AD-Click-Prediction

1. Importing Libraries

In [98]:

```
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import GridSearchCV
from sklearn.pipeline import Pipeline

matplotlib.rcParams['figure.dpi'] = 120 #resolution
matplotlib.rcParams['figure.figsize'] = (8,6) #figure size
sns.set_style('darkgrid')
color = sns.color_palette()

#Display all the columns ofthe dataframe
pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)

#to ignore the warnings
import warnings
warnings.filterwarnings("ignore")
```

2. Importing Data

In [5]:

```
root = '/Users/mac/Desktop/DataScience/Pojects_ds/Ad-Click-Prediction/'

df = pd.read_csv(root+'advertising.csv')
```

In [6]:

```
df.head()
```

Out[6]:

	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

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

3. Data Analysis

In [7]:

```
df.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
```

Checking for duplicated values

In [8]:

```
df.duplicated().sum()
```

Out[8]:

```
0
```

- There are no duplicate values.

In [10]:

```
df.describe()
```

Out[10]:

	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

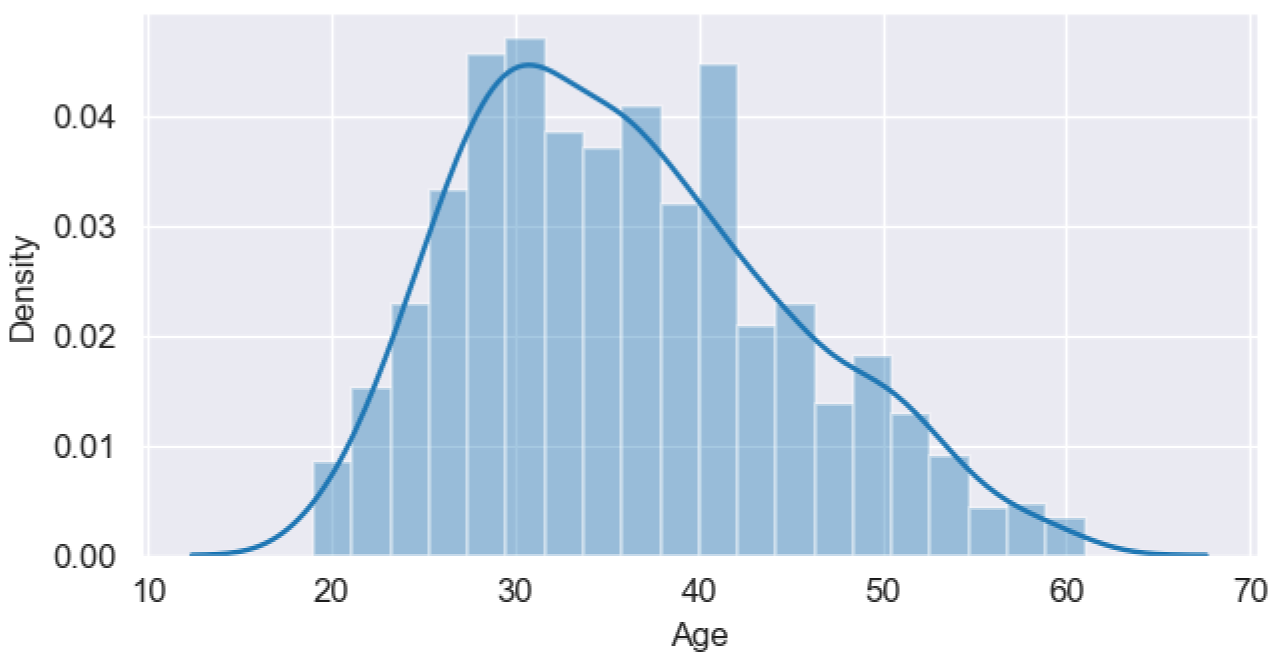
	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Male	Clicked on Ad
max	91.430000	61.000000	79484.800000	269.960000	1.000000	1.00000

Observations

- Interesting facts we can see from this table is that there are varied people who are engaging in the site. Like if we see the income feature, we can see that smallest income is dollar 13,996 and the highest is dollar 79,484. This means people are from different social groups. Also we are analyzing a popular website since the timeuser spend on the website in an average is 65 minutes and min time spent by them is 32 min and max time is 91 min in one seassion. These are huge numbers.
- Also, the average age of a visitor is 36 years. We see that the youngest user has 19 and the oldest is 61 years old. We can conclude that the site is targetting adult users. Finally, if we are wondering whether the site is visited more by men or women, we can see that the situation is almost equal (52% in favor of women).

What age group does the dataset majorly consist of?

In [18]:
```
plt.subplots(figsize=(6,3))
sns.distplot(df['Age'],bins = 20, kde=True)
plt.show()
```



In [20]:
```
df['Age'].skew()
```

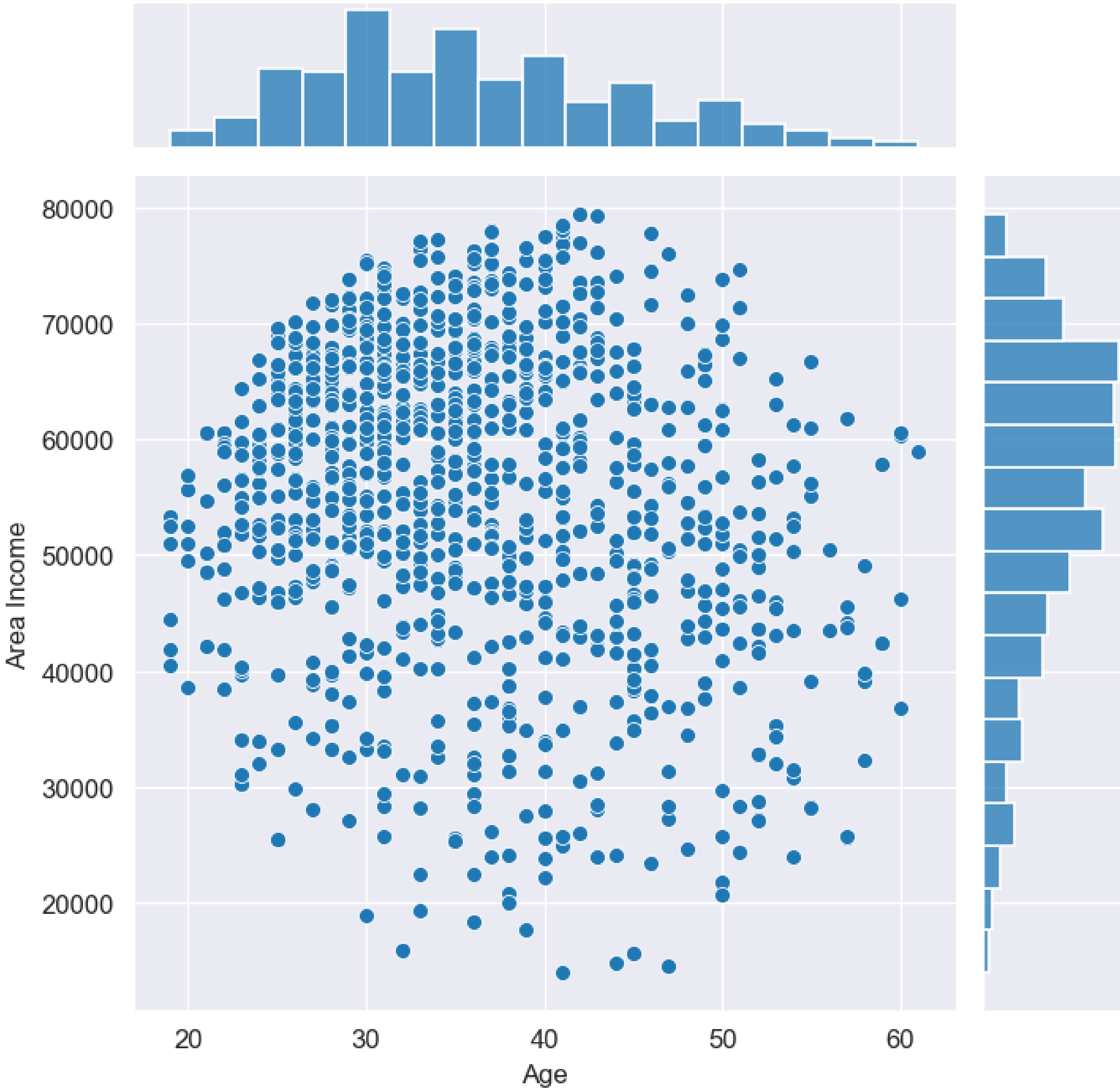
Out[20]: 0.4791416884125751

Observations

- Most of the customers are between 26-42 age.
- This age feature is almost normal distribution.

What is the income distribution in different age groups?

In [24]:
```
sns.jointplot(x='Age',y='Area Income',data=df,);
```

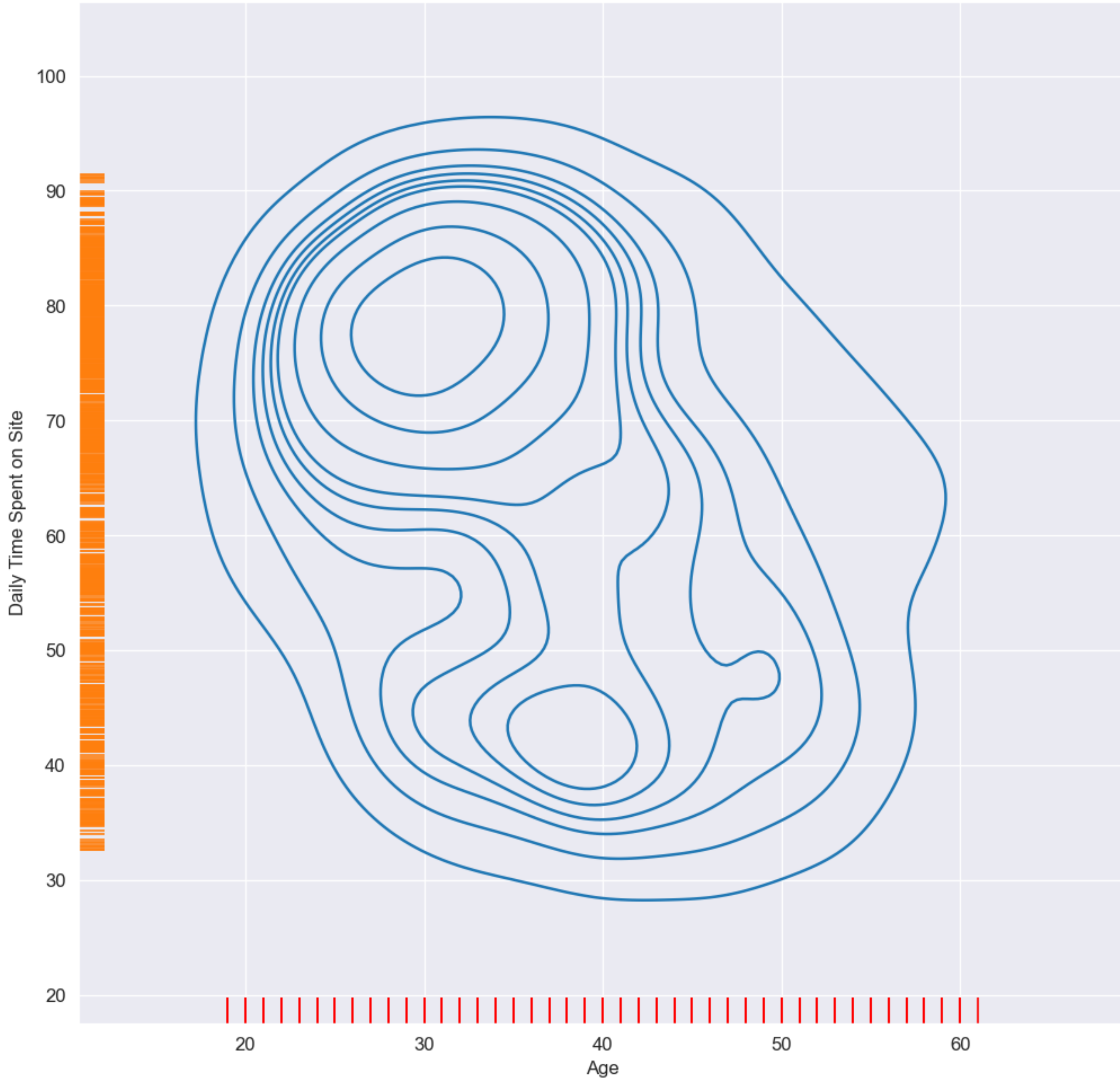


Observations

- Here, we can see that mostly teenagers are higher earners with age group of 20-40 earning 50k-70k.

Is there relation of age and the time they spent on website ?

In [31]:
```
f,ax = plt.subplots(figsize=(10,10))
sns.kdeplot(df['Age'],df['Daily Time Spent on Site'],ax=ax)
sns.rugplot(df['Age'],color='r',ax=ax)
sns.rugplot(df['Daily Time Spent on Site'],vertical=True,ax=ax)
plt.show()
```



Observation:

- From the graph, we can conclude that younger users spend more time on the site. This implies that users of the age between 20 and 40 years can be the main target group for the marketing campaign. Hypothetically, if we have a product intended for middle-aged people, this is the right site for advertising. Conversely, if we have a product intended for people over the age of 60, it would be a mistake to advertise on this site.

Max users are from whihc country and how much time they spend.

In [84]:

```
df['Country'].nunique()
```

Out[84]:

```
237
```

In [41]:

```
cont = df.groupby('Country')['Age'].count().reset_index().sort_values('Age',ascending=Fals
cont.rename(columns = {'Age':'Count'}, inplace = True)
cont.head(10)
```

Out[41]:

	Country	Count
70	France	9
54	Czech Republic	9
0	Afghanistan	8
12	Australia	8
216	Turkey	8
195	South Africa	8
187	Senegal	8
165	Peru	8
137	Micronesia	8
80	Greece	8

In [87]:

```
city = df.groupby('City')['Age'].count().reset_index().sort_values('Age',ascending=False)
city.rename(columns = {'Age':'Count'}, inplace = True)
city.head(10)
```

Out[87]:

	City	Count
426	Lisamouth	3
955	Williamsport	3
306	Johnstad	2
528	New Sheila	2
30	Benjaminchester	2
463	Millerbury	2
462	Michelleside	2
935	West Steven	2
390	Lake Jose	2
932	West Shannon	2

Observations:

- Max amount of traffic is coming from France and Czech Republic.
- We have already seen, there are 237 different unique countries in our dataset and no single country is too dominant. A large number of unique elements will not allow a machine learning model to establish easily valuable relationships. For that reason, this variable will be excluded too.Same is the case with city.

In [46]:

```
df.groupby('Country')['Daily Time Spent on Site','Clicked on Ad'].mean().reset_index().so
```

Out[46]:

	Country	Daily Time Spent on Site	Clicked on Ad
117	Lesotho	89.800000	0.000000
172	Reunion	88.150000	0.000000
192	Slovakia (Slovak Republic)	86.915000	0.000000

	Country	Daily Time Spent on Site	Clicked on Ad
79	Gibraltar	86.443333	0.000000
11	Aruba	86.410000	0.000000
198	Sri Lanka	82.450000	0.000000
148	Nepal	82.153333	0.000000
77	Germany	82.120000	1.000000
38	Cape Verde	81.750000	0.000000
127	Malaysia	81.496667	0.000000
23	Bermuda	80.940000	0.000000
24	Bhutan	80.600000	0.500000
36	Cameroon	79.014000	0.000000
113	Kyrgyz Republic	78.620000	0.166667
174	Russian Federation	78.493333	0.333333
143	Morocco	78.440000	0.333333
144	Mozambique	78.410000	0.000000
155	Niue	78.326667	0.000000
164	Paraguay	78.216667	0.333333
153	Nicaragua	78.170000	0.000000
211	Togo	77.986667	0.333333
39	Cayman Islands	77.624000	0.600000
210	Timor-Leste	77.438000	0.200000
146	Namibia	77.425000	0.500000
227	Uruguay	77.404000	0.200000
199	Sudan	77.355000	0.000000
51	Croatia	77.040000	0.000000
208	Tanzania	76.960000	0.333333
150	Netherlands Antilles	76.901667	0.333333
97	India	76.610000	0.000000
108	Kazakhstan	76.042500	0.500000
33	Burkina Faso	75.635000	0.250000
139	Monaco	75.313333	0.333333
14	Azerbaijan	75.220000	0.333333
10	Armenia	74.920000	0.333333
74	Gabon	74.661667	0.000000
30	British Virgin Islands	74.643333	0.333333
128	Maldives	74.610000	0.500000
145	Myanmar	74.422000	0.200000
25	Bolivia	74.335000	0.000000
162	Panama	74.120000	0.000000

	Country	Daily Time Spent on Site	Clicked on Ad
5	Angola	74.022500	0.250000
202	Swaziland	74.020000	0.000000
90	Haiti	73.980000	0.500000
125	Madagascar	73.931667	0.333333
191	Singapore	73.821667	0.166667
13	Austria	73.622000	0.200000
134	Mauritius	73.480000	0.250000
31	Brunei Darussalam	73.300000	0.400000
159	Pakistan	72.998000	0.200000
72	French Polynesia	72.944000	0.200000
175	Rwanda	72.396000	0.400000
147	Nauru	72.323333	0.333333
80	Greece	72.078750	0.375000
221	Ukraine	72.004000	0.200000
232	Wallis and Futuna	71.465000	0.250000
177	Saint Helena	71.442000	0.400000
203	Sweden	71.317500	0.250000
81	Greenland	71.300000	0.200000
56	Djibouti	71.260000	0.500000
26	Bosnia and Herzegovina	71.197143	0.428571
217	Turkmenistan	71.003333	0.333333
138	Moldova	71.000000	0.333333
19	Belarus	70.316667	0.500000
194	Somalia	70.096000	0.400000
107	Jordan	70.040000	0.000000
48	Cook Islands	69.963333	0.333333
68	Fiji	69.837143	0.428571
129	Mali	69.835000	0.250000
49	Costa Rica	69.763333	0.333333
103	Italy	69.510000	0.200000
34	Burundi	69.257143	0.285714
122	Luxembourg	69.195714	0.428571
225	United States Virgin Islands	69.142500	0.500000
176	Saint Barthelemy	69.085000	1.000000
20	Belgium	68.892000	0.400000
101	Isle of Man	68.646667	0.333333
18	Barbados	68.450000	0.400000
73	French Southern Territories	68.158000	0.200000

	Country	Daily Time Spent on Site	Clicked on Ad
213	Tonga	68.060000	0.400000
32	Bulgaria	67.736667	0.666667
58	Dominican Republic	67.722500	0.500000
156	Norfolk Island	67.696000	0.400000
205	Syrian Arab Republic	67.673333	0.333333
189	Seychelles	67.570000	0.333333
133	Mauritania	67.485000	0.500000
104	Jamaica	67.434000	0.400000
57	Dominica	67.394000	0.400000
102	Israel	67.382500	0.500000
95	Hungary	66.920000	0.833333
229	Vanuatu	66.896667	0.166667
222	United Arab Emirates	66.866667	0.500000
234	Yemen	66.850000	0.666667
171	Qatar	66.848333	0.333333
169	Portugal	66.716667	0.333333
64	Estonia	66.523333	0.333333
35	Cambodia	66.487143	0.285714
206	Taiwan	66.452857	0.571429
112	Kuwait	66.305000	0.500000
84	Guam	66.150000	0.500000
2	Algeria	66.011667	0.500000
106	Jersey	65.945000	0.666667
69	Finland	65.926000	0.200000
154	Niger	65.756667	0.666667
141	Montenegro	65.715000	1.000000
78	Ghana	65.642500	0.500000
230	Venezuela	65.580000	0.428571
187	Senegal	65.398750	0.625000
83	Guadeloupe	65.335000	0.500000
181	Saint Pierre and Miquelon	65.156000	0.600000
93	Honduras	65.080000	0.400000
9	Argentina	65.025000	0.500000
183	Samoa	65.000000	0.666667
41	Chad	64.897500	0.500000
186	Saudi Arabia	64.892500	0.750000
0	Afghanistan	64.782500	0.625000
105	Japan	64.775000	0.500000

	Country	Daily Time Spent on Site	Clicked on Ad
53	Cyprus	64.697500	0.500000
76	Georgia	64.527500	0.500000
27	Bouvet Island (Bouvetoya)	64.494000	0.400000
116	Lebanon	64.436667	0.666667
126	Malawi	64.260000	0.500000
87	Guinea	64.133333	0.666667
37	Canada	64.108000	0.600000
3	American Samoa	63.810000	0.600000
111	Korea	63.710000	0.600000
166	Philippines	63.621667	0.500000
66	Falkland Islands (Malvinas)	63.575000	0.500000
180	Saint Martin	63.440000	0.500000
70	France	63.431111	0.555556
47	Congo	63.390000	0.750000
1	Albania	63.371429	0.571429
197	Spain	63.330000	1.000000
98	Indonesia	63.135000	0.666667
226	United States of America	63.096000	0.600000
100	Ireland	63.083333	0.333333
92	Holy See (Vatican City State)	63.023333	0.333333
215	Tunisia	62.942500	0.250000
130	Malta	62.876667	0.500000
12	Australia	62.836250	0.875000
182	Saint Vincent and the Grenadines	62.820000	0.500000
236	Zimbabwe	62.685000	0.666667
223	United Kingdom	62.636667	0.666667
184	San Marino	62.566667	0.333333
170	Puerto Rico	62.490000	0.500000
59	Ecuador	62.376000	0.400000
55	Denmark	62.340000	0.666667
60	Egypt	62.192000	0.600000
17	Bangladesh	62.072500	0.500000
94	Hong Kong	62.043333	0.666667
28	Brazil	62.008000	0.600000
163	Papua New Guinea	61.918000	0.600000
219	Tuvalu	61.912500	0.750000
61	El Salvador	61.795000	0.666667
15	Bahamas	61.691429	0.571429

	Country	Daily Time Spent on Site	Clicked on Ad
43	China	61.645000	0.666667
54	Czech Republic	61.534444	0.444444
136	Mexico	61.405000	0.666667
96	Iceland	61.386667	0.333333
44	Christmas Island	61.076667	0.666667
99	Iran	61.020000	0.600000
235	Zambia	60.720000	0.750000
209	Thailand	60.682500	0.500000
218	Turks and Caicos Islands	60.574000	0.600000
168	Poland	60.503333	0.500000
91	Heard Island and McDonald Islands	60.170000	0.666667
233	Western Sahara	60.132857	0.571429
179	Saint Lucia	59.965000	0.500000
204	Switzerland	59.875000	0.750000
188	Serbia	59.778000	0.600000
119	Libyan Arab Jamahiriya	59.770000	0.500000
21	Belize	59.630000	0.600000
196	South Georgia and the South Sandwich Islands	59.595000	0.500000
195	South Africa	59.587500	0.750000
137	Micronesia	59.326250	0.500000
46	Comoros	59.315000	0.500000
193	Slovenia	58.950000	1.000000
63	Eritrea	58.918571	0.428571
16	Bahrain	58.852000	0.400000
124	Macedonia	58.680000	0.500000
75	Gambia	58.615000	0.500000
114	Lao People's Democratic Republic	58.585000	0.500000
207	Tajikistan	58.560000	0.666667
40	Central African Republic	58.520000	0.500000
200	Suriname	58.165000	0.500000
152	New Zealand	57.962500	0.500000
201	Svalbard & Jan Mayen Islands	57.598333	0.666667
216	Turkey	57.587500	0.875000
6	Anguilla	57.508333	0.500000
42	Chile	57.270000	0.750000
115	Latvia	57.230000	1.000000
228	Uzbekistan	57.165000	0.500000
120	Liechtenstein	57.138333	1.000000

	Country	Daily Time Spent on Site	Clicked on Ad
224	United States Minor Outlying Islands	57.040000	0.500000
50	Cote d'Ivoire	57.027500	0.750000
190	Sierra Leone	56.990000	1.000000
67	Faroe Islands	56.866667	0.666667
52	Cuba	56.818000	0.800000
62	Equatorial Guinea	56.752500	0.750000
71	French Guiana	56.655000	0.750000
142	Montserrat	56.640000	1.000000
149	Netherlands	56.607500	0.750000
82	Grenada	56.497500	0.500000
89	Guyana	56.426000	0.600000
160	Palau	56.202500	0.500000
165	Peru	56.148750	0.625000
157	Northern Mariana Islands	56.100000	0.666667
45	Colombia	56.015000	0.500000
86	Guernsey	55.383333	0.666667
22	Benin	55.310000	0.500000
132	Martinique	55.192500	0.750000
231	Vietnam	54.903333	0.666667
29	British Indian Ocean Territory (Chagos Archipe...	54.700000	1.000000
118	Liberia	54.488750	0.750000
85	Guatemala	54.432500	0.750000
140	Mongolia	53.646667	0.666667
212	Tokelau	53.625000	0.750000
109	Kenya	53.535000	1.000000
8	Antigua and Barbuda	53.474000	0.800000
151	New Caledonia	53.410000	1.000000
7	Antarctica (the territory South of 60 deg S)	53.180000	0.666667
220	Uganda	53.160000	1.000000
167	Pitcairn Islands	53.070000	0.500000
161	Palestinian Territory	52.910000	0.666667
88	Guinea-Bissau	50.870000	0.500000
178	Saint Kitts and Nevis	50.520000	1.000000
123	Macao	50.270000	1.000000
158	Norway	50.205000	0.500000
173	Romania	49.990000	1.000000
4	Andorra	49.805000	1.000000
214	Trinidad and Tobago	48.723333	0.666667

	Country	Daily Time Spent on Site	Clicked on Ad
65	Ethiopia	47.487143	1.000000
135	Mayotte	46.456667	0.833333
131	Marshall Islands	43.160000	1.000000
185	Sao Tome and Principe	42.320000	1.000000
121	Lithuania	42.073333	1.000000
110	Kiribati	36.370000	1.000000

Observations:

- The max traffic is coming from France and Czech Republic, but the average time spent by each users in country Lesotho,Reunion, Slovakia (Slovak Republic),Gibraltar,Aruba is more. Hence we need to analyze those customers and then target htose type of customers more.

In [53]:

```
df.groupby(['Clicked on Ad'])['Daily Time Spent on Site', 'Age', 'Area Income',
                               'Daily Internet Usage'].mean()
```

Out[53]:

Clicked on Ad	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage
0	76.85462	31.684	61385.58642	214.51374
1	53.14578	40.334	48614.41374	145.48646

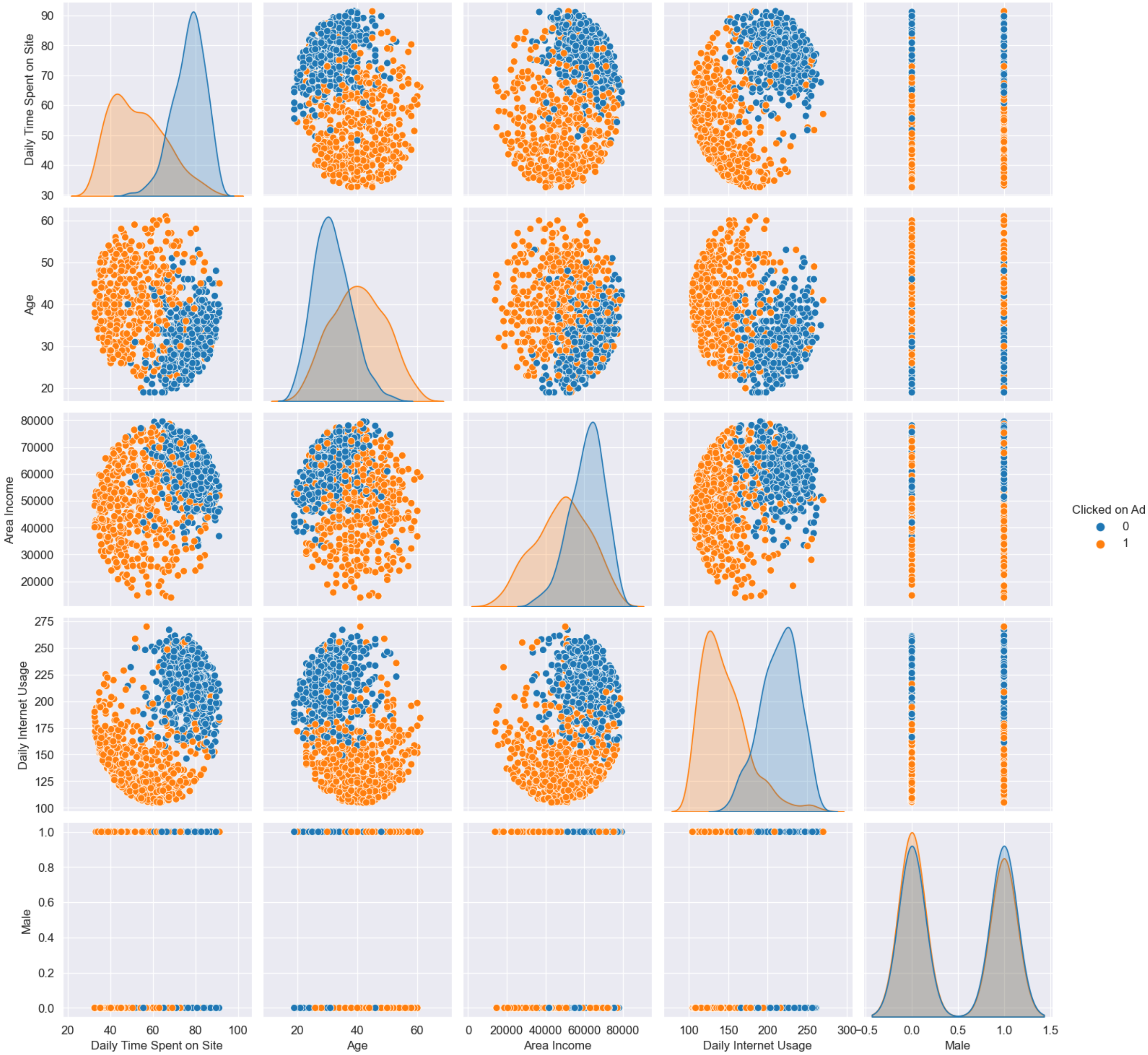
Observations:

- Female spend more time on an average on the website.

What is the relationship between different features?

In [56]:

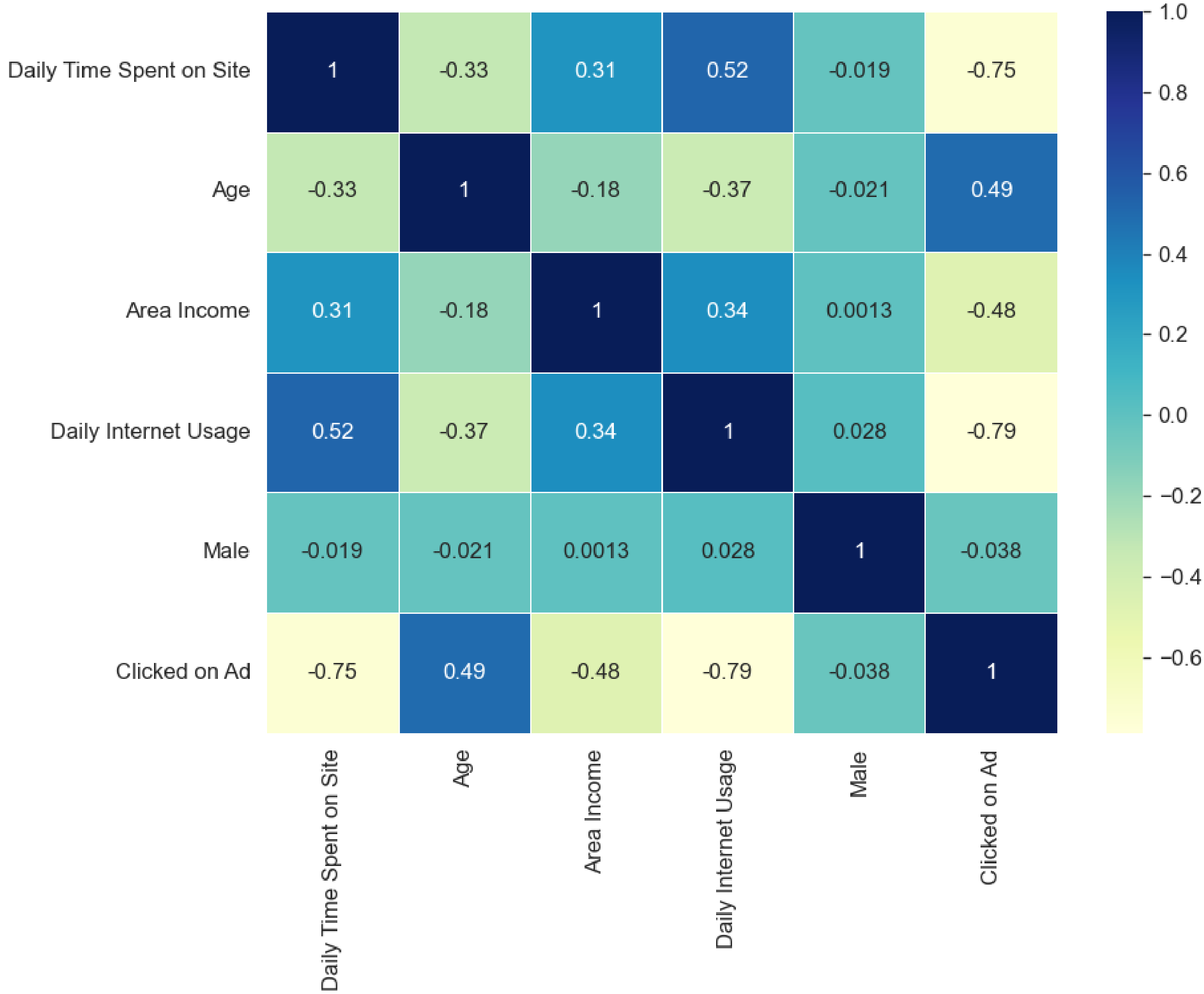
```
sns.pairplot(df,hue='Clicked on Ad')
plt.show()
```



Knowing the correlation of different features with Clicked on Ad

In [61]:
```
sns.heatmap(df.corr(),annot=True,linewidths=0.5,cmap="YlGnBu")
```

Out[61]: <AxesSubplot:>



Observations:

- We can see that there is a positive correatin of Clicked on Ads and age i.e when age increases Clicked on Ads also increases. This means old people click on ads more.
- We can see that there is negative correlation between Clicked on Ads and features like- Daily Time Spent on Site,Area Income,Daily Internet Usage.
- There is no relation of Clicked on Ads and Gender.

time

In [63]:

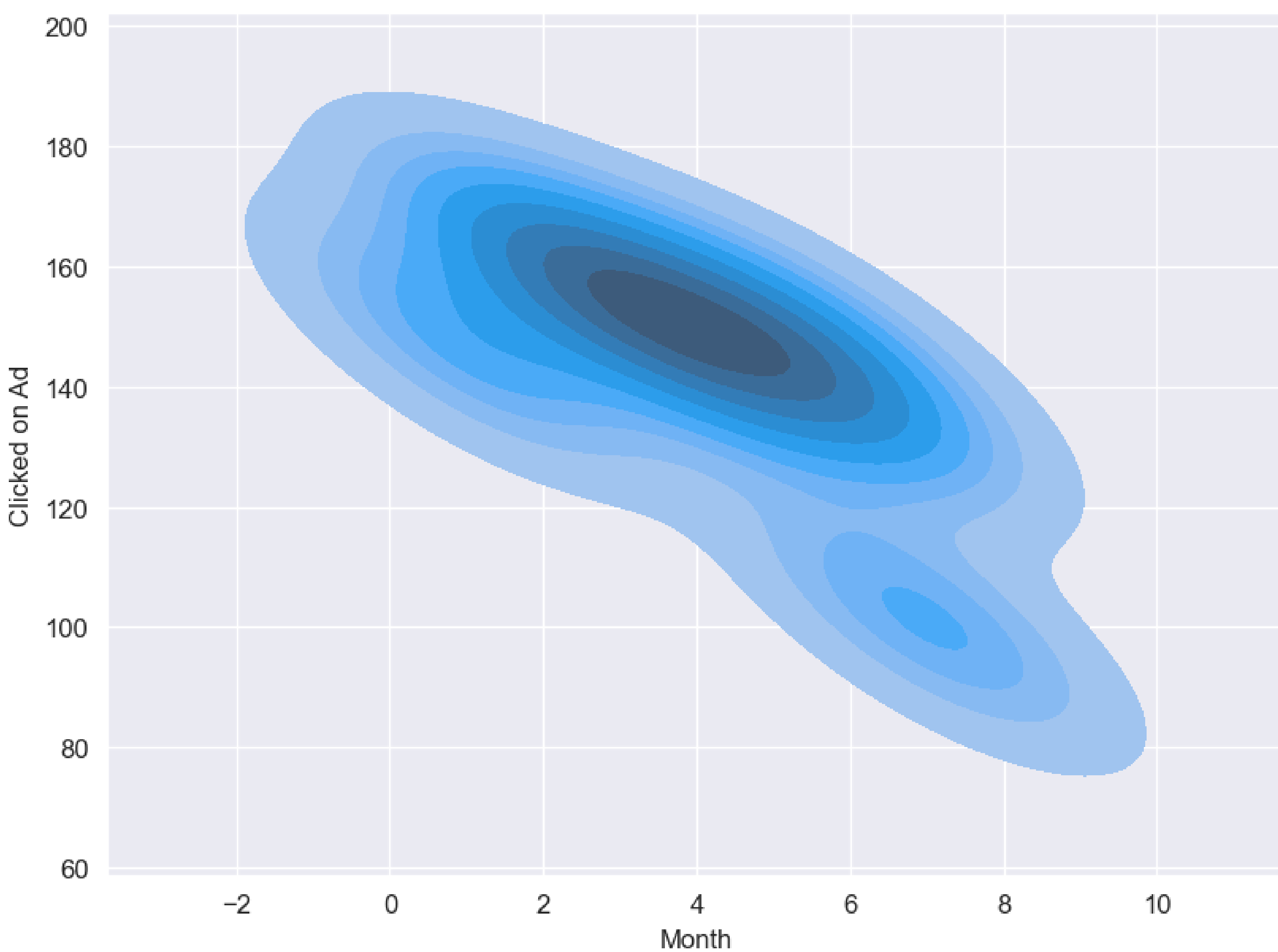
```
df['Timestamp'] = pd.to_datetime(df['Timestamp'])

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

df.head()
```

Out[63]:

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Clicked on Ad	Month	Day of the month	Da o th wee
0	68.95	35	61833.90	256.09	Cloned 5thgeneration orchestration	Wrightburgh	0	Tunisia	0	3	27	(
1	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	0	4	4	(
2	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	0	3	13	(
3	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	0	1	10	(
4	68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	0	6	3	

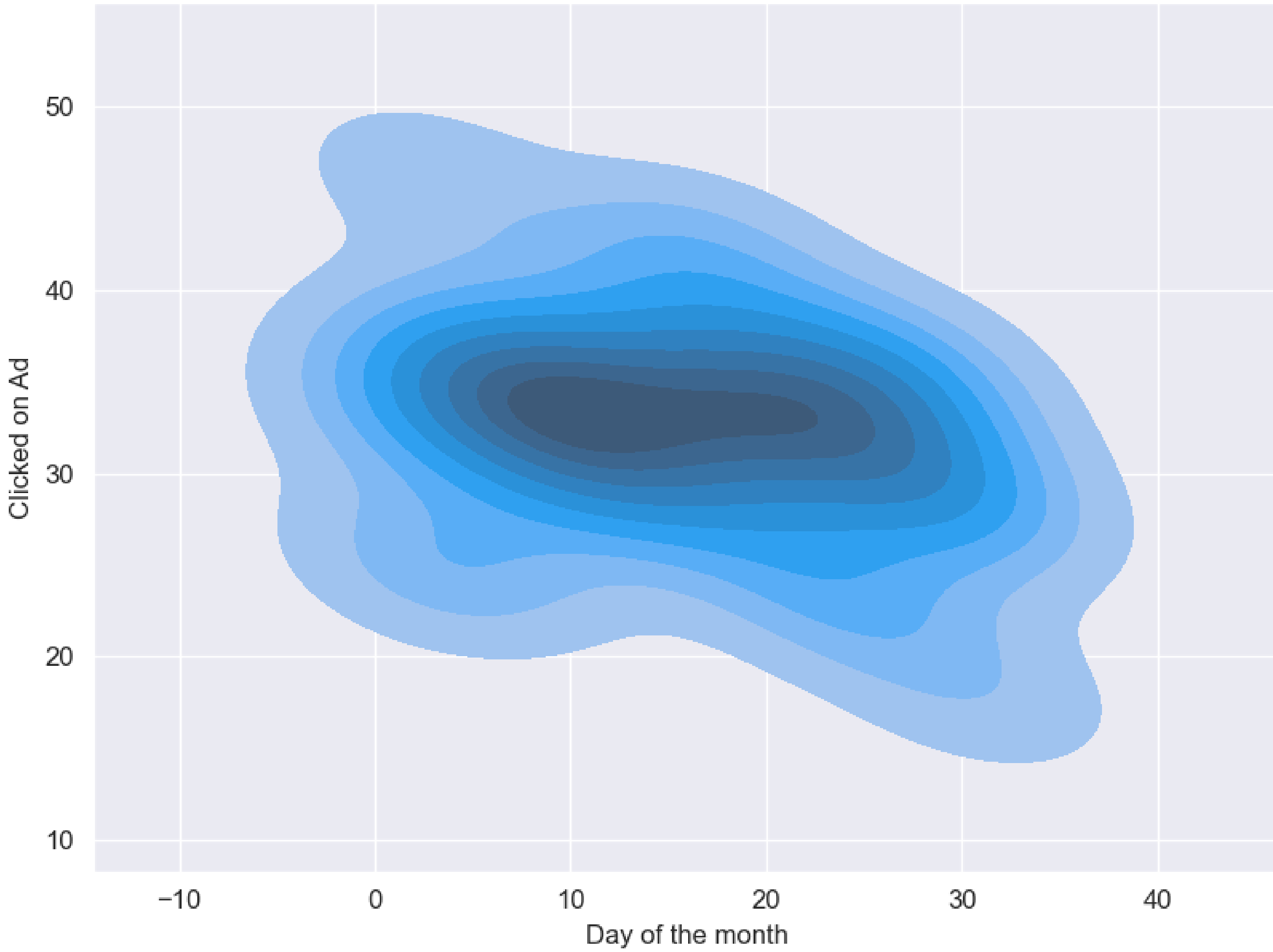
In [83]:

```
month = df.groupby('Month')['Clicked on Ad'].count().reset_index()
sns.kdeplot(x='Month',y='Clicked on Ad',data=month,shade=True);
```



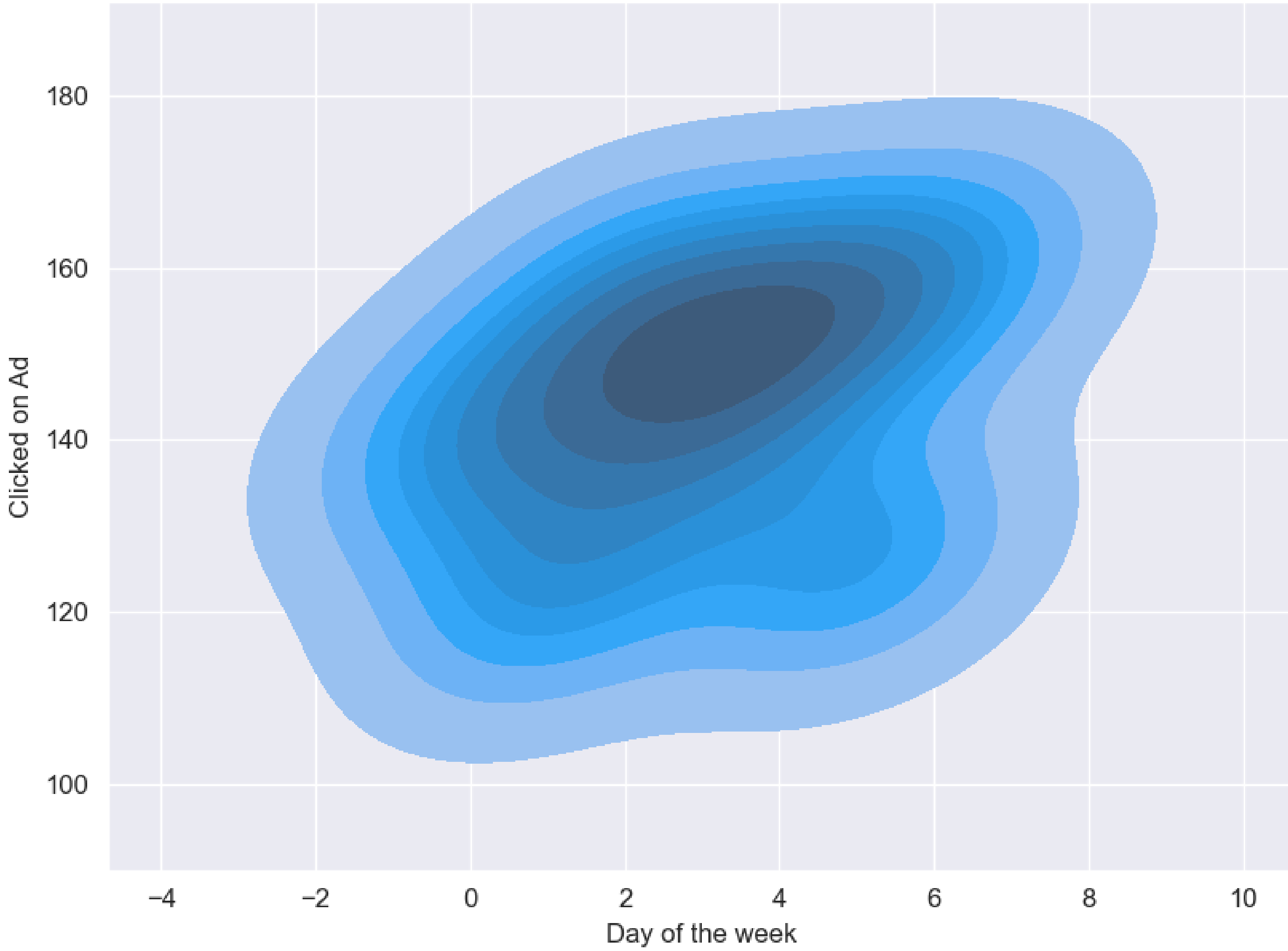
In [82]:

```
day_month = df.groupby('Day of the month')['Clicked on Ad'].count().reset_index()
sns.kdeplot(x='Day of the month',y='Clicked on Ad',data=day_month,shade=True);
```



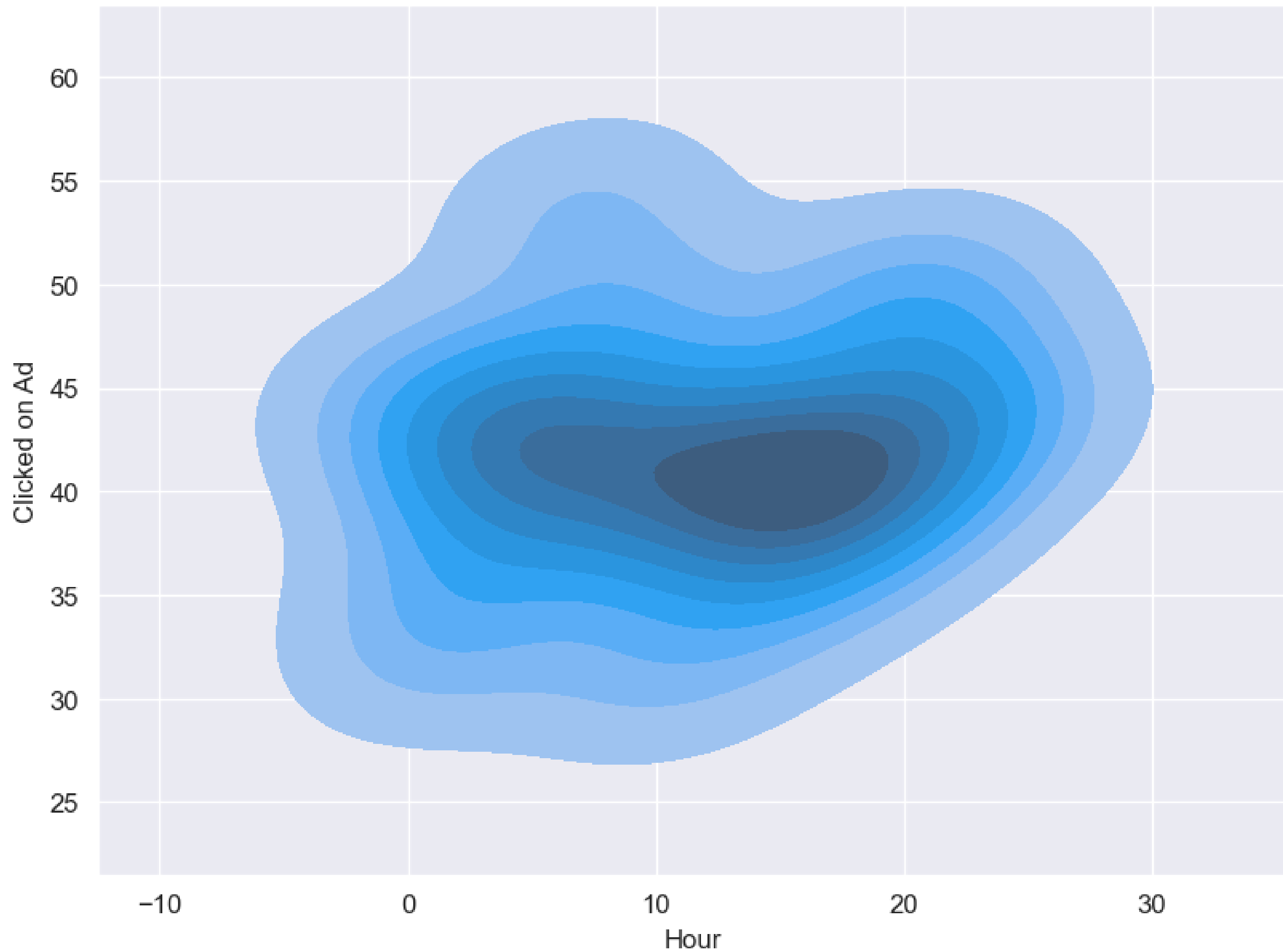
In [81]:

```
day_week = df.groupby('Day of the week')['Clicked on Ad'].count().reset_index()
sns.kdeplot(x='Day of the week',y='Clicked on Ad',data=day_week,shade=True);
```



In [80]:

```
hour = df.groupby('Hour')['Clicked on Ad'].count().reset_index()
sns.kdeplot(x='Hour',y='Clicked on Ad',data=hour,shade=True);
```



4. Data Preprocessing

In [86]:

```
df.head()
```

Out[86]:

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Clicked on Ad	Month	Day of the month	Da o th wee
0	68.95	35	61833.90	256.09	Cloned 5thgeneration orchestration	Wrightburgh	0	Tunisia	0	3	27	[illegible]
1	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	0	4	4	[illegible]
2	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	0	3	13	[illegible]
3	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	0	1	10	[illegible]
4	68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	0	6	3	[illegible]

In [88]:
```
df = df.drop(['Ad Topic Line', 'City', 'Country'], axis=1)
```

In [89]:
```
df.head()
```

Out[89]:

	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

5. Model Development

In [121…
```
X = df.drop('Clicked on Ad',axis=1)
y = df['Clicked on Ad']
```

In [122…
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.33,random_state=42)
```

Logistic Regression

In [108…
```
lr_clf = LogisticRegression()

parameters = {'solver':['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']}
lr_model = GridSearchCV(estimator=lr_clf,param_grid=parameters,cv=5,scoring='accuracy')
lr_model.fit(X_train,y_train)
```

Out[108…
```
GridSearchCV(cv=5, estimator=LogisticRegression(),
             param_grid={'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag',
                                    'saga']},
             scoring='accuracy')
```

In [110…
```
print('Best Parameter is:',lr_model.best_params_)
print('Accuracy:',lr_model.best_score_)
```

```
Best Parameter is: {'solver': 'newton-cg'}
Accuracy: 0.9701492537313434
```

Naive Bayes

In [130…
```
nb_clf = GaussianNB()
nb_clf.fit(X_train,y_train)
nb_pred = nb_clf.predict(X_test).reshape(-1,1)
nav_bayes_accuracy = accuracy_score(y_test,nb_pred)
print('Accuracy:',nav_bayes_accuracy*100)
```

```
Accuracy: 95.75757575757575
```

Random Forest

In [132…
```
rf_clf = RandomForestClassifier()
```

```
parameters = {'criterion':['gini', 'entropy', 'log_loss']}
rf_model = GridSearchCV(estimator=rf_clf,param_grid=parameters,cv=5,scoring='accuracy')
rf_model.fit(X_train,y_train)
```

Out[132…

```
GridSearchCV(cv=5, estimator=RandomForestClassifier(),
             param_grid={'criterion': ['gini', 'entropy', 'log_loss']},
             scoring='accuracy')
```

In [133…

```
print('Best Parameter is:',rf_model.best_params_)
print('Accuracy:',rf_model.best_score_)
```

```
Best Parameter is: {'criterion': 'gini'}
Accuracy: 0.9656716417910449
```

Decision Tree

In [135…

```
dr_clf = DecisionTreeClassifier()

parameters = {'criterion':['gini', 'entropy', 'log_loss'],
             'splitter':['best', 'random']}
dr_model = GridSearchCV(estimator=dr_clf,param_grid=parameters,cv=5,scoring='accuracy')
dr_model.fit(X_train,y_train)
```

Out[135…

```
GridSearchCV(cv=5, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy', 'log_loss'],
                         'splitter': ['best', 'random']},
             scoring='accuracy')
```

In [136…

```
print('Best Parameter is:',dr_model.best_params_)
print('Accuracy:',dr_model.best_score_)
```

```
Best Parameter is: {'criterion': 'gini', 'splitter': 'best'}
Accuracy: 0.946268656716418
```

6. Conclusion

- Comparing all the above implementation models, we conclude that Logistic Regression gives us the maximum accuracy for determining the click probability. We believe in future there will be fewer ads, but they will be more relevant. And also these ads will cost more and will be worth it. Our model can basically predict if a certain customer will see the ad or not and it will save us huge amount of money as it will tell us before hand only whether to invest on that customer or not.

In [ ]:

```

```