Rapport – Appentissage supervisé pour des données avec classes déséquilibrées

Florent Giauna (AMSD) et Zewei Lin (MLSD)

Dataset : Credit fraud

Tableau – Meilleur score obtenu par algorithme en fonction des méthodes de resampling utilisées

Algorithme	PR AUC (sans resampling)	AUC (SMOTE)	AUC (ADASYN)	AUC (Random Undersmpling)	AUC (TomekLinks)
DecisionTree	0.718	0.998	0.914	0.951	0.911
LogisticRegression	0.734	0.989	0.952	0.98	0.98
LinearSVC	0.715	NA	NA	NA	0.972
SVC (linear)	NA	NA	NA	0.978	NA
SVC (rbf)	NA	NA	NA	0.981	NA
SVC (poly)	NA	NA	NA	0.971	NA
SVC (sigmoid)	NA	NA	NA	0.931	NA
RandomForest	0.843	1	0.997	0.976	0.977
XGBoost	0.753	1	0.999	0.977	0.973

NA : indiqué pour les variations de SVC non implémentées en raison des capacités de calcul à notre disposition. LinearSVC n'est pas implémenté lorsque nous avons pu implémenter les SVC avec kernel. Par ailleurs seul le score du meilleur kernel après GridSearch est reporté, les autres scores sont ceux obtenus par défaut.

En prenant en compte le temps d'exécution pour choisir la meilleure combinaison d'algorithme et d'approche de resampling nous avons opté pour XGBoost avec Random Undersampling. Le rapport de classification obtenu est le suivant :

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.89      0.81      0.84        98

    accuracy                           1.00     56962
   macro avg       0.94      0.90      0.92     56962
weighted avg       1.00      1.00      1.00     56962
```

Cette combinaison donne les meilleurs résultats obtenus sur ce jeu de données. Toutes les métriques sont au dessus du deuxième meilleur modèle, un autre XGBoost par ailleurs plus complexe. Il peut donc être choisi pour passer en production.

Dataset : Bank marketing

Tableau – Meilleur score obtenu par algorithme en fonction des méthodes de resampling utilisées

Algorithme	PR AUC (sans resampling)	AUC (SMOTE)	AUC (ADASYN)	AUC (Random Undersmpling)	AUC (TomekLinks)
DecisionTree	0.548	0.956	0.952	0.902	0.919
LogisticRegression	0.552	0.920	0.902	0.908	0.921
LinearSVC	0.546	0.919	0.9	NA	0.921
SVC (linear)	NA	NA	NA	0.899	NA
SVC (rbf)	NA	NA	NA	0.905	NA
SVC (poly)	NA	NA	NA	0.907	NA
SVC (sigmoid)	NA	NA	NA	0.797	NA
RandomForest	0.582	0.995	0.994	0.916	0.931
XGBoost	0.593	0.993	0.993	0.919	0.931

NA : indiqué pour les variations de SVC non implémentées en raison des capacités de calcul à notre disposition. LinearSVC n'est pas implémenté lorsque nous avons pu implémenter les SVC avec kernel. Par ailleurs seul le score du meilleur kernel après GridSearch est reporté, les autres scores sont ceux obtenus par défaut.

En prenant en compte le temps d'exécution pour choisir la meilleure combinaison d'algorithme et d'approche de resampling nous avons opté pour RandomForest sans resampling. Le rapport de classification obtenu est le suivant :

```
              precision    recall  f1-score   support

           0       0.97      0.84      0.90      7310
           1       0.39      0.83      0.53       928

    accuracy                           0.84      8238
   macro avg       0.68      0.83      0.72      8238
weighted avg       0.91      0.84      0.86      8238
```

Le deuxième meilleur modèle, un XGBoost avec TomekLinks donne des résultats très légèrement meilleurs. Il a moins de faux négatifs mais plus de faux positifs. Le F1 score sur la classe majoritaire est meilleur mais celui sur la classe minoritaire est moins bon (0.47 contre 0.53). La classe minoritaire est moins bien prédite.

Le modèle RandomForest sans resampling devrait être privilégié. Le risque est de passer à côté de potentiels souscripteurs et de consacrer trop de temps de la campagne à des clients qui ont plus de chance de ne pas souscrire.

Dataset : Employee attrition

Tableau – Meilleur score obtenu par algorithme en fonction des méthodes de resampling utilisées

Algorithme	PR AUC (sans resampling)	AUC (SMOTE)	AUC (ADASYN)	AUC (Random Undersmpling)	AUC (TomekLinks)
DecisionTree	0.848	0.98	0.97	0.854	0.946
LogisticRegression	0.401	0.776	0.744	0.728	0.743
LinearSVC	NA	NA	NA	NA	NA
SVC (linear)	0.310	0.776	0.745	0.738	0.654
SVC (rbf)	0.967	1	0.999	0.961	0.975
SVC (poly)	0.528	0.791	0.749	0.695	0.834
SVC (sigmoid)	0.271	0.695	0.625	0.686	0.581
RandomForest	0.976	1	1	0.973	0.992
XGBoost	0.964	0.998	0.999	0.964	0.978

NA : indiqué pour les variations de SVC non implémentées en raison des capacités de calcul à notre disposition. LinearSVC n'est pas implémenté lorsque nous avons pu implémenter les SVC avec kernel. Par ailleurs seul le score du meilleur kernel après GridSearch est reporté, les autres scores sont ceux obtenus par défaut.

En prenant en compte le temps d'exécution pour choisir la meilleure combinaison d'algorithme et d'approche de resampling nous avons opté pour XGBoost ave ADASYN. Le rapport de classification obtenu est le suivant :

```
              precision    recall  f1-score   support

           0       0.91      0.95      0.93       736
           1       0.66      0.51      0.58       141

    accuracy                           0.88       877
   macro avg       0.79      0.73      0.75       877
weighted avg       0.87      0.88      0.87       877
```

Cette combinaison de sampling et de modèle donne les meilleurs résultats obtenus sur ce jeu de données.

Toutefois il y a un peu plus de faux négatifs qu'avec notre modèle RandomForest sans resampling. Un arbitrage devrait être pris en fonction de ce critère. Dans le cas de démissions il vaut peut être mieux privilégier la combinaison prédisant le plus petit nombre de faux négatifs pour éviter une pénurie d'employés.