Raffinement du Modèle et Soumission de Test

# Titre:

Analyse des habitudes des étudiants et performance académique

# Membres de l’équipe

1. Abate
2. Bassim
3. Daoud

# 1. Raffinement du Modèle

## 1.1 Aperçu

La phase de raffinement vise à améliorer les performances du modèle d'apprentissage automatique. Elle consiste à tester plusieurs approches, affiner les paramètres, et sélectionner les caractéristiques les plus pertinentes afin d’optimiser la précision et la robustesse du modèle.

## 1.2 Évaluation du Modèle

Lors de l’évaluation initiale, la régression linéaire a montré une bonne corrélation entre les heures d’étude et les scores d’examen. Cependant, les erreurs résiduelles indiquaient une variabilité inexpliquée, possiblement due à des interactions non modélisées ou à des caractéristiques inutiles.

## 1.3 Techniques de Raffinement

Nous avons exploré les techniques suivantes :  
- Ajustement des hyperparamètres du modèle  
- Comparaison avec Ridge et Lasso Regression  
- Sélection des caractéristiques les plus significatives

## 1.4 Réglage des Hyperparamètres

Des techniques comme la recherche par grille (GridSearchCV) ont permis de tester différentes valeurs du paramètre de régularisation (alpha). Cela a contribué à réduire le surapprentissage et à améliorer la généralisation.

## 1.5 Validation Croisée

Une validation croisée à 5 plis (k-fold) a été mise en œuvre au lieu d’une simple séparation train/test. Cela a permis une évaluation plus stable et moins dépendante du split initial.

## 1.6 Sélection des Caractéristiques

L’analyse des coefficients et la méthode RFE (Recursive Feature Elimination) ont été utilisées pour éliminer les variables peu significatives, ce qui a réduit la complexité du modèle tout en conservant ses performances.

# 2. Soumission de Test

## 2.1 Aperçu

Cette phase consiste à appliquer le modèle entraîné sur des données de test pour évaluer sa capacité à généraliser et simuler un déploiement réel.

## 2.2 Préparation des Données pour les Tests

Les données de test ont été soumises aux mêmes étapes de transformation que les données d'entraînement : encodage, normalisation et gestion des valeurs manquantes.

## 2.3 Application du Modèle

Le modèle final a été chargé et appliqué sur le jeu de données de test.  
Exemple :

y\_pred = model.predict(X\_test)

## 2.4 Métriques de Test

Les performances sur le test ont été mesurées avec R², RMSE et MAE. Ces résultats ont été comparés aux scores d'entraînement et de validation pour s'assurer de la cohérence du modèle.

## 2.5 Déploiement du Modèle

Le modèle peut être déployé à l'aide de bibliothèques comme Flask ou FastAPI pour une interface web, ou intégré dans un pipeline de données automatisé. Cette étape dépend du contexte du projet.

## 2.6 Implémentation du Code

# Exemple de pipeline de raffinement :  
from sklearn.linear\_model import RidgeCV  
model = RidgeCV(alphas=[0.1, 1.0, 10.0], cv=5)  
model.fit(X\_train, y\_train)  
y\_pred = model.predict(X\_test)

# 3. Conclusion

Le raffinement du modèle a permis d'améliorer significativement les performances. L'usage de la régularisation, de la validation croisée et de la sélection de caractéristiques a renforcé la robustesse du modèle. Les défis principaux comprenaient le choix des variables pertinentes et la gestion du biais/variance.

# 4. Références

- https://scikit-learn.org/  
- https://www.kaggle.com/datasets/jayaantanaath/student-habits-vs-academic-performance  
- Documentation pandas, matplotlib, numpy  
- Cours Machine Learning – Andrew Ng (Coursera)