BIG DATA PROJECT 1(PCA)

Project outline(translated from Greek)

Author:
Benekos Konstantinos, 1067754, up1067754@ac.upatras.gr

March 15, 2024

Subject 2

Along with the assignment, you will find two publications (files paper1.pdf and paper2.pdf). After you read these publications, write a brief summary (not more than 750-800 words) for each one.

Subject 3

I) The program you will write in R should execute the Principal Component Analysis algorithm on the Cars93 dataset, provided ready-made by the MASS library in R. Perform PCA using the covariance matrix on the Cars93 dataset.

II) The program you will write in Python should execute the PCA algorithm on the "Wine Quality Data Set". To download the dataset, you should visit the website Wine+Quality and download the data file named winequality-white.csv, which contains observations for the quality of 4898 white wines. Your Python code must necessarily use the pandas library to read the data file. For the creation of your program, use the implementation of the PCA algorithm available in the scikit-learn library, after ensuring that it has been installed on your system.

Both programs (R and Python) should, after executing the PCA algorithm on the datasets mentioned above, display the following elements on the screen: a) All eigenvalues of the eigenvectors resulting from the execution of the PCA method. b) All eigenvectors resulting from the execution of the PCA method. c) The percentage of variance explained by each principal component resulting from the PCA algorithm.

Subject 4

Write in R and Python a function named euclideanDistance, which takes as input two vectors and displays on the screen the Euclidean distance of the two vectors of dimension n, provided as an argument. Consider that the vectors have only real values. Additionally, the following are requested: I) For each pair of vectors x and y displayed below, calculate their Euclidean distance using the euclideanDistance function you have created in R and Python, calling it with the appropriate arguments.

- a) x=(1,2,3,4,5,6) and y= (1,2,3,4,5,6)
- b) x=(-0.5, 1, 7.3, 7, 9.4, -8.2, 9, -6, -6.3) and y=(0.5, -1, -7.3, -7, -9.4, 8.2, -9, 6, 6.3)
- c) x=(-0.5, 1, 7.3, 7, 9.4, -8.2) and y=(1.25, 9.02, -7.3, -7, 5, 1.3)
- d) x=(0, 0, 0.2) and y=(0.2, 0.2, 0)

Your code in both the R and Python environments should show the definition of the euclideanDistance function you have created as well as the call to the euclideanDistance function for computing the distances of the vectors provided in the prompt. Your program should display on the screen the Euclidean distance of the four pairs of vectors provided above. II) The following table is given, which displays the user profiles (rows) of a mobile telecommunications company. The "User" column contains a unique code for each user:

ΑΑ χρήστη	Διάρκεια ομιλίας (σε λεπτά)	Πλήθος SMS που έχουν σταλεί	Χρήση Internet (σε MB)
1	25000	14	7
2	42000	17	9
3	55000	22	5
4	27000	13	11
5	58000	21	13

I) Using the function for calculating Euclidean distance euclidean Distance that you have created in the environment of R and in Python, perform the necessary computations (in the environment of R and Python) to find the profile of the user that most resembles the profile of the user with code 5.

II) From the response and the values you found in question I), what would you answer if someone asked you the following: 'Which values have a greater influence on the value of the Euclidean distance that arises in each case?'

Subject 5

Write in the R and Python environment a function named cosineSimilarity which accepts as input two vectors and calculates or returns the cosine similarity of the two vectors. Consider that the vectors have only real values. Additionally, the following are requested: For each pair of vectors x and y displayed below, calculate the cosine similarity of the vectors mentioned in each case, using the cosineSimilarity function you have defined, calling it with the appropriate arguments in the R and Python environment. Your program should also display on the screen the cosine similarity of the vectors.

- a) x=(9.32, -8.3, 0.2) and y=(-5.3, 8.2, 7)
- b) x=(6.5, 1.3, 0.3, 16, 2.4, -5.2, 2, -6, -6.3) and y=(0.5, -1, -7.3, -7, -9.4, 8.2, -9, 6, 6.3)
- c) x=(-0.5, 1, 7.3, 7, 9.4, -8.2) and y=(1.25, 9.02, -7.3, -7, 15, 12.3)
- d) x=(2, 8, 5.2) and y=(2, 8, 5.2)

In the R and Python code you will provide, the definition of the cosineSimilarity function should be visible, as well as the call to the cosineSimilarity function with the appropriate arguments to calculate the distances of the vectors given in the statement.

Subject 6

Write in the R and Python environment a function named nominalDistance, which accepts two vectors as input and calculates their distance. The vectors given as input to the functions will have only nominal values. Define the nominalDistance function in R and Python as you deem appropriate. Then, for each pair of vectors x and y displayed below,

calculate their distance using the nominalDistance function defined in R and Python, calling it with the appropriate arguments in the R environment.

- a) x=("Green", "Potato", "Ford") and y=("Tyrian purple", "Pasta", "Opel")
- b) x=("Eagle", "Ronaldo", "Real madrid", "Prussian blue", "Michael Bay") and y=("Eagle", "Ronaldo", "Real madrid", "Prussian blue", "Michael Bay")
- c) x=("Werner Herzog", "Aquirre, the wrath of God", "Audi", "Spanish red") and y=("Martin Scorsese", "Taxi driver", "Toyota", "Spanish red")

Your programs in R and Python that you will create should display on the screen the distance of each pair of vectors using the nominalDistance function you have defined.