HW 10: Ensembling Method: Kaggle Yahoo Music Recommender

EE627

Team: DSDA
Members:
Sivankit Bhanot
Namadev Narne

Ensemble Method:

To apply the ensemble, we first need to attain all the various prior submissions that have been made to Kaggle. These are listed below:

Submission	Description	Accuracy Score (Public)
1 (based_on_mean.csv)	Mean of Album and Artist Ratings was calculated and all non-zero values were assigned '1' in the predictor. The rest were kept as '0'.	0.85745
2 (based_on_mode.csv)	Sum of Album, Artist and one Genre Rating and assigned 3 largest values a '1' for each user and the rest was '0'.	0.84909
3 (based_on_mode_1.csv	Sum of Album and Artist Rating. Same as Submission 2 above but only considering 2 ratings.	0.77711
4 (4genres.csv)	Max of sum of Artist, Album, Genres 1-4 Ratings	0.84402
5 (hw8_mat_fac.csv)	Matrix Factorization	0.67488
6 (lr_answer.csv)	Logistic Regression on Three Ratings (Album, Artist, Genre1)	0.85771
7 (dt_answer.csv)	Decision Tree on Three Ratings (Album, Artist, Genre1)	0.85756
8 (rf_answer.csv)	Random Forest on Three Ratings (Album, Artist, Genre1)	0.85756
9 (gbt_answer.csv)	Gradient-Boosted Tree on Three Ratings (Album, Artist, Genre1)	0.85753

Submission	Description	Accuracy Score (Public)
10 (lr_answer_six.csv)	LR on Six Ratings (Album, Artist, Genre1-4)	0.85595
11 (dt_answer_six.csv)	DT on Six Ratings (Album, Artist, Genre1-4)	0.85481
12 (rf_answer_six.csv)	RF on Six Ratings (Album, Artist, Genre1-4)	0.85709
13 (gbt_answer_six.csv)	GBT on Six Ratings (Album, Artist, Genre1-4)	0.85539

Table 1: Prior Submissions on Kaggle

From these submissions, which are all csv files consisting of two columns: 'ItemID' and 'Predictor', we attained the Predictor column vector from each.
The vector consists of a number of 0's and 1's. The 0's in each of the vectors was changed to '-1'. the 1's were kept as is.

All these column vectors were put together side by side to form the S matrix. All the accuracies for the 13 methods was also stored in a list.

The S matrix will be multiplied by the weight vector to yield that target vector. Since there are 120,000 instances and 13 methods, our S matrix will have dimensions of 120000 X 13. Our target vector (which will also consist of the the ratings for each of the 120000 values) will have dimensions of 120000 X 1. Therefore the weight vector must have dimensions of 13 X 1.

The main challenge is we want to find the optimized value of the weight vector that can give us the best performance. One method to attain this is to use the **Least Squares Solution**. However, this method requires us to know the ground truth for these 120,000 values. If we knew that, then there would not really be any challenge. In essence these 120,000 values are like the test set for us and if we want to use the LS Solution method, we would have to treat the values as a training set, for which we have the ground truth target values already.

For this reason, we have been able to estimate the $\boldsymbol{S^T x}$ part of the LS solution by a vector consisting of $N(2(P_i\text{-}1))$. Here P_i represents the score for the ith submission on Kaggle.

Replacing $\boldsymbol{S^T x}$ by the above in the LS Solution expression, we were able to attain a vector consisting of floating point values around 9 and -9, as shown in Fig 1 below.

```
In [188]: ans_1.head()
```

Out[188]:

	TrackID	Predictor
0	199810_208019	9.134308
1	199810_74139	9.617103
2	199810_9903	8.897845
3	199810_242681	8.897845
4	199810_18515	-8.897845

Fig 1: Top 5 values attained in the Predictor after applying Ensemble Method

From these, the 3 largest values were assigned a '1' and the rest a '0', to attain the Predictor in terms of just 1's and 0's rather than the floating point values as in the Fig 1 above.

However, we discovered that submitting this Predictor on Kaggle, we obtained only an accuracy of 0.20497. This raised a concern, as using all the methods that have a fairly high accuracy we did not expect the accuracy of the ensemble to so low.

This prompted us to investigate and discover that this maybe because the ratings of 1's and 0's in the Predictor may be getting swapped in the process and if we try swapping them, maybe we could attain a better accuracy score. After swapping the 1's and 0's in the 'Predictor' column of the csv file, we were able to obtain an accuracy of 0.79502.

Therefore, as we can see, the results we obtained from the ensemble is not necessarily the best. Among the 13 prior submissions that were used, the best performance was with Method 6 which used logistic regression on three ratings (Album, Artist and Genre 1). The accuracy obtained using the ensemble method is displayed along with the other methods in Fig 2.

Table 2 also shows the performance of the two ensembles (the original and then with swapped 1's and 0's).

Ensemble Submission	**Accuracy Score (Public)**
Ensemble 1	0.20497
Ensemble 1 (swapped 0's and 1's)	0.79502

Table 2: Ensemble Submissions

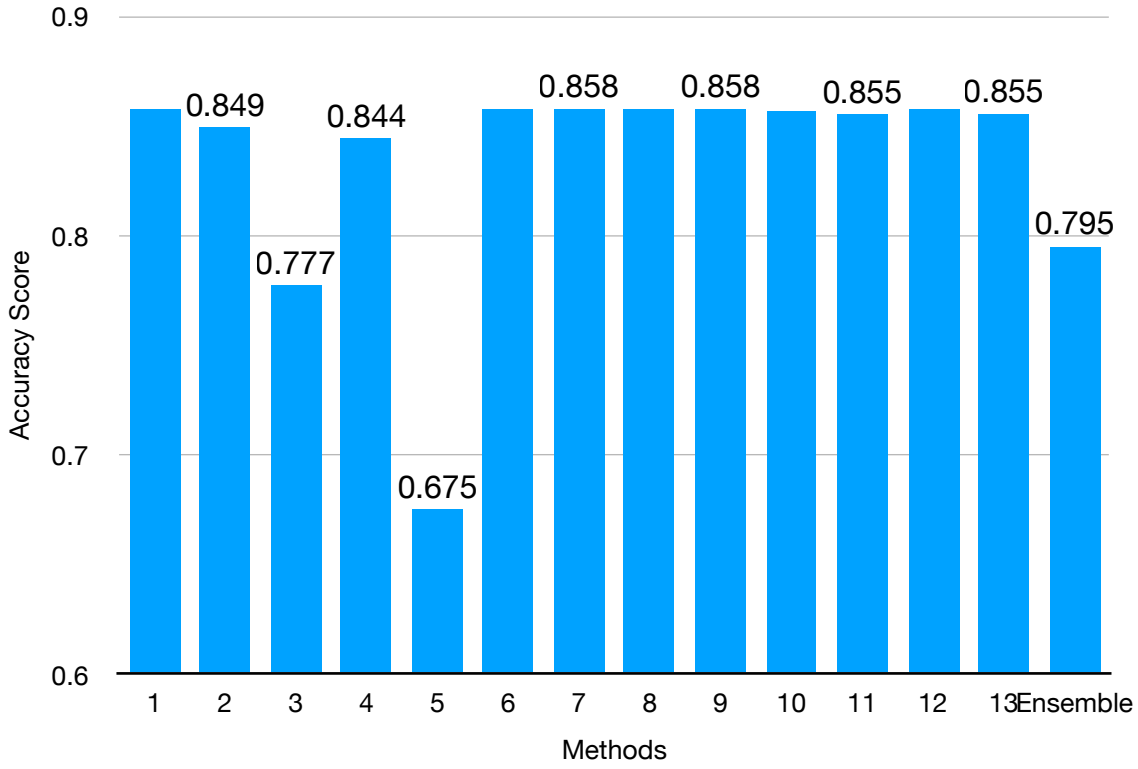

Fig 2: Accuracy of all 13 methods and Ensemble method

From the results obtained, we can see that the ensemble gives a performance that is better than only two of the previous submissions. The prior submission average at around 0.833, and the ensemble is a bit lower than that. It may do us more benefit to attach in some more submissions to get a better accuracy score using the ensemble. For future work, it would also be beneficial to look into extracting more ratings for different genres and then attempt to include those submissions in the ensemble. An ensemble of prior ensemble submissions could also aid the accuracy score.