Importing modules

In [1]:
```
import pandas as pd
import matplotlib as plt
import numpy as np
```

Importing data

In [2]:
```
ml_tr=pd.read_csv("ml_case_training_data.csv")
ml_tr_hist=pd.read_csv("ml_case_training_hist_data.csv")
ml_tr_out=pd.read_csv("ml_case_training_output.csv")
```

Examining transaction data

Exploratory data analysis

In [3]:
```
ml_tr.head()
```

Out[3]:

	id	activity_new	campaign_disc_ele	channel_sales	cons_12m	cons_gas_12m	cc
0	48ada52261e7cf58715202705a0451c9	esoiiifxdlbkcsluxmfuacbdckommixw	NaN	lmkebamcaaclubfxadlmueccxoimlema	309275	0	
1	24011ae4ebbe3035111d65fa7c15bc57	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	0	54946	
2	d29c2c54acc38ff3c0614d0a653813dd	NaN	NaN	NaN	4660	0	
3	764c75f661154dac3a6c254cd082ea7d	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	544	0	
4	bba03439a292a1e166f80264c16191cb	NaN	NaN	lmkebamcaaclubfxadlmueccxoimlema	1584	0	

5 rows × 32 columns

In [4]:
```
ml_tr_hist.head()
```

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

In [5]:
```
ml_tr_out.head()
```

Out[5]:

	id	churn
0	48ada52261e7cf58715202705a0451c9	0
1	24011ae4ebbe3035111d65fa7c15bc57	1
2	d29c2c54acc38ff3c0614d0a653813dd	0
3	764c75f661154dac3a6c254cd082ea7d	0
4	bba03439a292a1e166f80264c16191cb	0

In [6]:
```
ml_tr.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  object
 8   date_end                  16094 non-null  object
 9   date_first_activ          3508 non-null   object
```

```
 10  date_modif_prod           15939 non-null  object
 11  date_renewal              16056 non-null  object
 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: float64(16), int64(6), object(10)
memory usage: 3.9+ MB
```

Many inconsistancies. Needs work.

In [7]:
```
ml_tr = ml_tr.fillna(ml_tr.mean())
ml_tr=ml_tr.drop(columns=["campaign_disc_ele"])
```

dropping dulpicates

In [8]:
```
ml_tr=ml_tr.drop_duplicates('id',keep='first')
```

Converting types

In [9]:
```
ml_tr['date_renewal']= pd.to_datetime(ml_tr['date_renewal'])
ml_tr['date_modif_prod']= pd.to_datetime(ml_tr['date_modif_prod'])
ml_tr['date_first_activ']= pd.to_datetime(ml_tr['date_first_activ'])
ml_tr['date_end']= pd.to_datetime(ml_tr['date_end'])
ml_tr['date_activ']= pd.to_datetime(ml_tr['date_activ'])
```

In [10]:
```
ml_tr.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16096 entries, 0 to 16095
Data columns (total 31 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   id                        16096 non-null  object
 1   activity_new              6551 non-null   object
 2   channel_sales             11878 non-null  object
 3   cons_12m                  16096 non-null  int64
 4   cons_gas_12m              16096 non-null  int64
 5   cons_last_month           16096 non-null  int64
 6   date_activ                16096 non-null  datetime64[ns]
 7   date_end                  16094 non-null  datetime64[ns]
 8   date_first_activ          3508 non-null   datetime64[ns]
 9   date_modif_prod           15939 non-null  datetime64[ns]
 10  date_renewal              16056 non-null  datetime64[ns]
 11  forecast_base_bill_ele    16096 non-null  float64
 12  forecast_base_bill_year   16096 non-null  float64
 13  forecast_bill_12m         16096 non-null  float64
 14  forecast_cons             16096 non-null  float64
 15  forecast_cons_12m         16096 non-null  float64
 16  forecast_cons_year        16096 non-null  int64
 17  forecast_discount_energy  16096 non-null  float64
 18  forecast_meter_rent_12m   16096 non-null  float64
 19  forecast_price_energy_p1  16096 non-null  float64
 20  forecast_price_energy_p2  16096 non-null  float64
 21  forecast_price_pow_p1     16096 non-null  float64
 22  has_gas                   16096 non-null  object
 23  imp_cons                  16096 non-null  float64
 24  margin_gross_pow_ele      16096 non-null  float64
 25  margin_net_pow_ele        16096 non-null  float64
 26  nb_prod_act               16096 non-null  int64
 27  net_margin                16096 non-null  float64
 28  num_years_antig           16096 non-null  int64
 29  origin_up                 16009 non-null  object
 30  pow_max                   16096 non-null  float64
dtypes: datetime64[ns](5), float64(15), int64(6), object(5)
memory usage: 3.9+ MB
```

In [11]:
```
ml_tr_out.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16096 entries, 0 to 16095
Data columns (total 2 columns):
```

```
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   id      16096 non-null  object
 1   churn   16096 non-null  int64
dtypes: int64(1), object(1)
memory usage: 251.6+ KB
```

No data missing.

In [12]:
```
ml_tr_hist.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  191643 non-null  float64
 3   price_p2_var  191643 non-null  float64
 4   price_p3_var  191643 non-null  float64
 5   price_p1_fix  191643 non-null  float64
 6   price_p2_fix  191643 non-null  float64
 7   price_p3_fix  191643 non-null  float64
dtypes: float64(6), object(2)
memory usage: 11.8+ MB
```

Data clearning needed.

In [13]:
```
ml_tr_hist=ml_tr_hist.dropna()
ml_tr_hist.info()
```

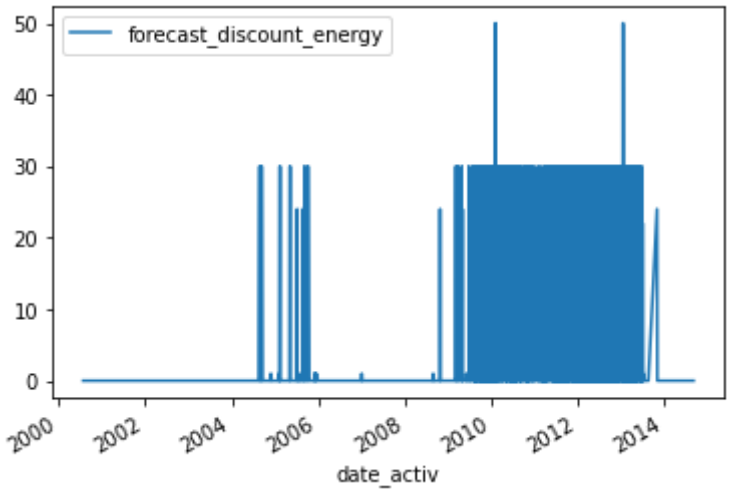
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 191643 entries, 0 to 193001
Data columns (total 8 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   id            191643 non-null  object
 1   price_date    191643 non-null  object
 2   price_p1_var  191643 non-null  float64
 3   price_p2_var  191643 non-null  float64
 4   price_p3_var  191643 non-null  float64
 5   price_p1_fix  191643 non-null  float64
```

```
 6   price_p2_fix  191643 non-null  float64
 7   price_p3_fix  191643 non-null  float64
dtypes: float64(6), object(2)
memory usage: 13.2+ MB
```

Sub-Task 2:

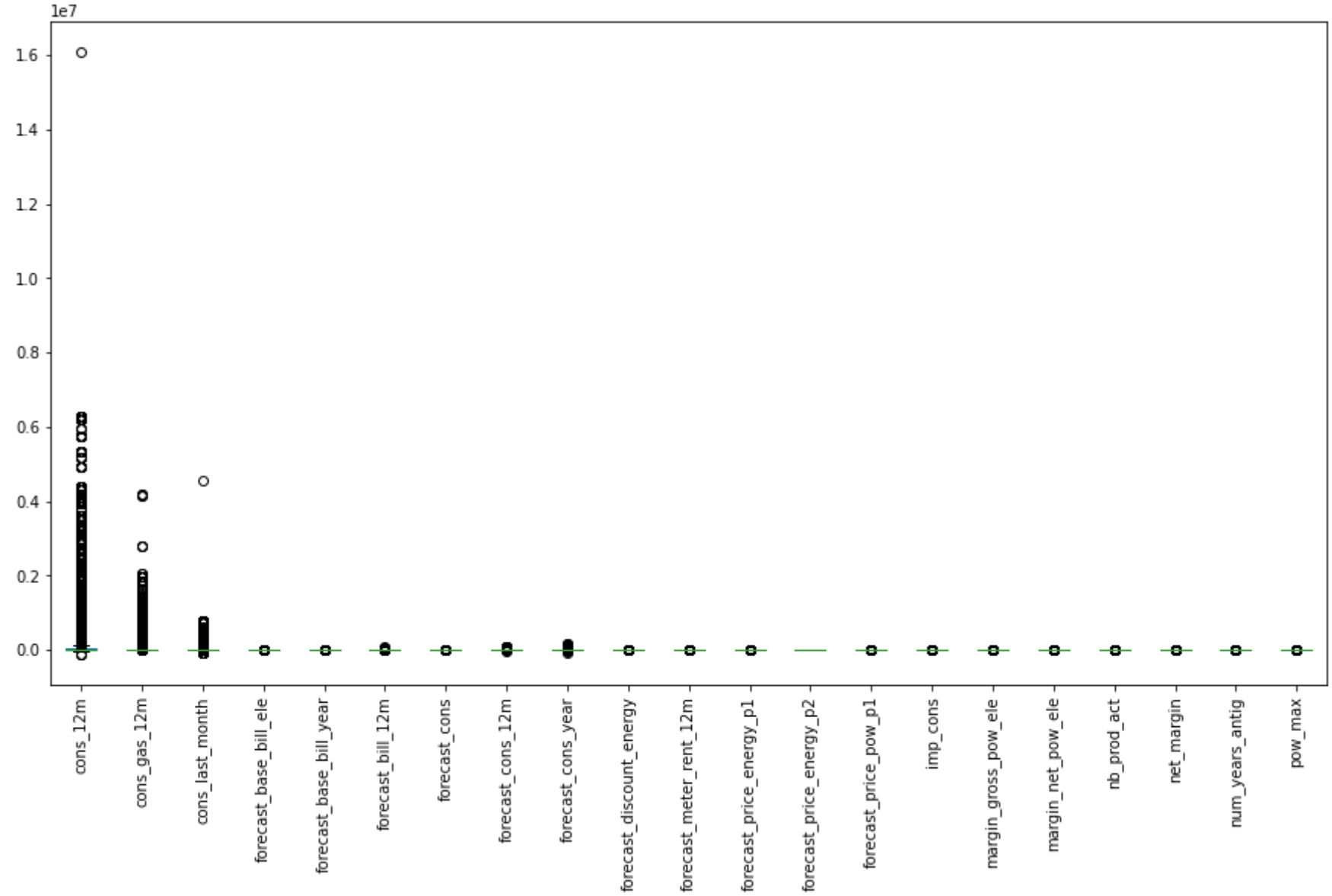
In [14]:
```
ml_tr.plot(x="date_activ",y="forecast_discount_energy")
```

Out[14]:
```
<AxesSubplot:xlabel='date_activ'>
```



In [19]:
```
ml_tr_scaled.plot(kind='box',figsize=(15,8),rot=90)
```

Out[19]:
```
<AxesSubplot:>
```


1e7
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
cons_12m
cons_gas_12m
cons_last_month
forecast_base_bill_ele
forecast_base_bill_year
forecast_bill_12m
forecast_cons
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
imp_cons
margin_gross_pow_ele
margin_net_pow_ele
nb_prod_act
net_margin
num_years_antig
pow_max