**Math 6350 fall 2021 MSDS Robert Azencott**
**Homework 2**
**due date sunday oct 3 , 2021 at midnight**

**Machine Learning Data Sets available on the web:**
The " university of california irvine repository of machine learning datasets " contains more that 400 publicly available data sets for machine learning. For this homework HW2, you will have to downlad from this site a large data set consisting of digitized images of standard characters typed in 153 different fonts

**Download the fonts.zip file** from https://archive.ics.uci.edu/ml/machine-learning-databases/00417/
fonts.zip contains 153 .csv files **one file for each font** with titles like AGENCY.csv, ..., TIMES.csv, etc.
Each file has a "header row" (= top row) listing the "attributes names", one name per column ( "attribute name " is equivalent to "feature name")
from *font.zip* EXTRACT four font files. Each Math6350 group should select *their own 4 font files*

**Description of a generic font data table FONT.csv :**
after the header row , each row of FONT.csv describes one single "case" (also called "instance")
the number of rows is the number of cases and will be **different for each font type**

each "case" describes a digitized image of a specific character (letter or digit) typed in specific font "FONT"
each image has size 20 x20 =400 pixels
each one of these 400 pixels has a "gray level" = integer between 0 and 255

for instance
in row # 372, the image IMG372 associated to case 372 has 400 feature. named r0c0, r0c1,r0c2, ... , r19c18, r19c19
The 400 features values for case 372 are listed in row # 372
these 400 values are the gray levels of the 400 pixels pix(i,j) of IMG372 with 0 <= i,j <= 19
in image IMG372 the pixel pix(6,13) has gray level "r6c13" which can be read in the table at ( row 372, column r6c13)

the **m_label** column : m_labels are integers, each such integer is the ""name"" of some specific character
in row 372 the m_label value is the ""name"" of the character displayed in image IMG372;

the **strength** column : strength= 0.4 for NORMAL characters ; strength= 0.7 for BOLD characters;
the **italic** column : italic= 1 for ITALIC character ; italic= 0 for NORMAL character ;

**Preliminary treatment of the data set**

in each one of your 4 font files FONT1.csv, FONT2.csv, FONT3.csv, FONT4.csv,
the header row BEGINS with the following 12 names
{ font, fontVariant, m_label, strength, italic, orientation, m_top, m_left, originalH, originalW, h, w }

**DISCARD the 9 columns listed below :**

{ *fontVariant, m_label, orientation, m_top, m_left, originalH, originalW, h, w* }
**KEEP the 3 columns {font, strength, italic}**

**KEEP also the 400 columns named r0c0, r0c1,r0c2, ... , r19c18, r19c19**
**any row containing missing numerical data should be be discarded**

define then four CLASSES CL1 CL2 CL3 CL4 of images of "normal" characters as follows
CLj = all rows of FONTj.csv for which {strength = 0.4 and italic=0}
***Display their respective sizes N1, N2, N3,N4, and the total number of cases N= n1 +n2 +n3 +n4***

The full data set (denoted DATA ) for the next questions will be the union of the four classes CL1 , CL2, CL3,CL4 and hence regroups
Case # n in DATA corresponds to a specific row "i" in the matrix DATA , and will be described by a vector of **400 feature values** , namely the 400 numbers listed in row "i" of DATA

The 400 feature values X1(i) ...X400(i) for Case # i correspond to the 400 columns
X1 = r0c0, X2 = r0c1, X3 = r0c2, ... , X399 = r19c18, X400 = r19c19
Each such feature Xj is observed N times, and its N observed values Xj(n) are listed in the column "j" of DATA.

HW2 attempts a rough automatic classification of DATA into four classes CL1 CL2 CL3 CL4

**PART 1**
1.1 Consider the feature X210. Compute its 4 means A1 A2 A3 A4 in the 4 classes CL1 CL2 CL3 CL4. Are they significantly different? Use a t-test to quantify this evaluation.
Compute and display the 4 histograms H1 H2 H3 H4 of feature X210 in the 4 classes CL1 CL2 CL3 CL4 .
Are they significantly different? Use a Kolmogorov-Smirnov KS-test to quantify this evaluation.
Interpret these results to roughly evaluate the power of X210 to discriminate between classes CL1 CL2 CL3 CL4

1.2. Compute the 400x400 correlation matrix CORR of the 400 original features X1 ...X400
Identify the 10 pairs Xi,Xj of features which have the 10 highest absolute values |CORR(i,j)|
display these 10 top highest correlation values ; identify the pixel positions corresponding to each such pair Xi and Xj

recall that gray level of pixel pix(6,13) for case #n can be read in row n of column r6c13 of DATA
explain how the correlation between the gray levels of pix(6,13) and pix(7,13) can be read from the correlation matrix CORR
compare to the correlations between pix(6,1) and pix(6,18) ; interpretation?

1.3. Compute m1 = mean(X1) .... m400 =mean(X400) and standard deviations s1 = std(X1) ....s400 = std(X400)
Standardize the features matrix DATA by centering/rescaling each feature Xj to create a new feature Yj= (Xj - mj) /sj;
the matrix DATA becomes a matrix of rescaled data SDATA, with coefficients given by
SDATA(n,j) = (DATA(n,j) - mj )/sj
Case # n is originally described by [X1(n)...X400(n)] = row "n" of DATA .
**Case #n will from now on described by**

**its vector of rescaled features Sn = [ Y1(n) , Y2(n), ..., Y400(n) ] = row"n" of SDATA**

1.4 Generate the "TrueClass" column vector TRUC of dimension N, such that TRUC(n) is the true class of Case # n
so that TRUC(n) = k whenever Case# n is in class CLk
Bind the column matrix TRUC and the rescaled data matrix SDATA to generate a "data frame" [TRUC, SDATA] of dimension (N,401)
The header row is not counted in the vertical dimension N
HW2 attempts to estimate TRUC(n) from the row vector Sn of rescaled features

**PART 2**

2.1. Among the N1 rows of class CL1 randomly select a number R1 ≃ 20% N1 of rows to define a set *testCL1* of R1 test cases ;
the remaining T1= N1-R1 ≃ 80% N1 rows of class CL1 will define a set *trainCL1* of T1 training cases .
Similarly partition each class CLj into *testCLj, trainCLj for j= 1, 2, 3, 4.* Generate the full training set TRAIN as the union of the four trainCLj for j=1,2,3,4. Generate the full test set TEST as the union of the four testCLj.
Concretely the successive random selections and unions are implemented on the full data frame [TRUC, SDATA]

2.2 Apply the k nearest neighbor (kNN) algorithm for automatic classification into the 4 classes CLj. Implement it successively for
k = 5, 10 , 15, 20, 30, 40, 50 . Use the rescaled data matrix SDATA and the TRAIN set defined above. Compute the two percentages of correct classifications $trainperf_k$ on TRAIN and $testperf_k$ on TEST. Plot the 2 curves $trainperf_k$ and $testperf_k$ versus k to try to identify a best range [a < k <b] of values for the integer k. Interpret the results.

2.3 Repeat the preceding exploration for a few more values of k within the range [a,b]. Conclude by selecting a "best" value k* for the integer k. Compute a 90% confidence interval for $testperf_k$ when k=k*.

2.4 Using the "best" k= k*, compute and interpret the 4x4 confusion matrix *testconf* on the TEST set ; make sure to compute the coefficients testconf(i,j) in row i of this matrix as percentages within class CLi;
compute and interpret confidence intervals for the 4 diagonal terms of the matrix *testconf* . Compare the performances among the 4 classes

2.5 Generate the list ERR21 , of class 2 cases which (in question 2.4 ) were **misclassified** as class 1 cases
Generate similarly the lists ERR23, ERR24 of class 2 cases misclassified as class 3 or class4
these lists will be graphically displayed in Part3.
pick 3 cases of misclassification (one in each list) and try to explain why they were misclassified , for instance by computing their closest neighbours explicitly

**Part 3**

3.1 compute the 400 eigenvalues L1 .> L2 > ... > L400 and the 400 column eigenvectors W1...W400 of the correlation matrix CORR;
denote W the 400x400 matrix with columns W1... W400; display the matrix W in an excel or csv document attached to your emailed report
Display the graph of Lm versus m

3.2 The first m eigenvectors W1... Wm provide a Percentage of Explained Variance PEV(m)
Compute PEV(m) by the formula PEV(m) = ( L1 + L2 + ... +Lm )/400
Display the graph PEV(m) versus m
Compute the **smallest integer "r" such that PEV(r) >= 90%**

3.3 Each case #n will be represented by its **first r principal components** [ PC1(n) , ..., PCr(n)]
These numbers are given by matrix products of the form [line vector]* [column vector]
PC1(n) = Sn *W1, PC2(n) = Sn*W2 , ... , PCr(n) = Sn*Wr
Case #n will now be **described by the row vector Zn = [PC1(n) ....PCr(n)] of its first "r" principal components**
the row vector Zn has dimension "r" and is given by Zn = Sn * W
Denote ZDATA the Nx400 matrix stacking up the N row vectors Z1; Z2; ... ZN
**Compute the matrix ZDATA by the matrix product** ZDATA = SDATA*W

3.4 Apply kNN classification with the same k=k* as in 2.4, and with the same TRAIN and TEST sets, but with each case #n described by the **"r" new features Zn = [PC1(n) ....PCr(n)]**
Note : this means that the matrix SDATA is replaced by ZDATA
Compute the new performance **newtestperf** on the TEST set
Compare performances as well as computing times between 2.4 and 3.4

3.5 Pick color1 color2 color3 color4, one color per class CL1 CL2 CL3 CL4
Represent each Case #n by its first 2 principal components [Z1(n),Z2(n)]
Case #n can be (very approximately) represented by the planar point Vn = [Z1(n), Z2(n)]
Display on the same planar graph the scatter plot of CL2 cases (in color2) and CL4 cases in color4
evaluate visually if these planar projections of CL2 and CL4 look easily separable or not
Implement similar displays for CL2, CL3 and CL2,CL1

3.6 On a *new graph* display as above the planar projection of CL2 in color2
add the display in color1 of all the misclassifed points listed in the list ERR21 ;
on the same figure add similar displays for the lists ERR23 and ERR24 in color3 and color 4
Interpret visually.