Universitat de Lleida
Escola Politècnica Superior

Artificial Intelligence

Gerard Mor Martinez
gerard.mor@udl.cat

Data scientist
CIMNE
beegroup

El arte de aprender es, saber preguntar(se)

“El secreto en la vida no es dar respuestas a viejas preguntas, sino hacernos nuevas preguntas para encontrar nuevos caminos”
Einstein

Course description

Year 2024/25 calendar

Theory

October 25: Introduction to ML

November 15: Unsupervised algorithms (1)

November 22: Supervised algorithms (1)

November 29: Supervised algorithms (2)

December 13: Models assessment

December 20: Q&A, examples of ML services at CIMNE

Lab

November 13, 20, 27: Project 2 presentation, data gathering processes and transformation.

December 4, 11, 18: Project 2 modelling using ML techniques

Course description

- Calendar
- Subject guide
- Materials
- Autonomous work:
    - mandatory = False
    - is_recoverable = False
    - min_mark = False
    - max_person_group = 2
- Contact: campus virtual

Introduction

Artificial Intelligence

Is the field of study

Machine Learning

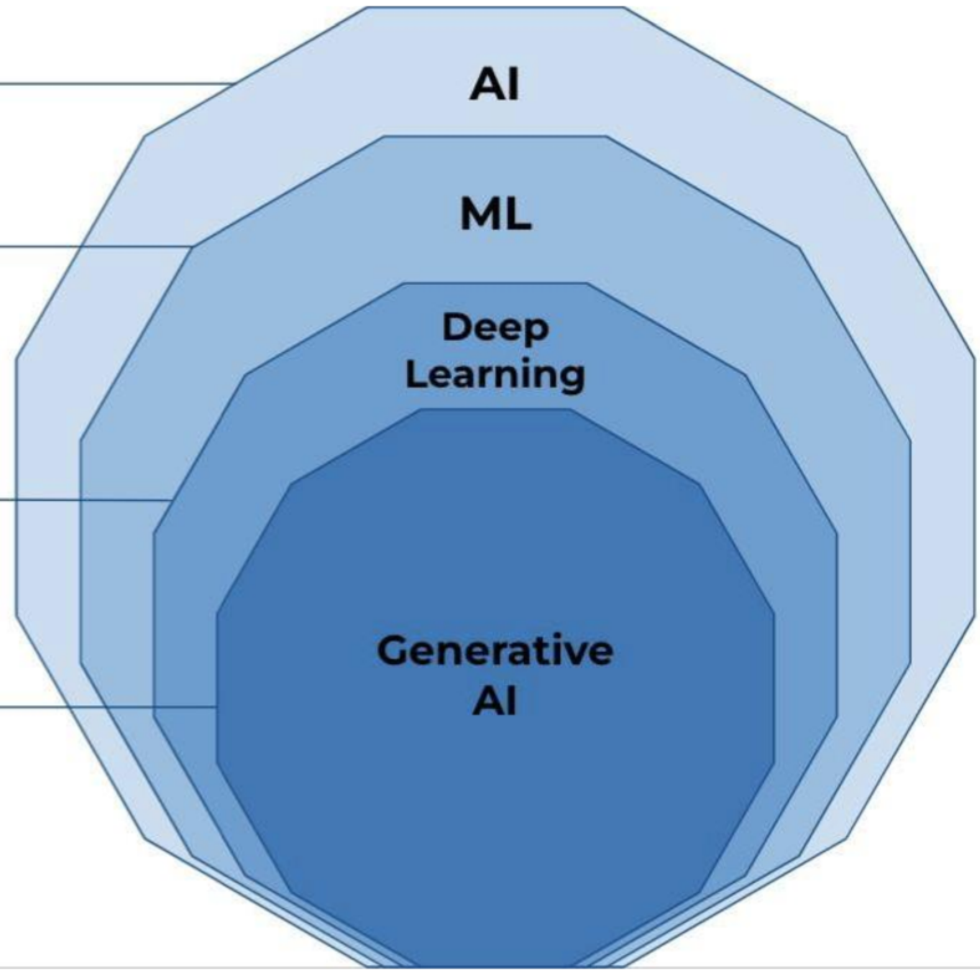
Is a branch of AI that focus on the creation of intelligent machines that learn from data. Another very well know branch inside AI is **Optimization**.

Deep Learning

Is a subset of Machine Learning methods, based on **Artificial Neural Networks.**
Examples: CNNs, RNNs

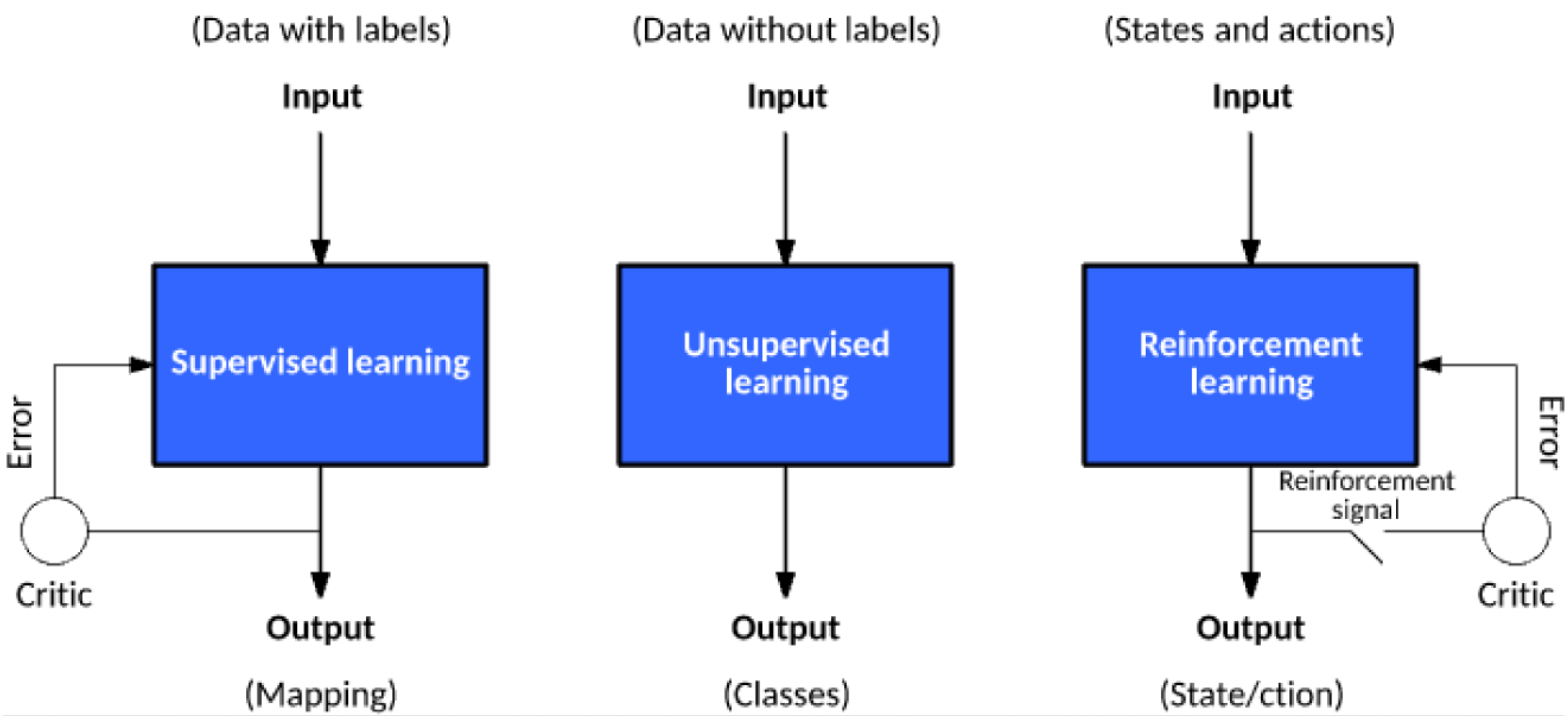
Generative AI

A type of ANNs that generate data that is similar to the data it was trained on.
Examples: GANs, LLMs



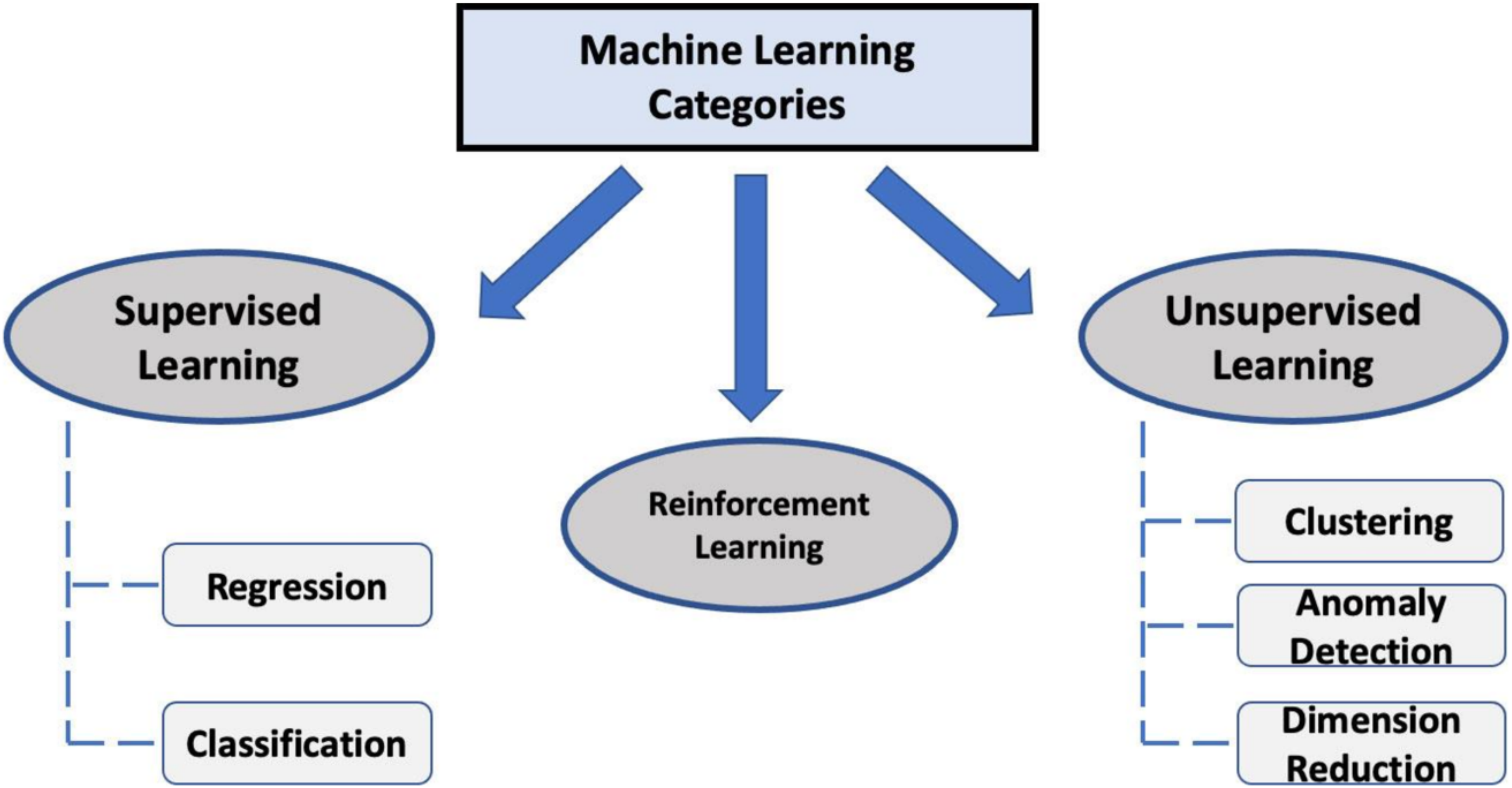
courtesy: https://gcloud.devoteam.com/blog/unlimited-creativity-how-generative-ai-is-transforming-the-world-of-innovation

Supervised, Unsupervised, & Reinforcement Learning



courtesy: https://machine-learning.paperspace.com/wiki/supervised-unsupervised-and-reinforcement-learning

Categories


Machine Learning Categories
Supervised Learning
Reinforcement Learning
Unsupervised Learning
Regression
Classification
Clustering
Anomaly Detection
Dimension Reduction

Parameters vs Hyperparameters in Machine Learning models

Parameters are internal variables that the model learns from training data, optimizing them to make predictions. Examples include weights in a neural network

Hyperparameters are external configuration settings that are set prior to the training process and are not learned from the data. They must be carefully chosen based on domain knowledge, experimentation, or optimization techniques. Examples of hyperparameters include the learning rate, the number of hidden layers in a neural network, or the regularization strength.

Parametric vs Non-Parametric models

Parametric Models	Non-Parametric Models
• They make assumptions about the mapping function • The number of parameters is constant, or independent of the number of training examples For instance: • Naive Bayes* , • Logistic Regression • Linear Discriminant Analysis • Neural Networks.	• Do not make assumptions regarding the form of the mapping function • The number of parameters grows with the number of training examples • They are free to learn any functional form from the training data For instance: • Decision Trees* • k-Nearest Neighbors • Support Vector Machines

(*) They will be studied indeep in this course

Parametric vs Non-Parametrics models (mathematically)

Parametric: Assume a parametric model *M* with a parameter vector θ. The model can be represented as:

$P(X|\theta,M)$

Here, X is the observed data, θ is the parameter vector, and $P(\cdot)$ denotes the probability distribution. The assumption is that the data follows a specific distribution defined by M with parameters θ.

Non-parametric: Consider a non-parametric model N that does not assume a predefined functional form. Instead, it estimates the underlying distribution directly from the data. This can be expressed as:

$P(X|N)$

In this case, the model does not assume a fixed parameter vector θ, allowing for greater flexibility in capturing the underlying data distribution.

Parametric vs Non-Parametrics models (summary)

Parametric ML Algorithms

Benefits

Simpler
Faster
Less Training Data

Limitations

Highly Contrained
Limited Complexithy
Poor fit

Non-Parametric ML Algorithms

Benefits

High Flexibility
Power
High Performance

Limitations

More Training Data
Slower
Overfitting Training Data

General overview

The primary goal of supervised learning is to model a function underlying the statistical relationship between a set of feature variables (i.e. independent variables) and the output (i.e. class, target or dependent variables). For simplicity, the supervised model can be formulated as:

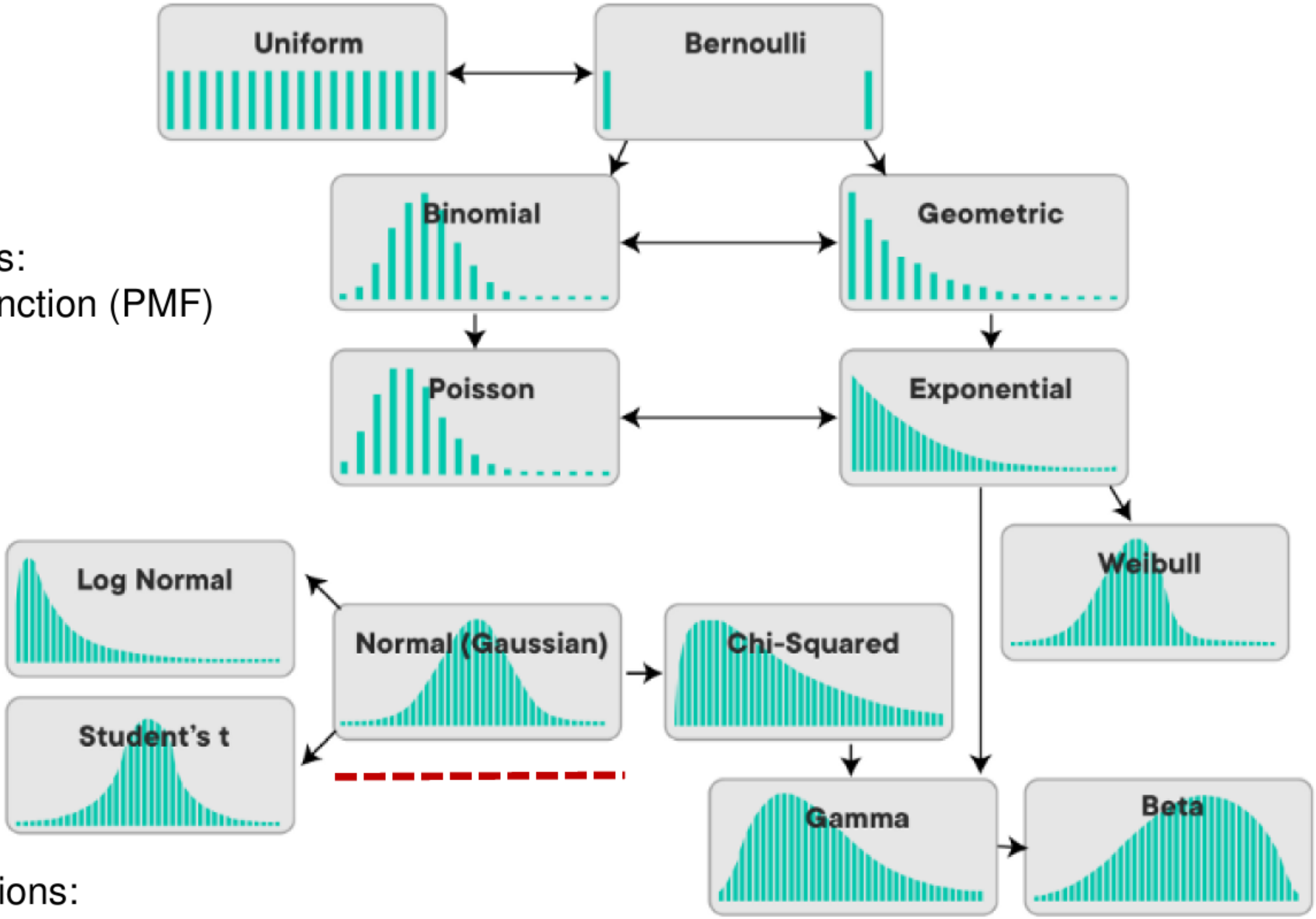
$$Y = f(X) + e$$

where Y denotes the output, $X = (x_1, \cdots, x_p)$ denotes a p-dimensional vector of features, f represents the mathematical function that maps X to Y, and e represents the random error (also known as irreducible error) that is independent of X.
Note that the true function f is typically unknown in practice, and can be only estimated or approximated.

The core step of supervised learning is to estimate $\hat{f}$ or learn the function f based on independent samples of paired features and outputs.

Data distributions

Discrete distributions:
Probability Mass Function (PMF)



Continuous distributions:
Probability Density Function (PDF)

courtesy: https://rman.eeb.utoronto.ca/04-stats1/04-03-sampling-distributions.html

Loss function conceptually

- **Definition**: The loss function represents the discrepancy or error between what the model predicts and the true outcomes.
- **Objective**: The goal is to minimize this loss during the training process, enabling the model to make more accurate predictions.
- **Significance**: The loss function guides the learning algorithm, steering the model towards optimal parameters by reducing prediction errors.

Different types of problems (regression, classification) and model architectures may use different loss functions tailored to their specific requirements.
The choice of an appropriate loss function depends on the nature of the task and the characteristics of the data.

Loss function examples

Regression:

Mean Squared Error (MSE)

- N is the number of rows

$$\mathrm{MSE} = \overbrace{\boxed{\frac{1}{n}\sum_{i=1}^{n}}}^{\text{Mean}}\Big(\overbrace{\boxed{Y_i - \hat{Y}_i}}^{\text{Error}}\Big)^{\overbrace{\boxed{2}}^{\text{Squared}}}$$

Classification:

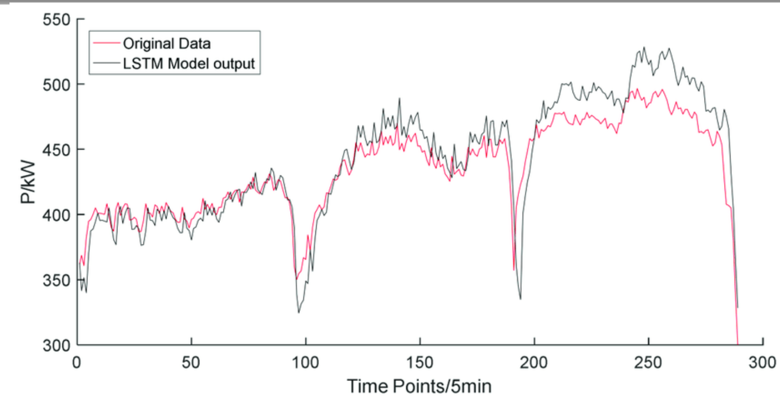
Cross-Entropy Loss

- N is the number of rows
- M is the number of classes

$$-\frac{1}{n}\underbrace{\sum_{j=1}^{M} \overset{\text{Indicator variable}}{y_j}\, log(\overbrace{p(y_j)}^{\text{Prob of class } j})}_{\text{Sum over classes}}$$

$$-\frac{1}{n}\overbrace{\sum_{i=1}^{N}}^{\text{Sum over trials}}\overbrace{\underset{\text{Label}}{y_i}\, log(\underbrace{p(\hat{y}_i)}_{\text{Prob of positive class}}) + (1-\underset{\text{Label}}{y_i})log(1-\underbrace{p(\hat{y}_i)}_{\text{Prob of positive class}})}^{\text{Sum over classes}}$$

Loss function examples



Regression:

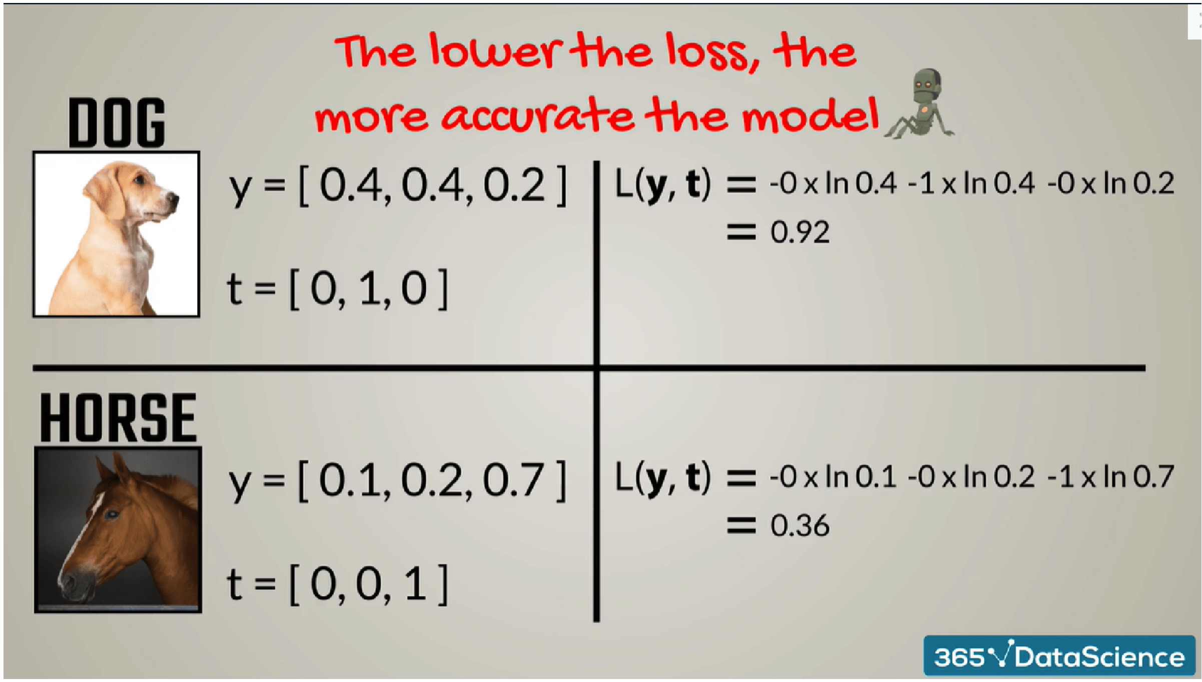
Mean Squared Error (MSE)

- N is the number of rows

Classification:

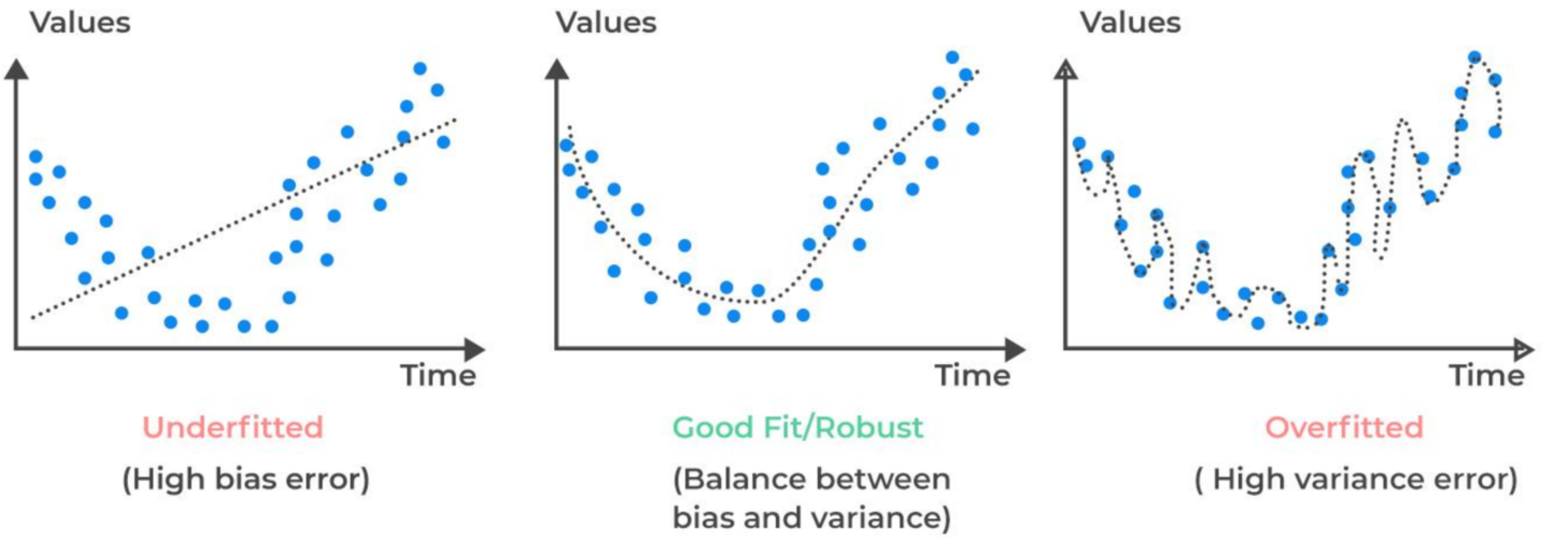
Cross-Entropy Loss

- N is the number of rows
- M is the number of classes



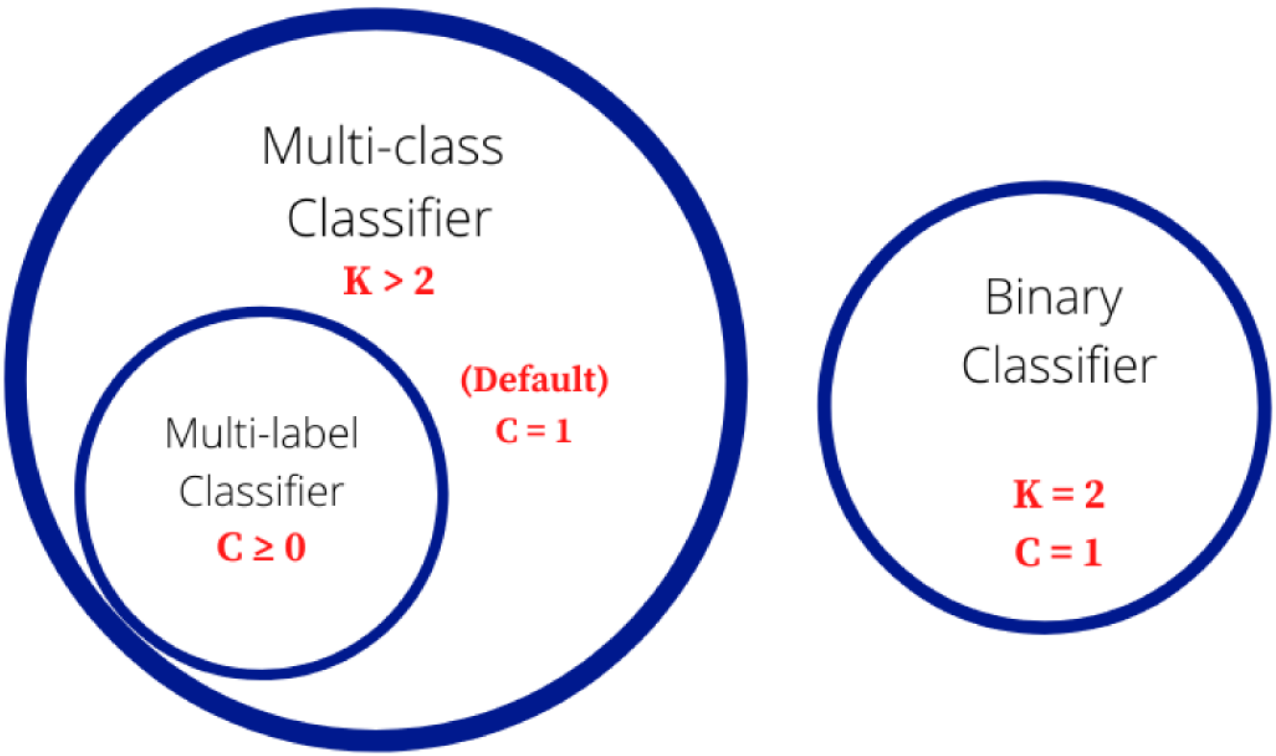
Model validation through training and test datasets

- **Underfitting** occurs when a model is too simple, failing to capture the underlying patterns in the data.
- **Overfitting** happens when a model is too complex, fitting the training data too closely and performing poorly on new, unseen data (test dataset).
- **Generalization** is the ability of a model to perform well on both the training and unseen data, striking a balance between underfitting and overfitting.



courtey: https://analystprep.com/study-notes/wp-content/uploads/2021/03/Img_13.jpg

Types of classification problems

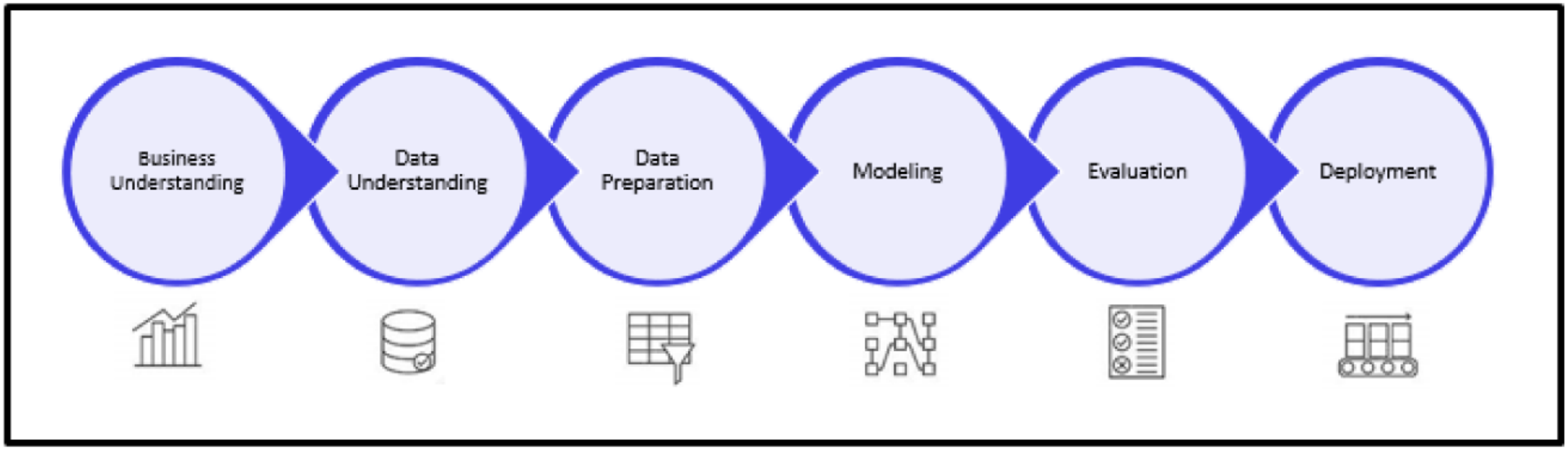


K = Total number of classes in the problem statement

C = Number of classes an item maybe assigned to

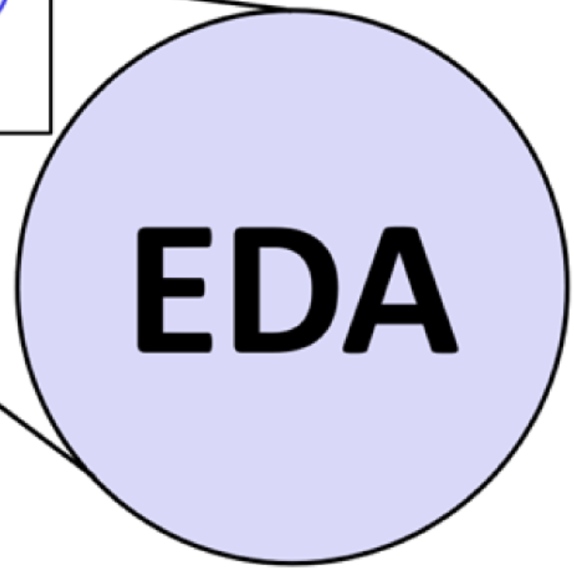
Undertanding a ML project road-map

In the ML world, this is quite different of traditional Software Development Life Cicle (SDLC).



courtey: https://www.analyticsvidhya.com/blog/2021/04/rapid-fire-eda-process-using-python-for-ml-implementation/

Exploratory Data Analysis (EDA)



Out Come of this phase as below

- Understanding the given dataset and helps clean up the given dataset.
- It gives you a clear picture of the features and the relationships between them.
- Providing guidelines for essential variables and leaving behind/removing non-essential variables.
- Handling Missing values or human error.
- Identifying outliers.
- EDA process would be maximizing insights of a dataset.

The most useful frameworks

Polars

SM StatsModels
Statistics in Python

Seaborn

plotly

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

xarray

scikit learn

scikit-image
image processing in python

NumPy

SciPy

matplotlib

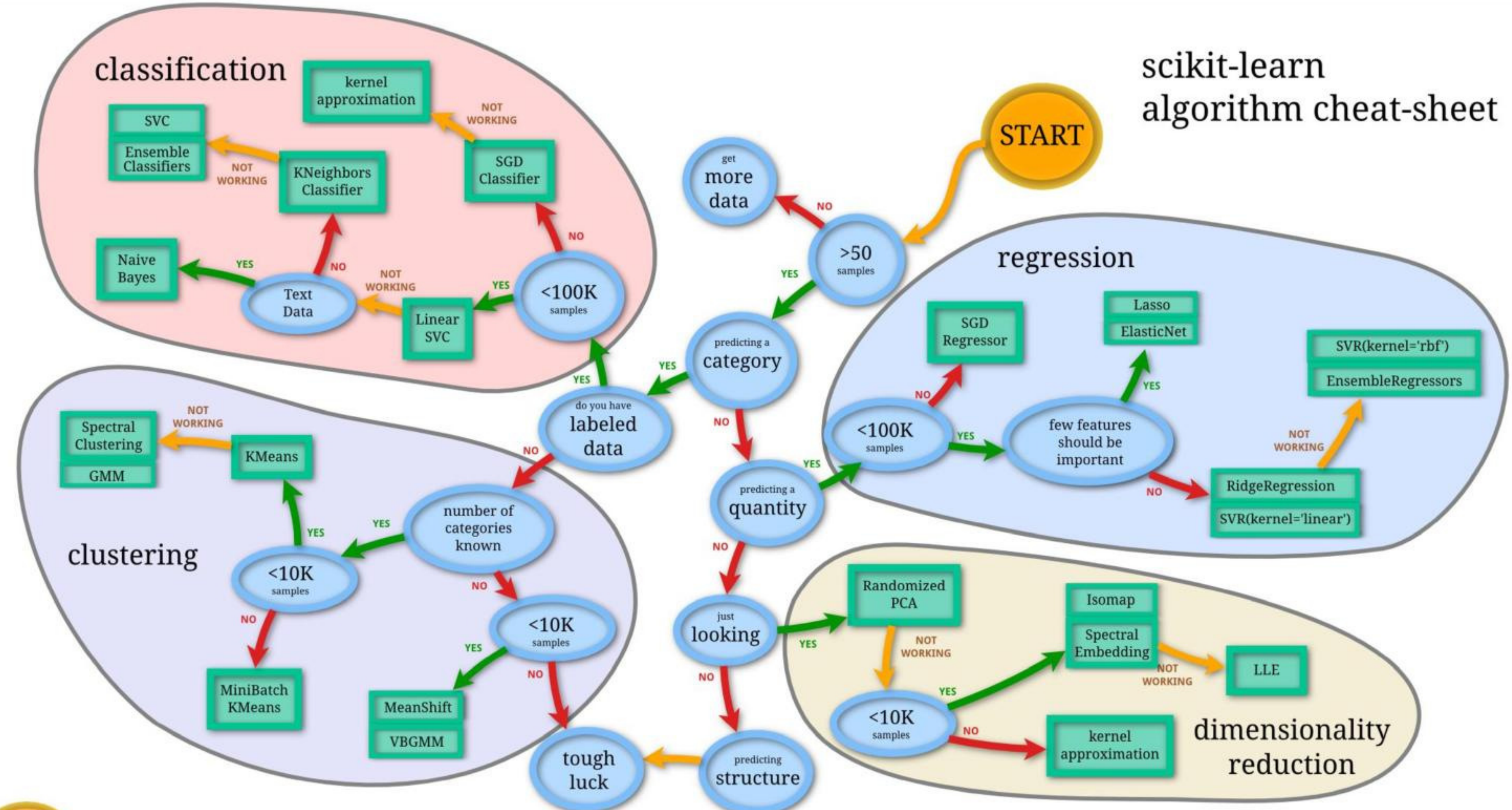
jupyter

python™

IP[y]:
IPython

courtey: https://chris35wills.github.io/courses/pydata_stack.png

Algorithms



Back

https://scikit-learn.org/stable/

Laboratory

1) Setup a professional python project (archetype)

PyCharm Pro

FREE LICENSES

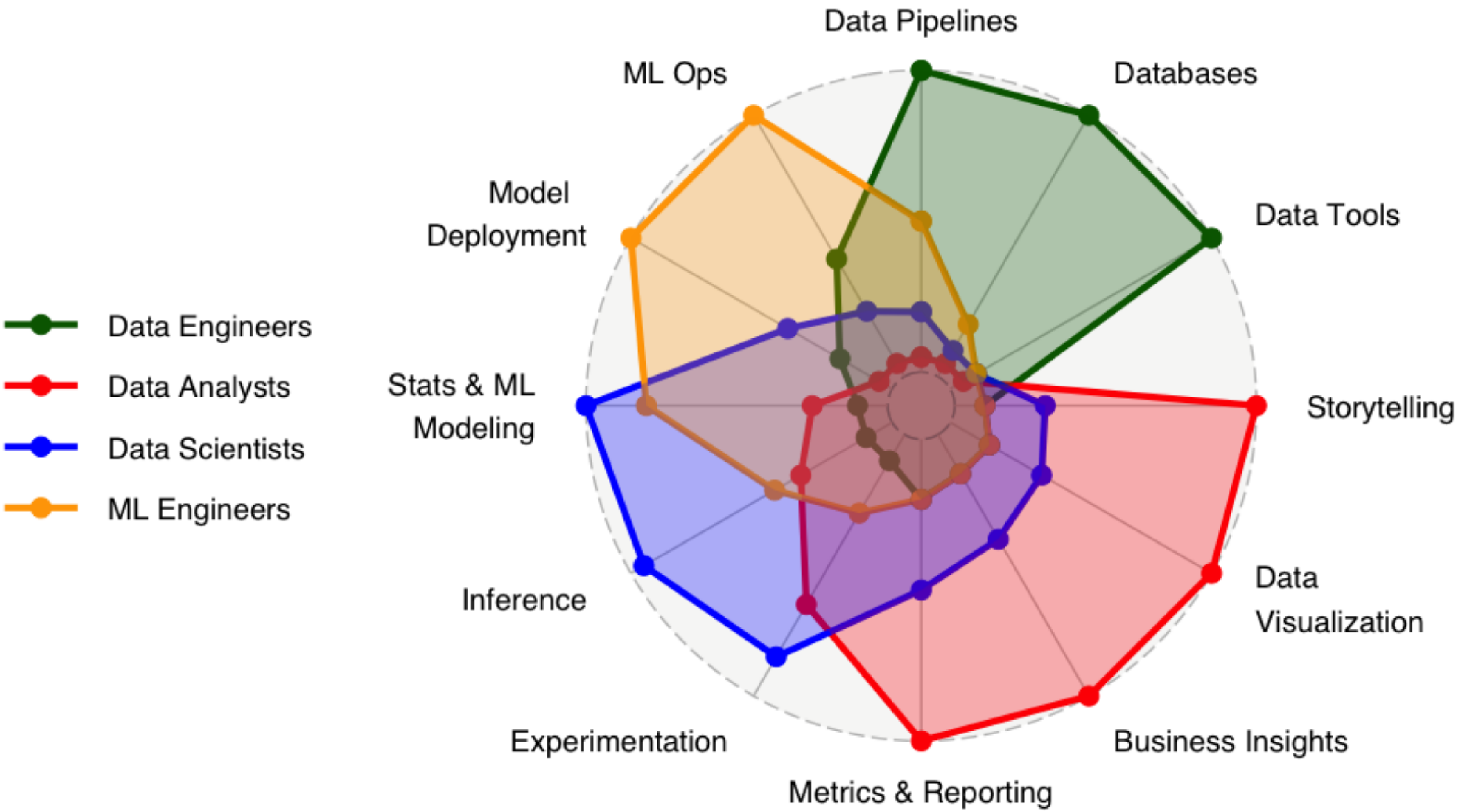
For students and teachers
JetBrains IDEs for individual
academic use

https://www.jetbrains.com/pycharm/download/

2) EDA example: Guide with sklearn and Titanic dataset

https://www.kaggle.com/code/samsonqian/titanic-guide-with-sklearn-and-eda

Roles



courtey: https://images.squarespace-cdn.com/content/v1/61fd85d490d950673294e700/1646607122183-FWTXD6D0P38XJGXE0N5X/Radar.png