Όνομα	Επίθετο	Α.Μ
Ένρι	Γκάτση	1115200900048
Gerald	Mema	1115200800108

1. Διευκρινήσεις

Δεν σας στέλνω τα ενδιάμεσα .csv που χρησιμοποιήθηκαν για τη παραγωγή των αποτελεσμάτων καθώς τότε το παραδοτέο θα έφτανε το μέγεθος εκατοντάδων byte.

2. Ερώτημα 1

Αυτό που καταλάβαμε από την εκφώνηση της εργασίας για να δείξουμε αν μπορεί να χρησιμοποιηθεί ο LDA για classification είναι να ακολουθήσουμε τα εξής βήματα.

1. Αρχικά φτιάξαμε 3 φορές(K=10,K=100,K=1000) το LdaModel από το LatantDA.py.
2. Στο ίδιο αρχείο κώδικα βρήκαμε τις πιθανότητες που αναλογούν σε κάθε έγγραφο να ανήκει σε ένα από τα K topics και από αυτές τις πιθανότητες κρατήσαμε την μεγαλύτερη και αντιστοιχίσαμε κάθε έγγραφο με ένα topic.
3. Το παραπάνω αποτέλεσμα το αντικαταστήσαμε με το Y_train και τρέξαμε τους classifier όπως ακριβώς στην πρώτη άσκηση με την διαφορά ότι τώρα ότι έχουμε να τα χωρίσουμε σε 10,100,1000 κατηγορίες.

Τα αποτελέσματα φαίνονται στα αρχεία εξόδου

- EvaluationMetric_10fold_K_10.csv
- EvaluationMetric_10fold_K_100.csv
- EvaluationMetric_10fold_K_1000.csv

Αυτό δηλαδή που θέλαμε να δείξουμε είναι ότι μπορούμε να χρησιμοποιήσουμε σαν classification την μέθοδο LDA αφού τα αποτελέσματα που μας υπέδειξε για κάθε κείμενο να ανήκει σε ένα topic συμφωνούσαν σε μεγάλο βαθμό με τα αποτελέσματα των classifier που είχαμε στην 1η άσκηση. Με K=10 είχαμε τα καλύτερα αποτελέσματα αφού οι classifier είχαν πιο περιορισμένο πεδίο προβλέψεων.

3. Ερώτημα 2

Για το ερώτημα 2 σκεφτήκαμε να χρησιμοποιήσουμε το lda για να κάνουμε model το κάθε κείμενο για να βρούμε σε ποια κατηγορία από αυτές που βρήκε είναι πιο πιθανό να ανήκει το συγκεκριμένο κείμενο. Από αυτή τη πιο πιθανή κατηγορία παίρνουμε τις 20 πιο πιθανές λέξεις και τις ενσωματώνουμε στο αντίστοιχο άρθρο του train_set. Ύστερα κάνουμε 10fold cross validation στο train_set. Τα αποτελέσματα αυτής τη προσπάθειας έχουν καταγραφεί στα αρχεία εξόδου:

- EvaluationMetric_10fold_ex1_10.csv (με 10 topics)
- EvaluationMetric_10fold_ex1_100.csv (με 100 topics)
- EvaluationMetric_10fold_ex1_1000.csv (με 1000 topics)

Εάν κάποιος θέλει να παράγει τα αποτελέσματα μόνος του θα πρέπει

1. Να τρέξει το αρχείο LatentLDA.py και να πειράξει τα categories στην εντολή `ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics=1000, id2word = dictionary)` προκειμένου να παραχθεί το αρχείο ldaWords_1000.
2. Αυτό το αρχείο περιέχει τις λέξεις με που παράγονται με τη λογική που αναφέρουμε πιο πάνω. Ύστερα με αυτό το αρχείο θα πρέπει να τρέξει το LDA_test.py και να το δώσει σαν παράμετρο στη μεταβλητή df2, τότε παράγεται το αρχείο που το όνομα του δίνεται στη γραμμή 55 μέσω της μεθόδου df2.to_csv(*όνομα αρχείου (ας πούμε X_train.csv, sep*)
3. Αυτό το αρχείο περιέχει το παλιό content και του έχουν προστεθεί 20 φορές οι λέξεις που αναφέρουμε στη προηγούμενη παράγραφο. Αυτό το αρχείο πρέπει να δώσει για να διαβάσει η μεταβλητή df2 στο αρχείο MyMethod.py προκειμένου να παράγει το EvaluationMetric_10fold_ex1_1000 που έχει τα τελικά αποτελέσματα.

4. Ερώτημα 3

Παίρνοντας τώρα το αρχείο X_train.csv που περιέχει το content του αρχείου train_set.csv μαζί με τις λέξεις του lda κάνουμε train όλους τους classifier που είδαμε και στην προηγούμενη άσκηση και προβλέπουμε τις κατηγορίες των εγγράφων του test_set. Τα αποτελέσματα έχουν αποθηκευθεί στο αρχείο testSet_categories.csv.