

```
# 📈 Week 2 – EV Sales Forecasting (Advanced: ARIMA + Linear Regression fallback)
# Intern: A H M Chandrashekhara | SKSVMA College of Engineering and Technology
# Forecast: 2025 - 2035

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
import warnings
warnings.filterwarnings("ignore")

# Step 1 – Load the dataset
df = pd.read_csv("Electric_Vehicle_Population_Data.csv")

# Step 2 – Group by Model Year to get EV sales per year
sales = df.groupby("Model Year").size().reset_index(name="Sales")
sales = sales[sales["Model Year"] >= 2015].reset_index(drop=True)
sales.columns = ["Year", "Sales"]
display(sales)

# Step 3 – Train baseline Linear Regression model
X = sales[["Year"]]
y = sales["Sales"]
lr = LinearRegression().fit(X, y)

# Step 4 – Forecast for 2025 - 2035
future_years = pd.DataFrame({"Year": list(range(2025, 2036))})

try:
    import pmdarima as pm
    model = pm.auto_arma(y, seasonal=False)
    forecast = model.predict(n_periods=len(future_years))
except:
    forecast = lr.predict(future_years)

future_years["Forecast"] = forecast
display(future_years)

# Step 5 – Plot graph
plt.figure(figsize=(10, 5))
plt.plot(sales["Year"], sales["Sales"], marker="o", label="Historical")
plt.plot(future_years["Year"], future_years["Forecast"], marker="o", label="Forecast")
plt.title("EV Sales Forecast (2025-2035)")
plt.xlabel("Year")
plt.ylabel("Sales")
plt.legend()
plt.grid(True)
plt.show()

# Save output
future_years.to_csv("ev_sales_forecast_2025_2035.csv", index=False)
print("✅ Forecast CSV generated.")
```