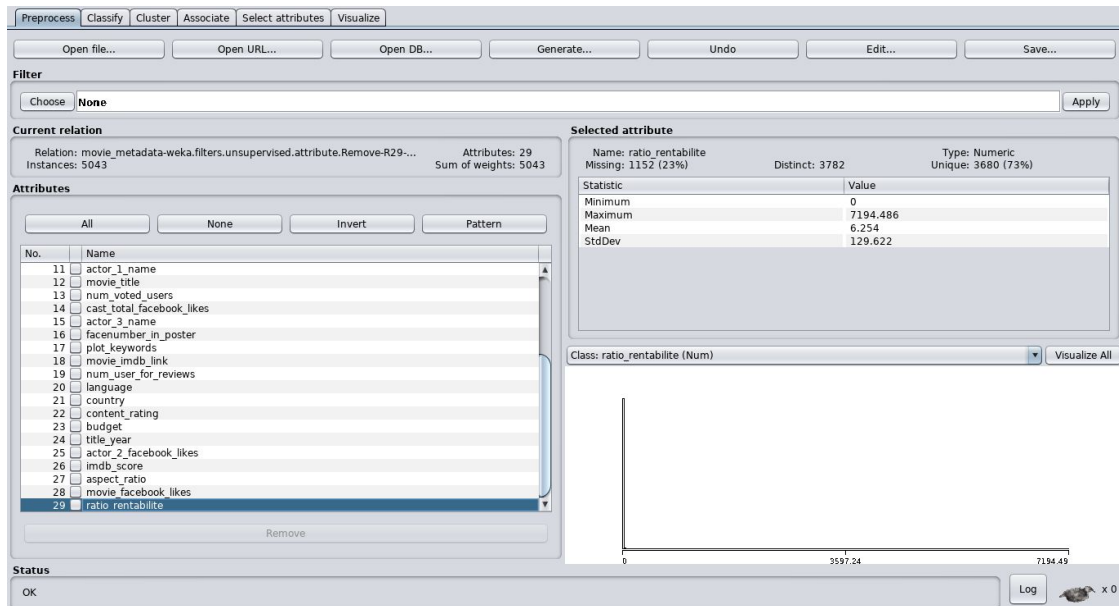


# Projet Fouille de données

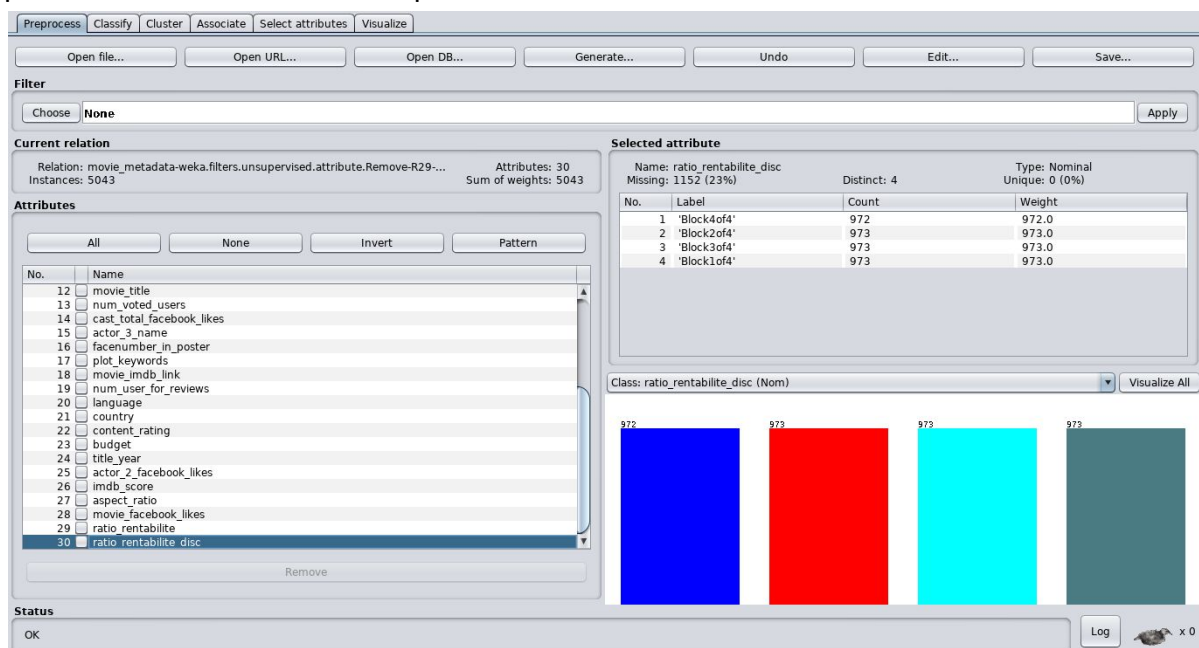
## Partie 1 – Analyse descriptive des données:

### Création d'attributs :

- Création un attribut « ratio\_rentabilité » qui est un ratio entre ce qu'a rapporté le film et ce qu'il a coûté.



- Création de l'attribut « ratio\_rentabilite\_disc » qui discrétisé de façon pertinente le ratio de rentabilité précédemment calculé.



- Création de l'attribut « score\_IMBD\_disc » qui discrétisera le score IMDB.

**Selected attribute**

Name: ratio\_rentabilite\_disc  
Missing: 1152 (23%)  
Distinct: 4  
Type: Nominal  
Unique: 0 (0%)

No.	Label	Count	Weight
1	'Block4of4'	972	972.0
2	'Block2of4'	973	973.0
3	'Block3of4'	973	973.0
4	'Block1of4'	973	973.0

Class: ratio\_rentabilite\_disc (Nom)

Visualize All

972 973 973 973

## Attributs et Statistique descriptive

1:L'attribut color → Type String, Couleur film ( noir ou blanc).

Remarque : si le film il est en couleur il reçoit plus de j'aime.

2:L'attribut director\_name → Type String, ça représente le réalisateur d'un film.

3:L'attribut num\_critic\_for\_reviews→ Type Numérique, ça représente le nombre commentaires sur un film.

4:L'attribut duration→ Type Numérique, ça représente la durée du film par ( heure, minutes , secondes).

5:L'attribut director\_facebook\_likes → Type Numérique, ça représente le nombre de j'aimes que le réalisateur a eu facebook a propos d'un film.

6:L'attribut actor\_3\_facebook\_likes → Type Numérique, ça représente le nombre de j'aime que le 3éme acteur d'un film à eu sur facebook.

7:L'attribut actor\_2\_name →Type String, ça représente le nom du deuxième acteur dans un film.

8:L'attribut **actor\_1\_facebook\_likes** → Type Numérique, ça représente le nombre de j'aime que le 1<sup>er</sup> acteur d'un film à eu sur facebook.

9:L'attribut **gross**→ Type Numérique, ça représente la somme d'argent gagné par un film ( les bénéfices d'un film ).

L'attribut **genres** → Type String, ça représente le style d'un film ( reportage , documentaire, film , série ) d'un film à eu sur facebook.

L'attribut **actor\_1\_name** → Type String, ça représente le nom du premier acteur du film.

L'attribut **movie\_title** → Type String, ça représente le titre du film.

L'attribut **num\_voted\_users** → Type Numérique, ça représente le nombre de personne qui ont voté pour un film.

L'attribut **cast\_total\_facebook\_likes** →Type numérique, ça représente le total de j'aimes du film parmi les autres films. ( genre d'un classement).

L'attribut **actor\_3\_name** →Type String, ça représente le nom du 3<sup>ème</sup> acteur du film.

L'attribut **facenumber\_in\_poster** → Type numérique , ça représente le nombre de visage sur le poster du film.

L'attribut **plot\_keywords** →Type String, ça représente les mots clé pour rechercher un film.

L'attribut **movie\_imdb\_link** → Type String, ça représente le lien pour accéder à un film qui est dans le site imdb.

L'attribut **num\_user\_for\_reviews** →Type numérique ça représente le nombre d'utilisateurs qui ont fait des critiques sur ce film.

L'attribut **language** →Type String, ça représente la langue dans laquelle le film est réalisé

L'attribut **country** → Type String, ça représente le pays dont le film à été tourné et réalisé.

L'attribut **content\_rating** → Type String, le nombre d'étoile pour faire une évaluation sur ce film.

L'attribut **budget** → Type numérique, ça représente le budget total du film.

L'attribut **title\_year** → Type numérique, ça représente l'année de la sortie du film.

L'attribut **actor\_2\_facebook\_likes** → Type numérique, ça représente le le nombre de j'aimes que le deuxième acteur à récolté.

L'attribut **imdb\_score** → Type numérique, ça représente le nombre du vote pour le film sur le site imdb.

L'attribut **aspect\_ratio** → Type numérique, le format de la réalisation du film ( exemple : 4 \* 3 , 16 \* 9 ).

L'attribut **movie\_facebook\_likes** → Type numérique, le nombre de j'aimes que le film a pu récolté sur facebook.

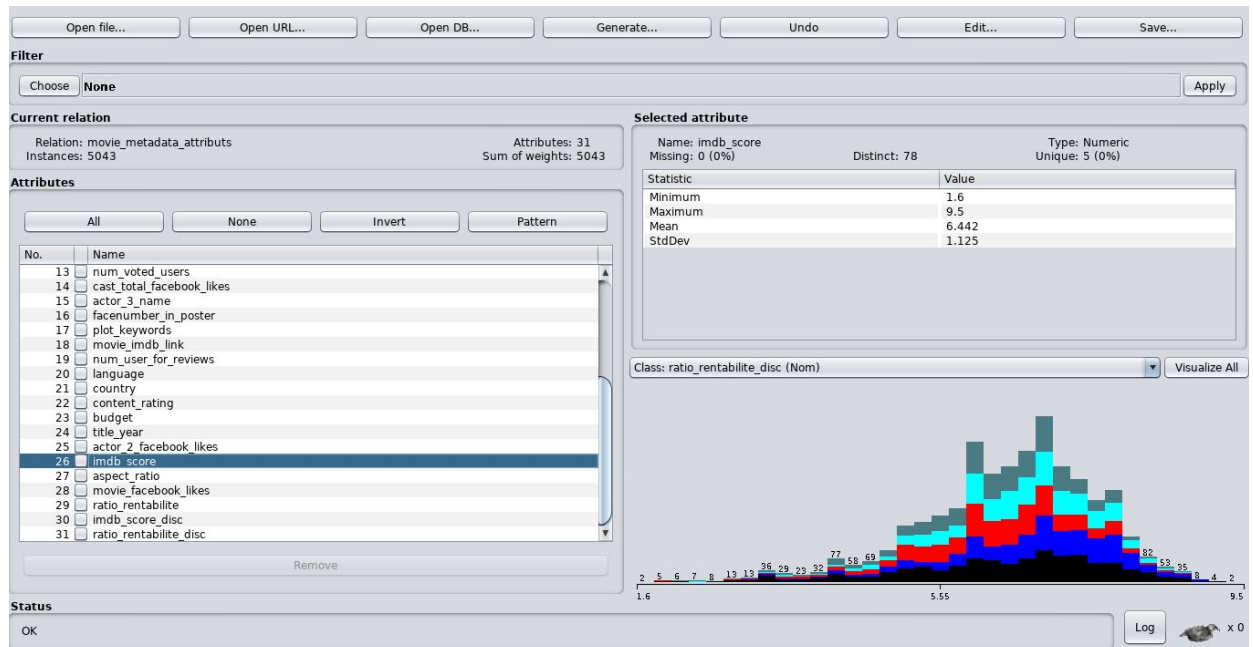
L'attribut **ratio\_rentabilite** → Type numérique, il représente la rentabilité du film.

L'attribut **ratio\_rentabilite\_disc** → Type numérique, il représente un interval qui sert à regrouper des films par rapport a leurs rentabilité.

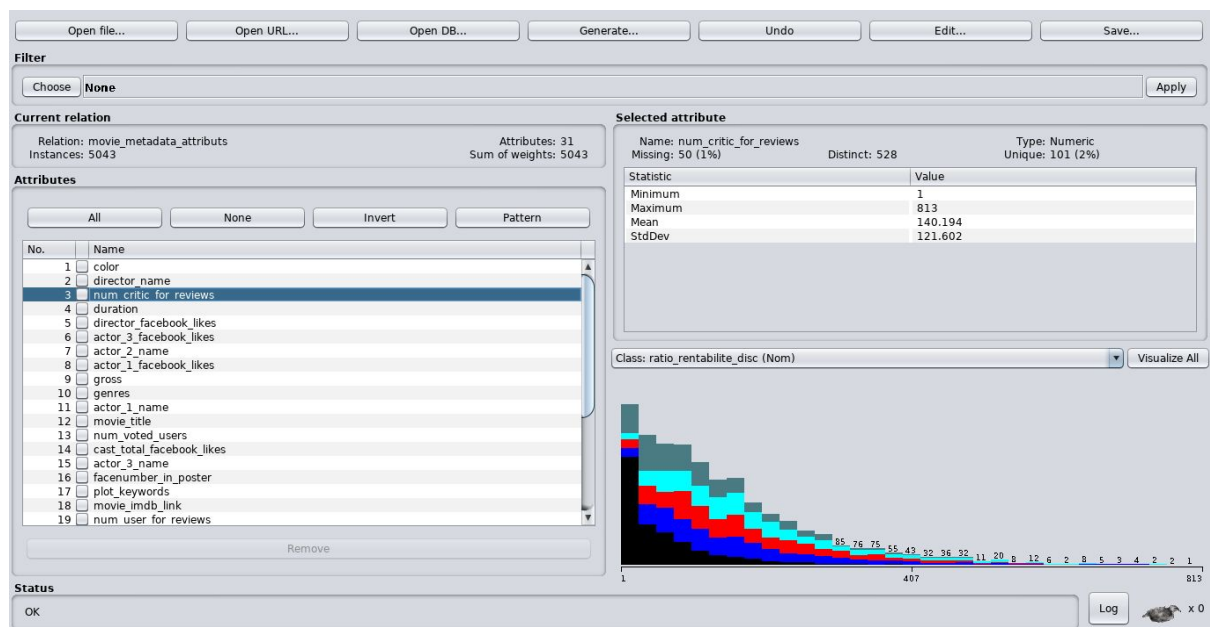
L'attribut **score\_IMBD\_disc** → Type numérique, il représente un interval qui sert à regrouper des films par rapport au score IMBD.

## Statistique descriptive:

- On remarque que la rentabilité (resp score imbc ) augmente avec l'augmentation du score imbc..



- On remarque que la rentabilité ( resp score imbc ) diminue avec l'augmentation des critiques.



- On remarque que la rentabilité ( resp score imbc ) diminue avec l'augmentation des critiques.

## Partie 2 – Segmentation de films

SimpleKmeans avec 4 clusters sur rentabilite:

The screenshot shows the Weka Clusterer window with the SimpleKMeans algorithm selected. The 'Cluster mode' section has 'Classes to clusters evaluation' selected with '(Nom) ratio\_rentabilite\_disc' as the evaluation attribute. The 'Clusterer output' pane displays the following information:

```
imdb_score_disc 'Block3of4' 'Block1of4' 'Block3of4' 'Block1of4' 'Block2of4'
```

Time taken to build model (full training data) : 0.16 seconds

=== Model and evaluation on training set ===

Clustered Instances

Cluster	Count	Percentage
0	1539	( 31%)
1	1619	( 32%)
2	376	( 7%)
3	1509	( 30%)

Class attribute: ratio\_rentabilite\_disc  
Classes to Clusters:

```
0 1 2 3 <-- assigned to cluster
717 670 202 535 | 'Block4of4'
310 280 42 341 | 'Block2of4'
247 406 25 295 | 'Block3of4'
265 263 107 338 | 'Block1of4'
```

Cluster 0 <-- 'Block4of4'  
Cluster 1 <-- 'Block3of4'  
Cluster 2 <-- 'Block1of4'  
Cluster 3 <-- 'Block2of4'

Incorrectly clustered instances : 3472.0 68.8479 %

Après avoir tester plusieurs découpages, on a fini par choisir un découpage en 4 clusters. On a obtenu un Incorrectly clustered instances de 3472 68.8479 %.

SimpleKmeans avec 4 clusters sur imdb score:

The screenshot shows the Weka Clusterer window with the SimpleKMeans algorithm selected. The 'Cluster mode' section has 'Classes to clusters evaluation' selected with '(Nom) imdb\_score\_disc' as the evaluation attribute. The 'Clusterer output' pane displays the following information:

```
ratio_rentabilite_disc 'Block2of4' 'Block1of4' 'Block2of4' 'Block2of4' 'Block2of4'
```

Time taken to build model (full training data) : 0.07 seconds

=== Model and evaluation on training set ===

Clustered Instances

Cluster	Count	Percentage
0	1587	( 31%)
1	2639	( 52%)
2	179	( 4%)
3	638	( 13%)

Class attribute: imdb\_score\_disc  
Classes to Clusters:

```
0 1 2 3 <-- assigned to cluster
340 605 79 194 | 'Block4of4'
442 677 42 178 | 'Block3of4'
395 618 29 134 | 'Block2of4'
410 739 29 132 | 'Block1of4'
```

Cluster 0 <-- 'Block3of4'  
Cluster 1 <-- 'Block1of4'  
Cluster 2 <-- 'Block2of4'  
Cluster 3 <-- 'Block4of4'

Incorrectly clustered instances : 3639.0 72.1594 %

On a obtenu un Incorrectly clustered instances de 3639 72.1594 %.

En utilisant SimpleKmeans: un meilleur clustering est obtenu avec le ratio rentabilité par rapport à au score imdb.

MakeDensityBasedCluster avec rentabilité:

The screenshot shows the Weka Clusterer window with the 'Cluster' tab selected. The 'Clusterer' dropdown is set to 'MakeDensityBasedClusterer'. The command line shows parameters: `-M 1.0E-6 -W weka.clusterers.SimpleKMeans --init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R`. The 'Cluster mode' section has 'Classes to clusters evaluation' selected with '(Nom) ratio\_rentabilite\_disc' as the attribute. The 'Clusterer output' pane displays the following information:

```
Attribute: ratio_rentabilite
Normal Distribution. Mean = 7.0022 StdDev = 149.0122
Attribute: imdb_score_disc
Discrete Estimator. Counts = 646 1320 687 24 (Total = 2677)

Time taken to build model (full training data) : 0.03 seconds

=== Model and evaluation on training set ===

Clustered Instances
0      3363 ( 67%)
1      1680 ( 33%)

Log likelihood: -82.09103

Class attribute: ratio_rentabilite_disc
Classes to Clusters:
0 1 <- assigned to cluster
1508 616 | 'Block4of4'
640 333 | 'Block2of4'
568 405 | 'Block3of4'
647 326 | 'Block1of4'

Cluster 0 <- 'Block4of4'
Cluster 1 <- 'Block3of4'

Incorrectly clustered instances :      3130.0  62.0662 %
```

Incorrectly clustered instances : 3130.0 62.0662 %

MakeDensityBasedCluster avec score\_imdb:

The screenshot shows the Weka Clusterer window with the 'Cluster' tab selected. The 'Clusterer' dropdown is set to 'MakeDensityBasedClusterer'. The command line is the same as in the previous screenshot. The 'Cluster mode' section has 'Classes to clusters evaluation' selected with '(Nom) imdb\_score\_disc' as the attribute. The 'Clusterer output' pane displays the following information:

```
Attribute: ratio_rentabilite
Normal Distribution. Mean = 6.6005 StdDev = 130.7141
Attribute: ratio_rentabilite_disc
Discrete Estimator. Counts = 579 2075 683 19 (Total = 3356)

Time taken to build model (full training data) : 0.03 seconds

=== Model and evaluation on training set ===

Clustered Instances
0      1464 ( 29%)
1      3579 ( 71%)

Log likelihood: -82.54542

Class attribute: imdb_score_disc
Classes to Clusters:
0 1 <- assigned to cluster
314 904 | 'Block4of4'
432 907 | 'Block3of4'
357 819 | 'Block2of4'
361 949 | 'Block1of4'

Cluster 0 <- 'Block3of4'
Cluster 1 <- 'Block1of4'

Incorrectly clustered instances :      3662.0  72.6155 %
```

Incorrectly clustered instances : 3662.0 72.6155 %

En utilisant MakeDensityBasedCluster: un meilleur clustering est obtenu avec le ratio rentabilité par rapport à au score imdb.

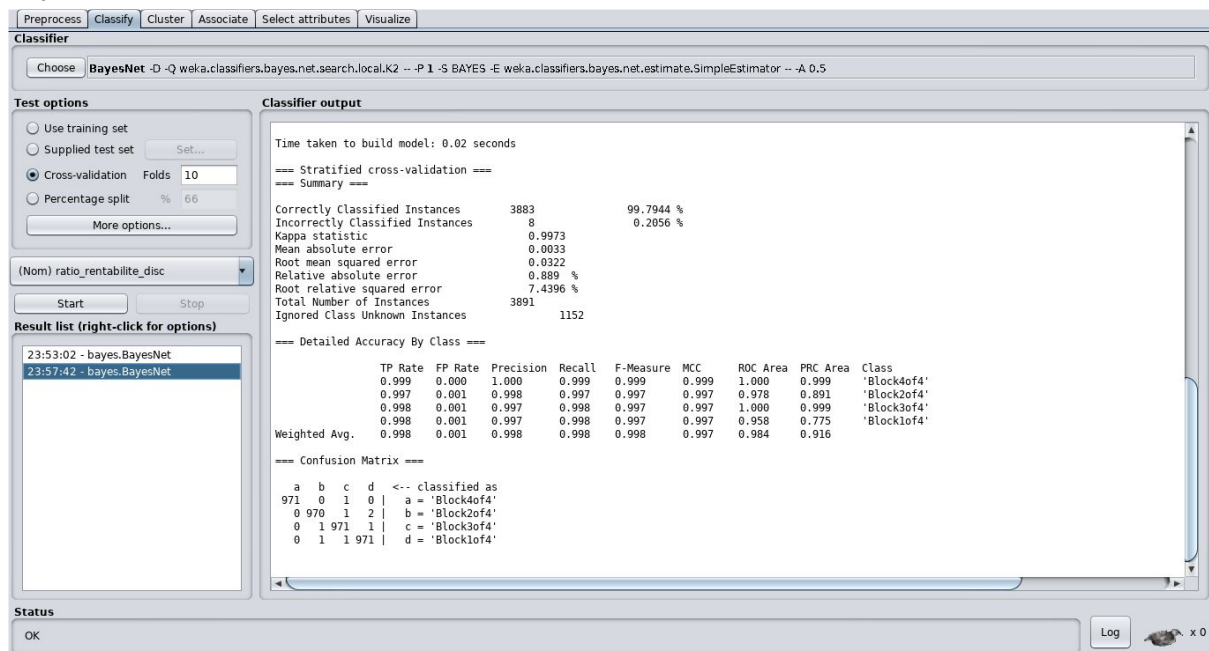
## Partie 3 – Prédiction

L'objectif est de trouver le meilleur algorithme pour trouver le score IMBD puis le ratio de rentabilité.

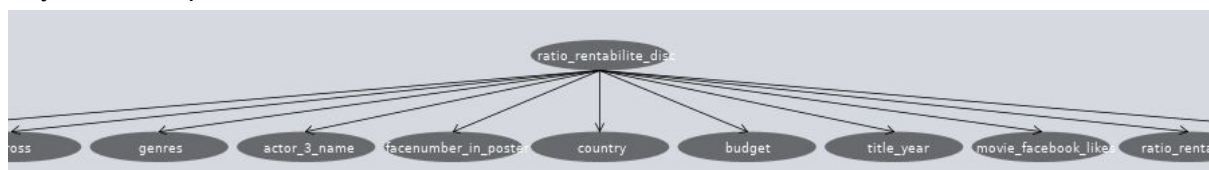
### Score IMDB:

### BayesNet:

### BayesNet résultat:



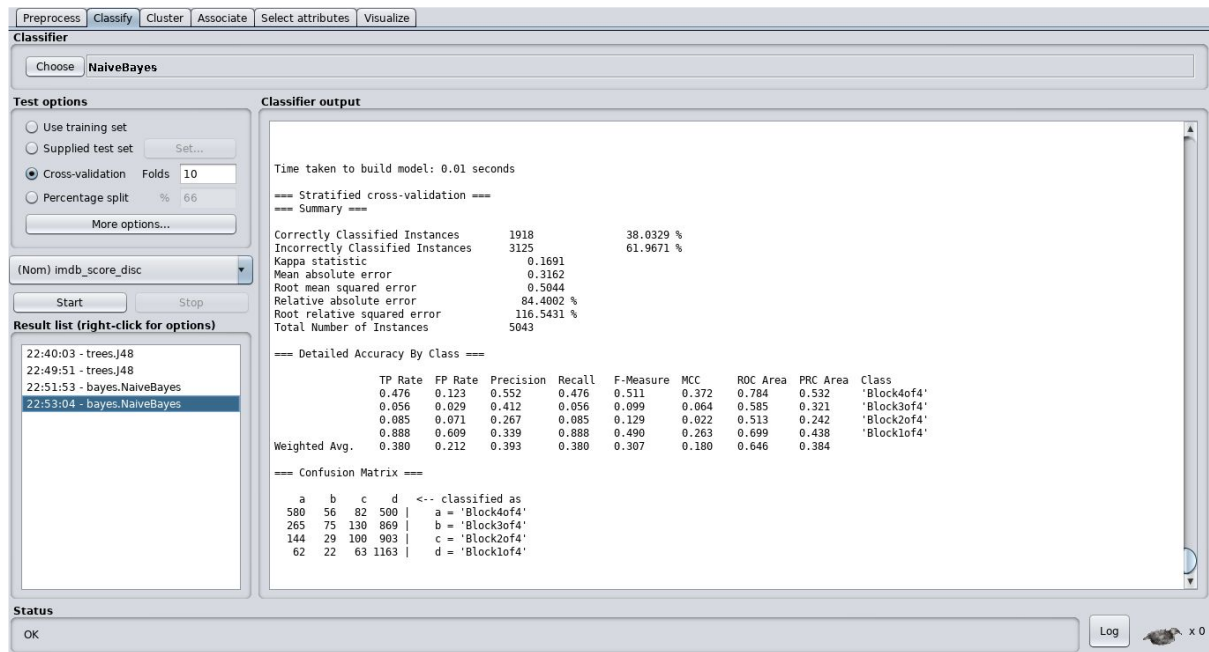
### BayesNet Graphe:



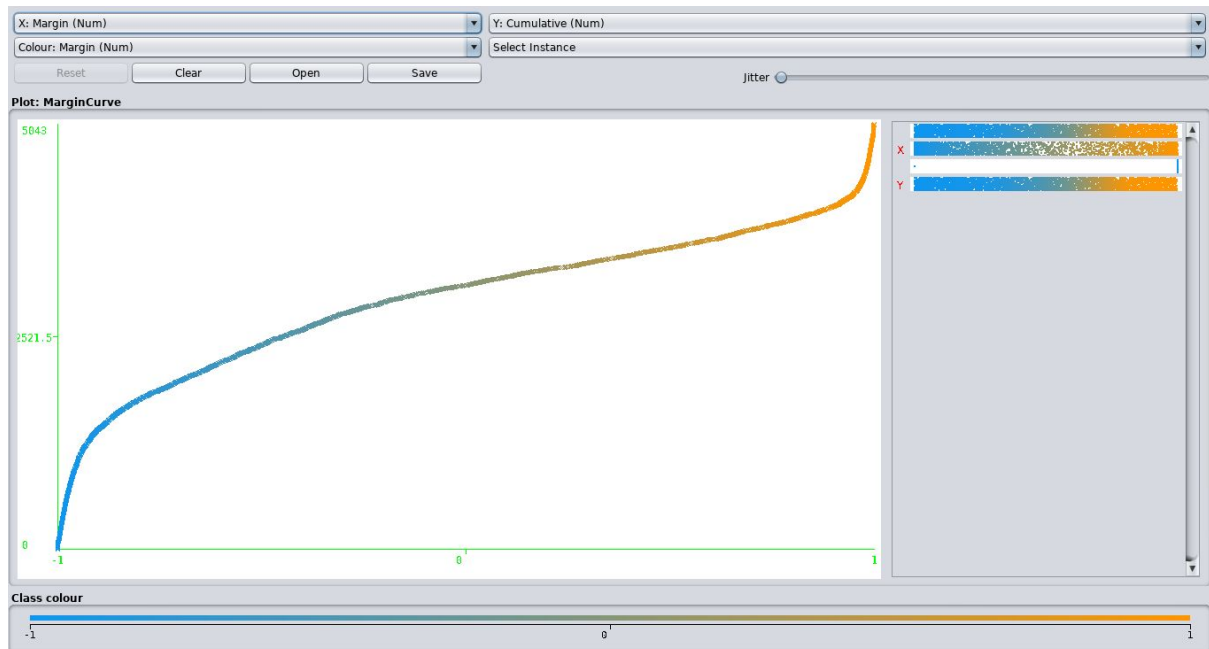


## NaiveBayes:

### NaiveBayes resultat:

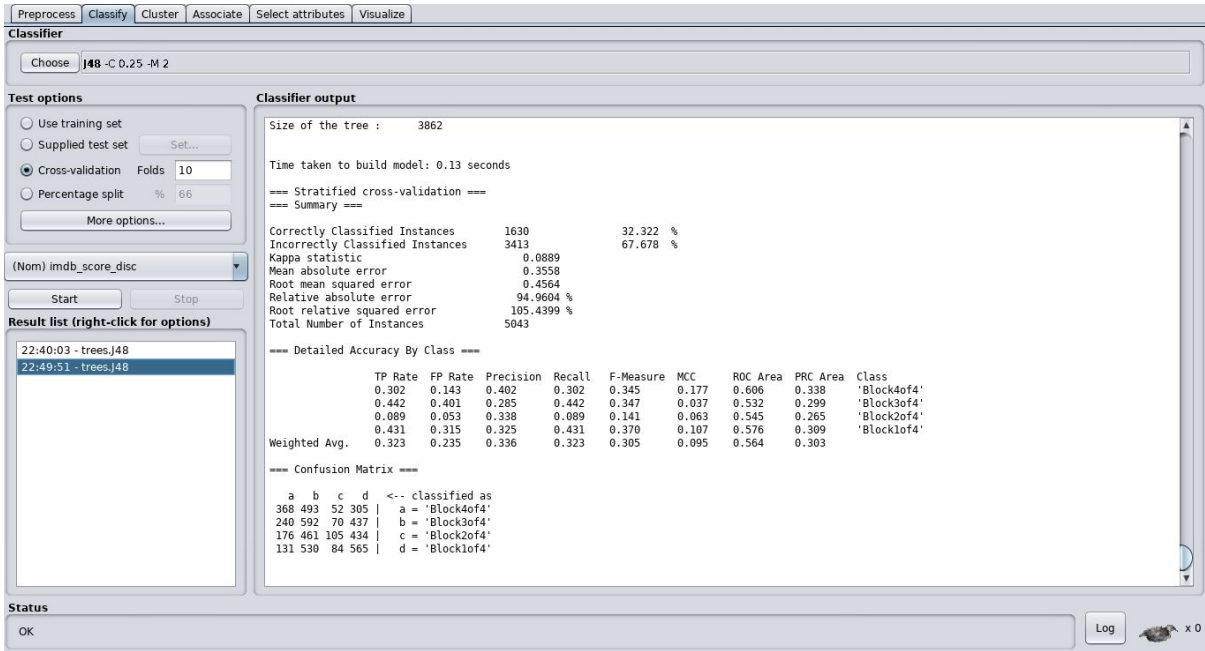


### NaiveBayes curve:

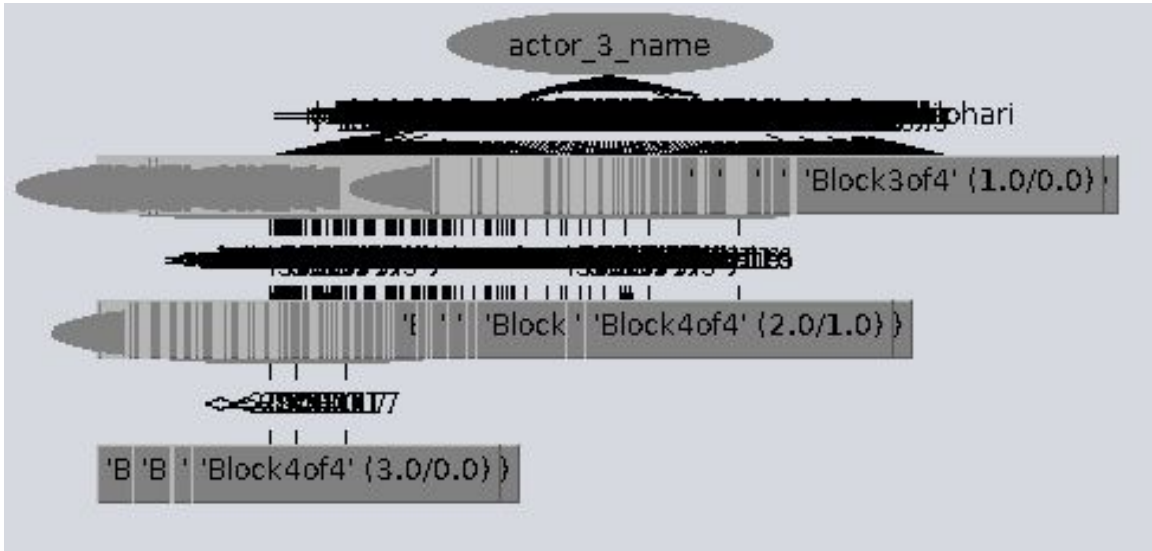


J48:

J48 résultat:



J48 arbre:



# DecisionStump:

## DecisionStump resultat:

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose

DecisionStump

Test options

☐ Use training set

☐ Supplied test set

☒ Cross-validation

☐ Percentage split

Folds

10

%

66

More options...

(Nom) ratio\_rentabilite\_disc

Start

Stop

Result list (right-click for options)

22:40:03 - trees.J48

22:49:51 - trees.J48

22:51:53 - bayes.NaiveBayes

22:53:04 - bayes.NaiveBayes

23:21:05 - trees.DecisionStump

23:21:15 - trees.DecisionStump

Classifier output

0.20.40.066666666666666670.3333333333333333

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances

1750

34.7016 %

Incorrectly Classified Instances

3293

65.2984 %

Kappa statistic

0.1245

Mean absolute error

0.3629

Root mean squared error

0.4263

Relative absolute error

96.8679 %

Root relative squared error

98.4901 %

Total Number of Instances

5043

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.493	0.248	0.387	0.493	0.434	0.227	0.618	0.333	'Block4of4'
	0.003	0.002	0.333	0.003	0.006	0.000	0.541	0.291	'Block3of4'
	0.000	0.000	0.000	0.000	0.000	0.000	0.531	0.244	'Block2of4'
	0.875	0.626	0.329	0.875	0.478	0.236	0.627	0.324	'Block1of4'
Weighted Avg.	0.347	0.223	0.268	0.347	0.231	0.118	0.580	0.299	

=== Confusion Matrix ===

a	b	c	d	<-- classified as
600	2	0	616	a = 'Block4of4'
495	4	0	840	b = 'Block3of4'
295	1	0	880	c = 'Block2of4'
159	5	0	1146	d = 'Block1of4'

Status

OK

Log

x 0

## Score rentabilité:

### BayesNet:

### BayesNet résultat

**Classifier**

Choose: **BayesNet** -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

**Test options**

☐ Use training set  
☐ Supplied test set Set...  
☒ Cross-validation Folds: **10**  
☐ Percentage split % 66  
More options...

(Nom) ratio\_rentabilite\_disc

Start Stop

**Result list (right-click for options)**

- 23:53:02 - bayes.BayesNet
- 23:57:42 - bayes.BayesNet

**Classifier output**

Time taken to build model: 0.02 seconds

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

Correctly Classified Instances	3883	99.7944 %
Incorrectly Classified Instances	8	0.2056 %
Kappa statistic	0.9973	
Mean absolute error	0.0033	
Root mean squared error	0.0322	
Relative absolute error	0.809 %	
Root relative squared error	7.4396 %	
Total Number of Instances	3891	
Ignored Class Unknown Instances	1152	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.999	0.000	1.000	0.999	0.999	0.999	1.000	0.999	'Block4of4'
	0.997	0.001	0.998	0.997	0.997	0.997	0.978	0.891	'Block2of4'
	0.998	0.001	0.997	0.998	0.997	0.997	1.000	0.999	'Block3of4'
	0.998	0.001	0.997	0.998	0.997	0.997	0.958	0.775	'Block1of4'
Weighted Avg.	0.998	0.001	0.998	0.998	0.998	0.997	0.984	0.916	

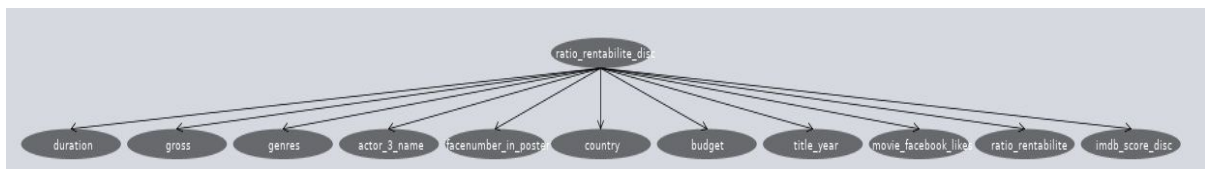
=== Confusion Matrix ===

a	b	c	d	<-- classified as
971	0	1	0	a = 'Block4of4'
0	970	1	2	b = 'Block2of4'
0	1	971	1	c = 'Block3of4'
0	1	1	971	d = 'Block1of4'

**Status**

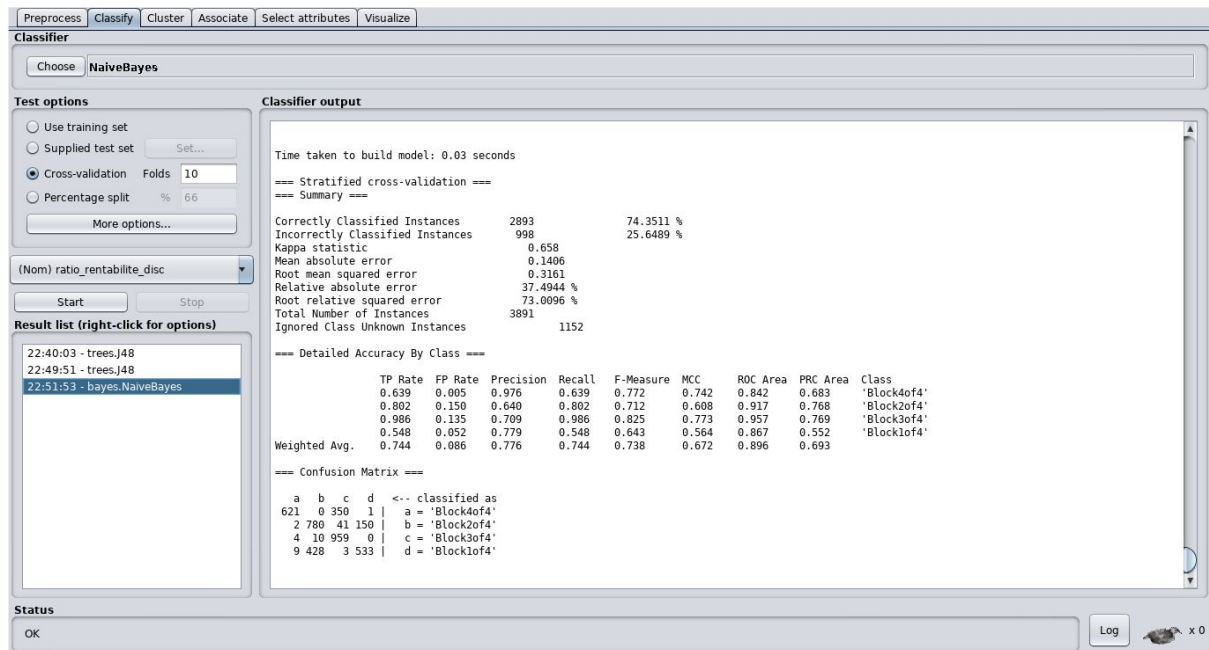
OK Log x 0

### BayesNet Graphe:

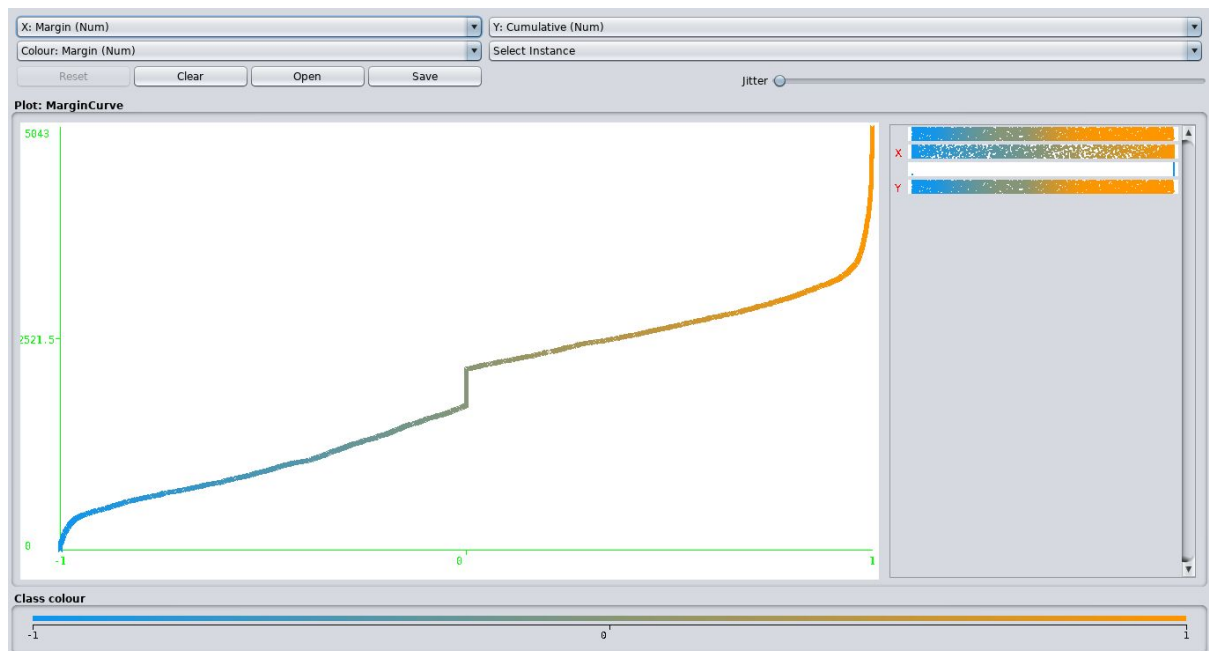


## NaiveBayes:

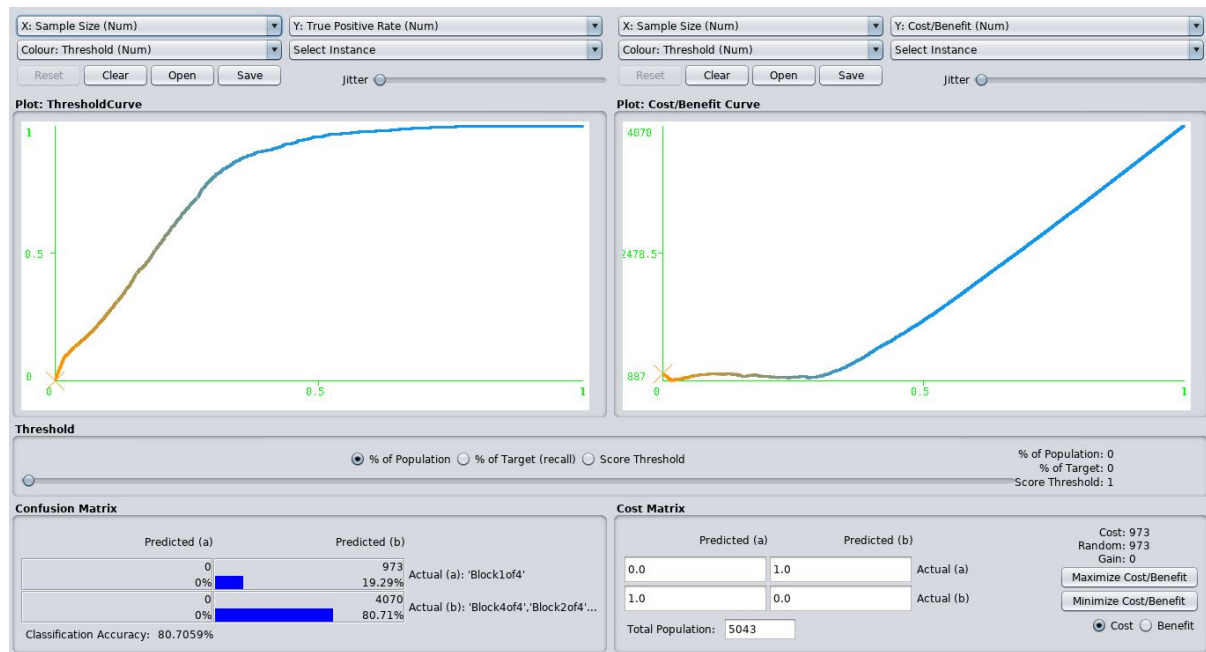
### NaiveBayes résultat:



## NaiveBayes curve:

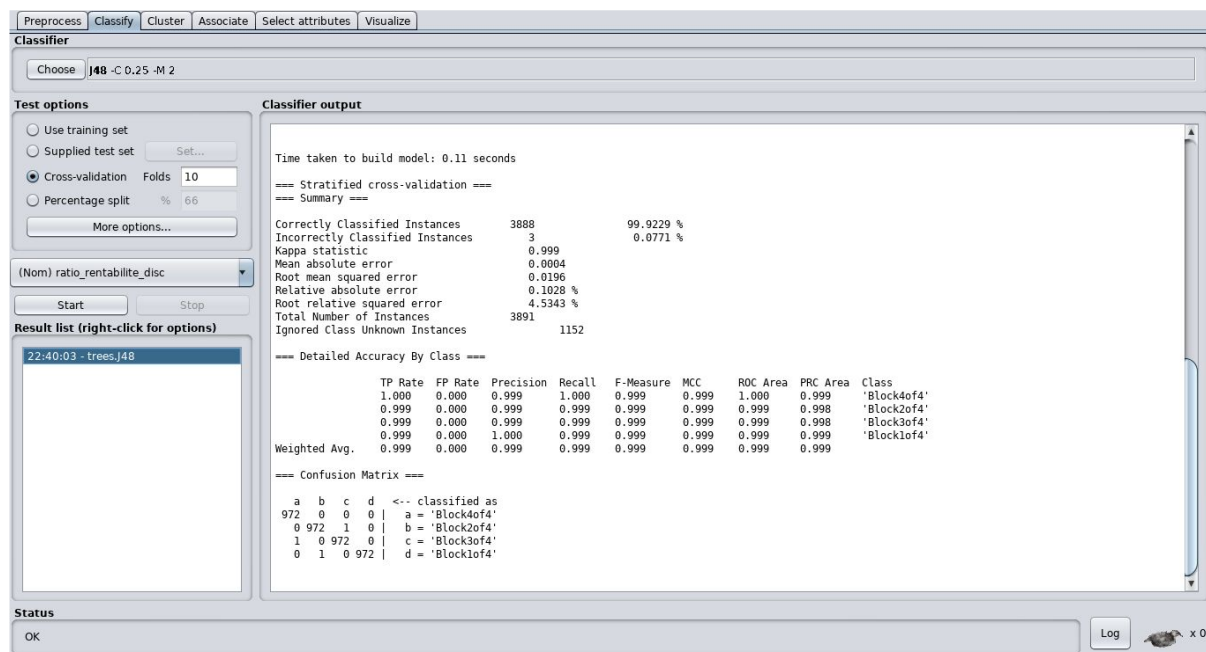


## NaiveBayes coût / bénéfice:

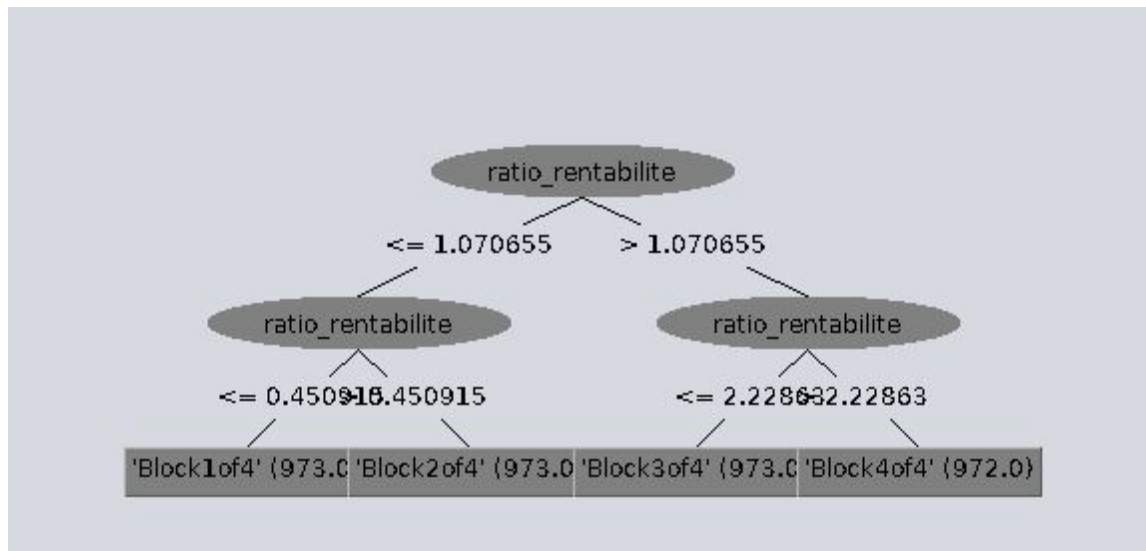


## J48:

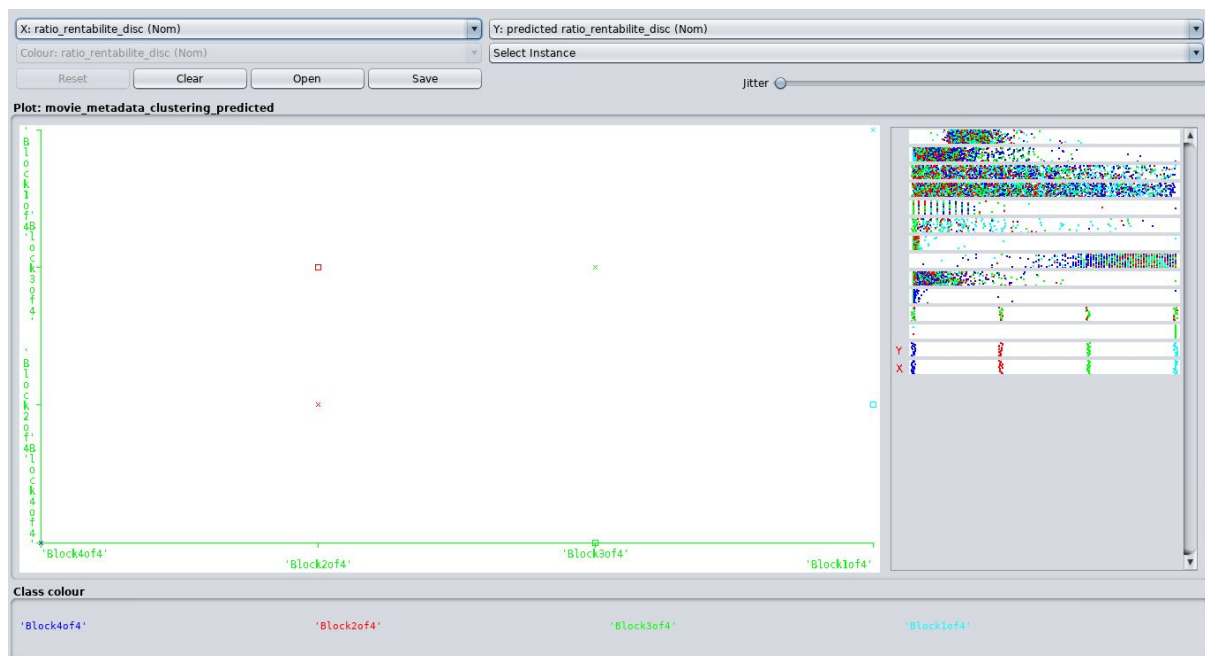
## J48 résultat:



J48 arbre:



J48 erreurs:



## DecisionStump:

## DecisionStump résultat:

The screenshot shows the Orange3 Classifier widget interface. The 'Test options' section on the left has 'Cross-validation' selected with 'Folds' set to 10. The '(Nom) ratio\_rentabilite\_disc' is selected in the dropdown. The 'Result list' on the left shows a list of runs, with '23:21:05 - trees.DecisionStump' selected. The 'Classifier output' pane on the right displays the following results:

Time taken to build model: 0.04 seconds

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

Metric	Value	Percentage
Correctly Classified Instances	1939	49.8329 %
Incorrectly Classified Instances	1952	50.1671 %
Kappa statistic	0.3311	
Mean absolute error	0.2501	
Root mean squared error	0.3537	
Relative absolute error	66.701 %	
Root relative squared error	81.6917 %	
Total Number of Instances	3891	
Ignored Class Unknown Instances	1152	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.200	0.067	0.497	0.200	0.285	0.191	0.880	0.499	'Block4of4'
0.697	0.234	0.498	0.697	0.581	0.420	0.879	0.498	'Block2of4'
0.798	0.267	0.499	0.798	0.614	0.469	0.879	0.499	'Block3of4'
0.299	0.101	0.497	0.299	0.374	0.240	0.880	0.499	'Block1of4'
Weighted Avg.	0.498	0.167	0.498	0.463	0.330	0.880	0.499	

=== Confusion Matrix ===

a	b	c	d	<-- Classified as
194	0	778	0	a = 'Block4of4'
0	678	1	294	b = 'Block2of4'
196	1	776	0	c = 'Block3of4'
0	682	0	291	d = 'Block1of4'

## DecisionStump rentabilité:

The screenshot shows the Orange3 Classifier widget interface. The 'Test options' section on the left has 'Cross-validation' selected with 'Folds' set to 10. The '(Num) ratio\_rentabilite' is selected in the dropdown. The 'Result list' on the left shows a list of runs, with '23:40:59 - trees.DecisionStump' selected. The 'Classifier output' pane on the right displays the following results:

facenumber\_in\_poster  
country  
budget  
title\_year  
movie\_facebook\_likes  
ratio\_rentabilite  
imdb\_score\_disc  
ratio\_rentabilite\_disc

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===  
Decision Stump  
Classifications

actor\_3\_name = amber armstrong : 7194.485533  
actor\_3\_name != amber armstrong : 4.407935784278351  
actor\_3\_name is missing : 3.6054281

Time taken to build model: 0.07 seconds

=== Cross-validation ===  
=== Summary ===

Metric	Value
Correlation coefficient	-0.0449
Mean absolute error	7.545
Root mean squared error	129.6428
Relative absolute error	82.7276 %
Root relative squared error	99.9981 %
Total Number of Instances	3891
Ignored Class Unknown Instances	1152



DecisionStump rentabilité error:

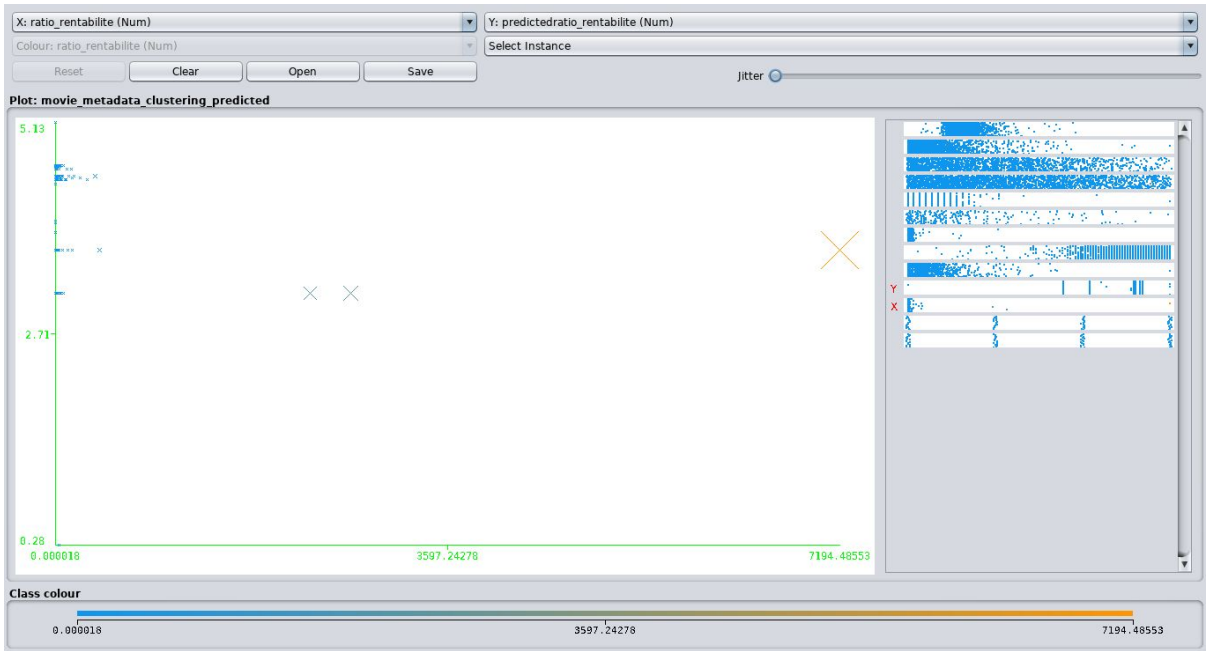


Tableau score IMDB:

	Temps de création du modèle		Taux d'erreur		Meilleur F-mesure	
	Données discrétisés	Données non discrétisés	Données discrétisés	Données non discrétisés	Données discrétisés	Données non discrétisés
Bayes Net	0.02 sec	Non fonctionnel	0.2056%	Non fonctionnel	0.999	Non fonctionnel
Naive Bayes	0.01 sec	Non fonctionnel	61.9671%	Non fonctionnel	0.511	Non fonctionnel
J48	0.13 sec	Non fonctionnel	67.678%	Non fonctionnel	0.402	Non fonctionnel
DecisionStump	0.02 sec	0 sec	65.2984%	Pas de Précision par classe	0.875	Pas de Précision par classe

### Tableau rentabilité:

	Temps de création du modèle		Taux d'erreur		Meilleur F-mesure	
	Données discrétisés	Données non discrétisés	Données discrétisés	Données non discrétisés	Données discrétisés	Données non discrétisés
<b>Bayes Net</b>	0.12 sec	Non fonctionnel	0.2056%	Non fonctionnel	0.999	Non fonctionnel
<b>Naive Bayes</b>	0.03 sec	Non fonctionnel	25.6489%	Non fonctionnel	0.825	Non fonctionnel
<b>J48</b>	0.11 sec	Non fonctionnel	0.0771%	Non fonctionnel	0.999	Non fonctionnel
<b>DecisionStump</b>	0.04 sec	0.07 sec	50.1671%	Pas de Précision par classe	0.798	

### Conclusion

Après analyse du tableau comparatif des résultats obtenus en appliquant chacun des algorithmes présents dans ce dernier.

Le meilleur algorithme pour prédire le **score IMBD** est :

BayesNet car il fournit la meilleur valeur de F-mesure (qui est donc une moyenne entre Precision et Recall) qui est 0.999. Il fournit le plus bas taux d'erreur avec un temps de création de modèle acceptable.

Le meilleur algorithme pour prédire le **ratio de rentabilité** est:

J48 car il fournit la meilleur valeur de F-mesure (qui est donc une moyenne entre Precision et Recall) qui est 0.999. Il fournit le plus bas taux d'erreur avec un temps de création de modèle acceptable.

On peut aussi choisir BayesNet car il est très proche en terme de performance de J48.