Projet App Meteo

Cahier des charges global

- Développer une application python permettant la lecture de données météo, leur manipulation ainsi que leur visualisation avec une mission globale donnée.

Les Missions:

1. Prévision des risques sur cumuls de pluie

- Fournir des cumuls sur differents intervalles (ex: 1H, 6H 12H et 24H)
- Créer un risque en fonction d'une combinaison de seuil (ex : RR1 > 10mm).

2. Humidex et Windchill

- Création des indicateurs de température ressentie.
  Le premier est utilisé en période estivale. Le second en période hivernale.

3. Facteur de charge éolien

- Calculer le facteur de charge prévu d'une éolienne en fonction de la force du vent.

4. Synthése de données météorologiques

- Mettre en place l'équivalent de l'application web de MétéoFrance pour une zone donnée (ou autre par ex bulletin météo à portée nationale).

Pour chacune de ces missions, il faudra développer des visualisations spécifiques qui pourront afficher les résultats à partir de jeux de données variées et pour des zones géographiques précises.

Les données à dispositions

Vous disposerez de trois types de données :

- des données de prévision déterministe AROME à 1.3km de résolution (grille EUR1S100) pour des échéances allant jusqu'à 50h.
  Nomenclature : `arome_forecast_{day.strftime('%Y%m%d%H')}.nc`
- des données de prévision ensembliste AROME à 2.5 km de résolution (grille EUR1S040) pour des échéances allant jusqu'à 50H. L'ensemble possède 25 membres différents. Chaque membre est dans un fichier spécifique.
  Nomenclature :
  `arome_forecast_{day.strftime('%Y%m%d%H')}_mb_{str(X).zfill(3)}.nc` où X est le numéro du membre.
- des masques (remplis de 1 et de NaN) délimitant différents types de zones géographiques (avec ou non un nom).

Contenu des fichiers AROME

```
>>> ds = xr.open_dataset("arome_forecast_2024102800.nc")
>>> ds
<xarray.Dataset> Size: 3GB
Dimensions:    (time: 50, latitude: 1400, longitude: 1400)
Coordinates:
  * time       (time) datetime64[ns] 400B 2024-10-28T01:00:00
  * latitude   (latitude) float64 11kB 53.4 53.39 53.38 ... 3
  * longitude  (longitude) float64 11kB -6.0 -5.99 -5.98 ...
Data variables:
    tirf       (time, latitude, longitude) float32 392MB ...
    t2m        (time, latitude, longitude) float32 392MB ...
    r2         (time, latitude, longitude) float32 392MB ...
    prmsl      (time, latitude, longitude) float32 392MB ...
    u10        (time, latitude, longitude) float32 392MB ...
    v10        (time, latitude, longitude) float32 392MB ...
    efg10      (time, latitude, longitude) float32 392MB ...
    nfg10      (time, latitude, longitude) float32 392MB ...
```

- tirf : Pluie cumulée depuis le début de la prévision (en mm)
- r2 : Humidité relative (en %)
- prmsl : Pression ramenée au niveau de la mer (en Pa)
- u10/v10 : composantes du vent à 10m (en m/s)
- efg10/nfg10 : Composantes rafales à 10 m (en m/s)

Les masques

Trois types de masques sont disponibles (sur la France Métropolitaine) :

- Les masques de zones climatiques homogènes. Il sont dénommés `zones_clim_{X}.nc` où `X` varie en fonction de la zone.
- Les masques de redécoupage de département (zones sympos).
  Ils sont dénommés `sympo_{Y}.nc` où Y est le département.
- Des masques sur des Bassins Versants. Pour ce dernier jeu de masques, deux niveaux emboîtés sont fournis.
  - `bassin_level1.nc` : Contient les 24 bassins versant sur la métropole
  - `bassin_level2_zone_{X}.nc` : Contient les bassins versant de la sous-zone `{X}`.

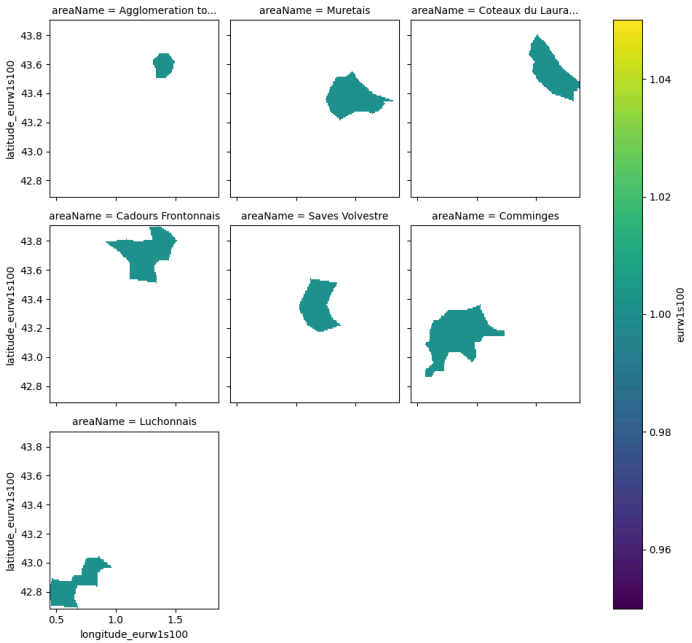
Chaque fichier contients plusieurs masques sur une zones données.

Les grilles

Chaque fichier de masques contient l'emprise des zones géographiques pour différentes grilles.
Avec les données AROME nous n'utiliseront que les grilles `eurw1s100` et `eurw1s40`.

Visualisation des Masques (Zones sympo pour le 31)

```
import xarray as xr
import matplotlib.pyplot as plt
path = "MYPATH/masks/Sympo/sympo_31.nc"
ds = xr.open_dataset(path)
ds = ds.set_coords("areaName") # Met le nom en coord
# Change de coordonnée
ds = ds["eurw1s100"].swap_dims({"id":"areaName"})
# Affiche l'ensemble des masques du fichier (en fonction
# de la coordonée areaName).
ds.plot(col="areaName",col_wrap=3)
plt.show()
```



Module 1 : Configuration du Projet et de l'Environnement

1. **Création de l'environnement virtuel** :
    - Créez un environnement virtuel Python pour ce projet et activez-le.
    - Installez toutes les bibliothèques nécessaires (`xarray` , `matplotlib` , `argparse` , `pyyaml` , etc.) en utilisant `pip` .

2. **Création de la structure des dossiers du projet** :

Nom_de_mon_application/
├── /sources/
│ ├── `__init__`.py
│ ├── data_loader.py
│ ├── data_manipulator.py
│ ├── visualizer.py
│ └── cli.py
├── input.yaml
├── .gitignore
├── requirements.txt
└── README.md

- Ne créer que les choses en **GRAS** !!

- Assurez-vous d'inclure un fichier `requirements.txt` contenant toutes les dépendances du projet.(liste des bibliothèques)
- Ajoutez au '.gitignore' la ligne '**pycache**/' ainsi que '*.nc'.

3. **Initialisation de Git** :
    - Initialisez un référentiel Git pour le projet.
    - Effectuez un premier commit pour sauvegarder la structure de base du projet.

Objectifs :

- Créer un environnement de travail propre et bien structuré.
- Apprendre à utiliser `git` pour versionner le code.

Module 2 : Chargement des Données Météo avec Xarray

Cahier des charges :

1. **Loader** :
    - Créez, au travers d'une classe, un objet Loader capable de :
        - prendre en entrée un chemin vers un fichier NetCDF et en faire un dataset xarray.
        - prendre en entrée une liste de fichier NetCFD et en faire un dataset xarray.
        - Ajoutez la possibilité de prétraiter les données en sélectionnant certaines partie des données via un fichier de masque extérieur .

1. **Testez la classe** :
    - Créez un fichier d'exemple pour tester le bon fonctionnement de la classe en chargeant un fichier local, et en prétraitant les données.

Module 3 : Manipulation et Analyse des Données

Créez un object capable de manipuler vos données en fonction des besoins de votre mission.

Vous pouvez commencer par discuter en groupe et écrire le détail des fonctionnalité que vous allez devoir implémenter pour mener à bien vos objectifs.