Challenge Data Science : Prédiction de la Réussite des Startups Tech

Dataset

Nous utiliserons le dataset "Company Success Prediction" de scikit-learn qui contient des informations sur 1000 startups tech avec leur statut après 5 ans (Succès/Échec/Acquisition).

startup_success_data.csv

Description du Dataset

Le dataset contient les colonnes suivantes :

- `funding_rounds` : Nombre de levées de fonds
- `total_funding` : Montant total levé
- `team_size` : Taille de l'équipe au lancement
- `tech_stack_size` : Nombre de technologies utilisées
- `patents` : Nombre de brevets déposés
- `burned_rate` : Taux de consommation du capital mensuel
- `revenue_growth` : Croissance du chiffre d'affaires (%)
- `market_size` : Taille du marché ciblé (en millions $)
- `competitors` : Nombre de concurrents directs
- `social_media_score` : Score d'engagement sur les réseaux sociaux
- `client_retention` : Taux de rétention client (%)
- `pivot_count` : Nombre de pivots stratégiques
- `regulatory_score` : Score de conformité réglementaire

Variable cible (`status`) :

- 0 : Échec (60% des cas)
- 1 : Succès (25% des cas)
- 2 : Acquisition (15% des cas)

Particularités du dataset :

- 20% de valeurs manquantes dans `revenue_growth` et `client_retention`
- Classes déséquilibrées
- Outliers dans `total_funding` et `team_size`

Objectifs du Challenge

1. Analyse Exploratoire

- Analyser la distribution des variables
- Étudier les corrélations
- Visualiser les relations entre variables

- Identifier et traiter les outliers
- Proposer au moins 3 visualisations originales et pertinentes
- Tirer des conclusions business des analyses

2. Préparation des Données

- Gérer les valeurs manquantes de manière créative
- Créer de nouvelles features pertinentes (feature engineering)
- Encoder les variables catégorielles si nécessaire
- Normaliser/standardiser les données
- Gérer le déséquilibre des classes
- Documenter et justifier chaque choix

3. Modélisation

Développer deux modèles :

1. Random Forest
    - Optimiser les hyperparamètres
    - Gérer la profondeur des arbres
    - Utiliser la validation croisée
2. XGBoost
    - Optimiser les hyperparamètres
    - Implémenter l'early stopping
    - Gérer le learning rate

Pour chaque modèle :

- Évaluer avec plusieurs métriques (accuracy, f1-score, ROC-AUC)
- Analyser l'importance des features
- Fournir une analyse des erreurs
- Comparer les performances des deux modèles

4. Déploiement

Créer une application web avec Streamlit ou Flask qui permet :

- Charger de nouvelles données
- Prétraiter automatiquement ces données
- Faire des prédictions avec les deux modèles
- Afficher les probabilités pour chaque classe
- Visualiser l'importance des features
- Comparer les prédictions des deux modèles