

*Paris, le 6 mars 2018*

**GT BIG DATA  
SGT CLUB ALGO  
RAPPORT D'ACTIVITE 2017**

**Table des matières**

<b>Préambule</b> .....	3
<b>Abstract</b> .....	4
<b>Introduction générale sur le machine learning (ML)</b> .....	6
<b>Première partie : Challenge Kaggle Porto Seguros</b> .....	11
<b>1/ Présentation du Challenge</b> .....	11
<b>2/ Présentation de l'approche du Club Algo</b> .....	13
<b>2.1 Qu'est-ce que la Mlbox</b> .....	14
<b>2.2 La Mlbox par rapport aux autres</b> .....	15
<b>2.3 La Mlbox appliquée au Kaggle</b> .....	15
<b>2.4 Autre approche proposée</b> .....	15
<b>Deuxième partie : Coursera « Neural Network and Deep Learning » d'Andrew Ng</b> .....	16
<b>Troisième partie : Le Deep Learning au service de l'assureur non-vie</b> .....	18
<b>Contexte</b> .....	18
<b>Quelques mots sur les réseaux profonds</b> .....	18
<b>Détection de modèle et de marques de véhicule</b> .....	19
<b>Événements 2017 du Club Algo</b> .....	20
<b>Conclusion générale</b> .....	21
<b>Annexes</b> .....	22

## *Encouragements*

*« Tous mes encouragements au Club Algo pour leurs travaux,*

*Xavier Conort<sup>1</sup>, DataRobot<sup>2</sup> »*

*« Bravo au Club Algo et au dynamisme de cette jeune communauté de data scientists au service de l'actuariat,*

*Florence Picard, présidente de la Commission Scientifique de l'Institut des Actuaire »*

*« Bonne continuation au Club Algo,*

*Arthur Charpentier, professeur université Rennes I »*

---

<sup>1</sup> Xavier Conort, actuaire et champion Kaggle, est Chief Data Scientist chez DataRobot

<sup>2</sup> DataRobot, spécialiste et pionnier dans la conception d'outils automatisés de "Machine Learning"

## Préambule

Le *Club Algo* a été lancé le 13 juin 2016, c'est un des sous-groupes du groupe de travail « Big Data » de l'Institut des actuaires créé par Florence Picard en décembre 2013.

Son objectif est double :

- Échanger librement sur l'utilisation des algorithmes et les bonnes pratiques.
- Animer des événements en relation avec les problématiques liées à la Data science et l'actuariat.

À la fin de chaque année, le Club rédige un rapport des travaux menés sur les algorithmes, à destination d'un public d'actuaires et à disposition de l'Institut des Actuaires.

Cinq études ont été menées **en 2016** :

1. **Prévention routière** : Études des accidents corporels
2. **Challenge Maif** : Décodage d'une formule de pricing
3. **Pricing Game 2016** : Proposer un tarif auto
4. **Assurance Prairie** : La France découpée grâce à la télédétection
5. **Text Mining** : Detecting Insults in Social Community

**En 2017, les sujets suivants ont été traités par le Club :**

1. **Kaggle** : Challenge de Porto Seguros
2. **Coursera** : Réseaux de neurones et Deep Learning (Andrew Ng)
3. **Deep Learning** : « Automatisation de la chaîne de confiance lors d'un sinistre auto »
4. **Événements** : Petit-déjeuner chez Dataiku le 20 septembre 2017 et atelier à 100% Datascience de l'Institut des actuaires du 16 novembre 2017

### *Plan du rapport*

Après une introduction générale sur le machine learning, nous verrons en première partie, comment le concours Kaggle proposé par Porto Seguros a été mené par le Club Algo.

En deuxième partie, nous décrivons rapidement le contenu du Coursera d'Andrew NG : « Neural Network and Deep Learnig ».

Puis en troisième partie, nous présenterons l'étude sur la reconnaissance de modèle et de marque de voiture et comment l'appliquer à un cas concret en assurance dommage auto.

## Abstract

**Key words** : Machine learning, neural networks, deep learning, Kaggle's challenge, Mlbox, Coursera, car's pictures, events.

The "Algo Club" was launched on June 13, 2016, it is one of the subgroups of the "Big Data" working group of the Institute of Actuaries created by Florence Picard in December 2013.

His goal is twofold :

- Exchange freely on the use of algorithms and best practices.
- Animate events related to issues related to Data Science and Actuarial Science

At the end of each year, the Club writes a report on the work done on the algorithms, intended for an audience of actuaries and to the Institute of Actuaries disposal.

Five studies were conducted **in 2016** :

1. **Road safety** : accident studies
2. **Maif Challenge** : Decoding a pricing formula
3. **Pricing Game 2016** : Propose a car rate
4. **Insurance Prairie** : Satellite insurance
5. **Text Mining** : Detecting Insults in Social Community

**In 2017**, the following topics were treated by the Algo Club :

1. **Kaggle** : Porto Seguros Challenge
2. **Coursera** : Neural Networks and Deep Learning (Andrew Ng)
3. **Deep Learning** : "Chain of trust automation during a car accident"
4. **Events** : Breakfast at Dataiku's office in Paris, September 20, 2017 and 100% Data science Workshop of the French Institute of Actuaries in Paris, November 16, 2017

### *Report plan*

After a general introduction to machine learning, we will see in the first part how the Algo Club has managed the Porto Seguros challenge on Kaggle platform.

In the second part, we will briefly describe the contents of Andrew NG's Coursera: "Neural Network and Deep Learning".

Then in the third part, we will present the study on model recognition and car brand and how to apply it to a concrete case in auto damage insurance.

## *Remerciements*

*« Tous mes remerciements aux membres contributeurs de ce rapport :*

*Anne-Claire Martial (Le Conservateur), Imen Saïd (Mazars Actuariat), Maryse Dama,  
Meryem Yankol (Aviva), Victor Reutenauer (Fotonower) et Vincent Grari (Axa)*

*Nathalie Ramos – Responsable du Club Algo »*

*Et tous nos remerciements à Dataiku pour leur chaleureux accueil et leur délicieux petit-déjeuner...*

*Club Algo*

## Introduction générale sur le machine learning (ML) (par Imen Saïd, Mazars Actuariat)

Dans un monde qui se définit comme l'ère de la digitalisation et du Big Data, il devient de plus en plus important de se doter de bons outils permettant d'optimiser le processus de prise de décision.

La modélisation statistique basée sur les algorithmes du Machine Learning (ML) s'est petit à petit imposée comme une solution efficace face aux défis des acteurs des différentes industries en général et de l'Actuariat en particulier où la modélisation de l'aléa est le cœur du métier. En effet, les modèles du ML permettent de profiter au mieux de l'abondance des données pour optimiser la chaîne de production et de prise de décision.

L'objectif d'un modèle de l'apprentissage statistique est de fournir des prédictions robustes sur une population non observée (population test) à partir des résultats de l'apprentissage effectué sur une population initiale (population de l'apprentissage). Les algorithmes issus de la théorie de l'apprentissage statistique sont divers et variés, mais les étapes d'un exercice de modélisation restent sensiblement les mêmes.

Dans le cadre des activités du Club Algo, nous avons travaillé sur plusieurs problématiques et utilisé des algorithmes de ML pour répondre à des problèmes liés à l'Actuariat. Nous partageons ici notre retour d'expérience sur les bonnes pratiques à appliquer qui sont valables pour tout exercice de modélisation.

Nous détaillons ci-dessous les grandes étapes que l'on retrouve communément dans ces analyses :



### 1/ Prise en main des données :

#### Statistiques descriptives et étude des corrélations

Avant de se lancer dans la construction du modèle, il est important de bien connaître et maîtriser les données dont on dispose. Dans le cadre de l'apprentissage supervisé, on observe d'un côté des variables explicatives et de l'autre la variable réponse correspondante. Par contre, dans le cadre de l'apprentissage non supervisé, nous n'avons aucune variable réponse, nous observons uniquement des individus qu'on cherche à classer dans des groupes homogènes. Dans tous les cas, il est indispensable de bien connaître la distribution des différentes variables qui constituent notre base de données. Cette étape est d'autant plus importante dans un monde où les bases de données sont souvent enrichies à partir de plusieurs sources de données : elle permet d'auditer rapidement les variables, d'en apprécier la qualité (données manquantes ou non normalisées), de détecter les variables bruits non discriminantes (par exemple celles présentant un taux important de données manquantes) et celles qui sont fortement corrélées entre elles. Ci-après, quelques axes d'analyses préliminaires à effectuer sur la BDD :

- Le nombre de variables qu'on s'apprête à utiliser : l'approche de modélisation d'un problème avec une dizaine de variables explicatives diffère de celle qu'on applique si on dispose d'une centaine de variables (espace hyper paramétrique).

- Le type de variables (numériques, factorielles) : le traitement et l'utilisation des variables numériques diffère des variables factorielles. De plus le modélisateur peut modifier le type de certaines variables : par exemple, on peut transformer la variable âge de numérique à factorielle en créant des classes d'âge selon des règles métiers qu'on veut appliquer.
- La distribution des variables explicatives et les corrélations existantes entre elles et avec la variable output si on l'observe. La forte corrélation entre les variables explicatives peut biaiser le programme d'optimisation utilisé pour la modélisation.
- Finalement, l'étude des corrélations entre les variables explicatives et la variable output dans le cas d'un exercice de l'apprentissage supervisé, en réalisant par exemple l'analyse en composantes principales (ACP/AFC) qui permet de voir les axes qui concentrent le plus d'inertie et les variables les plus discriminantes.

### Traitement des données manquantes

Bien que l'on soit à l'ère du « Big Data », l'abondance des données n'est pas forcément synonyme de qualité de données. Souvent, le data scientist se confronte à des bases de données avec des valeurs manquantes (missing values) engendrées par le processus même de récupération des données ou tout simplement par la non disponibilité de l'information pour certains individus. Face à ce problème, plusieurs solutions sont à prendre en compte : selon le nombre de données manquantes et la variable concernée, on peut choisir de :

- Exclure les lignes qui présentent des données manquantes ;
- Exclure les variables qui présentent plusieurs données manquantes ;
- Compléter les données manquantes en utilisant un proxy ou une méthode de classification pour remplir les données manquantes. Par exemple, nous pouvons :
  - Affecter la moyenne de la variable à toutes les cases manquantes (ou la modalité la plus fréquente dans le cas d'une variable catégorielle) ;
  - Utiliser des méthodes de classification pour « prédire » la valeur la plus adéquate, ou la plus proche pour l'observation manquante (par exemple l'utilisation de kNN : k voisins les plus proches).

Dans tous les cas, notons que le choix de la méthode de traitement des données manquantes a un impact sur les résultats de la modélisation. En effet, le choix d'une méthode ou une autre change de manière plus au moins significative la base sur laquelle on apprend notre modèle. L'absence d'une donnée peut d'ailleurs constituer une information.

## **2/ Choix de la structure générale du modèle selon l'objectif de la modélisation**

Après l'exploration et le traitement des données, vient le choix du modèle idoine. Rappelons que, dans la grande famille des algorithmes de l'apprentissage statistique, on distingue : le supervisé et le non supervisé.

- Apprentissage supervisé : on dispose d'un set de variables explicatives et d'une variable output identifiée. On cherche donc à modéliser la distribution de la variable output conditionnellement aux variables explicatives. Dans un deuxième temps, on cherchera à prédire l'output sur un nouvel individu dont on observe uniquement les variables explicatives.

- Apprentissage non supervisé : on dispose d'une population caractérisée par plusieurs variables. On ne distingue pas la variable réponse. Il s'agit d'un problème de classification/segmentation où on cherche à créer les groupes les plus homogènes possibles et qui auront donc le même label. Dans un deuxième temps, on cherchera à classer un nouvel individu parmi l'un des groupes.

La définition de notre problème et de l'objectif de la modélisation permet d'ores et déjà de connaître la structure générale du modèle à utiliser. Par exemple, les modèles de régression ne sont pas applicables dans le cadre d'un problème non supervisé. Par ailleurs, si l'output est une variable continue (coût d'un sinistre par exemple), les algorithmes de classification ne sont pas bien adaptés et peuvent censurer la distribution réelle de la variable à prédire.

### 3/ Comparaison de la performance de différents modèles

#### Risque de sur apprentissage et de sous apprentissage : Équilibre biais-variance

Pour le même exercice de modélisation et à partir de la même base de données, il existe plusieurs réponses possibles et différents algorithmes de ML à implémenter. Le défi du data scientist est de sélectionner le modèle le plus robuste. Or, pour comparer la performance de plusieurs modèles ou de versions de modèles, il faut commencer par définir les indicateurs de performance d'un modèle.

Rappelons que l'objectif de l'exercice est double : d'un côté, il faut réussir à modéliser le plus fidèlement possible la distribution de la population observée, et de l'autre, il faut prédire au mieux la réponse sur une nouvelle population (population test). En d'autres termes, il faut que le modèle arrive à trouver un équilibre entre le risque du sous-apprentissage et du sur-apprentissage.

Un modèle qui « sur-apprend », est un modèle qui apprend par cœur les réponses observées sur la population initiale. Ce modèle réduit fortement le risque du sous apprentissage, par contre, il a un pouvoir prédictif très limité :

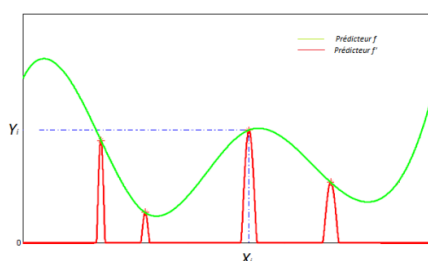


Figure 1: Illustration du surapprentissage (prédicteur rouge)

On introduit ainsi la notion de **l'équilibre biais-variance** primordiale dans le cadre de l'évaluation de la performance des modèles. Le meilleur modèle qui répond au problème du data scientist n'est pas forcément celui qui réplique aveuglément toutes les spécificités de la population observée. On préfère un modèle qui connaît moins la population d'apprentissage mais qui arrive à généraliser mieux sur une population inédite. Ceci est d'autant plus important quand on cherche à construire un processus de prise de décision robuste, peu sensible à la variabilité des données et capable de capter les axes d'analyses inhérents à la distribution de la variable réponse conditionnellement à la combinaison des variables explicatives :



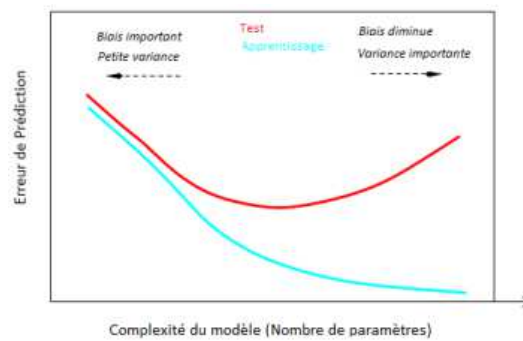


Figure 2: Illustration de la notion de l'équilibre biais-variance

### Présentation de certains indicateurs de mesure de la performance

Ainsi, pour comparer les différents modèles, il est important de mesurer la performance des différents prédicteurs sur la population d'apprentissage et sur la population test (sur laquelle il n'a pas appris). Ainsi, avant de lancer l'apprentissage, il est important de disposer de deux populations (dont on observe la réponse) : une pour mesurer le risque de sous-apprentissage et l'autre pour mesurer le risque du sur-apprentissage. Concrètement, nous pouvons diviser la BDD dont on dispose en deux, une pour l'apprentissage et l'autre pour la validation. Selon le problème posé, plusieurs indicateurs de performance du modèle sont utilisés. Ci-après quelques indicateurs classiques :

- La *Likelihood* : mesure la probabilité que la distribution des données soit issue de notre modèle. Cet indicateur mesure le pouvoir descriptif du modèle. Pour comparer plusieurs modèles, nous pouvons utiliser le test statistique *LRT (Likelihood Ratio Test)* permettant de décider si un modèle B améliore significativement (à un seuil choisi) la likelihood calculée par un premier modèle A.
- *L'AIC* : il s'agit d'une likelihood pénalisée par le nombre de paramètres utilisés dans le modèle. L'AIC intègre la complexité du modèle et par conséquent permet de lutter contre le risque du « *sur-apprentissage* » : il pénalise les modèles qui utilisent plus de variables explicatives (et qui risquent donc de coller à la population d'apprentissage).
- *L'AUC (Area Under ROC Curve)* : La courbe ROC permet de mesurer la performance d'un classifieur binaire. Elle représente le taux de vrais positifs par rapport aux faux positifs. Ci-dessous différentes ROC (à droite le hasard). Cet indicateur mesure la performance prédictive du modèle.

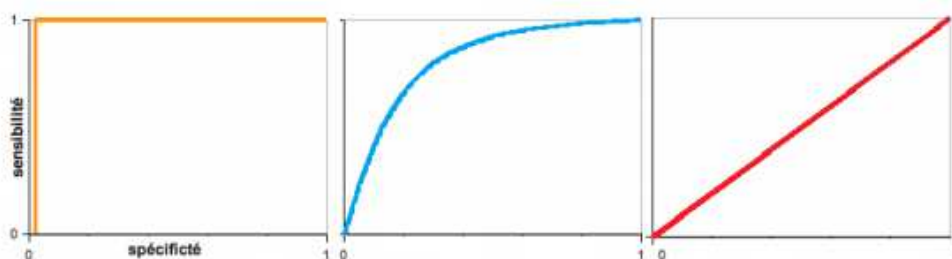


Figure 3: Courbes ROC des modèles parfait (à gauche), réel (au centre) et de l'aléatoire (à droite)

#### 4/ Amélioration de la performance prédictive du modèle lors de l'apprentissage :

Lors de la construction du modèle sur la population d'apprentissage, le data scientist peut utiliser certaines techniques permettant de lutter contre le risque du sur-apprentissage :

##### Technique de la cross-validation lors de l'apprentissage :

En introduisant de la variabilité sur la population d'apprentissage, cette technique permet de réduire l'erreur de la généralisation au moment d'apprendre le modèle. Elle consiste à diviser la population d'apprentissage en plusieurs sous bases d'apprentissage.

- A chaque étape de l'apprentissage, on exclut une sous base et on apprend le modèle sur le reste.
- On réitère cette étape plusieurs fois (le nombre de sous bases) en optimisant à chaque fois le problème de l'équilibre biais variance.

Avec cette technique, on obtient – à la fin de la phase de l'apprentissage- un modèle qui a appris en intégrant déjà la contrainte de la variabilité des données.

##### Sélection des variables :

Une autre manière d'augmenter la performance prédictive du modèle est la diminution du nombre de paramètres (la complexité du modèle) en sélectionnant les variables qui apportent le plus d'information au processus de prise de décision. La sélection de variables permet donc de lutter contre le risque du sur apprentissage.

Il existe plusieurs techniques pour la sélection des variables lors de l'apprentissage. On distingue deux types de méthodes de sélection de variables :

- Méthode discontinue : on choisit d'ajouter (ou de supprimer) un degré de complexité et on évalue l'impact de ceci sur les indicateurs de performance du modèle. La méthode discontinue se fait par itérations, et à la fin des différentes itérations, on garde la combinaison des paramètres qui fournit les meilleurs indicateurs. La fonction stepAIC de R permet d'effectuer une sélection de modèles par STEPWISE en optimisant l'AIC.
- Méthode continue : on intègre dans le programme d'optimisation un terme de pénalisation de la complexité. Par exemple, on peut utiliser la régression pénalisée (LASSO / RIDGE) qui permet de construire un modèle parcimonieux.

Dans le monde du « Big Data », où les variables explicatives se comptent en centaines, la sélection des variables est indispensable : elle permet de construire des modèles interprétables et de recentrer l'inertie autour des variables les plus importantes pour la prise de décision...

Pour conclure, rappelons que l'exercice de modélisation auquel l'actuaire data scientist devrait faire face, comporte d'autres aspects que cet article n'aborde pas. En particulier, les contraintes liées aux capacités de la machine et aux temps de calcul, à la logistique et le versionning des modèles dans le cadre d'un exercice de modélisation à plusieurs étapes et avec plusieurs intervenants, et surtout le défi du déploiement du modèle en production...

# Première partie : Challenge Kaggle Porto Seguros (Par Maryse Dama)

## 1/ Présentation du Challenge

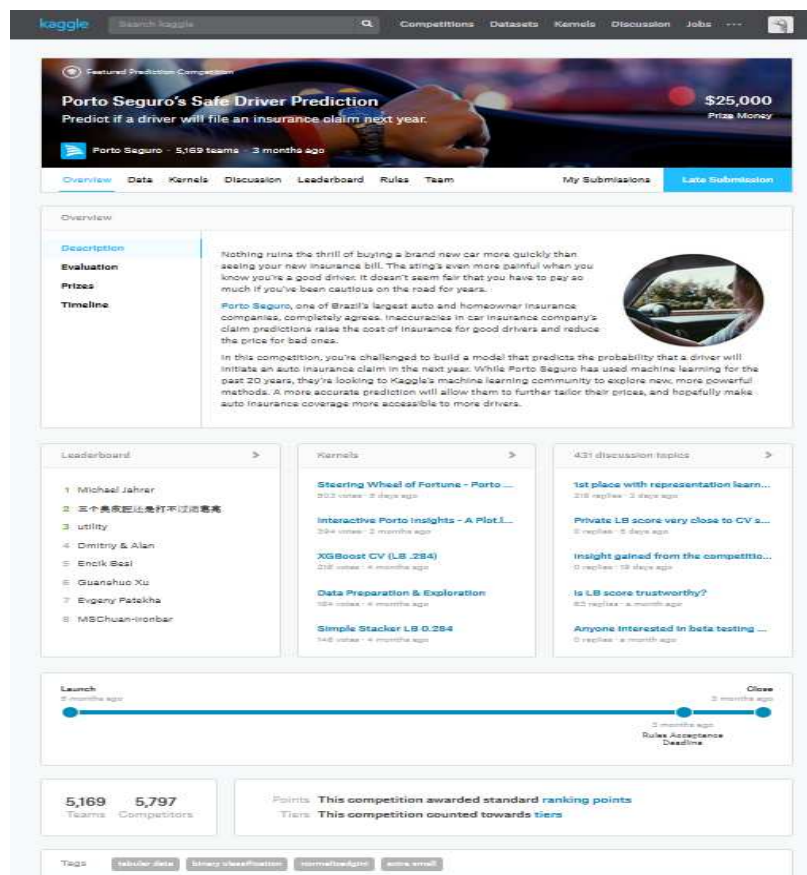


Figure 4 : Kaggle Porto Seguros

### Introduction

Porto Seguros<sup>3</sup> compagnie d'assurance brésilienne, lance le 29 septembre 2017 sur la plateforme de concours Kaggle<sup>4</sup> un challenge dont l'objectif est de calculer un score permettant de prédire la probabilité qu'un assuré ayant souscrit une police d'assurance ait un sinistre l'année suivante.

Les soumissions sont évaluées en utilisant le coefficient de Gini normalisé, et sont présentées sous la forme d'un fichier .csv contenant 2 colonnes : « id » (identifiant) et « target » (le score donné par l'algorithme). Pour chaque identifiant l'algorithme prédit une probabilité d'avoir un sinistre, comme suit :

<sup>3</sup> Porto Seguros est la troisième plus grande compagnie d'assurance au Brésil. Elle a été fondée en 1945 et compte plus de 13 000 employés. La compagnie offre principalement de l'assurance automobile, résidentielle, santé, vie (source Wikipédia).

<sup>4</sup> Kaggle est une plateforme web organisant des compétitions en datascience (source Wikipédia).

## Submission File

For each `id` in the test set, you must predict a probability of an insurance claim in the `target` column. The file should contain a header and have the following format:

```
id,target
0,0.1
1,0.9
2,1.0
etc.
```

Figure 5 : Format attendu pour le fichier de soumission

**Prix :** 1ere place 12.000\$, 2<sup>e</sup> place 8.000\$ et la 3<sup>e</sup> place 5.000\$.

Date du challenge : du 29 septembre au 29 novembre 2017 (2 mois).

### Description des données :

- Trois types de fichiers sont fournis par Porto Seguros :
  - o Le « train.csv », la base d'entraînement du modèle (113Mo, environ 595 000 observations anonymisées et 59 variables) ;
  - o Le « test.csv », la base de test du modèle (168Mo, environ 890 000 observations anonymisées et 58 variables) ;
  - o Et un exemple de table en sortie (508Ko).

### Leaderboard

Le Club Algo arrive **1265<sup>e</sup>** (avec 7 soumissions) sur **5169 équipes** en compétition et un score de :  
0.28960










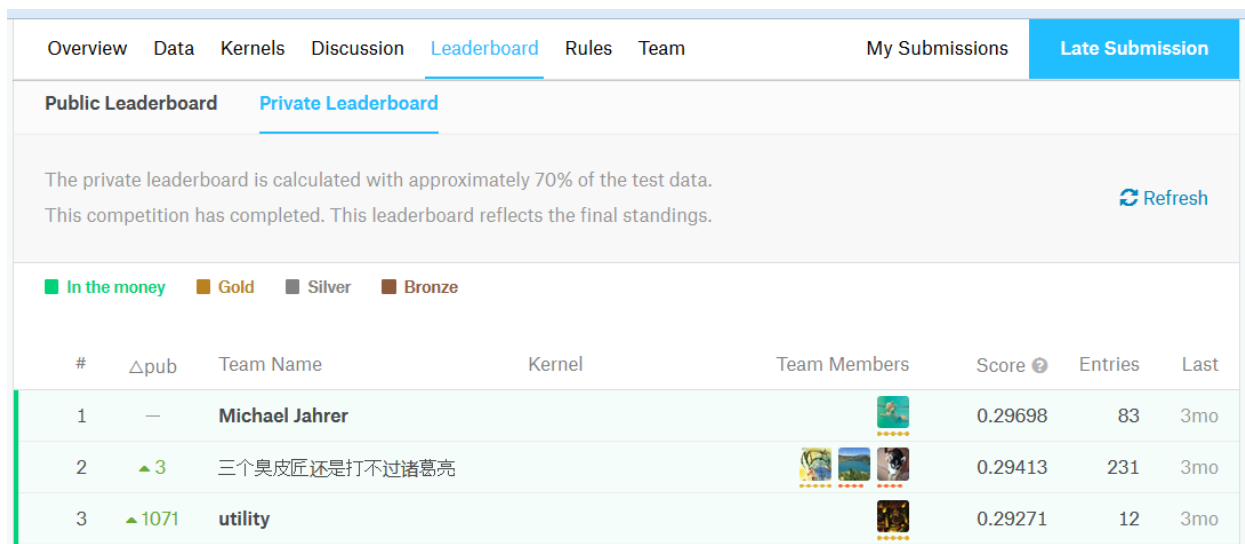
Overview	Data	Kernels	Discussion	Leaderboard	Rules	Team	My Submissions	Late Submission
1261	▲ 411	ChaosKin					 0.28960	17 3mo
1262	▲ 411	Vyom Bani					 0.28960	41 3mo
1263	▲ 411	Sichen					 0.28960	1 3mo
1264	▲ 412	Jeremy Han					 0.28960	5 3mo
1265	▲ 412	Club Algo IA					  0.28960	7 3mo
1266	▲ 412	Vijay					 0.28960	41 3mo
1267	▲ 413	yokonami					 0.28960	12 3mo
1268	▲ 413	RahulVashisht					 0.28960	17 3mo

Figure 6 : Place du Club Algo dans le leaderboard

Les trois premiers ont un score autour de 0.295 :



#	Δpub	Team Name	Kernel	Team Members	Score	Entries	Last
1	—	Michael Jahrer			0.29698	83	3mo
2	▲3	三个臭皮匠还是打不过诸葛亮			0.29413	231	3mo
3	▲1071	utility			0.29271	12	3mo

Figure 7 : Score des trois premiers du challenge

## 2/ Présentation de l'approche du Club Algo

Depuis quelques mois plusieurs solutions de machine learning automatisé ont vues le jour, nous citons ici les principales : MLBOX, EDGE-ML. Ces outils permettent de procéder automatiquement au « preprocessing/nettoyage/formatage » des données.

**MLBOX<sup>5</sup>** : La « MLBox » est une librairie<sup>6</sup> Python d'apprentissage automatique qui fournit les fonctionnalités suivantes :

- Lecture rapide et prétraitement des données distribuées / nettoyage / mise en forme
- Optimisation des hyper-paramètres
- Modèles prédictifs de pointe pour la classification et la régression (Deep Learning, Stacking, LightGBM, ...)
- Prédiction avec interprétation des modèles
- ...

**EDGE-ML<sup>7</sup>** : Edge-ML est une librairie d'algorithmes fondés sur l'approche mathématique MODL<sup>8</sup>. Elle constitue une chaîne de traitement de Machine Learning entièrement automatisée (auto ML) permettant d'entraîner un classifieur à partir de données numériques, catégorielles ou de type "séquence" (ex : sessions web, textes, logs...).

<sup>5</sup> Ne fonctionne qu'avec Linux

<sup>6</sup> Comment installer la MLBox avec Anaconda : [http://darques.eu/blog/index.php/2017/07/27/mlbox-a-short-regression\\_tutorial/](http://darques.eu/blog/index.php/2017/07/27/mlbox-a-short-regression_tutorial/)

<sup>7</sup> <http://www.edge-ml.com/index.php/fr/>

<sup>8</sup> MODL, de Marc BOULLÉ, chercheur au sein du groupe d'Orange Labs (France Télécome R&D)

**AUTO-ML de H2O<sup>9</sup>** : La plus connue. Le langage « AutoML » de H2O est utilisé pour automatiser une grande partie du workflow d'apprentissage automatique, ce qui inclut l'entraînement automatique et le réglage de nombreux modèles dans une limite de temps spécifiée par l'utilisateur. L'utilisateur peut également utiliser un critère d'arrêt basé sur une mesure de performances pour le processus « AutoML » plutôt qu'une contrainte de temps spécifique.

**Il a été décidé pour ce kaggle de tester le programme d'Axel de Romblay, compétiteur du concours qui utilise la MLBox, et qu'il a partagé avec les autres participants.  
Tous les travaux ont été mené sous des notebooks « Python ».**

## 2.1 Qu'est-ce que la MLbox<sup>10</sup>

La MLBOX est un pipeline entièrement automatisé qui se découpe en 3 sous-packages :

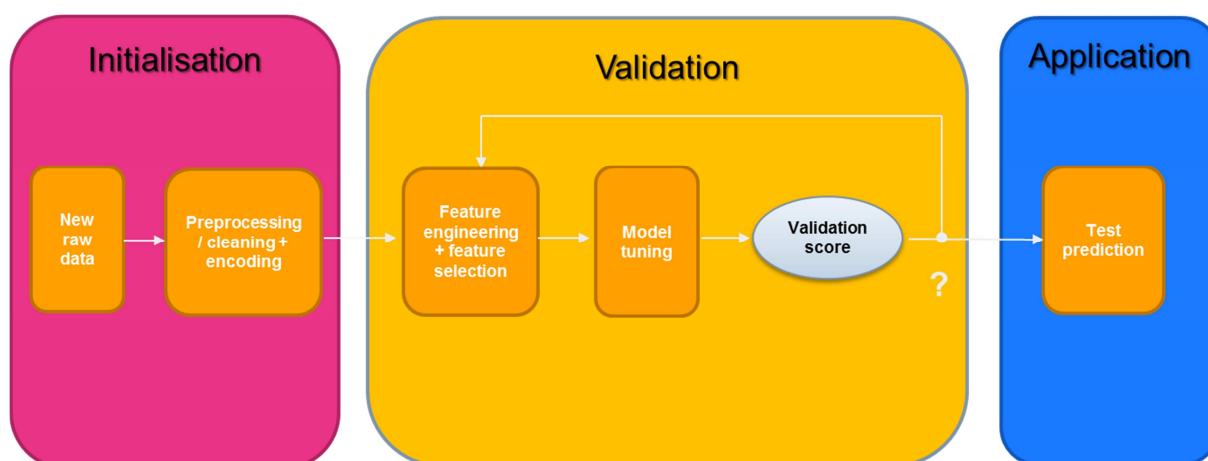


Figure 8 : Représentation des trois sous-packages de la MLBOX

### Initialisation :

Passage d'un **dataset brut** à un **dataset propre et numérisé** (donc exploitable).

- **Lecture des fichiers :**
  - Lecture de plusieurs fichiers (format csv, xls, json et hdf5)
  - **Détection automatique de la tâche** (classification binaire/multiclasse ou régression)
  - **Consolidation** de la base d'apprentissage et de test (suppression de variables manquantes...)
- **Preprocessing/cleaning :**
  - suppression des **duplicatas** (lignes identiques) et **variables constantes**
  - **Suppression de variables « instables »** (index, ...)
- **Encoding :**
  - conversion des variables sur un **format standard unique** (float si possible sinon str)
  - conversion des **variables complexes** (listes)

<sup>9</sup> <http://h2o-release.s3.amazonaws.com/h2o/master/3888/docs-website/h2o-docs/automl.html>

<sup>10</sup> <https://www.analyticsvidhya.com/blog/2017/07/mlbox-library-automated-machine-learning/>

- conversion des **dates** en timestamp. Extraction du mois, de l'année, jour et jour de la semaine...
- encoding de la **cible** (classification seulement)
- encoding des **variables catégorielles** (plusieurs stratégies proposées !)
- encoding des **valeurs manquantes** (plusieurs stratégies proposées !)

### Validation :

On teste des approches et techniques qui permettent d'augmenter la qualité du modèle et on valide.

- **Sur les données :**
  - **Feature engineering** : création de features à partir de réseau de neurones
  - **Feature selection** : méthodes filtres, enveloppes et régularisation L1
- **Sur la modélisation :**
  - Test d'un large panel de **modèles performants/transversaux** : Linéaires, Random Forest, XGBoost, LightGBM...
  - Possibilité d'agréger plusieurs modèles : **stacking**, boosting, bagging
  - **Tuning** de l'ensemble **des hyper-paramètres** du pipeline
- **Sur la validation :**
  - **Choix d'un large panel de métriques** : accuracy, log-loss, AUC (multi-classes), f1-score, MSE, MAE, ... **ou custom**
  - **Méthode de validation** :- nombre de folds, random state, ...

### Application :

On applique les étapes du **pipeline optimal** sur les nouvelles données de test non labelisées afin de **prédire** au mieux la sortie.

- **Prédiction :**
  - **Prédiction** des probabilités de chaque classe (classification) ou de la cible (régression)
  - **Sauvegarde** du modèle (format .obj python) et des prédictions (format .csv)
  - Estimation du **temps d'apprentissage** (pour la mise en production)
- **Interprétation des modèles :**
  - **Importance des variables** lors de l'apprentissage du modèle (enregistrement sous format .png)

## 2.2 La Mlbox par rapport aux autres

La MLBOX s'est focalisée essentiellement sur 3 méthodes :

- la détection des dérives entre la distribution de l'échantillon d'apprentissage et de l'échantillon test
- l'encodage des variables catégorielles
- l'optimisation des paramètres

## 2.3 La Mlbox appliquée au Kaggle

(cf. le script en annexe 1)

## 2.4 Autre approche proposée

par Meryem YANKOL (cf. annexe 2)

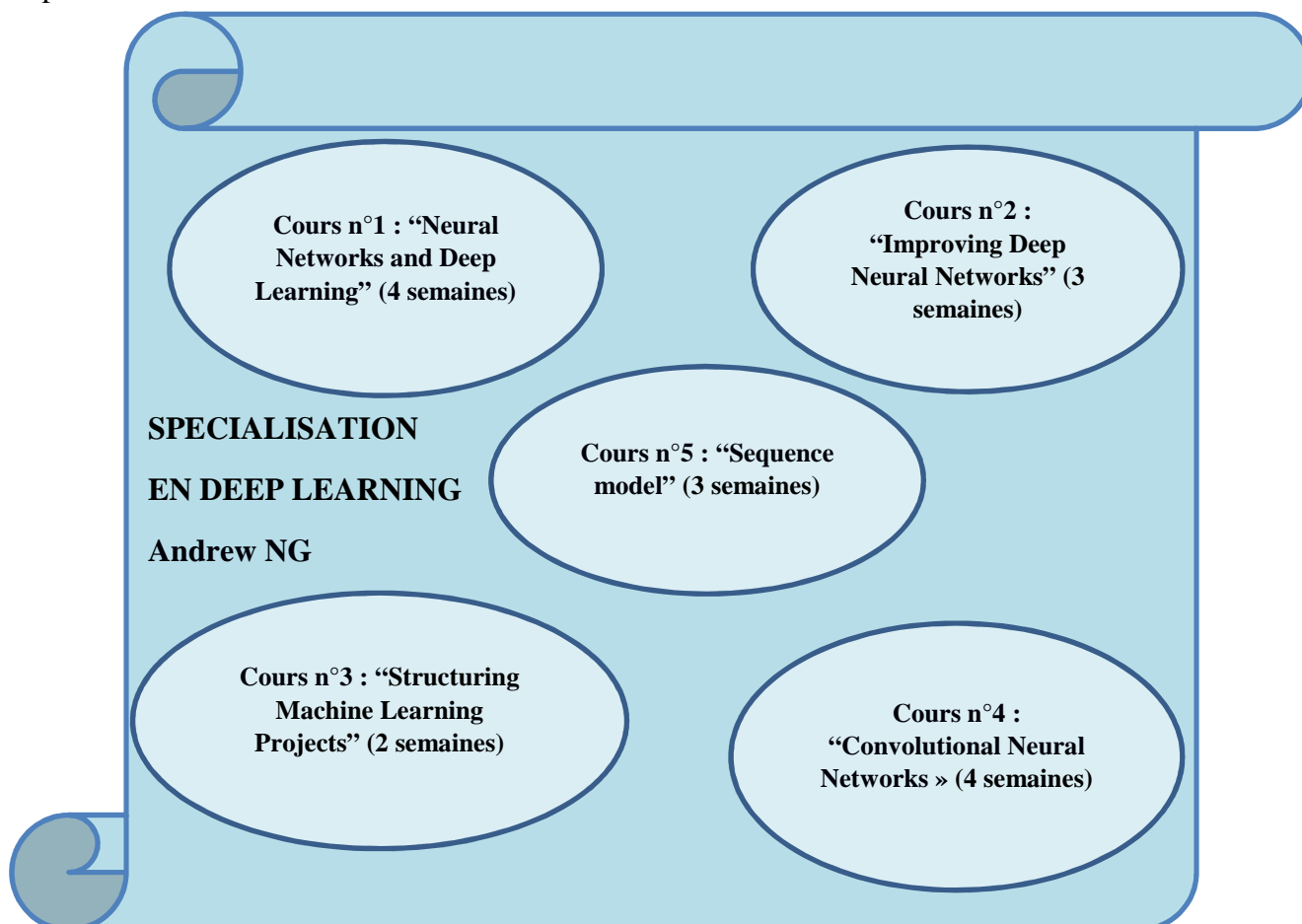
## Deuxième partie : Coursera<sup>11</sup> « Neural Network and Deep Learning » d'Andrew Ng

(Présenté par Nathalie Ramos)

Le Coursera de Spécialisation “Neural Network and Deep Learning<sup>12</sup>” a démarré en septembre 2017, et s’est achevé en février 2018. Cette spécialisation est constituée de 5 cours. Chaque cours est composé de 3 ou 4 semaines, (le nombre de semaines varie selon les cours. Pour valider un cours il est nécessaire de visionner de vidéos, réussir un quizz à la fin de chaque cours (environ une dizaine de questions) et compléter un (ou deux selon les cours) exercice de programmation sur un notebook Python directement accessible sur le hub (serveur) du site Coursera.

Les cours en ligne du Coursera sont payants, à ce jour (février 2018), le tarif est de 41€ par mois, et donne accès à plus de 2000 cours en ligne.

Chaque cours est indépendant et donne un certificat, l’ensemble des cours donne le certificat de spécialisation.



<sup>11</sup> Coursera est une entreprise numérique proposant des formations en ligne ouvertes à tous, fondée en 2012 par les professeurs d’informatique Andrew Ng et Daphné Koller de l’université Stanford, située à Mountain View, Californie. (source Wikipédia)

<sup>12</sup> Réseaux de neurones et apprentissage profond



### *La Spécialisation « Deep Learning »<sup>13</sup>(cf. annexe 3)*

En cinq cours, les bases de l'apprentissage en profondeur (deep learning) sont enseignées. Ce programme donne les clés nécessaires pour construire des réseaux neuronaux et explique comment mener à bien des projets d'apprentissage automatique.

Les réseaux de neurones suivants sont abordés : les réseaux de neurones convolutifs (CNN), les réseaux de neurones récurrents (RNN), LSTM (Long Short-Term Memory), Adam (algorithme d'optimisation), Dropout (algorithme de régularisation qui permet de réduire le risque de surapprentissage), l'initialisation de Xavier/He, et plus encore...

Des études de cas sont proposés concernant différents domaines : La santé, le véhicule autonome, la lecture en langue des signes, la génération de musique et le traitement du langage naturel.

Ce programme propose non seulement de maîtriser la théorie, mais aussi de voir comment elle est appliquée dans des cas concrets. Les exercices de programmation sont pratiqués avec Python (Jupyter en connexion sur le hub de Coursera) et avec TensorFlow.

Des interviews d'intervenants (optionnels) professionnel en la matière sont également proposées. Ces spécialistes partagent leurs histoires personnelles et donnent des conseils de carrière.

---

<sup>13</sup> Source : <https://fr.coursera.org/specializations/deep-learning>

## Troisième partie : Le Deep Learning au service de l'assureur non-vie (par Victor Reutenauer<sup>14</sup>)

### Contexte

Dans le cadre du Club Algo animé par Nathalie Ramos, différentes études ont été réalisées autour de l'utilisation du Deep Learning pour l'apprentissage supervisé.

### Quelques mots sur les réseaux profonds

L'essor des cartes graphiques depuis le début des années 2010 a permis l'utilisation de techniques dites d'*apprentissage automatique* pour la classification d'images notamment grâce à l'*apprentissage profond*.

Ses réseaux sont constitués de plusieurs couches. Les données de sorties de chaque couches sont les données d'entrées de la couche suivante. Chaque couche est constituée de plusieurs entités de calculs distinctes qu'on appelle communément des neurones. Il existe deux types principaux de neurones et donc de couches. Ceux qui sont linéaires et agrègent les données d'entrée de manière additive sous forme de combinaison linéaire. Et ceux qui sont non linéaires, principalement sous forme de fonction du type  $1_{x > K}$  ou du type  $\max(x-K, 0)$ .

Une dernière couche minimise les erreurs quadratiques de prédiction entre les labels d'apprentissage que le réseau cherche à prédire. Les poids des combinaisons linéaires des neurones du premier type cité sont les paramètres à calibrer pour obtenir un réseau résolvant un problème de classification.

Ces poids sont calibrés par une descente du gradient dites stochastique car à chaque itération seul une sous partie des données d'apprentissage (photo d'entrée et label de sortie à prédire) sont utilisés. De la même manière que dans la technique de descente du gradient de Newton, il est nécessaire d'évaluer la dérivée de la fonction cible donnée par la dernière couche en fonction de ces paramètres à trouver.

Une rétro-propagation du gradient entre les différentes couches permet de calculer le vecteur des dérivées partielles en fonction de chaque paramètre à optimiser.

Bien que le problème ne soit pas convexe et ne converge pas vers un minimum global, d'une part il est globalement convexe et donc ne diverge pas, d'autre part les tests empiriques ont montré que les minimum locaux atteints sont en général assez bons.

Le Club Algo a su vérifier ces propriétés en calibrant des réseaux de neurones pour effectuer des tâches qu'un être humain bon connaisseur des modèles de voitures est capable de réaliser. Différentes bibliothèques open source ont été utilisées pour cela, *Caffe* et *Keras* notamment en utilisant les ressources informatiques fournis entre autre par la société Fotonower<sup>15</sup>.

<sup>14</sup> Ce document reflète le point de vue de ses auteurs et n'engage en aucun cas l'Institut des Actuaire. victor@fotonower.com

<sup>15</sup> Start-up de reconnaissance d'image fondée par Victor Reutenauer.

## Détection de modèle et de marques de véhicule

(cf. annexe 4)

Le cas testé en détail a consisté à chercher à classier automatiquement des photos de véhicules de tourisme en fonction de leur marque ou du modèle précis du véhicule.

Plusieurs dizaines de milliers d'exemples de photos ont été utilisés pour l'apprentissage répartis en plusieurs centaines de modèle de véhicule. La précision était d'environ 80% pour le choix le plus probable donné par les réseaux calibrés et dépassait 90% en considérant les trois choix les plus probables, ceci pour une seule photo de chaque véhicule. Des tests ont mis en avant le fait que de ne garder que les photos des véhicules bien visibles de l'avant donnaient de meilleurs résultats à partir d'une photo.

D'autre part, il est apparu clairement qu'augmenter le nombre de données d'apprentissage permettait de réaliser de meilleurs scores de réussite. Une extension de ces méthodes consiste à utiliser plusieurs photos d'un même véhicule.

## Événements 2017 du Club Algo

*(cf annexe 4)*

Le Club Algo a effectué deux présentations publiques durant le deuxième semestre 2017.

D'une part au sein des locaux de Dataiku en septembre 2017 et d'autre part durant la conférence 100% Actuaire - 100% Datascience en Novembre.

Elles ont permis de présenter le travail effectué par les membres du Club Algo et de rendre public les résultats. Ces présentations montraient les aspects théoriques des réseaux de neurones, l'application de ces méthodes et même une démonstration en live de la reconnaissance de modèles de véhicules.

## Conclusion générale

L'objectif du Club Algo, outre d'échanger sur l'utilisation des algorithmes, est de tenter de rapprocher le monde de l'actuariat et celui de la data science en traitant des problématiques classiques, ou moins classiques, de l'actuaire avec les outils innovants offerts par les algorithmes du machine learning, et en appliquant ces techniques à des cas d'usage concrets pour l'assureur.

Pour l'année 2018, le Club Algo s'est fixé des objectifs ambitieux, comme l'interprétabilité des modèles, le Pricing Game 4<sup>e</sup> saison d'Arthur Charpentier, la continuation des travaux sur la reconnaissance d'image ainsi que la programmation de drone.

## Annexes

## ANNEXE 1 – PROGRAMME D'AXEL DE ROMBLAY AVEC PYTHON

### a) input et import

```
from mlbox.preprocessing import *
from mlbox.optimisation import *
from mlbox.prediction import *

paths = ["../input/train.csv", "../input/test.csv"]
target_name = "target"
```

### b) Lecture et nettoyage des données

```
rd = Reader(sep = ",")
df = rd.train_test_split(paths, target_name)
```

### c) Suppression des variables instables

```
dft = Drift_thresholder()
df = dft.fit_transform(df)
```

### d) Tuning de l'ensemble des hyper-paramètres du pipeline

```
def gini(actual, pred, cmpcol = 0, sortcol = 1):
    assert( len(actual) == len(pred) )
    all = np.asarray(np.c_[ actual, pred, np.arange(len(actual)) ],
dtype=np.float)
    all = all[ np.lexsort((all[:,2], -1*all[:,1])) ]
    totalLosses = all[:,0].sum()
    giniSum = all[:,0].cumsum().sum() / totalLosses
    giniSum -= (len(actual) + 1) / 2.
    return giniSum / len(actual)
def gini_normalized(a, p):
    return np.abs(gini(a, p) / gini(a, a))
opt = Optimiser(scoring = make_scorer(gini_normalized, greater_is_better=True,
needs_proba=True), n_folds=2)
space = {
    'est__strategy': {"search": "choice",
                    "space": ["LightGBM"]},
    'est__n_estimators': {"search": "choice",
                        "space": [700]},
    'est__colsample_bytree': {"search": "uniform",
                            "space": [0.77, 0.82]},
    'est__subsample': {"search": "uniform",
                     "space": [0.73, 0.8]},
    'est__max_depth': {"search": "choice",
                     "space": [5, 6, 7]},
    'est__learning_rate': {"search": "uniform",
                         "space": [0.008, 0.02]}
}
params = opt.optimise(space, df, 7)
```

### e) La prédiction

```
prd = Predictor()
prd.fit_predict(params, df)
```

### f) Le fichier à soumettre

```
submit = pd.read_csv("../input/sample_submission.csv", sep=',')
preds = pd.read_csv("save/"+target_name+"_predictions.csv")
submit[target_name] = preds["1.0"].values
submit.to_csv("mlbox.csv", index=False)
```

*ANNEXE 2 – PRESENTATION DES TRAVAUX DE MERYEM YANKOL SUR LE CHALLENGE  
PORTO SEGUROS*



Kaggle\_PortoSeguros  
\_Meryem



## *ANNEXE 3 – SPECIALISATION “DEEP LEARNING” d’Andrew Ng*

### *Cours n° 1 - Neural Networks and Deep Learning - 4 semaines*

Introduction au Deep Learning  
Logistic regression as a Neural Network  
Python and Vectorization  
Shallow Neural Network  
Deep Neural Network

### *Cours n° 2 - Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization - 3 semaines*

Setting up your Machine Learning Application  
Regularizing your neural network  
Setting up your optimization problem  
Optimization algorithms  
Hyperparameter tuning  
Batch Normalization  
Multi-class classification  
Introduction to programming frameworks (Tensorflow)

### *Cours n°3 - Structuring Machine Learning Projects - 2 semaines*

Machine learning strategy  
Introduction to ML strategy  
Setting up your goal  
Comparing to human-level performance  
Error analysis  
Mismatched training and dev/test set  
Learning from multiple tasks  
End-to-end deep learning

### *Cours n°4 - Convolutional Neural Networks - 4 semaines* *Convolutional Neural Networks*

Case studies  
Practical advices for using ConvNets  
Detection algorithms  
Face recognition  
Neural style transfer

### *Cours n°5 - Sequence models - 3 semaines*

Recurrent Neural Networks  
Introduction to word Embeddings  
Learning Word Embedding : Word2vec & Glove  
Applications using Word Embeddings  
Various sequence to sequence architectures  
Speech recognition - Audio data

*ANNEXE 4 – PRESENTATION DU PETIT-DEJEUNER DE SEPTEMBRE 2017 DANS LES  
LOCAUX DE DATAIKU*



Presentation\_pdj\_Clu  
bAlgo\_sept2017