Τεχνικές Εξόρυξης Δεδομένων

Εργασία 1η

Εαρινό Εξάμηνο 2017-18

των φοιτητών Χρήστου-Παρασκευά Δημητρόπουλου και Ιωάννας Ζαπαλίδη

0. Γενικά

Η εργασία αυτή αποτελεί μία πρώτη επαφή με τα βασικά στάδια της διαδικασίας που ακολουθείται για την εφαρμογή τεχνικών εξόρυξης δεδομένων.

Η εργασία αναπτύχθηκε σε περιβάλλοντα Linux (Ubuntu 16.04) και Windows 10, με χρήση MS Visual Studio, Terminal, Sublime Text. Χρησιμοποιήθηκαν, όπως ζητήθηκε, Python 2.7, Anaconda, Scikit-learn, Pandas και NumPy, μαζί με άλλα κατάλληλα frameworks.

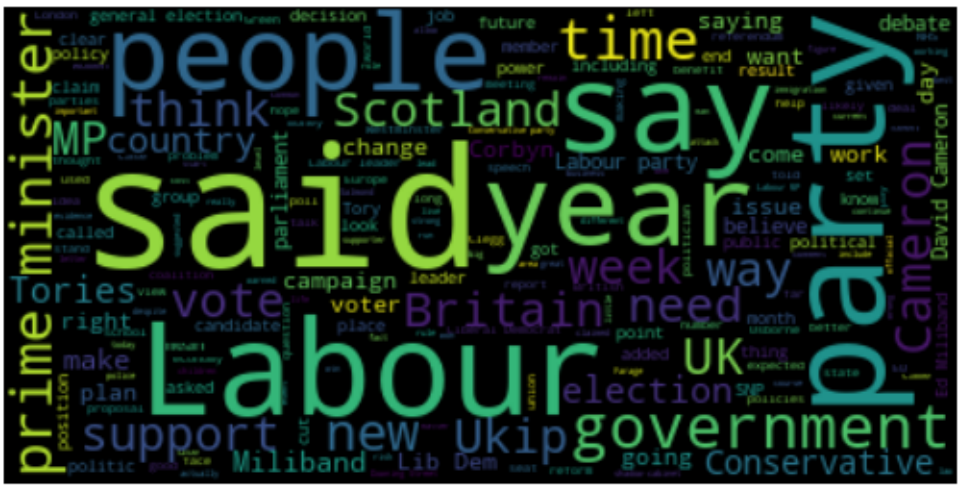
1. WordCloud

Στο terminal, στο directory όπου βρίσκεται η εργασία μας, με την εκτέλεση της εξής εντολής:

```
~/.bashrc
PATH=/home/username/anaconda2/bin:$PATH
python cloud.py
```

προκύπτουν τα ζητούμενα WordClouds ανά κατηγορία άρθρων, ένα ανά κατηγορία.

π.χ. Politics:



Συνοπτικά, ο κώδικάς μας κάνει τα εξής:
Ανοίγουμε το αρχείο train_set.csv, και βρίσκουμε τις κατηγορίες άρθρων που περιέχονται σε αυτό, αλλά και τα περιεχόμενα των άρθρων. Έπειτα, ανα κατηγορία, δημιουργούμε ένα προσωρινό αρχείο τύπου txt, όπου για κάθε άρθρο (άρα κάθε γραμμή) χωρίζουμε τις λέξεις του, αφαιρούμε τις stop words, και όσες διατηρούνται αποθηκεύονται στο αρχείο αυτό. Δημιουργούμε τέλος το αρχείο εικόνας και αυτό αποθηκεύεται στο directory όπου βρίσκεται το executable αρχείο μας.

Έχουμε προσθέσει στη δοθείσα λίστα stop words μερικά δικά μας, όπως “yes”, “just” κοκ.

2. Classification

Υλοποιήθηκε η κατηγοριοποίηση των άρθρων με κάθε ζητούμενο τρόπο.

SVM: Έγινε χρήση του GridSearchCV για αλλαγές στις ζητούμενες παραμέτρους (Kernel, C, gamma). Προκύπτει ότι οι καλύτερες παράμετροι για την εκτέλεση αυτής της μεθόδου είναι (από τις δοθείσες δικές μας – για όλο το πλήθος άρθρων):

Kernel	linear
C	0.1
gamma (γ)	0.001

Συνεπώς, με αυτές τις παραμέτρους εκτελούμε τον έλεγχο 10-fold cross validation.
Random Forests: Η παραμετροποίηση του classifier έχει ως εξής: ελάχιστο πλήθος features ανά μη-φύλλο node, δηλαδή min_samples_split, έχει την τιμή 10, χρησιμοποιήθηκε ως κριτήριο, criterion η τιμή "entropy", η οποία υπολογίζεται εύκολα, και όχι η τιμή gini που είναι η default. Τέλος το πλήθος n_estimators ισούται με 50. Εκτελούμε ξανά τον έλεγχο 10-fold cross validation.
Multinomial Naive Bayes: Χρησιμοποιήθηκαν οι default τιμές της συνάρτησης. Λόγω του ιδίου error που συζητήθηκε στο piazza, έγινε εκ νέου επεξεργασία των δεδομένων, χωρίς να γίνει χρήση LSI., οπότε δεν τυπώνεται γράφημα.
K-Nearest Neighbors (Δική μας υλοποίηση): Υλοποιήθηκε ο αλγόριθμος από εμάς, σε ξεχωριστό αρχείο (customKNN.py). Το predict γίνεται όπως ζητήθηκε, με βάση Majority Voting,

Σημείωση: Σε δικό μας υπολογιστή, παρατηρήθηκε ότι προκύπτει με κάθε εκτέλεση, το warning:

```
DeprecationWarning: This module was deprecated in version
0.18 in favor of the model_selection module into which all
the refactored classes and functions are moved. Also note
that the interface of the new CV iterators are different from
that of this module. This module will be removed in 0.20.
"This module will be removed in 0.20.", DeprecationWarning)
```

καθώς και το εξής:

```
DeprecationWarning: The truth value of an empty array is
ambiguous. Returning False, but in future this will result in
```

```
an error. Use `array.size > 0` to check that an array is not
empty.
  if diff:
```

χωρίς να δημιουργηθεί κανένα πρόβλημα στην εκτέλεση του προγράμματός μας τελικά. (Για τη μη εμφάνιση των warnings αυτών μπορεί η εκτέλεση να γίνει ως εξης:

```
python -W ignore classify.py
```

στο αντίστοιχο directory.

Για να δωθεί επιπλέον βαρύτητα στην πληροφορία του τίτλου ανά άρθρο, κάνουμε fit το μοντέλο μας με το dataset myData το οποίο ορίζεται ως εξής:

```
myData = train_data['Content'] + 5 * train_data['Title']
```

δηλαδή δίνουμε 5 φορές μεγαλύτερη βαρύτητα στον τίτλο από ότι στο περιεχόμενο του εκάστοτε άρθρου.

Τα ζητούμενα γραφήματα για την προεπεξεργασία των δεδομένων προκύπτουν από την εκτέλεση του ιδίου αυτού αρχείου και αποθηκεύονται στο directory όπου εργαζόμαστε, αλλά και προϋπάρχουν κατά το ανέβασμα της εργασίας. Κάθε διάγραμμα έχει ως άξονα Χ το πληθος των components, και ως άξονα Υ τις τιμές της ακρίβειας, ενώ τιτλοφορείται καταλλήλως. (Για την μέθοδο Multinomial Naive Bayes, δεν δημιουργείται γράφημα καθώς η διαδικασία αυτή, λόγω error, δεν επιδεχόταν προεπεξεργασία των data (LSI)). Με το πέρας των ελέγχων αυτών, εκτελούμε ξανά, με το ιδανικό component number, την κάθε μέθοδο (πλην του Naive Bayes, προφανώς), για να κρατήσουμε το τελικό, καλύτερο αποτέλεσμα.

Επίσης, να σημειωθεί ότι λόγω των πολλαπλών ελέγχων που γίνονται ανά μέθοδο, η διαδικασία αυτή είναι αρκετά χρονοβόρα, ακόμα και για μικρό κομμάτι των δεδομένων μας.

3. Beat the Benchmark

Στον κώδικα του αρχείου customKNN.py, φαίνεται η προσπάθειά μας για την υλοποίηση του ερωτήματος αυτού, καθώς τα samples του training set είναι ισομοιρασμένα ανά κατηγορία.

4. Αρχεία Εξόδου

Με την κλήση του classify.py όπως στο βήμα (2), δημιουργούνται τα ζητούμενα tab separated values αρχεία **EvaluationMetric_10fold.csv** και **testSet_categories.csv**.

Και τα δύο αρχεία ακολουθούν το ζητούμενο format, και εσωκλείονταιι στο zip της εργασίας μας. Δημιουργούνται στο κατώτερο σημείο του κώδικά μας στο αρχείο classify.py

Στο αρχείο EvaluationMetric_10fold.csv, οι τιμές των μετρικών τυπώνονται με ακρίβεια τεσσάρων δεκαδικών ψηφίων, ως ποσοστά επί τοις εκατό.

Για το *testSet_categories.csv*, να σημειωθεί ότι οι κατηγορίες που βρέθηκαν και αποδόθηκαν στα διάφορα άρθρα προκύπτουν από την τελευταία υλοποιημένη method, δηλαδή αυτή που βελτιστοποιήσαμε στο ερώτημα *3*.

5. Βιβλιογραφία

http://scikit-learn.org/stable/documentation.html
https://medium.com/machine-learning-101
https://classroom.udacity.com/courses/ud120
http://stackoverflow.com
https://geeksforgeeks.org/
http://mccormickml.com/2016/03/25/lsa-for-text-classification-tutorial/
https://docs.python.org
https://machinelearningmastery.com/