Problem: **Clustering Heart Disease Patient Data**

The solution to the following problem must be completed by you and only you. You may seek help only from the professor and the tutor, and you may use your texts, notes, online tutorials, etc., but the code must be your own. Bear in mind that the professor and the tutor are helping you with your Python problem and not writing your program! If you have trouble compiling or debugging your code, see the professor or the tutor as soon as possible.

This project is adopted from datacamp. Originally created for R language by Megan Robertson, Data Scientist at Nike.

1. Targeting treatment for heart disease patients

Clustering algorithms are used to group together items that are similar to one another. There are many industries where it would be beneficial and insightful to use an unsupervised learning algorithm - retailers want to group similar customers for targeted ad campaigns, biologists want to find plants that share similar characteristics, and more. We are going to explore if it would be appropriate to use some clustering algorithms to group medical patients.

We are going to look at anonymized patients who have been diagnosed with heart disease. Patients with similar characteristics might respond to the same treatments, and doctors would benefit from learning about the outcomes of patients similar to those they are treating. The data we are analyzing comes from the V.A. Medical Center in Long Beach, CA. For more information, see here.

Before beginning a project, it is important to get an idea of what the patient data looks like. In addition, the clustering algorithms used below require that the data be numeric, so it is necessary to ensure the patient data doesn't need any transformations.

Task 1: Instructions

Start by loading the data and learning more about the variables in the data set.

- Read the data located in `datasets/heart_disease_patients.csv` into a variable called `heart_disease`.
- Play with it, have fun!

2. Quantifying patient differences

It is important to conduct some exploratory data analysis to familiarize ourselves with the data before clustering. This will help us learn more about the variables and make an informed decision about whether we should scale the data. Because k-means and hierarchical clustering measures similarity between points using a distance formula, it can place extra emphasis on certain variables that have a larger scale and thus larger differences between points.

Exploratory data analysis helps us to understand the characteristics of the patients in the data. We need to get an idea of the value ranges of the variables and their distributions. This will also be helpful when we evaluate the clusters of patients from the algorithms. Are there more patients of one gender? What might an outlier look like?

Task 2: Instructions

Check if the data should be scaled before performing the clustering algorithms.

- Look at the distributions of the variables in `heart_disease`
- Remove the `id` variable from the data set.
- Scale the data and save in a new data frame `scaled`.
- Look at the distributions of the variables

3. Let's start grouping patients

Once we've figured out if we need to modify the data and made any necessary changes, we can now start the clustering process. For the k-means algorithm, it is necessary to select the number of clusters in advance.

It is also important to make sure that your results are reproducible when conducting a statistical analysis. This means that when someone runs your code on the same data, they will get the same results as you reported. Therefore, if you're conducting an analysis that has a random aspect, it is necessary to set a seed to ensure reproducibility.

Reproducibility is especially important since doctors will potentially be using our results to treat patients. It is vital that another analyst can see where the groups come from and be able to verify the results.

Task 3: Instructions

Run an iteration of the k-means algorithm.

- Set the seed to the number 10
- Define the number of clusters to be 5 and save as `k`.
- Run the k-means algorithm on the `scaled` data Name the returned k-means object `first_clust`.
- Find the counts of the number of each patients in each cluster

4. Another round of k-means

Because the k-means algorithm initially selects the cluster centers by randomly selecting points, different iterations of the algorithm can result in different clusters being created. If the algorithm is truly grouping together similar observations (as opposed to clustering noise), then cluster assignments will be somewhat robust between different iterations of the algorithm.

With regards to the heart disease data, this would mean that the same patients would be grouped together even when the algorithm is initialized at different random points. If patients are not in similar clusters with various algorithm runs, then the clustering method isn't picking up on meaningful relationships between patients.

We're going to explore how the patients are grouped together with another iteration of the k-means algorithm. We will then be able to compare the resulting groups of patients.

5. Comparing patient clusters

It is important that the clusters resulting from the k-means algorithm are stable. Even though the algorithm begins by randomly initializing the cluster centers, if the k-means algorithm is the right choice for the data, then different initializations of the algorithm will result in similar clusters.

The clusters from different iterations may not be exactly the same, but the clusters should be roughly the same size and have similar distributions of variables. If there is a lot of change in clusters between different iterations of the algorithm, then k-means clustering is not a good choice for the data.

It is not possible to validate that the clusters obtained from an algorithm are ground truth are accurate since there is no true labeling for patients. Thus, it is necessary to examine how the clusters change between different iterations of the algorithm. We're going to use some visualizations to get an idea of the cluster stabilities. That way we can see how certain patient characteristics may have been used to group patients together.

6. Hierarchical clustering: another clustering approach

An alternative to k-means clustering is hierarchical clustering. This method works well when the data has a nested structure. It is possible that the data from heart disease patients follows this type of structure. For example, if men are more likely to exhibit certain characteristics, those characteristics might be nested inside the gender variable. Hierarchical clustering also does not require the number of clusters to be selected prior to running the algorithm.

Clusters can be selected by using the dendrogram. The dendrogram allows one to see how similar observations are to one another and are useful in selecting the number of clusters to group the data. It is now time for us to see how hierarchical clustering groups the data.

7. Hierarchical clustering round two

In hierarchical clustering, there are multiple ways to measure the dissimilarity between clusters of observations. Complete linkage records the largest dissimilarity between any two points in the two clusters being compared. On the other hand, single linkage is the smallest dissimilarity between any two points in the clusters. Different linkages will result in different clusters being formed.

We want to explore different algorithms to group our heart disease patients. The best way to measure dissimilarity between patients could be to look at the smallest difference between patients and minimize that difference when grouping together clusters. It is always a good idea to explore different dissimilarity measures. Let's implement hierarchical clustering using a new linkage function.

8. Comparing clustering results

The doctors are interested in grouping similar patients together in order to determine appropriate treatments. Therefore, they want to have clusters with more than a few patients to see different treatment options. While it is possible for a patient to be in a cluster by themselves, this means that the treatment they received might not be recommended for someone else in the group.

As with the k-means algorithm, the way to evaluate the clusters is to investigate which patients are being grouped together. Are there patterns evident in the cluster assignments or do they seem to be groups of noise? We're going to examine the clusters resulting from the two hierarchical algorithms.

9. Visualizing the cluster contents

In addition to looking at the distributions of variables in each of the hierarchical clustering run, we will make visualizations to evaluate the algorithms. Even though the data has more than two dimensions, we can get an idea of how the data clusters by looking at a scatterplot of two variables. We want to look for patterns that appear in the data and see what patients get clustered together.

10. Conclusion

Now that we've tried out multiple clustering algorithms, it is necessary to determine if we think any of them will work for clustering our patients. For the k-means algorithm, it is imperative that similar clusters are produced for each iteration of the algorithm. We want to make sure that the algorithm is clustering signal as opposed to noise.

For the sake of the doctors, we also want to have multiple patients in each group so they can compare treatments. We only did some preliminary work to explore the performance of the algorithms. It is necessary to create more visualizations and explore how the algorithms group other variables. Based on the above analysis, are there any algorithms that you would want to investigate further to group patients? Remember that it is important the k-mean algorithm seems stable when running multiple iterations.