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

%matplotlib inline
```

In [ ]:
```
data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/advertising.csv')
```

In [ ]:
```
data.head(10)
```

Out[ ]:

	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	
1	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	2016-04-04 01:39:02	
2	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	2016-03-13 20:35:42	
3	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	2016-01-10 02:31:19	
4	68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	2016-06-03 03:36:18	
5	59.99	23	59761.56	226.74	Sharable client-driven software	Jamieberg	1	Norway	2016-05-19 14:30:17	
6	88.91	33	53852.85	208.36	Enhanced dedicated support	Brandonstad	0	Myanmar	2016-01-28 20:59:32	
7	66.00	48	24593.33	131.76	Reactive local challenge	Port Jefferybury	1	Australia	2016-03-07 01:40:15	
8	74.53	30	68862.00	221.51	Configurable coherent function	West Colin	1	Grenada	2016-04-18 09:33:42	
9	69.88	20	55642.32	183.82	Mandatory homogeneous architecture	Ramirezton	1	Ghana	2016-07-11 01:42:51	

In [ ]:
```
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
```

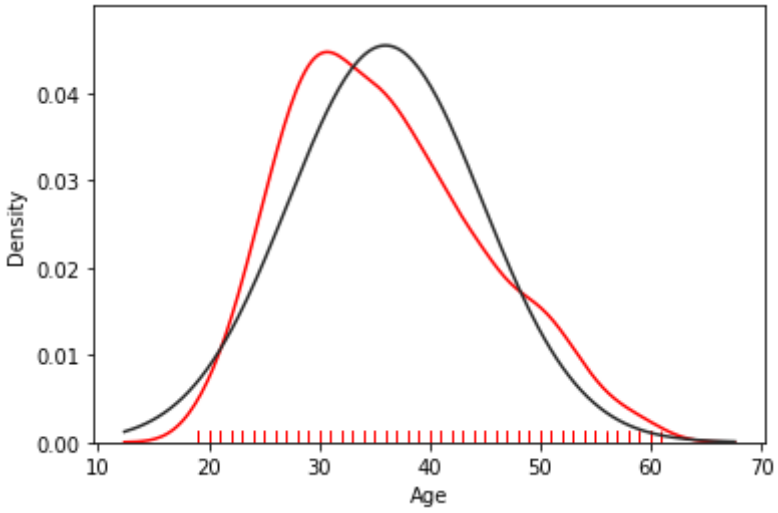
In [ ]:
```
data.describe()
```

Out[ ]:

	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

In [ ]:
```
from scipy.stats import norm
sns.distplot(data['Age'], hist=False, color='r', rug=True, fit=norm);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please
adapt your code to use either `displot` (a figure-level function with similar flexib
ility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2056: FutureWarning:
The `axis` variable is no longer used and will be removed. Instead, assign variables
directly to `x` or `y`.
  warnings.warn(msg, FutureWarning)
```

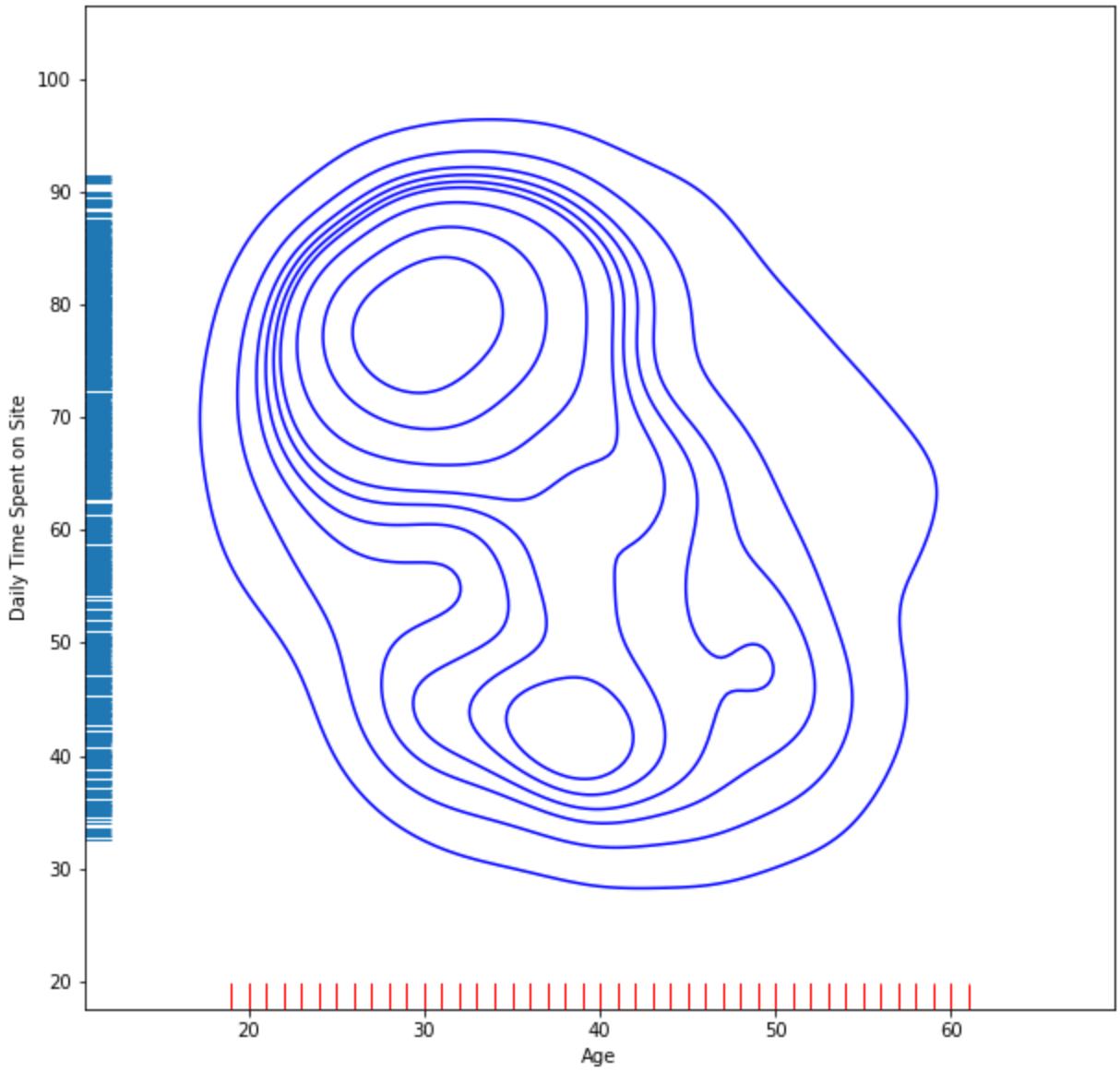


In [ ]:
```
f, ax = plt.subplots(figsize=(10, 10))
sns.kdeplot(data.Age, data['Daily Time Spent on Site'], color="b", ax=ax)
```

```
sns.rugplot(data.Age, color="r", ax=ax)
sns.rugplot(data['Daily Time Spent on Site'], vertical=True, ax=ax)
```

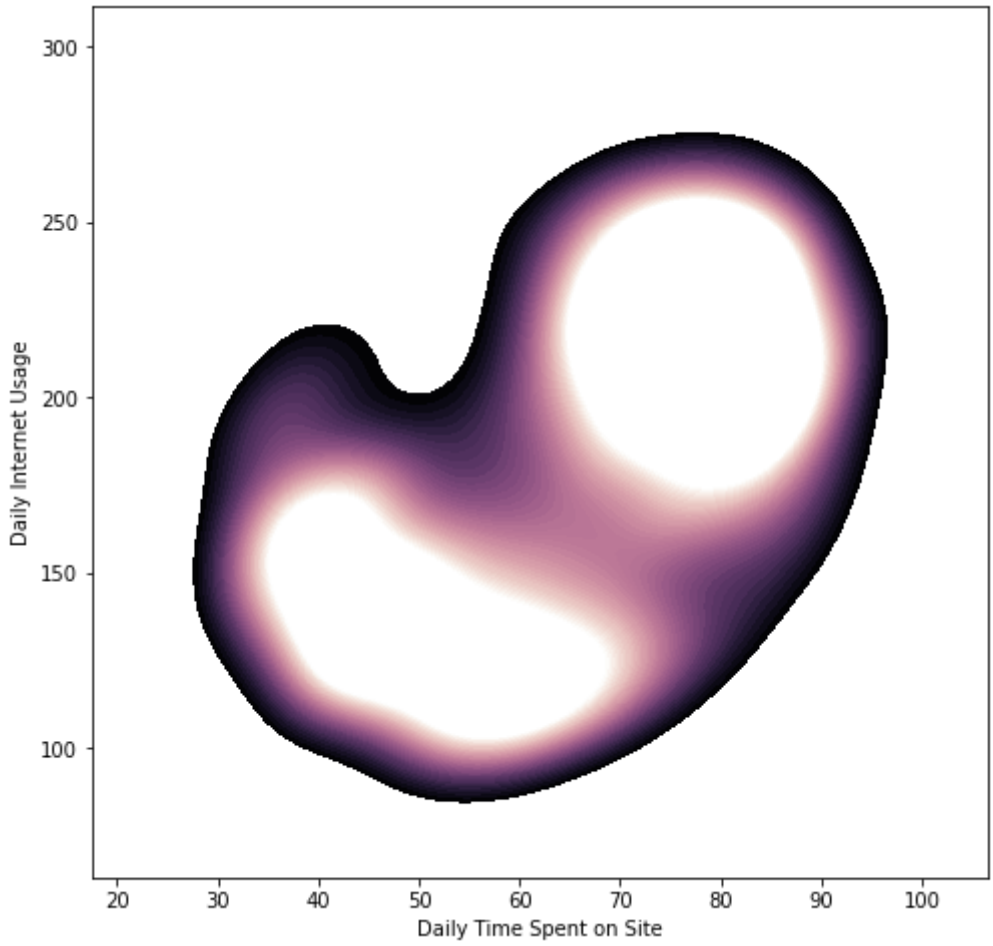
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
s the following variable as a keyword arg: y. From version 0.12, the only valid posi
tional argument will be `data`, and passing other arguments without an explicit keyw
ord will result in an error or misinterpretation.
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2065: FutureWarning:
Using `vertical=True` to control the orientation of the plot  is deprecated. Instea
d, assign the data directly to `y`.
  warnings.warn(msg, FutureWarning)
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe8793b9b10>



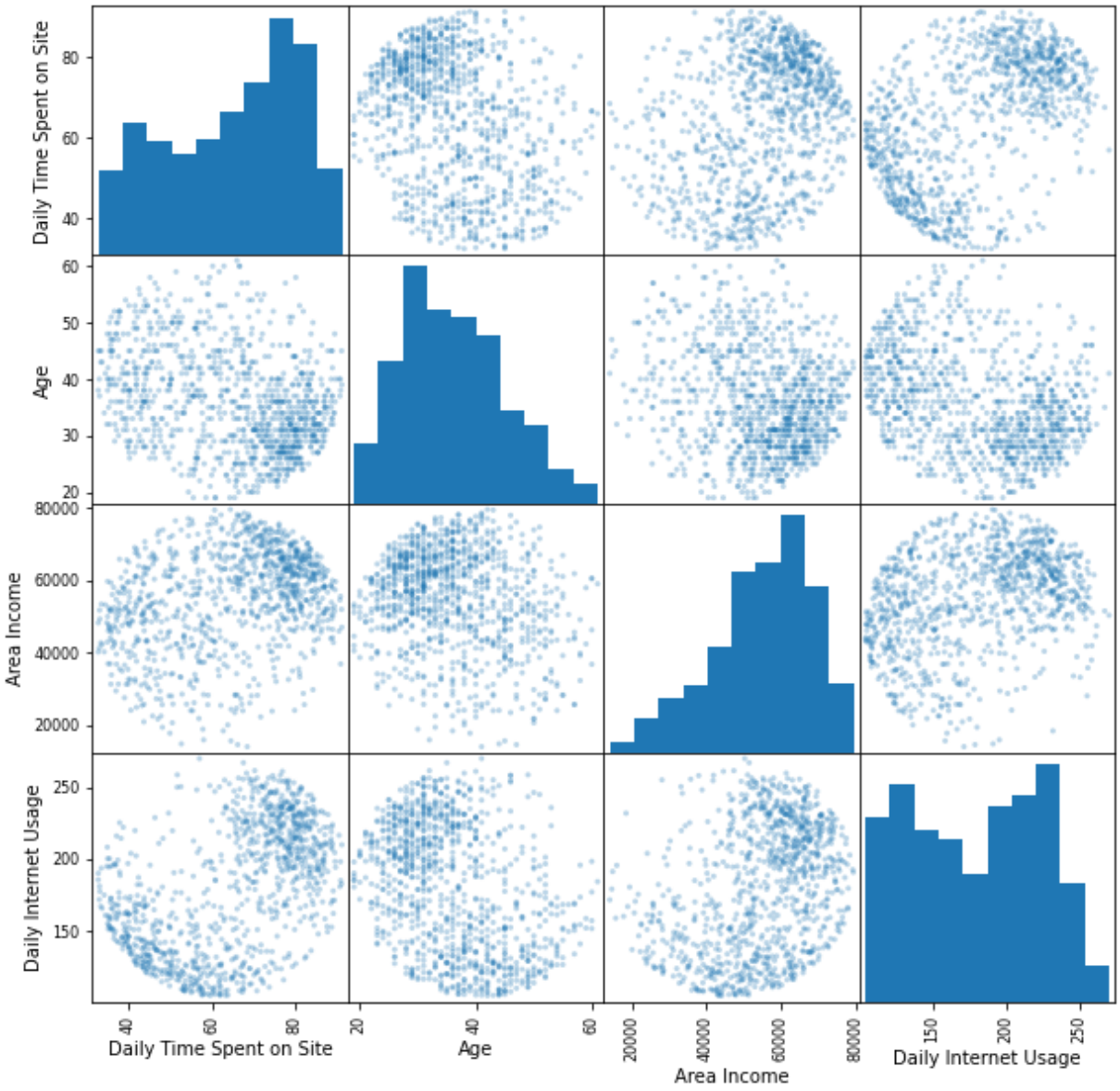
In [ ]:
```
f, ax = plt.subplots(figsize=(8, 8))
cmap = sns.cubehelix_palette(as_cmap=True, start=0, dark=0, light=3, reverse=True)
sns.kdeplot(data["Daily Time Spent on Site"], data['Daily Internet Usage'],
    cmap=cmap, n_levels=100, shade=True);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
s the following variable as a keyword arg: y. From version 0.12, the only valid posi
tional argument will be `data`, and passing other arguments without an explicit keyw
ord will result in an error or misinterpretation.
  FutureWarning
```



In [ ]:
```
from pandas.plotting import scatter_matrix
scatter_matrix(data[['Daily Time Spent on Site', 'Age','Area Income', 'Daily Interne
    alpha=0.3, figsize=(10,10))
```

Out[ ]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fe87777dd10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fe8777a67d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fe877765ed0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fe877720810>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fe8776e4f50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fe87769b3d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fe877651a50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fe877608b90>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fe87761f050>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fe8775d3790>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fe8775444d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fe8774f9b50>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fe8774bb210>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fe8774f0890>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fe8774a7f10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fe8774685d0>]],
      dtype=object)
```



In [ ]:
```
object_variables = ['Ad Topic Line', 'City', 'Country']
data[object_variables].describe(include=['O'])
```

Out[ ]:

	Ad Topic Line	City	Country
count	1000	1000	1000
unique	1000	969	237
top	Phased clear-thinking encoding	Lisamouth	Czech Republic
freq	1	3	9

In [ ]:
```
pd.crosstab(index=data['Country'], columns='count').sort_values(['count'], ascending
```

Out[ ]:

col_0	count
Country	
France	9
Czech Republic	9
Afghanistan	8
Australia	8

col_0	count
Country	
Turkey	8
South Africa	8
Senegal	8
Peru	8
Micronesia	8
Greece	8
Cyprus	8
Liberia	8
Albania	7
Bosnia and Herzegovina	7
Taiwan	7
Bahamas	7
Burundi	7
Cambodia	7
Venezuela	7
Fiji	7

In [1]:
```
# data = data.drop(['Ad Topic Line', 'City', 'Country'], axis=1)
```

In [ ]:
```
data['Timestamp'] = pd.to_datetime(data['Timestamp'])

data['Month'] = data['Timestamp'].dt.month
data['Day of the month'] = data['Timestamp'].dt.day
data["Day of the week"] = data['Timestamp'].dt.dayofweek
data['Hour'] = data['Timestamp'].dt.hour
data = data.drop(['Timestamp'], axis=1)

data.head()
```

Out[ ]:

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Male	Clicked on Ad	Month	Day of the month	Day of the week	Hour
0	68.95	35	61833.90	256.09	0	0	3	27	6	0
1	80.23	31	68441.85	193.77	1	0	4	4	0	1
2	69.47	26	59785.94	236.50	0	0	3	13	6	20
3	74.15	29	54806.18	245.89	1	0	1	10	6	2
4	68.37	35	73889.99	225.58	0	0	6	3	4	3

Train and Test datasets

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

```
X = data[['Daily Time Spent on Site', 'Age', 'Area Income', 'Daily Internet Usage',
    'Male', 'Month', 'Day of the month' ,'Day of the week']]
y = data['Clicked on Ad']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_sta
```

In [ ]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

In [ ]:
```
model_1 = LogisticRegression(solver='lbfgs')
model_1.fit(X_train, y_train)
predictions_LR = model_1.predict(X_test)

print('Logistic regression accuracy:', accuracy_score(predictions_LR, y_test))
print('')
print('Confusion matrix:')
print(confusion_matrix(y_test,predictions_LR))
```

```
Logistic regression accuracy: 0.906060606060606

Confusion matrix:
[[158   4]
 [ 27 141]]
```

DecisionTreeClassifier

In [ ]:
```
from sklearn.tree import DecisionTreeClassifier

model_2 = DecisionTreeClassifier()
model_2.fit(X_train, y_train)
predictions_DT = model_2.predict(X_test)

print('Decision tree accuracy:', accuracy_score(predictions_DT, y_test))
print('')
print('Confusion matrix:')
print(confusion_matrix(y_test,predictions_DT))
```

```
Decision tree accuracy: 0.9393939393939394

Confusion matrix:
[[153   9]
 [ 11 157]]
```

In [ ]:
```

```