Maral Nourimand
16.11.2023
Data Mining Exercise Week 03

Question #1

```
% Import the data from the Excel file
[~, ~, raw] = xlsread('bloodp.xlsx');

% Extract systolic (sbp) and diastolic (dbp) blood pressure values
sbp = cell2mat(raw(2:end, 1));  % 3873x1 double
dbp = cell2mat(raw(2:end, 2));  % 3873x1 double


% calculate the mean value
sbp_mean = round(mean(sbp(sbp > 0), 'omitnan'))  % 146
dbp_mean = round(mean(dbp(dbp > 0), 'omitnan'))  % 83


% replace zero values with mean values
sbp(sbp == 0) = sbp_mean;
dbp(dbp == 0) = dbp_mean;

% replace missing values with mean values
sbp(isnan(sbp)) = sbp_mean;
dbp(isnan(dbp)) = dbp_mean;


% Correct erroneous values
sbp(sbp < 80) = sbp(sbp < 80) * 10;
dbp(dbp < 40) = dbp(dbp < 40) * 10;


% Remove values that are impossible

% first find the problematic rows in
% both sbp and dbp
logical_array = (sbp > 300 | dbp > 160);

% then remove the rows which fulfill the condition(sbp > 300 or dbp > 160)
sbp(logical_array == 1) = [];
dbp(logical_array == 1) = [];

% corrected data
corrected_data = [sbp, dbp];
```

Corrected Blood Pressure Data:

```
  146   83
  146   83
```

```
  146   83
  146   83
  146   83
  146   83
  146   83
  146   83
  146   83
  115   85
  115   90
  146   83
  120   85
  125   80
...
```

Question #2

```
% create the observation matrix O
O = [ones(size(sbp)), sbp, dbp];

% select y=dbp and X=[1 sdb]
y = O(:, 3);
X = O(:, [1, 2]);

% compute coefficients manually
coefficients = (X' * X) \ (X' * y);

disp('Coefficients (Using manual Computation):');
disp(coefficients);


% use the regress function
coefficients_regress = regress(y, X);

disp('Coefficients (Using regress function):');
disp(coefficients_regress);
```

Coefficients (Using manual Computation):

```
  40.5814
   0.3079
```

Coefficients (Using regress function):

40.5814

0.3079

Question #3

```
% loading data given
S = {'word1', 'word2', 'word3', 'word4', 'word5'};
Fo = [15, 7, 6, 11, 4];
Nw = 500;

Fo1 = [1, 4, 3, 3, 6];
Nw1 = 200;

Fo2 = [20, 1, 5, 16, 9];
Nw2 = 210;

% normalize word occurrences
normalized_Fo = Fo / Nw;
normalized_Fo1 = Fo1 / Nw1;
normalized_Fo2 = Fo2 / Nw2;

% calculate the cosine distance
% dot = dot product of two vectors
% norm = Euclidean norm (magnitude) of a vector
cosine_distance_1 = 1 - dot(normalized_Fo, normalized_Fo1) / (norm(normalized_Fo) *
norm(normalized_Fo1));
cosine_distance_2 = 1 - dot(normalized_Fo, normalized_Fo2) / (norm(normalized_Fo) *
norm(normalized_Fo2));

disp('Cosine Distance between Reference and Document 1:');
disp(cosine_distance_1);

disp('Cosine Distance between Reference and Document 2:');
disp(cosine_distance_2);
```

Cosine Distance between Reference and Document 1:

0.3376

Cosine Distance between Reference and Document 2:

0.0599

The closer to zero, the more similar they are.

Question #4

```
% Load the dataset from the Excel file
power_data = xlsread('Tetuan City power consumption.csv');


% Binarize all variables
mean_values = mean(power_data);
binarized_data = power_data >= mean_values;

% Sample data
s = [0, 1, 0, 0, 0, 0, 0, 0];

% Binarize the sample
binarized_s = s >= mean_values;

% Calculate Hamming distance
% (the variable names are stored in the first row of the Excel sheet)
hamming_distances = sum(binarized_data ~= binarized_s, 2);

% Find the index of the nearest neighbor
[~, nearest_neighbor_index] = min(hamming_distances);  % the row index in power_data

% Display the nearest neighbor
nearest_neighbor = power_data(nearest_neighbor_index, :);
disp('Nearest Neighbor:');
disp(nearest_neighbor);
```

Nearest Neighbor:

1.0e+04 *

0.0015	0.0058	0.0000	0.0180	0.0060	2.8885	1.7770	1.6476

Question #5

```
% Correlation for binarized_data
R = corrcoef(binarized_data);
```

R =

1.0000	-0.2186	0.4078	0.2909	0.1929	0.2748	0.2662	0.3315
-0.2186	1.0000	-0.0851	-0.3506	-0.2082	-0.2095	-0.1964	-0.1063
0.4078	-0.0851	1.0000	0.1077	0.0350	0.1101	0.1233	0.2198
0.2909	-0.3506	0.1077	1.0000	0.6256	0.2151	0.1711	-0.0201
0.1929	-0.2082	0.0350	0.6256	1.0000	0.1838	0.1121	-0.0216
0.2748	-0.2095	0.1101	0.2151	0.1838	1.0000	0.6467	0.5166

```
  0.2662  -0.1964   0.1233   0.1711   0.1121   0.6467   1.0000   0.4204
  0.3315  -0.1063   0.2198  -0.0201  -0.0216   0.5166   0.4204   1.0000
```

Since I'm not sure whether binary correlation might differ from the normal correlation, I used another function for calculation, too. (bitxor in MATLAB, to calculate the XOR between the binary values of each pair of variables.)

```
% calculates the number of variables (columns) in the binarized dataset

num_variables = size(binarized_data, 2);
binary_correlation = zeros(num_variables);

for i = 1:num_variables
    for j = 1:num_variables
        % Calculate binary correlation using XOR
        binary_correlation(i, j) = sum(bitxor(binarized_data(:, i), binarized_data(:,
j)));
    end
end
```

```
Binary Correlation Matrix:
          0       31926       15779       19020       21505       19005       19233       17717
      31926           0       28837       35520       32023       31699       31403       29371
      15779       28837           0       21609       23146       23350       22868       19440
      19020       35520       21609           0        8553       20839       21705       24727
      21505       32023       23146        8553           0       21672       23168       24538
      19005       31699       23350       20839       21672           0        9262       12952
      19233       31403       22868       21705       23168        9262           0       15278
      17717       29371       19440       24727       24538       12952       15278           0
```

```
% Identify variables with the highest correlation, know the column and row
[max_correlation, ind] = max(binary_correlation(:));
[row, col] = ind2sub(size(binary_correlation), ind);  % 4 , 2

Variables with the Highest Binary Correlation:
Variable 4 and Variable 2 with Correlation 35520
```

Question #6

A distance measure is metric when it satisfies the following conditions:

1. positivity
2. reflexivity
3. symmetry
4. triangle inequality

Assume that we have 2 vectors:

A = [1, 0]

B = [−1, 0]

If calculate the cosine distance between them:
Cosine Similarity(A,B)=−1 which violates the first condition(positivity) and cannot be a metric.