For each of the questions below, answer as if you were in an interview, explaining and justifying your answer with two to three paragraphs as you see fit. For coding answers, explain the relevant choices you made writing the code.

1. We A/B tested two styles for a sign-up button on our company's product page. **100** visitors viewed page **A**, out of which **20** clicked on the button; whereas, **70** visitors viewed page **B**, and only **15** of them clicked on the button. Can you confidently say that page **A** is a better choice, or page **B**? Why?

   Answer: We can assume that there is no difference between page A and page B. The click rate for page A is 20/100 (20%) and for page B is 15/70 (21%). Suppose the significant level α is set to 5%, the rejection region is 2.5% for such a two-tailed test. For a standard normal distribution, the value will be larger than $\mu + 2\sigma$ or less than $\mu - 2\sigma$, which are basically larger than 2 or less than -2. However, the difference between page A and page B is (20 – 21) which is within the range $\mu - 2\sigma$ to $\mu + 2\sigma$. Therefore, page A and page B should not be considered as significant difference.

2. Can you devise a scheme to group Twitter users by looking only at their tweets? No demographic, geographic or other identifying information is available to you, just the messages they've posted, in plain text, and a timestamp for each message.

   In JSON format, they look like this:

   {

    "user_id": 3,

    "timestamp": "2016-03-22_11-31-20",

    "tweet": "It's #dinner-time!"

   }

   Assuming you have a stream of these tweets coming in, describe the process of collecting and analyzing them, what transformations/algorithms you would apply, how you would train and test your model, and present the results.

   Answer:

   We can use unsupervised machine learning method called Latent semantic analysis (LSA) which is one of natural language processing techniques based on singular value decomposition (SVD). Specifically, tweets can be transformed and TF-IDF information of each tweets will be extracted. Then we can cluster the tweets based on the TF-IDF data by using K-mean classification method. Once the classes have been determined for the collected tweets, we can use supervised machine learning method to train a classification model. The model can be validated with the validation data set and tested with testing data. This whole method also Once the model is established, any new tweet can be classified based on its TF-IDF data using the established classification model.

3. In a classification setting, given a dataset of labeled examples and a machine learning model you're trying to fit, describe a strategy to detect and prevent overfitting.

   Answer:

   For any machine learning algorithm, the prediction error can be broken down into three parts: Bias Error, Variance error and irreducible error. The irreducible error cannot be reduced regardless of what algorithm is used. As a machine learning practitioner, our goal is to achieve low bias error and low variance error. However, there is tradeoff between bias and variance. In general, bias error is caused by a too simple model and variance error caused by a too complex model. If a model is too complex, it always fails to generalize well. In such a scenario, it causes overfitting.

   To detect if a model has overfitting problem, one can compare the training error and validation (or testing) error. If the validation error increases while the training error steadily decrease than a situation of overfitting may have occurred. One of the most commonly solutions for overfitting is using k-fold cross validation. Increasing the size of data set is another way to solve the overfitting problem. For unbalanced classification problem, one should pay attention to what metric is more appropriate. For example, breast cancer prediction and credit card fraud detection, the accuracy is always very high and the accuracy itself only reflects the underlying class distribution.

4. Your team is designing the next generation user experience for your flagship 3D modeling tool. Specifically, you have been tasked with implementing a smart context menu that learns from a modeler's usage of menu options and shows the ones that would be most beneficial. E.g. I often use **Edit** > **Surface** > **Smooth Surface**, and wish I could just right click and there would be a **Smooth Surface** option just like **Cut**, **Copy** and **Paste**. Note that not all commands make sense in all contexts, for instance I need to have a surface selected to smooth it. How would you go about designing a learning system/agent to enable this behavior?

   Answer:

   We can design a learning system by applying reinforcement learning knowledge. In this particular case, there are different actions (a) that a user can take. For example, there are actions such as "right click", "click edit button" and "click something else". Each action results in different states (s). For example, if you click the 'edit', you will end up with 'Surface'. For each action, we can assign a different reward with the "right click" action has the highest score because this the favorable action. We also need a learning table (Q-table) which updates the Q score associated with each action with appropriate learning rate. The Q score for a given state(s) and action (a) reflect the current reward ® plus the maximum discounted (γ) future reward expected for the next state(s') and action (a') (see below).

   $$Q(s,a) = r + \gamma(\max(Q(s',a')))$$

   The learning system/agent will choose/suggest the what action to take based on the current Q score.

5. Give an example of a situation where regularization is necessary for learning a good model. How about one where regularization doesn't make sense?

   Answer:

   The main idea of using the regularization is to penalize complex models. The more complex models will have a greater penalty associated with them.

   In linear regression: $\omega^* = argmin_\omega \sum_{i=1}^{m}(y_i - \omega^T x_i)^2 + \lambda R(\omega)$

   In logistic regression: $\omega^* = argmax_\omega \sum_{i=1}^{m} logP(y_i|x_i, \omega) - \lambda R(\omega)$
   Here, $R(\omega) = \sum_{j=1}^{m}|\omega_j|^q$

   When q = 1, we have L1 Regularizer or LASSO.

   When q = 2, we have L2 regularizer, Ridge or Tikhonov.

   L2 regularizer is differentiable for every value of ω and more smooth, that's why this is the most popular regularization technique.

   Regularization is wildly used especially when the data is high-dimensional. Regularization can be applied in many algorithms such as convolutional neural network, support vector machine, Linear regression, Logistic regression and k nearest neighbor classifier.

   Suppose the data has more features than the number of instances, the $X^T X$ matrix is non-invertible/singular. In this case, adding a non-zero λI regularization can make it invertable.

   However, regularization can fail when the model errors are correlated or the number of features equals the number of instances.

6. Your neighborhood grocery store would like to give targeted coupons to its customers, ones that are likely to be useful to them. Given that you can access the purchase history of each customer and catalog of store items, how would you design a system that suggests which coupons they should be given? Can you measure how well the system is performing?

   Answer:

   To better target the right customers, a customer segmentation should perform according to the purchase history of each customer. Since the aim is to send the right coupons to the right customers and maximize the return on investment ROI, the relevant dimensions for such a segmentation can include customer needs, channel preferences, interests in specific product features, and customer values. For example, customers who used to purchase with coupons could be potentially a good target customer. Customers who purchase baby stuff regularly could potentially like a coupon for baby stuff. The customer segmentation helps identify a list of top customer segments.

   When implement a process/system, we can build predictive models to calculate the expected ROI for each process and pick the one with the highest expected ROI. To measure the performance of the system, we can use A/B test to test how well the systems is performing.

7. If you were hired for your machine learning position starting today, how do you see your role evolving over the next year? What are your long-term career goals, and how does this position help you achieve them?

   Answer:

   For the coming next year, I will spend my time on understanding the business economics, user motivation and related contextual information, learning the related domain knowledge and identifying what process could be improved based on current business process.

   My long-term career goal is become an expert who can deliver complex data-driving solutions with our team. For example, developing new methods for modeling end-user behavior with data-driven solutions. Redesign more efficient algorithms to improve the application performance. I hope the products we delivered will ensure your business on the right track and reshape your business to be more competitive.

   My experience in R, python, machine-learning as well as SQL and NoSQL databases fits for the successful candidate requirements which should help me quickly adapt the new environment. This position provides a platform where I can apply my knowledge in envisioning, designing, coding, testing and improving algorithms which are central to your mission.