In [1]:

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

In [2]:

```
data=pd.read_csv('CarPrice_Assignment.csv')
print(data.head())
print(data.info())
```

```
   car_ID  symboling                   CarName fueltype aspiration doornumber
\
0       1          3        alfa-romero giulia      gas        std        two
1       2          3       alfa-romero stelvio      gas        std        two
2       3          1  alfa-romero Quadrifoglio      gas        std        two
3       4          2               audi 100 ls      gas        std       four
4       5          2                audi 100ls      gas        std       four

       carbody drivewheel enginelocation  wheelbase  ...  enginesize  \
0  convertible        rwd          front       88.6  ...         130
1  convertible        rwd          front       88.6  ...         130
2    hatchback        rwd          front       94.5  ...         152
3        sedan        fwd          front       99.8  ...         109
4        sedan        4wd          front       99.4  ...         136

   fuelsystem  boreratio  stroke compressionratio horsepower  peakrpm citympg
\
0        mpfi       3.47    2.68              9.0        111     5000      21
1        mpfi       3.47    2.68              9.0        111     5000      21
2        mpfi       2.68    3.47              9.0        154     5000      19
3        mpfi       3.19    3.40             10.0        102     5500      24
4        mpfi       3.19    3.40              8.0        115     5500      18

   highwaympg    price
0          27  13495.0
1          27  16500.0
2          26  16500.0
3          30  13950.0
4          22  17450.0

[5 rows x 26 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   car_ID            205 non-null    int64
 1   symboling         205 non-null    int64
 2   CarName           205 non-null    object
 3   fueltype          205 non-null    object
 4   aspiration        205 non-null    object
 5   doornumber        205 non-null    object
 6   carbody           205 non-null    object
 7   drivewheel        205 non-null    object
 8   enginelocation    205 non-null    object
 9   wheelbase         205 non-null    float64
 10  carlength         205 non-null    float64
 11  carwidth          205 non-null    float64
 12  carheight         205 non-null    float64
 13  curbweight        205 non-null    int64
 14  enginetype        205 non-null    object
 15  cylindernumber    205 non-null    object
 16  enginesize        205 non-null    int64
 17  fuelsystem        205 non-null    object
 18  boreratio         205 non-null    float64
 19  stroke            205 non-null    float64
 20  compressionratio  205 non-null    float64
 21  horsepower        205 non-null    int64
 22  peakrpm           205 non-null    int64
 23  citympg           205 non-null    int64
 24  highwaympg        205 non-null    int64
 25  price             205 non-null    float64
```

```
dtypes: float64(8), int64(8), object(10)
memory usage: 41.8+ KB
None
```

In [3]:

```
data=data.drop(['CarName','car_ID'],axis=1)
data = pd.get_dummies(data, drop_first=True)
```

In [4]:

```
X=data.drop('price', axis=1)
Y=data['price']
```

In [5]:

```
scaler = StandardScaler()
X=scaler.fit_transform(X)
Y=scaler.fit_transform(np.array(Y).reshape(-1, 1))
```

In [6]:

```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=42
```

In [7]:

```
sgd_model= SGDRegressor(max_iter=1000, tol=1e-3)
```

In [8]:

```
sgd_model.fit(X_train, Y_train)
```

```
C:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:1143: Da
taConversionWarning: A column-vector y was passed when a 1d array was expected.
Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[8]:

```
▾ SGDRegressor

SGDRegressor()
```

In [9]:

```
y_pred = sgd_model.predict(X_test)
```

In [10]:

```
mse = mean_squared_error(Y_test, y_pred)
r2=r2_score(Y_test,y_pred)
mae= mean_absolute_error(Y_test, y_pred)
```

In [11]:

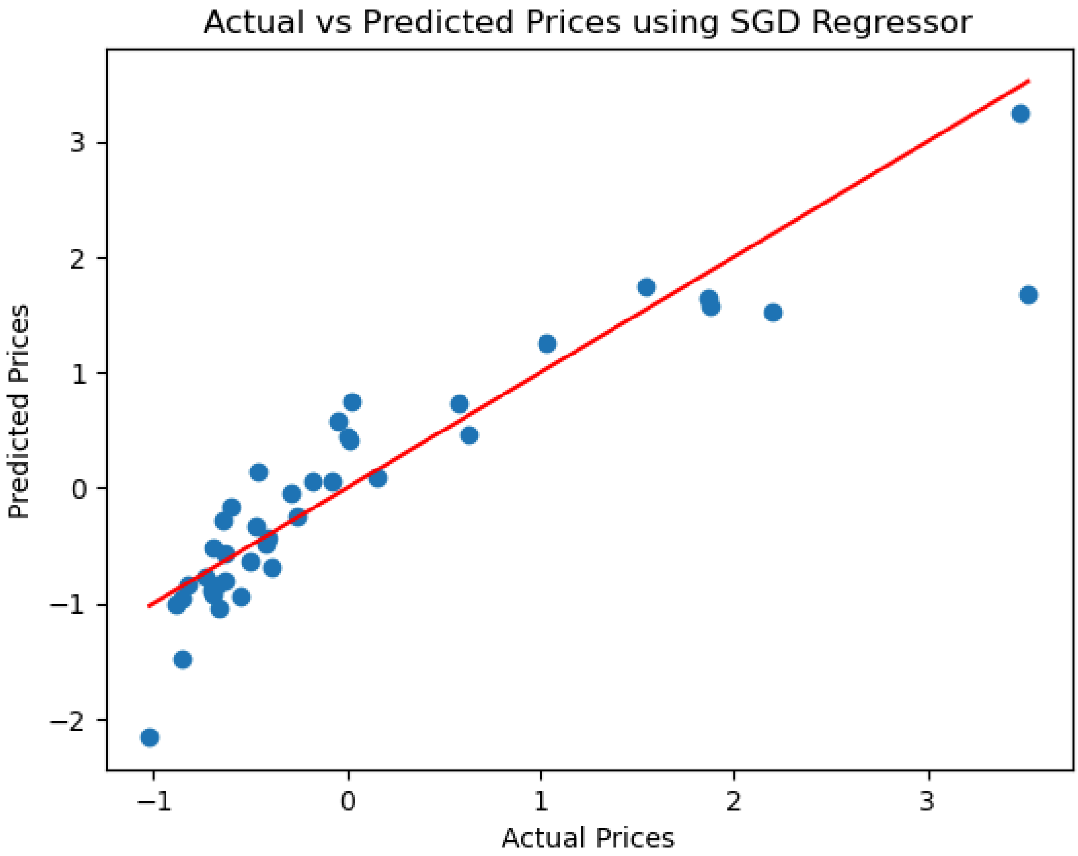
```
print('Name:RAGHUL.S ')
print('Reg no: 212225040325')
print("Mean Squared Error:",mse)
print("Mean Absolute Error:",mae)
print("R-Squared Score:",r2)
```

```
Name:RAGHUL.S
Reg no: 212225040325
Mean Squared Error: 0.20941302034292564
Mean Absolute Error: 0.3137138365510395
R-Squared Score: 0.8315273017580898
```

In [12]:
```
print("\nModel Coefficients:")
print("Coefficients:",sgd_model.coef_)
print("Intercept:",sgd_model.intercept_)
```

```
Model Coefficients:
Coefficients: [ 0.03686206  0.10319047  0.00533106  0.17993235  0.01682659  0.1
7919498
  0.29841084 -0.02525896 -0.08568881 -0.00261956  0.10372634  0.03949923
 -0.03493016 -0.05170662 -0.01670314  0.02001822  0.01542087 -0.02299429
 -0.09335112 -0.01847532 -0.09718638 -0.03063997  0.06334887  0.21566676
  0.01382336 -0.10354977  0.07434479 -0.01558528 -0.0034466   0.00955538
 -0.04232329 -0.1525486  -0.08306056  0.00068997 -0.02962348  0.00955538
 -0.0055237  -0.01610782  0.01670314 -0.01965514 -0.03575729 -0.04362516
 -0.01766087]
Intercept: [-0.00986063]
```

In [13]:
```
plt.scatter(Y_test,y_pred)
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual vs Predicted Prices using SGD Regressor")
plt.plot([min(Y_test),max(Y_test)],[min(Y_test),max(Y_test)],color='red')
plt.show()
```



In [ ]: