

# ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ 5

### Neural Networks

Σκοπός της συγκεκριμένης εργασίας είναι η εξοικείωση με την υλοποίηση του αλγορίθμου Multi-Layer Perceptron, που περιλαμβάνει τη δημιουργία τεχνητών Νευρωνικών Δικτύων (N.N., Neural Networks), που αποτελούνται από Perceptrons, μέσω της αντίστοιχης μεθόδου που παρέχει η βιβλιοθήκη sci-kit της Python.

Δημιουργήθηκαν διάφορα N.N., δίνοντας διάφορες τιμές στις χαρακτηριστικές παραμέτρους που επηρεάζουν την απόδοσή τους. Η εφαρμογή τους έγινε με χρήση του DataSet “Beast Cancer”, της βιβλιοθήκης sklearn. Τα δεδομένα, πριν εισαχθούν στα N.N. υφίστανται προεπεξεργασία (preprocessing), μέσω της κλάσης MinMaxScaler() της βιβλιοθήκης sklearn, η οποία τα κανονικοποιεί μετατρέποντάς τα στην κλίμακα (0,1). Η αξιολόγηση και η σύγκριση της απόδοσης των διαφορετικά παραμετροποιημένων N.N. έγινε, σύμφωνα με τις οδηγίες της εκφώνησης, με χρήση των ακόλουθων μετρικών (στον υπολογισμό των μετρικών επιλέχθηκε τιμή παραμέτρου `average='macro'`):

- **precision**: Ορίζεται ως  $TP / (TP+FP)$
- **recall**: Ορίζεται ως  $TP / (TP+FN)$
- **f1**: Ορίζεται ως  $2 * precision * recall / (precision + recall)$

#### A. Περιγραφή και μελέτη του αλγορίθμου

Σύμφωνα με τις απαιτήσεις της εκφώνησης δημιουργήθηκαν διάφορα N.N. διαφοροποιούμενα στις τιμές των παρακάτω παραμέτρων:

- **hidden\_layer\_sizes**: Καθορίζει το πλήθος και το μέγεθος των κρυφών επιπέδων. Χρησιμοποιήθηκαν από 1 έως και 3 κρυφά επίπεδα, διαφόρων μεγεθών.
- **solver**: Ο επιλυτής που είναι υπεύθυνος για τη βελτιστοποίηση των βαρών. Χρησιμοποιήθηκαν και οι 3 διαθέσιμες από τη βιβλιοθήκη επιλογές.
- **tol**: (Tolerance), αντιστοιχεί σε ένα κατώφλι. Αν η βελτίωση στην επίδοση του N.N. δεν το ξεπεράσει για έναν αριθμό 10 (default τιμή) συνεχόμενων κύκλων-εποχών, τότε θεωρείται ότι το N.N. έχει συγκλίνει. Χρησιμοποιήθηκαν διάφοροι μικροί δεκαδικοί αριθμοί.

Παράλληλα ορίστηκε σταθερή τιμή για τις παρακάτω παραμέτρους:

- **activation\_function**: ‘relu’.
- **maximun\_itations**: 100

Τα διάφορα N.N. εκτελέστηκαν διαδοχικά, με παράλληλη καταγραφή των τιμών των μετρικών. Ο κώδικας υλοποίησης και εκτέλεσης του αλγορίθμου υπάρχει στο συνοδευτικό αρχείο *Iliapikos\_ge5\_NN.py*. Η μεταβολή των τιμών των ζητούμενων μετρικών φαίνεται στο συνοδευτικό αρχείο *NN\_Results.xlsx*.

#### B. Σύγκριση των παραλλαγών του αλγορίθμου

Βλέποντας συγκριτικά τα αποτελέσματα, παρατηρώ τα παρακάτω αναφορικά με την επίδραση των διαφόρων παραμέτρων στην επίδοση του αλγορίθμου:

- **hidden\_layer\_sizes**: Διατηρώντας όλες τις άλλες παραμέτρους σταθερές, παρατηρώ ότι η αύξηση στο πλήθος ή/και το μέγεθος των κρυφών επιπέδων μειώνει την απόδοση του αλγορίθμου στο συγκεκριμένο DataSet.

- **solver**: Αν και μεταβάλλεται ανάμεσα στις διάφορες υλοποιήσεις η τιμή του `hidden_layer_sizes` και `tol`, παρατηρώ ότι τα καλύτερα αποτελέσματα τα δίνει ο `'sgd'`, χρησιμοποιώντας μάλιστα το μικρότερο N.N., αποτελούμενο από ένα κρυφό επίπεδο 10 Perceptrons. Ο διπλασιασμός στο μέγεθος του κρυφού επιπέδου ρίχνει δραστικά την επίδοση. Πρακτικά το ίδιο καλή επίδοση παρουσιάζει και ο `'adam'`, με χρήση επίσης περιορισμένου μεγέθους N.N., ενώ φαίνεται να μένει ανεπηρέαστος από τη ραγδαία αύξηση στο πλήθος και το μέγεθος των κρυφών επιπέδων. Αισθητά μειωμένη επίδοση παρουσιάζει ο `'lbfgs'`, ανεξάρτητα από το μέγεθος του χρησιμοποιούμενου N.N.
- **tol**: Δεν φαίνεται να επηρεάζει με κάποιο τρόπο την επίδοση των N.N.

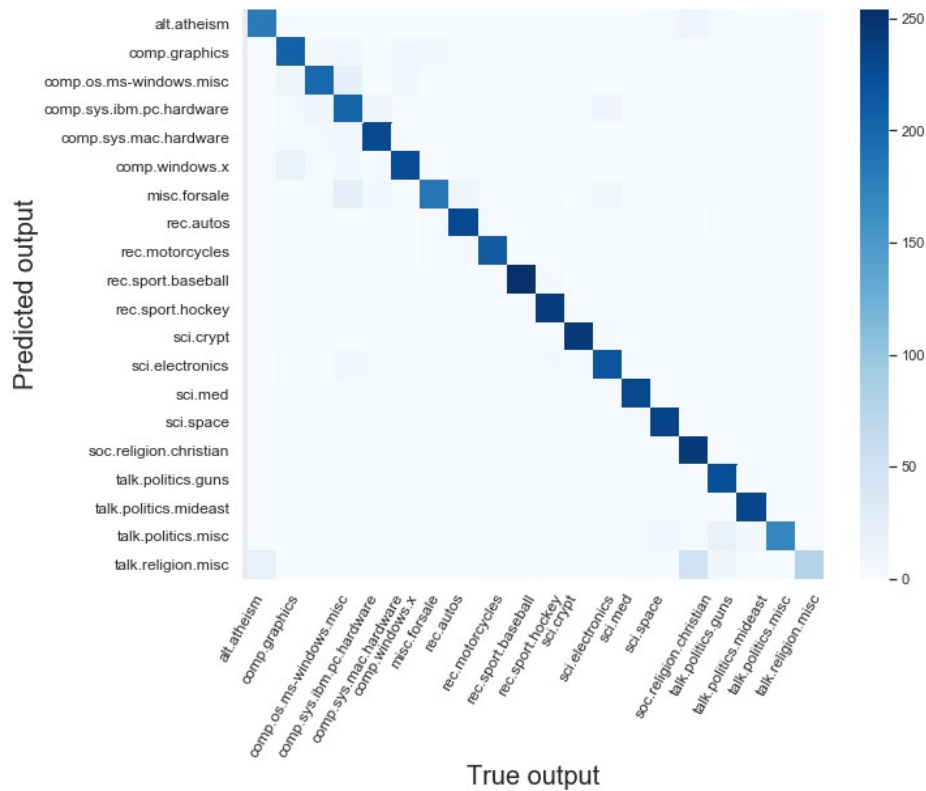
## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Υλικό μαθήματος.
2. Müller, A.C & Guido, S. - Introduction to Machine Learning with Python. A Guide for Data Scientists, O' Reilly, 2017.
3. Τεκμηρίωση από τον ιστότοπο της βιβλιοθήκης Sklearn.

## CONFUSION MATRIX ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

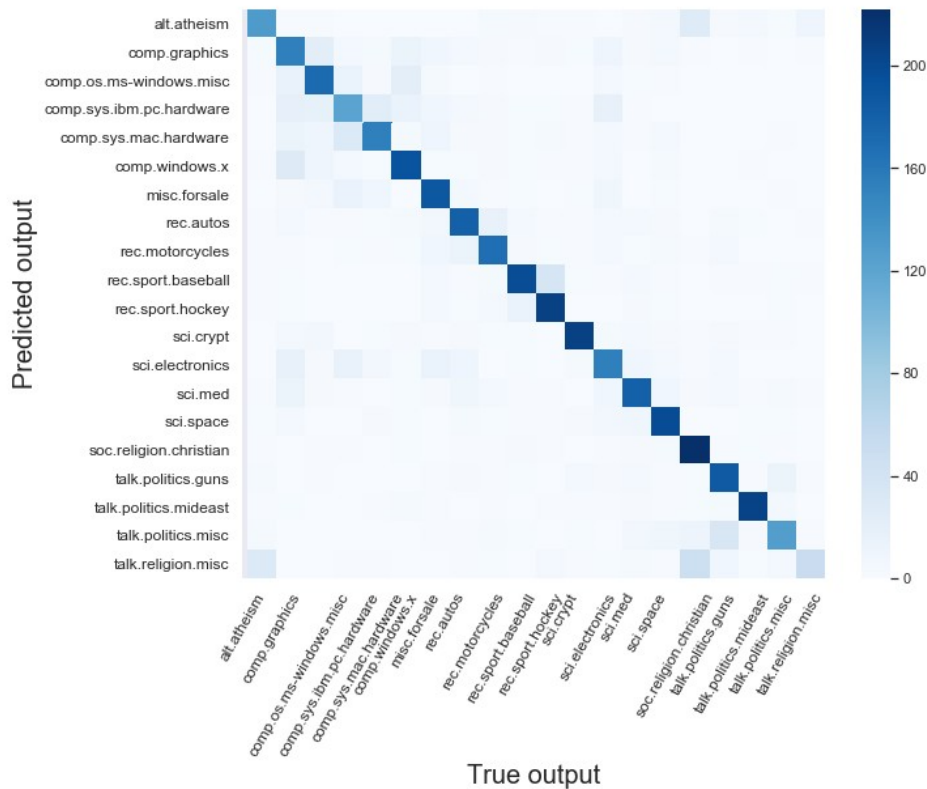
Default διαμόρφωση αλγορίθμου:

Multinomial NB - Confusion matrix ( $\alpha = 0.10$ ) [Prec = 0.90727, Rec = 0.89296, F1 = 0.89358]



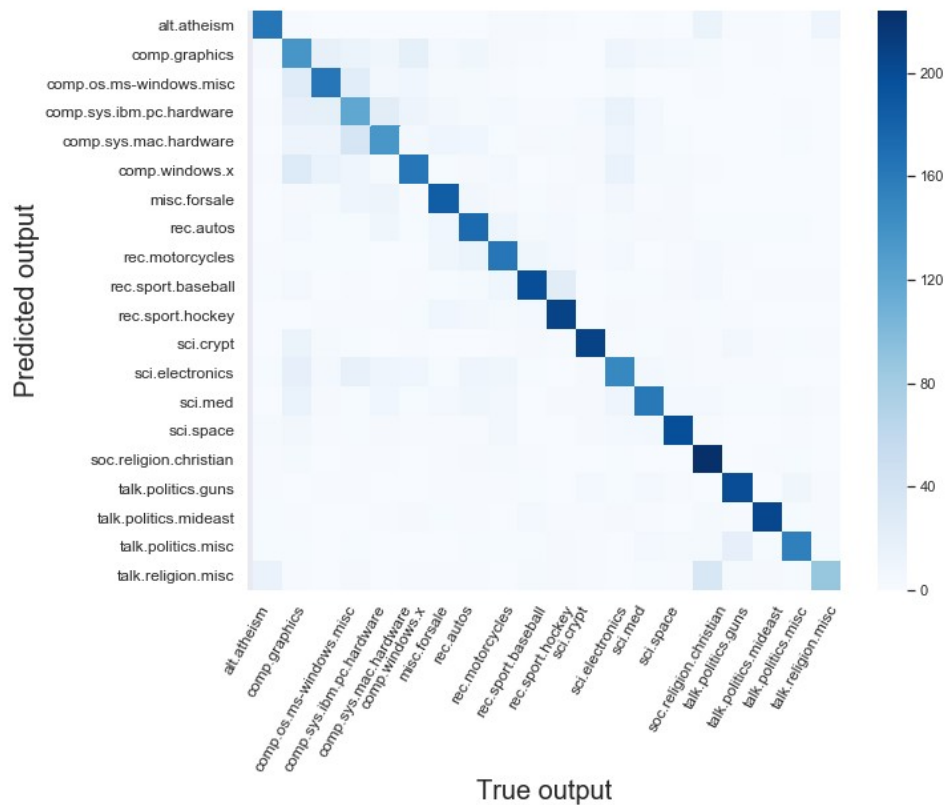
Παραμετροποιημένος αλγόριθμος: max\_features=1000, ngram\_range= (1,1)

Multinomial NB - Confusion matrix ( $\alpha = 0.10$ ) [Prec = 0.72424, Rec = 0.71322, F1 = 0.71211]



Παραμετροποιημένος αλγόριθμος: max\_features=5000, ngram\_range= (2,2)

Multinomial NB - Confusion matrix ( $\alpha = 0.10$ ) [Prec = 0.72906, Rec = 0.71829, F1 = 0.72048]



Παραμετροποιημένος αλγόριθμος: max\_features=12500, ngram\_range= (3,3)

Multinomial NB - Confusion matrix (a = 0.10) [Prec = 0.71411, Rec = 0.70156, F1 = 0.70508]

