Machine Learning Interview

Question 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?

If you straight away calculate the percentages, visitors who view page B will have a 21.42% percentage of clicking the button, as opposed to 20% in page A.

However, we should test to see if the difference between the two percentages is significant. Considering B is an improvement on A, we will use a one-tailed test to test if the percentage of users clicking the button in page B is greater than the number of users clicking the button in page A. Our hypothesis test would then be:

- H0: Pb <= Pa
- HA: Pb > Pa

If we reject the null hypothesis, it means the difference is statistically significant and we can confidently state page B is a better choice than page A.

Let's use the z core test for two population proportions, given by:

$$\frac{(\overline{p}_1 - \overline{p}_2) - 0}{\sqrt{\overline{p}(1 - \overline{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

p-value of the one-tailed test equals 0.40905, meaning the result is not significant at $p < 0.05$. Hence, we fail to reject the null hypothesis, and can not infer page B is a better choice than page A based on the data collected.

Another relevant factor is to isolate and identify independent variables. A simple example is let's say day of the week. People will be more willing to pay if they have just received their salary. So if test for page A is conducted on Thursday, and for page on Friday, that may explain the variances. To make sure these external factors don't impact the test, we need to define the independent variables ahead and control for them.

Question 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:**

{ "userid": 3, "timestamp": "2016-03-2211-31-20", "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.

We want to group Twitter users. The first step is to create the features that we will use to cluster our users. We could group them by location, usage pattern, interests, or even how long they have been a Twitter user. All of these features can be extracted from these variables shown.

- Location: We can infer location by analyzing the timestamp. We can guess the sleep time based on the timestamp of several tweets. If you place them across a timeline, a time window will appear, which will be more or less equivalent to the time the user sleeps. We can also correlate keywords with timestamps, such as the word dinner posted ad 11pm PST, to improve the proxy for location even further
- Usage Pattern: In my view that would be one of the most important groups to define in order to create customized offers. The usage patterns can be based on how many tweets per day an user posts, if he/she posts consistently every day (like a media person) or at random, the average length of the tweets, whether the content are links or original content.
- Active user for how long: Based on user_id, considering is sequential.
- Interests: the content can be extracted from the tweet text by using a strategy such as bag of words or more complex natural language processing techniques, and classified into groups, such as Food or Sports. Before analysis the tweet text needs to preprocessed - that is usually done by removing words that carry little meaning, such as # and prepositions, and applying stemming and/or lemmatization, which reduce the words to a common base.

After creating the features, the second step is using a non-supervised learning algorithm for clustering, such as k-Means or gaussian mixture models. We can optimize the algorithm in order to have the best separation between groups or to return a specific number of clusters pre-defined (say we want to divide our customers into 10 groups). The importance of each feature can be weighted before clusterization if one of the features is deemed more important by the domain expert.

Question 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.

Overfitting will occur when the model is adjusted to fit the training data, and it won't generalize to new data. There are several strategies to prevent overfitting. One of the most important is cross validation.

Before training a machine learning model, you need to divide your data into training and testing data, to accurately access the performance of your algorithm. When optimizing the model parameters, a common mistake is to use the test data to test different models and evaluate and choose the best performing one.

There is major data leakage in this case, since test data information is being used to choose the best model. In order to fix that, you can add another subset of the data, a validation set, and use it to optimize parameters. But even though you run the risk of overfitting to the validation set.

The crossvalidation techniques aims to avoid this issue. The general idea is: separate X% of the dataset as a validation set, say 10%, and train the model on the remaining data. Repeat this process several times taking different subsets as a validation set, and calculate the performance as the average performance between all trials. This will prevent overfitting to a fixed validation or testing dataset.

Question 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?

I would like for patterns in usage behavior that would indicate it is a common function. The importance of unsupervised learning algorithms is often understand, and this is a situation it would fit perfectly.

Detecting usage patterns would be to cluster similar sequence of commands in order to undertand which commands are used together. In this case, we can use a Gaussian Mixture Model.

Gaussian mixture models implements expectation-maximization algorithm and will give a probability of each command beloging to each cluster. That will allow us to prioritize the groups by number of members in the cluster weighted by their probability of belonging to that cluster. After prioritization we can create a menu option for the most important behaviors identified.

Question 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 is an important step to prevent overfitting. Regularization ensures no particular feature has a proeminent impact in the learning task in a way that prevents the convergence to an optimal solution. In a way, it ensures all features are within expected boundaries required for an algorithm to converge.

Regularization is useful for several algorithms that requires optimization algorithms to find a global maxima or minima, such as gradient descent. Linear Regression, Logistic Regression, SVMs and Neural Networks are algorithsm that require some sort of regularization ir order to ensure convergence.

Regularization would be less useful in cases when tge features have been previously normalized to a pre-defined range such as 0 to 1.

Question 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?

First, the data can be extracted from the database using simple select queries. Given access to the purchase history of each customer, we can infer which products is each customer most likely to buy. In order to do that, we can calculate the number of times a customer buys a specific item divided by the total number of times the customer went to shopping. For example, if a customer bought olive oil 9 out of 10 times he went shopping, the product will have a score of 0.9.

This heuristics will give a score per item per customer, which can be used in a recommendation system, an unsupervised learning technique. The recommendation algorithm will look at other customer with similar shopping patterns, and handle a suggestion to the user based on what other look alike customers are buying.

Another suggestion is to cluster the customers into segments, using an unsupervised learning algorithm such as k-Means or gaussian mixture model, and create a customized offer for each cluster based on the groups preference. To clusterize we can use the same product preference score, described above. That would be an easier idea to implement for a business since it customizes the offer by group instead of individuals.

The algorithm can be continuously improved with an online reinforcement learning algorithm. We can measure how well the system is performing by assessing whether or not the customer used the cupons, both in absolute terms and in comparison to other customers, and use this information to improve our view on the customers preference (the performance would be the reward for a reinforcement learning algorithm).

Question 7 - Pick a company of your choice and briefly describe a hypothetical Machine Learning Engineer role at that company you would like to apply for.

Now, if you were hired for that 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?

My long term career goals is to get a deeper understanding of the learning process, both human and machine. My interests are on artificial intelligence in general and applied, including machine learning, and cognitive computation.

In order to do that, my desire is to work on cutting edge problems, that requires deepening and research to design innovative solutions to a complex problem.

In the long term as an employee, I would like to carefully review the company offers compared to the industry offers and see in which ways I could apply my skills to innovate further and help to position the company above its competitors, while also fostering innovation and contributing to the advancement of science.

JOB DETAILS (profile from an recruitment agency, hiring company not specified)

- Use of cutting edge data mining, machine learning techniques for building advanced customer solutions.
- Use techniques from artificial intelligence/machine learning to solve supervised and unsupervised learning problems.
- Design solutions for complex business problems related to BIG Data by using NLP/Machine Learning/Text Mining techniques.
- Recommend and implement best practices around application of statistical modelling.
- Develop and implement solutions to fit business problems which may include applying algorithms from a standard statistical tool or custom algorithm development.
- Candidate should have working experience in analytics(predictive modelling, logistic regression etc

Skills Required: Big Data, Statistical Data Analysis, Machine Learning

Skills: Big Data - 50%, Machine Learning - 50%