In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Step 1: Load the medical dataset
# Replace 'medical_dataset.csv' with your actual dataset file
data = pd.read_csv('data[1].csv')
data
```

Out[1]:

	id	gender	dob	zipcode	employment_status	education	marital_status	children	ancestry	avg_commute	daily_internet_use	available_vehicles	military_service
0	Amelia Nixon	female	1944-03-09	89136	retired	bachelors	married	1	Portugal	13.38	2.53	2	no
1	Clara Hicks	female	1966-07-02	94105	employed	phd/md	married	4	Sweden	15.16	6.77	2	no
2	Mason Brown	male	1981-05-31	89127	employed	masters	married	2	Germany	23.60	3.63	1	no
3	Michael Rice	male	1945-02-13	44101	retired	bachelors	married	2	Denmark	19.61	5.00	3	no
4	Eleanor Ritter	female	1939-09-03	89136	retired	masters	married	3	Austria	36.55	7.75	1	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1995	Ethan Johnson	male	1942-04-13	89127	retired	masters	married	1	Switzerland	28.48	5.88	1	no
1996	Natalia Dominguez	female	1963-05-10	60612	unemployed	highschool	married	4	Denmark	21.09	5.92	0	no
1997	Joseph Zuniga	male	1965-07-12	94110	employed	bachelors	married	2	Russia	30.80	4.91	2	no
1998	Daniel Murphy	male	1926-08-10	90015	retired	bachelors	married	3	Finland	37.56	2.69	0	no
1999	Samuel Harris	male	1948-11-22	43210	retired	bachelors	married	5	Scotland	49.34	4.77	3	no

2000 rows × 14 columns

In [2]:

```
X= data[['daily_internet_use']]
y=data['available_vehicles']
```

In [3]:

```
# Split the data 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 [4]:

```
# Specify the degree of the polynomial (e.g., quadratic, cubic)
degree = 2

# Create polynomial features
poly_features = PolynomialFeatures(degree=degree)
X_train_poly = poly_features.fit_transform(X_train)
X_test_poly = poly_features.transform(X_test)
```

In [5]:

```
# Create a polynomial regression model
model = LinearRegression()

# Train the model on the polynomial features
model.fit(X_train_poly, y_train)
```

Out[5]:

```
▾ LinearRegression
LinearRegression()
```

In [6]:

```
# Make predictions on the testing data
y_pred = model.predict(X_test_poly)

# Evaluate the model's performance
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Print evaluation metrics
print("Mean Squared Error (MSE):", mse)
print("R-squared (R2):", r2)
```

```
Mean Squared Error (MSE): 1.2090248183750671
R-squared (R2): -0.011080528004906443
```

In [7]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Step 1: Load the medical dataset
# Replace 'medical_dataset.csv' with your actual dataset file
data = pd.read_csv('city_temperature[1].csv')
data
```

```
C:\Users\nishi\AppData\Local\Temp\ipykernel_36512\2357946565.py:11: DtypeWarning: Columns (2) have mixed types. Specify dtype option on impor
t or set low_memory=False.
  data = pd.read_csv('city_temperature[1].csv')
```

Out[7]:

	Region	Country	State	City	Month	Day	Year	AvgTemperature
0	Africa	Algeria	NaN	Algiers	1	1	1995	64.2
1	Africa	Algeria	NaN	Algiers	1	2	1995	49.4
2	Africa	Algeria	NaN	Algiers	1	3	1995	48.8
3	Africa	Algeria	NaN	Algiers	1	4	1995	46.4
4	Africa	Algeria	NaN	Algiers	1	5	1995	47.9
...	...	...	...	...	...	...	...	...
2906322	North America	US	Additional Territories	San Juan Puerto Rico	7	27	2013	82.4
2906323	North America	US	Additional Territories	San Juan Puerto Rico	7	28	2013	81.6
2906324	North America	US	Additional Territories	San Juan Puerto Rico	7	29	2013	84.2
2906325	North America	US	Additional Territories	San Juan Puerto Rico	7	30	2013	83.8
2906326	North America	US	Additional Territories	San Juan Puerto Rico	7	31	2013	83.6

2906327 rows × 8 columns

In [9]:

```
X= data[['Day']]
y=data['AvgTemperature']
```

In [10]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [11]:

```
# Specify the degree of the polynomial (e.g., quadratic, cubic)
degree = 2

# Create polynomial features
poly_features = PolynomialFeatures(degree=degree)
X_train_poly = poly_features.fit_transform(X_train)
X_test_poly = poly_features.transform(X_test)
```

In [12]:

```
# Create a polynomial regression model
model = LinearRegression()

# Train the model on the polynomial features
model.fit(X_train_poly, y_train)
```

Out[12]:

```
▾ LinearRegression
LinearRegression()
```

In [13]:

```
# Make predictions on the testing data
y_pred = model.predict(X_test_poly)

# Evaluate the model's performance
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Print evaluation metrics
print("Mean Squared Error (MSE):", mse)
print("R-squared (R2):", r2)
```

```
Mean Squared Error (MSE): 1028.5467147230481
R-squared (R2): 8.186790901754115e-06
```

In [ ]: