
29. ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ PYTHON – ΕΓΚΑΤΑΣΤΑΣΗ, ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ – ΜΕΤΑΒΛΗΤΕΣ, ΤΕΛΕΣΤΕΣ, ΤΥΠΟΙ, IDLE – ΜΑΘΗΜΑΤΑ 1-3

29.0.1 Λύσεις προηγούμενων ασκήσεων

Άσκηση 1: Data Splitting

Διαχωρίστε το dataset των κρασιών σε δύο training και testing sets χρησιμοποιώντας μια ποσοστοποίηση 70% - 30%.

Λύση:

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Φόρτωση του dataset
df = pd.read_csv("winequality-white.csv", sep=";")

# Προετοιμασία χαρακτηριστικών και δεδομένων
df_features = df.drop(columns='quality')
df_label = df['quality']
```

```
# Διαχωρισμός των δεδομένων (70% train, 30% test)
X_train, X_test, y_train, y_test =
train_test_split(df_features, df_label, test_size=0.30)

# Εκτύπωση των διαστάσεων των sets
print("Μέγεθος training set:", len(X_train))
print("Μέγεθος testing set:", len(X_test))
```

```
Μέγεθος training set: 3428
Μέγεθος testing set: 1470
```

Άσκηση 2: Εκπαίδευση μοντέλου

Εκπαιδεύστε ένα μοντέλο Logistic Regression επάνω στο wine dataset χρησιμοποιώντας τα training data και μετά, τυπώστε την ακρίβεια του μοντέλου επάνω στα testing data.

Λύση:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Φόρτωση του dataset
df = pd.read_csv("winequality-white.csv", sep=";")

# Προετοιμασία χαρακτηριστικών κι ετικετών
df_features = df.drop(columns='quality')
df_label = df['quality']

# Διαχωρισμός σε training και testing sets
X_train, X_test, y_train, y_test =
train_test_split(df_features, df_label, test_size=0.20)

# Δημιουργία κι εκπαίδευση Logistic Regression
# Εδώ ορίζουμε τις επαναλήψεις.
# Αυξάνοντας διαδοχικά σε 1000, 2000, 5000 και 10000,
σταματάμε να λαμβάνουμε warnings
```

```
logisticRegression = LogisticRegression(max_iter=10000)
logisticRegression.fit(X_train, y_train)

# Υπολογισμός κι εκτύπωση της ακρίβειας του μοντέλου
στο testing set
accuracy = logisticRegression.score(X_test, y_test)
print("Model Accuracy:", accuracy)
```

```
= RESTART: C:/Users/NK/AppData/Local/
Model Accuracy: 0.5387755102040817
```

Άσκηση 3: Οπτικοποίηση δένδρου αποφάσεων

Εκπαιδεύστε έναν ταξινομητή δένδρου αποφάσεων επάνω στο wine dataset χρησιμοποιώντας τα παραπάνω training data, και μετά οπτικοποιήστε το δένδρο αποφάσεων.

Λύση:

```
pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree

# Φόρτωση του dataset
df = pd.read_csv("winequality-white.csv", sep=";")

# Προετοιμασία χαρακτηριστικών κι ετικετών
df_features = df.drop(columns='quality')
df_label = df['quality']

# Διαχωρισμός δεδομένων σε training και testing sets
X_train, X_test, y_train, y_test =
train_test_split(df_features, df_label, test_size=0.20)

# Δημιουργία κι εκπαίδευση ταξινομητή Δένδρου αποφάσεων
```

```

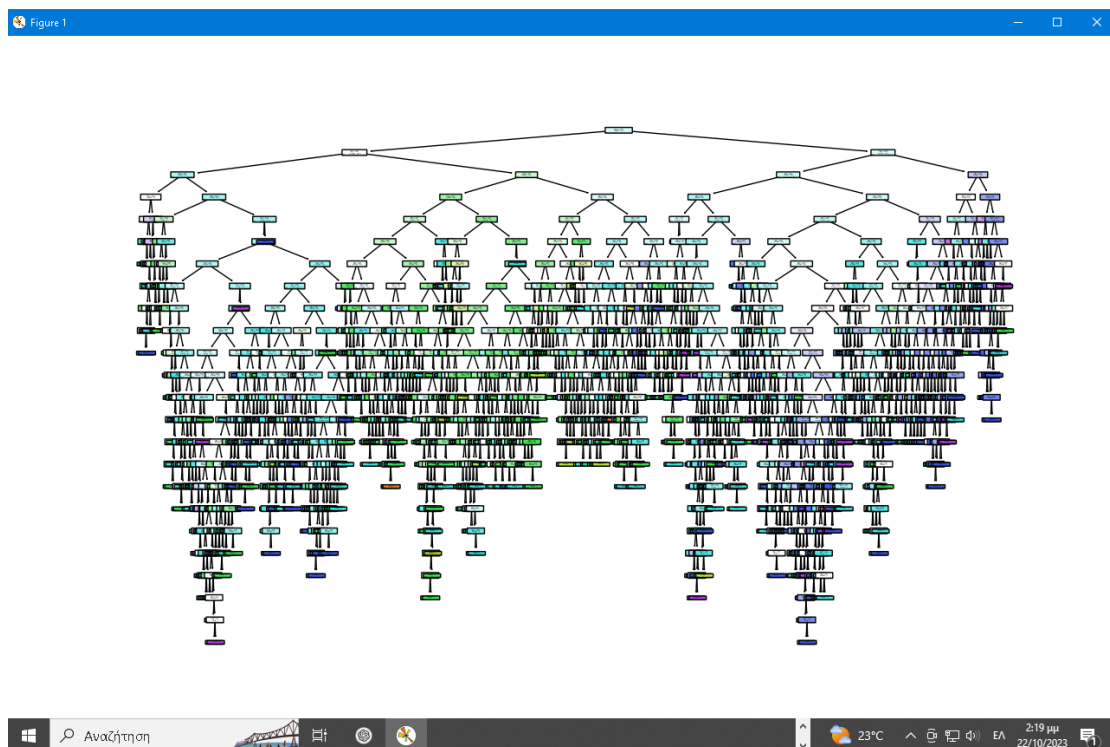
decisionTree = DecisionTreeClassifier()
decisionTree.fit(X_train, y_train)

# Δημιουργία μεγέθους του γραφήματος
plt.figure(figsize=(16, 8))

# Δημιουργία συνάρτησης (plot_tree()) για οπτικοποίηση
του δένδρου αποφάσεων.
plot_tree(decisionTree,
          filled=True,
          rounded=True)

# Εμφάνιση του γραφήματος
plt.show()

```



29.14.1 Unpacking

Στην Python, ο όρος "unpacking" αναφέρεται συνήθως στη διαδικασία εξαγωγής μεμονωμένων στοιχείων από ένα διαπερνόμενο αντικείμενο

(iterable), όπως μια λίστα ή μια πλειάδα, και την ανάθεσή τους σε ξεχωριστές μεταβλητές. Μας επιτρέπει να έχουμε εύκολη πρόσβαση στα στοιχεία μέσα σε μια δομή δεδομένων και να τα χρησιμοποιούμε ως μεμονωμένες τιμές.

Το unpacking μπορεί να γίνει με μερικούς διαφορετικούς τρόπους, ανάλογα με τη δομή του iterable αντικειμένου και τον τρόπο με τον οποίο θέλουμε να εκχωρήσουμε τις τιμές σε μεταβλητές:

1. Tuple Unpacking

Μπορούμε να χρησιμοποιήσουμε το tuple unpacking για να αναθέσουμε τιμές από ένα tuple σε ξεχωριστές μεταβλητές.

Για παράδειγμα:

```
my_tuple = (1, 2, 3)
a, b, c = my_tuple
print(a) # Output: 1
print(b) # Output: 2
print(c) # Output: 3
```

2. List Unpacking:

Το ίδιο όπως παραπάνω, αλλά με λίστες:

```
my_list = [4, 5, 6]
x, y, z = my_list
print(x) # Output: 4
print(y) # Output: 5
print(z) # Output: 6
```

3. Πολλαπλές αναθέσεις:

Μπορούμε να χρησιμοποιήσουμε το unpacking για να αναθέσουμε τιμές σε πολλαπλές μεταβλητές μονομιάς.

Για παράδειγμα:

```
a, b = 1, 2
print(a) # Output: 1
print(b) # Output: 2
```

4. Εκτεταμένο Unpacking:

Η Python επιτρέπει να χρησιμοποιήσουμε τον αστερίσκο (*) για να συλλέξουμε πολλαπλές τιμές όταν εφαρμόζουμε το unpacking.

Για παράδειγμα:

```
first, *rest = [1, 2, 3, 4, 5]
print(first) # Output: 1
print(rest) # Output: [2, 3, 4,
```

Το ξεπακετάρισμα (unpacking) είναι ένας βολικός τρόπος για να εργαστούμε με δεδομένα που είναι αποθηκευμένα σε iterables και χρησιμοποιείται συχνά στην Python, όταν χρειάζεται να χειριστούμε πολλαπλές τιμές ή στοιχεία.

29.15.1 Τελεστές Floor Division + Modulus]

Floor Division - Ακέραιη Διαίρεση (//):

Βασική χρήση Floor Division:

```
result = 10 // 3
print(result) # έξοδος: 3
```

Χρήση Floor Division με αρνητικούς αριθμούς:

```
result = -10 // 3  
print(result) # έξοδος : -4
```

Floor Division σε Loop:

Η ακέραιη διαίρεση μπορεί να μας φανεί χρήσιμη για τη διαπέραση ενός εύρους τιμών:

```
for i in range(1, 11):  
    quotient = i // 3  
    print(f"Quotient of {i} // 3 = {quotient}")
```

```
Quotient of 1 // 3 = 0  
Quotient of 2 // 3 = 0  
Quotient of 3 // 3 = 1  
Quotient of 4 // 3 = 1  
Quotient of 5 // 3 = 1  
Quotient of 6 // 3 = 2  
Quotient of 7 // 3 = 2  
Quotient of 8 // 3 = 2  
Quotient of 9 // 3 = 3  
Quotient of 10 // 3 = 3  
|
```

Modulus – Υπόλοιπο ακέραιας διαίρεσης (%)

Βασική χρήση:

```
remainder = 10 % 3  
print(remainder) # έξοδος: 1
```

Έλεγχος περιττού ή άρτιου:

```

number = 7
if number % 2 == 0:
    print(f"Ο {number} είναι άρτιος.")
else:
    print(f"Ο {number} είναι περιττός.") # Ο 7 είναι
    περιττός

```

Χρήση του Modulus σε Loop:

Χρήση του modulus σε loop για φιλτράρισμα άρτιων αριθμών.

```

for i in range(1, 11):
    if i % 2 == 0:
        print(f"Ο {i} είναι άρτιος.")
    else:
        print(f"Ο {i} είναι περιττός.")

```

```

Ο 1 είναι περιττός.
Ο 2 είναι άρτιος.
Ο 3 είναι περιττός.
Ο 4 είναι άρτιος.
Ο 5 είναι περιττός.
Ο 6 είναι άρτιος.
Ο 7 είναι περιττός.
Ο 8 είναι άρτιος.
Ο 9 είναι περιττός.
Ο 10 είναι άρτιος.

```

29.16.0 Πλήρης πίνακας προτεραιότητας τελεστών στην Python (για αναφορά)

Προτεραιότητα	Τελεστές	Περιγραφή
1	()	Παρενθέσεις (ομαδοποίηση)
2	f(args...)	Κλήση συνάρτησης

	x[index]	Συνδρομή (δείκτης)	
	x.στοιχείο	Πρόσβαση σε μέλος (μέλος)	
	x(args...)	Κλήση συνάρτησης (ορίσματα)	
	f"{x}"	Μορφοποίηση συμβολοσειράς	
	await x	Αναμονή ασύγχρονης λειτουργίας	
3	**	Ύψωση σε δύναμη	
4	+x, -x	Μοναδική θετική και αρνητική	
	~x	Δυαδική ΕΠΙΚΟΙΝΩΝΙΑ	
	not x	Λογικό ΌΧΙ	
5	*, /, %, //	Πολλαπλασιασμός, διαίρεση, υπόλοιπο,	
	ακέραιο διαίρεση		
6	+, -	Πρόσθεση και αφαίρεση	
7	<<, >>	Δυαδική αριστερή μετατόπιση και δυαδική	
	δεξιά μετατόπιση		
8	&	Δυαδικό ΚΑΙ	
9	^	Δυαδικό XOR	
10		Δυαδικό Ή	
11	in, not in	Έλεγχοι συμπερίληψης (περιλαμβάνεται,	
	δεν περιλαμβάνεται)		
	is, is not	Έλεγχοι ταυτότητας	
	<, <=, >, >=	Σύγκριση (μικρότερο από, μικρότερο από ή	
	ίσο, μεγαλύτερο από, μεγαλύτερο από ή ίσο)		
	==, !=	Ισότητα και ανισότητα	
12	not	Λογική ΌΧΙ	
13	and	Λογική ΚΑΙ	
14	or	Λογική Ή	
15	if else	Συνθετική έκφραση (τριαδικός τελεστής)	

29.21.0 Ασκήσεις

1. Γράψτε το παρακάτω ποίημα ώστε σαν έξοδο να έχετε την παρακάτω εκτύπωση:

```
Όταν ήμουν μικρό παιδί, όπως είναι  
τ' αρνάκια πού παίζουνε  
στό λειβάδι τήν άνοιξη,  
είχα φίλους τίσ πασχαλίτσες  
είχα φίλους τίσ λιμπελούλες  
είχα φίλους τά λουλουδάκια.  
Νικηφόρος Βρεττάκος
```

2. Γράψτε ένα πρόγραμμα που να σας δείχνει ποια έκδοση της Python έχετε (όχι στο command line).
3. Γράψτε ένα πρόγραμμα που να σας δείχνει την ημερομηνία και την ώρα αυτή τη στιγμή.
4. Γράψτε ένα πρόγραμμα που θα δέχεται έναν αριθμό δευτερολέπτων και θα υπολογίζει πόσα λεπτά και δευτερόλεπτα αντιστοιχούν.
5. Γράψτε ένα πρόγραμμα που θα ελέγχει αν ένας αριθμός είναι ζυγός ή περιττός χρησιμοποιώντας τον τελεστή %.
6. Γράψτε ένα πρόγραμμα που θα δέχεται δύο αριθμούς από το χρήστη και θα υπολογίζει το υπόλοιπο της διαίρεσης του πρώτου αριθμού από τον δεύτερο χρησιμοποιώντας τον τελεστή %.

7. Γράψτε ένα προγραμματάκι το οποίο τυπώνει το ημερολόγιο ενός μήνα κάποιου έτους. Ζητάει και τα δύο στοιχεία από τον χρήστη (πρώτα το έτος). Εισάγετε τη βιβλιοθήκη calendar.
8. Γράψτε ένα πρόγραμμα το οποίο να προσθέτει τα στοιχεία μιας λίστας το ένα μετά το άλλο, ακόμα κι αν είναι αριθμητικά. Δώστε 2 παραδείγματα, ένα με αριθμούς κι ένα με συμβολοσειρές. Η εκτύπωση πρέπει να είναι:

```
= RESTART: C:/Users  
15122  
H Python σκίζει  
|
```