Année universitaire	**2019-2020**		
Filière	**Data Science**	**Année**	**M2**
Matière	**Machine Learning**		
Enseignant	**Haytham Elghazel**		
Intitulé TD/TP :	**Atelier 3 : Détection d'anomalies avec Python**		
Contenu	• **Détection d'anomalies** • **Isolation Forest**		

Dans cet atelier pratique, vous allez expérimenter des algorithmes de traitement de données pour répondre à différents problèmes liés à la détection d'anomalies avec le langage **Python**.

Pour lancer le notebook Python, il faut taper la commande **jupyter notebook** dans votre dossier de travail. Une fenêtre va se lancer dans votre navigateur pour ouvrir l'application Jupyter. Créer un nouveau notebook Python et taper le code suivant dans une nouvelle cellule :

```
import numpy as np
np.set_printoptions(threshold=np.nan)
import pandas as pd
import warnings
import matplotlib.pyplot as plt
warnings.filterwarnings('ignore')
```

La détection d'anomalies (dite aussi détection d'outliers) est une tâche de l'apprentissage automatique qui consiste à déceler dans les données, les instances (individus) ayant un comportement différent (inhabituel) des autres instances de la base dites normales. Dans le cas de la détection des fraudes par exemple, toute dépense très étrange par carte de crédit est suspecte. Les variations possibles sont si nombreuses et les exemples de formation si rares, qu'il n'est pas possible de savoir à quoi ressemble une activité frauduleuse ou un incident. L'approche de la détection des anomalies consiste simplement à apprendre à quoi ressemble l'activité normale (à l'aide d'un historique de transactions supposées non-frauduleuses) et d'identifier tout ce qui est très différent. Différentes algorithmes ont été proposées dans la littérature dont certaines sont supervisées et d'autres non supervisées. Pour plus de détails considérant ces approches, vous pouvez vous référer vers les liens suivants : https://ngoix.github.io/nicolas_goix_osi_presentation.pdf et https://en.wikipedia.org/wiki/Anomaly_detection

Sickit-learn propose différentes approches pour la détection d'anomalies (http://scikit-learn.org/stable/modules/outlier_detection.html). Nous nous intéressons ici à une approche non supervisée appelée **Isolation Forest** (http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html). Nous allons appliquer cette approche sur deux jeux de données **"mouse.txt"** (https://elki-project.github.io/datasets/) et le fichier **"creditcard.csv"** du challenge Kaggle (https://www.kaggle.com/dalpozz/creditcardfraud).

1. Sur la base de données Mouse

Ce fichier contient 500 instances décrites par deux variables x1 et x2 représentant des points de la tête de Mickey Mouse. Les 10 dernières instances du fichier sont aberrantes (*outliers*).

- Télécharger ce jeu de données et analyser le.
- Donner une représentation graphique des données (***matplotlib.pyplot***).
- Appliquer la technique Isolation Forest pour détecter les anomalies dans ce jeu de données.
- Modifier votre représentation graphique précédente pour visualiser les données aberrantes.

2. Sur le jeu de données des cartes de crédits

L'objectif dans cette partie est de déceler les fraudes dans les transactions de cartes bancaires. Pour plus d'informations sur ce jeu de données, vous pouvez visiter le lien suivant (https://www.kaggle.com/dalpozz/creditcardfraud).

Ce jeu de données est très déséquilibré avec seulement 0.172% de fraudes (492 fraudes sur 284 807 transactions). A partir de ce jeu de données, garder aléatoirement 5000 transactions de cartes normales (***classe 0***) et toutes les transactions aberrants (***classe 1***). On pourra également travailler sur la totalité des données.

- Préparer ce jeu de données (ne pas utiliser la variable Time).
- Appliquer la technique Isolation Forest pour détecter les anomalies dans votre jeu de données.
- Retourner la matrice de confusion et analyser les instances aberrantes détectées.