Tirthajyoti Sarkar Follow

Editorial Associate "Towards Data Science" | Sr. Principal Engineer | Ph.D. in EE (U. of Iilinois)| AI/ML certification, Stanford, MIT | Open-source contributor

Aug 25 · 6 min read

25 fun questions for a machine learning interview

Can machine learning interview questions be funny and deep at the same time?

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Image source: https://xkcd.com/1838/

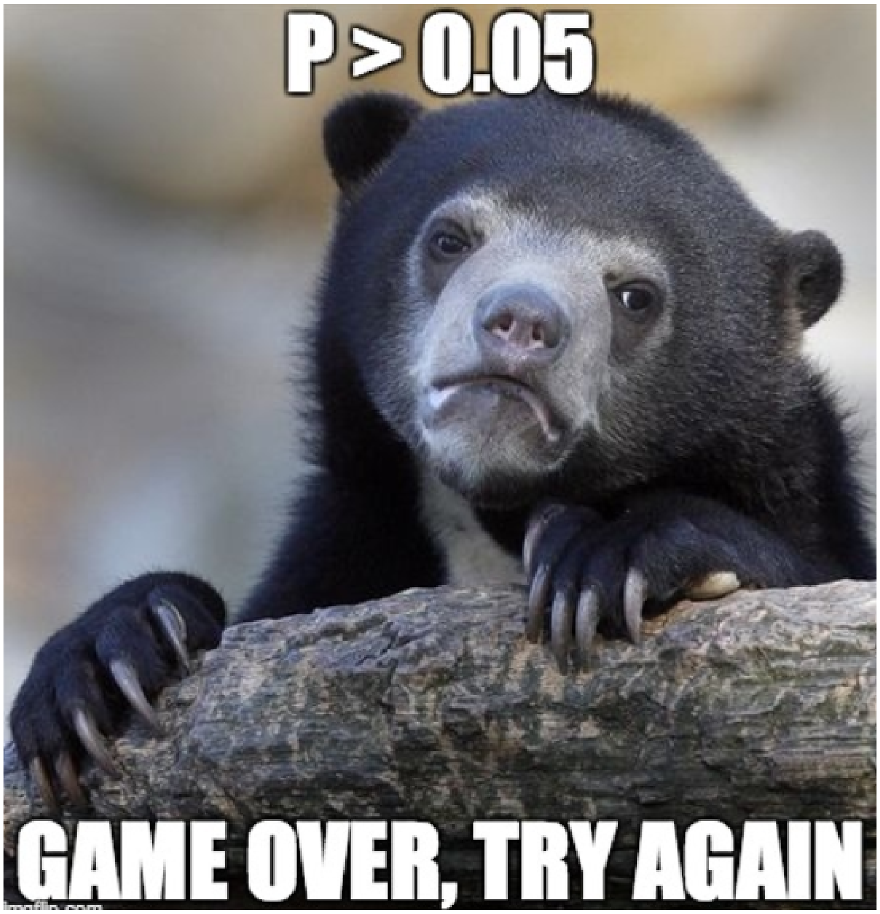
Many of the data scientists study machine learning (ML) mostly from a data practitioner’s point of view. Consequently, it is possible that we focus on learning about as many new packages, frameworks, techniques as possible and concentrate less on deep examination of the core theoretical aspects. And, here my definition of *machine learning* encompasses all of the standard statistical learning (i.e. it does not constitute only *deep learning*).

However, probing and contemplating with some effort, one can come up with so many wonderful ML questions, which, when answered and analyzed, can reveal deeper aspects beautifully. Basically, these questions may help us to get our head out of this pile shown above. We just do not want to stir a data set all day long, we want to dive deep into the properties, quirks, and intricacies of machine learning techniques and embrace them…

After all, there are plenty of article on the internet about "*standard interview questions for machine learning*". Can we do little different and interesting?

> **Disclaimer***: I am just posting the questions for thinking and stimulating discussion. No ready-made answer is given. Some questions have a hint but really they are for more discussion rather than a definitive answer. Each question is worth discussing in more detail. There is no set answer. Some questions are contrived, some are just for fun. Just enjoy :-) To boot, I have a funny meme inserted after every 5th question…*

Fun Questions



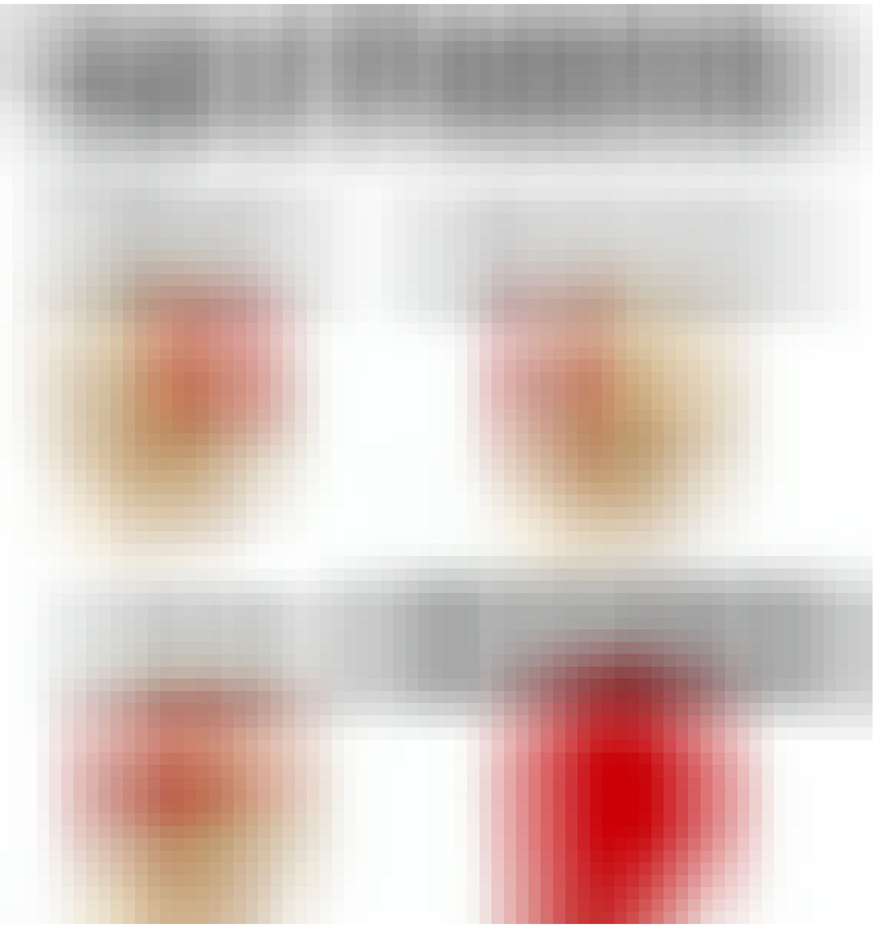
- I built a linear regression model showing 95% confidence interval. Does it mean that *there is a 95% chance that my model coefficients are the true estimate* of the function I am trying to approximate? (***Hint***: It actually means 95% of the time…)

- What is a similarity between Hadoop file system and *k*-nearest neighbor algorithm? (***Hint***: *'lazy'*)

- Which structure is more powerful in terms of expressiveness (i.e. it can represent a given Boolean function, accurately)—a *single-layer perceptron* or a *2-layer decision tree*? (***Hint***: XOR)

- And, which one is more powerful—a *2 layer decision tree* or a *2-layer neural network without any activation function*? (***Hint***: non-linearity?)

- Can a neural network be used as a tool for dimensionality reduction? Explain how.



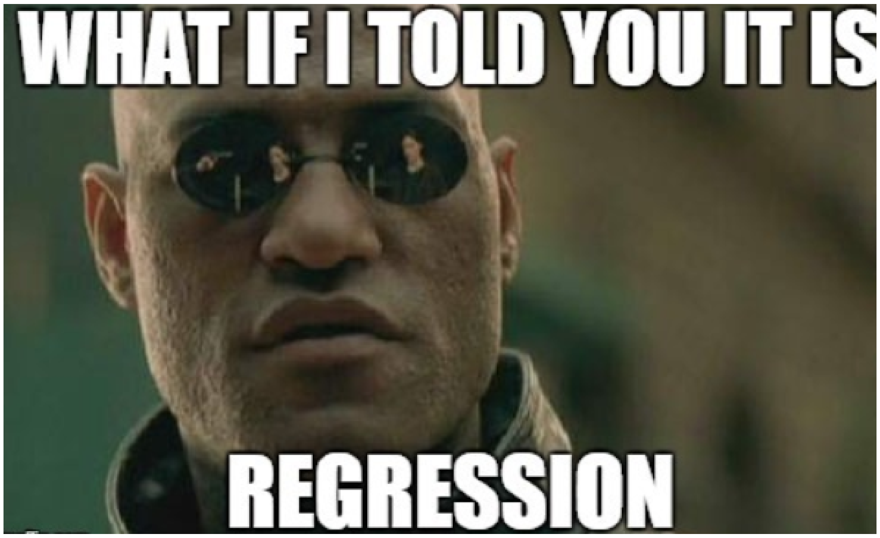
- Everybody maligns and belittles the intercept term in a linear regression model. Tell me one of its utilities. (***Hint***: noise/garbage collector)

- LASSO regularization reduces coefficients to exact zero. Ridge regression reduces them to very small but non-zero value. Can you explain the difference intuitively from the plots of two simple function|x| and x^2? (***Hint***: Those sharp corners in the |x| plot)

- Let's say that you don't know anything about the distribution from which a data set (continuous valued numbers) came and you are forbidden to assume that it is Normal Gaussian. Show by simplest possible arguments that no matter what the true distribution is, you can guarantee that ~89% of the data will lie within +/- 3 standard deviations away from the mean (***Hint***: Markov's Ph.D. adviser)

- Majority of machine learning algorithms involve some kind of matrix manipulation like multiplication or inversion. Give a simple mathematical argument why a mini-batch version of such ML algorithm might be computationally more efficient than a training with full data set. (***Hint***: Time complexity of matrix multiplication…)

- Don’t you think that a time series is a really simple linear regression problem with only one response variable and a single predictor—time? What’s the problem with a linear regression fit (not necessarily with a single linear term but even with polynomial degree terms) approach in case of a time series data? (***Hint***: Past is an indicator of future…)



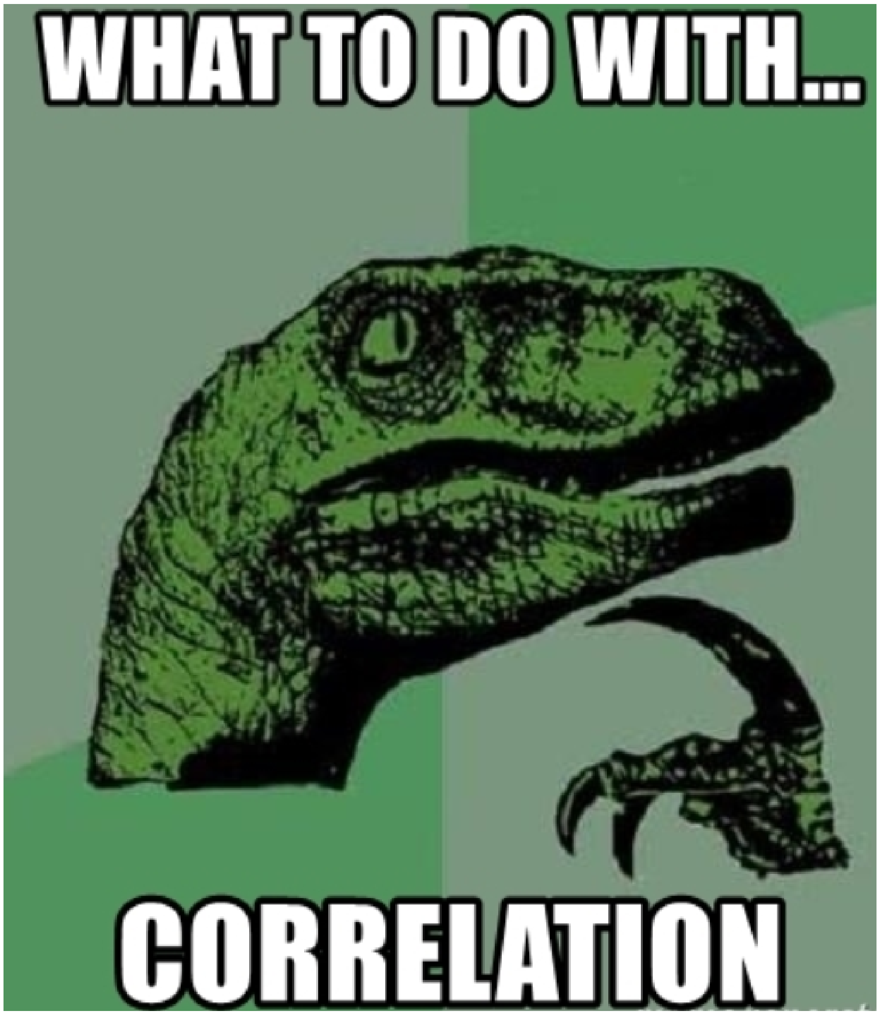
- Show by simple mathematical argument that finding the optimal decision trees for a classification problem among all the possible tree structures, can be an exponentially hard problem.(***Hint***: How many trees are there in the jungle anyway?)

- Both decision trees and deep neural networks are non-linear classifier i.e. they separates the space by complicated decision boundary. Why, then, it is so much easier for us to intuitively follow a decision tree model vs. a deep neural network?

- Back-propagation is the workhorse of deep learning. Name a few possible alternative techniques to train a neural network without using back-propagation. (***Hint***: Random search…)

- Let’s say you have two problems—a linear regression and a logistic regression (classification). Which one of them is more likely to be benefited from a newly discovered super-fast large matrix multiplication algorithm? Why? (***Hint***: Which one is more likely to use a matrix manipulation?)

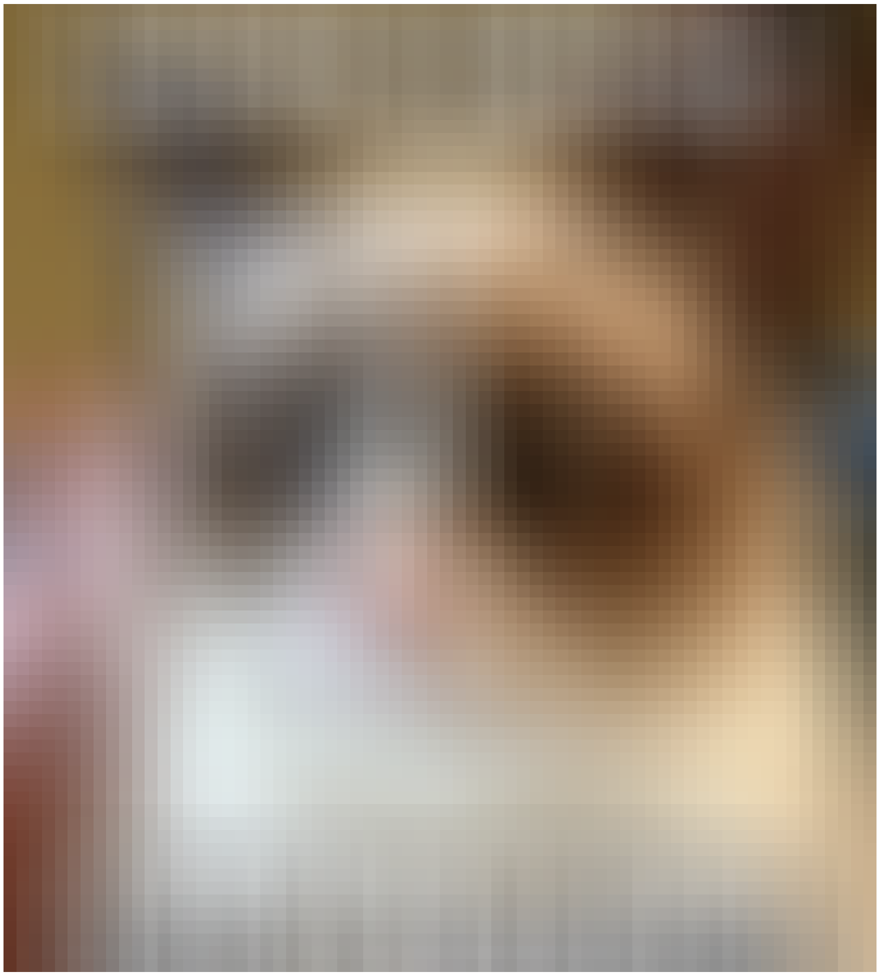
- What is the impact of correlation among predictors on *principal component analysis*? How can you tackle it?



- You are asked to build a classification model about meteorites impact with Earth (important project for human civilization). After preliminary analysis, you get 99% accuracy. Should you be happy? Why not? What can you do about it? (***Hint***: Rare event…)
- Is it possible capture the correlation between continuous and categorical variable? If yes, how?
- If you are working with gene expression data, there are often millions of predictor variables and only hundreds of sample. Give simple mathematical argument why ordinary-least-square is not a

good choice for such situation if you to build a regression model. (***Hint***: Some matrix algebra…)

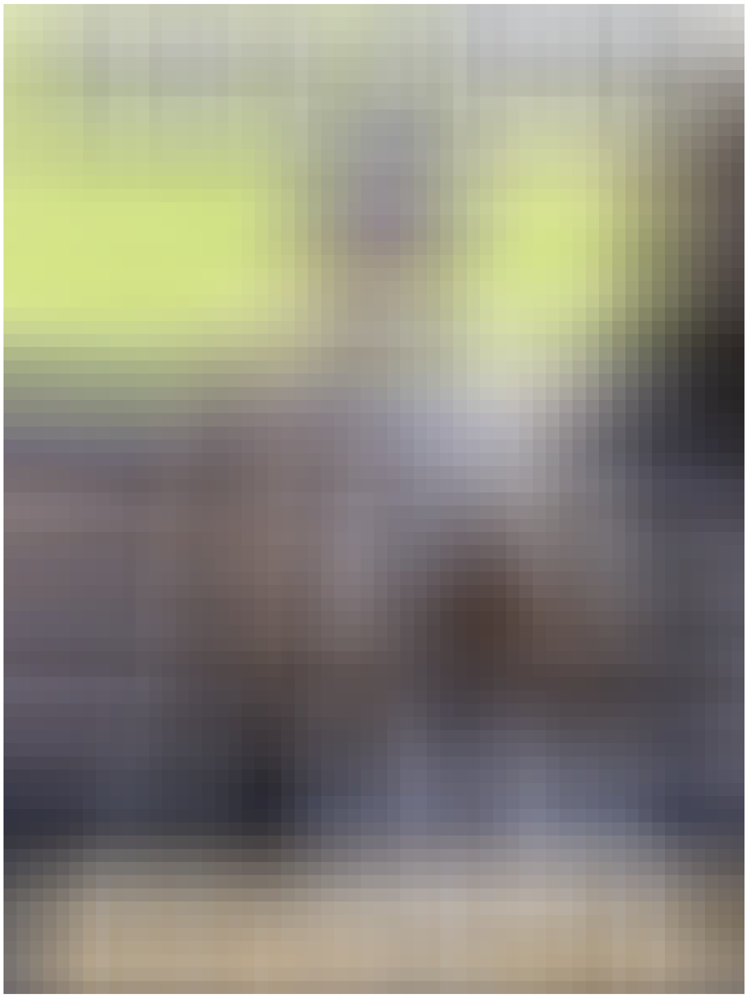
- Explain why *k*-fold cross-validation does not work well with time-series model. What can you do about it? (***Hint***: Immediate past is a close indicator of future…)
- Simple random sampling of training data set into training and validation set works well for the regression problem. But what can go wrong with this approach for a classification problem? What can be done about it? (***Hint***: Are all classes prevalent to the same degree?)



- Which is more important to you – model accuracy, or model performance?
- If you could take advantage of multiple CPU cores, would you prefer a *boosted-tree* algorithm over a *random forest*? Why? (***Hint***:

if you have 10 hands to do a task, you take advantage of it)

- Imagine your data set is known to be linearly separable and you have to guarantee the convergence and maximum number of iterations/steps of your algorithm (due to computational resource reason). Would you choose *gradient descent* in this case? What can you choose? (***Hint***: Which simple algorithm provides guarantee of finding solution?)
- Let's say you have a extremely small memory/storage. What kind of algorithm would you prefer—*logistic regression* or *k-nearest neighbor*? Why? (***Hint***: Space complexity)
- To build a machine learning model initially you had 100 data points and 5 features. To reduce bias, you doubled the features to include 5 more variables and collected 100 more data points. Explain if this is a right approach? (***Hint***: There is a curse on machine learning. Have you heard about it?)



• • •

If you have any other fun ML question or ideas to share, please contact the author here. Good questions are hard to generate and they give rise to curiosity and force one to think deeply. By asking funny and interesting question, you make the learning experience enjoyable and enriching at the same time. Hope you enjoyed this attempt of doing that.