STA 141C: Homework 2

- Homework due in Canvas: 05/12/2019 at 11:59PM. Please follow the instructions provided in Canvas about homeworks, carefully.

1. **Unsupervised Learning Basics: 5 points**. Answer true or false for each of the question below and give justification.

   (a) k-means is a linear dimension reduction technique.

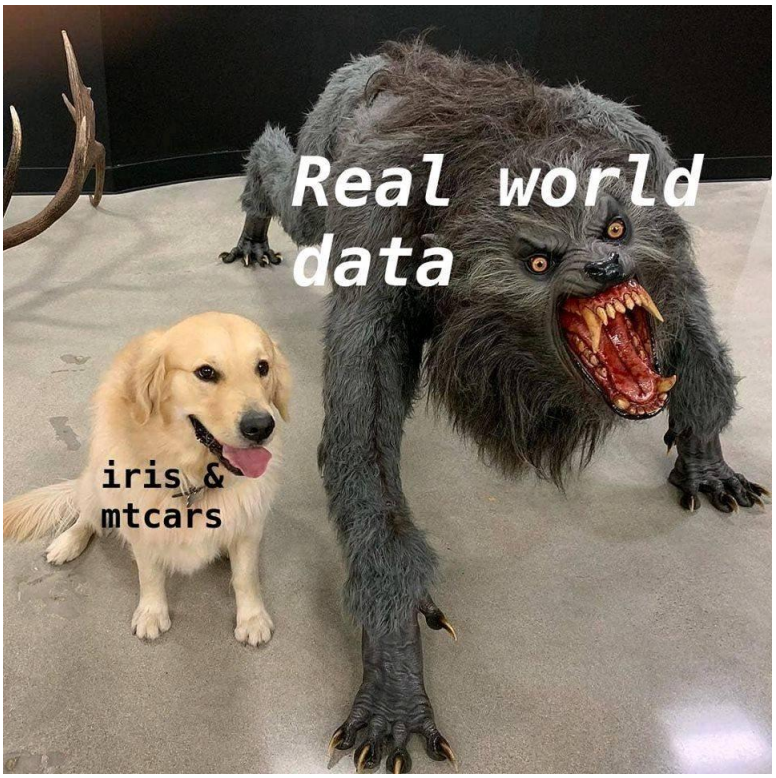
   (b) Kernel PCA is a linear dimension reduction technique.

   (c) k-means could be performed only on two-dimensional data.

   (d) Spectral Clustering is a non-linear dimension reduction technique.

   (e) All clustering techniques are based on Eigenvector Decompositions.

2. **Amazon Review Analysis** In class we used IRIS and MNIST data to visualize several dimensionality reduction techniques. But the difference between such datasets and real-data could be summarized nicely as below:



   In this question, we will take the raw reviews from Amazon and go through several steps to extract Document-Term matrix and TF-IDF matrix representation of the documents (each review is defined as a document). This process will result in a matrix of size `number of documents` x `number of words in our dictionary considered`. After this, we will try out different dimension reduction techniques on this dataset.

(a) The dataset `Amazon.RData` consists of real reviews of different products in Amazon. Unfortunately, it is provided to you in `Rdata` format (the preferred data format for `R programming language`). This scenario is quite common in practice. To process this data, you need to load the data in python. In order to proceed, install `pyreadr` package in Python. Note: to install with pip, use `pip install pyreadr`. You are welcome to explore any other ways of importing this data into Python. After loading the data, you will use only the `review` field in this question.

(b) The next pre-processing step we consider is called as **stemming**. This process fixes the words in our dictionary. For example, some common words (like 'the', 'and') are ignored, numbers are ignored, the 'root' of the word is used (i.e., running, ran are all treated as a single word run). To perform this step, execute the following commands:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
from nltk.stem.snowball import FrenchStemmer
stemmer =FrenchStemmer()
analyzer= CountVectorizer().build_analyzer()
def stemmed_words(doc):  return (stemmer.stem(w) for w in analyzer(doc))
```

(c) Now we will extract the Document-Term matrix of this dataset.This will help build a matrix where each row represents one document and each column represents a different word. To do so, use the command `CountVectorizer`. For the analyzer, use `stemmed_words` option. This will give you a matrix which corresponds term counts in each document.

(d) Next, we will extract the TF-IDF matrix of this dataset. To do so, use the command `TfidfVectorizer` with the option `token_pattern='[a-z]{3,15}'`. This will give you an alternate representation of the same dataset.

(e) How many rating values are present in the dataset ? How many reviews of each rating value are there in the entire dataset? You can think of the dataset as having roughly as many clusters as the number of rating values.

(f) Now perform, (i) kernel PCA, (2) Multi-Dimensional Scaling, (3) Isomap and (4) t-SNE on the Document-Term representation and TF-IDF representation respectively, all with number of components being set to 2. Note that, you might have to set other parameters as well for some of the above methods – you are welcome to explore different options. Produce the best figure (for each of the above method) that identifies the cluster structure (if it can) after dimension reduction (to 2 dimensions).