

Electric Car Sales Analytics Interactive Dashboard

Objective:

This capstone project analyzes global electric car sales trends using an interactive Streamlit dashboard.

It explores how electric vehicle (EV) adoption impacts:

- Global and regional sales
- Oil displacement
- Electricity demand
- EV stock share over time

The goal is to provide stakeholders, researchers, and enthusiasts with clear, filterable, visual insights.

Dataset Details:

- Source: Kaggle — Electric Car Sales 2010–2024
 - Dataset Link: [Kaggle EV Dataset](#)
 - Columns:
 - region: Countries or regions
 - category: Historical
 - parameter: (e.g., EV sales, oil displacement, stock share)
 - mode: Cars only
 - powertrain: BEV, PHEV, FCEV
 - year: Year
 - unit: Unit (vehicles, %, GWh, etc.)
 - value: Numeric value
-

Tools & Libraries Used:

- Python
 - pandas
 - plotly.express (for interactive charts)
 - streamlit (for web dashboard)
-

Key Steps Performed:

1 Load Libraries & Data

- Imported data with pandas, cached for speed in Streamlit.

2 Create Reusable Sidebar Filters

- Users can filter by region, powertrain, year range.

3 Build Multiple Dashboards with Tabs:

- 🌍 Global EV Sales Over Years: Total EV sales trend globally.
- 🇺🇸 Regional EV Sales Trends: Sales trends by country/region.
- ⚡ Electricity Demand: How EVs affect energy consumption.
- 🛢️ Oil Displacement: How EVs reduce oil use.
- 📈 EV Stock Share: % of EVs in total vehicle stock by region.

4 Create Interactive Visuals:

- Used plotly.express for all charts.
- Visuals update automatically based on filters.
- Tabs separate each dashboard clearly.

5 Deploy as an Interactive App:

- Runs locally with `streamlit run app.py`.
- Can be deployed on Streamlit Cloud or shared as a public link.

Outputs:

- Fully interactive web app.
- Users can explore multiple insights without writing code.
- Filters and tabs make the data easy to navigate.

Possible Extensions:

- Add forecasting for future EV adoption.
 - Include region maps with plotly or folium.
 - Export selected data as CSV from the app.
 - Integrate more real-time data with APIs.
-

Conclusion:

This capstone project demonstrates how Python + Streamlit + Plotly can turn static data into a professional, interactive dashboard — perfect for communicating EV adoption trends.

How to Run:

1. `pip install -r requirements.txt`
 2. `streamlit run app.py`
-

Prepared by:

[Yash Khobragade](#)