Project- Ad Click Prediction

Group Members:

Bojja Pranitha

Goal of the Project:

Goal of the project is to predict if a particular user is likely to click on particular ad or not based on his feature.

Problem statement:

Publicizing is a multi-billion-dollar industry that goes about as a scaffold among organizations and their clients. While the vast majority are aware of the promotions around them, they represent the intensity of those advertisements and the impact of publicizing all in all. Research proposes that basically making somebody mindful of items, occasions, and brands expands the chances of that individual really purchasing those items, going to those occasions, or supporting those brands. Further, if an advertisement catches a man's thoughtfulness regarding the degree that he or she has a prompt, positive response to it, those chances of direct item commitment spikes significantly.

In this project, we are going to work on an advertising dataset, indicating whether or not a particular internet user has clicked on an Advertisement.

The goal is to predict if a user would click on an advertisement based on the features of the user. Few assumptions made as a part of this project is:

1. User taken into consideration are between the age group of 19 to 61.
2. There is almost equal ratio of male and female internet users.
3. The ad topic is limited to what is given in the dataset.

Challenges Faced:

We are highly motivated to work on Ad click prediction dataset, few challenges in this area of study is:

1. There is very less publicly available data set for ad click.
2. New online ads that are coming up is not targeted to a particular set of users, using our prediction algorithm study companies will be able to target it to particular set of users.

Data Set:

The dataset consists of below features:

Daily Time Spent on Site: consumer time on site in minutes

Age: Customer age in years

Area Income: Avg. Income of geographical area of consumer

Daily Internet Usage: Avg. minutes a day consumer is on the internet

Ad Topic Line: Headline of the advertisement

City: City of consumer

Male: Whether or not consumer was male

Country: Country of consumer

We have done preliminary EDA(mention below) and we are planning to do more insight as we progress.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
dataframe = pd.read_csv("C:/Users/richa/Desktop/Ad-Click-Prediction-master/advertising.csv")
```

```
dataframe
```

```
     Daily Time Spent on Site  Age  Area Income  Daily Internet Usage  \
0                       68.95   35     61833.90                256.09
1                       80.23   31     68441.85                193.77
2                       69.47   26     59785.94                236.50
3                       74.15   29     54806.18                245.89
4                       68.37   35     73889.99                225.58
5                       59.99   23     59761.56                226.74
6                       88.91   33     53852.85                208.36
7                       66.00   48     24593.33                131.76
8                       74.53   30     68862.00                221.51
9                       69.88   20     55642.32                183.82
10                      47.64   49     45632.51                122.02
11                      83.07   37     62491.01                230.87
12                      69.57   48     51636.92                113.12
13                      79.52   24     51739.63                214.23
14                      42.95   33     30976.00                143.56
15                      63.45   23     52182.23                140.64
16                      55.39   37     23936.86                129.41
17                      82.03   41     71511.08                187.53
18                      54.70   36     31087.54                118.39
19                      74.58   40     23821.72                135.51
20                      77.22   30     64802.33                224.44
21                      84.59   35     60015.57                226.54
22                      41.49   52     32635.70                164.83
23                      87.29   36     61628.72                209.93
24                      41.39   41     68962.32                167.22
```

25	78.74	28	64828.00	204.79
26	48.53	28	38067.08	134.14
27	51.95	52	58295.82	129.23
28	70.20	34	32708.94	119.20
29	76.02	22	46179.97	209.82
..	...	...	...	...
970	40.18	29	50760.23	151.96
971	45.17	48	34418.09	132.07
972	50.48	50	20592.99	162.43
973	80.87	28	63528.80	203.30
974	41.88	40	44217.68	126.11
975	39.87	48	47929.83	139.34
976	61.84	45	46024.29	105.63
977	54.97	31	51900.03	116.38
978	71.40	30	72188.90	166.31
979	70.29	31	56974.51	254.65
980	67.26	57	25682.65	168.41
981	76.58	46	41884.64	258.26
982	54.37	38	72196.29	140.77
983	82.79	32	54429.17	234.81
984	66.47	31	58037.66	256.39
985	72.88	44	64011.26	125.12
986	76.44	28	59967.19	232.68
987	63.37	43	43155.19	105.04
988	89.71	48	51501.38	204.40
989	70.96	31	55187.85	256.40
990	35.79	44	33813.08	165.62
991	38.96	38	36497.22	140.67
992	69.17	40	66193.81	123.62
993	64.20	27	66200.96	227.63
994	43.70	28	63126.96	173.01
995	72.97	30	71384.57	208.58
996	51.30	45	67782.17	134.42
997	51.63	51	42415.72	120.37
998	55.55	19	41920.79	187.95
999	45.01	26	29875.80	178.35

	Ad Topic Line	City	Male
0	Cloned 5thgeneration orchestration	Wrightburgh	0
1	Monitored national standardization	West Jodi	1
2	Organic bottom-line service-desk	Davidton	0
3	Triple-buffered reciprocal time-frame	West Terrifurt	1
4	Robust logistical utilization	South Manuel	0
5	Sharable client-driven software	Jamieberg	1
6	Enhanced dedicated support	Brandonstad	0
7	Reactive local challenge	Port Jefferybury	1
8	Configurable coherent function	West Colin	1
9	Mandatory homogeneous architecture	Ramirezton	1
10	Centralized neutral neural-net	West Brandonton	0

11	Team-oriented grid-enabled Local Area Network	East Theresashire	1
12	Centralized content-based focus group	West Katiefurt	1
13	Synergistic fresh-thinking array	North Tara	0
14	Grass-roots coherent extranet	West William	0
15	Persistent demand-driven interface	New Travistown	1
16	Customizable multi-tasking website	West Dylanberg	0
17	Intuitive dynamic attitude	Pruittmouth	0
18	Grass-roots solution-oriented conglomeration	Jessicastad	1
19	Advanced 24/7 productivity	Millertown	1
20	Object-based reciprocal knowledgebase	Port Jacqueline	1
21	Streamlined non-volatile analyzer	Lake Nicole	1
22	Mandatory disintermediate utilization	South John	0
23	Future-proofed methodical protocol	Pamelamouth	1
24	Exclusive neutral parallelism	Harperborough	0
25	Public-key foreground groupware	Port Danielleberg	1
26	Ameliorated client-driven forecast	West Jeremyside	1
27	Monitored systematic hierarchy	South Cathyfurt	0
28	Open-architected impactful productivity	Palmerside	0
29	Business-focused value-added definition	West Guybury	0
..	...	...	...
970	Enhanced intangible portal	Lake Tracy	0
971	Down-sized background groupware	Taylormouth	1
972	Switchable real-time product	Dianaville	0
973	Ameliorated local workforce	Collinsburgh	0
974	Streamlined exuding adapter	Port Rachel	1
975	Business-focused user-facing benchmark	South Rebecca	1
976	Reactive bi-directional standardization	Port Joshuafort	1
977	Virtual bifurcated portal	Robinsontown	1
978	Integrated 3rdgeneration monitoring	Beckton	0
979	Balanced responsive open system	New Frankshire	1
980	Focused incremental Graphic Interface	North Derekville	1
981	Secured 24hour policy	West Sydney	0
982	Up-sized asymmetric firmware	Lake Matthew	0
983	Distributed fault-tolerant service-desk	Lake Zacharyfurt	1
984	Vision-oriented human-resource synergy	Lindsaymouth	1
985	Customer-focused explicit challenge	Sarahland	0
986	Synchronized human-resource moderator	Port Julie	1
987	Open-architected full-range projection	Michaelshire	1
988	Versatile local forecast	Sarafurt	1
989	Ameliorated user-facing help-desk	South Denise	0
990	Enterprise-wide tangible model	North Katie	1
991	Versatile mission-critical application	Mauricefurt	1
992	Extended leadingedge solution	New Patrick	0
993	Phased zero tolerance extranet	Edwardsmouth	1
994	Front-line bifurcated ability	Nicholasland	0
995	Fundamental modular algorithm	Duffystad	1
996	Grass-roots cohesive monitoring	New Darlene	1
997	Expanded intangible solution	South Jessica	1
998	Proactive bandwidth-monitored policy	West Steven	0
999	Virtual 5thgeneration emulation	Ronniemouth	0

	Country	Timestamp
0	Tunisia	2016-03-27 00:53:11
1	Nauru	2016-04-04 01:39:02
2	San Marino	2016-03-13 20:35:42
3	Italy	2016-01-10 02:31:19
4	Iceland	2016-06-03 03:36:18
5	Norway	2016-05-19 14:30:17
6	Myanmar	2016-01-28 20:59:32
7	Australia	2016-03-07 01:40:15
8	Grenada	2016-04-18 09:33:42
9	Ghana	2016-07-11 01:42:51
10	Qatar	2016-03-16 20:19:01
11	Burundi	2016-05-08 08:10:10
12	Egypt	2016-06-03 01:14:41
13	Bosnia and Herzegovina	2016-04-20 21:49:22
14	Barbados	2016-03-24 09:31:49
15	Spain	2016-03-09 03:41:30
16	Palestinian Territory	2016-01-30 19:20:41
17	Afghanistan	2016-05-02 07:00:58
18	British Indian Ocean Territory (Chagos Archipe...	2016-02-13 07:53:55
19	Russian Federation	2016-02-27 04:43:07
20	Cameroon	2016-01-05 07:52:48
21	Cameroon	2016-03-18 13:22:35
22	Burundi	2016-05-20 08:49:33
23	Korea	2016-03-23 09:43:43
24	Tokelau	2016-06-13 17:27:09
25	Monaco	2016-05-27 15:25:52
26	Tuvalu	2016-02-08 10:46:14
27	Greece	2016-07-19 08:32:10
28	British Virgin Islands	2016-04-14 05:08:35
29	Bouvet Island (Bouvetoya)	2016-01-27 12:38:16
..	...	...
970	Hong Kong	2016-06-25 04:21:33
971	Palau	2016-01-27 14:41:10
972	Malawi	2016-05-16 18:51:59
973	Uruguay	2016-02-27 20:20:25
974	Cyprus	2016-02-28 23:54:44
975	Mexico	2016-06-13 06:11:33
976	Niger	2016-05-05 11:07:13
977	France	2016-07-07 12:17:33
978	Japan	2016-05-24 17:07:08
979	Norfolk Island	2016-03-30 14:36:55
980	Bulgaria	2016-05-27 05:54:03
981	Uzbekistan	2016-01-03 16:30:51
982	Mexico	2016-06-25 18:17:53
983	Brunei Darussalam	2016-02-24 10:36:43
984	France	2016-03-03 03:13:48
985	Yemen	2016-04-21 19:56:24

986	Northern Mariana Islands	2016-04-06	17:26:37
987	Poland	2016-03-23	12:53:23
988	Bahrain	2016-02-17	07:00:38
989	Saint Pierre and Miquelon	2016-06-26	07:01:47
990	Tonga	2016-04-20	13:36:42
991	Comoros	2016-07-21	16:02:40
992	Montenegro	2016-03-06	11:36:06
993	Isle of Man	2016-02-11	23:45:01
994	Mayotte	2016-04-04	03:57:48
995	Lebanon	2016-02-11	21:49:00
996	Bosnia and Herzegovina	2016-04-22	02:07:01
997	Mongolia	2016-02-01	17:24:57
998	Guatemala	2016-03-24	02:35:54
999	Brazil	2016-06-03	21:43:21

	Clicked on Ad	City Codes	Country Codes	Month	Hour
0	0	961	215	03	00
1	0	903	147	04	01
2	0	111	184	03	20
3	0	939	103	01	02
4	0	805	96	06	03
5	0	282	158	05	14
6	0	46	145	01	20
7	1	671	12	03	01
8	0	884	82	04	09
9	0	712	78	07	01
10	1	878	171	03	20
11	0	180	34	05	08
12	1	907	60	06	01
13	0	606	26	04	21
14	1	944	18	03	09
15	1	541	197	03	03
16	1	892	161	01	19
17	0	707	0	05	07
18	1	298	29	02	07
19	1	468	174	02	04
20	0	666	36	01	07
21	0	404	36	03	13
22	1	799	34	05	08
23	0	617	111	03	09
24	1	242	212	06	17
25	0	653	139	05	15
26	1	901	219	02	10
27	1	779	80	07	08
28	1	616	30	04	05
29	0	898	27	01	12
..	...	...	...	...	...
970	1	410	94	06	04
971	1	832	160	01	14
972	1	123	126	05	18

```
973               0          94             227    02   20
974               1         694              53    02   23
975               1         813             136    06   06
976               1         677             154    05   11
977               1         739              70    07   12
978               0          29             105    05   17
979               0         492             156    03   14
980               1         567              32    05   05
981               0         936             228    01   16
982               1         397             136    06   18
983               0         412              31    02   10
984               0         423              70    03   03
985               1         759             234    04   19
986               0         679             157    04   17
987               1         459             168    03   12
988               0         758              16    02   07
989               0         787             181    06   07
990               1         580             213    04   13
991               1         444              46    07   16
992               1         520             141    03   11
993               0         191             101    02   23
994               1         547             135    04   03
995               1         126             116    02   21
996               1         488              26    04   02
997               1         798             140    02   17
998               0         935              85    03   02
999               1         744              28    06   21

[1000 rows x 14 columns]
```

```
dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
Daily Time Spent on Site   1000 non-null float64
Age                        1000 non-null int64
Area Income                 1000 non-null float64
Daily Internet Usage       1000 non-null float64
Ad Topic Line              1000 non-null object
City                       1000 non-null object
Male                       1000 non-null int64
Country                    1000 non-null object
Timestamp                  1000 non-null object
Clicked on Ad              1000 non-null int64
dtypes: float64(3), int64(3), object(4)
memory usage: 78.2+ KB
```

Are there any duplicate records present?

```
dataframe.duplicated().sum()
```

```
0
```

As the value above is zero, there areno duplicates.

Attribute Type Classification

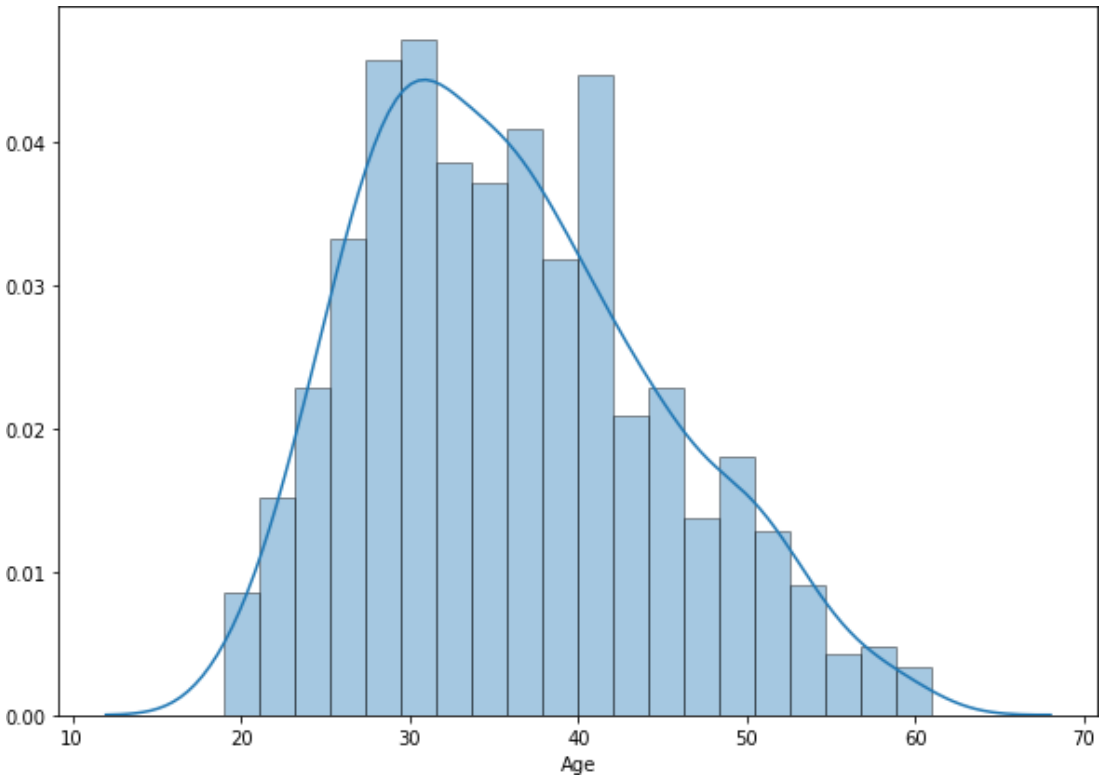
Determing the type of attributes in the given dataset

```
numeric_columns = ['Daily Time Spent on Site', 'Age', 'Area Income', 'Daily I
nternet Usage' ]

categorical_columns = [ 'Ad Topic Line', 'City', 'Male', 'Country', 'Clicked
on Ad' ]
```

Exploratory Data Analysis

What age group does the dataset majorly consist of?

```
plt.figure(figsize=(10,7))
sns.distplot(dataframe['Age'], bins = 20, kde=True, hist_kws=dict(edgecolor="
k", linewidth=1))

<matplotlib.axes._subplots.AxesSubplot at 0x1aff9441cf8>
```



Here, we can see that most of the internet users are having age in the range of 26 to 42 years.

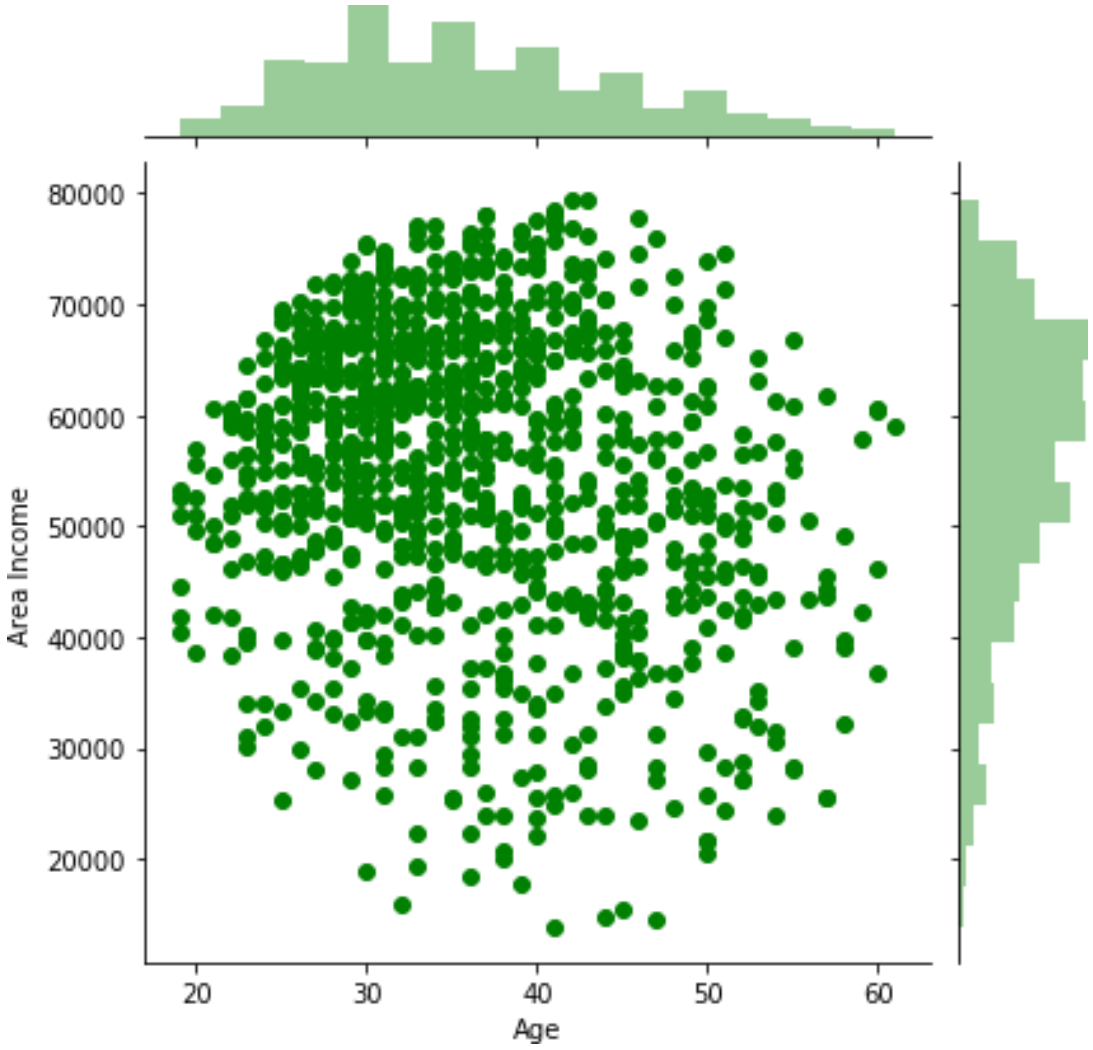
```
print('Age of the oldest person:', dataframe['Age'].max(), 'Years')
print('Age of the youngest person:', dataframe['Age'].min(), 'Years')
print('Average age in dataset:', dataframe['Age'].mean(), 'Years')
```

```
Age of the oldest person: 61 Years
Age of the youngest person: 19 Years
Average age in dataset: 36.009 Years
```

What is the income distribution in different age groups?

```
sns.jointplot(x='Age', y='Area Income', color= "green", data= dataframe)
```

```
<seaborn.axisgrid.JointGrid at 0x1aff92560f0>
```

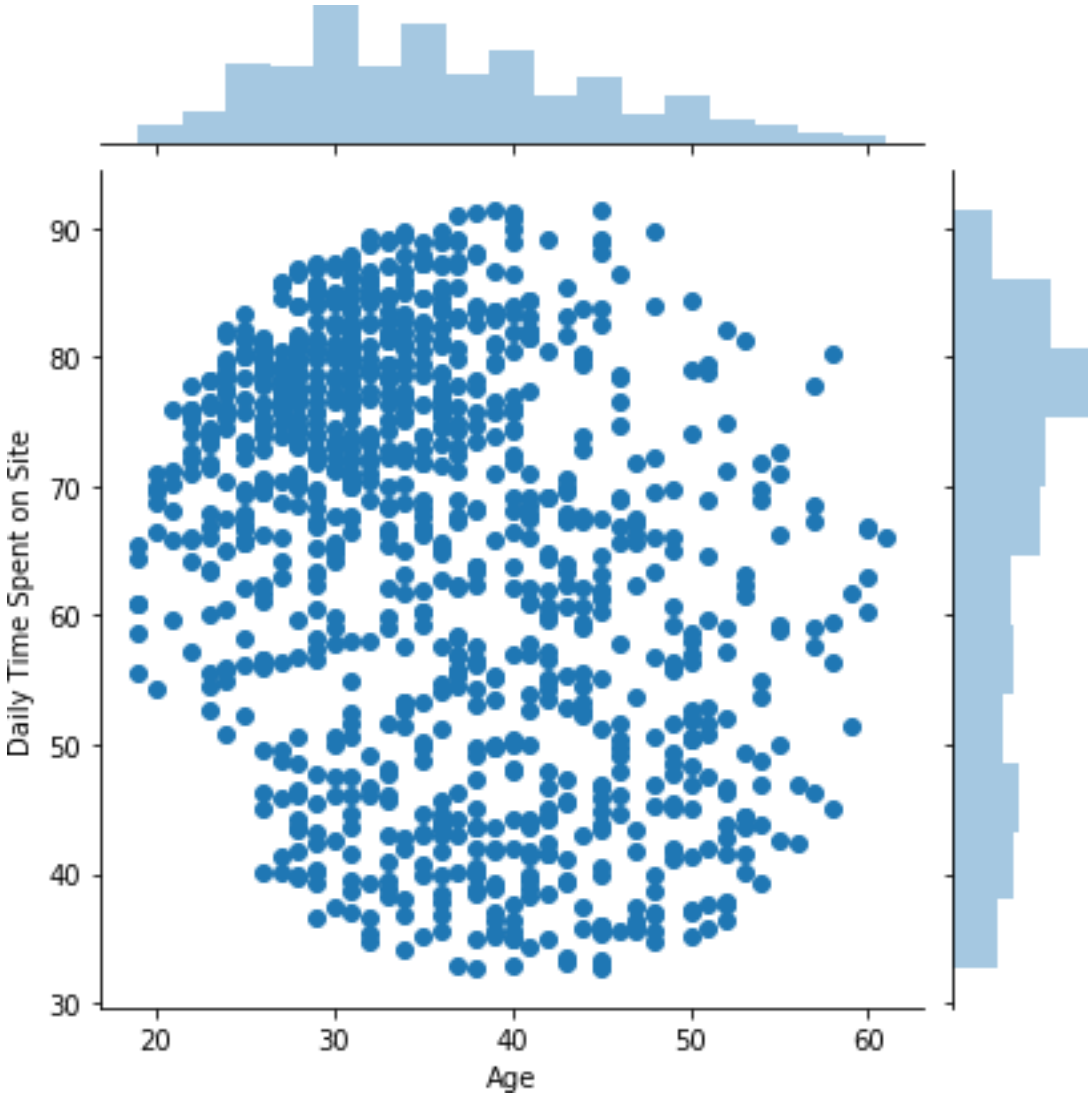


Here, we can see that mostly teenagers are higher earners with age group of 20-40 earning 50k-70k.

Which age group is spending maximum time on the internet?

```
sns.jointplot(x='Age', y='Daily Time Spent on Site', data= dataframe)
```

```
<seaborn.axisgrid.JointGrid at 0x1aff91992e8>
```



From the above plot its evident that the age group of 25-40 is most active on the internet.

Which gender has clicked more on online ads?

```
dataframe.groupby(['Male','Clicked on Ad'])['Clicked on Ad'].count().unstack(
)
```

```
Clicked on Ad    0    1
Male
0              250  269
1              250  231
```

Based on above data we can see that a greater number of females have clicked on ads compared to male.

Maximum number of internet users belong to which country in the given dataset?

```
pd.crosstab(index=dataframe['Country'],columns='count').sort_values(['count']
, ascending=False)
```

col_0	count
Country	
France	9
Czech Republic	9
Afghanistan	8
Australia	8
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
Ethiopia	7
Luxembourg	7
Eritrea	7
Western Sahara	7
Madagascar	6
Zimbabwe	6
Malta	6
Croatia	6
Mexico	6
Costa Rica	6
...	...
Slovakia (Slovak Republic)	2
Mauritania	2
Montenegro	2
Djibouti	2
Namibia	2
New Caledonia	2

```
Norway                                                 2
Panama                                                 2
Pitcairn Islands                                       2
Reunion                                                2
Saint Barthelemy                                       2
Benin                                                  2
Saint Lucia                                            2
Sao Tome and Principe                                  2
Andorra                                                2
Sierra Leone                                           2
British Indian Ocean Territory (Chagos Archipel...     1
Cape Verde                                             1
Bermuda                                                1
Slovenia                                               1
Lesotho                                                1
Germany                                                1
Jordan                                                 1
Kiribati                                               1
Marshall Islands                                       1
Montserrat                                             1
Mozambique                                             1
Romania                                                1
Saint Kitts and Nevis                                  1
Aruba                                                  1

[237 rows x 1 columns]
```

Based on the above data frame we can observe that maximum number of users are from France and Czech.

Did we match our baseline that we set?

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

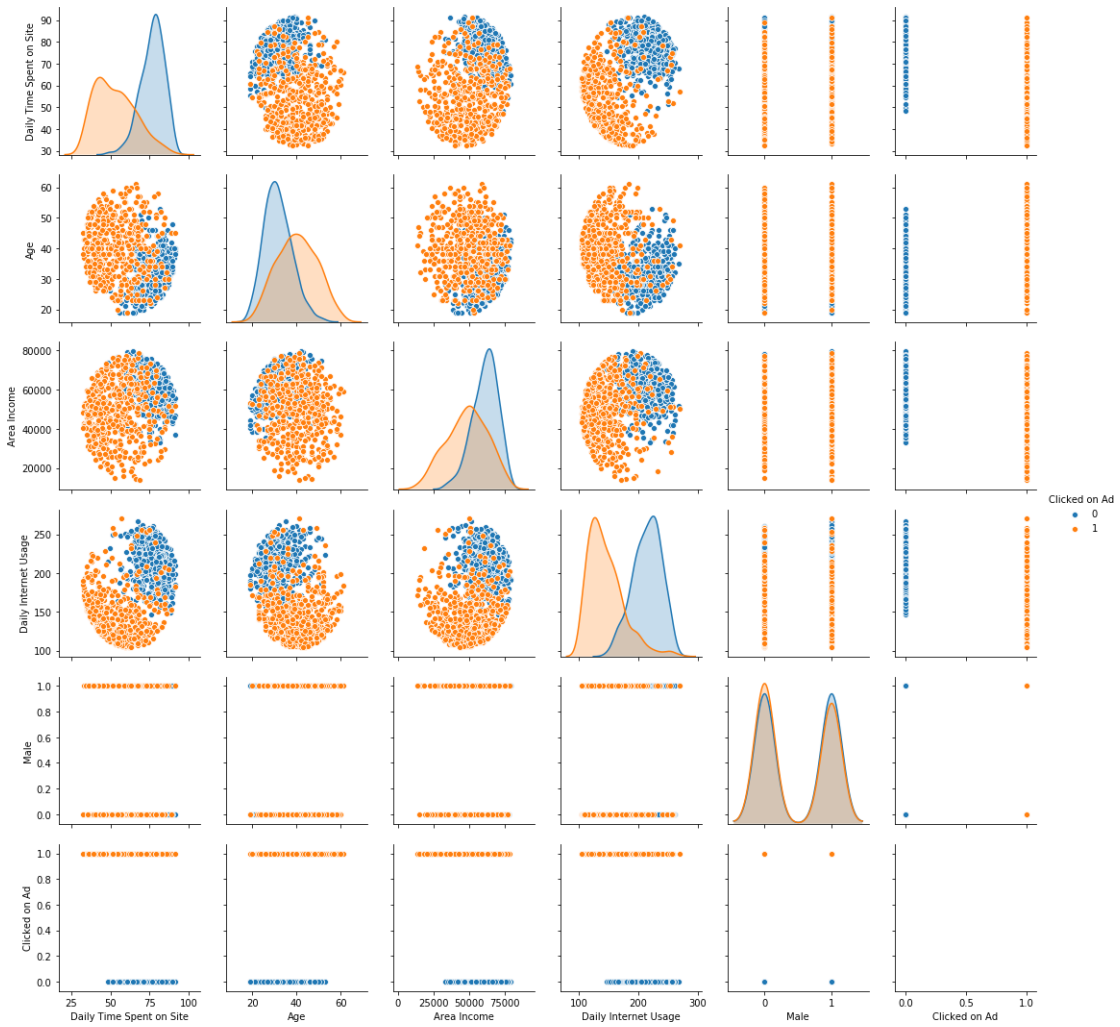
```
               Clicked on Ad  Daily Time Spent on Site     Age  Area Income
\
Clicked on Ad
0                        0.0                  76.85462  31.684  61385.58642
1                        1.0                  53.14578  40.334  48614.41374

               Daily Internet Usage
Clicked on Ad
0                         214.51374
1                         145.48646
```

What is the relationship between different features?

```
sns.pairplot(dataframe, hue='Clicked on Ad')
```

```
C:\Users\prani\Anaconda3\lib\site-packages\statsmodels\nonparametric\kde.py:4
87: RuntimeWarning: invalid value encountered in true_divide
  binned = fast_linbin(X, a, b, gridsize) / (delta * nobs)
C:\Users\prani\Anaconda3\lib\site-packages\statsmodels\nonparametric\kdetools
.py:34: RuntimeWarning: invalid value encountered in double_scalars
  FAC1 = 2*(np.pi*bw/RANGE)**2
```

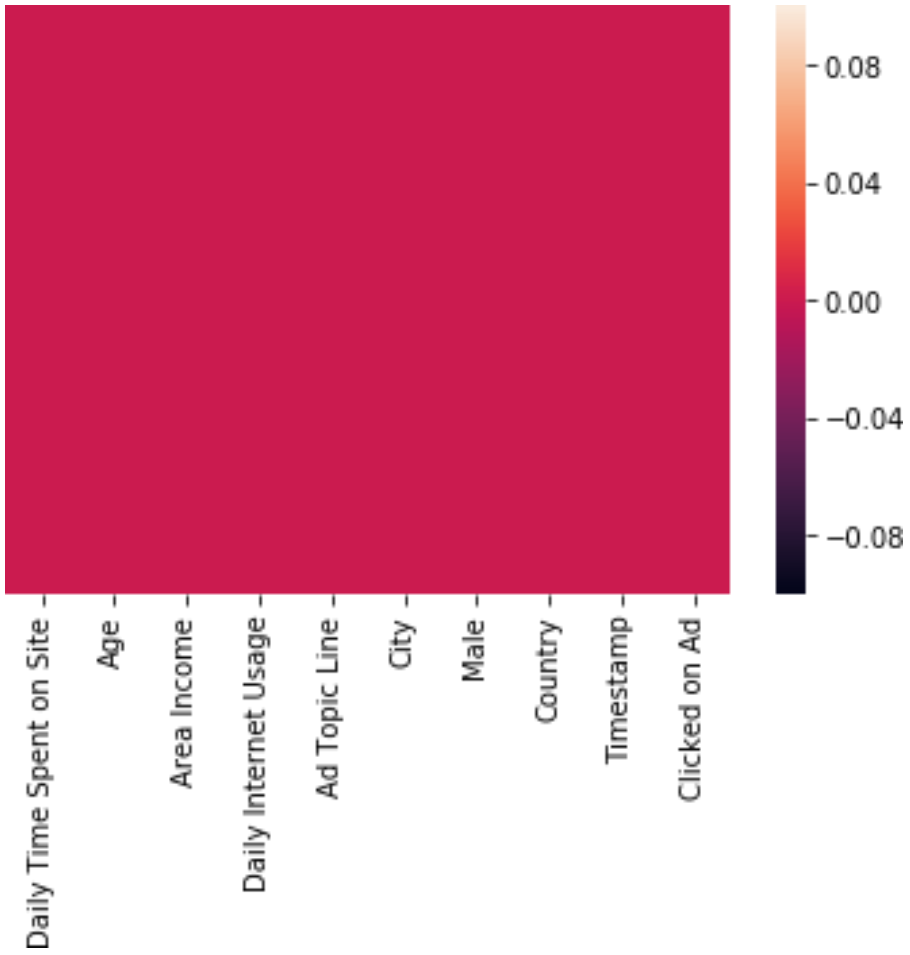
```
<seaborn.axisgrid.PairGrid at 0x1aff9341160>
```



Data Cleaning

```
sns.heatmap(dataframe.isnull(), yticklabels=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1affa60f710>
```



As we see, we don't have any missing data

Considering the 'Advertisement Topic Line', we decided to drop it. In any case, if we need to extract any form of interesting data from it, we can use Natural Language Processing.

As to 'City' and the 'Nation', we can supplant them by dummy variables with numerical features, Nonetheless, along these lines we got such a large number of new highlights.

Another methodology would be thinking about them as a categorical features and coding them in one numeric element.

Changing 'Timestamp' into numerical value is more complicated. So, we can change ‘Timestamp’ to numbers or convert them to spaces of time/day and consider it to be categorical and afterwards we converted it into numerical values. And we selected the month and the hour from the timestamp as features

```
dataframe['City Codes']= dataframe['City'].astype('category').cat.codes

dataframe['Country Codes'] = dataframe['Country'].astype('category').cat.code
s

dataframe[['City Codes','Country Codes']].head(5)
```

	City Codes	Country Codes
0	961	215
1	903	147
2	111	184
3	939	103
4	805	96

```
dataframe['Month'] = dataframe['Timestamp'].apply(lambda x: x.split('-')[1])
dataframe['Hour'] = dataframe['Timestamp'].apply(lambda x: x.split(':')[0].sp
lit(' ')[1])

dataframe[['Month','Hour']].head(5)
```

	Month	Hour
0	03	00
1	04	01
2	03	20
3	01	02
4	06	03

Data Model Implementation

Dropping

```
X = dataframe.drop(labels=['Ad Topic Line','City','Country','Timestamp','Clic
ked on Ad'], axis=1)

Y = dataframe['Clicked on Ad']
```

Splitting Dataset

```
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, rand
om_state = 42)
```

Implementing Logistic Regression Model

```
from sklearn.linear_model import LogisticRegression
```

```
log_reg_model = LogisticRegression()
```

```
log_reg_model.fit(X_train, Y_train)
```

```
C:\Users\prani\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:4
32: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify
a solver to silence this warning.
  FutureWarning)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
log_reg_pred = log_reg_model.predict(X_test)
```

Implementing Naive Bayes Model

```
from sklearn.naive_bayes import GaussianNB
```

```
nav_bayes_model = GaussianNB()
```

```
nav_bayes_model.fit(X_train, Y_train)
```

```
GaussianNB(priors=None, var_smoothing=1e-09)
```

```
nav_bayes_pred = nav_bayes_model.predict(X_test)
```

Implementing Decision Tree Model

```
from sklearn.tree import DecisionTreeClassifier
```

```
dec_tree_model = DecisionTreeClassifier()
```

```
dec_tree_model.fit(X_train, Y_train)
```

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

```
dec_tree_pred = dec_tree_model.predict(X_test)
```

Finding accuracy in each model

```
from sklearn.metrics import accuracy_score
```

Logistic Regression

```
log_reg_accuracy = accuracy_score(log_reg_pred, Y_test)
print(log_reg_accuracy*100)
```

```
90.0
```

Naive Bayes

```
nav_bayes_accuracy = accuracy_score(nav_bayes_pred, Y_test)
print(nav_bayes_accuracy*100)
```

```
96.0
```

Decision Tree

```
dec_tree_accuracy = accuracy_score(dec_tree_pred, Y_test)
print(dec_tree_accuracy*100);
```

```
93.33333333333333
```

Conclusion

Comparing all the above implementation models, it can be concluded that Naive Bayes Algorithm gives the maximum accuracy for determining the click probability. It can be believed that 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.