

```
# Import libraries
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Load the dataset
df = pd.read_csv("insurance.csv")
df.head()
```

	age	sex	bmi	children	smoker	region	charges	grid
0	19	female	27.900	0	yes	southwest	16884.92400	
1	18	male	33.770	1	no	southeast	1725.55230	
2	28	male	33.000	3	no	southeast	4449.46200	
3	33	male	22.705	0	no	northwest	21984.47061	
4	32	male	28.880	0	no	northwest	3866.85520	

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
# Encode categorical variables
le = LabelEncoder()
df['sex'] = le.fit_transform(df['sex'])
df['smoker'] = le.fit_transform(df['smoker'])
df['region'] = le.fit_transform(df['region'])

df.head()
```

	age	sex	bmi	children	smoker	region	charges	grid
0	19	0	27.900	0	1	3	16884.92400	
1	18	1	33.770	1	0	2	1725.55230	
2	28	1	33.000	3	0	2	4449.46200	
3	33	1	22.705	0	0	1	21984.47061	
4	32	1	28.880	0	0	1	3866.85520	

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
# Features and target
X = df.drop("charges", axis=1)
y = df["charges"]

# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
scaler_X = StandardScaler()
scaler_y = StandardScaler()

X_train_scaled = scaler_X.fit_transform(X_train)
X_test_scaled = scaler_X.transform(X_test)

# y needs reshaping for scaler
y_train_scaled = scaler_y.fit_transform(y_train.values.reshape(-1,1)).ravel()
y_test_scaled = scaler_y.transform(y_test.values.reshape(-1,1)).ravel()
```

```
from sklearn.svm import SVR

# RBF kernel
svr_rbf = SVR(kernel='rbf')
svr_rbf.fit(X_train_scaled, y_train_scaled)

# Polynomial kernel (degree=3)
svr_poly = SVR(kernel='poly', degree=3)
svr_poly.fit(X_train_scaled, y_train_scaled)
```

```
▼ SVR ⓘ ?  
SVR(kernel='poly')
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score  
import numpy as np  
  
models = {'SVR_RBF': svr_rbf, 'SVR_Poly': svr_poly}  
  
for name, model in models.items():  
    y_pred_scaled = model.predict(X_test_scaled)  
  
    # Convert back to original scale  
    y_pred = scaler_y.inverse_transform(y_pred_scaled.reshape(-1,1)).ravel()  
    y_test_orig = scaler_y.inverse_transform(y_test_scaled.reshape(-1,1)).ravel()  
  
    mae = mean_absolute_error(y_test_orig, y_pred)  
    rmse = np.sqrt(mean_squared_error(y_test_orig, y_pred))  
    r2 = r2_score(y_test_orig, y_pred)  
  
    print(f"{name} | MAE: {mae:.2f}, RMSE: {rmse:.2f}, R2: {r2:.4f}")
```

```
SVR_RBF | MAE: 2456.87, RMSE: 4575.72, R2: 0.8651  
SVR_Poly | MAE: 2338.54, RMSE: 4669.30, R2: 0.8596
```