```
1: ////////////////////////////////////////////////////
2: How to handle with Big Data
3: ____________________________________________________
4: maXbox Starter 59 - Big Data Scientist
5:
6: As you may know big data is a term applied to data sets whose size
   or type is beyond the ability of traditional relational databases
   to capture, manage, process and visualize the data with low-
   latency.
7: Big data comes from sensors,devices, video/audio,networks,blogs,
   firmware,log files,transactional applications,web, and social
   media - much of it generated in real time and very large scale.
8:
9: For me data reduction plays a key role to handle with big data.
   Using advanced analytics techniques such as text analytics,
   machine learning, predictive analytics, data mining, statistics
   like principal component analytics, decision trees or
   collaborative filters.
10: Big data can be analyzed for insights that lead to better
   decisions, predictions and strategic business moves.
11:
12: A helpful overview of algorithms:
13: http://bigdata.black/wp-content/uploads/2016/04/machine-learning-
   algorithms-style.jpg
14:
15: While the term "big data" is relatively new, the act of gathering
   and storing large amounts of information for eventual analysis is
   ages old. Now lets practice our steps with the IRIS data set (its
   not BIG but explains data reduction):
16:
17: This is perhaps the best known database to be found in the pattern
   recognition literature. Fisher's paper is a classic in the field
   and is referenced frequently to this day. (See Duda & Hart, for
   example.) The data set contains 3 classes of 50 instances each,
   where each class refers to a type of iris plant. One class is
   linearly separable from the other 2; the latter are NOT linearly
   separable from each other.
18:
19: Attribute Information:
20:
21: 1. sepal length in cm
22: 2. sepal width in cm
23: 3. petal length in cm
24: 4. petal width in cm
25: 5. class:
26:    -- Iris Setosa
27:    -- Iris Versicolour
28:    -- Iris Virginica
29:
30: Most now agree with the characterization of big data using the 3
   V's coined by Doug Laney of Gartner:
31:
32: 1. Volume: This refers to the vast amounts of data that is
33:    generated every second/minute/hour/day in our digitized world.
34:
```

```
35: 2. Velocity: This refers to the speed at which data is being
36:    generated and the pace at which data moves from one point
37:    to the next.
38:
39: 3. Variety: This refers to the ever-increasing different forms
40:    and types that data can come in, e.g., text, images, voice,
41:    geospatial.
42:
43: Lets back to data, first we have to import and structure data:
44:
45: Const Iris_DATASET = 'C:\maXbox\maxbox3\maxbox4\iris.data';
46: http://www.softwareschule.ch/examples/iris.txt
47:
48: procedure setupMDataIrisList;
49: var stlist: TStringlist;
50:     ix, z, zi: integer;
51: begin
52:   stlist:= TStringlist.create;
53:   if fileexists(Iris_DATASET) then begin
54:      sr:= LoadFileAsString(Iris_DATASET);
55:      StrToStrings(sr,#10,stlist, true)
56:      writeln('list of observations: '+itoa(stlist.count))
57:   end;
58:     for ix:= 1 to length(x)-1 do begin
59:       zi:=1;
60:       for j:= 1 to Nvar do begin
61:         X[ix][j]:= strtofloat(copy(stlist[ix],zi,3));
62:         zi:= zi+4
63:       end;
64:     end;
65:   stlist.Free;
66: end;
67:
68: The structure we get is:
69: Const
70:   N    = 150;  { Number of observations of iris flowers}
71:   Nvar = 4;    { Number of variables }
72: { Data }
73: var X : array[1..N] of array[1..Nvar] of Float;
74:
75: >>>
76: 5.1,3.5,1.4,0.2,Iris-setosa
77: 4.9,3.0,1.4,0.2,Iris-setosa
78: 4.7,3.2,1.3,0.2,Iris-setosa
79: 4.6,3.1,1.5,0.2,Iris-setosa
80: 5.0,3.6,1.4,0.2,Iris-setosa
81: 5.4,3.9,1.7,0.4,Iris-setosa
82: 4.6,3.4,1.4,0.3,Iris-setosa
83: 5.0,3.4,1.5,0.2,Iris-setosa
84: 4.4,2.9,1.4,0.2,Iris-setosa
85: 4.9,3.1,1.5,0.1,Iris-setosa
86: .....
87:
88: Second we want to analyze the data with PCA. The goal of Principal
    Component Analysis (PCA) is to replace a set of m variables x1 x2
    xm, which may be correlated, by another set f1; f2; fm, called the
    principal components or principal factors.
```

```
 89:
 90:  These factors are independent (uncorrelated) variables.
 91:
 92:    procedure PCA(R    : TMatrix;
 93:                  Nvar   : Integer;
 94:                  Lambda : TVector;
 95:                  C, Rc  : TMatrix); external 'PCA@dmath.dll';
 96: { Performs a principal component analysis of the correlation
     matrix R }
 97:
 98: Eigenvalues of correlation matrix:
 99: >>>
100:   2.73213593141507
101:   1.10980692067043
102:   0.138229964846805
103:   0.0198271830676986
104:   Sum: 4
105:
106: Usually, the algorithm starts with the correlation matrix R which
     is a m X m symmetric matrix such that Rij is the correlation
     coeficient between variable xi and variable xj.
107: The eigenvalues 2.73213593141507; 1.10980692067043; ... m (in
     decreasing order) of matrix R are the variances of the principal
     factors.
108:  Their sum Pp i=1 of i is equal to m. So, the percentage of
     variance associated with the i-th factor is equal to i=m in our
     case = 4 eigenvalues are 4 factors.
109:
110: Now where is the reduction?
111: We have changed our original data in terms of eigenvectors. This
     will reorient the data in the direction where the data is having
     maximum variance. The first 2 eigenvalues
112: >>>
113:   2.73213593141507
114:   1.10980692067043
115:
116: explains about 0.95 % of the variance!
117:
118: This is dimension reduction. We have reduced the problem from a 4D
     to a 2D problem, getting rid of two dimensions. Reducing
     dimensions helps to simplify the data and makes it easier to
     visualise.
119: It is also necessary to normalize the data in PCA because the
     motto of performing this exercise i.e. PCA is to find the
     components which show maximum variance. PCA aims to detect the
     correlation between variables. If a strong correlation between
     variables exists, the attempt to reduce the dimensionality only
     makes sense.
120:
121:
122: The script you found at:
123: http://www.softwareschule.ch/examples/813_PCA_datascience_iris3.
     txt
124: pic: http://www.softwareschule.ch/images/sierpinski4realhash.png
125:
```

```
126:
127: Ref:
128:     http://bigdata.black/featured/machine-learning-algorithms/
129:     http://bigdata.black/featured/what-is-big-data/
130:     http://www.softwareschule.ch/examples/machinelearning.jpg
131:     https://maxbox4.wordpress.com
132:
133:     https://www.linkedin.com/pulse/dimension-reduction-technique-
     principal-component-using-gandhi/
134:     https://plot.ly/ipython-notebooks/principal-component-
     analysis/
135:
136:
137:
138: Doc: DMATH PCA Stat Lib Interface:
139:
140:  function DMathFact(N: Integer): float;
141:   external 'Fact@dmath.dll';
142:
143:   function DPower(X, Y : Float): Float;
144:   external 'Power@dmath.dll';
145:
146:   procedure VecMean(X          : TMatrix;
147:                     Lb, Ub, Nvar : Integer;
148:                     M          : TVector); external
     'VecMean@dmath.dll';
149: { Computes the mean vector M from matrix X }
150:
151:   procedure MatVarCov(X          : TMatrix;
152:                       Lb, Ub, Nvar : Integer;
153:                       M          : TVector;
154:                       V          : TMatrix); external
     'MatVarCov@dmath.dll';
155:  { Computes the variance-covariance matrix V from matrix X }
156:
157:   procedure MatCorrel(V    : TMatrix;
158:                       Nvar : Integer;
159:                       R    : TMatrix); external
     'MatCorrel@dmath.dll';
160: { Computes the correlation matrix R from the var-cov matrix V }
161:
162:   procedure VecSD(X          : TMatrix;
163:                   Lb, Ub, Nvar : Integer;
164:                   M, S         : TVector); external
     'VecSD@dmath.dll';
165: { Computes the vector of standard deviations S from matrix X }
166:
167:    procedure ScaleVar(X          : TMatrix;
168:                       Lb, Ub, Nvar : Integer;
169:                       M, S         : TVector;
170:                       Z          : TMatrix); external
     'ScaleVar@dmath.dll';
171: { Scales a set of variables by subtracting means and dividing by
     SD's }
172:
```

```
173:    procedure PCA(R    : TMatrix;
174:                  Nvar   : Integer;
175:                  Lambda : TVector;
176:                  C, Rc  : TMatrix); external 'PCA@dmath.dll';
177: { Performs a principal component analysis of the correlation
     matrix R }
178:
179:    procedure PrinFac(Z          : TMatrix;
180:                      Lb, Ub, Nvar : Integer;
181:                      C, F         : TMatrix); external
     'PrinFac@dmath.dll';
182: { Computes principal factors }
183:
184:
185:  Abstract:
186:  The sheer size of data in the modern age is not only a challenge
     for computer hardware but also a main bottleneck for the
     performance of many machine learning algorithms. To annotate it
     directly, PCA basically strips off the redundant parts of the data
     keeping the vital components.
187:
188:
189:  The famous "Iris" dataset that has been deposited on the UCI
     machine learning repository
190:
191:  https://archive.ics.uci.edu/ml/datasets/Iris
192:
193: The iris dataset contains measurements for 150 iris flowers from
     three different species.
194: The three classes in the Iris dataset are:
195:
196:     Iris-setosa (n=50)
197:     Iris-versicolor (n=50)
198:     Iris-virginica (n=50)
199:
200: And the four features of in Iris dataset are:
201:
202:     sepal length in cm
203:     sepal width in cm
204:     petal length in cm
205:     petal width in cm
```