SWACHANDA

Google Ads hourly Analysis

Date : **20th June 23**

Project Start Date - End Date	● Start Date – 20-06 -2023 ● End Date – 21-06- 2023
Objectives	● To analyses how many people who clicked on the advertisement and highly interested to enroll in our course. ● General exploratory analyses. ● General descriptive analyse
Milestones accomplished the week of Start Date - End Date:	● Descriptive analyses ● Exploratory analyses ● Classification of data 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

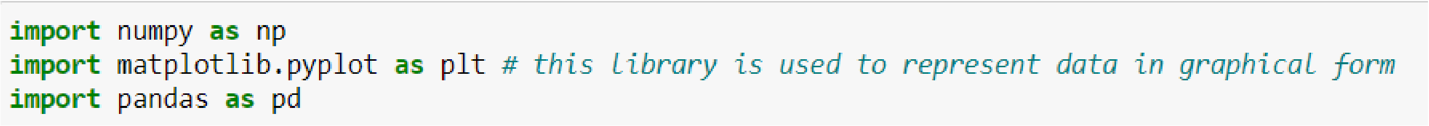
Student Name
Name : Shubham Kalyan Mane
Mobile: 972390175
Email: shubhammane656@gmail.com

Project Abstract

This dataset is all about showing Advertisement to clients and to enroll for the course . The main object was to understand at what time students are clicking on our ads and getting enrolled our course or to interested to buy courses. As we are looking for at which preferred time in a day. where we can do marketing and we will get sales definitely. For this dataset we have applied Decision Tree algorithm and performed exploratory and descriptive analysis.

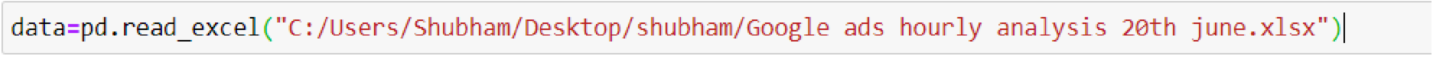
Google Ads Hourly Analysis

Importing libraries

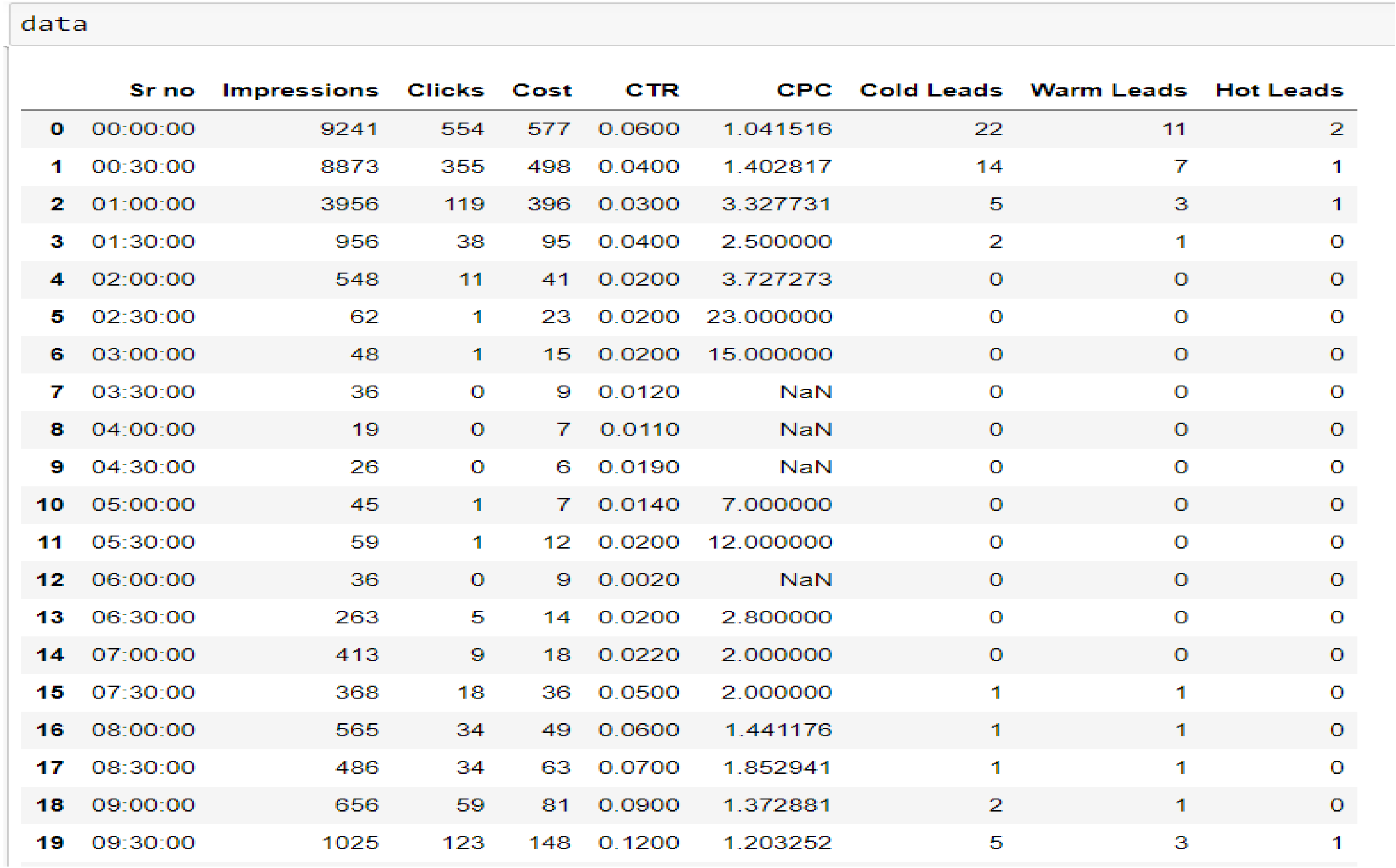


```
import numpy as np
import matplotlib.pyplot as plt # this library is used to represent data in graphical form
import pandas as pd
```

Importing dataset



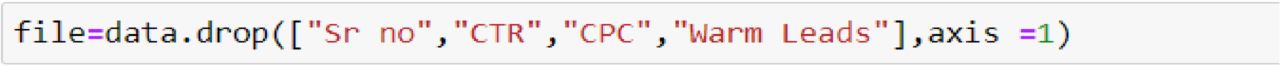
```
data=pd.read_excel("C:/Users/Shubham/Desktop/shubham/Google ads hourly analysis 20th june.xlsx")
```



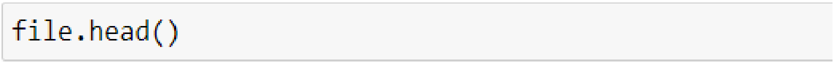
```
data
```

	Sr no	Impressions	Clicks	Cost	CTR	CPC	Cold Leads	Warm Leads	Hot Leads
0	00:00:00	9241	554	577	0.0600	1.041516	22	11	2
1	00:30:00	8873	355	498	0.0400	1.402817	14	7	1
2	01:00:00	3956	119	396	0.0300	3.327731	5	3	1
3	01:30:00	956	38	95	0.0400	2.500000	2	1	0
4	02:00:00	548	11	41	0.0200	3.727273	0	0	0
5	02:30:00	62	1	23	0.0200	23.000000	0	0	0
6	03:00:00	48	1	15	0.0200	15.000000	0	0	0
7	03:30:00	36	0	9	0.0120	NaN	0	0	0
8	04:00:00	19	0	7	0.0110	NaN	0	0	0
9	04:30:00	26	0	6	0.0190	NaN	0	0	0
10	05:00:00	45	1	7	0.0140	7.000000	0	0	0
11	05:30:00	59	1	12	0.0200	12.000000	0	0	0
12	06:00:00	36	0	9	0.0020	NaN	0	0	0
13	06:30:00	263	5	14	0.0200	2.800000	0	0	0
14	07:00:00	413	9	18	0.0220	2.000000	0	0	0
15	07:30:00	368	18	36	0.0500	2.000000	1	1	0
16	08:00:00	565	34	49	0.0600	1.441176	1	1	0
17	08:30:00	486	34	63	0.0700	1.852941	1	1	0
18	09:00:00	656	59	81	0.0900	1.372881	2	1	0
19	09:30:00	1025	123	148	0.1200	1.203252	5	3	1

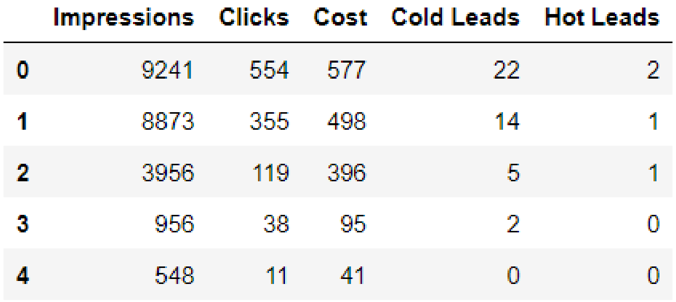
# Preprocessing the dataset



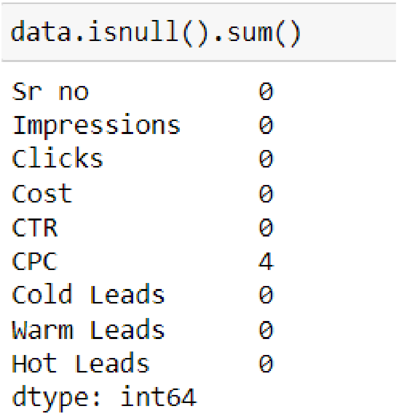
```
file=data.drop(["Sr no","CTR","CPC","Warm Leads"],axis =1)
```



```
file.head()
```

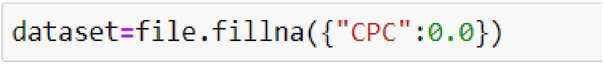


	Impressions	Clicks	Cost	Cold Leads	Hot Leads
0	9241	554	577	22	2
1	8873	355	498	14	1
2	3956	119	396	5	1
3	956	38	95	2	0
4	548	11	41	0	0

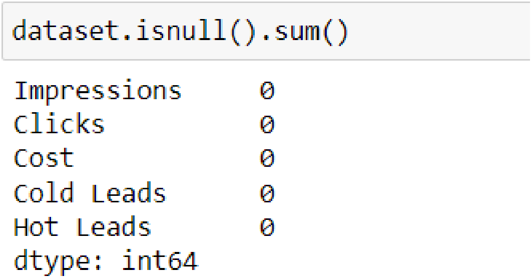


```
data.isnull().sum()
```

```
Sr no          0
Impressions    0
Clicks         0
Cost           0
CTR            0
CPC            4
Cold Leads     0
Warm Leads     0
Hot Leads      0
dtype: int64
```

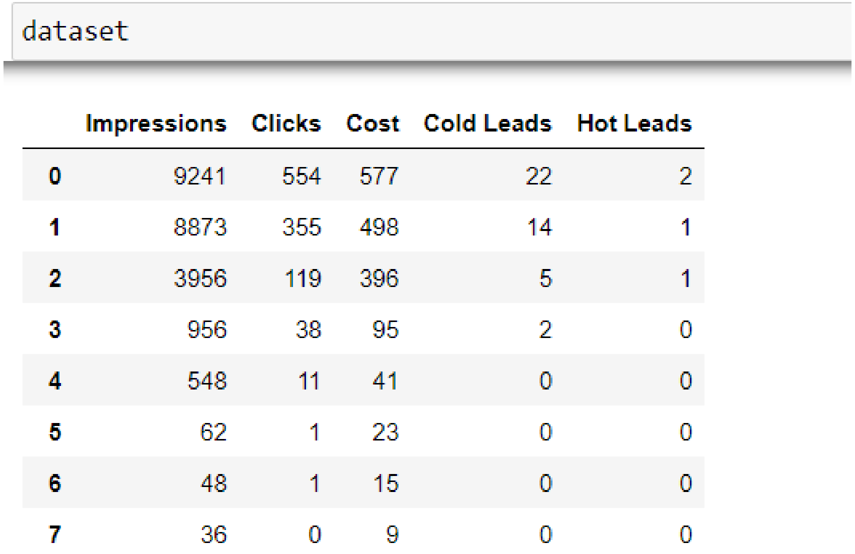


```
dataset=file.fillna({"CPC":0.0})
```



```
dataset.isnull().sum()
```

```
Impressions     0
Clicks          0
Cost            0
Cold Leads      0
Hot Leads       0
dtype: int64
```

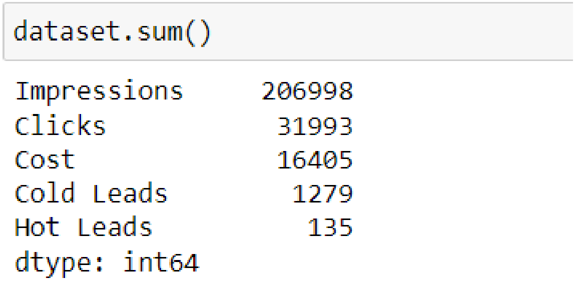


```
dataset
```

	Impressions	Clicks	Cost	Cold Leads	Hot Leads
0	9241	554	577	22	2
1	8873	355	498	14	1
2	3956	119	396	5	1
3	956	38	95	2	0
4	548	11	41	0	0
5	62	1	23	0	0
6	48	1	15	0	0
7	36	0	9	0	0

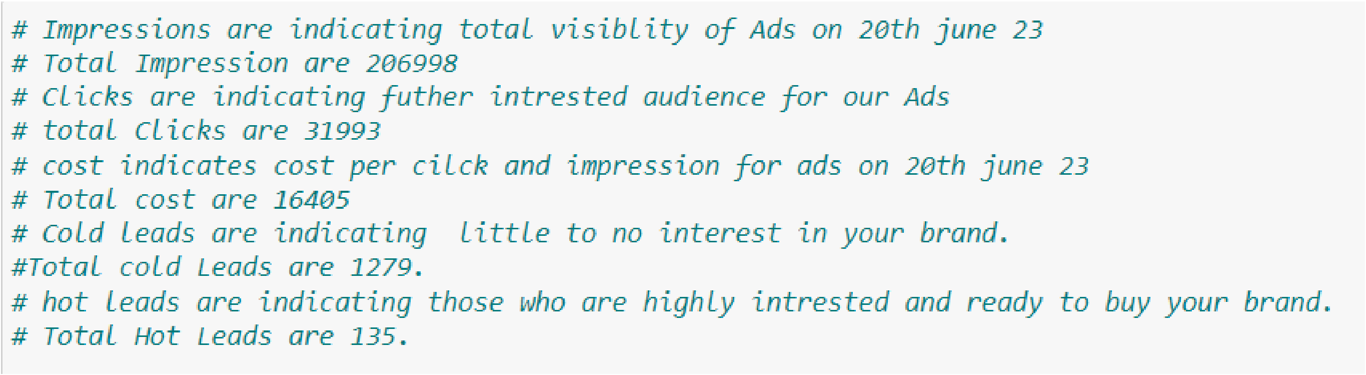
#Descriptive Analysis

```
# Discriptive Analysis
```

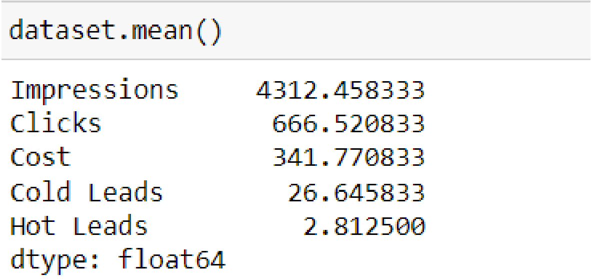


```
dataset.sum()
```

```
Impressions     206998
Clicks           31993
Cost             16405
Cold Leads        1279
Hot Leads          135
dtype: int64
```

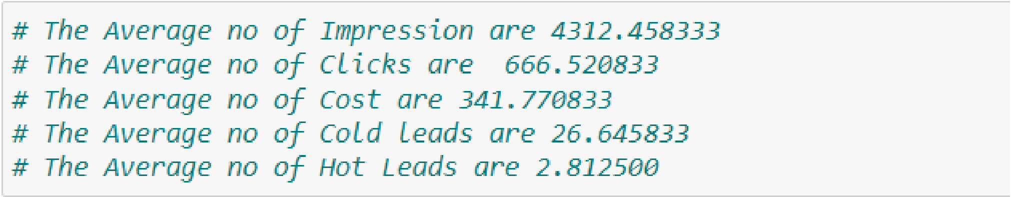


```
# Impressions are indicating total visiblity of Ads on 20th june 23
# Total Impression are 206998
# Clicks are indicating futher intrested audience for our Ads
# total Clicks are 31993
# cost indicates cost per cilck and impression for ads on 20th june 23
# Total cost are 16405
# Cold leads are indicating  little to no interest in your brand.
#Total cold Leads are 1279.
# hot leads are indicating those who are highly intrested and ready to buy your brand.
# Total Hot Leads are 135.
```

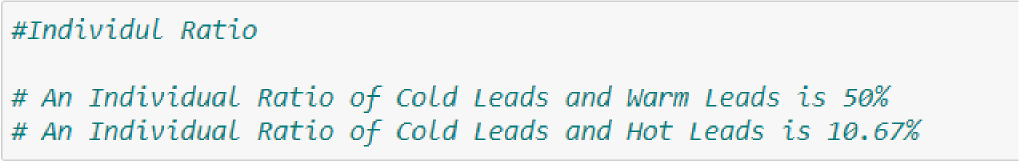


```
dataset.mean()
```

```
Impressions     4312.458333
Clicks           666.520833
Cost             341.770833
Cold Leads        26.645833
Hot Leads          2.812500
dtype: float64
```

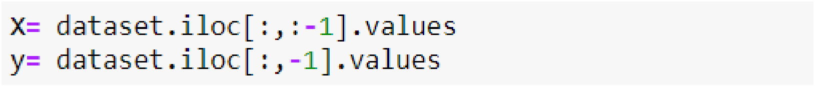


```
# The Average no of Impression are 4312.458333
# The Average no of Clicks are  666.520833
# The Average no of Cost are 341.770833
# The Average no of Cold leads are 26.645833
# The Average no of Hot Leads are 2.812500
```

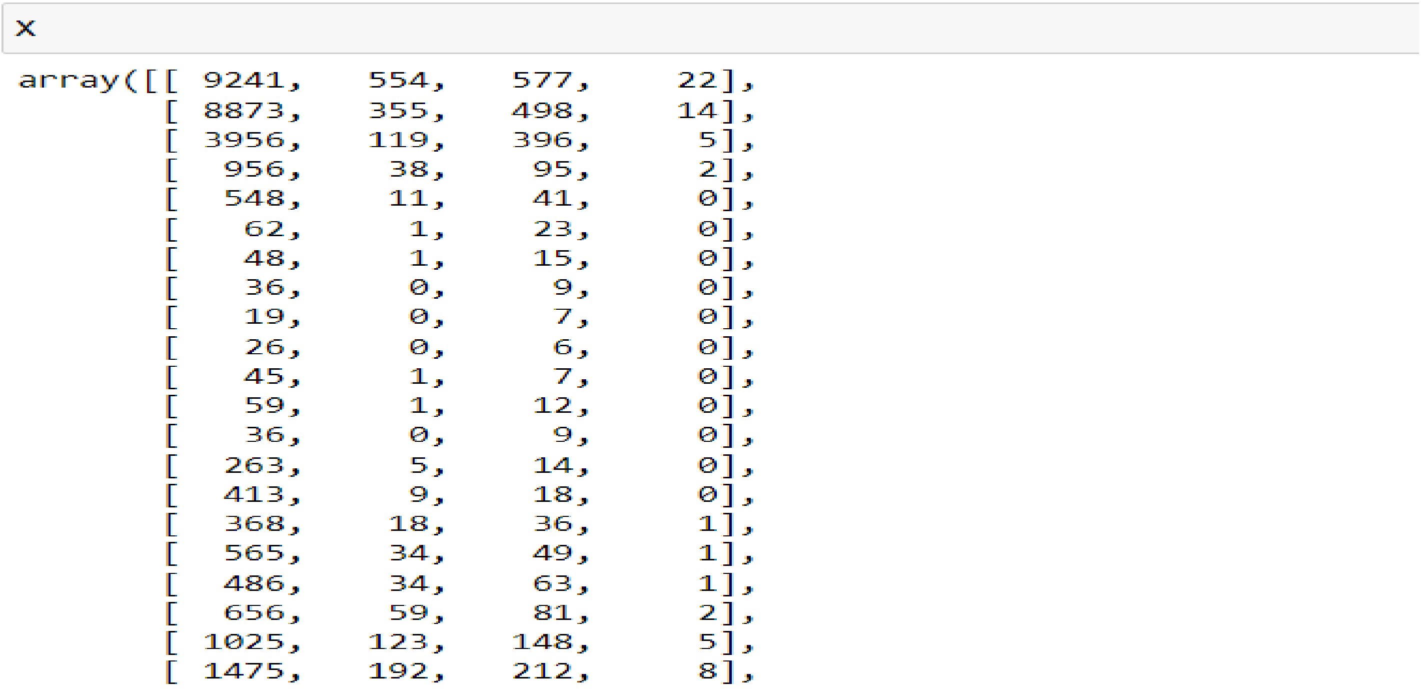


```
#Individul Ratio

# An Individual Ratio of Cold Leads and Warm Leads is 50%
# An Individual Ratio of Cold Leads and Hot Leads is 10.67%
```

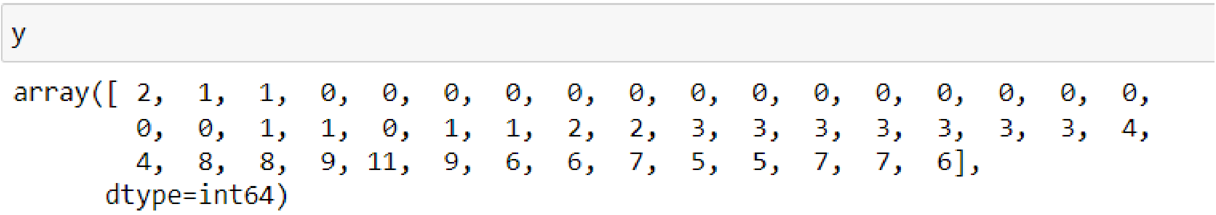


```
X= dataset.iloc[:,:-1].values
y= dataset.iloc[:,-1].values
```



```
X
```

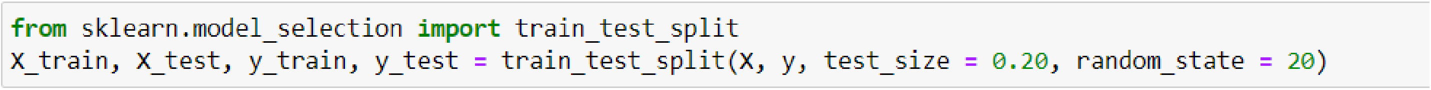
```
array([[ 9241,    554,    577,     22],
       [ 8873,    355,    498,     14],
       [ 3956,    119,    396,      5],
       [  956,     38,     95,      2],
       [  548,     11,     41,      0],
       [   62,      1,     23,      0],
       [   48,      1,     15,      0],
       [   36,      0,      9,      0],
       [   19,      0,      7,      0],
       [   26,      0,      6,      0],
       [   45,      1,      7,      0],
       [   59,      1,     12,      0],
       [   36,      0,      9,      0],
       [  263,      5,     14,      0],
       [  413,      9,     18,      0],
       [  368,     18,     36,      1],
       [  565,     34,     49,      1],
       [  486,     34,     63,      1],
       [  656,     59,     81,      2],
       [ 1025,    123,    148,      5],
       [ 1475,    192,    212,      8],
```



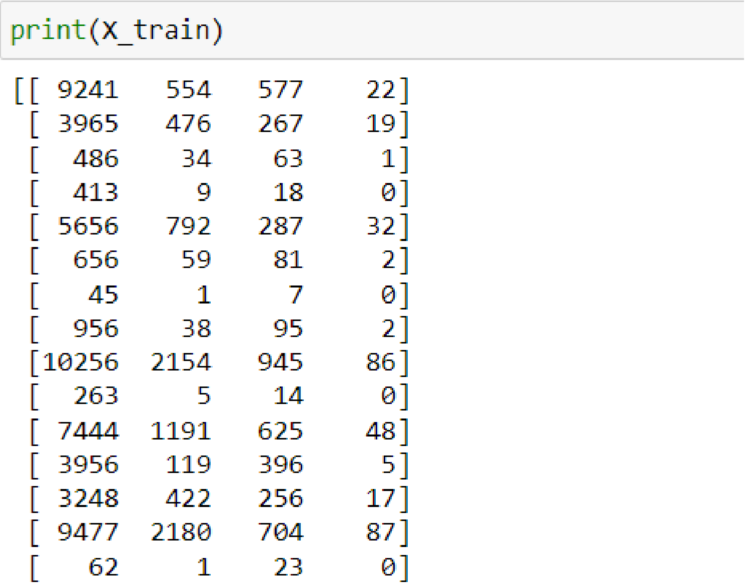
```
y
```

```
array([ 2,  1,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  1,  1,  0,  1,  1,  2,  2,  3,  3,  3,  3,  3,  3,  3,  4,
        4,  8,  8,  9, 11,  9,  6,  6,  7,  5,  5,  7,  7,  6],
      dtype=int64)
```

#Traning Model

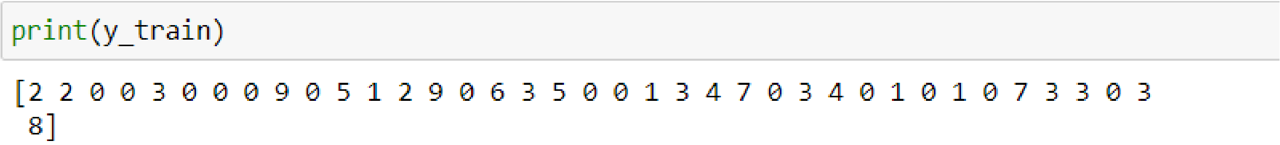


```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 20)
```



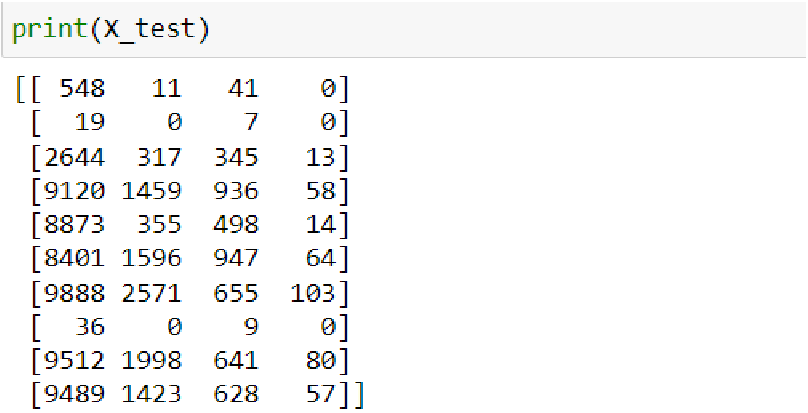
```
print(X_train)
```

```
[[ 9241   554   577    22]
 [ 3965   476   267    19]
 [  486    34    63     1]
 [  413     9    18     0]
 [ 5656   792   287    32]
 [  656    59    81     2]
 [   45     1     7     0]
 [  956    38    95     2]
 [10256  2154   945    86]
 [  263     5    14     0]
 [ 7444  1191   625    48]
 [ 3956   119   396     5]
 [ 3248   422   256    17]
 [ 9477  2180   704    87]
 [   62     1    23     0]
```



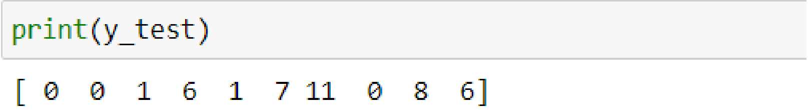
```
print(y_train)
```

```
[2 2 0 0 3 0 0 0 9 0 5 1 2 9 0 6 3 5 0 0 1 3 4 7 0 3 4 0 1 0 1 0 7 3 3 0 3
 8]
```



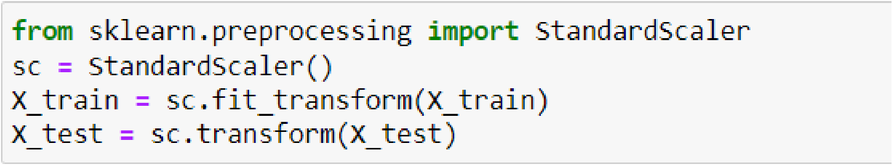
```
print(X_test)
```

```
[[ 548   11   41    0]
 [  19    0    7    0]
 [2644  317  345   13]
 [9120 1459  936   58]
 [8873  355  498   14]
 [8401 1596  947   64]
 [9888 2571  655  103]
 [  36    0    9    0]
 [9512 1998  641   80]
 [9489 1423  628   57]]
```

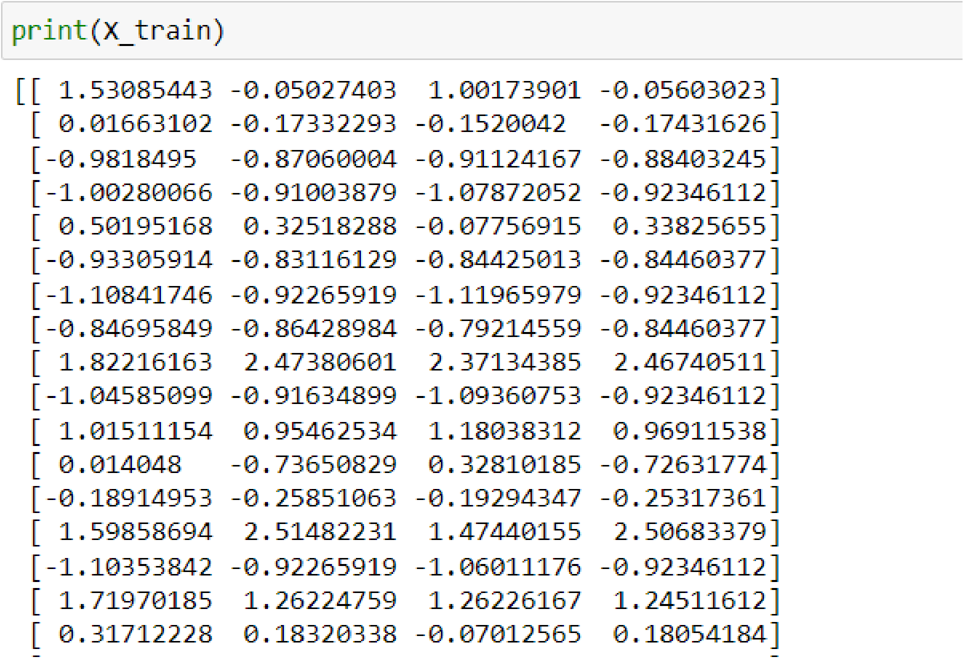


```
print(y_test)
```

```
[ 0  0  1  6  1  7 11  0  8  6]
```

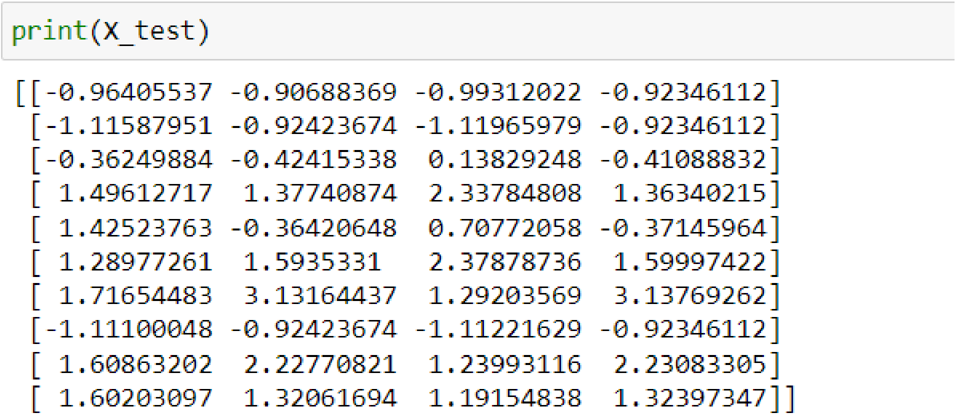


```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```



```python
print(X_train)
```

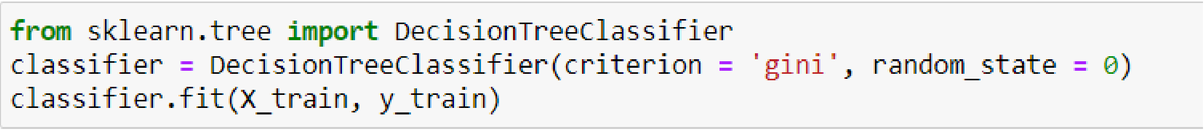
```
[[ 1.53085443 -0.05027403  1.00173901 -0.05603023]
 [ 0.01663102 -0.17332293 -0.1520042  -0.17431626]
 [-0.9818495  -0.87060004 -0.91124167 -0.88403245]
 [-1.00280066 -0.91003879 -1.07872052 -0.92346112]
 [ 0.50195168  0.32518288 -0.07756915  0.33825655]
 [-0.93305914 -0.83116129 -0.84425013 -0.84460377]
 [-1.10841746 -0.92265919 -1.11965979 -0.92346112]
 [-0.84695849 -0.86428984 -0.79214559 -0.84460377]
 [ 1.82216163  2.47380601  2.37134385  2.46740511]
 [-1.04585099 -0.91634899 -1.09360753 -0.92346112]
 [ 1.01511154  0.95462534  1.18038312  0.96911538]
 [ 0.014048   -0.73650829  0.32810185 -0.72631774]
 [-0.18914953 -0.25851063 -0.19294347 -0.25317361]
 [ 1.59858694  2.51482231  1.47440155  2.50683379]
 [-1.10353842 -0.92265919 -1.06011176 -0.92346112]
 [ 1.71970185  1.26224759  1.26226167  1.24511612]
 [ 0.31712228  0.18320338 -0.07012565  0.18054184]
```



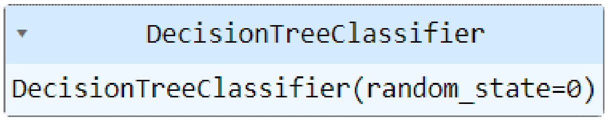
```python
print(X_test)
```

```
[[-0.96405537 -0.90688369 -0.99312022 -0.92346112]
 [-1.11587951 -0.92423674 -1.11965979 -0.92346112]
 [-0.36249884 -0.42415338  0.13829248 -0.41088832]
 [ 1.49612717  1.37740874  2.33784808  1.36340215]
 [ 1.42523763 -0.36420648  0.70772058 -0.37145964]
 [ 1.28977261  1.5935331   2.37878736  1.59997422]
 [ 1.71654483  3.13164437  1.29203569  3.13769262]
 [-1.11100048 -0.92423674 -1.11221629 -0.92346112]
 [ 1.60863202  2.22770821  1.23993116  2.23083305]
 [ 1.60203097  1.32061694  1.19154838  1.32397347]]
```

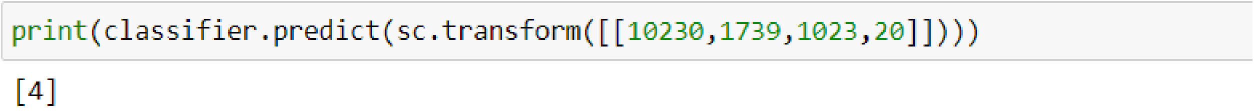
# Decision Tree



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



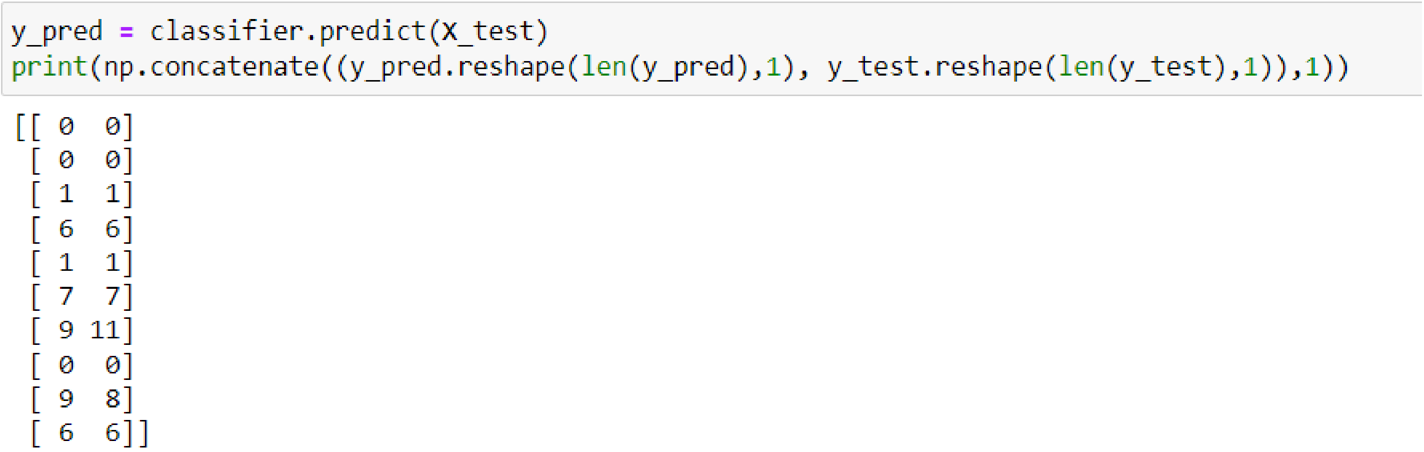
```
▾        DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```



```python
print(classifier.predict(sc.transform([[10230,1739,1023,20]])))
```

```
[4]
```

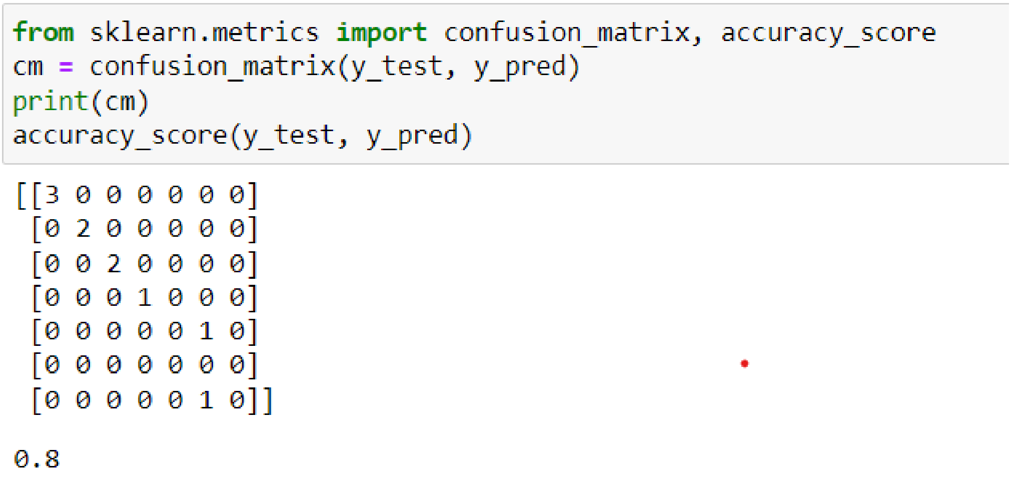
#Predicting the Test set results



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

```
[[ 0  0]
 [ 0  0]
 [ 1  1]
 [ 6  6]
 [ 1  1]
 [ 7  7]
 [ 9 11]
 [ 0  0]
 [ 9  8]
 [ 6  6]]
```

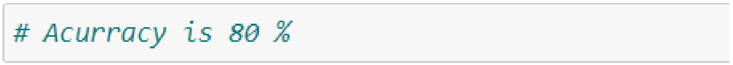
#Making the Confusion Matrix



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

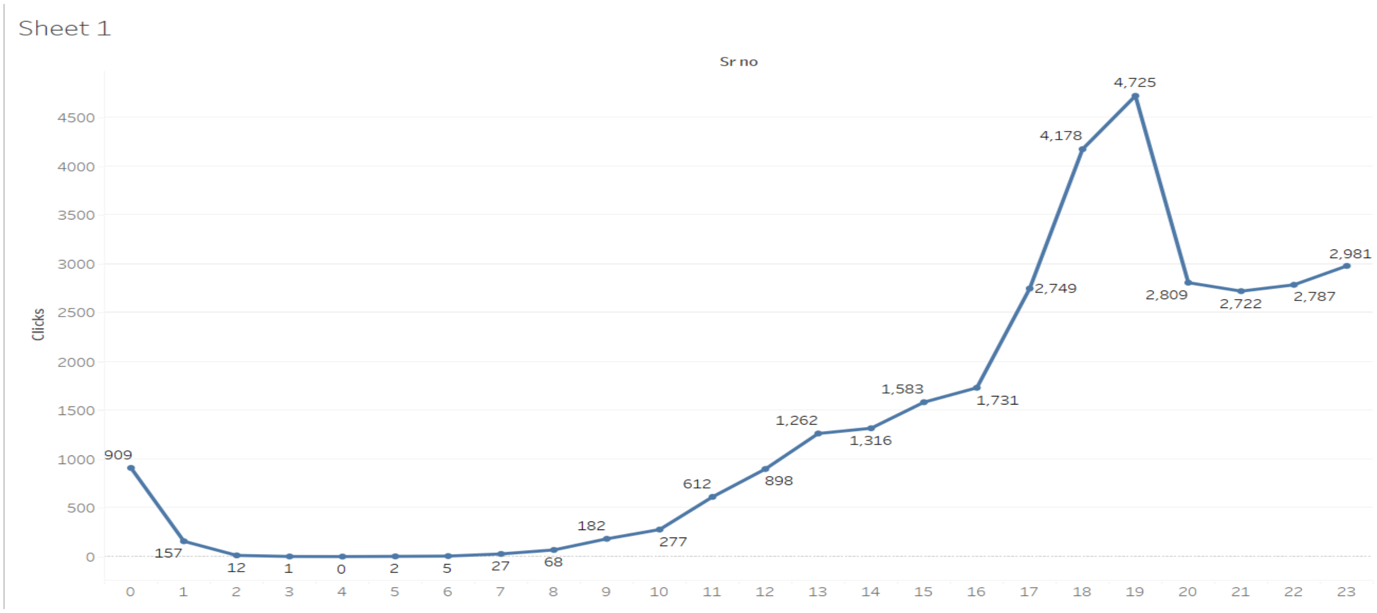
```
[[3 0 0 0 0 0 0]
 [0 2 0 0 0 0 0]
 [0 0 2 0 0 0 0]
 [0 0 0 1 0 0 0]
 [0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 0 1 0]]

0.8
```



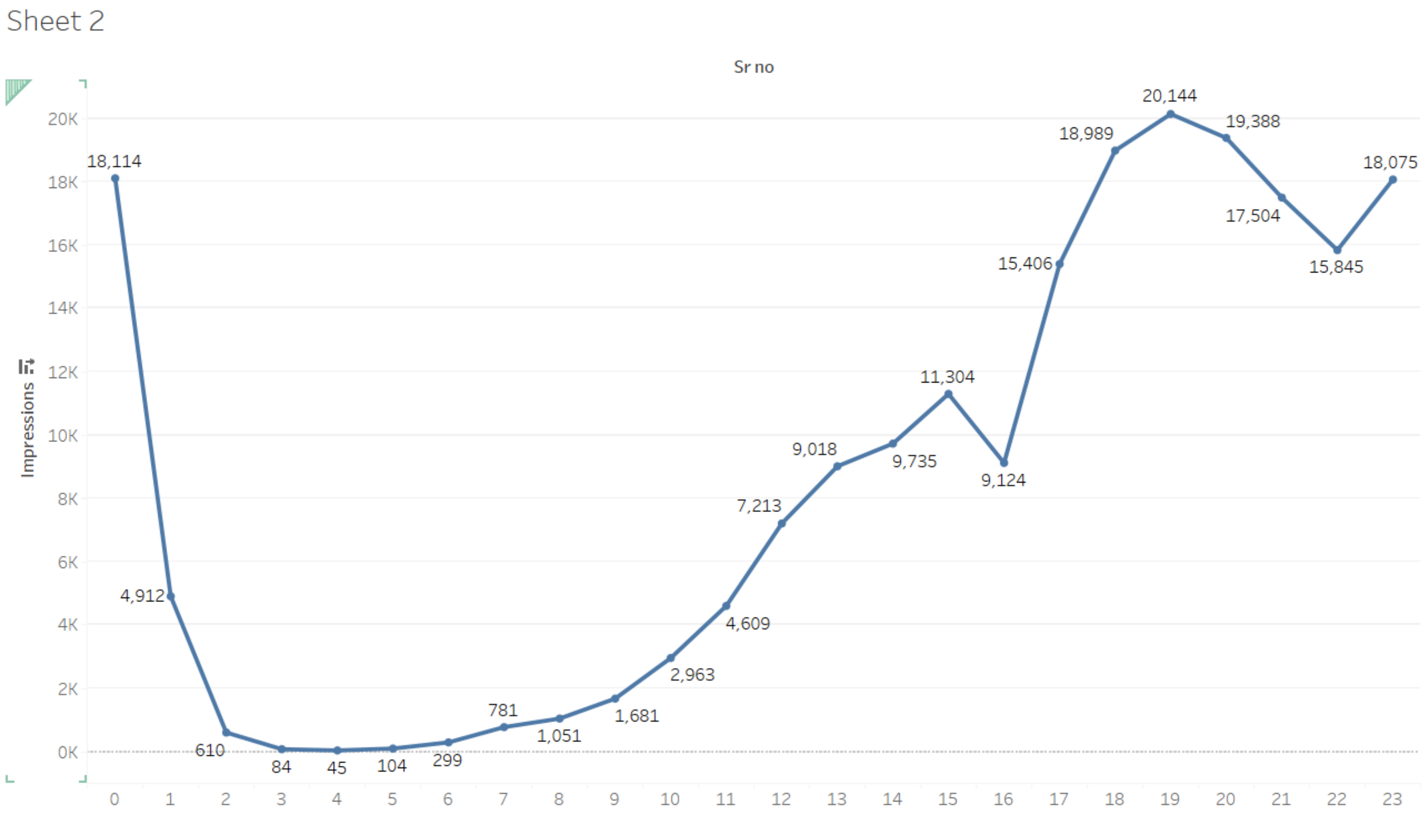
```
# Acurracy is 80 %
```

#Visualization



Insights of the Clicks Graph

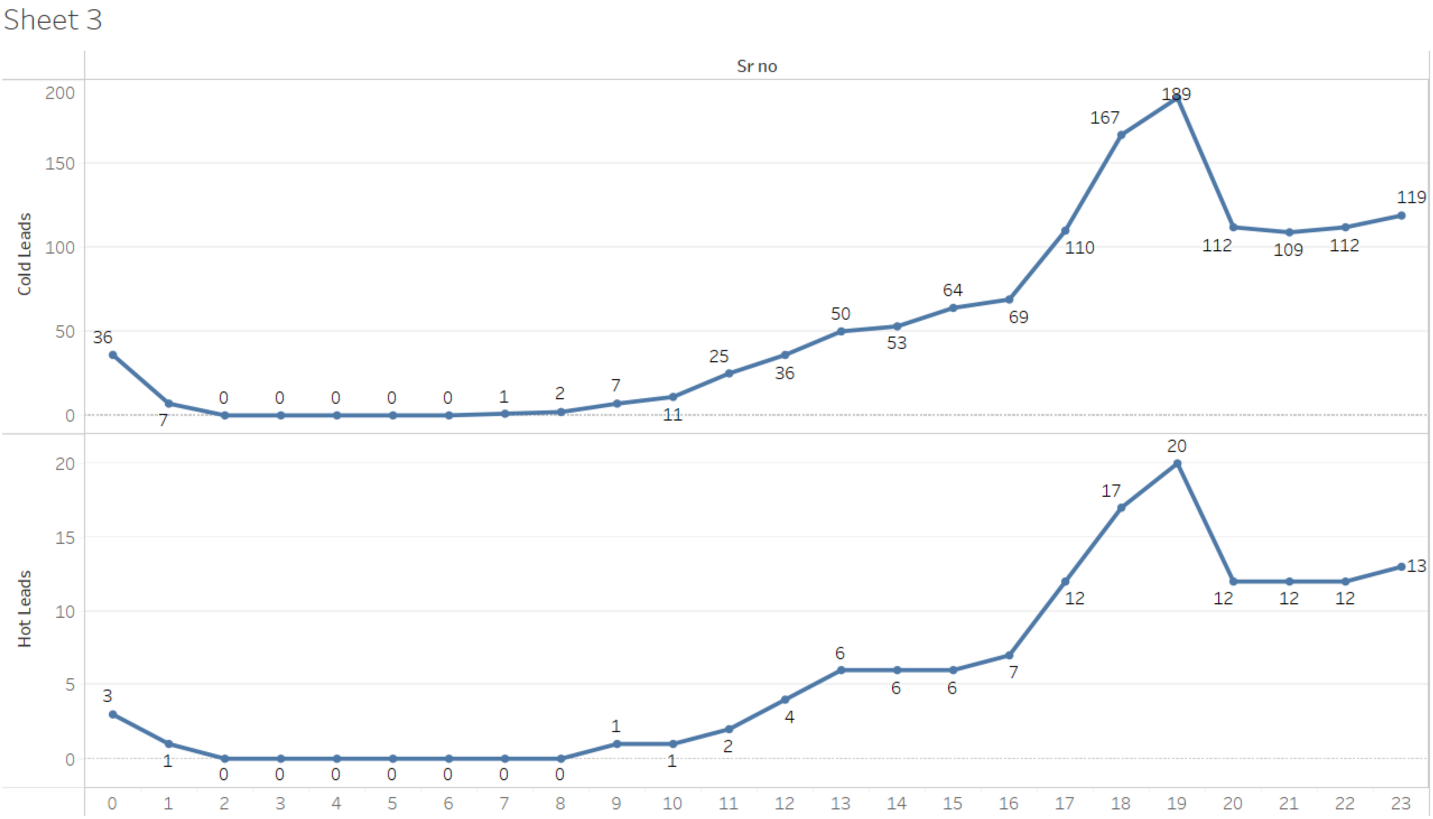
- Clicks **are high from 12:00 am to 1:00 am**. because lot of people are searching for new thing .
- Then from **1:30 am to 9:30 am** clicks are getting down as people are not online.
- From **10:00 am to 6:00 pm** clicks are **slightly high** than previous time.
- Then clicks are **going to increases** from **6:00 pm to 11:30 pm.**



Insights of the Impressions Graph

- Impressions are getting **high** at the night from **12:00 am to 1:00 am** as people are online**.**
- Then Impressions **are falling down after 1:30 am from 1:30 am to 6:00 am** because majority of the people are sleeping .
- From **6:30 am to 11:30 am** impressions are **slightly high and goes on increasing** because on that time lot of people are online and use sites.

- Then **from 12:00 pm to 11:30 pm** impressions are going to high and from evening 6:30 pm to **11:30 pm** impressions are very high. because more no of people is free.

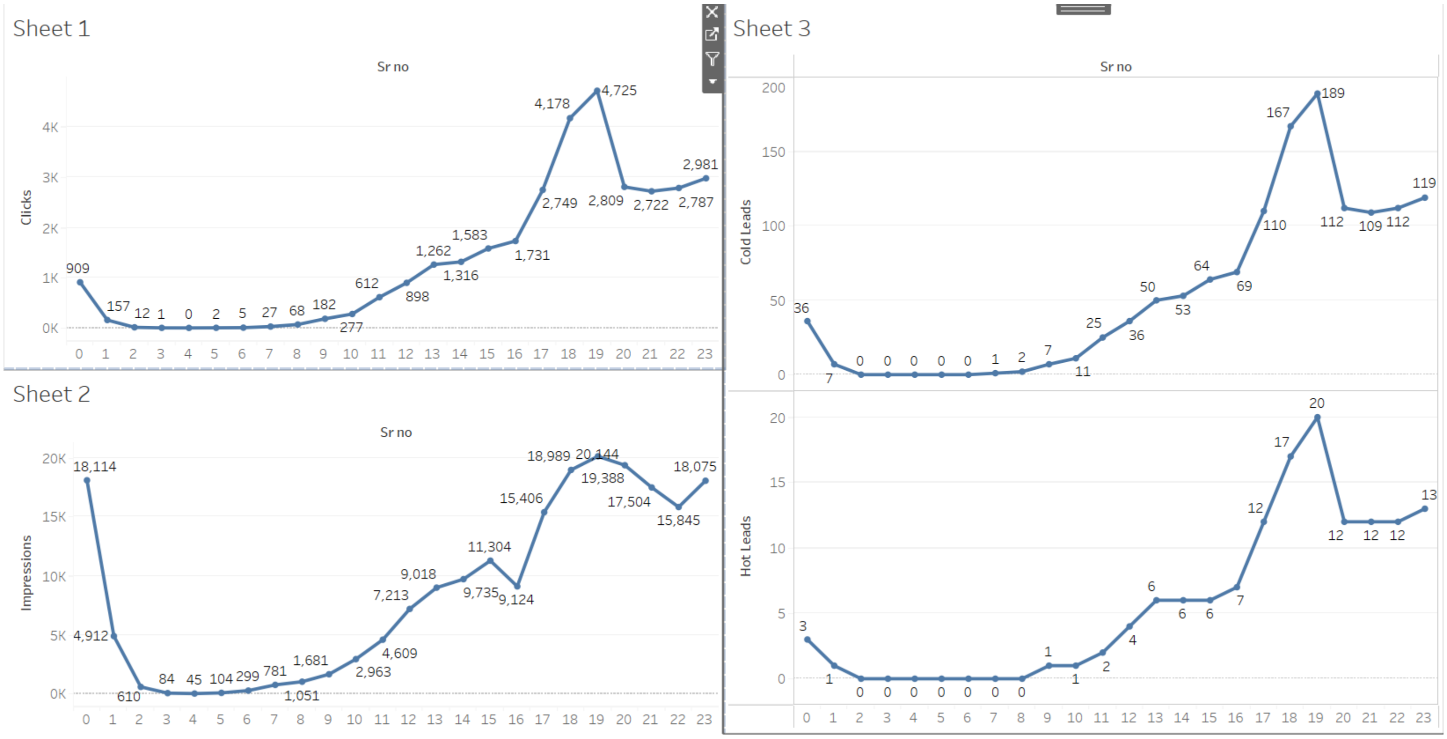


Insights of Cold Lead and Hot Lead Graph

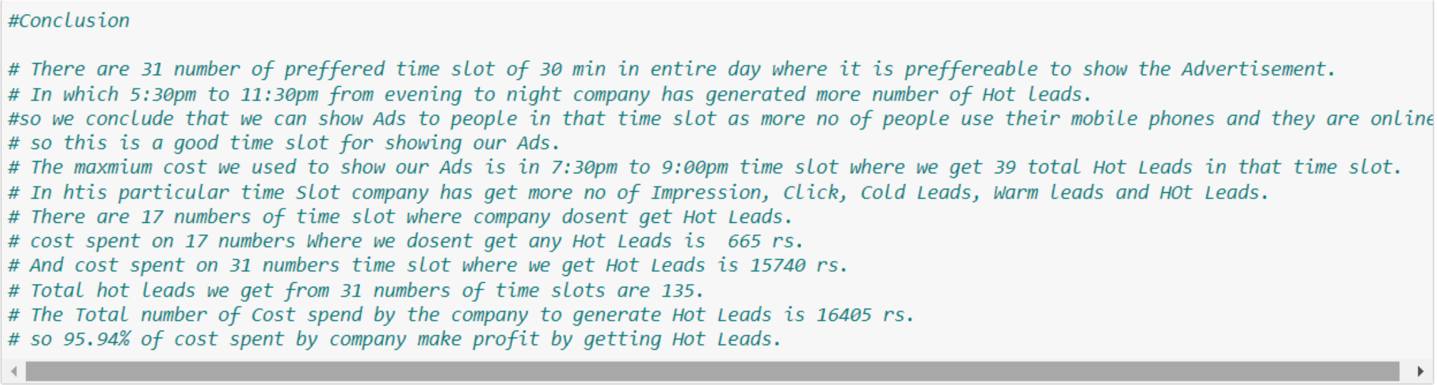
- Very few people are showing interest in that particular Ads **from 12:00 am to 1:00 am**
- Then after 1:00 am **not even one candidate** is showing interest in Ads from **1:30 am to 7:00 am** because lot of people are sleeping
- After 7:30 am 1,2 candidates are showings interest Ads in particular time slots from **7:30 am to 10:00 am**
- Then **from 10:30 am to 6:30** pm leads are **high than previous time slots** of leads.
- **From 12:00 am to 1:00 am** few people are showing interest and they like to purchase that service (they like to enroll in that Ad).
- Then from **1:30 am to 10:00 am no one** showing interest in that Ads .

- Then from **10:30 am to 5:00 pm** few people like 4 people are showing interest and they like to purchase that service (they like to enroll in that Ad).
- From **5:30 pm to 11:30** pm we get 135 Hot Leads who are showing interest and they like to purchase that service.

#DASHBOARD

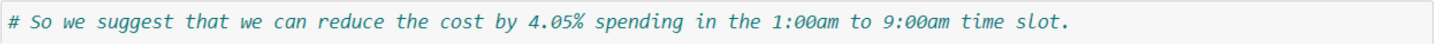


Conclusion



```
#Conclusion

# There are 31 number of preffered time slot of 30 min in entire day where it is preffereable to show the Advertisement.
# In which 5:30pm to 11:30pm from evening to night company has generated more number of Hot leads.
#so we conclude that we can show Ads to people in that time slot as more no of people use their mobile phones and they are online
# so this is a good time slot for showing our Ads.
# The maxmium cost we used to show our Ads is in 7:30pm to 9:00pm time slot where we get 39 total Hot Leads in that time slot.
# In htis particular time Slot company has get more no of Impression, Click, Cold Leads, Warm leads and HOt Leads.
# There are 17 numbers of time slot where company dosent get Hot Leads.
# cost spent on 17 numbers Where we dosent get any Hot Leads is  665 rs.
# And cost spent on 31 numbers time slot where we get Hot Leads is 15740 rs.
# Total hot leads we get from 31 numbers of time slots are 135.
# The Total number of Cost spend by the company to generate Hot Leads is 16405 rs.
# so 95.94% of cost spent by company make profit by getting Hot Leads.
```



```
# So we suggest that we can reduce the cost by 4.05% spending in the 1:00am to 9:00am time slot.
```