VIPS

योग: कर्मसु कौशलम्

IN PURSUIT OF PERFECTION

VIVEKANANDA SCHOOL OF ENGINEERING AND TECHNOLOGY

Unit 2
Foundations of Data Science

Dr. Sonakshi Vij
VSE&T

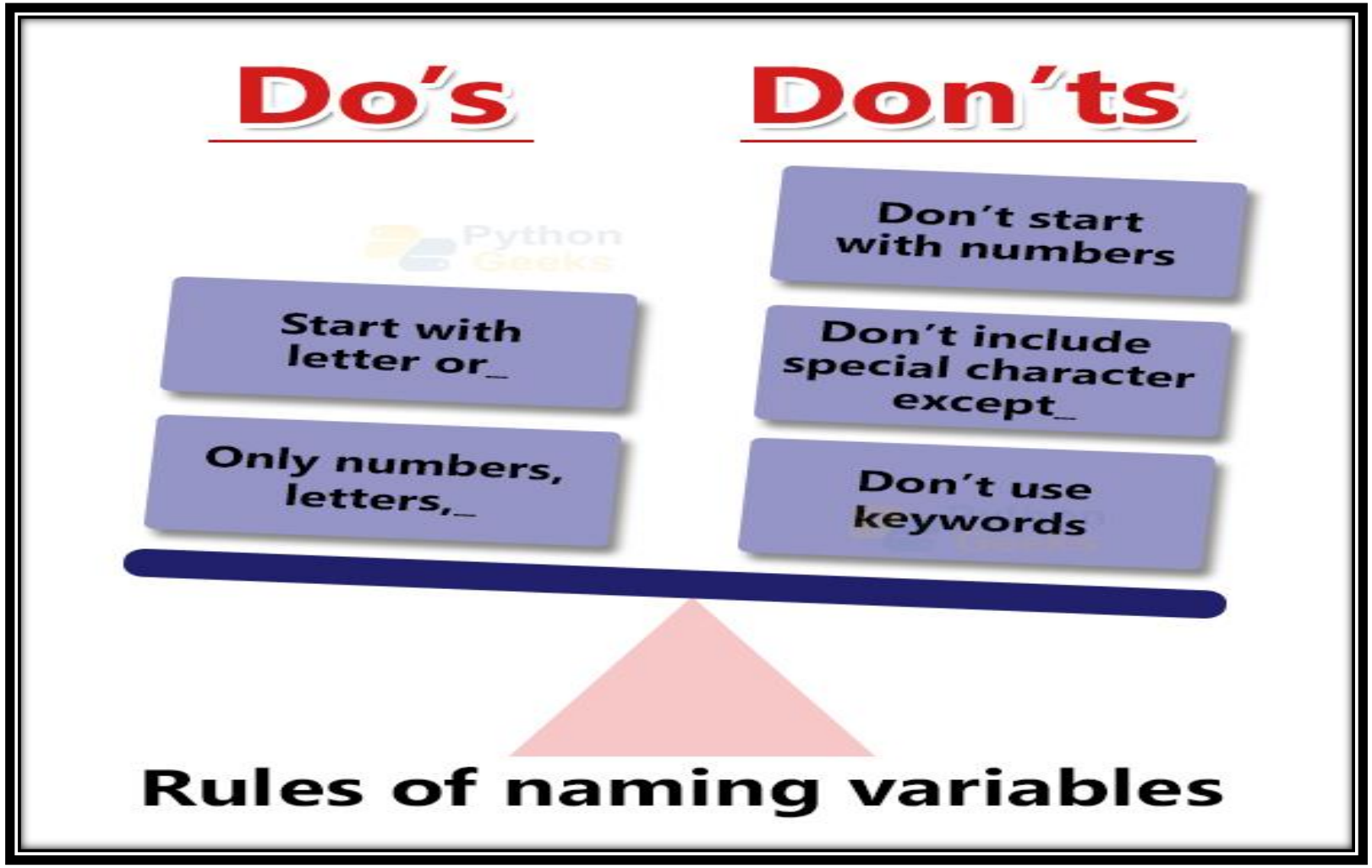
Python: Variables

- Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.
- Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Assigning Values to Variables:

- Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable. The equal sign (=) is used to assign values to variables.
- The operand to the left of the = operator is the name of the variable and the operand to the right of the = operator is the value stored in the variable.

Python: Variables



Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP\Downloads\vips work\python

C:\Users\HP\Downloads\vips work\python\experiment 2_understanding python.py

le.py | rainbow benzene turtle.py | spiral helix turtle.py | house drawing using turtle.py | experiment 2_understanding python.py*

```python
# -*- coding: utf-8 -*-
"""
Created on Thu Sep 22 09:09:06 2022

@author: HP
"""

print ("hello world")

age=29
first_name="Sonakshi"

name=input("what is your name")
print("hello"+name)

birth_year=input("enter your birth year")
age=2022-int(birth_year)
print(age)
```

Source | Console | Object

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

Variable explorer | Help | Plots | Files

Console 1/A

```
hello soankshi

enter your birth year 1992
30

In [5]: runfile('C:/Users/HP/Downloads/vips work/python/
experiment 2_understanding python.py', wdir='C:/Users/HP/
Downloads/vips work/python')
hello world

what is your name Dr. Sonakshi Vij
hello Dr. Sonakshi Vij

enter your birth year1992
30

In [6]:
```

IPython console | History

LSP Python: ready | conda (Python 3.8.8) | Line 19, Col 1 | UTF-8 | CRLF | RW | Mem 86%

Type here to search | 27°C | ENG | 09:51 22-09-2022

Python Operators

Python language supports the following types of operators.

a) Arithmetic Operators
b) Comparison (Relational) Operators
c) Assignment Operators
d) Logical Operators
e) Bitwise Operators
f) Membership Operators
g) Identity Operators

Python Operators

Operator	Description	Example
+ Addition	Adds values on either side of the operator.	a + b = 30
- Subtraction	Subtracts right hand operand from left hand operand.	a – b = -10
* Multiplication	Multiplies values on either side of the operator	a * b = 200
/ Division	Divides left hand operand by right hand operand	b / a = 2
% Modulus	Divides left hand operand by right hand operand and returns remainder	b % a = 0
** Exponent	Performs exponential (power) calculation on operators	a**b =10 to the power 20
//	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity) –	9//2 = 4 and 9.0//2.0 = 4.0, -11//3 = -4, -11.0//3 = -4.0

Python Operators

Operator	Description	Example
==	If the values of two operands are equal, then the condition becomes true.	(a == b) is not true.
!=	If values of two operands are not equal, then condition becomes true.	(a != b) is true.
<>	If values of two operands are not equal, then condition becomes true.	(a <> b) is true. This is similar to != operator.
>	If the value of left operand is greater than the value of right operand, then condition becomes true.	(a > b) is not true.
<	If the value of left operand is less than the value of right operand, then condition becomes true.	(a < b) is true.
>=	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	(a >= b) is not true.
<=	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	(a <= b) is true.

Python Operators

Operator	Description	Example
=	Assigns values from right side operands to left side operand	c = a + b assigns value of a + b into c
+= Add AND	It adds right operand to the left operand and assign the result to left operand	c += a is equivalent to c = c + a
-= Subtract AND	It subtracts right operand from the left operand and assign the result to left operand	c -= a is equivalent to c = c - a
*= Multiply AND	It multiplies right operand with the left operand and assign the result to left operand	c *= a is equivalent to c = c * a
/= Divide AND	It divides left operand with the right operand and assign the result to left operand	c /= a is equivalent to c = c / a
%= Modulus AND	It takes modulus using two operands and assign the result to left operand	c %= a is equivalent to c = c % a
**= Exponent AND	Performs exponential (power) calculation on operators and assign value to the left operand	c **= a is equivalent to c = c ** a
//= Floor Division	It performs floor division on operators and assign value to the left operand	c //= a is equivalent to c = c // a

Python Operators: Home Assessment

Operator	Description	Example
& Binary AND	Operator copies a bit to the result if it exists in both operands	(a & b) (means 0000 1100)
| Binary OR	It copies a bit if it exists in either operand.	(a | b) = 61 (means 0011 1101)
^ Binary XOR	It copies the bit if it is set in one operand but not both.	(a ^ b) = 49 (means 0011 0001)
~ Binary Ones Complement	It is unary and has the effect of 'flipping' bits.	(~a) = -61 (means 1100 0011 in 2's complement form due to a signed binary number.
<< Binary Left Shift	The left operands value is moved left by the number of bits specified by the right operand.	a << 2 = 240 (means 1111 0000)
>> Binary Right Shift	The left operands value is moved right by the number of bits specified by the right operand.	a >> 2 = 15 (means 0000 1111)

Python Operators

Operator	Description	Example
and Logical AND	If both the operands are true then condition becomes true.	(a and b) is true.
or Logical OR	If any of the two operands are non-zero then condition becomes true.	(a or b) is true.
not Logical NOT	Used to reverse the logical state of its operand.	Not(a and b) is false.

Python Operators

Operator	Description	Example
in	Evaluates to true if it finds a variable in the specified sequence and false otherwise.	x in y, here in results in a 1 if x is a member of sequence y.
not in	Evaluates to true if it does not finds a variable in the specified sequence and false otherwise.	x not in y, here not in results in a 1 if x is not a member of sequence y.

Operator	Description	Example
is	Evaluates to true if the variables on either side of the operator point to the same object and false otherwise.	x is y, here **is** results in 1 if id(x) equals id(y).
is not	Evaluates to false if the variables on either side of the operator point to the same object and true otherwise.	x is not y, here **is not** results in 1 if id(x) is not equal to id(y).

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP\Downloads\vips work\python

C:\Users\HP\Downloads\vips work\python\untitled0.py

rainbow benzene turtle.py | spiral helix turtle.py | house drawing using turtle.py | experiment 2_understanding python.py | untitled0.py*

```python
# Python program to show
# bitwise operators

a = 10
b = 4

# Print bitwise AND operation
print("a & b =", a & b)

# Print bitwise OR operation
print("a | b =", a | b)

# Print bitwise NOT operation
print("~a =", ~a)

# print bitwise XOR operation
print("a ^ b =", a ^ b)

# Python program to show arithmetic operators

a=10; b=3
print("addition of a:",a,"&b:",b,"is:",a+b)
print("substraction of a:",a,"&b:",b,"is:",a-b)
print("multiplication of a:",a,"&b:",b,"is:",a*b)
print("division of a:",a,"&b:",b,"is:",a/b)
print("floor divison of a:",a,"&b:",b,"is:",a//b)
print("moduli of a:",a,"&b:",b,"is:",a%b)
print("exponent of a:",a,"&b:",b,"is:",a**b)
```

Source | Console | Object

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

Variable explorer | Help | Plots | Files

Console 1/A

```
In [1]: runfile('C:/Users/HP/Downloads/vips work/python/untitled0.py', wdir='C:/Users/HP/Downloads/vips work/python')
a & b = 0
a | b = 14
~a = -11
a ^ b = 14
addition of a: 10 &b: 3 is: 13
substraction of a: 10 &b: 3 is: 7
multiplication of a: 10 &b: 3 is: 30
division of a: 10 &b: 3 is: 3.3333333333333335
floor divison of a: 10 &b: 3 is: 3
moduli of a: 10 &b: 3 is: 1
exponent of a: 10 &b: 3 is: 1000

In [2]:
```

IPython console | History

LSP Python: ready | conda (Python 3.8.8) | Line 30, Col 1 | UTF-8 | CRLF | RW | Mem 83%

Type here to search | 33°C | ENG | 13:09 04-10-2022

Python Operators

Precedence	Associativity	Operator	Description
18	Left-to-right	()	Parentheses (grouping)
17	Left-to-right	*f*(args...)	Function call
16	Left-to-right	*x*[index:index]	Slicing
15	Left-to-right	*x*[index]	Array Subscription
14	Right-to-left	**	Exponentiation
13	Left-to-right	~*x*	Bitwise not
12	Left-to-right	+*x* -*x*	Positive, Negative
11	Left-to-right	* / %	Multiplication Division Modulo
10	Left-to-right	+ -	Addition Subtraction
9	Left-to-right	<< >>	Bitwise left shift Bitwise right shift
8	Left-to-right	&	Bitwise AND
7	Left-to-right	^	Bitwise XOR
6	Left-to-right	|	Bitwise OR
5	Left-to-right	in, not in, is, is not, <, <=, >, >=, <>, == !=	Membership Relational Equality Inequality
4	Left-to-right	not *x*	Boolean NOT
3	Left-to-right	and	Boolean AND
2	Left-to-right	or	Boolean OR
1	Left-to-right	lambda	Lambda expression

Python: If Statement

Sr.No.	Statement & Description
1	if statements An **if statement** consists of a boolean expression followed by one or more statements.
2	if...else statements An **if statement** can be followed by an optional **else statement**, which executes when the boolean expression is FALSE.
3	nested if statements You can use one **if** or **else if** statement inside another **if** or **else if** statement(s).

Program: Even or Odd

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP\Downloads\vips work\python

C:\Users\HP\Downloads\vips work\python\untitled0.py

ore confusion matrix.py | understanding_nltk_wordnet.py | text_classification.py | converting words to features.py | untitled0.py*

```python
# Python program to check if the input number is odd or even.
# A number is even if division by 2 gives a remainder of 0.
# If the remainder is 1, it is an odd number.

num = int(input("Enter a number: "))
if (num % 2) == 0:
   print("{0} is Even".format(num))
else:
   print("{0} is Odd".format(num))
```

Source Console Object

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

Variable explorer Help Plots Files

Console 1/A

```
        one       two     three
a  0.929727  0.723468  1.059069
b  0.000000  0.000000  0.000000
c -0.017469  0.914947 -0.911446
d  0.000000  0.000000  0.000000
e  0.259261 -0.011042 -0.159459
f  0.718944 -0.145087 -0.187749
g  0.000000  0.000000  0.000000
h  0.512205  0.324055 -0.363729

In [4]: runfile('C:/Users/HP/Downloads/vips work/python/
untitled0.py', wdir='C:/Users/HP/Downloads/vips work/python')

Enter a number: 7
7 is Odd
```

Python: Loops

Python programming language provides following types of loops to handle looping requirements.

Sr.No.	Loop Type & Description
1	while loop Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body.
2	for loop Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
3	nested loops You can use one or more loop inside any another while, for or do..while loop.

Program: Factorial

VIPS Technical Campus — COLLEGE OF ENGINEERING

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP\Downloads\vips work\python

C:\Users\HP\Downloads\vips work\python\untitled0.py

core confusion matrix.py | understanding_nltk_wordnet.py | text_classification.py | converting words to features.py | untitled0.py*

```python
# Python program to find the factorial of a number provided by the user.

# change the value for a different result
num = 9

# To take input from the user
#num = int(input("Enter a number: "))

factorial = 1

# check if the number is negative, positive or zero
if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   for i in range(1,num + 1):
       factorial = factorial*i
   print("The factorial of",num,"is",factorial)
```

Source | Console | Object

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

Variable explorer | Help | Plots | Files

Console 1/A

```
In [5]: runfile('C:/Users/HP/Downloads/vips work/python/untitled0.py', wdir='C:/Users/HP/Downloads/vips work/python')
The factorial of 7 is 5040
```

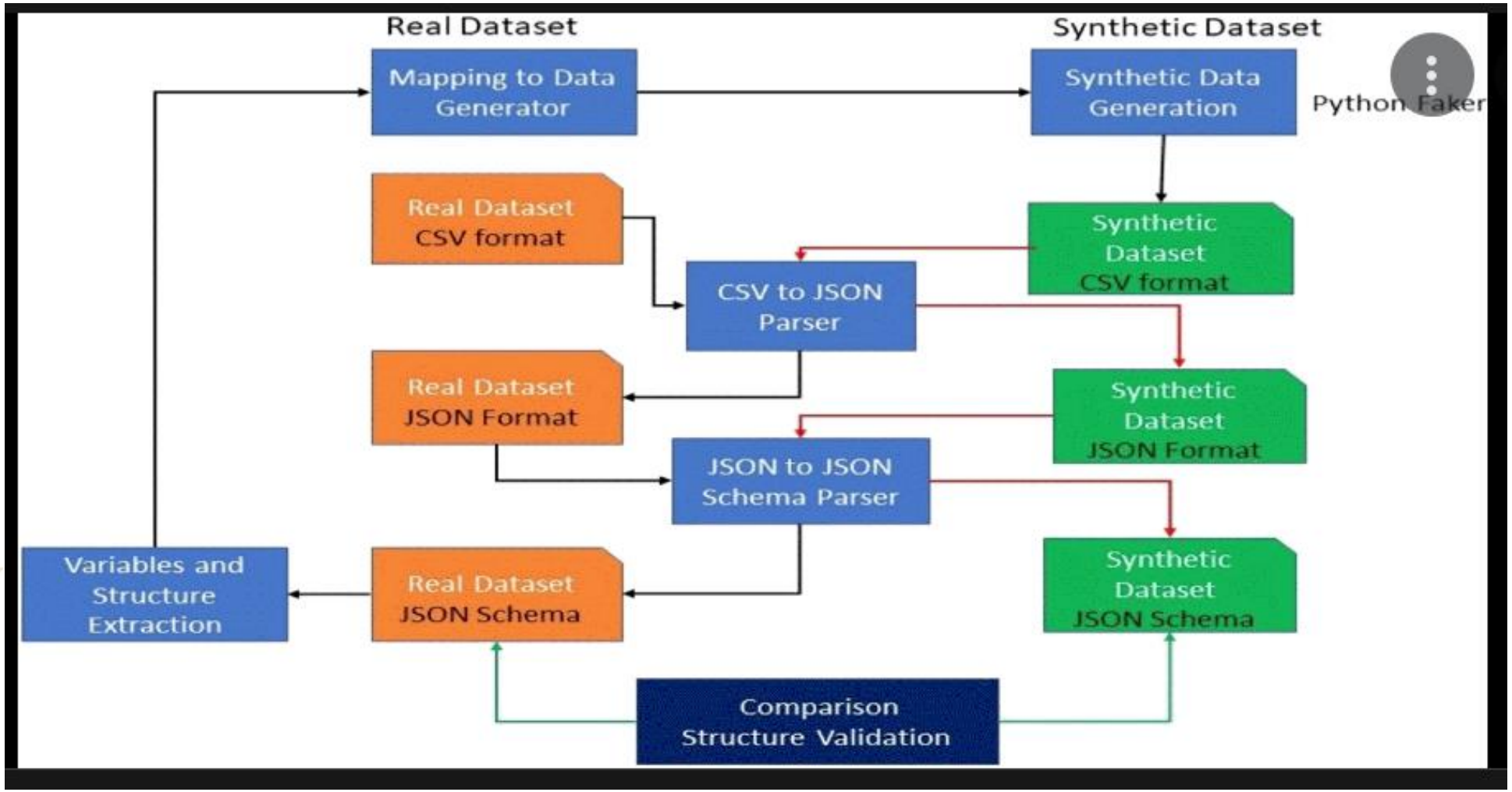
Synthetic Data Vault (SDV): A Python Library for Dataset Modelling

- In data science, you usually need a realistic dataset to test your proof of concept.
- Creating fake data that captures the behavior of the actual data may sometimes be a rather tricky task.
- Several python packages try to achieve this task.
- Few popular python packages are **Faker**, **Mimesis**.
- However, there are mostly generating simple data like generating names, addresses, emails, *etc*.

Synthetic Data Vault (SDV): A Python Library for Dataset Modelling

- To create data that captures the attributes of a complex dataset, like having time-series that somehow capture the actual data's statistical properties, we will need a tool that generates data using different approaches.
- **Synthetic Data Vault (SDV)** python library is a tool that models complex datasets using statistical and machine learning models.
- This tool can be a great new tool in the toolbox of anyone who works with data and modeling.

Synthetic Dataset Generation



Inbox (48) - sonakshi.vij@vips.edu
Netflix Movies and TV Shows | Ka
kaggle.com/datasets/shivamb/netflix-shows?resource=download
Symmetry | Free Ful... ATAL - AICTE Traini... dr shravan kumar in... Vulnerability Scann... Learning Cyber Inci... Colleges of Universi... Learn How to Test Y... Software Project M...
Search

Netflix Movies and TV Shows

Data | Code (1103) | Discussion (60) | Metadata

6872 | New Notebook | Download (1 MB)

About this file

All TV Shows and Movies meta data on Netflix. Updated every month.

show_id	type	title	director	cast
Unique ID for every Movie / Tv Show	Identifier - A Movie or TV Show	Title of the Movie / Tv Show	Director of the Movie	Actors involved in the movie / show
8807 unique values	Movie 70% TV Show 30%	8807 unique values	[null] 30% Rajiv Chilaka 0% Other (6154) 70%	[null] David Attenborough Other (7963)
s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	
s2	TV Show	Blood & Water		Ama Qamata, Khosi Ngema, Gail

archive (33).zip
Show all
Type here to search
32°C ENG 10:59 03-10-2022

Synthetic Dataset Generation

VIPS Technical Campus
योग: कर्मसु कौशलम्
IN PURSUIT OF PERFECTION
COLLEGE OF ENGINEERING

netflix_titles [Read-Only] - Excel

SONAKSHI VIJ

File | Home | Insert | Page Layout | Formulas | Data | Review | View | Help | Power Pivot | Tell me what you want to do

Clipboard | Font | Alignment | Number | Styles | Cells | Editing

Calibri 11 | Wrap Text | Merge & Center | General | Conditional Formatting | Format as Table | Cell Styles | Insert | Delete | Format | Sort & Filter | Find & Select

A1 | show_id

	A	B	C	D	E	F	G	H	I	J	K	L
1	show_id	type	title	director	cast	country	date_adde	release_ye	rating	duration	listed_in	description
2	s1	Movie	Dick Johns	Kirsten Johnson		United Sta	Septembe	2020	PG-13	90 min	Document	As her father nears the end of his life, filmmaker Kirsten Johnson stages his death in inv
3	s2	TV Show	Blood & Water		Ama Qam	South Afri	Septembe	2021	TV-MA	2 Seasons	Internatio	After crossing paths at a party, a Cape Town teen sets out to prove whether a private-s
4	s3	TV Show	Ganglands	Julien Lecl	Sami Bouajila, Tracy		Septembe	2021	TV-MA	1 Season	Crime TV S	To protect his family from a powerful drug lord, skilled thief Mehdi and his expert team
5	s4	TV Show	Jailbirds New Orleans				Septembe	2021	TV-MA	1 Season	Docuserie	Feuds, flirtations and toilet talk go down among the incarcerated women at the Orleans
6	s5	TV Show	Kota Factory		Mayur Mc	India	Septembe	2021	TV-MA	2 Seasons	Internatio	In a city of coaching centers known to train Indiaâ€™s finest collegiate minds, an earnes
7	s6	TV Show	Midnight I	Mike Flana	Kate Siegel, Zach Gilf		Septembe	2021	TV-MA	1 Season	TV Drama:	The arrival of a charismatic young priest brings glorious miracles, ominous mysteries ar
8	s7	Movie	My Little P	Robert Cu	Vanessa Hudgens, Ki		Septembe	2021	PG	91 min	Children &	Equestria's divided. But a bright-eyed hero believes Earth Ponies, Pegasi and Unicorns s
9	s8	Movie	Sankofa	Haile Geri	Kofi Ghana	United Sta	Septembe	1993	TV-MA	125 min	Dramas, I	On a photo shoot in Ghana, an American model slips back in time, becomes enslaved o
10	s9	TV Show	The Great	Andy Dev	Mel Giedr	United Kin	Septembe	2021	TV-14	9 Seasons	British TV	A talented batch of amateur bakers face off in a 10-week competition, whipping up the
11	s10	Movie	The Starlir	Theodore	Melissa M	United Sta	Septembe	2021	PG-13	104 min	Comedies,	A woman adjusting to life after a loss contends with a feisty bird that's taken over her g
12	s11	TV Show	Vendetta: Truth, Lies and The Mafia				Septembe	2021	TV-MA	1 Season	Crime TV S	Sicily boasts a bold "Anti-Mafia" coalition. But what happens when those trying to bring
13	s12	TV Show	Bangkok B	Kongkiat K	Sukollawat Kanarot,		Septembe	2021	TV-MA	1 Season	Crime TV S	Struggling to earn a living in Bangkok, a man joins an emergency rescue service and rea
14	s13	Movie	Je Suis Kar	Christian S	Luna Wed	Germany,	Septembe	2021	TV-MA	127 min	Dramas, I	After most of her family is murdered in a terrorist bombing, a young woman is unknow
15	s14	Movie	Confessio	Bruno Gar	Klara Castanho, Lucc		Septembe	2021	TV-PG	91 min	Children &	When the clever but socially-awkward TetÃª joins a new school, she'll do anything to fit
16	s15	TV Show	Crime Stories: India Detectives				Septembe	2021	TV-MA	1 Season	British TV	Cameras following Bengaluru police on the job offer a rare glimpse into the complex ar
17	s16	TV Show	Dear White People		Logan Bro	United Sta	Septembe	2021	TV-MA	4 Seasons	TV Comed	Students of color navigate the daily slights and slippery politics of life at an Ivy League c
18	s17	Movie	Europe's M	Pedro de	Echave GarcÃ-a, Pablo		Septembe	2020	TV-MA	67 min	Document	Declassified documents reveal the post-WWII life of Otto Skorzeny, a close Hitler ally w
19	s18	TV Show	Falsa identidad		Luis Ernes	Mexico	Septembe	2020	TV-MA	2 Seasons	Crime TV S	Strangers Diego and Isabel flee their home in Mexico and pretend to be a married coup
20	s19	Movie	Intrusion	Adam Salk	Freida Pinto, Logan M		Septembe	2021	TV-14	94 min	Thrillers	After a deadly home invasion at a coupleâ€™s new dream house, the traumatized wife
21	s20	TV Show	Jaguar		Blanca SuÃ¡rez, IvÃ¡r		Septembe	2021	TV-MA	1 Season	Internatio	In the 1960s, a Holocaust survivor joins a group of self-trained spies who seek justice ag
22	s21	TV Show	Monsters	Olivier Megaton			Septembe	2021	TV-14	1 Season	Crime TV S	In the late 1970s, an accused serial rapist claims multiple personalities control his behav

netflix_titles

Ready | Accessibility: Unavailable | 100%

Importing Data in Python

- When running python programs, we need to use datasets for data analysis.
- Python has various modules which help us in importing the external data in various file formats to a python program.
- In this example we will see how to import data of various formats to a python program

Importing Data in Python

Importing a CSV file

- The csv module enables us to read each of the row in the file using a comma as a delimiter.
- We first open the file in read only mode and then assign the delimiter.
- Finally use a for loop to read each row from the csv file.

Importing Data in Python

Example

```
import csv

with open("E:\customers.csv",'r') as custfile:
rows=csv.reader(custfile,delimiter=',')
for r in rows:
print(r)
```

Output

Running the above code gives us the following result –

```
['customerID', 'gender', 'Contract', 'PaperlessBilling', 'Churn']
['7590-VHVEG', 'Female', 'Month-to-month', 'Yes', 'No']
['5575-GNVDE', 'Male', 'One year', 'No', 'No']
['3668-QPYBK', 'Male', 'Month-to-month', 'Yes', 'Yes']
['7795-CFOCW', 'Male', 'One year', 'No', 'No']
……
…….
```

Delimeter

- A delimiter is the symbol or space which separates the data you wish to split.
- For example, if your column reads “Smith, John” you would select “Comma” as your delimiter.

Data Cleaning

- Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.
- When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.
- Data cleaning is a process by which inaccurate, poorly formatted, or otherwise messy data is organized and corrected.
- For example, if you conduct a survey and ask people for their phone numbers, people may enter their numbers in different formats.

DATA SCIENCE LIFECYCLE

sudeep.co

01 BUSINESS UNDERSTANDING

Ask relevant questions and define objectives for the problem that needs to be tackled.

02 DATA MINING

Gather and scrape the data necessary for the project.

03 DATA CLEANING

Fix the inconsistencies within the data and handle the missing values.

04 DATA EXPLORATION

Form hypotheses about your defined problem by visually analyzing the data.

05 FEATURE ENGINEERING

Select important features and construct more meaningful ones using the raw data that you have.

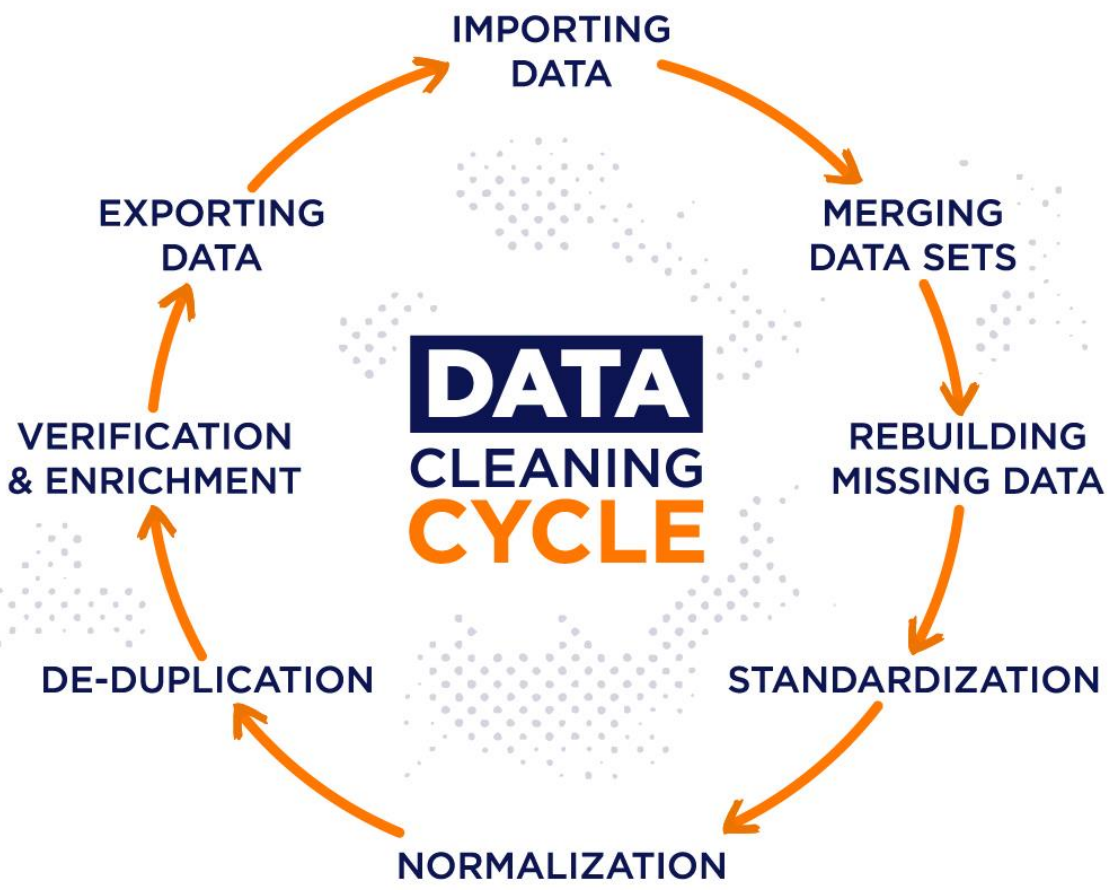
06 PREDICTIVE MODELING

Train machine learning models, evaluate their performance, and use them to make predictions.

07 DATA VISUALIZATION

Communicate the findings with key stakeholders using plots and interactive visualizations.

Data Cleaning



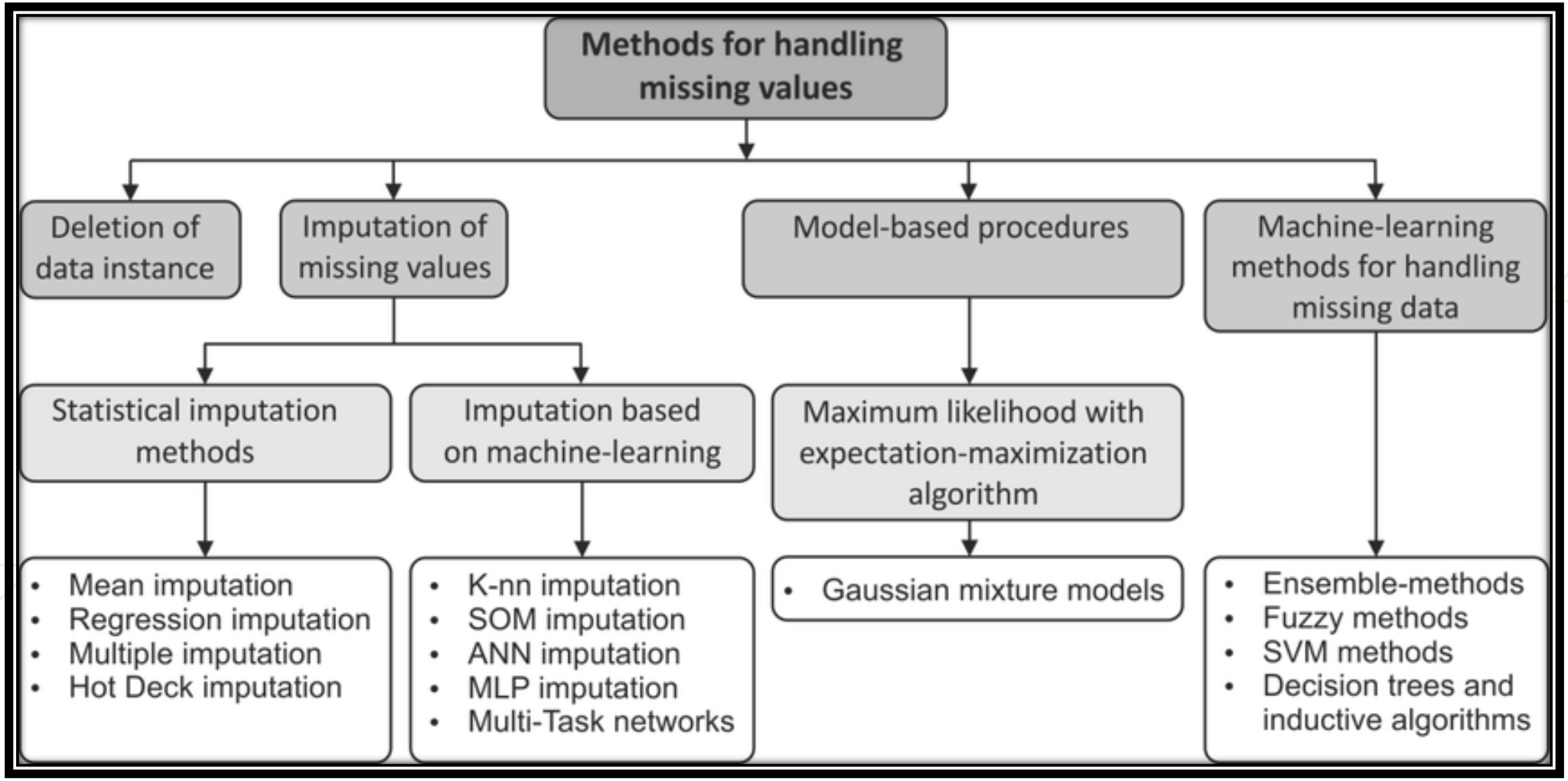
DATA WRANGLING
VERSUS
DATA CLEANING

DATA WRANGLING	DATA CLEANING
Process of transforming and mapping data from one raw data form into another form with the intent of making it more appropriate and valuable for various tasks	Process of detecting and removing corrupted or inaccurate records from a record set, table or database
Data munging is another name for data wrangling	Data cleansing is another name for data cleaning

Missing Values in Python

- Let us consider an online survey for a product.
- Many a times, people do not share all the information related to them.
- Few people share their experience, but not how long they are using the product; few people share how long they are using the product, their experience but not their contact information.
- Thus, in some or the other way a part of data is always missing, and this is very common in real time.
- Let us now see how we can handle missing values (say NA or NaN) using Pandas.

Missing Values in Python



Missing Values in Python

Imputation is a technique used for replacing the missing data with some substitute value to retain most of the data/information of the dataset.

Data Frames

- A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.
- You can think of it as an SQL table or a spreadsheet data representation.
- A pandas DataFrame can be created using various inputs like –
  a) Lists
  b) dict
  c) Series
  d) Numpy ndarrays
  e) Another DataFrame

Data Frames

Example

```
#import the pandas library and aliasing as pd
import pandas as pd
df = pd.DataFrame()
print df
```

Its **output** is as follows –

```
Empty DataFrame
Columns: []
Index: []
```

Data Frames

```
import pandas as pd
data = [1,2,3,4,5]
df = pd.DataFrame(data)
print df
```

Its **output** is as follows –

```
      0
0     1
1     2
2     3
3     4
4     5
```

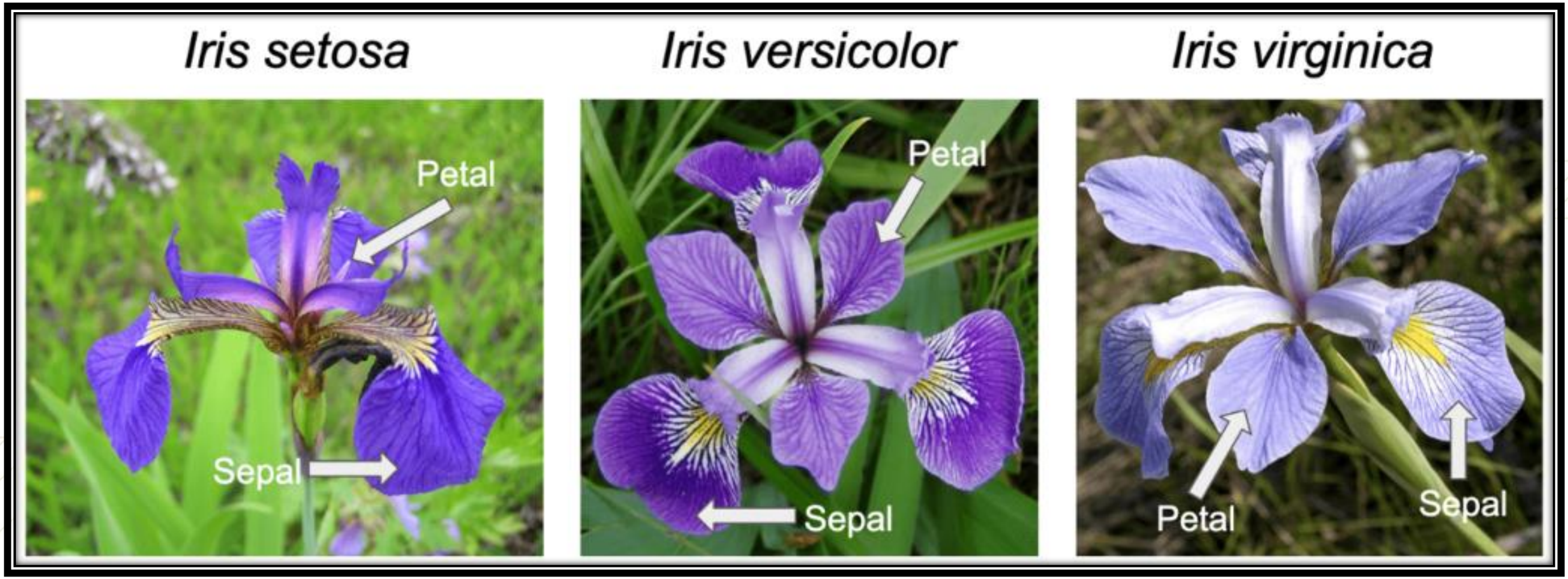
UCI-MLR

The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms

archive.ics.uci.edu/ml/datasets.php

Census Income	Multivariate	Classification	Categorical, Integer	48842	14	1996
Chess (King-Rook vs. King-Knight)	Multivariate, Data-Generator	Classification	Categorical, Integer		22	1988
Chess (King-Rook vs. King-Pawn)	Multivariate	Classification	Categorical	3196	36	1989
Chess (King-Rook vs. King)	Multivariate	Classification	Categorical, Integer	28056	6	1994
Chess (Domain Theories)	Domain-Theory					
Bach Chorales	Univariate, Time-Series		Categorical, Integer	100	6	
Connect-4	Multivariate, Spatial	Classification	Categorical	67557	42	1995
Credit Approval	Multivariate	Classification	Categorical, Integer, Real	690	15	
Japanese Credit Screening	Multivariate, Domain-	Classification	Categorical,	125		1992

Iris Dataset



Iris Dataset

Iris - Excel

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa

Iris Dataset

- Iris Data set contains information about 3 different species of Iris plant, with 50 instances for each of the species.
- It is a multivariate dataset normally used for the classification tasks using input numeric features and multiclass output.
- This dataset is readily available at the UCI Machine learning repository, and contains 4 input attributes for each instance.
- We will use these continuous attributes to develop a classification model for 3 species of Iris plant.
- Iris Dataset Link: https://archive.ics.uci.edu/ml/datasets/iris

Iris Dataset

Input Features

- **Sepal Length:** continuous; depicts the sepal length of Iris plant in cm
- **Sepal Width:** continuous; depicts the sepal width of Iris plant in cm
- **Petal Length:** continuous; depicts the petal length of Iris plant in cm
- **Petal Width:** continuous depicts the petal width of Iris plant in cm

Output Features

- **class:** discrete – Iris Setosa, Iris Versicolor, Iris Virginica; depicts the species of Iris plant

Iris Dataset: Discrete Data

- Discrete data is the type of data that has clear spaces between values.
- Continuous data is data that falls in a constant sequence.
- Discrete data is countable while continuous data is measurable

Iris Dataset: Discrete Data

- Discrete data is a count that involves integers.
- Only a limited number of values is possible.
- The discrete values cannot be subdivided into parts.
- For example, the number of children in a school is discrete data.
- You can count whole individuals. You can't count 1.5 kids.
- So, discrete data can take only certain values.

Iris Dataset: Continuous Data

- Continuous data is information that could be meaningfully divided into finer levels.
- It can be measured on a scale or continuum and can have almost any numeric value.
- For example, you can measure your height at very precise scales — meters, centimeters, millimeters and etc.
- You can record continuous data at so many different measurements – width, temperature, time, and etc.
- The continuous variables can take any value between two numbers. For example, between 50 and 72 inches, there are literally millions of possible heights: 52.04762 inches, 69.948376 inches and etc.

Deciding between different data types

A good common rule for defining if a data is continuous or discrete is that if the point of measurement can be reduced in half and still make sense, the data is continuous.

- We can display continuous data by histograms. Line graphs are also very helpful for displaying trends in continuous data.
- We can display discrete data by bar graphs. Stem-and-leaf-plot and pie chart are great for displaying discrete data too.

Iris Dataset: Advanced Libraries

- pandas
- sklearn.preprocessing
- sklearn.model_selection
- sklearn.svm
- sklearn.ensemble
- sklearn.metrics
- seaborn
- matplotlib.pyplot

Iris Dataset: Advanced Libraries

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt
```

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP\Downlo

C:\Users\HP\Downloads\vips work\python\untitled1.py

spiral helix turtle.py | house drawing using turtle.py | experiment 2_understanding python.py | untitled0.py* | untitl

```python
import pandas as pd
df=pd.read_csv(r'C:\Users\HP\Downloads\vips work\pandas\Iris.csv')

#printing the data:
print(df.head())
```

Variable explorer | Help | Plots | Files

Console 1/A

```
untitled1.py', wdir='C:/Users/HP/Downloads/vips work/python')
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
0   1            5.1           3.5            1.4           0.2
Iris-setosa
1   2            4.9           3.0            1.4           0.2
Iris-setosa
2   3            4.7           3.2            1.3           0.2
Iris-setosa
3   4            4.6           3.1            1.5           0.2
Iris-setosa
4   5            5.0           3.6            1.4           0.2
Iris-setosa

In [19]: runfile('C:/Users/HP/Downloads/vips work/python/
untitled1.py', wdir='C:/Users/HP/Downloads/vips work/python')
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
```

IPython console | History

Python: ready | conda (Python 3.8.8) | Line 11, Col 1 | UTF-8 | CRLF | RW | Mem 93%

22°C ENG 10:56 10-10-2022

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP\Downloads\vips work\python\untitled1.py

spiral helix turtle.py | house drawing using turtle.py | experiment 2_understanding python.py | untitled0.py* | untitled1.py*

```python
import pandas as pd
df=pd.read_csv(r'C:\Users\HP\Downloads\vips work\pandas\Iris.csv')

#printing the data:
print(df.head())

#rows and columns can be determined by:
print(df.shape)
```

Variable explorer | Help | Plots | Files

Console 1/A

```
In [19]: runfile('C:/Users/HP/Downloads/vips work/python/
untitled1.py', wdir='C:/Users/HP/Downloads/vips work/python')
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
0   1            5.1           3.5            1.4           0.2
Iris-setosa
1   2            4.9           3.0            1.4           0.2
Iris-setosa
2   3            4.7           3.2            1.3           0.2
Iris-setosa
3   4            4.6           3.1            1.5           0.2
Iris-setosa
4   5            5.0           3.6            1.4           0.2
Iris-setosa
(150, 6)

In [20]: runcell(0, 'C:/Users/HP/Downloads/vips work/python/
```

IPython console | History

on: ready | conda (Python 3.8.8) | Line 11, Col 1 | UTF-8 | CRLF | RW | Mem 95%

22°C ENG 10:57 10-10-2022

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP\Downloads\vips work\python\untitled1.py

spiral helix turtle.py | house drawing using turtle.py | experiment 2_understanding python.py | un

```python
import pandas as pd
df=pd.read_csv(r'C:\Users\HP\Downloads\vips work\pandas\Iris.csv')

#printing the data:
print(df.head())

#rows and columns can be determined by:
print(df.shape)

print(df.isnull())
```

Variable explorer | Help | Plots | Files

Console 1/A

```
(150, 6)
        Id  SepalLengthCm  SepalWidthCm  PetalLengthCm
PetalWidthCm  Species
0    False          False         False          False
False    False
1    False          False         False          False
False    False
2    False          False         False          False
False    False
3    False          False         False          False
False    False
4    False          False         False          False
False    False
..     ...            ...           ...            ...
...     ...
145  False          False         False          False
False    False
```

IPython console | History

Python: ready | conda (Python 3.8.8) | Line 11, Col 1 | UTF-8 | CRLF | RW | Mem 94%

22°C ENG 10:58 10-10-2022

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP\Downloads\vips work\python

C:\Users\HP\Downloads\vips work\python\untitled1.py

spiral helix turtle.py | house drawing using turtle.py | experiment 2_understanding python.py | untitled0.py* | untitled1.py*

```python
import pandas as pd
df=pd.read_csv(r'C:\Users\HP\Downloads\vips work\pandas\Iris.csv')

#printing the data:
print(df.head())

#rows and columns can be determined by:
print(df.shape)

#print all values that are null in terms of true or false
print(df.isnull())


print(df.isnull().sum())


print(df.isnull().sum().sum())
```

Name	Type	Size	Value
data	list	5	[1, 2, 3, 4, 5]
df	DataFrame	(150, 6)	Column names: Id, SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWid ...

Variable explorer | Help | Plots | Files

Console 1/A

```
False    False
148  False          False          False          False
False    False
149  False          False          False          False
False    False

[150 rows x 6 columns]
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
0

In [25]:
```

IPython console | History

LSP Python: ready | conda (Python 3.8.8) | Line 17, Col 30 | UTF-8 | CRLF | RW | Mem 82%

Type here to search | 22°C | ENG | 11:13 10-10-2022

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP\Downloads\vips work\python

C:\Users\HP\Downloads\vips work\python\untitled1.py

spiral helix turtle.py | house drawing using turtle.py | experiment 2_understanding python.py | untitled0.py* | untitled1.py*

```python
import pandas as pd
df=pd.read_csv(r'C:\Users\HP\Downloads\vips work\pandas\Iris.csv')

#printing the data:
print(df.head())

#rows and columns can be determined by:
print(df.shape)

#print all values that are null in terms of true or false
print(df.isnull())


print(df.isnull().sum())


print(df.isnull().sum().sum())
df2=df.fillna(value=0)

print(df2)
```

Name	Type	Size	Value
data	list	5	[1, 2, 3, 4, 5]
df	DataFrame	(150, 6)	Column names: Id, SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWid ...
df2	DataFrame	(150, 6)	Column names: Id, SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWid ...

Variable explorer | Help | Plots | Files

Console 1/A

```
0
      Id  SepalLengthCm  ...  PetalWidthCm         Species
0      1            5.1  ...           0.2     Iris-setosa
1      2            4.9  ...           0.2     Iris-setosa
2      3            4.7  ...           0.2     Iris-setosa
3      4            4.6  ...           0.2     Iris-setosa
4      5            5.0  ...           0.2     Iris-setosa
..   ...            ...  ...           ...             ...
145  146            6.7  ...           2.3  Iris-virginica
146  147            6.3  ...           1.9  Iris-virginica
147  148            6.5  ...           2.0  Iris-virginica
148  149            6.2  ...           2.3  Iris-virginica
149  150            5.9  ...           1.8  Iris-virginica

[150 rows x 6 columns]

In [26]:
```

IPython console | History

LSP Python: ready | conda (Python 3.8.8) | Line 22, Col 1 | UTF-8 | CRLF | RW | Mem 85%

Type here to search | 22°C | 99% available (plugged in) | 10-10-2022

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP\Downloads\vips work\python

C:\Users\HP\Downloads\vips work\python\missing_values.py

house drawing using turtle.py | experiment 2_understanding python.py | untitled0.py* | untitled1.py* | missing_values.py*

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Oct 10 11:20:24 2022

@author: HP
"""

import pandas as pd
df=pd.read_csv(r'C:\Users\HP\Downloads\vips work\pandas\Iriss.csv')

#printing the data:
print(df.head())

#rows and columns can be determined by:
print(df.shape)

#print all values that are null in terms of true or false
print(df.isnull())


print(df.isnull().sum())


print(df.isnull().sum().sum())
df2=df.fillna(value=0)

print(df2)

print(df2.isnull().sum())


print(df2.isnull().sum().sum())

```

Name	Type	Size	Value
data	list	5	[1, 2, 3, 4, 5]
df	DataFrame	(150, 6)	Column names: Id, SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWid ...
df2	DataFrame	(150, 6)	Column names: Id, SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWid ...

Variable explorer | Help | Plots | Files

Console 1/A

```
145  146            6.7  ...           2.3  Iris-virginica
146  147            6.3  ...           1.9  Iris-virginica
147  148            6.5  ...           2.0  Iris-virginica
148  149            6.2  ...           2.3  Iris-virginica
149  150            5.9  ...           1.8  Iris-virginica

[150 rows x 6 columns]
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
0

In [32]:
```

IPython console | History

LSP Python: ready | conda (Python 3.8.8) | Line 34, Col 1 | UTF-8 | CRLF | RW | Mem 85%

Type here to search | Brea... | ENG | 13:21 10-10-2022

Missing Values in Python

VIPS Technical Campus
योग: कर्मसु कौशलम्
IN PURSUIT OF PERFECTION
COLLEGE OF ENGINEERING

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP\Downloads\vips work\python

C:\Users\HP\Downloads\vips work\python\untitled0.py

ore confusion matrix.py | understanding_nltk_wordnet.py | text_classification.py | converting words to features.py | untitled0.py*

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Aug  8 14:02:16 2022

@author: HP
"""

# import the pandas library
import pandas as pd
import numpy as np

df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f',
'h'],columns=['one', 'two', 'three'])

df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])

print (df)
```

Source Console Object

Options

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

Variable explorer | Help | Plots | Files

Console 1/A

```
SyntaxError: Missing parentheses in call to 'print'. Did you mean
print(df)?


In [2]: runfile('C:/Users/HP/Downloads/vips work/python/
untitled0.py', wdir='C:/Users/HP/Downloads/vips work/python')
        one       two     three
a -0.386031  0.201007 -0.611764
b       NaN       NaN       NaN
c -0.536793 -0.919345 -2.393398
d       NaN       NaN       NaN
e -0.485307  1.101420 -0.758320
f -0.003903 -1.163808  1.135881
g       NaN       NaN       NaN
h -0.248394  0.954859  1.181379
```

Missing Values in Python

- Using reindexing, we have created a DataFrame with missing values. In the output, **NaN** means **Not a Number.**

Check for Missing Values:

- To make detecting missing values easier (and across different array dtypes), Pandas provides the **isnull()** and **notnull()** functions, which are also methods on Series and DataFrame objects

Missing Values in Python

Example

```
import pandas as pd
import numpy as np

df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f',
'h'],columns=['one', 'two', 'three'])

df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])

print df['one'].isnull()
```

Its **output** is as follows –

```
a False
b True
c False
d True
e False
f False
g True
h False
Name: one, dtype: bool
```

Missing Values in Python

Cleaning / Filling Missing Data

- Pandas provides various methods for cleaning the missing values.
- The fillna function can “fill in” NA values with non-null data in a couple of ways, which we have illustrated in the following sections.

Replace NaN with a Scalar Value

VIPS Technical Campus
योग: कर्मसु कौशलम्
IN PURSUIT OF PERFECTION
COLLEGE OF ENGINEERING

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\HP\Downloads\vips work\python

C:\Users\HP\Downloads\vips work\python\untitled0.py

core confusion matrix.py | understanding_nltk_wordnet.py | text_classification.py | converting words to features.py | untitled0.py*

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Aug  8 14:02:16 2022

@author: HP
"""

# import the pandas library
import pandas as pd
import numpy as np

df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f',
'h'],columns=['one', 'two', 'three'])

df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])

print (df)
print ("NaN replaced with '0':")
print (df.fillna(0))
```

Source | Console | Object

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

Variable explorer | Help | Plots | Files

Console 1/A

```
b       NaN       NaN       NaN
c -0.017469  0.914947 -0.911446
d       NaN       NaN       NaN
e  0.259261 -0.011042 -0.159459
f  0.718944 -0.145087 -0.187749
g       NaN       NaN       NaN
h  0.512205  0.324055 -0.363729
NaN replaced with '0':
        one       two     three
a  0.929727  0.723468  1.059069
b  0.000000  0.000000  0.000000
c -0.017469  0.914947 -0.911446
d  0.000000  0.000000  0.000000
e  0.259261 -0.011042 -0.159459
f  0.718944 -0.145087 -0.187749
g  0.000000  0.000000  0.000000
h  0.512205  0.324055 -0.363729
```

This program shows how you can replace "NaN" with "0"

Dropping Missing Values

- If you want to simply exclude the missing values, then use the **dropna** function along with the **axis** argument.
- By default, axis=0, i.e., along row, which means that if any value within a row is NA then the whole row is excluded.

Dropping Missing Values

Example

```
import pandas as pd
import numpy as np

df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f',
'h'],columns=['one', 'two', 'three'])

df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
print df.dropna()
```

Its **output** is as follows –

```
one two three
a 0.077988 0.476149 0.965836
c -0.390208 -0.551605 -2.301950
e -2.000303 -0.788201 1.510072
f -0.930230 -0.670473 1.146615
h 0.085100 0.532791 0.887415
```

Replace Missing or Generic Values

- Many times, we have to replace a generic value with some specific value. We can achieve this by applying the replace method.
- Replacing NA with a scalar value is equivalent behavior of the **fillna()** function.

Replace Missing or Generic Values

Example

```
import pandas as pd
import numpy as np
df = pd.DataFrame({'one':[10,20,30,40,50,2000],
'two':[1000,0,30,40,50,60]})
print df.replace({1000:10,2000:60})
```

Its **output** is as follows –

```
one two
0 10 10
1 20 0
2 30 30
3 40 40
4 50 50
5 60 60
```

References

- **Handling Missing Data in Python:**

https://www.tutorialspoint.com/python_pandas/python_pandas_missing_data.htm

- **Iris Dataset:**

https://www.embedded-robotics.com/iris-dataset-classification/