

## Applications de l'informatique quantique

### Consignes pour le projet de fin de module

#### ENONCE DU CAS D'ETUDE

Le dataset du Blood Transfusion Center de Taiwan (source: Kaggle) contient des informations concernant 748 donneurs de sang, collectées dans le but de déterminer quels critères peuvent permettre d'identifier les donneurs les plus susceptibles de donner à nouveau.

Pour chaque donneur, 4 informations sont recueillies :

- **Recency** : le nombre de mois écoulés depuis le dernier don du sang.
- **Frequency** : le nombre total de dons effectués par le donneur
- **Monetary** : le volume total (en centimètres cubes) de sang donné
- **Time** : le nombre de mois écoulés depuis le premier don du sang

Enfin, lors d'un passage du bus de collecte de sang en 2007, on a noté pour chaque personne si elle a donné son sang ou non. **Cette information sera la cible à prédire de notre étude.**

#### PROBLEME 1 : CLASSIFICATION

*Pour ce problème uniquement, vous créerez un dataset équilibré (50% donneurs, 50% non donneurs) de 40 points d'entraînement, et 20 points de test, par un tirage aléatoire sans remise dans le dataset de base. Cette taille de dataset devrait permettre de réaliser les calculs quantiques dans un temps raisonnable.*

Vous proposerez deux modèles, basés sur deux features map quantiques différentes, pour réaliser la prédiction du don du sang en fonction des 4 features proposées. La prédiction sera réalisée sur un simulateur idéal, puis sur un simulateur réaliste (on pourra par exemple utiliser le modèle `ibm_brisbane` du cours, ou tout autre simulateur bruité de votre choix)

La performance de ces modèles sera évaluée sur le jeu de test, et comparée avec les performances d'un modèle classique de votre choix.

On vise notamment, dans ce problème, à répondre aux questions suivantes : *le choix de la feature map a-t-il un impact important sur la performance de classification ? Comment cette performance se compare-t-elle aux performances classiques ? Comment le bruit affecte-t-il ces performances ?*

#### PROBLEME 2 : OPTIMISATION

*Pour ce problème, vous travaillerez sur l'ensemble du dataset.*

L'établissement du don du sang se demande quels sont les critères (parmi les 4 proposés) qui prédisent le mieux la tendance à donner son sang ou non, afin de mettre en place des règles métier pour recontacter certains anciens donneurs. On cherche à éviter d'utiliser des features deux à deux redondantes.

Vous formulerez ce problème sous forme QUBO (NB : on ne demande pas d'entraîner de modèle de classification ici), puis proposerez une résolution à l'aide d'un algorithme QAOA, pour présenter les solutions optimales utilisant 1, 2 et 3 critères. (NB : il y a donc trois solutions attendues)

L'algorithme sera exécuté sur simulateur idéal, puis sur un simulateur réaliste (on pourra par exemple utiliser le modèle `ibm_brisbane` du cours, ou tout autre simulateur bruité de votre choix), et ses performances seront comparées à celles d'une approche brute-force.

On vise notamment, dans ce problème, à répondre aux questions suivantes : *comment formuler le problème efficacement pour le résoudre via QAOA ? Les performances obtenues sont-elles compétitives par rapport à une brute-force classique, et sont-elles affectées par le bruit ?*

## **CONSIGNES GENERALES**

Par binôme, vous proposerez un notebook jupyter présentant et résolvant le problème 1 ou 2 (un seul problème de votre choix à traiter).

Le notebook devra être lisible et pédagogique : l'idée principale est de faire comprendre votre démarche et présenter les implications de vos expériences. Vous pourrez par exemple vous appuyer sur les questions proposées dans les énoncés pour guider votre réflexion, et utiliser les outils Jupyter (e.g. insertion de texte entre les cellules de code) pour guider le lecteur.

Le barème de notation est le suivant (pour une note totale sur 20):

- **7 points** pour le caractère pédagogique du notebook (explications claires et concises du contexte, de la démarche et des conclusions à tirer ; organisation logique des éléments du notebook).
- **3 points** pour la propreté du code (noms de variables utilisés, commentaires éventuels, pertinence des imports...)
- **4 points** pour un élément spécifique au problème choisi :
  - pour le problème de classification, il s'agit du choix des feature maps (et de la justification associée)
  - pour le problème d'optimisation, il s'agit de la formulation du problème sous forme QUBO.
- **2 points** pour l'application de la démarche appropriée pour mettre en oeuvre l'algorithme quantique (mise en oeuvre d'un classifieur quantique ou d'un optimiseur quantique)
- **2 points** pour la comparaison des performances observées avec les performances classiques (il s'agit ici de mettre en oeuvre la comparaison de manière pertinente ; l'obtention de performances inférieures à celles d'une approche classique ne sera pas pénalisée).
- **2 points** pour la présentation de l'influence du bruit sur les performances obtenues.