Fiche récap : Modélisez des séries temporelles

Data

Code Python

Ce code sert à **rendre la colonne de dates exploitable** :

```
df["date"] = pd.to_datetime(   → Transforme les dates brutes en vrai format.
    df["date_raw"],
    dayfirst=True,   → Indique que le jour vient avant le mois.
    errors="coerce"   → Remplace les dates invalides par NaT pour éviter une erreur.

)

df = (
    df
    .dropna(subset=["date"])   → Supprime les lignes où la date n'est pas convertie.
    .sort_values("date")   → Remet les observations dans l'ordre chronologique.
    .set_index("date")   → Fait de la date l'axe principal de la série.
)
```

Ce code sert à **produire une série journalière propre et continue** :

```
serie_daily = (
    df["ili_consultations"]
    .groupby(level=0)   → Regroupe les lignes ayant la même date.
    .mean()   → Prend la moyenne.
    .sort_index()   → Garantit l'ordre temporel.
)

serie_daily = (
    serie_daily
    .asfreq("D")    → Impose une fréquence quotidienne.
    .interpolate(method="time")    → Remplit les jours manquants en estimant
                                     une valeur cohérente selon le temps.
)
```

Ce code sert à **changer l'échelle temporelle** et à **créer une variable passée** :

```
serie_weekly = (
    serie_daily
    .resample("W-SUN")   → Agrège les données par semaine, avec une semaine
    .sum()                  qui se termine le dimanche.
)

features["lag_4w"] =   → Contient la valeur observée 4 semaines + tôt.
features["y"].shift(4)   → Décale la série de 4 périodes vers le bas.
```

Définitions

Série temporelle
Suite de valeurs observées à des dates successives.

Tendance
Évolution lente et durable du niveau moyen de la série.

Saisonnalité
Motif qui se répète à intervalles réguliers.

Stationnarité
Propriété d'une série dont les caractéristiques restent stables dans le temps.

Lag
Valeur passée utilisée comme variable de comparaison ou de prédiction.

Fuite temorelle
Utilisation involontaire d'informations futures pendant l'entraînement.

Interpolation
Estimation de valeurs manquantes entre deux dates connues en supposant une évolution progressive.

Décomposition STL
Séparation d'une série en trois composantes : tendance, saisonnalité et résidus.

Lissage exponentiel
Méthode de prévision qui donne plus de poids aux observations récentes via un paramètre alpha.

Bonnes pratiques

- ✅ Convertir les dates avec une stratégie explicite de parsing multi-formats.
- ✅ Trier les observations et définir un index temporel avant toute analyse.
- ✅ Documenter la règle appliquée pour traiter les doublons par date.
- ✅ Rendre la série continue sur la fréquence choisie avant la modélisation.
- ✅ Créer des variables causales en n'utilisant que le passé disponible.
- ✅ Vérifier la stationnarité avec les tests ADF et KPSS conjointement.
- ✅ Évaluer les modèles avec un split chronologique ou un walk-forward.
- ✅ Présenter les prévisions avec un intervalle de confiance exploitable.

Erreurs classiques

- ❌ Laisser des dates invalides ou incohérentes dans la série finale.
- ❌ Garder plusieurs lignes par date sans règle métier explicite.
- ❌ Supprimer des jours manquants quand la série doit rester continue.
- ❌ Utiliser des variables qui incorporent des informations futures.
- ❌ Mélanger les données avec un shuffle avant le split train-test.
- ❌ Confondre saisonnalité et tendance.