Data challenge 2024

Équipe Easy Code 2

Sujet Enedis

Présentation du problème

Aujourd'hui, pour évaluer l'état de son réseau électrique, Enedis envoie un hélicoptère survoler un tiers du réseau électrique chaque année, couvrant ainsi la totalité du réseau en 3 ans. Le but de ce survol est de détecter des anomalies, qui seront ensuite étudiées par une personne physique qui décidera de l'envoi ou non d'un drone pour confirmer l'anomalie. Si elle est confirmée, une maintenance est effectuée.

Notre projet vise à optimiser le trajet de l'hélicoptère afin de couvrir des zones où des anomalies ont une probabilité plus élevée d'apparaître.

Nous chercherons ainsi à prédire l'arrivée d'anomalies avérées, nécessitant une maintenance.

Modélisation

Variable	Type	Unité
Présence d'une anomalie avérée SLD	À expliquer	Indicatrice (0 ou 1)
Longueur de la section fragile SLD	Explicative	mètres
Longueur en plan aléa climatique SLD	Explicative	mètres
Âge du tronçon	Explicative	années
Nombre d'anomalies observées SLD	Explicative	unités
Nombre d'incidents SLD	Explicative	unités

* SLD = "sur le départ"

Nous avons construit la variable à expliquer sur le critère suivant : une anomalie est avérée s'il y a eu une maintenance sur le tronçon, et si cette maintenace est ultérieure au dernier passage de l'hélicoptère.

Démarche adoptée

Gestion des données déséquilibrées

Le principal défi de ce challenge était de gérer l'important déséquilibre dans les données :

```
##
##     0     1
## 71039  3175
```

0 codant l'absence d'anomalies et 1 leur présence, la classe minoritaire - qui nous intéresse - ne représente que 4% du jeu de données.

Nous avons testé les méthodes de sous- et sur-échantillonnage suivantes : * sous-échantillonnage "naïf" (suppression au hasard d'individus de la classe majoritaire) * sous-échantillonnage avec algorithme SMOTE * sur-échantillonnage avec algorithme SMOTE

Nous avons retenu la technique de sous-échantillonnage SMOTE, qui présente le meilleur compris coût/efficacité (beucoup plus court que le sur-échantillonnage pour 3% d'efficacité en moins).

Choix des algorithmes

Nous avons ici un problème de classification sur lequel nous avons testé 3 modèles : régression logistique, arbre simple, forêt aléatoire basique et forêt aléatoire avec probabilités.

L'algorithmes présentant les meilleurs résulats est la forêt aléatoire.

Centrage-réduction des données

Suite à notre choix de sous-échantillonner et d'utiliser une forêt aléatoire, nous avons également choisi de centrer-réduire les données pour des résultats plus fiables.

Résultats

Algorithme random forest

```
## Ranger result
##
## Call:
##  ranger(Class ~ ., data = under_train, mtry = 3)
##
## Type:                             Classification
## Number of trees:                  500
## Sample size:                      5000
## Number of independent variables:  5
## Mtry:                             3
## Target node size:                 1
## Variable importance mode:         none
## Splitrule:                        gini
## OOB prediction error:             12.60 %
```

L'erreur de prédiction OOB est assez élevée.

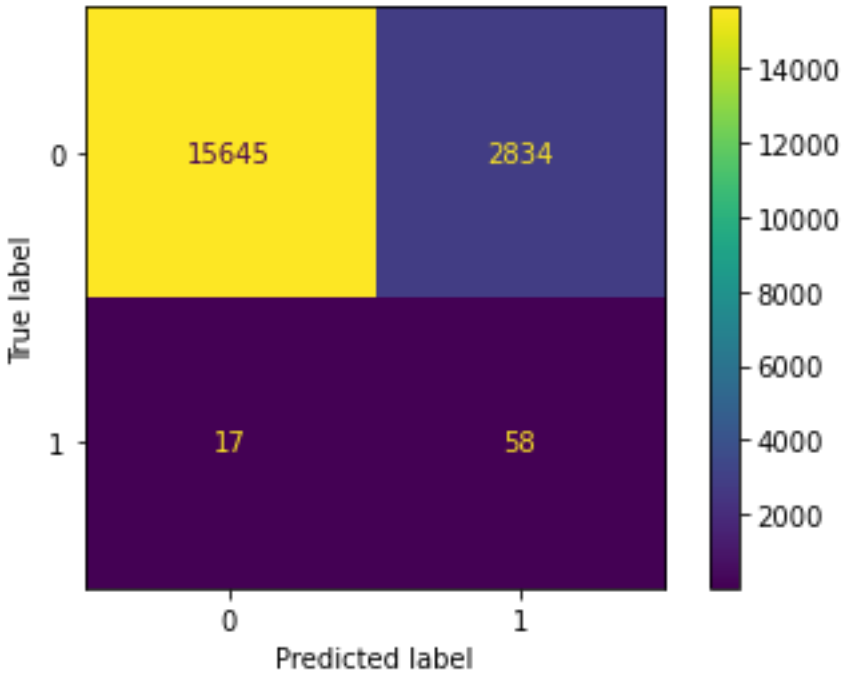
En appliquant la forêt sur notre échantillon d'entraînement, nous obtenons :

```
##
##       0   1
##   0 542  52
##   1 109 647
```

en colonne, la valeur prédite ; en ligne, la valeur réelle

L'algorithme est très performant sur notre sous-échantillon avec 629 anomalies correctement détectées, 88 faux positifs et 70 anomalies non détectées.

Application aux données Enedis



Notre meilleur algorithme a les propriétés suivantes :

- très bonne détection des anomalies avérées
- très grand nombre de faux positifs
- très faible nombre de faux négatifs

La MSE pâtit des faux positifs (elle vaut environ 0.15) et le F1-score se retrouve à 0.03.

Nos autres modèles présentent une MSE bien meilleure, mais donnent très peu de vrais positifs, voire aucun.

Nous n'avons également pas pu implémenter la limite des 25 km : les probabilités d'anomalie rendues par la forêt ne se traduisent pas dans la réalité. En l'absence de score viable, nous ne pouvons pas sélectionner de tronçons prioritaires.

Conclusion

Notre modèle est retenu par le taux important de faux positifs et le trop long kilométrage identifié comme prioritaire.

Cependant, en tirant les 25 km à survoler parmi ceux conseillés par le programme, on sera quand même plus efficace qu'à piocher au hasard.

Des corrections à cet algorithme pourraient inclure l'ajout d'une fonction de score pertinente afin de diminuer le taux de faux positifs.