Rapport Machine Learning S8

RAPPORT PORTANT SUR LA CREATION D'UN MODELE DE MACHINE LEARNING POUR LA PREDICTION DE MALADIE CARDIAQUE EN UTILISANT LA REGRESSION LOGISTIQUE

Bedino Gabriel

TABLE DES MATIERES

1) Traitement de la base de données

Afin de préparer les données pour l'entraînement d'un modèle de machine learning de prédiction du risque de maladie cardiaque, plusieurs étapes de prétraitement ont été réalisées. L'objectif était de s'assurer que les données soient exploitables, homogènes et optimisées pour l'apprentissage.

Chargement et exploration des données

Nous avons importé notre base de données (fichier CSV) à l'aide de la bibliothèque pandas. Cette étape a permis de visualiser les premières lignes et de commencer l'analyse exploratoire des variables disponibles.

Gestion des valeurs manquantes

Nous avons choisi de supprimer les lignes contenant des valeurs manquantes, afin d'éviter d'introduire du biais par imputation artificielle. Après cette étape, l'échantillon final contenait un volume réduit mais propre.

Séparation des variables

Nous avons défini :

- **X** : l'ensemble des variables explicatives (par exemple : âge, cholestérol, tabac, pression artérielle)
- **y** : la variable cible : TenYearCHD (risque de maladie cardiaque à 10 ans, binaire 0/1)

Identification des types de colonnes et transformation des données

Les colonnes numériques (âge, pression, cholestérol, IMC, etc.) ont été standardisées via StandardScaler (moyenne = 0, écart-type =1), indispensable pour les modèles sensibles à l'échelle.

Les colonnes binaires (sexe, fumeur, diabète, AVC, hypertension) ont été codées en 0/1 via des menus clairs pour l'interface.

2) Recherche du meilleur modèle

Une fois les données préparées, nous avons comparé plusieurs modèles de machine learning pour identifier celui offrant les meilleures performances en prédiction.

Modèles comparés

Nous avons sélectionné quatre modèles de régression classiques et robustes :

- **Logistic Regression :** modèle explicatif, simple, interprétable.
- **Random Forest Classifier :** ensemble d'arbres de décision, robuste au bruit**.**
- **Gradient Boosting Classifier :** modèle performant optimisé itérativement.

Chaque modèle a été intégré dans un pipeline de prétraitement et entraîné sur les mêmes données.

Séparation des données

Avant l'entraînement, nous avons séparé notre jeu de données en un ensemble d'entraînement et un ensemble de test :

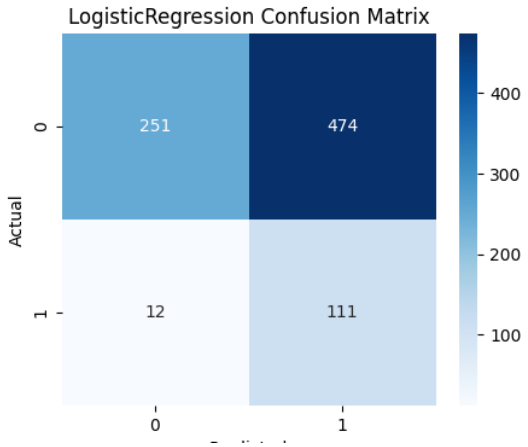
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Cela permet d'évaluer les performances du modèle sur des données jamais vues pendant l'entraînement, ce qui est indispensable pour tester la généralisation.

Évaluation des performances

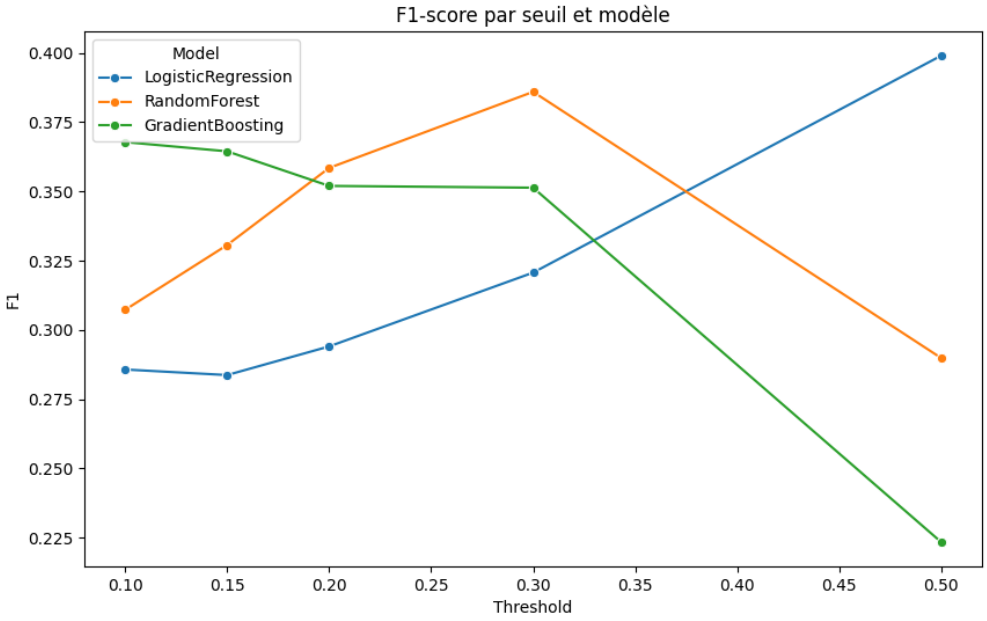
Pour chaque modèle, nous avons mesuré :

- **L'accuracy** : proportion de bonnes prédictions.
- **La précision** : proportion de vrais positifs parmi les prédits positifs.
- **Le recall** : proportion de vrais positifs détectés.
- **Le F1-score** : équilibre entre précision et recall.
- Une matrice de confusion a été tracée pour analyser les erreurs.



Matrice de confusion pour logistique regression à 0.15 treeshold

Résultats et sélection



Le modèle Logistic Regression (avec un seuil de 0.5) a été retenu, offrant le meilleur compromis précision/recall sur les classes déséquilibrées.

3) Développement du modèle et optimisation des hyperparamètres

Après avoir identifié le Logistic Regression comme le modèle le plus performant dans notre étude comparative, nous avons poursuivi avec une phase d'optimisation des hyperparamètres.

Cette étape permet de régler finement le comportement du modèle pour obtenir des performances maximales.

Pourquoi optimiser les hyperparamètres ?

Les modèles de machine learning ont des hyperparamètres, c'est-à-dire des paramètres externes au modèle, qu'il faut régler manuellement ou automatiquement (contrairement aux paramètres internes, appris pendant l'entraînement).

Les hyperparamètres influencent :

- La régularisation (C, penalty) ;
- Le solveur d'optimisation (solver) ;
- La robustesse au déséquilibre des classes (class_weight).

Une bonne combinaison permet d'améliorer significativement la précision et la robustesse du modèle. Une bonne combinaison de ces valeurs peut améliorer considérablement la précision et la robustesse du modèle.

Méthode d'optimisation utilisée

Nous avons utilisé GridSearchCV :

- Recherche exhaustive sur un espace réduit de combinaisons.
- Validation croisée à 5 plis.
- Évaluation selon le F1-score.

```python
for name, (model, params) in models_params.items():
    grid = GridSearchCV(model, params, cv=5, scoring='f1', n_jobs=-1)
    grid.fit(X_train_res, y_train_res)
    best_estimator = grid.best_estimator_
    y_proba = best_estimator.predict_proba(X_test)[:, 1]

    for threshold in thresholds:
        y_pred = (y_proba >= threshold).astype(int)
        acc = accuracy_score(y_test, y_pred)
        prec = precision_score(y_test, y_pred, zero_division=0)
        rec = recall_score(y_test, y_pred, zero_division=0)
        f1 = f1_score(y_test, y_pred, zero_division=0)
        results.append({
            'Model': name,
            'Threshold': threshold,
            'Accuracy': acc,
            'Precision': prec,
            'Recall': rec,
            'F1': f1
        })
```

Résultats et sélection du modèle optimisé

À l'issue des recherches, le meilleur modèle a été sélectionné avec :

- C = optimal,
- penalty = l1/l2,
- solver = liblinear.

Le modèle final a légèrement amélioré le F1-score et réduit les faux négatifs.

4) Création de l'interface utilisateur

Afin de rendre le modèle de prédiction accessible à des utilisateurs non techniques, nous avons développé une interface utilisateur intuitive et interactive à l'aide de la bibliothèque Streamlit. Cette application web permet de saisir des caractéristiques de véhicule et d'obtenir instantanément une estimation du prix de vente.

Objectifs de l'interface

L'objectif de l'interface est double :

- Faciliter l'utilisation du modèle de prédiction pour tout utilisateur (professionnel ou particulier) ;
- Améliorer l'expérience utilisateur en proposant une interface claire, responsive, et enrichie de fonctionnalités utiles.

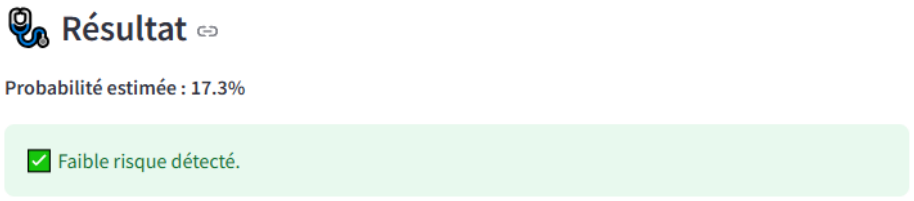
Fonctions principales intégrées

- **Saisie intuitive des caractéristiques**

  - Menus déroulants (sexe, fumeur, diabète).
  - Sliders interactifs (âge, cholestérol, tension, IMC).
  - Indications d'unités et explications pour chaque variable.

- **Prédiction automatique du risque**

  - Affichage de la probabilité et verdict.
  - Jauge graphique de visualisation.



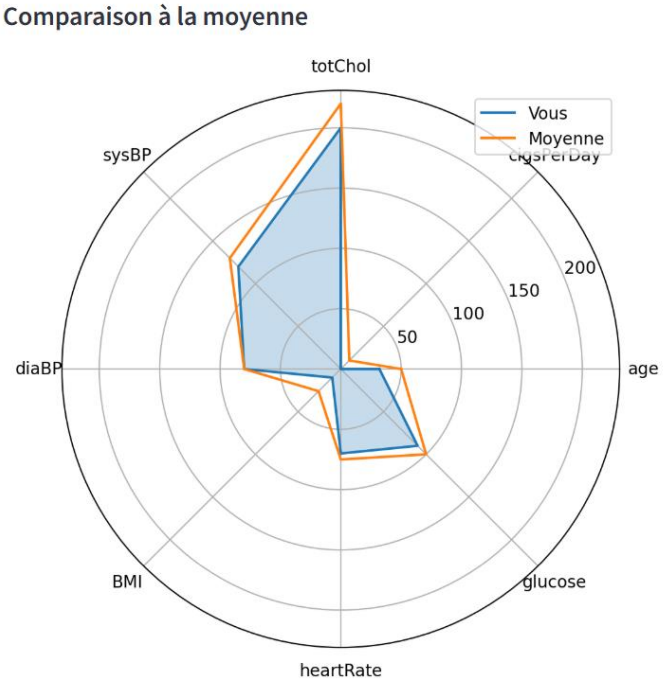
- **Historique des prédictions**

  - Table affichant toutes les prédictions réalisées pendant la session.

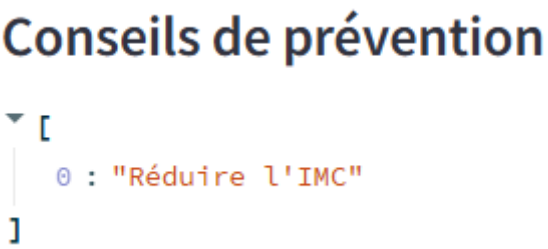
Historique de vos prédictions

	probabilité	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	preva
0	0.5406	1	50	1	1	0	1	1	
1	0.5406	1	50	1	1	0	1	1	
2	0.6396	1	50	1	1	50	1	1	

- **Visualisation interactive**
  - Graphique radar comparant les valeurs de l'utilisateur à la moyenne populationnelle.



- **Conseils de prévention**
  - Suggestions automatiques si certaines variables dépassent des seuils de risque.



- **Rapport PDF téléchargeable**
  - Rapport résumé incluant les données, la probabilité et les conseils.