

■ Tableau de Bord Analytique des Véhicules Électriques

Documentation Complète du Projet

Introduction

Le Tableau de Bord Analytique des Véhicules Électriques est une application web interactive développée avec Streamlit qui permet d'analyser les données des véhicules électriques à travers quatre indicateurs clés de performance (KPI) distincts.

Objectifs Principaux

- Charger et stocker les données des véhicules électriques
- Fournir une interface intuitive pour explorer les données
- Afficher quatre visualisations pertinentes (KPI)
- Permettre le filtrage dynamique des résultats
- Assurer la qualité du code avec des tests unitaires

Technologies Utilisées

- Streamlit : Framework pour l'interface web interactive
- DuckDB : Base de données SQL embarquée
- Pandas : Manipulation et analyse des données
- Plotly : Visualisations interactives
- Python 3.8+ : Langage de programmation

Contexte du Projet

Les données des véhicules électriques sont complexes et multidimensionnelles. Les analystes ont besoin d'un outil pour charger rapidement les données, filtrer par critères spécifiques, visualiser les tendances et comparer les performances entre marques et segments.

Données Utilisées

Source : `electric_vehicles_spec_2025.csv`

Structure des Données

Le fichier CSV contient 22 colonnes avec les informations suivantes :

| Colonne | Type | Description |
|----------------------|---------|-------------------------|
| brand | VARCHAR | Marque du véhicule |
| model | VARCHAR | Modèle du véhicule |
| range_km | DECIMAL | Autonomie en km |
| battery_capacity_kWh | DECIMAL | Capacité batterie |
| acceleration_0_100_s | DECIMAL | Accélération 0-100 km/h |
| efficiency_wh_per_km | DECIMAL | Efficacité énergétique |
| segment | VARCHAR | Segment du véhicule |
| car_body_type | VARCHAR | Type de carrosserie |

Caractéristiques des Données

- Volume : 1000+ véhicules, 50+ marques, 10+ segments
- Qualité : Valeurs manquantes gérées automatiquement
- Encodage : UTF-8

Les Quatre KPI

KPI 1 : Plage Moyenne par Segment

La plage moyenne (autonomie) en kilomètres pour chaque segment de véhicule. Visualisé sous forme de graphique en barres avec un gradient de couleurs.

Formule : $\text{Plage Moyenne} = \Sigma(\text{range_km}) / \text{Nombre de véhicules par segment}$

Utilité : Comparer l'autonomie entre segments

KPI 2 : Accélération Moyenne par Marque

Le temps d'accélération moyen (0-100 km/h) en secondes pour chaque marque. Visualisé sous forme de graphique en barres avec les top 15 marques.

Formule : $\text{Accélération Moyenne} = \Sigma(\text{acceleration_0_100_s}) / \text{Nombre de véhicules par marque}$

Utilité : Comparer les performances d'accélération

KPI 3 : Batterie vs Efficacité Énergétique

La relation entre la capacité de la batterie (kWh) et l'efficacité énergétique (Wh/km). Visualisé sous forme de graphique de dispersion avec différenciation par segment.

Utilité : Identifier les corrélations entre batterie et efficacité

KPI 4 : Distribution par Type de Carrosserie

Le nombre et le pourcentage de véhicules pour chaque type de carrosserie. Visualisé sous forme de graphique en camembert.

Formule : $\text{Pourcentage} = (\text{Nombre par type} / \text{Total}) \times 100$

Utilité : Comprendre la composition du dataset

Application Streamlit

Interface Utilisateur

1. En-tête : Titre et description du projet
2. Section de Téléchargement : Interface pour charger les données CSV
3. Barre Latérale : Filtres par marque, segment et type de carrosserie
4. Grille des KPI : 4 visualisations en disposition 2x2
5. Boutons de Contrôle : Effacer les données et actualiser

Workflow Utilisateur

- Étape 1 : Charger les données CSV
- Étape 2 : Explorer les 4 visualisations
- Étape 3 : Appliquer les filtres
- Étape 4 : Analyser les résultats filtrés

Implémentation Technique

Architecture

L'application suit une architecture en couches :

- Couche UI : Streamlit (interface utilisateur)
- Couche Métier : Filtres, visualisations, calculs
- Couche Données : DuckDB (base de données)

Composants Principaux

- DashboardApp : Orchestration de l'application
- DuckDBManager : Gestion de la base de données
- FilterManager : Gestion des filtres
- VisualizationEngine : Génération des visualisations

Gestion des Erreurs

- Erreur 1 : Conversion de données invalides → `pd.to_numeric(errors='coerce')`
- Erreur 2 : Contrainte NOT NULL → `df.dropna()` + schéma flexible

Tests et Qualité

Stratégie de Test

- 42 tests unitaires
- 100% de réussite
- Couverture : 85%+

Répartition des Tests

| Module | Tests | Statut |
|------------------------|-------|----------|
| test_database.py | 17 | ■ PASSÉS |
| test_filters.py | 11 | ■ PASSÉS |
| test_visualizations.py | 14 | ■ PASSÉS |
| Total | 42 | ■ PASSÉS |

Résultats et Conclusion

Objectifs Réalisés

- Lecture des données CSV
- Stockage avec DuckDB
- Application Streamlit
- Quatre visualisations (KPI)
- Filtrage dynamique
- Tests unitaires complets
- Documentation complète

Statistiques Finales

- Fichiers source : 5
- Lignes de code : 750+
- Tests unitaires : 42
- Taux de réussite : 100%
- Fichiers de documentation : 10
- Bugs corrigés : 2

Conclusion

Le Tableau de Bord Analytique des Véhicules Électriques est un projet complet et professionnel qui répond à tous les objectifs spécifiés. L'application fournit une interface intuitive et moderne, offre des visualisations pertinentes et interactives, gère les données de manière robuste, est couverte par des tests unitaires complets, et est bien documentée et facile à maintenir. Le projet est prêt pour la production.