Machine Learning

Impl – Olivetti faces data

Task 1: This is a classification task 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.
(b) Use *KNNC* with values of *K* = 1*;* 3*;* 5*;* 10*;* 20*;* 100. For each value of *K*, use *KNNC* based on Minkowski distance with *r* = 1*;* 2*; 1*. Also consider fractional norms with *r* = 0*:*8*;* 0*:*5*;* 0*:*3. Compute the **percentage accuracy** using **Leaveone-out-strategy** and report results..

(a) SOLUTION

CODE:

Please find the code for Classification committed as *KNNC_OlivettiFaceData_BaseImpl.py*

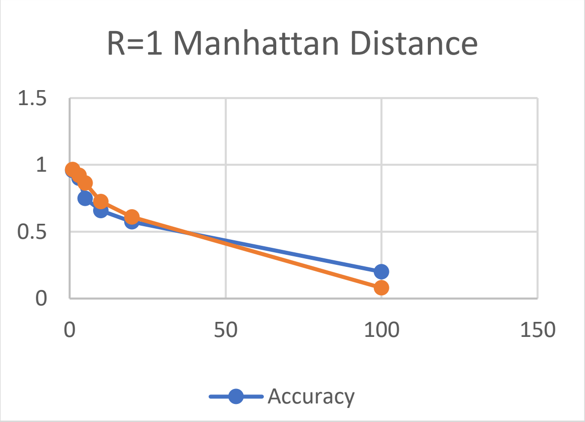
- Classification using KNNC, with r=1, 2 and infinity is obtained using Minkowski distance.
- KNN classifier is first trained using original datasets and using leave one out strategy, cross validation is done, and accuracy is calculated.
- Both Normal accuracy and accuracy using Leave one out strategy is calculated and tabulated as shown below for various values of K and r.

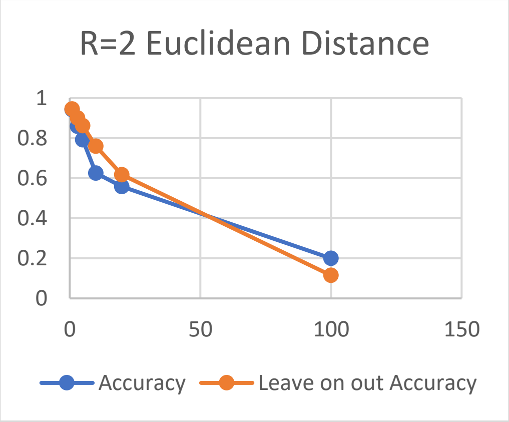
RESULT:

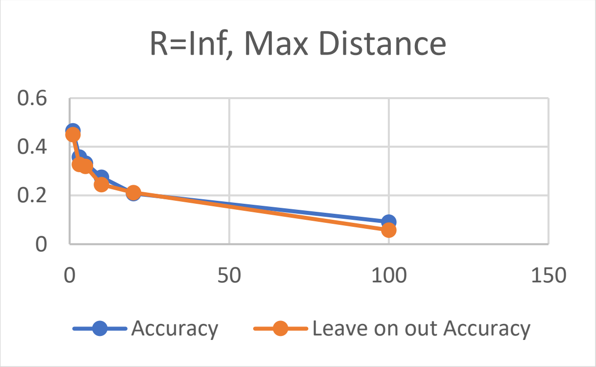
r - exp value in Minkowski distance	K - number of nearest neighbors	Accuracy	Leave on out Accuracy
1	1	0.958333	0.965
	3	0.9	0.922
	5	0.75	0.865
	10	0.658333	0.725
	20	0.575	0.61
	100	0.2	0.08
2	1	0.941667	0.945
	3	0.858333	0.9
	5	0.791667	0.863
	10	0.625	0.76
	20	0.558333	0.618
	100	0.2	0.115
Infinity	1	0.466667	0.45
	3	0.358333	0.328
	5	0.333333	0.32
	10	0.275	0.245
	20	0.208333	0.212
	100	0.091667	0.058

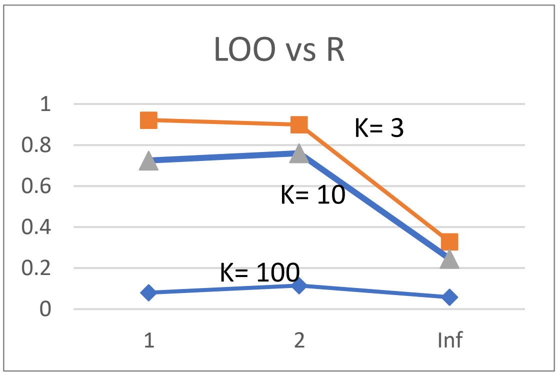
Represented accuracy in decimal format(not in %) for ease of showing on plot.

PLOT:









INFERENCE/ANALYSIS:

- **Accuracy** is the ratio of correctly **predicted data points** over **total number of test samples**

- Leave one out strategy can be used as a cross-validation strategy to validate the generated classification model. In Leave one out (LOO) method, in this example **399** samples are considered as **training set** and the classification is predicted for the **1 remaining test sample and accuracy is measured** after iterating over all samples**.**

- **Accuracy and LOO accuracy are high for lower K values (K<10). As K increases, accuracy and LOO accuracy is decreasing. This shows that when large values of K are considered for classification, the model may overfit due to many samples not belonging to same class. The domain knowledge in this question also shows that there are 10 faces of same person(class) and hence when more than 10 neighbours are considered, they likely belong to different classes and impact the model incorrectly. (Pls refer to Loo vs R graph for illustration)**

- R=1 and R=2 show nearly similar accuracy ranges, which implies that Manhattan or Euclidean distances are more appropriate for this data set than R=inf. In fact, R=1 gives even higher accuracy than R=2.

- **LOO accuracy generation takes a lot of computational resource of space and time as it has to go through each of the samples to generate accuracy and is hence not recommended when sample size is high.**
- For fractional norms r=0.8, 0.5, 0.3 – research papers are published that show that fractional norms give more meaningful results for K-means algorithms.

(Ref: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.23.7409&rep=rep1&type=pdf)

RESOURCES USED FOR THE ASSIGNMENT:

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