Task

December 9, 2025

1 Customer Churn Prediction

1.1 Import Libraries

[60]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from datetime import datetime

from sklearn.preprocessing import LabelEncoder
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

from sklearn.pipeline import Pipeline

from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix

import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

1.1.1 Load Dataset

[4]:
```
client = pd.read_csv('D:\kuliah\magang\Virtual Internship BCG X\client_data (1).
↪csv')
price = pd.read_csv('D:\kuliah\magang\Virtual Internship BCG X\price_data (1).
↪csv')
```

```
<>:1: SyntaxWarning: invalid escape sequence '\k'
<>:2: SyntaxWarning: invalid escape sequence '\k'
<>:1: SyntaxWarning: invalid escape sequence '\k'
<>:2: SyntaxWarning: invalid escape sequence '\k'
C:\Users\ASUS\AppData\Local\Temp\ipykernel_3264\559035398.py:1: SyntaxWarning:
invalid escape sequence '\k'
  client = pd.read_csv('D:\kuliah\magang\Virtual Internship BCG X\client_data
(1).csv')
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_3264\559035398.py:2: SyntaxWarning:
invalid escape sequence '\k'
  price = pd.read_csv('D:\kuliah\magang\Virtual Internship BCG X\price_data
(1).csv')
```

[5]:
```
client
```

[5]:
```
                                      id                    channel_sales  \
0      24011ae4ebbe3035111d65fa7c15bc57  foosdfpfkusacimwkcsosbicdxkicaua
1      d29c2c54acc38ff3c0614d0a653813dd                           MISSING
2      764c75f661154dac3a6c254cd082ea7d  foosdfpfkusacimwkcsosbicdxkicaua
3      bba03439a292a1e166f80264c16191cb  lmkebamcaaclubfxadlmueccxoimlema
4      149d57cf92fc41cf94415803a877cb4b                           MISSING
…                                    …                                 …
14601  18463073fb097fc0ac5d3e040f356987  foosdfpfkusacimwkcsosbicdxkicaua
14602  d0a6f71671571ed83b2645d23af6de00  foosdfpfkusacimwkcsosbicdxkicaua
14603  10e6828ddd62cbcf687cb74928c4c2d2  foosdfpfkusacimwkcsosbicdxkicaua
14604  1cf20fd6206d7678d5bcafd28c53b4db  foosdfpfkusacimwkcsosbicdxkicaua
14605  563dde550fd624d7352f3de77c0cdfcd                           MISSING

       cons_12m  cons_gas_12m  cons_last_month  date_activ    date_end  \
0             0         54946                0  2013-06-15  2016-06-15
1          4660             0                0  2009-08-21  2016-08-30
2           544             0                0  2010-04-16  2016-04-16
3          1584             0                0  2010-03-30  2016-03-30
4          4425             0              526  2010-01-13  2016-03-07
…             …             …                …           …           …
14601     32270         47940                0  2012-05-24  2016-05-08
14602      7223             0              181  2012-08-27  2016-08-27
14603      1844             0              179  2012-02-08  2016-02-07
14604       131             0                0  2012-08-30  2016-08-30
14605      8730             0                0  2009-12-18  2016-12-17

      date_modif_prod date_renewal  forecast_cons_12m  …  has_gas  imp_cons  \
0          2015-11-01   2015-06-23               0.00  …        t      0.00
1          2009-08-21   2015-08-31             189.95  …        f      0.00
2          2010-04-16   2015-04-17              47.96  …        f      0.00
3          2010-03-30   2015-03-31             240.04  …        f      0.00
4          2010-01-13   2015-03-09             445.75  …        f     52.32
…                 …            …                  …  …        …        …
14601      2015-05-08   2014-05-26            4648.01  …        t      0.00
14602      2012-08-27   2015-08-28             631.69  …        f     15.94
14603      2012-02-08   2015-02-09             190.39  …        f     18.05
14604      2012-08-30   2015-08-31              19.34  …        f      0.00
14605      2009-12-18   2015-12-21             762.41  …        f      0.00

       margin_gross_pow_ele  margin_net_pow_ele  nb_prod_act  net_margin  \
```

```
0                    25.44                25.44            2      678.99
1                    16.38                16.38            1       18.89
2                    28.60                28.60            1        6.60
3                    30.22                30.22            1       25.46
4                    44.91                44.91            1       47.98
…                    …                  …          …          …
14601                27.88                27.88            2      381.77
14602                 0.00                 0.00            1       90.34
14603                39.84                39.84            1       20.38
14604                13.08                13.08            1        0.96
14605                11.84                11.84            1       96.34

      num_years_antig                          origin_up  pow_max  churn
0                   3  lxidpiddsbxsbosboudacockeimpuepw   43.648      1
1                   6  kamkkxfxxuwbdslkwifmmcsiusiuosws   13.800      0
2                   6  kamkkxfxxuwbdslkwifmmcsiusiuosws   13.856      0
3                   6  kamkkxfxxuwbdslkwifmmcsiusiuosws   13.200      0
4                   6  kamkkxfxxuwbdslkwifmmcsiusiuosws   19.800      0
…               …                                  …      …    …
14601               4  lxidpiddsbxsbosboudacockeimpuepw   15.000      0
14602               3  lxidpiddsbxsbosboudacockeimpuepw    6.000      1
14603               4  lxidpiddsbxsbosboudacockeimpuepw   15.935      1
14604               3  lxidpiddsbxsbosboudacockeimpuepw   11.000      0
14605               6  ldkssxwpmemidmecebumciepifcamkci   10.392      0

[14606 rows x 26 columns]
```

[6]:
```
price
```

[6]:
```
                                     id  price_date  price_off_peak_var  \
0       038af19179925da21a25619c5a24b745  2015-01-01            0.151367
1       038af19179925da21a25619c5a24b745  2015-02-01            0.151367
2       038af19179925da21a25619c5a24b745  2015-03-01            0.151367
3       038af19179925da21a25619c5a24b745  2015-04-01            0.149626
4       038af19179925da21a25619c5a24b745  2015-05-01            0.149626
…                                    …           …                   …
192997  16f51cdc2baa19af0b940ee1b3dd17d5  2015-08-01            0.119916
192998  16f51cdc2baa19af0b940ee1b3dd17d5  2015-09-01            0.119916
192999  16f51cdc2baa19af0b940ee1b3dd17d5  2015-10-01            0.119916
193000  16f51cdc2baa19af0b940ee1b3dd17d5  2015-11-01            0.119916
193001  16f51cdc2baa19af0b940ee1b3dd17d5  2015-12-01            0.119916

        price_peak_var  price_mid_peak_var  price_off_peak_fix  \
0             0.000000            0.000000           44.266931
1             0.000000            0.000000           44.266931
2             0.000000            0.000000           44.266931
3             0.000000            0.000000           44.266931
```

```
4             0.000000            0.000000           44.266931
…                …                   …                  …
192997        0.102232            0.076257           40.728885
192998        0.102232            0.076257           40.728885
192999        0.102232            0.076257           40.728885
193000        0.102232            0.076257           40.728885
193001        0.102232            0.076257           40.728885

        price_peak_fix  price_mid_peak_fix
0              0.00000            0.000000
1              0.00000            0.000000
2              0.00000            0.000000
3              0.00000            0.000000
4              0.00000            0.000000
…                 …                   …
192997        24.43733           16.291555
192998        24.43733           16.291555
192999        24.43733           16.291555
193000        24.43733           16.291555
193001        24.43733           16.291555

[193002 rows x 8 columns]
```

1.2 EDA

[7]:
```
client.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14606 entries, 0 to 14605
Data columns (total 26 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              14606 non-null  object
 1   channel_sales                   14606 non-null  object
 2   cons_12m                        14606 non-null  int64
 3   cons_gas_12m                    14606 non-null  int64
 4   cons_last_month                 14606 non-null  int64
 5   date_activ                      14606 non-null  object
 6   date_end                        14606 non-null  object
 7   date_modif_prod                 14606 non-null  object
 8   date_renewal                    14606 non-null  object
 9   forecast_cons_12m               14606 non-null  float64
 10  forecast_cons_year              14606 non-null  int64
 11  forecast_discount_energy        14606 non-null  float64
 12  forecast_meter_rent_12m         14606 non-null  float64
 13  forecast_price_energy_off_peak  14606 non-null  float64
 14  forecast_price_energy_peak      14606 non-null  float64
 15  forecast_price_pow_off_peak     14606 non-null  float64
```

```
 16  has_gas                          14606 non-null  object
 17  imp_cons                         14606 non-null  float64
 18  margin_gross_pow_ele             14606 non-null  float64
 19  margin_net_pow_ele               14606 non-null  float64
 20  nb_prod_act                      14606 non-null  int64
 21  net_margin                       14606 non-null  float64
 22  num_years_antig                  14606 non-null  int64
 23  origin_up                        14606 non-null  object
 24  pow_max                          14606 non-null  float64
 25  churn                            14606 non-null  int64
dtypes: float64(11), int64(7), object(8)
memory usage: 2.9+ MB
```

[8]:
```
price.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_off_peak_var   193002 non-null  float64
 3   price_peak_var       193002 non-null  float64
 4   price_mid_peak_var   193002 non-null  float64
 5   price_off_peak_fix   193002 non-null  float64
 6   price_peak_fix       193002 non-null  float64
 7   price_mid_peak_fix   193002 non-null  float64
dtypes: float64(6), object(2)
memory usage: 11.8+ MB
```

pada kedua dataset ada kesalahan tipe data

[12]:
```
date_cols = ['date_activ', 'date_end', 'date_modif_prod', 'date_renewal']
for col in date_cols:
    client[col] = pd.to_datetime(client[col], errors='coerce')
```

[9]:
```
client.isnull().sum()
```

[9]:
```
id                              0
channel_sales                   0
cons_12m                        0
cons_gas_12m                    0
cons_last_month                 0
date_activ                      0
date_end                        0
date_modif_prod                 0
date_renewal                    0
forecast_cons_12m               0
```

```
forecast_cons_year                0
forecast_discount_energy          0
forecast_meter_rent_12m           0
forecast_price_energy_off_peak    0
forecast_price_energy_peak        0
forecast_price_pow_off_peak       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
churn                             0
dtype: int64
```

[10]:
```
price.isnull().sum()
```

[10]:
```
id                  0
price_date          0
price_off_peak_var  0
price_peak_var      0
price_mid_peak_var  0
price_off_peak_fix  0
price_peak_fix      0
price_mid_peak_fix  0
dtype: int64
```

tidak ada data kosong dalam dataset client dan price

[13]:
```
client.duplicated().sum()
```

[13]:
```
np.int64(0)
```

[14]:
```
price.duplicated().sum()
```

[14]:
```
np.int64(0)
```

[18]:
```
client.describe().T
```

[18]:
```
                                 count                           mean  \
cons_12m                       14606.0                  159220.286252
cons_gas_12m                   14606.0                   28092.375325
cons_last_month                14606.0                   16090.269752
date_activ                       14606  2011-01-28 07:54:18.879912448
date_end                         14606  2016-07-27 20:48:26.422018560
```

date_modif_prod	14606	2013-01-02 12:29:10.951663872	
date_renewal	14606	2015-07-21 06:59:00.353279488	
forecast_cons_12m	14606.0	1868.61488	
forecast_cons_year	14606.0	1399.762906	
forecast_discount_energy	14606.0	0.966726	
forecast_meter_rent_12m	14606.0	63.086871	
forecast_price_energy_off_peak	14606.0	0.137283	
forecast_price_energy_peak	14606.0	0.050491	
forecast_price_pow_off_peak	14606.0	43.130056	
imp_cons	14606.0	152.786896	
margin_gross_pow_ele	14606.0	24.565121	
margin_net_pow_ele	14606.0	24.562517	
nb_prod_act	14606.0	1.292346	
net_margin	14606.0	189.264522	
num_years_antig	14606.0	4.997809	
pow_max	14606.0	18.135136	
churn	14606.0	0.097152	

	min	25%	\
cons_12m	0.0	5674.75	
cons_gas_12m	0.0	0.0	
cons_last_month	0.0	0.0	
date_activ	2003-05-09 00:00:00	2010-01-15 00:00:00	
date_end	2016-01-28 00:00:00	2016-04-27 06:00:00	
date_modif_prod	2003-05-09 00:00:00	2010-08-12 00:00:00	
date_renewal	2013-06-26 00:00:00	2015-04-17 00:00:00	
forecast_cons_12m	0.0	494.995	
forecast_cons_year	0.0	0.0	
forecast_discount_energy	0.0	0.0	
forecast_meter_rent_12m	0.0	16.18	
forecast_price_energy_off_peak	0.0	0.11634	
forecast_price_energy_peak	0.0	0.0	
forecast_price_pow_off_peak	0.0	40.606701	
imp_cons	0.0	0.0	
margin_gross_pow_ele	0.0	14.28	
margin_net_pow_ele	0.0	14.28	
nb_prod_act	1.0	1.0	
net_margin	0.0	50.7125	
num_years_antig	1.0	4.0	
pow_max	3.3	12.5	
churn	0.0	0.0	

	50%	75%	\
cons_12m	14115.5	40763.75	
cons_gas_12m	0.0	0.0	
cons_last_month	792.5	3383.0	
date_activ	2011-03-04 00:00:00	2012-04-19 00:00:00	

date_end	2016-08-01 00:00:00	2016-10-31 00:00:00
date_modif_prod	2013-06-19 00:00:00	2015-06-16 00:00:00
date_renewal	2015-07-27 00:00:00	2015-10-29 00:00:00
forecast_cons_12m	1112.875	2401.79
forecast_cons_year	314.0	1745.75
forecast_discount_energy	0.0	0.0
forecast_meter_rent_12m	18.795	131.03
forecast_price_energy_off_peak	0.143166	0.146348
forecast_price_energy_peak	0.084138	0.098837
forecast_price_pow_off_peak	44.311378	44.311378
imp_cons	37.395	193.98
margin_gross_pow_ele	21.64	29.88
margin_net_pow_ele	21.64	29.88
nb_prod_act	1.0	1.0
net_margin	112.53	243.0975
num_years_antig	5.0	6.0
pow_max	13.856	19.1725
churn	0.0	0.0

	max	std
cons_12m	6207104.0	573465.264198
cons_gas_12m	4154590.0	162973.059057
cons_last_month	771203.0	64364.196422
date_activ	2014-09-01 00:00:00	NaN
date_end	2017-06-13 00:00:00	NaN
date_modif_prod	2016-01-29 00:00:00	NaN
date_renewal	2016-01-28 00:00:00	NaN
forecast_cons_12m	82902.83	2387.571531
forecast_cons_year	175375.0	3247.786255
forecast_discount_energy	30.0	5.108289
forecast_meter_rent_12m	599.31	66.165783
forecast_price_energy_off_peak	0.273963	0.024623
forecast_price_energy_peak	0.195975	0.049037
forecast_price_pow_off_peak	59.266378	4.485988
imp_cons	15042.79	341.369366
margin_gross_pow_ele	374.64	20.231172
margin_net_pow_ele	374.64	20.23028
nb_prod_act	32.0	0.709774
net_margin	24570.65	311.79813
num_years_antig	13.0	1.611749
pow_max	320.0	13.534743
churn	1.0	0.296175

berdasarkan analisis deskriptif, dapat ditarik beberapa pain points. 1. Banyak Variabel Konsumsi Memiliki Skewness Tinggi (Right-Skewed Parah Variabel seperti: * cons_12m → max 6,207,104 vs median 14,115 * cons_gas_12m → max 4,154,590 vs median 0 * cons_last_month → max 771,203 vs median 792

Distribusi sangat tidak normal (heavy-tail), ada outlier ekstrem, bisa mengganggu model predikt

2. Banyak Variabel dengan Median = 0 → Banyak Zero-Inflated Features
   - cons_gas_12m: median = 0
   - cons_last_month: Q1 = 0
   - forecast_cons_year: Q1 & median = 0

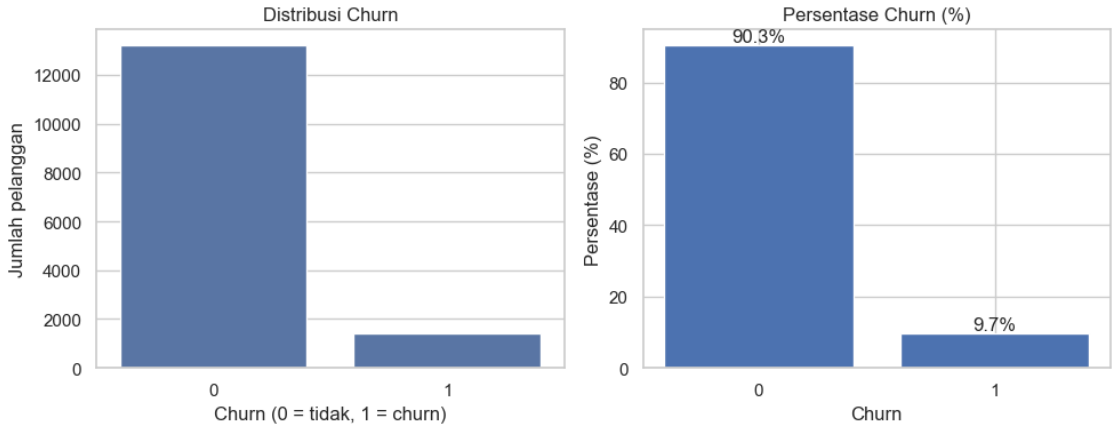
   Ada indikasi banyak pelanggan tidak memiliki konsumsi tertentu
3. Variabel nb_prod_act Sangat Tidak Seimbang
   - mean = 1.29
   - median = 1
   - max = 32 (!!)

   Hampir semua pelanggan hanya menggunakan 1 produk, tapi ada outlier hingga 32 produk → perlu normalisasi atau capping
4. Variabel churn Sangat Imbalanced
5. Variabel margin_gross_pow_ele dan margin_net_pow_ele Identik Mean, median, max, std semuanya sama. Hal tersebut dapatmenyebabkan redundansi.

[20]:
```
client["tenure_years"] = (client["date_end"] - client["date_activ"]).dt.days /␣
↪365.25
```

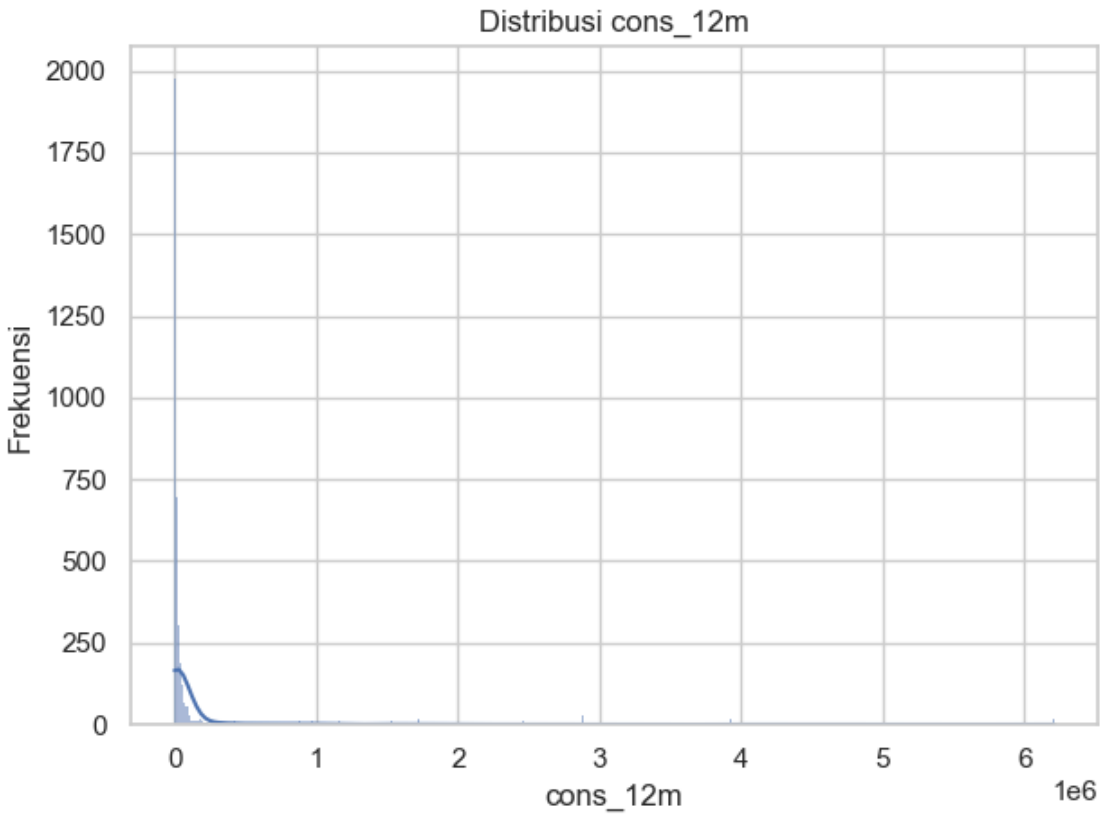
1.2.1 visualisasi

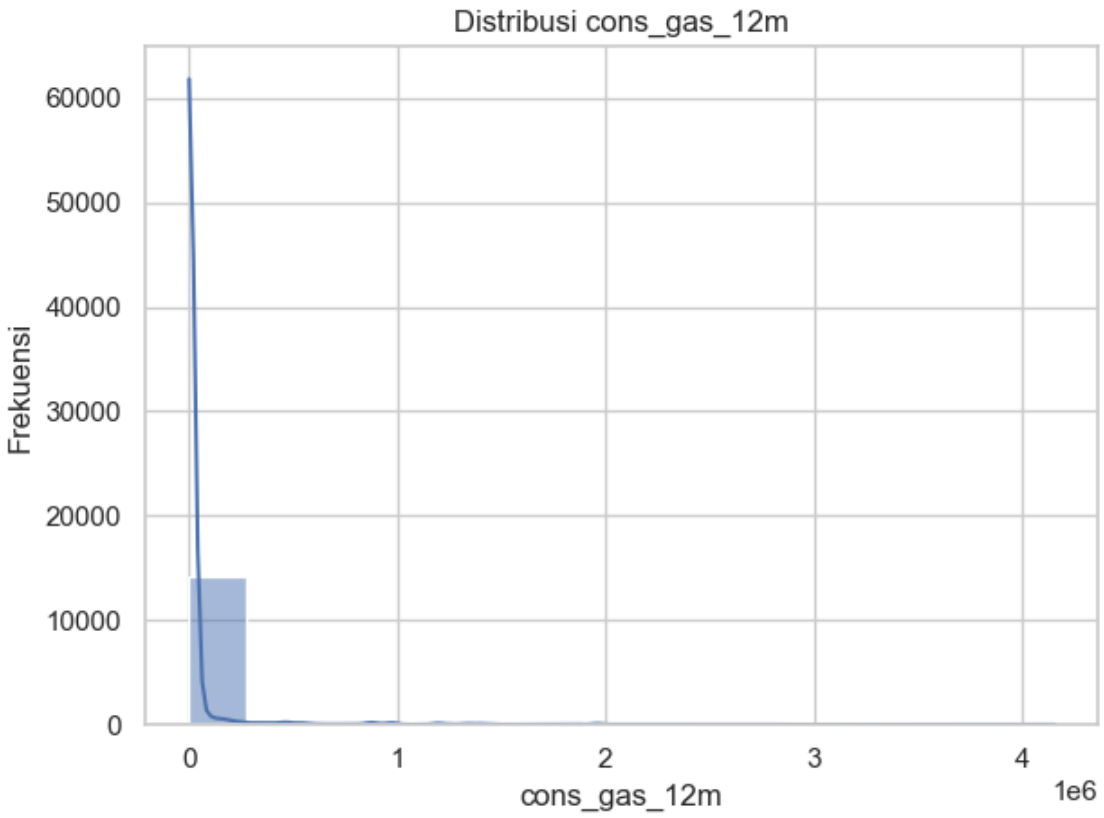
[25]:
```
fig, ax = plt.subplots(1, 2, figsize=(10, 4))

# Countplot
sns.countplot(data=client, x="churn", ax=ax[0])
ax[0].set_title("Distribusi Churn")
ax[0].set_xlabel("Churn (0 = tidak, 1 = churn)")
ax[0].set_ylabel("Jumlah pelanggan")
churn_rate = client["churn"].value_counts(normalize=True) * 100
ax[1].bar(churn_rate.index.astype(str), churn_rate.values)
ax[1].set_title("Persentase Churn (%)")
ax[1].set_xlabel("Churn")
ax[1].set_ylabel("Persentase (%)")

for i, v in enumerate(churn_rate.values):
    ax[1].text(i, v + 1, f"{v:.1f}%", ha="center")

plt.tight_layout()
```

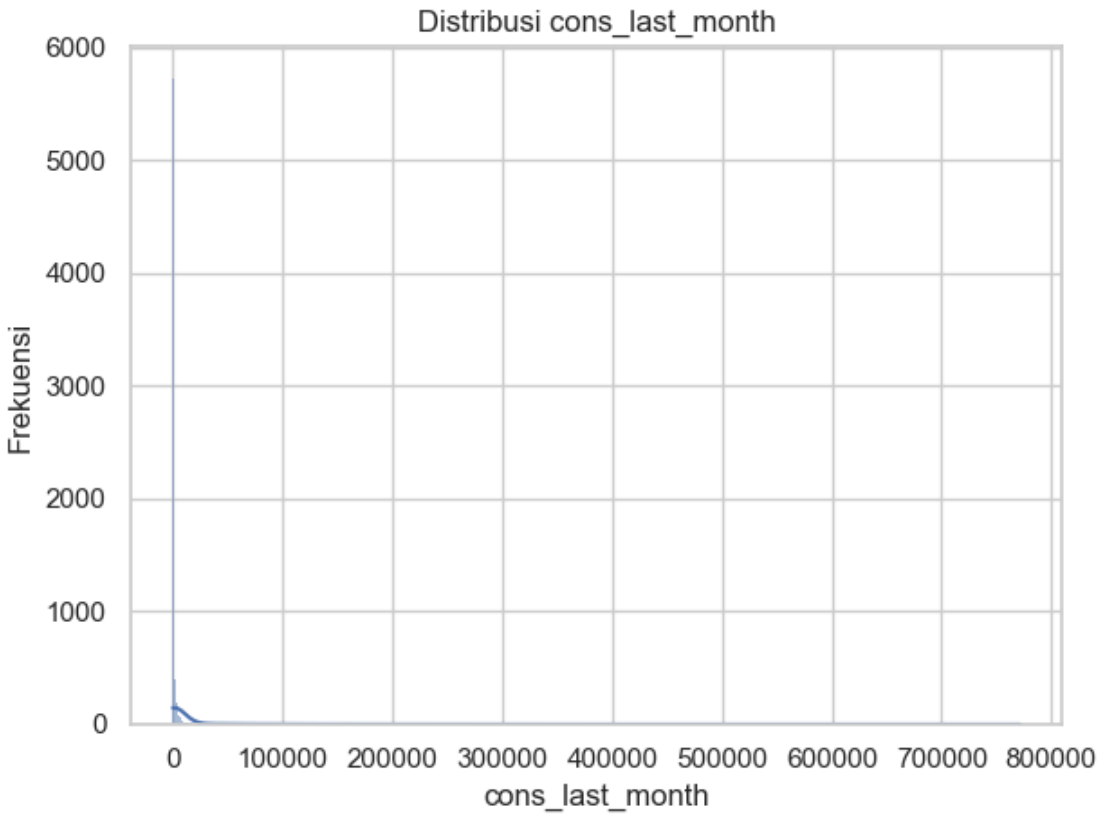


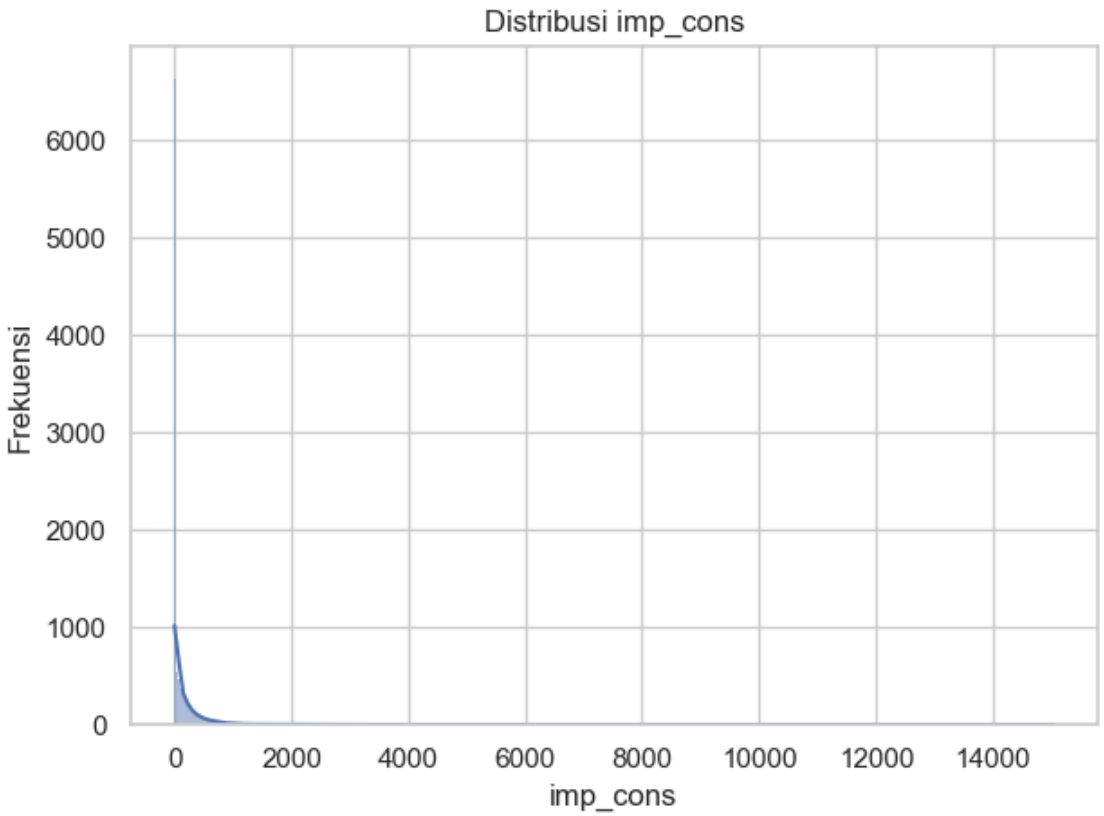
kelas target sangat imbalanced

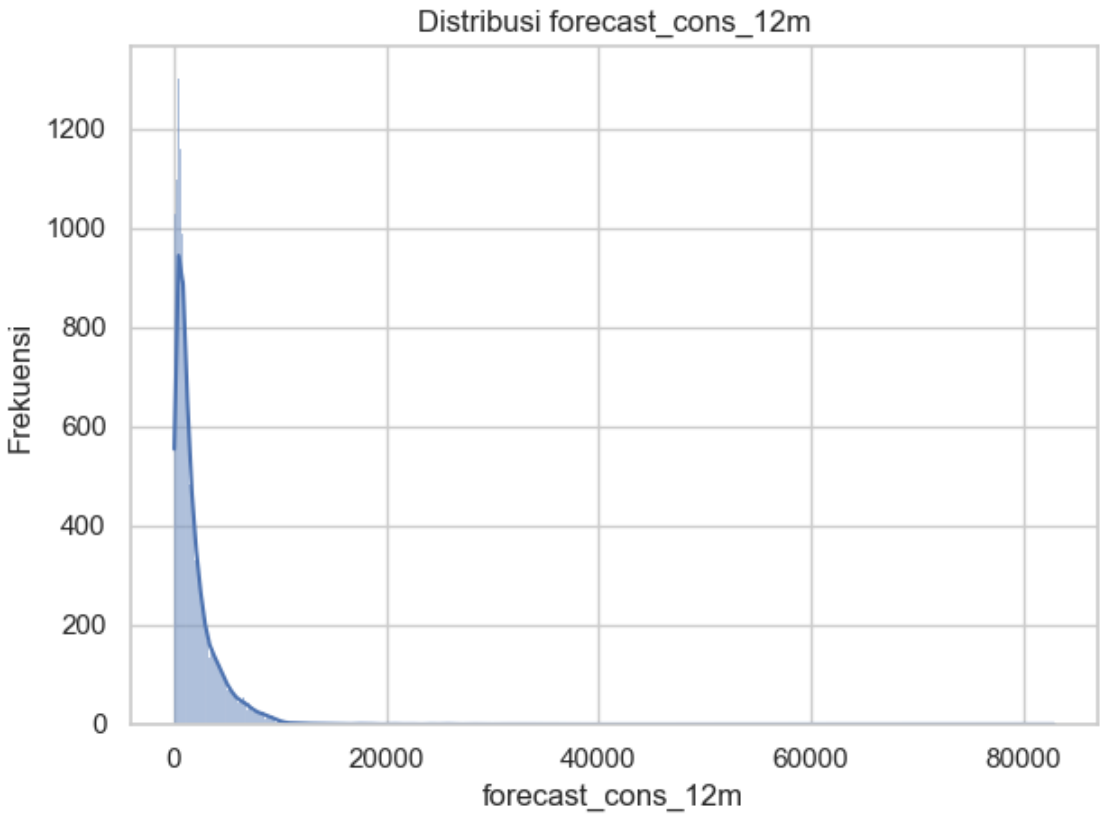
[27]:
```
numeric_cols = [
    "cons_12m", "cons_gas_12m", "cons_last_month",
    "imp_cons", "forecast_cons_12m", "forecast_cons_year",
    "margin_gross_pow_ele", "net_margin",
    "pow_max", "num_years_antig", "tenure_years"
]

# Bisa pakai log-scale untuk beberapa kolom yang sangat skewed
for col in numeric_cols:
    plt.figure()
    sns.histplot(client[col], kde=True)
    plt.title(f"Distribusi {col}")
    plt.xlabel(col)
    plt.ylabel("Frekuensi")
    plt.tight_layout()
    plt.show()
```

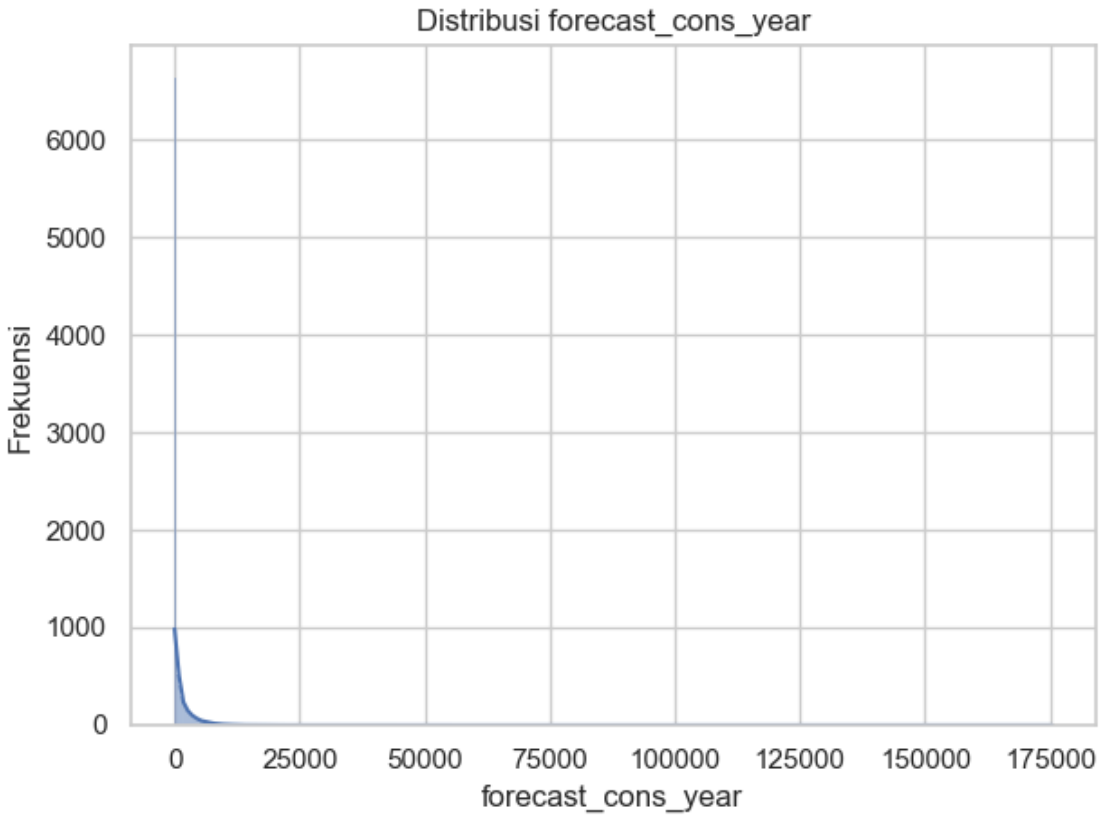

Distribusi cons_12m
Frekuensi
2000
1750
1500
1250
1000
750
500
250
0
0
1
2
3
4
5
6
cons_12m
1e6

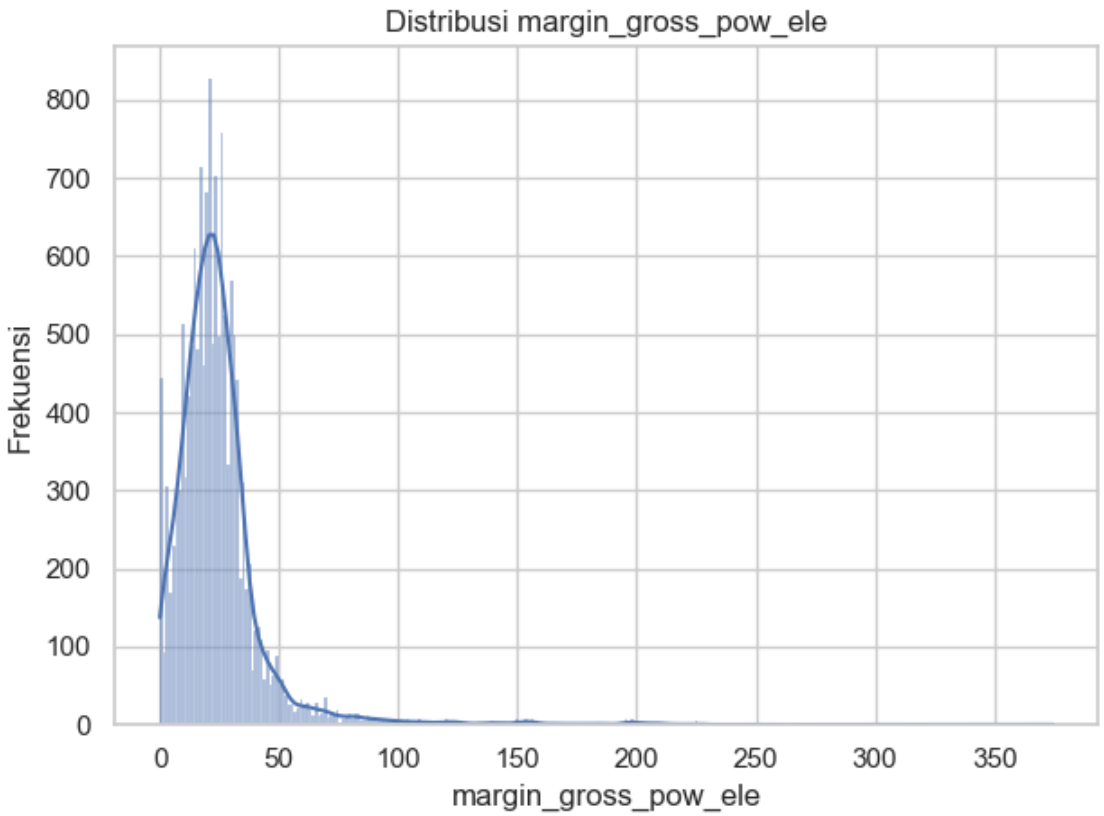

Distribusi cons_gas_12m
Frekuensi
60000
50000
40000
30000
20000
10000
0
0
1
2
3
4
cons_gas_12m
1e6

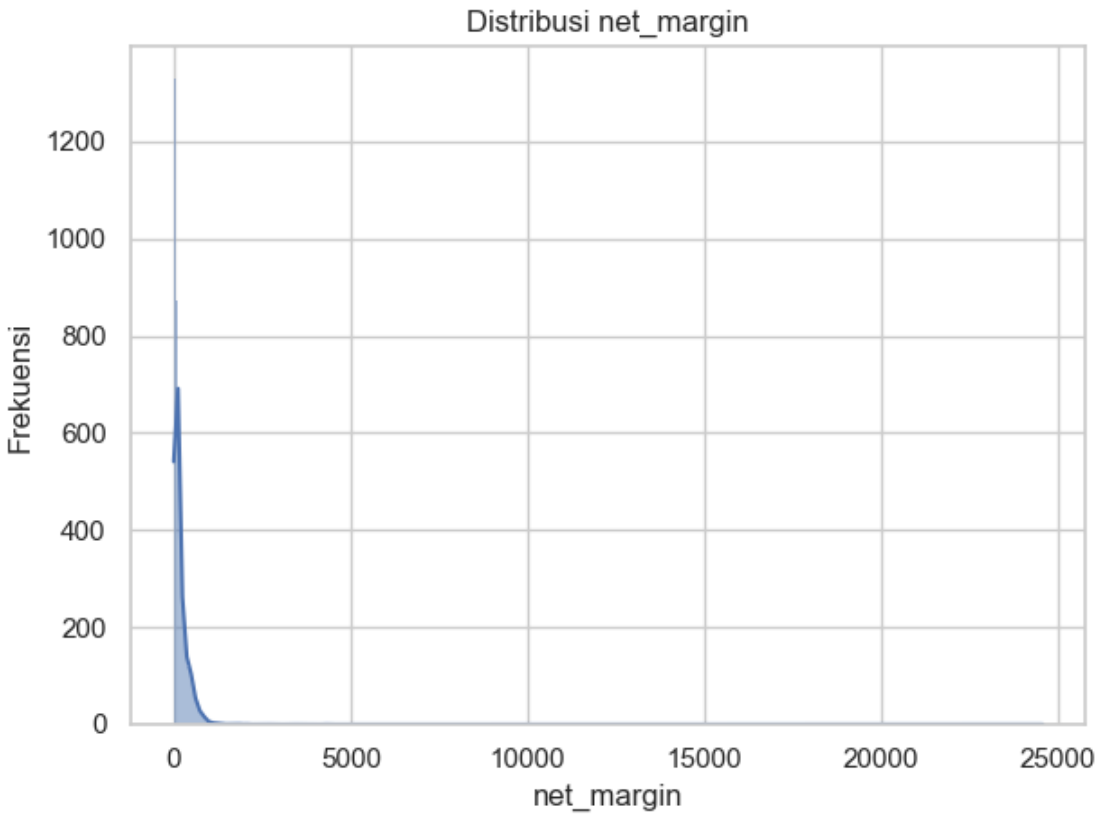

Distribusi cons_last_month
Frekuensi
6000
5000
4000
3000
2000
1000
0
0
100000
200000
300000
400000
500000
600000
700000
800000
cons_last_month

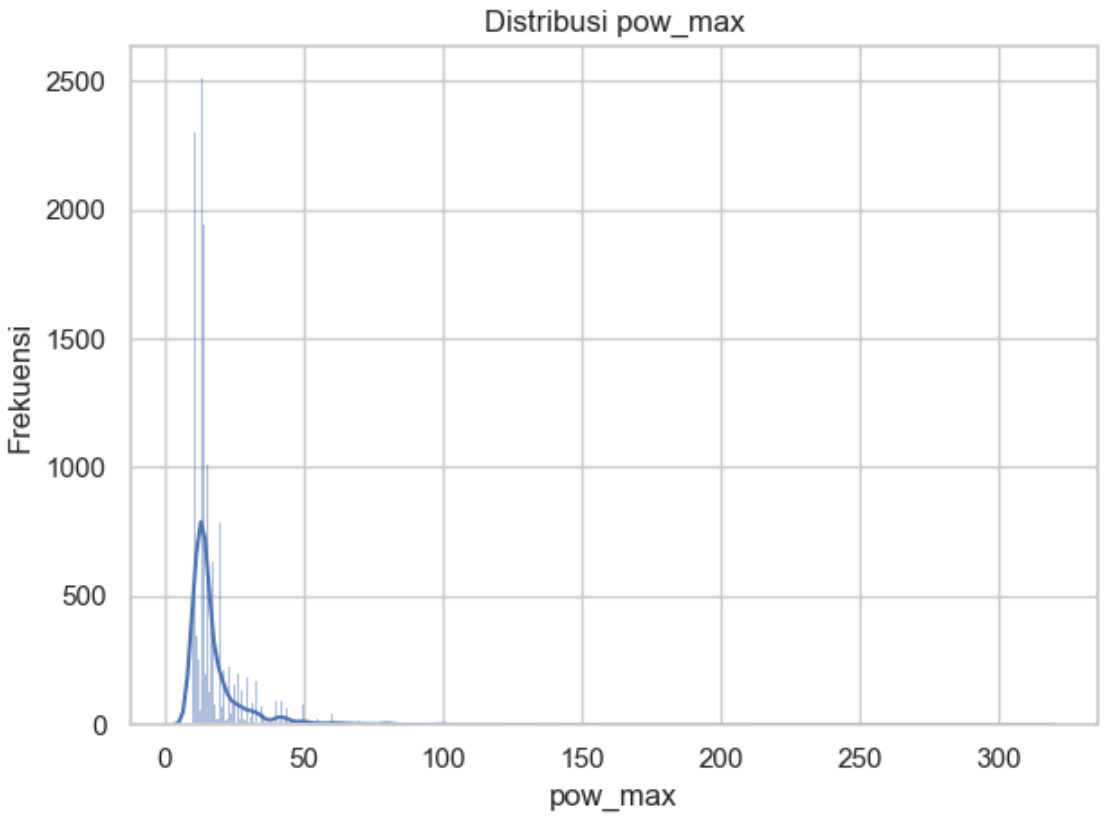

Distribusi imp_cons
Frekuensi
6000
5000
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
12000
14000
imp_cons

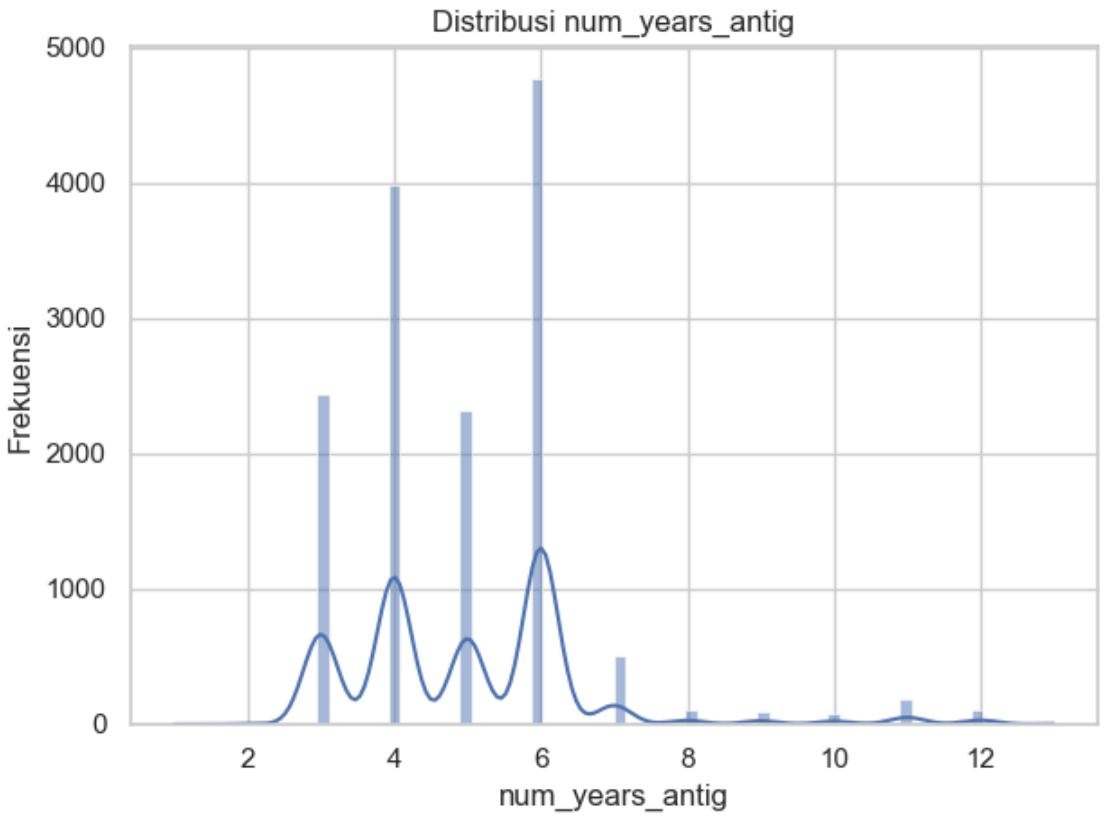

Distribusi forecast_cons_12m
Frekuensi
1200
1000
800
600
400
200
0
0
20000
40000
60000
80000
forecast_cons_12m

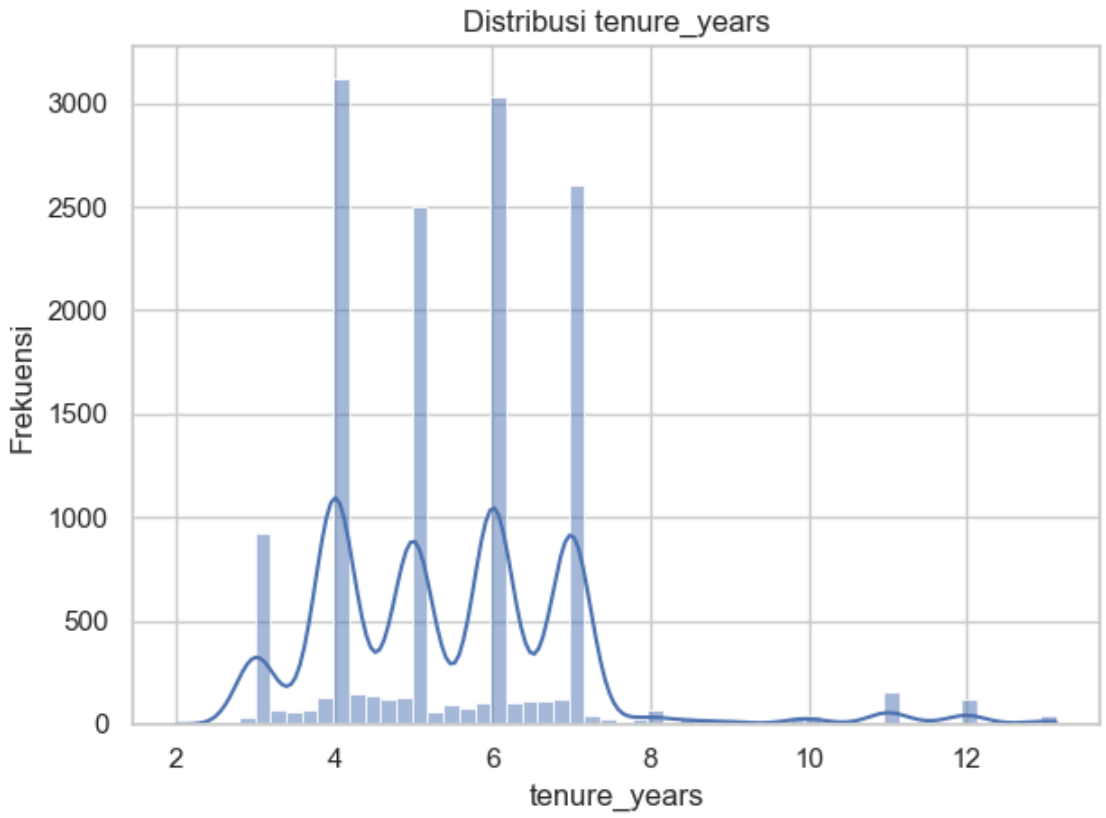

Distribusi forecast_cons_year
Frekuensi
6000
5000
4000
3000
2000
1000
0
0
25000
50000
75000
100000
125000
150000
175000
forecast_cons_year


Distribusi margin_gross_pow_ele
Frekuensi
0
100
200
300
400
500
600
700
800
0
50
100
150
200
250
300
350
margin_gross_pow_ele

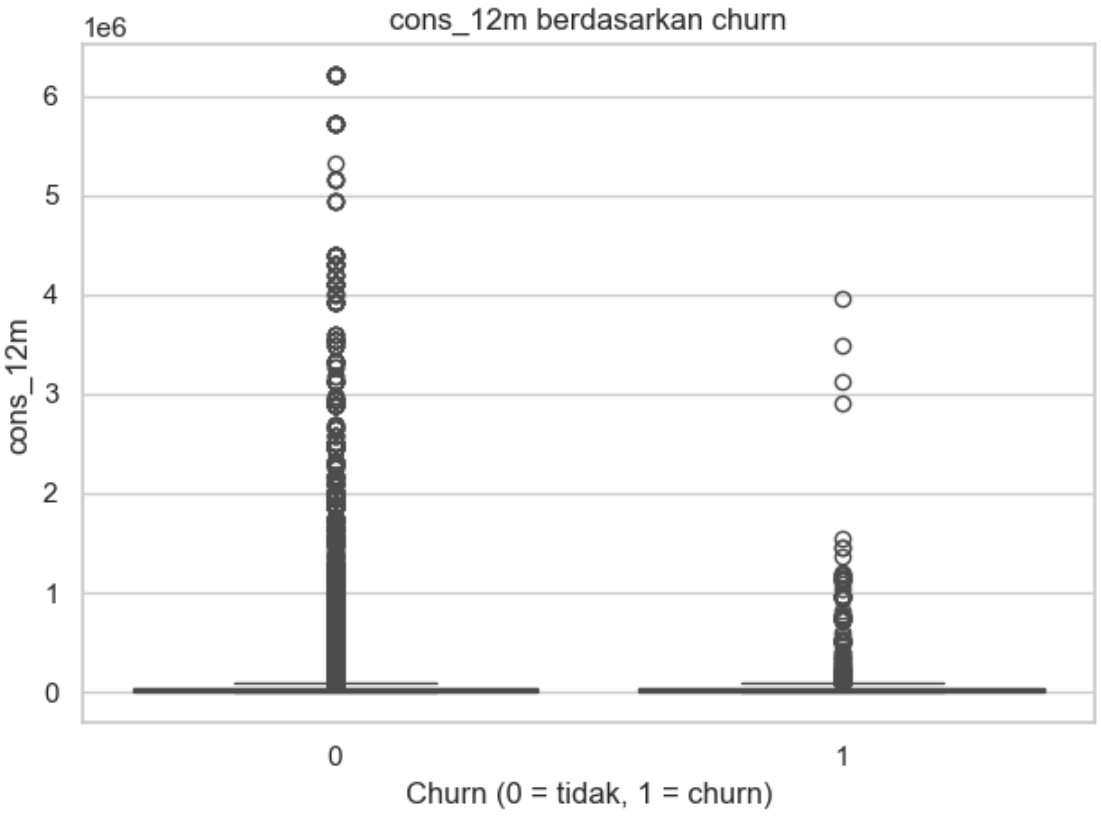

Distribusi net_margin
Frekuensi
1200
1000
800
600
400
200
0
0
5000
10000
15000
20000
25000
net_margin

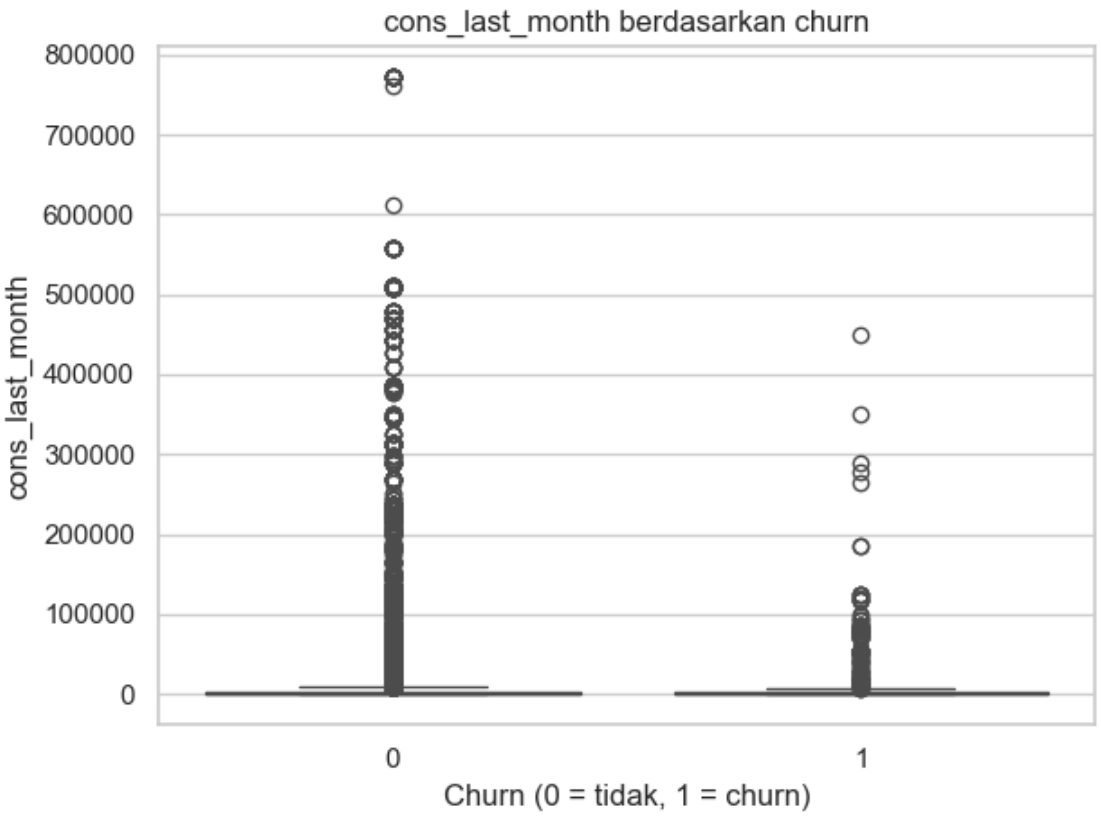

Distribusi pow_max
Frekuensi
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300
pow_max

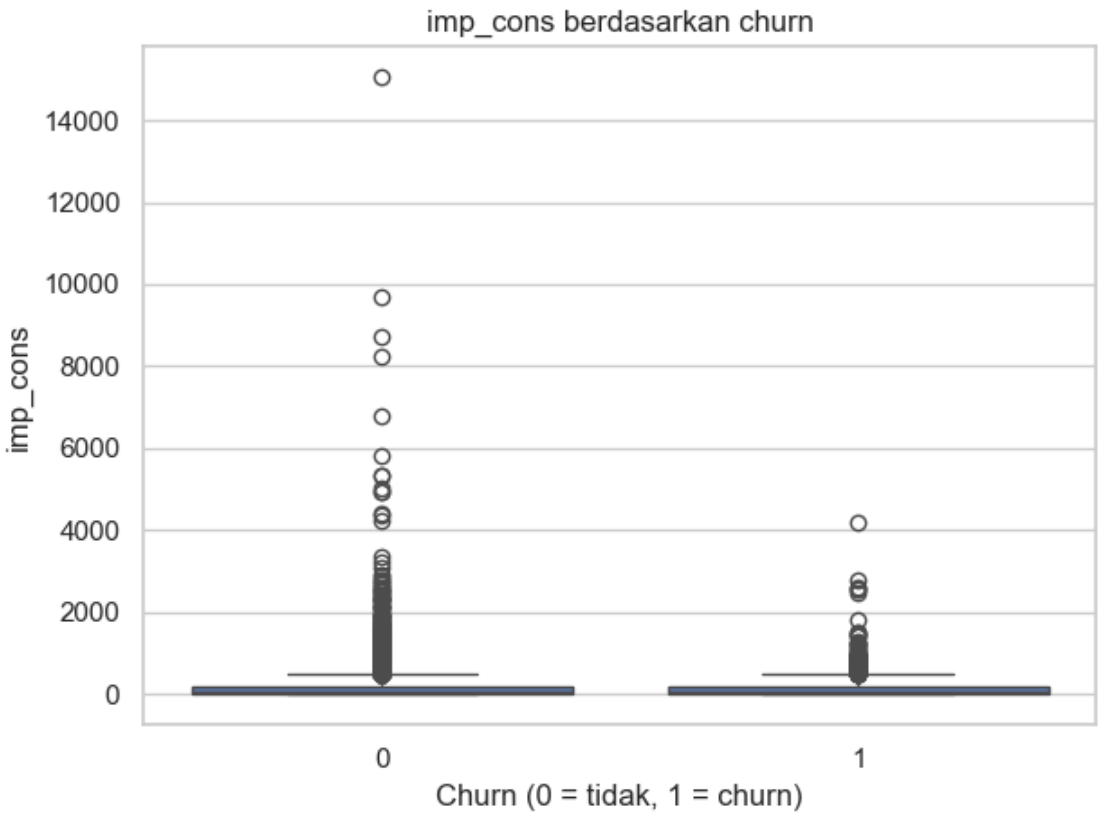

Distribusi num_years_antig
5000
4000
3000
2000
1000
0
Frekuensi
2
4
6
8
10
12
num_years_antig

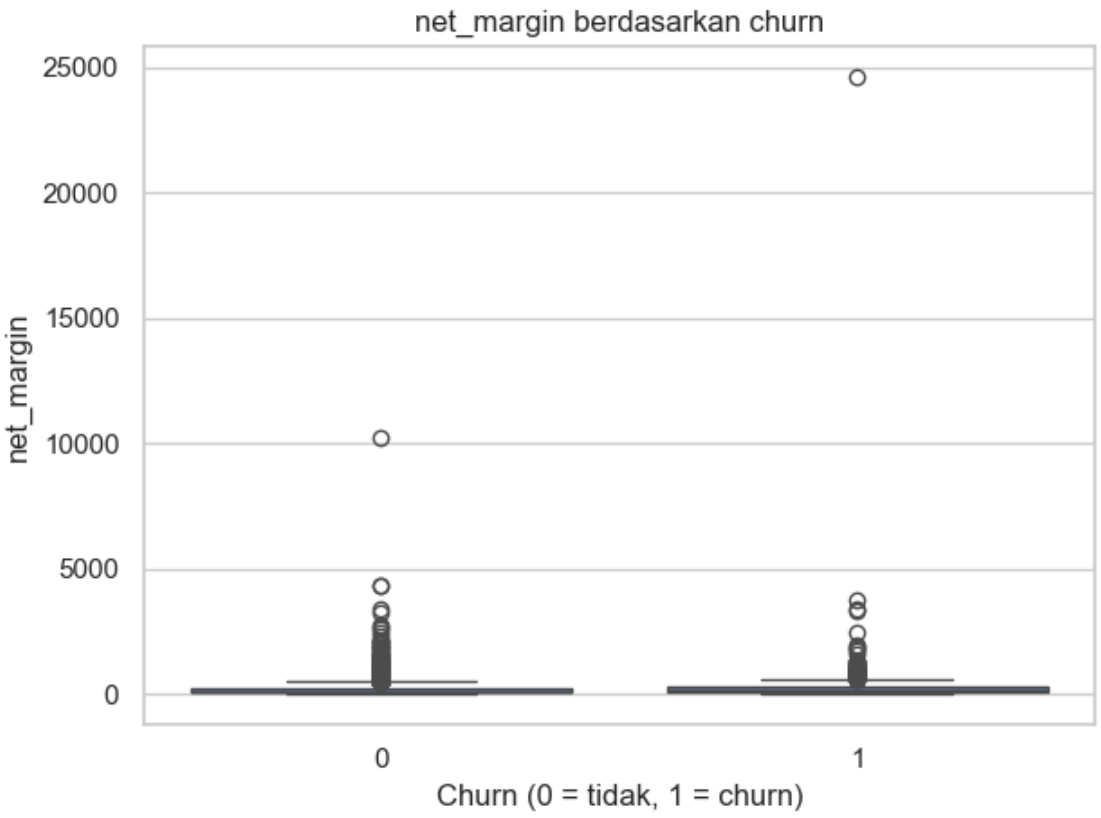


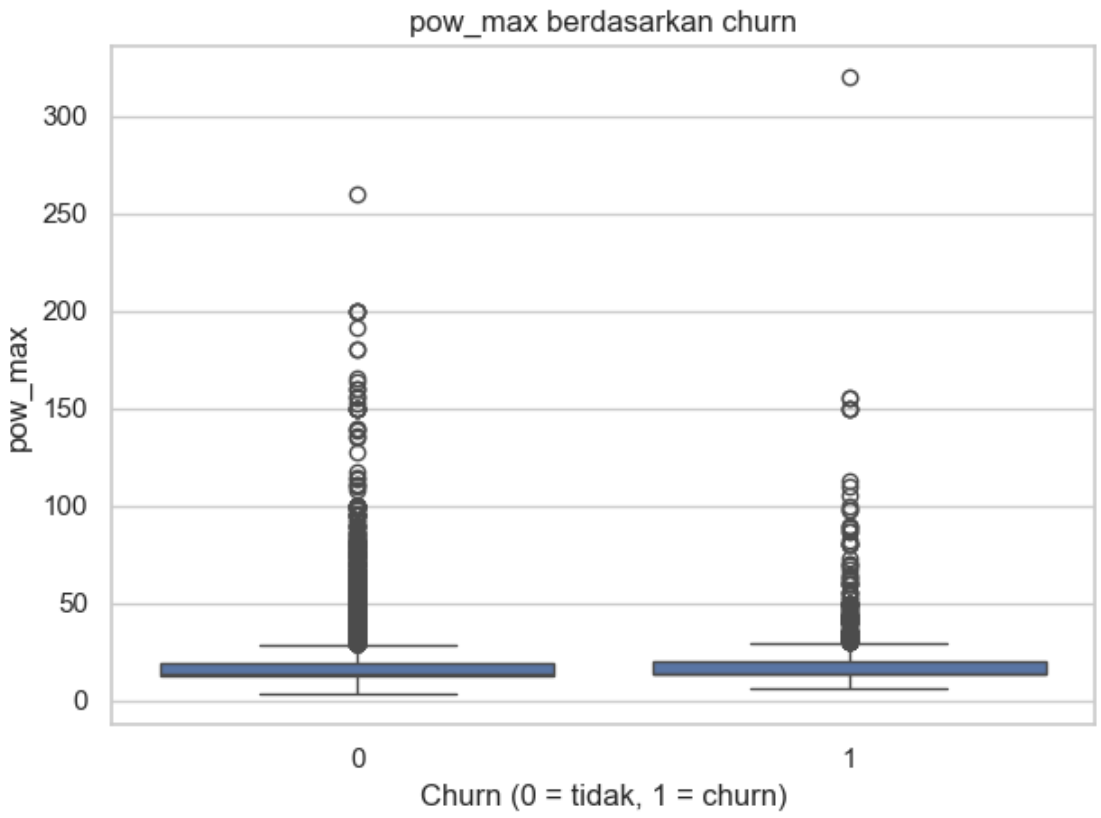
[29]:
```
box_cols = ["cons_12m", "cons_last_month", "imp_cons",
            "net_margin", "pow_max"]

for col in box_cols:
    plt.figure()
    sns.boxplot(data=client, x="churn", y=col)
    plt.title(f"{col} berdasarkan churn")
    plt.xlabel("Churn (0 = tidak, 1 = churn)")
    plt.ylabel(col)
    plt.tight_layout()
    plt.show()
```


cons_12m berdasarkan churn
1e6
6
5
4
3
2
1
0
cons_12m
0
1
Churn (0 = tidak, 1 = churn)

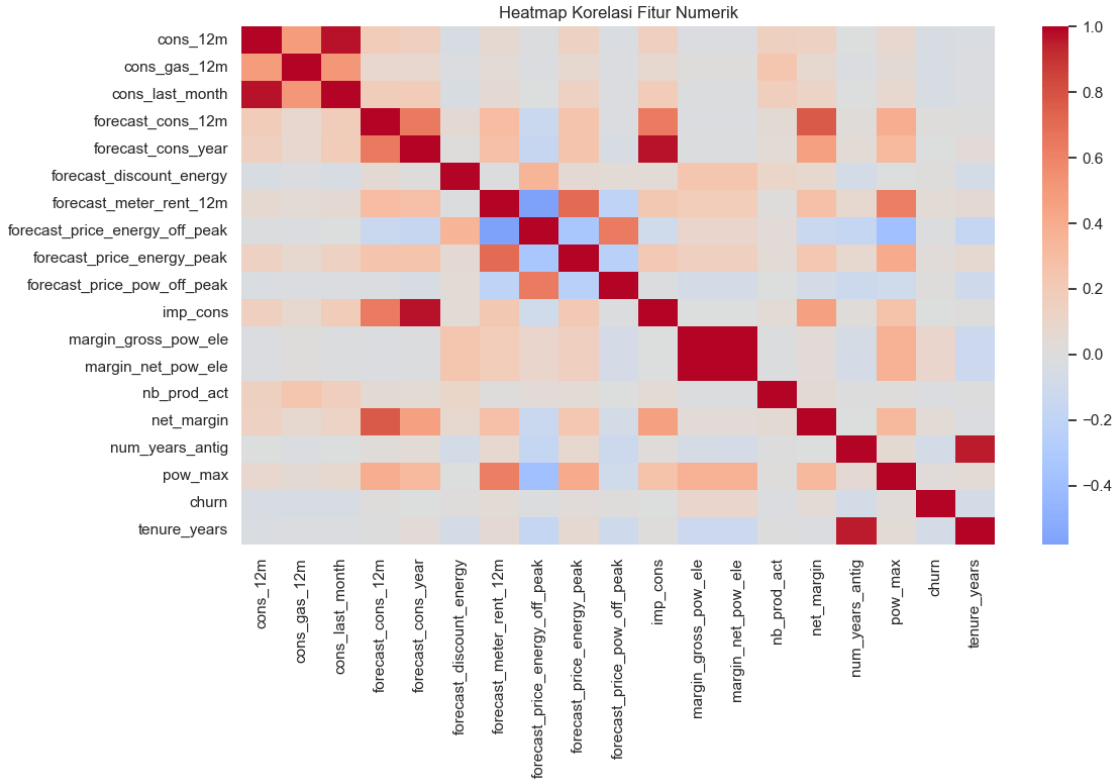

cons_last_month berdasarkan churn
800000
700000
600000
500000
400000
300000
200000
100000
0
cons_last_month
0
1
Churn (0 = tidak, 1 = churn)


imp_cons berdasarkan churn
imp_cons
14000
12000
10000
8000
6000
4000
2000
0
0
1
Churn (0 = tidak, 1 = churn)


net_margin berdasarkan churn
25000
20000
15000
10000
5000
0
net_margin
0
1
Churn (0 = tidak, 1 = churn)

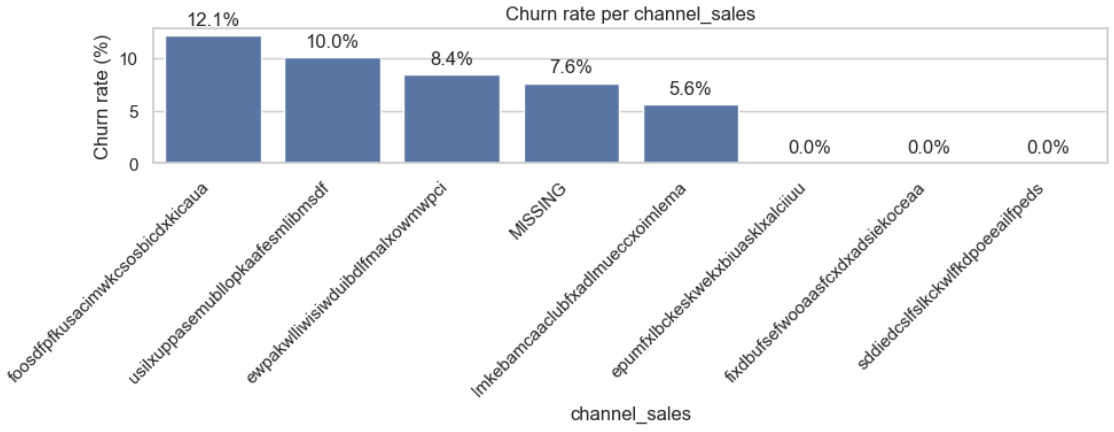


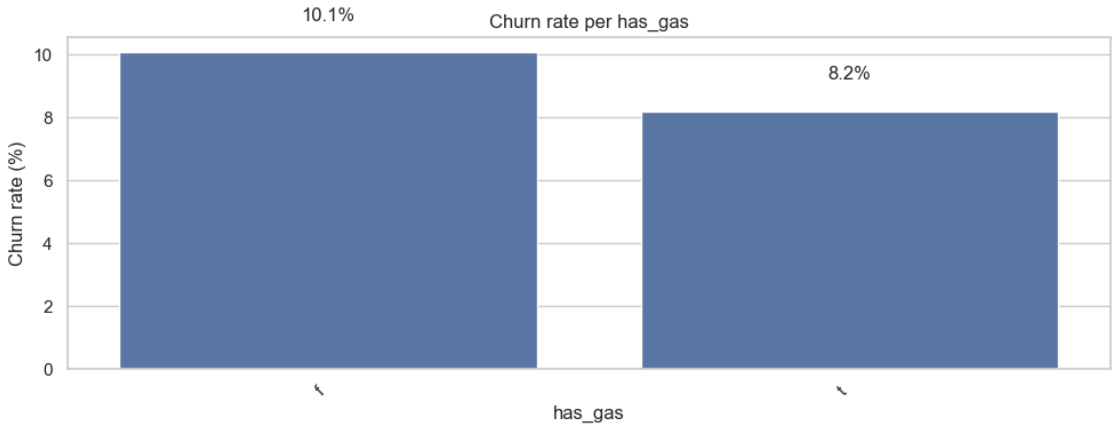
[30]:
```
num_cols_for_corr = client.select_dtypes(include=["int64", "float64"]).columns

corr = client[num_cols_for_corr].corr()

plt.figure(figsize=(12, 8))
sns.heatmap(corr, annot=False, cmap="coolwarm", center=0)
plt.title("Heatmap Korelasi Fitur Numerik")
plt.tight_layout()
plt.show()

# Jika mau fokus ke korelasi dengan churn saja:
churn_corr = corr["churn"].sort_values(ascending=False)
print(churn_corr)
```

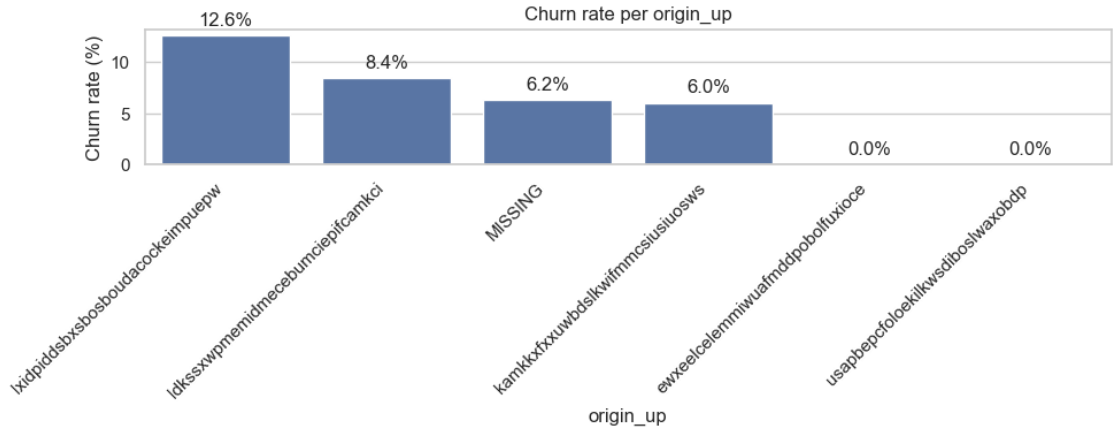



```
churn                            1.000000
margin_net_pow_ele               0.095772
margin_gross_pow_ele             0.095725
forecast_meter_rent_12m          0.044245
net_margin                       0.041135
pow_max                          0.030362
forecast_price_energy_peak       0.029315
forecast_discount_energy         0.017026
forecast_price_pow_off_peak      0.014778
forecast_cons_12m                0.012949
imp_cons                        -0.001583
forecast_cons_year              -0.002558
forecast_price_energy_off_peak  -0.010837
nb_prod_act                     -0.014930
cons_gas_12m                    -0.037957
cons_last_month                 -0.045284
cons_12m                        -0.045968
tenure_years                    -0.073919
num_years_antig                 -0.074140
Name: churn, dtype: float64
```

[31]:
```
cat_cols = ["channel_sales", "has_gas", "origin_up"]

for col in cat_cols:
    plt.figure(figsize=(10, 4))

    # Hitung churn rate per kategori
    grp = client.groupby(col)["churn"].mean().sort_values(ascending=False) * 100

    sns.barplot(x=grp.index, y=grp.values)
    plt.xticks(rotation=45, ha="right")
    plt.ylabel("Churn rate (%)")
    plt.xlabel(col)
    plt.title(f"Churn rate per {col}")

    # Tampilkan label persen di atas bar
    for i, v in enumerate(grp.values):
        plt.text(i, v + 1, f"{v:.1f}%", ha="center")

    plt.tight_layout()
    plt.show()
```







1.3 Feature Engineering

[32]:
```
price["price_date"] = pd.to_datetime(price["price_date"], format='%Y-%m-%d')
```

[34]:
```
price
```

[34]:

	id	price_date	price_off_peak_var	\
0	038af19179925da21a25619c5a24b745	2015-01-01	0.151367	
1	038af19179925da21a25619c5a24b745	2015-02-01	0.151367	
2	038af19179925da21a25619c5a24b745	2015-03-01	0.151367	
3	038af19179925da21a25619c5a24b745	2015-04-01	0.149626	
4	038af19179925da21a25619c5a24b745	2015-05-01	0.149626	
…	…	…	…	
192997	16f51cdc2baa19af0b940ee1b3dd17d5	2015-08-01	0.119916	
192998	16f51cdc2baa19af0b940ee1b3dd17d5	2015-09-01	0.119916	
192999	16f51cdc2baa19af0b940ee1b3dd17d5	2015-10-01	0.119916	

```
193000  16f51cdc2baa19af0b940ee1b3dd17d5 2015-11-01            0.119916
193001  16f51cdc2baa19af0b940ee1b3dd17d5 2015-12-01            0.119916

        price_peak_var  price_mid_peak_var  price_off_peak_fix  \
0             0.000000            0.000000           44.266931
1             0.000000            0.000000           44.266931
2             0.000000            0.000000           44.266931
3             0.000000            0.000000           44.266931
4             0.000000            0.000000           44.266931
…                ...                 ...                 …
192997        0.102232            0.076257           40.728885
192998        0.102232            0.076257           40.728885
192999        0.102232            0.076257           40.728885
193000        0.102232            0.076257           40.728885
193001        0.102232            0.076257           40.728885

        price_peak_fix  price_mid_peak_fix
0              0.00000            0.000000
1              0.00000            0.000000
2              0.00000            0.000000
3              0.00000            0.000000
4              0.00000            0.000000
…                 ...                 ...
192997        24.43733           16.291555
192998        24.43733           16.291555
192999        24.43733           16.291555
193000        24.43733           16.291555
193001        24.43733           16.291555

[193002 rows x 8 columns]
```

[35]:
```
# Group off-peak prices by companies and month
monthly_price_by_id = price.groupby(['id', 'price_date']).
 ↪agg({'price_off_peak_var': 'mean', 'price_off_peak_fix': 'mean'}).
 ↪reset_index()

# Get january and december prices
jan_prices = monthly_price_by_id.groupby('id').first().reset_index()
dec_prices = monthly_price_by_id.groupby('id').last().reset_index()

# Calculate the difference
diff = pd.merge(dec_prices.rename(columns={'price_off_peak_var': 'dec_1',␣
 ↪'price_off_peak_fix': 'dec_2'}), jan_prices.drop(columns='price_date'),␣
 ↪on='id')
diff['offpeak_diff_dec_january_energy'] = diff['dec_1'] -␣
 ↪diff['price_off_peak_var']
```

```
diff['offpeak_diff_dec_january_power'] = diff['dec_2'] -␣
 ↪diff['price_off_peak_fix']
diff = diff[['id',␣
 ↪'offpeak_diff_dec_january_energy','offpeak_diff_dec_january_power']]
diff
```

[35]:

	id	offpeak_diff_dec_january_energy	offpeak_diff_dec_january_power
0	0002203ffbb812588b632b9e628cc38d	-0.006192	0.162916
1	0004351ebdd665e6ee664792efc4fd13	-0.004104	0.177779
2	0010bcc39e42b3c2131ed2ce55246e3c	0.050443	1.500000
3	0010ee3855fdea87602a5b7aba8e42de	-0.010018	0.162916
4	00114d74e963e47177db89bc70108537	-0.003994	-0.000001
…	…	…	…
16091	ffef185810e44254c3a4c6395e6b4d8a	-0.050232	-0.335085
16092	fffac626da707b1b5ab11e8431a4d0a2	-0.003778	0.177779
16093	fffc0cacd305dd51f316424bbb08d1bd	-0.001760	0.164916
16094	fffe4f5646aa39c7f97f95ae2679ce64	-0.009391	0.162916
16095	ffff7fa066f1fb305ae285bb03bf325a	-0.009528	0.162916

[16096 rows x 3 columns]

[36]:
```
monthly_price_by_id
```

[36]:

	id	price_date	price_off_peak_var	\
0	0002203ffbb812588b632b9e628cc38d	2015-01-01	0.126098	
1	0002203ffbb812588b632b9e628cc38d	2015-02-01	0.126098	
2	0002203ffbb812588b632b9e628cc38d	2015-03-01	0.128067	
3	0002203ffbb812588b632b9e628cc38d	2015-04-01	0.128067	
4	0002203ffbb812588b632b9e628cc38d	2015-05-01	0.128067	
…	…	…	…	
192997	ffff7fa066f1fb305ae285bb03bf325a	2015-08-01	0.119916	
192998	ffff7fa066f1fb305ae285bb03bf325a	2015-09-01	0.119916	
192999	ffff7fa066f1fb305ae285bb03bf325a	2015-10-01	0.119916	
193000	ffff7fa066f1fb305ae285bb03bf325a	2015-11-01	0.119916	

```
193001  ffff7fa066f1fb305ae285bb03bf325a 2015-12-01            0.119916

        price_off_peak_fix
0                40.565969
1                40.565969
2                40.728885
3                40.728885
4                40.728885
…                      …
192997           40.728885
192998           40.728885
192999           40.728885
193000           40.728885
193001           40.728885

[193002 rows x 4 columns]
```

[37]:
```
df = pd.merge(client, diff, on='id')
df
```

[37]:
```
                                    id                    channel_sales  \
0      24011ae4ebbe3035111d65fa7c15bc57  foosdfpfkusacimwkcsosbicdxkicaua
1      d29c2c54acc38ff3c0614d0a653813dd                           MISSING
2      764c75f661154dac3a6c254cd082ea7d  foosdfpfkusacimwkcsosbicdxkicaua
3      bba03439a292a1e166f80264c16191cb  lmkebamcaaclubfxadlmueccxoimlema
4      149d57cf92fc41cf94415803a877cb4b                           MISSING
…                                   …                                 …
14601  18463073fb097fc0ac5d3e040f356987  foosdfpfkusacimwkcsosbicdxkicaua
14602  d0a6f71671571ed83b2645d23af6de00  foosdfpfkusacimwkcsosbicdxkicaua
14603  10e6828ddd62cbcf687cb74928c4c2d2  foosdfpfkusacimwkcsosbicdxkicaua
14604  1cf20fd6206d7678d5bcafd28c53b4db  foosdfpfkusacimwkcsosbicdxkicaua
14605  563dde550fd624d7352f3de77c0cdfcd                           MISSING

       cons_12m  cons_gas_12m  cons_last_month date_activ   date_end  \
0             0         54946                0 2013-06-15 2016-06-15
1          4660             0                0 2009-08-21 2016-08-30
2           544             0                0 2010-04-16 2016-04-16
3          1584             0                0 2010-03-30 2016-03-30
4          4425             0              526 2010-01-13 2016-03-07
…             …             …                …          …          …
14601     32270         47940                0 2012-05-24 2016-05-08
14602      7223             0              181 2012-08-27 2016-08-27
14603      1844             0              179 2012-02-08 2016-02-07
14604       131             0                0 2012-08-30 2016-08-30
14605      8730             0                0 2009-12-18 2016-12-17

      date_modif_prod date_renewal  forecast_cons_12m  …  \
```

```
0          2015-11-01   2015-06-23              0.00  …
1          2009-08-21   2015-08-31            189.95  …
2          2010-04-16   2015-04-17             47.96  …
3          2010-03-30   2015-03-31            240.04  …
4          2010-01-13   2015-03-09            445.75  …
…               …            …                 …  …
14601      2015-05-08   2014-05-26           4648.01  …
14602      2012-08-27   2015-08-28            631.69  …
14603      2012-02-08   2015-02-09            190.39  …
14604      2012-08-30   2015-08-31             19.34  …
14605      2009-12-18   2015-12-21            762.41  …

       margin_net_pow_ele  nb_prod_act  net_margin  num_years_antig  \
0                   25.44            2      678.99                3
1                   16.38            1       18.89                6
2                   28.60            1        6.60                6
3                   30.22            1       25.46                6
4                   44.91            1       47.98                6
…                   …          …         …               …
14601               27.88            2      381.77                4
14602                0.00            1       90.34                3
14603               39.84            1       20.38                4
14604               13.08            1        0.96                3
14605               11.84            1       96.34                6

                              origin_up  pow_max churn  tenure_years  \
0      lxidpiddsbxsbosboudacockeimpuepw   43.648     1      3.000684
1      kamkkxfxxuwbdslkwifmmcsiusiuosws   13.800     0      7.025325
2      kamkkxfxxuwbdslkwifmmcsiusiuosws   13.856     0      6.001369
3      kamkkxfxxuwbdslkwifmmcsiusiuosws   13.200     0      6.001369
4      kamkkxfxxuwbdslkwifmmcsiusiuosws   19.800     0      6.146475
…                                   …      …   …           …
14601  lxidpiddsbxsbosboudacockeimpuepw   15.000     0      3.956194
14602  lxidpiddsbxsbosboudacockeimpuepw    6.000     1      4.000000
14603  lxidpiddsbxsbosboudacockeimpuepw   15.935     1      3.997262
14604  lxidpiddsbxsbosboudacockeimpuepw   11.000     0      4.000000
14605  ldkssxwpmemidmecebumciepifcamkci   10.392     0      6.997947

       offpeak_diff_dec_january_energy  offpeak_diff_dec_january_power
0                             0.020057                        3.700961
1                            -0.003767                        0.177779
2                            -0.004670                        0.177779
3                            -0.004547                        0.177779
4                            -0.006192                        0.162916
…                                …                               …
14601                        -0.008653                        0.177779
14602                        -0.007395                        0.236694
```

```
14603                        -0.006192                        0.162916
14604                        -0.003767                        0.177779
14605                        -0.004628                       -0.000001

[14606 rows x 29 columns]
```

[ ]:
```
mean_prices = price.groupby(['id']).agg({
    'price_off_peak_var': 'mean',
    'price_peak_var': 'mean',
    'price_mid_peak_var': 'mean',
    'price_off_peak_fix': 'mean',
    'price_peak_fix': 'mean',
    'price_mid_peak_fix': 'mean'
}).reset_index()
```

[ ]:
```
mean_prices['off_peak_peak_var_mean_diff'] = mean_prices['price_off_peak_var']␣
 ↪- mean_prices['price_peak_var']
mean_prices['peak_mid_peak_var_mean_diff'] = mean_prices['price_peak_var'] -␣
 ↪mean_prices['price_mid_peak_var']
mean_prices['off_peak_mid_peak_var_mean_diff'] =␣
 ↪mean_prices['price_off_peak_var'] - mean_prices['price_mid_peak_var']
mean_prices['off_peak_peak_fix_mean_diff'] = mean_prices['price_off_peak_fix']␣
 ↪- mean_prices['price_peak_fix']
mean_prices['peak_mid_peak_fix_mean_diff'] = mean_prices['price_peak_fix'] -␣
 ↪mean_prices['price_mid_peak_fix']
mean_prices['off_peak_mid_peak_fix_mean_diff'] =␣
 ↪mean_prices['price_off_peak_fix'] - mean_prices['price_mid_peak_fix']
```

[40]:
```
columns = [
    'id',
    'off_peak_peak_var_mean_diff',
    'peak_mid_peak_var_mean_diff',
    'off_peak_mid_peak_var_mean_diff',
    'off_peak_peak_fix_mean_diff',
    'peak_mid_peak_fix_mean_diff',
    'off_peak_mid_peak_fix_mean_diff'
]
df = pd.merge(df, mean_prices[columns], on='id')
df
```

[40]:
```
                                    id                    channel_sales  \
0      24011ae4ebbe3035111d65fa7c15bc57  foosdfpfkusacimwkcsosbicdxkicaua
1      d29c2c54acc38ff3c0614d0a653813dd                           MISSING
2      764c75f661154dac3a6c254cd082ea7d  foosdfpfkusacimwkcsosbicdxkicaua
3      bba03439a292a1e166f80264c16191cb  lmkebamcaaclubfxadlmueccxoimlema
4      149d57cf92fc41cf94415803a877cb4b                           MISSING
…                                    …                                …
```

```
14601  18463073fb097fc0ac5d3e040f356987  foosdfpfkusacimwkcsosbicdxkicaua
14602  d0a6f71671571ed83b2645d23af6de00  foosdfpfkusacimwkcsosbicdxkicaua
14603  10e6828ddd62cbcf687cb74928c4c2d2  foosdfpfkusacimwkcsosbicdxkicaua
14604  1cf20fd6206d7678d5bcafd28c53b4db  foosdfpfkusacimwkcsosbicdxkicaua
14605  563dde550fd624d7352f3de77c0cdfcd                           MISSING

       cons_12m  cons_gas_12m  cons_last_month date_activ   date_end  \
0             0         54946                0 2013-06-15 2016-06-15
1          4660             0                0 2009-08-21 2016-08-30
2           544             0                0 2010-04-16 2016-04-16
3          1584             0                0 2010-03-30 2016-03-30
4          4425             0              526 2010-01-13 2016-03-07
…             …             …                …          …          …
14601     32270         47940                0 2012-05-24 2016-05-08
14602      7223             0              181 2012-08-27 2016-08-27
14603      1844             0              179 2012-02-08 2016-02-07
14604       131             0                0 2012-08-30 2016-08-30
14605      8730             0                0 2009-12-18 2016-12-17

      date_modif_prod date_renewal  forecast_cons_12m  …  churn  \
0          2015-11-01   2015-06-23               0.00  …      1
1          2009-08-21   2015-08-31             189.95  …      0
2          2010-04-16   2015-04-17              47.96  …      0
3          2010-03-30   2015-03-31             240.04  …      0
4          2010-01-13   2015-03-09             445.75  …      0
…                 …            …                  …  …    …
14601      2015-05-08   2014-05-26            4648.01  …      0
14602      2012-08-27   2015-08-28             631.69  …      1
14603      2012-02-08   2015-02-09             190.39  …      1
14604      2012-08-30   2015-08-31              19.34  …      0
14605      2009-12-18   2015-12-21             762.41  …      0

       tenure_years  offpeak_diff_dec_january_energy  \
0          3.000684                         0.020057
1          7.025325                        -0.003767
2          6.001369                        -0.004670
3          6.001369                        -0.004547
4          6.146475                        -0.006192
…             …                                …
14601      3.956194                        -0.008653
14602      4.000000                        -0.007395
14603      3.997262                        -0.006192
14604      4.000000                        -0.003767
14605      6.997947                        -0.004628

       offpeak_diff_dec_january_power  off_peak_peak_var_mean_diff  \
0                            3.700961                     0.024038
```

```
1                          0.177779                       0.142485
2                          0.177779                       0.082090
3                          0.177779                       0.151210
4                          0.162916                       0.020536
…                             …                            …
14601                      0.177779                       0.144124
14602                      0.236694                       0.011393
14603                      0.162916                       0.020545
14604                      0.177779                       0.142485
14605                     -0.000001                       0.081317

       peak_mid_peak_var_mean_diff off_peak_mid_peak_var_mean_diff  \
0                         0.034219                         0.058257
1                         0.007124                         0.149609
2                         0.088421                         0.170512
3                         0.000000                         0.151210
4                         0.030773                         0.051309
…                            …                             …
14601                     0.000000                         0.144124
14602                     0.024589                         0.035982
14603                     0.030633                         0.051178
14604                     0.007124                         0.149609
14605                     0.087344                         0.168662

       off_peak_peak_fix_mean_diff  peak_mid_peak_fix_mean_diff  \
0                        18.590255                     7.450670
1                        44.311375                     0.000000
2                        44.385450                     0.000000
3                        44.400265                     0.000000
4                        16.275263                     8.137629
…                           …                           …
14601                    44.370635                     0.000000
14602                    22.622294                    28.047961
14603                    16.280694                     8.140345
14604                    44.311375                     0.000000
14605                    44.266930                     0.000000

       off_peak_mid_peak_fix_mean_diff
0                         26.040925
1                         44.311375
2                         44.385450
3                         44.400265
4                         24.412893
…                            …
14601                     44.370635
14602                     50.670256
14603                     24.421038
```

```
14604                        44.311375
14605                        44.266930

[14606 rows x 35 columns]
```

[42]:
```
mean_prices_by_month = price.groupby(['id', 'price_date']).agg({
    'price_off_peak_var': 'mean',
    'price_peak_var': 'mean',
    'price_mid_peak_var': 'mean',
    'price_off_peak_fix': 'mean',
    'price_peak_fix': 'mean',
    'price_mid_peak_fix': 'mean'
}).reset_index()
```

[43]:
```
mean_prices_by_month['off_peak_peak_var_mean_diff'] =␣
 ↪mean_prices_by_month['price_off_peak_var'] -␣
 ↪mean_prices_by_month['price_peak_var']
mean_prices_by_month['peak_mid_peak_var_mean_diff'] =␣
 ↪mean_prices_by_month['price_peak_var'] -␣
 ↪mean_prices_by_month['price_mid_peak_var']
mean_prices_by_month['off_peak_mid_peak_var_mean_diff'] =␣
 ↪mean_prices_by_month['price_off_peak_var'] -␣
 ↪mean_prices_by_month['price_mid_peak_var']
mean_prices_by_month['off_peak_peak_fix_mean_diff'] =␣
 ↪mean_prices_by_month['price_off_peak_fix'] -␣
 ↪mean_prices_by_month['price_peak_fix']
mean_prices_by_month['peak_mid_peak_fix_mean_diff'] =␣
 ↪mean_prices_by_month['price_peak_fix'] -␣
 ↪mean_prices_by_month['price_mid_peak_fix']
mean_prices_by_month['off_peak_mid_peak_fix_mean_diff'] =␣
 ↪mean_prices_by_month['price_off_peak_fix'] -␣
 ↪mean_prices_by_month['price_mid_peak_fix']
```

[44]:
```
max_diff_across_periods_months = mean_prices_by_month.groupby(['id']).agg({
    'off_peak_peak_var_mean_diff': 'max',
    'peak_mid_peak_var_mean_diff': 'max',
    'off_peak_mid_peak_var_mean_diff': 'max',
    'off_peak_peak_fix_mean_diff': 'max',
    'peak_mid_peak_fix_mean_diff': 'max',
    'off_peak_mid_peak_fix_mean_diff': 'max'
}).reset_index().rename(
    columns={
        'off_peak_peak_var_mean_diff': 'off_peak_peak_var_max_monthly_diff',
        'peak_mid_peak_var_mean_diff': 'peak_mid_peak_var_max_monthly_diff',
        'off_peak_mid_peak_var_mean_diff':␣
 ↪'off_peak_mid_peak_var_max_monthly_diff',
        'off_peak_peak_fix_mean_diff': 'off_peak_peak_fix_max_monthly_diff',
```

```
        'peak_mid_peak_fix_mean_diff': 'peak_mid_peak_fix_max_monthly_diff',
        'off_peak_mid_peak_fix_mean_diff':␣
 ↪'off_peak_mid_peak_fix_max_monthly_diff'
    }
)
```

[45]:
```
columns = [
    'id',
    'off_peak_peak_var_max_monthly_diff',
    'peak_mid_peak_var_max_monthly_diff',
    'off_peak_mid_peak_var_max_monthly_diff',
    'off_peak_peak_fix_max_monthly_diff',
    'peak_mid_peak_fix_max_monthly_diff',
    'off_peak_mid_peak_fix_max_monthly_diff'
]

df = pd.merge(df, max_diff_across_periods_months[columns], on='id')
df.head()
```

[45]:

	id	channel_sales	\
0	24011ae4ebbe3035111d65fa7c15bc57	foosdfpfkusacimwkcsosbicdxkicaua	
1	d29c2c54acc38ff3c0614d0a653813dd	MISSING	
2	764c75f661154dac3a6c254cd082ea7d	foosdfpfkusacimwkcsosbicdxkicaua	
3	bba03439a292a1e166f80264c16191cb	lmkebamcaaclubfxadlmueccxoimlema	
4	149d57cf92fc41cf94415803a877cb4b	MISSING	

	cons_12m	cons_gas_12m	cons_last_month	date_activ	date_end	\
0	0	54946	0	2013-06-15	2016-06-15	
1	4660	0	0	2009-08-21	2016-08-30	
2	544	0	0	2010-04-16	2016-04-16	
3	1584	0	0	2010-03-30	2016-03-30	
4	4425	0	526	2010-01-13	2016-03-07	

	date_modif_prod	date_renewal	forecast_cons_12m	…	\
0	2015-11-01	2015-06-23	0.00	…	
1	2009-08-21	2015-08-31	189.95	…	
2	2010-04-16	2015-04-17	47.96	…	
3	2010-03-30	2015-03-31	240.04	…	
4	2010-01-13	2015-03-09	445.75	…	

	off_peak_mid_peak_var_mean_diff	off_peak_peak_fix_mean_diff	\
0	0.058257	18.590255	
1	0.149609	44.311375	
2	0.170512	44.385450	
3	0.151210	44.400265	
4	0.051309	16.275263	

```
   peak_mid_peak_fix_mean_diff  off_peak_mid_peak_fix_mean_diff  \
0                     7.450670                        26.040925
1                     0.000000                        44.311375
2                     0.000000                        44.385450
3                     0.000000                        44.400265
4                     8.137629                        24.412893

   off_peak_peak_var_max_monthly_diff  peak_mid_peak_var_max_monthly_diff  \
0                            0.060550                            0.085483
1                            0.151367                            0.085483
2                            0.084587                            0.089162
3                            0.153133                            0.000000
4                            0.022225                            0.033743

  off_peak_mid_peak_var_max_monthly_diff  off_peak_peak_fix_max_monthly_diff  \
0                               0.146033                           44.266930
1                               0.151367                           44.444710
2                               0.172468                           44.444710
3                               0.153133                           44.444710
4                               0.055866                           16.291555

   peak_mid_peak_fix_max_monthly_diff  off_peak_mid_peak_fix_max_monthly_diff
0                            8.145775                                44.26693
1                            0.000000                                44.44471
2                            0.000000                                44.44471
3                            0.000000                                44.44471
4                            8.145775                                24.43733

[5 rows x 41 columns]
```

[46]:
```
df['date_activ'] = pd.to_datetime(df['date_activ'])
df['date_end']   = pd.to_datetime(df['date_end'])

df['tenure'] = ((df['date_end'] - df['date_activ']).dt.days// 365)
```

[47]:
```
df.groupby(['tenure']).agg({'churn': 'mean'}).sort_values(by='tenure',␣
↪ascending=False)
```

[47]:
```
           churn
tenure
13      0.095238
12      0.083333
11      0.059783
10      0.045455
9       0.012500
8       0.047244
7       0.075472
```

```
6       0.075407
5       0.091999
4       0.127473
3       0.143874
2       0.176471
```

1.4 Modeling

1.5 Encoding

[55]:
```
drop_cols = ["id", "date_activ", "date_end", "date_modif_prod", "date_renewal"]

X = df.drop(columns=drop_cols + ["churn"])
y = df["churn"]
print(X.shape)
print(y.shape)
```

```
(14606, 36)
(14606,)
```

[56]:
```
numeric_features = X.select_dtypes(include=["int64", "float64"]).columns
categorical_features = X.select_dtypes(include=["object"]).columns
print("Numeric:", list(numeric_features))
print("Categorical:", list(categorical_features))
```

```
Numeric: ['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_off_peak',
'forecast_price_energy_peak', 'forecast_price_pow_off_peak', 'has_gas',
'imp_cons', 'margin_gross_pow_ele', 'margin_net_pow_ele', 'nb_prod_act',
'net_margin', 'num_years_antig', 'origin_up', 'pow_max', 'tenure_years',
'offpeak_diff_dec_january_energy', 'offpeak_diff_dec_january_power',
'off_peak_peak_var_mean_diff', 'peak_mid_peak_var_mean_diff',
'off_peak_mid_peak_var_mean_diff', 'off_peak_peak_fix_mean_diff',
'peak_mid_peak_fix_mean_diff', 'off_peak_mid_peak_fix_mean_diff',
'off_peak_peak_var_max_monthly_diff', 'peak_mid_peak_var_max_monthly_diff',
'off_peak_mid_peak_var_max_monthly_diff', 'off_peak_peak_fix_max_monthly_diff',
'peak_mid_peak_fix_max_monthly_diff', 'off_peak_mid_peak_fix_max_monthly_diff',
'tenure']
Categorical: []
```

[61]:
```
preprocess = ColumnTransformer(
    transformers=[
        ("cat", OneHotEncoder(handle_unknown="ignore"), categorical_features),
        ("num", "passthrough", numeric_features),
    ]
)
```

1.5.1 Split

[62]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,␣
↪random_state=42)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(10954, 36)
(10954,)
(3652, 36)
(3652,)
```

[63]:
```
model = RandomForestClassifier(
    n_estimators=1000
)
model.fit(X_train, y_train)
```

[63]:
```
RandomForestClassifier(n_estimators=1000)
```

[64]:
```
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
```

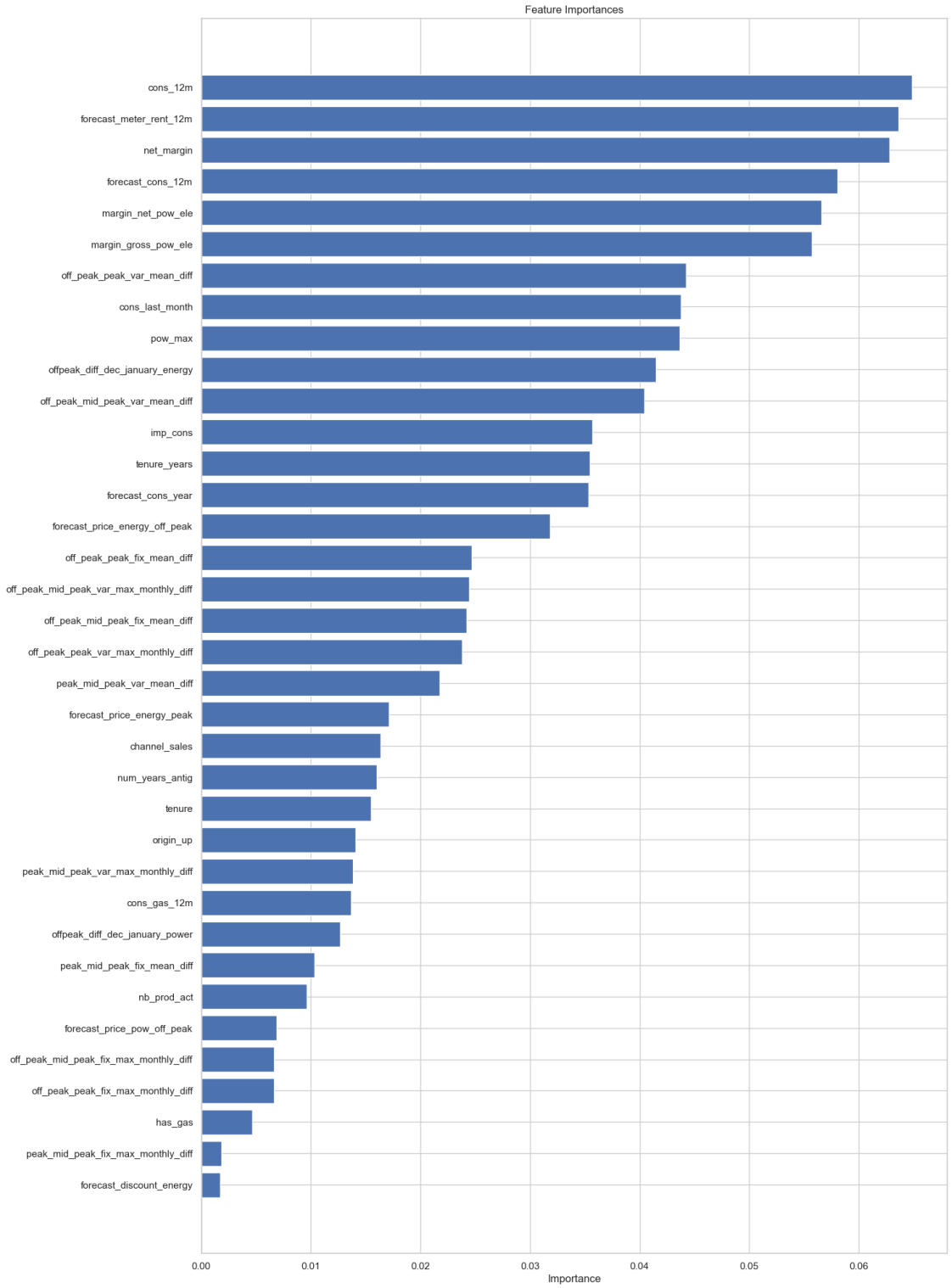
```
              precision    recall  f1-score   support

           0       0.90      1.00      0.95      3286
           1       0.79      0.04      0.08       366

    accuracy                           0.90      3652
   macro avg       0.85      0.52      0.51      3652
weighted avg       0.89      0.90      0.86      3652

[[3282    4]
 [ 351   15]]
```

1.5.2 Feature Importance

[65]:
```
feature_importances = pd.DataFrame({
    'features': X_train.columns,
    'importance': model.feature_importances_
}).sort_values(by='importance', ascending=True).reset_index()
```

[66]:
```
plt.figure(figsize=(15, 25))
plt.title('Feature Importances')
plt.barh(range(len(feature_importances)), feature_importances['importance'],␣
↪color='b', align='center')
```

```
plt.yticks(range(len(feature_importances)), feature_importances['features'])
plt.xlabel('Importance')
plt.show()
```



Dari chart ini, kita dapat mengamati beberapa poin berikut: - Net margin dan konsumsi selama 12 bulan merupakan faktor utama yang mendorong churn dalam model ini. - Margin pada langganan listrik (power subscription) juga menjadi faktor yang berpengaruh. - Faktor terkait waktu tampak cukup signifikan, terutama jumlah bulan pelanggan telah aktif, tenure, dan jumlah bulan sejak mereka terakhir memperbarui kontrak. - Fitur yang direkomendasikan oleh rekan kita berada di setengah bagian atas dalam hal pengaruh, dan beberapa fitur yang dibangun dari fitur tersebut bahkan memiliki kinerja lebih baik. - Fitur-fitur terkait sensitivitas harga tersebar dan bukan merupakan pendorong utama churn.