Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Τομέας Ηλεκτρονικής και Υπολογιστών

9ο Εξάμηνο
Φθινόπωρο 2021

Python Tutorial

1 Εισαγωγή στην Python

Η γλώσσα προγραμματισμού Python αποτελεί open source λογισμικό το οποίο είναι διαθέσιμο στην ιστοσελίδα *http://www.python.org/*. Αν και μπορεί να χρησιμοποιηθεί κανονικά απευθείας μετά την εγκατάσταση της, ακόμα και από τερματικό, υπάρχουν διαθέσιμα open source ολοκληρωμένα περιβάλλοντα ανάπτυξης (IDEs). Ένα από τα πιο γνωστά IDEs, το οποίο και θα χρησιμοποιήσουμε, είναι το Visual Studio Code (VS Code), το οποίο είναι διαθέσιμο στην ιστοσελίδα *http://code.visualstudio.com/* και παρέχει κάποιες διευκολύνσεις.

Οι συναρτήσεις στην Python είναι διαμοιρασμένες σε βιβλιοθήκες (libraries), κάποιες εκ των οποίων (οι βασικότερες) φορτώνονται αυτόματα κατά την έναρξη της εκτέλεσης του προγράμματος. Βασικές συναρτήσεις που φορτώνονται αυτόματα κατά την εκτέλεση του προγράμματος αφορούν τη διαχείριση Boolean μεταβλητών και τύπων, τη μετατροπή μεταξύ μεταβλητών διαφορετικού τύπου, τη δημιουργία βασικών δομών δεδομένων κ.ο.κ. Επίσης, πολλές βιβλιοθήκες που περιέχουν βασικές συναρτήσεις, όπως για παράδειγμα η βιβλιοθήκη "`numpy`", κατεβαίνουν αυτόματα κατά την εγκατάσταση της γλώσσας και για τη χρησιμοποίησή τους, ο χρήστης χρειάζεται απλά να τις εισάγει στον κώδικά του με την εντολή "`import numpy as np`". Ωστόσο, προκειμένου να χρησιμοποιηθούν άλλες χρήσιμες συναρτήσεις θα πρέπει ο χρήστης να κατεβάσει και να εγκαταστήσει επιπλέον βιβλιοθήκες. Αυτό πραγματοποιείται εύκολα, εκτελώντας σε ένα τερματικό την εντολή "`pip install library_name`".

Η Python περιλαμβάνει ένα πλήθος συναρτήσεων που μπορούν να μας βοηθήσουν να ελέγξουμε το path στο οποίο βρισκόμαστε, να αλλάξουμε path και να διαβάσουμε ένα αρχείο από οπουδήποτε στον υπολογιστή μας. Η βιβλιοθήκη που περιέχει αυτές τις συναρτήσεις είναι η `os`. Με την εντολή `os.getcwd()` μπορούμε να ελέγξουμε το path στο οποίο βρισκόμαστε, ενώ με την εντολή `os.chdir("C:\\...")` μπορούμε να αλλάξουμε αυτό το path. Με την εντολή `os.listdir("C:\\...")` μπορούμε να δούμε όλα τα αντικείμενα που βρίσκονται στο συγκεκριμένο path, ενώ με την εντολή `os.path.join("subpath1", "subpath2", …)` μπορούμε να δημιουργήσουμε ένα σύνθετο path, που αποτελείται από επιμέρους subpaths.

2 Βασικές δομές με παραδείγματα

2.1 Λίστες

Να δημιουργηθεί η λίστα A = [10, 5, 3, 100, -2, 5, -50] και να τυπωθεί στην οθόνη.

```
>>> A = [10, 5, 3, 100, -2, 5, -50]
>>> print(A)
>>> print(A[1:3])
```

Επιλογή των στοιχείων 1, 3, 4 και 5 της λίστας A.

```
>>> B = []
>>> for index in [1, 3, 4, 5]:
>>> B.append(A[index])
```

Ή

```
>>> B = [A[index] for index in [1, 3, 4, 5]]
```

Ποια στοιχεία του A έχουν τιμή μεγαλύτερη του 5;

```
>>> B = [el for el in A if el > 5]
>>> indices = [i for i, el in enumerate(A) if el > 5] # Επιστρέφει τις θέσεις των στοιχείων του A για τα οποία ισχύει η ανισότητα
```

Για να δούμε πόσα είναι αυτά αρκεί να χρησιμοποιήσουμε τη `len`.

```
>>> print(len(B))
```

Δημιουργία ενός δεύτερου διανύσματος B και ένωσή του με το διάνυσμα A.

```
>>> A = [10, 5, 3, 100, -2, 5, -50]
>>> B = [1, 2, 5, 6, 9, 0, 100]
>>> print(A + B)
>>> print(A.extend(B))
>>> print(A)
```

2.3 Data frames

Αποτελεί μία δομή συχνά χρησιμοποιούμενη στην Python, για την αποθήκευση πινάκων δεδομένων. Αποτελεί γενίκευση των πινάκων και μπορεί να συνδυάζει διαφορετικές μεταξύ τους δομές (π.χ. διανύσματα με χαρακτήρες, πίνακες με αριθμητικά κλπ.). Για την διαχείρισή τους χρησιμοποιείται η βιβλιοθήκη `pandas`.

Παράδειγμα δημιουργίας ενός data frame και προσπέλαση των στοιχείων του.

```
>>> n = [2, 3, 5]
>>> s = ["aa", "bb", "cc"]
>>> b = [True, False, True]
# Δημιουργία του data frame df και προβολή του στην οθόνη.
>>> df = pd.DataFrame(list(zip(n, s, b)), columns=['n', 's', 'b'])
>>> print(df)
# Προβολή της στήλης n του data frame με χρήση του χαρακτήρα '.'
>>> print(df.n)
```

2.4 Προσπέλαση Δεδομένων

Η βιβλιοθήκη `pandas` προσφέρει ένα πλήθος συναρτήσεων που μπορούν να μας βοηθήσουν να προσπελάσουμε ένα `dataframe`, επιλέγοντας είτε γραμμές είτε στήλες, ακόμα και φιλτράροντας τα δεδομένα.

Για να προσπελάσουμε τις στήλες ή τις γραμμές ενός dataframe, μπορούμε να χρησιμοποιήσουμε τη συνάρτηση loc, δίνοντας σαν όρισμα το σύνολο των γραμμών ή στηλών που θέλουμε να κρατήσουμε. Για παράδειγμα, στο προηγούμενο dataframe, μπορούμε να προσπελάσουμε τη δεύτερη γραμμή με την εντολή:
```
>>> print(df.loc[1])
```

και την τιμή της στήλης “b” στην δεύτερη γραμμή με την εντολή:

```
>>> print(df.loc[1, “b”])
```

Για να προσπελάσουμε τη στήλη “b” από όλες τις γραμμές, χρησιμοποιούμε την εντολή:

```
>>> print(df.loc[:, “b”])
```

Για να πάρουμε τις γραμμές στις οποίες ισχύει μια συγκεκριμένη συνθήκη, χρησιμοποιούμε την εντολή:

```
>>> print(df.loc[(df.b == True)])
```

3 Programming Tools

3.1 File I/O

Για να διαβάσουμε τα δεδομένα ενός αρχείου από το δίσκο μπορούμε να χρησιμοποιήσουμε τη συνάρτηση `read_csv` της βιβλιοθήκης `pandas`. Έστω ότι έχουμε το αρχείο `people.txt`, στον ίδιο φάκελο με το `.py` αρχείο μας, με δεδομένα:

```
Age, Height, Weight
25, 1.95, 85
18, 1.82, 80
44, 1.75, 82
26, 1.78, 75
34, 1.77, 78
```

Μπορούμε να το διαβάσουμε με την εντολή:

```
>>> people = pd.read_csv("people.txt")
```

Η συνάρτηση μας επιτρέπει να επιλέξουμε το χαρακτήρα οριοθέτησης με την παράμετρο `sep` (το default είναι το κόμμα) ενώ η παράμετρος `header` ελέγχει αν η πρώτη γραμμή είναι το header. Επίσης μπορούμε να θέσουμε ονόματα στηλών με την παράμετρο `names`.

Για τα παρακάτω δεδομένα:

```
25; 1.95; 85
18; 1.82; 80
44; 1.75; 82
26; 1.78; 75
34; 1.77; 78
```

εκτελούμε την εντολή:

```
>>> people = pd.read_csv("people.txt", sep=";", header=None,
names=["Age", "Height", "Weight"])
```

Σε περίπτωση που λείπουν κάποια δεδομένα (έχουμε missing values), μπορούμε να τα αντικαταστήσουμε με τη διάμεσο. Π.χ. για το παρακάτω dataset

```
25; 1.95; 85
18; 1.82; 80
44; 1.75;
26; 1.78; 75
34; 1.77; 78
```

αφού φορτώσουμε τα δεδομένα, μπορούμε να εκτελέσουμε τις παρακάτω εντολές:

```
>>> people = people.fillna(people.mean())
```

Μπορούμε, επίσης, να πάρουμε κάποιες στατιστικές πληροφορίες για τα δεδομένα, με τις παρακάτω εντολές:

```
>>> print(people.head()) # Τυπώνονται μόνο οι πρώτες εγγραφές του dataframe

>>> print(people.describe()) # Τυπώνεται μια λεπτομερής στατιστική αναφορά για τις στήλες του dataframe

>>> print(people.isnull().sum()) # Τυπώνεται ο αριθμός των null τιμών κάθε στήλης του dataframe
```

3.2 Βρόχοι

Δημιουργία ενός βρόχου for που τυπώνει όλες τις ακέραιες τιμές από 1 εώς 10.

```
>>> for i in range(1, 11):

+        print(i)
```

Εκτύπωση όλων των άρτιων αριθμών στο διάστημα [0,100]

```
>>> for i in range(0, 101):

+        if i % 2 == 0:

+             print(i)
```

4 One-Hot Encoding

Σε πολλά προβλήματα μηχανικής μάθησης, καλούμαστε να χρησιμοποιήσουμε “κατηγορικά” δεδομένα. Τα κατηγορικά δεδομένα είναι μεταβλητές που δέχονται σαν τιμές “ετικέτες” (strings), αντί για αριθμητικά δεδομένα και, συνήθως, ο αριθμός των διαφορετικών τιμών είναι περιορισμένος. Ένα παράδειγμα κατηγορικής μεταβλητής είναι η χρήση της μεταβλητής “καιρός”, όταν οι πιθανές τιμές του είναι “sunny” και “rainy”.

Πολλοί αλγόριθμοι μηχανικής μάθησης, όμως, δεν μπορούν να διαχειριστούν κατηγορικά δεδομένα και απαιτούν την πρότερη επεξεργασία τους και τη μετατροπή τους σε αριθμητικά. Η απλή μετατροπή τους σε αριθμητικές τιμές (π.χ. sunny = 0, rainy = 1) εισάγει μία πληροφορία “τάξεως” (το 0 είναι μικρότερο του 1), το οποίο μπορεί να δημιουργήσει σχέσεις μεταξύ των τιμών μιας μεταβλητής που στην πράξη δεν υπάρχουν. Για τον σκοπό αυτόν και όταν δεν υπάρχουν τέτοιες σχέσεις στις τιμές μιας μεταβλητής, η μετατροπή των κατηγορικών μεταβλητών σε αριθμητικές γίνεται με τη χρήση της κωδικοποίησης One-Hot (One-Hot Encoding). Στην One-Hot κωδικοποίηση, κάθε τιμή κωδικοποιείται με ένα διάνυσμα

“ενεργοποίησης”, το οποίο περιλαμβάνει n στήλες (όπου n ο αριθμός των διαφορετικών τιμών που μπορεί να πάρει η μεταβλητή) και αποτελείται από μηδενικά σε όλες τις στήλες, εκτός από την στήλη που αντιπροσωπεύει την πραγματική τιμή της μεταβλητής. Για παράδειγμα, στο πρόβλημα “καιρός” = “sunny” ή “rainy”, οι τιμές “sunny” και “rainy” έχουν την παρακάτω κωδικοποίηση:

- Sunny = [1, 0]
- Rainy = [0, 1]

Στην Python, η μετατροπή ενός dataframe που περιέχει κατηγορικές μεταβλητές σε αριθμητικές γίνεται εύκολα με τη χρήση της βιβλιοθήκης `sklearn.preprocessing`:

```
>>> from sklearn.preprocessing import OneHotEncoder

>>> encoder = OneHotEncoder(handle_unknown=”ignore”, sparse=False) # Create encoder

>>> encoder.fit(df)

>>> df_one_hot = encoder.transform(df)
```