In [29]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
```

In [2]:

```python
data = pd.read_csv('D:\Code\Python\Dataset\income.csv')
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1500 entries, 0 to 1499
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   ID         1500 non-null   int64
 1   Income     1500 non-null   int64
 2   Age        1500 non-null   int64
 3   Education  1500 non-null   int64
 4   Gender     1500 non-null   int64
dtypes: int64(5)
memory usage: 58.7 KB
```

In [3]:

```python
data.head()
```

Out[3]:

	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

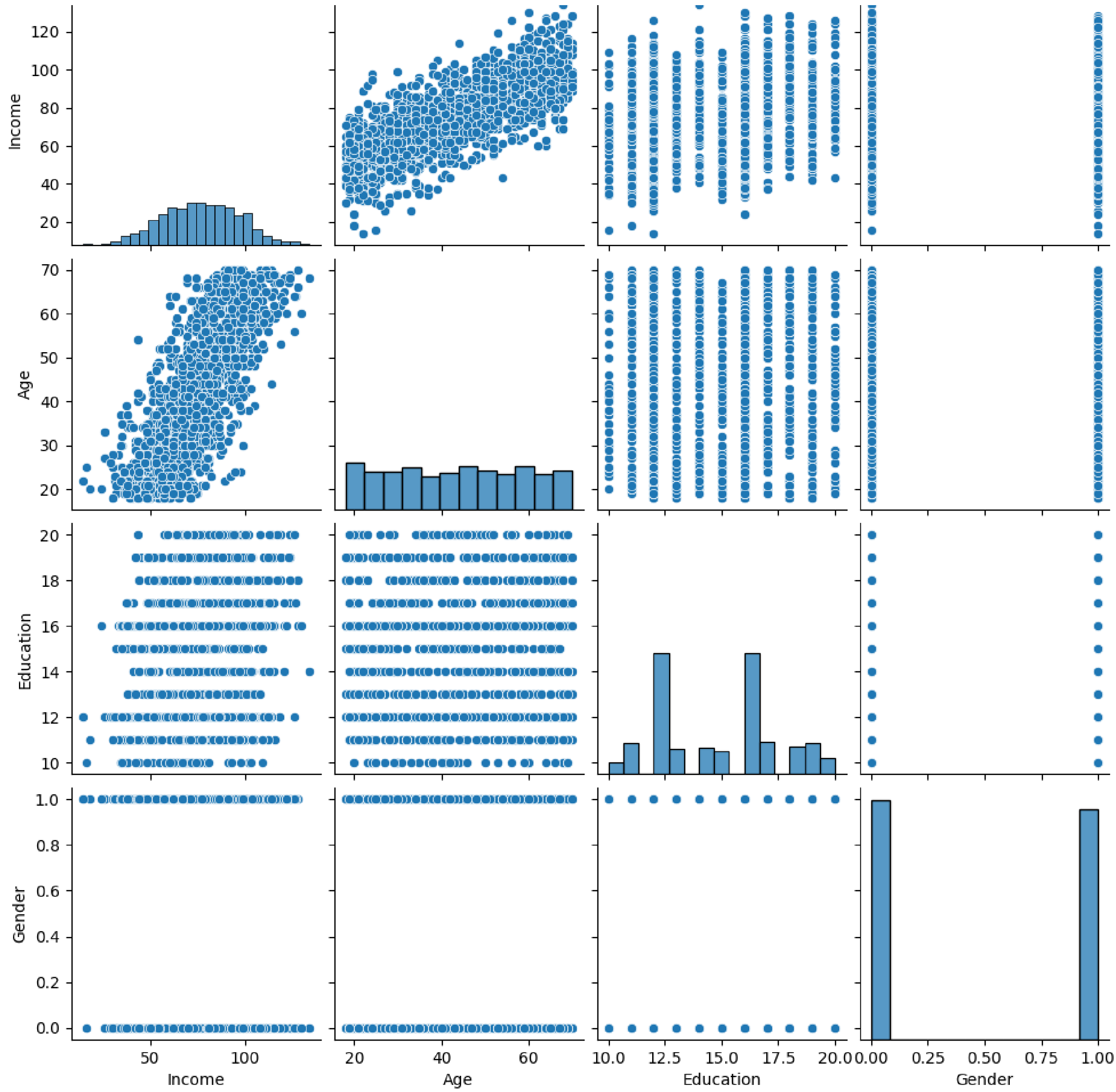
In [4]:

```python
data.drop_duplicates(inplace=True)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1500 entries, 0 to 1499
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   ID         1500 non-null   int64
 1   Income     1500 non-null   int64
 2   Age        1500 non-null   int64
 3   Education  1500 non-null   int64
 4   Gender     1500 non-null   int64
dtypes: int64(5)
memory usage: 70.3 KB
```

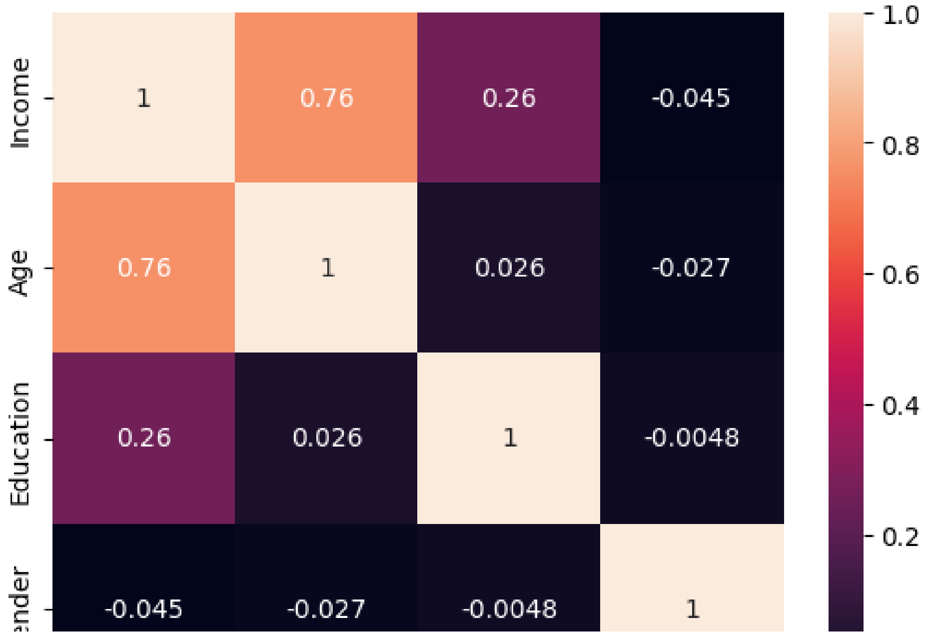
In [11]:

```python
sns.pairplot(data[['Income', 'Age', 'Education', 'Gender']])
plt.show()
```



In [14]:

```
sns.heatmap(data[['Income', 'Age', 'Education', 'Gender']].corr(), annot=True)
plt.show()
```



In [22]:

```
X = data['Age'].values.reshape(-1, 1)
y = data['Income']
```

In [23]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42
)
```

In [24]:

```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[24]:

```
LinearRegression()
```

In [25]:

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

In [26]:

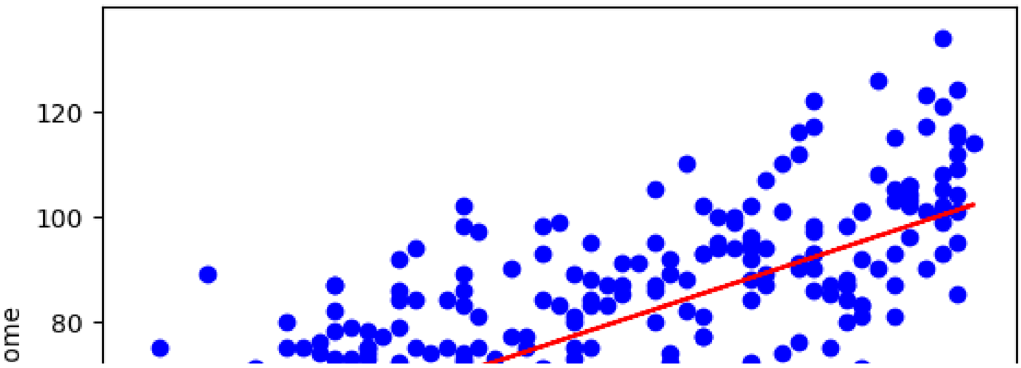
```
pred_vs_actual = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred, 'Difference': y_te
st - y_pred})
pred_vs_actual.describe()
```

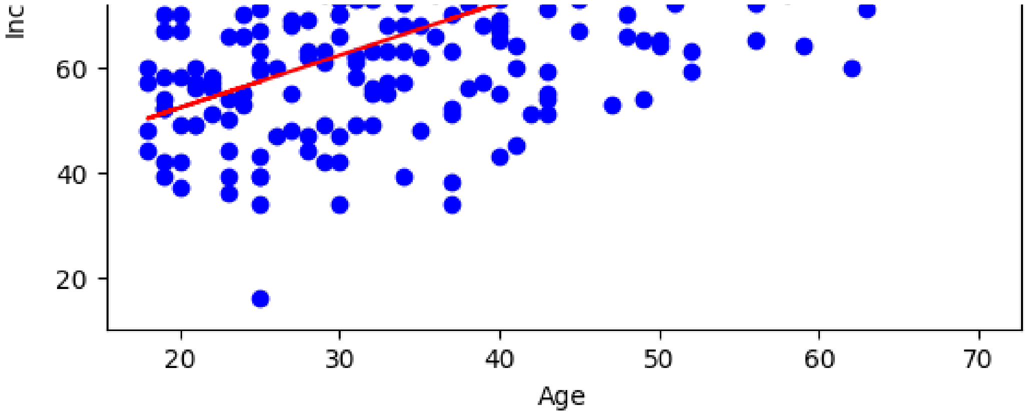
Out[26]:

	Actual	Predicted	Difference
count	300.000000	300.000000	300.000000
mean	76.076667	75.119477	0.957190
std	20.991417	15.437326	13.799281
min	16.000000	50.221960	-41.219389
25%	60.000000	62.217552	-8.210939
50%	75.000000	73.213511	2.287408
75%	91.250000	88.457908	9.782908
max	134.000000	102.202857	34.779509

In [27]:

```
plt.scatter(X_test, y_test, color='blue')
plt.plot(X_test, y_pred, color='red')
plt.xlabel('Age')
plt.ylabel('Income')
plt.show()
```





In [35]:

```
print('R2 score: %.2f' % r2_score(y_test, y_pred))
print('MAE\t: %.2f' % mean_absolute_error(y_test, y_pred))
print('MSE\t: %.2f' % mean_squared_error(y_test, y_pred))
print('RMSE\t: %.2f' % np.sqrt(mean_squared_error(y_test, y_pred)))
```

```
R2 score: 0.57
MAE : 11.08
MSE : 190.70
RMSE : 13.81
```