Université d'Antananarivo
Domaine Sciences et technologie
Département Mathématiques et informatique

Exemple sur :

Exploratory Data Analysis and Modeling

RANAIVOSON Lanto Mahaliana Finoana

Année Universitaire : 2024-2025

1 Contexte du Projet

Ce projet analyse les données d'une **campagne de marketing bancaire** menée par une institution financière portugaise. L'objectif principal est de **prédire si un client va souscrire à un dépôt à terme** (*term deposit*) suite à une campagne de télémarketing.

2 Description du Dataset (`bank.csv`)

Le dataset contient **4521 observations** avec **17 variables** décrivant les caractéristiques des clients et les détails des contacts de la campagne.

2.1 Variables démographiques

— **age** : Âge du client
— **job** : Type d'emploi (admin, blue-collar, entrepreneur, etc.)
— **marital** : Statut marital (married, divorced, single)
— **education** : Niveau d'éducation (primary, secondary, tertiary, unknown)

2.2 Variables financières

— **default** : Le client a-t-il un crédit en défaut ? (yes/no)
— **balance** : Solde annuel moyen du compte bancaire (en euros)
— **housing** : Le client a-t-il un prêt immobilier ? (yes/no)
— **loan** : Le client a-t-il un prêt personnel ? (yes/no)

2.3 Variables liées à la campagne

— **contact** : Type de communication (cellular, telephone, unknown)
— **day** : Dernier jour de contact du mois
— **month** : Dernier mois de contact de l'année
— **duration** : Durée du dernier contact (en secondes)
— **campaign** : Nombre de contacts effectués pendant cette campagne
— **pdays** : Nombre de jours depuis le dernier contact (-1 = jamais contacté)
— **previous** : Nombre de contacts avant cette campagne
— **poutcome** : Résultat de la campagne précédente (success, failure, unknown, other)

2.4 Variable cible

— **y** : Le client a-t-il souscrit à un dépôt à terme ? (yes/no)

3 Objectifs de l'Analyse

1. **Analyse Exploratoire des Données (EDA)** : Comprendre les distributions, les corrélations et les patterns présents dans les données
2. **Prétraitement** : Gérer le déséquilibre des classes, encoder les variables catégorielles et normaliser les variables explicatives
3. **Modélisation** : Entraîner et comparer huit algorithmes de classification

4. **Optimisation** : Sélectionner les trois meilleurs modèles et optimiser leurs hyperparamètres à l'aide de `GridSearchCV`
5. **Évaluation** : Comparer les performances à l'aide de métriques adaptées (Accuracy, Precision, Recall, F1-score)

3.1 Enjeux Business

L'identification des facteurs clés influençant la décision d'un client de souscrire à un dépôt à terme permettra à la banque de :
— Cibler plus efficacement les prospects
— Optimiser les coûts de la campagne marketing
— Améliorer le taux de conversion global

4 Modèles de classification utilisés

Dans ce projet, huit algorithmes ont été sélectionnés à partir de la bibliothèque `scikit-learn`. Ces modèles sont utilisés via leurs implémentations standards fournies par la bibliothèque, sans implémentation manuelle des algorithmes. Les modèles considérés sont les suivants :

— **Régression logistique** (`LogisticRegression`)
— **k plus proches voisins** (`KNeighborsClassifier`)
— **Machine à vecteurs de support** (`SVC`)
— **Naive Bayes gaussien** (`GaussianNB`)
— **Arbre de décision** (`DecisionTreeClassifier`)
— **Forêt aléatoire** (`RandomForestClassifier`)
— **AdaBoost** (`AdaBoostClassifier`)
— **Gradient Boosting** (`GradientBoostingClassifier`)

Ces modèles couvrent différentes approches de classification (linéaires, basées sur la distance, probabilistes et méthodes densemble), permettant une comparaison approfondie de leurs performances sur le jeu de données étudié.

5 Interprétation des résultats

Les résultats obtenus permettent de classer les modèles de machine learning selon leurs performances sur une tâche de classification binaire. Les métriques principales utilisées sont l'exactitude (*Accuracy*), la précision (*Precision*), le rappel (*Recall*) et le score F1 (*F1-score*).

5.1 Classement des modèles selon le score F1

Le score F1, étant une moyenne harmonique de la précision et du rappel, est particulièrement pertinent pour évaluer la performance globale des modèles sur des données potentiellement déséquilibrées.

1. **Random Forest** (F1 = 0.975) : meilleur modèle avec une excellente précision (0.951) et un rappel parfait (1.000), ce qui indique une très bonne capacité à identifier correctement toutes les classes.

2. **Decision Tree** (F1 = 0.958) : performant également avec un rappel parfait mais une précision légèrement inférieure à Random Forest.
3. **SVM (RBF)** (F1 = 0.914) : bon compromis entre précision (0.876) et rappel (0.955), mais moins performant que les modèles à base d'arbres.
4. **KNN** (F1 = 0.910) : rappel parfait mais précision plus faible (0.834), ce qui suggère un certain nombre de faux positifs.
5. **Gradient Boosting** (F1 = 0.877) : performances correctes mais nettement inférieures aux deux premiers modèles.
6. **Adaboost et Logistic Regression** (F1 0.805) : performances similaires et modérées.
7. **Gaussian NB** (F1 = 0.635) : modèle le moins performant avec un déséquilibre marqué entre précision (0.775) et rappel (0.538).

6 Optimisation des trois meilleurs modèles

Suite à l'analyse comparative initiale, les trois modèles les plus performants (**Random Forest**, **Decision Tree** et **SVM avec noyau RBF**) ont été optimisés à l'aide de `GridSearchCV` avec validation croisée (CV=3) et la métrique F1-score comme critère d'évaluation.

6.1 Analyse des résultats d'optimisation

6.1.1 1. Random Forest - Performance optimale

— **Paramètres optimaux** :
  — `max_depth` : None (pas de limitation)
  — `min_samples_split` : 2
  — `n_estimators` : 200 (parmi les valeurs testées)
— **Amélioration** : Légère augmentation du F1-score (de 0.975015 à 0.975610)
— **Caractéristique** : Maintient un rappel parfait (1.0000) tout en améliorant légèrement la précision
— **Implication** : Le modèle fonctionne mieux sans restriction de profondeur et avec des divisions minimales de 2 échantillons

6.1.2 Decision Tree - Stabilité remarquable

— **Paramètres optimaux** :
  — `max_depth` : None
  — `min_samples_split` : 2
— **Performance** : F1-score stable à 0.958084
— **Observation** : Les paramètres optimaux sont identiques à Random Forest, suggérant que les données supportent des arbres profonds non contraints
— **Avantage** : Modèle plus simple et interprétable que Random Forest avec une performance seulement légèrement inférieure

6.1.3 SVM (RBF) - Amélioration significative

— **Amélioration** : Gain notable du F1-score (de 0.913876 à 0.936830)
— **Impact** : Augmentation de l'accuracy de 2.375 points (de 91.0% à 93.375%)
— **Tendance** : Amélioration de la précision au détriment d'un léger recul du rappel
— **Signification** : Le modèle SVM est plus sensible à l'optimisation des hyperparamètres

Comparaison avant/après optimisation

Modèle	F1 avant	F1 après	Gain
Random Forest	0.975015	0.975610	+0.000595
Decision Tree	0.958084	0.958084	ś0.000000
SVM (RBF)	0.913876	0.936830	**+0.022954**

Table 1 – Gain de performance après optimisation

Conclusions de l'optimisation

1. **Random Forest reste le meilleur modèle** : Malgré une amélioration minime, il conserve sa position de leader avec le F1-score le plus élevé (0.9756).
2. **SVM bénéficie le plus de l'optimisation** : L'amélioration la plus significative est observée pour SVM (+0.023 en F1-score), démontrant l'importance cruciale de ses hyperparamètres.
3. **Decision Tree montre une robustesse** : Sa performance stable suggère que ses paramètres par défaut étaient déjà proches de l'optimum.
4. **Tendance générale** : Les trois modèles conservent un rappel très élevé (0.9825), indiquant une excellente capacité à identifier les vrais positifs.

7 Recommandations finales

— **En production** : Déployer **Random Forest** avec les paramètres optimisés pour une performance maximale.
— **Pour l'interprétation** : Utiliser **Decision Tree** qui offre un bon équilibre entre performance et explicabilité.
— **En cas de contraintes** : **SVM** représente une alternative viable après optimisation, notamment si des considérations de vitesse d'entraînement ou de scalabilité sont importantes.
— **Pour maintenance** : Mettre en place un suivi régulier des performances et une réoptimisation périodique des hyperparamètres.

L'optimisation a confirmé la supériorité de Random Forest tout en améliorant substantiellement SVM, validant ainsi l'approche de sélection et d'optimisation systématique des modèles.

8 Conclusion

Implémenter le modèle Random Forest optimisé pour cibler précisément les clients à fort potentiel, en concentrant les efforts sur les contacts de qualité et les clients avec historique positif.

Cette solution permet d'anticiper une augmentation significative du taux de conversion tout en réduisant les coûts de campagne, optimisant ainsi le retour sur investissement marketing.