BCG GAMMA VIRTUAL EXPERIENCE

POWER CO.

PROBLEM ANALYSIS : CLIENT PROBLEM OVERVIEW

- PowerCo is a major gas and electricity utility that supplies to corporate, SME (Small & Medium Enterprise), and residential customers. The power liberalisation of the energy market in Europe has led to significant customer churn, especially in the SME segment. They have partnered with BCG to help diagnose the source of churning SME customers.
- Some core values for the company are :
  - Low level of differentiation between the product available
  - Customer service
  - Keeping Customer for long term
  - Building Brand Loyalty

PROBLEM ANALYSIS : CLIENT HYPOTHESIS

- According to our client price changes affect the customer churn. Therefore, it will be helpful to know which customers are bound to churn so that a discount can be offered to them. This will ensure customers sticking to Power Co.
- Client is basically interested in knowing through our model which customers should be provided with a discount of 20%. In other words which customers are likely to churn.

EXPLORATORY DATA ANALYTICS

POWER CO.

DATA OVERVIEW

- WE HAVE BEEN PROVIDED WITH A DATA WHICH INCLUDES THREE TYPES OF DATA :
  - **HISTORICAL CUSTOMER DATA** : CUSTOMER DATA SUCH AS SIGNAGE, USAGE SIGN UP DATE, FORECASTED USAGE.
  - **HISTORICAL PRICING DATA** : VARIABLE AND FIXED PRICING DATA
  - **CHURN INDICATOR** : WHETHER EACH CUSTOMER HAS CHURNED OR NOT

HISTORICAL CUSTOMER DATA WITH CHURN INDICATOR

```
<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
```

client_data.csv

- id = client company identifier
- activity_new = category of the company's activity
- channel_sales = code of the sales channel
- cons_12m = electricity consumption of the past 12 months
- cons_gas_12m = gas consumption of the past 12 months
- cons_last_month = electricity consumption of the last month
- date_activ = date of activation of the contract
- date_end = registered date of the end of the contract
- date_modif_prod = date of the last modification of the product
- date_renewal = date of the next contract renewal
- forecast_cons_12m = forecasted electricity consumption for next 12 months
- forecast_cons_year = forecasted electricity consumption for the next calendar year
- forecast_discount_energy = forecasted value of current discount
- forecast_meter_rent_12m = forecasted bill of meter rental for the next 2 months
- forecast_price_energy_off_peak = forecasted energy price for 1st period (off peak)
- forecast_price_energy_peak = forecasted energy price for 2nd period (peak)
- forecast_price_pow_off_peak = forecasted power price for 1st period (off peak)
- has_gas = indicated if client is also a gas client
- imp_cons = current paid consumption
- margin_gross_pow_ele = gross margin on power subscription
- margin_net_pow_ele = net margin on power subscription
- nb_prod_act = number of active products and services
- net_margin = total net margin
- num_years_antig = antiquity of the client (in number of years)
- origin_up = code of the electricity campaign the customer first subscribed to
- pow_max = subscribed power
- churn = has the client churned over the next 3 months

PRICE DATA

```
<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
```

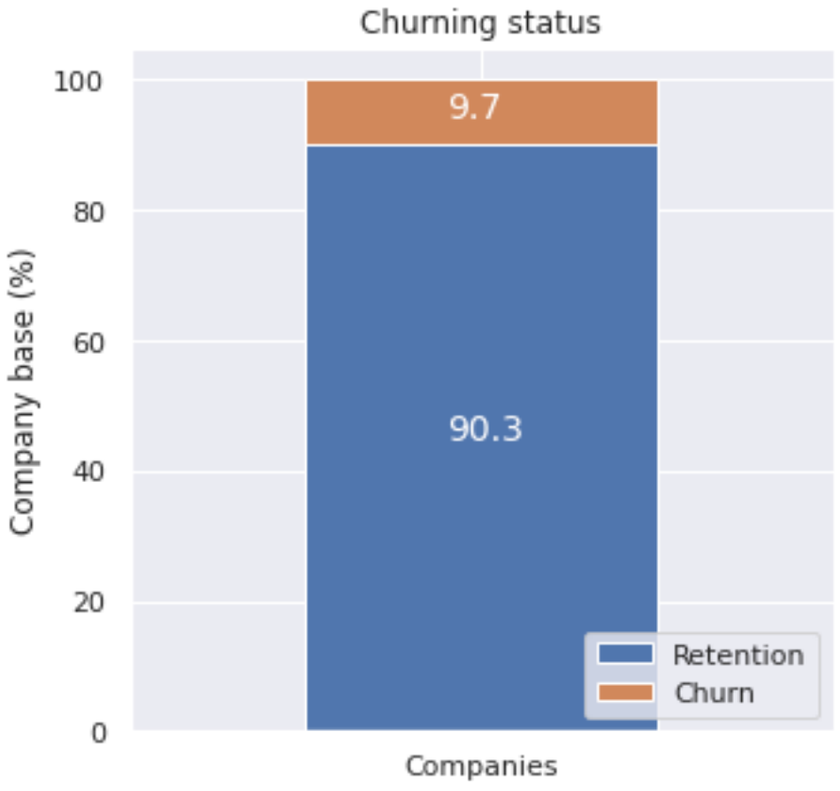
price_data.csv

- id = client company identifier
- price_date = reference date
- price_off_peak_var = price of energy for the 1st period (off peak)
- price_peak_var = price of energy for the 2nd period (peak)
- price_mid_peak_var = price of energy for the 3rd period (mid peak)
- price_off_peak_fix = price of power for the 1st period (off peak)
- price_peak_fix = price of power for the 2nd period (peak)
- price_mid_peak_fix = price of power for the 3rd period (mid peak)

ANALYSE : CHURN

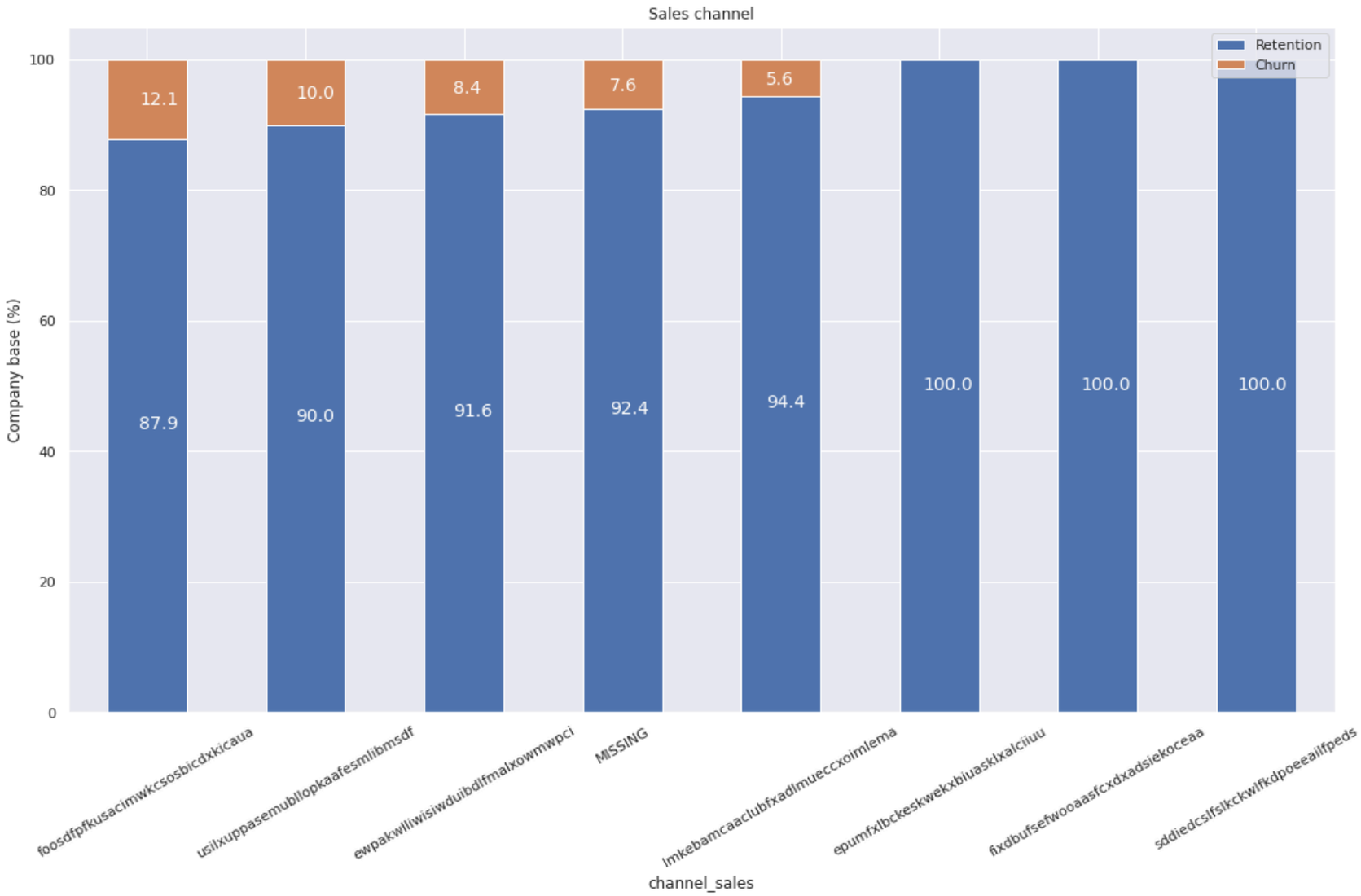
WE WILL FIRST ANALYSE OUR CHURN VARIABLE AND SEE HOW THAT VARIABLE IS BEHAVING. AS WE KNOW THE CHURN VARIABLE IS 0 OR 1, HENCE IN ORDER TO VISUALISE THIS VARIABLE WE MAY NEED TO KEEP THE FOLLOWING THINGS IN MIND.

AS WE KNOW THAT CHURN RATE FOR ANY COMPANY CANNOT BE MORE THAN 20% IN ORDER FOR THAT COMPANY TO BE SUCCESSFUL. HENCE WE NEED TO BE MINDFUL OF THE CHURN RATE FOR OUR USE CASE AS WELL.
IN ORDER TO VISUALISE THIS WE MAY HAVE A STACKED BAR PLOT WHICH WILL GIVE US A VERY CLEAR PICTURE.



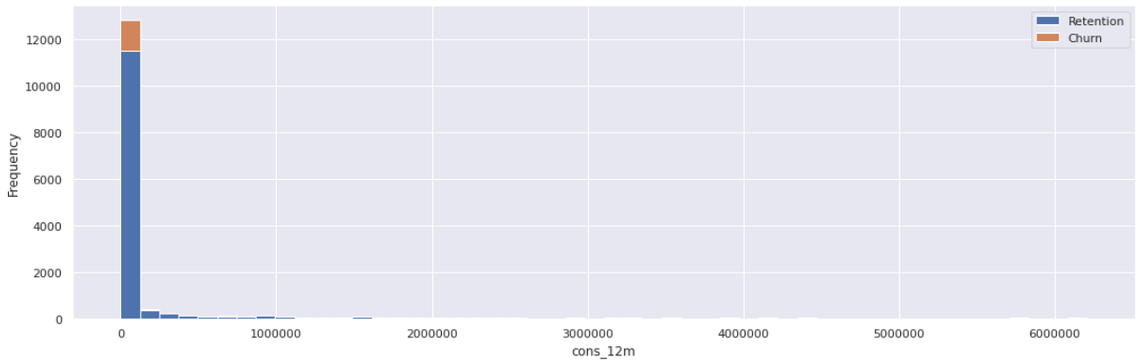
NOW AROUND 10% OF THE TOTAL CUSTOMER HAS CHURNED WHICH SEEMS ABOUT RIGHT. NOW LETS SEE HOW THE CHURN RATES VARY ACROSS VARIOUS SALES CHANNELS. THIS HELP US TO ANALYSE WHICH SALES CHANNEL HAS THE HIGHEST CHURN RATE AND WHICH HAS THE LOWEST CHURN RATE. OUR CLIENT MIGHT USE THIS INFORMATION TO PREDICT WHICH SALES CHANNEL IS PERFORMING WELL AND WHICH IS DOING NOT SO GOOD.

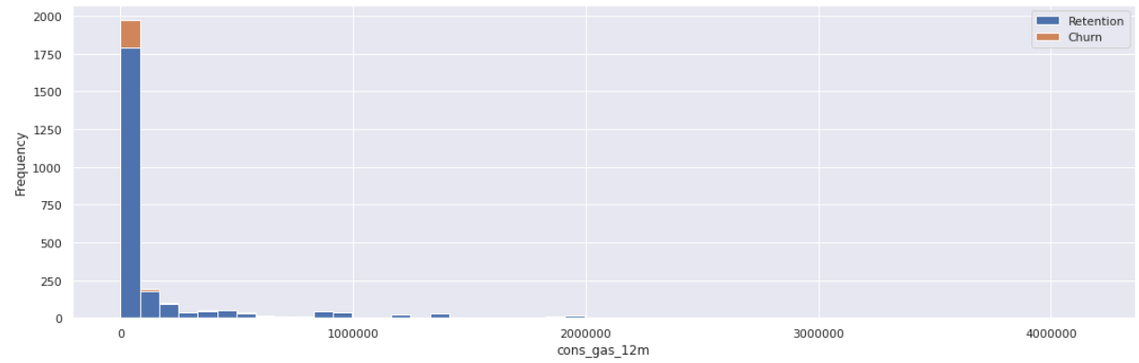
ANALYSE : CHURN WITH CHANNELS

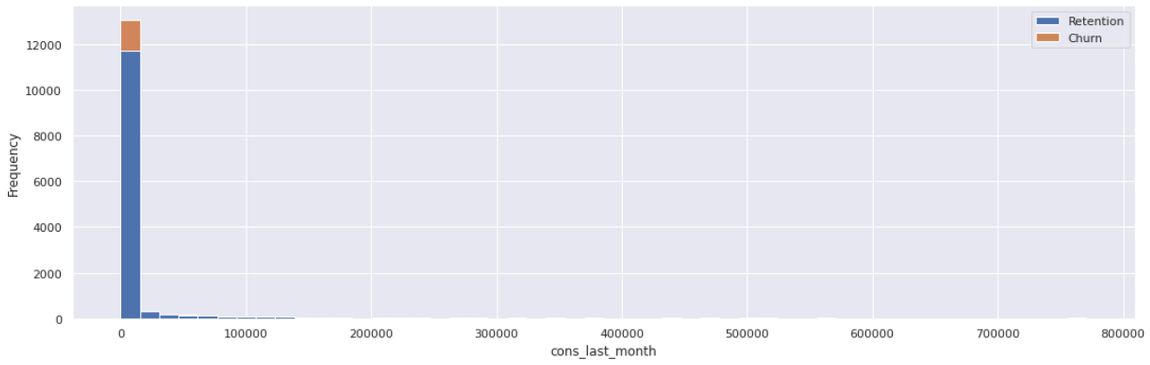


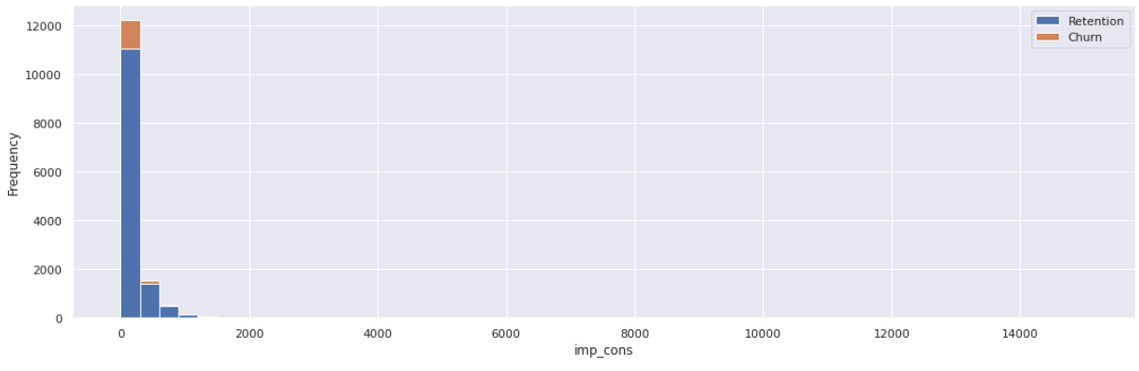
IN THE ABOVE FIGURE WE HAVE CREATED ONE CATEGORY OF MISSING SALES CHANNEL. THIS DATA WAS NOT PROVIDED BY THE CLIENT. SO WE LABELLED THIS DATA AS A MISSING DATA. WE CHECKED THE CHURN AND NON CHURN FOR THIS TO DETERMINE WHETHER THESE VALUES ARE IMPORTANT. AS WE SEE THAT FOR THESE MISSING CHANNEL VALUES ARE IMPORTANT AS THEY HAVE SIGNIFICANT CHRUN RATE OF 7.6 %. HENCE WE NEED TO ASK OUR CLIENT FOR THESE VALUES AS THEY ARE QUITE IMPORTANT FOR OUR ANALYSIS AND QUITE PRODUCTIVE.

ANALYSE : CONSUMPTION DATA



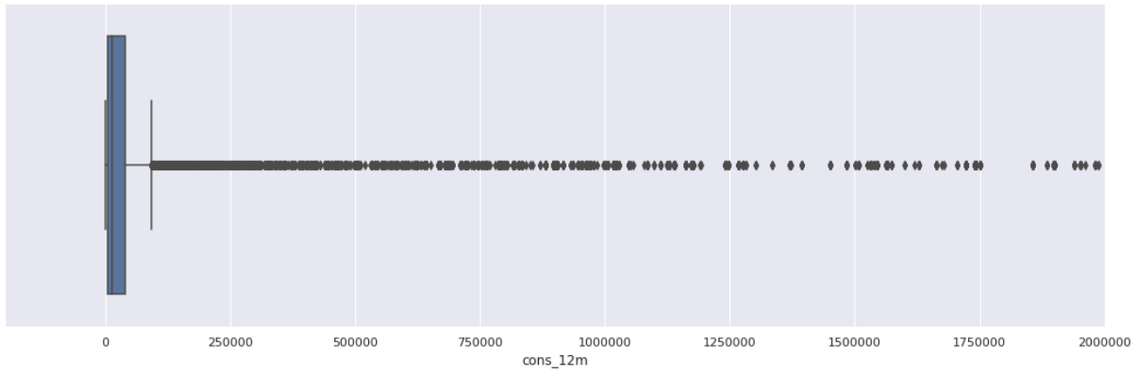


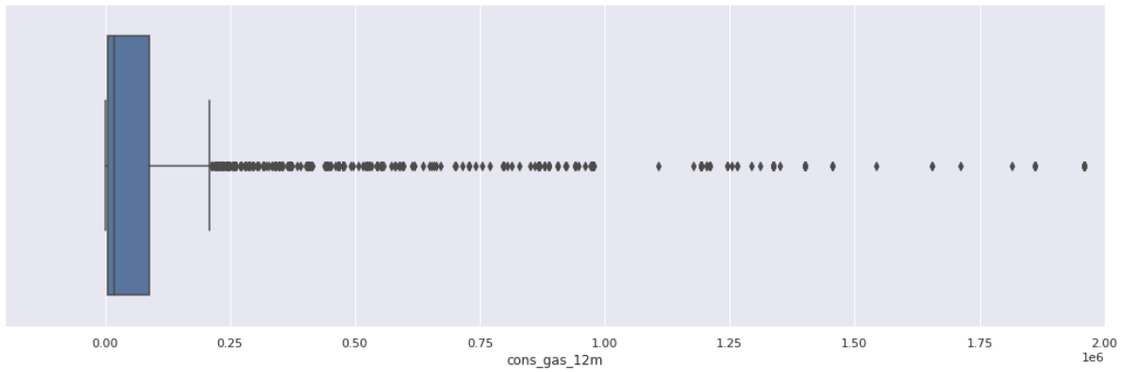


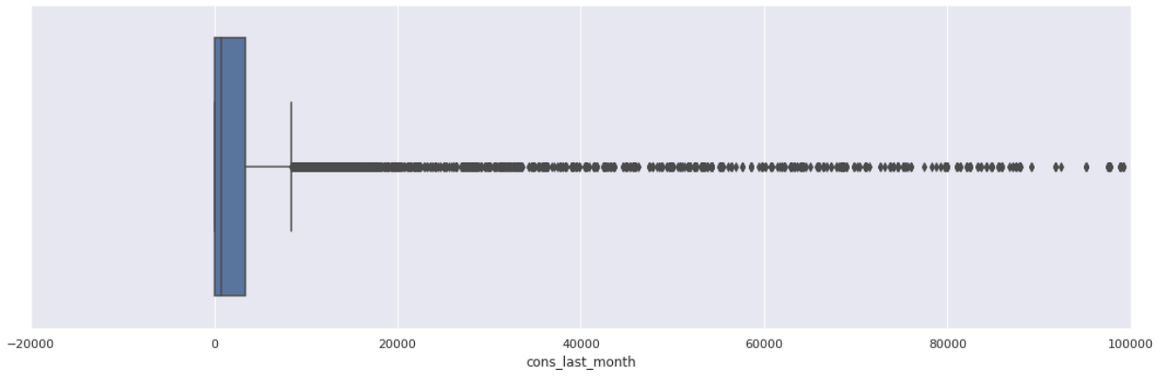


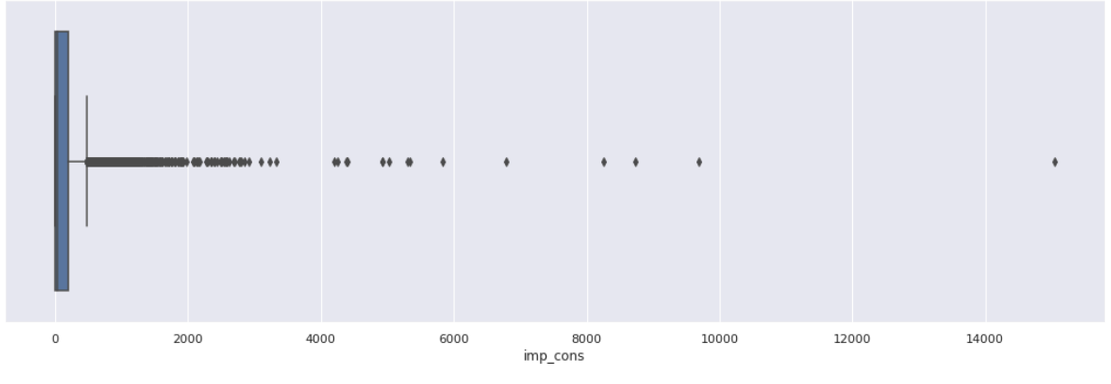
AS WE CAN OBSERVE FROM THE ABOVE FIGURES THAT DATA FOR CONSUMPTION IS RIGHTLY SKEWED OR POSITIVELY SKEWED. IN THIS DATA USUALLY MODE IS LESS THAN MEAN. IN SKEWED DATA ITS ALWAYS PREFERABLE TO GO FOR MEDIUM AS A REPLACEMENT OF OUTLIER VALUES AS THIS WILL REDUCE THE IMPACT OF OUTLIERS IN THE DATA. NOW LETS ANALYSE THE OUTLIERS BY TRADITIONAL BOX PLOT. MINIMUM–FIRST QUARTILE (Q1) – MEDIAN – THIRD QUARTILE (Q3)–MAXIMUM

ANALYSE : CONSUMPTION DATA







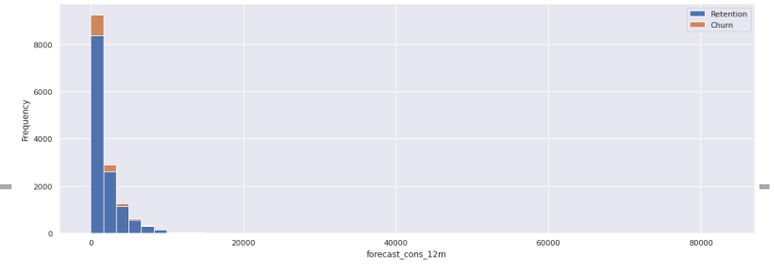


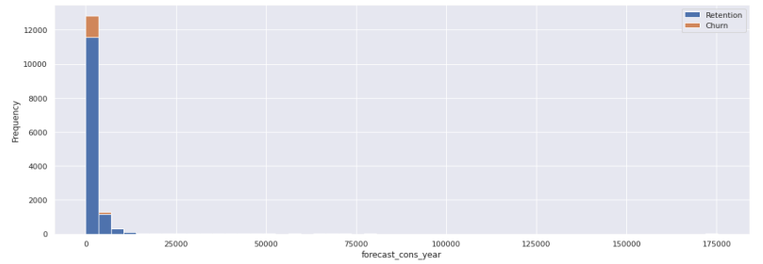
WE HAVE TO DEAL WITH OUTLIERS AND SKEWNESS OF THIS DATA SO THAT WE DO NOT BIAS OUR MACHINE LEARNING MODELS.

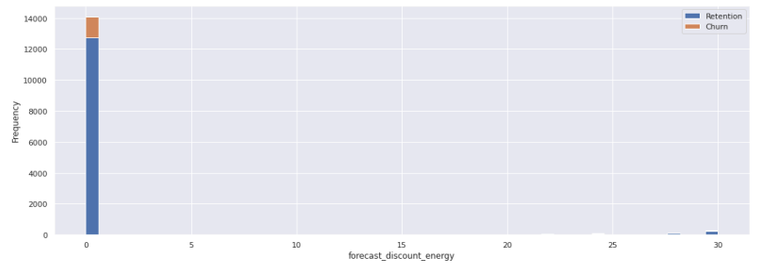
POWER CO.

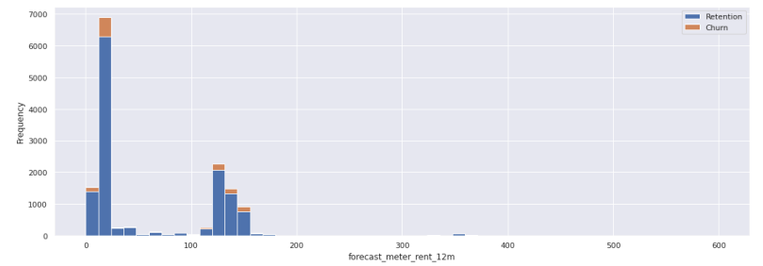
ANALYSE : FORECAST DATA

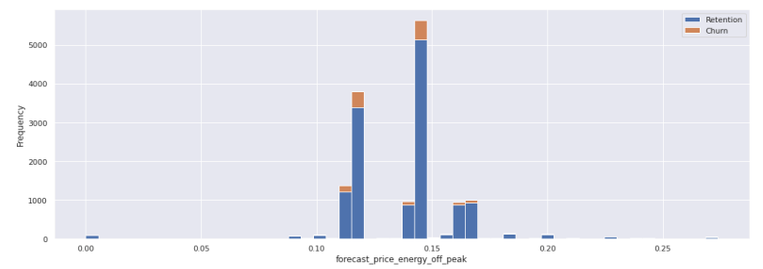
WE HAVE TO DEAL WITH OUTLIERS AND SKEWNESS OF THIS DATA SO THAT WE DO NOT BIAS OUR MACHINE LEARNING MODELS.

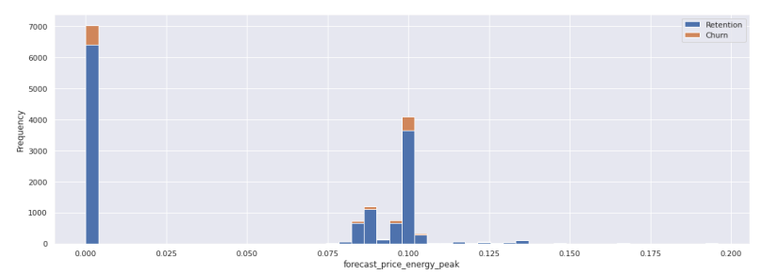


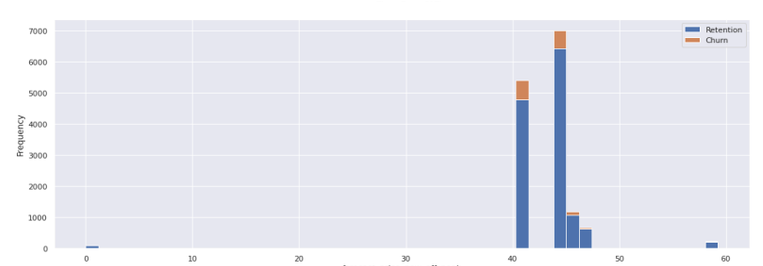




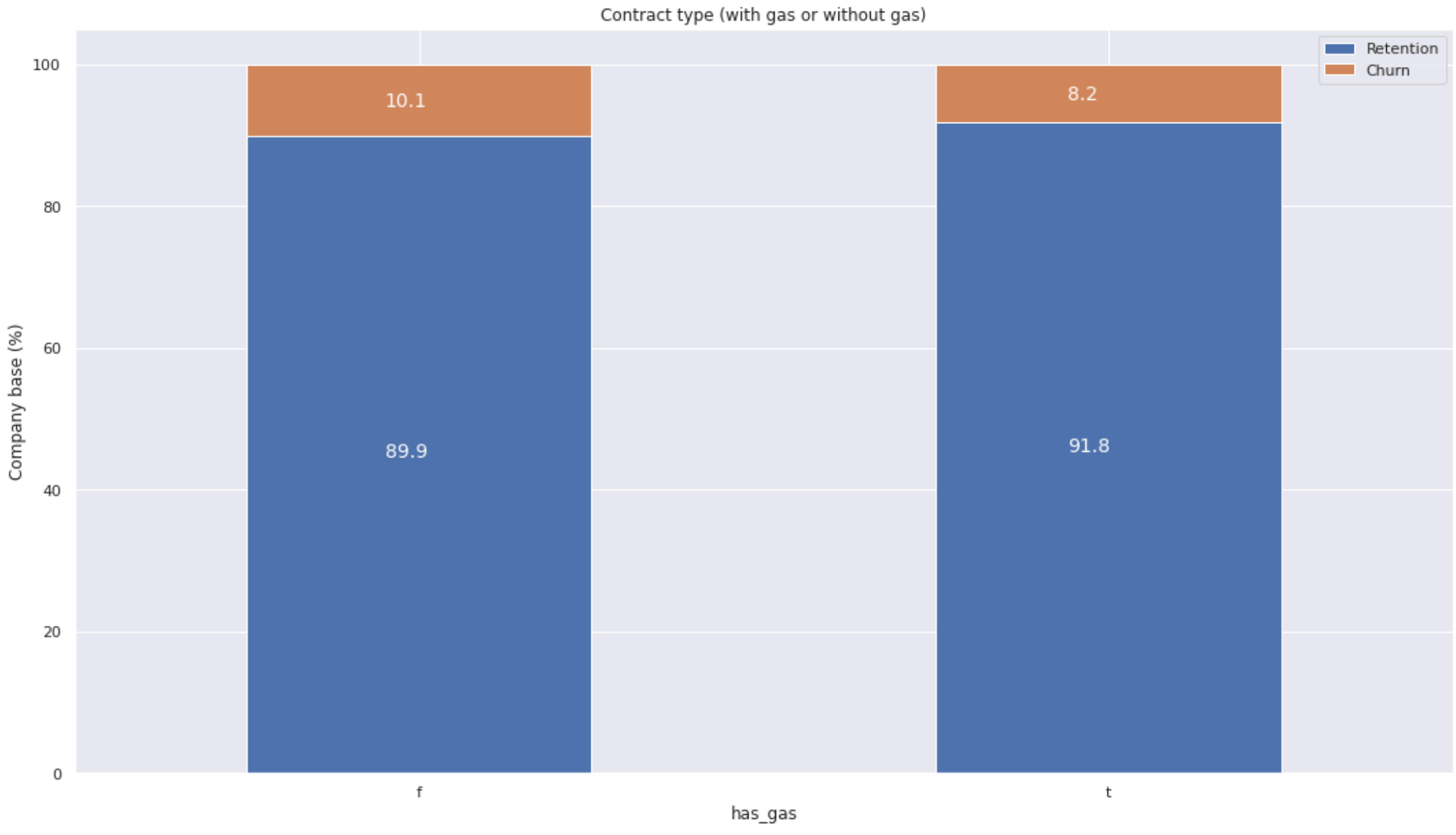






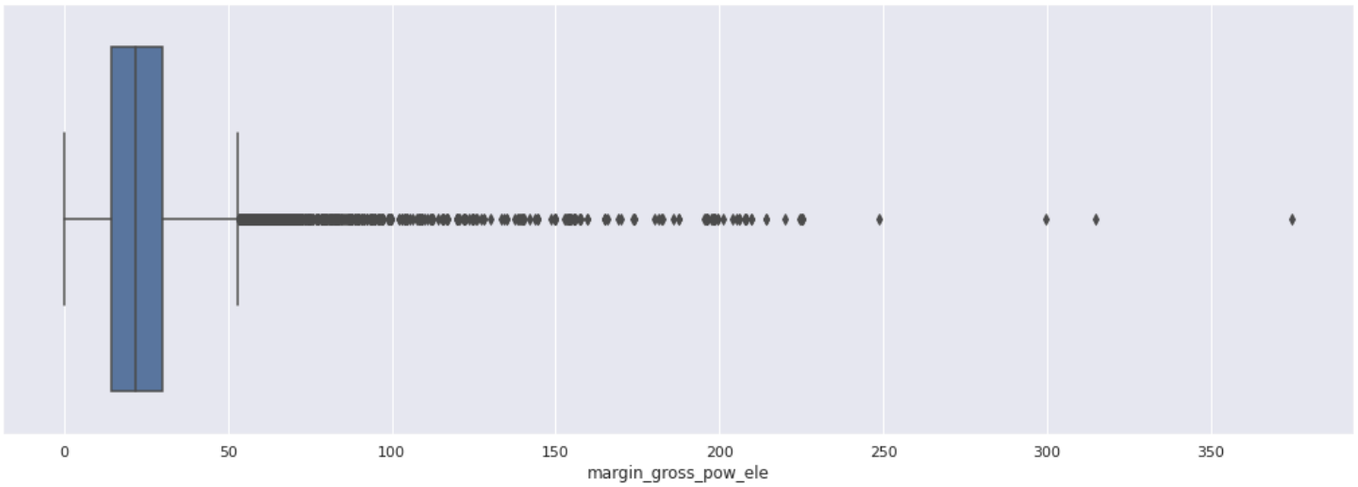


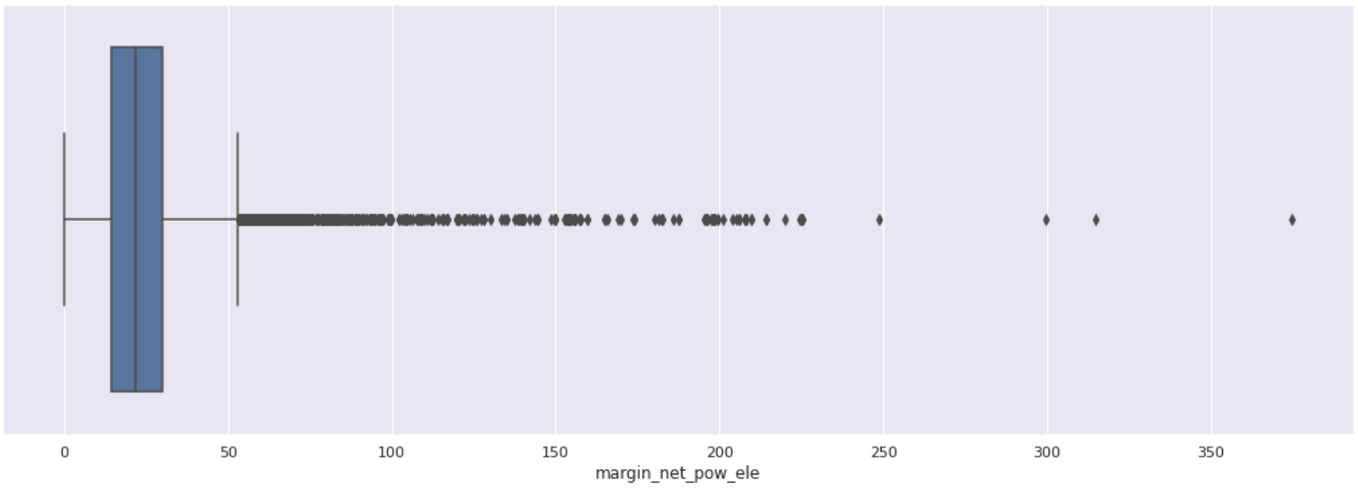
ANALYSE : HAS GAS

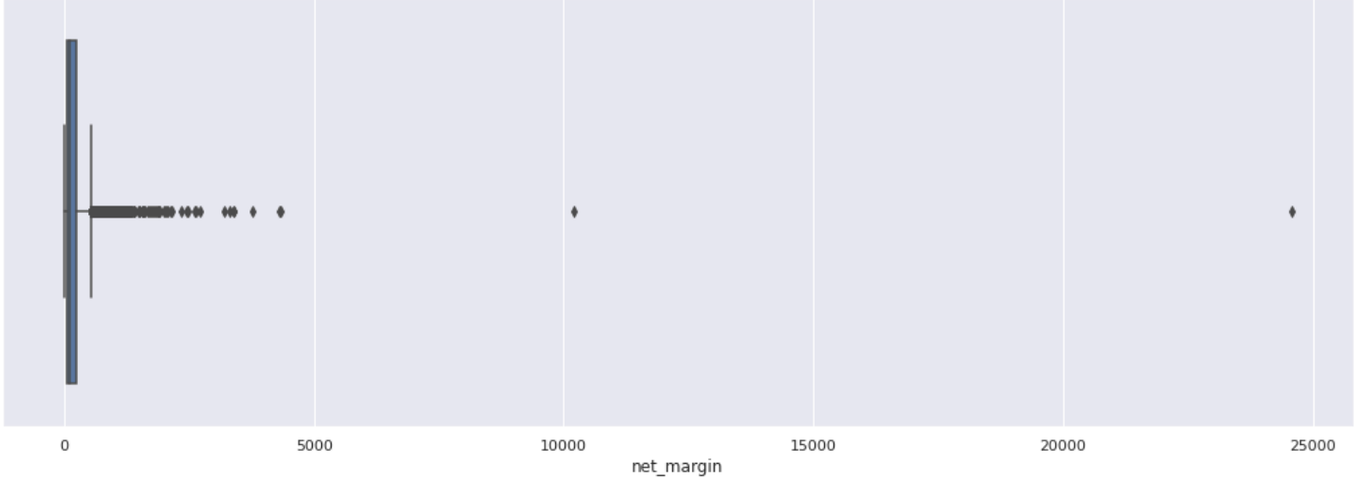


THERE IS NO SIGNIFICANT DIFFERENCE BETWEEN THE CHURN RATES OF PEOPLE WITH GAS SUBSCRIPTION AND NO GAS SUBSCRIPTION. FOR PEOPLE HAVING NO GAS SUBSCRIPTION THE CHURN RATE IS 10.1 % AND FOR PEOPLE WITH GAS THE CHURN RATE IS 8.2%.

ANALYSE : MARGINS ON POWER



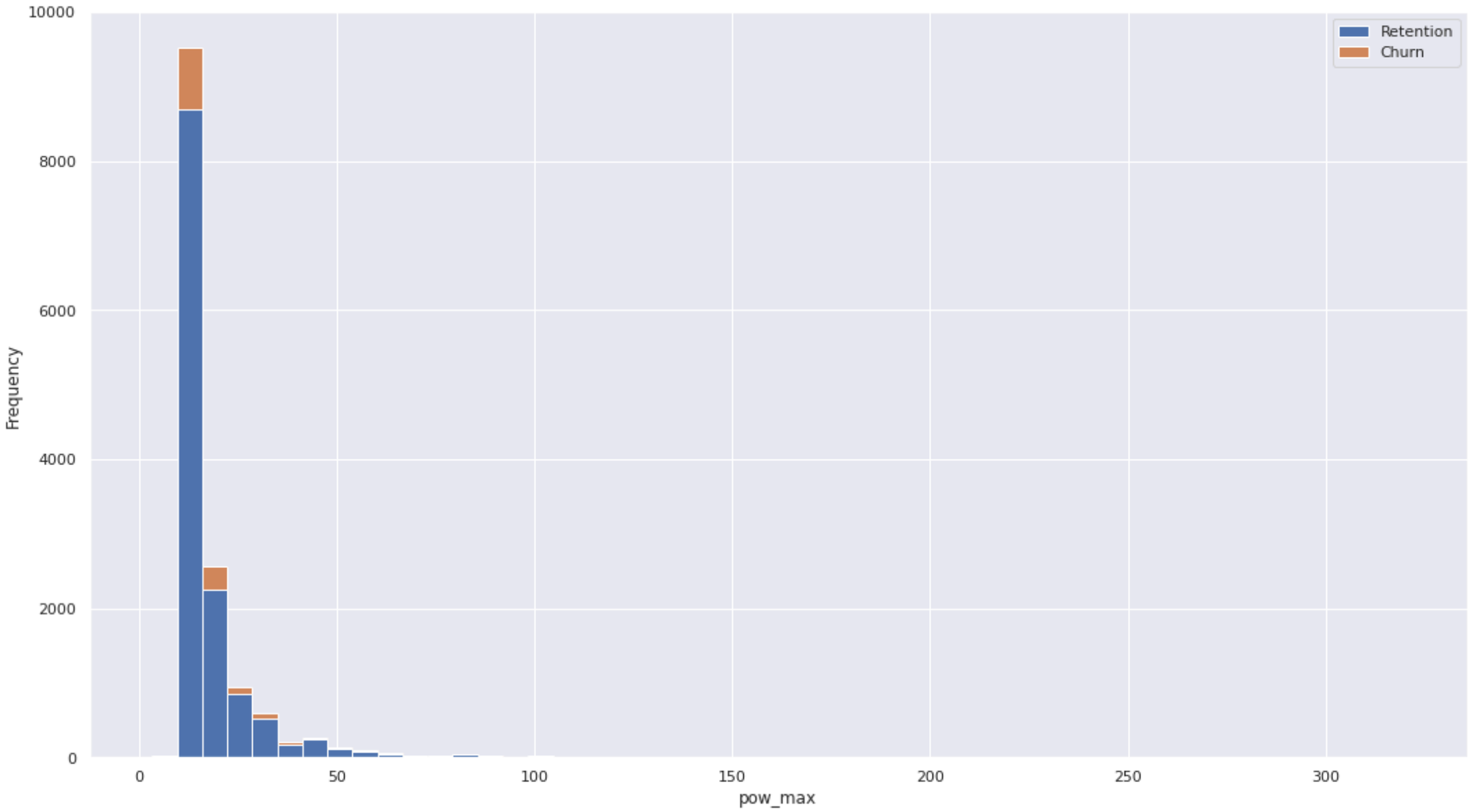




WE NEED TO DEAL WITH OUTLIERS IN THIS CASE. THERE ARE TWO WAYS TO UNDERSTAND THE OUTLIERS:

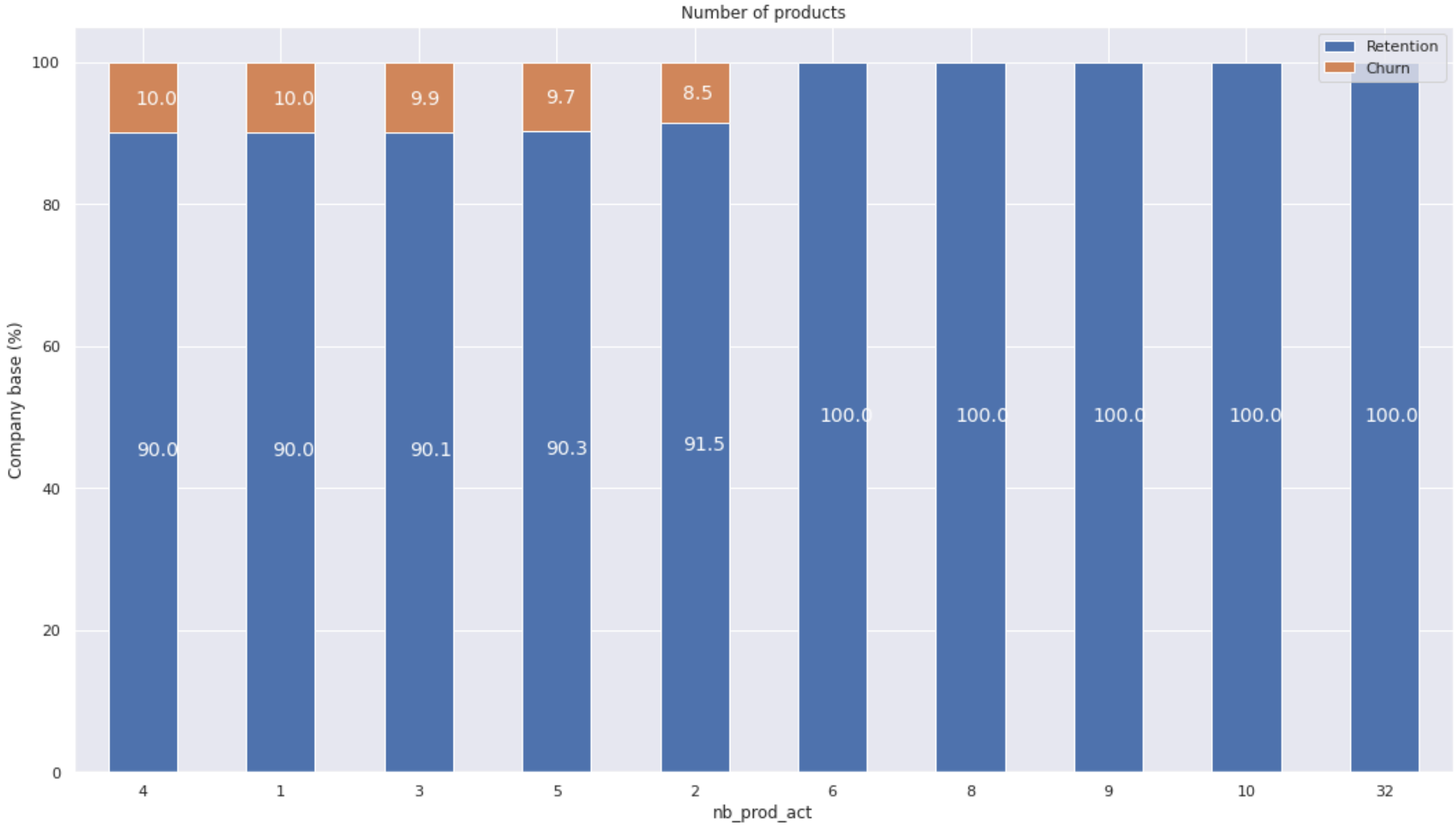
1. Z SCORE
2. BOX PLOTS (Q3 +1.5 IQR/ Q1–1.5*IQR)

ANALYSE : MARGINS ON POWER

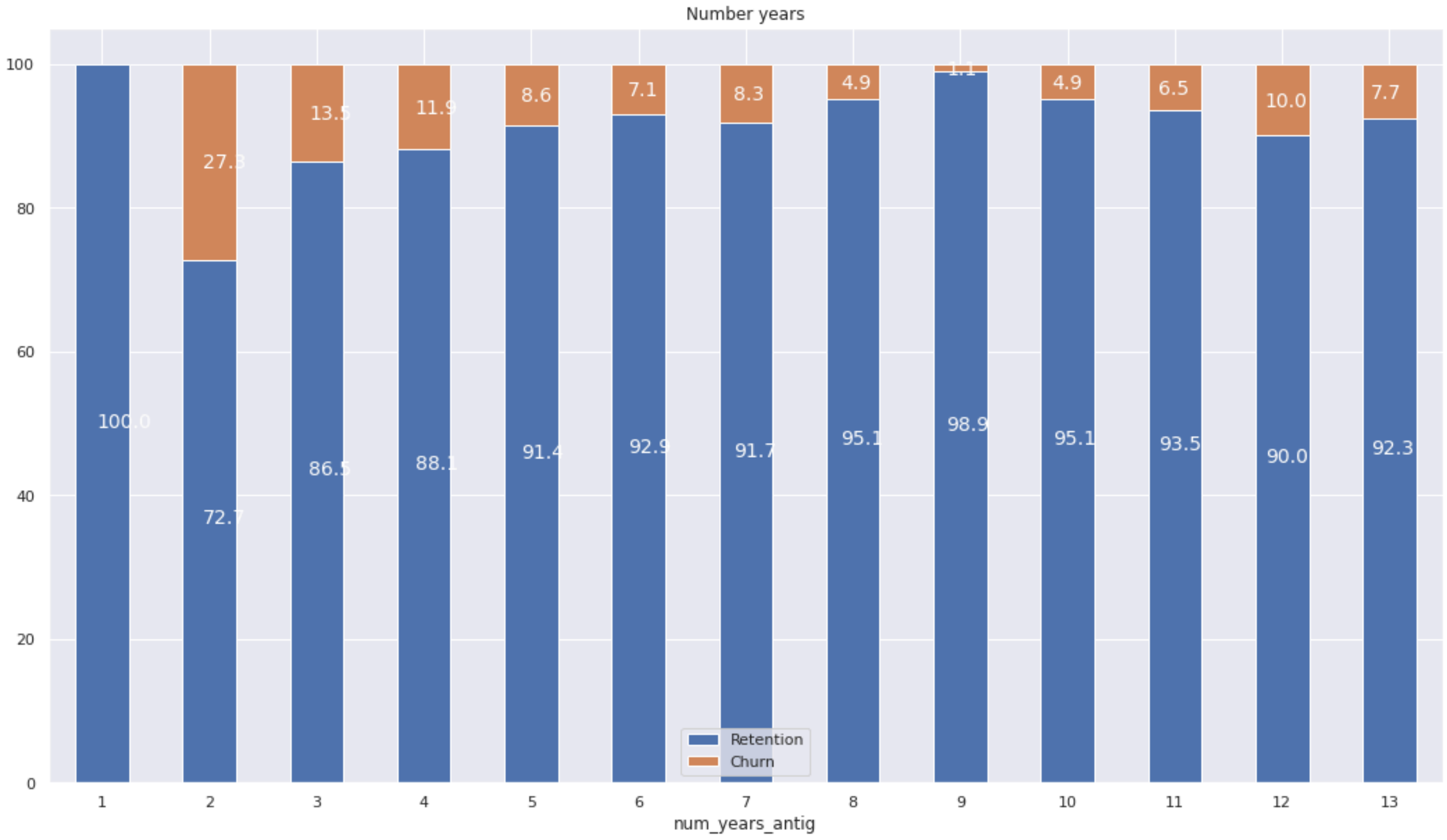


RIGHT SKEWED OR POSITIVELY SKEWED DATA. THIS DATA IS ALSO PRONE TO SOME OUTLIERS AND POSSESS A BIAS.

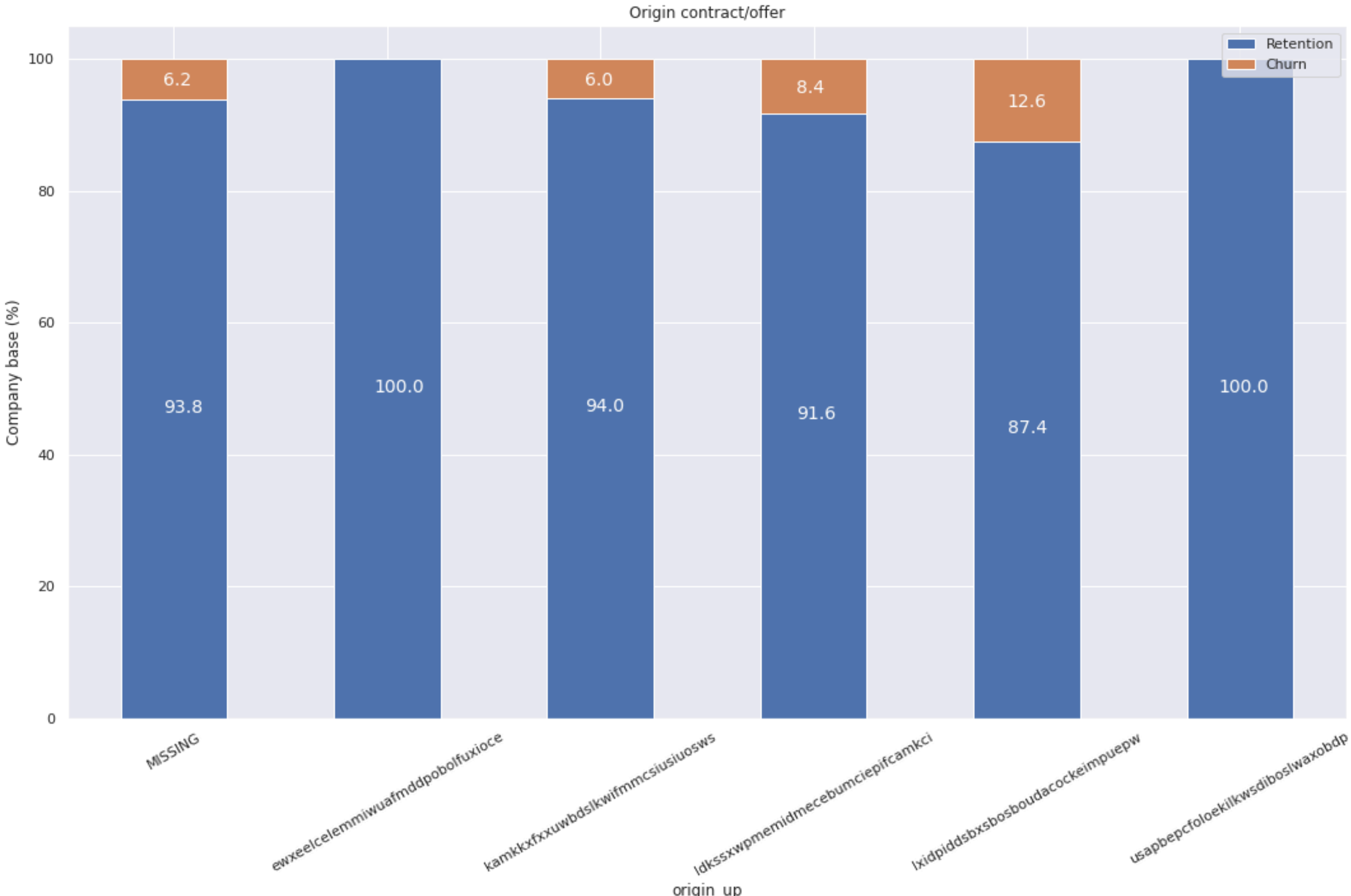
ANALYSE : OTHER FEATURES



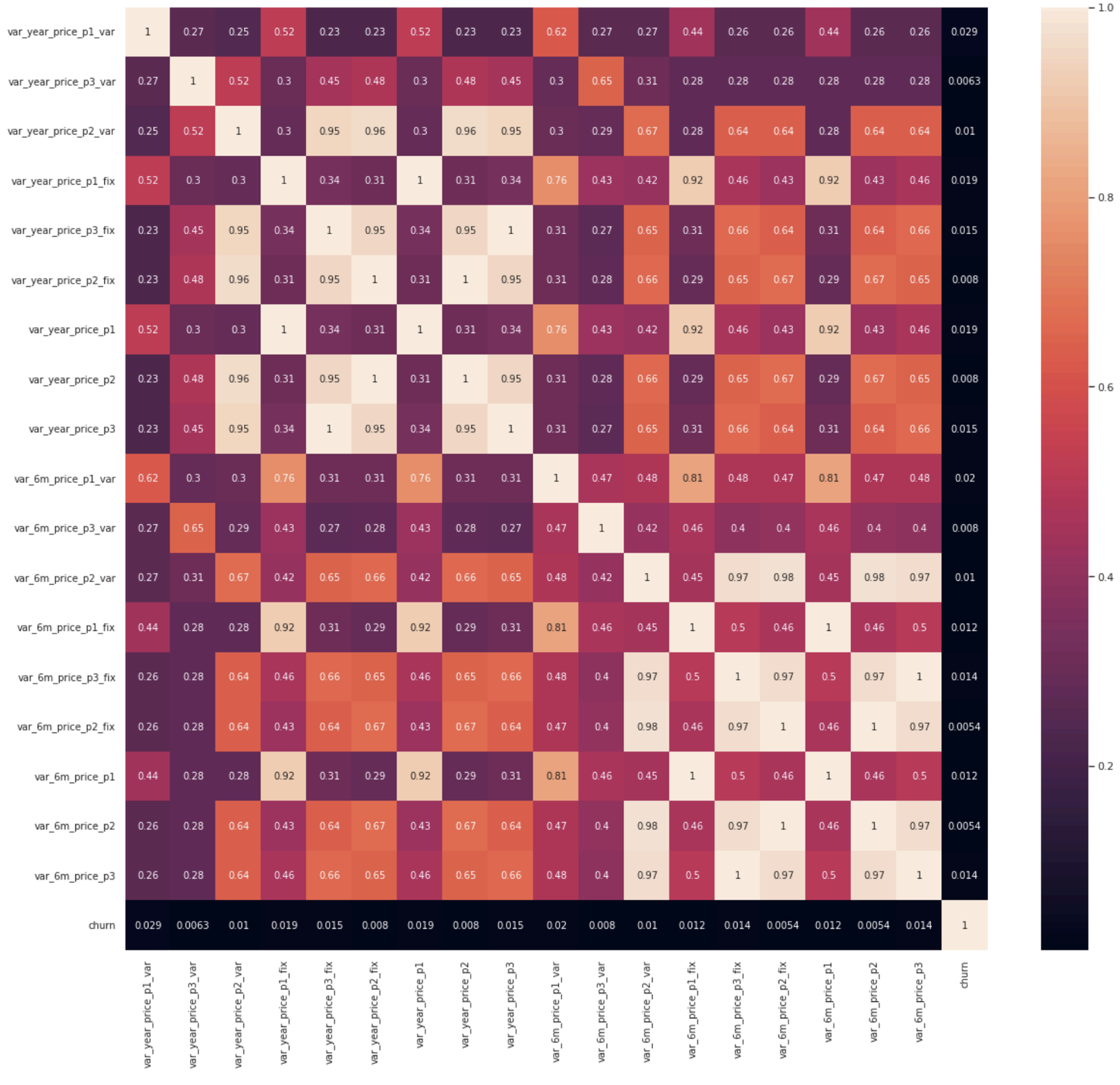
ANALYSE : OTHER FEATURES



ANALYSE : OTHER FEATURES



HYPOTHESIS INVESTIGATION



FROM THE CORRELATION PLOT, IT SHOWS THAT THE PRICE SENSITIVITY FEATURES A HIGH INTER-CORRELATION WITH EACH OTHER, BUT OVERALL THE CORRELATION WITH CHURN IS VERY LOW. THIS INDICATES THAT THERE IS A WEAK LINEAR RELATIONSHIP BETWEEN PRICE SENSITIVITY AND CHURN. THIS SUGGESTS THAT FOR PRICE SENSITIVE TO BE A MAJOR DRIVER FOR PREDICTING CHURN, WE MAY NEED TO ENGINEER THE FEATURES DIFFERENTLY.

FEATURE ENGINEERING

POWER CO.

BASIC OPERATIONS

- Changing the date variables from string/object to date time object in order to perform arithmetic operations on them.

```
[ ] 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')
```

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

PRICE DATA

	price_date	price_off_peak	price_peak	price_mid_peak
0	2015-01-01	43.369211	10.747871	6.486193
1	2015-02-01	43.380929	10.728772	6.479383
2	2015-03-01	43.397137	10.699607	6.460518
3	2015-04-01	43.431698	10.703311	6.453648
4	2015-05-01	43.458922	10.657812	6.421198
5	2015-06-01	43.491051	10.471024	6.311206
6	2015-07-01	43.488695	10.697605	6.452101
7	2015-08-01	43.503299	10.715283	6.462234
8	2015-09-01	43.489807	10.656512	6.422394
9	2015-10-01	43.492125	10.659144	6.426725
10	2015-11-01	43.560673	10.695108	6.452359
11	2015-12-01	43.642188	10.698066	6.457836

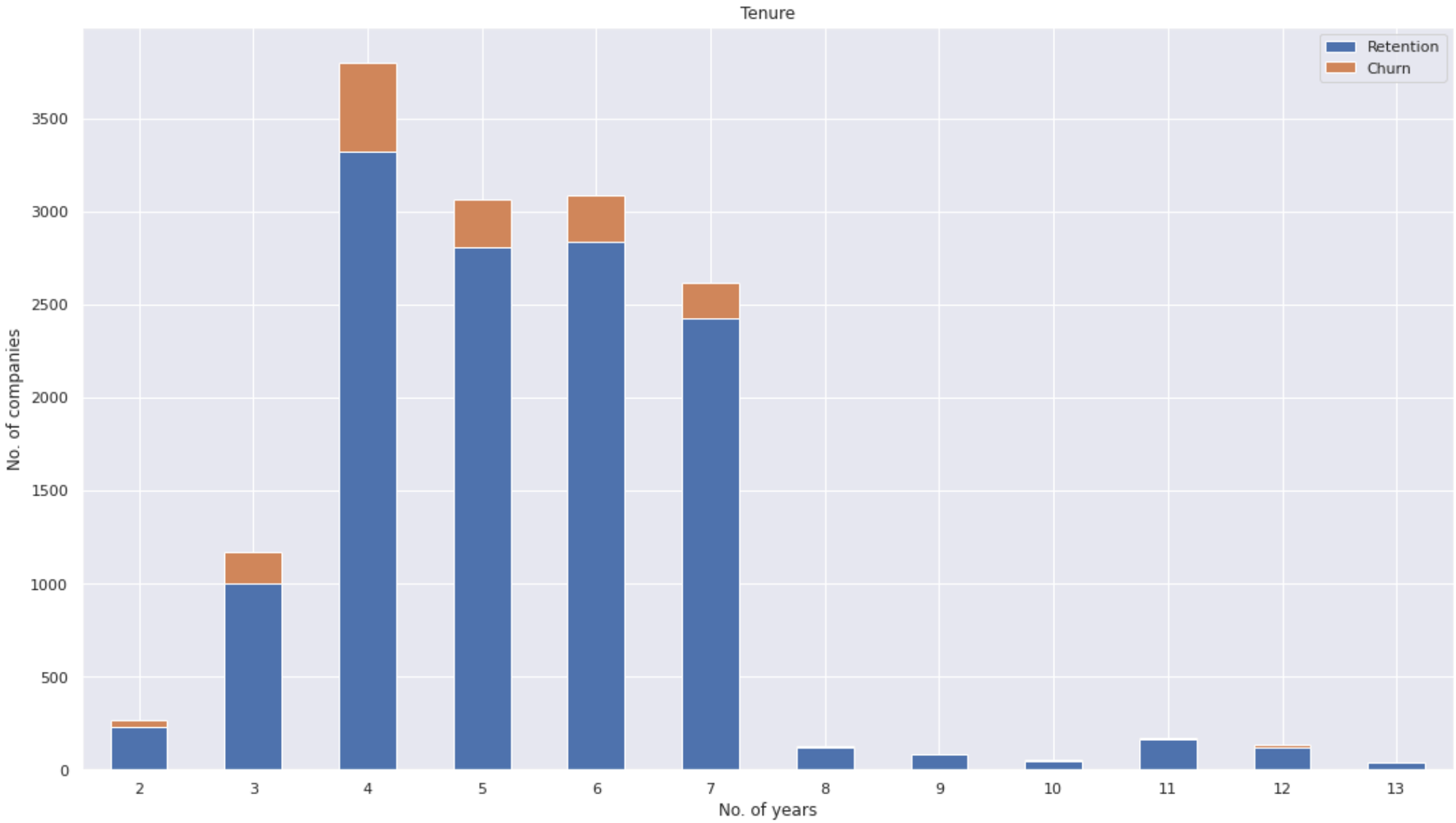
- If we go month wise ignoring the companies we note an interesting observation. The mean prices for all companies remains the same. Hence we need to aggregate the prices for the companies.
- Since we have the consumption data for each of the companies for the year 2015, we will create new features using the average of the year, the last six months, and the last three months to our model. This will provide us with more nuanced analysis into our price data and help our model predict better. Although we have the consumption data for each month of the companies, this is also a useful data but the data for six months, 3 months and a year will give in more concrete insights.
- Taking up the mean values for the prices of a particular company for the entire year in peak, mid peak and off peak season, from the monthly values

PRICE DATA

	id	price_off_peak_var	price_peak_var	price_mid_peak_var	price_off_peak_fix	price_peak_fix	price_mid_peak_fix
0	0002203ffbb812588b632b9e628cc38d	0.124338	0.103794	0.073160	40.701732	24.421038	16.280694
1	0004351ebdd665e6ee664792efc4fd13	0.146426	0.000000	0.000000	44.385450	0.000000	0.000000
2	0010bcc39e42b3c2131ed2ce55246e3c	0.181558	0.000000	0.000000	45.319710	0.000000	0.000000
3	0010ee3855fdea87602a5b7aba8e42de	0.118757	0.098292	0.069032	40.647427	24.388455	16.258971
4	00114d74e963e47177db89bc70108537	0.147926	0.000000	0.000000	44.266930	0.000000	0.000000

- mean values of price for the company for 6 months, 3 months and a year remains the same for peak, off peak and mid peak.
- Now as we have seen before that price variables are highly correlated so we may not need to include mean_6m and mean_3m into our target list. This will not yield us concrete results and hence we can have the mean prices for the year included in our feature maps.
- Lets take mean year price as our metrics for the companies.

TENURE OF THE COMPANIES : DATE ACTIVE – DATE END



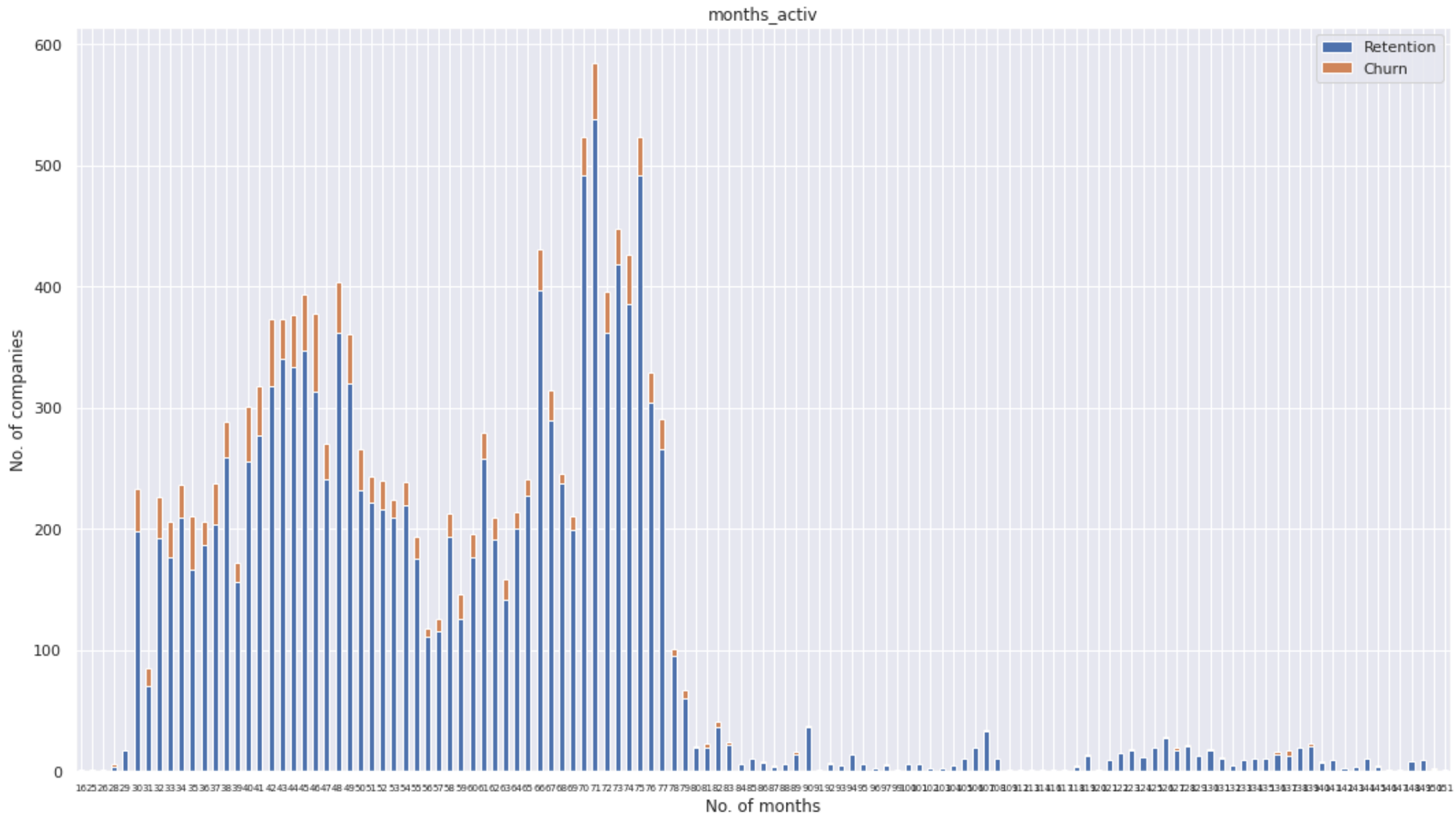
WE CAN CLEARLY THAT CHURN IS VERY LOW FOR COMPANIES WHICH JOINED RECENTLY OR THAT HAVE MADE THE CONTRACT A LONG TIME AGO. WITH THE HIGHER NUMBER OF CHURNERS WITHIN THE 3–7 YEARS OF TENURE.

TENURE OF THE COMPANIES : DATE ACTIVE – DATE END

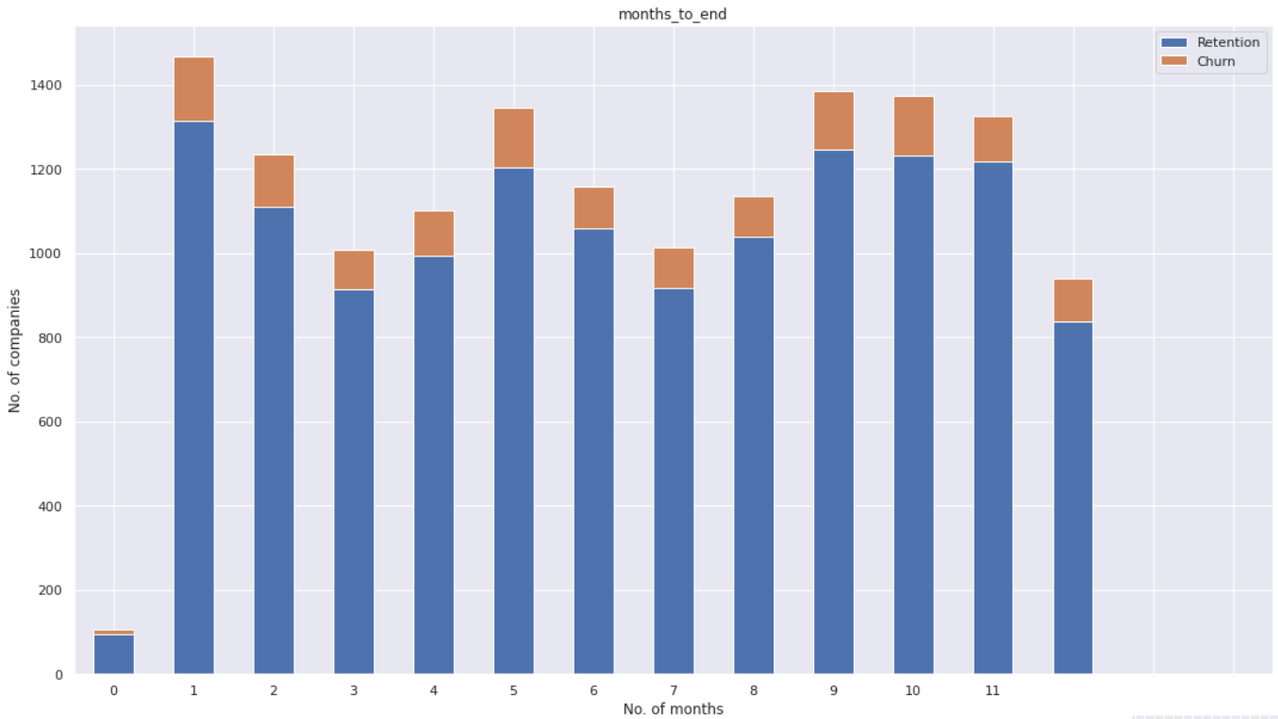
Most of the contracts are terminating for the companies between 2016 and 2017. Hence we will take a reference date of Jan 1st 2016 and will create the following features for more clarity.

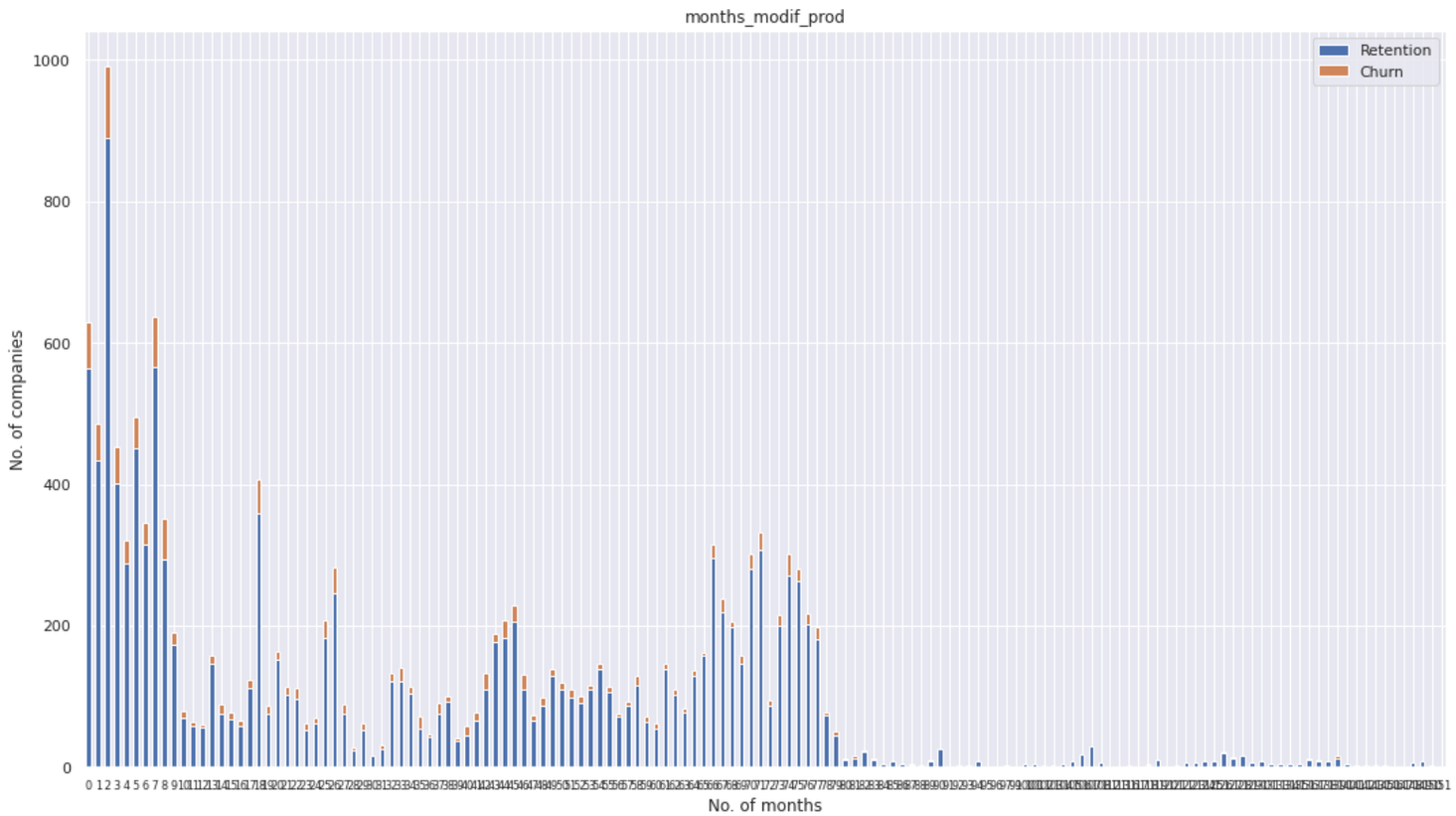
1. No. of months active according to the reference date
2. No. of months to the contract left at the refence date
3. No. of months since last modification
4. No. of months since last renewal

TENURE OF THE COMPANIES : DATE ACTIVE – DATE END

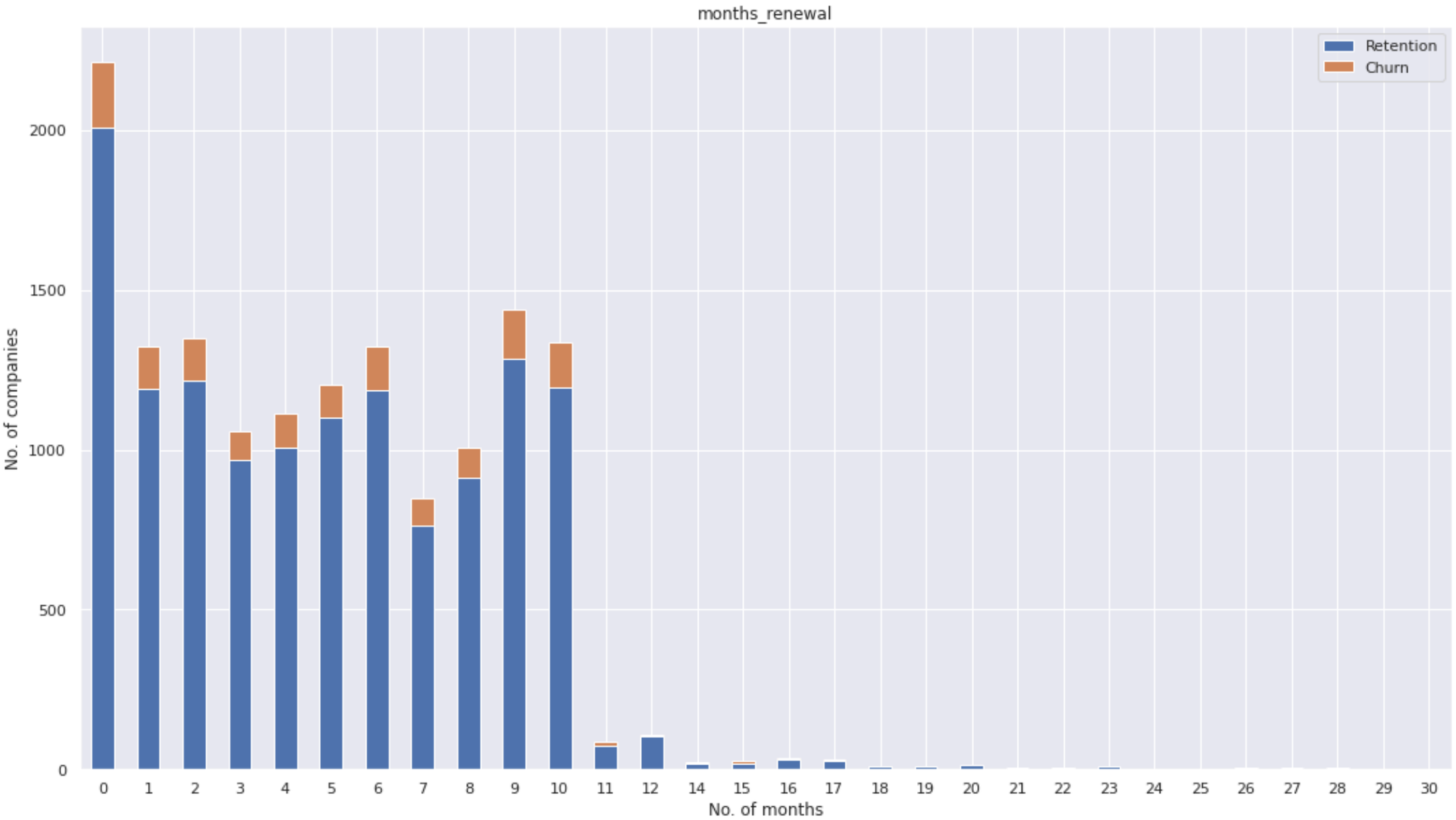


TENURE OF THE COMPANIES : DATE ACTIVE – DATE END





TENURE OF THE COMPANIES : DATE ACTIVE – DATE END



CATEGORICAL FEATURES

Lets handle the textual features as well. As we cannot pass text to our model we may need to label the textual features in our model. The problem with labelling multiple categorical features is that we may end up providing heirarchy to them which may not exist. For e.g we may label the features as 1,2,3,...

Hence we opt for one hot encoding making each category in the textual feature a new feature in itself. Also if we have 7 categories in textual features we may delete one category easily owing to its multicolinearity with other features. If I know the answer for 6 of them then its not very difficult to predict the 7th one.

	Samples in category
foosdfpfkusacimwkcsosbicdxkicaua	6754
MISSING	3725
lmkebamcaaclubfxadlmueccxoimlema	1843
usilxuppasemubllopkaafesmllbmsdf	1375
ewpakwlliwisiwduibdlfmalxowmwpci	893
sddiedcslfslkckwlfkdpoeeailfpeds	11
epumfxlbckeskwekxbiuaskIxalcliuu	3
fixdbufsefwooaasfcxdxadslekoceaa	2

	channel_MIS	channel_epu	channel_ewp	channel_fix	channel_foo	channel_lmk	channel_sdd	channel_usi
0	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0
3	0	0	0	0	0	1	0	0
4	1	0	0	0	0	0	0	0

SKEWED DATA

IN EDA we saw a lot of skewed data which was positively skewed. There is one way to normalise this data into a bell shaped gaussian curve. For this purpose we may use two types of transformations i.e log transform and exponential transform. It is very important to make this data normalised so as to prevent bias into the model.

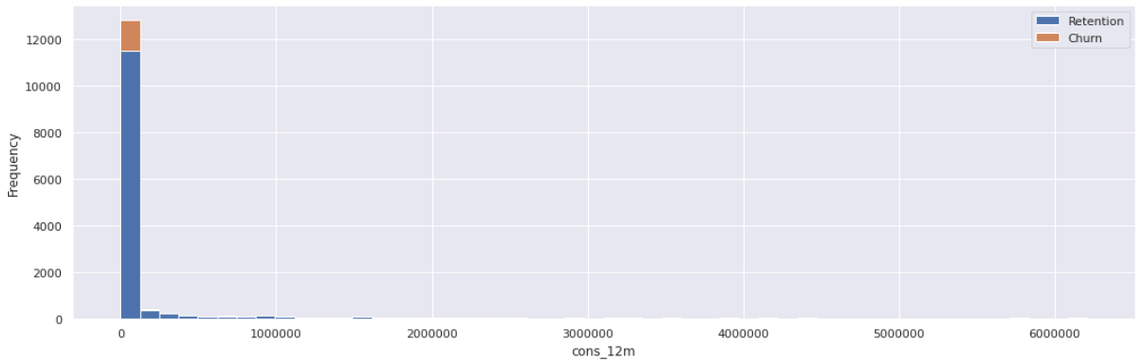
Skewness is not "bad" per se. Nonetheless, some predective models make fundamental assumptions related to variables being "normally distributed". Hence, the model will perform poorly if the data is highly skewed.

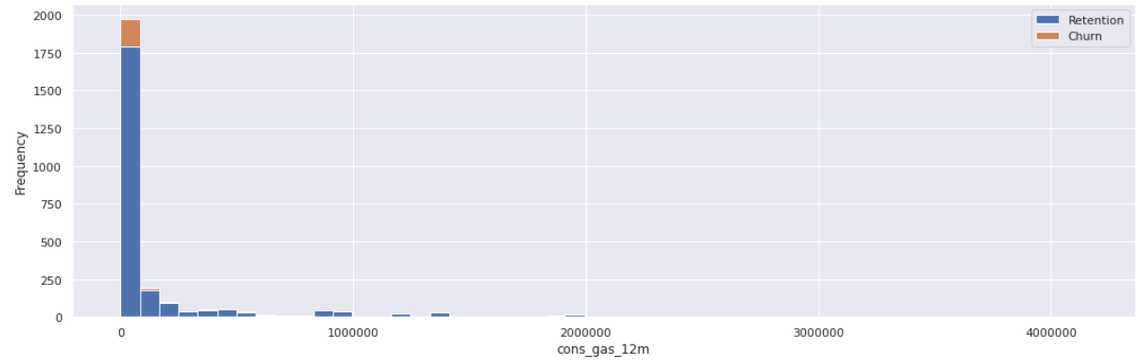
There are several methods in which we can reduce skewness such as square root , cube root , and log . In this case, we will use a log transformation which is usually recommended for right skewed data.

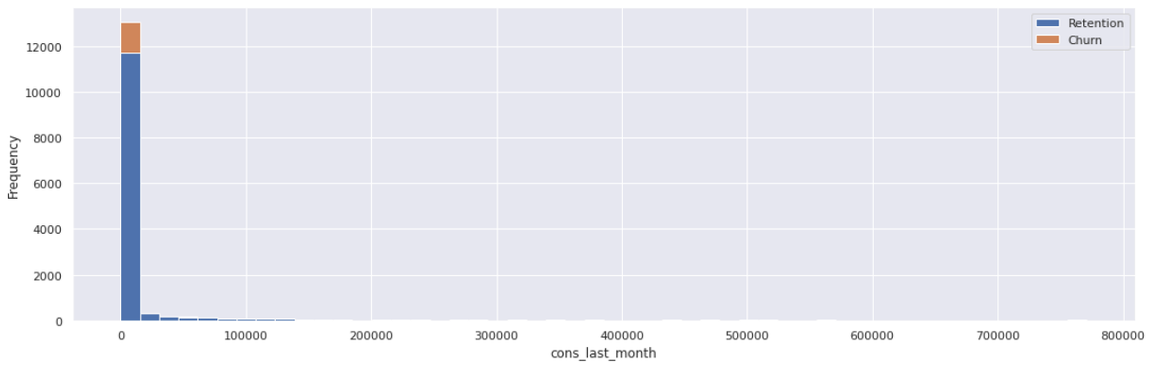
Particularly relevant to look at the standard deviation std which is very very high for some variables.Log transformation works differently for positive and negative data. Although we tend to remove the negative data or consider it as a NAN value in order to apply a uniform log transformation. In our dataset as described above we do not have any negative data. So we are good to go.

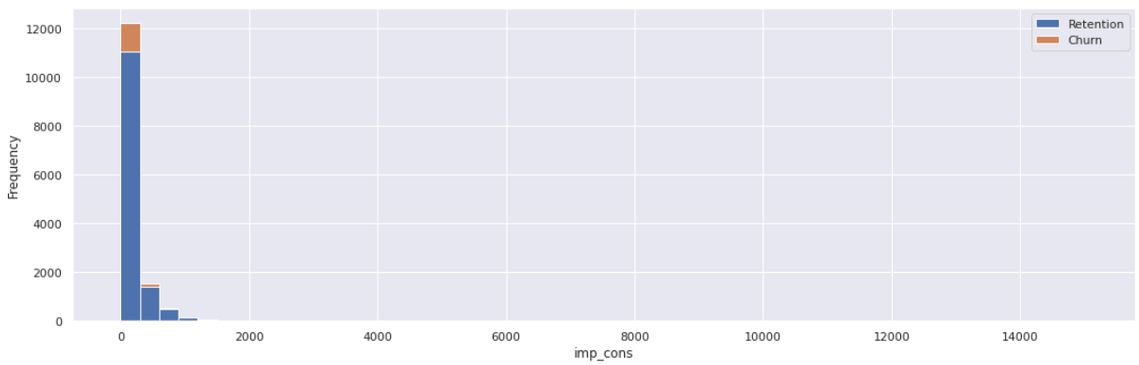
value x is transformed to value log(1+x)

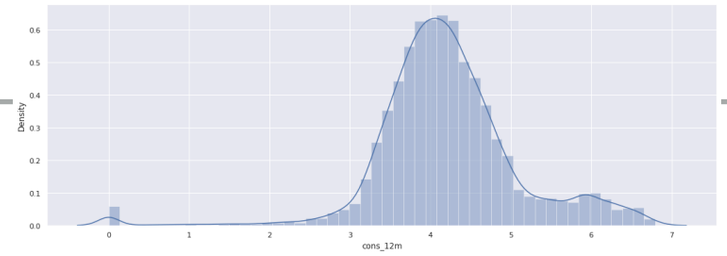
SKEWED DATA

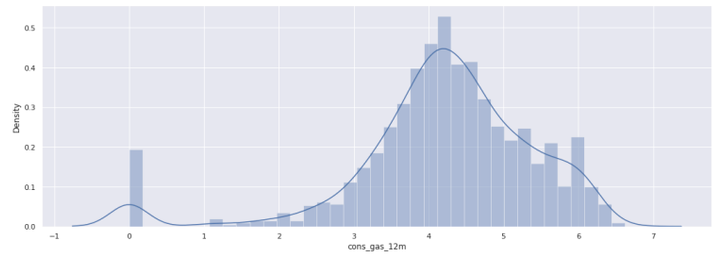

Retention
Churn
Frequency
cons_12m

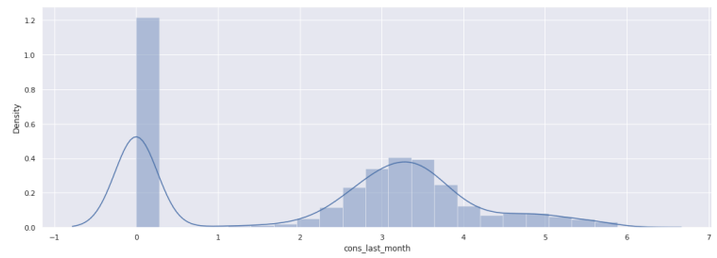

Retention
Churn
Frequency
cons_gas_12m

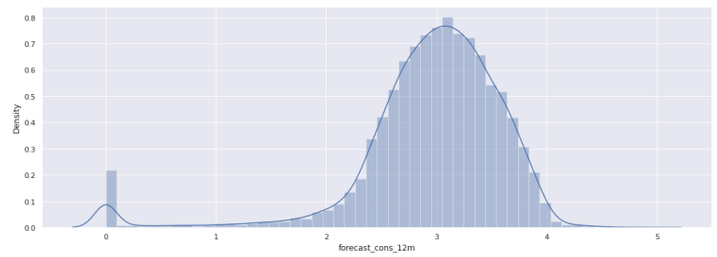

Retention
Churn
Frequency
cons_last_month

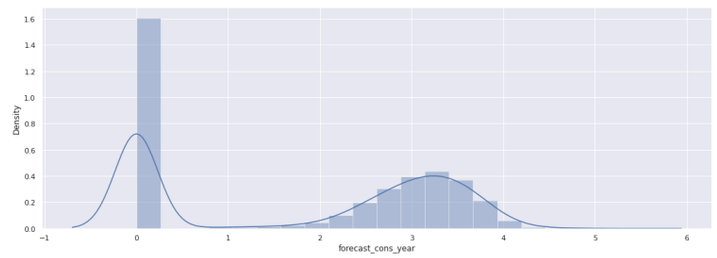

Retention
Churn
Frequency
imp_cons

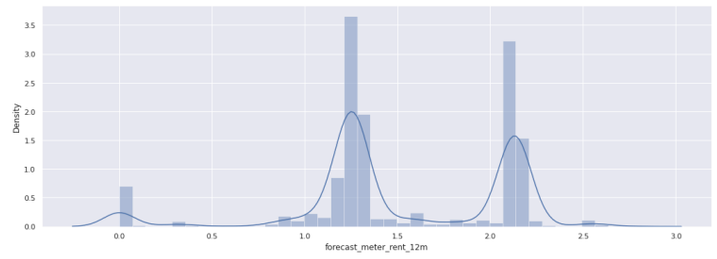

Density
cons_12m

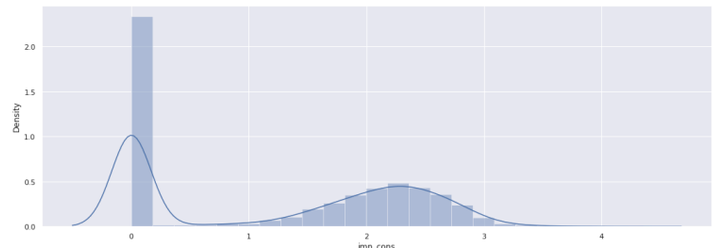

Density
cons_gas_12m


Density
cons_last_month


Density
forecast_cons_12m


Density
forecast_cons_year

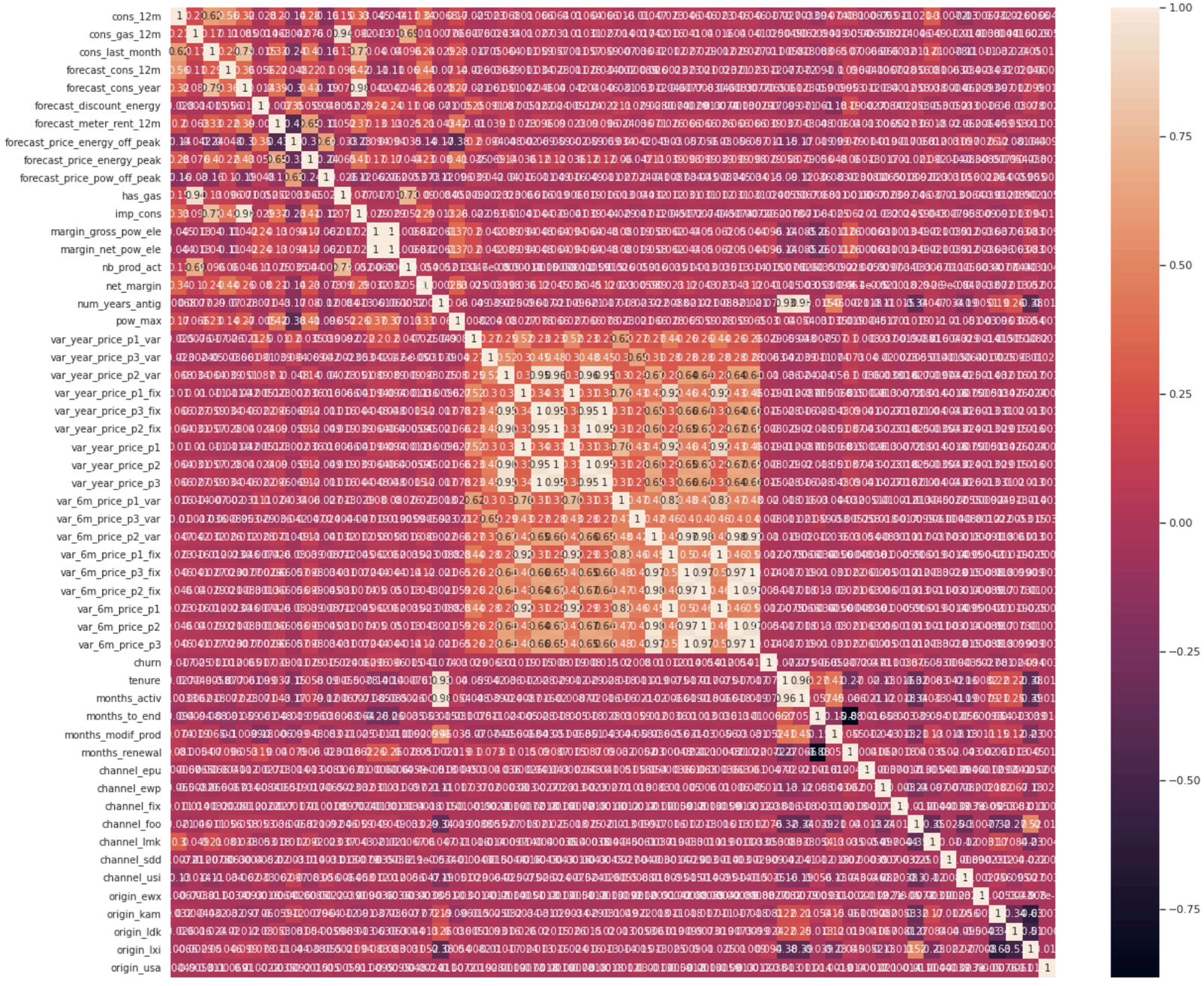

Density
forecast_meter_rent_12m


Density
imp_cons

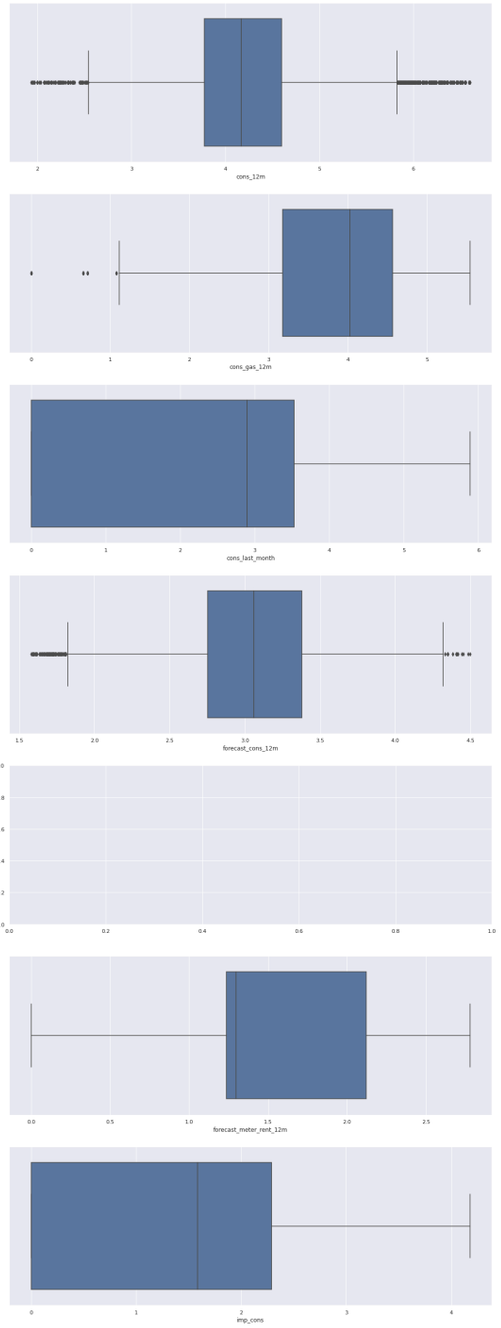
MULTICOLINERITY

All these price variables are highly correlated variables. We can remove highly correlated variables.Multicollinearity happens when one predictor variable in a multiple regression model can be linearly predicted from the others with a high degree of accuracy. This can lead to skewed or misleading results. Luckily, decision trees and boosted trees algorithms are immune to multicollinearity by nature. When they decide to split, the tree will choose only one of the perfectly correlated features. However, other algorithms like Logistic Regression or Linear Regression are not immune to that problem and should be fixed before training the model.

MULTICOLINERITY



OUTLIER REMOVAL

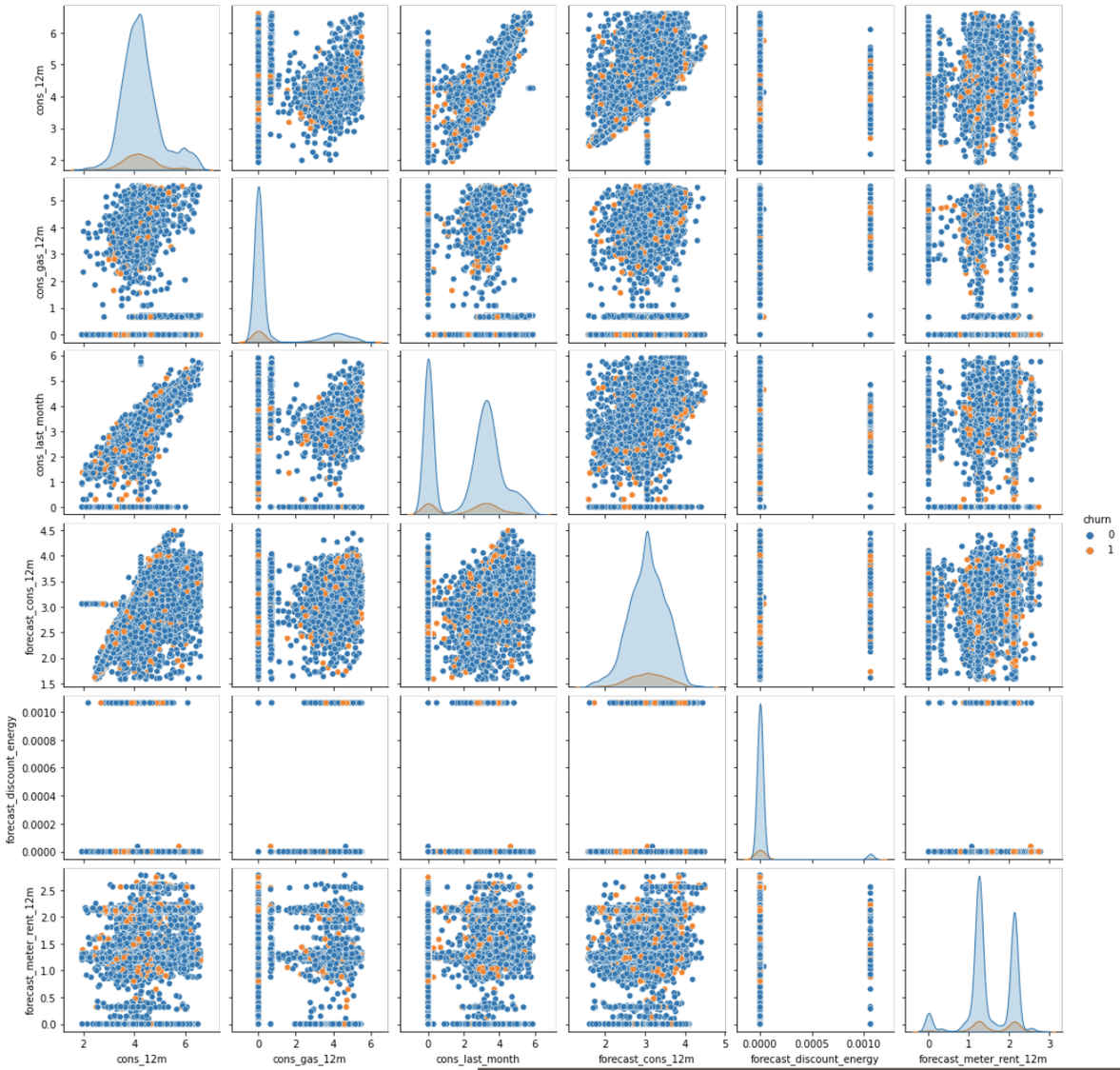


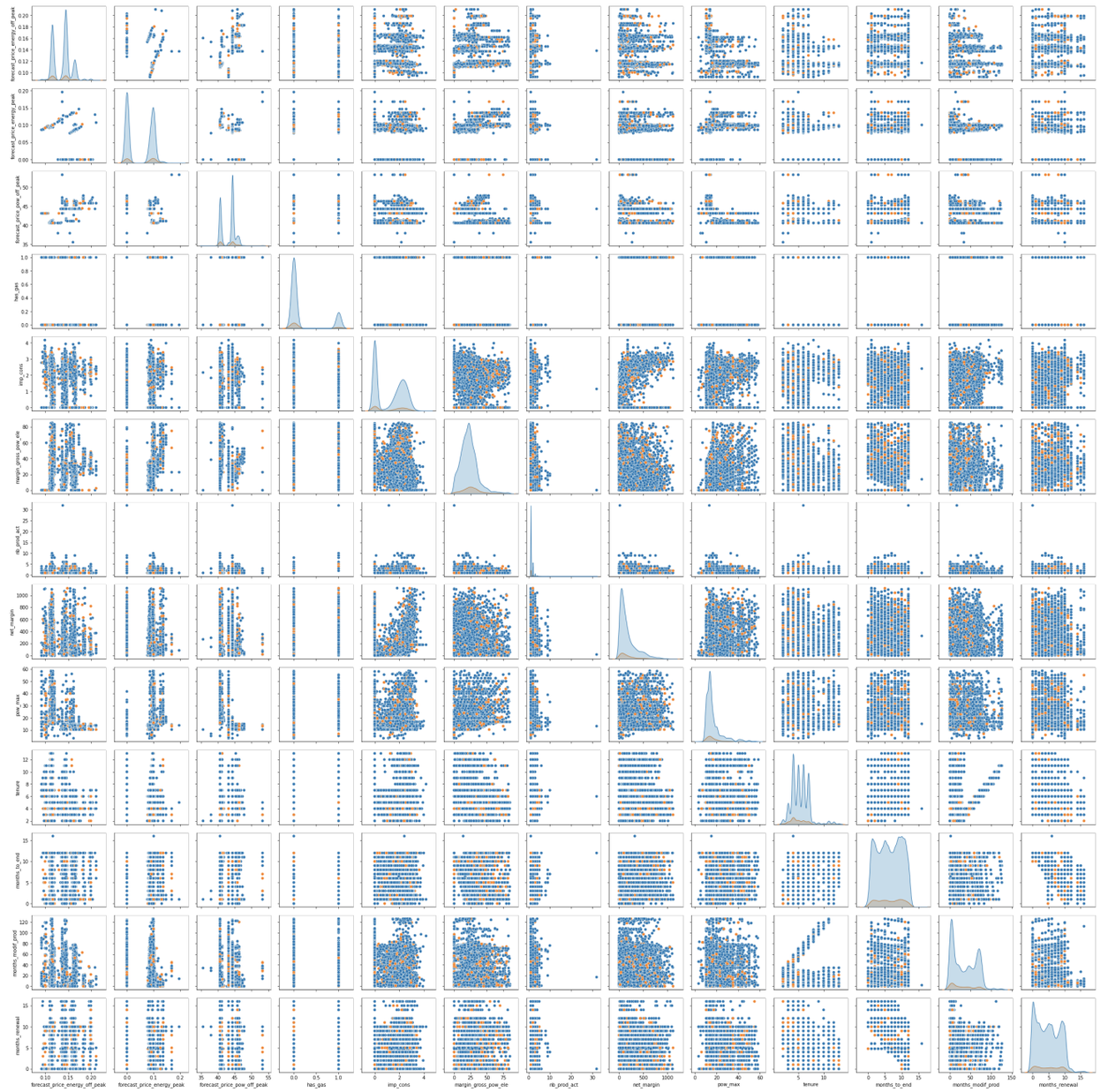
There are several ways to handle with those outliers such as removing them (this works well for massive datasets) or replacing them with sensible data(works better when the dataset is not that big).We will replace the outliers with the mean (average of the values excluding outliers). We could also replace them by median as median is not prone to outlier effect.

MODELLING AND EVALUATION

POWER CO.

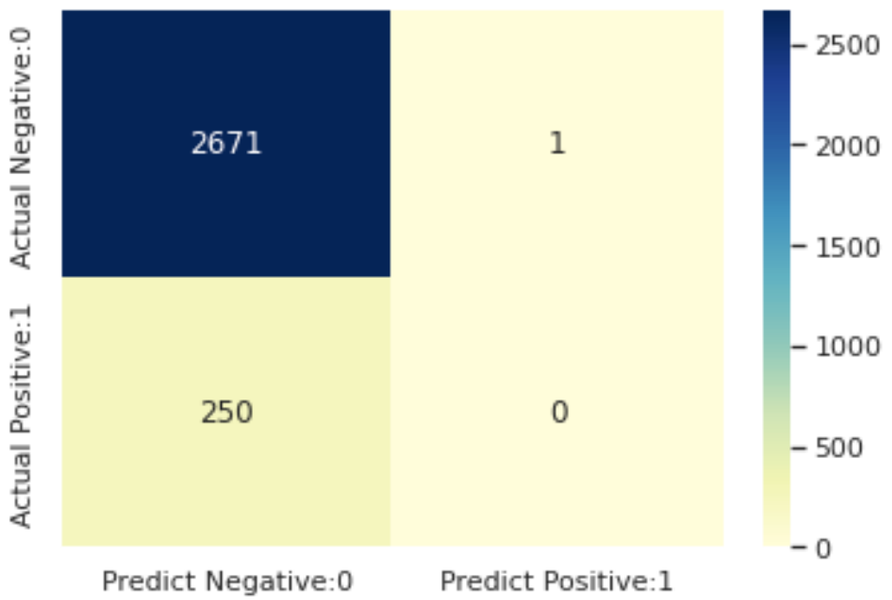
WHICH MODEL TO APPLY ?



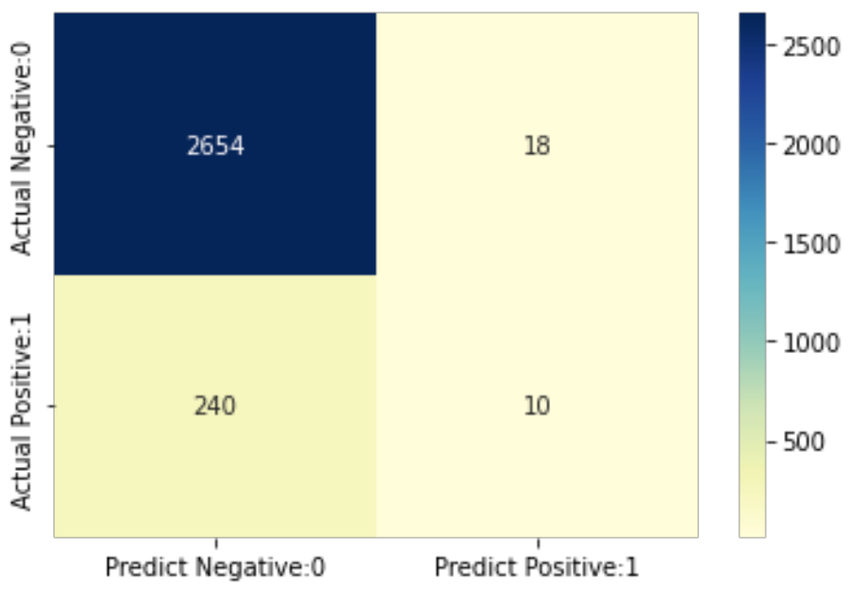


FROM THE PAIR PLOTS WE CAN UNDERSTAND THAT WE HAVE A VERY OVERLAPPING WITH NO CLEAR DISTINCTION. HENCE ANY ALGORITHM THAT USES LINEAR SEPARATION OR CLUSTERING MIGHT NOT BE A GOOD IDEA FOR THIS CASE. WE MAY GO WITH ENSEMBLE ALGORITHMS WHICH ARE DECISION TREE BASED ALGORITHMS.

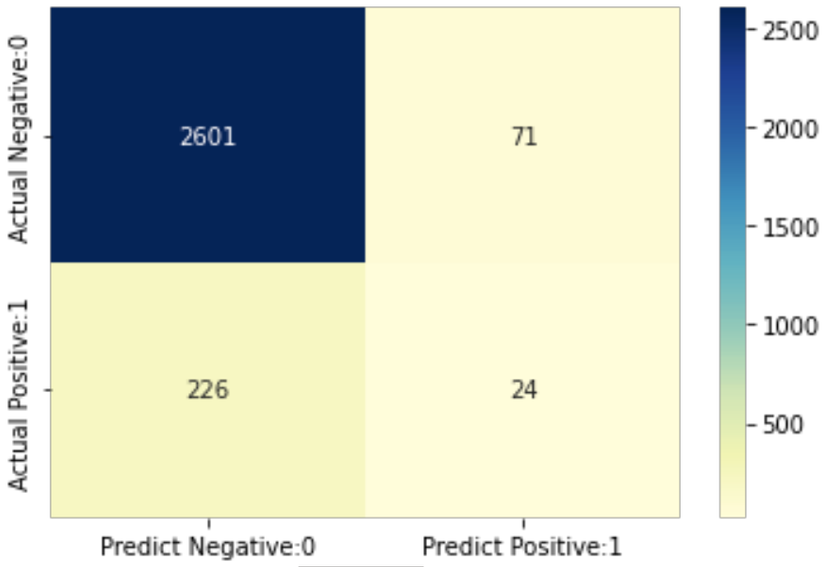
LETS HAVE A LOOK AT THE RESULTS OF ALL BASE CLASSIFICATION MODELS



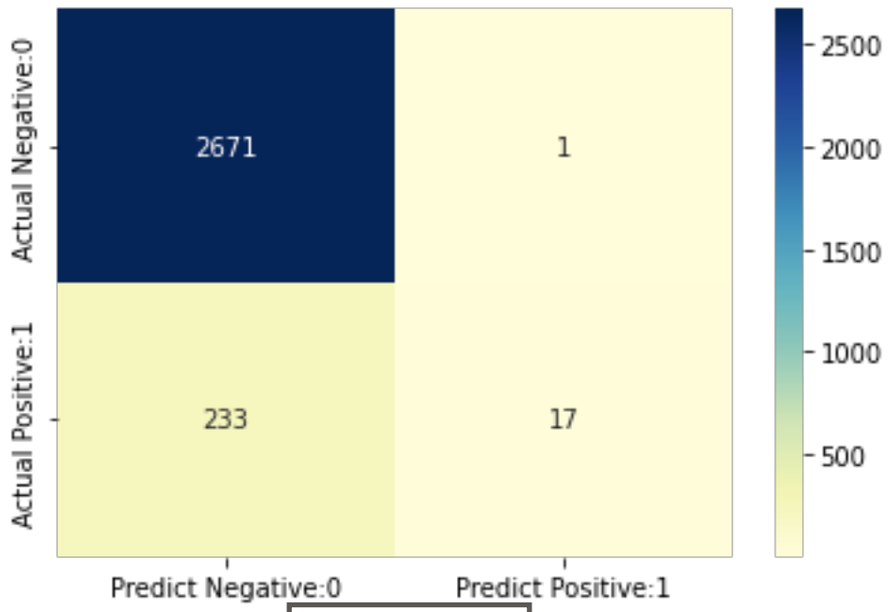
LOGISTIC REGRESSION



SVM

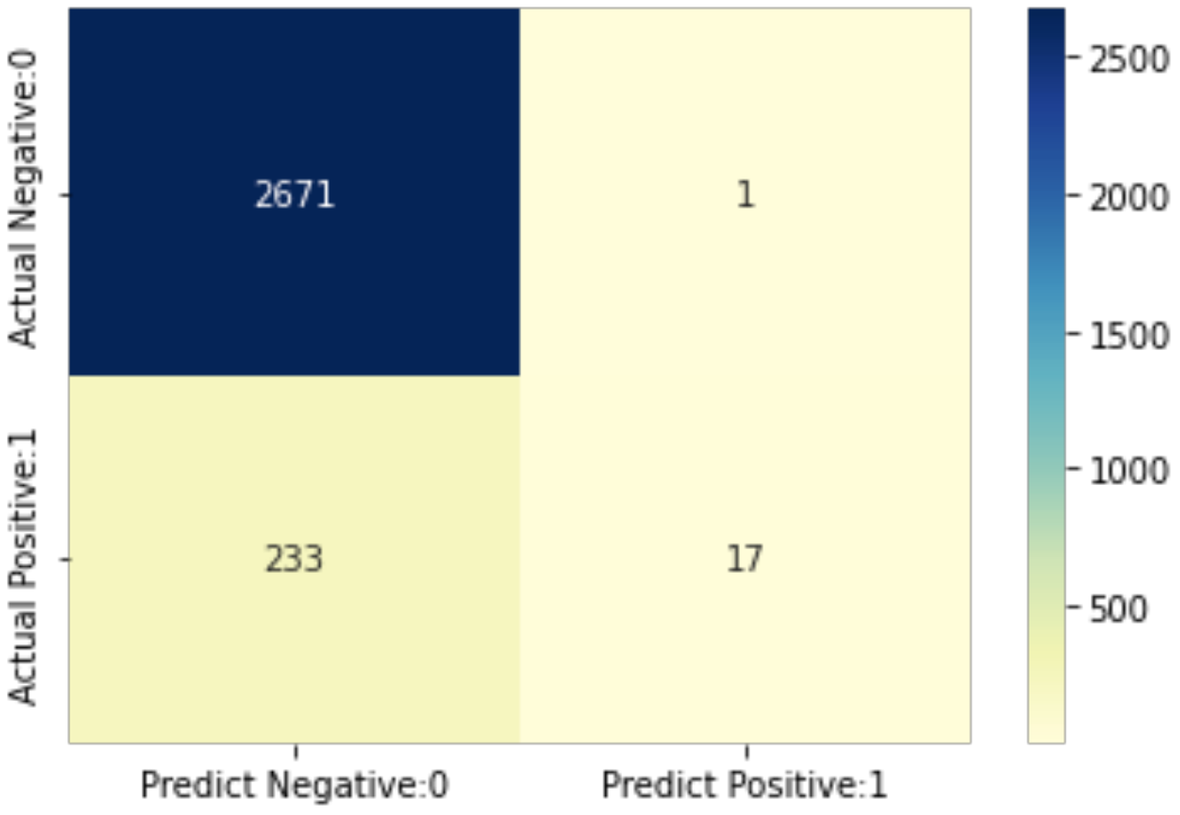


KNN

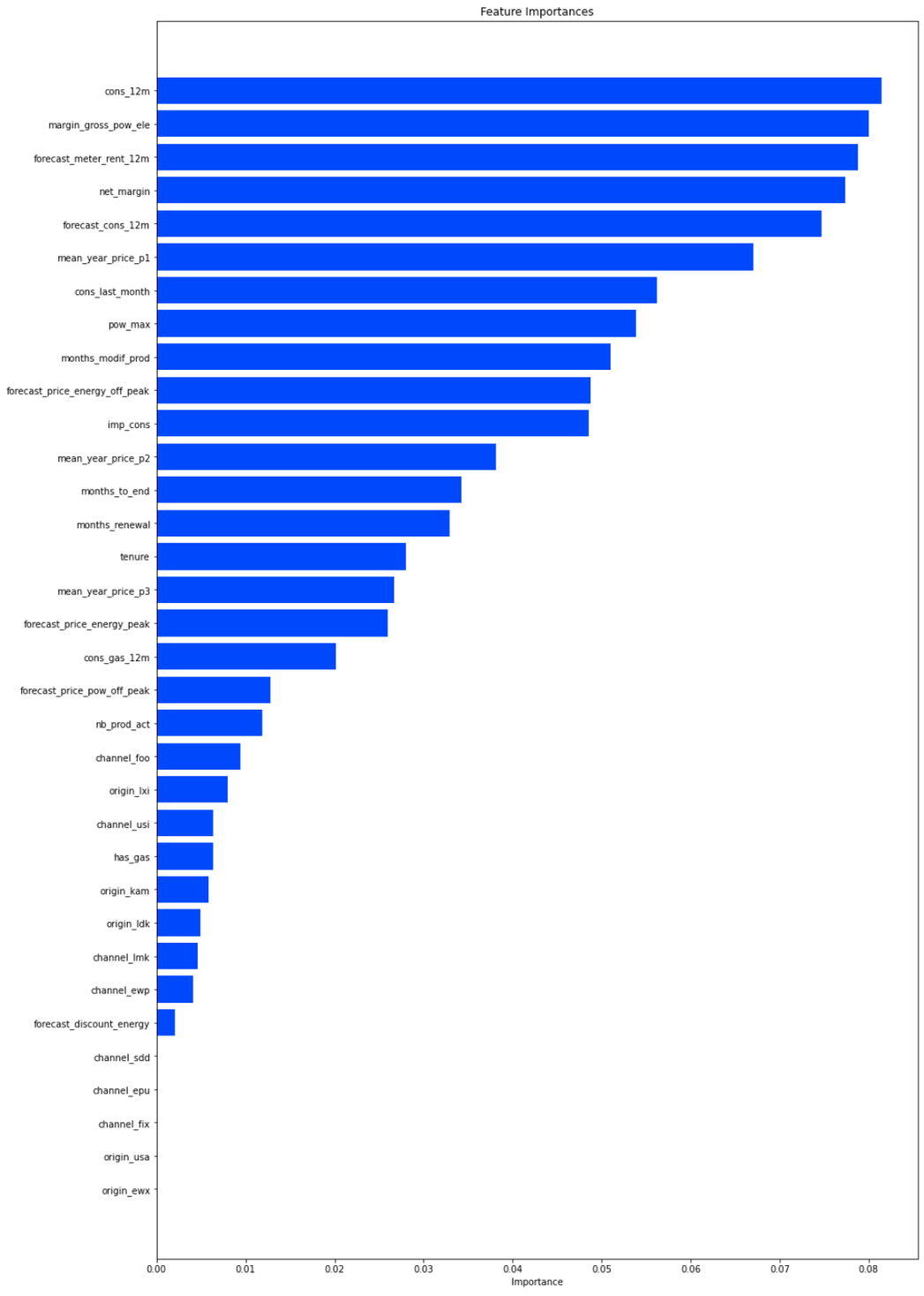


XG BOOST

LETS HAVE A LOOK AT THE RESULTS OF ALL BASE CLASSIFICATION MODELS



RANDOM FOREST



UNDERSTANDING FROM BASE CLASSIFICATION MODEL

- Accuracy of every model is good but the real issue lies within the precision of the models which is very poor. In case of linearly separable models like logistic regression we can understand that why this is happening as our data is not linearly separable. Why accuracy is not a very good metrics in imbalanced data?
- SVM we have used polynomial kernel as we could not converge on the linear kernel.
- There could be two more issues within our batch for the failure of XG Boost and Random Forest :
    - Imbalanced Dataset
    - Feature's obtained not being predictive enough.
- Imbalanced dataset is one of the biggest issues that makes our models biased. Lets deal with the imbalance and check our models again.

HOW TO DEAL WITH IMBALANCED DATA ?

- There are two primary ways of dealing with Imbalanced Dataset
  - Undersampling : Random Undersampler
  - Oversampling : Random Oversampler
    - Mixture of Undersampling and Oversampling : SMOTETomek
- The problem with under sampling is that we may loose some critical information from our data.
- The problem with over sampling is that it may lead to over fitting of our data.

DEALING WITH IMBALANCED DATASET

```
[ ] from imblearn.under_sampling import RandomUnderSampler
    under_sampler = RandomUnderSampler()
    X_under, y_under = under_sampler.fit_resample(X_train, y_train)
```

```
[ ] y_under.value_counts()
```

```
0    1135
1    1135
Name: churn, dtype: int64
```

```
from imblearn.combine import SMOTETomek
# In this example I use SMOTETomek which is a method of imblearn.
# which uses an under sampling method (Tomek) in with an over samp
os_us = SMOTETomek()

X_over_smote, y_over_smote = os_us.fit_resample(X_train, y_train)
```

```
y_over_smote.value_counts()
```

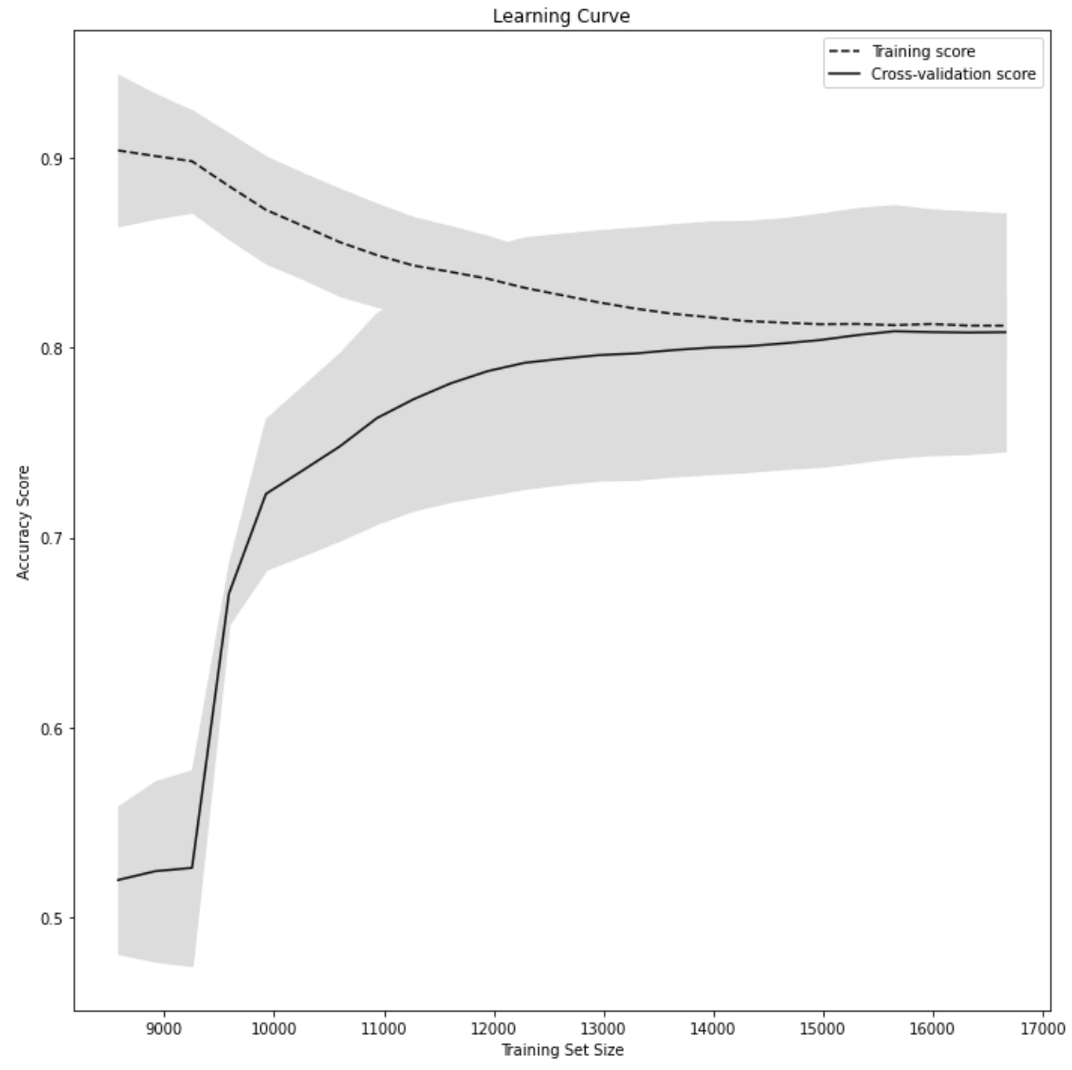
```
0    10423
1    10423
Name: churn, dtype: int64
```

```
[ ] ## RandomOverSampler to handle imbalanced data
    from imblearn.over_sampling import RandomOverSampler
    os =  RandomOverSampler()
    X_over, y_over = os.fit_resample(X_train, y_train)
```

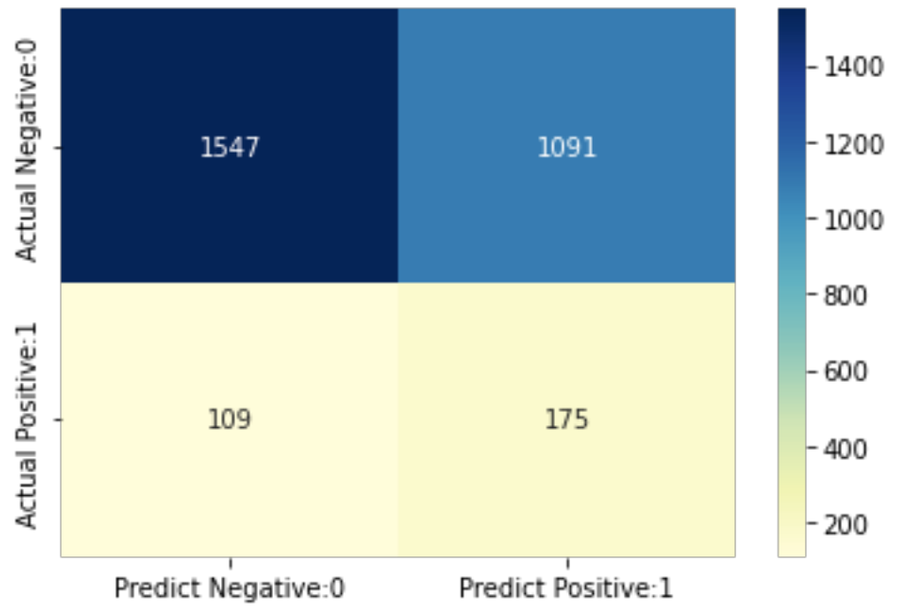
```
y_over.value_counts()
```

```
0    10549
1    10549
Name: churn, dtype: int64
```

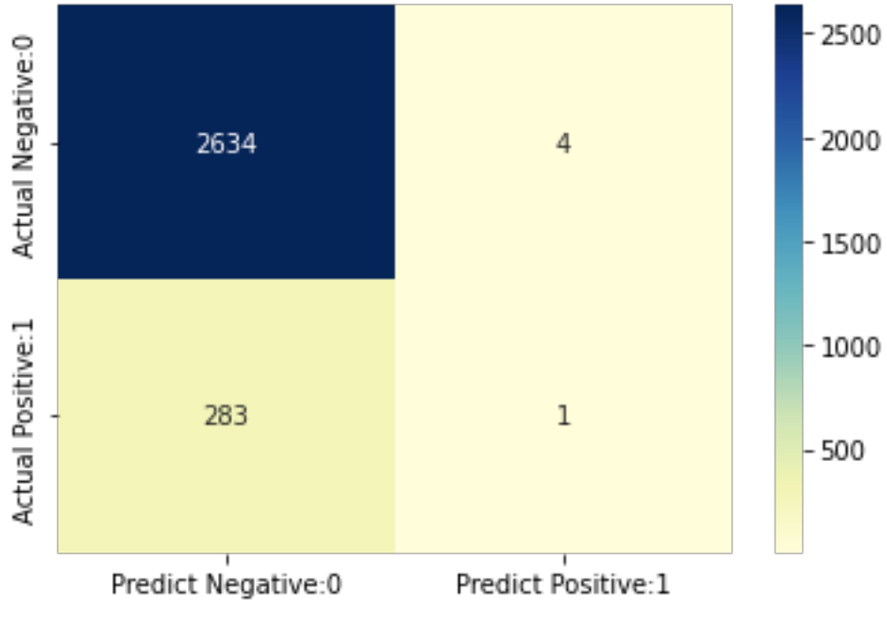
LOGISTIC REGRESSION



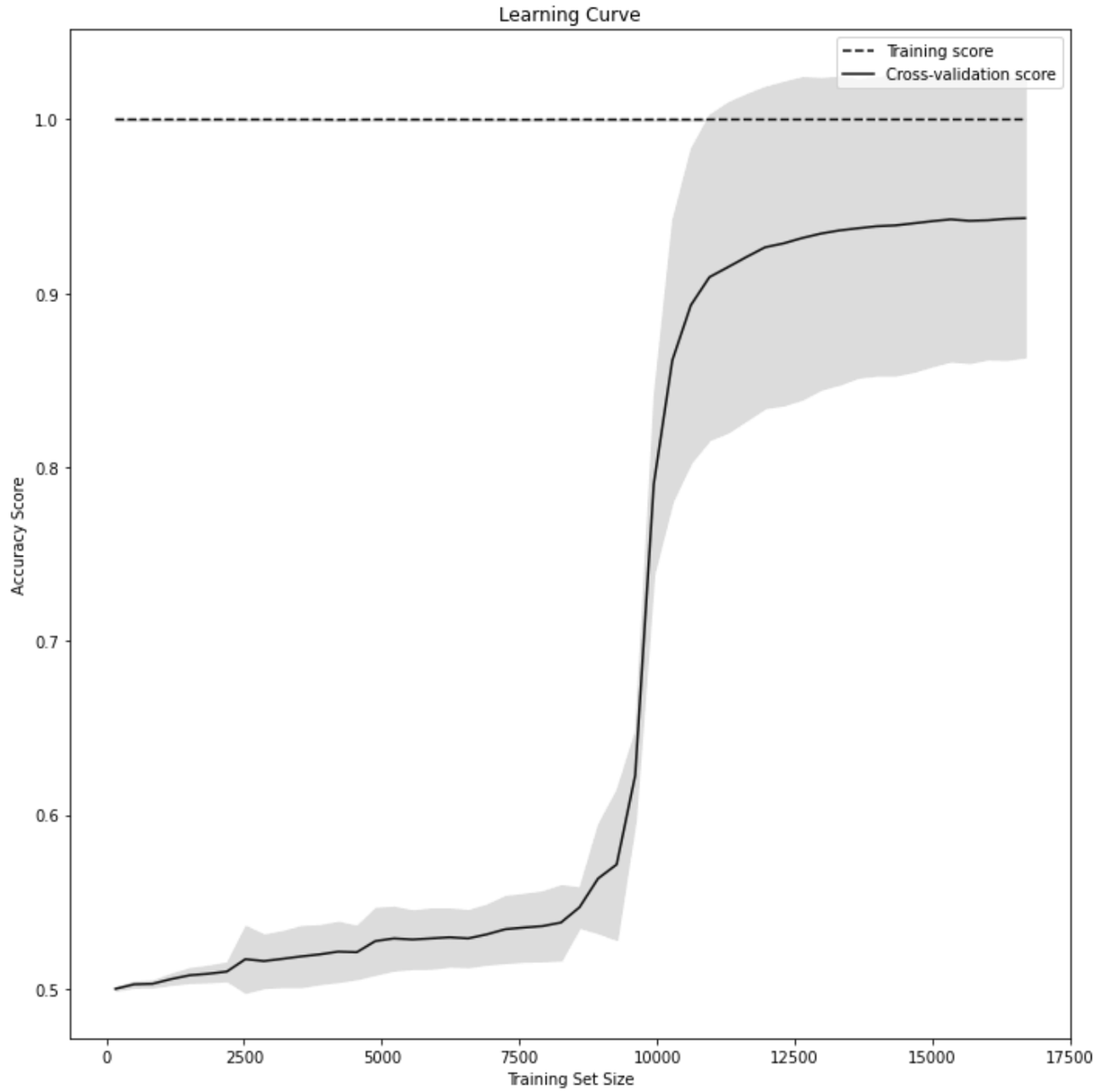
OVER SAMPLED DATA



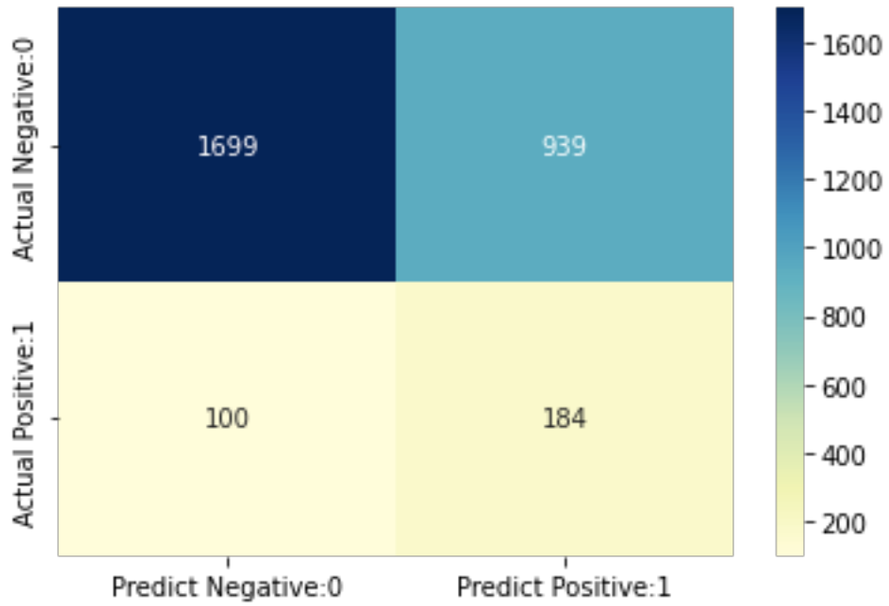
UNDER SAMPLED DATA



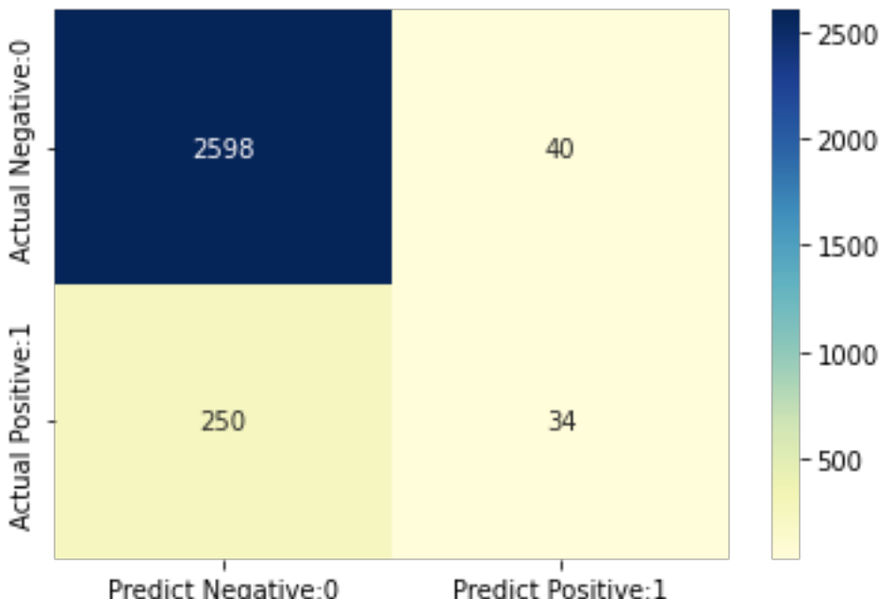
RANDOM FOREST



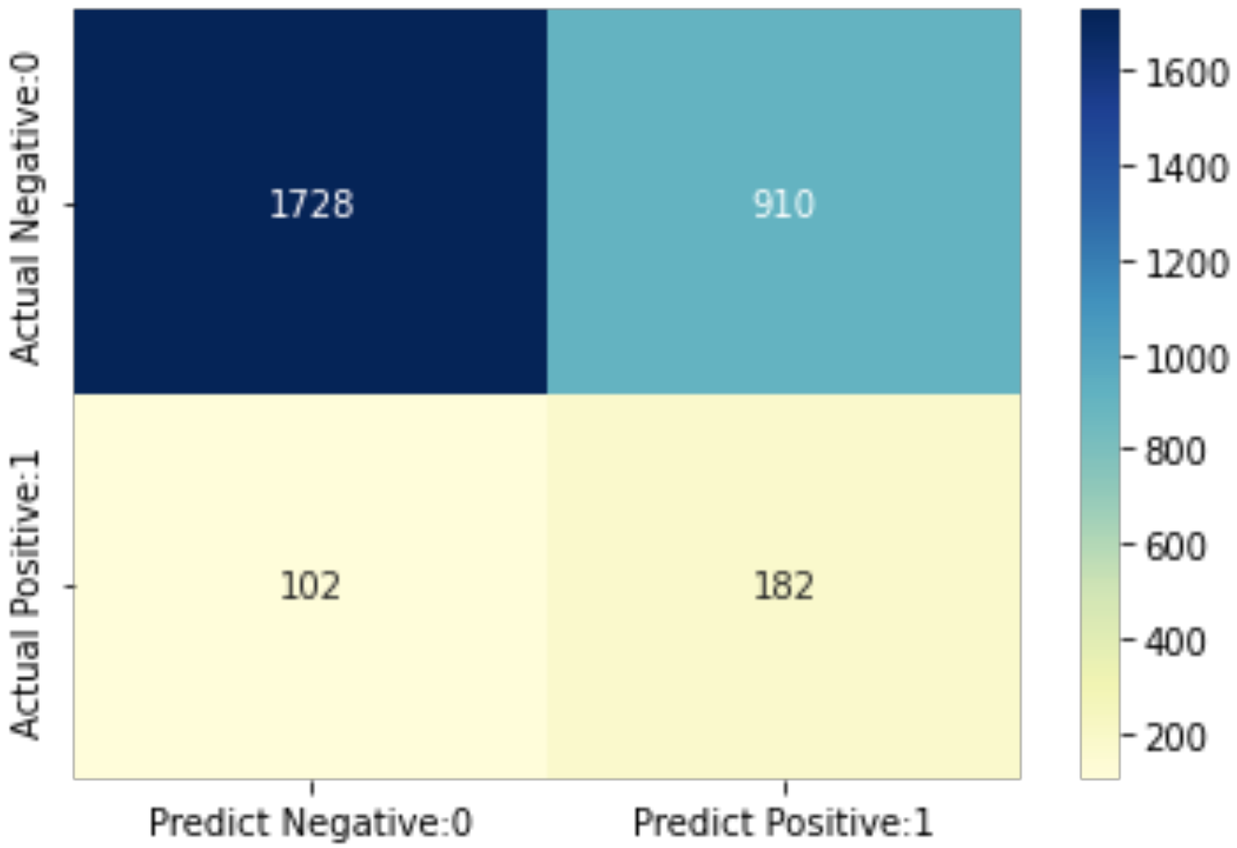
OVER SAMPLED DATA



UNDER SAMPLED DATA

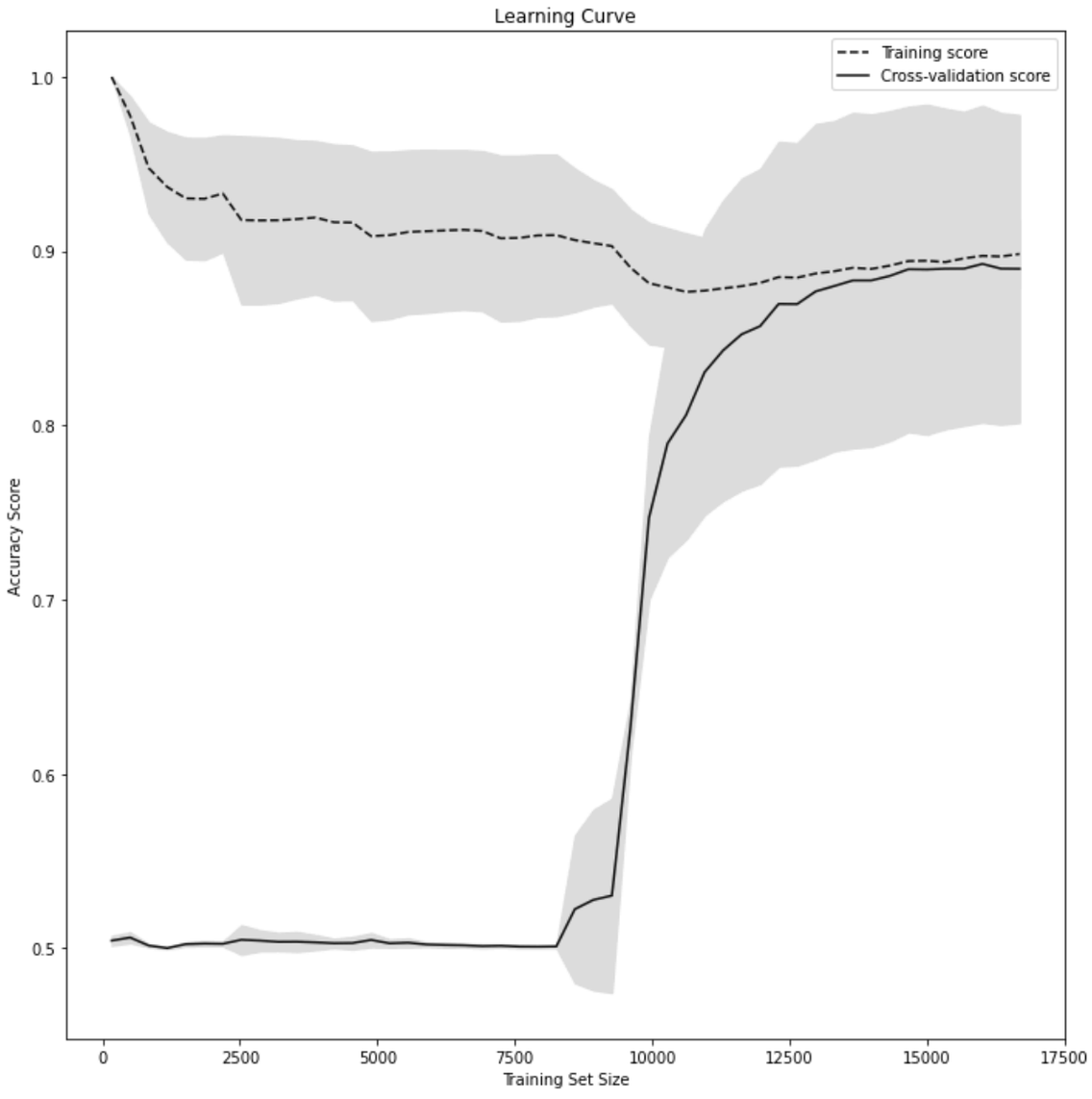


BALANCED RANDOM FOREST

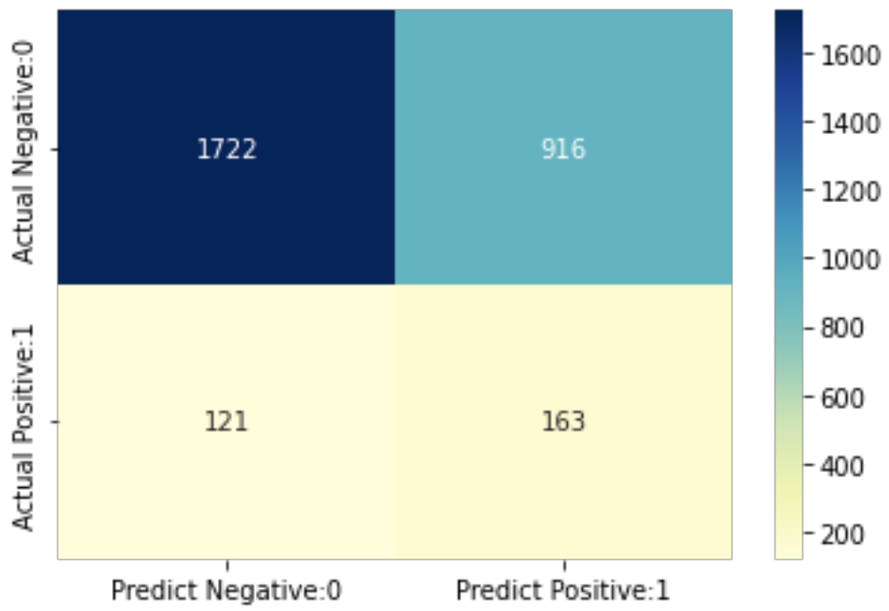


BALANCED RANDOM FOREST IS A METHODOLOGY DEFINED UNDER SKLEARN.IMBLEARN TO DEAL WITH IMBALANCED DATA. IT DEALS WITH THE WAY THE BAGGING IS PERFORMED OVER THE IMBALANCED DATA. WE GET SIMILAR RESULTS IN THIS CASE AS WE GET IN CASE OF RANDOM FOREST WITH UNDER SAMPLED DATA. THIS METHODS GIVES US BETTER PRECISION BUT POOR RECALL.

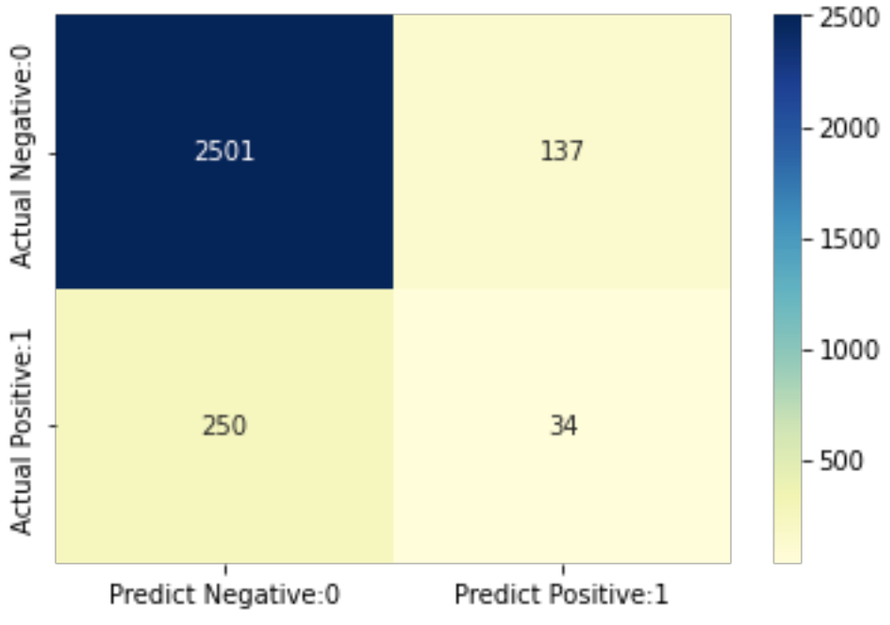
XG BOOST CLASSIFIER



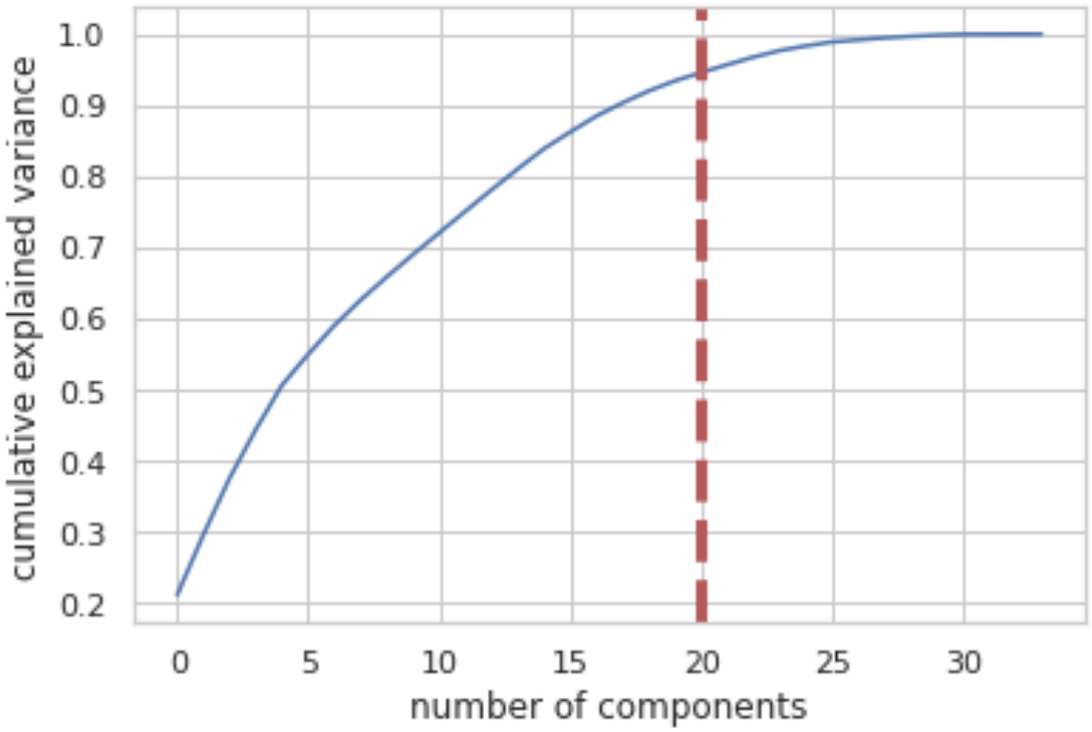
OVER SAMPLED DATA



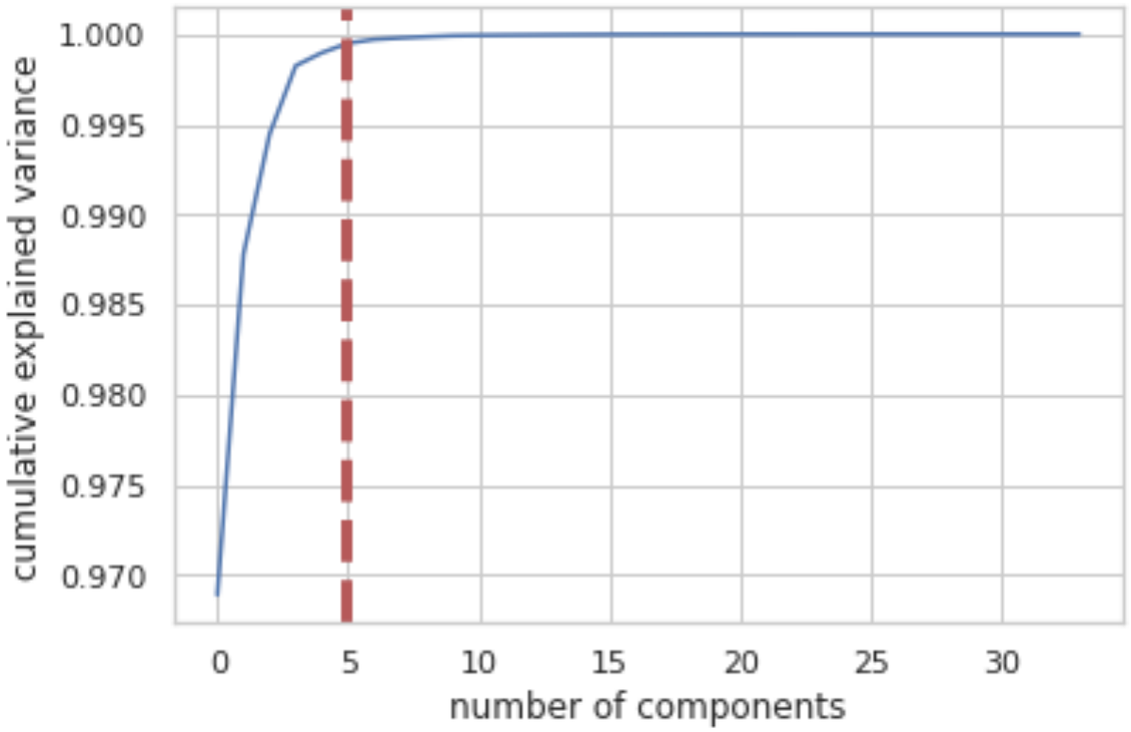
UNDER SAMPLED DATA



FEATURES NOT BEING PREDICTIVE ENOUGH : PCA WITH RANDOM FOREST



PCA WITH BASE DATA



PCA WITH UNDER SAMPLED DATA

THE PRECISION RESULTS WITH GET BY HAVING 25 COMPONENTS PCA WITH BASE DATA AND 5 COMPONENTS PCA WITH UNDER SAMPLED DATA ARE NOT QUALIFIED TO BE ACCEPTED. PRECISION IS VERY LOW FOR BOTH OF THE CASES. HENCE WE CAN UNDERSTAND THAT OUR PRINCIPLE COMPONENTS ARE NOT THAT PREDICTIVE OF CHURN AND NON CHURN FOR THE POWER COMPANIES. WE MAY NEED AN ADVICE OF SUBJECT MATTER EXPERT TO ENSURE WE HAVE DEFINED MORE PREDICTIVE FEATURES FOR OUR CHURN AND NON CHURN RATE. IF WE CLUB THIS ACTIVITY WITH OUR UNDER AND OVER SAMPLING WE MAY OBTAIN GOOD RESULTS FOR OUR MODEL.

EXECUTIVE SUMMARY

- Predictive model is able to predict churn but the main driver is not customer price sensitivity. Also the model best obtained precision is near to 0.25 which is not a good precision to have at a higher accuracy of 0.9 in case of an imbalanced dataset.
  - Yearly consumption, forecasted consumption and net margin are the 3 largest drivers.
  - There can be a better predictive model with more data available for churn cases by the client and with the help of SME to design better features.
- CHURN is high in SME division to the tune of 9.7% . Also there are some missing values for the channel sales which account for a churn of 7.6%. We must asked our client for these values for better understanding of the data.