

# ΤΕΧΝΙΚΗ ΑΝΑΦΟΡΑ 2<sup>ης</sup> ΕΡΓΑΣΙΑΣ

Μάθημα: Ανάλυση Δεδομένων

Παναγιώτα Βίτσα  
Τμήμα Ψηφιακών Συστημάτων  
Πανεπιστήμιο Πειραιά  
Πειραιάς, Ελλάδα  
[peggyvitsa@yahoo.com](mailto:peggyvitsa@yahoo.com)

## ΕΙΣΑΓΩΓΗ

Η εργασία είναι μία προσπάθεια για την εύρεση της καταλληλότερης εφαρμογής αλγορίθμου κατηγοριοποίησης δεδομένων σε ένα dataset, δηλαδή ουσιαστικά σκοπός είναι η σύγκριση αλγορίθμων.

## ΠΕΡΙΛΗΨΗ

Σκοπός της εργασίας είναι να βρούμε τον πιο ιδανικό αλγόριθμο κατηγοριοποίησης ενός συνόλου δεδομένων.

Για αρχή λοιπόν, ανοίγουμε το αρχείο. Στη συνέχεια πρέπει να ορίζουμε πόσα δεδομένα θα χρησιμοποιήσουμε και πόσα από αυτά θα ορίσουμε στο πρόγραμμα να προβλέψει. Έπειτα, ξεκινάμε να υλοποιούμε 3 αλγόριθμους κατηγοριοποίησης, ώστε να βρούμε τον πιο ιδανικό. Με τη βοήθεια συναρτήσεων υπολογίζουμε τα accuracy, precision, recall και f1 score.

## ΠΕΡΙΓΡΑΦΗ ΣΥΝΟΛΟΥ ΔΕΔΟΜΕΝΩΝ

Αυτό το σύνολο δεδομένων αποτελείται από πολλά αυτοκίνητα, και πιο συγκεκριμένα: τα χαρακτηριστικά αυτών των αυτοκινήτων. Κάθε αυτοκίνητο έχει τα εξής 6 χαρακτηριστικά, τα οποία αναγράφονται και σε σχετικό αρχείο που επιλέγουμε να κατεβάσουμε για να έχουμε καλύτερη κατανόηση των δεδομένων. Αποτελείται από:

- o buying – εύρος τιμής (low, med, high, vhigh)
- o maint – τιμή συντήρησης (low, med, high, vhigh)
- o doors – πλήθος πορτών (2, 3, 4, 5-more)
- o persons – πλήθος ανθρώπων που χωρά (2, 4, more)
- o lug\_boot – μέγεθος πορτ παγκαζ (small, med, big)
- o safety – εκτιμώμενη ασφάλεια αυτοκινήτου (low, med, high)

Παρατηρούμε πως επίσης υπάρχει μία ακόμα στήλη, 7<sup>η</sup>, αυτή της κατανομής κλάσης (αριθμός παρουσιών ανά κλάση) η οποία είναι το βασικό χαρακτηριστικό. Αυτή μπορεί να έχει τιμές unacc, acc, good, vgood.

## ΒΗΜΑΤΑ ΠΡΟ-ΕΠΕΞΕΡΓΑΣΙΑΣ

Ρίχνοντας μία ματιά στο σύνολο δεδομένων μας κρίνουμε ότι δεν είναι απαραίτητα κάποιες μορφές βήματα προ-επεξεργασίας.

## ΛΕΞΕΙΣ – ΚΛΕΙΔΙΑ

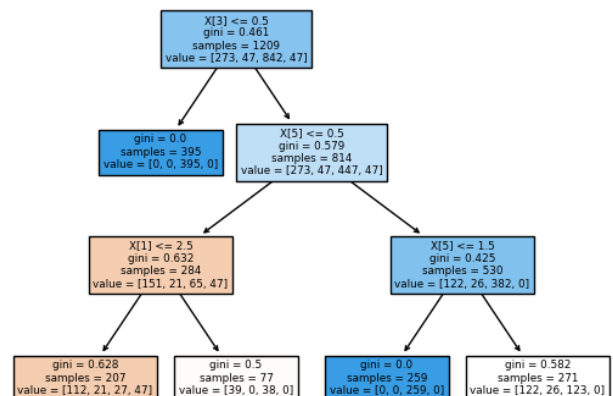
βάση δεδομένων, σύνολο δεδομένων, δεδομένα αυτοκινήτων, κατηγοριοποίηση, αλγόριθμοι κατηγοριοποίησης, πειραματική αξιολόγηση, δέντρο αποφάσεων, κ-κοντινότεροι γείτονες, naïve bayes.

## 1 ΑΛΓΟΡΙΘΜΟΙ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ

Έχοντας ήδη ορίσει ποια είναι τα `x_test`, `x_train`, `y_test` και `y_train` με τη βοήθεια της συνάρτησης `train_test_split` [πηγή 1], ξεκινάμε την υλοποίηση των παρακάτω αλγορίθμων.

### 1.1 ΔΕΝΤΡΟ ΑΠΟΦΑΣΕΩΝ

Ο πρώτος αλγόριθμος που θα υλοποιήσουμε είναι αυτός του δέντρου αποφάσεων. Με τη χρήση του `DecisionTreeClassifier` και παράμετρο το `gini` ως μέτρο επιλογής γνωρισμάτων φτιάχνουμε το παρακάτω δέντρο. Το εκτυπώνουμε στον χρήστη με τη βοήθεια του `plot_tree` και `show`. Επίσης εκτυπώνουμε τα `accuracy`, `precision`, `recall`



και f1 score, με τη χρήση αντίστοιχων συναρτήσεων, τα οποία βγαίνουν 0.77. [πηγή 2]

## 1.2 K-KONTINOTEROI ΓΕΙΤΟΝΕΣ

Με τη βοήθεια του KNeighborsClassifier εφαρμόζουμε τον αλγόριθμο και κάνουμε τη πρόβλεψη όπως και προηγουμένως με το predict. Στη συνέχεια εκτυπώνουμε τα accuracy, precision, recall και f1 score, με τον ίδιο τρόπο όπως και πριν. Αυτά βγάζουν τα αποτελέσματα 0.91. Το μόνο που διαφέρει είναι τα δεδομένα πρόβλεψης. [πηγή 3]

## 1.3 NAÏVE BAYES

Με τη βοήθεια του GaussianNB εφαρμόζουμε τον αλγόριθμο και κάνουμε τη πρόβλεψη όπως και προηγουμένως με το predict. Στη συνέχεια εκτυπώνουμε τα accuracy, precision, recall και f1 score, με τον ίδιο τρόπο όπως και πριν. Αυτά βγάζουν τα αποτελέσματα 0.67. Το μόνο που διαφέρει είναι τα δεδομένα πρόβλεψης. [πηγή 3]

## 2 ΜΕΘΟΔΟΛΟΓΙΑ

Για αρχή, ορίζουμε σε μία λίστα τα ονόματα των στηλών, και έπειτα ανοίγουμε το αρχείο προσθέτοντας ως παράμετρο τα ονόματα αυτά. Εφαρμόζουμε μία μορφή κωδικοποίησης στις στήλες. [πηγή 4]

Έπειτα πρέπει να χωρίσουμε τα δεδομένα από τα κομμάτια που θέλουμε να προβλεπτούν.

Οπότε πρέπει να φτιάξουμε μία λίστα με τα ονόματα των στηλών εκτός της τελευταίας, εφ' όσον ένα κομμάτι της τελευταίας θα βάλουμε να προβλέψουν οι αλγόριθμοι (συγκεκριμένα το 30%).

Σε ένα σύνολο δεδομένων X βάζουμε τις 6 στήλες, ενώ στην y την 7<sup>η</sup>.

Με τη βοήθεια μίας συνάρτησης train\_test\_split διαχωρίζουμε τα δεδομένα. Τα x\_train και x\_test είναι το 70% και 30% αντίστοιχα των δεδομένων του αλγόριθμου των πρώτων 6 στηλών. Κάτι αντίστοιχο ισχύει και για τα y\_train και y\_test για τη τελευταία στήλη.

Ξεκινώντας τον αλγόριθμο του δέντρου αποφάσεων, κάνουμε χρήση του DecisionTreeClassifier και βάζουμε ως παράμετρο το gini ως μέτρο επιλογής γνωρισμάτων, και ως 2<sup>η</sup> παράμετρο το 3 ως το ύψος του δέντρου.

Στη συνέχεια κάνουμε τη πρόβλεψη με το predict και παράμετρο δεδομένων το x\_test, δηλαδή το 30% των δεδομένων των 6 πρώτων στηλών, και την αποθηκεύουμε στο y\_pred.

Για τον υπολογισμό του accuracy, precision, recall και f1 score [πηγές 5, 6, 7, 8] κάνουμε χρήση συναρτήσεων με παραμέτρους πάντα τα δεδομένα που θέλουμε να συγκρίνουμε, δηλαδή την πρόβλεψη y\_pred και τα δεδομένα y\_test. Αυτό πραγματοποιείται με τον ίδιο ακριβώς τρόπο και στους 3 classifiers.

Τέλος εκτυπώνουμε το δέντρο στον χρήστη.

Στη συνέχεια ακολουθεί ο classifier k-neighbors. Με τη συνάρτηση KNeighborsClassifier γίνεται η συγκεκριμένη κατηγοριοποίηση, και με ίδιο τρόπο όπως προηγουμένως γίνεται το predict και η εκτύπωση των accuracy, precision κ.λπ.

Στη συνέχεια ακολουθεί ο classifier του naïve bayes. Με τη συνάρτηση GaussianNB γίνεται η συγκεκριμένη κατηγοριοποίηση, και με ίδιο τρόπο όπως προηγουμένως γίνεται το predict και η εκτύπωση των accuracy, precision κ.λπ.

## 3 ΠΕΙΡΑΜΑΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ

Τα accuracy, precision, recall και f1 score υπολογίστηκαν αυτόματα από έτοιμες συναρτήσεις, και το μόνο που άλλαξε είναι η παράμετρος y\_pred καθώς το y\_test παρέμεινε σταθερό. Δηλαδή το y\_pred είναι το σύνολο αποτελεσμάτων που προέβλεπε ο κάθε αλγόριθμος, και στη συνέχεια αυτό συγκρινόταν με τα πραγματικά δεδομένα (y\_test) για να δούμε αν υπέρχει μεγάλη απόκλιση. Όσο οι αριθμοί τείνουν προς το 1, τότε η απόκλιση θα είναι μικρότερη. [πηγή]

Ανά κατηγορία έχουμε τα εξής:

### 3.1 ΔΕΝΤΡΟ ΑΠΟΦΑΣΕΩΝ

Γι' αυτόν τον αλγόριθμο έχουμε τα αποτελέσματα:

```
Decision Tree:
Accuracy: 0.77
Precision: 0.77
Recall: 0.77
F1: 0.77
```

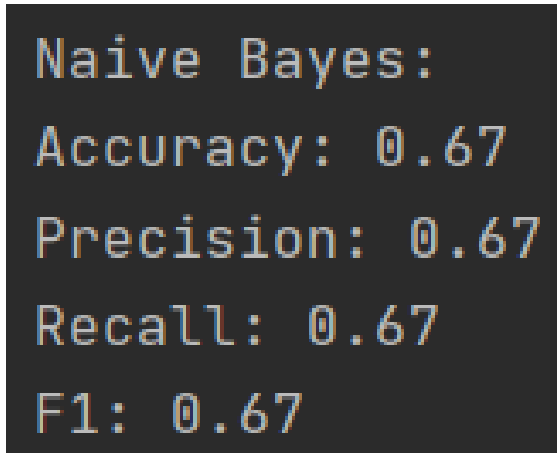
### 3.2 K-KONTINOTEROI ΓΕΙΤΟΝΕΣ

Γι' αυτόν τον αλγόριθμο έχουμε τα αποτελέσματα:

```
K-Nearest Neighbours:
Accuracy: 0.91
Precision: 0.91
Recall: 0.91
F1: 0.91
```

### 3.3 NAÏVE BAYES

Γι' αυτόν τον αλγόριθμο έχουμε τα αποτελέσματα:



```
Naive Bayes:
Accuracy: 0.67
Precision: 0.67
Recall: 0.67
F1: 0.67
```

## 4 ΣΥΜΠΕΡΑΣΜΑΤΑ

Συμπεραίνουμε ότι ο καλύτερος αλγόριθμος βάσει αποτελεσμάτων είναι ο classifier αυτός του K-Neighbors, με το μεγαλύτερο από τα υπόλοιπα accuracy: 0.91.

## ΑΝΑΦΟΡΕΣ ΚΑΙ ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΠΗΓΕΣ

- [1] <https://stackabuse.com/text-classification-with-python-and-scikit-learn/>
- [2] <https://www.datacamp.com/community/tutorials/decision-tree-classification-python>
- [3] <https://analyticsindiamag.com/7-types-classification-algorithms/>
- [4] <https://www.geeksforgeeks.org/ml-one-hot-encoding-of-datasets-in-python/>
- [5] [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html)
- [6] [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html)
- [7] [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html)
- [8] [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html)