In [1]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

In [2]:
```
!pip install phik
```

```
Requirement already satisfied: phik in /usr/local/lib/python3.7/dist-packages (0.11.1)
Requirement already satisfied: matplotlib>=2.2.3 in /usr/local/lib/python3.7/dist-packages (from phik) (3.2.2)
Requirement already satisfied: scipy>=1.5.2 in /usr/local/lib/python3.7/dist-packages (from phik) (1.6.1)
Requirement already satisfied: pandas>=0.25.1 in /usr/local/lib/python3.7/dist-packages (from phik) (1.1.5)
Requirement already satisfied: numpy>=1.18.0 in /usr/local/lib/python3.7/dist-packages (from phik) (1.19.5)
Requirement already satisfied: joblib>=0.14.1 in /usr/local/lib/python3.7/dist-packages (from phik) (1.0.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.2.3->phik) (2.4.7)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.2.3->phik) (1.3.1)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.2.3->phik) (2.8.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.2.3->phik) (0.10.0)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.25.1->phik) (2018.9)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.1->matplotlib>=2.2.3->phik) (1.15.0)
```

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
%matplotlib inline
from scipy import stats
import warnings
warnings.filterwarnings('ignore')
import matplotlib.gridspec as gridspec

import itertools
import phik
from phik import resources
from phik.binning import bin_data
from phik.report import plot_correlation_matrix
%matplotlib inline
```

In [4]:
```
data = df = pd.read_csv("/content/drive/MyDrive/case study 1/aps_failure_training_set.c
data.head()
```

Out[4]:

	class	aa_000	ab_000	ac_000	ad_000	ae_000	af_000	ag_000	ag_001	ag_002	ag_003	ag_
0	neg	76698	NaN	2.130706e+09	280.0	0.0	0.0	0.0	0.0	0.0	0.0	372
1	neg	33058	NaN	0.000000e+00	NaN	0.0	0.0	0.0	0.0	0.0	0.0	182

	class	aa_000	ab_000	ac_000	ad_000	ae_000	af_000	ag_000	ag_001	ag_002	ag_003	ag_
2	neg	41040	NaN	2.280000e+02	100.0	0.0	0.0	0.0	0.0	0.0	0.0	16
3	neg	12	0.0	7.000000e+01	66.0	0.0	10.0	0.0	0.0	0.0	318.0	22
4	neg	60874	NaN	1.368000e+03	458.0	0.0	0.0	0.0	0.0	0.0	0.0	437

5 rows × 171 columns

In [5]:
```
data.shape
```

Out[5]: (60000, 171)

In [6]:
```
data.describe()
```

Out[6]:

	aa_000	ab_000	ac_000	ad_000	ae_000	af_000	ag_0
count	6.000000e+04	13671.000000	5.666500e+04	4.513900e+04	57500.000000	57500.000000	5.932900e+
mean	5.933650e+04	0.713189	3.560143e+08	1.906206e+05	6.819130	11.006817	2.216364e+
std	1.454301e+05	3.478962	7.948749e+08	4.040441e+07	161.543373	209.792592	2.047846e+
min	0.000000e+00	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000e+
25%	8.340000e+02	0.000000	1.600000e+01	2.400000e+01	0.000000	0.000000	0.000000e+
50%	3.077600e+04	0.000000	1.520000e+02	1.260000e+02	0.000000	0.000000	0.000000e+
75%	4.866800e+04	0.000000	9.640000e+02	4.300000e+02	0.000000	0.000000	0.000000e+
max	2.746564e+06	204.000000	2.130707e+09	8.584298e+09	21050.000000	20070.000000	3.376892e+

8 rows × 170 columns

In [7]:
```
print("Columns name :\n",list(data.columns))
```

```
Columns name :
 ['class', 'aa_000', 'ab_000', 'ac_000', 'ad_000', 'ae_000', 'af_000', 'ag_000', 'ag_00
1', 'ag_002', 'ag_003', 'ag_004', 'ag_005', 'ag_006', 'ag_007', 'ag_008', 'ag_009', 'ah_
000', 'ai_000', 'aj_000', 'ak_000', 'al_000', 'am_0', 'an_000', 'ao_000', 'ap_000', 'aq_
000', 'ar_000', 'as_000', 'at_000', 'au_000', 'av_000', 'ax_000', 'ay_000', 'ay_001', 'a
y_002', 'ay_003', 'ay_004', 'ay_005', 'ay_006', 'ay_007', 'ay_008', 'ay_009', 'az_000',
'az_001', 'az_002', 'az_003', 'az_004', 'az_005', 'az_006', 'az_007', 'az_008', 'az_00
9', 'ba_000', 'ba_001', 'ba_002', 'ba_003', 'ba_004', 'ba_005', 'ba_006', 'ba_007', 'ba_
008', 'ba_009', 'bb_000', 'bc_000', 'bd_000', 'be_000', 'bf_000', 'bg_000', 'bh_000', 'b
i_000', 'bj_000', 'bk_000', 'bl_000', 'bm_000', 'bn_000', 'bo_000', 'bp_000', 'bq_000',
'br_000', 'bs_000', 'bt_000', 'bu_000', 'bv_000', 'bx_000', 'by_000', 'bz_000', 'ca_00
0', 'cb_000', 'cc_000', 'cd_000', 'ce_000', 'cf_000', 'cg_000', 'ch_000', 'ci_000', 'cj_
000', 'ck_000', 'cl_000', 'cm_000', 'cn_000', 'cn_001', 'cn_002', 'cn_003', 'cn_004', 'c
n_005', 'cn_006', 'cn_007', 'cn_008', 'cn_009', 'co_000', 'cp_000', 'cq_000', 'cr_000',
'cs_000', 'cs_001', 'cs_002', 'cs_003', 'cs_004', 'cs_005', 'cs_006', 'cs_007', 'cs_00
8', 'cs_009', 'ct_000', 'cu_000', 'cv_000', 'cx_000', 'cy_000', 'cz_000', 'da_000', 'db_
000', 'dc_000', 'dd_000', 'de_000', 'df_000', 'dg_000', 'dh_000', 'di_000', 'dj_000', 'd
k_000', 'dl_000', 'dm_000', 'dn_000', 'do_000', 'dp_000', 'dq_000', 'dr_000', 'ds_000',
```

```
'dt_000', 'du_000', 'dv_000', 'dx_000', 'dy_000', 'dz_000', 'ea_000', 'eb_000', 'ec_00',
'ed_000', 'ee_000', 'ee_001', 'ee_002', 'ee_003', 'ee_004', 'ee_005', 'ee_006', 'ee_00
7', 'ee_008', 'ee_009', 'ef_000', 'eg_000']
```

In [8]:

```
def missing_values_table(df):
        # Total missing values
        mis_val = df.isna().sum()

        # Percentage of missing values
        mis_val_percent = 100 * df.isnull().sum() / len(df)

        # Make a table with the results
        mis_val_table = pd.concat([mis_val, mis_val_percent], axis=1)

        # Rename the columns
        mis_val_table_ren_columns = mis_val_table.rename(
        columns = {0 : 'Missing Values', 1 : '% of Total Values'})

        # Sort the table by percentage of missing descending
        mis_val_table_ren_columns = mis_val_table_ren_columns[
            mis_val_table_ren_columns.iloc[:,1] != 0].sort_values(
        '% of Total Values', ascending=False).round(1)

        # Print some summary information
        print ("Your selected dataframe has " + str(df.shape[1]) + " columns.\n"
            "There are " + str(mis_val_table_ren_columns.shape[0]) +
              " columns that have missing values.")

        # Return the dataframe with missing information
        return mis_val_table_ren_columns
```

In [9]:

```
data_missing =  missing_values_table(data)
data_missing.head(20)
```

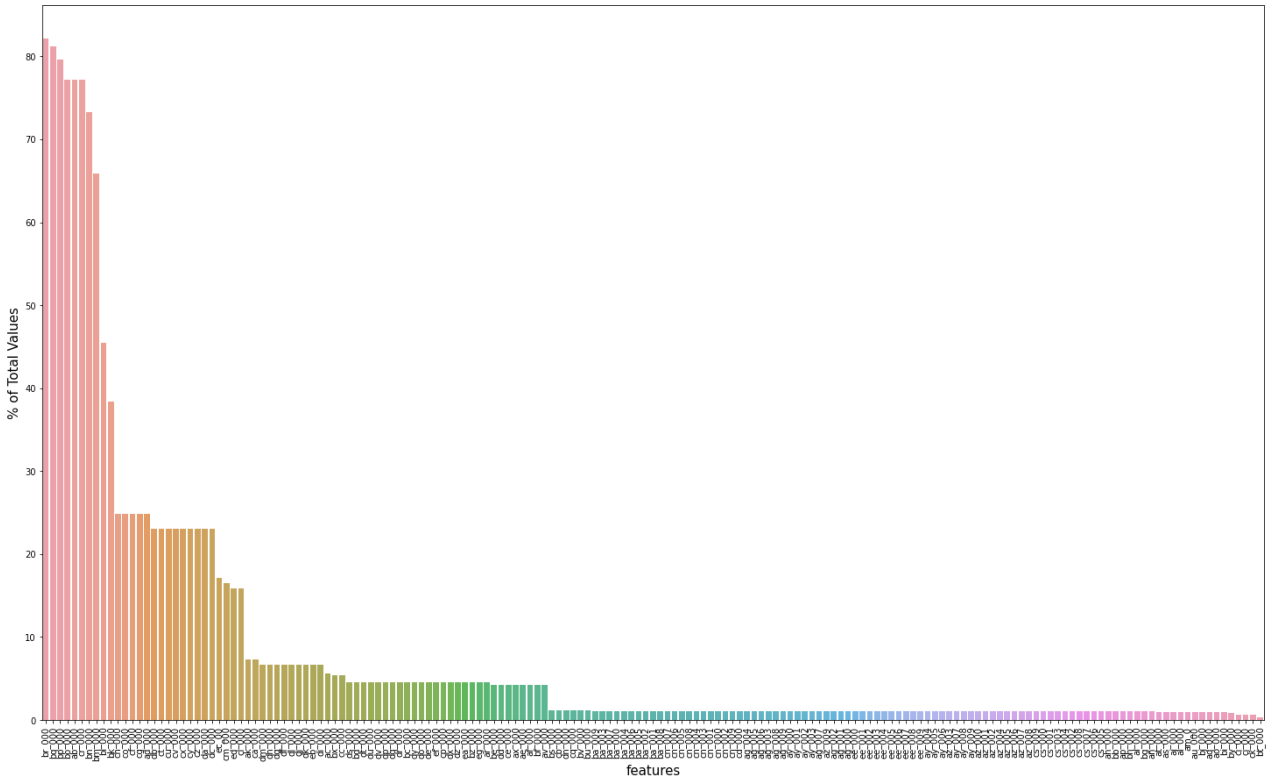
```
Your selected dataframe has 171 columns.
There are 169 columns that have missing values.
```

Out[9]:

	Missing Values	% of Total Values
br_000	49264	82.1
bq_000	48722	81.2
bp_000	47740	79.6
bo_000	46333	77.2
ab_000	46329	77.2
cr_000	46329	77.2
bn_000	44009	73.3
bm_000	39549	65.9
bl_000	27277	45.5
bk_000	23034	38.4
ch_000	14861	24.8
co_000	14861	24.8

	Missing Values	% of Total Values
cf_000	14861	24.8
cg_000	14861	24.8
ad_000	14861	24.8
db_000	13808	23.0
ct_000	13808	23.0
cu_000	13808	23.0
cv_000	13808	23.0
cx_000	13808	23.0

In [10]:
```
plt.figure(figsize = (25,15))
sns.barplot(data_missing.index, data_missing['% of Total Values'], alpha = 0.9,order=da
plt.xticks(rotation = 'vertical')
plt.xlabel('features', fontsize =15)
plt.ylabel('% of Total Values', fontsize = 15)
plt.show()
plt.draw()
```



<Figure size 432x288 with 0 Axes>

In [11]:
```
data = data_drop = data.drop(['br_000','bq_000','bp_000','bo_000','ab_000','cr_000','bn
```

In [12]:
```
data.shape
```

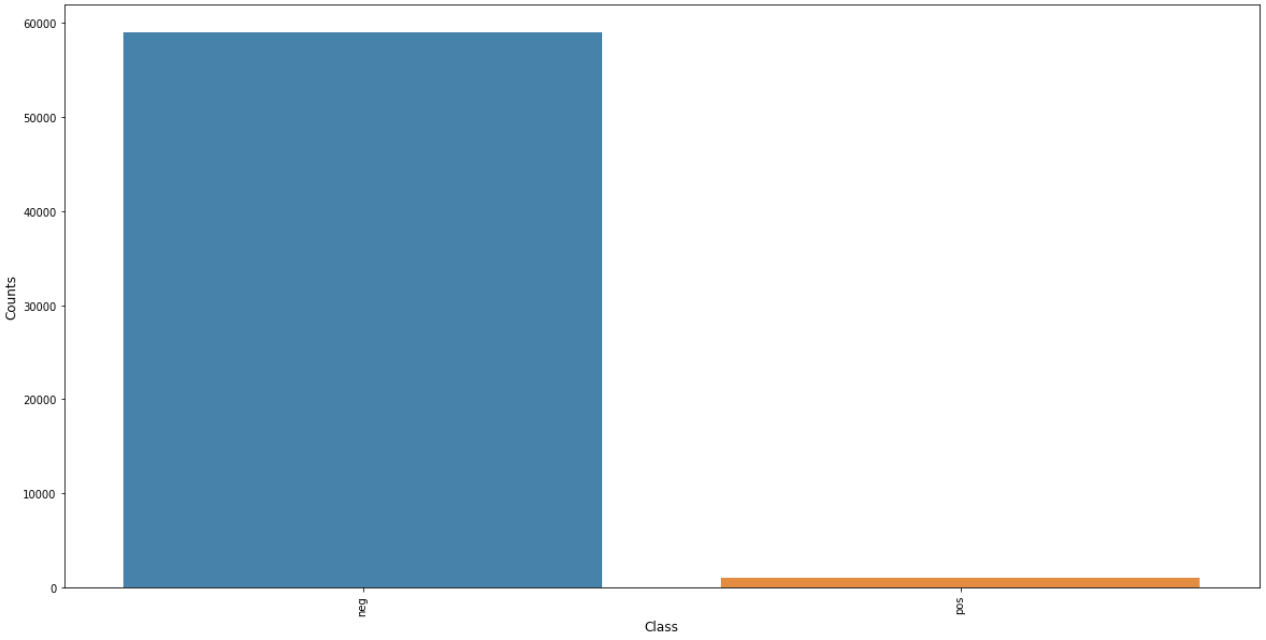
Out[12]: (60000, 163)

In [13]:
```
col = list(data.columns)
```

In [14]:
```
print ('The train data has {} unique labels'.format(df['class'].nunique()))
```

```
The train data has 2 unique labels
```

In [15]:
```
label_counts = df['class'].value_counts()
plt.figure(figsize = (20,10))
sns.barplot(label_counts.index, label_counts.values, alpha = 0.9)
plt.xticks(rotation = 'vertical')
plt.xlabel('Class', fontsize =12)
plt.ylabel('Counts', fontsize = 12)
plt.show()
```



In [16]:
```
print("Number of positive classes = ", sum(df['class'] == 'pos'))
print("Number of negative classes = ", sum(df['class'] == 'neg'))
```

```
Number of positive classes =  1000
Number of negative classes =  59000
```

In [17]:
```
X = data.drop("class",axis=1)
y = data['class']
```

In [18]:
```
data['class'] = data['class'].apply(lambda x: 0 if x == 'neg' else 1)
data.head()
```

Out[18]:

	class	aa_000	ac_000	ad_000	ae_000	af_000	ag_000	ag_001	ag_002	ag_003	ag_004	
0	0	76698	2.130706e+09	280.0	0.0	0.0	0.0	0.0	0.0	0.0	37250.0	143
1	0	33058	0.000000e+00	NaN	0.0	0.0	0.0	0.0	0.0	0.0	18254.0	65
2	0	41040	2.280000e+02	100.0	0.0	0.0	0.0	0.0	0.0	0.0	1648.0	37

	class	aa_000	ac_000	ad_000	ae_000	af_000	ag_000	ag_001	ag_002	ag_003	ag_004	a
3	0	12	7.000000e+01	66.0	0.0	10.0	0.0	0.0	0.0	318.0	2212.0	
4	0	60874	1.368000e+03	458.0	0.0	0.0	0.0	0.0	0.0	0.0	43752.0	196

5 rows × 163 columns

In [19]:
```
correlation_train = data.corr()
corr_dict = correlation_train['class'].sort_values(ascending=False).to_dict()
important_columns=[]
for key,value in corr_dict.items():
    if ((value>0.4) & (value<0.9)) | (value<=-0.4):
        important_columns.append(key)
```

In [20]:
```
len(important_columns)
```

Out[20]: 43

In [21]:
```
correlation_train = data.phik_matrix()
corr_dict = correlation_train['class'].sort_values(ascending=False).to_dict()
important_columns2 = []
for key,value in corr_dict.items():
    if ((value>0.5) & (value<0.9)):
        important_columns2.append(key)
```

```
interval columns not set, guessing: ['class', 'aa_000', 'ac_000', 'ad_000', 'ae_000', 'a
f_000', 'ag_000', 'ag_001', 'ag_002', 'ag_003', 'ag_004', 'ag_005', 'ag_006', 'ag_007',
'ag_008', 'ag_009', 'ah_000', 'ai_000', 'aj_000', 'ak_000', 'al_000', 'am_0', 'an_000',
'ao_000', 'ap_000', 'aq_000', 'ar_000', 'as_000', 'at_000', 'au_000', 'av_000', 'ax_00
0', 'ay_000', 'ay_001', 'ay_002', 'ay_003', 'ay_004', 'ay_005', 'ay_006', 'ay_007', 'ay_
008', 'ay_009', 'az_000', 'az_001', 'az_002', 'az_003', 'az_004', 'az_005', 'az_006', 'a
z_007', 'az_008', 'az_009', 'ba_000', 'ba_001', 'ba_002', 'ba_003', 'ba_004', 'ba_005',
'ba_006', 'ba_007', 'ba_008', 'ba_009', 'bb_000', 'bc_000', 'bd_000', 'be_000', 'bf_00
0', 'bg_000', 'bh_000', 'bi_000', 'bj_000', 'bk_000', 'bl_000', 'bs_000', 'bt_000', 'bu_
000', 'bv_000', 'bx_000', 'by_000', 'bz_000', 'ca_000', 'cb_000', 'cc_000', 'cd_000', 'c
e_000', 'cf_000', 'cg_000', 'ch_000', 'ci_000', 'cj_000', 'ck_000', 'cl_000', 'cm_000',
'cn_000', 'cn_001', 'cn_002', 'cn_003', 'cn_004', 'cn_005', 'cn_006', 'cn_007', 'cn_00
8', 'cn_009', 'co_000', 'cp_000', 'cq_000', 'cs_000', 'cs_001', 'cs_002', 'cs_003', 'cs_
004', 'cs_005', 'cs_006', 'cs_007', 'cs_008', 'cs_009', 'ct_000', 'cu_000', 'cv_000', 'c
x_000', 'cy_000', 'cz_000', 'da_000', 'db_000', 'dc_000', 'dd_000', 'de_000', 'df_000',
'dg_000', 'dh_000', 'di_000', 'dj_000', 'dk_000', 'dl_000', 'dm_000', 'dn_000', 'do_00
0', 'dp_000', 'dq_000', 'dr_000', 'ds_000', 'dt_000', 'du_000', 'dv_000', 'dx_000', 'dy_
000', 'dz_000', 'ea_000', 'eb_000', 'ec_00', 'ed_000', 'ee_000', 'ee_001', 'ee_002', 'ee
_003', 'ee_004', 'ee_005', 'ee_006', 'ee_007', 'ee_008', 'ee_009', 'ef_000', 'eg_000']
```

In [22]:
```
len(important_columns2)
```

Out[22]: 29

In [23]:
```
important_columns2
```

Out[23]:
```
['ci_000',
```

```
'aq_000',
'bb_000',
'cq_000',
'bu_000',
'bv_000',
'bj_000',
'bt_000',
'aa_000',
'cc_000',
'ao_000',
'bx_000',
'bh_000',
'ap_000',
'ck_000',
'dn_000',
'by_000',
'ee_005',
'dt_000',
'bi_000',
'ba_005',
'cn_004',
'ed_000',
'an_000',
'ah_000',
'bg_000',
'ee_000',
'ee_006',
'ay_008']
```

In [24]:
```
# Univariate analysis
```

In [25]:
```
df.head()
```

Out[25]:

	class	aa_000	ab_000	ac_000	ad_000	ae_000	af_000	ag_000	ag_001	ag_002	ag_003	ag_
0	neg	76698	NaN	2.130706e+09	280.0	0.0	0.0	0.0	0.0	0.0	0.0	372
1	neg	33058	NaN	0.000000e+00	NaN	0.0	0.0	0.0	0.0	0.0	0.0	182
2	neg	41040	NaN	2.280000e+02	100.0	0.0	0.0	0.0	0.0	0.0	0.0	16
3	neg	12	0.0	7.000000e+01	66.0	0.0	10.0	0.0	0.0	0.0	318.0	22
4	neg	60874	NaN	1.368000e+03	458.0	0.0	0.0	0.0	0.0	0.0	0.0	437

5 rows × 171 columns

In [26]:
```
data = df
```

Univariate analysis

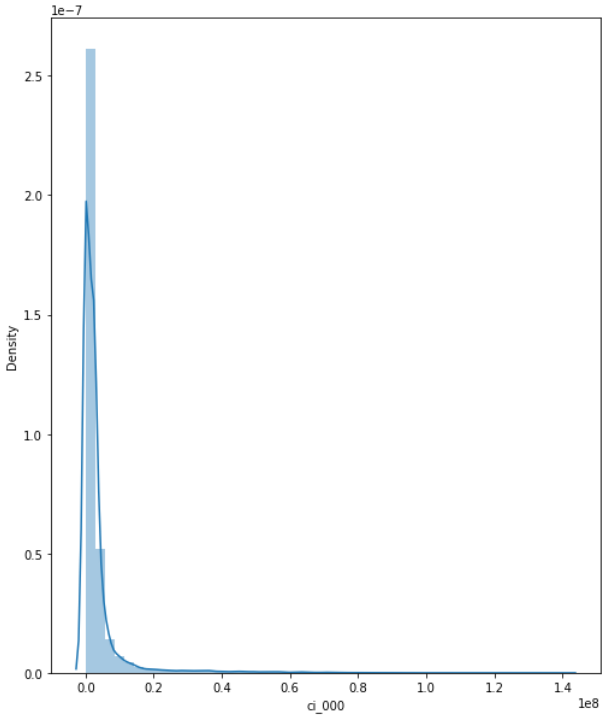
In [27]:
```
def EDA(col,a):
    fig, axes = plt.subplots(1, 2, figsize=(18, 10), sharex=True)
    sns.distplot(col,ax=axes[0])
    sns.boxplot(x =col, y = 'class', data = data,ax=axes[1])
```

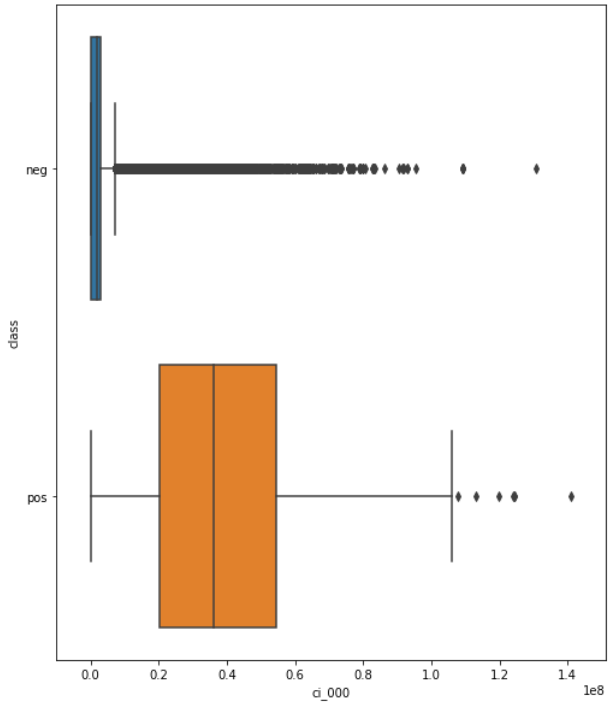
```
    sns.FacetGrid(data, hue="class", size=5).map(sns.distplot,str(a)).add_legend();
    print('Skew Dist    :',col.skew())
    print('Kurtosis Dist:',col.kurt())
    print("Mean         :",np.mean(col))
    print("Std-dev      :",np.std(col));
```

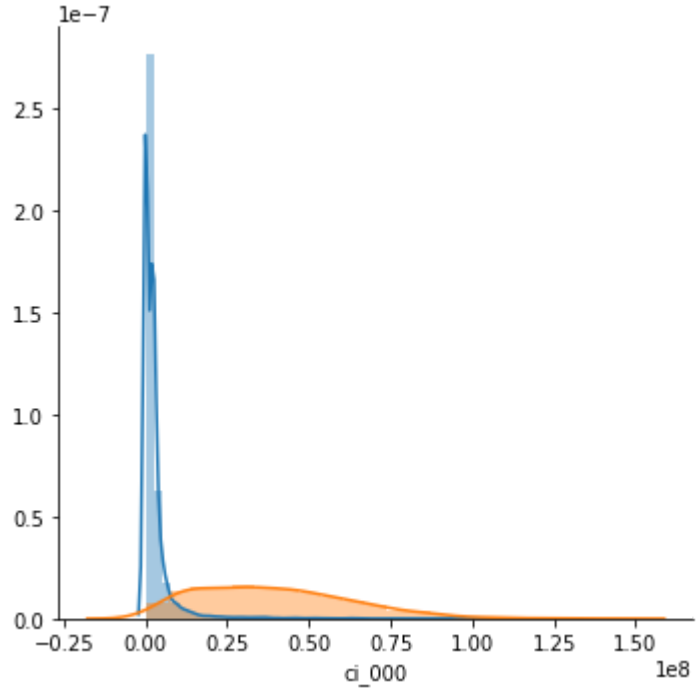
In [28]:

```
EDA(data.ci_000,'ci_000')
```

```
Skew Dist    : 5.836331522733422
Kurtosis Dist: 43.01384144955171
Mean         : 3481204.0487352014
Std-dev      : 8355926.9320374
```







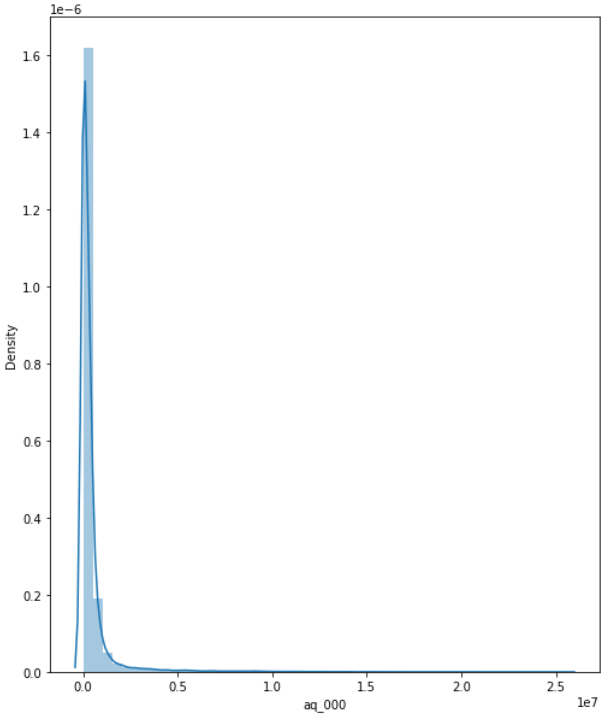
Observation

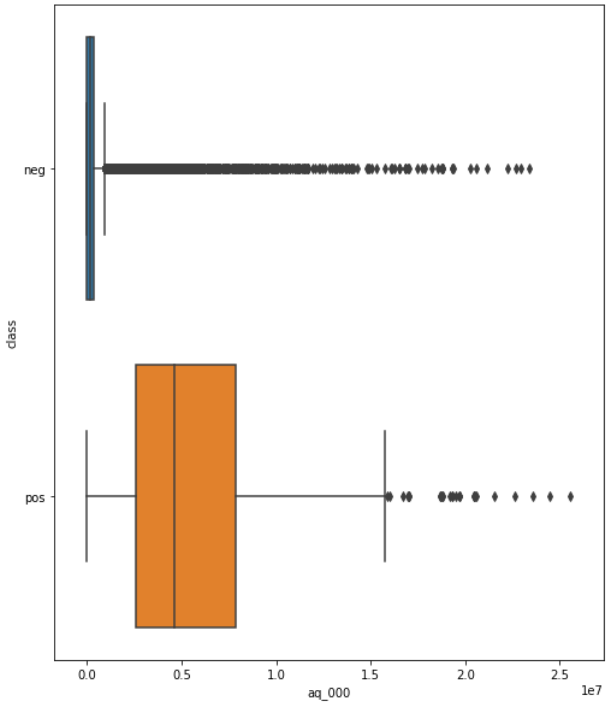
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class

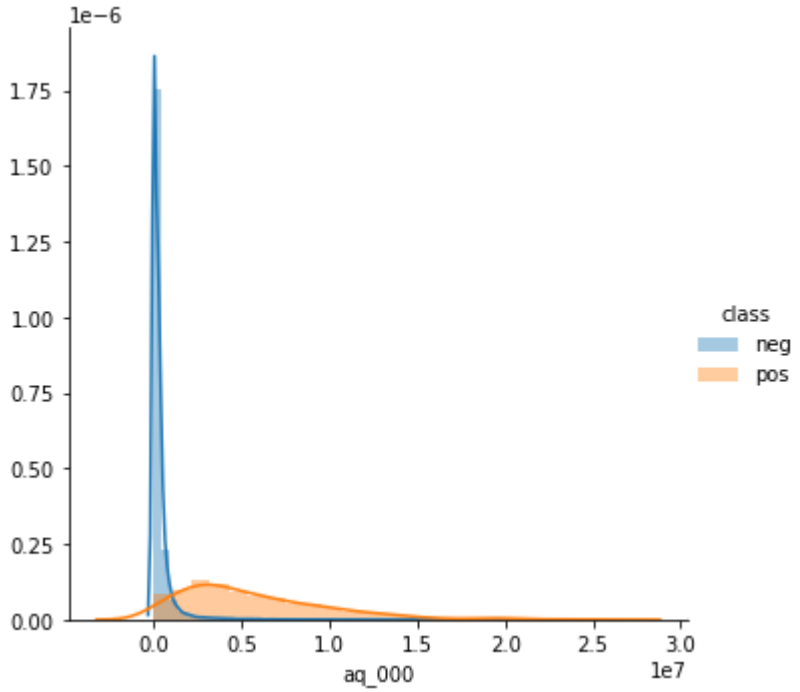
In [28]:

In [29]:
```
EDA(data.aq_000,'aq_000')
```

```
Skew Dist    : 7.87783819933435
Kurtosis Dist: 83.41271212149684
Mean         : 442404.46432478837
Std-dev      : 1262458.2697600161
```





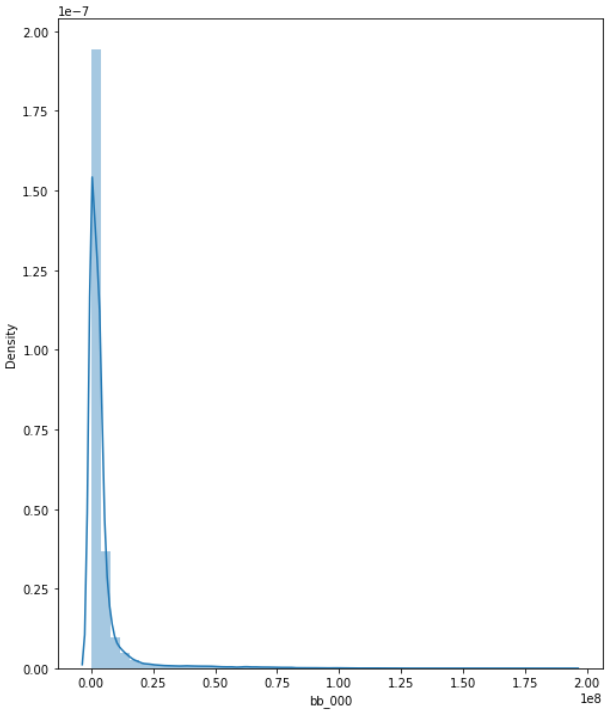


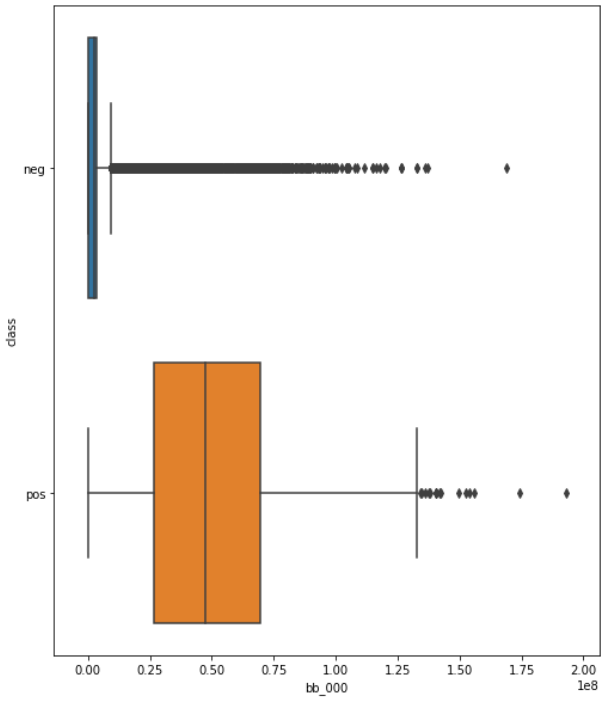
Observation

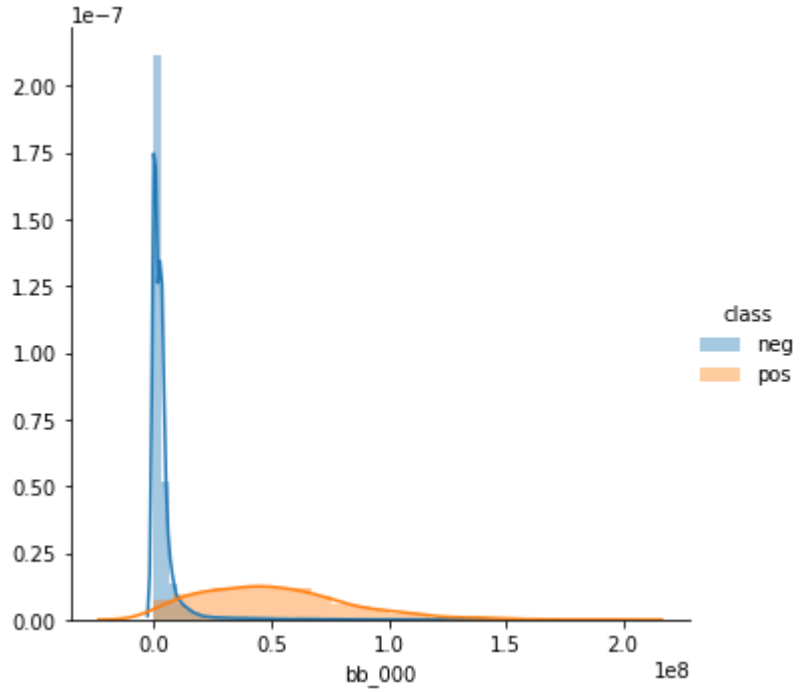
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class

In [30]:
```
EDA(data.bb_000,'bb_000')
```

```
Skew Dist    : 5.870093771169788
Kurtosis Dist: 43.654278149238614
Mean         : 4526177.187802207
Std-dev      : 10886644.92195377
```





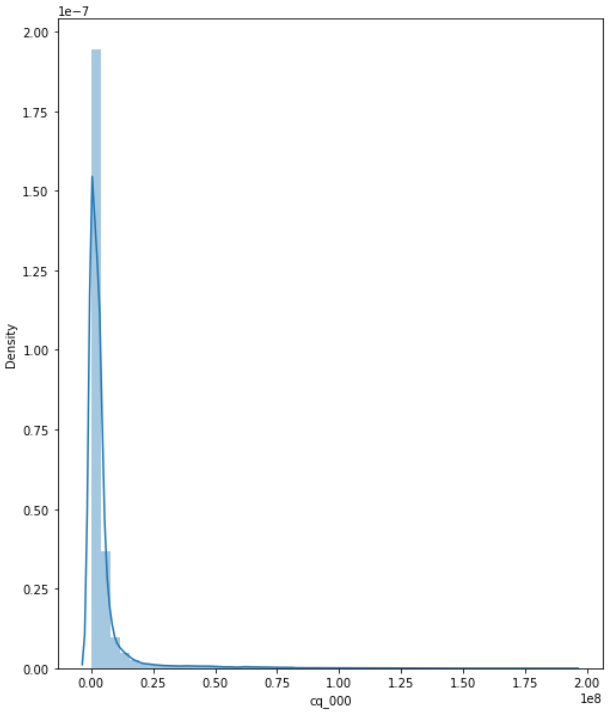


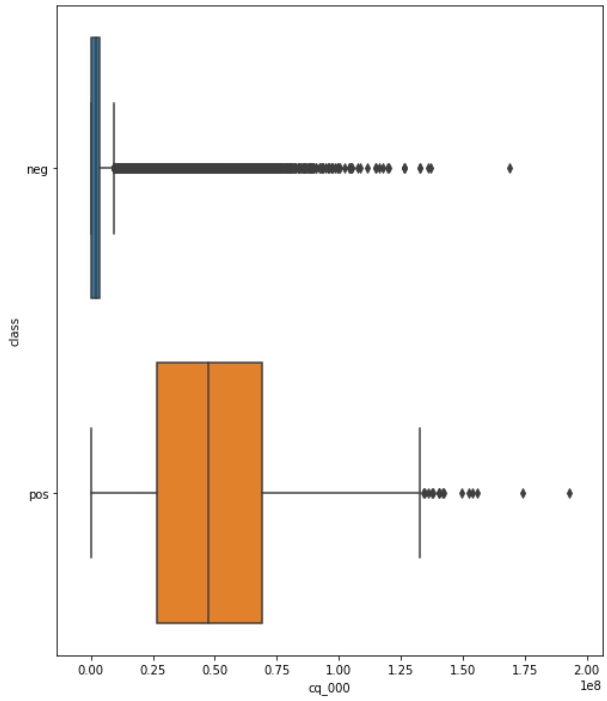
Observation

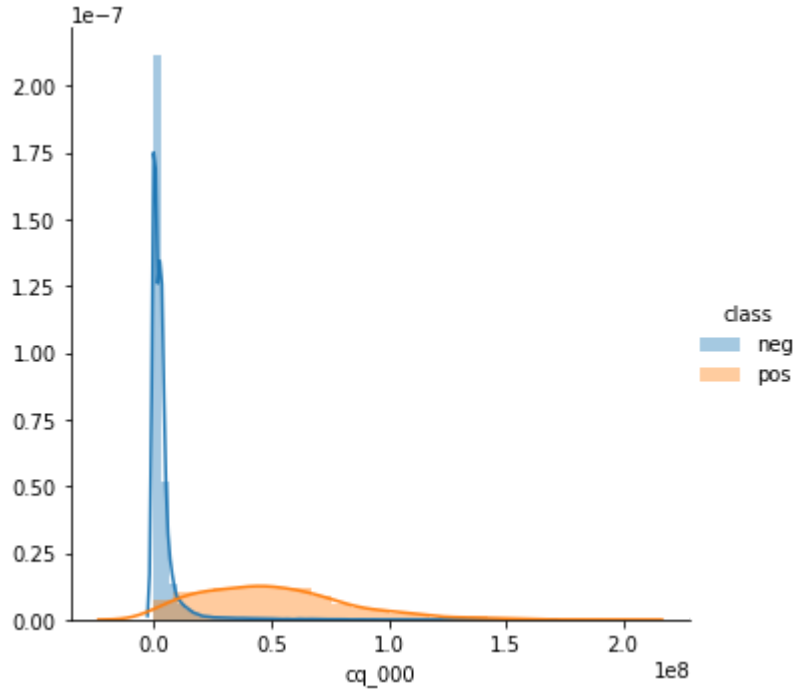
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class

In [31]:
```
EDA(data.cq_000,'cq_000')
```

```
Skew Dist    : 5.87814069028433
Kurtosis Dist: 43.82909379803299
Mean         : 4515325.180023268
Std-dev      : 10859812.536692042
```





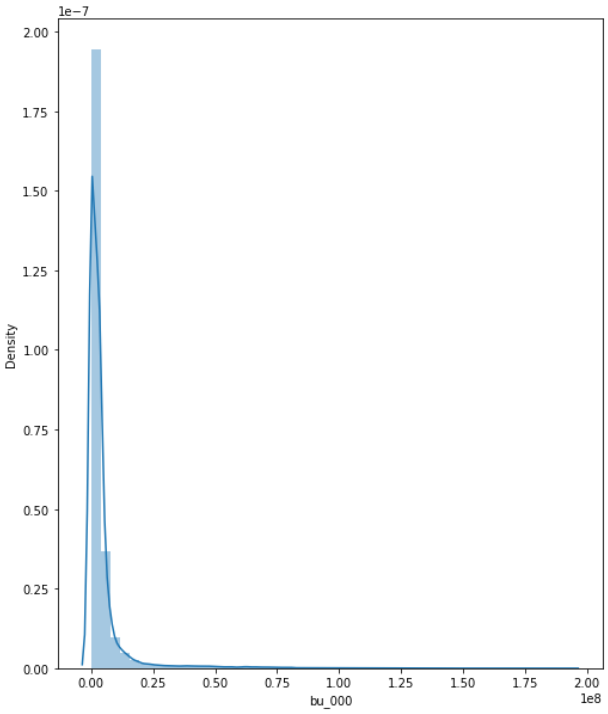


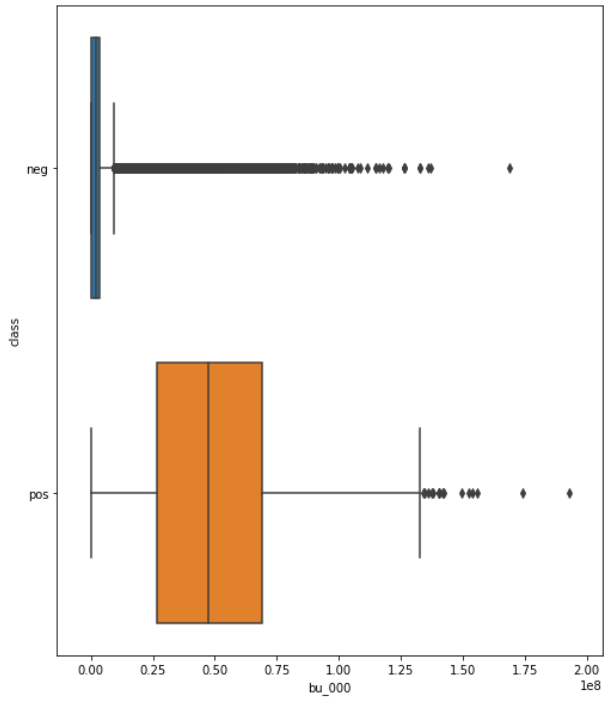
Observation

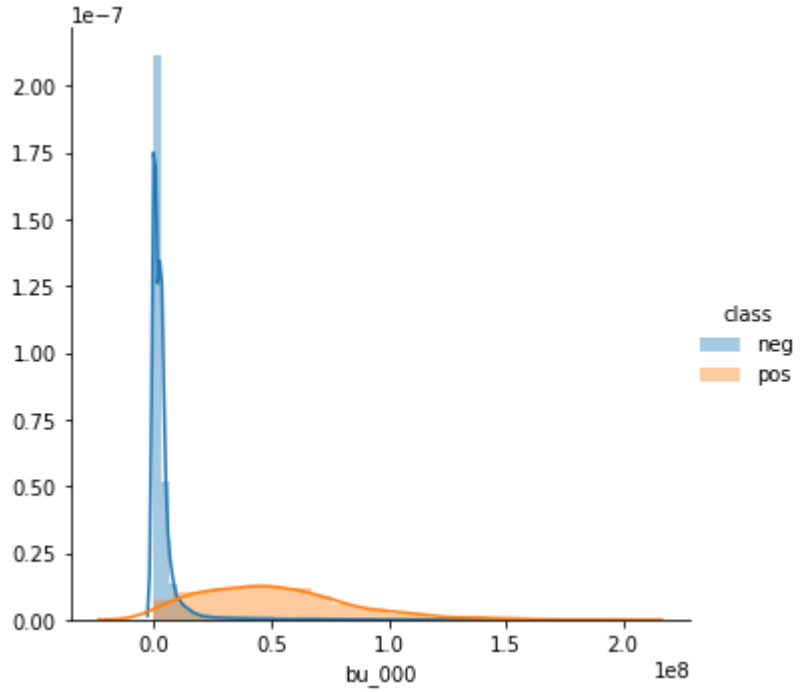
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class

In [32]:
```
EDA(data.bu_000,'bu_000')
```

```
Skew Dist    : 5.878140944889595
Kurtosis Dist: 43.82909786403321
Mean         : 4515324.704715979
Std-dev      : 10859812.286525175
```





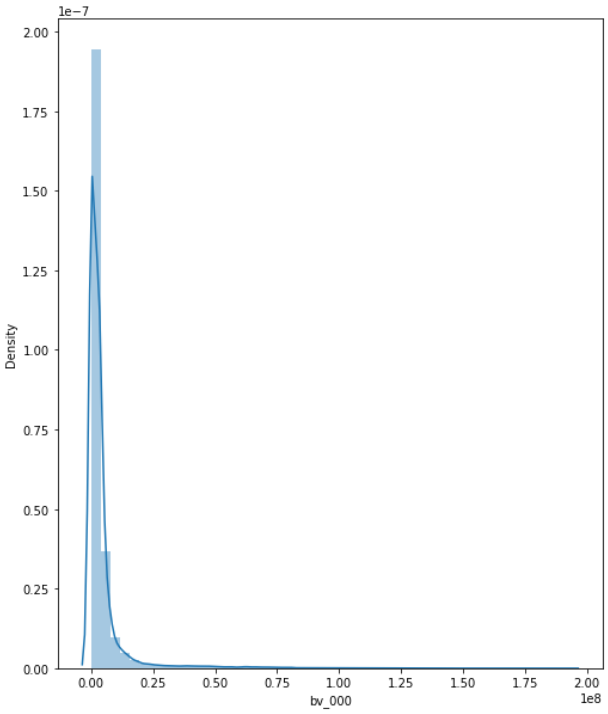


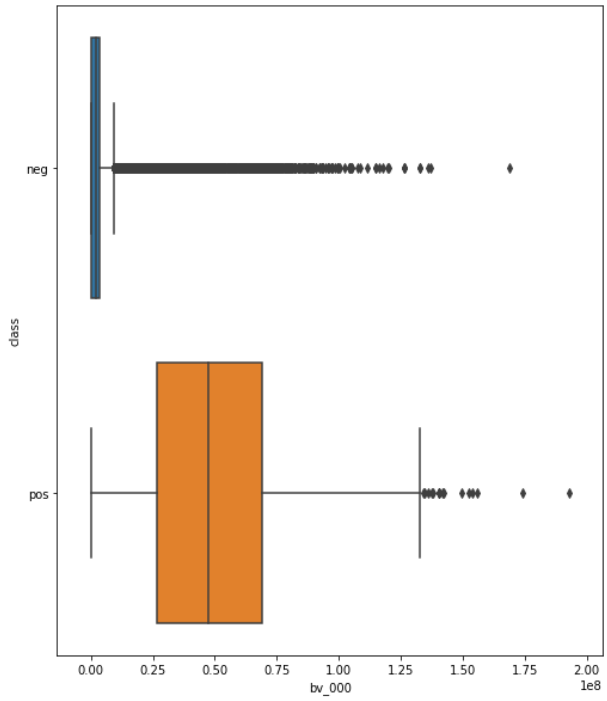
Observation

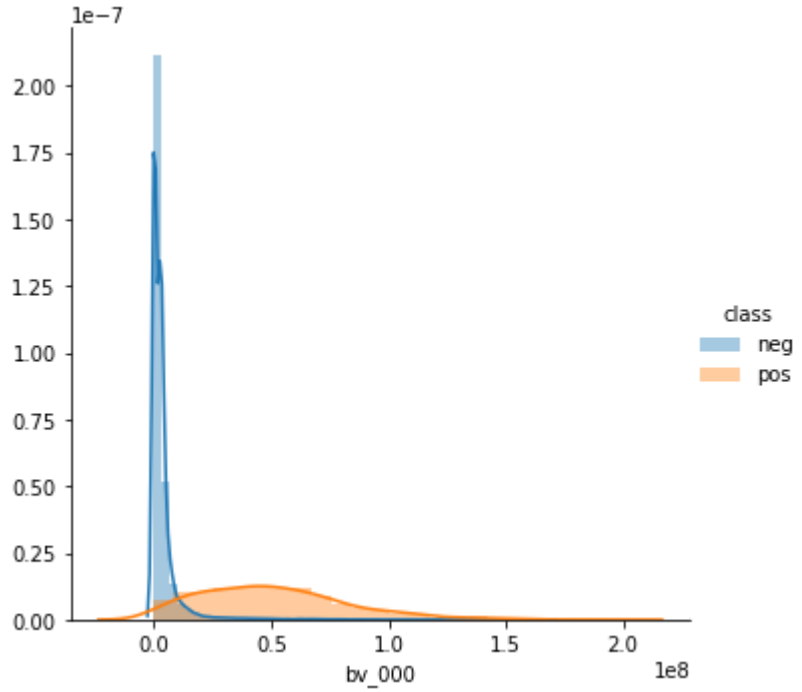
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class

In [33]:
```
EDA(data.bv_000,'bv_000')
```

```
Skew Dist    : 5.878140555033432
Kurtosis Dist: 43.829090105802464
Mean         : 4515325.287831526
Std-dev      : 10859812.948435431
```





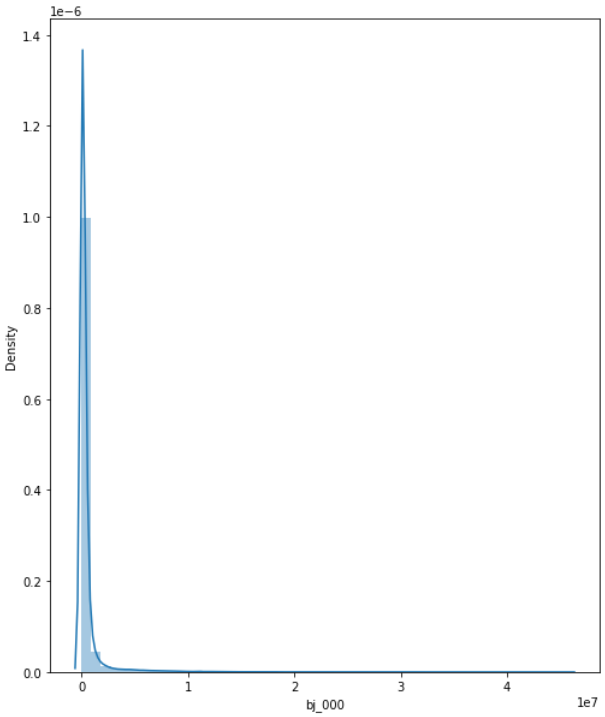


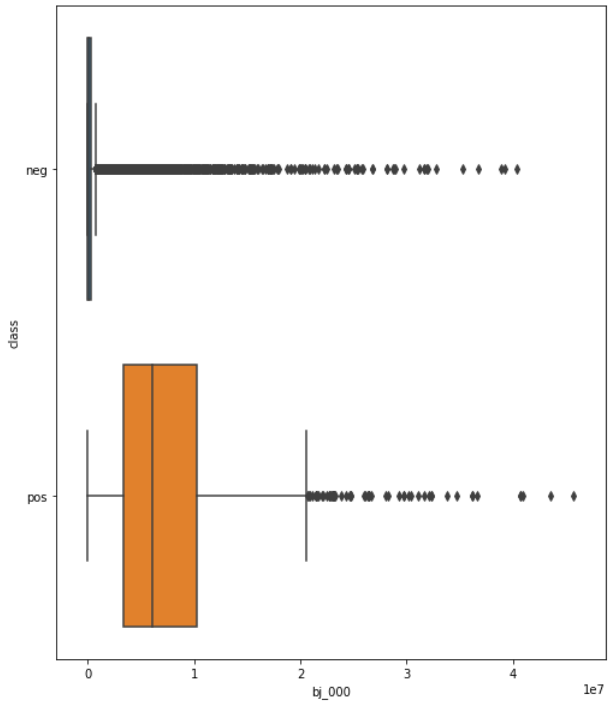
Observation

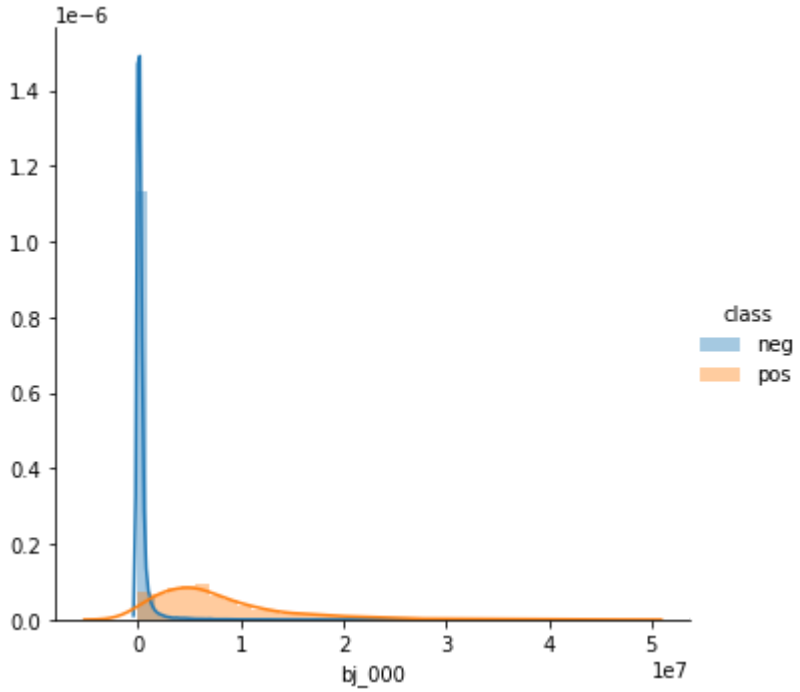
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class

In [34]:
```
EDA(data.bj_000,'bj_000')
```

```
Skew Dist    : 9.699980376452517
Kurtosis Dist: 129.7815414213665
Mean         : 510089.2305465318
Std-dev      : 1820089.5939223832
```





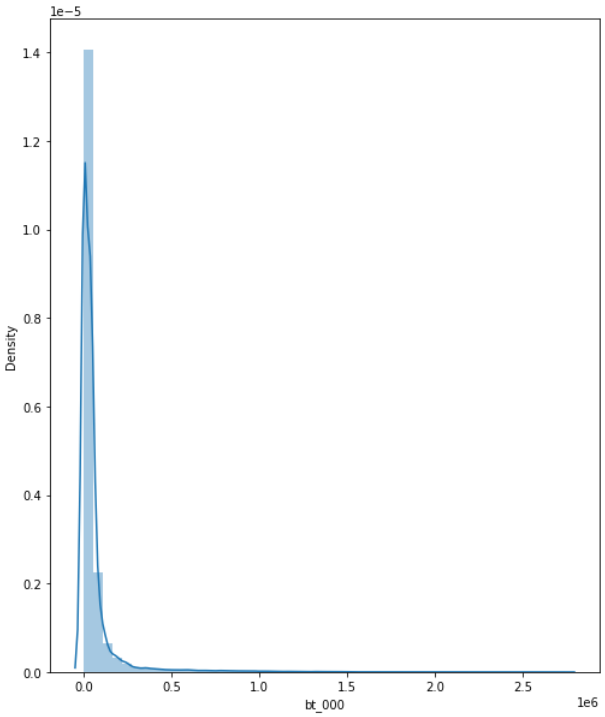


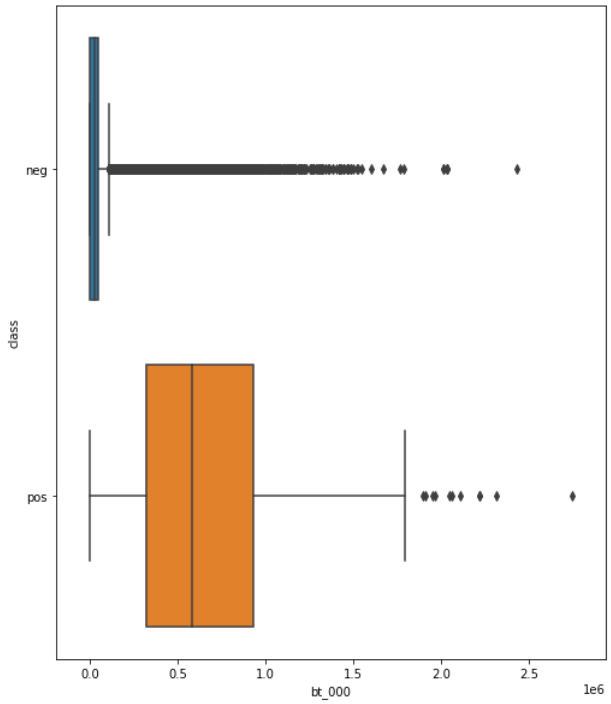
Observation

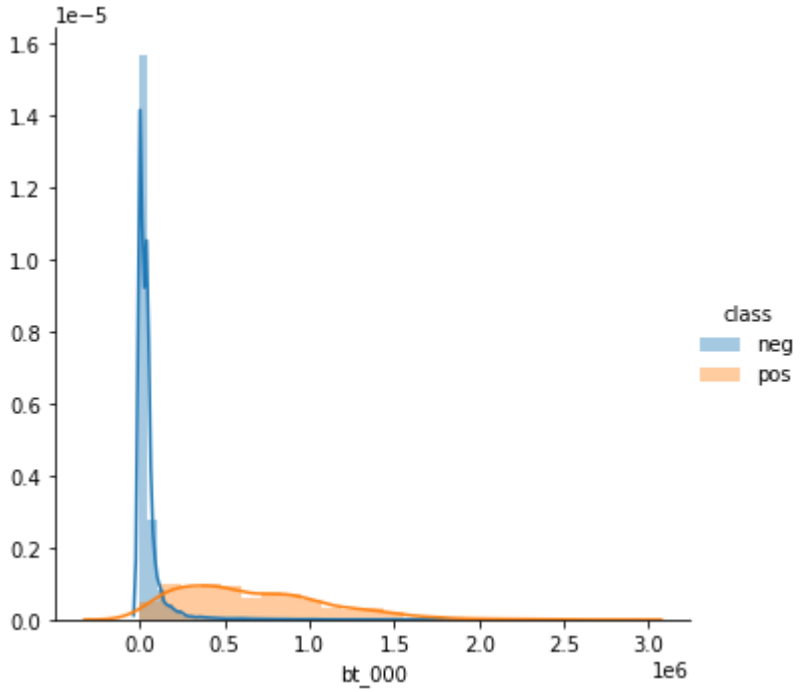
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class

In [35]:
```
EDA(data.bt_000,'bt_000')
```

```
Skew Dist    : 6.118994356901087
Kurtosis Dist: 48.349006446981065
Mean         : 59416.504414787116
Std-dev      : 145445.22192095238
```





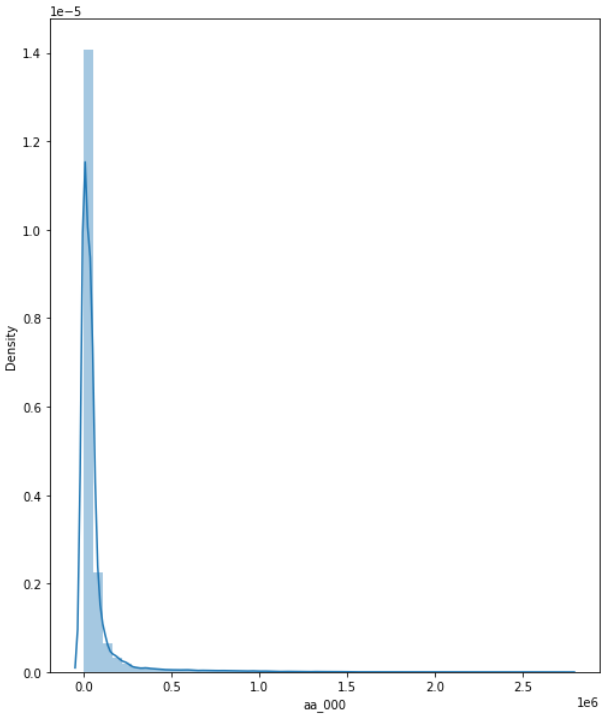


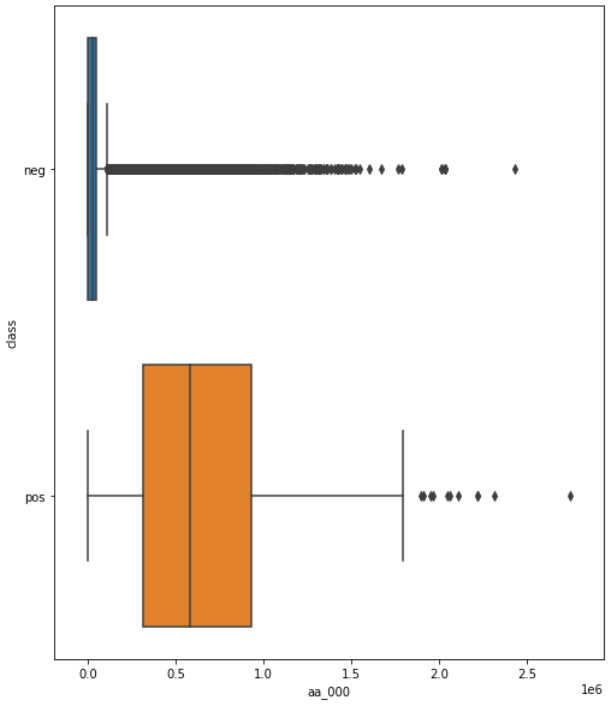
Observation

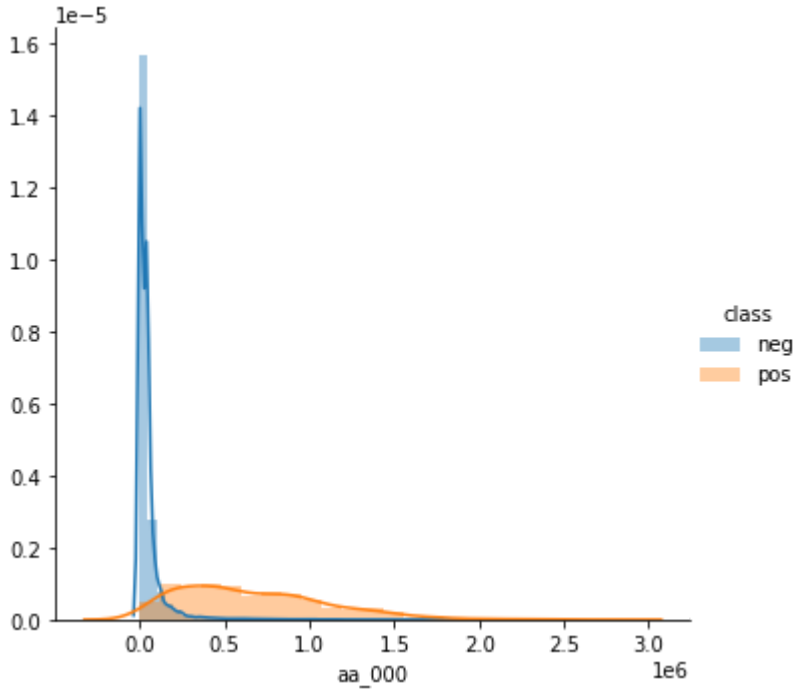
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class

In [36]:
```
EDA(data.aa_000,'aa_000')
```

```
Skew Dist    : 6.115752483633429
Kurtosis Dist: 48.29744108861675
Mean         : 59336.499566666665
Std-dev      : 145428.84460941362
```





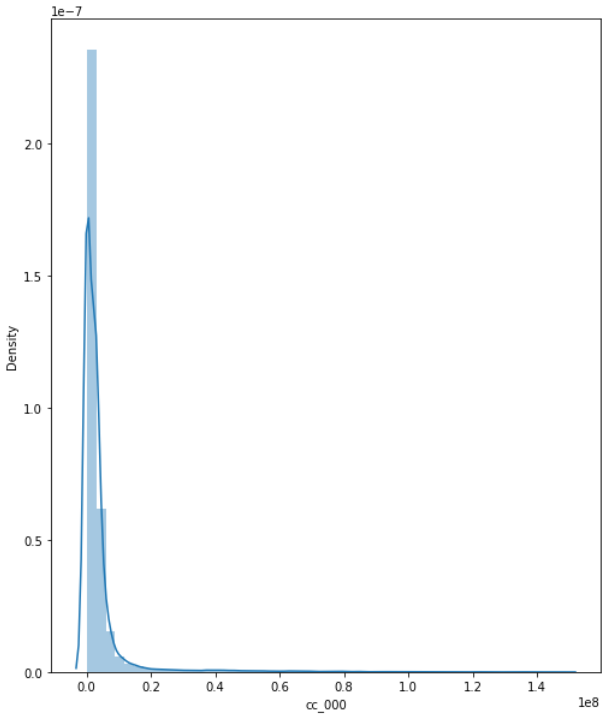


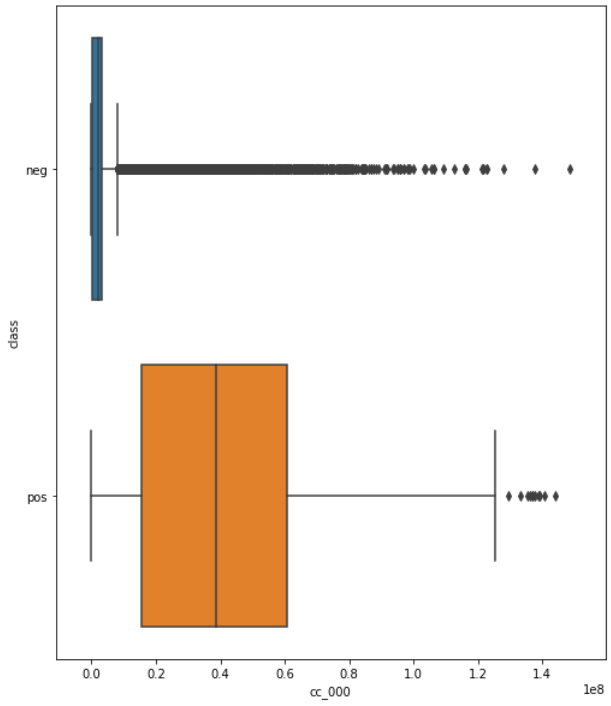
Observation

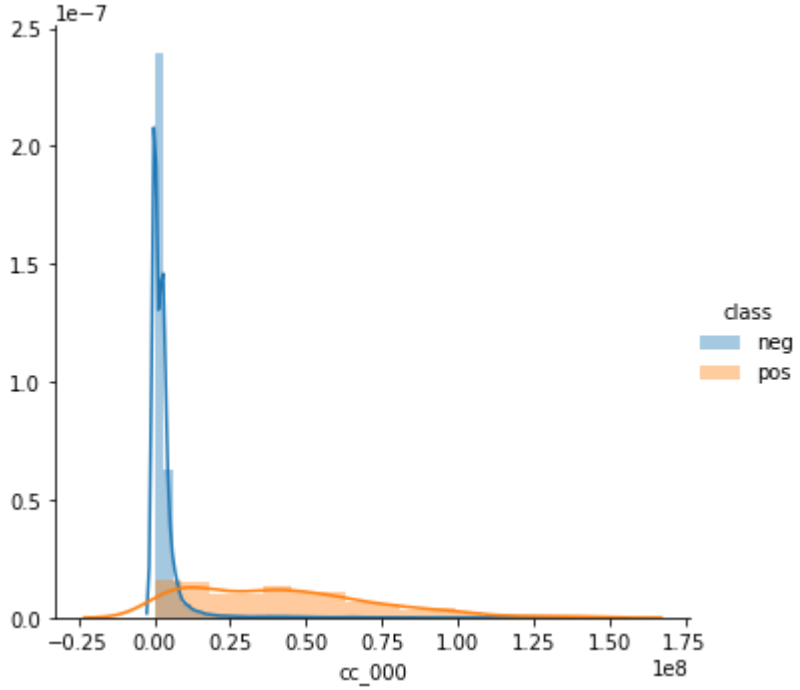
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class

In [37]:
```
EDA(data.cc_000,'cc_000')
```

```
Skew Dist    : 6.355293277597199
Kurtosis Dist: 50.3463115653065
Mean         : 3803443.5634857696
Std-dev      : 9625587.429590397
```

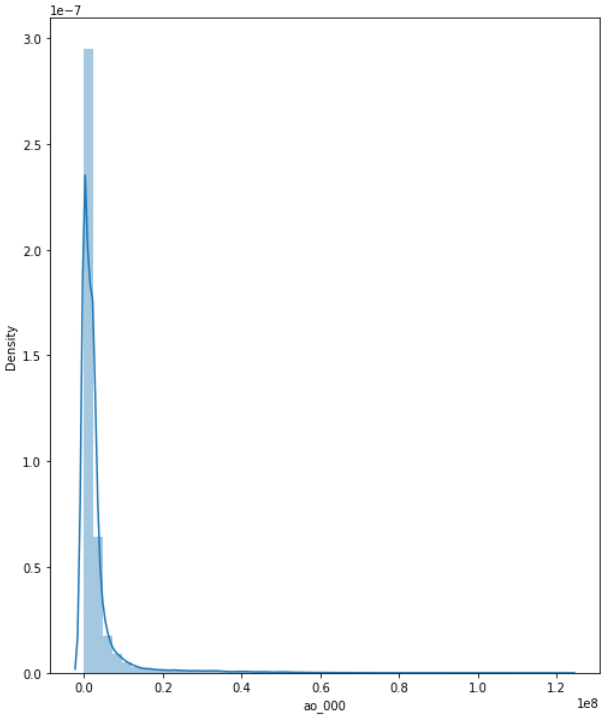


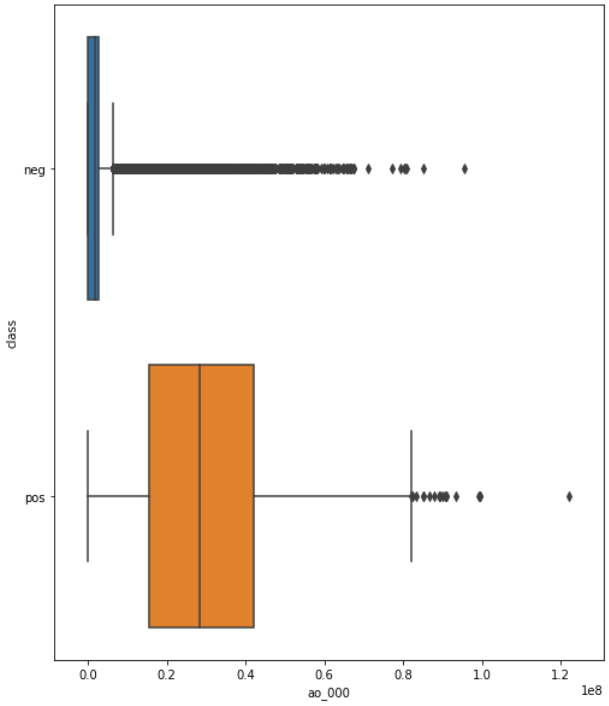


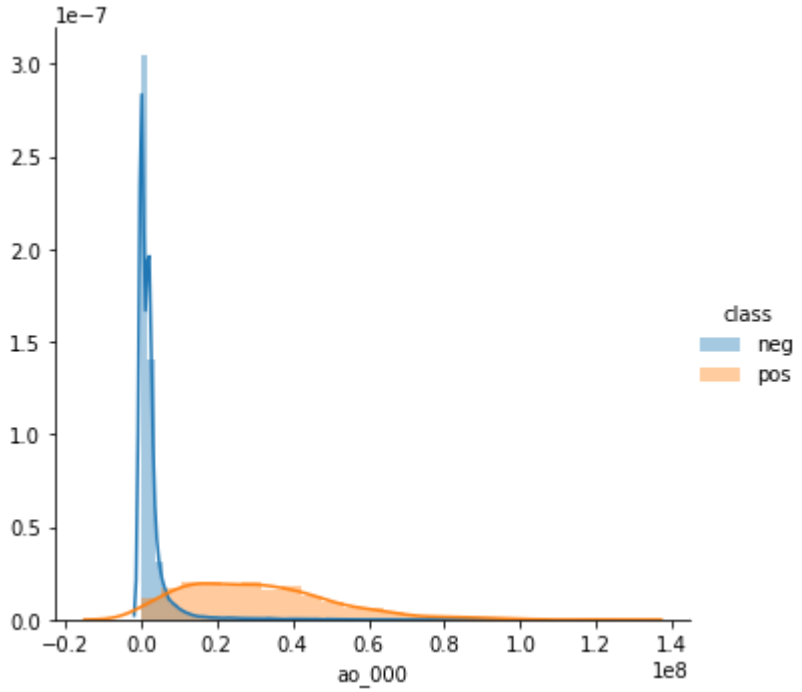


In [38]:
```
EDA(data.ao_000,'ao_000')
```

```
Skew Dist    : 5.668929156683823
Kurtosis Dist: 41.33756568417306
Mean         : 3002440.306946525
Std-dev      : 6819460.997853249
```





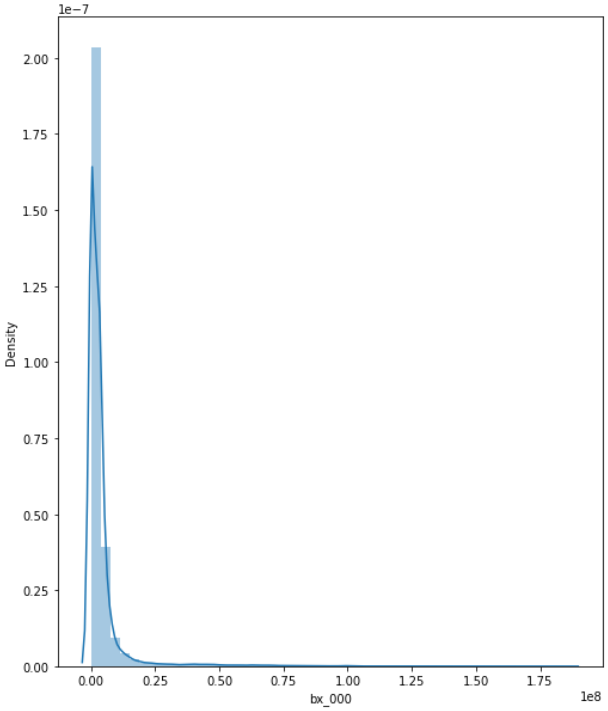


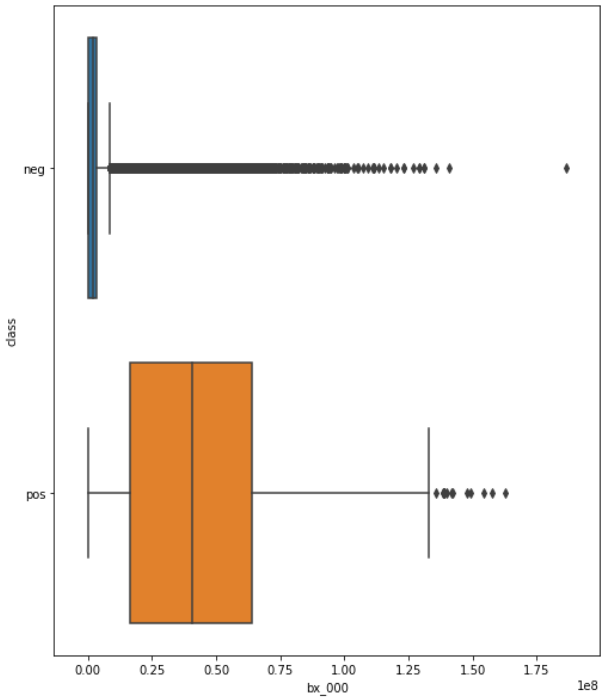
Observation

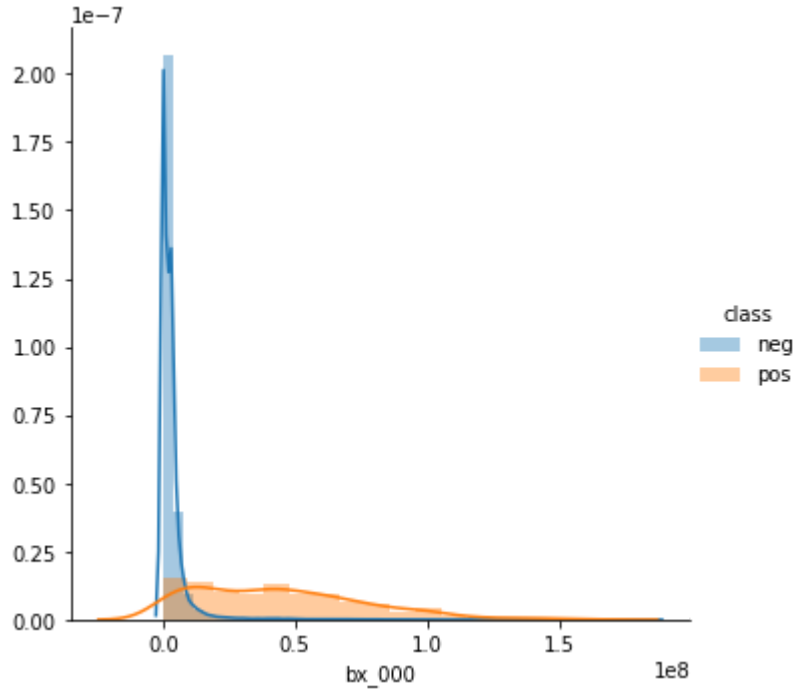
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- 90 % neg data point are below than the 0.2

In [39]:
```
EDA(data.bx_000,'bx_000')
```

```
Skew Dist    : 6.324345032219356
Kurtosis Dist: 50.01530449686252
Mean         : 4112218.096893009
Std-dev      : 10351447.281259991
```





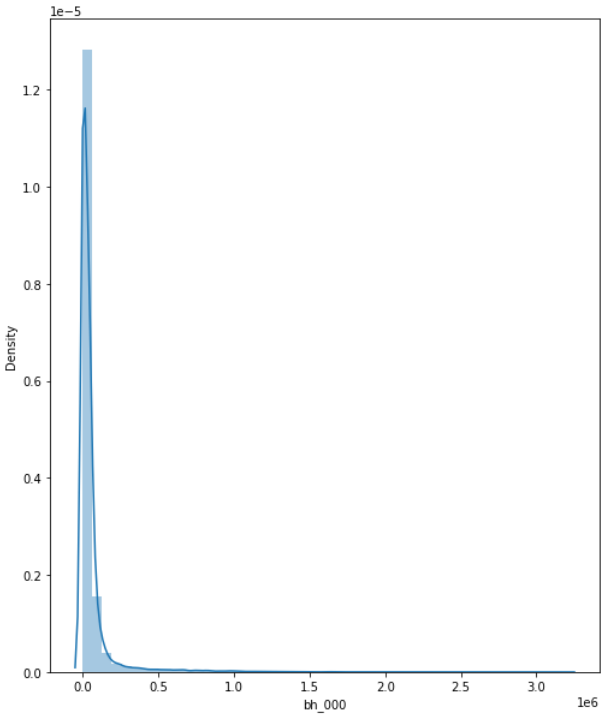


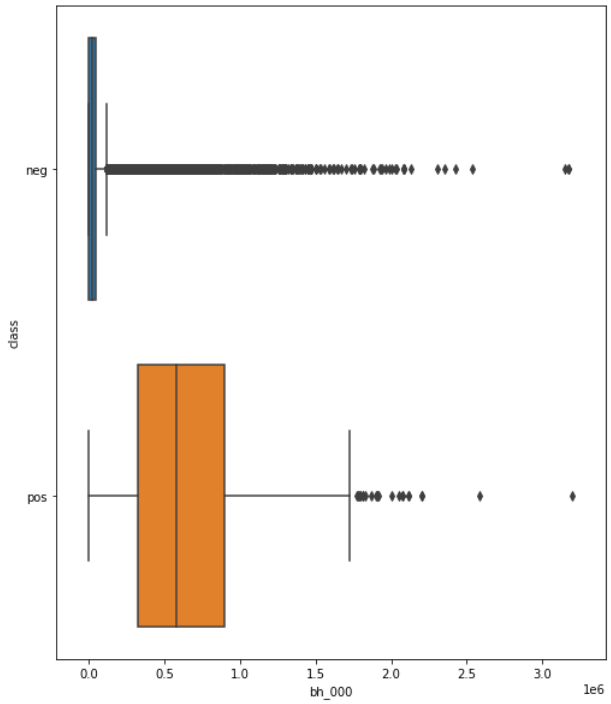
Observation

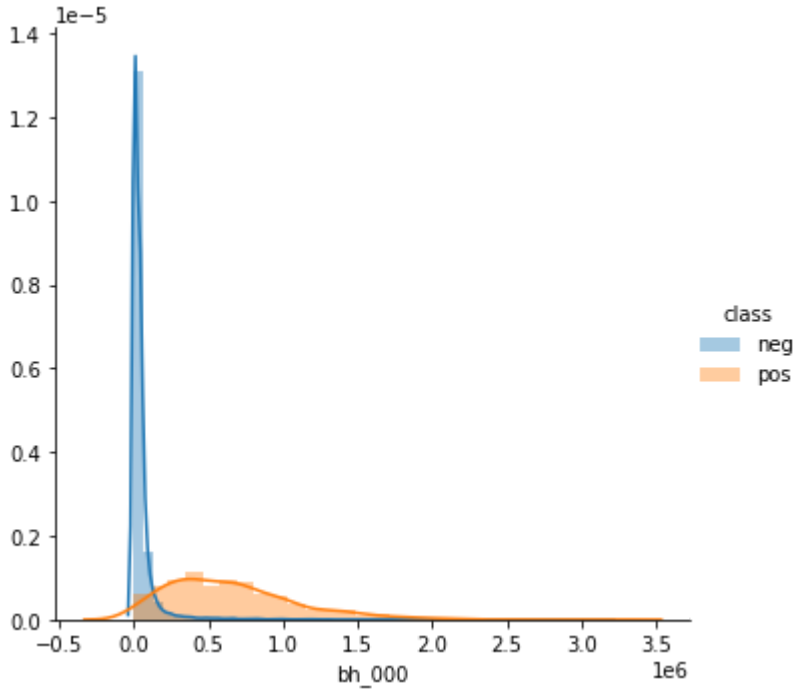
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class

In [40]:
```
EDA(data.bh_000,'bh_000')
```

```
Skew Dist    : 6.801442568917349
Kurtosis Dist: 63.23003482173449
Mean         : 57943.08221301257
Std-dev      : 152208.03896099696
```





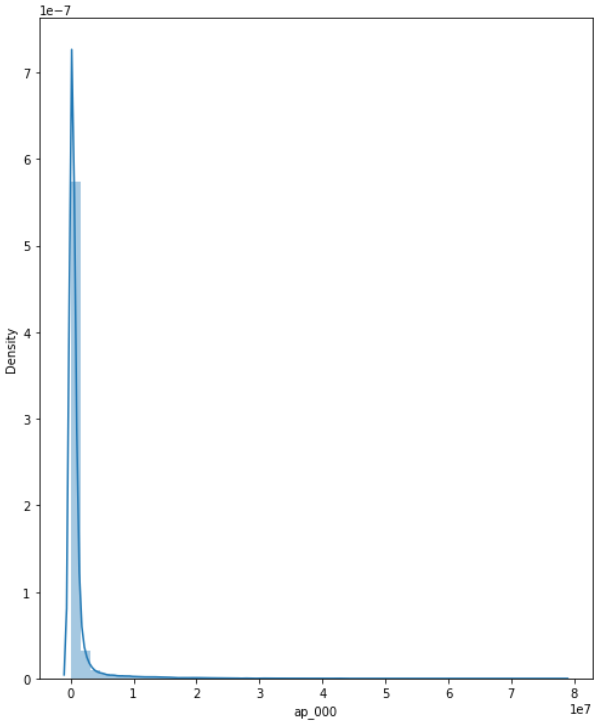


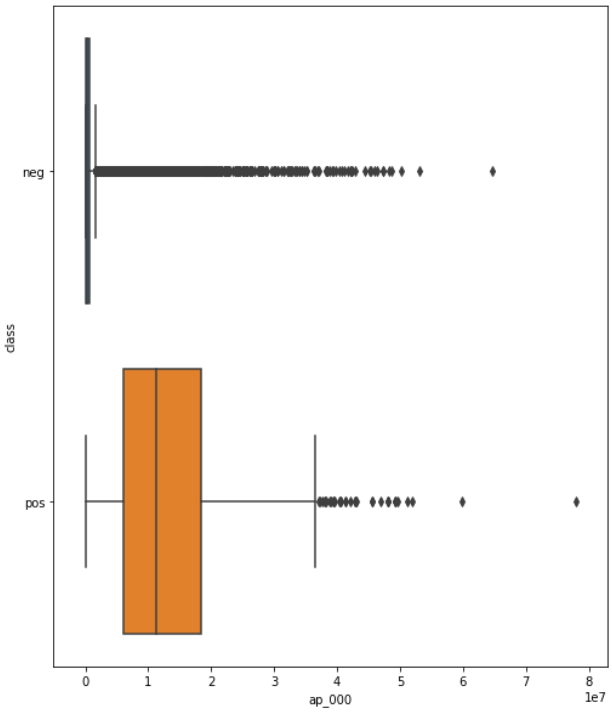
Observation

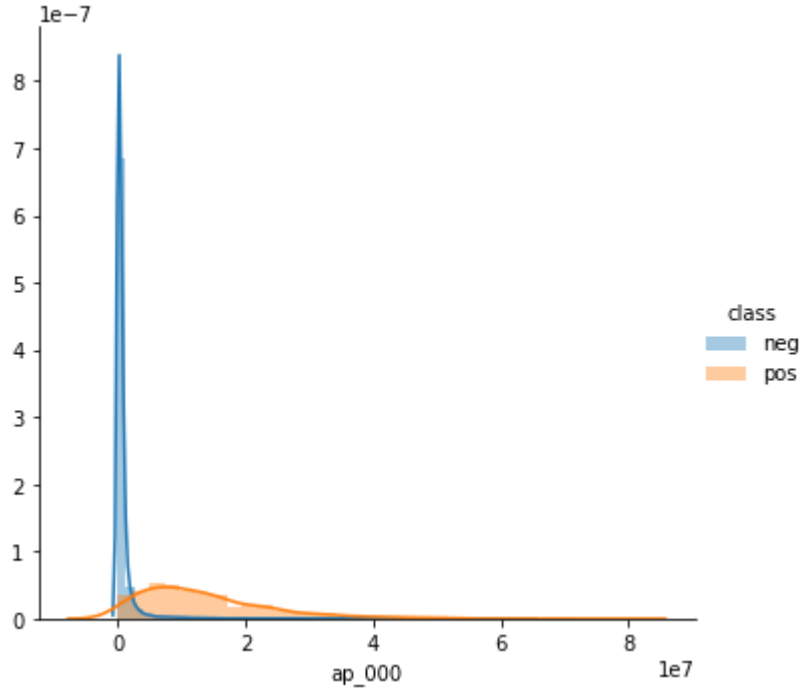
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- range of neg class is mostly less than 0.3

In [41]:
```
EDA(data.ap_000,'ap_000')
```

```
Skew Dist    : 7.807262184400477
Kurtosis Dist: 81.67357396443009
Mean         : 1004159.5569594663
Std-dev      : 3088431.018957874
```







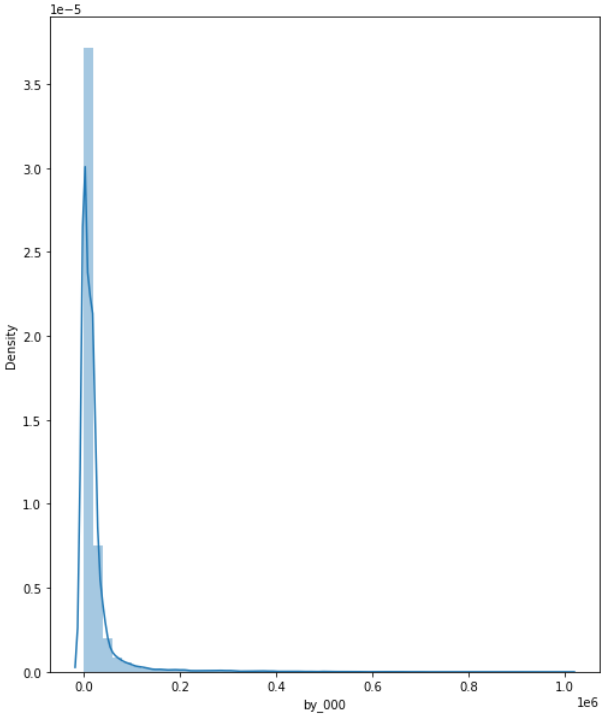
Observation

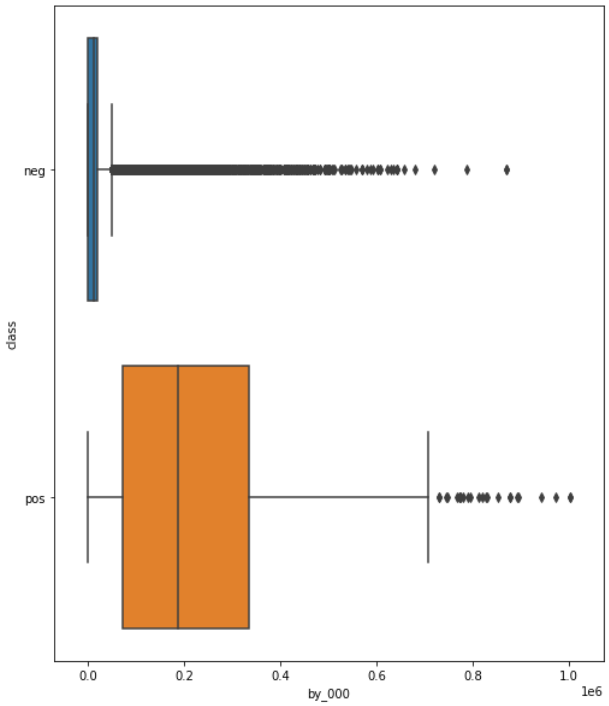
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- range of neg class is mostly less than 0.3

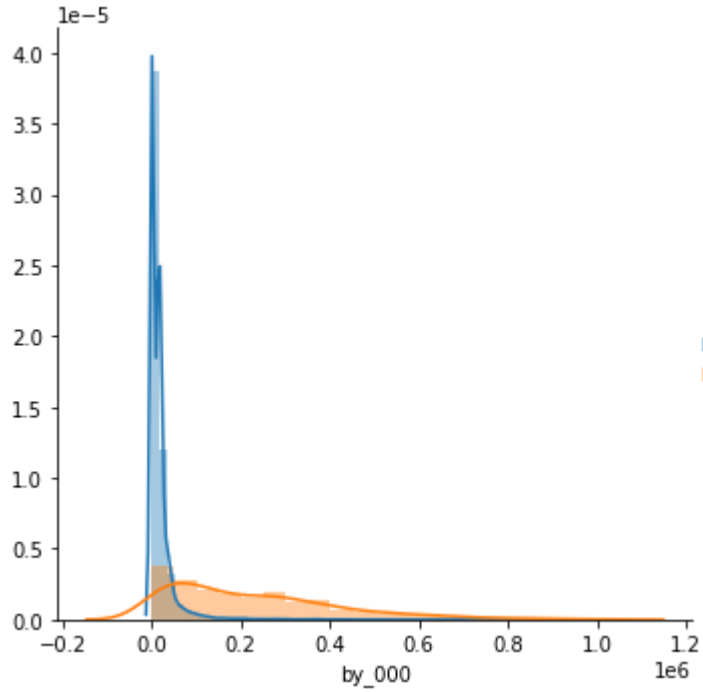
In [42]:
```
EDA(data.by_000,'by_000')
```

```
Skew Dist    : 6.974825247335402
Kurtosis Dist: 65.41624438983878
```

```
Mean          : 22028.93335797201
Std-dev       : 53992.37122064625
```







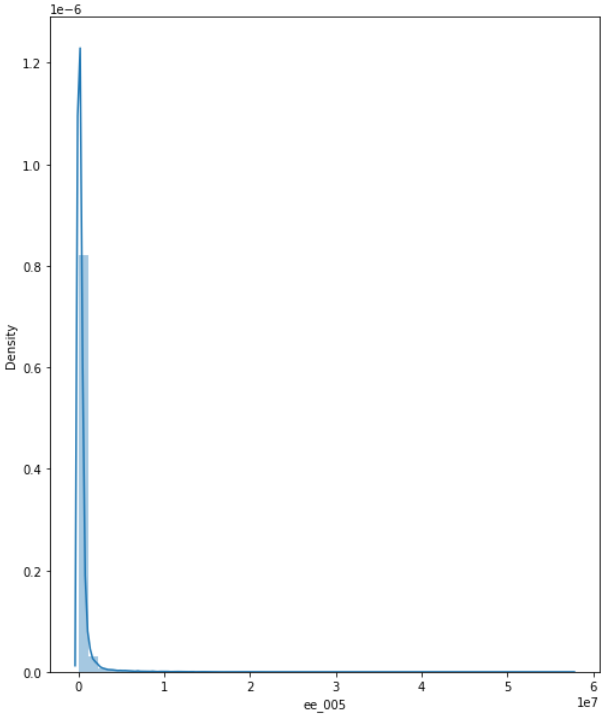
Observation

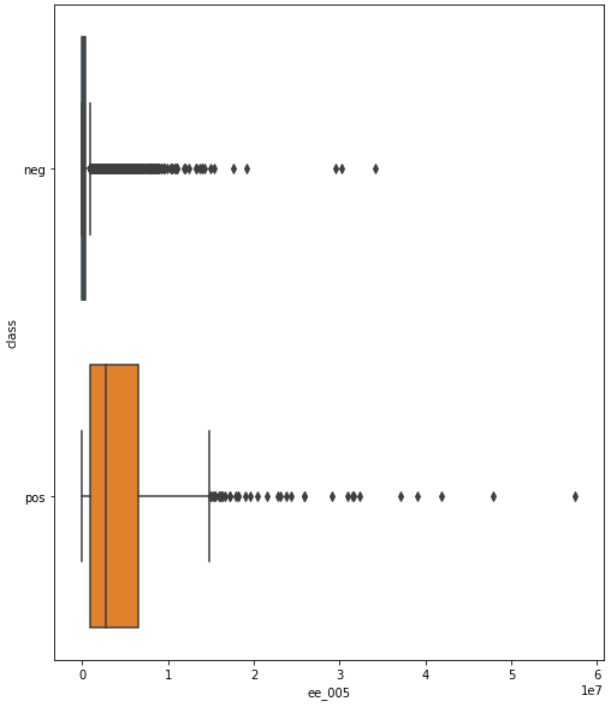
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- range of neg class is mostly less than 0.1

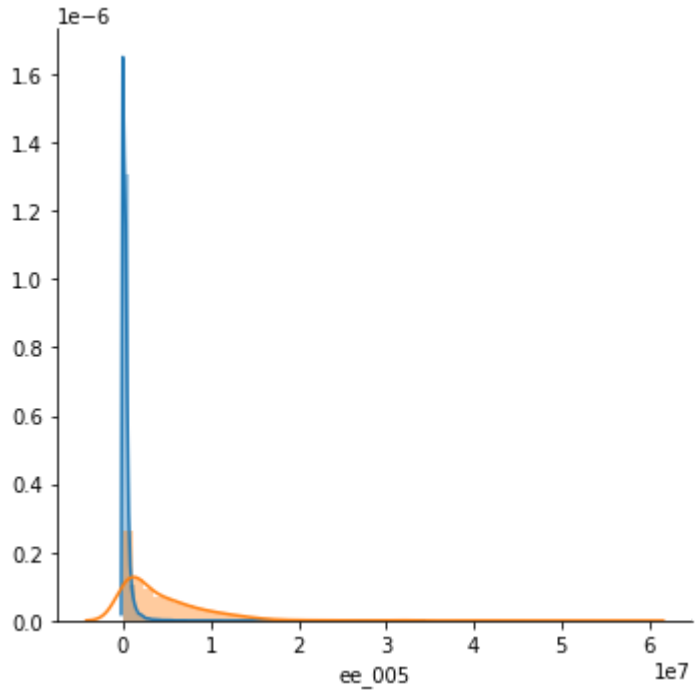
In [43]:
```
EDA(data.ee_005,'ee_005')
```

```
Skew Dist     : 14.555777045121376
```

```
Kurtosis Dist: 390.77780425267366
Mean         : 393946.1979807514
Std-dev      : 1121034.9589538286
```





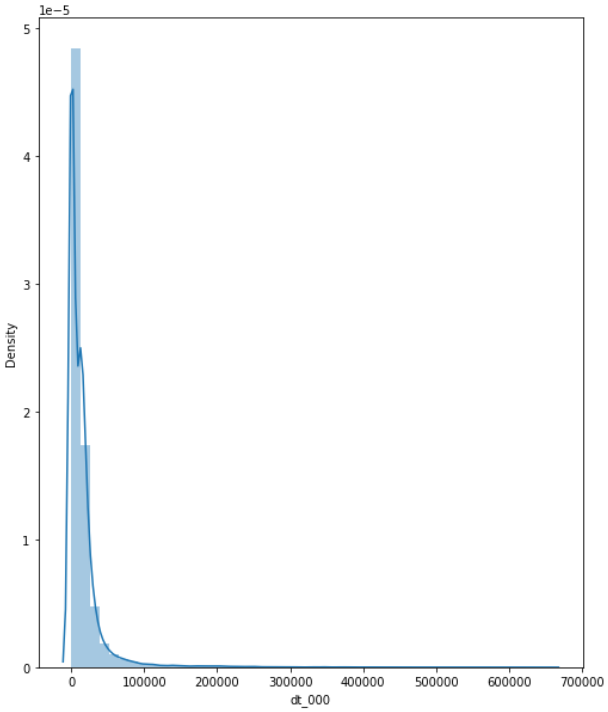


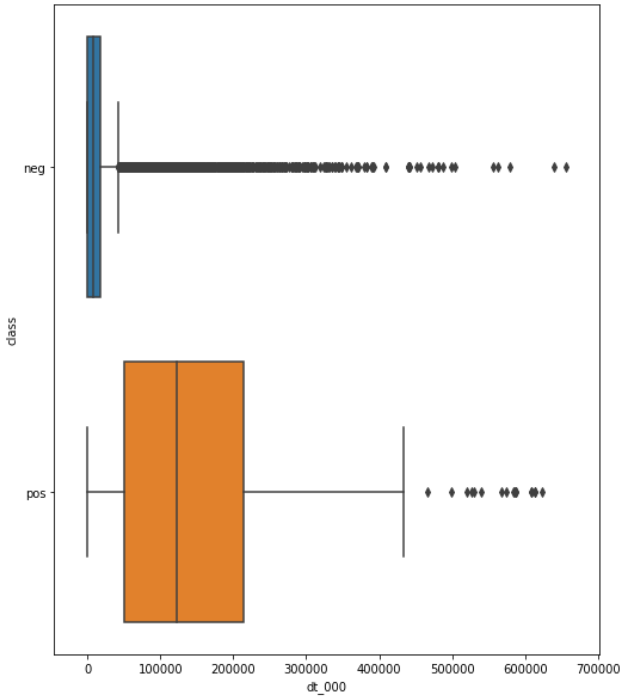
Observation

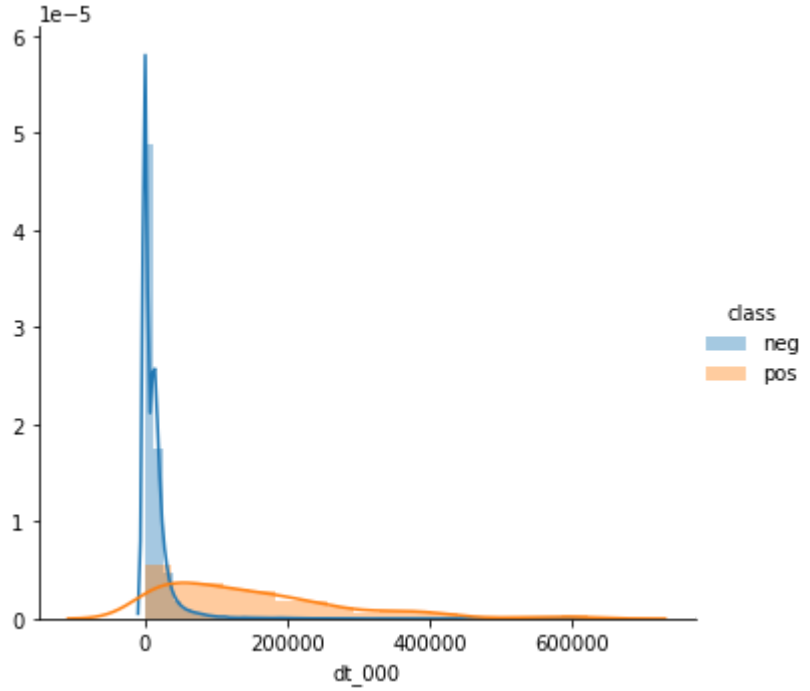
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- range of neg class is mostly less than 0.2

In [44]:
```
EDA(data.dt_000,'dt_000')
```

```
Skew Dist    : 7.053782942840553
Kurtosis Dist: 72.41925822721566
Mean         : 15403.354669739669
Std-dev      : 33800.72788692232
```





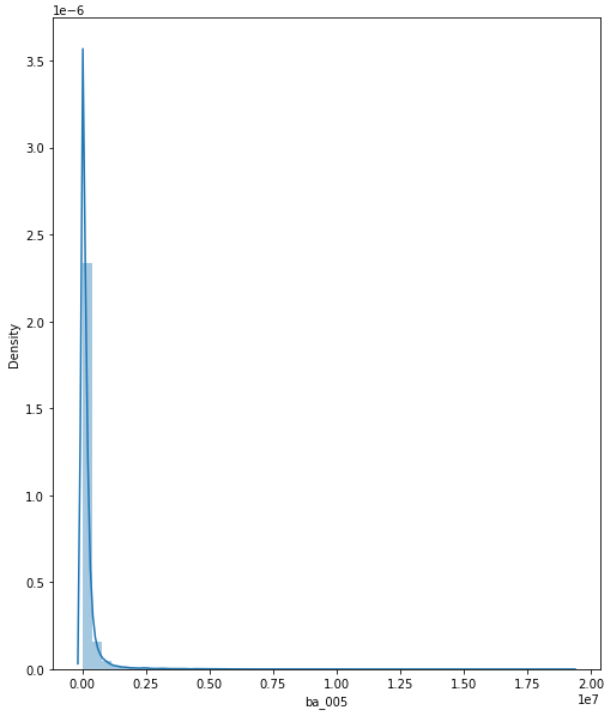


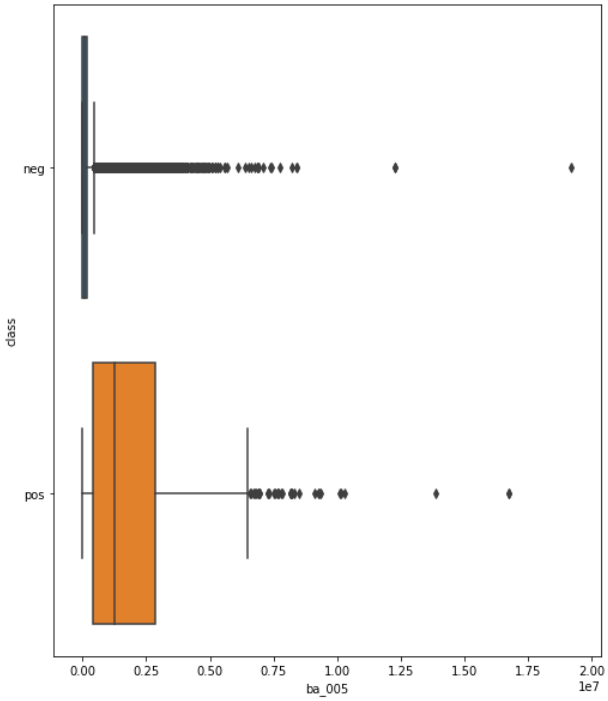
Observation

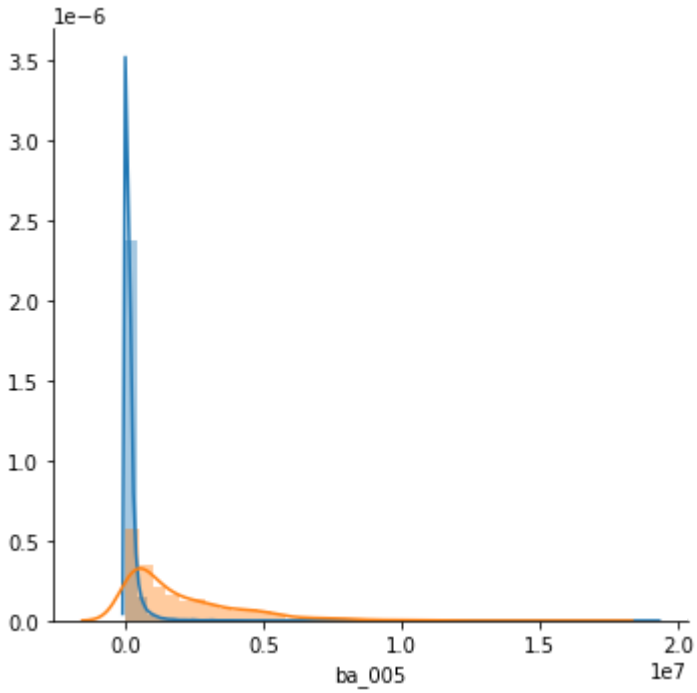
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- range of neg class is mostly less than 5000

In [45]:
```
EDA(data.ba_005,'ba_005')
```

```
Skew Dist    : 9.811978667551383
Kurtosis Dist: 164.46651730548814
Mean         : 188941.18940517938
Std-dev      : 509262.88634434657
```

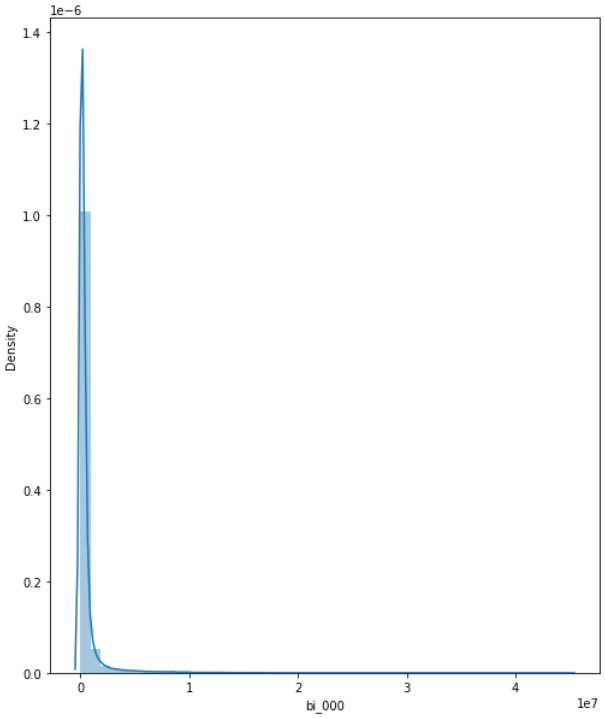


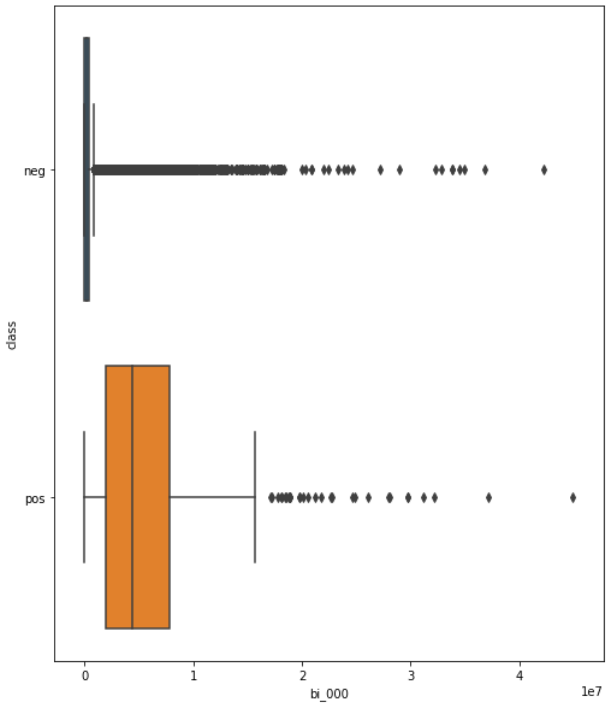


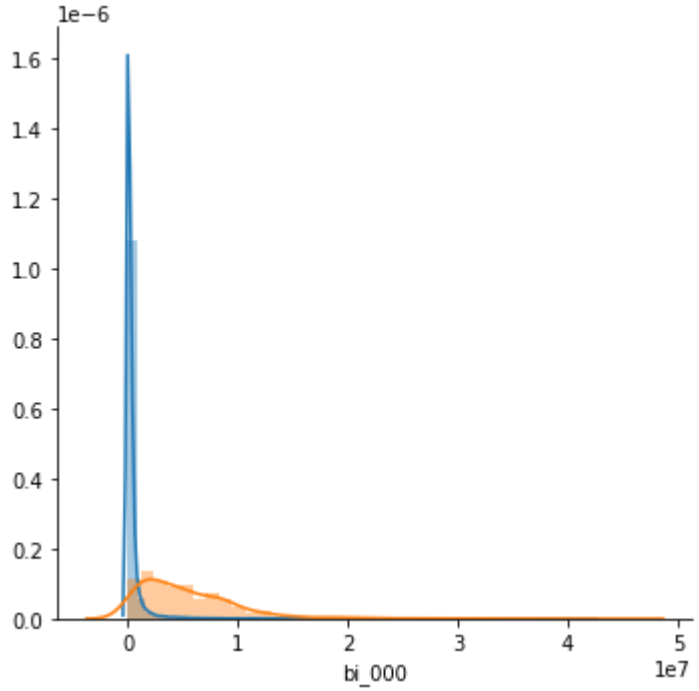


In [46]:
```
EDA(data.bi_000,'bi_000')
```

```
Skew Dist    : 9.198927541361796
Kurtosis Dist: 131.76960915235196
Mean         : 492207.57287371025
Std-dev      : 1485172.0072140254
```







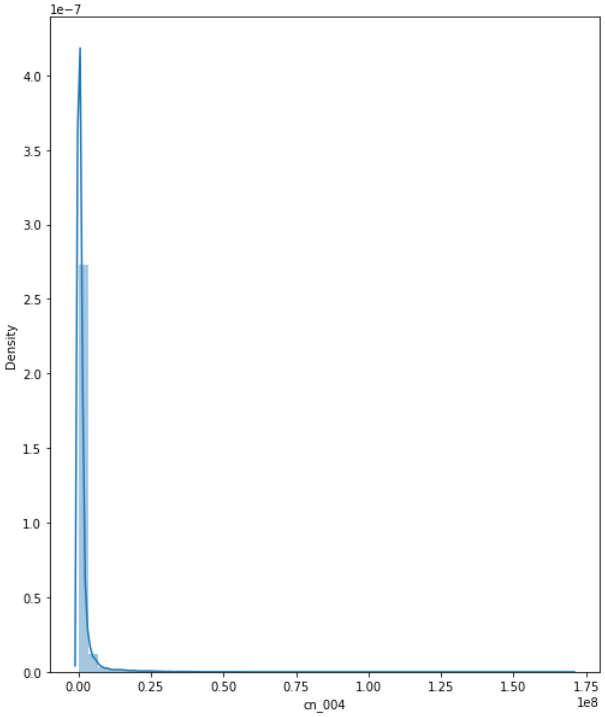
Observation

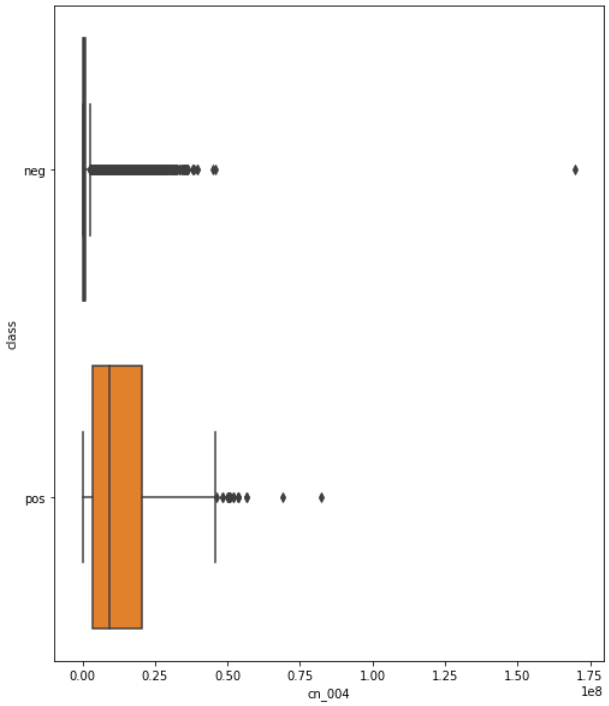
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- range of neg class is mostly less than 0.2

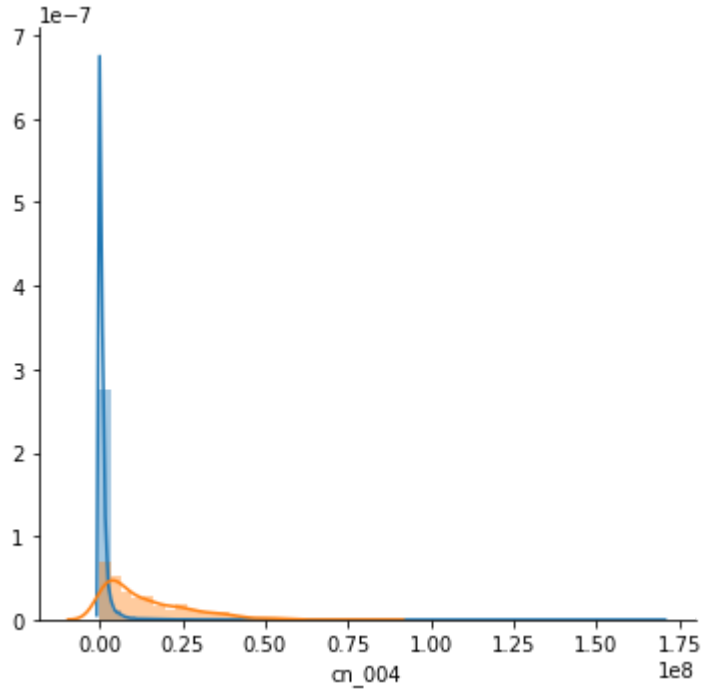
In [47]:
```
EDA(data.cn_004,'cn_004')
```

```
Skew Dist    : 8.9201627125974
Kurtosis Dist: 173.14740307837513
```

```
Mean          : 1282835.1424814123
Std-dev       : 3357226.0359253665
```







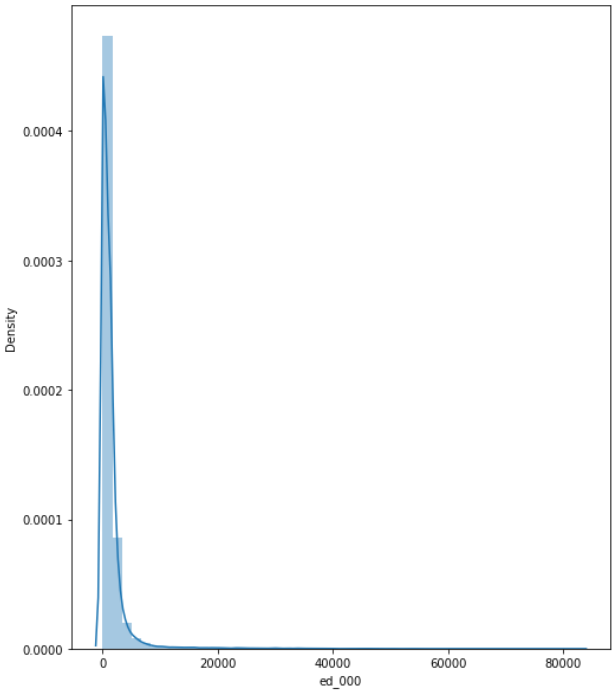
Observation

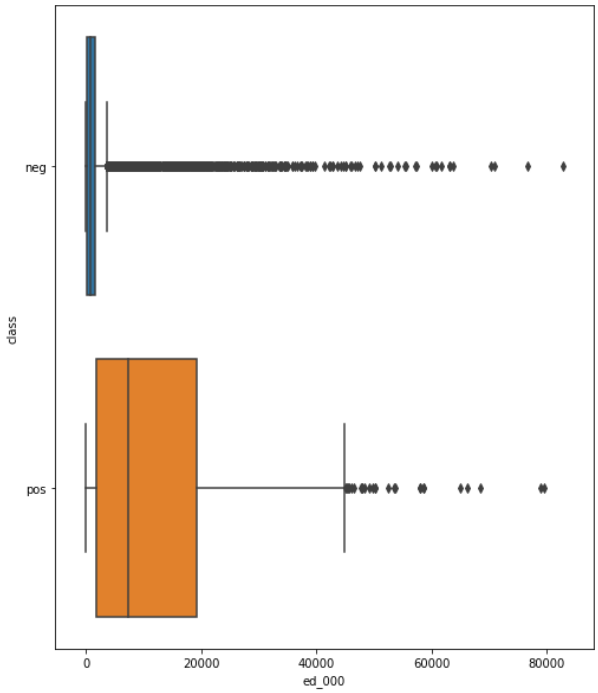
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class

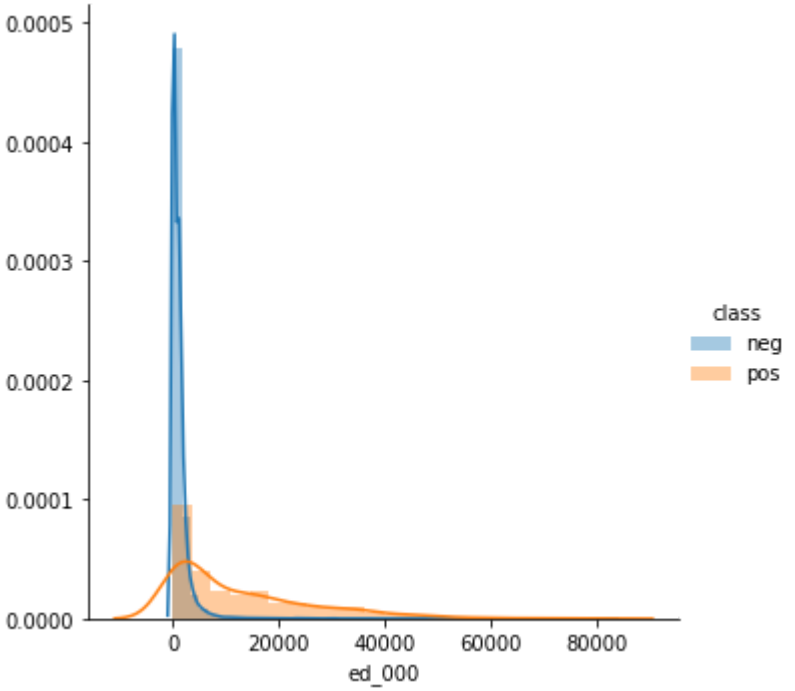
In [48]:
```
EDA(data.ed_000,'ed_000')
```

```
Skew Dist     : 8.815512491816088
Kurtosis Dist: 107.99278920548183
```

```
Mean          : 1452.155212401134
Std-dev       : 3524.993713764263
```





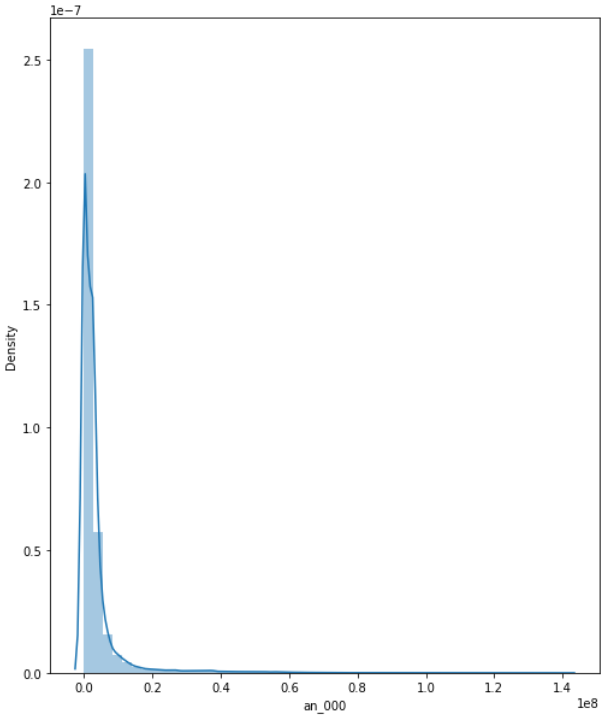


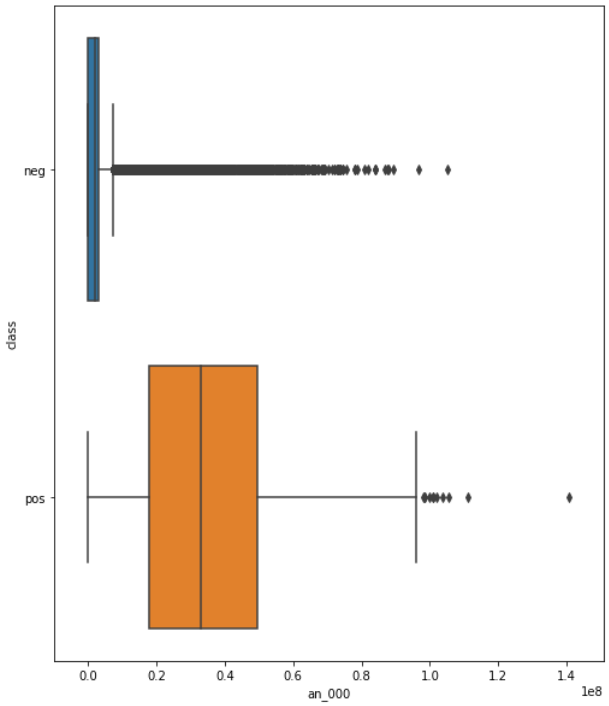
Observation

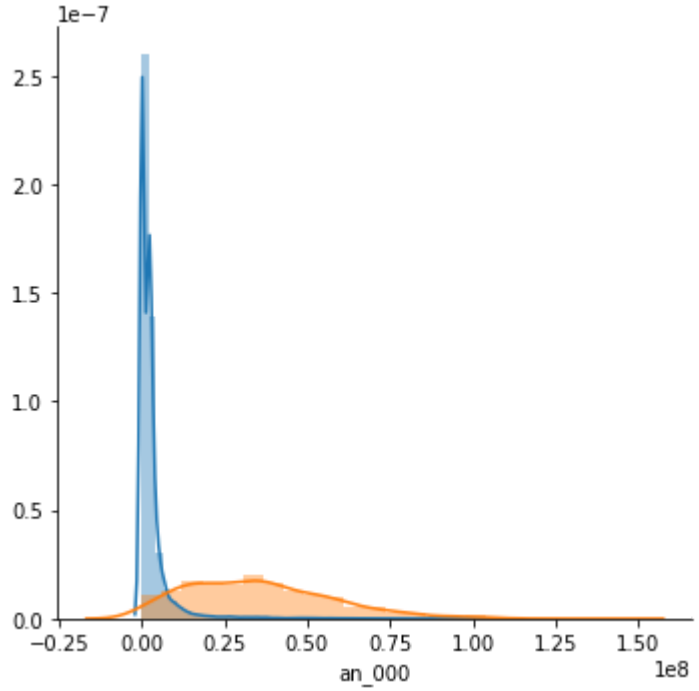
- the range of neg class is much lower than the postive class
- range of neg class is mostly less than 5000

In [49]:
```
EDA(data.an_000,'an_000')
```

```
Skew Dist     : 5.568640359723724
Kurtosis Dist: 39.40827246768601
Mean          : 3461037.0434987703
Std-dev       : 7790284.106562337
```







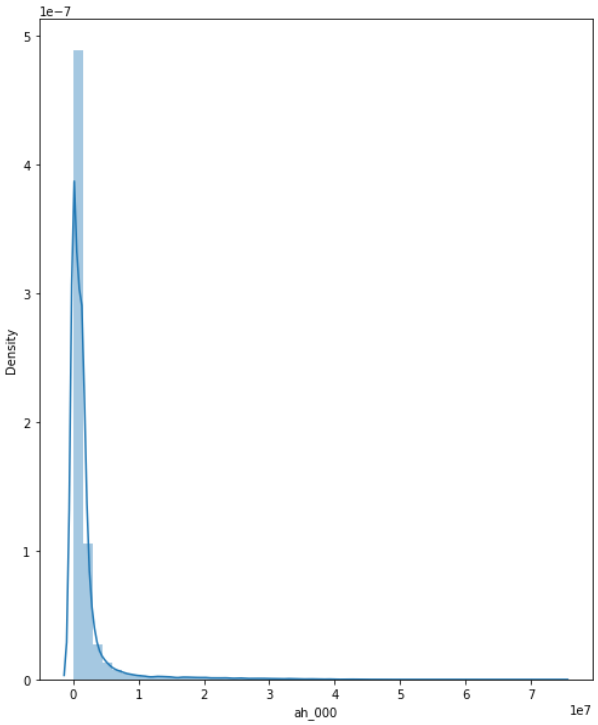
Observation

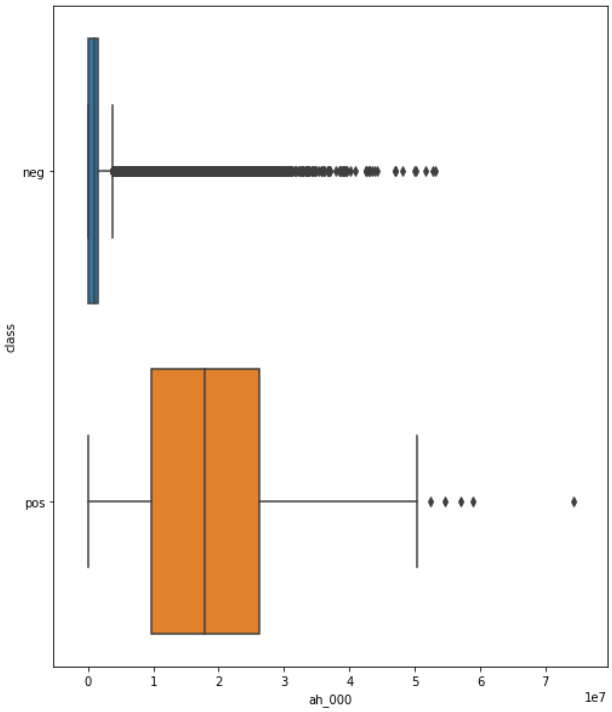
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- range of neg class is mostly less than 0.25

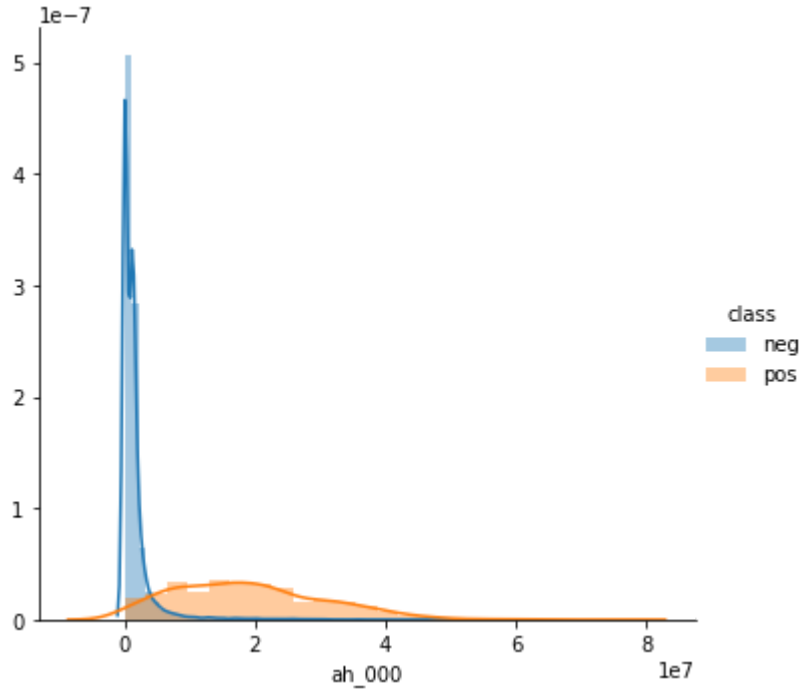
In [50]:
```
EDA(data.ah_000,'ah_000')
```

```
Skew Dist    : 5.534549192024902
Kurtosis Dist: 38.06354053629009
```

```
Mean         : 1809931.1761098476
Std-dev      : 4185704.873544476
```







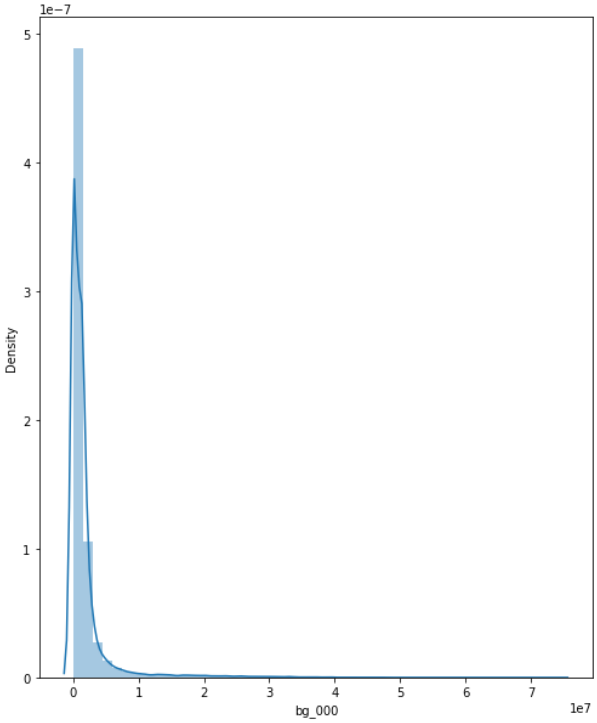
Observation

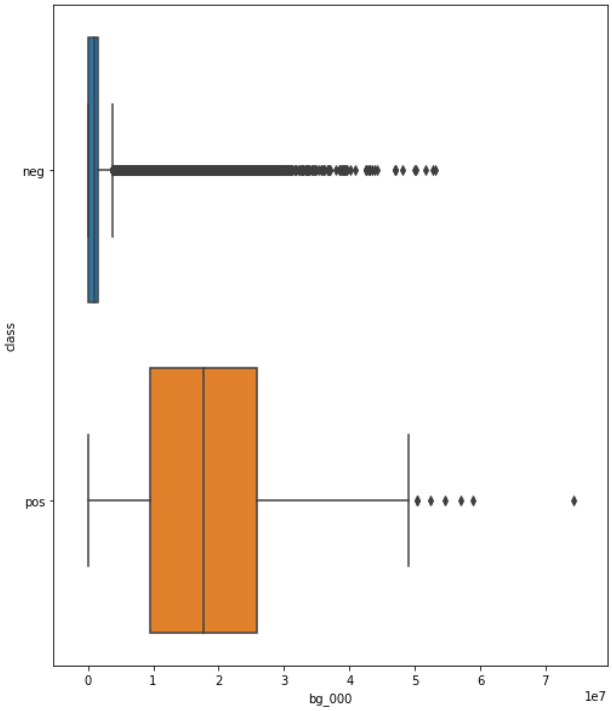
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- range of neg class is mostly less than 0.5

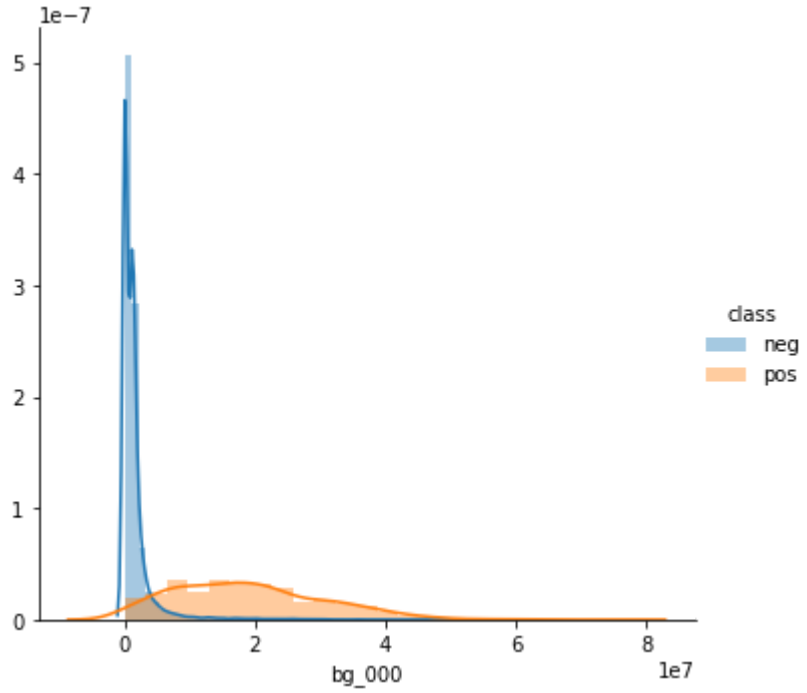
In [51]:
```
EDA(data.bg_000,'bg_000')
```

```
Skew Dist    : 5.534854018691377
```

```
Kurtosis Dist: 38.1069465011676
Mean         : 1809430.9845345193
Std-dev      : 4180164.565043624
```





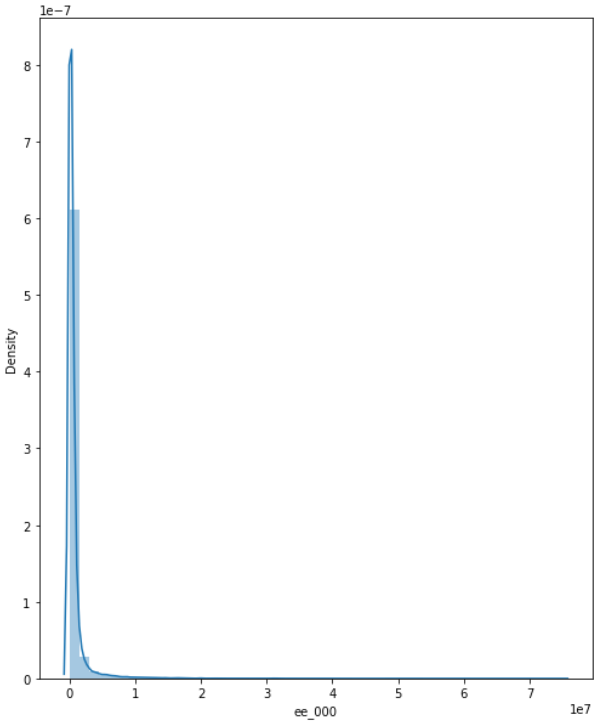


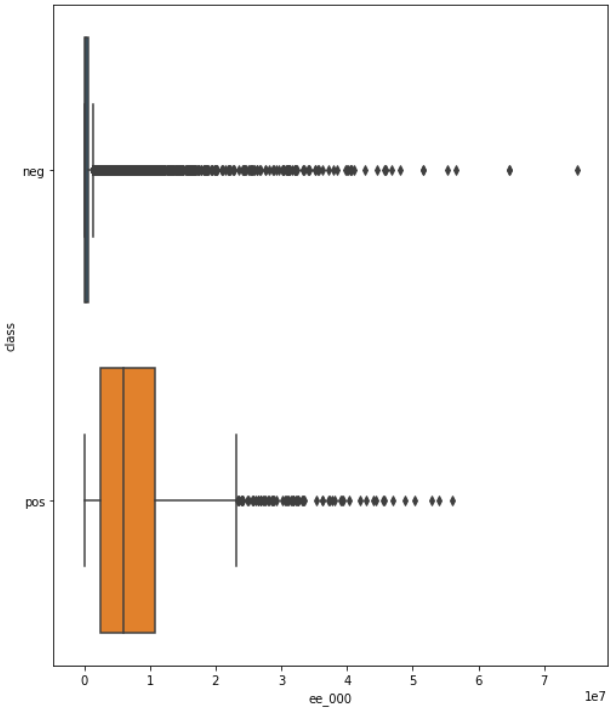
Observation

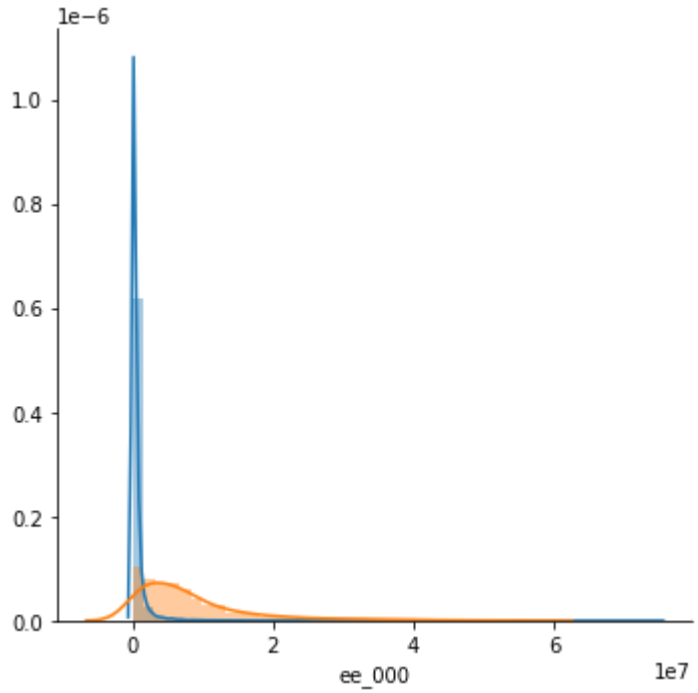
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- range of neg class is mostly less than 0.5

In [52]:
```
EDA(data.ee_000,'ee_000')
```

```
Skew Dist    : 10.599929827118538
Kurtosis Dist: 160.44235228123387
Mean         : 733404.2128132953
Std-dev      : 2416145.293050484
```





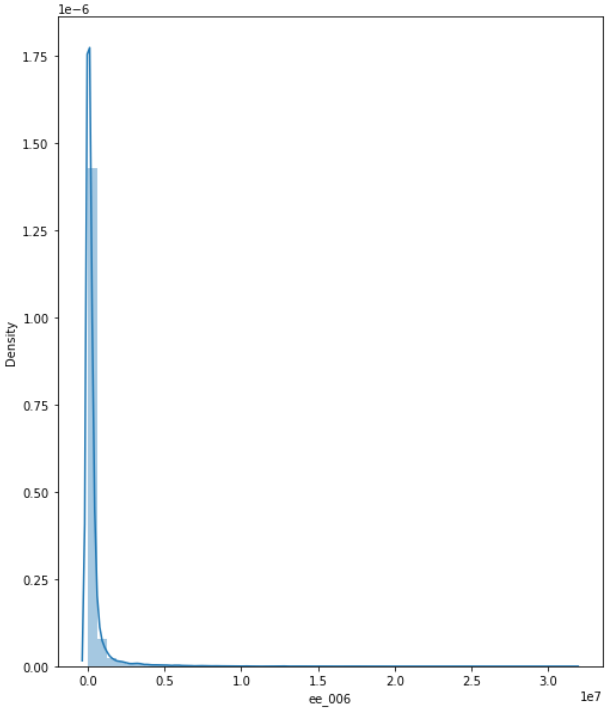


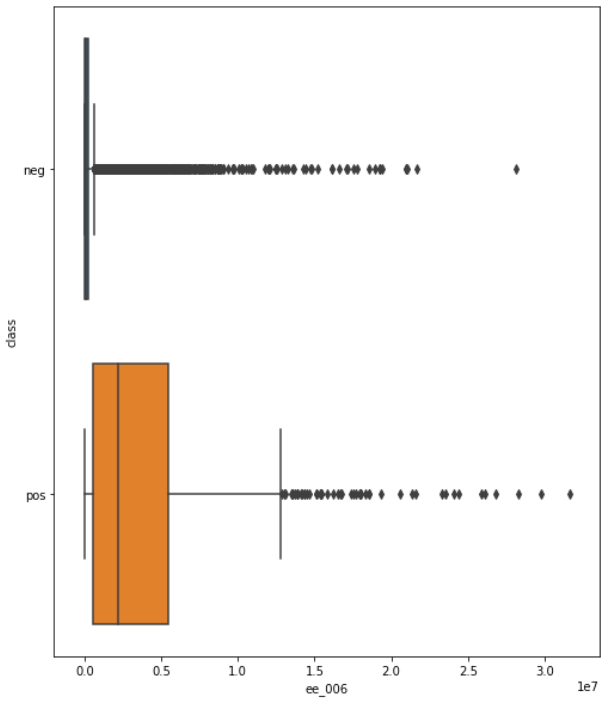
Observation

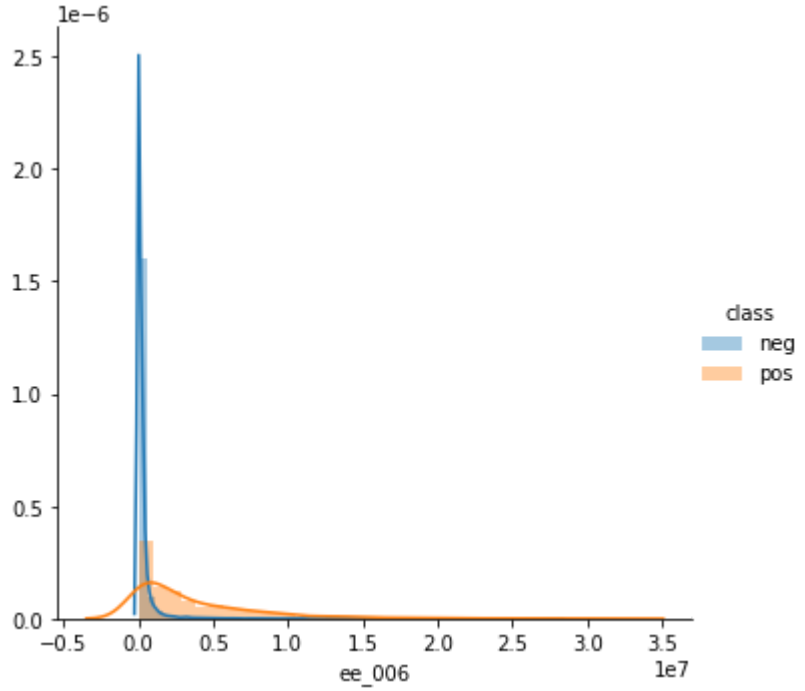
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- range of neg class is mostly less than 0.3

In [53]:
```
EDA(data.ee_006,'ee_006')
```

```
Skew Dist    : 10.296743303081483
Kurtosis Dist: 159.27265035789426
Mean         : 333058.240388343
Std-dev      : 1069150.6863446236
```





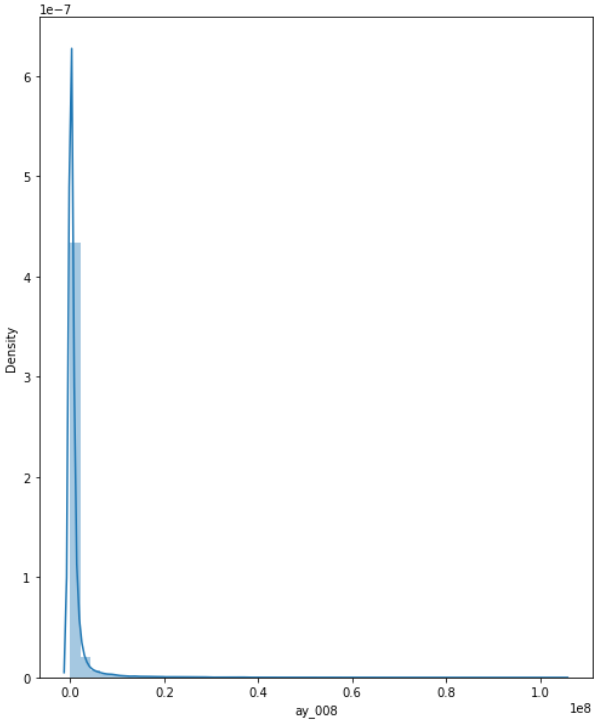


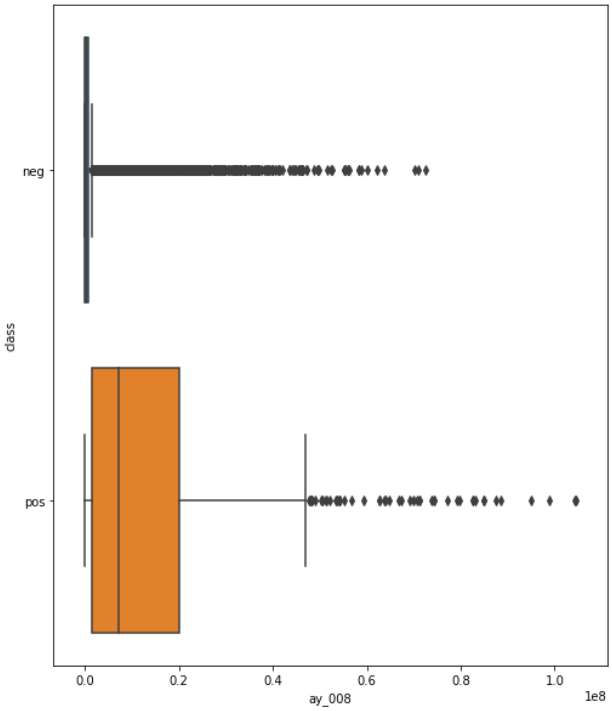
Observation

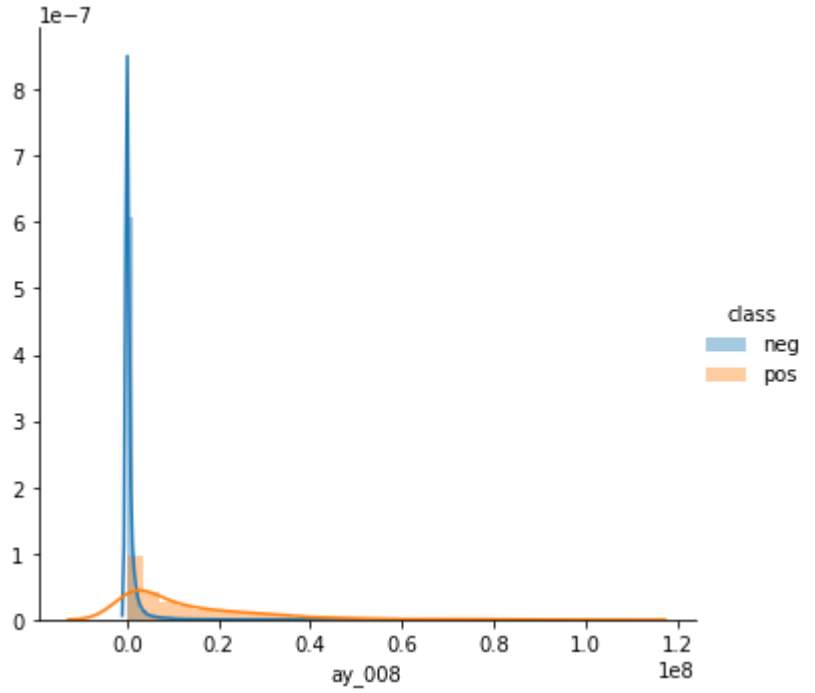
- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- range of neg class is mostly less than 0.2

In [54]:
```
EDA(data.ay_008,'ay_008')
```

```
Skew Dist    : 9.657899076456031
Kurtosis Dist: 132.02092339268398
Mean         : 1051122.964047936
Std-dev      : 3991015.895357171
```







Observation

- the range of neg class is much lower than the postive class
- neg class has high skewness in the histogram than the postive class
- range of neg class is mostly less than 0.1

In [54]:

Bivariate Analysis

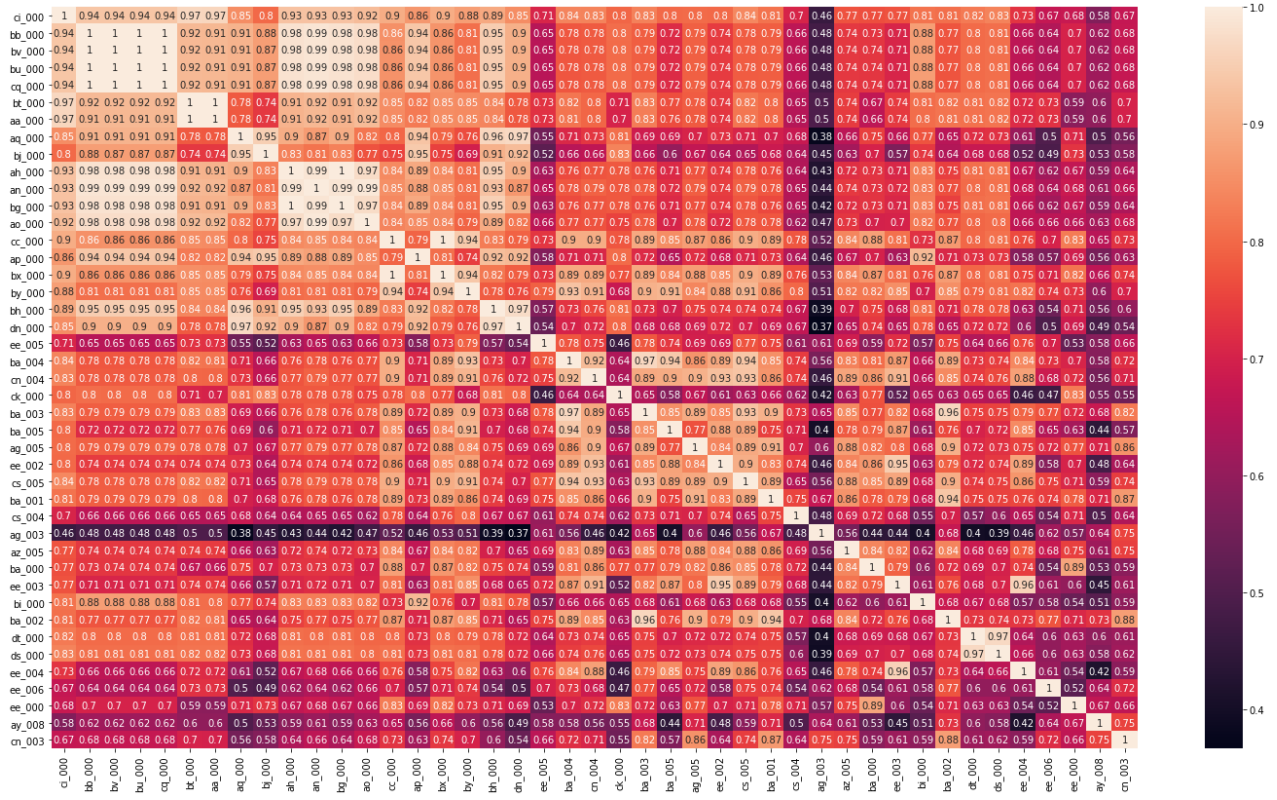
In [55]:
```
newdata = data[important_columns]
newdata.head()
```

Out[55]:

	ci_000	bb_000	bv_000	bu_000	cq_000	bt_000	aa_000	aq_000	bj_000	ah
0	5245752.00	6700214.0	6700214.0	6700214.0	6700214.0	76698.08	76698	1132040.0	799478.0	25516
1	2291079.36	3646660.0	3646660.0	3646660.0	3646660.0	33057.51	33058	338544.0	392208.0	13933
2	2322692.16	2673338.0	2673338.0	2673338.0	2673338.0	41040.08	41040	153698.0	139730.0	12341
3	2135.04	21614.0	21614.0	21614.0	21614.0	12.69	12	1014.0	3090.0	26
4	3565684.80	4289260.0	4289260.0	4289260.0	4289260.0	60874.03	60874	551022.0	399410.0	19740

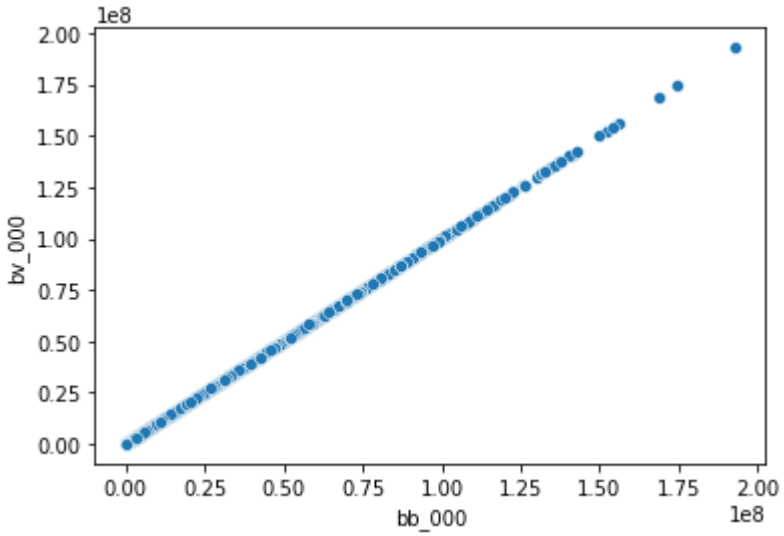
Correlation Matrix

In [56]:
```
plt.figure(figsize=(25, 14))
vg_corr = newdata.corr()
sns.heatmap(vg_corr, xticklabels = vg_corr.columns.values,yticklabels = vg_corr.columns
plt.show()
```



In [57]:
```
sns.scatterplot(data.bb_000,data.bv_000)
```

Out[57]: <matplotlib.axes._subplots.AxesSubplot at 0x7f742351fa10>



In [58]:
```
data = df[important_columns2]
data['class'] = df['class']
data.head()
```

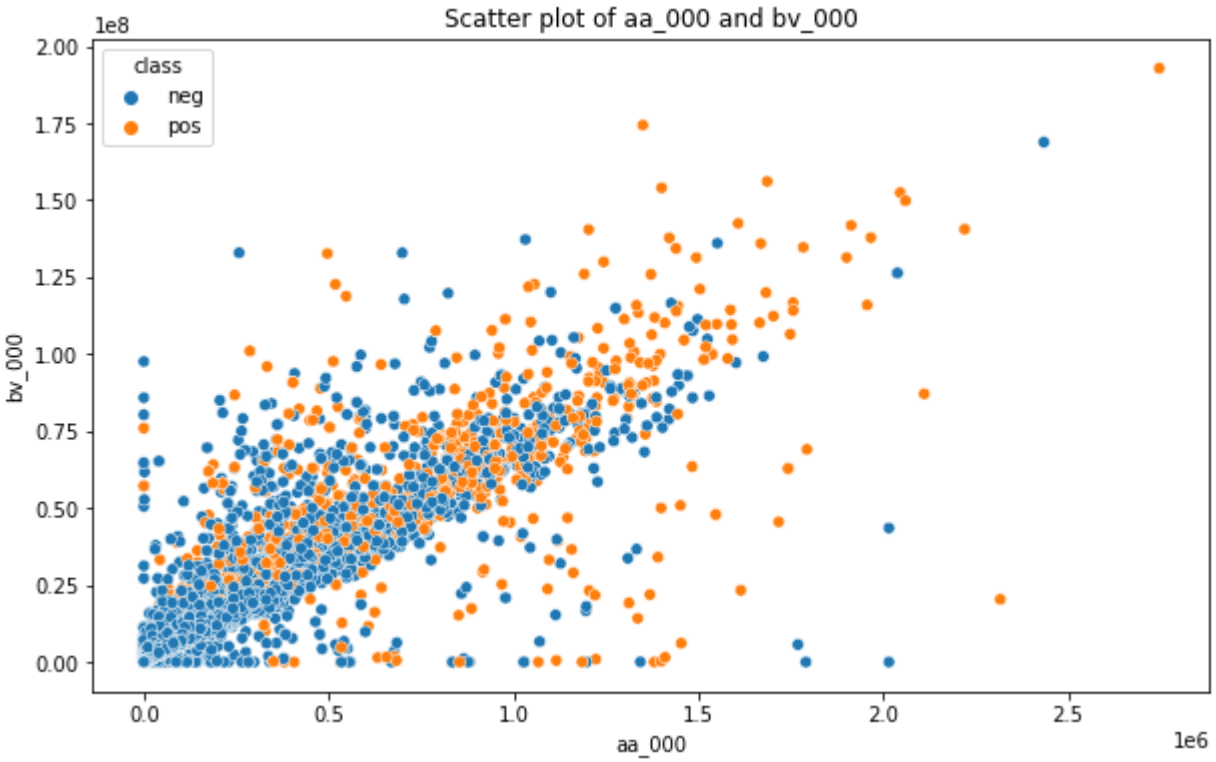
Out[58]:

	ci_000	aq_000	bb_000	cq_000	bu_000	bv_000	bj_000	bt_000	aa_000	cc
0	5245752.00	1132040.0	6700214.0	6700214.0	6700214.0	6700214.0	799478.0	76698.08	76698	61678
1	2291079.36	338544.0	3646660.0	3646660.0	3646660.0	3646660.0	392208.0	33057.51	33058	29428
2	2322692.16	153698.0	2673338.0	2673338.0	2673338.0	2673338.0	139730.0	41040.08	41040	25605
3	2135.04	1014.0	21614.0	21614.0	21614.0	21614.0	3090.0	12.69	12	77
4	3565684.80	551022.0	4289260.0	4289260.0	4289260.0	4289260.0	399410.0	60874.03	60874	39469

Scatter plots

In [59]:
```
plt.figure(figsize=(10,6))
ax = sns.scatterplot(x='aa_000',y='bv_000',data=data,hue='class')
ax.set_title('Scatter plot of aa_000 and bv_000')
```

Out[59]: Text(0.5, 1.0, 'Scatter plot of aa_000 and bv_000')

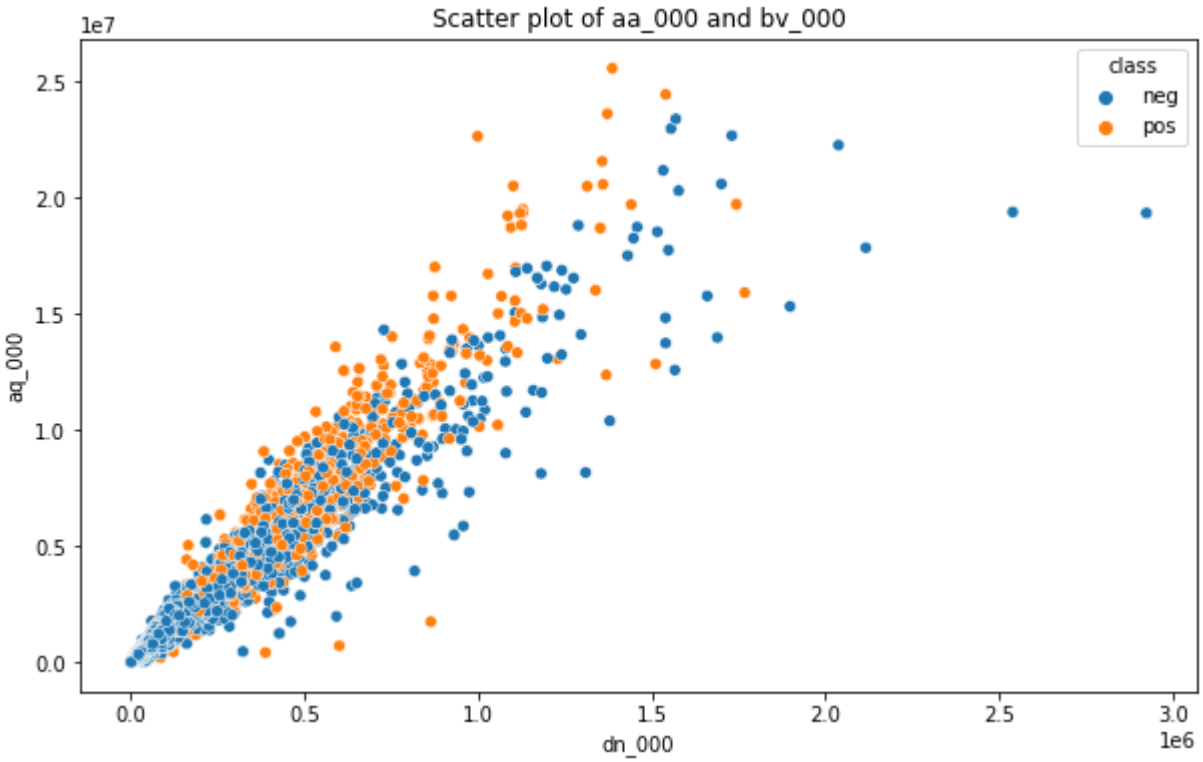


Observation

- BV_000 having less than 0,25 and aa_000 having the less than 0.5 are mostly negtive class data point

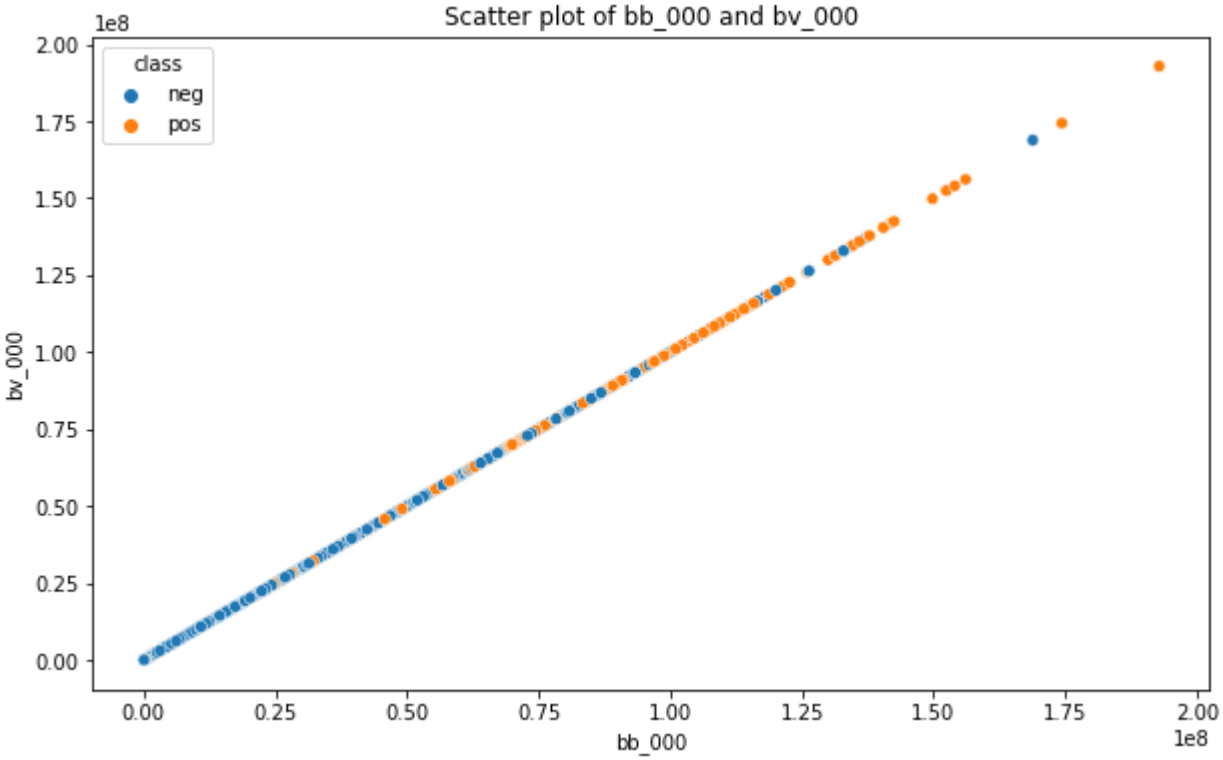
In [60]:
```
plt.figure(figsize=(10,6))
ax = sns.scatterplot(x='dn_000',y='aq_000',data=data,hue='class')
ax.set_title('Scatter plot of aa_000 and bv_000')
```

Out[60]:
```
Text(0.5, 1.0, 'Scatter plot of aa_000 and bv_000')
```



In [61]:
```
plt.figure(figsize=(10,6))
ax = sns.scatterplot(x='bb_000',y='bv_000',data=data,hue='class')
ax.set_title('Scatter plot of bb_000 and bv_000')
```

Out[61]: Text(0.5, 1.0, 'Scatter plot of bb_000 and bv_000')

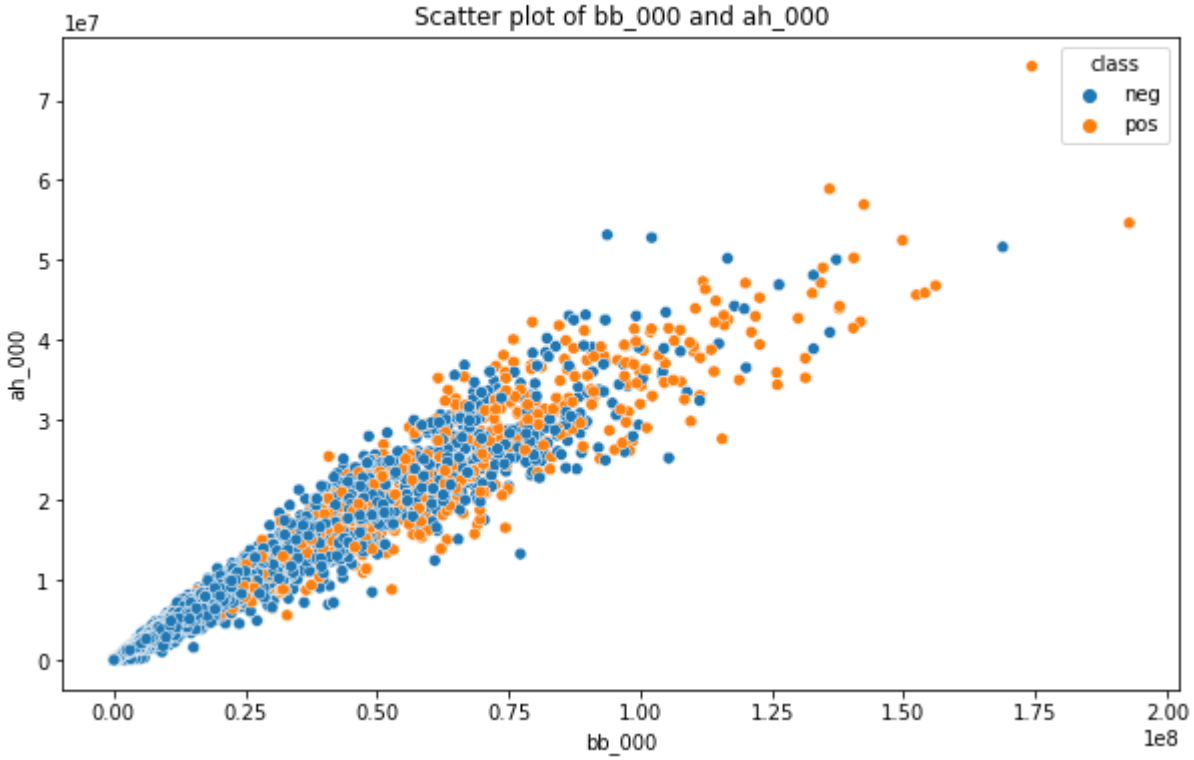


Observation

- mostly less than 0.50 data point are negtive class
- above 1.0 are postive class data point

In [62]:
```
plt.figure(figsize=(10,6))
ax = sns.scatterplot(x='bb_000',y='ah_000',data=data,hue='class')
ax.set_title('Scatter plot of bb_000 and ah_000')
```

Out[62]:
```
Text(0.5, 1.0, 'Scatter plot of bb_000 and ah_000')
```

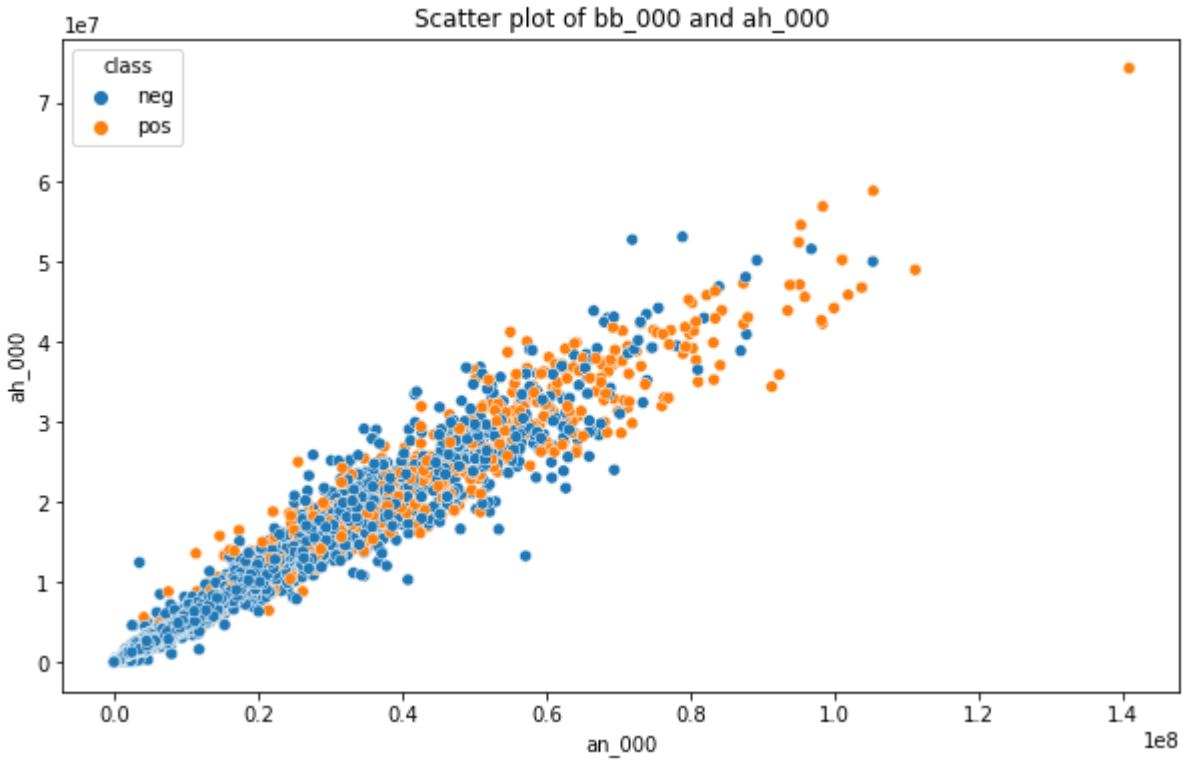


Observation

- ah_000 having less tham 1 and bb_000 having less than 0.25 are negetive class data point

In [63]:
```
plt.figure(figsize=(10,6))
ax = sns.scatterplot(x='an_000',y='ah_000',data=data,hue='class')
ax.set_title('Scatter plot of bb_000 and ah_000')
```

Out[63]:
```
Text(0.5, 1.0, 'Scatter plot of bb_000 and ah_000')
```



In [ ]:

Observation

- data point having ah_000 less than 1 and an_000 haaving less tham 0.2 are mostly neg class data point

Pair plot

In [64]:

```
sns.pairplot(data, hue="class")
plt.show()
```

In [64]: