HW03 – Appleton Clustering and Recommender Systems

FILES

1. Appleton_cluster.csv – See instruction 2 below.
2. large_cluster.csv – See instruction 4.2 below.

Note: You will conduct cluster analysis using KMeans, Inertias Models, and Hierarchal Clustering with Dendrograms for ALL customers. You will then conduct cluster analysis using the largest group.

Format of this Homework

It is very important that your Jupyter Notebook is formatted correctly with markdown, comments, and code that works.

You are to do the following for each section:

- Include a title as markdown Heading 2, for example: "Section 2.1 Scale the data"
- Include a description of the section detailing its purpose (*markdown*)
- Include your code and make sure it is executable and correct. (*code*)
- At the end of the section, include a summary **describing results**. **Your summaries should also provide answers to the questions listed in the instructions**. (*markdown*)

How to turn it in:

- **Create ONE Jupyter notebook named HW03_*LastNameFirstInitial*.**
- You are to turn in your Jupyter notebook file only. No data files and no folders.
- It is assumed that you created your Jupyter notebook in a folder named HW03_student and in that folder is a data folder. It is expected the path for importing data is in "data" folder, for example '**data/Appleton_cluster.csv'**.

Business and Technology Objectives

Look at bike rental data for numerical predictions and relationships

- ***Business Objective:*** Test if Appleton customers cluster into groups (1 – 3.3)
- ***Technology Objective:*** Use K-Means Cluster Analysis
- ***Business Objective:*** Test if Appleton largest customer group clusters into additional groups (4)
- ***Technology Objective:*** Use K-Means (4.2) and Hierarchal Cluster Analysis (4.3)

Cluster Analysis for ALL borrowers (member_id) – what are the customer groups?

1. **Import Libraries**
2. **Cluster Analysis for ALL borrowers (member_id) - Import Data**
   - Import the csv file Appleton_Cluster.csv as a dataframe and name it "**loan_cluster."**
   - Drop the "member_id" from **loan_cluster**.
   - Use StandardScaler for "**loan_cluster"** and name the new dataframe "**loan_scaled."**
3. **KMeans for ALL borrowers**

   3.1 KMeans for 3 Groups
   - Create a KMeans for 3 clusters
   - Create an inertias model using "ks = range(1, 21) to see how many clusters are ideal. This could take up to 5 minutes or more to run.
   - Add the line **print("Iteration {} done".format(k))** to the end of your for loop to keep a tab on where it is in the process. *(This may take up to ten minutes to complete.)*
   - For your predicted clusters create a DataFrame named **labels3**.
   - Create a new DataFrame named **results** that will concat **loan_cluster** with **labels3**.

   3.2 KMeans for 8 Groups
   - Create a KMeans cluster based on the inertias model chart (Look for the bump in the model chart, which is at 8 or 9, therefore use n_clusters=8).
   - For your predicted clusters (k=8) create a DataFrame named **labels8**.
   - Concat **labels8** to your **results** DataFrame.
   - Create a countplot for your results dataset. The Y variable should be your predicted label. The hue should be "loan_is_bad_num" to differentiate the loans that were good and the loans that defaulted.

   3.3 Summary of Clusters
   - Using Markdown, summarize your results and give some insightful comments pertaining to the data that you see.
     - Looking at the largest groups, what type of borrowers does this resemble?
     - Are outlier groups different than the rest of the groups? How so?
4. **Cluster Analysis for the largest group – what are the borrower groups within this specific sector of customers?** *Looking at the clusters in the analysis for ALL borrowers, you should have one group that is larger (number in the cluster) than the rest of the groups. We would like to delve deeper into the differences between borrowers within this one group.*

   4.1 Data Preparation (Don't forget to scale before you cluster)
   - Import the **large_cluster.csv** and name it **loan_largest**. (NOTE: index_col=None)

   4.2 Using the new dataframe for the one specific customer group – create a KMeans Cluster (using the loan_largest DataFrame)
   - Create a KMeans for 3 clusters.
   - Create an inertias model to see how many clusters are ideal. You should be able to use the same as given.
   - Create a KMeans cluster based on the inertias model chart.
   - Create a countplot that shows the number of loans in each cluster (similar to the countplot that was created above).

4.3 Hierarchal Clustering for Specific Group

- Create a dendrogram to visualize the number of possible clusters using hierarchal clustering. Note that this is computationally intensive and will likely take over a minute to compute on your system. (Also, due to the number of instances in the dataset, your dendrogram may be difficult to read on the x axis. The distance (y axis) is the number you really need.)
- Create clusters based on the dendrogram.

4.4 Summary of Clusters

- Create at least one visualization (barplot, boxplot, scatterplot, etc.) that best indicates a reason why groups may be interesting. Choose the cluster analysis that you feel best indicates differences.
- Based on all of your analysis, including your visualizations, use Markdown to summarize your results and give some insightful comments pertaining to the data that you see for the specific groups.
    - Did they group? If so, what are the characteristics of the new groups?
    - Are there differences in the loan amounts? What are their employment lengths, risk factors, etc.
    - Give some meaningful analysis that you feel would benefit Appleton.
    - This section should be insightful and not just a summary of the results. Include your opinion on how this is useful and why this could lead to actionable steps to better understand Appleton's customers.

Submission

Save your file as HW03_LastNameFirstInitial.ipynb and turn it in to D2L per the Dropbox instructions. (file only, no data)

Appendix A: Code for Inertias Model

```
ks = range(1,12)
inertias = []

for k in ks:
    model = KMeans(n_clusters = k)
    model.fit(loan_scaled)
    inertias.append(model.inertia_)
    print("iteration {} done".format(k))

plt.plot(ks, inertias, '-o')
plt.xlabel('number of clusters, k')
plt.ylabel('inertia')
plt.xticks(ks)
plt.show()
```