

Téléchargement et Préparation du Dataset Covertypes

Présentation de la Base de Données Covertypes

La base de données **Covertypes** provient de l'[UCI Machine Learning Repository](#). Elle est utilisée pour **classer les types de couvert forestier** à partir de mesures cartographiques (sol, altitude, pente, distance aux points d'eau, etc.).

Caractéristiques du Dataset

- **Nombre d'observations** : 581 012
- **Nombre de variables** : 54 (features continues et binaires)
- **Nombre de classes** : 7 types de couverture forestière (1 à 7)
- **Problème à résoudre** : Classification supervisée

Les classes ne sont **pas équilibrées**, ce qui peut influencer la performance des modèles de classification. Les proportions des classes dans l'ensemble original sont les suivantes :

Classe	Type de forêt	Effectif	Proportion (%)
1	Épicéa	211 840	36.5
2	Pin	283 301	48.8
3	Peuplier	35 754	6.2
4	Bouleau	2 747	0.5
5	Érable	9 493	1.6
6	Hêtre	17 367	3.0
7	Mélèze	18 510	3.2

Objectif de l'Échantillonnage

Pourquoi réduire la taille du dataset ?

Le dataset **Covertypes** est **très grand** (581 012 individus). En raison du **temps de calcul important**, nous avons décidé d'utiliser un **échantillon**

plus petit, tout en conservant la distribution des classes.

Comment gérer le déséquilibre des classes ?

Certaines classes sont **très majoritaires** (ex : **Pin** et **Épicéa** représentent à eux seuls **85% des données**), tandis que d'autres sont **très minoritaires** (ex : **Bouleau** à seulement **0.5%**).

Nous avons appliqué un **échantillonnage différencié** :

- **Sous-échantillonnage des classes majoritaires** (Épicéa et Pin) → **5% de leurs effectifs d'origine**
- **Sur-échantillonnage relatif des classes minoritaires** (Peuplier, Bouleau, Érable, Hêtre, Mélèze) → **20% de leurs effectifs d'origine**

Nous ne supprimons pas totalement le déséquilibre, car nous souhaitons tester nos modèles dans des conditions réalistes, où certaines classes restent plus rares que d'autres.

L'échantillon obtenu comptera un peu plus de 40 000 individus, dont 60% seront réservés à l'entraînement des modèles, 20% à la validation des hyperparamètres, et 20% aux tests.

Téléchargement et Préparation des Données

```
import os
import pandas as pd
from sklearn.model_selection import train_test_split

# Correction du chemin d'accès au fichier dataset
file_path = "../data/covertypes.gz" # On remonte d'un dossier
    ↪ pour atteindre data/

if not os.path.exists(file_path):
    raise FileNotFoundError(f" Le fichier {file_path} est
    ↪ introuvable. Vérifiez son emplacement.")

# Chargement des données
column_names = [f'Feature_{i}' for i in range(1, 55)] +
    ↪ ['Cover_Type']
data = pd.read_csv(file_path, header=None, names=column_names,
    ↪ compression='gzip')

# Échantillonnage différencié
sampling_rates = {1: 0.05, 2: 0.05, 3: 0.2, 4: 0.2, 5: 0.2, 6:
    ↪ 0.2, 7: 0.2}
sampled_data = pd.concat([
```

```

        data[data['Cover_Type'] ==
↪   cls].sample(frac=sampling_rates[cls], random_state=42)
        for cls in data['Cover_Type'].unique()
    ]).reset_index(drop=True)

# Division en train/val/test
train_data, temp_data = train_test_split(sampled_data,
↪   test_size=0.4, random_state=42,
↪   stratify=sampled_data['Cover_Type'])
val_data, test_data = train_test_split(temp_data, test_size=0.5,
↪   random_state=42, stratify=temp_data['Cover_Type'])

# Sauvegarde des fichiers dans data/
train_data.to_csv("../data/covertime_train.csv", index=False)
val_data.to_csv("../data/covertime_val.csv", index=False)
test_data.to_csv("../data/covertime_test.csv", index=False)

print(" Fichiers covertime_train.csv, covertime_val.csv et
↪   covertime_test.csv créés avec succès dans data/")

Fichiers covertime_train.csv, covertime_val.csv et covertime_test.csv créés avec succès da

```