Feature Engineering

---

1. Import packages
2. Load data
3. Feature engineering

---

1. Import packages

In [12]:
```
import pandas as pd
```

---

2. Load data

In [13]:
```
df = pd.read_csv('clean_eda_data.csv')
df["date_activ"] = pd.to_datetime(df["date_activ"], format='%Y-%m-%d')
df["date_end"] = pd.to_datetime(df["date_end"], format='%Y-%m-%d')
df["date_modif_prod"] = pd.to_datetime(df["date_modif_prod"], format='%Y-%m-%d')
df["date_renewal"] = pd.to_datetime(df["date_renewal"], format='%Y-%m-%d')
```

In [14]:
```
df.head(3)
```

Out[14]:

	Unnamed: 0	id	channel_sales	cons_12m	cons_gas_12m	cons_last_month	date_activ	dat
0	0	24011ae4ebbe3035111d65fa7c15bc57	foosdfpfkusacimwkcsosbicdxkicaua	0	54946	0	2013-06-15	20
1	1	d29c2c54acc38ff3c0614d0a653813dd	MISSING	4660	0	0	2009-08-21	20
2	2	764c75f661154dac3a6c254cd082ea7d	foosdfpfkusacimwkcsosbicdxkicaua	544	0	0	2010-04-16	20

3 rows × 54 columns

---

3. Feature engineering

Difference between off-peak prices in December and preceding January

Below is the code created by your colleague to calculate the feature described above. Use this code to re-create this feature and then think about ways to build on this feature to create features with a higher predictive power.

In [15]:
```
price_df = pd.read_csv('price_data.csv')
price_df["price_date"] = pd.to_datetime(price_df["price_date"], format='%Y-%m-%d')
price_df.head()
```

Out[15]:

	id	price_date	price_off_peak_var	price_peak_var	price_mid_peak_var	price_off_peak_fix	price_peak_fix	pr
0	038af19179925da21a25619c5a24b745	2015-01-01	0.151367	0.0	0.0	44.266931	0.0	
1	038af19179925da21a25619c5a24b745	2015-02-01	0.151367	0.0	0.0	44.266931	0.0	
2	038af19179925da21a25619c5a24b745	2015-03-01	0.151367	0.0	0.0	44.266931	0.0	
3	038af19179925da21a25619c5a24b745	2015-04-01	0.149626	0.0	0.0	44.266931	0.0	
4	038af19179925da21a25619c5a24b745	2015-05-01	0.149626	0.0	0.0	44.266931	0.0	

In [26]:

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

# Get january and december prices

jan_prices = monthly_price_by_id.groupby('id').first()

jan_prices = monthly_price_by_id.groupby('id').first().reset_index()
print(jan_prices)
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.d
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.head()
```

```
                                     id price_date  price_off_peak_var  \
0      0002203ffbb812588b632b9e628cc38d 2015-01-01            0.126098
1      0004351ebdd665e6ee664792efc4fd13 2015-01-01            0.148047
2      0010bcc39e42b3c2131ed2ce55246e3c 2015-01-01            0.150837
3      0010ee3855fdea87602a5b7aba8e42de 2015-01-01            0.123086
4      00114d74e963e47177db89bc70108537 2015-01-01            0.149434
...                                 ...        ...                 ...
16091  ffef185810e44254c3a4c6395e6b4d8a 2015-01-01            0.162720
16092  fffac626da707b1b5ab11e8431a4d0a2 2015-01-01            0.148825
16093  fffc0cacd305dd51f316424bbb08d1bd 2015-01-01            0.153159
16094  fffe4f5646aa39c7f97f95ae2679ce64 2015-01-01            0.127566
16095  ffff7fa066f1fb305ae285bb03bf325a 2015-01-01            0.129444

       price_off_peak_fix
0               40.565969
1               44.266931
2               44.444710
3               40.565969
4               44.266931
...                   ...
16091           41.063970
16092           44.266931
16093           41.063970
16094           40.565969
16095           40.565969

[16096 rows x 4 columns]
```

Out[26]:

	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

In [24]:

Out[24]:

	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

In [73]:

```
client_df = pd.read_csv('client_data.csv')

final=client_df.merge(diff, on='id', how='left')


fin=final.drop(['channel_sales','id','date_activ','date_end','date_modif_prod','date_renewal','origin_up'], axis=1)


X=fin.drop(['churn'], axis=1)
Y=fin['churn']

X['has_gas']=X['has_gas'].replace(['t','f'],[0, 1])
```

	cons_12m	cons_gas_12m	cons_last_month	forecast_cons_12m	\
0	0	54946	0	0.00	
1	4660	0	0	189.95	
2	544	0	0	47.96	
3	1584	0	0	240.04	
4	4425	0	526	445.75	
...	...	...	...	...	
14601	32270	47940	0	4648.01	
14602	7223	0	181	631.69	
14603	1844	0	179	190.39	
14604	131	0	0	19.34	
14605	8730	0	0	762.41	

	forecast_cons_year	forecast_discount_energy	forecast_meter_rent_12m	\
0	0	0.0	1.78	
1	0	0.0	16.27	
2	0	0.0	38.72	
3	0	0.0	19.83	
4	526	0.0	131.73	
...	...	...	...	
14601	0	0.0	18.57	
14602	181	0.0	144.03	
14603	179	0.0	129.60	
14604	0	0.0	7.18	
14605	0	0.0	1.07	

	forecast_price_energy_off_peak	forecast_price_energy_peak	\
0	0.114481	0.098142	
1	0.145711	0.000000	
2	0.165794	0.087899	
3	0.146694	0.000000	
4	0.116900	0.100015	
...	...	...	
14601	0.138305	0.000000	
14602	0.100167	0.091892	
14603	0.116900	0.100015	
14604	0.145711	0.000000	
14605	0.167086	0.088454	

	forecast_price_pow_off_peak	has_gas	imp_cons	margin_gross_pow_ele	\
0	40.606701	0	0.00	25.44	
1	44.311378	1	0.00	16.38	
2	44.311378	1	0.00	28.60	
3	44.311378	1	0.00	30.22	
4	40.606701	1	52.32	44.91	

```
...                         ...      ...       ...                  ...
14601                  44.311378        0      0.00                27.88
14602                  58.995952        1     15.94                 0.00
14603                  40.606701        1     18.05                39.84
14604                  44.311378        1      0.00                13.08
14605                  45.311378        1      0.00                11.84

       margin_net_pow_ele  nb_prod_act  net_margin  num_years_antig  pow_max  \
0                   25.44            2      678.99                3   43.648
1                   16.38            1       18.89                6   13.800
2                   28.60            1        6.60                6   13.856
3                   30.22            1       25.46                6   13.200
4                   44.91            1       47.98                6   19.800
...                   ...          ...         ...              ...      ...
14601               27.88            2      381.77                4   15.000
14602                0.00            1       90.34                3    6.000
14603               39.84            1       20.38                4   15.935
14604               13.08            1        0.96                3   11.000
14605               11.84            1       96.34                6   10.392

       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 20 columns]
```

In [74]:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, ConfusionMatrixDisplay

x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=4, stratify=Y)


rf = RandomForestClassifier()
rf.fit(x_train, y_train)
```

Out[74]:
```
RandomForestClassifier()
```

In [78]:
```
y_pred = rf.predict(x_test)

accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9055441478439425
```