

Trực quan hóa dữ liệu

PHÂN TÍCH VÀ DỰ ĐOÁN

XE ĐIỆN

Ở TIỂU BANG WASHINGTON

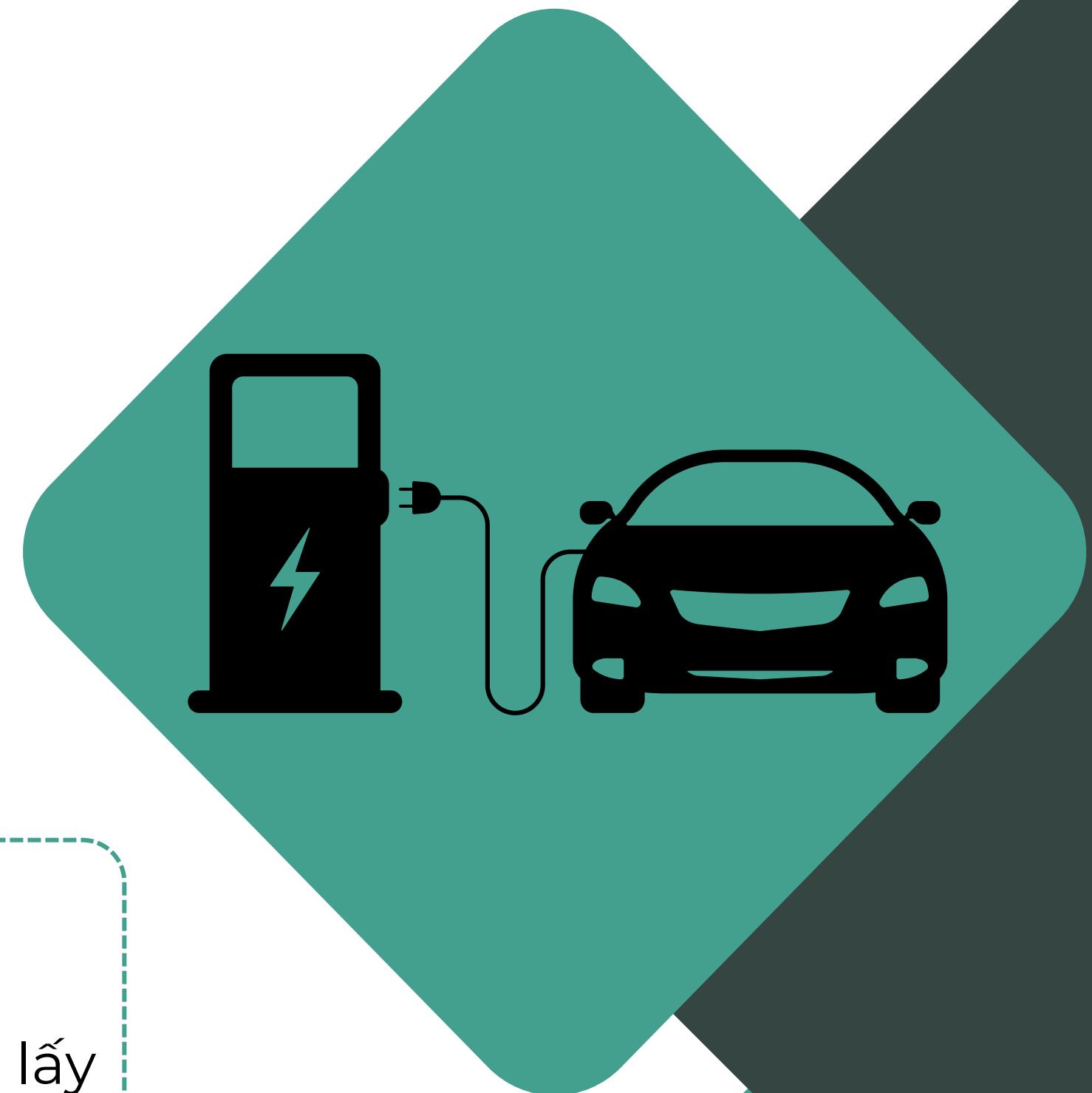


Tổng quan đề tài

Thị trường xe điện đang chứng kiến sự tăng trưởng mạnh mẽ ngành xe điện đang ngày càng được quan tâm.

Tài nguyên sử dụng

- Ngôn ngữ lập trình: Python.
- Bộ dữ liệu “Electric Vehicle Population Data” được lấy trên <https://data.wa.gov/> là nơi cung cấp dữ liệu mở của bang Washington.



Tổng quan dữ liệu

Bộ dữ liệu "Electric Vehicle Data" được sử dụng trong báo cáo lấy từ trang web [data.wa.gov](#) - một trang web phổ biến với hàng trăm bộ dữ liệu phân tích. Nó cung cấp thông tin về các xe điện được đăng ký thông qua Bộ cấp phép Tiểu bang Washington.

1997 đến 2024

153830 chiếc xe điện khác nhau

Mô tả dữ liệu

Tiền xử lý dữ liệu

Các bước tiền xử lý dữ liệu

- Tổng quan dữ liệu trước khi xử lý**
- Xử lý missing data**
- Xử lý outlier**
- Tổng quan dữ liệu sau khi xử lý**



Tổng quan dữ liệu

VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type
5UXTA6C05P	Yakima	Yakima	WA	98903	2023	BMW	X5	Plug-in Hybrid Electric Vehicle (PHEV)
5YJRE11B48			BC		2008	TESLA	ROADSTER	Battery Electric Vehicle (BEV)
5YJSA1E24G	King	Seattle	WA	98103	2016	TESLA	MODEL S	Battery Electric Vehicle (BEV)
1N4AZ1CP5J	King	Shoreline	WA	98177	2018	NISSAN	LEAF	Battery Electric Vehicle (BEV)
5YJ3E1EA6J	Island	Coupeville	WA	98239	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)

Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	Vehicle Location	Electric Utility	2020 Census Tract
30	0	14	227153587	POINT (-120.477805 46.553505)	PACIFICORP	53077002803
220	98950		143609049			
210	0	43	187728201	POINT (-122.34301 47.659185)	CITY OF SEATTLE - (WA) CITY OF TACOMA - (WA)	53033004902
151	0	32	249867971	POINT (-122.382425 47.77279)	CITY OF SEATTLE - (WA) CITY OF TACOMA - (WA)	53033020100
215	0	10	223792649	POINT (-122.6880708 48.2179983)	PUGET SOUND ENERGY INC	53029971100

Tổng quan dữ liệu

In [9]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 153830 entries, 0 to 153829
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   VIN (1-10)      153830 non-null   object 
 1   County          153827 non-null   object 
 2   City            153827 non-null   object 
 3   State           153830 non-null   object 
 4   Postal Code    153827 non-null   float64
 5   Model Year     153830 non-null   int64  
 6   Make            153830 non-null   object 
 7   Model           153830 non-null   object 
 8   Electric Vehicle Type  153830 non-null   object 
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility 153830 non-null   object 
 10  Electric Range  153830 non-null   int64  
 11  Base MSRP       153830 non-null   int64  
 12  Legislative District  153491 non-null   float64
 13  DOL Vehicle ID  153830 non-null   int64  
 14  Vehicle Location 153823 non-null   object 
 15  Electric Utility 153827 non-null   object 
 16  2020 Census Tract 153827 non-null   float64
dtypes: float64(3), int64(4), object(10)
memory usage: 20.0+ MB
```

Mô tả dữ liệu có kiểu Numerical

- 7 cột có định dạng dữ liệu là Numeric gồm 3 cột ‘Model Year’, ‘Electric range’, ‘Base MSRP’ định lượng 4 cột còn lại định danh
- Thực hiện quan sát trên 3 biến định lượng

	Model Year	Electric Range	Base MSRP
count	153830.000000	153830.000000	153830.000000
mean	2020.100780	65.727673	1273.032276
std	3.019617	95.147219	9086.044139
min	1997.000000	0.000000	0.000000
25%	2018.000000	0.000000	0.000000
50%	2021.000000	17.000000	0.000000
75%	2023.000000	84.000000	0.000000
max	2024.000000	337.000000	845000.000000

Mô tả dữ liệu có kiểu Categorical

County

```
df['County'].nunique()
```

178

	Count	Percentage (%)
King	80637	52.420576
Snohomish	17727	11.523985
Pierce	11804	7.673555
Clark	9066	5.893634
Thurston	5514	3.584546
...
Las Animas	1	0.000650
Bell	1	0.000650
Rock Island	1	0.000650
Hawaii	1	0.000650

City

```
df['city'].nunique()
```

684

	City	Percentage (%)
Seattle	26153	17.001567
Bellevue	7810	5.077132
Redmond	5597	3.638503
Vancouver	5429	3.529289
Bothell	4930	3.204899
...
Hanscom Afb	1	0.000650
Minneapolis	1	0.000650
Burr Ridge	1	0.000650
Bridgeport	1	0.000650

State

```
data['state'].nunique()
```

43

	Count	Percentage (%)
WA	153491	99.779627
CA	91	0.059156
VA	34	0.022102
MD	33	0.021452
TX	19	0.012351
IL	13	0.008451
NC	13	0.008451
CO	11	0.007151
AZ	10	0.006501
FL	10	0.006501
NJ	8	0.005201
OR	8	0.005201
HI	8	0.005201
CT	7	0.004550
NY	7	0.004550
GA	6	0.003900
NE	3	0.001950
MA	3	0.001950
KY	3	0.001950
KS	2	0.001300
PA	2	0.001300
UT	2	0.001300
ID	2	0.001300
AR	2	0.001300
IN	2	0.001300
BC	2	0.001300
OH	2	0.001300
OK	1	0.000650

Mô tả dữ liệu có kiểu Categorical

Make

```
df['Make'].nunique()
df['Make'].unique()
```

36

```
array(['TESLA', 'NISSAN', 'BMW', 'AUDI', 'TOYOTA', 'KIA', 'FIAT', 'FORD',
       'CHEVROLET', 'HYUNDAI', 'VOLVO', 'VOLKSWAGEN', 'CHRYSLER', 'SMART',
       'RIVIAN', 'SUBARU', 'JEEP', 'HONDA', 'LINCOLN', 'LUCID', 'PORSCHE',
       'MITSUBISHI', 'POLESTAR', 'MERCEDES-BENZ', 'MINI', 'JAGUAR',
       'CADILLAC', 'LEXUS', 'GENESIS', 'WHEEGO ELECTRIC CARS', 'FISKER',
       'MAZDA', 'BENTLEY', 'TH!NK', 'LAND ROVER', 'AZURE DYNAMICS'],
      dtype=object)
```

Electric Vehicle Type

```
df['Electric Vehicle Type'].nunique()
df['Electric Vehicle Type'].unique()
```

2

```
array(['Plug-in Hybrid Electric Vehicle (PHEV)',
       'Battery Electric Vehicle (BEV)'],
      dtype=object)
```

Clean Alternative Fuel Vehicle (CAFV) Eligibility

```
df['CAFV'].nunique()
df['CAFV'].unique()
```

3

```
array(['Clean Alternative Fuel Vehicle Eligible',
       'Eligibility unknown as battery range has not been researched',
       'Not eligible due to low battery range'],
      dtype=object)
```

Electric Utility

```
df['Electric Utility'].nunique()
```

75

Xử lý missing data

```
print("Số lượng missing data: ")
print(data.isnull().sum())
✓ 0.2s

Số lượng missing data:
VIN (1-10)           0
County              3
City                3
State               0
Postal Code          3
Model Year           0
Make                0
Model               0
Electric Vehicle Type 0
Clean Alternative Fuel Vehicle (CAFV) Eligibility 0
Electric Range        0
Base MSRP             0
Legislative District  339
DOL Vehicle ID        0
Vehicle Location       7
Electric Utility        3
2020 Census Tract      3
dtype: int64
```

Số lượng missing data 361 << 5%



Xóa dữ liệu

Xử lý outlier

Các biến có kiểu dữ liệu số

```
print(data.select_dtypes(include=['int', 'float']).columns.tolist())
```

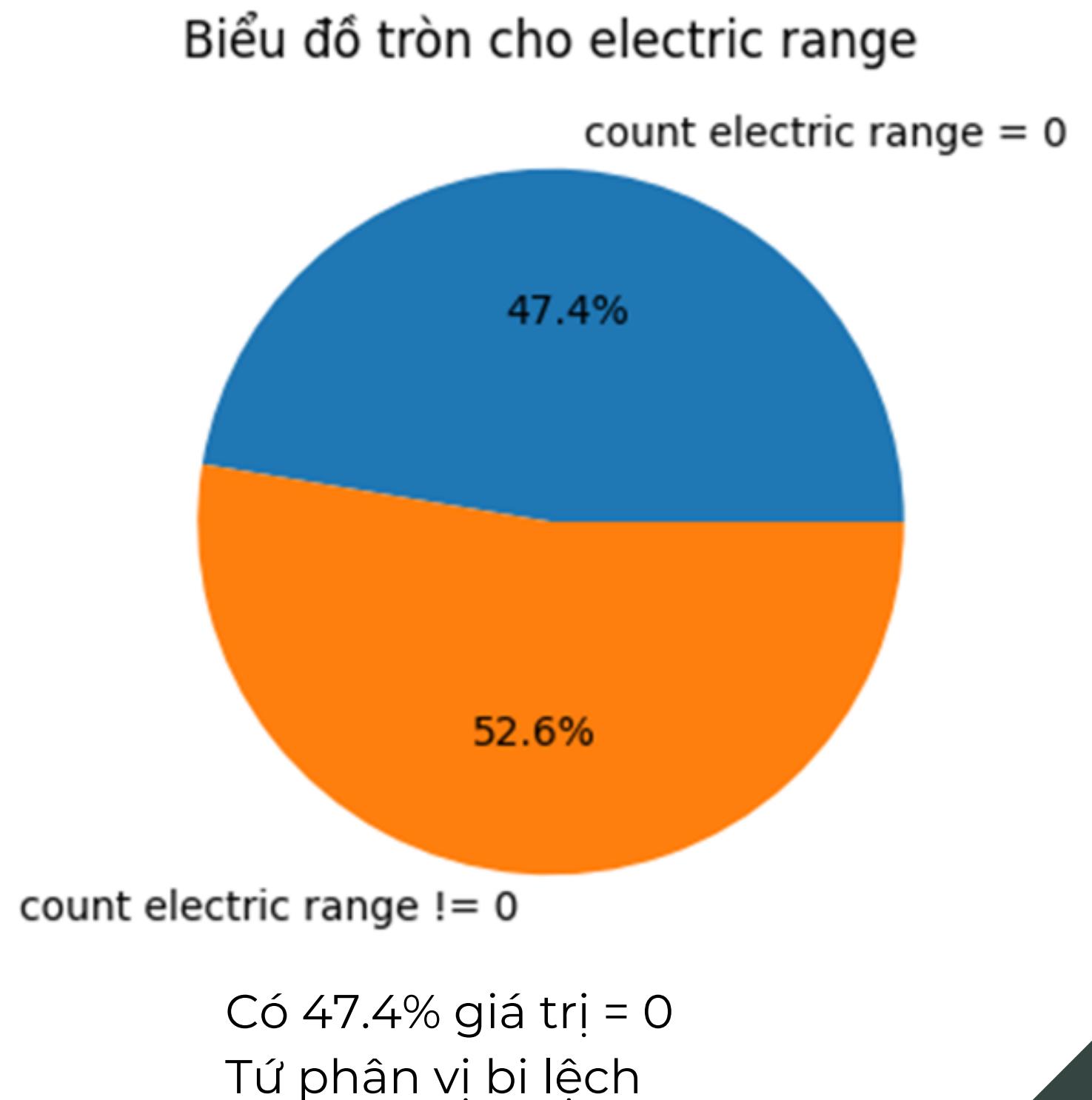
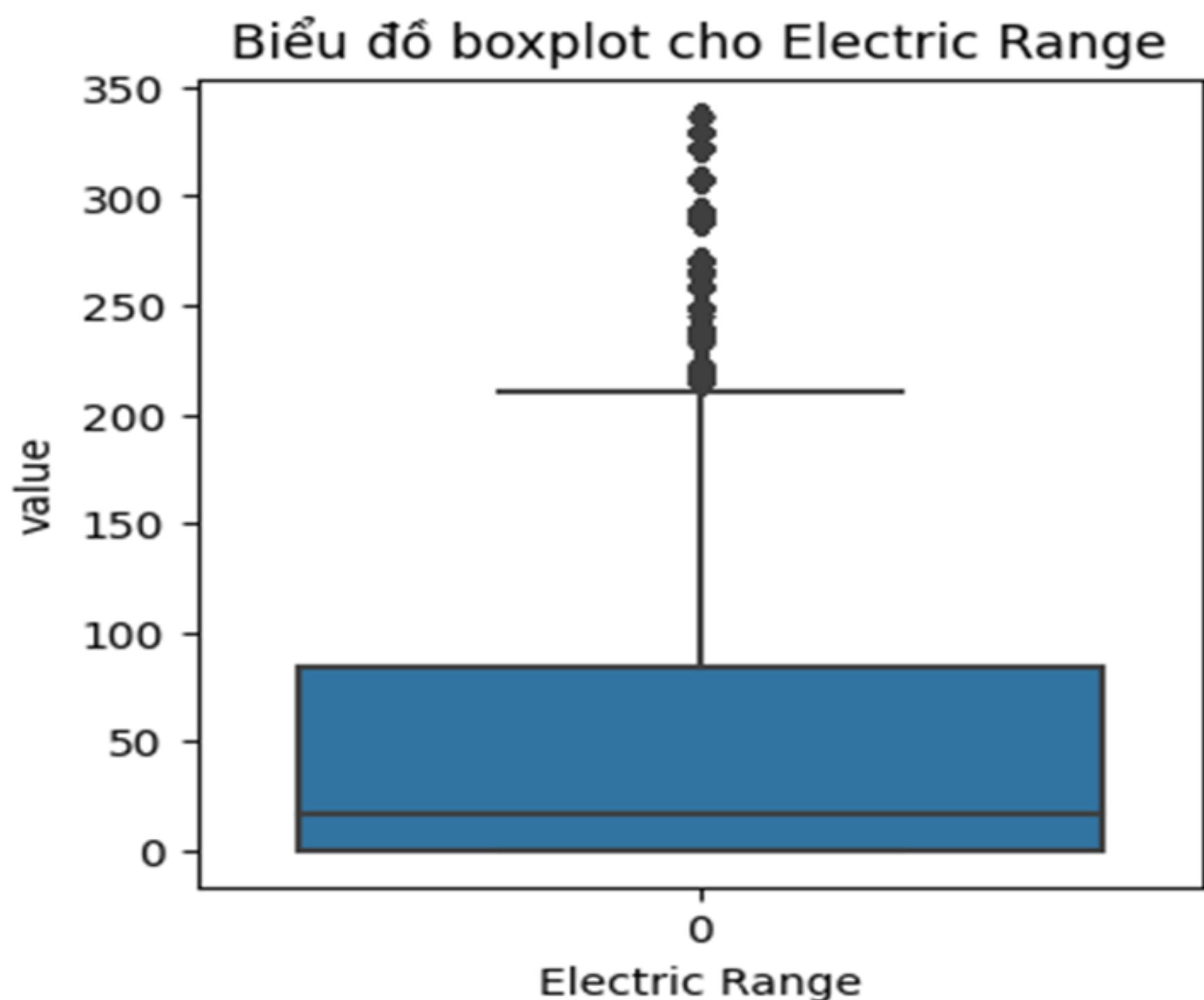
✓ 0.0s

```
['Postal Code', 'Model Year', 'Electric Range', 'Base MSRP', 'Legislative District', 'DOL Vehicle ID', '2020 Census Tract']
```

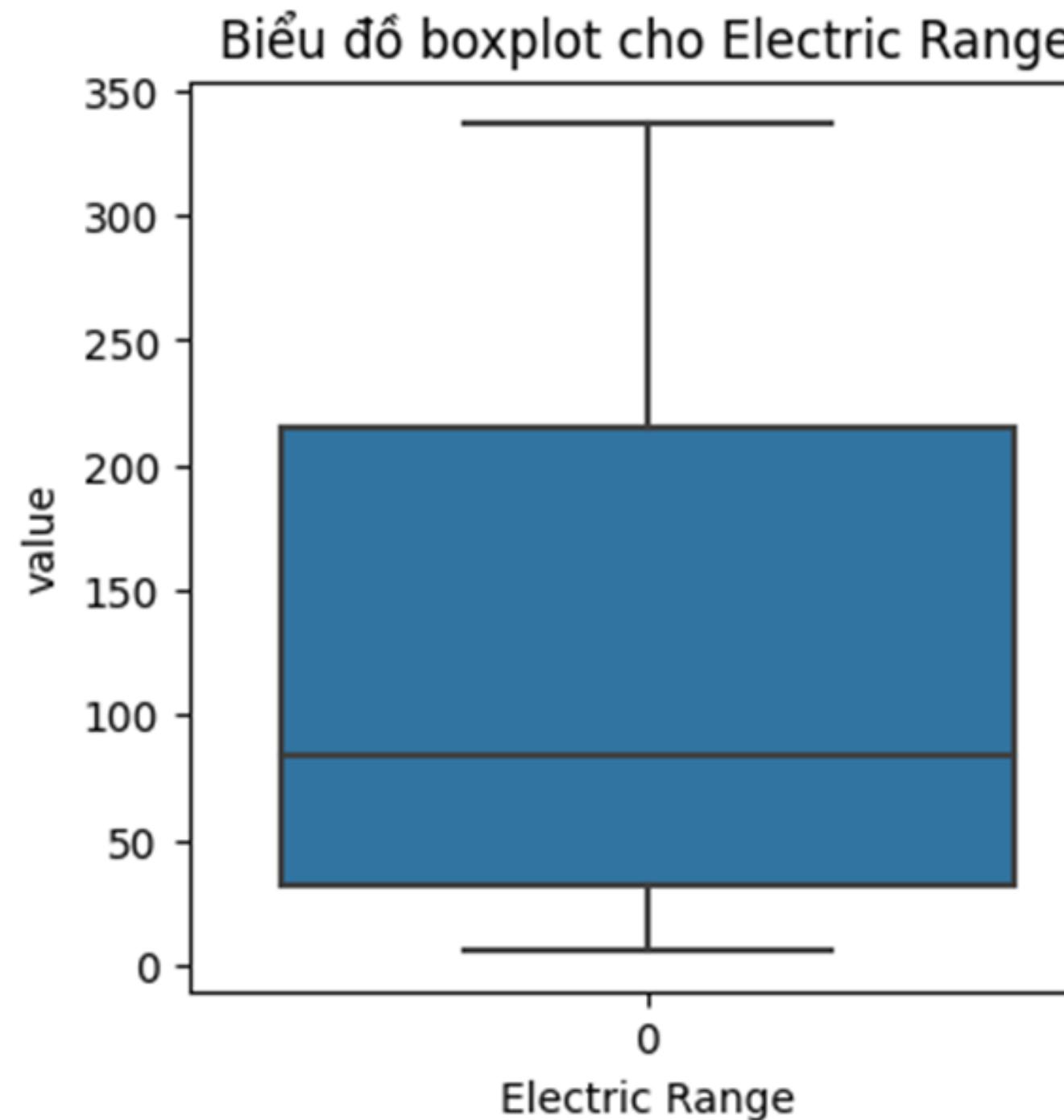


- Các biến 'Electric Range', 'Base MSRP' là các **biến định lượng**
- Các biến còn lại **biến định tính** thể hiện các đặc điểm, mã định danh, hoặc thông tin phân loại về địa lý
- Phân tích Outlier ở các biến định lượng

Xử lý outlier



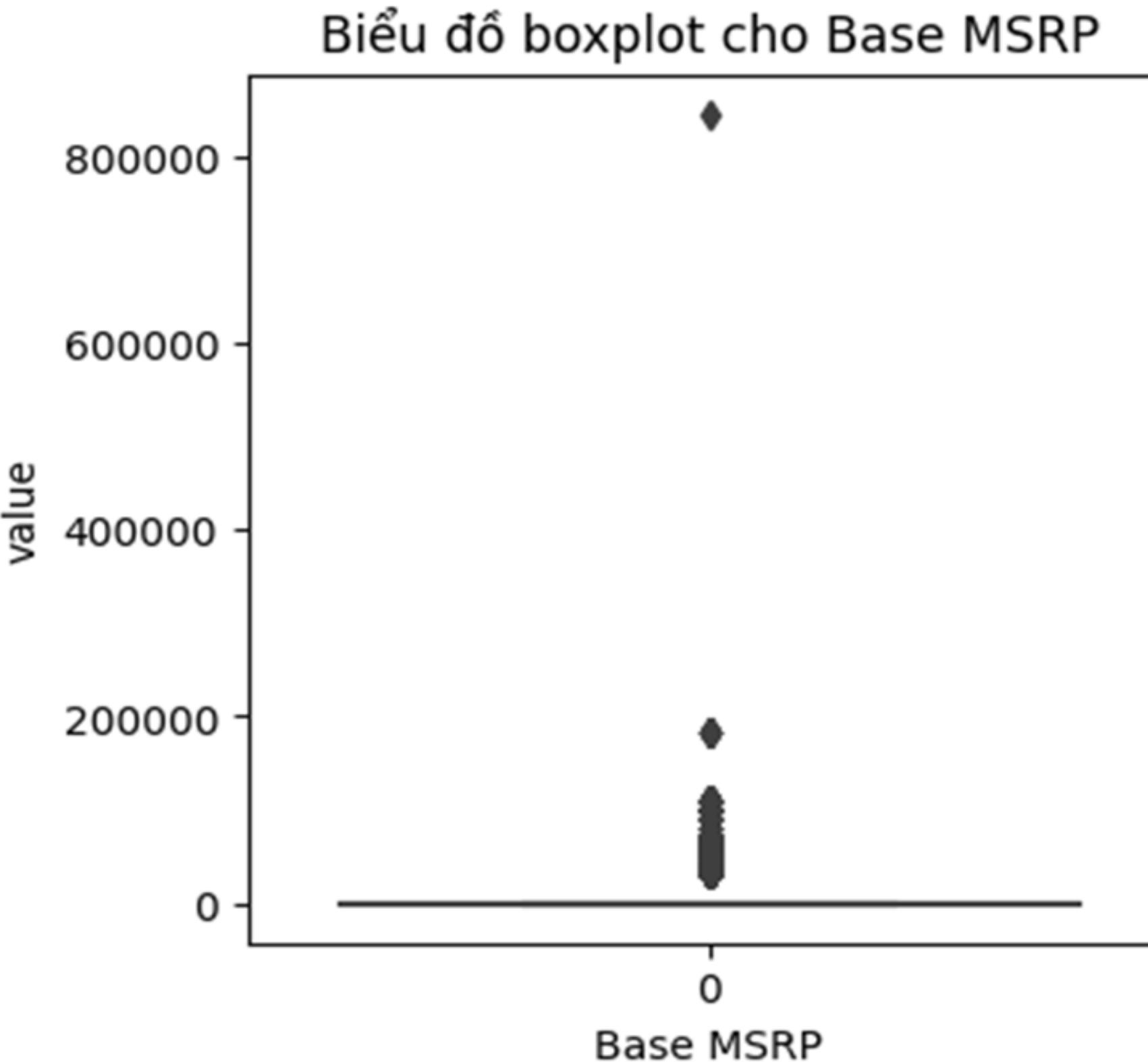
Xử lý outlier



Không xuất hiện outlier với các phần tử Electric Range khác 0

Không thực hiện xử lý outlier ở Electric Range

Xử lý outlier

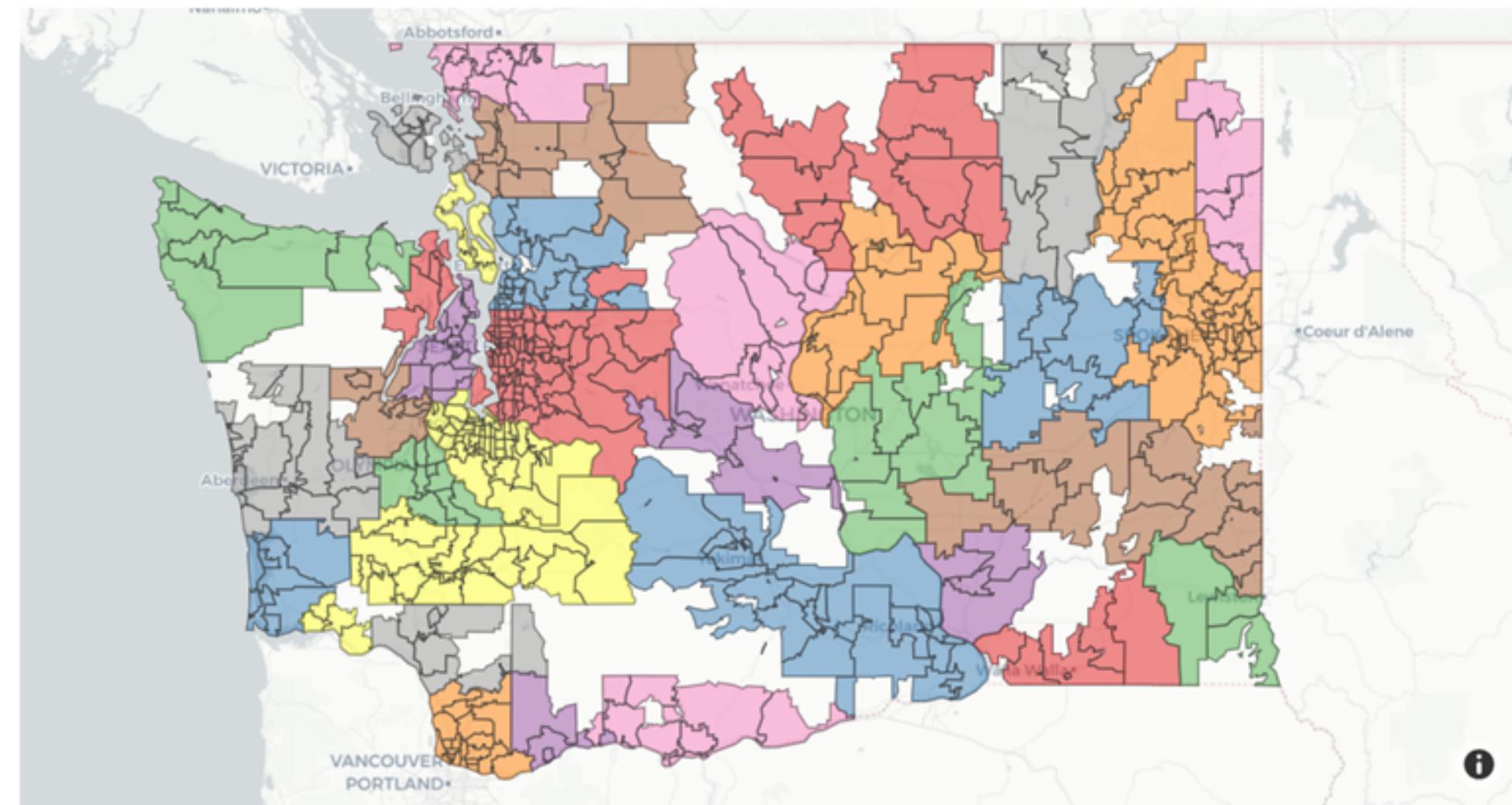


Xuất hiện nhiều giá trị 0

Chuẩn hóa nhị phân 0, 1

Xử lý outlier

County in Dataset



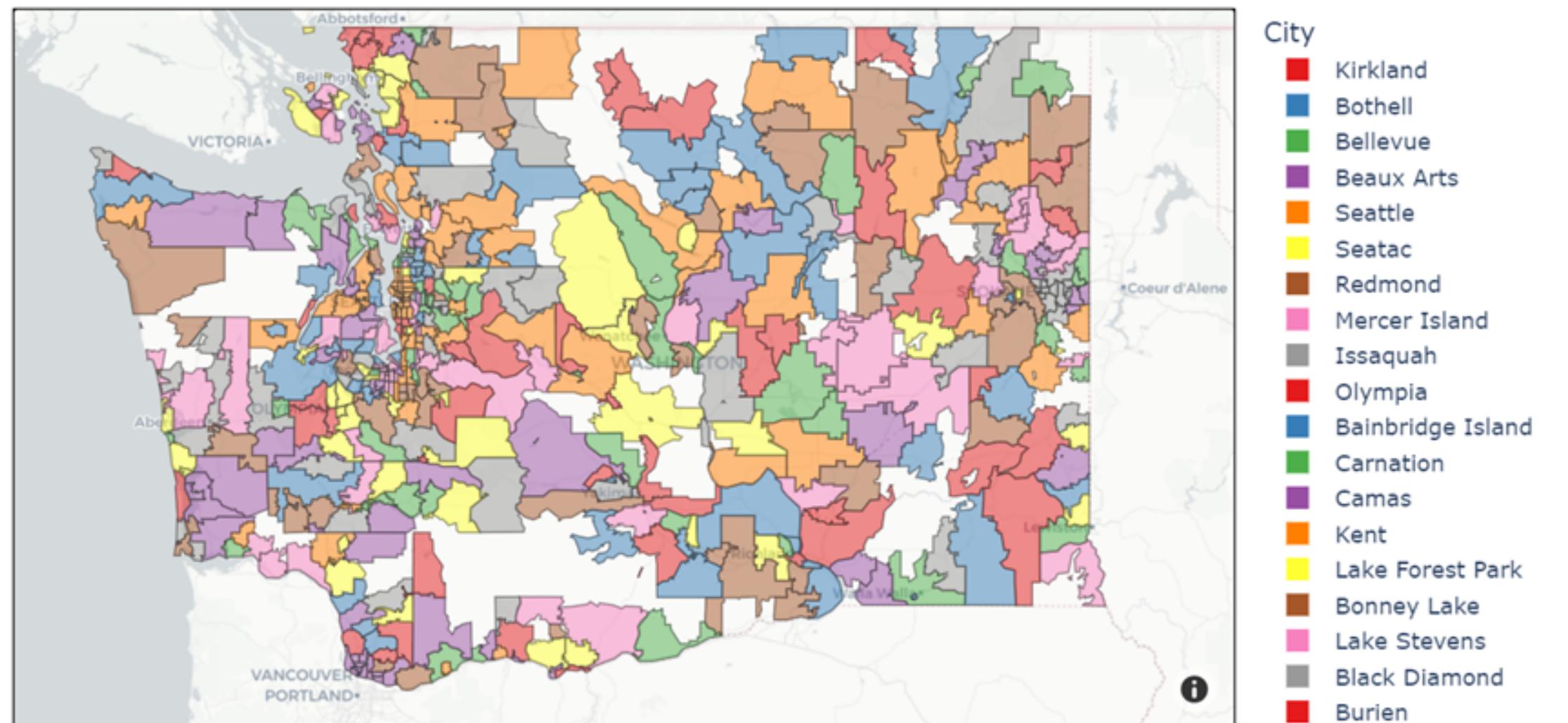
Không có county vi phạm lãnh
thổ Washington



Không có outlier ở biển County

Xử lý outlier

City in Dataset



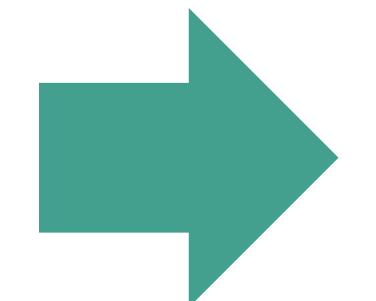
- Không có city vi phạm lãnh thổ washington
- Không có outlier ở biển City

Tách biến

Kết quả thu được

- Vehicle Location có định dạng là POINT (Longitude Latitude)
- phân tách cột dữ liệu thành 2 cột là Longitude và Latitude

```
phan_tach = data['Vehicle Location'].str.replace('POINT ', '')
phan_tach = phan_tach.str.replace('(', '')
phan_tach = phan_tach.str.replace(')', '')
data[['Longitude', 'Latitude']] = phan_tach.str.split(' ', 1, expand=True)
data[['Longitude', 'Latitude']] = data[['Longitude', 'Latitude']].astype(float)
```



```
print(data[['Longitude', 'Latitude']].info())
print("_____")
print(data[['Longitude', 'Latitude']].head(5))
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
Int64Index: 153487 entries, 0 to 153829
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Longitude   153487 non-null  float64 
 1   Latitude    153487 non-null  float64 
dtypes: float64(2)
memory usage: 3.5 MB
None

Longitude  Latitude
0 -120.477805 46.553505
2 -122.343010 47.659185
3 -122.382425 47.772790
4 -122.688071 48.217998
5 -122.847462 47.638360
```

Tổng quan dữ liệu sau xử lý

VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type	Clean Alternative Fuel Vehicle (CAFV) Eligibility
SUXTA6C05P	Yakima	Yakima	WA	98903	2023	BMW	X5	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible
5YJSA1E24G	King	Seattle	WA	98103	2016	TESLA	MODEL S	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible
1N4AZ1CP5J	King	Shoreline	WA	98177	2018	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible
5YJ3E1EA6J	Island	Coupeville	WA	98239	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible
1G1FW6S00H	Kitsap	Seabeck	WA	98380	2017	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible

Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	Vehicle Location	Electric Utility	2020 Census Tract	Binary Base MSRP	Longitude	Latitude
30	0	14	227153587	POINT (-120.477805 46.553505)	PACIFICORP	53077002803	0	-120.477805	46.553505
210	0	43	187728201	POINT (-122.34301 47.659185)	CITY OF SEATTLE - (WA) CITY OF TACOMA - (WA)	53033004902	0	-122.34301	47.659185
151	0	32	249867971	POINT (-122.382425 47.77279)	CITY OF SEATTLE - (WA) CITY OF TACOMA - (WA)	53033020100	0	-122.382425	47.77279
215	0	10	223792649	POINT (-122.6880708 48.2179983)	PUGET SOUND ENERGY INC	53029971100	0	-122.6880708	48.2179983
238	0	35	125032974	POINT (-122.847462 47.63836)	PUGET SOUND ENERGY INC	53035091301	0	-122.847462	47.63836

Tổng quan dữ liệu sau khi xử lý

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 153487 entries, 0 to 153829
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   VIN (1-10)      153487 non-null   object 
 1   County          153487 non-null   object 
 2   City            153487 non-null   object 
 3   State           153487 non-null   object 
 4   Postal Code    153487 non-null   float64
 5   Model Year     153487 non-null   int64  
 6   Make            153487 non-null   object 
 7   Model           153487 non-null   object 
 8   Electric Vehicle Type  153487 non-null   object 
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility 153487 non-null   object 
 10  Electric Range  153487 non-null   int64  
 11  Base MSRP       153487 non-null   int64  
 12  Legislative District  153487 non-null   float64
 13  DOL Vehicle ID  153487 non-null   int64  
 14  Vehicle Location 153487 non-null   object 
 15  Electric Utility 153487 non-null   object 
 16  2020 Census Tract 153487 non-null   float64
 17  Longitude        153487 non-null   float64
 18  Latitude         153487 non-null   float64
 19  Binary Base MSRP 153487 non-null   object 

dtypes: float64(5), int64(4), object(11)
memory usage: 24.6+ MB
```

Phân tích dữ liệu

Số lượng xe điện được đăng ký ở bang Washington

```
count_vehicle = data['DOL Vehicle ID'].count()  
print(f'Số lượng xe điện được đăng ký ở bang Washington: {count_vehicle}')
```

Số lượng xe điện được đăng ký ở bang Washington: 153487

Với biến DOL Vehicle ID là mã số được đại diện cho mỗi xe. Nhóm đã truy vấn được tất cả 153487 xe điện được đăng ký ở bang Washington.

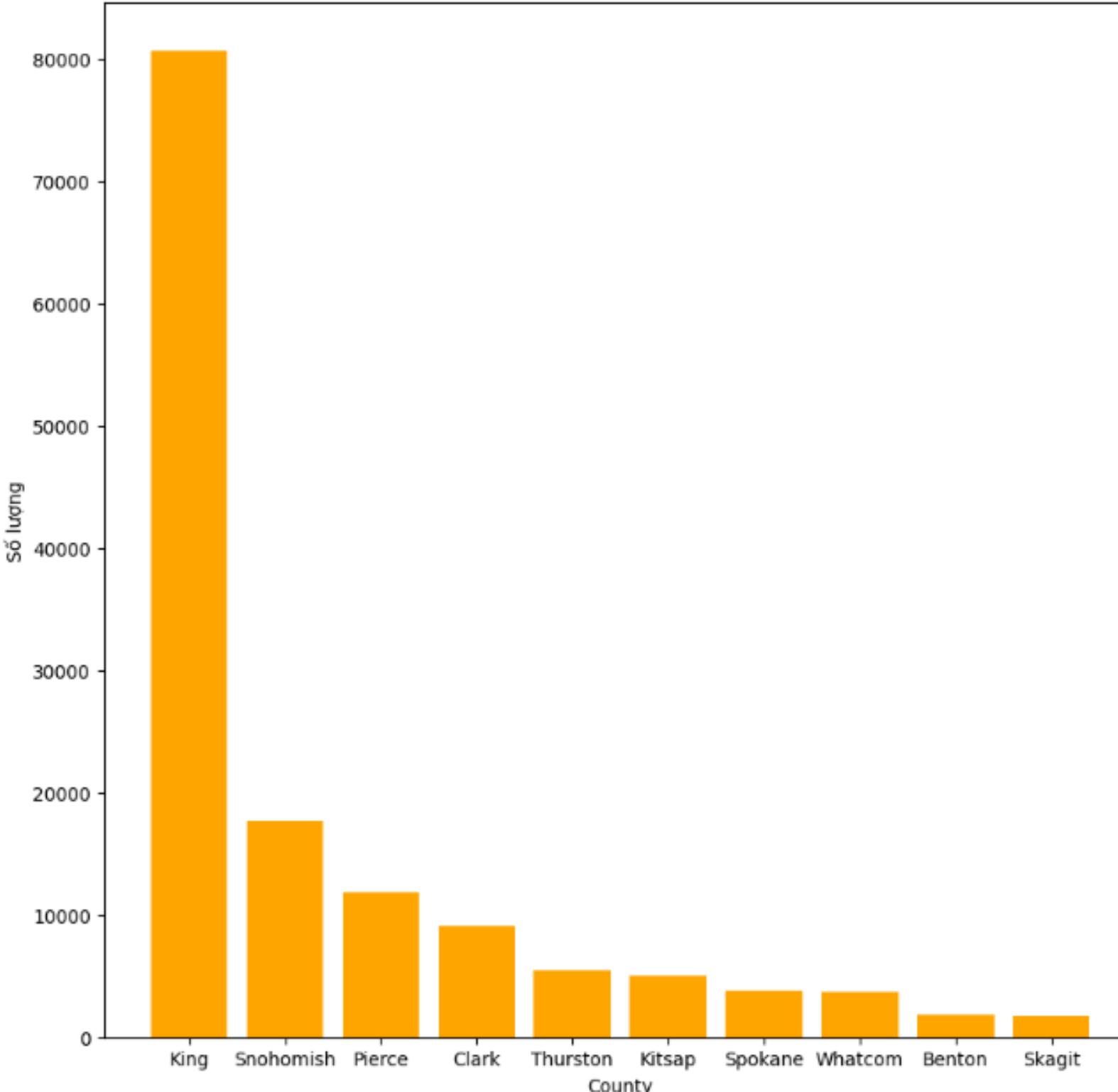
```
data_wa = data[data['State']=="WA"]  
evt_count = data_wa['Electric Vehicle Type'].value_counts()  
evt_count  
✓ 0.1s
```

Battery Electric Vehicle (BEV)	119173
Plug-in Hybrid Electric Vehicle (PHEV)	34314
Name: Electric Vehicle Type, dtype: int64	

Tổng số lượng của xe điện theo loại xe không có sự chênh lệch cao lắm giữa 2 loại xe này

Số lượng xe theo từng quận thuộc Washington

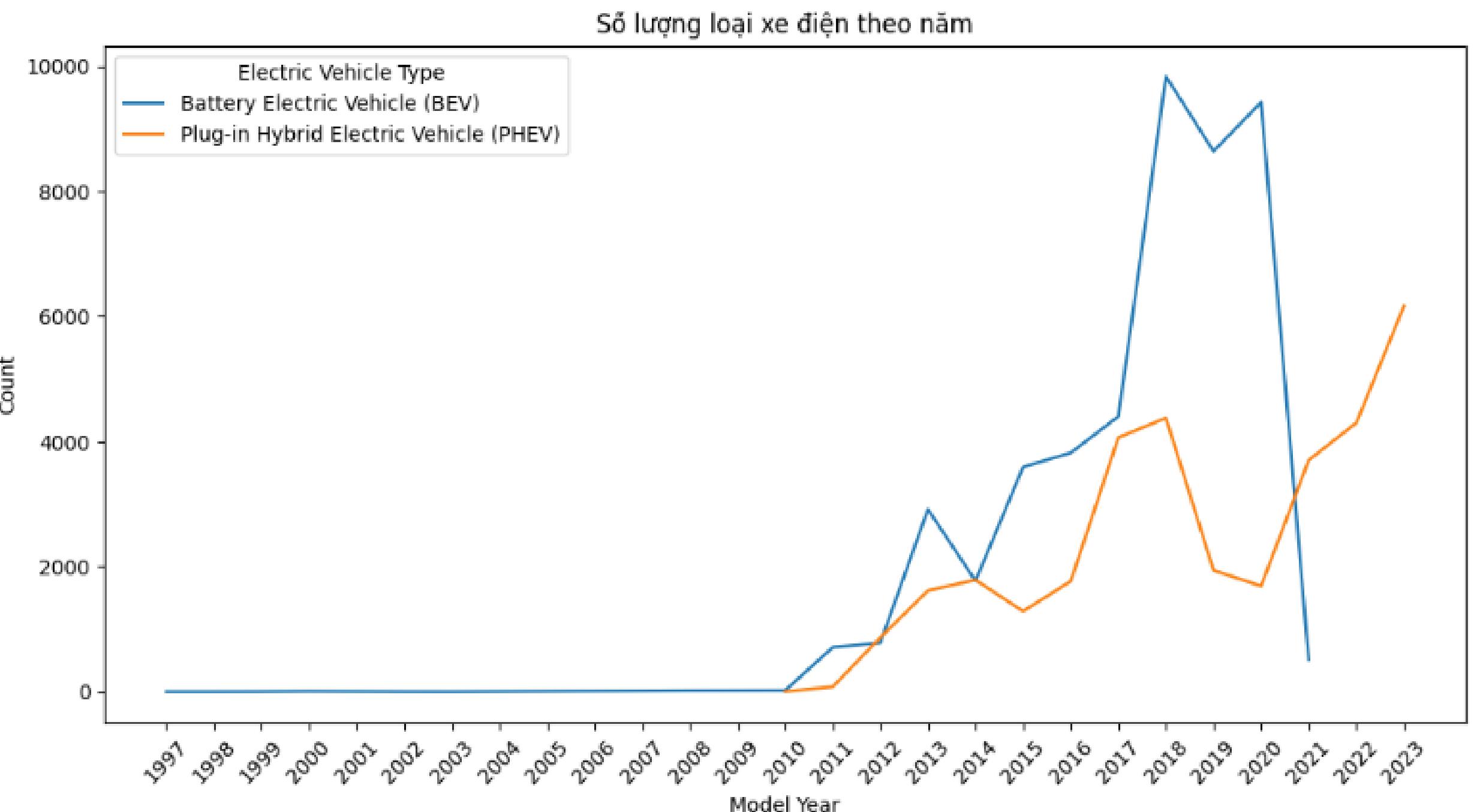
Biểu đồ thống kê số lượng xe theo County



- Quận King có số lượng xe cao nhất
- Quận Snohomish và Pierce cũng có số lượng xe khá cao.
- Các quận như Clark, Thurston, Kitsap, Spokane, Whatcom, Benton và Skagit có số lượng xe thấp hơn và khá tương đương nhau.
- Qua đó cho thấy **sự phổ biến của xe điện ở mỗi quận** có **sự phân tán khá rõ**.

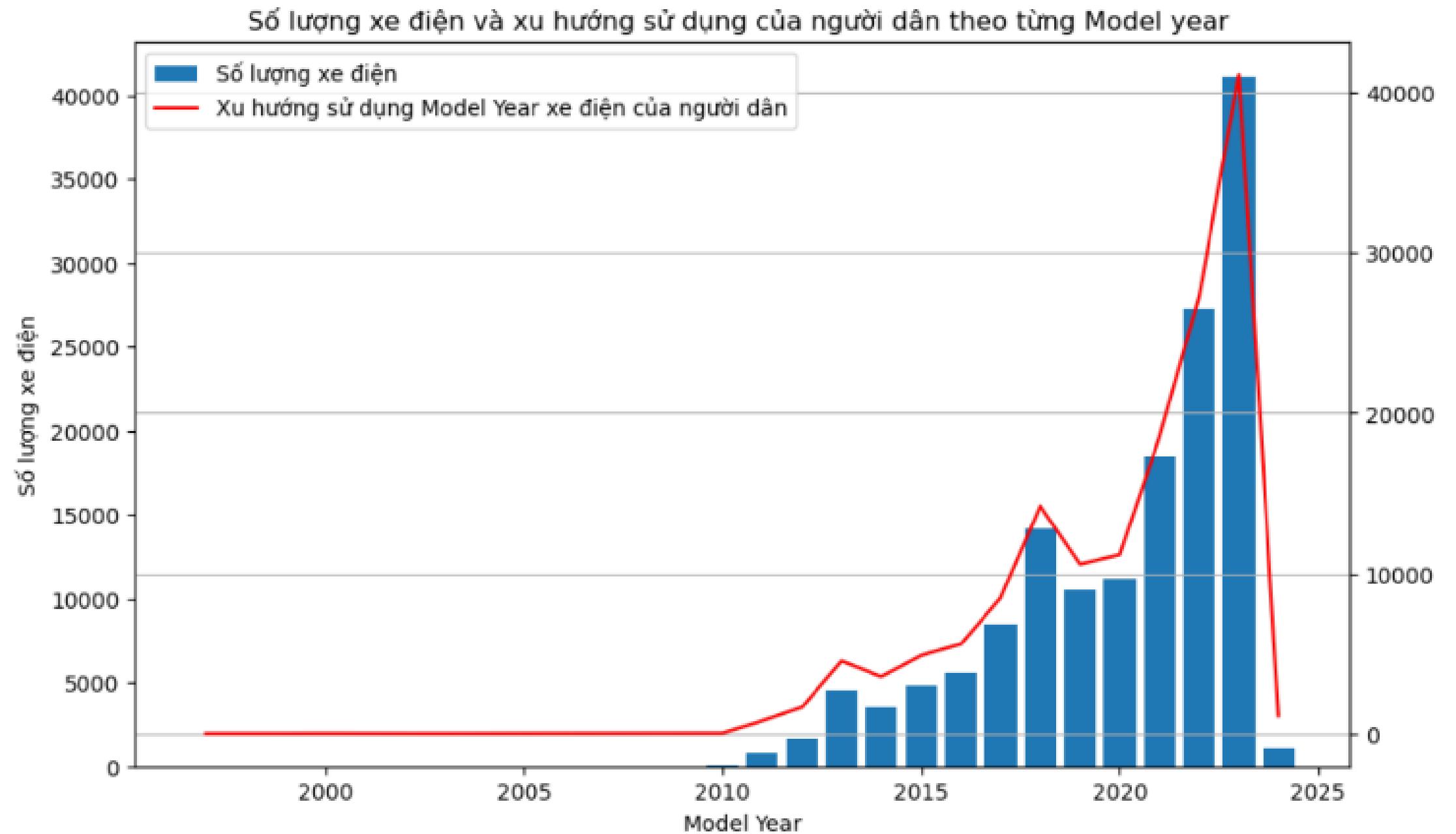


Phân loại xe điện ở tiểu bang Washington theo từng năm



- BEV và PHEV có **số lượng tương đối giống nhau** cho đến năm 2017
- Giai đoạn 2018 - 2020, số lượng BEV **tăng mạnh**, vượt xa số lượng xe PHEV đang được sử dụng.
- Giai đoạn 2020- 2023, số lượng xe BEV **giảm mạnh** kèm theo số lượng xe PHEV tăng đáng kể.
- Năm 2020 đã có **sự thay đổi** từ việc sản xuất xe BEV sang sản xuất PHEV.

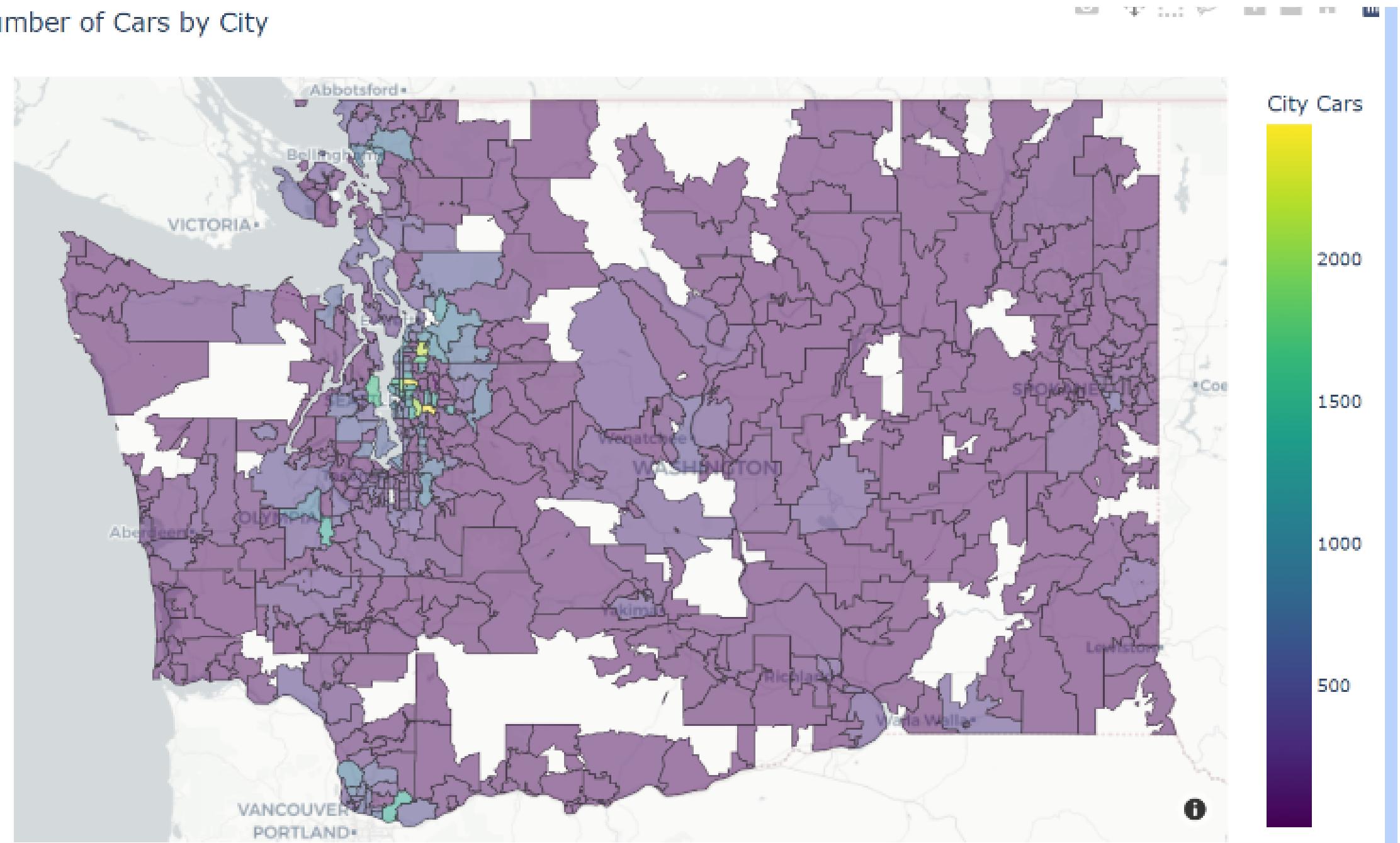
Phân tích theo Model Year và xu hướng sử dụng



- Model Year 2023 **được ưa chuộng nhất** với số lượng trên 40000 chiếc.
- Các Model 2024 **còn khá ít** vì dữ liệu được cập nhật liên tục lần gần nhất là vào ngày 30/9/2023 nên số lượng còn khá ít.
- **Dự đoán trong tương lai** dựa theo biểu đồ Trend ta có thể thấy số lượng xe có Model Year 2024 **sẽ giảm** so với 2023

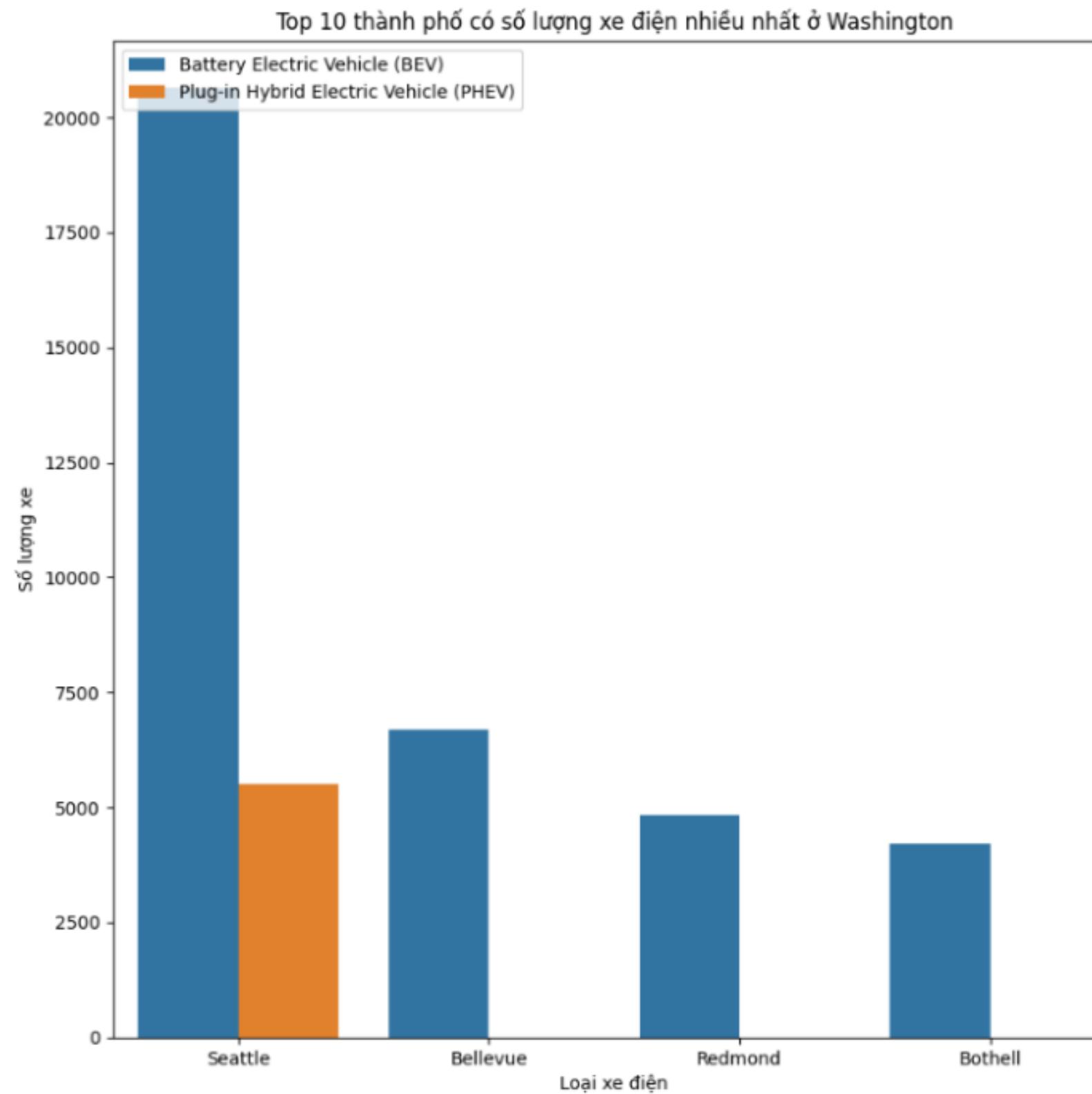
Phân bố xe điện dựa vào thành phố

Number of Cars by City



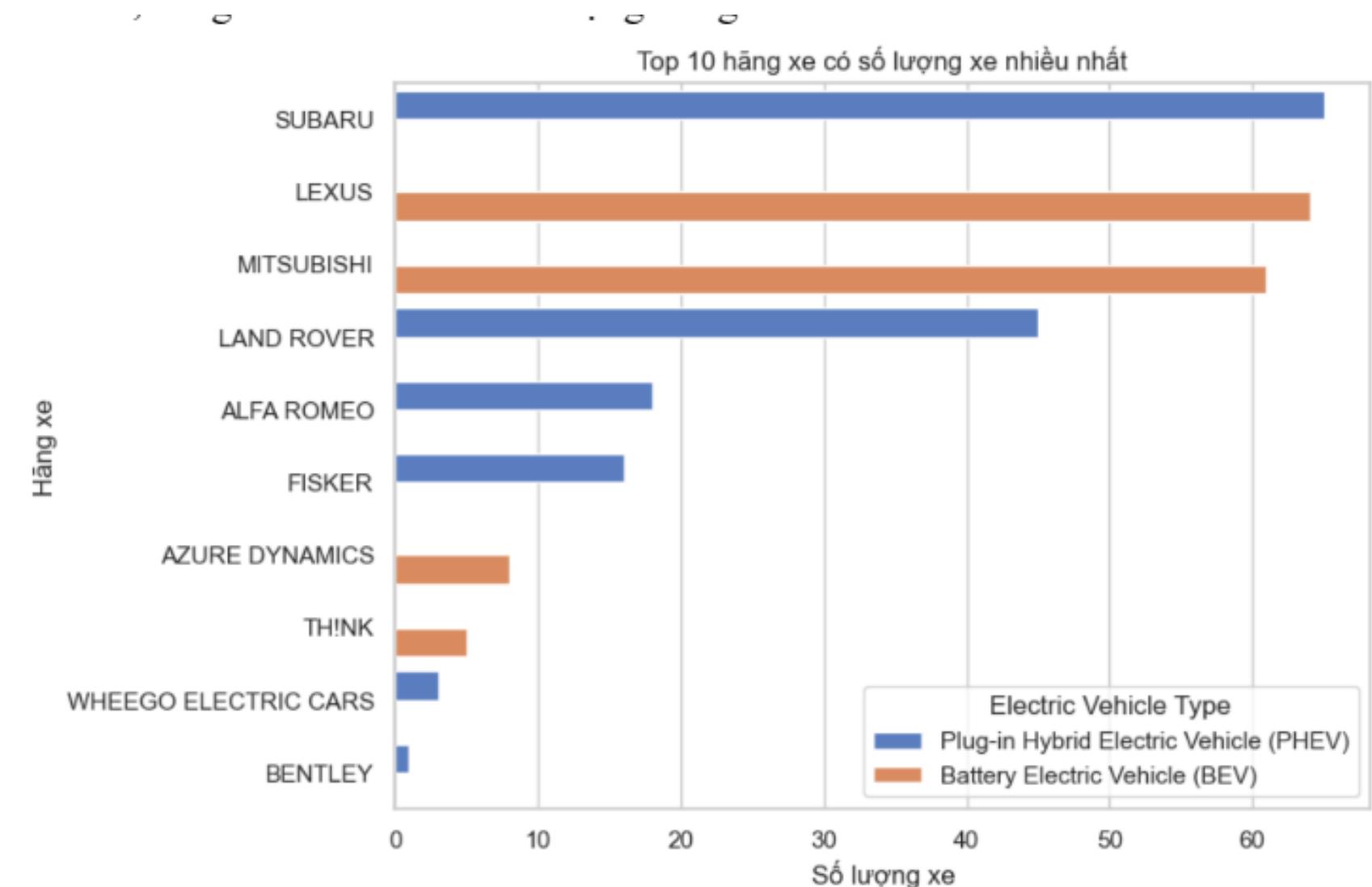
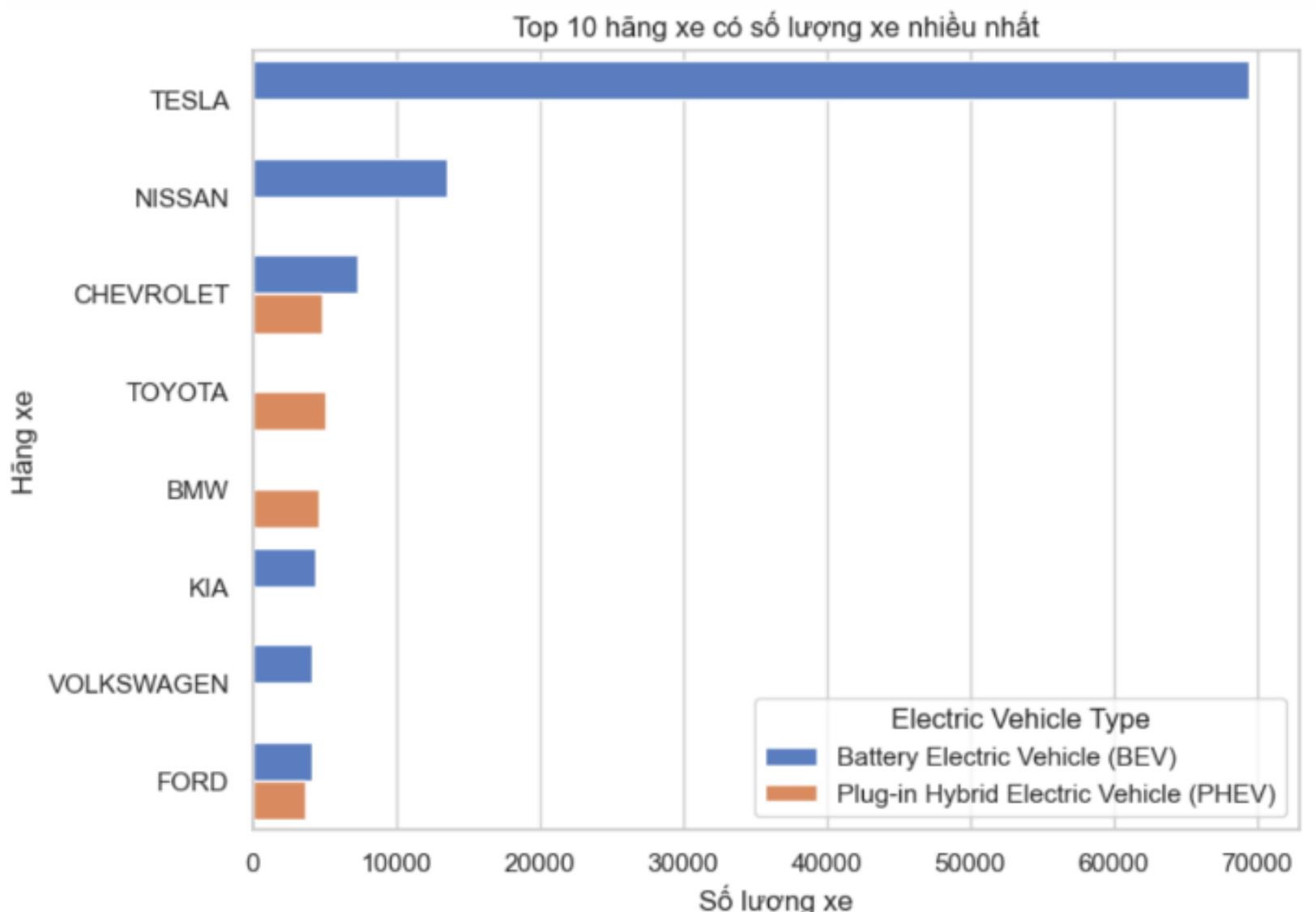
- Lượng xe phần lớn tập trung ở các thành phố trung tâm như **Bothell, Seattle,....** Đây là những thành phố ở vị trí trung tâm nằm gần eo biển Puget Sound thuận lợi về mặt vị trí địa lý giúp phát triển kinh tế
- Ngoài ra ở tiểu bang Washington xe điện **vẫn chưa được xuất hiện** ở toàn bộ thành phố

Các thành phố có xe điện nhiều nhất



Seattle vượt trội hơn các thành phố còn lại về cả Battery Electric Vehicle(BEV) và cả Plug-in hybrid Electric Vehicle(PHEV)

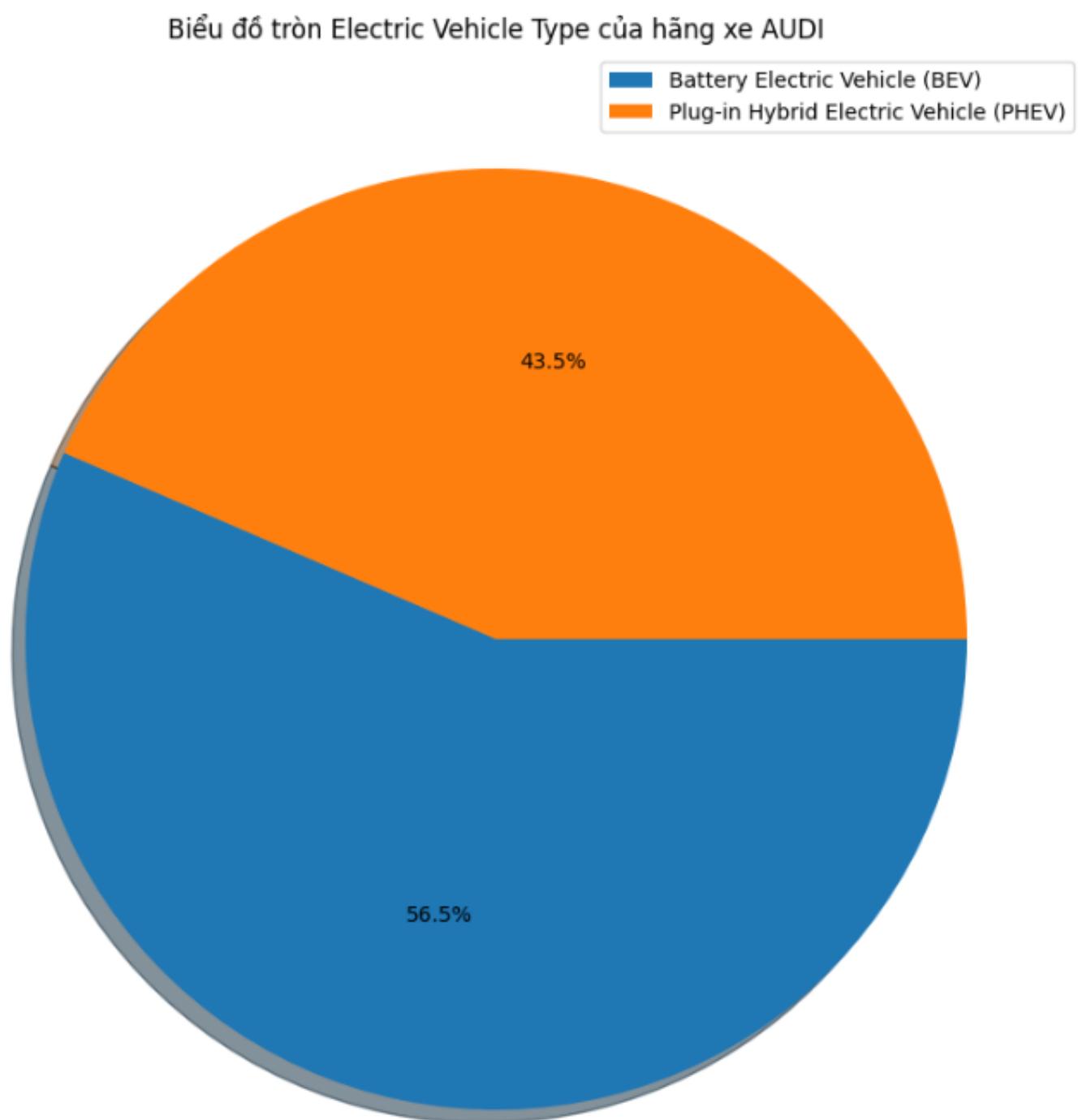
Phân tích 10 hãng xe có nhiều nhất và ít nhất ở Washington



TESLA dẫn đầu về số lượng, vượt trội xe ở bang Washington.

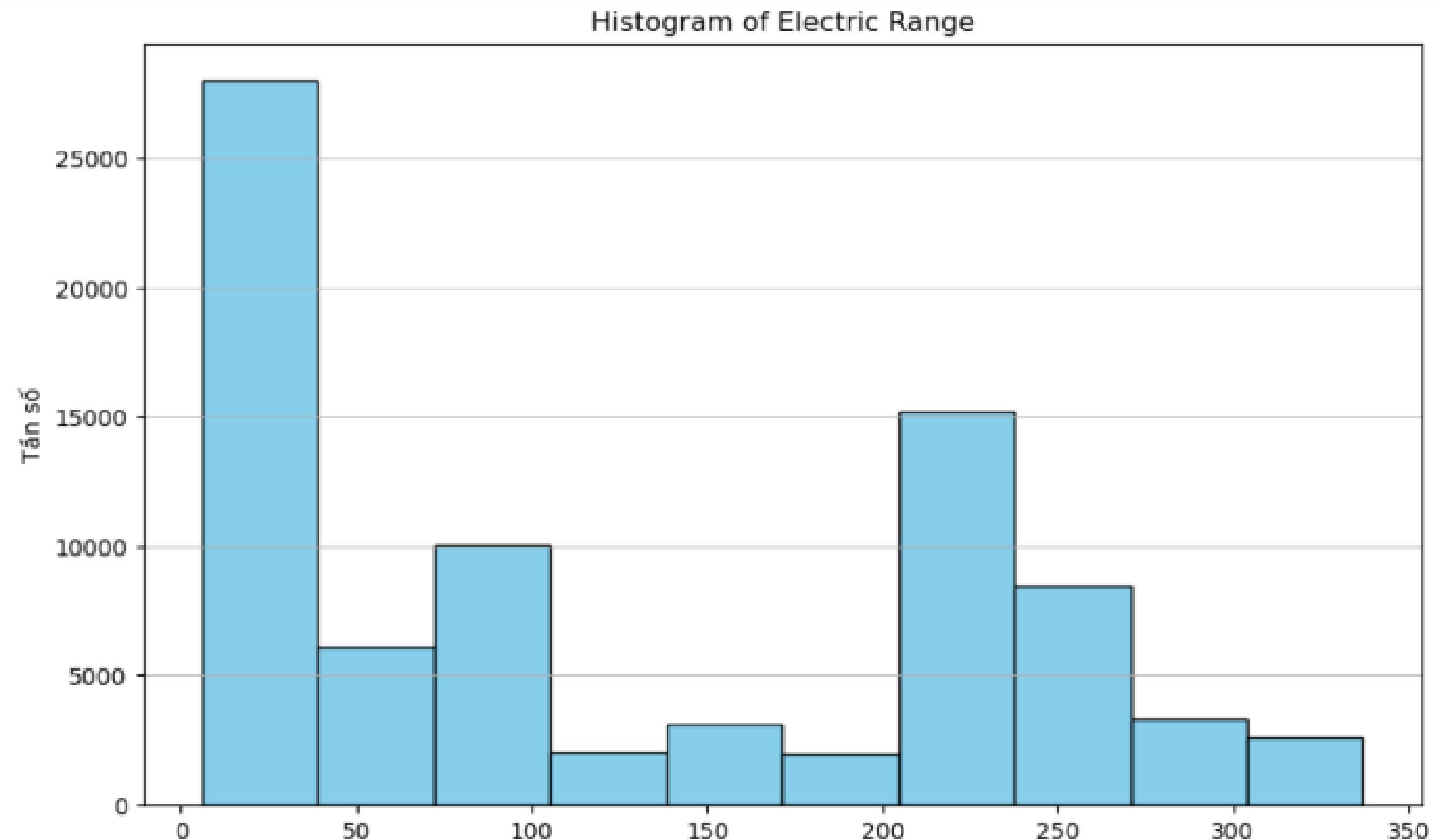
Các hãng xe **không có đa dạng** về các loại xe điện

Tỉ lệ phần trăm các loại xe điện của Audi



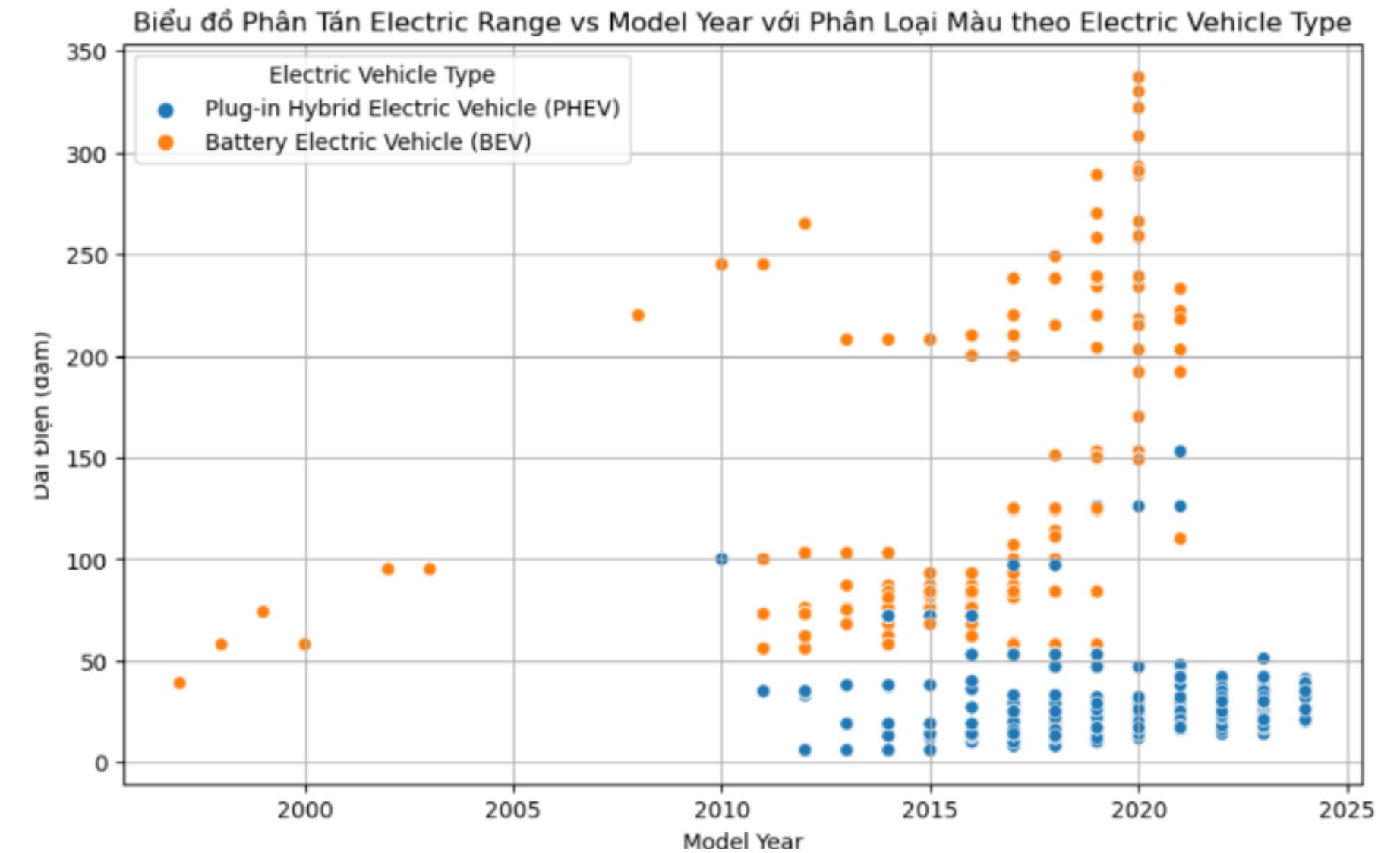
- Số lượng xe điện của Audi ở bang Washington không có sự chênh lệch đáng kể
- Xe loại **BEV** của Audi chiếm 56,5% trên tổng số lượng xe điện ở bang Washington
- **Vì sao Battery Electric Vehicle (BEV) lại có sự vượt trội hơn so với Plug-in Hybird Electric Vehicle (PHEV)?**

Lý do khiến BEV được ưa chuộng hơn PHEV ở Washington



- Phần lớn xe chưa đầy điện đi được **dưới 50 dặm**
- Phân bố Electric Range từ 100 đến 200 là thấp nhất

Scatter Diagram để phân tích kỹ hơn



Electric Range của BEV vượt trội hơn PHEV là điều mà ta có thể dễ dàng nhận thấy được

- Đa phần **Electric Range** của **PHEV** đều **dưới 50 dặm**, chiếm số lượng lớn
- Vẫn có những Model xe thuộc loại **PHEV** có **Electric Range vượt trội**
- Nhìn chung **BEV** có **Electric Range cao hơn** so với **PHEV** vì **đặc điểm của xe**.
- PHEV không có khoảng trống** để chứa pin lớn
- BEV là xe thuần điện**, diện tích chứa lớn

Phân tích các Model có Electric Range cao nhất

Model Year	Electric Range
0	265
1	208
2	208
3	208
4	210
5	210
6	249
7	270
8	337

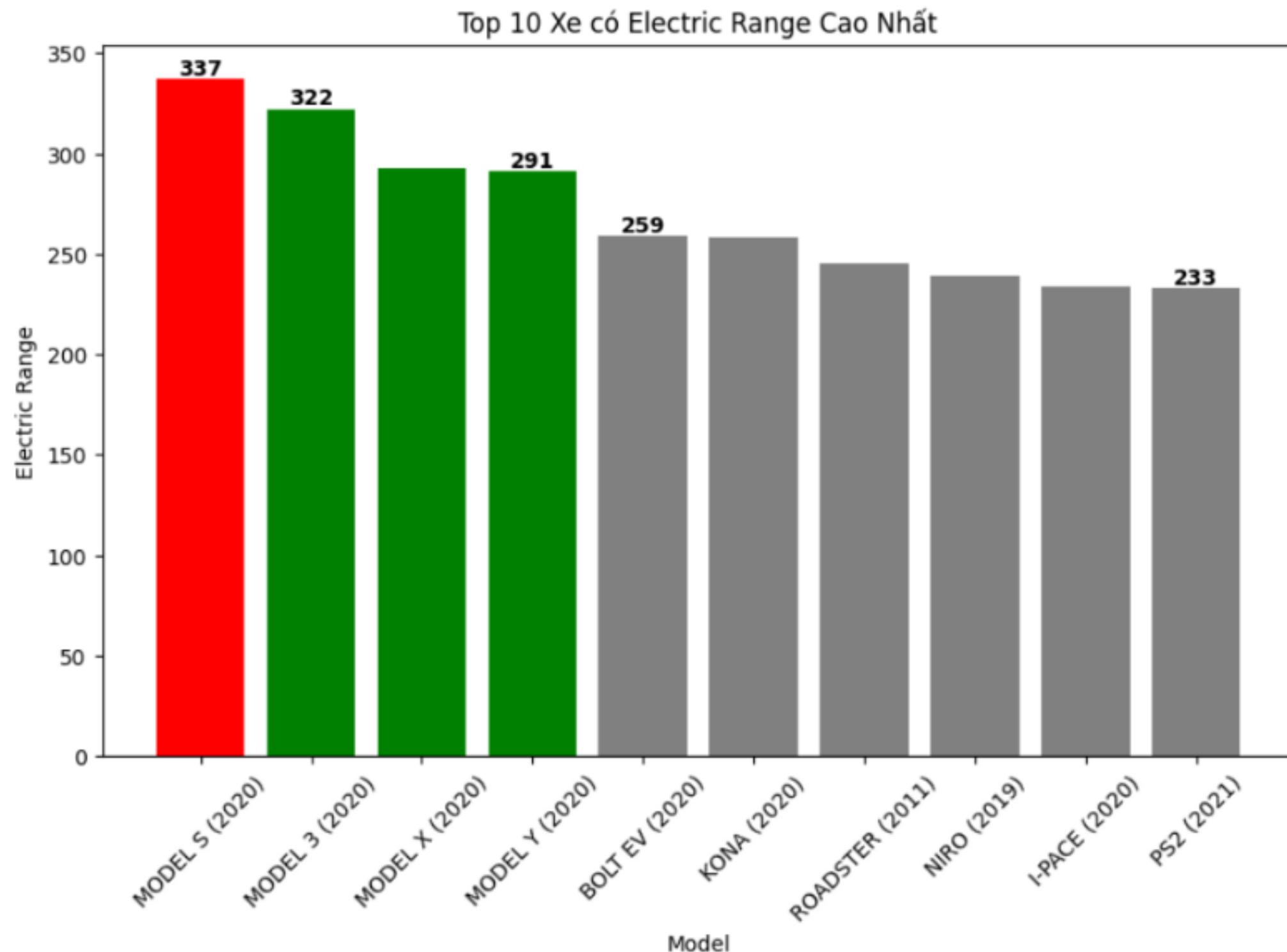
- Loại những Model có **Electric Range = 0**
- Phân tích Model S và Electric Range** tương ứng, quan sát được **khác Model Year** thì **khác Electric Range**

Phân tích các Model có Electric Range cao nhất

Top 10 mẫu xe có
Electric Range cao nhất

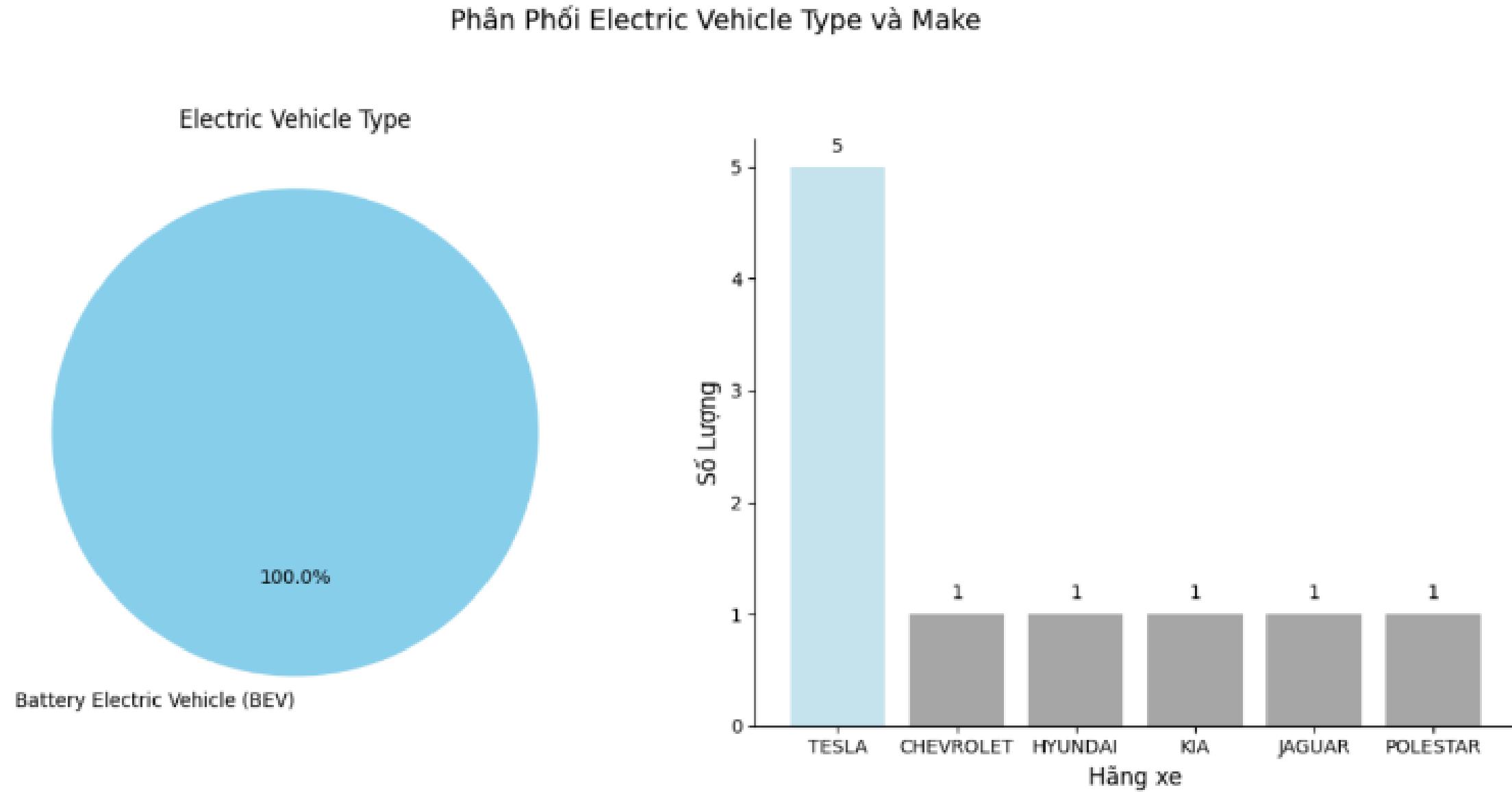
	Model	Electric Range	Model Year
99547	MODEL S	337	2020
88341	MODEL 3	322	2020
71378	MODEL X	293	2020
147538	MODEL Y	291	2020
14165	BOLT EV	259	2020
94660	KONA	258	2020
57014	ROADSTER	245	2010
106448	NIRO	239	2020
45061	I-PACE	234	2019
105795	PS2	233	2021

Phân tích các Model có Electric Range cao nhất



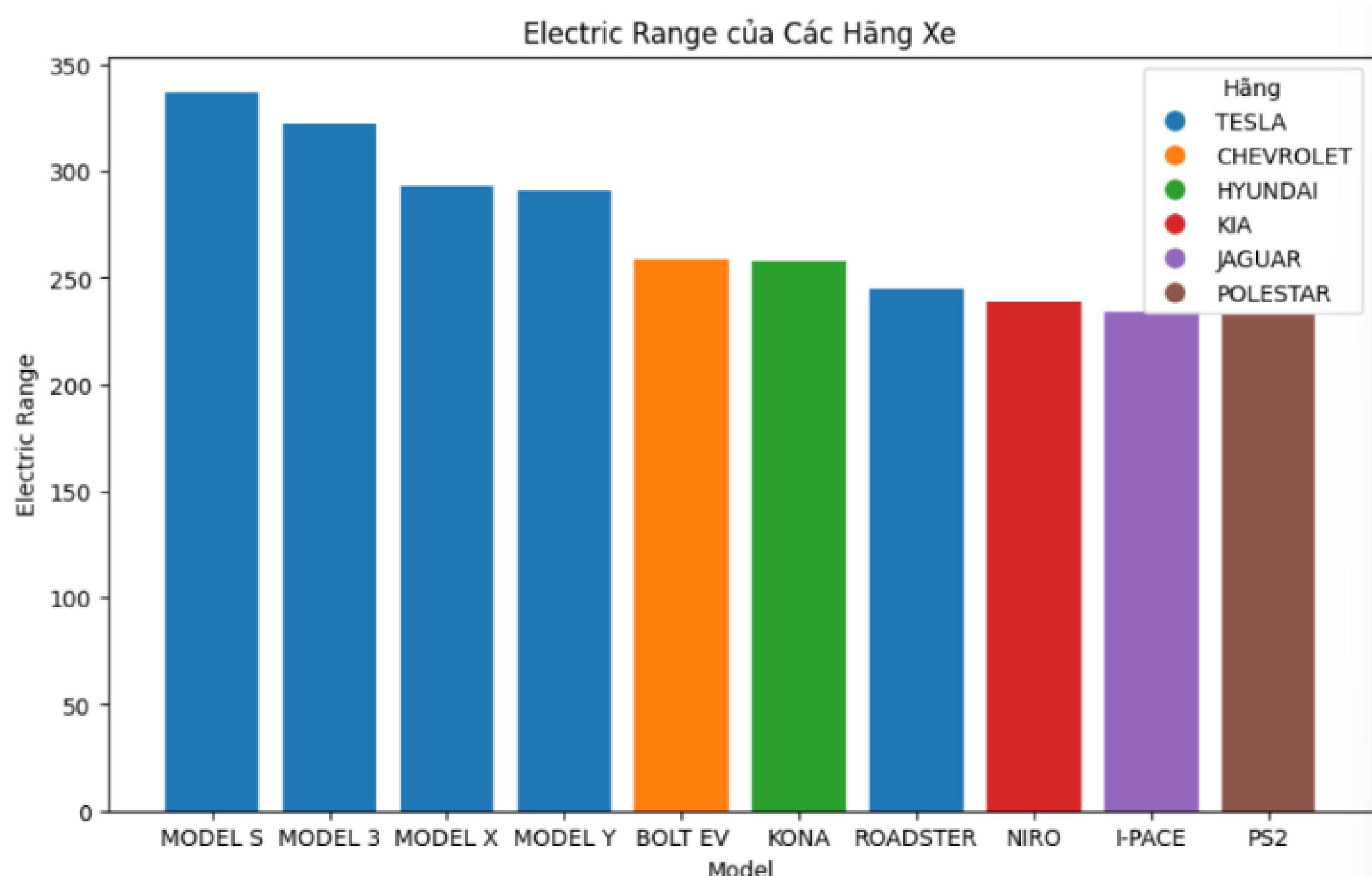
- **Model S(2020) có Electric Range cao nhất** và **sự chênh lệch nhỏ** với **Model 3(2020)**.
- Các Models còn lại khoảng từ **230 đến 300**.

Phân tích các Model có Electric Range cao nhất



- Các mẫu có loại **BEV** **vượt trội hơn (100% top 10 có cùng Electric Vehicle Type)**
- Hãng xe **làm tốt nhất** là **TESLA** (**5/10 xe có Electric Range cao nhất**)

Phân tích các Model có Electric Range cao nhất



- **TESLA** đang làm rất tốt việc phát triển các xe có khả năng **đi xa nhất trong 1 lần sạc**
- Electric Range tốt nhất(Top 4 mẫu xe **đi xa nhất** thuộc **TESLA**)

Phương Pháp Giảm Chiều Dữ Liệu

và Machine Learning

Bài toán đặt ra

Dự đoán Loại Xe Điện (BEV hoặc PHEV)

Giảm Chiều Dữ Liệu và Machine Learning

- Kiểm định các biến ảnh hưởng đến biến target trong mô hình
- Encoding dữ liệu định danh
- Giảm chiều dữ liệu (PCA)
- Mô hình máy học (Machine Learning)



KIỂM ĐỊNH GIỮA HAI BIẾN ĐỊNH TÍNH

- Trong trường hợp này, sử dụng **alpha = 0.001** nhằm giảm xác suất sai lầm loại I.
- Thực hiện kiểm định Chi-square:

```
for i in ['Vehicle Location', 'State', 'County', 'City', 'Postal Code', 'Model Year', 'Make', 'Model'
          , 'Clean Alternative Fuel Vehicle (CAFV) Eligibility', 'Electric Utility', 'DOL Vehicle ID', 'Binary Base MSRP', 'Longitude', 'Latitude']:
    Chi_square(0.001, data, i, 'Electric Vehicle Type')
```

- Thu được kết quả:

Có bằng chứng đủ để bác bỏ H_0 , có sự tương quan giữa hai biến Vehicle Location và Electric Vehicle Type

Không đủ bằng chứng để bác bỏ H_0 , không có sự tương quan giữa hai biến State và Electric Vehicle Type

Có bằng chứng đủ để bác bỏ H_0 , có sự tương quan giữa hai biến County và Electric Vehicle Type

Có bằng chứng đủ để bác bỏ H_0 , có sự tương quan giữa hai biến City và Electric Vehicle Type

Có bằng chứng đủ để bác bỏ H_0 , có sự tương quan giữa hai biến Postal Code và Electric Vehicle Type

Có bằng chứng đủ để bác bỏ H_0 , có sự tương quan giữa hai biến Model Year và Electric Vehicle Type

Có bằng chứng đủ để bác bỏ H_0 , có sự tương quan giữa hai biến Make và Electric Vehicle Type

Có bằng chứng đủ để bác bỏ H_0 , có sự tương quan giữa hai biến Model và Electric Vehicle Type

Có bằng chứng đủ để bác bỏ H_0 , có sự tương quan giữa hai biến Clean Alternative Fuel Vehicle (CAFV) Eligibility và Electric Vehicle Type

Có bằng chứng đủ để bác bỏ H_0 , có sự tương quan giữa hai biến Electric Utility và Electric Vehicle Type

Không đủ bằng chứng để bác bỏ H_0 , không có sự tương quan giữa hai biến DOL Vehicle ID và Electric Vehicle Type

Có bằng chứng đủ để bác bỏ H_0 , có sự tương quan giữa hai biến Binary Base MSRP và Electric Vehicle Type

Có bằng chứng đủ để bác bỏ H_0 , có sự tương quan giữa hai biến Longitude và Electric Vehicle Type

Có bằng chứng đủ để bác bỏ H_0 , có sự tương quan giữa hai biến Latitude và Electric Vehicle Type

Hệ số Cramér's V

Hệ số Cramér's V cho phép xác định mức độ tương quan giữa hai biến định tính và thể hiện nó dưới dạng một giá trị thập phân nằm trong khoảng từ 0 đến 1.

$$\tilde{V} = \sqrt{\frac{\tilde{\varphi}^2}{\min(\tilde{k} - 1, \tilde{r} - 1)}}$$

where

$$\tilde{\varphi}^2 = \max \left(0, \varphi^2 - \frac{(k - 1)(r - 1)}{n - 1} \right)$$

and

$$\tilde{k} = k - \frac{(k - 1)^2}{n - 1}$$

$$\tilde{r} = r - \frac{(r - 1)^2}{n - 1}$$

Phương thức tính độ tương quan của 2 biến định tính:

```
import pandas as pd
from scipy.stats import chi2_contingency
import numpy as np

def cramers_v(contingency_table):
    chi2, _, _, _ = chi2_contingency(contingency_table)
    n = contingency_table.sum()
    phi2 = chi2 / n
    r, k = contingency_table.shape
    phi2corr = max(0, phi2 - ((k-1)*(r-1))/(n-1))
    rcorr = r - ((r-1)**2)/(n-1)
    kcorr = k - ((k-1)**2)/(n-1)
    return np.sqrt(phi2corr / min((kcorr-1), (rcorr-1)))
```

- Thực hiện trên các biến còn lại qua câu lệnh:

```
for i in ['Vehicle Location', 'County', 'City', 'Postal Code', 'Model Year', 'Make', 'Model', 'Clean Alternative Fuel Vehicle (CAFV) Eligibility',  
         'Electric Utility', 'Binary Base MSRP', 'Longitude', 'Latitude']:  
    print(i)  
    contingency_table = pd.crosstab(data[i], data['Electric Vehicle Type'])  
  
    cramers_v_value = cramers_v(contingency_table.values)  
    print(f"Cramér's V: {cramers_v_value}")
```

- Thu được kết quả như sau:

Vehicle Location
Cramér's V: 0.1962189028268443
County
Cramér's V: 0.10936189036453406
City
Cramér's V: 0.17390502071537087
Postal Code
Cramér's V: 0.19621890282684432
Model Year
Cramér's V: 0.26264883713193615
Make
Cramér's V: 0.773704149655979

Model
Cramér's V: 0.9782002403467048
Clean Alternative Fuel Vehicle (CAFV) Eligibility
Cramér's V: 0.7394411692082606
Electric Utility
Cramér's V: 0.10254779689303864
Binary Base MSRP
Cramér's V: 0.04621982579266567
Longitude
Cramér's V: 0.19620266996161564
Latitude
Cramér's V: 0.19621890282684432

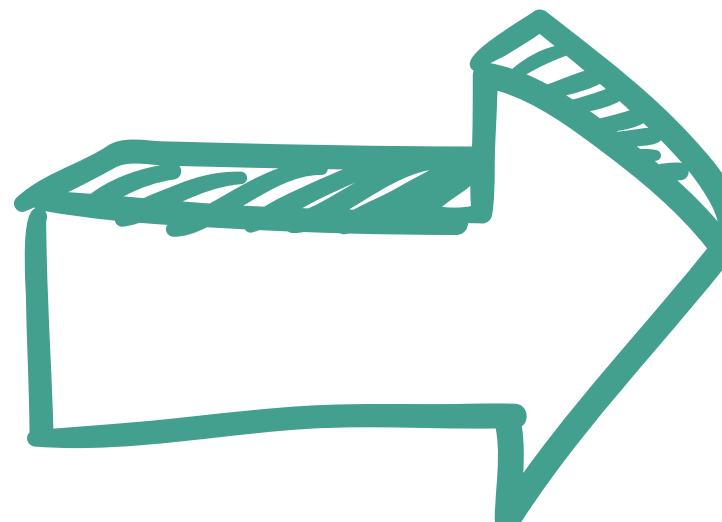
lấy các biến có độ tương quan lớn hơn 0.25

BIẾN ĐỊNH LƯỢNG

- Nhận thấy rằng biến ‘**BASE MSRP**’ chứa nhiều giá trị 0

```
print(len(data[data['Base MSRP'] == 0])/len(data['Base MSRP'])*100)  
✓ 0.0s  
97.77179826304507
```

- Giá trị 0 chiếm **hơn 97%**



Loại biến “**BASE MSRP**” ra khỏi
mô hình máy học

• Sau khi đã thực hiện kiểm định, giữ lại các biến trong bộ dữ liệu sau đây:

	Make	Model	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Model Year	Electric Vehicle Type
0	BMW	X5	Clean Alternative Fuel Vehicle Eligible	2023	Plug-in Hybrid Electric Vehicle (PHEV)
2	TESLA	MODEL S	Clean Alternative Fuel Vehicle Eligible	2016	Battery Electric Vehicle (BEV)
3	NISSAN	LEAF	Clean Alternative Fuel Vehicle Eligible	2018	Battery Electric Vehicle (BEV)
4	TESLA	MODEL 3	Clean Alternative Fuel Vehicle Eligible	2018	Battery Electric Vehicle (BEV)
5	CHEVROLET	BOLT EV	Clean Alternative Fuel Vehicle Eligible	2017	Battery Electric Vehicle (BEV)
...
153825	NISSAN	LEAF	Clean Alternative Fuel Vehicle Eligible	2013	Battery Electric Vehicle (BEV)
153826	TESLA	MODEL S	Clean Alternative Fuel Vehicle Eligible	2017	Battery Electric Vehicle (BEV)
153827	CHEVROLET	BOLT EV	Eligibility unknown as battery range has not b...	2023	Battery Electric Vehicle (BEV)
153828	FORD	MUSTANG MACH-E	Eligibility unknown as battery range has not b...	2021	Battery Electric Vehicle (BEV)
153829	NISSAN	LEAF	Clean Alternative Fuel Vehicle Eligible	2013	Battery Electric Vehicle (BEV)

153487 rows × 5 columns

ENCODING DỮ LIỆU ĐỊNH DANH

- Do phương pháp giảm chiều PCA không hỗ trợ cho dữ liệu định danh.
- Dữ liệu định danh của mô hình được mã hóa các chiều dữ liệu này theo 2 phương pháp:



Ordinal Encoding



One Hot Encoding



ORDINAL ENCODING

- Sử dụng với chiều dữ liệu ‘Model’
- Tạo ra cột mới mang tên ‘Model_encoded’

```
data_encoding_model = df.sort_values(by='Make')
encoder = OrdinalEncoder()
data_encoding_model['Model_encoded'] = encoder.fit_transform(data_encoding_model[['Model']])
data_encoding_model
```

• Thu được kết quả:

	Model	Model_encoded		Model	Model_encoded		Model	Model_encoded		Model	Model_encoded
0	330E	0.0	20	C-MAX	20.0	96	RAV4	96.0	112	SPORTAGE	112.0
1	500	1.0	21	C40	21.0	97	RAV4 PRIME	97.0	113	TAYCAN	113.0
2	530E	2.0	22	CAYENNE	22.0	98	ROADSTER	98.0	114	TONALE	114.0
3	740E	3.0	23	CITY	23.0	99	RS E-TRON GT	99.0	115	TRANSIT	115.0
4	745E	4.0	24	CLARITY	24.0	100	RZ 450E	100.0	116	TRANSIT CONNECT ELECTRIC	116.0
5	745LE	5.0	25	CORSAIR	25.0	101	S-10 PICKUP	101.0	117	TUCSON	117.0
6	918	6.0	26	COUNTRYMAN	26.0	102	S-CLASS	102.0	118	V60	118.0
7	A3	7.0	27	CROSSTREK	27.0	103	S60	103.0	119	VOLT	119.0
8	A7	8.0	28	CT6	28.0	104	S90	104.0	120	WHEEGO	120.0
9	A8 E	9.0	29	CX-90	29.0	105	SANTA FE	105.0	121	WRANGLER	121.0
10	ACCORD	10.0	30	E-GOLF	30.0	106	SOLTERRA	106.0	122	X3	122.0
11	AIR	11.0	31	E-TRON	31.0	107	SONATA	107.0	123	X5	123.0
12	ARIYA	12.0	32	E-TRON GT	32.0	108	SORENTO	108.0	124	XC40	124.0
13	AVIATOR	13.0	33	E-TRON SPORTBACK	33.0	109	SOUL	109.0	125	XC60	125.0
14	B-CLASS	14.0	34	EDV	34.0	110	SOUL EV	110.0	126	XC90	126.0
15	BENTAYGA	15.0	35	ELR	35.0	111	SPARK	111.0			
16	BOLTEUV	16.0	36	EQ FORTWO	36.0						
17	BOLTEV	17.0	37	EQB-CLASS	37.0						
18	BZ4X	18.0	38	EQE-CLASS SEDAN	38.0						

- **Tương tự với biến ‘Clean Alternative Fuel Vehicle (CAFV) Eligibility’.**
- **Thu được kết quả như sau:**

Clean Alternative Fuel Vehicle (CAFV) Eligibility		CAFV_encoded
0	Clean Alternative Fuel Vehicle Eligible	0.0
1	Eligibility unknown as battery range has not b...	1.0
2	Not eligible due to low battery range	2.0

ONE HOT ENCODING

- Sử dụng phương pháp OneHot Encoding bằng thư viện pandas với biến Make.

```
make_one_hot_encoded_make = pd.get_dummies(data_encoding_model['Make'], prefix='Make')
data_encoded = pd.concat([data_encoding_model, make_one_hot_encoded_make], axis=1)
data_encoded
```

- Kết quả thu được:

	Make	Model	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Model Year	Electric Vehicle Type	Model_encoded	Make_ALFA ROMEO	Make_AUDI	Make_AZURE DYNAMICS	Make_BENTLEY	...	Make_PORSCHE	Make_RIV
8236	ALFA ROMEO	TONALE	Clean Alternative Fuel Vehicle Eligible	2024	Plug-In Hybrid Electric Vehicle (PHEV)	114.0	1	0	0	0	...	0	
4076	ALFA ROMEO	TONALE	Clean Alternative Fuel Vehicle Eligible	2024	Plug-in Hybrid Electric Vehicle (PHEV)	114.0	1	0	0	0	...	0	
6058	ALFA ROMEO	TONALE	Clean Alternative Fuel Vehicle Eligible	2024	Plug-in Hybrid Electric Vehicle (PHEV)	114.0	1	0	0	0	...	0	
2742	ALFA ROMEO	TONALE	Clean Alternative Fuel Vehicle Eligible	2024	Plug-in Hybrid Electric Vehicle (PHEV)	114.0	1	0	0	0	...	0	

DỮ LIỆU SAU KHI ĐÃ ENCODING

- Xóa những cột đã được mã hóa như 'Make', 'Model', 'Clean Alternative Fuel Vehicle (CAFV) Eligibility'.**

#	Column	Non-Null Count	Dtype	Non-Null Count	Dtype	Non-Null Count	Dtype
0	Model Year	5 non-null	int64	22	Make_LINCOLN	5 non-null	uint8
1	Electric Vehicle Type	5 non-null	object	23	Make_LUCID	5 non-null	uint8
2	Model_encoded	5 non-null	float64	24	Make_MAZDA	5 non-null	uint8
3	Make_ALFA ROMEO	5 non-null	uint8	25	Make_MERCEDES-BENZ	5 non-null	uint8
4	Make_AUDI	5 non-null	uint8	26	Make_MINI	5 non-null	uint8
5	Make_AZURE DYNAMICS	5 non-null	uint8	27	Make_MITSUBISHI	5 non-null	uint8
6	Make_BENTLEY	5 non-null	uint8	28	Make_NISSAN	5 non-null	uint8
7	Make_BMW	5 non-null	uint8	29	Make_POLESTAR	5 non-null	uint8
8	Make_CADILLAC	5 non-null	uint8	30	Make_PORSCHE	5 non-null	uint8
9	Make_CHEVROLET	5 non-null	uint8	31	Make_RIVIAN	5 non-null	uint8
10	Make_CHRYSLER	5 non-null	uint8	32	Make_SMART	5 non-null	uint8
11	Make_FIAT	5 non-null	uint8	33	Make_SUBARU	5 non-null	uint8
12	Make_FISKER	5 non-null	uint8	34	Make_TESLA	5 non-null	uint8
13	Make_FORD	5 non-null	uint8	35	Make_TH!NK	5 non-null	uint8
14	Make_GENESIS	5 non-null	uint8	36	Make_TOYOTA	5 non-null	uint8
15	Make_HONDA	5 non-null	uint8	37	Make_VOLKSWAGEN	5 non-null	uint8
16	Make_HYUNDAI	5 non-null	uint8	38	Make_VOLVO	5 non-null	uint8
17	Make_JAGUAR	5 non-null	uint8	39	Make_WHEEGO ELECTRIC CARS	5 non-null	uint8
18	Make_JEEP	5 non-null	uint8	40	CAFV_encoded	5 non-null	float64
19	Make_KIA	5 non-null	uint8				
20	Make_LAND ROVER	5 non-null	uint8				
21	Make_LEXUS	5 non-null	uint8				

PCA

- **Đầu tiên xác định biến target trong 41 cột dữ liệu sau khi đã Encoding.**
- **Tách biến target khỏi danh sách các biến features.**

```
target = 'Electric Vehicle Type'
print('* Biến phân lớp:', target)
features = [col for col in df.columns if col != target]
nb_features = len(features)
print('* Số lượng features = %2d' % nb_features)
print(' Các features:', ', '.join(features))
```

```
* Biến phân lớp: Electric Vehicle Type
* Số lượng features = 40
 Các features: Model Year, Model_encoded, Make_ALFA ROMEO, Make_AUDI, Make_AZURE DYNAMICS, Make_BENTLEY, Make_BMW, Make_CADILLAC, Make_CHEVROLET, Make_CHRYSLER, Make_FIAT, Make_FISKER, Make_FORD, Make_GENESIS, Make_HONDA, Make_HYUNDAI, Make_JAGUAR, Make_JEEP, Make_KIA, Make_LAND ROVER, Make_LEXUS, Make_LINEAR, Make_LUCID, Make_MAZDA, Make_MERCEDES-BENZ, Make_MINI, Make_MITSUBISHI, Make_NISSAN, Make_POLESTAR, Make_PORSCHE, Make_RIVIAN, Make_SMART, Make_SUBARU, Make_TESLA, Make_THINK, Make_TOYOTA, Make_VOLKSWAGEN, Make_VOLVO, Make_WHEEGO ELECTRIC CARS, CAFV_encoded
```

PCA

- Tiếp theo, tính toán phương sai tích lũy sau khi giảm về số features k tương ứng.

```
features = df.drop('Electric Vehicle Type', axis=1)
target = 'Electric Vehicle Type'

pca = PCA()
pca.fit(features)

cumulative_variance = np.cumsum(pca.explained_variance_ratio_) * 100
x_values = np.arange(1, len(cumulative_variance) + 1)

plt.figure(figsize=(10, 6))
plt.plot(x_values, cumulative_variance, marker='o')
plt.xlabel('Number of Components (k)')
plt.ylabel('Cumulative Variance Explained (%)')
plt.title('Đồ thị biểu diễn % phương sai tích lũy theo số features (k)')
plt.grid(True)

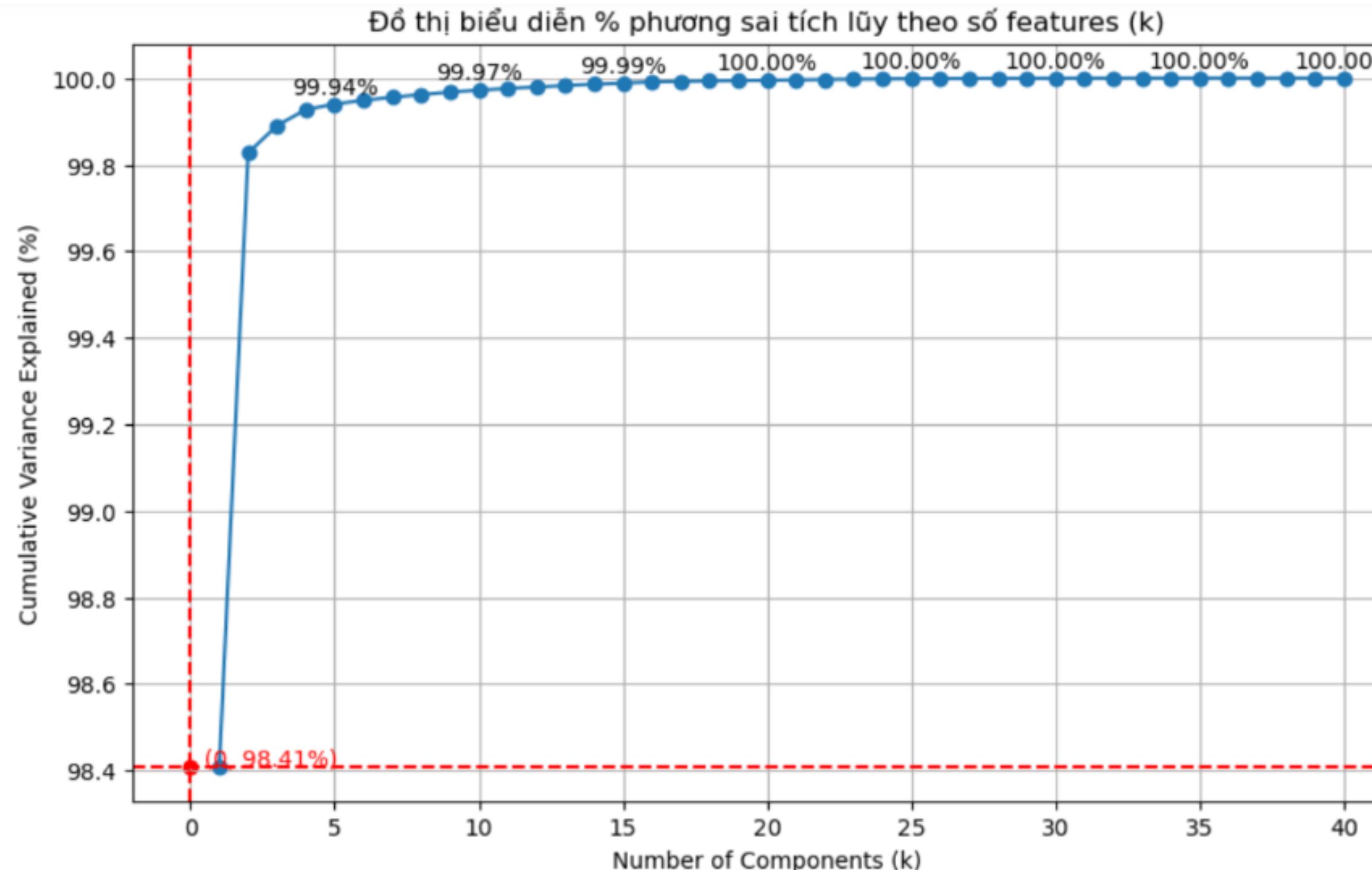
for k in range(5, len(cumulative_variance) + 1, 5):
    plt.text(k, cumulative_variance[k - 1] + 0.025, f'{cumulative_variance[k - 1]:.2f}%', ha='center')

elbow_point = np.argmax(cumulative_variance >= 95)
plt.axvline(x=elbow_point, color='r', linestyle='--')
plt.axhline(y=cumulative_variance[elbow_point], color='r', linestyle='--')
plt.scatter(elbow_point, cumulative_variance[elbow_point], color='red')
plt.text(elbow_point, cumulative_variance[elbow_point], f' ({elbow_point}, {cumulative_variance[elbow_point]:.2f}%)', color='red')

plt.show()
```

PCA

- Đồ thị biểu diễn % phương sai tích lũy theo số features k



PCA

- **Dữ liệu sau khi đã giảm về features(k) = 2**
- **Phương sai tích lũy ~ 99.83%, có 2 chiều PC1 và PC2**

```
pca_2d = PCA(n_components=2)
features_2d = pca_2d.fit_transform(features)

pca_df = pd.DataFrame(data=features_2d, columns=['PC1', 'PC2'])
pca_df[target] = df['Electric Vehicle Type']
print(pca_df)

plt.figure(figsize=(10, 8))
sns.scatterplot(data=pca_df, x='PC1', y='PC2', hue=target, palette='viridis')
plt.title('PCA Scatter Plot with 2 Components')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title=target, bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```

PCA

- Trước khi có biến Target
‘Electric Vehicle Type’

		PC1	PC2
0	-42.595627	-3.549907	
1	-42.595627	-3.549907	
2	-42.595627	-3.549907	
3	-42.595627	-3.549907	
4	-42.595627	-3.549907	
...
153482	-46.597111	-3.528147	
153483	-53.577131	-0.670595	
153484	-48.510409	10.418840	
153485	-48.510409	10.418840	
153486	-48.510409	10.418840	

PCA

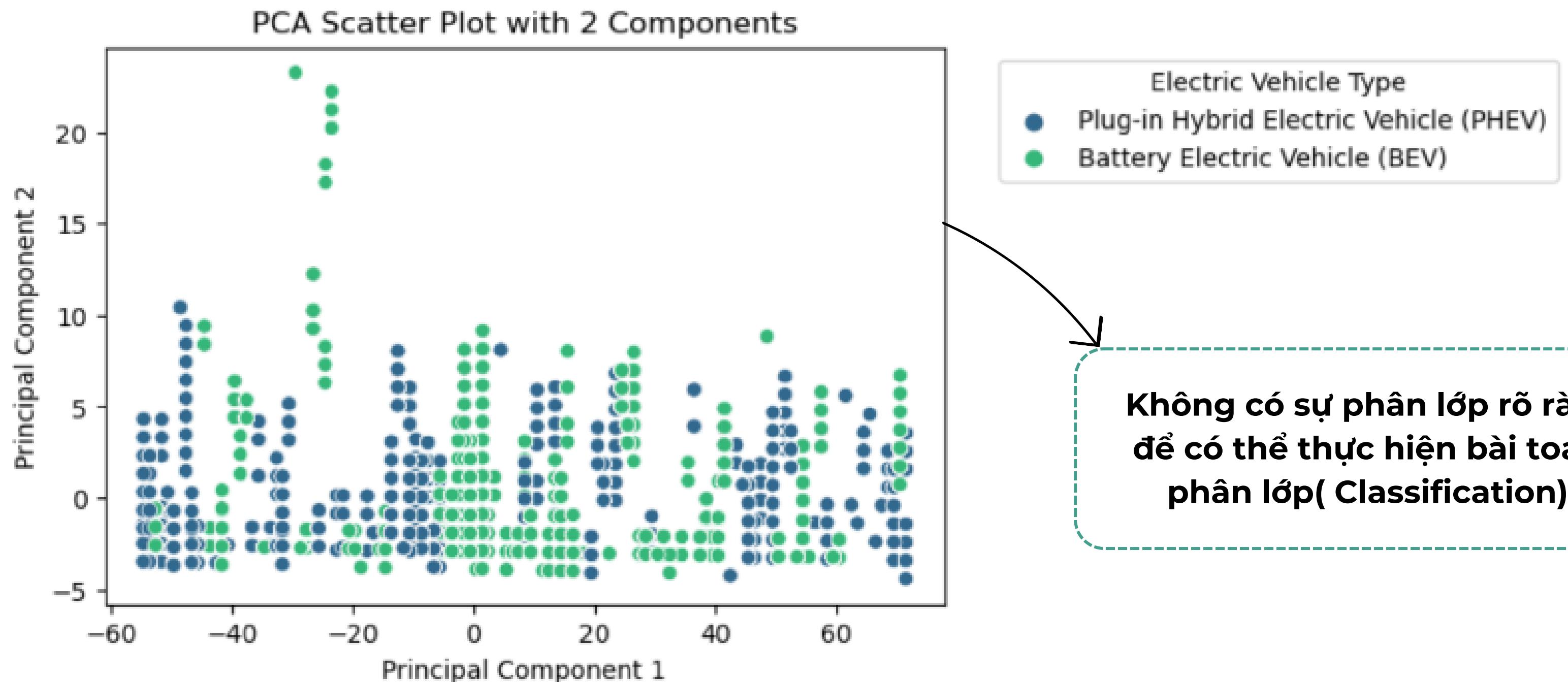
Sau khi có biến Target ‘Electric Vehicle Type’

	PC1	PC2	Electric Vehicle Type
0	-42.595627	-3.549907	Plug-in Hybrid Electric Vehicle (PHEV)
1	-42.595627	-3.549907	Plug-in Hybrid Electric Vehicle (PHEV)
2	-42.595627	-3.549907	Plug-in Hybrid Electric Vehicle (PHEV)
3	-42.595627	-3.549907	Plug-in Hybrid Electric Vehicle (PHEV)
4	-42.595627	-3.549907	Plug-in Hybrid Electric Vehicle (PHEV)
...
153482	-46.597111	-3.528147	Plug-in Hybrid Electric Vehicle (PHEV)
153483	-53.577131	-0.670595	Plug-in Hybrid Electric Vehicle (PHEV)
153484	-48.510409	10.418840	Plug-in Hybrid Electric Vehicle (PHEV)
153485	-48.510409	10.418840	Plug-in Hybrid Electric Vehicle (PHEV)
153486	-48.510409	10.418840	Plug-in Hybrid Electric Vehicle (PHEV)

[153487 rows x 3 columns]

PCA

Sau khi có biến target ‘Electric Vehicle Type’



PCA

- **Tìm giá k tương ứng sau khi thực hiện quá trình giảm chiều để có thể dễ dàng áp dụng bài toán phân lớp.**
- **Chạy kết quả phương sai tích lũy(Variance) để có thể đưa ra giá trị k mong muốn**

```
var = 0.0
for k in range(1, nb_features + 1):
    pca = PCA(k)
    pca.fit(features)

    newVar = pca.explained_variance_ratio_.sum() * 100
    print(' * k = %2d' %k, ': phương sai tích lũy ~ %.2f%%' %newVar,
          '--> tăng ~ %.5f%%' %(newVar - var))
    var = newVar
```

PCA

- **Thu được kết quả:**

* k = 1 : phương sai tích lũy ~ 98.41% --> tăng ~ 98.40927%
 * k = 2 : phương sai tích lũy ~ 99.83% --> tăng ~ 1.41819%
 * k = 3 : phương sai tích lũy ~ 99.89% --> tăng ~ 0.06292%
 * k = 4 : phương sai tích lũy ~ 99.93% --> tăng ~ 0.03702%
 * k = 5 : phương sai tích lũy ~ 99.94% --> tăng ~ 0.01278%
 * k = 6 : phương sai tích lũy ~ 99.95% --> tăng ~ 0.00858%
 * k = 7 : phương sai tích lũy ~ 99.96% --> tăng ~ 0.00696%
 * k = 8 : phương sai tích lũy ~ 99.96% --> tăng ~ 0.00654%
 * k = 9 : phương sai tích lũy ~ 99.97% --> tăng ~ 0.00552%
 * k = 10 : phương sai tích lũy ~ 99.97% --> tăng ~ 0.00455%
 * k = 11 : phương sai tích lũy ~ 99.98% --> tăng ~ 0.00379%
 * k = 12 : phương sai tích lũy ~ 99.98% --> tăng ~ 0.00354%
 * k = 13 : phương sai tích lũy ~ 99.98% --> tăng ~ 0.00328%
 * k = 14 : phương sai tích lũy ~ 99.99% --> tăng ~ 0.00290%
 * k = 15 : phương sai tích lũy ~ 99.99% --> tăng ~ 0.00281%
 * k = 16 : phương sai tích lũy ~ 99.99% --> tăng ~ 0.00250%
 * k = 17 : phương sai tích lũy ~ 99.99% --> tăng ~ 0.00127%
 * k = 18 : phương sai tích lũy ~ 99.99% --> tăng ~ 0.00107%
 * k = 19 : phương sai tích lũy ~ 99.99% --> tăng ~ 0.00095%
 * k = 20 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00084%

* k = 21 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00081%
 * k = 22 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00080%
 * k = 23 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00073%
 * k = 24 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00063%
 * k = 25 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00029%
 * k = 26 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00025%
 * k = 27 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00023%
 * k = 28 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00021%
 * k = 29 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00020%
 * k = 30 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00019%
 * k = 31 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00016%
 * k = 32 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00012%
 * k = 33 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00005%
 * k = 34 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00002%
 * k = 35 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00002%
 * k = 36 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00001%
 * k = 37 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00001%
 * k = 38 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00000%
 * k = 39 : phương sai tích lũy ~ 100.00% --> tăng ~ 0.00000%
 * k = 40 : phương sai tích lũy ~ 100.00% --> tăng ~ -0.00000%

PCA

- **Quyết định chọn mô hình có giá trị phương sai tích lũy lớn hơn 99,9%**

```
var = 0.0
for k in range(1, nb_features + 1):
    pca = PCA(k)
    pca.fit(features)

    newVar = pca.explained_variance_ratio_.sum() * 100
    print(' * k = %2d' %k, ': phương sai tích lũy ~ %.2f%%' %newVar,
          '--> tăng ~ %.5f%%' %(newVar - var))
    var = newVar
```

* Muốn phương sai tích lũy $\geq 99.90\%$ thì $k \geq 4 \rightarrow 99.93\%$

PCA

- Kết quả mô hình sau khi đã giảm chiều và kết hợp với biến Target ‘Electric Vehicle Type’**

		PC1	PC2	PC3	PC4	\
0	-42.595627	-3.549907	-0.728360	-0.861486		
1	-42.595627	-3.549907	-0.728360	-0.861486		
2	-42.595627	-3.549907	-0.728360	-0.861486		
3	-42.595627	-3.549907	-0.728360	-0.861486		
4	-42.595627	-3.549907	-0.728360	-0.861486		
...	
153482	-46.597111	-3.528147	-0.687403	-0.902076		
153483	-53.577131	-0.670595	1.390079	-0.111587		
153484	-48.510409	10.418840	0.272238	-0.007173		
153485	-48.510409	10.418840	0.272238	-0.007173		
153486	-48.510409	10.418840	0.272238	-0.007173		

	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	Plug-in Hybrid Electric Vehicle (PHEV)
4	Plug-in Hybrid Electric Vehicle (PHEV)
...	...
153482	Plug-in Hybrid Electric Vehicle (PHEV)
153483	Plug-in Hybrid Electric Vehicle (PHEV)
153484	Plug-in Hybrid Electric Vehicle (PHEV)
153485	Plug-in Hybrid Electric Vehicle (PHEV)
153486	Plug-in Hybrid Electric Vehicle (PHEV)

[153487 rows x 5 columns]

Machine Learning

- **Bài toán đặt ra: Dự đoán Loại Xe Điện (BEV hoặc PHEV)**
- **Mục Tiêu:** Dự đoán giá trị của biến target "**Electric Vehicle Type**" dựa trên các thuộc tính khác của xe.
- **Phương pháp:** Sử dụng 4 mô hình phân lớp
 - **SVM (Support Vector Machine)**
 - **Decision Tree**
 - **Logistic Regression**
 - **K-NN**

SVM (Support Vector Machine)

- **Đưa ra các chỉ số như Precision, Recall/Sensitivity, F1-Score.**

```
X = pca_df_4.drop('Electric Vehicle Type', axis=1)
y = pca_df_4['Electric Vehicle Type']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, shuffle = True)

svm_model = SVC()

svm_model.fit(X_train, y_train)

y_pred = svm_model.predict(X_test)
print('-----Suport Vector Machine-----')
print(classification_report(y_test, y_pred))
```

SVM

Thu được kết quả như sau:

Support Vector Machine					
	precision	recall	f1-score	support	
Battery Electric Vehicle (BEV)	0.96	0.97	0.97	23984	
Plug-in Hybrid Electric Vehicle (PHEV)	0.89	0.87	0.88	6714	
accuracy			0.95	30698	
macro avg	0.93	0.92	0.92	30698	
weighted avg	0.95	0.95	0.95	30698	

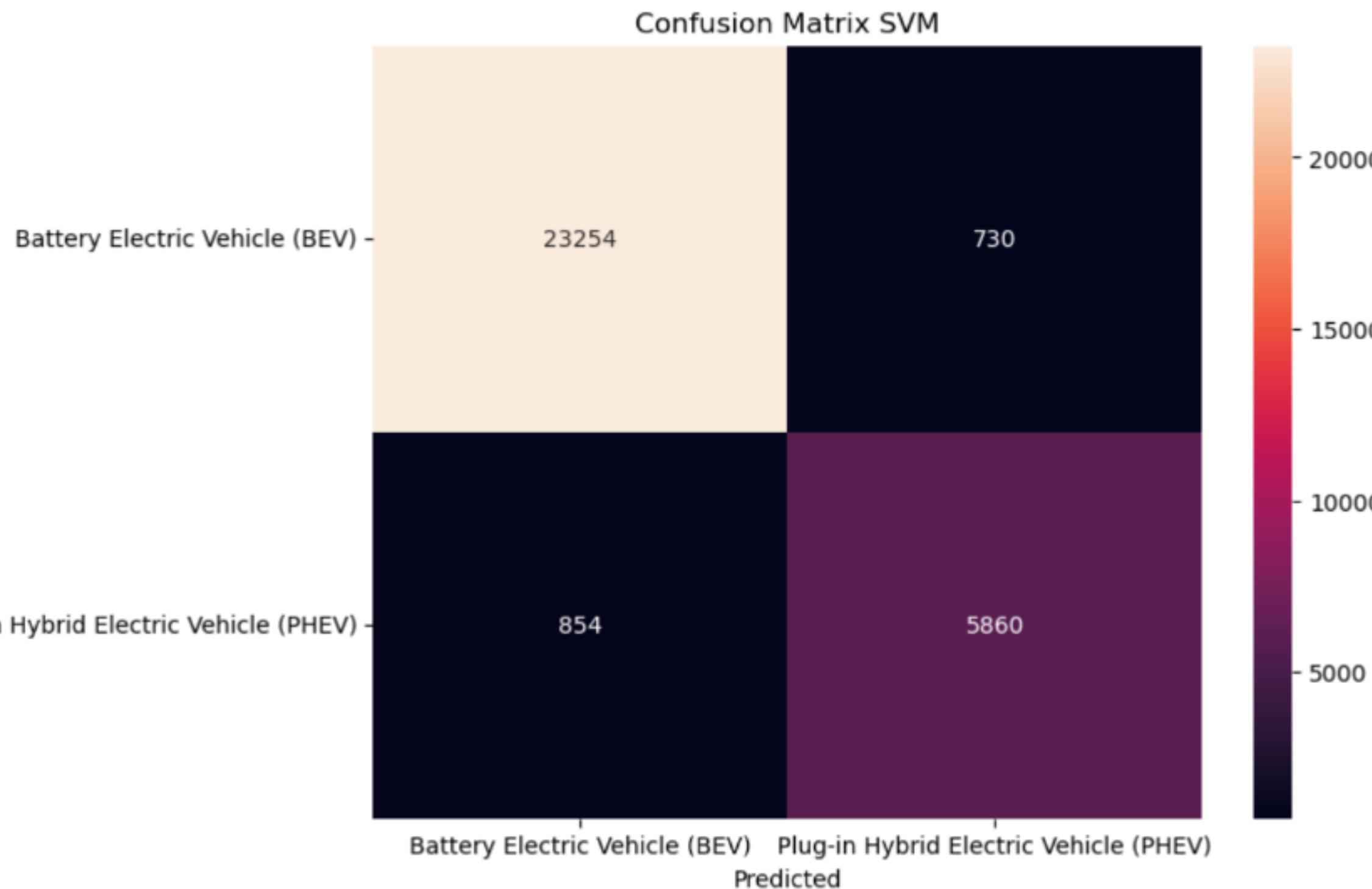


Mô hình có độ chính xác (Accuracy) là 95%

Có độ chính xác là 96% khi dự đoán lớp BEV và 89% khi dự đoán lớp PHEV

SVM

- **Ma trận nhầm lẫn của mô hình SVM**



Có 854 xe được dự đoán nhầm lẫn từ xe PHEV thành loại xe BEV.
Ngược lại có 730 xe loại BEV bị mô hình dự đoán nhầm lẫn thành loại xe PHEV.

Decision Tree

- **Đưa ra các chỉ số như Precision, Recall/Sensitivity, F1-Score.**

```
tree_model = DecisionTreeClassifier()
tree_model.fit(X_train, y_train)
y_pred = tree_model.predict(X_test)
print('-----Decision Tree-----')
print(classification_report(y_test, y_pred))
```

Decision Tree

Thu được kết quả như sau:

-----Decision Tree-----				
	precision	recall	f1-score	support
Battery Electric Vehicle (BEV)	1.00	1.00	1.00	23984
Plug-in Hybrid Electric Vehicle (PHEV)	0.98	1.00	0.99	6714
accuracy			1.00	30698
macro avg	0.99	1.00	0.99	30698
weighted avg	1.00	1.00	1.00	30698

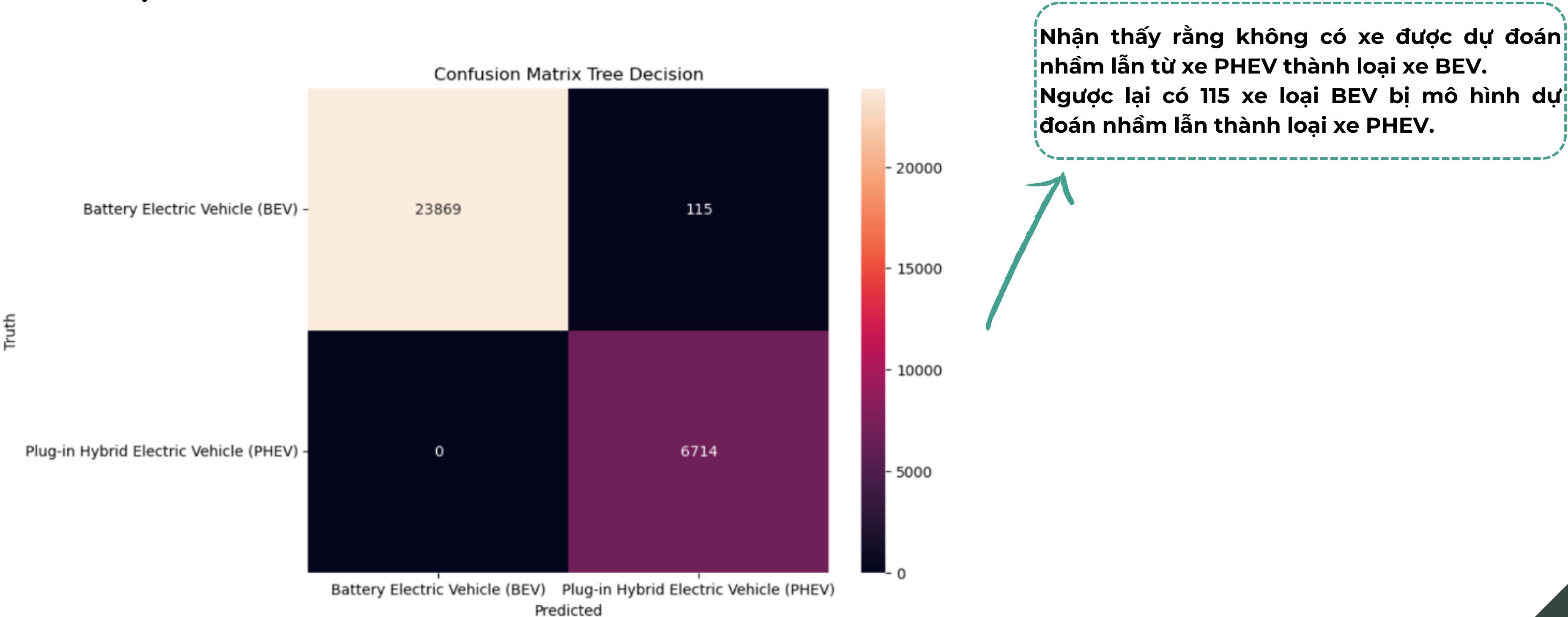


Mô hình có độ chính xác (Accuracy) xấp xỉ 100%

Có độ chính xác xấp xỉ 100% khi dự đoán lớp BEV và 98% khi dự đoán lớp PHEV.

Decision Tree

- Ma trận nhầm lẫn của mô hình Decision Tree



Logistic Regression

```
log_reg_model = LogisticRegression()  
  
log_reg_model.fit(X_train, y_train)  
  
y_pred = log_reg_model.predict(X_test)  
  
print('-----Logistic Regression-----')  
print(classification_report(y_test, y_pred))
```

-----Logistic Regression-----				
	precision	recall	f1-score	support
Battery Electric Vehicle (BEV)	0.89	0.96	0.93	23984
Plug-in Hybrid Electric Vehicle (PHEV)	0.81	0.60	0.69	6714
accuracy			0.88	30698
macro avg	0.85	0.78	0.81	30698
weighted avg	0.88	0.88	0.87	30698

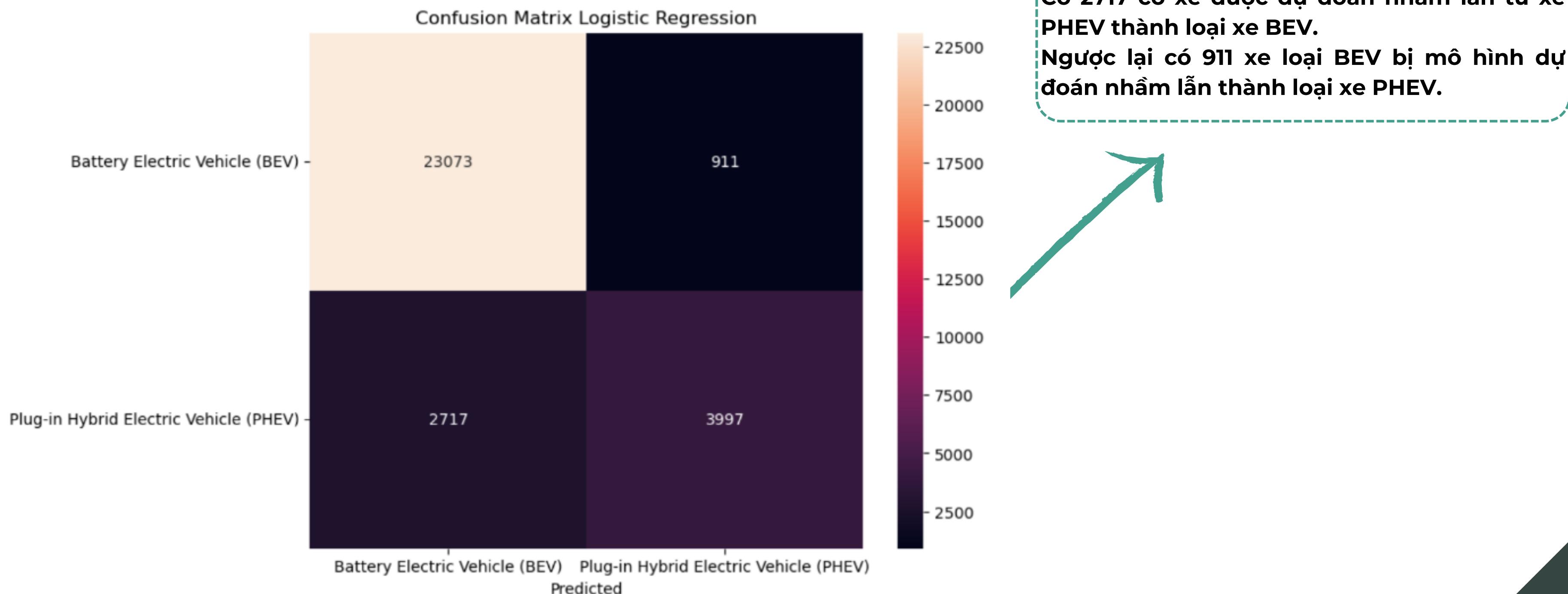


Mô hình có độ chính xác (Accuracy) là 88%

Có độ chính xác là 89% khi dự đoán lớp BEV và 81% khi dự đoán lớp PHEV.

Logistic Regression

- **Ma trận nhầm lẫn của mô hình Hồi quy Logistic**



K-NN

```
knn_model = KNeighborsClassifier(n_neighbors=5)

knn_model.fit(X_train, y_train)

y_pred = knn_model.predict(X_test)

print('-----K-Nearest Neighbors-----')
print(classification_report(y_test, y_pred))
```

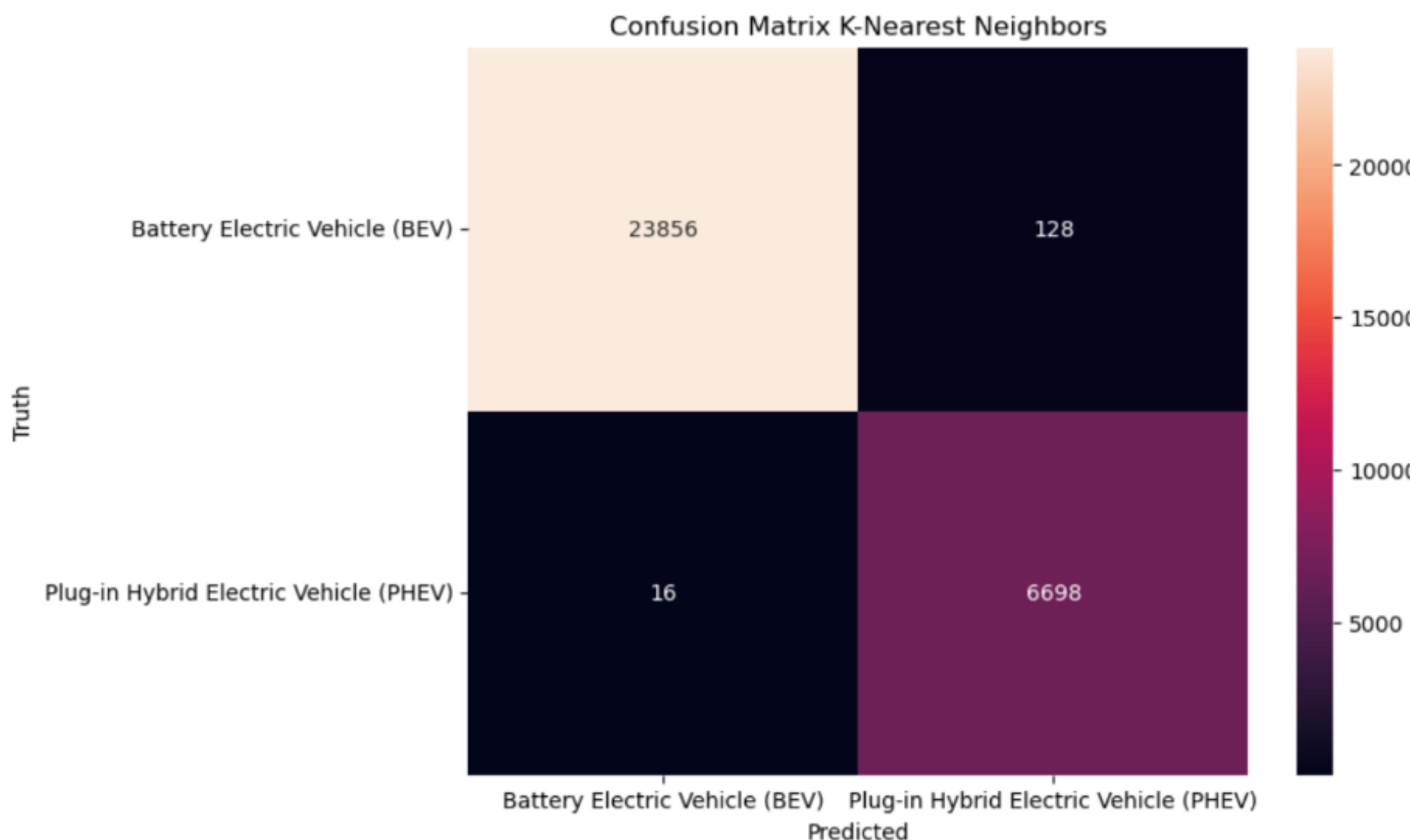
	precision	recall	f1-score	support
Battery Electric Vehicle (BEV)	1.00	1.00	1.00	23984
Plug-in Hybrid Electric Vehicle (PHEV)	0.99	1.00	0.99	6714
accuracy			1.00	30698
macro avg	0.99	1.00	0.99	30698
weighted avg	1.00	1.00	1.00	30698

Mô hình có độ chính xác (Accuracy) xấp xỉ 100%

Có độ chính xác xấp xỉ 100% khi dự đoán lớp BEV và 99% khi dự đoán lớp PHEV.

K-NN

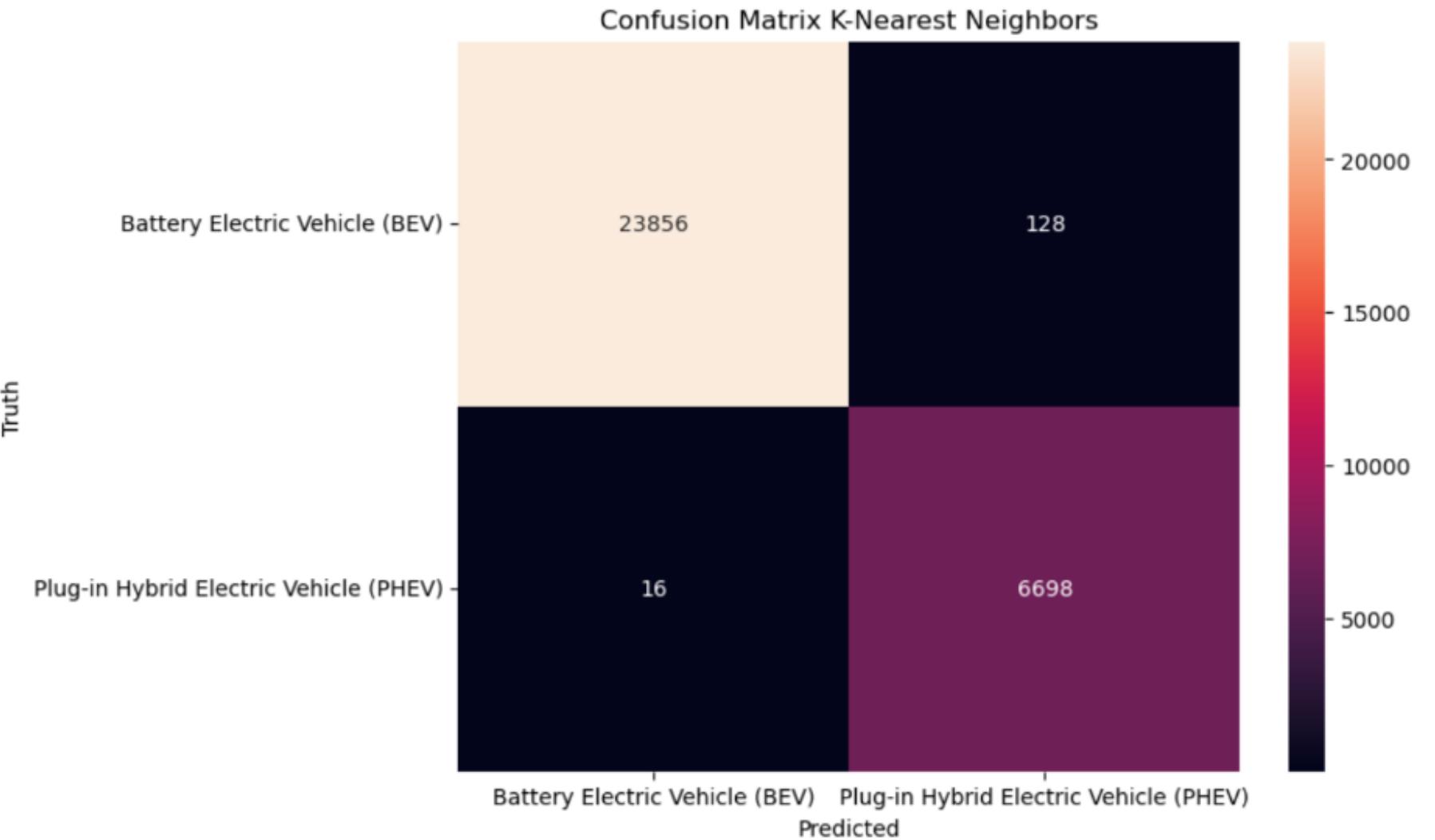
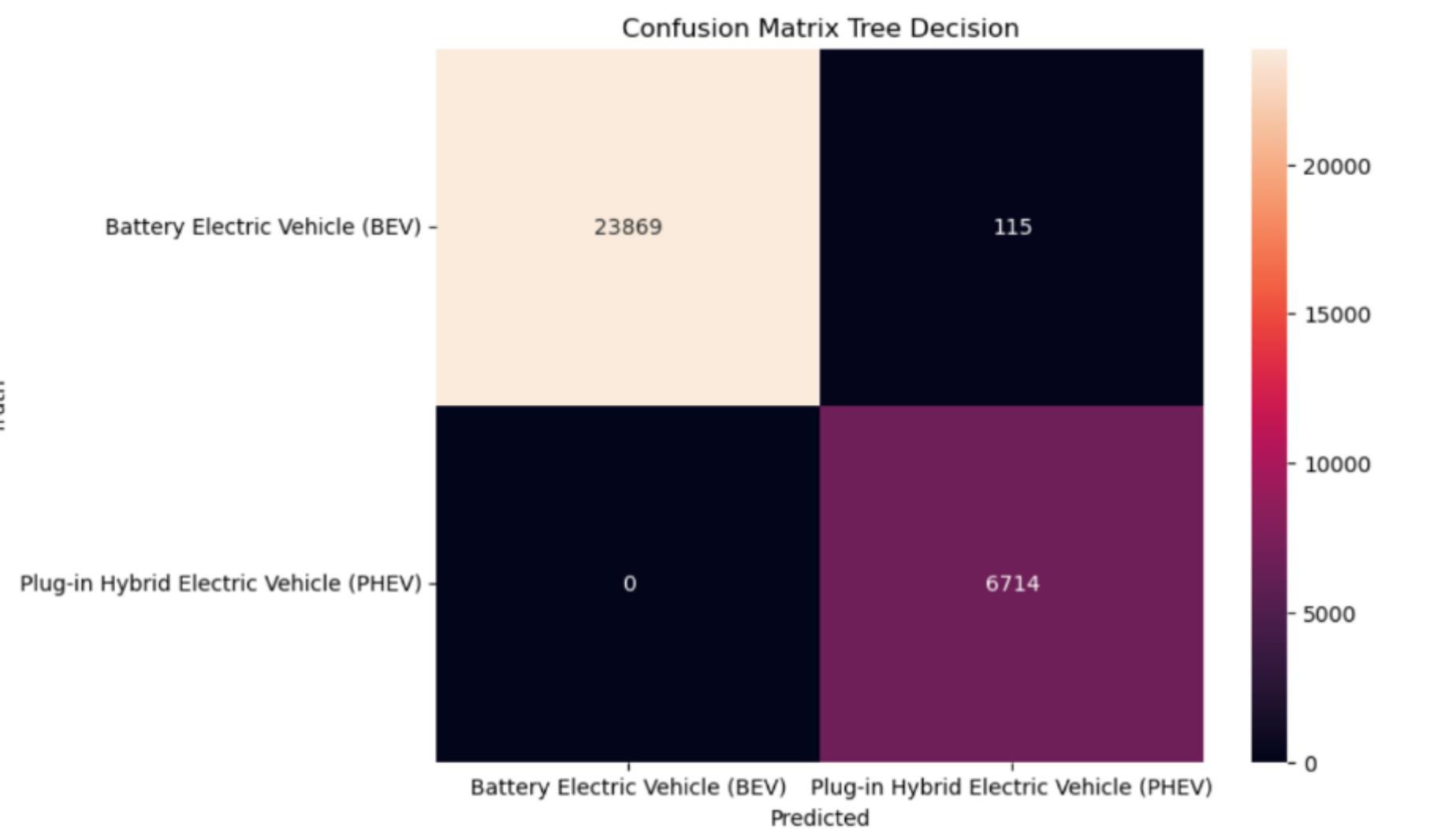
• Ma trận nhầm lẫn của mô hình KNN



Có 16 có xe được dự đoán nhầm lẫn từ xe PHEV thành loại xe BEV.
Ngược lại có 18 xe loại BEV bị mô hình dự đoán nhầm lẫn thành loại xe PHEV.

Machine Learning

Chọn ra 2 mô hình đó là Tree Decision và K-NN có giá trị Accuracy cao nhất 99%



**Sai lầm loại I và Sai lầm loại II của phân lớp dựa theo mô hình Tree Decision là 115(115 + 0)
Sai lầm loại I và Sai lầm loại II của phân lớp dựa theo mô hình K-NN là 144(128+16)**

Chọn mô hình có tổng số nhầm lẫn thấp nhất là mô hình Tree Decision.

**THANK YOU
FOR YOUR ATTENTION**

