Saba_cross

October 16, 2022

[10]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import GradientBoostingClassifier
import matplotlib.pylab as plt
from sklearn import metrics
from sklearn.metrics import classification_report
from dmba import plotDecisionTree, gainsChart, liftChart
from dmba import classificationSummary, regressionSummary
from sklearn.metrics import recall_score
%matplotlib inline


import warnings
warnings.filterwarnings('ignore')
```

[11]:
```
checking_dt=pd.DataFrame(pd.read_csv('cross_sell_dataset.tab', sep='\t'))
```

[12]:
```
checking_dt.head()
```

[12]:
```
   cross_buy  acad_title  age  calls  complaints  customer_tenure_months  \
0          0           0   60      0           0                     221
1          0           0   55      0           0                     227
2          0           0   61      0           0                     221
3          0           0   70      0           0                     222
4          0           1   61      0           0                     227

   directmails  gender  joint_account  inflows  …  volume_debit  \
0            0     0.0            0.0      0.0  …          0.00
1            0     0.0            1.0      0.0  …          3.28
2            0     1.0            0.0   3000.0  …      31963.13
3            0     0.0            0.0   6000.0  …      54048.40
4            0     1.0            0.0      0.0  …    1374743.09
```

```
   volume_debit_6months  ext_city_size ext_house_size  ext_purchase_power  \
0                  0.00            7.0            4.0                 5.0
1                  3.28            7.0            1.0                 7.0
2              28963.13            8.0            4.0                 4.0
3              48048.40            2.0            1.0                 7.0
4            1341722.91            4.0            1.0                 7.0

   ext_share_new_houses  ext_share_new_cars ext_car_power  \
0                   4.0                 4.0           3.0
1                   1.0                 3.0           1.0
2                   1.0                 7.0           3.0
3                   3.0                 5.0           4.0
4                   3.0                 6.0           5.0

   ext_living_duration  giro_mailing
0                  9.0             0
1                  NaN             0
2                  9.0             0
3                  9.0             0
4                  7.0             0

[5 rows x 35 columns]
```

[13]:
```
checking_dt.shape
```

[13]:
```
(100000, 35)
```

[14]:
```
checking_dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 35 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   cross_buy                    100000 non-null  int64
 1   acad_title                   100000 non-null  int64
 2   age                          100000 non-null  int64
 3   calls                        100000 non-null  int64
 4   complaints                   100000 non-null  int64
 5   customer_tenure_months       100000 non-null  int64
 6   directmails                  100000 non-null  int64
 7   gender                       99998 non-null   float64
 8   joint_account                99998 non-null   float64
 9   inflows                      99527 non-null   float64
 10  last_acc_opening_days        100000 non-null  int64
 11  logins_desktop               100000 non-null  int64
 12  logins_mobile                100000 non-null  int64
```

```
 13  marital_status              100000 non-null  object
 14  member_get_member_active    100000 non-null  int64
 15  member_get_member_passive   100000 non-null  int64
 16  nr_products                 100000 non-null  int64
 17  occupation                  48725 non-null   object
 18  outflows                    99527 non-null   float64
 19  prod_loan                   100000 non-null  int64
 20  prod_mortgages              100000 non-null  int64
 21  prod_brokerage              100000 non-null  int64
 22  prod_pensionplan            100000 non-null  int64
 23  prod_savings                100000 non-null  int64
 24  relocations                 100000 non-null  int64
 25  volume_debit                100000 non-null  float64
 26  volume_debit_6months        97002 non-null   float64
 27  ext_city_size               97338 non-null   float64
 28  ext_house_size              96953 non-null   float64
 29  ext_purchase_power          95435 non-null   float64
 30  ext_share_new_houses        97338 non-null   float64
 31  ext_share_new_cars          84845 non-null   float64
 32  ext_car_power               89866 non-null   float64
 33  ext_living_duration         90935 non-null   float64
 34  giro_mailing                100000 non-null  int64
dtypes: float64(13), int64(20), object(2)
memory usage: 26.7+ MB
```

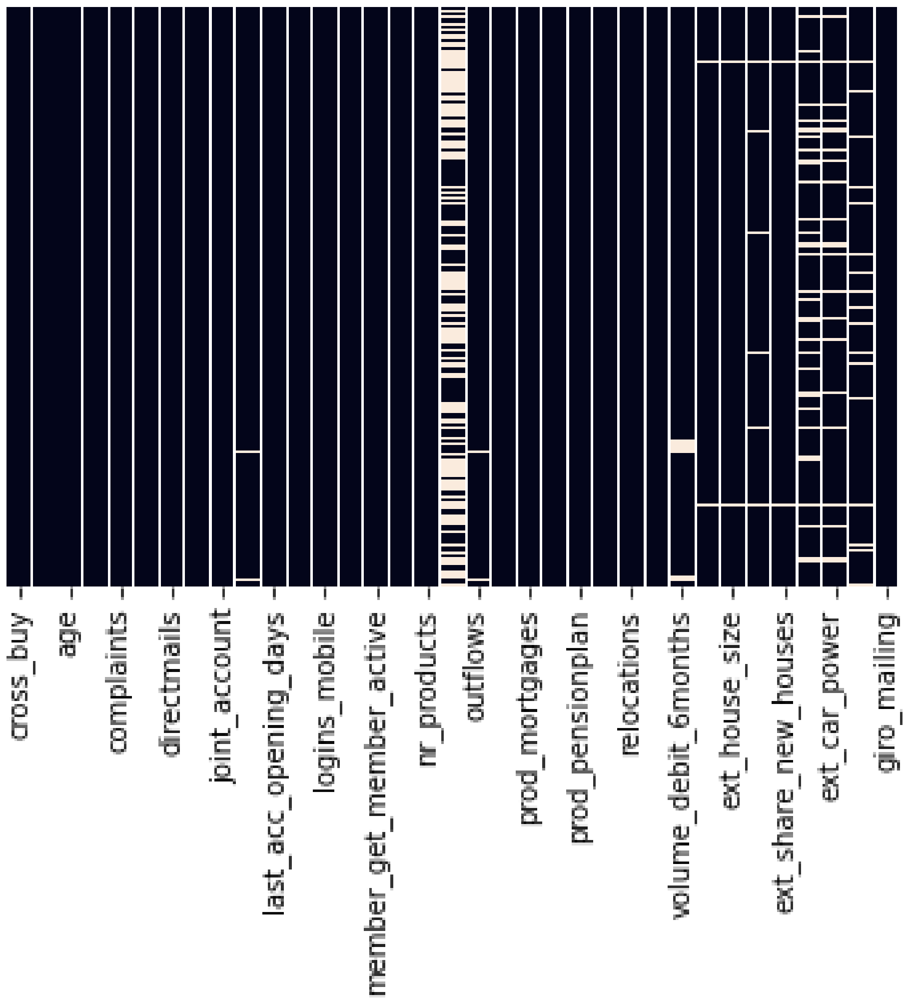
[15]:
```
checking_dt.isna().sum()
```

[15]:
```
cross_buy                        0
acad_title                       0
age                              0
calls                            0
complaints                       0
customer_tenure_months           0
directmails                      0
gender                           2
joint_account                    2
inflows                        473
last_acc_opening_days            0
logins_desktop                   0
logins_mobile                    0
marital_status                   0
member_get_member_active         0
member_get_member_passive        0
nr_products                      0
occupation                   51275
outflows                       473
prod_loan                        0
```

```
prod_mortgages                 0
prod_brokerage                 0
prod_pensionplan               0
prod_savings                   0
relocations                    0
volume_debit                   0
volume_debit_6months        2998
ext_city_size               2662
ext_house_size              3047
ext_purchase_power          4565
ext_share_new_houses        2662
ext_share_new_cars         15155
ext_car_power              10134
ext_living_duration         9065
giro_mailing                   0
dtype: int64
```

[16]:
```
#heat map of missing values
sns.heatmap(checking_dt.isnull(),yticklabels=False,cbar=False)
```

[16]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fecf8724810>
```



[17]:
```
checking_dt.describe()
```

[17]:

	cross_buy	acad_title	age	calls
count	100000.000000	100000.000000	100000.000000	100000.000000
mean	0.100000	0.020390	49.892260	0.104600
std	0.300002	0.141331	14.534085	0.564395
min	0.000000	0.000000	18.000000	0.000000
25%	0.000000	0.000000	39.000000	0.000000
50%	0.000000	0.000000	51.000000	0.000000
75%	0.000000	0.000000	60.000000	0.000000
max	1.000000	1.000000	80.000000	58.000000

	complaints	customer_tenure_months	directmails	gender
count	100000.000000	100000.000000	100000.000000	99998.000000
mean	0.003530	140.181700	0.470660	0.405588
std	0.078598	74.901654	0.915952	0.491008

```
min       0.000000              0.000000       0.000000       0.000000
25%       0.000000             78.000000       0.000000       0.000000
50%       0.000000            160.000000       0.000000       0.000000
75%       0.000000            201.000000       1.000000       1.000000
max       8.000000            567.000000       9.000000       1.000000

       joint_account       inflows  …  volume_debit  volume_debit_6months  \
count   99998.000000  9.952700e+04  …  1.000000e+05          9.700200e+04
mean        0.236675  2.818210e+03  …  2.262297e+04          2.252191e+04
std         0.425043  1.552502e+04  …  1.493077e+05          8.568004e+04
min         0.000000  0.000000e+00  …  0.000000e+00          0.000000e+00
25%         0.000000  0.000000e+00  …  1.247500e+00          2.782500e+00
50%         0.000000  0.000000e+00  …  1.355575e+03          1.404455e+03
75%         0.000000  9.000000e+02  …  1.857914e+04          1.865452e+04
max         1.000000  1.300000e+06  …  3.999071e+07          7.705563e+06

       ext_city_size  ext_house_size  ext_purchase_power  \
count   97338.000000    96953.000000        95435.000000
mean        4.386961        1.779048            4.594677
std         2.379942        1.185036            1.997959
min         1.000000        1.000000            1.000000
25%         2.000000        1.000000            3.000000
50%         4.000000        1.000000            5.000000
75%         7.000000        3.000000            6.000000
max         8.000000        5.000000            7.000000

       ext_share_new_houses  ext_share_new_cars  ext_car_power  \
count          97338.000000        84845.000000   89866.000000
mean               2.638117            4.307337       3.064674
std                1.353302            1.916396       1.464527
min                1.000000            1.000000       1.000000
25%                1.000000            3.000000       2.000000
50%                3.000000            4.000000       3.000000
75%                4.000000            6.000000       4.000000
max                5.000000            7.000000       5.000000

       ext_living_duration   giro_mailing
count         90935.000000  100000.000000
mean              7.161896       0.105080
std               2.194956       0.306658
min               1.000000       0.000000
25%               6.000000       0.000000
50%               8.000000       0.000000
75%               9.000000       0.000000
max               9.000000       1.000000

[8 rows x 33 columns]
```

[18]:
```python
# Correlation
checking_dt.corr()
```

[18]:
```
                              cross_buy  acad_title       age     calls  \
cross_buy                      1.000000   -0.009882 -0.189307  0.040220
acad_title                    -0.009882    1.000000  0.057615  0.004102
age                           -0.189307    0.057615  1.000000  0.027188
calls                          0.040220    0.004102  0.027188  1.000000
complaints                     0.011324    0.004323  0.013403  0.096050
customer_tenure_months        -0.125842    0.011414  0.291242 -0.054208
directmails                    0.072362   -0.010713 -0.209959  0.001855
gender                         0.041760    0.069316 -0.107541 -0.016002
joint_account                 -0.062406   -0.074176  0.192318  0.051165
inflows                        0.043726    0.040241  0.019642  0.056537
last_acc_opening_days         -0.146949   -0.004697  0.204092 -0.105667
logins_desktop                 0.046032    0.015761  0.020374  0.084036
logins_mobile                  0.090828    0.004411 -0.070243  0.061539
member_get_member_active       0.043120    0.001372 -0.024528  0.013942
member_get_member_passive      0.056280    0.003488 -0.054261  0.025706
nr_products                    0.076436    0.031484  0.038399  0.097643
outflows                      -0.011270   -0.034336 -0.032119 -0.046676
prod_loan                      0.027460   -0.028452 -0.070034  0.031063
prod_mortgages                -0.009592   -0.014775 -0.011224  0.101411
prod_brokerage                 0.101928    0.049662  0.009591  0.059143
prod_pensionplan              -0.005006   -0.000246  0.033538  0.005654
prod_savings                   0.019189    0.032476  0.085457 -0.030931
relocations                    0.051068   -0.006044 -0.124478  0.034239
volume_debit                   0.004752    0.033654  0.071497  0.025303
volume_debit_6months           0.009107    0.063094  0.129426  0.051731
ext_city_size                  0.003590    0.056620 -0.006993  0.009223
ext_house_size                 0.019381    0.004447 -0.059520  0.012980
ext_purchase_power             0.015075    0.052988  0.011680  0.004424
ext_share_new_houses           0.004484   -0.013002 -0.025952 -0.004475
ext_share_new_cars            -0.006054    0.029723  0.077255  0.007583
ext_car_power                 -0.002218    0.034030  0.035650  0.008019
ext_living_duration           -0.033288   -0.019744  0.196554 -0.008904
giro_mailing                   0.130787   -0.012288 -0.313721  0.027957

                              complaints  customer_tenure_months  directmails  \
cross_buy                       0.011324               -0.125842     0.072362
acad_title                      0.004323                0.011414    -0.010713
age                             0.013403                0.291242    -0.209959
calls                           0.096050               -0.054208     0.001855
complaints                      1.000000               -0.008546    -0.004048
customer_tenure_months         -0.008546                1.000000    -0.131963
directmails                    -0.004048               -0.131963     1.000000
gender                         -0.004968               -0.002996     0.041327
```

```
joint_account              0.015702               -0.042698   -0.010052
inflows                    0.023050                0.022929    0.026365
last_acc_opening_days     -0.025699                0.763507   -0.192201
logins_desktop             0.023835               -0.004164   -0.012080
logins_mobile              0.018044               -0.070072    0.035730
member_get_member_active   0.014075               -0.016697    0.015492
member_get_member_passive  0.017633               -0.094129    0.005597
nr_products                0.030911                0.082864    0.079356
outflows                  -0.023151               -0.035689   -0.043571
prod_loan                 -0.000004               -0.268161    0.162420
prod_mortgages             0.037286               -0.120971   -0.015122
prod_brokerage             0.019865                0.046196    0.064405
prod_pensionplan           0.003544                0.012568    0.006056
prod_savings              -0.008715                0.371350   -0.032924
relocations                0.006176               -0.069396    0.082247
volume_debit               0.012073                0.050782    0.017487
volume_debit_6months       0.025577                0.081041    0.035098
ext_city_size              0.006214               -0.015300    0.020206
ext_house_size             0.005924               -0.061752    0.033382
ext_purchase_power         0.009849               -0.003221    0.011723
ext_share_new_houses      -0.003605               -0.011169   -0.001411
ext_share_new_cars         0.005436                0.041144   -0.012223
ext_car_power              0.006597                0.011970   -0.005831
ext_living_duration       -0.005177                0.147168   -0.053855
giro_mailing              -0.000039               -0.177364    0.270552

                             gender  joint_account   inflows  ...  \
cross_buy                  0.041760      -0.062406  0.043726  ...
acad_title                 0.069316      -0.074176  0.040241  ...
age                       -0.107541       0.192318  0.019642  ...
calls                     -0.016002       0.051165  0.056537  ...
complaints                -0.004968       0.015702  0.023050  ...
customer_tenure_months    -0.002996      -0.042698  0.022929  ...
directmails                0.041327      -0.010052  0.026365  ...
gender                     1.000000      -0.459960  0.028738  ...
joint_account             -0.459960       1.000000 -0.007882  ...
inflows                    0.028738      -0.007882  1.000000  ...
last_acc_opening_days     -0.013485      -0.060886 -0.038986  ...
logins_desktop             0.049443      -0.003702  0.123411  ...
logins_mobile              0.084330      -0.031938  0.106936  ...
member_get_member_active   0.020597      -0.012256  0.029229  ...
member_get_member_passive  0.017415      -0.029128  0.035607  ...
nr_products                0.032865       0.078398  0.115144  ...
outflows                  -0.015884       0.003594 -0.320495  ...
prod_loan                  0.032143       0.082855 -0.047325  ...
prod_mortgages            -0.147388       0.346461 -0.030526  ...
prod_brokerage             0.141553      -0.067355  0.146753  ...
```

prod_pensionplan	-0.012519	0.004465	0.001435	…
prod_savings	0.054232	-0.191463	0.107790	…
relocations	0.014575	-0.031810	0.010644	…
volume_debit	0.012378	-0.004712	0.190881	…
volume_debit_6months	0.017588	-0.005617	0.232633	…
ext_city_size	0.005032	-0.052038	0.009528	…
ext_house_size	0.030996	-0.117709	-0.009599	…
ext_purchase_power	-0.002775	0.023079	0.036708	…
ext_share_new_houses	0.000886	0.015665	0.004767	…
ext_share_new_cars	-0.020855	0.019138	0.019496	…
ext_car_power	0.003049	-0.008803	0.022106	…
ext_living_duration	-0.026423	0.052400	-0.002667	…
giro_mailing	0.054532	-0.089840	0.044425	…

	volume_debit	volume_debit_6months	ext_city_size	\
cross_buy	0.004752	0.009107	0.003590	
acad_title	0.033654	0.063094	0.056620	
age	0.071497	0.129426	-0.006993	
calls	0.025303	0.051731	0.009223	
complaints	0.012073	0.025577	0.006214	
customer_tenure_months	0.050782	0.081041	-0.015300	
directmails	0.017487	0.035098	0.020206	
gender	0.012378	0.017588	0.005032	
joint_account	-0.004712	-0.005617	-0.052038	
inflows	0.190881	0.232633	0.009528	
last_acc_opening_days	-0.005420	-0.019670	-0.021383	
logins_desktop	0.076987	0.124372	0.000187	
logins_mobile	0.026915	0.045893	0.008981	
member_get_member_active	0.014987	0.029838	0.002801	
member_get_member_passive	0.000352	0.002767	0.007130	
nr_products	0.109444	0.190212	0.018215	
outflows	-0.158221	-0.424953	-0.004442	
prod_loan	-0.049536	-0.083790	-0.023308	
prod_mortgages	-0.034506	-0.060009	0.004798	
prod_brokerage	0.140670	0.247727	0.027899	
prod_pensionplan	0.021139	0.039274	-0.005896	
prod_savings	0.083256	0.141969	0.007125	
relocations	-0.009145	-0.015994	0.021371	
volume_debit	1.000000	0.612161	0.013689	
volume_debit_6months	0.612161	1.000000	0.017154	
ext_city_size	0.013689	0.017154	1.000000	
ext_house_size	-0.007614	-0.010971	0.451141	
ext_purchase_power	0.030468	0.050879	0.098991	
ext_share_new_houses	-0.007203	-0.012238	-0.239034	
ext_share_new_cars	0.026716	0.046391	0.073007	
ext_car_power	0.024850	0.042948	0.065945	
ext_living_duration	0.014172	0.033426	-0.088390	

```
giro_mailing                -0.004722            -0.008093       0.016330

                          ext_house_size  ext_purchase_power  \
cross_buy                       0.019381            0.015075
acad_title                      0.004447            0.052988
age                            -0.059520            0.011680
calls                           0.012980            0.004424
complaints                      0.005924            0.009849
customer_tenure_months         -0.061752           -0.003221
directmails                     0.033382            0.011723
gender                          0.030996           -0.002775
joint_account                  -0.117709            0.023079
inflows                        -0.009599            0.036708
last_acc_opening_days          -0.049470           -0.025504
logins_desktop                  0.006230            0.019049
logins_mobile                   0.019784            0.011753
member_get_member_active        0.004710            0.003044
member_get_member_passive       0.011572            0.002823
nr_products                    -0.025351            0.065181
outflows                        0.014802           -0.028468
prod_loan                       0.022999           -0.038335
prod_mortgages                 -0.062665            0.038343
prod_brokerage                  0.009465            0.066249
prod_pensionplan               -0.002977           -0.003345
prod_savings                   -0.005448            0.027171
relocations                     0.068742           -0.015441
volume_debit                   -0.007614            0.030468
volume_debit_6months           -0.010971            0.050879
ext_city_size                   0.451141            0.098991
ext_house_size                  1.000000           -0.187804
ext_purchase_power             -0.187804            1.000000
ext_share_new_houses           -0.119239           -0.012093
ext_share_new_cars             -0.058746            0.161210
ext_car_power                  -0.015200            0.180932
ext_living_duration            -0.141131           -0.021162
giro_mailing                    0.037636            0.008221

                          ext_share_new_houses  ext_share_new_cars  \
cross_buy                             0.004484           -0.006054
acad_title                           -0.013002            0.029723
age                                  -0.025952            0.077255
calls                                -0.004475            0.007583
complaints                           -0.003605            0.005436
customer_tenure_months               -0.011169            0.041144
directmails                          -0.001411           -0.012223
gender                                0.000886           -0.020855
joint_account                         0.015665            0.019138
```

inflows	0.004767	0.019496
last_acc_opening_days	-0.004915	0.021040
logins_desktop	0.001205	0.007124
logins_mobile	-0.000421	0.002445
member_get_member_active	-0.002535	-0.001395
member_get_member_passive	-0.000270	-0.001042
nr_products	0.001934	0.030629
outflows	0.000181	-0.016452
prod_loan	0.002571	-0.043396
prod_mortgages	0.023283	0.004463
prod_brokerage	-0.005131	0.029828
prod_pensionplan	-0.005805	0.003933
prod_savings	-0.014495	0.037408
relocations	0.001984	-0.009040
volume_debit	-0.007203	0.026716
volume_debit_6months	-0.012238	0.046391
ext_city_size	-0.239034	0.073007
ext_house_size	-0.119239	-0.058746
ext_purchase_power	-0.012093	0.161210
ext_share_new_houses	1.000000	-0.070569
ext_share_new_cars	-0.070569	1.000000
ext_car_power	-0.037692	0.146037
ext_living_duration	-0.013951	0.035083
giro_mailing	-0.002204	-0.013015

	ext_car_power	ext_living_duration	giro_mailing
cross_buy	-0.002218	-0.033288	0.130787
acad_title	0.034030	-0.019744	-0.012288
age	0.035650	0.196554	-0.313721
calls	0.008019	-0.008904	0.027957
complaints	0.006597	-0.005177	-0.000039
customer_tenure_months	0.011970	0.147168	-0.177364
directmails	-0.005831	-0.053855	0.270552
gender	0.003049	-0.026423	0.054532
joint_account	-0.008803	0.052400	-0.089840
inflows	0.022106	-0.002667	0.044425
last_acc_opening_days	0.000514	0.113907	-0.209515
logins_desktop	0.010031	0.000953	-0.020129
logins_mobile	0.001145	-0.022142	0.118943
member_get_member_active	0.003921	-0.003746	0.021948
member_get_member_passive	0.003156	-0.007500	0.073406
nr_products	0.023779	-0.004710	0.077915
outflows	-0.021374	-0.006269	-0.013284
prod_loan	-0.021233	-0.056570	0.013612
prod_mortgages	-0.008567	-0.040746	-0.043861
prod_brokerage	0.032512	0.001098	0.129173
prod_pensionplan	0.002607	0.012055	-0.010247

```
prod_savings                    0.025131           0.060480      0.042411
relocations                    -0.000892          -0.069009      0.110319
volume_debit                    0.024850           0.014172     -0.004722
volume_debit_6months            0.042948           0.033426     -0.008093
ext_city_size                   0.065945          -0.088390      0.016330
ext_house_size                 -0.015200          -0.141131      0.037636
ext_purchase_power              0.180932          -0.021162      0.008221
ext_share_new_houses           -0.037692          -0.013951     -0.002204
ext_share_new_cars              0.146037           0.035083     -0.013015
ext_car_power                   1.000000           0.014130      0.001550
ext_living_duration             0.014130           1.000000     -0.042524
giro_mailing                    0.001550          -0.042524      1.000000

[33 rows x 33 columns]
```

[19]:
```
#columns
checking_dt.columns
```

[19]:
```
Index(['cross_buy', 'acad_title', 'age', 'calls', 'complaints',
       'customer_tenure_months', 'directmails', 'gender', 'joint_account',
       'inflows', 'last_acc_opening_days', 'logins_desktop', 'logins_mobile',
       'marital_status', 'member_get_member_active',
       'member_get_member_passive', 'nr_products', 'occupation', 'outflows',
       'prod_loan', 'prod_mortgages', 'prod_brokerage', 'prod_pensionplan',
       'prod_savings', 'relocations', 'volume_debit', 'volume_debit_6months',
       'ext_city_size', 'ext_house_size', 'ext_purchase_power',
       'ext_share_new_houses', 'ext_share_new_cars', 'ext_car_power',
       'ext_living_duration', 'giro_mailing'],
      dtype='object')
```

[20]:
```
checking_dt['acad_title'].value_counts()
```

[20]:
```
0    97961
1     2039
Name: acad_title, dtype: int64
```

[21]:
```
checking_dt['complaints'].value_counts()
```

[21]:
```
0    99719
1      240
2       25
3        9
4        3
5        2
6        1
8        1
Name: complaints, dtype: int64
```

[22]:
```
checking_dt['directmails'].value_counts()
```

[22]:
```
0    71126
1    17318
2     7621
3     2480
4      744
5      331
6      248
7      112
8       17
9        3
Name: directmails, dtype: int64
```

[23]:
```
checking_dt['gender'].value_counts()
```

[23]:
```
0.0    59440
1.0    40558
Name: gender, dtype: int64
```

[24]:
```
checking_dt['joint_account'].value_counts()
```

[24]:
```
0.0    76331
1.0    23667
Name: joint_account, dtype: int64
```

[25]:
```
checking_dt['marital_status'].value_counts()
```

[25]:
```
married       53394
single        33889
unmarried      4559
divorced       3617
widowed        1938
cohabiting     1435
separated      1168
Name: marital_status, dtype: int64
```

[26]:
```
checking_dt['member_get_member_active'].value_counts()
```

[26]:
```
0    99708
1      292
Name: member_get_member_active, dtype: int64
```

[27]:
```
checking_dt['member_get_member_passive'].value_counts()
```

[27]:
```
0    99611
1      389
```

```
Name: member_get_member_passive, dtype: int64
```

[28]:
```
checking_dt['nr_products'].value_counts()
```

[28]:
```
1     68926
2     23129
3      5259
4      1747
5       554
6       237
7        77
8        35
9        16
11        7
10        6
12        3
15        2
14        1
17        1
Name: nr_products, dtype: int64
```

[29]:
```
checking_dt['occupation'].value_counts()
```

[29]:
```
white-collar worker    29272
self-employed           4059
public servant          3942
blue-collar worker      3880
pensioner/retiree       3480
freelancer               984
housewife                877
unemployed               791
university student       499
private means            348
student                  290
apprentice               166
soldier                  137
Name: occupation, dtype: int64
```

[30]:
```
checking_dt['prod_loan'].value_counts()
```

[30]:
```
0    88379
1    11070
2      540
3       10
4        1
Name: prod_loan, dtype: int64
```

[31]:
```
checking_dt['prod_mortgages'].value_counts()
```

[31]:
```
0     90636
1      5271
2      2881
3       856
4       246
5        69
6        24
7         9
8         3
9         2
10        1
13        1
12        1
Name: prod_mortgages, dtype: int64
```

[32]:
```
checking_dt['prod_brokerage'].value_counts()
```

[32]:
```
0     82282
1     16352
2      1155
3       152
4        42
5        13
8         2
7         1
10        1
Name: prod_brokerage, dtype: int64
```

[33]:
```
checking_dt['prod_pensionplan'].value_counts()
```

[33]:
```
0     99703
1       233
2        41
3        12
4         6
5         3
7         1
11        1
Name: prod_pensionplan, dtype: int64
```

[34]:
```
checking_dt['prod_savings'].value_counts()
```

[34]:
```
1     74872
0     17983
2      5765
```

```
3       979
4       254
5       129
6        13
7         2
8         1
10        1
11        1
Name: prod_savings, dtype: int64
```

[35]:
```
checking_dt['relocations'].value_counts()
```

[35]:
```
0    95942
1     3919
2      126
3       13
Name: relocations, dtype: int64
```

[36]:
```
checking_dt['ext_city_size'].value_counts()
```

[36]:
```
4.0    18260
8.0    17987
3.0    14970
1.0    13046
2.0    11425
5.0     8274
7.0     7650
6.0     5726
Name: ext_city_size, dtype: int64
```

[37]:
```
checking_dt['ext_house_size'].value_counts()
```

[37]:
```
1.0    61854
3.0    12451
2.0    10427
4.0     8682
5.0     3539
Name: ext_house_size, dtype: int64
```

[38]:
```
checking_dt['ext_purchase_power'].value_counts()
```

[38]:
```
7.0    22197
6.0    16750
5.0    14529
4.0    12370
3.0    10802
1.0     9491
```

```
2.0     9296
Name: ext_purchase_power, dtype: int64
```

[39]:
```
checking_dt['ext_share_new_houses'].value_counts()
```

[39]:
```
1.0    27920
4.0    20199
3.0    19860
2.0    19434
5.0     9925
Name: ext_share_new_houses, dtype: int64
```

[40]:
```
checking_dt['ext_car_power'].value_counts()
```

[40]:
```
5.0    21521
1.0    18040
2.0    18014
4.0    16864
3.0    15427
Name: ext_car_power, dtype: int64
```

[41]:
```
checking_dt['ext_living_duration'].value_counts()
```

[41]:
```
9.0    39927
7.0    17325
8.0     8529
4.0     5746
5.0     5512
6.0     5372
3.0     4270
2.0     2577
1.0     1677
Name: ext_living_duration, dtype: int64
```

[42]:
```
checking_dt['giro_mailing'].value_counts()
```

[42]:
```
0    89492
1    10508
Name: giro_mailing, dtype: int64
```

[43]:
```
checking_dt['cross_buy'].value_counts()
```

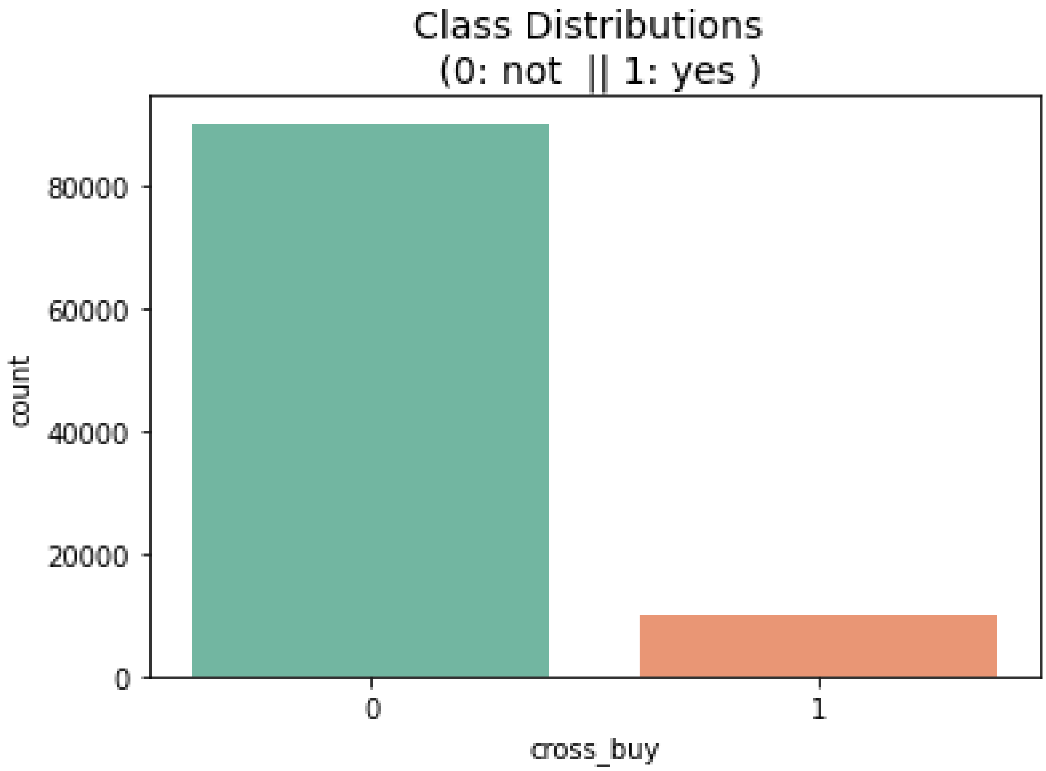
[43]:
```
0    90000
1    10000
Name: cross_buy, dtype: int64
```

[44]:
```
# The classes are unballanced
print('no', round(checking_dt['cross_buy'].value_counts()[0]/len(checking_dt) *␣
↪100,2), '% of the dataset')
print('yes', round(checking_dt['cross_buy'].value_counts()[1]/len(checking_dt)␣
↪* 100,2), '% of the dataset')
```

```
no 90.0 % of the dataset
yes 10.0 % of the dataset
```

[45]:
```
# Visualization of imbalanced data
sns.countplot(checking_dt['cross_buy'], palette = "Set2")
plt.title('Class Distributions \n (0: not  || 1: yes )', fontsize=14)
```

[45]:
```
Text(0.5, 1.0, 'Class Distributions \n (0: not  || 1: yes )')
```

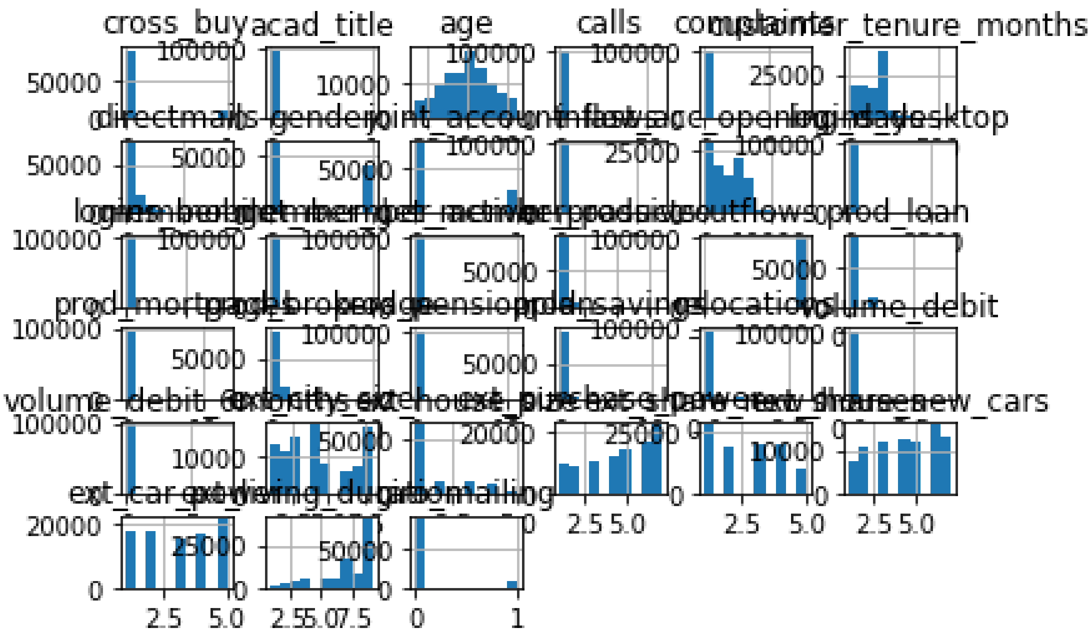


[45]:
```

```

Exploratory Data Analysis

Visualization of Numerical Variables

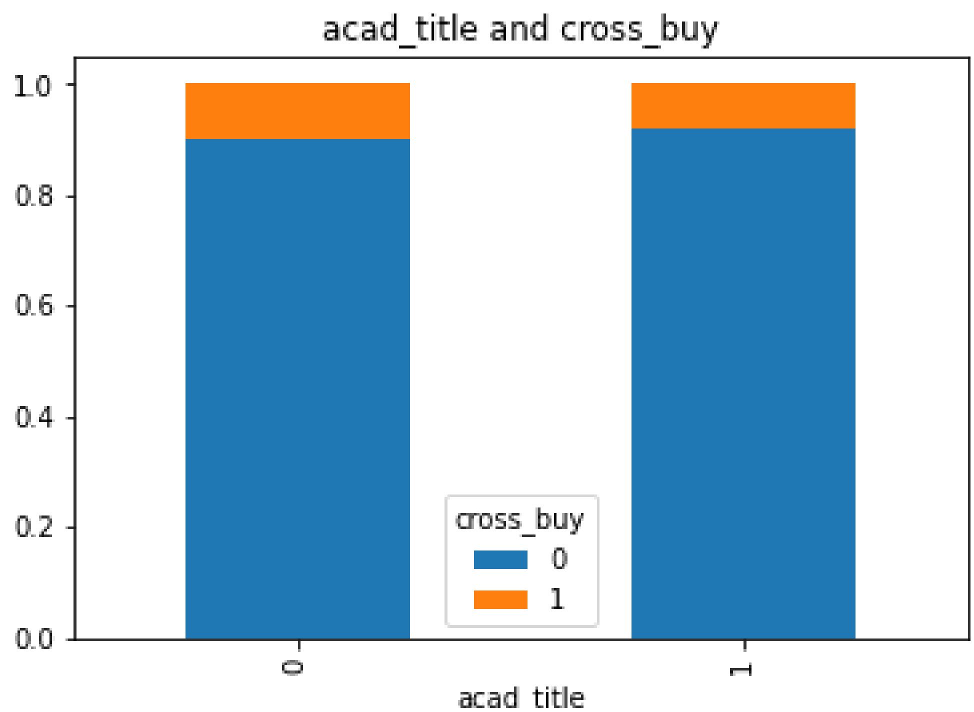
[46]:
```
hist = checking_dt.hist()
```



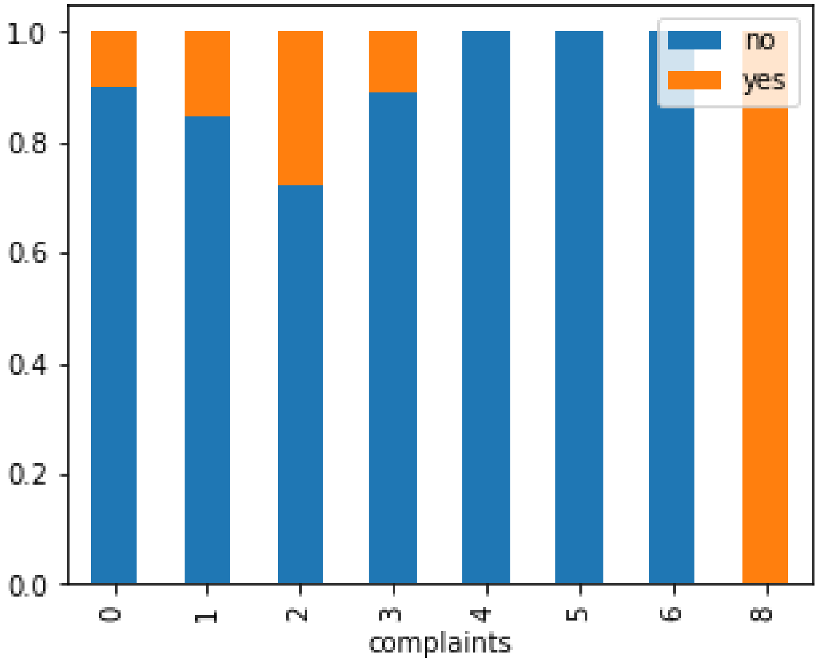
Visualizing the relationship of categorical and the response variable

[50]:
```
#Analysing acad_title and cross_buy
crosstab_01=pd.crosstab(checking_dt['acad_title'],checking_dt['cross_buy'])
crosstab_norm1=crosstab_01.div(crosstab_01.sum(1),axis=0)
crosstab_norm1.plot(kind='bar',stacked=True,title="acad_title and cross_buy")
```

[50]: <matplotlib.axes._subplots.AxesSubplot at 0x7fecf49b7990>

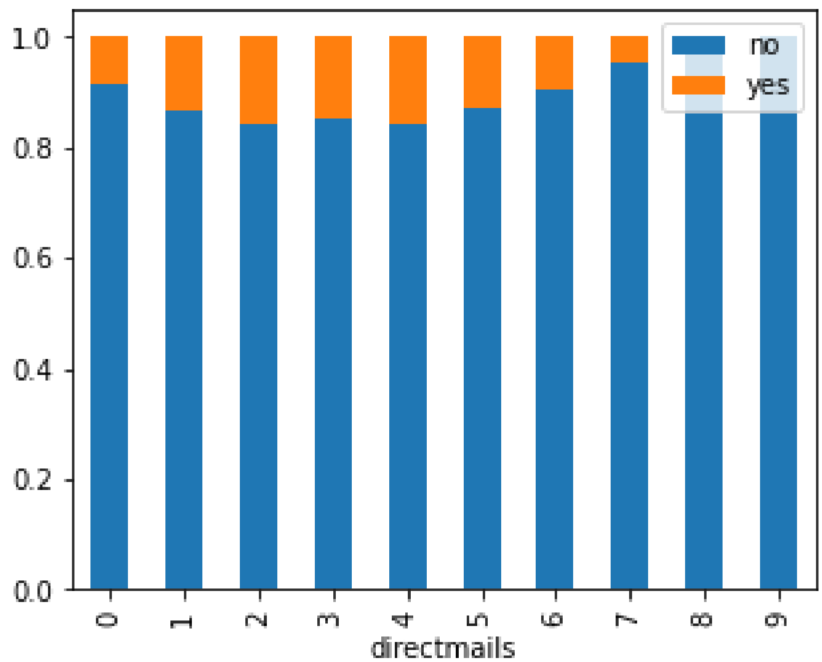


[51]:
```
#Analysing complaints and cross_buy
ax=pd.crosstab(checking_dt.complaints,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                no        yes
complaints
0           0.900180  0.099820
1           0.845833  0.154167
2           0.720000  0.280000
3           0.888889  0.111111
4           1.000000  0.000000
5           1.000000  0.000000
6           1.000000  0.000000
8           0.000000  1.000000
```

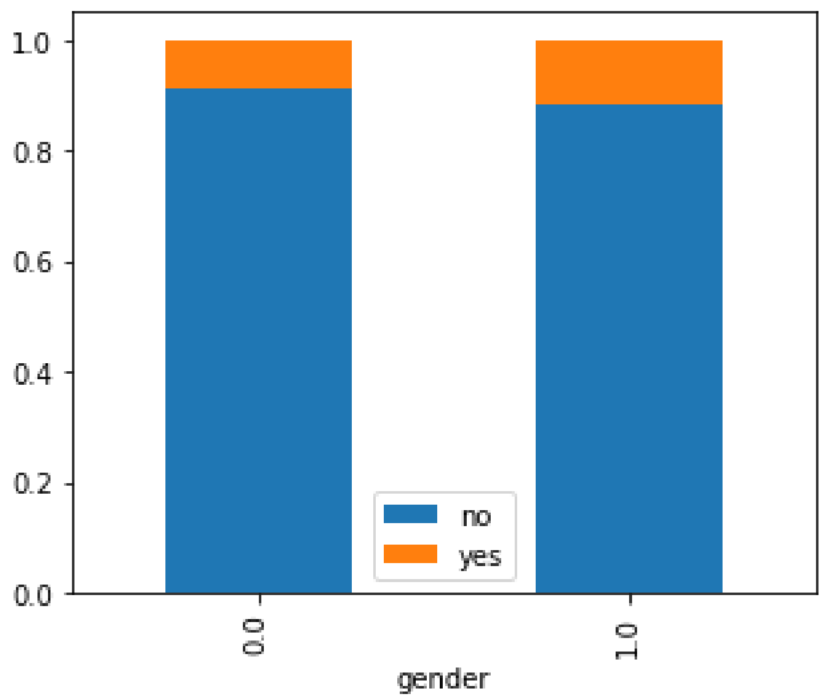
[52]:
```
#Analysing directmails and cross_buy
ax=pd.crosstab(checking_dt.directmails,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                  no       yes
directmails
0           0.916838  0.083162
1           0.865285  0.134715
2           0.843721  0.156279
3           0.850806  0.149194
4           0.840054  0.159946
5           0.870091  0.129909
6           0.903226  0.096774
7           0.955357  0.044643
8           1.000000  0.000000
9           1.000000  0.000000
```

[53]:

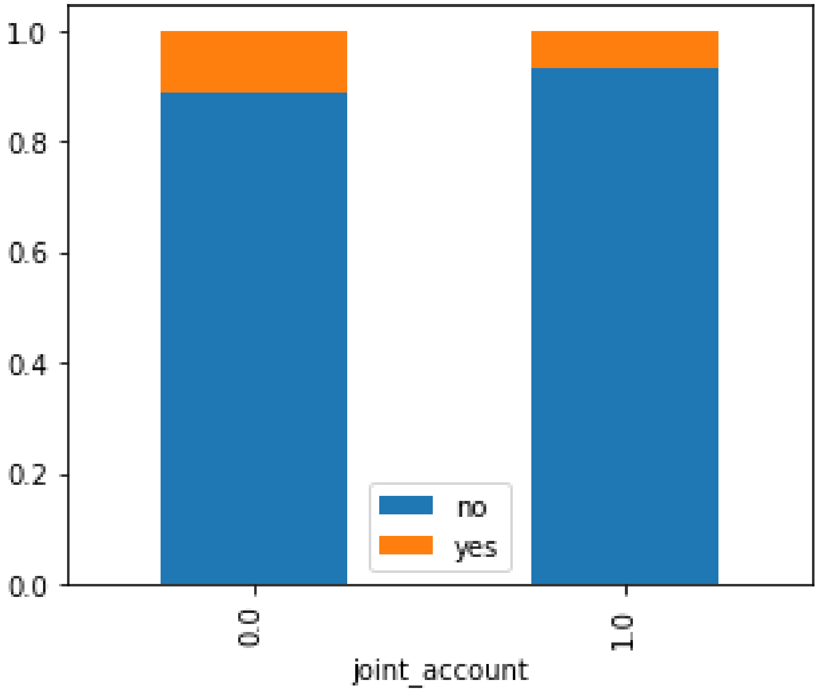
```
#Analysing gender and cross_buy
ax=pd.crosstab(checking_dt.gender,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
             no       yes
gender
0.0     0.910347  0.089653
1.0     0.884832  0.115168
```

[54]:

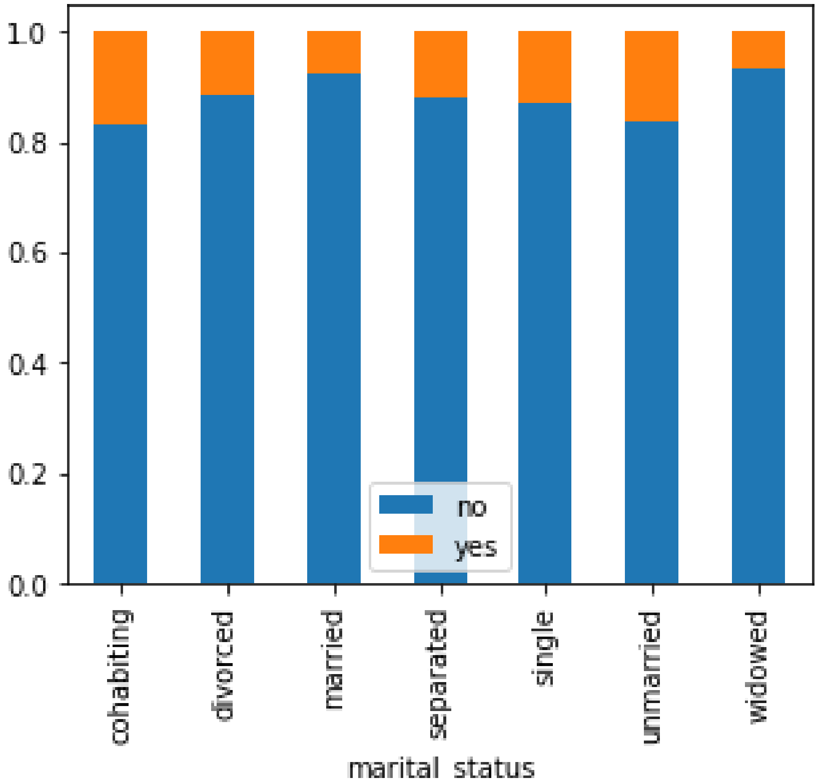
```
#Analysing joint_account and cross_buy
ax=pd.crosstab(checking_dt.joint_account,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                     no       yes
joint_account
0.0            0.889573  0.110427
1.0            0.933621  0.066379
```

[55]:

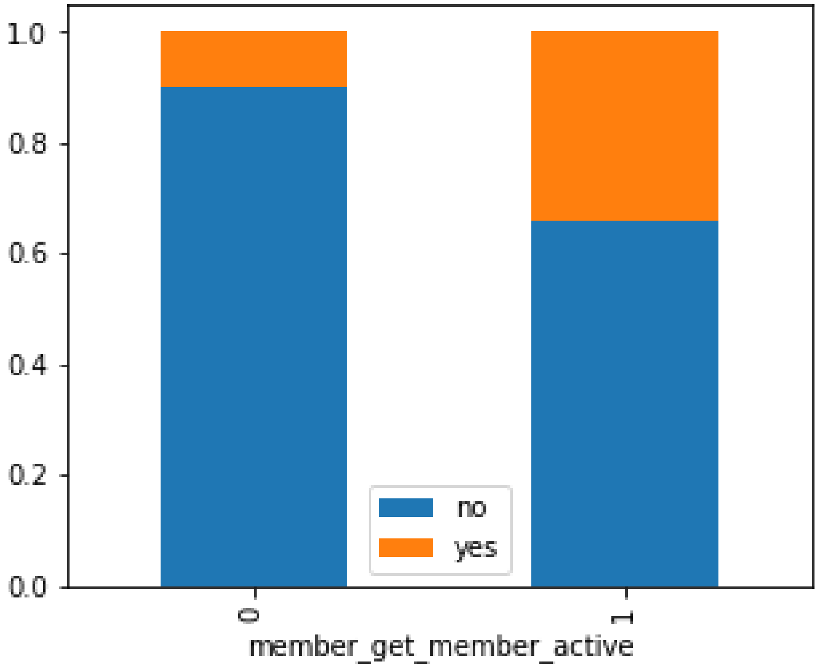
```
#Analysing marital_status and cross_buy
ax=pd.crosstab(checking_dt.marital_status,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                      no       yes
marital_status
cohabiting      0.832056  0.167944
divorced        0.883605  0.116395
married         0.924748  0.075252
separated       0.880993  0.119007
single          0.872614  0.127386
unmarried       0.838342  0.161658
widowed         0.934469  0.065531
```

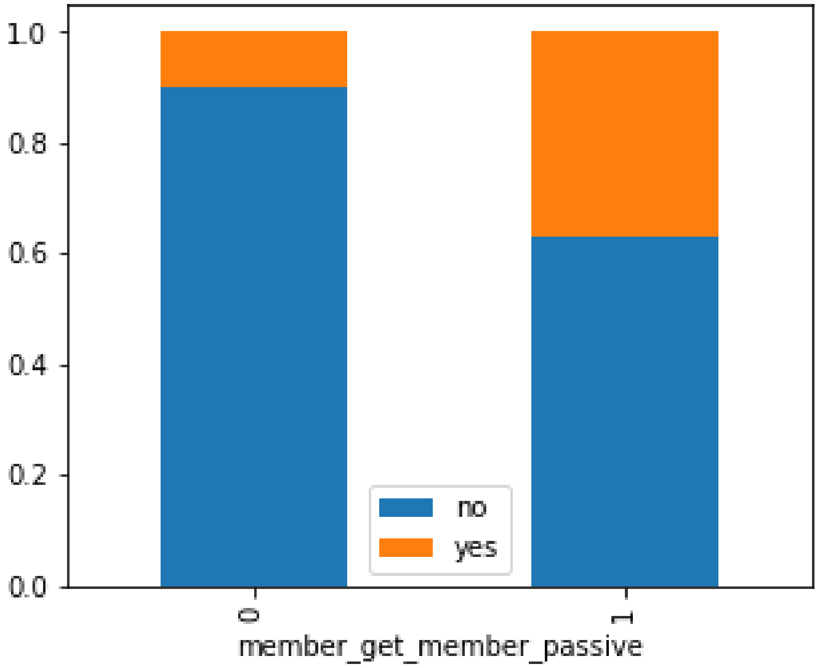
[56]:

```
#Analysing member_get_member_active and cross_buy
ax=pd.crosstab(checking_dt.member_get_member_active,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                              no       yes
member_get_member_active
0                       0.900700  0.099300
1                       0.660959  0.339041
```

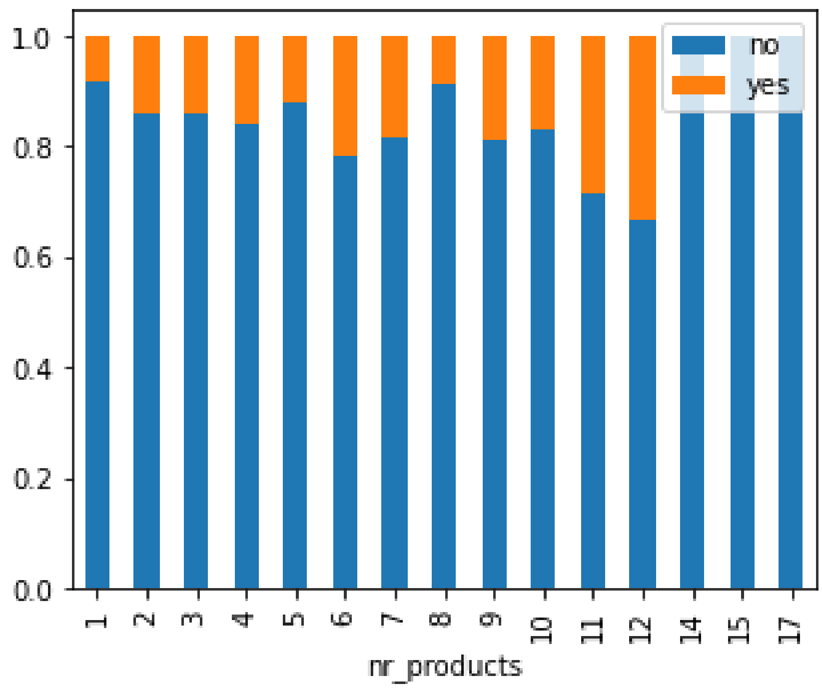
[57]:
```
#Analysing member_get_member_passive and cross_buy
ax=pd.crosstab(checking_dt.member_get_member_passive,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                                no       yes
member_get_member_passive
0                          0.901055  0.098945
1                          0.629820  0.370180
```

[58]:

```
#Analysing nr_products and cross_buy
ax=pd.crosstab(checking_dt.nr_products,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```

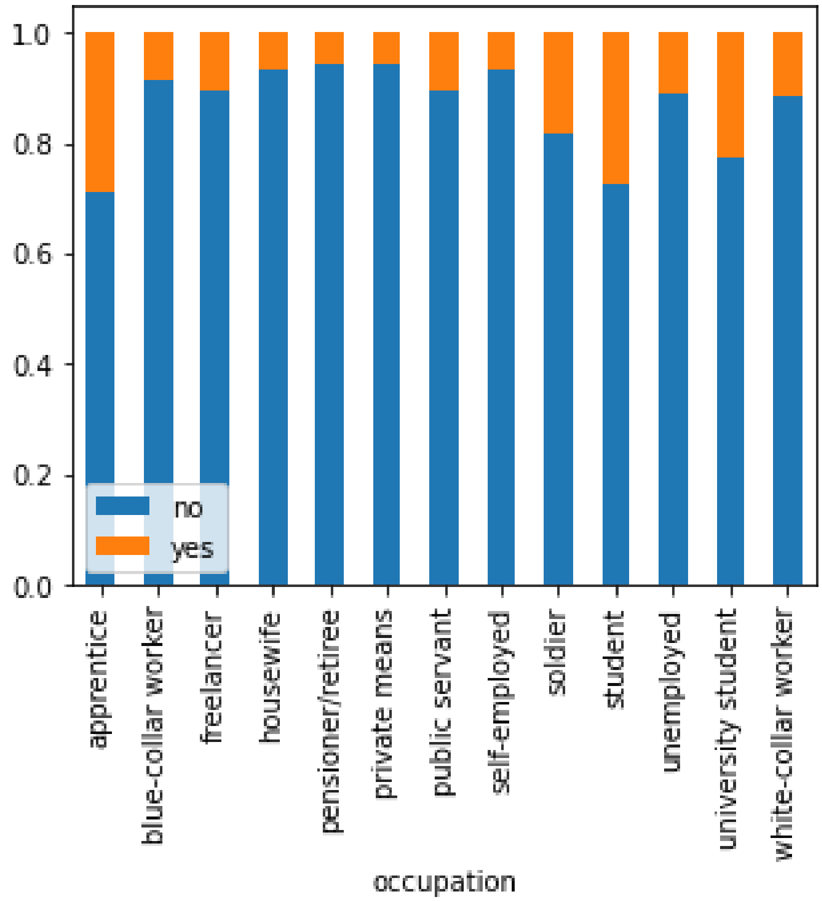



```
                  no       yes
nr_products
1           0.917767  0.082233
2           0.862164  0.137836
3           0.862521  0.137479
4           0.840870  0.159130
5           0.879061  0.120939
6           0.780591  0.219409
7           0.818182  0.181818
8           0.914286  0.085714
9           0.812500  0.187500
10          0.833333  0.166667
11          0.714286  0.285714
12          0.666667  0.333333
14          1.000000  0.000000
15          1.000000  0.000000
17          1.000000  0.000000
```

[59]:

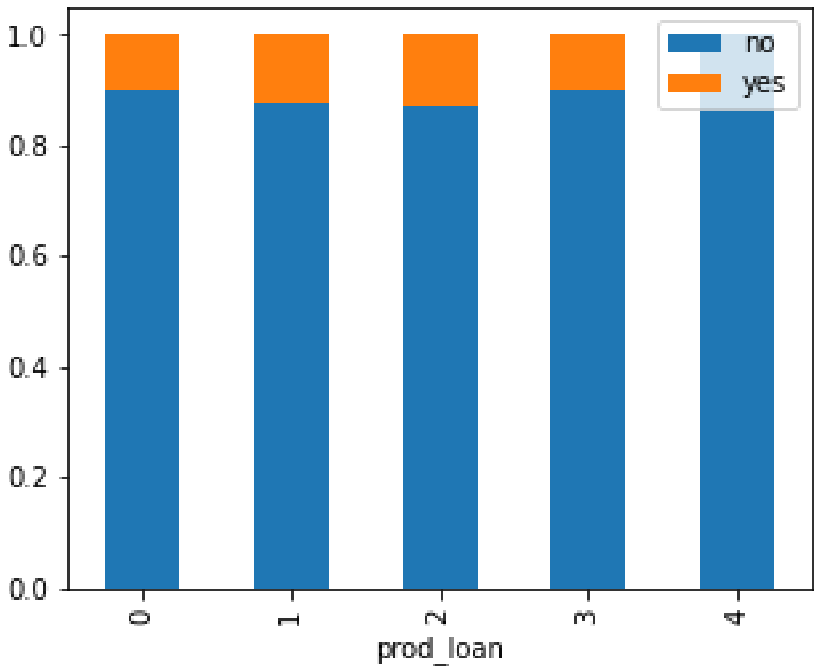
```
#Analysing occupation and cross_buy
ax=pd.crosstab(checking_dt.occupation,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))
```

```
plt.show()
print(ax)
```




```
                          no       yes
occupation
apprentice            0.710843  0.289157
blue-collar worker    0.915206  0.084794
freelancer            0.893293  0.106707
housewife             0.931585  0.068415
pensioner/retiree     0.941667  0.058333
private means         0.942529  0.057471
public servant        0.896499  0.103501
self-employed         0.932742  0.067258
soldier               0.817518  0.182482
student               0.724138  0.275862
unemployed            0.891277  0.108723
university student    0.775551  0.224449
white-collar worker   0.883472  0.116528
```

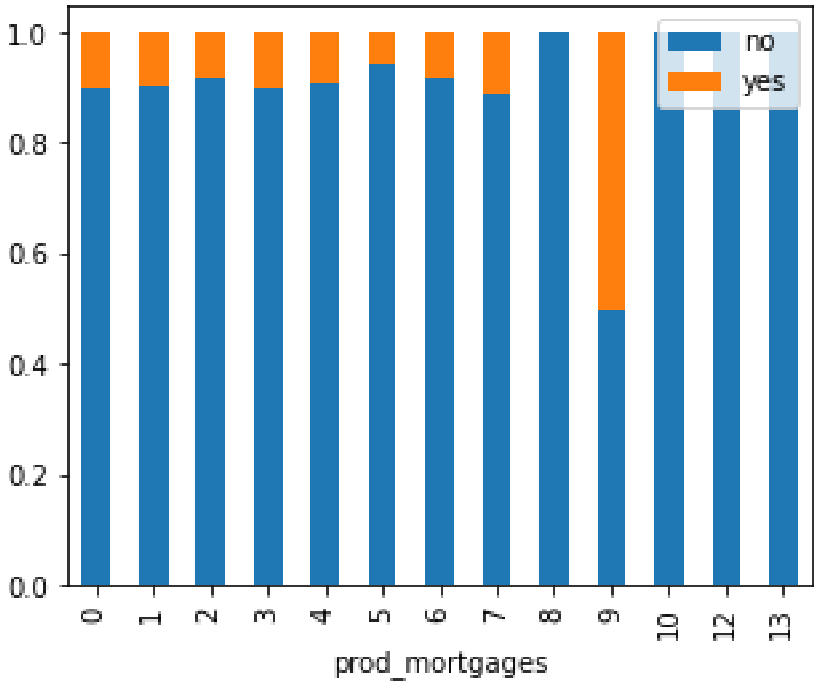
[60]:
```
#Analysing prod_loan and cross_buy
ax=pd.crosstab(checking_dt.prod_loan,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                no       yes
prod_loan
0         0.903031  0.096969
1         0.877236  0.122764
2         0.870370  0.129630
3         0.900000  0.100000
4         1.000000  0.000000
```

[61]:
```
#Analysing prod_mortgages and cross_buy
ax=pd.crosstab(checking_dt.prod_mortgages,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))
```

```
plt.show()
print(ax)
```

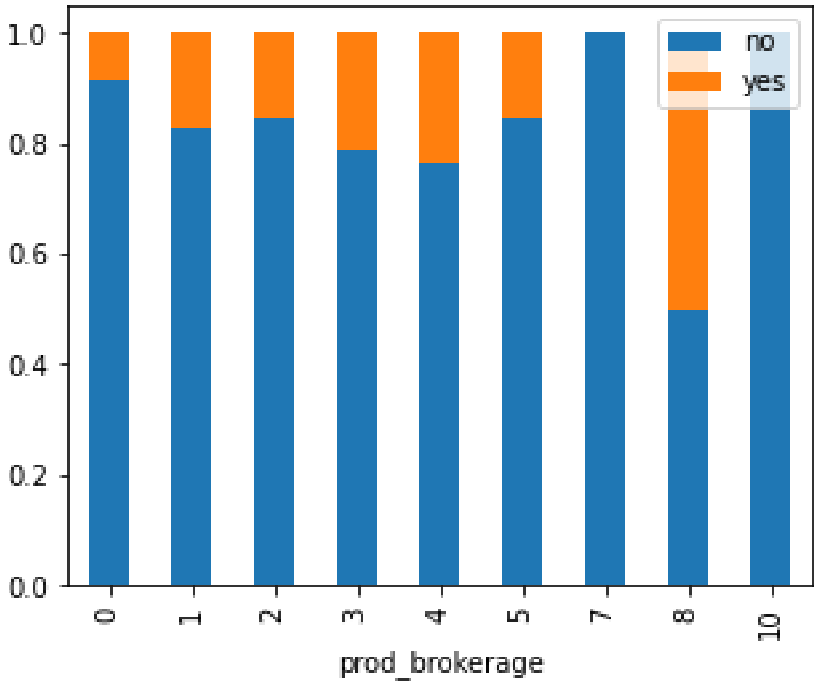



```
                      no       yes
prod_mortgages
0               0.899047  0.100953
1               0.905141  0.094859
2               0.919125  0.080875
3               0.899533  0.100467
4               0.906504  0.093496
5               0.942029  0.057971
6               0.916667  0.083333
7               0.888889  0.111111
8               1.000000  0.000000
9               0.500000  0.500000
10              1.000000  0.000000
12              1.000000  0.000000
13              1.000000  0.000000
```

[62]:

```
#Analysing prod_brokerage and cross_buy
ax=pd.crosstab(checking_dt.prod_brokerage,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
```

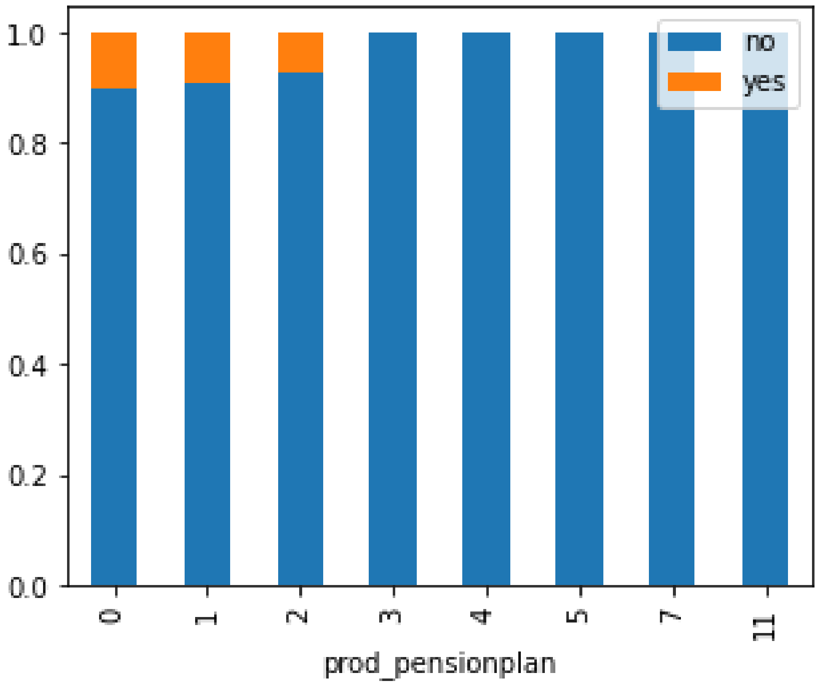
```
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                     no       yes
prod_brokerage
0              0.915109  0.084891
1              0.829195  0.170805
2              0.846753  0.153247
3              0.789474  0.210526
4              0.761905  0.238095
5              0.846154  0.153846
7              1.000000  0.000000
8              0.500000  0.500000
10             1.000000  0.000000
```

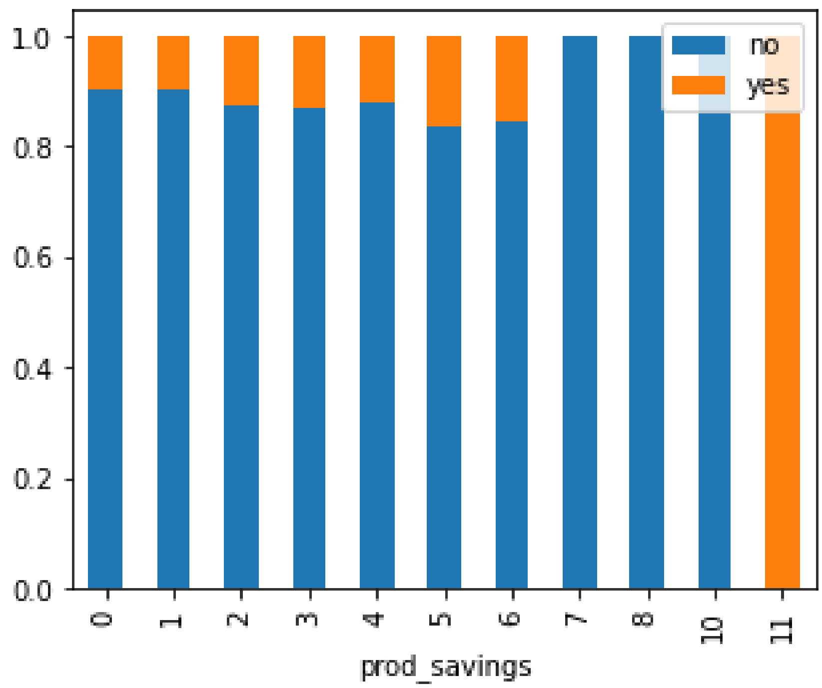
[63]:
```
#Analysing prod_pensionplan and cross_buy
ax=pd.crosstab(checking_dt.prod_pensionplan,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))
```

```
plt.show()
print(ax)
```



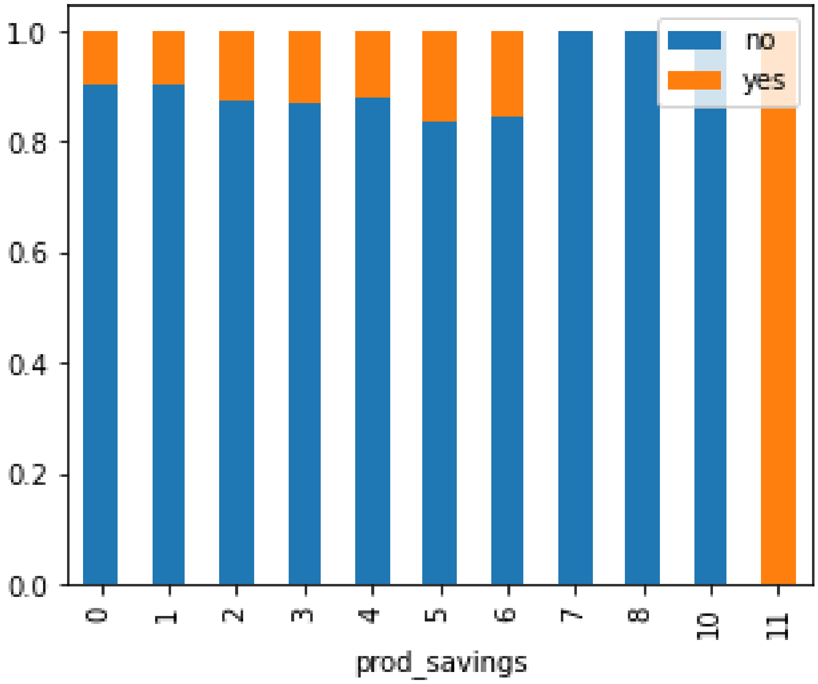

```
                      no       yes
prod_pensionplan
0                 0.899943  0.100057
1                 0.909871  0.090129
2                 0.926829  0.073171
3                 1.000000  0.000000
4                 1.000000  0.000000
5                 1.000000  0.000000
7                 1.000000  0.000000
11                1.000000  0.000000
```

[64]:
```
#Analysing prod_savings and cross_buy
ax=pd.crosstab(checking_dt.prod_savings,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                   no       yes
prod_savings
0            0.903742  0.096258
1            0.901552  0.098448
2            0.875976  0.124024
3            0.869254  0.130746
4            0.877953  0.122047
5            0.837209  0.162791
6            0.846154  0.153846
7            1.000000  0.000000
8            1.000000  0.000000
10           1.000000  0.000000
11           0.000000  1.000000
```

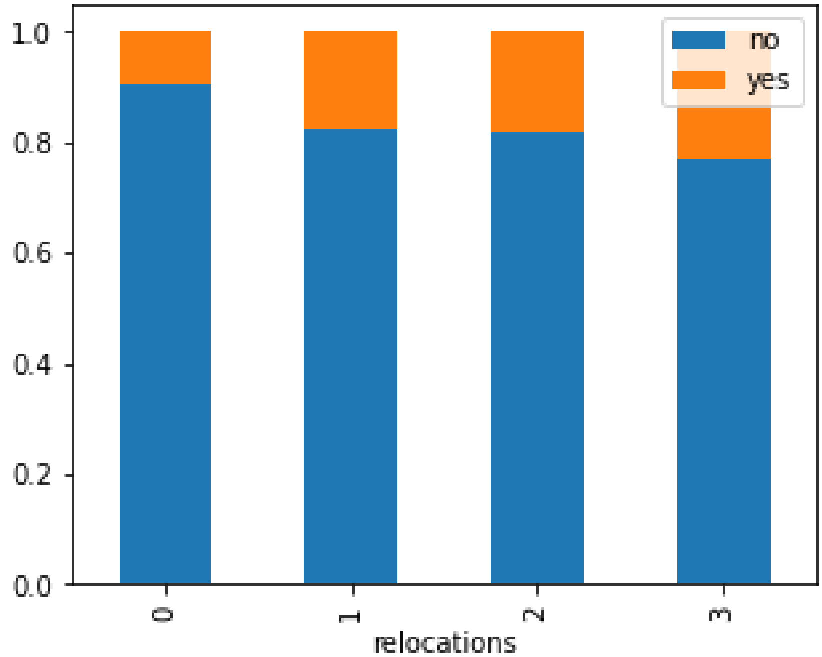
[65]:
```
#Analysing prod_savings and cross_buy
ax=pd.crosstab(checking_dt.prod_savings,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                   no       yes
prod_savings
0            0.903742  0.096258
1            0.901552  0.098448
2            0.875976  0.124024
3            0.869254  0.130746
4            0.877953  0.122047
5            0.837209  0.162791
6            0.846154  0.153846
7            1.000000  0.000000
8            1.000000  0.000000
10           1.000000  0.000000
11           0.000000  1.000000
```

[66]:

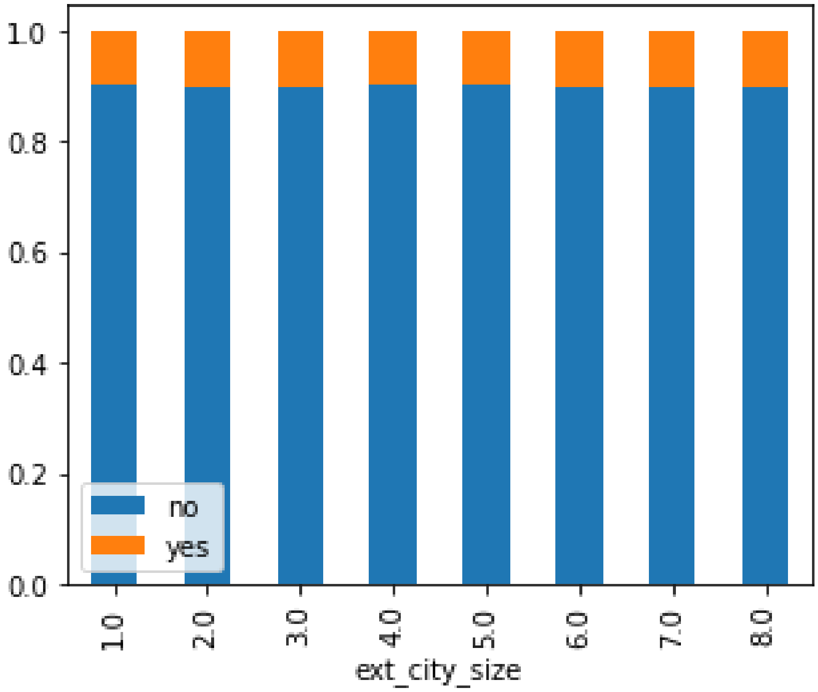
```
#Analysing relocations and cross_buy
ax=pd.crosstab(checking_dt.relocations,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                  no       yes
relocations
0           0.903192  0.096808
1           0.824955  0.175045
2           0.817460  0.182540
3           0.769231  0.230769
```

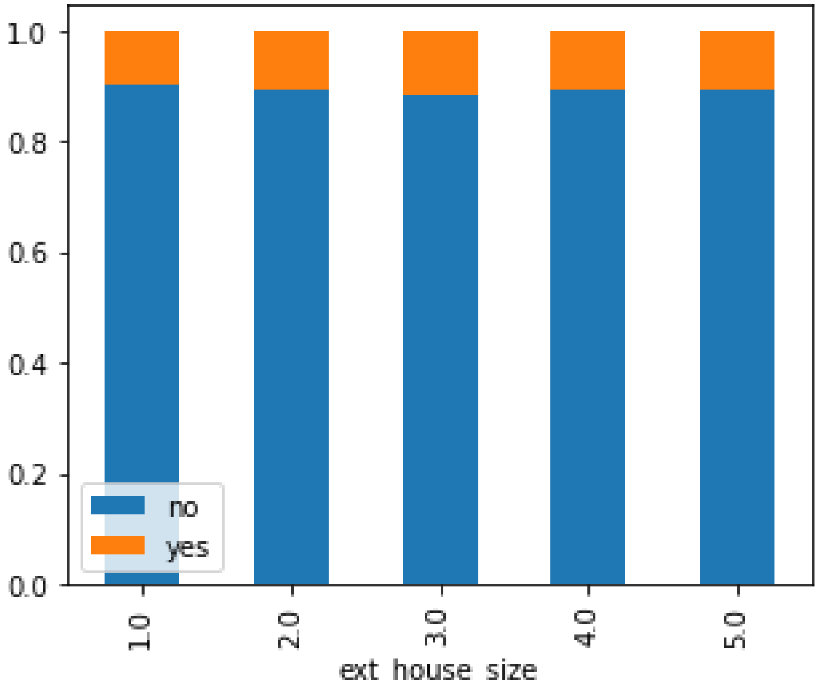
[67]:

```
#Analysing ext_city_size and cross_buy
ax=pd.crosstab(checking_dt.ext_city_size,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                    no       yes
ext_city_size
1.0           0.900736  0.099264
2.0           0.900481  0.099519
3.0           0.897595  0.102405
4.0           0.905422  0.094578
5.0           0.900774  0.099226
6.0           0.899930  0.100070
7.0           0.897255  0.102745
8.0           0.897482  0.102518
```

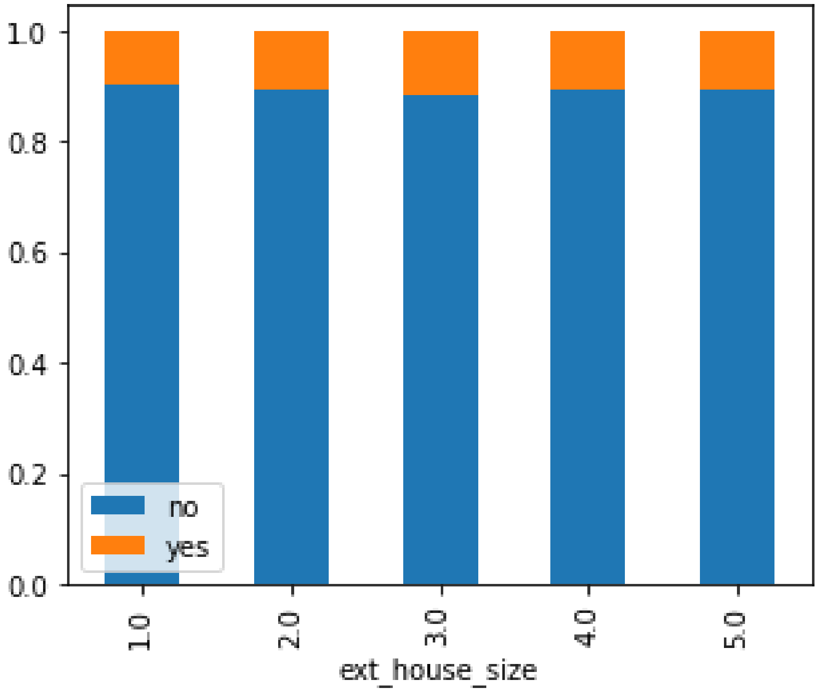
[68]:
```
#Analysing ext_house_size and cross_buy
ax=pd.crosstab(checking_dt.ext_house_size,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                      no       yes
ext_house_size
1.0             0.905406  0.094594
2.0             0.894984  0.105016
3.0             0.885069  0.114931
4.0             0.893803  0.106197
5.0             0.895733  0.104267
```

[69]:

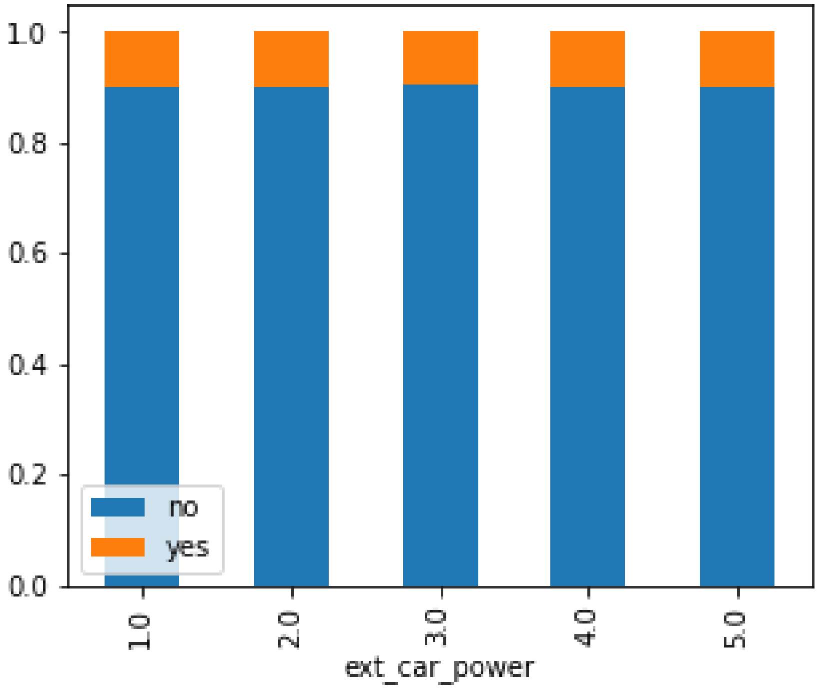
```
#Analysing ext_house_size and cross_buy
ax=pd.crosstab(checking_dt.ext_house_size,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                     no       yes
ext_house_size
1.0             0.905406  0.094594
2.0             0.894984  0.105016
3.0             0.885069  0.114931
4.0             0.893803  0.106197
5.0             0.895733  0.104267
```

[70]:

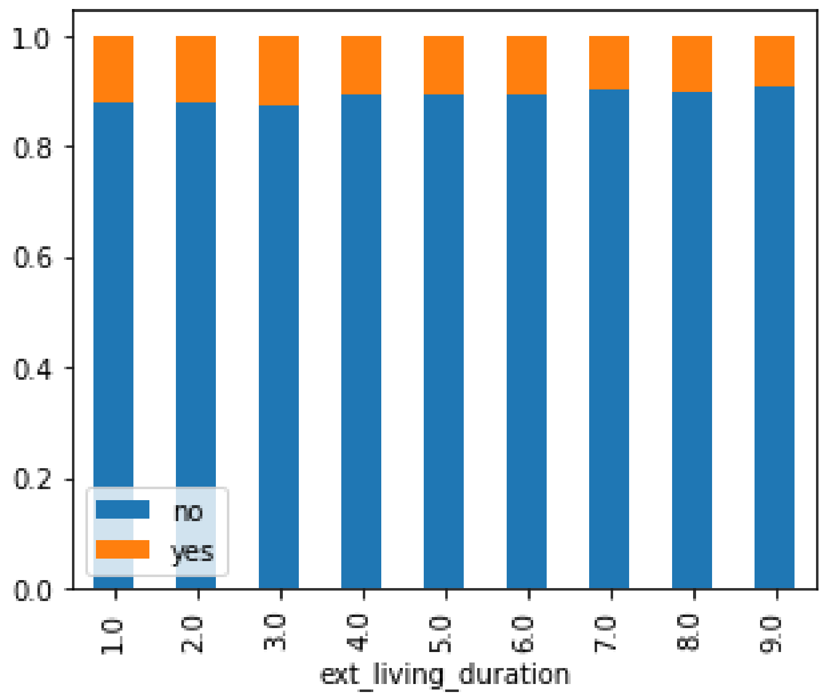
```
#Analysing ext_car_power and cross_buy
ax=pd.crosstab(checking_dt.ext_car_power,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                     no       yes
ext_car_power
1.0            0.899390  0.100610
2.0            0.899689  0.100311
3.0            0.903935  0.096065
4.0            0.900024  0.099976
5.0            0.901492  0.098508
```

[71]:

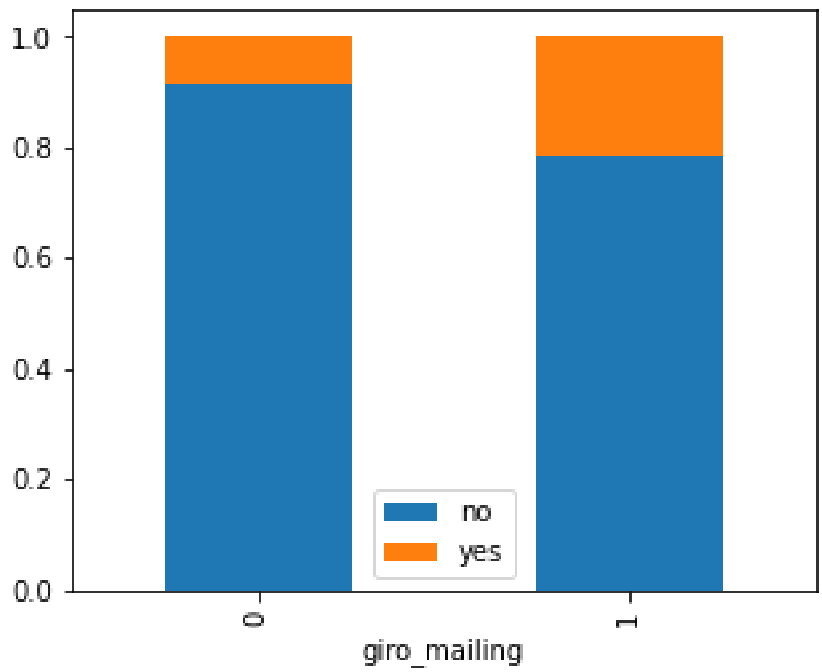
```
#Analysing ext_living_duration and cross_buy
ax=pd.crosstab(checking_dt.ext_living_duration,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                          no       yes
ext_living_duration
1.0                 0.878951  0.121049
2.0                 0.879317  0.120683
3.0                 0.874239  0.125761
4.0                 0.891229  0.108771
5.0                 0.892054  0.107946
6.0                 0.893150  0.106850
7.0                 0.902222  0.097778
8.0                 0.899168  0.100832
9.0                 0.910186  0.089814
```

[72]:

```
#Analysing giro_mailing and cross_buy
ax=pd.crosstab(checking_dt.giro_mailing,
               checking_dt.cross_buy,
               normalize='index')
ax.columns=['no','yes']
ax.plot(kind='bar',stacked=True,figsize=(5,4))

plt.show()
print(ax)
```




```
                    no       yes
giro_mailing
0             0.913445  0.086555
1             0.785497  0.214503
```

Checking skewness

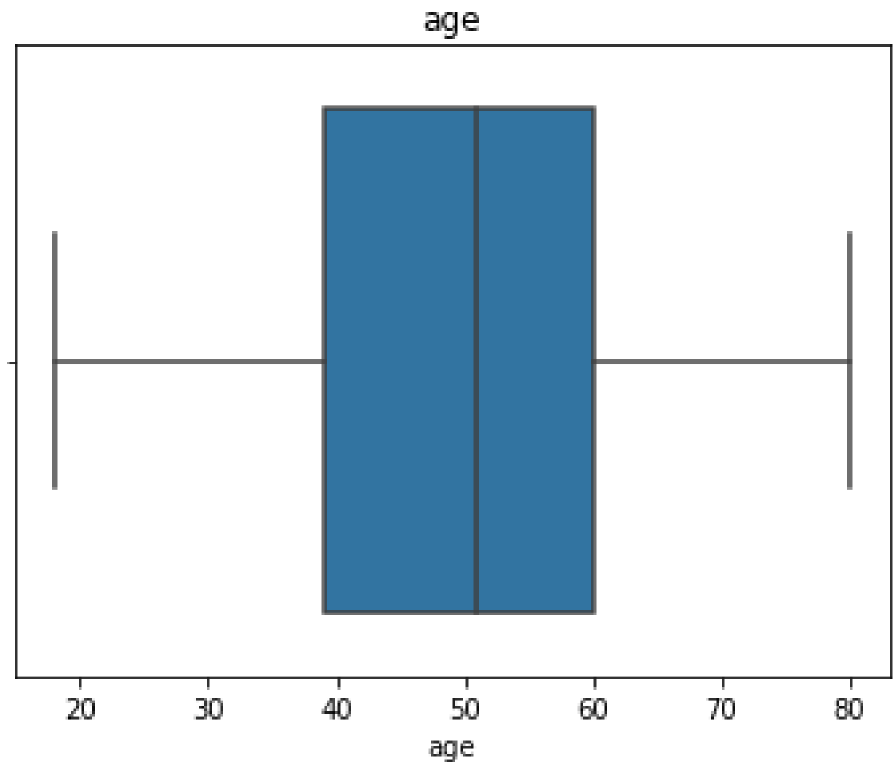
[73]:
```
checking_dt.skew()
```

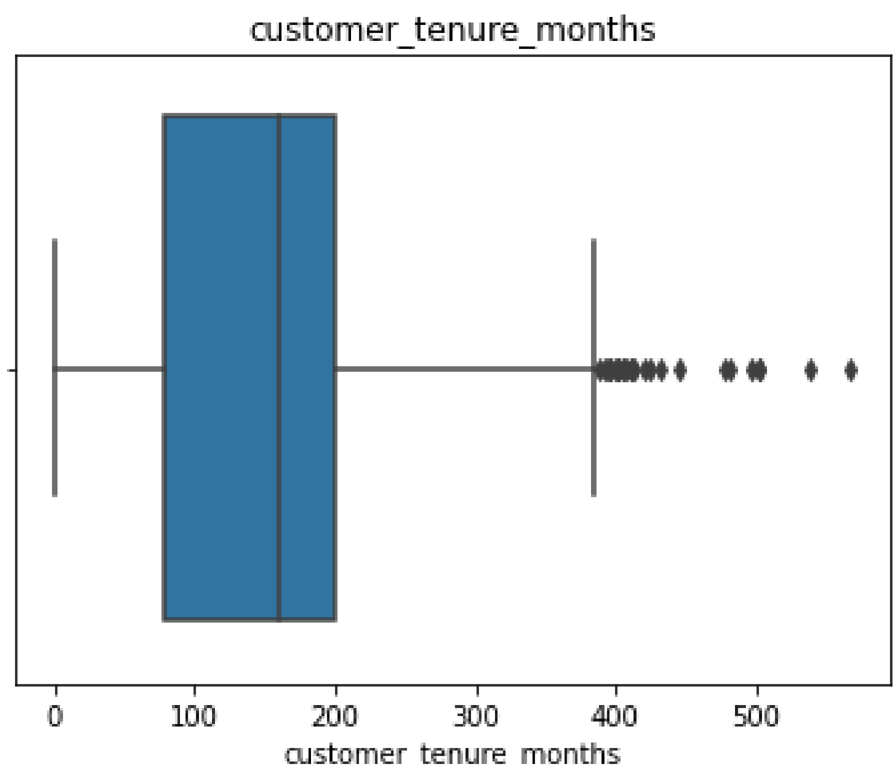
[73]:
```
cross_buy                     2.666707
acad_title                    6.787182
age                          -0.061094
calls                        21.114449
complaints                   38.029591
customer_tenure_months       -0.198338
directmails                   2.709763
gender                        0.384571
joint_account                 1.239077
inflows                      27.071333
last_acc_opening_days         0.187719
logins_desktop               25.147134
logins_mobile                14.612906
member_get_member_active     18.424952
member_get_member_passive    15.939916
nr_products                   2.952820
outflows                    -48.451305
```

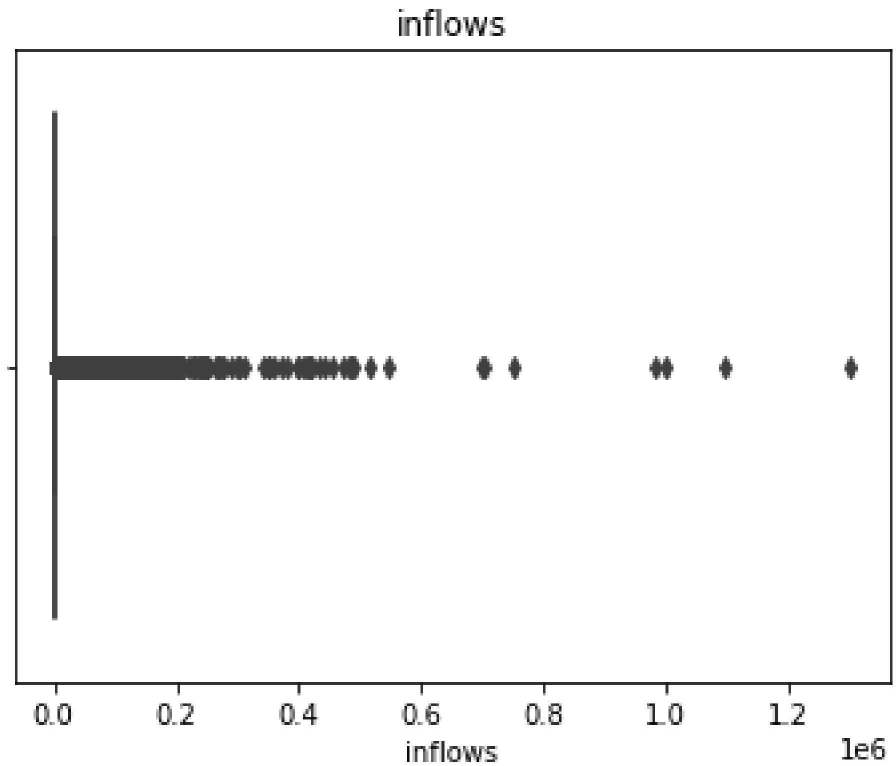
```
prod_loan                    2.747877
prod_mortgages               4.765646
prod_brokerage               2.723816
prod_pensionplan            42.633866
prod_savings                 1.302325
relocations                  5.179200
volume_debit               196.292726
volume_debit_6months        34.933565
ext_city_size                0.234372
ext_house_size               1.282385
ext_purchase_power          -0.385338
ext_share_new_houses         0.227788
ext_share_new_cars          -0.176475
ext_car_power               -0.037989
ext_living_duration         -1.089300
giro_mailing                 2.575689
dtype: float64
```

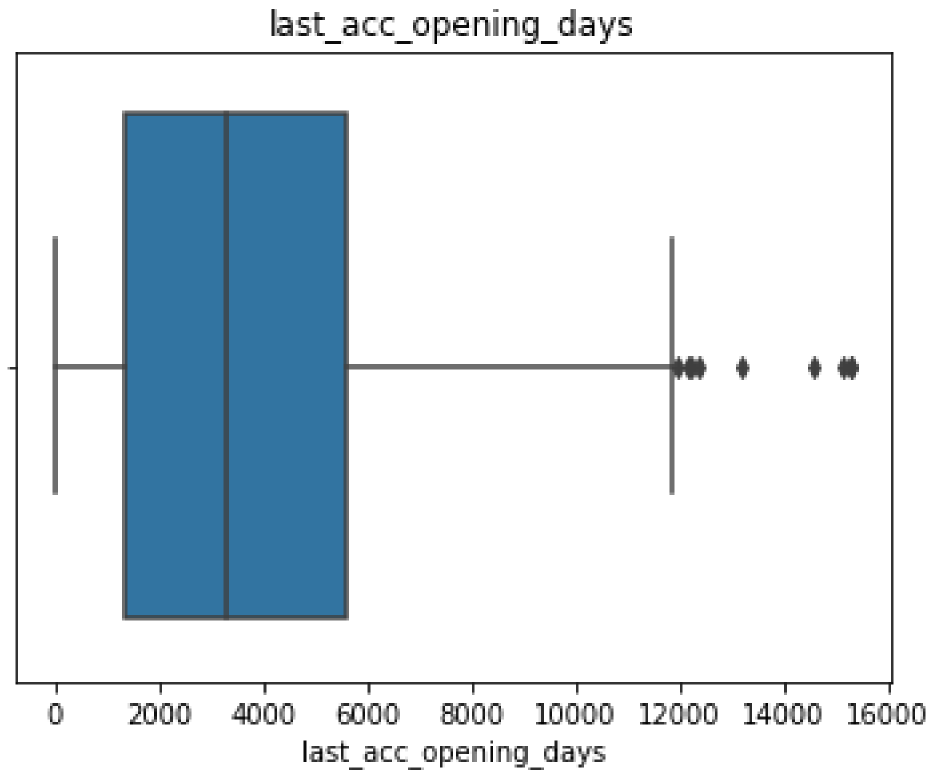
Visualizing Outliers

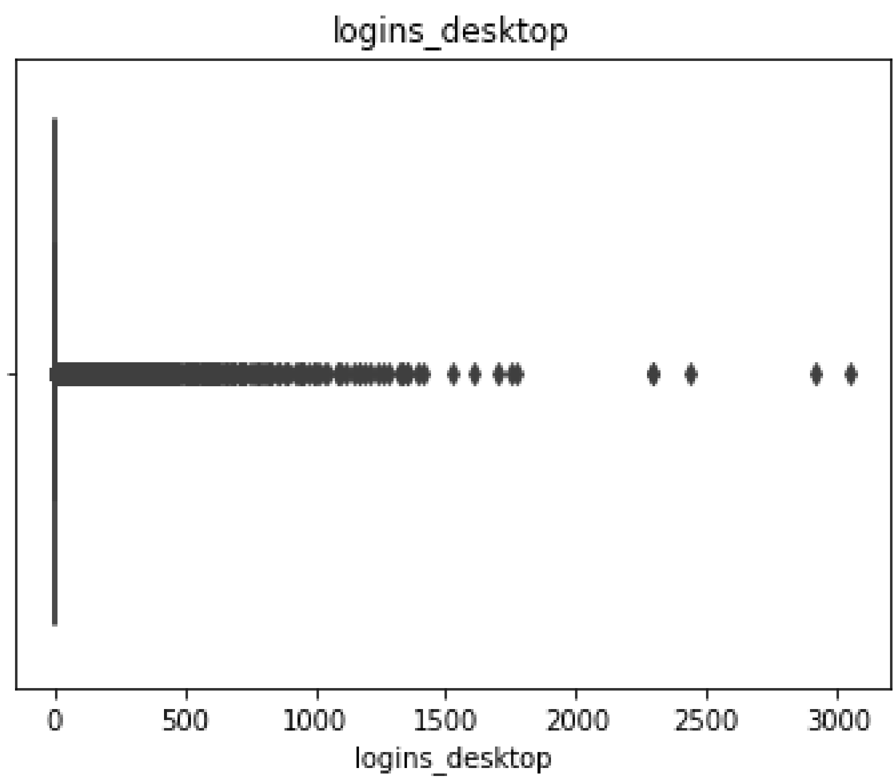
[74]:
```
# Visualizing each variables outlier
col=checking_dt[['age','customer_tenure_months','inflows',
                 'last_acc_opening_days','logins_desktop',
                 'outflows','volume_debit','volume_debit_6months']]
for i in col:
  n=1
  plt.figure(figsize=(20,20))
  plt.subplot(4,3,1)
  sns.boxplot(checking_dt[i])
  plt.title(i)
  plt.show()
  n=n+1
```

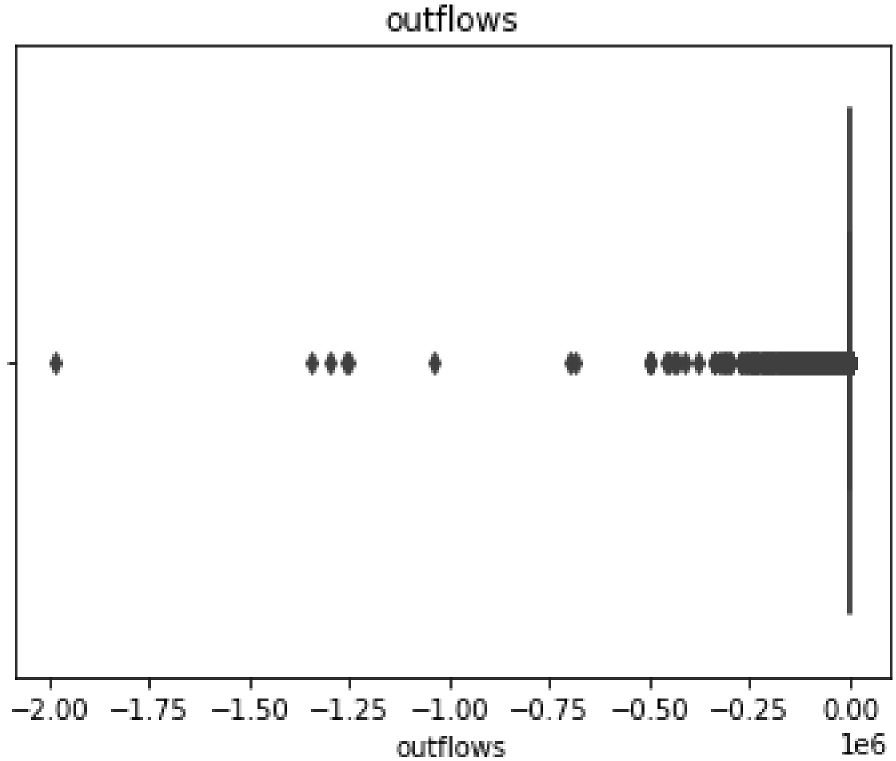

age
20
30
40
50
60
70
80
age

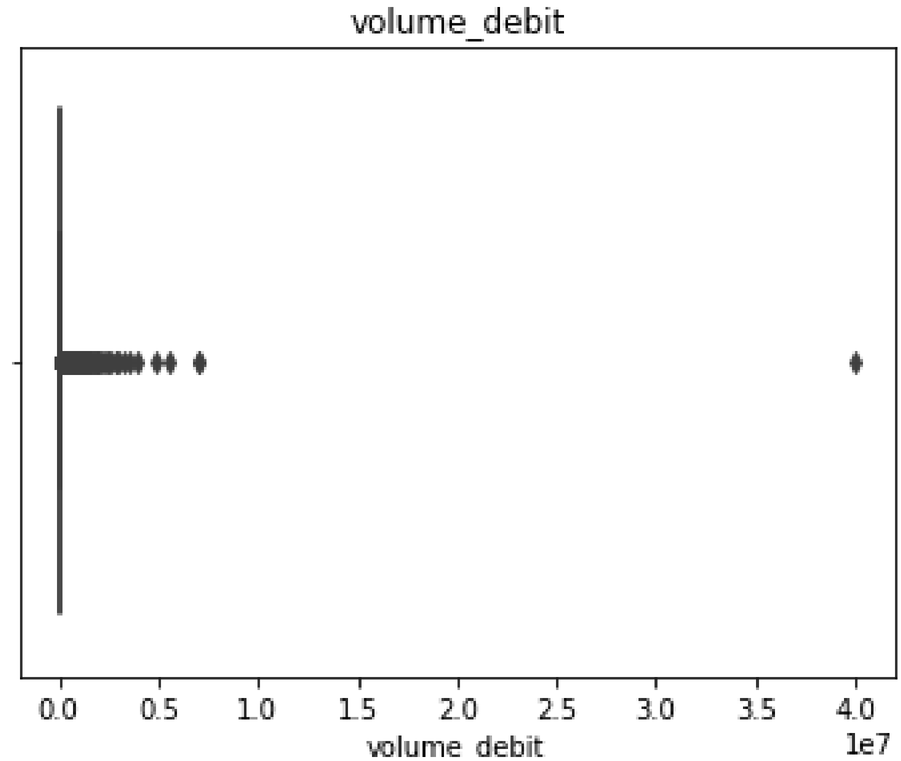

customer_tenure_months
0
100
200
300
400
500
customer_tenure_months

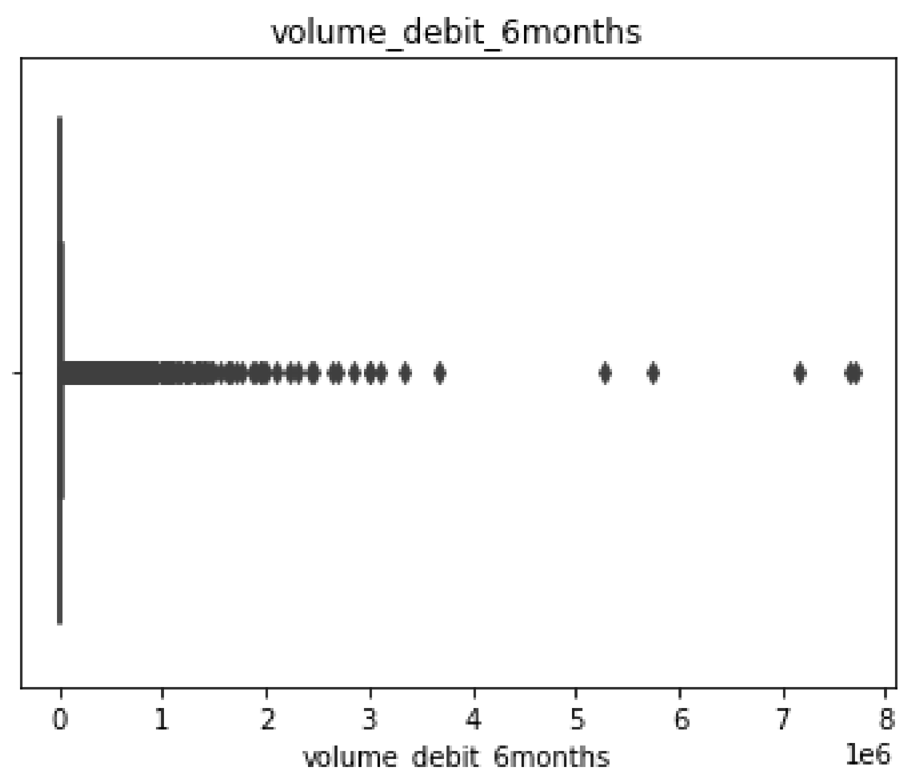

inflows
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e6
inflows


last_acc_opening_days
0
2000
4000
6000
8000
10000
12000
14000
16000
last_acc_opening_days


logins_desktop
0
500
1000
1500
2000
2500
3000
logins_desktop


outflows
−2.00
−1.75
−1.50
−1.25
−1.00
−0.75
−0.50
−0.25
0.00
1e6
outflows


volume_debit
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
volume_debit
1e7


volume_debit_6months
0
1
2
3
4
5
6
7
8
volume_debit_6months
1e6

Data Preprocessing

Dropping unrelated columns

[75]:
```
# Dropp "occupation" column
# it has more than 50% of missing values
checking_dt.drop(['occupation'], axis=1, inplace = True)
```

Removing Outliers

[75]:
```

```

Handling missing values

[76]:
```
# imputation
from sklearn.impute import SimpleImputer
mean_imputer = SimpleImputer(strategy='mean')
mode_imputer = SimpleImputer(strategy='most_frequent')

#imputattion with mean
checking_dt[['inflows','outflows','volume_debit_6months']] = pd.
↪DataFrame(mean_imputer.fit_transform(checking_dt[['inflows',
                                                                    ␣
↪                                   'outflows',
                                                                    ␣
↪                                   'volume_debit_6months']]))
```

[77]:
```
#imputation with most frequent
checking_dt[['gender',
            'joint_account',
            'ext_city_size',
            'ext_house_size',
            'ext_house_size',
            'ext_purchase_power',
            'ext_share_new_houses',
            'ext_share_new_cars',
            'ext_car_power',
            'ext_living_duration']] = pd.DataFrame(mode_imputer.
↪fit_transform(checking_dt[['gender',
                                                                    ␣
↪           'joint_account',
                                                                    ␣
↪           'ext_city_size',
                                                                    ␣
↪           'ext_house_size',
```

```
                                                                              ␣
↪          'ext_house_size',
                                                                              ␣
↪          'ext_purchase_power',
                                                                              ␣
↪          'ext_share_new_houses',
                                                                              ␣
↪          'ext_share_new_cars',
                                                                              ␣
↪          'ext_car_power',
                                                                              ␣
↪          'ext_living_duration']]))
```

[78]:
```
checking_dt.isna().sum()
```

[78]:
```
cross_buy                    0
acad_title                   0
age                          0
calls                        0
complaints                   0
customer_tenure_months       0
directmails                  0
gender                       0
joint_account                0
inflows                      0
last_acc_opening_days        0
logins_desktop               0
logins_mobile                0
marital_status               0
member_get_member_active     0
member_get_member_passive    0
nr_products                  0
outflows                     0
prod_loan                    0
prod_mortgages               0
prod_brokerage               0
prod_pensionplan             0
prod_savings                 0
relocations                  0
volume_debit                 0
volume_debit_6months         0
ext_city_size                0
ext_house_size               0
ext_purchase_power           0
ext_share_new_houses         0
ext_share_new_cars           0
ext_car_power                0
```

```
ext_living_duration          0
giro_mailing                 0
dtype: int64
```

[79]:
```
# creating dummy variables
checking_dt  = pd.get_dummies(checking_dt,
                           columns = ['complaints',
                                      'directmails',
                                      'marital_status',
                                      'nr_products',
                                      'prod_loan',
                                      'prod_pensionplan',
                                      'prod_savings',
                                      'relocations',
                                      'prod_brokerage'],
                            drop_first = True)
```

[80]:
```
checking_dt.head()
```

[80]:
```
   cross_buy  acad_title  age  calls  customer_tenure_months  gender  \
0          0           0   60      0                     221     0.0
1          0           0   55      0                     227     0.0
2          0           0   61      0                     221     1.0
3          0           0   70      0                     222     0.0
4          0           1   61      0                     227     1.0

   joint_account  inflows  last_acc_opening_days  logins_desktop  …  \
0            0.0      0.0                   1786               0  …
1            1.0      0.0                   6911               0  …
2            0.0   3000.0                   6728               0  …
3            0.0   6000.0                   6762               0  …
4            0.0      0.0                   6302              30  …

   relocations_2  relocations_3  prod_brokerage_1  prod_brokerage_2  \
0              0              0                 0                 0
1              0              0                 0                 0
2              0              0                 0                 0
3              0              0                 0                 0
4              0              0                 1                 0

   prod_brokerage_3  prod_brokerage_4  prod_brokerage_5  prod_brokerage_7  \
0                 0                 0                 0                 0
1                 0                 0                 0                 0
2                 0                 0                 0                 0
3                 0                 0                 0                 0
4                 0                 0                 0                 0
```

```
   prod_brokerage_8  prod_brokerage_10
0                 0                  0
1                 0                  0
2                 0                  0
3                 0                  0
4                 0                  0

[5 rows x 93 columns]
```

[81]:
```
# creating ordinal variables
from sklearn.preprocessing import OrdinalEncoder
enc = OrdinalEncoder()
enc.fit(checking_dt[["ext_city_size",
            "ext_house_size",
            "ext_purchase_power",
            "ext_share_new_houses",
            "ext_share_new_cars",
            "ext_car_power",
            "ext_living_duration",]])
checking_dt[["ext_city_size",
    "ext_house_size",
    "ext_purchase_power",
    "ext_share_new_houses",
    "ext_share_new_cars",
    "ext_car_power",
    "ext_living_duration",]] = enc.transform(checking_dt[["ext_city_size",
                                                          "ext_house_size",
                                                          "ext_purchase_power",
                                                         ␣
 ↪"ext_share_new_houses",
                                                          "ext_share_new_cars",
                                                          "ext_car_power",
                                                         ␣
 ↪"ext_living_duration"]])
```

[82]:
```
checking_dt.shape
```

[82]: (100000, 93)

Dimentionality reduction

[82]:
```

```

Data Spliting

[83]:
```
# Subseting the predictors and response variables
X = checking_dt.drop(['cross_buy'],axis = 1)
y = checking_dt['cross_buy']
```

```
#NearMiss under_sampling
from imblearn.under_sampling import NearMiss
nm = NearMiss()
X_res, y_res = nm.fit_resample(X, y)


X_train, X_test, y_train, y_test = train_test_split(X_res, y_res,
                                                    test_size=0.4,
                                                    random_state=1)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(12000, 92)
(8000, 92)
(12000,)
(8000,)
```

[84]:
```
# MinMaxScaler
from sklearn.preprocessing import MinMaxScaler
norm = MinMaxScaler().fit(X_train)
train_X = norm.transform(X_train)
test_X = norm.transform(X_test)
```

Logistic Regression

[85]:
```
model = LogisticRegression(class_weight = 'balanced')
grid={"C":np.logspace(-3,3,7),
      "penalty":["l1","l2"]}# l1 lasso l2 ridge
logreg_cv=GridSearchCV(model,grid,cv=10)
logreg_cv.fit(X_train, y_train)
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
log_pred =  logreg_cv.predict(X_test)

#Evaluation
print(classification_report(y_test,log_pred))
print("accuracy: ", metrics.accuracy_score(y_test, log_pred))
print("Recall:", recall_score(y_test,log_pred))

#Classification Error(Misclassification Rate)
print("Error_rate: ", 1 - metrics.accuracy_score(y_test, log_pred))

# Plot the cumulative gains chart of the expected spending
gains_df = pd.DataFrame({'actual': y_test,
```

```
                              'prob': log_pred})
gains_df = gains_df.sort_values(by=['prob'], ascending=False).
 ↪reset_index(drop=True)
gainsChart(gains_df.actual)
plt.show()
```

```
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'penalty': 'l2'}
accuracy : 0.8952499999999999
              precision    recall  f1-score   support

           0       0.84      0.98      0.91      4031
           1       0.97      0.82      0.89      3969

    accuracy                           0.90      8000
   macro avg       0.91      0.90      0.90      8000
weighted avg       0.91      0.90      0.90      8000

accuracy:  0.89675
Recall: 0.8150667674477198
Error_rate:  0.10324999999999995
```