Assignment 4

Due: Saturday, May 26

For this assignment you will experiment with Principal Component Analysis as a dimensionality reduction approach to assist in clustering high-dimensional data. You will also experiment with item-based recommendation for a joke recommender system.

---

1. **PCA for Reduced Dimensionality in Clustering** [**Dataset**: segmentation_data.zip]

   For this problem you will use an **image segmentation data set** for clustering. You will experiment with using PCA as an approach to reduce dimensionality and noise in the data. You will compare the results of clustering the data with and without PCA using the provided image class assignments as the ground truth. The data set is divided into three files. The file "segmentation_data.txt" contains data about images with each line corresponding to one image. Each image is represented by 19 features (these are the columns in the data and correspond to the feature names in the file "segmentation_names.txt". The file "segmentation_classes.txt" contains the class labels (the type of image) and a numeric class label for each of the corresponding images in the data file. After clustering the image data, you will use the class labels to measure completeness and homogeneity of the generated clusters. The data set used in this problem is based on the **Image Segmentation data set at the UCI Machine Learning Repository**.

   Your tasks in this problem are the following:

   a. Load in the image data matrix (with rows as images and columns as features). Also load in the numeric class labels from the segmentation class file. Using your favorite method (e.g., **sklearn's min-max scaler**), perform min-max normalization on the data matrix so that each feature is scaled to [0,1] range.
   b. Next, Perform Kmeans clustering (for this problem, use the Kmeans implementation in **scikit-learn**) on the image data (since there are a total 7 pre-assigned image classes, you should use K = 7 in your clustering). Use **Euclidean distance** as your distance measure for the clustering. Print the cluster centroids (use some formatting so that they are visually understandable). Compare your 7 clusters to the 7 pre-assigned classes by computing the **Completeness and Homogeneity** values of the generated clusters.
   c. Perform PCA on the normalized image data matrix. You may use the linear algebra package in Numpy or the Decomposition module in scikit-learn (the latter is much more efficient). Analyze the principal components to determine the number, **r**, of PCs needed to capture **at least 95% of variance** in the data. Then use these **r** components as features to transform the data into a reduced dimension space. [See the **PCA Clustering Notebook** from class for an example of how these steps are performed.]
   d. Perform Kmeans again, but this time on the lower dimensional transformed data. Then, compute the Completeness and Homogeneity values of the new clusters.
   e. Discuss your observations based on the comparison of the two clustering results.

---

2. **Item-Based Joke Recommendation** [**Dataset**: jokes.zip]

   For this problem you will use a modified version of the item-based recommender algorithm from Ch. 14 of Machine Learning in Action and use it on joke ratings data based on **Jester Online Joke Recommender System**. The modified version of the code is provided in the module **itemBasedRec.py**. Most of the module will be used as is, but you will add some additional functionality.

   The data set contains two files. The file "**modified_jester_data.csv**" contains the ratings on 100 jokes by 1000 users (each row is a user profile). The ratings have been normalized to be between 1 and 21 (a 20-point scale), with 1 being the lowest rating. A zero indicated a missing rating. The file "**jokes.csv**" contains the joke ids mapped to the actual text of the jokes.

   Your tasks in this problem are the following (please also see comments for the function stubs in the provided module):

   a. Load in the joke ratings data and the joke text data into appropriate data structures.
   b. Complete the definition for the function "**test**". This function iterates over all users and for each performs evaluation (by calling the provided "cross_validate_user" function), and returns the error information necessary

to compute Mean Absolute Error (MAE). Use this function to perform evaluation (wiht 20% test-ratio for each user) comparing MAE results using standard item-based collaborative filtering (based on the rating prediction function "standEst") with results using the SVD-based version of the rating item-based CF (using "svdEst" as the prediction engine). [**Note:** See comments provided in the module for hints on accomplishing these tasks.]

c. Write a new function "print_most_similar_jokes" which takes the joke ratings data, a query joke id, a parameter k for the number of nearest neighbors, and a similarity metric function, and prints the text of the query joke as well as the texts of the top k most similar jokes based on user ratings. [**Note:** For hints on how to accomplish this task, please see comments at the end of the provided module as well as comments for the provided stub function.]

Notes on Submission: You must submit your Jupyter Notebook (similar to examples in class) which includes your documented code, results of your interactions, and any discussions or explanations of the results. Please organize your notebook so that it's clear what parts of the notebook correspond to which problems in the assignment. Please submit the notebook in both IPYNB and HTML formats (along with any auxiliary files). Your assignment should be submitted via D2L.