```
import pandas as pd
df=pd.read_csv('/content/maaslar.csv')
```

```
df.head()
```

	unvan	Egitim Seviyesi	maas
0	Cayci	1	2250
1	Sekreter	2	2500
2	Uzman Yardimcisi	3	3000
3	Uzman	4	4000
4	Proje Yoneticisi	5	5500

```
from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()

for col in df.select_dtypes(include=['object']).columns:
    df[col] = encoder.fit_transform(df[col])
```

```
x = df.iloc[:,0]
y = df.iloc[:,1]
```

```
import numpy as np
x = np.array(x)
y = np.array(y)
```

```
x = x.reshape(-1,1)
y = y.reshape(-1,1)
```

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(
     x, y, test_size=0.5, random_state=42
)
print("Training sample:",x_train.shape)
print("testing sample:",x_test.shape)
```

```
Training sample: (5, 1)
testing sample: (5, 1)
```

```
from sklearn.svm import SVR
```

```
svr_model = SVR(kernel='rbf')
print("SVR model initialized with RBF kernel.")
```

```
SVR model initialized with RBF kernel.
```

```
param_grid = {'gamma': [0.2, 0.3, 0.4, 1], 'C': [0.1, 0.2, 0.3, 0.4, 0.5, 1]}
print("Parameter grid defined successfully.")
```

```
Parameter grid defined successfully.
```

```
print("Best parameters:", grid_search.best_params_)
print("Best cross-validation score:", grid_search.best_score_)
```

```
Best parameters: {'C': 0.1, 'gamma': 1}
Best cross-validation score: -0.5056448381555694
```

```
svr_model.fit(x_train, y_train)
print("SVR model trained successfully.")
```

```
SVR model trained successfully.
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:1408: DataConversionWarning: A column-vector y was passe
  y = column_or_1d(y, warn=True)
```

```
svr_model.fit(x_train, y_train.ravel())
print("SVR model trained successfully.")
```

```
SVR model trained successfully.
```

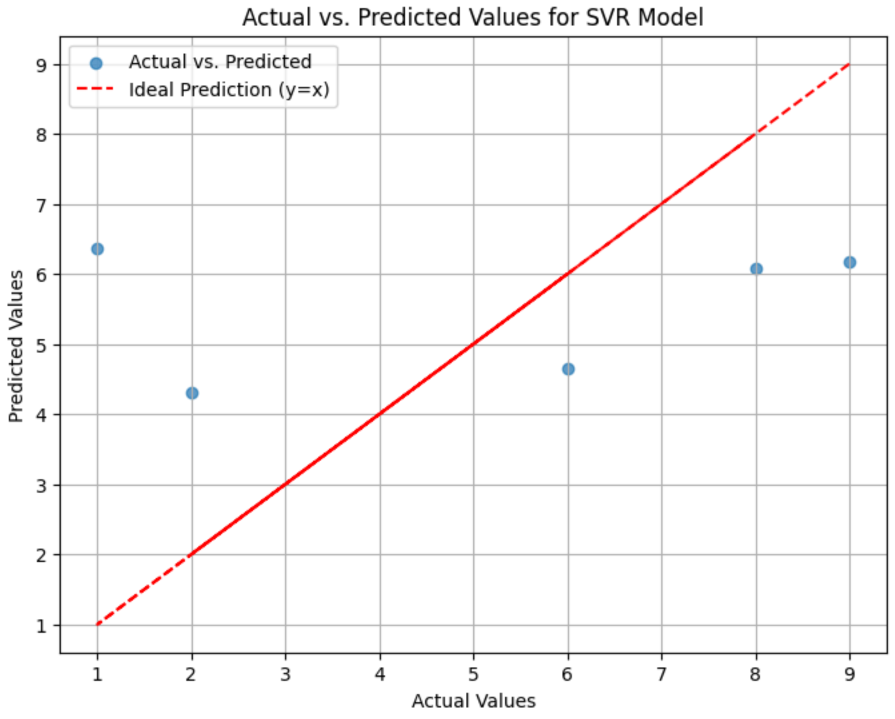
```
y_pred = svr_model.predict(x_test)
print("Predictions made successfully.")
```

```
Predictions made successfully.
```

```
from sklearn.metrics import r2_score, mean_squared_error
import numpy as np

r_squared = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print(f"R-squared: {r_squared:.4f}")
print(f"Mean Squared Error: {mse:.4f}")
print(f"Root Mean Squared Error: {rmse:.4f}")
```

```
R-squared: 0.0632
Mean Squared Error: 9.5180
Root Mean Squared Error: 3.0851
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, alpha=0.7, label='Actual vs. Predicted')
plt.plot(y_test, y_test, color='red', linestyle='--', label='Ideal Prediction (y=x)')
plt.title('Actual vs. Predicted Values for SVR Model')
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.legend()
plt.grid(True)
plt.show()
```


Actual vs. Predicted Values for SVR Model
● Actual vs. Predicted
Ideal Prediction (y=x)
Predicted Values
Actual Values