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?

   20/100 = 20% conversion rate for page A
   15/70 = 21.4% conversion rate for page B

   Although at first glance, 5 more people clicked on the specific button in page A, when AB testing it is more crucial to look at the conversion rate. The calculations above show that page B had the higher conversion rate because out of the 70 visitors, 21.4% clicked on the button whereas page A only saw a 20% conversion rate.

   P1 = .2
   P2 = .214
   P average = 35/170 = .206

   Z = (.2 - .214)/sqrt(.206*(1-.206)(1/100 + 1/70)) = -.222
   P value = .824

   However, taking a closer look with a z-test, the p-value is much larger than .05, so this is not a statistically significant result. Therefore, I cannot confidently say that page B is the better choice.

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.

   1. Data transformation: In order to have a standardized dataset that can be more easily analyzed, the data needs to be cleaned up.
      a. Remove user_id from the feature set as it is a non-feature attribute
      b. Remove stop words to prevent matches based on common words
      c. Remove casing so that the same words are matched regardless of casing
      d. Perform lemmatization so that different forms of the same words can still match
      e. Capture the hashtags that are used in the tweet
      f. Capture the non-hashtag words used in the tweet
   2. Perform TF-IDF on data to vectorize tweets
      a. This will allow the tweets to be more easily analyzed as they aren't just one long string but a mathematical representation of word frequency and uniqueness
   3. Fit KMeans clustering algorithm onto data using K=5 (can be adjusted through inspection of results in subsequent trials)

4. Visualize tweets by showing a diagram of their most frequent words along with the cluster they were placed in (clusters can possibly be indicated by color)

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.

   To prevent overfitting, the following can be used:
   - Tweak parameters to control overfitting (e.g. for Random Forest, limit the max depth of trees)
   - Increase training data to have a larger spectrum of cases to train your model on
   - Try multiple models to confirm similar results in different classifiers
   - Cross validation to make sure that an independent dataset tested on the trained model yields expected results and that the trained model isn't fitted to only the training data

   Overfitting can be detected by using cross validation. If the test data performs poorly on the trained model compared to the training data (high variance), you know there's overfitting going on.

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?

   - Collect data and build a feature set composed of the most frequent menu options that a user uses and the activities they are doing at each time they choose one. For example, if a user consistently chooses the same menu option while performing a specific action, that should be noted
     - The feature set would have two main features: the menu option selected (e.g. smooth surface) and which object is selected at the time (e.g. a surface was selected). Therefore, there needs to be a feature for all menu options and for each selectable item.
   - A Decision Tree Classifier is an excellent candidate for this multiclass classification problem.
   - When this model is trained, it can provide a prediction of the probability of each class given a user's current activity and history. Therefore, the smart menu can display the most probable menu options in the right click menu.
     - The target label would be 'Display' and there would be two options: yes or no.

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?

   Regularization prevents overfitting by adding a penalty term to the objective function to control complexity of the trained model. An example where regularization is necessary is in image classification. If the model overfits, it cannot generalize similar but different images and classify them together. This is seen when the model performs well on training data, but poorly on test data. Regularization doesn't make sense when the model's issue is underfitting, which can be easily seen if

the model shows high bias, which is when there's a lot of error in the model's scores for the training and test data, or if the model's complexity is too simple.

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?

    - Collaborative filtering or K-Nearest Neighbor should be used
        - Collaborative filtering: create a matrix for each customer of the catalog of store items and her purchase history, find customers whose matrices have sufficient overlap, and recommend coupons for products that are outside of that overlap
        - KNN: use store catalog as feature set, fit model to data, and recommend coupons according to different customers' purchase histories that are near each other
    - You can measure how well the system is performing if sales increases after implementing this coupon algorithm and/or there are new items in customers' purchase histories. If there is a new purchase in a purchase history and this also corresponds with a coupon that was sent out, you can know that this strategy is working.

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?

    Over the next year, I see my role in Machine Learning evolving from one of simply providing data that is asked for to constantly exploring the company's data and offering unforeseen insights. As a Machine Learning Engineer, I believe it is our duty to not just do what we're asked, but also to reveal unlooked-for conclusions and bring value to the company in new ways.

    Ingrooves's mission is to build the digital record label of the future. The competitive advantage of going digital is that there is exponentially more data available to offer partners greater transparency and strategy in music distribution. As a Data Scientist, I'm eager to contribute by applying cutting edge techniques to that plethora of data to discover trends, suggest recommendations, and build a more personal and effective service for the leadership and label partners.

    My goal is to be the best Data Scientist I can be and specifically an expert in Hadoop, a framework designed for distributed processing of large data sets across multiple clusters.

    This position is super helpful to achieving these goals because the job entails deriving recommendations and insights from gigabyte and terabyte sized datasets. This will allow me to constantly apply new Hadoop techniques to efficiently analyze this data and not only better myself but also bring value to the company.

    In the near future, my one year goal is to get a handle on Ingrooves offerings and services enough to mentor others in data science and lead a team to allow big data to become one of Ingrooves most marketable and effective features.