Technical Data Science Questions

https://github.com/kojino/120-Data-Science-Interview-Questions/blob/master/statistical-inference.md

Explain confidence intervals

You are a "blind" person and your boss has asked you to find a particular "dead" fish floating in water. In Statistics, this fish is our parameter of interest. You know the fish (parameter) is in the water (population) but you don't know it's exact location. You don't want to search an entire ocean for a single dead fish, so what you do is you go to a fishing area, a place where you know it is most likely to be there (sample data). Confidence interval is your fish net that can find the dead fish. You are blind, you don't know where the fish is and since it's dead, it isn't moving (parameter is constant).
You go fishing and can just hope that your net has that dead fish. It's time for you to present your findings by giving the net to your boss.

If you used a bigger net (99% confidence interval), you will have higher confidence that your net has the fish. But your boss won't be happy to receive a 100 feet long net for a single dead fish.

Point: You get accuracy with high confidence interval but you lose precision. With a smaller 'net' you get more precision but you lose accuracy with a lower confidence interval.

Explain A/B Testing

A/B testing, or more broadly, multivariate testing, is the testing of different elements of a user's experience to determine which variation helps the business achieve its goal more effectively and efficiently (i.e. increasing conversions, etc..) So for example on a website this can be this can be, changing button colors, different user interfaces, different email subject lines, calls to action, offers, etc.

Explain Linear Regression

Linear regression is a special case of regression analysis, which tries to explain the relationship between a dependent variable and one or more explanatory variables. Mathematical functions are used to predict or estimate the value of the dependent variables. In linear regression, these functions are linear. A simple but famous example of this is $y = mx + b$. y is our dependent variable and we can predict its value by using all other parts of the equation. This is applied in machine learning and modeling on a much larger scale.

What is R2? What are some other metrics that could be better than R2 and why?

- goodness of fit measure. variance explained by the regression / total variance
- the more predictors you add the higher R^2 becomes.
    - hence use adjusted R^2 which adjusts for the degrees of freedom
    - or train error metrics

What is the curse of dimensionality?

- High dimensionality makes clustering hard, because having lots of dimensions means that everything is "far away" from each other.
- For example, to cover a fraction of the volume of the data we need to capture a very wide range for each variable as the number of variables increases
- All samples are close to the edge of the sample. And this is bad news because prediction is much more difficult near the edges of the training sample.
- We should conduct PCA to reduce dimensionality where/when we can

Is more data always better?

In a perfect world yes it absolutely is, but the world isn't perfect so there are some issues.

- Statistically,
    - It depends on the quality of your data, so if your data is biased, getting more of it won't help.
    - It also depends on your model. If your model suffers from high bias, getting more data won't improve your results past a certain point. You'd need to add more features, etc.
- Practically,
    - There's a huge tradeoff between having more data and the additional storage, computational power, memory you would need to be able to use that data. There is always a cost to having more data.

How can you make sure that you don’t analyze something that ends up meaningless?

- Proper exploratory data analysis.

In every data analysis task, there's the exploratory phase where you're just graphing things, testing things on small sets of the data, summarizing simple statistics, and getting rough ideas of what hypotheses you might want to pursue further. Then there's the next phase, where you look deeply into a set of hypotheses.

The exploratory phase will generate lots of possible hypotheses, and then moving on from there will let you really understand a few of them. Focus on proper exploration of the data and you'll prevent yourself from wasting time on many things that end up meaningless, although not everything.

How do you deal with some of your predictors being missing?

- Remove rows with missing values - This works well if 1) the values are missing randomly 2) if you don't lose too much of the dataset after doing so.
- Specific models can handle missing values much better than others such as RandomForest and other tree models

You have several variables that are positively correlated with your response, and you think combining all of the variables could give you a good prediction of your response. However, you see that in the multiple linear regression, one of the weights on the predictors is negative. What could be the issue?

- Multicollinearity refers to a situation in which two or more explanatory variables in a multiple regression model are highly linearly related.
- Multicollinearity: In statistics, multicollinearity is a phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy. An issue with multicollinearity is that small changes to the input data can lead to large changes in the model, even resulting in changes of sign of parameter estimates.
- Leave the model as is, despite multicollinearity. The presence of multicollinearity doesn't affect the efficiency of extrapolating the fitted model to new data provided that the predictor variables follow the same pattern of multicollinearity in the new data as in the data on which the regression model is based.
- principal component regression

Let's say you're given an unfeasible amount of predictors in a predictive modeling task. What are some ways to make the prediction more feasible?

- PCA

Now you have a feasible amount of predictors, but you're fairly sure that you don't need all of them. How would you perform feature selection on the dataset?

- ridge / lasso / elastic net regression
- Univariate Feature Selection where a statistical test is applied to each feature individually. You retain only the best features according to the test outcome scores
- Recursive Feature Elimination: (MY FAVORITE)
    - First, train a model with all the feature and evaluate its performance on held out data.
    - Then drop let say the 10% weakest features (e.g. the feature with least absolute coefficients in a linear model) and retrain on the remaining features.
    - Continue until you see a sharp drop in the predictive accuracy of the model or a large increase in the error of the model.

Your linear regression didn't run and communicates that there are an infinite number of best estimates for the regression coefficients. What could be wrong?

- $p > n$.
- If some of the explanatory variables are perfectly correlated (positively or negatively) then the coefficients would not be unique.

You run your regression on different subsets of your data, and find that in each subset, the beta value for a certain variable varies wildly. What could be the issue here?

- The dataset might be heterogeneous. In which case, it is recommended to cluster datasets into different subsets wisely, and then draw different models for different subsets. Or, use models like non parametric models (trees) which can deal with heterogeneity quite nicely.

15. What is the main idea behind ensemble learning? If I had many different models that predicted the same response variable, what might I want to do to incorporate all of the models? Would you expect this to perform better than an individual model or worse?

- The assumption is that a group of weak learners can be combined to form a strong learner.
- Hence the combined model is expected to perform better than an individual model.
- Assumptions:
    - average out biases
    - reduce variance
- Bagging works because some underlying learning algorithms are unstable: slightly different inputs leads to very different outputs. If you can take advantage of this instability by running multiple instances, it can be seen that the increased stability leads to lower error.
- Boosting works because of the focus on better defining the "decision edge". By re-weighting examples near the margin (the positive and negative examples) you get a reduced error
- Use the outputs of your models as inputs to a meta-model.

For example, if you're doing binary classification, you can use all the probability outputs of your individual models as inputs to a final logistic regression (or any model, really) that can combine the probability estimates.

You have 100 mathletes and 100 math problems. Each mathlete gets to choose 10 problems to solve. Given data on who got what problem correct, how would you rank the problems in terms of difficulty?

- One way you could do this is by storing a "skill level" for each user and a "difficulty level" for each problem. We assume that the probability that a user solves a problem only depends on the skill of the user and the difficulty of the problem.* Then we maximize the likelihood of the data to find the hidden skill and difficulty levels.
- The Rasch model for dichotomous data takes the form:
  {\displaystyle \Pr\{X_{ni}=1\}={\frac {\exp({\beta _{n}}-{\delta _{i}})}{1+\exp({\beta _{n}}-{\delta _{i}})}},}
  where is the ability of person and is the difficulty of item}.

You have 5000 people that rank 10 sushis in terms of saltiness. How would you aggregate this data to estimate the true saltiness rank in each sushi?

- Some people would take the mean rank of each sushi. If I wanted something simple, I would use the median, since ranks are (strictly speaking) ordinal and not interval, so adding them is a bit risque (but people do it all the time and you probably won't be far wrong).

Given data on congressional bills and which congressional representatives co-sponsored the bills, how would you determine which other representatives are most similar to yours in voting behavior? How would you evaluate who is the most liberal? Most republican? Most bipartisan?

- collaborative filtering. you have your votes and we can calculate the similarity for each representatives and select the most similar representative
- for liberal and republican parties, find the mean vector and find the representative closest to the center point

How would you come up with an algorithm to detect plagiarism in online content?

- reduce the text to a more compact form (e.g. fingerprinting, bag of words) then compare those with other texts by calculating the similarity

You have data on all purchases of customers at a grocery store. Describe to me how you would program an algorithm that would cluster the customers into groups. How would you determine the appropriate number of clusters to include?

- KNN
- choose a small value of k that still has a low SSE (elbow method)
- https://bl.ocks.org/rpgove/0060ff3b656618e9136b

Let's say you're building the recommended music engine at Spotify to recommend people music based on past listening history. How would you approach this problem?

- collaborative filtering

What could be some issues if the distribution of the test data is significantly different than the distribution of the training data?

- So the model could have very high training accuracy but poor testing accuracy and you may jump to the conclusion that the model is just overfitting when in reality you sampled the data poorly. Cross validation can be something that can help here.
- When there is a change in data distribution, this is called the dataset shift. If the train and test data has a different distribution, then the classifier would likely overfit to the train data.
- What can cause this:
    - Training samples are obtained in a biased way. (sample selection bias)
    - Train is different from test because of temporal, spatial changes. (non-stationary environments)
- Solution to covariate shift
    - importance weighted cv

What are some ways I can make my model more robust to outliers?

- Changes to the algorithm:
    - Use tree-based models instead of regression as they are more resistant to outliers. For statistical tests, use non parametric tests (chi-square) instead of parametric ones.
    - Use robust error metrics such as MAE or Huber Loss instead of MSE.
- Changes to the data:
    - Winsorizing the data (limit extreme values to reduce effect of outliers)
    - Transforming the data (e.g. log)
    - Remove them only if you're certain they're anomalies not worth predicting

What are some differences you would expect in a model that minimizes squared error, versus a model that minimizes absolute error? In which cases would each error metric be appropriate?

- Mean Squared Error (MSE) is more strict to having outliers. Mean Absolute Error (MAE) is more robust in that sense, but is harder to fit the model for because it cannot be numerically optimized. So when there is less variability in the model and the model is computationally easy to fit, we should use MAE, and if that's not the case, we should use MSE.
- MSE: easier to compute the gradient, MAE: linear programming needed to compute the gradient
- MAE more robust to outliers. If the consequences of large errors are great, use MSE
- MSE corresponds to maximizing likelihood of Gaussian random variables

What error metric would you use to evaluate how good a binary classifier is? What if the classes are imbalanced? What if there are more than 2 groups?

- Accuracy: proportion of instances you predict correctly. **Pros**: intuitive, easy to explain, **Cons**: works poorly when the class labels are imbalanced
- AUROC: plot fpr on the x axis and tpr on the y axis for different threshold. Given a random positive instance and a random negative instance, the AUC is the probability that you can identify who's who. **Pros**: Works well when testing the ability of distinguishing the two classes, **Cons**: can't interpret predictions as probabilities (because AUC is determined by rankings), so can't explain the uncertainty of the model
- logloss/deviance: **Pros**: error metric based on probabilities, **Cons**: very sensitive to false positives, negatives
- When there are more than 2 groups, we can have k binary classifications and add them up for logloss. Some metrics like AUC is only applicable in the binary case.

What are various ways to predict a binary response variable? Can you compare two of them and tell me when one would be more appropriate? What's the difference between these? (SVM, Logistic Regression, Naive Bayes, Decision Tree, etc.)

- Things to look at: N, P, linearly separable? features independent? likely to overfit? speed, performance, memory usage
- Logistic Regression
    - features roughly linear, problem roughly linearly separable
    - robust to noise, use l1,l2 regularization for model selection, avoid overfitting
    - the output come as probabilities
    - efficient and the computation can be distributed
    - can be used as a baseline for other algorithms
    - (-) can hardly handle categorical features
- SVM
    - with a nonlinear kernel, can deal with problems that are not linearly separable
    - (-) slow to train, for most industry scale applications, not really efficient
- Naive Bayes
    - computationally efficient when P is large by alleviating the curse of dimensionality
    - works surprisingly well for some cases even if the condition doesn't hold
    - with word frequencies as features, the independence assumption can be seen reasonable. So the algorithm can be used in text categorization
    - (-) conditional independence of every other feature should be met
- Tree Ensembles
    - good for large N and large P, can deal with categorical features very well

    - non parametric, so no need to worry about outliers
    - GBT's work better but the parameters are harder to tune
    - RF works out of the box, but usually performs worse than GBT
- Deep Learning
    - works well for some classification tasks (e.g. image)
    - used to squeeze something out of the problem
    - Wants a lot of data, with smaller datasets you can get very similar results with much simpler algorithms. (overkill)

What is regularization and where might it be helpful? What is an example of using regularization in a model?

- Regularization is useful for reducing variance in the model, meaning avoiding overfitting . For example, we can use L1 regularization in Lasso regression to penalize large coefficients.

Why might it be preferable to include fewer predictors over many?

- When we add irrelevant features, it increases model's tendency to overfit because those features introduce more noise. When two variables are correlated, they might be harder to interpret in case of regression, etc.
- curse of dimensionality
- adding random noise makes the model more complicated but useless
- computational cost
- Ask someone for more details.

Given training data on tweets and their retweets, how would you predict the number of retweets of a given tweet after 7 days after only observing 2 days worth of data?

- Build a time series model with the training data with a seven day cycle and then use that for a new data with only 2 days data.
- Build a regression function to estimate the number of retweets as a function of time t
- to determine if one regression function can be built, see if there are clusters in terms of the trends in the number of retweets
- if not, we have to add features to the regression function
- features + # of retweets on the first and the second day -> predict the seventh day
- https://en.wikipedia.org/wiki/Dynamic_time_warping

How could you collect and analyze data to use social media to predict the weather?

- We can collect social media data using twitter, Facebook, instagram API's. Then, for example, for twitter, we can construct features from each tweet, e.g. the tweeted date, number of favorites, retweets, and of course, the features created from the tweeted content itself. Then use a multi variate time series model to predict the weather.

How would you construct a feed to show relevant content for a site that involves user interactions with items?

- We can do so using building a recommendation engine. The easiest we can do is to show contents that are popular other users, which is still a valid strategy if for example the contents are news articles. To be more accurate, we can build a content based filtering or collaborative filtering. If there's enough user usage data, we can try collaborative filtering and recommend contents other similar users have consumed. If there isn't, we can recommend similar items based on vectorization of items (content based filtering).
- Can also use collaborative filtering here much like music recommendation engine

How would you design the people you may know feature on LinkedIn or Facebook?

- Find strong unconnected people in weighted connection graph
    - Define similarity as how strong the two people are connected
    - Given a certain feature, we can calculate the similarity based on
        - friend connections (neighbors)
        - Check-in's people being at the same location all the time.
        - same college, workplace
        - Have randomly dropped graphs test the performance of the algorithm
- ref. News Feed Optimization
    - Affinity score: how close the content creator and the users are
    - Weight: weight for the edge type (comment, like, tag, etc.). Emphasis on features the company wants to promote
    - Time decay: the older the less important

How would you predict who someone may want to send a Snapchat or Gmail to?

- for each user, assign a score of how likely someone would send an email to
- the rest is feature engineering:
    - number of past emails, how many responses, the last time they exchanged an email, whether the last email ends with a question mark, features about the other users, etc.
- People who someone sent emails the most in the past, conditioning on time decay.

How would you suggest to a franchise where to open a new store?

- build a master dataset with local demographic information available for each location.
    - local income levels, proximity to traffic, weather, population density, proximity to other businesses
    - a reference dataset on local, regional, and national macroeconomic conditions (e.g. unemployment, inflation, prime interest rate, etc.)
    - any data on the local franchise owner-operators, to the degree the manager
- identify a set of KPIs acceptable to the management that had requested the analysis concerning the most desirable factors surrounding a franchise
    - quarterly operating profit, ROI, EVA, pay-down rate, etc.
- run econometric models to understand the relative significance of each variable
- run machine learning algorithms to predict the performance of each location candidate

In a search engine, given partial data on what the user has typed, how would you predict the user's eventual search query?

- Based on the past frequencies of words shown up given a sequence of words, we can construct conditional probabilities of the set of next sequences of words that can show up (n-gram). The sequences with highest conditional probabilities can show up as top candidates.
- To further improve this algorithm,
    - we can put more weight on past sequences which showed up more recently and near your location to account for trends
    - show your recent searches given partial data

Given a database of all previous alumni donations to your university, how would you predict which recent alumni are most likely to donate?

- Based on frequency and amount of donations, graduation year, major, etc, construct a supervised regression (or binary classification) algorithm.

You're Uber and you want to design a heat-map to recommend to drivers where to wait for a passenger. How would you approach this?

- Based on the past pickup location of passengers around the same time of the day, day of the week (month, year), construct
- Ask someone for more details.
- Based on the number of past pickups
    - account for periodicity (seasonal, monthly, weekly, daily, hourly)
    - special events (concerts, festivals, etc.) from tweets

How would you build a model to predict a March Madness bracket?

- One vector each for team A and B. Take the difference of the two vectors and use that as an input to predict the probability that team A would win by training the model. Train the models using past tournament data and make a prediction for the new tournament by running the trained model for each round of the tournament
- Some extensions:
    - Experiment with different ways of consolidating the 2 team vectors into one (e.g concatenating, averaging, etc)
    - Consider using a RNN type model that looks at time series data.

You want to run a regression to predict the probability of a flight delay, but there are flights with delays of up to 12 hours that are really messing up your model. How can you address this?

- This is equivalent to making the model more robust to outliers.
- See Q3.

What is a P-Value ?

- The probability to obtain a similar or more extreme result than observed when the null hypothesis is assumed.
- If the p-value is small, the null hypothesis is unlikely

In an A/B test, how can you check if assignment to the various buckets was truly random?

- Plot the distributions of multiple features for both A and B and make sure that they have the same shape. More rigorously, we can conduct a permutation test to see if the distributions are the same.
- MANOVA to compare different means

What might be the benefits of running an A/A test, where you have two buckets who are exposed to the exact same product?

- Verify the sampling algorithm is random.

What would be the hazards of letting users sneak a peek at the other bucket in an A/B test?

- The user might not act the same suppose had they not seen the other bucket. You are essentially adding additional variables of whether the user peeked the other bucket, which are not random across groups.

What would be some issues if blogs decide to cover one of your experimental groups?

- Same as the previous question. The above problem can happen in larger scale.

How would you conduct an A/B test on an opt-in feature?

- Ask someone for more details.

How would you run an A/B test for many variants, say 20 or more?

- one control, 20 treatment, if the sample size for each group is big enough.
- Ways to attempt to correct for this include changing your confidence level (e.g. Bonferroni Correction) or doing family-wide tests before you dive in to the individual metrics (e.g. Fisher's Protected LSD).

How would you run an A/B test if the observations are extremely right-skewed?

- lower the variability by modifying the KPI

- cap values
- percentile metrics
- log transform
- https://www.quora.com/How-would-you-run-an-A-B-test-if-the-observations-are-extremely-right-skewed

I have two different experiments that both change the sign-up button to my website. I want to test them at the same time. What kinds of things should I keep in mind?

- exclusive -> ok

What is a p-value? What is the difference between type-1 and type-2 error?

-
- type-1 error: rejecting Ho when Ho is true

- type-2 error: not rejecting Ho when Ha is true

You are AirBnB and you want to test the hypothesis that a greater number of photographs increases the chances that a buyer selects the listing. How would you test this hypothesis?

- For randomly selected listings with more than 1 pictures, hide 1 random picture for group A, and show all for group B. Compare the booking rate for the two groups.
- Ask someone for more details.

How would you design an experiment to determine the impact of latency on user engagement?

- The best way I know to quantify the impact of performance is to isolate just that factor using a slowdown experiment, i.e., add a delay in an A/B test.

What is maximum likelihood estimation? Could there be any case where it doesn't exist?

- A method for parameter optimization (fitting a model). We choose parameters so as to maximize the likelihood function (how likely the outcome would happen given the current data and our model).
- maximum likelihood estimation (MLE) is a method of estimating the parameters of a statistical model given observations, by finding the parameter values that maximize the likelihood of making the observations given the parameters. MLE can be seen as a special case of the maximum a posteriori estimation (MAP) that assumes a uniform prior distribution of the parameters, or as a variant of the MAP that ignores the prior and which therefore is unregularized.
- for gaussian mixtures, non parametric models, it doesn't exist

What's the difference between a MAP, MOM, MLE estimator? In which cases would you want to use each?

- MAP estimates the posterior distribution given the prior distribution and data which maximizes the likelihood function. MLE is a special case of MAP where the prior is uninformative uniform distribution.
- MOM sets moment values and solves for the parameters. MOM is not used much anymore because maximum likelihood estimators have higher probability of being close to the quantities to be estimated and are more often unbiased.

What is a confidence interval and how do you interpret it?

- For example, 95% confidence interval is an interval that when constructed for a set of samples each sampled in the same way, the constructed intervals include the true mean 95% of the time.
- if confidence intervals are constructed using a given confidence level in an infinite number of independent experiments, the proportion of those intervals that contain the true value of the parameter will match the confidence level.

What is unbiasedness as a property of an estimator? Is this always a desirable property when performing inference? What about in data analysis or predictive modeling?

- Unbiasedness means that the expectation of the estimator is equal to the population value we are estimating. This is desirable in inference because the goal is to explain the dataset as

accurately as possible. However, this is not always desirable for data analysis or predictive modeling as there is the bias variance tradeoff. We sometimes want to prioritize the generalizability and avoid overfitting by reducing variance and thus increasing bias.

Bias VS Variance

Scale Example: Your true weight is 150 lbs, you weigh yourself 10 times: Bias is the amount that average value differs from the true weight, whereas Variance is the amount of spread found in the range of those values.

Overfitting

How to stop overfitting?

Cross-validation: is a powerful preventative measure against overfitting. The idea is clever: Use your initial training data to generate multiple mini train-test splits. Use these splits to tune your model.

Train with more data: It won't work every time, but training with more data can help algorithms detect the signal better.

Remove features: Some algorithms have built-in feature selection. For those that don't, you can manually improve their generalizability by removing irrelevant input features.

Early stopping: When you're training a learning algorithm iteratively, you can measure how well each iteration of the model performs. Up until a certain number of iterations, new iterations improve the model. After that point, however, the model's ability to generalize can weaken as it begins to overfit the training data. Early stopping refers stopping the training process before the learner passes that point.

Use a different model: