PARCOURS : DATA SCIENTIST

Interprétabilité des modèles avec SHAP

Ouvrir la boîte noire — comprendre pourquoi votre modèle prédit ce qu'il prédit

PARCOURS : DATA SCIENTIST

DURÉE : ~90 MIN

Liora

À PROPOS DE CETTE MASTERCLASS

Ce que vous allez apprendre

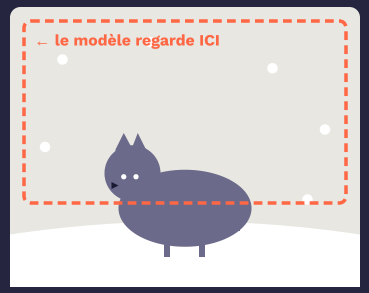
OBJECTIFS D'APPRENTISSAGE

- Expliquer **pourquoi l'interprétabilité compte** — au-delà de la seule performance
- Utiliser les **techniques classiques** (arbres, coefficients, ACP) et connaître leurs limites
- Appliquer **SHAP** aux modèles à base d'arbres pour des explications globales **et** locales
- Lire avec aisance les **summary, dependence et force plots**
- Étendre SHAP à une **classification textuelle multi-classes**
- Mener une **analyse SHAP autonome** sur votre propre jeu de données

Liora

LE PROBLÈME

Un modèle à 98% peut se tromper



L'HISTOIRE CLASSIQUE

Un classifieur loup vs. husky atteint **98% de précision**. Bon modèle ? Il regarde en fait **la neige en arrière-plan**, pas l'animal.

LA LEÇON

Une haute performance **ne suffit pas**. L'interprétabilité est le pont entre performance et **confiance**.

POURQUOI ÇA COMPTE

Quatre raisons d'ouvrir la boîte noire

1 · CONFIANCE

Personne ne déploie ce qu'il ne comprend pas.

2 · DÉBOGAGE

Repérer raccourcis, fuites et corrélations fallacieuses.

3 · ÉQUITÉ

Auditer les variables sensibles : genre, âge, origine.

4 · RÉGULATION

RGPD, AI Act, scoring crédit : un droit à l'explication.

Deux questions, toujours : Le modèle est-il efficace ? Et son raisonnement est-il acceptable ?

Liora

FEUILLE DE ROUTE

Notre parcours en trois parties

PARTIE	QUESTION TRAITÉE	DONNÉES	OUTIL PRINCIPAL
Partie 1	Jusqu'où l'interprétabilité classique peut nous mener ?	Census (tabulaire)	Arbre, coefficients, ACP, SHAP
Partie 2	La même logique tient-elle sur texte et multi-classes ?	Émotions (texte)	SHAP multi-classes
Partie 3	Pouvez-vous mener le workflow seul·e ?	Votre dataset	Analyse SHAP autonome

PARTIE 1

Des modèles simples à la boîte noire

Données tabulaires · Prédiction de revenus

LE JEU DE DONNÉES

Prédire les revenus du Census US

Cible

Cette personne gagne-t-elle **> 50K $** par an ? → classification binaire

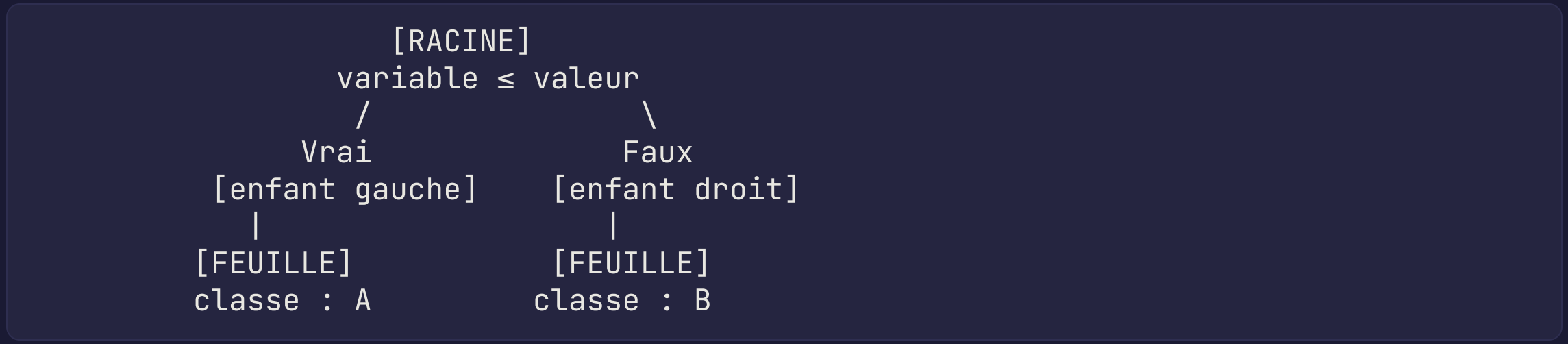
Variables

âge · éducation · statut marital · profession · genre · plus/moins-values · heures travaillées · pays

POURQUOI CE DATASET ?

Les variables sont **lisibles par un humain**. Quand le modèle met en avant **marital-status** ou **hours-per-week**, on peut juger immédiatement si la logique est raisonnable — ou à auditer.

Lire un arbre de décision



gini — impureté · 0 = feuille pure, 0.5 = mélangée

samples — combien d'exemples atteignent ce nœud

value — effectifs par classe → **classe majoritaire**

Dans le notebook → `DecisionTreeClassifier(max_depth=3, random_state=42)` — profondeur 3 pour rester lisible.

La variable à la racine est le premier séparateur le plus fort des données.

INTERPRÉTABLE PAR NATURE · 2

Ce que l'arbre dit (et ne dit pas)

CE QU'IL OFFRE

- **Vue globale** — règles de décision principales
- **Vue locale** — chemin d'un individu
- **Transparence totale** — lecture ligne à ligne

CE QU'IL CACHE

- Les **interactions** riches entre variables
- Une forte **performance prédictive**
- Une méthode **transférable** à d'autres modèles

Lisible uniquement parce qu'il est peu profond. Dès qu'on le laisse grandir, l'interprétabilité disparaît.

INTERPRÉTABLE PAR NATURE · 3

Feature importance & coefficients

IMPORTANCE ARBRES

Fréquence d'utilisation pour splitter × réduction d'impureté.

Limite — liée à l'arbre exact, instable.

COEFFICIENTS LINÉAIRES

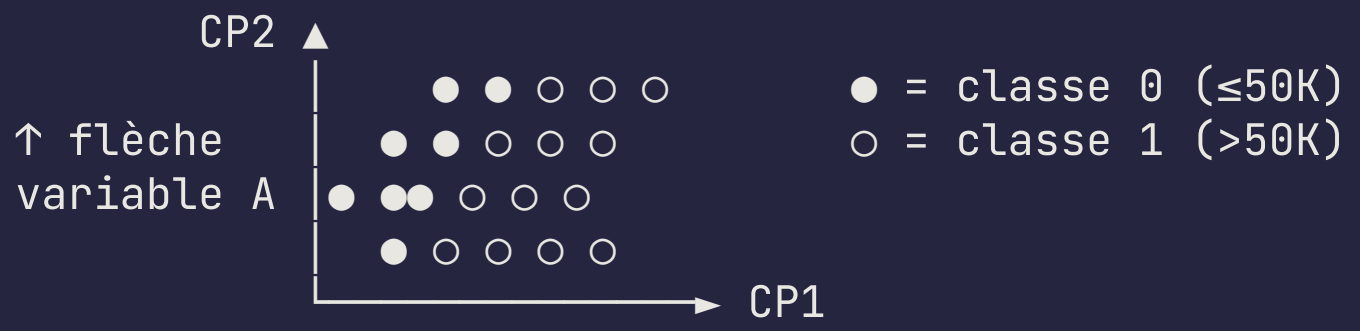
Positif = pousse vers classe 1 · négatif = vers classe 0.

Limite — suppose la linéarité, sensible à l'échelle.

⚠ Une variable sensible comme gender_Female en haut du classement = signal pour **auditer** le modèle.

Dans le notebook → `tree_clf.feature_importances_` · `pipeline['logistic_regression'].coef_[0]` (avec `StandardScaler`).

Biplot ACP — une carte globale



Points — un par individu

Direction flèche — où la variable est élevée

Longueur flèche — influence sur la projection 2D

L'ACP est excellente pour l'exploration. Mais la projection perd de l'information — jamais une explication finale.

Performance vs. interprétabilité

MODÈLE	SCORE F1	INTERPRÉTABILITÉ
Arbre de décision (profondeur 3)	~0.75	Règles lisibles
Régression logistique	~0.78	Coefficients directement inspectables
XGBoost	~0.81	Non inspectable directement

LE PROBLÈME

Meilleure performance → moins lisible. Pas d'arbre unique à inspecter. Pas de tableau de coefficients. Le raisonnement est **distribué sur des centaines d'arbres**.

Dans le notebook → `xgb.XGBClassifier(eval_metric='logloss', random_state=42)` — 100 arbres par défaut, `max_depth=6`. C'est ce modèle qu'on va expliquer avec SHAP.

PARTIE 2

SHAP

à la rescousse

Un langage unifié pour tout modèle

SHAP en une image mentale

LA QUESTION À LAQUELLE SHAP RÉPOND

Pour cette prédiction, **quelle est la contribution de chaque variable**, par rapport à une ligne de base ?

```
prédiction de base (expected value)
      + contribution de la variable 1
      + contribution de la variable 2
      + contribution de la variable 3
      + ...
      = prédiction pour cette personne
```

Voyez SHAP comme un **outil de décomposition de prédiction**. Il découpe une prédiction en parts additives lisibles.

LA FONDATION

Pourquoi SHAP est fiable

THÉORIE DES JEUX

Basée sur les **valeurs de Shapley** — idée primée au Nobel pour répartir équitablement le gain d'une équipe.

PROPRIÉTÉS ÉQUITABLES

Efficacité · Symétrie · Dummy · Additivité — ces **quatre axiomes** définissent SHAP de manière unique.

AGNOSTIQUE AU MODÈLE

Fonctionne pour arbres, linéaires, réseaux de neurones — avec un `TreeExplainer` optimisé pour XGBoost.

LE WORKFLOW

Quatre étapes pour appliquer SHAP

1. **Entraîner votre modèle**

   N'importe quel modèle — mais les boosters d'arbres (XGBoost, LightGBM) vont avec un `TreeExplainer` rapide.

2. **Créer l'explainer**

   `explainer = shap.TreeExplainer(xgb_model, data=X_train, model_output="probability")` · `data=X_train` sert de **jeu de fond** — c'est sur lui qu'on calcule la prédiction moyenne (= expected value).

3. **Calculer les valeurs SHAP**

   `shap_values = explainer.shap_values(X_test)` → une matrice **(n_samples, n_features)**

4. **Visualiser & interpréter**

   Bar · beeswarm · dependence · force — nous allons lire les quatre.

STRUCTURE DE SORTIE

La matrice des valeurs SHAP

	variable 1	variable 2	variable 3	...
personne 1	+0.12	-0.03	+0.01	
personne 2	-0.08	+0.10	0.00	
personne 3	+0.02	-0.01	-0.05	

LIGNES

Une par individu du test set

COLONNES

Une par variable

SIGNE

+ pousse vers le haut · **–** vers le bas

Proche de zéro → la variable a très peu influencé cette prédiction.

Expected value = prédiction moyenne

L'**expected value** est simplement la **prédiction moyenne du modèle** sur les données de fond — `E[f(X)] = np.mean(model.predict(X_background))`.

EXEMPLE CENSUS · BINAIRE

```
model.predict_proba(X_train)[:,1].mean()
≈ 0.24  ← expected_value
(~24% des gens gagnent >50K dans X_train)
```

Lecture — avant de voir les variables d'**une** personne, le modèle partirait de **0.24**. Les SHAP values mesurent l'écart entre cette moyenne et la prédiction **réelle** de cette personne.

```
expected_value + Σ shap_values[personne] = model.predict_proba(personne)
```

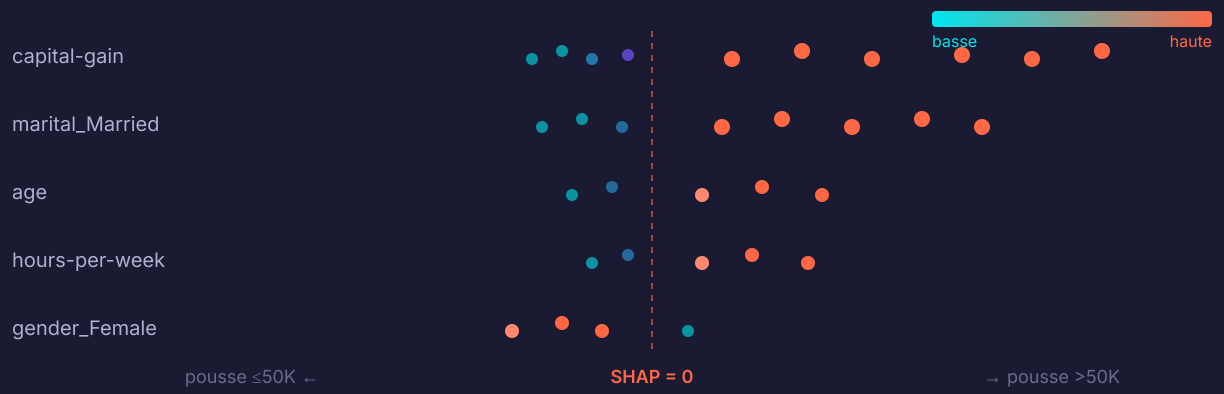
VUE GLOBALE · 1

Le bar plot — ce qui compte en moyenne

Chaque barre = **|SHAP| moyenne** — influence sans direction

⚠ Classe les variables — **sans** dire si elles poussent vers ≤50K ou >50K

Le beeswarm — comment et pour qui



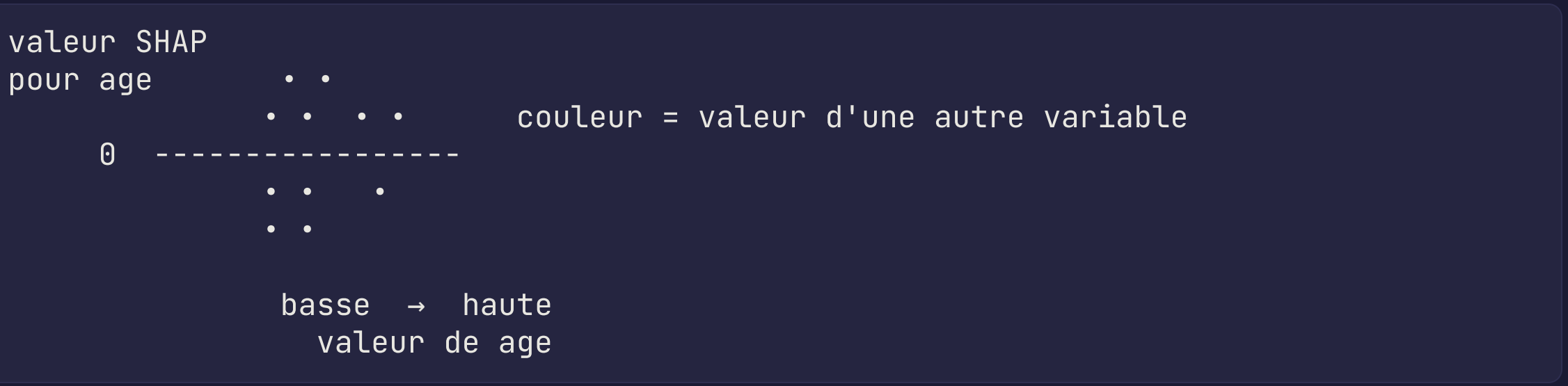
Chaque point = un individu

Position X = valeur SHAP

Couleur = basse (cyan) / haute (orange)

Fortes valeurs de capital-gain → >50K. gender_Female pousse légèrement vers ≤50K — signal à auditer.

Le dependence plot — comment une variable se comporte



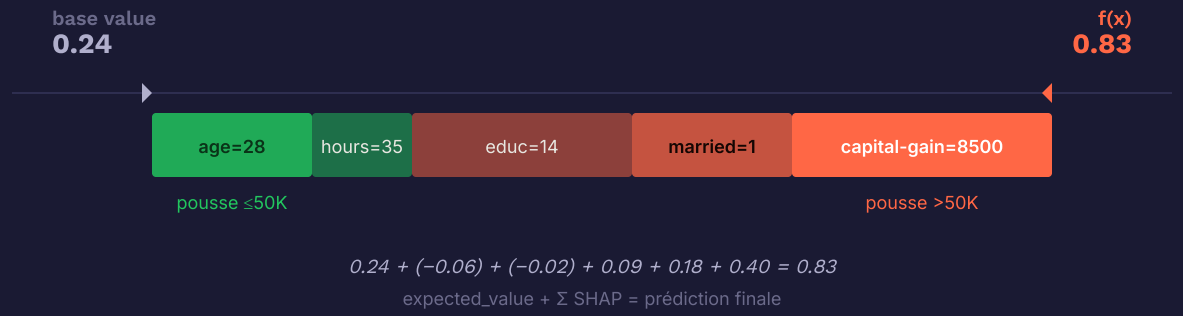
Tendance ascendante — une valeur plus élevée fait monter la prédiction

Plat — la variable pèse peu sur cet intervalle

Motif de couleur — révèle une **interaction** avec une autre variable

VUE LOCALE · 2

Le force plot — une décision précise



Blocs rouges → poussent vers le haut

Blocs verts → poussent vers le bas

Largeur = force de la contribution

Le force plot dit pourquoi cette personne précise a cette prédiction.

SHAP est puissant — pas magique

PRÉCAUTION	CE QUE CELA VEUT DIRE
Variables corrélées	Le crédit peut se répartir entre jumelles de manière floue
Coût de calcul	`TreeExplainer` est rapide — d'autres explainers peuvent être lents
Pas la causalité	Une forte valeur SHAP est une association, pas une **cause**
Instabilité locale	Des individus similaires → explications parfois différentes
Choix de l'explainer	Différents explainers se comportent différemment selon les modèles

Utilisez SHAP comme un **cadre structuré** pour comprendre le modèle — pas comme une vérité absolue.

PARTIE 3

SHAP sur texte et multi-classes

Classification d'émotions à 6 classes

Du tabulaire au texte, de 2 à 6 classes

ASPECT	PARTIE 1 (BINAIRE)	PARTIE 3 (MULTI-CLASSES TEXTE)
Variables	Colonnes tabulaires	Mots / tokens
Tâche	Binaire	6 classes (tristesse · joie · peur · colère · surprise · dégoût)
Espace SHAP	Probabilités	Logits (scores bruts)
Unité d'explication	Une par prédiction	Une par classe, par prédiction

✓ Même workflow — `TreeExplainer`, `shap_values`, force plots. Deux concepts nouveaux seulement.

Dans le notebook → `xgb.XGBClassifier(objective='multi:softprob', num_class=6)`

Logits vs. probabilités

scores bruts (logits)		—softmax—▶️	probabilités sommant à 1
tristesse	3.1		tristesse 0.72
joie	0.7		joie 0.08
peur	1.0		peur 0.10
colère	0.2		colère 0.05
surprise	-0.3		surprise 0.03
dégoût	-0.6		dégoût 0.02

$$\text{probabilité}_k = \frac{e^{\text{logit}_k}}{\sum_j e^{\text{logit}_j}}$$

En multi-classes, `TreeExplainer` explique les scores bruts — les probabilités viennent après la softmax.

NOUVEAU CONCEPT · 2

Une explication par classe

```
shap_values
│
├── classe 0: tristesse  → array (n_samples, n_features)
├── classe 1: joie       → array (n_samples, n_features)
├── classe 2: peur       → array (n_samples, n_features)
├── classe 3: colère     → array (n_samples, n_features)
├── classe 4: surprise   → array (n_samples, n_features)
└── classe 5: dégoût     → array (n_samples, n_features)
```

Format liste → `shap_values[3][5]` pour la personne 5, classe colère

Array 3-D → `shap_values[5, :, 3]` équivalent

LIRE LES FORCE PLOTS MULTI-CLASSES

Quels mots déclenchent chaque émotion ?

TRISTESSE · OBS #4

Des mots comme **horrible**, **ingrat** poussent le score tristesse vers le haut. D'autres tirent doucement de l'autre côté.

COLÈRE · OBS #5

Un seul mot fort (**profane**) domine. Des mots apaisants comme **sentir** apparaissent de l'autre côté.

L'important n'est pas juste quels mots apparaissent — c'est **quelle classe ils soutiennent** dans cette explication.

DERNIÈRE PARTIE

À votre tour maintenant

Appliquez le workflow SHAP complet sur vos données

Votre workflow SHAP en 5 étapes

1. **Charger & explorer**
   Valeurs manquantes, équilibre cible, colonnes sensibles.
2. **Préparer & entraîner**
   Encoder, splitter, entraîner un `XGBClassifier`, afficher `classification_report`.
3. **SHAP global — bar + beeswarm**
   Top 3 variables · direction · vérification métier.
4. **SHAP local — force plots**
   Un cas clairement positif · un négatif · un incertain.
5. **Contrôle équité / risque**
   Attributs sensibles · proxies · validation supplémentaire ?

SYNTHÈSE

À retenir

À RETENIR

- Haute précision ≠ fiable. Toujours demander : **pourquoi cette prédiction ?**
- Outils classiques (arbre, coefficients, ACP) sont **intuitifs** mais **limités**
- SHAP offre un **langage unifié** : décomposition = ligne de base + contributions
- Deux vues : **globale** (bar, beeswarm) et **locale** (force)
- Multi-classes ? Une explication **par classe** — en espace logit
- SHAP ≠ causalité. Il révèle des **motifs appris**, pas la vérité du monde

Que signifie une valeur SHAP positive ?

A — La variable est globalement importante sur tout le jeu de données

B — Cette variable a poussé cette prédiction **au-dessus** de la ligne de base ✓

C — La variable a causé le résultat dans le monde réel

D — La prédiction est certainement correcte

Les valeurs SHAP sont **locales** et **associatives** — une prédiction, une personne, une contribution.

POUR ALLER PLUS LOIN

Ressources recommandées

LIVRES

- **Interpretable ML** — C. Molnar
- **Hands-On ML** — A. Géron (Ch. 6 + bonus)

ARTICLES

- Lundberg & Lee (2017) — **A Unified Approach to Interpreting Model Predictions**
- Ribeiro et al. (2016) — **LIME: Why Should I Trust You?**

LIBRAIRIES

- `shap` — shap.readthedocs.io
- `lime` · `eli5` · `captum` (PyTorch)

MASTERCLASS TERMINÉE

Merci !

Des questions ? Échangeons.