

Documentation de l'application de prévisions hebdomadaires de fréquentation

I. Introduction

1.1 Objectif du document

Ce document vise à fournir une description exhaustive et accessible de l'application dédiée aux prévisions hebdomadaires de fréquentation. Il accompagne aussi bien les utilisateurs novices dans l'obtention de prévisions que les développeurs chevronnés en quête de compréhension du fonctionnement interne de l'application. Ce guide détaillé vous accompagnera à chaque étape, de l'installation à l'interprétation des résultats.

1.2 Description de l'application

Anticipation et précision sont les maîtres mots de cette application. Conçue pour prédire la fréquentation hebdomadaire d'un lieu précis, elle s'appuie sur la puissance du modèle de prévision statistique SARIMAX (Seasonal Autoregressive Integrated Moving Average with eXogenous regressors). Au-delà des simples données historiques de fréquentation, le modèle intègre astucieusement des variables externes influentes pour affiner ses prédictions :

- **Les caprices de la météo** : Températures maximales et minimales, précipitations, ensoleillement... rien n'échappe à son analyse.
- **Le rythme des vacances scolaires** : Les variations saisonnières de la fréquentation sont ainsi prises en compte.
- **Les jours fériés** : Leur impact sur la fréquentation est soigneusement intégré.
- **Les temps forts commerciaux** : Soldes, lancement de produits phares (comme le dernier iPhone), tout est passé au crible.

Grâce à cette approche multifactorielle, l'application se distingue par la précision et la fiabilité de ses prévisions, s'imposant comme un outil d'aide à la décision incontournable.

1.2.1 Fonctionnalités clés

L'application se veut accessible et intuitive. Elle offre un panel de fonctionnalités pensées pour simplifier son utilisation :

- **Maîtrise du temps** : Choisissez librement la période pour laquelle vous souhaitez obtenir des prévisions.
- **Clarté des résultats** : Visualisez les prévisions hebdomadaires de manière claire et concise grâce à des tableaux et des graphiques interactifs.
- **Gestion de l'incertitude** : Un intervalle de confiance accompagne chaque prévision, vous permettant d'appréhender les fluctuations potentielles.
- **Analyse comparative** : Comparez aisément les prévisions avec les données des années précédentes pour une analyse contextuelle approfondie.

1.2.2 Architecture générale

Développée en Python, langage de prédilection dans l'univers de la science des données, l'application s'articule autour d'un écosystème de bibliothèques spécialisées :

- **Streamlit** : L'interface utilisateur web, interactive et conviviale, est son œuvre.
- **Pandas** : La manipulation et l'analyse des données sont ses terrains de jeux.
- **Statsmodels** : C'est ici que réside l'implémentation du modèle de prévision SARIMAX.
- **Plotly** : Il orchestre la création de graphiques interactifs et visuellement attrayants.
- **Requests** : Il interroge les API externes pour récupérer les précieuses données météo.
- **Joblib** : C'est le gardien des modèles et des scalars, assurant leur sauvegarde et leur chargement pour une utilisation ultérieure.

1.3 Public cible

Ce document s'adresse à un public large, aux profils variés :

- **Les utilisateurs finaux** : Quiconque souhaite obtenir des prévisions de fréquentation, sans nécessairement posséder de compétences techniques avancées.
- **Les analystes de données** : Ces experts trouveront ici un outil performant pour l'analyse des données de fréquentation et l'exploitation de modèles prédictifs.
- **Les développeurs** : Ceux qui souhaitent explorer les coulisses du code source, l'adapter à leurs besoins spécifiques ou contribuer à son évolution y trouveront leur bonheur.

II. Guide d'installation et de configuration

2.1 Prérequis

2.1.1 Python et bibliothèques

Avant de plonger dans l'univers des prévisions, assurez-vous d'avoir installé Python 3.7 ou une version ultérieure sur votre système. Un environnement virtuel Python est fortement recommandé pour éviter tout conflit de dépendances. Vous pouvez créer un environnement virtuel avec la commande suivante :

```
python -m venv .venv
```

Activez ensuite l'environnement virtuel :

```
# Sur Windows
.venv\Scripts\activate

# Sur macOS/Linux
source .venv/bin/activate
```

Une fois Python et votre environnement virtuel prêts, installez les librairies nécessaires à l'aide de la commande `pip` :

```
pip install -r requirements.txt
```

Le fichier `requirements.txt` contient la liste complète des dépendances du projet.

2.1.2 Configuration du proxy CNTLM (pour le réseau interne de l'entreprise)

Si vous utilisez cette application au sein du réseau interne de l'entreprise, la configuration d'un proxy CNTLM est indispensable pour accéder aux API externes, notamment celle fournissant les données météorologiques. CNTLM agit comme un intermédiaire, permettant à votre application de se connecter à Internet via le serveur proxy de l'entreprise.

2.1.2.1 Installation de CNTLM

1. Téléchargez la dernière version du fichier zip CNTLM depuis le site officiel.
2. Décompressez le contenu du fichier zip dans le répertoire `C:\Applications\cntlm`.

2.1.2.2 Configuration de CNTLM

1. Ouvrez le fichier `cntlm.ini` situé dans le répertoire `C:\Applications\cntlm` à l'aide d'un éditeur de texte.
2. Localisez la ligne contenant `Password` et supprimez-la.
3. Remplacez les valeurs suivantes par vos propres informations d'identification :

```
Username VOTRE_CUID
Domain AD
Proxy proxybcp.si.francetelecom.fr:8080
Proxy proxy.si.francetelecom.fr:8080
```

4. Générez un hash pour votre mot de passe e-buro en exécutant la commande suivante dans une console Cmder ouverte dans le répertoire `C:\Applications\cntlm` :

```
cntlm -H -u VOTRE_CUID -d AD -c cntlm.ini http://www.google.com
```

Remplacez `VOTRE_CUID` par votre CUID. Entrez votre mot de passe e-buro lorsque vous y êtes invité. Copiez le hash généré (par exemple, `2AB3390716014AE4399917D8330A91F7`).

5. De retour dans le fichier `cntlm.ini`, ajoutez les lignes suivantes en remplaçant le hash par celui que vous venez de copier :

```
Auth NTLMv2
PassNTLMv2 2AB3390716014AE4399917D8330A91F7
```

2.1.2.3 Vérification de l'installation

1. Lancez CNTLM en mode console en exécutant la commande suivante dans une console Cmder ouverte dans le répertoire `C:\Applications\cntlm` :

```
cntlm -c cntlm.ini -v
```

2. Ouvrez une nouvelle console Cmder et vérifiez l'accès à un site web externe à l'aide de la commande `curl` :

```
curl -x localhost:3128 http://www.google.com
```

Si la configuration est correcte, vous devriez obtenir le code source HTML de la page d'accueil de Google.

2.1.2.4 Démarrage automatique de CNTLM

Pour éviter de lancer CNTLM manuellement à chaque démarrage de votre ordinateur, suivez ces étapes :

1. Dans le répertoire `C:\Applications\cntlm`, créez un fichier nommé `startCntlm.bat` avec le contenu suivant :

```
cntlm.exe -c cntlm.ini -f -v
cmd /k
```

2. Créez un raccourci vers `C:\Applications\cntlm\startCntlm.bat` dans le répertoire de démarrage de Windows :
`C:\Users\VOTRE_NOM_D_UTILISATEUR\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup`

2.1.2.5 Déclarer le proxy CNTLM comme proxy par défaut

1. Ouvrez une console **en tant qu'administrateur** et exécutez les commandes suivantes :

```
setx PROXY http://localhost:3128
setx HTTPS_PROXY http://localhost:3128
setx HTTP_PROXY http://localhost:3128
```

2. **Redémarrez votre ordinateur** pour que les modifications soient prises en compte.

2.2 Téléchargement du code source

Le code source de l'application est disponible sur le dépôt Git [insérer le lien du dépôt Git]. Vous pouvez le télécharger sous forme de fichier zip ou cloner le dépôt à l'aide de la commande `git clone`.

2.3 Configuration de l'application

2.3.1 Fichier `config.py`

Le fichier `config.py` est le centre névralgique de la configuration de l'application. Il contient les paramètres essentiels au bon fonctionnement de l'application, tels que :

- **MODEL_PATH** : Chemin vers le répertoire contenant les fichiers du modèle.
- **CIBLE** : Nom de la zone cible pour le modèle.
- **HISTORICAL_FILE** : Chemin vers le fichier Excel contenant les historiques de flux.
- **LAT** : Latitude de la zone d'étude.
- **LON** : Longitude de la zone d'étude.
- **API_METEO_URL** : URL de l'API d'archive météo.
- **USE_PROXY** : Indique si un proxy doit être utilisé pour les requêtes externes.
- **PROXY_URL** : Adresse du proxy (CNTLM par défaut sur localhost).

Exemple de fichier `config.py` :

```
# === Répertoires ===
# Répertoire principal pour les modèles
MODEL_PATH = r"C:/Chemin/Vers/Vos/Modeles"

# === Fichiers de Modèles ===
CIBLE = 'NomDeVotreCible' # Zone cible pour le modèle

# ... autres configurations ...
```

Paramètres à modifier :

- **MODEL_PATH** : Remplacez `C:/Chemin/Vers/Vos/Modeles` par le chemin réel vers le répertoire contenant les fichiers du modèle.
- **CIBLE** : Remplacez `NomDeVotreCible` par le nom de la zone cible pour le modèle.
- **HISTORICAL_FILE** : Remplacez le chemin par le chemin réel vers votre fichier Excel contenant les historiques de flux.
- **LAT** : Remplacez par la latitude de votre zone d'étude.
- **LON** : Remplacez par la longitude de votre zone d'étude.
- **USE_PROXY** : Définissez sur `True` si vous êtes sur le réseau de l'entreprise et que vous utilisez CNTLM, sinon `False`.

Assurez-vous de bien adapter ces paramètres à votre environnement et à vos besoins spécifiques.

III. Guide d'utilisation

3.1 Lancement de l'application

Une fois l'installation et la configuration terminées, vous pouvez lancer l'application. Pour cela, ouvrez votre terminal ou votre invite de commande, déplacez-vous dans le répertoire où se trouve le fichier `app.py` à l'aide de la commande `cd`, puis exécutez la commande suivante :

```
streamlit run app.py
```

Votre navigateur web par défaut s'ouvrira automatiquement et affichera l'interface utilisateur de l'application. Vous serez accueilli par un titre clair et une interface conviviale.

3.2 Sélection de la plage de prévision

L'application vous permet de choisir la période pour laquelle vous souhaitez obtenir des prévisions de fréquentation. Par défaut, la plage sélectionnée couvre les 4 semaines précédant la date du jour et les 4 semaines suivantes. Vous pouvez modifier cette plage en utilisant le sélecteur de dates, qui s'affiche clairement sur l'interface. N'hésitez pas à ajuster la période en fonction de vos besoins d'analyse.

3.3 Lancement de la prévision

Après avoir sélectionné la plage de dates souhaitée, cliquez sur le bouton "Lancer la prévision". L'application se met alors en marche et effectue les opérations suivantes :

1. **Récupération des données** : L'application collecte les données météorologiques et les variables exogènes pour la période sélectionnée.
2. **Chargement du modèle** : Le modèle de prévision SARIMAX, préalablement entraîné, est chargé en mémoire, accompagné des scalers et du composant PCA utilisés pour le prétraitement des données.
3. **Prévision** : Le modèle utilise les données collectées pour calculer les prévisions de fréquentation.
4. **Affichage des résultats** : Les résultats, clairs et concis, s'affichent sur l'interface.

3.4 Interprétation des résultats

L'application présente les résultats de manière intuitive et accessible, même pour les utilisateurs moins familiers avec les analyses statistiques.

3.4.1 Tableau de prévisions

Un tableau synthétique présente les prévisions hebdomadaires pour la période sélectionnée. Chaque ligne du tableau correspond à une semaine, avec les informations clés suivantes :

- **Date** : Premier jour de la semaine (lundi), pour une identification rapide.
- **Prévision** : Valeur prédite par le modèle pour la fréquentation de la semaine, vous donnant une idée précise de l'affluence attendue.
- **Borne inférieure** : Limite inférieure de l'intervalle de confiance à 95%, indiquant la borne basse plausible pour la prévision.
- **Borne supérieure** : Limite supérieure de l'intervalle de confiance à 95%, indiquant la borne haute plausible pour la prévision.

L'intervalle de confiance est un élément crucial pour évaluer la fiabilité de la prévision. Plus l'intervalle est étroit, plus la prévision est considérée comme fiable.

3.4.2 Graphique de prévisions

Un graphique interactif, plus visuel, vient compléter le tableau en affichant les prévisions, l'intervalle de confiance et les données historiques des deux années précédentes. Ce graphique offre plusieurs avantages :

- **Comparaison avec le passé** : Visualisez facilement les prévisions par rapport aux tendances observées les années précédentes.
- **Identification des tendances** : Repérez rapidement les périodes de forte et de faible fréquentation grâce à la représentation graphique.
- **Analyse des facteurs externes** : Analysez l'impact des variables exogènes, telles que la météo ou les vacances scolaires, sur les prévisions.

3.4.3 Graphique des historiques

Un graphique dédié à l'historique complet des données de fréquentation pour la zone cible est également disponible. Il permet de :

- **Visualiser l'évolution sur le long terme** : Observez les tendances de fréquentation sur une période plus longue.
- **Identifier les saisonnalités** : Repérez les schémas saisonniers et les événements exceptionnels qui ont pu influencer la fréquentation.
- **Comparer avec les prévisions** : Mettez en perspective les prévisions avec l'ensemble des données historiques pour une analyse complète.

IV. Description technique

4.1 Structure du code

L'application est structurée en plusieurs modules Python, chacun ayant un rôle spécifique :

4.1.1 `app.py`

Ce module représente le cœur de l'application. Il orchestre l'ensemble des fonctionnalités, de l'affichage de l'interface utilisateur à la récupération des données, en passant par le lancement des prévisions et la visualisation des résultats.

- **Interface utilisateur** : Définit la structure de l'interface utilisateur à l'aide de la librairie Streamlit.
- **Gestion des événements** : Gère les interactions de l'utilisateur, comme la sélection de la plage de dates et le clic sur le bouton de lancement des prévisions.
- **Orchestration des modules** : Appelle les fonctions des autres modules pour récupérer les données, charger le modèle, effectuer les prévisions et afficher les résultats.

4.1.2 `exogenous.py`

Ce module est responsable de la gestion des variables exogènes. Il contient les fonctions permettant de :

- **Récupérer les données météorologiques** : Utilise la classe `WeatherDataFetcher` pour interroger l'API météo et récupérer les données historiques ou prévues pour la zone d'étude.
- **Créer les variables exogènes** : Calcule les variables basées sur la date, comme les vacances scolaires, les soldes, les jours fériés, etc.
- **Agréger les données** : Agrège les données météorologiques et les variables exogènes à la semaine pour les aligner sur la granularité des prévisions.

4.1.3 `forecast.py`

Ce module est dédié à la prévision elle-même. Il contient les fonctions pour :

- **Charger le modèle et les scalers** : Charge le modèle SARIMAX, les scalers et le composant PCA préalablement entraînés et sauvegardés.
- **Préparer les données pour la prévision** : Transforme les données exogènes en utilisant les scalers et le composant PCA.

- **Effectuer la prévision** : Utilise le modèle SARIMAX pour générer les prévisions de fréquentation.
- **Dénormaliser les prévisions** : Convertit les prévisions normalisées en valeurs réelles à l'aide des scalers.

4.1.4 data_loader.py

Ce module est responsable du chargement et du prétraitement des données historiques de fréquentation. Il contient les fonctions pour :

- **Charger les données depuis le fichier Excel** : Lit le fichier Excel contenant les historiques de flux.
- **Préparer les données** : Convertit les colonnes 'Année' et 'Semaine' en une colonne de date, gère les périodes de confinement en remplaçant les valeurs manquantes, et interpole les données pour combler les trous.

4.1.5 visualizations.py

Ce module est dédié à la création des graphiques. Il contient les fonctions pour :

- **Tracer le graphique des prévisions** : Affiche les prévisions, l'intervalle de confiance et les données historiques des deux années précédentes.
- **Tracer le graphique des historiques complets** : Affiche l'historique complet des données de fréquentation disponibles.

4.1.6 config.py

Ce module centralise les paramètres de configuration de l'application, tels que :

- **Chemins vers les fichiers** : Définit les chemins vers les fichiers du modèle, les données historiques, etc.
- **Paramètres du modèle** : Définit le nom de la zone cible pour le modèle.
- **Paramètres de l'API météo** : Définit les coordonnées géographiques de la zone d'étude et l'URL de l'API météo.
- **Configuration du proxy** : Définit si un proxy doit être utilisé et son adresse.

4.2 Fonctionnement du modèle de prévision

4.2.1 Modèle SARIMAX

Le modèle de prévision utilisé est un modèle SARIMAX (Seasonal Autoregressive Integrated Moving Average with exogenous regressors). Ce modèle est particulièrement adapté à la prévision de séries temporelles présentant des tendances saisonnières et des influences de variables externes.

4.2.2 Variables exogènes

Les variables exogènes jouent un rôle crucial dans la précision des prévisions. L'application utilise les variables exogènes suivantes :

- **Température maximale et minimale** : Influence sur la fréquentation en fonction des conditions climatiques.
- **Précipitations** : Impact négatif sur la fréquentation en cas de pluie.
- **Vacances scolaires** : Augmentation de la fréquentation pendant les périodes de vacances.
- **Soldes** : Augmentation de la fréquentation pendant les périodes de soldes.
- **Lancement d'un nouvel iPhone** : Pic de fréquentation potentiel lors du lancement d'un nouveau modèle d'iPhone.
- **Jours fériés** : Impact variable sur la fréquentation en fonction du jour férié.

4.2.3 Chargement et prétraitement des données

Avant d'être utilisées pour la prévision, les données sont chargées, nettoyées et transformées :

- **Chargement** : Les données historiques de fréquentation sont chargées depuis un fichier Excel, tandis que les données météorologiques sont récupérées depuis l'API Open-Meteo.
- **Nettoyage** : Les périodes de confinement, pendant lesquelles les données de fréquentation sont aberrantes, sont identifiées et remplacées par des valeurs manquantes.
- **Interpolation** : Les valeurs manquantes sont interpolées pour combler les trous dans les données.
- **Transformation** : Les données sont normalisées à l'aide de scalers et une analyse en composantes principales (PCA) est appliquée pour réduire la dimensionnalité des variables exogènes.

4.3 Maintenance et mises à jour

Pour garantir la fiabilité des prévisions sur le long terme, il est important d'assurer la maintenance et la mise à jour régulière de l'application :

- **Mise à jour des données** : Intégrer les nouvelles données de fréquentation et météorologiques de manière régulière.
- **Réentraînement du modèle** : Réentraîner le modèle SARIMAX périodiquement avec les données mises à jour pour tenir compte des nouvelles tendances et des variations saisonnières.
- **Surveillance des performances** : Surveiller les performances du modèle en comparant les prévisions avec les valeurs réelles et ajuster les paramètres du modèle si nécessaire.
- **Veille technologique** : Se tenir informé des nouvelles techniques de prévision et des nouvelles API disponibles pour améliorer l'application.