MACHINE LEARNING

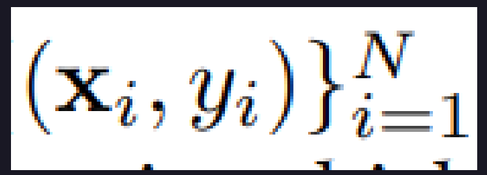
Machine learning is a subfield of computer science that is concerned with building algorithms which, to be useful, rely on a collection of examples of some phenomenon. These examples can come from nature, be handcrafted by humans or generated by another algorithm.

Machine learning can also be defined as the process of solving a practical problem by 1) gathering a dataset, and 2) algorithmically building a statistical model based on that dataset. That statistical model is assumed to be used somehow to solve the practical problem.

TYPES OF ML

- Supervised
- Unsupervised
- Re-enforcement

In supervised learning, the dataset is the collection of labeled examples:



- Let us say that we want to build a simple ML model which will predict the house price based on a number of factors.
- One can include the Location, Square Footage, Number of rooms, whether it has an open terrace or not etc. to try and determine the price of a given house.
- These "features" are represented as a vector Xi, called the feature vector.
- These feauture vectors are then mapped to a label yi by the model. Here, the labels are house prices.
- In a Supervised ML model, we feed in the input feature vectors with the labels of the training data and then try to predict the labels for the corresponding feature vectors of the tesing data.

TYPES OF ML

- Supervised
- Unsupervised
- Re-enforcement

In unsupervised learning, the dataset is a collection of unlabeled examples

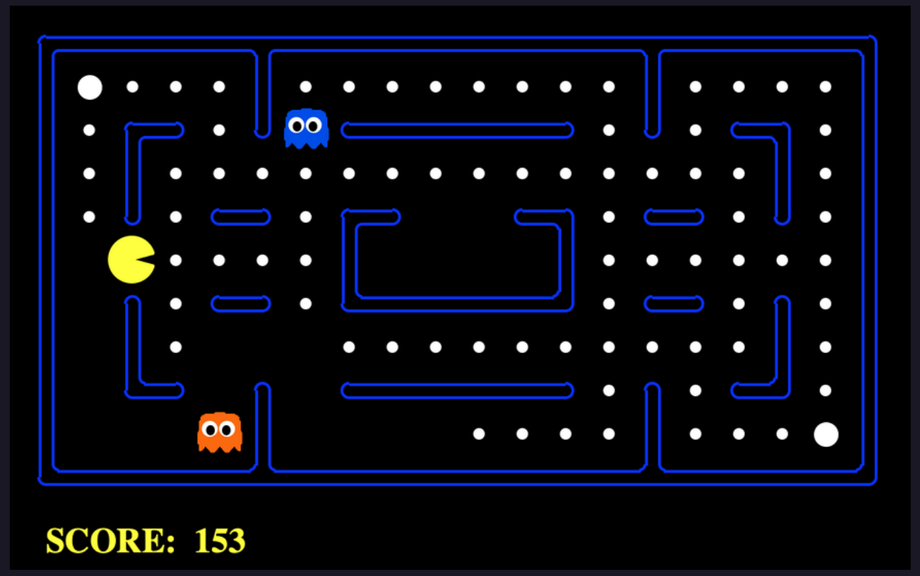
$$\{\mathbf{x}_i\}_{i=1}^{N}$$

- It is used for clustering and anamoly detection. We donot know anything about the data, and want the model to find patterns and group the data accordingly.
- Say, if you decide to buy Dove body wash products on Amazon, you'll probably be offered to add some toothpaste and a set of toothbrushes to your cart because the algorithm calculated that these products are often purchased together by other customers.

TYPES OF ML

- Supervised
- Unsupervised
- Re-enforcement

TRIAL and ERROR approch. Gain Points when a "Correct" move is made, and attempt



- A phenomenal RL application is in the Robotics field. A robot has to have the intelligence to perform unknown tasks when no success measure is given.
- It has to explore all the possibilities and reach its destined goal. A robot here is an agent if we consider the typical Reinforcement Learning nomenclature.
- The action is its movement. The environment is, say, a maze. The reward is the points that it receives on making a successful move.
- All four components taken together explain a Reinforcement Learning scenario.

YOUR FIRST STEP, NUMPY,PANDAS, MATPLOTLIB , SEABORN

- Familiarise yourself with some of the liabraries in Python.
- Numpy is a mathematical and computing library with a strong support for arrays,matrices and more. Pandas and Matplotlib,Seaborn helps you manipulate and visualise data.
- Go through the undermentioned guides to get a good understanding of how it works and of its basic functions.

Disclaimer:

Documentation & the internet is your BFF, use them before approaching us for doubts!

Topics we will be looking today :

1. Data Generation
2. Data Preprocessing
3. Data Visualization

DATA GENERATION

Methods we will discuss :

1. Using Scikit- Learn datasets
2. Web Scraping using Requests and BeautifulSoup

Using Scikit-Learn to generate data

Resources : https://scikit-learn.org/stable/datasets/sample_generators.html

1. sklearn.datasets.make_regression for generating data for a random regression problem.
2. sklearn.datasets.make_classification for generating data for a random n-class classification problem
3. sklearn.datasets.make_blobs for generating isotropic Gaussian blobs for clustering.

Web Scraping

To perform Web Scraping, we will be using the Requests and BeautifulSoup modules of Python.

Requests Module in Python: It allows us to send HTTP requests using Python. The different HTTP requests are :

1. GET: to retrieve information from the given server using a given URI.
   requests.get(url) : Sends a GET request to the specified url
2. POST : to send data to a server to create/update a resource.
   requests.post(url, parameters)
3. PUT : requests that the enclosed entity be stored under the supplied URI.
   requests.put(url, params)
4. PATCH : applies partial modifications to a resource.
5. HEAD : identical to that of a GET request, but without the response body.

BeautifulSoup Module in Python : It is used to extract information from the HTML and XML files. It provides a parse tree and the functions to navigate, search or modify this parse tree.

1. Install : pip install beautifulsoup4
2. Import in your python file : from bs4 import BeautifulSoup
3. Parsing the html : Create a BeautifulSoup object by specifying the parser we want to use. Examples of HTML parsers : html5lib, lxml, html.parser
   For more info. soup = BeautifulSoup(html_content, 'html.parser')
4. Prettifying: The prettify() method will turn a Beautiful Soup parse tree into a nicely formatted Unicode string, with a separate line for each tag and each string:
5. Searching the tree : find('tag_name') or find_all('tag_name')
   soup.find_all("a", class_="css_class_name")

Let's code :
https://colab.research.google.com/drive/1O86sGQnYYYzEHWLyV2jFoxuR0vhZc78h

Data Preprocessing

The steps involved in data preprocessing can be grouped as follow:

1. **Merging data** sets on common fields brings all data into a single table. The Pandas library contains suitable tools for merging, concatenation, and similar operations on datasets. pandas.merge() , pandas.concat() .
2. **Cleaning the data** by dealing with duplicate rows, incorrect or missing values, and other issues. Duplicated data needs to be removed to maintain accuracy and to avoid misleading statistics. A column with most of its values missing can be dropped as it provides less information. However, losing data can be bad sometimes. In that case, data imputation is done using measures of central tendency or advanced methods. Moreover, datasets with imbalanced classes are balanced using techniques like SMOTE.
3. **Feature engineering** to improve data quality, for example, using dimensionality reduction techniques (**PCA** and **LDA**) to build new features.
4. Building the training data sets by **standardizing** or **normalizing** the numerical data, **encoding** the categorical data, and splitting it into training and testing sets.

Refer to our **colab notebook** and the book '**Beginning Data Science with Python**' for examples.