ex2-system-evaluation

October 7, 2018

1 Practical work 3

Author: Romain Claret ## Exercice 2

a) Classify

```
In [1]: import pandas as pd
        dataset_classes = ['x0','x1','x2','x3','x4','x5','x6','x7','x8','x9','y','drop']
        dataset = pd.read_csv('ex2-system-a.csv', delimiter=';', header=None, names=dataset_cla
        dataset.drop('drop', axis=1, inplace=True)
        #print(dataset.head(3))

In [2]: import numpy as np
        def bayes_classify(data_previous):
            return np.argmax(data_previous)
```

b) error rate

```
In [3]: def get_error(dataset):
            wins = 0
            for i in range(dataset.shape[0]):
                row = dataset.loc[i:i].values[0]
                guess = bayes_classify(row[0:-1])
                if guess == row[-1]:
                    wins = wins + 1
            return (dataset.shape[0] - wins)/dataset.shape[0]

        print("error rate: ", get_error(dataset))
```

```
error rate:  0.1073
```

c) Compute confusion matrix

```
In [4]: def get_confused_matrix(dataset):
            features = dataset.shape[1]-1
            confused_matrix = np.zeros((features,features),dtype=int)
```

```
            for i in range(dataset.shape[0]):
                row = dataset.loc[i:i].values[0]
                guess = bayes_classify(row[0:-1])
                truth = np.int(row[-1])
                confused_matrix[truth][guess] = confused_matrix[truth][guess]+1

            return confused_matrix

        print(get_confused_matrix(dataset))
```

```
[[ 944    0   11    0    0    2   10    7    5    1]
 [   0 1112    2    3    1    4    3    1    9    0]
 [  10    6  921   12   15    3   19   15   26    5]
 [   1    1   31  862    2   72    5   14   12   10]
 [   2    3    6    2  910    1   12    6    4   36]
 [  12    3    6   29   19  768   19    9   21    6]
 [  14    3   21    2   22   28  865    0    3    0]
 [   0   14   30    9    7    2    1  929    3   33]
 [  12   16   18   26   24   46   22   19  772   19]
 [  10    4    6   22   53   18    0   48    4  844]]
```

d) best classes

```
In [5]: def get_precision_and_recall(confused_matrix):
            TP = [confused_matrix[i,i] for i in range(confused_matrix.shape[0])]
            FN = [sum(confused_matrix[i,:]) - TP[i] for i in range(confused_matrix.shape[0])]
            FP = [sum(confused_matrix[:,i]) - TP[i] for i in range(confused_matrix.shape[0])]

            precision = [TP[i]/(TP[i] + FP[i]) for i in range(confused_matrix.shape[0])]
            recall = [TP[i]/(TP[i] + FN[i]) for i in range(confused_matrix.shape[0])]

            return [precision,recall]

        confused_matrix = get_confused_matrix(dataset)
        result = get_precision_and_recall(confused_matrix)
        print("worst precision:", np.argsort(result[0])[-1])
        print("worst recall:", np.argsort(result[1])[-1])

        print("best precision:", np.argsort(result[0])[0])
        print("best recall:", np.argsort(result[1])[0])
```

```
worst precision: 1
worst recall: 1
best precision: 5
best recall: 8
```

e) do the same with ex1-system-b.csv

```
In [6]: dataset_classes_b = ['x0','x1','x2','x3','x4','x5','x6','x7','x8','x9','y','drop']
        dataset_b = pd.read_csv('ex2-system-b.csv', delimiter=';', header=None, names=dataset_
        dataset_b.drop('drop', axis=1, inplace=True)

In [7]: def get_f1(precision, recall):
            return np.mean([2 * (precision[i] * recall[i]) / (precision[i] + recall[i]) for i :

In [8]: confused_matrix_b = get_confused_matrix(dataset_b)
        result_b = get_precision_and_recall(confused_matrix_b)
        print("worst precision:", np.argsort(result_b[0])[-1])
        print("worst recall:", np.argsort(result_b[1])[-1])

        print("best precision:", np.argsort(result_b[0])[0])
        print("best recall:", np.argsort(result_b[1])[0])
```

```
worst precision: 1
worst recall: 1
best precision: 3
best recall: 5
```

```
In [9]: print("dataset_a f1 score:",get_f1(result[0], result[1]))
        print("dataset_b f1 score:",get_f1(result_b[0], result_b[1]))

        print("dataset_a error:",get_error(dataset))
        print("dataset_b error:",get_error(dataset_b))
```

```
dataset_a f1 score: 0.8907308492877297
dataset_b f1 score: 0.9608568150389065
dataset_a error: 0.1073
dataset_b error: 0.0387
```

Based on those results, dataset_b looks better than dataset_a