Math 122 Project

Abhishek Gupta

REMARK: Despite spending many many hours I wasn't able to find the right model for the other two clusters. Predictions for one of the clusters(in file `predictions_cluster_a.csv`) should be accurate, but the others will probably not be good. This is cluster 2 in the `Model_fitting_and_prediction_transcript.txt` file. As can be seen in the transcript, the training error for cluster 2 becomes quite low, but not for the other two clusters when the same method is applied.

1 Finding the right approach

1.1 Neural networks don't work

At first, I tried neural networks for a while, to perform regression. The input data was taken as user website history and the label was taken as the rating. Many architectures and configurations(activation functions, optimization parameters etc) for the network were tried. Obviously all of them took a 100 dimensional input and a single output (with no activation). Loss functions were chosen to be either mean absolute error or mean square error. With a train-test split of 0.2, these gave poor results on the validation set.

The code for all these is in the Failed Approaches ipython notebook. This file contains nothing relevant to the results and is just meant to highlight the approaches tried and their ineffectiveness.

1.2 Trying Cosine similarity

We then try calculating cosine similarity for all pairs. For this we took the dot products of pairs of user websity history vectors. This results in a 4500×4500 matrix. It is sufficent to look at the lower triangle part, because symmetry of cosine makes the matrix symmetric. Thus we calculate only the entries below diagonal. Plotting the result gives the figure 1 below. In the plot of the similarity matrix, the patterns of grey and white imply that there are certainly similarities between some of the users.

To check this further, we fix a user, find cosine similarity of all the users with respect to that and plot that as a scatter plot. Thus there are a total of 4500 plots, all of which can be found at the `https://drive.google.com/drive/folders/15mCt__H7z2lSn0r5he7jrWXFU1LLpbSN?usp=sharing`. Here we present two such plots for users 666 and 612:

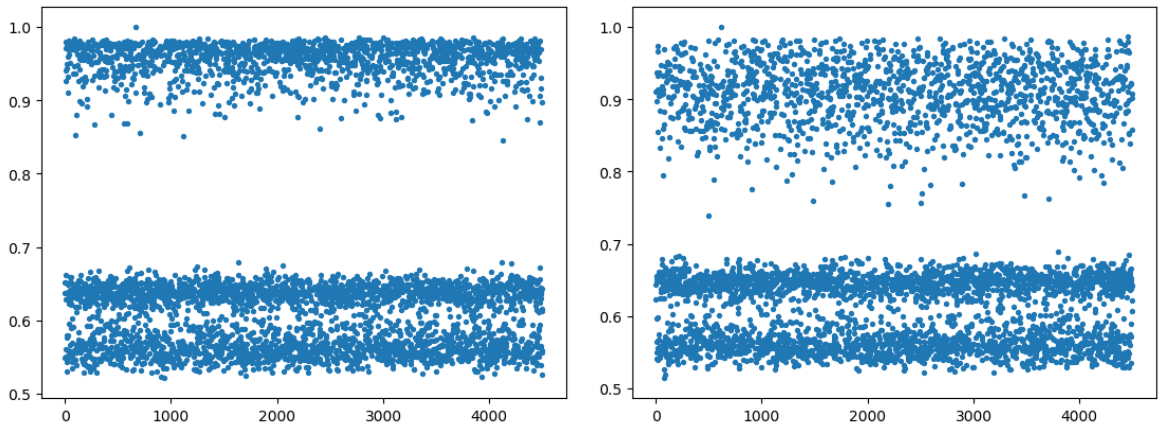

Figure 2: Cosine similarity scatter plots for users 666 and 612

Thus from either of the graphs, we see that there are 3 distince sets of users: one set which has users very similar to the current user and two sets which have similarities of about 0.55 and 0.65. This leads us to think that there might be 3 clusters of users.

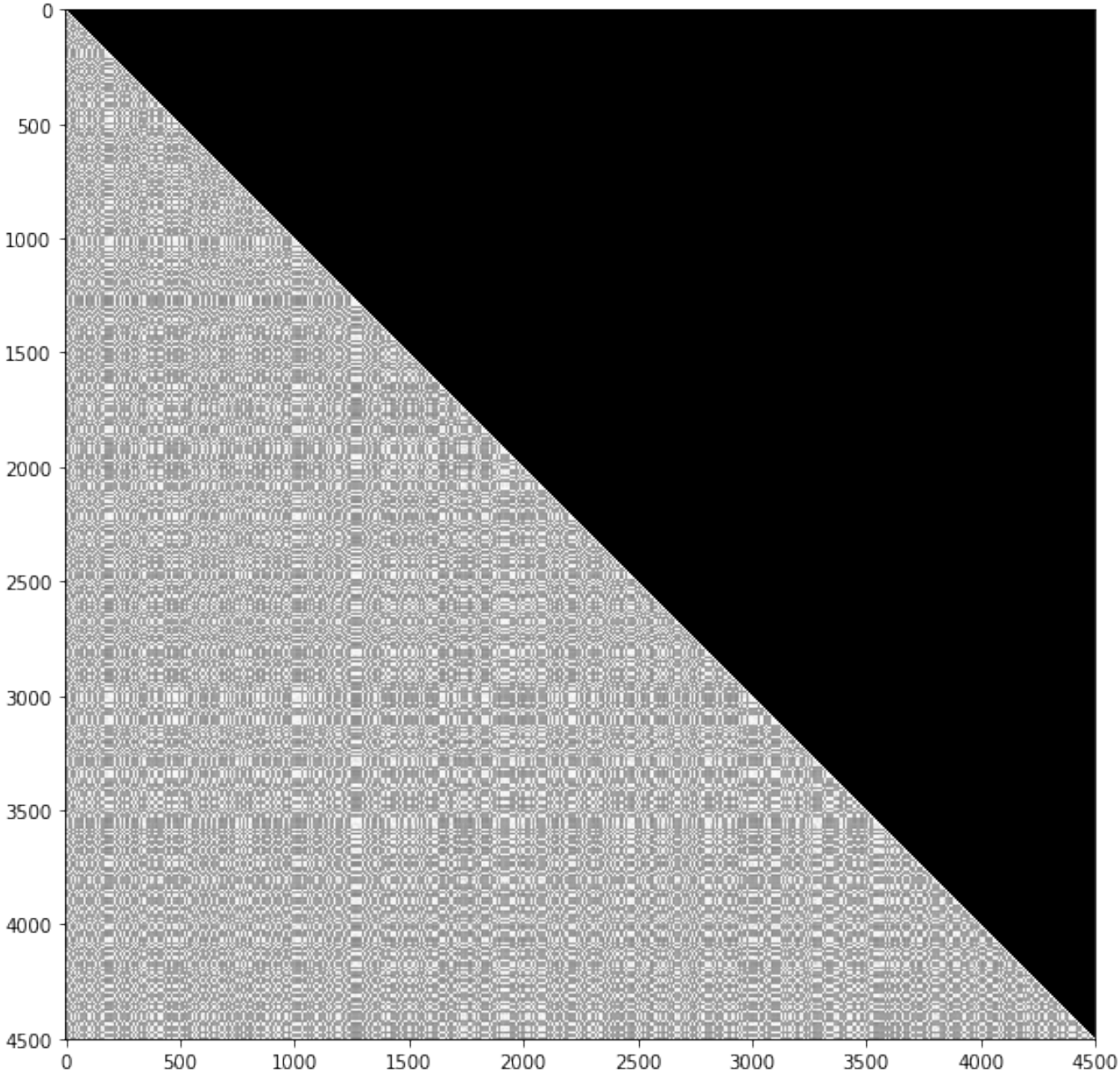

Figure 1: Cosine similarity matrix of pairs plotted

1.3 Trying clustering

We perform k-means clustering on the website history data. To find the right value of k, we plot the objective function vs k plot and look for an elbow. We vary k from 1 to 100.

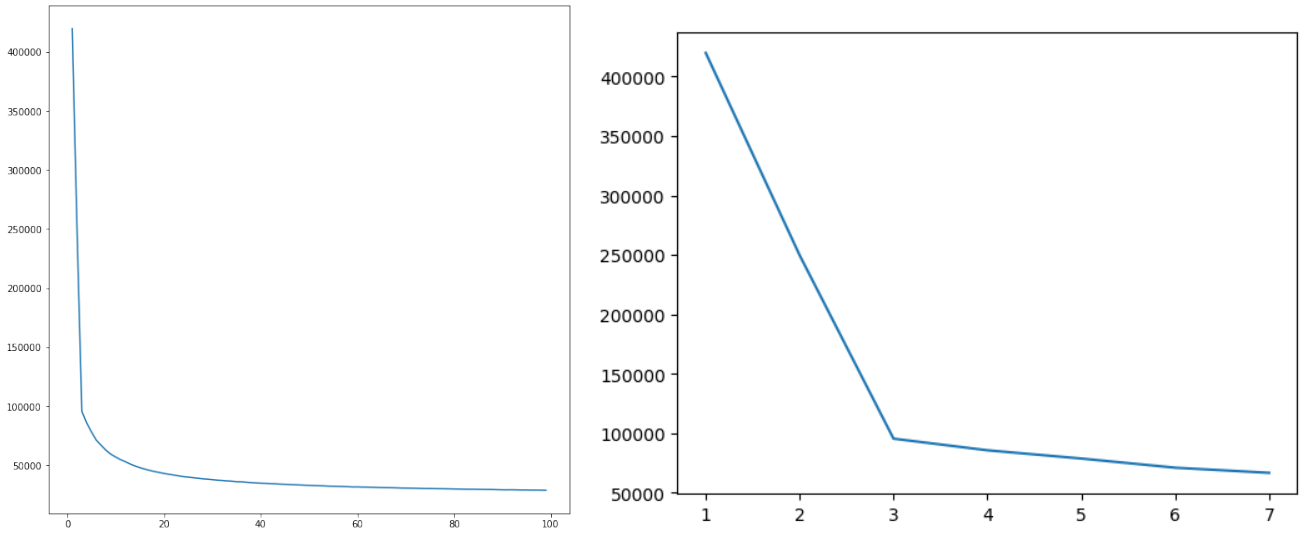

Figure 3: k-means for k =1 to 100: the full graph and zoomed graph near elbow

Zooming in we see an elbow at k=3 for the total error vs k curve. It's a strong indication that there are indeed three clusters.

2 Results of clustering

We now perform 3-means clustering and project onto the space formed by the first two principal components of the website history data. We see 3 clear clusters as shown below:

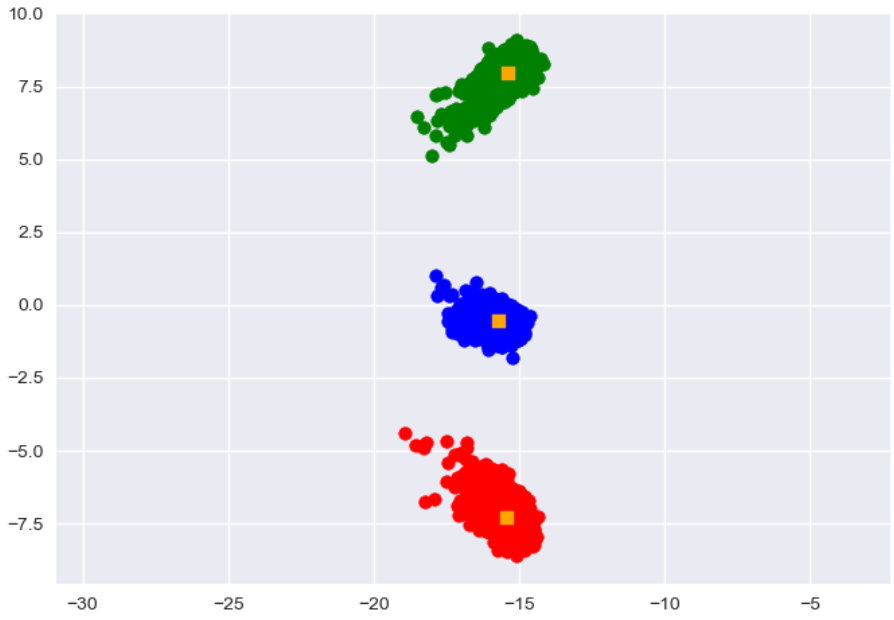

Figure 4: Clusters projected onto space of first two principal components

There are 1500 users in each cluster, totalling to the given 4500 number of users.
We project the clusters onto the first 3 principal components. The result is as shown:

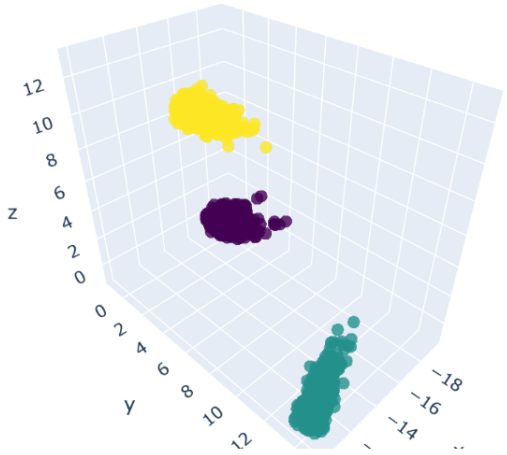

Figure 5: Clusters projected onto space of first 3 principal components

Projecting orhtogonally onto the space formed by the three centroids gives the following:

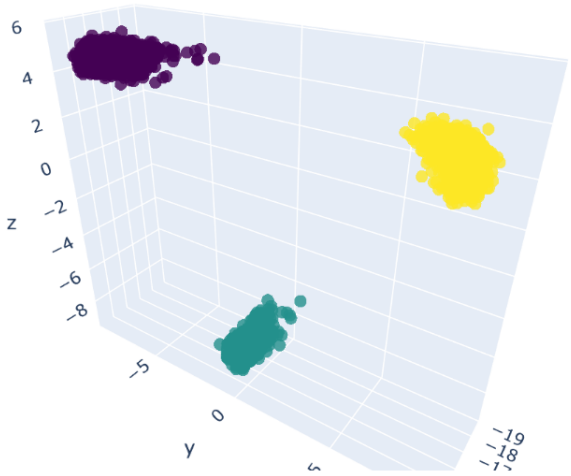

Figure 6: Clusters projected orthogonally onto space of centroids

The plots of the individual clusters projected onto the first 3 principal components:

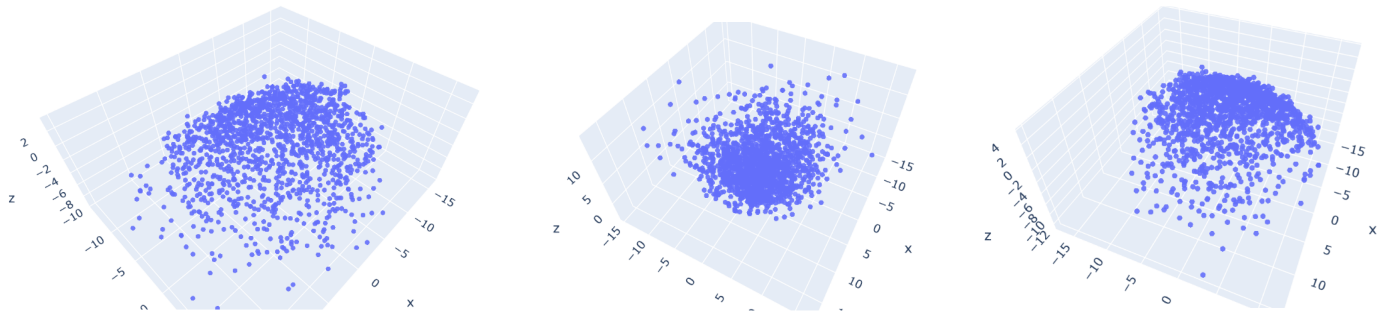

Figure 7: Plots of clusters

All the clusters projected onto their first two principal components plotted on the same graph. This is not a good graph for any quantitative conclusions as the standard deviations for the clusters might be differnt hence each cluster is drawn on a different scale on the same graph, which is technically not very sound. This graph is just drawn to a loose intuition.

Figure 8: Plots of clusters on the same graph

3 Visualizing product ratings

We now turn to visualizing the product ratings. For this we focus on one cluster at a time. We pick a cluster and a product. We then look at all the users from that cluster that have rated that product. Then we project the user history data vectors of all these users onto the first 3 principal components for that cluster. We further color code each point with the rating its user has for that product. The result is shown below for cluster 0(number may be different on different executions of the code) and the product 'alaska basil':

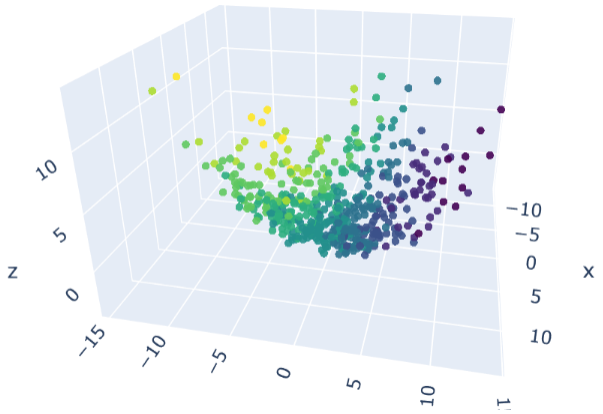

Figure 9: Color coded alaska basil ratings for cluster 0 raters projected onto first 3 principal components

We observe a clear pattern of the color change as an almost linear function of the x and y coordinates (first two principal components). This indicates a clear, possibly linear or close to linear dependence of ratings on principal components and hence website histories.
We drop the third principal component, represented by the z-coordinate in the above graph and instead make the z coordinate as the rating in the hope to see the dependence of the rating on the first two principal components more clearly.

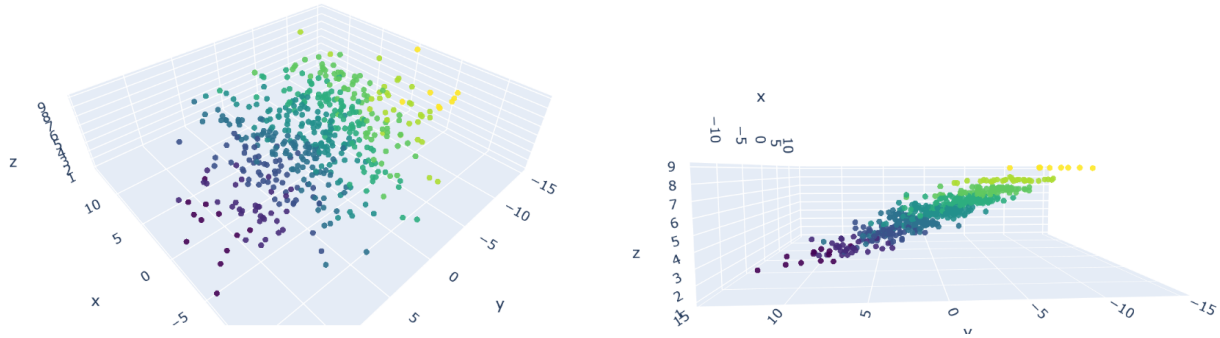

Figure 10: Two views of rating of 'alaska basil' as a function of first two principal components

4 Selecting the right model

4.1 Linear least squares regression

From the second graph above, we see the almost linear dependence of the ratings on the first two principal components! We take the given points and the corresponding ratings. Split them into training and test ratio of 0.2. Then we perform linear least squares regression on the training data and predict the rating on the test data. The results are shown below:

```
Difference counts on test set are:
  0.0    82
 1.0    11
-1.0     9
dtype: int64
Accuracy on test set is: 0.803921568627451
```

Figure 11: Result of linear least squares regression

We calculate the rating prediction on the test set. Then for each rating we find the difference of the predicted rating and the true rating. The figure above shows the number of times each difference is observed. We see that the differnce is 0(i.e. the prediction is correct) 82 out of 82+11+9=102 times. Not just that, when it is not correct it is off by just 1! This means that the linear least squares regression model is very accurate for this product and this cluster.
However this model does not work so well for other products. For example, for the product 'cinema complex', this model for the same cluster of users gives the result shown below:

```
Difference counts are:
  0.0    51
 1.0    21
-1.0    17
 2.0     6
-2.0     5
dtype: int64
Accuracy is: 0.51
```

Figure 12: Result of linear least squares regression is bad for 'cinema complex'

It has only 0.51 accuracy.

4.2 Recommender system model with sigmoid activation

Let's look at the graphs of other product ratings.

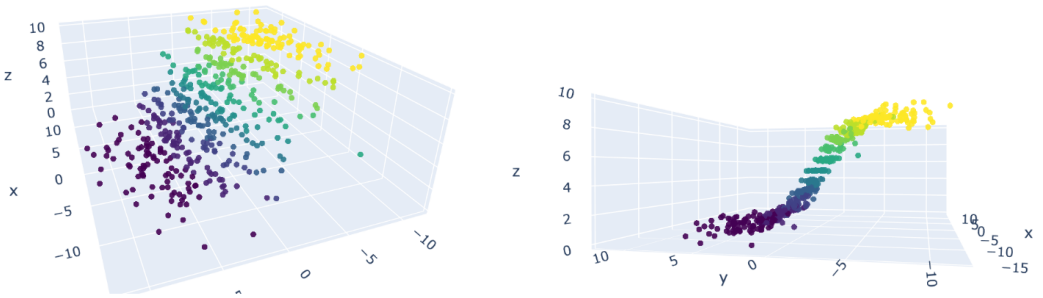

Figure 13: Two views of rating of 'alice ticket' as a function of first two principal components

We see that the ratings aren't exactly linear. It seems that some activation function (sigmoid/tanh) has been applied to the result of a linear model. The same thing is observed in the graph of 'cinema complex' below:

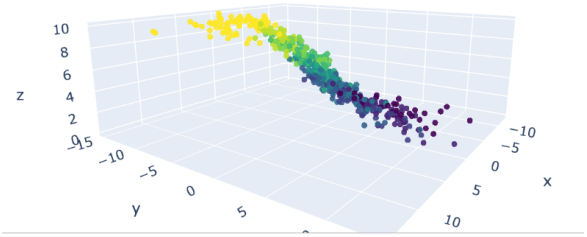

Figure 14: 'cinema complex' also indicating use of an activation function

Thus, this suggests that the recommender system model with the p's as user history data and sigmoid activations might be the right model. For a fixed cluster and a fixed product, this model takes p's to be the 100-dimensional user history vectors and solves using gradient descent, for a 100-vector q such that $\sum_i (r_i - p_i \cdot q)^2$ is minimal. The index i is summed over all the users of the cluster which have rated the said product (for testing purpose, a smaller subset maybe seprated instead of taking all the users). We do this and compare the results with linear least squares regression in the section below:

4.3 Results with Recommender system model

Linear least squares regression for 'comedy clean' gives

```
Difference counts on test set are:
  0.0    51
-1.0     24
 1.0     19
-2.0      2
 2.0      1
dtype: int64
Accuracy on test set is: 0.5257731958762887
```

Figure 15: Linear least squares regression for 'comedy clean' gives a low accuracy of 52%

We train our recommender system model for 5000 epochs with learning rate as 0.1. (this is not optimal for other products but works in this case). The results of this are as shown below:

```
Accuracy is: 0.8350515463917526
 Rating difference counts:
  0    81
 1     9
-1     7
dtype: int64
```

Figure 16: Using recommender system increases the accuracy to 83% for 'comedy clean'

Using recommender system increases the accuracy to 83% for 'comedy clean'. This happens for other products as well. Thus this is indeed the right model! This is the one we use for our predictions.

5 Problem with other clusters

Having found the right model that works on one cluster, we might think that it works on others too. But it doesn't. Despite spending over a dozen hours checking, rechecking and redoing many times, the predictions for the other clusters, I failed to find an error. If the same approach works and my code has an error, then the error is probably very subtle. The following reasoning confuses me further about finding the right model for the other clusters.

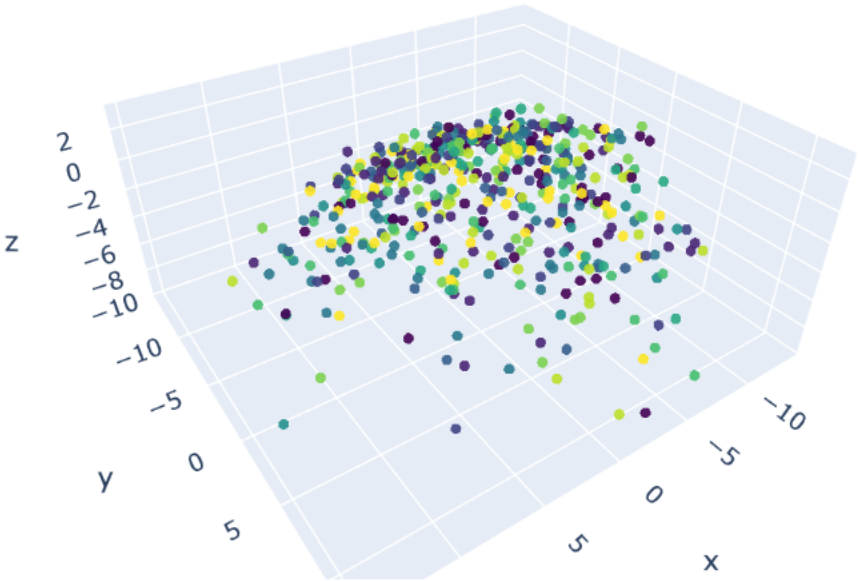

Figure 17: 'alaska basil' ratings for another cluster - no observable pattern

From the image above, it is not at all clear what the right model should be. Points nearby don't have the same rating. For example dots of color yellow are distributed throughout.

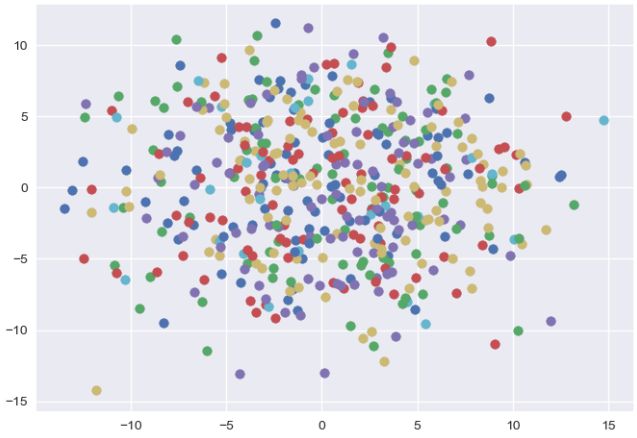

Figure 18: ‘alaska basil’ ratings projected only on first two principal components

In order for there to be a pattern, the points nearby should have similar ratings (or similar colors). The only way that the pattern can appear so random is if the rating does not depend on the first three principal components at all but only on the later principal components. Even then a neural network fails to capture any pattern (I tried several architectures). Because of this, we suspect that there might be no pattern in the ratings or a rather complicated one. Looking at the dots of the same color it seems that for each rating the points are just randomly normally distributed.

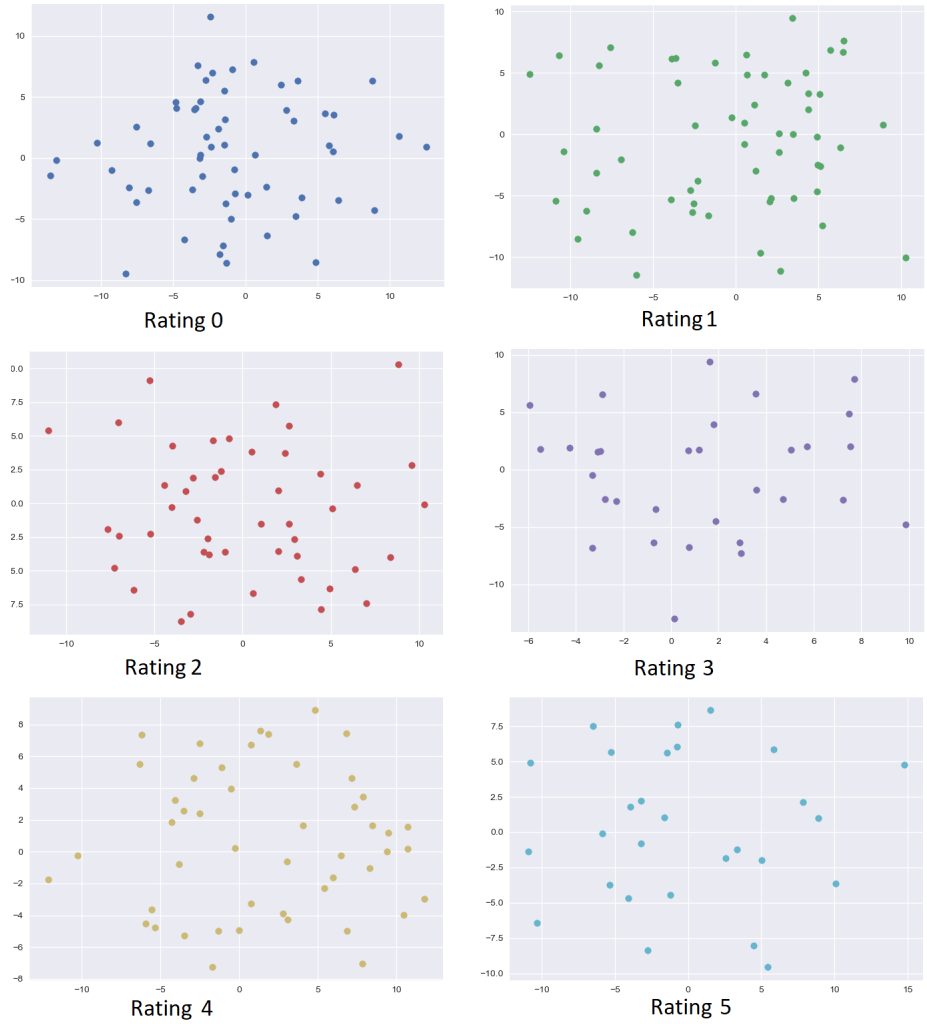

Figure 19: Points corresponding to different ratings plotted on different graphs

In the figure 19, we see that for any rating the corresponding user points are distributed all over, without any apparent

functional dependence on the x or y - coordinate of the plane (first two principal components). Thus any pattern if present is probably complex.

6 Creating prediction results

We train a recommender system model for each cluster and product. Model fitting transcript text file included with the submission shows the results of training proceedure. From that text file, we can indeed see that this model does not work well on the other two clusters.
Experimenting with learning rate and epochs for a variety of products we found that the following suit best:

- learning rate: 0.008 and larger can lead to the model diverging for some products
- epochs: 6000 models for some products converge faster, but overall these many epochs are needed for some products once we fix the learning rate, these many epochs are enough for models for all products to converge

We save the results in the file `predictions.csv` as required. The first 150000 lines correspond to users of the good cluster. The next 300000 lines correspond to users of the remaining 2 bad clusters. We also split the prediction file in three separate files for each cluster.