Google Ads Hourly Analysis

Date : 21th June 2023

Project Start Date - End Date	• Start Date –21-06 -2023 • End Date – 22–06-2023
Objectives	• To analyses how many people who clicked on the advertisement highly interested to enroll in our course • General exploratory analyses • General descriptive analyses
Milestones accomplished the week of Start Date - End Date:	• Descriptive analyses • Exploratory analyses • Classification of date with respect to term

Contact Information

This project is performed for educational purpose of under the guidance of Siddhivinayak Sir .

Project Manager
Name: Siddhivinayak Phulwadkar
Mobile: 9028965955
Email: siddhivinayakphulwadkar@gmail.com

Project Trainee
Name: Arti Sukhadev Patil
Mobile: 8623845944
Email: parti5863@gmail.com

Project Abstract

The dataset is about showing advertisement to students or clients for course enrollment. Our main objective was to understand on which time students are clicked on our ads and getting enroll our course or to interest to buy course. Problem statement is classify as we are looking for preferred timing in a day and night where we can do marketing and we will get definitely sales or hot lead. For this dataset we have applied decision tree algorithm and performed exploratory and descriptive analysis.

Google Ads Hourly Analysis

#Importing libraries

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

#Importing dataset

```
file=pd.read_excel("C:/Users/Administrator/Desktop/Google ads hourly analysis 21th june.xlsx")
```

```
file.head()
```

	Sr no	Impressions	Clicks	Cost	CTR	CPC	Cold Leads	Warm Leads	Hot Leads
0	00:00:00	7840	706	689	0.09	0.975921	42	4	2
1	00:30:00	5694	171	508	0.03	2.970760	10	1	1
2	01:00:00	3610	72	495	0.02	6.875000	4	0	0
3	01:30:00	485	5	64	0.01	12.800000	0	0	0
4	02:00:00	125	0	59	0.00	0.000000	0	0	0

#Preprocessing the dataset

```
dataset=file.drop("Sr no",axis=1)
```

```
dataset1=dataset.drop(["CPC","CTR","Warm Leads"],axis=1)
```

```
dataset1.head()
```

	Impressions	Clicks	Cost	Cold Leads	Hot Leads
0	7840	706	689	42	2
1	5694	171	508	10	1
2	3610	72	495	4	0
3	485	5	64	0	0
4	125	0	59	0	0

```
a=dataset.isnull().sum()
```

```
a
```

```
Impressions     0
Clicks          0
Cost            0
CTR             0
CPC             0
Cold Leads      0
Warm Leads      0
Hot Leads       0
dtype: int64
```

#Descriptive analysis

```
# descriptive analysis
```

```
dataset.sum()
```

```
Impressions     166257.000000
Clicks           23630.000000
Cost             18379.000000
CTR                  4.332000
CPC                144.808744
Cold Leads        1418.000000
Warm Leads         262.000000
Hot Leads          122.000000
dtype: float64
```

```
#Impressions are no of visible ads to customers
#Total impressions are 166257.000000
#Clicks indicates total no of customers who clicked on our ads
#total no of clicks ad 23630.000000
```

```
#Cost indicates cost per click and impression for ads
#Total cost are 18379.000000
```

```
#CPC indicates dividing the ads cost by the number of clicks generated by an ads
#total CPC is 144.808744
```

```
#Cold lead indicates those  customers shows only intrest in Ads
#Warm lead indicates those customers are showing intrest in Ads and providing their contact details
#Hot lead indicates those customers are highly intrested and are ready to buy
```

```
#Total no of Cold lead are 1418.000000
#Total no of Warm lead are 262.000000
#Total no of Hot lead are 122.000000
```

```
dataset.mean()
```

```
Impressions     3463.687500
Clicks           492.291667
Cost             382.895833
CTR                0.090250
CPC                3.016849
Cold Leads        29.541667
Warm Leads         5.458333
Hot Leads          2.541667
dtype: float64
```

```
#the average no of Impression are 3463.687500
#the average no of Clicks are 492.291667
#the average no of cost is 382.895833
#the average no of CTR is 0.090250
#the average no of CPC is 3.016849
#the average no of cold lead  29.541667
#the average no of warm lead 5.458333
#the average no of hot lead 2.541667
```

```
#Individual ratio
#Individual ratio of cold lead and warm lead is 9.52%
#Individual ratio of warm lead and hot lead is 50%
#Individual ratio of cold lead and hot lead is 4.76%
```

```
#On an average ratio
#on an average ratio of cold lead and warm lead is 18.47%
#on an average ratio of warm lead and hot lead is 46.56%
#on an average ratio of cold lead and hot lead is 8.60%
```

```
x=dataset1.iloc[:,:-1].values
y=dataset1.iloc[:,-1].values
```

```
x
```

```
array([[7840,  706,  689,   42],
       [5694,  171,  508,   10],
       [3610,   72,  495,    4],
       [ 485,    5,   64,    0],
       [ 125,    0,   59,    0],
       [  48,    0,   12,    0],
       [  12,    0,    7,    0],
       [   6,    0,    6,    0],
       [   9,    0,    9,    0],
       [   3,    0,   12,    0],
       [   6,    0,   16,    0],
       [   8,    0,    4,    0],
       [   9,    0,    6,    0],
       [  12,    0,   23,    0],
       [  46,    1,   27,    0],
       [ 152,    3,   50,    0],
       [ 164,    5,   80,    0],
       [ 174,    7,  150,    0],
       [2360,   24,  265,    1],
       [2159,  108,  314,    6],
       [1264,   88,  326,    5],
       [1963,  177,  465,   11],
       [1786,  143,  329,    9],
       [2109,  169,  384,   10],
       [2964,  593,  396,   36],
       [3015,  211,  465,   13],
       [3415,  376,  465,   23],
       [3625,  435,  402,   26],
```

```
y
```

```
array([2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
       1, 1, 2, 1, 1, 2, 1, 1, 4, 6, 2, 9, 3, 7, 6, 5, 7, 1, 7, 3, 9, 9,
       6, 8, 8, 6], dtype=int64)
```

#Training the model

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.10, random_state =0)
```

```
print(x_train)
```

```
[[7985 1198  623   72]
 [4526  815  569   49]
 [9465 2177  715  131]
 [   6    0    6    0]
 [   6    0   16    0]
 [   8    0    4    0]
 [4762  429  263   26]
 [3152  441  498   26]
 [3625  435  402   26]
 [8811  705  798   42]
 [3610   72  495    4]
 [6123  857  658   51]
 [2360   24  265    1]
 [ 152    3   50    0]
 [3648  328  426   20]
 [1786  143  329    9]
 [ 164    5   80    0]
 [7521 1805  795  108]
 [1264   88  326    5]
 [5969 1134  742   68]
 [   9    0    9    0]
 [  12    0   23    0]
 [3015  211  465   13]
 [  48    0   12    0]
 [ 174    7  150    0]
 [6495  844  548   51]
 [  46    1   27    0]
 [7495 1199  528   72]
 [5694  171  508   10]
```

```
print(y_train)
```

```
[5 3 7 0 0 0 6 9 2 6 0 8 0 0 1 1 0 3 1 9 0 0 1 0 0 7 0 7 1 0 9 2 0 1 6 1 1
 0 1 8 0 2 6]
```

```
print(x_test)
```

```
[[4059  325  486   20]
 [ 125    0   59    0]
 [3415  376  465   23]
 [4216  506  501   30]
 [4011  642  469   39]]
```

```
print(y_test)
```

```
[1 0 1 4 2]
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

```
print(x_train)
```

```
[[ 1.44694243  1.14189931  0.85763045  1.14307455]
 [ 0.33140535  0.50940546  0.66596367  0.5115099 ]
 [ 1.92424653  2.75863948  1.18417387  2.76317518]
 [-1.12630717 -0.83650183 -1.33234005 -0.83399741]
 [-1.12630717 -0.83650183 -1.2968462  -0.83399741]
 [-1.12566217 -0.83650183 -1.33943882 -0.83399741]
 [ 0.407516   -0.12804266 -0.42014812 -0.12005476]
 [-0.11171346 -0.10822562  0.41395734 -0.12005476]
 [ 0.04083035 -0.11813414  0.07321638 -0.12005476]
 [ 1.71332972  0.32774926  1.47877282  0.31929457]
 [ 0.03599281 -0.71759959  0.40330918 -0.72416008]
 [ 0.84644227  0.5787651   0.98185893  0.56642856]
 [-0.36713565 -0.79686775 -0.41304935 -0.80653808]
 [-1.07922177 -0.83154757 -1.17616712 -0.83399741]
 [ 0.04824791 -0.29483607  0.15840162 -0.28481075]
 [-0.55225224 -0.60034877 -0.18588872 -0.58686341]
 [-1.07535174 -0.82824473 -1.06968557 -0.83399741]
 [ 1.29730114  2.14431124  1.46812466  2.13161053]
 [-0.72059869 -0.69117687 -0.19653687 -0.69670075]
 [ 0.79677684  1.03620843  1.28000726  1.03323722]
 [-1.12533967 -0.83650183 -1.3216919  -0.83399741]
 [-1.12437216 -0.83650183 -1.27200051 -0.83399741]
 [-0.15589634 -0.48805221  0.29682763 -0.47702608]
 [-1.11276206 -0.83650183 -1.31104374 -0.83399741]
 [-1.07212671 -0.82494189 -0.82122862 -0.83399741]
 [ 0.9664133   0.55729664  0.59142658  0.56642856]
 [-1.11340706 -0.83485041 -1.25780297 -0.83399741]
 [ 1.28891607  1.14355073  0.52043888  1.14307455]
 [ 0.70808858 -0.55410901  0.44945119 -0.55940408]
```

```
print(x_test)
```

```
[[ 1.80796553e-01 -2.99790335e-01  3.71364717e-01 -2.84810754e-01]
 [-1.08792934e+00 -8.36501830e-01 -1.14422265e+00 -8.33997409e-01]
 [-2.68952310e-02 -2.15567916e-01  2.96827633e-01 -2.02432756e-01]
 [ 2.31429488e-01 -8.83317666e-04  4.24605491e-01 -1.02174261e-02]
 [ 1.65316420e-01  2.23709800e-01  3.11025173e-01  2.36916569e-01]]
```

#Training the Decision Tree Classification model

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'gini', random_state = 0)
classifier.fit(x_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```

```
print(classifier.predict(sc.transform([[9489,1423,628,57]])))
```

```
[5]
```

#Predicting the Test set results

```
y_pred = classifier.predict(x_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

```
[[1 1]
 [0 0]
 [1 1]
 [6 4]
 [9 2]]
```

#Making the Confusion Matrix

```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

```
[[1 0 0 0 0 0]
 [0 2 0 0 0 0]
 [0 0 0 0 0 1]
 [0 0 0 0 1 0]
 [0 0 0 0 0 0]
 [0 0 0 0 0 0]]

0.6
```

```
#Accuracy is 60%
```

#Conclusion

```
#Conclusion
#There are 31 no of preferred time slot of 30 min in a entire day where it is preferable to show the Advertisement
#In which 3:00 pm to 11:30 pm from Afternoon to night company has generated more no of hot leads so we concluded that
#We can show ads to people in above given time slots as more no of people use their mobile phones in evening and night
#so this time is preferrable.
#The maximum cost we used to show ad is in time slot of 3:00 pm to 11:30 pm where we generate 106 no of hot leads
#In each no of time slot
# In the time slots company has get more no of impressions, clicks,cold lead,warm lead,hot lead.
```

```
#There are 17 no of non preferred time slots and that company dosent get hot lead
```

```
#In time slots where company dosent get hot lead company spent 1285 rs
#In reamaning time slots where company got hot leads company spent 17094 rs
#So total cost spent by  company on every time slot is 18379 rs
#93.00% of cost spent by company make profit by getting HOT Leads.
```

```
#We suggest we can reduce the cost by 6.99% not spending in the 1:00 am to 9:00 am time slots
```

```
#On an average company spent 140.11rs for each hot lead.
```

Comparative Analysis of 19th June, 20th June,21st June 2023

I have started this live project working from 19th June to 21st June.

The data is about time slots showing ads and generate leads. Each time slot is divide into 30 min these time slots seems to be effective for showing ads and it shows a good time to focus on displaying ads to reach maximum leads and engage the leads. We saw that time slot from 5:00 pm to 12:00 am have generated hot leads in this slots. The total no of hot lead generated on 19th June are**73**. And ratio of converting cold lead **to hot lead is 18.62%.**and on 19th June we make profit of **96%** from cost spent on hot leads. Company spent total amount of **20057**for ads. **Company spent 264rs for each hot lead. CTR for 19th June is 5.21%.**

Data we observed on 20th June is bit similar to 20th June as at on same time slots of 5:00 pm to 12:00 am have generated higher no of hot leads in this slots. The total no of hot lead generated on 20th June are **135**. And ratio of converting cold lead to hot lead is **10.67%.**and on 20th June we make profit of **95.94%** from cost spent on hot leads. Company spent total amount of **16405** for ads. **Company spent 116rs for each hot lead. CTR for 20th June is 5.21%.**

Data we saw on 21st June by comparing 19th and 20th June we see that time slots is been changed from 3:00 pm to 12:00 am. The total no of hot lead generated on 21st June are **122**. And ratio of converting cold lead to hot lead is **8.60%.**and on 21st June we make profit of **93%** from cost spent on hot leads. Company spent total amount of **18379** for ads. **Company spent 140rs for each hot lead. CTR for 21st June is 4.33%.**

Daily Target limit for Advertisement cost is **20000rs**. We have not changed anything but we observed that cost spent on **20th June** and **21st June** is less than 19th June.