Linear Regression Using income dataset

In [14]:

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [5]:

```python
df = pd.read_csv('/content/income.csv')
```

In [6]:

```python
df.head()
```

Out[6]:

	ID	Income	Age	Education	Gender
0	1	113	69	12	1
1	2	91	52	18	0
2	3	121	65	14	0
3	4	81	58	12	0
4	5	68	31	16	1

In [7]:

```python
df.isnull().sum()
```

Out[7]:

```
ID           0
Income       0
Age          0
Education    0
Gender       0
dtype: int64
```

In [8]:

```python
df.describe()
```

Out[8]:

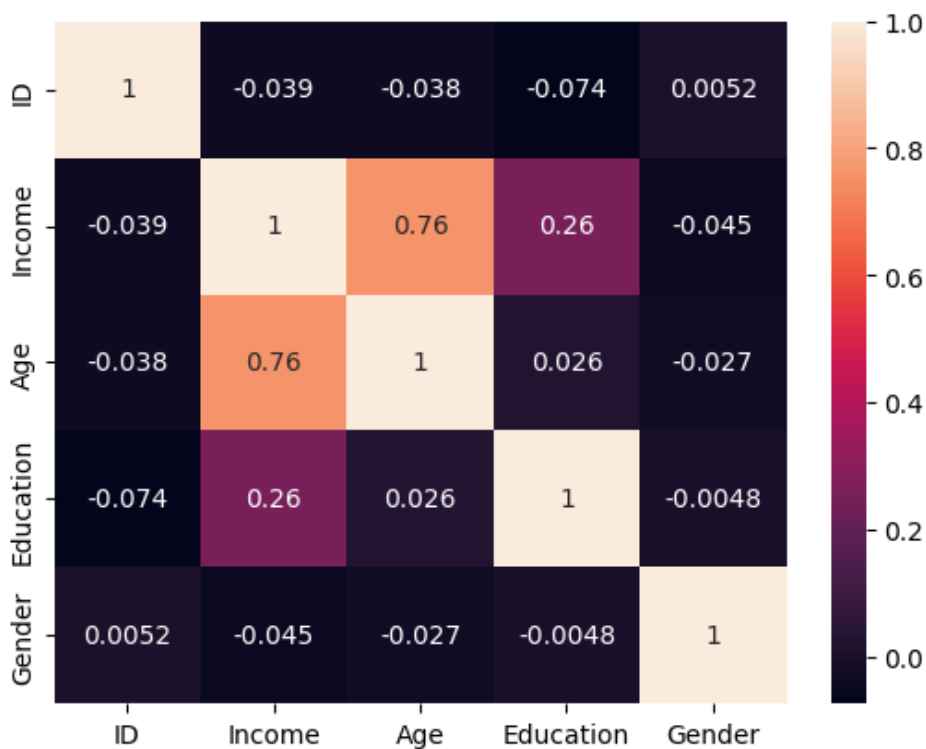|	ID	Income	Age	Education	Gender
count	1500.000000	1500.000000	1500.000000	1500.000000	1500.000000
mean	750.500000	75.986000	43.582000	14.681333	0.490000
std	433.157015	20.005215	15.169466	2.693812	0.500067
min	1.000000	14.000000	18.000000	10.000000	0.000000
25%	375.750000	62.000000	30.000000	12.000000	0.000000
50%	750.500000	76.000000	44.000000	15.000000	0.000000
75%	1125.250000	91.000000	57.000000	16.000000	1.000000
max	1500.000000	134.000000	70.000000	20.000000	1.000000

In [9]:

```
df.shape
```

Out[9]:

```
(1500, 5)
```

In [20]:

```
df.corr()
```

Out[20]:

	ID	Income	Age	Education	Gender
ID	1.000000	-0.038846	-0.037770	-0.074147	0.005246
Income	-0.038846	1.000000	0.761486	0.256634	-0.045060
Age	-0.037770	0.761486	1.000000	0.026254	-0.027242
Education	-0.074147	0.256634	0.026254	1.000000	-0.004843
Gender	0.005246	-0.045060	-0.027242	-0.004843	1.000000

In [19]:

```
sns.heatmap(df.corr(), annot=True)
plt.show()
```


In [21]:

```
X = df.drop(['ID','Income'],axis=1)
y = df['Income']
```

In [23]:

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

In [26]:

```
print(X_train.shape,X_test.shape)
```

```
(1200, 3) (300, 3)
```

In [27]:

```
print(X_train.head())
```

```
      Age  Education  Gender
1305   67         16       1
1130   33         12       1
699    64         18       0
94     25         12       1
1297   31         16       1
```

In [28]:

```
lr = LinearRegression()
```

In [29]:

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

Out[29]:

▼ LinearRegression
LinearRegression()

In [30]:

```
lr.predict(X_test)
```

Out[30]:

```
array([ 58.75408219, 105.40315189,  97.84353749,  61.57872602,
        45.88133961,  57.91514836,  93.28765052,  96.84752782,
        66.64046702,  81.65599544,  47.12873648,  57.83345576,
        99.50878642,  95.5247477 ,  69.95526649,  87.39328506,
        47.04704387,  49.9533803 ,  57.50668532,  65.63814797,
        82.90339231,  73.68791801,  74.93531488,  93.53272835,
        80.49660055,  62.97688937,  88.96114299,  67.6364767 ,
        71.36912822,  51.03108259,  84.65033385,  53.02310194,
        80.82337098,  76.75763965,  85.89142135,  56.76206284,
        85.80972874,  58.00315034,  98.26769892,  78.42288858,
        82.73369773,  55.92943837,  68.46279179,  87.71374612,
        67.79986191,  63.06489134,  91.54070899,  98.01631173,
        51.85739768,  65.71984058,  58.08484295, 103.07805273,
        94.93720107,  51.03108259,  67.46678211,  72.77360094,
        53.93110965,  84.81371906,  97.59845967,  61.66041862,
        66.39538919,  93.45103574,  57.34330011,  62.48673372,
        71.61420604,  97.42876509,  66.96723745,  68.79587159,
        96.3510628 ,  80.57829315,  88.38298537,  54.51865629,
        64.64844766,  84.81371906, 102.32712087,  94.76750648,
        83.57263156,  58.66608022,  53.02310194,  80.41490794,
        72.94329552,  93.19964855,  64.97521809,  84.39894665,
        57.50668532,  82.4069273 ,  58.25761718,  54.84542672,
        84.56864124,  53.69234119,  50.61631018,  75.93132456,
        67.04893006,  78.59258316,  87.55667027,  97.34707248,
        64.80552351,  94.2836602 ,  68.63248638,  95.5247477 ,
        92.69379452,  95.19797727,  87.63836288,  72.94329552,
        60.57640697,  79.33720565,  63.4827434 ,  86.79942905,
        98.34308216,  65.63814797,  70.95127616,  75.76162998,
        93.85949878, 100.5047961 ,  70.1186517 ,  91.94917203,
        79.17382044,  67.79986191,  99.50878642,  73.44284019,
        70.46112052,  66.05292038,  95.68813292,  66.39538919,
        73.36114758,  48.04305355,  70.12804072,  80.74798774,
        85.72803613,  91.21393856,  56.01113098, 104.23744762,
        86.71773644,  86.39727538,  75.27147433,  60.82148479,
        65.0569107 ,  74.52054247,  89.30361181,  72.61652509,
        82.4069273 ,  96.52075738,  65.0569107 ,  98.17969694,
        85.56465092,  56.51067565,  84.06278721,  83.07616654,
        48.04305355, 100.25971828,  86.39096601,  80.74798774,
        78.9193536 ,  82.57662188,  82.57031251,  89.63038224,
        79.34351502, 100.09002369,  77.4268789 ,  89.2219192 ,
        77.50857151,  90.54469931,  78.42288858,  60.98487001,
       104.31914023, 107.39517124,  54.10080423,  87.30528308,
```

```
          90.21792888,  63.73413059,  47.87335897,  71.205743  ,
          91.86747942,  89.79376746,  72.86160291,  69.54049408,
          90.54469931,  72.44683051,  87.72005549,  79.26182241,
          81.16583979,  98.51277674,  84.80740969,  97.59845967,
          95.68813292,  73.61253477,  59.6620899 ,  78.26581273,
          60.50102373,  51.85739768,  55.75974379,  95.68813292,
          60.82148479,  58.502695  ,  67.39139887,  75.76162998,
          57.00714066,  96.59614062,  54.51865629, 105.98438916,
          87.38697569, 102.25173763,  84.89541167,  59.75009187,
          87.79543873,  60.49471436,  77.4268789 ,  94.52873803,
          85.56465092, 100.5047961 ,  88.38298537,  84.89541167,
          75.43485954,  84.73202645,  57.00714066,  93.53272835,
          57.99684097,  61.74211123,  46.05103419,  54.68835087,
          51.77570507, 100.33510152,  60.33132914,  88.87945038,
          93.69611356,  71.4571302 ,  87.71374612,  62.97688937,
          64.47875308,  47.95505157,  69.70387929,  93.28765052,
          67.79355254,  80.41490794,  88.38929474,  54.51865629,
          59.25362686,  64.47875308,  87.38697569,  46.04472482,
         101.50080578,  81.74399742,  89.54238026,  60.82148479,
          84.89541167,  55.51466597,  94.68581387,  58.502695  ,
          62.81981352,  97.51045769,  98.51277674,  97.34707248,
          53.02310194,  50.11676552,  72.94329552,  89.30361181,
          84.72571709,  45.88133961,  80.91137295,  78.74965901,
          83.57263156,  68.95925681,  68.78956222,  89.38530442,
          71.29374498,  88.04682592,  53.84941704,  63.97920841,
          71.37543759,  61.49703341,  70.45481115,  98.43108413,
          64.80552351,  66.64046702,  67.55478409,  78.59258316,
          84.89541167,  95.35505312,  46.1327268 ,  87.55036091,
          84.23556144,  94.85550846,  56.68037023,  77.4268789 ,
          87.55667027,  81.65599544,  75.67993737,  89.46699702,
          92.28533148,  60.82148479,  82.90339231,  91.37101441,
          92.5304093 ,  69.95526649, 103.9923698 ,  67.63016733,
          74.52054247,  54.01911162,  87.88344071,  72.77360094])
```

In [32]:

```python
input = [50,12,0]

input_asnumpy = np.asarray(input)

input_reshape = input_asnumpy.reshape(1,-1)

prediction = lr.predict(input_reshape)

print(prediction)
```

```
[77.92334392]
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have
valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

In [ ]: