

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

In [4]: ad_data=pd.read_csv("advertising.csv")
ad_data.head()
```

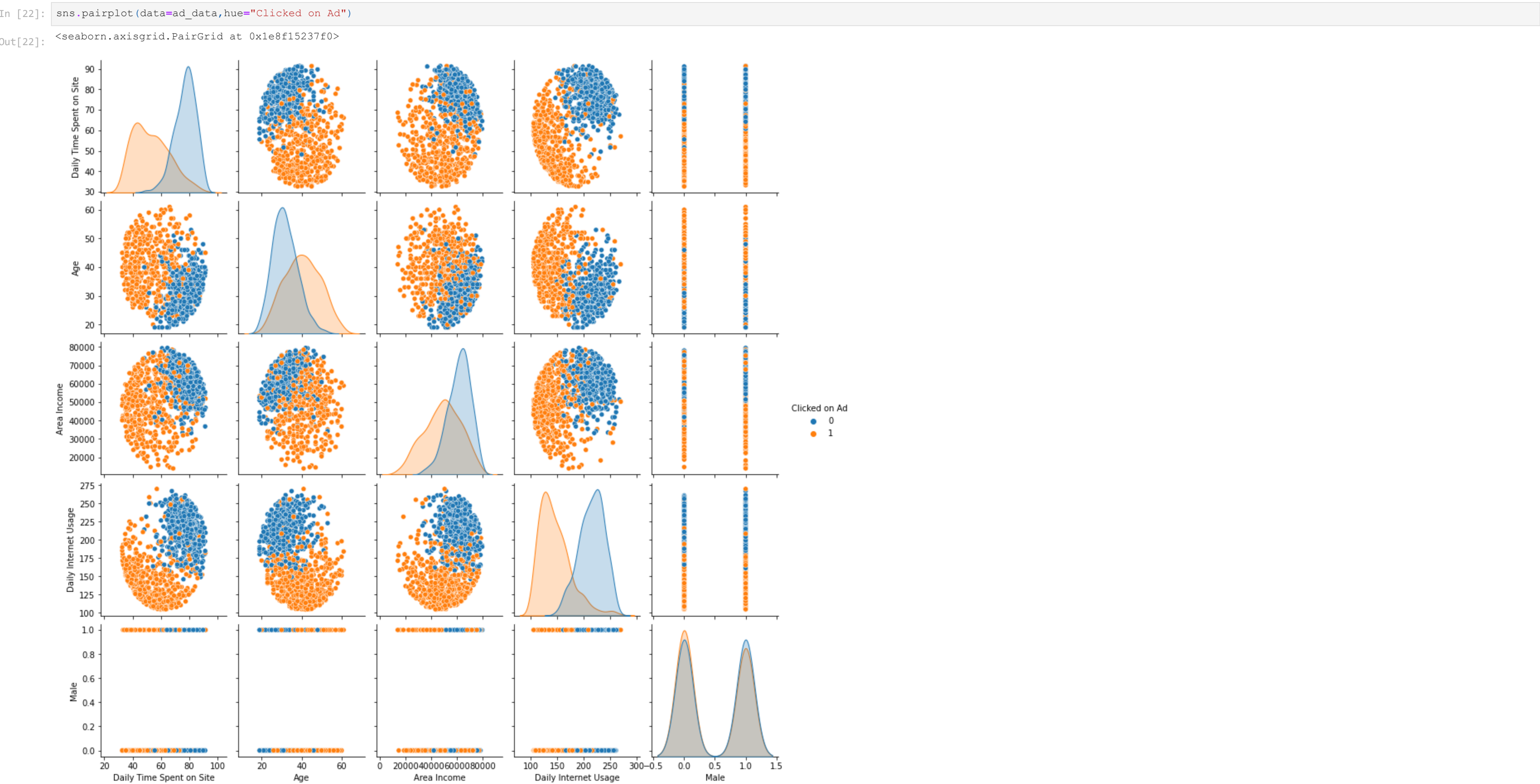
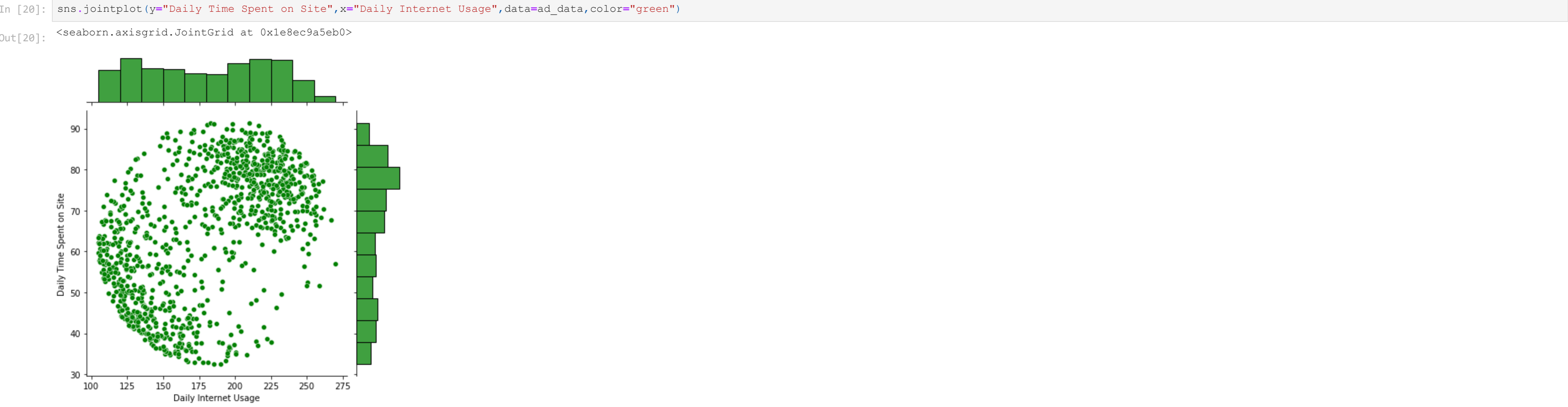
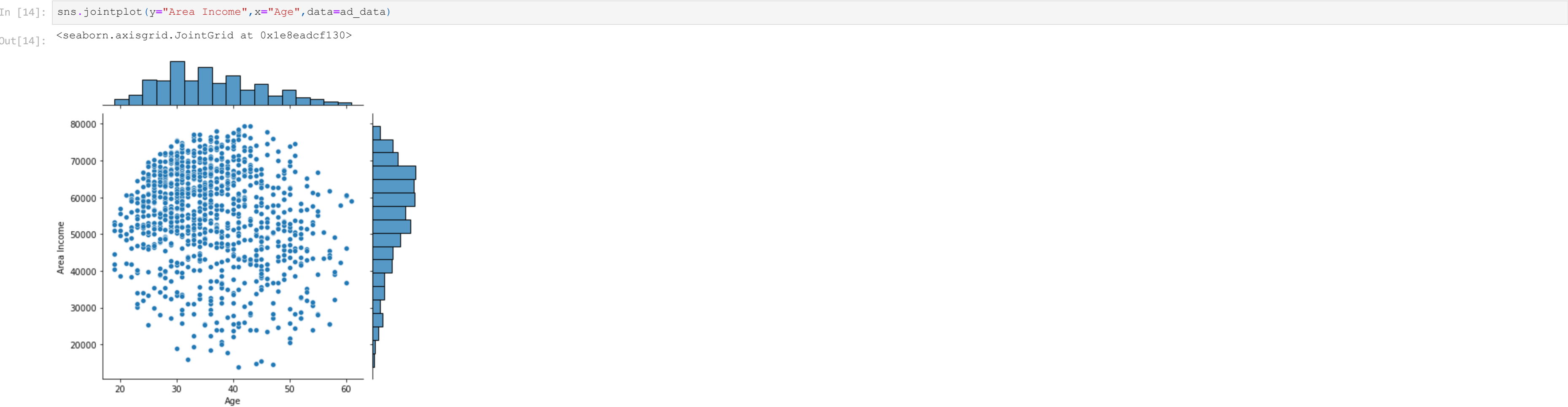
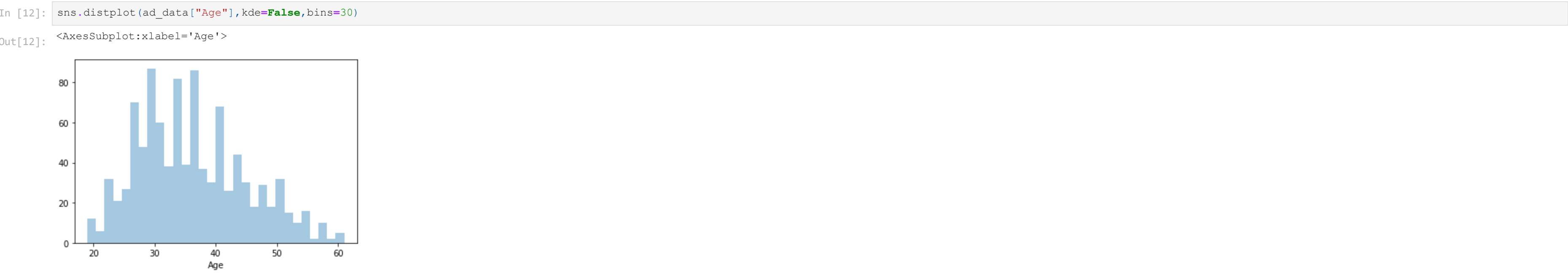
	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Timestamp	Clicked on Ad
0	68.95	35	61833.90	256.09	Cloned 5thgeneration orchestration	Wrightburgh	0	Tunisia	2016-03-27 00:53:11	0
1	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	2016-04-04 01:39:02	0
2	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	2016-03-13 20:35:42	0
3	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	2016-01-10 02:31:19	0
4	68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	2016-06-03 03:36:18	0

```
In [5]: ad_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
#   Column              Non-Null Count  Dtype  
---  --
0   Daily Time Spent on Site  1000 non-null  float64
1   Age                    1000 non-null  int64  
2   Area Income            1000 non-null  float64
3   Daily Internet Usage    1000 non-null  float64
4   Ad Topic Line          1000 non-null  object  
5   City                   1000 non-null  object  
6   Male                   1000 non-null  int64  
7   Country                1000 non-null  object  
8   Timestamp              1000 non-null  object  
9   Clicked on Ad          1000 non-null  int64  
dtypes: float64(3), int64(3), object(4)
memory usage: 78.2+ KB

In [6]: ad_data.describe()
```

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Male	Clicked on Ad
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	65.000200	36.009000	55000.000080	180.000100	0.481000	0.500000
std	15.853615	8.785562	13414.634022	43.902339	0.499889	0.50025
min	32.600000	19.000000	13996.500000	104.780000	0.000000	0.000000
25%	51.360000	29.000000	47031.802500	138.830000	0.000000	0.000000
50%	68.215000	35.000000	57012.300000	183.130000	0.000000	0.500000
75%	78.547500	42.000000	65470.635000	218.792500	1.000000	1.000000
max	91.430000	61.000000	79484.800000	269.960000	1.000000	1.000000



```
In [32]: ad_data.head()

Out[32]:
```

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Timestamp	Clicked on Ad
0	68.95	35	61833.90	256.09	Cloned 5thgeneration orchestration	Wrightburgh	0	Tunisia	2016-03-27 00:53:11	0
1	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	2016-04-04 01:39:02	0
2	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	2016-03-13 20:35:42	0
3	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	2016-01-10 02:31:19	0
4	68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	2016-06-03 03:36:18	0

```
In [39]: X=ad_data[['Daily Time Spent on Site', 'Age', 'Area Income','Daily Internet Usage','Male']]
y= ad_data["Clicked on Ad"]

In [40]: from sklearn.model_selection import train_test_split

In [41]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)

In [42]: from sklearn.linear_model import LogisticRegression

In [43]: lm=LogisticRegression()

In [44]: lm.fit(X_train,y_train)

Out[44]: LogisticRegression()

In [45]: predictions=lm.predict(X_test)

In [46]: from sklearn.metrics import classification_report

In [48]: print(classification_report(y_test,predictions))
```

	precision	recall	f1-score	support
0	0.86	0.96	0.91	162
1	0.96	0.85	0.90	168
accuracy			0.91	330
macro avg	0.91	0.91	0.91	330
weighted avg	0.91	0.91	0.91	330