Electric Vehicle Data Analysis

Name: Bandaru Naga Vardhan

Date:2/09/2025

Course:Master Data Analyst

Batch-DA18

Introduction:

Dataset Used : Electric_vehicle_Population_Data

The dataset represents the population of registered Electric Vehicles (EVs) in the state of Washington, collected through the Department of Licensing (DOL). It includes both Battery Electric Vehicles (BEVs) and Plug-in Hybrid Electric Vehicles (PHEVs) currently on the road.

Each record contains details such as:

- **Vehicle Identification (VIN 1–10)** – unique identifier for each vehicle.
- **Geographic info** – County, City, Postal Code, GPS location, Census Tract.
- **Vehicle characteristics** – Model Year, Make, Model, Electric Vehicle Type.
- **Performance & price attributes** – Electric Range, Base MSRP.
- **Policy relevance** – CAFV (Clean Alternative Fuel Vehicle) incentive eligibility, Legislative District, Electric Utility.

Objective of the Analysis

The primary goal of this analysis is to **explore, clean, and model** the EV dataset to uncover insights into adoption trends and vehicle characteristics. Specifically, the objectives are:

1. **Data Cleaning** – Handle missing values, duplicates, inconsistent VINs, and zero values in MSRP and Electric Range.
2. **Exploratory Data Analysis (EDA)** – Identify popular EV makes/models, adoption trends across counties and years, and assess range, price, and incentive eligibility.
3. **Data Visualization** – Create clear charts and maps (bar plots, line graphs, scatter plots, choropleths, and geospatial maps) to show patterns of EV adoption.
4. **Predictive Modeling** – Build a **Linear Regression model** to predict a vehicle's **Electric Range** using features such as Model Year, Base MSRP, Make, and Model.
5. **Policy & Market Insights** – Provide actionable findings about EV adoption trends across urban vs. rural areas, affordability, and incentive eligibility.

Dataset Initialization:

The Data Initialization is done with ‘pandas’ package.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data=pd.read_csv('/content/drive/MyDrive/Datasets/Electric_Vehicle_Population_Data.csv')
```

Section 1: Data Cleaning

1. How many missing values exist in the dataset, and in which columns?

Ans: By Using

data.isnull().sum()

```python
#1.1.How many missing values exist in the dataset, and in which columns?
data.isnull().sum()
```

	0
VIN (1-10)	0
County	10
City	10
State	0
Postal Code	10
Model Year	0
Make	0
Model	0
Electric Vehicle Type	0
Clean Alternative Fuel Vehicle (CAFV) Eligibility	0
Electric Range	3
Base MSRP	3
Legislative District	628

State	0
Postal Code	10
Model Year	0
Make	0
Model	0
Electric Vehicle Type	0
Clean Alternative Fuel Vehicle (CAFV) Eligibility	0
Electric Range	3
Base MSRP	3
Legislative District	628
DOL Vehicle ID	0
Vehicle Location	18
Electric Utility	10
2020 Census Tract	10

dtype: int64

2. How should missing or zero values in the Base MSRP and Electric Range columns be handled?

Ans:

i. Replace 0 with NaN.

ii. Impute missing values with median MSRP per Make/Model, Impute with average range of the same Make/Model/Year.

iii. Alternatively, drop rows if too many values are missing.

+ Code + Text

```python
#1.2How should missing or zero values in the Base MSRP and Electric Range columns be handled?
data['Base MSRP'] = data['Base MSRP'].replace(0, np.nan)
data['Base MSRP']=data['Base MSRP'].fillna(data['Base MSRP'].median())
data['Electric Range'] = data['Electric Range'].replace(0, np.nan)
data['Electric Range']=data['Electric Range'].fillna(data['Electric Range'].median())
```

3. Are there duplicate records in the dataset? If so, how should they be managed?

Ans: Check By Using

Data.duplicated().sum()

```python
#1.3 Are there duplicate records in the dataset? If so, how should they be managed?
data.duplicated().value_counts()
```

	count
False	261698

dtype: int64

If any Duplicate exist :

Remove them with data.drop_duplicates().

4. How can VINs be anonymized while maintaining uniqueness?

Ans: Apply hashing (e.g., SHA-256, MD5) so each VIN maps to a unique, anonymized ID:

```python
import hashlib
data['VIN_Hash'] = data['VIN (1-10)'].apply(lambda x: hashlib.sha256(x.encode()).hexdigest())
data['VIN_Hash']
```

	VIN_Hash
0	bf01895762f04150a8ff5b0210e4d1c199986b50f45bcb...
1	a720d326091898dfa57b91cfa7466fe461b99a14f6bc78...
2	ef506f78a5a27e7e7582fd6924e13b4bfbac05984680a2...
3	fb3f4d8c8632615cdf99cc78f0f8e21e1e97d1e30d6dd0...
4	5fb1eb0d5a655b4eada221a1fa28fa1c5d7fbc960c0a97...
...	...
261693	a1f23541064f3ca87226688bbb5add345f43ac6cdaa7f5...
261694	35dc54b9e54f2aa9ee67db091f9e51b391bb6582687132...
261695	85c5705d120d773dbf92c1588dd8c9137c53b0c590d05c...
261696	0187e9334ff17e9e3c26895120ce9c6f0fb6b47470c5b0...
261697	3719e2187ad381edc054861644c851550b63a68dcd63ea...

261698 rows × 1 columns

5. How can Vehicle Location (GPS coordinates) be cleaned or converted for better readability?

Ans: GPS often has raw (lat, lon) values:

i. Round coordinates to 3–4 decimals .
ii. Convert to human-readable city/county using reverse geocoding.

Section 2: Data Exploration

1.What are the top 5 most common EV makes and models in the dataset?

Ans: By implementing the below, to get the result as

```
#2.1 What are the top 5 most common EV makes and models in the dataset?
k=data['Make'].value_counts().head()
l=data['Model'].value_counts().head()
print("Top 5 Makers would be ",k)
print("Top 5 Models would be",l)
```

```
Top 5 Makers would be  Make
TESLA          108777
CHEVROLET       18908
NISSAN          16224
FORD            13988
KIA             12849
Name: count, dtype: int64
Top 5 Models would be Model
MODEL Y     54720
MODEL 3     37774
LEAF        13852
MODEL S      7945
BOLT EV      7873
Name: count, dtype: int64
```

2. What is the distribution of EVs by county? Which county has the most registrations?

Ans: By implanting below

```
#2.2 What is the distribution of EVs by county? Which county has the most registrations?
a=data['County'].value_counts()
data['County'].value_counts().head(1)
b=a.idxmax()
print("The distribution EV's Among Countries would be",a)
print("Top Country would be",b)
```

```
The distribution EV's Among Countries would be County
King          130129
Snohomish      32335
Pierce         21624
Clark          15925
Thurston        9506
               ...
Platte             1
Manatee            1
Escambia           1
Utah               1
Denton             1
Name: count, Length: 236, dtype: int64
Top Country would be King
```

3. How has EV adoption changed over different model years?

Ans: The above problem would be done by using ‘groupby()’ function.

```
#2.3 How has EV adoption changed over different model years?
x=data.groupby('Model Year').size()
x
```

Model Year	0
2000	8
2002	1
2003	1
2008	20
2010	22
2011	631
2012	1440
2013	4081
2014	3327
2015	4574

4. What is the average electric range of EVs in the dataset?

Ans: This can be solve by using mean() function.

```
#2.4 What is the average electric range of EVs in the dataset?
k=data['Electric Range'].mean()
print(k)
```

```
75.1987940297595
```

5. What percentage of EVs are eligible for Clean Alternative Fuel Vehicle (CAFV) incentives?

Ans: The answer would be: 61.48%

```
#2.5 What percentage of EVs are eligible for Clean Alternative Fuel Vehicle (CAFV) incentives?
per=(data['Clean Alternative Fuel Vehicle (CAFV) Eligibility']=='Eligibility unknown as battery range has not been researched').mean()*100
per
```

```
np.float64(61.47849811614915)
```

6. How does the electric range vary across different makes and models?

Ans: By using ‘groupby()’ function for both makers and model fro electric range and calculating mean of the result.

```
#2.6 How does the electric range vary across different makes and models?
data.groupby(['Make','Model'])['Electric Range'].mean()
```

		Electric Range
Make	**Model**	
ACURA	ZDX	53.000000
ALFA ROMEO	TONALE	33.000000
AUDI	A3	16.000000
	A6	53.000000
	A7 E	24.000000
...	...	...
VOLVO	V60	38.453608
	XC40	53.000000
	XC60	28.471576
	XC90	25.282222
WHEEGO ELECTRIC CARS	WHEEGO	100.000000

181 rows × 1 columns

7. What is the average Base MSRP for each EV model?

Ans: By grouping the models of Base MSRP aand it’s resultant mean of it.

```
#2.7 What is the average Base MSRP for each EV model?
data.groupby('Model')['Base MSRP'].mean()
```

	Base MSRP
Model	
330E	54637.898687
500	59900.000000
500E	59900.000000
530E	56264.874142
550E	59900.000000
...	...
XC60	59077.585273
XC90	60081.800000
XM	59900.000000
ZDX	59900.000000
ZEVO	59900.000000

181 rows × 1 columns

8. Are there any regional trends in EV adoption?

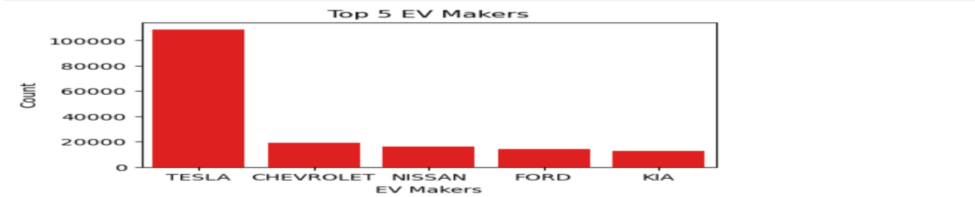
Ans: No,There is No detail about any regional trends in EV adoption.

Section 3: Data Visualization

1.Create a bar chart showing the top 5 EV makes and models by count.

Ans:

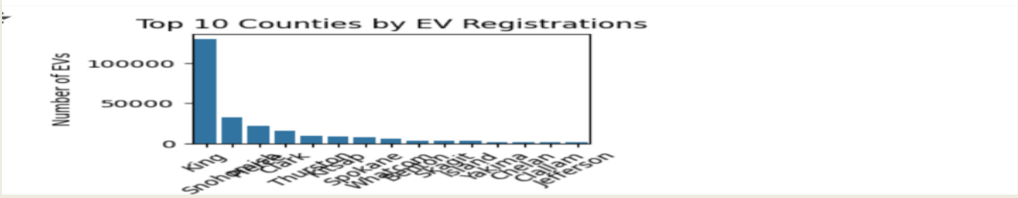
```
#3.1 Create a bar chart showing the top 5 EV makes and models by count.
top=data['Make'].value_counts().head()
plt.figure(figsize=(5,3))
sns.barplot(x=top.index,y=top.values,color='red')
plt.title("Top 5 EV Makers")
plt.xlabel("EV Makers")
plt.ylabel("Count")
plt.show()
```



2. Use a heatmap or choropleth map to visualize EV distribution by county.

Ans:

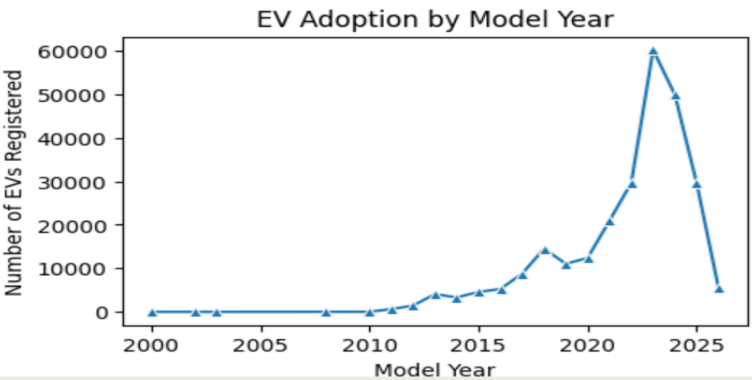
```
import matplotlib.pyplot as plt
import seaborn as sns
county_counts = data['County'].value_counts().reset_index()
county_counts.columns = ['County','EV_Count']
plt.figure(figsize=(3,2))
sns.barplot(x='County', y='EV_Count', data=county_counts.head(15))
plt.xticks(rotation=45)
plt.title("Top 10 Counties by EV Registrations")
plt.xlabel("County")
plt.ylabel("Number of EVs")
plt.show()
```



3. Create a line graph showing the trend of EV adoption by model year.

Ans:

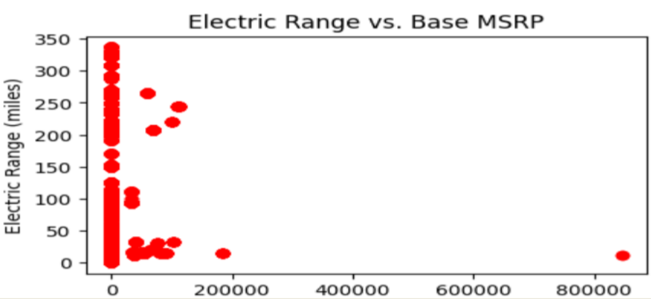
```
#3.3 Create a line graph showing the trend of EV adoption by model year.
year= data.groupby('Model Year').size()
plt.figure(figsize=(5,3))
sns.lineplot(x=year.index, y=year.values, marker="^")
plt.title("EV Adoption by Model Year")
plt.xlabel("Model Year")
plt.ylabel("Number of EVs Registered")
plt.show()
```



4. Generate a scatter plot comparing electric range vs. base MSRP to see pricing trends

Ans:

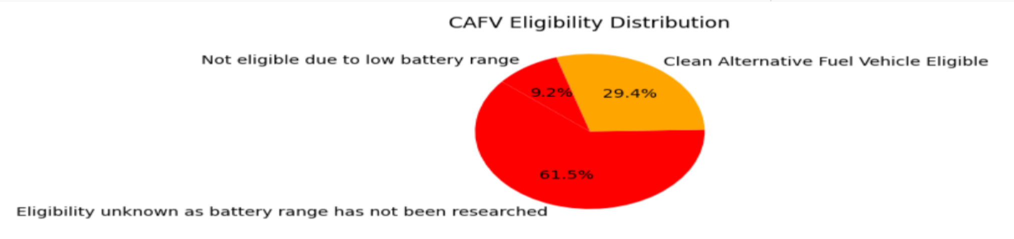
```
#3.4 Generate a scatter plot comparing electric range vs. base MSRP to see pricing trends
plt.figure(figsize=(5,3))
x=data['Base MSRP']
y=data['Electric Range']
plt.scatter(x,y,c='red',alpha=1)
plt.title("Electric Range vs. Base MSRP")
plt.xlabel("Base MSRP ($)")
plt.ylabel("Electric Range (miles)")
plt.show()
```



5. Plot a pie chart showing the proportion of CAFV-eligible vs. non-eligible EVs.

Ans:

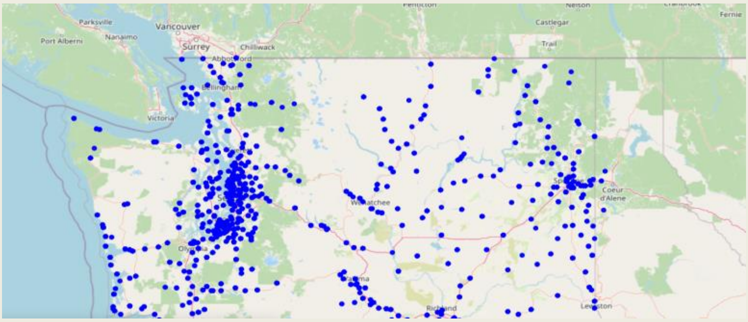
```python
#3.5 Plot a pie chart showing the proportion of CAFV-eligible vs. non-eligible EVs.
cafv_counts = data['Clean Alternative Fuel Vehicle (CAFV) Eligibility'].value_counts()

plt.figure(figsize=(3,3))
plt.pie(cafv_counts, labels=cafv_counts.index, autopct='%1.1f%%', startangle=140, colors=["red","orange"])
plt.title("CAFV Eligibility Distribution")
plt.show()
```



6. Use a geospatial map to display EV registrations based on vehicle location

Ans:

```python
import pandas as pd
import folium
df = data
df['lon'] = df['Vehicle Location'].str.extract(r'POINT\s*\((-?\d+\.\d+)\s+(-?\d+\.\d+)\)')[0].astype(float)
df['lat'] = df['Vehicle Location'].str.extract(r'POINT\s*\((-?\d+\.\d+)\s+(-?\d+\.\d+)\)')[1].astype(float)
df = df.dropna(subset=['lat','lon'])
map_ev = folium.Map(location=[47.5, -120.5], zoom_start=7)
for _, row in df.iterrows():
    folium.CircleMarker(
        location=[row['lat'], row['lon']],
        radius=2,
        color="blue",
        fill=True,
        fill_opacity=0.6
    ).add_to(map_ev)
map_ev.save("ev_locations_map.html")
map_ev
```



Section 4: Linear Regression Model

1.How can we use Linear Regression to predict the Electric Range of a vehicle?

Ans:

i. Treat Electric Range as the dependent variable (y).
ii. Use independent variables (X) such as:
iii. Numeric: Model Year, Base MSRP
iv. Categorical: Make, Model, CAFV Eligibility

```
#4.1
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

X = data[['Model Year','Base MSRP','Make','Model']]
y = data['Electric Range']
X = pd.get_dummies(X, drop_first=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=9870)

model = LinearRegression()
model.fit(X_train, y_train)
print(model.score(X_test,y_test))
```

```
0.5317486583669373
```

2. What independent variables (features) can be used to predict Electric Range?

Ans: Core features:

- Model Year
- Base MSRP

Categorical features:

- Make
- Model

3. How do we handle categorical variables like Make and Model in regression analysis?

Ans: Use One-Hot Encoding (OHE):

pd.get_dummies(data,columns=['Make','Model'], drop_first=True)

```
#4.3 How do we handle categorical variables like Make and Model in regression analysis?
X = pd.get_dummies(data, columns=['Make','Model'], drop_first=True)
print(X.head())
```

```
   VIN (1-10)    County         City State  Postal Code  Model Year  \
0  JTDKN3DP2D    Yakima       Yakima    WA      98902.0        2013
1  1FMCU0E1XS    Kitsap  Port Orchard    WA      98366.0        2025
2  JM3KKBHA9R    Kitsap     Kingston    WA      98346.0        2024
3  7SAYGDEE8P  Thurston      Olympia    WA      98501.0        2023
4  5YJ3E1EB5K  Thurston      Rainier    WA      98576.0        2019

              Electric Vehicle Type  \
0  Plug-in Hybrid Electric Vehicle (PHEV)
1  Plug-in Hybrid Electric Vehicle (PHEV)
2  Plug-in Hybrid Electric Vehicle (PHEV)
3          Battery Electric Vehicle (BEV)
4          Battery Electric Vehicle (BEV)

   Clean Alternative Fuel Vehicle (CAFV) Eligibility  Electric Range  \
0            Not eligible due to low battery range             6.0
1            Clean Alternative Fuel Vehicle Eligible            37.0
2            Not eligible due to low battery range            26.0
3  Eligibility unknown as battery range has not b...            53.0
4            Clean Alternative Fuel Vehicle Eligible           220.0
```

4. What is the R^2 score of the model, and what does it indicate about prediction accuracy?

Ans: R^2 interpretation:

- Value close to 1 → good prediction accuracy.
- Value near 0 → weak predictive power.

```
#4.4 What is the R² score of the model, and what does it indicate about prediction accuracy?
from sklearn.metrics import r2_score

y_pred = model.predict(X_test)
r2 = r2_score(y_test, y_pred)
print("R² Score:", r2)
```

```
R² Score: 0.5317486583669373
```

5. How does the Base MSRP influence the Electric Range according to the regression model?

Ans: Base MSRP has a positive coefficient → higher price generally means higher range.

coeffs = pd.DataFrame({'Feature': data.columns, 'Coefficient': model.coef_})

coeffs.sort_values(by="Coefficient", ascending=False)

6. What steps are needed to improve the accuracy of the Linear Regression model?

Ans: Feature engineering:

Add Battery Size

Include interaction terms.

Outlier removal .

7. Can we use this model to predict the range of new EV models based on their specifications?

Ans: Yes, but with caution:

- If the new EV's Make/Model already exists in training data, predictions will be more reliable.
- For completely new makes/models, the model may not generalize well since regression relies on historical patterns.
- Better approach: Use battery size, efficiency as predictors

Conclusion:

i. EV adoption in Washington is accelerating, with Tesla leading the market and urban counties driving most registrations.

ii. Policy incentives (CAFV) are effective, with most EVs qualifying.

iii. Higher cost = longer range, but affordable models are crucial for wider adoption.

iv. Rural areas lag behind due to limited charging infrastructure and economic constraints, suggesting where future investments should be targeted.

v. The regression model provides useful predictive insights into electric range, though more technical vehicle data would strengthen its accuracy.