Projet Data & Machine Learning

Titre du Projet :

Prédiction de la sinistralité et modulation tarifaire via Machine Learning

Réalisé par :

Alan-Long Le Menelec

Clément Duran

Formation :

Diplôme Universitaire Big Data,

Data Science et analyse des risques sous Python

Etablissement :

Université de Montpellier - Faculté d'économie

Année universitaire :

2024-2025

Contexte

Dans la continuité de l'analyse économétrique, cette seconde phase vise à exploiter des méthodes de **Machine Learning** pour :

1. **Prédire la probabilité** qu'un contrat déclare au moins un sinistre (has_claim).

2. **Estimer le montant** d'un sinistre (claim_amount) conditionnel à l'existence d'un sinistre en vue d'une application tarifaire. Les données proviennent de la base consolidée df_assurance (variables assurantielles et climatiques) .

a) Préparation des données

1. **Sélection des variables**
    - Variables assurantielles : caractéristiques du contrat (pol_bonus, pol_coverage, pol_pay_freq, etc.), conducteur (drv_age1, drv_age_lic1, drv_sex1), véhicule (vh_din, vh_value, vh_weight), géolocalisation (geo_cluster).
    - Variables dérivées :
        - vh_value_per_weight = valeur / poids,
        - drv_age_gap = âge conducteur – ancienneté permis.
    - Variables climatiques agrégées par département : temp_moyenne, precip_moy.
2. **Encodage et normalisation**
    - Encodage one-hot des qualitatives (pol_coverage, pol_pay_freq, vh_fuel, etc.).
    - Standardisation des variables numériques pour les réseaux de neurones et Tree-based models.
3. **Échantillonnage**
    - Jeu complet pour la classification (has_claim).
    - Sous-ensemble des contrats sinistrés (claim_amount > 0) pour la régression du montant.

b) Modélisation de la sinistralité (has_claim)

Pipeline

1. **Gestion du déséquilibre** : application de **SMOTETomek** pour équilibrer minorité/majorité.

2. **Modèle** : **XGBoost Classifier**

    - Recherche d'hyperparamètres via **RandomizedSearchCV** (profondeur, learning_rate, n_estimators, etc.).

    - Early stopping sur validation.

3. **Évaluation** sur test set :

    - **AUC (ROC)** ≃ 0.77

    - **Accuracy** ≃ 85 %

    - **Precision/Recall**, **F1-score** (classification report)

    - **Matrice de confusion**

Interprétation

- **SHAP Beeswarm** pour ranking des variables explicatives :

    1. Pol_sit_duration

    2. pol_coverage

    3. drv_drv2

    4. drv_sex2, etc.

- Les valeurs SHAP confirment que l'ancienneté du contrat, le niveau de couverture, la présence d'un second conducteur, les caractéristiques démographiques et de paiement ainsi que des variables climatiques jouent logiquement sur la probabilité de sinistre.

c) Modélisation du montant (`claim_amount`)

Pour modéliser le montant des sinistres on a donc logiquement filtré sur les contrats avec claim_amount > 0.

1. XGBoost Regressor

- Modèle de boosting d'arbres.
- Optimisation des hyperparamètres par RandomizedSearchCV (20 itérations, 3 folds).
- Bon compromis entre performance et rapidité.
- Résultat :
  - RMSE ≈ 1998 €
  - MAE ≈ 795 €
  - R^2 ≈ -0.03
- Modèle rapide et robuste mais limité par la forte variabilité des montants.

2. LightGBM Regressor

- Modèle basé sur des arbres de décision optimisé pour la rapidité.
- Même stratégie d'optimisation que XGBoost.
- Résultat :
  - RMSE ≈ 1997 €
  - MAE ≈ 802 €
  - R^2 ≈ -0.03
- Performances similaires à XGBoost parfois légèrement en dessous.

3. Réseau de neurones (MLP) optimisé

- Architecture dense profonde : 5 couches (192 ➔96 ➔64 ➔32 ➔1).
- Normalisation des données, Dropout (0.1), early stopping.
- Résultat :
  - RMSE ≈ 1961 €
  - MAE ≈ 772 €
  - R^2 ≈ 0.01
- Meilleure performance globale du projet sur les données brutes.

4. Réseau de neurones (MLP) – transformé (log1p)

- Même architecture que le modèle précédent.
- La variable claim_amount est transformée avec log1p pour atténuer l'impact des sinistres extrêmes.
- Les prédictions sont retranscrites dans l'échelle réelle (expm1) pour l'évaluation.
- Résultat :
  - RMSE ≈ 2059 €
  - MAE ≈ 739 €
  - R^2 ≈ -0.10

- Ce modèle a produit des prédictions plus lissées souvent moins précises (notamment sur les gros montants) mais qui fluctue énormément et donne parfois de meilleurs résultats.

5. Conclusion régression

En raison de l'asymétrie extrême et de la rareté des gros sinistres, le réseau dense peine à généraliser ; même si la transformation logarithmique réduit légèrement les erreurs moyennes, le signal demeure insuffisamment linéaire pour être correctement exploité.

d) Visualisations clés

- **ROC curve** (pour XGBoost).
- **SHAP Beeswarm** (explicabilité classification).
- **Scatter plots** y_true vs y_pred (régression brute & log).
- **Histogrammes des résidus** (distribution des erreurs).
- **Courbes d'apprentissage** (loss/val_loss par époque).
    -

Conclusion :

Cette phase de Machine Learning a permis de déployer un pipeline complet intégrant à la fois une étape de classification et une étape de régression. Le modèle de classification a obtenu un score AUC satisfaisant pour la sinistralité tout en restant compréhensible grâce aux analyses SHAP. En revanche, la tentative de prédiction du montant des sinistres à l'aide de réseaux de neurones a mis en évidence la difficulté de capturer la forte asymétrie des coûts.