In [10]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

In [11]:
```
# Load the dataset
df = pd.read_csv('EVPD.csv')

# Display the first few rows to understand the structure
print(df.head())
```

```
   VIN (1-10)    County      City State  Postal Code  Model Year     Make  \
0  KM8K33AGXL      King   Seattle    WA      98103.0        2020  HYUNDAI
1  1C4RJYB61N      King   Bothell    WA      98011.0        2022     JEEP
2  1C4RJYD61P    Yakima    Yakima    WA      98908.0        2023     JEEP
3  5YJ3E1EA7J      King  Kirkland    WA      98034.0        2018    TESLA
4  WBY7Z8C5XJ  Thurston   Olympia    WA      98501.0        2018      BMW

            Model                   Electric Vehicle Type  \
0            KONA          Battery Electric Vehicle (BEV)
1  GRAND CHEROKEE  Plug-in Hybrid Electric Vehicle (PHEV)
2  GRAND CHEROKEE  Plug-in Hybrid Electric Vehicle (PHEV)
3         MODEL 3          Battery Electric Vehicle (BEV)
4              I3  Plug-in Hybrid Electric Vehicle (PHEV)

  Clean Alternative Fuel Vehicle (CAFV) Eligibility  Electric Range  \
0           Clean Alternative Fuel Vehicle Eligible             258
1             Not eligible due to low battery range              25
2             Not eligible due to low battery range              25
3           Clean Alternative Fuel Vehicle Eligible             215
4           Clean Alternative Fuel Vehicle Eligible              97

   Base MSRP  Legislative District  DOL Vehicle ID  \
0          0                 43.0       249675142
1          0                  1.0       233928502
2          0                 14.0       229675939
3          0                 45.0       104714466
4          0                 22.0       185498386

                 Vehicle Location  \
0     POINT (-122.34301 47.659185)
1     POINT (-122.20578 47.762405)
2  POINT (-120.6027202 46.5965625)
3     POINT (-122.209285 47.71124)
4     POINT (-122.89692 47.043535)

                              Electric Utility  2020 Census Tract
0   CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)       5.303300e+10
1  PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)       5.303302e+10
2                                    PACIFICORP       5.307700e+10
3  PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)       5.303302e+10
4                        PUGET SOUND ENERGY INC       5.306701e+10
```

In [12]:
```
# Check for missing values
print(df.isnull().sum())
#df = df.dropna()
```

```
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                                 341
DOL Vehicle ID                                         0
Vehicle Location                                       7
Electric Utility                                       3
2020 Census Tract                                      3
dtype: int64
```

In [13]:

```
X = df[['Model Year', 'Make']]  # Independent variables
y = df['Electric Range']        # Dependent variable

# Convert Make to numeric using label encoding
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
X['Make'] = le.fit_transform(X['Make'])

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Training set shape:", X_train.shape)
print("Testing set shape:", X_test.shape)

# Display first few rows of training data
print("\
First few rows of training features:")
print(X_train.head())
print("\
First few rows of training target:")
print(y_train.head())
```

```
Training set shape: (120385, 2)
Testing set shape: (30097, 2)
First few rows of training features:
        Model Year  Make
150211        2021    31
40535         2020    31
55800         2022    10
100144        2018    31
78788         2023    15
First few rows of training target:
150211      0
40535     291
55800       0
100144    215
78788      21
Name: Electric Range, dtype: int64
```

```
C:\Users\mahesh narke\AppData\Local\Temp\ipykernel_7432\1212729005.py:7: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/
indexing.html#returning-a-view-versus-a-copy
  X['Make'] = le.fit_transform(X['Make'])
```

In [24]:
```
# Create a Linear Regression model
model = LinearRegression()
# Train the model
model.fit(X_train[['Model Year']], y_train)
```

Out[24]:
```
LinearRegression
LinearRegression()
```

In [25]:
```
# Make predictions on the test set
predictions = model.predict(X_test[['Model Year']])
```

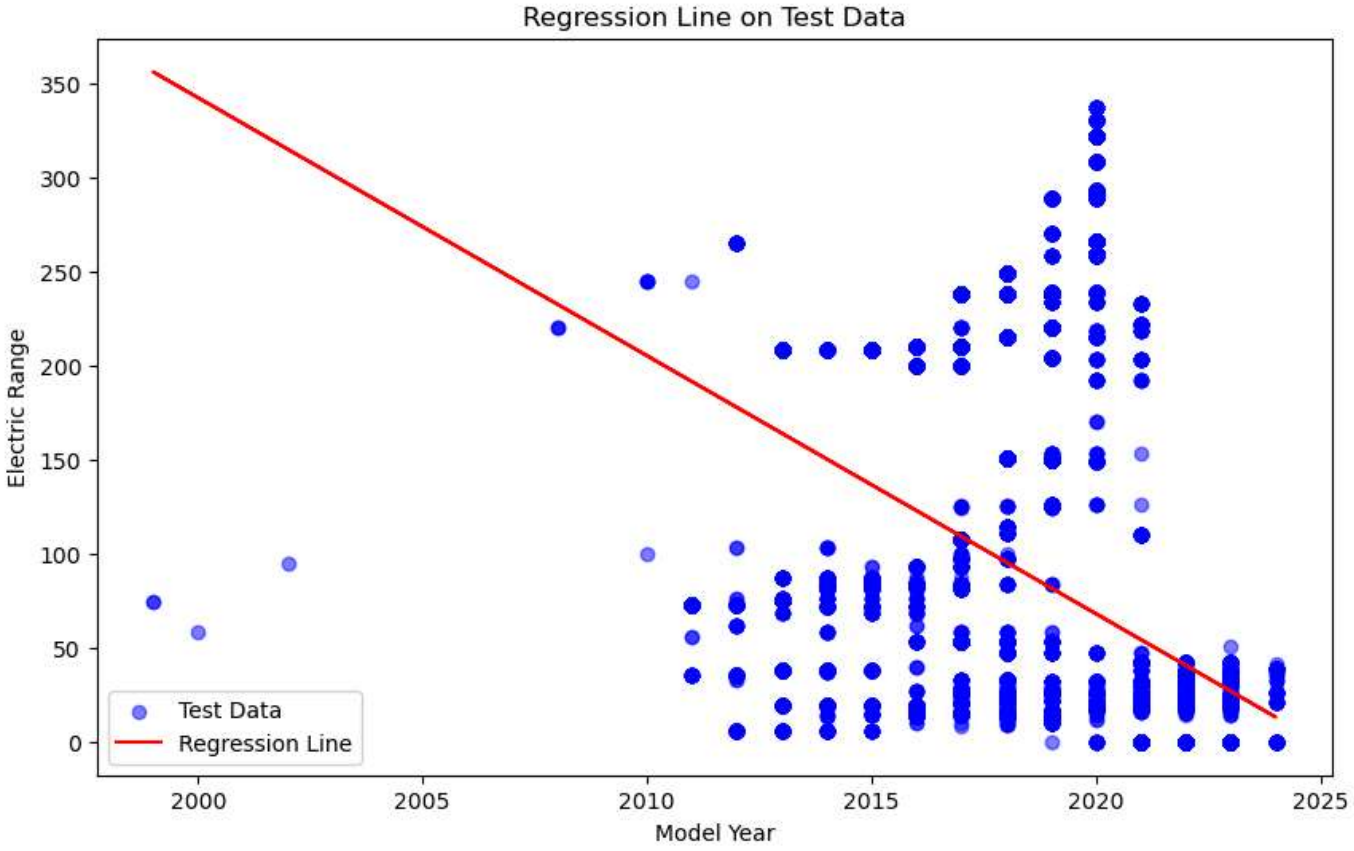
In [26]:
```
y_pred
```

Out[26]:
```
array([ -4.30875575,  28.9991295 , 155.52941663, ...,   8.99455993,
        78.91136134,  11.95085231])
```

In [27]:
```
# Calculate Mean Squared Error
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
# Calculate R-Squared value
r2 = r2_score(y_test, y_pred)
print(f"R-Squared: {r2}")
```

```
Mean Squared Error: 7300.891505474008
R-Squared: 0.21358449637394128
```

In [28]:
```
# Plot the test data and regression line
plt.figure(figsize=(10, 6))
plt.scatter(X_test['Model Year'], y_test, color='blue', alpha=0.5, label='Test Data')
plt.plot(X_test['Model Year'], predictions, color='red', label='Regression Line')
plt.xlabel('Model Year')
plt.ylabel('Electric Range')
plt.title('Regression Line on Test Data')
plt.legend()
plt.show()
```



In [ ]: