Name : Arul Kumar ARK ¶

Reg No. : 225229103

Lab 14 : Click through rate Predictive Mode

In [1]:

```
import numpy as np
import random
n = 40428967
sample_size = 1000000
skip_values = sorted(random.sample(range(1,n), n-sample_size))

types_train = {
    'id': np.dtype(int),
    'click': np.dtype(int),
    'hour': np.dtype(int),
    'C1': np.dtype(int),
    'banner_pos': np.dtype(int),
    'site_id': np.dtype(str),
    'site_domain': np.dtype(str),
    'site_category': np.dtype(str),
    'app_id': np.dtype(str),
    'app_domain': np.dtype(str),
    'app_category': np.dtype(str),
    'device_id': np.dtype(str),
    'device_ip': np.dtype(str),
    'device_model': np.dtype(str),
    'device_type': np.dtype(int),
    'device_conn_type': np.dtype(int),
    'C14': np.dtype(int),
    'C15': np.dtype(int),
    'C16': np.dtype(int),
    'C17': np.dtype(int),
    'C18': np.dtype(int),
    'C19': np.dtype(int),
    'C20': np.dtype(int),
    'C21':np.dtype(int)
}

types_test = {
    'id': np.dtype(int),
    'hour': np.dtype(int),
    'C1': np.dtype(int),
    'banner_pos': np.dtype(int),
    'site_id': np.dtype(str),
    'site_domain': np.dtype(str),
    'site_category': np.dtype(str),
    'app_id': np.dtype(str),
    'app_domain': np.dtype(str),
    'app_category': np.dtype(str),
    'device_id': np.dtype(str),
    'device_ip': np.dtype(str),
    'device_model': np.dtype(str),
    'device_type': np.dtype(int),
    'device_conn_type': np.dtype(int),
    'C14': np.dtype(int),
    'C15': np.dtype(int),
    'C16': np.dtype(int),
    'C17': np.dtype(int),
    'C18': np.dtype(int),
    'C19': np.dtype(int),
    'C20': np.dtype(int),
    'C21':np.dtype(int)
```

}

In [2]:

```
import pandas as pd
import gzip

parse_date = lambda val : pd.datetime.strptime(val, '%y%m%d%H')

with gzip.open('train.gz') as f:
    train = pd.read_csv(f, parse_dates = ['hour'], date_parser = parse_date, dtype=types

train.head()
```

Out[2]:

	id	click	hour	C1	banner_pos	site_id	site_domain	site_category	ap
0	601394868	0	2014-10-21	1005	0	030440fe	08ba7db9	76b2941d	ecad2
1	-59070594	0	2014-10-21	1005	1	0eb72673	d2f72222	f028772b	ecad2
2	-1859646727	0	2014-10-21	1005	0	6c5b482c	7687a86e	3e814130	ecad2
3	497487217	0	2014-10-21	1005	0	85f751fd	c4e18dd6	50e219e0	febd
4	-1852466777	0	2014-10-21	1005	0	1fbe01fe	f3845767	28905ebd	ecad2

5 rows × 24 columns

In [3]:

```
train.shape
```

Out[3]:

```
(1000000, 24)
```

In [4]:

```
train.dtypes
```

Out[4]:

```
id                            int32
click                         int32
hour                datetime64[ns]
C1                            int32
banner_pos                    int32
site_id                      object
site_domain                  object
site_category                object
app_id                       object
app_domain                   object
app_category                 object
device_id                    object
device_ip                    object
device_model                 object
device_type                   int32
device_conn_type              int32
C14                           int32
C15                           int32
C16                           int32
C17                           int32
C18                           int32
C19                           int32
C20                           int32
C21                           int32
dtype: object
```

In [5]:

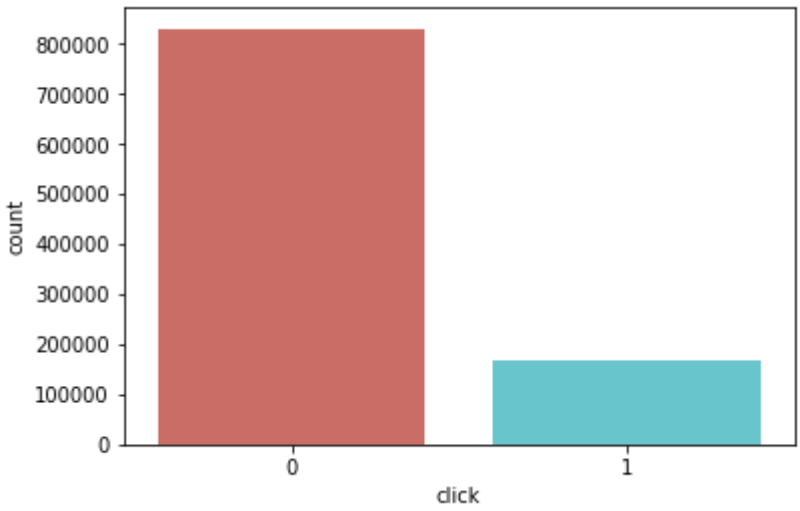
```
train['click'].value_counts()
```

Out[5]:

```
0    830423
1    169577
Name: click, dtype: int64
```

In [7]:

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.countplot(x='click',data=train, palette='hls')
plt.show();
```



In [8]:

```
train['click'].value_counts()/len(train)
```

Out[8]:

```
0    0.830423
1    0.169577
Name: click, dtype: float64
```

Click through rate is approx. 17%, and approx. 83% is not clicked.

In [9]:

```
train.hour.describe()
```

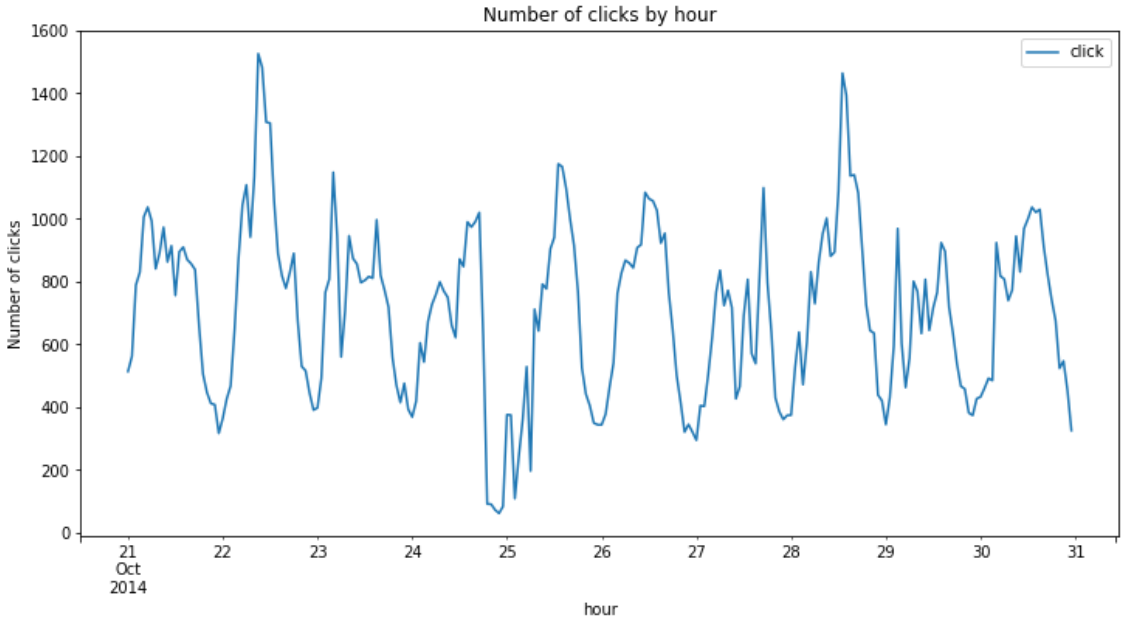
Out[9]:

```
count                 1000000
unique                    240
top       2014-10-22 09:00:00
freq                    11144
first     2014-10-21 00:00:00
last      2014-10-30 23:00:00
Name: hour, dtype: object
```

The data covers 10 days of click streams data from 2014-10-21 to 2014-10-30, that is 240 hours.

In [10]:

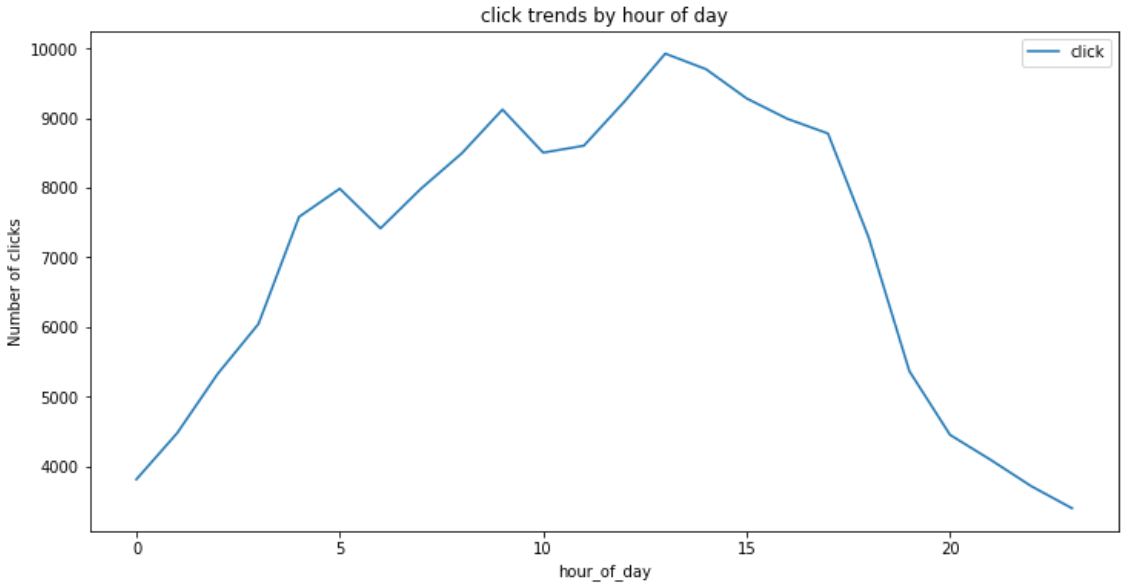
```
train.groupby('hour').agg({'click':'sum'}).plot(figsize=(12,6))
plt.ylabel('Number of clicks')
plt.title('Number of clicks by hour');
```



Feature engineering for date time features

In [11]:

```
train['hour_of_day'] = train.hour.apply(lambda x: x.hour)
train.groupby('hour_of_day').agg({'click':'sum'}).plot(figsize=(12,6))
plt.ylabel('Number of clicks')
plt.title('click trends by hour of day');
```



In general, the highest number of clicks is at hour 13 and 14 (1pm and 2pm), and the least number of clicks is at hour 0 (mid-night). It seems a useful feature for roughly estimation.

In [12]:

```
train.head(3)
```

Out[12]:

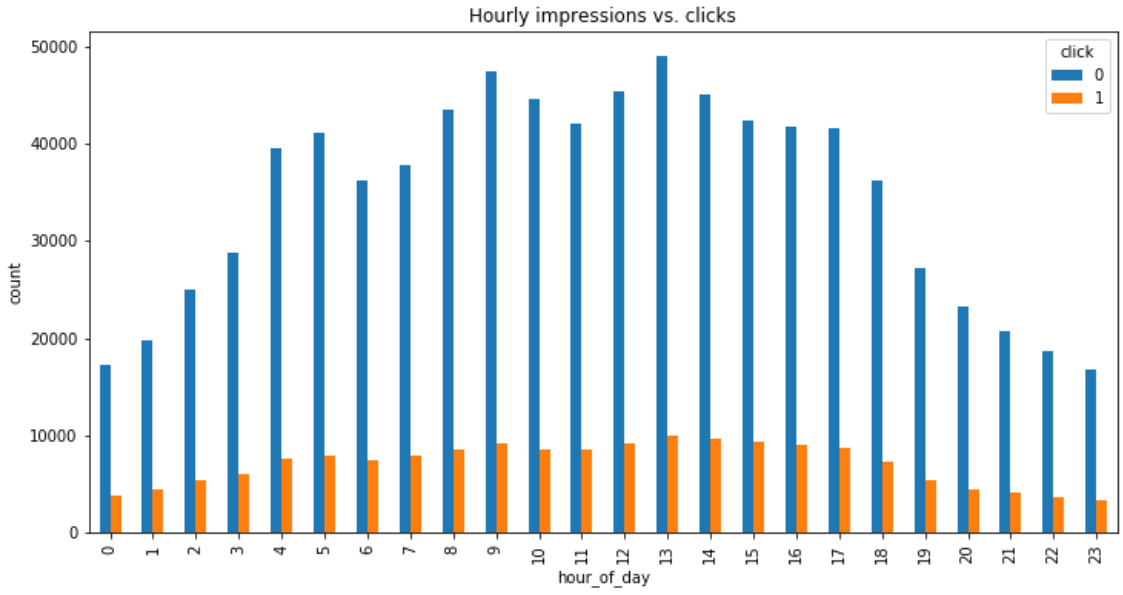
	id	click	hour	C1	banner_pos	site_id	site_domain	site_category	ap
0	601394868	0	2014-10-21	1005	0	030440fe	08ba7db9	76b2941d	ecad2
1	-59070594	0	2014-10-21	1005	1	0eb72673	d2f72222	f028772b	ecad2
2	-1859646727	0	2014-10-21	1005	0	6c5b482c	7687a86e	3e814130	ecad2

3 rows × 25 columns

Let's take impressions into consideration.

In [13]:

```
train.groupby(['hour_of_day', 'click']).size().unstack().plot(kind='bar', title="Hour of
plt.ylabel('count')
plt.title('Hourly impressions vs. clicks');
```



In [14]:

```
train.head(3)
```

Out[14]:

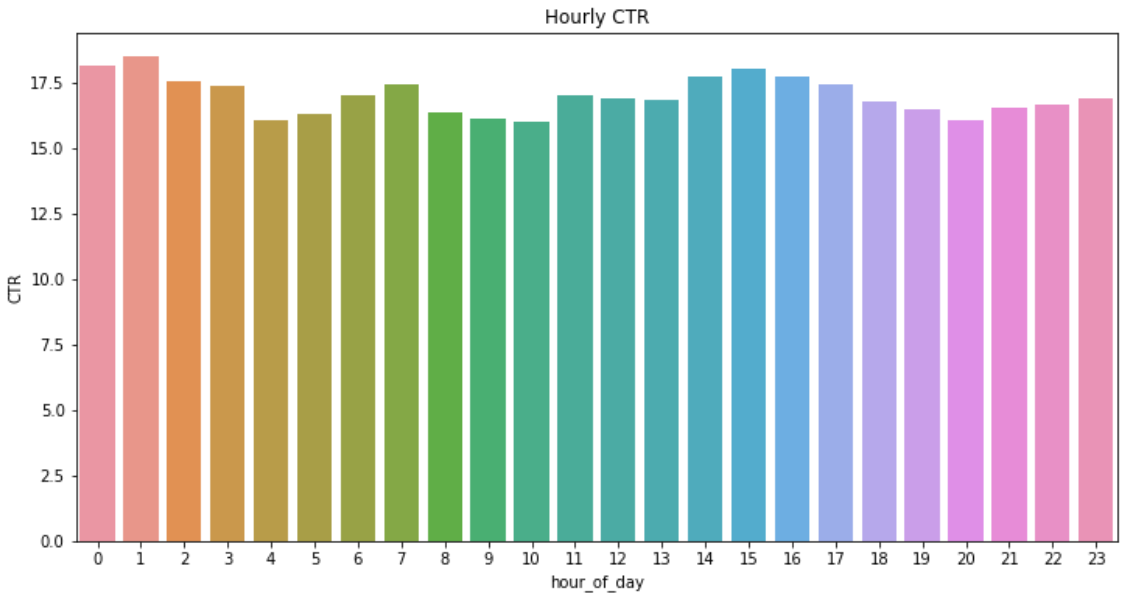
	id	click	hour	C1	banner_pos	site_id	site_domain	site_category	ap
0	601394868	0	2014-10-21	1005	0	030440fe	08ba7db9	76b2941d	ecad2
1	-59070594	0	2014-10-21	1005	1	0eb72673	d2f72222	f028772b	ecad2
2	-1859646727	0	2014-10-21	1005	0	6c5b482c	7687a86e	3e814130	ecad2

3 rows × 25 columns

Hourly CTR

In [15]:

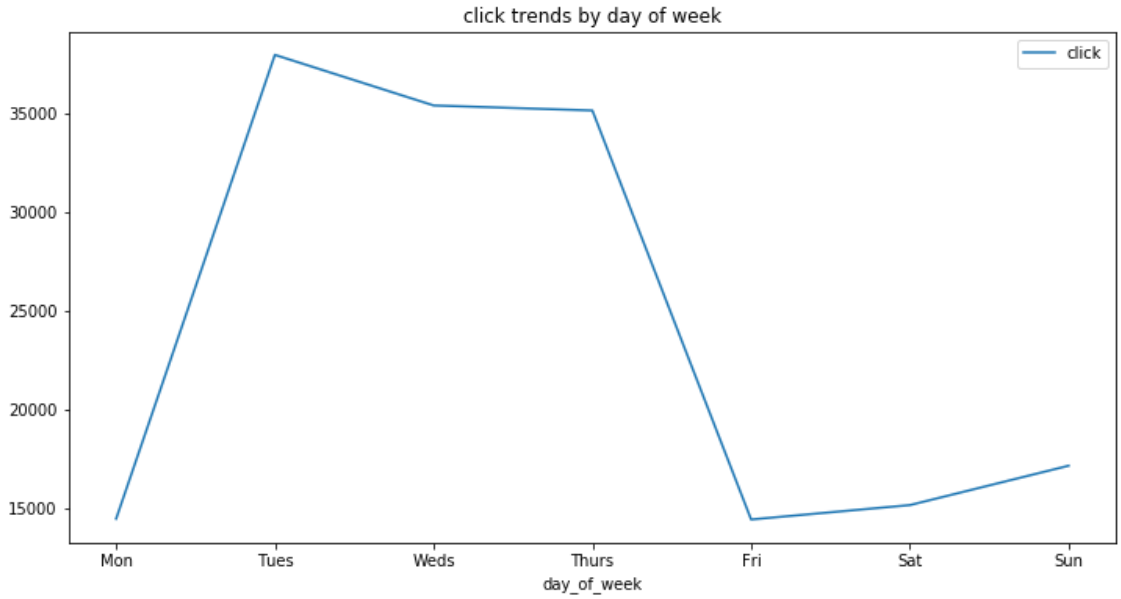
```
import seaborn as sns

df_click = train[train['click'] == 1]
df_hour = train[['hour_of_day','click']].groupby(['hour_of_day']).count().reset_index()
df_hour = df_hour.rename(columns={'click': 'impressions'})
df_hour['clicks'] = df_click[['hour_of_day','click']].groupby(['hour_of_day']).count().r
df_hour['CTR'] = df_hour['clicks']/df_hour['impressions']*100

plt.figure(figsize=(12,6))
sns.barplot(y='CTR', x='hour_of_day', data=df_hour)
plt.title('Hourly CTR');
```



Day of week

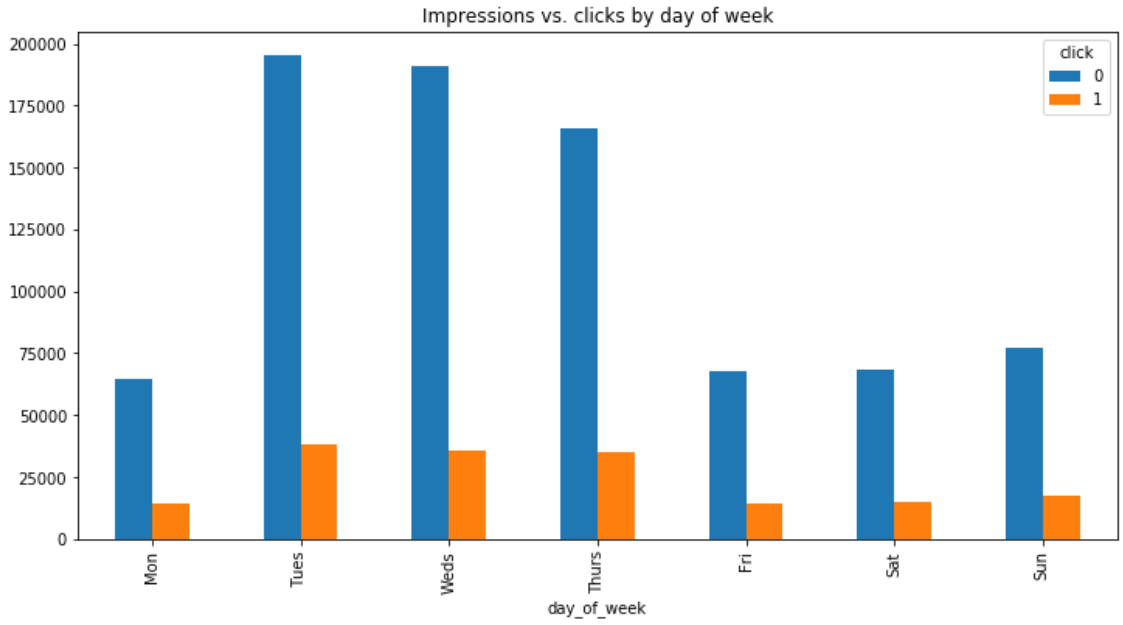
In [16]:

```
train['day_of_week'] = train['hour'].apply(lambda val: val.weekday_name)
cats = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
train.groupby('day_of_week').agg({'click':'sum'}).reindex(cats).plot(figsize=(12,6))
ticks = list(range(0, 7, 1)) # points on the x axis where you want the label to appear
labels = "Mon Tues Weds Thurs Fri Sat Sun".split()
plt.xticks(ticks, labels)
plt.title('click trends by day of week');
```



In [17]:

```
train.groupby(['day_of_week','click']).size().unstack().reindex(cats).plot(kind='bar', t
ticks = list(range(0, 7, 1)) # points on the x axis where you want the label to appear
labels = "Mon Tues Weds Thurs Fri Sat Sun".split()
plt.xticks(ticks, labels)
plt.title('Impressions vs. clicks by day of week');
```



Day of week CTR

In [18]:

```
train.head(3)
```

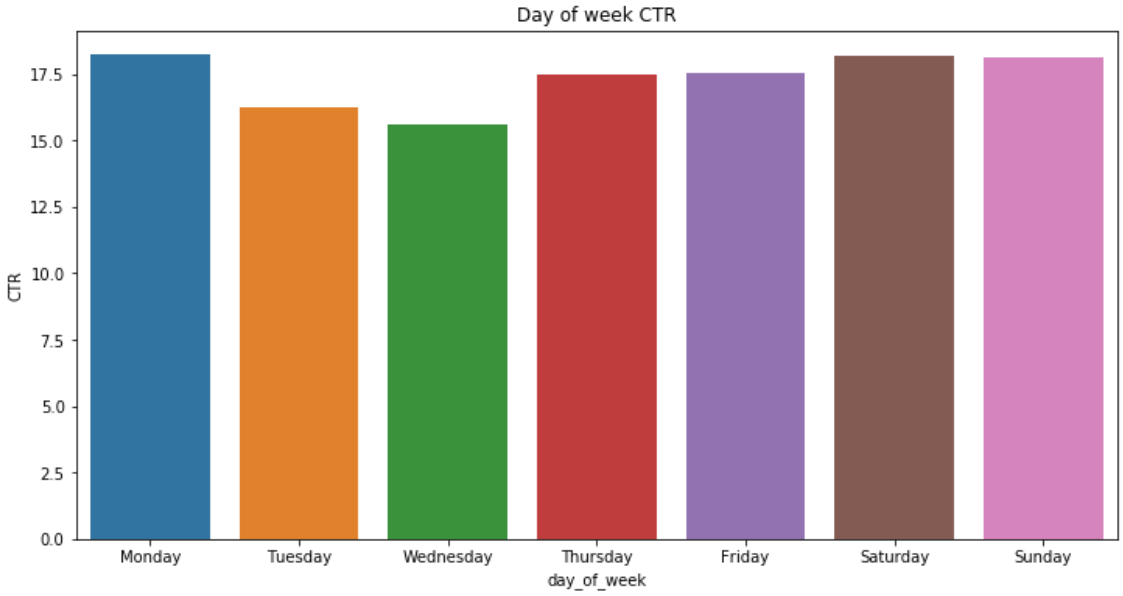
Out[18]:

	id	click	hour	C1	banner_pos	site_id	site_domain	site_category	ap
0	601394868	0	2014-10-21	1005	0	030440fe	08ba7db9	76b2941d	ecad2
1	-59070594	0	2014-10-21	1005	1	0eb72673	d2f72222	f028772b	ecad2
2	-1859646727	0	2014-10-21	1005	0	6c5b482c	7687a86e	3e814130	ecad2

3 rows × 26 columns

In [19]:

```
df_click = train[train['click'] == 1]
df_dayofweek = train[['day_of_week','click']].groupby(['day_of_week']).count().reset_inc
df_dayofweek = df_dayofweek.rename(columns={'click': 'impressions'})
df_dayofweek['clicks'] = df_click[['day_of_week','click']].groupby(['day_of_week']).cour
df_dayofweek['CTR'] = df_dayofweek['clicks']/df_dayofweek['impressions']*100

plt.figure(figsize=(12,6))
sns.barplot(y='CTR', x='day_of_week', data=df_dayofweek, order=['Monday', 'Tuesday', 'We
plt.title('Day of week CTR');
```



In [20]:

```
train.head(3)
```

Out[20]:

	id	click	hour	C1	banner_pos	site_id	site_domain	site_category	ap
0	601394868	0	2014-10-21	1005	0	030440fe	08ba7db9	76b2941d	ecad2
1	-59070594	0	2014-10-21	1005	1	0eb72673	d2f72222	f028772b	ecad2
2	-1859646727	0	2014-10-21	1005	0	6c5b482c	7687a86e	3e814130	ecad2

3 rows × 26 columns

C1 feature

In [21]:

```
print(train.C1.value_counts()/len(train))
```

```
1005    0.918462
1002    0.055026
1010    0.022414
1012    0.002856
1007    0.000873
1001    0.000245
1008    0.000124
Name: C1, dtype: float64
```

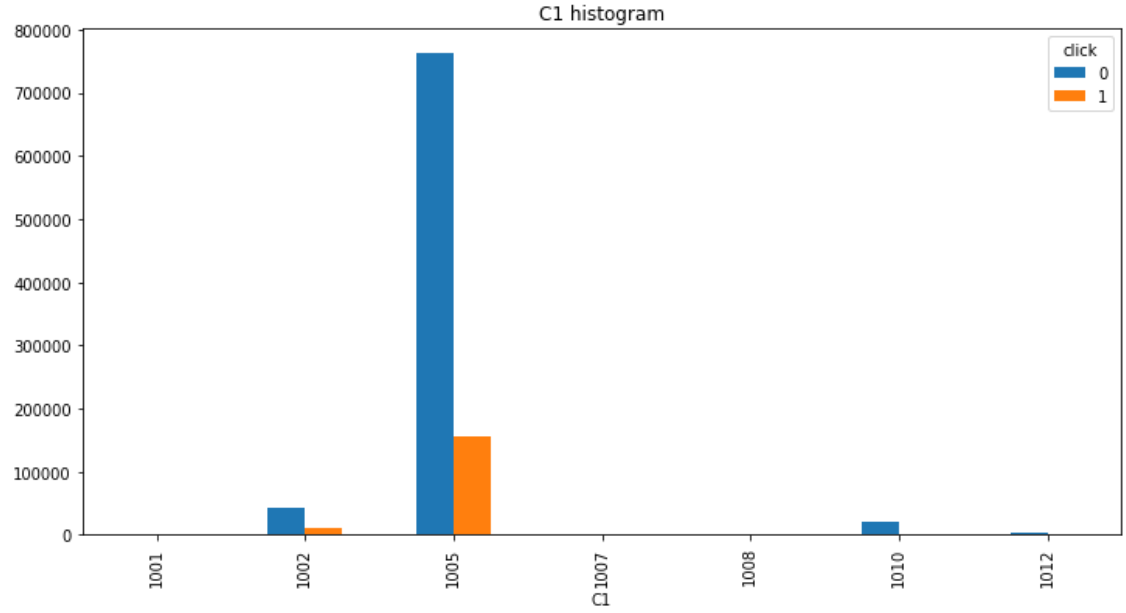
In [22]:

```
C1_values = train.C1.unique()
C1_values.sort()
ctr_avg_list=[]
for i in C1_values:
    ctr_avg=train.loc[np.where((train.C1 == i))].click.mean()
    ctr_avg_list.append(ctr_avg)
    print("for C1 value: {},  click through rate: {}".format(i,ctr_avg))
```

```
for C1 value: 1001,  click through rate: 0.02040816326530612
for C1 value: 1002,  click through rate: 0.2116454039908407
for C1 value: 1005,  click through rate: 0.1690423773656395
for C1 value: 1007,  click through rate: 0.048109965635738834
for C1 value: 1008,  click through rate: 0.1532258064516129
for C1 value: 1010,  click through rate: 0.094985277058981
for C1 value: 1012,  click through rate: 0.16701680672268907
```

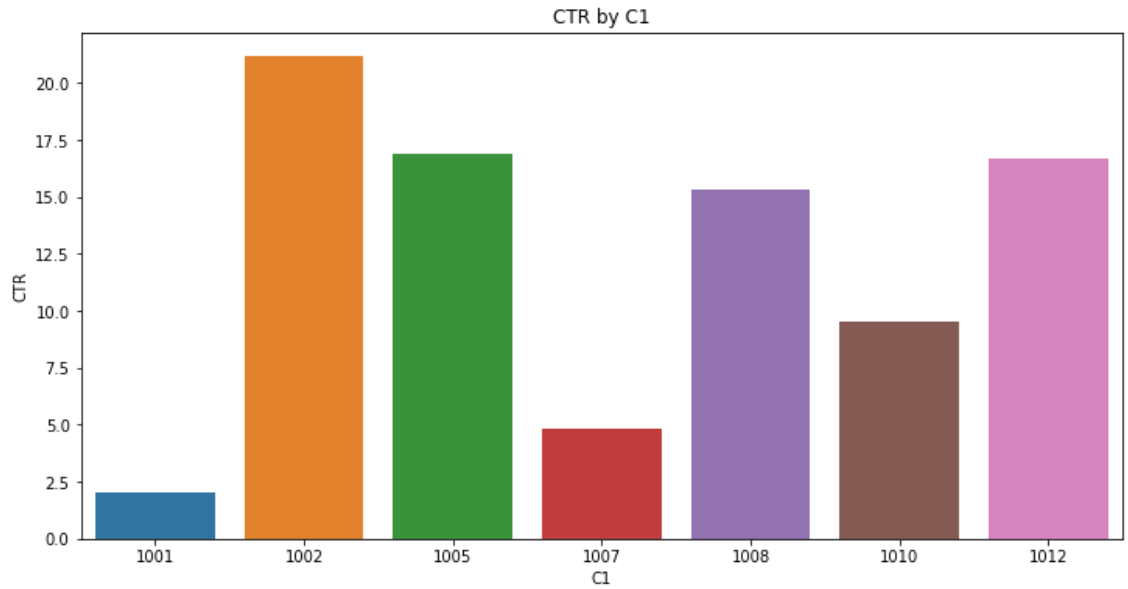
In [23]:

```
train.groupby(['C1', 'click']).size().unstack().plot(kind='bar', figsize=(12,6), title='
```



In [24]:

```
df_c1 = train[['C1','click']].groupby(['C1']).count().reset_index()
df_c1 = df_c1.rename(columns={'click': 'impressions'})
df_c1['clicks'] = df_click[['C1','click']].groupby(['C1']).count().reset_index()['click'
df_c1['CTR'] = df_c1['clicks']/df_c1['impressions']*100

plt.figure(figsize=(12,6))
sns.barplot(y='CTR', x='C1', data=df_c1)
plt.title('CTR by C1');
```



The average CTR in the data is 0.17.

In [98]:

```
train['click'].mean()
```

Out[98]:

```
0.169577
```

In [25]:

```
df_c1.CTR.describe()
```

Out[25]:

```
count     7.000000
mean     12.349054
std       7.041038
min       2.040816
25%       7.154762
50%      15.322581
75%      16.802959
max      21.164540
Name: CTR, dtype: float64
```

Banner position

In [26]:

```
print(train.banner_pos.value_counts()/len(train))
```

```
0    0.718949
1    0.279209
7    0.001108
2    0.000361
4    0.000200
5    0.000124
3    0.000049
Name: banner_pos, dtype: float64
```

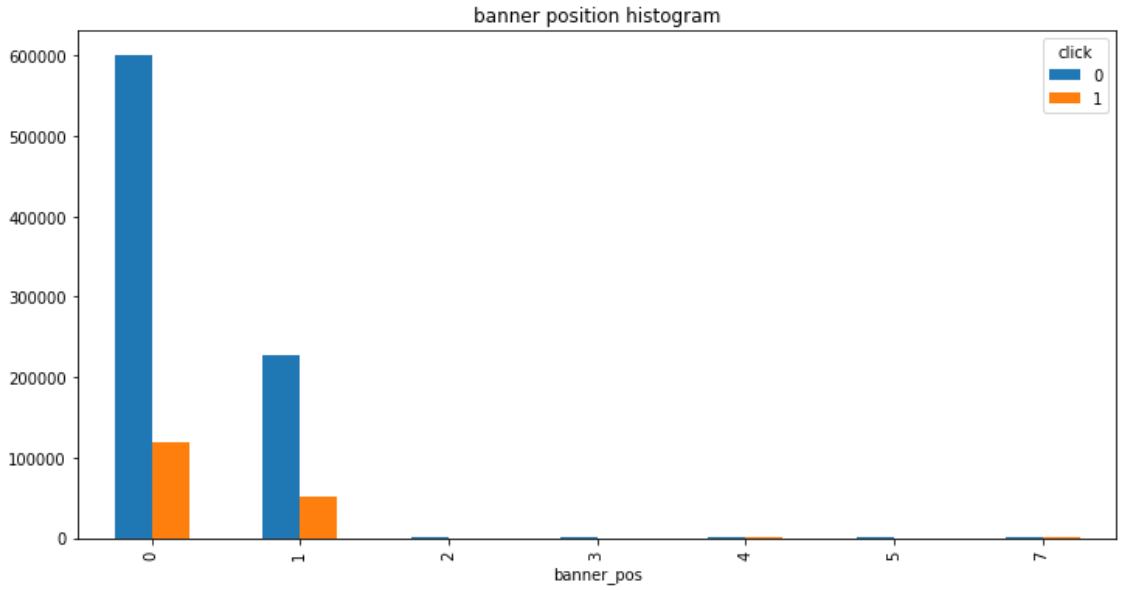
In [27]:

```
banner_pos = train.banner_pos.unique()
banner_pos.sort()
ctr_avg_list=[]
for i in banner_pos:
    ctr_avg=train.loc[np.where((train.banner_pos == i))].click.mean()
    ctr_avg_list.append(ctr_avg)
    print("for banner position: {},  click through rate: {}".format(i,ctr_avg))
```

```
for banner position: 0,  click through rate: 0.16418828039262867
for banner position: 1,  click through rate: 0.18292748442922685
for banner position: 2,  click through rate: 0.09695290858725762
for banner position: 3,  click through rate: 0.1836734693877551
for banner position: 4,  click through rate: 0.185
for banner position: 5,  click through rate: 0.1532258064516129
for banner position: 7,  click through rate: 0.3240072202166065
```

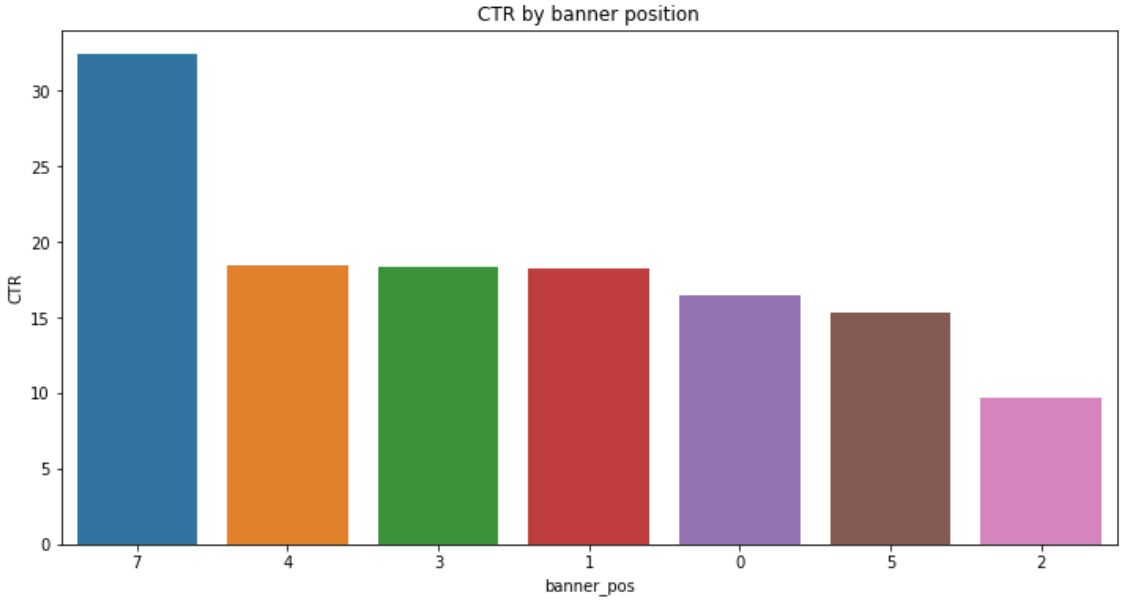
In [28]:

```
train.groupby(['banner_pos', 'click']).size().unstack().plot(kind='bar', figsize=(12,6),
```



In [29]:

```
df_banner = train[['banner_pos','click']].groupby(['banner_pos']).count().reset_index()
df_banner = df_banner.rename(columns={'click': 'impressions'})
df_banner['clicks'] = df_click[['banner_pos','click']].groupby(['banner_pos']).count().r
df_banner['CTR'] = df_banner['clicks']/df_banner['impressions']*100
sort_banners = df_banner.sort_values(by='CTR',ascending=False)['banner_pos'].tolist()
plt.figure(figsize=(12,6))
sns.barplot(y='CTR', x='banner_pos', data=df_banner, order=sort_banners)
plt.title('CTR by banner position');
```



In [30]:

```
df_banner.CTR.describe()
```

Out[30]:

```
count     7.000000
mean     18.428217
std       6.894499
min       9.695291
25%      15.870704
50%      18.292748
75%      18.433673
max      32.400722
Name: CTR, dtype: float64
```

Site features

site id

In [31]:

```
print("There are {} sites in the data set".format(train.site_id.nunique()))
```

```
There are 2624 sites in the data set
```

In [32]:

```
print('The top 10 site ids that have the most impressions')
print((train.site_id.value_counts()/len(train))[0:10])
```

```
The top 10 site ids that have the most impressions
85f751fd    0.360413
1fbe01fe    0.159926
e151e245    0.064992
d9750ee7    0.024076
5b08c53b    0.022792
5b4d2eda    0.019417
856e6d3f    0.019001
a7853007    0.011380
b7e9786d    0.009176
6399eda6    0.008599
Name: site_id, dtype: float64
```

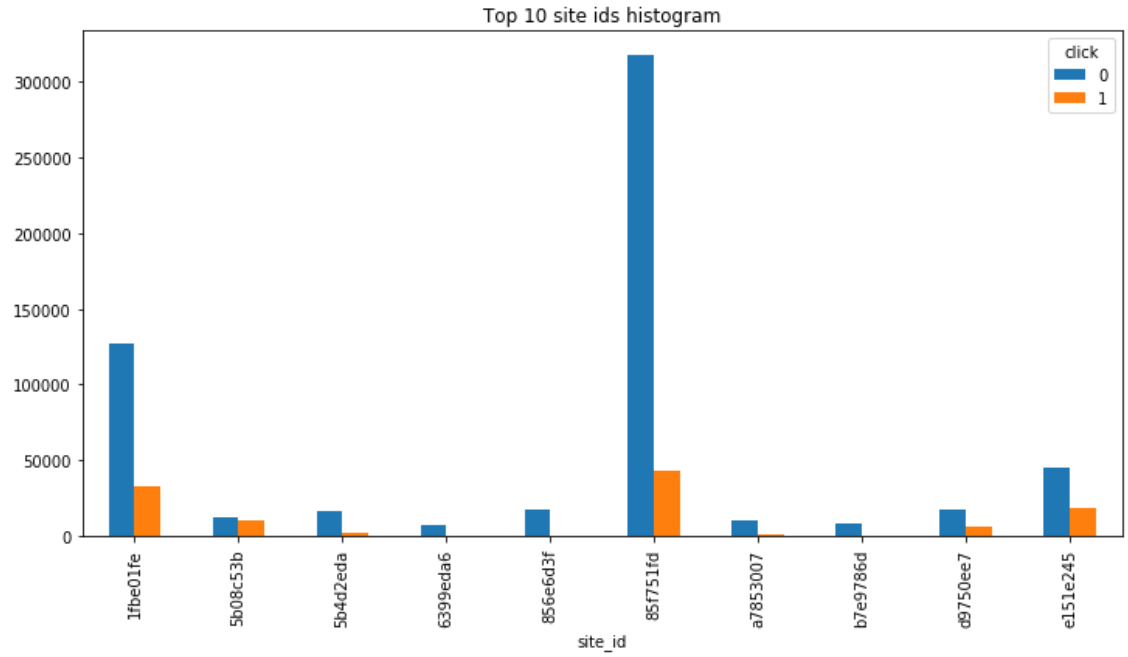
In [33]:

```
top10_ids = (train.site_id.value_counts()/len(train))[0:10].index
click_avg_list=[]

for i in top10_ids:
    click_avg=train.loc[np.where((train.site_id == i))].click.mean()
    click_avg_list.append(click_avg)
    print("for site id value: {},  click through rate: {}".format(i,click_avg))
```

```
for site id value: 85f751fd,  click through rate: 0.1196266505370228
for site id value: 1fbe01fe,  click through rate: 0.2047071770693946
for site id value: e151e245,  click through rate: 0.2956363860167405
for site id value: d9750ee7,  click through rate: 0.28310350556570857
for site id value: 5b08c53b,  click through rate: 0.4668304668304668
for site id value: 5b4d2eda,  click through rate: 0.12432404593912551
for site id value: 856e6d3f,  click through rate: 0.040629440555760225
for site id value: a7853007,  click through rate: 0.10439367311072056
for site id value: b7e9786d,  click through rate: 0.07301656495204882
for site id value: 6399eda6,  click through rate: 0.10291894406326317
```

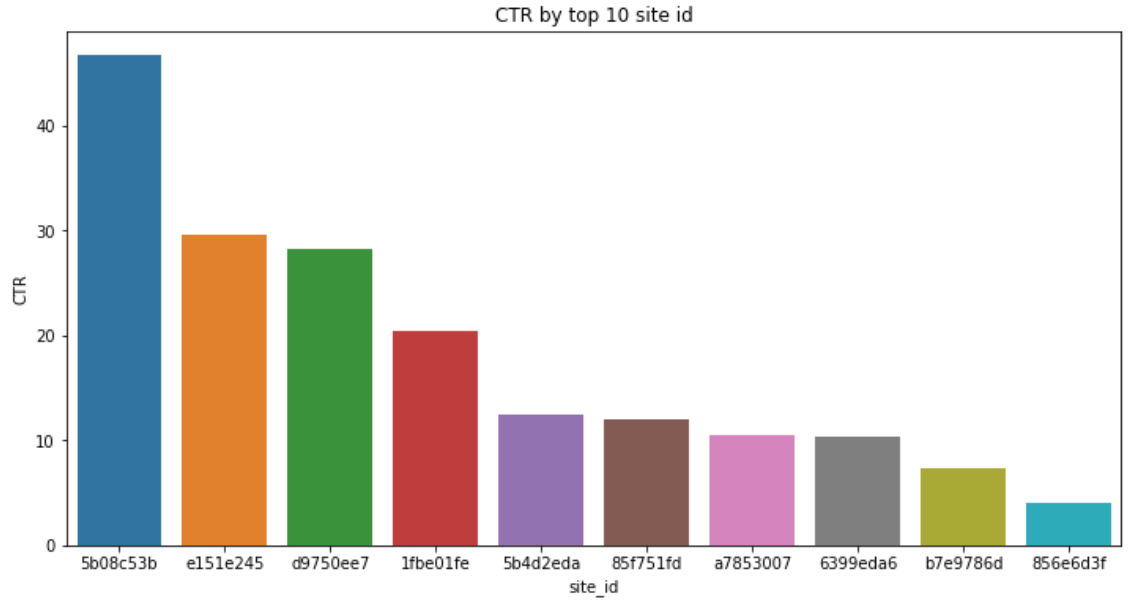
In [34]:

```
top10_sites = train[(train.site_id.isin((train.site_id.value_counts()/len(train))[0:10].
top10_sites_click = top10_sites[top10_sites['click'] == 1]
top10_sites.groupby(['site_id', 'click']).size().unstack().plot(kind='bar', figsize=(12,
```



In [35]:

```
df_site = top10_sites[['site_id','click']].groupby(['site_id']).count().reset_index()
df_site = df_site.rename(columns={'click': 'impressions'})
df_site['clicks'] = top10_sites_click[['site_id','click']].groupby(['site_id']).count().
df_site['CTR'] = df_site['clicks']/df_site['impressions']*100
sort_site = df_site.sort_values(by='CTR',ascending=False)['site_id'].tolist()
plt.figure(figsize=(12,6))
sns.barplot(y='CTR', x='site_id', data=df_site, order=sort_site)
plt.title('CTR by top 10 site id');
```



site domain

In [36]:

```
print("There are {} site domains in the data set".format(train.site_domain.nunique()))
```

```
There are 2856 site domains in the data set
```

In [37]:

```
print('The top 10 site domains that have the most impressions')
print((train.site_domain.value_counts()/len(train))[0:10])
```

```
The top 10 site domains that have the most impressions
c4e18dd6    0.373743
f3845767    0.159926
7e091613    0.081999
7687a86e    0.032128
98572c79    0.024894
16a36ef3    0.021436
58a89a43    0.019001
b12b9f85    0.009287
9d54950b    0.009274
17d996e6    0.008683
Name: site_domain, dtype: float64
```

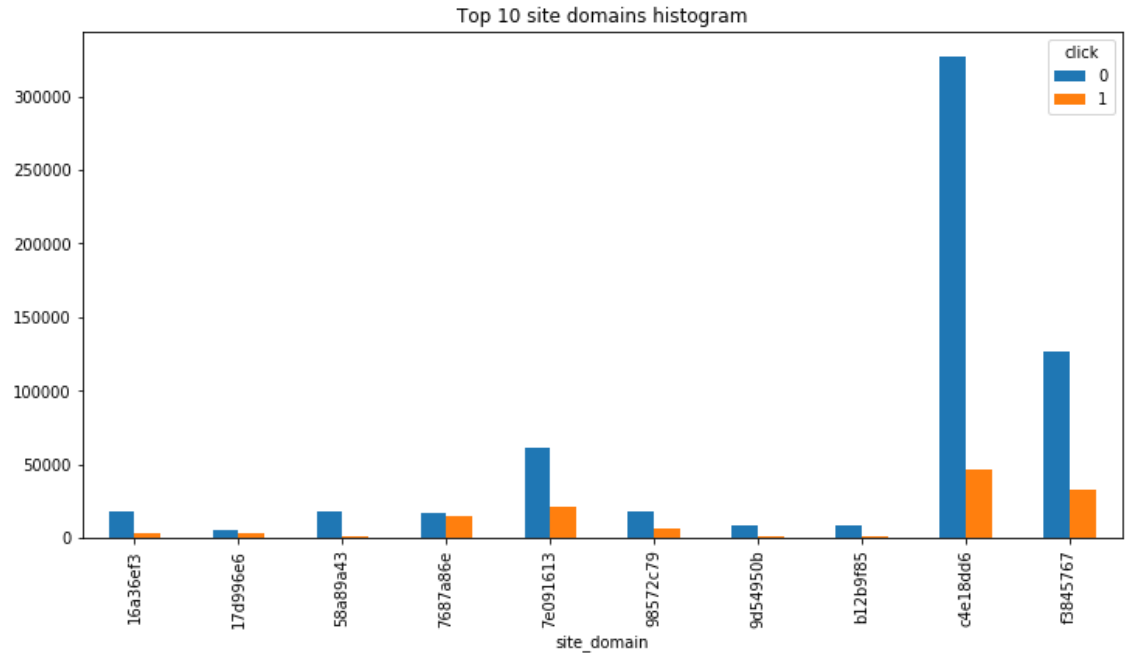
In [38]:

```
top10_domains = (train.site_domain.value_counts()/len(train))[0:10].index
click_avg_list=[]

for i in top10_domains:
    click_avg=train.loc[np.where((train.site_domain == i))].click.mean()
    click_avg_list.append(click_avg)
    print("for site domain value: {},  click through rate: {}".format(i,click_avg))
```

```
for site domain value: c4e18dd6,  click through rate: 0.12354746443411649
for site domain value: f3845767,  click through rate: 0.2047071770693946
for site domain value: 7e091613,  click through rate: 0.2581494896279223
for site domain value: 7687a86e,  click through rate: 0.4607507470119522
for site domain value: 98572c79,  click through rate: 0.2777777777777778
for site domain value: 16a36ef3,  click through rate: 0.13509983205821982
for site domain value: 58a89a43,  click through rate: 0.04062944055576022
5
for site domain value: b12b9f85,  click through rate: 0.07408205017766771
for site domain value: 9d54950b,  click through rate: 0.11063187405650204
for site domain value: 17d996e6,  click through rate: 0.32120234941840375
```

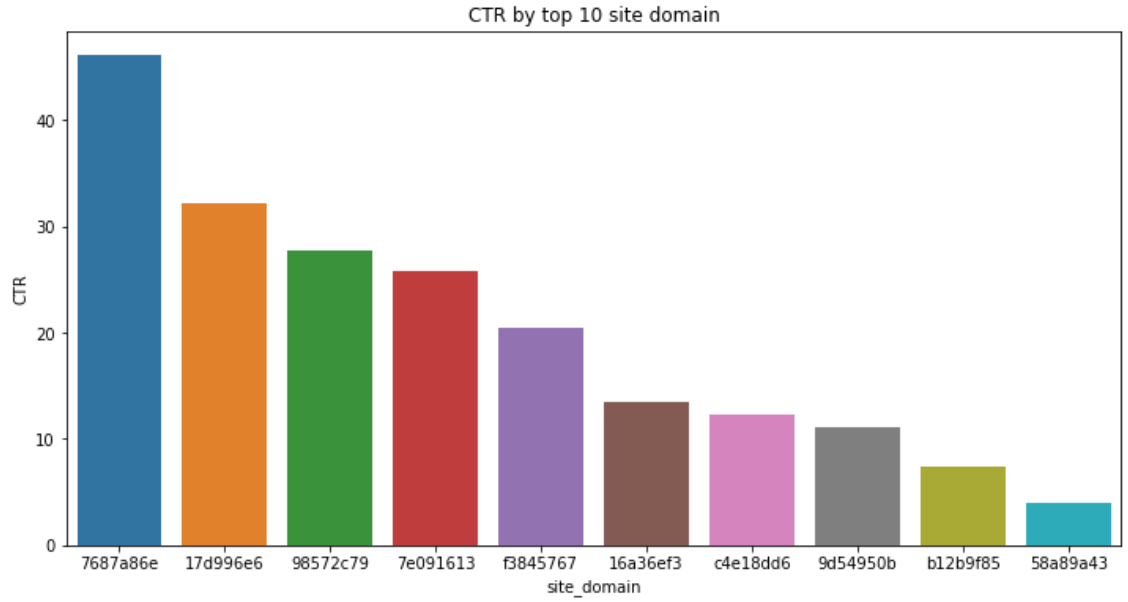
In [39]:

```
top10_domain = train[(train.site_domain.isin((train.site_domain.value_counts()/len(train
top10_domain_click = top10_domain[top10_domain['click'] == 1]
top10_domain.groupby(['site_domain', 'click']).size().unstack().plot(kind='bar', figsize
```



In [40]:

```
df_domain = top10_domain[['site_domain','click']].groupby(['site_domain']).count().reset
df_domain = df_domain.rename(columns={'click': 'impressions'})
df_domain['clicks'] = top10_domain_click[['site_domain','click']].groupby(['site_domain'
df_domain['CTR'] = df_domain['clicks']/df_domain['impressions']*100
sort_domain = df_domain.sort_values(by='CTR',ascending=False)['site_domain'].tolist()
plt.figure(figsize=(12,6))
sns.barplot(y='CTR', x='site_domain', data=df_domain, order=sort_domain)
plt.title('CTR by top 10 site domain');
```



Similar with the site_id feature, the site_domain feature seems important as well.

site category

In [41]:

```
print("There are {} site categories in the data set".format(train.site_category.nunique(
```

```
There are 22 site categories in the data set
```

In [42]:

```
print('The top 10 site categories that have the most impressions')
print((train.site_category.value_counts()/len(train))[0:10])
```

```
The top 10 site categories that have the most impressions
50e219e0    0.408487
f028772b    0.313889
28905ebd    0.181906
3e814130    0.075684
f66779e6    0.006175
75fa27f6    0.003969
335d28a8    0.003412
76b2941d    0.002638
c0dd3be3    0.001082
72722551    0.000697
Name: site_category, dtype: float64
```

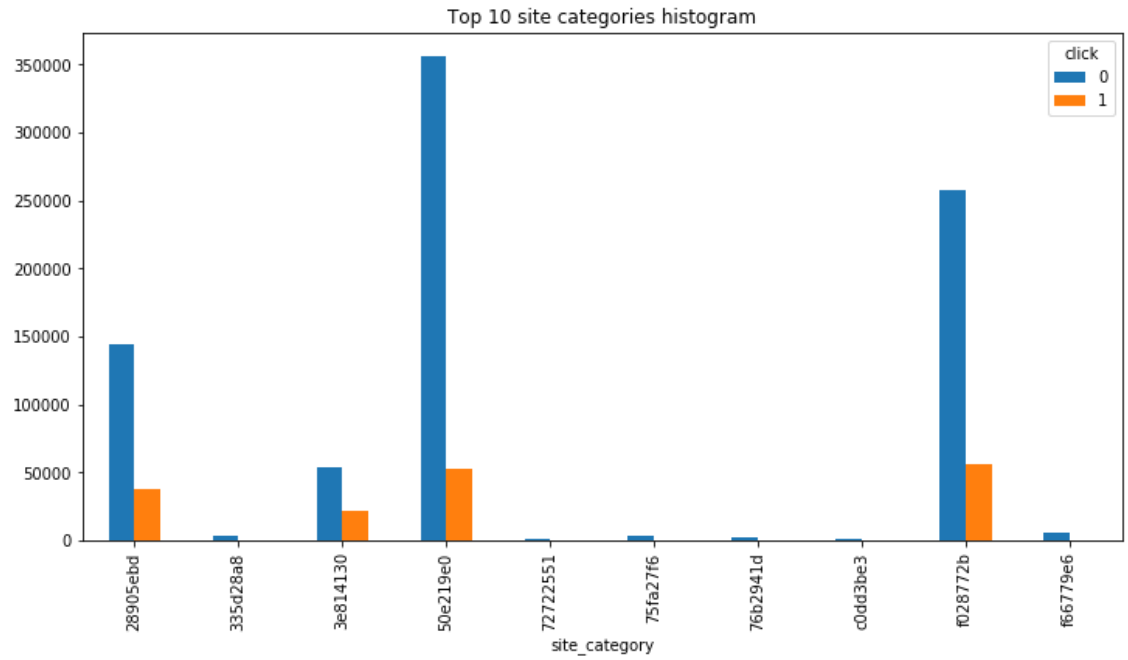
In [43]:

```
top10_categories = (train.site_category.value_counts()/len(train))[0:10].index
click_avg_list=[]

for i in top10_categories:
    click_avg=train.loc[np.where((train.site_category == i))].click.mean()
    click_avg_list.append(click_avg)
    print("for site category value: {},  click through rate: {}".format(i,click_avg))
```

```
for site category value: 50e219e0,  click through rate: 0.129372538171349
4
for site category value: f028772b,  click through rate: 0.178340113861906
6
for site category value: 28905ebd,  click through rate: 0.206821105406088
86
for site category value: 3e814130,  click through rate: 0.284075894508746
9
for site category value: f66779e6,  click through rate: 0.04
for site category value: 75fa27f6,  click through rate: 0.111111111111111
1
for site category value: 335d28a8,  click through rate: 0.089976553341148
89
for site category value: 76b2941d,  click through rate: 0.023123578468536
77
for site category value: c0dd3be3,  click through rate: 0.105360443622920
52
for site category value: 72722551,  click through rate: 0.063127690100430
41
```

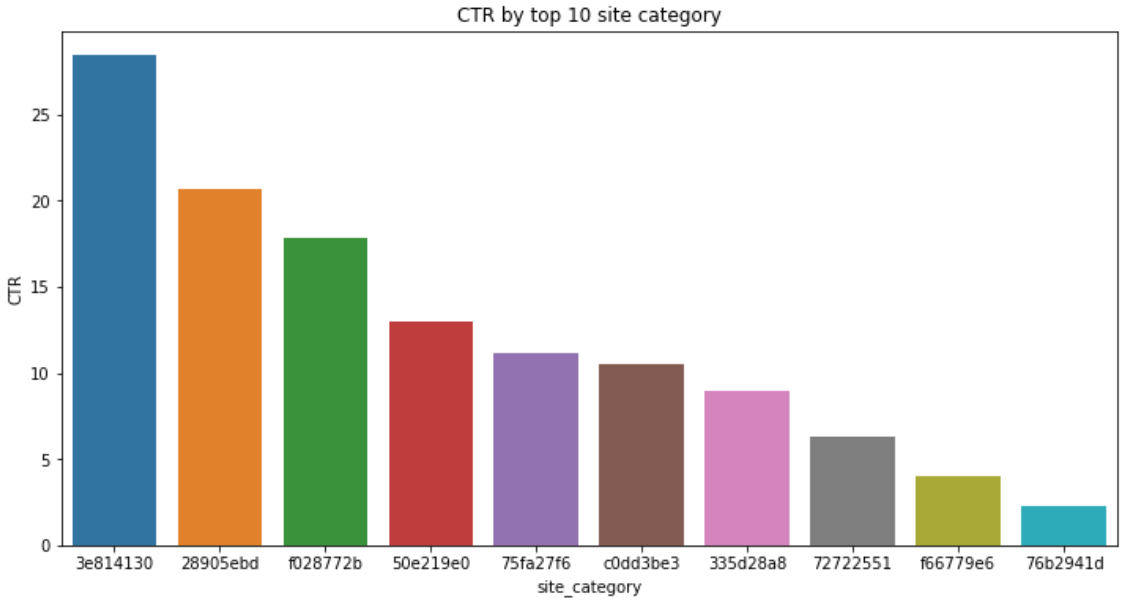
In [44]:

```
top10_category = train[(train.site_category.isin((train.site_category.value_counts()/ler
top10_category_click = top10_category[top10_category['click'] == 1]
top10_category.groupby(['site_category', 'click']).size().unstack().plot(kind='bar', fig
```



In [45]:

```
df_category = top10_category[['site_category','click']].groupby(['site_category']).count
df_category = df_category.rename(columns={'click': 'impressions'})
df_category['clicks'] = top10_category_click[['site_category','click']].groupby(['site_c
df_category['CTR'] = df_category['clicks']/df_category['impressions']*100
sort_category = df_category.sort_values(by='CTR',ascending=False)['site_category'].tolis
plt.figure(figsize=(12,6))
sns.barplot(y='CTR', x='site_category', data=df_category, order=sort_category)
plt.title('CTR by top 10 site category');
```



Device features

device id

In [46]:

```
print("There are {} devices in the data set".format(train.device_id.nunique()))
```

```
There are 150331 devices in the data set
```

In [47]:

```
print('The top 10 devices that have the most impressions')
print((train.device_id.value_counts()/len(train))[0:10])
```

```
The top 10 devices that have the most impressions
a99f214a    0.825328
0f7c61dc    0.000557
c357dbff    0.000474
936e92fb    0.000352
afeffc18    0.000212
cef4c8cc    0.000107
987552d1    0.000106
28dc8687    0.000101
d857ffbb    0.000097
b09da1c4    0.000094
Name: device_id, dtype: float64
```

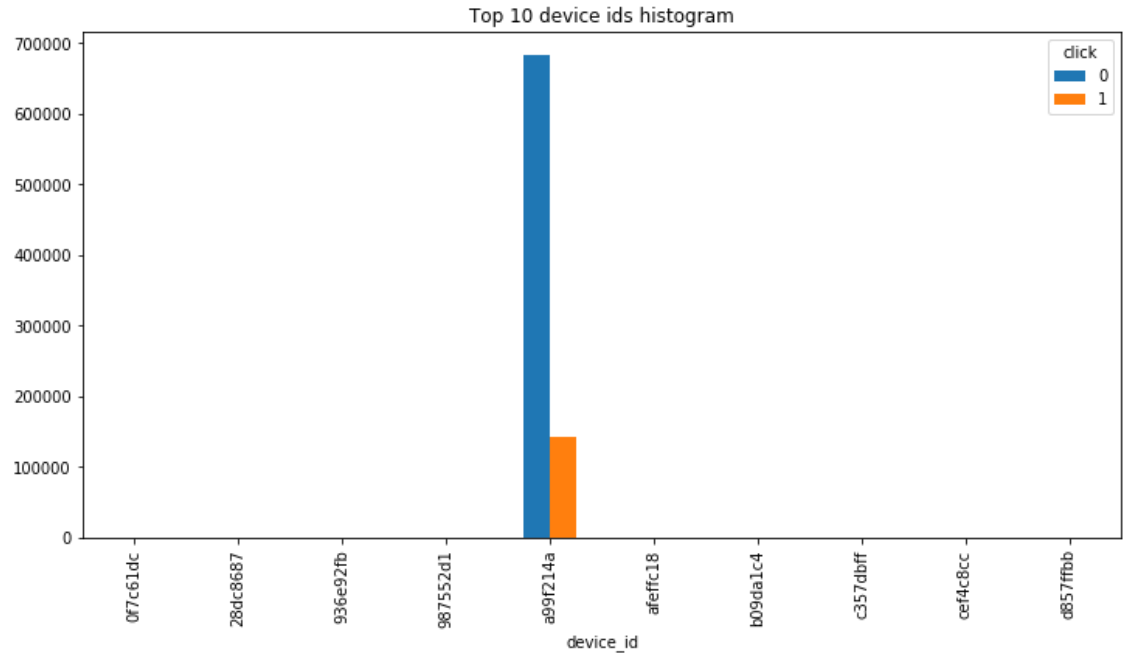
In [48]:

```
top10_devices = (train.device_id.value_counts()/len(train))[0:10].index
click_avg_list=[]

for i in top10_devices:
    click_avg=train.loc[np.where((train.device_id == i))].click.mean()
    click_avg_list.append(click_avg)
    print("for device id value: {},  click through rate: {}".format(i,click_avg))
```

```
for device id value: a99f214a,  click through rate: 0.1735952251710835
for device id value: 0f7c61dc,  click through rate: 0.7432675044883303
for device id value: c357dbff,  click through rate: 0.6540084388185654
for device id value: 936e92fb,  click through rate: 0.0625
for device id value: afeffc18,  click through rate: 0.21226415094339623
for device id value: cef4c8cc,  click through rate: 0.2523364485981308
for device id value: 987552d1,  click through rate: 0.0
for device id value: 28dc8687,  click through rate: 0.0
for device id value: d857ffbb,  click through rate: 0.18556701030927836
for device id value: b09da1c4,  click through rate: 0.14893617021276595
```

In [49]:

```
top10_device = train[(train.device_id.isin((train.device_id.value_counts()/len(train))[0
top10_device_click = top10_device[top10_device['click'] == 1]
top10_device.groupby(['device_id', 'click']).size().unstack().plot(kind='bar', figsize=(
```



You will see that most of device_id is a99f214a : Approx. 83% of the data, and the second major device_id is only 0.05% of the data. And there are some extremely high CTR here with device id at 0f7c61dc.

device ip

Device ip is more of a users ip address, so, there are a lot of them.

In [55]:

```
print("There are {} device ips in the data set".format(train.device_ip.nunique()))
print("There are {} device types in the data set".format(train.device_type.nunique()))
print("There are {} device models in the data set".format(train.device_model.nunique()))
print("There are {} device cnn types in the data set".format(train.device_conn_type.nuni
```

```
There are 555038 device ips in the data set
There are 5 device types in the data set
There are 5166 device models in the data set
There are 4 device cnn types in the data set
```

device type

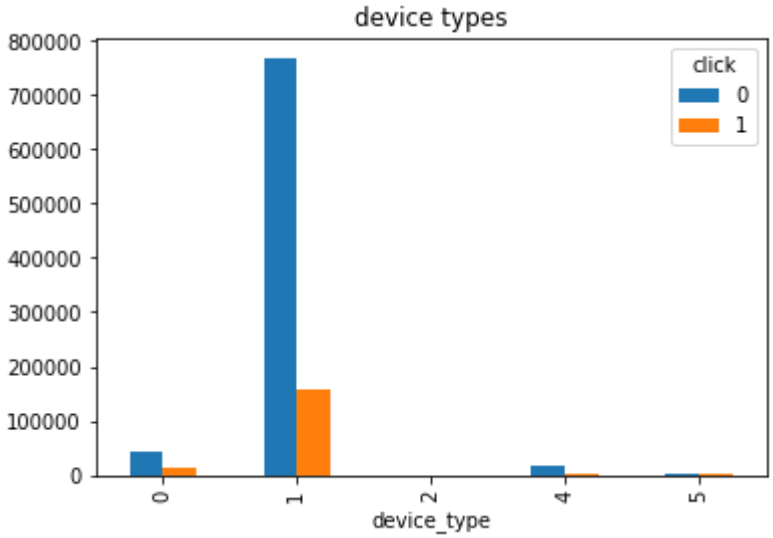
In [58]:

```
print('The impressions by device types')
print((train.device_type.value_counts()/len(train)))
```

```
The impressions by device types
1    0.922559
0    0.055026
4    0.019242
5    0.003172
2    0.000001
Name: device_type, dtype: float64
```

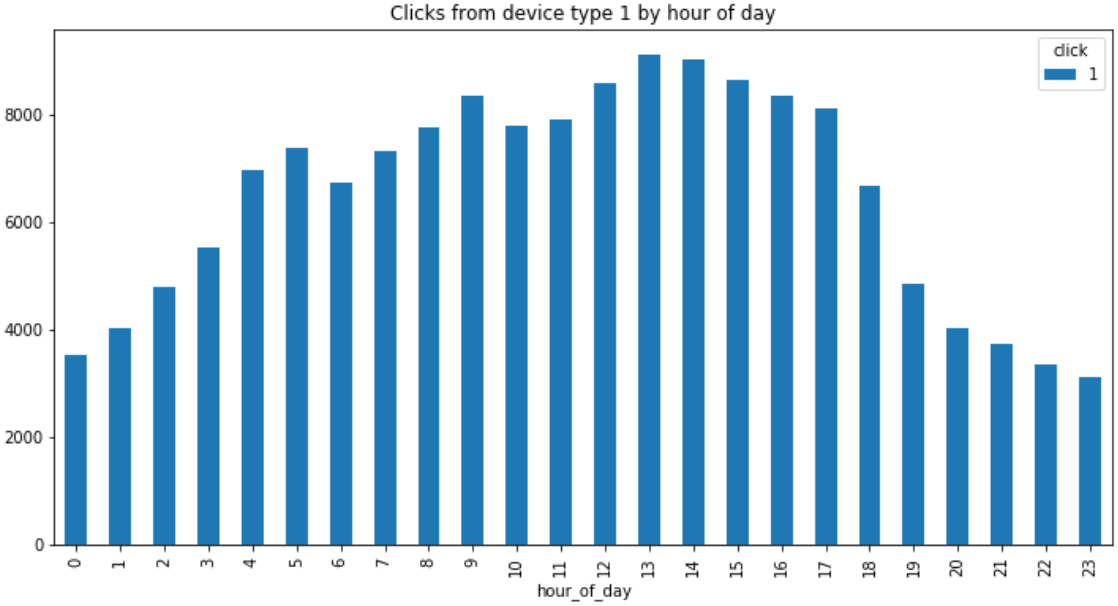
In [57]:

```
train[['device_type','click']].groupby(['device_type','click']).size().unstack().plot(ki
```



In [66]:

```
df_click[df_click['device_type']==1].groupby(['hour_of_day', 'click']).size().unstack().
```



In [71]:

```
device_type_click = df_click.groupby('device_type').agg({'click':'sum'}).reset_index()
device_type_impression = train.groupby('device_type').agg({'click':'count'}).reset_index
merged_device_type = pd.merge(left = device_type_click , right = device_type_impression,
```

In [75]:

```
merged_device_type['CTR'] = merged_device_type['click'] / merged_device_type['impressior
```

In [76]:

```
merged_device_type
```

Out[76]:

	device_type	click	impressions	CTR
0	0	11646	55026	21.164540
1	1	155802	922559	16.888026
2	4	1829	19242	9.505249
3	5	300	3172	9.457755

app features

In [80]:

```
print("There are {} apps in the data set".format(train.app_id.nunique()))
print("There are {} app domains in the data set".format(train.app_domain.nunique()))
print("There are {} app categories in the data set".format(train.app_category.nunique())
```

```
There are 3098 apps in the data set
There are 185 app domains in the data set
There are 25 app categories in the data set
```

In [81]:

```
print('The impressions by app categories')
print((train.app_category.value_counts()/len(train)))
```

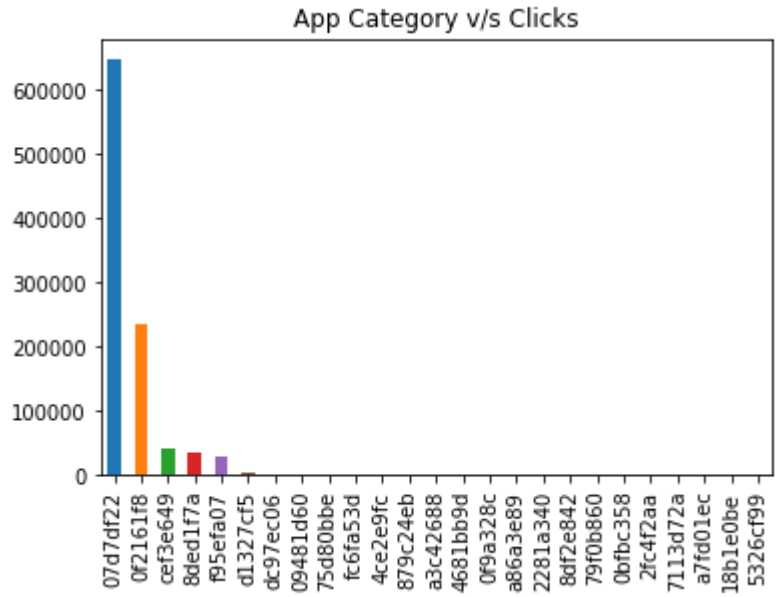
```
The impressions by app categories
07d7df22    0.647835
0f2161f8    0.236489
cef3e649    0.042549
8ded1f7a    0.036035
f95efa07    0.028202
d1327cf5    0.003043
dc97ec06    0.001385
09481d60    0.001359
75d80bbe    0.000960
fc6fa53d    0.000566
4ce2e9fc    0.000502
879c24eb    0.000283
a3c42688    0.000277
4681bb9d    0.000163
0f9a328c    0.000129
a86a3e89    0.000070
2281a340    0.000053
8df2e842    0.000039
79f0b860    0.000015
0bfbc358    0.000011
2fc4f2aa    0.000010
7113d72a    0.000009
a7fd01ec    0.000008
18b1e0be    0.000004
5326cf99    0.000004
Name: app_category, dtype: float64
```

In [83]:

```
train['app_category'].value_counts().plot(kind='bar', title='App Category v/s Clicks')
```

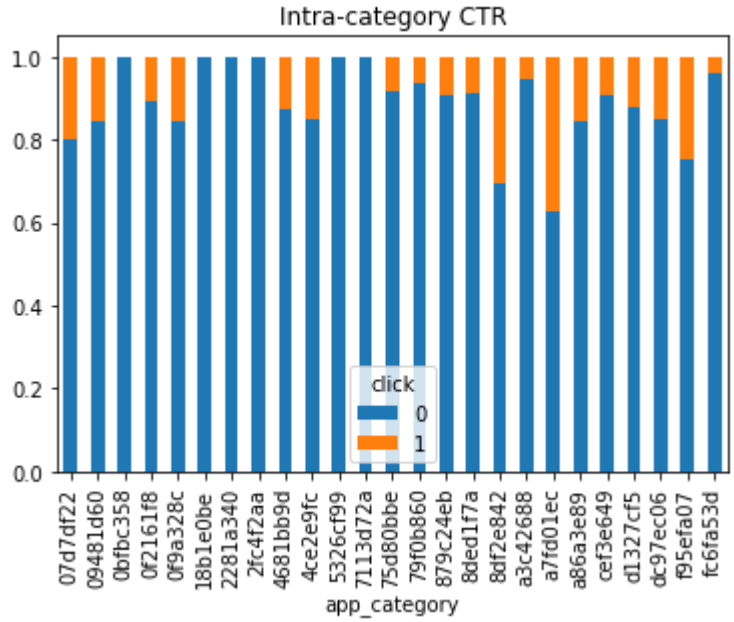
Out[83]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c2c4a002b0>
```



In [89]:

```
train_app_category = train.groupby(['app_category', 'click']).size().unstack()
train_app_category.div(train_app_category.sum(axis=1), axis=0).plot(kind='bar', stacked=
```



C14 - C21 features

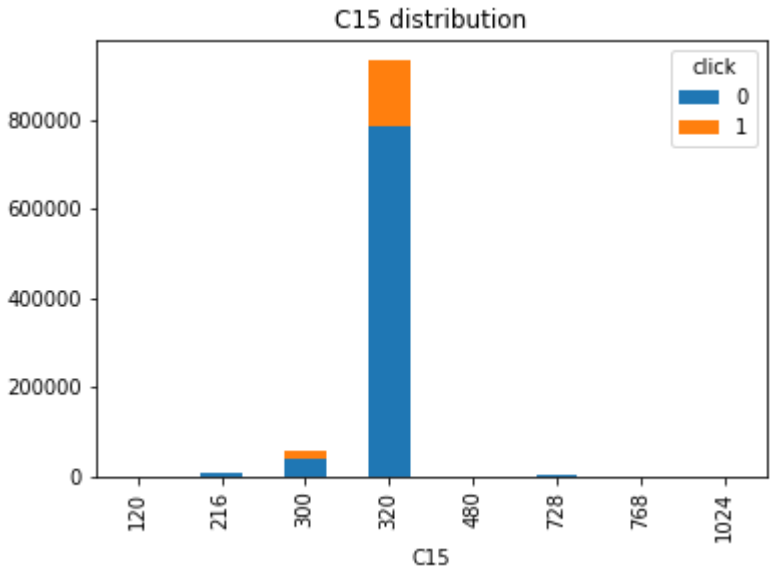
In [94]:

```
print("There are {} C14 in the data set".format(train.C14.nunique()))
print("There are {} C15 in the data set".format(train.C15.nunique()))
print("There are {} C16 in the data set".format(train.C16.nunique()))
print("There are {} C17 in the data set".format(train.C17.nunique()))
print("There are {} C18 in the data set".format(train.C18.nunique()))
print("There are {} C19 in the data set".format(train.C19.nunique()))
print("There are {} C20 in the data set".format(train.C20.nunique()))
```

```
There are 2253 C14 in the data set
There are 8 C15 in the data set
There are 9 C16 in the data set
There are 421 C17 in the data set
There are 4 C18 in the data set
There are 66 C19 in the data set
There are 162 C20 in the data set
```

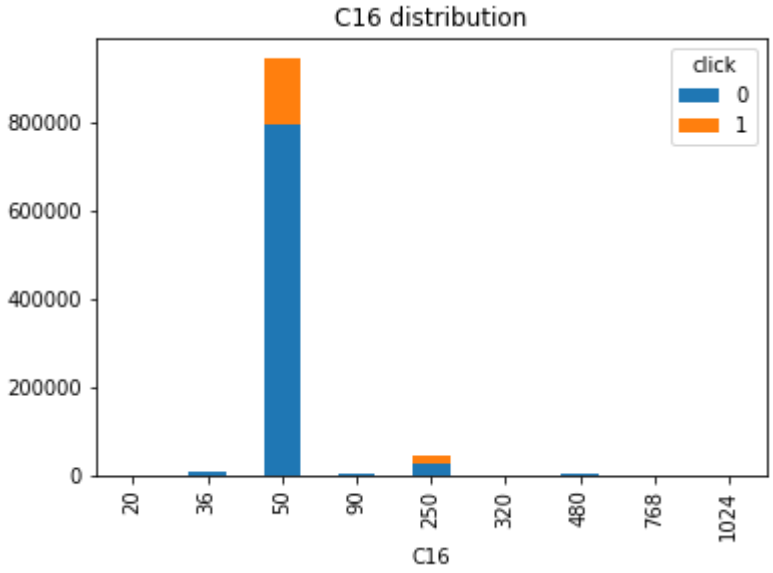
In [95]:

```
train.groupby(['C15', 'click']).size().unstack().plot(kind='bar', stacked=True, title='C
```



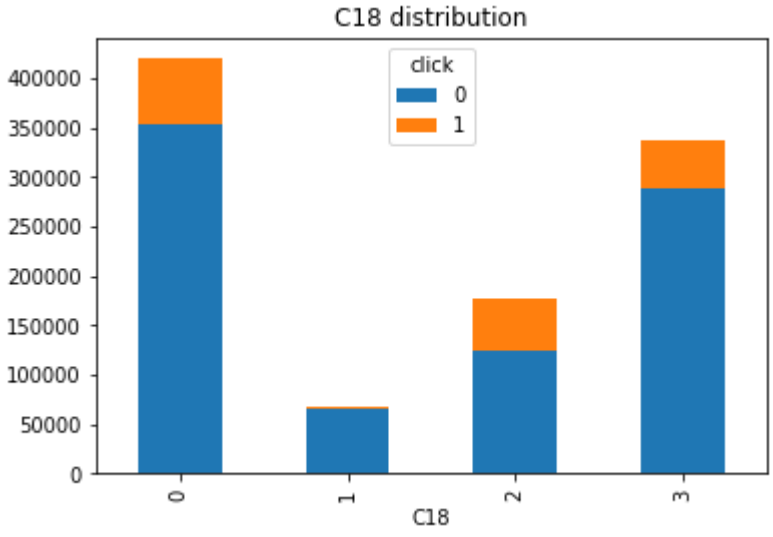
In [96]:

```
train.groupby(['C16', 'click']).size().unstack().plot(kind='bar', stacked=True, title='C
```



In [97]:

```
train.groupby(['C18', 'click']).size().unstack().plot(kind='bar', stacked=True, title='C
```



In [99]:

```
train.head(3)
```

Out[99]:

	id	click	hour	C1	banner_pos	site_id	site_domain	site_category	ap
0	601394868	0	2014-10-21	1005	0	030440fe	08ba7db9	76b2941d	ecad2
1	-59070594	0	2014-10-21	1005	1	0eb72673	d2f72222	f028772b	ecad2
2	-1859646727	0	2014-10-21	1005	0	6c5b482c	7687a86e	3e814130	ecad2

3 rows × 26 columns

In [102]:

```
def convert_obj_to_int(self):

    object_list_columns = self.columns
    object_list_dtypes = self.dtypes
    new_col_suffix = '_int'
    for index in range(0,len(object_list_columns)):
        if object_list_dtypes[index] == object :
            self[object_list_columns[index]+new_col_suffix] = self[object_list_columns[i
            self.drop([object_list_columns[index]],inplace=True,axis=1)
    return self
train = convert_obj_to_int(train)
```

In [103]:

```
train.head(3)
```

Out[103]:

	id	click	hour	C1	banner_pos	device_type	device_conn_type	C14	C15
0	601394868	0	2014-10-21	1005	0	1	0	18993	320
1	-59070594	0	2014-10-21	1005	1	1	0	16208	320
2	-1859646727	0	2014-10-21	1005	0	1	0	17654	300

3 rows × 26 columns

In [105]:

```
train.drop('hour', axis=1, inplace=True)
```

In [109]:

```
train.drop('id', axis=1, inplace=True)
```

In [112]:

```
import lightgbm as lgb
X_train = train.loc[:, train.columns != 'click']
y_target = train.click.values
#create lightgbm dataset
msk = np.random.rand(len(X_train)) < 0.8
lgb_train = lgb.Dataset(X_train[msk], y_target[msk])
lgb_eval = lgb.Dataset(X_train[~msk], y_target[~msk], reference=lgb_train)
```

In [121]:

```
# specify your configurations as a dict
params = {
    'task': 'train',
    'boosting_type': 'gbdt',
    'objective': 'binary',
    'metric': { 'binary_logloss'},
    'num_leaves': 31, # defauly leaves(31) amount for each tree
    'learning_rate': 0.08,
    'feature_fraction': 0.7, # will select 70% features before training each tree
    'bagging_fraction': 0.3, #feature_fraction, but this will random select part of data
    'bagging_freq': 5, #  perform bagging at every 5 iteration
    'verbose': 0
}

print('Start training...')
# train
gbm = lgb.train(params,
                lgb_train,
                num_boost_round=4000,
                valid_sets=lgb_eval,
                early_stopping_rounds=500)
```

```
Start training...
[1]     valid_0's binary_logloss: 0.450675
Training until validation scores don't improve for 500 rounds.
[2]     valid_0's binary_logloss: 0.446337
[3]     valid_0's binary_logloss: 0.442399
[4]     valid_0's binary_logloss: 0.439119
[5]     valid_0's binary_logloss: 0.436433
[6]     valid_0's binary_logloss: 0.433844
[7]     valid_0's binary_logloss: 0.431648
[8]     valid_0's binary_logloss: 0.429778
[9]     valid_0's binary_logloss: 0.428096
[10]    valid_0's binary_logloss: 0.426571
[11]    valid_0's binary_logloss: 0.425303
[12]    valid_0's binary_logloss: 0.424066
[13]    valid_0's binary_logloss: 0.422933
[14]    valid_0's binary_logloss: 0.421888
[15]    valid_0's binary_logloss: 0.421023
[16]    valid_0's binary_logloss: 0.420212
[17]    valid_0's binary_logloss: 0.419503
[18]    valid_0's binary_logloss: 0.418866
[19]    valid_0's binary_logloss: 0.418242
[20]    valid_0's binary_logloss: 0.417684
[21]    valid_0's binary_logloss: 0.417146
[22]    valid_0's binary_logloss: 0.416616
[23]    valid_0's binary_logloss: 0.416136
[24]    valid_0's binary_logloss: 0.415746
[25]    valid_0's binary_logloss: 0.41534
[26]    valid_0's binary_logloss: 0.414983
[27]    valid_0's binary_logloss: 0.414676
[28]    valid_0's binary_logloss: 0.414381
[29]    valid_0's binary_logloss: 0.414007
[30]    valid_0's binary_logloss: 0.413762
[31]    valid_0's binary_logloss: 0.4135
[32]    valid_0's binary_logloss: 0.413117
[33]    valid_0's binary_logloss: 0.412882
[34]    valid_0's binary_logloss: 0.412639
[35]    valid_0's binary_logloss: 0.412344
[36]    valid_0's binary_logloss: 0.412116
[37]    valid_0's binary_logloss: 0.411836
[38]    valid_0's binary_logloss: 0.411575
[39]    valid_0's binary_logloss: 0.411429
[40]    valid_0's binary_logloss: 0.411201
[41]    valid_0's binary_logloss: 0.411003
[42]    valid_0's binary_logloss: 0.410884
[43]    valid_0's binary_logloss: 0.410722
[44]    valid_0's binary_logloss: 0.410587
[45]    valid_0's binary_logloss: 0.41044
[46]    valid_0's binary_logloss: 0.410222
[47]    valid_0's binary_logloss: 0.410043
[48]    valid_0's binary_logloss: 0.409975
[49]    valid_0's binary_logloss: 0.409847
[50]    valid_0's binary_logloss: 0.409714
[51]    valid_0's binary_logloss: 0.4096
[52]    valid_0's binary_logloss: 0.409498
[53]    valid_0's binary_logloss: 0.409435
[54]    valid_0's binary_logloss: 0.40931
[55]    valid_0's binary_logloss: 0.409212
[56]    valid_0's binary_logloss: 0.409091
[57]    valid_0's binary_logloss: 0.408997
[58]    valid_0's binary_logloss: 0.408876
[59]    valid_0's binary_logloss: 0.408774
```

```
[60]    valid_0's binary_logloss: 0.408627
[61]    valid_0's binary_logloss: 0.408526
[62]    valid_0's binary_logloss: 0.408473
[63]    valid_0's binary_logloss: 0.408433
[64]    valid_0's binary_logloss: 0.408393
[65]    valid_0's binary_logloss: 0.408339
[66]    valid_0's binary_logloss: 0.408268
[67]    valid_0's binary_logloss: 0.408166
[68]    valid_0's binary_logloss: 0.408086
[69]    valid_0's binary_logloss: 0.408015
[70]    valid_0's binary_logloss: 0.407957
[71]    valid_0's binary_logloss: 0.407888
[72]    valid_0's binary_logloss: 0.407848
[73]    valid_0's binary_logloss: 0.407788
[74]    valid_0's binary_logloss: 0.407738
[75]    valid_0's binary_logloss: 0.407693
[76]    valid_0's binary_logloss: 0.407594
[77]    valid_0's binary_logloss: 0.407524
[78]    valid_0's binary_logloss: 0.407483
[79]    valid_0's binary_logloss: 0.407429
[80]    valid_0's binary_logloss: 0.407378
[81]    valid_0's binary_logloss: 0.407317
[82]    valid_0's binary_logloss: 0.407204
[83]    valid_0's binary_logloss: 0.40715
[84]    valid_0's binary_logloss: 0.407127
[85]    valid_0's binary_logloss: 0.407075
[86]    valid_0's binary_logloss: 0.406974
[87]    valid_0's binary_logloss: 0.40693
[88]    valid_0's binary_logloss: 0.406859
[89]    valid_0's binary_logloss: 0.406789
[90]    valid_0's binary_logloss: 0.40672
[91]    valid_0's binary_logloss: 0.406656
[92]    valid_0's binary_logloss: 0.406607
[93]    valid_0's binary_logloss: 0.406553
[94]    valid_0's binary_logloss: 0.406517
[95]    valid_0's binary_logloss: 0.40647
[96]    valid_0's binary_logloss: 0.406406
[97]    valid_0's binary_logloss: 0.406377
[98]    valid_0's binary_logloss: 0.406318
[99]    valid_0's binary_logloss: 0.406296
[100]   valid_0's binary_logloss: 0.406237
[101]   valid_0's binary_logloss: 0.40621
[102]   valid_0's binary_logloss: 0.406155
[103]   valid_0's binary_logloss: 0.406139
[104]   valid_0's binary_logloss: 0.406094
[105]   valid_0's binary_logloss: 0.40607
[106]   valid_0's binary_logloss: 0.406029
[107]   valid_0's binary_logloss: 0.405948
[108]   valid_0's binary_logloss: 0.405921
[109]   valid_0's binary_logloss: 0.405863
[110]   valid_0's binary_logloss: 0.405823
[111]   valid_0's binary_logloss: 0.405781
[112]   valid_0's binary_logloss: 0.405746
[113]   valid_0's binary_logloss: 0.405717
[114]   valid_0's binary_logloss: 0.405684
[115]   valid_0's binary_logloss: 0.405612
[116]   valid_0's binary_logloss: 0.405535
[117]   valid_0's binary_logloss: 0.40546
[118]   valid_0's binary_logloss: 0.405442
[119]   valid_0's binary_logloss: 0.405408
[120]   valid_0's binary_logloss: 0.405339
```

```
[121]   valid_0's binary_logloss: 0.40532
[122]   valid_0's binary_logloss: 0.4053
[123]   valid_0's binary_logloss: 0.405262
[124]   valid_0's binary_logloss: 0.405223
[125]   valid_0's binary_logloss: 0.4052
[126]   valid_0's binary_logloss: 0.405165
[127]   valid_0's binary_logloss: 0.405137
[128]   valid_0's binary_logloss: 0.405105
[129]   valid_0's binary_logloss: 0.405075
[130]   valid_0's binary_logloss: 0.405055
[131]   valid_0's binary_logloss: 0.405028
[132]   valid_0's binary_logloss: 0.405017
[133]   valid_0's binary_logloss: 0.404996
[134]   valid_0's binary_logloss: 0.404969
[135]   valid_0's binary_logloss: 0.404947
[136]   valid_0's binary_logloss: 0.404885
[137]   valid_0's binary_logloss: 0.404859
[138]   valid_0's binary_logloss: 0.404795
[139]   valid_0's binary_logloss: 0.40474
[140]   valid_0's binary_logloss: 0.404716
[141]   valid_0's binary_logloss: 0.404673
[142]   valid_0's binary_logloss: 0.404649
[143]   valid_0's binary_logloss: 0.404635
[144]   valid_0's binary_logloss: 0.40461
[145]   valid_0's binary_logloss: 0.404595
[146]   valid_0's binary_logloss: 0.404596
[147]   valid_0's binary_logloss: 0.404579
[148]   valid_0's binary_logloss: 0.404561
[149]   valid_0's binary_logloss: 0.404523
[150]   valid_0's binary_logloss: 0.404512
[151]   valid_0's binary_logloss: 0.404475
[152]   valid_0's binary_logloss: 0.40445
[153]   valid_0's binary_logloss: 0.404422
[154]   valid_0's binary_logloss: 0.404405
[155]   valid_0's binary_logloss: 0.404382
[156]   valid_0's binary_logloss: 0.404362
[157]   valid_0's binary_logloss: 0.40432
[158]   valid_0's binary_logloss: 0.404293
[159]   valid_0's binary_logloss: 0.404284
[160]   valid_0's binary_logloss: 0.404264
[161]   valid_0's binary_logloss: 0.404262
[162]   valid_0's binary_logloss: 0.40425
[163]   valid_0's binary_logloss: 0.404238
[164]   valid_0's binary_logloss: 0.404231
[165]   valid_0's binary_logloss: 0.404205
[166]   valid_0's binary_logloss: 0.404158
[167]   valid_0's binary_logloss: 0.404137
[168]   valid_0's binary_logloss: 0.404121
[169]   valid_0's binary_logloss: 0.404103
[170]   valid_0's binary_logloss: 0.404056
[171]   valid_0's binary_logloss: 0.403989
[172]   valid_0's binary_logloss: 0.403983
[173]   valid_0's binary_logloss: 0.403958
[174]   valid_0's binary_logloss: 0.403928
[175]   valid_0's binary_logloss: 0.403857
[176]   valid_0's binary_logloss: 0.403778
[177]   valid_0's binary_logloss: 0.403714
[178]   valid_0's binary_logloss: 0.403658
[179]   valid_0's binary_logloss: 0.403597
[180]   valid_0's binary_logloss: 0.403559
[181]   valid_0's binary_logloss: 0.403544
```

```
[182]   valid_0's binary_logloss: 0.40354
[183]   valid_0's binary_logloss: 0.403526
[184]   valid_0's binary_logloss: 0.403515
[185]   valid_0's binary_logloss: 0.403509
[186]   valid_0's binary_logloss: 0.403499
[187]   valid_0's binary_logloss: 0.40349
[188]   valid_0's binary_logloss: 0.403466
[189]   valid_0's binary_logloss: 0.403433
[190]   valid_0's binary_logloss: 0.403415
[191]   valid_0's binary_logloss: 0.403396
[192]   valid_0's binary_logloss: 0.403382
[193]   valid_0's binary_logloss: 0.403366
[194]   valid_0's binary_logloss: 0.403357
[195]   valid_0's binary_logloss: 0.403345
[196]   valid_0's binary_logloss: 0.403325
[197]   valid_0's binary_logloss: 0.403323
[198]   valid_0's binary_logloss: 0.40331
[199]   valid_0's binary_logloss: 0.403292
[200]   valid_0's binary_logloss: 0.403273
[201]   valid_0's binary_logloss: 0.403269
[202]   valid_0's binary_logloss: 0.40325
[203]   valid_0's binary_logloss: 0.403249
[204]   valid_0's binary_logloss: 0.403245
[205]   valid_0's binary_logloss: 0.403229
[206]   valid_0's binary_logloss: 0.403217
[207]   valid_0's binary_logloss: 0.403176
[208]   valid_0's binary_logloss: 0.403173
[209]   valid_0's binary_logloss: 0.403157
[210]   valid_0's binary_logloss: 0.403157
[211]   valid_0's binary_logloss: 0.403123
[212]   valid_0's binary_logloss: 0.403121
[213]   valid_0's binary_logloss: 0.403104
[214]   valid_0's binary_logloss: 0.403082
[215]   valid_0's binary_logloss: 0.403076
[216]   valid_0's binary_logloss: 0.403084
[217]   valid_0's binary_logloss: 0.403083
[218]   valid_0's binary_logloss: 0.403067
[219]   valid_0's binary_logloss: 0.403046
[220]   valid_0's binary_logloss: 0.403049
[221]   valid_0's binary_logloss: 0.403041
[222]   valid_0's binary_logloss: 0.403014
[223]   valid_0's binary_logloss: 0.403002
[224]   valid_0's binary_logloss: 0.402986
[225]   valid_0's binary_logloss: 0.402995
[226]   valid_0's binary_logloss: 0.402968
[227]   valid_0's binary_logloss: 0.402962
[228]   valid_0's binary_logloss: 0.402955
[229]   valid_0's binary_logloss: 0.402924
[230]   valid_0's binary_logloss: 0.402913
[231]   valid_0's binary_logloss: 0.402914
[232]   valid_0's binary_logloss: 0.4029
[233]   valid_0's binary_logloss: 0.402912
[234]   valid_0's binary_logloss: 0.402909
[235]   valid_0's binary_logloss: 0.402903
[236]   valid_0's binary_logloss: 0.402888
[237]   valid_0's binary_logloss: 0.40288
[238]   valid_0's binary_logloss: 0.402868
[239]   valid_0's binary_logloss: 0.402868
[240]   valid_0's binary_logloss: 0.402865
[241]   valid_0's binary_logloss: 0.402853
[242]   valid_0's binary_logloss: 0.402836
```

```
[243]   valid_0's binary_logloss: 0.402827
[244]   valid_0's binary_logloss: 0.402824
[245]   valid_0's binary_logloss: 0.402817
[246]   valid_0's binary_logloss: 0.402799
[247]   valid_0's binary_logloss: 0.402785
[248]   valid_0's binary_logloss: 0.402786
[249]   valid_0's binary_logloss: 0.402792
[250]   valid_0's binary_logloss: 0.402771
[251]   valid_0's binary_logloss: 0.40277
[252]   valid_0's binary_logloss: 0.402766
[253]   valid_0's binary_logloss: 0.402755
[254]   valid_0's binary_logloss: 0.402746
[255]   valid_0's binary_logloss: 0.40274
[256]   valid_0's binary_logloss: 0.402719
[257]   valid_0's binary_logloss: 0.402704
[258]   valid_0's binary_logloss: 0.40269
[259]   valid_0's binary_logloss: 0.402689
[260]   valid_0's binary_logloss: 0.40268
[261]   valid_0's binary_logloss: 0.402644
[262]   valid_0's binary_logloss: 0.402629
[263]   valid_0's binary_logloss: 0.40264
[264]   valid_0's binary_logloss: 0.402624
[265]   valid_0's binary_logloss: 0.40262
[266]   valid_0's binary_logloss: 0.402605
[267]   valid_0's binary_logloss: 0.402604
[268]   valid_0's binary_logloss: 0.402593
[269]   valid_0's binary_logloss: 0.402591
[270]   valid_0's binary_logloss: 0.402588
[271]   valid_0's binary_logloss: 0.402561
[272]   valid_0's binary_logloss: 0.402523
[273]   valid_0's binary_logloss: 0.402516
[274]   valid_0's binary_logloss: 0.402509
[275]   valid_0's binary_logloss: 0.402493
[276]   valid_0's binary_logloss: 0.402481
[277]   valid_0's binary_logloss: 0.402482
[278]   valid_0's binary_logloss: 0.402452
[279]   valid_0's binary_logloss: 0.402452
[280]   valid_0's binary_logloss: 0.40242
[281]   valid_0's binary_logloss: 0.402415
[282]   valid_0's binary_logloss: 0.402414
[283]   valid_0's binary_logloss: 0.402409
[284]   valid_0's binary_logloss: 0.402391
[285]   valid_0's binary_logloss: 0.402388
[286]   valid_0's binary_logloss: 0.40236
[287]   valid_0's binary_logloss: 0.402342
[288]   valid_0's binary_logloss: 0.40233
[289]   valid_0's binary_logloss: 0.402335
[290]   valid_0's binary_logloss: 0.402336
[291]   valid_0's binary_logloss: 0.402323
[292]   valid_0's binary_logloss: 0.402317
[293]   valid_0's binary_logloss: 0.402314
[294]   valid_0's binary_logloss: 0.402319
[295]   valid_0's binary_logloss: 0.40229
[296]   valid_0's binary_logloss: 0.402277
[297]   valid_0's binary_logloss: 0.402282
[298]   valid_0's binary_logloss: 0.402284
[299]   valid_0's binary_logloss: 0.402275
[300]   valid_0's binary_logloss: 0.40228
[301]   valid_0's binary_logloss: 0.402282
[302]   valid_0's binary_logloss: 0.402274
[303]   valid_0's binary_logloss: 0.402264
```

```
[304]   valid_0's binary_logloss: 0.40226
[305]   valid_0's binary_logloss: 0.402259
[306]   valid_0's binary_logloss: 0.402254
[307]   valid_0's binary_logloss: 0.402249
[308]   valid_0's binary_logloss: 0.402191
[309]   valid_0's binary_logloss: 0.402184
[310]   valid_0's binary_logloss: 0.402167
[311]   valid_0's binary_logloss: 0.402168
[312]   valid_0's binary_logloss: 0.402172
[313]   valid_0's binary_logloss: 0.40216
[314]   valid_0's binary_logloss: 0.402155
[315]   valid_0's binary_logloss: 0.40216
[316]   valid_0's binary_logloss: 0.402146
[317]   valid_0's binary_logloss: 0.402131
[318]   valid_0's binary_logloss: 0.40213
[319]   valid_0's binary_logloss: 0.402113
[320]   valid_0's binary_logloss: 0.402099
[321]   valid_0's binary_logloss: 0.402095
[322]   valid_0's binary_logloss: 0.402106
[323]   valid_0's binary_logloss: 0.402075
[324]   valid_0's binary_logloss: 0.402073
[325]   valid_0's binary_logloss: 0.402074
[326]   valid_0's binary_logloss: 0.402064
[327]   valid_0's binary_logloss: 0.402057
[328]   valid_0's binary_logloss: 0.402025
[329]   valid_0's binary_logloss: 0.402023
[330]   valid_0's binary_logloss: 0.402024
[331]   valid_0's binary_logloss: 0.402033
[332]   valid_0's binary_logloss: 0.402022
[333]   valid_0's binary_logloss: 0.402024
[334]   valid_0's binary_logloss: 0.402024
[335]   valid_0's binary_logloss: 0.402025
[336]   valid_0's binary_logloss: 0.402027
[337]   valid_0's binary_logloss: 0.402018
[338]   valid_0's binary_logloss: 0.402025
[339]   valid_0's binary_logloss: 0.402021
[340]   valid_0's binary_logloss: 0.40203
[341]   valid_0's binary_logloss: 0.402004
[342]   valid_0's binary_logloss: 0.402
[343]   valid_0's binary_logloss: 0.402006
[344]   valid_0's binary_logloss: 0.401999
[345]   valid_0's binary_logloss: 0.401963
[346]   valid_0's binary_logloss: 0.401949
[347]   valid_0's binary_logloss: 0.401942
[348]   valid_0's binary_logloss: 0.40193
[349]   valid_0's binary_logloss: 0.40192
[350]   valid_0's binary_logloss: 0.401916
[351]   valid_0's binary_logloss: 0.401894
[352]   valid_0's binary_logloss: 0.40188
[353]   valid_0's binary_logloss: 0.401885
[354]   valid_0's binary_logloss: 0.401883
[355]   valid_0's binary_logloss: 0.401887
[356]   valid_0's binary_logloss: 0.401889
[357]   valid_0's binary_logloss: 0.401878
[358]   valid_0's binary_logloss: 0.401869
[359]   valid_0's binary_logloss: 0.401823
[360]   valid_0's binary_logloss: 0.401826
[361]   valid_0's binary_logloss: 0.401817
[362]   valid_0's binary_logloss: 0.401787
[363]   valid_0's binary_logloss: 0.401787
[364]   valid_0's binary_logloss: 0.401785
```

```
[365]   valid_0's binary_logloss: 0.40179
[366]   valid_0's binary_logloss: 0.401774
[367]   valid_0's binary_logloss: 0.401765
[368]   valid_0's binary_logloss: 0.401757
[369]   valid_0's binary_logloss: 0.401765
[370]   valid_0's binary_logloss: 0.401761
[371]   valid_0's binary_logloss: 0.401749
[372]   valid_0's binary_logloss: 0.401721
[373]   valid_0's binary_logloss: 0.40172
[374]   valid_0's binary_logloss: 0.401704
[375]   valid_0's binary_logloss: 0.401708
[376]   valid_0's binary_logloss: 0.401692
[377]   valid_0's binary_logloss: 0.401691
[378]   valid_0's binary_logloss: 0.401682
[379]   valid_0's binary_logloss: 0.40169
[380]   valid_0's binary_logloss: 0.401673
[381]   valid_0's binary_logloss: 0.401654
[382]   valid_0's binary_logloss: 0.401649
[383]   valid_0's binary_logloss: 0.401637
[384]   valid_0's binary_logloss: 0.401623
[385]   valid_0's binary_logloss: 0.401613
[386]   valid_0's binary_logloss: 0.401608
[387]   valid_0's binary_logloss: 0.401612
[388]   valid_0's binary_logloss: 0.401607
[389]   valid_0's binary_logloss: 0.401605
[390]   valid_0's binary_logloss: 0.401603
[391]   valid_0's binary_logloss: 0.40159
[392]   valid_0's binary_logloss: 0.401578
[393]   valid_0's binary_logloss: 0.401567
[394]   valid_0's binary_logloss: 0.40155
[395]   valid_0's binary_logloss: 0.401545
[396]   valid_0's binary_logloss: 0.401536
[397]   valid_0's binary_logloss: 0.4015
[398]   valid_0's binary_logloss: 0.401485
[399]   valid_0's binary_logloss: 0.401477
[400]   valid_0's binary_logloss: 0.401479
[401]   valid_0's binary_logloss: 0.401467
[402]   valid_0's binary_logloss: 0.401462
[403]   valid_0's binary_logloss: 0.401457
[404]   valid_0's binary_logloss: 0.401449
[405]   valid_0's binary_logloss: 0.40144
[406]   valid_0's binary_logloss: 0.401436
[407]   valid_0's binary_logloss: 0.401431
[408]   valid_0's binary_logloss: 0.401436
[409]   valid_0's binary_logloss: 0.401432
[410]   valid_0's binary_logloss: 0.401427
[411]   valid_0's binary_logloss: 0.401407
[412]   valid_0's binary_logloss: 0.401406
[413]   valid_0's binary_logloss: 0.401393
[414]   valid_0's binary_logloss: 0.401392
[415]   valid_0's binary_logloss: 0.401398
[416]   valid_0's binary_logloss: 0.401387
[417]   valid_0's binary_logloss: 0.401372
[418]   valid_0's binary_logloss: 0.401364
[419]   valid_0's binary_logloss: 0.40133
[420]   valid_0's binary_logloss: 0.401328
[421]   valid_0's binary_logloss: 0.401323
[422]   valid_0's binary_logloss: 0.401312
[423]   valid_0's binary_logloss: 0.40131
[424]   valid_0's binary_logloss: 0.401305
[425]   valid_0's binary_logloss: 0.401297
```

```
[426]   valid_0's binary_logloss: 0.401294
[427]   valid_0's binary_logloss: 0.401293
[428]   valid_0's binary_logloss: 0.401283
[429]   valid_0's binary_logloss: 0.401274
[430]   valid_0's binary_logloss: 0.401273
[431]   valid_0's binary_logloss: 0.401247
[432]   valid_0's binary_logloss: 0.401229
[433]   valid_0's binary_logloss: 0.401232
[434]   valid_0's binary_logloss: 0.401235
[435]   valid_0's binary_logloss: 0.401227
[436]   valid_0's binary_logloss: 0.401221
[437]   valid_0's binary_logloss: 0.401218
[438]   valid_0's binary_logloss: 0.401215
[439]   valid_0's binary_logloss: 0.401218
[440]   valid_0's binary_logloss: 0.401206
[441]   valid_0's binary_logloss: 0.401189
[442]   valid_0's binary_logloss: 0.401176
[443]   valid_0's binary_logloss: 0.401155
[444]   valid_0's binary_logloss: 0.401148
[445]   valid_0's binary_logloss: 0.401143
[446]   valid_0's binary_logloss: 0.401133
[447]   valid_0's binary_logloss: 0.401122
[448]   valid_0's binary_logloss: 0.401121
[449]   valid_0's binary_logloss: 0.401126
[450]   valid_0's binary_logloss: 0.401129
[451]   valid_0's binary_logloss: 0.401129
[452]   valid_0's binary_logloss: 0.401114
[453]   valid_0's binary_logloss: 0.401117
[454]   valid_0's binary_logloss: 0.401126
[455]   valid_0's binary_logloss: 0.401121
[456]   valid_0's binary_logloss: 0.401117
[457]   valid_0's binary_logloss: 0.401104
[458]   valid_0's binary_logloss: 0.401104
[459]   valid_0's binary_logloss: 0.401102
[460]   valid_0's binary_logloss: 0.401098
[461]   valid_0's binary_logloss: 0.401095
[462]   valid_0's binary_logloss: 0.401092
[463]   valid_0's binary_logloss: 0.401089
[464]   valid_0's binary_logloss: 0.401086
[465]   valid_0's binary_logloss: 0.401082
[466]   valid_0's binary_logloss: 0.401075
[467]   valid_0's binary_logloss: 0.401072
[468]   valid_0's binary_logloss: 0.401062
[469]   valid_0's binary_logloss: 0.401058
[470]   valid_0's binary_logloss: 0.401054
[471]   valid_0's binary_logloss: 0.401056
[472]   valid_0's binary_logloss: 0.401059
[473]   valid_0's binary_logloss: 0.401067
[474]   valid_0's binary_logloss: 0.401072
[475]   valid_0's binary_logloss: 0.401073
[476]   valid_0's binary_logloss: 0.401069
[477]   valid_0's binary_logloss: 0.401069
[478]   valid_0's binary_logloss: 0.401065
[479]   valid_0's binary_logloss: 0.401055
[480]   valid_0's binary_logloss: 0.401028
[481]   valid_0's binary_logloss: 0.401031
[482]   valid_0's binary_logloss: 0.40102
[483]   valid_0's binary_logloss: 0.401004
[484]   valid_0's binary_logloss: 0.400989
[485]   valid_0's binary_logloss: 0.400993
[486]   valid_0's binary_logloss: 0.400986
```

```
[487]   valid_0's binary_logloss: 0.400987
[488]   valid_0's binary_logloss: 0.40098
[489]   valid_0's binary_logloss: 0.400974
[490]   valid_0's binary_logloss: 0.400963
[491]   valid_0's binary_logloss: 0.40096
[492]   valid_0's binary_logloss: 0.400955
[493]   valid_0's binary_logloss: 0.400951
[494]   valid_0's binary_logloss: 0.400949
[495]   valid_0's binary_logloss: 0.40095
[496]   valid_0's binary_logloss: 0.400968
[497]   valid_0's binary_logloss: 0.400973
[498]   valid_0's binary_logloss: 0.400983
[499]   valid_0's binary_logloss: 0.400968
[500]   valid_0's binary_logloss: 0.400976
[501]   valid_0's binary_logloss: 0.40097
[502]   valid_0's binary_logloss: 0.40097
[503]   valid_0's binary_logloss: 0.400963
[504]   valid_0's binary_logloss: 0.400961
[505]   valid_0's binary_logloss: 0.400959
[506]   valid_0's binary_logloss: 0.400966
[507]   valid_0's binary_logloss: 0.40096
[508]   valid_0's binary_logloss: 0.400963
[509]   valid_0's binary_logloss: 0.400967
[510]   valid_0's binary_logloss: 0.400973
[511]   valid_0's binary_logloss: 0.400971
[512]   valid_0's binary_logloss: 0.40097
[513]   valid_0's binary_logloss: 0.400965
[514]   valid_0's binary_logloss: 0.400968
[515]   valid_0's binary_logloss: 0.400941
[516]   valid_0's binary_logloss: 0.400931
[517]   valid_0's binary_logloss: 0.400929
[518]   valid_0's binary_logloss: 0.400918
[519]   valid_0's binary_logloss: 0.4009
[520]   valid_0's binary_logloss: 0.400894
[521]   valid_0's binary_logloss: 0.40089
[522]   valid_0's binary_logloss: 0.400892
[523]   valid_0's binary_logloss: 0.400876
[524]   valid_0's binary_logloss: 0.400885
[525]   valid_0's binary_logloss: 0.400892
[526]   valid_0's binary_logloss: 0.400884
[527]   valid_0's binary_logloss: 0.400894
[528]   valid_0's binary_logloss: 0.400896
[529]   valid_0's binary_logloss: 0.400889
[530]   valid_0's binary_logloss: 0.400885
[531]   valid_0's binary_logloss: 0.400888
[532]   valid_0's binary_logloss: 0.400884
[533]   valid_0's binary_logloss: 0.400877
[534]   valid_0's binary_logloss: 0.40088
[535]   valid_0's binary_logloss: 0.40088
[536]   valid_0's binary_logloss: 0.400888
[537]   valid_0's binary_logloss: 0.400884
[538]   valid_0's binary_logloss: 0.40088
[539]   valid_0's binary_logloss: 0.400883
[540]   valid_0's binary_logloss: 0.400862
[541]   valid_0's binary_logloss: 0.400861
[542]   valid_0's binary_logloss: 0.400856
[543]   valid_0's binary_logloss: 0.400865
[544]   valid_0's binary_logloss: 0.400872
[545]   valid_0's binary_logloss: 0.400868
[546]   valid_0's binary_logloss: 0.400855
[547]   valid_0's binary_logloss: 0.400846
```

```
[548]   valid_0's binary_logloss: 0.400848
[549]   valid_0's binary_logloss: 0.400849
[550]   valid_0's binary_logloss: 0.400853
[551]   valid_0's binary_logloss: 0.400842
[552]   valid_0's binary_logloss: 0.400842
[553]   valid_0's binary_logloss: 0.400839
[554]   valid_0's binary_logloss: 0.400836
[555]   valid_0's binary_logloss: 0.400818
[556]   valid_0's binary_logloss: 0.400818
[557]   valid_0's binary_logloss: 0.400814
[558]   valid_0's binary_logloss: 0.400807
[559]   valid_0's binary_logloss: 0.400809
[560]   valid_0's binary_logloss: 0.400807
[561]   valid_0's binary_logloss: 0.400802
[562]   valid_0's binary_logloss: 0.400796
[563]   valid_0's binary_logloss: 0.400794
[564]   valid_0's binary_logloss: 0.400786
[565]   valid_0's binary_logloss: 0.400782
[566]   valid_0's binary_logloss: 0.400776
[567]   valid_0's binary_logloss: 0.400773
[568]   valid_0's binary_logloss: 0.400774
[569]   valid_0's binary_logloss: 0.400762
[570]   valid_0's binary_logloss: 0.400768
[571]   valid_0's binary_logloss: 0.400758
[572]   valid_0's binary_logloss: 0.400757
[573]   valid_0's binary_logloss: 0.400767
[574]   valid_0's binary_logloss: 0.400753
[575]   valid_0's binary_logloss: 0.400753
[576]   valid_0's binary_logloss: 0.400744
[577]   valid_0's binary_logloss: 0.400741
[578]   valid_0's binary_logloss: 0.400725
[579]   valid_0's binary_logloss: 0.400723
[580]   valid_0's binary_logloss: 0.400724
[581]   valid_0's binary_logloss: 0.400722
[582]   valid_0's binary_logloss: 0.400725
[583]   valid_0's binary_logloss: 0.400719
[584]   valid_0's binary_logloss: 0.400717
[585]   valid_0's binary_logloss: 0.400714
[586]   valid_0's binary_logloss: 0.4007
[587]   valid_0's binary_logloss: 0.400704
[588]   valid_0's binary_logloss: 0.400706
[589]   valid_0's binary_logloss: 0.400697
[590]   valid_0's binary_logloss: 0.400689
[591]   valid_0's binary_logloss: 0.400673
[592]   valid_0's binary_logloss: 0.40066
[593]   valid_0's binary_logloss: 0.400647
[594]   valid_0's binary_logloss: 0.400647
[595]   valid_0's binary_logloss: 0.400642
[596]   valid_0's binary_logloss: 0.400647
[597]   valid_0's binary_logloss: 0.400639
[598]   valid_0's binary_logloss: 0.400632
[599]   valid_0's binary_logloss: 0.400643
[600]   valid_0's binary_logloss: 0.400625
[601]   valid_0's binary_logloss: 0.400614
[602]   valid_0's binary_logloss: 0.400606
[603]   valid_0's binary_logloss: 0.400611
[604]   valid_0's binary_logloss: 0.400622
[605]   valid_0's binary_logloss: 0.400628
[606]   valid_0's binary_logloss: 0.400621
[607]   valid_0's binary_logloss: 0.400616
[608]   valid_0's binary_logloss: 0.400612
```

```
[609]   valid_0's binary_logloss: 0.400612
[610]   valid_0's binary_logloss: 0.400614
[611]   valid_0's binary_logloss: 0.40061
[612]   valid_0's binary_logloss: 0.40061
[613]   valid_0's binary_logloss: 0.400614
[614]   valid_0's binary_logloss: 0.400603
[615]   valid_0's binary_logloss: 0.4006
[616]   valid_0's binary_logloss: 0.400601
[617]   valid_0's binary_logloss: 0.400603
[618]   valid_0's binary_logloss: 0.400607
[619]   valid_0's binary_logloss: 0.40061
[620]   valid_0's binary_logloss: 0.400615
[621]   valid_0's binary_logloss: 0.4006
[622]   valid_0's binary_logloss: 0.400597
[623]   valid_0's binary_logloss: 0.400601
[624]   valid_0's binary_logloss: 0.400591
[625]   valid_0's binary_logloss: 0.400595
[626]   valid_0's binary_logloss: 0.400594
[627]   valid_0's binary_logloss: 0.40059
[628]   valid_0's binary_logloss: 0.400588
[629]   valid_0's binary_logloss: 0.400581
[630]   valid_0's binary_logloss: 0.400583
[631]   valid_0's binary_logloss: 0.400592
[632]   valid_0's binary_logloss: 0.400599
[633]   valid_0's binary_logloss: 0.400591
[634]   valid_0's binary_logloss: 0.40059
[635]   valid_0's binary_logloss: 0.400579
[636]   valid_0's binary_logloss: 0.400586
[637]   valid_0's binary_logloss: 0.400589
[638]   valid_0's binary_logloss: 0.400602
[639]   valid_0's binary_logloss: 0.400594
[640]   valid_0's binary_logloss: 0.400609
[641]   valid_0's binary_logloss: 0.400607
[642]   valid_0's binary_logloss: 0.400606
[643]   valid_0's binary_logloss: 0.400602
[644]   valid_0's binary_logloss: 0.400604
[645]   valid_0's binary_logloss: 0.40061
[646]   valid_0's binary_logloss: 0.400603
[647]   valid_0's binary_logloss: 0.40059
[648]   valid_0's binary_logloss: 0.400596
[649]   valid_0's binary_logloss: 0.400597
[650]   valid_0's binary_logloss: 0.400603
[651]   valid_0's binary_logloss: 0.400593
[652]   valid_0's binary_logloss: 0.400587
[653]   valid_0's binary_logloss: 0.400582
[654]   valid_0's binary_logloss: 0.400579
[655]   valid_0's binary_logloss: 0.40058
[656]   valid_0's binary_logloss: 0.400569
[657]   valid_0's binary_logloss: 0.400563
[658]   valid_0's binary_logloss: 0.400562
[659]   valid_0's binary_logloss: 0.400571
[660]   valid_0's binary_logloss: 0.400574
[661]   valid_0's binary_logloss: 0.400567
[662]   valid_0's binary_logloss: 0.400563
[663]   valid_0's binary_logloss: 0.400569
[664]   valid_0's binary_logloss: 0.400581
[665]   valid_0's binary_logloss: 0.400581
[666]   valid_0's binary_logloss: 0.400577
[667]   valid_0's binary_logloss: 0.400584
[668]   valid_0's binary_logloss: 0.400579
[669]   valid_0's binary_logloss: 0.400577
```

```
[670]   valid_0's binary_logloss: 0.40057
[671]   valid_0's binary_logloss: 0.400561
[672]   valid_0's binary_logloss: 0.40055
[673]   valid_0's binary_logloss: 0.400536
[674]   valid_0's binary_logloss: 0.400539
[675]   valid_0's binary_logloss: 0.400539
[676]   valid_0's binary_logloss: 0.400538
[677]   valid_0's binary_logloss: 0.400539
[678]   valid_0's binary_logloss: 0.400536
[679]   valid_0's binary_logloss: 0.400523
[680]   valid_0's binary_logloss: 0.400529
[681]   valid_0's binary_logloss: 0.400532
[682]   valid_0's binary_logloss: 0.400515
[683]   valid_0's binary_logloss: 0.400492
[684]   valid_0's binary_logloss: 0.400497
[685]   valid_0's binary_logloss: 0.400482
[686]   valid_0's binary_logloss: 0.400473
[687]   valid_0's binary_logloss: 0.400467
[688]   valid_0's binary_logloss: 0.40047
[689]   valid_0's binary_logloss: 0.400476
[690]   valid_0's binary_logloss: 0.400467
[691]   valid_0's binary_logloss: 0.400463
[692]   valid_0's binary_logloss: 0.400459
[693]   valid_0's binary_logloss: 0.400457
[694]   valid_0's binary_logloss: 0.400455
[695]   valid_0's binary_logloss: 0.400457
[696]   valid_0's binary_logloss: 0.400448
[697]   valid_0's binary_logloss: 0.400455
[698]   valid_0's binary_logloss: 0.400449
[699]   valid_0's binary_logloss: 0.400458
[700]   valid_0's binary_logloss: 0.400459
[701]   valid_0's binary_logloss: 0.400443
[702]   valid_0's binary_logloss: 0.400439
[703]   valid_0's binary_logloss: 0.400444
[704]   valid_0's binary_logloss: 0.400451
[705]   valid_0's binary_logloss: 0.400441
[706]   valid_0's binary_logloss: 0.400438
[707]   valid_0's binary_logloss: 0.400441
[708]   valid_0's binary_logloss: 0.400452
[709]   valid_0's binary_logloss: 0.400449
[710]   valid_0's binary_logloss: 0.400454
[711]   valid_0's binary_logloss: 0.400453
[712]   valid_0's binary_logloss: 0.400462
[713]   valid_0's binary_logloss: 0.400458
[714]   valid_0's binary_logloss: 0.400456
[715]   valid_0's binary_logloss: 0.400468
[716]   valid_0's binary_logloss: 0.400462
[717]   valid_0's binary_logloss: 0.400455
[718]   valid_0's binary_logloss: 0.40045
[719]   valid_0's binary_logloss: 0.400451
[720]   valid_0's binary_logloss: 0.400441
[721]   valid_0's binary_logloss: 0.400443
[722]   valid_0's binary_logloss: 0.400447
[723]   valid_0's binary_logloss: 0.400437
[724]   valid_0's binary_logloss: 0.40044
[725]   valid_0's binary_logloss: 0.400447
[726]   valid_0's binary_logloss: 0.400444
[727]   valid_0's binary_logloss: 0.400445
[728]   valid_0's binary_logloss: 0.400433
[729]   valid_0's binary_logloss: 0.400432
[730]   valid_0's binary_logloss: 0.40042
```

```
[731]   valid_0's binary_logloss: 0.400413
[732]   valid_0's binary_logloss: 0.400411
[733]   valid_0's binary_logloss: 0.400411
[734]   valid_0's binary_logloss: 0.400409
[735]   valid_0's binary_logloss: 0.400409
[736]   valid_0's binary_logloss: 0.400406
[737]   valid_0's binary_logloss: 0.400392
[738]   valid_0's binary_logloss: 0.40039
[739]   valid_0's binary_logloss: 0.400393
[740]   valid_0's binary_logloss: 0.4004
[741]   valid_0's binary_logloss: 0.4004
[742]   valid_0's binary_logloss: 0.400403
[743]   valid_0's binary_logloss: 0.400405
[744]   valid_0's binary_logloss: 0.400403
[745]   valid_0's binary_logloss: 0.400411
[746]   valid_0's binary_logloss: 0.400405
[747]   valid_0's binary_logloss: 0.400401
[748]   valid_0's binary_logloss: 0.400398
[749]   valid_0's binary_logloss: 0.400392
[750]   valid_0's binary_logloss: 0.400397
[751]   valid_0's binary_logloss: 0.400394
[752]   valid_0's binary_logloss: 0.400391
[753]   valid_0's binary_logloss: 0.400399
[754]   valid_0's binary_logloss: 0.400393
[755]   valid_0's binary_logloss: 0.400385
[756]   valid_0's binary_logloss: 0.400384
[757]   valid_0's binary_logloss: 0.400374
[758]   valid_0's binary_logloss: 0.400374
[759]   valid_0's binary_logloss: 0.400372
[760]   valid_0's binary_logloss: 0.400371
[761]   valid_0's binary_logloss: 0.40037
[762]   valid_0's binary_logloss: 0.400373
[763]   valid_0's binary_logloss: 0.400374
[764]   valid_0's binary_logloss: 0.400376
[765]   valid_0's binary_logloss: 0.400383
[766]   valid_0's binary_logloss: 0.400381
[767]   valid_0's binary_logloss: 0.400385
[768]   valid_0's binary_logloss: 0.400381
[769]   valid_0's binary_logloss: 0.400379
[770]   valid_0's binary_logloss: 0.400381
[771]   valid_0's binary_logloss: 0.400384
[772]   valid_0's binary_logloss: 0.400376
[773]   valid_0's binary_logloss: 0.40037
[774]   valid_0's binary_logloss: 0.400366
[775]   valid_0's binary_logloss: 0.40037
[776]   valid_0's binary_logloss: 0.400364
[777]   valid_0's binary_logloss: 0.400365
[778]   valid_0's binary_logloss: 0.400363
[779]   valid_0's binary_logloss: 0.400361
[780]   valid_0's binary_logloss: 0.400367
[781]   valid_0's binary_logloss: 0.400354
[782]   valid_0's binary_logloss: 0.400355
[783]   valid_0's binary_logloss: 0.400359
[784]   valid_0's binary_logloss: 0.400348
[785]   valid_0's binary_logloss: 0.400353
[786]   valid_0's binary_logloss: 0.400355
[787]   valid_0's binary_logloss: 0.400355
[788]   valid_0's binary_logloss: 0.40035
[789]   valid_0's binary_logloss: 0.400341
[790]   valid_0's binary_logloss: 0.400343
[791]   valid_0's binary_logloss: 0.400348
```

```
[792]   valid_0's binary_logloss: 0.400352
[793]   valid_0's binary_logloss: 0.400348
[794]   valid_0's binary_logloss: 0.400349
[795]   valid_0's binary_logloss: 0.400346
[796]   valid_0's binary_logloss: 0.400341
[797]   valid_0's binary_logloss: 0.400341
[798]   valid_0's binary_logloss: 0.40034
[799]   valid_0's binary_logloss: 0.400331
[800]   valid_0's binary_logloss: 0.400334
[801]   valid_0's binary_logloss: 0.400332
[802]   valid_0's binary_logloss: 0.400327
[803]   valid_0's binary_logloss: 0.400319
[804]   valid_0's binary_logloss: 0.400318
[805]   valid_0's binary_logloss: 0.400326
[806]   valid_0's binary_logloss: 0.40032
[807]   valid_0's binary_logloss: 0.400309
[808]   valid_0's binary_logloss: 0.400304
[809]   valid_0's binary_logloss: 0.4003
[810]   valid_0's binary_logloss: 0.400303
[811]   valid_0's binary_logloss: 0.400292
[812]   valid_0's binary_logloss: 0.400287
[813]   valid_0's binary_logloss: 0.400277
[814]   valid_0's binary_logloss: 0.400274
[815]   valid_0's binary_logloss: 0.400267
[816]   valid_0's binary_logloss: 0.400269
[817]   valid_0's binary_logloss: 0.400278
[818]   valid_0's binary_logloss: 0.400282
[819]   valid_0's binary_logloss: 0.400282
[820]   valid_0's binary_logloss: 0.400277
[821]   valid_0's binary_logloss: 0.400269
[822]   valid_0's binary_logloss: 0.400278
[823]   valid_0's binary_logloss: 0.400271
[824]   valid_0's binary_logloss: 0.40027
[825]   valid_0's binary_logloss: 0.400282
[826]   valid_0's binary_logloss: 0.400276
[827]   valid_0's binary_logloss: 0.400276
[828]   valid_0's binary_logloss: 0.400267
[829]   valid_0's binary_logloss: 0.400268
[830]   valid_0's binary_logloss: 0.400261
[831]   valid_0's binary_logloss: 0.400257
[832]   valid_0's binary_logloss: 0.400261
[833]   valid_0's binary_logloss: 0.400247
[834]   valid_0's binary_logloss: 0.400253
[835]   valid_0's binary_logloss: 0.400249
[836]   valid_0's binary_logloss: 0.400235
[837]   valid_0's binary_logloss: 0.400239
[838]   valid_0's binary_logloss: 0.400233
[839]   valid_0's binary_logloss: 0.400234
[840]   valid_0's binary_logloss: 0.400234
[841]   valid_0's binary_logloss: 0.400241
[842]   valid_0's binary_logloss: 0.400236
[843]   valid_0's binary_logloss: 0.400235
[844]   valid_0's binary_logloss: 0.400244
[845]   valid_0's binary_logloss: 0.400246
[846]   valid_0's binary_logloss: 0.400247
[847]   valid_0's binary_logloss: 0.400245
[848]   valid_0's binary_logloss: 0.400227
[849]   valid_0's binary_logloss: 0.400223
[850]   valid_0's binary_logloss: 0.400221
[851]   valid_0's binary_logloss: 0.400226
[852]   valid_0's binary_logloss: 0.400231
```

```
[853]   valid_0's binary_logloss: 0.400233
[854]   valid_0's binary_logloss: 0.400241
[855]   valid_0's binary_logloss: 0.400254
[856]   valid_0's binary_logloss: 0.400246
[857]   valid_0's binary_logloss: 0.400246
[858]   valid_0's binary_logloss: 0.400257
[859]   valid_0's binary_logloss: 0.400258
[860]   valid_0's binary_logloss: 0.400257
[861]   valid_0's binary_logloss: 0.400253
[862]   valid_0's binary_logloss: 0.400258
[863]   valid_0's binary_logloss: 0.400261
[864]   valid_0's binary_logloss: 0.400263
[865]   valid_0's binary_logloss: 0.400267
[866]   valid_0's binary_logloss: 0.400259
[867]   valid_0's binary_logloss: 0.400261
[868]   valid_0's binary_logloss: 0.400263
[869]   valid_0's binary_logloss: 0.400264
[870]   valid_0's binary_logloss: 0.400279
[871]   valid_0's binary_logloss: 0.400268
[872]   valid_0's binary_logloss: 0.400276
[873]   valid_0's binary_logloss: 0.400273
[874]   valid_0's binary_logloss: 0.400273
[875]   valid_0's binary_logloss: 0.400277
[876]   valid_0's binary_logloss: 0.400279
[877]   valid_0's binary_logloss: 0.400279
[878]   valid_0's binary_logloss: 0.400282
[879]   valid_0's binary_logloss: 0.400286
[880]   valid_0's binary_logloss: 0.400287
[881]   valid_0's binary_logloss: 0.400286
[882]   valid_0's binary_logloss: 0.400294
[883]   valid_0's binary_logloss: 0.400298
[884]   valid_0's binary_logloss: 0.4003
[885]   valid_0's binary_logloss: 0.400307
[886]   valid_0's binary_logloss: 0.400304
[887]   valid_0's binary_logloss: 0.4003
[888]   valid_0's binary_logloss: 0.400303
[889]   valid_0's binary_logloss: 0.400302
[890]   valid_0's binary_logloss: 0.4003
[891]   valid_0's binary_logloss: 0.400298
[892]   valid_0's binary_logloss: 0.400297
[893]   valid_0's binary_logloss: 0.400299
[894]   valid_0's binary_logloss: 0.400293
[895]   valid_0's binary_logloss: 0.400289
[896]   valid_0's binary_logloss: 0.400287
[897]   valid_0's binary_logloss: 0.400285
[898]   valid_0's binary_logloss: 0.400284
[899]   valid_0's binary_logloss: 0.400273
[900]   valid_0's binary_logloss: 0.400271
[901]   valid_0's binary_logloss: 0.400264
[902]   valid_0's binary_logloss: 0.400261
[903]   valid_0's binary_logloss: 0.400263
[904]   valid_0's binary_logloss: 0.400265
[905]   valid_0's binary_logloss: 0.400262
[906]   valid_0's binary_logloss: 0.400258
[907]   valid_0's binary_logloss: 0.400261
[908]   valid_0's binary_logloss: 0.400258
[909]   valid_0's binary_logloss: 0.400252
[910]   valid_0's binary_logloss: 0.400248
[911]   valid_0's binary_logloss: 0.400248
[912]   valid_0's binary_logloss: 0.40025
[913]   valid_0's binary_logloss: 0.40026
```

```
[914]   valid_0's binary_logloss: 0.400266
[915]   valid_0's binary_logloss: 0.400268
[916]   valid_0's binary_logloss: 0.400265
[917]   valid_0's binary_logloss: 0.400261
[918]   valid_0's binary_logloss: 0.40026
[919]   valid_0's binary_logloss: 0.400275
[920]   valid_0's binary_logloss: 0.400278
[921]   valid_0's binary_logloss: 0.400271
[922]   valid_0's binary_logloss: 0.400263
[923]   valid_0's binary_logloss: 0.400267
[924]   valid_0's binary_logloss: 0.40027
[925]   valid_0's binary_logloss: 0.400274
[926]   valid_0's binary_logloss: 0.400272
[927]   valid_0's binary_logloss: 0.400277
[928]   valid_0's binary_logloss: 0.40028
[929]   valid_0's binary_logloss: 0.400282
[930]   valid_0's binary_logloss: 0.400274
[931]   valid_0's binary_logloss: 0.400253
[932]   valid_0's binary_logloss: 0.400256
[933]   valid_0's binary_logloss: 0.400249
[934]   valid_0's binary_logloss: 0.400235
[935]   valid_0's binary_logloss: 0.40023
[936]   valid_0's binary_logloss: 0.400227
[937]   valid_0's binary_logloss: 0.400223
[938]   valid_0's binary_logloss: 0.400226
[939]   valid_0's binary_logloss: 0.400226
[940]   valid_0's binary_logloss: 0.400223
[941]   valid_0's binary_logloss: 0.400214
[942]   valid_0's binary_logloss: 0.400215
[943]   valid_0's binary_logloss: 0.400226
[944]   valid_0's binary_logloss: 0.400227
[945]   valid_0's binary_logloss: 0.400227
[946]   valid_0's binary_logloss: 0.400225
[947]   valid_0's binary_logloss: 0.40022
[948]   valid_0's binary_logloss: 0.400221
[949]   valid_0's binary_logloss: 0.40021
[950]   valid_0's binary_logloss: 0.400213
[951]   valid_0's binary_logloss: 0.400204
[952]   valid_0's binary_logloss: 0.400201
[953]   valid_0's binary_logloss: 0.400202
[954]   valid_0's binary_logloss: 0.400201
[955]   valid_0's binary_logloss: 0.400198
[956]   valid_0's binary_logloss: 0.400198
[957]   valid_0's binary_logloss: 0.400196
[958]   valid_0's binary_logloss: 0.400196
[959]   valid_0's binary_logloss: 0.400199
[960]   valid_0's binary_logloss: 0.400204
[961]   valid_0's binary_logloss: 0.400199
[962]   valid_0's binary_logloss: 0.400199
[963]   valid_0's binary_logloss: 0.400201
[964]   valid_0's binary_logloss: 0.400211
[965]   valid_0's binary_logloss: 0.400213
[966]   valid_0's binary_logloss: 0.400223
[967]   valid_0's binary_logloss: 0.400232
[968]   valid_0's binary_logloss: 0.400233
[969]   valid_0's binary_logloss: 0.400231
[970]   valid_0's binary_logloss: 0.400241
[971]   valid_0's binary_logloss: 0.400237
[972]   valid_0's binary_logloss: 0.400237
[973]   valid_0's binary_logloss: 0.400242
[974]   valid_0's binary_logloss: 0.400246
```

```
[975]   valid_0's binary_logloss: 0.400252
[976]   valid_0's binary_logloss: 0.400248
[977]   valid_0's binary_logloss: 0.400251
[978]   valid_0's binary_logloss: 0.400257
[979]   valid_0's binary_logloss: 0.40025
[980]   valid_0's binary_logloss: 0.40024
[981]   valid_0's binary_logloss: 0.400245
[982]   valid_0's binary_logloss: 0.400247
[983]   valid_0's binary_logloss: 0.400254
[984]   valid_0's binary_logloss: 0.400248
[985]   valid_0's binary_logloss: 0.400251
[986]   valid_0's binary_logloss: 0.400251
[987]   valid_0's binary_logloss: 0.400254
[988]   valid_0's binary_logloss: 0.400249
[989]   valid_0's binary_logloss: 0.400257
[990]   valid_0's binary_logloss: 0.400255
[991]   valid_0's binary_logloss: 0.400251
[992]   valid_0's binary_logloss: 0.400246
[993]   valid_0's binary_logloss: 0.400234
[994]   valid_0's binary_logloss: 0.400222
[995]   valid_0's binary_logloss: 0.400213
[996]   valid_0's binary_logloss: 0.400217
[997]   valid_0's binary_logloss: 0.40021
[998]   valid_0's binary_logloss: 0.40021
[999]   valid_0's binary_logloss: 0.40021
[1000]  valid_0's binary_logloss: 0.400221
[1001]  valid_0's binary_logloss: 0.400215
[1002]  valid_0's binary_logloss: 0.40022
[1003]  valid_0's binary_logloss: 0.40023
[1004]  valid_0's binary_logloss: 0.400232
[1005]  valid_0's binary_logloss: 0.400241
[1006]  valid_0's binary_logloss: 0.400239
[1007]  valid_0's binary_logloss: 0.400231
[1008]  valid_0's binary_logloss: 0.400228
[1009]  valid_0's binary_logloss: 0.400232
[1010]  valid_0's binary_logloss: 0.400231
[1011]  valid_0's binary_logloss: 0.400233
[1012]  valid_0's binary_logloss: 0.400225
[1013]  valid_0's binary_logloss: 0.400214
[1014]  valid_0's binary_logloss: 0.400213
[1015]  valid_0's binary_logloss: 0.40021
[1016]  valid_0's binary_logloss: 0.400219
[1017]  valid_0's binary_logloss: 0.400227
[1018]  valid_0's binary_logloss: 0.400239
[1019]  valid_0's binary_logloss: 0.400248
[1020]  valid_0's binary_logloss: 0.400258
[1021]  valid_0's binary_logloss: 0.400274
[1022]  valid_0's binary_logloss: 0.400282
[1023]  valid_0's binary_logloss: 0.400281
[1024]  valid_0's binary_logloss: 0.400282
[1025]  valid_0's binary_logloss: 0.400286
[1026]  valid_0's binary_logloss: 0.400281
[1027]  valid_0's binary_logloss: 0.400289
[1028]  valid_0's binary_logloss: 0.400287
[1029]  valid_0's binary_logloss: 0.400283
[1030]  valid_0's binary_logloss: 0.400282
[1031]  valid_0's binary_logloss: 0.400274
[1032]  valid_0's binary_logloss: 0.400274
[1033]  valid_0's binary_logloss: 0.400272
[1034]  valid_0's binary_logloss: 0.400279
[1035]  valid_0's binary_logloss: 0.400273
```

```
[1036]  valid_0's binary_logloss: 0.400275
[1037]  valid_0's binary_logloss: 0.400284
[1038]  valid_0's binary_logloss: 0.40029
[1039]  valid_0's binary_logloss: 0.400295
[1040]  valid_0's binary_logloss: 0.400299
[1041]  valid_0's binary_logloss: 0.40031
[1042]  valid_0's binary_logloss: 0.400314
[1043]  valid_0's binary_logloss: 0.400298
[1044]  valid_0's binary_logloss: 0.40029
[1045]  valid_0's binary_logloss: 0.400296
[1046]  valid_0's binary_logloss: 0.400295
[1047]  valid_0's binary_logloss: 0.400288
[1048]  valid_0's binary_logloss: 0.400288
[1049]  valid_0's binary_logloss: 0.400285
[1050]  valid_0's binary_logloss: 0.400287
[1051]  valid_0's binary_logloss: 0.400285
[1052]  valid_0's binary_logloss: 0.400285
[1053]  valid_0's binary_logloss: 0.400287
[1054]  valid_0's binary_logloss: 0.400295
[1055]  valid_0's binary_logloss: 0.400304
[1056]  valid_0's binary_logloss: 0.40031
[1057]  valid_0's binary_logloss: 0.400304
[1058]  valid_0's binary_logloss: 0.400308
[1059]  valid_0's binary_logloss: 0.400309
[1060]  valid_0's binary_logloss: 0.400297
[1061]  valid_0's binary_logloss: 0.400296
[1062]  valid_0's binary_logloss: 0.400298
[1063]  valid_0's binary_logloss: 0.400309
[1064]  valid_0's binary_logloss: 0.400307
[1065]  valid_0's binary_logloss: 0.400307
[1066]  valid_0's binary_logloss: 0.400312
[1067]  valid_0's binary_logloss: 0.400319
[1068]  valid_0's binary_logloss: 0.400324
[1069]  valid_0's binary_logloss: 0.400326
[1070]  valid_0's binary_logloss: 0.400329
[1071]  valid_0's binary_logloss: 0.400328
[1072]  valid_0's binary_logloss: 0.400324
[1073]  valid_0's binary_logloss: 0.400323
[1074]  valid_0's binary_logloss: 0.400323
[1075]  valid_0's binary_logloss: 0.400322
[1076]  valid_0's binary_logloss: 0.400326
[1077]  valid_0's binary_logloss: 0.400324
[1078]  valid_0's binary_logloss: 0.40032
[1079]  valid_0's binary_logloss: 0.400323
[1080]  valid_0's binary_logloss: 0.40032
[1081]  valid_0's binary_logloss: 0.400312
[1082]  valid_0's binary_logloss: 0.400304
[1083]  valid_0's binary_logloss: 0.400298
[1084]  valid_0's binary_logloss: 0.400296
[1085]  valid_0's binary_logloss: 0.400303
[1086]  valid_0's binary_logloss: 0.400293
[1087]  valid_0's binary_logloss: 0.400289
[1088]  valid_0's binary_logloss: 0.400285
[1089]  valid_0's binary_logloss: 0.400292
[1090]  valid_0's binary_logloss: 0.400288
[1091]  valid_0's binary_logloss: 0.400286
[1092]  valid_0's binary_logloss: 0.400278
[1093]  valid_0's binary_logloss: 0.400275
[1094]  valid_0's binary_logloss: 0.400271
[1095]  valid_0's binary_logloss: 0.400269
[1096]  valid_0's binary_logloss: 0.400272
```

```
[1097]  valid_0's binary_logloss: 0.400271
[1098]  valid_0's binary_logloss: 0.400273
[1099]  valid_0's binary_logloss: 0.400273
[1100]  valid_0's binary_logloss: 0.400278
[1101]  valid_0's binary_logloss: 0.400264
[1102]  valid_0's binary_logloss: 0.40026
[1103]  valid_0's binary_logloss: 0.400249
[1104]  valid_0's binary_logloss: 0.400255
[1105]  valid_0's binary_logloss: 0.40025
[1106]  valid_0's binary_logloss: 0.400254
[1107]  valid_0's binary_logloss: 0.400246
[1108]  valid_0's binary_logloss: 0.400243
[1109]  valid_0's binary_logloss: 0.400239
[1110]  valid_0's binary_logloss: 0.40024
[1111]  valid_0's binary_logloss: 0.400241
[1112]  valid_0's binary_logloss: 0.400246
[1113]  valid_0's binary_logloss: 0.40025
[1114]  valid_0's binary_logloss: 0.400255
[1115]  valid_0's binary_logloss: 0.400255
[1116]  valid_0's binary_logloss: 0.400251
[1117]  valid_0's binary_logloss: 0.400255
[1118]  valid_0's binary_logloss: 0.400261
[1119]  valid_0's binary_logloss: 0.400262
[1120]  valid_0's binary_logloss: 0.40026
[1121]  valid_0's binary_logloss: 0.400251
[1122]  valid_0's binary_logloss: 0.400239
[1123]  valid_0's binary_logloss: 0.400236
[1124]  valid_0's binary_logloss: 0.40023
[1125]  valid_0's binary_logloss: 0.400221
[1126]  valid_0's binary_logloss: 0.400224
[1127]  valid_0's binary_logloss: 0.400228
[1128]  valid_0's binary_logloss: 0.400226
[1129]  valid_0's binary_logloss: 0.400221
[1130]  valid_0's binary_logloss: 0.400225
[1131]  valid_0's binary_logloss: 0.400219
[1132]  valid_0's binary_logloss: 0.400218
[1133]  valid_0's binary_logloss: 0.40022
[1134]  valid_0's binary_logloss: 0.400225
[1135]  valid_0's binary_logloss: 0.40024
[1136]  valid_0's binary_logloss: 0.400258
[1137]  valid_0's binary_logloss: 0.400236
[1138]  valid_0's binary_logloss: 0.400237
[1139]  valid_0's binary_logloss: 0.40022
[1140]  valid_0's binary_logloss: 0.400222
[1141]  valid_0's binary_logloss: 0.400224
[1142]  valid_0's binary_logloss: 0.40023
[1143]  valid_0's binary_logloss: 0.400229
[1144]  valid_0's binary_logloss: 0.400222
[1145]  valid_0's binary_logloss: 0.400231
[1146]  valid_0's binary_logloss: 0.400231
[1147]  valid_0's binary_logloss: 0.40023
[1148]  valid_0's binary_logloss: 0.400227
[1149]  valid_0's binary_logloss: 0.400224
[1150]  valid_0's binary_logloss: 0.400228
[1151]  valid_0's binary_logloss: 0.400214
[1152]  valid_0's binary_logloss: 0.400205
[1153]  valid_0's binary_logloss: 0.400215
[1154]  valid_0's binary_logloss: 0.400217
[1155]  valid_0's binary_logloss: 0.400227
[1156]  valid_0's binary_logloss: 0.400231
[1157]  valid_0's binary_logloss: 0.400237
```

```
[1158]  valid_0's binary_logloss: 0.400244
[1159]  valid_0's binary_logloss: 0.400239
[1160]  valid_0's binary_logloss: 0.400242
[1161]  valid_0's binary_logloss: 0.400246
[1162]  valid_0's binary_logloss: 0.400244
[1163]  valid_0's binary_logloss: 0.400252
[1164]  valid_0's binary_logloss: 0.400256
[1165]  valid_0's binary_logloss: 0.400263
[1166]  valid_0's binary_logloss: 0.400275
[1167]  valid_0's binary_logloss: 0.400277
[1168]  valid_0's binary_logloss: 0.400265
[1169]  valid_0's binary_logloss: 0.400265
[1170]  valid_0's binary_logloss: 0.400272
[1171]  valid_0's binary_logloss: 0.400272
[1172]  valid_0's binary_logloss: 0.400272
[1173]  valid_0's binary_logloss: 0.400285
[1174]  valid_0's binary_logloss: 0.400285
[1175]  valid_0's binary_logloss: 0.40028
[1176]  valid_0's binary_logloss: 0.400285
[1177]  valid_0's binary_logloss: 0.400283
[1178]  valid_0's binary_logloss: 0.400293
[1179]  valid_0's binary_logloss: 0.400285
[1180]  valid_0's binary_logloss: 0.40028
[1181]  valid_0's binary_logloss: 0.400278
[1182]  valid_0's binary_logloss: 0.400285
[1183]  valid_0's binary_logloss: 0.400299
[1184]  valid_0's binary_logloss: 0.400305
[1185]  valid_0's binary_logloss: 0.4003
[1186]  valid_0's binary_logloss: 0.400314
[1187]  valid_0's binary_logloss: 0.400322
[1188]  valid_0's binary_logloss: 0.400332
[1189]  valid_0's binary_logloss: 0.400336
[1190]  valid_0's binary_logloss: 0.400335
[1191]  valid_0's binary_logloss: 0.400338
[1192]  valid_0's binary_logloss: 0.400342
[1193]  valid_0's binary_logloss: 0.400339
[1194]  valid_0's binary_logloss: 0.40034
[1195]  valid_0's binary_logloss: 0.400336
[1196]  valid_0's binary_logloss: 0.400324
[1197]  valid_0's binary_logloss: 0.40033
[1198]  valid_0's binary_logloss: 0.40033
[1199]  valid_0's binary_logloss: 0.400334
[1200]  valid_0's binary_logloss: 0.400343
[1201]  valid_0's binary_logloss: 0.400343
[1202]  valid_0's binary_logloss: 0.400352
[1203]  valid_0's binary_logloss: 0.400353
[1204]  valid_0's binary_logloss: 0.400357
[1205]  valid_0's binary_logloss: 0.400364
[1206]  valid_0's binary_logloss: 0.400361
[1207]  valid_0's binary_logloss: 0.400361
[1208]  valid_0's binary_logloss: 0.400363
[1209]  valid_0's binary_logloss: 0.400371
[1210]  valid_0's binary_logloss: 0.400374
[1211]  valid_0's binary_logloss: 0.400372
[1212]  valid_0's binary_logloss: 0.400366
[1213]  valid_0's binary_logloss: 0.400377
[1214]  valid_0's binary_logloss: 0.400382
[1215]  valid_0's binary_logloss: 0.400392
[1216]  valid_0's binary_logloss: 0.400387
[1217]  valid_0's binary_logloss: 0.400392
[1218]  valid_0's binary_logloss: 0.400387
```

```
[1219]  valid_0's binary_logloss: 0.400383
[1220]  valid_0's binary_logloss: 0.400379
[1221]  valid_0's binary_logloss: 0.400373
[1222]  valid_0's binary_logloss: 0.40037
[1223]  valid_0's binary_logloss: 0.400372
[1224]  valid_0's binary_logloss: 0.400356
[1225]  valid_0's binary_logloss: 0.400364
[1226]  valid_0's binary_logloss: 0.400359
[1227]  valid_0's binary_logloss: 0.400352
[1228]  valid_0's binary_logloss: 0.400365
[1229]  valid_0's binary_logloss: 0.400369
[1230]  valid_0's binary_logloss: 0.400364
[1231]  valid_0's binary_logloss: 0.400359
[1232]  valid_0's binary_logloss: 0.400358
[1233]  valid_0's binary_logloss: 0.400359
[1234]  valid_0's binary_logloss: 0.400364
[1235]  valid_0's binary_logloss: 0.400361
[1236]  valid_0's binary_logloss: 0.400356
[1237]  valid_0's binary_logloss: 0.400349
[1238]  valid_0's binary_logloss: 0.400345
[1239]  valid_0's binary_logloss: 0.400339
[1240]  valid_0's binary_logloss: 0.400342
[1241]  valid_0's binary_logloss: 0.400344
[1242]  valid_0's binary_logloss: 0.400344
[1243]  valid_0's binary_logloss: 0.400349
[1244]  valid_0's binary_logloss: 0.400347
[1245]  valid_0's binary_logloss: 0.400345
[1246]  valid_0's binary_logloss: 0.400343
[1247]  valid_0's binary_logloss: 0.40035
[1248]  valid_0's binary_logloss: 0.400358
[1249]  valid_0's binary_logloss: 0.400349
[1250]  valid_0's binary_logloss: 0.400346
[1251]  valid_0's binary_logloss: 0.400337
[1252]  valid_0's binary_logloss: 0.400332
[1253]  valid_0's binary_logloss: 0.40032
[1254]  valid_0's binary_logloss: 0.400322
[1255]  valid_0's binary_logloss: 0.400324
[1256]  valid_0's binary_logloss: 0.400308
[1257]  valid_0's binary_logloss: 0.400298
[1258]  valid_0's binary_logloss: 0.400302
[1259]  valid_0's binary_logloss: 0.400301
[1260]  valid_0's binary_logloss: 0.400305
[1261]  valid_0's binary_logloss: 0.400299
[1262]  valid_0's binary_logloss: 0.400294
[1263]  valid_0's binary_logloss: 0.400297
[1264]  valid_0's binary_logloss: 0.40029
[1265]  valid_0's binary_logloss: 0.400297
[1266]  valid_0's binary_logloss: 0.400306
[1267]  valid_0's binary_logloss: 0.40031
[1268]  valid_0's binary_logloss: 0.400301
[1269]  valid_0's binary_logloss: 0.400298
[1270]  valid_0's binary_logloss: 0.400307
[1271]  valid_0's binary_logloss: 0.400313
[1272]  valid_0's binary_logloss: 0.400322
[1273]  valid_0's binary_logloss: 0.400325
[1274]  valid_0's binary_logloss: 0.400319
[1275]  valid_0's binary_logloss: 0.400329
[1276]  valid_0's binary_logloss: 0.400313
[1277]  valid_0's binary_logloss: 0.400313
[1278]  valid_0's binary_logloss: 0.400312
[1279]  valid_0's binary_logloss: 0.400306
```

```
[1280]  valid_0's binary_logloss: 0.400313
[1281]  valid_0's binary_logloss: 0.400321
[1282]  valid_0's binary_logloss: 0.400329
[1283]  valid_0's binary_logloss: 0.400342
[1284]  valid_0's binary_logloss: 0.400351
[1285]  valid_0's binary_logloss: 0.400361
[1286]  valid_0's binary_logloss: 0.400356
[1287]  valid_0's binary_logloss: 0.400351
[1288]  valid_0's binary_logloss: 0.400361
[1289]  valid_0's binary_logloss: 0.400352
[1290]  valid_0's binary_logloss: 0.40035
[1291]  valid_0's binary_logloss: 0.400339
[1292]  valid_0's binary_logloss: 0.400344
[1293]  valid_0's binary_logloss: 0.400338
[1294]  valid_0's binary_logloss: 0.400335
[1295]  valid_0's binary_logloss: 0.40034
[1296]  valid_0's binary_logloss: 0.400342
[1297]  valid_0's binary_logloss: 0.40034
[1298]  valid_0's binary_logloss: 0.400346
[1299]  valid_0's binary_logloss: 0.400347
[1300]  valid_0's binary_logloss: 0.400351
[1301]  valid_0's binary_logloss: 0.400358
[1302]  valid_0's binary_logloss: 0.400363
[1303]  valid_0's binary_logloss: 0.400369
[1304]  valid_0's binary_logloss: 0.400374
[1305]  valid_0's binary_logloss: 0.400383
[1306]  valid_0's binary_logloss: 0.400382
[1307]  valid_0's binary_logloss: 0.40038
[1308]  valid_0's binary_logloss: 0.400381
[1309]  valid_0's binary_logloss: 0.40038
[1310]  valid_0's binary_logloss: 0.400379
[1311]  valid_0's binary_logloss: 0.400386
[1312]  valid_0's binary_logloss: 0.400385
[1313]  valid_0's binary_logloss: 0.400386
[1314]  valid_0's binary_logloss: 0.400377
[1315]  valid_0's binary_logloss: 0.40038
[1316]  valid_0's binary_logloss: 0.400389
[1317]  valid_0's binary_logloss: 0.400388
[1318]  valid_0's binary_logloss: 0.400383
[1319]  valid_0's binary_logloss: 0.400382
[1320]  valid_0's binary_logloss: 0.400383
[1321]  valid_0's binary_logloss: 0.400387
[1322]  valid_0's binary_logloss: 0.400394
[1323]  valid_0's binary_logloss: 0.4004
[1324]  valid_0's binary_logloss: 0.400402
[1325]  valid_0's binary_logloss: 0.400412
[1326]  valid_0's binary_logloss: 0.400413
[1327]  valid_0's binary_logloss: 0.400412
[1328]  valid_0's binary_logloss: 0.400414
[1329]  valid_0's binary_logloss: 0.400409
[1330]  valid_0's binary_logloss: 0.400407
[1331]  valid_0's binary_logloss: 0.400407
[1332]  valid_0's binary_logloss: 0.400414
[1333]  valid_0's binary_logloss: 0.400422
[1334]  valid_0's binary_logloss: 0.400416
[1335]  valid_0's binary_logloss: 0.400412
[1336]  valid_0's binary_logloss: 0.400411
[1337]  valid_0's binary_logloss: 0.40041
[1338]  valid_0's binary_logloss: 0.400416
[1339]  valid_0's binary_logloss: 0.400423
[1340]  valid_0's binary_logloss: 0.400421
```

```
[1341]  valid_0's binary_logloss: 0.400415
[1342]  valid_0's binary_logloss: 0.400413
[1343]  valid_0's binary_logloss: 0.400407
[1344]  valid_0's binary_logloss: 0.400404
[1345]  valid_0's binary_logloss: 0.400409
[1346]  valid_0's binary_logloss: 0.400413
[1347]  valid_0's binary_logloss: 0.40041
[1348]  valid_0's binary_logloss: 0.400408
[1349]  valid_0's binary_logloss: 0.400403
[1350]  valid_0's binary_logloss: 0.400414
[1351]  valid_0's binary_logloss: 0.400414
[1352]  valid_0's binary_logloss: 0.400419
[1353]  valid_0's binary_logloss: 0.400421
[1354]  valid_0's binary_logloss: 0.400424
[1355]  valid_0's binary_logloss: 0.400428
[1356]  valid_0's binary_logloss: 0.400424
[1357]  valid_0's binary_logloss: 0.400424
[1358]  valid_0's binary_logloss: 0.400418
[1359]  valid_0's binary_logloss: 0.400412
[1360]  valid_0's binary_logloss: 0.400413
[1361]  valid_0's binary_logloss: 0.400402
[1362]  valid_0's binary_logloss: 0.400405
[1363]  valid_0's binary_logloss: 0.400407
[1364]  valid_0's binary_logloss: 0.400407
[1365]  valid_0's binary_logloss: 0.40041
[1366]  valid_0's binary_logloss: 0.400408
[1367]  valid_0's binary_logloss: 0.400406
[1368]  valid_0's binary_logloss: 0.400403
[1369]  valid_0's binary_logloss: 0.400411
[1370]  valid_0's binary_logloss: 0.400415
[1371]  valid_0's binary_logloss: 0.400414
[1372]  valid_0's binary_logloss: 0.400412
[1373]  valid_0's binary_logloss: 0.400412
[1374]  valid_0's binary_logloss: 0.400409
[1375]  valid_0's binary_logloss: 0.40041
[1376]  valid_0's binary_logloss: 0.400409
[1377]  valid_0's binary_logloss: 0.400408
[1378]  valid_0's binary_logloss: 0.400407
[1379]  valid_0's binary_logloss: 0.400417
[1380]  valid_0's binary_logloss: 0.400418
[1381]  valid_0's binary_logloss: 0.400407
[1382]  valid_0's binary_logloss: 0.400404
[1383]  valid_0's binary_logloss: 0.40039
[1384]  valid_0's binary_logloss: 0.400399
[1385]  valid_0's binary_logloss: 0.400398
[1386]  valid_0's binary_logloss: 0.400387
[1387]  valid_0's binary_logloss: 0.400389
[1388]  valid_0's binary_logloss: 0.400377
[1389]  valid_0's binary_logloss: 0.400378
[1390]  valid_0's binary_logloss: 0.400382
[1391]  valid_0's binary_logloss: 0.40038
[1392]  valid_0's binary_logloss: 0.400375
[1393]  valid_0's binary_logloss: 0.400376
[1394]  valid_0's binary_logloss: 0.400374
[1395]  valid_0's binary_logloss: 0.400373
[1396]  valid_0's binary_logloss: 0.400377
[1397]  valid_0's binary_logloss: 0.400374
[1398]  valid_0's binary_logloss: 0.400365
[1399]  valid_0's binary_logloss: 0.400368
[1400]  valid_0's binary_logloss: 0.400366
[1401]  valid_0's binary_logloss: 0.400367
```

In [119]:

```
print(gbm.best_score)
print(gbm.best_iteration)
```

```
defaultdict(<class 'dict'>, {'valid_0': {'binary_logloss': 0.400195551349
58557}})
918
```

In [148]:

```
from operator import itemgetter
from sklearn.model_selection import train_test_split
import xgboost as xgb
from sklearn.metrics import roc_auc_score

def run_default_test(train, test, features, target, random_state=0):
    eta = 0.1
    max_depth = 5
    subsample = 0.8
    colsample_bytree = 0.8
    print('XGBoost params. ETA: {}, MAX_DEPTH: {}, SUBSAMPLE: {}, COLSAMPLE_BY_TREE: {}'
    params = {
        "objective": "binary:logistic",
        "booster" : "gbtree",
        "eval_metric": "logloss",
        "eta": eta,
        "max_depth": max_depth,
        "subsample": subsample,
        "colsample_bytree": colsample_bytree,
        "silent": 1,
        "seed": random_state
    }
    num_boost_round = 260
    early_stopping_rounds = 20
    test_size = 0.2
    X_train, X_valid = train_test_split(train, test_size=test_size, random_state=random_
    y_train = X_train[target]
    y_valid = X_valid[target]
    dtrain = xgb.DMatrix(X_train[features], y_train)
    dvalid = xgb.DMatrix(X_valid[features], y_valid)
    watchlist = [(dtrain, 'train'), (dvalid, 'eval')]
    gbm = xgb.train(params, dtrain, num_boost_round, evals=watchlist, early_stopping_rou
```

```
[1402]  valid_0's binary_logloss: 0.400363
[1403]  valid_0's binary_logloss: 0.400362
[1404]  valid_0's binary_logloss: 0.400363
[1405]  valid_0's binary_logloss: 0.400361
[1406]  valid_0's binary_logloss: 0.400363
[1407]  valid_0's binary_logloss: 0.400358
[1408]  valid_0's binary_logloss: 0.400351
[1409]  valid_0's binary_logloss: 0.400353
[1410]  valid_0's binary_logloss: 0.400354
[1411]  valid_0's binary_logloss: 0.400356
[1412]  valid_0's binary_logloss: 0.400355
[1413]  valid_0's binary_logloss: 0.400355
[1414]  valid_0's binary_logloss: 0.400348
[1415]  valid_0's binary_logloss: 0.400343
[1416]  valid_0's binary_logloss: 0.400344
[1417]  valid_0's binary_logloss: 0.400351
[1418]  valid_0's binary_logloss: 0.400351
[1419]  valid_0's binary_logloss: 0.40035
[1420]  valid_0's binary_logloss: 0.400356
[1421]  valid_0's binary_logloss: 0.400355
[1422]  valid_0's binary_logloss: 0.400355
[1423]  valid_0's binary_logloss: 0.400363
[1424]  valid_0's binary_logloss: 0.400366
[1425]  valid_0's binary_logloss: 0.400372
[1426]  valid_0's binary_logloss: 0.400375
[1427]  valid_0's binary_logloss: 0.400378
[1428]  valid_0's binary_logloss: 0.400389
[1429]  valid_0's binary_logloss: 0.400395
[1430]  valid_0's binary_logloss: 0.400411
[1431]  valid_0's binary_logloss: 0.400407
[1432]  valid_0's binary_logloss: 0.400406
[1433]  valid_0's binary_logloss: 0.400402
[1434]  valid_0's binary_logloss: 0.400397
[1435]  valid_0's binary_logloss: 0.400394
[1436]  valid_0's binary_logloss: 0.400396
[1437]  valid_0's binary_logloss: 0.400392
[1439]  valid_0's binary_logloss: 0.400405
[1440]  valid_0's binary_logloss: 0.400411
[1441]  valid_0's binary_logloss: 0.40041
[1442]  valid_0's binary_logloss: 0.40041
[1443]  valid_0's binary_logloss: 0.400408
[1444]  valid_0's binary_logloss: 0.400415
[1445]  valid_0's binary_logloss: 0.40042
[1446]  valid_0's binary_logloss: 0.400417
[1447]  valid_0's binary_logloss: 0.400416
[1448]  valid_0's binary_logloss: 0.400407
[1449]  valid_0's binary_logloss: 0.400402
[1450]  valid_0's binary_logloss: 0.400399
[1451]  valid_0's binary_logloss: 0.400398
[1452]  valid_0's binary_logloss: 0.400392
[1453]  valid_0's binary_logloss: 0.400389
[1454]  valid_0's binary_logloss: 0.400395
[1455]  valid_0's binary_logloss: 0.400395
[1456]  valid_0's binary_logloss: 0.40039
[1457]  valid_0's binary_logloss: 0.400388
[1458]  valid_0's binary_logloss: 0.400395
Early stopping, best iteration is:
[958]   valid_0's binary_logloss: 0.400196
```

In [149]:

```
features = ['C1', 'banner_pos', 'device_type', 'device_conn_type', 'C14',
       'C15', 'C16', 'C17', 'C18', 'C19', 'C20', 'C21', 'hour_of_day',
       'site_id_int', 'site_domain_int', 'site_category_int', 'app_id_int',
       'app_domain_int', 'app_category_int', 'device_id_int', 'device_ip_int',
       'device_model_int', 'day_of_week_int']
run_default_test(train, y_target, features, 'click')
```

```
XGBoost params. ETA: 0.1, MAX_DEPTH: 5, SUBSAMPLE: 0.8, COLSAMPLE_BY_TR
E: 0.8
[0]     train-logloss:0.648232  eval-logloss:0.648223
Multiple eval metrics have been passed: 'eval-logloss' will be used for
arly stopping.

Will train until eval-logloss hasn't improved in 20 rounds.
[1]     train-logloss:0.61155   eval-logloss:0.611518
[2]     train-logloss:0.58168   eval-logloss:0.581626
[3]     train-logloss:0.556558  eval-logloss:0.55647
[4]     train-logloss:0.535377  eval-logloss:0.535272
[5]     train-logloss:0.517634  eval-logloss:0.517516
[6]     train-logloss:0.502628  eval-logloss:0.502504
[7]     train-logloss:0.490077  eval-logloss:0.489942
[8]     train-logloss:0.479384  eval-logloss:0.47925
[9]     train-logloss:0.470294  eval-logloss:0.470162
[10]    train-logloss:0.462493  eval-logloss:0.462368
[11]    train-logloss:0.455896  eval-logloss:0.455775
[12]    train-logloss:0.450228  eval-logloss:0.450099
[13]    train-logloss:0.445374  eval-logloss:0.445241
[14]    train-logloss:0.441245  eval-logloss:0.44111
[15]    train-logloss:0.437608  eval-logloss:0.437485
[16]    train-logloss:0.434544  eval-logloss:0.434416
[17]    train-logloss:0.432117  eval-logloss:0.431993
[18]    train-logloss:0.429779  eval-logloss:0.429651
[19]    train-logloss:0.42779   eval-logloss:0.427678
[20]    train-logloss:0.426088  eval-logloss:0.425999
[21]    train-logloss:0.424666  eval-logloss:0.424595
[22]    train-logloss:0.423311  eval-logloss:0.42325
[23]    train-logloss:0.422208  eval-logloss:0.422154
[24]    train-logloss:0.421275  eval-logloss:0.421255
[25]    train-logloss:0.420473  eval-logloss:0.420482
[26]    train-logloss:0.419799  eval-logloss:0.419823
[27]    train-logloss:0.419168  eval-logloss:0.419197
[28]    train-logloss:0.41857   eval-logloss:0.418609
[29]    train-logloss:0.417968  eval-logloss:0.418018
[30]    train-logloss:0.417485  eval-logloss:0.41757
[31]    train-logloss:0.416933  eval-logloss:0.416999
[32]    train-logloss:0.416559  eval-logloss:0.41663
[33]    train-logloss:0.41615   eval-logloss:0.416242
[34]    train-logloss:0.415732  eval-logloss:0.415851
[35]    train-logloss:0.415339  eval-logloss:0.415452
[36]    train-logloss:0.415014  eval-logloss:0.415166
[37]    train-logloss:0.414749  eval-logloss:0.414904
[38]    train-logloss:0.414566  eval-logloss:0.414739
[39]    train-logloss:0.41436   eval-logloss:0.41455
[40]    train-logloss:0.41412   eval-logloss:0.414327
[41]    train-logloss:0.413915  eval-logloss:0.414117
[42]    train-logloss:0.413575  eval-logloss:0.413784
[43]    train-logloss:0.413333  eval-logloss:0.413568
[44]    train-logloss:0.413138  eval-logloss:0.413388
[45]    train-logloss:0.412901  eval-logloss:0.413173
[46]    train-logloss:0.412725  eval-logloss:0.413009
[47]    train-logloss:0.412559  eval-logloss:0.412866
[48]    train-logloss:0.412457  eval-logloss:0.412772
[49]    train-logloss:0.412147  eval-logloss:0.412479
[50]    train-logloss:0.411933  eval-logloss:0.412286
[51]    train-logloss:0.411796  eval-logloss:0.412159
[52]    train-logloss:0.411628  eval-logloss:0.412014
[53]    train-logloss:0.411553  eval-logloss:0.411954
[54]    train-logloss:0.411419  eval-logloss:0.41182
```

```
[55]    train-logloss:0.411314  eval-logloss:0.411735
[56]    train-logloss:0.411096  eval-logloss:0.411532
[57]    train-logloss:0.410938  eval-logloss:0.411395
[58]    train-logloss:0.410879  eval-logloss:0.411336
[59]    train-logloss:0.410813  eval-logloss:0.411285
[60]    train-logloss:0.410695  eval-logloss:0.411174
[61]    train-logloss:0.410601  eval-logloss:0.411095
[62]    train-logloss:0.410435  eval-logloss:0.410948
[63]    train-logloss:0.410352  eval-logloss:0.410878
[64]    train-logloss:0.410267  eval-logloss:0.410803
[65]    train-logloss:0.410013  eval-logloss:0.410576
[66]    train-logloss:0.409881  eval-logloss:0.41046
[67]    train-logloss:0.409842  eval-logloss:0.410423
[68]    train-logloss:0.409736  eval-logloss:0.410312
[69]    train-logloss:0.409666  eval-logloss:0.410257
[70]    train-logloss:0.409498  eval-logloss:0.410095
[71]    train-logloss:0.409376  eval-logloss:0.409983
[72]    train-logloss:0.409304  eval-logloss:0.409914
[73]    train-logloss:0.40918   eval-logloss:0.409816
[74]    train-logloss:0.409049  eval-logloss:0.409701
[75]    train-logloss:0.40896   eval-logloss:0.409616
[76]    train-logloss:0.408809  eval-logloss:0.40948
[77]    train-logloss:0.408718  eval-logloss:0.40942
[78]    train-logloss:0.408643  eval-logloss:0.409358
[79]    train-logloss:0.408513  eval-logloss:0.409234
[80]    train-logloss:0.408397  eval-logloss:0.409121
[81]    train-logloss:0.408273  eval-logloss:0.409013
[82]    train-logloss:0.408168  eval-logloss:0.408923
[83]    train-logloss:0.408105  eval-logloss:0.40887
[84]    train-logloss:0.408003  eval-logloss:0.408773
[85]    train-logloss:0.407891  eval-logloss:0.408693
[86]    train-logloss:0.407807  eval-logloss:0.408621
[87]    train-logloss:0.407764  eval-logloss:0.408584
[88]    train-logloss:0.407723  eval-logloss:0.408557
[89]    train-logloss:0.407643  eval-logloss:0.408477
[90]    train-logloss:0.407581  eval-logloss:0.408427
[91]    train-logloss:0.407473  eval-logloss:0.408321
[92]    train-logloss:0.407419  eval-logloss:0.408286
[93]    train-logloss:0.407373  eval-logloss:0.408253
[94]    train-logloss:0.407222  eval-logloss:0.40811
[95]    train-logloss:0.407169  eval-logloss:0.408056
[96]    train-logloss:0.407038  eval-logloss:0.407926
[97]    train-logloss:0.406991  eval-logloss:0.407886
[98]    train-logloss:0.40693   eval-logloss:0.407833
[99]    train-logloss:0.40689   eval-logloss:0.407796
[100]   train-logloss:0.406831  eval-logloss:0.407764
[101]   train-logloss:0.406798  eval-logloss:0.407751
[102]   train-logloss:0.406725  eval-logloss:0.407687
[103]   train-logloss:0.406649  eval-logloss:0.407638
[104]   train-logloss:0.406609  eval-logloss:0.407606
[105]   train-logloss:0.406553  eval-logloss:0.40756
[106]   train-logloss:0.406487  eval-logloss:0.407494
[107]   train-logloss:0.406455  eval-logloss:0.407479
[108]   train-logloss:0.406314  eval-logloss:0.407359
[109]   train-logloss:0.406262  eval-logloss:0.40732
[110]   train-logloss:0.406158  eval-logloss:0.407231
[111]   train-logloss:0.406113  eval-logloss:0.407188
[112]   train-logloss:0.406082  eval-logloss:0.407167
[113]   train-logloss:0.406014  eval-logloss:0.407112
[114]   train-logloss:0.405992  eval-logloss:0.407102
[115]   train-logloss:0.405888  eval-logloss:0.407009
```

```
[116]   train-logloss:0.405814  eval-logloss:0.406954
[117]   train-logloss:0.405766  eval-logloss:0.406929
[118]   train-logloss:0.405718  eval-logloss:0.406891
[119]   train-logloss:0.405676  eval-logloss:0.406861
[120]   train-logloss:0.405594  eval-logloss:0.406802
[121]   train-logloss:0.405529  eval-logloss:0.406753
[122]   train-logloss:0.405468  eval-logloss:0.406705
[123]   train-logloss:0.405457  eval-logloss:0.4067
[124]   train-logloss:0.405371  eval-logloss:0.406627
[125]   train-logloss:0.405235  eval-logloss:0.406496
[126]   train-logloss:0.405219  eval-logloss:0.406484
[127]   train-logloss:0.405145  eval-logloss:0.406428
[128]   train-logloss:0.405104  eval-logloss:0.406399
[129]   train-logloss:0.405056  eval-logloss:0.406361
[130]   train-logloss:0.40503   eval-logloss:0.406348
[131]   train-logloss:0.404909  eval-logloss:0.406244
[132]   train-logloss:0.40484   eval-logloss:0.406193
[133]   train-logloss:0.404815  eval-logloss:0.406181
[134]   train-logloss:0.404703  eval-logloss:0.406078
[135]   train-logloss:0.404601  eval-logloss:0.405981
[136]   train-logloss:0.404517  eval-logloss:0.40592
[137]   train-logloss:0.404493  eval-logloss:0.405905
[138]   train-logloss:0.40444   eval-logloss:0.405866
[139]   train-logloss:0.404362  eval-logloss:0.405802
[140]   train-logloss:0.404321  eval-logloss:0.405778
[141]   train-logloss:0.404264  eval-logloss:0.405716
[142]   train-logloss:0.404216  eval-logloss:0.405683
[143]   train-logloss:0.404178  eval-logloss:0.405664
[144]   train-logloss:0.404126  eval-logloss:0.405633
[145]   train-logloss:0.404104  eval-logloss:0.405615
[146]   train-logloss:0.404005  eval-logloss:0.405533
[147]   train-logloss:0.403965  eval-logloss:0.405498
[148]   train-logloss:0.403949  eval-logloss:0.405492
[149]   train-logloss:0.403918  eval-logloss:0.405471
[150]   train-logloss:0.403855  eval-logloss:0.405425
[151]   train-logloss:0.403826  eval-logloss:0.405403
[152]   train-logloss:0.403745  eval-logloss:0.405341
[153]   train-logloss:0.40373   eval-logloss:0.405328
[154]   train-logloss:0.403615  eval-logloss:0.405225
[155]   train-logloss:0.403589  eval-logloss:0.405212
[156]   train-logloss:0.403564  eval-logloss:0.405201
[157]   train-logloss:0.403554  eval-logloss:0.405194
[158]   train-logloss:0.403485  eval-logloss:0.405141
[159]   train-logloss:0.403416  eval-logloss:0.4051
[160]   train-logloss:0.403284  eval-logloss:0.404985
[161]   train-logloss:0.403271  eval-logloss:0.404979
[162]   train-logloss:0.403216  eval-logloss:0.404933
[163]   train-logloss:0.403177  eval-logloss:0.4049
[164]   train-logloss:0.403079  eval-logloss:0.404811
[165]   train-logloss:0.403067  eval-logloss:0.404802
[166]   train-logloss:0.403031  eval-logloss:0.404785
[167]   train-logloss:0.403008  eval-logloss:0.404778
[168]   train-logloss:0.402993  eval-logloss:0.404776
[169]   train-logloss:0.40298   eval-logloss:0.40477
[170]   train-logloss:0.402937  eval-logloss:0.404732
[171]   train-logloss:0.402912  eval-logloss:0.404713
[172]   train-logloss:0.40288   eval-logloss:0.404697
[173]   train-logloss:0.402828  eval-logloss:0.404654
[174]   train-logloss:0.402806  eval-logloss:0.404637
[175]   train-logloss:0.402771  eval-logloss:0.404621
[176]   train-logloss:0.402726  eval-logloss:0.404585
```

```
[177]   train-logloss:0.402682  eval-logloss:0.404558
[178]   train-logloss:0.402669  eval-logloss:0.404554
[179]   train-logloss:0.402618  eval-logloss:0.404517
[180]   train-logloss:0.402572  eval-logloss:0.404486
[181]   train-logloss:0.402506  eval-logloss:0.404439
[182]   train-logloss:0.402438  eval-logloss:0.40438
[183]   train-logloss:0.402415  eval-logloss:0.404375
[184]   train-logloss:0.402378  eval-logloss:0.404351
[185]   train-logloss:0.402324  eval-logloss:0.404301
[186]   train-logloss:0.402315  eval-logloss:0.404298
[187]   train-logloss:0.402286  eval-logloss:0.404279
[188]   train-logloss:0.402252  eval-logloss:0.404248
[189]   train-logloss:0.402231  eval-logloss:0.404241
[190]   train-logloss:0.402202  eval-logloss:0.404228
[191]   train-logloss:0.402177  eval-logloss:0.404219
[192]   train-logloss:0.402133  eval-logloss:0.404185
[193]   train-logloss:0.402098  eval-logloss:0.404152
[194]   train-logloss:0.402045  eval-logloss:0.404103
[195]   train-logloss:0.402019  eval-logloss:0.404087
[196]   train-logloss:0.401991  eval-logloss:0.404077
[197]   train-logloss:0.401936  eval-logloss:0.404022
[198]   train-logloss:0.401901  eval-logloss:0.40399
[199]   train-logloss:0.401862  eval-logloss:0.403964
[200]   train-logloss:0.401819  eval-logloss:0.403943
[201]   train-logloss:0.401763  eval-logloss:0.403896
[202]   train-logloss:0.401733  eval-logloss:0.40388
[203]   train-logloss:0.401703  eval-logloss:0.403864
[204]   train-logloss:0.401691  eval-logloss:0.403854
[205]   train-logloss:0.40166   eval-logloss:0.403835
[206]   train-logloss:0.401619  eval-logloss:0.403812
[207]   train-logloss:0.401573  eval-logloss:0.403775
[208]   train-logloss:0.401562  eval-logloss:0.403772
[209]   train-logloss:0.401469  eval-logloss:0.403682
```

In [ ]: