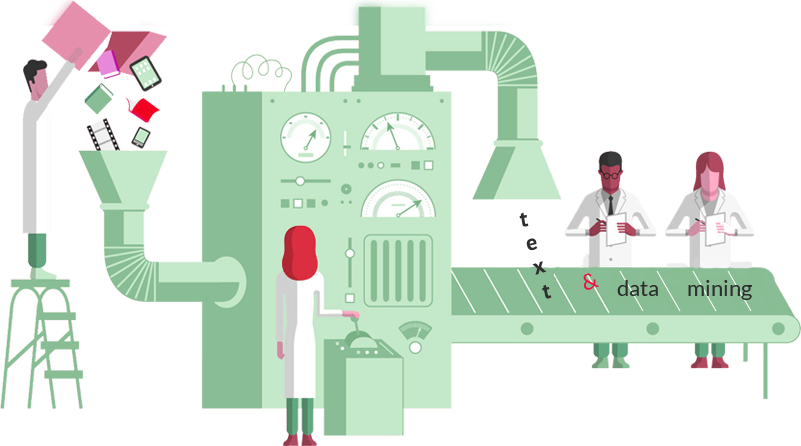
# Τεχνικές Εξόρυξης Δεδομένων: Εργασία 2η



Μέλη Ομάδας:

|  |  |  |
| --- | --- | --- |
| **Όνομα:** | **Επώνυμο:** | **Αριθμός Μητρώου:** |
| Αντωνία | Αθανασάκου | 1115201400004 |
| Στεφανία | Πάτσου | 1115201400156 |

Περιεχόμενα

[Τεχνικές Εξόρυξης Δεδομένων: Εργασία 2η 1](#_Toc483331094)

[Πρόλογος 3](#_Toc483331095)

[Δημιουργία Οπτικοποιημένων Δεδομένων 3](#_Toc483331096)

[Υλοποίηση Κατηγοριοποίησης (Classification) 3](#_Toc483331097)

[Support Vector Machines (SVM) 4](#_Toc483331098)

[Random Forests 4](#_Toc483331099)

[Naive Bayes 4](#_Toc483331100)

[Υλοποίηση Κατηγοριοποίησης στο “test.tsv ” 4](#_Toc483331101)

[Υπολογισμός Information Gain 4](#_Toc483331102)

[Επίλογος / Σχόλια 4](#_Toc483331103)

## Πρόλογος

Σε αυτή την εργασία κληθήκαμε να υλοποιήσουμε οπτικοποίηση δεδομένων ενός δοκιμαστικού αρχείου “train.tsv” και κατηγοριοποίηση του αρχείου “test\_set.csv” με βάση το δοκιμαστικό. Ακόμη, υπολογίσματε το information gain για κάθε feature και σχεδιάσαμε τα κατάλληλα γραφήματα. Αρχικά δημιουργήσαμε, για κάθε attribute του αρχείου “ train.tsv ”, ένα box plot/histogram ανάλογα με το αν το feature ήταν numerical/categorical αντίστοιχα. Στη συνέχεια, με βάση τα καλύτερα ποσοστά ακρίβειας, επιλέξαμε έναν αλγόριθμο κατηγοριοποίησης και τον εκτελέσαμε στο αρχείο “test.tsv”. Τέλος, με βάση το information gain σχεδιάσαμε το γράφημα των accuracies με βάση την αφαίρεση features αλλά και δημιουργήσαμε έναν πίνακα “infoGain.csv”.

## Δημιουργία Οπτικοποιημένων Δεδομένων

Για τη δημιουργία των οπτικοποιημένων δεδομένων, αποθηκεύσαμε τις στήλες του ‘Label’ σε good και bad. Ύστερα, από αυτές και για κάθε attribute, δημιουργήσαμε τα box plots και histograms. Αξίζει να σημειωθεί ότι πριν από αυτό το βήμα, είχαμε χωρίσει τις κατηγορίες σε numerical και categorical. Για την δημιουργία των box plots καλέσαμε την συνάρτηση .boxplot της βιβλιοθήκης matplotlib και με κατάλληλες προσθαφαιρέσεις ιδιοτήτων φτάσαμε στο ιδανικό αποτέλεσμα. Η ίδια διαδικασία επαναλήφθηκε για την δημιουργία των histograms μόνο που χρησιμοποιήσαμε την συνάρτηση value\_counts() η οποία είναι μέρος της βιβλιοθήκης pandas.Series και η οποία επιστρέφει μία λίστα που περιλαμβάνει τους το σύνολο της κάθε κατηγορίας του feature.

## Υλοποίηση Κατηγοριοποίησης (Classification)

Για την κατηγοριοποίηση των δεδομένων, δοκιμάσαμε τις μεθόδους Support Vector Machines (SVM), Random Forests και Naive Bayes. Χρησιμοποιήσαμε “10-fold Cross Validation”(στην οποία στην παράμετρο cv = k\_fold εγίνε με βάση το Kfold) για να αξιολογήσουμε την μετρική “Accuracy”. Τα αποτελέσματα των μετρικών έχουν τοποθετηθεί στο “EvaluationMetric\_10fold.csv”. Ο πίνακας για τις μετρικές φαίνεται και παρακάτω:

|  |  |  |  |
| --- | --- | --- | --- |
| **Statistic Measure** | **Naive Bayes** | **Random Forest** | **SVM** |
| Accuracy | 0.71375 | 0.725 | 0.70125 |

Όπως προκύπτει από τον πίνακα, τα πιο αποδοτικά αποτελέσματα δίνει ο αλγόριθμος Random Forest.

Προσθέσαμε, επίσης, σε κάθε γραμμή ως πρώτο στοιχείο το ‘ ’ για να φαίνονται καλύτερα τα αποτελέσματα.

### Support Vector Machines (SVM)

Η βιβλιοθήκη που χρησιμοποιήσαμε είναι η sklearn και συγκεκριμένα η svm.

### Random Forests

Χρησιμοποιήθηκε η βιβλιοθήκη RandomForestClassifier.

### Naive Bayes

Χρησιμοποιήθηκε η βιβλιοθήκη GaussianNB.

## Υλοποίηση Κατηγοριοποίησης στο “test.tsv ”

Από τα αποτελέσματα της κατηγοριοποίησης, καταλήξαμε ότι πιο αποδοτικός ήταν ο αλγόριθμος RandomForestClassifier. ‘Ετσι εφαρμόσαμε αυτόν τον αλγόριθμο για να προβλέψουμε τις κατηγορίες του αρχείου “test.tsv”. Τα αποτελέσματα υπάρχουν σε ένα αρχείο “ testSet\_Predictions.csv”.

## Υπολογισμός Information Gain

## Επίλογος / Σχόλια

Όλα τα αρχεία με τα αποτελέσματα είναι τοποθετημένα στον φάκελο “Output ”, έτσι ώστε να έχετε τη δυνατότητα να τα εξετάσετε, χωρίς να απαιτείται να εκτελέσετε το πρόγραμμα.