TA 1

Arun Kumar Rajasekaran

Outline

~ 13 weeks of Lectures and TA

~ Independent research project

Student interests:

1. Name, where are you from?
2. Subject interests (in priority order)
3. Coding interests anyone? (what language?)
4. Coding or ML experience before ? (that you like to share)
5. Hobbies?

Python

https://www.python.org/downloads/

Jupyter

https://jupyter.org/

Google colab

https://colab.research.google.com/?utm_source=scs-index

Why Python?

(Extensive libraries and frameworks)

NumPy: NumPy is a fundamental Python library for efficient numerical computations and array operations.

Scikit-learn: Scikit-learn is a comprehensive machine learning library that offers a wide range of tools for various tasks, including classification, regression, clustering, and more.

Pandas: Pandas is a powerful library for data analysis and manipulation, providing intuitive data structures like DataFrames and Series.

TensorFlow: TensorFlow is a cutting-edge deep learning library known for its distributed computing capabilities and robust ecosystem.

Theano: Theano is a Python library designed for fast numerical computation, particularly useful for training deep learning models.

Keras: Keras is an easy-to-use deep learning API that acts as an interface for TensorFlow, Theano, or Microsoft Cognitive Toolkit (CNTK), simplifying the creation and training of neural networks.

PyTorch: PyTorch is a dynamic deep learning library with a flexible computation graph, making it ideal for developing and training complex neural networks.

Why Python?

Easy-to-read syntax

Cross-platform compatibility

Scaling and Performance

Community Support and Documentation

(Applications: Youtube, Google, Spotify, Reddit, Instagram, ChatGPT etc)

Python performance?

Performance and Efficiency:

- **Python**: Python's performance has improved significantly with libraries like NumPy and JIT compilation techniques. Though not as fast as Java or C++, it provides an acceptable level of performance for most machine learning tasks.
- **R**: R can be slower than Python for certain operations, particularly when handling large datasets.
- **Java/C++**: Java and C++ are known for their high performance and efficiency, especially in computationally intensive tasks, but they often require more code to achieve the same functionality.

Coding standards

• Coding standards are guidelines for code style and documentation.

• They may be formal (IEEE) standards, or company specific standards.

• The aim is that everyone in the organization will be ableto read and work on the code.

• Coding standards cover a wide variety of areas:

– Program design

– Naming conventions

– Formatting conventions

– Documentation

– Use (or not) of language specific features

- Why bother with a coding standard?

  – Consistency between developers

  – Ease of maintenance and development

  – Readability, usability

- Example should make this obvious!

- No standard is perfect for every application.

  – If you deviate from the standard for any reason,

document it!

Coding style

• There are several examples of coding styles. Often they differ from company to company

• They typically have the following in common:

<u>– Names</u>

• Use full English descriptors

• Use mixed case to make names readable

• Use abbreviations sparingly and consistently

• Avoid long names

• Avid leading/trailing underscores

<u>– Documentation</u>

• Document the purpose of every variable

• Document why something is done, not just what

Coding style

– Accessors

• Use getX(), setX() functions on all class variables.

– Member function documentation

• What & why member function does what it does

• Parameters/return value

• How function modifies object

• Preconditions/postconditions

• Concurrency issues

• Restrictions

– Document why the code does things as well as what

it does.

Standards

• Standards rare documented agreements containing technical specifications or other precise criteria to be used consistently as guidelines, rules, or definitions of characteristics, to ensure that materials, products, processes and services are for for their purpose.

• International standards are supposed to contribute to making life simpler, and to increasing reliability and effectiveness of the goods and services we use.

• Standards represent best, or most appropriate, practice:
– They encapsulate historical knowledge often gained through trail and error.
– They preserve and codify organizational knowledge and memory
– They provide a framework for quality assurance.
– Ensure continuity over a project's lifecycle.

Standards

There are many industry standards governing all aspects of software development:

– Terminology

– Notation

– Requirements gathering

– Design

– Coding

– Documentation

– Human computer interaction

– Verification and validation

– Quality assurance

– Even ethics!

Who writes standards?

– ISO

International Organization for Standardization

– SAA

Standards Australia

– BSI

British Standards Institute

– ANSI

American National Standards Institute

– IEEE

Institute for Electronic and Electrical

Engineers

– And about 80 or so others!

• ISO 646 – 7-bit ASCII with national variants

• ISO 8859 – several 8-bit ASCII extensions:

– ISO 8859-1: West European languages (Latin-1)

– ISO 8859-2: East European languages (Latin-2)

– ISO 8859-5: Latin/Cyrillic

• ISO 6429 – ASCII control codes

• ISO 2382 – Information technology vocabulary

• ISO 8652 – the Ada programming language

• ISO 9899 – the C programming language

• ISO 9660 – CD-ROM volume and fie structure

• ISO 3166 – codes for the representation of names of counties:

– Defines a 2-letter, 3-letter and numeric code for every country.

– US/USA/840 = United States

– GB/GBR/826 = United Kingdom

• The 2-letter codes are well known as the internet top level domain names.
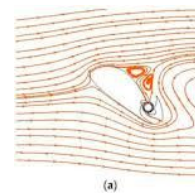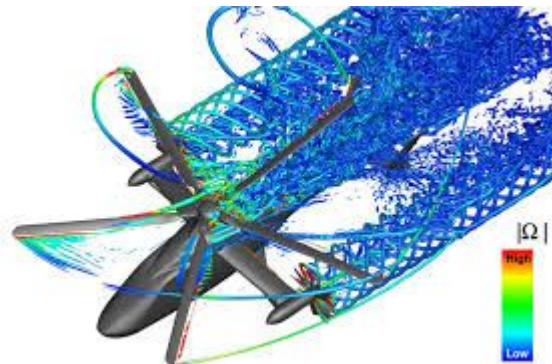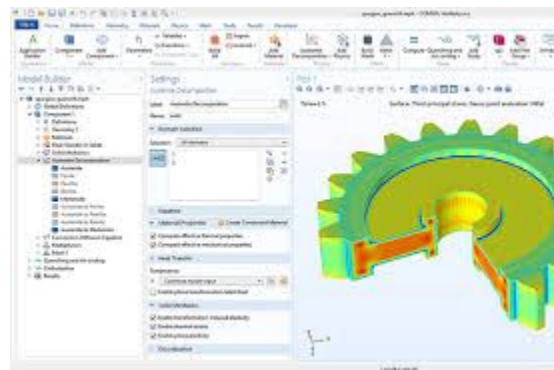
Why math?
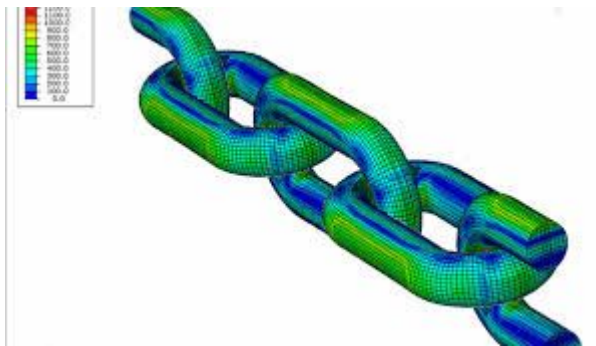
Vector calculus

Fourier Analysis

Complex Analysis

Numerical methods

Used in classical, quantum mechanics; fluid dynamics and signal processing

(a)

(b)
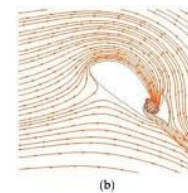
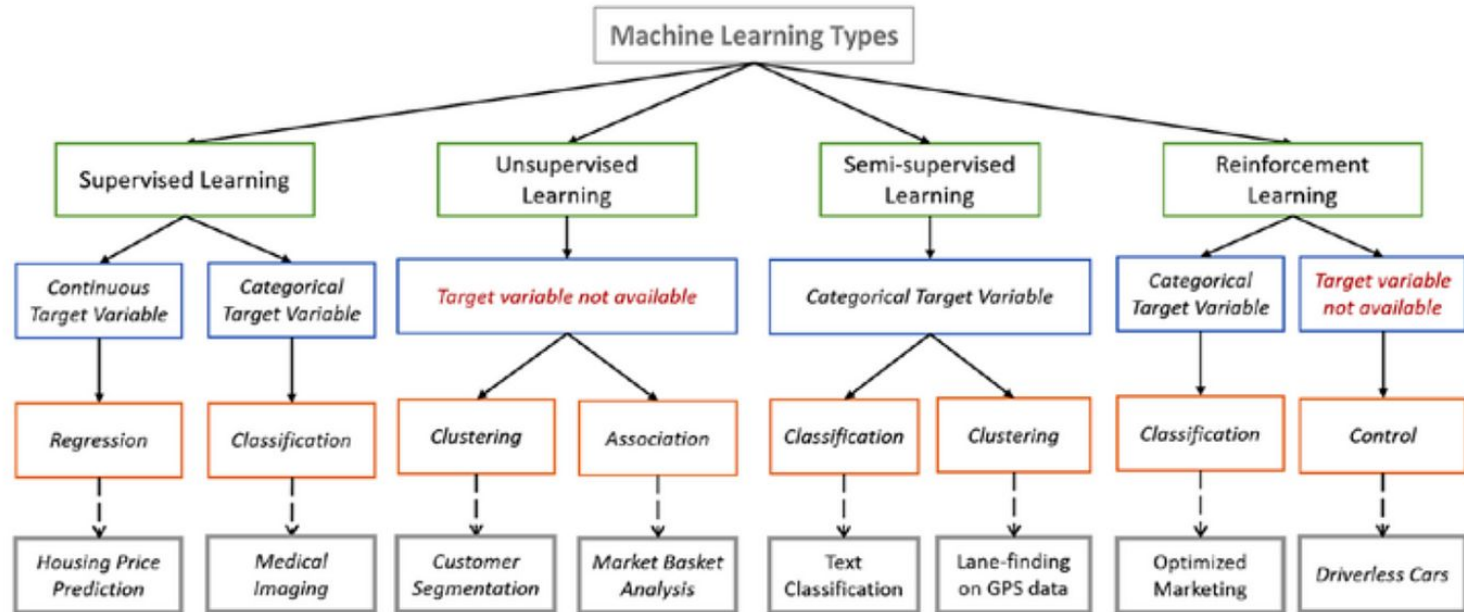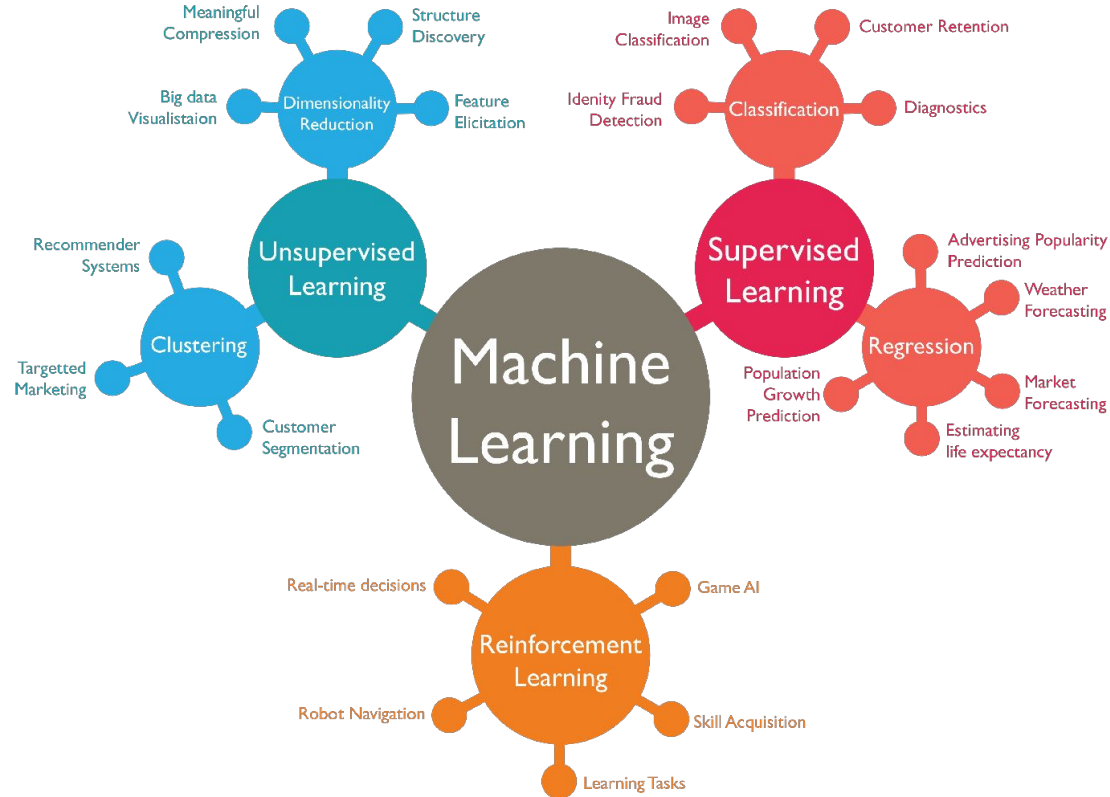Machine learning/ AI interests anyone?

Machine learning

Important ML milestones

1960S Decision trees	**1980** Backpropagation	**1998** CNN	**2010S** Deep Learning models	

1957 Perceptron	**1969** Multilayer perceptron	**1990S** SVM	**2001** Random Forest	**2017** Transformer models

ML types

ML applications
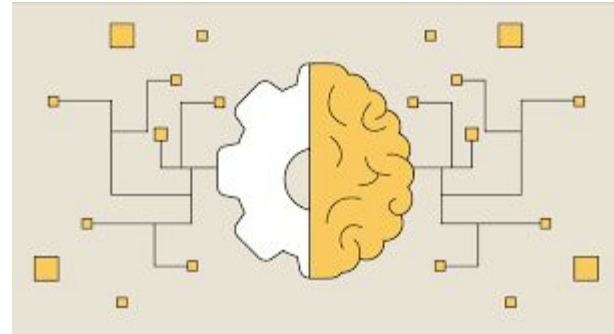
Whats' your favorite ML application area(s)?

Can you make a small list….

Some interesting applications

+ Chatbots
+ Fraud detection
+ Self driving cars
+ Stock/Finance
+ Translation/NLP
+ Healthcare/Climate research

A bit about course, timings, research project, post-course