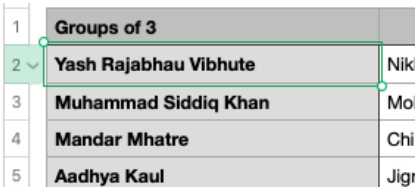
P556 Homework 3

Fall 2022

Due on Nov 20st, 11:59pm

The goal of this homework is to expose you to various python packages and functions. You can use whatever packages and functions you want to complete the following questions.

All homework assignements need to be submitted to IU github as a jupyter notebook (.ipynb). To make things easier, please name your submission with prefix "hw1_groupx_xx_xx_xx.ipynb". (Use the latest version of group assignment csv file. The group id is the index for your group assignment, see figure below. xx indicates each of your group member's last name). All tasks can be done in a single jupyter notebook.



	Groups of 3	
2	Yash Rajabhau Vibhute	Nik
3	Muhammad Siddiq Khan	Mo
4	Mandar Mhatre	Chi
5	Aadhya Kaul	Jig

Figure 1: Use the group index indicated in the group csv file. The group id starts from 2 as shown in this picture.

Question 1: Principle Component Analysis for Data Compression (30 points)

In our lecture, we have discussed about Principle Component Analysis (PCA), which can be used in multiple different applications, like data projection for dimensionality reduction. Here we will use PCA for image compression and reconstruction. Below is a picture for the Natural Park of Montseny. Can you use PCA to extract the components of the image and reconstruct the image with the first 100 components, 200, 300 components or even more? And also make a plot for accumulative variance (y-axis) with the number of principle components (x-axis). (You need to implement by yourself, not calling a function.)

Figure 2: Figure for q1.

Question 2: Naive Bayes (40 points)

During our lecture, we have talked about Naive Bayes. But we didn't got a change to go deep how to implement Naive Bayes for a real world application. First I encourage you to read through the additional learning material with different probability models. Then try to solve the following problem:

People's name can be connected to which country he/she comes from. Here we have 4000 (fake) names: Japanese, American, Arabic, and Greek. Implement a NB classifier that can make a prediction given a new name. Hint:

- First read through the additional material and think about which probabilistic model you would like to choose.
- Merge all the names and split them into training (70%) and testing (30%) with shuffle = True.
- You can use CountVectorizer from sklearn.feature_extraction.text to vectorize your input names as a preprocessing step.
- With vectorized representation of your input, then you can implement the algorithm.

Then report your testing accuracy with your algorithm.