In [4]:
```
import pandas as pd
import numpy as np
from pandas_profiling import ProfileReport
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import metrics
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [5]:
```
df=pd.read_csv(r'C:/Users/97150/Desktop/study/TA/PHY_TRAIN.csv')
```

In [6]:
```
df.head()
```

Out[6]:

	exampleid	target	feat1	feat2	feat3	feat4	feat5	feat6	feat7	feat8	...
0	1	0	0.000000	0.000000	0.000000	0	0.000000	0.0	0.000000	0	...
1	2	0	0.920167	0.817883	-0.646473	-1	0.000000	0.0	0.000000	0	...
2	3	1	0.868397	0.178202	0.150828	-1	0.000000	0.0	0.000000	0	...
3	4	0	0.000000	0.000000	0.000000	0	1.577894	0.0	-0.369792	-1	...
4	5	0	0.000000	0.000000	0.000000	0	0.000000	0.0	0.000000	0	...

5 rows × 80 columns

In [7]:
```
profile = ProfileReport(df)
```

```
C:\Users\97150\Anaconda3\lib\site-packages\pandas_profiling\describe.py:392:
FutureWarning: The join_axes-keyword is deprecated. Use .reindex or .reindex_
like on the result to achieve the same functionality.
  variable_stats = pd.concat(ldesc, join_axes=pd.Index([names]), axis=1)
```

In [8]:
```
profile
```

Out[8]:

Overview

Dataset info

Number of variables	80
Number of observations	50000
Total Missing (%)	4.5%
Total size in memory	30.5 MiB
Average record size in memory	640.0 B

Variables types

Numeric	56
Categorical	0
Boolean	7
Date	0
Text (Unique)	0
Rejected	17
Unsupported	0

Warnings

- feat1 has 42281 / 84.6% zeros Zeros
- feat2 has 43594 / 87.2% zeros Zeros
- feat3 has 42281 / 84.6% zeros Zeros
- feat4 has 42281 / 84.6% zeros Zeros
- feat5 has 44813 / 89.6% zeros Zeros
- feat6 has 45922 / 91.8% zeros Zeros
- feat7 has 44813 / 89.6% zeros Zeros
- feat8 has 44813 / 89.6% zeros Zeros
- feat9 has 638 / 1.3% zeros Zeros
- feat10 has 4669 / 9.3% zeros Zeros
- feat11 has 638 / 1.3% zeros Zeros
- feat12 has 638 / 1.3% zeros Zeros
- feat14 has 877 / 1.8% zeros Zeros
- feat15 has 5865 / 11.7% zeros Zeros
- feat16 has 24209 / 48.4% zeros Zeros
- feat17 has 42270 / 84.5% zeros Zeros
- feat18 is highly correlated with feat17 (ρ = 0.95692) Rejected
- feat20 has 34202 / 68.4% missing values Missing
- feat21 has 34202 / 68.4% missing values Missing
- feat22 has 34202 / 68.4% missing values Missing
- feat24 has 877 / 1.8% zeros Zeros
- feat25 has 5864 / 11.7% zeros Zeros
- feat26 has 19938 / 39.9% zeros Zeros
- feat27 has 19938 / 39.9% zeros Zeros
- feat28 has 19938 / 39.9% zeros Zeros
- feat29 has 30062 / 60.1% missing values Missing

- feat31 has 19938 / 39.9% zeros Zeros
- feat32 has 28050 / 56.1% zeros Zeros
- feat33 has 20807 / 41.6% zeros Zeros
- feat34 is highly correlated with feat27 (ρ = 0.90777) Rejected
- feat37 has 48823 / 97.6% zeros Zeros
- feat38 has 48823 / 97.6% zeros Zeros
- feat39 has 48823 / 97.6% zeros Zeros
- feat40 has 48823 / 97.6% zeros Zeros
- feat41 is highly skewed (γ1 = 27.959) Skewed
- feat41 has 48823 / 97.6% zeros Zeros
- feat42 is highly skewed (γ1 = 24.131) Skewed
- feat42 has 48823 / 97.6% zeros Zeros
- feat44 has 14469 / 28.9% missing values Missing
- feat44 has 19938 / 39.9% zeros Zeros
- feat45 is highly correlated with feat43 (ρ = 0.96715) Rejected
- feat46 has 14469 / 28.9% missing values Missing
- feat46 has 19938 / 39.9% zeros Zeros
- feat47 has constant value 0 Rejected
- feat48 has constant value 0 Rejected
- feat49 has constant value 0 Rejected
- feat50 has constant value 0 Rejected
- feat51 has constant value 0 Rejected
- feat52 is highly correlated with feat45 (ρ = 0.96715) Rejected
- feat53 is highly correlated with feat52 (ρ = 0.92105) Rejected
- feat54 has 31398 / 62.8% zeros Zeros
- feat55 has 18602 / 37.2% missing values Missing
- feat56 has 31398 / 62.8% zeros Zeros
- feat57 has 36532 / 73.1% zeros Zeros
- feat58 has 31982 / 64.0% zeros Zeros
- feat59 is highly correlated with feat53 (ρ = 0.93239) Rejected
- feat60 is highly correlated with feat44 (ρ = 0.99377) Rejected
- feat61 is highly correlated with feat60 (ρ = 0.90097) Rejected
- feat62 is highly correlated with feat46 (ρ = 0.97318) Rejected
- feat63 has 35946 / 71.9% zeros Zeros
- feat64 has 4744 / 9.5% zeros Zeros
- feat65 has 4744 / 9.5% zeros Zeros
- feat66 has 4744 / 9.5% zeros Zeros
- feat67 has 4744 / 9.5% zeros Zeros
- feat68 has 20406 / 40.8% zeros Zeros
- feat69 has 20406 / 40.8% zeros Zeros
- feat70 has 20406 / 40.8% zeros Zeros
- feat71 has 20406 / 40.8% zeros Zeros
- feat72 has 45621 / 91.2% zeros Zeros
- feat73 is highly correlated with feat2 (ρ = 0.91618) Rejected
- feat74 has 45621 / 91.2% zeros Zeros
- feat75 has 45621 / 91.2% zeros Zeros

- feat76 is highly correlated with feat72 (ρ = 0.90826) Rejected
- feat77 has 45621 / 91.2% zeros Zeros
- feat78 is highly correlated with feat76 (ρ = 0.92963) Rejected

Variables

exampleid

Numeric

Distinct count	50000
Unique (%)	100.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	25000
Minimum	1
Maximum	50000
Zeros (%)	0.0%

Toggle details

target

Boolean

Distinct count	2
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Mean	0.49722

0	25139
1	24861

Toggle details

feat1

Numeric

Distinct count	7706
Unique (%)	15.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0

Mean	0.15561
Minimum	0
Maximum	2.639
Zeros (%)	84.6%

Toggle details

feat2

Numeric

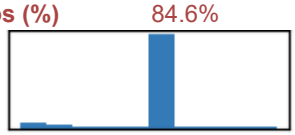
Distinct count	6397
Unique (%)	12.8%
Missing (%)	0.0%
Missing (n)	0
Infinfinite (%)	0.0%
Infinite (n)	0
Mean	0.084876
Minimum	0
Maximum	3.4296
Zeros (%)	87.2%

Toggle details

feat3

Numeric

Distinct count	7704
Unique (%)	15.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	-0.050354
Minimum	-1
Maximum	0.99995
Zeros (%)	84.6%



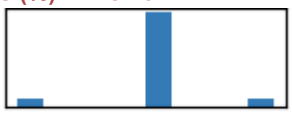
Toggle details

feat4

Numeric

Distinct count	3

Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	-6e-05
Minimum	-1
Maximum	1
Zeros (%)	84.6%



Toggle details

feat5

Numeric

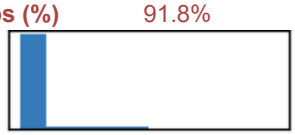
Distinct count	5175
Unique (%)	10.3%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.12657
Minimum	0
Maximum	2.719
Zeros (%)	89.6%

Toggle details

feat6

Numeric

Distinct count	4072
Unique (%)	8.1%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.049887
Minimum	0
Maximum	3.0546
Zeros (%)	91.8%

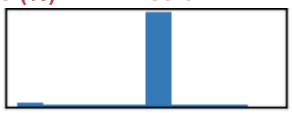


Toggle details

feat7

Numeric

Distinct count	5178
Unique (%)	10.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	-0.038344
Minimum	-1
Maximum	0.99927
Zeros (%)	89.6%

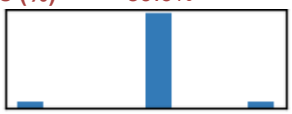


Toggle details

feat8

Numeric

Distinct count	3
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.00286
Minimum	-1
Maximum	1
Zeros (%)	89.6%



Toggle details

feat9

Numeric

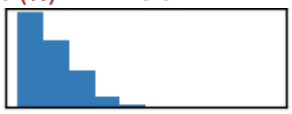
Distinct count	49086
Unique (%)	98.2%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.84835
Minimum	0
Maximum	6.6998
Zeros (%)	1.3%

Toggle details

feat10

Numeric

Distinct count	45130
Unique (%)	90.3%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.67349
Minimum	0
Maximum	5.2837
Zeros (%)	9.3%

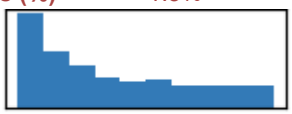


Toggle details

feat11

Numeric

Distinct count	49193
Unique (%)	98.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	-0.28392
Minimum	-1
Maximum	0.99991
Zeros (%)	1.3%



Toggle details

feat12

Numeric

Distinct count	3
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%

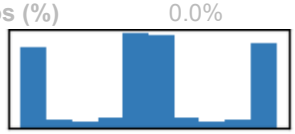
Infinite (n)	0
Mean	-0.01092
Minimum	-1
Maximum	1
Zeros (%)	1.3%

Toggle details

feat13

Numeric

Distinct count	48753
Unique (%)	97.5%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.0072603
Minimum	-1
Maximum	1
Zeros (%)	0.0%



Toggle details

feat14

Numeric

Distinct count	48807
Unique (%)	97.6%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.00085638
Minimum	-1
Maximum	0.99999
Zeros (%)	1.8%

Toggle details

feat15

Numeric

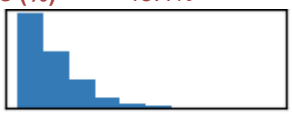
Distinct count	43605
Unique (%)	87.2%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	-0.00082113
Minimum	-0.99987
Maximum	0.9998
Zeros (%)	11.7%

Toggle details

feat16

Numeric

Distinct count	10
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.85558
Minimum	0
Maximum	9
Zeros (%)	48.4%

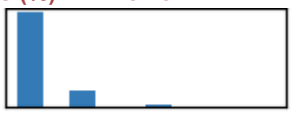


Toggle details

feat17

Numeric

Distinct count	5
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.16806
Minimum	0
Maximum	4
Zeros (%)	84.5%



Toggle details

~~feat18~~

Highly correlated

This variable is highly correlated with `feat17` *and should be ignored for analysis*

Correlation	0.95692

feat19

Numeric

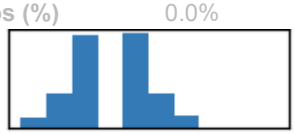
Distinct count	49422
Unique (%)	98.8%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	1.1212
Minimum	0
Maximum	6.0773
Zeros (%)	0.0%

Toggle details

feat20

Numeric

Distinct count	15761
Unique (%)	31.5%
Missing (%)	68.4%
Missing (n)	34202
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.0011184
Minimum	-2.4537
Maximum	4.5072
Zeros (%)	0.0%



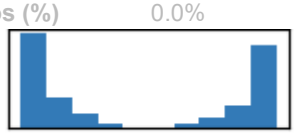
Toggle details

feat21

Numeric

Distinct count	15753
Unique (%)	31.5%
Missing (%)	68.4%
Missing (n)	34202

Infinite (%)	0.0%
Infinite (n)	0
Mean	-0.06716
Minimum	-1
Maximum	1
Zeros (%)	0.0%

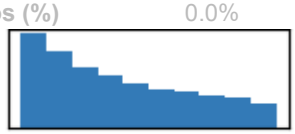


Toggle details

feat22

Numeric

Distinct count	15753
Unique (%)	31.5%
Missing (%)	68.4%
Missing (n)	34202
Infinite (%)	0.0%
Infinite (n)	0
Mean	-0.6199
Minimum	-0.99997
Maximum	-2.8348e-05
Zeros (%)	0.0%

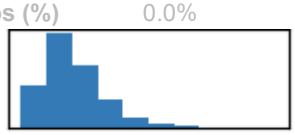


Toggle details

feat23

Numeric

Distinct count	49682
Unique (%)	99.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.92841
Minimum	0
Maximum	4.5083
Zeros (%)	0.0%

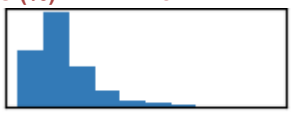


Toggle details

feat24

Numeric

Distinct count	48885
Unique (%)	97.8%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.82601
Minimum	0
Maximum	4.79
Zeros (%)	1.8%

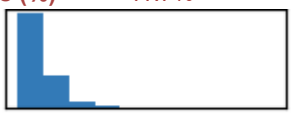


Toggle details

feat25

Numeric

Distinct count	43946
Unique (%)	87.9%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.52308
Minimum	0
Maximum	6.4519
Zeros (%)	11.7%

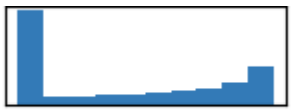


Toggle details

feat26

Numeric

Distinct count	29937
Unique (%)	59.9%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.40489
Minimum	0
Maximum	1
Zeros (%)	39.9%

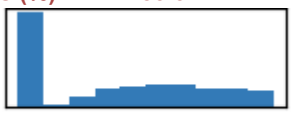


Toggle details

feat27

Numeric

Distinct count	29839
Unique (%)	59.7%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.091401
Minimum	0
Maximum	0.24999
Zeros (%)	39.9%

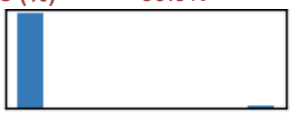


Toggle details

feat28

Numeric

Distinct count	29440
Unique (%)	58.9%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.014918
Minimum	0
Maximum	1
Zeros (%)	39.9%



Toggle details

feat29

Boolean

Distinct count	2
Unique (%)	0.0%
Missing (%)	60.1%
Missing (n)	30062
Mean	0

0.0	19938	
(Missing)		30062

Toggle details

feat30

Boolean

Distinct count	2
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Mean	0.40062

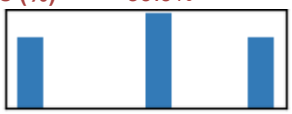
0		29969
1	20031	

Toggle details

feat31

Numeric

Distinct count	3
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.00168
Minimum	-1
Maximum	1
Zeros (%)	39.9%



Toggle details

feat32

Numeric

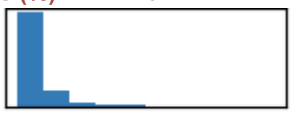
Distinct count	471
Unique (%)	0.9%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	361.68
Minimum	0
Maximum	10000
Zeros (%)	56.1%

Toggle details

feat33

Numeric

Distinct count	360
Unique (%)	0.7%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	4.3702
Minimum	0
Maximum	75
Zeros (%)	41.6%



Toggle details

~~feat34~~

Highly correlated

This variable is highly correlated with `feat27` *and should be ignored for analysis*

Correlation	0.90777

feat35

Boolean

Distinct count	2
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Mean	0.02354

0	48823
1	1177

Toggle details

feat36

Boolean

Distinct count	2
Unique (%)	0.0%

Missing (%)	0.0%
Missing (n)	0
Mean	0.02306

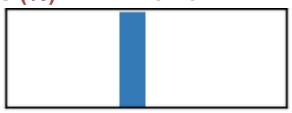
0	48847
1	1153

Toggle details

feat37

Numeric

Distinct count	1176
Unique (%)	2.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.0027108
Minimum	-6.4702
Maximum	7.5655
Zeros (%)	97.6%

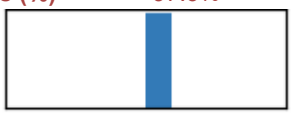


Toggle details

feat38

Numeric

Distinct count	1176
Unique (%)	2.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.0045754
Minimum	-10.772
Maximum	9.4373
Zeros (%)	97.6%



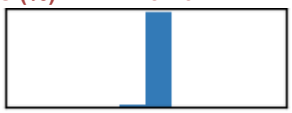
Toggle details

feat39

Numeric

Distinct count	1176
Unique (%)	2.4%

Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	-0.004823
Minimum	-17.322
Maximum	14.763
Zeros (%)	97.6%



Toggle details

feat40

Numeric

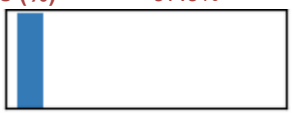
Distinct count	1176
Unique (%)	2.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.0031947
Minimum	0
Maximum	1.9385
Zeros (%)	97.6%

Toggle details

feat41

Numeric

Distinct count	1175
Unique (%)	2.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.0036662
Minimum	0
Maximum	3.5483
Zeros (%)	97.6%

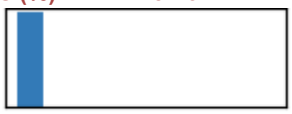


Toggle details

feat42

Numeric

Distinct count	1176
Unique (%)	2.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.0087853
Minimum	0
Maximum	6.1925
Zeros (%)	97.6%



Toggle details

feat43

Boolean

Distinct count	2
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Mean	0.60124

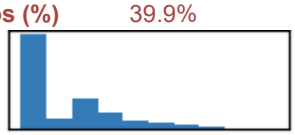
1	30062
0	19938

Toggle details

feat44

Numeric

Distinct count	15542
Unique (%)	31.1%
Missing (%)	28.9%
Missing (n)	14469
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.42765
Minimum	0
Maximum	2.8788
Zeros (%)	39.9%



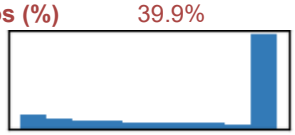
Toggle details

~~feat45~~
Highly correlated

This variable is highly correlated with `feat43` *and should be ignored for analysis*
Correlation 0.96715

feat46
Numeric

Distinct count	15556
Unique (%)	31.1%
Missing (%)	28.9%
Missing (n)	14469
Infinite (%)	0.0%
Infinite (n)	0
Mean	-0.26815
Minimum	-0.99999
Maximum	0
Zeros (%)	39.9%



Toggle details

~~feat47~~
Constant

This variable is constant and should be ignored for analysis
Constant value 0

~~feat48~~
Constant

This variable is constant and should be ignored for analysis
Constant value 0

~~feat49~~
Constant

This variable is constant and should be ignored for analysis
Constant value 0

~~feat50~~
Constant

This variable is constant and should be ignored for analysis
Constant value 0

~~feat51~~

Constant

This variable is constant and should be ignored for analysis

Constant value 0

~~feat52~~

Highly correlated

This variable is highly correlated with `feat45` *and should be ignored for analysis*

Correlation 0.96715

~~feat53~~

Highly correlated

This variable is highly correlated with `feat52` *and should be ignored for analysis*

Correlation 0.92105

feat54

Numeric

Distinct count	18361
Unique (%)	36.7%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.012835
Minimum	0
Maximum	1
Zeros (%)	62.8%

Toggle details

feat55

Boolean

Distinct count	2
Unique (%)	0.0%
Missing (%)	37.2%
Missing (n)	18602
Mean	0

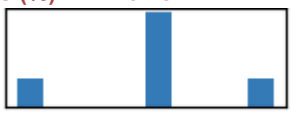
0.0	31398
(Missing)	18602

Toggle details

feat56

Numeric

Distinct count	3
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.00256
Minimum	-1
Maximum	1
Zeros (%)	62.8%



Toggle details

feat57

Numeric

Distinct count	420
Unique (%)	0.8%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	218.78
Minimum	0
Maximum	10000
Zeros (%)	73.1%

Toggle details

feat58

Numeric

Distinct count	337
Unique (%)	0.7%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	2.6589
Minimum	0
Maximum	75

Zeros (%)	64.0%

Toggle details

~~feat59~~

Highly correlated

This variable is highly correlated with `feat53` *and should be ignored for analysis*

Correlation	0.93239

~~feat60~~

Highly correlated

This variable is highly correlated with `feat44` *and should be ignored for analysis*

Correlation	0.99377

~~feat61~~

Highly correlated

This variable is highly correlated with `feat60` *and should be ignored for analysis*

Correlation	0.90097

~~feat62~~

Highly correlated

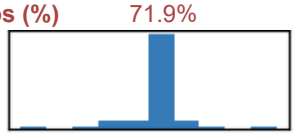
This variable is highly correlated with `feat46` *and should be ignored for analysis*

Correlation	0.97318

feat63

Numeric

Distinct count	27
Unique (%)	0.1%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.00234
Minimum	-22
Maximum	22
Zeros (%)	71.9%

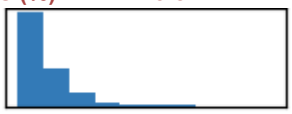


Toggle details

feat64

Numeric

Distinct count	45079
Unique (%)	90.2%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.010682
Minimum	0
Maximum	0.099991
Zeros (%)	9.5%



Toggle details

feat65

Numeric

Distinct count	44916
Unique (%)	89.8%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.96576
Minimum	0
Maximum	5.6262
Zeros (%)	9.5%

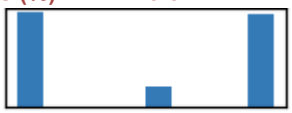
Toggle details

feat66

Numeric

Distinct count	3
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	-0.00364
Minimum	-1
Maximum	1

Zeros (%)	9.5%

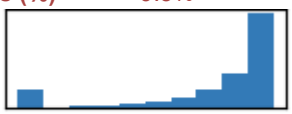


Toggle details

feat67

Numeric

Distinct count	44691
Unique (%)	89.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.78429
Minimum	0
Maximum	1
Zeros (%)	9.5%



Toggle details

feat68

Numeric

Distinct count	29523
Unique (%)	59.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.1599
Minimum	-0.99999
Maximum	1
Zeros (%)	40.8%

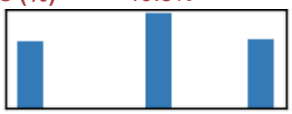
Toggle details

feat69

Numeric

Distinct count	3
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0

Infinite (%)	0.0%
Infinite (n)	0
Mean	0.00812
Minimum	-1
Maximum	1
Zeros (%)	40.8%



Toggle details

feat70

Numeric

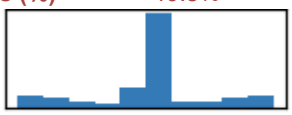
Distinct count	29276
Unique (%)	58.6%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.00047824
Minimum	-1
Maximum	1
Zeros (%)	40.8%

Toggle details

feat71

Numeric

Distinct count	29529
Unique (%)	59.1%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.0031982
Minimum	-0.908
Maximum	0.90774
Zeros (%)	40.8%



Toggle details

feat72

Numeric

Distinct count	4373
Unique (%)	8.7%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.052807
Minimum	0
Maximum	0.99995
Zeros (%)	91.2%

Toggle details

~~feat73~~

Highly correlated

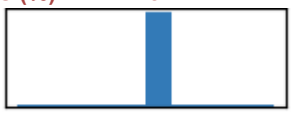
This variable is highly correlated with `feat2` *and should be ignored for analysis*

Correlation	0.91618

feat74

Numeric

Distinct count	4372
Unique (%)	8.7%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	-0.014101
Minimum	-1
Maximum	0.99987
Zeros (%)	91.2%



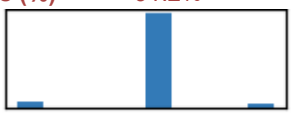
Toggle details

feat75

Numeric

Distinct count	3
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0

Mean	-0.00146
Minimum	-1
Maximum	1
Zeros (%)	91.2%



Toggle details

~~feat76~~

Highly correlated

This variable is highly correlated with `feat72` *and should be ignored for analysis*

Correlation	0.90826

feat77

Numeric

Distinct count	4374
Unique (%)	8.7%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.0028432
Minimum	0
Maximum	0.38551
Zeros (%)	91.2%

Toggle details

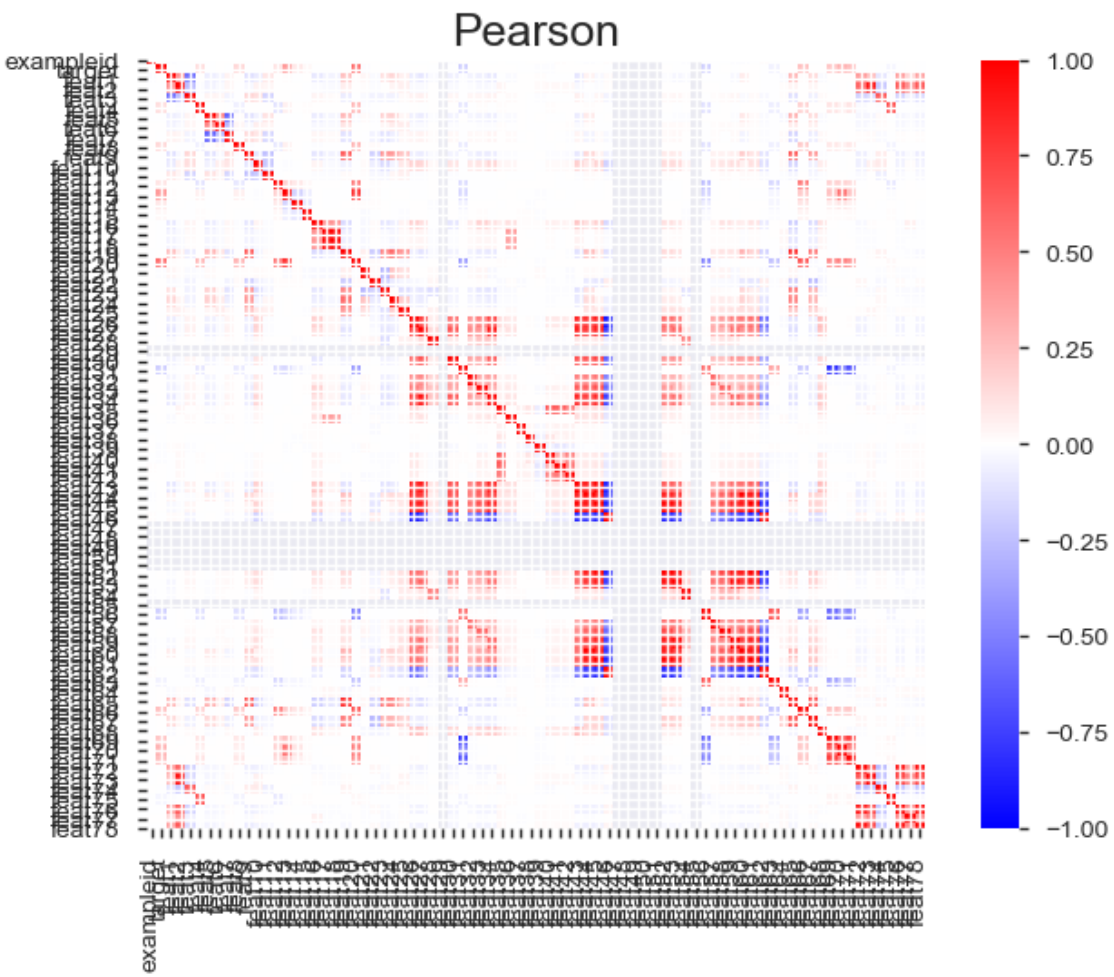
~~feat78~~

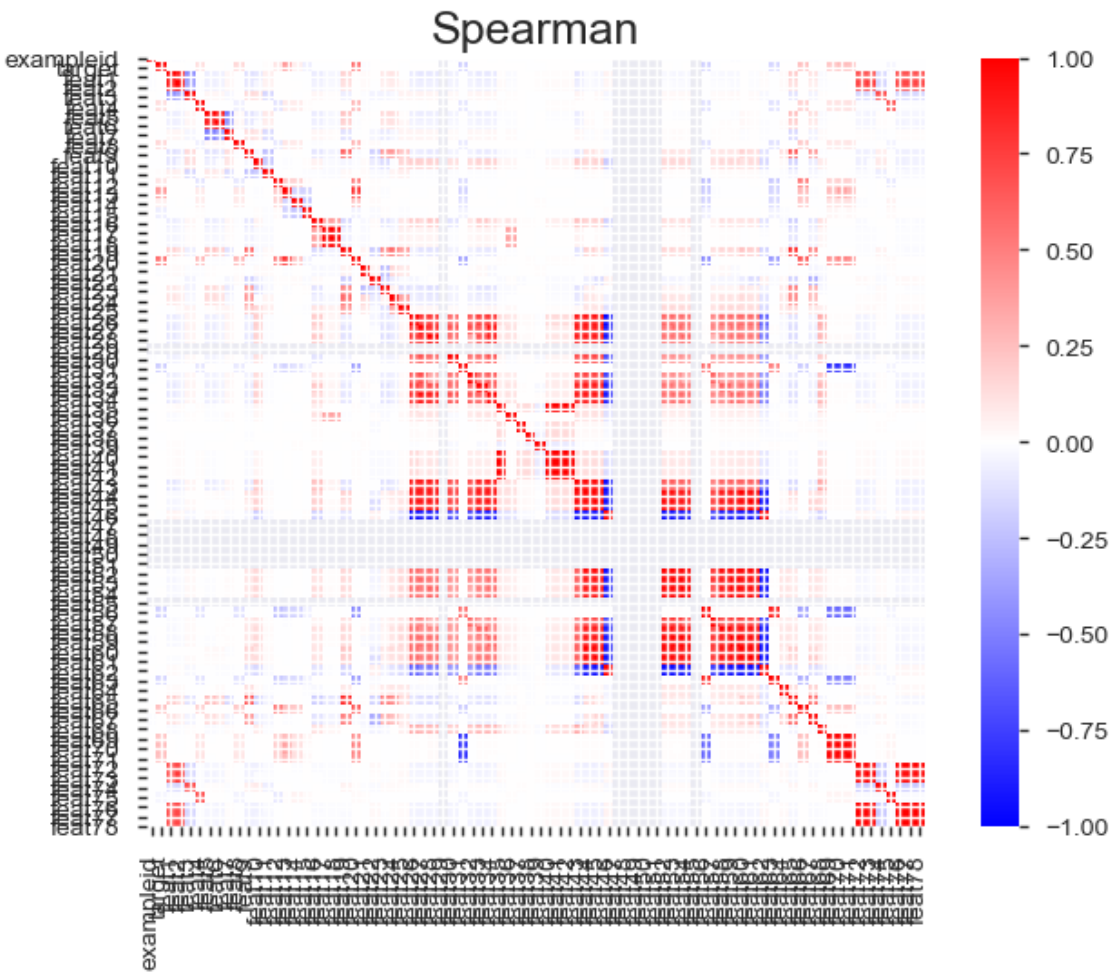
Highly correlated

This variable is highly correlated with `feat76` *and should be ignored for analysis*

Correlation	0.92963

Correlations


Pearson
exampleid
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00



Sample

	exampleid	target	feat1	feat2	feat3	feat4	feat5	feat6	feat7
0	1	0	0.000000	0.000000	0.000000	0	0.000000	0.0	0.000000
1	2	0	0.920167	0.817883	-0.646473	-1	0.000000	0.0	0.000000
2	3	1	0.868397	0.178202	0.150828	-1	0.000000	0.0	0.000000
3	4	0	0.000000	0.000000	0.000000	0	1.577894	0.0	-0.369792
4	5	0	0.000000	0.000000	0.000000	0	0.000000	0.0	0.000000

In [9]:
```
df1=df.copy()
col_list=df.columns
empty_col_list=[]
for i in col_list:
    if(df1[i].isnull().any()):
        empty_col_list.append(i)
empty_col_list
```

Out[9]:
```
['feat20',
 'feat21',
 'feat22',
 'feat29',
 'feat44',
 'feat45',
 'feat46',
 'feat55']
```

Missing value imputation

In [10]:
```
import numpy as np
from sklearn.impute import SimpleImputer
imp_mean = SimpleImputer(missing_values=np.nan, strategy='mean')
imp_mean.fit(df1)
df1_imp=pd.DataFrame(imp_mean.transform(df1))
df1_imp.columns=df1.columns
df1_imp
```

Out[10]:

	exampleid	target	feat1	feat2	feat3	feat4	feat5	feat6	feat7	feat8
0	1.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0
1	2.0	0.0	0.920167	0.817883	-0.646473	-1.0	0.000000	0.0	0.000000	0.0
2	3.0	1.0	0.868397	0.178202	0.150828	-1.0	0.000000	0.0	0.000000	0.0
3	4.0	0.0	0.000000	0.000000	0.000000	0.0	1.577894	0.0	-0.369792	-1.0
4	5.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0
...	...	...	...	...	...	...	...	...	...	...
49995	49996.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0
49996	49997.0	1.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0
49997	49998.0	1.0	0.918590	1.012605	-0.047045	-1.0	0.000000	0.0	0.000000	0.0
49998	49999.0	1.0	0.000000	0.000000	0.000000	0.0	0.855551	0.0	-0.849437	1.0
49999	50000.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0

50000 rows × 80 columns

Checking for Multicollinearity

Checking Pearson Correlation

In [11]:
```
from scipy.stats import pearsonr
```

In [12]:
```
pearson_corr={}
k=[]
for i in df1_imp.columns:
    if(i not in k):
        for j in df1_imp.columns:
            p=pearsonr(df1_imp[i],df1_imp[j])[0]
            if(i!=j):

                count=1
                if(p>0.9):
                    lst=[]
                    lst.append(p)
                    lst.append(j)
                    pearson_corr[i]=lst
                    #k.append(j)
```

```
C:\Users\97150\Anaconda3\lib\site-packages\scipy\stats\stats.py:3508: Pearson
RConstantInputWarning: An input array is constant; the correlation coefficent
is not defined.
  warnings.warn(PearsonRConstantInputWarning())
```

In [13]:
```
pearson_corr
```

Out[13]:
```
{'feat2': [0.9161779473031394, 'feat73'],
 'feat17': [0.9569214241113242, 'feat18'],
 'feat18': [0.9569214241113242, 'feat17'],
 'feat27': [0.9077698112619299, 'feat34'],
 'feat34': [0.9077698112619299, 'feat27'],
 'feat52': [0.9702970235248272, 'feat61'],
 'feat53': [0.9323928277108464, 'feat59'],
 'feat59': [0.9323928277108464, 'feat53'],
 'feat60': [0.9009721775401007, 'feat61'],
 'feat61': [0.9009721775401007, 'feat60'],
 'feat72': [0.9344205329332373, 'feat78'],
 'feat73': [0.9161779473031394, 'feat2'],
 'feat76': [0.9296293359816603, 'feat78'],
 'feat78': [0.9296293359816603, 'feat76']}
```

Dropping columns wich are highly correlated

In [14]:
```
df1_corr=df1_imp.drop(columns=['feat73','feat18','feat34','feat43','feat52','f
eat53','feat59','feat60','feat61','feat62','feat73','feat76','feat78'])
```

In [15]:
```
df1_corr
```

Out[15]:

	exampleid	target	feat1	feat2	feat3	feat4	feat5	feat6	feat7	feat8
0	1.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0
1	2.0	0.0	0.920167	0.817883	-0.646473	-1.0	0.000000	0.0	0.000000	0.0
2	3.0	1.0	0.868397	0.178202	0.150828	-1.0	0.000000	0.0	0.000000	0.0
3	4.0	0.0	0.000000	0.000000	0.000000	0.0	1.577894	0.0	-0.369792	-1.0
4	5.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0
...	...	...	...	...	...	...	...	...	...	...
49995	49996.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0
49996	49997.0	1.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0
49997	49998.0	1.0	0.918590	1.012605	-0.047045	-1.0	0.000000	0.0	0.000000	0.0
49998	49999.0	1.0	0.000000	0.000000	0.000000	0.0	0.855551	0.0	-0.849437	1.0
49999	50000.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0

50000 rows × 68 columns

Checking for Variation Inflation Factor among predictors and removing varibales which has VIF > 10

In [16]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [17]:
```
vif = pd.DataFrame([variance_inflation_factor(df1_corr.iloc[:,2:].values, i) f
or i in range(df1_corr.iloc[:,2:].shape[1])], index=df1_corr.iloc[:,2:].column
s, columns=['VIF_value'])
vif
```

```
C:\Users\97150\Anaconda3\lib\site-packages\statsmodels\regression\linear_mode
l.py:1638: RuntimeWarning: invalid value encountered in double_scalars
  return 1 - self.ssr/self.uncentered_tss
```

Out[17]:

	VIF_value
feat1	3.938347
feat2	2.069540
feat3	3.774698
feat4	2.209754
feat5	3.330959
...	...
feat71	2.460268
feat72	2.183206
feat74	2.629607
feat75	1.949437
feat77	1.318613

66 rows × 1 columns

In [18]:
```
vif_col=list(vif.loc[vif['VIF_value']>10].index)
```

In [19]:
```
vif_col
```

Out[19]:
```
['feat9', 'feat19', 'feat22', 'feat44', 'feat45', 'feat65', 'feat67']
```

In [20]:
```
vif_col=list(vif.loc[vif['VIF_value']>10].index)
vif_col
df1_vif=df1_corr.drop(columns=vif_col)
df1_vif
```

Out[20]:

	exampleid	target	feat1	feat2	feat3	feat4	feat5	feat6	feat7	feat8
0	1.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0
1	2.0	0.0	0.920167	0.817883	-0.646473	-1.0	0.000000	0.0	0.000000	0.0
2	3.0	1.0	0.868397	0.178202	0.150828	-1.0	0.000000	0.0	0.000000	0.0
3	4.0	0.0	0.000000	0.000000	0.000000	0.0	1.577894	0.0	-0.369792	-1.0
4	5.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0
...	...	...	...	...	...	...	...	...	...	...
49995	49996.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0
49996	49997.0	1.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0
49997	49998.0	1.0	0.918590	1.012605	-0.047045	-1.0	0.000000	0.0	0.000000	0.0
49998	49999.0	1.0	0.000000	0.000000	0.000000	0.0	0.855551	0.0	-0.849437	1.0
49999	50000.0	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0

50000 rows × 61 columns

Checking for constant zeros in entire variable and dropping the column

In [21]:
```
constant_columns=df1_vif.columns[df1_vif.nunique() <= 1]
df1_vif.nunique() <= 1
df1_vif=df1_vif.drop(columns=constant_columns)
```

Nomralizing data using standard scalar

In [22]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X=pd.DataFrame(sc.fit_transform(df1_vif.iloc[:,2:]))
Y=pd.DataFrame(df1_vif.target)
```

In [23]:
```
X.columns=df1_vif.iloc[:,2:].columns
```

In [24]:
```
X
```

Out[24]:

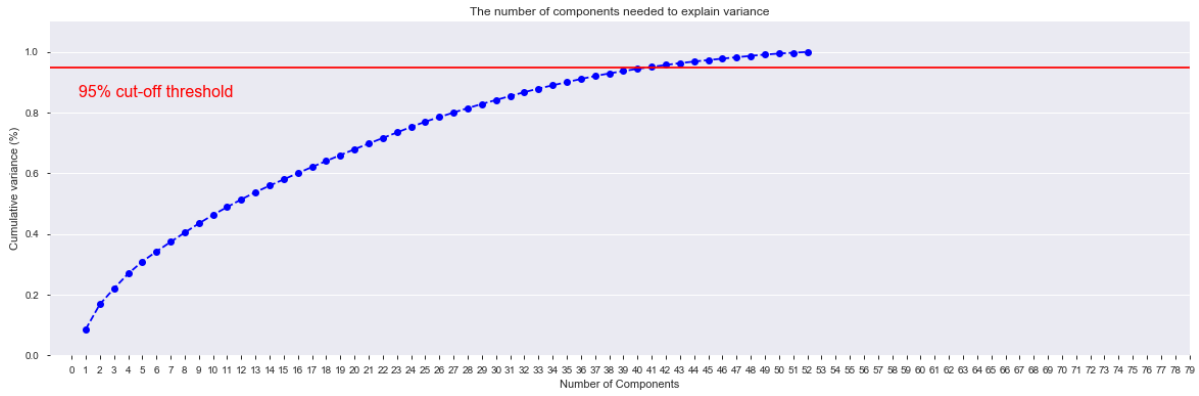
	feat1	feat2	feat3	feat4	feat5	feat6	feat7	feat8	
0	-0.375070	-0.287393	0.198443	0.000153	-0.315879	-0.222996	0.179040	-0.008880	-1.
1	1.842888	2.481980	-2.349283	-2.544945	-0.315879	-0.222996	0.179040	-0.008880	-0.
2	1.718103	0.316003	0.792852	-2.544945	-0.315879	-0.222996	0.179040	-0.008880	0.
3	-0.375070	-0.287393	0.198443	0.000153	3.622068	-0.222996	-1.547622	-3.113754	-0.
4	-0.375070	-0.287393	0.198443	0.000153	-0.315879	-0.222996	0.179040	-0.008880	0.
...	...	...	...	...	...	...	...	...	
49995	-0.375070	-0.287393	0.198443	0.000153	-0.315879	-0.222996	0.179040	-0.008880	-0.
49996	-0.375070	-0.287393	0.198443	0.000153	-0.315879	-0.222996	0.179040	-0.008880	-1.
49997	1.839087	3.141312	0.013042	-2.544945	-0.315879	-0.222996	0.179040	-0.008880	-1.
49998	-0.375070	-0.287393	0.198443	0.000153	1.819317	-0.222996	-3.787221	3.095994	-1.
49999	-0.375070	-0.287393	0.198443	0.000153	-0.315879	-0.222996	0.179040	-0.008880	-0.

50000 rows × 52 columns

PCA for feature Engineering

Determing number of components to choose in PCA

In [25]:
```
from sklearn.decomposition import PCA
pca = PCA().fit(X)

%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (20,6)

fig, ax = plt.subplots()
xi = np.arange(1, 53, step=1)
y = np.cumsum(pca.explained_variance_ratio_)

plt.ylim(0.0,1.1)
plt.plot(xi, y, marker='o', linestyle='--', color='b')

plt.xlabel('Number of Components')
plt.xticks(np.arange(0, 80, step=1)) #change from 0-based array index to 1-bas
ed human-readable label
plt.ylabel('Cumulative variance (%)')
plt.title('The number of components needed to explain variance')

plt.axhline(y=0.95, color='r', linestyle='-')
plt.text(0.5, 0.85, '95% cut-off threshold', color = 'red', fontsize=16)

ax.grid(axis='x')
plt.show()
```



assigning number of components = 43 as the curve starts to stabilize at this point

In [26]:
```
pca = PCA(n_components = 43)
X_PCA=pd.DataFrame(pca.fit_transform(X))

exp_variance=pca.explained_variance_ratio_
exp_variance
```

Out[26]:
```
array([0.08568541, 0.08246643, 0.05190548, 0.04893217, 0.04003197,
       0.03304703, 0.03157155, 0.03090242, 0.03009902, 0.02813723,
       0.02582478, 0.02530695, 0.02382285, 0.02140902, 0.02080392,
       0.02060629, 0.02020378, 0.01975249, 0.01947212, 0.01911592,
       0.01878339, 0.01840345, 0.01819927, 0.01813513, 0.01735437,
       0.0154188 , 0.01473917, 0.01458721, 0.01376855, 0.01309228,
       0.01302327, 0.01274674, 0.01181021, 0.01072778, 0.01060156,
       0.01024814, 0.00983895, 0.00853767, 0.00790185, 0.0071591 ,
       0.00655012, 0.00627445, 0.00609516])
```

Logistic Regression Without terms for PCA features

In [27]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(X_PCA,Y,test_size=0.3,random_state=
0)
```

In [28]:
```
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression(random_state=0).fit(xtrain, ytrain)
clf
```

```
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:4
32: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify
a solver to silence this warning.
  FutureWarning)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\utils\validation.py:724: D
ataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[28]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=0, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [29]:
```
clf.score(xtest,ytest)
```

Out[29]:
```
0.7042
```

Logistic Regression Without terms for features without PCA

In [30]:
```
from sklearn.model_selection import train_test_split
xtrain1,xtest1,ytrain1,ytest1=train_test_split(X,Y,test_size=0.3,random_state=
43)
```

In [31]:
```
import statsmodels.api as sm

model1 = sm.Logit(ytrain1,xtrain1)
result1 = model1.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.541056
         Iterations 6
```

In [32]:
```
result1.summary()
```

Out[32]:

Logit Regression Results

Dep. Variable:	target	**No. Observations:**	35000
Model:	Logit	**Df Residuals:**	34948
Method:	MLE	**Df Model:**	51
Date:	Wed, 22 Apr 2020	**Pseudo R-squ.:**	0.2194
Time:	16:31:43	**Log-Likelihood:**	-18937.
converged:	True	**LL-Null:**	-24259.
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
feat1	-0.0043	0.029	-0.148	0.882	-0.061	0.052
feat2	0.0061	0.018	0.337	0.736	-0.029	0.042
feat3	-0.0351	0.035	-1.012	0.312	-0.103	0.033
feat4	0.6844	0.028	24.323	0.000	0.629	0.740
feat5	-0.0026	0.020	-0.130	0.897	-0.042	0.037
feat6	-0.0114	0.015	-0.785	0.433	-0.040	0.017
feat7	-0.0175	0.017	-1.007	0.314	-0.052	0.017
feat8	0.2576	0.014	17.796	0.000	0.229	0.286
feat10	-0.0092	0.014	-0.676	0.499	-0.036	0.017
feat11	0.0187	0.013	1.423	0.155	-0.007	0.045
feat12	-0.0685	0.016	-4.289	0.000	-0.100	-0.037
feat13	0.9257	0.020	46.066	0.000	0.886	0.965
feat14	0.4538	0.016	29.121	0.000	0.423	0.484
feat15	0.1804	0.015	12.288	0.000	0.152	0.209
feat16	-0.0018	0.015	-0.118	0.906	-0.031	0.027
feat17	-0.0029	0.015	-0.200	0.841	-0.032	0.026
feat20	0.1966	0.019	10.447	0.000	0.160	0.233
feat21	-0.0194	0.015	-1.325	0.185	-0.048	0.009
feat23	-0.0023	0.014	-0.167	0.867	-0.029	0.025
feat24	0.0058	0.015	0.398	0.691	-0.023	0.034
feat25	-0.0063	0.015	-0.434	0.665	-0.035	0.022
feat26	0.0186	0.024	0.779	0.436	-0.028	0.065
feat27	-0.0149	0.022	-0.668	0.504	-0.058	0.029
feat28	0.0121	0.017	0.730	0.465	-0.020	0.044
feat30	-0.0041	0.018	-0.229	0.819	-0.039	0.031
feat31	-0.1480	0.022	-6.677	0.000	-0.191	-0.105
feat32	0.0249	0.017	1.426	0.154	-0.009	0.059

feat33	-0.0207	0.018	-1.121	0.262	-0.057	0.015
feat35	0.0247	0.022	1.142	0.254	-0.018	0.067
feat36	-0.0004	0.014	-0.027	0.979	-0.027	0.026
feat37	-0.0132	0.013	-1.035	0.300	-0.038	0.012
feat38	-0.0024	0.014	-0.175	0.861	-0.029	0.024
feat39	-0.0117	0.014	-0.841	0.401	-0.039	0.016
feat40	-0.0556	0.020	-2.847	0.004	-0.094	-0.017
feat41	-0.0085	0.020	-0.434	0.665	-0.047	0.030
feat42	0.0654	0.019	3.429	0.001	0.028	0.103
feat46	-0.0216	0.023	-0.957	0.339	-0.066	0.023
feat54	-0.0138	0.017	-0.797	0.425	-0.048	0.020
feat56	-0.0711	0.021	-3.439	0.001	-0.112	-0.031
feat57	-0.0332	0.018	-1.846	0.065	-0.068	0.002
feat58	0.0155	0.020	0.779	0.436	-0.024	0.055
feat63	0.1291	0.018	7.328	0.000	0.095	0.164
feat64	0.0040	0.012	0.330	0.742	-0.020	0.028
feat66	0.1724	0.017	10.408	0.000	0.140	0.205
feat68	-0.0048	0.014	-0.347	0.729	-0.032	0.022
feat69	-0.1640	0.027	-6.168	0.000	-0.216	-0.112
feat70	0.0017	0.021	0.081	0.936	-0.039	0.043
feat71	0.3204	0.020	15.995	0.000	0.281	0.360
feat72	-0.0131	0.020	-0.668	0.504	-0.051	0.025
feat74	0.0170	0.027	0.630	0.528	-0.036	0.070
feat75	-0.5952	0.024	-24.802	0.000	-0.642	-0.548
feat77	0.0200	0.015	1.365	0.172	-0.009	0.049

Rejecting variabes with less stastical significant(p> 0.05)

In [33]:
```
opt=pd.DataFrame(result1.pvalues[result1.pvalues<0.05])
```

In [34]:
```
opt
```

Out[34]:

	0
feat4	1.124544e-130
feat8	7.641749e-71
feat12	1.795996e-05
feat13	0.000000e+00
feat14	1.951651e-186
feat15	1.053943e-34
feat20	1.518693e-25
feat31	2.439942e-11
feat40	4.411882e-03
feat42	6.053734e-04
feat56	5.847514e-04
feat63	2.336659e-13
feat66	2.275366e-25
feat69	6.927581e-10
feat71	1.378890e-57
feat75	8.570518e-136

In [35]:
```
X_opt=X[['feat4','feat8','feat12','feat13','feat14','feat15','feat20','feat31'
,'feat40','feat42','feat56','feat63','feat66','feat69','feat71','feat75']]
X_opt
```

Out[35]:

	feat4	feat8	feat12	feat13	feat14	feat15	feat20	feat31
0	0.000153	-0.008880	-0.995511	0.234255	-0.124441	0.006868	1.504698e-17	1.287499
1	-2.544945	-0.008880	-0.995511	-1.550554	-0.037465	0.008326	-1.559263e+00	-1.291832
2	-2.544945	-0.008880	1.017494	1.387947	2.777807	-0.782209	1.307239e+00	-1.291832
3	0.000153	-3.113754	1.017494	0.000485	-0.018978	0.006622	1.504698e-17	-0.002167
4	0.000153	-0.008880	-0.995511	-0.011549	-0.002856	0.006928	1.504698e-17	-1.291832
...	...	...	...	...	...	...	...	...
49995	0.000153	-0.008880	-0.995511	-1.515229	0.011887	0.006959	1.504698e-17	-0.002167
49996	0.000153	-0.008880	-0.995511	0.026945	0.001851	0.001533	1.504698e-17	-0.002167
49997	-2.544945	-0.008880	1.017494	1.469257	0.129952	-0.076362	3.352785e+00	-0.002167
49998	0.000153	3.095994	-0.995511	1.528164	-0.000733	0.006698	1.504698e-17	-0.002167
49999	0.000153	-0.008880	-0.995511	-0.011537	-0.003087	0.007258	1.504698e-17	-0.002167

50000 rows × 16 columns

In [69]:
```
X_opt.describe()
```

Out[69]:

	feat4	feat8	feat12	feat13	feat14	feat15
count	5.000000e+04	5.000000e+04	5.000000e+04	5.000000e+04	5.000000e+04	5.000000e+04
mean	-7.760957e-16	5.623987e-16	-9.070744e-16	3.295617e-17	-9.416166e-19	-3.694116e-16
std	1.000010e+00	1.000010e+00	1.000010e+00	1.000010e+00	1.000010e+00	1.000010e+00
min	-2.544945e+00	-3.113754e+00	-9.955115e-01	-1.550663e+00	-3.298207e+00	-8.349277e+00
25%	1.527059e-04	-8.879940e-03	-9.955115e-01	-4.441958e-01	-4.656503e-03	6.609522e-03
50%	1.527059e-04	-8.879940e-03	1.099101e-02	-1.117477e-02	-2.822100e-03	6.862371e-03
75%	1.527059e-04	-8.879940e-03	1.017494e+00	5.359493e-01	-9.021375e-04	7.108908e-03
max	2.545251e+00	3.095994e+00	1.017494e+00	1.528309e+00	3.292549e+00	8.362368e+00

Applying Logistic regression

In [36]:
```
from sklearn.model_selection import train_test_split
xtrain_opt,xtest_opt,ytrain_opt,ytest_opt=train_test_split(X_opt,Y,test_size=
0.3,random_state=43)
```

In [37]:
```
from sklearn.linear_model import LogisticRegression
clf_log = LogisticRegression(random_state=0).fit(xtrain_opt, ytrain_opt)
clf_log
```

```
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:4
32: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify
a solver to silence this warning.
  FutureWarning)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\utils\validation.py:724: D
ataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[37]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=0, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

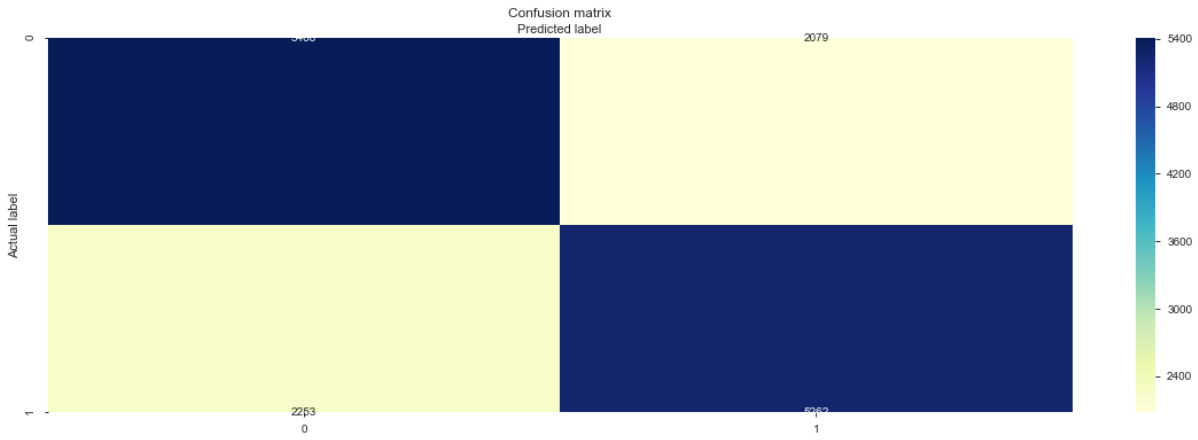
In [68]:
```
y_pred_opt=clf_log.predict(xtest_opt)
print(clf_log.score(xtest_opt,ytest_opt))
print(clf_log.intercept_)
```

```
0.7112
[-0.02451236]
```

In [39]:
```
cnf_matrix = metrics.confusion_matrix(ytest_opt, y_pred_opt)
print(cnf_matrix)
class_names=[0,1] # name  of classes
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
# create heatmap
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

```
[[5406 2079]
 [2253 5262]]
```

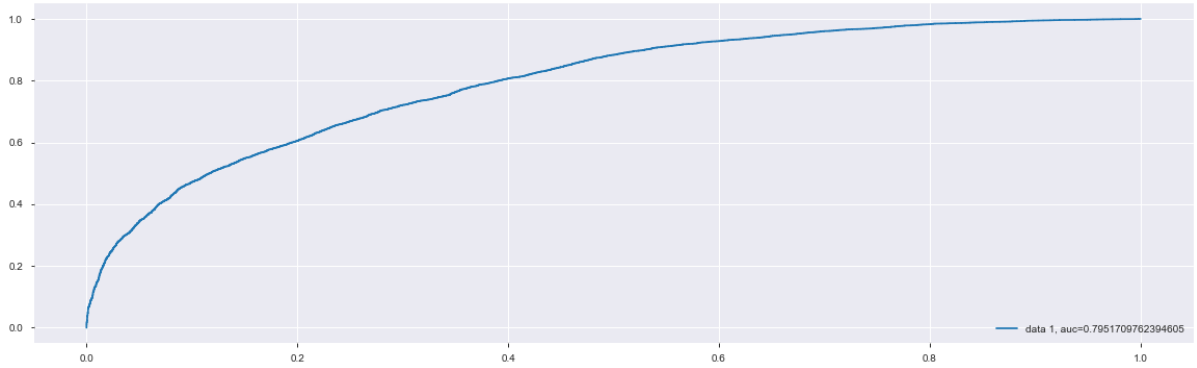
Out[39]: Text(0.5, 30.5, 'Predicted label')



In [40]:
```
print("Accuracy:",metrics.accuracy_score(ytest_opt, y_pred_opt))
print("Precision:",metrics.precision_score(ytest_opt, y_pred_opt))
print("Recall:",metrics.recall_score(ytest_opt, y_pred_opt))
```

```
Accuracy: 0.7112
Precision: 0.7167960768287699
Recall: 0.7001996007984032
```

In [41]:
```
y_pred_proba = clf_log.predict_proba(xtest_opt)[::,1]
fpr, tpr, _ = metrics.roc_curve(ytest_opt,  y_pred_proba)
auc = metrics.roc_auc_score(ytest_opt, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()
print("Auc:", auc)
```




```
Auc: 0.7951709762394605
```

Logistic Regression with Interaction terms

In [42]:
```
X_i=X
X_i['target']=Y
X_i
```

Out[42]:

	feat1	feat2	feat3	feat4	feat5	feat6	feat7	feat8	
0	-0.375070	-0.287393	0.198443	0.000153	-0.315879	-0.222996	0.179040	-0.008880	-1.1
1	1.842888	2.481980	-2.349283	-2.544945	-0.315879	-0.222996	0.179040	-0.008880	-0.3
2	1.718103	0.316003	0.792852	-2.544945	-0.315879	-0.222996	0.179040	-0.008880	0.1
3	-0.375070	-0.287393	0.198443	0.000153	3.622068	-0.222996	-1.547622	-3.113754	-0.4
4	-0.375070	-0.287393	0.198443	0.000153	-0.315879	-0.222996	0.179040	-0.008880	0.2
...	...	...	...	...	...	...	...	...	
49995	-0.375070	-0.287393	0.198443	0.000153	-0.315879	-0.222996	0.179040	-0.008880	-0.9
49996	-0.375070	-0.287393	0.198443	0.000153	-0.315879	-0.222996	0.179040	-0.008880	-1.0
49997	1.839087	3.141312	0.013042	-2.544945	-0.315879	-0.222996	0.179040	-0.008880	-1.3
49998	-0.375070	-0.287393	0.198443	0.000153	1.819317	-0.222996	-3.787221	3.095994	-1.3
49999	-0.375070	-0.287393	0.198443	0.000153	-0.315879	-0.222996	0.179040	-0.008880	-0.4

50000 rows × 53 columns

In [43]:
```
# Import libraries
import statsmodels.api as sm
from statsmodels.formula.api import glm

# Fit GLM and print model summary
model_int = glm('target ~ center(feat4) + center(feat12) + center(feat14) +cen
ter(feat15)+center(feat20)+center(feat31) +center(feat4):center(feat12)+center
(feat14):center(feat15)+center(feat20):center(feat31)',
                data =X_i , family = sm.families.Binomial()).fit()

# View model results
print(model_int.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
=
Dep. Variable:                  target   No. Observations:                5000
0
Model:                             GLM   Df Residuals:                    4999
0
Model Family:                Binomial   Df Model:
9
Link Function:                   logit   Scale:                          1.000
0
Method:                           IRLS   Log-Likelihood:                -3100
4.
Date:                Wed, 22 Apr 2020   Deviance:                       6200
8.
Time:                        16:31:45   Pearson chi2:                 4.93e+0
4
No. Iterations:                     5
Covariance Type:            nonrobust
==============================================================================
====================
                                 coef    std err          z      P>|z|
[0.025      0.975]
------------------------------------------------------------------------------
--------------------
Intercept                       -0.0107      0.010     -1.093      0.274
-0.030       0.008
center(feat4)                    0.1712      0.010     16.635      0.000
0.151       0.191
center(feat12)                   0.0040      0.010      0.397      0.692
-0.016       0.024
center(feat14)                   0.2002      0.010     19.707      0.000
0.180       0.220
center(feat15)                   0.0799      0.010      7.878      0.000
0.060       0.100
center(feat20)                   0.7500      0.014     55.341      0.000
0.723       0.777
center(feat31)                  -0.2618      0.010    -26.708      0.000
-0.281      -0.243
center(feat4):center(feat12)     0.0169      0.011      1.606      0.108
-0.004       0.038
center(feat14):center(feat15)   -0.0035      0.003     -1.090      0.276
-0.010       0.003
center(feat20):center(feat31)    0.0050      0.014      0.370      0.712
-0.022       0.032
==============================================================================
====================
```

In [44]:
```
pred_inttrms=(model_int.predict(xtest_opt) >= 0.5).astype(int)
```

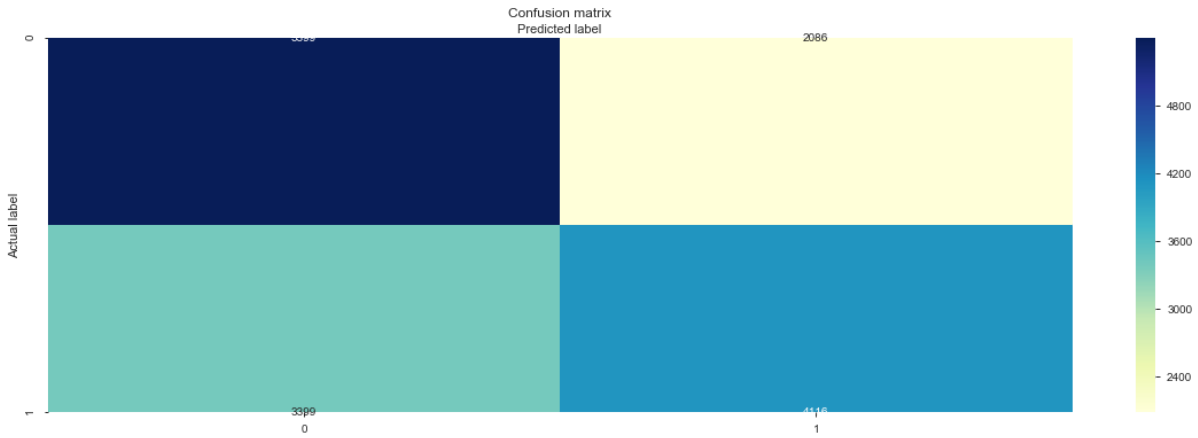
In [45]:
```
from sklearn.metrics import accuracy_score
score =accuracy_score(ytest_opt,pred_inttrms)
score
```

Out[45]: 0.6343333333333333

In [46]:
```
cnf_matrix = metrics.confusion_matrix(ytest_opt, pred_inttrms)
print(cnf_matrix)
class_names=[0,1] # name  of classes
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
# create heatmap
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

```
[[5399 2086]
 [3399 4116]]
```

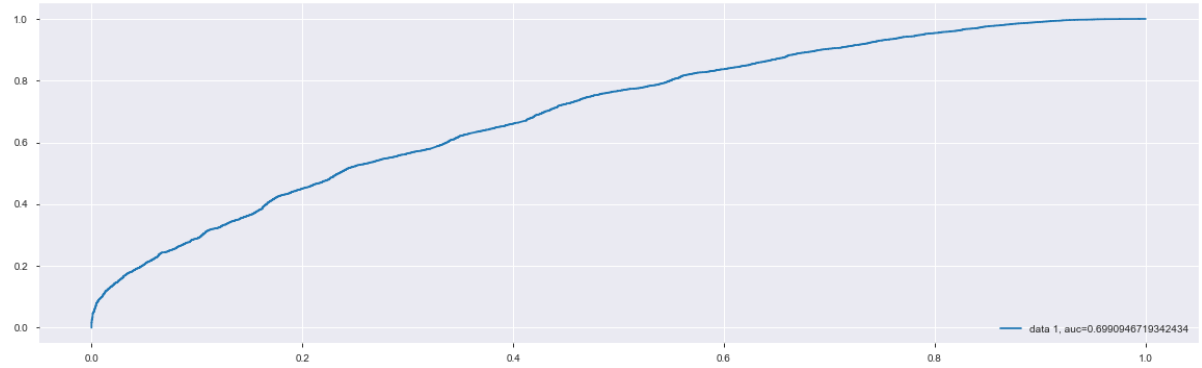
Out[46]: Text(0.5, 30.5, 'Predicted label')



In [47]:
```
print("Accuracy:",metrics.accuracy_score(ytest_opt, pred_inttrms))
print("Precision:",metrics.precision_score(ytest_opt, pred_inttrms))
print("Recall:",metrics.recall_score(ytest_opt, pred_inttrms))
```

```
Accuracy: 0.6343333333333333
Precision: 0.6636568848758465
Recall: 0.5477045908183633
```

In [48]:
```
y_pred_proba = model_int.predict(xtest_opt)
fpr, tpr, _ = metrics.roc_curve(ytest_opt, y_pred_proba  )
auc = metrics.roc_auc_score(ytest_opt, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()
print("Auc:", auc)
```




```
Auc: 0.6990946719342434
```

Random Forest

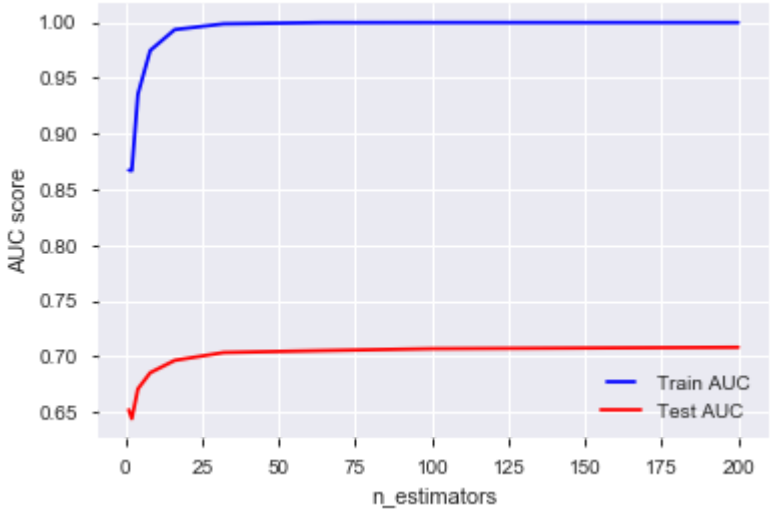
In [49]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
```

To find n_Estimators in Random Forest, we determine by plotting AUC on each estimators in loop

In [50]:
```
n_estimators = [1, 2, 4, 8, 16, 32, 64, 100, 200]
train_results = []
test_results = []
for estimator in n_estimators:
    rf = RandomForestClassifier(n_estimators=estimator, n_jobs=-1)
    rf.fit(xtrain_opt, ytrain_opt)
    train_pred = rf.predict(xtrain_opt)
    false_positive_rate, true_positive_rate, thresholds = roc_curve(ytrain_opt
, train_pred)
    roc_auc = auc(false_positive_rate, true_positive_rate)
    train_results.append(roc_auc)
    y_pred1 = rf.predict(xtest_opt)
    false_positive_rate, true_positive_rate, thresholds = roc_curve(ytest_opt,
y_pred1)
    roc_auc = auc(false_positive_rate, true_positive_rate)
    test_results.append(roc_auc)
```

```
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().

C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().

C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().

C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().

C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().

C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().

C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().

C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().

C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
```

In [51]:
```
%matplotlib inline
from matplotlib.legend_handler import HandlerLine2D
line1, = plt.plot(n_estimators, train_results, 'b', label="Train AUC")
line2, = plt.plot(n_estimators, test_results, 'r', label="Test AUC")
plt.legend(handler_map={line1: HandlerLine2D(numpoints=2)})
plt.ylabel('AUC score')
plt.xlabel('n_estimators')
plt.show()
```



Assiging range (50,75)

In [52]:
```
r_f={}
for i in range(50,75):
    rf = RandomForestClassifier(n_estimators=i, n_jobs=-1)
    rf.fit(xtrain_opt, ytrain_opt)
    y_pred1 = rf.predict(xtest_opt)
    r_f[i]= rf.score(xtest_opt,ytest_opt)
```

```
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
```

```
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
```

In [53]:
```python
max_value = max(r_f.values())  # maximum value
max_keys = [k for k, v in r_f.items() if v == max_value] # getting all keys co
ntaining the `maximum`

print(max_value, max_keys)
```

```
0.7092666666666667 [55]
```

Maximum Auc occurs when n_estimators = 67

In [54]:
```
rf = RandomForestClassifier(n_estimators=67, n_jobs=-1)
rf.fit(xtrain_opt, ytrain_opt)
y_pred1 = rf.predict(xtest_opt)
rf.score(xtest_opt,ytest_opt)
```

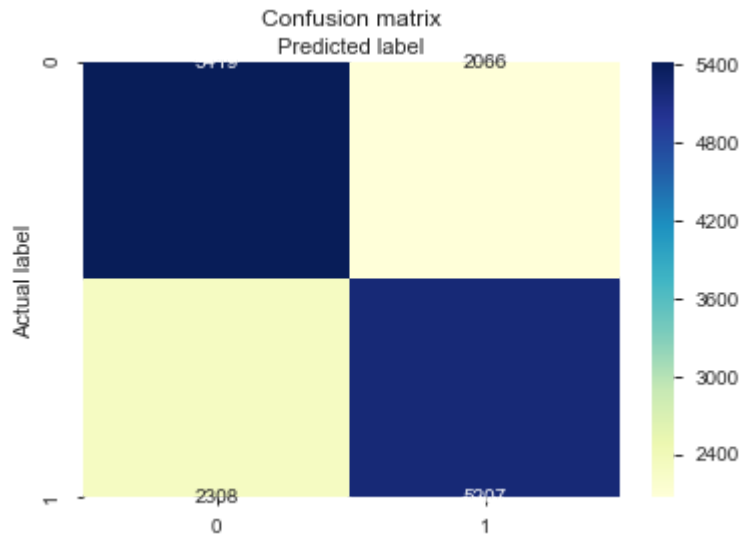
```
C:\Users\97150\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().
```

Out[54]:
```
0.7084
```

In [55]:
```
cnf_matrix = metrics.confusion_matrix(ytest_opt, y_pred1)
print(cnf_matrix)
class_names=[0,1] # name  of classes
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
# create heatmap
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

```
[[5419 2066]
 [2308 5207]]
```

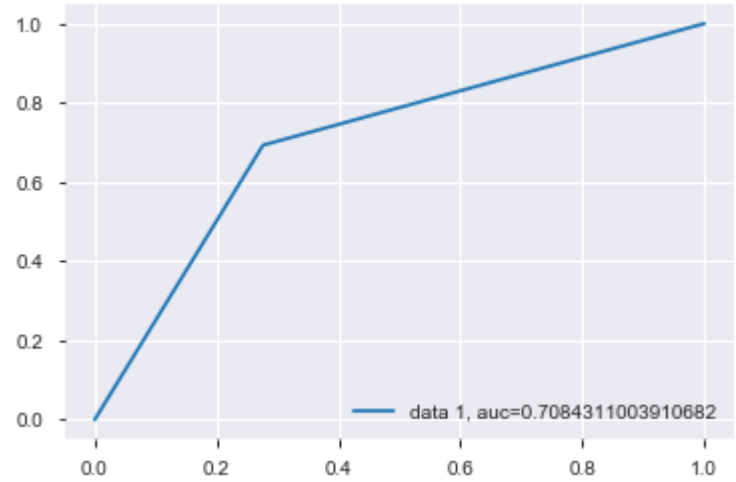
Out[55]:
```
Text(0.5, 12.5, 'Predicted label')
```



In [56]:
```
print("Accuracy:",metrics.accuracy_score(ytest_opt, y_pred1))
print("Precision:",metrics.precision_score(ytest_opt, y_pred1))
print("Recall:",metrics.recall_score(ytest_opt, y_pred1))
```

```
Accuracy: 0.7084
Precision: 0.7159356524130345
Recall: 0.6928809048569528
```

In [57]:
```
y_pred_proba = rf.predict(xtest_opt)
fpr, tpr, _ = metrics.roc_curve(ytest_opt, y_pred_proba)
auc = metrics.roc_auc_score(ytest_opt, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()
print("Auc:", auc)
```




```
Auc: 0.7084311003910682
```

Extreme Gradient Boosting

In [59]:
```
import xgboost as xgb
from xgboost import XGBClassifier
from sklearn.model_selection import cross_val_score, KFold
```

In [60]:
```
xgbc = XGBClassifier()

XGBClassifier(objective='binary:logistic',base_score=0.5, booster='gbtree', co
lsample_bylevel=1,
       colsample_bynode=1, colsample_bytree=1, gamma=0, learning_rate=0.1,
       max_delta_step=0, max_depth=3, min_child_weight=1, missing=None,
       n_estimators=100, n_jobs=1, nthread=None,
       random_state=0, reg_alpha=0,
       reg_lambda=1, scale_pos_weight=1, seed=None, silent=None,
       subsample=1, verbosity=1)
```

Out[60]:
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=None,
              importance_type='gain', interaction_constraints=None,
              learning_rate=0.1, max_delta_step=0, max_depth=3,
              min_child_weight=1, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=1, nthread=None, num_parallel_tree=Non
e,
              objective='binary:logistic', random_state=0, reg_alpha=0,
              reg_lambda=1, scale_pos_weight=1, seed=None, silent=None,
              subsample=1, tree_method=None, validate_parameters=False,
              verbosity=1)
```

In [61]:
```
xgbc.fit(xtrain_opt, ytrain_opt)
```

```
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:21
9: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:25
2: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
```

Out[61]:
```
XGBClassifier(base_score=0.5, booster=None, colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints=None,
              learning_rate=0.300000012, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=0, num_parallel_tree=1,
              objective='binary:logistic', random_state=0, reg_alpha=0,
              reg_lambda=1, scale_pos_weight=1, subsample=1, tree_method=Non
e,
              validate_parameters=False, verbosity=None)
```

In [62]:
```
xgbc.score(xtest_opt,ytest_opt)
```

Out[62]:
```
0.7235333333333334
```

Using Kfold Cross Validation

In [63]:
```
kfold = KFold(n_splits=10, shuffle=True)
kf_cv_scores =  cross_val_score(xgbc, xtrain_opt, ytrain_opt, cv=kfold )
print("K-fold CV average score: %.2f" % kf_cv_scores.mean())
kf_cv_roc = cross_val_score(xgbc, xtrain_opt, ytrain_opt,scoring='roc_auc',cv=
kfold )
print("K-fold AUC OF ROC average score: %.2f" % kf_cv_roc.mean())
```

```
K-fold CV average score: 0.72
```

```
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:21
9: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:25
2: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:21
9: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:25
2: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:21
9: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:25
2: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:21
9: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:25
2: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:21
9: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:25
2: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:21
9: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:25
2: DataConversionWarning: A column-vector y was passed when a 1d array was ex
```

```
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:21
9: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:25
2: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:21
9: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:25
2: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:21
9: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\97150\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:25
2: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)

K-fold AUC OF ROC average score: 0.82
```

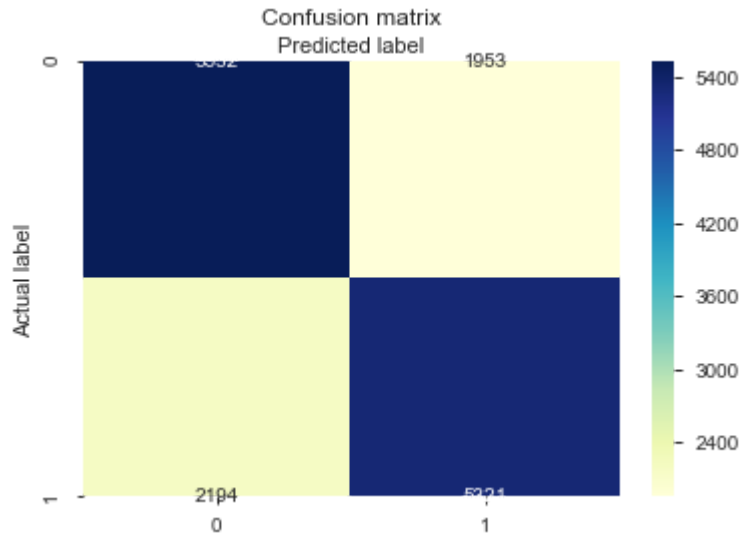
In [64]:

```
ypred = xgbc.predict(xtest_opt)
```

In [65]:
```
cnf_matrix = metrics.confusion_matrix(ytest_opt, ypred)
print(cnf_matrix)
class_names=[0,1] # name  of classes
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
# create heatmap
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

```
[[5532 1953]
 [2194 5321]]
```

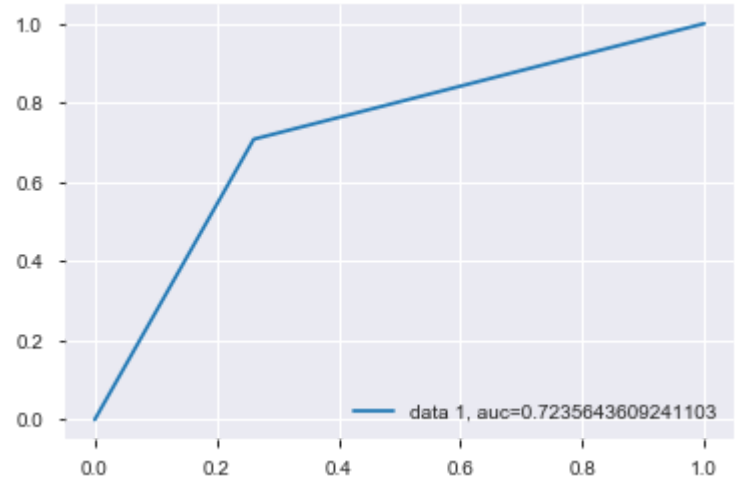
Out[65]:
```
Text(0.5, 12.5, 'Predicted label')
```



In [66]:
```
print("Accuracy:",metrics.accuracy_score(ytest_opt, ypred))
print("Precision:",metrics.precision_score(ytest_opt, ypred))
print("Recall:",metrics.recall_score(ytest_opt, ypred))
```

```
Accuracy: 0.7235333333333334
Precision: 0.731509485839978
Recall: 0.7080505655355954
```

In [67]:
```
y_pred_proba = xgbc.predict(xtest_opt)
fpr, tpr, _ = metrics.roc_curve(ytest_opt, y_pred_proba)
auc = metrics.roc_auc_score(ytest_opt, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()
print("Auc:", auc)
```




```
Auc: 0.7235643609241103
```

In [ ]:

In [ ]: