

TP2 Données massives & IOT

Aziz HACHICHA

2 IDSD 1

Objectifs du TP :

- Générer et stocker localement des données simulées de capteurs IoT.
- Analyser et visualiser ces données pour en extraire des tendances.
- Détecter des anomalies dans les mesures.
- Entraîner un modèle d'IA pour prédire la température.
- Développer une interface interactive avec Streamlit.

1. Préparation de l'environnement

Création d'un environnement virtuel (python -m venv venv).

2. Génération et stockage des données (data_acquisition.py)

- Génération réaliste de données de capteurs (température, humidité, pression).
- Stockage des données dans un fichier CSV pour une analyse ultérieure.

```
data > anomalies.csv
1 timestamp,sensor_id,temperature,humidity,pressure
2 2025-02-14 10:49:41.646847,2,20.756990663013102,75.64444808423062,997.8705761673884
3 2025-02-14 10:42:41.646847,2,38.64329480577848,84.78521090319262,1022.3820680938276
4 2025-02-14 10:40:41.646847,3,21.6184197450243,38.96608293943896,1027.5795446828502
5 2025-02-14 10:37:41.646847,5,20.113290891005512,68.72552816684424,991.230171882116
6 2025-02-14 10:33:41.646847,3,21.643201954735915,44.90132372901313,1026.1305536199725
7 2025-02-14 10:23:41.646847,5,20.449557037147308,56.04942883374967,1027.17974244758
8 2025-02-14 10:22:41.646847,3,38.869741221525786,58.77480917866899,1042.2786902114442
9 2025-02-14 10:17:41.646847,1,21.84244079179557,66.717703744052,974.6344154092776
10 2025-02-14 10:15:41.646847,1,20.130426670127832,70.48930680817685,1012.8591387524514
11 2025-02-14 10:10:41.646847,1,38.31706687136896,42.81266655284493,1004.1281891670365
12 2025-02-14 10:09:41.646847,4,21.58662985653251,82.70820078459079,950.468139293476
```

```
data > sensor_data.csv
1 timestamp,sensor_id,temperature,humidity,pressure
2 2025-02-14 10:49:41.646847,2,20.756990663013106,75.64444808423062,997.8705761673885
3 2025-02-14 10:48:41.646847,4,26.22273671183016,84.10929762981758,1016.2073184796427
4 2025-02-14 10:47:41.646847,4,24.645981488155307,82.85083811117809,1024.3057217126668
5 2025-02-14 10:46:41.646847,2,26.210156029627075,73.5229800403675,1031.439874916868
6 2025-02-14 10:45:41.646847,1,27.231473496811205,35.35072979060669,1035.2542448008962
7 2025-02-14 10:44:41.646847,5,27.182951320001422,70.29803236698137,953.3811455773764
8 2025-02-14 10:43:41.646847,2,33.116358123268,42.48668607487252,985.1681221620096
```

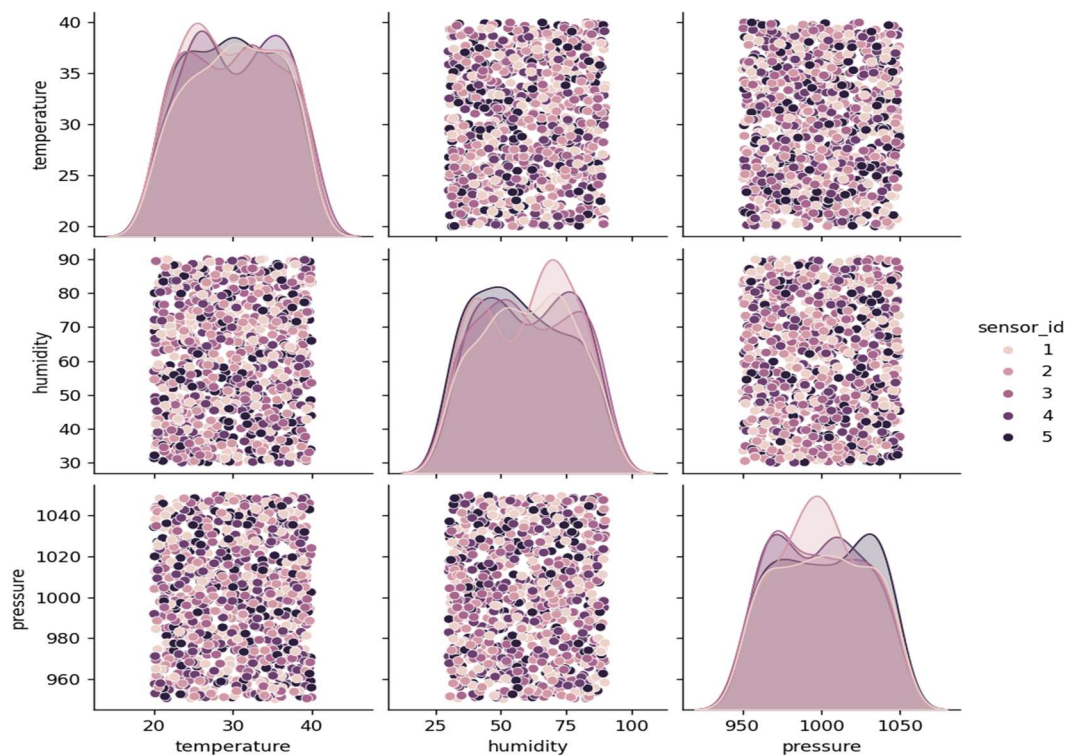
3. Analyse et visualisation (data_analysis.py)

- Conversion de la colonne timestamp au format datetime.
- Visualisation graphique des corrélations à l'aide de pairplot.
- Mise en place d'une détection des valeurs anormales (anomalies de température).

Statistiques descriptives :

	timestamp	sensor_id	temperature	humidity	pressure
count	1000	1000.000000	1000.000000	1000.000000	1000.000000
mean	2025-02-14 02:30:11.646846976	3.016000	30.067930	59.697880	999.815621
min	2025-02-13 18:10:41.646847	1.000000	20.023536	30.033781	950.022493
25%	2025-02-13 22:20:26.646846976	2.000000	25.178550	44.869388	974.464342
50%	2025-02-14 02:30:11.646846976	3.000000	30.105128	59.745338	999.098746
75%	2025-02-14 06:39:56.646846976	4.000000	35.071661	74.295587	1025.362453
max	2025-02-14 10:49:41.646847	5.000000	39.991394	89.991948	1049.944050
std	NaN	1.404178	5.747753	17.418212	28.642297

On a aussi effectué une statistique descriptive pour chaque attribut.



-L'axe diagonal montre la distribution de chaque variable (température, humidité, pression) par capteur.

-Chaque couleur représente un sensor_id :

Tu peux voir par exemple que les capteurs n'ont pas tous exactement la même distribution de température, humidité ou pression. Cela peut indiquer une variabilité naturelle ou une calibration différente entre capteurs.

-Hors diagonale montre la relation entre chaque paire de variables :

Température vs Humidité, Température vs Pression, Humidité vs Pression

-Les points sont très dispersés, ce qui suggère :

Peu ou pas de corrélation linéaire forte entre ces variables.

-Répartition par capteur (sensor_id) :

Les données sont colorées selon le capteur qui les a générées. Aucune couleur ne semble isolée, ce qui indique une bonne homogénéité entre les capteurs. Cependant, certaines distributions diagonales (notamment en température et pression) montrent que certains capteurs génèrent plus fréquemment certaines plages de valeurs.

4. Modélisation IA (model_training.py)

- Entraînement d'un modèle de régression linéaire.
- Variables utilisées : humidité et pression pour prédire la température.
- Séparation des données : 80 % pour l'entraînement, 20 % pour le test.
- Évaluation avec MAE et MSE.
- Sauvegarde du modèle via joblib pour intégration future.

```
Précision du modèle : MAE = 4.74, MSE = 30.73  
Modèle entraîné et sauvegardé dans models/iot_temperature_model.pkl
```

Interprétation des métriques :

MAE (Mean Absolute Error) = 4.74 : En moyenne, le modèle se trompe de ± 4.74 unités de température

MSE (Mean Squared Error) = 30.73 : Ici, un MSE de 30.73 signifie que, statistiquement, les erreurs peuvent aller jusqu'à $\sqrt{30.73} \approx 5.5^\circ\text{C}$ d'écart type.

3. Interface utilisateur avec Streamlit (dashboard.py)

- Développement d'un tableau de bord interactif.
- Utilisation de sliders pour ajuster les paramètres (humidité, pression).
- Affichage en temps réel des prédictions de température.
- Lancement via streamlit run dashboard.py.

Tableau de Bord IoT en Temps Réel

Aperçu des données

	timestamp	sensor_id	temperature	humidity	pressure
0	2025-02-14 10:49:41.646847	2	20.757	75.6444	997.8706
1	2025-02-14 10:48:41.646847	4	26.2227	84.1093	1,016.2073
2	2025-02-14 10:47:41.646847	4	24.646	82.8508	1,024.3057
3	2025-02-14 10:46:41.646847	2	26.2102	73.523	1,031.4399
4	2025-02-14 10:45:41.646847	1	27.2315	35.3507	1,035.2542

Prédiction de Température

Humidité (%)



Pression (hPa)



Température prédite : 30.01°C

On constate d'après la figure, que si on varie l'humidité ou la pression la température change comme :

Temperature predite=a·humidite'+b·pression+c