Hi Fred and Jenna,

We would like to have you updated early this week.

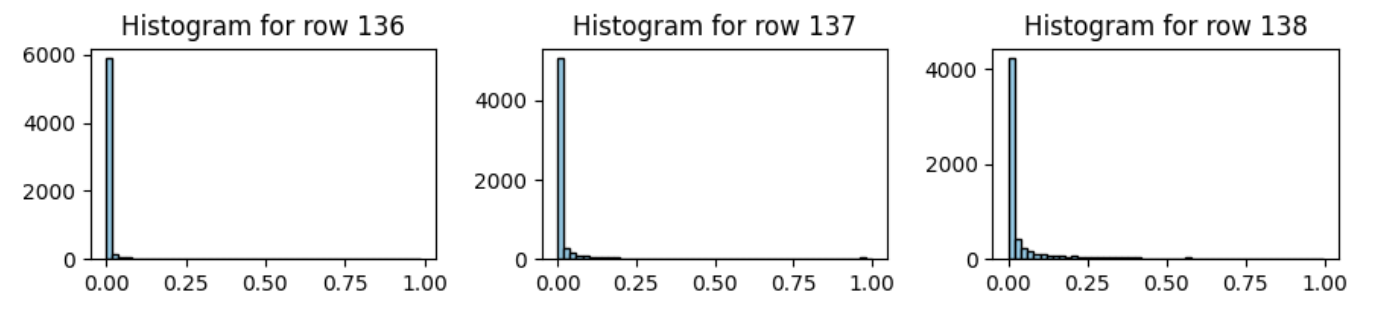
1. **GPT4**

We have experimented with different prompts to GPT 4.0.

We chose 150 (representative enough) articles in the *yahoo\_pairing* dataset and fed it to GPT4, the analysis is as followed:

* 1. Question\_head = "*Do you think these (Article Title, Video Title) pairs are good matches in news media? Try to focus on higher level information such as Subjects, Verticals, Domains, etc.*"
  2. After comparing GPT4 scores with human labels, the best result is: ***Acc=0.853, Precision=0.95, F1=0.838***. This means, if we directly ask GPT4 to score based **only on titles**, it is doing a **moderately good job** in **scoring good/bad**.
  3. Suppose GPT4 is a good tool, then the business rationale of training another NN to mimic GPT results is: Training our own network is **cheap and feasible** in production, while calling GPT4 API to match for each (article, video) pair would be highly expensive and low in efficiency.

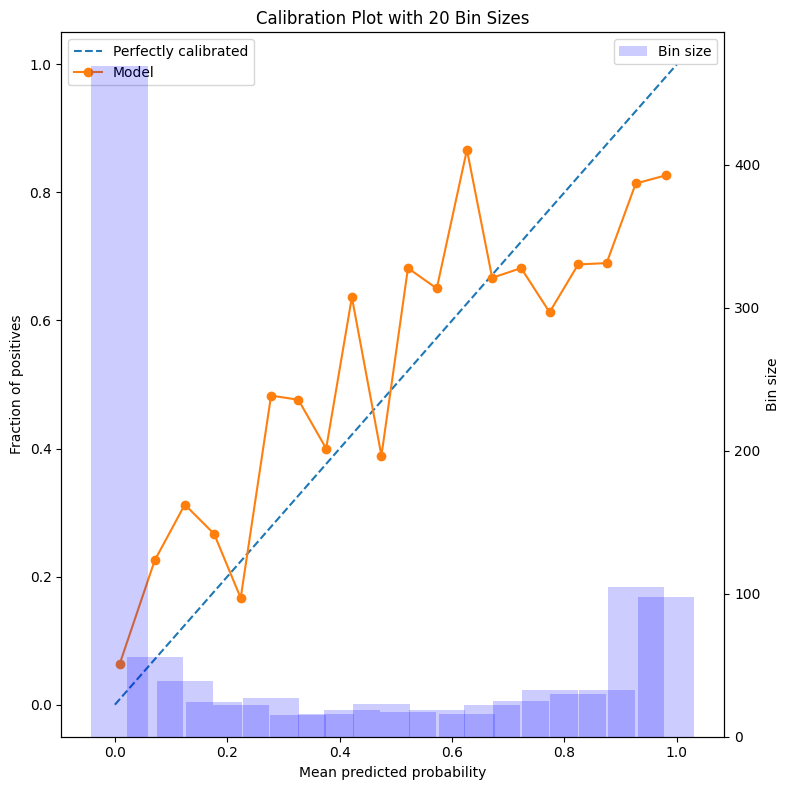
1. **Validation**
   1. We train-val-test-splitted the dataset by 8:1:1 with “good” as “match”(1) and “bad”/“offensive” as “unmatch”(0). We trained the NN on the train set for a binary classification problem of match vs. unmatch using Cross Entropy loss.
   2. The validation set is around the size of 550. We selected the model which performs the best F1-score on the validation set as our best model.
   3. Then we selected 150 articles from our validation set. For each article, the top-1 video would be the one with the highest probability (out of 6.3k videos) to be a match from our model.
   4. The model **does well in distinguishing match/unmatch** using the 5.5k *yahoo\_pairing* dataset. i.e. our model scoring of all 6.3k videos for a given article follows a good distribution for each article, an exponential distribution of raw probability and normal distribution of log probability.
   5. The current issue identified is: The dataset of size 5.5k, in nature, is NOT a good one for top1-recommendation tasks. From our DS experience, a robust recommendation system requires thousands or even millions of user data (clicks, view, interaction, etc.) with records like ***(article, good matches, bad matches)*** or records like***(user, articles\_read, videos\_watched)****.* That is to say, user behaviors may also make a great difference when deciding the good/bad match.



1. **Brier Score**

As you suggested last week, we evaluated our binary classification model with Brier score. Note our model is aimed at differentiating between good and bad matches on the 5k5 training samples. On the combination of validation and test set, the results show that our model achieves a Brier score of **0.153**, with the values of the 3 components respectively as: ***Reliability=0.015, Resolution=0.1, Uncertainty=0.237***.

Below shows the calibration plot of our mode with 20 bins, where the x-axis represents the probability of being a good match, and the left y-axis represents the actual percentage of good matches in each bin, the right y-axis represents the size of each bin, from which you can see a large number of negative samples are concentrated in the bin of probability of 0, but the positive samples are distributed more uniformly.



Looking forward to hearing your feedback in detail this Wednesday!