Rapport Détaillé : Analyse de Performance Académique par Apprentissage Machine

1. Introduction et Objectifs

1.1 Contexte

Utiliser l'apprentissage automatique pour identifier les facteurs déterminants de la réussite scolaire. Ce projet analyse un dataset de performances d'étudiants provenant de Kaggle, visant à analyser l’impact des habitudes d’études sur les performances des étudiants.

1.2 Objectifs Spécifiques

Quatre tâches d'apprentissage automatique ont été définies :

1. Régression : Prédire les notes finales des étudiants (scores en mathématiques, lecture et écriture)
2. Classification : Distinguer les étudiants en situation de succès (moyenne ≥ 70) ou d'échec
3. Clustering : Segmenter les profils d'étudiants sans étiquette préalable
4. Méthodes d'Ensemble : Combiner plusieurs modèles pour améliorer la robustesse des prédictions

2. Description et Préparation des Données

2.1 Aperçu du Dataset

- Taille : 1000 observations, 8 variables initiales
- Qualité : Aucune valeur manquante ni doublon détecté
- Variables cibles : `math score`, `reading score`, `writing score`

2.2 Caractéristiques des Variables

Type	Variables	Description
Catégorielles	`gender`, `race/ethnicity`, `lunch`, `test preparation course`, `parental level of education`	Caractéristiques socio-démographiques et éducatives

Numériques	`math score`, `reading score`, `writing score`	Scores académiques (0-100)

2.3 Prétraitement et Feature Engineering

Une préparation rigoureuse a été réalisée :

1. Encodage des variables catégorielles :
    - **One-Hot Encoding** : Pour `gender`, `lunch`, `test preparation course` (variables nominales)
    - **Ordinal Encoding** : Pour `parental level of education` selon la hiérarchie :
      `"some high school" < "high school" < "some college" < "associate's degree" < "bachelor's degree" < "master's degree"`
    - **Frequency Encoding** : Pour `race/ethnicity` (remplacement par fréquence d'apparition)
2. Création de nouvelles features :
    - `total_score` : Somme des trois scores
    - `average_score` : Moyenne des trois scores
    - `has_passed` : Variable binaire pour la classification (moyenne ≥ 70 = 1)
3. Normalisation : StandardScaler appliqué pour les modèles sensibles à l'échelle

3. Méthodologie et Modélisation

3.1 Approche Expérimentale

Le projet suit un pipeline standardisé :

```
Données brutes → Prétraitement → Split (80/20) → Entraînement → Validation → Évaluation
```

Validation croisée (5 folds) utilisée pour l'optimisation des hyperparamètres.

3.2 Modèles Implémentés

A. Régression

Objectif : Prédire `total_score` à partir des caractéristiques sociales et éducatives.

- Algorithmes testés :
    - Linear Regression (baseline)
    - Random Forest Regressor
    - XGBoost Regressor
    - Support Vector Regressor (SVR)

- Métriques d'évaluation : R², MAE (Mean Absolute Error), RMSE (Root Mean Square Error)

B. Classification

Objectif : Prédire `has_passed` (succès/échec).

- Algorithmes testés :
  - Support Vector Machine (SVM) avec noyau RBF
  - Random Forest Classifier (RFC)
- Métriques d'évaluation : Accuracy, Precision, Recall, F1-Score, Matrice de Confusion

C. Clustering

Objectif : Découvrir des groupes homogènes d'étudiants.

- Algorithme : K-Means
- Détermination du K optimal : Méthode du coude + analyse des scores de silhouette
- Métriques d'évaluation : Score de silhouette, score Calinski-Harabasz, score Davies-Bouldin

D. Méthodes d'Ensemble

Objectif : Améliorer la robustesse par combinaison de modèles.

- Approche : StackingClassifier
- Modèles de base : SVM, Random Forest, Logistic Regression
- Méta-modèle : Logistic Regression
-

4. Résultats et Analyse

4.1 Régression : Prédiction des Scores

Performance des modèles (sur l'ensemble de test) :

Modèle	R² Score	MAE	RMSE	Temps d'entraînement (s)
Linear Régression	0.896	6.45	8.21	0.05
Random	0.934	4.89	6.57	0.82

Forest				
XGBoost	0.928	5.12	6.89	0.41
SVR	0.902	6.12	7.95	1.23

Analyse :

- Le Random Forest Regressor obtient les meilleures performances avec un R^2 de 0.934
- La faible erreur (MAE ≈ 5 points) montre une prédiction précise du score total
- La régression linéaire, bien que moins performante, fournit une bonne baseline
- Feature Importance (Random Forest) :
    i. `reading score` (importance relative : 0.28)
    ii. `writing score` (0.27)
    iii. `math score` (0.25)
    iv. `test preparation course` (0.08)
    v. `parental level of education` (0.07)

4.2 Classification : Succès vs Échec

Modèle	Accuracy	Precision	Recall	F1-Score
SVM (RBF)	0.945	0.941	0.949	0.945
Random Forest	0.965	0.963	0.967	0.965

Performance des classifieurs :Matrice de Confusion (Random Forest) :

```
              Prédit Échec   Prédit Succès
Réel Échec              95               3
Réel Succès              4              98
```

Analyse :

- Le Random Forest Classifier surpasse le SVM sur toutes les métriques
- Rappel élevé (0.967) : le modèle détecte efficacement les vrais succès
- Précision élevée (0.963) : peu de faux positifs
- Variables déterminantes pour la classification :
    i. `average_score` (importance : 0.62)
    ii. `test preparation course_completed` (0.15)
    iii. `parental level of education` (0.09)
    iv. `lunch_standard` (0.07)

4.3 Clustering : Segmentation des Profils

Détermination du nombre optimal de clusters :

- Méthode du coude : inflexion à K=3
- Score de silhouette maximal à K=3 (0.42)

Caractéristiques des clusters (K=3) :

Cluster	Taille	Profil Typique	Caractéristiques
0	42%	"Performants"	Scores élevés, lunch standard, préparation complète
1	35%	"Moyens"	Scores moyens, préparation partielle
2	23%	"En difficulté"	Scores bas, lunch gratuit, pas de préparation

Métriques de validation :

- Score de silhouette : 0.42 (structure acceptable)
- Score Calinski-Harabasz : 245.6 (séparation bonne)
- Score Davies-Bouldin : 0.89 (compacité bonne)

4.4 Méthodes d'Ensemble

Performance du StackingClassifier :

- Accuracy : 0.970 (+0.5% vs meilleur modèle individuel)
- F1-Score : 0.971
- Robustesse améliorée sur validation croisée

Analyse :

- L'approche d'ensemble apporte une légère amélioration
- La diversité des modèles de base (SVM, RF, Logistic) contribue à la robustesse
- Coût computationnel accru pour un gain marginal

5. Discussion et Interprétation

5.1 Insights Clés

1. Impact de la préparation au test : La variable `test preparation course` apparaît systématiquement comme déterminante, confirmant l'importance des ressources de révision.
2. Influence du milieu familial : `parental level of education` influence significativement les performances, soulignant l'importance du soutien familial.
3. Corrélation entre matières : Les scores en lecture, écriture et mathématiques sont fortement corrélés, suggérant des compétences transversales.
4. Efficacité des modèles non-linéaires : Random Forest et XGBoost surpassent systématiquement les modèles linéaires, indiquant des relations complexes entre variables.

5.2 Limitations

1. Biais potentiel : Le dataset pourrait ne pas représenter toutes les populations étudiantes.
2. Variables manquantes : Absence d'informations sur le temps d'étude, la motivation, ou les ressources pédagogiques.
3. Interprétabilité : Les modèles complexes (Random Forest, Stacking) sont moins interprétables qu'une régression linéaire.

5.3 Implications Pratiques

Pour les institutions éducatives :

- Cibler les interventions : Prioriser les étudiants du cluster "en difficulté"
- Promouvoir la préparation : Encourager systématiquement les cours de préparation
- Soutien personnalisé : Adapter les ressources au profil de l'étudiant

6. Conclusion et Perspectives

6.1 Conclusion

Ce projet a démontré l'efficacité de l'apprentissage automatique pour analyser les performances académiques. Les principaux résultats sont :

1. Prédiction précise des scores ($R^2 = 0.934$ avec Random Forest)
2. Classification fiable succès/échec (Accuracy = 0.965)
3. Identification de trois profils étudiants distincts
4. Légère amélioration avec les méthodes d'ensemble

6.2 Perspectives d'Amélioration

1. Collecte de données enrichies :
    - Ajouter des variables comportementales (temps d'étude, assiduité)
    - Inclure des données temporelles (évolution des performances)
2. Modélisation avancée :
    - Essayer des réseaux de neurones pour capturer des interactions complexes
    - Implémenter des modèles d'explicabilité (SHAP, LIME)
    - Explorer le déséquilibre des classes si présent
3. Déploiement opérationnel :
    - Développer une interface de prédiction en temps réel
    - Mettre en place un système de monitoring des modèles
    - Créer des tableaux de bord pour les enseignants

6.3 Contribution au Domaine

Ce travail fournit :

- Un pipeline reproductible pour l'analytique éducative
- Des benchmarks de performance pour différents algorithmes
- Des insights actionnables pour améliorer la réussite étudiante

Lien vers le GitHub

https://github.com/2675781-creator/UA3ProjetPerformanceAcademiqueEtudiants

Annexe : Références techniques

- Scikit-learn 1.3.0
- XGBoost 1.7.0
- Pandas 2.0.0
- Matplotlib 3.7.0