Rapport Projet Intelligence Artificielle

Ce travail a été fait par Baye Lahad MBACKE et Mahamat OUAGAL

Etat d'avancement* : *L'ensemble des questions ont été répondues*(fonctionnel*)

2- Données

1- Le prix est encodé en 4 classes (0, 1, 2, 3), représentant respectivement les gammes de prix : "low cost", "mid range", "high range", "premium".

2-La valeur de d pour les données utilisées dans ce projet est **20**, car on a supprimé l'index et le label price_range

3-Il y a **4 classes** représentées dans le **dataset (0, 1, 2, 3)** donc **K = 4.**

4- Le nombre d'exemples dans chacune des classes pour les jeux de données d'apprentissage et de test est présenté dans les tableaux suivants :

Pour raw_train.csv :

```
train_data = pd.read_csv('raw_train.csv')
train_data['price_range'].value_counts()
```

```
2    304
0    302
1    298
3    296
Name: price_range, dtype: int64
```

Commentaire : Cela indique que notre jeu de données est presque parfaitement équilibré, avec un nombre presque égal d'exemples dans chaque classe de price_range. C'est une bonne situation car cela signifie qu'aucune classe n'est sous-représentée ou sur-représentée.

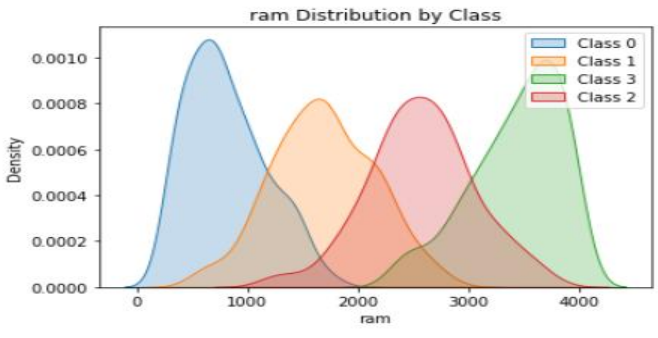
Pour raw_test.csv :

```
test_data = pd.read_csv('raw_test.csv')
test_data['price_range'].value_counts()
```

```
3    204
1    202
0    198
2    196
Name: price_range, dtype: int64
```

Commentaire : Cela indique que notre jeu de données de test est également presque parfaitement équilibré, avec un nombre presque égal d'exemples dans chaque classe de prix

6- L'attribut le plus discriminant devrait être celui dont les distributions de densité sont les plus séparées entre les différentes classes de prix. On trouve que c'est la **RAM**



3-Évaluation

1. L'intérêt d'évaluer un modèle sur un jeu de données différent de celui ayant servi à l'apprentissage réside dans la capacité de mesurer la performance du modèle sur des données inédites. Cela permet de vérifier si le modèle est capable de généraliser à partir des exemples qu'il n'a pas encore vus. Évaluer un modèle uniquement sur les données d'apprentissage pourrait donner une fausse impression de performance élevée, car le modèle pourrait simplement mémoriser les données(overfitting) d'apprentissage sans apprendre à généraliser. L'utilisation d'un jeu de données de test distinct permet de vérifier si le modèle est réellement capable de faire des prédictions précises sur des données nouvelles et inédites.

2. Indiquer la valeur de chacun des Ti lorsque EP = 1.

Si EP = 1, cela signifie que tous les exemples ont été correctement classifiés. Par conséquent, pour chaque classe i, tous les exemples de la vérité terrain i ont été correctement classés comme i. Cela signifie que Ti, le taux de bonne classification pour chaque classe i, serait également de 1 pour toutes les classes. Donc si EP = 1 alors Ti = 1

3. Même question lorsque EP = 0

Si EP est égal à 0, cela signifie que tous les exemples ont été mal classifiés. Par conséquent, pour chaque classe i, aucun des exemples de la vérité terrain i n'a été correctement classé comme i. Cela signifie que Ti serait également de 0 pour toutes les classes. Donc si EP = 0 alors Ti = 0

4.1 Apprentissage de l'auto-encodeur

1-L'entropie croisée est utilisée principalement pour les problèmes de classification, où les sorties sont des probabilités distribuées sur les différentes classes. Dans le cas de l'auto-encodeur, l'objectif ici est d'encoder les données d'entrées, et non de prédire des classes. Par conséquent, l'entropie croisée n'est pas adaptée comme critère d'optimisation pour un auto-encodeur.

2- La fonction d'activation tanh est utilisée dans la dernière couche de l'auto-encodeur. La fonction tanh prend des valeurs dans l'intervalle [-1, +1]. Pour que la sortie de l'auto-encodeur puisse être identique à son entrée, il est nécessaire de normaliser les données d'entrée entre -1 et +1.

4- L'apprentissage d'un auto-encodeur est non supervisé car dans le cas d'un auto-encodeur, les "étiquettes" auxquelles le modèle est formé pour prédire sont les mêmes que les données d'entrée - il n'y a pas d'étiquettes séparées fournies pour le modèle à apprendre. C'est pourquoi on dit que l'apprentissage est non supervisé.

4.2 Apprentissage de l'arbre de décision dans l'espace latent

3. On s'attend à ce que l'arbre de décision appris dans l'espace latent soit moins profond qu'un arbre appris sur les données brutes, car l'espace latent réduit la dimensionnalité des données et représente une version plus condensée de l'information. Cela facilite la séparation des classes et permet de construire un arbre de décision plus simple et moins profond.