Logistic Regression Project

About the project ; This is a fake advertising data set, indicating whether or not a particular internet user clicked on an Advertisement on a company website.

It was gotten from Pierian Data.

This data set contains the following features:

1. Daily Time Spent on Site': consumer time on site in minutes
2. Age: cutomer age in years
3. Area Income': Avg. Income of geographical area of consumer
4. Daily Internet Usage': Avg. minutes a day consumer is on the internet
5. Ad Topic Line': Headline of the advertisement
6. City: City of consumer
7. Male: Whether or not consumer was male
8. Country: Country of consumer
9. Timestamp: Time at which consumer clicked on Ad or closed window
10. Clicked on Ad': 0 or 1 indicated clicking on Ad

Analysing Data

Imprtoing Libraries

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

In [2]:
```
ad_data = pd.read_csv('advertising.csv')
```

In [3]:
```
ad_data.head ()
```

Out[3]:

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Timestamp	Clicked on Ad

	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 [4]:
```
ad_data.tail()
```

Out[4]:

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Timestamp	Clicked on Ad
995	72.97	30	71384.57	208.58	Fundamental modular algorithm	Duffystad	1	Lebanon	2016-02-11 21:49:00	1
996	51.30	45	67782.17	134.42	Grass-roots cohesive monitoring	New Darlene	1	Bosnia and Herzegovina	2016-04-22 02:07:01	1
997	51.63	51	42415.72	120.37	Expanded intangible solution	South Jessica	1	Mongolia	2016-02-01 17:24:57	1
998	55.55	19	41920.79	187.95	Proactive bandwidth-monitored policy	West Steven	0	Guatemala	2016-03-24 02:35:54	0
999	45.01	26	29875.80	178.35	Virtual 5thgeneration emulation	Ronniemouth	0	Brazil	2016-06-03 21:43:21	1

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 ()
```

Out[6]:

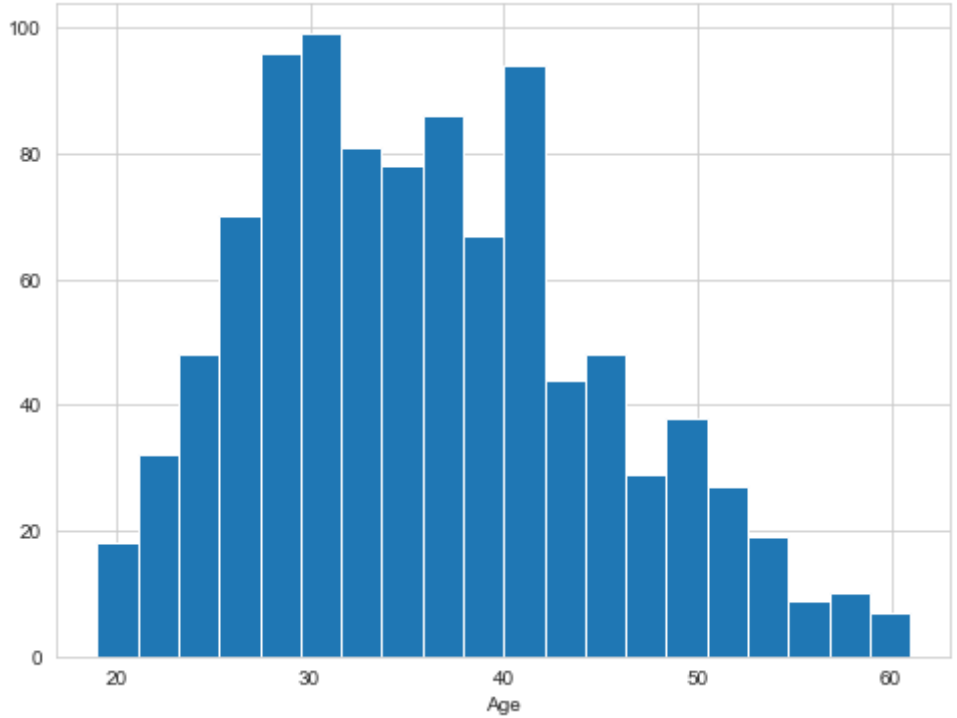
	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Male	Clicked on Ad
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.00000
mean	65.000200	36.009000	55000.000080	180.000100	0.481000	0.50000
std	15.853615	8.785562	13414.634022	43.902339	0.499889	0.50025
min	32.600000	19.000000	13996.500000	104.780000	0.000000	0.00000
25%	51.360000	29.000000	47031.802500	138.830000	0.000000	0.00000
50%	68.215000	35.000000	57012.300000	183.130000	0.000000	0.50000
75%	78.547500	42.000000	65470.635000	218.792500	1.000000	1.00000
max	91.430000	61.000000	79484.800000	269.960000	1.000000	1.00000

Exploratory Data Analysis

Creating a histogram of the Age

In [7]:
```
plt.figure(figsize=(8,6))
sns.set_style('whitegrid')
ad_data['Age'].hist(bins=20)
plt.xlabel('Age')
```

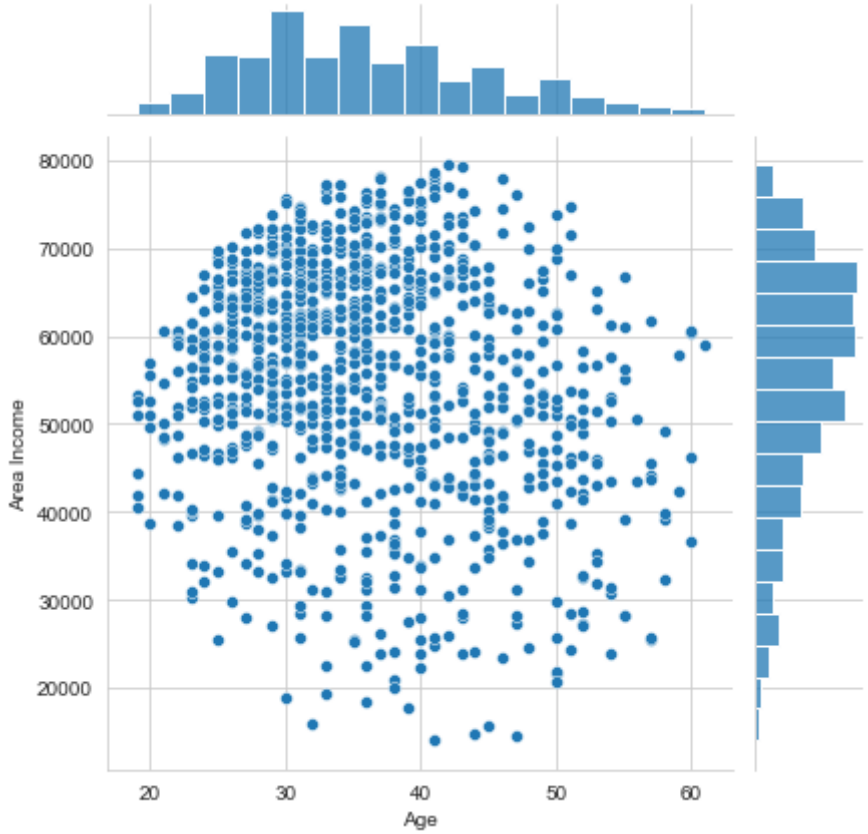
Out[7]: Text(0.5, 0, 'Age')



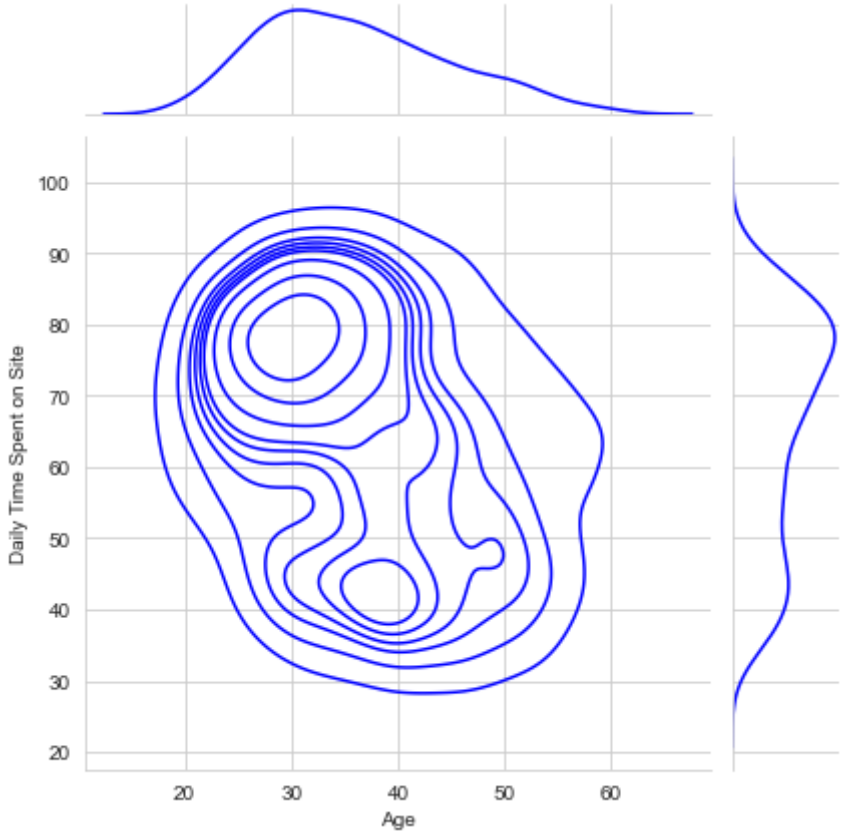
In [8]:
```
plt.figure(figsize=(8,6))
sns.jointplot (y='Area Income', x='Age', data=ad_data)
```

Out[8]: <seaborn.axisgrid.JointGrid at 0x16c27ab1040>

<Figure size 576x432 with 0 Axes>



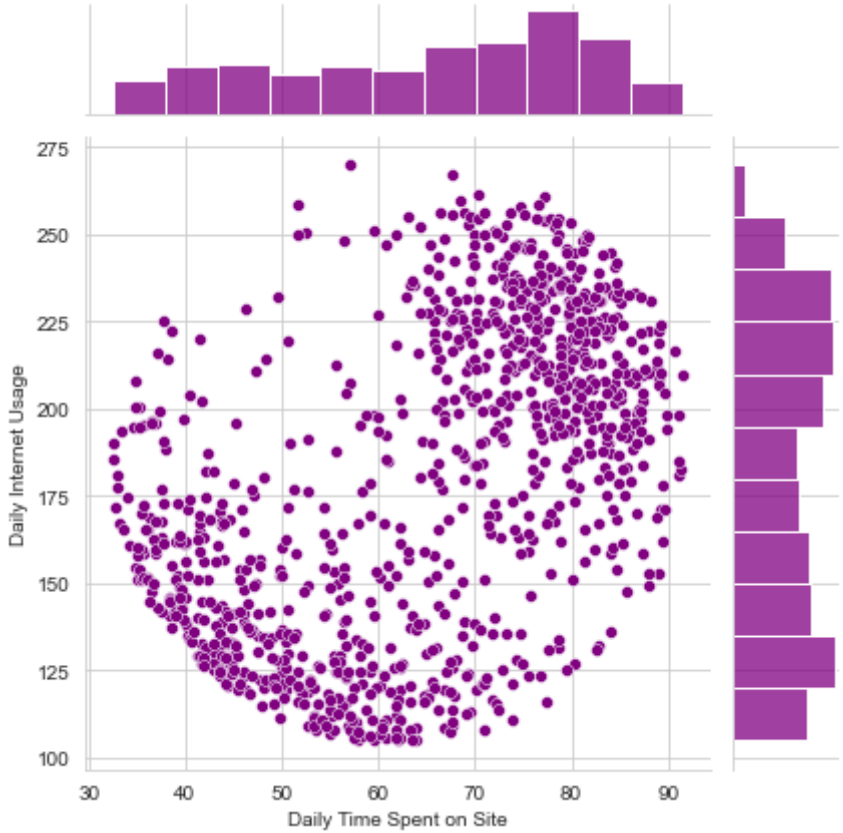
In [9]:
```
sns.jointplot(x='Age',y='Daily Time Spent on Site',data=ad_data,color='blue',kind='kde');
```



In [10]:
```
plt.figure(figsize=(8,6))   #plt.figure(figsize=(8,6))
sns.jointplot (x= 'Daily Time Spent on Site', y= 'Daily Internet Usage', data=ad_data, color='purple' )
```

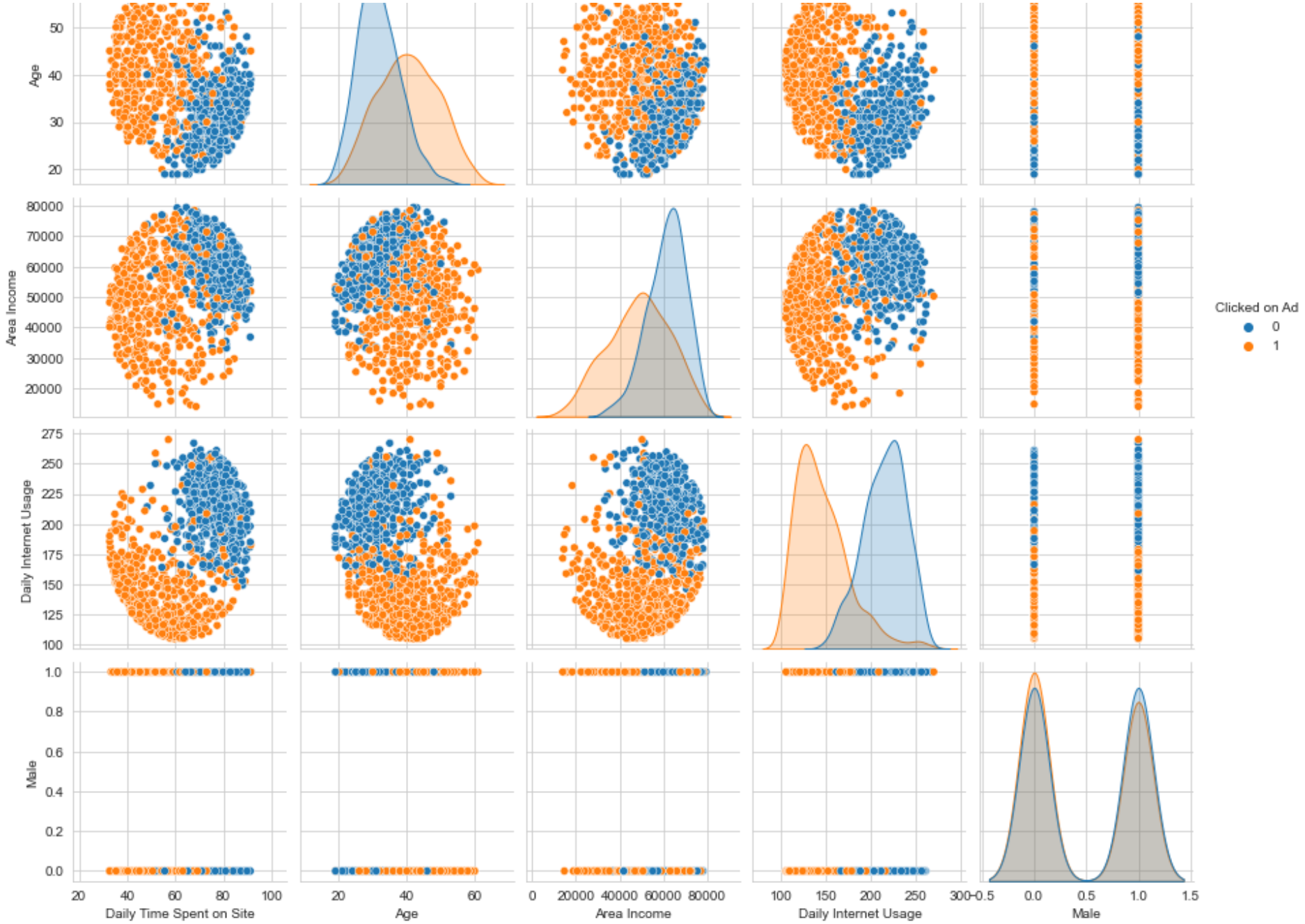
Out[10]:
```
<seaborn.axisgrid.JointGrid at 0x16c27d80220>

<Figure size 576x432 with 0 Axes>
```



In [11]:
```
sns.pairplot(data=ad_data, hue='Clicked on Ad')
```

Out[11]: <seaborn.axisgrid.PairGrid at 0x16c27e28a30>



Logistic Regression

In [12]:
```
from sklearn.model_selection import train_test_split
```

In [13]:
```
X = ad_data[['Daily Time Spent on Site', 'Age', 'Area Income','Daily Internet Usage', 'Male']]
y = ad_data['Clicked on Ad']
```

In [14]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

In [15]:
```
from sklearn.linear_model import LogisticRegression
```

In [16]:
```
logmodel = LogisticRegression()
logmodel.fit(X_train, y_train)
```

Out[16]:
```
LogisticRegression()
```

Predictions and Evaluations

In [17]:
```
predictions = logmodel.predict(X_test)
```

In [19]:
```
from sklearn.metrics import classification_report,confusion_matrix
print(classification_report(y_test,predictions))
#print(confusion_martix(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.91      0.95      0.93       157
           1       0.94      0.90      0.92       143

    accuracy                           0.93       300
   macro avg       0.93      0.93      0.93       300
weighted avg       0.93      0.93      0.93       300
```

In [ ]:
```

```