TOOL

Loading Data

Before you start a problem, it is very important to know what kind of data you have. Below are some steps you should familiarize yourself with when starting a problem.

1	Look for documentation about your data
2	Read the documentation and form an understanding of your data and its structure
3	Set up how you will load your data based on the data type

Once you better understand the nature of your dataset, you will benefit from using Python packages or functions that are best suited to your data type. The following is a non-exhaustive list of python packages or functions that you can use to load data, based on the data type:

	SOLUTION	DOCUMENTATION
Structured/ Tabular Data	Python's built-in read function Pandas Numpy	https://docs.python.org/3/library/csv.html https://pandas.pydata.org/ https://www.numpy.org/
Images/ Videos	PIL opencv	https://pillow.readthedocs.io/en/stable https://docs.opencv.org/master/d9/df8/tutorial_root.html
Text Data	NLTK	https://www.nltk.org/

Example:

- If your data is stored in a .csv file and you'd like to use it to train a sci-kitlearn model, you would load it as follows: `df = pd.read_csv('data_file.csv')`
- You can preview the first n rows of the loaded data using: `df.head(n)`

Leakage:

When you are loading your data from the .csv file, be cautious of labels on additional columns. A common mistake people make when dealing with tabular data is that they include the labels of datapoint as data they are going to feed to their learning algorithms. This causes label leakage.

What is leakage? Leakage is created when unexpected additional information is provided into the training data, causing the machine learning algorithm to make unrealistic predictions. It results in inaccuracy, which is not usable. Having extra information, such as labels, simply returns it as the prediction.