

Machine Learning

What's the trade-off between bias and variance?

Bias is error due to erroneous or overly simplistic assumptions in the learning algorithm you're using. This can lead to the model underfitting your 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 error due to too much complexity in the learning algorithm you're using. This leads to the algorithm being highly sensitive to high degrees of variation in your training data, which can lead your model to overfit the data. You'll be carrying too much noise from your 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 tradeoff bias and variance. You don't want either high bias or high variance in your model.

What is the difference between supervised and unsupervised machine learning?

Supervised learning requires training labeled data. For example, in order to do classification (a supervised learning task), you'll need to first label the data you'll use to train the model to classify data into your labeled groups. Unsupervised learning, in contrast, does not require labeling data explicitly.

How is KNN different from k-means clustering?

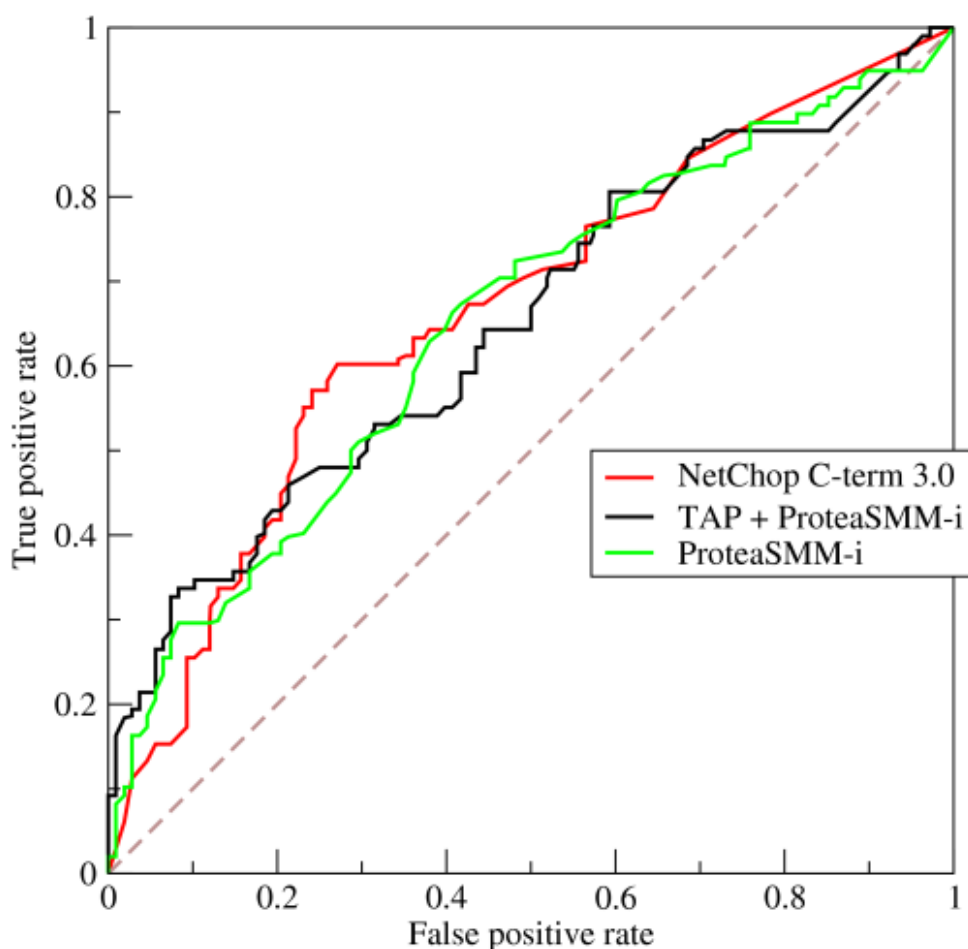
K-Nearest Neighbors is a supervised classification algorithm, while k-means clustering is an unsupervised clustering algorithm. While the mechanisms may seem similar at first, what this really means is that in order for K-Nearest Neighbors to work, you need labeled data you want to classify an unlabeled point into (thus the nearest neighbor part). K-means clustering requires only a set of unlabeled

points and a threshold: the algorithm will take unlabeled points and gradually learn how to cluster them into groups by computing the mean of the distance between different points.

The critical difference here is that KNN needs labeled points and is thus supervised learning, while k-means doesn't — and is thus unsupervised learning.

Explain how a ROC curve works.

The ROC curve is a graphical representation of the contrast between true positive rates and the false positive rate at various thresholds. It's often used as a proxy for the trade-off between the sensitivity of the model (true positives) vs the fall-out or the probability it will trigger a false alarm (false positives).



Define precision and recall.

Recall is also known as the true positive rate: the number of positives your model claims compared to the actual number of positives there are throughout the data.

Precision is also known as the positive predictive value, and it is a measure of the number of accurate positives your model claims compared to the number of positives it actually claims.

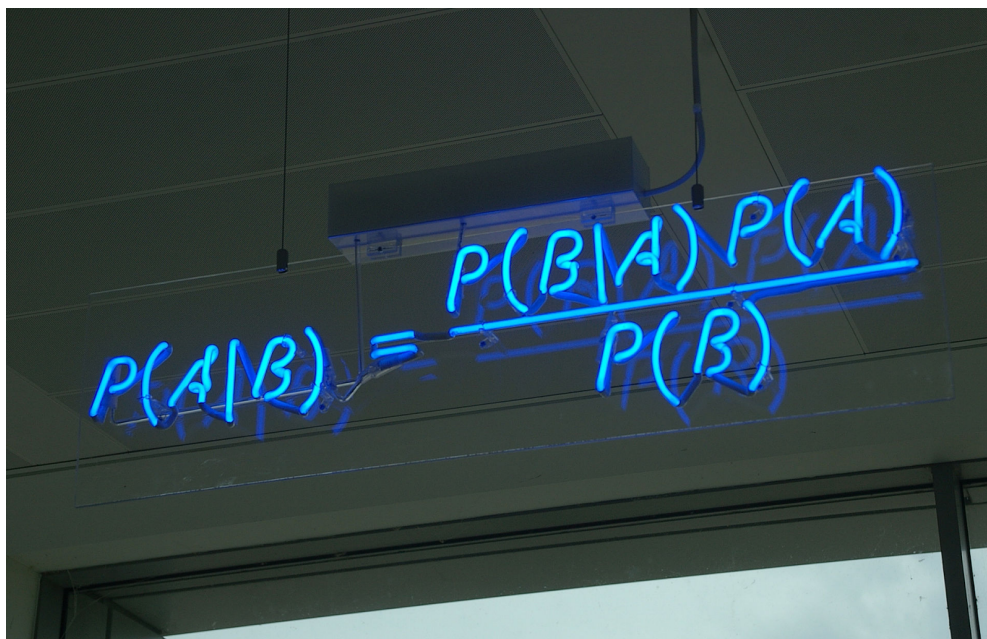
It can be easier to think of recall and precision in the context of a case where you've predicted that there were 10 apples and 5 oranges in a case of 10 apples. You'd have perfect recall (there are actually 10 apples, and you predicted there would be 10) but 66.7% precision because out of the 15 events you predicted, only 10 (the apples) are correct.

What is Bayes' Theorem? How is it useful in a machine learning context?

Bayes' Theorem gives you the posterior probability of an event given what is known as prior knowledge.

Mathematically, it's expressed as the true positive rate of a condition sample divided by the sum of the false positive rate of the population and the true positive rate of a condition. Say you had a 60% chance of actually having the flu after a flu test, but out of people who had the flu, the test will be false 50% of the time, and the overall population only has a 5% chance of having the flu. Would you actually have a 60% chance of having the flu after having a positive test?

Bayes' Theorem says no. It says that you have a $(.6 * 0.05)$ (True Positive Rate of a Condition Sample) / $(.6 * 0.05)$ (True Positive Rate of a Condition Sample) + $(.5 * 0.95)$ (False Positive Rate of a Population) = 0.0594 or 5.94% chance of getting a flu.


$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayes' Theorem is the basis behind a branch of machine learning that most notably includes the Naive Bayes classifier. That's something important to consider when you're faced with machine learning interview questions.

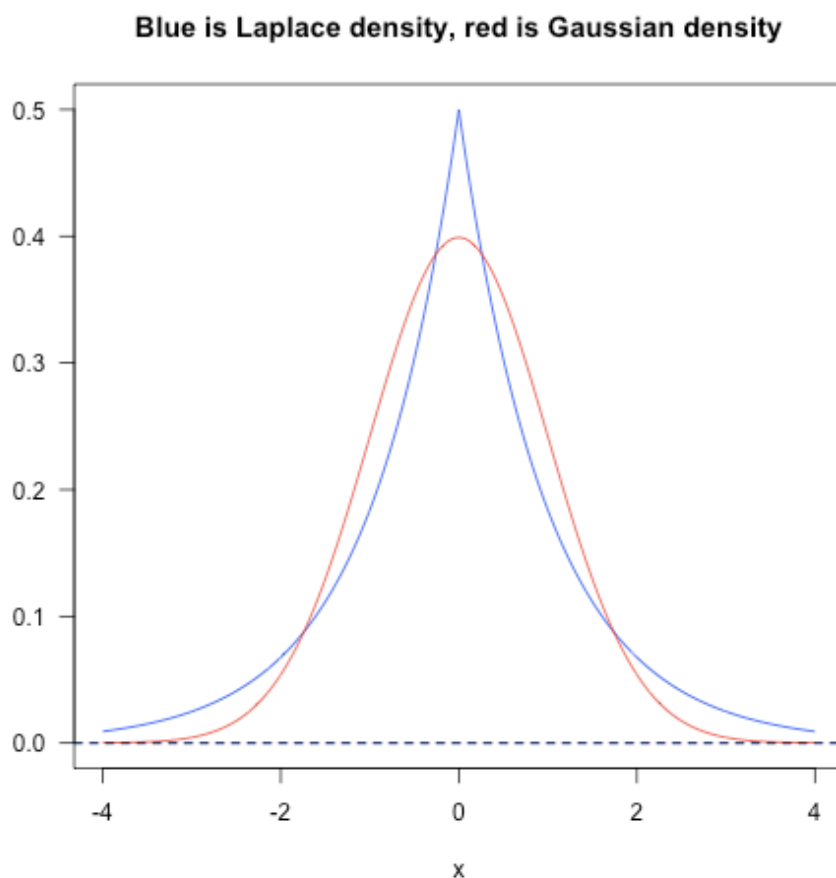
Why is "Naive" Bayes naive?

Despite its practical applications, especially in text mining, Naive Bayes is considered "Naive" because it makes an assumption that is virtually impossible to see in real-life data: the conditional probability is calculated as the pure product of the individual probabilities of components. This implies the absolute independence of features — a condition probably never met in real life.

A Naive Bayes classifier that figured out that you liked pickles and ice cream would probably naively recommend you a pickle ice cream.

Explain the difference between L1 and L2 regularization.

L2 regularization tends to spread error among all the terms, while L1 is more binary/sparser, with many variables either being assigned a 1 or 0 in weighting. L1 corresponds to setting a Laplacian prior on the terms, while L2 corresponds to a Gaussian prior.



What's your favorite algorithm, and can you explain it to me in less than a minute?

This type of question tests your understanding of how to communicate complex and technical nuances with poise and the ability to summarize quickly and efficiently. Make sure you have a choice and make sure you can explain different algorithms so simply and effectively that a five-year-old could grasp the basics!

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

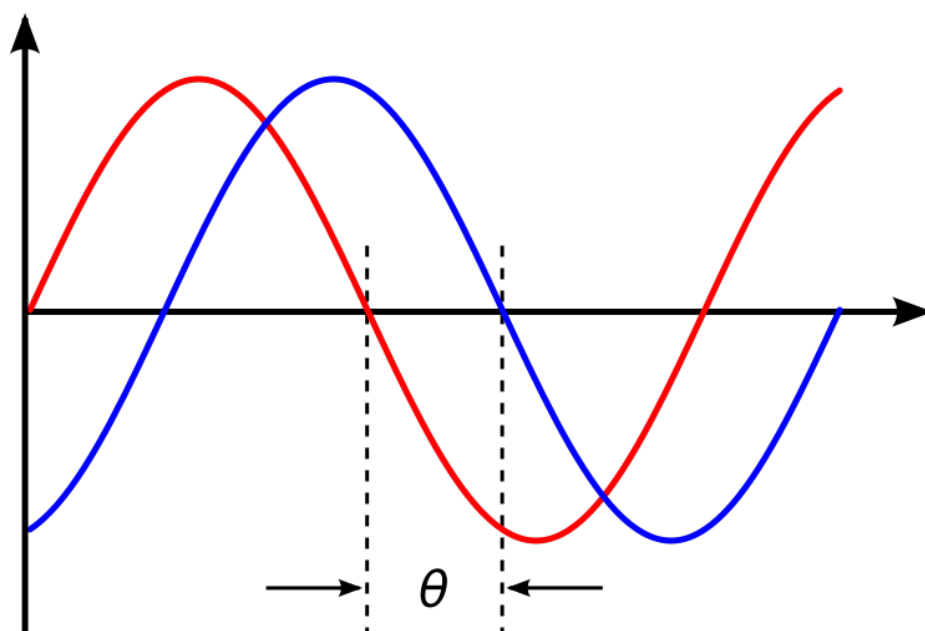
Type I error is a false positive, while Type II error is a false negative. Briefly stated, Type I error means claiming something has happened when it hasn't, while Type II error means that you claim nothing is happening when in fact something is.

A clever way to think about this is to think of Type I error as telling a man he is pregnant, while Type II error means you tell a pregnant woman she isn't carrying a baby.

What's a Fourier transform?

A Fourier transform is a generic method to decompose generic functions into a superposition of symmetric functions. Or as this [more intuitive tutorial](#) puts it, given a smoothie, it's how we find the recipe. The Fourier transform finds the set of cycle speeds, amplitudes and phases to match any time signal. A Fourier transform converts a signal from time to frequency domain — it's a very common way to extract features from audio signals or other time series such as sensor data.

What's the difference between probability and likelihood?



What is deep learning, and how does it contrast with other machine learning algorithms?

Deep learning is a subset of machine learning that is concerned with neural networks: how to use backpropagation and certain principles from neuroscience to more accurately model large sets of unlabeled or semi-structured data. In that

sense, deep learning represents an unsupervised learning algorithm that learns representations of data through the use of neural nets.

What's the difference between a generative and discriminative model?

A generative model will learn categories of data while a discriminative model will simply learn the distinction between different categories of data. Discriminative models will generally outperform generative models on classification tasks.

What cross-validation technique would you use on a time series dataset?

Instead of using standard k-folds cross-validation, you have to pay attention to the fact that a time series is not randomly distributed data — it is inherently ordered by chronological order. If a pattern emerges in later time periods for example, your model may still pick up on it even if that effect doesn't hold in earlier years!

You'll want to do something like forward chaining where you'll be able to model on past data then look at forward-facing data.

- a. fold 1: training [1], test [2]
- b. fold 2: training [1 2], test [3]
- c. fold 3: training [1 2 3], test [4]
- d. fold 4: training [1 2 3 4], test [5]
- e. fold 5: training [1 2 3 4 5], test [6]

How is a decision tree pruned?

Pruning is what happens in decision trees when branches that have weak predictive power are removed in order to reduce the complexity of the model and increase the predictive accuracy of a decision tree model. Pruning can happen bottom-up and top-down, with approaches such as reduced error pruning and cost complexity pruning.

Reduced error pruning is perhaps the simplest version: replace each node. If it doesn't decrease predictive accuracy, keep it pruned. While simple, this heuristic actually comes pretty close to an approach that would optimize for maximum accuracy.

Which is more important to you— model accuracy, or model performance?

Well, it has everything to do with how model accuracy is only a subset of model performance, and at that, a sometimes misleading one. For example, if you wanted to detect fraud in a massive dataset with a sample of millions, a more accurate model would most likely predict no fraud at all if only a vast minority of cases were fraud. However, this would be useless for a predictive model — a model designed to find fraud that asserted there was no fraud at all! Questions like this help you demonstrate that you understand model accuracy isn't the be-all and end-all of model performance.

What's the F1 score? How would you use it?

The F1 score is a measure of a model's performance. It is a weighted average of the precision and recall of a model, with results tending to 1 being the best, and those tending to 0 being the worst. You would use it in classification tests where true negatives don't matter much.

How would you handle an imbalanced dataset?

An imbalanced dataset is when you have, for example, a classification test and 90% of the data is in one class. That leads to problems: an accuracy of 90% can be skewed if you have no predictive power on the other category of data! Here are a few tactics to get over the hump:

- a. Collect more data to even the imbalances in the dataset.
- b. Resample the dataset to correct for imbalances.
- c. Try a different algorithm altogether on your dataset.

What's important here is that you have a keen sense for what damage an unbalanced dataset can cause, and how to balance that.

When should you use classification over regression?

Classification produces discrete values and dataset to strict categories, while regression gives you continuous results that allow you to better distinguish differences between individual points. You would use classification over regression

if you wanted your results to reflect the belongingness of data points in your dataset to certain explicit categories (ex: If you wanted to know whether a name was male or female rather than just how correlated they were with male and female names.)

Name an example where ensemble techniques might be useful.

Ensemble techniques use a combination of learning algorithms to optimize better predictive performance. They typically reduce overfitting in models and make the model more robust (unlikely to be influenced by small changes in the training data).

You could list some examples of ensemble methods, from bagging to boosting to a “bucket of models” method and demonstrate how they could increase predictive power.

How do you ensure you’re not overfitting with a model?

This is a simple restatement of a fundamental problem in machine learning: the possibility of overfitting training data and carrying the noise of that data through to the test set, thereby providing inaccurate generalizations.

There are three main methods to avoid overfitting:

- a. Keep the model simpler: reduce variance by taking into account fewer variables and parameters, thereby removing some of the noise in the training data.
- b. Use cross-validation techniques such as k-folds cross-validation.
- c. Use regularization techniques such as LASSO that penalize certain model parameters if they’re likely to cause overfitting.

What evaluation approaches would you work to gauge the effectiveness of a machine learning model?

You would first split the dataset into training and test sets, or perhaps use cross-validation techniques to further segment the dataset into composite sets of training and test sets within the data. You should then implement a choice selection of performance metrics: here is a fairly [comprehensive list](#). You could use measures such as the F1 score, the accuracy, and the confusion matrix. What’s important

here is to demonstrate that you understand the nuances of how a model is measured and how to choose the right performance measures for the right situations.

How would you evaluate a logistic regression model?

A subsection of the question above. You have to demonstrate an understanding of what the typical goals of a logistic regression are (classification, prediction etc.) and bring up a few examples and use cases.

What's the "kernel trick" and how is it useful?

The Kernel trick involves kernel functions that can enable in higher-dimension spaces without explicitly calculating the coordinates of points within that dimension: instead, kernel functions compute the inner products between the images of all pairs of data in a feature space. This allows them the very useful attribute of calculating the coordinates of higher dimensions while being computationally cheaper than the explicit calculation of said coordinates. Many algorithms can be expressed in terms of inner products. Using the kernel trick enables us effectively run algorithms in a high-dimensional space with lower-dimensional data.

These machine learning interview questions test your knowledge of programming principles you need to implement machine learning principles in practice. Machine learning interview questions tend to be technical questions that test your logic and programming skills: this section focuses more on the latter.

How do you handle missing or corrupted data in a dataset?

You could find missing/corrupted data in a dataset and either drop those rows or columns or decide to replace them with another value.

In Pandas, there are two very useful methods: `isnull()` and `dropna()` that will help you find columns of data with missing or corrupted data and drop those values. If you want to fill the invalid values with a placeholder value (for example, 0), you could use the `fillna()` method.

Do you have experience with Spark or big data tools for machine learning?

You'll want to get familiar with the meaning of big data for different companies and the different tools they'll want. Spark is the big data tool most in demand now, able to handle immense datasets with speed. Be honest if you don't have experience with the tools demanded, but also take a look at job descriptions and see what tools pop up: you'll want to invest in familiarizing yourself with them.

Pick an algorithm. Write the pseudo-code for a parallel implementation.

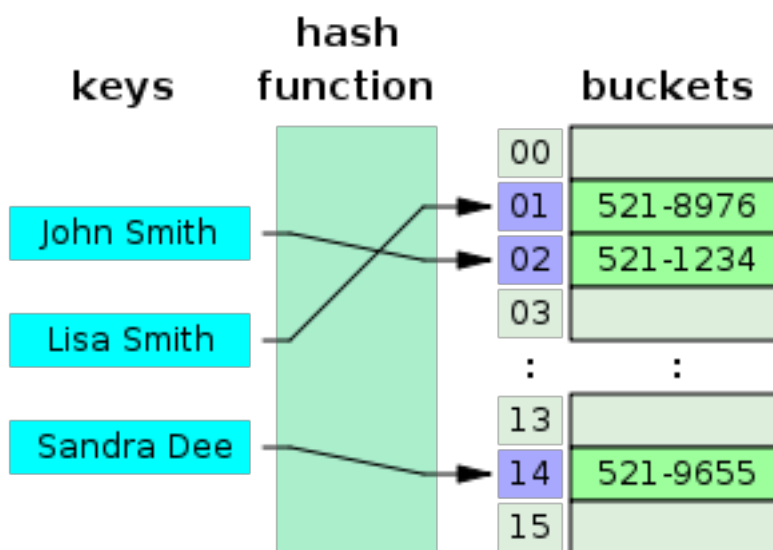
This kind of question demonstrates your ability to think in parallelism and how you could handle concurrency in programming implementations dealing with big data. Take a look at pseudocode frameworks such as Peril-L and visualization tools such as Web Sequence Diagrams to help you demonstrate your ability to write code that reflects parallelism.

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

An array is an ordered collection of objects. A linked list is a series of objects with pointers that direct how to process them sequentially. An array assumes that every element has the same size, unlike the linked list. A linked list can more easily grow organically: an array has to be pre-defined or re-defined for organic growth. Shuffling a linked list involves changing which points direct where — meanwhile, shuffling an array is more complex and takes more memory.

Describe a hash table.

A hash table is a data structure that produces an associative array. A key is mapped to certain values through the use of a hash function. They are often used for tasks such as database indexing.



Which data visualization libraries do you use? What are your thoughts on the best data visualization tools?

What's important here is to define your views on how to properly visualize data and your personal preferences when it comes to tools. Popular tools include R's ggplot, Python's seaborn and matplotlib, and tools such as Plot.ly and Tableau.

How would you implement a recommendation system for our company's users?

A lot of machine learning interview questions of this type will involve implementation of machine learning models to a company's problems. You'll have to research the company and its industry in-depth, especially the revenue drivers the company has, and the types of users the company takes on in the context of the industry it's in.

How can we use your machine learning skills to generate revenue?

This is a tricky question. The ideal answer would demonstrate knowledge of what drives the business and how your skills could relate. For example, if you were interviewing for music-streaming startup Spotify, you could remark that your skills at developing a better recommendation model would increase user retention, which would then increase revenue in the long run.

The startup metrics SlideShare linked above will help you understand exactly what performance indicators are important for startups and tech companies as they think about revenue and growth.

What do you think of our current data process?

This kind of question requires you to listen carefully and impart feedback in a manner that is constructive and insightful. Your interviewer is trying to gauge if you'd be a valuable member of their team and whether you grasp the nuances of why certain things are set the way they are in the company's data process based on company- or industry-specific conditions. They're trying to see if you can be an intellectual peer. Act accordingly.

What are the last machine learning papers you've read?

Keeping up with the latest scientific literature on machine learning is a must if you want to demonstrate interest in a machine learning position. This overview of [deep learning in Nature](#) by the scions of deep learning themselves (from Hinton to Bengio to LeCun) can be a good reference paper and an overview of what's happening in deep learning — and the kind of paper you might want to cite.

Do you have research experience in machine learning?

Related to the last point, most organizations hiring for machine learning positions will look for your formal experience in the field. Research papers, co-authored or supervised by leaders in the field, can make the difference between you being hired and not. Make sure you have a summary of your research experience and papers ready — and an explanation for your background and lack of formal research experience if you don't.

What are your favorite use cases of machine learning models?

The Quora thread above contains some examples, such as decision trees that categorize people into different tiers of intelligence based on IQ scores. Make sure that you have a few examples in mind and describe what resonated with you. It's important that you demonstrate an interest in how machine learning is implemented.

How would you approach the “Netflix Prize” competition?

The Netflix Prize was a famed competition where Netflix offered \$1,000,000 for a better collaborative filtering algorithm. The team that won called BellKor had a 10% improvement and used an ensemble of different methods to win. Some familiarity with the case and its solution will help demonstrate you’ve paid attention to machine learning for a while.

Where do you usually source datasets?

Machine learning interview questions like these try to get at the heart of your machine learning interest. Somebody who is truly passionate about machine learning will have gone off and done side projects on their own, and have a good idea of what great datasets are out there. If you’re missing any, check out [Quandl](#) for economic and financial data, and [Kaggle’s Datasets](#) collection for another great list.

How do you think Google is training data for self-driving cars?

Machine learning interview questions like this one really test your knowledge of different machine learning methods, and your inventiveness if you don’t know the answer. Google is currently using [recaptcha](#) to source labeled data on storefronts and traffic signs. They are also building on training data collected by Sebastian Thrun at GoogleX — some of which was obtained by his grad students driving buggies on desert dunes!

How would you simulate the approach AlphaGo took to beat Lee Sidol at Go?

AlphaGo beating Lee Sidol, the best human player at Go, in a best-of-five series was a truly seminal event in the history of machine learning and deep learning. The Nature paper above describes how this was accomplished with “Monte-Carlo tree search with deep neural networks that have been trained by supervised learning, from human expert games, and by reinforcement learning from games of self-play.”

Why do we need a validation set and test set? What is the difference between them?

When training a model, we divide the available data into three separate sets:

- The training dataset is used for fitting the model's parameters. However, the accuracy that we achieve on the training set is not reliable for predicting if the model will be accurate on new samples.
- The validation dataset is used to measure how well the model does on examples that weren't part of the training dataset. The metrics computed on the validation data can be used to tune the hyperparameters of the model. However, every time we evaluate the validation data and we make decisions based on those scores, we are leaking information from the validation data into our model. The more evaluations, the more information is leaked. So we can end up overfitting to the validation data, and once again the validation score won't be reliable for predicting the behaviour of the model in the real world.
- The test dataset is used to measure how well the model does on previously unseen examples. It should only be used once we have tuned the parameters using the validation set.

So, if we omit the test set and only use a validation set, the validation score won't be a good estimate of the generalization of the model.

What is stratified cross-validation and when should we use it?

Cross-validation is a technique for dividing data between training and validation sets. On typical cross-validation this split is done randomly. But in *stratified* cross-validation, the split preserves the ratio of the categories on both the training and validation datasets.

For example, if we have a dataset with 10% of category A and 90% of category B, and we use stratified cross-validation, we will have the same proportions in training and validation. In contrast, if we use simple cross-validation, in the worst case we may find that there are no samples of category A in the validation set.

Stratified cross-validation may be applied in the following scenarios:

- **On a dataset with multiple categories.** The smaller the dataset and the more imbalanced the categories, the more important it will be to use stratified cross-validation.
- **On a dataset with data of different distributions.** For example, in a dataset for autonomous driving, we may have images taken during the day and at night. If we do not ensure that both types are present in training and validation, we will have generalization problems.

Why do ensembles typically have higher scores than individual models?

An ensemble is the combination of multiple models to create a single prediction. The key idea for making better predictions is that the models should make different errors. That way the errors of one model will be compensated by the right guesses of the other models and thus the score of the ensemble will be higher.

We need diverse models for creating an ensemble. Diversity can be achieved by:

- Using different ML algorithms. For example, you can combine logistic regression, k-nearest neighbors, and decision trees.
- Using different subsets of the data for training. This is called *bagging*.
- Giving a different weight to each of the samples of the training set. If this is done iteratively, weighting the samples according to the errors of the ensemble, it's called *boosting*.

Many winning solutions to data science competitions are ensembles. However, in real-life machine learning projects, engineers need to find a balance between execution time and accuracy.

What is regularization? Can you give some examples of regularization techniques?

Regularization is any technique that aims to improve the validation score, sometimes at the cost of reducing the training score.

Some regularization techniques:

- **L1** tries to minimize the absolute value of the parameters of the model. It produces sparse parameters.
- **L2** tries to minimize the square value of the parameters of the model. It produces parameters with small values.
- **Dropout** is a technique applied to neural networks that randomly sets some of the neurons' outputs to zero during training. This forces the network to learn better representations of the data by preventing complex interactions between the neurons: Each neuron needs to learn useful features.
- **Early stopping** will stop training when the validation score stops improving, even when the training score may be improving. This prevents overfitting on the training dataset.

What is the curse of dimensionality? Can you list some ways to deal with it?

The curse of dimensionality is when the training data has a high feature count, but the dataset does not have enough samples for a model to learn correctly from so many features. For example, a training dataset of 100 samples with 100 features will be very hard to learn from because the model will find random relations between the features and the target. However, if we had a dataset of 100k samples with 100 features, the model could probably learn the correct relationships between the features and the target.

There are different options to fight the curse of dimensionality:

- **Feature selection.** Instead of using all the features, we can train on a smaller subset of features.

- **Dimensionality reduction.** There are many techniques that allow to reduce the dimensionality of the features. Principal component analysis (PCA) and using autoencoders are examples of dimensionality reduction techniques.
- **L1 regularization.** Because it produces sparse parameters, L1 helps to deal with high-dimensionality input.
- **Feature engineering.** It's possible to create new features that sum up multiple existing features. For example, we can get statistics such as the mean or median.

What is an imbalanced dataset? Can you list some ways to deal with it?

An imbalanced dataset is one that has different proportions of target categories. For example, a dataset with medical images where we have to detect some illness will typically have many more negative samples than positive samples—say, 98% of images are without the illness and 2% of images are with the illness.

There are different options to deal with imbalanced datasets:

- **Oversampling or under sampling.** Instead of sampling with a uniform distribution from the training dataset, we can use other distributions, so the model sees a more balanced dataset.
- **Data augmentation.** We can add data in the less frequent categories by modifying existing data in a controlled way. In the example dataset, we could flip the images with illnesses, or add noise to copies of the images in such a way that the illness remains visible.
- **Using appropriate metrics.** In the example dataset, if we had a model that always made negative predictions, it would achieve a precision of 98%. There are other metrics such as precision, recall, and F-score that describe the accuracy of the model better when using an imbalanced dataset.

Can you explain the differences between supervised, unsupervised, and reinforcement learning?

In supervised learning, we train a model to learn the relationship between input data and output data. We need to have labeled data to be able to do supervised learning.

With unsupervised learning, we only have unlabeled data. The model learns a representation of the data. Unsupervised learning is frequently used to initialize the parameters of the model when we have a lot of unlabeled data and a small fraction of labeled data. We first train an unsupervised model and, after that, we use the weights of the model to train a supervised model.

In reinforcement learning, the model has some input data and a reward depending on the output of the model. The model learns a policy that maximizes the reward. Reinforcement learning has been applied successfully to strategic games such as Go and even classic Atari video games.

What are some factors that explain the success and recent rise of deep learning?

The success of deep learning in the past decade can be explained by three main factors:

1. **More data.** The availability of massive labeled datasets allows us to train models with more parameters and achieve state-of-the-art scores. Other ML algorithms do not scale as well as deep learning when it comes to dataset size.
2. **GPU.** Training models on a GPU can reduce the training time by orders of magnitude compared to training on a CPU. Currently, cutting-edge models are trained on multiple GPUs or even on specialized hardware.
3. **Improvements in algorithms.** ReLU activation, dropout, and complex network architectures have also been very significant factors.

What is data augmentation? Can you give some examples?

Data augmentation is a technique for synthesizing new data by modifying existing data in such a way that the target is not changed, or it is changed in a known way.

Computer vision is one of fields where data augmentation is very useful. There are many modifications that we can do to images:

- Resize
- Horizontal or vertical flip
- Rotate
- Add noise
- Deform
- Modify colors

Each problem needs a customized data augmentation pipeline. For example, on OCR, doing flips will change the text and won't be beneficial; however, resizes and small rotations may help.

What are convolutional networks? Where can we use them?

Convolutional networks are a class of neural network that use convolutional layers instead of fully connected layers. On a fully connected layer, all the output units have weights connecting to all the input units. On a convolutional layer, we have some weights that are repeated over the input.

The advantage of convolutional layers over fully connected layers is that the number of parameters is far smaller. This results in better generalization of the model. For example, if we want to learn a transformation from a 10x10 image to another 10x10 image, we will need 10,000 parameters if using a fully connected layer. If we use two convolutional layers, the first one having nine filters and the second one having one filter, with a kernel size of 3x3, we will have only 90 parameters.

Convolutional networks are applied where data has a clear dimensionality structure. Time series analysis is an example where one-dimensional convolutions are used; for images, 2D convolutions are used; and for volumetric data, 3D convolutions are used.

You are given a train data set having 1000 columns and 1 million rows. The data set is based on a classification problem. Your manager has asked you to reduce the dimension of this data so that model computation time can be reduced. Your machine has memory constraints. What would you do? (You are free to make practical assumptions.)

Processing a high dimensional data on a limited memory machine is a strenuous task, your interviewer would be fully aware of that. Following are the methods you can use to tackle such situation:

- a. Since we have lower RAM, we should close all other applications in our machine, including the web browser, so that most of the memory can be put to use.
- b. We can randomly sample the data set. This means, we can create a smaller data set, let's say, having 1000 variables and 300000 rows and do the computations.
- c. To reduce dimensionality, we can separate the numerical and categorical variables and remove the correlated variables. For numerical variables, we'll use correlation. For categorical variables, we'll use chi-square test.
- d. Also, we can use PCA and pick the components which can explain the maximum variance in the data set.
- e. Using online learning algorithms like Vowpal Wabbit (available in Python) is a possible option.
- f. Building a linear model using Stochastic Gradient Descent is also helpful.
- g. We can also apply our business understanding to estimate which all predictors can impact the response variable. But, this is an intuitive approach, failing to identify useful predictors might result in significant loss of information.

Is rotation necessary in PCA? If yes, Why? What will happen if you don't rotate the components?

Yes, rotation (orthogonal) is necessary because it maximizes the difference between variance captured by the component. This makes the components easier to interpret. Not to forget, that's the motive of doing PCA where, we aim to select fewer components (than features) which can explain the maximum variance in the data set. By doing rotation, the relative location of the components doesn't change, it only changes the actual coordinates of the points.

If we don't rotate the components, the effect of PCA will diminish and we'll have to select a greater number of components to explain variance in the data set.

You are given a data set. The data set has missing values which spread along 1 standard deviation from the median. What percentage of data would remain unaffected? Why?

This question has enough hints for you to start thinking! Since, the data is spread across median, let's assume it's a normal distribution. We know, in a normal distribution, ~68% of the data lies in 1 standard deviation from mean (or mode, median), which leaves ~32% of the data unaffected. Therefore, ~32% of the data would remain unaffected by missing values.

You are given a data set on cancer detection. You've built a classification model and achieved an accuracy of 96%. Why shouldn't you be happy with your model performance? What can you do about it?

If you have worked on enough data sets, you should deduce that cancer detection results in imbalanced data. In an imbalanced data set, accuracy should not be used as a measure of performance because 96% (as given) might only be predicting majority class correctly, but our class of interest is minority class (4%) which is the people who actually got diagnosed with cancer. Hence, in order to evaluate model performance, we should use Sensitivity (True Positive Rate), Specificity (True Negative Rate), F measure to determine class wise performance of the classifier. If the minority class performance is found to be poor, we can undertake the following steps:

We can use under sampling, oversampling or SMOTE to make the data balanced.

- a. We can alter the prediction threshold value by doing probability calibration and finding a optimal threshold using AUC-ROC curve.
- b. We can assign weight to classes such that the minority classes get larger weight.
- c. We can also use anomaly detection.

Why is naive Bayes so 'naive'?

Naive Bayes is so 'naive' because it assumes that all of the features in a data set are equally important and independent. As we know, these assumptions are rarely true in real world scenario.

Explain prior probability, likelihood and marginal likelihood in context of Naïve Bayes algorithm?

Prior probability is nothing but, the proportion of dependent (binary) variable in the data set. It is the closest guess you can make about a class, without any further information. For example: In a data set, the dependent variable is binary (1 and 0). The proportion of 1 (spam) is 70% and 0 (not spam) is 30%. Hence, we can estimate that there are 70% chances that any new email would be classified as spam.

Likelihood is the probability of classifying a given observation as 1 in presence of some other variable. For example: The probability that the word 'FREE' is used in previous spam message is likelihood. Marginal likelihood is, the probability that the word 'FREE' is used in any message.

You are working on a time series data set. Your manager has asked you to build a high accuracy model. You start with the decision tree algorithm, since you know it works fairly well on all kinds of data. Later, you tried a time series regression model and got higher accuracy than decision tree model. Can this happen? Why?

Time series data is known to possess linearity. On the other hand, a decision tree algorithm is known to work best to detect non – linear interactions. The reason why decision tree failed to provide robust predictions because it couldn't map the linear relationship as good as a regression model did. Therefore, we learned that, a linear

regression model can provide robust prediction given the data set satisfies its linearity assumptions.

You are assigned a new project which involves helping a food delivery company save more money. The problem is, company's delivery team aren't able to deliver food on time. As a result, their customers get unhappy. And, to keep them happy, they end up delivering food for free. Which machine learning algorithm can save them?

You might have started hopping through the list of ML algorithms in your mind. But, wait! Such questions are asked to test your machine learning fundamentals.

This is not a machine learning problem. This is a route optimization problem. A machine learning problem consist of three things:

1. There exists a pattern.
2. You cannot solve it mathematically (even by writing exponential equations).
3. You have data on it.

Always look for these three factors to decide if machine learning is a tool to solve a particular problem.

You came to know that your model is suffering from low bias and high variance. Which algorithm should you use to tackle it? Why?

Low bias occurs when the model's predicted values are near to actual values. In other words, the model becomes flexible enough to mimic the training data distribution. While it sounds like great achievement, but not to forget, a flexible model has no generalization capabilities. It means, when this model is tested on an unseen data, it gives disappointing results.

In such situations, we can use bagging algorithm (like random forest) to tackle high variance problem. Bagging algorithms divides a data set into subsets made with repeated randomized sampling. Then, these samples are used to generate a set of models using a single learning algorithm. Later, the model predictions are combined using voting (classification) or averaging (regression).

Also, to combat high variance, we can:

- a. Use regularization technique, where higher model coefficients get penalized, hence lowering model complexity.
- b. Use top n features from variable importance chart. May be, with all the variable in the data set, the algorithm is having difficulty in finding the meaningful signal.

You are given a data set. The data set contains many variables, some of which are highly correlated, and you know about it. Your manager has asked you to run PCA. Would you remove correlated variables first? Why?

Chances are, you might be tempted to say No, but that would be incorrect. Discarding correlated variables have a substantial effect on PCA because, in presence of correlated variables, the variance explained by a particular component gets inflated.

For example: You have 3 variables in a data set, of which 2 are correlated. If you run PCA on this data set, the first principal component would exhibit twice the variance than it would exhibit with uncorrelated variables. Also, adding correlated variables lets PCA put more importance on those variables, which is misleading.

After spending several hours, you are now anxious to build a high accuracy model. As a result, you build 5 GBM models, thinking a boosting algorithm would do the magic. Unfortunately, neither of models could perform better than benchmark score. Finally, you decided to combine those models. Though, ensembled models are known to return high accuracy, but you are unfortunate. Where did you miss?

As we know, ensemble learners are based on the idea of combining weak learners to create strong learners. But these learners provide superior result when the combined models are uncorrelated. Since, we have used 5 GBM models and got no accuracy improvement, suggests that the models are correlated. The problem with correlated models is, all the models provide same information.

For example: If model 1 has classified User1122 as 1, there are high chances model 2 and model 3 would have done the same, even if its actual value is 0. Therefore,

ensemble learners are built on the premise of combining weak uncorrelated models to obtain better predictions.

How is kNN different from K-means clustering?

Don't get misled by 'k' in their names. You should know that the fundamental difference between both these algorithms is, K-means is unsupervised in nature and kNN is supervised in nature. K-means is a clustering algorithm. kNN is a classification (or regression) algorithm.

K-means algorithm partitions a data set into clusters such that a cluster formed is homogeneous and the points in each cluster are close to each other. The algorithm tries to maintain enough separability between these clusters. Due to unsupervised nature, the clusters have no labels.

kNN algorithm tries to classify an unlabeled observation based on its k (can be any number) surrounding neighbors. It is also known as lazy learner because it involves minimal training of model. Hence, it doesn't use training data to make generalization on unseen data set.

How is True Positive Rate and Recall related? Write the equation.

True Positive Rate = Recall. Yes, they are equal having the formula $(TP/TP + FN)$.

You have built a multiple regression model. Your model R^2 isn't as good as you wanted. For improvement, you remove the intercept term, your model R^2 becomes 0.8 from 0.3. Is it possible? How?

Yes, it is possible. We need to understand the significance of intercept term in a regression model. The intercept term shows model prediction without any independent variable i.e. mean prediction. The formula of $R^2 = 1 - \frac{\sum(y - y')^2}{\sum(y - y_{\text{mean}})^2}$ where y' is predicted value.

When intercept term is present, R^2 value evaluates your model wrt. to the mean model. In absence of intercept term (y_{mean}), the model can make no such evaluation, with large denominator, $\frac{\sum(y - y')^2}{\sum(y)^2}$ equation's value becomes smaller than actual, resulting in higher R^2 .

After analyzing the model, your manager has informed that your regression model is suffering from multicollinearity. How would you check if he's true? Without losing any information, can you still build a better model?

To check multicollinearity, we can create a correlation matrix to identify & remove variables having correlation above 75% (deciding a threshold is subjective). In addition, we can use calculate VIF (variance inflation factor) to check the presence of multicollinearity. VIF value ≤ 4 suggests no multicollinearity whereas a value of ≥ 10 implies serious multicollinearity. Also, we can use tolerance as an indicator of multicollinearity.

But, removing correlated variables might lead to loss of information. In order to retain those variables, we can use penalized regression models like ridge or lasso regression. Also, we can add some random noise in correlated variable so that the variables become different from each other. But, adding noise might affect the prediction accuracy, hence this approach should be carefully used.

When is Ridge regression favorable over Lasso regression?

You can quote ISLR's authors Hastie, Tibshirani who asserted that, in presence of few variables with medium / large sized effect, use lasso regression. In presence of many variables with small / medium sized effect, use ridge regression.

Conceptually, we can say, lasso regression (L1) does both variable selection and parameter shrinkage, whereas Ridge regression only does parameter shrinkage and end up including all the coefficients in the model. In presence of correlated variables, ridge regression might be the preferred choice. Also, ridge regression works best in situations where the least square estimates have higher variance. Therefore, it depends on our model objective.

Rise in global average temperature led to decrease in number of pirates around the world. Does that mean that decrease in number of pirates caused the climate change?

After reading this question, you should have understood that this is a classic case of "causation and correlation". No, we can't conclude that decrease in number of

pirates caused the climate change because there might be other factors (lurking or confounding variables) influencing this phenomenon.

Therefore, there might be a correlation between global average temperature and number of pirates but based on this information we can't say that pirates died because of rise in global average temperature.

While working on a data set, how do you select important variables? Explain your methods.

Following are the methods of variable selection you can use:

- a. Remove the correlated variables prior to selecting important variables
- b. Use linear regression and select variables based on p values
- c. Use Forward Selection, Backward Selection, Stepwise Selection
- d. Use Random Forest, Xgboost and plot variable importance chart
- e. Use Lasso Regression
- f. Measure information gain for the available set of features and select top n features accordingly.

What is the difference between covariance and correlation?

Correlation is the standardized form of covariance.

Covariances are difficult to compare. For example: if we calculate the covariances of salary (\$) and age (years), we'll get different covariances which can't be compared because of having unequal scales. To combat such situation, we calculate correlation to get a value between -1 and 1, irrespective of their respective scale.

Is it possible to capture the correlation between continuous and categorical variable? If yes, how?

Yes, we can use ANCOVA (analysis of covariance) technique to capture association between continuous and categorical variables.

Both being tree-based algorithm, how is random forest different from Gradient boosting algorithm (GBM)?

The fundamental difference is, random forest uses bagging technique to make predictions. GBM uses boosting techniques to make predictions.

In bagging technique, a data set is divided into n samples using randomized sampling. Then, using a single learning algorithm a model is build on all samples. Later, the resultant predictions are combined using voting or averaging. Bagging is done in parallel. In boosting, after the first round of predictions, the algorithm weighs misclassified predictions higher, such that they can be corrected in the succeeding round. This sequential process of giving higher weights to misclassified predictions continue until a stopping criterion is reached.

Random forest improves model accuracy by reducing variance (mainly). The trees grown are uncorrelated to maximize the decrease in variance. On the other hand, GBM improves accuracy by reducing both bias and variance in a model.

Running a binary classification tree algorithm is the easy part. Do you know how does a tree splitting takes place i.e. how does the tree decide which variable to split at the root node and succeeding nodes?

A classification trees makes decision based on Gini Index and Node Entropy. In simple words, the tree algorithm finds the best possible feature which can divide the data set into purest possible children nodes.

Gini index says, if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure. We can calculate Gini as following:

1. Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure ($p^2 + q^2$).
2. Calculate Gini for split using weighted Gini score of each node of that split

Entropy is the measure of impurity as given by (for binary class):

$$\text{Entropy} = -p \log_2 p - q \log_2 q$$

Here p and q is probability of success and failure respectively in that node. Entropy is zero when a node is homogeneous. It is maximum when a both the classes are present in a node at 50% – 50%. Lower entropy is desirable.

You've built a random forest model with 10000 trees. You got delighted after getting training error as 0.00. But, the validation error is 34.23. What is going on? Haven't you trained your model perfectly?

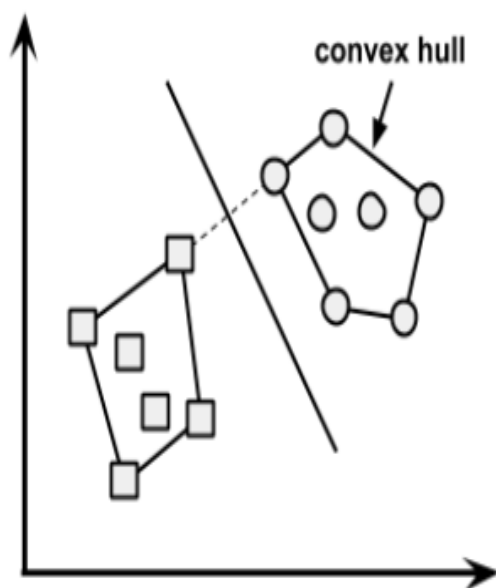
The model has overfitted. Training error 0.00 means the classifier has mimicked the training data patterns to an extent, that they are not available in the unseen data. Hence, when this classifier was run on unseen sample, it couldn't find those patterns and returned prediction with higher error. In random forest, it happens when we use larger number of trees than necessary. Hence, to avoid these situations, we should tune number of trees using cross validation.

You've got a data set to work having p (no. of variable) $>$ n (no. of observation). Why is OLS as bad option to work with? Which techniques would be best to use? Why?

In such high dimensional data sets, we can't use classical regression techniques, since their assumptions tend to fail. When $p > n$, we can no longer calculate a unique least square coefficient estimate, the variances become infinite, so OLS cannot be used at all.

To combat this situation, we can use penalized regression methods like lasso, LARS, ridge which can shrink the coefficients to reduce variance. Precisely, ridge regression works best in situations where the least square estimates have higher variance.

Among other methods include subset regression, forward stepwise regression.



What is convex hull ? (Hint: Think SVM)

In case of linearly separable data, convex hull represents the outer boundaries of the two group of data points. Once convex hull is created, we get maximum margin hyperplane (MMH) as a perpendicular bisector between two convex hulls. MMH is the line which attempts to create greatest separation between two groups.

We know that one hot encoding increasing the dimensionality of a data set. But, label encoding doesn't. How?

Don't get baffled at this question. It's a simple question asking the difference between the two.

Using one hot encoding, the dimensionality (a.k.a features) in a data set get increased because it creates a new variable for each level present in categorical variables. For example let's say we have a variable 'color'. The variable has 3 levels namely Red, Blue and Green. One hot encoding 'color' variable will generate three new variables as **Color.Red**, **Color.Blue** and **Color.Green** containing 0 and 1 value.

In label encoding, the levels of a categorical variables gets encoded as 0 and 1, so no new variable is created. Label encoding is majorly used for binary variables.

What cross validation technique would you use on time series data set? Is it k-fold or LOOCV?

Answer: Neither.

In time series problem, k fold can be troublesome because there might be some pattern in year 4 or 5 which is not in year 3. Resampling the data set will separate these trends, and we might end up validation on past years, which is incorrect. Instead, we can use forward chaining strategy with 5 fold as shown below:

- a. fold 1: training [1], test [2]
- b. fold 2: training [1 2], test [3]
- c. fold 3: training [1 2 3], test [4]
- d. fold 4: training [1 2 3 4], test [5]
- e. fold 5: training [1 2 3 4 5], test [6]

You are given a data set consisting of variables having more than 30% missing values? Let's say, out of 50 variables, 8 variables have missing values higher than 30%. How will you deal with them?

We can deal with them in the following ways:

- a. Assign a unique category to missing values, who knows the missing values might decipher some trend
- b. We can remove them blatantly.
- c. Or, we can sensibly check their distribution with the target variable, and if found any pattern we'll keep those missing values and assign them a new category while removing others.

'People who bought this, also bought...' recommendations seen on amazon is a result of which algorithm?

The basic idea for this kind of recommendation engine comes from collaborative filtering.

Collaborative Filtering algorithm considers "User Behavior" for recommending items. They exploit behavior of other users and items in terms of transaction

history, ratings, selection and purchase information. Other users behavior and preferences over the items are used to recommend items to the new users. In this case, features of the items are not known.

What do you understand by Type I vs Type II error?

Type I error is committed when the null hypothesis is true and we reject it, also known as a 'False Positive'. Type II error is committed when the null hypothesis is false and we accept it, also known as 'False Negative'.

In the context of confusion matrix, we can say Type I error occurs when we classify a value as positive (1) when it is actually negative (0). Type II error occurs when we classify a value as negative (0) when it is actually positive.

You are working on a classification problem. For validation purposes, you've randomly sampled the training data set into train and validation. You are confident that your model will work incredibly well on unseen data since your validation accuracy is high. However, you get shocked after getting poor test accuracy. What went wrong?

In case of classification problem, we should always use stratified sampling instead of random sampling. A random sampling doesn't take into consideration the proportion of target classes. On the contrary, stratified sampling helps to maintain the distribution of target variable in the resultant distributed samples also.

You have been asked to evaluate a regression model based on R^2 , adjusted R^2 and tolerance. What will be your criteria?

Tolerance ($1 / VIF$) is used as an indicator of multicollinearity. It is an indicator of percent of variance in a predictor which cannot be accounted by other predictors. Large values of tolerance is desirable.

We will consider adjusted R^2 as opposed to R^2 to evaluate model fit because R^2 increases irrespective of improvement in prediction accuracy as we add more variables. But, adjusted R^2 would only increase if an additional variable improves the accuracy of model, otherwise stays same. It is difficult to commit a general threshold value for adjusted R^2 because it varies between data sets. For example: a gene mutation data set might result in lower adjusted R^2 and still provide fairly

good predictions, as compared to a stock market data where lower adjusted R^2 implies that model is not good.

In k-means or kNN, we use Euclidean distance to calculate the distance between nearest neighbors. Why not Manhattan distance?

We don't use Manhattan distance because it calculates distance horizontally or vertically only. It has dimension restrictions. On the other hand, euclidean metric can be used in any space to calculate distance. Since, the data points can be present in any dimension, Euclidean distance is a more viable option.

Example: Think of a chess board, the movement made by a bishop or a rook is calculated by Manhattan distance because of their respective vertical & horizontal movements.

Explain machine learning to me like a 5 year old.

It's simple. It's just like how babies learn to walk. Every time they fall down, they learn (unconsciously) & realize that their legs should be straight and not in a bend position. The next time they fall down, they feel pain. They cry. But, they learn 'not to stand like that again'. In order to avoid that pain, they try harder. To succeed, they even seek support from the door or wall or anything near them, which helps them stand firm.

I know that a linear regression model is generally evaluated using Adjusted R^2 or F value. How would you evaluate a logistic regression model?

We can use the following methods:

- a. Since logistic regression is used to predict probabilities, we can use AUC-ROC curve along with confusion matrix to determine its performance.
- b. Also, the analogous metric of adjusted R^2 in logistic regression is AIC. AIC is the measure of fit which penalizes model for the number of model coefficients. Therefore, we always prefer model with minimum AIC value.
- c. Null Deviance indicates the response predicted by a model with nothing but an intercept. Lower the value, better the model. Residual deviance indicates the response predicted by a model on adding independent variables. Lower the value, better the model.

Considering the long list of machine learning algorithm, given a data set, how do you decide which one to use?

You should say, the choice of machine learning algorithm solely depends of the type of data. If you are given a data set which is exhibits linearity, then linear regression would be the best algorithm to use. If you give to work on images, audios, then neural network would help you to build a robust model.

If the data comprises of nonlinear interactions, then a boosting or bagging algorithm should be the choice. If the business requirement is to build a model which can be deployed, then we'll use regression or a decision tree model (easy to interpret and explain) instead of black box algorithms like SVM, GBM etc.

In short, there is no one master algorithm for all situations. We must be scrupulous enough to understand which algorithm to use.

Do you suggest that treating a categorical variable as continuous variable would result in a better predictive model?

For better predictions, categorical variable can be considered as a continuous variable only when the variable is ordinal in nature.

When does regularization becomes necessary in Machine Learning?

Regularization becomes necessary when the model begins to overfit / underfit. This technique introduces a cost term for bringing in more features with the objective function. Hence, it tries to push the coefficients for many variables to zero and hence reduce cost term. This helps to reduce model complexity so that the model can become better at predicting (generalizing).

What do you understand by Bias Variance trade off?

The error emerging from any model can be broken down into three components mathematically. Following are these components:

$$Err(x) = \left(E[\hat{f}(x)] - f(x) \right)^2 + E \left[\hat{f}(x) - E[\hat{f}(x)] \right]^2 + \sigma_e^2$$

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

Bias error is useful to quantify how much on an average are the predicted values different from the actual value. A high bias error means we have a under-performing model which keeps on missing important trends. **Variance** on the other side quantifies how are the prediction made on same observation different from each other. A high variance model will over-fit on your training population and perform badly on any observation beyond training.

OLS is to linear regression. Maximum likelihood is to logistic regression. Explain the statement.

OLS and Maximum likelihood are the methods used by the respective regression methods to approximate the unknown parameter (coefficient) value. In simple words,

Ordinary least square (OLS) is a method used in linear regression which approximates the parameters resulting in minimum distance between actual and predicted values. Maximum Likelihood helps in choosing the the values of parameters which maximizes the likelihood that the parameters are most likely to produce observed data.

What do you understand by Machine Learning?

Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

Give an example that explains Machine Learning in industry.

Robots are replacing humans in many areas. It is because robots are programmed such that they can perform the task based on data they gather from sensors. They learn from the data and behaves intelligently.

What are the different Algorithm techniques in Machine Learning?

- a. Supervised
- b. Unsupervised
- c. Semi-supervised
- d. Transduction
- e. Learning to Learn

What is the difference between supervised and unsupervised machine learning?

This is the basic Machine Learning Interview Questions asked in an interview. A Supervised learning is a process where it requires training labeled data While Unsupervised learning it doesn't require data labeling.

What is the function of Unsupervised Learning?

The function of Unsupervised Learning is as below:

- a. Find clusters of the data of the data
- b. Find low-dimensional representations of the data
- c. Find interesting directions in data
- d. Interesting coordinates and correlations
- e. Find novel observations

What is the function of Supervised Learning?

The function of Supervised Learning are as below:

- a. Classifications
- b. Speech recognition
- c. Regression
- d. Predict time series
- e. Annotate strings

What are the advantages of Naive Bayes?

The advantages of Naive Bayes are:

- a. The classifier will converge quicker than discriminative models
- b. It cannot learn the interactions between features

What are the disadvantages of Naive Bayes?

The disadvantages of Naive Bayes are:

- a. It is because the problem arises for continuous features
- b. It makes a very strong assumption on the shape of your data distribution
- c. It can also happen because of data scarcity

Why is naive Bayes so naive?

Naive Bayes is so naive because it assumes that all of the features in a dataset are equally important and independent.

What is Overfitting in Machine Learning?

This is the popular Machine Learning Interview Questions asked in an interview. Overfitting in Machine Learning is defined as when a statistical model describes random error or noise instead of the underlying relationship or when a model is excessively complex.

What are the conditions when Overfitting happens?

One of the important reason and possibility of overfitting is because the criteria used for training the model is not the same as the criteria used to judge the efficacy of a model.

How can you avoid overfitting?

We can avoid overfitting by using:

- a. Lots of data
- b. Cross-validation

What are the five popular algorithms for Machine Learning?

Below is the list of five popular algorithms of Machine Learning:

- a. Decision Trees
- b. Probabilistic networks
- c. Nearest Neighbor
- d. Support vector machines
- e. Neural Networks

What are the different use cases where machine learning algorithms can be used?

The different use cases where machine learning algorithms can be used are as follows:

- a. Fraud Detection
- b. Face detection
- c. Natural language processing
- d. Market Segmentation
- e. Text Categorization
- f. Bioinformatics

What are parametric models and Non-Parametric models?

Parametric models are those with a finite number of parameters and to predict new data, you only need to know the parameters of the model. Non Parametric models are those with an unbounded number of parameters, allowing for more flexibility and to predict new data, you need to know the parameters of the model and the state of the data that has been observed.

What are the three stages to build the hypotheses or model in machine learning?

This is the frequently asked Machine Learning Interview Questions in an interview. The three stages to build the hypotheses or model in machine learning are:

- a. Model building
- b. Model testing
- c. Applying the model

What is Inductive Logic Programming in Machine Learning (ILP)?

Inductive Logic Programming (ILP) is a [subfield of machine learning](#) which uses logical [programming](#) representing background knowledge and examples.

What is the difference between classification and regression?

The difference between classification and regression are as follows:

- a. Classification is about identifying group membership while regression technique involves predicting a response.
- b. Classification and Regression techniques are related to prediction
- c. Classification predicts the belonging to a class whereas regression predicts the value from a continuous set

- d. Classification technique is preferred over regression when the results of the model need to return the belongingness of data points in a dataset with specific explicit categories

What is the difference between inductive machine learning and deductive machine learning?

The difference between inductive machine learning and deductive machine learning are as follows:

Machine learning where the model learns by examples from a set of observed instances to draw a generalized conclusion whereas in deductive learning the model first draws the conclusion and then the conclusion is drawn.

What are the advantages decision trees?

The advantages decision trees are:

- Decision trees are easy to interpret
- Nonparametric
- There are relatively few parameters to tune

What are the disadvantages of decision trees?

Decision trees are prone to be overfit. However, this can be addressed by ensemble methods like random forests or boosted trees.

What are the advantages of neural networks?

This is the advanced Machine Learning Interview Questions asked in an interview. Neural networks have led to performance breakthroughs for unstructured datasets such as images, audio, and video. Their incredible flexibility allows them to learn patterns that no other Machine Learning algorithm can learn.

What are the disadvantages of neural networks?

Neural Network requires a large amount of training data to converge. It's also difficult to pick the right architecture, and the internal "hidden" layers are incomprehensible.

What is the difference between L1 and L2 regularization?

The difference between L1 and L2 regularization are as follows:

- a. L1/Laplace tends to tolerate both large values as well as very small values of coefficients more than L2/Gaussian
- b. L1 can yield sparse models while L2 doesn't
- c. L1 and L2 regularization prevents overfitting by shrinking on the coefficients
- d. L2 (Ridge) shrinks all the coefficient by the same proportions but eliminates none, while L1 (Lasso) can shrink some coefficients to zero, performing variable selection
- e. L1 is the first-moment norm $|x_1 - x_2|$ that is simply the absolute distance between two points where L2 is second-moment norm corresponding to Euclidean Distance that is $|x_1 - x_2|^2$
- f. L2 regularization tends to spread error among all the terms, while L1 is more binary/sparse