Task 3 - Feature Engineering

This notebook is for Feature Engineering

Data Dictionary

Summary

1. Drop columns that has high amounts of zeros and missing values (> 5% ot total)
2. Decided to drop forecast columns since not sure how accurate the values are in future
3. Decided not to use the other historical energy and power data since too many zeros

Import Libraries

In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
import datetime
from datetime import datetime, timedelta
import scipy.stats
import pandas_profiling
from pandas_profiling import ProfileReport


%matplotlib inline
#sets the default autosave frequency in seconds
%autosave 60
sns.set_style('dark')
sns.set(font_scale=1.2)

import warnings
warnings.filterwarnings('ignore')

#import feature_engine.missing_data_imputers as mdi
#from feature_engine.outlier_removers import Winsorizer
#from feature_engine import categorical_encoders as ce

pd.set_option('display.max_columns',None)
#pd.set_option('display.max_rows',None)
pd.set_option('display.width', 1000)

np.random.seed(0)
np.set_printoptions(suppress=True)
```

```
Autosaving every 60 seconds
```

In [2]:
```python
df = pd.read_csv("train.csv",
                 parse_dates=['date_activ', 'date_end', 'date_first_activ', 'date_modif_prod', 'date_renewal
                dayfirst=True)
```

In [3]:
```python
df
```

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
...	...	...	...	...	...
16091	18463073fb097fc0ac5d3e040f356987	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	32270
16092	d0a6f71671571ed83b2645d23af6de00	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	7223
16093	10e6828ddd62cbcf687cb74928c4c2d2	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	1844
16094	1cf20fd6206d7678d5bcafd28c53b4db	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	131
16095	563dde550fd624d7352f3de77c0cdfcd	NaN	NaN	NaN	8730

16096 rows × 33 columns

Exploratory Data Analysis

In [4]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16096 entries, 0 to 16095
Data columns (total 33 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
 32  churn                     16096 non-null  int64
dtypes: datetime64[ns](5), float64(16), int64(7), object(5)
memory usage: 4.1+ MB
```

In [5]:
```python
df.describe()
```

Out[5]:

	campaign_disc_ele	cons_12m	cons_gas_12m	cons_last_month	forecast_base_bill_ele	forecast_base_bill_year	forecast_bill_12m	for
count	0.0	1.609600e+04	1.609600e+04	1.609600e+04	3508.000000	3508.000000	3508.000000	3
mean	NaN	1.948044e+05	3.191164e+04	1.946154e+04	335.843857	335.843857	3837.441866	
std	NaN	6.795151e+05	1.775885e+05	8.235676e+04	649.406000	649.406000	5425.744327	
min	NaN	-1.252760e+05	-3.037000e+03	-9.138600e+04	-364.940000	-364.940000	-2503.480000	
25%	NaN	5.906250e+03	0.000000e+00	0.000000e+00	0.000000	0.000000	1158.175000	
50%	NaN	1.533250e+04	0.000000e+00	9.010000e+02	162.955000	162.955000	2187.230000	
75%	NaN	5.022150e+04	0.000000e+00	4.127000e+03	396.185000	396.185000	4246.555000	
max	NaN	1.609711e+07	4.188440e+06	4.538720e+06	12566.080000	12566.080000	81122.630000	9

In [6]:
```python
df.columns
```

Out[6]:
```
Index(['id', 'activity_new', 'campaign_disc_ele', 'channel_sales', 'cons_12m', 'cons_gas_12m', 'cons_last_mo
nth', 'date_activ', 'date_end', 'date_first_activ', 'date_modif_prod', 'date_renewal', 'forecast_base_bill_e
le', 'forecast_base_bill_year', 'forecast_bill_12m', 'forecast_cons', 'forecast_cons_12m', 'forecast_cons_ye
ar', 'forecast_discount_energy', 'forecast_meter_rent_12m', 'forecast_price_energy_p1', 'forecast_price_ener
gy_p2', 'forecast_price_pow_p1', 'has_gas', 'imp_cons', 'margin_gross_pow_ele', 'margin_net_pow_ele', 'nb_pr
od_act', 'net_margin', 'num_years_antig', 'origin_up', 'pow_max', 'churn'], dtype='object')
```

Drop unwanted features

In [7]:
```python
df.columns
```

Out[7]:
```
Index(['id', 'activity_new', 'campaign_disc_ele', 'channel_sales', 'cons_12m', 'cons_gas_12m', 'cons_last_mo
nth', 'date_activ', 'date_end', 'date_first_activ', 'date_modif_prod', 'date_renewal', 'forecast_base_bill_e
le', 'forecast_base_bill_year', 'forecast_bill_12m', 'forecast_cons', 'forecast_cons_12m', 'forecast_cons_ye
ar', 'forecast_discount_energy', 'forecast_meter_rent_12m', 'forecast_price_energy_p1', 'forecast_price_ener
gy_p2', 'forecast_price_pow_p1', 'has_gas', 'imp_cons', 'margin_gross_pow_ele', 'margin_net_pow_ele', 'nb_pr
od_act', 'net_margin', 'num_years_antig', 'origin_up', 'pow_max', 'churn'], dtype='object')
```

In [8]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16096 entries, 0 to 16095
Data columns (total 33 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
 32  churn                     16096 non-null  int64
dtypes: datetime64[ns](5), float64(16), int64(7), object(5)
memory usage: 4.1+ MB
```

In [9]:
```python
df.drop(['activity_new', 'campaign_disc_ele', 'channel_sales','date_first_activ', 'cons_12m', 'cons_gas_12m'
         'date_modif_prod', 'date_renewal', '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','origin_up'], axis=1, inplace=True)
```

In [10]:
```python
df.head()
```

Out[10]:

	id	cons_last_month	date_activ	date_end	has_gas	margin_gross_pow_ele	margin_net_pow_ele	nb_prod_a
0	48ada52261e7cf58715202705a0451c9	10025	2012-11-07	2016-11-06	f	-41.76	-41.76	
1	24011ae4ebbe3035111d65fa7c15bc57	0	2013-06-15	2016-06-15	t	25.44	25.44	
2	d29c2c54acc38ff3c0614d0a653813dd	0	2009-08-21	2016-08-30	f	16.38	16.38	
3	764c75f661154dac3a6c254cd082ea7d	0	2010-04-16	2016-04-16	f	28.60	28.60	
4	bba03439a292a1e166f80264c16191cb	0	2010-03-30	2016-03-30	f	30.22	30.22	

In [11]:
```python
df["duration"] = (df.date_end - df.date_activ).dt.days
```

In [12]:
```python
df.head()
```

Out[12]:

	id	cons_last_month	date_activ	date_end	has_gas	margin_gross_pow_ele	margin_net_pow_ele	nb_prod_a
0	48ada52261e7cf58715202705a0451c9	10025	2012-11-07	2016-11-06	f	-41.76	-41.76	
1	24011ae4ebbe3035111d65fa7c15bc57	0	2013-06-15	2016-06-15	t	25.44	25.44	
2	d29c2c54acc38ff3c0614d0a653813dd	0	2009-08-21	2016-08-30	f	16.38	16.38	
3	764c75f661154dac3a6c254cd082ea7d	0	2010-04-16	2016-04-16	f	28.60	28.60	
4	bba03439a292a1e166f80264c16191cb	0	2010-03-30	2016-03-30	f	30.22	30.22	

In [13]:
```python
df2 = pd.read_csv("customerchurn.csv")
```

In [14]:
```python
df2
```

Out[14]:

	id	price_p1_var	price_p2_var	price_p3_var	price_p1_fix	price_p2_fix	price_p3_fix	churn
0	0002203ffbb812588b632b9e628cc38d	0.124338	0.103794	0.073160	40.701732	24.421038	16.280694	0
1	0004351ebdd665e6ee664792efc4fd13	0.146427	0.000000	0.000000	44.385450	0.000000	0.000000	0
2	0010bcc39e42b3c2131ed2ce55246e3c	0.181559	0.000000	0.000000	45.319710	0.000000	0.000000	0
3	0010ee3855fdea87602a5b7aba8e42de	0.118757	0.098292	0.069032	40.647427	24.388455	16.258971	0
4	00114d74e963e47177db89bc70108537	0.147926	0.000000	0.000000	44.266930	0.000000	0.000000	0
...	...	...	...	...	...	...	...	...
16091	ffef185810e44254c3a4c6395e6b4d8a	0.138863	0.115125	0.080780	40.896427	24.637456	16.507972	0
16092	fffac626da707b1b5ab11e8431a4d0a2	0.147137	0.000000	0.000000	44.311375	0.000000	0.000000	0
16093	fffc0cacd305dd51f316424bbb08d1bd	0.153879	0.129497	0.094842	41.160171	24.895768	16.763569	0
16094	fffe4f5646aa39c7f97f95ae2679ce64	0.123858	0.103499	0.073735	40.606699	24.364017	16.242678	0
16095	ffff7fa066f1fb305ae285bb03bf325a	0.125360	0.104895	0.075635	40.647427	24.388455	16.258971	0

16096 rows × 8 columns

In [15]:
```python
df3 = pd.merge(left=df, right=df2, on="id", how="inner")
```

In [16]:
```python
df3
```

Out[16]:

	id	cons_last_month	date_activ	date_end	has_gas	margin_gross_pow_ele	margin_net_pow_ele	nb_p
0	48ada52261e7cf58715202705a0451c9	10025	2012-11-07	2016-11-06	f	-41.76	-41.76	
1	24011ae4ebbe3035111d65fa7c15bc57	0	2013-06-15	2016-06-15	t	25.44	25.44	
2	d29c2c54acc38ff3c0614d0a653813dd	0	2009-08-21	2016-08-30	f	16.38	16.38	
3	764c75f661154dac3a6c254cd082ea7d	0	2010-04-16	2016-04-16	f	28.60	28.60	
4	bba03439a292a1e166f80264c16191cb	0	2010-03-30	2016-03-30	f	30.22	30.22	
...	...	...	...	...	...	...	...	
16091	18463073fb097fc0ac5d3e040f356987	0	2012-05-24	2016-05-08	t	27.88	27.88	
16092	d0a6f71671571ed83b2645d23af6de00	181	2012-08-27	2016-08-27	f	0.00	0.00	
16093	10e6828ddd62cbcf687cb74928c4c2d2	179	2012-02-08	2016-02-07	f	39.84	39.84	
16094	1cf20fd6206d7678d5bcafd28c53b4db	0	2012-08-30	2016-08-30	f	13.08	13.08	
16095	563dde550fd624d7352f3de77c0cdfcd	0	2009-12-18	2016-12-17	f	11.84	11.84	

16096 rows × 20 columns

In [17]:
```python
df3.drop(['date_activ', 'date_end', 'id', 'churn_x'],axis=1,inplace=True)
```

In [18]:
```python
df3.head()
```

Out[18]:

	cons_last_month	has_gas	margin_gross_pow_ele	margin_net_pow_ele	nb_prod_act	net_margin	num_years_antig	pow_max	duration	pr
0	10025	f	-41.76	-41.76	1	1732.36	3	180.000	1460.0	
1	0	t	25.44	25.44	2	678.99	3	43.648	1096.0	
2	0	f	16.38	16.38	1	18.89	6	13.800	2566.0	
3	0	f	28.60	28.60	1	6.60	6	13.856	2192.0	
4	0	f	30.22	30.22	1	25.46	6	13.200	2192.0	

Data Preprocessing

Treat Missing Values

In [19]:
```python
df3.isnull().sum()
```

Out[19]:
```
cons_last_month          0
has_gas                  0
margin_gross_pow_ele    13
margin_net_pow_ele      13
nb_prod_act              0
net_margin              15
num_years_antig          0
pow_max                  3
duration                 2
price_p1_var             2
price_p2_var             2
price_p3_var             2
price_p1_fix             2
price_p2_fix             2
price_p3_fix             2
churn_y                  0
dtype: int64
```

In [20]:
```python
df3.dropna(inplace=True)
```

In [21]:
```python
df3.isnull().sum()
```

Out[21]:
```
cons_last_month         0
has_gas                 0
margin_gross_pow_ele    0
margin_net_pow_ele      0
nb_prod_act             0
net_margin              0
num_years_antig         0
pow_max                 0
duration                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
churn_y                 0
dtype: int64
```

Treat Duplicate Values

In [22]:
```python
df3.duplicated(keep='first').sum()
```

Out[22]: 1

In [23]:
```python
df3.drop_duplicates(ignore_index=True, inplace=True)
```

In [24]:
```python
df3.duplicated(keep='first').sum()
```

Out[24]: 0

One-hot encoding

In [25]:
```python
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16076 entries, 0 to 16075
Data columns (total 16 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   cons_last_month       16076 non-null  int64
 1   has_gas               16076 non-null  object
 2   margin_gross_pow_ele  16076 non-null  float64
 3   margin_net_pow_ele    16076 non-null  float64
 4   nb_prod_act           16076 non-null  int64
 5   net_margin            16076 non-null  float64
 6   num_years_antig       16076 non-null  int64
 7   pow_max               16076 non-null  float64
 8   duration              16076 non-null  float64
 9   price_p1_var          16076 non-null  float64
 10  price_p2_var          16076 non-null  float64
 11  price_p3_var          16076 non-null  float64
 12  price_p1_fix          16076 non-null  float64
 13  price_p2_fix          16076 non-null  float64
 14  price_p3_fix          16076 non-null  float64
 15  churn_y               16076 non-null  int64
dtypes: float64(11), int64(4), object(1)
memory usage: 2.0+ MB
```

In [26]:
```python
df3["has_gas"] = pd.get_dummies(data=df["has_gas"],drop_first=True)
```

In [27]:
```python
df3
```

Out[27]:

	cons_last_month	has_gas	margin_gross_pow_ele	margin_net_pow_ele	nb_prod_act	net_margin	num_years_antig	pow_max	duration
0	10025	0	-41.76	-41.76	1	1732.36	3	180.000	1460.(
1	0	1	25.44	25.44	2	678.99	3	43.648	1096.(
2	0	0	16.38	16.38	1	18.89	6	13.800	2566.(
3	0	0	28.60	28.60	1	6.60	6	13.856	2192.(
4	0	0	30.22	30.22	1	25.46	6	13.200	2192.(
...	...	...	...	...	...	...	...	...	...
16071	0	0	27.88	27.88	2	381.77	4	15.000	1445.(
16072	181	0	0.00	0.00	1	90.34	3	6.000	1461.(
16073	179	0	39.84	39.84	1	20.38	4	15.935	1460.(
16074	0	0	13.08	13.08	1	0.96	3	11.000	1461.(
16075	0	1	11.84	11.84	1	96.34	6	10.392	2556.(

16076 rows × 16 columns

In [28]:
```python
df3.rename(columns={"churn_y":"churn"}, inplace=True)
```

In [29]:
```python
#df3.to_csv("final2.csv",index=False)
```