Online Experimentation

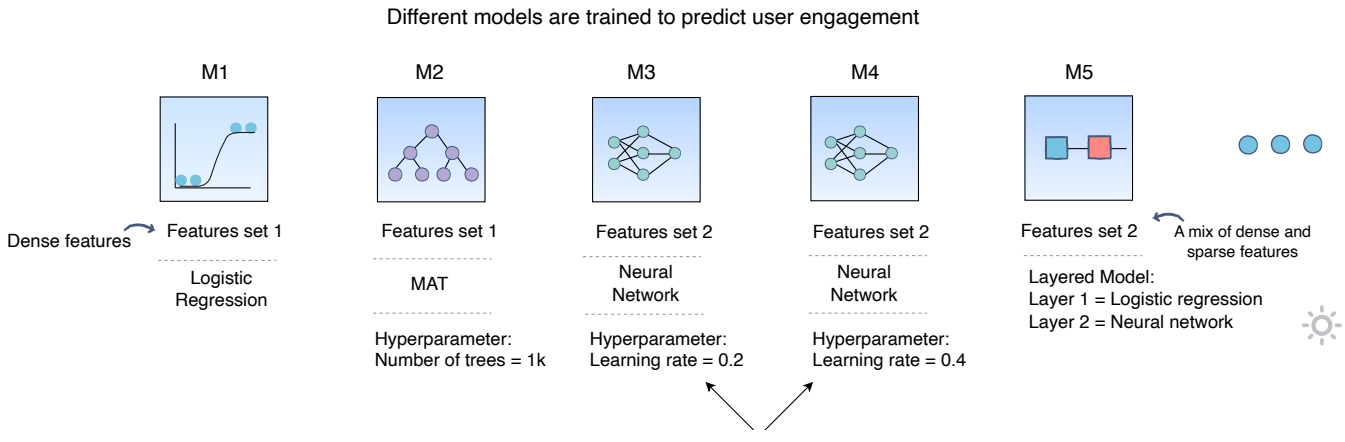
Let's see how to evaluate the model's performance through online experimentation.

We'll cover the following

Let’s look at the steps from training the model to deploying it.

Step 1: Training different models#

Earlier, in the training data generation lesson, we discussed a method of splitting the training data for training and validation purposes. After the split, the training data is utilized to train, say, **fifteen** different models, each with a different combination of hyperparameters, features, and machine learning algorithms.



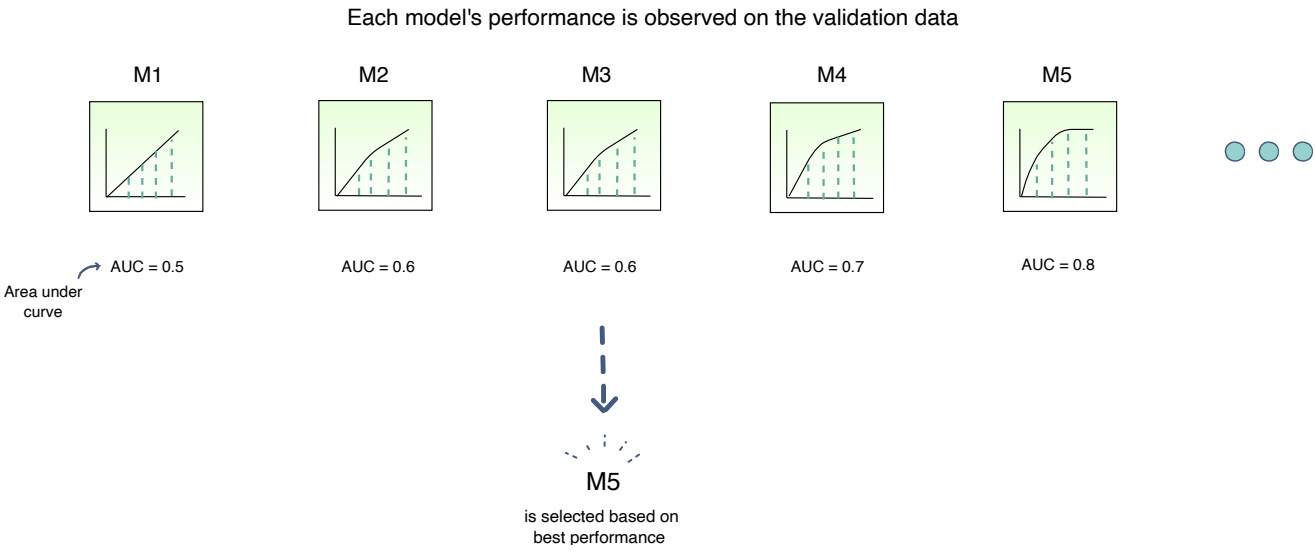
Trying out different hyperparameters for the same feature set and model

Different models are trained to predict user engagement

The above diagram shows different models that you can train for our tweet engagement prediction problem. Several combinations of feature sets, modeling options, and hyperparameters are tried.

Step 2: Validating models offline#

Once these **fifteen** models have been trained, you will use the validation data to select the best model offline. The use of unseen validation data will serve as a sanity check for these models. It will allow us to see if these models can generalise well on unseen data.



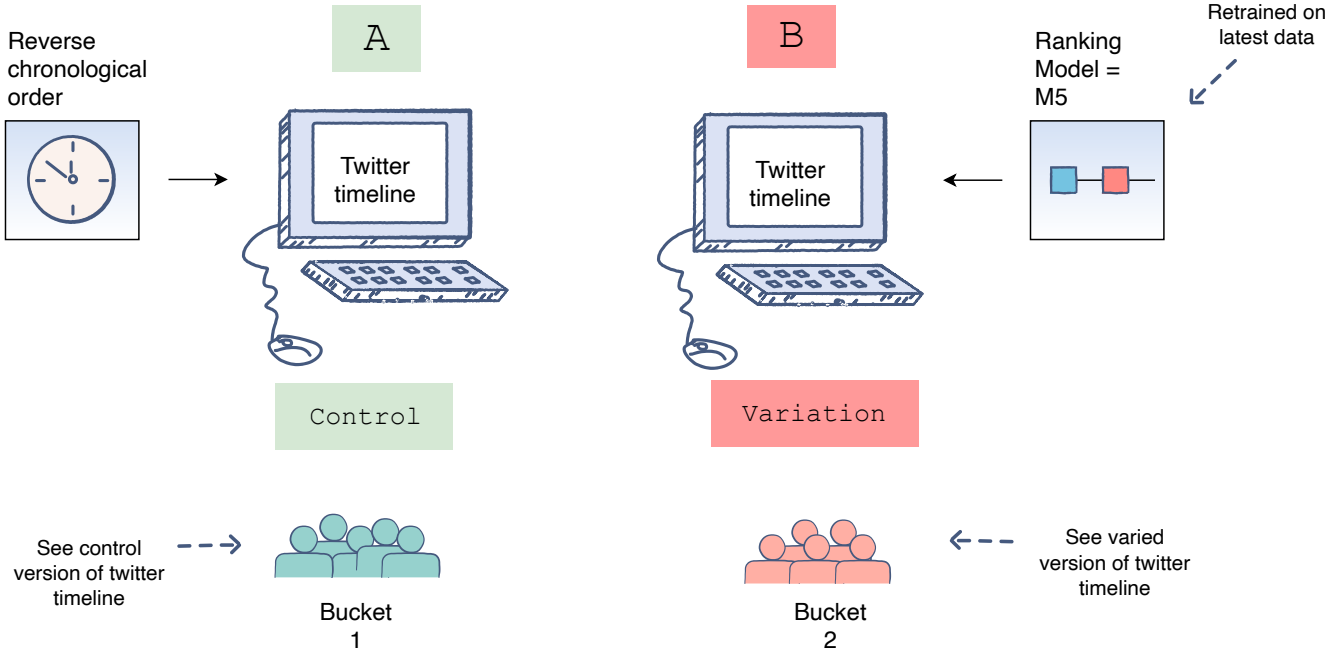
Each model's performance is observed on the validation data

Step 3: Online experimentation#

Now that you have selected the best model offline, you will use A/B testing to compare the performance of this model with the currently deployed model, which displays the feed in reverse chronological order. You will select 1% of the five-hundred million active users, i.e., five million users for the A/B test.

Two buckets of these users will be created each having 2.5 million users.

Bucket one users will be shown twitter timelines according to the time-based model; this will be the control group. Bucket two users will be shown the Twitter timeline according to the new ranking model.



Bucket one users see the control version, whereas Bucket two users see the varied version of the Twitter timeline

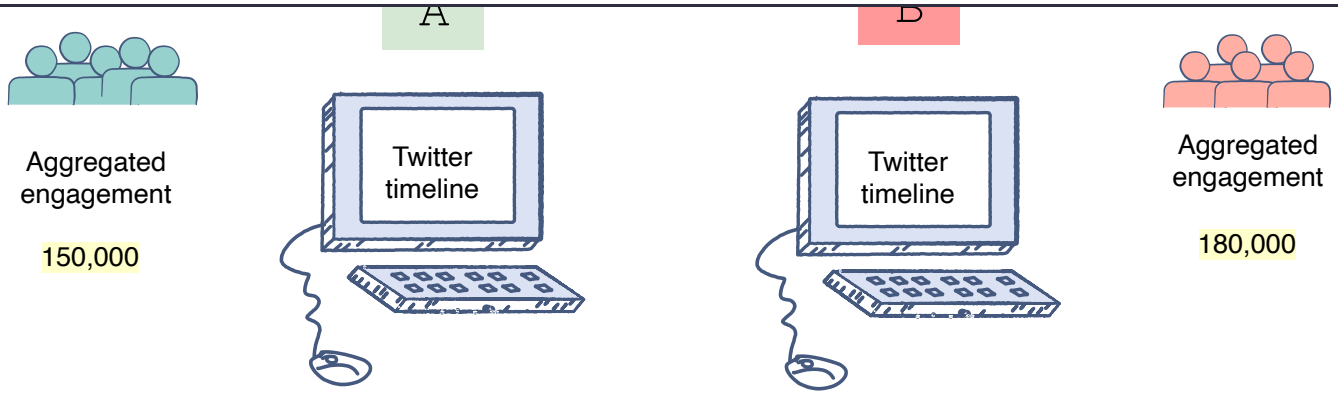
However, before you perform this A/B test, you need to retrain the ranking model.

Recall that you withheld the most recent partition of the training data to use for validation and testing. This was done to check if the model would be able to predict future engagements on tweets given the historical data. However, now that you have performed the validation and testing, you need to retrain the model using the recent partitions of training data so that it *captures the most recent phenomena*.

Step 4: To deploy or not to deploy#

The results of the A/B tests will help decide whether you should deploy the

new ranking model across the platform.



Engagement aggregates for both buckets of users

You can observe that the Twitter feeds generated by the new ranking model had thirty (180k-150k) more engagements.

Increase in engagement (*gain*) = $\frac{180{,}000-150{,}000}{150{,}000} * 100 = 20\%$

This model is clearly able to outperform the current production, or live state. You should use statistical significance (like p-value) to ensure that the gain is real.

> Learn more about online experimentation in this lesson.

Another aspect to consider when deciding to launch the model on production, especially for *smaller gains*, is the increase in complexity. If the new model increases the complexity of the system significantly without any significant gains, you should not deploy it.

> You should go for complex solutions (based on new features, or data, etc.) only if you anticipate it to bring larger gains in the future.

To wrap up, if, after an A/B experiment, you see an engagement gain by the

model that is statistically significant and worth the complexi d h

system, it makes sense to replace the current live system with the new model.

Back

Diversity

Next

Problem Statement

Mark as Completed

Report an Issue