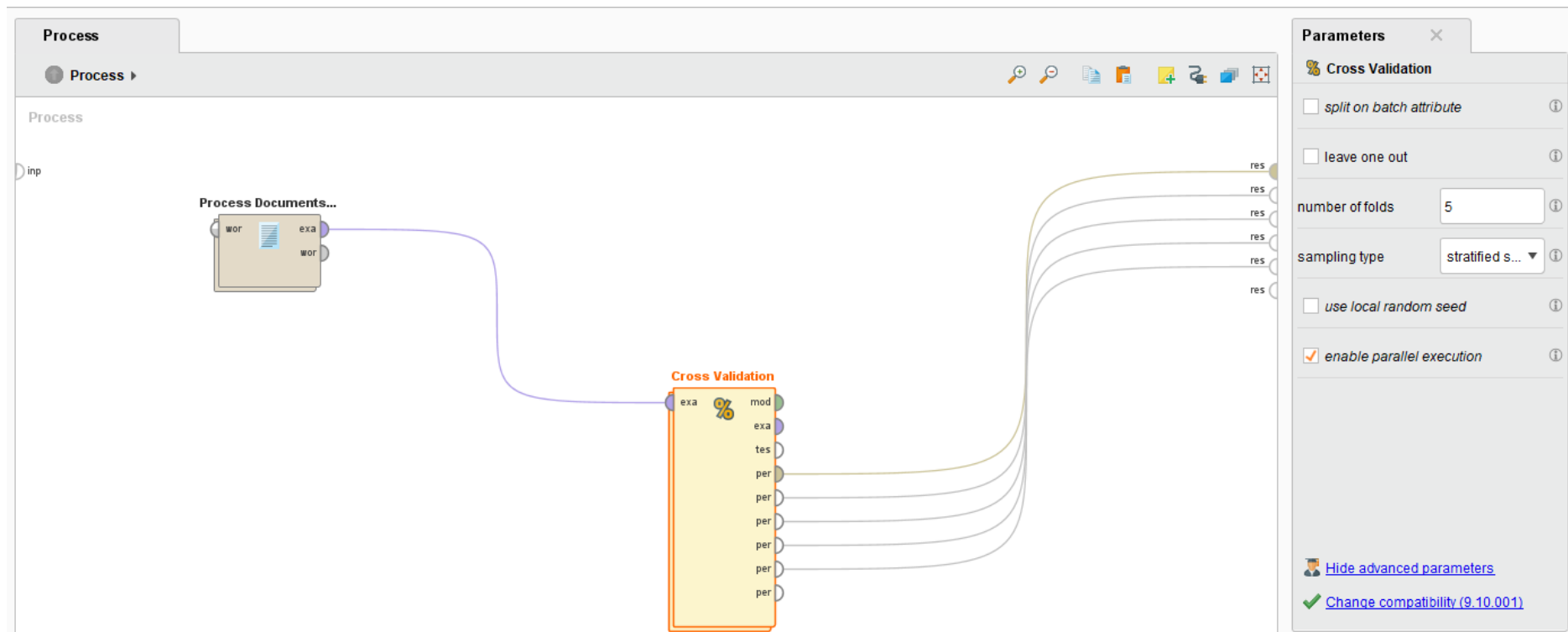


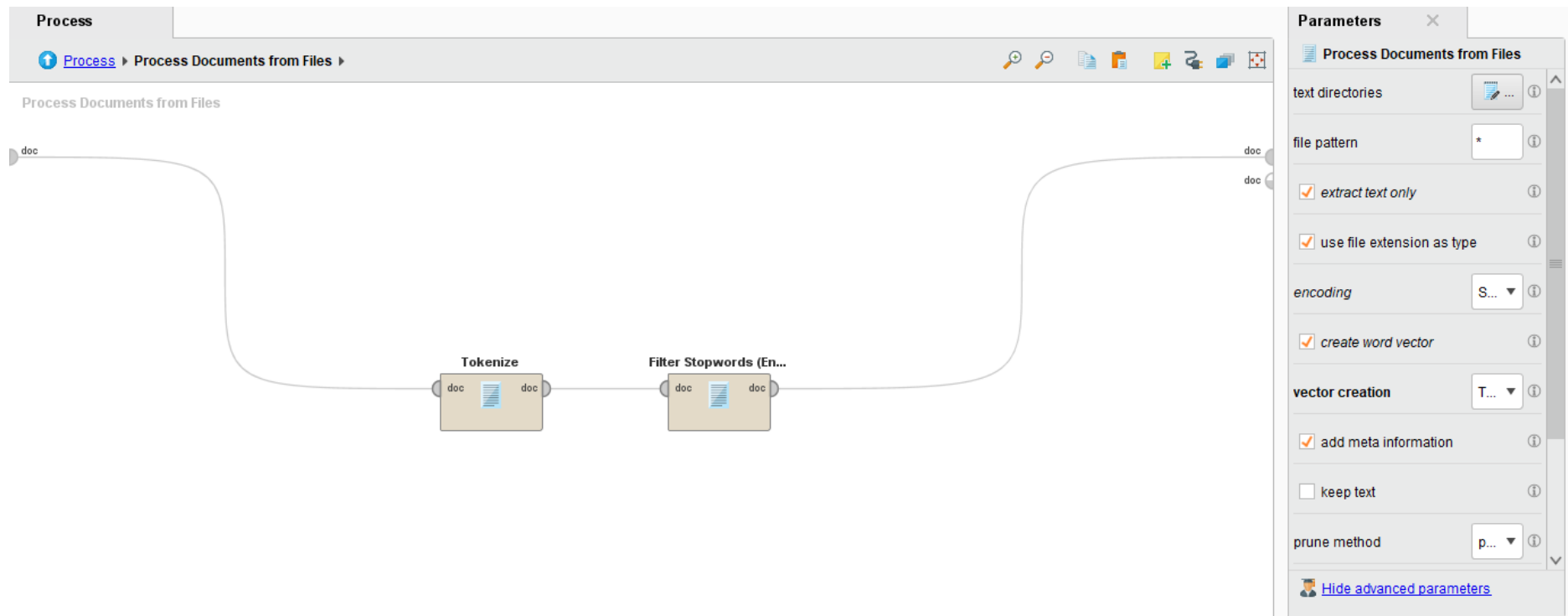
Πρόβλημα 1 :

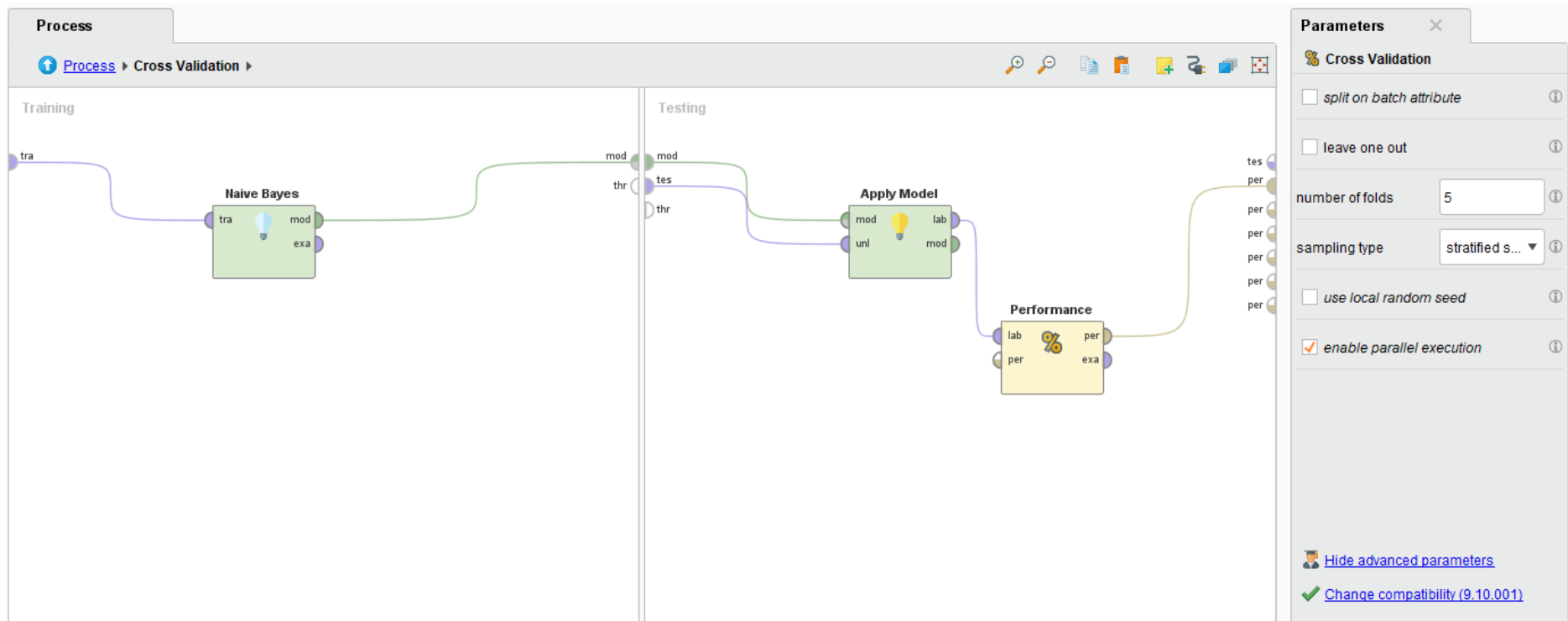
Παρουσίαση της τελικής διαδικασίας προεπεξεργασίας για τη βέλτιστη απόδοση :

Στην αρχή , έσυρα μέσα στο pipeline το Process Document from files . Στη συνέχεια με διπλό κλικ σε αυτό , από το εσωτερικό μενού και μέσω του Edit List πρόσθεσα τις κλάσεις και τα αντίστοιχα δεδομένα που μου δόθηκαν . Μετά από πολλές δοκιμές κατέληξα στο συμπέρασμα ότι τα εξής φίλτρα , Tokenize(non letters) και Filter stopwords(Eng) πετυχαίνουν την βέλτιστη απόδοση σε συνδυασμό με prune below 4%(κόβοντας δηλαδή τις λέξεις που εμφανίζονται σε κάτω από το 4% των εγγράφων) και prune above 96%(κόβοντας δηλαδή τις λέξεις που εμφανίζονται σε πάνω από το 96% των εγγράφων) και τέλος επέλεξα **Vector : term occurrences** . Έπειτα , αφού πέρασα τα φίλτρα βγήκα πάλι στο αρχικό pipeline και πρόσθεσα το Cross Validation και συγκεκριμένα με 5 fold όπως και ζητήθηκε. Ξανά με διπλό κλικ στο Cross Validation , πρόσθεσα τον Naïve Bayes στο training set και σύνδεσα την έξοδο του (mod) με την είσοδο(mod) του Apply Model στο testing , καθώς και το tes στην είσοδο του Apply Model (unl). Τέλος σύνδεσα την έξοδο(lab) του Apply Model με την είσοδο(lab) του Performance και την έξοδο (per) αυτού με το per του testing . Για την εξαγωγή αποτελεσμάτων , απλά πάτησα το “Run”.

Παρακάτω φαίνεται και η διαδικασία που προανέφερα και σε εικόνες .







Παρακάτω , παραθέτω τα αποτελέσματα της βέλτιστης απόδοσης της οποίας πέτυχα :

Στο αρχείο εξέλ (Αποτελέσματα 1^{ου} Προβλήματος) έχω συμπεριλάβει και τις μετρήσεις απόδοσης μη βέλτιστων παραμετροποιήσεων προκειμένου να αποφανθεί και να τεκμηριωθεί η επιλογή της διαδικασίας προεπεξεργασίας με τη βέλτιστη απόδοση. Δεν έχω να σχολιάσω κάτι περαιτέρω όσο αναφορά με τις διάφορες τιμές pruning και τα είδη των διαφορετικών φίλτρων καθώς υπάρχουν τα αποτελέσματα από όλους τους διάφορους συνδυασμούς στο αρχείο εξέλ .

Βάση των παρακάτω αποτελεσμάτων , την πιο ευδιάκριτη κατηγορία αποτελεί η κατηγορία sport . Ανάλογα με τα διαφορετικά είδη vector αυξομειώνονται τα αποτελέσματα (recall,precision) των κατηγοριών . Για παράδειγμα , στην περίπτωση που έχουμε vector : TF/IDF με τις ίδιες τιμές pruning σε σχέση με την παραπάνω βέλτιστη διαδικασία έχουμε μικρότερο class recall και λίγο μεγαλύτερο class precision ομοίως και για τις υπόλοιπες κατηγορίες προκύπτουν αυτές οι διάφορες αυξομειώσεις .



Views:

Design

Results

Turbo Prep

Auto Model

Deployments

Result History

 PerformanceVector (Performance) ×

Performance



Description



Annotations

Criterion

accuracy

weighted mean recall

weighted mean preci...

☒ Table View ☐ Plot View

accuracy: 92.27% +/- 1.32% (micro average: 92.27%)

	true business	true entertainment	true politics	true sport	true tech	class precision
pred. business	460	12	17	6	11	90.91%
pred. entertainment	7	331	4	8	4	93.50%
pred. politics	19	6	394	8	4	91.42%
pred. sport	1	5	1	486	0	98.58%
pred. tech	23	32	1	3	382	86.62%
class recall	90.20%	85.75%	94.48%	95.11%	95.26%	

FileEditProcessViewConnectionsSettingsExtensionsHelp

Views:DesignResultsTurbo PrepAuto ModelDeployments

Result History

% PerformanceVector (Performance) X

%
Performance

Description

Annotations

Criterion

accuracy

weighted mean recall

weighted mean preci...

Table View

Plot View

weighted_mean_recall: 92.16% +/- 1.38% (micro average: 92.16%), weights: 1, 1, 1, 1, 1

	true business	true entertainment	true politics	true sport	true tech	class precision
pred. business	460	12	17	6	11	90.91%
pred. entertainment	7	331	4	8	4	93.50%
pred. politics	19	6	394	8	4	91.42%
pred. sport	1	5	1	486	0	98.58%
pred. tech	23	32	1	3	382	86.62%
class recall	90.20%	85.75%	94.48%	95.11%	95.26%	



Views:

Design

Results

Turbo Prep

Auto Model

Deployments

Result History

PerformanceVector (Performance)



Performance



Description



Annotations

Criterion
accuracy
weighted mean recall
weighted mean preci...

☒ Table View ☐ Plot View

weighted_mean_precision: 92.27% +/- 1.35% (micro average: 92.21%), weights: 1, 1, 1, 1, 1

	true business	true entertainment	true politics	true sport	true tech	class precision
pred. business	460	12	17	6	11	90.91%
pred. entertainment	7	331	4	8	4	93.50%
pred. politics	19	6	394	8	4	91.42%
pred. sport	1	5	1	486	0	98.58%
pred. tech	23	32	1	3	382	86.62%
class recall	90.20%	85.75%	94.48%	95.11%	95.26%	

Πρόβλημα 2 :

Παρουσίαση της τελικής διαδικασίας προεπεξεργασίας για τη βέλτιστη απόδοση :

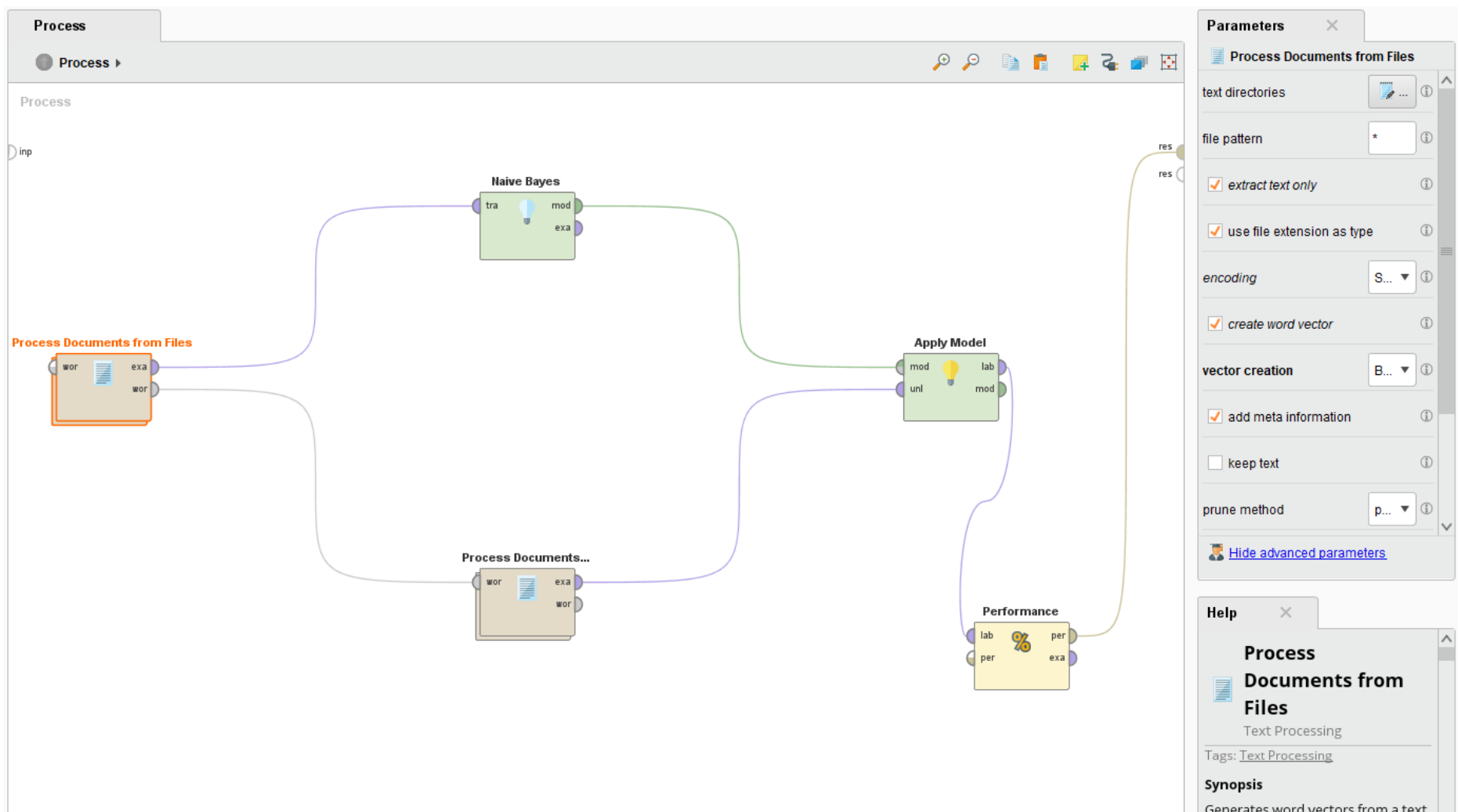
Σε αυτή την περίπτωση η διαδικασία που ακολούθησα είναι η ίδια με παραπάνω με τη διαφορά ότι , χρειάστηκε η δημιουργία ενός δεύτερου Process Documents from files το οποίο έχει τα 200neg και 200pos δεδομένα για testing ενώ το πρώτο έχει τα 800neg και 800 positive για training . Στην ίδια λογική προσθέσαμε τον Naïve Bayes , Apply Model και Performance . Μετά από πολλές δοκιμές κατέληξα στο συμπέρασμα ότι τα εξής φίλτρα , Tokenize(non letters) και Filter stopwords(Eng) πετυχαίνουν την βέλτιστη απόδοση σε συνδυασμό με prune below 4%(κόβοντας δηλαδή τις λέξεις που εμφανίζονται σε κάτω από το 4% των εγγράφων) και prune above 96%(κόβοντας δηλαδή τις λέξεις που εμφανίζονται σε πάνω από το 96% των εγγράφων) και τέλος επέλεξα **Vector : binary term occurrences** .

Παρακάτω , παραθέτω τα αποτελέσματα της βέλτιστης απόδοσης της οποίας πέτυχα και την εικόνα με την προεπεξεργασία :

Στο αρχείο εξέλ (Αποτελέσματα 2^{ου} Προβλήματος) έχω συμπεριλάβει και τις μετρήσεις απόδοσης μη βέλτιστων παραμετροποιήσεων προκειμένου να αποφανθεί και να τεκμηριωθεί η επιλογή της διαδικασίας προεπεξεργασίας με τη βέλτιστη απόδοση. Δεν έχω να σχολιάσω κάτι περαιτέρω όσο αναφορά με τις διάφορες τιμές pruning και τα είδη των διαφορετικών φίλτρων καθώς υπάρχουν τα αποτελέσματα από όλους τους διάφορους συνδυασμούς στο αρχείο εξέλ .

Βάση των παρακάτω αποτελεσμάτων και οι 2 κατηγορίες είναι το ίδιο ευδιάκριτες και σε αυτή την περίπτωση ανάλογα με τα διαφορετικά είδη vector αυξομειώνονται τα αποτελέσματα(recall,precision) των κατηγοριών .

Παρακάτω φαίνεται και η διαδικασία που προανέφερα και σε εικόνες . Ότι φαίνεται στην δεύτερη εικόνα ισχύει και για το Process Documents from files(2) δηλαδή το δεύτερο με την μόνη διαφορά ότι το δεύτερο έχει τα testing δεδομένα ενώ αυτό που φαίνεται στην εικόνα τα training .



Process

Process Documents from Files

Process Documents from Files

doc

Filter Stopwords (En...)

doc

doc

Tokenize

doc

doc

doc

doc

Parameters

Process Documents from Files

text directories

file pattern

☒ extract text only

☒ use file extension as type

encoding

☒ create word vector

vector creation

☒ add meta information

☐ keep text

prune method

Hide advanced parameters

FileEditProcessViewConnectionsSettingsExtensionsHelp

Views:DesignResultsTurbo PrepAuto ModelDeployments

Result History

%

PerformanceVector (Performance)

%

Performance

Description

Annotations

Criterion

accuracy

weighted mean recall

weighted mean preci...

Table View

Plot View

accuracy: 81.75%

	true pos	true neg	class precision
pred. pos	154	27	85.08%
pred. neg	46	173	79.00%
class recall	77.00%	86.50%	

FileEditProcessViewConnectionsSettingsExtensionsHelp

Views:DesignResultsTurbo PrepAuto ModelDeployments

Result History

% PerformanceVector (Performance) X

%

Performance

Description

Annotations

Criterion

accuracy

weighted mean recall

weighted mean preci...


Table View


Plot View


weighted_mean_recall: 81.75%, weights: 1, 1

	true pos	true neg	class precision
pred. pos	154	27	85.08%
pred. neg	46	173	79.00%
class recall	77.00%	86.50%	

Result History

 Performance

 Description

 Annotations

Criterion
accuracy
weighted mean recall
weighted mean preci...

☒ Table View ☐ Plot View

weighted_mean_precision: 82.04%, weights: 1, 1

	true pos	true neg	class precision
pred. pos	154	27	85.08%
pred. neg	46	173	79.00%
class recall	77.00%	86.50%	

Παρατηρούμε και στις 2 περιπτώσεις πως όποια κλάση έχει μεγαλύτερο precision θα έχει μικρότερο recall σε σχέση με την άλλη και το αντίστροφο . Απ'ότι βλέπουμε παραπάνω , προκύπτει ότι στην πρώτη περίπτωση πετυχαίνουμε μεγαλύτερο ποσοστό ακρίβειας σε σχέση με την δεύτερη περίπτωση . Ωστόσο , αυτό δεν είναι απόλυτο διότι παίζει ρόλο το πλήθος των δεδομένων , η ταχύτητα εκτέλεσης και η ποιότητα των δεδομένων . Για παράδειγμα , στην δεύτερη περίπτωση αν είχαμε περισσότερα δεδομένα στο training set πολύ πιθανόν να είχαμε ακόμη μεγαλύτερα ποσοστά ακρίβειας . Παραδόξως , όπως φαίνεται και στα γενικά αποτελέσματα που έχω βάλει στα εξέλ , το πλήθος των φίλτρων δε παίζει καθοριστικό ρόλο καθώς και στις 2 περιπτώσεις πετυχαίνουμε μεγαλύτερα ποσοστά ακρίβειας έχοντας στη διάθεση μας λιγότερα φίλτρα .