ISIFC 3, Cryptographie.

Jean-François COUCHOT
couchot[arobase]femto-st[point]fr

Examen d'octobre 2023

Organisation

Cette épreuve est à réaliser par groupe de 2. Vous n'avez pas le droit de vous faire aider ni de communiquer avec quiconque. Par contre, vous pouvez utiliser toutes les sources d'information possibles : votre cours, vos TDs, et notes personnelles, des exemples en ligne. Vous n'avez pas le droit de partager des éléments avec quiconque. La première chose à faire est de renseigner sur la feuille de présence votre numéro de binôme.

Le jeu de données original est téléchargeable à l'URL

https://drive.google.com/file/d/13YQWnhrEW0h0TYMDEroa7qP-UA4Oa9Wz/view?usp=sharing.

Une description des attributs et d'une méthode d'apprentissage sur ces données est accessible à l'URL

https://towardsdatascience.com/heart-disease-uci-diagnosis-prediction-b1943ee835a7

Les réponses textuelles à cet examen doivent être consignées dans ce fichier que vous nommerez `Nom1Nom2.odt`.

Tout le code python que vous rédigerez est à réaliser dans un jupiter-notebook que vous téléchargerez à la fin de la séance au format `ipynb` et que vous nommerez `Nom1Nom2.ipynb`

A la fin de la séance, construire une archive `Nom1Nom2.zip` avec tous les fichiers demandés :

— `heart4.csv`.
— `Nom1Nom2.deid`.
— `csvNom1Nom2.json`
— `Nom1Nom2.ipynb`
— `Nom1Nom2.odt`

et l'envoyer par mail.

Dans cet examen, vous allez avoir deux rôles successifs. En premier lieu vous serez le.la docteur.e (section 1) capable tout d'abord d'assainir ces données par 4-anonymat puis de transmettre ensuite une version chiffrée avec AES de celles-ci à un.e data-scientist.e. En second lieu, vous serez ce.tte data-scientist.e. (section 2) qui déchiffrera tout d'abord un fichier de données assaini fourni (et un peu nettoyé) puis qui mettra en place un algorithme d'apprentissage sur ces données.

1 Rôle de docteur.e

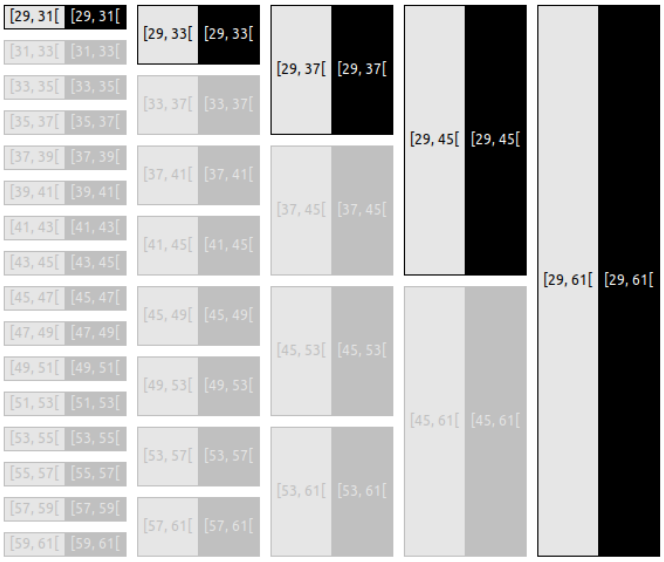
1.1 Assainir avec du 4-anonymat

Dans cette partie on importera dans ARX tout d'abord le jeu de données préalablement téléchargé. Tous les attributs sont numériques. On prendra donc garde à ce que l'attribut "oldpeak" soit bien un attribut décimal, dont le séparateur est le pont (".") comme en langue anglaise (EN).

Exercice 1.1 (Définition et généralisation des quasi-identifiants).

Deux attributs age *et* sex *seront considérés comme des quasi-identifiants dont on vous demande de créer des hierarchies de généralisation.*

— *Pour l'âge, définir une hierarchie de généralisation comme celle donnée ci dessous, c'est à dire avec des intervalles dont l'amplitude est* $2^0, 2^1, 2^2, \ldots, 2^5$.



— *Pour le genre, proposer un seul niveau qui regroupe le genres féminin et masculin* (0 *et* 1).

Exercice 1.2 (Générer un modèle 4-anonyme).

Choisir comme modèle d'assainissement le 4*-anonymat autorisant 5% de suppression de données et choisir la génération qui est proposée par défaut.*

1. *Quelle est l'amplitude des intervalles pour chaque âge ? Justifier (par une capture d'écran éventuellement).*
2. *Le genre est-il généralisé ? Justifier (par une capture d'écran éventuellement)*
3. *Dans le jeu de donné engendré, combien reste-t-il de personnes de 74 ans et plus ? Justifier (par une capture d'écran éventuellement)*
4. *Exporter le jeu de données sous la forme d'un fichier* `heart4.csv`.
5. *Enregistrer le projet Arx sous le nom* `Nom1Nom2.deid`.

1.2 Envoi de données chiffrées

Exercice 1.3 (Partage d'une clé pour AES).

1. *Récupérer la clé AES qui a été générée pour votre binôme. Quelle taille fait-elle ? Justifier.*
2. *Dans une démarche de chiffrement par AES, expliquez en détail comment aurait dû être partagée cette clé entre votre binôme et votre enseignant.*

Exercice 1.4 (Chiffrement du fichier `heart4.csv` avec AES).

1. *Pourquoi privilégie-t-on le mode GCM dans AES ?*
2. *Écrire le code qui permet de construire un fichier* `json` *nommé* `csvNom1Nom2.json` *qui contient les éléments suivants :*
   — *un* `nonce` *et un* `header` *comme usuellement ;*
   — `ciphert` *et un* `tag` *où la valeur associée à* `ciphert` *est le chiffré du fichier* `heart4.csv` *et celle associée à* `tag` *est le code de vérification, l'ensemble étant généré par AES selon le mode GCM. Pour transformer votre fichier* `heart4.csv` *en une séquence d'octets (pour pouvoir le chiffrer ensuite), vous pouvez utiliser le code suivant :*

```
import pandas as pd
df = pd.read_csv('heart4.csv')
csv_byte = df.to_csv(index=False).encode('utf-8')
```

3. *Exécuter ce code et enregistrez le fichier* `csvNom1Nom2.json`.

2 Rôle de data-scientist.e

2.1 Récupération du contenu du fichier `heart4.csv`

Exercice 2.1 (Déchiffrement de fichier avec AES (GCM)).

1. *Récupérer le fichier* `json` *contenant les données chiffrées qui vous a été transmis en même temps que la clé.*
2. *Montrer que les clefs de ce fichier* `json` *sont* `['nonce', 'header', 'ciphert', 'tag']`*. On pourra par exemple ouvrir ce fichier au moyen d'un éditeur de texte.*
3. *Écrire le code permettant de déchiffrer son contenu sachant qu'il a été chiffré avec la clé transmise selon AES en mode GCM et stocker ce contenu dans une variable* `pt`*.*

Exercice 2.2 (Construction du dataframe à partir de `pt`).

1. *Écrire le code qui construit une variable* `df` *de type* `DataFrame` *à partir de* `pt`*. On pourra exploiter le code suivant :*

```
import io
pt = ... # ce que vous avez fait à l'exo précédent.
df = pd.read_csv(io.StringIO(pt.decode('utf-8')))
```

2.2 Apprentissage de la classe sur des données anonymisées

L'objectif est de mettre en place un algorithme d'apprentissage permettant d'inférer la valeur de l'attribut **`class`** à partir des 12 autres colonnes et ce, au moyen de 2 démarches d'apprentissage supervisés.

Exercice 2.3 (Construction des datasets d'entrainement et de test).

1. *Écrire le code qui permet d'avoir dans les variables* `X` *et* `y` *les 12 premières colonnes et celle de la sortie respectivement.*
2. *Écrire le code qui permet de partager* `X` *et* `y` *en* `X_train`*,* `X_test`*,* `y_train`*,* `y_test` *respectivement de sorte que (*`X_train`*,* `y_train`*) représente 80% du jeu de données et que cette partition soit reproductible.*

Exercice 2.4 (Apprentissage supervisé par méthode naïve bayésienne gaussienne).

1. *Écrire le code qui permet de mettre en place un apprentissage supervisé de la valeur de l'attribut classe* **`class`** *au moyen de la méthode naïve bayésienne gaussienne.*
2. *Écrire le code qui permet d'évaluer la précision de la méthode.*
3. *Analyser cette précision.*

Exercice 2.5 (Apprentissage supervisé par une régression linéaire multiple).

1. *Écrire le code qui permet de mettre en place un apprentissage supervisé de la valeur de l'attribut* **`class`** *au moyen d'une régression linéaire multiple.*
2. *Écrire le code qui permet d'évaluer la précision de la méthode. On prendra garde au fait qu'il y a 4 classes à deviner (0,1,2,3).*
3. *Analyser cette précision et la comparer avec la précédente.*

N'oubliez d'ajouter votre notebook `Nom1Nom2.ipynb` *et ce document* `Nom1Nom2.odt` *à votre archive.*