Documentation du Projet: EXAMEN\_IA1\_WEATHER\_TOURISM

1. Introduction

Ce document fournit une documentation concise du projet `EXAMEN\_IA1\_WEATHER\_TOURISM`, un pipeline ETL (Extract, Transform, Load) conçu pour collecter des données météorologiques, les transformer en y ajoutant un score de tourisme, et les préparer pour l'analyse et la visualisation. Le projet utilise Apache Airflow pour l'orchestration du pipeline.

2. Structure du Projet

Le dépôt GitHub est organisé comme suit:

```

EXAMEN\_IA1\_WEATHER\_TOURISM/

├── weather\_tourism/

│ ├── dags/ # Définitions des DAG Airflow (openweather\_tourism\_dag.py)

│ ├── data/ # Fichiers de données (current\_weather.csv, weather\_combined.csv)

│ └── scripts/ # Scripts ETL (extract.py, transform.py, load.py)

```

3. Pipeline ETL (Extract, Transform, Load)

Le pipeline ETL automatise la collecte, le traitement et la préparation des données météorologiques.

3.1. Extraction (`extract.py`)

Ce script récupère les données météorologiques actuelles de l'API OpenWeatherMap pour des villes prédéfinies et les sauvegarde dans `current\_weather.csv`.

3.2. Transformation (`transform.py`)

Ce script lit `current\_weather.csv` (et potentiellement des données historiques), calcule un `tourism\_score` basé sur la température, l'humidité et la vitesse du vent, puis combine et sauvegarde les données transformées dans `weather\_combined.csv`.

3.3. Chargement (`load.py`)

Ce script est responsable du chargement des données transformées de `weather\_combined.csv` vers une destination finale, typiquement une base de données ou un entrepôt de données pour l'analyse.

3.4. Orchestration avec Apache Airflow (`openweather\_tourism\_dag.py`)

Le DAG Airflow orchestre les tâches d'extraction, de transformation et de chargement de manière séquentielle, assurant l'automatisation et la planification du pipeline.

4.Explication du Modèle en Étoile:

Table de Faits: `Fact\_Weather\_Tourism`\*\*: Contient les mesures numériques (température, humidité, score de tourisme) et les clés étrangères (`date\_id`, `city\_id`, `weather\_id`) vers les tables de dimensions.

Table de Dimension: `Dim\_Date`\*\*: Contient les attributs liés au temps (date, année, mois, jour).

\* \*\*Table de Dimension: `Dim\_City`\*\*: Contient les attributs descriptifs sur les villes (nom de la ville, latitude, longitude).

\* \*\*Table de Dimension: `Dim\_Weather`\*\*: Contient les attributs descriptifs sur les conditions météorologiques (description météorologique).

5. Analyse Exploratoire des Données (EDA)

Une EDA est générée sous forme de notebook Jupyter (`eda\_weather\_tourism.ipynb`) et peut être consultée en format HTML (`eda\_weather\_tourism.html`). Elle inclut:

\* \*\*Informations Générales\*\*: Aperçu des données (`df.info()`, `df.head()`, `df.describe()`).

\* \*\*Distribution des Variables Clés\*\*: Histogrammes pour la température, l'humidité et le score de tourisme.

\* \*\*Analyse des Corrélations\*\*: Matrice de corrélation visualisée par une heatmap.

6. Dashboard Power BI

Le fichier `weather\_combined.csv` est la source de données pour un dashboard Power BI. Les visualisations probables incluent des cartes des villes, des graphiques de tendances temporelles, des tableaux de bord des conditions actuelles, et des comparaisons entre villes, le tout avec des filtres interactifs.

7. Conclusion

Le projet `EXAMEN\_IA1\_WEATHER\_TOURISM` offre une solution robuste pour la collecte, la transformation et la visualisation des données météorologiques à des fins touristiques, avec un pipeline ETL automatisé et un modèle de données optimisé pour l'analyse.