

MLG Pratical Work 01

Rapport

Sathiya Kirushnapillai & Mathieu Monteverde

3. Quick numpy introduction

Max of each row and max of each column

Code

On utilise la fonction **amax** pour récupérer un tableau des plus grands éléments selon un axe donné.

```
print("The max of each row is {}".format(np.amax(array1, axis = 1)))  
print("The max of each column is {}".format(np.amax(array1, axis = 0)))
```

Output

```
The max of each row is [3 6]  
The max of each column is [4 5 6]
```

Print the elements of array1 that are less than 4

Code

On crée un masque sur la condition **< 4** et on l'applique ensuite au tableau.

```
mask = array1 < 4  
print("Elements of array1 that are < 4: {}".format(array1[mask]))
```

Output

```
Elements of array1 that are < 4: [1 2 3]
```

Convert the values of array2 to the range [0, 1]

Code

On utilise ici la valeur maximum du tableau pour normaliser les valeurs du tableau entre 0 et 1 (valeur max \rightarrow 1, 0 \rightarrow 0).

```
max = np.amax(array2)
ranger = np.vectorize(lambda t : t/max)
ranger(array2)
```

Output

```
array([0.55555556, 0.88888889, 0.22222222, 1.         , 0.44444444,
       0.33333333])
```

5. Quick Introduction to Pandas

Code

```
print(df.groupby('class').describe()['alcohol'])
```

Output

	count	mean	std	min	25%	50%	75%	max
class								
1	59.0	13.744746	0.462125	12.85	13.400	13.750	14.100	14.83
2	71.0	12.278732	0.537964	11.03	11.915	12.290	12.515	13.86
3	48.0	13.153750	0.530241	12.20	12.805	13.165	13.505	14.34

Classe	% d'alcool en moyenne	Médiane % d'alcool	Max % d'alcool	Min % d'alcool
1	13.74	13.75	14.83	12.85
2	12.28	12.29	13.86	11.03
3	13.15	13.17	14.34	12.20

6. Visual exploratory analysis of data

Q: By looking at the boxplot, which features seems the most discriminative ?

Il est difficile de désigner un trait de vin pour lequel les classes sont absolument distinctes. Une boîte à moustache permet de visualiser, pour une distribution de valeurs donnée, si une valeur peut être considérée comme aberrante ou non (respectivement en dehors ou en dedans des deux moustaches). Le cas idéal que l'on rechercherait serait donc un trait de vin pour lequel les moustaches des boîtes à moustaches de chaque classe forment des plages qui ne se chevauchent pas.

Il est évident que nous n'avons pas trouvé un tel trait, nous allons donc considérer les traits dont les boîtes de la boîte à moustache ne se chevauchent pas (ou peu).

On peut donc retenir par exemple les traits de vin suivants comme relativement distincts entre les différentes classes :

- flavanoids
- proline
- color intensity

7. Classifying the wine data

Code

```
len(df.loc[df['class'] == df['prediction']]) / len(df)
```

Output

```
0.33707865168539325
```

En utilisant la formule proposée, on obtient une performance de environ 38% prédictions correctes. Ce n'est pas très bon.

8. Performance evaluation

```
pred = []

for row in df['flavanoids']:
    if row >= 2.5:
        pred.append(1);
    elif row > 1.8 and row < 2.5 :
        pred.append(2);
    else:
        pred.append(3)

# A new column is added to the dataframe
df['prediction'] = pred
```

Q: Compute the confusion matrix of the resulting rule-based system ?

Populating the interactive namespace from numpy and matplotlib

	precision	recall	f1-score	support
1	0.75	0.90	0.82	59
2	0.81	0.37	0.50	71
3	0.64	1.00	0.78	48
avg / total	0.74	0.71	0.68	178

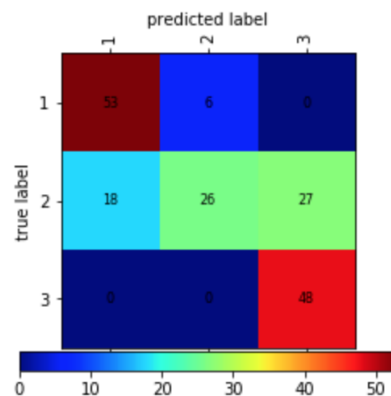


Illustration 1: Matrice de confusion pour le système basé sur les règles du point précédent

Q: Compute the precision, the recall and the f1-score of the system for a given class using the values of the confusion matrix ?

```

tp = C[0, 0]
tn = C[1, 1] + C[2, 2]
fp = C[1, 0] + C[2, 0]
fn = C[0, 1] + C[0, 2]

precision = tp / (tp + fp)
print('precision: ', precision)

recall = tp / (tp + fn)
print('recall: ', recall)

f1_score = 2 * precision * recall / (precision + recall)
print('f1-score: ', f1_score)

```

À l'arrondi à deux décimales près, on voit qu'on obtient les mêmes résultats que le tableau ci-dessus en calculant à la main.

precision: 0.75

recall: 0.90

f1-score: 0.82