Random mushroom or random forest ?

Σύγκριση αλγορίθμων μηχανικής μάθησης για πρόβλεψη εδωδιμότητας άγριων μανιταριών

Βασισμένη στο πακέτο δεδομένων “Mushroom” του UC Irvine Machine Learning Repository

UC Irvine Machine Learning Repository

https://archive.ics.uci.edu/dataset/73/mushroom

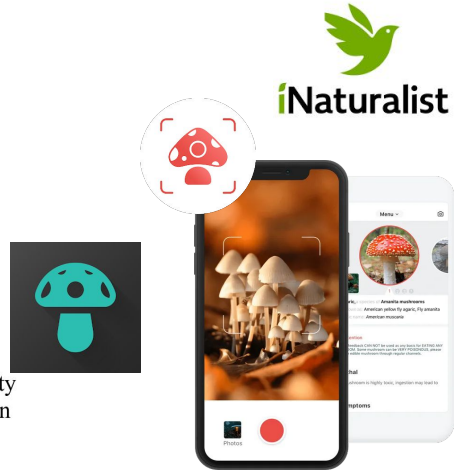
Θοδωρής Γιωννάς-Μασσέλος
ΠΜΣ “Βιοπληροφορική”
Ιατρική Σχολή
Πανεπιστήμιο Κρήτης
12/11/2025

Γιατί να ταυτοποιήσουμε τα μανιτάρια με ML ?

Επιστήμη των πολιτών

Μανιταροσυλλέκτες

Πρόληψη δηλητηριάσεων



Fatal Fungi: Enhancing Mushroom Safety through Machine Learning Identification

Yasmine D. Subbagh

› Clin Toxicol (Phila). 2023 Mar;61(3):166-172. doi: 10.1080/15563650.2022.2162917. Epub 2023 Feb 15.

A comparison of the accuracy of mushroom identification applications using digital photographs

Sarah E Hodgson [1,2], Christine McKenzie [1], Tom W May [3], Shaun L Greene [1,2]

Conclusions: Mushroom identification applications may be useful future tools to assist clinical toxicologists and the general public in the accurate identification of mushrooms species but, at present, are not reliable enough to exclude exposure to potentially poisonous mushrooms when used alone.

› Mycologia. 2018 Jul-Aug;110(4):637-641. doi: 10.1080/00275514.2018.1479561. Epub 2018 Jul 31.

Mushroom poisoning epidemiology in the United States

William E Brandenburg [1,2], Karlee J Ward [3]

US mushroom exposures as reported by the NPDS from 1999 to 2016. Over the last 18 years, 133 700 cases (7428/year) of mushroom exposure, mostly by ingestion, have been reported. Cases are most

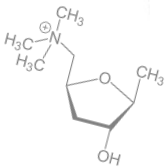
ingestions resulted in no or minor harm, although some groups of mushroom toxins or irritants, such as cyclopepides, ibotenic acid, and monomethylhydrazine, have been deadly. Misidentification of edible mushroom species appears to be the most common cause and may be preventable through education.

Το σετ δεδομένων

8124
Παρατηρήσεις

Μανιτάρια με ελάσματα:

Lepiota & *Agaricus*
Υποθετικά δείγματα



23
Μεταβλητές

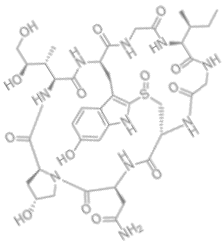
- Μορφολογία
- Ενδιαίτημα
- Χρώμα

Κατηγορικές ή Δυαδικές

- cap : shape, surface, color
- bruises
- odor
- gills : attachment, spacing, size, color
- stalk : surface, color (above/below ring)
- veil
- ring
- spore print
- population
- habitat

Μία δυαδική
μεταβλητή
στόχος

- Εδωδιμότητα

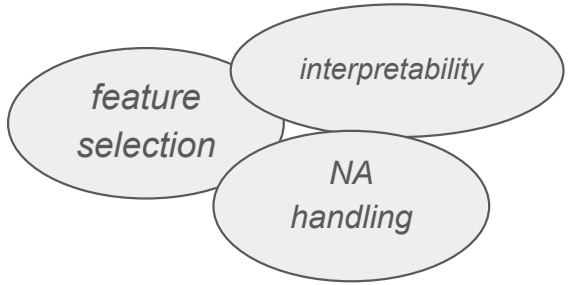


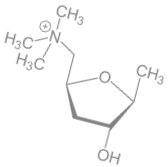
Ένα πρόβλημα ταξινόμησης

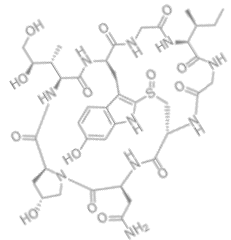
Στόχος: εφαρμογή μεθόδων ML για ταξινόμηση, σύγκριση της αποδοτικότητας τους & αξιολόγηση σημαντικότητας χαρακτηριστικών

Random forest

K-Nearest Neighbors



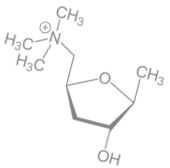


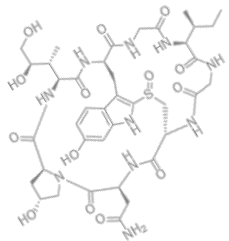


Βήματα ανάλυσης

ML implementation in Scikit-learn

- Variable encoding
- NA handling (remove for kNN, let be for Random Forest)
- Data split → test & train
- ML implementation
- Feature importance & selection
- Score metrics calculation





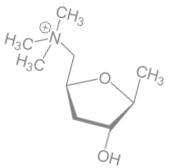
Αλγόριθμοι ML

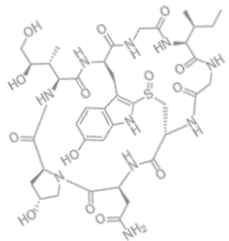
Random Forest:

- Hyperparameter optimization (Grid Search and Cross-validation)
- Feature selection : Mean Decrease Impurity, Permutation Feature Importance

k Nearest Neighbors:

- Classic kNN
- kNN with Neighborhood Component Analysis

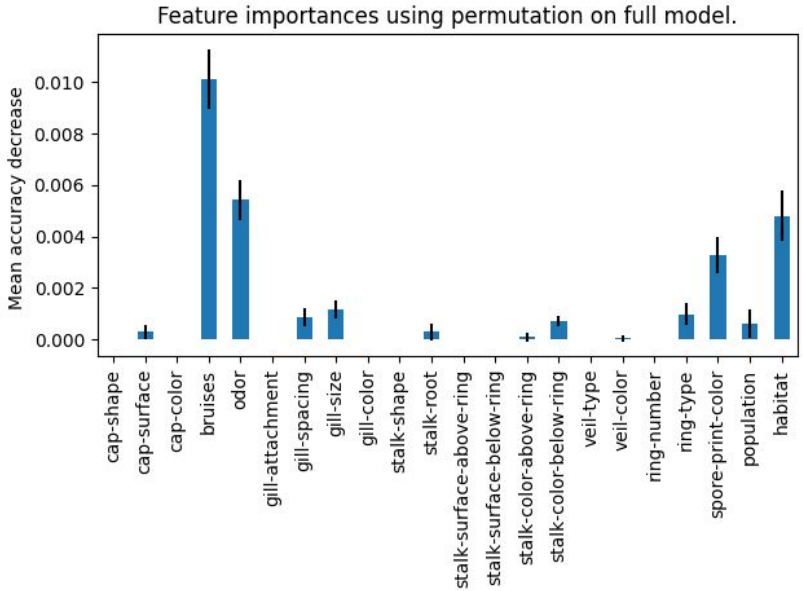


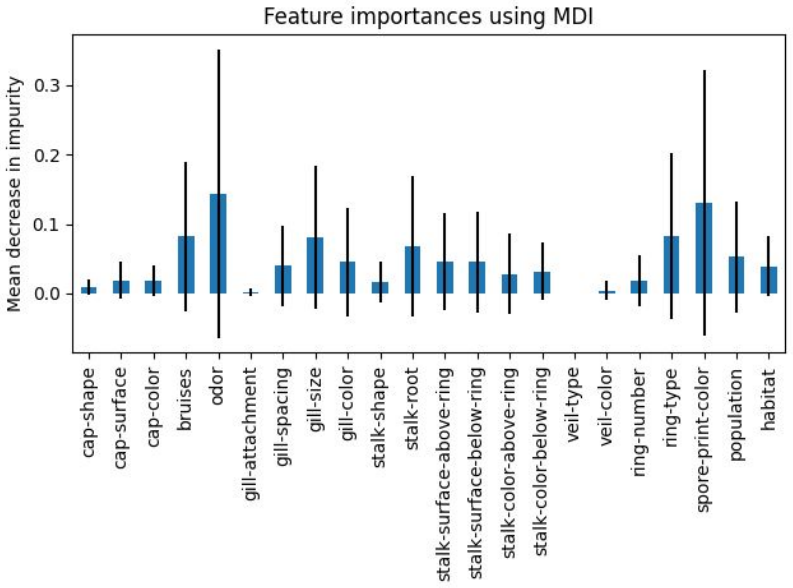


Αποτελέσματα: *Random forest*

Accuracy :	1.0
Precision :	1.0
Recall:	1.0

Feature Selection:

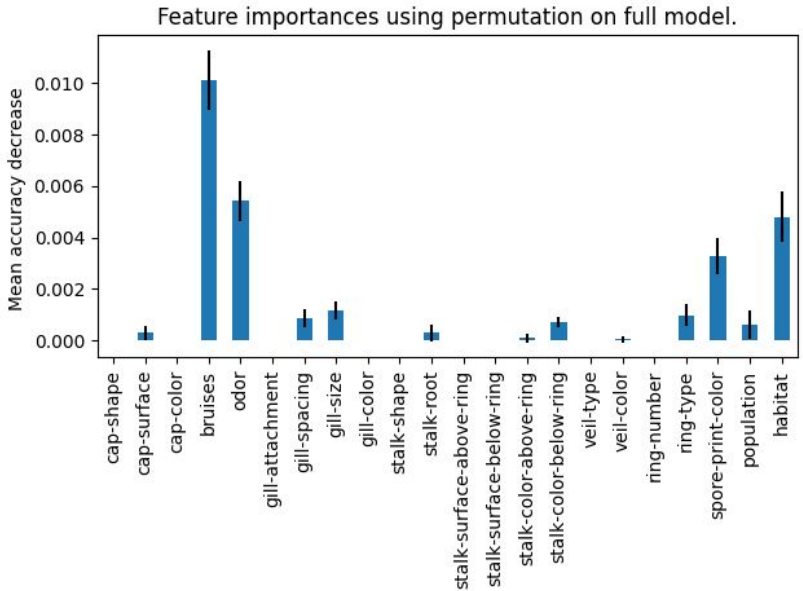


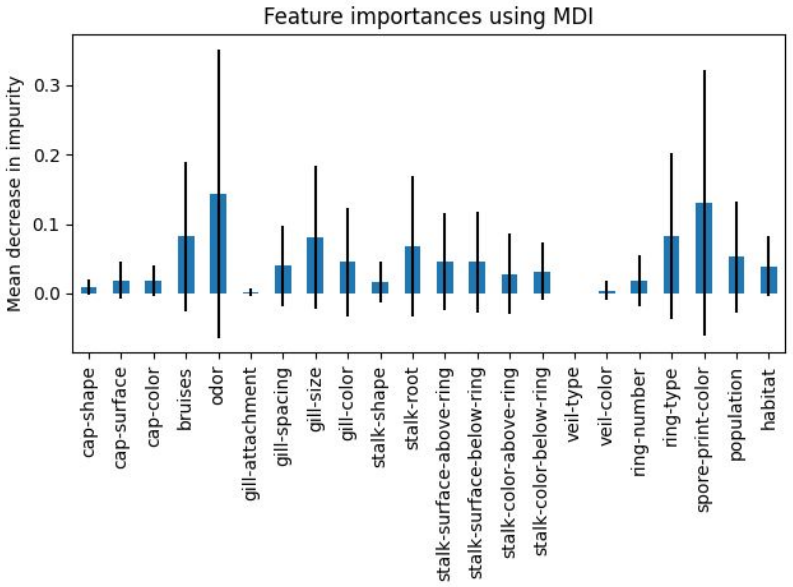


Αποτελέσματα: *Random forest feature selection*

['odor', 'spore-print-color', 'gill-size', 'habitat', 'bruises']

['cap-shape', 'cap-color', 'cap-surface', 'odor', 'bruises']





Αποτελέσματα: *Random forest*

['odor', 'spore-print-color', 'gill-size', 'habitat', 'bruises']

vars. from permutation model

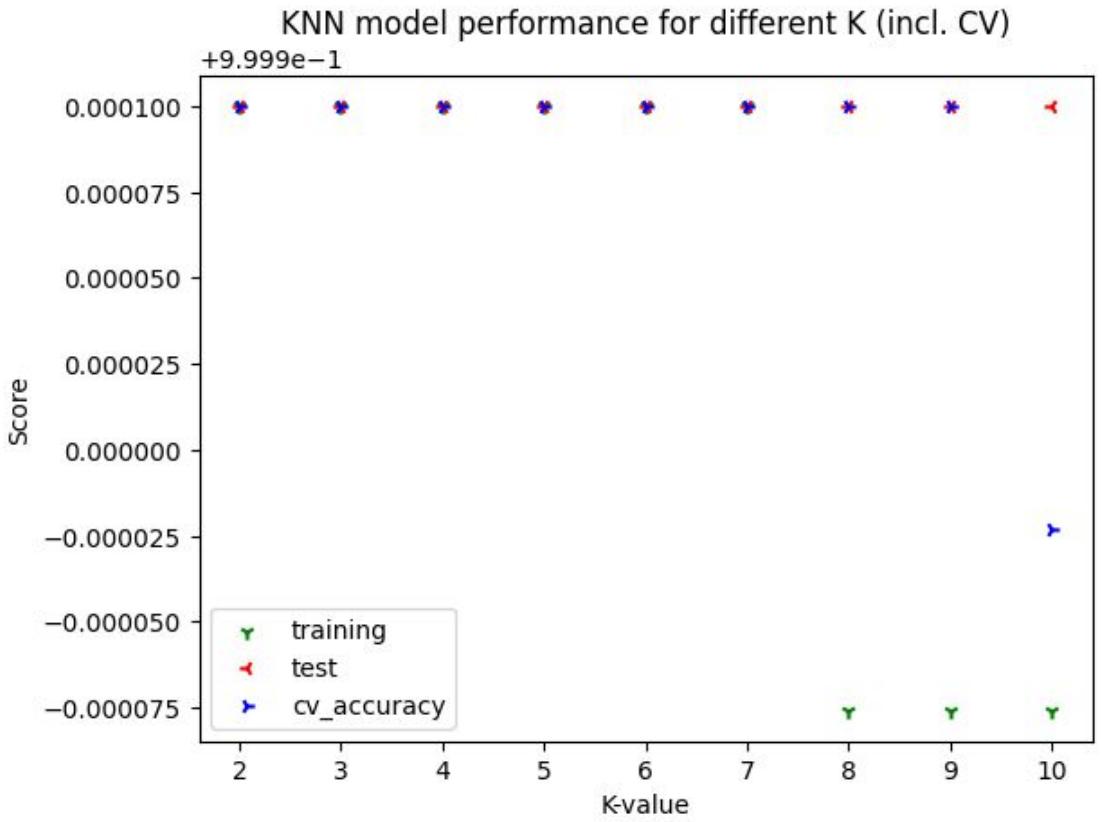
Precision : 1.0
Accuracy : 0.999
Recall: 0.988

['cap-shape', 'cap-color', 'cap-surface', 'odor', 'bruises']

vars from MDI

Precision : 1.0
Accuracy : 0.994
Recall: 0.988

Αποτελέσματα: *k Nearest Neighbors*



Precision for train, test datasets, accuracy for the cross-validated train set

Accuracy : 1.0
Precision : 1.0
Recall: 1.0

…για όλα K=2 έως K=7

KNN model with CV

	training	test	CV_mean_accuracy	Precision	Recall
K=2:	100.00%	100.00%	100.00%	100.00%t	100.00%
K=3:	100.00%	100.00%	100.00%	100.00%t	100.00%
K=4:	100.00%	100.00%	100.00%	100.00%t	100.00%
K=5:	100.00%	100.00%	100.00%	100.00%t	100.00%
K=6:	100.00%	100.00%	100.00%	100.00%t	100.00%
K=7:	100.00%	100.00%	100.00%	100.00%t	100.00%
K=8:	99.98%	100.00%	100.00%	100.00%t	100.00%
K=9:	99.98%	100.00%	100.00%	100.00%t	100.00%
K=10:	99.98%	100.00%	99.99%	100.00%t	100.00%

KNN model using NCA with CV

	CV accuracy(train)	Accuracy score (test)	Precision	Recall
K=2:	100.000%	100.000%	100.000%	100.000%
K=3:	100.000%	100.000%	100.000%	100.000%
K=4:	99.965%	100.000%	100.000%	100.000%
K=5:	99.965%	100.000%	100.000%	100.000%
K=6:	99.930%	100.000%	100.000%	100.000%
K=7:	99.965%	100.000%	100.000%	100.000%
K=8:	99.930%	99.877%	100.000%	99.747%
K=9:	99.965%	99.877%	100.000%	99.747%
K=10:	99.965%	99.877%	100.000%	99.747%

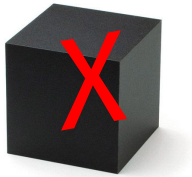
Συζήτηση: Σύγκριση μοντέλων

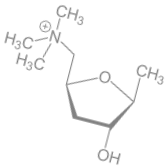
Πολύ υψηλή ακρίβεια και από τα δύο μοντέλα!
Overfitting στο συγκεκριμένο dataset ;

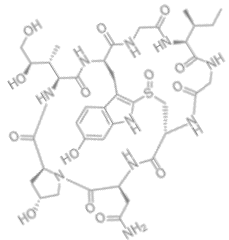
Το kNN ακόμα και με K=2, K=3 έχει υψηλή απόδοση!
→ Δεδομένα χαμηλή πολυπλοκότητας
→ Δεν επιτρέπει επιλογή σημαντικών μεταβλητών
→ Δεν μπορεί να χειριστεί εισαγωγές με τιμές σε έλλειψη

Random forest:
→Πολύ υψηλοί δείκτες, απόλυτη ταξινόμηση
→Δυνατότητα διάκρισης μεταβλητών

Πρακτική εφαρμογή ;







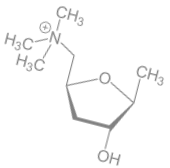
Συζήτηση

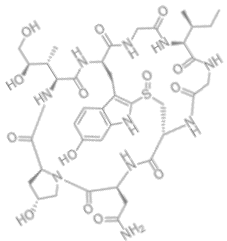
Τα μοντέλα αρχικά φαίνεται να έχει υψηλή ακρίβεια! …είναι όμως επιτυχημένο;

Η επιτυχία είναι μόνο για ένα dataset!
Πλεονάζουσες μεταβλητές → άλλες οικογένειες μπορείς να χρειάζονται περισσότερα features. “Σχετικό overfitting;”

Εγγενής ενδοειδική ποικιλομορφία δεν επιτρέπει πάντα την διάκριση των χαρακτήρων που απαιτεί το μοντέλο, για τον ίδιο λόγο ούτε η φωτογράφιση μπορεί να δώσει πάντα όλες τις απαραίτητες πληροφορίες.

Η δημιουργία μεθόδων που να μπορούν να ταυτοποιούν με ακρίβεια πολλά είδη μανιταριών παραμένει ως πρόκληση.





Μέθοδοι συμπληρωματικό υλικό

Key Metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

```
start_time = time.time()
parameters_to_test = {'n_estimators' : np.arange(50, 70, 100),
                      'max_depth' : [3, 5, 7, 9],
                      'max_features' : np.arange(0.1, 1),
                      'max_samples' : [0.5, 0.7]}

clf_model = GSCV(RandomForestClassifier(), parameters_to_test,
cv = 5, scoring = 'accuracy', n_jobs = -1)
clf_model.fit(Xc_train, yc_train)
```