DIABETES DIAGNOSIS TASK COMPETITION

Clément Bourvic
2412279
clement.bourvic@telecom-sudparis.eu

Julien Segonne
2409827
julien.segonne@hotmail.com

Antoine Barberin
2390973
antoine.barberin@gmail.com

Nom d'équipe Kaggle : _datanonymous

ABSTRACT

Ce rapport présente notre travail sur la compétition Kaggle *Diabetes diagnosis task competition* dans la matière INF8245E. Il comprend l'explication de l'analyse des données que nous avons effectuée au préalable, les algorithmes que nous avons implémentés, la méthodologie adoptée ainsi qu'une analyse de nos résultats.

1 ANALYSE EXPLORATOIRE DES DONNÉES

L'analyse exploratoire des données a pour objectif de mieux comprendre la structure des données et d'identifier les relations entre les caractéristiques (*features*) et la variable cible, `Diabetes_binary`, représentant la présence ou l'absence de diabète. Cette exploration permet également de détecter des incohérences dans les données et de préparer des combinaisons logiques de *features* pertinentes.

1.1 DESCRIPTION DES DONNÉES

Le jeu de données de formation contient environ 203 000 exemples répartis sur 28 *features*, sans aucune valeur manquante, or un gros désequilibre de classe est présent, seulement 13% des exemples sont diabétique. Cependant, des incohérences ont été relevées :

- La variable `Age_Group` est incorrectement construite et nécessite une reconstruction.
- Certaines variables catégoriques existantes, telles que `Healthy_Diet` ou `Physical_Activity`, sont des combinaisons d'autres *features* présentes (*e.g.*, fruits et légumes pour `Healthy_Diet`).
- Incohérence observée : certains individus ayant un cholestérol élevé (`HighChol`) n'ont pas vérifié leur cholestérol (`CholCheck`).

1.2 PRÉPARATION DES DONNÉES

Pour affiner l'analyse, certaines étapes de prétraitement ont été réalisées :

- Création de nouvelles variables catégoriques à partir des variables continues, telles que `BMI` (catégorisé en *Normal*, *Overweight*, *Obese-1*, etc.) et `Age` (groupe d'âges).
- Identification et combinaison logique de *features*, telles que :
  - **`HighBP et HighChol`** : La combinaison de pression artérielle élevée et de cholestérol élevé (`HighBP_HighChol`) augmente le risque de diabète.

  - **`Healthy_Diet` et `Physical_Activity`** : La combinaison de ces facteurs réduit significativement le risque de diabète (`HealthyDiet_PhysActivity`).

- Vérification de la logique des variables existantes : par exemple, `Healthy_Diet` correspond à la somme logique des *features* `Fruits` et `Veggies`.

1.3 ANALYSE DES RELATIONS AVEC LA CIBLE

Chaque *feature* a été analysé pour son rapport avec la cible `Diabetes_binary`. Les résultats principaux incluent :

- Les variables liées à la santé, telles que `HighBP`, `HighChol`, `BMI`, et `Physical_Activity`, montrent une forte corrélation avec le diabète.
- Des combinaisons logiques, comme `HighBP_HighChol` et `HealthyDiet_PhysActivity`, permettent de capturer des interactions importantes.
- Certaines variables, comme `CholCheck`, présentent des incohérences avec les données (*e.g.*, des individus ayant un cholestérol élevé sans vérification préalable).

Comme illustré dans la Figure 1, la combinaison de `HighBP` et `HighChol` est significative.

1.4 CORRÉLATIONS AVEC LA CIBLE

Une analyse de corrélation a été réalisée pour évaluer les relations entre les caractéristiques (y compris les *features* créées) et la variable cible `Diabetes_binary`. Ces visualisations permettent d'identifier les *features* les plus pertinentes pour la modélisation.

Les résultats montrent que certaines caractéristiques, telles que `GenHlth` et `DiffWalk`, présentent une forte corrélation positive avec la cible, indiquant une influence notable sur le diabète. À l'inverse, des *features* comme `Income` ou `PhysActivity` présentent une forte corrélation négative, suggérant qu'elles protègent contre le diabète. Par ailleurs, des variables telles que `Sex` ou `AnyHealthCare` présentent une corrélation quasi nulle avec la cible et sont donc peu informatives pour des modèles linéaire.

Ces observations sont illustrées dans la Figure 2, qui présente la matrice de corrélation entre toutes les *features* et la variable cible.

1.5 CONCLUSION DE L'EDA

L'analyse exploratoire des données a permis de détecter plusieurs incohérences, d'identifier des regroupements logiques de *features*, et de mieux comprendre les relations clés entre les variables explicatives et la cible. Ces résultats constituent une base solide pour la sélection des caractéristiques et l'entraînement des modèles. À l'issue de cette analyse, les décisions suivantes ont été prises concernant la suppression de certaines variables :

- `Age_Group` : incohérente.
- `CholCheck` : absence de corrélation avec la cible.
- `AnyHealthcare` : absence de corrélation avec la cible.
- `MentHlth` : utilisation de `Mental_Health_Risk`, qui présente une meilleure corrélation.
- `Fruits` et `Veggies` : fusionnées, car leur combinaison présente une corrélation plus élevée.
- `BMI_Category` : utilisation de `BMI`, qui offre une meilleure corrélation sans catégorisation.

2 ALGORITHMES

2.1 RANDOM FOREST

Le modèle *Random Forest* est un ensemble d'arbres de décision, où chaque arbre est entraîné sur un sous-ensemble de données. Ce modèle est robuste face aux données bruitées et aux valeurs aberrantes.

Avantages de Random Forest :

- Gestion efficace des données déséquilibrées.
- Flexibilité pour des données pouvant être utilisées

Hyperparamètres principaux :

- Le paramètre *n_estimators* influe directement sur la robustesse et la stabilité du modèle, tandis que *max_depth* joue un rôle crucial dans la gestion du surapprentissage, ce qui est particulièrement important compte tenu de la complexité des données. Les paramètres *min_samples_split* et *min_samples_leaf* permettent de contrôler la complexité des arbres, limitant ainsi le risque de surajustement. Enfin, le choix du *criterion* est essentiel pour capturer efficacement les interactions entre les différentes caractéristiques.
- Les recherches ont montré que les meilleurs hyperparamètres dans ce cas étaient un *n_estimators* autour de 500, une profondeur maximale (*max_depth*) fixée à *None* ou 25, et une préférence systématique pour le critère *entropy*.

2.2 XGBOOST

XGBoost est un algorithme de boosting basé sur les arbres de décision, conçu pour optimiser les performances tout en maintenant une faible complexité. Il applique un ensemble d'arbres successifs, où chaque nouvel arbre corrige les erreurs des précédents. **Avantages de XGBoost :**

- Gestion automatique des valeurs manquantes.
- Pondération des classes via le paramètre `scale_pos_weight` pour traiter le déséquilibre des classes.
- Optimisations internes comme le parallélisme et la régularisation (via les paramètres `gamma`, `reg_alpha`, et `reg_lambda`).

Hyperparamètres principaux :

- Les paramètres *n_estimators* et *max_depth* jouent un rôle similaire à ceux de *Random Forest*, en influençant respectivement le nombre total d'arbres et la complexité des modèles. Le *learning_rate* impacte directement la vitesse d'apprentissage et la convergence du modèle, ce qui en fait un hyperparamètre essentiel à optimiser. De plus, *subsample* et *colsample_bytree* déterminent la fraction des échantillons et des caractéristiques utilisées pour chaque arbre, ce qui améliore la diversité des arbres générés, un aspect crucial compte tenu de la complexité des données.
- Les recherches ont montré que les configurations optimales incluent un *n_estimators* entre 200 et 400, une *max_depth* fixée à 7, et un *learning_rate* idéalement ajusté à 0.1.

2.3 SVM (SUPPORT VECTOR MACHINE)

Le Support Vector Machine (SVM) est une méthode d'apprentissage supervisé qui vise à trouver un hyperplan séparant les classes dans un espace à dimensions élevées. Il maximise la marge entre les points de données les plus proches de l'hyperplan (les vecteurs de support), garantissant une meilleure généralisation du modèle.

Avantages du SVM :

- Performances élevées sur des jeux de données avec peu de bruit et bien séparés.

- Flexibilité grâce aux noyaux (linéaire, polynomial, gaussien, etc.) pour gérer des relations non linéaires.
- Robustesse contre le surapprentissage, surtout avec des jeux de données de petite taille.

Hyperparamètres principaux :

- Le paramètre *C* est le paramètre de régularisation. Une valeur faible favorise un modèle plus simple avec une marge large, tandis qu'une valeur élevée vise à minimiser les erreurs d'entraînement mais peut avoir pour conséquence du surapprentissage. La valeur optimale trouvée lors de la grid search est 1.
- Le paramètre *kernel* définit la fonction noyau utilisée pour transformer les données. Le choix du noyau est primordial car il détermine comment les données sont projetées dans un espace de caractéristiques plus élevé. Le noyau choisi est *rbf* (radial basis function).
- Le paramètre *gamma* est le coefficient de la fonction noyau. Une valeur élevée concentre l'impact sur des points proches, tandis qu'une valeur faible généralise davantage mais rend le modèle plus rigide. Le gamma choisi est 0.1.

2.4 MLP (MULTI-LAYER PERCEPTRON)

Le Multi-Layer Perceptron (MLP) est un modèle d'apprentissage supervisé basé sur des réseaux de neurones artificiels. Il est composé de couches de neurones organisées en une architecture hiérarchique : une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie. Chaque neurone applique une fonction d'activation non linéaire pour modéliser des relations complexes dans les données.

Avantages du MLP :

- Capacité à capturer des relations complexes et non linéaires entre les variables.
- Extensibilité : adapté aussi bien aux tâches de classification qu'à celles de régression.
- Hautement paramétrable via le choix des hyperparamètres (nombre de couches, de neurones, type de fonction d'activation, etc.).

Hyperparamètres principaux :

- Le paramètre *hidden_layer_sizes* détermine le nombre de neurones et la structure des couches cachées. Ce paramètre permet de capturer la complexité des données. Trop de neurones résulteront à du surapprentissage.
- Le paramètre *activation* correspond à la fonction d'activation utilisée pour chaque neurone. Le choix de cette fonction influence la capacité du modèle à modéliser des relations non linéaires. La grid search a systématiquement sélectionné la fonction relu.
- Le paramètre *alpha* contrôle la régularisation L2, qui pénalise les poids élevés afin de réduire le risque de surapprentissage. Une valeur plus élevée provoque une régularisation plus forte.
- Le paramètre *learning_rate_init* correspond au taux d'apprentissage initial.

3 MÉTHODOLOGIE

3.1 PRÉPARATION DES DONNÉES

- **Traitement du déséquilibre des classes :** Application de *SMOTEENN* pour augmenter les exemples de la classe minoritaire (en créant des exemples synthétiques à l'aide de la méthode des k-plus proches voisins) et nettoyer les points mal classés de la classe majoritaire. Cela a aussi pour effet de réduire le surapprentissage.
- **Encodage et normalisation :** Encodage des variables catégoriques avec `OneHotEncoder` et normalisation des variables numériques avec `RobustScaler` pour réduire l'impact des valeurs aberrantes.

3.2 DIVISION DES DONNÉES

Les données ont été divisées en deux sous-ensembles : 80% pour l'entraînement et 20% pour le test. Une division stratifiée a été utilisée pour conserver la distribution des classes dans chaque sous-ensemble.

3.3 OPTIMISATION DES HYPERPARAMÈTRES

- **Recherche d'hyperparamètres :** Utilisation de `HalvingRandomSearchCV`, une méthode efficace de recherche avec élimination progressive des candidats moins prometteurs en augmentant de plus en plus les données utilisées.
- **Approche progressive pour la sélection des hyperparamètres :** Une première exploration a été réalisée sur une plage large d'hyperparamètres, suivie d'un affinement basé sur les résultats obtenus lors de cette recherche initiale.
- **Parallélisation :** Optimisation de la recherche en utilisant plusieurs cœurs du CPU pour accélérer le processus d'exploration des hyperparamètres.

3.4 VALIDATION ET MÉTRIQUES

Une validation croisée à 5 folds a été utilisée pour évaluer les performances des modèles lors de l'entraînement. La métrique principale choisie est le score F1, qui équilibre précision et rappel dans un contexte de classification déséquilibrée.

3.5 AJUSTEMENT DU SEUIL DE CLASSIFICATION

Pour chaque modèle, un seuil optimal a été déterminé à partir de la courbe Précision-Rappel, en maximisant le score F1. Les prédictions finales sur les données de test ont été ajustées selon ce seuil.

4 RÉSULTATS

4.1 RANDOM FOREST

Les résultats 1 obtenus avec le modèle Random Forest montrent une précision globale satisfaisante pour la classe (0), avec un F1-score de 89 %. Cependant, la performance sur la classe minoritaire (1) reste faible, avec un F1-score de 44 % et un rappel de 52 %. Cela indique que le modèle détecte légèrement plus de cas positifs par rapport aux autres modèles, mais génère encore un nombre important de faux négatifs. L'exactitude globale de 82 % reflète principalement le bon classement de la classe majoritaire.

4.2 XGBOOST

Le modèle XGBoost offre des performances 2 globales similaires à celles de Random Forest, avec un F1-score légèrement inférieur pour la classe majoritaire (87 %) mais une amélioration notable du rappel pour la classe minoritaire (61 %). Cela se traduit par une meilleure détection des cas positifs, bien que la précision pour cette classe reste faible (36 %), ce qui entraîne un grand nombre de faux positifs. Avec une exactitude globale de 79 %, le modèle reste biaisé vers la classe majoritaire. Cependant, l'amélioration du rappel pour la classe minoritaire peut le rendre plus adapté, il est généralement mieux de classé un non-diabétique diabétique que l'inverse.

4.3 SVM

Les performances 3 du SVM sont assez contrastées. Bien que le F1-score de la classe majoritaire soit élevé (83 %), les résultats pour la classe minoritaire montrent un F1-score faible de 44 %, malgré un rappel élevé de 74 %. Cela signifie que le modèle identifie de nombreux cas positifs, mais au prix d'une précision très faible (31 %), générant un grand nombre de faux positifs. L'exactitude globale de 74 % reflète ces déséquilibres. Ce modèle pourrait être envisagé si la priorité est d'assurer un

rappel maximal pour la classe minoritaire, bien que cela doive être compensé par une gestion des faux positifs.

4.4 MLP

Les résultats 4 du MLP ne sont pas vraiment satisfaisants car ils n'offrent pas une amélioration significative du F1 score. Nous pouvons constater que le F1 score sur l'ensemble de test est encore faible (40%). Le modèle obtient 48% de recall ce qui montre qu'il génère encore trop de faux négatifs (très problématique dans ce cas d'étude), de plus sa précision est seulement de 34% ce qui signifie qu'il crée un très grand nombre de faux positifs, ces deux résultats expliquent le F1 score aussi bas. Enfin, 80% des patients ont bien été classés, toutefois ce chiffre est discutable : classé un non-diabétique comme diabétique n'a pas de grandes conséquences, en revanche ne pas détecter une personne diabétique peut s'avérer être grave.

5 CONCLUSION

Dans cette étude, nous avons exploré différents algorithmes de machine learning pour diagnostiquer le diabète à partir d'un jeu de données complexe et déséquilibré. Nos travaux ont mis en évidence plusieurs points importants :

L'analyse exploratoire des données a révélé des incohérences et des niveaux de corrélations différents entre les données et la sortie attendu. Quatre modèles ont été évalués - Random Forest, XGBoost, SVM et MLP - chacun présentant des forces et des faiblesses distinctes face à ce défi de classification.

Random Forest a obtenu les meilleures performances globales avec 82% d'exactitude mais c'est XGBoost qui a donné le meilleur score dans la compétition Kaggle, sûrement car il a obtenu un meilleur score F1 pour la classe minoritaire SVM et MLP ont été moins performants pour la détection des cas de diabète

Le défi majeur reste la prédiction de la classe minoritaire (patients diabétiques), avec des F1-scores variant entre 40% et 45% au mieux pour cette classe.

Pour améliorer la précision de notre détection, nous pourrions faire un grid search plus approfondie en testant plus de paramètres ou peut-être essayer d'utiliser un LLM pré-entraîné et de le fine-tuner sur notre tâche de classification car la plupart des données importantes sont textuelles ou peuvent être transformées en données textuelles.

Les résultats soulignent la complexité du diagnostic du diabète et la nécessité de développer des modèles capables de minimiser les faux négatifs, étant donné les conséquences potentiellement graves d'un diagnostic manqué en médecine.

6 STATEMENT OF CONTRIBUTIONS

Clément :

- Exploration des données et mise en place du pré-traitement
- Tests de combinaisons de features
- Implémentation de l'équilibrage des données à l'aide de SMOTEENN
- Implémentation, exécution et analyse des résultats du modèle Random Forest
- Implémentation, exécution et analyse des résultats du modèle XGBoost
- Exécution du modèle SVM avec d'autres hyper-paramètres
- Tests d'autres méthodes de recherche d'hyperparamètres, comme Optuna
- Rédaction du rapport

Antoine :

- Exécution et analyse des résultats du second modèle MLP

- Exécution du premier modèle MLP avec une nouvelle grille de recherche pour les hyper-paramètres
- Ajustement des données : tests après manipulation du dataset (suppression et/ou ajout de nouveaux features, normalisation...)
- Correction de bugs
- Rédaction du rapport
- Essai d'entraînement d'un MLP en privilégiant le Auc-Score pour améliorer les prédicitions sur un jeu de données ayant une disproportion de classes

Julien :

- Implémentation, exécution et analyse des résultats du modèle SVM
- Implémentation, exécution et analyse des résultats du modèle MLP
- Exécution du modèle XGBoost avec une nouvelle grille de recherche pour les hyper-paramètres
- Ajustement des données : tests après manipulation du dataset (suppression et/ou ajout de nouveaux features, normalisation...)
- Correction de bugs
- Rédaction du rapport
- Rédaction du fichier Readme

A ANNEXES

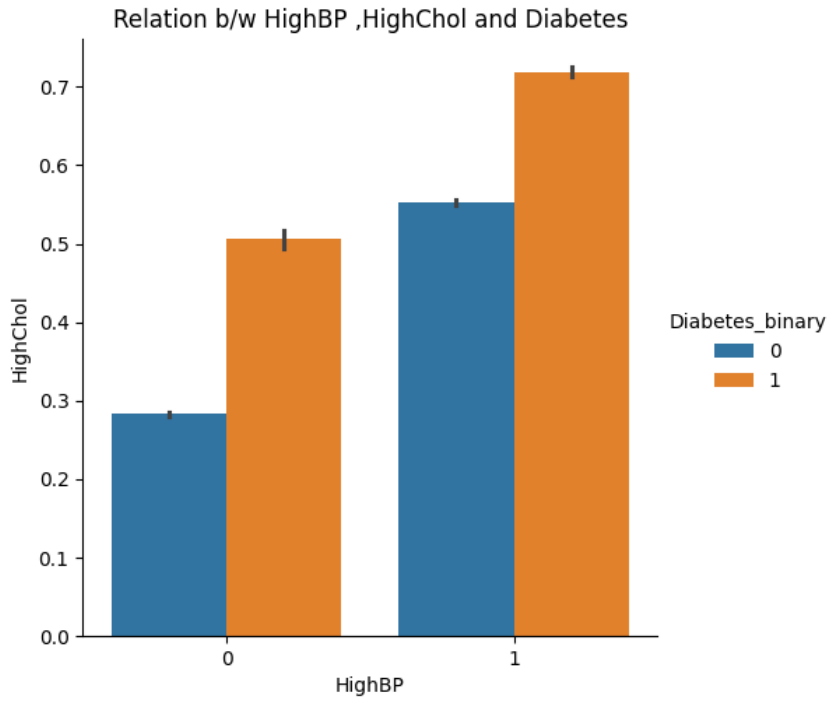

Figure 1: Relation entre `HighBP+HighChol` et la variable cible `Diabetes_binary`.

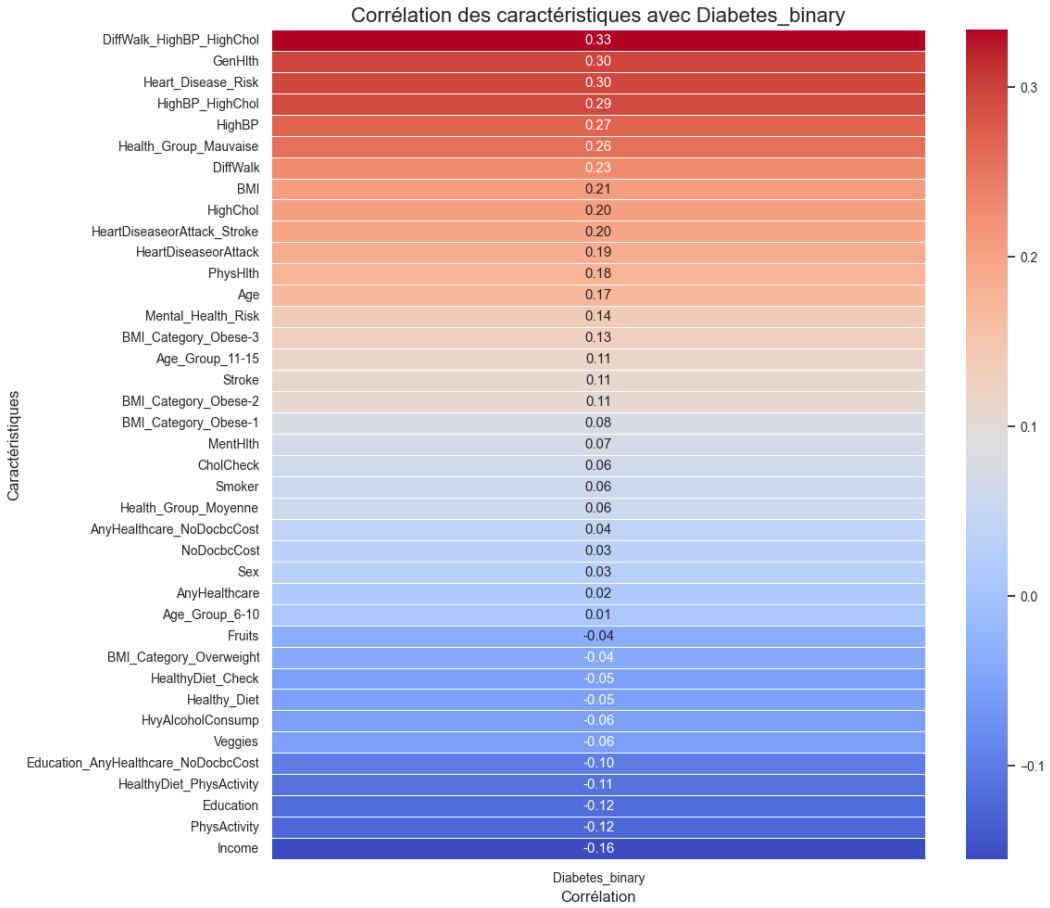

Figure 2: Matrice de corrélation des caractéristiques avec la variable cible `Diabetes_binary`.

	Precision	**Recall**	**F1-score**	**Support**
0	0.92	0.87	0.89	34919
1	0.39	0.52	0.44	5670
accuracy			0.82	40589
macro avg	0.65	0.69	0.67	40589
weighted avg	0.84	0.82	0.83	40589

Table 1: Testing Set Classification Report Random Forest

	Precision	**Recall**	**F1-score**	**Support**
0	0.93	0.82	0.87	34919
1	0.36	0.61	0.45	5670
accuracy			0.79	40589
macro avg	0.64	0.72	0.66	40589
weighted avg	0.84	0.79	0.81	40589

Table 2: Testing Set Classification Report XGBoost

	Precision	**Recall**	**F1-score**	**Support**
0	0.95	0.73	0.83	34919
1	0.31	0.74	0.44	5670
accuracy			0.74	40589
macro avg	0.63	0.74	0.63	40589
weighted avg	0.86	0.74	0.77	40589

Table 3: Testing Set Classification Report SVM

	Precision	**Recall**	**F1-score**	**Support**
0	0.91	0.85	0.88	34919
1	0.34	0.48	0.40	5670
accuracy			0.80	40589
macro avg	0.62	0.66	0.64	40589
weighted avg	0.83	0.80	0.81	40589

Table 4: Testing Set Classification Report MLP