Ads Click-Through Rate Prediction

Ads Click Through Rate is the ratio of how many users clicked on your ad to how many users viewed your ad.

For example, 5 out of 100 users click on the ad while watching a youtube video. So, in this case, the CTR of the youtube ad will be 5%. Analyzing the click-through rate help companies in finding the best ad for their target audience.

In [1]:

```
#Lets us import the necessary Python libraries and the dataset:
import pandas as pd
import plotly.graph_objects as go
import plotly.express as px
import plotly.io as pio
import numpy as np
pio.templates.default = "plotly_white"

data = pd.read_csv("ad_10000records.csv")
print(data.head())
```

```
   Daily Time Spent on Site   Age  Area Income  Daily Internet Usage  \
0                     62.26  32.0     69481.85                172.83   
1                     41.73  31.0     61840.26                207.17   
2                     44.40  30.0     57877.15                172.83   
3                     59.88  28.0     56180.93                207.17   
4                     49.21  30.0     54324.73                201.58   

                         Ad Topic Line             City  Gender  \
0       Decentralized real-time circuit         Lisafort    Male   
1        Optional full-range projection  West Angelabury    Male   
2  Total 5thgeneration standardization       Reyesfurt  Female   
3           Balanced empowering success    New Michael  Female   
4  Total 5thgeneration standardization    West Richard  Female   

                        Country            Timestamp  Clicked on Ad  
0  Svalbard & Jan Mayen Islands  2016-06-09 21:43:05              0  
1                     Singapore  2016-01-16 17:56:05              0  
2                    Guadeloupe  2016-06-29 10:50:45              0  
3                        Zambia  2016-06-21 14:32:32              0  
4                         Qatar  2016-07-21 10:54:35              1  
```

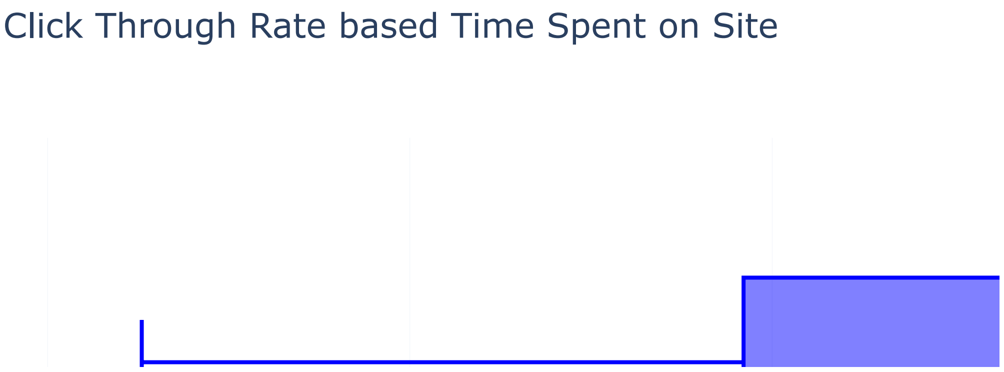
The “Clicked on Ad” column contains 0 and 1 values, where 0 means not clicked, and 1 means clicked.

In [2]:

```
#I’ll transform these values into “yes” and “no”:
data["Clicked on Ad"] = data["Clicked on Ad"].map({0: "No",
                                                   1: "Yes"})
```

Click Through Rate Analysis

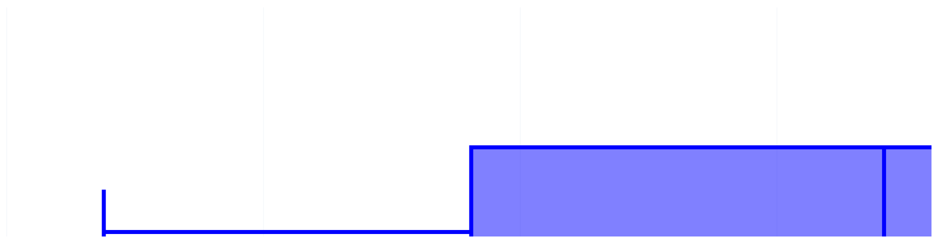
```
In [3]: #let’s analyze the click-through rate based on the time spent by the users on th
        fig = px.box(data,
                     x="Daily Time Spent on Site",
                     color="Clicked on Ad",
                     title="Click Through Rate based Time Spent on Site",
                     color_discrete_map={'Yes':'blue',
                                         'No':'red'})
        fig.update_traces(quartilemethod="exclusive")
        fig.show()
```



The above graph, we can see that the users who spend more time on the website click more on ads

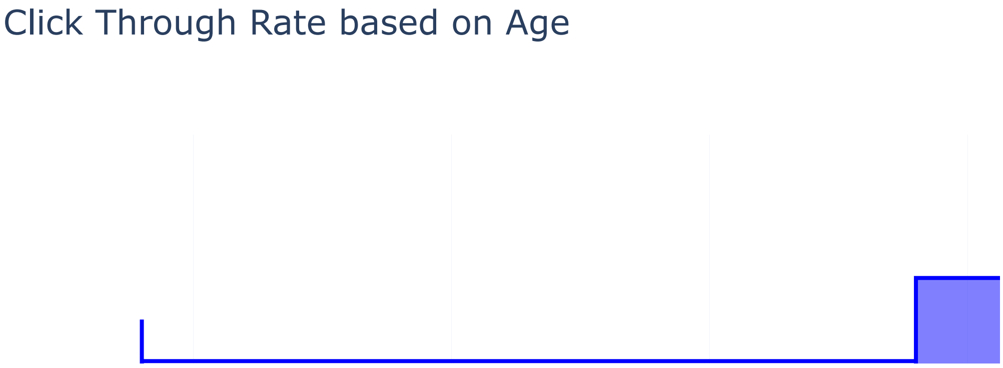
```
In [4]: #let’s analyze the click-through rate based on the daily internet usage of the u
        fig = px.box(data,
                     x="Daily Internet Usage",
                     color="Clicked on Ad",
                     title="Click Through Rate based on Daily Internet Usage",
                     color_discrete_map={'Yes':'blue',
                                         'No':'red'})
        fig.update_traces(quartilemethod="exclusive")
        fig.show()
```

Click Through Rate based on Daily Internet Usage



The users with high internet usage click less on ads compared to the users with low internet usage.

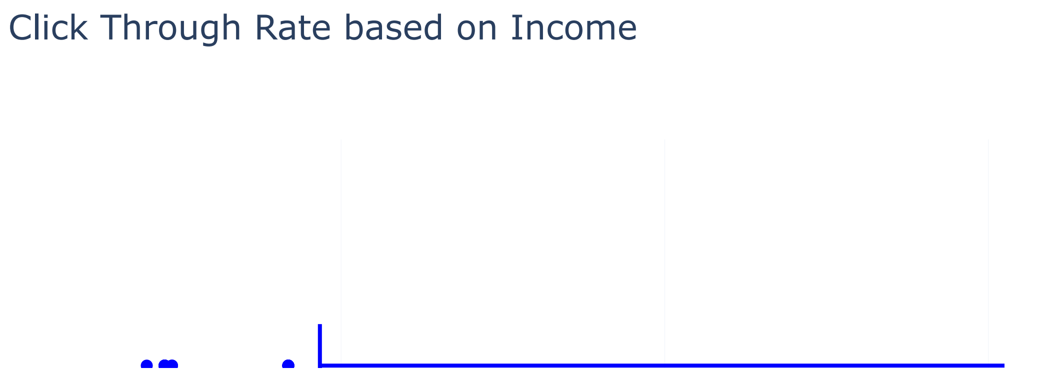
```
In [5]: #let’s analyze the click-through rate based on the age of the users:
fig = px.box(data,
             x="Age",
             color="Clicked on Ad",
             title="Click Through Rate based on Age",
             color_discrete_map={'Yes':'blue',
                                 'No':'red'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



The users around 40 years click more on ads compared to users around 27-36 years old.

In [6]:

```
#let’s analyze the click-through rate based on the income of the users:
fig = px.box(data,
             x="Area Income",
             color="Clicked on Ad",
             title="Click Through Rate based on Income",
             color_discrete_map={'Yes':'blue',
                                 'No':'red'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



There’s not much difference, but people from high-income areas click less on ads.

Calculating CTR of Ads

```
In [7]: #Let’s calculate the overall Ads click-through rate. Here we need to calculate t
        data["Clicked on Ad"].value_counts()

Out[7]: Clicked on Ad
        No     5083
        Yes    4917
        Name: count, dtype: int64

In [8]: #Let’s calculate the CTR:
        click_through_rate = 4917 / 10000 * 100
        print(click_through_rate)

       49.17
```

Click Through Rate Prediction Model

```
In [9]: #Let’s move on to training a Machine Learning model to predict click-through rat
        data["Gender"] = data["Gender"].map({"Male": 1,
                                             "Female": 0})
```

```
x=data.iloc[:,0:7]
x=x.drop(['Ad Topic Line','City'],axis=1)
y=data.iloc[:,9]

from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,
                                           test_size=0.2,
                                           random_state=4)
```

In [10]:

```
#let’s train the model using the random forecast classification algorithm:
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(x, y)
```

Out[10]:

```
▾ RandomForestClassifier

RandomForestClassifier()
```

In [11]:

```
#let’s have a look at the accuracy of the model:
from sklearn.metrics import accuracy_score
y_pred = model.predict(xtest)
print(accuracy_score(ytest,y_pred))
```

```
0.961
```

In [12]:

```
#let’s test the model by making predictions:
print("Ads Click Through Rate Prediction : ")
a = float(input("Daily Time Spent on Site: "))
b = float(input("Age: "))
c = float(input("Area Income: "))
d = float(input("Daily Internet Usage: "))
e = input("Gender (Male = 1, Female = 0) : ")

features = np.array([[a, b, c, d, e]])
print("Will the user click on ad = ", model.predict(features))
```

```
Ads Click Through Rate Prediction :
Daily Time Spent on Site: 120
Age: 26
Area Income: 50000
Daily Internet Usage: 200
Gender (Male = 1, Female = 0) : 1
Will the user click on ad =  ['No']
```

```
C:\Users\Sethu\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\base.py:464: UserWarning:

X does not have valid feature names, but RandomForestClassifier was fitted with f
eature names
```