Ανάκτηση Πληροφορίας και Μηχανές Αναζήτησης
(Information Retrieval and Search Engines)
(AIE 702)

Προγραμματιστική Εργασία 2 - Κατηγοριοποίηση Κειμένου

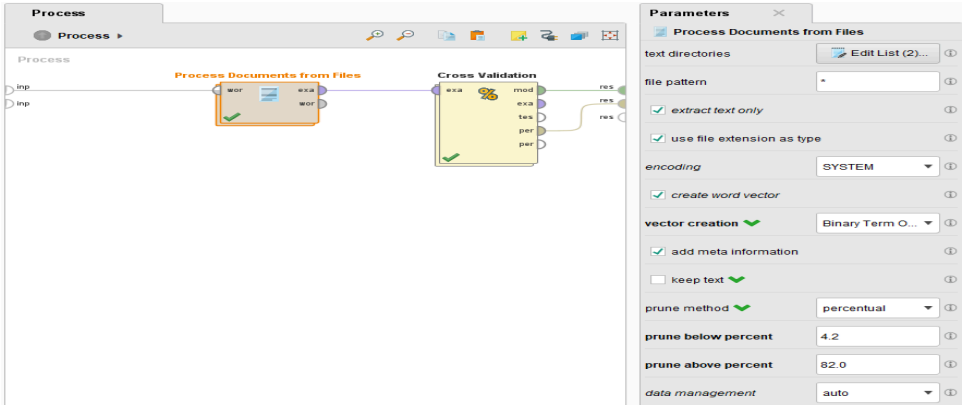
ΟΝΟΥΡ ΙΜΠΡΑΧΗΜ (ics 21007)

Α) Κατηγοριοποίηση των εγγράφων με RapidMiner

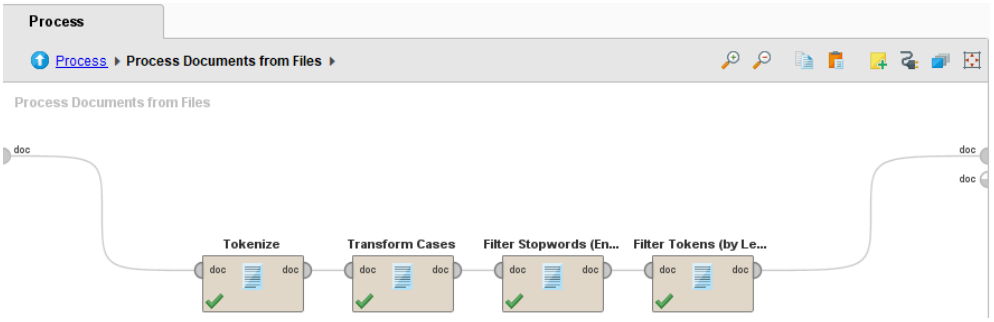
Διαδικασία σχεδίασης του Naive bayes

Εγκαθιστούμε την επέκταση **Text Processing** (από **Extensions > Marketplace > Top Rated > Text Processing 10.0.0**), που είναι απαραίτητη για την ανάλυση κειμένων.

1) **Ανάκτηση δεδομένων** (εγγράφων)
   Προσθήκη του **"Process Documents from Files"** στην καρτέλα Process(Design) και ρύθμιση παραμέτρων :
   a) **Edit list** : προσθήκη κλάσεων pos και neg με τα αντίστοιχα directory φακέλων που περιέχουν της κριτικές ταινιών ως .txt.
   b) **Vector Creation**: Επιλογή μεταξύ binary term occurrences ή term occurrences για τη μετατροπή των εγγράφων σε διανύσματα.
   c) **Prune Method**: percentual με τιμές below percent = 4 και above percent = 82, όπου δίνουν accuracy = 79.75% . Αυτές οι τιμές μειώνουν το χρόνο επεξεργασίας κατα 20 δευτερόλεπτα και βελτιώνουν την ακρίβεια κατα 10 % σε σχέση χωρίς prune method.
   d) **enable parallel execution:** Ενεργοποίηση. Χωρίς το Prune Method: percentual το λογισμικό κράσαρε.



2) **Προεπεξεργασία δεδομένων** (εγγράφων)
   Παρακάτω φαίνεται η προεπεξεργασία των εγγράφων (διπλό κλικ στο Process Documents from Files)



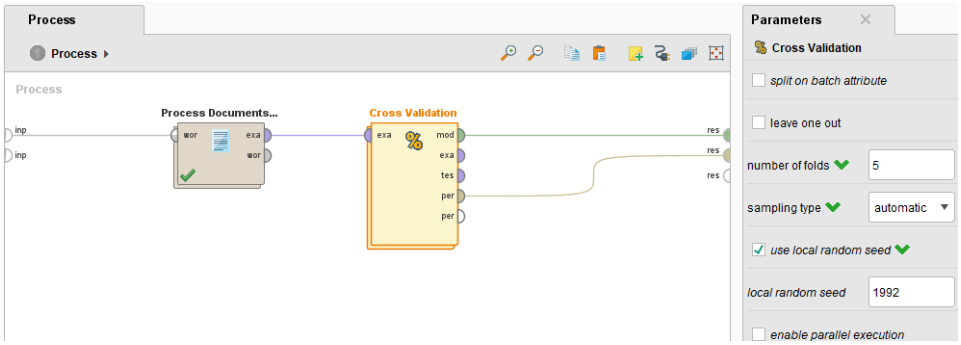
Tokenize: Σπάει το κείμενο σε λέξεις.
Transform Cases: Μετατροπή σε πεζά γράμματα ώστε οι λέξεις να μην διαφοροποιούνται από τη μορφή τους (π.χ., "The"-"the" να θεωρούνται ίδια λέξη).
Filter Stopwords (English): Αφαίρεση λέξεων που δεν έχουν σημαντική πληροφορία (π.χ., "the", "and") για μειώσει το θόρυβο στο κείμενο, επικεντρώνοντας

στην ανάλυση των λέξεων που έχουν μεγαλύτερη σημασία για την κατηγοριοποίηση και την ανάλυση.
Filter Tokens (by Length) (με min chars = 3): Αφαίρεση tokens που είναι πολύ μικρά ή πολύ μεγάλα γιατί είναι λιγότερο χρήσιμα.

Δεν πρόσθεσα το **Generate n-Grams (Word N-Grams),** παρόλο που αποκαλύπτει περισσότερες σχέσεις μεταξύ των λέξεων, γιατί το RapidMiner κατέρρεε με parallel execution. Χωρίς το parallel execution, η διαδικασία έπαιρνε 16 λεπτά με accuracy 76.50%. Με percentual prune method, ο χρόνος έπεφτε στα 12 δευτερόλεπτα και το accuracy ανέβαινε στο 79.90%. Τελικά, χωρίς n-Grams, με μόνο percentual prune method και τα υπόλοιπα, ο χρόνος μειωνόταν στα 2 δευτερόλεπτα με 79.75% accuracy (διαφορά 0.20%). Έτσι διάλεξη **να μην χρησιμοποιήσω** το Generate n-Grams μία που έχει πολύ μικρή διαφορά στο accuracy.

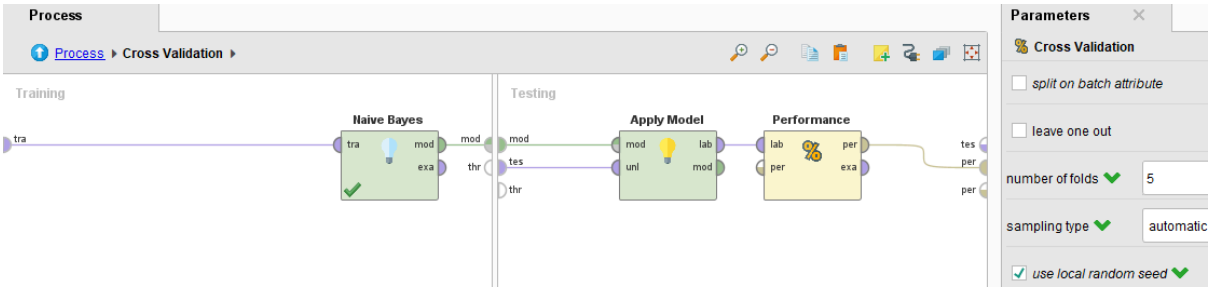
Μετά την ανάκτηση και προεπεξεργασία γίνεται εκπαίδευση με 5-fold cross validation και κατηγοριοποίηση με Naive Bayes Classifier με Laplace Correlation (για να χειριστούμε μηδενικές πιθανότητες).

1) **Μεθοδος εκπαίδευσης 5-fold cross validation**
   Προσθήκη του “Cross Validation” απο την καρτέλα Operators στην σχεδίαση Process και σύνδεση της εξόδου example set (exa) από τον Process Documents from Files στην είσοδο training set (tra) του Cross Validation Operator.
   Στους παραμέτρους, επιλογή:
   a) **number of folds 5**, σημαίνει ότι τα δεδομένα θα χωριστούν σε 5 ισομεγέθη τμήματα (folds) και η διαδικασία εκπαίδευσης θα πραγματοποιηθεί σε 4 επαναλήψεις και το σετ ελέγχου/δοκιμής (test set) σε 1.
   b) **sampling type ως “automatic”** , αφού τα δεδομένα είναι ισορροπημένα (ίσος αριθμός εγγράφων θετικών και αρνητικών)
   c) **Use local random seed** ενεργοποιημένη, έτσι χρησιμοποιείται το ίδιο τυχαίο seed σε κάθε εκτέλεση, με αποτέλεσμα η δειγματοληψία κατά τη διάρκεια του cross-validation να είναι η ίδια κάθε φορά.



2) **Κατηγοριοποίηση με Naive Bayes Classifier**
   Μέσα στο Cross Validation υπάρχουν 2 μέρη, το Training και το Testing.
   Στην καρτέλα :
   a) **Training**, προσθήκη του κατηγοριοποιητή Naive Bayes και ενεργοποίηση ‘laplace correction’ απο την καρτέλα Parameters.
   b) **Testing**, Προσθήκη του :
      i) **Apply Model**, που χρησιμοποιείται για να εφαρμόσει το μοντέλο που έχει εκπαιδευτεί.

ii) **Performance**, που αξιολογεί την απόδοση του μοντέλου με βάση τις προβλέψεις που παράχθηκαν από το "Apply Model".
Επιλογή από την καρτέλα Parameters του Performance της μετρήσης accuracy, weighted mean recall και weighted mean precision.



Σύνδεση:

a) Τα δεδομένα εκπαίδευσης (tra) εισάγονται στο Naive Bayes για εκπαίδευση, και το εκπαιδευμένο μοντέλο (mod) εξάγεται στην είσοδο (mod) της καρτέλας Testing.
b) Το εκπαιδευμένο μοντέλο (mod) εισάγονται στο (mod) του Apply Model για να παράγει προβλέψεις (lab). Και η έξοδος (tes) στο (unl) του Apply Model.
c) Οι έξοδος προβλέψεις (lab) συνδέεται στην είσοδο (lab) του Performance όπου συγκρίνονται και εξάγονται τα αποτελέσματα αξιολόγησης στην έξοδο (per) της καρτέλας Training.

Αποτελέσματα Μοντέλου με Διαφορετικές Μεθόδους Vector Creation :

1. Αποτελέσματα με Vector Creation: **Binary Term Occurrences**
   a. Ακρίβεια (Accuracy)**: 79.75%** ± 2.35% (micro average: 79.75%)
   b. Confusion Matrix**:**

	True Neg	True Pos
Pred. Neg	833	238
Pred. Pos	167	762

   i. **833 αρνητικά** παραδείγματα ταξινομήθηκαν σωστά ως αρνητικά (TN).
   ii. **238 θετικά** παραδείγματα ταξινομήθηκαν λανθασμένα ως αρνητικά (FN).
   iii. **762 θετικά** παραδείγματα ταξινομήθηκαν σωστά ως θετικά (TP).
   iv. **167 αρνητικά** παραδείγματα ταξινομήθηκαν λανθασμένα ως θετικά (FP).
   c. Weighted Mean Recall**: 79.75% ± 2.35%**
   d. Weighted Mean Precision**: 79.94% ± 2.44%**

2. Αποτελέσματα με Vector Creation: **Term Occurrences**
    a. Ακρίβεια (Accuracy): **73.40%** ± 1.59% (micro average: 73.40%)
    b. Confusion Matrix:

	True Neg	True Pos
Pred. Neg	820	352
Pred. Pos	180	648

        i. **820 αρνητικά** παραδείγματα ταξινομήθηκαν σωστά ως αρνητικά (TN).
        ii. **352 θετικά** παραδείγματα ταξινομήθηκαν λανθασμένα ως αρνητικά (FN).
        iii. **648 θετικά** παραδείγματα ταξινομήθηκαν σωστά ως θετικά (TP).
        iv. **180 αρνητικά** παραδείγματα ταξινομήθηκαν λανθασμένα ως θετικά (FP).
    c. Weighted Mean Recall: **73.40%** ± 1.59%
    d. Weighted Mean Precision: **74.13%** ± 1.53%

Σύγκριση των Μεθόδων Vector Creation

- **Accuracy:** Το μοντέλο με τη μέθοδο **Binary Term Occurrences** παρουσιάζει υψηλότερη ακρίβεια κατά 6.35%, γεγονός που υποδεικνύει καλύτερη απόδοση σε σχέση με τη μέθοδο Term Occurrences.
- **Precision:** Η μέθοδος **Binary Term Occurrences** είχε επίσης υψηλότερη απόδοση κατά περίπου 5.81%, δείχνοντας μεγαλύτερη ακρίβεια στην ταξινόμηση των θετικών παραδειγμάτων.
- **Recall:** Επίσης υψηλότερο για τη μέθοδο Binary Term Occurrences, επιβεβαιώνοντας ότι η συγκεκριμένη μέθοδος είναι καλύτερη στην αναγνώριση θετικών παραδειγμάτων.
- **Confusion Matrix**:
    - Με τη μέθοδο **Binary Term Occurrences**, το μοντέλο ταξινόμησε σωστά περισσότερα θετικά και αρνητικά παραδείγματα.
    - Με τη μέθοδο **Term Occurrences**, το μοντέλο είχε περισσότερα σφάλματα, με 358 αρνητικά παραδείγματα να ταξινομούνται λανθασμένα ως θετικά και 189 θετικά να ταξινομούνται ως αρνητικά.
    - Η μέθοδος **Binary Term Occurrences** εμφανίζει λιγότερα σφάλματα συνολικά, καθώς έχει 238 FN και 167 FP, σε σύγκριση με τα 352 FN και 180 FP της μεθόδου **Term Occurrences**.

Β) Κατηγοριοποίηση των εγγράφων με κώδικα python

Εισαγωγή

Ο κώδικας χρησιμοποιεί την ίδια τεχνική προεπεξεργασίας που χρησιμοποιήθηκε στο RapidMiner. Υλοποιείται η κατηγοριοποίηση κειμένων με τη χρήση του αλγορίθμου Naive Bayes σε δύο διαφορετικές εκδοχές: **Multinomial Naive Bayes** με βάση τις εμφανίσεις όρων (Term Occurrences) και **Bernoulli Naive Bayes** με βάση δυαδικές εμφανίσεις όρων (Binary Term Occurrences).

Ο κώδικας πρώτα φορτώνει τα δεδομένα από φακέλους, όπου τα θετικά και αρνητικά κείμενα είναι αποθηκευμένα σε ξεχωριστούς φακέλους. Στη συνέχεια, τα δεδομένα διαχωρίζονται σε σύνολα εκπαίδευσης και δοκιμής (50% για το καθένα όπως ήταν και στο RapidMiner), και προετοιμάζονται τα διανύσματα χαρακτηριστικών χρησιμοποιώντας τον αλγόριθμο **CountVectorizer**, με την:

1. Αφαίρεση κοινών λέξεων (stopwords)
2. Φιλτράρισμα λέξεων ανά μήκος (Filter Tokens by Length με min_chars=3 όπως ήταν και στο Rapidminer)
3. Μετατροπή σε πεζά (Transform Cases) και
4. Τον καθορισμό συγκεκριμένου ελάχιστου και μέγιστου ποσοστού εμφάνισης όρων (percentual pruning)
   (min_df = 0.04 (4%) & max_df = 0.82(82%) όπως ήταν και στο Rapidminer)

Επιπρόσθετα, ο κώδικας δίνει τη δυνατότητα ελέγχου παραπάνων τιμών.

Τα δύο μοντέλα εκπαιδεύονται και δοκιμάζονται πάνω στα σύνολα δεδομένων, μετρώντας τη συνολική ακρίβεια (accuracy), τη μήτρα σύγχυσης (Confusion Matrix), και τις επιδόσεις τους μέσω αναφοράς κατάταξης (Classification Report). Επιπλέον, καταγράφεται ο χρόνος εκτέλεσης για κάθε μοντέλο, παρέχοντας μία ολοκληρωμένη αξιολόγηση της απόδοσης κάθε προσέγγισης.

Συνολικά, η διαδικασία αξιολογεί τα μοντέλα με μετρικές όπως η ακρίβεια, η μήτρα σύγχυσης και η αναφορά κατάταξης, ενώ καταγράφει τον χρόνο εκτέλεσης για κάθε μοντέλο, ώστε να δοθεί μια πλήρης εικόνα της απόδοσης κάθε μεθόδου κατηγοριοποίησης.

Για περισσότερες πληροφορίες δείτε το readme.md αρχείο.

Παρακάτω υπάρχουν τα αποτελεσματα απο των κώδικα και οι συγκρισης του **Multinomial Naive Bayes** και **Bernoulli Naive Bayes.**

Αποτελέσματα Bernoulli & Multinomial Naive Bayes Classifier

1) Multinomial Naive Bayes Classifier με term occurrences

```
==================================================
Multinomial Naive Bayes with Term Occurrences
==================================================
Execution Time: 0.0000 seconds (with loading data: 0.1694 seconds)

Confusion Matrix:
[[384 116]
 [110 390]]

Accuracy: 0.7740

Classification Report:
              precision    recall  f1-score   support

         neg     0.7773    0.7680    0.7726       500
         pos     0.7708    0.7800    0.7753       500

    accuracy                         0.7740      1000
   macro avg     0.7740    0.7740    0.7740      1000
weighted avg     0.7740    0.7740    0.7740      1000
```

a) **Ακρίβεια (Accuracy)**: **77.40%**

b) **Confusion Matrix:**

	True Neg	True Pos
Pred. Neg (neg)	384	110
Pred. Pos (pos)	116	390

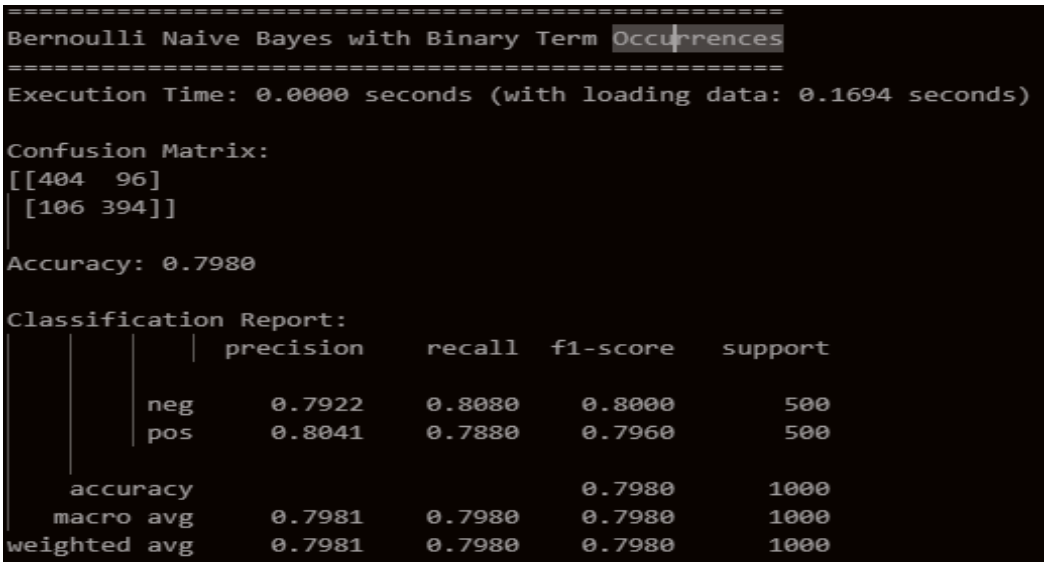
i) **384 αρνητικά** παραδείγματα ταξινομήθηκαν σωστά ως αρνητικά (TN).
ii) **110 θετικά** παραδείγματα ταξινομήθηκαν λανθασμένα ως αρνητικά (FN).
iii) **390 θετικά** παραδείγματα ταξινομήθηκαν σωστά ως θετικά (TP).
iv) **384 αρνητικά** παραδείγματα ταξινομήθηκαν λανθασμένα ως θετικά (FP).

c) **Recall:**
   i) **neg**: 77.73%
   ii) **pos**: 77.08%

d) **Precision:**
   i) **neg:** 76.80%
   ii) **pos:** 78.00%

e) **F1-Score:**
   i) **neg:** 77.26%
   ii) **pos:** 77.53%

2) **Bernoulli Naive Bayes με Binary term occurrences**



a) **Ακρίβεια (Accuracy)**: **79.80%**
b) **Confusion Matrix:**

	True Neg	True Pos
Pred. Neg (neg)	404	106
Pred. Pos (pos)	96	394

   i) **404 αρνητικά** παραδείγματα ταξινομήθηκαν σωστά ως αρνητικά (TN).
   ii) **106 θετικά** παραδείγματα ταξινομήθηκαν λανθασμένα ως αρνητικά (FN).
   iii) **394 θετικά** παραδείγματα ταξινομήθηκαν σωστά ως θετικά (TP).
   iv) **96 αρνητικά** παραδείγματα ταξινομήθηκαν λανθασμένα ως θετικά (FP).

c) **Recall:**
   i) **neg**: 80.80%
   ii) **pos**: 78.80%
d) **Precision:**
   i) **neg:** 79.22%
   ii) **pos:** 80.41%
e) **F1-Score:**
   i) **neg:** 80.00%
   ii) **pos:** 79.60%

Σύγκριση Bernoulli & Multinomial Naive Bayes Classifier

Το **Bernoulli Naive Bayes Classifier υπερτερεί του Multinomial Naive Bayes Classifier** σε όλες σχεδόν τις κατηγορίες:

1. **Ακρίβεια (Accuracy)**: Το Bernoulli Naive Bayes έχει **2.40%** υψηλότερη ακρίβεια (79.80%), που σημαίνει ότι κάνει λιγότερα λάθη στις προβλέψεις σε σχέση με το Multinomial Naive Bayes (77.40%).
2. **Confusion Matrix**: Το Bernoulli Naive Bayes έχει **5.21%** περισσότερα **True Negatives** (404 έναντι 384) και **17.24%** λιγότερα False Positives (96 έναντι 116), υποδεικνύοντας ότι το Bernoulli είναι καλύτερο στην αναγνώριση αρνητικών παραδειγμάτων. Επίσης, έχει **0.77%** περισσότερα True Positives (394 έναντι 390), κάτι που το καθιστά ελαφρώς πιο ακριβές στις θετικές προβλέψεις.
3. **Precision:** Το Bernoulli Naive Bayes έχει **1.49%** καλύτερο precision για την κατηγορία neg (79.22% έναντι 77.73%) και **3.33%** για την κατηγορία pos (80.41% έναντι 77.08%). Αυτό δείχνει ότι το Bernoulli κάνει λιγότερα λάθη στις θετικές προβλέψεις. Αυτό δείχνει ότι το Bernoulli κάνει λιγότερα λάθη στις προβλέψεις θετικών παραδειγμάτων.
4. **Recall:** Το Bernoulli Naive Bayes έχει **4.00%** καλύτερο recall για την κατηγορία neg (80.80% έναντι 76.80% του Multinomial), ενώ το Multinomial Naive Bayes έχει ελαφρώς καλύτερο recall με διαφορά **0.80%** στην κατηγορία pos (78.00% έναντι 78.80% του Bernoulli).
5. **F1-Score:** Το Bernoulli Naive Bayes έχει κατα **2.74%** μεγαλύτερο F1-Score για την κατηγορία neg (**80.00%** έναντι **77.26%**) και **2.07%** για την κατηγορία pos (**79.60%** έναντι **77.53%**). Αυτό δείχνει καλύτερη ισορροπία μεταξύ precision και recall.

	TN	**FP**	**FN**	**TP**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Multinomial Naive Bayes	384	116	110	390	77.40%	77.73%	76.80%	77.26%
Bernoulli Naive Bayes	404	96	106	394	79.80%	79.22%	80.80%	80.00%

Συμπερασματικά, το Bernoulli Naive Bayes Classifier παρουσιάζει καλύτερη συνολική απόδοση από το Multinomial Naive Bayes Classifier.

Σύγκριση Python και Rapidminer

- **Bernoulli Naive Bayes με Binary Term Occurrences:**
  - Ακρίβεια: **79.80% (Python)**, **79.75% (RapidMiner)**
  - Καλύτερο σε precision, recall, και F1-score.
  - Ισορροπημένη ταξινόμηση αρνητικών και θετικών παραδειγμάτων.
- **Multinomial Naive Bayes με Term Occurrences:**
  - Ακρίβεια: **77.40% (Python)**, **73.40% (RapidMiner)**
  - Χαμηλότερη απόδοση, περισσότερα λάθη στις θετικές κατηγορίες.

Το **Bernoulli Naive Bayes** είναι πιο αποδοτικό και αξιόπιστο τόσο στο Python όσο και στο RapidMiner αλλά 2 φορές πιό γρήγορο στο Python με εκτέλεση ολόκληρου του κώδικα και των 2 Naive Bayes **σε 2 δευτερόλεπτα.(** `[Done] exited with code=0 in 2.141 seconds.` Ενώ στο Rapidminer το καθένα κρατούσε 2 δευτερόλεπτα, άρα 4 δευτερόλεπτα τα 2.