Step-by-Step Support Vector Regression Model - Canada Per Capita Income

Step 1: Prepare and Scale the Data

```
from sklearn.preprocessing import StandardScaler

X = df[['year']].values
y = df['income'].values.reshape(-1, 1)

scaler_X = StandardScaler()
scaler_y = StandardScaler()
X_scaled = scaler_X.fit_transform(X)
y_scaled = scaler_y.fit_transform(y).ravel()
```

Step 2: Train/Test Split

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y_scaled, test_size=0.2, random_state=42
)
```

Step 3: Fit SVR Model

```
from sklearn.svm import SVR

svr_model = SVR(kernel='rbf')
svr_model.fit(X_train, y_train)
```

Step 4: Make Predictions

```
y_pred_scaled = svr_model.predict(X_test)
y_pred = scaler_y.inverse_transform(y_pred_scaled.reshape(-1, 1))
```

Step 5: Evaluate

```
from sklearn.metrics import mean_squared_error, r2_score

y_test_orig = scaler_y.inverse_transform(y_test.reshape(-1, 1))
mse = mean_squared_error(y_test_orig, y_pred)
r2 = r2_score(y_test_orig, y_pred)
```

```
print("MSE:", mse)
print("R² Score:", r2)
```

Step 6: Visualize

```
import matplotlib.pyplot as plt
import numpy as np

X_plot_sorted = np.sort(X_scaled, axis=0)
y_plot_pred = svr_model.predict(X_plot_sorted)
y_plot_pred_orig = scaler_y.inverse_transform(y_plot_pred.reshape(-1, 1))

plt.scatter(X, y, color='blue', label='Actual Data')
plt.plot(scaler_X.inverse_transform(X_plot_sorted), y_plot_pred_orig, color='green', label='SVR Prediction')
plt.xlabel('Year')
plt.ylabel('Income')
plt.title('Support Vector Regression - Per Capita Income')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Evaluation Output

```
Mean Squared Error: 3,295,144.63
R² Score: 0.973
```