```
import pandas as pd, numpy as np, matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
```

```
path = "/mnt/data/used_cars_dataset.csv"
try:
    df = pd.read_csv(path)
except:
    from google.colab import files
    print("Upload used_cars_dataset.csv when prompted")
    uploaded = files.upload()
    df = pd.read_csv(next(iter(uploaded.keys())))

print("Shape:", df.shape)
display(df.head())
```

```
Upload used_cars_dataset.csv when prompted
```

Choose Files used_cars_dataset.csv

used_cars_dataset.csv(text/csv) - 354633 bytes, last modified: 11/9/2025 - 100% done

```
Saving used_cars_dataset.csv to used_cars_dataset.csv
Shape: (4340, 8)
```

	name	year	km_driven	fuel	seller_type	transmission	owner	selling_price
0	Maruti 800 AC	2007	70000	Petrol	Individual	Manual	First Owner	60000
1	Maruti Wagon R LXI Minor	2007	50000	Petrol	Individual	Manual	First Owner	135000
2	Hyundai Verna 1.6 SX	2012	100000	Diesel	Individual	Manual	First Owner	600000
3	Datsun RediGO T Option	2017	46000	Petrol	Individual	Manual	First Owner	250000
4	Honda Amaze VX i-DTEC	2014	141000	Diesel	Individual	Manual	Second Owner	450000

```
df = df.dropna().reset_index(drop=True)
for c in df.select_dtypes('object'):
    df[c] = LabelEncoder().fit_transform(df[c])
```

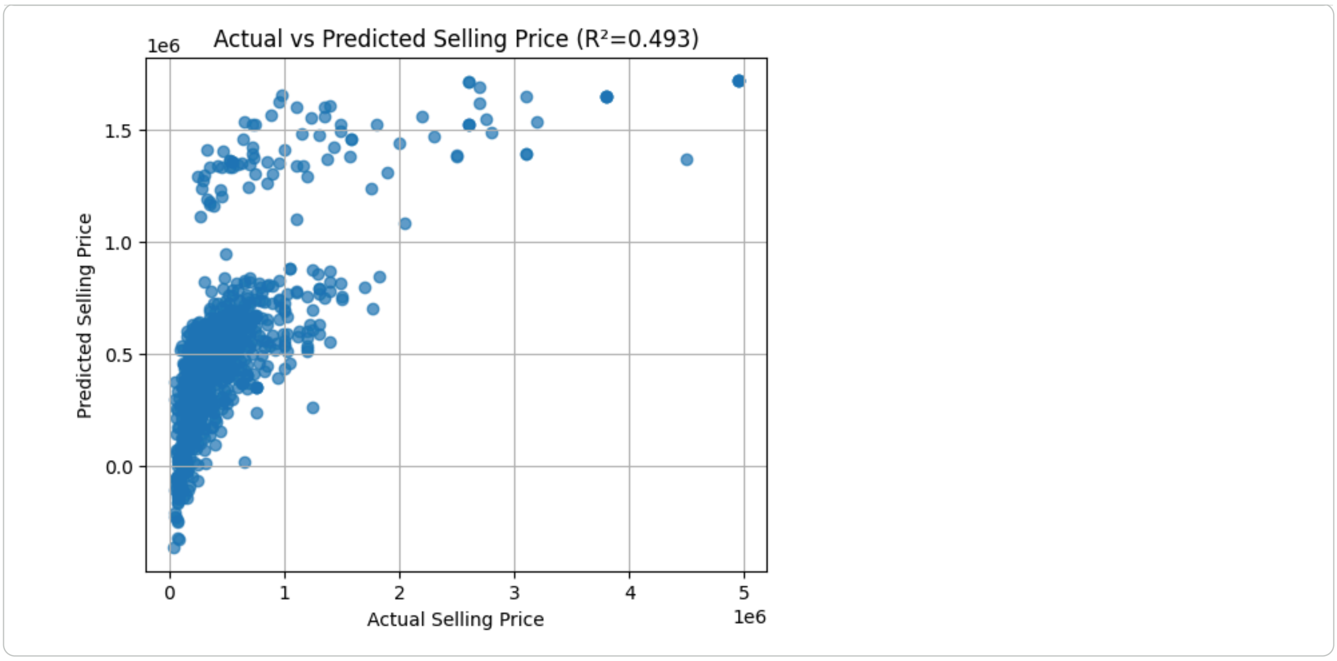
```
y = df['Selling_Price'] if 'Selling_Price' in df.columns else df.iloc[:,-1]
X = df.drop(columns=[y.name])
X = StandardScaler().fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
model = LinearRegression().fit(X_train, y_train)
y_pred = model.predict(X_test)
```

```
r2 = r2_score(y_test, y_pred)
mae, mse = mean_absolute_error(y_test, y_pred), mean_squared_error(y_test, y_pred)
print(f"R²: {r2:.4f}, MAE: {mae:.3f}, MSE: {mse:.3f}")
```

```
R²: 0.4930, MAE: 246377.165, MSE: 188260440693.097
```

```
plt.figure(figsize=(6,5))
plt.scatter(y_test, y_pred, alpha=0.7)
plt.xlabel("Actual Selling Price")
plt.ylabel("Predicted Selling Price")
plt.title(f"Actual vs Predicted Selling Price (R²={r2:.3f})")
plt.grid(True)
plt.show()
```


Actual vs Predicted Selling Price (R²=0.493)
1e6
1.5
1.0
0.5
0.0
Predicted Selling Price
0
1
2
3
4
5
1e6
Actual Selling Price