```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

data = {
    'Room Size (sq ft)': [100, 200, 150, 300, 250, 120, 180, 220, 350,
270],
    'Number of Windows': [1, 2, 1, 3, 2, 1, 1, 2, 3, 2],
    'Temperature (°F)': [70, 72, 68, 75, 74, 69, 71, 73, 76, 74]
}

df = pd.DataFrame(data)

print(df.head())
```

```
   Room Size (sq ft)  Number of Windows  Temperature (°F)
0                100                  1                70
1                200                  2                72
2                150                  1                68
3                300                  3                75
4                250                  2                74
```

```python
X = df[['Room Size (sq ft)', 'Number of Windows']]
y = df['Temperature (°F)']

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

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

k = 3
knn_regressor = KNeighborsRegressor(n_neighbors=k)

knn_regressor.fit(X_train_scaled, y_train)
```

```
KNeighborsRegressor(n_neighbors=3)
```

```python
y_pred = knn_regressor.predict(X_test_scaled)

rmse = mean_squared_error(y_test, y_pred, squared=False)
r2 = r2_score(y_test, y_pred)

print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'R^2 Score: {r2}')
```

```
Root Mean Squared Error (RMSE): 1.6666666666666714
R^2 Score: 0.3055555555555516
```

```
new_room = [[250, 2]]
new_room_scaled = scaler.transform(new_room)
predicted_temperature = knn_regressor.predict(new_room_scaled)

print(f'Predicted Temperature: {predicted_temperature[0]} °F')
```

```
Predicted Temperature: 73.66666666666667 °F
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439:
UserWarning: X does not have valid feature names, but StandardScaler
was fitted with feature names
  warnings.warn(
```