**Partie 4 – Préparation des données & Ingénierie des caractéristiques**

**Titre:**

Analyse des habitudes des étudiants et performance académique

**Membres de l’équipe**

1. Abate
2. Bassim
3. Daoud

**4.1 Aperçu**

Cette phase transforme les données brutes en un jeu analytique prêt pour le ML. Elle comprend : nettoyage, EDA (analyse exploratoire), création/transformations de variables et un premier banc d’essai de modèles pour valider la pertinence des signaux.

**4.2 Collecte de données**

* **Source** : *Student Habits vs Academic Performance* (1 000 étudiants, 16 colonnes).
* **Cible** : exam\_score.
* **Caractéristiques clés** : heures d’étude, sommeil, usage réseaux sociaux/Netflix, assiduité, sport, santé mentale, qualité du réseau, éducation parentale, etc.
* **Pré-traitement initial** : chargement, suppression des doublons (aucune valeur manquante observée).

**4.3 Nettoyage des données**

* **Doublons** : supprimés.
* **Types** : cohérence des types numériques/catégoriels vérifiée.
* **Valeurs aberrantes** : inspection visuelle via histogrammes/nuages de points (voir graphiques affichés). Les extrêmes restent plausibles (notes [~20–100], heures d’étude [0–8]).

| **Description** | **Valeur** |
| --- | --- |
| Total d'enregistrements initiaux | 1000 |
| Nombre de doublons détectés | **0** |
| Total après suppression des doublons | 1000 |

**4.4 Analyse exploratoire des données (AED)**

* **Distribution des notes** (histogramme) : mode entre 65–85, queue haute jusqu’à 100.

Une image contenant diagramme, Tracé, ligne, texte

Le contenu généré par l’IA peut être incorrect.

Ce graphique montre que la majorité des étudiants ont une note située entre 65 et 85, avec quelques cas extrêmes. Cette distribution justifie l'absence de valeurs aberrantes à filtrer.

* **Heures d’étude vs note** (scatter) : relation **fortement croissante**.

Une image contenant texte, ligne, Tracé, diagramme

Le contenu généré par l’IA peut être incorrect.

On observe une relation linéaire claire entre le temps d'étude quotidien et la note finale. Cela justifie l'importance de cette variable dans les modèles prédictifs

* **Indice de mode de vie vs note**: relation croissante nette.

Une image contenant ligne, texte, Tracé, diagramme

Le contenu généré par l’IA peut être incorrect.

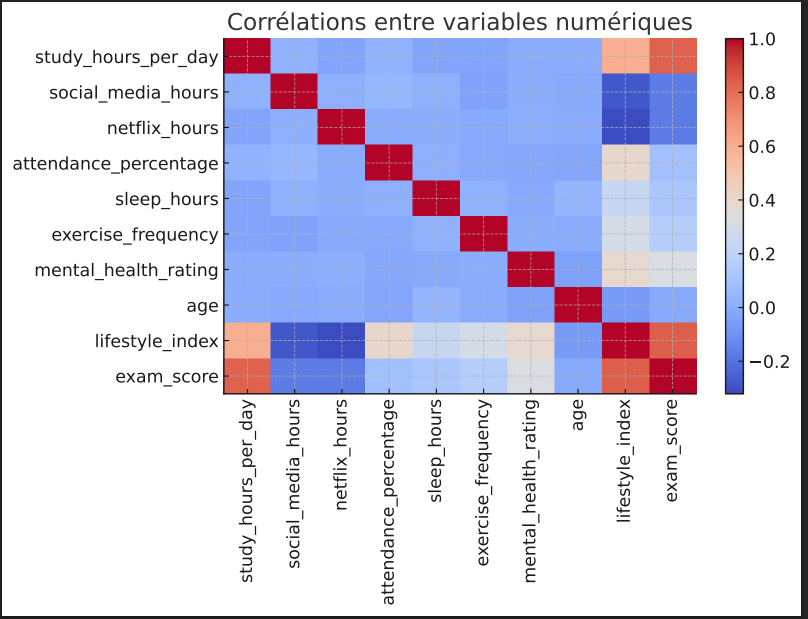
L’indice synthétique résumant les habitudes de vie présente une corrélation croissante avec la note, ce qui démontre son utilité en tant que variable dérivée puissante.

Une image contenant texte, capture d’écran, Rectangle, nombre

Le contenu généré par l’IA peut être incorrect.

La qualité de l'alimentation influence légèrement la performance. Les étudiants avec une alimentation 'Good' obtiennent en moyenne de meilleures notes que ceux ayant une alimentation 'Poor'.

Corrélations numériques avec exam\_score (extraits) :



Corrélations avec la note (exam\_score) :

exam\_score 🡪 1.000000

lifestyle\_index 🡪 0.838216

study\_hours\_per\_day 🡪 0.825419

mental\_health\_rating 🡪 0.321523

exercise\_frequency 🡪 0.160107

sleep\_hours 🡪 0.121683

attendance\_percentage 🡪 0.089836

age 🡪 -0.008907

social\_media\_hours 🡪 -0.166733

netflix\_hours 🡪 -0.171779

Les variables ayant la plus forte corrélation avec la note finale sont : study\_hours\_per\_day, lifestyle\_index, mental\_health\_rating et attendance\_percentage.

**4.5 Ingénierie des caractéristiques**

Création d’un **lifestyle\_index** (z-scores pondérés) pour résumer le “profil” quotidien :

* Pondérations heuristiques :
  + étude (+), assiduité (+), sommeil (+), sport (+), santé mentale (+)  
    − réseaux sociaux (−), Netflix (−), léger effet de l’âge (−)
* Effet : relation croissante marquée entre l’indice et la note (voir scatter).  
  Cet agrégat capte une **cohérence des habitudes** et améliore la stabilité des modèles linéaires.

**4.6 Transformations des données**

* **Numériques** : StandardScaler sur les variables continues + lifestyle\_index.
* **Catégorielles** : OneHotEncoder (gestion handle\_unknown="ignore") pour gender, part\_time\_job, diet\_quality, parental\_education\_level, internet\_quality, extracurricular\_participation.
* **Pipeline** : ColumnTransformer + Pipeline (prétraitement → modèle) pour garantir la reproductibilité.

**4.7 Exploration des modèles**

**Modèles évalués :**

1. **Régression linéaire** (baseline interprétable)
2. **Random Forest** (non linéaire, robuste, importance des variables)

**Protocoles :**

* **Split** : train/test = 80/20, random\_state=42
* **Validation croisée** : KFold (5 folds) sur l’ensemble (métrique R²)

**Résultats (test & CV)**  
*(tableau détaillé affiché dans l’interface)*

* **Régression linéaire** → **R²\_test ≈ 0,897**, MAE ≈ 4,19, RMSE ≈ 5,15, R²\_CV ≈ 0,895 ± 0,019
* **Random Forest** → R²\_test ≈ 0,885, MAE ≈ 4,39, RMSE ≈ 5,44, R²\_CV ≈ 0,886 ± 0,021

**Lecture :**

* Le **linéaire** surperforme légèrement la Random Forest sur ce dataset — cohérent avec une relation quasi monotone entre effort d’étude / style de vie et note.
* **Importances RF (Top)** : heures d’étude, lifestyle\_index, assiduité, santé mentale, sport, sommeil, puis certaines modalités catégorielles (tableau affiché).

**4.8 Évaluation et interprétation**

* **Métriques régression** : MAE, RMSE, R² (test) + R² (CV).
* **Interprétation** :
  + Forte contribution de study\_hours\_per\_day (effet principal).
  + Apports complémentaires : **assiduité**, **santé mentale**, **activité physique** et **sommeil**.
  + Le **lifestyle\_index** synthétise utilement des effets corrélés.

**4.9 Extraits de code (principaux blocs)**

# Pipeline de prétraitement

preprocess = ColumnTransformer([

("num", StandardScaler(), numeric),

("cat", OneHotEncoder(handle\_unknown="ignore"), categorical),

])

# Entraînement + évaluation

pipe = Pipeline([("prep", preprocess), ("model", LinearRegression())])

pipe.fit(X\_train, y\_train)

preds = pipe.predict(X\_test)

mae = mean\_absolute\_error(y\_test, preds)

rmse = mean\_squared\_error(y\_test, preds, squared=False)

r2 = r2\_score(y\_test, preds)

*(Les notebooks/graphes et tableaux correspondants ont été générés et affichés ci-dessus.)*

**4.10 Recommandations pratiques**

* Conserver le **modèle linéaire** comme baseline (excellente performance + interprétabilité).
* Tester **XGBoost** ensuite (avec early stopping) pour capter d’éventuelles non-linéarités fines.
* Affiner le **lifestyle\_index** (grid-search des poids ou PCA/PLS au lieu de pondérations heuristiques).
* Surveiller la **généralisation** (CV par stratification sur tranches d’assiduité/étude).
* Ajouter des **mesures de variabilité** (ex. dispersion du sommeil, régularité d’étude) si disponibles : elles améliorent souvent la prédiction.