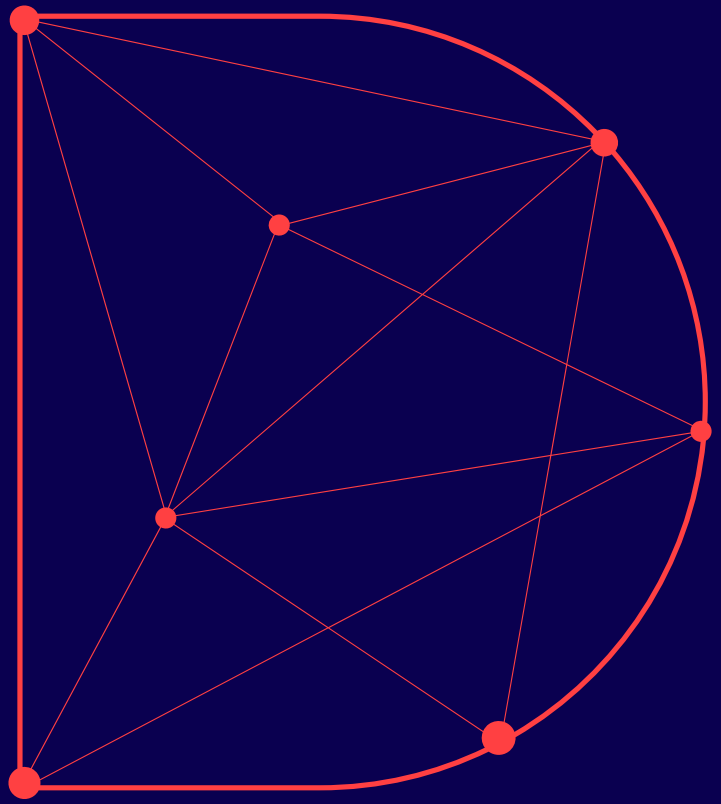
MODULE 4: Introduction to Machine Learning and Python

OBJECTIVES	✓ Understand the goals of Artificial Intelligence (AI) ✓ Define learning, machine learning, and understand the goal of learning ✓ Understand the need for machine learning, and the capabilities of machine learning ✓ Understand the difference between the learning algorithm, and the learned classifier ✓ Describe the process of training data generation ✓ Explain the process of devising a machine learning (ML) solution ✓ Understand the basics of artificial neural networks ✓ Differentiate between the different types of learning ✓ Understand the potential of ML learning in earth observation (EO) ✓ Understand how ML is used in image classification ✓ Understand the different approaches to feature extraction from EO data ✓ Understand the basics of the programming language Python and the application Jupyter Notebook ✓ Install Jupyter Notebook and practice basic coding ✓ Understand the basics of Sentinel Hub as an EO Processing Platform ✓ Simple spatial data collection on the field and analysis
METHODS	Live session, reading material, video's, links to resources, application exercises, quizzes & discussions
DURATION	6.5 hours for participants

SESSION			**DURATION**	**PARTICIPANTS…**
Online	1.0	Introduction to Machine Learning and Python	60 min.	✓ Get exposed to the basic concepts of ML for EO
	1.1	Introduction to Machine Learning	30 min.	✓ Learn about the basic concepts of artificial intelligence
	1.2	ML training data and web mapping	45 min.	✓ Learn about the practical aspects of the key steps in Machine Learning.
	1.3	Machine learning algorithms for image classification	45 min.	✓ Learn about the general concepts and algorithms for image classification.
	1.4	Python programming and Jupyter Notebook	90 min.	✓ Get an understanding of the basics of the programming language Python and Jupyter Notebook. ✓ Reflect on contents and share experience with peers. ✓ Get an understanding of the basics of Sentinel Hub as an EO Processing Platform.
	1.5	Fieldwork: image classification	120 min.	✓ Conduct a simple spatial data collection on the field ✓ Reflect on content and share experience with peers



Digital Transformation Center Rwanda

german cooperation
DEUTSCHE ZUSAMMENARBEIT

Implemented by
giz Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH

Digital Transformation Center Rwanda

FAIR FORWARD
Artificial Intelligence for all.

RSA
Rwanda Space Agency

Carnegie Mellon University Africa

GFA CONSULTING GROUP

OPEN MACHINE LEARNING FOR EARTH OBSERVATION (ML4EO)

MODULE 4: Introduction to Machine Learning & Python

FAIR FORWARD
Artificial Intelligence for all.

RSA
Rwanda Space Agency

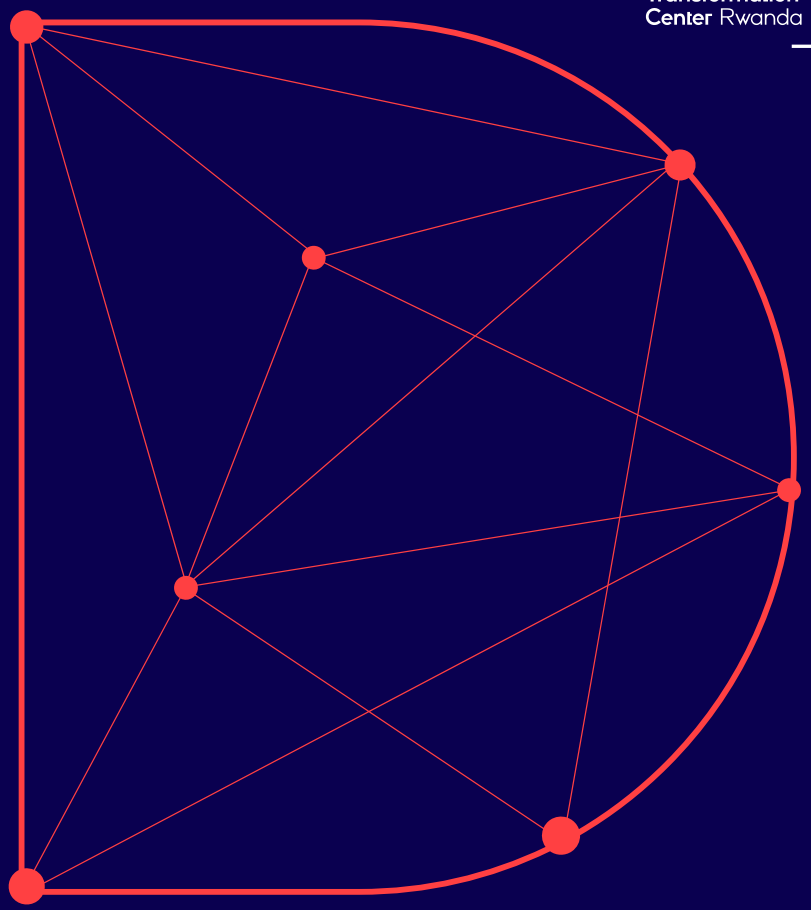
Carnegie Mellon University Africa

GFA CONSULTING GROUP

german cooperation
DEUTSCHE ZUSAMMENARBEIT

Implemented by
giz Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH

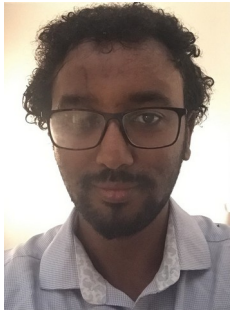
Digital Transformation Center Rwanda



Digital Transformation Center Rwanda

Before we start

- Trainers' team today



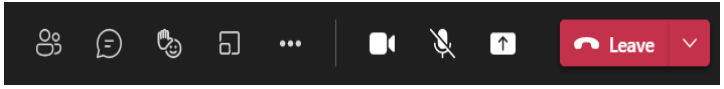
Abrham Gebreselasie
CMU Research Associate

Clarisse Goffard
Education Expert

- Communication:
  - Please mute your device if you are not speaking, and switch on the camera/video
  - Please post your questions in the chat and we will answer them on the spot, or during the Q&A session
- Material and presentation will be shared

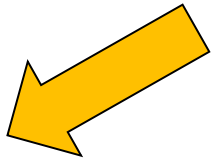
Joanne Schuiteman
Moodle Expert

Anselme Ndikuryyayo
Education Expert



- The presentation will be recorded, we assume your consent

Structure of today's session



- Course overview
- Review Module 3
- Introduction to Module 4
- Q&A
- Next steps / assignments
- Closure

Course overview – UPDATED to 10 Modules

The course is composed of **11 Modules**:

- Module 1: Introduction to GIS
- Module 2: Introduction to Remote Sensing
- Module 3: GIS data collection methods
- Module 4: Introduction to ML and Python
- Module 5: Data curation for ML
- Module 6: Predictive modelling using local RS data
- Module 7: ML workflows and best practice
- Module 8: Deploying remote sensing-based models
- Module 9: Business model generation and value proposition design
- Module 10: Project development

Modules 1-4: online + field work
- MS Teams live sessions
- Self-learning on Moodle

Modules 5-9: face-to-face
- Block 1: 15-16 April 2023 (M5)
- Block 2: 22-23 April 2023 (M6)
- Block 3: May 2023 (M7-M9)

Module 10: online & face-to-face
- Project development
- Pitch event

Review of Module 3 – FORUM

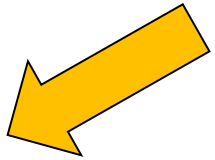
- 27 participants have uploaded their reports for the application exercise in the forum (well done !)
- 13 participants have not yet completed module 3 (keep going, you will soon be there !)
- Our technical experts are currently revising the reports posted:
  - 15 reports have been graded as complete and correct
  - 7 participants did not complete fully (something is missing or wrong)
  - 2 participants posted only comments, no screenshots
  - rest is under review.
- General Feedback: There was not sufficient number of points, lines and poligons included

VIRTUAL CONSULTATION on M3 application exercise – TOMORROW 2 pm CAT

?? QUESTIONS ??

Structure of the Live Session

- Course Overview
- Review Module 3
- Module 4: Introduction to Machine Learning and Python
    - Defining Machine Learning
    - Key Steps in Machine Learning
    - Image Classification
    - Applications of ML4EO
- Q&A
- Next steps / assignments
- Closure



What do you know about ML ?

We will collect your current ideas via Mentimeter.

Follow the steps:

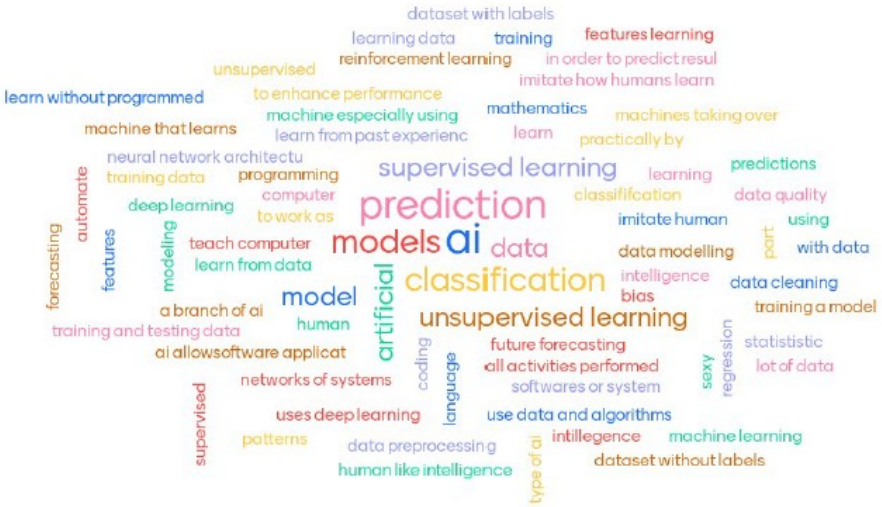
- Either: Click on the link in the chat

- Or: go to www.menti.com + submit the code (given in the chat)

- THEN participate by answering the questions

What do you know about ML? Give us some keywords!

Mentimeter



27

german cooperation DEUTSCHE ZUSAMMENARBEIT

Implemented by giz Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH

Digital Transformation Center Rwanda

FAIR FORWARD Artificial Intelligence for all.

RSA Rwanda Space Agency

Carnegie Mellon University Africa

GFA CONSULTING GROUP

Digital Transformation Center Rwanda

MODULE 4:

Introduction to Machine Learning and Python

What is Learning?

- Learning is a mechanism by which an agent improves its performance by observing the world
- Learning is the use of experience to gain expertise
- If the agent that is learning is a computer we call it machine learning
  - **Classifying spam email:**
    - Let's say you decided to write your own spam email detector program
    - When your program gets a new email it compares it to all emails that have been labelled spam in the past and if it matches one of these it classifies it as spam if it does not match any past spam email it classifies it as not-spam
    - Whenever the program fails to detect a spam email because it does not match one of the past emails you inform it that this email was spam. The program then filters out future emails that match the newly added spam email.
    - This is known as **learning by memorization**

Why and When do we need Machine Learning?

Tasks that are too hard to program

- Image recognition: this can be done easily by humans and animals but it is hard to write as a rule-based program
- Driving vehicles
- Speech Recognition

Why and When do we need Machine Learning?

Tasks that can't be handled by humans

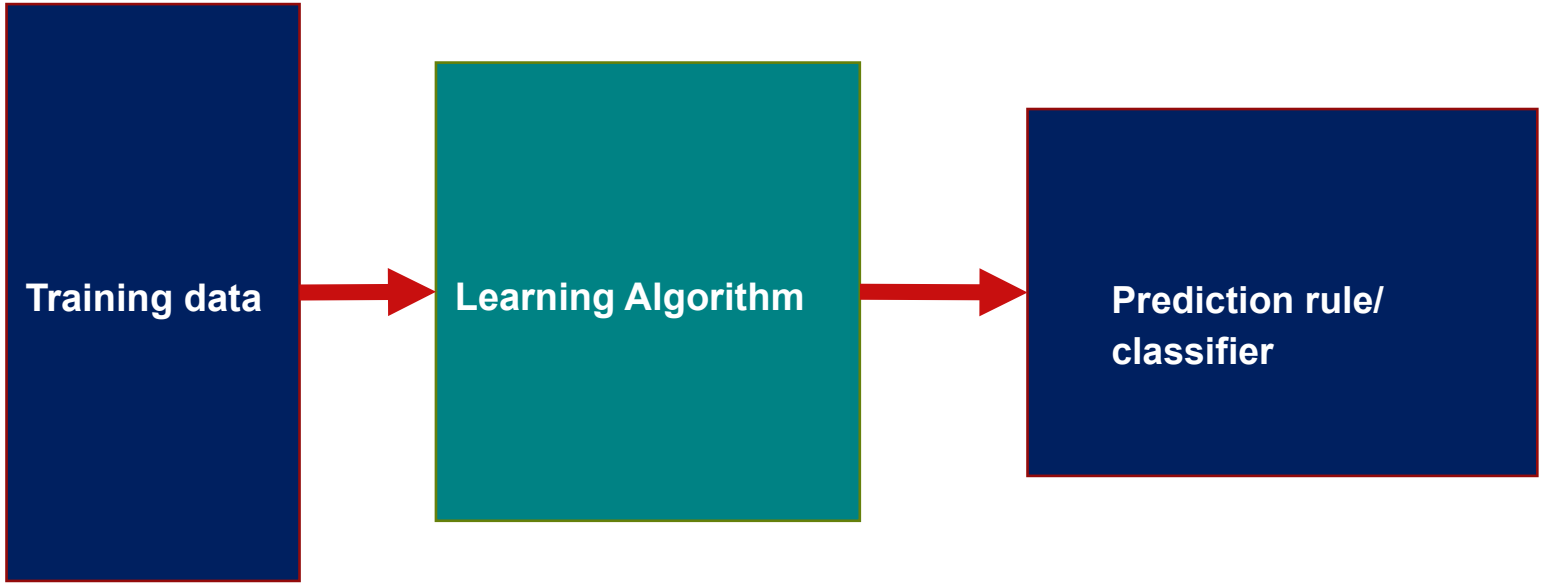
- Analyzing genomic data of billions of humans
- Determining which ads/products to show to which users when you have hundreds of millions to billions of users

Why and When do we need Machine Learning?

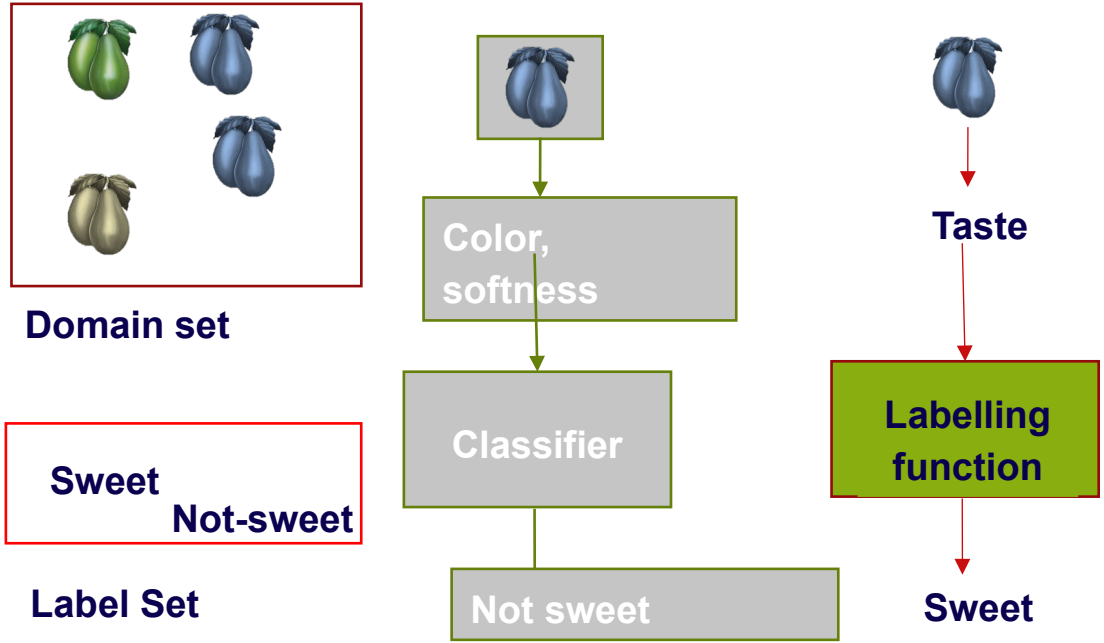
Tasks that require continuous adaptability to their input

- Detecting spam emails:
  - Spammers change their techniques everyday writing new programs to cope with this requires a team on standby
- Consumer interests change over time so using machine learning to recommend products to the customer is necessary
  - A person that loved action movies last year may now be interested in watching comedy
  - A user that loved buying video games online at age 16 may want to buy books at age 30

Learner (Learning Algorithm)



Inputs and Outputs of a Learner



Key Steps in ML

1. Identifying the domain set, the label set, and the feature selection
2. Data Collection
3. Preparing the Data
4. Choosing a Model
5. Training the Model
6. Hyperparameter Tuning
7. Evaluating the Model
8. Deployment

ML in Remote Sensing

1. Machine learning with remote sensing help
2. Improve predictions about the behavior of environmental systems,
3. improve the automation of data analysis,
4. lead to a better management of resources and the discovery of new insights from complex data sets

Image Classification

1. Image classification is the task of assigning a label or class to an entire image.

Applications of ML in Remote Sensing

Geology
- Mineral detection
- Cover homogeneity

Forestry
- Infected trees
- Status monitoring
- Forest clearing

Sea/ice/coastal
- Oil spills monitoring
- Water quality

Precision agriculture
- Crop stress location
- Crop productivity

Atmosphere
- Air quality, pollutants
- Global/local change

Land management
- Crop monitoring/phenology
- Land use/cover change

Defense
- Target detection
- Mine detection

Public safety
- Logistics & operations
- Fire risk, floods

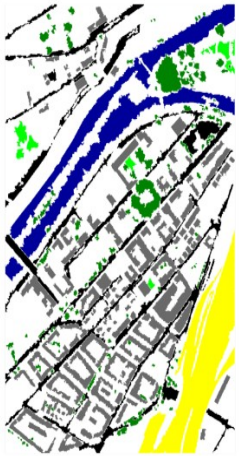
Regulation & Policy making
- Urban growth
- Settlements, population movements

Machine Learning Tasks in Remote Sensing

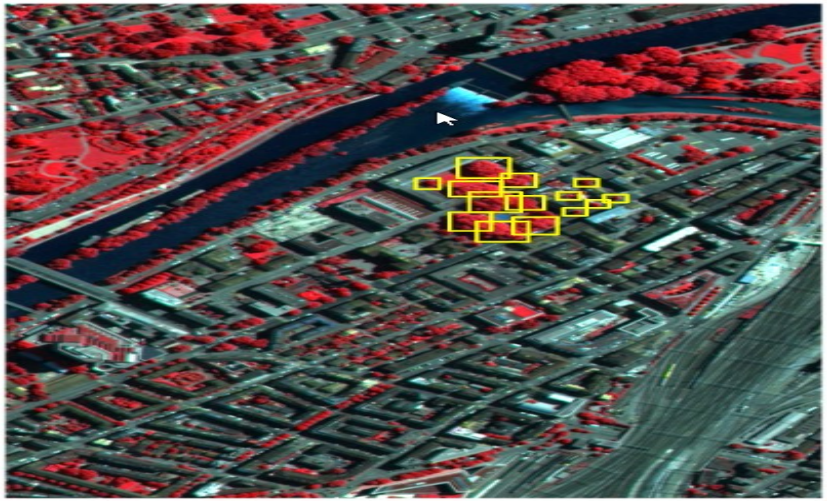
1. Generally, there are two approaches to applying ML to a remote sensing data.

2. pixel-based classification: the spectral band features of a single pixel are used to assign it a label.

3. Object-based: uses both spatial and spectral features to output classifications for regions in the input.



ID	Color	Label	ID	Color	Label
1		Roads	2		Buildings
3		Trees	4		Grass
5		Bare Soil	6		Water
7		Rails	8		Pools



General Steps to Process EO Data

1. Select best available image according to pre-defined thresholds

2. Select best features (channels, spatial) that describe the problem (classification, retrieval)

3. Remove noise and distortions due to clouds, acquisition (sun glint) or transmission (vertical stripes)

4. Use band operations to create band ratios, and indices through linear/non-linear combinations of existing bands

5. Pass the input (along with the extracted features) and use an ML algorithm to assign semantic classes to objects (pixels, patches, regions) in the scene

Q & A

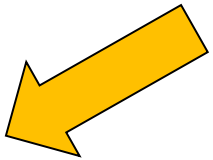
DO YOU HAVE ANY QUESTIONS?

Please raise your hand

or write your questions into the chat box!

Structure of today's session

- Course overview
- Review Module 3
- Introduction to Module 4
- Q&A
- Next steps / assignments
- Closure



Self-study part Moodle – Module 4

- To be completed by Sunday **26.03.2023**
  - Read all contents of all sessions
  - Watch all videos
  - answer all quiz questions after each session (3 attempts)
  - Perform the application exercise

Group Assignment on Moodle – Module 4

- To be completed by Sunday **26.03.2023**

Contribution to the Forum:

- Post your results of the application exercise and your answers to the questions in the forum
- Read the contributions of the other group members
- Answer any question you got in response to your own post

IMPORTANT ANNOUNCEMENT: Module 5 & Module 6

Presence-based block 1 has now been split into two presence sessions:

BLOCK 1: 15-16 April 2023 – Module 5 (formerly M5+M6, now merged)

BLOCK 2: 22-23 April 2023 – Module 6 (formerly M7)

Module 5 moodle TASKS → **to be completed before 14th April**:

- Read M5 READER on moodle (reader open on moodle by April 1st)

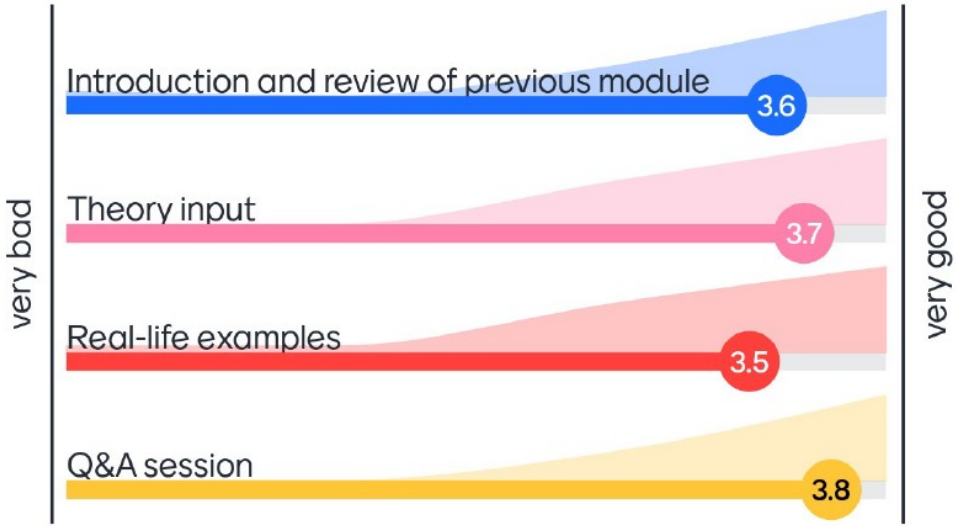
Evaluation of the Live Session

- Before we close… please do a short evaluation of our live session
- Either: Click on the link in the chat

- Or: go to www.menti.com + submit the code (given in the chat)
- THEN participate by answering the questions

Rate the content of todays' live session

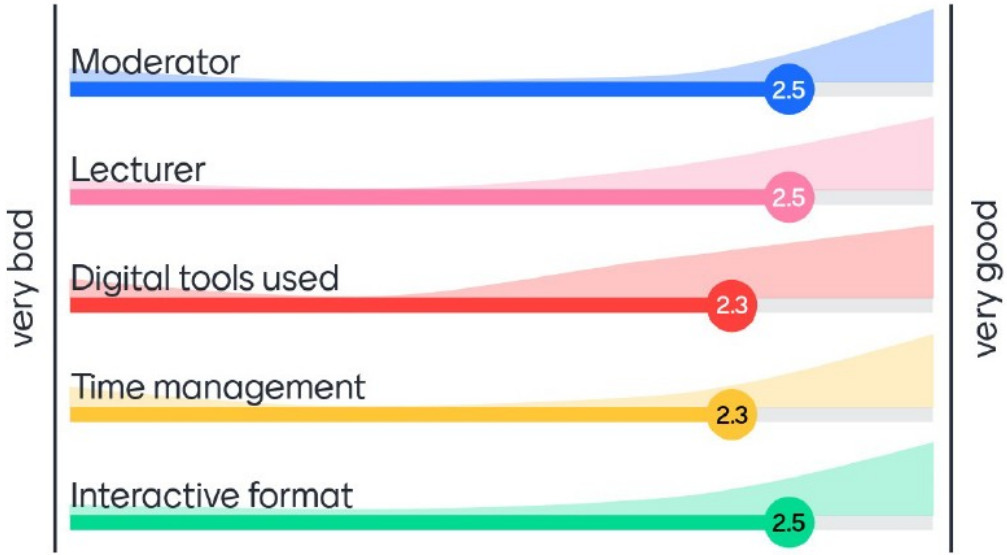
Mentimeter



24

Rate the quality of moderation

Mentimeter

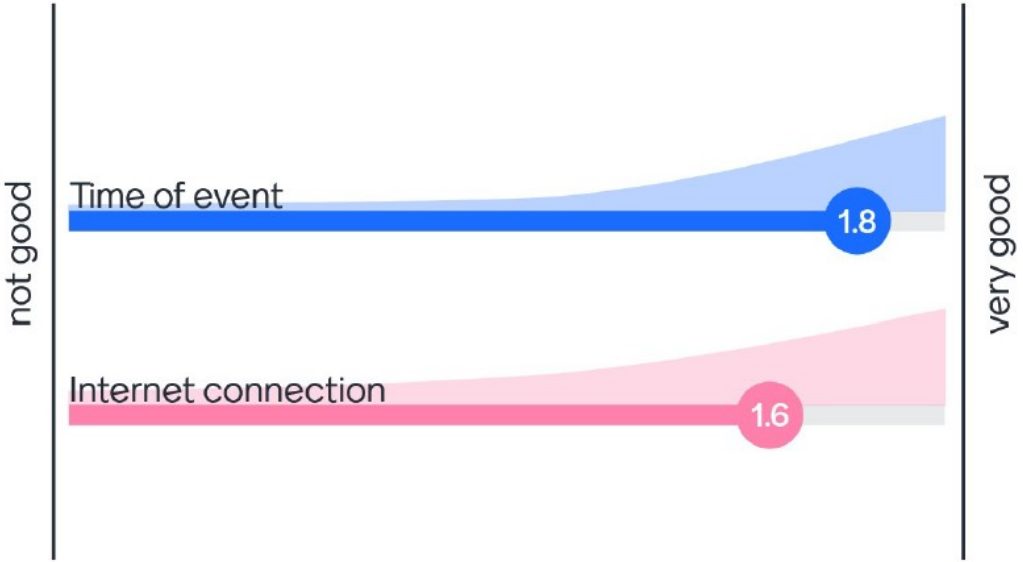


21

german cooperation DEUTSCHE ZUSAMMENARBEIT
Implemented by giz
Digital Transformation Center Rwanda
FAIR FORWARD Artificial Intelligence for all.
RSA Rwanda Space Agency
Carnegie Mellon University Africa
GFA CONSULTING GROUP

How did the logistics work for you?

Mentimeter



20

german cooperation DEUTSCHE ZUSAMMENARBEIT
Implemented by giz
Digital Transformation Center Rwanda
FAIR FORWARD Artificial Intelligence for all.
RSA Rwanda Space Agency
Carnegie Mellon University Africa
GFA CONSULTING GROUP

CLOSING

THANK YOU

for sharing this time with us!

See you next time

Presence Block 1: 15-16 April 2023

Training Manual: ML4EO
Chapter 4: Introduction to ML and Python
Introduction to ML

4.2 Introduction to Machine Learning

Reading material

Now let us start with the first session of the module: Introduction to Machine Learning (ML)

You will learn about the basic concepts of artificial intelligence. This session will address the following questions: How does machine learning work? What are the main steps? What can machine learning do? What are advantages and disadvantages of machine learning?

All of you have a background in data science, so normally this module should just be a refresher, and some may even find it too easy and somewhat redundant. However, considering the various levels of experiences within the group, it is important that we introduce the basic principles of machine learning again to everyone.

This is why this module is designed to give a first introduction into the topic. The good news is that because the module is online and self-paced, you can advance at your own speed according to your background and prior knowledge.

Ready?

Let's go!

Basic concepts of AI and ML

Artificial Intelligence is defined as the branch of computer science that is concerned with the automation of intelligence behaviour. It is a field of study that seeks to explain and emulate intelligent behavior in terms of computational processes. (Lugar et al ,1993)

Artificial Intelligence (AI) is the simulation of human intelligence by machines, it is the ability to:

- Perceive the inter-relationship of facts
- Learn and understand from experience
- Acquire and retain knowledge
- Respond quickly and successfully to a new situation

The central principles of AI include:

- Reasoning, knowledge, planning, learning and communication.

AI is about bringing together computers and humans in ways that enhance human life

What is Learning?

- Learning is a mechanism by which an agent improves its performance by observing the world.
- Learning is the use of experience to gain expertise

Now let us start with the first session of the module: Introduction to Machine Learning (ML)

You will learn about the basic concepts of artificial intelligence. This session will address the following questions: How does machine learning work? What are the main steps? What can machine learning do? What are advantages and disadvantages of machine learning?

All of you have a background in data science, so normally this module should just be a refresher, and some may even find it too easy and somewhat redundant. However, considering the various levels of experiences within the group, it is important that we introduce the basic principles of machine learning again to everyone.

This is why this module is designed to give a first introduction into the topic. The good news is that because the module is online and self-paced, you can advance at your own speed according to your background and prior knowledge.

Ready?

Let's go!

Basic concepts of AI and ML

Artificial Intelligence is defined as the branch of computer science that is concerned with the automation of intelligence behaviour. It is a field of study that seeks to explain and emulate intelligent behavior in terms of computational processes. (Lugar et al ,1993)

Artificial Intelligence (AI) is the simulation of human intelligence by machines, it is the ability to:

- Perceive the inter-relationship of facts
- Learn and understand from experience
- Acquire and retain knowledge
- Respond quickly and successfully to a new situation

The central principles of AI include:

- Reasoning, knowledge, planning, learning and communication.

AI is about bringing together computers and humans in ways that enhance human life

What is Learning?

- Learning is a mechanism by which an agent improves its performance by observing the world.
- Learning is the use of experience to gain expertise.
- If the agent that is learning is a computer, we call it machine learning

Some examples of learning - classifying spam email:

- Let's say you decided to write your own spam email detector program.
- When your program gets a new email, it compares it to all emails that have been labelled in the past and if it matches one of the spam emails seen in the past it is classified it as spam if it does not match any past spam email it classifies it as not-spam.
- Whenever the program fails to detect a spam email because it does not match one of the past emails you inform it that this email was spam. The program then filters out future emails that match the newly added spam email.

Source: Understanding Machine Learning: From Theory to Algorithms: https://dl.acm.org/doi/book/10.5555/2621980

The learning by memorization has one big disadvantage: It is not very useful when a radically different situation is encountered such as when you see a spam email you have never seen before.

The ability to use past experience to make inference about novel experiences is known as **generalization.**

A good learning system must have the ability to generalize to broader population than it has seen.

Source: Understanding Machine Learning: From Theory to Algorithms: https://dl.acm.org/doi/book/10.5555/2621980

Why and when do we need Machine Learning?

1. Tasks that are too hard to program

- Image recognition: this can be done easily by humans and animals, but it is hard to write as a rule-based program
- Driving vehicles
- Speech Recognition

Source: https://www.fritz.ai/image-recognition/

2. Tasks that can't be handled by humans:

- Analyzing genomic data of billions of humans
- Determining which ads/products to show to which users when you have hundreds of millions to billions of users.

https://www.flickr.com/photos/genomegov/27862777945

3. Tasks that require to continuous adaptability to their input:

- Detecting spam emails: Spammers change their techniques everyday writing a program to cope with this requires a team on standby.
- Consumer interests change over time; thus, using machine learning to recommend products

to the customer is necessary.
    - A person that loved action movies last year may now be interested in watching comedy.
    - A user that loved buying video games online at age 16 may want to buy books at age 30.

What can Machine Learning do?

Here are some examples:

1. **Recognize patterns:**
    - Facial identities or facial expressions
    - Handwritten or spoken words
    - Medical images
2. **Generate patterns**
    - Generating images or motion

3. **Recognizing anomalies**
    - Banks check loan applications before making a decision. If the system detects that some of the documents are fraudulent, for example, that your tax number doesn't exist in the system, it will notify the bank employer.
    - Help doctors with diagnosis detecting unusual patterns in MRI and test results
4. **Predictions**
    - Future stock prices or currency exchange rates

Machine Learning: Advantages/Disadvantages

Advantages of Machine Learning

- Efficient Management of Data: Machine Learning is beneficial because it primarily facilitates efficient data management in computers. By managing vast amounts of data and closely observing it, computers make use of ML and interpret data. This management is very helpful in the field of technology and has made the process of data management rapid. With the help of numerous methods of ML, computers identify relevant patterns that help humans to carry out otherwise tedious tasks.
- Valuable Use in Technology: With so many applications of Machine Learning, the world of technology revolves around machine learning. Its valuable use in the field is highly commendable. From Natural Language Processing to Artificial Intelligence, machine learning is omnipresent as it empowers technology to leap forward. With so many benefits of machine learning, the concept of artificial intelligence is advanced more than ever.

- Automated Operations: Machine Learning is defined as the process wherein machines such as computers are made to learn human skills without needing human assistance. This merit of machine learning- its ability to carry out automatic operation- has made it a reputed and regarded concept in the realm of technology. Not only does it work independently, but it also keeps humans away from the process, thus saving time and energy from most of the processes involved in the concept.

Disadvantages of Machine Learning

- Manual Algorithm Selection: While much of the machine learning process is automated and works with the help of computers, the process of algorithm selection in machine learning is still manual and requires human assistance. Is perhaps one of the biggest limitations of

machine learning. This means that while computers design the algorithms, humans designate the algorithms that are supposed to be included in the interpretation of data. This can be a tedious task as it requires humans to run data through various algorithms and identify the one that works the best.

- Delayed Resolution of errors: While management of data is a forte of machine learning, the issue of errors remains a highlighting drawback. Furthermore, the delayed resolution of errors is what keeps the concept of machine learning from becoming perfect! This means that even though the errors occur in the process of Machine Learning, the resolution of errors is a much-delayed process due to the strong reliance of machines on algorithms. This particular trait can be a major disadvantage as it can hamper the process of data management.

- Requirement of Extensive Resources: Lastly, the requirement of extensive resources in machine learning is a concern for many. Why? Because the interpretation of data can be a time-taking process that requires many other equipment's attached to your computer.
- Some ML models are black box models meaning that the models can get good results, but no one is able to explain why the models get good results. Neural networks are the most well-known black box ML models.

Some Types of Machine Learning

Let's say the company *example.com* hires you to create a spam filtering system.

Case 1: The company provided 10000 emails that were labelled as spam/not-spam by email analysis experts and asked you to create a system that generalizes beyond the 10000 emails.

➔ When you use features of your data and labels to teach a machine learning model it is known as **supervised learning.**

Case 2: The company provided 10000 emails that were not labelled and asks you to identify unusual emails.

➔ When you use features of your data **only** to detect patterns in your data it is known as **unsupervised learning**

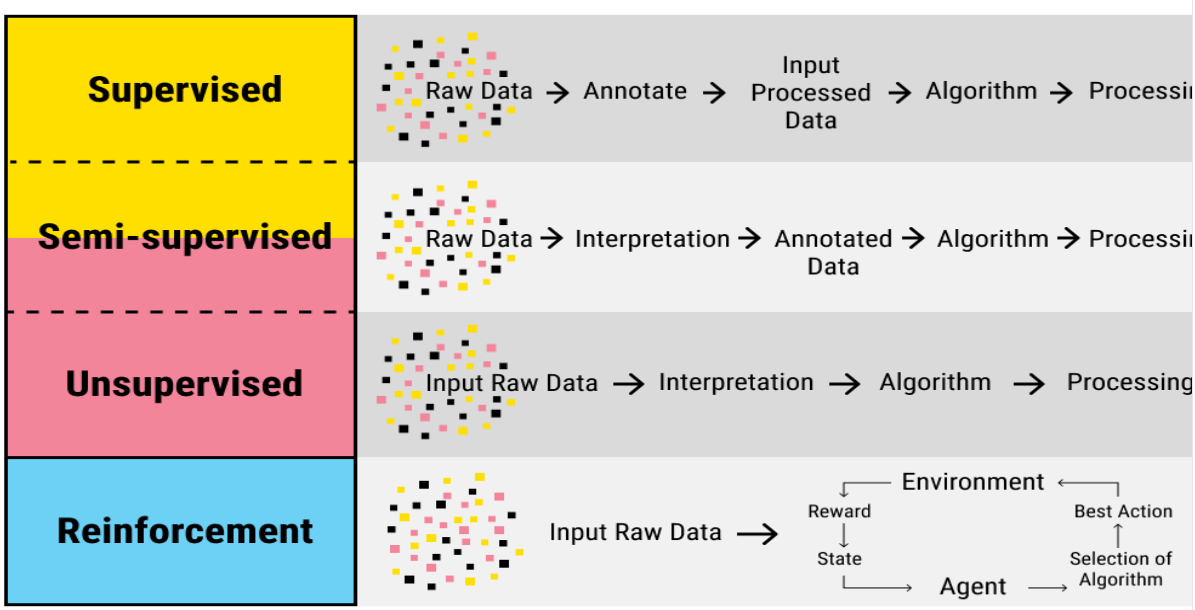
Case 3: The company provided 10000 emails but only 3000 of those were labelled as spam/not-spam. But the company asked you to use all 10000 emails to train a model by devising a way to label the unlabeled 7000 examples.

➔ When you use labelled and unlabeled examples together for training a machine learning model it is called **semi-supervised learning**. Semi-supervised learning requires some approach that generates labels for unlabeled examples.

Case 4: Now, let's say the company asked you to create an email summarization system that returns a summary of the email with gist of the email unchanged but with as few words as possible. The company gave you the 10000 emails it has.

➔ You decide to train the system by having it output a summary of the examples you gave it and penalizing it for producing a summary that contains more than 50 words. Every word more than 50 words constitutes a penalty. To make sure you have a sound text output you also penalize your model whenever it produces a grammatically incorrect output.

➔ When you use rewards and penalties to train an ML model to behave in a desired manner it is called reinforcement learning. In Reinforcement learning, the model is trained by interacting with the environment/data producing an output and receiving a reward or penalty for the output it received.



Source: https://labelyourdata.com/articles/machine-learning-and-training-data

A simple example: Supervised Learning

Imagine you have just arrived in Zodirian island to harvest and export papayas to Rwanda.

You want to predict whether a papaya you see is sweet or not. Based on your experience with oranges you decide to determine papaya sweetness based on the colour of the papaya and the softness of the papaya. We will refer to the colour and softness as the selected features. You want to learn a rule that allows you to predict the sweetness of a papaya given the papaya's colour and softness. To learn something, we need a learner. In this case, learner is an **algorithm that takes training data to output a prediction rule (classifier)**.

Source: Understanding Machine Learning: From Theory to Algorithms: https://dl.acm.org/doi/book/10.5555/2621980

Image source: https://freesvg.org/1546870776

Inputs of a Learner: Training Data

- The input of a learner is a training data.
- **Domain Set**: The set of all objects that we would like to make predictions for
  - In the papaya classification example, the domain set is the set of all papayas in the island.
  - Sometimes objects in domain set maybe represented by their features (colour and softness) in our case.
- **Label Set**: The set of all applicable labels to the objects in the domain set
  - In the papaya classification example, the labels are sweet/not-sweet.

- **Training data**: is a set of labelled domain points.

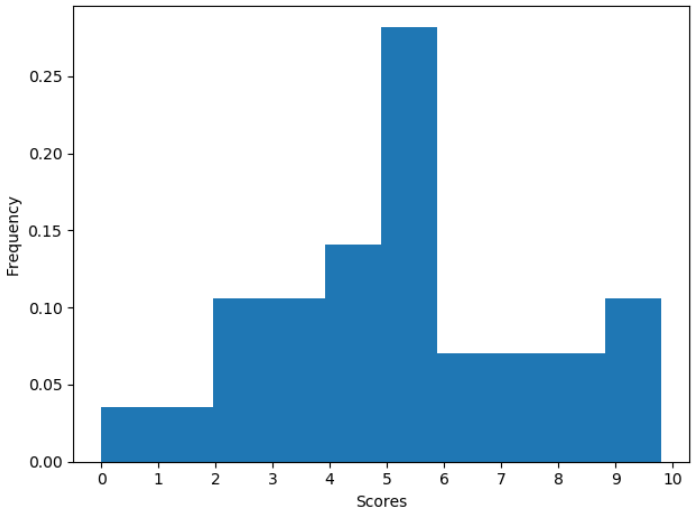
{
(papaya1_color, papaya1_softness, sweet),
(papaya2_color, papaya2_softness, sweet),
(papaya3_color, papaya3_softness, not-sweet),
…,
(papaya50_color, papaya50_softness, not-sweet),
…}

The training data above is a set of (feature1, feature2, label). In an unsupervised ML task, you will not have a label in the training data.

Distribution of Labels

To understand the content of the next few concepts, we need to understand the concept of distribution.

Distribution is a function that tells how likely a certain characteristic is in some group or population. For example, the following distribution indicates the likelihood of scores in some exam administered at some school.



The above figure shows based on data collected over many years how likely a student is to get a score of 1, 2, 3, ...,10, out of 10. The figure shows that there is less than 5% chance that a student scores a 1 out of 10 and greater than 25% chance that a student scores 5.

So, the distribution of student scores tells us the likelihood of scores.

Every observable phenomenon has a distribution of the events it contains. For instance, the distribution of number of lightning strikes per hour during a rainstorm, the distribution of number of car accidents in a day in Kigali. Let's take the second example, the distribution of number of car accidents in Kigali tells us that how likely it is to have no accidents, one accident, two accidents, … in the city of Kigali.

Every observable quantity has a distribution over other quantity. For example, the distribution of weight over people that are taller than 170 cm. If we know the distribution of weight for people taller than 170 cm, we can say many things about the group like 40% of people taller than 170cm weigh more than 70kg, only 10% of people taller than 170cm weigh less than 50kg, and so on…

Training Data Generation

- We have seen that any observable phenomenon or quantity has a distribution. Let's see how we can use this with our example.
- Thus, there is a distribution of sweetness of papayas over all attributes of papaya. We know this because sweetness is an observable attribute of papayas.
    - What does this mean? Sweet papayas occur in certain frequencies for a given papaya attribute. Look at the following simple statements about the distribution of sweetness.
        - 30% of yellow papayas are sweet
        - 60% of hard papayas are not sweet.
        - 45% of mushy and green papayas are sweet.
    - All of the above example tells us the likelihood of being sweet/not-sweet given attributes (color and/or softness) of our data

- No one knows what the actual distribution of sweetness over the papaya attributes is
- If we knew the distribution, we wouldn't need machine learning.
    - We would simply use the distribution to determine the best classifier.

- Since we have no idea what the distribution of the labels over our domain set is, we use the next best thing: samples collected from our domain set.
    - In the papaya classification example, we go to the Zodirian island and collect papayas.

Simply collecting papayas is not enough to create a training set. Why?

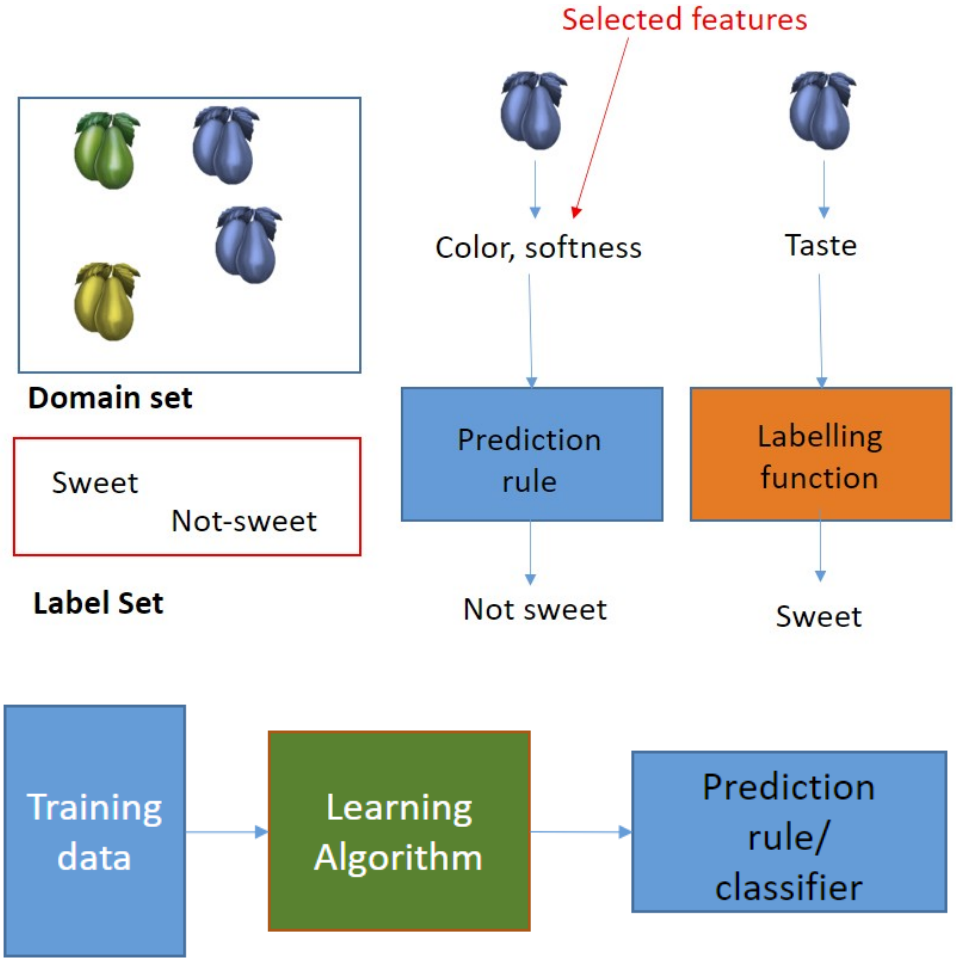
Training Data Generation: Labelling Function

- A training set needs to have points in the domain set and their respective labels (in supervised ML task)
- To get labels, we need a labelling function.
- We use the term labelling function to encapsulate any human or system that can provide labels for a given ML task.
    - In many instances, the labels are provided by humans. In this case, the human can be considered a labelling function that takes the features as input and give you a label for the sample as output.
    - In some instances, labels can be computed in an automated manner. Let's say you want to train a model that predicts crop field area from satellite images. To obtain labels for your training dataset you may simply use an API to query the local district office's database for the size of the crop fields at the coordinates of interest. Once the model is trained on area where field area is available the ML model can be applied in an area where the local district office doesn't have such data.
    - In this case, the code you use to query the database to obtain labels can be considered a labelling function.
- Where do we get the labels from?
    - A labelling function is a function that takes **features** of your data to output the reference label or the ground truth.
    - In the case of papaya sweetness, it can be any function (or human) that can use a papaya in some way to **correctly** tell you whether a papaya is sweet or not.
    - Remember, selected features are the features we chose to use to do our ML task.
    - If we know a labelling function that can take the **selected features** and label all instances in the domain set correctly, we don't need machine learning.
        - We already have a solution
        - In general, the labelling function must use features other than the set of **selected features**.
        - There are some exceptions where the features for prediction/training and labelling are the same.
    - To get the labels we use a labelling function that can use features other than the selected features.
        - In the papaya classification example, we use a human as labelling function.
            - The human tastes the papayas to label the papayas we collected as sweet/not sweet.
            - Notice how the feature used by the "labelling function" is different from the features we would like to use with our classifier (taste for labelling vs. colour and softness for classification)
- This section discussed the general approach to generating training data the following section will further discuss many practical aspects of training data generation such as sources of training data.

Source: Understanding Machine Learning: From Theory to Algorithms: https://dl.acm.org/doi/book/10.5555/2621980

Story so far

- We have a domain set over which we would like to make predictions.
- We have a label set which contains the set of all possible labels.
- A prediction rule/classifier that takes the selected features to output a predicted label.
- A labelling function that takes some features not in the selected features to output the true labels



How can we determine if the prediction rule/classifier is good?

Measure of Success

- The **error of a classifier** (prediction rule) is the probability that the **classifier** outputs the **wrong label** for a random data point sampled from the domain set.
  - This error is also known as the generalization error, or true error.
- What does it mean to output a wrong label?
  - A classifier is said to output a wrong label if the label it produced does not match the label produced by the labelling function for the same input.

- **The goal of machine learning is to minimize generalization error for the task at hand.**
- A model with good generalization is a model with the smallest generalization error.

Source: Understanding Machine Learning: From Theory to Algorithms: https://dl.acm.org/doi/book/10.5555/2621980

Learner Input and Output: Recapped

- A learner is an **algorithm** that takes a training set generated from an **unknown distribution** as input and outputs a classifier.
- The classifier takes features of an element in the domain set.
- The classifier outputs labels
- The learner does not have access to the distribution of labels or the labelling function.
    - Without knowing these two it is not possible to determine the generalization error. Why?
    - So how will the learner evaluate the quality of the classifier it outputs?

Training Error

- Since a learner can't assess the generalization error of a classifier it uses the training error to assess the quality of the learned classifier **during training**
- Training error is defined as the fraction of training examples that a learned classifier produces wrong output on.
    - If the labels produced by a classifier on 20 of 100 training examples is wrong the training error will be 0.2
- Training error is useful as it does not require access to the distribution.
- However, training error can be misleading and lead to overfitting.

Source: Understanding Machine Learning: From Theory to Algorithms: https://dl.acm.org/doi/book/10.5555/2621980

Overfitting

- Look at the image of your favorite island at the bottom.
- Different papayas with different features are shown on the image.
- The labels assigned by the labelling function are shown on the papayas.
    - Not sweet ✗
    - Sweet ✓

Image source: https://freesvg.org/1546870776

Source: Understanding Machine Learning: From Theory to Algorithms: https://dl.acm.org/doi/book/10.5555/2621980

- Now assume you only collected data from the region encircled by the rectangle.
- This data can be considered noisy because it misrepresents aspects of the true distribution.
- A good ML model is expected to learn most of the things that are generally true and disregard noisy aspects of the data.
- From the data shown in this region some bad learner might output a classifier that predicts sweet if the papaya is yellow and not-sweet otherwise. This can happen because the collected training dataset that contains only yellow sweet papayas.
- A good learner is expected to base its classification on as many of the selected features as possible and give rules that are generally true.
- A bad learner will simply give rules that are true for the training dataset (maybe even perfectly true) that are not generally true.
- Clearly the rule the classifier learned is not true outside the rectangle and the classifier will not generalize well.

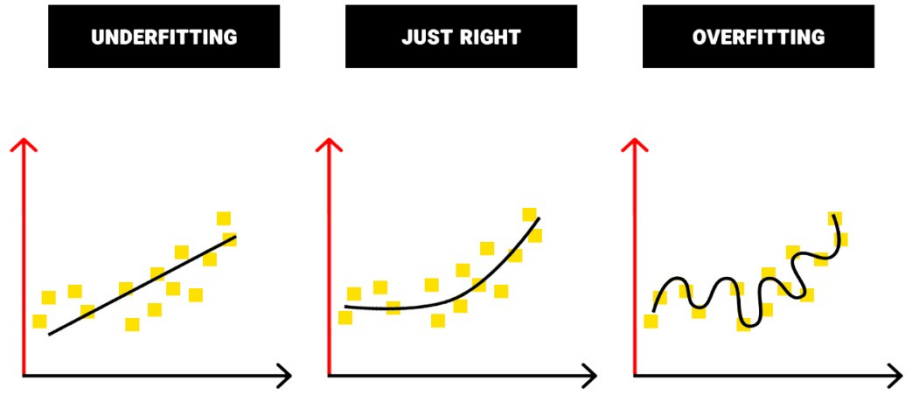
Image source: https://freesvg.org/1546870776

- Overfitting is when the classifier learns so much about the training data that it learns facts about the training data that are not generally true.
- In formal terms, overfitting is when a classifier learns the noise in the training data.
- A classifier that overfits the training data is not good at generalizing to unseen examples.
- The primary goal of learning is generalization; thus, a classifier that overfits the training data is not a good classifier.
- Overfitting can result from poorly collected training set or using a classifier that is too big.

Source: Shai Shalev-Shwartz and Shai Ben-David (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press 40 W. 20 St. New York, NY, United States, or https://dl.acm.org/doi/book/10.5555/2621980

Underfitting

An ML model can also underfit a dataset. A model is said to be **underfitted** when it is not able to capture the underlying trend of the data. It means the model shows poor performance even with the training dataset. In most cases, underfitting issues occur when the model is not perfectly suitable for the problem that we are trying to solve. The graphic below shows how models can overfit or underfit a given dataset.



Source : https://labelyourdata.com/articles/machine-learning-and-training-data

Measuring Generalization Error

- To determine the actual usefulness of a model one must measure the generalization error.
- It is impossible to measure the actual generalization error of a classifier without knowing the distribution of your labels over your domain set.
    - But no one knows the distribution.
- It is possible to approximate the generalization error by measuring the error on randomly sampled data that was not part of the training data

- A randomly sampled data that is used for measuring the generalization error of a classifier is known as test data.
- The error of a classifier on unseen data is known as test error.
- In practice, the goal of machine learning is minimizing test error.
- Determining the actual generalization error requires access to the distribution which no one knows

Recap

So far what we have studied are the overarching goals of Machine Learning.

We defined learning as “gaining expertise from experience.” Everything we have discussed so far is related with this statement.

Learner: Uses experience to produce an expert

Prediction rule (classifier): Is the expert

Training data: Is the experience that teaches the expert

Test error: Determines the level of expertise of the expert. The lower the test error the better the expert is.

Putting this in mind we will now look at the key steps in Machine Learning. Then, the following section will also go into further detail on many practical aspects of machine learning.

Key Steps in Machine Learning

1. **Identifying the domain set, the label set, and the feature selection**

- Identifying the domain set and label allows you to determine the input and output of the classifier you will train
- The features you will use also determine the structure of your classifier.
    - You determine the features to use for an ML problem using domain knowledge, and experience with similar problems

2. **Data collection**
    - Data collection is the process of obtaining training data that will be used as input to the learning algorithm and the test data that will be used to measure the generalization error.
    - When collecting data, we must always ensure that the collected data has similar characteristics to the domain set. This is called representativeness.
        - If you are training a model to determine credit eligibility of a Rwandan but you only use people from Kigali as your training data, your model might fail frequently when dealing with non-Kigalians.
        - Similarly, if your data collected from city of Butare contains only credit eligible individuals but your data from Musanze contains credit ineligible individuals your classifier might learn to simply reject everyone from Musanze and accept everyone from Butare. This is known as a bias.
    - You must always use up-to-date and correct data to prevent wrong outcomes or predictions.
    - Good data is relevant, contains very few missing and repeated values, and has a good representation of the various subcategories/classes present.

3. **Preparing the Data**
    - After collecting data, they have to be prepared. Here are some of the steps that can be used to prepare data that will be discussed in greater detail in 4.2.
        - Data is put together and randomized. This helps make sure that data is evenly distributed, and the ordering does not affect the learning process.
        - Data cleaning
        - Data visualization
        - Splitting the cleaned data into two sets
        - Balancing Label frequency
        - Standardization
        - One hot encoding
        - Augmentation

Source: https://www.kaggle.com/code/dansbecker/using-categorical-data-with-one-hot-encoding

4. **Choosing a Model**
    - A machine learning model determines the output to be obtained after running a machine learning algorithm on the collected data. It is important to choose a model which is relevant to the task at hand. Over the years, scientists and engineers developed various models suited for different tasks like speech recognition, image recognition, prediction, etc. Apart from this, it is better to check if the selected model is suited for numerical or categorical data and choose accordingly.
5. **Training the Model**
    - Training is the most important step in machine learning. In training, you pass the prepared data to your machine learning model to find patterns and make predictions.

The learning algorithm (learner) modifies the classifier (prediction rule) based on the training data.

- Generally, the objective of the learning algorithm in training is to minimize training errors. It is important to take great care to avoid overfitting to the training data.

6. **Hyperparameter Tuning**
   - Once you have created /selected the model, see if its accuracy can be improved in any way. This is done by tuning the hyperparameters present in the model. The following section will discuss what hyperparameters are in detail.
7. **Evaluating the Model**
   - After training the model, it is necessary to check how it's performing. This is done by testing the performance of the model on previously unseen data.. If testing was done on the same data which is used for training, you will not get an accurate measure, as the model is already used to the data, and finds the same patterns in it, as it previously did. This will give disproportionately high accuracy.
8. **Deployment**
   - In the end, you can use your model in a production environment to make predictions on unseen data accurately.
   - Deploying a model may require further setup depending on the computing environment that the classifier will be deployed to. Deploying to a distributed computing environment with hundreds of nodes is very different from deploying to a single server. Similarly, deploying to a single server is very different from deploying to microcontroller unit.

Exercise materials and tasks

Quiz questions

Please answer the following questions to test your understanding of artificial intelligence:

1. Machine Learning is the use of experience to gain expertise. The level of expertise of a learned model is evaluated by:
    a. The fraction of examples seen in the past the ML model can exactly remember and use to compare future examples.
    b. The probability that the ML model will make a mistake on a randomly chosen unseen example.
    c. The probability that the ML model will make a mistake on a randomly chosen member of validation set.
    d. The fraction of training examples that a learned classifier produces wrong output on

2. For which of the following tasks would it make sense to use machine learning?
    a. To predict the output of a chemical reaction whose input elements and output compounds are perfectly known.
    b. To do a task such as face recognition that humans can do easily.
    c. To determine a prediction rule for a task where the distribution of the labels over the selected features is perfectly known.
    d. When there is a classification task that can be achieved by using certain features but for some reason, we would like different set of features.

3. Which of the following cannot cause overfitting?
    a. Using a model so simple that it can't perform well on the training data
    b. Training a large and complex model to the point that it learns everything about the data.
    c. Using a poorly sampled data that is not representative of the domain set.

4. You learned that ML has several advantages and disadvantages. For the given statements, sort them into advantages and disadvantages: *efficient management of data, black box models, interpretation of data can be a time-taking process, valuable use in technology, automatic operations without human assistance, algorithm selection is manual, delayed resolution of errors*

Advantages	Disadvantages

Correct answer:

Advantages	Disadvantages
efficient management of data, valuable use in technology, automatic operations without human assistance	**black box models, interpretation of data can be a time-taking process, algorithm selection is manual, delayed resolution of errors**

5. Which of the following is true about the learning algorithm and classifier?
    a. The learning algorithm (learner) and the classifier are the same thing.
    b. Once a classifier is trained the learning algorithm and the classifier are used to classify data
    c. The learning algorithm is an algorithm that modifies a given classifier using the training data.
    d. A classifier uses the selected features to output a predicted class.

4.3 ML training data and web mapping

Reading material

Machine Learning in Practice

In the previous section we defined ML and discussed the key steps in Machine Learning. In this section, we will discuss the practical aspects of the key steps in Machine Learning in greater detail starting with data collection.

Importance of Data Collection

In the previous section, we discussed that the role of training data is to serve as the experience a learner uses to produce an expert (classifier). Thus, if we provide the wrong data the learner will produce a classifier that doesn't do what we want it to do.

Without a foundation of high-quality training data, even the most performant algorithms can be rendered useless. Indeed, robust machine learning models can be crippled when they are trained on inadequate, inaccurate, or irrelevant data in the early stages. When it comes to training data for machine learning, a longstanding premise remains painfully true: Garbage In, Garbage Out. Therefore, ***no element is more essential in machine learning than quality of training data***.

Simply put, training data teaches the machine learning model. The learning algorithm (learner) continuously updates the classifier (model) based on the data.

For these reasons, the need for quality, accurate, complete, and relevant data starts early on in the training process.

It is also important to have the right amount of data. For many tasks, even if you have high quality data it is not possible to produce a good classifier with a small amount of data. Greater quantity of correctly sampled data reduces the chances that the classifier we train overfits to the dataset.

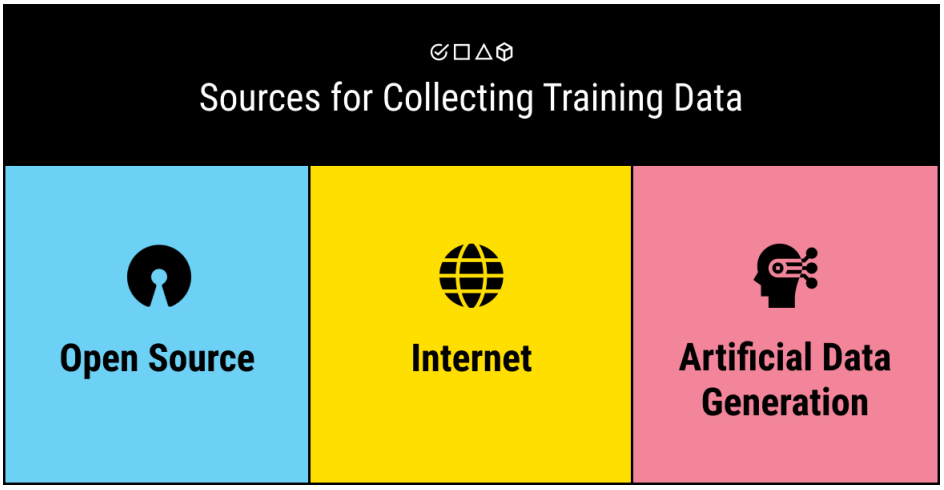
All in all, we must ensure we use quality sources of data, collect the right amount of data, and have some sort of quality assurance to ensure a successful training of our ML model.

Data Collection: Sources of Data

In the Papaya classification example in the previous section, we discussed how to create a training dataset for ML. In many instances one does not need to collect data on their own given that there are a multitude of high-quality open data sources.

It is possible to gather your own data and label it yourself. But since data collection is a lengthy process you can use an in-house team, crowdsourcing, or a data labeling service to do the work for you. You can also purchase training data that is labeled for the data features you determine are relevant to the machine learning model you are developing.

There are quite a few different sources to get the training data sets from, and the choice of these sources depends on your goals, the requirements of your machine learning project, as well as your budget, time, and personnel restrictions.



Source: https://labelyourdata.com/articles/machine-learning-and-training-data

1. Open-source training data sets: This might be an acceptable solution if you're very lucky or otherwise for smaller businesses and start-ups that don't have enough free resources to spend on data collection and labelling. The great benefit of this option is that it's free and it's already collected. But there's a catch (isn't there always?): such data sets were not initially tailored for your algorithm's specific purposes but for some other project's. What this means for you is that you'll need to tweak and probably re-annotate the data set to fit your training needs. Many open-source data sets are available on the internet from sources such as World Bank, Yahoo Finance, and Quandl.

2. Web scraping: Web scraping is the process of extracting data from a website. This is a very common way of collecting training data sets that many machine learning companies use. This means that you use a program to extract data from a web page on the Internet.

3. Internet of Things (IoT): Sometimes sensors, cameras, and other smart devices may provide you with the raw data that you will later need to annotate by hand. This way of gathering a training data set is much more tailored to your project because you're the one collecting and annotating the data. On the downside, it requires a lot of time and resources, not to mention the specialists that know how to clean, standardize, anonymize, and label the data.

4. Artificial training data sets (Data Augmentation): This is the way that has started to gain traction in recent years. What it basically means is that you first create a machine learning model that will generate your data. This is a great way if you need large volumes of unique data to train your algorithm. It saves financial and personnel resources as it only needs to spend them on designing the model to create your data. Still, this method of collecting training data requires a lot of computational power, which is not usually freely accessible for small and middle-sized businesses.

Besides, if you need truly vast amounts of data, it will take some time to generate a voluminous high-quality training data set.

In some cases, it is possible to generate artificial data deterministically; that is, without using a machine learning model. One popular application of machine learning that heavily uses artificial data generated deterministically is optical character recognition.

5. In-situ data collection: any observation taken by an instrument in direct contact with the medium it "senses" is called an in-situ observation. In other words, In-situ data collection is collection of data by direct measurements. Temperatures measured by standard thermometers, wind speeds and

directions measured by a cup anemometer and wind vane, and precipitation measured by a rain gauge are all very common in-situ weather observations. You may have even taken your own "homemade" in-situ weather observations before. Picking up blades of grass and tossing them in the air to get a sense for the wind direction, for example, would be an example of an in-situ observation. In remote-sensing, in-situ data is used to verify different measurements taken by remote sensing equipment.

Source: https://www.e-education.psu.edu/meteo3/node/2224

Data Collection: Quantity of Data

How much training data is needed?

There's no clear answer - no magical mathematical equation to answer this question - but more quality data is better. The amount of training data you need to create a machine learning model depends on the complexity of both the problem you seek to solve and the algorithm you develop to do it. One way to discover how much training data you will need is to build your model with the data you have and see how it performs, by trial and error.

There are a few very broad guidelines that might help you get a basic idea about why this question has no answer:

- Usually, more sophisticated models with more attributes and links between them (like the Artificial Neural Networks) will require more data to train properly.
- The scope of application along with the complexity of the real-life phenomena that your model is being trained to predict also plays a role in how much training data will be needed. Beware of the exceptions and blind spots.
- With time, you will likely need to re-train or tweak your model as the trends that it predicts change, which will require more data in the long-term.

As you can see, a lot of factors play into the understanding of how much training data is enough. As a rule of thumb, experienced engineers have at least a general idea about the amount of data that will suffice to train your model. You should start listening to them, and then get more training data as you go.

Data Collection: Quality of training data

For a quality training data, the below points should be considered:

1. ***Relevant:*** The very first quality of training data should be relevant to the problem that you are going to solve. It means that whatever data you are using should be relevant to the current problem. For example, if you are building a model to analyze social media data, then data should be taken from different social sites such as Twitter, Facebook, Instagram, etc.
2. ***Uniform:*** There should always be uniformity among the features of a dataset. It means all data for a particular problem should be taken from the same source with the same attributes.
3. ***Consistent:*** In the dataset, the similar attributes must always correspond to the similar label in order to ensure uniformity in the dataset
4. ***Comprehensive:*** The training data must be large enough to represent sufficient features that you need to train the model in a better way. With a comprehensive dataset, the model will be able to learn all the edge cases.

Quality training data is vital when you are creating reliable algorithms.

Source: https://www.cloudfactory.com/training-data-guide

Data Collection: Factors Affecting Data Quality

What affects training data quality?

There are three main factors that can help you predict the level of quality you can expect from the people who work with your data - whether your workers are in-house, crowdsourced, or outsourced teams.

- **People**: The selection, development, and management of workers
- **Process**: How workers do the work - from onboarding to task instructions to quality control workflow
- **Tools**: The technology to access the work, manage workers, and maximize quality and throughput

Source: https://www.cloudfactory.com/training-data-guide

Data Preparation

We will now focus on the process of data preparation, the third step in the key steps of ML. In section 4.1, we discussed how the training data is used to teach the model, but a randomly sampled data is used to evaluate the test error. We call the data that is used to evaluate the test error the test data.

What is a test dataset?

Once the model is trained with the training dataset, it's time to test the model with the test dataset. This dataset evaluates the performance of the model and ensures that the model can generalize well with the new or unseen dataset. The test dataset is another subset of original data, which is

independent of the training dataset. However, it has some similar types of features and class probability distribution and uses it as a benchmark for model evaluation once the model training is completed. Test data is a well-organized dataset that contains data for each type of scenario for a given problem that the model would be facing when used in the real world. Usually, the test dataset is approximately 20-25% of the total original data for a machine learning project. If you have a large dataset (with 1 million+ entries), as small as 2% of the total original data can be used as a test set.

At this stage, we can also check and compare the testing accuracy with the training accuracy, which means how accurate our model is with the test dataset against the training dataset. If the accuracy of the model on training data is greater than that on testing data, then the model is said to have **overfitting**.

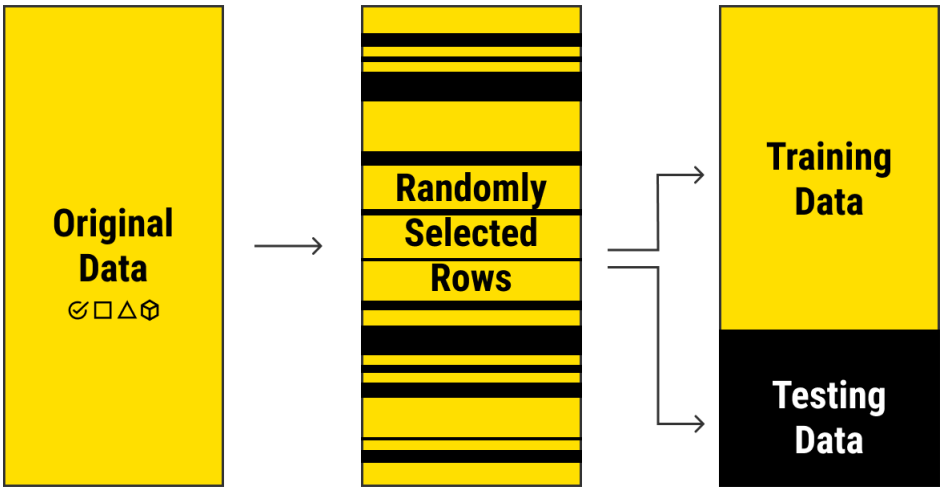
The testing data should:

- Represent a part of the original dataset.
- It should be large enough to give meaningful predictions.

Data Preparation: Train-Test Split

Splitting the dataset into train and test sets is one of the important parts of data pre-processing.

If the model is trained with a training set and then tested with a completely different test dataset, then the model will not be able to understand the correlations between the features. Therefore, training and testing the model with two different datasets will decrease the performance of the model. Hence it is important to split a dataset into two parts, i.e., **train and test set.**



Source: https://labelyourdata.com/articles/machine-learning-and-training-data

In this way, it is easy to evaluate the performance of the model. Such as, if it performs well with the training data, but does not perform well with the test dataset, then it is estimated that the model may be overfitted.

Recently, there are different machine learning approaches that are trained with one dataset and tested on a completely different dataset. The task of designing classifiers that work well on data that looks different from the one they were trained on is called domain adaptation. For example, if you have a self-driving car trained with street data from the city of London, you still will want the car to work as well in the streets of Kigali. To achieve this one can use different domain adaptation

techniques.

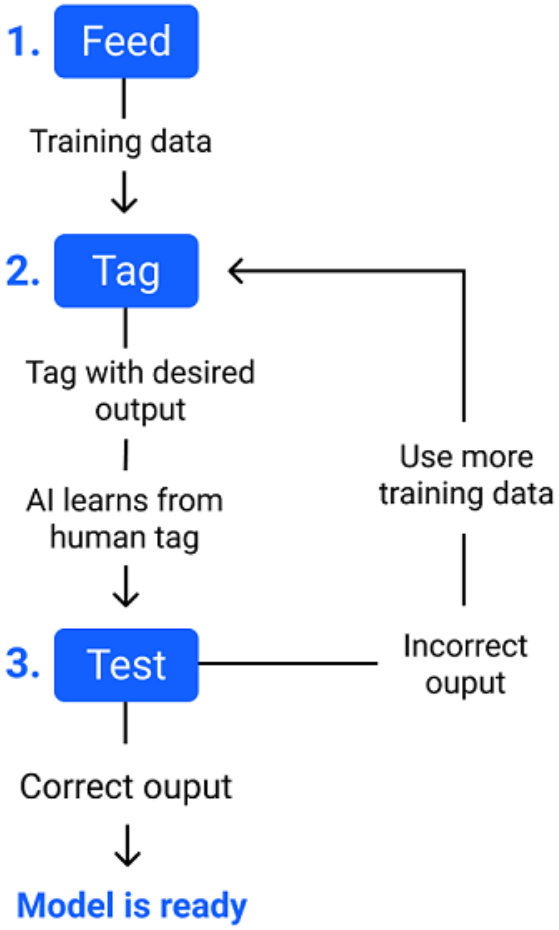
Data Preparation: Recap Training Vs. Test Data

- The main difference between training data and testing data is that training data is the subset of original data that is used to train the machine learning model, whereas testing data is used to check the accuracy of the model.
- The training dataset is generally larger in size compared to the testing dataset. The general ratios of splitting train and test datasets are 80:20, 70:30, or 90:10.
- Training data is well known to the model as it is used to train the model, whereas testing data is like unseen/new data to the model

In most ML applications, collecting data once and training may not meet performance expectations. Sometimes model evaluation on the test data might indicate more data is needed; hence, the data collection, and data preparation steps might have to be redone after every data collection. We can understand the whole process of training and testing in three steps, which are as follows:

1. **Feed**: Firstly, we need to train the model by feeding it with training input data.
2. **Define**: Now, training data is tagged with the corresponding outputs (in supervised learning), and the model transforms the training data into text vectors or a number of data features.
3. **Test**: In the last step, we test the model by feeding it with the test data/unseen dataset. This step ensures that the model is trained efficiently and can generalize well.

The above process is explained using a flowchart given below:



Source: https://www.javatpoint.com/train-and-test-datasets-in-machine-learning

Data Preparation: Some Data Preparation Steps

A lot of elaborate data preparation techniques can be applied. Here we discuss some of the widely used data preparation steps.

- Data cleaning: cleaning the data to remove unwanted data, missing values, rows, and columns, duplicate values, data type conversion, etc. You might even have to restructure the dataset and change the rows and columns or index of rows and columns.
- Data visualization: just to understand how data is structured and understand the relationship between various variables and classes present.
- Splitting the cleaned data into two sets - a training set and a testing set. The training set is the set your model learns from. A testing set is used to check the accuracy of your model after training.

- Label frequency: If there are 10.000 samples of sweet papayas in 10 samples of not-sweet papayas than model can assume that all papayas are sweet because doing so results in a small training error (< 0.0001). A dataset with skewed label proportions is called imbalanced. The class that has a large proportion in the dataset is called majority class.

- Standardization: standardization is the rescaling of features to ensure the mean and the standard deviation to be 0 and 1, respectively. This is important when you have measurements in different units. If you have two columns in your data, one measured in kilometers and another in millimeters, the mm measurements will be large numbers while the km measurements will be small numbers. Consequently, your model might learn to weight the mm measurements more than the km measurements. When this is not desired, the features are standardized so all measurements have similar orders of magnitude.

- One hot encoding: One-hot encoding is a way of representing categorical data as an array of zeros and ones. In one-hot encoding, you will always have an array (vector) with length equal to the number of categories. One of the elements in the array will be one while the rest are zero. The position of the one in the array indicates the category the data point belongs to. Look at the following example,
  In the image below, having a 1 on the first column (and zero elsewhere) indicates the data point is Red. Thus, the array (vector) [1, 0, 0] indicates a Red item.

Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

Source: https://www.kaggle.com/code/dansbecker/using-categorical-data-with-one-hot-encoding

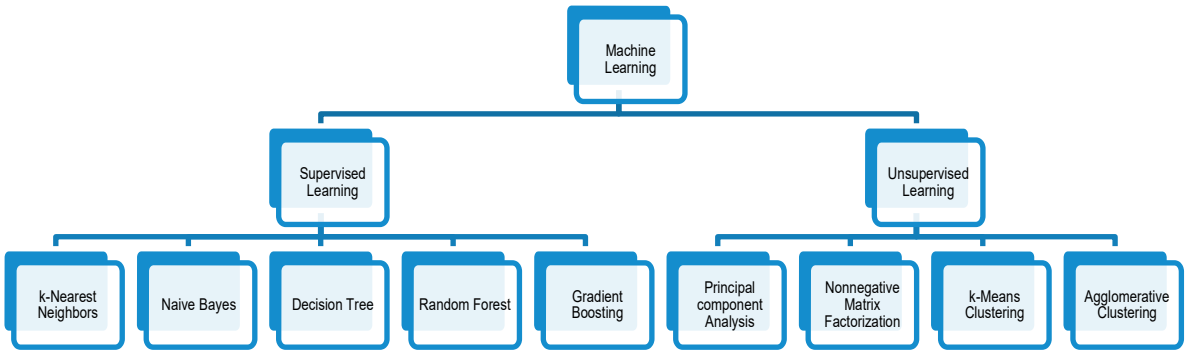
- Augmentation: This is artificially generating training data to your dataset. Section 4.2.12 discusses this in greater detail

Model Choice

Model choice is the third key step in ML.

After the data is prepared we have to select the kind of model we will train. There are many classes of ML methods to choose from. A big part of the choice depends on the data we collected. If the data collected is unlabelled, then we must use models amenable to unsupervised learning but if we have labels we can use models that require labels.

The following image shows different types of ML models that can be used in supervised and unsupervised settings (excluding deep learning models).



Model Choice: Considerations

We have seen that a large part of the model choice depends on the kind of data available. However, there are other considerations as presented below.

1. **Explainability:** In many situations, explaining the results of a model is paramount. Explainability is important in medical applications, legal applications, and critical industrial applications.
   Unfortunately, many algorithms work like black boxes, and the results are hard to explain regardless of how good they are. The lack of explainability may be a deal-breaker in those situations. Linear Regression and Decision Trees are good candidates when explainability is an issue. Neural networks, not so much. Understanding how easy it is to interpret the result of each model is important before picking a good candidate.
2. **Complexity:** A complex model can find more interesting patterns in the data, but at the same time, it will be harder to maintain and explain. A couple of loose generalizations to keep in mind: More complexity can lead to better performance but also larger costs. Complexity is inversely proportional to explainability. The more complex the model is, the harder it will be to explain its results. Putting explainability aside, the cost of building and maintaining a model is a crucial factor for a successful project. A complex setup will have an increasing impact during the entire lifecycle of a model.
3. **Dataset Size:** The amount of training data available is one of the main factors you should consider when choosing a model. A Neural Network is really good at processing and synthesizing tons of data. A KNN (K-Nearest Neighbors) model is much better with fewer examples. Going beyond the amount of available data, a related consideration is how much of it you truly need to achieve good results. Sometimes you can build a great solution with 100 training examples; sometimes, you need 100,000. Use this information about your problem and the amount of data to choose a model that's capable of processing it.
4. **Training time and cost:** How long it takes, and how much it costs to train a model? Would you choose a 98%-accurate model that costs $100,000 to train or a 97%-accurate model that costs $10,000? Of course, the answer to this question depends on your individual circumstances. Models that need to incorporate new knowledge in near real-time can't afford long training cycles. For example, a recommendation system that needs to be constantly updated with every user's action benefits from an inexpensive training cycle. Balancing time, costs, and performance is crucial when designing a scalable solution.
5. **Inference time:** How long does it take to run a model and make a prediction? Imagine a self-driving system: it needs to make decisions in real-time, so any model that takes too long to

run can't be considered. For example, most of the processing needed to develop predictions using KNN happens during inference time. This makes it expensive to run. However, a Decision Tree will be lighter during inference time and will require more time during training.

Source: https://towardsdatascience.com/considerations-when-choosing-a-machine-learning-model-aa31f52c27f3

Model Choice

Oftentimes ML engineers train multiple models and use the evaluation results to make a model choice. Although the considerations discussed can be helpful to narrow down model choice, they should not be used to pick a single model. It is better to train multiple models that meet the requirements of the ML task and base your decisions on the performance of the models.

Training the Model

The following key step in ML is training the model. The way model training is carried out depends on the model used, the data storage, and the type of computational resource used. But in general, in this step the training data is processed by the learning algorithm to output a classifier.

Hyperparameter Tuning

The next step in ML is hyperparameter tuning. Before we discuss hyperparameter tuning we must define hyperparameter and distinguish it from parameter.

Parameter Vs Hyperparameter

- A model parameter is a configuration variable that is **internal to the model** and whose value can be **estimated from the given data**.

- Parameters:
  - They are required by the model when making predictions.
  - Their values define the skill of the model on your problem.
  - They are estimated or learned from data.
  - They are often **not set** manually by the practitioner.
  - They are often saved as part of the learned model.

- A model hyperparameter is a configuration that is external to the model and whose value cannot be estimated from data.
  - They are often used in processes to help estimate model parameters.
  - They are often specified by the practitioner.
  - They can often be set using heuristics.
  - They are often tuned for a given predictive modeling problem.

Source: https://www.datacamp.com/tutorial/parameter-optimization-machine-learning-models

Hyperparameter Tuning: Comparing Classifiers

- Let's say you have trained two same classifiers called SVM-1 and SVM-2 with different hyperparameters for our papaya classification task and the classifiers have the performance shown in the table below.

- If you had to use one of the two models for production which one would you use? Why?
- Now your boss comes and tells you that a test error of 0.1 or better is required and you must retrain one of the models. Which one would you re-train?

- Let's say you chose to retrain SVM-1 because it has smaller test error and retrained to get a test error of 0.08.
  Is the test error of the retrained model a good approximation of the generalization error of the retrained model?
  - No, it is not. The reason is by using the information that SVM-1 performs better on the test data to choose it for retraining we can no longer consider the test data unseen.
  - If you have used some data to change anything on one of your classifier, learning algorithm, training settings, or model settings it cannot be considered unseen data.
  - In this case, the test error was used to inform the choice between the settings used for svm-1 and svm-2.
  - So how can such choices about model settings be made and still have a test error that approximates the generalization error?

Model	Training Error	Test Error
SVM-1	0.1	0.14
SVM-2	0.05	0.20

- A third set of data known as validation data is used to make such choices.
- The learning algorithm never sees the validation data as its training input.

Training Vs Validation Vs Test

Training data:
- Is the data that will be seen by the learning algorithm.
- Used by the learning algorithm to produce a good classifier

Validation data:
- Is never used to train the learning algorithm exceptions aside
- Used to evaluate a trained model for further use
- Used by the machine learning practitioner to determine what training settings, classifier settings, … to use for better generalization

Test data:
- Is never used to train the learning algorithm
- Is never used to make any decision about classifier choice, classifier settings, training settings, …
- Used to obtain an approximation of the generalization error of a classifier

Note: There are other approaches to choosing classifier settings that aren't discussed here.
Read more in the following link https://www.cs.cmu.edu/~schneide/tut5/node42.html

Evaluating the Model:

As was previously discussed, we use test data to evaluate the performance of the tuned model. This will give us the final verdict on how well a model does on unseen data.

Deployment:

Machine learning model deployment is the process of placing a finished machine learning model into a production environment where it can be used for its intended purpose. Models can be deployed in a wide range of environments, and they are often integrated with apps through an API (Application Programming Interface) so they can be accessed by end users. The environments where an ML model can be deployed to range from small microcontrollers, to standalone servers, to distributed server cluster, to cloud environments.

The process of actually deploying the model requires several different steps or actions, some of which will be done concurrently.

First, the model needs to be moved into its deployed environment, where it has access to the hardware resources it needs as well as the data source that it can draw its data from. Let's take the ever popular ChatGPT model as an example. ChatGPT is deployed on Microsoft Azure cloud infrastructure. Thus, the trained model will have to be configured to work on the virtual machine Microsoft Azure provided.

Second, the model needs to be integrated into a process. This can include, for example, making it accessible from an end user's laptop using an API or integrating it into software currently being used by the end user. In the case of ChatGPT, the developers have made an API for programmers available at https://api.openai.com/v1/chat/completions. Whenever a request is sent to the url the request is processed and forwarded to the model deployed on Microsoft Azure. The output of the model is forwarded as an HTTP data to the caller. This API is meant for use by developers.

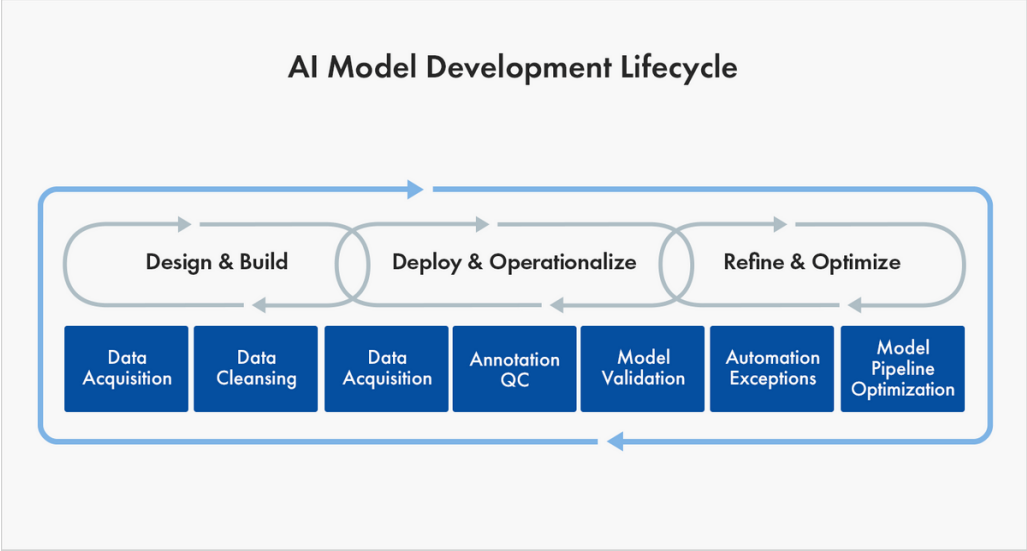
Third, the people who will be using the model need to be trained in how to activate it, access its data and interpret its output. Sometimes we would like to provide access to the power of a model to non-technical users. In such cases, we may design an interface where people can interact with our model by simply using graphical or command line tools. For example, ChatGPT allows interaction with the model through a web interface at https://chat.openai.com

Source: https://www.dominodatalab.com/blog/machine-learning-model-deployment#:~:text=Machine%20learning%20model%20deployment%20is,be%20accessed%20by%20end%20users.

ML Process as a Cycle

Until now we have described the key steps in ML as a straight process. However, the process is actually cyclic that involves multiple loops. Many times the loops in ML cycle might take us as far back as collecting new data. For example, training data can evolve with time.

Training data is used not only to train but to retrain the model throughout the artificial intelligence development lifecycle. Training data is not static: as real-world conditions evolve, the initial training dataset may be less accurate in its representation of ground truth as time goes on, requiring to update the training data to reflect those changes and retrain the model.



Source : https://www.cloudfactory.com/training-data-guide

ML in Remote Sensing

We have seen the key steps in ML in detail. But why do we use ML in remote sensing?

Environmental remote sensing involves the use of satellites and other air-borne instruments to collect data about the environment.

The rapid increase in the volume of remote sensing data obtained from different platforms has encouraged scientists to develop advanced, innovative, and robust data processing methodologies. Machine learning methods are widely applied to remote sensing datasets; they have been used to classify ships from remote sensing images, determine the distribution of palm trees in a forest from images and so much more.

Machine learning algorithms allow a system to learn and improve from data and experience without being specifically programmed, reducing the level of human intervention. This data-driven approach means valuable information about a natural phenomenon can be extracted from the data alone.

Key Points:

- Machine learning with remote sensing can help to improve predictions about the behavior of environmental systems, improve the automation of data analysis, lead to a better management of resources and the discovery of new insights from complex data sets.
- Applications include improved weather forecasting, flood and drought prediction, precision agriculture, forest management and in marine conservation and coastal clean-up projects.
- Wider implementation for remote sensing is limited by the availability of accessible and representative datasets for training the machine learning algorithm. Specific challenges include: the availability of *Analysis Ready Data* which is data that have been processed to a minimum set of requirements and organized into a form that allows immediate analysis with a minimum of additional user effort and interoperability both through time and with other datasets - it requires expertise, time and computational power to prepare; the demands on storage, transfer and processing of large data sets; and the demands on having an accurate and well-developed training data set.

In the following section, we will discuss the image classification and discuss some applications of ML in remote sensing.

Exercise materials and tasks

Quiz questions

Please answer the following questions to test your understanding so far:

1. What is training data?
   a. **A dataset used to train machine learning models**
   b. A dataset that evaluates the performance of the model
   c. A new dataset that the model can generalize well

2. Please connect the steps of the process of training and testing data work in ML with the corresponding activities:

Step 1	The model is tested by feeding it with the test data/ unseen dataset
Step 2	Training data is tagged with the corresponding outputs
Step 3	The model is fed with training input data

System feedback:

- Student answers different from below: “Sorry, that’s not correct, please check the process of training and testing data work in ML”

Step 1	**The model is fed with training input data**
Step 2	**Training data is tagged with the corresponding outputs**
Step 3	**The model is tested by feeding it with the test data/ unseen dataset**

System feedback:

- Student answers like follows: ”Correct, these are the steps of the process of training and testing data work in ML:”

3. How much training data do we need for machine learning?
   a. Algorithms tell you when it’s sufficient
   b. We don’t know
   c. **You'll get perception with experience**

4.4 Machine Learning algorithms for image classification

Reading material

Machine Learning Algorithms for Image Classification

So far you learned what machine learning is and the key steps in devising an ML solution. In this session you will have a look at the general concepts and algorithms for image classification.

General Concepts

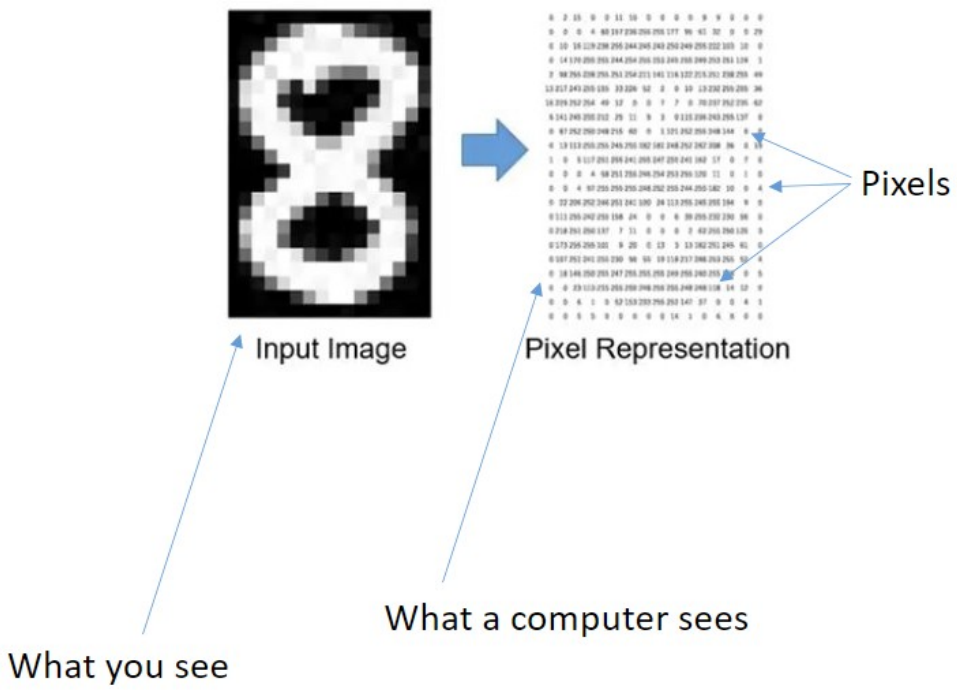
As a broad subfield of artificial intelligence, machine learning is concerned with algorithms and techniques that allow computers to “learn” by example. The major focus of machine learning is to extract information from data automatically by computational and statistical methods. It allows us **to give our data a voice**. Machine learning is now being routinely used to work with large volumes of data in a variety of formats such as image, video, sensor, health records, etc.

When machine learning is used for classification, empirical models are built to classify the data into different categories, aiding in the more accurate analysis and visualization of the data. Applications of classification include facial recognition, credit scoring, and cancer detection. When it is used for clustering, or unsupervised classification, it aids in finding the natural groupings and patterns in data.

Applications of clustering include medical imaging, object recognition, and pattern mining. Object recognition is a process for identifying a specific object in a digital image or video. Object recognition algorithms rely on matching, learning, or pattern recognition algorithms using appearance-based or feature-based techniques.

What is an image?

- For the purposes of computer science, an image is an array of pixels.
- A pixel (picture element) is the smallest addressable unit in an image and is represented by a single number or a group of numbers and its position in the array.
- Since a pixel is just a number or a group of numbers and an image is an array of pixels then an image is just an array of numbers
- Depending on the type of the pixels an image can be a two-dimensional or three-dimensional array



Source: https://medium.com/analytics-vidhya/cnn-series-part-1-how-do-computers-see-images-32462a0b33ca

Source: https://www.ibm.com/docs/bg/gddm?topic=ivu-what-is-image

What is image classification?

- Image classification is the task of assigning a label or class to an entire image.
- A model takes an image as input and outputs probability that the image contains an object that belongs to one of the classes in the label set.
- The main aim of any image classification-based system is to assign semantic labels to captured images and consequently.

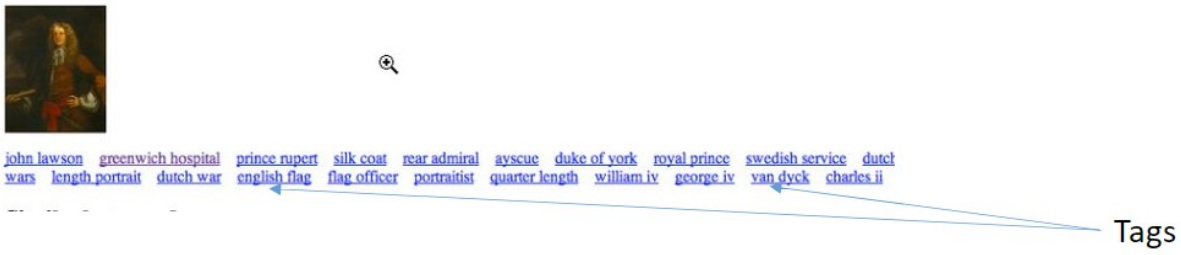
Source: https://huggingface.co/tasks/image-classification

- Image classification can be used for image search
  - Image search is when you use text to find images that contain the object you

described in the text

- Image classification can be used for automatic image tagging
  - Automatic image tagging adding textual or categorical description to an image
  - For example, when a user uploads an image containing an orange cat you may want to add descriptions such as orange, cat, Garfield
- The state-of-the-art image classification systems use deep learning. For example, in the challenge known as ImageNet Large Scale Visual Recognition Challenge (ILSVRC) deep learning methods perform far better than the classical ML methods in classifying 1000 classes of objects.

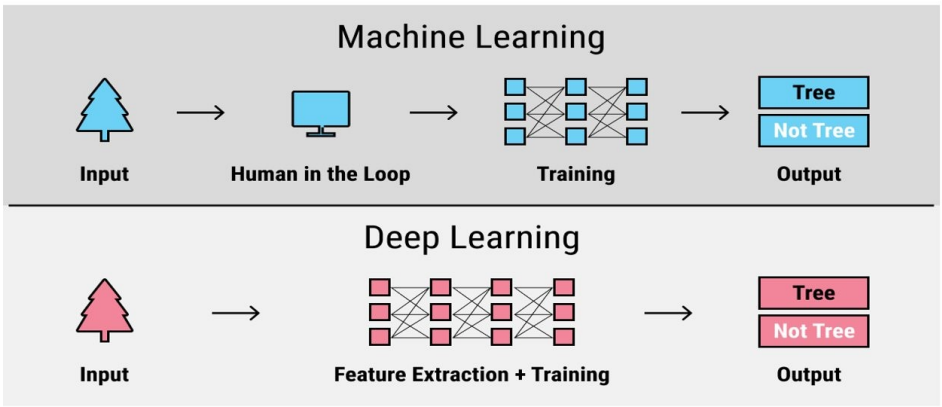
- **What is deep learning?**



Source: https://www.flickr.com/photos/eatyourgreens/2635712607/

Deep Learning

- Deep learning is a subset of machine learning that uses an Artificial Neural Network (ANN) with three or more layers as its classifier. For now, we will not define what ANN is or what it means to have three or more layers. These concepts are discussed in-depth in Module 7.
- Why deep learning?
  - One of the biggest advantages of deep learning as compared to the rest of machine learning is its ability to handle unstructured data well
  - In the rest of machine learning, a human expert curates the set of features to be used for each machine learning task. With deep learning it is not necessary to do this. Deep learning classifiers can identify relevant features on their own.



Source: https://labelyourdata.com/articles/machine-learning-and-training-data

Source: https://www.ibm.com/topics/deep-learning

Is it still hard to grasp what machine learning and deep learning is? Let's try to explain it again with a video.

https://www.youtube.com/watch?v=q6kJ71tEYqM (time duration 7:48)

Traditional Image Classification Approaches

Before the beginning of use of deep learning for image classification, image classification was highly reliant on some standard descriptors (image features).

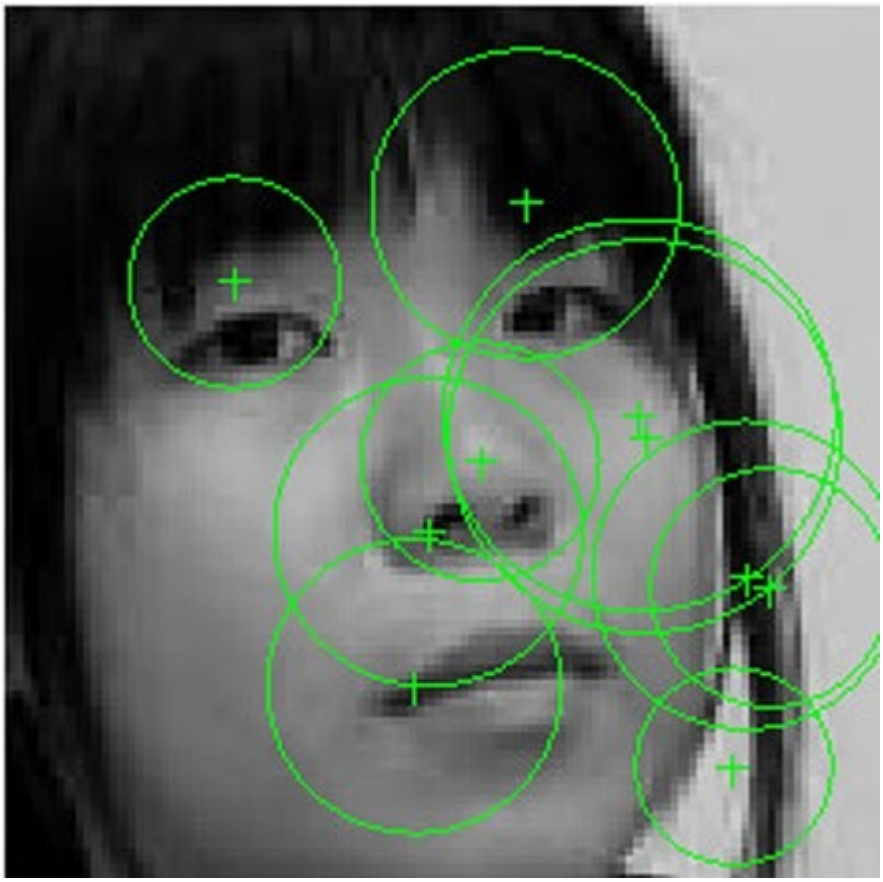
Descriptors are small “interesting”, descriptive or informative patches in images. Several image processing algorithms, such as edge detection, corner detection or threshold segmentation may be involved in this step. As many features as practicable are extracted from images and these features form a definition (known as a bag-of-words) of each object class. At the deployment stage, these definitions are searched for in other images. If a significant number of features from one bag-of-words are in another image, the image is classified as containing that specific object (i.e. chair, horse, etc.).

The difficulty with this traditional approach is that it is necessary to choose which features are important in each given image. As the number of classes to classify increases, feature extraction becomes more and more cumbersome. It is up to the computer vision engineer’s judgment and a long trial and error process to decide which features best describe different classes of objects. Moreover, each feature definition requires dealing with a plethora of parameters, all of which must be fine-tuned by the computer vision engineer.

Source: https://arxiv.org/abs/1910.13796

Traditional Image Classification

The face image below shows a traditional descriptor known as SURF descriptor. The SURF descriptor describes blobs in the image that can be used to classify the image. In the given image you can see there are detected features around the eyes, the nose, the mouth, the chin and the jaw. Aren’t these face features what we use to recognize faces?

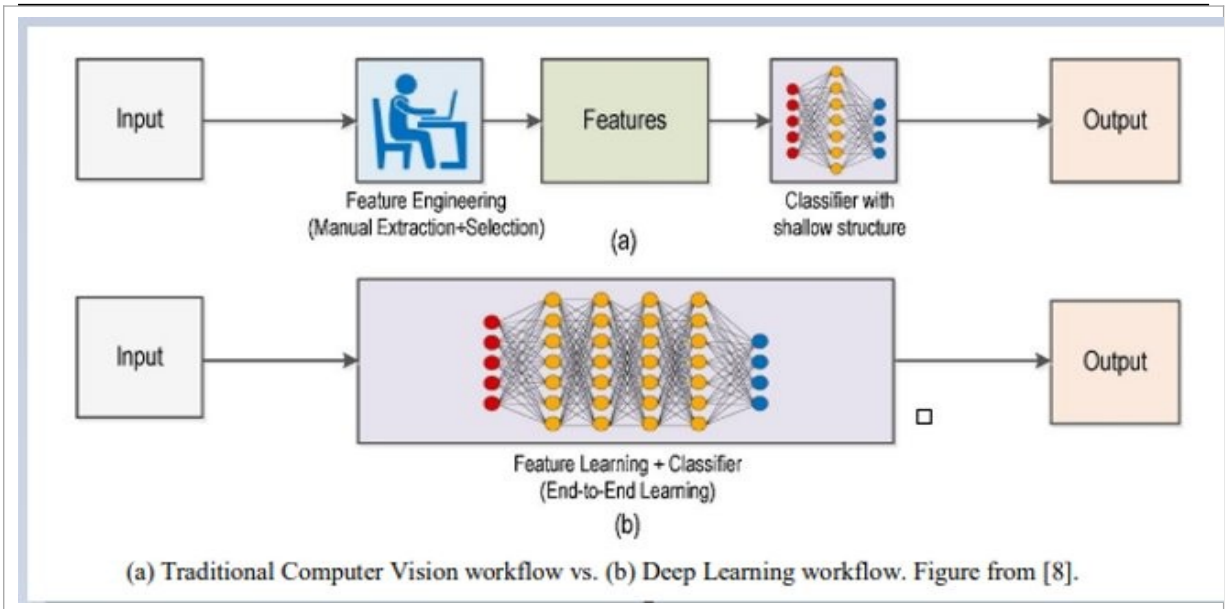


Source: https://www.flickr.com/photos/daniel-sikar/49800412293

Traditional ML-based Image Classification Vs Deep Learning Image Classification

The following image shows the differences between the traditional ML workflow and the deep learning workflow for image classification. In the case of deep learning, the image is simply given as input to the model while in traditional ML a human makes several decisions on parameters used to extract features and/or how to combine them. In the case of SURF, it is unlikely you will have good results for your application domain if you don't experiment with the SURF feature extraction parameters.

There are different parameters such as threshold, number of octaves, number of layers which must be decided by a human expert based on experimentation or to get good results. A human does not need to be involved at any level for feature extraction in the case of deep learning.



(a) Traditional Computer Vision workflow vs. (b) Deep Learning workflow. Figure from [8].

Image Classification in Remote Sensing

The traditional approaches for remote sensing image analysis are based on low-level and mid-level feature extraction and representation. These techniques have shown good performance by using different feature combinations and machine learning approaches. These earlier approaches have used small scale image dataset. The recent trends for remote sensing image analysis are shifted to the use of machine learning models which help to process a huge amount of dataset in accurate and rapid way.

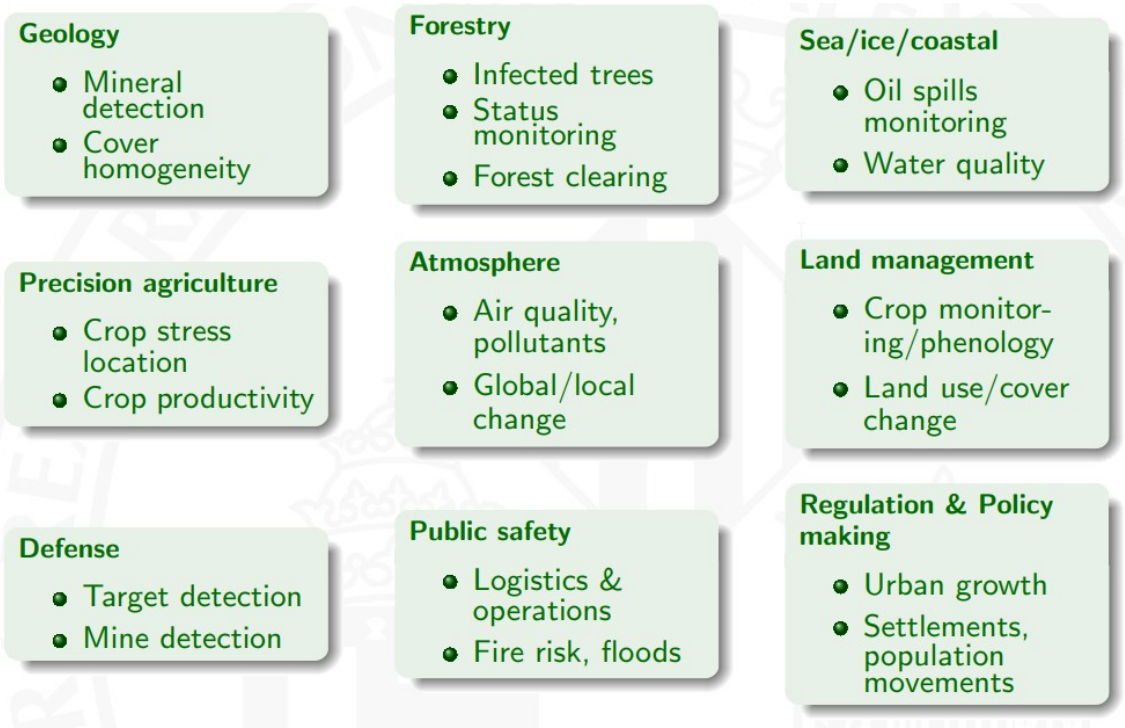
Satellite image classification is a multilevel process that starts from extracting features from images to classifying them into categories. Image classification is a step-by-step process that starts with designing a scheme for classification of desired images.

After that, the images are preprocessed which includes image enhancement, and scaling. Thereafter, the algorithm is applied on the images to get the desired classification. Corrective actions are made after applying algorithms, which is also called post processing.

In the final phase the accuracy of this classification is assessed. It is possible to apply both supervised and unsupervised machine learning methods for image classification in remote sensing.

Source: Remote Sensing Image Classification: A Comprehensive Review and Applications by Maryam Mehmood et al (March,2022), https://doi.org/10.1155/2022/5880959

Application Domains of ML in Remote Sensing Data



Source: https://artemisat2.ulpgc.es/wp-content/uploads/2019/11/MachineLearningRemoteSensing-GustauCamps-Valls.pdf

Machine Learning Tasks in Remote Sensing

Machine learning has been used for various tasks in remote sensing.

As you discussed in Modules 1 – 3 of this course, a remote-sensing dataset may contain data from visible and invisible bands. An ML algorithm is expected to take some of or all of the available bands or extracted features and output classifications for each pixel or regions in the input area.

Generally, there are two approaches to applying ML to remote sensing. The first one is pixel-based classification and the second one is object-based classification. In pixel-based classification, the spectral band features of a single pixel are used to assign it a label. Object-based, however, uses both spatial and spectral features to output classifications for regions in the input.

Semantic segmentation

In semantic segmentation task, each pixel is assigned to one pre-defined class and pixels of the same class are grouped together to one semantic segment.

ID	Color	Label	ID	C
1		Roads	2	
3		Trees	4	
5		Bare Soil	6	
7		Rails	8	

Source: https://uni-bonn.sciebo.de/s/kixZbCxTErCjW2Y

Object-based Image Classification

Object-based classification uses both spectral and spatial information for classification. The process involves categorization of pixels based on their spectral characteristics, shape, texture and spatial relationship with the surrounding pixels. Object-based classification methods were developed relatively recently compared to traditional pixel-based classification techniques. While pixel-based classification is based solely on the spectral information in each pixel, object-based classification is based on information from a set of similar pixels called objects or image objects. Object-based classification is a two-step process, first the image is segmented or broken into discrete objects or features and then each object is classified. For example, in object-based classification, we might be interested in classifying buildings and football fields from remote sensing images.

General Steps to Process EO Data

In Modules 1 – 3, you studied that EO data contains multiple bands. In section 4.2 of this module, we discussed the key steps in ML. In this section, we will discuss general steps that can be used to select and process EO data. Please note these are general steps and some of them may not be relevant to some tasks.

1. Select best available image according to pre-defined thresholds
2. Select best features (channels, spatial) that describe the problem (classification, retrieval)
3. Remove noise and distortions due to clouds, acquisition (sun glint) or transmission (vertical stripes)
4. Use band operations to create band ratios, and indices through linear/non-linear combinations of existing bands
5. Pass the input (along with the extracted features) and use an ML algorithm to assign semantic classes to objects (pixels, patches, regions) in the scene

All the steps above are in some way related to feature extraction from EO data. Let's see why feature extraction is important for EO data.

Importance of Feature Extraction from EO Data

Extracting features from remote sensing images is essential to:

- ➔ Make image processing algorithms more robust (to noise)
- ➔ Visualize data characteristics
- ➔ Understand the underlying physical relations

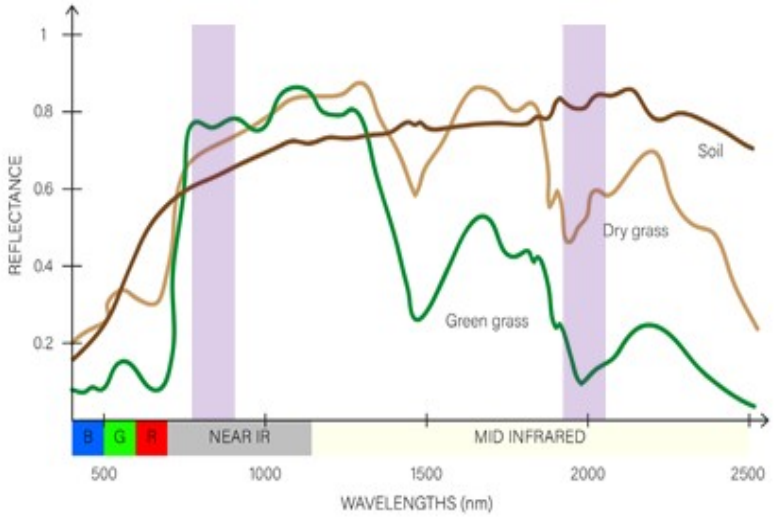
Specific feature extraction techniques will be discussed in subsequent modules. We will now discuss how the different types of Machine Learning can be applied to EO data at a high level.

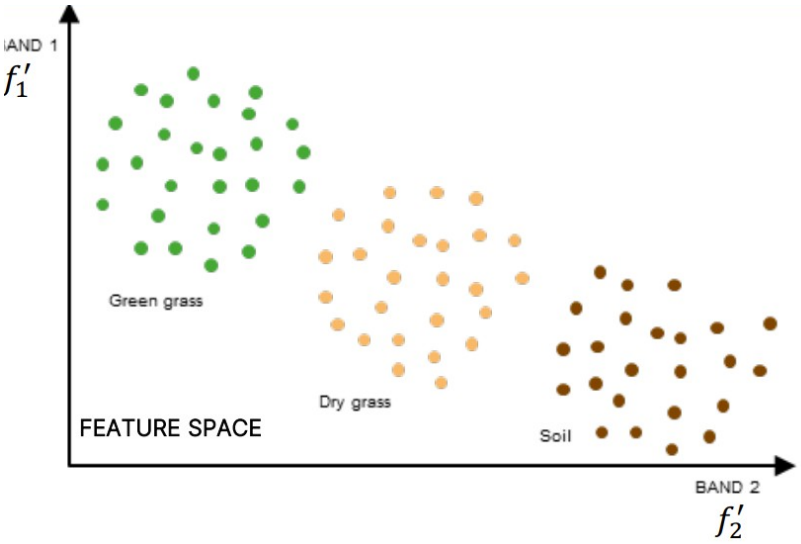
Supervised Learning for Image Classification in Remote Sensing

Supervised learning is based on labeled training data consisting of a set of training samples (input-output pairs), it generally has two types: ***classification*** and ***regression.***

- **Classification** is used when the output variables are categorical (i.e., with 2 or more classes). It helps to map the data from a certain dimensional feature space (number of bands) to one-dimensional space of the classes. The different classes are represented as integers.

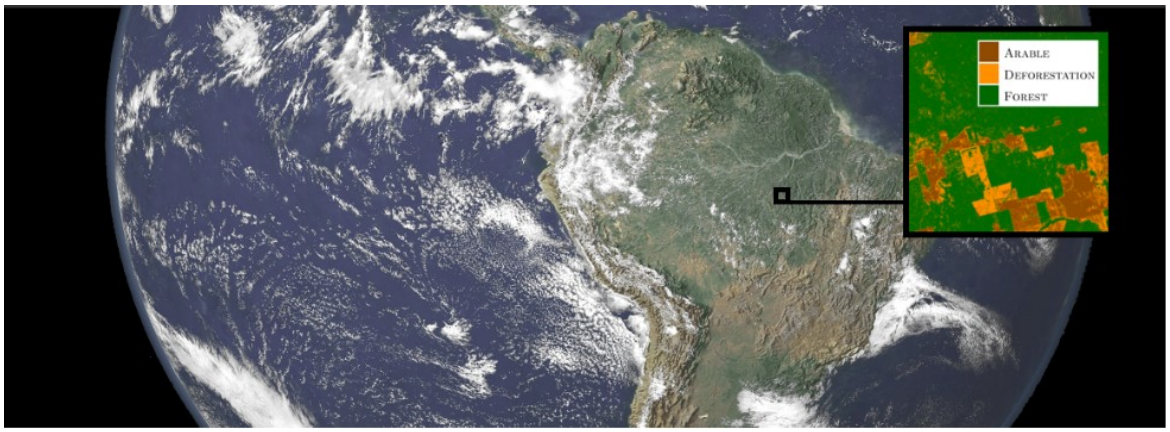
- For example, if you need to separate two grass classes (dry and green) from information contained in two bands (refer to figure below), you can pass the band information as input to a machine learning model and produce either a green grass (class 1) or a dry grass (class 2) as output.



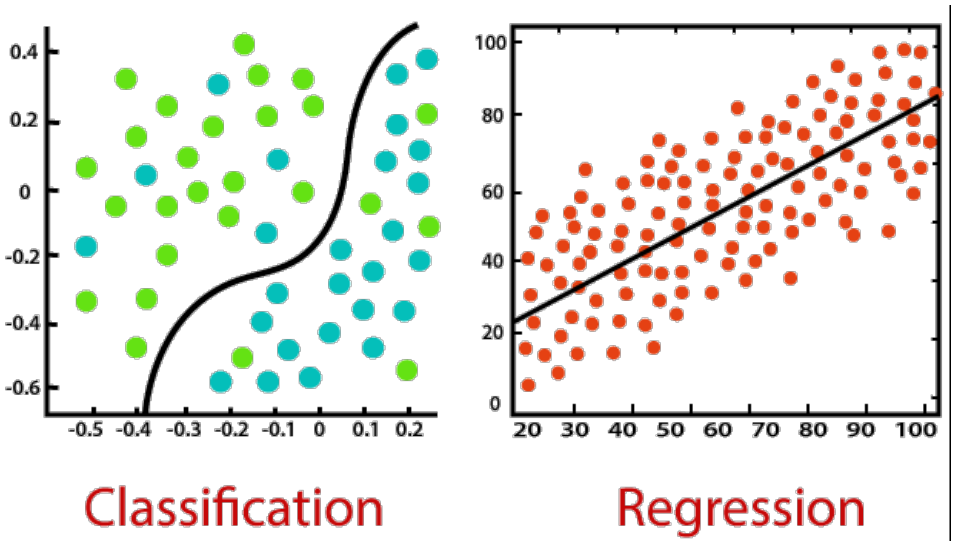


Source: Prof. Gabriele Cavallaro, School of Engineering and Natural Sciences, University of Iceland, Lecture 2 from the Course REI506M “Applications of Data Science in Remote Sensing” (Fall 2022), p.44

Supervised classification has been applied to assign semantic labels to areas in a remote sensing dataset.



- **Regression** is defined as a statistical method used to establish the relationship between a dependent variable and one or more independent variables. In machine learning, Regression is a supervised learning technique since the dataset contains input features and the corresponding labels as well. Regression is used when the number of possible values for the dependent (target) variable are not countable. For example, if the target variable is a rational number there are infinitely many possibilities for what the target variable can be. In remote sensing regression can be used to estimate continuous quantitative metrics such as crop yield and building energy needs. Generally, regression has been applied to predict temperatures, stock prices, and credit scores. In the case of stock price prediction, for example, the target variable stock price can assume any positive value. Thus, we must apply regression methods to be able to output stock price that can assume any value in the valid range.



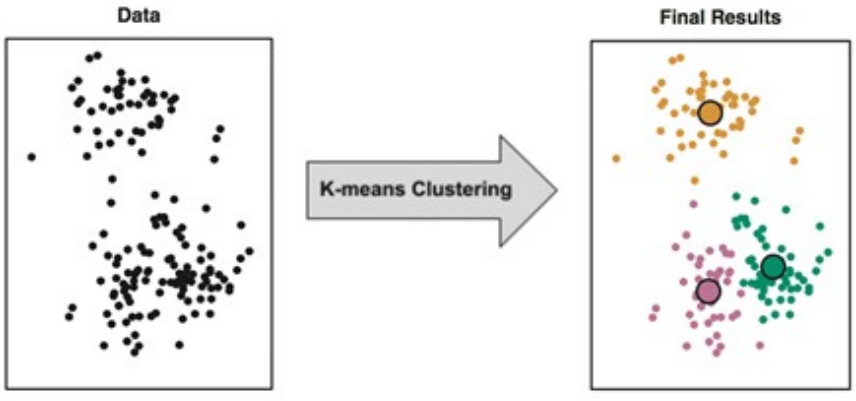
Source : https://www.simplilearn.com/regression-vs-classification-in-machine-learning-article

This part was a little bit more complex, wasn't it? We know that it's getting more abstract. To help you understand the concepts of regression and classification, please watch the following video:

https://www.youtube.com/watch?v=TJveOYsK6MY (time duration 2.47)

Unsupervised learning

In this case, data without labels is given. The classification algorithm is expected to find "hidden" patterns. Unsupervised learning methods generate clusters without semantic meaning. It is the task of the human to make sense of these clusters. Its classical frameworks include **clustering and dimensionality reduction.** Dimensionality reduction refers to techniques that reduce the number of input variables in a dataset.



Source: Remote Sensing Image Classification: A Comprehensive Review and Applications by Maryam Mehmood et al (March,20220), https://doi.org/10.1155/2022/5880959

Please watch this video to learn more on reinforcement learning:

https://www.youtube.com/watch?v=2xATEwcRpy8 (time duration 2:27)

Deep Learning in Remote Sensing

- Many deep learning approaches have found applications in remote sensing
- Some of the most used deep learning models in remote sensing are convolutional neural networks,
- One popular application of deep learning in remote sensing is target recognition. Target recognition is the identification of objects such as ships, cars, and planes that exist in an image. In the context of remote sensing, target recognition is very challenging.
- Target recognition in remote sensing is hard because
    - The objects appear much smaller in an RS video
    - The objects appear in a complex neighboring environment
- Deep learning methods can be used for pixel wise classification or regional classification. Moreover, deep learning methods can work with spatial features, spectral features or both.

Exercise materials and tasks

Quiz questions

As recap of this session, please complete below quiz on machine learning for image classification subject:

1. Which of the following is not an example of image classification?
    a. **Using ML to identify the edges of cats in an image**
    b. Using ML to Recognizing faces from images
    c. Using ML to Identify different bird species from their images
    d. **Extracting features (descriptors) from images**

2. Which of the following is true about the traditional image classification approaches and deep learning-based image classification approaches?
    a. **Traditional ML image classification approaches require heavy involvement of a human expert to get good performance**
    b. **Deep Learning-based image classification can work well without explicit human specified features**
    c. Even if deep learning image classification can work well without human intervention the state-of-the-art image classification systems use traditional ML image classification methods
    d. For a deep learning image classification approach the input images must not be pre-processed in any way.

3. Which of the following requires a regression approach?
    a. Classifying images of domestic animals

b. Recognizing a speaker from a sound file
c. Determining the price of a house based on its location, and number of bathrooms
d. Showing a customized advertisement to a user on a website

4. Which of the following is not true about Deep Learning?
   a. Supervised deep learning approaches can extract good features themselves without the help of a human
   b. Deep learning approaches can work well on unstructured data
   c. **An Artificial Neural Network that contains two layers of three neurons each can be considered deep learning**
   d. **There are some deep learning approaches that do not use Artificial Neural Networks**

5. If you are given some satellite images and are asked to group parts of each image into three groups, what kind of learning would you have to use?
   a. Supervised
   b. Unsupervised
   c. Reinforcement

4.5 Introduction to the Programming Language Python and the application Jupyter Notebooks

Reading material

Introduction to the Programming Language Python and the application Jupyter Notebooks

Get ready, this is getting interesting!

Since this is a course on machine learning, we will start this session with the programming languages Python and the application JupyterLab. Python's expansive library of open-source data analysis tools, web frameworks, and testing instruments make its ecosystem one of the largest out of any programming community. Python is an accessible language for new programmers because the community provides many introductory resources.

You will use Python and JupyterLab as one of the technologies that enables interactive computing and you will also use them later in the course for developing your own project in Module 10 and Module 11.

Why is this relevant for you in the frame of this course? Well, next steps to be performed for Earth Observation will need the Python programming language and JupyterLab because they are mostly cloud based. During the course, we are mainly going to use Sentinel Hub as an EO Processing Platform. Before we register you on Sentinel Hub, it is important that you get introduced to Python and Jupyter Notebook.

Ready? Then first read the following background material on Python, Jupyter Notebook, and then download the instruction file to install these programmes.

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. It is a dynamic programming language which supports several different programming paradigms:

- Procedural programming
- Object oriented programming
- Functional programming

What is it used for?

Python is used for…

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.

- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Python can be treated in a procedural, an object-oriented or a functional way.

Python Syntax compared to other programming languages

- Python was designed for readability and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope, such as the scope of loops, functions and classes. Other programming languages often use curly brackets for this purpose.

Introduction to Jupyter Notebook and JupyterLab

Project Jupyter

Project Jupyter is a project and community whose goal is to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages. It is possible to run it over 70 different programming languages. The project was born out of the IPython Project in 2014. The name “Jupyter” is a combination of the programming languages “Julia”, “Python”, and “R”.

Project Jupyter has developed and supported the interactive computing products Jupyter Notebook, JupyterHub, and JupyterLab:

1. **Jupyter Notebook**: A browser-based application that allows you to create and share documents (i.e., Jupyter Notebook files) that contain live code, equations, visualizations, and narrative text. The Jupyter Notebook file format is “.ipynb”, which is short for “interactive python notebook”.
2. **JupyterLab**: A browser-based application that allows you to access multiple Jupyter Notebook files as well as other code and data files. It is a new version of Jupyter Notebook that includes Notebook, text editor, console and a file explorer.
3. **JupyterHub**: A multi-person version of Jupyter Notebook and Lab that can be run on a server. It is an encapsulated environment for multiple users.

Link to Jupyter: https://jupyter.org/

Why Jupyter Notebook?

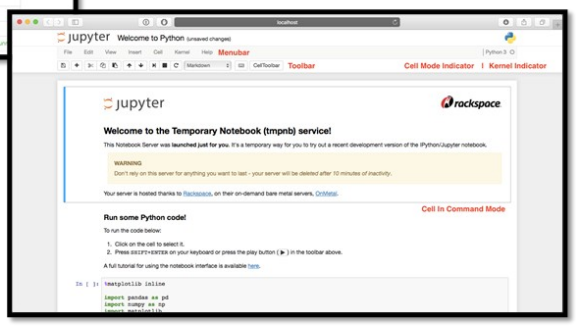
- It is easy to use
- It provides a single document that combines explanations with executable code and its output
- It is easy to share
- It allows to write and run code interactively
- An ideal way to provide:
  1. reproducible research results
  2. documentation of processes
  3. instructions
  4. tutorials and training materials

- Jupyter Notebook is very easy to use -it only consists of a file browser and an editor view.
- JupyterLab is the next generation of the Jupyter Notebook. It has a more complicated user interface, however, with more capabilities.
- Using JupyterLab, you can open several notebooks or files (e.g., HTML, Text, Markdowns, etc.) as tabs in the same window.
- JupyterLab offers within the same interface a file browser, consoles, terminals, text editors, Markdown editors, CSV editors, JSON editors, interactive maps, widgets, and so on

It uses the same server and file format as the classic Jupyter Notebook. So, it is fully compatible with the existing notebooks and kernels.

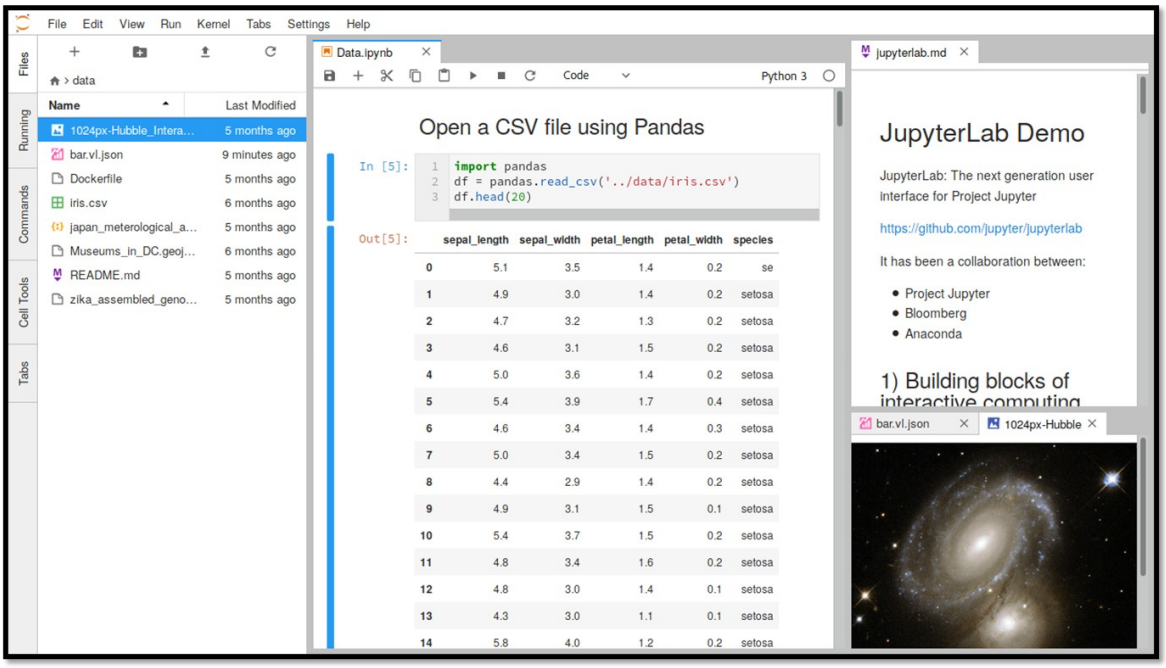
Jupyter Notebook interface

Jupyter Notebook Dashboard



Jupyter Notebook Editor

Source: https://jupyter-notebook.readthedocs.io/en/stable/ui_components.html



Source: https://jupyter-notebook.readthedocs.io/en/stable/ui_components.html

Exercise materials and tasks

Exercise Jupyter Notebook

Now that you know more about the programmes, and before you go any further, you will need to take time to install all required programs. Use the instruction file to download all required programs.

Once you have installed the programmes, download the folder with images you will use for the practical exercise below.

Then follow the next steps to perform the exercise.

Exercise on JupyterLab:
Download attached Jupyter files (*.ipynb).

Before we start with JupyetLab, you'll need to install the following packages for this exercise.

On your computer, search for "Windows PowerShell" and open it.

For this step, you will only need to copy/rewrite code line into PowerShell and run the program with enter.

"pip install numpy"

```
pip install numpy
```

When it is installed, you will see a message on the screen that the process is complete. After that, you still need to install 4 more in the same way as before.

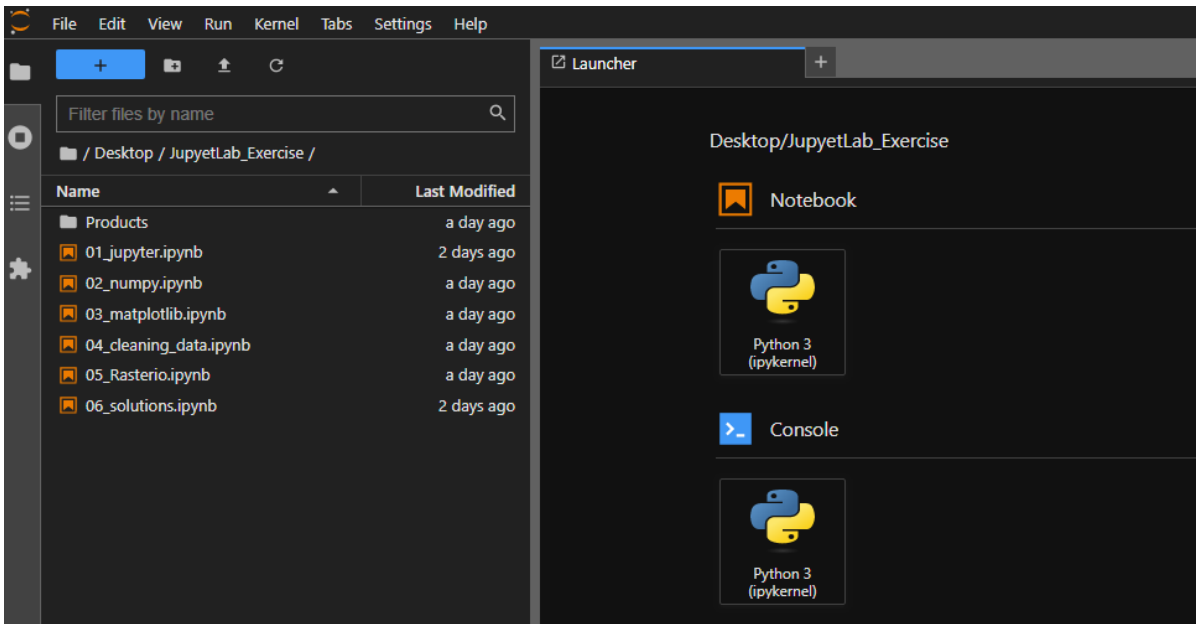
"pip install matplotlib"
"pip install pandas"
"pip install rasterio"
"pip install folium"

If you have any difficulties while installing, try to use following links:

- NumPy https://numpy.org/install/
- Matplotlib https://matplotlib.org/stable/users/installing/index.html
- Pandas https://pandas.pydata.orgc
- Rasterio https://rasterio.readthedocs.io/en/latest/index.html
- Folium https://pypi.org/project/folium/

You only need to install these packages once, they are stored on your PC. After installing the required packages, we can start with our introduction to the JupyterLab. Copy/rewrite code into Powershell and run it in with enter and wait for the JupyterLab to open in your browser.
"python -m jupyterlab"

In your JupyterLab on your left side, locate your downloaded map with files and open it.



In the exercises, you will be introduced to how to use Jupyter Notebooks as well as some Python libraries that are essential in ML.

Open file “01_jupyter.ipnyb” with double click and start following the instructions. After completing first file, open second one and start with the exercises. At the end of each file there is a small task to complete to determine the material learned. If you have trouble with completing task, you can use file “06_solution.ipynb” as a help (but first try without the help!).

Once you have finished the exercise, take a screenshot with the solution and post the four screenshots in the forum.

4.6 Field work and application exercise

Exercise materials and tasks

Field work and Application exercise

This exercise covers basics of data collection and the analysis of the collected data. The exercise will also allow students to explore selected field data collection techniques. This exercise will show students how to use cloud services for quick analysis without having to download data. For data analysis purposes, we will use the Sentinel Hub platform.
This exercise consists of three parts:

1. data collection in the field,
2. an analysis of the collected data, and
3. Application report summarizing your findings.

Please familiarize yourself with the report requirements prior to starting your field work (in the next slides).

Part 1: Fieldwork: data collection
To collect data in the field we suggest using the SW Maps mobile app. Before any further instructions, please download SW Maps on your smart device (SW Maps - GIS & Data Collector – Aplikacije v Googlu Play (google.com)).

The following video shows how the application is used and how you will use it to collect data https://www.youtube.com/watch?v=bNRmrhdmujU. You are now ready to start the fieldwork:

Individual fieldwork task:

- Please select an area close to your home. Please collect data using the SW Mapp application on your smart device.
- Using the application, please generate:
  - at least 5 reference points for 5 classes, remember that every reference point is made of coordinates (X, Y) and class name
  - one agriculture field (near your home) that consists of at least 1 acre in size (1 acre = 0.004 sq.km)
  - linear data for at least 2 classes

Note: depending on your location, define classes accordingly. For points, define classes such as lamppost, tree, parking lot… For linear classes use main street, ally, walking path, bicycle path, ….

Part 2: Data analysis:
After you have completed your data collection in the field, it's time for part 2 of the exercise – the data analysis.
As shown in the video on how to use SW Maps mobile app, you need to download your collected data. Please save it on your personal computer.

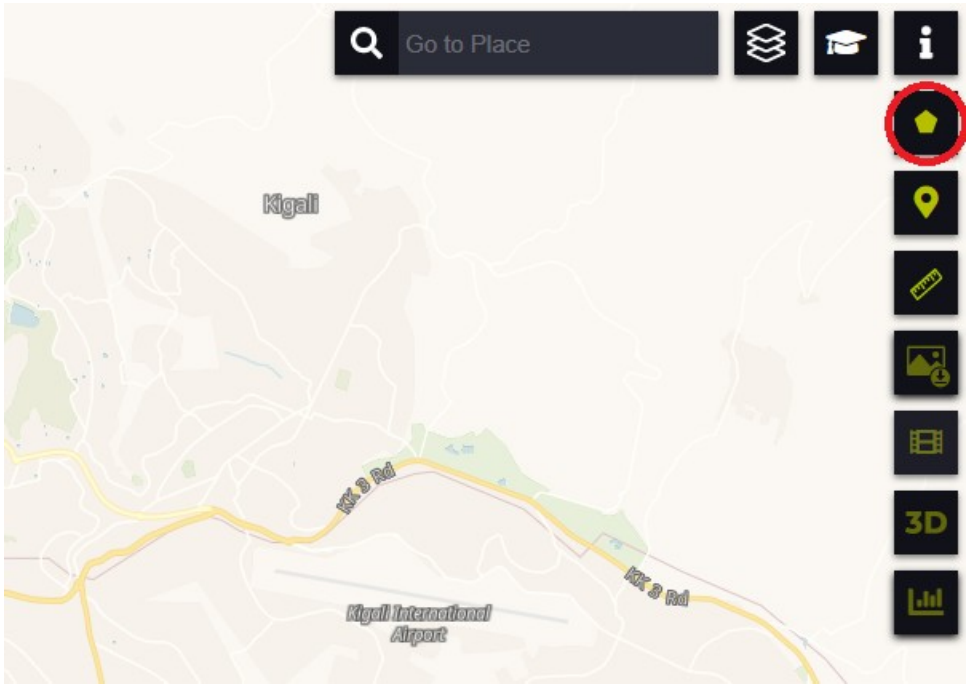
Thereafter, please apply the following steps for data analysis purposes:

Step 1:

- Use QGIS and open a new project
- Under Data Source Manager > Vector, add exported layers (*.shp) from the downloaded data from fieldwork
- Try to use Google map or OpenStreetMap to check your data (Hint: XYZ Tiles or WMS)
- Under layers locate your agriculture polygon and right click on it > Export > Save Feature As …
- Save it as GeoJSON in the same map in which you have saved the rest of your data.

Step 2:

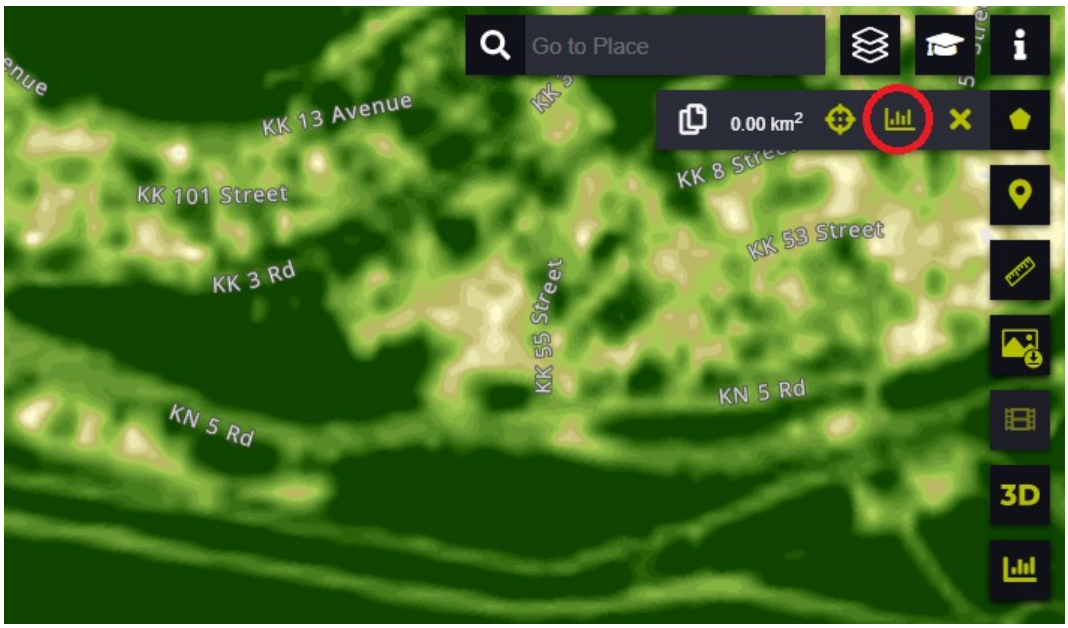
- Open Sentinel Hub's EO browser (link: https://apps.sentinel-hub.com/eo-browser/)
- Click on the Create an area of interest (picture below) > Upload a file to create an area of interest



- Upload the area GeoJSON file you previously saved in Step 1

Step 3:

- Search for Sentinel 2 data and find NDVI index
- Calculate the statistical info (refer to the picture below) of your area of interest for the last year. While calculating NDVI or any other index, try to lower max. cloud coverage!



Forum instructions

Generate report of the application exercise and share your results in the forum

Welcome to the forum of Module 4!

As a last step, please generate a report about your findings and share your results in the dedicated forum.

The report you shall post in the forum should contain the following information:

- Make a screenshot of data in QGIS
- Make a screenshot of the statistical info for NDVI
- Provide a brief description of the selected area
- Discuss obtained results of statistical info (examples: what they represent, why we have a curve, what happens when we don't lower max. cloud coverage, etc)
- Respond to the following questions:
    - Have you tried calculating statistical information of your area using different indices?
    - Which ones?
    - What can they tell us/show us?

Once you have finalised, please read the contributions of the other group members. Post at least one comment or question to another participant's contribution with the idea of exchanging experiences.

Please don't forget to answer any question you got in response to your post in the forum.

Additional resources and material

Before we move to the next module, we have listed below a few more resources with the aim of deepening your knowledge on the Module 4 topics. These are optional readings.

For more details (including sources of Images), please refer to the following links:

- Supervised and Unsupervised Learning in Machine Learning: https://www.youtube.com/watch?v=kE5QZ8G_78c&ab_channel=Simplilearn
- L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," in IBM Journal of Research and Development, vol. 3, no. 3, pp. 210-229, July 1959
- Sentinel-2 Satellite Imagery, http://geocento.com/satellite-imagery-gallery/sentinel-2/
- CORINE Land Cover - Copernicus Land Monitoring Service, https://land.copernicus.eu/pan-european/corine-land-cover
- https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/ (Understanding Machine Learning: From Theory to Algorithms c 2014 by Shai Shalev-Shwartz and Shai Ben-David Published 2014 by Cambridge University Press)

Do you want to learn more about Jupyter Notebook (JupyterLab)? Then please check out the following link:

- https://realpython.com/jupyter-notebook-introduction/

If you have a question about Python, it's a good idea to try the FAQ, which answers the most asked questions.

For more information about this topic, you can also listen to first 15 minutes of the following video: https://www.youtube.com/watch?v=5NHmxYkYoZg. For enthusiasts the whole video watch is also recommended.