INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

METEO FRANCE

Prédiction du cumul de pluie journalier dans le quart Nord-Ouest de la France

Défi IA 2022 - AI Frameworks

Célia DULUC
Thomas NIVELET
Elisa ESCANEZ
Sébastien CASTETS

5e année – Mathématiques appliquées

7 janvier 2022
Année universitaire 2021/2022

Université de Toulouse

Introduction

Objectif : Prédire la précipitation totale sur une journée pour une station donnée.

Notations :

- Y : variable à prédire = « Ground_truth »
- X : variables explicatives

Plan

I. Présentation des données et pre-processing

II. Idées d'implémentation

III. Méthode finale retenue

I. Présentation des données et pre-processing

Données historiques ($X_{station}$)

- Données réelles observées chaque heure
- Identification de la station : Id
- 6 indicateurs météorologiques: ff, t, td, hu, dd, precip

X_{train}	26%
X_{test}	28%

Part de NaN dans les données $X_{station}$ brutes

1. Remplissage par les valeurs les plus proches (méthodes `bfill` et `ffill` de la fonction Python `fillna`) **uniquement pour les valeurs d'un même jour et d'une même station.**
2. **Agrégation/Groupement** (`groupby`) des indicateurs météo aux 24 heures sur une journée :
   - **Somme** pour les précipitations
   - **Moyenne** pour les autres indicateurs
3. La variable « month » traitée comme une catégorie.

I. Présentation des données et pre-processing

Données de prévision de Météo-France ($X_{forecast}$)

- 3 modèles physiques : AROME 2D, ARPEGE 2D et 3D
- Prévisions faites par météo France sur des maillages (AROME plus fin que ARPEGE) à chaque heure.
- Modèles 2D : Prévisions de 9 variables météo: "ws", "p3031", "u10", "v10", "t2m", "d2m", "r", "tp", "msl »
  - « tp » = total précipitation
- Modèles 3D : prévision de la pression à différentes altitudes (7 variables)

1. Détermination **des k points les plus proches** de chaque station.
2. Pour chaque jour et chaque station, **récupération des données météo** en ces k points.
   1. Pour « tp »: valeur prise à $h = 24$
   2. Sinon moyenne sur la journée
3. **Interpolation des données** par pondération inverse à la distance.

$$u(x) = \frac{\sum_{k=0}^{K} w_k\,(x)^{\,p}\,u_k}{\sum_{k=0}^{K} w_k\,(x)^{p}}$$

w_k : fonction de pondération ($w_k(x) = 1/d(x, x_k)$)
d : distance euclidienne
x_k : point du maillage (connu)
p : paramètre de puissance, degré de lissage (ici $p = 1$, donne plus d'importance aux points les plus proches) $u_k = u(x_k)$

I. Présentation des données et pre-processing

Construction d'un unique jeu de données

- **Concaténation** de $X_{station}$ et $X_{forecast}$ et Y
- Au sein de chaque ligne :
  - Données **historiques** du jour $J-1$,
  - **Prévision** du jour J,
  - Variable cible : volume de précipitations réellement observé le jour J.

Traitement des données manquantes

- Lignes dont la réponse (*Ground_truth*) était manquante **supprimées** du jeu d'entrainement :
  Apprentissage sur données « pures »

- Imputation du reste des donnés manquantes par *MissForest*

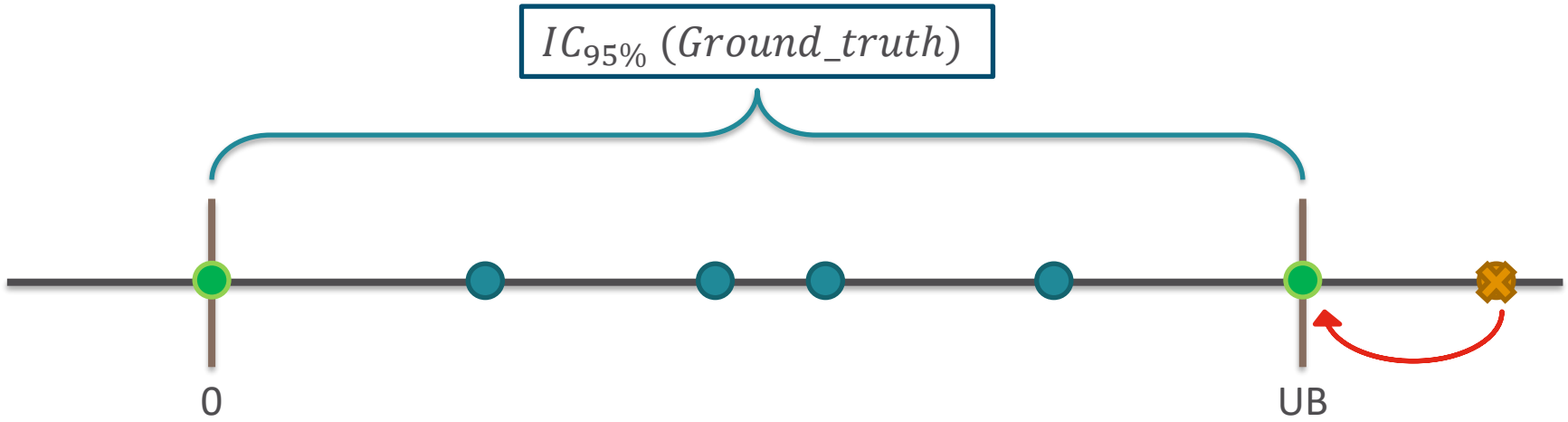
< 5%	[5% ; 10%[	[10% ; 25%[	> 25%
9	18	0	4

Répartition des pourcentages de NaN en fonction des variables météo avant la *MissForest*

I. Présentation des données et pre-processing

Traitement final

- Objectif : Limiter l'effet de valeurs extrêmes (potentiellement aberrantes) sur la prédiction
- **Restriction** des valeurs de la réponse (*Ground_truth*) à celles comprises dans un IC à 95%



- Normalisation des données

Nombre de variables : 43
Taille du train : 162 000 x 43
Taille du test: 85140 x 43

II. Idées d'implémentation

Utilisation d'un classifieur

Rappel du problème
- Prédiction des précipitations accumulées sur une journée
- Problème de régression (variable quantitative continue)

Constat
Beaucoup de valeurs observées nulles pour la variable à prédire

Idée
Opérer une classification avant la régression

⊕ Erreur exactement nulle pour les prédictions bien classées dans la classe (0)

⊖ Beaucoup de prédictions nulles pour des valeurs observées non nulles
Résultats moins bons que par la régression directe

Démarche

1. Problème de classification
- Une classe pour les précipitations nulles (0)
- Une classe pour les précipitations non nulles (1)

2. Problème de régression
- Appliquer un régresseur sur les données prédites dans la classe (1)
- Régresseur entraîné pour les précipitations observées non nulles.

3. Concaténation des résultats de classification et de régression

II. Idées d'implémentation

Utilisation de modèles existants

Constat
Modèles météo déjà existants (AROME et ARPEGE)

Idée
Corriger les biais de ces modèles

Démarche

1. Prédiction de l'erreur commise par ces modèles (AROME ou ARPEGE)

2. Ajout de l'erreur prédite à la valeur prédite par Météo-France

⊖ Beaucoup de prédictions négatives
Résultats moins bons que par la régression directement sur la variable à prédire

? Quel modèle choisir entre AROME et ARPEGE ?

II. Idées d'implémentation

Algorithme LGBM

Light Gradient Boosting Method

- Agrégation de plusieurs arbres de décision
- Optimisation du découpage: feuille qui maximise la diminution de la fonction de perte lors du partage en deux nœuds fils
- Algorithme itératif de descente de gradient, avec optimisation d'une fonction de perte différentiable

Pourquoi choisir cet algorithme ?

- Meilleure précision. Mais arbres plus complexes (attention au sur-apprentissage)
- Rapidité à l'apprentissage, réduction des besoins mémoires.

Résultats à peu près équivalents à ceux du réseau de neurones
Choix arbitraire du réseau de neurones

Paramètres de l'algorithme

- n_estimators: nombre d'arbres
- max_depth: profondeur de l'arbre. Contrôle du problème de sur-apprentissage.
- num_leaves: nombre de feuilles dans un arbre.
- min_child_samples: nombre de données minimum que doit contenir une feuille
- learning_rate: contrôle de la vitesse d'apprentissage.
- reg_alpha: terme de régularisation L1.

Optimisation des paramètres

Validation croisée sur l'échantillon d'apprentissage (*RandomizedSearchCV()* et *GridSearchCV()*)

III. Méthode finale retenue

Approche et algorithme utilisés

Approche : Prédiction directe sur *ground_truth* (régresseur)

Données : $X_{station}$ + $X_{forecast}$ (valeurs extrêmes clippées)

Modèle : Réseau de neurones (ANN) : 20 couches de 32 neurones

Paramètres :
- Batch_size $= 128$
- Nb_epochs $= 150$
- LOSS function : *Mean absolute error (MAE)*

Après entrainement du modèle

MAPE loss $= \mathbf{18.812712}$

MAE loss $= \mathbf{0.609}$

$\boldsymbol{R^2} = \mathbf{0.69}$

Entraînement par rapport à la MAPE :

$$MAPE = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

Surapprentissage !

III. Méthode finale retenue

Résultats obtenus (prédiction sur dataset test)

- **MAPE loss (score *Kaggle*)** $= 30,60$

Synthèse du cheminement

- **Démarche et résultats reproductibles (Dépôt GitHub)**
  - Pre-processing optionnel

Conclusion et perspectives

- Appréhension de plusieurs **modèles** et de plusieurs **stratégies** pour répondre à un problème complexe
- Résultats de prédiction **acceptables**

- Ajouter de nouvelles features : *rolling, lag* sur les heures d'une journée
- Utiliser la fonction `interpolate` pour combler des valeurs manquantes
- Certaines variables contiennent jusqu'à 40% de NaN (ff, td, dd, hu) : ajout de bruit ?

Merci de votre attention !