disease, results in a significant number of false positives in women"

**143. What is a false negative?**

**Ans.** A test result which wrongly indicates that a particular condition or attribute is absent.

Example – "it's possible to have a false negative—the test says you aren't pregnant when you are"

**144. What is the error term composed of in regression?**

**Ans.** Error is a sum of bias error+variance error+ irreducible error in regression. Bias and variance error can be reduced but not the irreducible error.

**145. Which performance metric is better R2 or adjusted R2?**

**Ans.** Adjusted R2 because the performance of predictors impacts it. R2 is independent of predictors and shows performance improvement through increase if the number of predictors is increased.

**146. What's the difference between Type I and Type II error?**

Type I and Type II error in machine learning refers to false values. Type I is equivalent to a False positive while Type II is equivalent to a False negative. In Type I error, a hypothesis which ought to be accepted doesn't get accepted. Similarly, for Type II error, the hypothesis gets rejected which should have been accepted in the first place.

**147. What do you understand by L1 and L2 regularization?**

L2 regularization: It tries to spread error among all the terms. L2 corresponds to a Gaussian prior.

L1 regularization: It is more binary/sparse, with many variables either being assigned a 1 or 0 in weighting. L1 corresponds to setting a Laplacean prior on the terms.

**148. Which one is better, Naive Bayes Algorithm or Decision Trees?**

Although it depends on the problem you are solving, but some general advantages are following:

**Naive Bayes:**

- Work well with small dataset compared to DT which need more data
- Lesser overfitting
- Smaller in size and faster in processing

**Decision Trees:**

- Decision Trees are very flexible, easy to understand, and easy to debug
- No preprocessing or transformation of features required
- Prone to overfitting but you can use pruning or Random forests to avoid that.

**149. What do you mean by the ROC curve?**

Receiver operating characteristics (ROC curve): ROC curve illustrates the diagnostic ability of a binary classifier. It is calculated/created by plotting True Positive against False Positive at various threshold settings. The performance metric of ROC curve is AUC (area under curve). Higher the area under the curve, better the prediction power of the model.

### 150. What do you mean by AUC curve?

AUC (area under curve). Higher the area under the curve, better the prediction power of the model.

### 151. What is log likelihood in logistic regression?

It is the sum of the likelihood residuals. At record level, the natural log of the error (residual) is calculated for each record, multiplied by minus one, and those values are totaled. That total is then used as the basis for deviance (2 x ll) and likelihood (exp(ll)).

The same calculation can be applied to a naive model that assumes absolutely no predictive power, and a saturated model assuming perfect predictions.

The likelihood values are used to compare different models, while the deviances (test, naive, and saturated) can be used to determine the predictive power and accuracy. Logistic regression accuracy of the model will always be 100 percent for the development data set, but that is not the case once a model is applied to another data set.

### 152. How would you evaluate a logistic regression model?

Model Evaluation is a very important part in any analysis to answer the following questions,

How well does the model fit the data?, Which predictors are most important?, Are the predictions accurate?

So the following are the criterion to access the model performance,

- **Akaike Information Criteria (AIC)**: In simple terms, AIC estimates the relative amount of information lost by a given model. So the less information lost the higher the quality of the model. Therefore, we always prefer models with minimum AIC.
- **Receiver operating characteristics (ROC curve)**: ROC curve illustrates the diagnostic ability of a binary classifier. It is calculated/ created by plotting True Positive against False Positive at various threshold settings. The performance metric of ROC curve is AUC (area under curve). Higher the area under the curve, better the prediction power of the model.
- **Confusion Matrix**: In order to find out how well the model does in predicting the target variable, we use a confusion matrix/ classification rate. It is nothing but a tabular representation of actual Vs predicted values which helps us to find the accuracy of the model.

### 153. What are the advantages of SVM algorithms?

SVM algorithms have basically advantages in terms of complexity. First I would like to clear that both Logistic regression as well as SVM can form non linear decision surfaces and can be coupled with the kernel trick. If Logistic regression can be coupled with kernel then why use SVM?

● SVM is found to have better performance practically in most cases.

● SVM is computationally cheaper O(N^2*K) where K is no of support vectors (support vectors are those points that lie on the class margin) where as logistic regression is O(N^3)

● Classifier in SVM depends only on a subset of points . Since we need to maximize distance between closest points of two classes (aka margin) we need to care about only a subset of points unlike logistic regression.

**154. Why does XGBoost perform better than SVM?**

First reason is that XGBoos is an ensemble method that uses many trees to make a decision so it gains power by repeating itself.

SVM is a linear separator, when data is not linearly separable SVM needs a Kernel to project the data into a space where it can separate it, there lies its greatest strength and weakness, by being able to project data into a high dimensional space SVM can find a linear separation for almost any data but at the same time it needs to use a Kernel and we can argue that there's not a perfect kernel for every dataset.

**155. What is the difference between SVM Rank and SVR (Support Vector Regression)?**

One is used for ranking and the other is used for regression.

There is a crucial difference between *regression* and *ranking*. In regression, the absolute value is crucial. A real number is predicted.

In ranking, the only thing of concern is the ordering of a set of examples. We only want to know which example has the highest rank, which one has the second-highest, and so on. From the data, we only know that example 1 should be ranked higher than example 2, which in turn should be ranked higher than example 3, and so on. We do not know by *how much* example 1 is ranked higher than example 2, or whether this difference is bigger than the difference between examples 2 and 3.

**156. What is the difference between the normal soft margin SVM and SVM with a linear kernel?**

**Hard-margin**

You have the basic SVM – hard margin. This assumes that data is very well behaved, and you can find a perfect classifier – which will have 0 error on train data.

**Soft-margin**

Data is usually not well behaved, so SVM hard margins may not have a solution at all. So we allow for a little bit of error on some points. So the training error will not be 0, but average error over all points is minimized.

**Kernels**

The above assume that the best classifier is a straight line. But what is it is not a straight line. (e.g. it is a circle, inside a circle is one class, outside is another class). If we are able to map the data into higher dimensions – the higher dimension may give us a straight line.

### 157. How is linear classifier relevant to SVM?

An svm is a type of linear classifier. If you don't mess with kernels, it's arguably the most simple type of linear classifier.

Linear classifiers (all?) learn linear fictions from your data that map your input to scores like so: scores = Wx + b. Where W is a matrix of learned weights, b is a learned bias vector that shifts your scores, and x is your input data. This type of function may look familiar to you if you remember y = mx + b from high school.

A typical svm loss function ( the function that tells you how good your calculated scores are in relation to the correct labels ) would be hinge loss. It takes the form: Loss = sum over all scores except the correct score of max(0, scores – scores(correct class) + 1).

### 158. What are the advantages of using a naive Bayes for classification?

- Very simple, easy to implement and fast.
- If the NB conditional independence assumption holds, then it will converge quicker than discriminative models like logistic regression.
- Even if the NB assumption doesn't hold, it works great in practice.
- Need less training data.
- Highly scalable. It scales linearly with the number of predictors and data points.
- Can be used for both binary and mult-iclass classification problems.
- Can make probabilistic predictions.
- Handles continuous and discrete data.
- Not sensitive to irrelevant features.

### 159. Are Gaussian Naive Bayes the same as binomial Naive Bayes?

Binomial Naive Bayes: It assumes that all our features are binary such that they take only two values. Means 0s can represent "word does not occur in the document" and 1s as "word occurs in the document".

Gaussian Naive Bayes: Because of the assumption of the normal distribution, Gaussian Naive Bayes is used in cases when all our features are continuous. For example in Iris dataset features are sepal width, petal width, sepal length, petal length. So its features can have different values in the data set as width and length can vary. We can't represent features in terms of their occurrences. This means data is continuous. Hence we use Gaussian Naive Bayes here.

### 160. What is the difference between the Naive Bayes Classifier and the Bayes classifier?

Naive Bayes assumes conditional independence, P(X|Y, Z)=P(X|Z)

P(X|Y,Z)=P(X|Z)

P(X|Y,Z)=P(X|Z), Whereas more general Bayes Nets (sometimes called Bayesian Belief Networks), will allow the user to specify which attributes are, in fact, conditionally independent.

For the Bayesian network as a classifier, the features are selected based on some scoring functions like Bayesian scoring function and minimal description length(the two are equivalent in theory to each other given that there is enough training data). The scoring functions mainly restrict the structure (connections and directions) and the parameters(likelihood) using the data. After the structure has been learned the class is only determined by the nodes in the Markov blanket(its parents, its children, and the parents of its children), and all variables given the Markov blanket are discarded.

**161. In what real world applications is Naive Bayes classifier used?**

Some of real world examples are as given below

- To mark an email as spam, or not spam?
- Classify a news article about technology, politics, or sports?
- Check a piece of text expressing positive emotions, or negative emotions?
- Also used for face recognition software

**162. Is naive Bayes supervised or unsupervised?**

First, Naive Bayes is not one algorithm but a family of Algorithms that inherits the following attributes:

- Discriminant Functions
- Probabilistic Generative Models
- Bayesian Theorem
- Naive Assumptions of Independence and Equal Importance of feature vectors.

Moreover, it is a special type of Supervised Learning algorithm that could do simultaneous multi-class predictions (as depicted by standing topics in many news apps).

Since these are generative models, so based upon the assumptions of the random variable mapping of each feature vector these may even be classified as Gaussian Naive Bayes, Multinomial Naive Bayes, Bernoulli Naive Bayes, etc.

**163. What do you understand by selection bias in Machine Learning?**

Selection bias stands for the bias which was introduced by the selection of individuals, groups or data for doing analysis in a way that the proper randomization is not achieved. It ensures that the sample obtained is not representative of the population intended to be analyzed and sometimes it is referred to as the selection effect. This is the part of distortion of a statistical analysis which results from the method of collecting samples. If you don’t take the selection bias into the account then some conclusions of the study may not be accurate.

The types of selection bias includes:

- **Sampling bias**: It is a systematic error due to a non-random sample of a population causing some members of the population to be less likely to be included than others resulting in a biased sample.
- **Time interval**: A trial may be terminated early at an extreme value (often for ethical reasons), but the extreme value is likely to be reached by the variable with the largest variance, even if all variables have a similar mean.
- **Data**: When specific subsets of data are chosen to support a conclusion or rejection of bad data on arbitrary grounds, instead of according to previously stated or generally agreed criteria.
- **Attrition**: Attrition bias is a kind of selection bias caused by attrition (loss of participants) discounting trial subjects/tests that did not run to completion.

**164. What do you understand by Precision and Recall?**

In pattern recognition, The information retrieval and classification in machine learning are part of **precision**. It is also called as positive predictive value which is the fraction of relevant instances among the retrieved instances.

**Recall** is also known as sensitivity and the fraction of the total amount of relevant instances which  were actually retrieved.

Both precision and recall are therefore based on an understanding and measure of relevance.

**165. What Are the Three Stages of Building a Model in Machine Learning?**

To build a model in machine learning, you need to follow few steps:

- Understand the business model
- Data acquisitions
- Data cleaning
- Exploratory data analysis
- Use machine learning algorithms to make a model
- Use unknown dataset to check the accuracy of the model

**166. How Do You Design an Email Spam Filter in Machine Learning?**

- Understand the business model: Try to understand the related attributes for the spam mail
- Data acquisitions: Collect the spam mail to read the hidden pattern from them
- Data cleaning: Clean the unstructured or semi structured data
- Exploratory data analysis: Use statistical concepts to understand the data like spread, outlier, etc.
- Use machine learning algorithms to make a model: can use naive bayes or some other algorithms as well
- Use unknown dataset to check the accuracy of the model

**167. What is the difference between Entropy and Information Gain?**

The **information gain** is based on the decrease in **entropy** after a dataset is split on an attribute. Constructing a decision tree is all about finding the attribute that returns the highest

**information gain** (i.e., the most homogeneous branches). Step 1: Calculate **entropy** of the target.

**168. What are collinearity and multicollinearity?**

**Collinearity** is a linear association **between** two predictors. **Multicollinearity** is a situation where two or more predictors are highly linearly related.

**169. What is Kernel SVM?**

SVM algorithms have basically advantages in terms of complexity. First I would like to clear that both Logistic regression as well as SVM can form non linear decision surfaces and can be coupled with the kernel trick. If Logistic regression can be coupled with kernel then why use SVM?

● SVM is found to have better performance practically in most cases.

● SVM is computationally cheaper O(N^2*K) where K is no of support vectors (support vectors are those points that lie on the class margin) where as logistic regression is O(N^3)

● Classifier in SVM depends only on a subset of points . Since we need to maximize distance between closest points of two classes (aka margin) we need to care about only a subset of points unlike logistic regression.

**170. What is the process of carrying out a linear regression?**

**Linear Regression** Analysis consists of more than just fitting a **linear** line through a cloud of data points. It consists of 3 stages–

- analyzing the correlation and directionality of the data,
- estimating the **model**, i.e., fitting the line,
- evaluating the validity and usefulness of the **model**.