CME 466: Design of a Digital System

Winter 2023

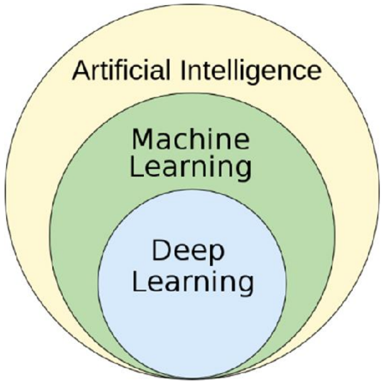
Introduction to Machine Learning

KHAN A. WAHID, PhD, PEng, SMIEEE

Professor
Electrical and Computer Engineering
University of Saskatchewan
Email: khan.wahid@usask.ca

1. What is Machine Learning (ML)?

- Definition from Wikipedia[1]: Machine Learning (ML) is a field of inquiry devoted to understanding and building methods that "learn" – that is, methods that leverage data to improve performance on some set of tasks.



- It is seen as a part of Artificial Intelligence (AI)

- In general, a learning problem considers a set of *n* samples of data and then tries to predict properties of unknown data.

- The algorithm is normally evaluated by splitting the data set into two types.

1.1 Two sets of data:

- **Training data:** sample data based on which an ML algorithm builds a model in order to make predictions or decisions

- **Testing data:** unknown data on which the "learned" model will be applied (or tested) to do some specific tasks and preferably improve performance

1.2 History of ML:

[1] https://en.wikipedia.org/wiki/Machine_learning

- The term Machine Learning was first used in 1959 by Arthur Samuel, an IBM employee that would give computers the ability to learn without being explicitly programmed

- Until the late 1970s, ML used to be a part of AI; however, it then branched off to evolve on its own.

- Tom Mitchell[2] provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: *A computer program is said to have learnt from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E*

1.3 Application of ML:

- Modern-day machine learning has two objectives:
  - To classify data based on models which have been developed
  - To make predictions for future outcomes based on these models

- Machine learning is responsible for many significant advancements in technology[3]. Listed below are some common ways we are using ML:

  - **Analyzing Data:** Computer aided diagnosis, Medical imaging, Health informatics, Agriculture, Smart grid
  - **Real-Time Analytics:** Promoting customer experience, Computer vision, Robotics, Autonomous vehicles
  - **Fraud Detection:** Detecting pattern changes, Biometrics, Face recognition, Safeguarding internet
  - **Product Recommendations:** Customer personalization, Optimize productivity, Dynamic pricing based on a need or demand

[2] Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 978-0-07-042807-2
[3] https://www.dataversity.net/a-brief-history-of-machine-learning/

- **Learning Management Systems:** Decision-making programs, IoT analytics
- **Natural Language Processing:** Speaking with humans, Speech recognition, Robotics

1.4 "Features" in ML:

- Features are measurable quantities or characteristics obtained from the patterns based on which the classification task takes place.

- Features are usually numeric values, but structural features such as strings, graphs, or shapes are also used

- In computer vision, features are standard deviation or mean of pixels, histograms, edges or objects.

- In character recognition, features may include counting the number of black pixels along horizontal and vertical directions, number of internal holes, and many others.

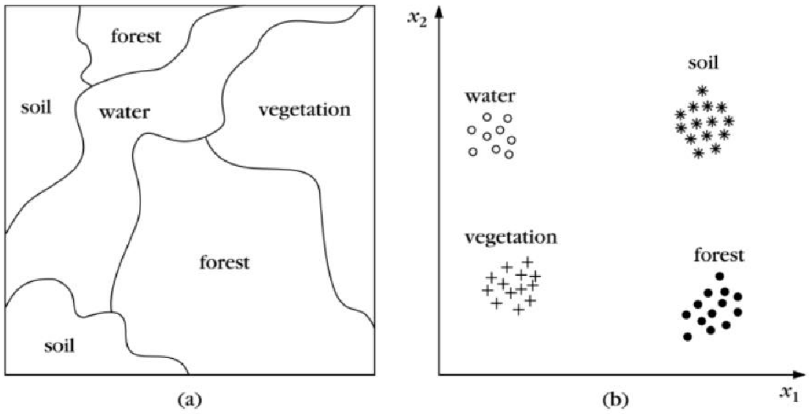
1.5 Types of ML:

- **Supervised**[4]: Patterns with known attributes (or classes) are used for training. This can be either:

  - **Classification:** Here, samples belong to two or more classes and we want to learn from already labeled data how to predict the class of unlabeled data. An example of a classification problem would be handwritten digit recognition, in which the aim is to assign each input vector to one of a finite number of discrete categories.

[4] https://scikit-learn.org/stable/tutorial/basic/tutorial.html#machine-learning-the-problem-setting

Satellite image



(a) An illustration of various types of ground cover and (b) clustering of the respective features for multispectral imaging using two bands.

- **Regression:** Here, prediction is made as a function of input data. The desired output often consists of one or more continuous variables. An example of a regression problem would be the prediction of the height of a man/woman as a function of their age and race.

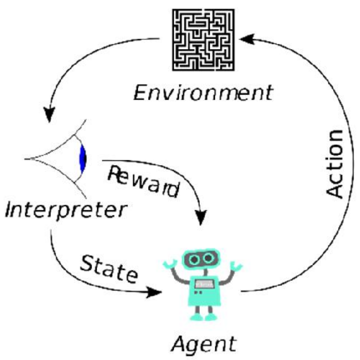
- **Unsupervised**: The number of classes/groups is unknown and no training patterns or data sets are available. This may be as follows:

  - **Clustering**: The goal in such problems may be to discover groups of similar characteristics or features within the data. There are many applications. For example, to classify antibiotics according to their antibacterial activity, or to group the shopping items available on the web into a set of unique products in online shopping.

- **Density estimation**: to determine the probability distribution of data within the input space. For example, to find the probability of women having diabetes in a certain ethnic group and age group.

- **Semi-supervised**: A mixed type of patterns is available. For some of them, their corresponding class is known and for the rest is not. Methods from both supervised and unsupervised learning may be applied here.

- **Reinforcement**: In Reinforcement leaning, software programs (a.k.a intelligent agent[5]) take actions autonomously in order to improve the performance with learning in a dynamic situation. It differs from supervised learning in not needing labelled input/output pairs to be presented. A typical example is autonomous vehicles to operate on a street in real-time, or in learning to play a game against a human opponent.



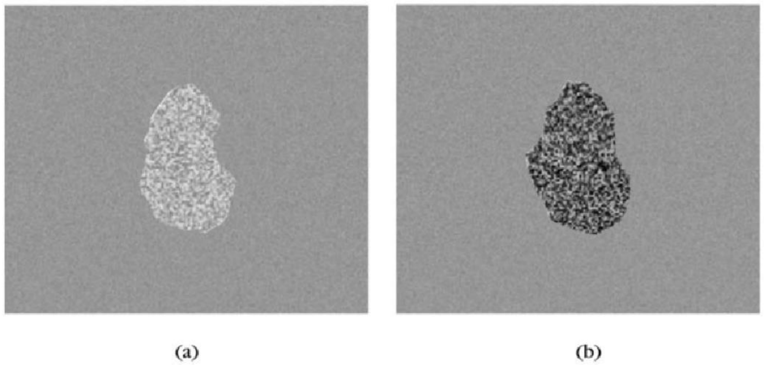
  The typical framing of a Reinforcement Learning (RL) scenario[6]: an agent takes actions in an environment, which is interpreted into a reward and a representation of the state, which are fed back into the agent.

1.6 Some simple examples of ML:

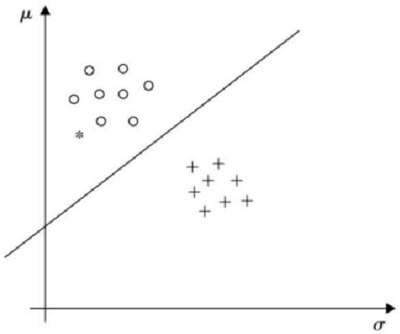
- **Classification in Medical Imaging[7]:** Let us look at the two images below, each having a distinct region inside it. These two regions are visually different. Image (a) shows benign lesion (class A) and image (b) shows malignant or

[5] https://www.javatpoint.com/agents-in-ai
[6] https://en.wikipedia.org/wiki/Reinforcement_learning
[7] Pattern Recognition (4th Edition), by Sergios Theodoridis and Konstantinos Koutroumbas, 2008

cancer lesion (class B). We assume that we have access to an image database with a number of patterns similar to these two.



Examples of image regions corresponding to (a) class A and (b) class B.



Plot of the mean value versus the standard deviation for a number of different images originating from class A (○) and class B (+). In this case, a straight line separates the two classes.

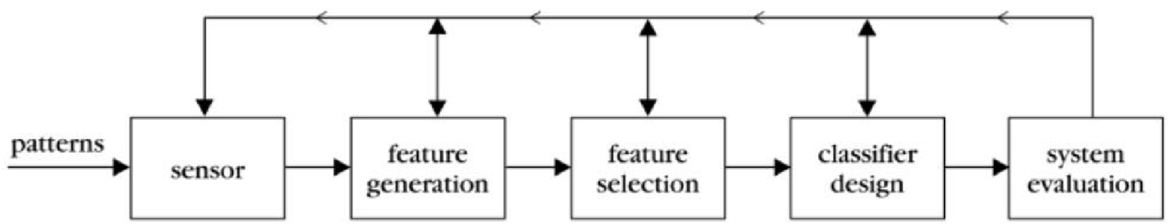
The first step is to identify features (or measureable quantities) that mark these two regions distinct from each other. If we plot the mean intensity (μ) of each region versus the corresponding standard deviation (σ), we can clearly see that a straight line would be a good candidate for separating these two classes.

Therefore, the straight line in known as the ***decision*** line that constitutes a ***classifier*** whose role is to divide ***features*** into either class A or class B.

Now, if we are given a new image (class is unknown), we can simply find the mean and standard deviation, and plot the point (shown as *). It is sensible to assume that this new unknown image belongs to class A (not class B).

Questions: This is just a prediction, and does not necessarily mean the decision is correct. If it is not correct, surely a misclassification has occurred. Therefore, some important questions to ask:

- Why is it not correct?
- Is the database large enough?
- How are the features generated?
- Are mean and standard deviation the best features? What other features can be used?
- How can design a better or more optimum classifier? (here a straight line was generated; but in practice, the pattern may be more complicated)
- How can we assess the performance of the classifier (which one is the best)? Leads to evaluation stage



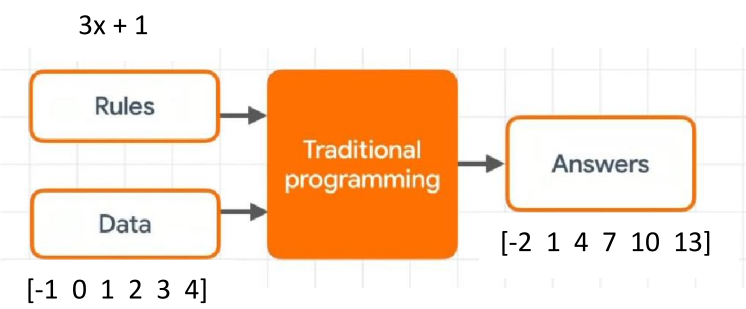
The basic stages involved in the design of a classification system. [8]

It is seen that each stage of the classification system is interrelated. As a result, depending on the result, one needs to go back to earlier stages to improve the overall performance.

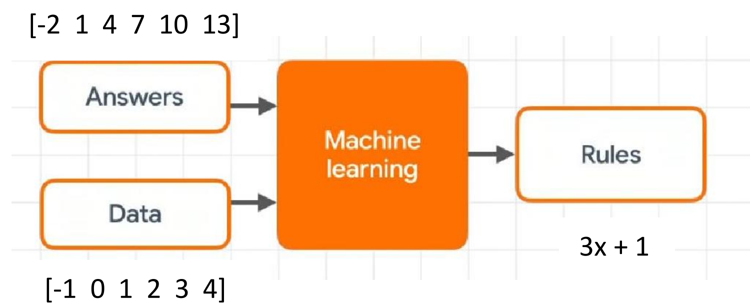
[8] Pattern Recognition (4th Edition), by Sergios Theodoridis and Konstantinos Koutroumbas, 2008

- **Ordinary least squares Linear Regression[9]:** Linear Regression fits a linear model to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

Let us look at the following example. In traditional programming (fig a), we have rules and data that are used to generate answers (or outputs).



(a)



(b)

However, in machine learning, instead of telling the systems the rules, a lot of data and answers (these data pairs are called ***labelling***) are fed into it. The system will then figure out the rules (fig b).

Let us see it firsthand using scikit-learn.

Virtual environment in lab computers: In your lab computers, a virtual environment along with some necessary packages is already setup. At first, you will need to switch to sandbox so the correct version of python is used. Here is the command (from terminal) that you need to run:

```
source /opt/CME466/bin/activate
```

[9] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

You can then launch **Jupyter Notebook** in the virtualenv (/opt/CME466) from the terminal using the commands:

```
jupyter notebook
```

Here are some outputs to show that you've correctly switched to the sandbox:

```
[chandler@chandler-vm6 opt]$ which python3
/usr/bin/python3
[chandler@chandler-vm6 opt]$ source /opt/CME466/bin/activate
(CME466) [chandler@chandler-vm6 opt]$ which python3
/opt/CME466/bin/python3
```

Setting up scikit-learn[10]: We will be using scikit-learn as the main library for all computer implementation and programming. Make sure you have the library and all prerequisites. Also, please review the Python prerequisites (http://www.scipy-lectures.org).

If you are using Anaconda, these libraries should be already installed. For other distributions, you can manually install them using:

```
pip install numpy matplotlib pandas scipy scikit-learn scikit-image
```

Launch Jupyter Notebook in the virtualenv and run the following codes.

```
# need to import numpy to handle data array
```

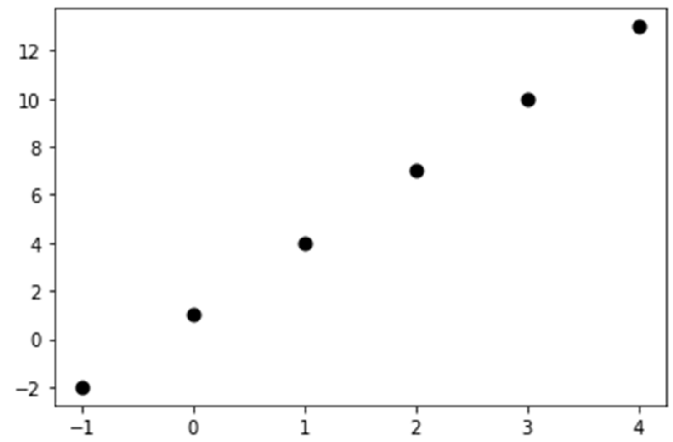
[10] https://scikit-learn.org/stable/

```
import numpy as np
import matplotlib.pyplot as plt
# import linear regression class from scikit-learn
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
print(lin_reg)
# define the following data inputs
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0])
ys = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0])
# need to reshape as column
xs_t = xs.reshape(-1,1)
ys_t = ys.reshape(-1,1)
print(xs)
print(xs_t)
```

```
[-1.  0.  1.  2.  3.  4.]
[[-1.]
 [ 0.]
 [ 1.]
 [ 2.]
 [ 3.]
 [ 4.]]
```

```
plt.scatter(xs, ys, color='black')
# train the classifier with our dataset
lin_reg.fit(xs_t,ys_t)
# Now let us test with a known data point
lin_reg.predict([[2]])
           array([[7.]])

# Now let us test with an unknown data point
lin_reg.predict([[15]])
           array([[46.]])
reg.coef_
           array([[3.]])
reg.intercept_
           array([1.])
reg.score(xs_t, ys_t)
           1.0
```



As you can now that the model is trained with the set of data to predict any new unknown data point. The best fit model is a straight line with a coefficient of 3 and intercept of 1. The coefficient of determination (R^2) of the prediction is 1, which is the best possible score (since it was a simple problem and we could easily see the model in bare eyes; however, complex models will have value less than 1)[11]

This is the power machine learning.

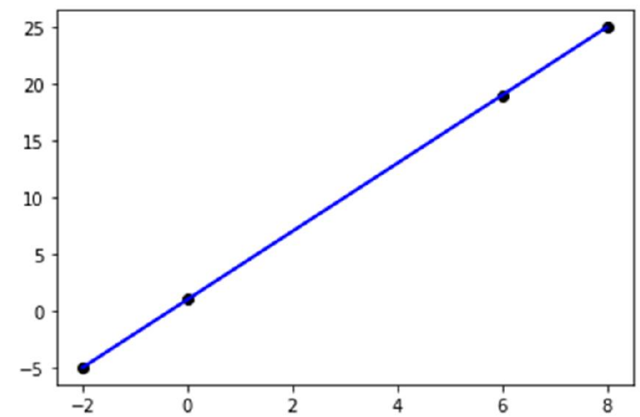
[11] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

```
# test on unseen data
xt = np.array([-2.0, 0.0, 6.0, 8.0])
xt_t = xt.reshape(-1,1)


yt=lin_reg.predict(xt_t)

print(yt)


plt.scatter(xt, yt, color='black')
plt.plot(xt, yt, color='blue', linewidth=2)
```



1.7 Online resources

- Theodoridis, Pikrakis, Koutroumbas, Cavouras: Introduction to Pattern Recognition: A Matlab Approach, 2010. https://booksite.elsevier.com/9780123744869/?ISBN=9780123744869
- scikit-learn: Machine Learning in Python https://scikit-learn.org/stable/index.html
- An introduction to machine learning with scikit-learn https://scikit-learn.org/stable/tutorial/basic/tutorial.html#machine-learning-the-problem-setting