

## **INFORME NLP - Pràctica 2**

Per a dur a terme la pràctica de Gender Identification, un subproblema dins el camp d'Author Profiling, hem seguit els passos marcats a l'enunciat de la pràctica:

- **N paraules més freqüents:**

Disposem d'un corpus de 1260 textos escrits per homes o dones. El nom d'aquests fitxers correspon al label d'identificació per l'algoritme de machine learning: *1\_female* on cada label indica el número del text i si aquest ha sigut escrit per un home o una dona.

L'objectiu d'aquest apartat és extreure les N paraules que s'utilitzen més en tot aquest corpus de textos.

Degut a que hi ha molts tipus diferents de paraules, i moltes d'elles són articles, pronoms febles i paraules amb poca rellevància, utilitzem un fitxer "stopwords.txt", per a detectar aquestes paraules i eliminar-les del nostre anàlisi.

Per a realitzar aquest anàlisi, passem per cada text del corpus i el parsejem per paraules. El parse es realitza en dues fases diferents, la primera es centra en eliminar els símbols de puntuació, i la segona s'assegura que les paraules estan ben separades per a guardar-les correctament en una llista. Per finalment, guardar el recompte de paraules que cada fitxer conté. (mètode **parse\_files** de la classe **Classifier**).

Un cop obtingudes totes les paraules de cada text, obtenim les paraules més freqüents amb la instrucció següent:

```
self.most_frequent = Counter(self.most_frequent).most_common(self.N)
```

- **Càlcul dels vectors de features:**

Un cop obtingudes les N paraules més freqüents del corpus, passem a calcular el vector de features per aquestes paraules.

Passem per cada una de les paraules de cada text, i actualitzem el vector de features en relació al nombre d'ocurrències de cada paraula "freqüent" dins de cada text.

Aquest procés es veu reflexat en el mètode `compute_features` de la classe **Classifier**, on com acabem d'esmentar es calcula el vector de features per cada fitxer donades les N paraules més freqüents del corpus.

- **Pasar vectors a arff**

Utilitzarem el Weka per a executar els algorismes de machine learning que aquest programa disposa amb les dades que hem obtingut fins ara.

Per a això, hem de convertir les dades del vector de features a un fitxer amb format .arff, ja que és el format de dades que demana Weka.

Aquesta conversió té lloc al mètode `generate_arff` de la classe `Classifier`:

```
def generate_arff(self):
    """
    Generate the results file formatted with the data of the execution.

    Args:
        self.

    Returns:
        generate the file with the results in the same directory
    """
    file_name = str(self.N) + "-results_StopwordsRemoved-" + str(self.remove_stopwords) + ".arff"
    with open(file_name, "w") as results:
        results.write("%1. Title: Results of Features")
        results.write("\n%2. Sources:")
        results.write("\n%\tAuthors: Ferran Cantarino i Marc Rabat")
        results.write("\n@RELATION " + str(self.N) + "_Features")
        for item in self.most_frequent:
            aux = item
            if "'" in item:
                aux = item.replace("'", ".")
            results.write("\n@ATTRIBUTE " + str(aux) + " NUMERIC")
        results.write("\n@ATTRIBUTE class {male, female}")
        results.write("\n@DATA\n")
        for file in self.files:
            for k, v in file.features.items():
                results.write(str(str(v) + ","))
            results.write(file.gender)
            results.write("\n")
    results.close()
```

- **Avaluació de resultats per diferents N i classificadors:**

Hem variat les dades obtingudes a través de la quantitat de paraules més freqüents a escollir. Hem variat N amb valors de 5, 50, 100 i 150.

I hem executat cada una d'aquestes diferents mostres amb classificadors diferents:

## SMO:

N = 5:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      894          70.9524 %
Incorrectly Classified Instances    366          29.0476 %
Kappa statistic                    0.4183
Mean absolute error                 0.2905
Root mean squared error            0.539
Relative absolute error             58.0956 %
Root relative squared error        107.7917 %
Total Number of Instances         1260

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
          0,922    0,505    0,648     0,922    0,761      0,462    0,709     0,636    male
          0,495    0,078    0,864     0,495    0,630      0,462    0,709     0,679    female
Weighted Avg.   0,710    0,292    0,755     0,710    0,696      0,462    0,709     0,658

=== Confusion Matrix ===

  a    b  <-- classified as
583  49 |   a = male
317 311 |   b = female
```

N = 50:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1016          80.6349 %
Incorrectly Classified Instances     244          19.3651 %
Kappa statistic                    0.6126
Mean absolute error                 0.1937
Root mean squared error            0.4401
Relative absolute error             38.7304 %
Root relative squared error        88.0116 %
Total Number of Instances         1260

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
          0,851    0,239    0,782     0,851    0,815      0,615    0,806     0,740    male
          0,761    0,149    0,836     0,761    0,797      0,615    0,806     0,755    female
Weighted Avg.   0,806    0,194    0,809     0,806    0,806      0,615    0,806     0,748

=== Confusion Matrix ===

  a    b  <-- classified as
538  94 |   a = male
150 478 |   b = female
```

N = 100:

=== Summary ===

Correctly Classified Instances	1056	83.8095 %
Incorrectly Classified Instances	204	16.1905 %
Kappa statistic	0.6762	
Mean absolute error	0.1619	
Root mean squared error	0.4024	
Relative absolute error	32.3811 %	
Root relative squared error	80.4748 %	
Total Number of Instances	1260	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,837	0,161	0,840	0,837	0,838	0,676	0,838	0,785	male
	0,839	0,163	0,837	0,839	0,838	0,676	0,838	0,782	female
Weighted Avg.	0,838	0,162	0,838	0,838	0,838	0,676	0,838	0,783	

=== Confusion Matrix ===

```

a   b   <-- classified as
529 103 |   a = male
101 527 |   b = female

```

N = 150:

=== Summary ===

Correctly Classified Instances	1078	85.5556 %
Incorrectly Classified Instances	182	14.4444 %
Kappa statistic	0.7111	
Mean absolute error	0.1444	
Root mean squared error	0.3801	
Relative absolute error	28.8891 %	
Root relative squared error	76.0117 %	
Total Number of Instances	1260	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,843	0,132	0,865	0,843	0,854	0,711	0,856	0,808	male
	0,868	0,157	0,846	0,868	0,857	0,711	0,856	0,800	female
Weighted Avg.	0,856	0,144	0,856	0,856	0,856	0,711	0,856	0,804	

=== Confusion Matrix ===

```

a   b   <-- classified as
533  99 |   a = male
 83 545 |   b = female

```

Pels resultats obtinguts amb el classificador SMO, veiem un progrés positiu en l'algoritme. Només fixant-nos en les dades generals, com la precisió veiem que augmenta considerablement, augmentant un 10% de N = 5 a N 0 150, arribant a una precisió del 85,9%

Podem observar també que en el cas d'N = 5, les xifres de precisió, recall i F-measure són bastant diferents. Fet que per altres valors d'N observats en el nostre anàlisis no és així, obtenint valors pels tres tipus d'indicadors molt similars.

Aquest fet pot ser degut a que els valors d'Fp i Fn, usats per a calcular la Precisió i el Recall respectivament, són molt similars, tal i com podem veure observant la confusion matrix.

## Naive Bayes:

N = 5:

```
=== Summary ===

Correctly Classified Instances      883          70.0794 %
Incorrectly Classified Instances    377          29.9206 %
Kappa statistic                    0.4007
Mean absolute error                 0.3092
Root mean squared error             0.4565
Relative absolute error             61.8447 %
Root relative squared error         91.293 %
Total Number of Instances          1260

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,929	0,529	0,639	0,929	0,757	0,450	0,801	0,768	male
	0,471	0,071	0,868	0,471	0,611	0,450	0,801	0,822	female
Weighted Avg.	0,701	0,301	0,753	0,701	0,684	0,450	0,801	0,795	

```

=== Confusion Matrix ===

  a   b  <-- classified as
587  45 |  a = male
332 296 |  b = female

```

N = 50:

```
=== Summary ===

Correctly Classified Instances      931          73.8889 %
Incorrectly Classified Instances    329          26.1111 %
Kappa statistic                    0.4772
Mean absolute error                 0.2608
Root mean squared error             0.5009
Relative absolute error             52.1676 %
Root relative squared error        100.172 %
Total Number of Instances          1260

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,903	0,427	0,681	0,903	0,776	0,505	0,864	0,846	male
	0,573	0,097	0,855	0,573	0,686	0,505	0,859	0,849	female
Weighted Avg.	0,739	0,262	0,768	0,739	0,731	0,505	0,861	0,848	

```

=== Confusion Matrix ===

  a   b  <-- classified as
571  61 |  a = male
268 360 |  b = female

```

N = 100:

```

=== Summary ===

Correctly Classified Instances      966          76.6667 %
Incorrectly Classified Instances    294          23.3333 %
Kappa statistic                    0.5332
Mean absolute error                 0.2319
Root mean squared error             0.464
Relative absolute error             46.378 %
Root relative squared error         92.8058 %
Total Number of Instances          1260

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0,805    0,272    0,749     0,805    0,776     0,535    0,877    0,874    male
      0,728    0,195    0,788     0,728    0,757     0,535    0,876    0,849    female
Weighted Avg.   0,767    0,234    0,768     0,767    0,766     0,535    0,877    0,861

=== Confusion Matrix ===

  a    b  <-- classified as
509 123 |  a = male
171 457 |  b = female

```

N = 150:

```

=== Summary ===

Correctly Classified Instances      972          77.1429 %
Incorrectly Classified Instances    288          22.8571 %
Kappa statistic                    0.5427
Mean absolute error                 0.2278
Root mean squared error             0.4673
Relative absolute error             45.5609 %
Root relative squared error         93.4671 %
Total Number of Instances          1260

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0,813    0,271    0,751     0,813    0,781     0,545    0,874    0,859    male
      0,729    0,187    0,795     0,729    0,761     0,545    0,873    0,833    female
Weighted Avg.   0,771    0,229    0,773     0,771    0,771     0,545    0,873    0,846

=== Confusion Matrix ===

  a    b  <-- classified as
514 118 |  a = male
170 458 |  b = female

```

Mirant primerament les dades més generals obtingudes amb el classificador Naive Bayes, veiem que des de bon començament són pitjors valors que amb l'anterior classificador SMO. I a la vegada que aquest presentava un creixement constant, Naive Bayes, presenta un creixement que sembla començar a estancar-se quan augmentem considerablement el valor d'N.

Podem intuir que aquest fet diferencial és degut al comportament del classificador Naive bayes, el qual tracta cada un dels elements del vector de



features per separat, provocant que amb molt augment d'aquest vector no impliqui un augment directe de la precisió del algoritme.

A part d'aquest fet, veiem que l'algoritme també té un comportament similar pel que fa als valors de precisió, recall i F-measure, els valors quals amb un augment d'N, s'estabilitzen i s'igualen.

## Random Forest:

N = 5:

```
=== Summary ===

Correctly Classified Instances      875          69.4444 %
Incorrectly Classified Instances    385          30.5556 %
Kappa statistic                    0.3888
Mean absolute error                 0.3155
Root mean squared error             0.4552
Relative absolute error              63.0934 %
Root relative squared error         91.049 %
Total Number of Instances          1260

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
          0,722    0,333    0,686     0,722    0,703      0,389    0,793     0,791    male
          0,667    0,278    0,704     0,667    0,685      0,389    0,793     0,799    female
Weighted Avg.   0,694    0,306    0,695     0,694    0,694      0,389    0,793     0,795

=== Confusion Matrix ===

  a    b  <-- classified as
456 176 |   a = male
209 419 |   b = female
```

N = 50:

```
=== Summary ===

Correctly Classified Instances      1074          85.2381 %
Incorrectly Classified Instances     186          14.7619 %
Kappa statistic                    0.7047
Mean absolute error                 0.2652
Root mean squared error             0.3352
Relative absolute error              53.0341 %
Root relative squared error         67.0432 %
Total Number of Instances          1260

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
          0,861    0,156    0,847     0,861    0,854      0,705    0,935     0,938    male
          0,844    0,139    0,858     0,844    0,851      0,705    0,935     0,935    female
Weighted Avg.   0,852    0,148    0,852     0,852    0,852      0,705    0,935     0,936

=== Confusion Matrix ===

  a    b  <-- classified as
544   88 |   a = male
 98 530 |   b = female
```

El fet més significatiu que veiem amb el classificador Random Forest és que seleccionant una  $N$  molt petita, com veiem amb el primer exemple per  $N = 5$ , el percentatge d'acert és molt reduït, d'un 69%

En canvi, al augmentar la  $N$  amb un nombre com 50, el percentatge d'acert augmenta molt considerablement fins a arribar a un 85,2%. Aquest, amb altres dades d' $N$  es manté amb valors molt similars.

Com que el funcionament del Random Forest, es divideix en diferents parts "Trees" i cada un d'ells realitza una votació basada en l'aprenentatge que han tingut cada un d'ells per separat. Aquest aprenentatge i votació serà molt més fiable amb nombres més grans d' $N$ .