Feature Engineering:

In [1]:
```
import random
import pandas as pd
```

In [81]:
```
[w1,w2,w3,w4,w5]=[0.474,0.456,0.429,0.475,0.471]
val=[]
for i in range(1000):
    x1=random.randint(1,100);
    x2=random.randint(1,4);
    x3=random.randint(1,3);
    x4=random.randint(1,5);
    x5=random.randint(1,100);
    eq=w1*+w2*x2+w3*x3+w4*x4+w5*x5;
    val.append([x1,x2,x3,x4,x5,eq])
    df=pd.DataFrame(val,columns=['Author Popularity','Languages','Price','Rating','Best Selling','Buyer Rating'])
    df.to_csv('Dataset of Books.csv',index=False)
    df
```

In [23]:
```
import matplotlib.pyplot as plt
```

In [87]:
```
from sklearn.linear_model import LinearRegression
X_train, X_test, Y_train, Y_test = train_test_split(df[['Best Selling', 'Buyer Rating']], df['Buyer Rating'], test_size=0.25)
mlr=LinearRegression()

mlr.fit(X_train,Y_train)

pred_mlr = mlr.predict(X_test)
pred_mlr
```

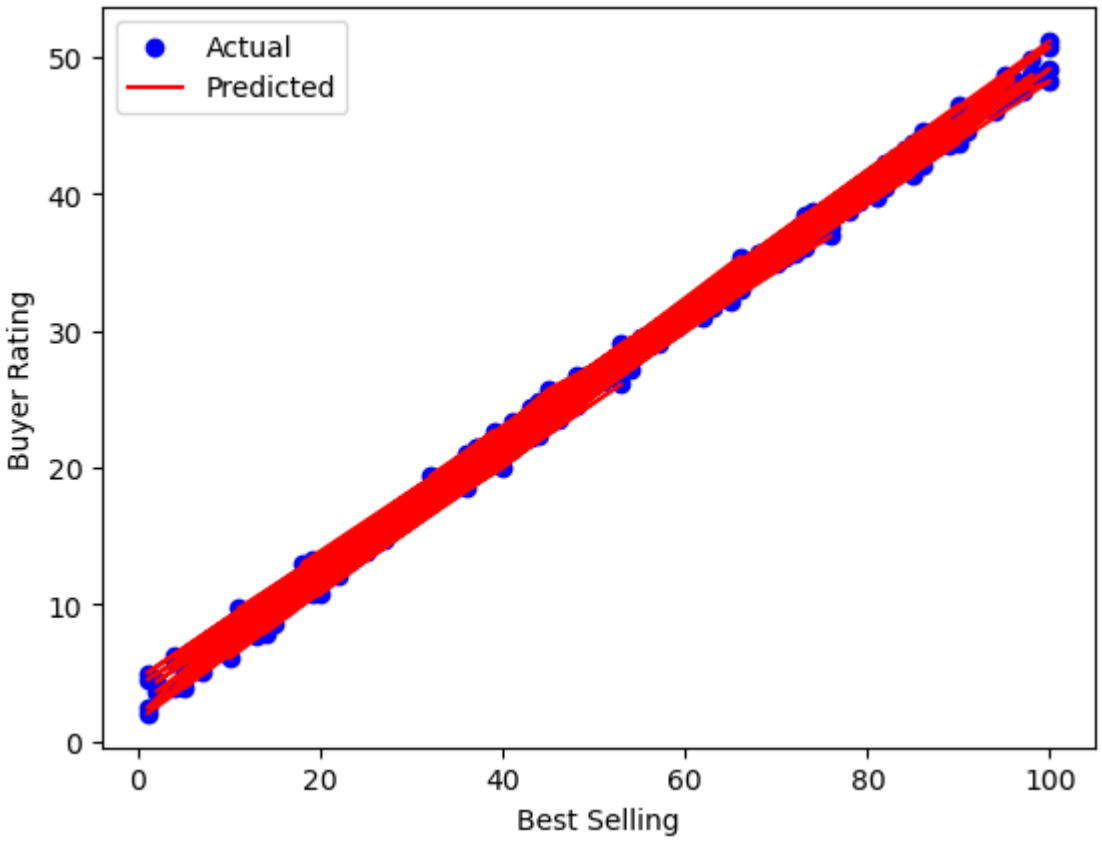
Out[87]:
```
array([14.888576, 46.229432, 47.684432,  3.695288, 27.682288, 34.322288,
        9.691576, 27.636288, 22.786144,  4.602576, 36.550576, 18.687288,
       44.889144, 41.337288,  8.580144, 47.931288, 30.326144,  6.449288,
       19.629288, 25.141144, 45.793144, 18.992144,  6.398576, 10.166576,
       37.109576, 19.421144,  7.471288, 38.048288,  4.645288, 39.874288,
       17.783288, 32.858576, 29.851144, 37.140288,  5.830144, 39.962288,
       18.050144, 33.325576, 13.382144, 49.047432, 41.761576,  4.645288,
        9.270576, 49.178144, 45.503576, 48.842576,  3.699288,  8.575432,
       47.761144, 46.437576, 19.200288, 46.997288, 10.204576, 22.798144,
       42.496144, 35.167576, 22.790144, 10.471432, 14.185432, 35.646576,
       26.083144, 27.685576, 43.950432, 45.355432, 48.147432, 17.015432,
        8.791576, 34.017432, 49.094144, 25.586144, 23.659432, 12.343432,
       18.979432, 43.271288, 36.206288, 43.646288, 25.573432, 42.287288,
       49.090144,  9.517432, 46.039288, 25.752288, 10.178576, 38.773432,
       23.273144, 13.471576, 45.610288, 37.009576,  5.719432, 12.827144,
       14.783144, 29.041288, 49.310288, 14.455576, 19.450432, 35.944144,
       12.351432, 26.953144, 33.330288, 26.694288, 10.042432, 31.183432,
       10.433432,  9.347288, 33.550432, 43.658288, 20.366432, 11.239288,
       46.336144, 14.227432, 37.526576, 23.265144, 32.354288, 37.564576,
       23.698144, 18.212288, 15.262144,  8.849576, 19.933432, 26.987144,
        8.841576, 13.336144, 46.001288,  9.020432, 11.911144, 31.844576,
       19.968144, 34.183576, 27.389432, 27.685576, 43.916432, 48.827288,
       31.704432, 11.660288, 32.126144, 18.292288, 34.199576, 41.638144,
       25.844288, 46.016576, 25.327288, 45.402144, 19.018144, 37.021576,
       34.670576, 15.906576, 33.793288, 35.419144, 15.440288, 40.437288,
       30.249432, 48.406288, 24.168432, 27.890432,  4.387144, 22.802144,
       18.980144, 38.218432, 40.090432, 30.728432, 35.855432, 25.581432,
       22.710144, 28.358144, 48.151432, 19.424432, 28.103288, 14.923288,
       39.874288, 48.397576, 31.920576, 36.338432, 49.051432, 11.452144,
       42.967144, 16.150144, 28.450144, 17.582432, 31.704432, 27.218576,
        9.259288, 25.273288, 25.493432, 21.003576,  7.684144, 26.686288,
       36.457144, 40.348576,  4.849432, 27.172576, 18.254288,  8.583432,
       20.404432, 46.993288, 45.572288, 20.791432, 24.547432,  3.919432,
       16.760576,  8.112432,  8.113144, 37.306432, 12.602288, 49.822576,
       21.482576, 18.895432,  9.050432, 27.470144, 37.488576,  6.445288,
        6.483288, 46.951288, 18.656576, 29.485576, 35.639288, 43.619576,
       26.516144, 25.501432, 13.594288, 39.885576, 26.303288, 14.748432,
       11.914432, 45.838432, 20.347144, 26.562144, 13.814432, 23.659432,
       41.630144, 39.503288, 46.472288, 29.392144, 44.163288, 22.960288,
       37.997576,  8.800288,  4.082288, 14.972576, 14.953288,  5.494576,
       38.503288, 41.291288, 16.540432, 34.060144, 37.739432, 24.130432,
       36.163576, 40.569432, 34.666576,  8.804288])
```

In [93]:
```
from sklearn.metrics import r2_score

r2_score = r2_score(Y_test, pred_mlr)

print("R2 score:", r2_score)
```

R2 score: 1.0

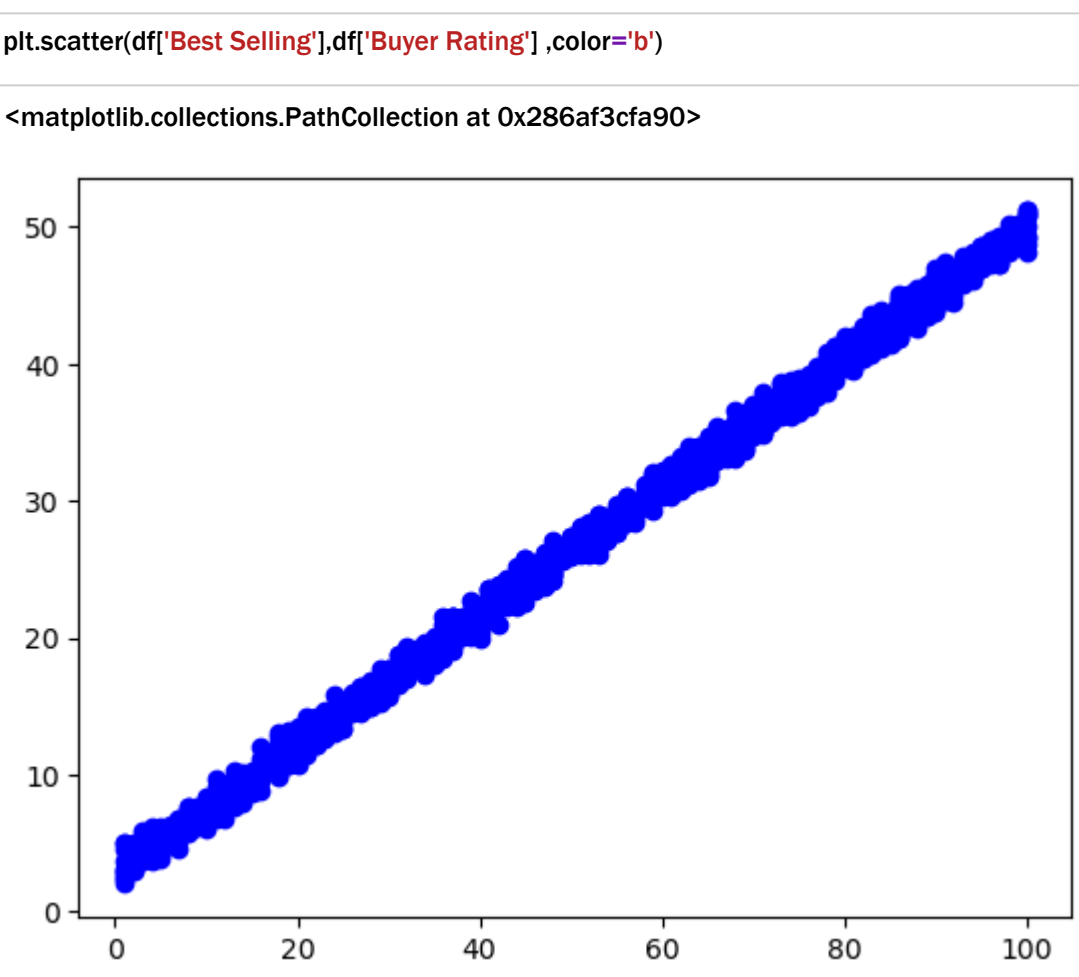
The most common metric used to evaluate the performance of a regression model is the R-squared score. The R-squared score is a measure of how well the model explains the variation in the target variable. It is a value between 0 and 1, with 1 being a perfect fit and 0 being no fit at all. A higher R-squared score indicates that the model is better able to predict the target variable.

In [72]:
```
import matplotlib.pyplot as plt

plt.scatter(X_test["Best Selling"], Y_test, color='b', label='Actual')
plt.plot(X_test["Best Selling"], pred_mlr, color='r', label='Predicted')
plt.xlabel("Best Selling")
plt.ylabel("Buyer Rating")
plt.legend()
plt.show()
```



In [75]:
```
plt.scatter(df['Best Selling'],df['Buyer Rating'] ,color='b')
```

Out[75]: <matplotlib.collections.PathCollection at 0x286af3cfa90>



In [ ]: