Quiz Debrief

Here is a debrief of the quiz answers:

Question 1:

You are responsible for a recommender system for an e-commerce site that has three slots in which to recommend products at check-out time. In general, you are most interested in whether people buy additional products based on the recommendations, and you will be measuring sales and lift when your system is live, but you want an offline measure to help determine which potential recommenders are worth trying online. A few other details: this is a domain where people do occasionally re-purchase items, but not frequently. Most regular customers visit between once a week and once every three months. The site has complete details on all previous purchases, and has customer ratings for about 5% of the purchased items. Also, the site does a pretty good job recommending to new customers based on demographics and overall popularity, so you are focused on finding recommenders that take advantage of information learned from customers who have already purchased several items. Which of the following evaluation plans/metrics seems best? (Select all that apply)

Possible Answers:

RMSE-based accuracy evaluation. In this plan, we will take any customers who have at least 10 different prior ratings and measure the accuracy of predictions from the recommender using the leave-one-out method. We'll average over all predicted ratings, and will identify the best few algorithm candidates based on lowest RMSE.

Top-n precision evaluation. In this plan we'll use all customers who have at least 10 product purchases, and measure the top-3 precision of recommendations using a 5-fold cross-validation against a random 80% training/20% test set. We measure as a "hit" anything withheld from the test set. We will identify the best few algorithm candidates based on highest top-3 precision.

Spearman rank correlation evaluation. In this plan, we'll use all customers who have at least 10 product purchases, compute a recommendation list (using and 80/20 random training/test data set and five-fold cross validation), and measure the Spearman correlation between the between the withheld items and the recommedation list. We will identify the best few algorithms based on the highest Spearman correlation.

New user bootstrapping evaluation. In this plan, we evaluate the success of our top recommendation for new users (with no ratings), then our top recommendation for new users with one rating (their first), and so forth, with the goal of seeing how well we can recommend a single item to new users at different stages.

Answer:

(b) is the best answer. Top-3 precision most directly measures what you're interested in (the chance that the user is interested in one or more of the 3 items you present to her).

(a) and (c) are poor metric choices here. RMSE over all possible predictions doesn’t directly address whether the top-3 will be good or bad. This is not a place for accuracy, but for decision support or rank metrics. The RMSE plan also throws out all the purchase data (95% of the data). Spearman is a rank metric, but it penalizes errors equally at the bottom of the list and at the top. Also, there isn’t an obvious way to rank the 95% of data that is simply yes/no purchase data.

(d) completely ignores the fact that we're trying to focus on users that already have several items.

Question 2:

Which of the following is a situation in which Mean Absolute Error is a reasonable choice as a metric for evaluating recommender performance?

Possible Answers:

You are running a streaming music site with a recommender inside that selects music to provide listeners with a personalized music listening experience (somewhat like Pandora). The basic interface is very small, it shows the name of the song and artist, and has controls that allow the user to click on a star (for favorite) or on an X (for don’t play this again). It also allows users to click a forward arrow (skip to next song). Your goal is to have users enjoy the music enough to continue listening to the site (which is paid for by advertisements, which the user cannot skip).

You are running an online news site that presents users with an on-screen newspaper (somewhat like Google news). The site places articles into categories, the first of which is “top stories for you,” and the others are traditional news categories (local news, world news, sports, business, etc.). The contents of each category are selected by the recommender, as is the order of presentation of items. Each items is displayed as a headline, and the first story in each category also has a few sentences from the start of the story. User feedback is entirely implicit -- users either read the full article or don’t. Your goal is to have high site usage, as the site also contains advertisements.

You are running a travel-related recommendation site where users can look for hotels and restaurants (somewhat like TripAdvisor). For a given hotel or restaurant, you have a collection of data, including user ratings (lots of them), tags, written reviews and objective attributes (e.g., swimming pool, 24 hour restaurant). When displaying an item, the system shows a predicted rating (in this case personalized according to the user’s profile, unlike TripAdvisor’s average). Some usage is through search for specific places (e.g., Holiday Inn Hong Kong Kowloon), and search for category/location (e.g., Hotel swimming pool Paris). But about 80% of accesses go directly to a particular hotel or restaurant’s page driven from search engines such as Google and Bing. The recommender is used both to produce the predicted ratings and as part of producing a ranked list of results for internal searches (where it is merged together with a search algorithm that measures quality of match to the search terms). Your site benefits both from visitors (through ads) and from booking referral payments.

You are running an e-commerce site for a shoe store (somewhat like Zappos). Your site has a large set of shoes, with structured product data for each (colors, materials, sizes, etc.). It also has user purchase data and user rating data. While some customers mostly visit the site to re-purchase shoes they already own that have worn out, most of the profit comes from people with large shoe collections who regularly buy new pairs of shoes. Your responsibility is a recommender system that sends out a periodic e-mail to the store's best customers suggesting items for purchase and offering a premium if the user makes a purchase within a certain time period. A typical e-mail would have pictures and descriptions of four pairs of shoes, along with a promotion such as a free travel shoe bag if placing an order for more than $100 before November 15th. Your site is a pure commerce site -- your revenue comes from sales. The goal for the e-mail program is to generate sales, though it is not considered important whether the customers buy the recommended items, or end up choosing to buy other items. All that matters is how much they buy.

Answer:

(c) is the best answer. The site displays actual predictions, so knowing how accurate those predictions are can be particularly relevent. Significantly higher errors will undermine user confidence.

(a) does not display predictions, and it doesn't provide a scale on which users would think about predictions. It is about choosing whether (and perhaps how often) to play a song.

(b) is primarily about top-n (and particularly top-1) success -- rank matters but accuracy does not.

(d) is a case where MAE may be reasonable, but it isn't the best fit. Given that there are explicit ratings, MAE could be a useful indicator of how well the system knows user preferences, but in the end, (d) is more about some combination of diversity, novelty, and top-n precision (the goal is to recommend a set of items where one of them is interesting enough to motivate the user to visit the site).

Question 3:

Which of the following is a situation in which it would be most useful to tune the recommender using a metric such as the receiver operating characteristic -- specifically, tuning the algorithm to find the right trade-off between true positive and false positive rates?

Possible Answers:

You are running a streaming music site with a recommender inside that selects music to provide listeners with a personalized music listening experience (somewhat like Pandora). The basic interface is very small, it shows the name of the song and artist, and has controls that allow the user to click on a star (for favorite) or on an X (for don't play this again). It also allows users to click a forward arrow (skip to next song). Your goal is to have users enjoy the music enough to continue listening to the site (which is paid for by advertisements, which the user cannot skip).

You are running an online news site that presents users with an on-screen newspaper (somewhat like Google news). The site places articles into categories, the first of which is "top stories for you," and the others are traditional news categories (local news, world news, sports, business, etc.). The contents of each category are selected by the recommender, as is the order of presentation of items. Each items is displayed as a headline, and the first story in each category also has a few sentences from the start of the story. User feedback is entirely implicit -- users either read the full article or don't. Your goal is to have high site usage, as the site also contains advertisements.

You are running a travel-related recommendation site where users can look for hotels and restaurants (somewhat like TripAdvisor). For a given hotel or restaurant, you have a collection of data, including user ratings (lots of them), tags, written reviews and objective attributes (e.g., swimming pool, 24 hour restaurant). When displaying an item, the system shows a predicted rating (in this case personalized according to the user's profile, unlike TripAdvisor's average). Some usage is through search for specific places (e.g., Holiday Inn Hong Kong Kowloon), and search for category/location (e.g., Hotel swimming pool Paris). But about 80% of accesses go directly to a particular hotel or restaurant's page driven from search engines such as Google and Bing. The recommender is used both to produce the predicted ratings and as part of producing a ranked list of results for internal searches (where it is merged together with a search algorithm that measures quality of match to the search terms). Your site benefits both from visitors (through ads) and from booking referral payments.

You are running an e-commerce site for a shoe store (somewhat like Zappos). Your site has a large set of shoes, with structured product data for each (colors, materials, sizes, etc.). It also has user purchase data and user rating data. While some customers mostly visit the site to re-purchase shoes they already own that have worn out, most of the profit comes from people the large shoe collections who regularly buy new pairs of shoes. Your responsibility is a recommender system that sends out a periodic e-mail to the store's best customers suggesting items for purchase and offering a premium if the user makes a purchase within a certain time period. A typical e-mail would have pictures and descriptions of four pairs of shoes, along with a promotion such as a free travel shoe bag if placing an order for more than $100 before November 15th. Your site is a pure commerce site -- your revenue comes from sales. The goal for the e-mail program is to generate sales, though it is not considered important whether the customers buy the recommended items, or end up choosing to buy other items. All that matters is how much they buy.

Answer:

(a) is the best answer. The quality of the music stream is based on playing as many songs the user likes as possible, but as few disliked songs as possible as well. Plotting an ROC curve would allow tuning the recommender to the "knee" where the trade-off is best.

(b) is an application where ROC could help, but given the limits of the number of articles shown, it is less important to focus on finding the right trade-off and more important to focus on precision in the top n. Also, unlike the music case, there is much greater importance in a news site on the specific lead stories for each section.

(c) and (d) are both examples where this type of optimization isn't particularly useful. In (c) the focus is on rank and on predictions -- there's no filtering good from bad. In (d) it is conceivable that one could try to recommend all desirable products over time, but this would assume a fairly small set of desirable products.

Question 4:

All of these situations are ones where it would make sense to test different recommenders empirically through A/B tests or other field tests. In which situation is it LEAST LIKELY that you could get useful data by asking users which set of outputs they prefer? In other words, in which situation are users least likely to know whether the recommender is actually achieving its goals?

Possible Answers

You are running a streaming music site with a recommender inside that selects music to provide listeners with a personalized music listening experience (somewhat like Pandora). The basic interface is very small, it shows the name of the song and artist, and has controls that allow the user to click on a star (for favorite) or on an X (for don't play this again). It also allows users to click a forward arrow (skip to next song). Your goal is to have users enjoy the music enough to continue listening to the site (which is paid for by advertisements, which the user cannot skip).

You are running an online news site that presents users with an on-screen newspaper (somewhat like Google news). The site places articles into categories, the first of which is "top stories for you," and the others are traditional news categories (local news, world news, sports, business, etc.). The contents of each category are selected by the recommender, as is the order of presentation of items. Each items is displayed as a headline, and the first story in each category also has a few sentences from the start of the story. User feedback is entirely implicit -- users either read the full article or don't. Your goal is to have high site usage, as the site also contains advertisements.

You are running a travel-related recommendation site where users can look for hotels and restaurants (somewhat like TripAdvisor). For a given hotel or restaurant, you have a collection of data, including user ratings (lots of them), tags, written reviews and objective attributes (e.g., swimming pool, 24 hour restaurant). When displaying an item, the system shows a predicted rating (in this case personalized according to the user's profile, unlike TripAdvisor's average). Some usage is through search for specific places (e.g., Holiday Inn Hong Kong Kowloon), and search for category/location (e.g., Hotel swimming pool Paris). But about 80% of accesses go directly to a particular hotel or restaurant's page driven from search engines such as Google and Bing. The recommender is used both to produce the predicted ratings and as part of producing a ranked list of results for internal searches (where it is merged together with a search algorithm that measures quality of match to the search terms). Your site benefits both from visitors (through ads) and from booking referral payments.

You are running an e-commerce site for a shoe store (somewhat like Zappos). Your site has a large set of shoes, with structured product data for each (colors, materials, sizes, etc.). It also has user purchase data and user rating data. While some customers mostly visit the site to re-purchase shoes they already own that have worn out, most of the profit comes from people the large shoe collections who regularly buy new pairs of shoes. Your responsibility is a recommender system that sends out a periodic e-mail to the store's best customers suggesting items for purchase and offering a premium if the user makes a purchase within a certain time period. A typical e-mail would have pictures and descriptions of four pairs of shoes, along with a promotion such as a free travel shoe bag if placing an order for more than $100 before November 15th. Your site is a pure commerce site -- your revenue comes from sales. The goal for the e-mail program is to generate sales, though it is not considered important whether the customers buy the recommended items, or end up choosing to buy other items. All that matters is how much they buy.

Answer:

(d) is the best answer. Because the end goal isn't specifically to get the user to evaluate or buy the recommended products, but simply to induce any purchasing, users are least likely to be aware of whether they would work. Indeed, recommending ugly shoes might lead the customer to feel insecure and to shop for new better-looking shoes.

(c) is an area where users might also have trouble assessing -- particularly the part of the recommender that is used for ranking. The challenge here is that the ranking is a mix of relevance and predicted desirability, which is harder to assess.

(a) and (b) are examples where a user could likely compare alternatives and give good feedback on which songs or news they like best.

Question 5:

We have argued that the real proof of a recommender system is in the usage, and that offline evaluation has serious problems. At the same time, there are many situations where offline evaluation does make sense. Which of these is a valid reason for carrying out off-line metric-based evaluation rather than a live user study of a recommender?

Possible Answers:

The recommender may not yet exist yet, but there is data that can be used to pre-test the idea of the recommender.

The recommender designer wants to test a wide variety of alternative recommenders, at least to narrow them down to a few candidates that can be user tested.

Both of the above

Neither of the above

Answer:

(c) is the correct answer. Both of these reasons are valid reasons for doing offline metric-based evaluation.

Question 6:

Which of the following is a valid objection to the validity of offline evaluation using metrics such as MAE, top-n Precision, or nDCG?

Possible Answers:

The offline metrics only assess the ability to “recommend” items that have already been consumed or rated. Real recommenders should usually be suggesting new items not already known to the user. Hence, something with low offline metrics might acually be better at finding new items of interest.

The offline metrics depend completely on the scale of the data being presented. For instance if you change a rating scale from 5 points to 100 points, the MAE will increase by a factor of 20.

Offline metrics come in different groups. Some measure accuracy. Some address decision correctness. Some look at rank. That means that we can’t tell if any metric is relevant for any particular recommender system.

All of the above objections are valid.

Answer:

(a) is the correct answer. Particularly in a situation where it is reasonable to believe that users may not have found the “best” items for them (e.g., a user who has consumed 30 out of 3 million books), we run into the risk that a recommender that’s “smart enough” to recommend unconsumed items that are better than the already-consumed ones will be penalized by the offline metrics.

(b) is true, but it isn’t a reason to avoid offline metrics. They can be normalized (and some are normalized to begin with), and in any case metrics rarely compare across different data sets (offline or online).

(c) is partly true -- there are different groups of metrics, but it is possible to determine which are most applicable by considering the properties of the recommender problem being studied.