Skup podataka je California Housing, a cilj je predvideti cenu nekretnine.

In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```python
from sklearn import model_selection
from sklearn import datasets
from sklearn import preprocessing
from sklearn import metrics
from sklearn import svm
```

In [3]:
```python
data = datasets.fetch_california_housing()
X = data.data
y = data.target
```

In [4]:
```python
X_train_and_val, X_test, y_train_and_val, y_test = model_selection.train_test_split(
    X, y, test_size = 0.3, random_state = 7)
```

In [5]:
```python
X_train, X_val, y_train, y_val = model_selection.train_test_split(
    X_train_and_val, y_train_and_val, train_size = 0.8, random_state = 7)
```

In [6]:
```python
scaler = preprocessing.StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_val = scaler.transform(X_val)
X_test = scaler.transform(X_test)
```

In [7]:
```python
Cs = [0.01, 0.05, 0.1, 0.5, 1, 5, 10]
gammas = [0.01, 0.05, 0.1, 0.5, 1, 5, 10]
```

In [8]:
```python
best_mse = 10000
best_C = None
best_gamma = None
```

In [9]:
```python
for C in Cs:
    for gamma in gammas:
        model = svm.SVR(kernel='rbf', gamma=gamma, C = C)
        model.fit(X_train, y_train)
        mse = metrics.mean_squared_error(y_val, model.predict(X_val))
        if mse < best_mse:
            best_mse = mse
            best_C = C
            best_gamma = gamma
```

In [10]:
```python
print('Najbolji hiperparametri modela su: ', best_C, best_gamma)
```
```
Najbolji hiperparametri modela su:  5 0.5
```

In [11]:
```python
print('Najmanja srednjekvadratna greska na validacionom skupu je: ', best_mse)
```
```
Najmanja srednjekvadratna greska na validacionom skupu je:  0.30594453711377617
```

In [12]:
```python
best_model = svm.SVR(kernel='rbf', gamma=best_gamma, C=best_C)
best_model.fit(X_train, y_train)
```

Out[12]:
```
▼        SVR
SVR(C=5, gamma=0.5)
```

```
In [13]: metrics.mean_squared_error(y_test, best_model.predict(X_test))
```

Out[13]: 0.30741759706586064

```
In [14]: metrics.r2_score(y_test, best_model.predict(X_test))
```

Out[14]: 0.7698766256707872

In [ ]: