# Progress File

Chi squere

Δουλεύει με tf-idf weighting το πήρα έτομο από το sklearn

Χρησιμοποιεί το selectkbest όπως υπάρχει στο sklearn example

RDF

Χρησιμοποιεί το countvectorizer για να κάνει ένα αρχικό relative\_term pool  
Φτιάχνει ένα αρχικό train\_set  
Στην συνέχεια φτιάχνει μια λίστα με τους υποψήφιους προς αποχώρηση όρους και τους αφαιρεί από την αρχικό train\_set .  
Κατόπιν παίρνει το νέο train\_set και το κάνει tf-idf. ?? ( ρώτα αραμπατζή)  
Για να μειώσω τα χαρακτηριστικά βάζω threshold min\_df= 12 δηλαδή θα πρέπει ο όρος να εμφανίζεται τουλάχιστον σε 12 κείμενα relative.

Example\_unifromity\_Cumulative

Χρησιμοποιεί countvectorizer για να πάρει τα relative feat   
Φτιάχνει το timeline το οποίο είναι ουσιαστικά το άθροισμα των relative κειμένων του train\_set  
Το step= 1/ timeline  
Υπόθεση αν ένα χαρακτηριστικό είναι uniform τότε θα εμφανίζεται τουλάχιστον από μια φορά σε κάθε relative document του train\_set  
Οπότε φτιάχνουμε την βέλτιστη συναθροιστική uniform κατανομή η οποία είναι ίσα steps από 0 εώς 1  
Φτιάχνουμε για κάθε feature την δικιά του κατανομή με το εξής κριτήριο :

* Αν έχει occurrence παίρνει κ\*βήμα
* Αν δεν έχει μένει στάσιμο
* Στο τέλος έχει 1

Έπειτα χρησιμοποιώ επαναληπτικά την ks2samp από scipy.stats και παίρνω τα pvalues για κάθε feature.  
Αποθηκεύω τα ζευγάρια p\_value – feature σε νέο dataframe και μετά το κάνω sort . Από εκεί μπορώ να πάρω τους τοπ όρους που θέλω.

FeatureSelection class

Random\_select method   
 Παίρνει σαν όρισμα τον αριθμό των feature ,έπειτα παίρνει όλο το trainset βρίσκει τα feature τα ανακατεύει με την random.shuffle και επιλέγει τα [0:select] . Τέλος αφαιρεί όλα τα άλλα feature και φτιάχνει νέο train set.

Uniform

βήμα=έγγραφο ή μέρα, decision\_thres= πότε θα είναι uniform η μέρα από πόσα doc και πάνω, και topk είναι τα καλύτερα χαρακτηριστικά

timeline είναι ο χρονικός άξονας που κινούνται τα docs. Διαμορφώνεται ανάλογα με το ποιο τρόπο διαλέγουμε το uniformity.

Αν έχουμε σε μέρες τότε κάθε χρονικό σημείο είναι μια μέρα και ανάλογα με το ποσοστό των σχετικών αρχείων για εκείνη την ημέρα και το threshold βγαίνει το αντίστοιχο uniformity.

Αν έχω single σαν βήμα τότε κάθε σημείο στο timeline αναπαρίσταται από ένα text\_entry μέσα στο csv.

Για όλα τα features βρίσκω το αντίστοιχο p\_value , συγκρίνοντας το optimal uniformity με αυτό που έχει. ΠΧ αν έχω σαν timeline μέρα ανα μέρα τότε το όπτιμαλ είναι 365 σκαλοπάτια προς το 1. Αν έχω σαν σημεία τα text\_entries, τότε το timeline είναι το πλήθος των relevant text\_entries και το optimal uniformity είναι μια σκάλα με ίδιο μήκος προς τον 1.

**Ερώτηση :** Θα πρέπει μετά το feature\_selection να αφαιρέσω εκείνα τα documents τα οποία είναι άδεια ?

Τα train είναι 384699 και τα τεστ είναι 419718

Για τα σχετικά train δεδομένα:

count 116.000000

mean 13954.913793

std 27187.457254

min 0.000000

25% 1181.000000

50% 4235.500000

75% 14089.000000

max 184456.000000

Σε ποσοστό όλων των κειμένων είναι 0.03627488970104795.

Για τα σχετικά τεστ δεδομένα:

count 116.000000

mean 15264.586207

std 30063.375386

min 5.000000

25% 1312.000000

50% 4616.500000

75% 15276.250000

max 196866.000000

*Σε ποσοστό όλων των τεστ κειμένων τα σχετικά είναι :* 0.0363686718389408