Introduction to Machine Learning and ML Tools

CS-482/CS-682

Dr. Saroja Kanchi

Material Taken From IMLP and APM

Table of Contents

What is Machine Learning

"Learning is any process by which a system improves performance from experience."

- Herbert Simon

Definition by Tom Mitchell (1998):

Machine Learning is the study of algorithms that

- improve their performance P
- at some task T
- with experience E.

A well-defined learning task is given by <P, T, E>.

What is Machine Learning

T: Playing checkers
P: Percentage of games won against an arbitrary opponent
E: Playing practice games against itself

T: Recognizing hand-written words
P: Percentage of words correctly classified
E: Database of human-labeled images of handwritten words

T: Driving on four-lane highways using vision sensors
P: Average distance traveled before a human-judged error
E: A sequence of images and steering commands recorded while observing a human driver.

T: Categorize email messages as spam or legitimate.
P: Percentage of email messages correctly classified.
E: Database of emails, some with human-given labels

Machine Learning Versus Other programs

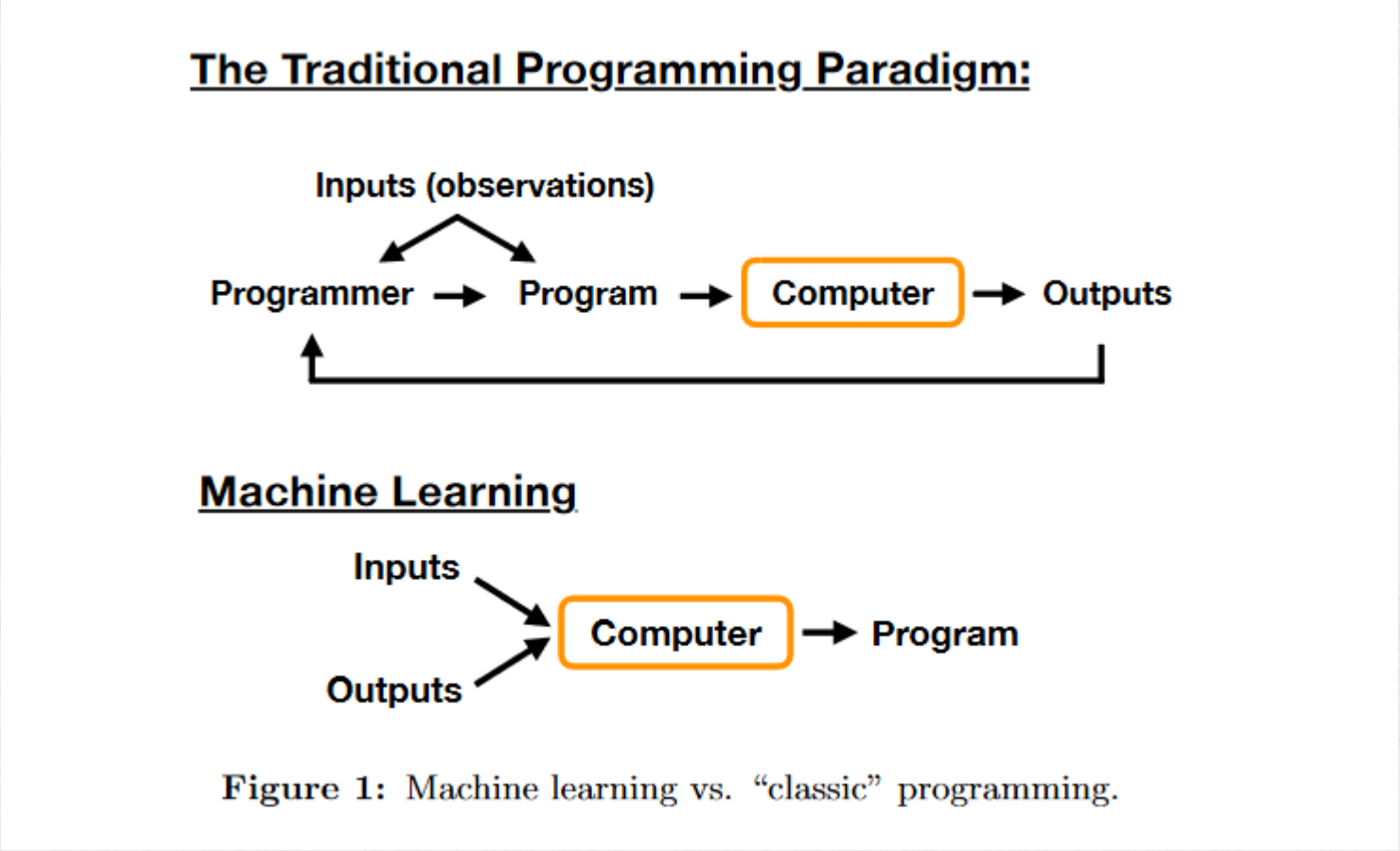


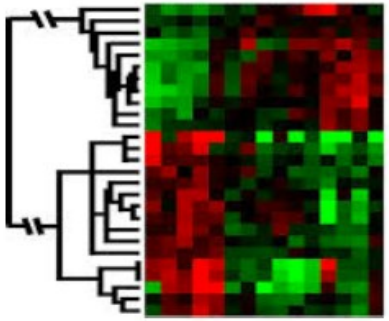
Figure 1: Machine learning vs. "classic" programming.

When do we use Machine Learning

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)



Learning isn't always useful:

- There is no need to "learn" to calculate payroll

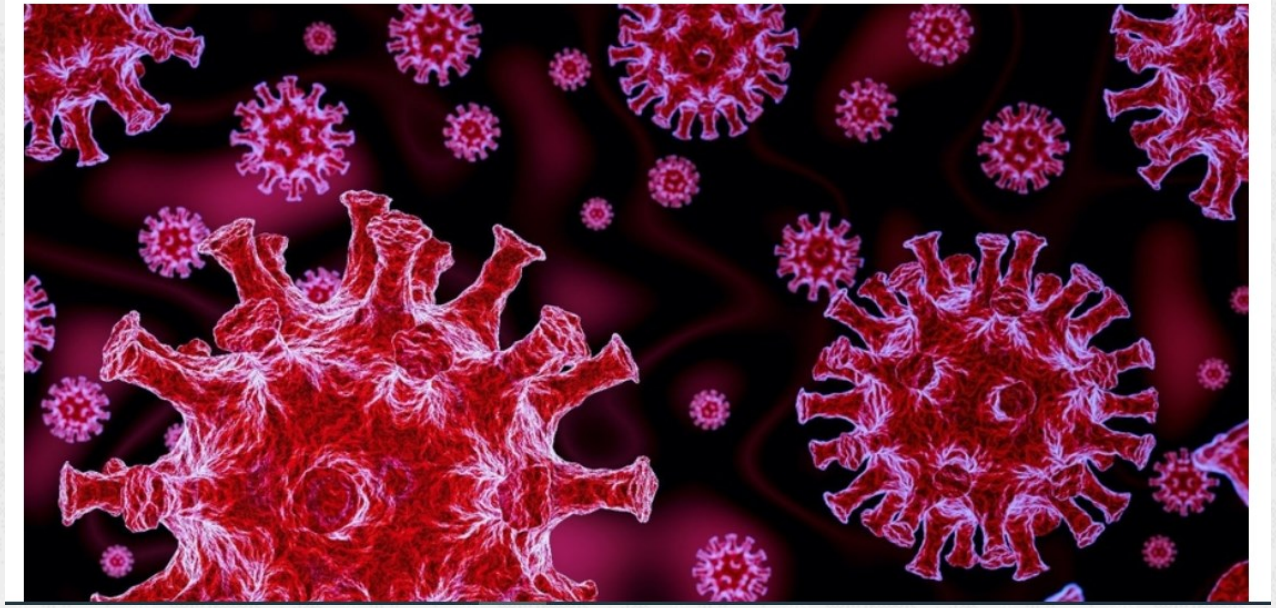
In the news…

Deep learning helps predict new drug combinations to fight Covid-19

Neural network identifies synergistic drug blends for treating viruses like SARS-CoV-2.

Rachel Gordon | MIT CSAIL

September 24, 2021



In the news..

AI Can Write Code Like Humans—Bugs and All

New tools that help developers write software also generate similar mistakes.

In the news…

Enlisting the power of AI to fight California wildfires

by Cynthia Dillon, University of Southern California

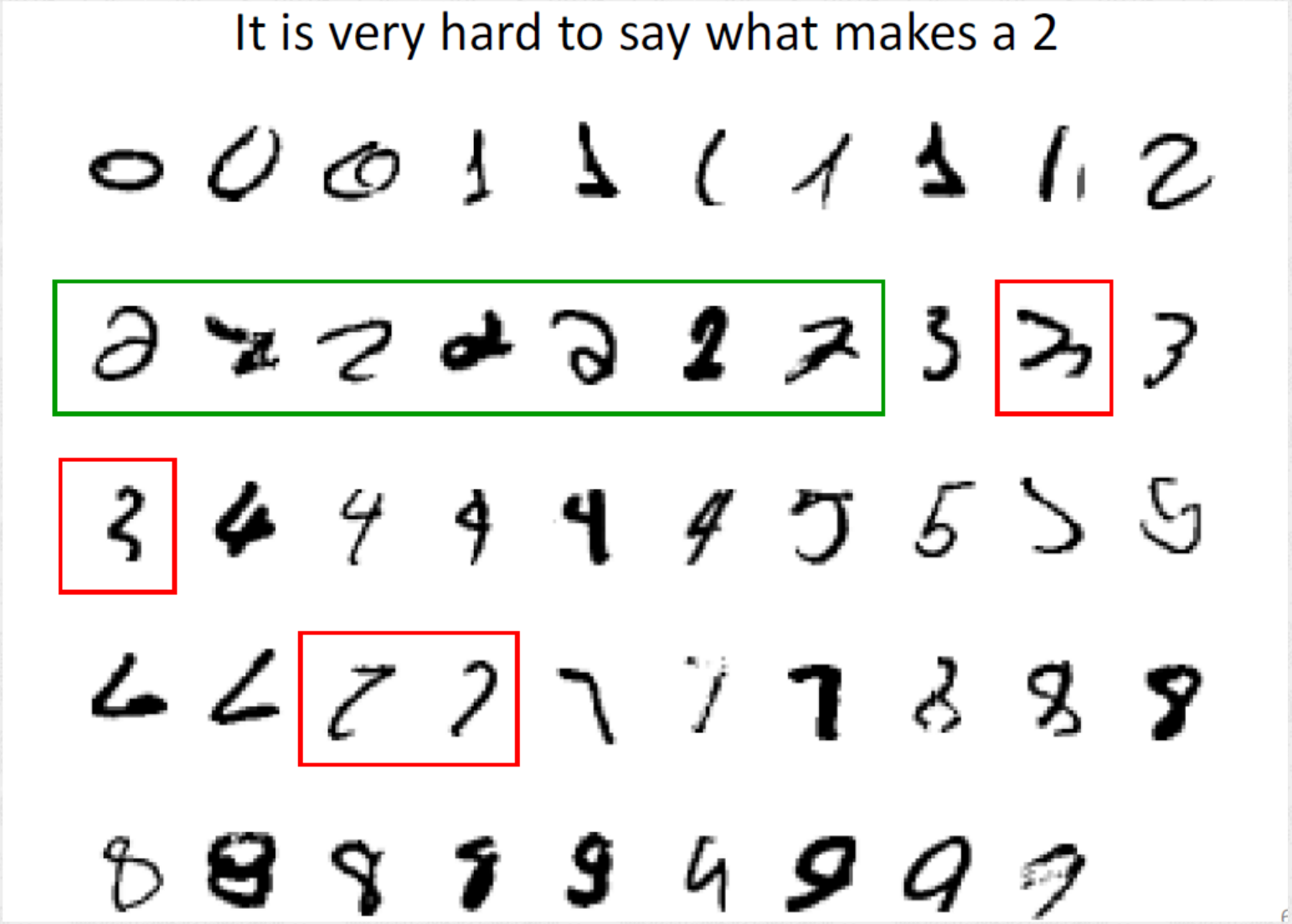
SoCal fire. Credit: Eddiem360, CC BY-SA 4.0 <https://creativecommo…

In the news…

Artificial intelligence driving autonomous vehicle development

30 January 2020

Classic Example of Machine Learning



Material Taken From IMLP and APM Slide by Geoffrey Hilton

More examples where ML is useful

- Recognizing patterns:
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns:
  - Generating images or motion sequences
- Recognizing anomalies:
  - Unusual credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant
- Prediction:
  - Future stock prices or currency exchange rates

Types of Machine Learning

Supervised (inductive) learning

– Given: training data + desired outputs (labels)

- **Unsupervised learning**

– Given: training data (without desired outputs)

- **Semi-supervised learning**

– Given: training data + a few desired outputs

- **Reinforcement learning**

– Rewards from sequence of actions

Supervised Learning

- Majority of the class is devoted to Supervised Learning

Two types of supervised learning

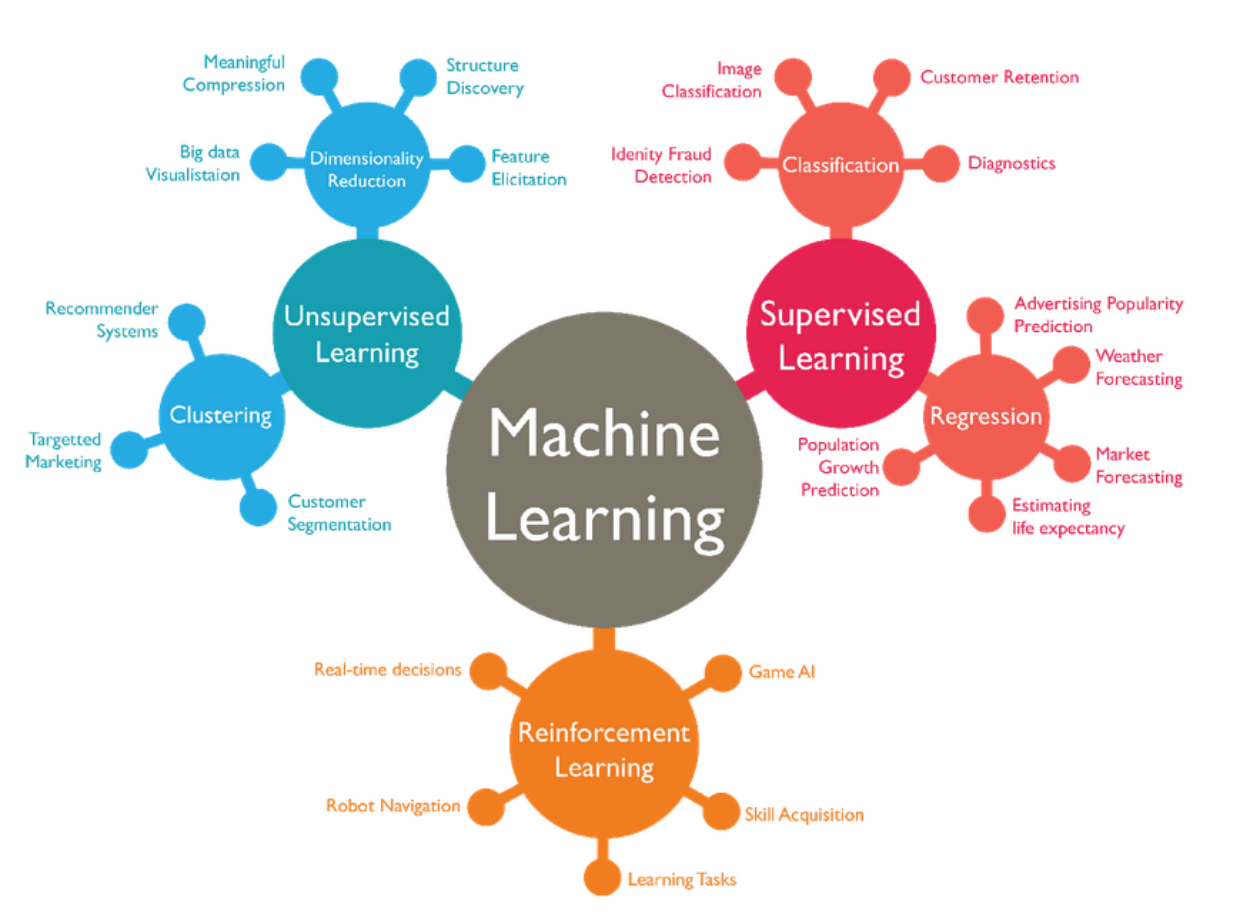
--- ***Classification***

The output (target) is the name of a class. Such as rose, lilly, daisies, carnations etc.

---- ***Regression***

The output is a continuous value such as house prices, size of tumor, time to failure of a part etc.

Types of Machine Learning



Material Taken From IMLP and APM

ML in a Nutshell

- Tens of thousands of machine learning algorithms
  - Hundreds new every year

- Every ML algorithm has three components:
  - **Representation**
  - **Optimization**
  - **Evaluation**

Different names for Machine Learning

- Predictive Modeling
- Machine Learning
- Pattern Recognition
- Data Mining
- Predictive Analytics
- (Artificial Intelligence)
- Knowledge Discovery
- Statistical Learning

We define Predictive Modeling as

“The process of developing a mathematical model that provides accurate prediction”

Questions one would like to decide include

- How many copies of books will we sell?
- Should I switch my phone service?
- How much will my house sell for?
- Does the patient have a specific disease?
- Is the mail a spam?
- Should I sell this stock now?

Why do Predictive Models Fail?

- Inadequate Pre-processing of Data
- Inadequate Validation of Data
- Unjustified Extrapolation (applying model to data that it very dissimilar to what it has seen)
- Overfitting (under generalization)

How complex can we make the model?

- Model details are not as important as accuracy of prediction Zillow.com

(we need good prediction of house prices.. we don't really care how it does it)

- Models can be therefore very complex (may be simplified and yield same output, but we may not be interested in simplification)

Understanding Data

- These days LOTS of data can be collected and lots are available.

- Data MUST be relevant
  (example, if a large number of patients that took a certain medicine to treat nausea as also had leukemia model may conclude that leukemia is side effect of nausea medicine, the expert know that people who already had leukemia took the medicine to nausea).

- Models are NOT SUBSTITUTE for EXPERT intuition but they support the experts in prediction.

- On the other hand, spam filters do not need experts even if the model may not be 100% accurate.

Understanding Task

- What question(s) am I trying to answer? Do I think the data collected can answer that question?

- What is the best way to phrase my question(s) as a machine learning problem?

- Have I collected enough data to represent the problem I want to solve?

- What features of the data did I extract, and will these enable the right predictions?

- How will I measure success in my application?

- How will the machine learning solution interact with other parts of my research or business product?

TERMINOLOGY

- Data point – Sample – Observation
- Features/Predictors
- Training Set
- Test Set
- Validation Set
- Response /Target
- Outcome
- Continuous
- Categorical Data
- Discrete Data
- Model Building / model training/model tuning/parameter estimation

Large Scale Machine Learning in Practice

- When you look at a complex website like Facebook, Amazon, or Netflix, it is
- Very likely that every part of the site contains multiple machine learning models.

Identify the types of Machine Learning

- Identifying the zip code from handwritten digits on an envelope
- Determining whether a tumor is benign based on a medical image
- Detecting fraudulent activity in credit card transactions
- Identifying topics in a set of blog posts
- Segmenting customers into groups with similar preferences
- Detecting abnormal access patterns to a website

End Session

TOOLS USED IN MACHINE LEARNING

1. Python
2. Scikit-learn
3. Pandas
4. NumPy
5. SciPy
6. matplotlib

Python

- It combines the power of general-purpose programming languages with the ease of use of domain-specific scripting languages like MATLAB or R.
- Python has libraries for data loading, visualization, statistics, natural language processing, image processing, and more.
- This vast toolbox provides data scientists with a large array of general- and special-purpose functionality.
- One of the main advantages of using Python is the ability to interact directly with the code, using a terminal other tools like the Jupyter Notebook, which we'll look at shortly.

Scikit-Learn

- Open Source developed in 2007 at Google, Later modified by Iniria in 2010
- Written in Python and Cython
- Provides libraries for machine learning algorithms
- Uses NumPy, SciPy

Top level documentation be found at:
https://scikit-learn.org/stable/index.html
User Guide can be found at:
https://scikit-learn.org/stable/user_guide.html
API Specification can be found at:
https://scikit-learn.org/stable/modules/classes.html

Scikit Learn Class and Methods

sklearn.linear_model.LogisticRegression

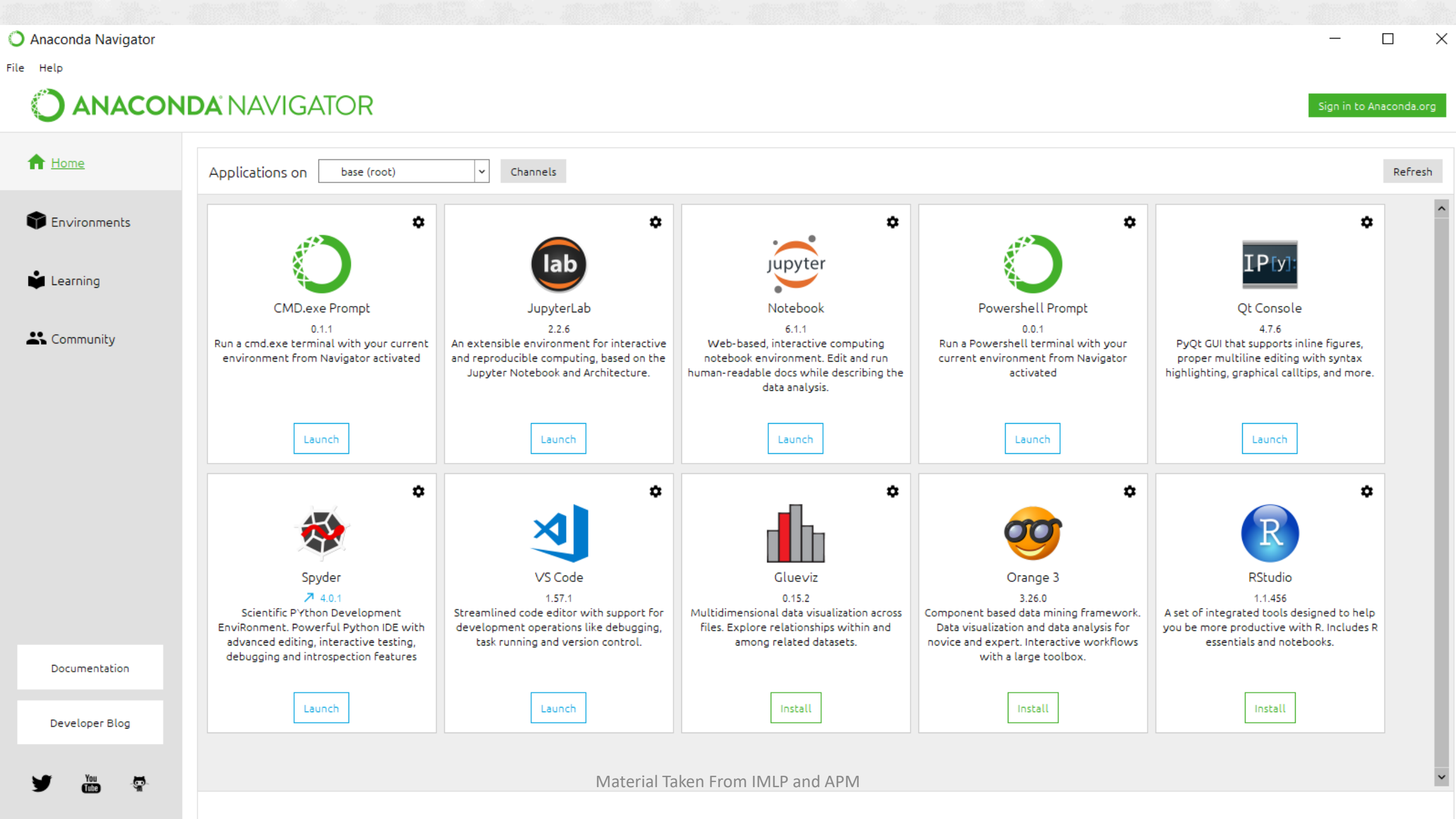
class sklearn.linear_model.**LogisticRegression**(*penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*) [source]

Methods

decision_function(X)	Predict confidence scores for samples.
densify()	Convert coefficient matrix to dense array format.
fit(X, y[, sample_weight])	Fit the model according to the given training data.
get_params([deep])	Get parameters for this estimator.
predict(X)	Predict class labels for samples in X.
predict_log_proba(X)	Predict logarithm of probability estimates.
predict_proba(X)	Probability estimates.
score(X, y[, sample_weight])	Return the mean accuracy on the given test data and labels.
set_params(**params)	Set the parameters of this estimator.
sparsify()	Convert coefficient matrix to sparse format.

Anaconda

- Is a development environment with simplified package manager
- Anaconda distribution comes with over 250 packages automatically installed
- Over 7,500 additional open-source packages can be installed from https://www.anaconda.com/products/individual
- It also includes a GUI, **Anaconda Navigator**,[12] as a graphical alternative to the command line interface (CLI).
- Automatically includes the packages we need numpPy, sciPy, scikit-learn, matplotlib and pandas



Material Taken From IMLP and APM

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\skanchi

...s\MachineLearning\PossibleResearchAreasGraphsAlgorithmsML\CapsGNN-master\CapsGNN-master\src\utils.py

temp.py | CNN_SandPaper.py | copy_of_midas_python_worshop_2020_FrFeng.py | awesome.py | main.py | utils.py

```
"""Data reading and printing utils."""

from texttable import Texttable

def tab_printer(args):
    """
    Function to print the logs in a nice tabular format.
    :param args: Parameters used for the model.
    """
    args = vars(args)
    keys = sorted(args.keys())
    t = Texttable()
    t.add_rows([["Parameter", "Value"]])
    t.add_rows([[k.replace("_", " ").capitalize(), args[k]] for k in keys])
    print(t.draw())

def create_numeric_mapping(node_properties):
    """
    Create node feature map.
    :param node_properties: List of features sorted.
    :return : Feature numeric map.
    """
    return {value:i for i, value in enumerate(node_properties)}

```

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

Help | Variable explorer | Plots | Files

Console 1/A

```
Python 3.7.6 (default, Jan  8 2020, 20:23:39) [
1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for mo
information.

IPython 7.18.1 -- An enhanced Interactive Pytho

In [1]:
```

Material Taken From IMLP and APM

NumPy

- NumPy is one of the fundamental packages for scientific computing in Python.
- It contains functionality for multidimensional arrays, high-level mathematical functions
- Includes linear algebra operations and the Fourier transform etc.
- In scikit-learn, the NumPy array is the fundamental data structure. scikit-learn takes in data in the form of NumPy arrays.
- We will be using NumPy *a lot* in this book, and we will refer to objects of the NumPy ndarray class as “NumPy arrays” or just “arrays.”
- Documentation for NumPy can be found at:

https://numpy.org/doc/stable/reference/index.html

Sample NumPy Code

01-introduction.ipynb

SciPy

- SciPy library is one of the core packages that make up the SciPy stack
- It provides many user-friendly and efficient numerical routines, such as routines for numerical integration, interpolation, optimization, linear algebra, and statistics
- scikit-learn draws from SciPy's collection of functions for implementing its algorithms.
- The most important part of SciPy for us is scipy.sparse: this provides *sparse matrices*, which are another representation that is used for data in scikit-learn.

https://docs.scipy.org/doc/scipy/reference/

- Sparse matrices are used whenever we want to store a 2D array that contains mostly zeros:

Code from the textbook

Authors code is available at:

https://github.com/amueller/introduction_to_ml_with_python.

Sample SciPy Code

Can be seen in

01-introduction.ipynb from the author

matplotlib

- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. The documents are available at:

https://matplotlib.org/stable/api/index.html

is available at:

pandas

- **pandas** is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language

- 10 minutes to Pandas

https://pandas.pydata.org/docs/user_guide/index.html

- Sample Code is available at:

01-introduction.ipynb from the author's code

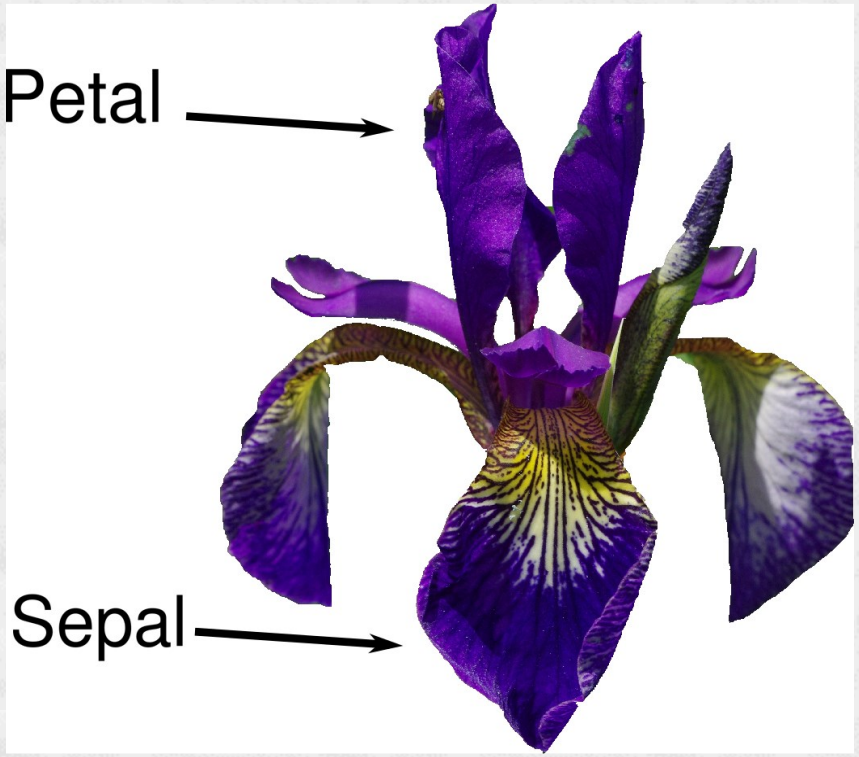
mglearn

- Author's library of modules

- We will see the usage of `mglearn` in examples

Sample Machine Learning Algorithm

We want to predict the type of iris flower, given details of sepal and petal measurements. The possible types of iris are *setosa*, *versicolor* and *verginica*. This is a multiclass classification problem.



Step 1: Meet the data

- The data we will use for this example is the Iris dataset, a classical dataset in machine learning and statistics.
- It is included in scikit-learn in the dataset module. We can load it by calling the load_iris function and see its details.

01-introduction.ipynb contains meet the data section

Step 2: Training and Testing data

- We want to build a machine learning model from this data that can predict the species of iris for a new set of measurements.
- Unfortunately, we cannot use the data we used to build the model to evaluate it.
- scikit-learn contains a function that shuffles the dataset and splits it for you: the train_test_split function. The split could be 75%:25%
- `P` (some books use X) used for the predictor matrix (all of the rows of data except the target column) and `y` is used for vector of target values.

01-introduction.ipynb contains Training and Testing Split

Step 3: Viewing the training data

- We need to make sure the training data is NOT all of the same type of target. (biased data)
- We have sufficient data
- Correlation issues. Etc.
- Graphical examination of data is useful when we have small data sets.
- Mathematical studies of data will be performed on large data sets

Training and Testing data can be viewed at

01-introduction.ipynb

Step 4: Making your model

- Here we will use a *k*-nearest neighbors classifier, which is easy to understand.
- Building this model only consists of storing the training set.
- To make a prediction for a new data point, the algorithm finds the point in the training set that is closest to the new point.

Then it assigns the label of this training point to the new data point.

- Build your model section can be seen at the following code from author

01-introduction.ipynb

Step 5: Making Predictions

Example of making predictions

01-introduction.ipynb from the author's code

- How reliable is the prediction? Should we trust it?
- We need to know how accurate the predictions are.
- We use test data for this.

Step 6: Evaluating the Model

Evaluating the Model using metrics can be sampled at

01-introduction.ipynb