Assignment 2

The deadline for the second assignment is May 2, before class. Turn in the code, with instructions on how to run it. The report should include detailed observations on the results. For late submissions the late policy on the page of the course will be applied. Details for the turn-in, and how to write reports are on the Assignments web page of the course. The will be an oral examination of the Assignment.

Question 1

An exponential distribution has probability density function $f(x) = \lambda e^{-\lambda x}$, where λ is the parameter of the distribution. You are given a set of observations $X = \{x_1, \dots, x_n\}$ that are generated from an exponential distribution. Use the Maximum Likelihood Estimation method we described in class to find the parameter of the distribution that best fits the data observations.

Question 2 (Principal Component Analysis)

You are given the file "data2.txt" which contains a 500×20 sparse matrix. The matrix is stored in tab-separated triples (row, column, value). The matrix holds information about the consumption of 20 chemical substances by 500 users. The values are the number of times that each user has used a substance. There are 10 legal substances (columns 0-9) and 10 illegal substances (columns 10-19).

Load the data and apply Principal Component Analysis (use the PCA package from sklearn). Examine the first two components. Plot the points in one and two dimensions. What do you observe? How do you explain the results? Hand in a notebook with the plots and a report with your analysis.

Question 3 (Recommendation systems)

The goal of this question is to experiment with recommendation algorithms.

You will use the Yelp dataset that you used in Assignment 1. In this Assignment, you will use the `business.json` and `review.json` files (the last one is more than 3GB so you will need space to store it, and you take this into account when processing it). Using this data you will create a user-business rating matrix with the ratings of all users for all businesses in the city of "Toronto". Keep only users that have at least 10 ratings, and businesses that have at least 10 ratings (perform this iteratively until all users and all businesses in your matrix have at least 10 ratings).

Remove a randomly selected 10% of the ratings. The goal is to estimate these ratings by applying the collaborative filtering techniques we saw in class with the remaining 90% of the data. You will try the following algorithms for predicting for user u their rating for business b:

1. **User Average (UA):** Use the mean $\overline{r(u)}$ of the ratings of u as the prediction.
2. **Business Average (BA):** Use the mean $\overline{r(b)}$ of the ratings for b as the prediction.

3. **User-based Collaborative Filtering** (**UCF**): For user u find the set $N_k(u)$ of the k most similar users that have rated business b. Then use the following formula for your prediction:

$$p(u,b) = \overline{r(u)} + \frac{\sum_{u' \in N_k(u)} s(u,u')(r(u',b) - \overline{r(u')})}{\sum_{u' \in N_k(u)} s(u,u')}$$

   For the similarity measure use the correlation coefficient (the cosine similarity after removing the mean of each row). If the value becomes less than 1 or greater than 5 round up to 1 or 5.
4. **Item-based Collaborative Filtering** (**ICF**): For business b find the set $N_k(b)$ of the k most similar businesses (according to their cosine similarity) that have been rated by user u. Then use the following formula for your prediction:

$$p(u,b) = \frac{\sum_{b' \in N_k(m)} s(b,b')r(u,b')}{\sum_{b' \in N_k(b)} s(b,b')}$$

5. **Singular Value Decomposition** (**SVD**)**:** Apply the Singular Value Decomposition on the ratings matrix R, and keep the k highest singular vectors to obtain a rank-k matrix R_k (use the singular values to decide the value of k). Then use the following formula for your prediction: $p(u,b) = R_k(u,b)$. (Bonus: Experiment with Principal Component Analysis as well – does it make a difference?)

For the evaluation and comparison of the algorithms you will use the RMSE (Root Mean Square Error) metric. If $r_1, r_2, \dots, r_n$ are the ratings that we want to predict, and $p_1, p_2, \dots, p_n$ are the predictions of an algorithm, then the Sum of Square Errors of the algorithm is defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - p_i)^2}$$

Create plots for the RMSE error for different values of k for all algorithms.

You should turn in the following:

- All the code that you have written yourselves.
- A short report with a description of what you did, a comparison of the performance of the different algorithms, and a commentary on the output.

Notes:

- The file with the reviews is very large so you cannot load it to memory. Also, the number of user/business pairs is also large, so in order to handle them you should use the appropriate data structures.
- Use the python functions for computing distances and matrix operations, which are much more efficient.

Question 3 (Clustering)

In this Question you will practice with the application of clustering algorithms. We will use a dataset similar to that you used for Assignment 1. You will take again the businesses from the city of Toronto that are in neighborhoods with at least **300** businesses. Using this data you will look at the following problems:

1. Take the geographical coordinates of the businesses and apply clustering algorithms in the two-dimensional points. Use the algorithms k-means, agglomerative and DBSCAN. Use (whenever necessary) number of clusters equal to the number of the neighborhoods in the data. For DBSCAN experiment with different values for the parameters (Bonus: do the plot with the distance to the minPts-closest neighbor to decide the value of eps). Create the confusion matrix for the three algorithms and report the precision and recall. Visualize the data with a plot, and place them on the map of the city. Examine the data visually, and comment on what you observe.
2. For each business in the list, take the associated categories. Keep the businesses that have in their category list the categories "Food" and "Restaurants" (both). Keep the categories that appear in at least 5% of the businesses (remove the categories "Food" and "Restaurants" which appear everywhere). Keep the businesses that have at least one category from the ones you selected. For the businesses that have more than one categories, keep the one that is most frequent in your data. Remove the categories (and the corresponding businesses) that appear in less than 15 businesses.

   For each of the businesses in the final set of businesses, from the file review.json take all the reviews for the business and create a large document. Use these documents to create tf-idf representation of the businesses (use the existing library of python to obtain this representation – you can make your own choices for the parameters of the library). Cluster the businesses using the k-means and agglomerative clustering algorithms, with cluster equal to the number of the categories. Examine if the clusters that you find correspond to the categories, using the confusion matrix and the precision/recall metrics. Comment on the results.
3. It is possible that the documents do not cluster in a way that agrees with the categories. For the dataset you created in the previous step, using the k-means algorithm, create the silhouette plot to decide the number of clusters. For the number of clusters you selected, examine the clusters output by the k-means algorithm using the words and the categories (you can use the full list of categories) and try to determine what type of restaurant corresponds to each cluster.

Hand in your code and a report with a detailed commentary and analysis of your results.