基础知识

- Naïve Bayes: naïve assumption is assuming condition independence among features, meaning the presence of a feature is independent of all other features. This enables: P(C|x) = p(c) * p(x1|c) * p(x2|c) * ,,,
  - Pros: easy to implement even with large data set. Still works in multi-class prediction. Performed better with categorical input features than numeric features (which usually assumes Gaussian distribution). Better than logistic regression if the independence assumption justifies
  - Cons: assumption is usually not the case. Zero frequency – meaning if some categories are not observed in the train data. All training results will be zero. Sometimes it would need pseudocount (small corrections, called Laplace smoothing)

- Confusion matrix: evaluation metrics for classification
  https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

	ACTUAL_POSITIVE (1)	**ACTUAL_NEGATIVE (0)**
PREDICTED_POSITIVE(1)	TP	FP (Type I error)
PREDICTED_NEGATIVE(0)	FN (Type ii error)	TN

  - Positive/negative 指 predicted 是 positive 还是 negative
  - Recall or TPR (True-positive-rate) or Sensitivity= TP / (TP+FN) = TP/Total Real Positives (lhs): out of all real positive cases, how much we can predict correctly
  - Precision = TP / (TP + FP) (1st row): out of all cases that predicted positive, how much we can predict correctly
  - FPR (False-positive-rate) = FP / (FP+TN) =FP/Total real negatives(rhs!)
  - Specificity (和 sensitivity 反向关系) = TN / (TN+FP) = 1 – Precision
  - Recall and Precision usually trade off b/w each other
  - For a PR curve, a good classifier aims for the upper right corner of the chart but upper left for the ROC curve
  - F-1 score: = 2 * Recall * Precision / (Recall + Precision), as some model has low recall and high precision or vice versa
  - Accuracy: = (TN+TP) / All
  - Power: Power is the probability of making a correct decision (to reject the null hypothesis) when the null hypothesis is false, which is 1-beta
    What affects Power:

Significance level alpha. Alpha 大-beta 小-power 大。 因为 higher alpha means larger probability of rejecting null hypothesis.

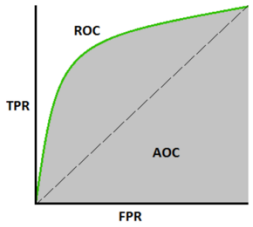
Sample size n: n 大-beta 小-power 大. B/c higher sample size narrows the distribution of test statisitc, which makes the hypothesized distribution of the test statistic and the true distribution of the test statistic more distinct from one another.

Magnitude of effect, = difference between hypothesis value of a parameter and its true vlaue. Magnitude of effect 大-越容易拒绝 null-power 大。This is because it’ s easier to detect a larger effect. (when the effect is large, the true distribution of the test statistic is far from its hypothesized distribution, so the two distributions are more distinct.)

The inherent variability in the measured response variable. As the variability increases, the power of the test of significance decreases

- 如何决定 false positive or false negative 重要。其实这个时候还是要结合产品来说。比如某信用卡 fraud detection 产品，那么减小 false posititive 就比较重要，因为如果你的 false positive rate 太高了，很多不应该被 decline 的人都被 decline 了，大家就不会用你的产品了。反之如果是癌症 detection 模型，那么 false negtive 就比较重要了，因为本来人家得了绝症，但是你没有发现，结果延误了治疗。

- Auc-roc curve (receiver operating characteristics): TPR against FPR. ROC is probability curve, AUC shows how much the model can distinguish between classes

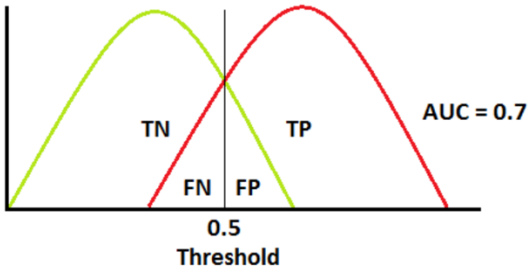


Why AUC is not sensitive to class distribution change (class imbalance): 因为如果 total number of positive (or negative) cases increase 2x in one class, both TP and FN will increase 2x and TPR will not change. Same to FPR.

理解 AUC vs real examples
https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

理解 auc = 1 是 ideal condition, no overlap b/w TN and TP, distinguish perfect

auc = 70%: 70% chance the model can distinguish b/w positive, negative. 此时 TP and TN 有 overlap, wrong classification 分成 FN,FP. 随着 threshold increase（右移，即 alpha 减小），there will be more negative cases so TPR 减小,specificity (TN)增加从而 FPR 减小；随着 threshold decrease, there will be more positive cases so TPR increase, specificity (TN) decrease 从而 FPR 增加 (ROC curve 就是这么来的)



(和 alpha / beta 图一样)

auc=0.5: TN TP 完全重合，model has no distinguishing capacity

auc = -1:预测完全反了

multi-class AUC-ROC: use One vs All: 几个 class 就有几个 curve, 每个都是 classify 一个 class vs all rest classes

- MLE: maximum likelihood estimation (MLE) is a method of estimating the parameters of a statistical model, given observations. The method obtains the parameter estimates by finding the parameter values that maximize the likelihood function.
- Likelihood function: joint probability distribution of observed data expressed as a function of statistical parameters.
- Central limit theorem: given a set of independent and identically distributed (i.i.d.) random variables drawn from a distribution of expected value given by μ and finite variance given by σ2, the sampling distribution of the means of those samples will become approximately normally distributed with population mean μ and standard deviation σ/√ N as the sample size (N) becomes larger, regardless of the population distribution.

- Random forest

Algorithm 15.1 *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

---

the bias of bagged trees is the same as that of the individual (bootstrap) trees, and the only hope of improvement is through variance reduction. This is in contrast to boosting, where the trees are grown in an adaptive way to remove bias,

The idea in random forests is to improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the variance too much. This is achieved in the tree-growing process through random selection of the input variables

Variable importance:

- Calculate decrease of Gini impurity score (or increase of information gains?) for all of the nodes that were split on the specific variable
- Another method is to use OOB (out-of-bag) sample in Random forest to construct a different variable- importance measure, apparently to measure the prediction strength of each variable. When the bth tree is grown, the oob samples are passed down the tree and obtain the prediction accuracy. Then the values for the jth variable are randomly permuted in the oob samples, and the accuracy is again computed. The decrease in accuracy as a result of

this permuting is averaged over all trees, and is used as a measure of the importance of variable j in the random forest.

Out-of-bag: For each observation zi = (xi,yi), construct its random forest predictor by averaging only those trees corresponding to bootstrap samples in which zi did not appear. An oob error estimate is almost identical to that obtained by N-fold cross- validation;

Hyperparameters: number of trees, max depth of a tree, minimum data points in the node before spliting, minimum data points in leaves

- Boosting (XGBoost):
    - trains a large number of "weak" learners in sequence.
    - Key regularization parameter: 除了 l1, l2，还有 Shrinkage - The simplest implementation of shrinkage in the context of boosting is to scale the contribution of each tree by a factor $0 < \nu < 1$ when it is added to the current approximation. ν can be regarded as controlling the learning rate of the boosting procedure. Smaller values of ν lead to larger values of M (number of iteration) for the same training risk, so that there is a tradeoff between them. Thus, both ν and M control prediction risk on the training data.
    - Subsample parameter: variance-reduction. Defines fraction of data sampled in each iteration.
    - 其他 hyperparamter： size of tree, minimum node size…
    - Adaboost vs Gradient Boosting

      -Adaboost either requires the users to specify a set of weak learners or randomly generates the weak learners before the actual learning process. The weight of each learner is adjusted at every step depending on whether it predicts a sample correctly.

      -On the other hand, Gradient Boosting builds the first learner on the training dataset to predict the samples, calculates the loss function (Difference between real value and output of the first learner). And use this loss to build an improved learner in the second stage.

      At every step, the derivative of the loss function w.r.t predictive function obtained in last step is calculated using the Gradient Descent Method and the new pseudo-residual becomes a objective function for the subsequent iteration.

- SVM: is a classifier defined by a separating hyperplane. Given labelled training data, SVM can output an optimal hyperplane to categorize new examples.
    - In two dimensional space, hyperplane is a line
    - -When a line cannot separate two classes in two dimensions, kernals are introduced to apply transformation of data into high dimensional and then categorize (Polynomial and exponential kernels (exp{-gamma * ||x1-x2||^2) calculates separation line in higher dimension.
    - -kernel: computes inner products in the transformed space
      Mapping an input from X to a vector in d-dimensional is called feature extraction or featurization
    - large feature space, when p >> N
      With p ≫ N the models are already sufficiently complex and overfitting is always a danger. Yet despite the high dimensionality, radial kernels can help in these high dimensional problems. The radial kernel tends to dampen inner products between points far away from each other (when gamma is large), which in turn leads to robustness to outliers.

      Very large feature spaces have two potential issues: Overfitting and Memory and computational costs. Overfitting we handle with regularization. Kernel methods can (sometimes) help with memory and computational costs.
      The **kernel matrix** for a kernel k on $x_1, \ldots, x_n \in \mathcal{X}$ is
      $$K = \big(k(x_i, x_j)\big)_{i,j} = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \dots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{pmatrix} \in \mathbf{R}^{n \times n}.$$
      New kernel may correspond to a very high-dimensional feature space. Once the kernel matrix is computed, the computational cost depends on number of data points, rather than the dimension of feature space.
    - parameter w and function
      w = C*sum(alpha * y_i * x_i)
      optimal function f(x) = C * sum(alpha * yi * K(x,xi))
      • Observations correctly classified and outside their margins. They have yi *f (xi) > 1, and Lagrange multipliers αi = 0.
      • Observations sitting on their margins with yi*f(xi) = 1, with Lagrange multipliers αi ∈ [0, 1].

• Observations inside their margins have yi*f (xi) < 1, with αi = 1.

- SVM regularization:

  -cost parameter or regularization parameter C in objective function:

  min ||w||^2 + C/n * max(1-y(wx+b),0)

  when C is large, max(1-m,0) needs to be very small and therefore result in a smaller-margin and train strictly to all data points (possible to overfit); when C is small, it is '~~heavier regularization'~~ that tends to get a large-margin hyperplane and misclassification is allowed (loose regularization)

  -gamma in exponential kernals:

  when gamma is large, for far-away data points, their Gaussian kernals (distance) are close to zero. Therefore, only nearby data points will be considered in generating hyperplane. Too high gamma will loss non-linearity in high-dimension

  when gamma is small, all data points would be more easily to consider. Too low gamma will result in overfit.

  -margin is distance from hyperplane to closest data points. Good margin is not too close to either classes, or as far as possible for both side.

- When random forest is better than SVM?
  - Multi/binary class: Random Forest is intrinsically suited for multiclass problems, while SVM is intrinsically two-class. For multiclass problem you will need to reduce it into multiple binary classification problems.
  - Variable type: Random Forest works well with a mixture of numerical and categorical features. When features are on the various scales, it is also fine. Roughly speaking, with Random Forest you can use data as they are. SVM maximizes the "margin" and thus relies on the concept of "distance" between different points. As a consequence, one-hot encoding for categorical features is a must-do. Further, min-max or other scaling is highly recommended at preprocessing step.
  - Sample size: If you have data with n points and m features, an intermediate step in SVM is constructing an $n \times n$ matrix (think about memory requirements for storage) by calculating n^2 dot products (computational complexity). Therefore, SVM is hardly scalable beyond 10^5 points. Large number of features (homogeneous features with meaningful distance, pixel of image would be a perfect example) is generally not a problem.
  - Result interpretation: For a classification problem Random Forest gives you probability of belonging to class. SVM gives you distance to the boundary, you still need to convert it to probability somehow if you need probability.

  For those problems, where SVM applies, it generally performs better than Random Forest.

SVM gives you "support vectors", that is points in each class closest to the boundary between classes. They may be of interest by themselves for interpretation.

- Regression when p >> N: 其实这就是一个 feature selection process

  PCA can identify linear combination of features with largest variance, but it might not be able to select those with high correlation with outcome variable. An alternate method is to 1) ran univaraite regression on each feature and select most k correlated features; 2) then run PCA on subset of data consisting of selected features. 3) and run regression with top few principal components

Algorithm 18.1 *Supervised Principal Components.*

1. Compute the standardized univariate regression coefficients for the outcome as a function of each feature separately.
2. For each value of the threshold θ from the list $0 \leq \theta_1 < \theta_2 < \cdots < \theta_K$:
   (a) Form a reduced data matrix consisting of only those features whose univariate coefficient exceeds θ in absolute value, and compute the first m principal components of this matrix.
   (b) Use these principal components in a regression model to predict the outcome.
3. Pick θ (and m) by cross-validation.

- Elastic net

  there are often strong correlations among the variables. The lasso penalty is somewhat indifferent to the choice among a set of strong but correlated variables (sparse model). The ridge penalty, on the other hand, tends to shrink the coefficients of correlated variables toward each other. The elastic net penalty is a linear combination of l1 and l2 regularization, which can encourages highly correlated features to be averaged, and also encourages a sparse solution in the coefficients of these averaged features.
  'lasso regression could result in multiple optimal solutions when highly correlated variables exist.'

- Person correlation and cosine similarity in recommender system

  Pearson correlation coefficient defined as the covariance between two vectors divided by their standard deviations.
  Cosine similarity defined as vector based similarity measure, which is the angular similarity between two vectors. angle 0 defines match and 90 otherwise
  Adjusted cosine or cosine over mean centred vectors is similar to pearson similarity except if two are calculated over different set of rated vectors: pearson correlation works over co-rated vector where as adjusted cosine consider all the rated vector

- The Gauss-Markov theorem implies that the least squares estimator has the smallest mean squared error of all linear estimators with no bias.
- Linear assumption

Linearity: The relationship between X and the mean of Y is linear.
Homoscedasticity: The variance of residual is the same for any value of X.
Independence: Observations are independent of each other.
Normality: For any fixed value of X, Y is normally distributed.
No autocorrelation in residuals

- heteroscedasticity

What is heteroscedasticity and how do you prove?

Generally assume conditional variance var(u|X)=sigma2, it is homoscedasticity. For heteroscedasticity, the variance vary for different samples.

(1) Regression u^2 on all X, do F-test. If reject null, then hetero (cross sectional regression)
(2) Compare standard error and robust hetero standard error for coefficients.

$$\frac{\sum\left(x_i-\overline{x}\right)^2\hat{u}_i^2}{SST_x^2},$$

$$\widehat{Var}\left(\hat{\beta}_j\right)=\frac{\sum\hat{r}_{ij}^2\hat{u}_i^2}{SSR_j^2}$$

where

- $\hat{r}_{ij}$ is the i^{th} residual from regressing x_j on all other independent variables
- SSR_j is the sum of squared residuals from this regression

- multi collinearity

How to check multi-collinearity?

(too more variables are included in the model)

(1) The t-tests for each of the individual slopes are non-significant (P > 0.05), but the overall F-test for testing all of the slopes are simultaneously 0 is significant (P < 0.05).
(2) VIF **Variance inflation factor**
Multicollinearity can make standard errors of coefficients to be inflated. So VIF measures how much the variance of the estimated regression coefficient beta_k is "inflated" by the existence of correlation among the predictor.
If VIFs exceeding 4 warrant further investigation, while VIFs exceeding 10 are signs of serious multicollinearity requiring correction.

$$VIF_k=\frac{Var(beta_k)\text{多元回归}}{Var(beta_k)\text{一元回归}}=\frac{1}{1-R_k^2}$$

where R_k^2 is the R^2-value obtained by regressing the k^{th} predictor on the remaining predictors. Note that a variance inflation factor exists for *each of the k predictors* in a multiple regression model.

R: fit<-lm(Y~X,data=..); sqrt(vif(fit))>2

use the **Durbin–Watson** to detect the presence of autocorrelation in the residuals from a regression analysis [R: durbinWatsonTest(lm(y~x))]. Numerical analysis can also includes the lack-of-fit test, for assessing the correctness of the functional part of the model can aid in interpreting a borderline residual plot. It is F-test and the null hypothesis that says that a proposed model fits well.

-

How to exam a distribution is a normal distribution?

Graphical: quantile-quantile plot(QQ), For normal data the points plotted in the QQ plot should fall approximately on a straight line, indicating high positive correlation between sample data and normal quantiles. Quantile function is called Probit function. **For Q-Q plot, if a set of data is actually a sample of a normal distribution, a plot of the values against their probit scores will be approximately linear.**

Shapiro-will test R: Sha'piro.test(x), p<alpha then population is not normally distributed

* back-of-the-envelope test takes the sample maximum and minimum and computes their z-score, or more properly t-statistic, and compares it to the 68–95–99.7 rule: if one has a 3σ event and substantially fewer than 300 samples, or a 4s event and substantially fewer than 15,000 samples, then a normal distribution is not enough to explain the maximum magnitude of deviations in the sample data.

-

When do you see a negative R squared

R2 compares the fit of the chosen model with that of a horizontal straight line (the null hypothesis). **If the chosen model fits worse than a horizontal line, which is the sample mean to**

-

track the dependent variable, then R2 is negative. A negative R2 is possible with linear regression **when either the intercept or the slope are constrained, or no intercept is included in the model,** so that the regression line fits worse than a horizontal line.

> If the regressors do not include a constant but (as some regression software packages do) you nevertheless calculate R^2 by the formula
>
> $$R^2 = 1 - \frac{\sum_{i=1}^{n} e_i^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$
>
> then the R^2 can be negative. This is because, without the benefit of an intercept, the regression could do worse than the sample mean in terms of tracking the dependent variable (i.e., the numerator could be greater than the denominator).

- model assessment

If we are in a data-rich situation, the best approach for both problems is to randomly divide the dataset into three parts: a training set, a validation set, and a test set (50/25/25). The training set is used to fit and tune the models (via cross-validation); the *validation set is used to estimate prediction error for model selection*; the test set is used for assessment of the generalization error of the final chosen model.

- K-fold cross validation

  For the kth part, we fit the model to the other K − 1 parts of the data, and calculate the prediction error of the fitted model when predicting the kth part of the data. We do this for k = 1,2,...,K and combine the K estimates of prediction error

  P >> N:
  Consider a classification problem with a large number of predictors, as may arise, for example, in genomic or proteomic applications. A typical strategy for analysis might be as follows:
  1. Screen the predictors: find a subset of "good" predictors that show fairly strong (univariate) correlation with the class labels
  2. Using just this subset of predictors, build a multivariate classifier.
  3. Use cross-validation to estimate the unknown tuning parameters and to estimate the prediction error of the final model.

  When p >> N:
  Consider a scenario with N = 50 samples in two equal-sized classes, and p = 5000 quantitative predictors (standard Gaussian) that are independent of the class labels. The true (test) error rate of any classifier is 50%.
  Here is the correct way to carry out cross-validation in this example:
  1. Divide the samples into K cross-validation folds (groups) at random.
  2. For each fold k = 1,2,...,K
     (a)Find a subset of "good" predictors (100) that show fairly strong (univariate) correlation with the class labels, using all of the samples except those in fold k.
     (b) Using just this subset of predictors, build a multivariate classifier, using all of the samples except those in fold k.
     (c) Use the classifier to predict the class labels for the samples in fold k.

- Bootstrap

  Pr{observation i ∈ at least one bootstrap sample b}is 0.632

- Neural network backpropagation

  Is the generic approach to minimizing R(θ) loss function by gradient descent,
  In the forward pass, the current weights are fixed and the predicted values $\hat{f}(x)$ are computed. In the backward pass, the errors δ in output layers are computed, and then back-propagated to give the errors in hidden layer. Both sets of errors are then used to compute the gradients for the updates

  Initials: Usually starting values for weights are chosen to be random values near zero. Like a Gaussian random vectors. Hence the model starts out nearly linear, and becomes nonlinear as the weights increase.

  Number of hidden layers: Generally speaking it is better to have too many hidden units than too few. With too few hidden units, the model might not have enough flexibility to capture the nonlinearities in the data; with too many hidden units, the

extra weights can be shrunk toward zero if appropriate regularization is used. Typically the number of hidden units is somewhere in the range of 5 to 100, with the number increasing with the number of inputs and number of training cases. It is most common to put down a reasonably large number of units and train them with regularization. cross-validation can be used to estimate the optimal number and the regularization parameter. Choice of the number of hidden layers is guided by background knowledge and experimentation. Each layer extracts features of the input for regression or classification. Use of multiple hidden layers reduces the need for feature engineering and allows construction of hierarchical features at different levels of resolution.

- Binomial test

Statistical significance (alpha) is the measure of the strength of the evidence based on the presence of the effect.
Generally, the statistical tests have to follow the following steps:
i) Collection of data.
ii) Set a null hypothesis and alternative hypothesis.
iii) The second step is to specify the α level which is also known as the significance level.
iii) Determine a number which is known as test statistic. It finds the degree of deviation of the observed data from expectation in the null hypothesis.
iv) The third step is to compute the probability value (also known as the p value). This is the probability of obtaining a sample statistic as different or more different from the observations given that the null hypothesis is true.
v) On the basis of this P value, null hypothesis is either accepted or rejected. When a probability value is below the α level, the effect is statistically significant and the null hypothesis is rejected.
(Failure to reject the null hypothesis does not constitute support for the null hypothesis. It just means you do not have sufficiently strong data to reject it.)

one sample t-test: ~t(n-1)

$$t = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} \qquad s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

two independent sample t-test (equal variance!) ~t(n1+n2-2)

$$t = \frac{\left(\bar{x}_1 - \bar{x}_2\right) - \left(\mu_1 - \mu_2\right)}{\sqrt{s_p^2\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

where

$$s_p^2 = \frac{\left(n_1 - 1\right)s_1^2 + \left(n_2 - 1\right)s_2^2}{n_1 + n_2 - 2}$$

pair sample t test ~t(n-1),

$$t = \frac{\bar{d} - (\mu_1 - \mu_2)}{\frac{s_d}{\sqrt{n}}}$$

two sample t-test (unequal)

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_{\bar{\Delta}}}$$

where

$$s_{\bar{\Delta}} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}.$$

the following formula will tell us the probability of getting k successes from n observations of the random variable when the probability of a success equals p: P(K|n, p) = (n choose k) * p^k * (1-p) ^ (n-p)

when n is large, np > 5, binomial test can be approximated to a z test

binomial distribution is approximated with normal (np, sqrt(np(1-p))

normal 95th quantile = 1.65, 97.5th quantile = 1.96; normal 68-95-99.7

Nixon thinks that he's a better at the game "rock, paper, scissors" than his friend Kissinger. To find out if this is the case, he challenges Kissinger to 49 bouts of rock, paper, scissors. Nixon wins 31 of these bouts. Can we reject the null hypothesis that the two men are equally good at the game (i.e., P(Nixon wins)=P(Nixon loses)=.5) at an alpha level of .05?

Because our measurements are binary (Nixon wins or loses), the null hypothesis is binomially distributed with the following parameters: n=49, p=.5. Because n is large we can approximate the distribution with a normal distribution with a mean of 24.5 and standard deviation of 3.5. We can now conduct a z-test.

Null Hypothesis	$N(\mu=24.5, \sigma=3.5)$
Alternative Hypothesis	$\mu>24.5$
Tail of Test	upper tailed
Type of Test	z-test
Alpha level	$\alpha=.05$
Critical Value(s) of Test Statistic	z=1.65
Observed Value of Test Statistic	$z(n=49)$=1.86
p-value of Observed Value of Test Statistic	p=.0314
Conclusion	Reject the Null Hypothesis

1.65 is our critical z-score because just less than 5% of the area under the standard normal distribution lies between it and positive infinity (4.95% of it to be more exact). To get out test statistic, we convert the number of Nixon's victories, 31, into a z-score:

$$z = \frac{x-\mu}{\sigma} = \frac{31-24.5}{3.5} \approx 1.86$$

Since the z-score of our sample exceeds our critical z-score, we reject the null hypothesis that Nixon and Kissinger are equally good in favor of the alternative hypothesis that Nixon is better. To get a sense of how significant 31 victories are, we compute the p-value of our sample, 3.14%, which is the percentage of the area under the standard

normal distribution that lies between 1.86 and positive infinity. Our hypothesis test is thus concluded.

Note, another way you could have performed the binomial test is to have used the MEAN number of wins rather than the TOTAL number of wins. This might be an easier way to communicate your results (i.e., It might be easier to understand that Nixon won 63% of the time than it is to understand that he won 31 out of 49 times). If you take this approach, the mean and standard deviation of the null hypothesis are:

$$\mu = p,\ \sigma = \sqrt{p(1-p)}$$

and the mean and standard error of the mean (i.e., the standard deviation of the sampling distribution of the mean) are:

$$\mu_{\bar{X}} = p,\ \sigma_{\bar{X}} = \sqrt{\frac{p(1-p)}{n}}$$

Our observed z-score remains the same:

$$z = \frac{\bar{X} - \mu_{\bar{X}}}{\sigma_{\bar{X}}} \approx \frac{.6327 - .5}{\frac{.5}{\sqrt{49}}} \approx \frac{.1327}{.0714} \approx 1.86$$

as does the rest of our hypothesis test:

Null Hypothesis	N(μ=.5, σ=.5)
Alternative Hypothesis	μ>.5
Tail of Test	upper tailed
Type of Test	z-test
Alpha level	α=.05
Critical Value(s) of Test Statistic	z=1.65
Observed Value of Test Statistic	$z(n=49)$=1.86
p-value of Observed Value of Test Statistic	p=.0314

- Curse of dimension: especially relevant in clustering algorithms relying on distances Certain phenomenas only appear in high-dimension space. When volume of space increases, data become very sparse. It would require huge amount of data for certain statistical significance (samping issue arises). And, distance functions cannot work as distances are very similar between pair of data points in high dimension. Therefore, KNN cannot work ('Hubness': more data points appear in more KNN lists of other data points). Similary, anomaly detection cannot work (should use one-class svm to resolve!)
- Data cleaning:
    - Remove duplicate and irrelavant observations (可以参考 exploratory analysis)
    - Fix structural errors in features (typos, capitalization, symbols..)
    - Check mislabeled class, for exp, 'N/A' and 'Not Applicable' should be same. Some feature names have both abbreviation and full name – those should be one value
    - Outlier (anamoly detection), graph/boxplot, z-score threshold, DBScan (DBScan is a density based clustering alrogithm, finding neighbours by

density on a n-dimensional sphere. It can define core point, border point and outliers (out of cluster, not density-reachable))

- Handle missing data: 1. dropping observations that have missing values; 2. Imputing the missing values based on other observations

  The best way to handle missing data for categorical features is to simply label them as a new class for the feature.

  For missing numeric data, you should flag and fill the values: Flag the observation with an indicator variable of missingness. Then, fill the original missing value with 0 (or mean!) just to meet the technical requirement of no missing values.

- Dimension reduction:
  - Feature selection: some supervised learning have built-in feature selection, like regularized regression, random forest.

    Variance thresholds method: set Variance thresholds to remove features whose values don't change much from observation to observation (i.e. their variance falls below a threshold). These features provide little value.

    Stepwise search method: For forward stepwise search, you start without any features. Then, you'd train a 1-feature model using each of your candidate features and keep the version with the best performance. You'd continue adding features, one at a time, until your performance improvements stall. Backward stepwise search is the same process, just reversed: start with all features in your model and then remove one at a time until performance starts to drop substantially.
  - Feature extraction: deep learning has built-in feature extraction through hidden layers (extract increasingly useful representaion from raw input data through consecutive hidden layers)

    PCA (unsupervised): Orthogonal transformation on covariance matrix to get a series of uncorrelated variables, principle components. They have order of importance, such as the first principal component has the largest variance, it explains the most variation of original variables set.

    PCA creates linear combinations of the original features. The new features are orthogonal, which means that they are uncorrelated. Furthermore, they are ranked in order of their "explained variance (The first principal component (PC1) explains the most variance in your datase) And PCA needs data normalization!

    LDA Linear Discriminant analysis (Supervised): also creates linear combinations of your original features. However, unlike PCA, LDA maximizes the separability between classes.

- Feature Engineering:
  - Could add indicator variables (binary variable 0/1) based on domain knowledges
  - Create Interation features: product/sum/diff of two features
  - Group sparse classes (cate'gorical feature with very few observations) or similar classes into one class
  - Add dummy variable for categorical features
  - Remove redundant features, like ID, text description

- P >> N
  You can use AIC or BIC to penalize models with more predictors. You can choose random sets of variables and asses their importance using cross-validation. You can use ridge-regression, the lasso, or the elastic net for regularization. Or you can choose a technique, such as a support vector machine or random forest that deals well with a large number of predictors.

- Imbalanced dataset
  比如 credit card fraud or electricity theft 都是小概率事件（p<5%）The accuracy measure from regular consfusion matrix cannot show classifier's performance （如果 assign all prediction to the majority class, accuracy can still be very high)
  solutions:
  https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/
    - resampling original data to provide balanced classes.
      Like over-sampling to increase the instances in minority class by replicating thoses instances many times and resample oveall:
      例如 Total Observations = 1000
      Fraudulent Observations =20
      Non Fraudulent Observations = 980
      Event Rate= 2 %

      方法：In this case we replicate 20 fraud observations 20 times.
      Non Fraudulent Observations =980
      Fraudulent Observations after replicating the minority class observations= 400
      Total Observations in the new data set after oversampling=1380
      Event Rate for the new data set after under sampling= 400/1380 = 29 %
      Pros: no information loss, solve class imbalance. Cons: could lead to overfit as it replicates minority events

      或者 cluster-based over samlping. 先对两个 class 分别作 k-means clustering, then each cluster is oversampled such that all clusters of the same class have an equal number of instances and all classes have the same size
    - resolve class imbalance in algorithms, 比如 bagging method (train model on each bootstraped samples and ensemble all classifiers into one strong classifier). Boosting method trained weak classifier serially and increased weight on misclassified instances, which automatically resolve the class imbalance issue.

- Model comparison

MODEL	PROS	CONS
SVM	Kernels can embed non-linear decision boundaries than logistic regression (kernal matrix n*n). Good for high-dimension. Robust against overfitting especially in high-dimensional space (regularized by cost parameter, and gamma in RBF kernel)	Not suitable for large data (N is large). Memory intensive. Random forest usually performs better than SVM
Random Forest	No dummy variables needed. No need to scale variables (not like SVM using geometry distances to find decision boundary). Suitable for both numeric and categoric variables. Usually very low bias. Can figure out variable importance. RF methods can handle large amount of training data efficiently and are inherently suited for multi-class problems. Easier to tune than GB (number of trees and number of features selected in each split). Harder to overfit than GB.	~~Hyperparamters tuning needed.~~ Prediction function not continuous. Might not very interpretive in large trees. A large number of trees may make the algorithm slow for real-time prediction. Might underperform when only small subset of features dominates the relevance for the output (due to feature sampling)
Gradient boosted macine (GBM)	Low bias. Works well on highly unbalanced dataset (like credit card transaction). It performs the optimization in function space (rather than in parameter space) which makes the use of custom loss functions much easier (RF cannot). Boosting focuses step by step on difficult examples that gives a nice strategy to deal with unbalanced datasets by strengthening the impact of the positive class.	Compared with RF: take longer time to train. Easier to overfit than RF. More hyperparameters to tune (subsample, shrinkage, number of iterations, and tree parameters)
Logistics regression	Output is a nice probability interpretation. Easy to train and regularize.	Lack of non-linearities.
Naïve bayes	Easy to implement and scalable with large dataset.	Naïve assuption doesn't hold in most of times.
K means	Easy to implement	Performance highly dependent on what K is

		defined. Initial centroids determination also impact performance. objective never increases, but no guarantee to find minimizer. General recommendation is to re-run with several random starting initial centroids

Interview questions

1. 从深度上讲，一方面，能完整的掌握几种机器学习的算法。不仅仅知道算法是干什么的，更要知道与之相关的数学推理、技术细节。比如 Naive Bayes 怎么利用 Naive Assumption 简化，比如 AUC 为什么对于数据不平衡问题不敏感等等。另一方面，能够对算法进行横向比较。比如什么情况下 Random Forests 比 Gradient Boosting 好，什么情况下不如 GBM，为什么选择随机森林而非其他模型，比如朴素贝叶斯或者支持向量机

2. 从经验上讲，侧重的是考察与实际项目有关但是在课堂或教科书里一般不会涉及的内容。比如如何进行 feature engineering，如果数据量比 feature 量少怎么办，如何解决 imbalanced data classification 的问题，如果模型的 performance 没有达到预期应该怎么办等等。
3. 什么是 p 值，置信区间，最大似然估计，中心极限定理，大数定律
4. 与 data 相关的题目被问到的几率相对较高，比如 find median from data stream, median of two sorted array 等。另外，简化版的 k-means, tf-idf 等机器学习的算法也有被要求过现场写代码
5. case study: 首先明确问题并将其转换成建模问题，然后确定需要什么样的数据，之后进行 feature 的构建及选择，模型评估方法的选择，模型的构建与测试评估，最后谈一下结果的 deliver
6. 有没有遇到过 xxx 问题”也是一类经常被问的问题。 这类问题典型的有：有没有遇到过数据不足的情况；有没有缺少可信的 labeled data 或者数据质量突变的情况；如果数据量太大不能放到内存或者一张硬盘中应该怎么处理等等
7. 模型简介 - 这类问题同样是机器学习面试中最普遍最常见的一类问题，面试的形式一般为介绍一个你最喜欢的模型，或是介绍项目中应用的某种模型。与项目简介相同，模型简介也应力求简洁，用最简短的几句话，讲清楚模型是用了什么样的原理完成了怎样的目标。wikipedia 中关于随机森林的定义给我们提供了一个非常好的学习模板，可以用来借鉴：

Random forests is a notion of the general technique of random decision forests that are an ensemble learning method(怎样的方法) for classification, regression and other tasks(解决了什么问题), that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees(基本原理).

8. 模型的优缺点比较 随机森林有什么优点，如：a. 对于很多数据集表现良好，精确度比较高；b. 不容易过拟合；c. 可以得到变量的重要性排序；d. 既能处理离散型数据，也能处理连续型数据，且不需要进行归一化处理； e. 能够很好的处理缺失数据；f. 容易并行化等等。
在模型的优缺点中，我们提到了随机森林可以对变量重要性进行排序。相应地，我们应该能够解释随机森林是如何对变量重要性进行排序，有哪几种常见的排序指标，比如利用 OOB 误分率的改变或者分裂时信息增益的变化等。当然，问题并非到此终止，基于上面提到的两种常见的变量重要性排序指标，又可以衍生出新的问题。例如，针对 OOB 误分率这个指标，解释一下什么是 OOB，随机森林中 OOB 是如何计算的，它有什么样的优缺点；针对信息增益，同样会有很多与之有关的问题，如什么是信息增益，如何计算信息增益，什么是熵，什么是 GINI 指数，他们之间的区别是什么，他们之间的区别会对建树产生怎样的影响等。

machine Learning 常见面试问题

1). What is overfitting? / Please briefly describe what is bias vs. variance.
2). How do you overcome overfitting? Please list 3-5 practical experience.
3) What is 'Dimension Curse'? How to prevent?
3). Please briefly describe the Random Forest classifier. How did it work? Any pros and cons in practical implementation?
4). Please describe the difference between GBM tree model and Random Forest.
5). What is SVM? what parameters you will need to tune during model training? How is different kernel changing the classification result?

6). Briefly rephrase PCA in your own way. How does it work? And tell some goods and bads about it.
7). Why doesn't logistic regression use R^2?
8). When will you use L1 regularization compared to L2?
9). List out at least 4 metrics you will use to evaluate model performance and tell the advantage for each of them. (F1 score, ROC curve, recall, etc…)
10). What would you do if you have > 30% missing value in an important field before building the model?

统计常见面试问题

1). What is p-value? What is confidence interval? Explain them to a product manager or non-technical person.. (很明显人家不想让你回答: 画个正态分布然后两边各卡 5
2) How do you understand the "Power" of a statistical test?
3). If a distribution is right-skewed, what's the relationship between medium, mode, and mean?
4). When do you use T-test instead of Z-test? List some differences between these two.

5). Dice problem-1: How will you test if a coin is fair or not? How will you design the process(有时会要求编程实现)? what test would you use?
6). Dice problem-2: How to simulate a fair coin with one unfair coin?
7). 3 door questions. (自行 google 吧，经典题之一)
8). Bayes Questions: Tom takes a cancer test and the test is advertised as being 99% accurate: if you have cancer you will test positive 99% of the time, and if you don't have cancer, you will test negative 99% of the time. If 1% of all people have cancer and Tom tests positive, what is the prob that Tom has the disease? (非常经典的 cancer screen 的题，做会这一道，其他都没问题了)
9). How do you calculate the sample size for an A/B testing?
10). If after running an A/B testing you find the fact that the desired metric(i.e, Click Through Rate) is going up while another metric is decreasing(i.e., Clicks). How would you make a decision?
11). Now assuming you have an A/B testing result reflecting your test result is kind of negative (i.e, p-value ~= 20%). How will you communicate with the product manager?
If given the above 20% p-value, the product manager still decides to launch this new feature, how would you claim your suggestions and alerts?

SQL 常见题

1). What is the difference between union and union all? where and having?
2). Table【in_app_purchase】:
uid: unique user id.
timestamp: specific timestamp detailed to seconds.
purchase amount: the amount of a one-time purchase.
This is a table containing in-app purchase data. A certain user could have multiple purchases on the same day
Question 1: List out the top 3 names of the users who have the most purchase amount on '2018-01-01'
Question 2: Sort the table by timestamp for each user. Create a new column named "cum amount" which calculates the cumulative amount of a certain user of purchase on the same day.
Question 3: For each day, calculate the growth rate of purchase amount compared to the previous day. if no result for a previous day, show 'Null'.
Question 4: For each day, calculate a 30day rolling average purchase amount.
3). Table【Friending】
time = timestamp of the action
date = human-readable timestamp, i.e, 20108-01-01
action = {'send', 'accept'}
actor_id = uid of the person pressing the button to take the action
target_id = uid of another person who is involved in the action
Question: what was the friend request acceptance rate for requests sent out on 2018-01-01?
...

题目二涵盖了简单的 aggregate 问题，cumulative 问题，rolling window 问题等等。搞定这些，其他的都只是一些简单变形。

题目三涵盖了 self-join，并且有一些 tricky 的大于等于号的应用，有兴趣可以在地里查一下 Facebook 面经的解答。

其他的题目无非是多了一些 table，join 麻烦一些或者加了一些 case when，难度都不会有太大的变化。做好几个经典题，然后自己整理好就可以以不变应万变了

Product sense 常见问题

常见面试问题

1). Today you immediately notice that our app's new users are doubled. What could be the reason? Do you think it's good or not?

2). If we have an app with in-app purchase, name at least 4 metrics you would like to monitor in your dashboard.

3). If you are running an A/B testing and find that the result is very positive, thus you decide to launch it. In the first 2 weeks, the performance of our website is very positive.

However, with time flying by, all metrics seem to go back to normal. How will you explain this result?

4). Assume we are Facebook and we would like to add a new 'love' button. should we do this?

5). We are running 30 tests at the same time, trying different versions of our home page. In only one case test wins against the old home page. P-value is 0.04. Would you make the change?

6). If after running an A/B testing you find the fact that the desired metric(i.e, Click Through Rate) is going up while another metric is decreasing(i.e., Clicks). How would you make a decision?

7). Assume that you are assigned to estimate the LTV(lifetime value) of our game app player. what kind of metrics would you like to calculate so as to make a good prediction?

Assume that you already collect all that you want. How would you make this prediction/estimation?

8). If you got a chance to add on new features for our app to increase our profit within a very short term. What will you do?

你可以发现的是，大多是围绕着 metrics 和如何提高 product performance 来展开的

小 k：

- 先是大面上：你需要理解这个产品，你最好经常用这个产品，知道大体上它是做什么的，有什么功能，谁会去用它，大体的 business model 什么样，产品最需要取悦谁？怎么做到的？什么因素会影响用户的满意度，回头率或者 retention rate/churn?
  举例：MITBBS 这个网站，怎么回答上述这些问题？
  用户有：

潜水的大多数（提问：大约什么比例的用户会发帖？how would you measure it?）
发帖的小小部分
广告商（提问，你估计一下大概广告费用是多少？他们的费用是怎么决定的？）
jiaoyou 用户（这些人跟 MIT 注册用户是同一批人吗？什么比例在交友有账号？什么比例有收费账号？他们里面转化率如何？他们一般缴费多久？life time value 如何计算？etc）

- 你认为网站可以收集什么数据？应该收集什么数据？收集了然后怎么用来回答类似这样的问题：
    - 有什么什么功能你想怎么提高？看什么样的 metric 可以知道产品的健康程度？一个就够吗？不够的话要你多设计几个，你用哪几个？为什么
    - 假定有 ABCDE 种 metrics，A 变大，BC 变小了，如何解释？全都变大如何解释？现有信息够得出结论吗？够，为什么，不够，还需要什么？为什么？你提出的 metric 如果可以 track 短期的产品健康度，那长期呢
    - 还是上述 MIT 的例子，如果要提高广告收入，怎么办？如果想提高浏览量怎么办？如果想提高 engage 的人数怎么搞？首页放多少条广告最好?
    - 如果要提高 jiaoyou 用户 subscription conversion rate 怎么办？如果是提高 retention rate 怎么办？
    - 短期把所有首页条文都做成广告会短期提高收益，但是长期这么行吗？不行的话，长期健康程度如何衡量？
- 假设你提出了改进产品的方法，需要做实验来验证，如何设计这个实验？
  继续用 MIT 例子，想知道首页广告条数多少条能最大化广告收益，怎么设计？
  随机对照怎么 allocate? 怎么计算 metric? 怎么衡量你看见的结果是否可信？

A/B testing

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=470867&extra=&page=1
https://towardsdatascience.com/a-summary-of-udacity-a-b-testing-course-9ecc32dedbb1

A/B testing (sometimes called split testing) is basically statistical hypothesis testing applied to web page comparison. You compare two versions of web pages by showing the two variants (call them A and B) randomly to two equally sized groups of visitors at the same time, the one that gives better conversion rate wins.

1) Describe the process of A/B test
   A/B test can be summarized into the 5 steps below:
   (1). choose and characterize metrics to evaluate your experiment, i.e. what do you care about, how do you want to measure the effect.

Brain storm potential metrics. Use customer conversion funnel to summarize the process. Invariant metric does not relate to the change. Evaluation metrics are related to the change.
(2). choose significant level (alpha), statistical power (1-beta) and practical significance level you really want to launch the change if the test is statistically significant. check 1point3acres for more.
(3). Calculate required sample size
(4). Take sample for control/ treatment groups and run the test
(5). Analyze the results and draw valid conclusions
Sanity check: invariant metric does not change in experiment and control

Analyze evaluation metrics
Using pooled mean/conversion probability, then calculate pooled standard deviation, then calculate margin of error (z*sd). Then compare the difference between control and experiment and calculate upper and lower bound of the difference (P-diff +/ - margin of error). Compare with 0 (statistically significant) or required difference to be practically different.

Sign test: confirm the result with sign test. The number of success out of total trial is statistically significant.

2) Situations we can't analyze through A/B test
   A/B test can't test new experience, because (1) what 's the base of your comparison (2) how much time it will take for the users to adapt to the new experience.
   Long term effect is hard to test with A/B test

3) How many variates should we have in A/B test
   The goal of A/B test should be clear. A number of factors from each different design can muddy the test result water. We suggest running two versions against each other, and then running a second test afterwards to compare the winners.

4) What do I do if I do not trust the results?
   If you really don't trust the results and have ruled out any errors or challenges to the test's validity, the best thing to do is to run the same test again. Treat it as an entirely separate test and see if you can replicate the results. If you can replicate again and again, you probably have a solid set of results.

5) What if I do not have control?
   A control is the existing version of a landing page or webpage that you are testing against. Sometimes you may want to test two versions of a page that never existed before… and that's oaky. Just choose one of the variations and call that one the control. Try to pick the one that's the most similar to how you currently design pages and use the other as the treatment.

6) When A/B test is not useful, what you can do?

Analyze the user activity logs
Conduct retrospective analysis
Conduct user experience research
Focus groups and surveys
Human evaluation

7) Metrics

The metrics we choose for sanity check are called invariant metrics. They are not supposed to be affected by the experiment. They should not change across control and experiment groups.

Evaluation metrics are used to measure which variation is better. For example daily active users (DAU) to measure user engagement; click through rate (CTR) to measure a button design on a webpage.

There are four categories of metrics:

- Sums and counts
- Distribution (mean, median, percentiles)
- Probability and rates (click through probability and click through rate)
- Ratios: any two numbers divide by each other

Sensitivity and robustness:.
You want to choose a metric that has high sensitivity, so the metric can pick up the change you care about. You also want the metric to be robust against changes you don't care about. There is a balance between the sensitivity and robustness, you need to look into the data to find out which metric to use.
How to measure the sensitivity and robustness?

- Run experiments
- Use A/A test to see if metrics pick up difference (if yes, then the metric is not robust)
- Retrospective analysis

8) Significance level, statistical power and practical significance level

Usually the significance level is 0.05 and power is 0.8. practical significance level varies depends on each individual test. Practical significance level is higher than statistical significance level. You may not want to launch a change even the test is statistically significant because you need to consider

- The business impact of the change
- Whether it is worth to launch considering the engineering cost, customer support, sales issue and opportunity cost

9) How to estimate(calculate) sample size?

Sample size required for valid hypothesis test depends on 5 of the following parameters

1. The conversion rate value of control variation (baseline value)
2. The minimum difference between control and experiment which is to be identified.

The smaller the difference between experiment and control to be identified, the bigger the sample size is required.

3. Chosen confidence/significance level
4. Chosen statistical power
5. Type of the test: one or two tailed test. Sample size for two tailed test is relatively bigger.

There are different kinds of online testing tools, G-power, Evan Miller, google analytics, etc. .

If using R, first calculate the z value based on alpha using qnorm(). Then using a grid of sample size values to calculate beta (the pdf of reject the null when the null is true) using pnorm(), so the smallest sample size corresponds to beta <= required beta is the required sample size for valid test. This make use of the fact that as sample size getting big, the estimated standard deviation become smaller, so the power of the test gets big.

Formula:

10) How to split sample?

The sample size in control and experiment should be statistically equal.

11) Correlational VS causal

You observe the churn rates for users using/not-using your feature:

25% of new users that do NOT use your feature churn (stop using product 30 days later)

10% of new users that use your feature churn

[Wrong] Conclusion: your feature reduces churn and thus critical for retention

Flaw: Relationship between the feature and retention is correlational and not causal

The feature may improve or degrade retention: the data above is insufficient for any causal conclusion

Example: Users who see error messages in Office 365 churn less.

This does NOT mean we should show more error messages.

They are just heavier users of Office 365

See Best Refuted Causal Claims from Observations Studies

for great examples of this common analysis flaw

(a) Common cause: women have smaller palm and on average lives longer

(b) Misses time related factors, such as external events, weekends, holidays, seasonality

12) Advantages of A/B test

Scientific way to prove causality, i.e. the changes in metrics are caused by changes introduced in the treatment.
Sensitivity: you can detect tiny changes to metrics
Detect unexpected consequences

behavior questions

（1）Leadership and how to influence others
（2）A hard challenge faced and How to solve it
（3）A true failure and how to turn it around
（4）A proud success made with team together

有一个 behavior 很好的模板叫 S（Situation）.T（Task）.A（Action）.R（Result）可以用来 frame 几乎所有的 behaviroral 和 culture fit 的素材。在准备的时候一定要强调你做了什么，如果你能够量化结果的花那就更优秀了

Product & Case question

在我看来 IT 界（不是咨询界）所有的 product 和 case question 到最后都可以被归纳到 fb 的两轮 product 面试之下： Product Interpretation 和 Applied Data。

第一个内容的最终落脚点一般都是 find a metrics to evaluate XXX。这个要求我们明白产品的用户，用户的问题，产品如何帮助用户解决问题，进而明确用户的 goal，公司的 goal，最后作为 DS，我们的任务是找到 metrics 去 quantify 这些 goal。 每一个公司，因为业务模式不同，最后都会一个独特但唯一的 north star metrics。在面试之前，想清楚这个 metrics 是什么和为什么是这个在我看来是很重要的。在面试之中，当我们 clarify 了 scope 和 ambiguous term 之后，也应该按照步骤一步一步地和面试官讨论，把问题，产品的 solution，goal 这些东西都一步一步地聊出来。有的人建议先 confirm goal，但是我觉得 goal 是在你和面试官都 align 了问题和产品后才能聊得出来的东西，这个大家如果有不同意见欢迎讨论。但是总结来说，这个部分的产品题，需要我们花时间去了解产品，然后一步步地去聊出面试官问你的问题的 context。

在选择 metrcis 的时候，一定要清楚地描述分子分母，你的 unit of diversion 是什么，你 aggregate 的 time frame 是什么。另外要注意的是，metrics 分为三种，**short term metrics， long term metrics 和 counter metrics。**第一种的特点是见效快但是描述的记过不够核心，第二个的特点就正好反过来了，比如 FB 的 CTR 就是 ST， retention rate 就是 LT。 counter metrics 是为了描述一些你不愿意看到的负向变化的。比如在 FB feed 里放更多的视频，你的 time spend 可能长了，但是你的 engagement 可能就会下降，因为视频是 passive consume 的产品，你很喜欢未必会点赞或者评论。

Applied Data

https://www.1point3acres.com/bbs/thread-483072-1-1.html

这一大类的问法都是 what data would you use to XXX（我在后面会沿用同样的格式），让你 brainstrom 用什么 data 去解决问题，也就是考察在实际工作中 operationalize data 的能力。这里可以考察的点有很多，我争取每一个自己能想到的点都举一个我自己面试的一个实例出来供大家讨论：

1） what data would you use to 描述 impact？

e.g 某平台上突然在某一个时间点上有人说出现了很多的 fake news，现在让我很短的时间出一个给 VP level 的报告用来描述该事件的影响。

楼主在被问到这个题的第一反应是 VP level 的人想要 care 什么 impact，然后就会去想这个平台 care 什么 impact，就去套 top line metrics 比如说 engagement 和 retention。后来面试完才发现这里漏了一个点：事件本身的影响范围究竟有多大？ 这个平台上有多少 fakenews 在被产生？有多少个 view 是 fake news？ 有多少用户看到了 fake news？ 后来得到的结论是，当面对影响类的问题时，在你描述它引发的问题之前，你的第一责任应该是描述问题的 scope，或者引用 UX Designer 经常会问的一个问题：先要搞清楚这是不是一个问题

（2）what data would you use to signal something/find something?

e.g 某平台希望你找出 business traveller

回答这一类问题，我觉得先要做一些功课：想想这个产品有什么类型的第一方数据，每一个类型的数据下面有可能有什么数据，比如这个例子里，我觉得我们能获得的有用户数据，用户关系数据，用户产品使用数据，以及在使用产品时留下的源数据（metadata：e.g device，ip，gps etc）。然后有时间的话，我会再想想我能怎么吧某个数据汇总，或者吧多个数据联系起来产生某种信息，这样就多了一些 derivative data points。遇到类似问题的时候就可以了调用你的信息库了。还有一个比较有用的思路是我们除了找 ‘肯定能证实的信息’ ，也可以找 ‘肯定能证伪的信息’ 。

说回这个具体的问题，首先因为是 business traveller，当然就要有 job。 然后关于 travel，我当时选择的是先找出用户的根据地，这个可以用用户信息中的地址和用户访问源数据的最经常访问的 gps 和 ip 来结合定位。然后用 gps 和 ip 确定用户在距离足够长的地方登陆的频率，根据 percentile 的 threshold 来判断。面试完发现，这个答案是有问题的。首先，我们对相当一部分用户收集不到他们的 gps 和 ip 信息，其次我们判断的方式完全没有 validation，最后我们利用的信息太少了。所以我个人的结论是：对于这类问题，不要只用 analytics 的方式去解决，应该要有 ML， 应该要花时间去 validate 并人工 label 一些数据，然后 invovle 更多的 feature。

（3）what data would you use to find the reason behind a increase & decrease of a

certain metrics ?

e.g 某手机应用商店发现某日的应用下载量下降了，怎么找原因？

这个问题在 cracking the pm interview 里有，地里的小伙伴也总结过不止一次。今天我想基于我看到的所有对这类问题的解法给一个自己认为比较全面的解：

1. 在解决这类问题前要先明白，数据在这里能提供的帮助大概率不是提供最后能用来和你的同事&面试官讨论的结论，而且提供让你们找到结论的 context

2.我们需要了解以下几个方面的“context”：

2.1 trend：sudden change or gradual change ?

seasonality ? if so， maybe its normal

any special event happened internally or externally ?（new PR，new launch,system outage,new marekting campaign from competitors

2.2 breakdown the target metrics：

这个问题里面，download 是一个 funnel 的结果，在 download 之前，有访问，点击，下载，下载完成这几个步骤，每一个步骤都有绝对的量和转化率两个数字需要关注。还有一个 metrics 本身就是 ratio，那么就要从分子分母两个方面做类似于 funnel 的拆解。

2.3 analyze segments：

by country， by OS， by OSV，by desktop/mobile/，etc.

每分析一个 segment，需要关注的点有两个，一个是这个变化是发生在一个 segment option 上的，还是全 options 都在变化。另一个是不同 segment 之间的比例有没有变化，这里涉及的就是 confounding 和 simpton paradox 的问题了。

3. 很多人做完 2 就结束了，我觉得当我们获得了足够的 context 之后，应该要和面试官再聊一下基于这些 context，我们应该去找谁 validate 什么 assumption。

Case Study

https://www.1point3acres.com/bbs/thread-330947-1-1.html
https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=111681

Case study: 重中之重！这块有点像咨询公司的 case interview，也是考察能不能把问题 break down，能不能发散把各种情况想的比较全面。不同的是，我们 ds 面 case interview，更关注从数据的角度去解决问题，从用户使用产品的 life cycle 出发去定义 metrics。但是我还是建议去看一下咨询公司面试题目，学习一下他们答题的 stucture，如何跟面试官保持一个很愉快 conversation。这里推荐 Victor Cheng 的 case study interview, 可以从喜马拉雅 fm 里面听，很适合上下班的时候听：

http://www.ximalaya.com/5269453/album/6414597?feed=reset。 下面我来总结一下典型的 case study 常见题目以及答题思路。

1) **Diagnostic problem**: 这类问题通常是发现某个 kpi 某天突然下降了，该咋办？这类问题就是考察你能不能把一个 business 问题一步一步的 break down，找到问题所在。一般的思路都是先从数据搜集的是否正确，是否这个 kpi 有 seasonality，这个 kpi 有哪些不同的 segment，等等这几点出发。举个最简单的例子：某社交网络发现用户使用 likes 的功能今天突然下降了 10%，你觉得是为啥？首先你可以问面试官，咱们的数据搜集的**正确与否**，这个功能是不是有 **seasonality,** 是不是今天有个 **speicial events** 发生（比如自然灾害，大家都断网了）。切记要 keep this as a conversation，不要你一个劲儿的在那里说说说。根据面试官给你的引导，下一步就是如何 break down, 你可以先说，likes 下降了是指哪个部分下降了，有 friends post, pages, or events。你还可以从另外一个角度 break down，比如是哪个地区的 likes 下降了。记住，**understand the context is very important**!。不要先急着答题，要先把数据的背景了解清楚。基本你把背景了解清楚了，也就找到了问题所在了。
2) **How to measure the success of a new product/feature**: 这类问题一般就是我想改变或者增加一个产品的功能，如何衡量是不是成功了。这种问题的思路一般都是先搞清楚产品的目标是什么，你可以问面试官：**what is the goal of this product?** 从目标出发，先定义衡量的 metrics，然后就是做实验了，根据实验结果来判断是否成功。在定义 **metrics** 的时候要想的全，哪些 metrics 对你的目标是最重要的，另外也不要忘了定义 **counter metrics**，我在下面会用一个例子来解释。定义 metrics 注意一点就是不要仍给面试官一堆 list，想 3 个最重要的就可以了。做完实验后面试官会有一些 follow up 问题，比如 metric a 增加，metric b 减少；或者 metric a 增加了 2%，那么这些情况下是否应该 launch 这个产品呢。我下面用一个例子来说明这类问题的思路。还是以社交网络为例子，他们改变 friends recommendation 算法，希望用户可以通过这个功能多加好友，如何衡量这个 feature 是不是成功呢。我们先明确这个功能的目标，是希望多加好友，加了好友以后，希望在平台上有更多的交涉。那么这个时候的 **metrics 可以应用 funnel 的思路来定义**：好友增加率（friends request/accept rate)，月活量（monthly active user)，参与度(engagement），**这些方面来定义 metrics，尽量想到至少 3 个。然后做 ab test，比较两个 group 的差别。**这个时候面试官就会有一些 follow up 的问题，比如我们发现 monthly active user(mau)增加了 5%， 但是 engagement 减少了 3%，那么这个产品是好是坏呢。这个时候 engagement 就是我们的 counter metrics，因为我们不仅希望加了好友以后，每个人的 connection 增加了，我们更看重是不是有意义的 connection, 所以 engagement 也是一个很重要的指标。**一般这种题目可以从短期 vs 长期效应来考虑。**比如我们的 mau 增加了，虽然短期的 engagement 减少了，但是长期看的话，由于每个人的 connection 增加了，由于 network effect，大家相互影响，engagement 长期看也许会增加的，这个时候就要去衡量 long term effect 了。然后面试官可能会接着问，假设我们的 metrics 都正向增加了，比如 2%，那么如何判定这个 2% 是个好的增长，可以 launch to everyone? 这个时候可以**往量化方面想**，比如 2%的增加对应了多

少 popualtion，这个 2%的增加带来的潜在 revenue 是多少，如果是好几百万的话，那即便 2%，也是个很好的提升！

3) **How to identify opportunity**: 这类问题一般都是在做 ab test 之前，我如何去说服别人我的假设是合理的，我的假设值得去做一些 test 来衡量 impact。这种问题很重要的一点就是 identify opportunity sizing。也就是你的假设会影响多少人，如果只影响到 5%的人，可能你带来的影响不会很大，但是如果超过 20%的话，这个时候你的影响就大了。比如某电商想加个产品线，这个时候你如何去说服你的领导这个产品线值得去做一些实验呢。首先你要告诉你的领导，**我加了这个产品线以后，会影响多少人，会带来潜在的多少收入**。接下来就是如何去做这个产品线，比如我们应该 **target 目标群是啥，应该以什么样的方式让目标群更好的了解我们的新产品线**。再接下来就是**如何做实验**了，**这样就回到了上面的 how to measure the success 的问题**。从这里大家也可以看出，data science 其实是一环套一环，有一个完整的周期的，我推荐的那篇 airbnb 的 blog 也说明了这一点。

做实验： 常见问题：如何 randommize sample, 如何决定 sample size, 决定 sample size 的几个因素对 sample size 如何影响，如何决定 test 跑多久，什么是 p value, confidence interval, type 1, type 2 error， 熟悉 t test, z test 公式跟原理。要注意不能只把定义背下来，要真正理解了，并且可以给 non technical 的人解释清楚。可以参考 penn stats 的假设检验章节：https://onlinecourses.science.psu.edu/stat414/node/290

Data challenge

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=326201&extra=page%3D4

（1）天马行空地去 brainstrom，从最直接能想到的点去分析，到开始尝试一些需要思考才能想到的点，想到什么就分析什么，看看数据会不会带给你惊喜。

（2）判卷子的人最在乎的是你这一通分析对别人的价值，不太在乎你做的多辛苦，所以不要吧自己熬了几个晚上做的所有东西都写进报告里。报告里的东西越少越好，但你要在你的分析中找出那些是对解决问题最后价值的，按次序有选择地 showcase 你的 deliverable。

（3）如何 frame solution？ 我的看法是：describe 图表-->总结出 insights-->给出 recommendation。describution，insight， recommendation 是一个完整的逻辑闭环，它能帮助批卷子的人很快地明白了发现了什么，总结出了什么，并且依据你的总结准备建议出什么。

这种 take home data challenge 的难点在于问题比较开放性+时间限制。短则 3-4 个小时，长的最多一周。下面我来说一下前期准备工作，以及拿到题目后如何短时间内把握住要领，写出面试官满意的报告来。

前期准备：

代码熟练：不管是 sql,或者 r, python，随便你选，但是一定要选你用的比较熟练的。因为你要短时间内完成数据分析+写报告，如果代码不熟练的话可能做不完。建议可以先准备一些模版，比如画图的，做模型的，做 ab test 的。我用的 python，所以画图都是 seaborn + matplotlib, 需要建模一律用 random forest from h2O package。这里强烈推荐 h2O random forest，自带 auto bin 的功能，解决了 categorical level 多的问题。不需要将 categorical variable 转化成 numerical（对于 python 同学来说），不需要 impute missing value。至于我为什么只用 random forest, 下面会讲到

预习一些题目：这里推荐买这本书 “A Collection of Data Science Take-Home Challenges”。我以前买的时候可以单独买这本书，50 块，现在好像得买整个 package，有些小贵。这本书主要是给了几个例子，以及用 r 来做的详细解答。非常好的参考例子，我就是看了这个书以后才开窍的

下面言归正传，题目拿到手以后改咋办：

1) 明确产品的目标： 一般都会给你描述一个产品，比如某社交网络公司想提高 rentention rate，某电商公司想提高 conversion rate。你下面的所有的分析一定要围绕这个目标来做。这个说起来容易，但是很多同学题目拿到手，都会脑补很多东西，想的太多了，反倒无从下手。建议就从跟产品目标最直观的开始分析

2) 定义 metrics：在清楚了产品的目标以后，哪些 metrics 可以用来衡量产品的成功与否呢。对于互联网产品，基本都是从 user acquisition, retention, engagement. monetization 相关的这些目标来定义 metrics 的。多了解用户使用产品的漏斗模型（AAARRR)。然后定义 metrics 的时候思考产品特点以及目标，往漏斗模型上面靠，每一层应该用什么 metrics 来衡量。可以看这篇科普的：
http://startitup.co/guides/374/aarrr-startup-metrics

3) 数据清理：也就是所谓的 data cleaning。基本就是看看哪些变量的 missing value 太多了，或者某个变量只有一个 level。这种情况下可以去掉那些没什么用的数据。另外如果你用 h2O random forest 建模，不用去 impute missing value。

有些公司的会很注重 data cleaning & processing，说白了就是 data 里面有雷。没有发现的话都是会被扣分的。除了简单 duplication 和 missing 之外，还要想一些和 biz case 相关的东西。比如时间上是不是 make sense，比如有没有可疑地 fraud data 等等。不然花了很多时间做 model 或者分析，因为这些小东西一眼没看到被扣分很不值当。

4) 提取跟产品目标相关的变量：比如 uber 想提高 driver rentention rate，你拿到数据后，看一下每个变量都什么意思，想想哪些变量有可能跟目标相关。下面说一下我遇到的比较普遍的需要做一些 data munipulation 的相关变量

- 时间变量：可以提取 day of week, month, time of the day 这种变量。还有一些 time difference， 比如 user sign up date，first time use this product，这里面的时间差也就是用户登记后多久开始使用产品，这也会是一个很重要的变量。
- 需要求平均值，次数求和这种变量：比如一周内使用了多少次产品，平均每次花了多少钱
- 去掉跟结果直接相关的变量：比如某个变量跟结果是显而易见的相关，虽然加入这个变量你的模型预测准确度达到 99.9999%， 但是对于你后面做的产品推荐没有任何意义。比如某电商想看看用户的哪些行为能够促使最后花钱买产品，有个变量是是否到了 check out 页面。很显然用户到了 check out 页面，购买的意向就已经很高了。在建模的时候要去掉这个变量，因为不用分析就知道这个变量重要。

5) 如何鉴别重要的变量：一般的问题都是让你鉴别哪些变量对结果影响最大。选 3-4 个重要变量即可，千万不要把所有的都分析了，因为你没有时间！下面说两种我常用的方法

- 看分布：比如你觉得 time difference 是个很重要的变量，可以画个 box plot，或者 histogram，分别对 retain and churn 的人做图
- 直接用模型：根据模型结果看 feature importance。我只用 random forest。因为第一我建模的目的只是为了看哪个变量重要，并不需要很精确的预测；第二用 h2O 的 random forest 基本不用调试，结果就很不错了；第三我觉得 random forest 在鉴别 feature importance 比别的模型要好，因为它每次是取所有变量的一个子集来建立决策树，所以每个决策树选的变量都不太一样。最后平均下来看哪个 feature 最重要。感觉这种算法更可靠一些。不过哪种模型不重要，关键你把重要的变量选出来就好。这里提示一点：千万不要花时间去把模型调的很精确，只要模型结果可以接受就行。因为你是在做分析，你的重点是在做后面的产品改进推荐

6) 产品改进推荐：也是最最最重要的一点！很多同学做模型啊，分析啊做的天花乱坠，然后都挂在这步了。一定要记住一点，你的模型是为了产品推荐用的，不是为了 production 用的。比如你发现用户登记以后越快使用你的产品，他们的 rentention 越高，那么就要想办法如何让用户尽快使用你的产品。你不能只说让用户尽快使用产品，要给出更具体的建议。比如给登记的用户发 promotion，第一次购买可以便宜一些。有的职位偏重 AB test，会问你接下来如何设计实验来测量你的推荐的有效性

7) 实验设计：必看资料是 udacity 上面的 AB test by Google https://www.udacity.com/course/ab-testing--ud257. 一般做题目常用的无非就是 test mean difference or proportional difference (t and z test), 上面都讲的很清楚应该如何做，如何选 sample size。下面简要说一下如何分析结果

- 影响有多大：也就是 what is the opportunity sizing. 这一点很重要，如果你的产品推荐只会对很少一部分人有影响，比如小于 5%，那么你这个推荐是没有用的。但是有一个特例就是如果你那 5%的人可以带来好几个 million 的收入增加，那么还是值得做的。
- 分析比较结果：比较爱问的问题有
  - Is this amount of lift enough? 比如做实验后发现有 2% lift，这个结果好不好？这种题目一般就要看 2% lift 带来的实际影响，比如 2%带来了几个 million 的收入的增加，那么就是好的。
  - Metric A going up, Metric B going down, should we still launch this product? 一般看哪个是最重要的 metric，另外就是有些 metric 需要时间长一些才能看出来。比如某个社交网络的用户参与度增加了但是用户增长变慢了，假设这个产品改善后是希望增加用户的参与度。这种情况就要考虑 network effect，随着用户的参与度的增加，用户的 connection 也会受到影响，久而久之他们也会变成日活或者月活的用户。

8) 分析做完了，写报告应该注意啥：

- 思路清晰，言简意赅：看似是废话，但是很多同学，包括我以前，都恨不得做个特别复杂完美的图跟表格，然后展示给面试官我的技术有多牛掰。其实他们更看重的是你的分析是不是通俗易懂，非 technical 的人能不能一看你的图或者分析就知道怎么回事了。
- 图文并茂：这里强烈推荐大家都鄙视的 excel 作图功能，个人觉得比 seaborn, ggplot, matplotlib 都好用多了。也许是因为我代码能力不强，改个图得 debug 半天，还经常弄不出自己想要的效果，但是用 excel 简直是神器，轻松做出非常专业的图来，改起来也很方便。我一般简单的图，比如 boxplot, heatmap，用 seaborn 这种直接出，但是要做一些复杂的 cohort analysis，就上 excel 了。
- 不要写的太长：很多同学把 data challenge 当成论文来写，弄个几十页的报告，把能分析的都分析了一遍，结果还挂了。因为人家面试官根本没有时间看你的论文报告。确保他们花 10-15 分钟时间能把你的分析跟结论看懂。

9) 最后总结一下主要步骤： 明确产品目标， 定义相关 metrics，建模去预测关键指标，模型结果对产品改进有啥建议。希望这篇总结能对正在战斗或者打算战斗的战友们有点帮助。

1) 编程

第一题

https://www.geeksforgeeks.org/ma ... g-character-string/

用 linear time

第二题

https://www.geeksforgeeks.org/ma ... -the-end-of-string/

第三题 dp

lara 可以做两种 bonquets A.三朵玫瑰 value p， B.一朵玫瑰一朵波斯菊 value q

给出一串 stirng 0=玫瑰 1=波斯菊

eg 10001000 可以做 2 份 b 一份 a 或者两份 a

求最大获利

还有四个 complexity 的问题

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=499449&highlight=VIA

第一道是曼哈顿距离，给 manhattan distance function, 处理 input 格式, call function 后, print 出 distance 即可

第二题是 get tickets to fans。有很多帖子有再讨论可以翻一下, 主要作法是创建了一个新的 class, Event, 里面有 event 的 id, position, 各种票价, 对于每一个 user, 需排序 events,依照 manhattan distance -> 最便宜的票价-> id 作为排序的依据, 再遍历每一个 event, 找出最适合的 event 及价格, 最后再注意票卖光了的 case 即可

后一个是 Getting Listins From Supplier:
https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=488236
这部分我另外写了一个 Supplier 的 class, initial 根据不同 supplier (A or B)去 map 相对应的 function 执行, 但对外的街口都是一致的, 因此若之后有新的 Supplier 加入, 只需在这新增该 supplier 的 function 即可. 不知道还有没有更好的思路

编程 是关于重构代码的，使其能够 add new 10 supplier 更加容易
https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=510111&highlight=VIA

2) 问做过的项目，怎么选 model，怎么做 feature selection 并举例，怎么 Evaluation， 结果. 询问了 A/B test， metrics, 设计 model， 结论如何应用，production