



Data_Set(SmartWatt Predictor)

Prédiction de la Consommation Énergétique d'une Maison Basée sur les Données IoT

Membre du groupe

- TCHECOUNNOU Sessi Linerol
- MOUSSA YARI Hanim

Introduction

La consommation énergétique d'une maison est influencée par plusieurs facteurs tels que la température, l'humidité, la présence d'occupants et l'éclairage naturel. L'objectif du projet **SmartWatt Predictor** est d'exploiter les données historiques issues des capteurs IoT pour anticiper la consommation énergétique future d'une maison. Cela permet une gestion optimisée des ressources, une meilleure efficacité énergétique et une réduction du gaspillage.

Github Project

BlvckHL/SmartWatt-Predictor

Owner: BlvckHL • Star count: 0 • Created: Fri, Mar 7, 2025

 github.com

<https://github.com/BlvckHL/SmartWatt-Predictor>

Données Utilisées

Les données historiques collectées comprennent les paramètres suivants :

- **Température (°C)**
- **Humidité (%)**
- **Présence d'occupants (0/1)**
- **Éclairage naturel (%)**
- **Consommation énergétique (kWh) [Variable cible]**
- **Jour de la semaine**
- **Heure de la journée**

Approche

Nous allons utiliser une approche basée sur l'apprentissage automatique avec un modèle de régression pour prédire la consommation énergétique à partir des données collectées.

1. Préparation des Données

- Chargement des données.
- Nettoyage et gestion des valeurs manquantes.
- Transformation des variables temporelles.
- Normalisation et encodage des données.

2. Construction du Modèle

- Division des données en ensembles d'entraînement et de test.

- Entraînement d'un modèle de régression.
- Optimisation des hyperparamètres.

3. Validation et Prédiction

- Évaluation des performances du modèle.
- Prédiction sur de nouvelles données.
- Visualisation des résultats.

Implémentation en Python

Voici une implémentation en Python utilisant **scikit-learn** pour la régression.

Outils utilisés et leur pertinence

- **pandas** : Utilisé pour la manipulation et l'analyse des données, permettant le chargement et le prétraitement efficace des jeux de données.
- **numpy** : Fournit des outils pour manipuler des tableaux et effectuer des calculs numériques avancés.
- **matplotlib** : Permet la visualisation des résultats sous forme de graphiques.
- **scikit-learn** : Fournit des outils pour la normalisation des données, la séparation en ensembles d'entraînement/test et la mise en œuvre du modèle de régression.
- **RandomForestRegressor** : Algorithme de régression basé sur les forêts aléatoires, permettant de capturer les relations complexes entre les variables.