In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
```

In [2]:

```
train_df=pd.read_csv("C:\\Users\\USER\\Desktop\\Data Science\\SEM 5\\ML\\Kaggle\\Regression\\train.csv")
test_df=pd.read_csv("C:\\Users\\USER\\Desktop\\Data Science\\SEM 5\\ML\Kaggle\\Regression\\test.csv")
```

In [3]:

```
train_df.shape, test_df.shape
```

Out[3]:

```
((13375, 18), (3344, 16))
```

In [4]:

```
train_df.head()
```

Out[4]:

	id	index	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Gl
0	10717	5596	Darkwatch	PS2	2005.0	Shooter	Ubisoft	0.16	0.12	0.00	0.04	
1	13346	12819	Ken to Mahou to Gakuen Mono. 3D	3DS	2011.0	Role-Playing	Acquire	0.00	0.00	0.06	0.00	
2	1063	6098	FIFA 15	PC	2014.0	Sports	Electronic Arts	0.00	0.27	0.00	0.02	
3	12660	4932	The Darkness II	X360	2012.0	Shooter	Take-Two Interactive	0.24	0.11	0.00	0.03	
4	11264	14483	Suggoi! Arcana Heart 2	PS2	2009.0	Fighting	AQ Interactive	0.00	0.00	0.03	0.00	

In [5]:

```
gs=train_df['Global_Sales'].values
train_df.drop(['Global_Sales'], axis=1, inplace=True)
train_df['Global_Sales'] =gs
train_df.head()
```

Out[5]:

	id	index	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Cr
0	10717	5596	Darkwatch	PS2	2005.0	Shooter	Ubisoft	0.16	0.12	0.00	0.04	
1	13346	12819	Ken to Mahou to Gakuen Mono. 3D	3DS	2011.0	Role-Playing	Acquire	0.00	0.00	0.06	0.00	
2							Electronic					

	id	index	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Cri
2	1063	6098	FIFA 15	PC	2014.0	Sports	Arts	0.00	0.27	0.00	0.02	
3	12660	4932	The Darkness II	X360	2012.0	Shooter	Take-Two Interactive	0.24	0.11	0.00	0.03	
4	11264	14483	Suggoi! Arcana Heart 2	PS2	2009.0	Fighting	AQ Interactive	0.00	0.00	0.03	0.00	

In [6]:

```
train=train_df.copy(deep=True)
train.head()
```

Out[6]:

	id	index	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Cri
0	10717	5596	Darkwatch	PS2	2005.0	Shooter	Ubisoft	0.16	0.12	0.00	0.04	
1	13346	12819	Ken to Mahou to Gakuen Mono. 3D	3DS	2011.0	Role-Playing	Acquire	0.00	0.00	0.06	0.00	
2	1063	6098	FIFA 15	PC	2014.0	Sports	Electronic Arts	0.00	0.27	0.00	0.02	
3	12660	4932	The Darkness II	X360	2012.0	Shooter	Take-Two Interactive	0.24	0.11	0.00	0.03	
4	11264	14483	Suggoi! Arcana Heart 2	PS2	2009.0	Fighting	AQ Interactive	0.00	0.00	0.03	0.00	

In [7]:

```
test=test_df.copy(deep=True)
test.head()
```

Out[7]:

	id	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Critic_Score
0	11	Caesars Palace II	PS	1998.0	Misc	Interplay	0.14	0.09	0.00	0.02	NaN
1	12	WWE '12	Wii	2011.0	Fighting	THQ	0.23	0.08	0.00	0.03	74.0
2	17	Dynasty Warriors 2	PS2	2000.0	Action	THQ	0.24	0.19	0.34	0.06	75.0
3	23	Crysis: Maximum Edition	PC	2009.0	Shooter	Electronic Arts	0.00	0.07	0.00	0.02	NaN
4	27	Pursuit Force	PSP	2005.0	Racing	Sony Computer Entertainment	0.14	0.01	0.00	0.01	75.0

In [8]:

```
train=train.loc[:,['NA_Sales','EU_Sales','JP_Sales','Other_Sales','Global_Sales']]
test=test.loc[:,['id','NA_Sales','EU_Sales','JP_Sales','Other_Sales']]
```

In [9]:

```
train.head()
```

```
train.head()
```

Out[9]:

	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	0.16	0.12	0.00	0.04	0.32
1	0.00	0.00	0.06	0.00	0.06
2	0.00	0.27	0.00	0.02	0.28
3	0.24	0.11	0.00	0.03	0.39
4	0.00	0.00	0.03	0.00	0.03

In [10]:

```
train=train.sample(frac=1, random_state=42).reset_index(drop=True)
```

In [11]:

```
len(train) *0.8
```

Out[11]:

```
10700.0
```

In [12]:

```
val_set=train.iloc[10700:, :].reset_index(drop=True)
```

In [13]:

```
train=train.iloc[:10700:, :]
```

In [14]:

```
train.shape, val_set.shape
```

Out[14]:

```
((10700, 5), (2675, 5))
```

In [15]:

```
X_train=train.iloc[:,:-1].values
X_val=val_set.iloc[:,:-1].values
y_train=train.iloc[:,-1].values
y_val=val_set.iloc[:,-1].values
```

In [16]:

```
from sklearn.linear_model import SGDRegressor
```

In [17]:

```
sgd_obj=SGDRegressor(loss='epsilon_insensitive',epsilon=0,
                     early_stopping=False,alpha=0.01,learning_rate='adaptive',tol=1e-3)
```

In [18]:

```
sgd_obj.fit(X_train, y_train)
sgd_obj.coef_
sgd_obj.intercept_
```

Out[18]:

```
array([-3.27865278e-18])
```

In [19]:

```
preds_train=sgd_obj.predict(X_train)
```

```
preds_val=sgd_obj.predict(X_val)
```

In [20]:

```
from sklearn.metrics import mean_absolute_error
```

In [21]:

```
mean_absolute_error(y_true=y_train, y_pred=preds_train)
```

Out[21]:

```
0.0027341280336097987
```

In [22]:

```
mean_absolute_error(y_true=y_val, y_pred=preds_val)
```

Out[22]:

```
0.002639712633661972
```

In [23]:

```
X_test=test.iloc[:, 1:].values
preds_test=sgd_obj.predict(X_test)
```

In [24]:

```
sub_df=pd.DataFrame(index=range(len(test)))
sub_df['id'] =test['id'].values
sub_df['Global_Sales'] =preds_test
```

In [25]:

```
sub_df.to_csv('InClass_ML_Comp1_submission.csv', index=False)
```

In [ ]:

```

```