Model enhancement

Adding Data sources and user behaviour features:

MVP Features:

1. f_cat_propensity_sim - min max mean cosine/ jaccard similarity of examined leaf/lv2/meta to all leafs /lv2/meta categories candidates of purchase, views in last x days.
2. Time_since_last_action_from_cat_in_days - time in days since last purchase/view from examined leaf/lv2/meta
3. f_num_actions_from_cat_in_last_x_days- num actions of purchases /views from cat in last x days

User behaviour features:

More details:

PS_Model_enhancement

PS_Github_features

1. Aggregation of Time Differences Between Purchases and Views (Leaf/L2/Meta Categories)

1.aggregation (min/max/mean) over time between purchase from category type L (leaf/L2/meta)and view from that same category :

- Per user and leaf/l2/meta: Per all user purchases add first view since the purchase : calc diff in days. And return min, max, mean of all days diff.

- This set of features calculates the difference in days between a user's purchase from a category (leaf, L2, or meta) and their first view of an item from the same category after the purchase.
- Aggregation statistics (min, max, mean) are computed for these time differences for each user across all their purchases.

2.Aggregation of Number of Views Between Purchases (Leaf/L2/Meta Categories)

- Track the number of views a user has between consecutive purchases from the same category (leaf/L2/meta). For each user and category, we calculate the minimum, maximum, and mean number of views between purchases.
- Additionally, the number of views is divided by the time period between purchases, capturing the frequency of views.
- aggregation of (min/max/mean) over number of views from category type L (leaf/L2/meta) and the previous purchase from that same category

  Per user and leaf/l2/meta: Per purchases add views from first to last view since the purchase,(view before next purchase) and count num views he had, for all these counts calc min,max, mean.

3.Per user and category: num views since last purchase

- This feature calculates the total number of views a user has made in a specific category since their last purchase. It helps measure the level of ongoing engagement (via views) in a category after a purchase, indicating whether the user continues to interact with the category.

4.Per user and category: days from last view to last purchase.

5. Category Diversity (Entropy) of User Activity

Feature Definition:

- This feature measures the diversity of a user's purchases or views across different categories (leaf/L2/meta). Entropy is used to quantify how concentrated or dispersed a user's activity is across various categories.
- The higher the entropy, the more balanced the distribution of purchases across categories.

Entropy = 0: All purchases in one category (no diversity).

$$H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

6. Difference Between User Activity and Population Mean

Feature Definition:

- This feature compares a user's number of purchases or views within a given period to the average activity of the entire population. The difference is normalized by the population's standard deviation (z-score).

Feature Extraction:

  - For each user and time period, calculate the number of purchases or views.

Compute the mean and standard deviation of the population's activity for the same period.

$$\frac{(\text{User Mean} - \text{Population Mean})}{\text{Population Std Dev}}$$

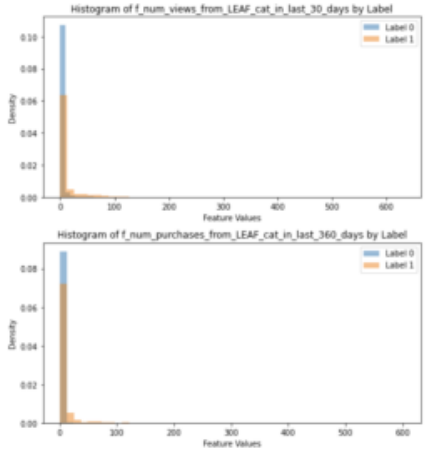
7.Normalizing views and purchases:

As top features are num views, and num purchases:

```
Feature importance of ranker:
 1) f_num_views_from_LEAF_cat_in_last_30_days 0.150610
 2) f_num_purchases_from_LEAF_cat_in_last_360_days 0.133895
```

And these features are 10x stronger in importance, and by the features distribution it is possible to see that users with more views/purchases tend to gain interest (label==1) than users with no future interest.

Top feature's distribution by label:



Idea is to normalize features of num views , and num purchases in cat in last x days by: num views / purchases across all cats in cat type of buyer in last x days,and by this better differentiate between users likely to show future interest and those who are not.

Normalizing views and purchases in this way can be beneficial. By expressing a user's engagement in a specific category relative to their total engagement across categories, it highlights their preference for that category. This normalization can improve the model's ability to identify users with a high likelihood of future interest (label 1) compared to those without interest (label 0), especially if users with a strong preference for specific categories are more likely to become repeat customers or engage further.

8.View-to-purchase ratio by category in last x days : Calculate the ratio of views to purchases for each category . Users who view more than they purchase in a particular category could have a different engagement pattern, hinting at browsing vs. buying intent.

9.Feature of weighted sum for user actions:

Weighted sum of user actions across different categories (LEAF, LVL2, and META).

- **Time Windows**: The actions are tracked over several time windows (2, 7, 30, 90, 180, and 360 days). The weights applied to each time window reflect the assumption that more recent actions are more indicative of future behavior, so they should have a higher impact on the overall feature value.
- **Weighted Calculation**: The actions within each category and time window are assigned weights that emphasize more recent activity. For example, views and purchases from the last 2 days have a higher weight (0.35) than those from the last 360 days (0.05). The weighted sum is calculated as:
- $$\text{Weighted Sum} = \sum (\text{Action Count} \times \text{Weight})$$

Data source:

Add watch data source : Watches often indicating a user's interest in a particular item or category. By including watch data, the model can better distinguish between users who are likely to have interest in category and those who have no interest and enhancing the understanding of user engagement patterns.

Process: Add watch data source, extract features from this data, and retrain the model to incorporate these new insights for improved predictions.