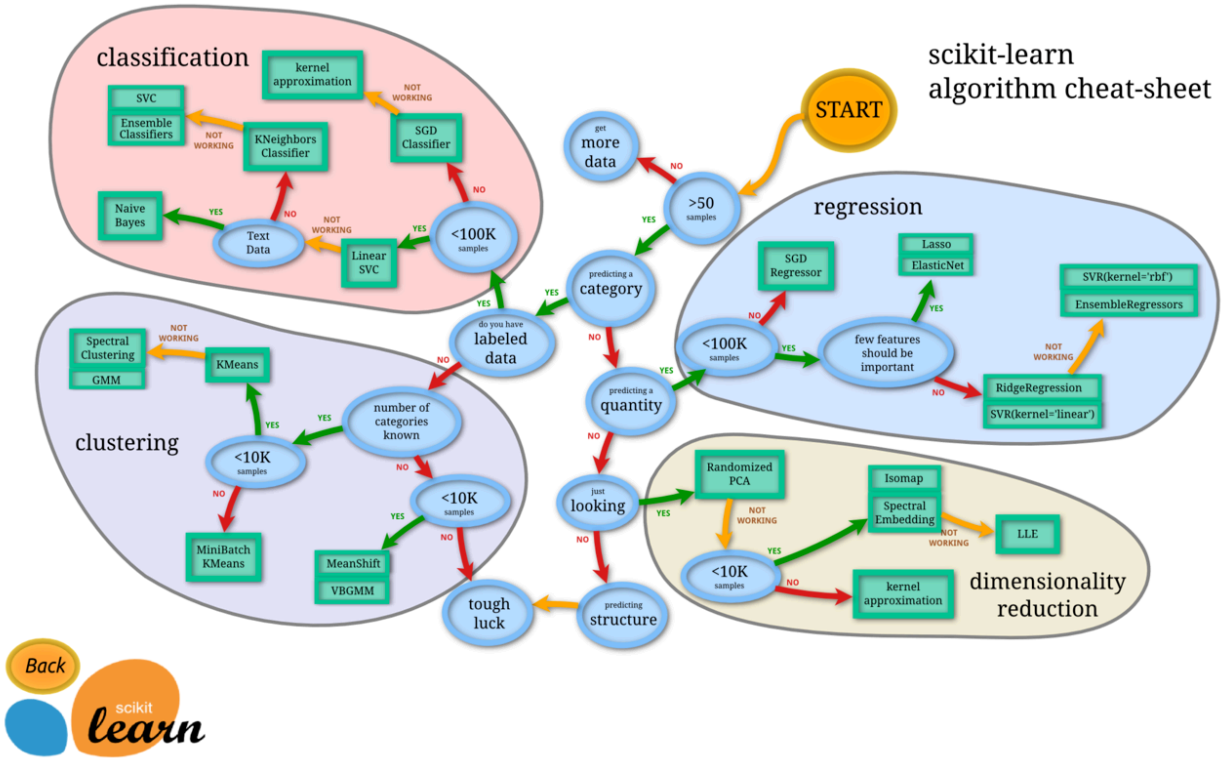
Rapport de Projet

Projet Data Mining

Analyse et Prédiction dans le Domaine Sport-Santé

Calories brûlées – Type d'entraînement – État de santé

Parcours : BUT 3 – Data Science



Introduction

L'analyse des données occupe aujourd'hui une place essentielle dans les domaines du sport, de la santé et du bien-être. Les outils modernes de mesure (montres connectées, applications de fitness, …) produisent en continu de grandes quantités d'informations sur l'activité physique, la nutrition ou encore l'état de forme des individus. L'objectif principal de ce projet est d'utiliser ces données pour mieux comprendre les habitudes sportives et identifier les facteurs qui influencent l'état de santé et la dépense énergétique. Dans ce travail, nous cherchons à répondre à trois questions importantes :

- Une personne est-elle en bonne santé ?
  Problème de classification binaire : prédire la variable **is_healthy**.
- Quel type d'entraînement correspond à son profil ?
  Problème de classification à 4 catégories : **Workout_Type**.
- Combien de calories va-t-elle brûler pendant sa séance ?
  Problème de régression : prédire **Calories_Burned**.

Ces trois problématiques couvrent les trois grandes familles classiques du Machine Learning supervisé : classification binaire, classification multi-classe, et régression.

L'objectif n'est pas seulement de créer des modèles, mais surtout de comprendre quelles variables influencent réellement la santé ou la performance, et d'interpréter les résultats de façon claire et accessible.

Dans toutes nos tâches, les variables cibles sont déjà présentes dans le dataset. Notre objectif n'est pas de créer des groupes, mais bien de prédire ces valeurs à partir des autres informations. L'apprentissage supervisé est donc la méthode la plus cohérente et la plus adaptée pour répondre aux objectifs du projet.

I. Préparation des données

Le dataset ***New_data.csv*** utilisé dans cette étude regroupe un ensemble d'informations liées à la santé, à la pratique sportive et aux habitudes de vie des individus. Chaque ligne correspond à une personne, et chaque colonne représente une caractéristique mesurée ou calculée.

1. Prétraitement des données

Avant d'entraîner les modèles, plusieurs traitements ont été réalisés pour garantir la qualité des prédictions.

a. Standardisation de certaines variables

Certaines méthodes, comme KNN et SVM, utilisent des distances. Il est donc nécessaire d'harmoniser l'échelle des variables.

Un **StandardScaler** a été appliqué sur toutes les variables explicatives dans les pipelines de modélisation. Cela permet d'éviter que des variables à grande échelle influencent trop les modèles.

```
preprocessor = ColumnTransformer(
    transformers=[
        ("num", StandardScaler(), features_calories_burned)
    ]
)
```

b. Encodage des variables catégorielles

L'encodage est indispensable pour permettre aux algorithmes de traitement numérique de fonctionner correctement.

```
mapping_workout = {"Cardio": 1, "HIIT": 2, "Strength": 3, "Yoga": 4}
df["Workout_Type_encode"] = df["Workout_Type"].map(mapping_workout)

df["Experience_Level"] = df["Experience_Level"].round().astype(int)
df['is_healthy_encode'] = df['is_healthy'].round().astype(int)
```

c. Sélection des variables pertinentes

Toutes les variables ne peuvent pas être utilisées dans les modèles.
Un tri méthodique a été effectué selon :

- la corrélation avec la cible,
- l'évitement de la fuite d'information,

- la cohérence physiologique (d'ailleurs on n'y avait pas pensé au début).

2. Séparation des ensembles d'entraînement et de test

Pour les trois prédictions, un découpage des données a été effectué :

- un ensemble d'entraînement pour apprendre les modèles, soit 80% du dataset,
- un ensemble de tests pour évaluer leurs performances réelles, soit 20% du dataset.

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=42, stratify=y
)
```

Le paramètre **stratify=y** préserve la proportion des classes dans les deux ensembles.

3. Métriques

Afin de vérifier l'adéquation de nos modèles, nous utilisons plusieurs critères usuels dans le cadre de la régression :

Formules des métriques utilisées ¶

1. R^2 — Coefficient de détermination

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

2. RMSE — Root Mean Squared Error

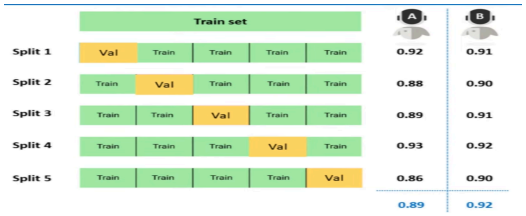
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

3. MAE — Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Le RMSE et le MAE caractérisent l'écart moyen à la valeur à prédire. Cependant, le RMSE est plus sensible à la présence de forts écarts à cause de la présence du carré. Ces deux critères sont à minimiser et le R2 est à avoir aussi proche de 1 que possible afin d'avoir un bon modèle .

4. Validation croisée (Cross Validation)



Principe :

On découpe le dataset en K parties (souvent K=5). On entraîne K modèles, chacun utilisant :

- K−1 parties pour apprendre
- 1 partie pour tester

On fait la moyenne des performances.

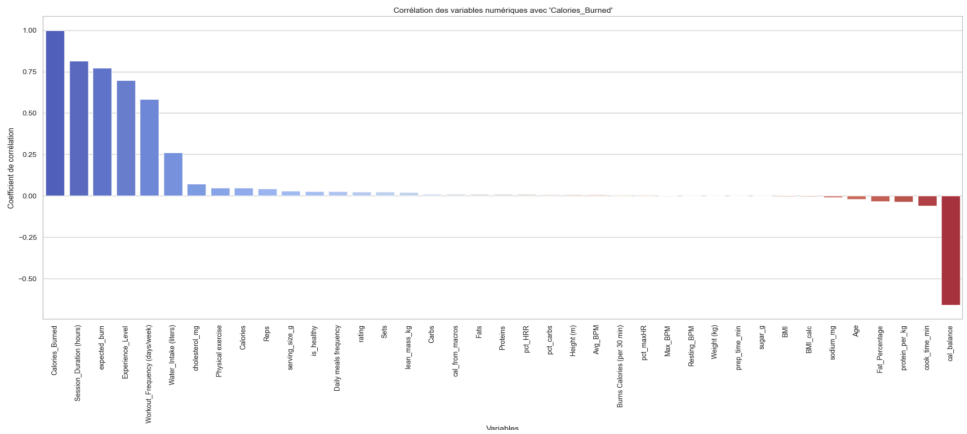
II. Prédiction des calories brûlées (Calories_Burned)

1. Objectif

L'un des objectifs de ce projet est d'estimer le nombre de calories brûlées pendant une séance d'entraînement, en fonction des caractéristiques physiques de la personne, de son niveau sportif et des paramètres de la séance.

La variable cible **Calories_Burned** est numérique continue. Il s'agit donc d'un problème de régression.

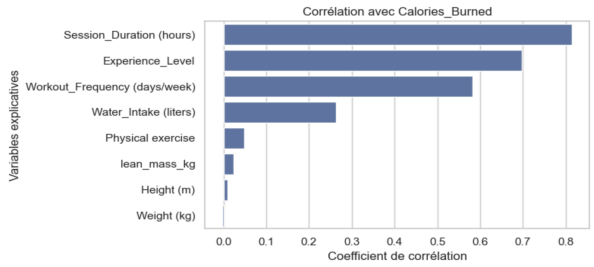
2. Sélection des variables explicatives



Lors de la sélection des variables explicatives, certaines variables doivent impérativement être retirées car elles sont directement liées ou proportionnelles à **Calories_Burned**. Les utiliser fausserait totalement le modèle, car celui-ci récupérerait indirectement la réponse qu'il doit prédire.

Variable	Motif d'exclusion
expected_burn	Corrélation très élevée (plus de 0.77). Variable presque dupliquée, ce qui revient à donner la réponse au modèle.
Workout_Type	Utilisé pour générer ou influencer `expected_burn`. Dépendance directe.
cal_balance	Calculé directement avec `Calories_Burned`, fuite totale.
Burns Calories (per 30 min)	Formule physiologique proportionnelle à la dépense calorique.
Burns Calories (per 30 min)_bc	Duplication de la variable précédente.
Calories (ingérées)	Peut être confondu avec Calories_Burned, mélange apport/dépense calorique.
Reps / Sets	Variables trop indirectes, introduisent du bruit et une forte imprécision.
BMI, BMI_calc, protein_per_kg	Déjà dérivées de Height et Weight. Leur utilisation double l'information et complexifie inutilement le modèle.

Voici les variables retenues pour prédire **Calories_Burned**



- Session_Duration (hours) : plus une séance dure longtemps, plus la dépense calorique est élevée. C'est le facteur principal de la dépense énergétique.
- Experience_Level : les personnes expérimentées s'entraînent plus efficacement et à une intensité plus élevée, ce qui augmente les calories brûlées.
- Workout_Frequency (days/week) : un entraînement régulier améliore la condition physique et permet d'effectuer des séances plus intenses.
- Water_Intake (liters) : une meilleure hydratation reflète souvent une pratique sportive plus sérieuse favorisant des efforts plus soutenus.
- Physical exercise : indique le niveau d'activité générale d'une personne. Une personne active brûle en moyenne plus de calories lors de ses séances.
- lean_mass_kg : la masse musculaire augmente fortement la dépense calorique, car les muscles consomment beaucoup d'énergie.
- Height (m) : les personnes grandes ont un gabarit plus important, ce qui nécessite plus d'énergie pour le mouvement.

- Weight (kg) : un poids plus élevé demande plus d'effort mécanique, ce qui augmente la dépense calorique lors d'une activité.

Dans l'ensemble, toutes les variables retenues présentent une relation logique avec la dépense calorique et constituent un ensemble pertinent pour modéliser et prédire **Calories_Burned**.

3. Modèles de régression utilisés

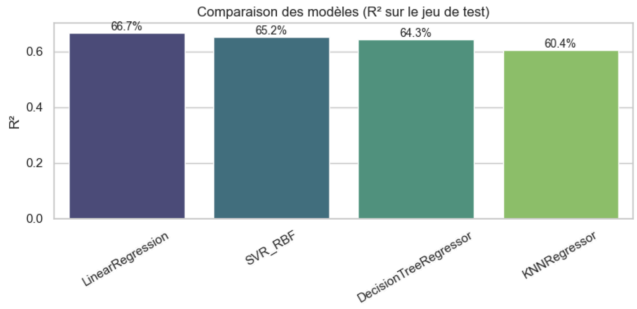
La variable que nous cherchons à prédire, **Calories_Burned**, est une quantité numérique continue. Ce n'est ni une catégorie, ni une classe, mais une valeur qui peut prendre n'importe quel nombre (ex : 450, 1200, 2350…).

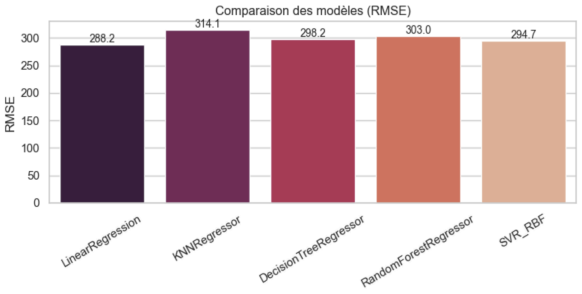
Dans ce cas, les modèles adaptés sont des modèles de régression, car :

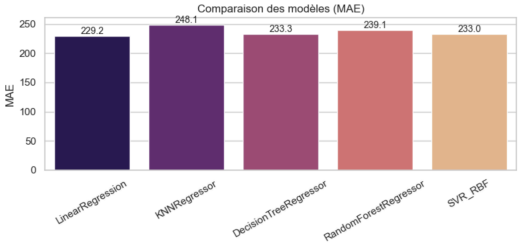
1. Ils permettent de prédire une valeur numérique à partir d'un ensemble de variables explicatives.
2. Ils mesurent la relation entre l'effort fourni et les calories brûlées, ce qui correspond exactement à notre objectif.
3. Ils donnent des indicateurs précis (R^2, RMSE, MAE) pour évaluer la qualité des prédictions.
4. Ils s'adaptent bien à des phénomènes physiologiques, où la sortie est continue, comme la dépense énergétique.
5. Ils permettent de comparer plusieurs approches (linéaire, distances, arbres…) pour trouver le modèle le plus précis.

Ainsi, la régression est la méthode la plus logique et la plus pertinente pour prédire combien de calories une personne brûle en fonction de la durée de sa séance, de son niveau d'expérience, de son poids, de sa masse musculaire et d'autres facteurs liés à l'activité physique.

Cinq modèles ont été testés : Régression linéaire, KNN Regressor, Decision Tree Regressor, SVR (RBF). Après la validation croisée (KFold), chaque modèle a été évalué avec : R^2 , RMSE (erreur quadratique moyenne), MAE (erreur absolue moyenne).







Interprétation globale des modèles

Les trois graphiques montrent clairement que la régression linéaire est le modèle le plus performant pour prédire **Calories_Burned**.

- R^2 : c'est elle qui explique la plus grande part de la variation des calories brûlées (≈ 66,7 %), devant SVR et l'arbre de décision.
- RMSE : elle a les erreurs moyennes au carré les plus faibles, donc ses prédictions s'éloignent moins des valeurs réelles.
- MAE : elle montre aussi les écarts moyens les plus petits, ce qui confirme sa précision.

Cela signifie qu'une relation globalement linéaire existe entre les variables explicatives retenues et les calories brûlées.

En résumé : *la régression linéaire est le modèle le plus fiable et le plus précis pour ce problème.*

4. Tests manuels

Une fonction de prédiction manuelle a été construite pour estimer les calories brûlées pour un profil donné :

```
predict_calories_with_explanation(
    model = linreg_pipe,
    session_duration = 1.0,
    experience_level = 3,
    workout_freq = 4,
    water_intake = 2.0,
    physical_exercise = 3,
    lean_mass = 60,
    height = 1.80,
    weight = 75
)
```

```
=== Prédiction manuelle ===
Calories brûlées estimées : 1182.5 kcal
Interprétation : séance très intense, forte dépense énergétique.
```

III. Prédiction de l'état de santé (is_healthy)

1. Objectif

L'objectif de cette première partie est de prédire si un individu est en bonne santé ou non, en utilisant les informations liées à son activité physique, sa morphologie et ses habitudes de vie. La variable cible **is_healthy** contient deux classes : 0 pour non healthy et 1 pour healthy. Il s'agit donc d'un problème de classification binaire.

2. Variables explicatives retenues

On a choisi les variables explicatives de la façon la plus pertinente pour la prédiction de **is_Healthy**, selon les mêmes critères utilisés pour la prediction de **Calories_Burned**.

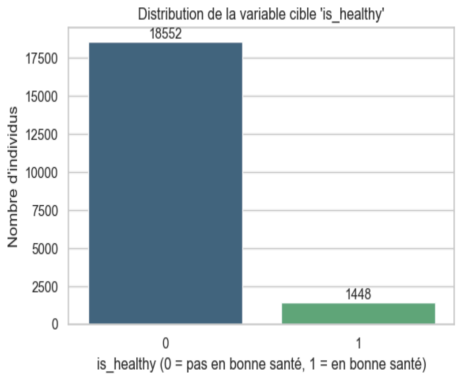
Variables explicatives pour "is_Healthy" ¶

```
features = ['sugar_g','rating','cholesterol_mg','Water_Intake (liters)','Weight (kg)',
            'lean_mass_kg','Calories','Reps','expected_burn','Session_Duration (hours)']
```

Variables explicatives pour "Workout_Type"

```
features = ["Session_Duration (hours)","Calories_Burned",
    "Experience_Level","Weight (kg)"]
```

3. Distribution de la cible



La cible **is_healthy** est extrêmement déséquilibrée :

- 18 552 personnes ne sont pas healthy (classe 0)
- 1 448 seulement sont healthy (classe 1)

Ainsi, plus de 92 % des individus appartiennent à la classe 0.

4. Modèles utilisés

Plusieurs modèles de classification ont été testés : Régression Logistique, KNN Classifier, Arbre de décision, SVM (noyau RBF). Ces modèles couvrent différentes familles d'algorithmes : linéaire, à distance, à base d'arbres, et à noyaux.

5. Comparaison des modèles

	Model	CV_accuracy_mean	CV_accuracy_std	Test_accuracy
0	KNN	0.982875	0.001805	0.98175
1	RandomForest	0.979625	0.001072	0.97800
2	DecisionTree	0.970063	0.002961	0.96800
3	SVM_RBF	0.932625	0.001212	0.93875
4	LogReg	0.927625	0.000250	0.92875

Interpretations : Le déséquilibre de la variables **is_healthy** explique en grande partie les accuracy très élevées des modèles. En effet, un modèle très simple pourrait atteindre 92 % d'accuracy simplement en prédisant *toujours* la classe 0, sans jamais apprendre quoi que ce soit sur les personnes healthy. Même si les modèles testés (KNN, RandomForest, DecisionTree, SVM, LogReg) semblent obtenir des scores entre 92 % et 98 %, ces valeurs ne reflètent pas réellement leur capacité à reconnaître les personnes "healthy".

Un modèle peut :

- être très bon sur la classe majoritaire (0),
- être mauvais sur la classe minoritaire (1),
- mais afficher quand même une accuracy élevée.

C'est ce que révèle la distribution de la cible : la majorité écrase les performances globales.

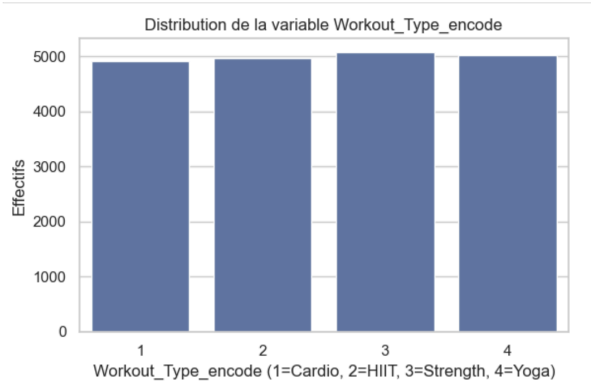
IV. Prédiction du type d'entraînement (Workout_Type)

1. Objectif

L'un des objectifs de cette étude est de prédire le type d'entraînement qu'un individu est susceptible d'effectuer, en fonction de son profil, de ses caractéristiques physiques et de ses habitudes sportives. La variable cible **Workout_Type** possède 4 catégories : **Cardio, HIIT, Strength (musculation)** et **Yoga.** Il s'agit donc d'un problème de classification multi-classes, plus complexe que la classification binaire réalisée dans la section précédente.

2. Encodage et distribution de la cible

Workout_Type	Code
Cardio	1
HIIT	2
Strength	3
Yoga	4



La distribution de **Workout_Type** est parfaitement équilibrée entre les quatre catégories.

Chaque type d'entraînement compte environ 5000 individus, ce qui permet d'entraîner un modèle de classification multi-classes sans risque de biais dû au déséquilibre.

Les performances obtenues par les modèles sont donc fiables et directement interprétables.

3. Variables explicatives retenues

```
features = ["Session_Duration (hours)","Calories_Burned",
    "Experience_Level","Weight (kg)"]
```

Pour ce modèle, seules quatre variables ont été utilisées. Elles ont été choisies car elles influencent réellement la manière dont une personne construit ou réalise sa séance d'entraînement. Les variables retenues sont :

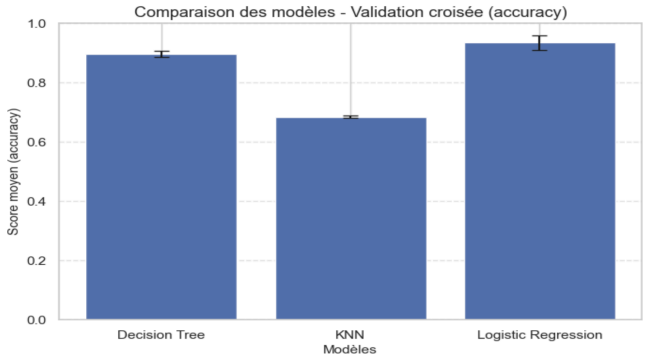
- Session_Duration (hours) : la durée de la séance permet de différencier les entraînements. Les séances longues sont plus proches du Cardio ou du Yoga, tandis que les séances courtes mais explosives sont typiques du HIIT.
- Calories_Burned : la dépense énergétique reflète l'intensité. Elle aide à distinguer un entraînement très coûteux (ex. HIIT ou musculation) d'un entraînement plus doux (ex. Yoga).
- Experience_Level : plus le niveau d'expérience est élevé, plus l'individu est susceptible de pratiquer des entraînements exigeants ou techniques (HIIT, Strength).
- Weight (kg) : le poids peut influer sur la façon dont une personne réalise certains exercices, notamment pour les séances de musculation ou à haute intensité.

Ces quatre variables permettent de construire des profils réalistes :

- **Cardio** : dépense modérée et durée souvent longue
- **HIIT** : durée courte, dépenses très élevées, nécessite une bonne expérience
- **Strength** : forte dépense énergétique, praticiens souvent plus expérimentés ou avec un poids plus important
- **Yoga** : intensité faible, calories brûlées plus basses

4. Modèles utilisés

Les modèles suivants ont été testés : Régression Logistique, KNN Classifier, Decision Tree.



La régression logistique est le meilleur modèle, avec une accuracy d'environ 86,8%, et une petite barre d'erreur.

5. **Tests manuels**

```
import numpy as np

def predict_workout(model, session_duration, calories_burned, experience_level, weight):
    x = np.array([session_duration, calories_burned, experience_level, weight]).reshape(1, -1)
    prediction = model.predict(x)[0]
    mapping = {1: "Cardio",2: " HIIT",3: "Strength",4: "Yoga"}

    print(f"Type d'entraînement prédit : {mapping.get(prediction, 'Inconnu')} (classe {prediction})")
```

```
predict_workout(model4, 2, 2000, 2, 57)
```

```
Type d'entraînement prédit : Strength (classe 3)
```

Problèmes rencontrés au cours du projet :

Lors de ce projet nous nous sommes rendus compte au cours de nos premiers tests de modèles que quelque chose n'allait pas car nous trouvons systématiquement des modèles extrêmement fiables de 99% à 100% de précisions. Cela nous a mis la puce à l'oreille et nous avons cherché à mieux comprendre.

Nous nous sommes rendus compte que certaines données de test figuraient déjà dans les données d'entrainements et donc cela expliquait tous ces modèles différents extrêmement performants. De plus, pour la variable **Calories_burned** on s'est rendu compte que certaines variables parmi les plus importantes dans la prédiction étaient calculées à partir **Calories_burned**, comme c'était le cas de la variable **Calorie_Burn_Efficiency**.

De plus, nous nous sommes aperçus que la variable de poids n'avait pas d'impact significatif dans la prédiction du modèle des calories brûlées d'après la matrice de correlation, or on sait que dans la réalité il s'agit d'un facteur important et avéré. Cependant notre jeu de données nous laisse entendre le contraire.

Nous avons corrigé le problème en appliquant une séparation stricte des données d'entraînement et de test, en supprimant les variables responsables de fuite d'information et en analysant les corrélations pour identifier puis exclure les incohérences du dataset.

Définitions des modèles utilisés

- Régression linéaire: Modèle qui cherche une relation linéaire entre les variables explicatives et une valeur numérique à prédire. Utilisé pour la régression.
- Régression logistique: Modèle linéaire utilisé pour la classification, malgré son nom. Il prédit la probabilité qu'un individu appartienne à une classe.

- KNN (K-Nearest Neighbors) : Modèle qui prédit en regardant les K individus les plus proches du point à classer ou à estimer. Utilisé pour la classification et la régression.
- Decision Tree (Arbre de décision) : Modèle qui prend des décisions en suivant des règles simples sous forme de questions (“si durée > X alors…”). Utilisé pour séparer des classes ou prédire une valeur.
- Random Forest : Ensemble de plusieurs arbres de décision. Chaque arbre vote, ce qui rend la prédiction plus stable et moins sensible au bruit.
- SVM (Support Vector Machine) – Noyau RBF : Modèle qui cherche une frontière optimale pour séparer les classes. Le noyau RBF permet de capturer des relations non linéaires.
- KNN Regressor : Version régressive du KNN : la prédiction est la moyenne des valeurs des K voisins les plus proches
- Decision Tree Regressor : Version régressive de l’arbre de décision : il découpe les données en zones et attribue à chaque zone une prédiction moyenne.
- SVR (Support Vector Regression) : Version régressive du SVM : cherche une fonction qui reste au plus près des données tout en étant la plus plate possible.

Interprétation des hyperparamètres

- max_iter (régression logistique) : Ce paramètre fixe le nombre maximal d’itérations de l’algorithme d’optimisation. Une valeur trop faible peut empêcher le modèle de converger.
- n_neighbors (KNN) : C’est le nombre de voisins utilisés pour prédire la classe. Un k trop petit rend le modèle sensible au bruit, tandis qu’un k trop grand lisse trop la décision.
- max_depth (arbre de décision) : Une profondeur trop élevée entraîne du sur-apprentissage. Une profondeur trop faible entraîne un modèle trop simple. L’analyse des performances permet ensuite de vérifier quel modèle est réellement le meilleur.

Conclusion générale

Ce projet avait pour objectif d'exploiter un ensemble de données sport-santé afin de construire trois systèmes de prédiction :estimer les calories brûlées lors d'une séance, identifier le type d'entraînement, déterminer si un individu peut être considéré comme healthy ou non.

Pour chaque problématique, un travail complet a été réalisé : exploration du dataset, sélection rigoureuse des variables, choix des modèles, validation croisée, comparaison des performances et interprétations simples et compréhensibles. Les résultats principaux peuvent être résumés ainsi :

- Régression **(Calories_Burned)** : la régression linéaire est le modèle le plus performant. Elle montre que la dépense énergétique dépend surtout de la durée de la séance, de l'expérience, de la masse musculaire et de certaines caractéristiques physiques.
- Classification multi-classes **(Workout_Type)** : la régression logistique fournit les meilleures performances. Bien que ce modèle soit simple, il parvient à distinguer les quatre catégories d'entraînement grâce à des relations linéaires entre les variables clés (durée, calories brûlées, expérience, poids).
- Classification binaire **(is_healthy)** : le **KNN** est le modèle le plus efficace. Il classe correctement les individus healthy ou non healthy en s'appuyant sur leur alimentation, leur hydratation, leurs habitudes sportives et leur morphologie.

Au-delà des performances brutes, ce projet a permis de développer une démarche structurée :

- meilleure compréhension des données,
- importance du prétraitement et de la sélection des variables,
- nécessité d'éviter les fuites d'informations,
- intérêt de la validation croisée pour évaluer la robustesse d'un modèle,
- apport des visualisations pour comprendre les relations entre les variables.

Cette étude montre qu'il est possible, même à partir d'un dataset imparfait, de construire des modèles simples, interprétables et cohérents.