RAPPORT DU PROJET ANALYSE DE PREDICTION DES RISQUES DE CREDIT

Nom : NADEGE KANGNI-SOUKPE

1. Introduction

Dans le cadre de la gestion des crédits bancaires, il est essentiel d'évaluer correctement le risque que représentent les emprunteurs. Ce projet a pour objectif de concevoir une base de données relationnelle autour de la gestion des prêts et de développer un modèle prédictif capable d'estimer le niveau de risque de chaque client.

2. Structure de la Base de Données (SQL)

Pour structurer les données, plusieurs tables ont été créées :

- **Clients** : contient les informations personnelles et financières des clients
  - `client_id`, `nom`, `prénom`, `age`, `sexe`, `situation_pro`, `revenu_annuel`, `historique_credit`
- **Prets** : représente les demandes de prêt associées à chaque client
  - `pret_id`, `client_id`, `montant`, `taux_interet`, `duree_mois`, `statut_paiement`
- **Paiements** : enregistre les paiements effectués pour chaque prêt
  - `paiement_id`, `pret_id`, `date_paiement`, `montant_paye`, `statut`
- **Transactions** : contient les opérations bancaires diverses
  - `transaction_id`, `client_id`, `type_transaction`, `montant`, `date_transaction`, `solde_apres`

Des optimisations ont été apportées via :

- Index sur `client_id` et `pret_id`
- Fonctions fenêtres pour analyser les paiements
- Jointures SQL pour analyser les comportements clients

3. Préparation des Données (Python + SQL)

Les données ont été extraites de la base SQLite et traitées avec `pandas`.

- Aucune valeur manquante n'a été détectée dans `X_train`, `X_test`, `y_train` ni `y_test`.
- Les variables explicatives utilisées sont :
  - `age`, `revenu_annuel`, `montant`, `taux_interet`, `duree_mois`
- La variable cible est : `statut_paiement` avec les valeurs :
  - `-1.0` : défaut de paiement
  - `0.0` : paiement en cours / normal
  - `1.0` : remboursement complet

Les données ont été standardisées et réparties en ensembles d'entraînement/test.

4. Modélisation Prédictive

Le modèle choisi est la **Régression Logistique**, en raison de sa simplicité et de son interprétabilité.

- Les classes ont été équilibrées grâce à **SMOTE** (sur-échantillonnage des classes minoritaires).
- Performance obtenue :

	precision	recall	f1-score	support
-1.0	0.50	0.50	0.50	4
0.0	0.50	0.50	0.50	6
1.0	0.71	0.71	0.71	7
accuracy			0.59	17
macro avg	0.57	0.57	0.57	17
weighted avg	0.59	0.59	0.59	17

- Le modèle prédit avec une précision correcte malgré un jeu de données limité.

5. Score de Risque et Intégration SQL

Le score de risque a été calculé grâce à `predict_proba` de la régression logistique, donnant un score entre 0 et 1.

Exemple de sortie :

```
   age  revenu_annuel  montant  taux_interet  duree_mois  y_test  y_pred  \
0   35        45000.0  10000.0          3.50          36     1.0    -1.0
1   59        33081.0  39952.0          1.80          18    -1.0     0.0
2   76        80115.0  30443.0          5.46          36     1.0     1.0
3   51        55201.0  10046.0          5.93          35     0.0     1.0
4   42        86518.0  17564.0          1.95          20    -1.0    -1.0

   score_risque
0      0.269752
1      0.200789
2      0.480244
3      0.349156
4      0.234710
```

Ces résultats ont été intégrés dans une nouvelle table SQL `ScoresRisque` pour exploitation ultérieure.

6. Analyse des Résultats

- Le modèle fonctionne de manière raisonnable sur les classes `1.0` (bons payeurs)
- Des erreurs apparaissent notamment sur les classes `-1.0` (défauts) dues à la petite taille de l'échantillon
- Les prédictions sont cohérentes avec les profils de risque
- Le score permet une hiérarchisation des dossiers clients selon leur probabilité de défaillance

7. Conclusion et Perspectives

Ce projet a permis de créer une base de données complète et un pipeline de détection du risque de crédit avec un modèle de régression logistique.

Améliorations possibles :

- Test d'autres modèles (Random Forest, XGBoost)
- Ajout de nouvelles variables (historique de paiements, comportement transactionnel)
- Création d'un dashboard interactif pour visualiser les scores de risque

8. Annexes

- Scripts SQL de création de tables
- Scripts Python (traitement, modélisation)
- Extraits de données et visualisations