Assignment 5: Logistic Regression

A. Import a few libraries you think you'll need.

In [1]:

```
import pandas as pd
import seaborn as sb
import numpy as np
import sklearn
from matplotlib import pyplot as plt
```

B. Read in the advertising.csv file and set it to a data frame called ad_data

In [2]:

```
ad_data=pd.read_csv("advertising.csv")
```

In [3]:

```
ad_data
```

Out[3]:

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Timestamp
0	68.95	35	61833.90	256.09	Cloned 5thgeneration orchestration	Wrightburgh	0	Tunisia	27-03-2016 00:5
1	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	04-04-2016 01:3
2	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	13-03-2016 20:3
3	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	10-01-2016 02:3
4	68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	03-06-2016 03:3
...	...	...	...	...	...	...	...	...	..
995	72.97	30	71384.57	208.58	Fundamental modular algorithm	Duffystad	1	Lebanon	11-02-2016 21:4
996	51.30	45	67782.17	134.42	Grass-roots cohesive monitoring	New Darlene	1	Bosnia and Herzegovina	22-04-2016 02:0
997	51.63	51	42415.72	120.37	Expanded intangible solution	South Jessica	1	Mongolia	01-02-2016 17:2
998	55.55	19	41920.79	187.95	Proactive bandwidth-monitored policy	West Steven	0	Guatemala	24-03-2016 02:3
999	45.01	26	29875.80	178.35	Virtual 5thgeneration emulation	Ronniemouth	0	Brazil	03-06-2016 21:4

1000 rows × 10 columns

C. Check the head of ad_data

In [4]:

```
ad_data.head()
```

Out[4]:

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

D. Check whether any missing data are there or not

In [5]:

```
ad_data.isnull().sum()
```

Out[5]:

```
Daily Time Spent on Site    0
Age                         0
Area Income                 0
Daily Internet Usage        0
Ad Topic Line               0
City                        0
Male                        0
Country                     0
Timestamp                   0
Clicked on Ad               0
dtype: int64
```

E. Calculate several statistical measures on numerical attributes.

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

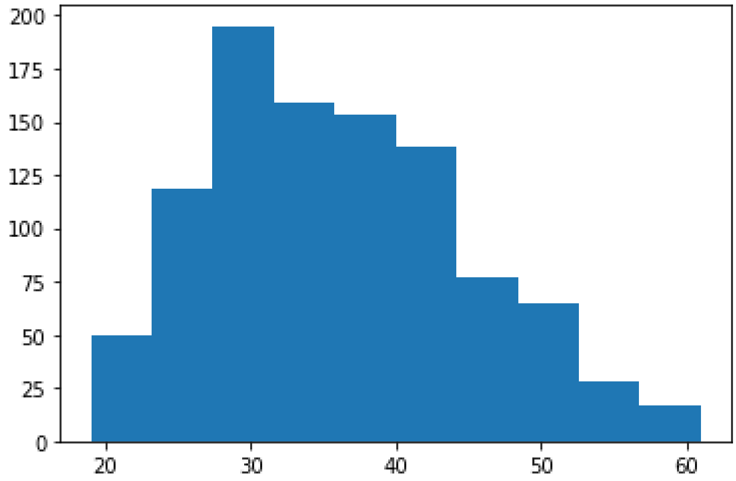
F. Create a histogram of the ‘Age’ using seaborn package.

In [7]:

```
plt.hist(ad_data['Age'])
```

Out[7]:

```
(array([ 50., 118., 195., 159., 153., 138.,  77.,  65.,  28.,  17.]),
 array([19. , 23.2, 27.4, 31.6, 35.8, 40. , 44.2, 48.4, 52.6, 56.8, 61. ]),
 <BarContainer object of 10 artists>)
```



G. Create a jointplot showing ‘Area Income’ versus ‘Age’.

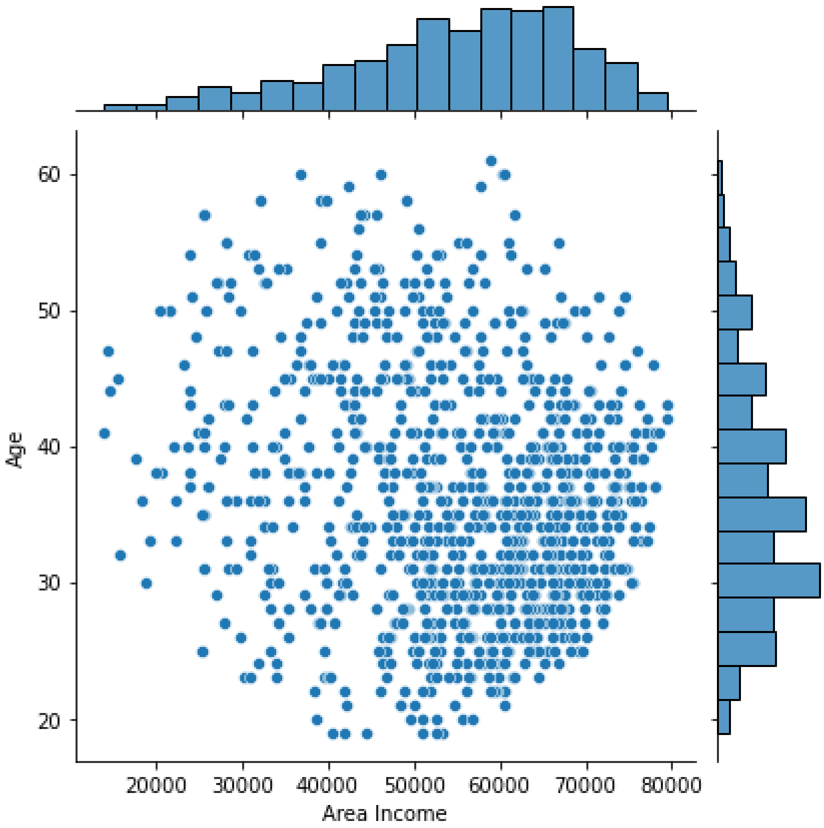
In [8]:

```
sb.jointplot(ad_data['Area Income'],ad_data['Age'])
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretat
ion.
  warnings.warn(
```

Out[8]:

```
<seaborn.axisgrid.JointGrid at 0x1ef86c75fa0>
```



H. Create a jointplot showing the kde distributions of ‘Daily Time spent on site’ vs. ‘Age’.

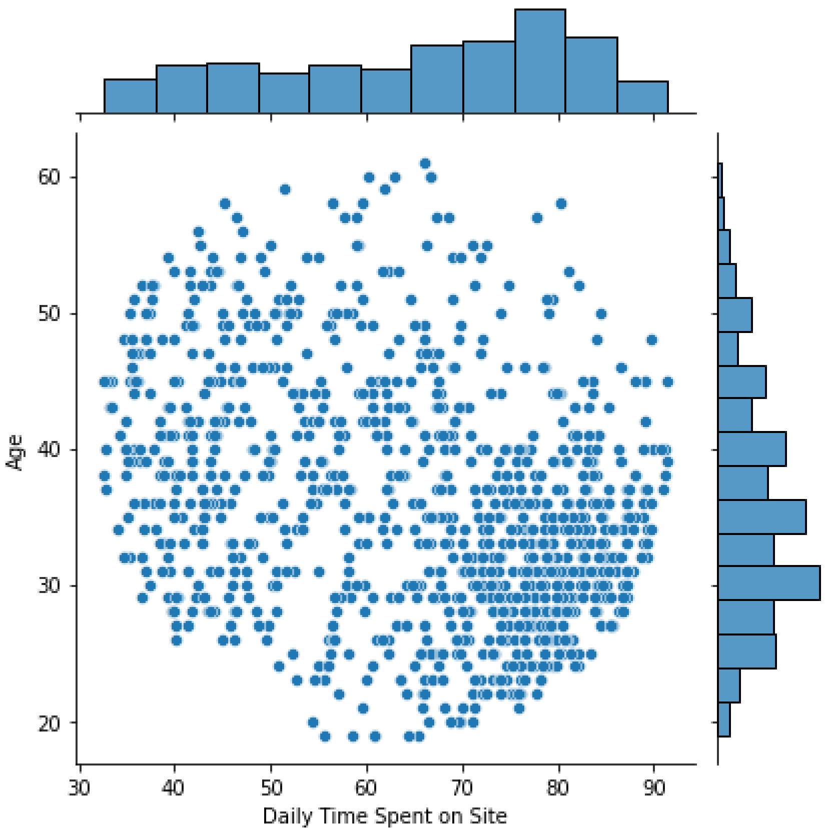
In [9]:

```
sb.jointplot(ad_data['Daily Time Spent on Site'],ad_data['Age'])
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretat
ion.
  warnings.warn(
```

Out[9]:

```
<seaborn.axisgrid.JointGrid at 0x1ef8c3ce6d0>
```



I. Create a jointplot of 'Daily Time Spent on Site' vs. 'Daily Internet Usage'.

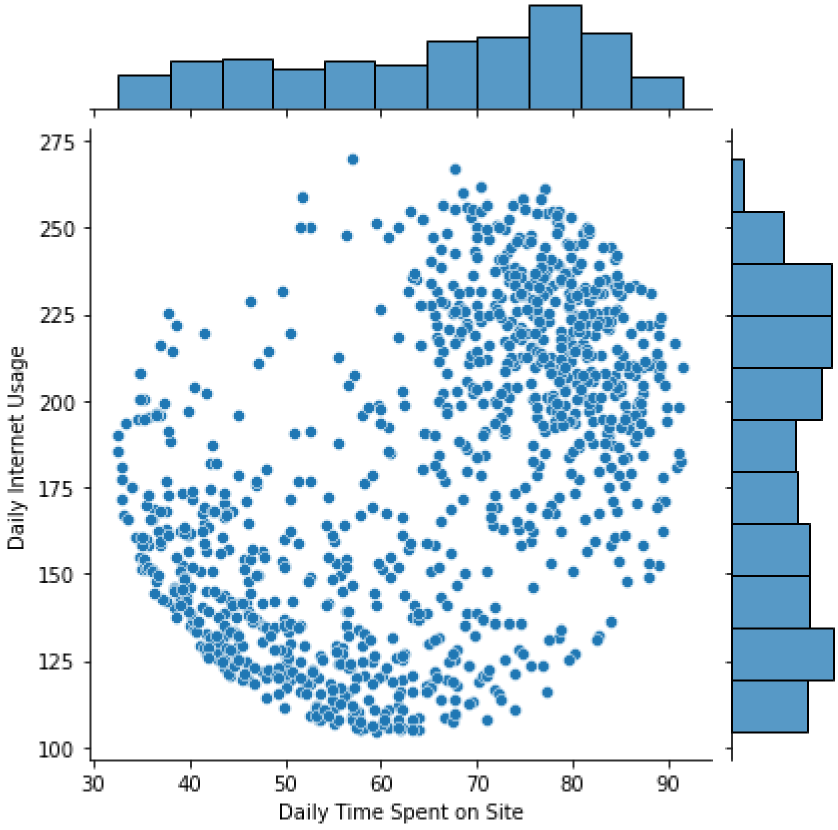
In [10]:

```
sb.jointplot(ad_data['Daily Time Spent on Site'],ad_data['Daily Internet Usage'])
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretat
ion.
  warnings.warn(
```

Out[10]:

```
<seaborn.axisgrid.JointGrid at 0x1ef8c4f0ca0>
```



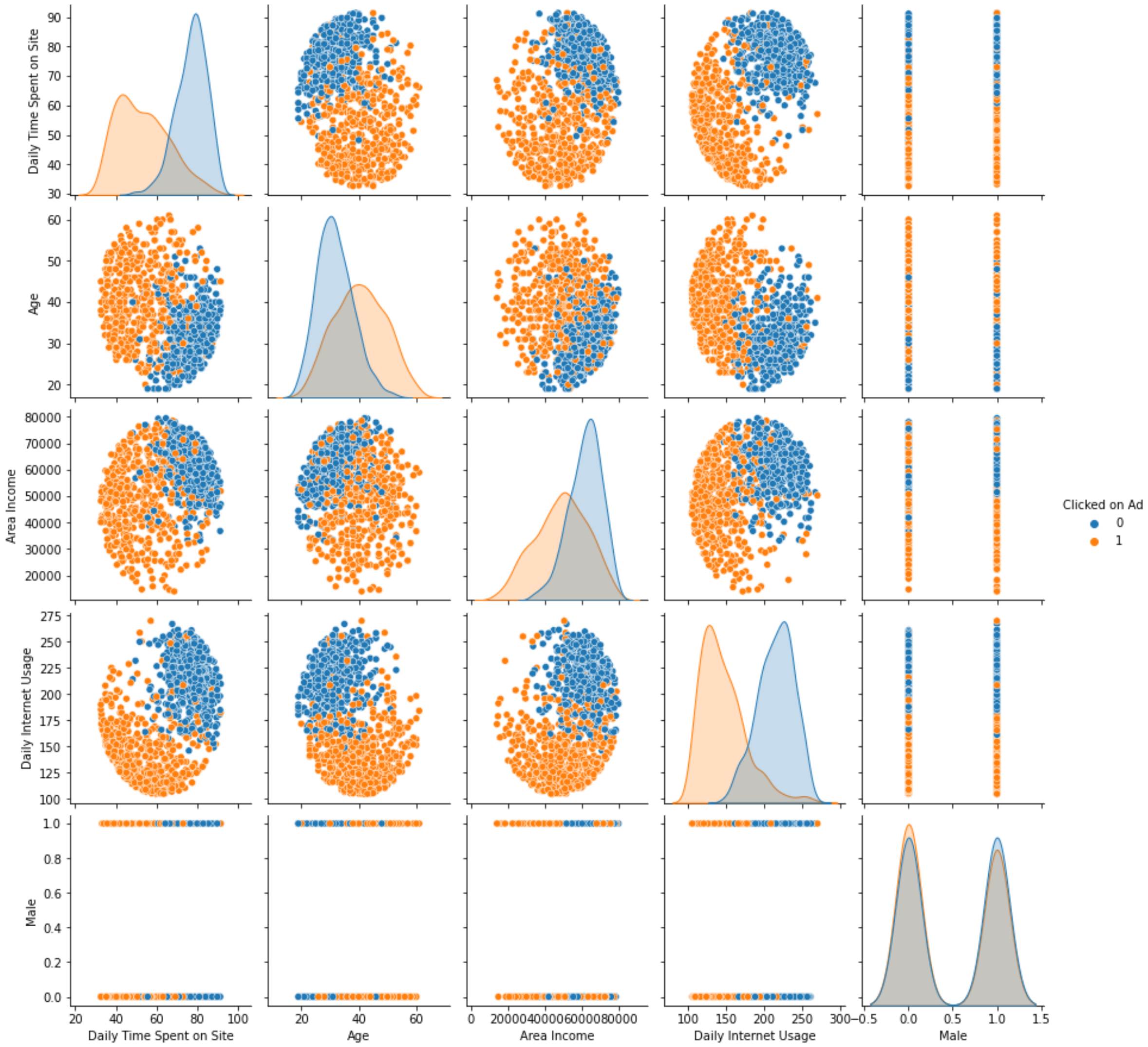
J. Create a pairplot with the hue defined by the 'Clicked on Ad' column feature.

In [11]:

```
sb.pairplot(ad_data,hue='Clicked on Ad')
```

Out[11]:

```
<seaborn.axisgrid.PairGrid at 0x1ef8c5add90>
```



K. Choose columns that you want to use for Logistic Regression.

In [12]:

```
x=ad_data[['Daily Time Spent on Site','Area Income','Daily Internet Usage']]
y=ad_data[['Clicked on Ad']]
```

L. Split the data into training set and testing set so that your testing set consists 25% of total data.

In [13]:

```
from sklearn.model_selection import train_test_split
```

In [14]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=42)
```

M. Train and fit a logistic regression model on the training set.

In [15]:

```
from sklearn.linear_model import LogisticRegression
logic=LogisticRegression()
logic.fit(x_train,y_train)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\utils\validation.py:72: D
ataConversionWarning: A column-vector y was passed when a 1d array was expec
ted. Please change the shape of y to (n_samples, ), for example using ravel
().
  return f(**kwargs)
```

Out[15]:

```
LogisticRegression()
```

N. Display all the coefficients values of the fitted Logistic Regression Model.

In [16]:

```
logic.coef_
```

Out[16]:

```
array([[-1.57040367e-01, -1.05229723e-04, -6.98627182e-02]])
```

O. Predict class label for the testing dataset

In [17]:

```
predict=logic.predict(x_test)
```

In [18]:

```
predict
```

Out[18]:

```
array([1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1,
       1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1,
       1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0,
       0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1,
       1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1,
       1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1,
       0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1,
       0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1], dtype=int64)
```

P. Create a classification report for the model.

In [19]:

```
from sklearn.metrics import classification_report,confusion_matrix
print(classification_report(y_test,predict))
```

```
              precision    recall  f1-score   support

           0       0.92      0.97      0.94       120
           1       0.97      0.92      0.94       130

    accuracy                           0.94       250
   macro avg       0.94      0.94      0.94       250
weighted avg       0.95      0.94      0.94       250
```

Q. Create the Confusion matrix for your fitted model and hence calculate the model accuracy

In [20]:

```
print(confusion_matrix(y_test,predict))
```

```
[[116   4]
 [ 10 120]]
```