[SLIDE 3]

Tout au long de la démarche de modélisation, il s'agira de trouver un compromis, comme c'est toujours le cas, entre le biais et la variance des prévisions, ce compromis étant expliqué en annexe 1. On a donc essayé d'inclure le maximum de features significatives tout en contrôlant le risque d'overfitting, et aussi bien évidemment de prétraiter les données du modèle

[SLIDE 5]

On a conservé tous les immeubles quel que soit leur compliance status, cet état n'indiquant que le fait que le rapport énergétique a été rendu ou non avant la date légale annuelle imposée par la ville de Seattle et étant donc sans conséquence sur la qualité ou la complétude des données

[SLIDE 9]

- Age median 58 ans
- Env. 1 immeuble sur 5 a moins de 20 ans et 2 sur 5 plus de 60 ans
- Nombre significatif d'immeubles centenaires (189, soit env. 13%)
- Plus de 9 immeubles sur 10 ont moins de 10 étages

[SLIDE 10]

- Usages principaux : bureaux et les entrepôts
- Types d'énergie principaux : gaz ET électricité (2/3), électricité seule env. ¼
- Pas de corrélation apparente entre age et ENERGYSTARScore

[SLIDE 11]

Utilisation pipelines différents selon modèle ensembliste ou non aux étapes 4 et 9

Séparation train / val / test faite avant iterative imputer et avant standard scaler & correction du skew dans le pipeline pour éviter le data leakage.

Une fois le meilleur modèle détermine sur le jeu de validation, on réentraînera le modèle sur le jeu de train + validation avant de le tester sur le jeu de test pour l'évaluation finale de la performance.

L'Energy Star Score contient plus de 30% de valeurs manquantes, on a donc choisi de l'imputer plutôt que de supprimer les immeubles pour lesquels sa valeur est manquante.

[SLIDE 12]

K-fold teste entre 2 et 10 folds => on a gardé 10 qui donnait le meilleur résultat et « économiser » sur les itérations et le fit time

Voir notes méthodologiques ci-après concernant le K-fold + l'intérêt de la stratification pour obtenir des résultats reproductibles (et non liés au hasard de l'échantillonnage)

[SLIDE 15]

L'optimisation a été faite avec la méthode GridSearchCV de ScikitLearn - convergence difficile sur GHGEmissions d'où paramètres testés plus nombreux

[SLIDE 16] & [SLIDE 17]

Critères d'évaluation – R2 et MAE + fit time

Difficultés de modélisation de GHGE => plus de paramètres testés et inclusion des prédictions de consommation énergétiques dans la modélisation

[SLIDE 18]

On réentraîne le meilleur modèle sur le jeu de train + validation et on l'évalue sur le jeu de test (complètement inconnu, donc) – les résultats sont confirmés par un R2 relativement bon, puisque notre modélisation explique dans les 2 cas

près des 2/3 des variations de la variable cible – sans différences notoire si l'on inclut l'Energy star score.

[SLIDE 21]

Il s'agit du même immeuble numéro 42.

Annexe 1 – compromis biais variance

Annexe 2 – Il semblerait en fait que l'inclusion de l'Energie star score améliore les prédictions de la consommation d'Energie pour un sous-groupe d'immeubles (points jaunes proches de la bissectrice sur le graphique de gauche) dont il conviendrait d'étudier les caractéristiques – pas raisonnable considère le temps imparti a ce projet mais c'est une piste d'amélioration.

Par ailleurs, la modélisation de GHGE est un exercice artificiel et parfaitement inutile, cette donnée pouvant se déduire à partir de SiteEnergyUseWN(kBtu)_pred et d'une formule (cf. description des variables sur website ville de Seattle).

Notes méthodologiques

1 - Régression linéaire

Le modèle de régression linéaire dans Scikit-learn est un algorithme d'apprentissage supervisé utilisé pour prédire une variable cible continue à partir d'une ou plusieurs variables indépendantes. Voici un aperçu de son fonctionnement et de ses caractéristiques principales.

La régression linéaire cherche à établir une relation linéaire entre les variables indépendantes (features) et la variable dépendante (target). Elle peut être représentée par l'équation :

$$y = b_0 + b_1 x_1 + b_2 x_2 + ... + b_n x_n$$

où yy est la variable cible, b0b0 est l'ordonnée à l'origine (intercept), et b1,b2,...,bnb1,b2,...,bn sont les coefficients associés aux variables indépendantes x1,x2,...,xnx1,x2,...,xn respectivement

La régression linéaire est appréciée pour sa simplicité et sa capacité à fournir des interprétations claires des relations entre les variables. Elle est largement utilisée dans des

domaines comme l'économie, la biologie, et l'ingénierie pour des tâches telles que la prévision des ventes ou l'analyse des tendances

La régression linéaire standard (via LinearRegression dans Scikit-learn) n'inclut pas de régularisation, ce qui signifie qu'elle cherche simplement à minimiser l'erreur quadratique entre les valeurs prédites et réelles. Cela peut entraîner un surajustement lorsque le modèle est trop complexe par rapport à la quantité de données disponibles.

2 - Elastic Net :

Elastic Net, quant à lui, combine deux types de pénalités : L1 (Lasso) et L2 (Ridge). Cette combinaison permet de contrôler la complexité du modèle tout en maintenant une certaine flexibilité. Elastic Net est particulièrement utile lorsque les données contiennent des caractéristiques corrélées, car il peut sélectionner plusieurs variables corrélées simultanément, contrairement à Lasso qui en choisit généralement une seule.

Les pénalités Ridge et Lasso sont deux techniques de régularisation utilisées dans la régression pour éviter le surajustement et améliorer la performance prédictive des modèles. Voici une explication détaillée de chacune :

Pénalité Ridge (L2)

La régression Ridge, également connue sous le nom de régularisation L2, ajoute une pénalité basée sur la somme des carrés des coefficients du modèle. La fonction de coût modifiée pour la régression Ridge est :

$$\text{Loss} = \text{MSE} + \lambda \sum_{i=1}^{n} w_i^2$$

où :

- MSE est l'erreur quadratique moyenne,
- λ est un hyperparamètre qui contrôle l'intensité de la régularisation,
- wi sont les coefficients du modèle.

Caractéristiques de la pénalité Ridge :

- Réduction des coefficients : Elle réduit les valeurs des coefficients, mais ne les force pas à être exactement zéro. Cela signifie que tous les prédicteurs restent dans le modèle, mais avec des coefficients plus petits.
- Multicolinéarité : Ridge est particulièrement efficace lorsque les variables indépendantes sont corrélées, car elle stabilise les estimations des coefficients en les contraignant.
- Interprétation : Moins interprétable que Lasso, car elle n'effectue pas de sélection de caractéristiques.

Pénalité Lasso (L1)

La régression Lasso, ou régularisation L1, impose une pénalité basée sur la somme des valeurs absolues des coefficients. La fonction de coût pour Lasso est formulée comme suit :

$$\text{Loss} = \text{MSE} + \lambda \sum_{i=1}^{n} |w_i|$$

Caractéristiques de la pénalité Lasso :

- Sparsité : Lasso peut réduire certains coefficients à zéro, ce qui permet d'effectuer une sélection automatique des caractéristiques. Cela simplifie le modèle en éliminant les variables non significatives.
- Interprétabilité : En réduisant le nombre de variables dans le modèle, Lasso rend les résultats plus faciles à interpréter.
- Utilisation : Idéal pour les situations où l'on s'attend à ce que seules quelques variables soient réellement pertinentes pour la prédiction.

En résumé, Ridge et Lasso sont deux techniques complémentaires qui aident à améliorer la performance des modèles en régularisant les coefficients. Le choix entre ces deux méthodes dépend souvent de la nature des données et des objectifs spécifiques du modèle.

3 - Principe du KNN en régression

Dans le cadre de la régression KNN, l'algorithme prédit la valeur d'une variable cible en se basant sur les valeurs des k voisins les plus proches dans l'espace des caractéristiques. La prédiction est généralement calculée comme la moyenne (ou une moyenne pondérée) des valeurs cibles des voisins sélectionnés. Si k=1, la valeur prédite sera simplement celle du voisin le plus proche.

Avantages de KNN pour la régression

1. Simplicité et intuitivité : KNN est facile à comprendre et à mettre en œuvre, ce qui en fait un bon choix pour les débutants en apprentissage automatique
2. Aucune hypothèse sur la distribution des données : Contrairement à d'autres méthodes de régression qui nécessitent des hypothèses sur la distribution des données (comme la normalité), KNN ne fait aucune supposition à cet égard, ce qui le rend flexible pour différents types de données

3. Capacité à capturer des relations complexes : KNN peut modéliser des interactions complexes entre les variables sans avoir besoin de définir un modèle statistique paramétrique

Inconvénients de KNN pour la régression

1. Coût computationnel élevé : KNN nécessite de calculer les distances entre le point à prédire et tous les points du jeu de données d'entraînement, ce qui peut devenir très coûteux en termes de temps et de mémoire, surtout avec de grands ensembles de données

2. Sensibilité à la dimensionnalité : L'algorithme souffre du "curse of dimensionality". À mesure que le nombre de dimensions augmente, les distances entre les points deviennent moins significatives, ce qui peut nuire à la performance du modèle

3. Prone à l'overfitting : En raison de sa nature non paramétrique et de sa dépendance au choix du nombre k, KNN peut facilement surajuster les données d'entraînement, surtout si k est trop petit. Un k trop grand peut également conduire à un sous-ajustement

4. Pas de modèle généralisé : Contrairement aux modèles paramétriques, KNN ne produit pas un modèle explicite pouvant être utilisé pour faire des prédictions sur de nouvelles données sans recalculer les distances

Conclusion

KNN est un algorithme puissant pour la régression, particulièrement adapté aux ensembles de données de petite à moyenne taille où les relations entre les variables sont complexes et non linéaires. Cependant, ses limitations en termes de coût computationnel et de sensibilité à la dimensionnalité doivent être prises en compte lors du choix d'une méthode pour un problème spécifique.

4 - Support Vector Regression

(SVR) est une technique de régression qui utilise les principes des machines à vecteurs de support (SVM) pour prédire des valeurs continues. Voici une explication détaillée de son fonctionnement, de ses caractéristiques et de ses avantages.

Principe de fonctionnement

SVR cherche à trouver une fonction qui prédit les valeurs cibles tout en maintenant une marge d'erreur acceptable. Contrairement aux méthodes de régression traditionnelles qui minimisent l'erreur globale, SVR se concentre sur le fait de s'assurer que la majorité des points de données se situent à l'intérieur d'une certaine tolérance, appelée epsilon (ε). Les points qui se trouvent en dehors de cette marge contribuent à la fonction de perte.

Équation de la fonction de perte

La fonction de perte dans SVR est définie comme suit :

- Les points à l'intérieur de la marge (ε) ne sont pas pénalisés.
- Les points en dehors de la marge sont pénalisés proportionnellement à leur distance par rapport à la marge.

Cela permet à SVR d'être moins sensible aux outliers, car il ne se concentre pas uniquement sur les erreurs globales.

Hyperparamètres

SVR a plusieurs hyperparamètres importants qui doivent être ajustés pour optimiser le modèle :

1. C : Ce paramètre régule la trade-off entre maximiser la marge et minimiser l'erreur d'entraînement. Un C élevé peut conduire à un modèle plus complexe, tandis qu'un C faible peut rendre le modèle plus simple.

2. Epsilon (ε) : Définit la largeur de la marge dans laquelle aucune pénalité n'est appliquée pour les erreurs. Cela influence directement le nombre de points considérés comme support vectors.

3. Type de noyau : SVR peut utiliser différents types de noyaux (linéaire, polynomial, RBF) pour capturer des relations non linéaires entre les variables d'entrée et les valeurs cibles.

Avantages de SVR

- Robustesse aux outliers : Grâce à sa structure basée sur des marges, SVR est moins affecté par les valeurs aberrantes par rapport aux méthodes traditionnelles.

- Modélisation des relations complexes : Avec l'utilisation de noyaux, SVR peut capturer des relations non linéaires complexes entre les caractéristiques et les cibles.

- Efficacité mémoire : SVR utilise uniquement un sous-ensemble des données d'entraînement (les support vectors) pour définir le modèle, ce qui le rend plus efficace en termes d'utilisation mémoire.

Inconvénients

- Sensibilité aux hyperparamètres : La performance du modèle dépend fortement du choix des hyperparamètres, ce qui nécessite souvent un ajustement minutieux.

- Coût computationnel élevé : L'entraînement d'un modèle SVR, surtout avec des noyaux complexes ou sur de grands ensembles de données, peut être intensif en termes de calcul.

- Difficulté avec les grands ensembles de données : SVR peut rencontrer des problèmes lorsqu'il est appliqué à des ensembles de données très volumineux ou lorsque le nombre de caractéristiques dépasse le nombre d'échantillons.

Conclusion

SVR est une méthode puissante pour la régression, capable de gérer des relations complexes et d'être robuste face aux outliers. Cependant, son efficacité dépend fortement du choix approprié des hyperparamètres et il peut devenir coûteux en termes computationnels pour des ensembles de données plus importants. En raison de ces caractéristiques, SVR est souvent utilisé dans des applications où une modélisation précise et robuste est requise.

5 – Modeles ensemblistes

Les modèles basés sur des arbres, tels que les arbres de décision, les forêts aléatoires et les algorithmes de boosting (comme XGBoost), ne nécessitent pas de mise à l'échelle des caractéristiques (feature scaling) pour plusieurs raisons fondamentales.

1. Nature des splits

Les arbres de décision fonctionnent en divisant les données en fonction des valeurs des caractéristiques. Lorsqu'un arbre effectue une division (split), il compare les valeurs d'une caractéristique à un seuil spécifique. Par exemple, un arbre pourrait diviser les données en deux groupes : ceux dont la valeur d'une caractéristique est inférieure à un certain seuil et ceux dont la valeur est supérieure. Cette opération est basée sur l'ordre des valeurs et non sur leur échelle absolue. Par conséquent, le fait que les valeurs soient mises à l'échelle ou non n'affecte pas la manière dont les données sont séparées.

2. Invariance aux transformations monotoniques

Les modèles basés sur des arbres sont invariants aux transformations monotoniques des caractéristiques. Cela signifie que si vous appliquez une transformation monotone (comme une normalisation ou une standardisation) à une caractéristique, cela n'affectera pas le résultat final des splits effectués par l'arbre. Les décisions prises par l'arbre restent inchangées, car il ne s'agit que de comparer les valeurs relatives entre elles, indépendamment de leur échelle absolue

3. Robustesse aux outliers

Les arbres de décision sont également robustes face aux valeurs aberrantes (outliers). Étant donné qu'ils se concentrent sur la séparation des données plutôt que sur la minimisation d'une fonction de coût globale (comme dans la régression linéaire), les valeurs extrêmes n'influencent pas significativement les splits, ce qui réduit encore la nécessité de mise à l'échelle

4. Impact minimal sur les performances

Bien que vous puissiez mettre à l'échelle les données avant d'entraîner un modèle basé sur un arbre, cela n'a généralement qu'un impact minime sur les performances du modèle. Dans certains cas, cela peut même entraîner des différences insignifiantes dans les prédictions en raison de la manière dont les ordinateurs gèrent les nombres

En conséquence, la mise à l'échelle peut être considérée comme une étape superflue pour ces types de modèles.

Conclusion

En résumé, les modèles basés sur des arbres ne nécessitent pas de mise à l'échelle des caractéristiques en raison de leur fonctionnement intrinsèque basé sur des comparaisons relatives et leur robustesse face aux variations d'échelle. Cela simplifie le processus de prétraitement des données pour ces modèles et permet aux praticiens de se concentrer sur d'autres aspects plus critiques du développement du modèle, tels que la qualité des caractéristiques et la taille du jeu de données.

5.1 - Random Forest Regression

est un algorithme d'apprentissage supervisé qui utilise une approche d'ensemble pour prédire des valeurs continues. Voici une explication détaillée de son fonctionnement, de ses avantages et de ses inconvénients.

Principe de fonctionnement

Le Random Forest Regression fonctionne en construisant plusieurs arbres de décision pendant l'entraînement et en combinant leurs prédictions pour obtenir une estimation finale. Voici les étapes clés :

1. Construction des arbres : Un nombre donné d'arbres de décision est créé à partir de sous-échantillons aléatoires avec remise du jeu de données d'entraînement (méthode de bootstrap). Chaque arbre est construit en utilisant un sous-ensemble aléatoire des caractéristiques disponibles à chaque nœud, ce qui aide à réduire la corrélation entre les arbres.

2. Prédiction : Chaque arbre de décision fait sa propre prédiction pour une entrée donnée. Pour obtenir la prédiction finale du modèle Random Forest, on calcule la moyenne des prédictions de tous les arbres. Cela permet d'atténuer les erreurs individuelles et d'améliorer la précision globale du modèle

Avantages du Random Forest Regression

- Robustesse aux outliers : L'algorithme est moins sensible aux valeurs aberrantes grâce à sa structure d'ensemble, qui réduit l'impact des points extrêmes sur la prédiction finale

- Capacité à gérer des relations non linéaires : Random Forest peut modéliser efficacement des relations complexes entre les variables indépendantes et dépendantes sans nécessiter de transformation préalable des données

- Pas besoin de mise à l'échelle des données : Contrairement à d'autres algorithmes, Random Forest ne nécessite pas que les caractéristiques soient normalisées ou standardisées, car il s'appuie sur des règles basées sur des seuils plutôt que sur des calculs de distance

- Performance élevée : En combinant plusieurs arbres, Random Forest tend à offrir une meilleure précision et à équilibrer le compromis biais-variance, ce qui le rend efficace dans divers scénarios

Inconvénients du Random Forest Regression

- Complexité et interprétabilité : Les modèles Random Forest peuvent être difficiles à interpréter en raison de leur nature "boîte noire". Bien qu'il soit possible d'évaluer l'importance des caractéristiques, comprendre le raisonnement derrière chaque prédiction peut être complexe

- Coût computationnel : La création et l'évaluation d'un grand nombre d'arbres peuvent être coûteuses en termes de mémoire et de temps, surtout avec de grands ensembles de données ou un nombre élevé d'arbres

- Problèmes d'extrapolation : Les prédictions sont limitées aux valeurs observées dans le jeu d'entraînement. Par conséquent, le modèle ne peut pas extrapoler au-delà de ces valeurs, ce qui peut poser problème si les nouvelles données se situent en dehors de l'intervalle observé

Conclusion

Random Forest Regression est un outil puissant pour la prédiction de valeurs continues grâce à sa robustesse, sa capacité à gérer des relations complexes et son efficacité globale. Cependant, ses limitations en matière d'interprétabilité et son coût computationnel doivent être pris en compte lors du choix d'un modèle pour un problème spécifique.

5.2 - Gradient Boosting Regression

est une technique d'apprentissage supervisé qui utilise le principe de boosting pour prédire des valeurs continues. Cette méthode combine plusieurs modèles faibles (généralement des arbres de décision) pour créer un modèle robuste et précis. Voici une explication détaillée de son fonctionnement, de ses avantages et de ses inconvénients.

Principe de fonctionnement

1. Boosting

Le boosting est une approche qui consiste à entraîner des modèles faibles de manière séquentielle. Chaque modèle est formé pour corriger les erreurs du modèle précédent. Dans le cas de la régression, l'objectif est de minimiser la fonction de perte, souvent la moyenne des erreurs quadratiques (MSE).

2. Construction du modèle

Le processus de Gradient Boosting se déroule en plusieurs étapes :

- Initialisation : On commence par faire une prédiction initiale, qui est généralement la moyenne des valeurs cibles du jeu de données.

- Calcul des résidus : Pour chaque itération, on calcule les résidus, c'est-à-dire les erreurs entre les prédictions actuelles et les valeurs réelles.

- Ajout d'un nouvel arbre : Un nouvel arbre de décision est construit pour prédire ces résidus. Cet arbre vise à corriger les erreurs du modèle précédent.

- Mise à jour des prédictions : Les prédictions sont mises à jour en ajoutant la contribution du nouvel arbre, pondérée par un taux d'apprentissage (learning rate). Ce taux détermine l'importance de chaque nouvel arbre dans la prédiction finale.

- Répétition : Ce processus est répété jusqu'à ce qu'un certain nombre d'arbres soit atteint ou que l'ajout d'arbres n'améliore plus significativement le modèle.

3. Fonctionnement itératif

Chaque nouvel arbre est construit pour s'attaquer aux erreurs laissées par les arbres précédents, ce qui permet au modèle d'apprendre progressivement et d'améliorer ses performances sur les données d'entraînement.

Avantages du Gradient Boosting Regression

- Précision élevée : Gradient Boosting peut souvent atteindre une précision supérieure à celle des autres méthodes comme les forêts aléatoires, surtout sur des données complexes et non linéaires
- Flexibilité : Il peut être appliqué à différents types de problèmes, y compris ceux avec des caractéristiques manquantes ou des valeurs aberrantes
- Capacité d'adaptation : Le modèle peut être ajusté pour utiliser différentes fonctions de perte, ce qui le rend adaptable à divers types de données et objectifs
- Moins sujet au surajustement : Grâce à des techniques comme l'arrêt précoce et la régularisation, Gradient Boosting peut réduire le risque de surajustement par rapport à d'autres méthodes

Inconvénients

- Coût computationnel élevé : L'entraînement peut être plus lent que d'autres méthodes en raison de la nature séquentielle du processus
- Sensibilité aux hyperparamètres : Le choix du taux d'apprentissage et du nombre d'arbres peut grandement influencer les performances du modèle, nécessitant un réglage minutieux
- Risque de surajustement : Si le modèle n'est pas correctement régularisé, il peut surajuster les données d'entraînement, surtout dans le cas de données bruyantes

Conclusion

Gradient Boosting Regression est une méthode puissante pour construire des modèles prédictifs efficaces en combinant plusieurs arbres faibles. Bien qu'elle offre une grande précision et flexibilité, elle nécessite une attention particulière lors du réglage des hyperparamètres et peut être coûteuse en termes de calcul. C'est un choix populaire dans divers domaines tels que la finance, la santé et le traitement du langage naturel en raison de sa capacité à gérer des relations complexes dans les données.

6 - La bibliothèque SHAP (SHapley Additive exPlanations)

est un outil puissant en Python utilisé pour expliquer les prédictions des modèles d'apprentissage automatique. Elle repose sur des concepts de la théorie des jeux, en particulier les valeurs de Shapley, pour attribuer une importance à chaque caractéristique dans le processus de décision du modèle. Voici un aperçu détaillé de ses fonctionnalités et de son utilisation.

Principes de base des valeurs SHAP

1. Explication des prédictions : Les valeurs SHAP permettent d'expliquer comment chaque caractéristique contribue à la prédiction d'un modèle. Chaque valeur SHAP représente l'impact d'une caractéristique sur la sortie du modèle, qu'elle soit positive ou négative.

2. Additivité : Les valeurs SHAP sont additives, ce qui signifie que la somme des contributions de chaque caractéristique égale la différence entre la sortie attendue du modèle et la sortie réelle pour une instance donnée.

3. Interprétabilité : En décomposant les prédictions en contributions individuelles, SHAP offre une interprétation claire et intuitive des décisions du modèle, facilitant ainsi la communication des résultats aux parties prenantes.

Caractéristiques de la bibliothèque SHAP

- Modèle-agnostique : SHAP peut être utilisé avec n'importe quel modèle d'apprentissage automatique, y compris les régressions linéaires, les arbres de décision, les forêts aléatoires et les réseaux de neurones.

- Robustesse aux données manquantes : Les valeurs SHAP sont conçues pour être robustes face aux caractéristiques manquantes, garantissant que celles-ci n'affectent pas l'interprétation.

- Visualisations : La bibliothèque fournit plusieurs outils de visualisation pour aider à comprendre l'importance des caractéristiques et leurs interactions, comme les graphiques de dépendance et les graphiques en barres.

Interprétation des résultats

- Valeurs SHAP : Chaque valeur dans shap_values indique combien chaque caractéristique contribue à la prédiction par rapport à la moyenne globale.

- Visualisations : Les graphiques générés par shap.summary_plot montrent l'importance relative des caractéristiques ainsi que leur impact sur les prédictions.

Conclusion

La bibliothèque SHAP est un outil essentiel pour améliorer l'interprétabilité des modèles d'apprentissage automatique. En fournissant une méthode cohérente et objective pour évaluer l'impact des caractéristiques sur les prédictions, elle permet aux data scientists et aux analystes de mieux comprendre le comportement des modèles complexes et d'assurer une communication claire avec les parties prenantes.

6.1 - Le SHAP summary plot

est un outil visuel puissant utilisé pour interpréter les valeurs SHAP dans les modèles d'apprentissage automatique. Voici comment interpréter ce graphique :

Structure du SHAP Summary Plot

1. Axes du graphique

- Axe Y : Représente les caractéristiques (features) du modèle, classées par ordre d'importance de haut en bas. Les caractéristiques les plus influentes apparaissent en haut.

- Axe X : Indique les valeurs SHAP, qui mesurent l'impact de chaque caractéristique sur la prédiction du modèle. Les valeurs peuvent être positives ou négatives.

2. Points individuels

- Chaque point sur le graphique représente une instance (ou un échantillon) du jeu de données. La position d'un point sur l'axe X indique l'effet de la caractéristique correspondante sur la prédiction.

- Les points à droite (valeurs SHAP positives) indiquent que la caractéristique a un effet positif sur la prédiction, tandis que ceux à gauche (valeurs SHAP négatives) indiquent un effet négatif.

3. Couleurs des points

- La couleur des points est souvent utilisée pour représenter la valeur de la caractéristique elle-même. Par exemple :

- Rouge peut indiquer des valeurs élevées de la caractéristique.

- Bleu peut indiquer des valeurs faibles.

- Cela permet d'observer comment les différentes valeurs d'une caractéristique influencent la prédiction.

Interprétation des résultats

Importance des caractéristiques

- Les caractéristiques sont classées par leur importance dans le modèle. Plus une caractéristique est en haut du graphique, plus elle a un impact significatif sur les prédictions.

Impact directionnel

- En examinant la répartition des points pour chaque caractéristique, vous pouvez comprendre comment elle influence les prédictions :

- Si la majorité des points d'une caractéristique se trouvent à droite, cela signifie que cette caractéristique a tendance à augmenter les prédictions.

- Si beaucoup de points se trouvent à gauche, cela indique un effet réducteur sur les prédictions.

Exemples concrets

- Par exemple, si une caractéristique comme "Âge" est en haut du graphique et que ses points sont principalement rouges, cela suggère que les âges plus élevés augmentent la probabilité d'une certaine prédiction (comme survivre dans le cas du Titanic). À l'inverse, si une autre caractéristique comme "Tarif" montre principalement des points bleus à gauche, cela pourrait indiquer que des tarifs plus bas sont associés à une probabilité réduite de survie.

Conclusion

Le SHAP summary plot offre une vue d'ensemble précieuse sur l'importance et l'impact des caractéristiques dans un modèle d'apprentissage automatique. En utilisant ce graphique, vous pouvez non seulement identifier quelles caractéristiques sont les plus influentes, mais

aussi comprendre comment leurs valeurs spécifiques affectent les prédictions. Cela contribue à rendre les modèles plus transparents et interprétables, facilitant ainsi la communication des résultats aux parties prenantes.

6.2 - Le SHAP decision plot

est un outil visuel qui permet d'interpréter les décisions d'un modèle d'apprentissage automatique en utilisant les valeurs SHAP. Voici comment interpréter ce type de graphique :

Structure du SHAP Decision Plot

1. Axes du graphique

- Axe Y : Représente les caractéristiques (features) du modèle, généralement classées par ordre d'importance. Les caractéristiques les plus influentes sont affichées en haut.
- Axe X : Indique la sortie du modèle, qui peut être soit une valeur continue, soit une probabilité (selon le lien utilisé, par exemple, logit pour les probabilités).

2. Lignes individuelles

- Chaque ligne sur le graphique représente une instance ou une observation spécifique. La position de cette ligne sur l'axe X montre comment les valeurs des caractéristiques ont contribué à la prédiction finale.
- Les lignes commencent à un point de base, qui est souvent la valeur attendue (ou moyenne) du modèle, et se déplacent vers la droite ou la gauche selon que les caractéristiques ont un effet positif ou négatif sur la prédiction.

3. Couleurs des lignes

- La couleur des lignes peut être utilisée pour représenter les valeurs des caractéristiques elles-mêmes, permettant ainsi de visualiser comment différentes valeurs influencent la prédiction.

Interprétation des résultats

Cumul des effets

- En montant le long de la ligne, vous pouvez voir comment chaque caractéristique contribue au score final de la prédiction. Cela fournit une représentation cumulative des effets des caractéristiques.
- Le point où chaque ligne croise l'axe X représente la prédiction finale pour cette observation.

Impact directionnel

- Si une ligne se déplace vers la droite, cela indique que les caractéristiques ont un effet positif sur la prédiction. Inversement, si elle se déplace vers la gauche, cela signifie un effet négatif.
- Cela permet d'identifier facilement quelles caractéristiques ont le plus grand impact sur les décisions du modèle.

Comparaison entre observations

- En traçant plusieurs observations sur le même graphique, il est possible de comparer comment différentes instances sont influencées par les mêmes caractéristiques.
- Cela est particulièrement utile pour identifier des modèles ou des comportements similaires entre différentes prédictions.

Cas d'utilisation

Le SHAP decision plot est particulièrement utile dans plusieurs scénarios :

- Visualisation des effets cumulés : Pour comprendre comment plusieurs caractéristiques interagissent pour influencer une prédiction.
- Exploration de scénarios hypothétiques : Pour examiner comment différents ensembles de valeurs de caractéristiques pourraient affecter les prédictions.
- Comparaison entre modèles : Pour évaluer et comparer les décisions prises par différents modèles ou différentes configurations d'un même modèle.

Conclusion

Le SHAP decision plot est un outil essentiel pour déchiffrer les décisions prises par des modèles complexes. En fournissant une vue claire et cumulative de l'impact des caractéristiques sur les prédictions, il aide à rendre les modèles plus transparents et interprétables, facilitant ainsi leur utilisation dans des contextes critiques où l'interprétabilité est primordiale.

6.3 - Le SHAP summary plot

avec des points (dot plot) est un outil visuel qui permet d'interpréter les valeurs SHAP pour un modèle d'apprentissage automatique. Voici comment interpréter ce graphique :

Structure du SHAP Summary Plot avec Points

1. Axes du graphique

- Axe Y : Représente les caractéristiques (features) du modèle, classées par ordre d'importance. Les caractéristiques les plus influentes sont affichées en haut.
- Axe X : Indique les valeurs SHAP, qui mesurent l'impact de chaque caractéristique sur la prédiction du modèle. Les valeurs peuvent être positives ou négatives.

2. Points individuels

- Chaque point sur le graphique représente une instance (ou un échantillon) du jeu de données. La position d'un point sur l'axe X indique l'effet de la caractéristique correspondante sur la prédiction.

- Les points à droite (valeurs SHAP positives) indiquent que la caractéristique a un effet positif sur la prédiction, tandis que ceux à gauche (valeurs SHAP négatives) indiquent un effet négatif.

3. Couleurs des points

- La couleur des points est utilisée pour représenter la valeur de la caractéristique elle-même :

- Rouge : indique une valeur élevée de la caractéristique.

- Bleu : indique une valeur faible.

- Cela permet d'observer comment les différentes valeurs d'une caractéristique influencent la prédiction.

Interprétation des résultats

Importance des caractéristiques

- Les caractéristiques sont classées par leur importance dans le modèle. Plus une caractéristique est en haut du graphique, plus elle a un impact significatif sur les prédictions.

Impact directionnel

- En examinant la répartition des points pour chaque caractéristique, vous pouvez comprendre comment elle influence les prédictions :

- Si la majorité des points d'une caractéristique se trouvent à droite, cela signifie que cette caractéristique a tendance à augmenter les prédictions.

- Si beaucoup de points se trouvent à gauche, cela indique un effet réducteur sur les prédictions.

Distribution des valeurs

- La distribution des points peut également fournir des informations précieuses :

- Une concentration de points dans une certaine zone peut indiquer que cette plage de valeurs a un impact similaire sur les prédictions.

- Des points éparpillés peuvent suggérer une variabilité dans l'effet de cette caractéristique.

Exemples concrets

Par exemple, dans un modèle prédisant la survie lors du naufrage du Titanic :

- Si "Sex" est en haut du graphique et que ses points sont principalement rouges, cela suggère que les passagers de sexe féminin ont une probabilité plus élevée de survie.

- Si "Fare" montre une majorité de points bleus à gauche, cela pourrait indiquer que des tarifs plus bas sont associés à une probabilité réduite de survie.

Conclusion

Le SHAP summary plot avec des points offre une vue d'ensemble précieuse sur l'importance et l'impact des caractéristiques dans un modèle d'apprentissage automatique. En utilisant ce graphique, vous pouvez non seulement identifier quelles caractéristiques sont les plus influentes, mais aussi comprendre comment leurs valeurs spécifiques affectent les prédictions. Cela contribue à rendre les modèles plus transparents et interprétables, facilitant ainsi la communication des résultats aux parties prenantes.

6.4 - Le SHAP beeswarm

plot est un outil visuel puissant qui permet d'interpréter les valeurs SHAP pour un modèle d'apprentissage automatique. Voici comment interpréter ce graphique :

Structure du SHAP Beeswarm Plot

1. Axes du graphique

- Axe Y : Représente les caractéristiques (features) du modèle, classées par ordre d'importance. Les caractéristiques les plus influentes sont affichées en haut.
- Axe X : Indique les valeurs SHAP, qui mesurent l'impact de chaque caractéristique sur la prédiction du modèle. Les valeurs peuvent être positives ou négatives.

2. Points individuels

- Chaque point sur le graphique représente une instance (ou un échantillon) du jeu de données. La position d'un point sur l'axe X indique l'effet de la caractéristique correspondante sur la prédiction.
- Les points à droite (valeurs SHAP positives) indiquent que la caractéristique a un effet positif sur la prédiction, tandis que ceux à gauche (valeurs SHAP négatives) indiquent un effet négatif.

3. Couleurs des points

- La couleur des points est utilisée pour représenter la valeur de la caractéristique elle-même :
- Rouge : indique une valeur élevée de la caractéristique.
- Bleu : indique une valeur faible.
- Cela permet d'observer comment les différentes valeurs d'une caractéristique influencent la prédiction et d'identifier des tendances.

Interprétation des résultats

Importance des caractéristiques

- Les caractéristiques sont classées par leur importance dans le modèle. Plus une caractéristique est en haut du graphique, plus elle a un impact significatif sur les prédictions.

Impact directionnel

- En examinant la répartition des points pour chaque caractéristique, vous pouvez comprendre comment elle influence les prédictions :

- Si la majorité des points d'une caractéristique se trouvent à droite, cela signifie que cette caractéristique a tendance à augmenter les prédictions.

- Si beaucoup de points se trouvent à gauche, cela indique un effet réducteur sur les prédictions.

Distribution des valeurs

- La distribution des points peut également fournir des informations précieuses :

- Une concentration de points dans une certaine zone peut indiquer que cette plage de valeurs a un impact similaire sur les prédictions.

- Des points éparpillés peuvent suggérer une variabilité dans l'effet de cette caractéristique.

Exemples concrets

Par exemple, si vous analysez un modèle prédisant le revenu :

- Si "Âge" est en haut du graphique et que ses points sont principalement rouges, cela pourrait indiquer que les personnes plus âgées ont tendance à avoir un revenu plus élevé.

- Si "Niveau d'éducation" montre une majorité de points bleus à gauche, cela pourrait suggérer que des niveaux d'éducation plus bas sont associés à des revenus plus faibles.

Conclusion

Le SHAP beeswarm plot offre une vue d'ensemble précieuse sur l'importance et l'impact des caractéristiques dans un modèle d'apprentissage automatique. En utilisant ce graphique, vous pouvez non seulement identifier quelles caractéristiques sont les plus influentes, mais aussi comprendre comment leurs valeurs spécifiques affectent les prédictions. Cela contribue à rendre les modèles plus transparents et interprétables, facilitant ainsi la communication des résultats aux parties prenantes.

6.5 - Le SHAP force plot

est un outil visuel qui illustre comment les caractéristiques d'une instance spécifique influencent la prédiction d'un modèle d'apprentissage automatique. Voici une explication détaillée de sa structure et de son interprétation.

Structure du SHAP Force Plot

1. Axes du graphique

- Axe Y : Représente les caractéristiques (features) du modèle, généralement affichées dans un ordre qui reflète leur impact sur la prédiction.

- Axe X : Indique la valeur cumulée des contributions des caractéristiques à la prédiction. Les contributions positives sont affichées à gauche, tandis que les contributions négatives sont affichées à droite.

2. Points et flèches

- Chaque caractéristique est représentée par une flèche ou un point qui s'étend vers la droite ou la gauche :

- Flèches vers la droite : Indiquent une contribution positive à la prédiction (augmentation de la valeur prédite).

- Flèches vers la gauche : Indiquent une contribution négative (diminution de la valeur prédite).

3. Valeur de prédiction

- Le point où toutes les contributions se rejoignent correspond à la prédiction finale pour l'observation donnée. Cela permet de visualiser comment chaque caractéristique a contribué à cette prédiction par rapport à une valeur de référence.

Interprétation des résultats

Impact des caractéristiques

- Le graphique montre clairement quelles caractéristiques ont le plus grand impact sur la prédiction :

- Si une flèche est longue et pointée vers la droite, cela signifie que cette caractéristique a un effet significatif et positif sur la prédiction.

- Inversement, une flèche longue pointée vers la gauche indique un impact significatif mais négatif.

Cumul des effets

- En ajoutant les contributions positives et négatives, vous pouvez voir comment elles s'équilibrent pour déterminer la prédiction finale. Cela aide à comprendre les interactions entre différentes caractéristiques.

Comparaison entre observations

- En traçant plusieurs observations sur le même graphique (dans le cas d'un graphique en force empilé), vous pouvez comparer comment différentes instances sont influencées par les mêmes caractéristiques. Cela est particulièrement utile pour identifier des modèles ou des comportements similaires entre différentes prédictions.

Exemples d'utilisation

Le SHAP force plot est utile dans divers scénarios :

- Analyse de cas individuels : Pour examiner comment les caractéristiques spécifiques d'une observation influencent sa prédiction.

- Identification des erreurs de classification : Pour comprendre pourquoi certaines instances ont été mal classées en examinant les contributions des caractéristiques.

- Communication des résultats : Pour expliquer les décisions du modèle aux parties prenantes de manière claire et intuitive.

Conclusion

Le SHAP force plot est un outil essentiel pour déchiffrer les décisions prises par des modèles complexes. En fournissant une vue claire et cumulative de l'impact des caractéristiques sur les prédictions, il aide à rendre les modèles plus transparents et interprétables, facilitant ainsi leur utilisation dans des contextes critiques où l'interprétabilité est primordiale.

6.6 – Shap waterfall plot

Le SHAP waterfall plot et le SHAP force plot sont deux outils visuels qui aident à interpréter les valeurs SHAP d'un modèle d'apprentissage automatique, mais ils présentent des différences dans leur présentation et leur utilisation. Voici une explication de chacun et comment ils se comparent.

SHAP Waterfall Plot

Structure et Fonctionnement

Orientation : Le waterfall plot est organisé verticalement. Il commence par la valeur de base (ou valeur attendue) du modèle, puis montre comment chaque caractéristique contribue à la prédiction finale en ajoutant ou en soustrayant des valeurs.

Contributions : Chaque barre dans le graphique représente l'impact d'une caractéristique sur la prédiction. Les contributions positives (qui augmentent la prédiction) sont généralement affichées en rouge, tandis que les contributions négatives (qui diminuent la prédiction) sont affichées en bleu.

Visualisation cumulative : Le graphique illustre comment les contributions cumulées des caractéristiques mènent à la prédiction finale. La somme des valeurs SHAP de toutes les caractéristiques montre le déplacement de la valeur de base vers la prédiction.

Interprétation

Le waterfall plot permet de visualiser facilement l'effet individuel de chaque caractéristique sur la décision du modèle, ce qui aide à comprendre comment chaque facteur influence le résultat final pour une instance donnée.

SHAP Force Plot

Structure et Fonctionnement

Orientation : Le force plot est organisé horizontalement. Les contributions positives sont affichées à gauche, tandis que les contributions négatives sont affichées à droite.

Visualisation additive : Chaque flèche ou barre dans le graphique représente une contribution d'une caractéristique, avec des longueurs proportionnelles à l'importance de cette contribution. L'effet net est montré par la position finale sur l'axe X, qui indique la prédiction finale.

Comparaison des contributions : Le force plot met en évidence comment les caractéristiques "s'opposent" les unes aux autres pour influencer la prédiction, ce qui peut être utile pour visualiser les interactions entre les caractéristiques.

Interprétation

Le force plot est particulièrement utile pour voir rapidement quelles caractéristiques ont un impact positif ou négatif sur une prédiction spécifique, offrant une perspective intuitive sur les décisions du modèle.

Comparaison entre Waterfall Plot et Force Plot

Caractéristique	Waterfall Plot	Force Plot
Orientation	Vertical (de bas en haut)	Horizontal (de gauche à droite)
Affichage des contributions	Barres représentant l'impact cumulatif	Flèches montrant les contributions individuelles
Visualisation cumulative	Montre comment chaque contribution s'ajoute à la valeur de base	Montre comment les contributions s'opposent
Utilisation typique	Expliquer l'impact détaillé d'une seule instance	Comparer rapidement les impacts des caractéristiques

Conclusion

Le SHAP waterfall plot et le SHAP force plot sont deux outils complémentaires pour interpréter les valeurs SHAP dans les modèles d'apprentissage automatique. Le waterfall plot est idéal pour visualiser l'impact cumulatif des caractéristiques sur une prédiction spécifique, tandis que le force plot offre une vue intuitive des contributions individuelles et de leurs effets opposés. Ensemble, ces graphiques améliorent la compréhension des décisions du modèle et facilitent l'interprétation des résultats pour les utilisateurs.

7 – R2 et MAE

7.1 - Intro

Le R^2 (coefficient de détermination) et le Mean Absolute Error (MAE) sont deux métriques couramment utilisées pour évaluer la performance des modèles de régression. Voici comment ces métriques s'appliquent et leur pertinence pour les modèles discutés précédemment.

R^2 (Coefficient de Détermination)

Pertinence

- R^2 mesure la proportion de la variance des données qui est expliquée par le modèle. Il est particulièrement utile pour évaluer des modèles linéaires, car il donne une idée de la qualité de l'ajustement du modèle aux données.

- Utilisation : R^2 est souvent utilisé dans les contextes où l'on souhaite comprendre le pouvoir explicatif d'un ensemble de variables indépendantes sur une variable dépendante.

Limitations

- R^2 peut être trompeur, surtout si le modèle est complexe ou si le nombre de prédicteurs augmente sans amélioration significative du modèle, ce qui peut conduire à un surajustement. C'est pourquoi l'Adjusted R^2 est parfois préféré, car il pénalise l'ajout de variables non pertinentes.

MAE (Erreur Absolue Moyenne)

Pertinence

- MAE mesure l'erreur moyenne entre les valeurs prédites et les valeurs réelles, en traitant toutes les erreurs de manière égale. Cela en fait une métrique robuste, surtout en présence d'outliers.

- Utilisation : MAE est souvent utilisé lorsque l'on souhaite avoir une mesure claire et interprétable de la précision des prédictions d'un modèle, sans accorder trop d'importance aux grandes erreurs.

Limitations

- Bien que MAE soit utile pour évaluer la performance générale des modèles, il ne fournit pas d'informations sur la proportion de variance expliquée comme le fait R^2. De plus, il peut être moins sensible aux erreurs importantes par rapport à des métriques comme le MSE ou le RMSE.

Application aux Modèles Discutés

Modèles Linéaires et Non Linéaires

- Pour les modèles linéaires (comme la régression linéaire), R^2 est particulièrement pertinent car il donne une bonne indication de l'ajustement du modèle.

- Pour les modèles non linéaires ou complexes (comme les forêts aléatoires ou les réseaux de neurones), R^2 peut être moins informatif. Dans ces cas, MAE peut fournir une évaluation plus directe des performances prédictives.

Ensemble d'Entraînement vs. Ensemble de Test

- Les valeurs de R^2 et MAE peuvent différer entre l'ensemble d'entraînement et l'ensemble de test. Un modèle peut avoir un bon R^2 sur l'ensemble d'entraînement mais un MAE élevé sur l'ensemble de test, indiquant un surajustement.

Conclusion

R^2 et MAE sont tous deux pertinents pour évaluer les modèles discutés, mais leur utilisation dépend du contexte et des objectifs spécifiques :

- R² est utile pour comprendre la proportion de variance expliquée, surtout dans les modèles linéaires.

- MAE offre une mesure intuitive et robuste des erreurs prédictives, particulièrement bénéfique lorsque les outliers sont présents ou lorsque toutes les erreurs doivent être traitées équitablement.

En résumé, il est souvent judicieux d'utiliser plusieurs métriques d'évaluation pour obtenir une vue complète des performances du modèle.

7.2 - R², ou coefficient de détermination

est une mesure statistique qui indique la proportion de la variance d'une variable dépendante qui peut être expliquée par une ou plusieurs variables indépendantes dans un modèle de régression. Voici une explication détaillée de ce qu'est R² et comment son interprétation peut varier selon qu'il est calculé sur l'ensemble d'entraînement ou l'ensemble de test.

Qu'est-ce que R² ?

Définition

R² est défini comme suit :

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$

où :

- RSS (Residual Sum of Squares) est la somme des carrés des résidus, c'est-à-dire la somme des carrés des différences entre les valeurs observées et les valeurs prédites.

- TSS (Total Sum of Squares) est la somme des carrés des différences entre les valeurs observées et la moyenne des valeurs observées.

Interprétation

- Valeurs : R² varie entre 0 et 1. Un R² de 0 signifie que le modèle n'explique aucune variance des données, tandis qu'un R² de 1 indique que le modèle explique toute la variance.

- Signification : Par exemple, un R² de 0,75 signifie que 75 % de la variance de la variable dépendante peut être expliquée par le modèle, ce qui indique un bon ajustement.

Interprétation selon l'ensemble de données

Ensemble d'entraînement vs. Ensemble de test

L'interprétation de R² peut différer selon qu'il est mesuré sur l'ensemble d'entraînement ou l'ensemble de test :

1. Ensemble d'entraînement :

- Lorsque R² est calculé sur l'ensemble d'entraînement, il a tendance à être plus élevé. Cela est dû au fait que le modèle a été ajusté spécifiquement à ces données, ce qui peut conduire à un surajustement (overfitting). Un modèle trop complexe peut expliquer presque toute la variance dans les données d'entraînement, mais cela ne garantit pas qu'il généralise bien à de nouvelles données.

2. Ensemble de test :

- Lorsque R² est calculé sur l'ensemble de test, qui n'a pas été utilisé pour entraîner le modèle, il peut être significativement plus bas. Cela reflète la capacité du modèle à généraliser ses prédictions sur des données non vues. Si le R² sur l'ensemble de test est faible comparé à celui sur l'ensemble d'entraînement, cela peut indiquer que le modèle a surajusté les données d'entraînement et ne capture pas bien les tendances sous-jacentes dans les données.

Exemples pratiques

Des études ont montré que des modèles peuvent avoir un R² très élevé sur l'ensemble d'entraînement (ex. : 0,90) mais un R² beaucoup plus faible sur l'ensemble de test (ex. : 0,30), ce qui souligne un problème potentiel d'overfitting

Conclusion

R² est une mesure utile pour évaluer la qualité d'un modèle de régression, mais son interprétation doit être faite avec prudence. Il est crucial de comparer les valeurs obtenues sur les ensembles d'entraînement et de test pour évaluer la capacité du modèle à généraliser. Un bon ajustement du modèle doit se traduire par des valeurs R² similaires sur les deux ensembles ; sinon, cela pourrait indiquer un problème d'overfitting où le modèle s'adapte trop étroitement aux données d'entraînement au détriment de sa performance sur des données nouvelles.

7.3 - Le Mean Absolute Error (MAE), ou erreur absolue moyenne

est une métrique utilisée pour évaluer la précision des prédictions d'un modèle de régression. Voici une explication détaillée de ce qu'est le MAE et comment il s'interprète.

Définition du MAE

Le MAE mesure la moyenne des erreurs absolues entre les valeurs prédites par le modèle et les valeurs réelles observées. Il est calculé à l'aide de la formule suivante :

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

où :

- nn est le nombre d'observations,

- yiyi est la valeur réelle,
- y^iy^i est la valeur prédite.

Interprétation du MAE

1. Magnitude des erreurs

Le MAE fournit une mesure directe de la magnitude moyenne des erreurs dans les prédictions. Par exemple, si le MAE est de 5, cela signifie qu'en moyenne, les prédictions du modèle sont à 5 unités des valeurs réelles. Cela donne une idée claire de l'exactitude des prédictions.

2. Robustesse aux outliers

Le MAE est moins sensible aux valeurs aberrantes (outliers) par rapport à d'autres métriques comme le Mean Squared Error (MSE). Cela est dû au fait qu'il ne pénalise pas les erreurs plus élevées de manière disproportionnée, car il utilise la valeur absolue des erreurs plutôt que de les élever au carré. Cela en fait un choix approprié lorsque les données contiennent des outliers ou lorsque l'on souhaite une évaluation plus équilibrée des erreurs.

3. Interprétabilité

Le MAE est exprimé dans les mêmes unités que les données d'origine, ce qui facilite son interprétation. Par exemple, dans un modèle prédisant des températures, un MAE de 2 degrés indique que, en moyenne, les prévisions sont à 2 degrés des valeurs réelles.

Comparaison avec d'autres métriques

- MSE (Mean Squared Error) : Contrairement au MAE, le MSE pénalise plus fortement les grandes erreurs en les élevant au carré. Cela peut être utile dans certains contextes où il est important de réduire les grandes erreurs, mais cela peut également rendre le MSE plus sensible aux outliers.
- RMSE (Root Mean Square Error) : Le RMSE est la racine carrée du MSE et a également tendance à accentuer l'impact des grandes erreurs tout en restant dans les mêmes unités que les données d'origine.

Conclusion

Le MAE est une mesure simple et efficace pour évaluer la précision des modèles de régression. Il fournit une indication claire de l'erreur moyenne dans les prédictions et est particulièrement utile lorsque l'on souhaite minimiser l'impact des valeurs aberrantes. En raison de sa robustesse et de sa facilité d'interprétation, le MAE est largement utilisé dans divers domaines tels que la prévision, la modélisation financière et l'analyse des séries temporelles.

8 - Les transformations Box-Cox et Yeo-Johnson

sont deux méthodes utilisées pour transformer des données afin de les rendre plus conformes à une distribution normale, ce qui est souvent une hypothèse clé dans de nombreuses analyses statistiques. Voici une explication détaillée des deux transformations, de leurs différences et de leurs applications.

Box-Cox Transformation

Définition

La transformation Box-Cox est une méthode paramétrique qui transforme des variables dépendantes non normales en une forme plus normale. Elle est définie par la formule suivante :

$$Y' = \begin{cases} \frac{Y^\lambda - 1}{\lambda} & \text{si } \lambda \neq 0 \\ \log(Y) & \text{si } \lambda = 0 \end{cases}$$

où :

- Y est la variable d'origine,
- Y' est la variable transformée,
- λ est un paramètre qui détermine la nature de la transformation.

Conditions

- Données positives : La transformation Box-Cox nécessite que toutes les valeurs de Y soient strictement positives. Si des valeurs nulles ou négatives sont présentes, la transformation ne peut pas être appliquée directement.

Applications

- La transformation Box-Cox est souvent utilisée pour stabiliser la variance et rendre les résidus d'un modèle de régression plus normaux, ce qui permet d'améliorer la validité des tests statistiques.

Yeo-Johnson Transformation

Définition

La transformation Yeo-Johnson est une généralisation de la transformation Box-Cox qui peut gérer à la fois des valeurs positives et négatives. Elle est définie par :

$$Y' = \begin{cases} \frac{(Y+1)^\lambda - 1}{\lambda} & \text{si } Y \geq 0 \\ -\frac{(-Y+1)^{2-\lambda} - 1}{2-\lambda} & \text{si } Y < 0 \end{cases}$$

Conditions

- Données positives et négatives : Contrairement à Box-Cox, la transformation Yeo-Johnson peut être appliquée à des données contenant des valeurs nulles ou négatives, ce qui en fait une option plus flexible pour divers ensembles de données.

Applications

- Comme pour Box-Cox, la transformation Yeo-Johnson est utilisée pour rendre les données plus normales et stabiliser la variance. Elle est particulièrement utile dans les cas où les données contiennent des valeurs négatives, ce qui rendrait l'utilisation de Box-Cox impossible.

Comparaison entre Box-Cox et Yeo-Johnson

Caractéristique	Box-Cox Transformation	Yeo-Johnson Transformation
Données acceptées	Strictement positives	Positives, nulles et négatives
Formule	Dépend du paramètre λ	Différente pour $Y \geq 0$ et $Y < 0$
Flexibilité	Moins flexible (pas de valeurs négatives)	Plus flexible (valeurs variées acceptées)
Utilisation typique	Stabilisation de variance	Stabilisation de variance et normalisation
Paramètre λ	Un seul paramètre	Deux paramètres effectifs (pour chaque cas)

Conclusion

Les transformations Box-Cox et Yeo-Johnson sont des outils puissants pour traiter les problèmes de non-normalité dans les données. Le choix entre elles dépend principalement de la nature des données à transformer. Si toutes les valeurs sont positives, Box-Cox peut être utilisé ; sinon, Yeo-Johnson est préférable en raison de sa capacité à gérer les valeurs nulles et négatives. Les deux transformations visent à améliorer l'ajustement des modèles statistiques en rendant les résidus plus conformes à une distribution normale.

9 - Stratified K-Fold Cross-Validation

est une technique d'évaluation des modèles en apprentissage automatique qui améliore la méthode de K-Fold Cross-Validation standard, en veillant à ce que chaque pli (fold) contienne une distribution équilibrée des classes cibles. Voici une explication détaillée de son fonctionnement et de ses applications.

Fonctionnement de Stratified K-Fold

1. Division des données

- Sélection du nombre de plis (k) : Tout d'abord, vous choisissez le nombre de plis (k) que vous souhaitez utiliser pour la validation croisée.

- Stratification : Contrairement à la K-Fold classique, où les données sont divisées de manière aléatoire, Stratified K-Fold s'assure que chaque pli contient environ la même proportion d'échantillons de chaque classe que l'ensemble complet. Cela est particulièrement important lorsque les classes sont déséquilibrées.

2. Processus d'entraînement et de validation

- Pour chaque itération :

- Un pli est utilisé comme ensemble de test, tandis que les k-1 autres plis sont combinés pour former l'ensemble d'entraînement.

- Le modèle est entraîné sur l'ensemble d'entraînement et validé sur le pli de test.

- Ce processus est répété k fois, chaque pli étant utilisé une fois comme ensemble de test.

3. Calcul des résultats

- À la fin des k itérations, les résultats des validations sont agrégés (par exemple, en calculant la moyenne des scores) pour obtenir une évaluation globale du modèle.

Utilisations et Avantages

1. Gestion des déséquilibres de classe

Stratified K-Fold est particulièrement utile dans les cas où il y a un déséquilibre significatif entre les classes cibles. Par exemple, dans un ensemble de données où 90 % des échantillons appartiennent à une classe et seulement 10 % à une autre, utiliser K-Fold standard pourrait conduire à certains plis ne contenant que des exemples d'une seule classe. Cela fausse l'évaluation du modèle. Stratified K-Fold garantit que chaque pli reflète la distribution réelle des classes.

2. Amélioration de la robustesse du modèle

En maintenant une distribution équilibrée des classes dans chaque pli, Stratified K-Fold permet d'obtenir des estimations plus fiables de la performance du modèle. Cela réduit le risque d'overfitting et assure que le modèle généralise bien sur des données non vues.

3. Applications variées

Cette méthode est principalement utilisée dans les problèmes de classification, mais elle peut également être adaptée pour certaines situations en régression, notamment lorsque les valeurs cibles présentent une distribution déséquilibrée.

Conclusion

Stratified K-Fold Cross-Validation est un outil essentiel pour évaluer les modèles d'apprentissage automatique, en particulier dans les contextes où les classes cibles sont déséquilibrées. En garantissant que chaque pli représente fidèlement la distribution des classes dans l'ensemble complet, cette méthode aide à obtenir des évaluations plus précises et fiables de la performance du modèle.

10 – GridSearchCV

effectue une recherche exhaustive sur un ensemble de paramètres spécifiés pour un modèle donné. Il évalue chaque combinaison possible de ces paramètres en utilisant la validation croisée pour déterminer celle qui offre les meilleures performances selon une métrique choisie.

Fonctionnement

1. Définition de la grille de paramètres : Vous devez d'abord définir un dictionnaire (appelé "param_grid") qui contient les hyperparamètres à optimiser et les valeurs à tester pour chacun d'eux. Par exemple, pour un modèle SVM, vous pourriez vouloir ajuster les paramètres C et gamma.

2. Création d'un objet GridSearchCV : Vous créez ensuite une instance de GridSearchCV en spécifiant le modèle (estimator), la grille de paramètres et la métrique de performance à utiliser pour évaluer les différentes combinaisons.

3. Entraînement du modèle : En appelant la méthode fit, GridSearchCV entraîne le modèle sur chaque combinaison d'hyperparamètres en utilisant la validation croisée, ce qui permet d'évaluer la performance du modèle sur différents sous-ensembles des données.

4. Sélection des meilleurs paramètres : Après avoir évalué toutes les combinaisons, GridSearchCV sélectionne celle qui a donné les meilleures performances selon la métrique spécifiée et fournit également le modèle entraîné avec ces meilleurs paramètres.

Avantages de GridSearchCV

1. Automatisation du processus d'optimisation : GridSearchCV automatise la recherche des meilleures combinaisons d'hyperparamètres, ce qui réduit le temps et l'effort nécessaires par rapport à une recherche manuelle.

2. Validation croisée intégrée : En utilisant la validation croisée, il permet d'évaluer la capacité de généralisation du modèle et aide à éviter le surajustement (overfitting).

3. Flexibilité : Il peut être utilisé avec n'importe quel estimateur disponible dans Scikit-learn, ce qui en fait un outil polyvalent pour divers types de modèles.

4. Mesures multiples : Vous pouvez spécifier plusieurs métriques de performance pour évaluer les modèles, ce qui permet une évaluation plus complète des performances.

Limitations

- Coût computationnel : La recherche exhaustive peut être très coûteuse en temps et en ressources, surtout si le nombre de combinaisons possibles est élevé.

- Dépendance à la grille définie : Les "meilleurs" hyperparamètres trouvés ne sont que ceux inclus dans la grille définie ; si des valeurs optimales se trouvent en dehors de cette grille, elles ne seront pas considérées.

Conclusion

GridSearchCV est un outil puissant pour l'optimisation des hyperparamètres dans l'apprentissage automatique. En automatisant le processus d'évaluation et en utilisant la validation croisée pour garantir que les résultats sont fiables, il aide les praticiens à améliorer significativement les performances des modèles tout en économisant du temps et des efforts.

11 - Combiner GridSearchCV avec Stratified K-Fold

est une pratique courante en apprentissage automatique, surtout dans les problèmes de classification. Voici pourquoi cette combinaison est bénéfique :

1. Gestion des classes déséquilibrées

Importance de la stratification

- Stratified K-Fold garantit que chaque pli (fold) dans la validation croisée contient une proportion similaire de chaque classe par rapport à l'ensemble de données complet. Cela est particulièrement important dans les cas où les classes sont déséquilibrées (par exemple, lorsque certaines classes sont beaucoup moins fréquentes que d'autres).

- En utilisant Stratified K-Fold avec GridSearchCV, vous vous assurez que le modèle est évalué sur des sous-ensembles qui reflètent la distribution réelle des classes, ce qui permet d'obtenir des estimations de performance plus fiables.

2. Évaluation robuste des hyperparamètres

Validation croisée intégrée

- GridSearchCV effectue une recherche exhaustive sur un ensemble de paramètres tout en utilisant la validation croisée pour évaluer chaque combinaison. Lorsque vous combinez cela avec Stratified K-Fold, chaque combinaison d'hyperparamètres est testée sur plusieurs plis, ce qui réduit le risque d'overfitting et fournit une évaluation plus robuste de la performance du modèle.

- Cela permet de s'assurer que les meilleurs hyperparamètres sélectionnés ne sont pas simplement adaptés à un sous-ensemble particulier des données, mais qu'ils fonctionnent bien sur l'ensemble des données.

3. Automatisation du processus

Simplification du flux de travail

- En intégrant Stratified K-Fold directement dans GridSearchCV via le paramètre cv, vous simplifiez le processus d'optimisation des hyperparamètres. Cela permet d'automatiser la gestion des plis tout en s'assurant que chaque pli est stratifié.

- Cela réduit également le risque d'erreurs manuelles lors de la séparation des données et garantit que les étapes de prétraitement (comme la normalisation ou la vectorisation) sont appliquées correctement uniquement aux données d'entraînement.

4. Utilisation efficace des données

Maximisation de l'utilisation des données

- En utilisant Stratified K-Fold avec GridSearchCV, chaque observation dans votre ensemble de données a la possibilité d'être utilisée à la fois pour l'entraînement et pour la validation. Cela maximise l'utilisation des données disponibles, ce qui est particulièrement important lorsque vous avez un ensemble de données limité.

Conclusion

La combinaison de GridSearchCV avec Stratified K-Fold est essentielle pour optimiser les hyperparamètres tout en garantissant une évaluation précise et fiable du modèle, surtout dans les contextes où les classes sont déséquilibrées. Cette approche améliore non seulement la robustesse du modèle, mais simplifie également le processus global d'optimisation et d'évaluation.

12 - La séparation train-validation-test

offre plusieurs avantages par rapport à la simple séparation train-test dans le contexte du développement de modèles d'apprentissage automatique. Voici les principaux bénéfices :

1. Amélioration de l'évaluation du modèle

Ensemble de validation pour l'ajustement des hyperparamètres

- L'ensemble de validation permet une évaluation non biaisée des performances du modèle lors de l'ajustement des hyperparamètres. En utilisant un ensemble de validation distinct, vous pouvez ajuster les paramètres sans introduire de biais provenant de l'ensemble de test, garantissant ainsi que les métriques de performance finales reflètent la capacité du modèle à se généraliser à des données non vues.

Éviter le surajustement

- Avec une séparation train-validation-test, vous pouvez surveiller comment votre modèle performe sur l'ensemble de validation pendant l'entraînement. Cela aide à détecter le surajustement tôt, permettant d'apporter des ajustements (comme arrêter l'entraînement ou changer la complexité du modèle) avant d'évaluer sur l'ensemble de test.

2. Meilleure évaluation de la généralisation

Ensemble de test séparé

- Un ensemble de test dédié permet une évaluation finale des performances du modèle après que tous les processus d'entraînement et de validation soient complets. Cela garantit que l'évaluation est effectuée sur des données complètement non vues, fournissant une estimation plus précise de la façon dont le modèle se comportera dans des applications réelles.

3. Sélection de modèle plus robuste

Comparaison entre plusieurs modèles

- En utilisant un ensemble de validation, vous pouvez comparer plusieurs modèles ou configurations en fonction de leurs performances sur les données de validation. Cette

approche systématique aide à sélectionner le meilleur modèle avant d'effectuer une évaluation finale sur l'ensemble de test.

4. Flexibilité dans l'utilisation des données

Adaptabilité à différents scénarios

- La séparation train-validation-test permet une flexibilité dans la manière dont les données sont allouées à chaque sous-ensemble en fonction des besoins spécifiques. Par exemple, si l'ajustement des hyperparamètres nécessite plus de données, vous pouvez ajuster les proportions en conséquence tout en conservant suffisamment de données pour un test robuste.

5. Réduction de la variance dans les métriques de performance

Lissage des estimations de performance

- En utilisant un ensemble de validation, vous pouvez moyenniser les métriques de performance sur plusieurs exécutions ou configurations, réduisant ainsi la variance dans vos estimations. Cela conduit à des conclusions plus fiables concernant les performances du modèle par rapport à une simple séparation train-test.

Conclusion

En résumé, utiliser une séparation train-validation-test présente des avantages significatifs par rapport à une simple séparation train-test en permettant un meilleur ajustement des hyperparamètres, en évitant le surajustement, en assurant une sélection robuste du modèle et en offrant une évaluation plus claire des capacités de généralisation. Cette approche conduit finalement à des modèles d'apprentissage automatique plus fiables et efficaces qui performe bien sur des données non vues.