

Fondamentaux de l'apprentissage automatique : 8INF867

TP2: Dimensionnalité et anomalies

Membres du groupe:

- Hélène Barbillon - **BARH30530200**
- Samia Carchaf - **CARS05550300**
- Chaimaa Oulmalme - **OULC12590000**

Table des matières:

| | |
|--|----------|
| I. Contraintes de fonctionnements | 2 |
| II. Explication du travail | 2 |
| Contexte | 2 |
| Visualisations | 2 |
| Classification | 3 |
| Undersampling | 4 |
| Oversampling | 5 |
| III. Observations et résultats | 5 |

I. Contraintes de fonctionnements

Le projet est en python, dans des notebook jupyter. Il utilise les librairies: scikitlearn, unbalanced-learn, umap, ucimlrepo, T-SNE, ainsi que pandas, matplotlib et numpy. Les dépendances nécessaires pour faire fonctionner le projet se trouvent dans le fichier `requirements.txt`, et toutes les instructions pour les installer dans un environnement virtuel python sont indiquées dans le fichier `README.md` du projet.

II. Explication du travail

Contexte

Ce projet porte sur le dataset "[Coverttype](#)", représentant 7 types de couvertures forestières en fonction de plusieurs paramètres tels que l'altitude, l'exposition, la pente, l'ombrage des collines, le type de sol, et bien d'autres.

Ce dataset contient 581 012 instances, et 54 attributs (en plus de CoverType). Il est déséquilibré, la répartition des données est inégale:

| Cover_Type | |
|------------|--------|
| 2 | 283301 |
| 1 | 211840 |
| 3 | 35754 |
| 7 | 20510 |
| 6 | 17367 |
| 5 | 9493 |
| 4 | 2747 |

En raison de la grande taille de cette base de données, et suite à des difficultés d'exécution de certains algorithmes, nous l'avons réduite dans certains cas, de façon aléatoire avec la fonction `test_train_split` de scikit-learn.

Nous cherchons à voir l'influence de l'oversampling et de l'undersampling sur cette base de données, grâce à des visualisations et de la classification.

Notre code est réparti dans 3 notebook jupyter dans le dossier `src` : un pour l'oversampling, un pour l'undersampling, et un pour les données brutes.

Visualisations

Nous avons utilisé les algorithmes UMAP et T-SNE pour pouvoir visualiser dans un graphique le dataset. Ces algorithmes permettent de réduire à 2 dimensions (ou 3 pour un graphique en 3D) le dataset, afin de mieux visualiser la répartition de CoverType.

L'algorithme UMAP a plusieurs hyperparamètres configurables comme `n_neighbors`, `min_dist`, et `n_components`. `N_components` est égal à 2 pour réduire le dataset à 2

dimensions, ou 3 pour de la 3D. Après plusieurs tests en modifiant les hyperparamètres, nous avons choisi de laisser ceux par défaut, puisqu'ils convenaient.

Classification

Nous avons implémenté un modèle de classification basé sur un **Random Forest** pour prédire les classes du jeu de données "Forest Cover Type". Les données ont été prétraitées en réduisant la taille (0.9) pour accélérer l'entraînement tout en maintenant la distribution des classes. Un pipeline a été créé pour standardiser les données et entraîner le modèle. Les résultats montrent une précision globale (**accuracy**) de **95,04%** et un F1-score pondéré de **95,01%**, indiquant des performances élevées et équilibrées sur la plupart des classes. Les classes majoritaires, comme les types de couvertures 1 et 2, ont obtenu les meilleures précisions et rappels, tandis que les classes minoritaires, comme 4 et 5, ont montré des performances légèrement inférieures.

```
Résultats :  
Accuracy : 0.9518  
F1-Score : 0.9516  
  
Rapport de classification détaillé :
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 1 | 0.96 | 0.94 | 0.95 | 62917 |
| 2 | 0.95 | 0.97 | 0.96 | 84141 |
| 3 | 0.94 | 0.96 | 0.95 | 10619 |
| 4 | 0.92 | 0.86 | 0.89 | 816 |
| 5 | 0.93 | 0.74 | 0.83 | 2819 |
| 6 | 0.93 | 0.90 | 0.91 | 5158 |
| 7 | 0.98 | 0.94 | 0.96 | 6091 |
| accuracy | | | 0.95 | 172561 |
| macro avg | 0.94 | 0.90 | 0.92 | 172561 |
| weighted avg | 0.95 | 0.95 | 0.95 | 172561 |

Après avoir appliqué un **oversampling** sur les données, les performances du modèle de classification basé sur Random Forest se sont encore améliorées. Les classes comme les types de couvertures 4, 5, et 7, qui avaient précédemment des scores légèrement inférieurs, affichent des rappels et F1-scores meilleurs, atteignant même **1.00** dans certains cas.

| Classification Report: | | | | |
|------------------------|-----------|--------|----------|---------|
| | precision | recall | f1-score | support |
| 1 | 0.97 | 0.96 | 0.97 | 84990 |
| 2 | 0.97 | 0.96 | 0.96 | 84990 |
| 3 | 0.99 | 0.99 | 0.99 | 84990 |
| 4 | 1.00 | 1.00 | 1.00 | 84991 |
| 5 | 0.99 | 1.00 | 1.00 | 84991 |
| 6 | 0.99 | 1.00 | 0.99 | 84990 |
| 7 | 1.00 | 1.00 | 1.00 | 84991 |
| accuracy | | | 0.99 | 594933 |

| | | | | |
|----------------|------|------|------|--------|
| accuracy | | | 0.99 | 594933 |
| macro avg | 0.99 | 0.99 | 0.99 | 594933 |
| weighted avg | 0.99 | 0.99 | 0.99 | 594933 |
| Accuracy: 0.99 | | | | |

Après avoir effectué l'**undersampling**, les performances du modèle de classification basé sur Random Forest ont légèrement diminuées en termes de précision globale (accuracy) et de F1-score pondéré, atteignant respectivement **86,29%** et **86,14%**.

Rapport de classification détaillé :

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 1 | 0.79 | 0.79 | 0.79 | 824 |
| 2 | 0.78 | 0.70 | 0.74 | 824 |
| 3 | 0.86 | 0.81 | 0.83 | 824 |
| 4 | 0.93 | 0.97 | 0.95 | 824 |
| 5 | 0.89 | 0.94 | 0.92 | 824 |
| 6 | 0.83 | 0.88 | 0.85 | 824 |
| 7 | 0.94 | 0.96 | 0.95 | 825 |
| accuracy | | | 0.86 | 5769 |
| macro avg | 0.86 | 0.86 | 0.86 | 5769 |
| weighted avg | 0.86 | 0.86 | 0.86 | 5769 |

Résultats :

Accuracy : 0.8629

F1-Score : 0.8614

Undersampling

L'undersampling est une technique de prétraitement des données utilisée pour traiter les problèmes de déséquilibre de classe dans les ensembles de données. Lorsqu'une classe est largement sous-représentée par rapport à une autre, cela peut biaiser les modèles d'apprentissage automatique. L'undersampling consiste à réduire la taille de la classe majoritaire en supprimant aléatoirement des exemples de cette classe, de sorte que l'équilibre entre les classes soit rétabli.

- Répartition des classes du dataset avant undersampling:

```
Cover_Type
2      283301
1      211840
3       35754
7       20510
6       17367
5        9493
4        2747
Name: count, dtype: int64
```

- Répartition des classes du dataset après undersampling:

```
Cover_Type
1        2747
2        2747
3        2747
4        2747
5        2747
6        2747
7        2747
Name: count, dtype: int64
```

Oversampling

L'oversampling consiste à créer artificiellement des données, pour réduire l'effet de déséquilibre du dataset. La méthode utilisée est SMOTE, avec comme hyperparamètre k neighbors, qui est à 5 par défaut. Voici la répartition des classes avant et après oversampling:

| Cover_Type | |
|------------|--------|
| 2 | 283301 |
| 1 | 211840 |
| 3 | 35754 |
| 7 | 20510 |
| 6 | 17367 |
| 5 | 9493 |
| 4 | 2747 |

| Cover_Type | |
|------------|--------|
| 5 | 283301 |
| 2 | 283301 |
| 1 | 283301 |
| 7 | 283301 |
| 3 | 283301 |
| 6 | 283301 |
| 4 | 283301 |

III. Observations et résultats

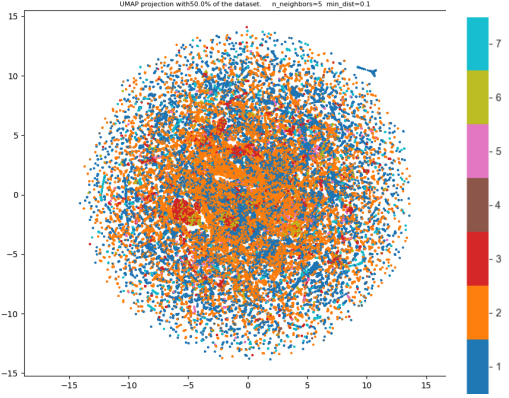
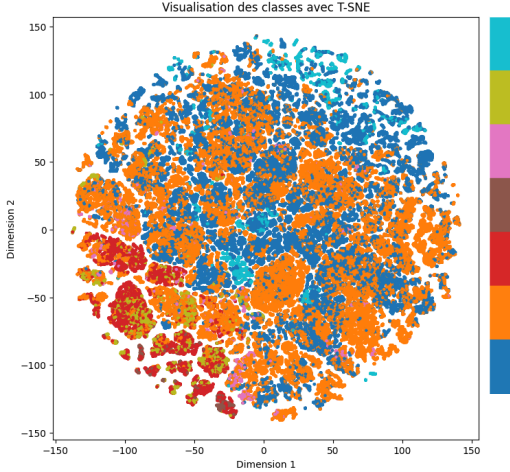
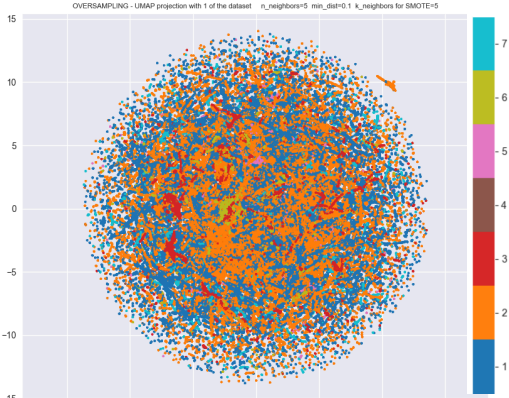
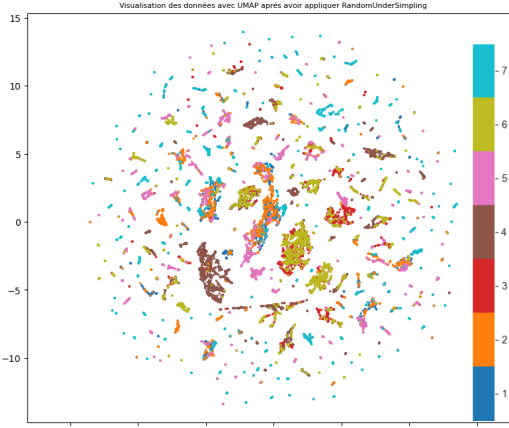
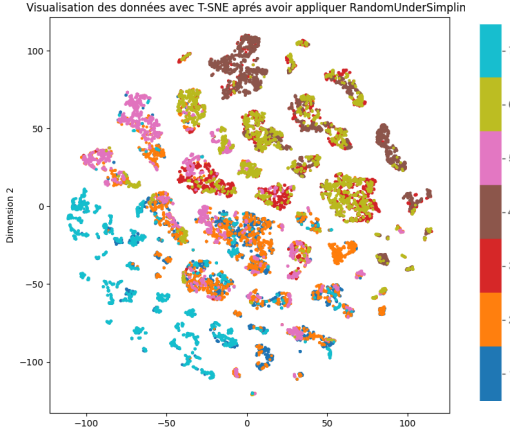
Nos résultats globaux sont rassemblés dans le tableau sur la prochaine page. Les graphiques se trouvent également dans le dossier `images` du projet en meilleure qualité.

On observe que T-SNE produit des graphes plus faciles à interpréter que UMAP, mais avec un temps de calcul plus élevé.

On remarque que l'oversampling génère des données artificielles pour équilibrer les classes, ce qui peut améliorer la précision, au risque de provoquer un sur-apprentissage.

L'undersampling quant à lui réduit la taille des données en supprimant des instances, ce qui accélère le traitement, mais entraîne une perte d'information qui diminue la précision.

Le choix de l'utilisation d'une méthode ou d'une autre dépend au final de l'utilisation de la base de données.

| Dataset | UMAP | T-SNE | Scores |
|-------------------------|---|--|--|
| raw (50% of dataset) |  |  | Accuracy = 0.9344 F1-Score = 0.9339 |
| Oversampled |  | Pas de visualisation pour autant de données (trop long) | Accuracy = 0.98 F1-score = 0.9829 |
| Undersampled |  |  | Accuracy = 0.8629 F1-Score = 0.8614 |