Udemy - Machine Learning A-Z Become Kaggle Master

Day 1: 16-Sep-19

1. Python Fundametals

Kaggle

Den Baker Machine Learning course by kaggle education

**loop:**

a=[‘a’,”b”,”23c#”,6]

For I in range(5):

print(a[i])

* a
* b
* 23c#
* 6

[‘ ’ “ ” ‘ ’ ] allowed!

print (len(a[i]))

* Error as last element in the list is integer, bz object of int type has no len() function

\*\*\*IMP\*\*\*

for i in range(x) # x should be less than list size

otherwise it will throw error as =>list index out of range

#### for loop ####

a=[1,2,3,4,5,6,7,8,9]

print (a)

print (“difference of 2”)

for i in range(1,9,2):

print(a[i])

print("now difference of 3")

for i in range(1,9,3):

print(a[i])

=>

1 2 3 4 5 6 7 8 9

difference of 2

2

4

6

8

now difference of 3

2

5

8

* Range(STARTING INDEX, LAST INDEX, JUMP BETWEEN INDEX)
* difference of 2, is same as [1::2] means from 1 to last index jumping by 2

#end

Code

#print star pattern

str1=' '

for i in range(0,9):

if i<5:

str1 +='\* ' # note the spaces '\* '

print(str1)

elif i>4:

str1 = str1[:-2] #[:-2] means reducing the stars by 2 after 4rth iteration

print(str1)

=>

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

Code end

Enumerate

string = " my name is Ajinkya"

for n, alphabet in enumerate(string):

print (alphabet,n)

0

m 1

y 2

3

n 4

a 5

m 6

e 7

8

i 9

s 10

11

A 12

j 13

i 14

n 15

k 16

y 17

a 18

it creates a index starting from 0 and assign to each of the element till end of the string

WHILE loop

Prime number code

#prime number

#num= int(input("enter number")) #for user input

num = 13

isDivisible = False;

i=2;

while i<num:

if num % i == 0:

isDivisible = True;

print("{} is Divisible by {}".format(num,i))

i+=1;

if isDivisible:

print( " {} is not prime num".format(num) )

else: print( " {} is a prime num".format(num) )

* 13 is a prime number

Functions

* print ( fun.\_\_doc\_\_) for showing all the comments inside the function
* Lamda function is an anonymous function which acts as bridge between two points keeping the data security

String

* rstrip(\_character) is used to remove the \_character
* rstrip stands for strip right
* lstrip stands for strip left
* we can strip from both side str1.rstrip(\_character).lstrip(\_characteratleft)

Slicing

* during slicing [left : right] so, right is always right-1

a='ajinkya jawale'

print (a[ : -6]) => Ajinkya #jawale gayab

print(a[3:-2]) => nkya jawa #note changes

string part2

* format() put the data in {}

Data Structures in python

1. list.append(‘\_character\_’) appends the element to the list
2. pop() pops the element from the end of the list
3. but if you provide the index in the pop() it pops the element at given index.
4. remove() function to remove
5. \*\*\* when we perform = operator on list it still shares the same memory address

x=['a','b','c']

y=x

x.append('d')

print(x)

=> ['a', 'b', 'c', 'd'] #funny na? hahaha

print(id(x))

print(id(y))

=> 2449311420744

2449311420744

\* #now we dont want to them share same memory address

\* use old. copy()

y=x.copy()

print(id(y))

=> 2449311203720 Look memory location is different

\*some list functions

1. sorted(\_list)

2. max(\_list) 3. min(\_list)

Tuple

1. Once tuple created its values can not be modified
2. Tuple= ( , , , , , , )
3. Tuples can have heterogeneous values

Sets

1. intersection
2. union
3. difference

NumPy

Ndarray basic datastructure of the numpy

All the elements has to be of same data type

ndarray is like list then why numpy?

Very fast

#want 6 10 14 as output so we have to use bundle list

#a[2,3,6] => error too many indices for array

a[[2,3,6]]

\*in numpy array if one assign array to another array it will still point to original location

a=[1,2,3] b=[4,5,6]

b=a

b[0]=7

print a =>7 2 3

print b =>7 2 3 #it will make changes in a as well bz of they share same memory address

hahahahaha ! ☺ so use copy()

x=np.arange(10)

y=x.copy()

y[0] = 99

y =>array([99, 1, 2, 3, 4, 5, 6, 7, 8, 9])

x=>array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

Pandas

# Data structures in pandas

#series and dataframe

