In [1]:
```
import pandas as pd
import pickle
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
a=pd.read_csv("C:\\Users\\reshma_koduri\\OneDrive\\Documents\\archive 2\\Electric_Ve
a
```

Out[2]:

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type
0	2C4RC1N71H	Kitsap	Bremerton	WA	98311.0	2017	CHRYSLER	PACIFICA	Plug-in Hybrid Electric Vehicle (PHEV)
1	2C4RC1N7XL	Stevens	Colville	WA	99114.0	2020	CHRYSLER	PACIFICA	Plug-in Hybrid Electric Vehicle (PHEV)
2	KNDC3DLCXN	Yakima	Yakima	WA	98908.0	2022	KIA	EV6	Battery Electric Vehicle (BEV)
3	5YJ3E1EA0J	Kitsap	Bainbridge Island	WA	98110.0	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)
4	1N4AZ1CP7J	Thurston	Tumwater	WA	98501.0	2018	NISSAN	LEAF	Battery Electric Vehicle (BEV)
...	...	...	...	...	...	...	...	...	...
159462	KM8JBDA2XP	Skamania	Underwood	WA	98651.0	2023	HYUNDAI	TUCSON	Plug-in Hybrid Electric Vehicle (PHEV)
159463	1G1FZ6S02M	Skagit	Bow	WA	98232.0	2021	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)
159464	YV4H60CX2P	King	Sammamish	WA	98029.0	2023	VOLVO	XC90	Plug-in Hybrid Electric

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type
									Vehicle (PHEV)
159465	5YJ3E1EA7K	Whatcom	Bellingham	WA	98225.0	2019	TESLA	MODEL 3	Battery Electric Vehicle (BEV)
159466	7SAYGDEF6N	Island	Camano Island	WA	98282.0	2022	TESLA	MODEL Y	Battery Electric Vehicle (BEV)

159467 rows × 17 columns

In [3]:

```
a.head(5)
```

Out[3]:

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type	Clea Alternati Fu Vehic (CAF Eligibili
0	2C4RC1N71H	Kitsap	Bremerton	WA	98311.0	2017	CHRYSLER	PACIFICA	Plug-in Hybrid Electric Vehicle (PHEV)	Clea Alternati Fu Vehic Eligib
1	2C4RC1N7XL	Stevens	Colville	WA	99114.0	2020	CHRYSLER	PACIFICA	Plug-in Hybrid Electric Vehicle (PHEV)	Clea Alternati Fu Vehic Eligib
2	KNDC3DLCXN	Yakima	Yakima	WA	98908.0	2022	KIA	EV6	Battery Electric Vehicle (BEV)	Eligibil unknov as batte range h not b
3	5YJ3E1EA0J	Kitsap	Bainbridge Island	WA	98110.0	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clea Alternati Fu Vehic Eligib

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type	Clea Alternati Fu Vehic (CAF Eligibili
4	1N4AZ1CP7J	Thurston	Tumwater	WA	98501.0	2018	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clea Alternati Fu Vehic Eligib

In [4]:

```
a.tail(5)
```

Out[4]:

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type
159462	KM8JBDA2XP	Skamania	Underwood	WA	98651.0	2023	HYUNDAI	TUCSON	Plug-in Hybrid Electric Vehicle (PHEV)
159463	1G1FZ6S02M	Skagit	Bow	WA	98232.0	2021	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)
159464	YV4H60CX2P	King	Sammamish	WA	98029.0	2023	VOLVO	XC90	Plug-in Hybrid Electric Vehicle (PHEV)
159465	5YJ3E1EA7K	Whatcom	Bellingham	WA	98225.0	2019	TESLA	MODEL 3	Battery Electric Vehicle (BEV)
159466	7SAYGDEF6N	Island	Camano Island	WA	98282.0	2022	TESLA	MODEL Y	Battery Electric Vehicle (BEV)

In [5]:

```
a.describe()
```

Out[5]:

	Postal Code	Model Year	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	20
count	159463.000000	159467.000000	159467.000000	159467.00000	159106.000000	1.594670e+05	1.59

	Postal Code	Model Year	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	20:
mean	98170.373635	2020.192510	64.283319	1227.63716	29.261675	2.140242e+08	5.29
std	2453.354932	3.010564	94.634277	8930.03468	14.843878	7.959275e+07	1.62
min	1730.000000	1997.000000	0.000000	0.00000	1.000000	4.385000e+03	1.08
25%	98052.000000	2018.000000	0.000000	0.00000	18.000000	1.731016e+08	5.30
50%	98122.000000	2021.000000	14.000000	0.00000	33.000000	2.198450e+08	5.30
75%	98370.000000	2023.000000	84.000000	0.00000	43.000000	2.448363e+08	5.30
max	99577.000000	2024.000000	337.000000	845000.00000	49.000000	4.792548e+08	5.60

In [6]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159467 entries, 0 to 159466
Data columns (total 17 columns):
 #   Column                                             Non-Null Count   Dtype  
---  ------                                             --------------   -----  
 0   VIN (1-10)                                         159467 non-null  object 
 1   County                                             159463 non-null  object 
 2   City                                               159463 non-null  object 
 3   State                                              159467 non-null  object 
 4   Postal Code                                        159463 non-null  float64
 5   Model Year                                         159467 non-null  int64  
 6   Make                                               159467 non-null  object 
 7   Model                                              159467 non-null  object 
 8   Electric Vehicle Type                              159467 non-null  object 
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility  159467 non-null  object 
 10  Electric Range                                     159467 non-null  int64  
 11  Base MSRP                                          159467 non-null  int64  
 12  Legislative District                               159106 non-null  float64
 13  DOL Vehicle ID                                     159467 non-null  int64  
 14  Vehicle Location                                   159458 non-null  object 
 15  Electric Utility                                   159463 non-null  object 
 16  2020 Census Tract                                  159463 non-null  float64
dtypes: float64(3), int64(4), object(10)
memory usage: 20.7+ MB
```

In [7]:

```
list(a)
```

Out[7]:

```
['VIN (1-10)',
 'County',
 'City',
 'State',
 'Postal Code',
 'Model Year',
 'Make',
 'Model',
 'Electric Vehicle Type',
 'Clean Alternative Fuel Vehicle (CAFV) Eligibility',
 'Electric Range',
 'Base MSRP',
 'Legislative District',
 'DOL Vehicle ID',
 'Vehicle Location',
```

```
 'Electric Utility',
 '2020 Census Tract']
```

In [8]:
```
a.isna().sum()
```

Out[8]:
```
VIN (1-10)                                             0
County                                                 4
City                                                   4
State                                                  0
Postal Code                                            4
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                                 361
DOL Vehicle ID                                         0
Vehicle Location                                       9
Electric Utility                                       4
2020 Census Tract                                      4
dtype: int64
```

In [9]:
```
a.fillna(33,inplace=True)
```

In [10]:
```
a.isna().sum()
```

Out[10]:
```
VIN (1-10)                                           0
County                                               0
City                                                 0
State                                                0
Postal Code                                          0
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                                 0
DOL Vehicle ID                                       0
Vehicle Location                                     0
Electric Utility                                     0
2020 Census Tract                                    0
dtype: int64
```

In [11]:
```
b=a.drop(['VIN (1-10)','County','City','State','Vehicle Location','Electric Utility'
b
```

Out[11]:

	Model Year	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	2020 Census Tract
0	2017	Clean Alternative Fuel Vehicle Eligible	33	0	23.0	349437882	5.303509e+10
1	2020	Clean Alternative Fuel Vehicle Eligible	32	0	7.0	154690532	5.306595e+10

	Model Year	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	2020 Census Tract
2	2022	Eligibility unknown as battery range has not b...	0	0	14.0	219969144	5.307700e+10
3	2018	Clean Alternative Fuel Vehicle Eligible	215	0	23.0	476786887	5.303509e+10
4	2018	Clean Alternative Fuel Vehicle Eligible	151	0	35.0	201185253	5.306701e+10
...	...	...	...	...	...	...	...
159462	2023	Clean Alternative Fuel Vehicle Eligible	33	0	14.0	235949514	5.305995e+10
159463	2021	Eligibility unknown as battery range has not b...	0	0	40.0	148544168	5.305795e+10
159464	2023	Clean Alternative Fuel Vehicle Eligible	32	0	5.0	240200754	5.303303e+10
159465	2019	Clean Alternative Fuel Vehicle Eligible	220	0	40.0	156680590	5.307300e+10
159466	2022	Eligibility unknown as battery range has not b...	0	0	10.0	208285619	5.302997e+10

159467 rows × 7 columns

In [12]:

```
c=pd.get_dummies(b,dtype=int)
c
```

Out[12]:

	Model Year	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	2020 Census Tract	Clean Alternative Fuel Vehicle (CAFV) Eligibility_Clean Alternative Fuel Vehicle Eligible	Clean Alter Fuel Vehicle (Eligibility_Elig unkno battery rang not resea
0	2017	33	0	23.0	349437882	5.303509e+10	1	
1	2020	32	0	7.0	154690532	5.306595e+10	1	
2	2022	0	0	14.0	219969144	5.307700e+10	0	
3	2018	215	0	23.0	476786887	5.303509e+10	1	
4	2018	151	0	35.0	201185253	5.306701e+10	1	
...	...	...	...	...	...	...	...	
159462	2023	33	0	14.0	235949514	5.305995e+10	1	
159463	2021	0	0	40.0	148544168	5.305795e+10	0	
159464	2023	32	0	5.0	240200754	5.303303e+10	1	
159465	2019	220	0	40.0	156680590	5.307300e+10	1	

	Model Year	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	2020 Census Tract	Clean Alternative Fuel Vehicle (CAFV) Eligibility_Clean Alternative Fuel Vehicle Eligible	Clean Alter Fuel Vehicle (Eligibility_Elig unkno battery rang not resea
159466	2022	0	0	10.0	208285619	5.302997e+10	0	

159467 rows × 9 columns

In [13]:

```
b.groupby(['Clean Alternative Fuel Vehicle (CAFV) Eligibility']).count()
```

Out[13]:

Clean Alternative Fuel Vehicle (CAFV) Eligibility	Model Year	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	2020 Census Tract
Clean Alternative Fuel Vehicle Eligible	63824	63824	63824	63824	63824	63824
Eligibility unknown as battery range has not been researched	77195	77195	77195	77195	77195	77195
Not eligible due to low battery range	18448	18448	18448	18448	18448	18448

In [14]:

```
b['Clean Alternative Fuel Vehicle (CAFV) Eligibility'].unique()
```

Out[14]:

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

In [15]:

```
y=c['Electric Range']
y
```

Out[15]:

```
0          33
1          32
2           0
3         215
4         151
         ...
159462     33
159463      0
159464     32
159465    220
159466      0
Name: Electric Range, Length: 159467, dtype: int64
```

In [16]:

```
x=c.drop(['Electric Range'],axis=1)
x
```

Out[16]:

	Model Year	Base MSRP	Legislative District	DOL Vehicle ID	2020 Census Tract	Clean Alternative Fuel Vehicle (CAFV) Eligibility_Clean Alternative Fuel Vehicle Eligible	Clean Alternative Fuel Vehicle (CAFV) Eligibility_Eligibility unknown as battery range has not been researched	El b
0	2017	0	23.0	349437882	5.303509e+10	1	0	
1	2020	0	7.0	154690532	5.306595e+10	1	0	
2	2022	0	14.0	219969144	5.307700e+10	0	1	
3	2018	0	23.0	476786887	5.303509e+10	1	0	
4	2018	0	35.0	201185253	5.306701e+10	1	0	
...	...	...	...	...	...	...	...	
159462	2023	0	14.0	235949514	5.305995e+10	1	0	
159463	2021	0	40.0	148544168	5.305795e+10	0	1	
159464	2023	0	5.0	240200754	5.303303e+10	1	0	
159465	2019	0	40.0	156680590	5.307300e+10	1	0	
159466	2022	0	10.0	208285619	5.302997e+10	0	1	

159467 rows × 8 columns

In [17]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [18]:

```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(x_train,y_train)
```

Out[18]:

```
LinearRegression()
```

In [19]:

```
ypred=reg.predict(x_test)
ypred
```

Out[19]:

```
array([169.6015951 , 156.15857364, 158.32818466, ..., 140.75469442,
         0.67918796, 131.23103593])
```

In [20]:

```
from sklearn.metrics import r2_score
r2_score(y_test,ypred)
```

Out[20]:

```
0.624486235257481
```

In [21]:

```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,ypred)
```

Out[21]:

```
3352.7280556454557
```

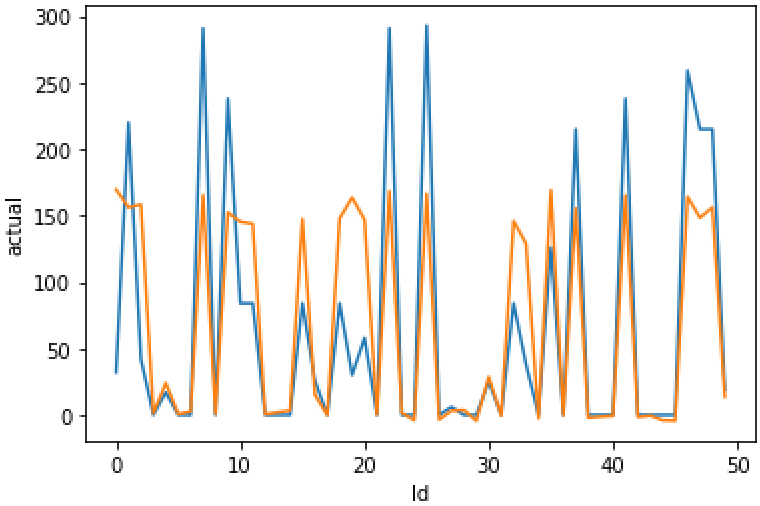
In [22]:
```
Results= pd.DataFrame(columns=['actual','Predicted'])
Results['actual']=y_test
Results['Predicted']=ypred
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(10)
```

Out[22]:

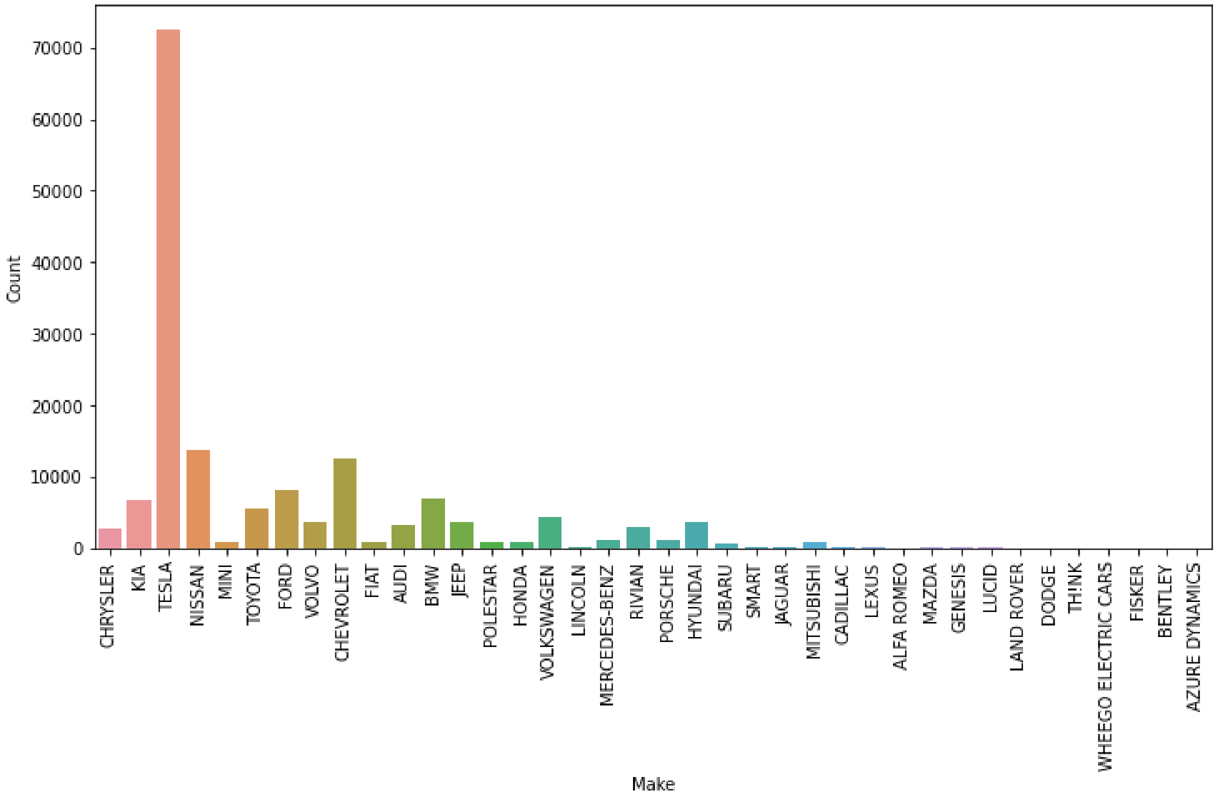
	index	actual	Predicted	Id
0	47762	32	169.601595	0
1	118101	220	156.158574	1
2	109691	42	158.328185	2
3	60166	0	1.242540	3
4	96268	17	24.047933	4
5	43696	0	0.721593	5
6	152277	0	2.555701	6
7	127242	291	165.509246	7
8	149989	0	0.979065	8
9	82736	238	152.332062	9

In [23]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='Id',y='actual',data=Results.head(50))
sns.lineplot(x='Id',y='Predicted',data=Results.head(50))
plt.plot()
```

Out[23]: []

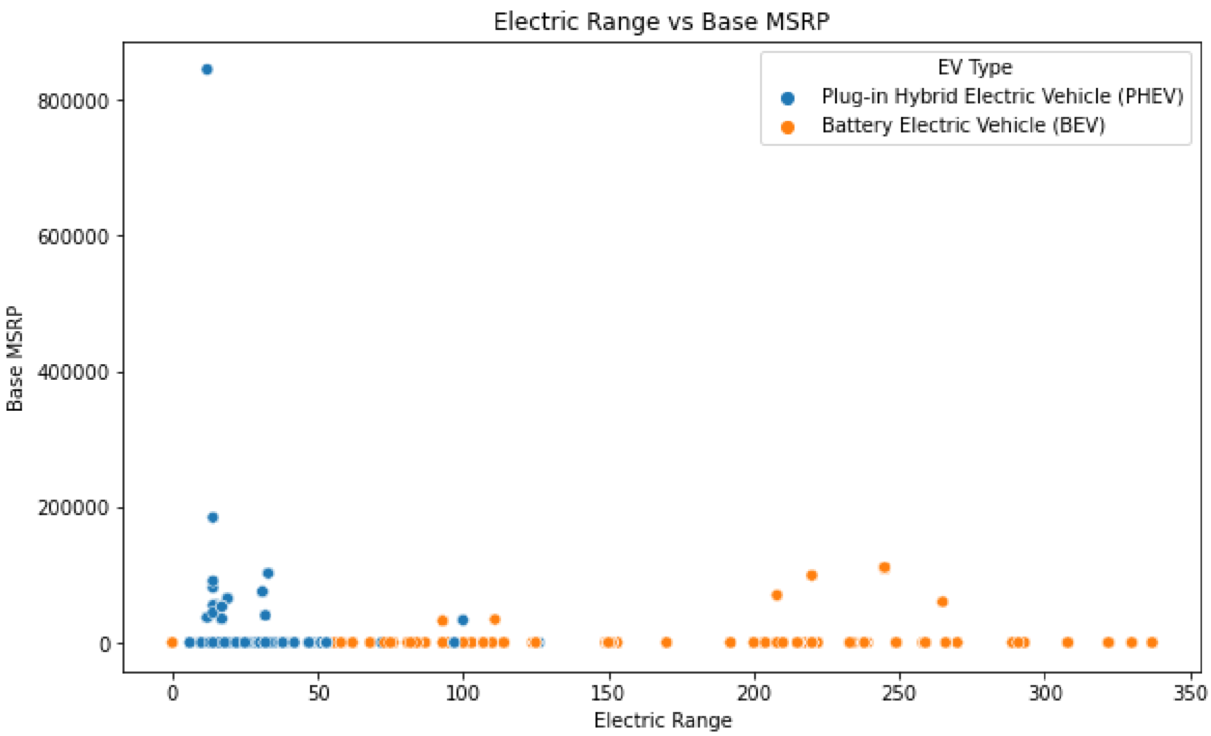


In [24]:
```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(12,6))
sns.countplot(data=a,x='Make')
plt.xticks(rotation=90)
plt.xlabel('Make')
plt.ylabel('Count')
plt.show()
```



In [25]:

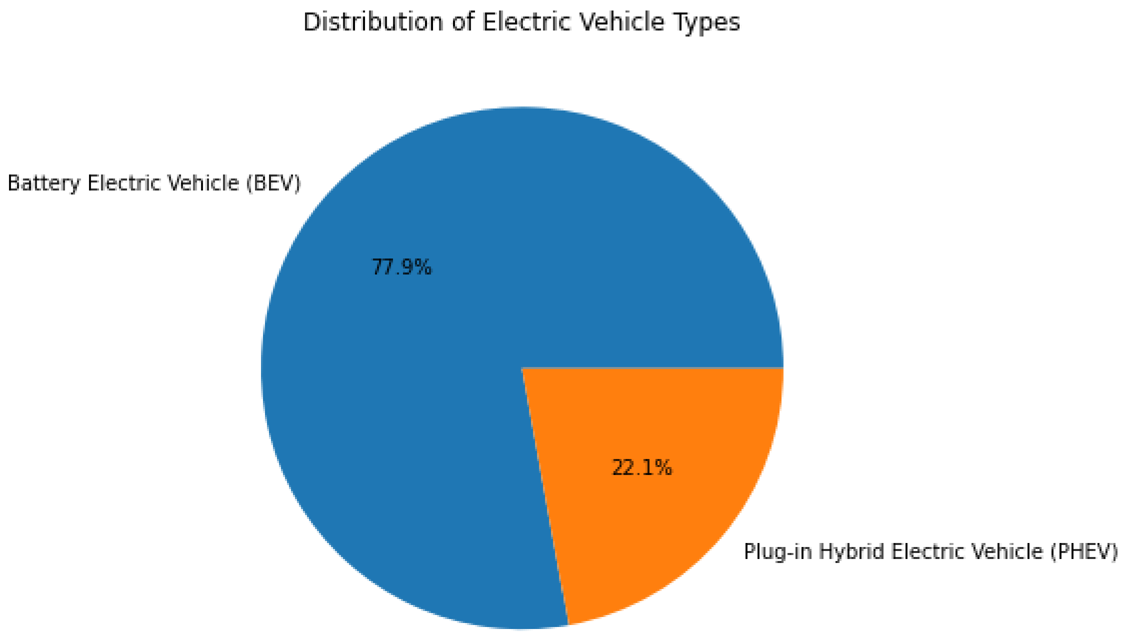
```
plt.figure(figsize=(10, 6))
sns.scatterplot(data=a, x='Electric Range', y='Base MSRP', hue='Electric Vehicle Typ
plt.title('Electric Range vs Base MSRP')
plt.xlabel('Electric Range')
plt.ylabel('Base MSRP')
plt.legend(title='EV Type')
plt.show()
```



In [26]:

```
plt.figure(figsize=(8, 6))
a['Electric Vehicle Type'].value_counts().plot.pie(autopct='%1.1f%%')
plt.title('Distribution of Electric Vehicle Types')
```

```
plt.ylabel('')
plt.show()
```



In [27]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Ridge
alpha = [1e-15, 1e-10, 1e-8, 1e-4, 1e-3,1e-2, 1, 5, 10, 20,30]
ridge=Ridge()
parameters={'alpha':alpha}
regressor=GridSearchCV(ridge,parameters)
regressor.fit(x_train,y_train)
```

Out[27]:
```
GridSearchCV(estimator=Ridge(),
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.0001, 0.001, 0.01, 1,
                                   5, 10, 20, 30]})
```

In [28]:
```
regressor.best_params_
```

Out[28]:
```
{'alpha': 1}
```

In [29]:
```
ridge=Ridge(1)
ridge.fit(x_train,y_train)
y_pred=ridge.predict(x_test)
y_pred
```

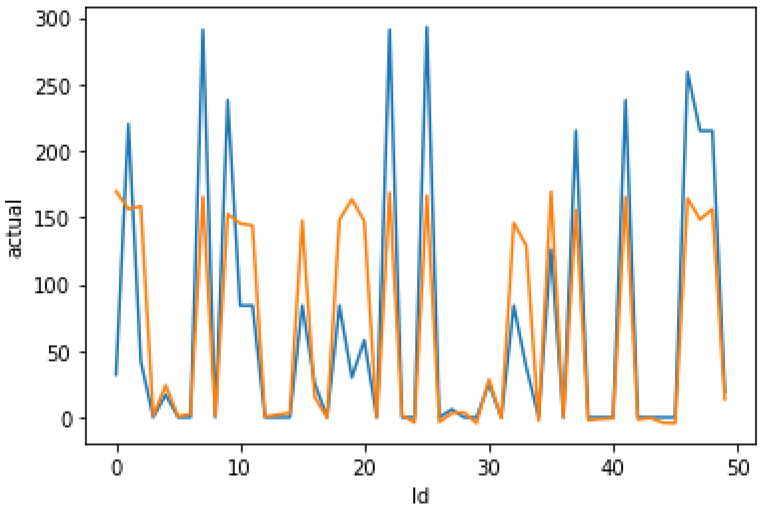
Out[29]:
```
array([169.59596321, 156.15559326, 158.32390049, ..., 140.75561725,
         0.68128732, 131.23020564])
```

In [30]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

Out[30]:
```
0.6244861665899158
```

In [31]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='Id',y='actual',data=Results.head(50))
sns.lineplot(x='Id',y='Predicted',data=Results.head(50))
plt.plot()
```

Out[31]: []



In [32]:

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestRegressor
reg=RandomForestRegressor()
n_estimators=[25,50,75,100,125,150,175,200]
criterion=['mse']
max_depth=[3,5,10]
parameters={'n_estimators': n_estimators,'criterion':criterion,'max_depth':max_depth
rfc_reg = GridSearchCV(reg, parameters)
rfc_reg.fit(x_train,y_train)
```

Out[32]:

```
GridSearchCV(estimator=RandomForestRegressor(),
             param_grid={'criterion': ['mse'], 'max_depth': [3, 5, 10],
                         'n_estimators': [25, 50, 75, 100, 125, 150, 175, 200]})
```

In [33]:

```
rfc_reg.best_params_
```

Out[33]:

```
{'criterion': 'mse', 'max_depth': 10, 'n_estimators': 200}
```

In [34]:

```
reg=RandomForestRegressor(criterion='mse', max_depth=10, n_estimators= 200)
```

In [35]:

```
reg.fit(x_train,y_train)
```

Out[35]:

```
RandomForestRegressor(max_depth=10, n_estimators=200)
```

In [41]:

```
y_pred=reg.predict(x_test)
y_pred
```

Out[41]:

```
array([ 33.69381541, 208.69647711,  85.57906289, ...,  69.13005261,
         0.        , 121.42459955])
```

In [42]:

```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

Out[42]:

```
0.8740342038272381
```

In [43]:

```
Results= pd.DataFrame(columns=['actual','Predicted'])
Results['actual']=y_test
Results['Predicted']=y_pred
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(10)
```

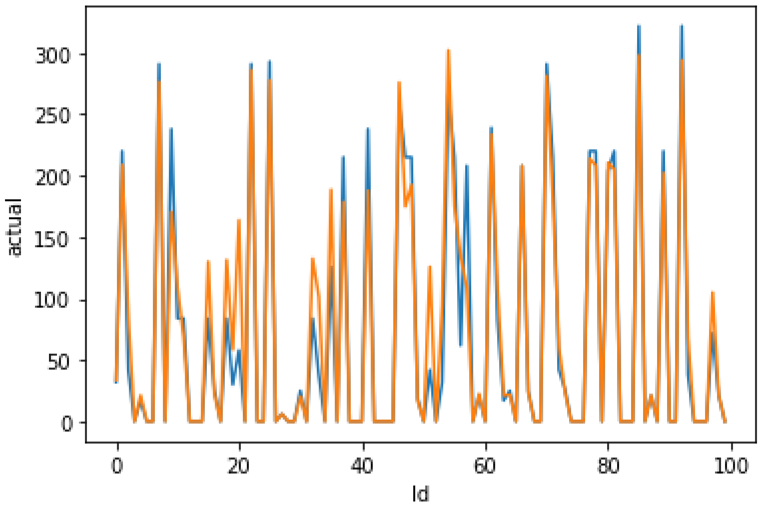
Out[43]:

	index	actual	Predicted	Id
0	47762	32	33.693815	0
1	118101	220	208.696477	1
2	109691	42	85.579063	2
3	60166	0	0.000000	3
4	96268	17	21.218208	4
5	43696	0	0.000000	5
6	152277	0	0.000000	6
7	127242	291	276.525236	7
8	149989	0	0.000000	8
9	82736	238	170.516474	9

In [45]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='Id',y='actual',data=Results.head(100))
sns.lineplot(x='Id',y='Predicted',data=Results.head(100))
plt.plot()
```

Out[45]: []



In [ ]: