CS579 Project 2 weekly sample report

Bhanuja Arekatla(A20449753)
Navya Medarametla(A20442648)

WEEK-5 Date 11/26/2020

This week's work

Generating Caliberated Recommendations(which only optimizes for score, s)

```
class Item:

    def __init__(self, _id, name, categories, score=None):
        self.id = _id
        self.name = name
        self.score = score
        self.categories = categories
```

For a given user, generate the list of items that we can recommend, during this step, we will a lso attach the recommender's score to each item.
Next, we created an objective function for computing the utility score for the list of recommen ded items.
Start with an empty recommendation list ,loop over the topn cardinality and during each iteration update the list with the item that maximizes the utility function.

Then we compared the calibrated recommendation (which only optimizes for score, s) and he original recommendation and the user's interaction distribution.

lmbda : float, 0.0 ~ 1.0, default 0.5 Lambda term controls the score and calibration tradeoff, the higher the lambda the higher the resulting recommendation will be calibrated.
Lambda is keyword in Python, so it's lmbda instead.

```
original reco kl-divergence score: 6.643856189774724
calibrated reco kl-divergence score: 4.070966521354143
```

Printing out the categories distribution from the calibrated recommendation list shows that this list covers more categories and its distribution closely resembles the distribution of the user's past historical interaction and our quantitative calibration metric, KL-divergence also confirms this. i.e. the calibrated recommendation's KL-divergence is lower than the original recommendation's score.

Re ranking the recommendations using KL-divergence value and continued the process until a certain threshold is achieved.