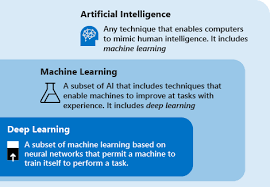
# **Programme Intelligence Artificielle**



# A lire – Important :

Modalités pédagogiques, livrables et évaluation : [Cliquer ici](https://1drv.ms/w/s!AmJGbSlW18YGq-YHEcsgT5hYDviImg?e=ym6EPO)

# Grands Projets (pour la soutenance) :

**Accéder à la liste des** [**grands projets**](https://1drv.ms/w/s!AmJGbSlW18YGq-dbhQeOJHtujQ9s_w?e=wf4R8f)

# **Démo-Déroulé complet :**

## Démo 1 :

Reconnaître des chiffres écrits à la main (***image classification***)

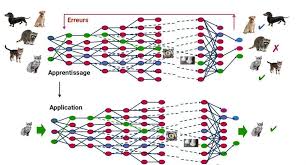
* [*Enoncé MNST*](https://1drv.ms/w/s!AmJGbSlW18YGq-lfjn3po-TXhj_8OA?e=BQwbY4)*,* [Notebook](https://1drv.ms/u/s!AmJGbSlW18YGq-hKIbALwx1b6MlpEw?e=st8lVm) data : **MNIST** 28x28 pixels

## Démo 2 :

IA battle [Notebook](https://1drv.ms/u/s!AmJGbSlW18YGq-hw0ZsIat1G4U6vrw?e=cEzQA0) [data](https://1drv.ms/u/s!AmJGbSlW18YGp-pmboA2LShpEODC8g?e=kLMijP) [moulinette en ligne de tf](https://playground.tensorflow.org/) en utilisant le deep learning

## Démo 3 :

Mettre en prod une IA[DeepLearningCovid19](https://covid-19tn.mybluemix.net/ui/?fbclid=IwAR0yhTWfxGpft-RYvDCcsFv23dt6I-O3Mo2849WC51O5FwZZwprFDSmUorc#!/0?socketid=YWdZRZcYi_3MgJWlAAJf), [TreeDecisionCovid](https://1drv.ms/u/s!AmJGbSlW18YGq_Q9jhqjVJuYlDhJRQ?e=PcuJt8)



# **Installations**

* + - Installer [sklearn](https://scikit-learn.org/stable/), tensorFlow, Keras, [PyTorch](https://pytorch.org/get-started/locally/)

# **Antisèches**

* + - Révision **Python**  [[memonto complet](https://1drv.ms/b/s!AmJGbSlW18YGq-kIg9Xr_JQqg0Kzmg?e=q5gvoY)] [niveau de base] [[intermédiaire](https://1drv.ms/b/s!AmJGbSlW18YGp-gJ1Gei5cYMJRUtcw?e=5ysauh)]
    - **Pandas** : Data Wrangling [[cliquer ici](https://1drv.ms/b/s!AmJGbSlW18YGp-gIt2sV8pNk4bmsQQ?e=bRdzfd)]
    - **Numpy** [[datacamp](https://1drv.ms/b/s!AmJGbSlW18YGq-YKtNXAWLTgaRDpvw?e=LZ4hEt)] [[dataquest](https://1drv.ms/b/s!AmJGbSlW18YGq-YJAI7r6ItLkKL5WA?e=GgNRFU)]
    - **Machine Learning :** Différents types de **modèles** [[cliquer ici](https://1drv.ms/b/s!AmJGbSlW18YGp-gBb10CEw5Da26Uzw?e=9Vr4xK)]
    - **Deep Learning :** Différents types d’architectures de **réseaux de neurones** [[Cliquer ici](https://1drv.ms/b/s!AmJGbSlW18YGq_oAIv1gp7QGCvkxjQ?e=lACuzu)]
    - Machine learning avec **sklearn** [datacomp] [Eudereka](https://1drv.ms/b/s!AmJGbSlW18YGp-gGwcRdt-RL131AfQ?e=MTBSwp)]
    - Aide mémoire [Keras](https://1drv.ms/b/s!AmJGbSlW18YGq_lvHvG0aq0U9LkgLg?e=Gi34W4)
    - Aide mémoire [PyTorch](https://pytorch.org/tutorials/beginner/ptcheat.html)
    - ML : Tips ans tricks from Stanford University [cliquer ici]
    - Cours Deep Learning du MIT [[cliquer ici](http://introtodeeplearning.com/)]

# **Agilité avec Python pour faire de l’IA**

## Chargeons des jeux de données (en vue d’y appliquer de l’IA) depuis :

* Sklearn : [*Cours*](https://1drv.ms/u/s!AmJGbSlW18YGp8ozCSyGp0UZxslR8g?e=VbLdKa) [*TP*](https://1drv.ms/u/s!AmJGbSlW18YGp8oxP8AsVZOcO3diNQ?e=fdDDna) correction
* [UCI ML Repo](https://archive.ics.uci.edu/ml/index.php) : [*Cours*](https://1drv.ms/u/s!AmJGbSlW18YGp8o2JjdobI7YCELH1Q?e=KTVzoT)
* Tensorflow, keras, [PyTorch](https://pytorch.org/) : *Cours* *TP* correction

## Travailler avec des images (en vue d’y appliquer de l’IA) ?

* ***TP****: Charger les jeux d’images suivants :* 
  + *Données :* [*fashion MNSIT*](https://www.kaggle.com/zalando-research/fashionmnist) *Help : un article sur* [*meduim*](https://medium.com/analytics-vidhya/classification-of-images-based-on-fashion-mnist-dataset-bb11e4bcdefb)
  + *Données :* [*Dog Vs Cat*](https://www.kaggle.com/c/dogs-vs-cats/data) *Help : Série de 3 articles sur* [*medium*](https://medium.com/@mrgarg.rajat/kaggle-dogs-vs-cats-challenge-complete-step-by-step-guide-part-1-a347194e55b1)

## TensorFlow, Keras et PyTorch : un nouveau type de données : [tensor](https://pytorch.org/docs/stable/tensors.html) ?

# **Questions méthodologiques**

## Un même jeu de données ne peut à la fois être juge et partie :

Q : *Comment réaliser un train/test split élégant et efficace* ?

* + - **Avec sklearn :**
  + Article à lire absolument : *6 fautes d’amateurs à éviter lors d’un de la mise en place d’un train/test split* : [[cliquer ici](https://towardsdatascience.com/6-amateur-mistakes-ive-made-working-with-train-test-splits-916fabb421bb)]
  + [*Cours*](https://1drv.ms/w/s!AmJGbSlW18YGq-l6cAMHrQQpqhHy2Q?e=EVDco6)
    - **Avec PyTorch :**
  + **Question stackoverflow :** [**How do I split a custom dataset into training and test datasets?**](https://stackoverflow.com/questions/50544730/how-do-i-split-a-custom-dataset-into-training-and-test-datasets)
    - [TP](https://1drv.ms/t/s!AmJGbSlW18YGq-kFHocpvFRvgOpXHA?e=udfC42) *Correction*

## Assurons la reproductibilité des résultats ? :

* + [*Démo*](https://1drv.ms/u/s!AmJGbSlW18YGqNUQ7s1W1xEZ_mxoBw?e=HriUXU)

## Les différentes méthodes de Preprocessing :

* + 1è Article à lire avec des exercices à faire : [*AnalyticsVidhya*](https://www.analyticsvidhya.com/blog/2016/07/practical-guide-data-preprocessing-python-scikit-learn/)
  + 2è Article (juste lecture) : [*TowardsDataScience*](https://towardsdatascience.com/preprocessing-with-sklearn-a-complete-and-comprehensive-guide-670cb98fcfb9)

## Mesurer la performance d’un modèle dans le cadre d’un problème de régression :

* + [les différentes métriques](https://1drv.ms/t/s!AmJGbSlW18YGp8pUbkzDOmbFVBQ_og?e=nytSZi) ?

## Overfitting, Underfitting et comment les prévenir :

*Article très sympa à lire* [[Cliquer ici](https://elitedatascience.com/overfitting-in-machine-learning)]

## Améliorons les performances d’un modèle ? :

[La validation croisée](https://1drv.ms/u/s!AmJGbSlW18YGp8pQSk63RSnNKDyPDQ?e=lmywU5) [Hyperparametres tuning](https://1drv.ms/u/s!AmJGbSlW18YGp8o9LUd0XdSa2_ocNQ?e=6UyMP0) [[Video](https://www.youtube.com/watch?v=w_bLGK4Pteo)]

## Les différentes fonctions d’activation

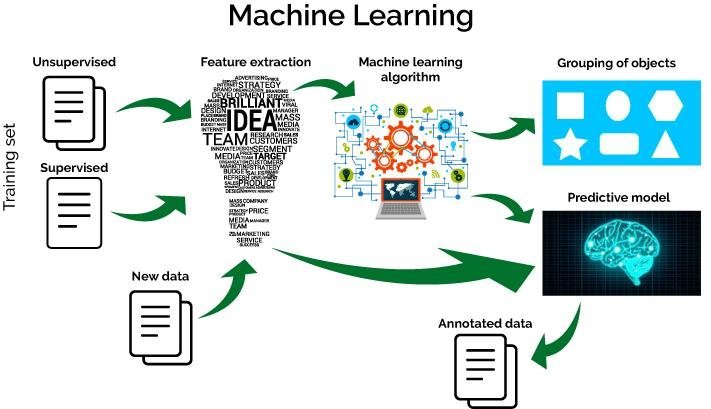
* + Guide pratique : avantages et limites des différentes fonctions d’activation [[Cliquer ici](https://towardsdatascience.com/complete-guide-of-activation-functions-34076e95d044)]

Fonction sigmoide, tangente hyperbolique, softmax, ReLu

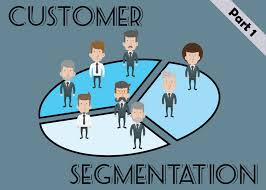
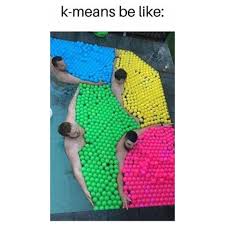
## Dessinons les frontières de décision d’un classifieur ?

* + - Ressources à consulter :
* Article 1 sur iris dataset [[Cliquer ici](https://michael.hahsler.net/SMU/EMIS7332/R/viz_classifier.html)]
* Article 2 sur twoClass dataset Polytechnique [[Cliquer ici](http://www.cmap.polytechnique.fr/~lepennec/R/Learning/Learning.html)]
* [Article 3](https://towardsdatascience.com/decision-boundary-visualization-a-z-6a63ae9cca7d) : Math
  + - [Cours](https://1drv.ms/u/s!AmJGbSlW18YGp-pklle7hR0wJ9CUFw?e=3nl7pR) [*TP*](https://1drv.ms/u/s!AmJGbSlW18YGp-pmboA2LShpEODC8g?e=kLMijP) ; Coder une fonction générique qui permet de dessiner les frontières de décision d’un classifier. Elle prend comme argument le nom du classifier et la data ?

# **Machine Learning : Les modèles**



## **K-means :**



* + ***Learning By Teaching* :** *Groupe concerné - prez 20 min*

*Pour vous aider :* ***Ressources classe inversée*** :

Introduction et implémentation avec Python [[article 1](https://lovelyanalytics.com/2016/09/06/k-means-comment-ca-marche/), [article](https://mrmint.fr/algorithme-k-means) 2, [vidéo](https://www.youtube.com/watch?v=ikt0sny_ImY)]

* + ***Cours* :**
    - Application 1 : Ouvrir la boîte noire du Kmeans [[Enoncé](https://1drv.ms/t/s!AmJGbSlW18YGp-Q8Ujs8LHgxbV2wTw?e=ttXMKq)] [[Exercice](https://1drv.ms/t/s!AmJGbSlW18YGp-Q9VoOH1WjWMegV0g?e=KiCGWW)]
    - Application 2 : Classifier automatiquement des biens [[Python](https://1drv.ms/u/s!AmJGbSlW18YGp-weXtFJZ_D4w2S0YA?e=ybgrsQ)]
    - Application 3 : Classifier automatiquement des voitures suivant leurs caractéristiques
  + ***Learning By Doing* :**
    - TP : Segmenter les clients de **AirBnB** [[article medium](https://medium.com/@sygong/k-means-clustering-for-customer-segmentations-a-practical-real-world-example-196a10323b9f) pour vous aider]
    - TP : kmeans pour classifier des **images** : reconnaître les chiffres écrits à la main [[Accéder aux données](http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits)]
    - Sans être un expert en basket-ball, votre objectif est de classer les jouers du NBA en k classes (k à trouver ?) au sein desquelles les jouers sont le plus semblables possibles [[Accéder aux données](https://1drv.ms/u/s!AmJGbSlW18YGq_M2-2dX4DSse2WRlw?e=CBrCs3)].
  + ***Correction du TP***

## **KNN :**



* + ***Learning By Teaching* :** *Groupe concerné - prez 20 min*
    - Appliquer un KNN sur un problème de **classification** (10 min)
    - Appliquer un KNN sur un problème **régression** (10 min)

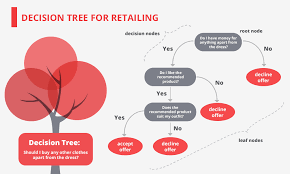
*Pour vous aider :* ***Ressources classe inversée*** :

* + - Introduction et implémentation avec Python [[article medium](https://medium.com/datadriveninvestor/k-nearest-neighbors-knn-7b4bd0128da7), [vidéo](https://www.youtube.com/watch?v=otolSnbanQk)]
    - Résultats et interprétations à obtenir suite à l’application du KNN [[cliquer ici](https://help.xlstat.com/s/article/classification-knn-ou-k-plus-proches-voisins-dans-excel?language=fr)]
  + ***Cours* :**
    - *Application* : Problème de **classification** – jeu de données **iris**
* Démarche avec [[Python](https://1drv.ms/u/s!AmJGbSlW18YGp-MN28eFyI6tUBpdLA?e=lwBNDB)]
* Optimiser le classifieur à l’aide de la validation croisée [[Python](https://1drv.ms/u/s!AmJGbSlW18YGp8pQSk63RSnNKDyPDQ?e=lmywU5)]
* Industrialiser l’optimisation : Hyperparametrs tuning [[Python](https://1drv.ms/u/s!AmJGbSlW18YGp8o9LUd0XdSa2_ocNQ?e=6UyMP0)]
  + - *Application* : Problème de **régression** – jeu de données **AirBnB**

Démarche avec [[python](https://1drv.ms/u/s!AmJGbSlW18YGp-JrtX_z5IQKko1CgA?e=lrkbah)]

* + ***Learning By Doing :*** 
    - [TP](http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits) : knn pour classifier des **images** : reconnaître les chiffres écrits à la main
    - [TP](https://1drv.ms/u/s!AmJGbSlW18YGp-pLV9HSnBYH0B2f5w?e=oJX7wO) : Prédire si le client va acheter ou non ?
  + ***Correction du TP***

## **Decision tree:**



* + ***Learning By Teaching* :** *Groupe concerné - prez 20 min*
    - Arbres de **classification** (10 min)
    - Arbres de **régression** (10 min)

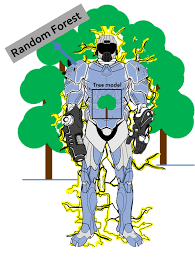
*Pour vous aider :* ***Ressources classe inversée*** : [[Article Medium](https://medium.com/cracking-the-data-science-interview/decision-trees-how-to-optimize-my-decision-making-process-e1f327999c7a)] , [[vidéo](https://www.youtube.com/watch?v=RmajweUFKvM&list=RDQMg__F5u--oIU&index=2)]

* + ***Cours* :**
    - Théorie et sortie logicielle souhaitée [[cliquer ici](https://1drv.ms/t/s!AmJGbSlW18YGp8lStH1yVyd7xUv3BA?e=5pPQfa)]
    - Application : Prédire les survivants du Titanic

[*Code complet d’implémentation*](https://1drv.ms/u/s!AmJGbSlW18YGp8lRiNENbE5H-ZstyQ?e=N8OBRH)

* + - Application : Prédire la décision d’aller jouer au Golf (construire un arbre à la main) : [*Enoncé*](https://1drv.ms/u/s!AmJGbSlW18YGp8kzoapfmwl5-c-BVw?e=gWzldo), *Correction*
    - Entropie et gain d’information : [*Déf*](https://1drv.ms/w/s!AmJGbSlW18YGp8o43EPYDktD3-zsIQ?e=hBZQ1i)*,* [*exercice*](https://1drv.ms/t/s!AmJGbSlW18YGp8o3cYv7FPZ-FmAtuA?e=YlSZSk)*,* [*correction*](https://1drv.ms/u/s!AmJGbSlW18YGp8o53JjrJeGxYIVayw?e=4Y7PDA)
    - **Arbre de régression** : Prédire les prix des biens immobiliers [[Python](https://1drv.ms/u/s!AmJGbSlW18YGp_QtQbdfQK6lTihltg?e=SsTMLE)]
  + ***Learning By Doing* :**
    - Prédire les personnes ayant le diabète [[cliquer ici](https://1drv.ms/x/s!AmJGbSlW18YGp-lK7V_W3-VGnFh5ag?e=qcFDmo)]
  + ***Correction TP***

## **Rondom Forest :**



* + ***Learning By Teaching* :** *Groupe concerné - prez 20 min* 
    - Appliquer un RF sur un problème de **classification** (10 min)
    - Appliquer un RF sur un problème **régression** (10 min)

*Pour vous aider :* ***Ressources classe inversée*** :

* + - Introduction et implémentation avec Python [[article medium](https://medium.com/datadriveninvestor/ensemble-strategy-in-machine-learning-to-surpass-any-single-model-a75a92c3f1f0), [article towards](https://towardsdatascience.com/understanding-random-forest-58381e0602d2)datascience, [vidéo](https://www.youtube.com/watch?v=ok2s1vV9XW0) RF sur digits]
    - Résultats et interprétations à obtenir suite à l’application du RF [[Pb de classification](https://help.xlstat.com/s/article/foret-aleatoire-de-classification-dans-excel?language=fr), [Pb de régression](https://help.xlstat.com/s/article/forets-aleatoires-de-regression-dans-excel?language=fr)]

* + ***Cours* :**
    - *Application* 1 : Problème de **classification** – jeu de données **iris**

Démarche [[Python](https://1drv.ms/u/s!AmJGbSlW18YGp-g3uM-ypkyoJzddbQ?e=3bUIvE)]

Optimiser les hyperparameters [[Python](https://1drv.ms/u/s!AmJGbSlW18YGp-g1ZNwDFo0x92LUgg?e=8GlqDp)]

* + ***Learning By Doing* :**TP à rendre (au choix)

[TP](https://www.kaggle.com/willkoehrsen/a-complete-introduction-and-walkthrough) : *Costa Rican Household Poverty Level Prediction (****données réelles****)*

[TP](https://archive.ics.uci.edu/ml/datasets/optical+recognition+of+handwritten+digits) : *randomForest sur pour reconnaître les chiffres écrits à la main*

* + ***Correction TP***

## **CatBoost, Light GBM, Gradient Boosting and XGBoost :**

* + ***Learning By Teaching* :** *Groupe concerné - prez 20 min* 
    - Appliquer un Gradient Boosting (10 min)
    - Appliquer un RF sur un problème **régression** (10 min)

*Pour vous aider :* ***Ressources classe inversée*** :

* + - Introduction et implémentation du Gradient Boosting avec Python [[article medium](https://medium.com/@gabrieltseng/gradient-boosting-and-xgboost-c306c1bcfaf5)]
    - CatBoost vs. LightGBM vs. XGBoost [[article towardsdatascience](https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db)]

## **Régression linéaire et régression logistique**

* + ***Learning By Teaching* :** *Groupe concerné - prez 20 min* 
    - La régression linéaire ? (10 min)

*Pour vous aider :* ***Ressources classe inversée*** :

* + - * Réaliser une régression linéaire avec sklearn[[article sympa avec des viz et des animations](https://towardsdatascience.com/a-beginners-guide-to-linear-regression-in-python-with-scikit-learn-83a8f7ae2b4f)]
    - La régression logistique ? (10 min)

*Pour vous aider :* ***Ressources classe inversée*** :

* + - * Réaliser une régression logistique avec sklearn pas à pas[[Cliquer ici](https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8)]

Données **réelle** d’une campagne **Marketing** d’une banque

Classification binaire, **Courbe ROC**.

* + - * Classifier des images et comparer la performance des solveurs[[Cliquer ici](https://towardsdatascience.com/logistic-regression-using-python-sklearn-numpy-mnist-handwriting-recognition-matplotlib-a6b31e2b166a)]

Données **réelle** : *MNIST*

**Régression multinomiale**, **descente de gradient : *SGD, lbfgs*, *NAG, Adagrad, Adadelta, Rmsprop***.

* + ***Cours* :**
    - *Application* : Régression linéaire simple et Multiple dans l’art
    - *Application* : Régression logistique sur pima dataset ? [[data](https://1drv.ms/t/s!AmJGbSlW18YGq_J4efAoGRkTpNisog?e=uix5bi)] [[Py](https://1drv.ms/u/s!AmJGbSlW18YGq_J3ou_utiI8wCMb5g?e=xJ4tkB)]
  + ***Learning By Doing* :**TP à rendre (au choix)
  + ***Correction TP***

## **Réseaux de neurones et Algo de descente de gradient :**

### Descente de gradient :

* + ***Learning By Teaching* :** *Groupe concerné - prez 20 min* 
    - La descente de gradient ? (10 min)

*Pour vous aider :* ***Ressources classe inversée*** :

* + - * La Descente de gradient ? : scripts python qui permettent de créer des GIF pour faciliter la compréhension de la descente de gradient [[Cliquer ici](https://github.com/Shathra/gradient-descent-demonstration)]
  + ***Cours* :**
    - Application : Régression linéaire Multiple avec la descente de gradient [[Enoncé](https://1drv.ms/t/s!AmJGbSlW18YGq5xj59xzULPGidiYPw?e=xQ48rf)] [[Données](https://1drv.ms/t/s!AmJGbSlW18YGp-tGQx4QVClUQHg7QA?e=2MNTyJ)] [[Py](https://1drv.ms/u/s!AmJGbSlW18YGp-tInF2U-CQHLIwrSQ?e=UGV5pm)]
    - Application : Régression linéaire multiple avec la descente de gradient sur un très grand jeu de données.
    - Application : Décrypter les signaux d'un sonar : mine *Vs*. Metal [[Cliquer ici](https://1drv.ms/u/s!AmJGbSlW18YGq_Iz6KTorYu8CuW7LQ?e=rnn4Tq)]

Mots-clés : **Régression logistique** avec et sans la **descente de gradient stochastique**, **Courbe ROC**.

* + ***Learning By Doing* :** TP à rendre (au choix) :

### Réseau de neurones :

* + ***Learning By Teaching* :**Prez du matin (Learning By Doing, groupe 1 : 20 min)

*Pour vous aider :* ***Ressources classe inversée*** :

* + - * Perceptron simple avec sklearn ? [[CodePy](https://damianmingle.com/perceptron-in-scikit-learn/)]
      * Perceptron multicouche avec sklearn ? [[CodePy](https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/)]
  + ***Cours* :**
    - [Application](https://auvencecom-my.sharepoint.com/:f:/g/personal/sayf_bejaoui_ynov_com/EppS_kNfg6lKgwaNUUbFm_gBGKC5jGrh793vkGZBleY4Gg?e=sr504b) : Paramétrer un perceptron multicouche  (deep learning) [data](https://1drv.ms/x/s!AmJGbSlW18YGq_MWTHtY-RJBPTGFqw?e=elMG0k) [Cliquer ici](https://www.pluralsight.com/guides/machine-learning-neural-networks-scikit-learn)
  + ***Learning By Doing* :**TP à rendre (au choix)

# **Deep learning**

## **Image Classification :**

### Single-label image classification

* + ***Learning By Teaching* :** *Groupe concerné - prez 20 min*

*Pour vous aider :* ***Ressources classe inversée***

* + - Meilleur cours à suivre « de chez Google » [[Cliquer ici](https://developers.google.com/machine-learning/practica/image-classification/leveraging-pretrained-models)] :

Ce cours comporte 3 applications de classification **Dogs vs. Cats** à réaliser (Très important) :

* + - * Créer un Réseau de neurones convultif from scratch
      * Prévenir le risque de surapprentissage
      * Transfert Learning
    - Différents types d’architectures de réseaux de neurones suivant la taille des images [[Cliquer ici](https://peltarion.com/knowledge-center/documentation/cheat-sheets/single-label-image-classification-/-cheat-sheet)]
  + ***Cours* :**
    - Application 1 :
  + ***Learning By Doing* :**
    - TP : *Données* [fashion MNSIT](https://www.kaggle.com/zalando-research/fashionmnist) *Help : un article sur* [*meduim*](https://medium.com/analytics-vidhya/classification-of-images-based-on-fashion-mnist-dataset-bb11e4bcdefb)
    - TP : *Données :* [Dog *Vs* Cat](https://www.kaggle.com/c/dogs-vs-cats/data) *Help : Série de 3 articles sur* [*medium*](https://medium.com/@mrgarg.rajat/kaggle-dogs-vs-cats-challenge-complete-step-by-step-guide-part-1-a347194e55b1)
    - TP : *Données* [[CIFAR10](https://www.cs.toronto.edu/~kriz/cifar.html), [Caltech-UCSD Birds](http://www.vision.caltech.edu/visipedia/CUB-200-2011.html)] ; Classifier Bird vs. Non Bird *Help : un article d’une série de 8 articles* [[Cliquer ici](https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721) ]
    - TP : *Données* [Car Detection](https://cogcomp.seas.upenn.edu/Data/Car/)
  + ***Correction du TP***