In [1]:
```python
##Setting directory
import os
os.chdir('C:/Users/TOTAGOUSER4/Documents/Totago Technologies/Data Science/Projects/BCG')
```

Datasets

In [2]:
```python
import pandas as pd
cust_data = pd.read_csv('ml_case_training_data.csv')
price_data = pd.read_csv('ml_case_training_hist_data.csv')
churn_data = pd.read_csv('ml_case_training_output.csv')
```

In [3]:
```python
#Customer Dataset
cust_data.head(10)
```

Out[3]:

	id	activity_new	campaign_disc_ele	channel_sales	cons_12m
0	48ada52261e7cf58715202705a0451c9	esoiiifxdlbkcsluxmfuacbdckommixw	NaN	lmkebamcaaclubfxadlmueccxoimlema	309275
1	24011ae4ebbe3035111d65fa7c15bc57	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	0
2	d29c2c54acc38ff3c0614d0a653813dd	NaN	NaN	NaN	4660
3	764c75f661154dac3a6c254cd082ea7d	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	544
4	bba03439a292a1e166f80264c16191cb	NaN	NaN	lmkebamcaaclubfxadlmueccxoimlema	1584
5	568bb38a1afd7c0fc49c77b3789b59a3	sfisfxfcocfpcmckuekokxuseixdaoeu	NaN	foosdfpfkusacimwkcsosbicdxkicaua	121335
6	149d57cf92fc41cf94415803a877cb4b	NaN	NaN	NaN	4425
7	1aa498825382410b098937d65c4ec26d	NaN	NaN	usilxuppasemubllopkaafesmlibmsdf	8302
8	7ab4bf4878d8f7661dfc20e9b8e18011	sscfoipxikopfskekuobeuxkxmwsuucb	NaN	foosdfpfkusacimwkcsosbicdxkicaua	45097
9	01495c955be7ec5e7f3203406785aae0	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	29552

10 rows × 32 columns

In [4]:
```python
#Pricing Data
price_data.head(10)
```

Out[4]:

	id	price_date	price_p1_var	price_p2_var	price_p3_var	price_p1_fix	price_p2_fix	price_p3_fix
0	038af19179925da21a25619c5a24b745	2015-01-01	0.151367	0.0	0.0	44.266931	0.0	0.0
1	038af19179925da21a25619c5a24b745	2015-02-01	0.151367	0.0	0.0	44.266931	0.0	0.0
2	038af19179925da21a25619c5a24b745	2015-03-01	0.151367	0.0	0.0	44.266931	0.0	0.0
3	038af19179925da21a25619c5a24b745	2015-04-01	0.149626	0.0	0.0	44.266931	0.0	0.0
4	038af19179925da21a25619c5a24b745	2015-05-01	0.149626	0.0	0.0	44.266931	0.0	0.0
5	038af19179925da21a25619c5a24b745	2015-06-01	0.149626	0.0	0.0	44.266930	0.0	0.0
6	038af19179925da21a25619c5a24b745	2015-07-01	0.150321	0.0	0.0	44.444710	0.0	0.0
7	038af19179925da21a25619c5a24b745	2015-08-01	0.145859	0.0	0.0	44.444710	0.0	0.0
8	038af19179925da21a25619c5a24b745	2015-09-01	0.145859	0.0	0.0	44.444710	0.0	0.0
9	038af19179925da21a25619c5a24b745	2015-10-01	0.145859	0.0	0.0	44.444710	0.0	0.0

In [5]:
```python
#Churn data
churn_data.head(10)
```

Out[5]:

	id	churn
0	48ada52261e7cf58715202705a0451c9	0
1	24011ae4ebbe3035111d65fa7c15bc57	1
2	d29c2c54acc38ff3c0614d0a653813dd	0
3	764c75f661154dac3a6c254cd082ea7d	0
4	bba03439a292a1e166f80264c16191cb	0
5	568bb38a1afd7c0fc49c77b3789b59a3	0
6	149d57cf92fc41cf94415803a877cb4b	0
7	1aa498825382410b098937d65c4ec26d	1
8	7ab4bf4878d8f7661dfc20e9b8e18011	1
9	01495c955be7ec5e7f3203406785aae0	0

Data Cleaning

Dealing with missing data

In [6]:
```python
#Checking for null values in Customer Details Dataset
cust_data.isnull().sum()
```

Out[6]:
```
id                              0
activity_new                 9545
campaign_disc_ele           16096
channel_sales                4218
cons_12m                        0
cons_gas_12m                    0
cons_last_month                 0
date_activ                      0
date_end                        2
date_first_activ            12588
date_modif_prod               157
date_renewal                   40
forecast_base_bill_ele      12588
forecast_base_bill_year     12588
forecast_bill_12m           12588
forecast_cons               12588
forecast_cons_12m               0
forecast_cons_year              0
forecast_discount_energy      126
forecast_meter_rent_12m         0
forecast_price_energy_p1      126
forecast_price_energy_p2      126
forecast_price_pow_p1         126
has_gas                         0
imp_cons                        0
margin_gross_pow_ele           13
margin_net_pow_ele             13
nb_prod_act                     0
net_margin                     15
num_years_antig                 0
origin_up                      87
pow_max                         3
dtype: int64
```

Columns with null values - activity_new, campaign_disc_ele, channel_sales, date_end, date_first_activ, date_modif_prod, date_renewal, forecast_base_bill_ele, forecast_base_bill_year, forecast_bill_12m, forecast_cons, forecast_discount_energy, forecast_price_energy_p1, forecast_price_energy_p2, forecast_price_pow_p1, margin_gross_pow_ele, margin_net_pow_ele, net_margin, origin_up, pow_max

In [7]:
```python
#Checking for null values in Price Dataset
price_data.isnull().sum()
```

Out[7]:
```
id                 0
price_date         0
price_p1_var    1359
price_p2_var    1359
price_p3_var    1359
price_p1_fix    1359
price_p2_fix    1359
price_p3_fix    1359
dtype: int64
```

Columns with missing data - price_p1_var, price_p2_var, price_p3_var, price_p1_fix, price_p2_fix, price_p3_fix

In [8]:
```python
#Checking for null values in Churn Dataset
churn_data.isnull().sum()
```

Out[8]:
```
id       0
churn    0
dtype: int64
```

No nulls here

In [9]:
```python
#Converting the dates to datetime
import datetime
for a in ['date_activ', 'date_end', 'date_first_activ', 'date_modif_prod', 'date_renewal']:
    cust_data[a] = pd.to_datetime(cust_data[a], format='%Y %m %d ')

cust_data.head(10)
```

Out[9]:

	id	activity_new	campaign_disc_ele	channel_sales	cons_12m
0	48ada52261e7cf58715202705a0451c9	esoiiifxdlbkcsluxmfuacbdckommixw	NaN	lmkebamcaaclubfxadlmueccxoimlema	309275
1	24011ae4ebbe3035111d65fa7c15bc57	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	0
2	d29c2c54acc38ff3c0614d0a653813dd	NaN	NaN	NaN	4660
3	764c75f661154dac3a6c254cd082ea7d	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	544
4	bba03439a292a1e166f80264c16191cb	NaN	NaN	lmkebamcaaclubfxadlmueccxoimlema	1584
5	568bb38a1afd7c0fc49c77b3789b59a3	sfisfxfcocfpcmckuekokxuseixdaoeu	NaN	foosdfpfkusacimwkcsosbicdxkicaua	121335
6	149d57cf92fc41cf94415803a877cb4b	NaN	NaN	NaN	4425
7	1aa498825382410b098937d65c4ec26d	NaN	NaN	usilxuppasemubllopkaafesmlibmsdf	8302
8	7ab4bf4878d8f7661dfc20e9b8e18011	sscfoipxikopfskekuobeuxkxmwsuucb	NaN	foosdfpfkusacimwkcsosbicdxkicaua	45097
9	01495c955be7ec5e7f3203406785aae0	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	29552

10 rows × 32 columns

In [10]:
```python
#Checking the data type
cust_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16096 entries, 0 to 16095
Data columns (total 32 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   id                        16096 non-null  object
 1   activity_new              6551 non-null   object
 2   campaign_disc_ele         0 non-null      float64
 3   channel_sales             11878 non-null  object
 4   cons_12m                  16096 non-null  int64
 5   cons_gas_12m              16096 non-null  int64
 6   cons_last_month           16096 non-null  int64
 7   date_activ                16096 non-null  datetime64[ns]
 8   date_end                  16094 non-null  datetime64[ns]
 9   date_first_activ          3508 non-null   datetime64[ns]
 10  date_modif_prod           15939 non-null  datetime64[ns]
 11  date_renewal              16056 non-null  datetime64[ns]
 12  forecast_base_bill_ele    3508 non-null   float64
 13  forecast_base_bill_year   3508 non-null   float64
 14  forecast_bill_12m         3508 non-null   float64
 15  forecast_cons             3508 non-null   float64
 16  forecast_cons_12m         16096 non-null  float64
 17  forecast_cons_year        16096 non-null  int64
 18  forecast_discount_energy  15970 non-null  float64
 19  forecast_meter_rent_12m   16096 non-null  float64
 20  forecast_price_energy_p1  15970 non-null  float64
 21  forecast_price_energy_p2  15970 non-null  float64
 22  forecast_price_pow_p1     15970 non-null  float64
 23  has_gas                   16096 non-null  object
 24  imp_cons                  16096 non-null  float64
 25  margin_gross_pow_ele      16083 non-null  float64
 26  margin_net_pow_ele        16083 non-null  float64
 27  nb_prod_act               16096 non-null  int64
 28  net_margin                16081 non-null  float64
 29  num_years_antig           16096 non-null  int64
 30  origin_up                 16009 non-null  object
 31  pow_max                   16093 non-null  float64
dtypes: datetime64[ns](5), float64(16), int64(6), object(5)
memory usage: 3.9+ MB
```

In [11]:
```python
# Filling up nulls in Customer Details Dataset

#Filling with zeros
for col in ['forecast_base_bill_ele', 'forecast_base_bill_year', 'forecast_bill_12m', 'forecast_cons',
'forecast_discount_energy', 'forecast_price_energy_p1', 'forecast_price_energy_p2', 'forecast_price_pow
_p1', 'margin_gross_pow_ele', 'margin_net_pow_ele', 'net_margin', 'pow_max']:
    cust_data[col].fillna(0, inplace = True)

#Filling with null keyword
for col in ['channel_sales', 'origin_up']:
    cust_data[col].fillna('Null', inplace = True)

#Filling Dates with activation date
for col in ['date_end', 'date_first_activ', 'date_modif_prod', 'date_renewal']:
    cust_data[col].fillna(cust_data['date_activ'], inplace = True)

#Dropping Some columns
cust_data.drop(columns=['activity_new', 'campaign_disc_ele', ], inplace=True)

cust_data.head(10)
```

Out[11]:

	id	channel_sales	cons_12m	cons_gas_12m	cons_last_month	date_activ	date_e
0	48ada52261e7cf58715202705a0451c9	lmkebamcaaclubfxadlmueccxoimlema	309275	0	10025	2012-11-07	2016-1
1	24011ae4ebbe3035111d65fa7c15bc57	foosdfpfkusacimwkcsosbicdxkicaua	0	54946	0	2013-06-15	2016-0
2	d29c2c54acc38ff3c0614d0a653813dd	Null	4660	0	0	2009-08-21	2016-0
3	764c75f661154dac3a6c254cd082ea7d	foosdfpfkusacimwkcsosbicdxkicaua	544	0	0	2010-04-16	2016-0
4	bba03439a292a1e166f80264c16191cb	lmkebamcaaclubfxadlmueccxoimlema	1584	0	0	2010-03-30	2016-0
5	568bb38a1afd7c0fc49c77b3789b59a3	foosdfpfkusacimwkcsosbicdxkicaua	121335	0	12400	2010-04-08	2016-0
6	149d57cf92fc41cf94415803a877cb4b	Null	4425	0	526	2010-01-13	2016-0
7	1aa498825382410b098937d65c4ec26d	usilxuppasemubllopkaafesmlibmsdf	8302	0	1998	2011-12-09	2016-1
8	7ab4bf4878d8f7661dfc20e9b8e18011	foosdfpfkusacimwkcsosbicdxkicaua	45097	0	0	2011-12-02	2016-1
9	01495c955be7ec5e7f3203406785aae0	foosdfpfkusacimwkcsosbicdxkicaua	29552	0	1260	2010-04-21	2016-0

10 rows × 30 columns

In [12]:
```python
# Filling up nulls in Price Dataset

for col in ['price_p1_var', 'price_p2_var', 'price_p3_var', 'price_p1_fix', 'price_p2_fix', 'price_p3_f
ix']:
    price_data[col].fillna(0, inplace = True)

price_data.head(10)
```

Out[12]:

	id	price_date	price_p1_var	price_p2_var	price_p3_var	price_p1_fix	price_p2_fix	price_p3_fix
0	038af19179925da21a25619c5a24b745	2015-01-01	0.151367	0.0	0.0	44.266931	0.0	0.0
1	038af19179925da21a25619c5a24b745	2015-02-01	0.151367	0.0	0.0	44.266931	0.0	0.0
2	038af19179925da21a25619c5a24b745	2015-03-01	0.151367	0.0	0.0	44.266931	0.0	0.0
3	038af19179925da21a25619c5a24b745	2015-04-01	0.149626	0.0	0.0	44.266931	0.0	0.0
4	038af19179925da21a25619c5a24b745	2015-05-01	0.149626	0.0	0.0	44.266931	0.0	0.0
5	038af19179925da21a25619c5a24b745	2015-06-01	0.149626	0.0	0.0	44.266930	0.0	0.0
6	038af19179925da21a25619c5a24b745	2015-07-01	0.150321	0.0	0.0	44.444710	0.0	0.0
7	038af19179925da21a25619c5a24b745	2015-08-01	0.145859	0.0	0.0	44.444710	0.0	0.0
8	038af19179925da21a25619c5a24b745	2015-09-01	0.145859	0.0	0.0	44.444710	0.0	0.0
9	038af19179925da21a25619c5a24b745	2015-10-01	0.145859	0.0	0.0	44.444710	0.0	0.0

In [13]:
```python
#Cross checking nulls and checking variable types for Customer Details Dataset
cust_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16096 entries, 0 to 16095
Data columns (total 30 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   id                        16096 non-null  object
 1   channel_sales             16096 non-null  object
 2   cons_12m                  16096 non-null  int64
 3   cons_gas_12m              16096 non-null  int64
 4   cons_last_month           16096 non-null  int64
 5   date_activ                16096 non-null  datetime64[ns]
 6   date_end                  16096 non-null  datetime64[ns]
 7   date_first_activ          16096 non-null  datetime64[ns]
 8   date_modif_prod           16096 non-null  datetime64[ns]
 9   date_renewal              16096 non-null  datetime64[ns]
 10  forecast_base_bill_ele    16096 non-null  float64
 11  forecast_base_bill_year   16096 non-null  float64
 12  forecast_bill_12m         16096 non-null  float64
 13  forecast_cons             16096 non-null  float64
 14  forecast_cons_12m         16096 non-null  float64
 15  forecast_cons_year        16096 non-null  int64
 16  forecast_discount_energy  16096 non-null  float64
 17  forecast_meter_rent_12m   16096 non-null  float64
 18  forecast_price_energy_p1  16096 non-null  float64
 19  forecast_price_energy_p2  16096 non-null  float64
 20  forecast_price_pow_p1     16096 non-null  float64
 21  has_gas                   16096 non-null  object
 22  imp_cons                  16096 non-null  float64
 23  margin_gross_pow_ele      16096 non-null  float64
 24  margin_net_pow_ele        16096 non-null  float64
 25  nb_prod_act               16096 non-null  int64
 26  net_margin                16096 non-null  float64
 27  num_years_antig           16096 non-null  int64
 28  origin_up                 16096 non-null  object
 29  pow_max                   16096 non-null  float64
dtypes: datetime64[ns](5), float64(15), int64(6), object(4)
memory usage: 3.7+ MB
```

In [14]:
```python
#Cross checking nulls and checking variable types for price Dataset
price_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 193002 entries, 0 to 193001
Data columns (total 8 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   id            193002 non-null  object
 1   price_date    193002 non-null  object
 2   price_p1_var  193002 non-null  float64
 3   price_p2_var  193002 non-null  float64
 4   price_p3_var  193002 non-null  float64
 5   price_p1_fix  193002 non-null  float64
 6   price_p2_fix  193002 non-null  float64
 7   price_p3_fix  193002 non-null  float64
dtypes: float64(6), object(2)
memory usage: 11.8+ MB
```

In [ ]:

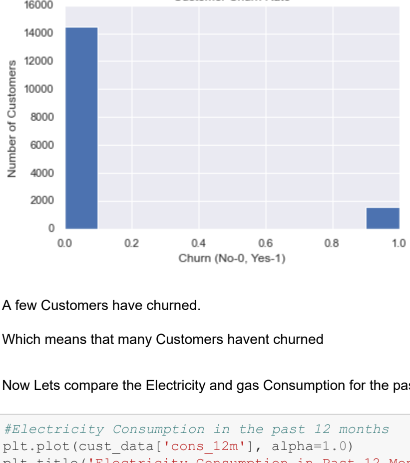
Data Analysis

In [15]:
```python
#For better visualization import matplotlib and seaborn

import matplotlib.pyplot as plt
plt.style.use('classic')
%matplotlib inline
import numpy as np
import seaborn as sns
sns.set()
```

Looking at the rate of those who are also gas clients.

In [16]:
```python
#Histogram showing number of customers that have gas

plt.hist(cust_data['has_gas'], alpha=1.0)
plt.title('Customers that are Gas Clients')
plt.ylabel('Number of Customers', fontdict = None, labelpad = None)
plt.xlabel('True/False', fontdict = None, labelpad = None)
plt.plot()
```

Out[16]: []

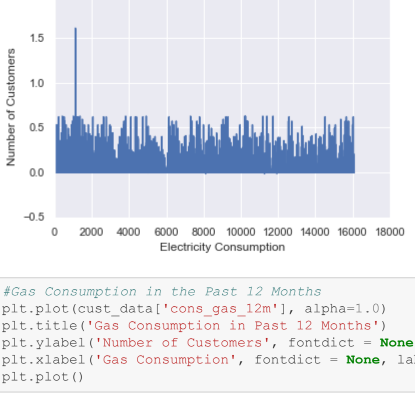


From the histogram above, we see that a few of the customers are also gas clients.

Now lets compare that to those that churn.

In [17]:
```python
plt.hist(churn_data['churn'], alpha=1.0)
plt.title('Customer Churn Rate')
plt.ylabel('Number of Customers', fontdict = None, labelpad = None)
plt.xlabel('Churn (No-0, Yes-1)', fontdict = None, labelpad = None)
plt.plot()
```

Out[17]: []



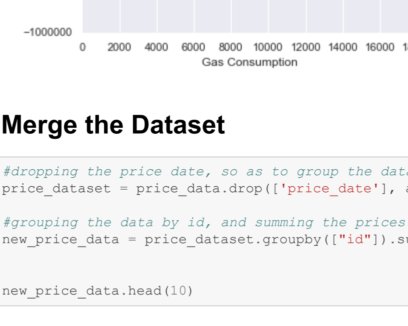
A few Customers have churned.

Which means that many Customers havent churned

Now Lets compare the Electricity and gas Consumption for the past 12 months.

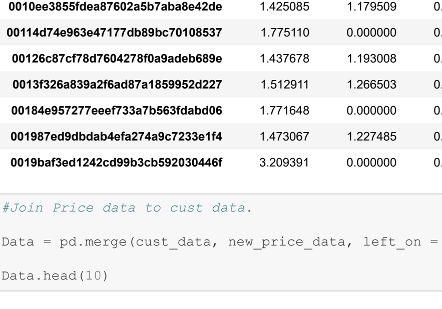
In [18]:
```python
#Electricity Consumption in the past 12 months
plt.plot(cust_data['cons_12m'], alpha=1.0)
plt.title('Electricity Consumption in Past 12 Months')
plt.ylabel('Number of Customers', fontdict = None, labelpad = None)
plt.xlabel('Electricity Consumption', fontdict = None, labelpad = None)
plt.plot()
```

Out[18]: []



In [19]:
```python
#Gas Consumption in the Past 12 Months
plt.plot(cust_data['cons_gas_12m'], alpha=1.0)
plt.title('Gas Consumption in Past 12 Months')
plt.ylabel('Number of Customers', fontdict = None, labelpad = None)
plt.xlabel('Gas Consumption', fontdict = None, labelpad = None)
plt.plot()
```

Out[19]: []



Merge the Dataset

In [20]:
```python
#dropping the price date, so as to group the data by id
price_dataset = price_data.drop(['price_date'], axis = 1)

#grouping the data by id, and summing the prices
new_price_data = price_dataset.groupby(["id"]).sum()


new_price_data.head(10)
```

Out[20]:

id	price_p1_var	price_p2_var	price_p3_var	price_p1_fix	price_p2_fix	price_p3_fix
0002203ffbb812588b632b9e628cc38d	1.492061	1.245525	0.877924	488.420789	293.052462	195.368327
0004351ebdd665e6ee664792efc4fd13	1.757118	0.000000	0.000000	532.625404	0.000000	0.000000
0010bcc39e42b3c2131ed2ce55246e3c	2.178702	0.000000	0.000000	543.836520	0.000000	0.000000
0010ee3855fdea87602a5b7aba8e42de	1.425085	1.179509	0.828384	487.769130	292.661462	195.107656
00114d74e963e47177db89bc70108537	1.775110	0.000000	0.000000	531.203164	0.000000	0.000000
00126c87cf78d7604278f0a9adeb689e	1.437678	1.193008	0.843651	487.932042	292.759214	195.172828
0013f326a839a2f6ad87a1859952d227	1.512911	1.266503	0.899048	488.746620	293.247960	195.498660
00184e957277eeef733a7b563fdabd06	1.771648	0.000000	0.000000	531.203164	0.000000	0.000000
001987ed9dbdab4efa274a9c7233e1f4	1.473067	1.227485	0.876366	487.769126	292.661464	195.107662
0019baf3ed1242cd99b3cb592030446f	3.209391	0.000000	0.000000	695.543164	0.000000	0.000000

In [21]:
```python
#Join Price data to cust data.

Data = pd.merge(cust_data, new_price_data, left_on = 'id', right_on = 'id', indicator = True)

Data.head(10)
```

Out[21]:

	id	channel_sales	cons_12m	cons_gas_12m	cons_last_month	date_activ	date_e
0	48ada52261e7cf58715202705a0451c9	lmkebamcaaclubfxadlmueccxoimlema	309275	0	10025	2012-11-07	2016-1
1	24011ae4ebbe3035111d65fa7c15bc57	foosdfpfkusacimwkcsosbicdxkicaua	0	54946	0	2013-06-15	2016-0
2	d29c2c54acc38ff3c0614d0a653813dd	Null	4660	0	0	2009-08-21	2016-0
3	764c75f661154dac3a6c254cd082ea7d	foosdfpfkusacimwkcsosbicdxkicaua	544	0	0	2010-04-16	2016-0
4	bba03439a292a1e166f80264c16191cb	lmkebamcaaclubfxadlmueccxoimlema	1584	0	0	2010-03-30	2016-0
5	568bb38a1afd7c0fc49c77b3789b59a3	foosdfpfkusacimwkcsosbicdxkicaua	121335	0	12400	2010-04-08	2016-0
6	149d57cf92fc41cf94415803a877cb4b	Null	4425	0	526	2010-01-13	2016-0
7	1aa498825382410b098937d65c4ec26d	usilxuppasemubllopkaafesmlibmsdf	8302	0	1998	2011-12-09	2016-1
8	7ab4bf4878d8f7661dfc20e9b8e18011	foosdfpfkusacimwkcsosbicdxkicaua	45097	0	0	2011-12-02	2016-1
9	01495c955be7ec5e7f3203406785aae0	foosdfpfkusacimwkcsosbicdxkicaua	29552	0	1260	2010-04-21	2016-0

10 rows × 37 columns

In [22]:
```python
#Join churn data.

Dataset = pd.merge(Data, churn_data, left_on = 'id', right_on = 'id')

Dataset.head(10)
```

Out[22]:

	id	channel_sales	cons_12m	cons_gas_12m	cons_last_month	date_activ	date_e
0	48ada52261e7cf58715202705a0451c9	lmkebamcaaclubfxadlmueccxoimlema	309275	0	10025	2012-11-07	2016-1
1	24011ae4ebbe3035111d65fa7c15bc57	foosdfpfkusacimwkcsosbicdxkicaua	0	54946	0	2013-06-15	2016-0
2	d29c2c54acc38ff3c0614d0a653813dd	Null	4660	0	0	2009-08-21	2016-0
3	764c75f661154dac3a6c254cd082ea7d	foosdfpfkusacimwkcsosbicdxkicaua	544	0	0	2010-04-16	2016-0
4	bba03439a292a1e166f80264c16191cb	lmkebamcaaclubfxadlmueccxoimlema	1584	0	0	2010-03-30	2016-0
5	568bb38a1afd7c0fc49c77b3789b59a3	foosdfpfkusacimwkcsosbicdxkicaua	121335	0	12400	2010-04-08	2016-0
6	149d57cf92fc41cf94415803a877cb4b	Null	4425	0	526	2010-01-13	2016-0
7	1aa498825382410b098937d65c4ec26d	usilxuppasemubllopkaafesmlibmsdf	8302	0	1998	2011-12-09	2016-1
8	7ab4bf4878d8f7661dfc20e9b8e18011	foosdfpfkusacimwkcsosbicdxkicaua	45097	0	0	2011-12-02	2016-1
9	01495c955be7ec5e7f3203406785aae0	foosdfpfkusacimwkcsosbicdxkicaua	29552	0	1260	2010-04-21	2016-0

10 rows × 38 columns

Feature Engineering

In [23]:
```python
#Checking for nulls
Dataset.isnull().sum()
```

Out[23]:
```
id                          0
channel_sales               0
cons_12m                    0
cons_gas_12m                0
cons_last_month             0
date_activ                  0
date_end                    0
date_first_activ            0
date_modif_prod             0
date_renewal                0
forecast_base_bill_ele      0
forecast_base_bill_year     0
forecast_bill_12m           0
forecast_cons               0
forecast_cons_12m           0
forecast_cons_year          0
forecast_discount_energy    0
forecast_meter_rent_12m     0
forecast_price_energy_p1    0
forecast_price_energy_p2    0
forecast_price_pow_p1       0
has_gas                     0
imp_cons                    0
margin_gross_pow_ele        0
margin_net_pow_ele          0
nb_prod_act                 0
net_margin                  0
num_years_antig             0
origin_up                   0
pow_max                     0
price_p1_var                0
price_p2_var                0
price_p3_var                0
price_p1_fix                0
price_p2_fix                0
price_p3_fix                0
_merge                      0
churn                       0
dtype: int64
```

no null values

In [24]:
```python
Dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16096 entries, 0 to 16095
Data columns (total 38 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   id                        16096 non-null  object
 1   channel_sales             16096 non-null  object
 2   cons_12m                  16096 non-null  int64
 3   cons_gas_12m              16096 non-null  int64
 4   cons_last_month           16096 non-null  int64
 5   date_activ                16096 non-null  datetime64[ns]
 6   date_end                  16096 non-null  datetime64[ns]
 7   date_first_activ          16096 non-null  datetime64[ns]
 8   date_modif_prod           16096 non-null  datetime64[ns]
 9   date_renewal              16096 non-null  datetime64[ns]
 10  forecast_base_bill_ele    16096 non-null  float64
 11  forecast_base_bill_year   16096 non-null  float64
 12  forecast_bill_12m         16096 non-null  float64
 13  forecast_cons             16096 non-null  float64
 14  forecast_cons_12m         16096 non-null  float64
 15  forecast_cons_year        16096 non-null  int64
 16  forecast_discount_energy  16096 non-null  float64
 17  forecast_meter_rent_12m   16096 non-null  float64
 18  forecast_price_energy_p1  16096 non-null  float64
 19  forecast_price_energy_p2  16096 non-null  float64
 20  forecast_price_pow_p1     16096 non-null  float64
 21  has_gas                   16096 non-null  object
 22  imp_cons                  16096 non-null  float64
 23  margin_gross_pow_ele      16096 non-null  float64
 24  margin_net_pow_ele        16096 non-null  float64
 25  nb_prod_act               16096 non-null  int64
 26  net_margin                16096 non-null  float64
 27  num_years_antig           16096 non-null  int64
 28  origin_up                 16096 non-null  object
 29  pow_max                   16096 non-null  float64
 30  price_p1_var              16096 non-null  float64
 31  price_p2_var              16096 non-null  float64
 32  price_p3_var              16096 non-null  float64
 33  price_p1_fix              16096 non-null  float64
 34  price_p2_fix              16096 non-null  float64
 35  price_p3_fix              16096 non-null  float64
 36  _merge                    16096 non-null  category
 37  churn                     16096 non-null  int64
dtypes: category(1), datetime64[ns](5), float64(21), int64(7), object(4)
memory usage: 4.7+ MB
```

Data types are fine

Now, Let's generate some features.

Features needed;

Features present are; channel_sales, cons_12m, cons_gas_12m, cons_last_month, forecast_cons_12m, forecast_cons_year, forecast_discount_energy, forecast_meter_rent_12m, forecast_price_energy_p1, forecast_price_energy_p2, forecast_price_pow_p1, has_gas, imp_cons, margin_gross_pow_ele, margin_net_pow_ele, nb_prod_act, net_margin, num_years_antig, origin_up, pow_max, price_p1_var, price_p2_var, price_p3_var, price_p1_fix, price_p2_fix, price_p3_fix

Features to be generated; num_years_renewal (time before contract renewal), avg_month_elec (Average consuption of electricity per month), avg_month_gas (Average consumption of gas per month),

In [25]:
```python
#generating Average consuption of electricity per month
Dataset['avg_month_elec'] = round(Dataset['cons_12m']/12, 2)

#generating Average consumption of gas per month
Dataset['avg_month_gas'] = round(Dataset['cons_gas_12m']/12, 2)

# generating time before contract renewal
Dataset['num_years_renewal']  = ((Dataset['date_renewal']).dt.year - (Dataset['date_activ']).dt.year)

Dataset.head(10)
```

Out[25]:

	id	channel_sales	cons_12m	cons_gas_12m	cons_last_month	date_activ	date_e
0	48ada52261e7cf58715202705a0451c9	lmkebamcaaclubfxadlmueccxoimlema	309275	0	10025	2012-11-07	2016-1
1	24011ae4ebbe3035111d65fa7c15bc57	foosdfpfkusacimwkcsosbicdxkicaua	0	54946	0	2013-06-15	2016-0
2	d29c2c54acc38ff3c0614d0a653813dd	Null	4660	0	0	2009-08-21	2016-0
3	764c75f661154dac3a6c254cd082ea7d	foosdfpfkusacimwkcsosbicdxkicaua	544	0	0	2010-04-16	2016-0
4	bba03439a292a1e166f80264c16191cb	lmkebamcaaclubfxadlmueccxoimlema	1584	0	0	2010-03-30	2016-0
5	568bb38a1afd7c0fc49c77b3789b59a3	foosdfpfkusacimwkcsosbicdxkicaua	121335	0	12400	2010-04-08	2016-0
6	149d57cf92fc41cf94415803a877cb4b	Null	4425	0	526	2010-01-13	2016-0
7	1aa498825382410b098937d65c4ec26d	usilxuppasemubllopkaafesmlibmsdf	8302	0	1998	2011-12-09	2016-1
8	7ab4bf4878d8f7661dfc20e9b8e18011	foosdfpfkusacimwkcsosbicdxkicaua	45097	0	0	2011-12-02	2016-1
9	01495c955be7ec5e7f3203406785aae0	foosdfpfkusacimwkcsosbicdxkicaua	29552	0	1260	2010-04-21	2016-0

10 rows × 41 columns

Feature Selection

In [26]:
```python
#Feature Selection

features = Dataset[['channel_sales', 'cons_12m', 'cons_gas_12m', 'cons_last_month', 'forecast_cons_12m'
, 'forecast_cons_year',
             'forecast_discount_energy', 'forecast_meter_rent_12m', 'forecast_price_energy_p1', 'foreca
st_price_energy_p2',
             'forecast_price_pow_p1', 'has_gas', 'imp_cons', 'margin_gross_pow_ele', 'margin_net_pow_el
e', 'nb_prod_act',
             'net_margin', 'num_years_antig', 'origin_up', 'pow_max', 'price_p1_var', 'price_p2_var',
'price_p3_var',
             'price_p1_fix', 'price_p2_fix', 'price_p3_fix', 'num_years_renewal', 'avg_month_elec', 'av
g_month_gas']]

target = Dataset['churn']
```