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

```
from google.colab import drive
drive.mount("/content/drive")
```

```
Drive already mounted at /content/drive; to attempt to forcibly re
```

```
trainpath="/content/drive/MyDrive/dataset/Doceree-HCP_Train.csv"
train=pd.read_csv(trainpath,engine='python',encoding='latin1')
train.head()
```

	ID	DEVICETYPE	PLATFORM_ID	BIDREQUESTIP	USERPLATFORMUID
0	1001	Desktop	2	170.173.0.22	6974dcaa-f932-480e-9fb5-c52e20e1393a
1	1002	Desktop	2	65.216.253.25	c12f3f8f-8fcf-484a-90e1-1ac04db8cdcf
2	1003	Desktop	2	66.232.79.22	a698de4b-e200-46dd-b5fb-40402175ae18
3	1004	Desktop	3	137.54.125.246	45967533-75c8-4fbd-a00c-e6ff20447aaa
4	1005	Mobile	7	174.202.231.99	a17e25be-532d-4cf5-b916-9308c8c3961f

```
testpath="/content/drive/MyDrive/dataset/Doceree-HCP_Test.csv"
test=pd.read_csv(testpath)
test.head()
```

CompleteSolution.csv

1 to 10 of 28493 entries Filter

ID	IS_HCP	taxonomy
115501	0	2084P0800X
115502	0	2084P0800X
115503	0	2084P0800X
115504	0	207Q00000X
115505	0	2084P0800X
115506	0	2084P0800X
115507	0	2084P0800X
115508	0	2084P0800X
115509	0	2084P0800X
115510	0	2084P0800X

Show 10 per page

1 2 10 100 1000 2000 2800 2840 2850

	ID	DEVICETYPE	PLATFORM_ID	BIDREQUESTIP	USERPLATFORMUI
0	115501	Desktop	2	75.189.231.103	0d5041ff-f0b6 4d1a-9ad7 0a29f7d485b
1	115502	Mobile	2	24.101.33.158	c8396dd0-969 4d99-a40b b7bb1f51615
2	115503	Desktop	2	172.118.216.142	3c97a081-6518 43f8-9f26 369759cfb47
3	115504	Desktop	7	71.105.120.171	3e2578c8-f794 41af-a38c c5cfb3c0f01
4	115505	Desktop	2	73.82.211.73	ec2ae7ce-6a8c 4156-98a7 07203e60f48

```
print(train.shape)
print(test.shape)
```

```
(113937, 14)
(28493, 12)
```

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113937 entries, 0 to 113936
Data columns (total 14 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   ID               113937 non-null  int64
 1   DEVICETYPE       113937 non-null  object
 2   PLATFORM_ID      113937 non-null  int64
 3   BIDREQUESTIP     113937 non-null  object
 4   USERPLATFORMUID  113933 non-null  object
 5   USERCITY         107578 non-null  object
 6   USERZIPCODE      109345 non-null  float64
 7   USERAGENT        113935 non-null  object
 8   PLATFORMTYPE     113937 non-null  object
 9   CHANNELTYPE      113937 non-null  object
 10  URL              113937 non-null  object
 11  KEYWORDS         113937 non-null  object
 12  TAXONOMY         32313 non-null   object
 13  IS_HCP           113936 non-null  float64
dtypes: float64(2), int64(2), object(10)
memory usage: 12.2+ MB
```

```
train['IS_HCP']
```

```
0         0.0
1         0.0
2         0.0
3         1.0
4         0.0
         ...
113932    1.0
113933    1.0
113934    1.0
113935    1.0
113936    1.0
Name: IS_HCP, Length: 113937, dtype: float64
```

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28493 entries, 0 to 28492
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   ID               28493 non-null  int64
 1   DEVICETYPE       28493 non-null  object
 2   PLATFORM_ID      28493 non-null  int64
 3   BIDREQUESTIP     28493 non-null  object
 4   USERPLATFORMUID  28493 non-null  object
 5   USERCITY         26934 non-null  object
 6   USERZIPCODE      27378 non-null  float64
 7   USERAGENT        28492 non-null  object
 8   PLATFORMTYPE     28493 non-null  object
 9   CHANNELTYPE      28493 non-null  object
 10  URL              28493 non-null  object
 11  KEYWORDS         28493 non-null  object
dtypes: float64(1), int64(2), object(9)
memory usage: 2.6+ MB
```

```
train.isnull().sum()
```

```
ID                     0
DEVICETYPE             0
PLATFORM_ID            0
BIDREQUESTIP           0
USERPLATFORMUID        4
USERCITY            6359
USERZIPCODE         4592
USERAGENT              2
PLATFORMTYPE           0
CHANNELTYPE            0
URL                    0
KEYWORDS               0
TAXONOMY           81624
```

```
IS_HCP                 1
dtype: int64
```

```
train["USERPLATFORMUID"].value_counts()
```

```
3e2578c8-f794-41af-a38c-c5cfb3c0f014    1447
d76f7c0b-1a64-4d4b-b36d-86c832be8837    1406
d00f28d6-3a50-46cd-92f3-8601bd57ad0e     675
fcd3d327-0ad5-425e-8ea8-33ffe6a31543     629
fe53f32c-4646-4abe-afea-90b21d6b8531     617
                                        ...
6759cbf8-cf4f-4d0e-9eb8-00808e69e841       1
bd2007ca-d0b5-4620-8078-b7c6db62037c       1
0534212a-e0e9-49e7-ace9-608e0d0eae8a       1
26a06716-2716-4ac2-9131-fa0f01f84526       1
d26887c5-15f9-4b72-80dc-728039529e60       1
Name: USERPLATFORMUID, Length: 53780, dtype: int64
```

```
train["USERPLATFORMUID"].fillna('3e2578c8-f794-41af-a38c-c5cfb3c0f014',
train.isnull().sum()
```

```
ID                     0
DEVICETYPE             0
PLATFORM_ID            0
BIDREQUESTIP           0
USERPLATFORMUID        0
USERCITY            6359
USERZIPCODE         4592
USERAGENT              2
PLATFORMTYPE           0
CHANNELTYPE            0
URL                    0
KEYWORDS               0
TAXONOMY           81624
IS_HCP                 1
dtype: int64
```

```
train["USERCITY"].value_counts()
```

```
New York          4943
Brooklyn          3237
St Louis          2545
Los Angeles       2154
Houston           1747
                  ...
Foothill Ranch       1
Henrietta            1
Wartburg             1
Winfield             1
Gambier              1
Name: USERCITY, Length: 4420, dtype: int64
```

```
train["USERCITY"].fillna('New York',inplace=True)
train.isnull().sum()
```

```
ID                     0
DEVICETYPE             0
PLATFORM_ID            0
BIDREQUESTIP           0
```

```
USERPLATFORMUID        0
USERCITY               0
USERZIPCODE         4592
USERAGENT              2
PLATFORMTYPE           0
CHANNELTYPE            0
URL                    0
KEYWORDS               0
TAXONOMY           81624
IS_HCP                 1
dtype: int64
```

```
test["USERCITY"].value_counts()
```

```
New York       1204
Brooklyn        801
St Louis        665
Los Angeles     575
Houston         417
                ...
Ferndale          1
Los Lunas         1
Fox Lake          1
Fleetwood         1
Horsham           1
Name: USERCITY, Length: 3060, dtype: int64
```

```
test["USERCITY"].fillna('New York',inplace=True)
test.isnull().sum()
```

```
ID                    0
DEVICETYPE            0
PLATFORM_ID           0
BIDREQUESTIP          0
USERPLATFORMUID       0
USERCITY              0
USERZIPCODE        1115
USERAGENT             1
PLATFORMTYPE          0
CHANNELTYPE           0
URL                   0
KEYWORDS              0
dtype: int64
```

```
train["USERZIPCODE"].value_counts()
```

```
63169.0        2116
11226.0        1752
22202.0        1259
10001.0        1190
90060.0        1058
               ...
56748.0           1
77021.0           1
74880.0           1
93630.0           1
760865488.0       1
Name: USERZIPCODE, Length: 11278, dtype: int64
```

```
train['USERZIPCODE'].fillna("63169.0", inplace=True)
train.isnull().sum()
```

```
ID                     0
DEVICETYPE             0
PLATFORM_ID            0
BIDREQUESTIP           0
USERPLATFORMUID        0
USERCITY               0
USERZIPCODE            0
USERAGENT              2
PLATFORMTYPE           0
CHANNELTYPE            0
URL                    0
KEYWORDS               0
TAXONOMY           81624
IS_HCP                 1
dtype: int64
```

```
test["USERZIPCODE"].value_counts()
```

```
63169.0    557
11226.0    426
22202.0    316
10001.0    292
90060.0    281
          ...
14618.0      1
4064.0       1
33710.0      1
43209.0      1
98424.0      1
Name: USERZIPCODE, Length: 7037, dtype: int64
```

```
test['USERZIPCODE'].fillna('63169.0', inplace=True)
test.isnull().sum()
```

```
ID                 0
DEVICETYPE         0
PLATFORM_ID        0
BIDREQUESTIP       0
USERPLATFORMUID    0
USERCITY           0
USERZIPCODE        0
```

```
USERAGENT          1
PLATFORMTYPE       0
CHANNELTYPE        0
URL                0
KEYWORDS           0
dtype: int64
```

```
train['IS_HCP'].dtype
```

```
dtype('float64')
```

```
train["USERAGENT"].value_counts()
```

```
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36
8380
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
5721
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36
3868
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36
3757
Mozilla/5.0 (iPhone; CPU iPhone OS 16_3_1 like Mac OS X)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.3
Mobile/15E148 Safari/604.1                            3480

...
Mozilla/5.0 (Linux; Android 13; SM-G998U) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/110.0.0.0 Mobile Safari/537.36
EdgA/110.0.1587.66                                 1
Mozilla/5.0 (Linux; Android 12; SM-S124DL) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/111.0.0.0 Mobile Safari/537.36
1
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/85.0.4183.83 Safari/537.36
1
Mozilla/5.0 (Linux; Android 12; SM-A516V Build/SP1A.210812.016;
wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0
Chrome/105.0.5195.136 Mobile Safari/537.36       1
Mozilla/5.0 (iPhone; CPU iPhone OS 15_4 like Mac OS X)
AppleWebKit/605.1.15 (KHTML, like Gecko) CriOS/86.0.4240.93
Mobile/15E148 Safari/604.1                            1
Name: USERAGENT, Length: 4288, dtype: int64
```

```
train["USERAGENT"].fillna('Mozilla/5.0 (Windows NT 10.0; Win64; x64) App
train.isnull().sum()
```

```
ID                     0
DEVICETYPE             0
PLATFORM_ID            0
BIDREQUESTIP           0
```

```
USERPLATFORMUID        0
USERCITY               0
USERZIPCODE            0
USERAGENT              0
PLATFORMTYPE           0
CHANNELTYPE            0
URL                    0
KEYWORDS               0
TAXONOMY           81624
IS_HCP                 1
dtype: int64
```

```
test["USERAGENT"].value_counts()
```

```
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36
2058
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
1430
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36
979
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36
941
Mozilla/5.0 (iPhone; CPU iPhone OS 16_3_1 like Mac OS X)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.3
Mobile/15E148 Safari/604.1          863

...
Mozilla/5.0 (iPhone; CPU iPhone OS 15_5 like Mac OS X)
AppleWebKit/605.1.15 (KHTML, like Gecko) GSA/216.0.453113025
Mobile/15E148 Safari/604.1       1
Mozilla/5.0 (Linux; Android 12; SM-F936U1) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Mobile Safari/537.36
1
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100
Safari/537.36                            1
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_6)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0
Safari/537.36                                1
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/98.0.4758.80 Safari/537.36
1
Name: USERAGENT, Length: 2379, dtype: int64
```

```
test["USERAGENT"].fillna('Mozilla/5.0 (Windows NT 10.0; Win64; x64) App
test.isnull().sum()
```

```
ID                 0
DEVICETYPE         0
PLATFORM_ID        0
BIDREQUESTIP       0
USERPLATFORMUID    0
```

```
USERCITY           0
USERZIPCODE        0
USERAGENT          0
PLATFORMTYPE       0
CHANNELTYPE        0
URL                0
KEYWORDS           0
dtype: int64
```

```
train["TAXONOMY"].value_counts()
```

```
2084P0800X    7930
2084N0400X    6621
207Q00000X    3997
207R00000X    3190
208000000X     837
                ...
163WC0400X       1
246QM0706X       1
1835N1003X       1
207RI0008X       1
207XX0801X       1
Name: TAXONOMY, Length: 207, dtype: int64
```

```
train["TAXONOMY"].fillna('2084P0800X',inplace=True)
train.isnull().sum()
```

```
ID                 0
DEVICETYPE         0
PLATFORM_ID        0
BIDREQUESTIP       0
USERPLATFORMUID    0
USERCITY           0
USERZIPCODE        0
USERAGENT          0
PLATFORMTYPE       0
CHANNELTYPE        0
URL                0
KEYWORDS           0
TAXONOMY           0
IS_HCP             1
dtype: int64
```

```
train["IS_HCP"].value_counts()
```

```
0.0    79756
1.0    34180
Name: IS_HCP, dtype: int64
```

```
train["IS_HCP"].fillna('0',inplace=True)
train.isnull().sum()
```

```
ID                 0
DEVICETYPE         0
PLATFORM_ID        0
BIDREQUESTIP       0
USERPLATFORMUID    0
USERCITY           0
USERZIPCODE        0
USERAGENT          0
PLATFORMTYPE       0
CHANNELTYPE        0
URL                0
KEYWORDS           0
TAXONOMY           0
IS_HCP             0
dtype: int64
```

```
train['IS_HCP']=train["IS_HCP"].apply(np.int64)
```

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

```
0    79757
1    34180
Name: IS_HCP, dtype: int64
```

```
taxonomy=train.iloc[27927:56420,[12]].values
```

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

```
2084P0800X    89554
2084N0400X     6621
207Q00000X     3997
207R00000X     3190
208000000X      837
               ...
163WC0400X        1
246QM0706X        1
1835N1003X        1
207RI0008X        1
207XX0801X        1
Name: TAXONOMY, Length: 207, dtype: int64
```

```
ID=test['ID'].values
ID
```

```
array([115501, 115502, 115503, ..., 143991, 143992, 143993])
```

```
#ID=test['ID'].values
```

```
test.drop(columns=['ID',],inplace=True)

train.drop(columns=['ID',],inplace=True)
```

```
#train.drop(columns=['TAXONOMY',],inplace=True)

train.shape
```

```
(113937, 13)
```

```
train.drop(columns=['USERPLATFORMUID',],inplace=True)

test.drop(columns=['USERPLATFORMUID',],inplace=True)

#train.drop(columns=['PLATFORM_ID',],inplace=True)

#test.drop(columns=['PLATFORM_ID',],inplace=True)
```

```
train.shape
```

```
(113937, 12)
```

```
test.shape
```

```
(28493, 10)
```

```
train["DEVICETYPE"].value_counts()
```

```
Desktop    78423
Mobile     32065
Tablet      3425
Unknown       24
Name: DEVICETYPE, dtype: int64
```

```
train.drop(columns=['USERAGENT',],inplace=True)
```

```
test.drop(columns=['USERAGENT',],inplace=True)
```

```
test.shape
```

```
(28493, 9)
```

```
train.shape
```

```
(113937, 11)
```

```
train
```

	DEVICETYPE	PLATFORM_ID	BIDREQUESTIP	USERCITY	USERZI
0	Desktop	2	170.173.0.22	Portland	97
1	Desktop	2	65.216.253.25	Arlington	22
2	Desktop	2	66.232.79.22	Meadows	83

```
train=train[['PLATFORM_ID','DEVICETYPE','BIDREQUESTIP','USERCITY','USERZ
```

```
train
```

	PLATFORM_ID	DEVICETYPE	BIDREQUESTIP	USERCITY	USERZI
0	2	Desktop	170.173.0.22	Portland	97
1	2	Desktop	65.216.253.25	Arlington	22
2	2	Desktop	66.232.79.22	New Meadows	83
3	3	Desktop	137.54.125.246	New York	229114
4	7	Mobile	174.202.231.99	Houston	77
...	...	...	...	...	
113932	2	Desktop	68.82.97.126	Philadelphia	19
113933	2	Desktop	104.172.11.109	Van Nuys	914014
113934	7	Desktop	174.21.94.113	New York	98
113935	2	Mobile	69.253.129.131	Wilmington	19
113936	2	Mobile	108.41.233.175	White Plains	10

113937 rows × 10 columns

```
for col in train.columns:
  print(col ,': ', len(train[col].unique()),'labels')
```

```
PLATFORM_ID :  15 labels
DEVICETYPE :  4 labels
BIDREQUESTIP :  33664 labels
USERCITY :  4420 labels
USERZIPCODE :  11279 labels
PLATFORMTYPE :  5 labels
CHANNELTYPE :  1 labels
URL :  5231 labels
KEYWORDS :  2460 labels
IS_HCP :  2 labels
```

```
train.DEVICETYPE.value_counts().sort_values(ascending=False).head(20)
```

```
Desktop    78423
Mobile     32065
Tablet      3425
Unknown       24
Name: DEVICETYPE, dtype: int64
```

```
top_10= [x for x in train.DEVICETYPE.value_counts().sort_values(ascendi
top_10
```

```
['Desktop', 'Mobile', 'Tablet', 'Unknown']
```

```
for label in top_10:
  train[label]=np.where(train['DEVICETYPE']==label,1,0)

train[['DEVICETYPE']+top_10].head(40)
```

	DEVICETYPE	Desktop	Mobile	Tablet	Unknown
0	Desktop	1	0	0	0
1	Desktop	1	0	0	0
2	Desktop	1	0	0	0
3	Desktop	1	0	0	0
4	Mobile	0	1	0	0
5	Desktop	1	0	0	0
6	Desktop	1	0	0	0
7	Mobile	0	1	0	0
8	Mobile	0	1	0	0
9	Desktop	1	0	0	0
10	Desktop	1	0	0	0
11	Desktop	1	0	0	0
12	Desktop	1	0	0	0
13	Desktop	1	0	0	0
14	Desktop	1	0	0	0
15	Mobile	0	1	0	0
16	Desktop	1	0	0	0
17	Desktop	1	0	0	0
18	Desktop	1	0	0	0
19	Mobile	0	1	0	0
20	Desktop	1	0	0	0

```
#getting whole set of dummy varibles for all categorcal variables
def one_hot_top_x(df,variable,top_x_labels):

  for label in top_x_labels:
    df[variable+'_'+label]=np.where(train[variable]==label ,1,0)

train=train[['PLATFORM_ID','DEVICETYPE','BIDREQUESTIP','USERCITY','USER

one_hot_top_x(train ,'DEVICETYPE',top_10)
train.head()
```

```
<ipython-input-2764-4433c9b59184>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
  df[variable+'_'+label]=np.where(train[variable]==label ,1,0)
<ipython-input-2764-4433c9b59184>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
  df[variable+'_'+label]=np.where(train[variable]==label ,1,0)
<ipython-input-2764-4433c9b59184>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
  df[variable+'_'+label]=np.where(train[variable]==label ,1,0)
```

	PLATFORM_ID	DEVICETYPE	BIDREQUESTIP	USERCITY	USERZIPCODE
0	2	Desktop	170.173.0.22	Portland	97206.0
1	2	Desktop	65.216.253.25	Arlington	22202.0
				New	

```
#top_10=[x for x in train.PLATFORM_ID.value_counts().sort_values(ascend

#one_hot_top_x(train,'PLATFORM_ID',top_10)
#train.head()
```

```
top_10=[x for x in train.USERCITY.value_counts().sort_values(ascending=

one_hot_top_x(train,'USERCITY',top_10)
train.head()
```

PLATFORM_ID	DEVICETYPE	BIDREQUESTIP	USERCITY	USERZIPCODE

```
top_10=[x for x in train.CHANNELTYPE.value_counts().sort_values(ascendi

one_hot_top_x(train,'CHANNELTYPE',top_10)
train.head()
```

	PLATFORM_ID	DEVICETYPE	BIDREQUESTIP	USERCITY	USERZIPCODE
0	2	Desktop	170.173.0.22	Portland	97206.0
1	2	Desktop	65.216.253.25	Arlington	22202.0
2	2	Desktop	66.232.79.22	New Meadows	83654.0
3	3	Desktop	137.54.125.246	New York	229114624.0
4	7	Mobile	174.202.231.99	Houston	77008.0

5 rows × 25 columns

```
#top_10=[x for x in train.URL.value_counts().sort_values(ascending=False

#one_hot_top_x(train,'URL',top_10)
#train.head()
```

```
top_10=[x for x in train.KEYWORDS.value_counts().sort_values(ascending=

one_hot_top_x(train,'KEYWORDS',top_10)
train.head()
```

	PLATFORM_ID	DEVICETYPE	BIDREQUESTIP	USERCITY	USERZIPCODE
0	2	Desktop	170.173.0.22	Portland	97206.0
1	2	Desktop	65.216.253.25	Arlington	22202.0
2	2	Desktop	66.232.79.22	New Meadows	83654.0
3	3	Desktop	137.54.125.246	New York	229114624.0
4	7	Mobile	174.202.231.99	Houston	77008.0

```
#top_10=[x for x in train.BIDREQUESTIP.value_counts().sort_values(ascend

#one_hot_top_x(train,'BIDREQUESTIP',top_10)
#train.head()
```

```
train.drop(columns=['PLATFORM_ID','DEVICETYPE','USERZIPCODE','BIDREQUEST

train.head
```

```
113934                                                    0
113935                                                    0
113936                                                    0


KEYWORDS_Erythema|Hypersensitivity|Arthritis|Clinical|Bone
Marrow|General|Intravenous|False|Refractory|Liver|Biopsy|Diagnos
Diseases|Chronic|Cardiology|Medicine|Oral|Urology|Flu|Small|Auto
\
0                                                         0
1                                                         0
2                                                         0
3                                                         0
4                                                         0
...                                                     ...
113932                                                    0
113933                                                    0
113934                                                    0
113935                                                    0
113936                                                    0

        KEYWORDS_Bone
Marrow|Radiography|Chronic|Oncology|Psychiatry|Intestine|Small
Intestines|Physicians|Gastroenterology|Salivary
Glands|Autoimmune|Hemolytic|False|Hereditary|Total|Cardiology|Su
Cells|Lung|Biopsy|General|Large|Small|Ophthalmology|Medicine|Lym
Tissue|Rheumatology|B-
```

```
train
```

	IS_HCP	DEVICETYPE_Desktop	DEVICETYPE_Mobile	DEVICETYPE_
0	0	1	0	
1	0	1	0	
2	0	1	0	
3	1	1	0	
4	0	0	1	
...	...	...	...	
113932	1	1	0	
113933	1	1	0	
113934	1	1	0	
113935	1	0	1	
113936	1	0	1	

113937 rows × 25 columns

```
X=train.drop("IS_HCP" ,axis=1)
y=train["IS_HCP"]
```

```
X.shape
X.head()
```

	DEVICETYPE_Desktop	DEVICETYPE_Mobile	DEVICETYPE_Tablet	DEVI(
0	1	0	0	
1	1	0	0	
2	1	0	0	
3	1	0	0	
4	0	1	0	

5 rows × 24 columns

```
y
```

```
0         0
1         0
2         0
3         1
4         0
         ..
113932    1
113933    1
113934    1
113935    1
113936    1
Name: IS_HCP, Length: 113937, dtype: int64
```

```
# separate dataset into train and test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,ra
```

```
X_train.shape, X_test.shape
```

```
((91149, 24), (22788, 24))
```

```
y_train.shape ,y_test.shape
```

```
((91149,), (22788,))
```

```
X_train.corr()
```

KEYWORDS_Family Practice|Drainage|Clir

KEYWORDS_Transplantation|Technology|Chronic|N Defects|Neonatal|Psychological|Stomach|Metabolic|Hospitals|Graves|E

KEYWORDS_Health|Male|Neurological Surgery|Otolary Practice|Abscess|Dentistry|S

KEYWORDS_Myopathies|Small|Rheumatoid|Psychiatry|Urology|

KEYWORDS_Hemoptysis|Oral|Rheumatology|Cardiovascular|Small|Rheum

KEYWORDS_Eryt

KEYWORDS_Bone Marrow|Radiography|Chronic|Oncology|Psychiatry|Int

KEYWORDS_Genetic|Large|Autoimmune|Health|I

KEYWORDS_Health|Medicine|Chronic

24 rows × 24 columns

```
import seaborn as sns
#Using Pearson Correlation
plt.figure(figsize=(12,10))
cor = X_train.corr()
sns.heatmap(cor, annot=True, cmap=plt.cm.CMRmap_r)
plt.show()
```

```
# with the following function we can select highly correlated features
# it will remove the first feature that is correlated with anything othe

def correlation(dataset, threshold):
    col_corr = set()  # Set of all the names of correlated columns
    corr_matrix = dataset.corr()
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if abs(corr_matrix.iloc[i, j]) > threshold: # we are intere
                colname = corr_matrix.columns[i]  # getting the name of
                col_corr.add(colname)
    return col_corr
```

```
corr_features = correlation(X_train, 0.7)
len(set(corr_features))
```

```
1
```

```
corr_features
```

```
{'DEVICETYPE_Mobile'}
```

```
X_train.drop(corr_features,axis=1,inplace=True)
X_test.drop(corr_features,axis=1,inplace=True)
```

```
X_test.shape
```

```
(22788, 23)
```

```
X_train.shape
```

```
(91149, 23)
```

```
X_train
```

	DEVICETYPE_Desktop	DEVICETYPE_Tablet	DEVICETYPE_Unknown
54558	1	0	0
81192	0	1	0
17088	0	0	0
67920	0	0	0
112199	0	0	0
...	...	...	...
76820	1	0	0
110268	1	0	0
103694	1	0	0
860	0	1	0
15795	1	0	0

91149 rows × 23 columns

```
#label encoding for test case
```

```
test
```

	DEVICETYPE	PLATFORM_ID	BIDREQUESTIP	USERCITY	USERZIP
0	Desktop	2	75.189.231.103	Fayetteville	28
1	Mobile	2	24.101.33.158	Conneaut Lake	16
2	Desktop	2	172.118.216.142	Covina	91
3	Desktop	7	71.105.120.171	Brooklyn	11
4	Desktop	2	73.82.211.73	Marietta	30
...	...	...	...	...	
28488	Desktop	2	69.202.233.241	Brooklyn	11
28489	Desktop	7	75.4.190.65	Miami	33
28490	Desktop	7	137.52.180.45	Fort Lauderdale	33
28491	Desktop	8	66.249.66.4	New York	63
28492	Desktop	2	107.194.33.149	Wilmington	28

28493 rows × 9 columns

```
test=test[['PLATFORM_ID','DEVICETYPE','BIDREQUESTIP','USERCITY','USERZI
```

```
test
```

	PLATFORM_ID	DEVICETYPE	BIDREQUESTIP	USERCITY	USERZIP
0	2	Desktop	75.189.231.103	Fayetteville	283
1	2	Mobile	24.101.33.158	Conneaut Lake	163
2	2	Desktop	172.118.216.142	Covina	917
3	7	Desktop	71.105.120.171	Brooklyn	112
4	2	Desktop	73.82.211.73	Marietta	30(
...	...	...	...	...	
28488	2	Desktop	69.202.233.241	Brooklyn	112
28489	7	Desktop	75.4.190.65	Miami	331
28490	7	Desktop	137.52.180.45	Fort	333

```
for col in test.columns:
  print(col ,': ', len(test[col].unique()),'labels')
```

```
PLATFORM_ID :  14 labels
DEVICETYPE :  4 labels
BIDREQUESTIP :  14255 labels
USERCITY :  3060 labels
USERZIPCODE :  7038 labels
PLATFORMTYPE :  4 labels
CHANNELTYPE :  1 labels
URL :  2740 labels
KEYWORDS :  1685 labels
```

```
test.DEVICETYPE.value_counts().sort_values(ascending=False).head(20)
```

```
Desktop    19731
Mobile      7946
Tablet       810
Unknown        6
Name: DEVICETYPE, dtype: int64
```

```
top_10= [x for x in test.DEVICETYPE.value_counts().sort_values(ascending
top_10
```

```
['Desktop', 'Mobile', 'Tablet', 'Unknown']
```

```
for label in top_10:
  test[label]=np.where(test['DEVICETYPE']==label,1,0)

test[['DEVICETYPE']+top_10].head(40)
```

```
<ipython-input-2795-0f3f0d72c3f8>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
  test[label]=np.where(test['DEVICETYPE']==label,1,0)
<ipython-input-2795-0f3f0d72c3f8>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
  test[label]=np.where(test['DEVICETYPE']==label,1,0)
<ipython-input-2795-0f3f0d72c3f8>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
  test[label]=np.where(test['DEVICETYPE']==label,1,0)
<ipython-input-2795-0f3f0d72c3f8>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
  test[label]=np.where(test['DEVICETYPE']==label,1,0)
```

	DEVICETYPE	Desktop	Mobile	Tablet	Unknown
0	Desktop	1	0	0	0
1	Mobile	0	1	0	0
2	Desktop	1	0	0	0
3	Desktop	1	0	0	0
4	Desktop	1	0	0	0
5	Desktop	1	0	0	0
6	Desktop	1	0	0	0
8	Desktop	1	0	0	0

```
#getting whole set of dummy varibles for all categorcal variables
def one_hot_top_x(df,variable,top_x_labels):

  for label in top_x_labels:
    df[variable+'_'+label]=np.where(test[variable]==label ,1,0)

test=test[['PLATFORM_ID','DEVICETYPE','BIDREQUESTIP','USERCITY','USERZI

one_hot_top_x(test ,'DEVICETYPE',top_10)
test.head()
```

```
<ipython-input-2796-6d06283231a9>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
  df[variable+'_'+label]=np.where(test[variable]==label ,1,0)
<ipython-input-2796-6d06283231a9>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
  df[variable+'_'+label]=np.where(test[variable]==label ,1,0)
<ipython-input-2796-6d06283231a9>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
  df[variable+'_'+label]=np.where(test[variable]==label ,1,0)
```

	PLATFORM_ID	DEVICETYPE	BIDREQUESTIP	USERCITY	USERZIPCODE
0	2	Desktop	75.189.231.103	Fayetteville	28305.0
1	2	Mobile	24.101.33.158	Conneaut Lake	16316.0

```
#top_10=[x for x in test.PLATFORM_ID.value_counts().sort_values(ascendir

#one_hot_top_x(test,'PLATFORM_ID',top_10)
#test.head()
```

```
top_10=[x for x in test.USERCITY.value_counts().sort_values(ascending=Fa

one_hot_top_x(test,'USERCITY',top_10)
test.head()
```

	PLATFORM_ID	DEVICETYPE	BIDREQUESTIP	USERCITY	USERZIPCODE
0	2	Desktop	75.189.231.103	Fayetteville	28305.0
1	2	Mobile	24.101.33.158	Conneaut Lake	16316.0

```
top_10=[x for x in test.CHANNELTYPE.value_counts().sort_values(ascending

one_hot_top_x(test,'CHANNELTYPE',top_10)
test.head()
```

	PLATFORM_ID	DEVICETYPE	BIDREQUESTIP	USERCITY	USERZIPCODE
0	2	Desktop	75.189.231.103	Fayetteville	28305.0
1	2	Mobile	24.101.33.158	Conneaut Lake	16316.0
2	2	Desktop	172.118.216.142	Covina	91724.0
3	7	Desktop	71.105.120.171	Brooklyn	11226.0
4	2	Desktop	73.82.211.73	Marietta	30062.0

5 rows × 24 columns

```
#top_10=[x for x in test.BIDREQUESTIP.value_counts().sort_values(ascend

#one_hot_top_x(test,'BIDREQUESTIP',top_10)
#test.head()
```

```
#top_10=[x for x in test.URL.value_counts().sort_values(ascending=False

#one_hot_top_x(test,'URL',top_10)
#test.head()
```

```
top_10=[x for x in test.KEYWORDS.value_counts().sort_values(ascending=Fa

one_hot_top_x(test,'KEYWORDS',top_10)
test.head()
```

	PLATFORM_ID	DEVICETYPE	BIDREQUESTIP	USERCITY	USERZIPCODE
0	2	Desktop	75.189.231.103	Fayetteville	28305.0
1	2	Mobile	24.101.33.158	Conneaut Lake	16316.0
2	2	Desktop	172.118.216.142	Covina	91724.0
3	7	Desktop	71.105.120.171	Brooklyn	11226.0
4	2	Desktop	73.82.211.73	Marietta	30062.0

5 rows × 34 columns

```
test.drop(columns=['PLATFORM_ID','BIDREQUESTIP','DEVICETYPE','USERZIPCOD

test
```

	DEVICETYPE_Desktop	DEVICETYPE_Mobile	DEVICETYPE_Tablet	[
0	1	0	0	
3	1	0	0	
...	...	...	...	
28490	1	0	0	

```
test.drop(corr_features,axis=1,inplace=True)
```

```
test.shape
```

```
(28493, 23)
```

```
xfinal=X_train
xfinal.values
```

```
array([[1, 0, 0, ..., 0, 0, 0],
       [0, 1, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [1, 0, 0, ..., 0, 0, 0],
       [0, 1, 0, ..., 0, 0, 0],
       [1, 0, 0, ..., 0, 0, 1]])
```

```
yfinal=y_train
yfinal.values
```

```
yfinal.shape
```

```
(91149,)
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test=train_test_split(xfinal,yfinal,test_si
```

```
model=RandomForestClassifier()
model.fit(X_train,y_train)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
y_pred=model.predict(X_test)
```

```
y_pred.shape
```

```
(18230,)
```

```
y_pred
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

```
y_test.shape
```

```
(18230,)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred )
```

```
0.7386176631925397
```

```
test
```

	DEVICETYPE_Desktop	DEVICETYPE_Tablet	DEVICETYPE_Unknown
0	1	0	0
1	0	0	0
2	1	0	0
3	1	0	0
4	1	0	0
...	...	...	...
28488	1	0	0
28489	1	0	0
28490	1	0	0
28491	1	0	0
28492	1	0	0

28493 rows × 23 columns

```
Xf=test
Xf.values
```

```
array([[1, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [1, 0, 0, ..., 0, 0, 0],
       ...,
```

```
       [1, 0, 0, ..., 0, 0, 0],
       [1, 0, 0, ..., 0, 0, 0],
       [1, 0, 0, ..., 0, 0, 0]])
```

```
y_final=model.predict(Xf.values)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserW
  warnings.warn(
```

```
y_final.shape
```

```
(28493,)
```

```
ID.shape
```

```
(28493,)
```

```
Scoring_Solution=pd.DataFrame()
Scoring_Solution
```

```
Scoring_Solution['ID']=ID
Scoring_Solution['IS_HCP']=y_final
```

```
Scoring_Solution['IS_HCP'].value_counts()
```

```
0    26476
1     2017
Name: IS_HCP, dtype: int64
```

```
Scoring_Solution
```

	ID	IS_HCP
0	115501	0

```
Scoring_Solution.to_csv('scoringSolution.csv',index=False)
```

```
Complete_Solution=pd.DataFrame()
Complete_Solution
```

```
Complete_Solution['ID']=ID
Complete_Solution['IS_HCP']=y_final
Complete_Solution['taxonomy']=taxonomy
```

```
Complete_Solution['IS_HCP'].value_counts()
```

```
0    26476
1     2017
Name: IS_HCP, dtype: int64
```

```
Complete_Solution.to_csv('CompleteSolution.csv',index=False)
```