Lab1

April 5, 2022

1 Lab 1

1.1 ## Amit Avigdor 316178144, Barak Bonker 316177708

1.2 Task 1

```
[ ]: import pandas as pd
```

```
[130]: file = pd.read_csv('dmc2010_train.txt', delimiter = ";")
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:2882:
DtypeWarning: Columns (11) have mixed types.Specify dtype option on import or
set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
[ ]: file.head(5)
```

```
[ ]:    customernumber        date  salutation  title  domain datecreated  \
     0           41191  2008-12-01           0      0       9  2008-12-01
     1           38860  2008-12-16           1      0       4  2008-12-16
     2           61917  2008-08-19           0      0      12  2008-08-19
     3           40647  2008-06-16           1      0       8  2008-06-16
     4            1347  2008-08-08           0      0       1  2008-08-08

        newsletter  model  paymenttype  deliverytype  …  w2 w3  w4 w5  w6  w7  \
     0           0      2            2             0  …   0  0   0  0   0   0
     1           0      1            1             1  …   0  0   0  0   0   0
     2           0      1            0             0  …   0  0   0  0   0   0
     3           0      1            0             0  …   0  0   0  2   0   0
     4           0      1            1             1  …   2  0   0  0   0   0

        w8  w9  w10  target90
     0   0   0    0         0
     1   0   0    0         0
     2   0   1    0         0
     3   0   0    0         0
     4   0   0    0         0
```

```
[5 rows x 38 columns]
```

1.3 1.a.

```
[ ]: file.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32428 entries, 0 to 32427
Data columns (total 38 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   customernumber       32428 non-null  int64
 1   date                 32428 non-null  object
 2   salutation           32428 non-null  int64
 3   title                32428 non-null  int64
 4   domain               32428 non-null  int64
 5   datecreated          32428 non-null  object
 6   newsletter           32428 non-null  int64
 7   model                32428 non-null  int64
 8   paymenttype          32428 non-null  int64
 9   deliverytype         32428 non-null  int64
 10  invoicepostcode      32428 non-null  int64
 11  delivpostcode        1392 non-null   object
 12  voucher              32428 non-null  int64
 13  advertisingdatacode  6523 non-null   object
 14  case                 32428 non-null  int64
 15  numberitems          32428 non-null  int64
 16  gift                 32428 non-null  int64
 17  entry                32428 non-null  int64
 18  points               32428 non-null  int64
 19  shippingcosts        32428 non-null  int64
 20  deliverydatepromised 32428 non-null  object
 21  deliverydatereal     32428 non-null  object
 22  weight               32428 non-null  int64
 23  remi                 32428 non-null  int64
 24  cancel               32428 non-null  int64
 25  used                 32428 non-null  int64
 26  w0                   32428 non-null  int64
 27  w1                   32428 non-null  int64
 28  w2                   32428 non-null  int64
 29  w3                   32428 non-null  int64
 30  w4                   32428 non-null  int64
 31  w5                   32428 non-null  int64
 32  w6                   32428 non-null  int64
 33  w7                   32428 non-null  int64
 34  w8                         32428 non-null  int64
 35  w9                         32428 non-null  int64
 36  w10                        32428 non-null  int64
 37  target90                   32428 non-null  int64
dtypes: int64(32), object(6)
memory usage: 9.4+ MB
```

1.4 1.b.

customernumber - numeric date - ordinal salutation - numeric title - binary domain - numeric datecreated - ordinal newsletter - binary model - numeric paymenttype - numeric deliverytype - binary invoicepostcode - numeric delivpostcode - numeric voucher - binary advertisingdatacode - numinal case - ordinal numberitems - numeric gift - binary entry - binary points - binary shippingcosts - binary deliverydatepromised - ordinal deliverydatereal - ordinal weight - numeric remi - numeric cancel - numeric used - numeric w0-10 - numeric target90 - binary

1.5 1.c.

[ ]:

```python
for i in range(1,38):
    if i in range(22,36) or i == 15:
        print(file.columns[i] + " average:")
        print(file[file.columns[i]].mean(), end="\n\n")
    else:
        print(file.columns[i] + " most common:")
        print(file[file.columns[i]].mode()[0], end="\n\n")
```

```
date most common:
2008-12-15

salutation most common:
0

title most common:
0

domain most common:
12

datecreated most common:
2008-12-15

newsletter most common:
0

model most common:
1
```

```
paymenttype most common:
0

deliverytype most common:
0

invoicepostcode most common:
44

delivpostcode most common:
22.0

voucher most common:
0

advertisingdatacode most common:
BQ

case most common:
4

numberitems average:
2.0195510053040584

gift most common:
0

entry most common:
0

points most common:
0

shippingcosts most common:
0

deliverydatepromised most common:
2008-12-23

deliverydatereal most common:
0000-00-00

weight average:
637.9208091772542

remi average:
0.059979030467497224
```

```
cancel average:
0.06161342050080178

used average:
0.06886024423337857

w0 average:
0.9021216232885161

w1 average:
0.4043419267299864

w2 average:
0.276643641297644

w3 average:
0.01890341680029604

w4 average:
0.047027260392253606

w5 average:
0.18098556802763044

w6 average:
0.027907980757370172

w7 average:
0.023128160848649316

w8 average:
0.00018502528678919454

w9 average:
0.16498088072036513

w10 most common:
0

target90 most common:
0
```

1.6 1.d.

```
[ ]: for i in range(1,38):
         if i in range(22,36) or i == 15:
             print(file.columns[i] + " min and max:")
             print(file[file.columns[i]].min() , file[file.columns[i]].max(),␣
      ↪end="\n\n")
         else:
             print(file.columns[i] + " unique values:")
             print(file[file.columns[i]].unique(), end="\n\n")
```

```
date unique values:
['2008-12-01' '2008-12-16' '2008-08-19' '2008-06-16' '2008-08-08'
 '2008-08-10' '2008-12-21' '2008-09-04' '2008-06-25' '2008-08-02'
 '2008-12-20' '2008-04-01' '2008-09-12' '2008-05-06' '2008-10-02'
 '2008-04-02' '2008-12-08' '2008-11-18' '2008-07-15' '2008-10-03'
 '2008-10-20' '2008-11-27' '2008-12-30' '2008-04-13' '2008-04-25'
 '2008-08-30' '2008-05-17' '2008-08-24' '2008-12-19' '2008-04-21'
 '2008-10-27' '2008-07-30' '2008-06-11' '2008-09-30' '2008-04-23'
 '2008-11-04' '2008-11-20' '2008-12-15' '2008-12-18' '2008-05-31'
 '2008-06-27' '2008-08-26' '2008-10-04' '2008-08-20' '2008-04-09'
 '2008-08-12' '2008-11-28' '2008-12-14' '2008-05-22' '2008-10-19'
 '2008-05-29' '2008-12-03' '2008-11-21' '2008-12-22' '2008-09-20'
 '2008-12-02' '2008-09-10' '2008-06-13' '2008-10-28' '2008-12-12'
 '2008-04-20' '2008-12-26' '2008-12-29' '2008-12-28' '2008-10-26'
 '2008-05-30' '2008-08-25' '2008-05-18' '2008-07-07' '2008-09-05'
 '2008-07-26' '2008-04-24' '2008-07-31' '2008-10-07' '2008-11-05'
 '2008-05-04' '2008-08-05' '2008-04-30' '2008-06-06' '2008-09-06'
 '2008-04-07' '2008-07-09' '2008-11-13' '2008-05-19' '2008-11-25'
 '2008-10-09' '2008-10-21' '2008-05-28' '2008-08-06' '2008-09-19'
 '2008-06-17' '2008-08-15' '2008-11-08' '2008-07-16' '2008-04-10'
 '2008-09-08' '2008-12-17' '2008-09-25' '2008-12-04' '2008-08-09'
 '2008-05-13' '2008-06-28' '2008-04-16' '2008-07-23' '2008-12-24'
 '2008-06-02' '2008-09-28' '2008-05-08' '2008-10-11' '2008-10-01'
 '2008-11-09' '2008-11-22' '2008-12-11' '2008-05-21' '2008-04-06'
 '2008-10-10' '2008-07-08' '2008-08-04' '2008-08-27' '2008-09-23'
 '2008-04-03' '2008-06-14' '2008-07-02' '2008-09-09' '2008-12-23'
 '2008-04-26' '2008-08-28' '2008-11-02' '2008-11-16' '2008-07-20'
 '2008-07-03' '2008-08-03' '2008-10-05' '2008-10-13' '2008-05-01'
 '2008-12-27' '2008-06-07' '2008-06-21' '2008-12-06' '2008-10-14'
 '2008-04-04' '2008-07-10' '2008-05-27' '2008-07-01' '2008-07-24'
 '2008-09-07' '2008-11-12' '2008-08-22' '2008-07-18' '2008-10-08'
 '2008-11-07' '2008-09-27' '2008-08-18' '2008-09-29' '2008-06-20'
 '2008-06-10' '2008-12-25' '2008-06-15' '2008-07-28' '2008-10-12'
 '2008-10-22' '2008-06-26' '2008-07-21' '2008-08-17' '2008-09-15'
 '2008-08-01' '2008-11-23' '2008-08-23' '2008-07-13' '2008-10-06'
 '2008-05-26' '2008-09-02' '2008-05-02' '2008-12-13' '2008-11-19'
 '2008-05-03' '2008-09-22' '2008-04-27' '2008-05-25' '2008-04-12'
 '2009-03-09' '2008-06-29' '2008-11-10' '2008-08-13' '2008-05-24'
 '2008-08-11' '2008-06-08' '2009-01-03' '2008-10-30' '2008-10-24'
 '2008-09-03' '2008-04-17' '2008-10-29' '2008-12-09' '2008-07-19'
 '2008-11-14' '2008-10-25' '2008-05-20' '2008-11-06' '2008-10-15'
 '2008-04-15' '2008-07-12' '2008-11-11' '2008-08-16' '2008-12-05'
 '2008-12-10' '2008-09-26' '2008-07-22' '2008-08-21' '2008-06-30'
 '2008-07-27' '2008-07-11' '2008-11-03' '2008-05-15' '2008-04-29'
 '2008-06-19' '2008-04-11' '2008-06-05' '2008-09-01' '2008-07-29'
 '2008-09-17' '2008-04-19' '2008-09-21' '2008-07-04' '2008-04-28'
 '2008-10-18' '2008-05-11' '2008-06-12' '2008-06-04' '2008-11-26'
 '2008-04-22' '2008-08-31' '2008-05-12' '2008-09-18' '2008-11-30'
 '2008-06-01' '2008-08-07' '2008-06-23' '2008-10-23' '2008-08-29'
 '2008-11-24' '2008-11-29' '2008-06-24' '2008-07-14' '2008-10-17'
 '2008-10-16' '2008-10-31' '2008-09-16' '2008-07-06' '2008-04-08'
 '2008-05-09' '2008-06-22' '2008-06-03' '2008-04-14' '2008-08-14'
 '2008-06-18' '2008-11-01' '2008-09-14' '2008-05-05' '2008-12-07'
 '2008-04-18' '2008-11-17' '2008-04-05' '2008-05-23' '2008-05-16'
 '2008-11-15' '2008-07-17' '2008-05-07' '2008-05-14' '2008-09-11'
 '2008-12-31' '2009-03-12' '2008-06-09' '2008-07-05' '2008-09-13'
 '2008-05-10' '2008-09-24' '2009-01-20' '2009-03-11' '2009-02-07'
 '2009-01-18' '2009-03-17' '2008-07-25' '2009-03-13' '2009-01-17'
 '2009-02-02' '2009-03-01' '2009-02-21' '2009-03-25' '2009-01-13'
 '2009-01-10' '2009-02-03' '2009-01-04' '2009-02-26' '2009-01-22'
 '2009-01-09' '2009-02-18' '2009-02-27' '2009-01-23' '2009-03-04'
 '2009-02-06' '2009-01-21' '2009-03-30' '2009-02-12' '2009-01-08'
 '2009-01-06' '2009-01-14' '2009-03-24' '2009-03-31' '2009-03-22'
 '2009-03-16' '2009-01-29' '2009-01-11' '2009-02-15' '2009-02-25'
 '2009-03-26' '2009-03-27' '2009-01-02' '2009-03-15' '2009-02-09'
 '2009-01-27' '2009-02-11' '2009-01-05' '2009-02-08' '2009-01-12'
 '2009-01-07' '2009-02-05' '2009-01-19' '2009-01-30' '2009-03-06'
 '2009-02-13' '2009-01-26' '2009-01-25' '2009-03-03' '2009-02-24'
 '2009-01-28' '2009-01-01' '2009-02-10' '2009-02-19' '2009-01-16'
 '2009-03-10' '2009-01-24' '2009-03-20' '2009-03-19' '2009-03-14'
 '2009-03-02' '2009-02-16' '2009-02-01' '2009-01-15' '2009-03-28'
 '2009-02-14']

salutation unique values:
[0 1 2]

title unique values:
[0 1]

domain unique values:
[ 9  4 12  8  1  6  2 11  5  3  0 10  7]

datecreated unique values:
['2008-12-01' '2008-12-16' '2008-08-19' '2008-06-16' '2008-08-08'
```

```
'2008-08-10' '2008-12-21' '2008-09-04' '2008-06-24' '2008-08-02'
'2008-12-20' '2008-04-01' '2008-09-12' '2008-05-06' '2008-10-02'
'2008-04-02' '2008-12-08' '2008-11-18' '2008-07-15' '2008-10-03'
'2008-10-20' '2008-11-21' '2008-12-30' '2008-04-13' '2008-04-25'
'2008-08-30' '2008-05-17' '2008-08-24' '2008-12-19' '2008-04-21'
'2008-10-27' '2008-11-27' '2008-07-30' '2008-06-11' '2008-09-30'
'2008-04-23' '2008-11-04' '2008-11-20' '2008-12-15' '2008-12-18'
'2008-05-31' '2008-06-27' '2008-08-26' '2008-10-04' '2008-08-20'
'2008-04-09' '2008-08-12' '2008-11-28' '2008-12-14' '2008-05-22'
'2008-10-19' '2008-05-29' '2008-06-09' '2008-12-22' '2008-09-20'
'2008-12-02' '2008-09-10' '2008-06-13' '2008-10-28' '2008-12-03'
'2008-12-12' '2008-04-20' '2008-12-26' '2008-12-29' '2008-12-28'
'2008-10-26' '2008-05-30' '2008-08-25' '2008-05-18' '2008-07-07'
'2008-09-05' '2008-07-26' '2008-04-24' '2008-07-31' '2008-11-10'
'2008-10-07' '2008-11-05' '2008-05-04' '2008-08-05' '2008-04-30'
'2008-06-06' '2008-09-06' '2008-04-07' '2008-07-09' '2008-11-13'
'2008-05-19' '2008-11-25' '2008-10-09' '2008-10-21' '2008-05-28'
'2008-08-06' '2008-09-17' '2008-06-17' '2008-08-15' '2008-11-08'
'2008-07-16' '2008-04-10' '2008-09-08' '2008-12-17' '2008-09-25'
'2008-12-04' '2008-08-09' '2008-05-13' '2008-06-28' '2008-04-16'
'2008-08-11' '2008-07-23' '2008-12-24' '2008-06-02' '2008-09-28'
'2008-05-08' '2008-10-11' '2008-10-01' '2008-11-09' '2008-11-22'
'2008-12-11' '2008-05-21' '2008-04-06' '2008-04-12' '2008-10-10'
'2008-07-08' '2008-08-04' '2008-05-20' '2008-04-03' '2008-06-14'
'2008-07-02' '2008-09-09' '2008-12-23' '2008-04-26' '2008-08-28'
'2008-11-02' '2008-11-16' '2008-07-03' '2008-08-03' '2008-10-05'
'2008-07-20' '2008-10-13' '2008-05-01' '2008-12-27' '2008-06-07'
'2008-06-21' '2008-12-06' '2008-10-14' '2008-04-04' '2008-07-10'
'2008-09-19' '2008-05-27' '2008-07-01' '2008-07-06' '2008-07-24'
'2008-09-07' '2008-11-12' '2008-06-25' '2008-08-22' '2008-07-18'
'2008-10-08' '2008-11-07' '2008-09-27' '2008-08-18' '2008-09-29'
'2008-06-20' '2008-06-10' '2008-12-25' '2008-06-15' '2008-07-28'
'2008-10-12' '2008-07-21' '2008-08-17' '2008-09-14' '2008-08-01'
'2008-11-23' '2008-08-23' '2008-07-13' '2008-10-06' '2008-09-02'
'2008-05-02' '2008-12-13' '2008-11-19' '2008-05-03' '2008-09-22'
'2008-04-27' '2008-05-25' '2008-06-29' '2008-08-13' '2008-05-24'
'2008-06-08' '2008-10-30' '2008-10-24' '2008-09-03' '2008-04-17'
'2008-06-26' '2008-10-29' '2008-12-09' '2008-07-19' '2008-11-14'
'2008-10-25' '2008-11-06' '2008-10-15' '2008-04-15' '2008-07-12'
'2008-11-11' '2008-08-16' '2008-12-05' '2008-12-10' '2008-09-26'
'2008-07-22' '2008-08-21' '2008-06-30' '2008-07-27' '2008-07-11'
'2008-11-03' '2008-05-15' '2008-09-23' '2008-08-27' '2008-08-14'
'2008-04-29' '2008-06-19' '2008-04-11' '2008-06-05' '2008-09-01'
'2008-07-29' '2008-11-24' '2008-12-07' '2008-04-19' '2008-09-21'
'2008-07-04' '2008-04-28' '2008-06-04' '2008-10-18' '2008-05-11'
'2008-06-12' '2008-11-26' '2008-04-22' '2008-08-31' '2008-05-12'
'2008-09-18' '2008-11-30' '2008-06-01' '2008-08-07' '2008-06-23'
'2008-10-23' '2008-08-29' '2008-11-29' '2008-07-14' '2008-10-17'
```

```
 '2008-10-16' '2008-10-31' '2008-09-16' '2008-04-08' '2008-05-09'
 '2008-05-26' '2008-06-22' '2008-04-14' '2008-10-22' '2008-06-03'
 '2008-11-01' '2008-05-05' '2008-04-18' '2008-11-17' '2008-04-05'
 '2008-05-23' '2008-05-16' '2008-11-15' '2008-07-17' '2008-05-14'
 '2008-09-15' '2008-09-11' '2008-06-18' '2008-05-07' '2008-12-31'
 '2008-07-05' '2008-09-13' '2008-05-10' '2008-09-24' '2008-07-25']

newsletter unique values:
[0 1]

model unique values:
[2 1 3]

paymenttype unique values:
[2 1 0 3]

deliverytype unique values:
[0 1]

invoicepostcode unique values:
[58 34 51 25 41 95 78 77 86 97 50 40 99 85 88 17 70 30 68 15 12 63 18 89
 49 20 42 21 73 22 55 23  7 65 79 57 46 27 10 52 66 91 24 74 61 56 76 38
 26 32 39 84 29 33  1 16 13 80  2 14 45 81 53 90 60  3 64 54 71 28 36 67
 44 31 47 35  9  4 69 59 92 93 37 48 94 96 72 83 75  8 82  6 87 19 98 11
  0]

delivpostcode unique values:
[nan 99.0 97.0 15.0 70.0 50.0 14.0 53.0 35.0 24.0 44.0 22.0 41.0 45.0 88.0
 58.0 64.0 91.0 81.0 42.0 46.0 27.0 4.0 52.0 72.0 17.0 23.0 21.0 60.0 55.0
 82.0 28.0 30.0 10.0 90.0 67.0 86.0 40.0 51.0 59.0 65.0 57.0 94.0 63.0
 49.0 7.0 92.0 85.0 71.0 47.0 96.0 80.0 73.0 34.0 37.0 76.0 13.0 48.0 16.0
 6.0 25.0 12.0 89.0 66.0 69.0 8.0 20.0 1.0 79.0 0.0 33.0 32.0 98.0 83.0
 95.0 93.0 78.0 31.0 61.0 38.0 36.0 74.0 54.0 2.0 19.0 56.0 18.0 68.0 75.0
 3.0 29.0 87.0 9.0 11.0 39.0 26.0 77.0 84.0 '61' '53' '23' '52' '81' '25'
 '44' '13' '33' '50' '99' '65' '56' '14' '68' '72' '49' '46' '22' '76'
 '04' '86' '41' '57' '45' '38' '55' '66' '37' '71' '21' '89' '47' '20'
 '40' '58' '97' '64' '88' '48' '31' '69' '36' '26' '09' '06' '60' '30'
 '84' '39' '82' '51' '91' '87' '16' '90' '10' '85' '42' '70' '18' '59'
 '74' '17' '08' '29' '27' '01' '12' '28' '24' '73' '80' '32' 'EN' '79'
 '63' '19' '92' '78' '35' '03' '83' '93' '95' '96' '98' '75' '67' '02'
 '54' '00' '34' '77' 'Nl']

voucher unique values:
[1 0]

advertisingdatacode unique values:
[nan 'BR' 'BQ' 'AP' 'CA' 'BD' 'AB' 'BC' 'BI' 'BT' 'AE' 'AQ' 'AX' 'AF' 'AH'
 'BF' 'AR' 'BL' 'BO' 'BZ' 'AV' 'BB' 'AT' 'BM' 'BA' 'AZ' 'BY' 'AW' 'AG'
 'AK' 'AO' 'AL' 'AU' 'AY' 'AI' 'AM' 'BS' 'BX' 'BV' 'BG' 'AC' 'BK' 'AD'
 'BE' 'AS']

case unique values:
[2 1 3 4 5]

numberitems min and max:
1 50

gift unique values:
[0 1]

entry unique values:
[1 0]

points unique values:
[0]

shippingcosts unique values:
[0 1]

deliverydatepromised unique values:
['2008-12-03' '2008-12-30' '2008-09-02' '2008-06-17' '2008-08-11'
 '2008-08-12' '2008-12-23' '2008-09-05' '2008-06-26' '2008-08-05'
 '2008-04-03' '2008-09-15' '2008-05-08' '2008-10-03' '2008-12-10'
 '2008-11-20' '2008-07-16' '2008-10-17' '2008-10-22' '2008-12-01'
 '2009-01-13' '2008-04-15' '2008-04-28' '2008-05-20' '2008-08-26'
 '2008-12-22' '2008-05-06' '2008-10-28' '2008-07-31' '2008-06-13'
 '2008-10-01' '2008-04-24' '2008-12-18' '2008-11-06' '2008-11-24'
 '2008-12-17' '2008-06-16' '2008-07-01' '2008-08-27' '2008-10-07'
 '2008-08-22' '2009-04-13' '2009-08-17' '2009-02-16' '2009-01-19'
 '2008-12-16' '2009-05-26' '2008-10-21' '2008-06-02' '2008-11-05'
 '2008-12-04' '2008-06-18' '2008-11-25' '2008-12-24' '2008-09-23'
 '2009-01-01' '2008-11-04' '2008-09-24' '2008-10-29' '2008-12-05'
 '2008-04-22' '2008-12-31' '2008-07-08' '2008-09-09' '2008-04-29'
 '2008-04-25' '2008-08-01' '2009-07-09' '2008-11-21' '2008-10-09'
 '2008-11-07' '2008-08-06' '2008-05-02' '2008-06-10' '2008-04-09'
 '2008-07-10' '2008-11-14' '2008-07-09' '2008-05-21' '2008-11-27'
 '2008-10-10' '2008-05-30' '2008-08-07' '2008-08-18' '2008-07-30'
 '2008-04-11' '2009-09-29' '2008-05-15' '2008-04-18' '2008-11-19'
 '2008-07-24' '2008-04-02' '2008-12-26' '2008-08-21' '2008-09-30'
 '2008-05-13' '2008-10-14' '2009-05-21' '2009-11-12' '2008-12-19'
 '2008-12-12' '2008-05-23' '2008-05-09' '2008-04-08' '2008-10-13'
 '2009-07-10' '2009-05-06' '2008-09-10' '2008-04-16' '2008-10-08'
 '2008-07-03' '2008-05-22' '2008-11-18' '2008-08-04' '2008-10-27'
 '2008-07-07' '2008-07-22' '2008-12-02' '2009-05-05' '2008-06-24'
 '2008-10-16' '2008-12-08' '2009-01-06' '2008-07-14' '2008-09-11'
 '2008-09-22' '2008-05-29' '2009-02-03' '2008-07-17' '2008-07-28'
```

```
'2008-10-02' '2008-08-25' '2008-11-11' '2008-06-23' '2008-06-11'
'2008-10-24' '2008-06-27' '2008-07-23' '2008-08-19' '2008-09-17'
'2008-08-14' '2008-07-11' '2009-04-28' '2009-10-08' '2008-12-25'
'2008-06-19' '2008-07-15' '2008-09-16' '2008-05-27' '2008-05-07'
'2008-06-25' '2009-12-22' '2009-12-15' '2009-12-30' '2008-12-09'
'2008-04-07' '2009-10-07' '2009-03-11' '2008-11-12' '2008-08-15'
'2008-10-06' '2008-12-29' '2008-10-15' '2009-08-05' '2008-11-03'
'2008-09-08' '2008-05-05' '2008-04-21' '2009-07-31' '2009-12-01'
'2008-06-03' '2008-11-10' '2009-01-07' '2008-12-11' '2009-12-11'
'2008-09-18' '2008-11-17' '2008-11-13' '2009-09-08' '2008-12-15'
'2009-08-19' '2008-07-18' '2008-04-14' '2008-06-20' '2008-06-30'
'2008-10-23' '2008-07-29' '2008-06-12' '2009-07-30' '2010-01-04'
'2008-05-19' '2008-11-26' '2008-08-13' '2008-08-29' '2008-06-06'
'2008-06-09' '2009-04-14' '2009-04-09' '2009-08-25' '2008-09-19'
'2009-07-21' '2008-09-04' '2008-05-26' '2008-07-21' '2008-04-23'
'2008-10-30' '2008-09-03' '2008-05-14' '2008-06-05' '2008-11-28'
'2009-06-02' '2008-04-10' '2009-07-28' '2008-08-08' '2008-09-26'
'2009-10-14' '2008-04-17' '2009-11-02' '2009-11-16' '2008-07-02'
'2008-10-20' '2009-12-02' '2009-04-10' '2008-08-20' '2008-10-31'
'2009-05-29' '2008-04-30' '2009-10-30' '2008-09-01' '2009-09-28'
'2009-12-18' '2008-07-04' '2009-08-04' '2009-08-07' '2009-12-03'
'2009-12-21' '2008-08-28' '2009-12-28' '2008-05-16' '2009-09-11'
'2008-07-25' '2009-05-08' '2009-12-08' '2009-01-12' '2009-11-27'
'2008-06-04' '2009-04-24' '2008-09-12' '2009-08-18' '2009-11-03'
'2008-09-29' '2008-09-25' '2009-08-03' '2009-11-09' '2009-03-16'
'2010-01-01' '2009-10-27' '2009-04-03' '2009-02-06' '2009-05-20'
'2009-06-18' '2009-10-13' '2009-12-10' '2009-04-17' '2009-11-17'
'2009-11-26' '2009-02-09' '2009-01-22' '2009-03-12' '2009-06-16'
'2009-06-29' '2009-02-10' '2008-05-28' '2009-06-08' '2009-05-19'
'2009-11-19' '2009-09-25' '2009-04-23' '2009-06-23' '2009-09-21'
'2009-01-20' '2009-10-09' '2009-04-07' '2009-11-24' '2009-01-08'
'2009-09-22' '2009-07-16' '2009-03-19' '2009-01-05' '2009-11-20'
'2009-09-24' '2009-10-01' '2009-08-26' '2009-07-14' '2009-06-17'
'2009-08-12' '2009-05-12' '2009-09-30' '2009-08-06' '2009-08-31'
'2009-11-18' '2009-05-07' '2009-09-10' '2009-12-24' '2009-09-09'
'2009-03-17' '2009-02-02' '2009-10-29' '2009-12-14' '2009-07-23'
'2009-01-02' '2009-11-30' '2009-09-07' '2009-12-07' '2008-04-04'
'2009-04-21' '2009-10-02' '2009-10-26' '2009-06-04' '2009-06-12'
'2009-02-04' '2009-01-27' '2009-03-03' '2009-02-24' '2009-10-06'
'2009-06-30' '2009-01-29' '2009-09-18' '2009-05-25' '2009-04-06'
'2009-07-24' '2009-08-24' '2009-04-16' '2009-10-20' '2009-01-14'
'2009-02-13' '2009-10-16' '2009-12-23' '2009-12-16' '2009-04-27'
'2009-06-19' '2009-11-06' '2009-08-27' '2009-10-21' '2009-07-15'
'2009-10-15' '2010-01-13' '2009-09-01' '2009-07-13' '2009-06-11'
'2009-05-11' '2009-09-03' '2009-11-05' '2009-08-11' '2009-08-13'
'2009-07-03' '2009-02-05' '2009-08-28' '2009-09-15' '2009-05-15'
'2009-05-01' '2009-03-13' '2009-12-17' '2009-04-08' '2009-01-09'
'2009-06-03' '2009-11-23' '2009-07-07' '2009-02-27' '2009-08-10'
```

```
 '2009-11-10' '2009-06-26' '2009-02-19' '2009-12-25' '2009-01-23'
 '2009-03-10' '2010-01-07' '2010-01-05' '2009-10-05' '2009-10-22'
 '2009-06-09' '2009-12-04' '2009-10-12' '2009-06-24' '2009-12-31'
 '2009-05-04' '2009-03-05' '2009-04-15' '2009-08-14' '2010-02-22'
 '2009-07-06' '2009-04-01' '2009-08-21' '2009-06-25' '2009-12-29'
 '2009-01-16' '2009-07-20' '2009-07-02' '2009-09-04' '2009-04-30'
 '2010-01-19' '2009-03-25' '2009-04-02' '2009-01-30' '2009-07-01'
 '2009-09-23' '2009-07-27' '2009-07-22' '2009-11-13' '2009-03-02'
 '2009-05-28' '2009-06-01' '2009-09-02' '2009-01-21' '2009-02-17'
 '2009-10-23' '2010-02-17' '2009-09-16' '2009-11-25' '2009-05-22'
 '2009-03-27' '2009-11-11' '2009-06-05' '2009-11-04' '2009-06-15'
 '2009-09-14' '2009-03-31' '2009-05-27' '2009-05-18' '2009-01-15'
 '2009-01-28' '2009-08-20' '2009-01-26' '2009-12-09' '2009-07-17'
 '2009-09-17' '2009-03-26' '4746-07-23' '2009-04-22' '2009-03-30'
 '2009-04-20' '4746-11-26' '2009-02-26' '2009-03-24' '2009-06-22'
 '2009-05-14' '4746-06-13' '2009-07-08' '2009-03-09' '2009-04-29'
 '2009-03-18' '2009-02-12' '2009-03-04' '2009-10-28' '2009-05-13'
 '2009-06-10' '2009-02-20' '2009-02-11' '2010-03-09' '4746-11-15'
 '4746-05-08' '4746-05-15' '2009-10-19' '2009-03-23' '4746-10-01'
 '2009-02-25' '4746-11-18' '2009-02-18' '2010-02-12' '2010-02-24'
 '2010-02-23']

deliverydatereal unique values:
['2008-12-02' '2009-02-03' '2008-08-28' '0000-00-00' '2008-08-08'
 '2008-08-11' '2008-12-22' '2008-09-04' '2008-06-27' '2008-08-04'
 '2008-04-02' '2008-09-13' '2008-05-07' '2008-10-05' '2008-12-09'
 '2008-12-21' '2008-11-19' '2008-07-15' '2008-10-10' '2008-10-21'
 '2008-11-27' '2009-01-03' '2008-04-14' '2008-04-25' '2008-09-01'
 '2008-05-19' '2008-11-28' '2008-06-12' '2008-09-30' '2008-04-23'
 '2008-12-17' '2008-11-06' '2008-11-21' '2008-12-16' '2008-12-20'
 '2008-07-18' '2008-06-28' '2008-08-27' '2008-10-06' '2008-08-21'
 '2008-05-08' '2008-08-25' '2009-02-13' '2009-01-16' '2008-07-01'
 '2008-10-20' '2008-05-30' '2008-11-04' '2008-06-17' '2008-11-24'
 '2008-12-23' '2008-09-22' '2008-12-29' '2008-10-22' '2008-12-04'
 '2008-09-19' '2008-06-25' '2008-10-28' '2008-12-13' '2008-04-21'
 '2008-12-27' '2008-12-30' '2008-10-31' '2008-08-26' '2008-07-08'
 '2008-09-06' '2008-07-30' '2008-07-31' '2008-12-06' '2008-11-20'
 '2008-10-08' '2008-05-05' '2008-04-30' '2008-06-07' '2008-04-08'
 '2008-07-09' '2008-11-13' '2008-05-20' '2008-11-26' '2008-11-12'
 '2008-04-03' '2008-09-20' '2008-08-16' '2008-07-19' '2008-04-11'
 '2008-09-09' '2008-12-19' '2009-07-02' '2008-12-05' '2008-04-17'
 '2008-11-18' '2008-04-01' '2008-06-10' '2008-08-20' '2008-09-29'
 '2008-08-06' '2008-05-09' '2009-02-07' '2008-10-13' '2008-05-23'
 '2008-10-02' '2008-12-18' '2008-12-11' '2008-04-07' '2008-10-11'
 '2008-08-23' '2008-08-05' '2008-05-29' '2008-04-10' '2008-09-24'
 '2008-04-05' '2008-10-07' '2009-01-06' '2008-09-10' '2008-12-24'
 '2008-04-26' '2008-07-10' '2008-11-03' '2008-11-22' '2008-11-17'
 '2008-07-21' '2008-07-04' '2009-11-23' '2008-10-27' '2008-06-16'
```

```
'2008-06-09' '2008-06-23' '2008-12-14' '2008-10-15' '2008-08-01'
'2008-07-11' '2008-05-28' '2009-12-21' '2008-07-02' '2008-04-24'
'2008-07-16' '2008-07-24' '2008-09-08' '2008-10-01' '2008-08-22'
'2008-07-25' '2008-10-09' '2008-11-08' '2008-06-20' '2008-06-11'
'2008-05-16' '2008-09-12' '2009-01-02' '2008-08-18' '2008-09-16'
'2008-11-05' '2008-08-13' '2008-07-14' '2008-06-03' '2009-01-23'
'2008-09-11' '2008-05-27' '2008-05-06' '2008-05-02' '2009-03-17'
'2008-04-04' '2008-11-11' '2008-04-28' '2008-05-26' '2009-03-10'
'2008-09-27' '2008-12-03' '2008-05-03' '2008-08-12' '2009-01-05'
'2008-10-14' '2008-10-24' '2008-09-05' '2008-06-26' '2008-11-01'
'2008-12-10' '2008-11-15' '2008-05-24' '2008-12-15' '2009-01-08'
'2008-06-18' '2008-12-07' '2008-10-16' '2008-04-15' '2008-10-29'
'2008-06-19' '2008-11-14' '2008-05-22' '2008-12-12' '2008-11-07'
'2008-07-22' '2008-07-23' '2008-08-30' '2008-08-09' '2009-01-17'
'2008-11-25' '2008-05-14' '2008-05-01' '2008-06-05' '2008-09-02'
'2008-06-06' '2008-07-28' '2008-10-25' '2008-06-13' '2008-10-23'
'2008-09-18' '2008-07-05' '2008-04-22' '2008-08-19' '2008-07-17'
'2008-08-29' '2008-06-04' '2008-06-02' '2009-02-21' '2008-05-15'
'2009-08-10' '2008-10-17' '2008-04-16' '2008-08-15' '2008-09-25'
'2009-01-15' '2008-12-08' '2008-05-21' '2008-09-17' '2008-06-24'
'2008-07-07' '2008-05-31' '2008-07-03' '2008-10-30' '2008-09-23'
'2008-08-02' '2009-05-05' '2009-03-07' '2008-12-01' '2008-06-14'
'2008-04-29' '2008-11-10' '2008-07-29' '2008-05-13' '2008-10-18'
'2008-04-18' '2009-01-14' '2009-01-26' '2008-06-21' '2008-06-30'
'2009-01-24' '2008-08-07' '2009-02-10' '2008-05-17' '2008-05-10'
'2008-11-29' '2008-04-19' '2008-04-09' '2008-07-12' '2009-01-12'
'2008-09-03' '2009-02-09' '2009-03-13' '2008-09-26' '2008-09-15'
'2009-02-05' '2009-01-21' '2008-08-14' '2009-03-23' '2009-03-12'
'2008-04-12' '2009-09-19' '2009-06-16' '2009-01-19' '2008-11-30'
'2009-03-31' '2009-03-18' '2009-07-29' '2009-01-30' '2009-07-01'
'2009-02-28' '2009-02-04' '2009-04-20' '2009-01-07' '2009-01-10'
'2009-08-15' '2009-06-23' '2009-03-02' '2009-02-24' '2009-02-06'
'2009-10-10' '2009-03-28' '2009-02-11' '2009-11-20' '2009-05-23'
'2009-01-09' '2009-01-13' '2009-01-22' '2009-03-03' '2009-03-27'
'2008-07-26' '2009-05-11' '2009-01-20' '2009-05-30' '2009-10-24'
'2009-02-12' '2009-01-31' '2009-02-27' '2009-03-11' '2009-04-27'
'2009-03-14' '2009-12-17' '2009-01-28' '2009-02-18' '2009-05-26'
'2009-12-18' '2009-02-25' '2009-09-04' '2009-03-21' '2009-10-09'
'2009-07-20' '2009-10-22' '2009-08-17' '2009-03-05' '2009-05-18'
'2009-04-18' '2009-04-07' '2009-04-24' '2009-12-12' '2009-04-17'
'2009-04-28' '2009-10-17' '2009-12-06' '2009-06-22' '2009-04-01'
'2009-08-21' '2009-02-02' '2009-03-26' '2009-04-25' '2009-06-19'
'2009-06-06' '2009-04-03' '2009-11-03' '2009-04-16' '2009-02-16'
'2009-04-11' '2009-05-02' '2009-07-08' '2009-07-28' '2009-01-27'
'2009-08-04' '2009-09-08' '2009-10-26' '2009-11-21' '2009-06-17'
'2008-05-18' '2009-03-25' '2009-06-18' '2009-07-17' '2009-12-04'
'2009-09-28' '2009-02-20' '2009-05-15' '2009-02-14' '2009-03-16'
'2009-10-28' '2009-03-24' '2009-08-11' '2009-04-15' '2009-09-05'
```

```
 '2009-12-29' '2009-04-29' '2009-03-06' '2009-10-30' '2009-07-03'
 '2009-02-17' '2009-05-04' '2009-09-21' '2009-05-08' '2009-10-29'
 '2009-11-30' '2009-04-14' '2009-09-02' '2009-07-30' '2009-02-19'
 '2009-11-11' '2009-11-14' '2009-04-09' '2009-06-25' '2009-05-16'
 '2008-08-17' '2009-12-16' '2009-06-05' '2009-10-14' '2009-03-20'
 '2009-01-29' '2009-08-07' '2009-04-02' '2009-04-04' '2009-12-15'
 '2009-11-27' '2009-11-29' '2009-08-26' '2009-04-23' '2009-12-20'
 '2009-10-12' '2009-10-01' '2009-10-08' '2009-04-08' '2009-12-02'
 '2009-09-12' '2009-03-04' '2009-04-22' '2009-06-13' '2009-12-30'
 '2009-03-30' '2009-09-26']

weight min and max:
0 20076

remi min and max:
0 19

cancel min and max:
0 17

used min and max:
0 19

w0 min and max:
0 99

w1 min and max:
0 84

w2 min and max:
0 90

w3 min and max:
0 15

w4 min and max:
0 36

w5 min and max:
0 14

w6 min and max:
0 27

w7 min and max:
0 55

w8 min and max:
```

```
0 1

w9 min and max:
0 48

w10 unique values:
[ 0  1  4  2  3 50  5  7 10 17 12  6  8  9 38 27 13 20 15]

target90 unique values:
[0 1]
```

1.7 1.e.

```
[ ]: for col in file.columns[file.isnull().any()].tolist():
         print(col ,"    ", file[col].isnull().sum())
```

```
delivpostcode      31036
advertisingdatacode      25905
```

1.8 1.f.

```
[131]: file.corr()
```

```
[131]:                    customernumber  salutation     title    domain  newsletter  \
        customernumber          1.000000    0.001253  0.001822 -0.004371   -0.000181
        salutation              0.001253    1.000000  0.033064  0.115485   -0.059480
        title                   0.001822    0.033064  1.000000  0.011686   -0.004252
        domain                 -0.004371    0.115485  0.011686  1.000000   -0.026329
        newsletter             -0.000181   -0.059480 -0.004252 -0.026329    1.000000
        model                  -0.000049   -0.069440 -0.010413 -0.002483    0.056183
        paymenttype             0.004337    0.106040  0.032492  0.014988    0.001172
        deliverytype            0.002176   -0.053046 -0.007056 -0.003161    0.020485
        invoicepostcode         0.007331    0.011715  0.004702 -0.001469    0.000238
        voucher                -0.004133   -0.036099  0.007428 -0.028032    0.002350
        case                    0.008751    0.054662  0.025803  0.037712    0.031380
        numberitems             0.001103   -0.082457  0.003132 -0.013007    0.070124
        gift                    0.001854    0.007554 -0.000173  0.001797   -0.004980
        entry                  -0.004161   -0.089164 -0.012572 -0.015039    0.062586
        points                       NaN         NaN       NaN       NaN         NaN
        shippingcosts          -0.000619   -0.001578 -0.006258 -0.006895   -0.042458
        weight                 -0.000008   -0.073064  0.005074 -0.007890    0.055792
        remi                    0.010021   -0.009339  0.009960 -0.006736    0.003458
        cancel                 -0.003556    0.008864 -0.003533  0.004169   -0.005258
        used                    0.005534   -0.022389 -0.008253 -0.011834    0.002346
        w0                     -0.008916    0.007940  0.000251  0.007155    0.019816
```

```
w1                     0.005185   -0.036641  0.006470 -0.003498    0.032213
w2                     0.001725   -0.031698 -0.010820 -0.010177   -0.000269
w3                    -0.005051    0.018696  0.003987  0.011289    0.008138
w4                     0.008476    0.009198  0.016536  0.009472   -0.003494
w5                     0.002725    0.023606 -0.007854 -0.008717    0.013799
w6                    -0.003050    0.011877  0.005801  0.007259   -0.001201
w7                     0.002238    0.005003 -0.001132  0.004525   -0.003705
w8                    -0.001200    0.006041 -0.001140 -0.003140    0.024071
w9                     0.005347   -0.004678  0.010506  0.003996    0.011323
w10                    0.009121   -0.014162 -0.008495  0.015541    0.002762
target90               0.001242   -0.028074 -0.001114  0.008615    0.083011

                    model  paymenttype  deliverytype  invoicepostcode  \
customernumber  -0.000049     0.004337      0.002176         0.007331   
salutation      -0.069440     0.106040     -0.053046         0.011715   
title           -0.010413     0.032492     -0.007056         0.004702   
domain          -0.002483     0.014988     -0.003161        -0.001469   
newsletter       0.056183     0.001172      0.020485         0.000238   
model            1.000000    -0.024386      0.357522        -0.017881   
paymenttype     -0.024386     1.000000     -0.000454         0.017520   
deliverytype     0.357522    -0.000454      1.000000        -0.025051   
invoicepostcode -0.017881     0.017520     -0.025051         1.000000   
voucher         -0.042948    -0.063817     -0.221198         0.011827   
case             0.106936    -0.002971      0.050234         0.005619   
numberitems      0.074731    -0.017015     -0.005476        -0.003444   
gift            -0.002627     0.017525     -0.034063        -0.012364   
entry            0.901104    -0.032496      0.318617        -0.030854   
points                NaN          NaN           NaN              NaN   
shippingcosts   -0.121870    -0.041964     -0.211830        -0.000931   
weight           0.128865    -0.097110      0.022725        -0.008013   
remi            -0.006554    -0.020976      0.022961         0.004620   
cancel          -0.057497     0.005062      0.002377         0.004860   
used            -0.113677    -0.045222     -0.073014         0.001377   
w0               0.086105    -0.057287      0.020425         0.009140   
w1               0.061857    -0.034977      0.010292         0.008815   
w2               0.034043    -0.028199      0.007220        -0.033952   
w3              -0.045133     0.064036     -0.037499         0.014360   
w4              -0.000218     0.095178      0.056076         0.000579   
w5              -0.192192     0.143398     -0.162077         0.016125   
w6               0.002683     0.082177      0.049499        -0.002245   
w7              -0.002247     0.042727      0.037001        -0.007027   
w8              -0.007910     0.020738     -0.001196         0.004233   
w9              -0.027977     0.009638      0.009322         0.016541   
w10             -0.009326     0.027184     -0.015386         0.007098   
target90         0.048831    -0.006011      0.061510         0.009634   

                  voucher  …         w2        w3        w4        w5  \
customernumber  -0.004133  …  0.001725 -0.005051  0.008476  0.002725
salutation      -0.036099  … -0.031698  0.018696  0.009198  0.023606
title            0.007428  … -0.010820  0.003987  0.016536 -0.007854
domain          -0.028032  … -0.010177  0.011289  0.009472 -0.008717
newsletter       0.002350  … -0.000269  0.008138 -0.003494  0.013799
model           -0.042948  …  0.034043 -0.045133 -0.000218 -0.192192
paymenttype     -0.063817  … -0.028199  0.064036  0.095178  0.143398
deliverytype    -0.221198  …  0.007220 -0.037499  0.056076 -0.162077
invoicepostcode  0.011827  … -0.033952  0.014360  0.000579  0.016125
voucher          1.000000  … -0.065736 -0.025517 -0.012543  0.126056
case            -0.382483  …  0.187622 -0.044408  0.069065 -0.220079
numberitems     -0.013321  …  0.254221  0.058759  0.111132 -0.033483
gift            -0.008666  … -0.010118 -0.005047  0.004256 -0.021816
entry            0.089605  …  0.021277 -0.049409 -0.000912 -0.217122
points                NaN  …       NaN       NaN       NaN       NaN
shippingcosts   -0.093200  … -0.047703 -0.031389 -0.030711 -0.135669
weight          -0.020410  …  0.240639 -0.065648  0.019155 -0.283716
remi            -0.031891  …  0.027767  0.001011  0.001376 -0.023162
cancel          -0.052566  … -0.028930 -0.014969 -0.009017 -0.064339
used            -0.044240  …  0.051283 -0.010819 -0.011078 -0.046762
w0              -0.026944  …  0.003538 -0.040639 -0.019600 -0.175614
w1               0.040801  … -0.034726 -0.021371 -0.015187 -0.092368
w2              -0.065736  …  1.000000 -0.015231 -0.018875 -0.065830
w3              -0.025517  … -0.015231  1.000000 -0.008072 -0.022501
w4              -0.012543  … -0.018875 -0.008072  1.000000 -0.034891
w5               0.126056  … -0.065830 -0.022501 -0.034891  1.000000
w6              -0.001013  … -0.017953 -0.006938  0.009103 -0.029986
w7              -0.002399  … -0.010741 -0.004291  0.027348 -0.018547
w8               0.000172  … -0.002780 -0.001014 -0.001473 -0.004383
w9              -0.034390  … -0.035607 -0.014698 -0.013206 -0.063529
w10             -0.030023  … -0.025303 -0.011341 -0.006356 -0.049018
target90        -0.029298  …  0.016079  0.018920 -0.007758  0.032107

                       w6        w7        w8        w9       w10  target90
customernumber  -0.003050  0.002238 -0.001200  0.005347  0.009121  0.001242
salutation       0.011877  0.005003  0.006041 -0.004678 -0.014162 -0.028074
title            0.005801 -0.001132 -0.001140  0.010506 -0.008495 -0.001114
domain           0.007259  0.004525 -0.003140  0.003996  0.015541  0.008615
newsletter      -0.001201 -0.003705  0.024071  0.011323  0.002762  0.083011
model            0.002683 -0.002247 -0.007910 -0.027977 -0.009326  0.048831
paymenttype      0.082177  0.042727  0.020738  0.009638  0.027184 -0.006011
deliverytype     0.049499  0.037001 -0.001196  0.009322 -0.015386  0.061510
invoicepostcode -0.002245 -0.007027  0.004233  0.016541  0.007098  0.009634
voucher         -0.001013 -0.002399  0.000172 -0.034390 -0.030023 -0.029298
case             0.052259  0.028637  0.021300  0.090156  0.084834  0.030245
numberitems      0.115549  0.039547 -0.008036  0.176943  0.049139  0.060062
gift            -0.004776 -0.002759 -0.000921 -0.001869 -0.005056 -0.004247
```

```
entry           -0.004227  0.002963 -0.006847 -0.022468 -0.013011  0.041292
points                NaN       NaN       NaN       NaN       NaN       NaN
shippingcosts    0.051956  0.009880 -0.005728 -0.020379 -0.012094 -0.070894
weight           0.019317 -0.007185 -0.007082  0.107474  0.020886  0.043502
remi             0.008656  0.001978 -0.002099  0.029402  0.017976  0.065579
cancel           0.018180 -0.002804 -0.002732  0.079804 -0.013266 -0.014917
used            -0.009823 -0.007223 -0.001974 -0.004770 -0.016439  0.029418
w0              -0.026626 -0.016679 -0.007416 -0.070057 -0.038499  0.016755
w1              -0.011589 -0.006871 -0.003900 -0.032253 -0.023562  0.033917
w2              -0.017953 -0.010741 -0.002780 -0.035607 -0.025303  0.016079
w3              -0.006938 -0.004291 -0.001014 -0.014698 -0.011341  0.018920
w4               0.009103  0.027348 -0.001473 -0.013206 -0.006356 -0.007758
w5              -0.029986 -0.018547 -0.004383 -0.063529 -0.049018  0.032107
w6               1.000000  0.006161 -0.001266 -0.006552 -0.004558  0.004522
w7               0.006161  1.000000 -0.000783 -0.008507 -0.004735 -0.007082
w8              -0.001266 -0.000783  1.000000 -0.002682 -0.002070 -0.000696
w9              -0.006552 -0.008507 -0.002682  1.000000 -0.022151  0.019271
w10             -0.004558 -0.004735 -0.002070 -0.022151  1.000000 -0.014007
target90         0.004522 -0.007082 -0.000696  0.019271 -0.014007  1.000000

[32 rows x 32 columns]
```

1.9 1.g.

```
[ ]: for i in range(22,36):
         print(file[file.columns[i]].describe(), end="\n\n")
```

```
count    32428.000000
mean       637.920809
std        724.358131
min          0.000000
25%          3.000000
50%        494.000000
75%        920.000000
max      20076.000000
Name: weight, dtype: float64

count    32428.000000
mean         0.059979
std          0.388740
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         19.000000
Name: remi, dtype: float64
```

```
count    32428.000000
mean         0.061613
std          0.306833
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         17.000000
Name: cancel, dtype: float64

count    32428.000000
mean         0.068860
std          0.474444
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         19.000000
Name: used, dtype: float64

count    32428.000000
mean         0.902122
std          1.654767
min          0.000000
25%          0.000000
50%          1.000000
75%          1.000000
max         99.000000
Name: w0, dtype: float64

count    32428.000000
mean         0.404342
std          1.410395
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         84.000000
Name: w1, dtype: float64

count    32428.000000
mean         0.276644
std          1.353981
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
```

```
max         90.000000
Name: w2, dtype: float64

count    32428.000000
mean         0.018903
std          0.253596
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         15.000000
Name: w3, dtype: float64

count    32428.000000
mean         0.047027
std          0.434265
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         36.000000
Name: w4, dtype: float64

count    32428.000000
mean         0.180986
std          0.561751
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         14.000000
Name: w5, dtype: float64

count    32428.000000
mean         0.027908
std          0.299862
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         27.000000
Name: w6, dtype: float64

count    32428.000000
mean         0.023128
std          0.401782
min          0.000000
25%          0.000000
```

```
50%          0.000000
75%          0.000000
max         55.000000
Name: w7, dtype: float64

count    32428.000000
mean         0.000185
std          0.013601
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max          1.000000
Name: w8, dtype: float64

count    32428.000000
mean         0.164981
std          0.836705
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         48.000000
Name: w9, dtype: float64
```

1.10 1.h.

```
[ ]: for i in range(1,15):
         print(file[file.columns[i]].value_counts(), end="\n\n")
```

```
2008-12-15    318
2008-12-17    289
2008-12-16    260
2008-12-18    238
2008-12-09    226
             …
2009-01-19      1
2009-03-04      1
2009-01-12      1
2009-03-26      1
2009-02-14      1
Name: date, Length: 351, dtype: int64

0    17840
1    11614
2     2974
```

```
Name: salutation, dtype: int64

0    32202
1      226
Name: title, dtype: int64

12    7734
9     6953
4     6627
8     3694
11    1422
5     1311
2     1196
0     1173
1     1139
6      548
3      381
10     137
7      113
Name: domain, dtype: int64

2008-12-15    315
2008-12-17    288
2008-12-16    260
2008-12-18    242
2008-12-02    227
             …
2008-07-26     62
2008-10-11     55
2008-10-04     54
2008-05-03     54
2008-05-10     50
Name: datecreated, Length: 275, dtype: int64

0    26932
1     5496
Name: newsletter, dtype: int64

1    18808
3     7358
2     6262
Name: model, dtype: int64

0    15063
1     6549
2     6537
3     4279
Name: paymenttype, dtype: int64
```

```
0    25879
1     6549
Name: deliverytype, dtype: int64

44    1244
50    1045
45    1035
52     917
41     815
      …
7      104
2       56
98      45
11       4
0        1
Name: invoicepostcode, Length: 97, dtype: int64

22.0    28
45      26
41.0    26
44      25
50.0    25
        ..
11.0     1
EN       1
39.0     1
57       1
Nl       1
Name: delivpostcode, Length: 192, dtype: int64

0    27174
1     5254
Name: voucher, dtype: int64

BQ    2631
AB     758
CA     552
BD     478
AR     448
AX     423
AQ     195
BR     178
AP     149
BL     106
BO     102
AH      85
BT      70
```

```
AE      59
BZ      44
BF      31
BC      30
AZ      24
AV      23
BI      20
AT      18
BM      18
BY      14
AI      11
AW       8
BA       6
AK       6
AO       4
BX       4
BB       4
AF       4
BS       3
BG       3
AL       2
AU       2
BK       2
AY       1
AG       1
AM       1
BV       1
AC       1
AD       1
BE       1
AS       1
Name: advertisingdatacode, dtype: int64

4    8648
3    7125
1    6349
2    6230
5    4076
Name: case, dtype: int64
```

1.11 1.i.

i. we can ignore those with the missing data, fill it manually or automaticaly with the avarage or with the value of some that have the same conditions.

ii. equal or close to equal distribution will help us the most. we can try to use algorithms to generate synthtic samples like Bayes and SMOTE.

iii. To evaluate the model, we can look at the confusion matrix and with it calculate other indicators for evaluating the model, such as recall, precision, f-score. We will choose to use a particular indicator based on the purpose of the learning. For example, if the model is dealing with human life we would like to choose an indicator that knows how to classify more observations from that class correctly.

1.12 Task 2

- Normalization and Standardization are created to achieve a similar target. they are required when we are dealing with attributes on a different scale.

Normalization uses to scale the data of an attribute to match a smaller range, such as -1 to 1 or 0 to 1 .

Standardization uses to scale the data of an attribute to match a Gaussian distribution. where =0 and =1, also called z scores.

The main differences between normalization and standardization is that normalization is useful when we don't know about the distribution when standardization is useful when the feature distribution is Normal or Gaussian. also normalization is affected by outliers where standardization is affected much less. Normalization Scales values are bounded and standardization is not bounded.

- **Min-max normalization** helps us to normalize data and understand the data more easily. It will scale the data between 0 and 1. 0 is the minimum value of the attribute and 1 is the maximum value of the attribute.

Z-score is used for standardizing scores on the same scale by dividing a score's deviation by the standard deviation in a data set. The result is a standard score.

Decimal Scaling in this technique, we move the decimal point of values of the attribute.This movement of decimal points totally depends on the maximum value among all values in the attribute.

[110]:

```python
# min-Max
def minMax(vector):
  vectorA=[]
  for i in range(len(vector)):
    vectorA.append((vector[i]-min(vector))/(max(vector)-min(vector)))
  return vectorA
```

[111]:

```python
# z-score
import numpy as np
def zScore(vector):
  vectorA=[]
  for i in range(len(vector)):
    vectorA.append((vector[i]-(sum(vector)/len(vector)))/np.std(vector))
  return vectorA
```

[112]:

```python
# Decimal Scaling
def decimalScaling(vector):
  vectorA=[]
  count =0
  vectorMax = max(vector)
  while (vectorMax >= 1):
       vectorMax=vectorMax/10
       count=count+1
       if vectorMax<1:
         count= pow(10,count)
  for i in range(len(vector)):
    vectorA.append((vector[i]/count))
  return vectorA
```

[113]:

```
vector=[1,0.2,0.3,0.22]
print(minMax(vector))
print(zScore(vector))
print(decimalScaling(vector))
```

```
[1.0, 0.0, 0.12499999999999997, 0.024999999999999988]
[1.7209630341216302, -0.6944236804350437, -0.39250034111545945,
-0.6340390125711268]
[0.1, 0.02, 0.03, 0.022]
```

[114]:

```
# min-Max using libraries
from sklearn import preprocessing
vector=[1,0.2,0.3,0.22]
scaler = preprocessing.minmax_scale(vector)
print(scaler)
```

```
[1.    0.    0.125 0.025]
```

[115]:

```
# z-score using libraries
import scipy.stats as stats
vector=[1,0.2,0.3,0.22]
print(stats.zscore(vector))
```

```
[ 1.72096303 -0.69442368 -0.39250034 -0.63403901]
```

1.13 Task 3

1. dividing continuous attribute into ranges (like for age split to adult, kid and so on) and replace the new named ranges with the original values. convert continous data into discrete data
2. dividing the values to several ranges that every group will have almost the same number of values.

[ ]:

```python
arr = [5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]
a = len(arr)
n = int(a / 3)
for i in range(0, 3):
    arr1 = []
    for j in range(i * n, (i + 1) * n):
        if j >= a:
            break
        arr1 = arr1 + [arr[j]]
    print(arr1)
```

```
[5, 10, 11, 13]
[15, 35, 50, 55]
[72, 92, 204, 215]
```

3. The algorithm divides the data into k groops of equal size. The width of intervals is: w = (max-min)/k

[ ]:

```python
a = len(arr)
w = int((max(arr) - min(arr)) / 3)
min1 = min(arr)
arr1 = []
for i in range(0, 3 + 1):
    arr1 = arr1 + [min1 + w * i]
arr2=[]

for i in range(0, 3):
    temp = []
    for j in arr:
        if j >= arr1[i] and j <= arr1[i+1]:
            temp += [j]
    arr2 += [temp]
print(arr2)
```

```
[[5, 10, 11, 13, 15, 35, 50, 55, 72], [92], [204, 215]]
```

4.

[ ]:

```python
print("Equal-frequency", end=" - ")
print(pd.qcut(arr, q=3, precision=0, labels=False))
print("Equal-width", end=" - ")
print(pd.cut(arr, bins=3,precision=0, labels=False))
```

```
Equal-frequency - [0 0 0 0 1 1 1 1 2 2 2 2]
Equal-width - [0 0 0 0 0 0 0 0 0 1 2 2]
```

1.14 Task 4

2 4.a.

Data smoothing is used to remove noise from data set which help to reveal patterns easier.

3 4.b.

Moving avarage is a type of data smoothing like we explain above. Moving averages are mainly used in time series analysis and time series forecasting.

uses a window size that defines the number of raw observations used to calculate the moving average value. The moving part is used to calculate the average values in the new series. With the help of the new series, the values in the data are comprised and therefore the data gets smoother.

4 4.c

i. Explained above

[4]:

```
# moving avarge
def simpleMovingAverage(n,vector):
  i=0
  moving_averages = []
  while i < len(vector) - n + 1:
    window = vector[i : i + n]
    window_average = (sum(window) / n)
    moving_averages.append(window_average)
    i += 1
  print(moving_averages)
```

[5]:

```
vector = [1,2,3,4,5,6]
simpleMovingAverage(3,vector)
```

```
[2.0, 3.0, 4.0, 5.0]
```

ii. Weighted Moving Average- is a technical indicator that gives a bigger weighting to the recent data points, to the less recent data gives a smaller weighting to the data points. The method works by multiplying by weight of each of the observations according to their position in the data.

[77]:

```
# Weighted Moving Average
def weightedMovingAverage(vector,vectorWeights, n):
  i=0
  weighted = []
  sumi=0
  while i < (len(vector) - n + 1):
    window = vector[i : i + n]
    for j in range(len(vectorWeights)):
      sumi = sumi + window[j]*vectorWeights[j]
    weighted.append(sumi/sum(vectorWeights))
    sumi=0
    i = i + 1
  print(weighted)
```

[78]:

```
vector = [2,4,6,8,9]
vectorWeights = [0.1,0.15,0.2]
weightedMovingAverage(vector,vectorWeights,3)
```

```
[4.444444444444445, 6.444444444444444, 8.0]
```

iii. Exponential Moving Average- is a technique that similar to the weighted moving average technique but more effective when used in order to get predictions quickly.frequently used in the production and inventory environment, where only the next period's value is required to be forecast.can be used if the data is stationary, that is there is a clear trend present.

[86]:

```
def ExponentialMovingAverage(vector,number):
  i=1
  weighted = []
  alfa = 2 / float(1 + number)
  f_prev = vector[0]
  f_curr = 0
  for i in range(len(vector)):
    f_curr = vector[i]*alfa + f_prev*(1-alfa)
    weighted.append(f_curr)
    f_prev = f_curr
  print(weighted)
```

[87]:

```
vector = [2,4,6,8,12,14,16,18,20]
ExponentialMovingAverage(vector,2)
```

```
[2.0, 3.3333333333333333, 5.111111111111111, 7.037037037037036,
10.345679012345679, 12.781893004115226, 14.92729766803841, 16.975765889346135,
18.991921963115377]
```

5 4.d.

Binning Methods for Data Smoothing - is another method to handle noisy data, first the values are sorted and then divided into several bins. Then there are some approaches:

i. Smoothing by bin means : in this smoothing method each value in the bin is replaced by the average value of the whole bin

```
[101]: def binByMean(vector,sizeOfBin):
           sizeOfVec=len(vector)
           i=0
           binned = []
           while (i<sizeOfVec-sizeOfBin+1):
               sum = 0
               for j in range (i,i+sizeOfBin):
                   sum+=vector[j]
               mean = sum / sizeOfBin
               for j in range (i,i+sizeOfBin):
                   binned.append(mean)
               i+=sizeOfBin

           if(sizeOfVec%sizeOfBin!=0):
               sum=0
               for j in range (i,sizeOfVec):
                   sum+=vector[j]
               mean=sum/(sizeOfVec-i)
               for j in range(i, sizeOfVec):
                   binned.append(mean)
           return binned
```

```
[102]: data=[8,16, 9, 15, 21, 21, 24, 30, 26, 27, 30, 34]
       binByMean(data, 4)
```

```
[102]: [12.0, 12.0, 12.0, 12.0, 24.0, 24.0, 24.0, 24.0, 29.25, 29.25, 29.25, 29.25]
```

ii. Smoothing by bin boundaries: After division, the minimum and maximum values in each bin are identified as limits and each value is replaced by the value of the nearest limit.

```
[103]: def binByBoundry(vector,sizeOfBin):
           sizeOfVec=len(vector)
           i=0
           binned = []
           while (i<sizeOfVec-sizeOfBin+1):
               sum = 0
               for j in range (i,i+sizeOfBin):
                 maxVal = max(vector[i:i+sizeOfBin])
                 minVal = min(vector[i:i+sizeOfBin])
                 if(vector[j]-minVal >= maxVal-vector[j]):
                   binned.append(maxVal)
                 else:
                   binned.append(minVal)
               i+=sizeOfBin

           if(sizeOfVec%sizeOfBin!=0):
               sum=0
```

```
        for j in range (i,sizeOfVec):
          maxVal = max(vector[i:sizeOfVec])
          minVal = min(vector[i:sizeOfVec])
          if(vector[j]-minVal >= maxVal-vector[j]):
            binned.append(maxVal)
          else:
            binned.append(minVal)
    return binned
```

[104]:

```
data=[8,16, 9, 15, 21, 21, 24, 30, 26, 27, 30, 34]
binByBoundry(data, 4)
```

[104]:

```
[8, 16, 8, 16, 21, 21, 21, 30, 26, 26, 34, 34]
```

6 4.e

[117]:

```
# simple moving averages using pandas
import pandas as pd

def SMA(vector, window_size):
  numbers_series = pd.Series(vector)
  windows = numbers_series.rolling(window_size)
  moving_averages = windows.mean()
  moving_averages_list = moving_averages.tolist()
  final_list = moving_averages_list[window_size - 1:]
  print(final_list)
```

[119]:

```
vector = [1,2,3,4,5,6]
SMA(vector, 3)
```

```
[2.0, 3.0, 4.0, 5.0]
```

[126]:

```
# exponential moving averages with pandas
import pandas as pd

def EMA(vector):
  numbers_series = pd.Series(vector)
  moving_averages = round(numbers_series.ewm(alpha=0.5, adjust=False).mean(), 2)
  moving_averages_list = moving_averages.tolist()
  print(moving_averages_list)
```

[127]:

```
vector = [2,4,6,8,12,14,16,18,20]
EMA(vector)
```

```
[2.0, 3.0, 4.5, 6.25, 9.12, 11.56, 13.78, 15.89, 17.95]
```

We couldn't find libraries that knows how to preform smooting by Weighted Moving Average and Smoothing by bin means/ boundaries.