In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
data = pd.read_csv('C:/Users/prajw/Downloads/SPPU-TE-COMP-SEM-2-2019-PATTERN-DSBDA-main/SPPU-TE-COMP-SEM-2-2019-PATTERN-DSBDA-m
```

In [3]:
```
data
```

Out[3]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

In [4]:
```
data.head(5)
```

Out[4]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

In [5]:
```
data.tail()
```

Out[5]:

	User ID	Gender	Age	EstimatedSalary	Purchased
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

In [6]:
```
data.shape
```

Out[6]:
```
(400, 5)
```

In [7]:
```
data.columns
```

Out[7]:
```
Index(['User ID', 'Gender', 'Age', 'EstimatedSalary', 'Purchased'], dtype='object')
```

In [8]:
```
data.describe()
```

Out[8]:

	User ID	Age	EstimatedSalary	Purchased
count	4.000000e+02	400.000000	400.000000	400.000000
mean	1.569154e+07	37.655000	69742.500000	0.357500
std	7.165832e+04	10.482877	34096.960282	0.479864
min	1.556669e+07	18.000000	15000.000000	0.000000
25%	1.562676e+07	29.750000	43000.000000	0.000000
50%	1.569434e+07	37.000000	70000.000000	0.000000
75%	1.575036e+07	46.000000	88000.000000	1.000000
max	1.581524e+07	60.000000	150000.000000	1.000000

In [9]:

```
data.isnull().sum()
```

Out[9]:

```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

In [10]:

```
data.iloc[:,2:4]
```

Out[10]:

	Age	EstimatedSalary
0	19	19000
1	35	20000
2	26	43000
3	27	57000
4	19	76000
...	...	...
395	46	41000
396	51	23000
397	50	20000
398	36	33000
399	49	36000

400 rows × 2 columns

In [11]:

```
data.iloc[:,2:4].values
```

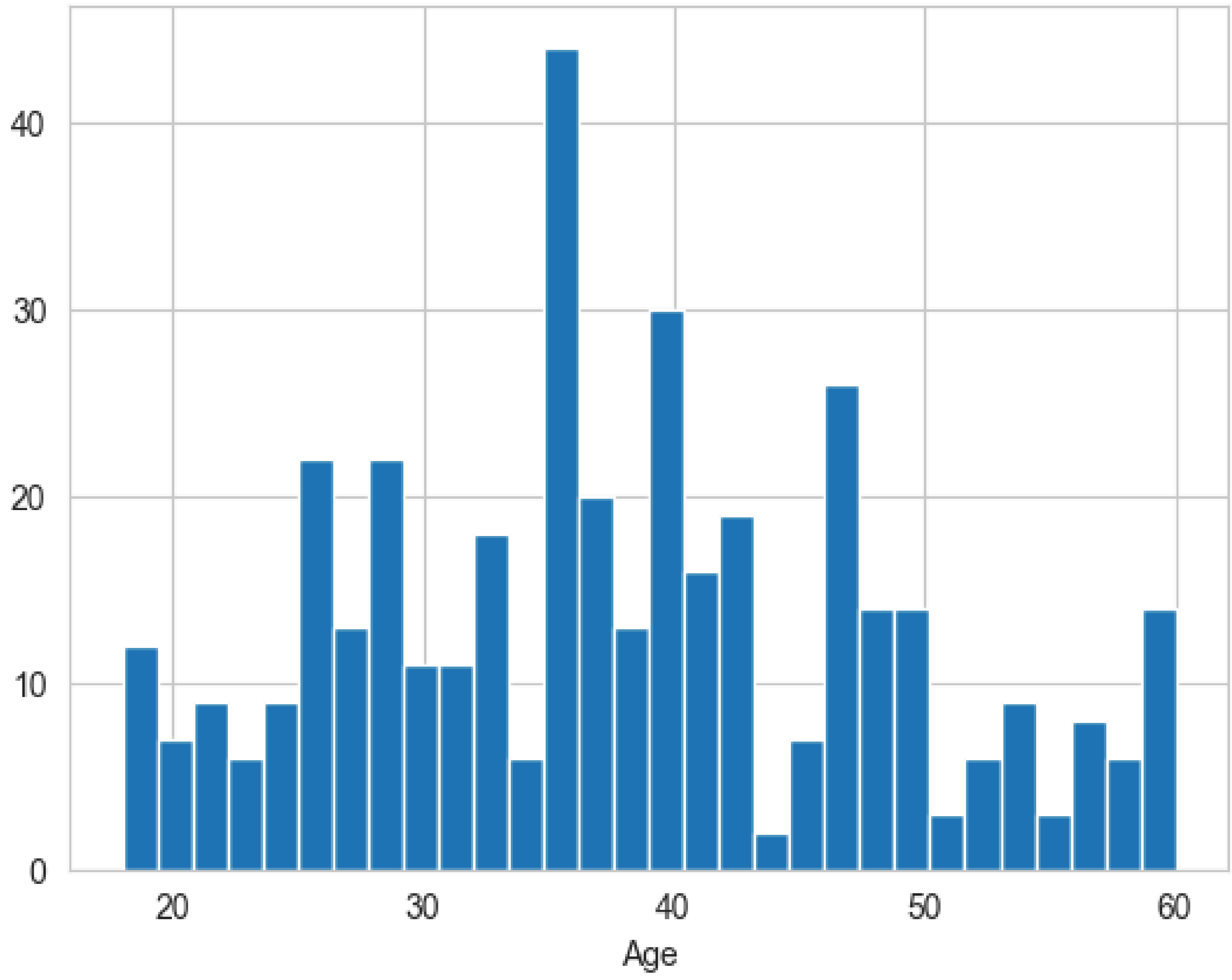
```
Out[11]:  array([[    19,  19000],
                 [    35,  20000],
                 [    26,  43000],
                 [    27,  57000],
                 [    19,  76000],
                 [    27,  58000],
                 [    27,  84000],
                 [    32, 150000],
                 [    25,  33000],
                 [    35,  65000],
                 [    26,  80000],
                 [    26,  52000],
                 [    20,  86000],
                 [    32,  18000],
                 [    18,  82000],
                 [    29,  80000],
                 [    47,  25000],
                 [    45,  26000],
                 [    46,  28000],
                 [    48,  29000],
                 [    45,  22000],
                 [    47,  49000],
                 [    48,  41000],
                 [    45,  22000],
                 [    46,  23000],
                 [    47,  20000],
                 [    49,  28000],
                 [    47,  30000],
                 [    29,  43000],
                 [    31,  18000],
                 [    31,  74000],
                 [    27, 137000],
                 [    21,  16000],
                 [    28,  44000],
                 [    27,  90000],
                 [    35,  27000],
                 [    33,  28000],
                 [    30,  49000],
                 [    26,  72000],
                 [    27,  31000],
                 [    27,  17000],
                 [    33,  51000],
                 [    35, 108000],
                 [    30,  15000],
                 [    28,  84000],
                 [    23,  20000],
                 [    25,  79000],
                 [    27,  54000],
                 [    30, 135000],
                 [    31,  89000],
                 [    24,  32000],
                 [    18,  44000],
                 [    29,  83000],
                 [    35,  23000],
                 [    27,  58000],
                 [    24,  55000],
                 [    23,  48000],
                 [    28,  79000],
                 [    22,  18000],
                 [    32, 117000],
                 [    27,  20000],
                 [    25,  87000],
                 [    23,  66000],
                 [    32, 120000],
                 [    59,  83000],
                 [    24,  58000],
                 [    24,  19000],
                 [    23,  82000],
                 [    22,  63000],
                 [    31,  68000],
                 [    25,  80000],
                 [    24,  27000],
                 [    20,  23000],
                 [    33, 113000],
                 [    32,  18000],
                 [    34, 112000],
                 [    18,  52000],
                 [    22,  27000],
                 [    28,  87000],
                 [    26,  17000],
                 [    30,  80000],
                 [    39,  42000],
                 [    20,  49000],
                 [    35,  88000],
                 [    30,  62000],
                 [    31, 118000],
                 [    24,  55000],
[    28,  85000],
[    26,  81000],
[    35,  50000],
[    22,  81000],
[    30, 116000],
[    26,  15000],
[    29,  28000],
[    29,  83000],
[    35,  44000],
[    35,  25000],
[    28, 123000],
[    35,  73000],
[    28,  37000],
[    27,  88000],
[    28,  59000],
[    32,  86000],
[    33, 149000],
[    19,  21000],
[    21,  72000],
[    26,  35000],
[    27,  89000],
[    26,  86000],
[    38,  80000],
[    39,  71000],
[    37,  71000],
[    38,  61000],
[    37,  55000],
[    42,  80000],
[    40,  57000],
[    35,  75000],
[    36,  52000],
[    40,  59000],
[    41,  59000],
[    36,  75000],
[    37,  72000],
[    40,  75000],
[    35,  53000],
[    41,  51000],
[    39,  61000],
[    42,  65000],
[    26,  32000],
[    30,  17000],
[    26,  84000],
[    31,  58000],
[    33,  31000],
[    30,  87000],
[    21,  68000],
[    28,  55000],
[    23,  63000],
[    20,  82000],
[    30, 107000],
[    28,  59000],
[    19,  25000],
[    19,  85000],
[    18,  68000],
[    35,  59000],
[    30,  89000],
[    34,  25000],
[    24,  89000],
[    27,  96000],
[    41,  30000],
[    29,  61000],
[    20,  74000],
[    26,  15000],
[    41,  45000],
[    31,  76000],
[    36,  50000],
[    40,  47000],
[    31,  15000],
[    46,  59000],
[    29,  75000],
[    26,  30000],
[    32, 135000],
[    32, 100000],
[    25,  90000],
[    37,  33000],
[    35,  38000],
[    33,  69000],
[    18,  86000],
[    22,  55000],
[    35,  71000],
[    29, 148000],
[    29,  47000],
[    21,  88000],
[    34, 115000],
[    26, 118000],
[    34,  43000],
[    34,  72000],
[    23,  28000],
[    35,  47000],
[    25,  22000],
[    24,  23000],
[    31,  34000],
[    26,  16000],
[    31,  71000],
[    32, 117000],
[    33,  43000],
[    33,  60000],
[    31,  66000],
[    20,  82000],
[    33,  41000],
[    35,  72000],
[    28,  32000],
[    24,  84000],
[    19,  26000],
[    29,  43000],
[    19,  70000],
[    28,  89000],
[    34,  43000],
[    30,  79000],
[    20,  36000],
[    26,  80000],
[    35,  22000],
[    35,  39000],
[    49,  74000],
[    39, 134000],
[    41,  71000],
[    58, 101000],
[    47,  47000],
[    55, 130000],
[    52, 114000],
[    40, 142000],
[    46,  22000],
[    48,  96000],
[    52, 150000],
[    59,  42000],
[    35,  58000],
[    47,  43000],
[    60, 108000],
[    49,  65000],
[    40,  78000],
[    46,  96000],
[    59, 143000],
[    41,  80000],
[    35,  91000],
[    37, 144000],
[    60, 102000],
[    35,  60000],
[    37,  53000],
[    36, 126000],
[    56, 133000],
[    40,  72000],
[    42,  80000],
[    35, 147000],
[    39,  42000],
[    40, 107000],
[    49,  86000],
[    38, 112000],
[    46,  79000],
[    40,  57000],
[    37,  80000],
[    46,  82000],
[    53, 143000],
[    42, 149000],
[    38,  59000],
[    50,  88000],
[    56, 104000],
[    41,  72000],
[    51, 146000],
[    35,  50000],
[    57, 122000],
[    41,  52000],
[    35,  97000],
[    44,  39000],
[    37,  52000],
[    48, 134000],
[    37, 146000],
[    50,  44000],
[    52,  90000],
[    41,  72000],
[    40,  57000],
[    58,  95000],
[    45, 131000],
[    35,  77000],
[    36, 144000],
[    55, 125000],
[    35,  72000],
[    48,  90000],
[    42, 108000],
[    40,  75000],
[    37,  74000],
[    47, 144000],
[    40,  61000],
[    43, 133000],
[    59,  76000],
[    60,  42000],
[    39, 106000],
[    57,  26000],
[    57,  74000],
[    38,  71000],
[    49,  88000],
[    52,  38000],
[    50,  36000],
[    59,  88000],
[    35,  61000],
[    37,  70000],
[    52,  21000],
[    48, 141000],
[    37,  93000],
[    37,  62000],
[    48, 138000],
[    41,  79000],
[    37,  78000],
[    39, 134000],
[    49,  89000],
[    55,  39000],
[    37,  77000],
[    35,  57000],
[    36,  63000],
[    42,  73000],
[    43, 112000],
[    45,  79000],
[    46, 117000],
[    58,  38000],
[    48,  74000],
[    37, 137000],
[    37,  79000],
[    40,  60000],
[    42,  54000],
[    51, 134000],
[    47, 113000],
[    36, 125000],
[    38,  50000],
[    42,  70000],
[    39,  96000],
[    38,  50000],
[    49, 141000],
[    39,  79000],
[    39,  75000],
[    54, 104000],
[    35,  55000],
[    45,  32000],
[    36,  60000],
[    52, 138000],
[    53,  82000],
[    41,  52000],
[    48,  30000],
[    48, 131000],
[    41,  60000],
[    41,  72000],
[    42,  75000],
[    36, 118000],
[    47, 107000],
[    38,  51000],
[    48, 119000],
[    42,  65000],
[    40,  65000],
[    57,  60000],
[    36,  54000],
[    58, 144000],
[    35,  79000],
[    38,  55000],
[    39, 122000],
[    53, 104000],
[    35,  75000],
[    38,  65000],
[    47,  51000],
[    47, 105000],
[    41,  63000],
[    53,  72000],
[    54, 108000],
[    39,  77000],
[    38,  61000],
[    38, 113000],
[    37,  75000],
[    42,  90000],
[    37,  57000],
[    36,  99000],
[    60,  34000],
[    54,  70000],
[    41,  72000],
[    40,  71000],
[    42,  54000],
[    43, 129000],
[    53,  34000],
[    47,  50000],
[    42,  79000],
[    42, 104000],
[    59,  29000],
[    58,  47000],
[    46,  88000],
[    38,  71000],
[    54,  26000],
[    60,  46000],
[    60,  83000],
[    39,  73000],
[    59, 130000],
[    37,  80000],
[    46,  32000],
[    46,  74000],
[    42,  53000],
[    41,  87000],
[    58,  23000],
[    42,  64000],
[    48,  33000],
[    44, 139000],
[    49,  28000],
[    57,  33000],
[    56,  60000],
[    49,  39000],
[    39,  71000],
[    47,  34000],
[    48,  35000],
[    48,  33000],
[    47,  23000],
[    45,  45000],
[    60,  42000],
[    39,  59000],
[    46,  41000],
[    51,  23000],
[    50,  20000],
[    36,  33000],
[    49,  36000]])
```

In [12]:

```
sns.set_style('whitegrid')
data['Age'].hist(bins=30)
plt.xlabel('Age')
```

Out[12]:

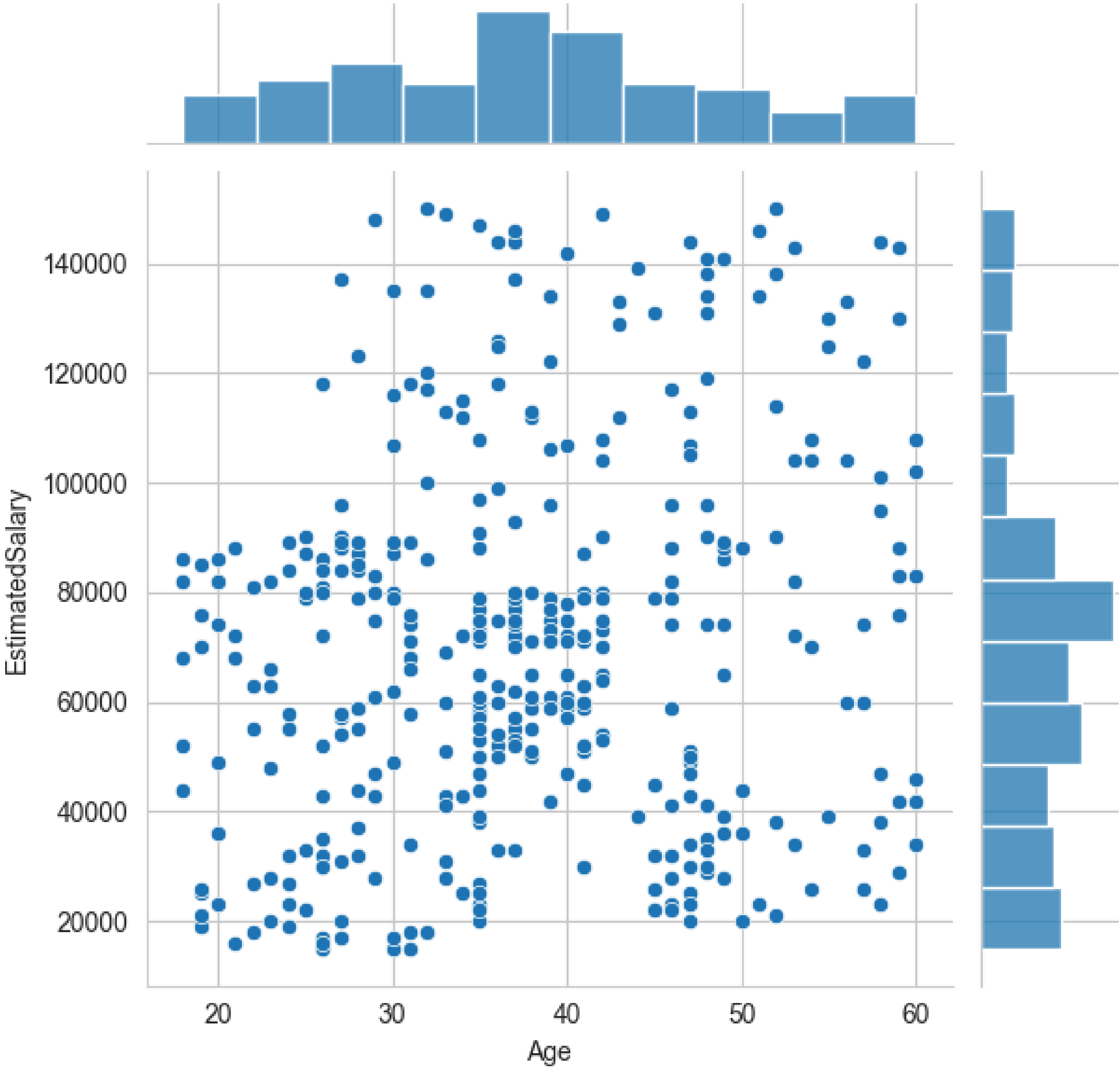
```
Text(0.5, 0, 'Age')
```



In [13]:

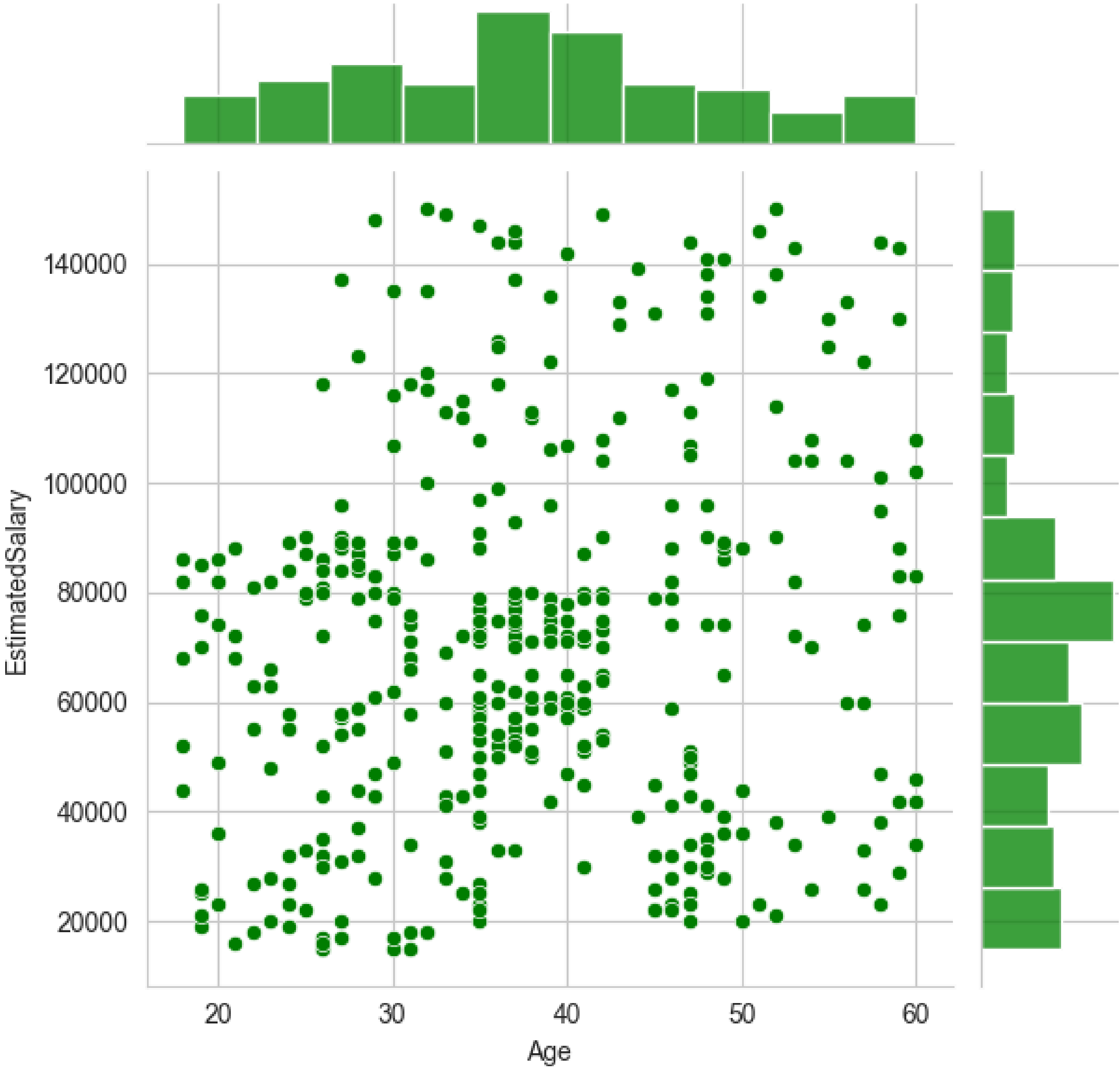
```
sns.jointplot(x='Age', y='EstimatedSalary', data = data)
```

```
Out[13]: <seaborn.axisgrid.JointGrid at 0x262f75b5490>
```



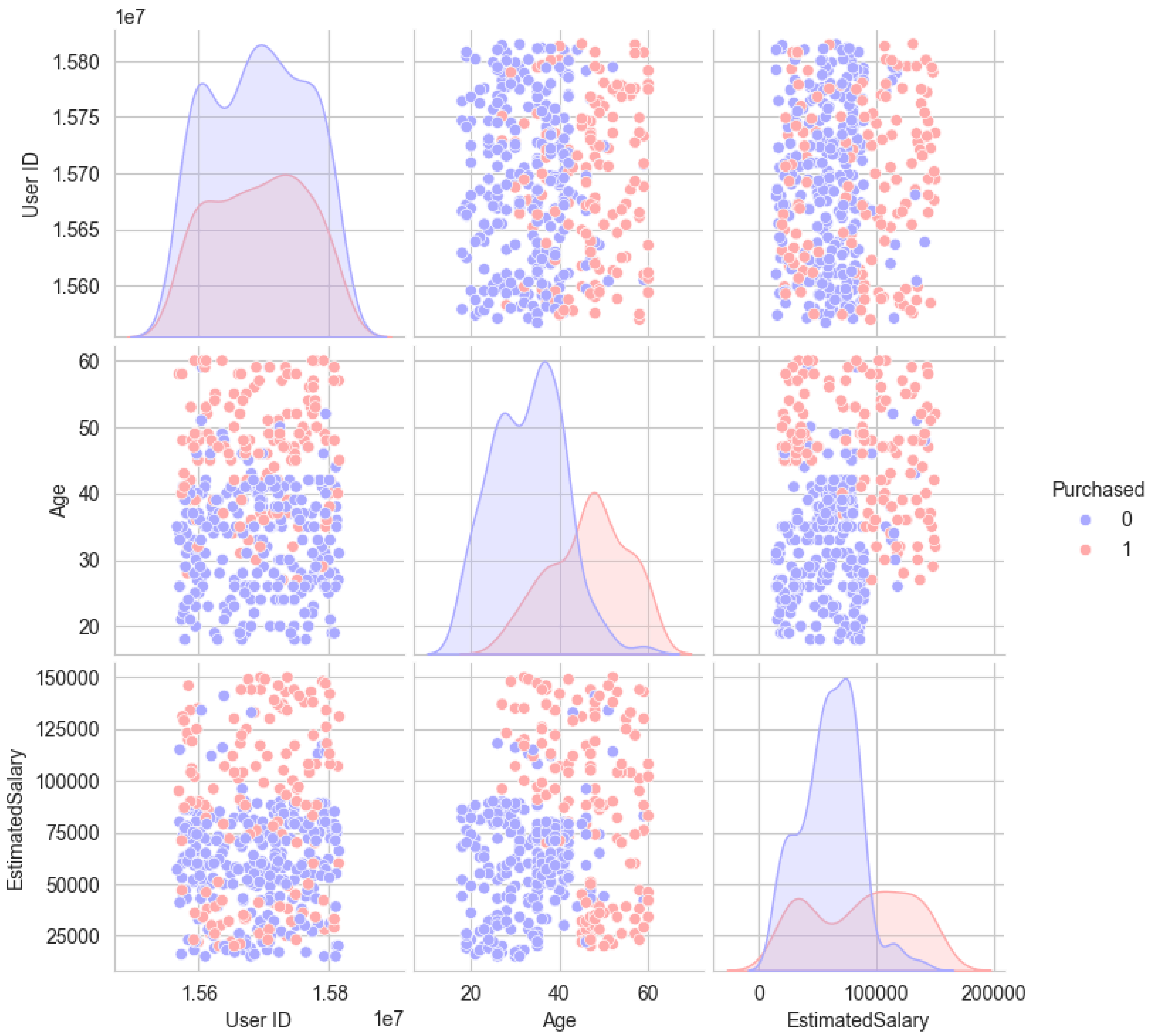
```
In [14]: sns.jointplot(x='Age',y='EstimatedSalary',data= data,color='green')
```

```
Out[14]: <seaborn.axisgrid.JointGrid at 0x262f748c770>
```



```
In [15]: sns.pairplot(data,hue='Purchased',palette='bwr')
```

```
Out[15]: <seaborn.axisgrid.PairGrid at 0x262f758a4e0>
```



In [16]:
```
from sklearn.model_selection import train_test_split
```

In [17]:
```
X = data[['Age','EstimatedSalary']]
y = data['Purchased']
```

In [19]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
```

In [20]:
```
X_train.shape
```

Out[20]:
```
(300, 2)
```

In [21]:
```
X_test.shape
```

Out[21]:
```
(100, 2)
```

In [22]:
```
y_train.shape
```

Out[22]:
```
(300,)
```

In [23]:
```
y_test.shape
```

Out[23]:
```
(100,)
```

In [24]:
```
from sklearn.linear_model import LogisticRegression
```

In [25]:
```
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)
```

Out[25]:
```
▾ LogisticRegression
LogisticRegression()
```

In [26]:
```
predictions = logmodel.predict(X_test)
```

In [27]:
```
from sklearn.metrics import classification_report
```

In [28]:
```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.89      0.96      0.92        68
           1       0.89      0.75      0.81        32

    accuracy                           0.89       100
   macro avg       0.89      0.85      0.87       100
weighted avg       0.89      0.89      0.89       100
```

In [29]:

```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, predictions)
print(cm)
```

```
[[65  3]
 [ 8 24]]
```

In [ ]: