

README : Classification des Performances des Étudiants

Introduction

Ce notebook est conçu pour classer les performances des étudiants en utilisant divers modèles d'apprentissage automatique. Le jeu de données utilisé provient du jeu de données xAPI-Edu-Data, qui comprend des informations sur la démographie des étudiants, leurs performances académiques et leur participation à des activités parascolaires. L'objectif est de prédire la catégorie de performance finale des étudiants.

Description du Jeu de Données

Le jeu de données comprend les caractéristiques suivantes :

- **Genre** : Le genre de l'étudiant (Masculin/Féminin).
- **Niveau d'éducation des parents** : Le niveau d'éducation atteint par les parents de l'étudiant.
- **Type d'école** : Type d'école fréquentée par l'étudiant (Publique/Privée).
- **Temps d'étude** : Nombre d'heures d'étude par semaine.
- **Notes** : Notes obtenues lors de diverses évaluations.
- **Activités parascolaires** : Participation à des activités parascolaires (Oui/Non).
- **Absences** : Nombre de jours d'école manqués.

La variable cible est la catégorie de performance, qui classe les étudiants en différents niveaux en fonction de leurs réalisations académiques.

Modèles Implémentés

Trois modèles de classification différents ont été implémentés et évalués :

1. Classifieur à Vecteurs de Support (SVC)

- **Description** : Ce modèle utilise des hyperplans pour séparer les points de données en différentes classes. Il est efficace dans les espaces de haute dimension et convient pour une classification non linéaire.
- **Implémentation** :

```
models['SVC'] = SVC()
```

2. Classifieur par Arbre de Décision

- **Description** : Un arbre de décision est une structure de type organigramme où les nœuds internes représentent des caractéristiques, les branches représentent des règles de décision et chaque feuille représente un résultat. Il est facile à interpréter et utile pour les données catégoriques.
- **Implémentation** :

```
models['DecisionTreeClassifier'] = DecisionTreeClassifier()
```

3. Régression Logistique

- **Description** : La régression logistique est un modèle linéaire utilisé pour les problèmes de classification binaire. Elle estime la probabilité d'une réponse binaire en fonction d'une ou plusieurs variables prédictives.
- **Paramètres** :
 - `max_iter = 480` pour assurer la convergence lors de l'entraînement.
- **Implémentation** :

```
models['LogisticRegression'] = LogisticRegression(max_iter=480)
```

Évaluation des Performances

Les modèles sont évalués à l'aide d'une matrice de confusion, qui fournit un aperçu de la précision et des erreurs de classification pour chaque classe.

Exemple de Matrice de Confusion pour le SVC :

```
confusion_matrix_svc = [[22, 1, 0],  
                        [4, 31, 10],  
                        [0, 12, 16]]
```

- **Classe 0** : 22 corrects, 1 mal classé en Classe 1.
- **Classe 1** : 31 corrects, 4 mal classés en Classe 0, 10 en Classe 2.
- **Classe 2** : 16 corrects, 12 mal classés en Classe 1.

Conclusion

Ce notebook démontre l'implémentation de plusieurs modèles de classification pour prédire les performances des étudiants. Les résultats indiquent des points forts et des axes d'amélioration potentiels, notamment en réduisant les erreurs de classification entre certaines classes. Un ajustement supplémentaire des paramètres des modèles et l'exploration de caractéristiques additionnelles pourraient améliorer la précision des prédictions.