Projet IAS

Prédiction du risque du diabète

Auteurs : Ghiles Kemiche

Nom de l'organisme : Département d'informatique
Professeur Encadrant : Paul Lerner

Licence double diplôme Mathématiques - Informatique

2022 - 2023

Table des matières

1

Introduction

Le diabète est une maladie chronique grave qui peut causer des complications majeures comme des maladies cardiaques, etc. Bien qu'il n'y ait pas de cure pour le diabète, les stratégies de traitement peuvent aider à atténuer les effets nocifs de la maladie. Les modèles prédictifs pour le risque de diabète sont donc importants pour les responsables de la santé publique, car un diagnostic précoce peut conduire à des changements de mode de vie et à un traitement plus efficace.

1.1 Données

Pour ce projet, un fichier csv du jeu de données disponible sur Kaggle pour l'année 2015 a été utilisé. Ce jeu de données contient 253 680 réponses à l'enquête BRFSS2015 du CDC. La variable cible Diabetes012 a 3 classes. 0 représente l'absence de diabète ou seulement pendant la grossesse, 1 représente le prédiabète et 2 représente le diabète. Il y a un déséquilibre de classes dans ce jeu de données. Ce jeu de données comporte 21 variables explicatives (Features).

1.2 Plan de traitement des donneées

Nous commençons d'abord par prétraiter les données en supprimant les redondances. Comme les classes sont mal réparties, nous allons essayer de les répartir équitablement. Étant donné que le nombre de diabétiques est faible, une répartition équitable

des trois classes entraînerait une perte importante de données. Pour éviter cela, une approche consiste à regrouper les classes 1 et 2 en une seule classe diabétique. Les individus pré-diabétiques sont inclus dans cette classe pour tenir compte du risque de développer un diabète. Ensuite, pour équilibrer l'ensemble de données, on peut sélectionner au hasard des échantillons de diabétiques et de non-diabétiques de manière à ce que chaque groupe représente 50% de l'ensemble de données. Il convient de souligner que certains individus peuvent ne pas encore avoir été diagnostiqués comme pré-diabétiques ou diabétiques.

1.3 Features et variable cible

Les features de nos donées sont : **'HighBP', 'HighChol', 'CholCheck', 'BMI', 'Smoker', 'Stroke', 'HeartDiseaseorAttack', 'PhysActivity', 'Fruits', 'Veggies', 'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'GenHlth', 'MentHlth', 'PhysHlth', 'DiffWalk', 'Sex', 'Age', 'Education', 'Income'**

Nous distinguons 3 types de variables pour les features :

- **Variables booléennes :** variables qui prennent soit 0 soit 1 et sont : HighBP, HighChol, CholCheck, Smoker, HeartDiseaseorAttackStroke, PhysActivity, Fruits, Veggies, HvyAlcoholConsump, AnyHealthcare, NoDocbcCost, DiffWalk, Sex.
- **Varibles dicrètes :** variables qui prennent des valeurs discretes en dehors des variables booleenes et sont : GenHlth, MentHlth, PhysHlth, Age, Education, Income.
- **Variables continues :** Varible qui prennent des valeurs continues, la seule variable continue est : BMI

Le target **'Diabetes_binary'** est donc est soit **1 :** la personne est atteinte du diabète **0 :** la personne n'est pas atteinte du diabète.

2

Préprocessing

Nous avons travaillé avec un jeu de données sur le diabète, contenant plusieurs types de variables :

- Les variables booléennes : ces variables indiquent la présence ou l'absence de certains symptômes chez les patients. Nous n'avons pas effectué de prétraitement supplémentaire, car ces variables sont déjà binaires.
- Les variables numériques discrètes : ces variables représentent des quantités entières pour chaque patient. Nous avons utilisé l'encodeur one-hot pour transformer ces variables en un format utilisable pour l'entraînement de notre modèle.
- Les variables numériques continues : ces variables représentent des quantités réelles pour chaque patient. Nous avons utilisé un échelleur standard pour standardiser ces variables afin de les mettre à l'échelle pour l'entraînement de notre modèle.

2.1 Prétraitement

Pour le prétraitement des données, nous avons utilisé le module sklearn.compose de Scikit-learn pour créer un transformateur de colonnes qui applique les prétraitements spécifiques à chaque type de variable
Ensuite, nous avons utilisé la méthode ColumnTransformer() pour appliquer ces prétraitements à chaque colonne spécifique de notre jeu de données. Enfin, nous avons concaténé les colonnes binaires transformées et les colonnes continues transformées pour obtenir notre ensemble de données prétraité final.

3

Modèles et Métriques d'évaluation

Dans cette étude, nous cherchons à sélectionner le meilleur modèle de classification pour prédire si un individu est atteint d'une maladie (1 : malade, 0 : non malade) en utilisant un ensemble de données. Notre démarche consiste à comparer plusieurs algorithmes de classification avec différents paramètres et choisir le meilleur modèle en fonction de la métrique appropriée.

3.1 Méthodologie

3.1.1 Préparation des données

Nous commençons diviser l'ensemble de données en ensembles d'entraînement et de test en utilisant la méthode `train_test_split` de la bibliothèque scikit-learn.

3.1.2 Choix des algorithmes et des paramètres

Nous sélectionnons plusieurs algorithmes de classification couramment utilisés pour comparer leurs performances. Pour chaque algorithme, nous définissons un ensemble de paramètres à tester. Les algorithmes et leurs paramètres sont les suivants :

Decision Tree, Random Forest, Logistic Regression, Support Vector Machine (SVM), XGBoost, K-neighrest, Naive Bayes,K-Nearest Neighbors

3.1.3 Ajustement des modèles et sélection des paramètres optimaux

Nous utilisons la méthode `GridSearchCV` de scikit-learn pour ajuster chaque modèle avec les différentes combinaisons de paramètres spécifiées. Cette méthode effectue une recherche exhaustive sur les paramètres en utilisant une validation croisée k-fold pour évaluer les performances de chaque combinaison de paramètres. À la fin de cette étape, nous obtenons les meilleurs paramètres pour chaque algorithme en termes de performance de validation croisée.

3.1.4 Comparaison des modèles et sélection du meilleur modèle

Après avoir ajusté chaque modèle avec les meilleurs paramètres, nous évaluons leurs performances sur l'ensemble de test. En fonction de la métrique choisie (F1-score ou AUC-ROC), nous sélectionnons le modèle avec la meilleure performance.