Machine Learning

KNNC Olivetti Faces Data – Dimensionality Reduction using Principal Components

This is a classification implementation using KNNC.
(a) Download the Olivetti faces dataset. There are 40 classes (corresponding to 40 people), each class having 10 faces of the individual; so there are a total of 400 images. Here each face is viewed as an image of size 64 × 64 (= 4096) pixels; each pixel having values 0 to 255 which are ultimately converted into floating numbers in the range [0,1]. Visit https://scikit-learn.org/0.19/datasets/olivetti_faces.html for more details.

Task 2: Here, you need to use **bootstrapping** to generate 10 more training patterns from each class (person), as follows:

(a) Let $\mathcal{X}$ be the training dataset of 400 face images.

(b) Let the set $RESAMPLES$ be empty.

(c) For each of the training patterns $X_i \in \mathcal{X}$ (for $i = 1, .., 400$) do the following:

i. Let X_i be the training pattern.

ii. Let $X_i^1, X_i^2, \cdots, X_i^P$ be the P nearest neighbors of X_i from the **remaining patterns of the same class as that of** X_i.

iii. Let

$$X_i' = \frac{1}{P+1} \sum_{j=0}^{P} X_i^j,$$

where $X_i^0 = X_i$ itself.

iv. Add X_i' to set $RESAMPLES$.

(d) Note that there are 400 patterns in $\mathcal{X}$. Obtain 400 more in $RESAMPLES$ using $P = 3$. Now update $\mathcal{X}$ as

$$\mathcal{X} = \mathcal{X} \cup RESAMPLES.$$

Task 5: In this task you are supposed to reduce the dimensionality using *l* principal components for the values of *l* = 200*;* 400*;* 600*;* 800 on the dataset obtained in task 2 step(d). Compute the *KNNC* accuracy for different values of *K* and different distances as in task 4.

CODE:

Please find the code committed for **PCA as** *KNNC_OlivettiFaceData_Task5_Dimentionality_Reduction_PCA_impl.py*

- X resampled data is first obtained using the bootstrapping method mentioned in TASK2.
- **PCA** is used to reduce the dimensionality of the data from 4096 to given values of l=200,400,600 and 800.
- With reduced dimensionality, KNNC is applied for different values of K and R and LOO accuracy is computed.

RESULT:

- The result is tabulated for all the L PCA values listed in the problem.
- Accuracy and LOO accuracy values are also tabulated and corresponding plots are generated.

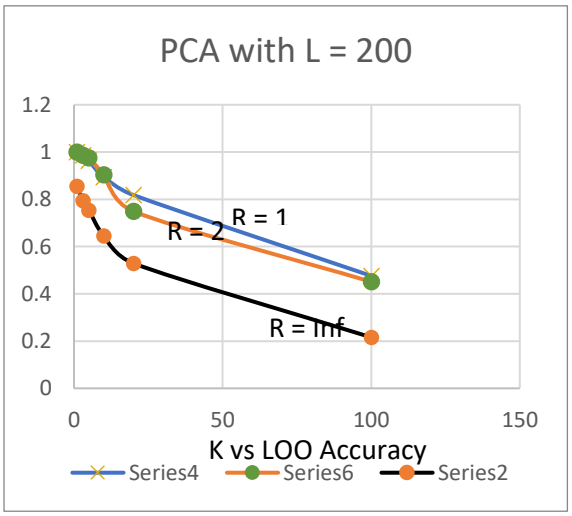
Output_file_Task5.xlsx

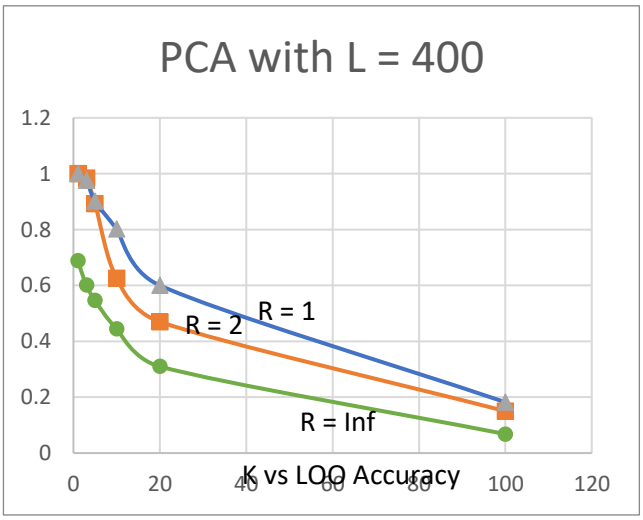
- Result sheet is attached for all values of L.

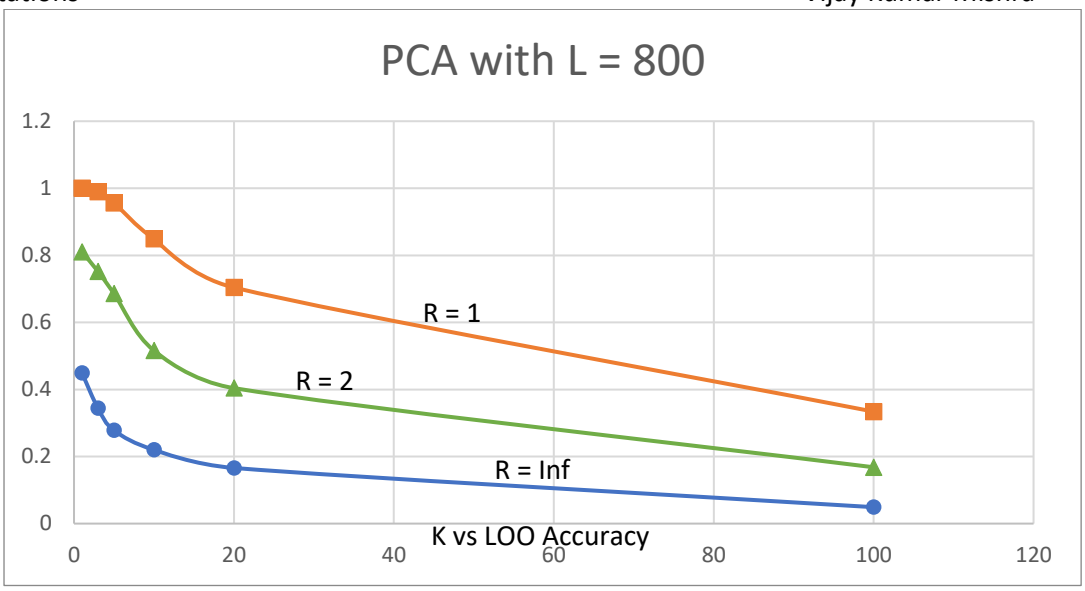
Example for L = 200

L	r - (exp value in Minkowski distance)	K	Accuracy	Leave on out Accuracy (Resamples, n=800 samples)
200	1	1	0.991667	1
		3	0.954167	0.986
		5	0.941667	0.963
		10	0.808333	0.894
		20	0.666667	0.818
		100	0.333333	0.476
	2	1	0.991667	1
		3	0.958333	0.986
		5	0.920833	0.976
		10	0.754167	0.904
		20	0.5375	0.749
		100	0.258333	0.451
	Inf	1	0.8	0.855
		3	0.708333	0.794
		5	0.629167	0.754
		10	0.533333	0.645
		20	0.404167	0.529
		100	0.1625	0.216

PLOTS:







INFERENCE/ANALYSIS:

- **The task here required to reduce the image dimensionality from 4096 to various L values like 200,400,600 and 800 Principal components.**
- PCA is implemented using sklearn and input is transformed to L values and KNNC is applied different K and R values.
- The result shows that lesser values of K (K<10) are sufficient to provide higher accuracy when PCA transformation is used on the data.
- With lesser K values, the usage will be more optimized to predict the test values.

RESOURCES USED FOR THE ASSIGNMENT:

- **Environment:**
  Anaconda, Jupyter notebook
- **Software :**
  Python
  Python libraries/modules: Pandas, Numpy, SkLearn etc