In [1]: print("Welcome to Numpy-1") Welcome to Numpy-1 Agenda • Installing and Importing Numpy • Introduction to use case • Motivation: Why to use Numpy? - How is it different from Python Lists? • Creating a Basic Numpy Array From a List - array(), shape, ndim From a range and stepsize - arange() type() ndarray • How numpy works under the hood? • Indexing and Slicing on 1D Indexing Slicing Masking (Fancy Indexing) Operation on array • Universal Functions (ufunc) on 1D array Aggregate Function/ Reduction functions - sum(), mean(), min(), max() • Usecase: calculate NPS loading data: np.loadtxt() np.empty() np.unique() **DETRACTORS PASSIVES PROMOTERS Net Promoter Score** % Promoters % Detractors Dataset --> https://drive.google.com/file/d/1c0ClC8SrPwJq5rrkyMKyPn80nyHcFikK/view?usp=sharing What is a good NPS score? **NEEDS IMPROVEMENT** GOOD GREAT EXCELLENT (71-100) (-100-0)100 2021 NPS Benchmarks Insurance Consulting Construction Airlines Logistics & Transport In [ ]: # Numpy --> Numerical Python In [3]: !pip install numpy Requirement already satisfied: numpy in /Users/nikhilsanghi/opt/anaconda3/lib/python3.9/site-packages (1.20.3) [notice] A new release of pip available: 22.3.1 -> 23.0 [notice] To update, run: pip install --upgrade pip In [4]: import numpy as np a=[1,2,3,4,5] a [1, 2, 3, 4, 5] In [6]: type(a) b=[i\*\*2 **for** i **in** a] b [1, 4, 9, 16, 25] In [8]: c=np.array([1,2,3,4,5]) array([1, 2, 3, 4, 5]) In [9]: type(c) Out[9]: numpy.ndarray In [10]: Out[10]: array([ 1, 4, 9, 16, 25]) Out[11]: array([3, 4, 5, 6, 7]) Out[12]: array([ 2, 4, 6, 8, 10]) In [14]: Traceback (most recent call last) TypeError: can only concatenate list (not "int") to list In [15]: array([1, 2, 3, 4, 5]) d=range(100000) range(0, 100000) %timeit [i\*\*2 for i in d] 26.1 ms  $\pm$  92.4  $\mu$ s per loop (mean  $\pm$  std. dev. of 7 runs, 10 loops each) e=np.array(range(100000)) array([ 0, 1, 2, ..., 99997, 99998, 99999]) In [25]: %timeit e\*\*2  $30.5~\mu s~\pm~905~ns$  per loop (mean  $\pm~std.$  dev. of 7 runs, 10000 loops each) In [ ]: np.array([1,2,3,4.0]) Out[26]: array([1., 2., 3., 4.]) In [27]: np.array([1,2,3,4.0,"scaler"]) Out[27]: array(['1', '2', '3', '4.0', 'scaler'], dtype='<U32') In [28]: np.array([1,2,3,4.0, "scaler", True]) array(['1', '2', '3', '4.0', 'scaler', 'True'], dtype='<U32') In [39]: Out[39]: '2', '3', '4.0'], dtype='<U56') In [30]: np.array([1,2,3,True]) array([1, 2, 3, 1]) np.array([1,2,3,4.0,True,False]) array([1., 2., 3., 4., 1., 0.]) In [35]: np.array([1,2,3,4.0,True]) array([1., 2., 3., 4., 1.]) # type(None) np.array([1,2,3,4.5,**True**]) array([1. , 2. , 3. , 4.5, 1. ]) In [ ]: In [40]: Out[40]: array([1, 2, 3, 4, 5]) c.ndim Out[41]: 1 f=np.array([[1,2,3],[4,5,6],[7,8,9]]) array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) Out[44]: f.shape Out[45]: (3, 3) type((4)) int Out[47]: In [48]: type((4,)) Out[48]: tuple In [ ]: In [49]: range(100) range(0, 100) Out[49]: In [50]: np.arange(100) array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]) [i for i in range(1,10,2)] [1, 3, 5, 7, 9] np.arange(1,10,2) array([1, 3, 5, 7, 9]) In [54]: [i for i in range(1,10,0.5)] Traceback (most recent call last) /var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel\_44594/2511718923.py in <module> ----> 1 [i for i in range(1,10,0.5)] TypeError: 'float' object cannot be interpreted as an integer In [55]: np.arange(1,10,0.5) array([1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. , 5.5, 6. , 6.5, 7. , 7.5, 8. , 8.5, 9. , 9.5]) In [ ]: np.array([1,2,3,'4'],dtype="float") array([1., 2., 3., 4.]) np.array([1,2,3,'Scaler'],dtype="float") Traceback (most recent call last) ----> 1 np.array([1,2,3,'Scaler'],dtype="float") ValueError: could not convert string to float: 'Scaler' In [ ]: In [ ]: In [60]: Out[60]: [1, 2, 3, 4, 5] Out[61]: array([1, 2, 3, 4, 5]) type(c) numpy.ndarray In [63]: c.dtype dtype('int64') g=np.array([1.0, 2.0, 3.0, 4.0, 5.0]) array([1., 2., 3., 4., 5.]) g.dtype dtype('float64') Out[66]: In [ ]: In [ ]: