CSE 258 Assignment 1 Report

Vaibhav Kant Tiwari – A59005342

Kaggle Username – vktiwari33

Cook Prediction: **Rank – 884, Pub-0.69820, Priv-0.69940**

Approach 1: -

- Approach from homework 1 was the starting point for this task.
- All the items were ranked according to their popularity, with most popular items at the top. Only a subset of this ranked list was considered for this approach, set using a 'popularity_threshold' variable
- A 'similarity_threshold' variable was defined which was used to define the minimum threshold of Jaccard similarity above which two items would be considered to be similar.
- For each user_id and recipe_id pair, either when the recipe is popular or if it is similar to one of the recipes the user has already cooked do we predict the recipe to be made.
- Got decent accuracy with this approach but needed to grid search amongst the values of 'popularity_threshold' and 'similarity_threshold' to obtain max accuracy.
- For 'popularity_threshold' = 0.48 and 'similarity_threshold' = 0.10, obtained max accuracy of 0.68490

Approach 2: -

- Various ways of feature engineering were tried for applying simple classification models.
- Simply performing feature hashing on IDs and including timestamps produced poor results giving 0.59100 accuracy.
- The second way of performing feature engineering involved getting similarity values for a given pair of user_id and recipe_id along with the popularity of the recipe.
- Similarities were calculated in 2 ways and both were included as a feature. First, similarity by recipe and second similarity by user. The average of `topK` similarities is returned as used a feature for that record.
- Item popularity is normalized count of its occurrence.
- The feature looks like below:
    - [[avgTopKsimByUser, avgTopKsimByItem, itemPopularity]]
- Negative interactions were also added to the dataset.
- The hyper parameters on which grid search was performed were 'topK' and 'C' (regularization for logistic regression model)

- Best accuracy of 0.69820 was achieved for topK = 2, C = 1

Approach 3: -

- Using tensorflow, a neural net with 3 hidden layers [128, 264, 128] was used to train on the data with feature engineering described in approach 2.
- Rating was also added as a feature for this training.
- binary_crossentropy loss function and adam optimizer were used.
- The pipeline performed quite poorly even after 100 epochs of training with the best accuracy achieved 0.59060.

Recipe Rating Prediction **Rank–239, Pub-0.82396, Priv–0.79811**

- Used `surprise` package to directly load contents of `trainInteraction.csv.gz` into a dataset which was split 90:10
- Various algorithms from the `surprise` package were tried.
- Various algorithms tried include: SVD, SVD++, KNNBasic, KNNWithMeans
- Performance with KNNBasic, KNNWithMeans was not that good compared to SVD algorithms.
- Training time with SVD++ was large and thus optimum parameters for SVD++ were not searched.
- Grid search was applied on SVD model to find hyperparameters 'n_epochs', 'lr_all', 'reg_all', and 'n_factors'
- Optimal hyperparameters were found to be following:
    - n_epochs: 17
    - lr_all: 0.005
    - reg_all: 0.2
    - n_factors: 90
- Best RMSE attained for the optimal hyperparameters was 0.82457