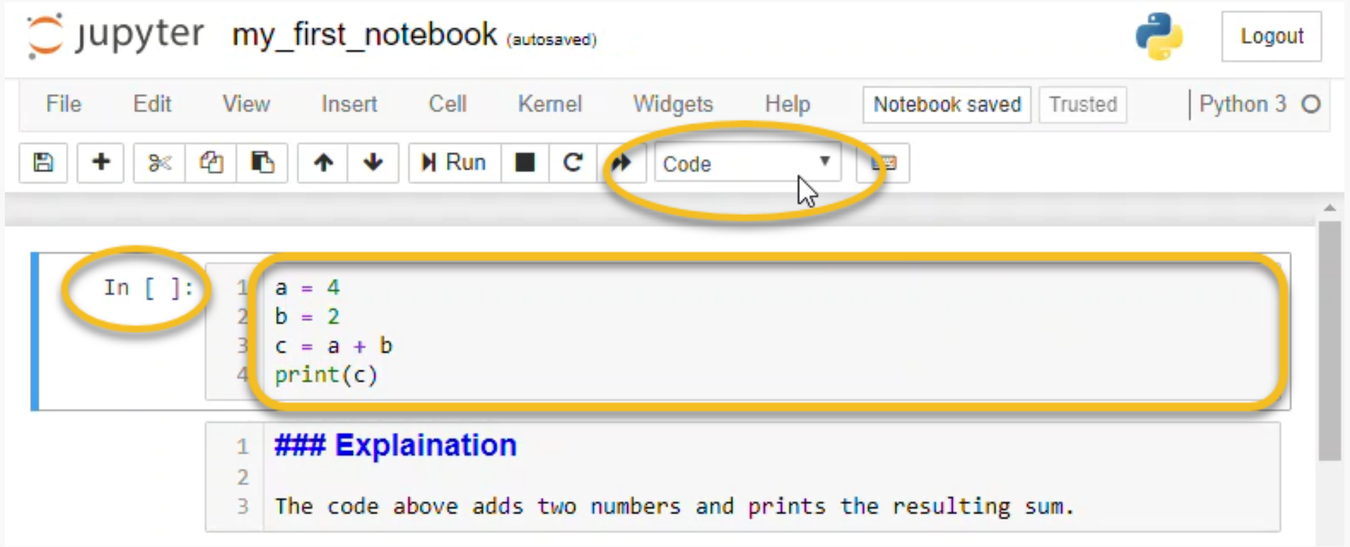
Machine learning algorithms: Practice 1.

Python libraries for ML
Classification algorithms

Alexandr Gavrilko
MLA Course for CS-2227

Developing environment: Interactive notebooks



Jupyter Notebook / Jupyter Lab

Self-hosted / Cloud

> pip install jupyterlab
> jupyter lab

> pip install notebook
> jupyter notebook

Google Colab

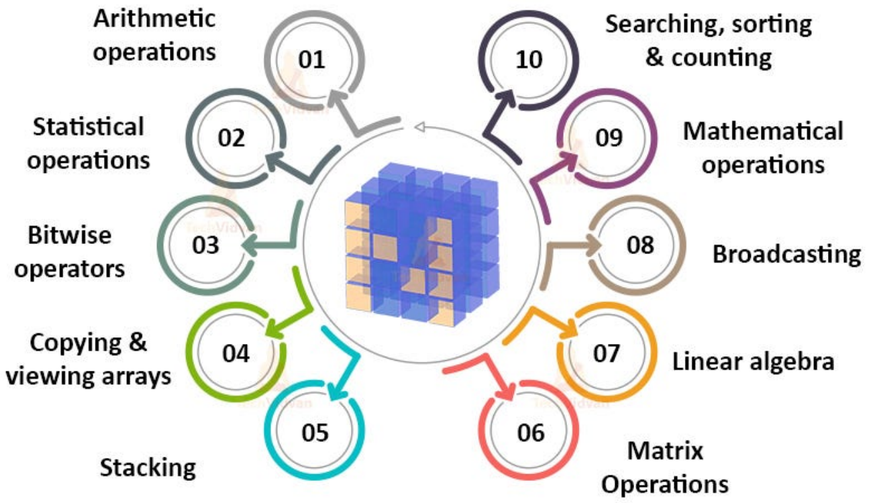
Cloud

Python libraries for Machine Learning

NumPy

> pip install numpy

Uses of NumPy




```
[15] # Import numpy library
     import numpy as np
```

```
[16] # Create a 1D array
     arr1d = np.array([1, 2, 3, 4, 5])
     print("Array 1: ", arr1d)

     # Create a 2D array
     arr2d = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
     print("Array 2: \n", arr2d)
```

```
Array 1:  [1 2 3 4 5]
Array 2:
 [[1 2 3]
 [4 5 6]
 [7 8 9]]
```

```
[17] # Shape and size of numpy arrays
     print("Shape of arr2d:", arr2d.shape)
     print("Number of elements in arr2d:", arr2d.size)
```

```
Shape of arr2d: (3, 3)
Number of elements in arr2d: 9
```

```
[28] # Indexing
     element_at_index_2 = arr1d[2]
     print(f'Element value at position 2: ', element_at_index_2)
```

```
Element value at position 2:  3
```

```
[19] # Array slicing
     sliced_arr = arr1d[1:4]
     print("Sliced array: ", sliced_arr)
```

```
Sliced array:  [2 3 4]
```

```
[20] # Element-wise addition
     added_arr = arr1d + 2
     print("Added array: ", added_arr)

     # Element-wise multiplication
     multiplied_arr = arr1d * 3
     print("Multiplied array: ", multiplied_arr)
```

```
Added array:  [3 4 5 6 7]
Multiplied array:  [ 3  6  9 12 15]
```

```
[21] # Matrix multiplication
     mat1 = np.array([[1, 2], [3, 4]])
     mat2 = np.array([[5, 6], [7, 8]])
     mat_result = np.dot(mat1, mat2)
     print("Matrix multiplication: \n", mat_result)
```

```
Matrix multiplication:
 [[19 22]
 [43 50]]
```

```
[22] # Mean and standard deviation
     mean_val = np.mean(arr1d)
     std_dev = np.std(arr1d)
     print(f'Mean: {mean_val} | Standard deviation: {std_dev}')
```

```
Mean: 3.0 | Standard deviation: 1.4142135623730951
```

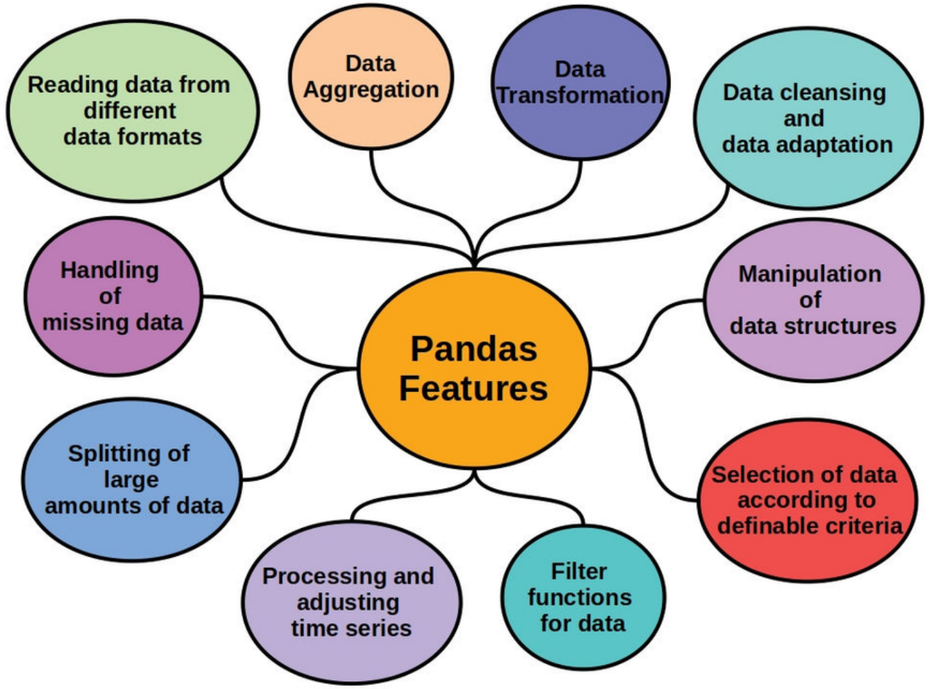
```
[23] # Generate random numbers
     random_arr = np.random.rand(3, 3)
     print("Random array: \n", random_arr)
```

```
Random array:
 [[0.56142489 0.68641876 0.93541358]
 [0.35087826 0.71485634 0.25045174]
 [0.2620588  0.64646094 0.69233354]]
```

Python libraries for Machine Learning

pandas

> pip install pandas




```
[1]  # Import pandas
     import pandas as pd
```

```
[6]  # Read data into pandas dataframe
     trips_data = pd.read_excel('./Table1.xlsx')

     # Display first 5 rows
     trips_data.head(5)
```

	Salary	City	Age	Vacation_preferences	Transport_preferences	Family members	Target
0	196000	Krasnodar	25	Shopping	Car	1	New York
1	152000	Ekaterinburg	60	Shopping	Plain	2	London
2	83000	Tomsk	49	Architecture	Train	1	Sydney
3	146000	Krasnodar	41	Architecture	Car	1	New York
4	59000	Krasnodar	58	Architecture	Car	2	Sydney

```
[13] # Checking for missing values
     print('Null values: \n', trips_data.isnull().sum(), '\n')

     # Dropping rows with missing values
     df_cleaned = trips_data.dropna()

     # Filling missing values with a specific value
     df_filled = df.fillna(0)
```

```
Null values:
 Salary                  0
City                     0
Age                      0
Vacation_preferences     0
Transport_preferences    0
Family members           0
Target                   0
dtype: int64

Original dataset shape: (40, 7)
Shape after dropping NaN values: (40, 7)
```