Tutorial 10 Reduction

This tutorial consists of two data explorations. The first uses data that is relatively similar to the data in the lecture notes and examples. The second is more unstructured and open ended.

Notebooks: We will be using Colab to create Python Notebooks on your Google Drive although you may also use Jupyter. Read the document Tutorials.pdf from Moodle to see how to set these up.

Naming: For each exploration you should create a notebook and save it when you have finished. You should name the two notebooks Tut06-A.ipynb and Tut06-B.ipynb.

Structure: Every numbered item in the exploration should have a code section and a markdown section underneath where you discuss your findings. There should also be a code section at the top of the Notebook with the imports.

Exploration A

The company who supplied the Products data in the lecture notes also want an investigation into their website pages. This week you will investigate the structure of this data using t-SNE.

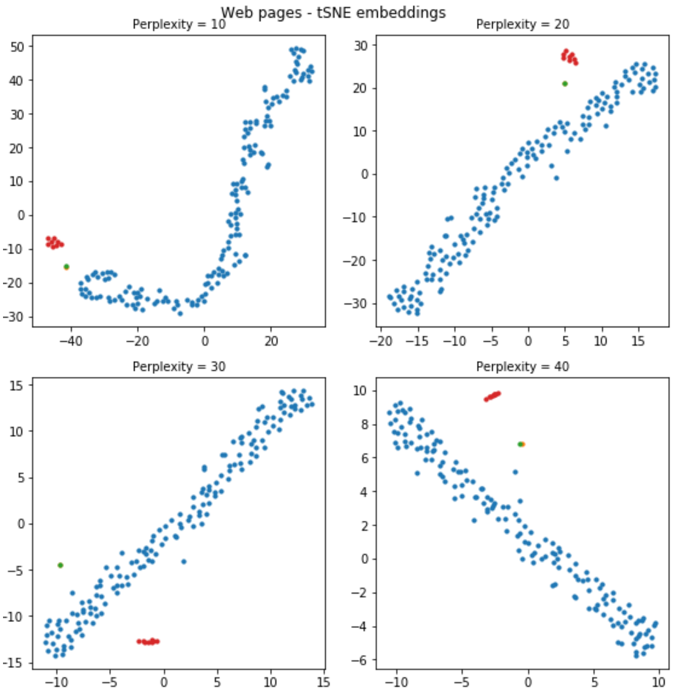
1. First read all the data in and compile a summary dataframe, similar to the lecture examples but changing the folder to "Pages" and the file names to the ones above. The dataframe should have 6 columns / variables (one for each file above and one for the sum of daily hits) and 167 rows (one for each page). [**Hint**: you should have already created the same dataframe for tutorials 6 & 9.]

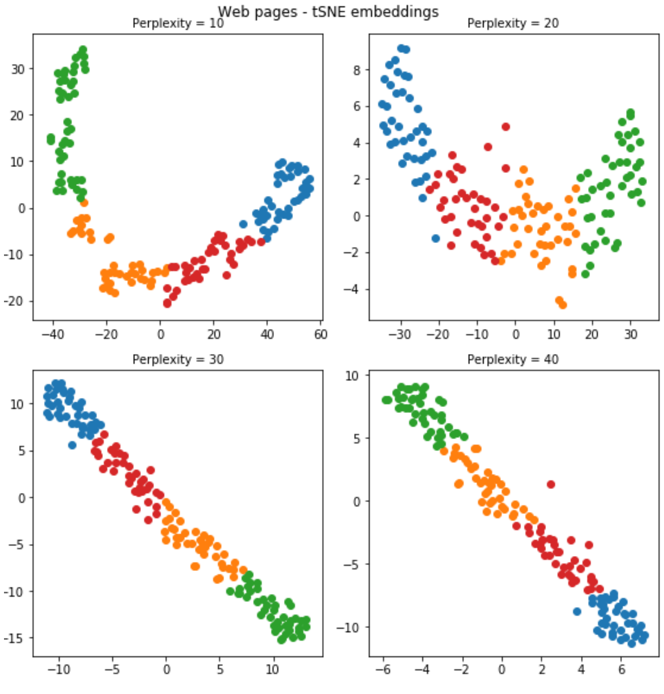
2. Last week you should have found that the elbow method suggests 4 clusters. So cluster the data using k-means with k = 4 and then create an embedding into 2 dimensions using t-SNE. Now experiment with the perplexity values to generate 4 scatter plots each with different perplexity value but showing broadly similar structures. [Hint: there are no correct answers but generally a wider range of perplexity values is better as it demonstrates that the structure is fairly robust. In fact if all the shapes are exactly the same that indicates that you may not have explored a wide enough range.]

3. Now drop the high and medium volume pages and repeat the exercise. Do you consider that the t-SNE algorithm is "seeing" the same structure as the k-means clustering and, if so, why?

   [**Hint**: to drop the high and medium volume pages use `raw_data = raw_data.drop(index=rows)`, where `rows` is a list of high and medium volume pages.]

   Your two visualisations should look something like those below.





Exploration B

Once again the second exploration looks at the Iris dataset (see Tutorial 04 for more details).

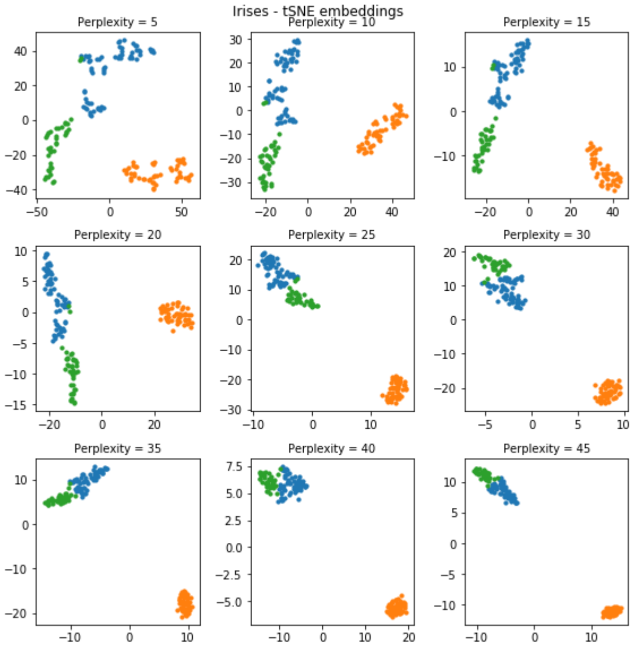
This week you will look at what t-SNE makes of the dataset.

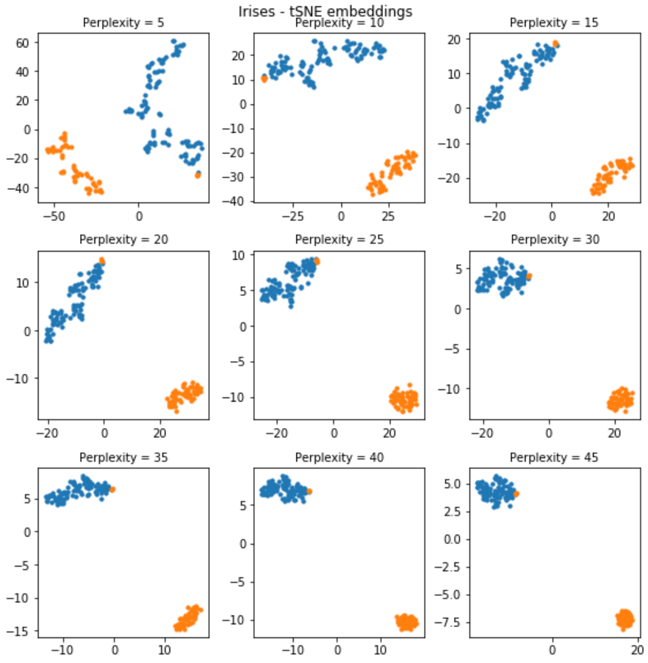
1. As previously, the data in this dataset has no obvious index column (recall that an index column usually has a unique identifier in each row). So first read the data in and check it is in the right format – there should be 5 columns (4 for the measurements and 1 for the variety) and 150 rows. [**Hint**: don't include the parameter `index_col=0` when you read it in.]

2. First create a k-means clustering with k = 3 and then, using a range of 9 different perplexities, 5, 10, 15, …, 45 compute t-SNE embeddings into 2 dimensions for the 4 measurements, 'sepal.length', 'sepal.width', 'petal.length', 'petal.width'. Comment on whether the k-means clustering generally "agrees" with the t-SNE embedding. [**Hint**: you will need to adjust subplot grid from 2 x 2 to 3 x 3.]

3. In fact, from last week's tutorial, the silhouette method suggested just 2 clusters. So now create a k-means clustering with k = 2 and compute another 9 t-SNE embeddings with the same perplexity values as above. You will probably notice that some of the irises which have been put into the orange cluster by the k-means algorithm appear to have been grouped with the blue structure by the t-SNE algorithm. Which algorithm do you think has done a better job and why (i.e. has k-means put them in the wrong cluster or has t-SNE embedded them in the wrong place)? [Hint: we know that there are 50 entries for each different type of iris so you could justify your answer by printing out the size of each cluster.]

   Your two visualisations should like something like the ones below:


Irises - tSNE embeddings
Perplexity = 5
Perplexity = 10
Perplexity = 15
Perplexity = 20
Perplexity = 25
Perplexity = 30
Perplexity = 35
Perplexity = 40
Perplexity = 45


Irises - tSNE embeddings
Perplexity = 5
Perplexity = 10
Perplexity = 15
Perplexity = 20
Perplexity = 25
Perplexity = 30
Perplexity = 35
Perplexity = 40
Perplexity = 45