

```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression, Ridge
from sklearn.metrics import mean_squared_error, r2_score
```

```
df = pd.read_csv('/content/sample_data/california_housing_train.csv')
df.head()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	popu
0	-114.31	34.19	15.0	5612.0	1283.0	
1	-114.47	34.40	19.0	7650.0	1901.0	
2	-114.56	33.69	17.0	720.0	174.0	
3	-114.57	33.64	14.0	1501.0	337.0	
4	-114.57	33.57	20.0	1454.0	326.0	

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

```
numeric_df = df.select_dtypes(include=['number'])
numeric_df.head()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	popu
0	-114.31	34.19	15.0	5612.0	1283.0	
1	-114.47	34.40	19.0	7650.0	1901.0	
2	-114.56	33.69	17.0	720.0	174.0	
3	-114.57	33.64	14.0	1501.0	337.0	
4	-114.57	33.57	20.0	1454.0	326.0	

Next steps: [Generate code with numeric\\_df](#) [New interactive sheet](#)

```
numeric_df = numeric_df.fillna(numeric_df.mean())
```

```
X = numeric_df.drop('median_house_value', axis=1)
y = numeric_df['median_house_value']
```

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

```
scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
lr = LinearRegression()
lr.fit(X_train, y_train)

y_pred_lr = lr.predict(X_test)
```

```
ridge = Ridge(alpha=1.0)
ridge.fit(X_train, y_train)

y_pred_ridge = ridge.predict(X_test)
```

```
mse_lr = mean_squared_error(y_test, y_pred_lr)
rmse_lr = np.sqrt(mse_lr)
r2_lr = r2_score(y_test, y_pred_lr)

print("Linear Regression Results:")
print("MSE:", mse_lr)
print("RMSE:", rmse_lr)
print("R2 Score:", r2_lr)
```

```
Linear Regression Results:
MSE: 4634658406.223357
RMSE: 68078.32552452621
R2 Score: 0.6636396350243802
```

```
mse_ridge = mean_squared_error(y_test, y_pred_ridge)
rmse_ridge = np.sqrt(mse_ridge)
r2_ridge = r2_score(y_test, y_pred_ridge)

print("\nRidge Regression Results:")
print("MSE:", mse_ridge)
print("RMSE:", rmse_ridge)
print("R2 Score:", r2_ridge)
```

Ridge Regression Results:  
MSE: 4634762362.709135  
RMSE: 68079.08902672784  
R2 Score: 0.6636320903817263

```
coeff_comparison = pd.DataFrame({  
    'Feature': X.columns,  
    'LinearRegression Coef': lr.coef_,  
    'Ridge Coef': ridge.coef_  
})  
  
coeff_comparison
```

	Feature	LinearRegression Coef	Ridge Coef	grid icon
0	longitude	-87098.433836	-86996.183107	edit icon
1	latitude	-91983.102014	-91881.849909	
2	housing_median_age	14256.837611	14266.106800	
3	total_rooms	-19175.064162	-19145.360231	
4	total_bedrooms	47893.657219	47777.029657	
5	population	-41374.290581	-41364.484397	
6	households	17328.745023	17409.948629	
7	median_income	76754.043069	76751.096787	

Next steps:

[Generate code with coeff\\_comparison](#)[New interactive sheet](#)