aivancity
PARIS-CACHAN

Coding for AI & Data Science

Projet Prediction Bancaire

Répartition des Tâches :

- **Levis Junior** : Data Scientist - Exploration et Préparation des Données
- **Hergi Diangue :** Data Engineer - Création, Entraînement, et Évaluation des Modèles

Sous la supervision de : Mr Alioune Nar SAMBE

Année académique :

2024-2025

Sommaire

Contexte du Projet

Le projet vise à prédire si un client souscrira à un dépôt à terme en se basant sur des données collectées lors de campagnes de télémarketing. Ces informations permettent à la banque de maximiser le taux de conversion tout en réduisant les coûts d'opportunité et d'améliorer l'efficacité des campagnes.

Étapes Réalisées

Étape 1 : Chargement des Données et Nettoyage

- Chargement des données brutes à partir du fichier bank.csv.

- Suppression des doublons pour garantir l'unité des observations.

- Gestion des valeurs manquantes en utilisant le mode pour chaque colonne.

- Conversion de la variable cible (y) en format binaire : 1 pour "yes" et 0 pour "no".

Étape 2 : Préparation des Données

- Séparation des données en caractéristiques (X) et variable cible (y).

- Identification des colonnes numériques et catégorielles.

- Prétraitement avec ColumnTransformer :

  - StandardScaler pour les colonnes numériques.

  - OneHotEncoder pour les colonnes catégorielles.

Étape 3 : Équilibrage des Données

- Utilisation de SMOTE pour équilibrer les classes minoritaires en créant des échantillons synthétiques.

- Stratégie d'échantillonnage à 70 % de la classe majoritaire pour garantir un équilibre suffisant.

Étape 4 : Division des Données

- Division en ensembles d'entraînement (70 %) et de test (30 %).

Étape 5 : Entraînement et Évaluation des Modèles

- Modèles testés :
  - Régression Logistique
  - Random Forest
  - Gradient Boosting
  - SVC
- Métriques d'évaluation :
  - Précision
  - ROC AUC Score
  - Matrice de confusion
  - Classification Report
- Comparaison des performances des modèles avec des graphiques.

Étape 6 : Optimisation des Hyperparamètres

- Optimisation de Random Forest avec GridSearchCV :
  - Meilleurs paramètres identifiés : n_estimators, max_depth, min_samples_split.
- Ré-entraînement du modèle avec les hyperparamètres optimaux.

Étape 7 : Validation et Généralisation

- Validation croisée sur l'ensemble d'entraînement pour vérifier la robustesse.
- Évaluation finale sur l'ensemble de test.

Étape 8 : Calcul du Bénéfice Net Business

- Métriques financières :
  - Gain par client converti (VP) = 1 000 €
  - Coût d'opportunité manqué (FN) = 500 €

- Gain pour une tentative évitée (VN) = 100 €

- Coût d'une tentative infructueuse (FP) = 50 €

- Calcul :

- Utilisation de la matrice de confusion pour déterminer le bénéfice net.

Étape 9 : Sauvegarde du Modèle

- Sauvegarde du meilleur modèle optimisé (Random Forest) dans best_model.pkl.

Étape 10 : Déploiement avec Streamlit

- Déploiement d'une interface interactive :

- Entrée utilisateur (âge, emploi, état civil, etc.).

- Pour déployer mon application Streamlit avec Anaconda, j'ai commencé par créer un environnement avec la commande `conda create -n model python=3.9` et je l'ai activé avec `conda activate model `. Ensuite, j'ai installé Streamlit en utilisant `pip install streamlit` et préparé mon fichier source (par exemple `Déploiement.py`). J'ai lancé l'application localement avec `streamlit run Déploiement.py`. Pour un déploiement public, je peux envisager des plateformes comme Streamlit Cloud ou Heroku.

- Affichage de la prédiction et du bénéfice net estimé.

Problèmes Rencontrés et Solutions

Problème 1 : Désynchronisation des Colonnes lors de la Prédiction

- Cause : Les colonnes encodées différaient entre l'entraînement et la prédiction.

- Solution : Les transformations ont été standardisées à travers toutes les étapes en sauvegardant le préprocesseur dans preprocessor.pkl.

Problème 2 : Modèle Non Ajusté

- Cause : Erreur NotFittedError lors de l'utilisation du ColumnTransformer.

- Solution : Vérification et ajustement des objets avant de sauvegarder.

Problème 3 : Absence de Fichiers Nécessaires

- Cause : Certains fichiers (préprocesseur, modèle) manquaient lors du déploiement.

- Solution : Tous les objets critiques (préprocesseur, modèle) ont été sauvegardés et chargés correctement.

Problème 4 : Erreur lors de l'Affichage des Scores de Performance et de la Courbe ROC

- Cause : Utilisation incorrecte des données pour calculer la courbe ROC et les scores de performance, menant à des erreurs de taille d’échantillon.

- Solution : Modification des méthodes pour utiliser des échantillons cohérents entre les prédictions et les valeurs réelles, garantissant ainsi la validité des métriques affichées.

Conclusions

- Meilleur Modèle : Le Random Forest optimisé s'est avéré être le modèle le plus performant avec des scores élevés en précision et AUC.

- impact Business : L'intégration du bénéfice net a montré un potentiel important pour maximiser la rentabilité des campagnes marketing.

- Déploiement : Le tableau de bord Streamlit offre une interface interactive et intuitive pour les utilisateurs.

Ce projet a permis de développer un modèle prédictif robuste capable de prévoir si un client souscrira à un dépôt à terme. Grâce à l'utilisation d'algorithmes de machine learning avancés, à l'optimisation des hyperparamètres et à l'équilibrage des données, nous avons réussi à atteindre des résultats satisfaisants en termes de précision et de ROC AUC. Le déploiement via Streamlit a offert une solution conviviale pour permettre aux utilisateurs finaux d'explorer les prédictions et de comprendre les bénéfices financiers associés. L'intégration du calcul de bénéfice net a également montré l'importance de prendre en compte des aspects financiers dans l'évaluation de la performance des modèles prédictifs. En conclusion, ce projet a démontré une approche complète, depuis l'acquisition des données jusqu'au déploiement final, avec un impact significatif sur l'optimisation des campagnes marketing bancaires.