BAN 4550 - 02

Group 6

Nisarg Thakkar, Dhiraj Sharma, Vikas Jangra

Final Project - Understanding variables affecting CTR of digital ad campaigns

Dataset Link: https://www.kaggle.com/louischen7/2020-digix-advertisement-ctr-prediction

Project Description:

Click Through Rate in Digital Advertising is one of the most important metrics. It measures the number of clicks a particular advertiser received on their ad per the number of impressions (impressions = number of times the ad was shown to a user). It gives a broad view to the advertiser on how well their ad is received by the users. Following are some of the reasons why CTR is important for advertisers:

- Gives a base of potential users who will convert
- Helps create a benchmark rate for future ad campaigns
- Aids in understanding which ad copy works better, from a call-to-action point of view
- Understand the dynamics and behavior of the target audience

Research Plan:

Significance of Study:

This study is extremely important for digital marketing managers of various companies that use internet marketing. This study deep dives into various variables that impact the click rate of a digital ad campaign. It further explores and explains which ad type is more clickable and also helps understand user behavior basis demographic variables such as age, gender, city rank, etc. This study will help the manager plan future digital campaigns keeping in mind the effect of each variable on click

Research Objectives:

With the above research we will figure out which variables plays the key role in determining the click rate. This research also determines how a particular value or feature of a variable attracts more clicks as compared to the rest. And, we also proved a correlation between two variables and click rate.

Anticipated Results

We strongly believe that certain ad types (inter_ad) attracts more clicks than the rest. And, similarly various variables like age, gender, net type, city rank are independently or coherently has an affect on click rate.

Objective:

To understand which are the various variables that have an impact on the click rate and to what extent.

Assumptions:

1. Label: represents the status of clicking.

   0 - not clicked

   1 - Clicked

1. inter_type_cd: represents the display form of the ad.

   3 - Image

   4 - GIF

   5 - Videos

1. Gender:

   0 - Female

   1 - Male

   2 - Others

1. Slot_id: represents the placement of the ad on the screen.

   11 to 22 - represents the various location of the ad placement.

1. Net_type: represents the status of the net.

   2 to 6 - 2 being the weakest and 6 being the strongest.

1. City_rank: represents Level of the resident city of a user

   2 to 5: 2 being the lowest and 5 being highest.

In [1]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt
import random
import statsmodels.formula.api as smf
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.model_selection import StratifiedShuffleSplit
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint, uniform
```

About the Data

In [37]:

```
#Since the dataset contains 1 millions entries, we are selecting the random 1

file='/Users/vikasjangra/Documents/MS BA/Fall 2021/BAN 4550 Analytics Program
num_lines = sum(1 for l in open(file))
num_lines
skip = sorted(random.sample(range(1,num_lines+1),num_lines-100000))
orig_file=pd.read_csv(file , skiprows=skip)
pd.set_option('max_columns', None)
df=orig_file.copy()
df.head()
```

Out[37]:

	label	uid	task_id	adv_id	creat_type_cd	adv_prim_id	dev_id	inter_type_cd	slot_id
0	0	1718762	5224	4790	6	175	60	4	18
1	0	1567652	5086	6793	7	193	37	5	21
2	0	1068169	1374	3702	7	171	27	5	12
3	0	1216872	2112	6869	7	207	17	5	12
4	0	1198258	1177	5183	4	112	60	3	14

Data Dictionary

1. label - Label
2. uid - Unique user ID after data anonymization
3. task_id - Unique ID of an ad task
4. adv_id - Unique ID of an ad material
5. creat_type_cd - Unique ID of an ad creative type
6. adv_prim_id - Advertiser ID of an ad task
7. dev_id - Developer ID of an ad task
8. inter_typ_cd - Display form of an ad material
9. slot_id - Ad slot ID
10. spread_app_id - App ID of an ad task
11. tags - App tag of an ad task
12. app_first_class - App level-1 category of an ad task
13. app_second_class - App level-2 category of an ad task
14. age - User age
15. city - Resident city of a user
16. city_rank - Level of the resident city of a user
17. device_name - Phone model used by a user
18. device_size - Size of the phone used by a user
19. career - User occupation
20. gender - User gender
21. net_type - Network status when a behavior occurs
22. residence - Resident province of a user
23. his_app_size - App storage size
24. his_on_shelf_time - Release time
25. app_score - App rating score
26. emui_dev - EMUI version
27. list_time - Model release time
28. device_price - Device price
29. up_life_duration - HUAWEI ID lifecycle
30. up_membership_grade - Service membership level
31. membership_life_duration - Membership lifecycle
32. consume_purchase - Paid user tag
33. communication_onlinerate - Active time by mobile phone
34. communication_avgonline_30d - Daily active time by mobile phone
35. indu_name - Ad industry information
36. pt_d - Date when a behavior occurs

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 36 columns):
 #   Column                       Non-Null Count   Dtype 
---  ------                       --------------   ----- 
 0   label                        100000 non-null  int64 
 1   uid                          100000 non-null  int64 
 2   task_id                      100000 non-null  int64 
 3   adv_id                       100000 non-null  int64 
 4   creat_type_cd                100000 non-null  int64 
 5   adv_prim_id                  100000 non-null  int64 
 6   dev_id                       100000 non-null  int64 
 7   inter_type_cd                100000 non-null  int64 
 8   slot_id                      100000 non-null  int64 
 9   spread_app_id                100000 non-null  int64 
 10  tags                         100000 non-null  int64 
 11  app_first_class              100000 non-null  int64 
 12  app_second_class             100000 non-null  int64 
 13  age                          100000 non-null  int64 
 14  city                         100000 non-null  int64 
 15  city_rank                    100000 non-null  int64 
 16  device_name                  100000 non-null  int64 
 17  device_size                  100000 non-null  int64 
 18  career                       100000 non-null  int64 
 19  gender                       100000 non-null  int64 
 20  net_type                     100000 non-null  int64 
 21  residence                    100000 non-null  int64 
 22  his_app_size                 100000 non-null  int64 
 23  his_on_shelf_time            100000 non-null  int64 
 24  app_score                    100000 non-null  int64 
 25  emui_dev                     100000 non-null  int64 
 26  list_time                    100000 non-null  int64 
 27  device_price                 100000 non-null  int64 
 28  up_life_duration             100000 non-null  int64 
 29  up_membership_grade          100000 non-null  int64 
 30  membership_life_duration     100000 non-null  int64 
 31  consume_purchase             100000 non-null  int64 
 32  communication_onlinerate     100000 non-null  object
 33  communication_avgonline_30d  100000 non-null  int64 
 34  indu_name                    100000 non-null  int64 
 35  pt_d                         100000 non-null  int64 
dtypes: int64(35), object(1)
memory usage: 27.5+ MB
```

Dataset Information

Data Cleaning and information

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
label                           0
uid                             0
task_id                         0
adv_id                          0
creat_type_cd                   0
adv_prim_id                     0
dev_id                          0
inter_type_cd                   0
slot_id                         0
spread_app_id                   0
tags                            0
app_first_class                 0
app_second_class                0
age                             0
city                            0
city_rank                       0
device_name                     0
device_size                     0
career                          0
gender                          0
net_type                        0
residence                       0
his_app_size                    0
his_on_shelf_time               0
app_score                       0
emui_dev                        0
list_time                       0
device_price                    0
up_life_duration                0
up_membership_grade             0
membership_life_duration        0
consume_purchase                0
communication_onlinerate        0
communication_avgonline_30d     0
indu_name                       0
pt_d                            0
dtype: int64
```

There are no null values in the dataset.

In [5]:

```
# Creating the histogram of all the columns in the dataset.
#df.hist(figsize=(40,30), bins=50)
```

Creating a correlation of the dataset.

In [6]:

```
corr_df=df.corr()
corr_df
```

Out[6]:

	label	uid	task_id	adv_id	creat_type_cd	ad
label	1.000000	-0.000822	-0.001574	0.011048	0.003622	

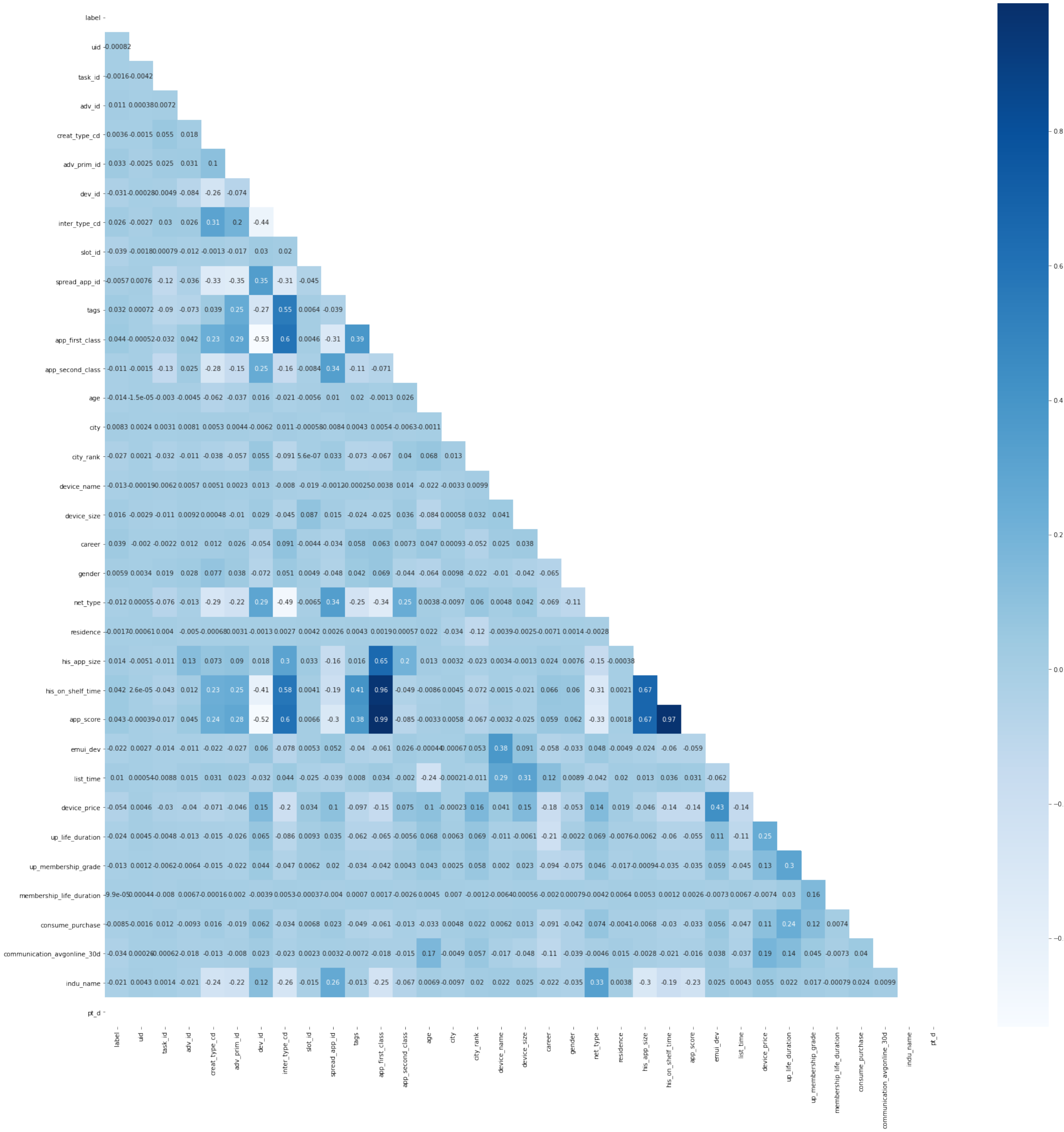
uid	-0.000822	1.000000	-0.004188	0.000383	-0.001477
task_id	-0.001574	-0.004188	1.000000	0.007158	0.055389
adv_id	0.011048	0.000383	0.007158	1.000000	0.018440
creat_type_cd	0.003622	-0.001477	0.055389	0.018440	1.000000
adv_prim_id	0.033426	-0.002498	0.024770	0.031360	0.104981
dev_id	-0.030881	-0.000278	-0.004908	-0.083595	-0.257759
inter_type_cd	0.025719	-0.002722	0.029702	0.026320	0.305151
slot_id	-0.039166	-0.001809	0.000793	-0.011897	-0.001265
spread_app_id	-0.005666	0.007629	-0.123420	-0.035512	-0.333620
tags	0.032221	0.000724	-0.090149	-0.073110	0.038874
app_first_class	0.044423	-0.000523	-0.032445	0.041700	0.232086
app_second_class	-0.010865	-0.001472	-0.132964	0.025260	-0.276676
age	-0.013878	-0.000015	-0.002985	-0.004548	-0.062400
city	0.008309	0.002360	0.003054	0.008071	0.005340
city_rank	-0.027241	0.002107	-0.031996	-0.011040	-0.037858
device_name	-0.013132	-0.000195	-0.006199	0.005696	0.005126
device_size	0.015765	-0.002885	-0.010901	0.009184	0.000483
career	0.038805	-0.002014	-0.002207	0.012092	0.012331
gender	0.005860	0.003417	0.018899	0.028326	0.077419
net_type	-0.012235	0.000546	-0.075503	-0.012668	-0.286825
residence	-0.001714	-0.000609	0.003975	-0.005036	-0.000677
his_app_size	0.013738	-0.005118	-0.011141	0.125116	0.073369

his_on_shelf_time	0.042348	0.000026	-0.042947	0.012215	0.225276
app_score	0.042690	-0.000385	-0.016880	0.044833	0.241419
emui_dev	-0.022404	0.002717	-0.014120	-0.011142	-0.021761
list_time	0.010213	0.000541	-0.008820	0.014821	0.031079
device_price	-0.053862	0.004615	-0.029966	-0.039779	-0.071297
up_life_duration	-0.024188	0.004475	-0.004754	-0.012596	-0.014871
up_membership_grade	-0.012772	0.001235	-0.006241	-0.006405	-0.015290
membership_life_duration	-0.000099	0.000442	-0.007989	0.006703	-0.000164
consume_purchase	-0.008522	-0.001639	0.012397	-0.009349	0.015822
communication_avgonline_30d	-0.033593	0.000257	-0.000625	-0.018423	-0.012993
indu_name	-0.021086	0.004281	0.001426	-0.021428	-0.244144
pt_d	NaN	NaN	NaN	NaN	NaN

Creating a heatmap relationship among a the variables in the dataset.

In [7]:

```
plt.figure(figsize=(30, 30))      # Credit is to be given to the coder.
mask = np.zeros_like(corr_df) #masking the null values of the upper half tria
mask[np.triu_indices_from(mask)] = True
corr_heatmap=sns.heatmap(df.corr(),cmap="Blues", annot=True,mask=mask) #creat
```



The heat map predicts the correlations among the variables here.

Some variables provide cardinal relation among the variables that are useful for the decisons makers.

Upon careful review of the above heatmap, it is evident that App rating score (app_score) has a strong positive relation to display form of an ad (inter_typ_cd) - it might suggest that some ad forms are better than others to increase popularity of an app.

Calculating the total number of the clicks and the not clicks.

0 --> Not Clicked

1 --> Clicked.

In [8]:

```
click_count=df['label'].value_counts()
click_count

#click_count.cumsum()
```

Out[8]:

```
0    96297
1     3703
Name: label, dtype: int64
```

In [9]:

```
rel_freq_click=click_count/len(df)
rel_freq_click
```

Out[9]:

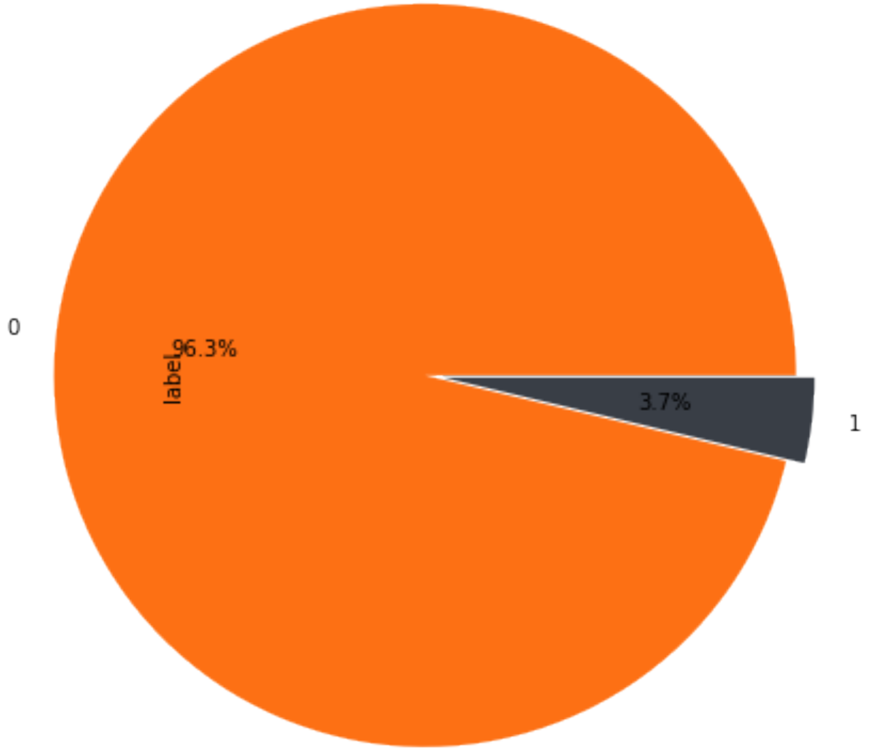
```
0    0.96297
1    0.03703
Name: label, dtype: float64
```

In [64]:

```
click_count.plot(kind='pie',autopct='%1.1f%%',radius=2,colors=colors,explode=
```

Out[64]:

```
<AxesSubplot:ylabel='label'>
```



Among all the users (100,000 user), approximately of 3.7% user, i.e., 3700 user clicks on the advertisement published.

Hypotheses 1

H0 – The type of an ad (video, image, GIF, etc.) has no effect on the click rate.

H1 – The type of an ad (video, image, GIF, etc.) influences the click rate.

In [11]:

```
df.inter_type_cd.unique()
```

Out[11]:

```
array([5, 3, 4])
```

In [12]:

```
#calculating the number of clicks at different platforms.

len_Clicked_5 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5)])
len_notClicked_5 = len(df.loc[(df.label == 0) & (df.inter_type_cd == 5)])


len_Clicked_4 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4)])
len_notClicked_4 = len(df.loc[(df.label == 0) & (df.inter_type_cd == 4)])

len_Clicked_3 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3)])
len_notClicked_3 = len(df.loc[(df.label == 0) & (df.inter_type_cd == 3)])

print("len_Clicked_5 -->  ",len_Clicked_5)
print('len_notClicked_5 -->',len_notClicked_5)
print()
print("len_Clicked_4 -->  ",len_Clicked_4)
print('len_notClicked_4 -->',len_notClicked_4)
print()
print("len_Clicked_3 -->  ",len_Clicked_3)
print('len_notClicked_3 -->',len_notClicked_3)

df_Clicked=[len_Clicked_5,len_Clicked_4,len_Clicked_3]
df_notClicked=[len_notClicked_5,len_notClicked_4,len_notClicked_3]

df_clicks=[df_Clicked,df_notClicked]
```

```
len_Clicked_5 -->   3078
len_notClicked_5 --> 77495

len_Clicked_4 -->   479
len_notClicked_4 --> 9257

len_Clicked_3 -->   146
len_notClicked_3 --> 9545
```

In [13]:

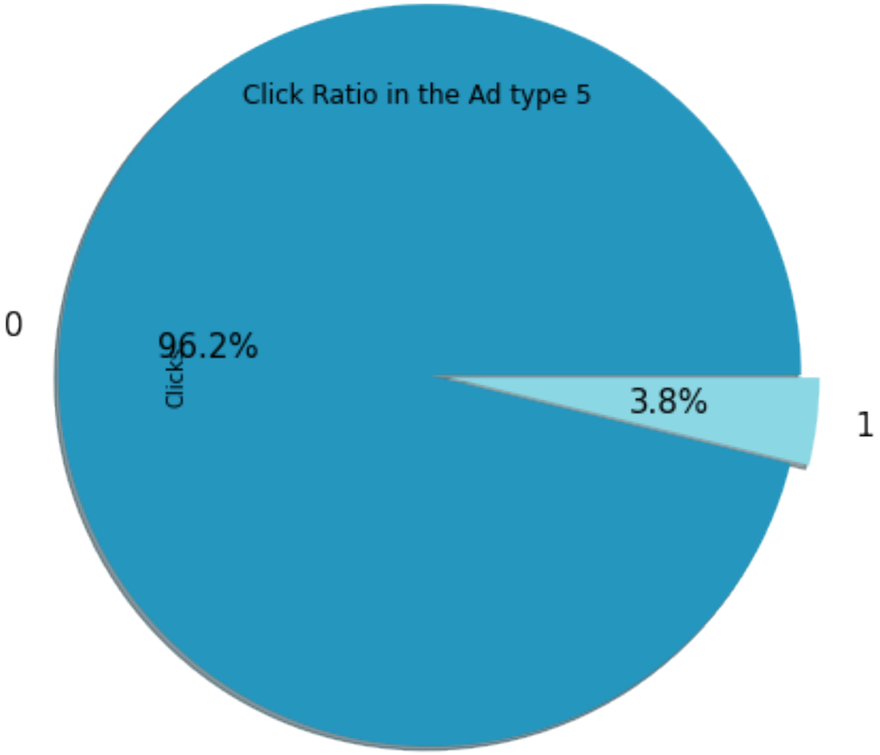
```
value_5=[len_notClicked_5,len_Clicked_5]
df_5 = pd.DataFrame(value_5, columns = ['Clicks'])
df_5
```

Out[13]:

	Clicks
0	77495
1	3078

In [67]:

```
plot = df_5.plot.pie( title="Click Ratio in the Ad type 5", legend=False, \
                        autopct='%1.1f%%',shadow=True,radius=2,subplots='True',exp
```



In [15]:

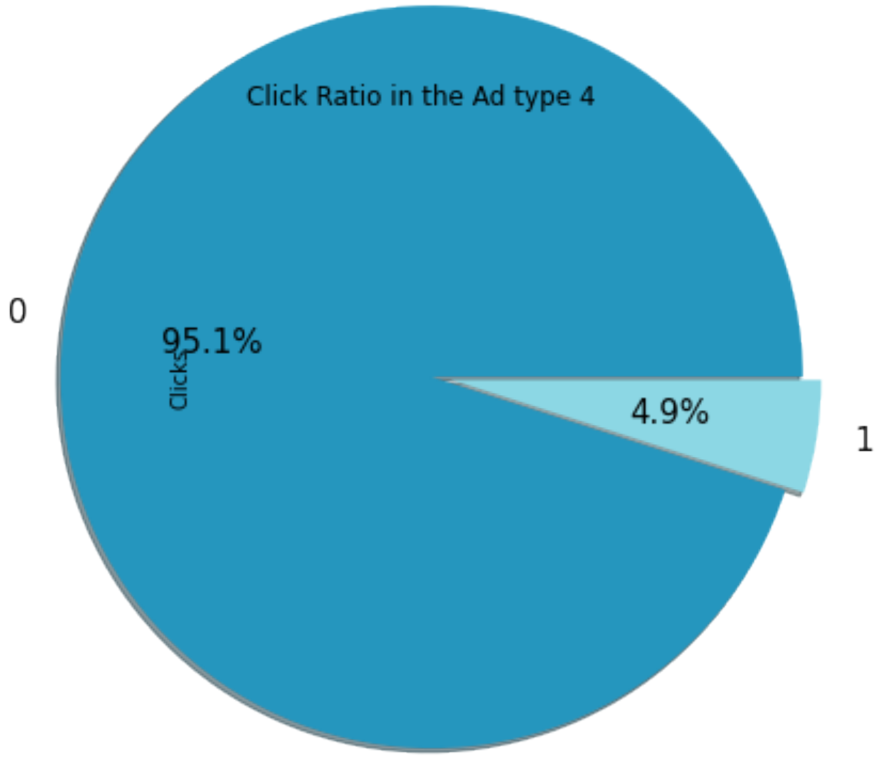
```
value_4=[len_notClicked_4,len_Clicked_4]
df_4 = pd.DataFrame(value_4, columns = ['Clicks'])
df_4
```

Out[15]:

	Clicks
0	9257
1	479

In [68]:

```
plot = df_4.plot.pie( title="Click Ratio in the Ad type 4", legend=False, \
                   autopct='%1.1f%%',shadow=True,subplots='True',explode=(0,
```



In [17]:

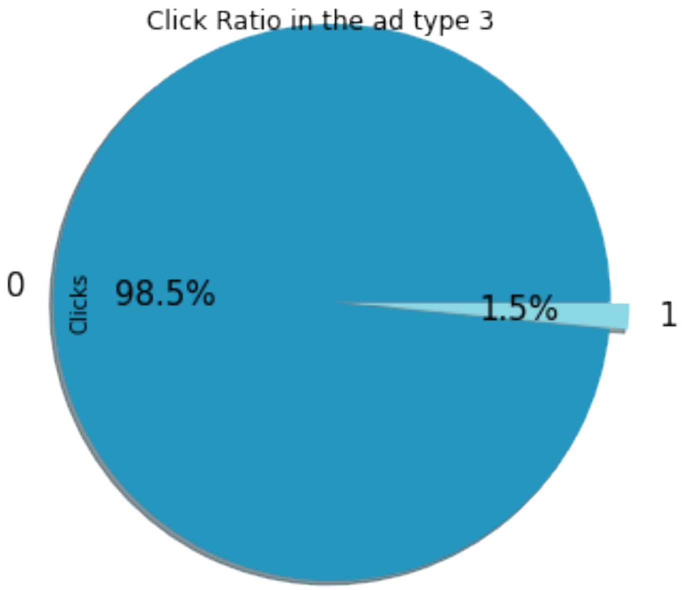
```
value_3=[len_notClicked_3,len_Clicked_3]
df_3 = pd.DataFrame(value_3, columns = ['Clicks'])
df_3
```

Out[17]:

	Clicks
0	9545
1	146

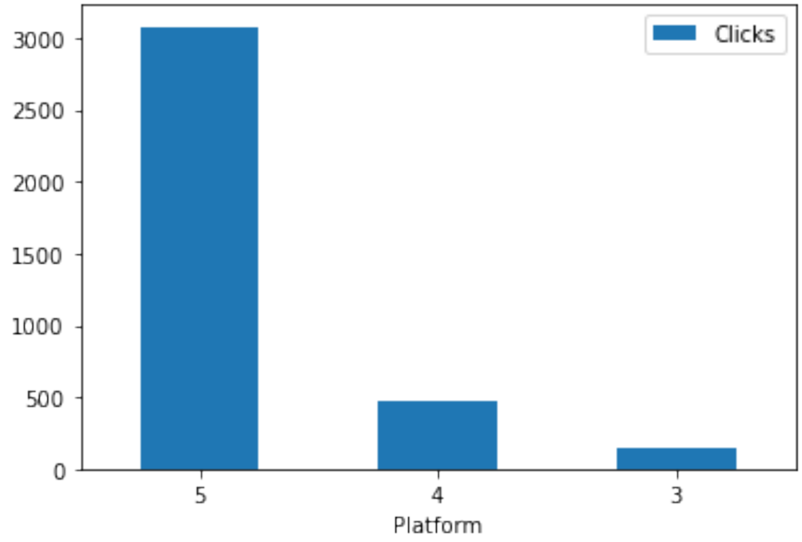
In [73]:

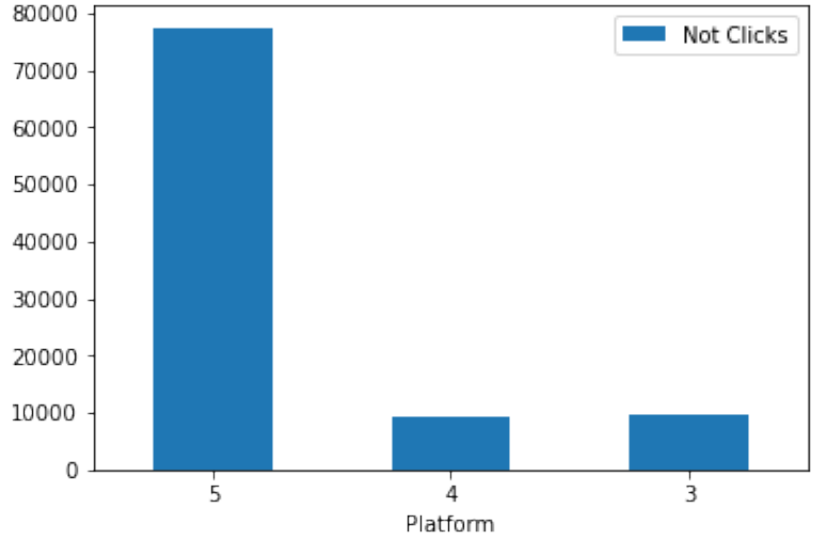
```
plot = df_3.plot.pie( title="Click Ratio in the ad type 3", legend=False, \
                   autopct='%1.1f%%',shadow=True,subplots='True',explode=(0,
```

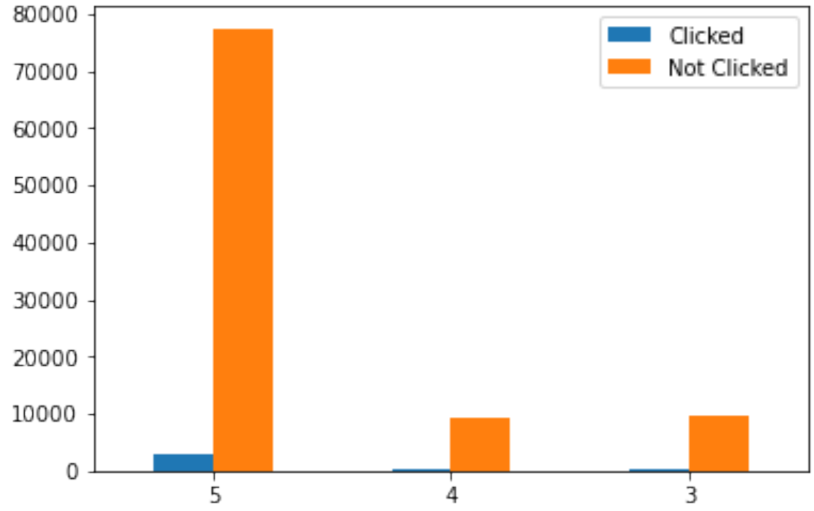


In [19]:

```
df_clicks = pd.DataFrame({'Platform':['5', '4', '3'], 'Clicks':[len_Clicked_5
ax1 = df_clicks.plot.bar(x='Platform', y='Clicks', rot=0)

df_not_clicks = pd.DataFrame({'Platform':['5', '4', '3'], 'Not Clicks':[len_n
ax2 = df_not_clicks.plot.bar(x='Platform', y='Not Clicks', rot=0)

Platform = ['5', '4', '3']
clicked = [len_Clicked_5 ,len_Clicked_4,len_Clicked_3]
notclicked = [len_notClicked_5,len_notClicked_4,len_notClicked_3]
df = pd.DataFrame({'Clicked': clicked,'Not Clicked': notclicked}, index=Platf
ax = df.plot.bar(rot=0)
#We have to do the truncating here in the last bar plot.
```


Clicks
3000
2500
2000
1500
1000
500
0
5
4
3
Platform


Not Clicks
80000
70000
60000
50000
40000
30000
20000
10000
0
5
4
3
Platform


Clicked
Not Clicked
80000
70000
60000
50000
40000
30000
20000
10000
0
5
4
3

Findings:

Ad type 5 has got the highest number of clicks and has seen the highest number of non-clicks. However, compared to the other ad types, this ad type has worked best and can be considered as our most preferred ad type amongst users.

Hypotheses 2

H0 – The city rank of the user has no effect on the click rate.

H1 – The city rank of the user influences the click rate.

In [20]:

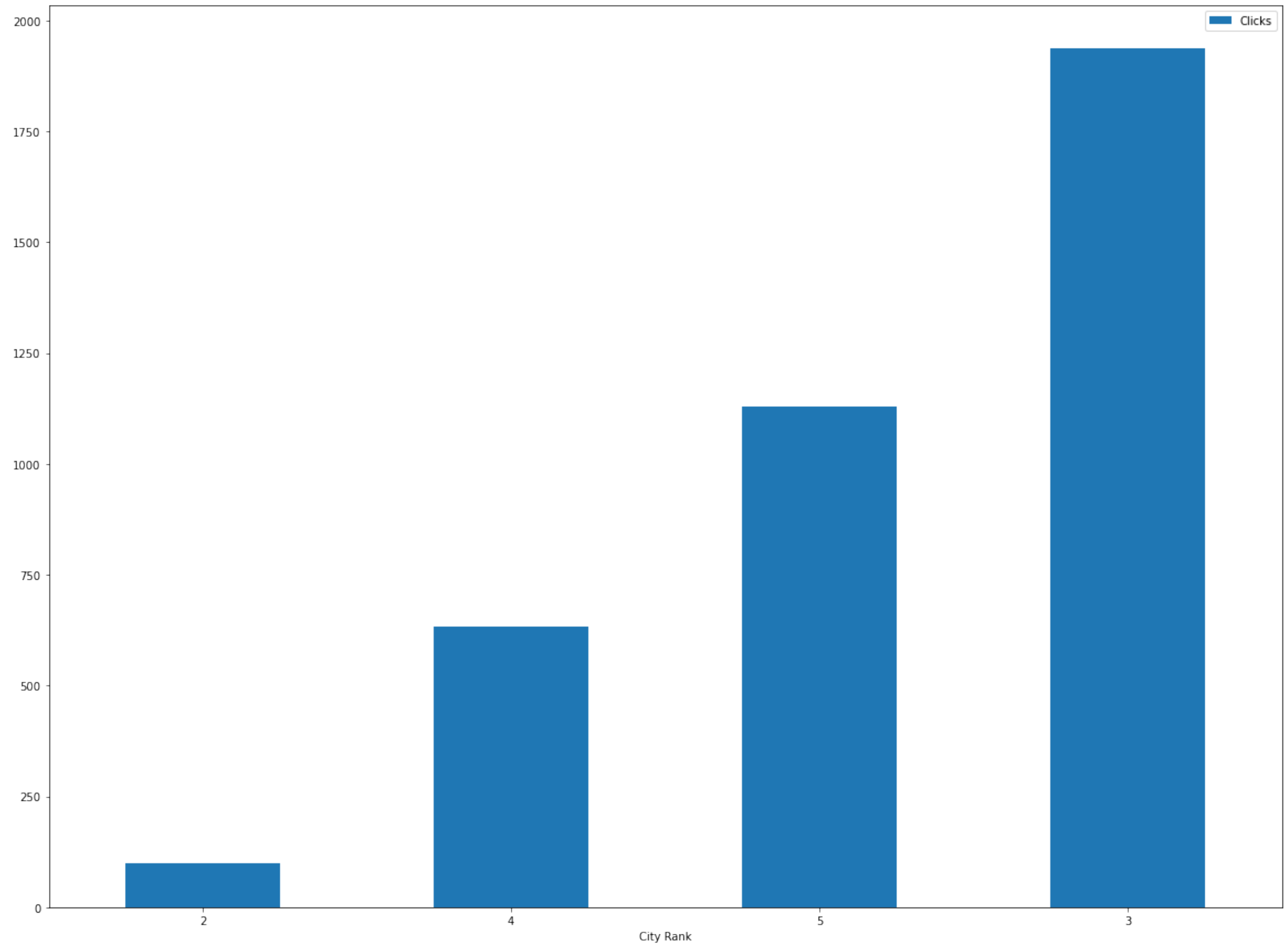
```
#df_sizes = pd.DataFrame(df['device_size'])
#df_sizes['device_size'] = pd.cut(x=df_sizes['device_size'], bins=[100, 150,
```

In [25]:

```
len_city_rank_2 = len(df.loc[(df.label == 1) & (df.city_rank == 2)])
len_city_rank_3 = len(df.loc[(df.label == 1) & (df.city_rank == 3)])
len_city_rank_4 = len(df.loc[(df.label == 1) & (df.city_rank== 4)])
len_city_rank_5 = len(df.loc[(df.label == 1) & (df.city_rank== 5)])
```

In [28]:

```
city_rank={'2':len_city_rank_2,'3':len_city_rank_3,'4':len_city_rank_4,'5':le

sorted_values1 = sorted(city_rank.values()) # Sort the values
sorted_dict1 = {}

for i in sorted_values1:
    for k in city_rank.keys():
        if city_rank[k] == i:
            sorted_dict1[k] = city_rank[k]
            break

#print(sorted_dict)
values_slot_dict1 = sorted_dict1.values()
values_list1 = list(values_slot_dict1)
#print(values_list)


# Creating a list of the sorted KEYS of the dict. KEYS not VALUES

def getList1(dict1):
    return dict1.keys()
# Driver program
dict1 = sorted_dict1
key_list_sorted_dict1=getList(dict1)
key_list1 = list(key_list_sorted_dict1)
#print(key_list)



# Creating the bar plot
df_not_clicks1 = pd.DataFrame({'Clicks':values_list1, 'City Rank':key_list1})
ax21 = df_not_clicks1.plot.bar(x='City Rank', y='Clicks', rot=0, figsize=(20,
```



Findings:

Cities with rank 3 has the highest number of clicks followed by cities with rank 5 and 4 respectively.

Hypotheses 3

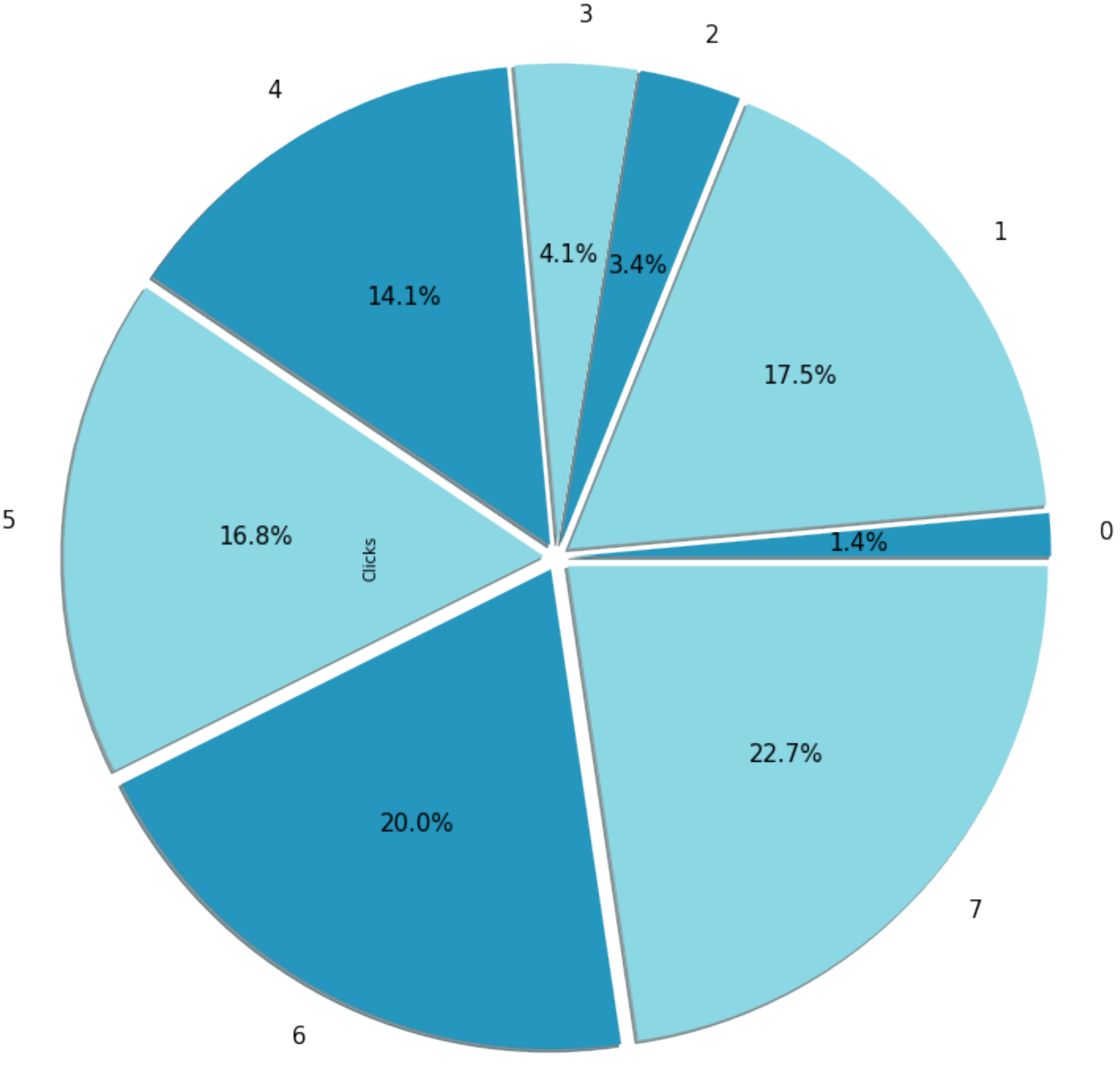
H0 - Age group of the user, type of ad and click are correlated.

H1- Age group of the user, type of ad and click are not correlated.

In [29]:

```
#len_Clicked_5
#calculating the number of the clicks of the when type is 5 and different age
len_Clicked_5_age0 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_age1 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_age2 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_age3 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_age4 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_age5 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_age6 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_age7 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (

print(len_Clicked_5_age0)
print(len_Clicked_5_age1)
print(len_Clicked_5_age2)
print(len_Clicked_5_age3)
print(len_Clicked_5_age4)
print(len_Clicked_5_age5)
print(len_Clicked_5_age6)
print(len_Clicked_5_age7)

#df_age_5=

clicked_age_5 = [len_Clicked_5_age0,len_Clicked_5_age1,len_Clicked_5_age2,len_
age=['-1','1','2','3','4','5','6','7']
df_clicked_age_5 = pd.DataFrame(clicked_age_5, columns = ['Clicks'])


colors = ["#2596be","#8cd7e4"]
plot = df_clicked_age_5.plot.pie(legend=False, \
                   autopct='%1.1f%%',shadow=True,radius=3.5,subplots='True',e
```

```
46
555
109
130
448
535
634
720
```



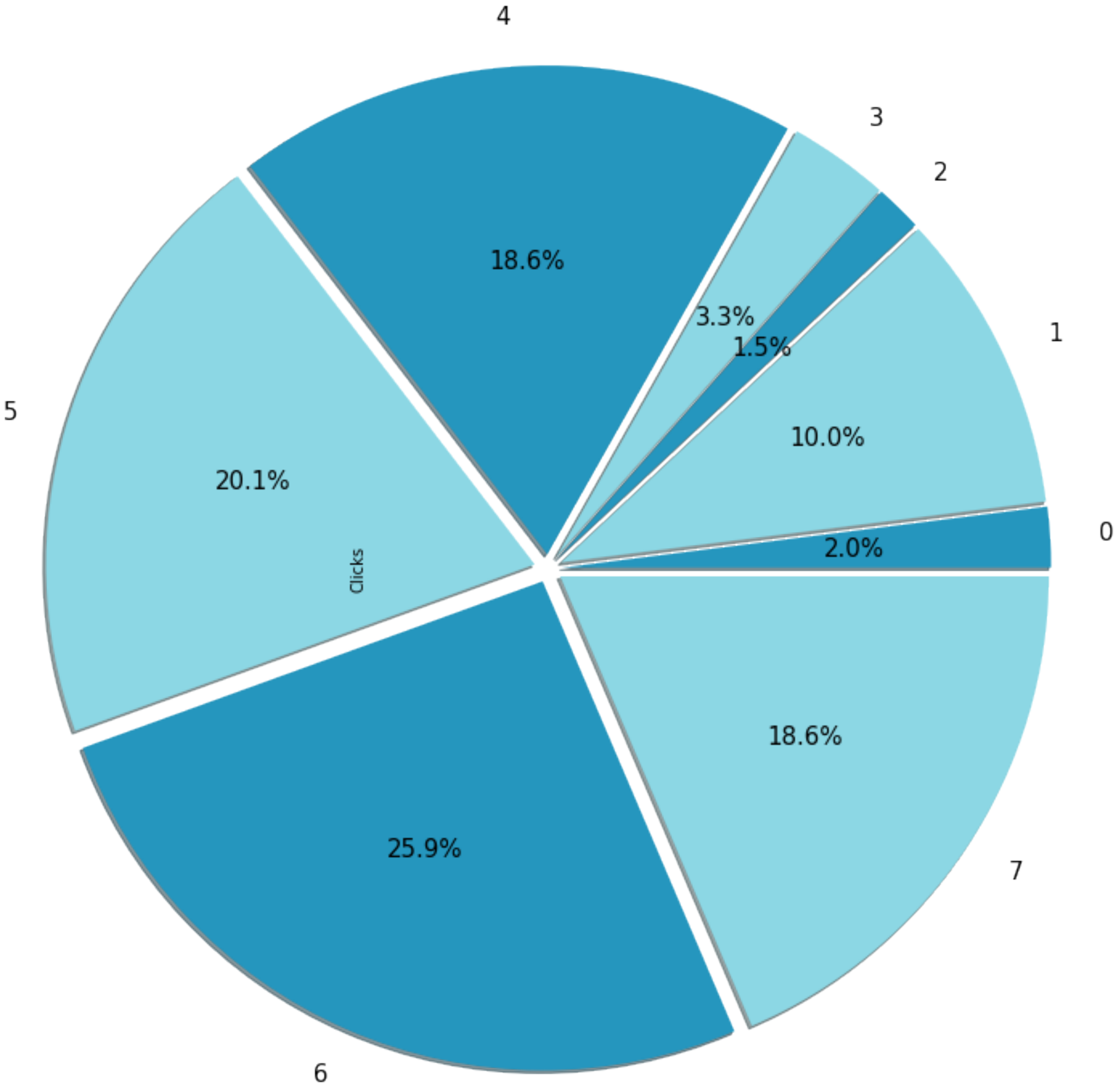
Findings:

Ad type 5 received most number of clicks from age group 7. This means, that this age group is clicking on certain type of ad (image, video, etc) and should be targeted with that particular type only

In [30]:

```
#len_Clicked_4
#calculating the number of the clicks of the when type is 5 and different age
len_Clicked_4_age0 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_age1 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_age2 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_age3 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_age4 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_age5 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_age6 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_age7 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (

print(len_Clicked_4_age0)
print(len_Clicked_4_age1)
print(len_Clicked_4_age2)
print(len_Clicked_4_age3)
print(len_Clicked_4_age4)
print(len_Clicked_4_age5)
print(len_Clicked_4_age6)
print(len_Clicked_4_age7)

#df_age_5=

clicked_age_4 = [len_Clicked_4_age0,len_Clicked_4_age1,len_Clicked_4_age2,len_
age=['-1','1','2','3','4','5','6','7']
df_clicked_age_4 = pd.DataFrame(clicked_age_4, columns = ['Clicks'])


colors = ["#2596be","#8cd7e4"]
plot = df_clicked_age_4.plot.pie(legend=False, \
                   autopct='%1.1f%%',shadow=True,radius=3.5,subplots='True',e
```

```
9
45
7
15
84
91
117
84
```



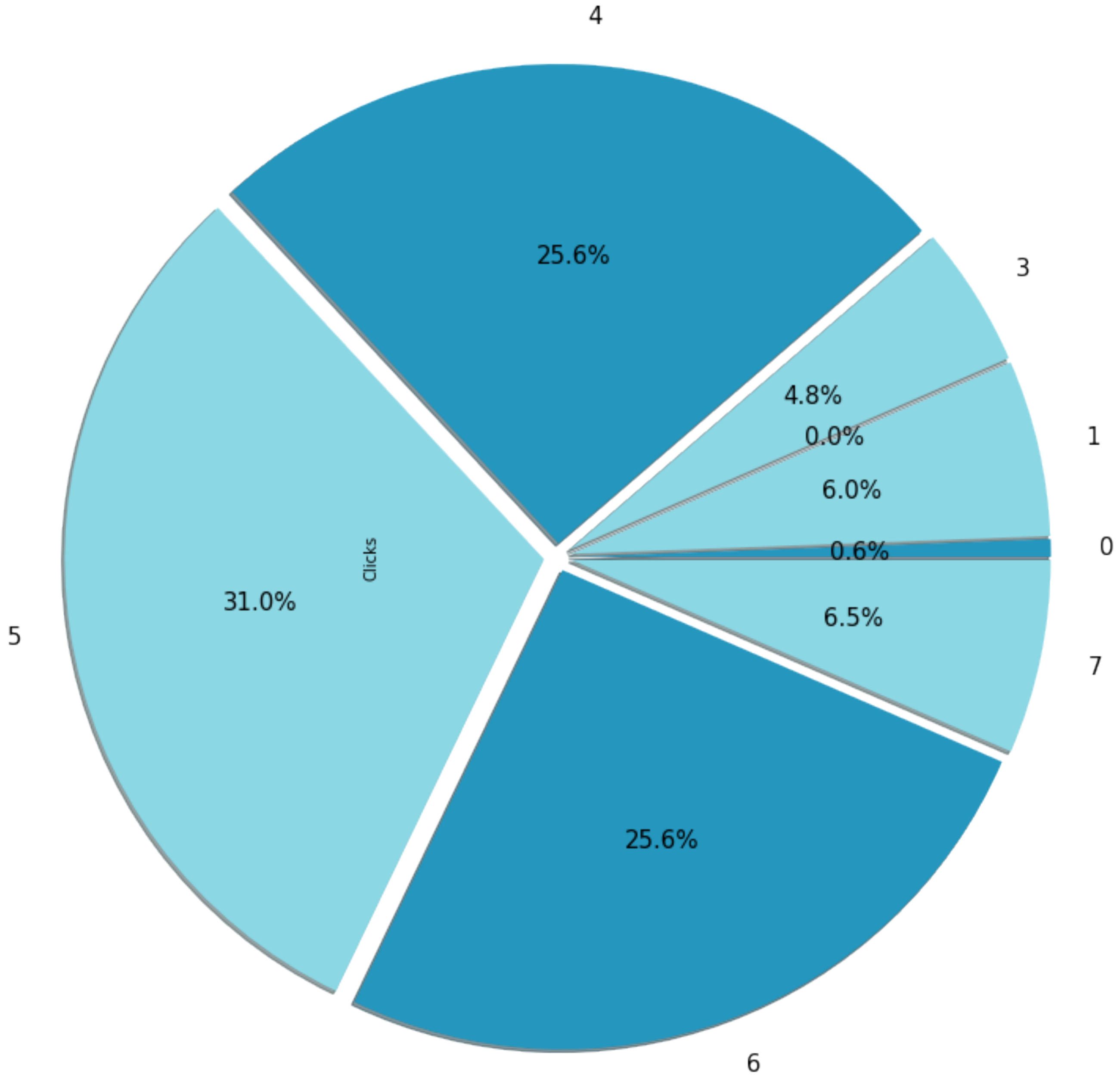
Findings:

Ad type 4 was one of the most popular type with majority of clicks coming from age group 5,6,7. That means, most of the age groups that we target ads with are clicking this particular ad type.

In [31]:

```
#len_Clicked_3
#calculating the number of the clicks of the when type is 5 and different age
len_Clicked_3_age0 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_age1 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_age2 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_age3 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_age4 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_age5 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_age6 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_age7 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (

print(len_Clicked_3_age0)
print(len_Clicked_3_age1)
print(len_Clicked_3_age2)
print(len_Clicked_3_age3)
print(len_Clicked_3_age4)
print(len_Clicked_3_age5)
print(len_Clicked_3_age6)
print(len_Clicked_3_age7)

clicked_age_3 = [len_Clicked_3_age0,len_Clicked_3_age1,len_Clicked_3_age2,len_
age=['-1','1','2','3','4','5','6','7']
df_clicked_age_3 = pd.DataFrame(clicked_age_3, columns = ['Clicks'])

colors = ["#2596be","#8cd7e4"]
plot = df_clicked_age_3.plot.pie(legend=False, \
                   autopct='%1.1f%%',shadow=True,radius=3.5,subplots='True',e
```

```
1
10
0
8
43
52
43
11
```



Findings:

Almost 31% of clicks on ad type 3 were generated from age group 5.

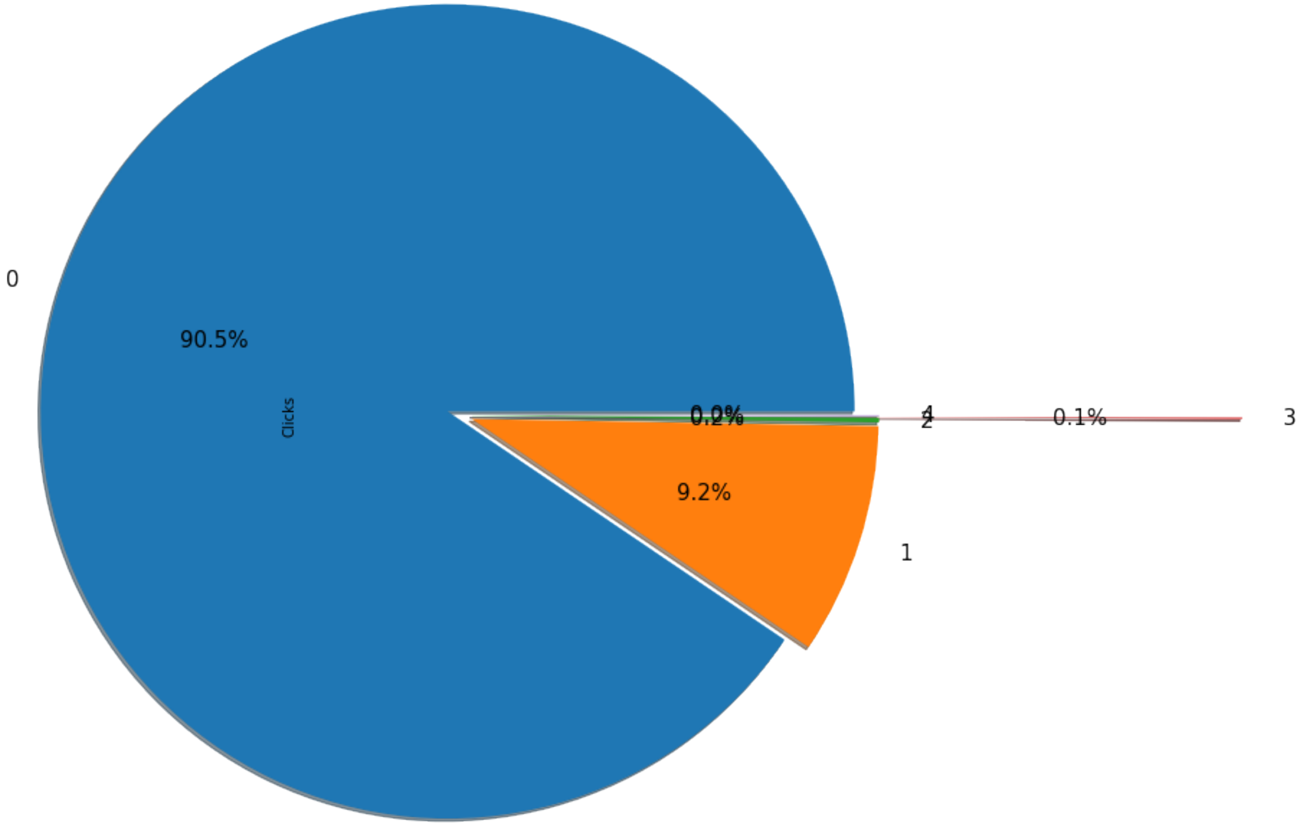
Hypotheses 4

H0 – Network status has no influence on the click rate.

H1 - Network status has influence on the click rate.

In [33]:

```
len_Clicked_5_net2 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_net3 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_net4 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_net5 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_net6 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (

print(len_Clicked_5_net2)
print(len_Clicked_5_net3)
print(len_Clicked_5_net4)
print(len_Clicked_5_net5)
print(len_Clicked_5_net6)



clicked_5_net = [len_Clicked_5_net2,len_Clicked_5_net3,len_Clicked_5_net4,len_
net=['2','3','4','5','6']
df_clicked_5_net = pd.DataFrame(clicked_5_net, columns = ['Clicks'])


#colors = ["#2596be","#8cd7e4"]
plot = df_clicked_5_net.plot.pie(legend=False, \
                   autopct='%1.1f%%',shadow=True,radius=3.25,subplots='True',
2876
292
6
2
1
```

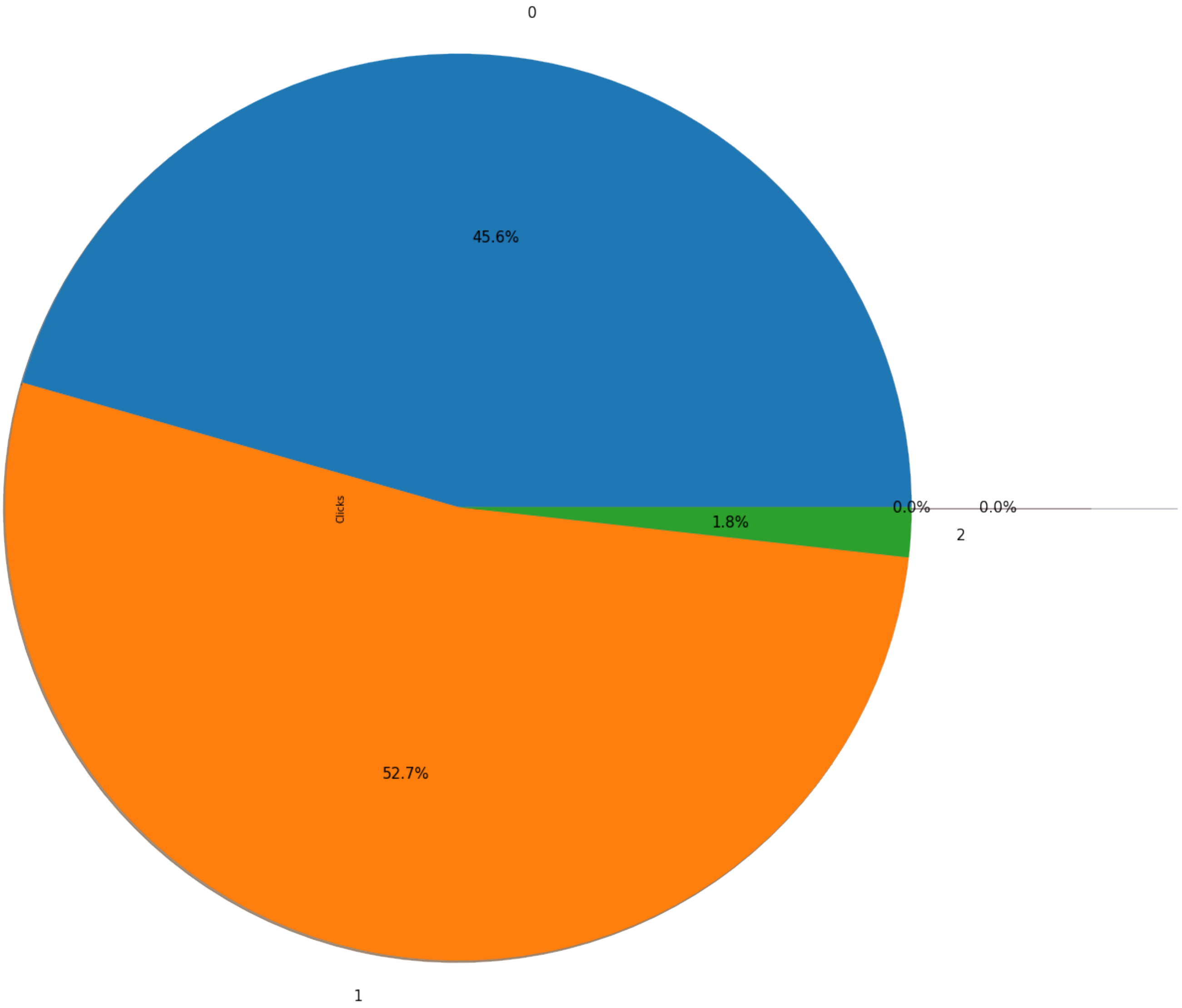


Findings:

About 90 % of clicks that were made on creative type 5 had a network status of 0.

In [34]:

```
len_Clicked_4_net2 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_net3 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_net4 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_net5 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_net6 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (

print(len_Clicked_4_net2)
print(len_Clicked_4_net3)
print(len_Clicked_4_net4)
print(len_Clicked_4_net5)
print(len_Clicked_4_net6)


clicked_4_net = [len_Clicked_4_net2,len_Clicked_4_net3,len_Clicked_4_net4,len_
net=['2','3','4','5','6']
df_clicked_4_net = pd.DataFrame(clicked_4_net, columns = ['Clicks'])


#colors = ["#2596be","#8cd7e4"]
plot = df_clicked_4_net.plot.pie(legend=False, \
                   autopct='%1.1f%%',shadow=True,radius=5.25,subplots='True',
206
238
8
0
0
```



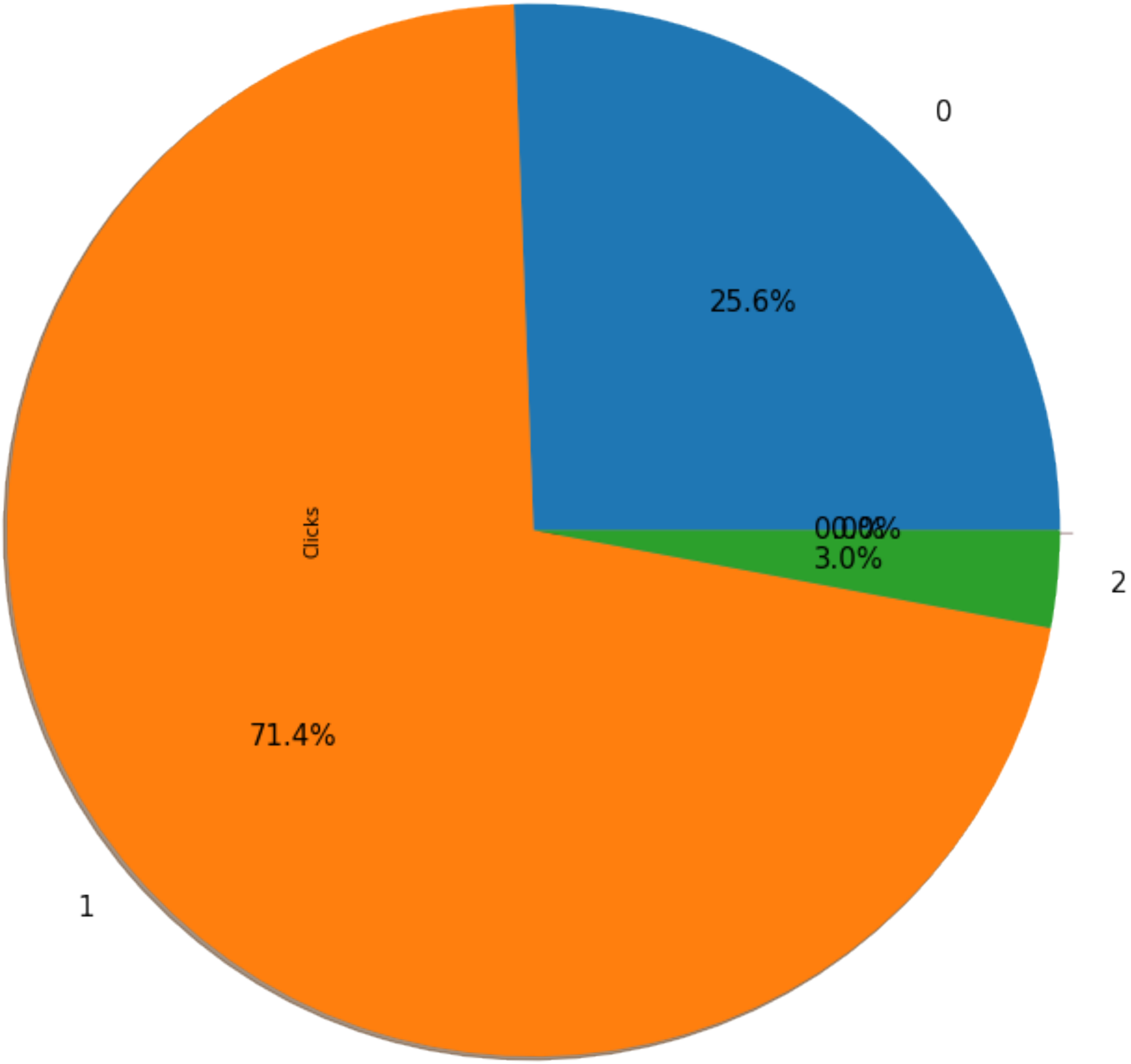
Findings:

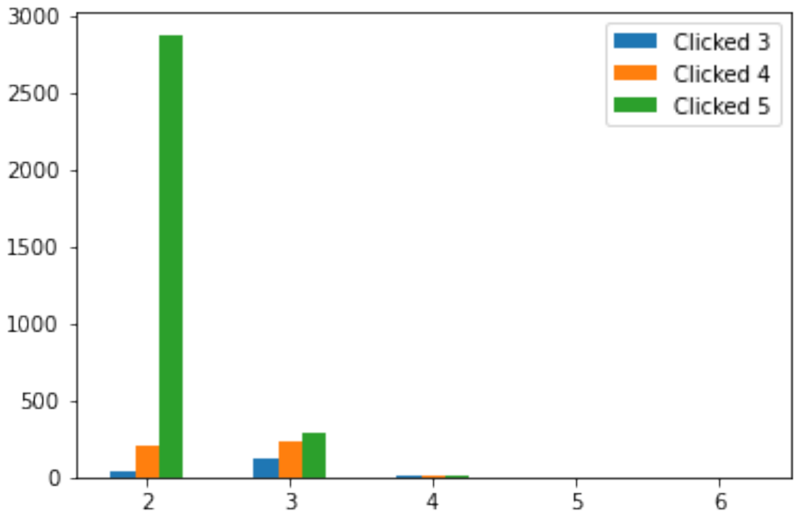
About 53 % of clicks that were made on creative type 4 had a network status of 1.

In [35]:

```
len_Clicked_3_net2 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_net3 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_net4 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_net5 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_net6 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (

print(len_Clicked_3_net2)
print(len_Clicked_3_net3)
print(len_Clicked_3_net4)
print(len_Clicked_3_net5)
print(len_Clicked_3_net6)


clicked_3_net = [len_Clicked_3_net2,len_Clicked_3_net3,len_Clicked_3_net4,len_
net=['2','3','4','5','6']
df_clicked_3_net = pd.DataFrame(clicked_3_net, columns = ['Clicks'])


#colors = ["#2596be","#8cd7e4"]
plot = df_clicked_3_net.plot.pie(legend=False, \
                   autopct='%1.1f%%',shadow=True,radius=3.25,subplots='True',





net = ['2', '3', '4','5','6']
clicked_ = [len_Clicked_5 ,len_Clicked_4,len_Clicked_3]
notclicked = [len_notClicked_5,len_notClicked_4,len_notClicked_3]
df = pd.DataFrame({'Clicked 3': clicked_3_net,'Clicked 4': clicked_4_net,'Cli
ax = df.plot.bar(rot=0)
```

```
43
120
5
0
0
```


0
25.6%
Clicks
0.0%
0.0%
3.0%
2
71.4%
1


3000
2500
2000
1500
1000
500
0
2
3
4
5
6
Clicked 3
Clicked 4
Clicked 5

Findings:

About 71.5 % of clicks that were made on creative type 3 had a network status of 1.

Hypotheses 5

H0 - Gender of the user, ad type and clicks are correlated.

H1 - Gender of the user, ad type and clicks are not correlated.

When ad type is 3

In [50]:

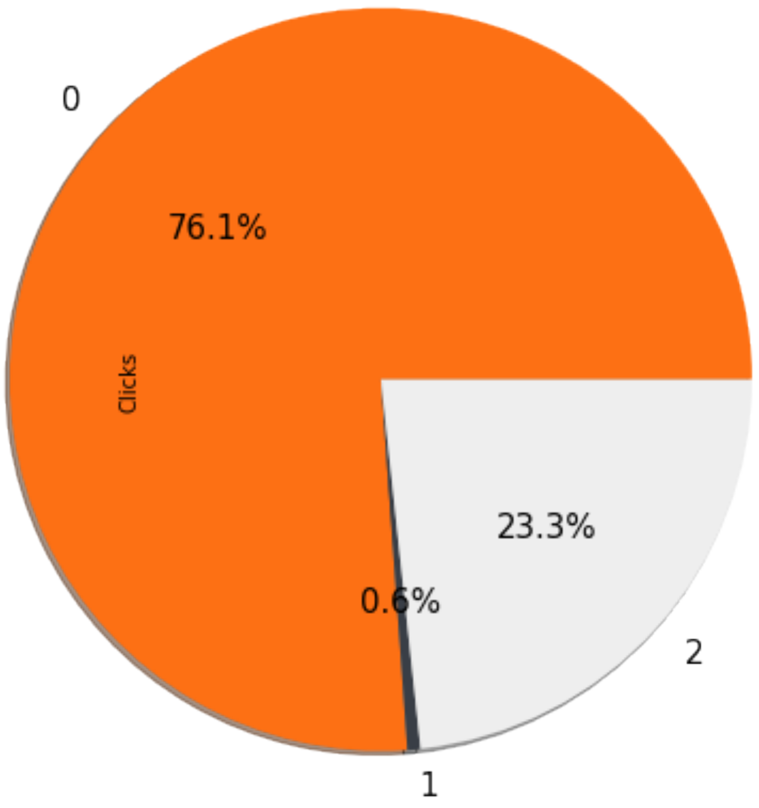
```
len_Clicked_3_gen2 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_gen3 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (
len_Clicked_3_gen4 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 3) & (

print("Number of clicks when ad type is 3 and Gender is 0 -->",len_Clicked_3_
print("Number of clicks when ad type is 3 and Gender is 1 -->",len_Clicked_3_
print("Number of clicks when ad type is 5 and Gender is 2 -->",len_Clicked_3_

clicked_ad_3_gen = [len_Clicked_3_gen2,len_Clicked_3_gen3,len_Clicked_3_gen4]
gen=['2','3','4']
df_clicked_ad_3_gen = pd.DataFrame(clicked_ad_3_gen, columns = ['Clicks'])
```

```
Number of clicks when ad type is 3 and Gender is 0 --> 134
Number of clicks when ad type is 3 and Gender is 1 --> 1
Number of clicks when ad type is 5 and Gender is 2 --> 41
```

Pie Chart for clicks on Ad type 3

In [62]:

```
colors = ["#FD7014","#393E46","#EEEEEE"]


plot = df_clicked_ad_3_gen.plot.pie(legend=False, \
                  autopct='%1.1f%%',shadow=True,radius=2,subplots='True',expl
```



Findings:

Around 76.1 % of gender code 0 users clicked on ad type 3

When Ad type is 4

In [56]:

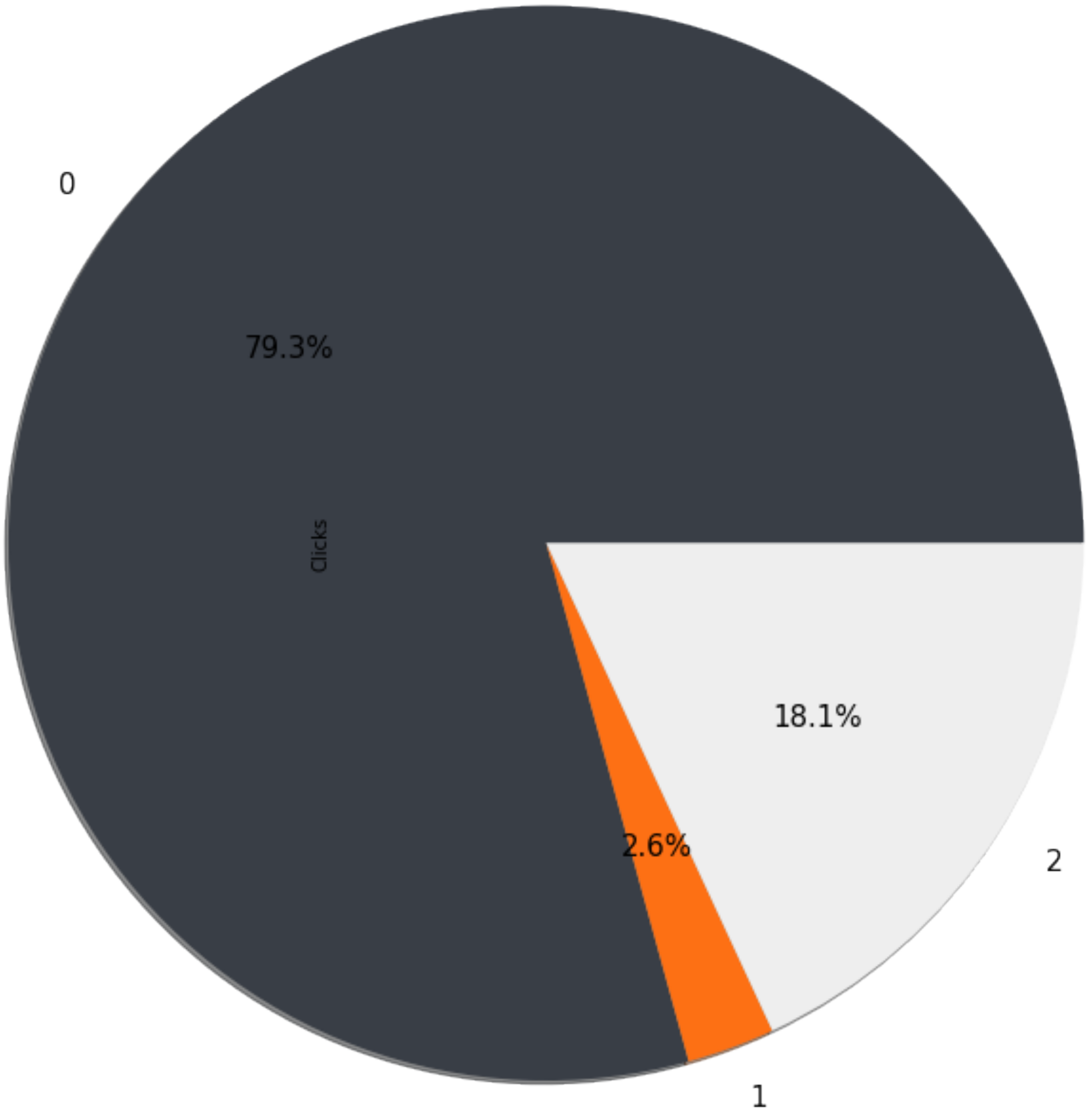
```
# Calculating the lenght of the clicks made by the different gender when the

len_Clicked_4_gen2 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_gen3 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (
len_Clicked_4_gen4 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 4) & (

print("Number of clicks when ad type is 4 and Gender is 2 -->",len_Clicked_4_
print("Number of clicks when ad type is 4 and Gender is 3 -->",len_Clicked_4_
print("Number of clicks when ad type is 4 and Gender is 4 -->",len_Clicked_4_

#creating the data frame of the lenghts.
clicked_ad_4_gen = [len_Clicked_4_gen2,len_Clicked_4_gen3,len_Clicked_4_gen4]
gen=['2','3','4']
df_clicked_ad_4_gen = pd.DataFrame(clicked_ad_4_gen, columns = ['Clicks'])
```

```
Number of clicks when ad type is 4 and Gender is 2 --> 390
Number of clicks when ad type is 4 and Gender is 3 --> 13
Number of clicks when ad type is 4 and Gender is 4 --> 89
```

Pie Chart for clicks on Ad type 4

In [57]:

```
#Plotting the pie chart graph.
colors = ["#393E46","#FD7014","#EEEEEE"]
plot = df_clicked_ad_4_gen.plot.pie(legend=False, \
                   autopct='%1.1f%%',shadow=True,radius=3.25,subplots='True',
```



Findings:

Ad type 4 was clicked on by 79% of gender code 0 then by gender 2 users.

In [59]:

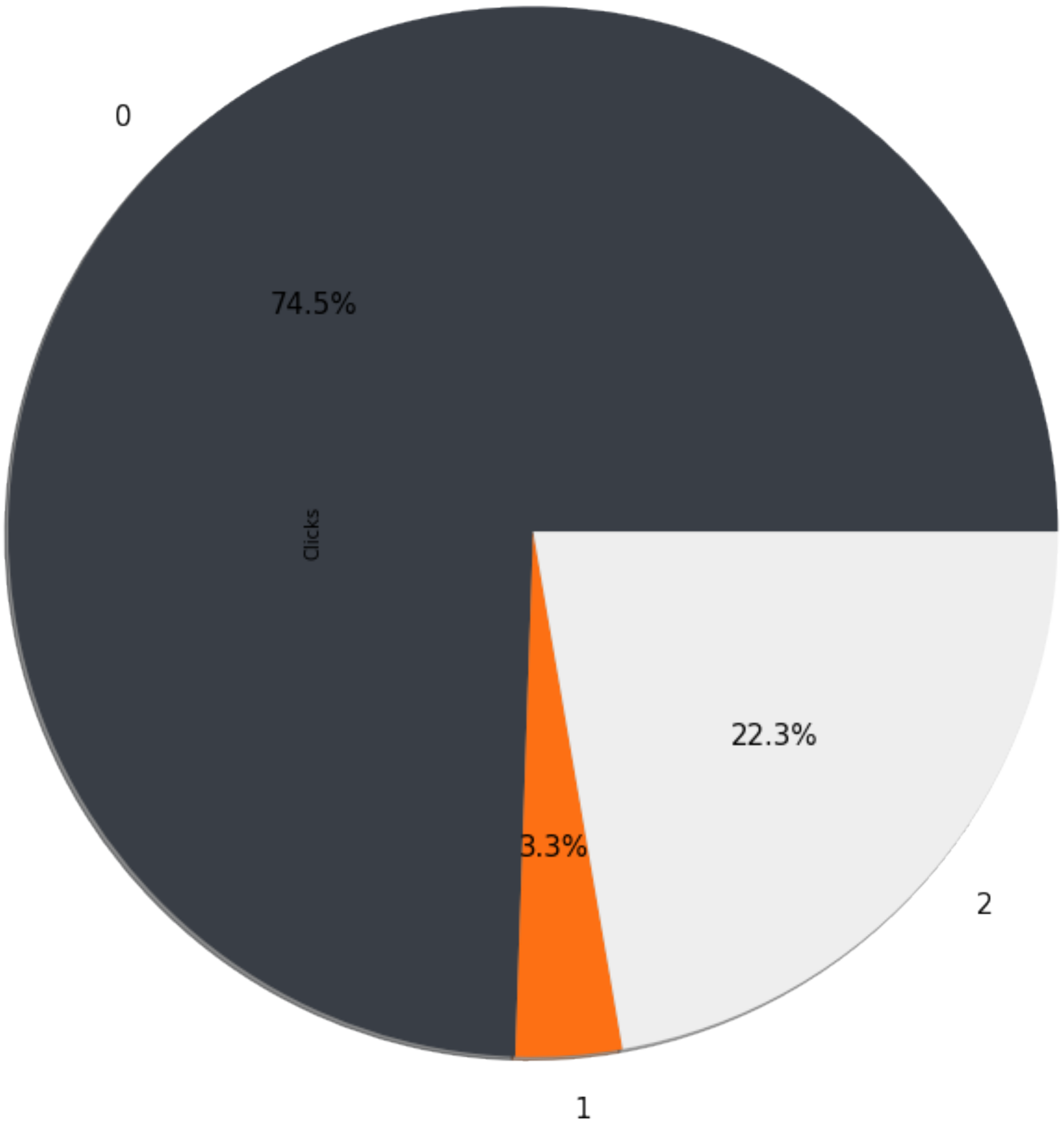
```
# Calculating the lenght of the clicks made by the different gender when the

len_Clicked_5_gen2 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_gen3 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (
len_Clicked_5_gen4 = len(df.loc[(df.label == 1) & (df.inter_type_cd == 5) & (

print("Number of clicks when ad type is 5 and Gender is 2 -->",len_Clicked_5_
print("Number of clicks when ad type is 5 and Gender is 3 -->",len_Clicked_5_
print("Number of clicks when ad type is 5 and Gender is 4 -->",len_Clicked_5_
print()
print("Total Number of clicks when ad type is 5 -->",len_Clicked_5_gen4+len_C



#creating the data frame of the lenghts.

clicked_ad_5_gen = [len_Clicked_5_gen2,len_Clicked_5_gen3,len_Clicked_5_gen4]
gen=['2','3','4']
df_clicked_ad_5_gen = pd.DataFrame(clicked_ad_5_gen, columns = ['Clicks'])
```

```
Number of clicks when ad type is 5 and Gender is 2 --> 2390
Number of clicks when ad type is 5 and Gender is 3 --> 105
Number of clicks when ad type is 5 and Gender is 4 --> 715

Total Number of clicks when ad type is 5 --> 3210
```

Pie Chart for clicks on Ad type 5

In [60]:

```
#Plotting the pie chart graph.
colors = ["#393E46","#FD7014","#EEEEEE"]
plot = df_clicked_ad_5_gen.plot.pie(legend=False, \
                   autopct='%1.1f%%',shadow=True,radius=3.25,subplots='True',
```



Findings:

Ad type 5 was popular amongst users with gender code 0

Hypotheses 6

H0 – Slot of the ad (placement) has no influence on the click rate.

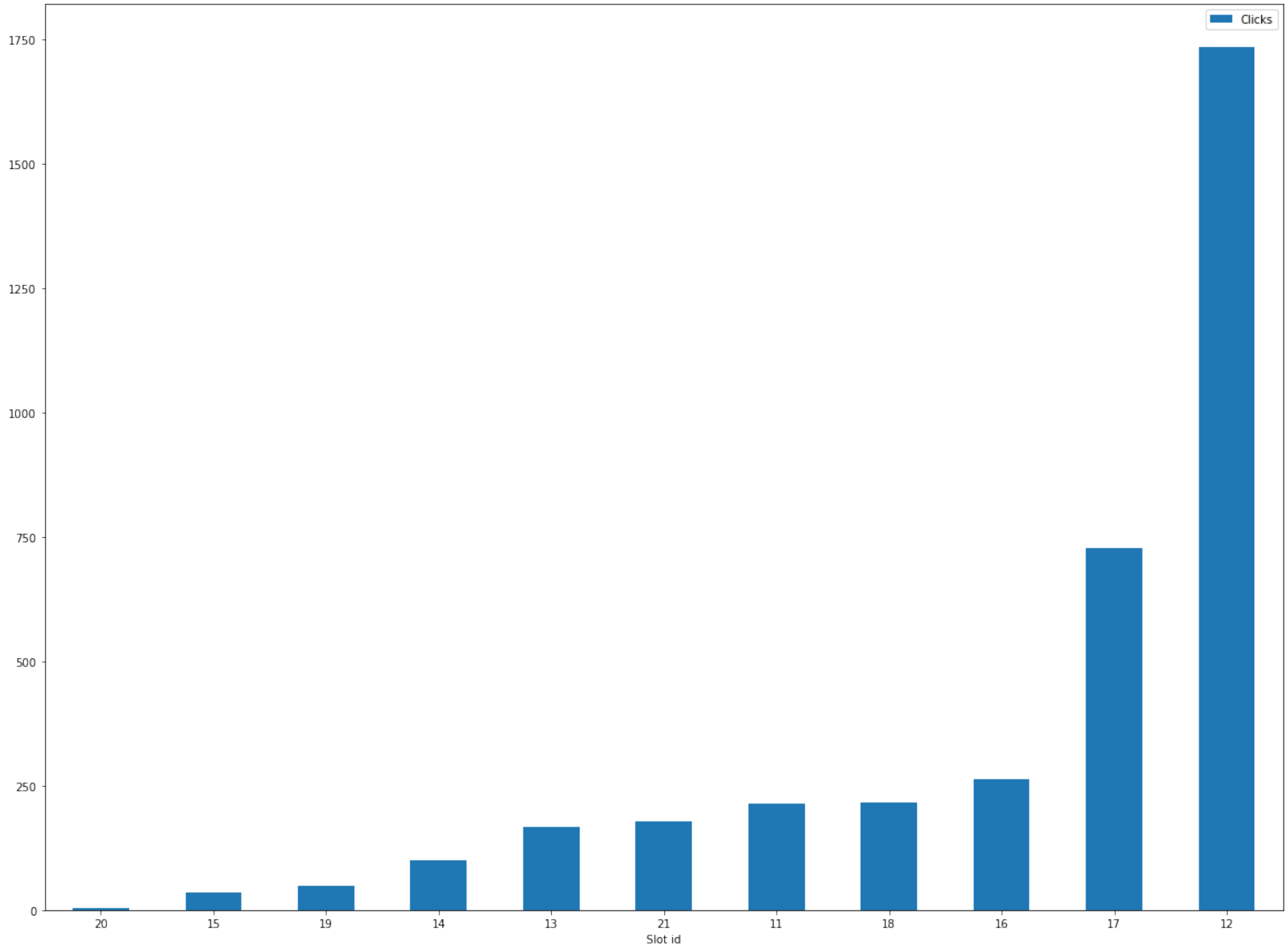
H1 - Slot of the ad (placement) has influence on the click rate.

In [27]:

In [27]:

```
# Calculating the number of the clicks corresponding to each slot id

len_Clicked_slot_11 = len(df.loc[(df.label == 1) & (df.slot_id == 11)])
len_Clicked_slot_12 = len(df.loc[(df.label == 1) & (df.slot_id == 12)])
len_Clicked_slot_13 = len(df.loc[(df.label == 1) & (df.slot_id == 13)])
len_Clicked_slot_14 = len(df.loc[(df.label == 1) & (df.slot_id == 14)])
len_Clicked_slot_15 = len(df.loc[(df.label == 1) & (df.slot_id == 15)])
len_Clicked_slot_16 = len(df.loc[(df.label == 1) & (df.slot_id == 16)])
len_Clicked_slot_17 = len(df.loc[(df.label == 1) & (df.slot_id == 17)])
len_Clicked_slot_18 = len(df.loc[(df.label == 1) & (df.slot_id == 18)])
len_Clicked_slot_19 = len(df.loc[(df.label == 1) & (df.slot_id == 19)])
len_Clicked_slot_20 = len(df.loc[(df.label == 1) & (df.slot_id == 20)])
len_Clicked_slot_21 = len(df.loc[(df.label == 1) & (df.slot_id == 21)])
len_Clicked_slot_22 = len(df.loc[(df.label == 1) & (df.slot_id == 22)])




# creating the dictionary of the lenght of the slot id.

slot_dict={'11':len_Clicked_slot_11,'12':len_Clicked_slot_12,'13':len_Clicked
           '15':len_Clicked_slot_15,'16':len_Clicked_slot_16,'17':len_Clicked
           '19':len_Clicked_slot_19,'20':len_Clicked_slot_20,'21':len_Clicked



# creating a list of the sorted values of the dict. VALUES not KEY

sorted_values = sorted(slot_dict.values()) # Sort the values
sorted_dict = {}

for i in sorted_values:
    for k in slot_dict.keys():
        if slot_dict[k] == i:
            sorted_dict[k] = slot_dict[k]
            break

#print(sorted_dict)
values_slot_dict = sorted_dict.values()
values_list = list(values_slot_dict)
#print(values_list)





# Creating a list of the sorted KEYS of the dict. KEYS not VALUES

def getList(dict):
    return dict.keys()
# Driver program
dict = sorted_dict
key_list_sorted_dict=getList(dict)
key_list = list(key_list_sorted_dict)
```

```
#print(key_list)



#print(sorted_dict)



# Creating the bar plot
df_not_clicks = pd.DataFrame({'Clicks':values_list, 'Slot id':key_list})
ax2 = df_not_clicks.plot.bar(x='Slot id', y='Clicks', rot=0, figsize=(20, 15)


# Creating the pareto chart of the bar graphs made.
#ax1=pareto_plot(ax2, x='Slot id', y='Clicks', title='Ice Cream Sales')
```



Findings:

Slot id 12 generated the highest number of clicks amongst the users on all ad types.

Conclusion

The summary of findings:

Hypotheses 1

Ad type 5 has got the highest number of clicks and has seen the highest number of non-clicks.

Hypotheses 2

Cities with rank 3 has the highest number of clicks followed by cities with rank 5 and 4 respectively

Hypotheses 3

Ad type 5 received most number of clicks from age group 7. This means, that this age group is clicking on certain type of ad (image, video, etc) and should be targeted with that particular type only

Ad type 4 was one of the most popular type with majority of clicks coming from age group 5,6,7. That means, most of the age groups that we target ads with are clicking this particular ad type.

Almost 31% of clicks on ad type 3 were generated from age group 5.

Hypotheses 4

About 90 % of clicks that were made on creative type 5 had a network status of 0.

About 53 % of clicks that were made on creative type 4 had a network status of 1.

About 71.5 % of clicks that were made on creative type 3 had a network status of 1.

Hypotheses 5

Around 76.1 % of gender code 0 users clicked on ad type 3

Ad type 4 was clicked on by 79% of gender code 0 and then by gender 2 users.

Ad type 5 was popular amongst users with gender code 0

Hypotheses 6

Slot id 12 generated the highest number of clicks amongst the users on all ad types.

Business implications for audiences

Internet marketing and advertising is evolving by the day. New ad technology, availability of data, and increasing use of internet has made the digital advertising market very competitive. In order to effectively allocate funds to different digital platforms for advertising, ascertaining the target audience, this study was important. Digital Marketing Managers can do a thorough study like we have done above and evaluate the variables that are important to them before planning a new campaign. This will not only help them understand their audience better but also optimize their ad performance and increase profitability.

Limitations of this project

We are retreiving a subset of the dataset, that is 1/10th of the actual length of the datset. So, the results might be affected with a slight percentage with every run.

References

1. https://towardsdatascience.com/all-about-heatmaps-bb7d97f099d7
2. https://pandas.pydata.org
3. https://matplotlib.org/stable/index.html
4. https://www.geeksforgeeks.org
5. https://www.kaggle.com/tomerzemelman/ctr-data-exploration