ASSIGNMENT 4: Document classification

1.1 Research plan

The main task is to train a classifier based on Machine Learning for recognizing documents that belong to one of the selected categories on the basis of the content of the documents.

1.2 Datasets

The 5 categories selected to study are:

1. Rec.motorcycles
2. Comp.sys.ibm.pc.hardware
3. Soc.religion.christian
4. Rec.sport.hockey
5. Misc.forsale

The datasets used in the following experiments are two:

- Dataset 1 (2972, 10001): The first dataset was created by selecting, in the 2972 documents from the 5 categories, the 10 000 words that are most frequent and occur in more than 1 document.
- Dataset 2 (2972, 4906): The second dataset was created by selecting the most frequent words that occur in more than 5 documents. For this dataset we only have 4906 words passing these restrictions.

1.3 Research environment

The enviroment to study this practice has been a MacBook Air 13” with a processor Intel Core i5 (1,8 GHz).

1.4 Measures

We will focus on the two machine learning algorithms:

• Naive Bayesian Classifier (NB),

• Decision Trees (DT).

Additionally, we will see another machine learning algorithm:

• Multinomial Logistic Regression with a ridge estimator (MLR).

1.5 Experiments

Experiment 1

Goal: Study the parameters of Naïve Bayesian Classifier and see which is the best combination of them for Dataset 1.

Assumptions:

Constants:
- output-debug-info : False
- do-not-check-capabilities: False
- cross-validation: 10 folds

Variables:
- num-decimal-places
- batch-size

Results:

Experiment 1.1

- num-decimal-places = 2
- batch-size = 100

Correctly Classified Instances	*2667*	*89.7376 %*
Incorrectly Classified Instances	*305*	*10.2624 %*

Experiment 1.2

- num-decimal-places = 5
- batch-size = 100

Correctly Classified Instances	*2667*	*89.7376 %*
Incorrectly Classified Instances	*305*	*10.2624 %*

Experiment 1.3

- num-decimal-places = 2
- batch-size = 5

Correctly Classified Instances	*2667*	*89.7376 %*

Incorrectly Classified Instances 305 10.2624 %

There is no difference between changing the parameters of this experiment, so we will leave them as default like in the first experiment. But it should be remarked that when you only have 2 decimals it is not possible to see many individual probabilities of the words as they come out as 0 when they really are not exactly 0.

Experiment 2

Goal: Study the parameters of Decision Trees and see which is the best combination of them for Dataset 1.

Assumptions:

Constants:
- batchSize : 100
- output-debug-info : False
- do-not-check-capabilities: False
- initial_count: 0.0
- minVarianceProp: 0.001
- numDecimalPlaces: 2
- seed: 1
- spreadInitialCount: False

Variables:
- max_depth: It determines the maximum depth of your decision tree. By default, it is -1 which means the algorithm will automatically control the depth. But you can manually tweak this value to get the best results on your data.
- minNum: Minimum number of instances per leaf. If not mentioned, the tree will keep splitting till all leaf nodes have only one class associated with it. Pruning means to automatically cut back on a leaf node that does not contain much information. This keeps the decision tree simple and easy to interpret.
- noPruning: Pruning means to automatically cut back on a leaf node that does not contain much information. This keeps the decision tree simple and easy to interpret.
- numFolds: The specified number of folds of data will be used for pruning the decision tree. The rest will be used for growing the rules.

Results:

Experiment 2.1

- max_depth = -1
- minNum = 2.0
- noPruning = False
- numFolds = 3

Correctly Classified Instances 2168 72.9475 %

Incorrectly Classified Instances 804 27.0525 %

Observations: First experiment shows us the number of correct and incorrect classified instances with the default parameters of the algorithm.

Experiment 2.2

- max_depth = 200
- minNum = 2.0
- noPruning = False
- numFolds = 3

Correctly Classified Instances 2168 72.9475 %

Incorrectly Classified Instances 804 27.0525 %

Observations: Second experiment shows us that changing the max_depth to a big number doesn't make any changes to the results.

Experiment 2.3

- max_depth = -1
- minNum = 4.0
- noPruning = False
- numFolds = 3

Correctly Classified Instances 2153 72.4428 %

Incorrectly Classified Instances 819 27.5572 %

Observations: Third experiment shows us that changing minNum to 4.0 only reduces a little bit the correctly classification, so we will keep as default.

Experiment 2.4

- max_depth = -1
- minNum = 2.0
- noPruning = True
- numFolds = 3

Correctly Classified Instances 2221 74.7308 %

Incorrectly Classified Instances 751 25.2692 %

Observations: Fourth experiment shows us that changing the parameter noPruining to True creates a big tree impossible to interpret but improves the results. This tree is not cutting back when there's no much information.

Experiment 2.5

- max_depth = 200
- minNum = 2.0

- noPruning = True
- numFolds = 10

Correctly Classified Instances 2221 74.7308 %

Incorrectly Classified Instances 751 25.2692 %

Observations: Fifth experiment shows us that changing the parameter numFolds = 10 doesn't change the results.

Looking at the results of experiment 2 we will keep the configuration of the experiment 2.4 for Decision Tree algorithm.

Experiment 3

Goal: Study the Naïve Bayesian Classifier for Dataset 2.

Assumptions:

Constants:
- output-debug-info : False
- do-not-check-capabilities: False
- cross-validation: 10 folds
- num-decimal-places = 2
- batch-size = 100

Variables:
Empty

Results:

Experiment 3.1

Correctly Classified Instances 2642 88.8964 %

Incorrectly Classified Instances 330 11.1036 %

Observations: Experiment 3 shows us that is better to use Dataset 1 than Dataset 2 with Naïve Bayesian Classifier. But then we will check if is the same for Decision Trees algorithm.

Experiment 4

Goal: Study the Decision Trees algorithm for Dataset 2.

Assumptions:

Constants:
- batchSize : 100
- output-debug-info : False
- do-not-check-capabilities: False
- initial_count: 0.0

- minVarianceProp: 0.001
- numDecimalPlaces: 2
- seed: 1
- spreadInitialCount: False
- max_depth = -1
- minNum = 2.0
- noPruning = True
- numFolds = 3

Variables:
Empty

Results:

Experiment 4.1

Correctly Classified Instances	*2216*	*74.5626 %*
Incorrectly Classified Instances	*756*	*25.4374 %*

Observations: Experiment 4 shows us that is better to use Dataset 1 than Dataset 2 with Decision Trees algorithm.

Feature Selection

- It enables the machine learning algorithm to train faster.
- It reduces the complexity of a model and makes it easier to interpret.
- It improves the accuracy of a model if the right subset is chosen.
- It reduces overfitting.

Experiment 5

Goal: Study the Naïve Bayesian Classifier for Dataset 1 with Correlation Based Feature Selection.

Assumptions:

Constants:
- output-debug-info : False
- do-not-check-capabilities: False
- cross-validation: 10 folds
- num-decimal-places = 2
- batch-size = 100

Variables:
- threshold feature selection

Results:

Experiment 5.1

- threshold feature selection = 0.01

Correctly Classified Instances 2673 89.9394 %

Incorrectly Classified Instances 299 10.0606 %

Observations: Experiment 5.1 adds a feature selection algorithm to pass from 10000 features to 9860 and improve a little bit the result.

Experiment 5.2

- threshold feature selection = 0.02

Correctly Classified Instances 2683 90.2759 %

Incorrectly Classified Instances 289 9.7241 %

Observations: Experiment 5.2 adds a feature selection algorithm to pass from 10000 features to 6413 and thanks for that we can reduce the noise and improve the result.

Experiment 5.3

- threshold feature selection = 0.03

Correctly Classified Instances 2647 89.0646 %

Incorrectly Classified Instances 325 10.9354 %

Observations: Experiment 5.3 adds a feature selection algorithm to pass from 10000 features to 2647 and gets a little bit worse.

Experiment 5.4

- threshold feature selection = 0.05

Correctly Classified Instances 2488 83.7147 %

Incorrectly Classified Instances 484 16.2853 %

Observations: Experiment 5.4 adds a feature selection algorithm to pass from 10000 features to 2647 and gets totally worse.

Experiment 5.5

- threshold feature selection = 0.015

Correctly Classified Instances 2686 90.3769 %

Incorrectly Classified Instances *286* *9.6231 %*

Observations: Experiment 5.5 adds a feature selection algorithm to pass from 10000 features to 8798 and gets the best solution at the moment.

Experiment 6

Goal: Study the Naïve Bayesian Classifier for Dataset 1 with Information Gain Based Feature Selection.

Assumptions:

Constants:
- output-debug-info : False
- do-not-check-capabilities: False
- cross-validation: 10 folds
- num-decimal-places: 2
- batch-size: 100
- search method: Ranker

Variables:
- threshold feature selection

Results:

Experiment 6.1

- threshold feature selection = 0.01

Correctly Classified Instances *2557* *86.0363 %*

Incorrectly Classified Instances *415* *13.9637 %*

Observations: Experiment 6.1 adds a feature selection algorithm to pass from 10000 features to 791 and get worse the result.

Experiment 6.2

- threshold feature selection = 0.005

Correctly Classified Instances *2628* *88.4253 %*

Incorrectly Classified Instances *344* *11.5747 %*

Observations: Experiment 6.2 adds a feature selection algorithm to pass from 10000 features to 1874 and thanks for that we can reduce the noise and improve a little bit the result from before.

Experiment 6.3

- threshold feature selection = 0.001

Correctly Classified Instances	*2656*	*89.3674 %*
Incorrectly Classified Instances	*316*	*10.6326 %*

Observations: Experiment 6.2 adds a feature selection algorithm to pass from 10000 features to 2975 and thanks for that we can reduce the noise and improve a little bit the result from before. From here no matter how much we lower the threshold we will not be able to get more than 2975 features as the rest have the value 0.

• influence of different parameters characteristic for the algorithms,

• feature selection methods and their parameters,

• discovering bad and strong aspects of the examined algorithms,

• finding out sets of features that are characteristic for different classes of documents

and what they tell us about these classes,

• behaviour of classifiers for different subsets of the decision classes (it would be good

to formulate earlier what do we expect to discover).

1.6 Bad and strong aspects of the examined algorithms

Naïve Bayes Classifier

- Bad Aspects
  - Really difficult to validate the assumption of fully independence between features in real world cases.
  - Problems with zero frequency.

- Strong Aspects
  - This algorithm executes quickly and can save a lot of time.
  - If we get independence of our features, it can perform really good and fast.

Decision Trees Classifier

- Bad Aspects

- A small change in the data can cause a large change in the structure of the decision tree.
- Can get really complex trees impossible to interpret.
- Decision tree involves higher time to train the model.

- Strong Aspects
- Can work with numerical and categorical features.
- Feature selection happens automatically: unimportant features will not influence the result. The presence of features that depend on each other (multicollinearity) also doesn't affect the quality.

1.7 Sets of features that are characteristic for different classes

- rec.motorcycles: bike, people, riding, think, say, just.
- misc.forsale: sale, offer, new.
- soc.religion.christian: god, people, Jesus, shipping, think, church.
- comp.sys.ibm.pc.hardware: bus, ide, card, scsi, drive, controller.
- rec.sport.hockey: game, play, team, season, hockey.

1.8 Best possible text classification system

The best possible text classification system found is using the first dataset. In the dataset we use Correlation Based Feature Selection to select the most important features and finally we use with Naïve Bayes classifier to do the classification of the documents. We have succeeded a 90.3769 % of correctly documents classified using a 10-Fold cross validation.

The next information shows us the commands used to run the classifier and basic information about the dataset.

=== Run information ===

Scheme: weka.classifiers.bayes.NaiveBayesMultinomial

Relation: dataset1-weka.filters.supervised.attribute.AttributeSelection-Eweka.attributeSelection.CorrelationAttributeEval-Sweka.attributeSelection.Ranker -T 0.015 -N -1

Instances: 2972

Attributes: 8798

[list of attributes omitted]

Test mode: 10-fold cross-validation

As we can see the independent probability of a class is always 0.2.

=== Classifier model (full training set) ===

The independent probability of a class

----

soc.religion.christian 0.2

comp.sys.ibm.pc.hardware 0.2

misc.forsale 0.2

rec.motorcycles 0.2

rec.sport.hockey 0.2

Finally we have a summary of all the results obtained and a confusion matrix that shows when the classifier rights and when misses.

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	*2686*	*90.3769 %*
Incorrectly Classified Instances	*286*	*9.6231 %*
Kappa statistic	*0.8797*	
Mean absolute error	*0.0421*	
Root mean squared error	*0.1725*	
Relative absolute error	*13.1553 %*	
Root relative squared error	*43.1364 %*	
Total Number of Instances	*2972*	

=== Detailed Accuracy By Class ===

TP Rate	*FP Rate*	*Precision*	*Recall*	*F-Measure*	*MCC*	*ROC Area*	*PRC Area*	*Class*

	0,896	0,027	0,893	0,896	0,895	0,868	0,990	0,964	soc.religion.christian
	0,819	0,020	0,911	0,819	0,862	0,833	0,974	0,929	comp.sys.ibm.pc.hardware
	0,967	0,018	0,932	0,967	0,949	0,936	0,997	0,991	misc.forsale
	0,917	0,033	0,873	0,917	0,894	0,868	0,987	0,955	rec.motorcycles
	0,918	0,023	0,911	0,918	0,915	0,893	0,995	0,982	rec.sport.hockey
Weighted Avg.	0,904	0,024	0,904	0,904	0,903	0,880	0,988	0,964	

=== Confusion Matrix ===

a	b	c	d	e	<-- classified as
536	16	18	8	20	a = soc.religion.christian
32	479	4	58	12	b = comp.sys.ibm.pc.hardware
5	1	579	5	9	c = misc.forsale
6	25	5	541	13	d = rec.motorcycles
21	5	15	8	551	e = rec.sport.hockey

1.7 Conclusions

To sum up, we have worked to find the best text classifier for our own dataset and to do that we created different experiments to find the best parameterization. Doing more experiments probably we could find a better solution, but we are satisfied with the results. We have a model that classifies documents of 5 different categories with a 90.3769% of correctly classification. The feature selection has been crucial to improve the solution because we deleted noise about the dataset and we also improve the efficiency of the algorithm.