le cnam

PROJET RCP209

Apprentissage statistique 2

Décision d'accorder ou non un crédit bancaire

PLUSQUELLEC Valérie
Janvier 2026

TABLE DES MATIERES

INTRODUCTION

Ce document repose sur l'étude d'un jeu de données *Statlog (German Credit Data)* [Hofmann, 1994], référencé sur le site de UCI Machine Learning Repository. Il compile des données de crédit bancaire des années 90, décrivant 1000 clients d'une banque Allemande ayant fait une demande de prêt.

À partir de ces données, l'objectif est de construire un modèle décisionnel pouvant classer des individus en 2 classes : crédit non risqué (0) ou crédit risqué (1).

Dans un premier temps, une analyse exploratoire des données permettra d'examiner la distribution des variables, la répartition des classes, et les corrélations entre variables. Cette étape justifiera le choix des modèles décisionnels et l'usage de métriques adaptées. Les résultats des modèles sélectionnés seront ensuite présentés. Enfin, la comparaison de ces résultats permettra de déterminer le modèle optimal pour ce problème de classification.

I. ANALYSE EXPLORATOIRE DES DONNEES

A. DESCRIPTION DES DONNEES

Avant toute analyse, une étude détaillée du jeu de données est nécessaire. Les 1000 individus sont évalués sur 20 variables qui sont décrites dans le tableau ci-dessous :

	Nom	Type	Description	Modalités possibles
1	status_checking	Qualitatif	Catégorie de l'individu selon la somme d'argent sur le compte bancaire existant	De A11 à A14
2	duration	Quantitatif	Durée du prêt en mois	
3	credit_history	Qualitatif	Historique des crédits	De A30 à A34
4	purpose	Qualitatif	Motif de la demande de prêt	De A40 à A49 et A410
5	credit_amount	Quantitatif	Montant du prêt en Deutschemark (DM)	
6	savings	Qualitatif	Catégorie selon les économies placées	De A61 à A65
7	employment	Qualitatif	Ancienneté dans l'emploi actuel	De A71 à A75
8	installment_rate	Quantitatif	Taux de mensualité en pourcentage du revenu disponible	
9	personal_status_sex	Qualitatif	Statut marital et sexe	De A91 à A95
10	other_debtors	Qualitatif	Présence d'un co-emprunteur ou d'un garant	De A101 à A103
11	residence_since	Quantitatif	Ancienneté dans l'habitation principale actuelle en années	
12	property	Qualitatif	Placements, propriétés	De A121 à A124
13	age	Quantitatif	Age en années	
14	other_installment_plans	Qualitatif	Existence de crédits dans d'autres banques ou d'achats en magasin en plusieurs fois	De A141 à A143
15	housing	Qualitatif	Habitant propriétaire ou locataire	De A151 à A153
16	number_existing_credits	Quantitatif	Nombre de credits existants	
17	job	Qualitatif	Catégorie professionnelle	De A171 à A174
18	people_liable	Quantitatif	Nombre de personnes à charge	
19	telephone	Qualitatif	Téléphone enregistré	De A191 à A192
20	foreign_worker	Qualitatif	Travailleur étranger	De A201 à A202

Fig.I-1 Liste des variables (modalités détaillées en annexe A.1)

On dénombre ainsi 13 variables qualitatives et 7 variables quantitatives.
A ces variables, s'ajoute une variable cible (label), de nature binaire, qui indique pour chaque client si le prêt a été remboursé ou si cela n'a pas été le cas (valeur 0 pour oui, valeur 1 pour non).

B. ANALYSE STATISTIQUE UNIVARIÉE

	duration	credit_amount	installment_rate	residence_since	age	number_existing_credits	people_liable	label
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	20.903000	3271.258000	2.973000	2.845000	35.546000	1.407000	1.155000	0.300000
std	12.058814	2822.736876	1.118715	1.103718	11.375469	0.577654	0.362086	0.458487
min	4.000000	250.000000	1.000000	1.000000	19.000000	1.000000	1.000000	0.000000
25%	12.000000	1365.500000	2.000000	2.000000	27.000000	1.000000	1.000000	0.000000
50%	18.000000	2319.500000	3.000000	3.000000	33.000000	1.000000	1.000000	0.000000
75%	24.000000	3972.250000	4.000000	4.000000	42.000000	2.000000	1.000000	1.000000
max	72.000000	18424.000000	4.000000	4.000000	75.000000	4.000000	2.000000	1.000000

Fig.I-2 Résumé statistique des variables quantitatives.

Le tableau ci-dessus révèle en particulier que :

- la durée moyenne d'emprunt est de 20,9 mois (médiane de 18 mois), ce qui signifie que les prêts courts sont dominants ;
- le montant moyen emprunté est de 3 271 DM (médian de 2319, maximum de 18 424 DM), ce qui indique l'existence de quelques prêts hors norme (*outliers*) pouvant influencer les modèles linéaires ;
- l'âge moyen est de 35,5 ans (médian de 33 ans), ce qui désigne une population d'emprunteurs composée de jeunes adultes.

En observant les répartitions des modalités (voir annexes A.1 et A.2), on peut ajouter que la plupart des individus de la banque de données ont majoritairement un bon historique de crédit, ont peu d'économies, sont des hommes célibataires, n'ont ni dettes, ni charges, sont propriétaires de leur habitation et sont des employés qualifiés d'origine allemande.

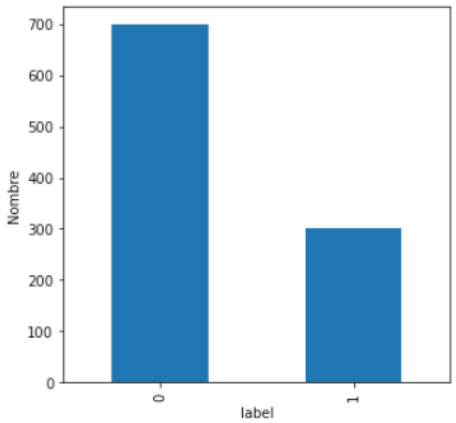


Fig.I-3 Répartition des labels dans le jeu de données

Enfin, le diagramme ci-dessus indique que le jeu de données est déséquilibré, avec une classe prédominante de bons payeurs.

C. ANALYSE STATISTIQUE BIVARIÉE

Dans cette partie, on s'intéresse aux relations entre les variables.

	Corrélation avec risque	Abs
duration	0.214927	0.214927
credit_amount	0.154739	0.154739
age	-0.091127	0.091127
installment_rate	0.072404	0.072404
number_existing_credits	-0.045732	0.045732
credit_history	0.036472	0.036472
status_checking	0.028267	0.028267
foreign_worker	0.019853	0.019853
other_debtors	0.015300	0.015300
purpose	0.011868	0.011868

Fig.I-4 Tableau des 10 variables les plus corrélées avec le label

Ce tableau montre que les trois variables les plus corrélées avec le label sont la durée d'emprunt, le montant du crédit, et l'âge.

En effet, les fréquences de mauvais payeurs selon les valeurs sur ces variables (exemple à la figure I-5) peuvent s'interpréter ainsi : un prêt sera d'autant plus risqué qu'il sera remboursé sur une longue durée, que son montant sera élevé, et que l'emprunteur est jeune.

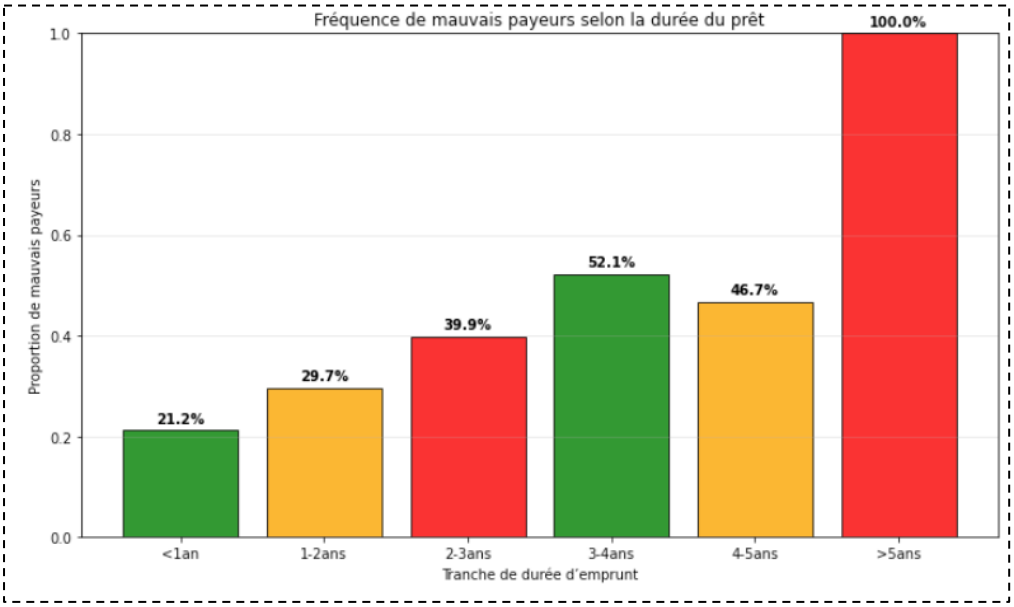


Fig.I-5 Fréquence de mauvais payeurs par rapport à la durée de l'emprunt (extrait de l'annexe A.3)

La figure I-4 permet aussi de remarquer l'absence de liaison linéaire avec le label. En observant la matrice de corrélations entre les variables quantitatives (figure I-6), on peut ajouter qu'il n'existe pas, non plus, de liaisons linéaires entre elles.

Fig.I-6 Matrice de corrélations entre les variables quantitatives

Il apparaît donc justifié de faire appel à des modèles décisionnels pour répondre à la problématique de cette étude.

D. CONSEQUENCES SUR LA METHODOLOGIE

Ainsi, l'étude reposera sur des méthodes d'apprentissage supervisé. Le jeu de données fourni sera séparé en deux : 70% servira à l'apprentissage (training set), 30% à tester le modèle obtenu (test set), et le split prendra en compte la répartition inégale des labels (stratification).

Un prétraitement (preprocessing) sur les données est nécessaire pour normaliser les 7 variables numériques (StandardScaler) et encoder les 13 variables catégorielles (OneHotEncoder).

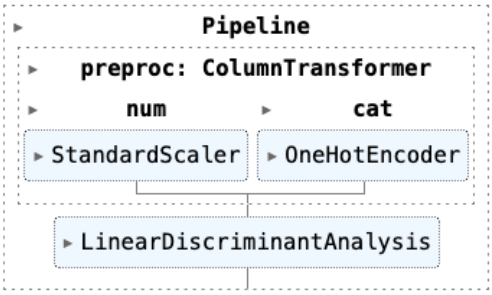


Fig.I-7 Exemple de pipeline preprocessing+modèle pour l'AFD

Ce preprocessing restera le même pour tous les modèles, ce qui garantira l'équité au moment de les comparer.

Selon les modèles, une validation croisée 5-fold stratifiée combinée à GridSearchCV sélectionnera les hyperparamètres optimaux maximisant le score sur les données d'entraînement. Le principe est le suivant :

- les données d'entraînement sont partitionnées en 5 plis égaux (pour 700 observations, cela correspond à 140 observations par pli) ;
- pour chaque pli, les 4 autres plis servent à entraîner le modèle, et le pli à évaluer les performances du modèle ; ce processus est répété 5 fois (chaque pli est testé une fois) ;
- les 5 scores AUC obtenus sont moyennés, et les hyperparamètres ayant donné le meilleur score CV sont retenus pour l'évaluation finale sur le jeu de test.

Enfin, aux données *Statlog (German Credit Data)*, est annexée une matrice de coût :

Vraie classe\ Prédiction	0 (Bon)	1 (Mauvais)
0 (Bon)	0	1
1 (Mauvais)	5	0

Fig.I-8 Matrice de coût

Il est ainsi considéré comme cinq fois plus grave d'accorder un crédit à un mauvais payeur que de le refuser à un bon payeur. Dans l'analyse des résultats, cette matrice sera utilisée pour calculer un coût total du modèle :

$$\text{Coût total} = 5 \times FN + FP \qquad \text{(avec FN = faux négatifs, FP = faux positifs)}$$

Ce coût total, et les métriques classiques (accuracy et AUC), permettront d'évaluer les modèles décisionnels testés.

L'objectif étant d'établir une classification en deux classes, et le nombre de données étant réduit (1000 individus), les méthodes retenues qui seront présentées et comparées par la suite seront :

- une **analyse factorielle discriminante** (modèle de base linéaire)
- un **arbre de décision CART** (modèle décisionnel simple)
- une **forêt aléatoire** (agrégation parallèle d'arbres)
- un modèle **AdaBoost** (agrégation séquentielle d'arbres)
- une **machine à vecteurs de support non linéaire à noyau Gaussien** (modèle non linéaire)
- un **perceptron multicouches** (modèle non-linéaire pour relations complexes)

Ces modèles ont été sélectionnés pour leur complémentarité et leur pertinence face au déséquilibre des classes cible et aux faibles corrélations linéaires observées.

II. MODÈLES DÉCISIONNELS

A. ANALYSE FACTORIELLE DISCRIMINANTE

L'Analyse Factorielle Discriminante (AFD) recherche la frontière de décision linéaire optimale séparant les bons et les mauvais payeurs, en maximisant la variance inter-classes.
Il est à noter que l'AFD n'a pas d'hyperparamètres à tuner (c'est un modèle paramétrique déterministe).

L'accuracy (ou taux de bonne classification) mesure la proportion de prédictions correctes faites par un modèle de classification.

```
=== Analyse Factorielle Discriminante ===
Accuracy Train: 0.790
Accuracy Test:  0.757
```

Fig.II-1 Comparaison des accuracy train/test.

Cette comparaison permet de remarquer que l'AFD surapprend légèrement, avec un écart train/test d'environ 3,3%. Un accuracy entre 75,7% et 79% est très satisfaisant pour un premier modèle.

Mais l'*accuracy* est une métrique biaisée en faveur de la classe la plus représentée (ici, celle des bons payeurs). Pour la suite, l'accuracy ne servira qu'à évaluer le surapprentissage (*overfitting*) des modèles.

Par le déséquilibre des classes cible, il est plus intéressant ici de faire apparaître l'AUC Test (Area Under Curve Test) qui mesure la capacité du modèle à discriminer entre "bon payeur" et "mauvais payeur".

```
AUC Test: 0.798
```

Fig.II-4 AUC obtenu pour l'AFD

Ce résultat signifie que sur 100 paires aléatoires (bon/mauvais), l'AFD classe correctement l'un et l'autre dans 79.8 % des cas, ce qui est un très bon score.

D'autres mesures adaptées à cette étude sont la précision et le rappel de chaque classe. Dans le cas présent, c'est notamment la précision et le rappel de la classe 1 (mauvais payeur) qui sont révélatrices.

```
========== Métriques pour l'AFD ============
              precision    recall  f1-score   support

           0       0.80      0.86      0.83       210
           1       0.61      0.51      0.56        90

    accuracy                           0.76       300
   macro avg       0.71      0.69      0.69       300
weighted avg       0.75      0.76      0.75       300
```

Fig.II-2 Métriques de l'AFD

Ainsi, sur 100 individus classés « mauvais payeurs » dans le jeu de test, seuls 61 ont été correctement classés par l'AFD, et sur 100 mauvais payeurs, seuls 51 sont détectés. Ces performances sur la classe des « mauvais payeurs » génèrent un coût total de 249 comme peut l'indiquer la figure ci-dessous.

```
Matrice de confusion:
[[181  29]
 [ 44  46]]
Coût total (5*FN + FP): 249
```

Fig.II-3 Matrice de confusion et calcul du coût sur les données test

L'AFD récolte donc de bons scores, mais le coût reste élevé : les mauvais payeurs ne sont pas détectés de manière satisfaisante. Par la suite, cette étude cherchera à minimiser ce coût, tout en conservant des scores satisfaisants.

B. ARBRE DE DECISION CART

Les arbres de décision sont des méthodes d'apprentissage non paramétriques utilisées pour des problèmes de classification et de régression. Ils peuvent donc être appliqués pour cette étude. Plus précisément, une implémentation CART semble adaptée ici : elle génère des arbres binaires, et trouve le meilleur split (seuil/variable) à chaque nœud pour homogénéiser les classes.

Pour éviter le surapprentissage, on cherche à tuner la profondeur maximale de l'arbre et nombre de feuilles minimal. On utilise la validation croisée 5-fold avec GridSearchCV (présentée dans l'introduction), qui cherche à optimiser le score AUC.

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
Meilleurs paramètres : {'cart__max_depth': 5, 'cart__min_samples_leaf': 20}
Meilleur score CV : 0.7239067055393585
```

Fig.II-5 Résultat du GridSearch 5-fold

```
=== Arbre CART OPTIMAL ===
Accuracy Train: 0.760
Accuracy Test:  0.703

        ========= Métriques CART optimal ============
              precision    recall  f1-score   support

           0       0.78      0.80      0.79       210
           1       0.51      0.48      0.49        90

    accuracy                           0.70       300
   macro avg       0.64      0.64      0.64       300
weighted avg       0.70      0.70      0.70       300

AUC Test: 0.708
```

Fig.II-6 Résultats du modèle avec les hyperparamètres obtenus par GridSearch 5-fold

L'arbre CART optimal obtenu a une profondeur maximale de 5 et 20 feuilles au minimum. Son accuracy montre une bonne généralisation avec un écart train/test de 5,7%, bien que légèrement plus marqué que l'AFD (3,3%).

L'AUC Test de 0,708 indique une capacité discriminante correcte : sur 100 paires aléatoires (bon/mauvais payeur), l'arbre CART classe correctement les paires dans 70,8% des cas (vs 79,8% AFD).

Sur la classe des « mauvais payeurs », le modèle CART n'obtient pas de meilleurs résultats que l'AFD. Sur 100 individus classés « mauvais payeurs », 51 sont correctement identifiés, et sur 100 mauvais payeurs, seuls 48 sont détectés.

```
Matrice de confusion:
[[168  42]
 [ 47  43]]
Coût total (5*FN + FP): 277
```

Fig.II-7 Matrice de confusion CART et coût

Le modèle CART génère un coût total de 277, légèrement supérieur à l'AFD (249), ce qui n'est pas satisfaisant.
Il est possible d'ajouter un paramètre (*class_weight='balanced'*) dans le modèle qui pénalise les erreurs sur la classe minoritaire. La formule du poids se fait en divisant le nombres d'individus par le produit du nombre de classes et du nombre d'individus de la classe.
Ainsi, avec ce paramètre, la classe (0) a un poids d'environ 0,7, et la classe 1 un poids d'environ 1,7.

Les résultats obtenus sont alors bien meilleurs :

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
Meilleurs paramètres : {'cart__max_depth': 3, 'cart__min_samples_leaf': 20}
Meilleur score CV : 0.7185374149659863

=== Arbre CART OPTIMAL ===
Accuracy Train: 0.673
Accuracy Test:  0.637

        ========= Métriques CART optimal ============
              precision    recall  f1-score   support

           0       0.84      0.59      0.69       210
           1       0.44      0.74      0.55        90

    accuracy                           0.64       300
   macro avg       0.64      0.67      0.62       300
weighted avg       0.72      0.64      0.65       300

AUC Test: 0.712

Matrice de confusion:
[[124  86]
 [ 23  67]]
Coût total (5*FN + FP): 201
```

Fig II-8 Résultats de CART avec poids

Désormais, l'overfitting passe à 3,6%, l'AUC est de 71,2% et le rappel de la classe (1) devient excellent (74%). Le coût baisse alors à 201.

L'ajout du paramètre de poids sur les classes est donc pertinent, et sera appliqué par défaut pour les modèles qui suivent.

C. FORÊT ALEATOIRE

La forêt aléatoire agrège en parallèle les prédictions de multiples arbres CART (*bagging*) entraînés sur des sous-ensembles aléatoires des données et des variables, réduisant la variance.

Les hyperparamètres à optimiser par GridSearchCV sont le nombre d'arbres dans la forêt, et les hyperparamètres communs à chaque arbre (profondeur maximale et nombre de feuilles minimal).

```
Fitting 5 folds for each of 27 candidates, totalling 135 fits
Meilleurs hyperparamètres: {'rf__max_depth': 10, 'rf__min_samples_leaf': 10,
'rf__n_estimators': 20}
Score CV AUC: 0.771

=== Forêt Aléatoire OPTIMALE ===
Accuracy Train: 0.783
Accuracy Test:  0.680

        ========= Métriques RF ============
              precision    recall  f1-score   support

           0       0.82      0.69      0.75       210
           1       0.48      0.66      0.55        90

    accuracy                           0.68       300
   macro avg       0.65      0.67      0.65       300
weighted avg       0.72      0.68      0.69       300

AUC Test: 0.751

Matrice de confusion:
[[145  65]
 [ 31  59]]
Coût total (5*FN + FP): 220
```

Fig.II-9 Résultats de la forêt aléatoire

La forêt aléatoire optimale obtenue (20 arbres avec une profondeur maximale égale à 10 et un minimum de feuilles à10) présente un surapprentissage élevé de 10,3% (vs 3,6% pour CART).

L'AUC Test de 0,751 indique une très bonne capacité discriminante : sur 100 paires aléatoires (bon/mauvais payeur), 75,1% sont bien classés.

Sur la classe des « mauvais payeurs », le modèle obtient des résultats corrects. Sur 100 individus classés « mauvais payeurs », 48 sont correctement identifiés, et sur 100 mauvais payeurs, 66 sont détectés.

Ce modèle génère un coût total de 220, ce qui est un coût plus élevé que celui obtenu avec un seul arbre. La forêt aléatoire reste donc sensible au déséquilibre des données et a tendance à surapprendre ici.

D. AGRÉGATION D'ARBRES ADABOOST

Face aux performances décevantes de la forêt aléatoire, AdaBoost est une alternative d'agrégation reposant sur du *boosting* séquentiel. AdaBoost construit itérativement des arbres réduits à 1 nœud (*stumps*) qui corrigent séquentiellement les erreurs des prédécesseurs : chaque stump difficile reçoit un poids croissant pour l'itération suivante.

Les hyperparamètres à optimiser par GridSearchCV sont alors le nombre de stumps et la vitesse d'adaptation des poids.

```
Fitting 5 folds for each of 9 candidates, totalling 45 fits
Meilleurs hyperparamètres: {'ada__learning_rate': 0.1, 'ada__n_estimators': 200}

=== AdaBoost OPTIMAL ===
Accuracy Train: 0.740
Accuracy Test:  0.713
Score CV AUC:   0.766

        ========= Métriques AdaBoost ============
              precision    recall  f1-score   support

           0       0.86      0.71      0.78       210
           1       0.52      0.72      0.60        90

    accuracy                           0.71       300
   macro avg       0.69      0.72      0.69       300
weighted avg       0.75      0.71      0.72       300

AUC Test: 0.791

Matrice de confusion:
[[149  61]
 [ 25  65]]
Coût total (5*FN + FP): 186
```

Fig.II-10 Résultats d'AdaBoost

Le modèle AdaBoost optimal obtenu (200 stumps) présente un surapprentissage faible de 2,7% (bien meilleur que celui de la forêt aléatoire).

L'AUC Test de 0,791 indique une excellente capacité discriminante : sur 100 paires aléatoires (bon/mauvais payeur), 79,1% sont bien classés.

Sur la classe des « mauvais payeurs », le modèle obtient encore de meilleurs résultats : sur 100 individus classés « mauvais payeurs », 52 sont correctement identifiés, et sur 100 mauvais payeurs, 72 sont détectés.

Ce modèle génère un coût total de 186, ce qui est le coût le plus intéressant observé jusqu'à présent. Cela confirme la supériorité théorique du boosting face au bagging pour des données déséquilibrées.

E. MACHINE À VECTEURS DE SUPPORT NON LINEAIRE

La machine à vecteurs de support non-linéaire à noyau Gaussien (SVM RBF) projette les données dans un espace de dimension infinie via un noyau gaussien pour tracer une frontière de décision optimale séparant les bons et mauvais payeurs.

Les hyperparamètres à optimiser par GridSearchCV sont :

- C (paramètre de régularisation) qui contrôle le compromis entre exactitude d'entraînement et marge de séparation.
- gamma (paramètre d'échelle du noyau RBF) qui détermine la portée d'influence de chaque vecteur support.

```
Fitting 5 folds for each of 9 candidates, totalling 45 fits
Meilleurs hyperparamètres: {'svm__C': 1, 'svm__gamma': 0.01}
Score CV AUC: 0.763

=== SVM RBF OPTIMAL ===
Accuracy Train: 0.721
Accuracy Test:  0.697

         ========= Métriques SVM ============
               precision    recall  f1-score   support

            0       0.88      0.65      0.75       210
            1       0.50      0.80      0.61        90

     accuracy                           0.70       300
    macro avg       0.69      0.73      0.68       300
 weighted avg       0.77      0.70      0.71       300

AUC Test: 0.799

Matrice de confusion:
[[137  73]
 [ 18  72]]
Coût total (5*FN + FP): 163
```

Fig.II-11 Résultats de SVM RBF

La SVM RBF optimale obtenue (paramètre de régularisation égal à 1, paramètre d'échelle du noyau égal à 0,01) présente un faible surapprentissage (résultat excellent de 2,4%).

L'AUC Test de 0,799 indique une très bonne capacité discriminante : sur 100 paires aléatoires (bon/mauvais payeur), 79,9% sont bien classés.

Sur 100 individus classés « mauvais payeurs », 50 sont correctement identifiés, et sur 100 mauvais payeurs, 80 sont détectés, ce qui est un excellent rappel.

La matrice de confusion révèle 18 faux négatifs dans les données de test (mauvais payeurs non détectés) et 73 faux positifs, générant un coût total de 163, ce qui est à ce stade le meilleur coût obtenu jusqu'à présent.

F. PERCEPTRON MULTICOUCHES

Le perceptron multicouche (MLP) est un réseau de neurones artificiels à plusieurs couches qui capture des relations complexes et non-linéaires entre les variables grâce à des transformations successives et une optimisation par gradient.

Les hyperparamètres à optimiser par GridSearchCV sont :

- le nombre de neurones pour chacune des couches cachées ;
- alpha qui pénalise les poids élevés (et qui remplace le paramètre de poids utilisé jusqu'à présent) ;
- le pas d'apprentissage initial ;
- le choix de l'optimiseur (adam ou sgd).

```
 Meilleurs hyperparamètres: {'mlp__alpha': 0.01, 'mlp__hidden_layer_sizes':
(100, 50), 'mlp__learning_rate_init': 0.001, 'mlp__solver': 'sgd'}
Score CV AUC: 0.751

=== MLP OPTIMAL ===
Accuracy Train: 0.850
Accuracy Test:  0.773

         ========= Métriques MLP ============
               precision    recall  f1-score   support

            0       0.83      0.84      0.84       210
            1       0.62      0.61      0.62        90

     accuracy                           0.77       300
    macro avg       0.73      0.73      0.73       300
weighted avg       0.77      0.77      0.77       300

AUC Test: 0.815

Matrice de confusion:
[[177  33]
 [ 35  55]]
Coût total (5*FN + FP): 208
```

Fig.II-12 Résultats de MLP

Le MLP optimal obtenu est composé de deux couches cachées, une de 100 neurones, et une autre de 50 neurones, avec alpha = 0,01, un pas d'apprentissage de 0,01 et un optimiseur par descente de gradient stochastique. Ce modèle présente un surapprentissage modéré (écart train/test de 7,7%).

L'AUC Test est de 81,5%, ce qui est excellent, et le meilleur score obtenu.

Sur la classe des « mauvais payeurs », le modèle obtient des résultats corrects sur le jeu de test : sur 100 individus classés « mauvais payeurs », 62 sont correctement identifiés, et sur 100 mauvais payeurs, 61 sont détectés.

La matrice de confusion révèle 35 faux négatifs dans les données de test (mauvais payeurs non détectés) et 33 faux positifs, générant un coût total de 208.

CONCLUSION

Pour déterminer si un crédit est risqué ou non, plusieurs modèles de classification supervisée ont été entraînés et optimisés sur le jeu de données Statlog German Credit, en appliquant un même schéma de prétraitement et de validation croisée 5-fold (GridSearchCV) garantissant une comparaison équitable.

Les performances de ces modèles ont ensuite été évaluées sur un jeu de test indépendant, via l'AUC et une matrice de coût asymétrique.

L'ensemble des résultats obtenus peut être synthétisé dans le tableau ci-dessous :

Modèle	Coût	AUC	Rappel (1)	Précision (1)	Overfitting
AFD	260	79,8%	51%	61%	3,3%
Arbre CART	201	71,2%	74%	44%	3,7%
Forêt aléatoire	220	75,1%	66%	48%	10,3%
AdaBoost	186	79,1%	72%	52%	2,7%
SVM RBF	163	79,9%	80%	50%	2,4%
MLP	208	81,5%	61%	62%	7,7%

Fig.III Synthèse comparative

La **SVM RBF** domine les autres modèles avec un coût record de 163, grâce à un rappel exceptionnel de 80% sur les mauvais payeurs (il détecte 4 mauvais payeurs sur 5). Sa précision modérée de 50% signifie qu'il classe correctement la moitié des mauvais payeurs qu'il repère, mais ce choix privilégie la détection des risques plutôt que la perfection dans les cas positifs. Son surapprentissage minimal (2,4%) assure une excellente généralisation sur de nouvelles données.

Le modèle **AdaBoost** confirme l'intérêt du boosting séquentiel sur les données déséquilibrées avec un coût de 186 (juste derrière celui de SVM), une excellente généralisation de 2,7% et un très bon rappel de 72%.

Le **MLP** excelle en AUC (81,5%) grâce à sa capacité à capturer des relations complexes non-linéaires. Son surapprentissage modéré (7,7%) assure une bonne généralisation, mais son coût de 208.

L'arbre **CART**, sans être le plus performant des modèles, surprend par de bons résultats (3e du classement sur le coût et le rappel).

Le bagging de la **forêt aléatoire** échoue sur le déséquilibre des données : le surapprentissage est de 10,3% et le coût est avant dernier sur l'ensemble des modèles testés.

L'**AFD** est le dernier des modèles du classement, car limité par l'absence de corrélations linéaires fortes révélée en analyse exploratoire des données.

La SVM RBF est donc retenue pour son équilibre parfait entre coût, rappel et généralisation, validée par la matrice de coût métier bancaire.

Si les priorités évoluaient vers une discrimination pure, ne pénalisant plus les erreurs sur les faux-négatifs, le MLP deviendrait préférable.

REFERENCES

Cours et TD de l'UE RCP209, CNAM
Arnaud Breloy, Javiera Castillo Navarro, Marin Ferecatu

Source de la banque de données :
https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data

ANNEXES

Annexe A.1 - Liste des modalités –

Attribute 1 : Status of existing checking account (qualitative)
A11 : ... < 0 DM
A12 : 0 <= ... < 200 DM
A13 : ... >= 200 DM/salary assignments for at least 1 year
A14 : no checking account

Attribute 3 : Credit history (qualitative)
A30 : no credits taken/all credits paid back duly
A31 : all credits at this bank paid back duly
A32 : existing credits paid back duly till now
A33 : delay in paying off in the past
A34 : critical account/other credits existing (not at this bank)

Attribute 4 : Purpose (qualitative)
A40 : car (new)
A41 : car (used)
A42 : furniture/equipment
A43 : radio/television
A44 : domestic appliances
A45 : repairs
A46 : education
A47 : (vacation - does not exist?)
A48 : retraining
A49 : business
A410 : others

Attibute 6 : Savings account/bonds (qualitative)
A61 : ... < 100 DM
A62 : 100 <= ... < 500 DM
A63 : 500 <= ... < 1000 DM
A64 : .. >= 1000 DM
A65 : unknown/ no savings account

Attribute 7 : Present employment since (qualitative)
A71 : unemployed

A72 : ... < 1 year
A73 : 1 <= ... < 4 years
A74 : 4 <= ... < 7 years
A75 : .. >= 7 years

Attribute 9: Personal status and sex (qualitative)
A91 : male : divorced/separated
A92 : female : divorced/separated/married
A93 : male : single
A94 : male : married/widowed
A95 : female : single

Attribute 10 : Other debtors / guarantors (qualitative)
A101 : none
A102 : co-applicant
A103 : guarantor

Attribute 12 : Property (qualitative)
A121 : real estate
A122 : if not A121 : building society savings agreement/life insurance
A123 : if not A121/A122 : car or other, not in attribute6
A124 : unknown / no property

Attribute 14 : Other installment plans (qualitative)
A141 : bank
A142 : stores
A143 : none

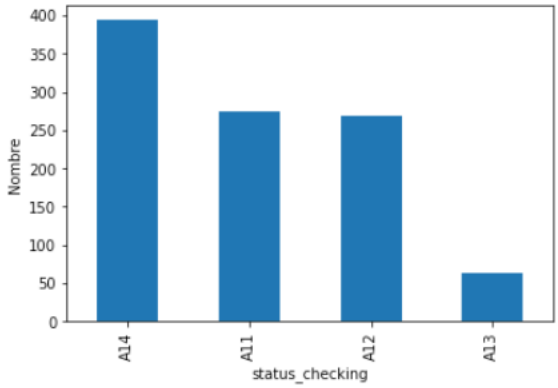
Attribute 15 : Housing (qualitative)
A151 : rent
A152 : own
A153 : for free

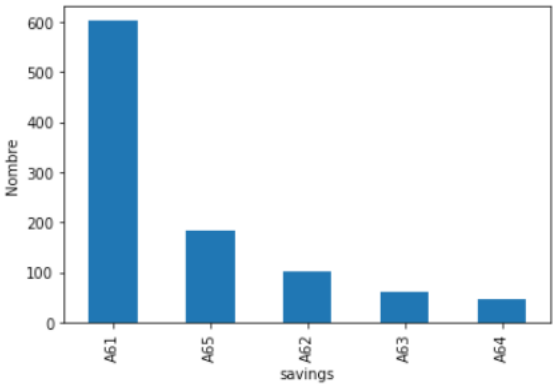
Attribute 17 : Job (qualitative)
A171 : unemployed/ unskilled - non-resident
A172 : unskilled - resident
A173 : skilled employee / official
A174 : management/ self-employed/highly qualified employee/ officer

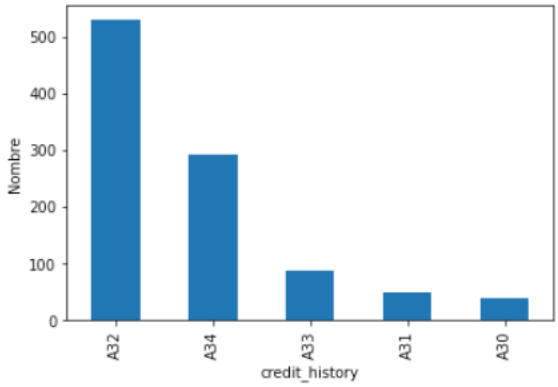
Attribute 19 : Telephone (qualitative)
A191 : none
A192 : yes, registered under the customers name

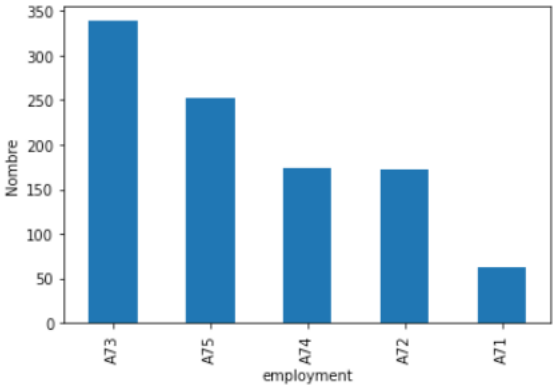
Attribute 20 : foreign worker (qualitative)
A201 : yes
A202 : no

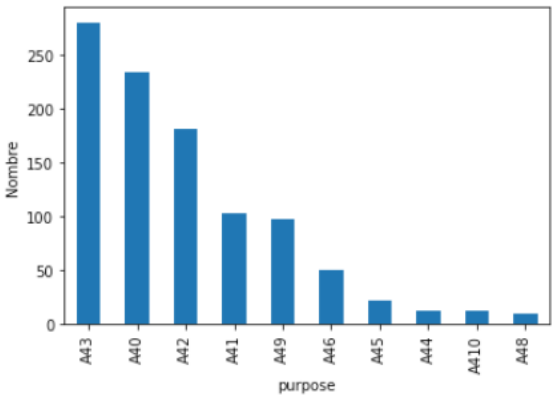
Annexe A.2 – Répartition des modalités selon les variables –

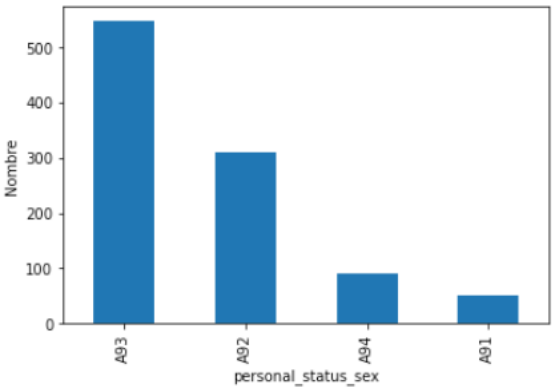


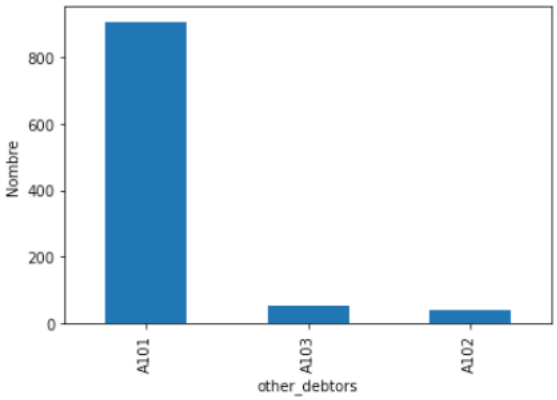


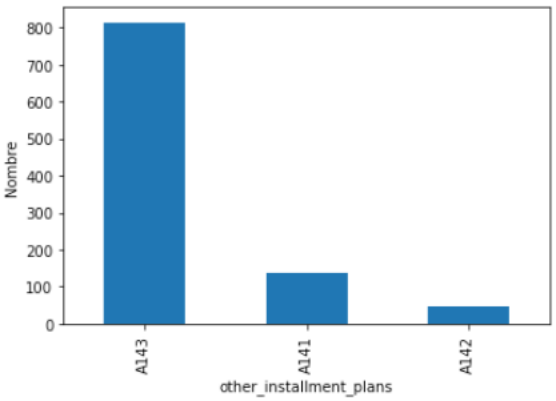


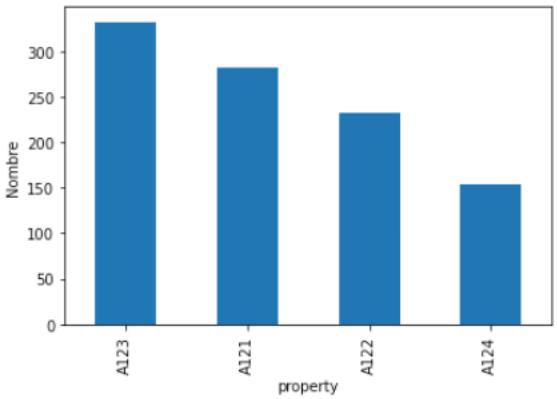


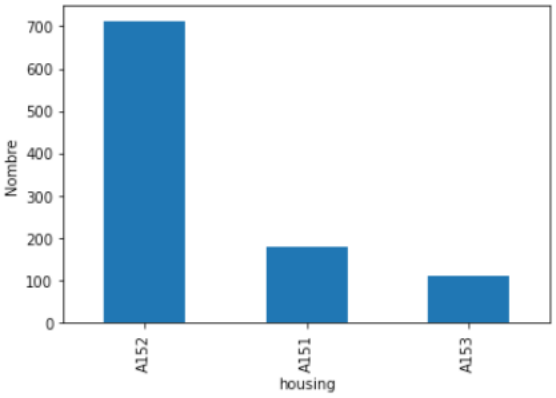


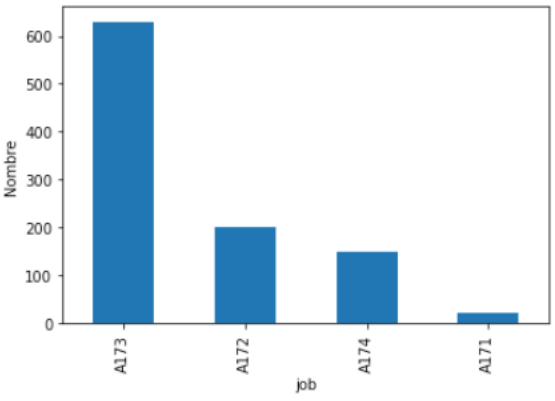


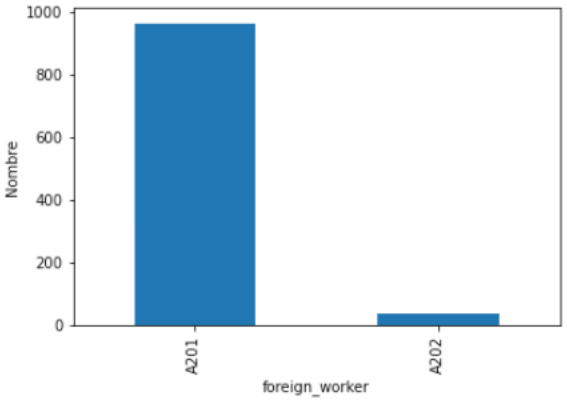

Nombre
0
200
400
600
800
A101
A103
A102
other_debtors

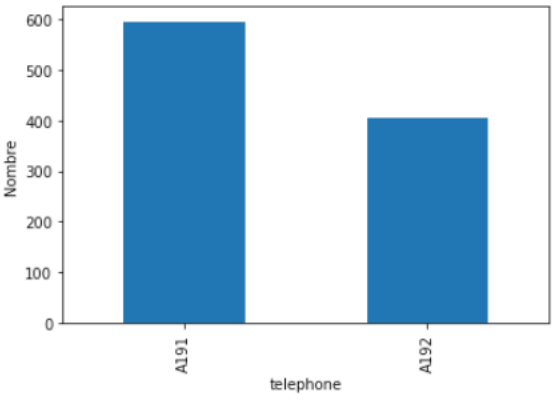

Nombre
0
100
200
300
400
500
600
700
800
A143
A141
A142
other_installment_plans


Nombre
0
50
100
150
200
250
300
A123
A121
A122
A124
property


Nombre
0
100
200
300
400
500
600
700
A152
A151
A153
housing


Nombre
0
100
200
300
400
500
600
A173
A172
A174
A171
job


Nombre
0
200
400
600
800
1000
A201
A202
foreign_worker


Nombre
0
100
200
300
400
500
600
A191
A192
telephone

Annexe A.3 – Fréquence des mauvais payeurs sur le top 3 des insights –