In [1]:
```
### Electric Vehicle Sales by State in India ###
```

In [2]:
```
## EV_Dataset==
# This dataset is valuable for analysts, data scientists, and researchers aiming to
# understand electric vehicle (EV) adoption trends across India. It is versatile and ideal
# for geographic market segmentation, trend analysis, and predictive modeling. By
# offering insights into regional EV sales patterns, the dataset supports strategic
# decision-making in market planning and infrastructure investment.
```

In [3]:
```
# Step 1: Data Collection
```

In [4]:
```
# Import necessary libraries
import pandas as pd
import numpy as np
```

In [5]:
```
# Load the dataset
df = pd.read_csv("ev.csv")
df
```

Out[5]:

	Year	Month_Name	Date	State	Vehicle_Class	Vehicle_Category	Vehicle_Type	EV_Sales_Quantity
0	2014.0	jan	1/1/2014	Andhra Pradesh	ADAPTED VEHICLE	Others	Others	0.0
1	2014.0	jan	1/1/2014	Andhra Pradesh	AGRICULTURAL TRACTOR	Others	Others	0.0
2	2014.0	jan	1/1/2014	Andhra Pradesh	AMBULANCE	Others	Others	0.0
3	2014.0	jan	1/1/2014	Andhra Pradesh	ARTICULATED VEHICLE	Others	Others	0.0
4	2014.0	jan	1/1/2014	Andhra Pradesh	BUS	Bus	Bus	0.0
...	...	...	...	...	...	...	...	...
96840	2023.0	dec	12/1/2023	Andaman & Nicobar Island	MOTOR CAR	4-Wheelers	4W_Personal	1.0
96841	2023.0	dec	12/1/2023	Andaman & Nicobar Island	MOTOR CYCLE/SCOOTER-USED FOR HIRE	2-Wheelers	2W_Shared	5.0
96842	2023.0	dec	12/1/2023	Andaman & Nicobar Island	OMNI BUS	Bus	Bus	0.0
96843	2023.0	dec	12/1/2023	Andaman & Nicobar Island	THREE WHEELER (GOODS)	3-Wheelers	3W_Goods	0.0
96844	2023.0	dec	12/1/2023	Andaman & Nicobar Island	THREE WHEELER (PASSENGER)	3-Wheelers	3W_Shared	0.0

96845 rows × 8 columns

In [6]:
```
# Display the first few rows of the dataset
print(df.head())
```

```
   Year Month_Name  Date                State  Vehicle_Class  \
0  2014.0        jan  1/1/2014  Andhra Pradesh  ADAPTED VEHICLE
1  2014.0        jan  1/1/2014  Andhra Pradesh  AGRICULTURAL TRACTOR
2  2014.0        jan  1/1/2014  Andhra Pradesh             AMBULANCE
3  2014.0        jan  1/1/2014  Andhra Pradesh  ARTICULATED VEHICLE
4  2014.0        jan  1/1/2014  Andhra Pradesh                   BUS

  Vehicle_Category Vehicle_Type  EV_Sales_Quantity
0           Others       Others                0.0
1           Others       Others                0.0
2           Others       Others                0.0
3           Others       Others                0.0
4              Bus          Bus                0.0
```

In [7]:
```
# Step 2: Data Preprocessing
```

In [8]:
```
# Convert 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'])
```

In [9]:
```
# Check for missing values
print(df.isnull().sum())
```

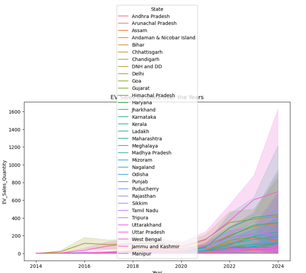
```
Year                   0
Month_Name             0
Date                   0
State                  0
Vehicle_Class          0
Vehicle_Category       0
Vehicle_Type           0
EV_Sales_Quantity      0
dtype: int64
```

In [10]:
```
# Fill missing values (if any) using median for numerical columns or mode for categorical columns
df['EV_Sales_Quantity'].fillna(df['EV_Sales_Quantity'].median(), inplace=True)
df.fillna(df.mode().iloc[0], inplace=True)
```

In [11]:
```
## Step 3: Exploratory Data Analysis (EDA)
```

In [12]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [13]:
```
# Plot EV sales over the years
plt.figure(figsize=(15, 6))
sns.lineplot(data=df, x='Year', y='EV_Sales_Quantity',hue='State')
plt.title('EV Sales by State over the Years')
plt.show()
```

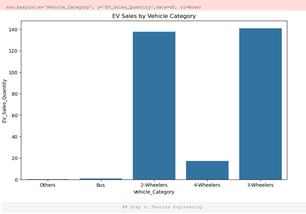


In [14]:
```
# Plot sales by vehicle category
plt.figure(figsize=(15, 6))
sns.barplot(x='Vehicle_Category', y='EV_Sales_Quantity',data=df, ci=None)
plt.title('EV Sales by Vehicle Category')
plt.show()
```

```
C:\Users\intel\AppData\Local\Temp\ipykernel_9284\7618446.py:3: FutureWarning:

The 'ci' parameter is deprecated. Use 'errorbar=None' for the same effect.

  sns.barplot(x='Vehicle_Category', y='EV_Sales_Quantity',data=df, ci=None)
```



In [15]:
```
## Step 4: Feature Engineering
```

In [16]:
```
# Extract Month and Day from the Date column
df['Month'] = df['Date'].dt.month
df['Day'] = df['Date'].dt.day
```

In [17]:
```
# Encode categorical variables using one-hot encoding
df_encoded = pd.get_dummies(df, columns=['State', 'Vehicle_Class', 'Vehicle_Category', 'Vehicle_Type'],drop_first=True)
df_encoded
```

Out[17]:

	Year	Month_Name	Date	EV_Sales_Quantity	Month	Day	State_Andhra Pradesh	State_Arunachal Pradesh	State_Assam	State_Bihar	...	Vehicle_Type_3W_Goods	Vehicle_Type_3W_Goods_LowSpeed	Vehicle_Type_3W_Personal	Vehicle_Typ
0	2014.0	jan	2014-01-01	0.0	1	1	True	False	False	False	...	False	False	False	
1	2014.0	jan	2014-01-01	0.0	1	1	True	False	False	False	...	False	False	False	
2	2014.0	jan	2014-01-01	0.0	1	1	True	False	False	False	...	False	False	False	
3	2014.0	jan	2014-01-01	0.0	1	1	True	False	False	False	...	False	False	False	
4	2014.0	jan	2014-01-01	0.0	1	1	True	False	False	False	...	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
96840	2023.0	dec	2023-12-01	1.0	12	1	False	False	False	False	...	False	False	False	
96841	2023.0	dec	2023-12-01	5.0	12	1	False	False	False	False	...	False	False	False	
96842	2023.0	dec	2023-12-01	0.0	12	1	False	False	False	False	...	False	False	False	
96843	2023.0	dec	2023-12-01	0.0	12	1	False	False	False	False	...	True	False	False	
96844	2023.0	dec	2023-12-01	0.0	12	1	False	False	False	False	...	False	False	False	

96845 rows × 126 columns

In [18]:
```
# Drop unnecessary columns (like date, Month_Name (if already extracted into numerical values)
df_encoded.drop(['Date', 'Month_Name'], axis=1, inplace=True)
```

In [19]:
```
# Step 5: Modeling
```

In [20]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
```

In [21]:
```
# Split the data into features and target variable
X = df_encoded.drop('EV_Sales_Quantity', axis=1)
y = df_encoded['EV_Sales_Quantity']
```

In [22]:
```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.2, random_state=42)
```

In [23]:
```
# Initialize the model
model = RandomForestRegressor(n_estimators=100,random_state=42)
```

In [24]:
```
# Train the model
model.fit(X_train, y_train)
```

Out[24]:
```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```

In [25]:
```
# Make predictions
y_pred = model.predict(X_test)
y_pred
```

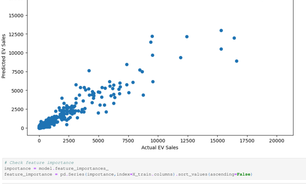
Out[25]:
```
array([ 0. ,  0. ,  0. , ..., 0. , 193.34,  0. ])
```

In [26]:
```
# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
```

In [27]:
```
print("Root Mean Squared Error: {:#.4f}".
Root Mean Squared Error: 130.5817523772101
```

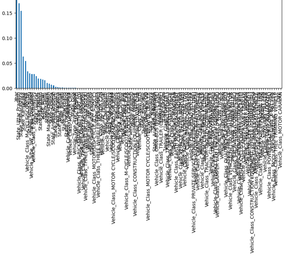
In [28]:
```
## Step 6: Model Evaluation
```

In [29]:
```
# Plot actual vs predicted sales
plt.figure(figsize=(15, 6))
plt.scatter(y_test, y_pred)
plt.xlabel('Actual EV Sales')
plt.ylabel('Actual EV Sales')
plt.xlabel('Actual EV Sales')
plt.ylabel('Predicted EV Sales')
plt.show()
```



In [30]:
```
# Check feature importance
importance = model.feature_importances_
feature_importance = pd.Series(importance,index=X_train.columns).sort_values(ascending=False)
```

In [31]:
```
# Plot top 20 most important features
plt.figure(figsize=(15, 6))
feature_importance.plot.bar(color='blue')
plt.title('Feature Importance')
plt.show()
```


Feature Importance
Year
Month
Day
State_Andhra Pradesh
State_Arunachal Pradesh
State_Assam
State_Bihar
State_Chandigarh
State_Chhattisgarh
State_DNH and DD
State_Delhi
State_Goa
State_Gujarat
State_Haryana
State_Himachal Pradesh
State_Jammu and Kashmir
State_Jharkhand
State_Karnataka
State_Kerala
State_Ladakh
State_Madhya Pradesh
State_Maharashtra
State_Manipur
State_Meghalaya
State_Mizoram
State_Nagaland
State_Odisha
State_Puducherry
State_Punjab
State_Rajasthan
State_Sikkim
State_Tamil Nadu
State_Tripura
State_Uttar Pradesh
State_Uttarakhand
State_West Bengal
Vehicle_Class_ADAPTED VEHICLE
Vehicle_Class_AGRICULTURAL TRACTOR
Vehicle_Class_AMBULANCE
Vehicle_Class_ARTICULATED VEHICLE
Vehicle_Class_BUS
Vehicle_Class_CONSTRUCTION EQUIPMENT VEHICLE
Vehicle_Class_DUMPER/TIPPER
Vehicle_Class_E-RICKSHAW(P)
Vehicle_Class_E-RICKSHAW WITH CART (G)
Vehicle_Class_EARTH MOVERS
Vehicle_Class_FIRE FIGHTING VEHICLE
Vehicle_Class_MOTOR CAR
Vehicle_Class_MOTOR CYCLE/SCOOTER-USED FOR HIRE
Vehicle_Class_MOTOR CYCLE/SCOOTER-WITH TRAILER
Vehicle_Class_OMNI BUS
Vehicle_Class_PRIVATE SERVICE VEHICLE
Vehicle_Class_STATE PROJECT VEHICLE
Vehicle_Class_THREE WHEELER (GOODS)
Vehicle_Class_THREE WHEELER (PASSENGER)
Vehicle_Class_TRAILER
Vehicle_Class_TRACTOR-COMMERCIAL
Vehicle_Class_Total
Vehicle_Class_Unclassified
Vehicle_Class_Unclassified
Vehicle_Class_Unclassified