



INTRODUCTION & OVERVIEW

(Module 1)

CONTENTS

- Introduction session
- Review of the course outline and expectations
- Introduction and overview of AI, Machine Learning and Deep learning
- Background
- Categories
- Challenges
- ML Pipeline

IMPORTANT INFORMATION

Professor Name: Ms. Ulya Sabeel

Email: usabeel@my.centennialcollege.ca

Office Hours: During class or by appointment through email

Zoom Link for Class:

Tuesday: <https://centennialcollege.zoom.us/j/94763713164?pwd=ZjdBanpJUThJUWIFNGVMaDBYVWZ3QT09>

Thursday: <https://centennialcollege.zoom.us/j/98519685519?pwd=TIhjRUZkY1RxbFNVS1M5OThqdjF6dz09>

Class Timings: Tuesdays (8:30 am - 11:20 am) and Thursdays (12:00 pm - 2:50 pm)

Course Duration: Mar 08, 2022 - April 21, 2022

Prerequisites: Basics of Python programming, Python, Anaconda, and Jupyter Notebook installed on a personal computer

TENTATIVE COURSE OUTLINE

LECTURE	DATE	TIME	TOPICS
1	Mar 08, 2022	8:30 am - 11:20 am EDT	Introduction to AI & ML, Background, Categories, Challenges, ML Pipeline
2	Mar 10, 2022	12:00 pm - 2:50 pm EDT	Building a ML Classifier, Pre-processing, Feature Selection, Dimension Reduction, Normalization, Training, Evaluation, Hyperparameter Optimization
3	Mar 15, 2022	8:30 am - 11:20 am EDT	Basics of Logistic Regression, Overview of Shallow Neural Networks
4	Mar 17, 2022	12:00 pm - 2:50 pm EDT	Introduction to Deep Learning (DL), Why DL?, Key Concepts, Introduction to Keras and Tensorflow
5	Mar 22, 2022	8:30 am - 11:20 am EDT	Building a Deep Neural Network step by step with case studies, Introduction to TensorBoard
6	Mar 24, 2022	12:00 pm - 2:50 pm EDT	Deep Learning for Computer Vision, Introduction to Convolutional Neural Network
7	Mar 29, 2022	8:30 am - 11:20 am EDT	Introduction To Recurrent Neural Networks (RNN), Case study time series forecasting
8	Mar 31, 2022	12:00 pm - 2:50 pm EDT	Mid-Term test (Test #1)

TENTATIVE COURSE OUTLINE CONTD.

LECTURE	DATE	TIME	TOPICS
9	Apr 05, 2022	8:30 am - 11:20 am EDT	Introduction to Deep Autoencoders (DAE), Anomaly detection case study
10	Apr 07, 2022	12:00 pm - 2:50 pm EDT	Introduction to Deep Generative Modeling, Generative Adversarial Networks and its variants
11	Apr 12, 2022	8:30 am - 11:20 am EDT	Introduction to Deep Reinforcement Learning
12	Apr 14, 2022	12:00 pm - 2:50 pm EDT	AI bias and fairness, Types, Identify, Evaluate
13	Apr 19, 2022	8:30 am - 11:20 am EDT	Key concepts review, DL Limitations, Future of DL, Final words
14	Apr 21, 2022	12:00 pm - 2:50 pm EDT	Final Exam (Test # 2)

TEXT & OTHER INSTRUCTIONAL MATERIALS

Textbook(s) referred for this course:

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by Aurelien Geron, first edition Published by O'Reilly Media, Inc. 2017, ISBN: 978-1-491-96229-9
2. Deep Learning with Python by François Chollet, published by Manning Publications Co., 2018, ISBN: 978-1-61729-443-3

Some online resource(s) referred for this course:

1. <https://www.anaconda.com/>
2. <https://www.tensorflow.org/>
3. <https://www.tensorflow.org/tutorials/index.html>
4. <https://scikit-learn.org/stable/>
5. <https://keras.io/>
6. <https://pandas.pydata.org/>
7. <https://www.deeplearningbook.org/>

NOTE: Additional reading materials may be recommended/assigned during this course.

EVALUATION SCHEME (Tentative)

- ⇒ **Quizzes** – Total 2 Quizzes, 15% each. (Quiz 1: Mar 15, Quiz 2: Apr 07)
- ⇒ **Lab Assignments** – Total 2 Assignments, 12.5% each (Assignment 1: Mar 22, Assignment 2: Apr 14)
- ⇒ **Mid-Term Test** – AI and Machine Learning (ML) concepts, Logistic Regression, Shallow Neural Network, Deep Neural Network, Convolutional Neural Network (Mar 31, 12:00 pm - 2:50 pm)
- ⇒ **Final Exam** – Recurrent Neural Network, Deep Autoencoder, Deep Generative Modeling, Deep reinforcement Learning, AI bias and Fairness (Apr 21, 12:00 pm - 2:50 pm)

Evaluation Name	Weight/100
Quizzes	30
Individual Project Assignments	25
Mid-Term Test	20
Final Exam	25
Total	100%

WHAT IS EXPECTED?



NOTE: Plagiarized or late submissions will **NOT** be marked and will receive a grade of **ZERO**.

ARTIFICIAL INTELLIGENCE

- 1950- Alan Turing “Father of Computer Science” → “Computer Machinery and Intelligence”
- *Can machines think?*
- Turing Test → Distinguish between a computer or human text response
- Stuart Russel and Peter Norvig → “Artificial Intelligence: A Modern Approach”
- Differentiate computers based on rationality, thinking versus acting
- *Human approach:*
 - ✓ Systems that think like humans
 - ✓ Systems that act like humans
- *Ideal approach:*
 - ✓ Systems that think rationally
 - ✓ Systems that act rationally

DEFINITION OF ARTIFICIAL INTELLIGENCE

- **John McCarthy** → Father of AI
- “The science and engineering of making intelligent machines, especially intelligent computer programs”. - *John McCarthy*

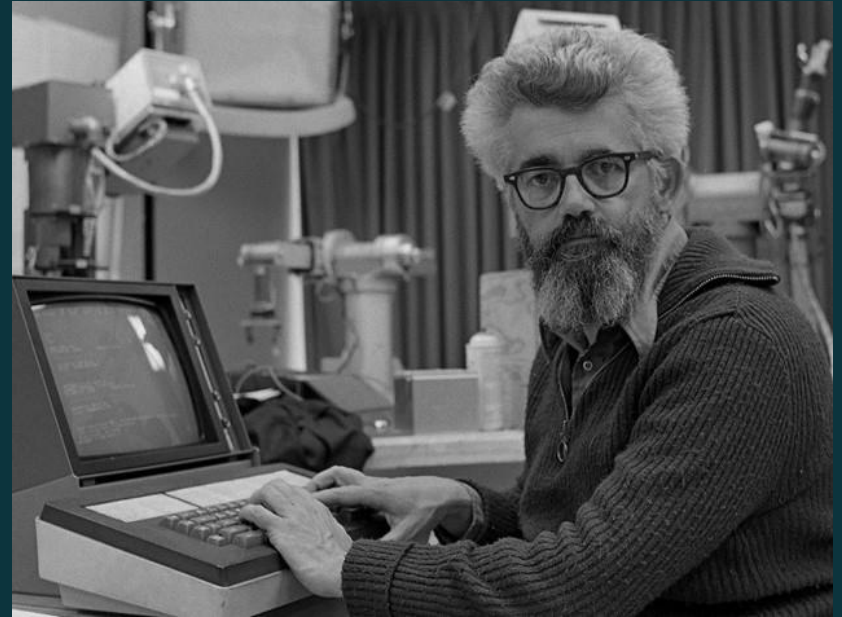


Image source: <https://becominghuman.ai/introduction-to-artificial-intelligence-5fba0148ec99>

OTHER DEFINITIONS OF ARTIFICIAL INTELLIGENCE

- AI is "the study of agents that receive percepts from the environment and perform actions." (Russel and Norvig)
- "Algorithms enabled by constraints, exposed by representations that support models targeted at loops that tie thinking, perception and action together." (Patrick Winston, the Ford professor of artificial intelligence and computer science at MIT)
- "AI is a computer system able to perform tasks that ordinarily require human intelligence... Many of these artificial intelligence systems are powered by machine learning, some of them are powered by deep learning and some of them are powered by very boring things like rules." (DataRobot CEO Jeremy Achin, 2017)

INTRODUCING ARTIFICIAL INTELLIGENCE

- The effort to automate the intellectual tasks using a machine which are normally performed by humans (Deep Learning with Python by François Chollet)
- Augmented Intelligence → Extend human potential (not replace it) to perform difficult tasks



Human Intelligence



Artificial Intelligence



AI Solution

- Blend of multiple fields of study
- Computer Science and Electric engineering → implement in software and hardware
- Statistics and math → algorithms and performance evaluation
- Psychology → understanding the working based on human brain

CATEGORIES OF ARTIFICIAL INTELLIGENCE

- Based on strength, breadth and applications



Narrow AI

- Weak AI, Applied AI
- Simulation of Human Intelligence
- Operate under constraints
- Perform specific tasks not learn new ones
- Decisions based on train data, programmed algorithms
- Google search, Siri, Alexa, Google Assistant, Self-driving cars, image recognition software



Artificial General Intelligence (AGI)

- Strong AI, Generalized AI
- Machine with general intelligence like a human being
- Learn from experience
- Wide variety of independent and unrelated tasks
- Generalized Intelligence → solve new problems
- Robots from Westworld
- Data from Star Trek: The Next Generation



Super AI

- Conscious AI
- Human-level consciousness → self-aware
- Not created yet! Difficult to measure consciousness



Image source: <https://screenrant.com/star-trek-data-storylines-unsolved/>

SOME APPLICATIONS OF AI

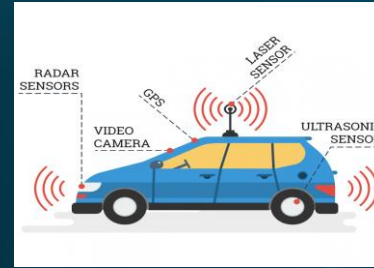
[1] Speech Recognition: Alexa, Siri



[2] Customer service: Chatbots (E-commerce sites)



[3] Computer Vision: Self-driving cars



[4] Computer Vision: Medical Imaging



[5] Recommendation Systems: Online shopping



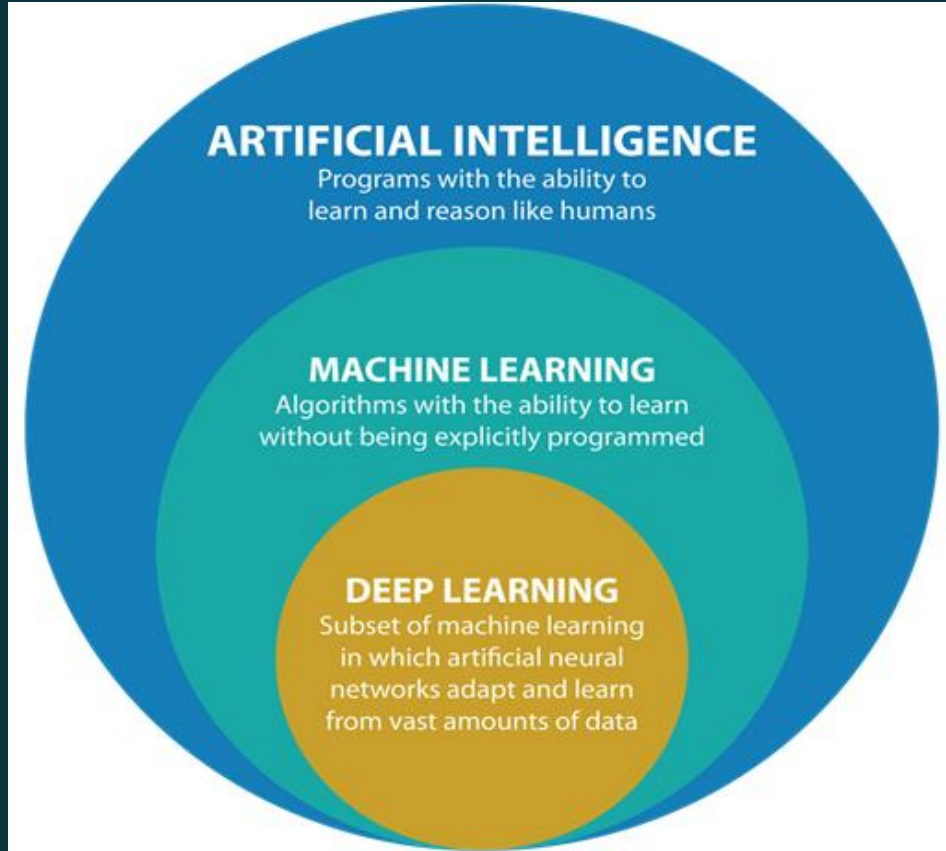
[6] AI Stock Trading



[7] Advanced Security Systems : Network security, Credit card Fraud detection, spam detection



AI, MACHINE LEARNING & DEEP LEARNING



COMPARISON

Data Science	Machine Learning	Deep Learning	Artificial Intelligence
<ul style="list-style-type: none">• Need of entire analytics universe• Branch that deals with data• Different operations related to data i.e.<ul style="list-style-type: none">▪ Data Gathering▪ Data Cleaning▪ Data Subsetting▪ Data Manipulation▪ Data Insights [Data Mining]	<ul style="list-style-type: none">• Combination of Machine and Data Science• Machines utilize Data Science techniques to learn about the data hence called as Machine Learning• Model Building, Model Evaluation and Validation• 3 Types:<ul style="list-style-type: none">▪ Unsupervised Learning▪ Reinforcement Learning▪ Supervised Learning• Most popular tools are Python, R and SAS	<ul style="list-style-type: none">• Specific branch of Machine Learning that deals with different flavours of Neural Network• Examples<ul style="list-style-type: none">▪ Simple Neural Network▪ Convolutional Neural Network▪ Recurrent Neural Network▪ Long Short Term Memory• Mainly utilized in..<ul style="list-style-type: none">▪ Object detection in Image and Video▪ Speech Recognition▪ Natural Language Processing and Understandings	<ul style="list-style-type: none">• Big Umbrella• Empowering machines to take decisions on their own• As the name suggest imparting humans' natural intelligence in machines• Thus machines have ability to understand and react according to the situation

MACHINE LEARNING (ML)

- *Machine Learning* is a subset of artificial intelligence
- “*Shallow Learning*” → only 1-2 layered representations
- Allows the machines to learn and make predictions based on the knowledge gained from train data
- learn useful representations of the input data to get closer to the expected output
- **For example**, predict the expected weight of a person based on its height as shown in the figure
- Draw a simple line to predict the weight (W) based on the height (H). For example, $W = H - 100$
- main aim → minimize the error between the predicted value and actual value
- Straight line fits through all points → minimizing loss/error
- Improve the model → add more data and features (e.g. Gender)

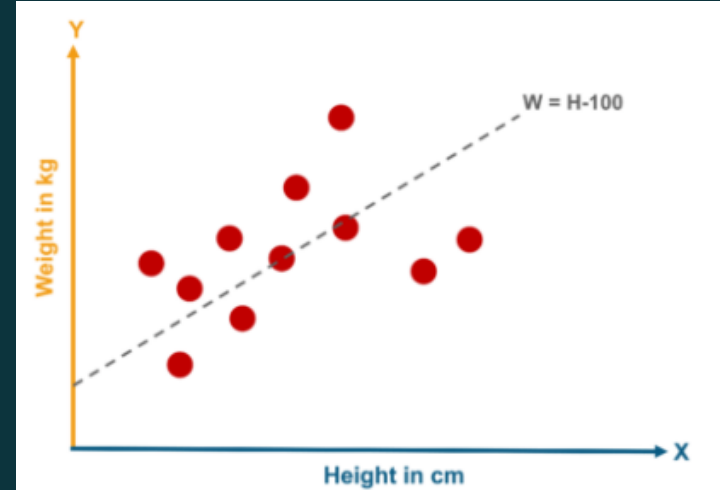


Image Source : <https://www.edureka.co/blog/ai-vs-machine-learning-vs-deep-learning/>

DEEP LEARNING (DL)

- DL is a subfield of machine learning
- learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations
- “Deep” → successive layers of representations
- *layered representations learning*
- *hierarchical representations learning*
- For example, a model which recognizes cat and dog images
- When using ML, we manually define the features of the animal such as mouth, nose, eyes, whiskers, tail
- Deep Learning automatically finds out the features which are important for classification

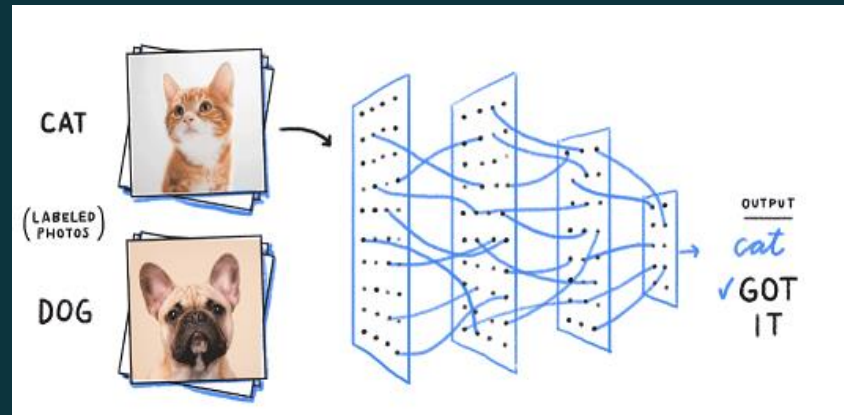
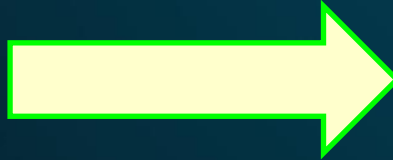


Image Source : <https://www.edureka.co/blog/ai-vs-machine-learning-vs-deep-learning/>

HOW DOES AI LEARN?



Provide with the capability to analyze examples and create machine learning/ deep learning models based on certain inputs to give the desired outputs.

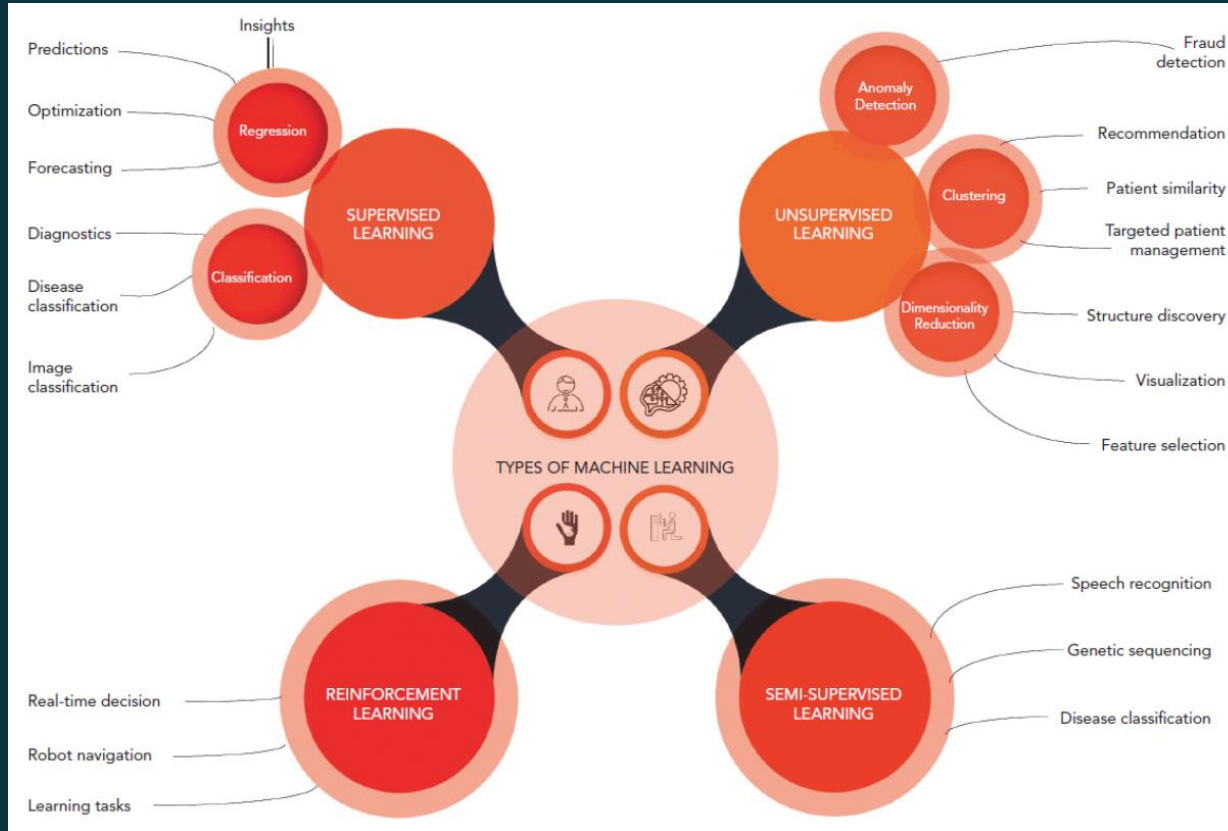
Supervised Learning

Unsupervised Learning

Semi-supervised Learning

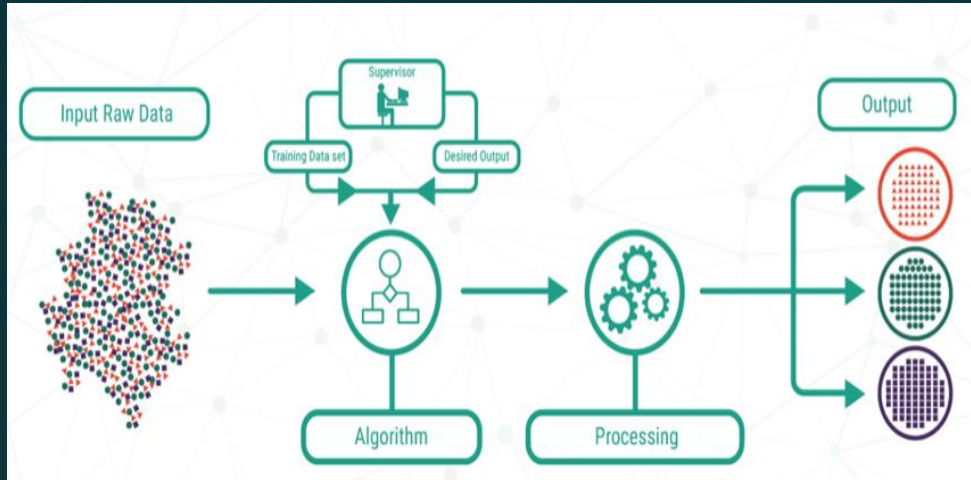
Reinforcement Learning

TYPES OF ML/DL TECHNIQUES



SUPERVISED LEARNING

- The training set you feed to the algorithm includes the desired solutions, called labels

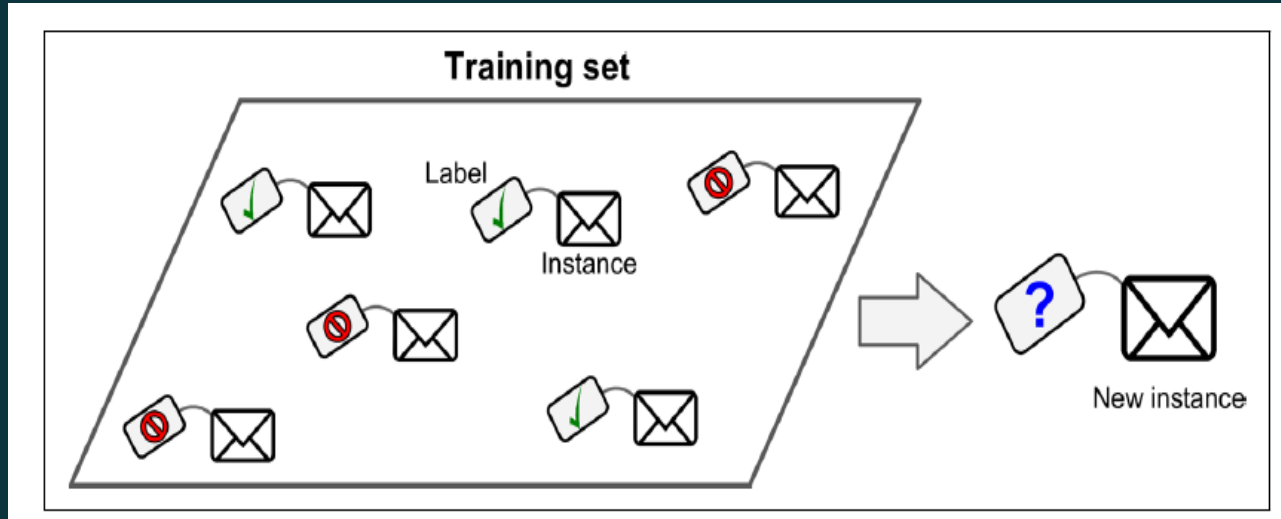


Most common supervised ML algorithms:

- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks

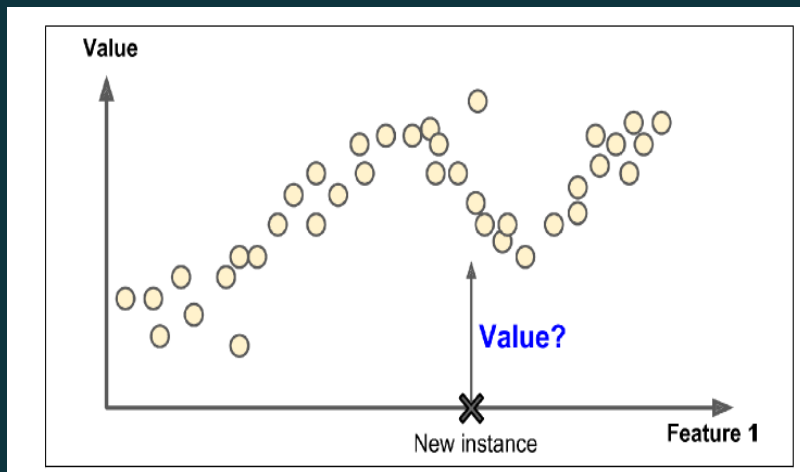
SUPERVISED LEARNING- CLASSIFICATION

- A labeled training set for spam *classification* (an example of supervised learning)
- **Example:** trained with many example emails along with their class (spam or no-spam)→learn how to classify new emails



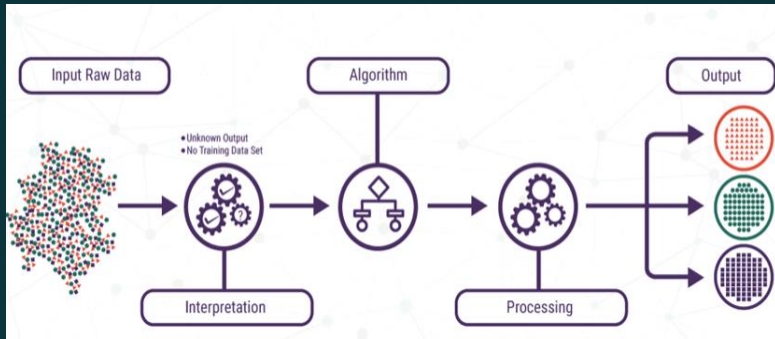
SUPERVISED LEARNING- REGRESSION

- A **regression** problem: predict a value, given an input feature (there are usually multiple input features, and sometimes multiple output values)
- **For example**, predict a target numeric value, such as the price of a car, given a set of features (mileage, age, brand, etc.) called predictors
- To train the system, you need to give it many examples of cars, including both their predictors and their labels (i.e., their prices).



UNSUPERVISED LEARNING

- The training data is unlabeled, system tries to learn without a teacher.



Most common unsupervised ML algorithms:

- **Clustering**

- K-Means
- Density-based spatial clustering of applications with noise (DBSCAN)
- Hierarchical Cluster Analysis (HCA)

- **Visualization and dimensionality reduction**

- Principal Component Analysis (PCA)
- Kernel PCA
- Locally Linear Embedding (LLE)
- t-Distributed Stochastic Neighbor Embedding (t-SNE)

- **Anomaly detection and novelty detection**

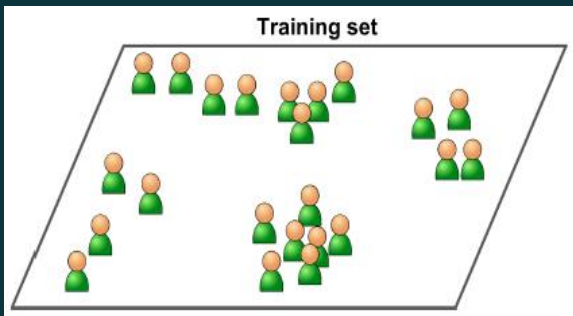
- One-class SVM
- Isolation Forest
- Autoencoder

- **Association rule learning**

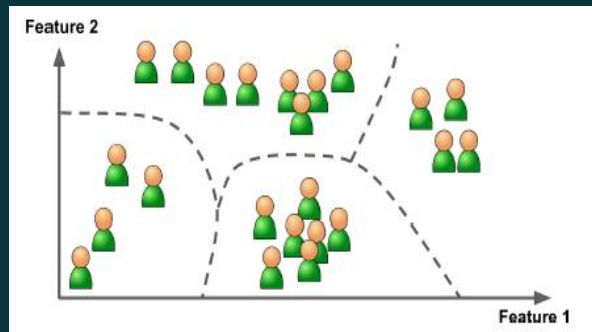
- Apriori
- Eclat

UNSUPERVISED LEARNING-CLUSTERING

- Identifying a blog's visitors
- run a clustering algorithm to try to detect groups of similar visitors
- Finds connections without an annotator
- For example, it might notice that 40% of your visitors are males who love comic books and generally read your blog in the evening, while 20% are young sci-fi lovers who visit during the weekends
- *Hierarchical clustering* → subdivide each group into smaller subgroups



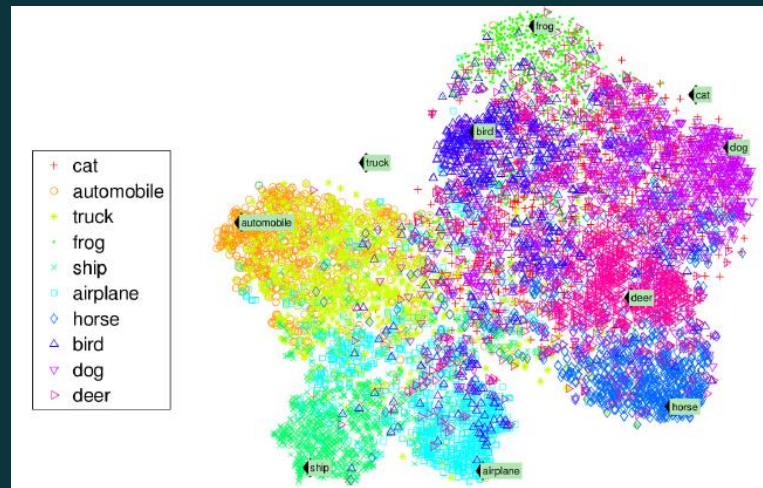
An unlabeled training set for unsupervised learning



Clustering

UNSUPERVISED LEARNING- VISUALIZATION

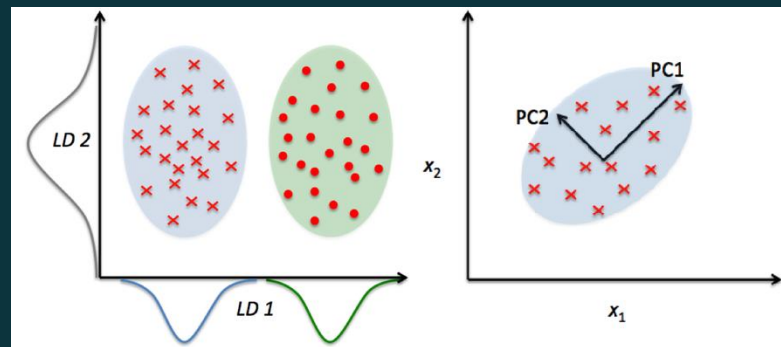
- feed them complex and unlabeled data as input
- output a 2D or 3D representation of your data that can easily be plotted
- preserve as much structure as they can → separate clusters → no overlapping
- Understand how the data is organized → identify unsuspected patterns
- Notice horses are close to deer but far from birds



Example of a t-Distributed Stochastic Neighbor Embedding (t-SNE) visualization highlighting semantic clusters

UNSUPERVISED LEARNING- DIMENSION REDUCTION

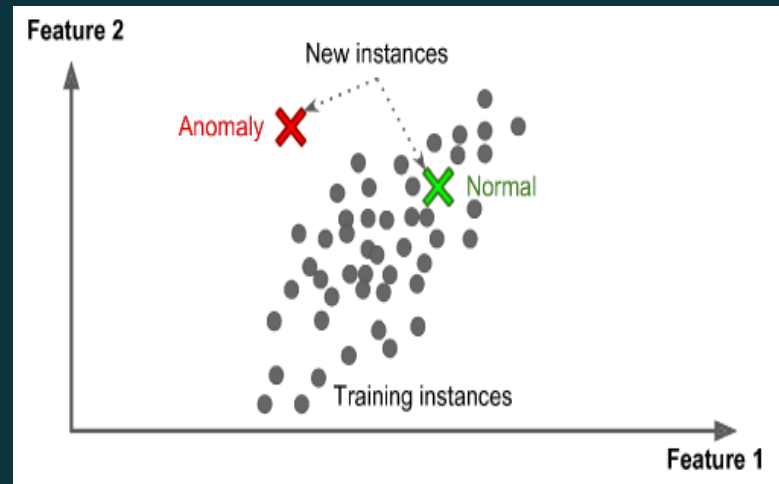
- Reduce data dimensions without losing much information
- Feature extraction
- merge several correlated features into one
- For example, a car's mileage may be strongly correlated with its age, so the dimensionality reduction algorithm will merge them into one feature that represents the car's wear and tear.
- **For example,** reduce dimensions or features before training a supervised Machine Learning Algorithm
- Fast run-time, less memory or processing power, better performance



Dimension Reduction using Principal Component Analysis (PCA)

UNSUPERVISED LEARNING- ANOMALY DETECTION

- Detection of *outliers* or *anomalies*
- Trained using normal instances only
- New instance → classify as normal or anomaly
- Credit Card Fraud
- Detection of manufacturing defects
- Removal of outliers from a dataset
- Novelty detection → detect new instances that look different from all instances in the training data
- **For example**, thousands of dog pictures, 1% represent Chihuahuas, novelty detection will not detect new pictures of Chihuahuas as novelties
- Anomaly detection → may consider Chihuahuas as anomalies



UNSUPERVISED LEARNING- ASSOCIATION RULES

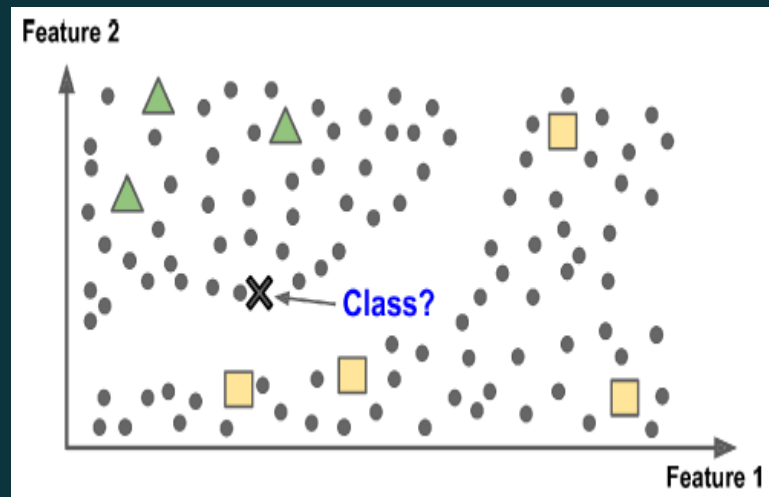
- Examine large amount of data to discover meaningful relationships between the features
- **For example**, for a supermarket sales logs → apply association rules
- Customers who purchase barbecue sauce and potato chips also tend to buy steak
- Keep these items in close to each other



Image Source: <https://medium.com/@manilwagle/association-rules-unsupervised-learning-in-retail-69791aef99a>

SEMI-SUPERVISED LEARNING

- Combines both *supervised* and *unsupervised* learning
- small amount of labeled data with a large amount of unlabeled data
- Knowledge from supervised learning and apply labels to unlabeled data later during prediction
- Reduce the cost and time required for labeling examples
- Example: **Google photos** → Family photos → automatically recognize the same person in multiple photos
- Just add one label per person, the AI can then identify every photo by putting them in the same cluster (unsupervised learning)
- Sometimes error → manually clean the clusters or add more labels



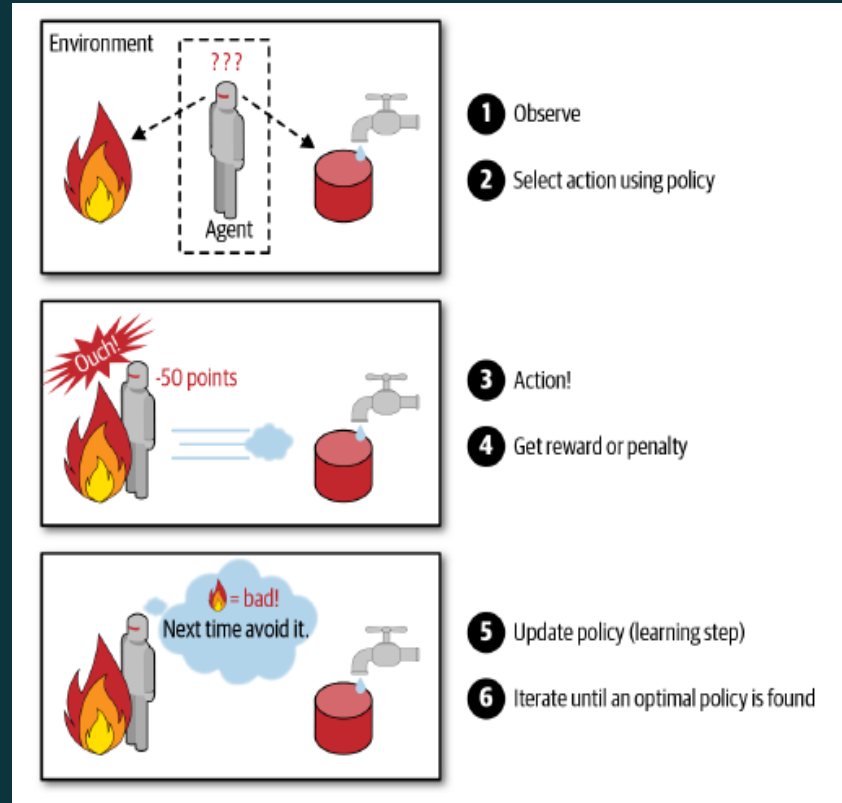
Semisupervised learning with two classes (triangles and squares): the unlabeled examples (circles) help classify a new instance (the cross) into the triangle class rather than the square class, even though it is closer to the labeled squares

REINFORCEMENT LEARNING

- Learning System → **Agent**
- Responsibilities of an Agent:
 - ✓ *observes the environment*
 - ✓ *select and perform actions*
 - ✓ *get **rewards** in return (or **penalties**)*

Agent learns by itself to get the most reward over time

- **Policy** → best learning strategy, an action agent takes when in a given situation
- For example, Robots learning to walk
- DeepMind's AlphaGo program → May 2017 when it beat the world champion Ke Jie at the game of Go by just applying the policy it had learned



BATCH VERSUS ONLINE LEARNING

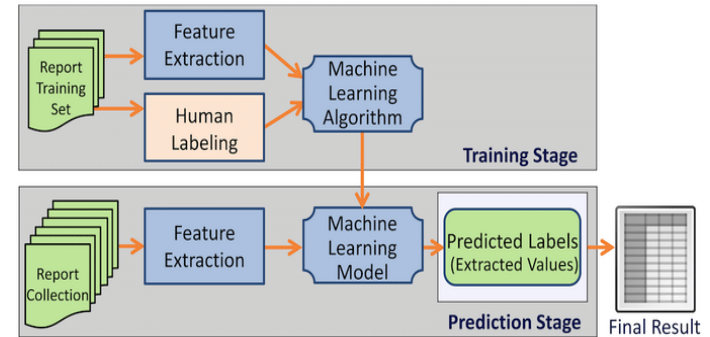
Batch (Offline) Learning:

- Cannot learn incrementally
- Trained offline using available data
- Time consuming, more computation
- Training & prediction stages separate

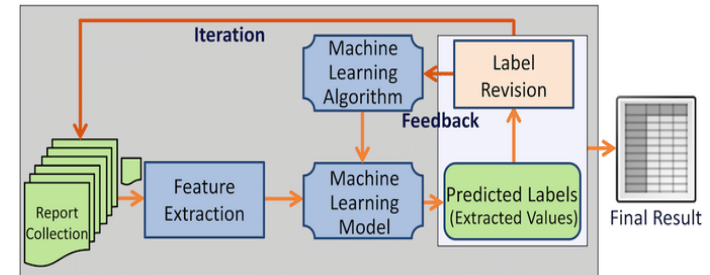
Online (Active) Learning:

- Learns incrementally → sequential or small batches (mini-batches)
- Rapidly changing data (e.g. Stock Prices)
- Fast and cheaper
- Train on huge datasets

Out-of-core training: Train on parts of data and repeat to save computation → offline → incremental



(a) Traditional Batch Machine Learning



(b) Online Machine Learning

Image Source: [8]

SOFTWARE REQUIREMENTS



```
mirror_mod = mod... ob...
    # Set mirror object to mirror_mod
    mirror_mod.mirror_object = mirror_obj

    # operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
    # operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
    # operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

    # selection at the end -add
    mirror_ob.select= 1
    mirror_ob.select=1
    context.scene.objects.active = mirror_ob
    ("Selected" + str(modifier.name))
    mirror_ob.select = 0
    bpy.context.selected_objects = mirror_ob
    data.objects[one.name].select = 1
    print("please select exactly one object")

-- OPERATOR CLASSES -----

bpy.types.Operator):
    # X mirror to the selected object
    object.mirror_mirror_x"
    mirror X"

    context):
    context.active_object is not None
```

INSTALLING JUPYTER NOTEBOOK



Individual Edition

Your data science toolkit

With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Anaconda Individual Edition

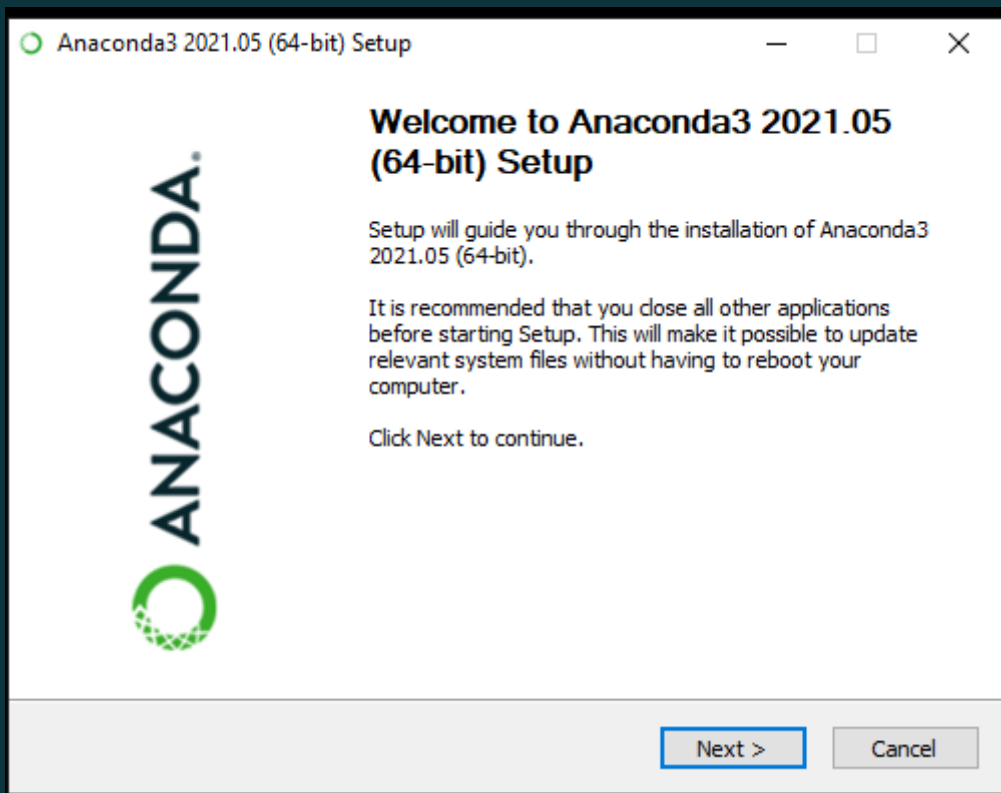
[Download](#) 

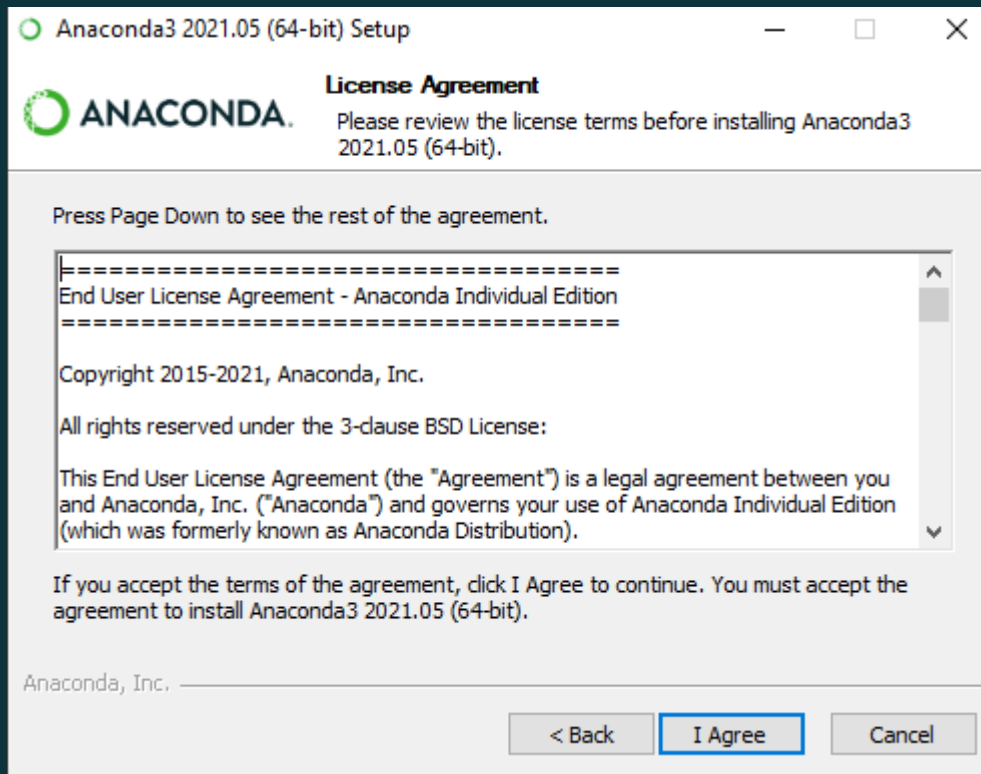
For Windows

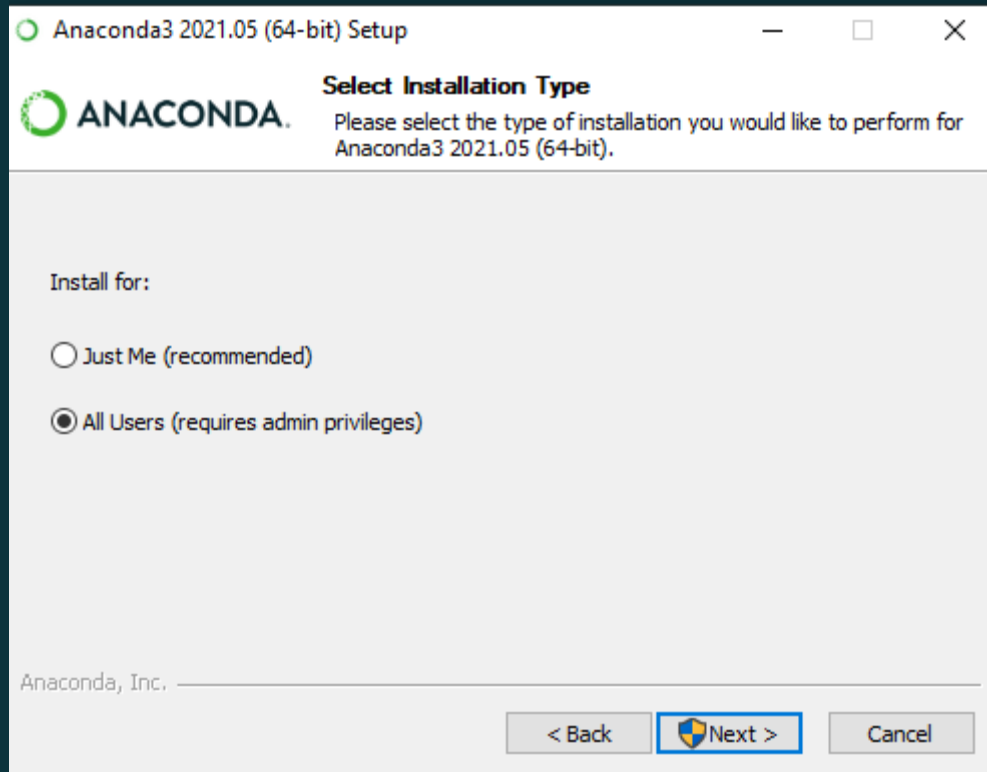
Python 3.8 • 64-Bit Graphical Installer • 477 MB

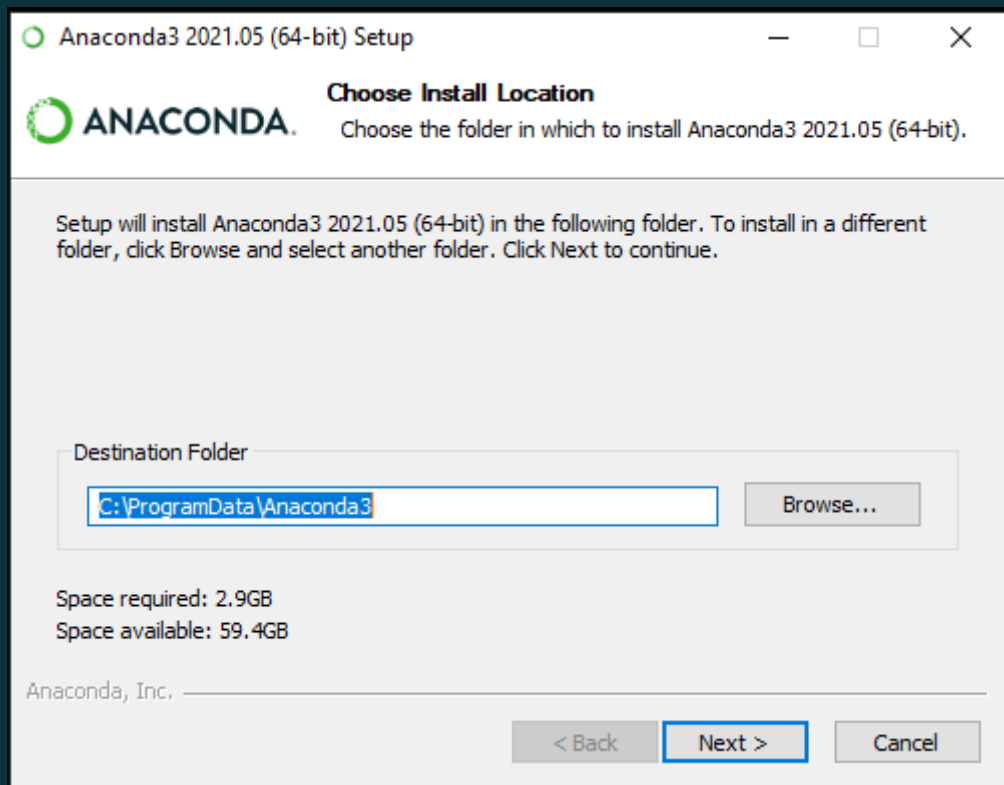
Get Additional Installers

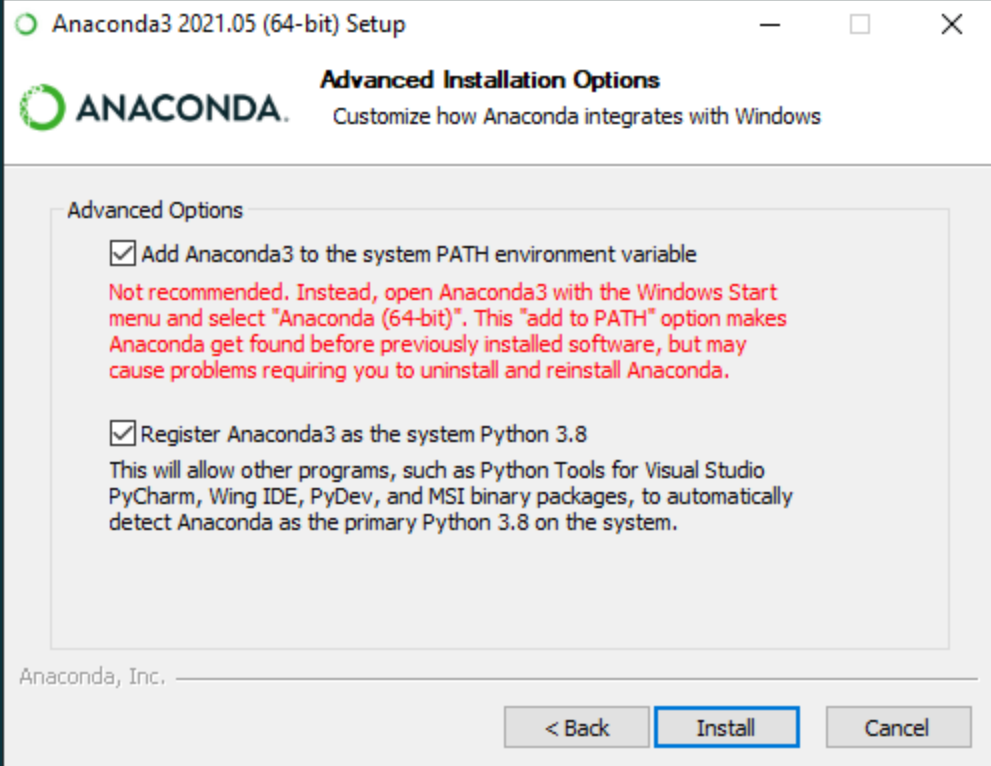














Thanks for installing Anaconda3!

Anaconda is the most popular Python data science platform.

Share your notebooks, packages, projects and environments on Anaconda Cloud!

☐ Learn more about Anaconda Cloud

☐ Learn how to get started with Anaconda

< Back


Finish

Cancel


CREATING YOUR FIRST PROJECT IN JUPYTER NOTEBOOK


AllAppsDocumentsWebMore


Best match


 **Anaconda Prompt (Anaconda3)**
App

Apps


 **Anaconda Navigator (Anaconda3)** >


 **Anaconda Powershell Prompt (Anaconda3)** >

 **Reset Spyder Settings (Anaconda3)** >



Anaconda Prompt (Anaconda3)
App

 Open

 Run as administrator

```
(base) C:\Users\ULYA>jupyter notebook
[I 13:05:19.279 NotebookApp] JupyterLab extension loaded from C:\Users\ULYA\Anaconda3\lib\site-packages\jupyterlab
[I 13:05:19.280 NotebookApp] JupyterLab application directory is C:\Users\ULYA\Anaconda3\share\jupyter\lab
[I 13:05:19.282 NotebookApp] Serving notebooks from local directory: C:\Users\ULYA
[I 13:05:19.282 NotebookApp] The Jupyter Notebook is running at:
[I 13:05:19.282 NotebookApp] http://localhost:8888/?token=8d0372d464554c55765e27706682c304fcc9c10914feaa51
[I 13:05:19.282 NotebookApp] or http://127.0.0.1:8888/?token=8d0372d464554c55765e27706682c304fcc9c10914feaa51
[I 13:05:19.282 NotebookApp] down all kernels (twice to skip confirmation).
[C 13:05:19.344 NotebookApp]
```

To access the
file:///C:
Or copy and pa
http://loc
or http://127

How do you want to open this file?

Keep using this app



Google Chrome

Featured in Windows 10



Microsoft Edge

Do more online with the new browser from Microsoft.

Other options




Internet Explorer

12616-open.html

c9c10914feaa51
c9c10914feaa51

← → ↻ ⓘ localhost:8888/tree 🔍 ☆ ≡ U

 Quit Logout

FilesRunningClusters


Select items to perform actions on them. Upload New ↕

☐ 0 ▾ 📁 /

Name ▾Last ModifiedFile size

<input type="checkbox"/>	📁 3D Objects	18 days ago	
<input type="checkbox"/>	📁 Anaconda3	5 minutes ago	
<input type="checkbox"/>	📁 Contacts	18 days ago	
<input type="checkbox"/>	📁 Desktop	18 minutes ago	
<input type="checkbox"/>	📁 Documents	18 days ago	
<input type="checkbox"/>	📁 Downloads	20 minutes ago	
<input type="checkbox"/>	📁 Favorites	18 days ago	
<input type="checkbox"/>	📁 Links	18 days ago	




← → ↻ ⓘ localhost:8888/tree/Desktop 🔍 ☆ ≡ U

 jupyter Quit Logout

Files **Running** Clusters

Select items to perform actions on them.

☐ 0 ▾ / Desktop Name ▾ Upload New ▾ ↻

- ..
- ☐  [CICFlowmeter](#)
- ☐  [EndNote 9.1](#)
- ☐  [My EndNote Library.Data](#)

Notebook:

[Python 3](#)

Other:

[Create a new notebook with Python](#)

[Text File](#)













[Folder](#)

[Terminal](#)


→ ↻ ⓘ localhost:8888/notebooks/Desktop/Untitled.ipynb?kernel_name=python3

 **jupyter** Untitled Last Checkpoint: a minute ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

        Run    Code ▼ 


Run

In []:  `print("Welcome")`


In []: 

In []: 

→ ↻ ⓘ localhost:8888/notebooks/Desktop/Untitled.ipynb?kernel_name=python3

 **jupyter** Untitled Last Checkpoint: 2 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

New Notebook ▶ ↑ ↓ ▶ Run ■ ↺ ▶▶ Code ▼ 

Open...

Make a Copy...

Save as...

Rename...

Save and Checkpoint

Revert to Checkpoint ▶

```
print("Welcome")
```

Welcome

MAIN CHALLENGES OF MACHINE LEARNING

1) Insufficient Quantity of Training Data:

- Simple problems → need for thousands of examples for training
- Complex problems → Image recognition → millions of training samples needed
- Researchers in [9], showed that different Machine Learning algorithms, performed almost identically well on a complex problem of natural language disambiguation once they were given enough data as shown in the figure.

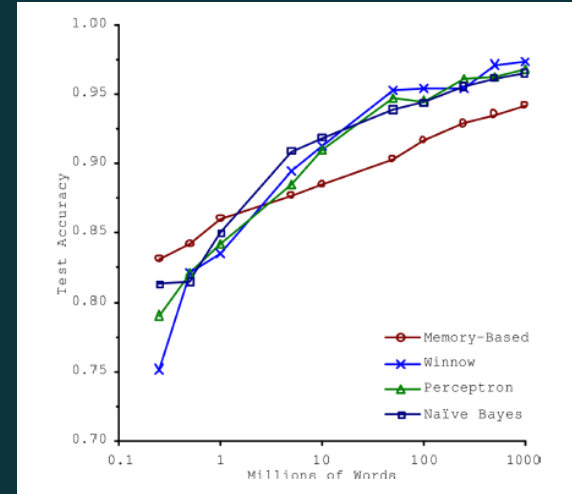


Figure: The importance of data versus algorithms

MAIN CHALLENGES OF MACHINE LEARNING

2) Nonrepresentative Training Data:

- To make better generalization, it is important that training data must be representative of the new instances that we want to make inferences upon
- Figure 2 → Dotted line less representative, solid line more representative
- Adding more countries clarifies that a linear model will not always work, example for richer countries which are not happier than moderately rich countries and poorer countries seem happier than many rich countries
- Such a model with unrepresentative train data may not generalize well for very poor or very rich countries
- Sample too small → *Sampling Noise*
- Large nonrepresentative sample → *Sampling Bias*

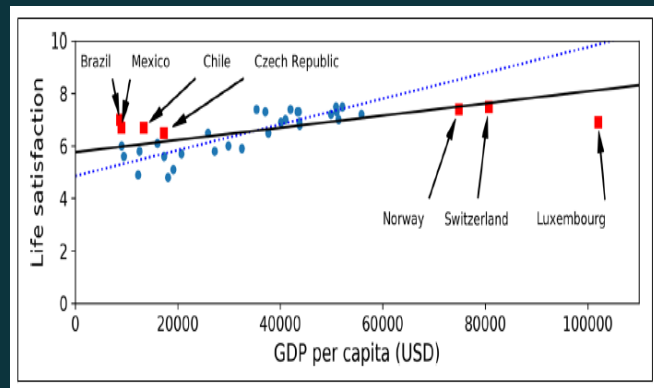


Figure : Solid line more representative training data

MAIN CHALLENGES OF MACHINE LEARNING

3) Poor Quality Data:

- If training data is full of errors, outliers, and noise → poor measurements
- difficult to identify underlying patterns
- The AI model will not generalize well

Possible solutions:

- Data cleaning and pre-processing may solve this problem
- Removing outliers
- remove missing values (e.g., 5% of your customers did not specify their age)
- fill in missing values (e.g., with the median age)

MAIN CHALLENGES OF MACHINE LEARNING

4) Irrelevant features:

- If training data has irrelevant features → poor generalization

Possible solutions:

- **Feature engineering** → critical step for a Machine Learning (ML) Algorithm
- *Feature selection* (selecting the most useful features to train on among existing features)
- *Feature extraction* (combining existing features to produce a more useful ones → dimensionality reduction algorithms)
- Creating new features by gathering new data

MAIN CHALLENGES OF MACHINE LEARNING

5) Overfitting of training data:

- Overfitting means when the model performs well on the training data, but it does not generalize well during production.
- Complex deep learning models such as Deep Neural Network (DNN) can detect subtle patterns in the train data.
- If data is noisy, it can detect patterns in noise and hence will not generalize well on new data

Possible solutions:

- Simplify the model by selecting fewer parameters
- Gather more training data
- Reduce noise in the data
- Putting constraints on the model → regularization
- For example, adjusting the height and slope parameters for a Linear model

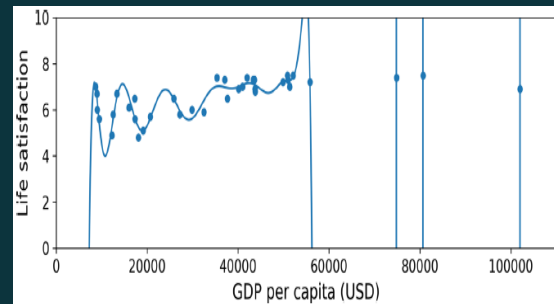


Figure: shows an example of a high-degree polynomial life satisfaction model that strongly overfits the training data.

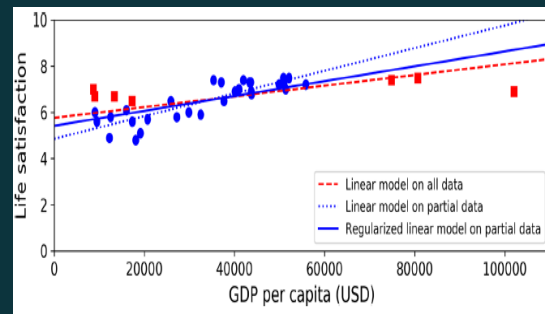


Figure: Regularization reduces overfitting

MAIN CHALLENGES OF MACHINE LEARNING

6) Underfitting the training Data:

- when your model is too simple to learn the underlying structure of the data.
- For example, a linear model of life satisfaction is prone to underfitting
- Real world scenarios are more complex
- The linear model seems very simple, so its predictions may be inaccurate even on train data.

Possible solutions:

- Powerful model → more parameters
- Feature engineering
- Reduce the regularization constraints or other hyperparameter constraints

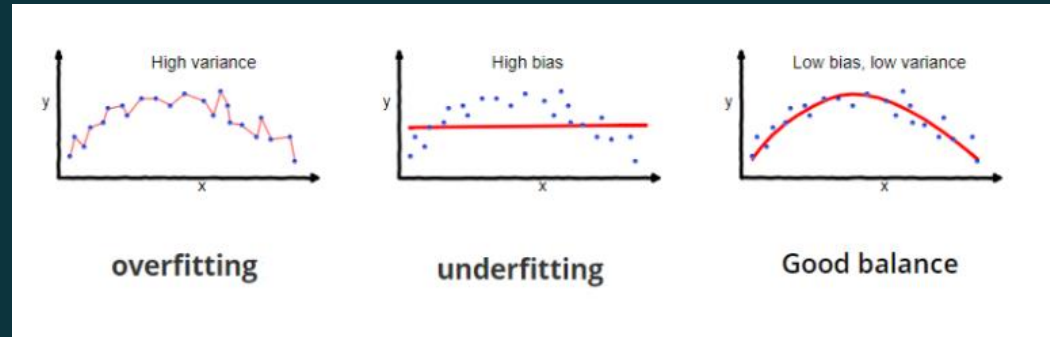


Image source: <https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229>

MAIN CHALLENGES OF MACHINE LEARNING

7) Train and Validation Data Issues:

- For better generalization, divide the data into train and test subsets.
- It is common to use 80% of the data for training and hold out 20% for testing
- But if the size of train data is only 10000 samples, while the test size is 2000, it may not be enough for generalization
- On the other hand, if the dataset contains 10 million samples, then holding out 1% means your test set will contain 100,000 instances → good estimate of the generalization error
- Also hold-out dataset from the same data as train data may share the same characteristics and may not be representative of all the classes in the train data → especially for minority classes

MAIN CHALLENGES OF MACHINE LEARNING

8) Hyperparameter Tuning and Model Selection:

- *Hyperparameter* is a parameter of a learning algorithm (not of the model).
- must be set prior to training and remains constant during training
- During hyperparameter tuning, train multiple models with various hyperparameters on the reduced training set (i.e., the full training set minus the validation set), and you select the model that performs best on the validation set, then evaluate on the test set
- If size of validation set is too small, model evaluations will be imprecise → sub-optimal model
- Very long training time → drawback

Possible solutions:

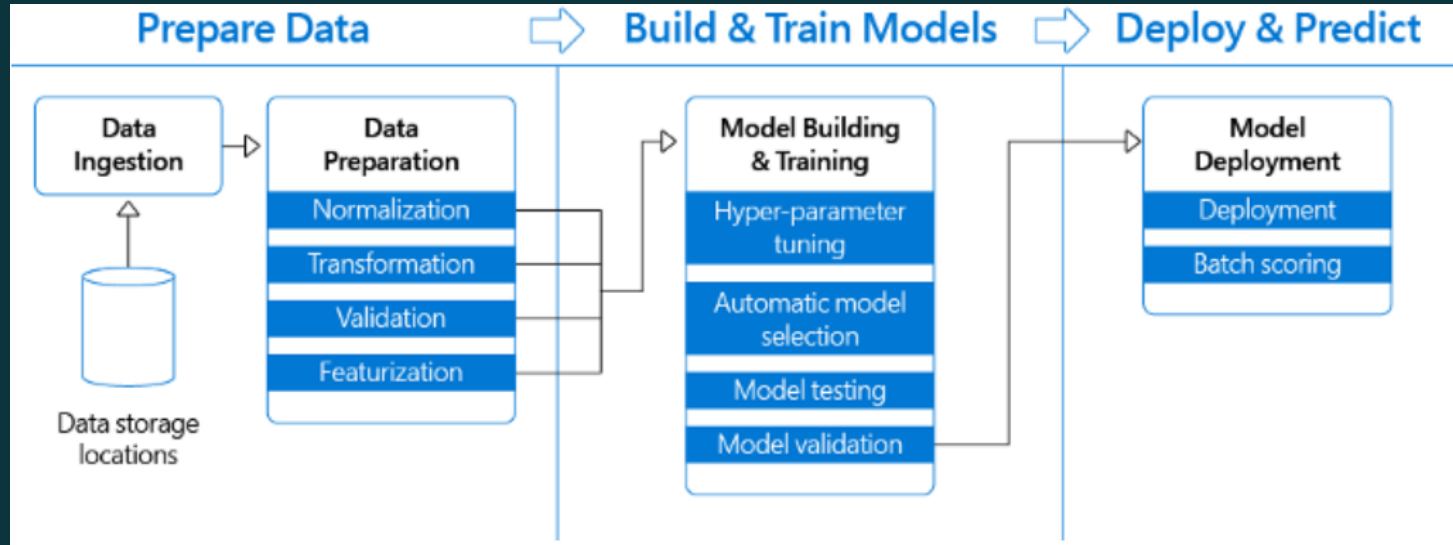
- Cross validation → many small validation datasets → averaging model results

MAIN CHALLENGES OF MACHINE LEARNING

9) Data Mismatch:

- Large training data but nonrepresentative of data used during evaluation (production)
- For example, mobile application for taking pictures to determine and label the species automatically
- If train data only includes pictures from internet sources and not the ones taken from the mobile application, the data is nonrepresentative for the AI algorithm.
- Including pictures both from internet as well as taken through the mobile application into the training data → improved generalization

ML/DL PIPELINE



ML/DL PIPELINE STEPS

Step 1: Data Collection

- Gather data from different sources and merge it into one dataset.
- For example, data for user transactions, demographics, items purchased, etc.

Step 2: Data Cleaning

- For example, find and remove duplicate records, handling missing values, NAN and Infinite values, categorical features

Step 3: Feature Engineering

- Transforming raw data into features, best feature selection or dimensionality reduction to create new features removing correlation among previous features
- Critical and important step for a ML or DL model

Step 4: Normalization or Standardization

- Using Standard Scalar, Min-Max Scalar or Normalization
- For example, for a dataset consisting of two features with a different range of values. Say “age” with a range of 1-100 and “income” with a range of 0-100000. For some ML models (e.g., Multivariate Linear Regression) , features with a higher range influence its output more as compared to features with a lower range thus generating biased predictions.

ML/DL PIPELINE STEPS CONTD.

Step 5: Hyperparameter Tuning

- Choose the best parameters for a learning algorithm → minimize the loss function
- Controls the training process
- Manual Search, Random Search, Grid Search, Bayesian Search, Evolutionary.
- For example, Learning rate, number of neurons, number of hidden layers for Neural Networks

Step 6: Training and Validation

- Train the model on the benchmark dataset, hold-out data for validation (80:20 ratio, commonly used)

Step 7: Generalization (Evaluation) and Deployment

- Evaluate the trained model's performance on different test data subsets → Inference (Generalization)
- Finally deploy the trained model in a real environment

Exercises

Exercise 1: Training and running a Linear model using Scikit-Learn

Exercise 2: Visualizing MNIST dataset using t-SNE

REFERENCES

- [1] <https://becominghuman.ai/voice-recognition-beyond-smart-speakers-6b6c61c7b9e8>
- [2] <https://labs.tadigital.com/index.php/2019/06/24/chatbots-in-ecommerce-the-next-face-of-tomorrow/>
- [3] <https://www.eejournal.com/article/self-driving-cars-what-the-engineers-think/>
- [4] <https://www.aitrends.com/healthcare/machine-learning-advancing-medical-imaging-and-analysis/>
- [5] <https://analyticsindiamag.com/from-months-to-just-a-few-days-building-recommendation-engine-has-become-super-easy/>
- [6] <https://daytradingz.com/artificial-intelligence-stock-trading-software/>
- [7] <https://www.scnsoft.com/blog/ai-threats-cybersecurity>
- [8] Zheng, Shuai, James J. Lu, Nima Ghasemzadeh, Salim S. Hayek, Arshed A. Quyyumi, and Fusheng Wang. "Effective information extraction framework for heterogeneous clinical reports using online machine learning and controlled vocabularies." *JMIR medical informatics* 5, no. 2 (2017): e12.
- [9] Banko, Michele, and Eric Brill. "Scaling to very very large corpora for natural language disambiguation." In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pp. 26-33. 2001.



THANKS!

Do you have any questions?