TP Ensemble Learning - Explicabilité
5e SDBD

Marie-José HUGUET
Année 2025-2026

Résumé

Ces TP d'apprentissage supervisé ont deux objectifs. Il s'agit dans un premier temps de **mettre en place un processus d'apprentissage supervisé** pour obtenir des modèles précis et performant sur un jeu de données de grande taille. Les modèles considérés sont des modèles d'ensemble learning. Par la suite, il s'agit d'**analyser et d'expliquer les résultats des classifications obtenues**. L'organisation du travail est la suivante :

- TP0 : Etudier le jeu de données (ACSIncome, Etat de Californie (USA))
- TP1 : Etablir des prédictions pour des modèles d'ensemble learning : *Random Forest*, *AdaBoost*, et *XGBoost*
- TP2 : Explicabilité des prédictions
- Remarque : du travail personnel est attendu, en particulier pour fournir une analyse approfondie des résultats obtenus et pour préparer une présentation orale.

L'évaluation de ces TP s'effectue par une présentation à rendre sur moodle en respectant impérativement les consignes (convention de nommage, date de rendu, ...).

Encadrants

Marie-José Huguet, Marianne Defresne, Mohamed Siala

Support

Alexandre Huyghe

1 Compréhension du jeu de données

Nous allons étudier un jeu de données appelé *ACISIncome*. Dans ce jeu de donnée, chaque exemple est un individu. La tâche de classification (binaire) consiste à prédire si un individu a un revenu supérieur à 50 000 dollars en fonction de plusieurs caractéristiques. Pour le TP, nous considérons les données issues de l'état de Californie (année 2018). Les valeurs des features et du label sont fournies dans deux fichiers séparés sur la page moodle de l'enseignement.

Le jeu de données est décrit dans l'annexe B1 de l'article Ding, Frances, et al. "Retiring adult : New datasets for fair machine learning." Advances in neural information processing systems 34 (2021) : 6478-6490.

L'annexe décrivant les attributs du jeu de données est fournie en page suivante [1].

Le travail à réaliser est :

1. **Effectuez une analyse des attributs** du jeu de données. Vous trouverez également sur moodle des explicitations de l'attribut *OCCP*. Quelles sont les distributions des valeurs ? Observez-vous des corrélations ?
2. **Préparez les données** pour pouvoir ensuite appliquer des méthodes d'apprentissage automatique (nombres, valeurs binaires, catégories, ...). Le type des attributs doit être compatible avec le fonctionnement d'une méthode d'apprentissage de la famille des arbres de décision. Vous pouvez retirer certains attributs.
3. **Partitionner le jeu de données** pour définir un ensemble d'entrainement et un ensemble de test (mélanger les données avant de partitionner). Si vous appliquez une standardisation de certains attributs : expliquez lesquels.

Evaluation. Les observations / analyses effectuées sont à retenir pour la présentation.

1. L'article complet est accessible sur le lien : `https://arxiv.org/pdf/2108.04884.pdf`

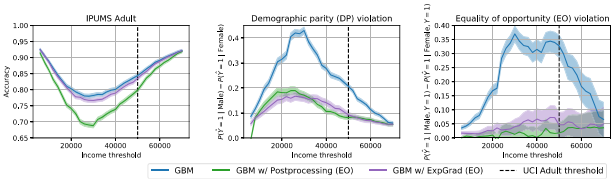

Figure 5: Fairness interventions with varying income threshold on IPUMS Adult. Comparison of in-processing and post-processing methods for achieving equality of opportunity (EO). LFR does not target EO, so we exclude it from the comparison. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for equality of opportunity.

B.1 ACSIncome

Predict whether US working adults' yearly income is above $50,000.

Target: PINCP (Total person's income): an individual's label is 1 if PINCP $>$ 50000, otherwise 0. Note that with our software package, this chosen income threshold can be toggled easily to label the ACS PUMS data differently, and construct a new prediction task.

Features:

- AGEP (Age): Range of values:
  - 0 - 99 (integers)
  - 0 indicates less than 1 year old.
- COW (Class of worker): Range of values:
  - N/A (not in universe)
  - 1: Employee of a private for-profit company or business, or of an individual, for wages, salary, or commissions
  - 2: Employee of a private not-for-profit, tax-exempt, or charitable organization
  - 3: Local government employee (city, county, etc.)
  - 4: State government employee
  - 5: Federal government employee
  - 6: Self-employed in own not incorporated business, professional practice, or farm
  - 7: Self-employed in own incorporated business, professional practice or farm
  - 8: Working without pay in family business or farm

  - 9: Unemployed and last worked 5 years ago or earlier or never worked
- SCHL (Educational attainment): Range of values:
  - N/A (less than 3 years old)
  - 1: No schooling completed
  - 2: Nursery school/preschool
  - 3: Kindergarten
  - 4: Grade 1
  - 5: Grade 2
  - 6: Grade 3
  - 7: Grade 4
  - 8: Grade 5
  - 9: Grade 6
  - 10: Grade 7
  - 11: Grade 8
  - 12: Grade 9
  - 13: Grade 10
  - 14: Grade 11
  - 15: 12th Grade - no diploma
  - 16: Regular high school diploma
  - 17: GED or alternative credential
  - 18: Some college but less than 1 year
  - 19: 1 or more years of college credit but no degree
  - 20: Associate's degree
  - 21: Bachelor's degree
  - 22: Master's degree
  - 23: Professional degree beyond a bachelor's degree
  - 24: Doctorate degree
- MAR (Marital status): Range of values:
  - 1: Married
  - 2: Widowed

  - 3: Divorced
  - 4: Separated
  - 5: Never married or under 15 years old
- OCCP (Occupation): Please see ACS PUMS documentation for the full list of occupation codes
- POBP (Place of birth): Range of values includes most countries and individual US states; please see ACS PUMS documentation for the full list.
- RELP (Relationship): Range of values:
  - 0: Reference person
  - 1: Husband/wife
  - 2: Biological son or daughter
  - 3: Adopted son or daughter
  - 4: Stepson or stepdaughter
  - 5: Brother or sister
  - 6: Father or mother
  - 7: Grandchild
  - 8: Parent-in-law
  - 9: Son-in-law or daughter-in-law
  - 10: Other relative
  - 11: Roomer or boarder
  - 12: Housemate or roommate
  - 13: Unmarried partner
  - 14: Foster child
  - 15: Other nonrelative
  - 16: Institutionalized group quarters population
  - 17: Noninstitutionalized group quarters population
- WKHP (Usual hours worked per week past 12 months): Range of values:
  - N/A (less than 16 years old / did not work during the past 12 months)
  - 1 - 98 integer valued: usual hours worked
  - 99: 99 or more usual hours
- SEX (Sex): Range of values:

  - 1: Male
  - 2: Female
- RAC1P (Recoded detailed race code): Range of values:
  - 1: White alone
  - 2: Black or African American alone
  - 3: American Indian alone
  - 4: Alaska Native alone
  - 5: American Indian and Alaska Native tribes specified, or American Indian or Alaska Native, not specified and no other races
  - 6: Asian alone
  - 7: Native Hawaiian and Other Pacific Islander alone
  - 8: Some Other Race alone
  - 9: Two or More Races

Filters:

- AGEP (Age): Must be greater than 16
- PINCP (Total person's income): Must be greater than 100
- WKHP (Usual hours worked per week past 12 months): Must be greater than 0
- PWGTP (Person weight (relevant for re-weighting dataset to represent the general US population most accurately)): Must be greater than or equal to 1

B.2 ACSPublicCoverage

Predict whether a low-income individual, not eligible for Medicare, has coverage from public health insurance.

Target: PUBCOV (Public health coverage): an individual's label is 1 if PUBCOV == 1 (with public health coverage), otherwise 0.

Features:

- AGEP (Age): Range of values:
  - 0 - 99 (integers)
  - 0 indicates less than 1 year old.
- SCHL (Educational attainment): Range of values:
  - N/A (less than 3 years old)

2 Recherche de bons modèles

Les algorithmes d'apprentissage étudiés dans ces TP sont : RandomForest, AdaBoost, et Gradient-Boosting. Il est important d'avoir compris le principe de ces algorithmes.

2.1 Qualité d'apprentissage avec le paramétrage par défaut (Expe 1)

Pour chaque méthode d'apprentissage, commencer par déterminer la qualité du modèle obtenu avec le paramétrage par défaut de `scikit`***learn*** (accuracy, classification_report, confusion_matrix) en utilisant une validation croisée.

Reportez la taille du jeu de données considéré (voir Table 1). Puis, reportez les résultats en entrainement (de la forme Table 2) et les résultats en test (de la forme Table 3).

Expérimentation 1	Train	Test
Taille Jeu de Données		

Table 1 – Expérimentation (1) : Taille du jeu de données en entrainement et en test

Résultats en **entrainement** (hyper-param. par défaut)	Random Forest	AdaBoost	XGBoost
accuracy			
temps de calcul (sec.)			
matrice de confusion			

Table 2 – Expérimentation (1) : Résultats obtenus en entrainement

Résultats en **test** (hyper-param. par défaut)	Random Forest	AdaBoost	XGBoost
accuracy			
matrice de confusion			

Table 3 – Expérimentation (1) : Résultats obtenus en test

2.2 Optimisation des hyperparamètres des modèles (Expe 2)

Pour chaque méthode d'apprentissage, vous devez ensuite déterminer les valeurs des hyperparamètres permettant d'obtenir le modèle ayant la meilleure qualité d'apprentissage. Utilisez `gridsearchCV` pour cela.

Tout d'abord, précisez la taille du jeu de données utilisé (Table 4) pour cette deuxième expérience (ce doit être le même jeu de données pour toutes les optimisations des modèles).

Expérimentation 2	Train	Test
Taille Jeu de Données		

Table 4 – Expérimentation (2) - Taille du jeu de données

Pour chacune des 3 méthodes :

1. Définissez la liste des hyper-paramètres que vous allez explorer et proposez une justification de vos choix.
2. Précisez également le nombre de plis dans la validation croisée.
3. Estimez combien d'entrainements vont être effectués à partir de vos choix.

4. Construisez un tableau de résultats de la forme de celui proposé pour les Random Forest (voir Table 5).
5. Quelles analyses pouvez-vous apporter pour chacune des méthodes étudiées ?

	Train		Test
	Accuracy	Cpu time	Accuracy
Random Forest (par défaut)			
Valeurs hyperparamètres :			
Random Forest (après optimisation)			
Valeurs hyperparamètres :			

TABLE 5 – Expérimentation (2) - Résultat après optimisation des hyperparamètres pour Random Forest

2.3 Analyse des résultats (Expe 3)

Comparez et analysez et comparez les résultats des différents modèles d'apprentissage. Reprenez les mêmes tableaux que pour la première expérience pour rapporter vos résultats.

2.4 Inférence sur un autre jeu de données (Expe 4)

Peut-on avoir des prédictions pertinentes à partir des (meilleurs) modèles appris sur les données de l'état de Californie pour les états du Nevada et du Colorado (fournis sur la page moodle) ? Analysez vos résultats.

2.5 Impact de la taille du jeu de données (Expe 5)

Quel est l'impact de la taille des données d'entrainement sur la qualité des prédictions ? Considérez uniquement le meilleur réglage obtenu pour les hyperparamètres.

3 Explicabilité des prédictions

Dans cette section, nous allons nous baser sur les meilleurs modèles obtenus dans la section précédente pour étudier des outils d'explications globales et locales des prédictions. Pour cela : choisissez un seul des meilleurs modèles obtenus dans la section 2.

3.1 Classement des attributs dans la prédiction

A partir du modèle entrainé sélectionné, mettez en place une méthode d'évaluation de l'importance des attributs par permtutation des valeurs (*permutation feature importance*). Expliquez le principe de la méthode que vous avez implémentée.

Appliquez votre méthode et donnez un graphique de classement des attributs.

Que vous apportent ces résultats dans la compréhension des prédictions effectuées ? Pourriez-vous réaliser une inférence "à la main" sur des exemples du jeu de données de test ?

3.2 Explications locales

1. **LIME.** En utilisant la librairie `lime`, visualisez les explications fournies pour les prédictions de quelques exemples du jeu de données de test. Analysez les résultats obtenus.

2. **SHAP.** En utilisant la librairie `shap`, visualisez les explications fournies pour les prédictions de quelques exemples du jeu de données de test (via un "Waterfall plot"). Analysez les résultats obtenus.

3. **Comparaison.** Comparez les explications LIME et SHAP sur quelques exemples.

4. **Approfondissement 1.** Avec SHAP, visualisez via un "summary_plot" les contributions des attributs à la prédiction de chaque exemple du jeu de données de test (pour la classe True). Commentez les résultats obtenus.

5. **Approfondissement 2.** Isolez les explications pour quatre sous-groupes via un "summary_plot" : True Positives, True Negatives, False Positives, False Negatives. Commentez les résultats. Pouvez-vous établir si attributs peuvent tromper le modèle ?

3.3 Explication contrefactuelle

A partir d'exemples précis du jeu de données de test, pour chaque attribut étudiez l'impact de la variation de valeur sur la prédiction. Pouvez-vous inverser la prédiction ? Commentez vos résultats.