

Part 1 - Aggregate Service Incidents, Interactions, Cases and Install Base

February 12, 2021

```
[1]: # import relevant libraries
import pandas as pd
import scipy.stats
import numpy as np
import datetime as dt
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('float_format', '{:.2f}'.format)
```

1 Importing datasets

```
[2]: # Reading data tables
cases = pd.read_csv('Cases.csv')
interactions = pd.read_csv('Interactions.csv')
installbase = pd.read_csv('Install_Base.csv')
service = pd.read_csv('Incidents.csv')
```

2 Creating aggregated features

Before join with revenue table, we created aggregated features for Service Incidents, Cases, Interactions and Install Base table.

2.1 Service Incidents

```
[3]: service.head()
```

```
[3]:   NO_OF_SVC INCIDENTS GROUP_NAME INCIDENT_YEAR LAST_SERVICE_EVENT_DATE \
0           16      WT03        2015          9/8/2015
1           15      WT05        2015         10/31/2015
2           25      NE04        2015         12/14/2015
```

```

3           17      WT07        2015      10/30/2015
4          100      WT03        2015      12/29/2015

NO_OF_REPEAT_CALLS_30D NO_OF_FTF_CALLS_30D CUSTOMER_SITE_ID DUNS_NUMBER \
0                      6                  10          183312    792229767
1                      8                  7           519632    78711799
2                      9                  16          67170     2147445
3                      6                  11          62480     152543450
4                     35                 65          175434   55634273

SFDC_ID
0 0018000000drGQYAA2
1 001C0000010IQIfIAO
2 0018000000drI9QAAU
3 0018000000WFCRGAA1
4 0018000000drGBmAAM

```

[4]: service.shape

[4]: (42774, 9)

2.1.1 Preprocessing before aggregating

[5]: # check duplicates
service.drop_duplicates(inplace = True)
service.shape

[5]: (42774, 9)

[6]: # check column names and null values
service.info()

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 42774 entries, 0 to 42773
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   NO_OF_SVC INCIDENTS      42774 non-null   int64  
 1   GROUP_NAME          42774 non-null   object  
 2   INCIDENT_YEAR       42774 non-null   int64  
 3   LAST_SERVICE_EVENT_DATE 42774 non-null   object  
 4   NO_OF_REPEAT_CALLS_30D 42774 non-null   int64  
 5   NO_OF_FTF_CALLS_30D   42774 non-null   int64  
 6   CUSTOMER_SITE_ID     42774 non-null   int64  
 7   DUNS_NUMBER          42774 non-null   int64  
 8   SFDC_ID              42774 non-null   object  
dtypes: int64(6), object(3)

```

```
memory usage: 3.3+ MB
```

```
[7]: # drop unnecessary columns
service.drop(['DUNS_NUMBER', 'SFDC_ID', 'GROUP_NAME'], axis = 1, inplace = True)
```

```
[8]: service.head()
```

```
[8]:   NO_OF_SVC INCIDENTS INCIDENT_YEAR LAST_SERVICE_EVENT_DATE \
0           16      2015          9/8/2015
1           15      2015         10/31/2015
2           25      2015         12/14/2015
3           17      2015         10/30/2015
4          100      2015         12/29/2015

  NO_OF_REPEAT_CALLS_30D NO_OF_FTF_CALLS_30D CUSTOMER_SITE_ID
0                  6                 10        183312
1                  8                  7        519632
2                  9                 16        67170
3                  6                 11        62480
4                 35                 65       175434
```

2.1.2 Aggregating

```
[9]: # aggregate NO_OF_SVC INCIDENTS, NO_OF_REPEAT_CALLS_30D, NO_OF_FTF_CALLS_30D, ↴
      ↴LAST_SERVICE_EVENT_DATE
service = service.groupby('CUSTOMER_SITE_ID').agg({'NO_OF_SVC INCIDENTS': 'sum',
                                                    'NO_OF_REPEAT_CALLS_30D': ↴
                                                    ↴'sum',
                                                    'NO_OF_FTF_CALLS_30D': ↴
                                                    ↴'sum'}).reset_index()
service.head()
```

```
[9]:   CUSTOMER_SITE_ID NO_OF_SVC INCIDENTS NO_OF_REPEAT_CALLS_30D \
0           24            13                 7
1           36           111                38
2           85             1                 0
3           86             11                 1
4           90             57                13

  NO_OF_FTF_CALLS_30D
0                  6
1                 73
2                  1
3                 10
4                 44
```

```
[10]: service.shape
```

```
[10]: (13581, 4)
```

```
[11]: # rename columns
service = service.rename(columns = {
                                "NO_OF_SVC INCIDENTS": "Total_SVC_Incidents",
                                "NO_OF_REPEAT_CALLS_30D": "Total_Repeat_Calls",
                                "NO_OF_FTF_CALLS_30D": "Total_FTF_Calls"})

```

```
[12]: service.head()
```

```
[12]:   CUSTOMER_SITE_ID  Total_SVC_Incidents  Total_Repeat_Calls  Total_FTF_Calls
0                24                  13                   7                  6
1                36                 111                  38                 73
2                85                  1                   0                  1
3                86                  11                  1                  10
4                90                  57                  13                 44
```

```
[13]: # check all customer sites are unique
len(service['CUSTOMER_SITE_ID'])/len(service['CUSTOMER_SITE_ID'].unique())
```

```
[13]: 1.0
```

2.2 Interactions

```
[14]: interactions.head()
```

```
[14]: SFDC_ID Task_Subtype          Created_Date \
0 0018000000drGW3AAM      Call 2019-04-16 19:36:23.0000000
1 0018000000drGW3AAM      Call 2019-06-11 20:58:44.0000000
2 0018000000drGW3AAM      Call 2019-06-12 14:35:19.0000000
3 0018000000drGW3AAM      Call 2019-06-12 14:25:22.0000000
4 0018000000drGW3AAM      Call 2019-07-18 15:43:21.0000000

Interaction_Amount  Type          Assigned  Visit  Task \
0             nan  Call  Stephen Czekanski  1     1
1             nan  Call        April Clark  1     1
2             nan  Call  Stephen Czekanski  1     1
3             nan  Call  Stephen Czekanski  1     1
4             nan  Call  Stephen Czekanski  1     1

Last_Modified_Date  CUSTOMER_SITE_ID
0 2019-04-16 19:47:43.0000000  196,773.00
1 2019-06-11 20:58:44.0000000  196,773.00
2 2019-06-12 14:47:03.0000000  196,773.00
3 2019-06-12 14:31:19.0000000  196,773.00
```

```
4 2019-07-18 15:48:19.0000000 196,773.00
```

```
[15]: interactions.shape
```

```
[15]: (905864, 10)
```

2.2.1 Preprocessing before aggregating

```
[16]: # check duplicates  
interactions.drop_duplicates(inplace = True)  
interactions.shape
```

```
[16]: (894839, 10)
```

```
[17]: # check column names and null values  
interactions.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 894839 entries, 0 to 905863  
Data columns (total 10 columns):  
 #   Column           Non-Null Count  Dtype     
---  --  
 0   SFDC_ID          894839 non-null  object    
 1   Task_Subtype     894839 non-null  object    
 2   Created_Date     894839 non-null  object    
 3   Interaction_Amount 0 non-null    float64  
 4   Type              792292 non-null  object    
 5   Assigned          894839 non-null  object    
 6   Visit              894839 non-null  int64     
 7   Task               894839 non-null  int64     
 8   Last_Modified_Date 894839 non-null  object    
 9   CUSTOMER_SITE_ID   652467 non-null  float64  
dtypes: float64(2), int64(2), object(6)  
memory usage: 75.1+ MB
```

```
[18]: # drop unnecessary columns  
interactions.drop('Interaction_Amount', axis = 1, inplace = True)
```

```
[19]: # fill in null values in Type  
interactions['Type'] = interactions['Type'].fillna('Other')
```

```
[20]: # drop rows with null values  
interactions = interactions.dropna()
```

```
[21]: interactions.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 652467 entries, 0 to 905863
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   SFDC_ID          652467 non-null   object  
 1   Task_Subtype     652467 non-null   object  
 2   Created_Date     652467 non-null   object  
 3   Type             652467 non-null   object  
 4   Assigned         652467 non-null   object  
 5   Visit            652467 non-null   int64  
 6   Task              652467 non-null   int64  
 7   Last_Modified_Date 652467 non-null   object  
 8   CUSTOMER_SITE_ID 652467 non-null   float64 
dtypes: float64(1), int64(2), object(6)
memory usage: 49.8+ MB

```

[22]: `interactions.head()`

```

[22]: 
      SFDC_ID Task_Subtype           Created_Date  Type \
0  0018000000drGW3AAM    Call  2019-04-16 19:36:23.0000000  Call
1  0018000000drGW3AAM    Call  2019-06-11 20:58:44.0000000  Call
2  0018000000drGW3AAM    Call  2019-06-12 14:35:19.0000000  Call
3  0018000000drGW3AAM    Call  2019-06-12 14:25:22.0000000  Call
4  0018000000drGW3AAM    Call  2019-07-18 15:43:21.0000000  Call

      Assigned  Visit  Task           Last_Modified_Date \
0  Stephen Czekanski     1     1  2019-04-16 19:47:43.0000000
1        April Clark     1     1  2019-06-11 20:58:44.0000000
2  Stephen Czekanski     1     1  2019-06-12 14:47:03.0000000
3  Stephen Czekanski     1     1  2019-06-12 14:31:19.0000000
4  Stephen Czekanski     1     1  2019-07-18 15:48:19.0000000

      CUSTOMER_SITE_ID
0            196,773.00
1            196,773.00
2            196,773.00
3            196,773.00
4            196,773.00

```

[23]: `# drop unnecessary columns`
`interactions.`
`→drop(['Task_Subtype','Last_Modified_Date','SFDC_ID','Assigned','Created_Date'],axis=`
`→= 1, inplace = True)`

[24]: `interactions.head()`

```
[24]:   Type Visit Task CUSTOMER_SITE_ID
0 Call    1     1      196,773.00
1 Call    1     1      196,773.00
2 Call    1     1      196,773.00
3 Call    1     1      196,773.00
4 Call    1     1      196,773.00
```

2.2.2 Aggregating

```
[25]: # aggregate Type
interactions1 = interactions[['CUSTOMER_SITE_ID', 'Type']]
interactions2 = interactions1.groupby('CUSTOMER_SITE_ID')['Type'].apply(lambda x: x.mode()[0]).reset_index()
interactions2.head()
```

```
[25]:   CUSTOMER_SITE_ID  Type
0          20.00  Other
1          24.00   Call
2          29.00  Dial
3          31.00   Call
4          36.00   Call
```

```
[26]: # # check all customer sites are unique
len(interactions2['CUSTOMER_SITE_ID'])/len(interactions2['CUSTOMER_SITE_ID'].
unique())
```

```
[26]: 1.0
```

```
[27]: # aggregate Visit and Task
interactions = interactions.drop('Type', axis = 1)
interactions3 = interactions.groupby('CUSTOMER_SITE_ID')[['Visit', 'Task']].sum().reset_index()
interactions3.head()
```

```
[27]:   CUSTOMER_SITE_ID  Visit  Task
0          20.00      4     7
1          24.00     81    73
2          29.00      1     1
3          31.00     25    26
4          36.00     84    88
```

```
[28]: # combine aggregated variables
interactions = pd.merge(interactions2, interactions3, how = 'inner')
interactions.head()
```

```
[28]:   CUSTOMER_SITE_ID  Type  Visit  Task
      0           20.00  Other     4     7
      1           24.00  Call     81    73
      2           29.00  Dial     1     1
      3           31.00  Call     25    26
      4           36.00  Call     84    88
```

```
[29]: # rename columns
interactions = interactions.rename(columns = {"Type": "Most_Frequent_Interaction_Type",
                                             "Visit": "Total_Visits",
                                             "Task": "Total_Tasks"})
interactions.head()
```

```
[29]:   CUSTOMER_SITE_ID Most_Frequent_Interaction_Type  Total_Visits  Total_Tasks
      0           20.00                      Other        4          7
      1           24.00                      Call       81         73
      2           29.00                      Dial        1          1
      3           31.00                      Call       25         26
      4           36.00                      Call       84        88
```

```
[30]: ## check all customer sites are unique
len(interactions['CUSTOMER_SITE_ID'])/len(interactions['CUSTOMER_SITE_ID']).
unique()
```

```
[30]: 1.0
```

2.3 Cases

```
[31]: cases.sort_values(by='NO_OF_CASES', ascending=False).head(10)
```

	NO_OF_CASES	CASE_ORIGIN	CASE_REASON	CX_CASE_L1_REASON	CX_CASE_L2_REASON	MAX_MODIFIED_DATE	DUNS_NUMBER	SFDC_ID
1635	50	Email - VTI NACC	Customer Experience			NaN		
2128	10	Email - VTI NACC	Customer Experience			NaN		
2828	8	Email - VTI NACC	Customer Experience			NaN		
309	8	Email - VTI NACC	Customer Experience			NaN		
1258	7	Email - VTI NACC	Customer Experience			NaN		
1985	7	TS Survey Followup	CX: Tech Support			NaN		
980	6	Email - VTI NACC	Customer Experience			NaN		
1671	6	Email - VTI NACC	Customer Experience			NaN		
153	6	Email - VTI NACC	Customer Experience			NaN		
2011	6	Email - VTI NACC	Customer Experience			NaN		

```

309          NaN    2/7/2020 0:00 206,945,144.00 0018000000dsClhAAE
1258         NaN    2/20/2020 0:00 122,324,999.00 0011A00001XPTF4QAP
1985         NaN    2/10/2020 0:00 51,113,330.00 0018000000drEC9AAM
980          NaN    2/27/2020 0:00 115,993,883.00 001C000001J9SyxIAF
1671         NaN    3/12/2020 0:00 15,772,159.00 0018000000drJccAAE
153          NaN   12/20/2019 0:00 246,997,816.00 0018000000drEkeAAE
2011         NaN   4/23/2020 0:00 80,742,445.00 0011A00001VbI2SQAV

      CUSTOMER_SITE_ID
1635        20.00
2128       64,948.00
2828       702,114.00
309        112,471.00
1258      8,317,382.00
1985        20.00
980        736,520.00
1671       206,712.00
153        72,087.00
2011      7,096,062.00

```

[32]: cases.shape

[32]: (3027, 9)

2.3.1 Preprocessing before aggregating

```

[33]: # check duplicates
cases.drop_duplicates(inplace = True)
cases.shape

```

[33]: (3018, 9)

```

[34]: # check column names and null values
cases.info()

```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 3018 entries, 0 to 3026
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   NO_OF_CASES      3018 non-null   int64  
 1   CASE_ORIGIN      3016 non-null   object  
 2   CASE_REASON      3018 non-null   object  
 3   CX_CASE_L1_REASON 571 non-null   object  
 4   CX_CASE_L2_REASON 426 non-null   object  
 5   MAX_MODIFIED_DATE 3018 non-null   object  
 6   DUNS_NUMBER       2950 non-null   float64

```

```
7    SFDC_ID           2948 non-null   object
8    CUSTOMER_SITE_ID  2950 non-null   float64
dtypes: float64(2), int64(1), object(6)
memory usage: 235.8+ KB
```

```
[35]: # drop rows with null values in CUSTOMER_SITE_ID, CASE_ORIGIN
cases = cases.dropna(axis = 0, subset = ['CUSTOMER_SITE_ID'])
cases = cases.dropna(axis = 0, subset = ['CASE_ORIGIN'])
cases.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2948 entries, 0 to 3026
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype  
---  --  
 0   NO_OF_CASES       2948 non-null   int64  
 1   CASE_ORIGIN      2948 non-null   object  
 2   CASE_REASON      2948 non-null   object  
 3   CX_CASE_L1_REASON 555 non-null   object  
 4   CX_CASE_L2_REASON 416 non-null   object  
 5   MAX_MODIFIED_DATE 2948 non-null   object  
 6   DUNS_NUMBER       2948 non-null   float64 
 7   SFDC_ID           2946 non-null   object  
 8   CUSTOMER_SITE_ID  2948 non-null   float64 
dtypes: float64(2), int64(1), object(6)
memory usage: 230.3+ KB
```

```
[36]: # drop unnecessary columns
cases.
    ↪drop(['SFDC_ID', 'DUNS_NUMBER', 'MAX_MODIFIED_DATE', 'CX_CASE_L1_REASON', 'CX_CASE_L2_REASON'],
          axis = 1, inplace = True)
cases.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2948 entries, 0 to 3026
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype  
---  --  
 0   NO_OF_CASES       2948 non-null   int64  
 1   CASE_ORIGIN      2948 non-null   object  
 2   CASE_REASON      2948 non-null   object  
 3   CUSTOMER_SITE_ID 2948 non-null   float64 
dtypes: float64(1), int64(1), object(2)
memory usage: 115.2+ KB
```

```
[37]: cases.head()
```

```
[37]:    NO_OF_CASES      CASE_ORIGIN      CASE_REASON CUSTOMER_SITE_ID
0            1  CC Survey Followup  CX: Customer Care      10,599.00
1            1  CC Survey Followup  CX: Customer Care      73,688.00
2            1  CC Survey Followup  CX: Customer Care       895.00
3            1  CC Survey Followup  CX: Customer Care     229,512.00
4            1  TS Survey Followup  CX: Customer Care     362,868.00
```

2.3.2 Aggregating

```
[38]: # aggregate CASE_ORIGIN
cases01 = cases.groupby(['CUSTOMER_SITE_ID', 'CASE_ORIGIN'])['NO_OF_CASES'].
    ↪sum().reset_index()
cases02 = cases01.groupby('CUSTOMER_SITE_ID')['NO_OF_CASES'].max().reset_index()
cases03 = cases02.merge(cases01, how = 'inner', ↪
    ↪on=['CUSTOMER_SITE_ID', 'NO_OF_CASES'])
cases03 = cases03.groupby('CUSTOMER_SITE_ID')['CASE_ORIGIN'].apply(lambda x: x.
    ↪mode()[0]).reset_index()
cases03.head(10)
```

```
[38]:    CUSTOMER_SITE_ID      CASE_ORIGIN
0            20.00  Email - VTI NACC
1            36.00  Email - VTI NACC
2            86.00  Email - VTI NACC
3           114.00  Email - VTI NACC
4           126.00          Phone
5           136.00          Phone
6           169.00          Phone
7          367.00  TS Survey Followup
8           403.00          Phone
9           429.00  Email - VTI NACC
```

```
[39]: len(cases03.CUSTOMER_SITE_ID.value_counts()) / len(cases03)
```

```
[39]: 1.0
```

```
[40]: # aggregate CASE_REASON
cases11 = cases.groupby(['CUSTOMER_SITE_ID', 'CASE_REASON'])['NO_OF_CASES'].
    ↪sum().reset_index()
cases12 = cases11.groupby('CUSTOMER_SITE_ID')['NO_OF_CASES'].max().reset_index()
cases13 = cases12.merge(cases11, how = 'inner', ↪
    ↪on=['CUSTOMER_SITE_ID', 'NO_OF_CASES'])
cases13 = cases13.groupby('CUSTOMER_SITE_ID')['CASE_REASON'].apply(lambda x: x.
    ↪mode()[0]).reset_index()
cases13.head(10)
```

```
[40]:    CUSTOMER_SITE_ID      CASE_REASON
0            20.00  Customer Experience
1            36.00  Customer Experience
2            86.00  Customer Experience
3           114.00  Customer Experience
4           126.00  Customer Experience
5           136.00  Customer Experience
6           169.00  Customer Experience
7          367.00  CX: Tech Support
8          403.00  Customer Experience
9          429.00  Customer Experience
```

```
[41]: len(cases13.CUSTOMER_SITE_ID.value_counts()) / len(cases13)
```

```
[41]: 1.0
```

```
[42]: # aggregate NO_OF_CASES
cases1 = cases[['CUSTOMER_SITE_ID', 'NO_OF_CASES']]
cases2 = cases1.groupby('CUSTOMER_SITE_ID')['NO_OF_CASES'].sum().reset_index()
cases = cases2
cases2.head()
```

```
[42]:    CUSTOMER_SITE_ID  NO_OF_CASES
0            20.00          83
1            36.00          2
2            86.00          1
3           114.00          1
4           126.00          1
```

```
[43]: # rename columns
cases = cases.rename(columns = {"NO_OF_CASES": "Total_Cases"})
cases.head()
```

```
[43]:    CUSTOMER_SITE_ID  Total_Cases
0            20.00          83
1            36.00          2
2            86.00          1
3           114.00          1
4           126.00          1
```

```
[44]: cases = pd.merge(cases, cases03, on='CUSTOMER_SITE_ID').merge(cases13, on='CUSTOMER_SITE_ID')
```

```
[45]: cases = cases.rename(columns = {'CASE_ORIGIN': 'Max_Case-Origin', 'CASE_REASON': 'Max_Case_Reason'})
```

```
[46]: cases.head()
```

```
[46]:   CUSTOMER_SITE_ID  Total_Cases  Max_Case_Origin      Max_Case_Reason
0            20.00          83 Email - VTI NACC Customer Experience
1            36.00           2 Email - VTI NACC Customer Experience
2            86.00           1 Email - VTI NACC Customer Experience
3           114.00           1 Email - VTI NACC Customer Experience
4           126.00           1       Phone Customer Experience
```

```
[47]: # # check all customer sites are unique
len(cases['CUSTOMER_SITE_ID'])/len(cases['CUSTOMER_SITE_ID'].unique())
```

```
[47]: 1.0
```

2.4 Install Base

```
[48]: installbase.head()
```

```
[48]:   INSTANCE_ID          STATUS INSTALL_DATE CONTRACT_FLAG \
0      225594 Inactive - Unconfirmed 5/7/2004          N
1      214815 Inactive - Unconfirmed 4/16/2004         N
2      159150 Inactive - Unconfirmed 4/23/2004         N
3      199246 Inactive - Unconfirmed 8/16/2004         N
4      199247 Inactive - Unconfirmed 7/23/2004         N

      CONTRACT_START_DATE CONTRACT_END_DATE DUNS_NUMBER CUSTOMER_SITE_ID \
0                  NaN             NaN 137361531        10970
1                  NaN             NaN 49061187        10983
2                  NaN             NaN 137361531        10970
3                  NaN             NaN 85497652        10967
4                  NaN             NaN 85497652        10967

      SITE_NUMBER          SFDC_ID PRODUCT_FAMILY INSTANCE_LAST_UPDATE_DATE \
0    13140518 0018000000drGe2AAE      GRAPHICS        2/17/2004
1    100026717 0018000000drEebAAE      GRAPHICS        3/15/2004
2    13140518 0018000000drGe2AAE      GRAPHICS        4/23/2004
3    13140513 0018000000drGe0AAE      GRAPHICS        3/24/2004
4    13140513 0018000000drGe0AAE      GRAPHICS        3/24/2004

      CONTRACT_CATEGORY
0                  NaN
1                  NaN
2                  NaN
3                  NaN
4                  NaN
```

```
[49]: installbase.shape
```

```
[49]: (120450, 13)
```

2.4.1 Preprocessing before aggregating

```
[50]: # check duplicates
installbase.drop_duplicates(inplace = True)
installbase.shape
```

```
[50]: (120149, 13)
```

```
[51]: # check column names and null values
installbase.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 120149 entries, 0 to 120449
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   INSTANCE_ID      120149 non-null   int64  
 1   STATUS            119629 non-null   object  
 2   INSTALL_DATE      120032 non-null   object  
 3   CONTRACT_FLAG     120149 non-null   object  
 4   CONTRACT_START_DATE 38425 non-null   object  
 5   CONTRACT_END_DATE 38425 non-null   object  
 6   DUNS_NUMBER       120149 non-null   int64  
 7   CUSTOMER_SITE_ID 120149 non-null   int64  
 8   SITE_NUMBER        120149 non-null   object  
 9   SFDC_ID            120148 non-null   object  
 10  PRODUCT_FAMILY    120149 non-null   object  
 11  INSTANCE_LAST_UPDATE_DATE 120149 non-null   object  
 12  CONTRACT_CATEGORY 38425 non-null   object  
dtypes: int64(3), object(10)
memory usage: 12.8+ MB
```

```
[52]: # copy INSTALL_DATE for later steps
installbase['INSTALL_DATE1'] = installbase['INSTALL_DATE']
```

```
[53]: # convert date variables to datetime data type
installbase[["INSTALL_DATE", "INSTALL_DATE1", "CONTRACT_START_DATE", ↴
    "CONTRACT_END_DATE", "INSTANCE_LAST_UPDATE_DATE"]] = ↴
    installbase[["INSTALL_DATE", "INSTALL_DATE1", "CONTRACT_START_DATE", ↴
    "CONTRACT_END_DATE", "INSTANCE_LAST_UPDATE_DATE"]].apply(pd.to_datetime)
```

```
[54]: # check data types
installbase.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 120149 entries, 0 to 120449
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   INSTANCE_ID      120149 non-null   int64  
 1   STATUS            119629 non-null   object  
 2   INSTALL_DATE      120032 non-null   object  
 3   CONTRACT_FLAG     120149 non-null   object  
 4   CONTRACT_START_DATE 38425 non-null   object  
 5   CONTRACT_END_DATE 38425 non-null   object  
 6   DUNS_NUMBER       120149 non-null   int64  
 7   CUSTOMER_SITE_ID 120149 non-null   int64  
 8   SITE_NUMBER        120149 non-null   object  
 9   SFDC_ID            120148 non-null   object  
 10  PRODUCT_FAMILY    120149 non-null   object  
 11  INSTANCE_LAST_UPDATE_DATE 120149 non-null   object  
 12  CONTRACT_CATEGORY 38425 non-null   object  
 13  INSTALL_DATE1     120149 non-null   datetime64[ns]
```

```

0   INSTANCE_ID           120149 non-null  int64
1   STATUS                 119629 non-null  object
2   INSTALL_DATE           120032 non-null  datetime64[ns]
3   CONTRACT_FLAG          120149 non-null  object
4   CONTRACT_START_DATE    38425 non-null   datetime64[ns]
5   CONTRACT_END_DATE      38425 non-null   datetime64[ns]
6   DUNS_NUMBER            120149 non-null  int64
7   CUSTOMER_SITE_ID       120149 non-null  int64
8   SITE_NUMBER             120149 non-null  object
9   SFDC_ID                120148 non-null  object
10  PRODUCT_FAMILY          120149 non-null  object
11  INSTANCE_LAST_UPDATE_DATE 120149 non-null  datetime64[ns]
12  CONTRACT_CATEGORY      38425 non-null   object
13  INSTALL_DATE1          120032 non-null  datetime64[ns]
dtypes: datetime64[ns](5), int64(3), object(6)
memory usage: 13.7+ MB

```

```
[55]: # create a new column called Contract_length
installbase['Contract_length'] = (installbase['CONTRACT_END_DATE'] - 
                                   installbase['CONTRACT_START_DATE']).dt.days

# fill missing value in this column using 0
installbase['Contract_length'] = installbase['Contract_length'].fillna(0)
```

```
[56]: # drop unnecessary columns
installbase.
       →drop(['INSTANCE_ID','SFDC_ID','SITE_NUMBER','DUNS_NUMBER','PRODUCT_FAMILY','CONTRACT_START_DATE'],
              axis = 1, inplace = True)
installbase.head()
```

```
[56]: STATUS INSTALL_DATE CONTRACT_FLAG CUSTOMER_SITE_ID \
0  Inactive - Unconfirmed  2004-05-07          N        10970
1  Inactive - Unconfirmed  2004-04-16          N        10983
2  Inactive - Unconfirmed  2004-04-23          N        10970
3  Inactive - Unconfirmed  2004-08-16          N        10967
4  Inactive - Unconfirmed  2004-07-23          N        10967

INSTANTCE_LAST_UPDATE_DATE CONTRACT_CATEGORY INSTALL_DATE1 Contract_length
0                      2004-02-17          NaN  2004-05-07        0.00
1                      2004-03-15          NaN  2004-04-16        0.00
2                      2004-04-23          NaN  2004-04-23        0.00
3                      2004-03-24          NaN  2004-08-16        0.00
4                      2004-03-24          NaN  2004-07-23        0.00
```

```
[57]: # check column names and null values
installbase.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 120149 entries, 0 to 120449
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   STATUS            119629 non-null   object  
 1   INSTALL_DATE      120032 non-null   datetime64[ns]
 2   CONTRACT_FLAG     120149 non-null   object  
 3   CUSTOMER_SITE_ID  120149 non-null   int64   
 4   INSTANCE_LAST_UPDATE_DATE  120149 non-null   datetime64[ns]
 5   CONTRACT_CATEGORY 38425 non-null    object  
 6   INSTALL_DATE1     120032 non-null   datetime64[ns]
 7   Contract_length   120149 non-null   float64 
dtypes: datetime64[ns](3), float64(1), int64(1), object(3)
memory usage: 8.2+ MB

```

[58]: `installbase['CONTRACT_FLAG'].value_counts()`

```

N      81724
Y      38425
Name: CONTRACT_FLAG, dtype: int64

```

[59]: `# substitute Y/N with 1/0 in CONTRACT_FLAG`
`installbase['CONTRACT_FLAG'] = installbase['CONTRACT_FLAG'].apply(lambda x: 1 if x=='Y' else 0)`

[60]: `installbase.head()`

	STATUS	INSTALL_DATE	CONTRACT_FLAG	CUSTOMER_SITE_ID	\
0	Inactive - Unconfirmed	2004-05-07	0	10970	
1	Inactive - Unconfirmed	2004-04-16	0	10983	
2	Inactive - Unconfirmed	2004-04-23	0	10970	
3	Inactive - Unconfirmed	2004-08-16	0	10967	
4	Inactive - Unconfirmed	2004-07-23	0	10967	

	INSTANCE_LAST_UPDATE_DATE	CONTRACT_CATEGORY	INSTALL_DATE1	Contract_length
0	2004-02-17	NaN	2004-05-07	0.00
1	2004-03-15	NaN	2004-04-16	0.00
2	2004-04-23	NaN	2004-04-23	0.00
3	2004-03-24	NaN	2004-08-16	0.00
4	2004-03-24	NaN	2004-07-23	0.00

[61]: `# check column names and null values`
`installbase.info()`

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 120149 entries, 0 to 120449
Data columns (total 8 columns):

```

```

#   Column           Non-Null Count   Dtype  
---  --  
0   STATUS          119629 non-null    object  
1   INSTALL_DATE    120032 non-null    datetime64[ns] 
2   CONTRACT_FLAG   120149 non-null    int64  
3   CUSTOMER_SITE_ID 120149 non-null    int64  
4   INSTANCE_LAST_UPDATE_DATE 120149 non-null    datetime64[ns] 
5   CONTRACT_CATEGORY 38425 non-null    object  
6   INSTALL_DATE1   120032 non-null    datetime64[ns] 
7   Contract_length 120149 non-null    float64 
dtypes: datetime64[ns](3), float64(1), int64(2), object(2) 
memory usage: 8.2+ MB

```

[62]: # drop rows with null values in STATUS, INSTALL_DATE
installbase = installbase.dropna(axis = 0, subset = ['STATUS'])
installbase = installbase.dropna(axis = 0, subset = ['INSTALL_DATE'])

[63]: # check column names and null values
installbase.info()

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 119512 entries, 0 to 120449
Data columns (total 8 columns):
#   Column           Non-Null Count   Dtype  
---  --  
0   STATUS          119512 non-null    object  
1   INSTALL_DATE    119512 non-null    datetime64[ns] 
2   CONTRACT_FLAG   119512 non-null    int64  
3   CUSTOMER_SITE_ID 119512 non-null    int64  
4   INSTANCE_LAST_UPDATE_DATE 119512 non-null    datetime64[ns] 
5   CONTRACT_CATEGORY 38348 non-null    object  
6   INSTALL_DATE1   119512 non-null    datetime64[ns] 
7   Contract_length 119512 non-null    float64 
dtypes: datetime64[ns](3), float64(1), int64(2), object(2) 
memory usage: 8.2+ MB

```

[64]: installbase.CONTRACT_CATEGORY.value_counts()

```

[64]: FSMA            33496
      Full Care       4789
      Supportive      55
      WFC              8
      Name: CONTRACT_CATEGORY, dtype: int64

```

[65]: # fill in null values in CONTRACT_CATEGORY
installbase['CONTRACT_CATEGORY'] = installbase['CONTRACT_CATEGORY'].fillna('No_U
→Contract')

```
[66]: # check column names and null values
installbase.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 119512 entries, 0 to 120449
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   STATUS            119512 non-null   object  
 1   INSTALL_DATE      119512 non-null   datetime64[ns]
 2   CONTRACT_FLAG     119512 non-null   int64   
 3   CUSTOMER_SITE_ID  119512 non-null   int64   
 4   INSTANCE_LAST_UPDATE_DATE  119512 non-null   datetime64[ns]
 5   CONTRACT_CATEGORY 119512 non-null   object  
 6   INSTALL_DATE1     119512 non-null   datetime64[ns]
 7   Contract_length   119512 non-null   float64 
dtypes: datetime64[ns](3), float64(1), int64(2), object(2)
memory usage: 8.2+ MB
```

```
[67]: installbase.head()
```

```
STATUS    INSTALL_DATE  CONTRACT_FLAG  CUSTOMER_SITE_ID \
0  Inactive - Unconfirmed  2004-05-07          0        10970
1  Inactive - Unconfirmed  2004-04-16          0        10983
2  Inactive - Unconfirmed  2004-04-23          0        10970
3  Inactive - Unconfirmed  2004-08-16          0        10967
4  Inactive - Unconfirmed  2004-07-23          0        10967

INSTANCE_LAST_UPDATE_DATE  CONTRACT_CATEGORY  INSTALL_DATE1  Contract_length
0                      2004-02-17       No Contract    2004-05-07        0.00
1                      2004-03-15       No Contract    2004-04-16        0.00
2                      2004-04-23       No Contract    2004-04-23        0.00
3                      2004-03-24       No Contract    2004-08-16        0.00
4                      2004-03-24       No Contract    2004-07-23        0.00
```

2.4.2 Aggregating

```
[68]: # copy columns to be aggregated
installbase2 =_
    →installbase[['INSTALL_DATE', 'INSTALL_DATE1', 'CONTRACT_FLAG', 'CUSTOMER_SITE_ID',
                  'INSTANCE_LAST_UPDATE_DATE', 'Contract_length',_ 
    →'CONTRACT_CATEGORY']]
```

```
[69]: installbase2.head()
```

```
INSTALL_DATE  INSTALL_DATE1  CONTRACT_FLAG  CUSTOMER_SITE_ID \
0    2004-05-07    2004-05-07          0        10970
```

```

1 2004-04-16 2004-04-16 0 10983
2 2004-04-23 2004-04-23 0 10970
3 2004-08-16 2004-08-16 0 10967
4 2004-07-23 2004-07-23 0 10967

INSTANCE_LAST_UPDATE_DATE Contract_length CONTRACT_CATEGORY
0 2004-02-17 0.00 No Contract
1 2004-03-15 0.00 No Contract
2 2004-04-23 0.00 No Contract
3 2004-03-24 0.00 No Contract
4 2004-03-24 0.00 No Contract

```

```
[70]: # count number of install bases
installbase3 = installbase[['CUSTOMER_SITE_ID']]
installbase3 = installbase3.groupby('CUSTOMER_SITE_ID').size().reset_index()
installbase3 = installbase3.rename(columns = {0: 'Num_of_Install_Bases'})
installbase3.head()
```

```
[70]: CUSTOMER_SITE_ID Num_of_Install_Bases
0 24 5
1 36 43
2 85 1
3 86 10
4 90 6
```

```
[71]: # aggregate CONTRACT_FLAG, Contract_length, INSTANCE_LAST_UPDATE_DATE, ↴
      ↴INSTALL_DATE

installbase2['Contract_length'] = installbase2['Contract_length'].replace(0, np.nan)
installbase2 = installbase2.groupby('CUSTOMER_SITE_ID').agg({'CONTRACT_FLAG': 'sum',
                                                               'Contract_length': 'mean',
                                                               'Total_Contracts': 'mean'}).reset_index()
# rename columns
installbase2 = installbase2.rename(columns = {"CONTRACT_FLAG": "Total_Contracts"})

installbase2['Contract_length'] = installbase2['Contract_length'].replace(np.nan, 0)
installbase2.head()
```

```
C:\Users\jayan\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

This is separate from the ipykernel package so we can avoid doing imports until

```
[71]:   CUSTOMER_SITE_ID  Total_Contracts  Contract_length
0              24                  0            0.00
1              36                 11           625.00
2              85                  0            0.00
3              86                  4           1,095.00
4              90                  6           1,003.00
```

```
[72]: # check all customer sites are unique
len(installbase2['CUSTOMER_SITE_ID'])/len(installbase2['CUSTOMER_SITE_ID']).
    ↪unique()
```

```
[72]: 1.0
```

```
[73]: # merge the above 2 tables
installbase4 = pd.merge(installbase2, installbase3, how = 'inner')
installbase4.head()
```

```
[73]:   CUSTOMER_SITE_ID  Total_Contracts  Contract_length  Num_of_Install_Bases
0              24                  0            0.00                  5
1              36                 11           625.00                43
2              85                  0            0.00                  1
3              86                  4           1,095.00                10
4              90                  6           1,003.00                  6
```

```
[74]: # aggregate STATUS
installbase['Num_of_Active_Install_Bases'] = installbase['STATUS'].
    ↪map({'Inactive - Unconfirmed': 0,
          'Inactive - Confirmed': 1,
          'Active - Unconfirmed': 0,
          'Active - Confirmed': 1})
installbase5 = installbase.
    ↪groupby('CUSTOMER_SITE_ID')['Num_of_Active_Install_Bases'].sum().
    ↪reset_index()
installbase5.head()
```

```
[74]:   CUSTOMER_SITE_ID  Num_of_Active_Install_Bases
0              24                  5
1              36                 39
2              85                  0
```

```
3          86          6
4          90          6
```

```
[75]: # aggregate CONTRACT_CATEGORY
installbase6 = installbase.groupby('CUSTOMER_SITE_ID')[['CONTRACT_CATEGORY']].
    ↪apply(lambda x: x.mode()[0]).reset_index()
installbase6 = installbase6.rename(columns = {"CONTRACT_CATEGORY": ↪
    'Contract_Category'})
installbase6.head()
```

```
[75]:   CUSTOMER_SITE_ID Contract_Category
0            24      No Contract
1            36      No Contract
2            85      No Contract
3            86      No Contract
4            90        FSMA
```

```
[76]: # merge the above tables
installbase = pd.merge(installbase5, installbase4, how = 'inner').
    ↪merge(installbase6, how = 'inner')
installbase.head()
```

```
[76]:   CUSTOMER_SITE_ID  Num_of_Active_Install_Bases  Total_Contracts \
0                  24                      5                  0
1                  36                     39                 11
2                  85                      0                  0
3                  86                      6                  4
4                  90                      6                  6

      Contract_length  Num_of_Install_Bases Contract_Category
0             0.00                  5      No Contract
1           625.00                 43      No Contract
2             0.00                  1      No Contract
3         1,095.00                 10      No Contract
4         1,003.00                  6        FSMA
```

```
[77]: installbase.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20059 entries, 0 to 20058
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   CUSTOMER_SITE_ID    20059 non-null   int64  
 1   Num_of_Active_Install_Bases 20059 non-null   int64  
 2   Total_Contracts     20059 non-null   int64  
 3   Contract_length     20059 non-null   float64
```

```
4    Num_of_Install_Bases      20059 non-null  int64
5    Contract_Category        20059 non-null  object
dtypes: float64(1), int64(4), object(1)
memory usage: 1.1+ MB
```

```
[78]: # calculate a column called Number_of_Inactive_Sites
installbase['Num_of_Inactive_Install_Bases'] =_
    ~installbase['Num_of_Install_Bases'] -_
    ~installbase['Num_of_Active_Install_Bases']
```

```
[79]: installbase.head()
```

```
[79]:   CUSTOMER_SITE_ID  Num_of_Active_Install_Bases  Total_Contracts \
0                  24                      5                  0
1                  36                     39                 11
2                  85                      0                  0
3                  86                      6                  4
4                  90                      6                  6

  Contract_length  Num_of_Install_Bases Contract_Category \
0          0.00                  5      No Contract
1       625.00                 43      No Contract
2          0.00                  1      No Contract
3      1,095.00                 10      No Contract
4      1,003.00                  6        FSMA

  Num_of_Inactive_Install_Bases
0                          0
1                          4
2                          1
3                          4
4                          0
```

```
[80]: # check all customer sites are unique
len(installbase['CUSTOMER_SITE_ID'])/len(installbase['CUSTOMER_SITE_ID']).
    ~unique()
```

```
[80]: 1.0
```

3 Exporting result tables

```
[81]: service.to_csv("service_aggregated.csv")
interactions.to_csv("interactions_aggregated.csv")
cases.to_csv("cases_aggregated.csv")
installbase.to_csv("installbase_aggregated.csv")
```

Part 2 - Join All Tables

February 12, 2021

```
[1]: # import relevant libraries
import pandas as pd
import scipy.stats
import numpy as np
import datetime as dt
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('float_format', '{:.2f}'.format)
```

1 Importing datasets

```
[2]: # import datasets
cases = pd.read_csv('cases_aggregated.csv', index_col = 0)
interactions = pd.read_csv('interactions_aggregated.csv', index_col = 0)
installbase = pd.read_csv('installbase_aggregated.csv', index_col = 0)
service = pd.read_csv('service_aggregated.csv', index_col = 0)
combined = pd.read_csv('Combined_Site_Rev_Prod_Price.csv')
```

C:\Users\jayan\AppData\Local\Continuum\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3063: DtypeWarning: Columns (0,19) have mixed types.Specify dtype option on import or set low_memory=False.
interactivity=interactivity, compiler=compiler, result=result)

2 Checking the table combined in SQL

```
[3]: combined.head()
```

```
[3]:   Site_ID  ITEM_ID  yYEAR  SLS_USD  Qty  Site_Level_Price_Index_WS \
0  102036258    268183      16  1,246.32  6.00                      1,001.87
1  102005330    598740      16    644.37  3.00                      588.99
2  102036258    268183      16  1,246.32  6.00                      1,001.87
3  102005330    598740      16    644.37  3.00                      588.99
```

4	102036258	268183	16	1,246.32	6.00		1,001.87
0			Site_Level_Price_Index_STU	Site_Level_Price_Index	CUSTOMER_ID	\	
1			1,246.32	0.80	117841		
2			644.37	0.91	113032		
3			1,246.32	0.80	117841		
4			644.37	0.91	113032		
			1,246.32	0.80	117841		
0	CUSTOMER_SITE_ID	SITE_CREATION_DATE			SIC_CODE_DESC	\	
1	609636	8/15/2013	2033-Canned fruits and specialties				
2	578406	3/25/2013	3721-Aircraft				
3	609636	8/15/2013	2033-Canned fruits and specialties				
4	578406	3/25/2013	3721-Aircraft				
	609636	8/15/2013	2033-Canned fruits and specialties				
0	SHORT_VERTICAL		CITY STATE COUNTRY POSTAL_CODE SELLING_ORG	\			
1	FRUIT & VEGETABLE	SALEM	OR US 97301	USA			
2	AERO/AUTO	CRESTVIEW	FL US 32539	USA			
3	FRUIT & VEGETABLE	SALEM	OR US 97301	USA			
4	AERO/AUTO	CRESTVIEW	FL US 32539	USA			
	FRUIT & VEGETABLE	SALEM	OR US 97301	USA			
0	CUSTOMER_CLASS	SHIP_TO_SITE_NUMBER	CORPORATE_CODE LAST_MODIFIED_DATE	\			
1	END USER	102036258	NaN 9/30/2019				
2	END USER	102005330	NaN 10/1/2019				
3	END USER	102036258	NaN 9/30/2019				
4	END USER	102005330	NaN 10/1/2019				
	END USER	102036258	NaN 9/30/2019				
0	TERRITORY_REGION	TERRITORY_TYPE	SUPPLIES_SEGMENTATION	\			
1	NW	Industrial	S				
2	SE	Industrial	S				
3	NW	Industrial	S				
4	SE	Industrial	S				
	NW	Industrial	S				
0	SUPPLIES_DECLINE_REASON	IB_CONFIRMED_DATE	IB_STATUS	\			
1	Over Stocked / Timing	4/10/2017	Active				
2	NaN	5/8/2020	Active				
3	Over Stocked / Timing	4/10/2017	Active				
4	NaN	5/8/2020	Active				
	Over Stocked / Timing	4/10/2017	Active				
0	IB_STATUS_INACTIVE_REASON	SFDC_ID	DUNS_NUMBER	CUSTOMER_TRX_ID	\		
1	NaN 001C00000187X5zIAE	78842640		8731929			
	NaN 001C00000158vAsIAI	43202248		9305874			

```

2           NaN 001C00000187X5zIAE    78842640      9093354
3           NaN 001C00000158vAsIAI    43202248      9195970
4           NaN 001C00000187X5zIAE    78842640      9079815

  DW_INVOICE_ID ITEM_ID.1          SFDC_ID.1     TRX_DATE   TRX_AMT_USD \
0      108745302    268183 001C00000187X5zIAE  2/5/2016      207.72
1      276000052    598740 001C00000158vAsIAI 12/16/2016      214.79
2      223068431    268183 001C00000187X5zIAE  8/26/2016      207.72
3      247802392    598740 001C00000158vAsIAI 10/19/2016      429.58
4      219738849    268183 001C00000187X5zIAE  8/19/2016      623.16

  TRX_COST_USD INVOICE_CURRENCY_CODE SALES_CHANNEL INVOICE_TYPE QUANTITY \
0      19.13             USD        Online      INVOICE      1.00
1      15.85             USD        Esker      INVOICE      1.00
2      19.13             USD        Online      INVOICE      1.00
3      31.69             USD        Esker      INVOICE      2.00
4      57.39             USD        Online      INVOICE      3.00

  SHIP_TO_SITE_ID SHIP_TO_CUSTOMER_ID BILL_TO_CUSTOMER_ID INVOICE_NUM \
0      609636            117841          117841      3206875
1      578406            113032          97674       3294491
2      609636            117841          117841      3263477
3      578406            113032          97674       3278903
4      609636            117841          117841      3261624

  ORDER_TYPE      ORDER_NUM DUNS_NUMBER.1 ITEM_ID.2 PRODUCT_GROUP \
0 STANDARD 511,759,211.00    78842640    268183    SUPPLIES
1 STANDARD 511,868,043.00    43202248    598740    SUPPLIES
2 STANDARD 511,823,154.00    78842640    268183    SUPPLIES
3 STANDARD 511,849,315.00    43202248    598740    SUPPLIES
4 STANDARD 511,823,154.00    78842640    268183    SUPPLIES

```

	PRODUCT_FAMILY	PRODUCT_MODEL	UOM	PFV_FAMILY	PFV_MODEL_GROUP
0	CIJ	MAKE-UP	Each	CIJ	CIJ - LEGACY
1	CIJ	VALUE PACK	Each	CIJ	CIJ - LEGACY
2	CIJ	MAKE-UP	Each	CIJ	CIJ - LEGACY
3	CIJ	VALUE PACK	Each	CIJ	CIJ - LEGACY
4	CIJ	MAKE-UP	Each	CIJ	CIJ - LEGACY

[4]: combined.shape

[4]: (628207, 56)

[5]: # check duplicates
combined.drop_duplicates(inplace = True)
combined.shape

[5]: (628077, 56)

[6]: # check column names
combined.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 628077 entries, 0 to 628206
Data columns (total 56 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Site_ID          628077 non-null   object 
 1   ITEM_ID          628077 non-null   int64  
 2   yYEAR            628077 non-null   int64  
 3   SLS_USD          628077 non-null   float64
 4   Qty              628077 non-null   float64
 5   Site_Level_Price_Index_WS 628077 non-null   float64
 6   Site_Level_Price_Index_STU 628077 non-null   float64
 7   Site_Level_Price_Index 628077 non-null   float64
 8   CUSTOMER_ID      628077 non-null   int64  
 9   CUSTOMER_SITE_ID 628077 non-null   int64  
 10  SITE_CREATION_DATE 628077 non-null   object 
 11  SIC_CODE_DESC    628077 non-null   object 
 12  SHORT_VERTICAL   628077 non-null   object 
 13  CITY              628043 non-null   object 
 14  STATE             568591 non-null   object 
 15  COUNTRY           628077 non-null   object 
 16  POSTAL_CODE       624605 non-null   object 
 17  SELLING_ORG      628077 non-null   object 
 18  CUSTOMER_CLASS    628077 non-null   object 
 19  SHIP_TO_SITE_NUMBER 628077 non-null   object 
 20  CORPORATE_CODE    143199 non-null   object 
 21  LAST_MODIFIED_DATE 628077 non-null   object 
 22  TERRITORY_REGION 626838 non-null   object 
 23  TERRITORY_TYPE    626838 non-null   object 
 24  SUPPLIES_SEGMENTATION 628077 non-null   object 
 25  SUPPLIES_DECLINE_REASON 281187 non-null   object 
 26  IB_CONFIRMED_DATE 585229 non-null   object 
 27  IB_STATUS          616981 non-null   object 
 28  IB_STATUS_INACTIVE_REASON 24595 non-null   object 
 29  SFDC_ID            628069 non-null   object 
 30  DUNS_NUMBER         628077 non-null   int64  
 31  CUSTOMER_TRX_ID    628077 non-null   int64  
 32  DW_INVOICE_ID       628077 non-null   int64  
 33  ITEM_ID.1          628077 non-null   int64  
 34  SFDC_ID.1          628069 non-null   object 
 35  TRX_DATE           628077 non-null   object 
 36  TRX_AMT_USD        628076 non-null   float64
```

```

37 TRX_COST_USD           628076 non-null float64
38 INVOICE_CURRENCY_CODE  628077 non-null object
39 SALES_CHANNEL          628073 non-null object
40 INVOICE_TYPE            628077 non-null object
41 QUANTITY               628077 non-null float64
42 SHIP_TO_SITE_ID         628077 non-null int64
43 SHIP_TO_CUSTOMER_ID    628077 non-null int64
44 BILL_TO_CUSTOMER_ID    628077 non-null int64
45 INVOICE_NUM             628077 non-null int64
46 ORDER_TYPE              628074 non-null object
47 ORDER_NUM               628074 non-null float64
48 DUNS_NUMBER.1           628077 non-null int64
49 ITEM_ID.2               628077 non-null int64
50 PRODUCT_GROUP           628077 non-null object
51 PRODUCT_FAMILY           628077 non-null object
52 PRODUCT_MODEL            628077 non-null object
53 UOM                     628077 non-null object
54 PFV_FAMILY               628077 non-null object
55 PFV_MODEL_GROUP          628077 non-null object
dtypes: float64(9), int64(14), object(33)
memory usage: 273.1+ MB

```

```
[7]: # drop unnecessary columns
combined.drop(['Site_ID', 'yYEAR', 'ITEM_ID.1', 'SFDC_ID.1', 'DUNS_NUMBER.1', ↴
    'ITEM_ID.2'], axis = 1, inplace = True)
```

```
[8]: # check column names
combined.info()
```

#	Column	Non-Null Count	Dtype
0	ITEM_ID	628077	int64
1	SLS_USD	628077	float64
2	Qty	628077	float64
3	Site_Level_Price_Index_WS	628077	float64
4	Site_Level_Price_Index_STU	628077	float64
5	Site_Level_Price_Index	628077	float64
6	CUSTOMER_ID	628077	int64
7	CUSTOMER_SITE_ID	628077	int64
8	SITE_CREATION_DATE	628077	object
9	SIC_CODE_DESC	628077	object
10	SHORT_VERTICAL	628077	object
11	CITY	628043	object
12	STATE	568591	object
13	COUNTRY	628077	object

```

14 POSTAL_CODE           624605 non-null object
15 SELLING_ORG          628077 non-null object
16 CUSTOMER_CLASS        628077 non-null object
17 SHIP_TO_SITE_NUMBER   628077 non-null object
18 CORPORATE_CODE        143199 non-null object
19 LAST_MODIFIED_DATE    628077 non-null object
20 TERRITORY_REGION     626838 non-null object
21 TERRITORY_TYPE        626838 non-null object
22 SUPPLIES_SEGMENTATION 628077 non-null object
23 SUPPLIES_DECLINE_REASONS 281187 non-null object
24 IB_CONFIRMED_DATE    585229 non-null object
25 IB_STATUS             616981 non-null object
26 IB_STATUS_INACTIVE_REASON 24595 non-null object
27 SFDC_ID               628069 non-null object
28 DUNS_NUMBER           628077 non-null int64
29 CUSTOMER_TRX_ID       628077 non-null int64
30 DW_INVOICE_ID          628077 non-null int64
31 TRX_DATE              628077 non-null object
32 TRX_AMT_USD           628076 non-null float64
33 TRX_COST_USD          628076 non-null float64
34 INVOICE_CURRENCY_CODE  628077 non-null object
35 SALES_CHANNEL          628073 non-null object
36 INVOICE_TYPE            628077 non-null object
37 QUANTITY               628077 non-null float64
38 SHIP_TO_SITE_ID        628077 non-null int64
39 SHIP_TO_CUSTOMER_ID    628077 non-null int64
40 BILL_TO_CUSTOMER_ID    628077 non-null int64
41 INVOICE_NUM             628077 non-null int64
42 ORDER_TYPE              628074 non-null object
43 ORDER_NUM               628074 non-null float64
44 PRODUCT_GROUP           628077 non-null object
45 PRODUCT_FAMILY           628077 non-null object
46 PRODUCT_MODEL            628077 non-null object
47 UOM                     628077 non-null object
48 PFV_FAMILY               628077 non-null object
49 PFV_MODEL_GROUP          628077 non-null object
dtypes: float64(9), int64(10), object(31)
memory usage: 244.4+ MB

```

3 Join all tables using left join

```
[9]: # merge all tables
df = pd.merge(combined, service, how = 'left',on = 'CUSTOMER_SITE_ID')
df = pd.merge(df, interactions, how = 'left',on = 'CUSTOMER_SITE_ID')
df = pd.merge(df, cases, how = 'left',on = 'CUSTOMER_SITE_ID')
df = pd.merge(df, installbase, how = 'left',on = 'CUSTOMER_SITE_ID')
```

[10]: df.head()

```
[10]:    ITEM_ID  SLS_USD  Qty  Site_Level_Price_Index_WS  \
0      268183  1,246.32  6.00                  1,001.87
1      598740    644.37  3.00                  588.99
2      268183  1,246.32  6.00                  1,001.87
3      598740    644.37  3.00                  588.99
4      268183  1,246.32  6.00                  1,001.87

      Site_Level_Price_Index_STU  Site_Level_Price_Index  CUSTOMER_ID  \
0                  1,246.32                  0.80      117841
1                  644.37                  0.91      113032
2                  1,246.32                  0.80      117841
3                  644.37                  0.91      113032
4                  1,246.32                  0.80      117841

  CUSTOMER_SITE_ID SITE_CREATION_DATE          SIC_CODE_DESC  \
0      609636        8/15/2013  2033-Canned fruits and specialties
1      578406        3/25/2013            3721-Aircraft
2      609636        8/15/2013  2033-Canned fruits and specialties
3      578406        3/25/2013            3721-Aircraft
4      609636        8/15/2013  2033-Canned fruits and specialties

      SHORT_VERTICAL          CITY STATE COUNTRY POSTAL_CODE SELLING_ORG  \
0  FRUIT & VEGETABLE      SALEM   OR     US      97301      USA
1      AERO/AUTO  CRESTVIEW   FL     US      32539      USA
2  FRUIT & VEGETABLE      SALEM   OR     US      97301      USA
3      AERO/AUTO  CRESTVIEW   FL     US      32539      USA
4  FRUIT & VEGETABLE      SALEM   OR     US      97301      USA

  CUSTOMER_CLASS SHIP_TO_SITE_NUMBER CORPORATE_CODE LAST_MODIFIED_DATE  \
0      END USER           102036258           NaN      9/30/2019
1      END USER           102005330           NaN      10/1/2019
2      END USER           102036258           NaN      9/30/2019
3      END USER           102005330           NaN      10/1/2019
4      END USER           102036258           NaN      9/30/2019

  TERRITORY_REGION TERRITORY_TYPE SUPPLIES_SEGMENTATION  \
0             NW      Industrial           S
1             SE      Industrial           S
2             NW      Industrial           S
3             SE      Industrial           S
4             NW      Industrial           S

  SUPPLIES_DECLINE_REASON IB_CONFIRMED_DATE IB_STATUS  \
0  Over Stocked / Timing        4/10/2017    Active
1                   NaN        5/8/2020    Active
```

2	Over Stocked / Timing		4/10/2017	Active
3		NaN	5/8/2020	Active
4	Over Stocked / Timing		4/10/2017	Active

	IB_STATUS_INACTIVE_REASON	SFDC_ID	DUNS_NUMBER	CUSTOMER_TRX_ID	\
0	NaN	001C00000187X5zIAE	78842640	8731929	
1	NaN	001C00000158vAsIAI	43202248	9305874	
2	NaN	001C00000187X5zIAE	78842640	9093354	
3	NaN	001C00000158vAsIAI	43202248	9195970	
4	NaN	001C00000187X5zIAE	78842640	9079815	

	DW_INVOICE_ID	TRX_DATE	TRX_AMT_USD	TRX_COST_USD	INVOICE_CURRENCY_CODE	\
0	108745302	2/5/2016	207.72	19.13	USD	
1	276000052	12/16/2016	214.79	15.85	USD	
2	223068431	8/26/2016	207.72	19.13	USD	
3	247802392	10/19/2016	429.58	31.69	USD	
4	219738849	8/19/2016	623.16	57.39	USD	

	SALES_CHANNEL	INVOICE_TYPE	QUANTITY	SHIP_TO_SITE_ID	SHIP_TO_CUSTOMER_ID	\
0	Online	INVOICE	1.00	609636	117841	
1	Esker	INVOICE	1.00	578406	113032	
2	Online	INVOICE	1.00	609636	117841	
3	Esker	INVOICE	2.00	578406	113032	
4	Online	INVOICE	3.00	609636	117841	

	BILL_TO_CUSTOMER_ID	INVOICE_NUM	ORDER_TYPE	ORDER_NUM	\
0	117841	3206875	STANDARD DOMESTIC	511,759,211.00	
1	97674	3294491	STANDARD DOMESTIC	511,868,043.00	
2	117841	3263477	STANDARD DOMESTIC	511,823,154.00	
3	97674	3278903	STANDARD DOMESTIC	511,849,315.00	
4	117841	3261624	STANDARD DOMESTIC	511,823,154.00	

	PRODUCT_GROUP	PRODUCT_FAMILY	PRODUCT_MODEL	UOM	PFV_FAMILY	PFV_MODEL_GROUP	\
0	SUPPLIES	CIJ	MAKE-UP	Each	CIJ	CIJ - LEGACY	
1	SUPPLIES	CIJ	VALUE PACK	Each	CIJ	CIJ - LEGACY	
2	SUPPLIES	CIJ	MAKE-UP	Each	CIJ	CIJ - LEGACY	
3	SUPPLIES	CIJ	VALUE PACK	Each	CIJ	CIJ - LEGACY	
4	SUPPLIES	CIJ	MAKE-UP	Each	CIJ	CIJ - LEGACY	

	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\
0	11.00	3.00	8.00	
1	21.00	5.00	16.00	
2	11.00	3.00	8.00	
3	21.00	5.00	16.00	
4	11.00	3.00	8.00	

	Most_Frequent_Interaction_Type	Total_Visits	Total_Tasks	Total_Cases	\
--	--------------------------------	--------------	-------------	-------------	---

```

0             Call      18.00      17.00      nan
1             Email     58.00      62.00      1.00
2             Call      18.00      17.00      nan
3             Email     58.00      62.00      1.00
4             Call      18.00      17.00      nan

  Max_Case_Origin      Max_Case_Reason  Num_of_Active_Install_Bases \
0           NaN            NaN                  6.00
1 Email - VTI NACC Customer Experience          4.00
2           NaN            NaN                  6.00
3 Email - VTI NACC Customer Experience          4.00
4           NaN            NaN                  6.00

  Total_Contracts Contract_length  Num_of_Install_Bases Contract_Category \
0         0.00          0.00          6.00      No Contract
1         4.00        1,011.25        4.00          FSMA
2         0.00          0.00          6.00      No Contract
3         4.00        1,011.25        4.00          FSMA
4         0.00          0.00          6.00      No Contract

  Num_of_Inactive_Install_Bases
0                 0.00
1                 0.00
2                 0.00
3                 0.00
4                 0.00

```

[11]: df.shape

[11]: (628077, 65)

[12]: # check column names and null values
df.isnull().sum()

ITEM_ID	0
SLS_USD	0
Qty	0
Site_Level_Price_Index_WS	0
Site_Level_Price_Index_STU	0
Site_Level_Price_Index	0
CUSTOMER_ID	0
CUSTOMER_SITE_ID	0
SITE_CREATION_DATE	0
SIC_CODE_DESC	0
SHORT_VERTICAL	0
CITY	34
STATE	59486

COUNTRY	0
POSTAL_CODE	3472
SELLING_ORG	0
CUSTOMER_CLASS	0
SHIP_TO_SITE_NUMBER	0
CORPORATE_CODE	484878
LAST_MODIFIED_DATE	0
TERRITORY_REGION	1239
TERRITORY_TYPE	1239
SUPPLIES_SEGMENTATION	0
SUPPLIES_DECLINE_REASONS	346890
IB_CONFIRMED_DATE	42848
IB_STATUS	11096
IB_STATUS_INACTIVE_REASON	603482
SFDC_ID	8
DUNS_NUMBER	0
CUSTOMER_TRX_ID	0
DW_INVOICE_ID	0
TRX_DATE	0
TRX_AMT_USD	1
TRX_COST_USD	1
INVOICE_CURRENCY_CODE	0
SALES_CHANNEL	4
INVOICE_TYPE	0
QUANTITY	0
SHIP_TO_SITE_ID	0
SHIP_TO_CUSTOMER_ID	0
BILL_TO_CUSTOMER_ID	0
INVOICE_NUM	0
ORDER_TYPE	3
ORDER_NUM	3
PRODUCT_GROUP	0
PRODUCT_FAMILY	0
PRODUCT_MODEL	0
UOM	0
PFV_FAMILY	0
PFV_MODEL_GROUP	0
Total_SVC_Incidents	73870
Total_Repeat_Calls	73870
Total_FTF_Calls	73870
Most_Frequent_Interaction_Type	9656
Total_Visits	9656
Total_Tasks	9656
Total_Cases	413167
Max_Case-Origin	413167
Max_Case_Reason	413167
Num_of_Active_Install_Bases	58008

```

Total_Contracts          58008
Contract_length          58008
Num_of_Install_Bases     58008
Contract_Category        58008
Num_of_Inactive_Install_Bases 58008
dtype: int64

```

4 Exporting result table

[13]: df.info()

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 628077 entries, 0 to 628076
Data columns (total 65 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   ITEM_ID          628077 non-null   int64  
 1   SLS_USD          628077 non-null   float64 
 2   Qty               628077 non-null   float64 
 3   Site_Level_Price_Index_WS 628077 non-null   float64 
 4   Site_Level_Price_Index_STU 628077 non-null   float64 
 5   Site_Level_Price_Index 628077 non-null   float64 
 6   CUSTOMER_ID       628077 non-null   int64  
 7   CUSTOMER_SITE_ID 628077 non-null   int64  
 8   SITE_CREATION_DATE 628077 non-null   object  
 9   SIC_CODE_DESC     628077 non-null   object  
 10  SHORT_VERTICAL    628077 non-null   object  
 11  CITY              628043 non-null   object  
 12  STATE             568591 non-null   object  
 13  COUNTRY           628077 non-null   object  
 14  POSTAL_CODE       624605 non-null   object  
 15  SELLING_ORG      628077 non-null   object  
 16  CUSTOMER_CLASS    628077 non-null   object  
 17  SHIP_TO_SITE_NUMBER 628077 non-null   object  
 18  CORPORATE_CODE    143199 non-null   object  
 19  LAST_MODIFIED_DATE 628077 non-null   object  
 20  TERRITORY_REGION 626838 non-null   object  
 21  TERRITORY_TYPE    626838 non-null   object  
 22  SUPPLIES_SEGMENTATION 628077 non-null   object  
 23  SUPPLIES_DECLINE_REASON 281187 non-null   object  
 24  IB_CONFIRMED_DATE 585229 non-null   object  
 25  IB_STATUS          616981 non-null   object  
 26  IB_STATUS_INACTIVE_REASON 24595 non-null   object  
 27  SFDC_ID            628069 non-null   object  
 28  DUNS_NUMBER         628077 non-null   int64  
 29  CUSTOMER_TRX_ID    628077 non-null   int64

```

```

30 DW_INVOICE_ID           628077 non-null int64
31 TRX_DATE                628077 non-null object
32 TRX_AMT_USD              628076 non-null float64
33 TRX_COST_USD              628076 non-null float64
34 INVOICE_CURRENCY_CODE      628077 non-null object
35 SALES_CHANNEL             628073 non-null object
36 INVOICE_TYPE                628077 non-null object
37 QUANTITY                  628077 non-null float64
38 SHIP_TO_SITE_ID            628077 non-null int64
39 SHIP_TO_CUSTOMER_ID        628077 non-null int64
40 BILL_TO_CUSTOMER_ID        628077 non-null int64
41 INVOICE_NUM                  628077 non-null int64
42 ORDER_TYPE                  628074 non-null object
43 ORDER_NUM                  628074 non-null float64
44 PRODUCT_GROUP                628077 non-null object
45 PRODUCT_FAMILY               628077 non-null object
46 PRODUCT_MODEL                 628077 non-null object
47 UOM                         628077 non-null object
48 PFV_FAMILY                  628077 non-null object
49 PFV_MODEL_GROUP              628077 non-null object
50 Total_SVC_Incidents          554207 non-null float64
51 Total_Repeat_Calls            554207 non-null float64
52 Total_FTF_Calls                554207 non-null float64
53 Most_Frequent_Interaction_Type 618421 non-null object
54 Total_Visits                  618421 non-null float64
55 Total_Tasks                  618421 non-null float64
56 Total_Cases                  214910 non-null float64
57 Max_Case-Origin                214910 non-null object
58 Max_Case_Reason                214910 non-null object
59 Num_of_Active_Install_Bases     570069 non-null float64
60 Total_Contracts                570069 non-null float64
61 Contract_length                570069 non-null float64
62 Num_of_Install_Bases            570069 non-null float64
63 Contract_Category                570069 non-null object
64 Num_of_Inactive_Install_Bases    570069 non-null float64
dtypes: float64(20), int64(10), object(35)
memory usage: 316.3+ MB

```

```
[14]: df.to_csv("Final_Merged_File.csv")
```

Part 3 - Data Cleaning

February 12, 2021

```
[1]: # import relevant libraries
import pandas as pd
import scipy.stats
import numpy as np
import datetime as dt
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import datetime as dt
import warnings
warnings.filterwarnings("ignore")

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('float_format', '{:.2f}'.format)
```

1 Importing dataset

```
[2]: # import dataset
df = pd.read_csv('Final_Merged_File.csv', index_col = 0)
df.head()
```

```
[2]: ITEM_ID  SLS_USD  Qty  Site_Level_Price_Index_WS  \
0    268183  1,246.32  6.00          1,001.87
1    598740    644.37  3.00          588.99
2    268183  1,246.32  6.00          1,001.87
3    598740    644.37  3.00          588.99
4    268183  1,246.32  6.00          1,001.87

      Site_Level_Price_Index_STU  Site_Level_Price_Index  CUSTOMER_ID  \
0                  1,246.32            0.80        117841
1                  644.37            0.91        113032
2                  1,246.32            0.80        117841
3                  644.37            0.91        113032
4                  1,246.32            0.80        117841
```

	CUSTOMER_SITE_ID	SITE_CREATION_DATE		SIC_CODE_DESC	\
0	609636	8/15/2013	2033-Canned fruits and specialties		
1	578406	3/25/2013		3721-Aircraft	
2	609636	8/15/2013	2033-Canned fruits and specialties		
3	578406	3/25/2013		3721-Aircraft	
4	609636	8/15/2013	2033-Canned fruits and specialties		
	SHORT_VERTICAL	CITY	STATE	COUNTRY	POSTAL_CODE SELLING_ORG \
0	FRUIT & VEGETABLE	SALEM	OR	US	97301 USA
1	AERO/AUTO	CRESTVIEW	FL	US	32539 USA
2	FRUIT & VEGETABLE	SALEM	OR	US	97301 USA
3	AERO/AUTO	CRESTVIEW	FL	US	32539 USA
4	FRUIT & VEGETABLE	SALEM	OR	US	97301 USA
	CUSTOMER_CLASS	SHIP_TO_SITE_NUMBER	CORPORATE_CODE	LAST_MODIFIED_DATE	\
0	END USER	102036258	NaN	9/30/2019	
1	END USER	102005330	NaN	10/1/2019	
2	END USER	102036258	NaN	9/30/2019	
3	END USER	102005330	NaN	10/1/2019	
4	END USER	102036258	NaN	9/30/2019	
	TERRITORY_REGION	TERRITORY_TYPE	SUPPLIES_SEGMENTATION		\
0	NW	Industrial		S	
1	SE	Industrial		S	
2	NW	Industrial		S	
3	SE	Industrial		S	
4	NW	Industrial		S	
	SUPPLIES_DECLINE_REASON	IB_CONFIRMED_DATE	IB_STATUS		\
0	Over Stocked / Timing	4/10/2017	Active		
1		5/8/2020	Active		
2	Over Stocked / Timing	4/10/2017	Active		
3		5/8/2020	Active		
4	Over Stocked / Timing	4/10/2017	Active		
	IB_STATUS_INACTIVE_REASON	SFDC_ID	DUNS_NUMBER	CUSTOMER_TRX_ID	\
0	NaN	001C00000187X5zIAE	78842640	8731929	
1	NaN	001C00000158vAsIAI	43202248	9305874	
2	NaN	001C00000187X5zIAE	78842640	9093354	
3	NaN	001C00000158vAsIAI	43202248	9195970	
4	NaN	001C00000187X5zIAE	78842640	9079815	
	DW_INVOICE_ID	TRX_DATE	TRX_AMT_USD	TRX_COST_USD	INVOICE_CURRENCY_CODE \
0	108745302	2/5/2016	207.72	19.13	USD
1	276000052	12/16/2016	214.79	15.85	USD
2	223068431	8/26/2016	207.72	19.13	USD
3	247802392	10/19/2016	429.58	31.69	USD

4	219738849	8/19/2016	623.16	57.39	USD
0	SALES_CHANNEL	INVOICE_TYPE	QUANTITY	SHIP_TO_SITE_ID	SHIP_TO_CUSTOMER_ID \
0	Online	INVOICE	1.00	609636	117841
1	Esker	INVOICE	1.00	578406	113032
2	Online	INVOICE	1.00	609636	117841
3	Esker	INVOICE	2.00	578406	113032
4	Online	INVOICE	3.00	609636	117841
0	BILL_TO_CUSTOMER_ID	INVOICE_NUM	ORDER_TYPE	ORDER_NUM	\
0		117841	3206875	STANDARD DOMESTIC	511,759,211.00
1		97674	3294491	STANDARD DOMESTIC	511,868,043.00
2		117841	3263477	STANDARD DOMESTIC	511,823,154.00
3		97674	3278903	STANDARD DOMESTIC	511,849,315.00
4		117841	3261624	STANDARD DOMESTIC	511,823,154.00
0	PRODUCT_GROUP	PRODUCT_FAMILY	PRODUCT_MODEL	UOM	PFV_FAMILY PFV_MODEL_GROUP \
0	SUPPLIES	CIJ	MAKE-UP	Each	CIJ CIJ - LEGACY
1	SUPPLIES	CIJ	VALUE PACK	Each	CIJ CIJ - LEGACY
2	SUPPLIES	CIJ	MAKE-UP	Each	CIJ CIJ - LEGACY
3	SUPPLIES	CIJ	VALUE PACK	Each	CIJ CIJ - LEGACY
4	SUPPLIES	CIJ	MAKE-UP	Each	CIJ CIJ - LEGACY
0	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\	
0		11.00	3.00	8.00	
1		21.00	5.00	16.00	
2		11.00	3.00	8.00	
3		21.00	5.00	16.00	
4		11.00	3.00	8.00	
0	Most_Frequent_Interaction_Type	Total_Visits	Total_Tasks	Total_Cases	\
0		Call	18.00	17.00	nan
1		Email	58.00	62.00	1.00
2		Call	18.00	17.00	nan
3		Email	58.00	62.00	1.00
4		Call	18.00	17.00	nan
0	Max_Case_Origin	Max_Case_Reason	Num_of_Active_Install_Bases	\	
0		NaN	NaN	6.00	
1	Email - VTI NACC	Customer Experience		4.00	
2		NaN	NaN	6.00	
3	Email - VTI NACC	Customer Experience		4.00	
4		NaN	NaN	6.00	
0	Total_Contracts	Contract_length	Num_of_Install_Bases	Contract_Category	\
0		0.00	0.00	6.00	No Contract
1		4.00	1,011.25	4.00	FSMA

```

2          0.00          0.00      6.00      No Contract
3          4.00     1,011.25      4.00      FSMA
4          0.00          0.00      6.00      No Contract

Num_of_Inactive_Install_Bases
0              0.00
1              0.00
2              0.00
3              0.00
4              0.00

```

[3]: df.shape

[3]: (628077, 65)

[4]: # check duplicates
df.drop_duplicates(inplace = True)
df.shape

[4]: (628077, 65)

[5]: # make a copy and analyze on the copied data
data = df.copy()

2 Filtering

[6]: # keep positive transactions only
data = data[data['TRX_AMT_USD'] > 0]

keep only US
data = data[data['COUNTRY'] == "US"]
data.shape

[6]: (544668, 65)

3 Checking Each Variable

3.1 Converting date variables to datetime type

[8]: # set date columns to datetime
data['TRX_DATE'] = pd.to_datetime(data['TRX_DATE'])
data['SITE_CREATION_DATE'] = pd.to_datetime(data['SITE_CREATION_DATE'])
data['LAST_MODIFIED_DATE'] = pd.to_datetime(data['LAST_MODIFIED_DATE'])
data['IB_CONFIRMED_DATE'] = pd.to_datetime(data['IB_CONFIRMED_DATE'])

3.2 SFDC_ID: drop column

```
[10]: data.drop(["SFDC_ID"],axis = 1, inplace = True)
```

```
[11]: data.shape
```

```
[11]: (544668, 64)
```

```
[12]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 544668 entries, 0 to 628076
Data columns (total 64 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   ITEM_ID          544668 non-null   int64  
 1   SLS_USD          544668 non-null   float64 
 2   Qty              544668 non-null   float64 
 3   Site_Level_Price_Index_WS 544668 non-null   float64 
 4   Site_Level_Price_Index_STU 544668 non-null   float64 
 5   Site_Level_Price_Index    544668 non-null   float64 
 6   CUSTOMER_ID       544668 non-null   int64  
 7   CUSTOMER_SITE_ID  544668 non-null   int64  
 8   SITE_CREATION_DATE 544668 non-null   datetime64[ns]
 9   SIC_CODE_DESC     544668 non-null   object  
 10  SHORT_VERTICAL    544668 non-null   object  
 11  CITY              544668 non-null   object  
 12  STATE             544668 non-null   object  
 13  COUNTRY           544668 non-null   object  
 14  POSTAL_CODE       544665 non-null   object  
 15  SELLING_ORG      544668 non-null   object  
 16  CUSTOMER_CLASS    544668 non-null   object  
 17  SHIP_TO_SITE_NUMBER 544668 non-null   object  
 18  CORPORATE_CODE    126294 non-null   object  
 19  LAST_MODIFIED_DATE 544668 non-null   datetime64[ns]
 20  TERRITORY_REGION  543541 non-null   object  
 21  TERRITORY_TYPE    543541 non-null   object  
 22  SUPPLIES_SEGMENTATION 544668 non-null   object  
 23  SUPPLIES_DECLINE_REASONS 245552 non-null   object  
 24  IB_CONFIRMED_DATE 509264 non-null   datetime64[ns]
 25  IB_STATUS          535091 non-null   object  
 26  IB_STATUS_INACTIVE_REASON 21892 non-null   object  
 27  DUNS_NUMBER        544668 non-null   int64  
 28  CUSTOMER_TRX_ID    544668 non-null   int64  
 29  DW_INVOICE_ID      544668 non-null   int64  
 30  TRX_DATE           544668 non-null   datetime64[ns]
 31  TRX_AMT_USD       544668 non-null   float64 
 32  TRX_COST_USD      544668 non-null   float64 
```

```

33 INVOICE_CURRENCY_CODE           544668 non-null object
34 SALES_CHANNEL                 544665 non-null object
35 INVOICE_TYPE                  544668 non-null object
36 QUANTITY                      544668 non-null float64
37 SHIP_TO_SITE_ID               544668 non-null int64
38 SHIP_TO_CUSTOMER_ID           544668 non-null int64
39 BILL_TO_CUSTOMER_ID           544668 non-null int64
40 INVOICE_NUM                   544668 non-null int64
41 ORDER_TYPE                    544665 non-null object
42 ORDER_NUM                     544665 non-null float64
43 PRODUCT_GROUP                 544668 non-null object
44 PRODUCT_FAMILY                544668 non-null object
45 PRODUCT_MODEL                 544668 non-null object
46 UOM                           544668 non-null object
47 PFV_FAMILY                    544668 non-null object
48 PFV_MODEL_GROUP              544668 non-null object
49 Total_SVC_Incidents          483736 non-null float64
50 Total_Repeat_Calls           483736 non-null float64
51 Total_FTF_Calls              483736 non-null float64
52 Most_Frequent_Interaction_Type 536976 non-null object
53 Total_Visits                  536976 non-null float64
54 Total_Tasks                   536976 non-null float64
55 Total_Cases                   187829 non-null float64
56 Max_Case_Origin              187829 non-null object
57 Max_Case_Reason               187829 non-null object
58 Num_of_Active_Install_Bases   495174 non-null float64
59 Total_Contracts               495174 non-null float64
60 Contract_length               495174 non-null float64
61 Num_of_Install_Bases          495174 non-null float64
62 Contract_Category             495174 non-null object
63 Num_of_Inactive_Install_Bases 495174 non-null float64
dtypes: datetime64[ns](4), float64(20), int64(10), object(30)
memory usage: 270.1+ MB

```

3.3 INVOICE_CURRENCY_CODE: filter USD rows / drop column

```
[13]: data = data[data['INVOICE_CURRENCY_CODE'] == "USD"]
```

```
[14]: data.shape
```

```
[14]: (544606, 64)
```

```
[15]: data.drop(['INVOICE_CURRENCY_CODE'], axis = 1, inplace = True)
```

```
[16]: data.shape
```

```
[16]: (544606, 63)
```

3.4 SALES_CHANNEL: drop NA

```
[17]: # drop rows with null values in SALES_CHANNEL  
data = data.dropna(axis = 0, subset = ['SALES_CHANNEL'])
```

```
[18]: data.shape
```

```
[18]: (544603, 63)
```

3.5 QUANTITY: convert to nearest integer

```
[19]: data['QUANTITY'] = data['QUANTITY'].apply(np.ceil).astype(int)
```

```
[20]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 544603 entries, 0 to 628076  
Data columns (total 63 columns):  
 #   Column           Non-Null Count  Dtype     
---  --  
 0   ITEM_ID          544603 non-null  int64    
 1   SLS_USD          544603 non-null  float64  
 2   Qty              544603 non-null  float64  
 3   Site_Level_Price_Index_WS  544603 non-null  float64  
 4   Site_Level_Price_Index_STU 544603 non-null  float64  
 5   Site_Level_Price_Index  544603 non-null  float64  
 6   CUSTOMER_ID       544603 non-null  int64    
 7   CUSTOMER_SITE_ID  544603 non-null  int64    
 8   SITE_CREATION_DATE 544603 non-null  datetime64[ns]  
 9   SIC_CODE_DESC     544603 non-null  object    
 10  SHORT_VERTICAL    544603 non-null  object    
 11  CITY              544603 non-null  object    
 12  STATE             544603 non-null  object    
 13  COUNTRY           544603 non-null  object    
 14  POSTAL_CODE       544600 non-null  object    
 15  SELLING_ORG      544603 non-null  object    
 16  CUSTOMER_CLASS    544603 non-null  object    
 17  SHIP_TO_SITE_NUMBER 544603 non-null  object    
 18  CORPORATE_CODE    126290 non-null  object    
 19  LAST_MODIFIED_DATE 544603 non-null  datetime64[ns]  
 20  TERRITORY_REGION  543476 non-null  object    
 21  TERRITORY_TYPE    543476 non-null  object    
 22  SUPPLIES_SEGMENTATION 544603 non-null  object    
 23  SUPPLIES_DECLINE_REASONS 245546 non-null  object    
 24  IB_CONFIRMED_DATE 509246 non-null  datetime64[ns]  
 25  IB_STATUS          535040 non-null  object    
 26  IB_STATUS_INACTIVE_REASON 21886 non-null  object
```

```

27 DUNS_NUMBER           544603 non-null int64
28 CUSTOMER_TRX_ID      544603 non-null int64
29 DW_INVOICE_ID          544603 non-null int64
30 TRX_DATE              544603 non-null datetime64[ns]
31 TRX_AMT_USD           544603 non-null float64
32 TRX_COST_USD          544603 non-null float64
33 SALES_CHANNEL          544603 non-null object
34 INVOICE_TYPE            544603 non-null object
35 QUANTITY                544603 non-null int32
36 SHIP_TO_SITE_ID         544603 non-null int64
37 SHIP_TO_CUSTOMER_ID     544603 non-null int64
38 BILL_TO_CUSTOMER_ID      544603 non-null int64
39 INVOICE_NUM               544603 non-null int64
40 ORDER_TYPE                544603 non-null object
41 ORDER_NUM                  544603 non-null float64
42 PRODUCT_GROUP             544603 non-null object
43 PRODUCT_FAMILY              544603 non-null object
44 PRODUCT_MODEL                544603 non-null object
45 UOM                         544603 non-null object
46 PFV_FAMILY                 544603 non-null object
47 PFV_MODEL_GROUP             544603 non-null object
48 Total_SVC_Incidents        483733 non-null float64
49 Total_Repeat_Calls         483733 non-null float64
50 Total_FTF_Calls            483733 non-null float64
51 Most_Frequent_Interaction_Type 536936 non-null object
52 Total_Visits                  536936 non-null float64
53 Total_Tasks                  536936 non-null float64
54 Total_Cases                  187822 non-null float64
55 Max_Case-Origin              187822 non-null object
56 Max_Case_Reason                187822 non-null object
57 Num_of_Active_Install_Bases   495163 non-null float64
58 Total_Contracts                495163 non-null float64
59 Contract_length                  495163 non-null float64
60 Num_of_Install_Bases            495163 non-null float64
61 Contract_Category                495163 non-null object
62 Num_of_Inactive_Install_Bases   495163 non-null float64
dtypes: datetime64[ns](4), float64(19), int32(1), int64(10), object(29)
memory usage: 263.8+ MB

```

3.6 SHIP_TO_SITE_ID / SHIP_TO_SITE_NUMBER / SHIP_TO_CUSTOMER_ID: drop column

```
[21]: data.drop(["SHIP_TO_SITE_ID", "SHIP_TO_SITE_NUMBER", "SHIP_TO_CUSTOMER_ID",  
    ↴"BILL_TO_CUSTOMER_ID"], axis = 1, inplace = True)
```

```
[22]: data.shape
```

[22]: (544603, 59)

[23]: data.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 544603 entries, 0 to 628076
Data columns (total 59 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   ITEM_ID          544603 non-null   int64  
 1   SLS_USD          544603 non-null   float64 
 2   Qty              544603 non-null   float64 
 3   Site_Level_Price_Index_WS 544603 non-null   float64 
 4   Site_Level_Price_Index_STU 544603 non-null   float64 
 5   Site_Level_Price_Index 544603 non-null   float64 
 6   CUSTOMER_ID      544603 non-null   int64  
 7   CUSTOMER_SITE_ID 544603 non-null   int64  
 8   SITE_CREATION_DATE 544603 non-null   datetime64[ns]
 9   SIC_CODE_DESC    544603 non-null   object  
 10  SHORT_VERTICAL   544603 non-null   object  
 11  CITY              544603 non-null   object  
 12  STATE             544603 non-null   object  
 13  COUNTRY           544603 non-null   object  
 14  POSTAL_CODE       544600 non-null   object  
 15  SELLING_ORG      544603 non-null   object  
 16  CUSTOMER_CLASS    544603 non-null   object  
 17  CORPORATE_CODE    126290 non-null   object  
 18  LAST_MODIFIED_DATE 544603 non-null   datetime64[ns]
 19  TERRITORY_REGION 543476 non-null   object  
 20  TERRITORY_TYPE    543476 non-null   object  
 21  SUPPLIES_SEGMENTATION 544603 non-null   object  
 22  SUPPLIES_DECLINE_REASONS 245546 non-null   object  
 23  IB_CONFIRMED_DATE 509246 non-null   datetime64[ns]
 24  IB_STATUS          535040 non-null   object  
 25  IB_STATUS_INACTIVE_REASON 21886 non-null   object  
 26  DUNS_NUMBER         544603 non-null   int64  
 27  CUSTOMER_TRX_ID    544603 non-null   int64  
 28  DW_INVOICE_ID       544603 non-null   int64  
 29  TRX_DATE            544603 non-null   datetime64[ns]
 30  TRX_AMT_USD        544603 non-null   float64 
 31  TRX_COST_USD       544603 non-null   float64 
 32  SALES_CHANNEL      544603 non-null   object  
 33  INVOICE_TYPE         544603 non-null   object  
 34  QUANTITY            544603 non-null   int32  
 35  INVOICE_NUM          544603 non-null   int64  
 36  ORDER_TYPE           544603 non-null   object  
 37  ORDER_NUM            544603 non-null   float64
```

```

38 PRODUCT_GROUP           544603 non-null object
39 PRODUCT_FAMILY          544603 non-null object
40 PRODUCT_MODEL           544603 non-null object
41 UOM                     544603 non-null object
42 PFV_FAMILY              544603 non-null object
43 PFV_MODEL_GROUP         544603 non-null object
44 Total_SVC_Incidents    483733 non-null float64
45 Total_Repeat_Calls     483733 non-null float64
46 Total_FTF_Calls        483733 non-null float64
47 Most_Frequent_Interaction_Type 536936 non-null object
48 Total_Visits            536936 non-null float64
49 Total_Tasks              536936 non-null float64
50 Total_Cases              187822 non-null float64
51 Max_Case_Origin         187822 non-null object
52 Max_Case_Reason          187822 non-null object
53 Num_of_Active_Install_Bases 495163 non-null float64
54 Total_Contracts          495163 non-null float64
55 Contract_length          495163 non-null float64
56 Num_of_Install_Bases     495163 non-null float64
57 Contract_Category        495163 non-null object
58 Num_of_Inactive_Install_Bases 495163 non-null float64
dtypes: datetime64[ns](4), float64(19), int32(1), int64(7), object(28)
memory usage: 247.2+ MB

```

3.7 PRODUCT_GROUP: drop column

```
[24]: data.drop(["PRODUCT_GROUP"], axis = 1, inplace = True)
```

```
[25]: data.shape
```

```
[25]: (544603, 58)
```

```
[26]: data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 544603 entries, 0 to 628076
Data columns (total 58 columns):
 #   Column                 Non-Null Count  Dtype  
--- 
 0   ITEM_ID                544603 non-null   int64  
 1   SLS_USD                544603 non-null   float64 
 2   Qty                     544603 non-null   float64 
 3   Site_Level_Price_Index_WS 544603 non-null   float64 
 4   Site_Level_Price_Index_STU 544603 non-null   float64 
 5   Site_Level_Price_Index  544603 non-null   float64 
 6   CUSTOMER_ID             544603 non-null   int64  
 7   CUSTOMER_SITE_ID        544603 non-null   int64  
 8   SITE_CREATION_DATE      544603 non-null   datetime64[ns]

```

9	SIC_CODE_DESC	544603	non-null	object
10	SHORT_VERTICAL	544603	non-null	object
11	CITY	544603	non-null	object
12	STATE	544603	non-null	object
13	COUNTRY	544603	non-null	object
14	POSTAL_CODE	544600	non-null	object
15	SELLING_ORG	544603	non-null	object
16	CUSTOMER_CLASS	544603	non-null	object
17	CORPORATE_CODE	126290	non-null	object
18	LAST_MODIFIED_DATE	544603	non-null	datetime64[ns]
19	TERRITORY_REGION	543476	non-null	object
20	TERRITORY_TYPE	543476	non-null	object
21	SUPPLIES_SEGMENTATION	544603	non-null	object
22	SUPPLIES_DECLINE_REASONS	245546	non-null	object
23	IB_CONFIRMED_DATE	509246	non-null	datetime64[ns]
24	IB_STATUS	535040	non-null	object
25	IB_STATUS_INACTIVE_REASON	21886	non-null	object
26	DUNS_NUMBER	544603	non-null	int64
27	CUSTOMER_TRX_ID	544603	non-null	int64
28	DW_INVOICE_ID	544603	non-null	int64
29	TRX_DATE	544603	non-null	datetime64[ns]
30	TRX_AMT_USD	544603	non-null	float64
31	TRX_COST_USD	544603	non-null	float64
32	SALES_CHANNEL	544603	non-null	object
33	INVOICE_TYPE	544603	non-null	object
34	QUANTITY	544603	non-null	int32
35	INVOICE_NUM	544603	non-null	int64
36	ORDER_TYPE	544603	non-null	object
37	ORDER_NUM	544603	non-null	float64
38	PRODUCT_FAMILY	544603	non-null	object
39	PRODUCT_MODEL	544603	non-null	object
40	UOM	544603	non-null	object
41	PFV_FAMILY	544603	non-null	object
42	PFV_MODEL_GROUP	544603	non-null	object
43	Total_SVC_Incidents	483733	non-null	float64
44	Total_Repeat_Calls	483733	non-null	float64
45	Total_FTF_Calls	483733	non-null	float64
46	Most_Frequent_Interaction_Type	536936	non-null	object
47	Total_Visits	536936	non-null	float64
48	Total_Tasks	536936	non-null	float64
49	Total_Cases	187822	non-null	float64
50	Max_Case_Origin	187822	non-null	object
51	Max_Case_Reason	187822	non-null	object
52	Num_of_Active_Install_Bases	495163	non-null	float64
53	Total_Contracts	495163	non-null	float64
54	Contract_length	495163	non-null	float64
55	Num_of_Install_Bases	495163	non-null	float64
56	Contract_Category	495163	non-null	object

```
57  Num_of_Inactive_Install_Bases    495163 non-null  float64
dtypes: datetime64[ns](4), float64(19), int32(1), int64(7), object(27)
memory usage: 243.1+ MB
```

3.8 SIC_CODE_DESC: drop column

```
[27]: data.drop(["SIC_CODE_DESC"],axis = 1, inplace = True)
```

```
[28]: data.shape
```

```
[28]: (544603, 57)
```

```
[29]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 544603 entries, 0 to 628076
Data columns (total 57 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   ITEM_ID          544603 non-null   int64  
 1   SLS_USD          544603 non-null   float64 
 2   Qty               544603 non-null   float64 
 3   Site_Level_Price_Index_WS  544603 non-null   float64 
 4   Site_Level_Price_Index_STU 544603 non-null   float64 
 5   Site_Level_Price_Index  544603 non-null   float64 
 6   CUSTOMER_ID       544603 non-null   int64  
 7   CUSTOMER_SITE_ID  544603 non-null   int64  
 8   SITE_CREATION_DATE 544603 non-null   datetime64[ns]
 9   SHORT_VERTICAL    544603 non-null   object  
 10  CITY              544603 non-null   object  
 11  STATE             544603 non-null   object  
 12  COUNTRY           544603 non-null   object  
 13  POSTAL_CODE       544600 non-null   object  
 14  SELLING_ORG      544603 non-null   object  
 15  CUSTOMER_CLASS    544603 non-null   object  
 16  CORPORATE_CODE    126290 non-null   object  
 17  LAST_MODIFIED_DATE 544603 non-null   datetime64[ns]
 18  TERRITORY_REGION  543476 non-null   object  
 19  TERRITORY_TYPE    543476 non-null   object  
 20  SUPPLIES_SEGMENTATION 544603 non-null   object  
 21  SUPPLIES_DECLINE_REASON 245546 non-null   object  
 22  IB_CONFIRMED_DATE 509246 non-null   datetime64[ns]
 23  IB_STATUS          535040 non-null   object  
 24  IB_STATUS_INACTIVE_REASON 21886 non-null   object  
 25  DUNS_NUMBER        544603 non-null   int64  
 26  CUSTOMER_TRX_ID    544603 non-null   int64  
 27  DW_INVOICE_ID      544603 non-null   int64  
 28  TRX_DATE           544603 non-null   datetime64[ns]
```

```

29 TRX_AMT_USD           544603 non-null float64
30 TRX_COST_USD          544603 non-null float64
31 SALES_CHANNEL          544603 non-null object
32 INVOICE_TYPE            544603 non-null object
33 QUANTITY                 544603 non-null int32
34 INVOICE_NUM              544603 non-null int64
35 ORDER_TYPE                544603 non-null object
36 ORDER_NUM                  544603 non-null float64
37 PRODUCT_FAMILY             544603 non-null object
38 PRODUCT_MODEL              544603 non-null object
39 UOM                         544603 non-null object
40 PFV_FAMILY                  544603 non-null object
41 PFV_MODEL_GROUP             544603 non-null object
42 Total_SVC_Incidents        483733 non-null float64
43 Total_Repeat_Calls         483733 non-null float64
44 Total_FTF_Calls            483733 non-null float64
45 Most_Frequent_Interaction_Type 536936 non-null object
46 Total_Visits                536936 non-null float64
47 Total_Tasks                  536936 non-null float64
48 Total_Cases                  187822 non-null float64
49 Max_Case-Origin             187822 non-null object
50 Max_Case_Reason              187822 non-null object
51 Num_of_Active_Install_Bases 495163 non-null float64
52 Total_Contracts             495163 non-null float64
53 Contract_length              495163 non-null float64
54 Num_of_Install_Bases         495163 non-null float64
55 Contract_Category             495163 non-null object
56 Num_of_Inactive_Install_Bases 495163 non-null float64
dtypes: datetime64[ns](4), float64(19), int32(1), int64(7), object(26)
memory usage: 238.9+ MB

```

3.9 COUNTRY: drop column

```
[30]: data.drop(["COUNTRY"], axis = 1, inplace = True)
```

```
[31]: data.shape
```

```
[31]: (544603, 56)
```

```
[32]: data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 544603 entries, 0 to 628076
Data columns (total 56 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   ITEM_ID          544603 non-null  int64  
 1   SLS_USD          544603 non-null  float64

```

2	Qty	544603	non-null	float64
3	Site_Level_Price_Index_WS	544603	non-null	float64
4	Site_Level_Price_Index_STU	544603	non-null	float64
5	Site_Level_Price_Index	544603	non-null	float64
6	CUSTOMER_ID	544603	non-null	int64
7	CUSTOMER_SITE_ID	544603	non-null	int64
8	SITE_CREATION_DATE	544603	non-null	datetime64[ns]
9	SHORT_VERTICAL	544603	non-null	object
10	CITY	544603	non-null	object
11	STATE	544603	non-null	object
12	POSTAL_CODE	544600	non-null	object
13	SELLING_ORG	544603	non-null	object
14	CUSTOMER_CLASS	544603	non-null	object
15	CORPORATE_CODE	126290	non-null	object
16	LAST_MODIFIED_DATE	544603	non-null	datetime64[ns]
17	TERRITORY_REGION	543476	non-null	object
18	TERRITORY_TYPE	543476	non-null	object
19	SUPPLIES_SEGMENTATION	544603	non-null	object
20	SUPPLIES_DECLINE_REASONS	245546	non-null	object
21	IB_CONFIRMED_DATE	509246	non-null	datetime64[ns]
22	IB_STATUS	535040	non-null	object
23	IB_STATUS_INACTIVE_REASON	21886	non-null	object
24	DUNS_NUMBER	544603	non-null	int64
25	CUSTOMER_TRX_ID	544603	non-null	int64
26	DW_INVOICE_ID	544603	non-null	int64
27	TRX_DATE	544603	non-null	datetime64[ns]
28	TRX_AMT_USD	544603	non-null	float64
29	TRX_COST_USD	544603	non-null	float64
30	SALES_CHANNEL	544603	non-null	object
31	INVOICE_TYPE	544603	non-null	object
32	QUANTITY	544603	non-null	int32
33	INVOICE_NUM	544603	non-null	int64
34	ORDER_TYPE	544603	non-null	object
35	ORDER_NUM	544603	non-null	float64
36	PRODUCT_FAMILY	544603	non-null	object
37	PRODUCT_MODEL	544603	non-null	object
38	UOM	544603	non-null	object
39	PFV_FAMILY	544603	non-null	object
40	PFV_MODEL_GROUP	544603	non-null	object
41	Total_SVC_Incidents	483733	non-null	float64
42	Total_Repeat_Calls	483733	non-null	float64
43	Total_FTF_Calls	483733	non-null	float64
44	Most_Frequent_Interaction_Type	536936	non-null	object
45	Total_Visits	536936	non-null	float64
46	Total_Tasks	536936	non-null	float64
47	Total_Cases	187822	non-null	float64
48	Max_Case-Origin	187822	non-null	object
49	Max_Case_Reason	187822	non-null	object

```

50  Num_of_Active_Install_Bases      495163 non-null float64
51  Total_Contracts                 495163 non-null float64
52  Contract_length                 495163 non-null float64
53  Num_of_Install_Bases            495163 non-null float64
54  Contract_Category               495163 non-null object
55  Num_of_Inactive_Install_Bases   495163 non-null float64
dtypes: datetime64[ns](4), float64(19), int32(1), int64(7), object(25)
memory usage: 234.8+ MB

```

3.10 CITY: convert to upper

```
[33]: data['CITY'] = data['CITY'].str.upper()
```

3.11 POSTAL_CODE: remove NA and “SHIP TO” rows and retain first 5 digits

```
[34]: data = data[data['POSTAL_CODE'] != "SHIP TO"]
```

```
[35]: data = data.dropna(axis=0, subset=['POSTAL_CODE'])
```

```
[36]: data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 544594 entries, 0 to 628076
Data columns (total 56 columns):
 #   Column           Non-Null Count Dtype  
 --- 
 0   ITEM_ID          544594 non-null int64   
 1   SLS_USD          544594 non-null float64 
 2   Qty              544594 non-null float64 
 3   Site_Level_Price_Index_WS 544594 non-null float64 
 4   Site_Level_Price_Index_STU 544594 non-null float64 
 5   Site_Level_Price_Index 544594 non-null float64 
 6   CUSTOMER_ID       544594 non-null int64   
 7   CUSTOMER_SITE_ID 544594 non-null int64   
 8   SITE_CREATION_DATE 544594 non-null datetime64[ns]
 9   SHORT_VERTICAL    544594 non-null object  
 10  CITY              544594 non-null object  
 11  STATE             544594 non-null object  
 12  POSTAL_CODE        544594 non-null object  
 13  SELLING_ORG        544594 non-null object  
 14  CUSTOMER_CLASS      544594 non-null object  
 15  CORPORATE_CODE       126290 non-null object  
 16  LAST_MODIFIED_DATE  544594 non-null datetime64[ns]
 17  TERRITORY_REGION    543467 non-null object  
 18  TERRITORY_TYPE       543467 non-null object  
 19  SUPPLIES_SEGMENTATION 544594 non-null object 

```

```

20 SUPPLIES_DECLINE_REASONS      245537 non-null  object
21 IB_CONFIRMED_DATE            509237 non-null  datetime64[ns]
22 IB_STATUS                     535031 non-null  object
23 IB_STATUS_INACTIVE_REASON    21877 non-null   object
24 DUNS_NUMBER                   544594 non-null  int64
25 CUSTOMER_TRX_ID              544594 non-null  int64
26 DW_INVOICE_ID                 544594 non-null  int64
27 TRX_DATE                      544594 non-null  datetime64[ns]
28 TRX_AMT_USD                  544594 non-null  float64
29 TRX_COST_USD                 544594 non-null  float64
30 SALES_CHANNEL                 544594 non-null  object
31 INVOICE_TYPE                  544594 non-null  object
32 QUANTITY                      544594 non-null  int32
33 INVOICE_NUM                   544594 non-null  int64
34 ORDER_TYPE                     544594 non-null  object
35 ORDER_NUM                      544594 non-null  float64
36 PRODUCT_FAMILY                 544594 non-null  object
37 PRODUCT_MODEL                  544594 non-null  object
38 UOM                            544594 non-null  object
39 PFV_FAMILY                     544594 non-null  object
40 PFV_MODEL_GROUP                544594 non-null  object
41 Total_SVC_Incidents           483733 non-null  float64
42 Total_Repeat_Calls             483733 non-null  float64
43 Total_FTF_Calls                483733 non-null  float64
44 Most_Frequent_Interaction_Type 536927 non-null  object
45 Total_Visits                   536927 non-null  float64
46 Total_Tasks                     536927 non-null  float64
47 Total_Cases                     187822 non-null  float64
48 Max_Case-Origin                187822 non-null  object
49 Max_Case_Reason                 187822 non-null  object
50 Num_of_Active_Install_Bases     495163 non-null  float64
51 Total_Contracts                 495163 non-null  float64
52 Contract_length                  495163 non-null  float64
53 Num_of_Install_Bases             495163 non-null  float64
54 Contract_Category                495163 non-null  object
55 Num_of_Inactive_Install_Bases    495163 non-null  float64
dtypes: datetime64[ns](4), float64(19), int32(1), int64(7), object(25)
memory usage: 234.8+ MB

```

```
[37]: data['POSTAL_CODE'] = data['POSTAL_CODE'].str[:5]
```

3.12 SELLING_ORG: drop column

```
[38]: data.drop(['SELLING_ORG'],axis = 1, inplace = True)
```

```
[39]: data.shape
```

[39]: (544594, 55)

3.13 CORPORATE_CODE: convert to STRATEGIC_ACCOUNTS

```
[40]: # identifying strategic accounts
prefix = ['PA', 'SA', 'TA', 'IA', 'NA']
data['STRATEGIC_ACCOUNTS'] = data.CORPORATE_CODE.str.startswith(tuple(prefix), u
    ↳na = False).astype(int)
data.drop(columns = 'CORPORATE_CODE', inplace = True)
```

```
[41]: data.head()
```

```
[41]:   ITEM_ID  SLS_USD  Qty  Site_Level_Price_Index_WS \
0    268183  1,246.32  6.00                  1,001.87
1    598740    644.37  3.00                  588.99
2    268183  1,246.32  6.00                  1,001.87
3    598740    644.37  3.00                  588.99
4    268183  1,246.32  6.00                  1,001.87

      Site_Level_Price_Index_STU  Site_Level_Price_Index  CUSTOMER_ID \
0                1,246.32            0.80        117841
1                  644.37            0.91        113032
2                1,246.32            0.80        117841
3                  644.37            0.91        113032
4                1,246.32            0.80        117841

  CUSTOMER_SITE_ID SITE_CREATION_DATE      SHORT_VERTICAL      CITY STATE \
0       609636     2013-08-15  FRUIT & VEGETABLE  SALEM   OR
1       578406     2013-03-25      AERO/AUTO  CRESTVIEW  FL
2       609636     2013-08-15  FRUIT & VEGETABLE  SALEM   OR
3       578406     2013-03-25      AERO/AUTO  CRESTVIEW  FL
4       609636     2013-08-15  FRUIT & VEGETABLE  SALEM   OR

  POSTAL_CODE CUSTOMER_CLASS LAST_MODIFIED_DATE TERRITORY_REGION \
0       97301    END USER        2019-09-30          NW
1       32539    END USER        2019-10-01          SE
2       97301    END USER        2019-09-30          NW
3       32539    END USER        2019-10-01          SE
4       97301    END USER        2019-09-30          NW

  TERRITORY_TYPE SUPPLIES_SEGMENTATION SUPPLIES_DECLINE_REASON \
0  Industrial                      S  Over Stocked / Timing
1  Industrial                      S                 NaN
2  Industrial                      S  Over Stocked / Timing
3  Industrial                      S                 NaN
4  Industrial                      S  Over Stocked / Timing
```

	IB_CONFIRMED_DATE	IB_STATUS	IB_STATUS_INACTIVE_REASON	DUNS_NUMBER	\	
0	2017-04-10	Active		NaN	78842640	
1	2020-05-08	Active		NaN	43202248	
2	2017-04-10	Active		NaN	78842640	
3	2020-05-08	Active		NaN	43202248	
4	2017-04-10	Active		NaN	78842640	
	CUSTOMER_TRX_ID	DW_INVOICE_ID	TRX_DATE	TRX_AMT_USD	TRX_COST_USD	\
0	8731929	108745302	2016-02-05	207.72	19.13	
1	9305874	276000052	2016-12-16	214.79	15.85	
2	9093354	223068431	2016-08-26	207.72	19.13	
3	9195970	247802392	2016-10-19	429.58	31.69	
4	9079815	219738849	2016-08-19	623.16	57.39	
	SALES_CHANNEL	INVOICE_TYPE	QUANTITY	INVOICE_NUM	ORDER_TYPE	\
0	Online	INVOICE	1	3206875	STANDARD DOMESTIC	
1	Esker	INVOICE	1	3294491	STANDARD DOMESTIC	
2	Online	INVOICE	1	3263477	STANDARD DOMESTIC	
3	Esker	INVOICE	2	3278903	STANDARD DOMESTIC	
4	Online	INVOICE	3	3261624	STANDARD DOMESTIC	
	ORDER_NUM	PRODUCT_FAMILY	PRODUCT_MODEL	UOM	PFV_FAMILY	\
0	511,759,211.00	CIJ	MAKE-UP	Each	CIJ	
1	511,868,043.00	CIJ	VALUE PACK	Each	CIJ	
2	511,823,154.00	CIJ	MAKE-UP	Each	CIJ	
3	511,849,315.00	CIJ	VALUE PACK	Each	CIJ	
4	511,823,154.00	CIJ	MAKE-UP	Each	CIJ	
	PFV_MODEL_GROUP	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\	
0	CIJ - LEGACY	11.00	3.00	8.00		
1	CIJ - LEGACY	21.00	5.00	16.00		
2	CIJ - LEGACY	11.00	3.00	8.00		
3	CIJ - LEGACY	21.00	5.00	16.00		
4	CIJ - LEGACY	11.00	3.00	8.00		
	Most_Frequent_Interaction_Type	Total_Visits	Total_Tasks	Total_Cases	\	
0	Call	18.00	17.00	nan		
1	Email	58.00	62.00	1.00		
2	Call	18.00	17.00	nan		
3	Email	58.00	62.00	1.00		
4	Call	18.00	17.00	nan		
	Max_Case_Origin	Max_Case_Reason	Num_of_Active_Install_Bases	\		
0	NaN	NaN	6.00			
1	Email - VTI NACC	Customer Experience	4.00			
2	NaN	NaN	6.00			
3	Email - VTI NACC	Customer Experience	4.00			

4	NaN	NaN	6.00	
	Total_Contracts	Contract_length	Num_of_Install_Bases	Contract_Category \
0	0.00	0.00	6.00	No Contract
1	4.00	1,011.25	4.00	FSMA
2	0.00	0.00	6.00	No Contract
3	4.00	1,011.25	4.00	FSMA
4	0.00	0.00	6.00	No Contract
	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS		
0	0.00	0		
1	0.00	0		
2	0.00	0		
3	0.00	0		
4	0.00	0		

[42]: data.info()

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 544594 entries, 0 to 628076
Data columns (total 55 columns):
 #   Column           Non-Null Count  Dtype  
 ---  --  
 0   ITEM_ID          544594 non-null   int64  
 1   SLS_USD          544594 non-null   float64 
 2   Qty               544594 non-null   float64 
 3   Site_Level_Price_Index_WS  544594 non-null   float64 
 4   Site_Level_Price_Index_STU 544594 non-null   float64 
 5   Site_Level_Price_Index  544594 non-null   float64 
 6   CUSTOMER_ID       544594 non-null   int64  
 7   CUSTOMER_SITE_ID  544594 non-null   int64  
 8   SITE_CREATION_DATE 544594 non-null   datetime64[ns]
 9   SHORT_VERTICAL    544594 non-null   object  
 10  CITY              544594 non-null   object  
 11  STATE             544594 non-null   object  
 12  POSTAL_CODE       544594 non-null   object  
 13  CUSTOMER_CLASS    544594 non-null   object  
 14  LAST_MODIFIED_DATE 544594 non-null   datetime64[ns]
 15  TERRITORY_REGION  543467 non-null   object  
 16  TERRITORY_TYPE    543467 non-null   object  
 17  SUPPLIES_SEGMENTATION 544594 non-null   object  
 18  SUPPLIES_DECLINE_REASON 245537 non-null   object  
 19  IB_CONFIRMED_DATE 509237 non-null   datetime64[ns]
 20  IB_STATUS          535031 non-null   object  
 21  IB_STATUS_INACTIVE_REASON 21877 non-null   object  
 22  DUNS_NUMBER        544594 non-null   int64  
 23  CUSTOMER_TRX_ID   544594 non-null   int64  

```

```

24 DW_INVOICE_ID           544594 non-null int64
25 TRX_DATE                544594 non-null datetime64[ns]
26 TRX_AMT_USD              544594 non-null float64
27 TRX_COST_USD              544594 non-null float64
28 SALES_CHANNEL            544594 non-null object
29 INVOICE_TYPE               544594 non-null object
30 QUANTITY                  544594 non-null int32
31 INVOICE_NUM                 544594 non-null int64
32 ORDER_TYPE                  544594 non-null object
33 ORDER_NUM                  544594 non-null float64
34 PRODUCT_FAMILY             544594 non-null object
35 PRODUCT_MODEL               544594 non-null object
36 UOM                         544594 non-null object
37 PFV_FAMILY                  544594 non-null object
38 PFV_MODEL_GROUP             544594 non-null object
39 Total_SVC_Incidents        483733 non-null float64
40 Total_Repeat_Calls         483733 non-null float64
41 Total_FTF_Calls             483733 non-null float64
42 Most_Frequent_Interaction_Type 536927 non-null object
43 Total_Visits                 536927 non-null float64
44 Total_Tasks                  536927 non-null float64
45 Total_Cases                  187822 non-null float64
46 Max_Case_Origin             187822 non-null object
47 Max_Case_Reason              187822 non-null object
48 Num_of_Active_Install_Bases 495163 non-null float64
49 Total_Contracts              495163 non-null float64
50 Contract_length              495163 non-null float64
51 Num_of_Install_Bases         495163 non-null float64
52 Contract_Category            495163 non-null object
53 Num_of_Inactive_Install_Bases 495163 non-null float64
54 STRATEGIC_ACCOUNTS          544594 non-null int32
dtypes: datetime64[ns](4), float64(19), int32(2), int64(7), object(23)
memory usage: 228.5+ MB

```

3.14 TERRITORY_REGION: reassign with mode of each state

```
[43]: # Removing Graphics, OEM, Distributor, Postal, Canada, NAs
# keeping MC, NW, NE, MW, SE, SC
data['TERRITORY_REGION'].
    →replace(['Graphics','OEM','Distributor','Postal','Canada'], '-',
    →True)
data['TERRITORY_REGION'].fillna('-', inplace = True)
data['TERRITORY_REGION'] = data['TERRITORY_REGION'].replace('-', np.nan)
```

```
[44]: from scipy import stats
region = data.groupby(['STATE'])['TERRITORY_REGION'].agg(lambda x: stats.
    →mode(x)[0][0]).reset_index()
```

```
region['TERRITORY_REGION'][region['STATE'] == 'DC'] = 'NE'
```

```
[45]: data.drop(columns = 'TERRITORY_REGION', inplace = True)
```

```
[46]: # reassigning values  
data = pd.merge(data, region, how = 'left', on = 'STATE')
```

```
[47]: data.TERRITORY_REGION.value_counts()
```

```
[47]: NE    109893  
MC     95730  
NW     95154  
MW     90512  
SE     86455  
SC     66850  
Name: TERRITORY_REGION, dtype: int64
```

```
[48]: data.shape
```

```
[48]: (544594, 55)
```

3.15 TERRITORY_TYPE: replace NA with mode of each city or state

```
[49]: data['TERRITORY_TYPE'].value_counts()
```

```
[49]: Industrial    516671  
Graphics        25345  
Postal          1451  
Name: TERRITORY_TYPE, dtype: int64
```

```
[50]: # filling NAs using mode by CITY if mode exists  
data['TERRITORY_TYPE'] = data['TERRITORY_TYPE'].fillna(data.  
    ↪groupby('CITY')['TERRITORY_TYPE'].transform(lambda x: stats.mode(x)[0][0])).  
    ↪astype(str)
```

```
[51]: # filling other using mode by STATE  
data['TERRITORY_TYPE'] = data['TERRITORY_TYPE'].replace('0', np.NaN)  
data['TERRITORY_TYPE'] = data['TERRITORY_TYPE'].fillna(data.  
    ↪groupby('STATE')['TERRITORY_TYPE'].transform(lambda x: stats.mode(x)[0][0])).  
    ↪astype(str)
```

```
[52]: data['TERRITORY_TYPE'].value_counts()
```

```
[52]: Industrial    517778  
Graphics        25365  
Postal          1451  
Name: TERRITORY_TYPE, dtype: int64
```

```
[53]: data.shape
```

```
[53]: (544594, 55)
```

```
[54]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 544594 entries, 0 to 544593
Data columns (total 55 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   ITEM_ID          544594 non-null   int64  
 1   SLS_USD          544594 non-null   float64 
 2   Qty              544594 non-null   float64 
 3   Site_Level_Price_Index_WS  544594 non-null   float64 
 4   Site_Level_Price_Index_STU 544594 non-null   float64 
 5   Site_Level_Price_Index  544594 non-null   float64 
 6   CUSTOMER_ID       544594 non-null   int64  
 7   CUSTOMER_SITE_ID  544594 non-null   int64  
 8   SITE_CREATION_DATE 544594 non-null   datetime64[ns]
 9   SHORT_VERTICAL    544594 non-null   object  
 10  CITY              544594 non-null   object  
 11  STATE             544594 non-null   object  
 12  POSTAL_CODE       544594 non-null   object  
 13  CUSTOMER_CLASS    544594 non-null   object  
 14  LAST_MODIFIED_DATE 544594 non-null   datetime64[ns]
 15  TERRITORY_TYPE    544594 non-null   object  
 16  SUPPLIES_SEGMENTATION 544594 non-null   object  
 17  SUPPLIES_DECLINE_REASONS 245537 non-null   object  
 18  IB_CONFIRMED_DATE 509237 non-null   datetime64[ns]
 19  IB_STATUS          535031 non-null   object  
 20  IB_STATUS_INACTIVE_REASON 21877 non-null   object  
 21  DUNS_NUMBER        544594 non-null   int64  
 22  CUSTOMER_TRX_ID    544594 non-null   int64  
 23  DW_INVOICE_ID       544594 non-null   int64  
 24  TRX_DATE           544594 non-null   datetime64[ns]
 25  TRX_AMT_USD        544594 non-null   float64 
 26  TRX_COST_USD       544594 non-null   float64 
 27  SALES_CHANNEL      544594 non-null   object  
 28  INVOICE_TYPE        544594 non-null   object  
 29  QUANTITY           544594 non-null   int32  
 30  INVOICE_NUM         544594 non-null   int64  
 31  ORDER_TYPE          544594 non-null   object  
 32  ORDER_NUM           544594 non-null   float64 
 33  PRODUCT_FAMILY      544594 non-null   object  
 34  PRODUCT_MODEL       544594 non-null   object  
 35  UOM                544594 non-null   object
```

```

36 PFV_FAMILY           544594 non-null object
37 PFV_MODEL_GROUP     544594 non-null object
38 Total_SVC_Incidents 483733 non-null float64
39 Total_Repeat_Calls   483733 non-null float64
40 Total_FTF_Calls      483733 non-null float64
41 Most_Frequent_Interaction_Type 536927 non-null object
42 Total_Visits          536927 non-null float64
43 Total_Tasks            536927 non-null float64
44 Total_Cases             187822 non-null float64
45 Max_Case_Origin        187822 non-null object
46 Max_Case_Reason         187822 non-null object
47 Num_of_Active_Install_Bases 495163 non-null float64
48 Total_Contracts        495163 non-null float64
49 Contract_length        495163 non-null float64
50 Num_of_Install_Bases    495163 non-null float64
51 Contract_Category       495163 non-null object
52 Num_of_Inactive_Install_Bases 495163 non-null float64
53 STRATEGIC_ACCOUNTS     544594 non-null int32
54 TERRITORY_REGION        544594 non-null object
dtypes: datetime64[ns](4), float64(19), int32(2), int64(7), object(23)
memory usage: 228.5+ MB

```

3.16 SUPPLIES_DECLINE_REASONs: replace NA with “None” and combine some values

```
[55]: data["SUPPLIES_DECLINE_REASONs"].value_counts()
```

[55]: Migration to 1000 Line	36913
Duplicate Site	26939
AP Competitive Displacement	26598
Over Stocked	24964
Production / Code Reduction	22956
Over Stocked / Timing	21730
Data Integration	19410
Off Brand	13226
Production Down (timing)	10720
Migration to 1000 Line/TIJ/TTO/LCM/LPA	7921
Site Closed	7467
Migration to Lasers	4437
Served by Authorized Distributor	3755
Code Reduction	3557
Recent Regain/Win-back	3444
Recent regain / win back	2470
Seasonal Producer	2237
Partially Moved Equipment	1490
Moved Equipment	1221
Project Based	1194

Seasonal	1099
No More Coding Requirement	1040
Printing/EQ downtime Issues	353
VJ Operations Issues	182
Financial Distress/Credit Hold	87
Pricing / Discounting	58
None	39
Credit Hold	30

Name: SUPPLIES_DECLINE_REASONs, dtype: int64

```
[56]: data["SUPPLIES_DECLINE_REASONs"] = data["SUPPLIES_DECLINE_REASONs"].\n      →fillna('None')
```

```
[57]: data['SUPPLIES_DECLINE_REASONs']=data['SUPPLIES_DECLINE_REASONs'].\\\n      replace('Migration to 1000 Line','Migration to 1000 Line/TIJ/TTO/LCM/LPA')\n      data['SUPPLIES_DECLINE_REASONs']=data['SUPPLIES_DECLINE_REASONs'].\\\n      replace('Over Stocked','Over Stocked / Timing')\n      data['SUPPLIES_DECLINE_REASONs']=data['SUPPLIES_DECLINE_REASONs'].\\\n      replace('Recent regain / win back','Recent Regain/Win-back')\n      data['SUPPLIES_DECLINE_REASONs']=data['SUPPLIES_DECLINE_REASONs'].\\\n      replace('Credit Hold','Financial Distress/Credit Hold')\n      data['SUPPLIES_DECLINE_REASONs']=data['SUPPLIES_DECLINE_REASONs'].\\\n      replace('Code Reduction','Production / Code Reduction')\n      data['SUPPLIES_DECLINE_REASONs']=data['SUPPLIES_DECLINE_REASONs'].\\\n      replace('Seasonal','Seasonal Producer')\n      data['SUPPLIES_DECLINE_REASONs']=data['SUPPLIES_DECLINE_REASONs'].\\\n      replace('Partially Moved Equipment','Moved Equipment')
```

```
[58]: data["SUPPLIES_DECLINE_REASONs"].value_counts()
```

None	299096
Over Stocked / Timing	46694
Migration to 1000 Line/TIJ/TTO/LCM/LPA	44834
Duplicate Site	26939
AP Competitive Displacement	26598
Production / Code Reduction	26513
Data Integration	19410
Off Brand	13226
Production Down (timing)	10720
Site Closed	7467
Recent Regain/Win-back	5914
Migration to Lasers	4437
Served by Authorized Distributor	3755
Seasonal Producer	3336
Moved Equipment	2711
Project Based	1194
No More Coding Requirement	1040

```

Printing/EQ downtime Issues           353
VJ Operations Issues                182
Financial Distress/Credit Hold     117
Pricing / Discounting              58
Name: SUPPLIES_DECLINE_REASONS, dtype: int64

```

```
[59]: # Duplicates and Data Integration refers to shift of business to other sites
      ↵and hence removed to avoid skewness
data = data[(data['SUPPLIES_DECLINE_REASONS'] != "Data Integration") &
      ↵(data['SUPPLIES_DECLINE_REASONS'] != "Duplicate Site")]

```

3.17 IB_CONFIRMED_DATE: drop column

```
[60]: data.drop(['IB_CONFIRMED_DATE'], axis = 1, inplace = True)
```

```
[61]: data.shape
```

```
[61]: (498245, 54)
```

```
[62]: data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 498245 entries, 0 to 544593
Data columns (total 54 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   ITEM_ID          498245 non-null   int64  
 1   SLS_USD          498245 non-null   float64 
 2   Qty               498245 non-null   float64 
 3   Site_Level_Price_Index_WS  498245 non-null   float64 
 4   Site_Level_Price_Index_STU 498245 non-null   float64 
 5   Site_Level_Price_Index  498245 non-null   float64 
 6   CUSTOMER_ID       498245 non-null   int64  
 7   CUSTOMER_SITE_ID  498245 non-null   int64  
 8   SITE_CREATION_DATE 498245 non-null   datetime64[ns]
 9   SHORT_VERTICAL    498245 non-null   object  
 10  CITY              498245 non-null   object  
 11  STATE             498245 non-null   object  
 12  POSTAL_CODE       498245 non-null   object  
 13  CUSTOMER_CLASS    498245 non-null   object  
 14  LAST_MODIFIED_DATE 498245 non-null   datetime64[ns]
 15  TERRITORY_TYPE    498245 non-null   object  
 16  SUPPLIES_SEGMENTATION 498245 non-null   object  
 17  SUPPLIES_DECLINE_REASONS 498245 non-null   object  
 18  IB_STATUS          489871 non-null   object  
 19  IB_STATUS_INACTIVE_REASON 10273 non-null   object  
 20  DUNS_NUMBER        498245 non-null   int64  

```

```

21 CUSTOMER_TRX_ID           498245 non-null int64
22 DW_INVOICE_ID              498245 non-null int64
23 TRX_DATE                   498245 non-null datetime64[ns]
24 TRX_AMT_USD                498245 non-null float64
25 TRX_COST_USD               498245 non-null float64
26 SALES_CHANNEL               498245 non-null object
27 INVOICE_TYPE                 498245 non-null object
28 QUANTITY                     498245 non-null int32
29 INVOICE_NUM                  498245 non-null int64
30 ORDER_TYPE                   498245 non-null object
31 ORDER_NUM                     498245 non-null float64
32 PRODUCT_FAMILY               498245 non-null object
33 PRODUCT_MODEL                 498245 non-null object
34 UOM                           498245 non-null object
35 PFV_FAMILY                    498245 non-null object
36 PFV_MODEL_GROUP              498245 non-null object
37 Total_SVC_Incidents          447499 non-null float64
38 Total_Repeat_Calls           447499 non-null float64
39 Total_FTF_Calls              447499 non-null float64
40 Most_Frequent_Interaction_Type 491811 non-null object
41 Total_Visits                  491811 non-null float64
42 Total_Tasks                   491811 non-null float64
43 Total_Cases                   174903 non-null float64
44 Max_Case-Origin              174903 non-null object
45 Max_Case_Reason               174903 non-null object
46 Num_of_Active_Install_Bases   460306 non-null float64
47 Total_Contracts               460306 non-null float64
48 Contract_length                460306 non-null float64
49 Num_of_Install_Bases          460306 non-null float64
50 Contract_Category              460306 non-null object
51 Num_of_Inactive_Install_Bases  460306 non-null float64
52 STRATEGIC_ACCOUNTS            498245 non-null int32
53 TERRITORY_REGION               498245 non-null object
dtypes: datetime64[ns](3), float64(19), int32(2), int64(7), object(23)
memory usage: 205.3+ MB

```

3.18 IB_STATUS: replace NA with “Active” as last modified date is after 2017

```
[63]: data['IB_STATUS'].value_counts()
```

```
[63]: Active      469693
      Inactive    20178
      Name: IB_STATUS, dtype: int64
```

```
[64]: data['IB_STATUS'].fillna("Active", inplace = True)
```

```
[65]: data.shape
```

```
[65]: (498245, 54)
```

```
[66]: data['IB_STATUS'].value_counts()
```

```
[66]: Active      478067  
Inactive     20178  
Name: IB_STATUS, dtype: int64
```

```
[67]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 498245 entries, 0 to 544593  
Data columns (total 54 columns):  
 #   Column           Non-Null Count  Dtype     
---  --  
 0   ITEM_ID          498245 non-null  int64    
 1   SLS_USD          498245 non-null  float64  
 2   Qty              498245 non-null  float64  
 3   Site_Level_Price_Index_WS 498245 non-null  float64  
 4   Site_Level_Price_Index_STU 498245 non-null  float64  
 5   Site_Level_Price_Index 498245 non-null  float64  
 6   CUSTOMER_ID       498245 non-null  int64    
 7   CUSTOMER_SITE_ID 498245 non-null  int64    
 8   SITE_CREATION_DATE 498245 non-null  datetime64[ns]  
 9   SHORT_VERTICAL    498245 non-null  object    
 10  CITY              498245 non-null  object    
 11  STATE             498245 non-null  object    
 12  POSTAL_CODE       498245 non-null  object    
 13  CUSTOMER_CLASS    498245 non-null  object    
 14  LAST_MODIFIED_DATE 498245 non-null  datetime64[ns]  
 15  TERRITORY_TYPE    498245 non-null  object    
 16  SUPPLIES_SEGMENTATION 498245 non-null  object    
 17  SUPPLIES_DECLINE_REASON 498245 non-null  object    
 18  IB_STATUS          498245 non-null  object    
 19  IB_STATUS_INACTIVE_REASON 10273 non-null  object    
 20  DUNS_NUMBER        498245 non-null  int64    
 21  CUSTOMER_TRX_ID    498245 non-null  int64    
 22  DW_INVOICE_ID      498245 non-null  int64    
 23  TRX_DATE           498245 non-null  datetime64[ns]  
 24  TRX_AMT_USD        498245 non-null  float64  
 25  TRX_COST_USD       498245 non-null  float64  
 26  SALES_CHANNEL      498245 non-null  object    
 27  INVOICE_TYPE        498245 non-null  object    
 28  QUANTITY           498245 non-null  int32    
 29  INVOICE_NUM         498245 non-null  int64    
 30  ORDER_TYPE          498245 non-null  object    
 31  ORDER_NUM           498245 non-null  float64
```

```

32 PRODUCT_FAMILY           498245 non-null object
33 PRODUCT_MODEL            498245 non-null object
34 UOM                      498245 non-null object
35 PFV_FAMILY               498245 non-null object
36 PFV_MODEL_GROUP          498245 non-null object
37 Total_SVC_Incidents      447499 non-null float64
38 Total_Repeat_Calls       447499 non-null float64
39 Total_FTF_Calls          447499 non-null float64
40 Most_Frequent_Interaction_Type 491811 non-null object
41 Total_Visits              491811 non-null float64
42 Total_Tasks                491811 non-null float64
43 Total_Cases                174903 non-null float64
44 Max_Case-Origin           174903 non-null object
45 Max_Case_Reason            174903 non-null object
46 Num_of_Active_Install_Bases 460306 non-null float64
47 Total_Contracts            460306 non-null float64
48 Contract_length             460306 non-null float64
49 Num_of_Install_Bases        460306 non-null float64
50 Contract_Category           460306 non-null object
51 Num_of_Inactive_Install_Bases 460306 non-null float64
52 STRATEGIC_ACCOUNTS         498245 non-null int32
53 TERRITORY_REGION           498245 non-null object
dtypes: datetime64[ns](3), float64(19), int32(2), int64(7), object(23)
memory usage: 205.3+ MB

```

3.19 IB_STATUS_INACTIVE_REASON: drop column

```
[68]: data["IB_STATUS_INACTIVE_REASON"].value_counts()
```

```

[68]: Site Closed                  4471
      AP competitive displacement   3414
      Moved equipment              863
      No more coding requirement    806
      Data Integration              607
      Duplicate Site                 73
      None                           39
      Name: IB_STATUS_INACTIVE_REASON, dtype: int64

```

```
[69]: data.drop(["IB_STATUS_INACTIVE_REASON"], axis = 1, inplace = True)
```

```
[70]: data.shape
```

```
[70]: (498245, 53)
```

```
[71]: data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 498245 entries, 0 to 544593

```

Data columns (total 53 columns):

#	Column	Non-Null Count	Dtype
0	ITEM_ID	498245	non-null int64
1	SLS_USD	498245	non-null float64
2	Qty	498245	non-null float64
3	Site_Level_Price_Index_WS	498245	non-null float64
4	Site_Level_Price_Index_STU	498245	non-null float64
5	Site_Level_Price_Index	498245	non-null float64
6	CUSTOMER_ID	498245	non-null int64
7	CUSTOMER_SITE_ID	498245	non-null int64
8	SITE_CREATION_DATE	498245	non-null datetime64[ns]
9	SHORT_VERTICAL	498245	non-null object
10	CITY	498245	non-null object
11	STATE	498245	non-null object
12	POSTAL_CODE	498245	non-null object
13	CUSTOMER_CLASS	498245	non-null object
14	LAST_MODIFIED_DATE	498245	non-null datetime64[ns]
15	TERRITORY_TYPE	498245	non-null object
16	SUPPLIES_SEGMENTATION	498245	non-null object
17	SUPPLIES_DECLINE_REASONS	498245	non-null object
18	IB_STATUS	498245	non-null object
19	DUNS_NUMBER	498245	non-null int64
20	CUSTOMER_TRX_ID	498245	non-null int64
21	DW_INVOICE_ID	498245	non-null int64
22	TRX_DATE	498245	non-null datetime64[ns]
23	TRX_AMT_USD	498245	non-null float64
24	TRX_COST_USD	498245	non-null float64
25	SALES_CHANNEL	498245	non-null object
26	INVOICE_TYPE	498245	non-null object
27	QUANTITY	498245	non-null int32
28	INVOICE_NUM	498245	non-null int64
29	ORDER_TYPE	498245	non-null object
30	ORDER_NUM	498245	non-null float64
31	PRODUCT_FAMILY	498245	non-null object
32	PRODUCT_MODEL	498245	non-null object
33	UOM	498245	non-null object
34	PFV_FAMILY	498245	non-null object
35	PFV_MODEL_GROUP	498245	non-null object
36	Total_SVC_Incidents	447499	non-null float64
37	Total_Repeat_Calls	447499	non-null float64
38	Total_FTF_Calls	447499	non-null float64
39	Most_Frequent_Interaction_Type	491811	non-null object
40	Total_Visits	491811	non-null float64
41	Total_Tasks	491811	non-null float64
42	Total_Cases	174903	non-null float64
43	Max_Case-Origin	174903	non-null object
44	Max_Case_Reason	174903	non-null object

```

45 Num_of_Active_Install_Bases      460306 non-null float64
46 Total_Contracts                 460306 non-null float64
47 Contract_length                460306 non-null float64
48 Num_of_Install_Bases            460306 non-null float64
49 Contract_Category              460306 non-null object
50 Num_of_Inactive_Install_Bases   460306 non-null float64
51 STRATEGIC_ACCOUNTS             498245 non-null int32
52 TERRITORY_REGION               498245 non-null object
dtypes: datetime64[ns](3), float64(19), int32(2), int64(7), object(22)
memory usage: 201.5+ MB

```

3.20 Site_Level_Price_Index: replace NA by grouping

```
[72]: len(data[data['Site_Level_Price_Index'] == 0])
[72]: 1461

[73]: data['Site_Level_Price_Index'].replace('0', np.nan, inplace = True)

[74]: data['TRX_YEAR'] = pd.DatetimeIndex(data['TRX_DATE']).year

[75]: data['Site_Level_Price_Index'] = data.groupby(['CUSTOMER_SITE_ID', 'ITEM_ID',
   ↴'TRX_YEAR'])['Site_Level_Price_Index'].transform(lambda x: x.fillna(x.
   ↴mean()))
data['Site_Level_Price_Index'] = data.groupby(['CUSTOMER_ID', 'ITEM_ID',
   ↴'TRX_YEAR'])['Site_Level_Price_Index'].transform(lambda x: x.fillna(x.
   ↴mean()))
data['Site_Level_Price_Index'] = data.groupby(['ITEM_ID',
   ↴'TRX_YEAR'])['Site_Level_Price_Index'].transform(lambda x: x.fillna(x.
   ↴mean()))

[76]: data['Site_Level_Price_Index'] = data.groupby(['CUSTOMER_SITE_ID',
   ↴'TRX_YEAR'])['Site_Level_Price_Index'].transform(lambda x: x.fillna(x.
   ↴mean()))
data['Site_Level_Price_Index'] = data.groupby(['CUSTOMER_ID',
   ↴'TRX_YEAR'])['Site_Level_Price_Index'].transform(lambda x: x.fillna(x.
   ↴mean()))

[77]: data['Site_Level_Price_Index'] = data.
   ↴groupby(['TRX_YEAR'])['Site_Level_Price_Index'].transform(lambda x: x.
   ↴fillna(x.mean()))

[78]: data['Site_Level_Price_Index'].isna().sum()
[78]: 0

[79]: data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 498245 entries, 0 to 544593
Data columns (total 54 columns):
 #   Column           Non-Null Count Dtype  
 --- 
 0   ITEM_ID          498245 non-null int64   
 1   SLS_USD          498245 non-null float64 
 2   Qty              498245 non-null float64 
 3   Site_Level_Price_Index_WS 498245 non-null float64 
 4   Site_Level_Price_Index_STU 498245 non-null float64 
 5   Site_Level_Price_Index 498245 non-null float64 
 6   CUSTOMER_ID       498245 non-null int64   
 7   CUSTOMER_SITE_ID 498245 non-null int64   
 8   SITE_CREATION_DATE 498245 non-null datetime64[ns]
 9   SHORT_VERTICAL    498245 non-null object  
 10  CITY              498245 non-null object  
 11  STATE             498245 non-null object  
 12  POSTAL_CODE       498245 non-null object  
 13  CUSTOMER_CLASS    498245 non-null object  
 14  LAST_MODIFIED_DATE 498245 non-null datetime64[ns]
 15  TERRITORY_TYPE    498245 non-null object  
 16  SUPPLIES_SEGMENTATION 498245 non-null object  
 17  SUPPLIES_DECLINE_REASON 498245 non-null object  
 18  IB_STATUS          498245 non-null object  
 19  DUNS_NUMBER         498245 non-null int64   
 20  CUSTOMER_TRX_ID    498245 non-null int64   
 21  DW_INVOICE_ID       498245 non-null int64   
 22  TRX_DATE           498245 non-null datetime64[ns]
 23  TRX_AMT_USD        498245 non-null float64 
 24  TRX_COST_USD       498245 non-null float64 
 25  SALES_CHANNEL      498245 non-null object  
 26  INVOICE_TYPE        498245 non-null object  
 27  QUANTITY           498245 non-null int32  
 28  INVOICE_NUM          498245 non-null int64   
 29  ORDER_TYPE          498245 non-null object  
 30  ORDER_NUM           498245 non-null float64 
 31  PRODUCT_FAMILY      498245 non-null object  
 32  PRODUCT_MODEL        498245 non-null object  
 33  UOM                498245 non-null object  
 34  PFV_FAMILY          498245 non-null object  
 35  PFV_MODEL_GROUP     498245 non-null object  
 36  Total_SVC_Incidents 447499 non-null float64 
 37  Total_Repeat_Calls  447499 non-null float64 
 38  Total_FTF_Calls     447499 non-null float64 
 39  Most_Frequent_Interaction_Type 491811 non-null object  
 40  Total_Visits         491811 non-null float64 
 41  Total_Tasks          491811 non-null float64 
 42  Total_Cases          174903 non-null float64

```

```

43 Max_Case-Origin          174903 non-null object
44 Max_Case-Reason          174903 non-null object
45 Num_of_Active_Install_Bases 460306 non-null float64
46 Total_Contracts          460306 non-null float64
47 Contract_length          460306 non-null float64
48 Num_of_Install_Bases     460306 non-null float64
49 Contract_Category        460306 non-null object
50 Num_of_Inactive_Install_Bases 460306 non-null float64
51 STRATEGIC_ACCOUNTS      498245 non-null int32
52 TERRITORY_REGION         498245 non-null object
53 TRX_YEAR                 498245 non-null int64
dtypes: datetime64[ns](3), float64(19), int32(2), int64(8), object(22)
memory usage: 205.3+ MB

```

3.21 Total_SVC_Incidents / Total_Repeat_Calls / Total_FTF_Calls: replace NA by grouping

```
[80]: # filling NAs using mean of customer, customer class
data['Total_SVC_Incidents'] = data.
    ↪groupby('CUSTOMER_SITE_ID')['Total_SVC_Incidents'].transform(lambda x: x.
    ↪fillna(x.mean()))
data['Total_SVC_Incidents'] = data.
    ↪groupby('CUSTOMER_ID')['Total_SVC_Incidents'].transform(lambda x: x.fillna(x.
    ↪mean()))
data['Total_SVC_Incidents'] = data.
    ↪groupby('CUSTOMER_CLASS')['Total_SVC_Incidents'].transform(lambda x: x.
    ↪fillna(x.mean()))
```

```
[81]: data['Total_SVC_Incidents'].isna().sum()
```

```
[81]: 0
```

```
[82]: # filling NAs using mean of customer, customer class
data['Total_Repeat_Calls'] = data.
    ↪groupby('CUSTOMER_SITE_ID')['Total_Repeat_Calls'].transform(lambda x: x.
    ↪fillna(x.mean()))
data['Total_Repeat_Calls'] = data.groupby('CUSTOMER_ID')['Total_Repeat_Calls'].
    ↪transform(lambda x: x.fillna(x.mean()))
data['Total_Repeat_Calls'] = data.
    ↪groupby('CUSTOMER_CLASS')['Total_Repeat_Calls'].transform(lambda x: x.
    ↪fillna(x.mean()))
```

```
[83]: data['Total_Repeat_Calls'].isna().sum()
```

```
[83]: 0
```

```
[84]: data['Total_FTF_Calls'] = data.groupby('CUSTOMER_SITE_ID')['Total_FTF_Calls'].
      →transform(lambda x: x.fillna(x.mean()))
data['Total_FTF_Calls'] = data.groupby('CUSTOMER_ID')['Total_FTF_Calls'].
      →transform(lambda x: x.fillna(x.mean()))
data['Total_FTF_Calls'] = data.groupby('CUSTOMER_CLASS')['Total_FTF_Calls'].
      →transform(lambda x: x.fillna(x.mean()))
```

```
[85]: data['Total_FTF_Calls'].isna().sum()
```

```
[85]: 0
```

3.22 Most_Frequent_Interaction_Type: replace NA using “Call”

```
[86]: data['Most_Frequent_Interaction_Type'].value_counts()
```

```
[86]: Call          413396
Email         42770
Other         31697
Meeting        3030
Customer Meeting    747
Dial            70
Contact Customer   49
Make Qualified Sales Call 23
Callback        15
TS Task          13
Survey           1
Name: Most_Frequent_Interaction_Type, dtype: int64
```

```
[87]: data['Most_Frequent_Interaction_Type'] = data['Most_Frequent_Interaction_Type'].
      →replace(np.nan, "Call")
```

```
[88]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 498245 entries, 0 to 544593
Data columns (total 54 columns):
 #   Column           Non-Null Count  Dtype  
 ---  --  
 0   ITEM_ID          498245 non-null   int64  
 1   SLS_USD          498245 non-null   float64 
 2   Qty              498245 non-null   float64 
 3   Site_Level_Price_Index_WS 498245 non-null   float64 
 4   Site_Level_Price_Index_STU 498245 non-null   float64 
 5   Site_Level_Price_Index 498245 non-null   float64 
 6   CUSTOMER_ID       498245 non-null   int64  
 7   CUSTOMER_SITE_ID 498245 non-null   int64  
 8   SITE_CREATION_DATE 498245 non-null   datetime64[ns]
```

```

9   SHORT_VERTICAL           498245 non-null object
10  CITY                     498245 non-null object
11  STATE                    498245 non-null object
12  POSTAL_CODE              498245 non-null object
13  CUSTOMER_CLASS            498245 non-null object
14  LAST_MODIFIED_DATE        498245 non-null datetime64[ns]
15  TERRITORY_TYPE            498245 non-null object
16  SUPPLIES_SEGMENTATION     498245 non-null object
17  SUPPLIES_DECLINE_REASONS 498245 non-null object
18  IB_STATUS                 498245 non-null object
19  DUNS_NUMBER                498245 non-null int64
20  CUSTOMER_TRX_ID            498245 non-null int64
21  DW_INVOICE_ID               498245 non-null int64
22  TRX_DATE                  498245 non-null datetime64[ns]
23  TRX_AMT_USD                498245 non-null float64
24  TRX_COST_USD               498245 non-null float64
25  SALES_CHANNEL              498245 non-null object
26  INVOICE_TYPE                498245 non-null object
27  QUANTITY                   498245 non-null int32
28  INVOICE_NUM                  498245 non-null int64
29  ORDER_TYPE                  498245 non-null object
30  ORDER_NUM                   498245 non-null float64
31  PRODUCT_FAMILY              498245 non-null object
32  PRODUCT_MODEL                498245 non-null object
33  UOM                         498245 non-null object
34  PFV_FAMILY                  498245 non-null object
35  PFV_MODEL_GROUP              498245 non-null object
36  Total_SVC_Incidents          498245 non-null float64
37  Total_Repeat_Calls            498245 non-null float64
38  Total_FTF_Calls               498245 non-null float64
39  Most_Frequent_Interaction_Type 498245 non-null object
40  Total_Visits                  491811 non-null float64
41  Total_Tasks                   491811 non-null float64
42  Total_Cases                   174903 non-null float64
43  Max_Case-Origin              174903 non-null object
44  Max_Case_Reason                174903 non-null object
45  Num_of_Active_Install_Bases    460306 non-null float64
46  Total_Contracts                460306 non-null float64
47  Contract_length                460306 non-null float64
48  Num_of_Install_Bases             460306 non-null float64
49  Contract_Category               460306 non-null object
50  Num_of_Inactive_Install_Bases   460306 non-null float64
51  STRATEGIC_ACCOUNTS              498245 non-null int32
52  TERRITORY_REGION                498245 non-null object
53  TRX_YEAR                      498245 non-null int64
dtypes: datetime64[ns](3), float64(19), int32(2), int64(8), object(22)
memory usage: 205.3+ MB

```

3.23 Total_Visits / Total_Tasks: replace NA by grouping

```
[89]: # filling NAs using mean of customer, customer class
data['Total_Visits'] = data.groupby('CUSTOMER_SITE_ID')['Total_Visits'].
    →transform(lambda x: x.fillna(x.mean()))
data['Total_Visits'] = data.groupby('CUSTOMER_ID')['Total_Visits'].
    →transform(lambda x: x.fillna(x.mean()))
data['Total_Visits'] = data.groupby('CUSTOMER_CLASS')['Total_Visits'].
    →transform(lambda x: x.fillna(x.mean()))

data['Total_Tasks'] = data.groupby('CUSTOMER_SITE_ID')['Total_Tasks'].
    →transform(lambda x: x.fillna(x.mean()))
data['Total_Tasks'] = data.groupby('CUSTOMER_ID')['Total_Tasks'].
    →transform(lambda x: x.fillna(x.mean()))
data['Total_Tasks'] = data.groupby('CUSTOMER_CLASS')['Total_Tasks'].
    →transform(lambda x: x.fillna(x.mean()))
```

```
[90]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 498245 entries, 0 to 544593
Data columns (total 54 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   ITEM_ID          498245 non-null   int64  
 1   SLS_USD          498245 non-null   float64 
 2   Qty              498245 non-null   float64 
 3   Site_Level_Price_Index_WS 498245 non-null   float64 
 4   Site_Level_Price_Index_STU 498245 non-null   float64 
 5   Site_Level_Price_Index 498245 non-null   float64 
 6   CUSTOMER_ID       498245 non-null   int64  
 7   CUSTOMER_SITE_ID 498245 non-null   int64  
 8   SITE_CREATION_DATE 498245 non-null   datetime64[ns]
 9   SHORT_VERTICAL    498245 non-null   object  
 10  CITY              498245 non-null   object  
 11  STATE             498245 non-null   object  
 12  POSTAL_CODE       498245 non-null   object  
 13  CUSTOMER_CLASS    498245 non-null   object  
 14  LAST_MODIFIED_DATE 498245 non-null   datetime64[ns]
 15  TERRITORY_TYPE    498245 non-null   object  
 16  SUPPLIES_SEGMENTATION 498245 non-null   object  
 17  SUPPLIES_DECLINE_REASONS 498245 non-null   object  
 18  IB_STATUS          498245 non-null   object  
 19  DUNS_NUMBER         498245 non-null   int64  
 20  CUSTOMER_TRX_ID    498245 non-null   int64  
 21  DW_INVOICE_ID       498245 non-null   int64  
 22  TRX_DATE           498245 non-null   datetime64[ns]
```

```

23 TRX_AMT_USD           498245 non-null float64
24 TRX_COST_USD          498245 non-null float64
25 SALES_CHANNEL          498245 non-null object
26 INVOICE_TYPE            498245 non-null object
27 QUANTITY                498245 non-null int32
28 INVOICE_NUM              498245 non-null int64
29 ORDER_TYPE                498245 non-null object
30 ORDER_NUM                 498245 non-null float64
31 PRODUCT_FAMILY             498245 non-null object
32 PRODUCT_MODEL              498245 non-null object
33 UOM                      498245 non-null object
34 PFV_FAMILY                  498245 non-null object
35 PFV_MODEL_GROUP             498245 non-null object
36 Total_SVC_Incidents        498245 non-null float64
37 Total_Repeat_Calls         498245 non-null float64
38 Total_FTF_Calls            498245 non-null float64
39 Most_Frequent_Interaction_Type 498245 non-null object
40 Total_Visits                498245 non-null float64
41 Total_Tasks                  498245 non-null float64
42 Total_Cases                  174903 non-null float64
43 Max_Case-Origin              174903 non-null object
44 Max_Case_Reason               174903 non-null object
45 Num_of_Active_Install_Bases   460306 non-null float64
46 Total_Contracts              460306 non-null float64
47 Contract_length                460306 non-null float64
48 Num_of_Install_Bases           460306 non-null float64
49 Contract_Category              460306 non-null object
50 Num_of_Inactive_Install_Bases  460306 non-null float64
51 STRATEGIC_ACCOUNTS            498245 non-null int32
52 TERRITORY_REGION              498245 non-null object
53 TRX_YEAR                     498245 non-null int64
dtypes: datetime64[ns](3), float64(19), int32(2), int64(8), object(22)
memory usage: 205.3+ MB

```

3.24 Total_Cases: replace NA by grouping

```
[91]: # filling NAs using mean of customer, customer class
data['Total_Cases'] = data.groupby('CUSTOMER_SITE_ID')['Total_Cases'].
    ↪transform(lambda x: x.fillna(x.mean()))
data['Total_Cases'] = data.groupby('CUSTOMER_ID')['Total_Cases'].
    ↪transform(lambda x: x.fillna(x.mean()))
data['Total_Cases'] = data.groupby('CUSTOMER_CLASS')['Total_Cases'].
    ↪transform(lambda x: x.fillna(x.mean()))
```

```
[92]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

Int64Index: 498245 entries, 0 to 544593

Data columns (total 54 columns):

#	Column	Non-Null Count	Dtype
0	ITEM_ID	498245	non-null int64
1	SLS_USD	498245	non-null float64
2	Qty	498245	non-null float64
3	Site_Level_Price_Index_WS	498245	non-null float64
4	Site_Level_Price_Index_STU	498245	non-null float64
5	Site_Level_Price_Index	498245	non-null float64
6	CUSTOMER_ID	498245	non-null int64
7	CUSTOMER_SITE_ID	498245	non-null int64
8	SITE_CREATION_DATE	498245	non-null datetime64[ns]
9	SHORT_VERTICAL	498245	non-null object
10	CITY	498245	non-null object
11	STATE	498245	non-null object
12	POSTAL_CODE	498245	non-null object
13	CUSTOMER_CLASS	498245	non-null object
14	LAST_MODIFIED_DATE	498245	non-null datetime64[ns]
15	TERRITORY_TYPE	498245	non-null object
16	SUPPLIES_SEGMENTATION	498245	non-null object
17	SUPPLIES_DECLINE_REASONS	498245	non-null object
18	IB_STATUS	498245	non-null object
19	DUNS_NUMBER	498245	non-null int64
20	CUSTOMER_TRX_ID	498245	non-null int64
21	DW_INVOICE_ID	498245	non-null int64
22	TRX_DATE	498245	non-null datetime64[ns]
23	TRX_AMT_USD	498245	non-null float64
24	TRX_COST_USD	498245	non-null float64
25	SALES_CHANNEL	498245	non-null object
26	INVOICE_TYPE	498245	non-null object
27	QUANTITY	498245	non-null int32
28	INVOICE_NUM	498245	non-null int64
29	ORDER_TYPE	498245	non-null object
30	ORDER_NUM	498245	non-null float64
31	PRODUCT_FAMILY	498245	non-null object
32	PRODUCT_MODEL	498245	non-null object
33	UOM	498245	non-null object
34	PFV_FAMILY	498245	non-null object
35	PFV_MODEL_GROUP	498245	non-null object
36	Total_SVC_Incidents	498245	non-null float64
37	Total_Repeat_Calls	498245	non-null float64
38	Total_FTF_Calls	498245	non-null float64
39	Most_Frequent_Interaction_Type	498245	non-null object
40	Total_Visits	498245	non-null float64
41	Total_Tasks	498245	non-null float64
42	Total_Cases	498245	non-null float64
43	Max_Case-Origin	174903	non-null object

```

44 Max_Case_Reason           174903 non-null object
45 Num_of_Active_Install_Bases 460306 non-null float64
46 Total_Contracts           460306 non-null float64
47 Contract_length           460306 non-null float64
48 Num_of_Install_Bases       460306 non-null float64
49 Contract_Category          460306 non-null object
50 Num_of_Inactive_Install_Bases 460306 non-null float64
51 STRATEGIC_ACCOUNTS        498245 non-null int32
52 TERRITORY_REGION           498245 non-null object
53 TRX_YEAR                   498245 non-null int64
dtypes: datetime64[ns](3), float64(19), int32(2), int64(8), object(22)
memory usage: 205.3+ MB

```

3.25 Max_Case-Origin/Max_Case_Reason: replace NA with ‘unknown’

```
[93]: data['Max_Case_Origin'].value_counts()
```

```

[93]: Email - VTI NACC           94923
      Phone                         50717
      CC Survey Followup            16512
      TS Survey Followup            6771
      FS Survey Followup            2634
      Email/Fax - VTI CS           1856
      Install Complete              631
      CX Survey Detractor          583
      Email                          273
      Email - VTI CC Sales Escalations 3
      Name: Max_Case_Origin, dtype: int64

```

```
[94]: data['Max_Case_Reason'].value_counts()
```

```

[94]: Customer Experience     146775
      CX: Customer Care         13882
      CX: Tech Support            7849
      CX: Field Service           4279
      CX: Credit                  1351
      CX: Field Sales              363
      CX: Other Team               349
      CX: Manufacturing             55
      Name: Max_Case_Reason, dtype: int64

```

```
[95]: data['Max_Case_Origin'] = data['Max_Case_Origin'].fillna('unknown')
      data['Max_Case_Reason'] = data['Max_Case_Reason'].fillna('unknown')
```

```
[96]: data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 498245 entries, 0 to 544593

```

Data columns (total 54 columns):

#	Column	Non-Null Count	Dtype
0	ITEM_ID	498245	non-null int64
1	SLS_USD	498245	non-null float64
2	Qty	498245	non-null float64
3	Site_Level_Price_Index_WS	498245	non-null float64
4	Site_Level_Price_Index_STU	498245	non-null float64
5	Site_Level_Price_Index	498245	non-null float64
6	CUSTOMER_ID	498245	non-null int64
7	CUSTOMER_SITE_ID	498245	non-null int64
8	SITE_CREATION_DATE	498245	non-null datetime64[ns]
9	SHORT_VERTICAL	498245	non-null object
10	CITY	498245	non-null object
11	STATE	498245	non-null object
12	POSTAL_CODE	498245	non-null object
13	CUSTOMER_CLASS	498245	non-null object
14	LAST_MODIFIED_DATE	498245	non-null datetime64[ns]
15	TERRITORY_TYPE	498245	non-null object
16	SUPPLIES_SEGMENTATION	498245	non-null object
17	SUPPLIES_DECLINE_REASONS	498245	non-null object
18	IB_STATUS	498245	non-null object
19	DUNS_NUMBER	498245	non-null int64
20	CUSTOMER_TRX_ID	498245	non-null int64
21	DW_INVOICE_ID	498245	non-null int64
22	TRX_DATE	498245	non-null datetime64[ns]
23	TRX_AMT_USD	498245	non-null float64
24	TRX_COST_USD	498245	non-null float64
25	SALES_CHANNEL	498245	non-null object
26	INVOICE_TYPE	498245	non-null object
27	QUANTITY	498245	non-null int32
28	INVOICE_NUM	498245	non-null int64
29	ORDER_TYPE	498245	non-null object
30	ORDER_NUM	498245	non-null float64
31	PRODUCT_FAMILY	498245	non-null object
32	PRODUCT_MODEL	498245	non-null object
33	UOM	498245	non-null object
34	PFV_FAMILY	498245	non-null object
35	PFV_MODEL_GROUP	498245	non-null object
36	Total_SVC_Incidents	498245	non-null float64
37	Total_Repeat_Calls	498245	non-null float64
38	Total_FTF_Calls	498245	non-null float64
39	Most_Frequent_Interaction_Type	498245	non-null object
40	Total_Visits	498245	non-null float64
41	Total_Tasks	498245	non-null float64
42	Total_Cases	498245	non-null float64
43	Max_Case-Origin	498245	non-null object
44	Max_Case_Reason	498245	non-null object

```

45 Num_of_Active_Install_Bases      460306 non-null float64
46 Total_Contracts                 460306 non-null float64
47 Contract_length                 460306 non-null float64
48 Num_of_Install_Bases            460306 non-null float64
49 Contract_Category               460306 non-null object
50 Num_of_Inactive_Install_Bases   460306 non-null float64
51 STRATEGIC_ACCOUNTS             498245 non-null int32
52 TERRITORY_REGION                498245 non-null object
53 TRX_YEAR                        498245 non-null int64
dtypes: datetime64[ns](3), float64(19), int32(2), int64(8), object(22)
memory usage: 205.3+ MB

```

3.26 Contract_Category: drop rows with NA

```
[97]: data = data.dropna(axis=0, subset=['Contract_Category'])
```

4 Converting data types

```
[98]: # convert to right dtype
data['CUSTOMER_SITE_ID'] = data['CUSTOMER_SITE_ID'].astype(str)
data['CUSTOMER_ID'] = data['CUSTOMER_ID'].astype(str)
data['CUSTOMER_TRX_ID'] = data['CUSTOMER_TRX_ID'].astype(str)
data['DW_INVOICE_ID'] = data['DW_INVOICE_ID'].astype(str)
data['ITEM_ID'] = data['ITEM_ID'].astype(str)
data['INVOICE_NUM'] = data['INVOICE_NUM'].astype(str)
data['ORDER_NUM'] = data['ORDER_NUM'].astype(str)
```

```
[99]: data.head()
```

```
[99]: ITEM_ID  SLS_USD  Qty  Site_Level_Price_Index_WS \
0  268183  1,246.32 6.00          1,001.87
1  598740    644.37 3.00          588.99
2  268183  1,246.32 6.00          1,001.87
3  598740    644.37 3.00          588.99
4  268183  1,246.32 6.00          1,001.87

Site_Level_Price_Index_STU  Site_Level_Price_Index CUSTOMER_ID \
0                  1,246.32          0.80      117841
1                  644.37          0.91      113032
2                  1,246.32          0.80      117841
3                  644.37          0.91      113032
4                  1,246.32          0.80      117841

CUSTOMER_SITE_ID SITE_CREATION_DATE      SHORT_VERTICAL      CITY STATE \
0           609636  2013-08-15  FRUIT & VEGETABLE      SALEM     OR
1           578406  2013-03-25  AERO/AUTO      CRESTVIEW     FL
```

2	609636	2013-08-15	FRUIT & VEGETABLE	SALEM	OR		
3	578406	2013-03-25	AERO/AUTO	CRESTVIEW	FL		
4	609636	2013-08-15	FRUIT & VEGETABLE	SALEM	OR		
	POSTAL_CODE	CUSTOMER_CLASS	LAST_MODIFIED_DATE	TERRITORY_TYPE	\		
0	97301	END USER	2019-09-30	Industrial			
1	32539	END USER	2019-10-01	Industrial			
2	97301	END USER	2019-09-30	Industrial			
3	32539	END USER	2019-10-01	Industrial			
4	97301	END USER	2019-09-30	Industrial			
	SUPPLIES_SEGMENTATION	SUPPLIES_DECLINE_REASON	IB_STATUS	DUNS_NUMBER	\		
0	S	Over Stocked / Timing	Active	78842640			
1	S	None	Active	43202248			
2	S	Over Stocked / Timing	Active	78842640			
3	S	None	Active	43202248			
4	S	Over Stocked / Timing	Active	78842640			
	CUSTOMER_TRX_ID	DW_INVOICE_ID	TRX_DATE	TRX_AMT_USD	TRX_COST_USD	\	
0	8731929	108745302	2016-02-05	207.72	19.13		
1	9305874	276000052	2016-12-16	214.79	15.85		
2	9093354	223068431	2016-08-26	207.72	19.13		
3	9195970	247802392	2016-10-19	429.58	31.69		
4	9079815	219738849	2016-08-19	623.16	57.39		
	SALES_CHANNEL	INVOICE_TYPE	QUANTITY	INVOICE_NUM	ORDER_TYPE	\	
0	Online	INVOICE	1	3206875	STANDARD DOMESTIC		
1	Esker	INVOICE	1	3294491	STANDARD DOMESTIC		
2	Online	INVOICE	1	3263477	STANDARD DOMESTIC		
3	Esker	INVOICE	2	3278903	STANDARD DOMESTIC		
4	Online	INVOICE	3	3261624	STANDARD DOMESTIC		
	ORDER_NUM	PRODUCT_FAMILY	PRODUCT_MODEL	UOM	PFV_FAMILY	PFV_MODEL_GROUP	\
0	511759211.0	CIJ	MAKE-UP	Each	CIJ	CIJ - LEGACY	
1	511868043.0	CIJ	VALUE PACK	Each	CIJ	CIJ - LEGACY	
2	511823154.0	CIJ	MAKE-UP	Each	CIJ	CIJ - LEGACY	
3	511849315.0	CIJ	VALUE PACK	Each	CIJ	CIJ - LEGACY	
4	511823154.0	CIJ	MAKE-UP	Each	CIJ	CIJ - LEGACY	
	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls				\
0	11.00	3.00	8.00				
1	21.00	5.00	16.00				
2	11.00	3.00	8.00				
3	21.00	5.00	16.00				
4	11.00	3.00	8.00				
	Most_Frequent_Interaction_Type	Total_Visits	Total_Tasks	Total_Cases			\

0	Call	18.00	17.00	1.97
1	Email	58.00	62.00	1.00
2	Call	18.00	17.00	1.97
3	Email	58.00	62.00	1.00
4	Call	18.00	17.00	1.97
0	Max_Case_Origin	Max_Case_Reason	Num_of_Active_Install_Bases	\
1	Email - VTI NACC	Customer Experience		6.00
2	unknown	unknown		4.00
3	Email - VTI NACC	Customer Experience		6.00
4	unknown	unknown		4.00
0	Total_Contracts	Contract_length	Num_of_Install_Bases	Contract_Category \
1	0.00	0.00	6.00	No Contract
2	4.00	1,011.25	4.00	FSMA
3	0.00	0.00	6.00	No Contract
4	4.00	1,011.25	4.00	FSMA
0	0.00	0.00	6.00	No Contract
1	0.00	0.00	6.00	FSMA
2	0.00	0.00	6.00	No Contract
3	0.00	0.00	6.00	FSMA
4	0.00	0.00	6.00	No Contract
0	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	TERRITORY_REGION	\
1	0.00	0	NW	
2	0.00	0	SE	
3	0.00	0	NW	
4	0.00	0	SE	
0	0.00	0	NW	
1	0.00	0	SE	
2	0.00	0	NW	
3	0.00	0	SE	
4	0.00	0	NW	
0	TRX_YEAR			
1	2016			
2	2016			
3	2016			
4	2016			

5 Dropping non-relevant columns

```
[100]: data.drop(['ITEM_ID', 'SLS_USD', 'Qty', 'Site_Level_Price_Index_WS',
   'Site_Level_Price_Index_STU', 'SITE_CREATION_DATE', 'CITY',
   'STATE', 'IB_STATUS', 'CUSTOMER_TRX_ID',
   ↴'DW_INVOICE_ID', 'Num_of_Install_Bases', 'Total_Tasks', 'PFV_FAMILY', 'PFV_MODEL_GROUP', 'UOM',
   ↵= 1,inplace = True)
```



```
[101]: # create Margin column
data['TRX_COST_USD'] = data['TRX_AMT_USD'] - data['TRX_COST_USD']
```

```
[102]: data.rename(columns={'TRX_COST_USD':'Margin'},inplace = True)
```

```
[103]: data.shape
```

```
[103]: (460306, 35)
```

```
[104]: data.head()
```

```
[104]:   Site_Level_Price_Index CUSTOMER_ID CUSTOMER_SITE_ID      SHORT_VERTICAL \
0           0.80       117841        609636  FRUIT & VEGETABLE
1           0.91       113032        578406    AERO/AUTO
2           0.80       117841        609636  FRUIT & VEGETABLE
3           0.91       113032        578406    AERO/AUTO
4           0.80       117841        609636  FRUIT & VEGETABLE

  POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE SUPPLIES_SEGMENTATION \
0      97301      END USER     Industrial          S
1      32539      END USER     Industrial          S
2      97301      END USER     Industrial          S
3      32539      END USER     Industrial          S
4      97301      END USER     Industrial          S

  SUPPLIES_DECLINE_REASON DUNS_NUMBER   TRX_DATE  TRX_AMT_USD Margin \
0  Over Stocked / Timing    78842640 2016-02-05      207.72  188.59
1             None        43202248 2016-12-16      214.79  198.94
2  Over Stocked / Timing    78842640 2016-08-26      207.72  188.59
3             None        43202248 2016-10-19      429.58  397.89
4  Over Stocked / Timing    78842640 2016-08-19      623.16  565.77

  SALES_CHANNEL QUANTITY ORDER_TYPE ORDER_NUM PRODUCT_FAMILY \
0      Online         1  STANDARD DOMESTIC  511759211.0        CIJ
1      Esker          1  STANDARD DOMESTIC  511868043.0        CIJ
2      Online         1  STANDARD DOMESTIC  511823154.0        CIJ
3      Esker          2  STANDARD DOMESTIC  511849315.0        CIJ
4      Online         3  STANDARD DOMESTIC  511823154.0        CIJ

  PRODUCT_MODEL Total_SVC_Incidents Total_Repeat_Calls Total_FTF_Calls \
0      MAKE-UP            11.00                3.00            8.00
1  VALUE PACK            21.00                5.00            16.00
2      MAKE-UP            11.00                3.00            8.00
3  VALUE PACK            21.00                5.00            16.00
4      MAKE-UP            11.00                3.00            8.00

  Most_Frequent_Interaction_Type Total_Visits Total_Cases Max_Case_Origin \
0                  Call        18.00      1.97      unknown
1                  Email        58.00      1.00  Email - VTI NACC
2                  Call        18.00      1.97      unknown
```

```

3                           Email        58.00      1.00 Email - VTI NACC
4                           Call         18.00      1.97 unknown

      Max_Case_Reason  Num_of_Active_Install_Bases  Total_Contracts \
0           unknown                  6.00          0.00
1 Customer Experience                   4.00          4.00
2           unknown                  6.00          0.00
3 Customer Experience                   4.00          4.00
4           unknown                  6.00          0.00

      Contract_length Contract_Category  Num_of_Inactive_Install_Bases \
0            0.00        No Contract                  0.00
1      1,011.25             FSMA                  0.00
2            0.00        No Contract                  0.00
3      1,011.25             FSMA                  0.00
4            0.00        No Contract                  0.00

      STRATEGIC_ACCOUNTS TERRITORY_REGION  TRX_YEAR
0                 0             NW       2016
1                 0             SE       2016
2                 0             NW       2016
3                 0             SE       2016
4                 0             NW       2016

```

```
[105]: data.drop_duplicates(inplace = True)
```

6 Dropping discrepancies

```
[106]: # contract mismatch rows to be removed
data = data.loc[~((data['Contract_Category'] == "No Contract") &
                  (data['Contract_length'] > 0)),:]
```

```
[107]: # drop negative margin cases
data = data[data['Margin'] > 0]
```

```
[108]: data.shape
```

```
[108]: (380952, 35)
```

```
[109]: data.head()
```

```
[109]: Site_Level_Price_Index CUSTOMER_ID CUSTOMER_SITE_ID    SHORT_VERTICAL \
0            0.80     117841      609636 FRUIT & VEGETABLE
1            0.91     113032      578406 AERO/AUTO
2            0.80     117841      609636 FRUIT & VEGETABLE
3            0.91     113032      578406 AERO/AUTO
```

4	0.80	117841	609636	FRUIT & VEGETABLE
POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE SUPPLIES_SEGMENTATION \				
0	97301	END USER	Industrial	S
1	32539	END USER	Industrial	S
2	97301	END USER	Industrial	S
3	32539	END USER	Industrial	S
4	97301	END USER	Industrial	S
SUPPLIES_DECLINE_REASON DUNS_NUMBER TRX_DATE TRX_AMT_USD Margin \				
0	Over Stocked / Timing	78842640	2016-02-05	207.72 188.59
1	None	43202248	2016-12-16	214.79 198.94
2	Over Stocked / Timing	78842640	2016-08-26	207.72 188.59
3	None	43202248	2016-10-19	429.58 397.89
4	Over Stocked / Timing	78842640	2016-08-19	623.16 565.77
SALES_CHANNEL QUANTITY ORDER_TYPE ORDER_NUM PRODUCT_FAMILY \				
0	Online	1	STANDARD DOMESTIC	511759211.0 CIJ
1	Esker	1	STANDARD DOMESTIC	511868043.0 CIJ
2	Online	1	STANDARD DOMESTIC	511823154.0 CIJ
3	Esker	2	STANDARD DOMESTIC	511849315.0 CIJ
4	Online	3	STANDARD DOMESTIC	511823154.0 CIJ
PRODUCT_MODEL Total_SVC_Incidents Total_Repeat_Calls Total_FTF_Calls \				
0	MAKE-UP	11.00	3.00	8.00
1	VALUE PACK	21.00	5.00	16.00
2	MAKE-UP	11.00	3.00	8.00
3	VALUE PACK	21.00	5.00	16.00
4	MAKE-UP	11.00	3.00	8.00
Most_Frequent_Interaction_Type Total_Visits Total_Cases Max_Case_Origin \				
0	Call	18.00	1.97	unknown
1	Email	58.00	1.00	Email - VTI NACC
2	Call	18.00	1.97	unknown
3	Email	58.00	1.00	Email - VTI NACC
4	Call	18.00	1.97	unknown
Max_Case_Reason Num_of_Active_Install_Bases Total_Contracts \				
0	unknown	6.00	0.00	
1	Customer Experience	4.00	4.00	
2	unknown	6.00	0.00	
3	Customer Experience	4.00	4.00	
4	unknown	6.00	0.00	
Contract_length Contract_Category Num_of_Inactive_Install_Bases \				
0	0.00	No Contract	0.00	
1	1,011.25	FSMA	0.00	

```

2          0.00      No Contract          0.00
3      1,011.25      FSMA              0.00
4          0.00      No Contract          0.00

  STRATEGIC_ACCOUNTS TERRITORY_REGION  TRX_YEAR
0                  0          NW      2016
1                  0          SE      2016
2                  0          NW      2016
3                  0          SE      2016
4                  0          NW      2016

```

7 Removing one-time purchasers

```

[110]: num_trxn_overall = data.groupby('CUSTOMER_SITE_ID')['TRX_DATE'].count().
       reset_index()
num_trxn_overall.rename(columns = {'TRX_DATE': 'Num_of_Trxns'}, inplace = True)

[111]: # remove customers with a single transaction (one-time purchasers)
single_purchaser = num_trxn_overall[num_trxn_overall['Num_of_Trxns'] <= 1]

[112]: single_purchaser_list = list(single_purchaser['CUSTOMER_SITE_ID'])

[113]: len(num_trxn_overall)

[113]: 8918

[114]: data = data[~data['CUSTOMER_SITE_ID'].isin(single_purchaser_list)]

[115]: data.shape

[115]: (380401, 35)

[116]: data.head()

[116]:   Site_Level_Price_Index CUSTOMER_ID CUSTOMER_SITE_ID      SHORT_VERTICAL \
0            0.80      117841      609636  FRUIT & VEGETABLE
1            0.91      113032      578406    AERO/AUTO
2            0.80      117841      609636  FRUIT & VEGETABLE
3            0.91      113032      578406    AERO/AUTO
4            0.80      117841      609636  FRUIT & VEGETABLE

  POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE SUPPLIES_SEGMENTATION \
0      97301      END USER    Industrial             S
1      32539      END USER    Industrial             S
2      97301      END USER    Industrial             S
3      32539      END USER    Industrial             S

```

4	97301	END USER	Industrial	S			
		SUPPLIES_DECLINE_REASON	DUNS_NUMBER	TRX_DATE	TRX_AMT_USD	Margin	\
0	Over Stocked / Timing	None	78842640	2016-02-05	207.72	188.59	
1		None	43202248	2016-12-16	214.79	198.94	
2	Over Stocked / Timing	None	78842640	2016-08-26	207.72	188.59	
3		None	43202248	2016-10-19	429.58	397.89	
4	Over Stocked / Timing		78842640	2016-08-19	623.16	565.77	
	SALES_CHANNEL	QUANTITY	ORDER_TYPE	ORDER_NUM	PRODUCT_FAMILY	\	
0	Online	1	STANDARD DOMESTIC	511759211.0		CIJ	
1	Esker	1	STANDARD DOMESTIC	511868043.0		CIJ	
2	Online	1	STANDARD DOMESTIC	511823154.0		CIJ	
3	Esker	2	STANDARD DOMESTIC	511849315.0		CIJ	
4	Online	3	STANDARD DOMESTIC	511823154.0		CIJ	
	PRODUCT_MODEL	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\		
0	MAKE-UP	11.00	3.00	8.00			
1	VALUE PACK	21.00	5.00	16.00			
2	MAKE-UP	11.00	3.00	8.00			
3	VALUE PACK	21.00	5.00	16.00			
4	MAKE-UP	11.00	3.00	8.00			
	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	Max_Case_Origin	\		
0	Call	18.00	1.97	unknown			
1	Email	58.00	1.00	Email - VTI NACC			
2	Call	18.00	1.97	unknown			
3	Email	58.00	1.00	Email - VTI NACC			
4	Call	18.00	1.97	unknown			
	Max_Case_Reason	Num_of_Active_Install_Bases	Total_Contracts	\			
0	unknown	6.00	0.00				
1	Customer Experience	4.00	4.00				
2	unknown	6.00	0.00				
3	Customer Experience	4.00	4.00				
4	unknown	6.00	0.00				
	Contract_length	Contract_Category	Num_of_Inactive_Install_Bases	\			
0	0.00	No Contract	0.00				
1	1,011.25	FSMA	0.00				
2	0.00	No Contract	0.00				
3	1,011.25	FSMA	0.00				
4	0.00	No Contract	0.00				
	STRATEGIC_ACCOUNTS	TERRITORY_REGION	TRX_YEAR				
0	0	NW	2016				
1	0	SE	2016				

2	0	NW	2016
3	0	SE	2016
4	0	NW	2016

8 Exporting result table

```
[117]: data.to_csv("cleaned_data.csv")
```

Part 4 - Variable Creation

February 12, 2021

```
[1]: # import relevant libraries
import pandas as pd
import scipy.stats
import numpy as np
import datetime as dt
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import datetime as dt
import warnings
warnings.filterwarnings("ignore")

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('float_format', '{:.2f}'.format)
```

1 Importing dataset

```
[2]: # import dataset
df = pd.read_csv('cleaned_data.csv', index_col = 0)
```

```
[3]: df.shape
```

```
[3]: (380401, 35)
```

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 380401 entries, 0 to 544593
Data columns (total 35 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   Site_Level_Price_Index    380401 non-null   float64 
 1   CUSTOMER_ID              380401 non-null   int64  
 2   CUSTOMER_SITE_ID         380401 non-null   int64  
 3   SHORT_VERTICAL            380401 non-null   object  
 4   POSTAL_CODE               380401 non-null   int64
```

```

5   CUSTOMER_CLASS           380401 non-null object
6   TERRITORY_TYPE          380401 non-null object
7   SUPPLIES_SEGMENTATION   380401 non-null object
8   SUPPLIES_DECLINE_REASONS 380401 non-null object
9   DUNS_NUMBER              380401 non-null int64
10  TRX_DATE                 380401 non-null object
11  TRX_AMT_USD             380401 non-null float64
12  Margin                    380401 non-null float64
13  SALES_CHANNEL            380401 non-null object
14  QUANTITY                  380401 non-null int64
15  ORDER_TYPE                380401 non-null object
16  ORDER_NUM                  380401 non-null float64
17  PRODUCT_FAMILY             380401 non-null object
18  PRODUCT_MODEL              380401 non-null object
19  Total_SVC_Incidents       380401 non-null float64
20  Total_Repeat_Calls        380401 non-null float64
21  Total_FTF_Calls           380401 non-null float64
22  Most_Frequent_Interaction_Type 380401 non-null object
23  Total_Visits               380401 non-null float64
24  Total_Cases                 380401 non-null float64
25  Max_Case-Origin            380401 non-null object
26  Max_Case_Reason              380401 non-null object
27  Num_of_Active_Install_Bases 380401 non-null float64
28  Total_Contracts             380401 non-null float64
29  Contract_length              380401 non-null float64
30  Contract_Category            380401 non-null object
31  Num_of_Inactive_Install_Bases 380401 non-null float64
32  STRATEGIC_ACCOUNTS          380401 non-null int64
33  TERRITORY_REGION             380401 non-null object
34  TRX_YEAR                   380401 non-null int64
dtypes: float64(13), int64(7), object(15)
memory usage: 104.5+ MB

```

```
[5]: # set date columns to datetime
df['TRX_DATE'] = pd.to_datetime(df['TRX_DATE'])

# set trx year to str
df['TRX_YEAR'] = df['TRX_YEAR'].astype(str)
```

```
[6]: df.head()
```

```
[6]:   Site_Level_Price_Index CUSTOMER_ID CUSTOMER_SITE_ID      SHORT_VERTICAL \
0                  0.80     117841      609636  FRUIT & VEGETABLE
1                  0.91     113032      578406      AERO/AUTO
2                  0.80     117841      609636  FRUIT & VEGETABLE
3                  0.91     113032      578406      AERO/AUTO
4                  0.80     117841      609636  FRUIT & VEGETABLE
```

	POSTAL_CODE	CUSTOMER_CLASS	TERRITORY_TYPE	SUPPLIES_SEGMENTATION	\
0	97301	END USER	Industrial		S
1	32539	END USER	Industrial		S
2	97301	END USER	Industrial		S
3	32539	END USER	Industrial		S
4	97301	END USER	Industrial		S
	SUPPLIES_DECLINE_REASON	DUNS_NUMBER	TRX_DATE	TRX_AMT_USD	Margin \
0	Over Stocked / Timing	78842640	2016-02-05	207.72	188.59
1	None	43202248	2016-12-16	214.79	198.94
2	Over Stocked / Timing	78842640	2016-08-26	207.72	188.59
3	None	43202248	2016-10-19	429.58	397.89
4	Over Stocked / Timing	78842640	2016-08-19	623.16	565.77
	SALES_CHANNEL	QUANTITY	ORDER_TYPE	ORDER_NUM	PRODUCT_FAMILY \
0	Online	1	STANDARD DOMESTIC	511,759,211.00	CIJ
1	Esker	1	STANDARD DOMESTIC	511,868,043.00	CIJ
2	Online	1	STANDARD DOMESTIC	511,823,154.00	CIJ
3	Esker	2	STANDARD DOMESTIC	511,849,315.00	CIJ
4	Online	3	STANDARD DOMESTIC	511,823,154.00	CIJ
	PRODUCT_MODEL	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\
0	MAKE-UP	11.00	3.00	8.00	
1	VALUE PACK	21.00	5.00	16.00	
2	MAKE-UP	11.00	3.00	8.00	
3	VALUE PACK	21.00	5.00	16.00	
4	MAKE-UP	11.00	3.00	8.00	
	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	Max_Case_Origin	\
0	Call	18.00	1.97	unknown	
1	Email	58.00	1.00	Email - VTI NACC	
2	Call	18.00	1.97	unknown	
3	Email	58.00	1.00	Email - VTI NACC	
4	Call	18.00	1.97	unknown	
	Max_Case_Reason	Num_of_Active_Install_Bases	Total_Contracts	\	
0	unknown	6.00	0.00		
1	Customer Experience	4.00	4.00		
2	unknown	6.00	0.00		
3	Customer Experience	4.00	4.00		
4	unknown	6.00	0.00		
	Contract_length	Contract_Category	Num_of_Inactive_Install_Bases	\	
0	0.00	No Contract	0.00		
1	1,011.25	FSMA	0.00		
2	0.00	No Contract	0.00		

3	1,011.25	FSMA	0.00
4	0.00	No Contract	0.00
	STRATEGIC_ACCOUNTS	TERRITORY_REGION	TRX_YEAR
0	0	NW	2016
1	0	SE	2016
2	0	NW	2016
3	0	SE	2016
4	0	NW	2016

2 Calculating aggregation variables

2.1 Recency, Frequency, Tenure, Churn

```
[7]: print('The earliest transaction date: ', min(df.TRX_DATE))
print('The latest transaction date: ', max(df.TRX_DATE))
```

The earliest transaction date: 2015-01-02 00:00:00

The latest transaction date: 2020-06-03 00:00:00

```
[8]: # set now to max transaction date + 1
from datetime import timedelta
NOW = max(df.TRX_DATE) + timedelta(days=1)
```

```
[9]: # calculate recency
rec = df.groupby('CUSTOMER_SITE_ID').agg({'TRX_DATE': lambda x: (NOW - x.max()).
    .days}).reset_index()
rec['TRX_DATE'] = rec['TRX_DATE'].astype(int)
rec.rename(columns = {'TRX_DATE': 'Recency'}, inplace = True)
rec.head(10)
```

```
[9]: CUSTOMER_SITE_ID  Recency
0                 24      24
1                 90      50
2                111     415
3                114       6
4                126      49
5                141      80
6                158      78
7                234     447
8                247      17
9                287      64
```

```
[10]: # update variables table
variables = rec
```

```
[11]: # calculate frequency
freq = df.groupby('CUSTOMER_SITE_ID').agg({'TRX_DATE': lambda x: (x.max() - x.
    ↪min()).days + 1,
                                             'ORDER_NUM': lambda x: len(x)}).
    ↪reset_index()
freq['TRX_DATE'] = freq['TRX_DATE'].astype(int)
freq.rename(columns = {'TRX_DATE': 'days',
                      'ORDER_NUM': 'order_counts'}, inplace = True)
freq['Frequency'] = freq['days'] / (freq['order_counts']-1)
freq.head(10)
```

	CUSTOMER_SITE_ID	days	order_counts	Frequency
0	24	53	4.00	17.67
1	90	1829	98.00	18.86
2	111	1457	21.00	72.85
3	114	1968	174.00	11.38
4	126	1115	19.00	61.94
5	141	1883	92.00	20.69
6	158	1896	65.00	29.62
7	234	1522	176.00	8.70
8	247	1869	54.00	35.26
9	287	1882	43.00	44.81

```
[12]: (freq['Frequency'] == np.inf).sum()
```

```
[12]: 0
```

```
[13]: freq['Frequency'] = freq['Frequency'].replace(np.inf, 0)
```

```
[14]: # update variables table
variables['Frequency'] = freq['Frequency']
variables['Tenure'] = freq['days']
variables.head()
```

	CUSTOMER_SITE_ID	Recency	Frequency	Tenure
0	24	24	17.67	53
1	90	50	18.86	1829
2	111	415	72.85	1457
3	114	6	11.38	1968
4	126	49	61.94	1115

```
[15]: (variables['Frequency'] == np.inf).sum()
```

```
[15]: 0
```

```
[16]: variables['Churned_365'] = (variables['Recency'] > 365).replace({False: 0, True:
    ↪ 1})
```

```
variables.head()
```

```
[16]:   CUSTOMER_SITE_ID  Recency  Frequency  Tenure  Churned_365
0                 24        24     17.67      53          0
1                 90        50     18.86    1829          0
2                111       415     72.85    1457          1
3                114        6     11.38    1968          0
4                126       49     61.94   1115          0
```

```
[17]: variables['Churned_365'].sum()/len(variables)
```

```
[17]: 0.2604278713995458
```

```
[18]: variables.head()
```

```
[18]:   CUSTOMER_SITE_ID  Recency  Frequency  Tenure  Churned_365
0                 24        24     17.67      53          0
1                 90        50     18.86    1829          0
2                111       415     72.85    1457          1
3                114        6     11.38    1968          0
4                126       49     61.94   1115          0
```

2.2 Transaction variables

```
[19]: # make a copy of useful variables from the original data
df1 = df[['CUSTOMER_SITE_ID', 'TRX_DATE', 'Margin',
           'TRX_AMT_USD', 'QUANTITY', 'PRODUCT_FAMILY', 'PRODUCT_MODEL']].
      ↪sort_values(['CUSTOMER_SITE_ID', 'TRX_DATE'])

#df1 = df1.set_index('TRX_DATE')
df1.head()
```

```
[19]:   CUSTOMER_SITE_ID  TRX_DATE  Margin  TRX_AMT_USD  QUANTITY \
540298                  24 2020-03-20  1,381.20     1,855.74      6
81899                   24 2020-04-15   537.02      552.68      2
81898                   24 2020-05-04  4,843.37     5,000.00     20
81900                   24 2020-05-11  4,843.37     5,000.00     20
68989                  90 2015-04-14   405.99      563.34      1

  PRODUCT_FAMILY  PRODUCT_MODEL
540298          LASER    FUME EXTRACTION
81899            LCM        INK
81898            LCM        INK
81900            LCM        INK
68989            CIJ        INK
```

2.2.1 Number of transactions: count

```
[20]: num_trxn_overall = df1.groupby('CUSTOMER_SITE_ID')['TRX_DATE'].count().  
      ↪reset_index()  
      num_trxn_overall.rename(columns = {'TRX_DATE': 'Num_of_Trxns'}, inplace = True)
```

```
[21]: # update variables table
```

```
variables = pd.merge(variables, num_trxn_overall, how = 'left', on =  
      ↪'CUSTOMER_SITE_ID').fillna(0)
```

```
[22]: variables.head()
```

```
[22]:   CUSTOMER_SITE_ID  Recency  Frequency  Tenure  Churned_365  Num_of_Trxns  
0           24          24       17.67     53            0             4  
1           90          50       18.86    1829            0            98  
2          111         415       72.85   1457            1            21  
3          114          6       11.38   1968            0            174  
4          126         49       61.94  1115            0            19
```

2.2.2 TRX_AMT_USD: average

```
[23]: # calculate average transaction amount
```

```
amt_avg_overall = df1.groupby('CUSTOMER_SITE_ID').agg({'TRX_AMT_USD': lambda x:  
      ↪x.mean()}).reset_index()  
amt_avg_overall.rename(columns = {'TRX_AMT_USD': 'Avg_Trxn_Amt'}, inplace =  
      ↪True)
```

```
[24]: # update variables table
```

```
variables = pd.merge(variables, amt_avg_overall, how = 'left', on =  
      ↪'CUSTOMER_SITE_ID').fillna(0)  
variables.head()
```

```
[24]:   CUSTOMER_SITE_ID  Recency  Frequency  Tenure  Churned_365  Num_of_Trxns  \  
0           24          24       17.67     53            0             4  
1           90          50       18.86    1829            0            98  
2          111         415       72.85   1457            1            21  
3          114          6       11.38   1968            0            174  
4          126         49       61.94  1115            0            19  
  
          Avg_Trxn_Amt  
0      3,102.11  
1      233.90  
2      220.75  
3      145.43
```

```
4           835.78
```

2.2.3 Margin: average

```
[25]: # calculate average margin amount

margin_avg_overall = df1.groupby('CUSTOMER_SITE_ID').agg({'Margin': lambda x: x.
    ↴mean()}).reset_index()
margin_avg_overall.rename(columns = {'Margin': 'Avg_Margin'}, inplace = True)
```

```
[26]: # update variables table
```

```
variables = pd.merge(variables, margin_avg_overall, how = 'left', on =
    ↴'CUSTOMER_SITE_ID').fillna(0)
variables.head()
```

```
[26]:   CUSTOMER_SITE_ID  Recency  Frequency  Tenure  Churned_365  Num_of_Trxns  \
0                 24        24     17.67      53          0             4
1                 90        50     18.86    1829          0            98
2                111       415     72.85    1457          1            21
3                114        6     11.38    1968          0           174
4                126       49     61.94   1115          0            19

      Avg_Trxn_Amt  Avg_Margin
0      3,102.11    2,901.24
1      233.90      172.39
2      220.75      188.43
3      145.43      103.28
4      835.78      605.25
```

2.2.4 QUANTITY: average

```
[27]: # calculate average quantity

qt_avg_overall = df1.groupby('CUSTOMER_SITE_ID').agg({'QUANTITY': lambda x: x.
    ↴mean()}).reset_index()
qt_avg_overall.rename(columns = {'QUANTITY': 'Avg_Quantity'}, inplace = True)
```

```
[28]: # update variables table
```

```
variables = pd.merge(variables, qt_avg_overall, how = 'left', on =
    ↴'CUSTOMER_SITE_ID').fillna(0)
variables.head()
```

```
[28]:   CUSTOMER_SITE_ID  Recency  Frequency  Tenure  Churned_365  Num_of_Trxns  \
0                 24        24     17.67      53          0             4
1                 90        50     18.86    1829          0            98
```

2	111	415	72.85	1457	1	21
3	114	6	11.38	1968	0	174
4	126	49	61.94	1115	0	19

	Avg_Trxn_Amt	Avg_Margin	Avg_Quantity
0	3,102.11	2,901.24	12.00
1	233.90	172.39	11.37
2	220.75	188.43	1.05
3	145.43	103.28	4.43
4	835.78	605.25	8.84

2.2.5 PRODUCT_FAMILY / PRODUCT_MODEL: mode, count

```
[29]: # calculate number of product family /product model

prod_mode_overall = df1.groupby('CUSTOMER_SITE_ID').agg({'PRODUCT_FAMILY': lambda x: x.mode()[0],
                                                               'PRODUCT_MODEL': lambda x: x.mode()[0]}).reset_index()
prod_mode_overall.rename(columns = {'PRODUCT_FAMILY': 'Mode_of_Product_Family',
                                    'PRODUCT_MODEL': 'Mode_of_Product_Model'}, inplace = True)
```

```
[30]: # update variables table
```

```
variables = pd.merge(variables, prod_mode_overall, how = 'left', on = 'CUSTOMER_SITE_ID')
variables.head()
```

0	CUSTOMER_SITE_ID	Recency	Frequency	Tenure	Churned_365	Num_of_Trxns	\
1	24	24	17.67	53	0	4	
2	90	50	18.86	1829	0	98	
3	111	415	72.85	1457	1	21	
4	114	6	11.38	1968	0	174	
5	126	49	61.94	1115	0	19	
6	Avg_Trxn_Amt	Avg_Margin	Avg_Quantity	Mode_of_Product_Family	LCM		\
7	3,102.11	2,901.24	12.00	CIJ			
8	233.90	172.39	11.37	CIJ			
9	220.75	188.43	1.05	CIJ			
10	145.43	103.28	4.43	CIJ			
11	835.78	605.25	8.84	TIJ			
12	Mode_of_Product_Model						\
13	0	INK					
14	1	MAKE-UP					

```
2          MAKE-UP
3          MAKE-UP
4          INK
```

```
[31]: # calculate number of product family /product model

prod_count_overall = df1.groupby('CUSTOMER_SITE_ID').agg({'PRODUCT_FAMILY': □
    ↳lambda x: x.nunique(), □
                           'PRODUCT_MODEL': □
    ↳lambda x: x.nunique()}).reset_index()
prod_count_overall.rename(columns = {'PRODUCT_FAMILY': □
    ↳'Types_of_Product_Family', □
                           'PRODUCT_MODEL': 'Types_of_Product_Model'}, □
    ↳inplace = True)
```

```
[32]: # update variables table
```

```
variables = pd.merge(variables, prod_count_overall, how = 'left', on = □
    ↳'CUSTOMER_SITE_ID').fillna(0)
variables.head()
```

```
[32]: CUSTOMER_SITE_ID  Recency  Frequency  Tenure  Churned_365  Num_of_Trxns  \
0           24          24       17.67      53          0             4
1           90          50       18.86     1829          0            98
2          111         415       72.85     1457          1            21
3          114           6       11.38     1968          0            174
4          126          49       61.94     1115          0            19

   Avg_Trxn_Amt  Avg_Margin  Avg_Quantity Mode_of_Product_Family  \
0      3,102.11    2,901.24        12.00              LCM
1       233.90     172.39        11.37              CIJ
2       220.75     188.43        1.05              CIJ
3       145.43     103.28        4.43              CIJ
4       835.78     605.25        8.84              TIJ

Mode_of_Product_Model  Types_of_Product_Family  Types_of_Product_Model
0                  INK                      2                      2
1          MAKE-UP                      1                      3
2          MAKE-UP                      1                      2
3          MAKE-UP                      3                      4
4                  INK                      1                      1
```

2.3 Other variables

2.3.1 SALES_CHANNEL: mode

```
[33]: # calculate most frequent sales channel
sales = df.groupby('CUSTOMER_SITE_ID').agg({'SALES_CHANNEL': lambda x: x.
    ↴mode()[0]}).reset_index()
sales.rename(columns = {'SALES_CHANNEL': 'Most_Frequent_Sales_Channel'}, ↴
    ↴inplace = True)
sales.head()
```

```
[33]:   CUSTOMER_SITE_ID Most_Frequent_Sales_Channel
0           24                  Copy
1           90                  EDI
2          111                  EDI
3          114                  EDI
4          126                  EDI
```

```
[34]: # update variables table
variables['Most_Frequent_Sales_Channel'] = sales['Most_Frequent_Sales_Channel']
variables.head()
```

```
[34]:   CUSTOMER_SITE_ID  Recency  Frequency  Tenure  Churned_365  Num_of_Trxns \
0           24          24      17.67      53          0              4
1           90          50      18.86     1829          0             98
2          111         415      72.85     1457          1             21
3          114          6      11.38     1968          0            174
4          126         49      61.94     1115          0             19

  Avg_Trxn_Amt  Avg_Margin  Avg_Quantity Mode_of_Product_Family \
0      3,102.11    2,901.24        12.00                 LCM
1       233.90     172.39        11.37                 CIJ
2       220.75     188.43        1.05                 CIJ
3       145.43     103.28        4.43                 CIJ
4       835.78     605.25        8.84                 TIJ

  Mode_of_Product_Model  Types_of_Product_Family  Types_of_Product_Model \
0                   INK                      2                      2
1          MAKE-UP                      1                      3
2          MAKE-UP                      1                      2
3          MAKE-UP                      3                      4
4                   INK                      1                      1

  Most_Frequent_Sales_Channel
0                  Copy
1                  EDI
2                  EDI
3                  EDI
```

2.3.2 ORDER_TYPE: mode

```
[35]: # calculate most frequent order type
ot = df.groupby('CUSTOMER_SITE_ID').agg({'ORDER_TYPE': lambda x: x.mode()[0]}).
      ↵reset_index()
ot.rename(columns = {'ORDER_TYPE': 'Most_Frequent_Order_Type'}, inplace = True)
ot.head()
```

```
[35]:   CUSTOMER_SITE_ID Most_Frequent_Order_Type
0                 24           STANDARD DOMESTIC
1                 90                  EDI
2                111                  EDI
3                114                  EDI
4                126                  EDI
```

```
[36]: # update variables table
variables['Most_Frequent_Order_Type'] = ot['Most_Frequent_Order_Type']
variables.head()
```

```
[36]:   CUSTOMER_SITE_ID  Recency  Frequency  Tenure Churned_365  Num_of_Trxns \
0                 24        24     17.67      53            0             4
1                 90        50     18.86    1829            0            98
2                111       415     72.85    1457            1            21
3                114        6     11.38    1968            0           174
4                126       49     61.94   1115            0            19

      Avg_Trxn_Amt  Avg_Margin  Avg_Quantity Mode_of_Product_Family \
0      3,102.11    2,901.24      12.00                  LCM
1      233.90      172.39      11.37                  CIJ
2      220.75      188.43      1.05                  CIJ
3      145.43      103.28      4.43                  CIJ
4      835.78      605.25      8.84                  TIJ

      Mode_of_Product_Model  Types_of_Product_Family  Types_of_Product_Model \
0                      INK                         2                         2
1          MAKE-UP                         1                         3
2          MAKE-UP                         1                         2
3          MAKE-UP                         3                         4
4                      INK                         1                         1

  Most_Frequent_Sales_Channel Most_Frequent_Order_Type
0                   Copy           STANDARD DOMESTIC
1                   EDI                  EDI
2                   EDI                  EDI
3                   EDI                  EDI
```

4

EDI

EDI

2.3.3 PRICE_IDX: average

```
[37]: # calculate average price index
#pi_avg = df.groupby(['CUSTOMER_SITE_ID', 'TRX_YEAR']).agg({'PRICE_IDX': lambda x: x.mean()}).reset_index()
pi_avg = df.groupby('CUSTOMER_SITE_ID').agg({'Site_Level_Price_Index': lambda x: x.mean()}).reset_index()
pi_avg.rename(columns = {'Site_Level_Price_Index': 'Avg_Price_Index'}, inplace=True)
pi_avg.head()
```

```
[37]: CUSTOMER_SITE_ID  Avg_Price_Index
0                 24      0.79
1                 90      1.35
2                111      1.28
3                114      1.25
4                126      0.75
```

```
[38]: # update variables table
variables['Avg_Price_Index'] = pi_avg['Avg_Price_Index']
variables.head()
```

```
[38]: CUSTOMER_SITE_ID  Recency  Frequency  Tenure  Churned_365  Num_of_Trxns \
0                 24        24     17.67      53          0            4
1                 90        50     18.86    1829          0           98
2                111       415     72.85    1457          1            21
3                114         6     11.38    1968          0           174
4                126       49     61.94   1115          0            19

    Avg_Trxn_Amt  Avg_Margin  Avg_Quantity Mode_of_Product_Family \
0      3,102.11    2,901.24        12.00             LCM
1      233.90      172.39        11.37             CIJ
2      220.75      188.43        1.05             CIJ
3      145.43      103.28        4.43             CIJ
4      835.78      605.25        8.84             TIJ

    Mode_of_Product_Model  Types_of_Product_Family  Types_of_Product_Model \
0                  INK                      2                  2
1          MAKE-UP                      1                  3
2          MAKE-UP                      1                  2
3          MAKE-UP                      3                  4
4                  INK                      1                  1

Most_Frequent_Sales_Channel  Most_Frequent_Order_Type  Avg_Price_Index
```

0		Copy	STANDARD DOMESTIC	0.79
1		EDI	EDI	1.35
2		EDI	EDI	1.28
3		EDI	EDI	1.25
4		EDI	EDI	0.75

[39]: variables.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8367 entries, 0 to 8366
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   CUSTOMER_SITE_ID    8367 non-null   int64  
 1   Recency             8367 non-null   int64  
 2   Frequency           8367 non-null   float64 
 3   Tenure              8367 non-null   int64  
 4   Churned_365         8367 non-null   int64  
 5   Num_of_Trxns        8367 non-null   int64  
 6   Avg_Trxn_Amt        8367 non-null   float64 
 7   Avg_Margin          8367 non-null   float64 
 8   Avg_Quantity         8367 non-null   float64 
 9   Mode_of_Product_Family 8367 non-null   object  
 10  Mode_of_Product_Model 8367 non-null   object  
 11  Types_of_Product_Family 8367 non-null   int64  
 12  Types_of_Product_Model 8367 non-null   int64  
 13  Most_Frequent_Sales_Channel 8367 non-null   object  
 14  Most_Frequent_Order_Type   8367 non-null   object  
 15  Avg_Price_Index       8367 non-null   float64 
dtypes: float64(5), int64(7), object(4)
memory usage: 1.1+ MB
```

3 Joining calculated variables with CUSTOMER_SITE_ID unique variables

[40]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 380401 entries, 0 to 544593
Data columns (total 35 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   Site_Level_Price_Index 380401 non-null   float64 
 1   CUSTOMER_ID            380401 non-null   int64  
 2   CUSTOMER_SITE_ID       380401 non-null   int64  
 3   SHORT_VERTICAL          380401 non-null   object 
```

```

4  POSTAL_CODE           380401 non-null int64
5  CUSTOMER_CLASS        380401 non-null object
6  TERRITORY_TYPE        380401 non-null object
7  SUPPLIES_SEGMENTATION 380401 non-null object
8  SUPPLIES_DECLINE_REASONS 380401 non-null object
9  DUNS_NUMBER            380401 non-null int64
10 TRX_DATE               380401 non-null datetime64[ns]
11 TRX_AMT_USD            380401 non-null float64
12 Margin                 380401 non-null float64
13 SALES_CHANNEL          380401 non-null object
14 QUANTITY                380401 non-null int64
15 ORDER_TYPE              380401 non-null object
16 ORDER_NUM                380401 non-null float64
17 PRODUCT_FAMILY           380401 non-null object
18 PRODUCT_MODEL            380401 non-null object
19 Total_SVC_Incidents      380401 non-null float64
20 Total_Repeat_Calls       380401 non-null float64
21 Total_FTF_Calls          380401 non-null float64
22 Most_Frequent_Interaction_Type 380401 non-null object
23 Total_Visits             380401 non-null float64
24 Total_Cases              380401 non-null float64
25 Max_Case-Origin          380401 non-null object
26 Max_Case_Reason           380401 non-null object
27 Num_of_Active_Install_Bases 380401 non-null float64
28 Total_Contracts           380401 non-null float64
29 Contract_length           380401 non-null float64
30 Contract_Category         380401 non-null object
31 Num_of_Inactive_Install_Bases 380401 non-null float64
32 STRATEGIC_ACCOUNTS        380401 non-null int64
33 TERRITORY_REGION          380401 non-null object
34 TRX_YEAR                  380401 non-null object
dtypes: datetime64[ns](1), float64(13), int64(6), object(15)
memory usage: 104.5+ MB

```

```
[41]: # extract the CUSTOMER_SITE_ID unique variables from df
df_unique_vairables = df[['CUSTOMER_SITE_ID', 'SHORT_VERTICAL', 'POSTAL_CODE',
                           'CUSTOMER_CLASS', 'TERRITORY_TYPE',
                           'SUPPLIES_SEGMENTATION', ↴
                           'SUPPLIES_DECLINE_REASONS', 'DUNS_NUMBER',
                           'Total_SVC_Incidents', 'Total_Repeat_Calls',
                           'Total_FTF_Calls', ↴
                           'Most_Frequent_Interaction_Type', 'Total_Visits',
                           'Total_Cases', 'Num_of_Active_Install_Bases',
                           'Total_Contracts', 'Contract_length',
                           'Contract_Category',
                           'STRATEGIC_ACCOUNTS', 'TERRITORY_REGION'
]]
```

```
df_unique_vairables.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 380401 entries, 0 to 544593
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   CUSTOMER_SITE_ID    380401 non-null   int64  
 1   SHORT_VERTICAL      380401 non-null   object  
 2   POSTAL_CODE         380401 non-null   int64  
 3   CUSTOMER_CLASS      380401 non-null   object  
 4   TERRITORY_TYPE      380401 non-null   object  
 5   SUPPLIES_SEGMENTATION 380401 non-null   object  
 6   SUPPLIES_DECLINE_REASONS 380401 non-null   object  
 7   DUNS_NUMBER         380401 non-null   int64  
 8   Total_SVC_Incidents 380401 non-null   float64 
 9   Total_Repeat_Calls   380401 non-null   float64 
 10  Total_FTF_Calls     380401 non-null   float64 
 11  Most_Frequent_Interaction_Type 380401 non-null   object  
 12  Total_Visits        380401 non-null   float64 
 13  Total_Cases         380401 non-null   float64 
 14  Num_of_Active_Install_Bases 380401 non-null   float64 
 15  Total_Contracts     380401 non-null   float64 
 16  Contract_length     380401 non-null   float64 
 17  Contract_Category   380401 non-null   object  
 18  STRATEGIC_ACCOUNTS 380401 non-null   int64  
 19  TERRITORY_REGION    380401 non-null   object  
dtypes: float64(8), int64(4), object(8)
memory usage: 60.9+ MB
```

```
[42]: # drop duplicates to keep one row for each CUSTOMER_SITE_ID
df_unique_vairables.drop_duplicates(inplace = True)
df_unique_vairables.shape
```

```
[42]: (8367, 20)
```

```
[43]: df_unique_vairables = df_unique_vairables.sort_values(by = 'CUSTOMER_SITE_ID')
```

```
[44]: df_unique_vairables.shape
```

```
[44]: (8367, 20)
```

```
[45]: variables.shape
```

```
[45]: (8367, 16)
```

```
[46]: # combine the calculated variables and CUSTOMER_SITE_ID unique variables
```

```
all_variables = pd.merge(variables, df_unique_vairables, on = 'CUSTOMER_SITE_ID')
```

```
[47]: all_variables.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8367 entries, 0 to 8366
Data columns (total 35 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   CUSTOMER_SITE_ID    8367 non-null   int64  
 1   Recency            8367 non-null   int64  
 2   Frequency          8367 non-null   float64 
 3   Tenure             8367 non-null   int64  
 4   Churned_365        8367 non-null   int64  
 5   Num_of_Trxns       8367 non-null   int64  
 6   Avg_Trxn_Amt      8367 non-null   float64 
 7   Avg_Margin         8367 non-null   float64 
 8   Avg_Quantity       8367 non-null   float64 
 9   Mode_of_Product_Family 8367 non-null   object  
 10  Mode_of_Product_Model 8367 non-null   object  
 11  Types_of_Product_Family 8367 non-null   int64  
 12  Types_of_Product_Model 8367 non-null   int64  
 13  Most_Frequent_Sales_Channel 8367 non-null   object  
 14  Most_Frequent_Order_Type 8367 non-null   object  
 15  Avg_Price_Index    8367 non-null   float64 
 16  SHORT_VERTICAL     8367 non-null   object  
 17  POSTAL_CODE        8367 non-null   int64  
 18  CUSTOMER_CLASS     8367 non-null   object  
 19  TERRITORY_TYPE     8367 non-null   object  
 20  SUPPLIES_SEGMENTATION 8367 non-null   object  
 21  SUPPLIES_DECLINE_REASON 8367 non-null   object  
 22  DUNS_NUMBER        8367 non-null   int64  
 23  Total_SVC_Incidents 8367 non-null   float64 
 24  Total_Repeat_Calls 8367 non-null   float64 
 25  Total_FTF_Calls    8367 non-null   float64 
 26  Most_Frequent_Interaction_Type 8367 non-null   object  
 27  Total_Visits        8367 non-null   float64 
 28  Total_Cases         8367 non-null   float64 
 29  Num_of_Active_Install_Bases 8367 non-null   float64 
 30  Total_Contracts    8367 non-null   float64 
 31  Contract_length     8367 non-null   float64 
 32  Contract_Category   8367 non-null   object  
 33  STRATEGIC_ACCOUNTS 8367 non-null   int64  
 34  TERRITORY_REGION    8367 non-null   object 
```

```
dtypes: float64(13), int64(10), object(12)
memory usage: 2.3+ MB
```

```
[48]: all_variables.head()
```

```
[48]:    CUSTOMER_SITE_ID  Recency  Frequency  Tenure  Churned_365  Num_of_Trxns  \
0              24          24      17.67      53          0             4
1              90          50      18.86     1829          0            98
2             111         415      72.85     1457          1            21
3             114          6      11.38     1968          0           174
4             126         49      61.94     1115          0            19

   Avg_Trxn_Amt  Avg_Margin  Avg_Quantity Mode_of_Product_Family  \
0      3,102.11    2,901.24        12.00                LCM
1       233.90      172.39        11.37                CIJ
2       220.75      188.43        1.05                CIJ
3       145.43      103.28        4.43                CIJ
4       835.78      605.25        8.84                TIJ

   Mode_of_Product_Model  Types_of_Product_Family  Types_of_Product_Model  \
0                  INK                      2                     2
1          MAKE-UP                      1                     3
2          MAKE-UP                      1                     2
3          MAKE-UP                      3                     4
4                  INK                      1                     1

   Most_Frequent_Sales_Channel  Most_Frequent_Order_Type  Avg_Price_Index  \
0                   Copy            STANDARD DOMESTIC            0.79
1                   EDI                 EDI            1.35
2                   EDI                 EDI            1.28
3                   EDI                 EDI            1.25
4                   EDI                 EDI            0.75

   SHORT_VERTICAL  POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE  \
0      GRAPHICS      60085    END USER    Industrial
1  CHEMICALS      65802    END USER    Industrial
2      GRAPHICS      56073    END USER    Industrial
3  PHARMA & MEDICAL  92614    END USER    Industrial
4  PHARMA & MEDICAL     8822    END USER    Industrial

   SUPPLIES_SEGMENTATION  SUPPLIES_DECLINE_REASON DUNS_NUMBER  \
0                   S                  None  144782380
1                   S                  None  43937895
2                   S                  None  119130057
3                   M                  None  84160407
4                   S  Over Stocked / Timing  36781508
```

```

Total_SVC_Incidents  Total_Repeat_Calls  Total_FTF_Calls  \
0                  13.00                7.00                6.00
1                  57.00                13.00               44.00
2                  1.00                 0.00                1.00
3                  57.00                14.00               43.00
4                  1.00                 0.00                1.00

Most_Frequent_Interaction_Type  Total_Visits  Total_Cases  \
0                      Call        81.00      1.97
1                      Call        53.00      3.03
2                      Call        22.00      3.03
3                      Call        70.00      1.00
4                      Call        13.00      1.00

Num_of_Active_Install_Bases  Total_Contracts  Contract_length  \
0                      5.00          0.00          0.00
1                      6.00          6.00        1,003.00
2                      2.00          0.00          0.00
3                     15.00         11.00       521.64
4                      2.00          0.00          0.00

Contract_Category  STRATEGIC_ACCOUNTS  TERRITORY_REGION
0      No Contract            0          MW
1           FSMA              0          MC
2      No Contract            0          MW
3           FSMA              0          NW
4      No Contract            0          NE

```

[49]: all_variables.shape

[49]: (8367, 35)

[50]: all_variables.head()

```

CUSTOMER_SITE_ID  Recency  Frequency  Tenure  Churned_365  Num_of_Trxns  \
0                  24        24        17.67     53            0             4
1                  90        50        18.86    1829            0            98
2                 111       415        72.85   1457            1            21
3                 114        6        11.38   1968            0           174
4                 126       49        61.94  1115            0            19

Avg_Trxn_Amt  Avg_Margin  Avg_Quantity Mode_of_Product_Family  \
0      3,102.11    2,901.24        12.00                  LCM
1       233.90     172.39        11.37                  CIJ
2       220.75     188.43        1.05                  CIJ
3       145.43     103.28        4.43                  CIJ
4       835.78     605.25        8.84                  TIJ

```

	Mode_of_Product_Model	Types_of_Product_Family	Types_of_Product_Model	\
0	INK	2	2	
1	MAKE-UP	1	3	
2	MAKE-UP	1	2	
3	MAKE-UP	3	4	
4	INK	1	1	

	Most_Frequent_Sales_Channel	Most_Frequent_Order_Type	Avg_Price_Index	\
0	Copy	STANDARD DOMESTIC	0.79	
1	EDI	EDI	1.35	
2	EDI	EDI	1.28	
3	EDI	EDI	1.25	
4	EDI	EDI	0.75	

	SHORT_VERTICAL	POSTAL_CODE	CUSTOMER_CLASS	TERRITORY_TYPE	\
0	GRAPHICS	60085	END USER	Industrial	
1	CHEMICALS	65802	END USER	Industrial	
2	GRAPHICS	56073	END USER	Industrial	
3	PHARMA & MEDICAL	92614	END USER	Industrial	
4	PHARMA & MEDICAL	8822	END USER	Industrial	

	SUPPLIES_SEGMENTATION	SUPPLIES_DECLINE_REASON	DUNS_NUMBER	\
0	S	None	144782380	
1	S	None	43937895	
2	S	None	119130057	
3	M	None	84160407	
4	S	Over Stocked / Timing	36781508	

	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\
0	13.00	7.00	6.00	
1	57.00	13.00	44.00	
2	1.00	0.00	1.00	
3	57.00	14.00	43.00	
4	1.00	0.00	1.00	

	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	\
0	Call	81.00	1.97	
1	Call	53.00	3.03	
2	Call	22.00	3.03	
3	Call	70.00	1.00	
4	Call	13.00	1.00	

	Num_of_Active_Install_Bases	Total_Contracts	Contract_length	\
0	5.00	0.00	0.00	
1	6.00	6.00	1,003.00	
2	2.00	0.00	0.00	

3		15.00	11.00	521.64
4		2.00	0.00	0.00
	Contract_Category	STRATEGIC_ACCOUNTS	TERRITORY_REGION	
0	No Contract	0	MW	
1	FSMA	0	MC	
2	No Contract	0	MW	
3	FSMA	0	NW	
4	No Contract	0	NE	

```
[51]: # move churn to the last column
temp = all_variables['Churned_365']
all_variables.drop('Churned_365',axis = 1,inplace = True)
all_variables['Churned_365']=temp
all_variables.head()
```

	CUSTOMER_SITE_ID	Recency	Frequency	Tenure	Num_of_Trxns	Avg_Trxn_Amt	\
0	24	24	17.67	53	4	3,102.11	
1	90	50	18.86	1829	98	233.90	
2	111	415	72.85	1457	21	220.75	
3	114	6	11.38	1968	174	145.43	
4	126	49	61.94	1115	19	835.78	
	Avg_Margin	Avg_Quantity	Mode_of_Product_Family	Mode_of_Product_Model			\
0	2,901.24	12.00		LCM		INK	
1	172.39	11.37		CIJ		MAKE-UP	
2	188.43	1.05		CIJ		MAKE-UP	
3	103.28	4.43		CIJ		MAKE-UP	
4	605.25	8.84		TIJ		INK	
	Types_of_Product_Family	Types_of_Product_Model					\
0	2	2					
1	1	3					
2	1	2					
3	3	4					
4	1	1					
	Most_Frequent_Sales_Channel	Most_Frequent_Order_Type	Avg_Price_Index				\
0	Copy	STANDARD DOMESTIC	0.79				
1	EDI	EDI	1.35				
2	EDI	EDI	1.28				
3	EDI	EDI	1.25				
4	EDI	EDI	0.75				
	SHORT_VERTICAL	POSTAL_CODE	CUSTOMER_CLASS	TERRITORY_TYPE			\
0	GRAPHICS	60085	END USER	Industrial			
1	CHEMICALS	65802	END USER	Industrial			

2	GRAPHICS	56073	END USER	Industrial
3	PHARMA & MEDICAL	92614	END USER	Industrial
4	PHARMA & MEDICAL	8822	END USER	Industrial
SUPPLIES_SEGMENTATION SUPPLIES_DECLINE_REASON DUNS_NUMBER \				
0	S	None	144782380	
1	S	None	43937895	
2	S	None	119130057	
3	M	None	84160407	
4	S	Over Stocked / Timing	36781508	
Total_SVC_Incidents Total_Repeat_Calls Total_FTF_Calls \				
0	13.00	7.00	6.00	
1	57.00	13.00	44.00	
2	1.00	0.00	1.00	
3	57.00	14.00	43.00	
4	1.00	0.00	1.00	
Most_Frequent_Interaction_Type Total_Visits Total_Cases \				
0	Call	81.00	1.97	
1	Call	53.00	3.03	
2	Call	22.00	3.03	
3	Call	70.00	1.00	
4	Call	13.00	1.00	
Num_of_Active_Install_Bases Total_Contracts Contract_length \				
0	5.00	0.00	0.00	
1	6.00	6.00	1,003.00	
2	2.00	0.00	0.00	
3	15.00	11.00	521.64	
4	2.00	0.00	0.00	
Contract_Category STRATEGIC_ACCOUNTS TERRITORY_REGION Churned_365				
0	No Contract	0	MW	0
1	FSMA	0	MC	0
2	No Contract	0	MW	1
3	FSMA	0	NW	0
4	No Contract	0	NE	0

[52]: all_variables.shape

[52]: (8367, 35)

3.1 Churn definition updation

```
[53]: # If marked as churned and Frequency > 365 and supplies decline reason does not indicate a potential churn
weak_churn_reason = ['Over Stocked / Timing',
                     'Migration to 1000 Line/TIJ/TTO/LCM/LPA', 'Moved',
                     'Equipment',
                     'Production Down (timing)',
                     'Migration to Lasers',
                     'Production / Code Reduction', 'Recent Regain',
                     'Win-back',
                     'Served by Authorized Distributor',
                     'Seasonal Producer', 'Project Based',
                     'Printing/EQ downtime Issues', 'VJ Operations',
                     'Issues',
                     'Pricing / Discounting', 'Financial Distress/Credit',
                     'Hold']

all_variables['Churned_365'] = np.where(((all_variables['Churned_365'] == 1) &
                                         (all_variables['Frequency'] > 365) &
                                         (all_variables['SUPPLIES_DECLINE_REASONS'].isin(weak_churn_reason))), 0, all_variables['Churned_365'])
```

```
[54]: # Accounts marked as not churned but show churned signals

strong_churn_reason = ['Off Brand', 'Site Closed', 'AP Competitive',
                       'Displacement', 'No More Coding Requirement',]

all_variables['Churned_365'] = np.where(((all_variables['Churned_365'] == 0) &
                                         (all_variables['Contract_Category'] == "No Contract") &
                                         (all_variables['SUPPLIES_DECLINE_REASONS'].isin(strong_churn_reason)) &
                                         ((all_variables['Recency']/all_variables['Frequency']) >=
                                         2)), 1, all_variables['Churned_365'])
```

```
[55]: # update tenure based on churn
all_variables.loc[all_variables['Churned_365'] == 0, 'New Tenure'] =
    all_variables['Recency'] + all_variables['Tenure']
all_variables.loc[all_variables['Churned_365'] == 1, 'New Tenure'] =
    all_variables['Tenure']
all_variables.drop('Tenure', inplace=True, axis=1)
all_variables.rename(columns = {'New Tenure': 'Tenure'}, inplace = True)
```

4 Exporting result table

```
[56]: all_variables.head()
```

```
[56]:    CUSTOMER_SITE_ID  Recency  Frequency  Num_of_Trxns  Avg_Trxn_Amt  \
0              24          24       17.67           4      3,102.11
1              90          50       18.86          98      233.90
2             111         415       72.85          21      220.75
3             114          6       11.38          174      145.43
4             126         49       61.94          19      835.78

   Avg_Margin  Avg_Quantity Mode_of_Product_Family Mode_of_Product_Model  \
0  2,901.24        12.00            LCM                  INK
1  172.39          11.37            CIJ                MAKE-UP
2  188.43          1.05            CIJ                MAKE-UP
3  103.28          4.43            CIJ                MAKE-UP
4  605.25          8.84            TIJ                  INK

  Types_of_Product_Family  Types_of_Product_Model  \
0                      2                      2
1                      1                      3
2                      1                      2
3                      3                      4
4                      1                      1

  Most_Frequent_Sales_Channel Most_Frequent_Order_Type  Avg_Price_Index  \
0                   Copy             STANDARD DOMESTIC            0.79
1                   EDI                  EDI            1.35
2                   EDI                  EDI            1.28
3                   EDI                  EDI            1.25
4                   EDI                  EDI            0.75

  SHORT_VERTICAL  POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE  \
0      GRAPHICS      60085    END USER    Industrial
1  CHEMICALS      65802    END USER    Industrial
2      GRAPHICS      56073    END USER    Industrial
3  PHARMA & MEDICAL  92614    END USER    Industrial
4  PHARMA & MEDICAL     8822    END USER    Industrial

  SUPPLIES_SEGMENTATION SUPPLIES_DECLINE_REASON DUNS_NUMBER  \
0                     S                 None  144782380
1                     S                 None  43937895
2                     S                 None  119130057
3                     M                 None  84160407
4                     S  Over Stocked / Timing  36781508

  Total_SVC_Incidents  Total_Repeat_Calls  Total_FTF_Calls  \

```

0	13.00	7.00	6.00
1	57.00	13.00	44.00
2	1.00	0.00	1.00
3	57.00	14.00	43.00
4	1.00	0.00	1.00

	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	\
0	Call	81.00	1.97	
1	Call	53.00	3.03	
2	Call	22.00	3.03	
3	Call	70.00	1.00	
4	Call	13.00	1.00	

	Num_of_Active_Install_Bases	Total_Contracts	Contract_length	\
0	5.00	0.00	0.00	
1	6.00	6.00	1,003.00	
2	2.00	0.00	0.00	
3	15.00	11.00	521.64	
4	2.00	0.00	0.00	

	Contract_Category	STRATEGIC_ACCOUNTS	TERRITORY_REGION	Churned_365	Tenure
0	No Contract	0	MW	0	77.00
1	FSMA	0	MC	0	1,879.00
2	No Contract	0	MW	1	1,457.00
3	FSMA	0	NW	0	1,974.00
4	No Contract	0	NE	0	1,164.00

```
[57]: all_variables.to_csv('variables.csv', index = False)
```

Data Preprocessing

1. Import Dataset and relevant Python libraries

```
In [1]: 1 # import required libraries for dataframe and visualization
2
3 import numpy as np
4 import pandas as pd
5 import matplotlib.pyplot as plt
6 import datetime as dt
7 import seaborn as sns
8
9 # import required libraries for clustering
10 import sklearn
11 from sklearn.cluster import KMeans
12 from sklearn.metrics import silhouette_score
13 from sklearn.decomposition import PCA
14
15 pd.set_option('display.max_columns', None)
16 pd.set_option('display.max_rows', None)
17 pd.set_option('float_format', '{:.2f}'.format)
```

```
In [2]: 1 # import dataset
2 df = pd.read_csv('variables.csv')
3 df.head()
```

Out[2]:

	CUSTOMER_SITE_ID	Recency	Frequency	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	Avg_Qua
0	24	24	17.67	4	3,102.11	2,901.24	1
1	90	50	18.86	98	233.90	172.39	1
2	111	415	72.85	21	220.75	188.43	
3	114	6	11.38	174	145.43	103.28	
4	126	49	61.94	19	835.78	605.25	

```
In [3]: 1 df.shape
```

Out[3]: (8367, 35)

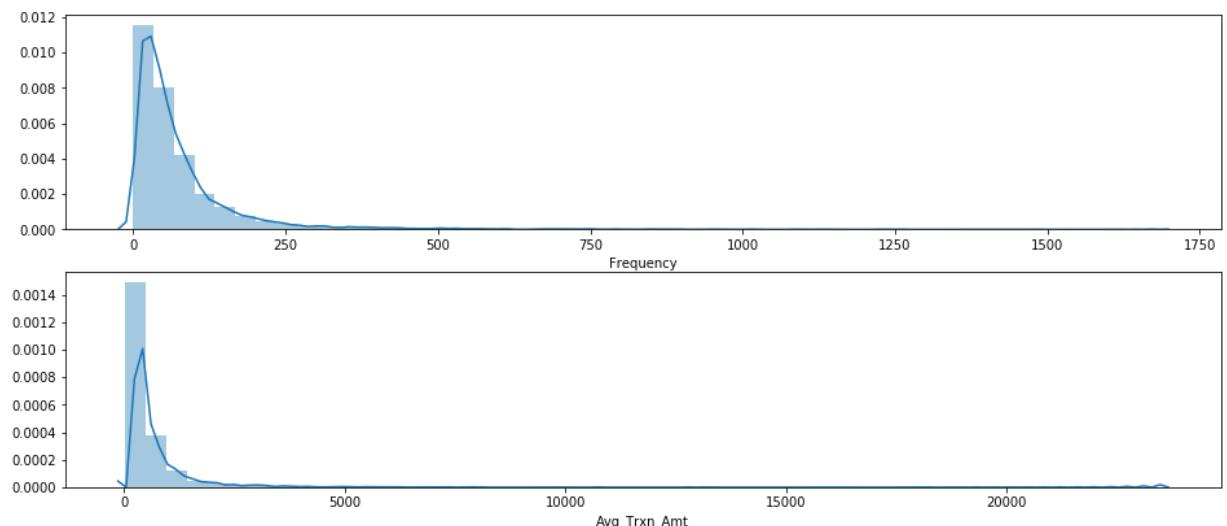
3. Extract FMP variables for clustering

```
In [4]: 1 # extract RMFP variables
2 cluster_data = df[['Frequency', 'Avg_Trxn_Amt', 'Mode_of_Product_Family']]
```

4. Check data skewness

In [5]:

```
1  ### check data skewness
2
3  # Plot RFM distributions
4  plt.figure(figsize=(16,14))
5
6  # Plot distribution of F
7  plt.subplot(4, 1, 2); sns.distplot(cluster_data['Frequency'])
8  # Plot distribution of M
9  plt.subplot(4, 1, 3); sns.distplot(cluster_data['Avg_Trxn_Amt'])
10
11 # Show the plot
12 plt.show()
```



```
In [6]: 1 # apply log transformation
2 cluster_data['Avg_Trxn_Amt'] = np.log(cluster_data['Avg_Trxn_Amt'] + 0.00000
3 cluster_data['Frequency'] = np.log(cluster_data['Frequency'] + 0.000000000000
```

/Users/shui/opt/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:2:
SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

/Users/shui/opt/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:3:
SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

This is separate from the ipykernel package so we can avoid doing imports until

5. Get dummy variables for Products field and save to the data frame

```
In [7]: 1 cluster_data = pd.get_dummies(cluster_data,columns = ['Mode_of_Product_Famil
```

```
In [8]: 1 cluster_data.head()
```

	Frequency	Avg_Trxn_Amt	Mode_of_Product_Family_BINARY ARRAY	Mode_of_Product_Family_CIJ	Mode
0	2.87	8.04	0	0	0
1	2.94	5.45	0	1	1
2	4.29	5.40	0	1	1
3	2.43	4.98	0	1	1
4	4.13	6.73	0	0	0

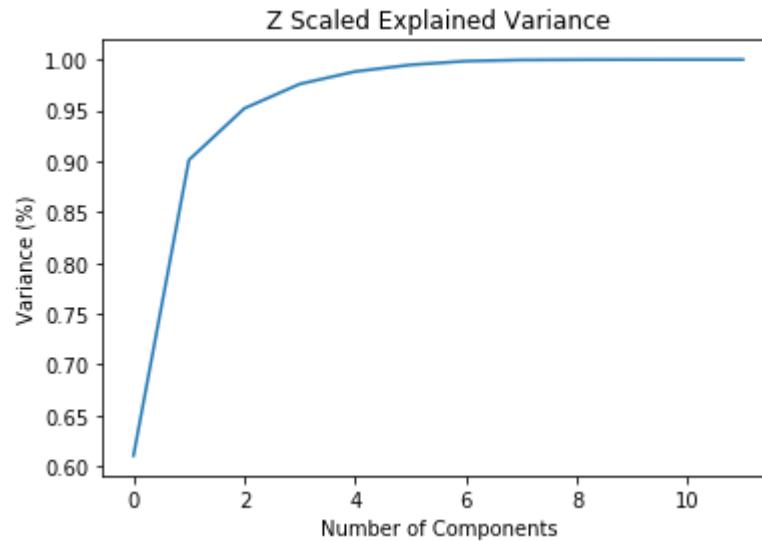
6. Conduct PCA to reduce data dimensions

In [9]:

```

1 #Fitting the PCA algorithm with our Data
2 pca = PCA().fit(cluster_data)
3
4 #Plotting the Cumulative Summation of the Explained Variance
5
6 plt.plot(np.cumsum(pca.explained_variance_ratio_))
7
8 plt.xlabel('Number of Components')
9 plt.ylabel('Variance (%)') #for each component
10 plt.title('Z Scaled Explained Variance')
11 plt.show()

```



- choose 3 components

In [10]:

```

1 components = 3
2 pca = PCA(n_components = components)
3 pca_arr = pca.fit_transform(cluster_data)

```

In [11]:

```

1 # Saving the PCA values in a dataframe
2 cols_PCA = ['PCA'+str(i) for i in range(1,components+1)]
3 pca_df = pd.DataFrame(pca_arr, columns = cols_PCA)

```

In [12]: 1 pca_df.head()

Out[12]:

	PCA1	PCA2	PCA3
0	0.58	2.43	0.46
1	0.73	-0.33	-0.10
2	-0.61	-0.48	-0.19
3	1.25	-0.76	0.06
4	-0.60	1.06	0.42

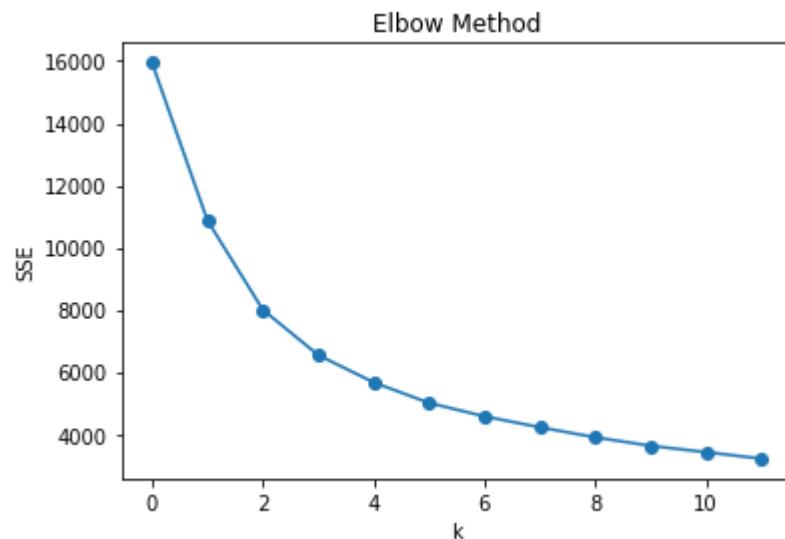
Clustering Model Fitting

7. Choose optimal number of k

In [13]:

```
1 # Elbow-curve/SSD
2
3 ssd = []
4 range_n_clusters = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
5 for num_clusters in range_n_clusters:
6     kmeans = KMeans(n_clusters=num_clusters, max_iter=50)
7     kmeans.fit(pca_df)
8
9     ssd.append(kmeans.inertia_)
10
11 # plot the SSDs for each n_clusters
12 plt.plot(ssd, marker = 'o')
13 plt.title('Elbow Method')
14 plt.ylabel('SSE')
15 plt.xlabel('k')
16 #plt.savefig('elbow_method.png')
```

Out[13]: Text(0.5, 0, 'k')



In [14]:

```

1 # Silhouette analysis
2 total_cluster = []
3 silhouette_list = []
4 range_n_clusters = [2, 3, 4, 5, 6, 7, 8]
5
6 for num_clusters in range_n_clusters:
7
8     # initialise kmeans
9     kmeans = KMeans(n_clusters=num_clusters, max_iter=50)
10    kmeans.fit(pca_df)
11
12    cluster_labels = kmeans.labels_
13
14    # silhouette score
15    silhouette_avg = silhouette_score(pca_df, cluster_labels)
16    total_cluster.append(num_clusters)
17    silhouette_list.append(silhouette_avg)
18    print("For n_clusters={0}, the silhouette score is {1}".format(num_clust

```

For n_clusters=2, the silhouette score is 0.37663706432534816
 For n_clusters=3, the silhouette score is 0.3465127565360991
 For n_clusters=4, the silhouette score is 0.368326816470208
 For n_clusters=5, the silhouette score is 0.3471132983114668
 For n_clusters=6, the silhouette score is 0.3119614984149661
 For n_clusters=7, the silhouette score is 0.318973892952047
 For n_clusters=8, the silhouette score is 0.31094783537270193

In [15]:

```

1 cluster_df = pd.DataFrame(list(zip(total_cluster, silhouette_list)))
2 cluster_df.columns=['n_clusters','silhouette_score']
3 cluster_df = cluster_df[cluster_df['n_clusters'] > 2]
4 best_cluster = cluster_df.loc[cluster_df['silhouette_score'] == cluster_df['
5 best_cluster = best_cluster.iloc[0]

```

9. Fit model

In [16]:

```

1 # k-means model fitting and setting parameter
2
3 kmeans = KMeans(n_clusters=best_cluster, max_iter=50)
4 kmeans.fit(pca_df)
5
6

```

Out[16]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=50, n_clusters=4, n_init=10, n_jobs=None, precompute_distances='auto', random_state=None, tol=0.0001, verbose=0)

10. Assign K means label to original dataset

```
In [17]: 1 # assigning cluster labels to the original dataset
          2 kmeans_labels = kmeans.labels_
          3 df['Cluster_Id'] = kmeans_labels
```

```
In [18]: 1 df.head()
```

Out[18]:

	CUSTOMER_SITE_ID	Recency	Frequency	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	Avg_Qua
0	24	24	17.67	4	3,102.11	2,901.24	1
1	90	50	18.86	98	233.90	172.39	1
2	111	415	72.85	21	220.75	188.43	
3	114	6	11.38	174	145.43	103.28	
4	126	49	61.94	19	835.78	605.25	

Clustering Results

```
In [19]: 1 # check number of customers in each cluster
          2 df['Cluster_Id'].value_counts()
```

Out[19]:

1	3634
3	2736
2	1506
0	491

Name: Cluster_Id, dtype: int64

```
In [20]: 1 # check cluster summary statistics
          2 df.groupby(['Cluster_Id']).agg({
          3     'Recency': ['mean'],
          4     'Frequency': ['mean'],
          5     'Tenure': ['mean'],
          6     'Avg_Trxn_Amt': ['mean']}).round(2)
```

Out[20]:

	Recency	Frequency	Tenure	Avg_Trxn_Amt
	mean	mean	mean	mean
Cluster_Id				
0	942.27	0.91	104.53	234.44
1	329.18	92.51	1,303.54	196.66
2	339.32	156.60	1,264.82	1,329.22
3	208.02	20.89	1,474.89	639.02

```
In [21]: 1 df.groupby('Cluster_Id') \
2   .apply(lambda x: pd.Series({
3     'Total_Monetary_Value' : (x['Avg_Trxn_Amt'] * x['Num_of_Trxns']).sum()
4     'Total_Trxn_Amt_for_each_customer' : (x['Avg_Trxn_Amt'] * x['Num_of_Trxns'])
5   }))
```

Out[21]: Total_Monetary_Value Total_Trxn_Amt_for_each_customer

Cluster_Id	Total_Monetary_Value	Total_Trxn_Amt_for_each_customer
0	5,057,829.32	10,301.08
1	14,887,143.74	4,096.63
2	49,394,695.28	32,798.60
3	205,970,111.28	75,281.47

```
In [22]: 1 df['Churned_365'].sum()/len(df)
```

Out[22]: 0.26532807457870206

```
In [23]: 1 df.groupby('Cluster_Id')['Churned_365'].mean()
```

Out[23]: Cluster_Id
0 0.72
1 0.26
2 0.29
3 0.17
Name: Churned_365, dtype: float64

```
In [24]: 1 df.head()
```

Out[24]: CUSTOMER_SITE_ID Recency Frequency Num_of_Trxns Avg_Trxn_Amt Avg_Margin Avg_Qua

0	24	24	17.67	4	3,102.11	2,901.24	1
1	90	50	18.86	98	233.90	172.39	1
2	111	415	72.85	21	220.75	188.43	
3	114	6	11.38	174	145.43	103.28	
4	126	49	61.94	19	835.78	605.25	

```
In [25]: 1 # moving churn to last
2 temp = df['Churned_365']
3 df.drop('Churned_365', axis=1, inplace=True)
4 df['Churned_365'] = temp
```

11. Exporting result to csv

```
In [26]: 1 df.to_csv('cluster_result.csv')
```

Part 6 - Churn Prediction

February 12, 2021

[1]: # relevant libraries

```
import pandas as pd
import scipy.stats
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('float_format', '{:.2f}'.format)
```

[2]: # import dataset

```
df = pd.read_csv('cluster_result.csv',index_col = 0)
df.head()
```

[2]: CUSTOMER_SITE_ID Recency Frequency Num_of_Trxns Avg_Trxn_Amt \

0	24	24	17.67	4	3,102.11
1	90	50	18.86	98	233.90
2	111	415	72.85	21	220.75
3	114	6	11.38	174	145.43
4	126	49	61.94	19	835.78

Avg_Margin Avg_Quantity Mode_of_Product_Family Mode_of_Product_Model \

0	2,901.24	12.00	LCM	INK
1	172.39	11.37	CIJ	MAKE-UP
2	188.43	1.05	CIJ	MAKE-UP
3	103.28	4.43	CIJ	MAKE-UP
4	605.25	8.84	TIJ	INK

Types_of_Product_Family Types_of_Product_Model \

0	2	2
1	1	3
2	1	2

3	3	4	
4	1	1	
0	Most_Frequent_Sales_Channel	Copy	STANDARD DOMESTIC
1		EDI	EDI
2		EDI	EDI
3		EDI	EDI
4		EDI	EDI
0	SHORT_VERTICAL	POSTAL_CODE	CUSTOMER_CLASS TERRITORY_TYPE
1	GRAPHICS	60085	END USER Industrial
2	CHEMICALS	65802	END USER Industrial
3	GRAPHICS	56073	END USER Industrial
4	PHARMA & MEDICAL	92614	END USER Industrial
4	PHARMA & MEDICAL	8822	END USER Industrial
0	SUPPLIES_SEGMENTATION	SUPPLIES_DECLINE_REASON	DUNS_NUMBER
1		S None	144782380
2		S None	43937895
3		S None	119130057
4	M	None	84160407
4	S	Over Stocked / Timing	36781508
0	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls
1		13.00 7.00	6.00
2		57.00 13.00	44.00
3		1.00 0.00	1.00
4		57.00 14.00	43.00
4		1.00 0.00	1.00
0	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases
1		Call 81.00	1.97
2		Call 53.00	3.03
3		Call 22.00	3.03
4		Call 70.00	1.00
4		Call 13.00	1.00
0	Num_of_Active_Install_Bases	Total_Contracts	Contract_length
1		5.00 0.00	0.00
2		6.00 6.00	1,003.00
3		2.00 0.00	0.00
4		15.00 11.00	521.64
4		2.00 0.00	0.00
0	Contract_Category	STRATEGIC_ACCOUNTS	TERRITORY_REGION
0	No Contract	0	MW
0			Tenure 77.00
0			Cluster_Id 3

1	FSMA	0	MC	1,879.00	3
2	No Contract	0	MW	1,457.00	1
3	FSMA	0	NW	1,974.00	3
4	No Contract	0	NE	1,164.00	2
Churned_365					
0	0				
1	0				
2	1				
3	0				
4	0				

1 Predicting churn using lifetimes package

```
[3]: # copy columns that would be used
data = df[['CUSTOMER_SITE_ID', 'Num_of_Trxns', 'Tenure', 'Recency', 'Churned_365']]
```

```
[4]: # reformat data to match the package requirement
data = data.set_index(['CUSTOMER_SITE_ID'])

data['frequency'] = data['Num_of_Trxns'] - 1
data['T'] = np.where(data['Churned_365'] == 1, data['Tenure'] + data['Recency'], data['Tenure'])
data['recency'] = np.where(data['Churned_365'] == 1, data['Tenure'], data['Tenure'] - data['Recency'])

data = data.drop(columns = ['Tenure', 'Recency', 'Num_of_Trxns'])
```

- frequency: the number of repeat purchases the customer has made.
- T: the duration between a customer's first purchase and the end of the period under study.
- recency the duration between a customer's first purchase and their latest purchase.

```
[5]: data.head()
```

```
[5]:
```

CUSTOMER_SITE_ID	Churned_365	frequency	T	recency
24	0	3	77.00	53.00
90	0	97	1,879.00	1,829.00
111	1	20	1,872.00	1,457.00
114	0	173	1,974.00	1,968.00
126	0	18	1,164.00	1,115.00

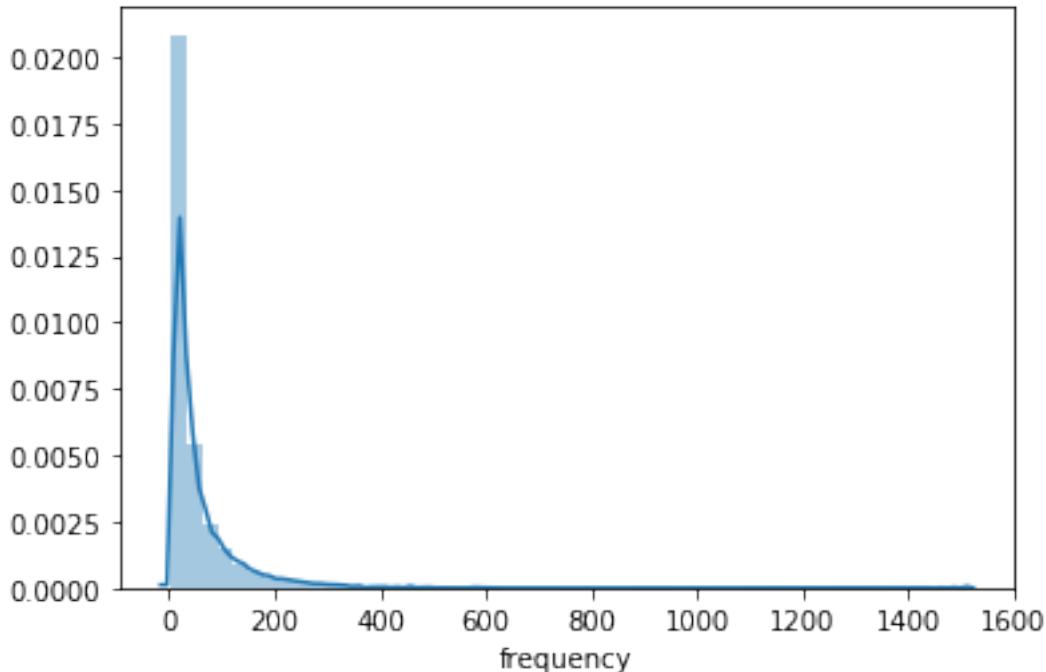
1.1 Fitting a BG-NBD model

BG-NBD(Beta Geometric Negative Binomial Distribution) model has the following assumptions:

- 1) While active, transactions made by a customer in time period t is **Poisson** distributed with mean λt
- 2) Differences in transaction rate between customers follows a **gamma** distribution with shape r and scale α
- 3) Each customer becomes inactive after each transaction with probability p
- 4) Differences in p follows a **beta** distribution with shape parameters a and b

```
[6]: sns.distplot(data.frequency)
```

```
[6]: <matplotlib.axes._subplots.AxesSubplot at 0x1a182b0208>
```



```
[7]: # fit a model
from lifetimes import ModifiedBetaGeoFitter

mbgf = ModifiedBetaGeoFitter(penalizer_coef = 0.001)
mbgf.fit(data['frequency'], data['recency'], data['T'], verbose = True)
display(mbgf)
```

```
Optimization terminated successfully.
    Current function value: -161.996609
    Iterations: 25
    Function evaluations: 28
    Gradient evaluations: 28
```

```
<lifetimes.ModifiedBetaGeoFitter: fitted with 8367 subjects, a: 0.12, alpha: 30.11, b: 3.25, r
```

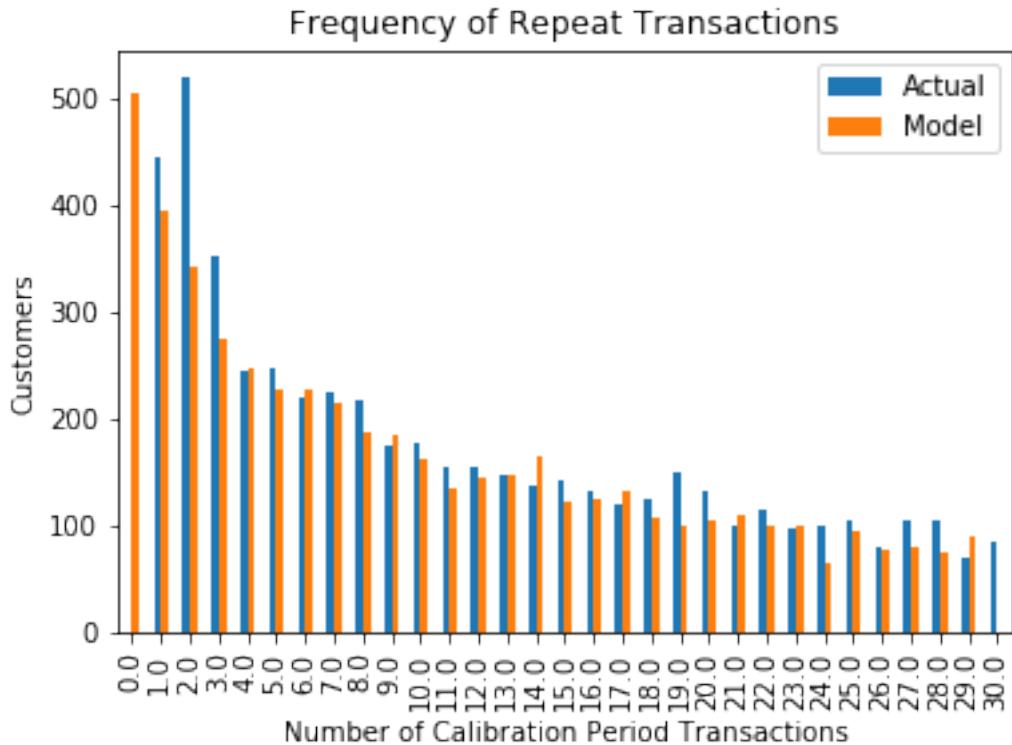
```
[8]: # fitted model parameters  
mbgf.summary
```

```
[8]:      coef  se(coef)  lower 95% bound  upper 95% bound  
r       1.06      0.02          1.03          1.09  
alpha  30.11      0.62         28.89         31.34  
a       0.12      0.00          0.12          0.13  
b       3.25      0.13          2.99          3.51
```

1.2 Assessing model fit

```
[9]: np.random.seed(1)  
  
from lifetimes.plotting import plot_period_transactions  
  
plot_period_transactions(mbgf, max_frequency = 30)  
  
# this plot compares between original data and  
# the artificial data simulated with the fitted model's parameters
```

```
[9]: <matplotlib.axes._subplots.AxesSubplot at 0x1a18aa5898>
```



```
[10]: # create simulate data
np.random.seed(1)

model = mbgf

n = model.data.shape[0]
simulated_data = model.generate_new_data(size=n)

model_counts = pd.DataFrame(model.data["frequency"].value_counts().sort_index())
simulated_counts = pd.DataFrame(simulated_data["frequency"].value_counts() .
    sort_index())
combined_counts = model_counts.merge(simulated_counts, how="outer", .
    left_index=True, right_index=True).fillna(0)
combined_counts.columns = ["Actual", "Model"]
combined_counts = combined_counts[combined_counts.Actual != 0]
```

```
[11]: # compute
diff = abs(combined_counts.Actual.iloc[:100] - combined_counts.Model.iloc[:100])
```

```
[12]: # compute MAPE MAD
MAPE = round((diff/combined_counts.Actual.iloc[:100]).mean(),2)

MAD = diff.mean()
print(f'MAPE: {MAPE}, MAD: {MAD}')
```

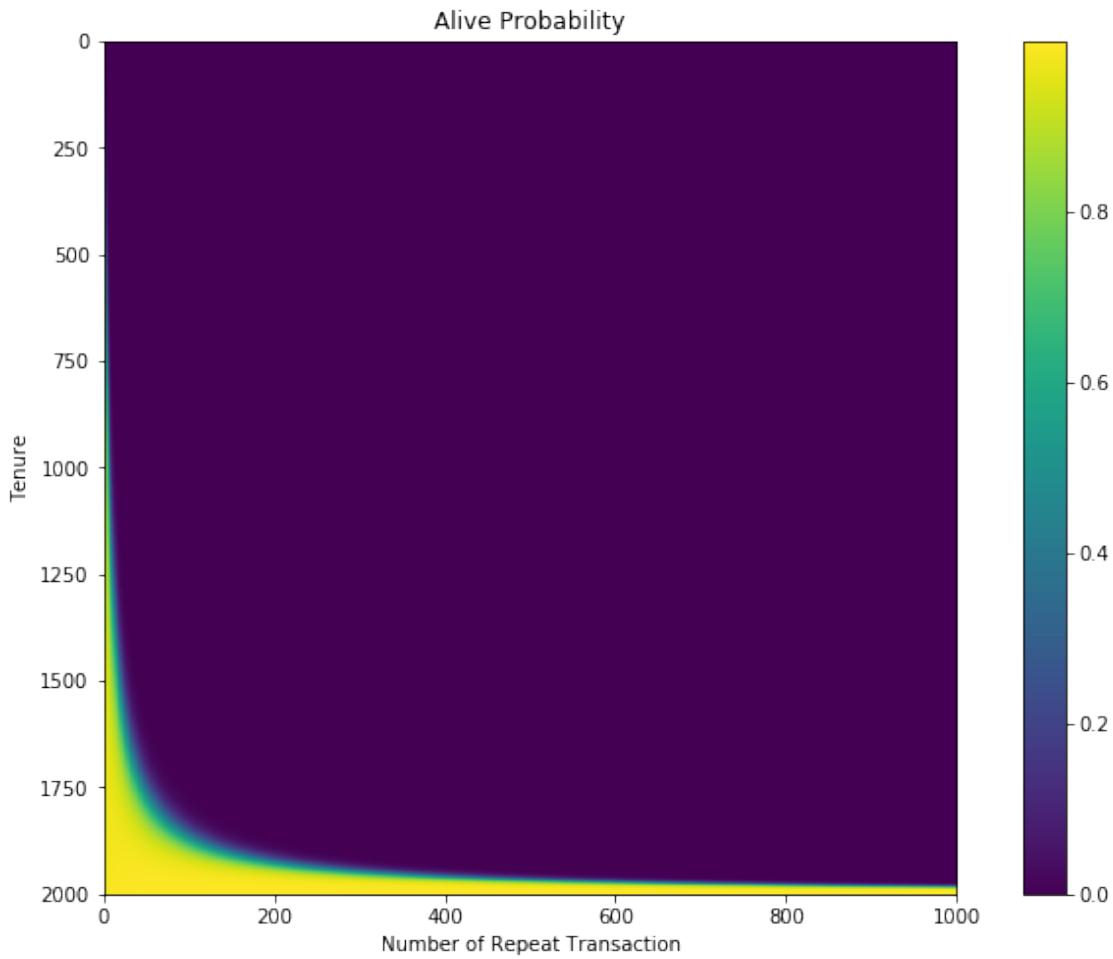
MAPE: 0.25, MAD: 11.98

1.3 Predicting alive probability

```
[13]: from lifetimes.plotting import plot_probability_alive_matrix

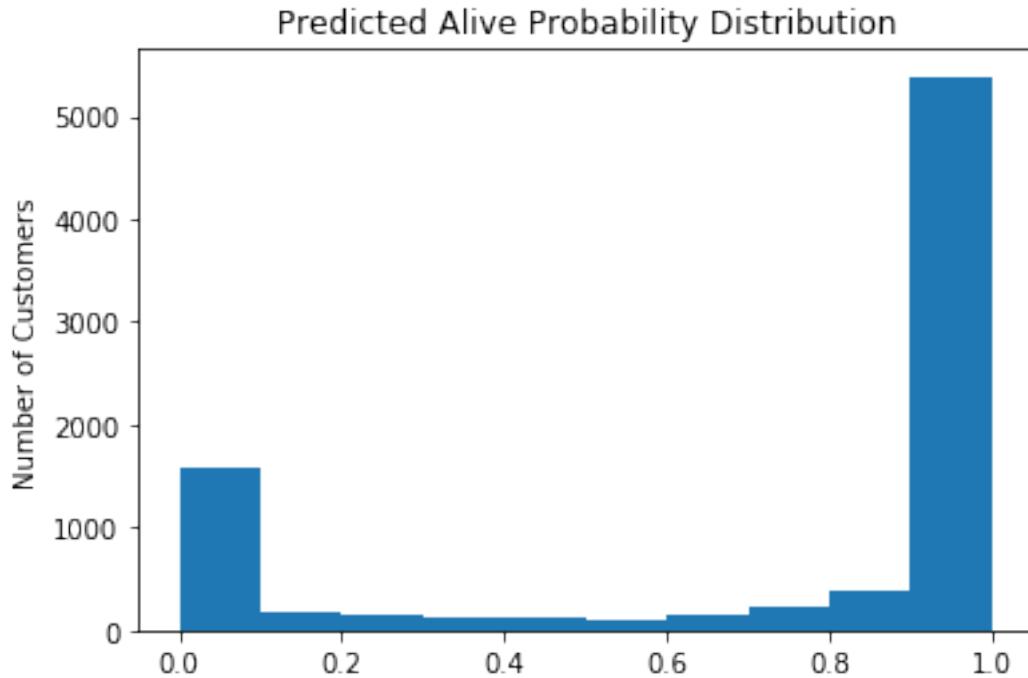
fig = plt.figure(figsize=(12,8))
plot_probability_alive_matrix(mbgf, max_recency = 2000, max_frequency = 1000,
                             title = 'Alive Probability',
                             xlabel = 'Number of Repeat Transaction',
                             ylabel = 'Tenure')
```

```
[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1056b6f60>
```



```
[14]: # distribution of alive probability
data['alive_prob'] = mbgf.conditional_probability_alive(data['frequency'], ▾
    ↪data['recency'], data['T'])
data['alive_prob'].plot.hist()
plt.title('Predicted Alive Probability Distribution')
plt.ylabel('Number of Customers')
```

```
[14]: Text(0, 0.5, 'Number of Customers')
```



```
[15]: data.head()
```

```
[15]:
```

	Churned_365	frequency	T	recency	alive_prob
CUSTOMER_SITE_ID					
24	0	3	77.00	53.00	0.95
90	0	97	1,879.00	1,829.00	0.98
111	1	20	1,872.00	1,457.00	0.51
114	0	173	1,974.00	1,968.00	1.00
126	0	18	1,164.00	1,115.00	0.99

```
[16]: data = data.reset_index()
```

```
[17]: data.head()
```

```
[17]:
```

	CUSTOMER_SITE_ID	Churned_365	frequency	T	recency	alive_prob
0	24	0	3	77.00	53.00	0.95
1	90	0	97	1,879.00	1,829.00	0.98
2	111	1	20	1,872.00	1,457.00	0.51
3	114	0	173	1,974.00	1,968.00	1.00
4	126	0	18	1,164.00	1,115.00	0.99

```
[18]: # calculate churn probability
data['churn_prob_BGNBD'] = 1-data['alive_prob']
```

1.4 Creating churn variable based on prediction

```
[19]: df = pd.concat([df, data['churn_prob_BGNBD']], axis = 1)
temp = df['Churned_365']
df.drop('Churned_365',axis = 1,inplace = True)
df['Churned_365'] = temp

df.head()
```

```
[19]:    CUSTOMER_SITE_ID  Recency  Frequency  Num_of_Trxns  Avg_Trxn_Amt \
0              24        24      17.67            4       3,102.11
1              90        50      18.86           98       233.90
2             111       415      72.85           21       220.75
3             114        6      11.38          174       145.43
4             126       49      61.94           19       835.78

   Avg_Margin  Avg_Quantity Mode_of_Product_Family Mode_of_Product_Model \
0     2,901.24         12.00                  LCM                   INK
1      172.39          11.37                 CIJ                  MAKE-UP
2     188.43           1.05                 CIJ                  MAKE-UP
3     103.28           4.43                 CIJ                  MAKE-UP
4      605.25          8.84                 TIJ                   INK

   Types_of_Product_Family  Types_of_Product_Model \
0                      2                      2
1                      1                      3
2                      1                      2
3                      3                      4
4                      1                      1

   Most_Frequent_Sales_Channel  Most_Frequent_Order_Type  Avg_Price_Index \
0                     Copy          STANDARD DOMESTIC            0.79
1                     EDI              EDI                1.35
2                     EDI              EDI                1.28
3                     EDI              EDI                1.25
4                     EDI              EDI                0.75

   SHORT_VERTICAL  POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE \
0      GRAPHICS      60085    END USER    Industrial
1  CHEMICALS      65802    END USER    Industrial
2      GRAPHICS      56073    END USER    Industrial
3  PHARMA & MEDICAL  92614    END USER    Industrial
4  PHARMA & MEDICAL     8822    END USER    Industrial

   SUPPLIES_SEGMENTATION  SUPPLIES_DECLINE_REASON DUNS_NUMBER \
0                      S                  None    144782380
1                      S                  None    43937895
```

```

2           S          None    119130057
3           M          None    84160407
4           S  Over Stocked / Timing    36781508

  Total_SVC_Incidents  Total_Repeat_Calls  Total_FTF_Calls  \
0            13.00           7.00        6.00
1            57.00          13.00       44.00
2             1.00           0.00        1.00
3            57.00          14.00       43.00
4             1.00           0.00        1.00

  Most_Frequent_Interaction_Type  Total_Visits  Total_Cases  \
0                  Call        81.00      1.97
1                  Call        53.00      3.03
2                  Call        22.00      3.03
3                  Call        70.00      1.00
4                  Call        13.00      1.00

  Num_of_Active_Install_Bases  Total_Contracts  Contract_length  \
0            5.00            0.00          0.00
1            6.00            6.00        1,003.00
2            2.00            0.00          0.00
3           15.00            11.00       521.64
4            2.00            0.00          0.00

  Contract_Category  STRATEGIC_ACCOUNTS  TERRITORY_REGION  Tenure  Cluster_Id  \
0   No Contract            0            MW    77.00        3
1      FSMA                0            MC  1,879.00        3
2   No Contract            0            MW  1,457.00        1
3      FSMA                0            NW  1,974.00        3
4   No Contract            0            NE 1,164.00        2

  churn_prob_BGNBD  Churned_365
0            0.05          0
1            0.02          0
2            0.49          1
3            0.00          0
4            0.01          0

```

```
[20]: df['Churned_365'].value_counts()
```

```
[20]: 0    6147
1    2220
Name: Churned_365, dtype: int64
```

```
[21]: df['Churned_365'].sum()/len(df['Churned_365'])
```

[21]: 0.26532807457870206

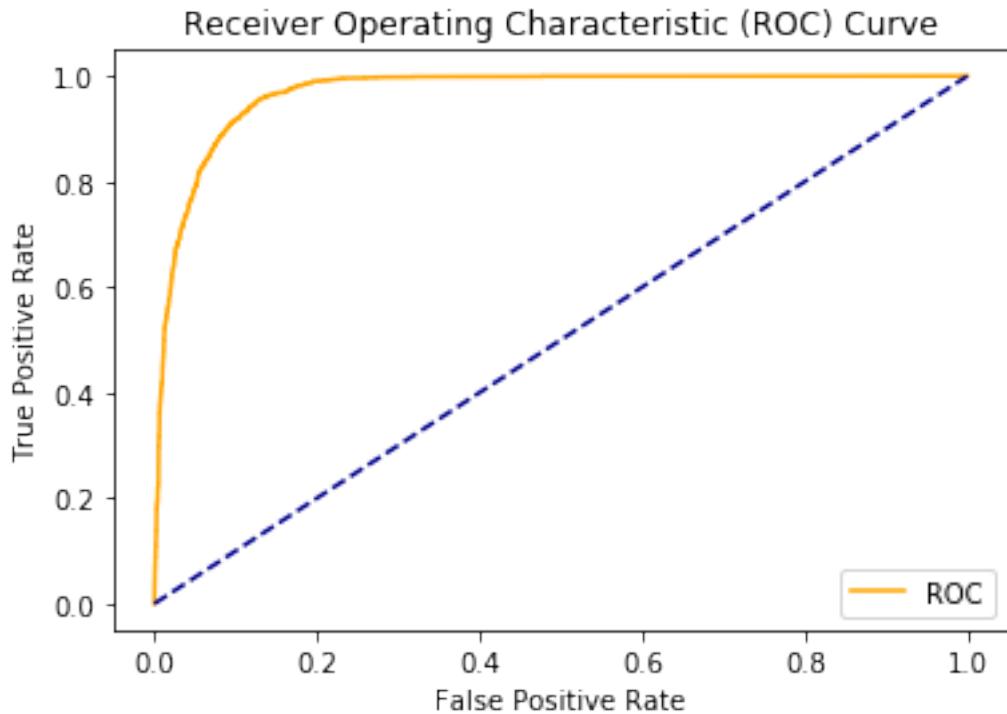
```
[22]: from sklearn.metrics import roc_curve,roc_auc_score

def plot_roc_curve(fpr, tpr):
    plt.plot(fpr, tpr, color='orange', label='ROC')
    plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver Operating Characteristic (ROC) Curve')
    plt.legend()
    plt.show()

y_true = np.array(df['Churned_365'])
y_scores = np.array(df['churn_prob_BGNBD'])

fpr, tpr, thresholds = roc_curve(y_true, y_scores)
optimal_idx = np.argmax(tpr - fpr)
optimal_threshold = thresholds[optimal_idx]
print("Threshold value is:", optimal_threshold)
plot_roc_curve(fpr, tpr)
```

Threshold value is: 0.11933220611087714



```
[23]: df['Churned_BGNBD'] = (df['churn_prob_BGNBD'] > optimal_threshold).
    ↪replace({False: 0, True: 1})
```

```
[24]: df.head()
```

```
[24]:   CUSTOMER_SITE_ID  Recency  Frequency  Num_of_Trxns  Avg_Trxn_Amt \
0              24        24      17.67            4       3,102.11
1              90        50      18.86           98       233.90
2             111       415      72.85           21       220.75
3             114         6     11.38          174       145.43
4             126       49      61.94           19       835.78

   Avg_Margin  Avg_Quantity Mode_of_Product_Family Mode_of_Product_Model \
0   2,901.24        12.00               LCM                  INK
1    172.39        11.37               CIJ                MAKE-UP
2   188.43         1.05               CIJ                MAKE-UP
3   103.28         4.43               CIJ                MAKE-UP
4    605.25        8.84               TIJ                  INK

   Types_of_Product_Family  Types_of_Product_Model \
0                      2                      2
1                      1                      3
2                      1                      2
3                      3                      4
4                      1                      1

   Most_Frequent_Sales_Channel Most_Frequent_Order_Type  Avg_Price_Index \
0                   Copy           STANDARD DOMESTIC            0.79
1                     EDI                 EDI            1.35
2                     EDI                 EDI            1.28
3                     EDI                 EDI            1.25
4                     EDI                 EDI            0.75

   SHORT_VERTICAL  POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE \
0      GRAPHICS      60085    END USER    Industrial
1  CHEMICALS      65802    END USER    Industrial
2      GRAPHICS      56073    END USER    Industrial
3  PHARMA & MEDICAL      92614    END USER    Industrial
4  PHARMA & MEDICAL      8822    END USER    Industrial

   SUPPLIES_SEGMENTATION SUPPLIES_DECLINE_REASON DUNS_NUMBER \
0                      S                    None  144782380
1                      S                    None  43937895
2                      S                    None  119130057
3                      M                    None  84160407
4          S Over Stocked / Timing  36781508
```

	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\		
0	13.00	7.00	6.00			
1	57.00	13.00	44.00			
2	1.00	0.00	1.00			
3	57.00	14.00	43.00			
4	1.00	0.00	1.00			
	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	\		
0	Call	81.00	1.97			
1	Call	53.00	3.03			
2	Call	22.00	3.03			
3	Call	70.00	1.00			
4	Call	13.00	1.00			
	Num_of_Active_Install_Bases	Total_Contracts	Contract_length	\		
0	5.00	0.00	0.00			
1	6.00	6.00	1,003.00			
2	2.00	0.00	0.00			
3	15.00	11.00	521.64			
4	2.00	0.00	0.00			
	Contract_Category	STRATEGIC_ACCOUNTS	TERRITORY_REGION	Tenure	Cluster_Id	\
0	No Contract	0	MW	77.00	3	
1	FSMA	0	MC	1,879.00	3	
2	No Contract	0	MW	1,457.00	1	
3	FSMA	0	NW	1,974.00	3	
4	No Contract	0	NE	1,164.00	2	
	churn_prob_BGNBD	Churned_365	Churned_BGNBD			
0	0.05	0	0			
1	0.02	0	0			
2	0.49	1	1			
3	0.00	0	0			
4	0.01	0	0			

```
[25]: (df['Churned_BGNBD'] == df['Churned_365']).mean()
```

```
[25]: 0.8953029759770527
```

```
[26]: df.to_csv("variables_with_churn.csv")
```

1.5 Finding top customers with highest churn probability

```
[27]: # keep only customers who have purchased in the past 1 year
recent_1yr = df[df['Recency'] < 365]
```

```
[28]: potential_churn = recent_1yr[recent_1yr['churn_prob_BGNBD'] > 0.95].
      ↪sort_values(by = ['Avg_Margin'], ascending = False)
```

```
[29]: recent_1yr.head()
```

	CUSTOMER_SITE_ID	Recency	Frequency	Num_of_Trxns	Avg_Trxn_Amt	\
0	24	24	17.67	4	3,102.11	
1	90	50	18.86	98	233.90	
3	114	6	11.38	174	145.43	
4	126	49	61.94	19	835.78	
5	141	80	20.69	92	720.52	

	Avg_Margin	Avg_Quantity	Mode_of_Product_Family	Mode_of_Product_Model	\
0	2,901.24	12.00	LCM	INK	
1	172.39	11.37	CIJ	MAKE-UP	
3	103.28	4.43	CIJ	MAKE-UP	
4	605.25	8.84	TIJ	INK	
5	556.79	38.05	CIJ	MAKE-UP	

	Types_of_Product_Family	Types_of_Product_Model	\
0	2	2	
1	1	3	
3	3	4	
4	1	1	
5	1	3	

	Most_Frequent_Sales_Channel	Most_Frequent_Order_Type	Avg_Price_Index	\
0	Copy	STANDARD DOMESTIC	0.79	
1	EDI	EDI	1.35	
3	EDI	EDI	1.25	
4	EDI	EDI	0.75	
5	Esker	STANDARD DOMESTIC	1.05	

	SHORT_VERTICAL	POSTAL_CODE	CUSTOMER_CLASS	TERRITORY_TYPE	\
0	GRAPHICS	60085	END USER	Industrial	
1	CHEMICALS	65802	END USER	Industrial	
3	PHARMA & MEDICAL	92614	END USER	Industrial	
4	PHARMA & MEDICAL	8822	END USER	Industrial	
5	OTHER	98903	END USER	Industrial	

	SUPPLIES_SEGMENTATION	SUPPLIES_DECLINE_REASON	DUNS_NUMBER	\
0	S	None	144782380	
1	S	None	43937895	
3	M	None	84160407	
4	S	Over Stocked / Timing	36781508	
5	L	Migration to 1000 Line/TIJ/TTO/LCM/LPA	832676089	

	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\		
0	13.00	7.00	6.00			
1	57.00	13.00	44.00			
3	57.00	14.00	43.00			
4	1.00	0.00	1.00			
5	124.00	31.00	93.00			
	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	\		
0	Call	81.00	1.97			
1	Call	53.00	3.03			
3	Call	70.00	1.00			
4	Call	13.00	1.00			
5	Email	50.00	1.97			
	Num_of_Active_Install_Bases	Total_Contracts	Contract_length	\		
0	5.00	0.00	0.00			
1	6.00	6.00	1,003.00			
3	15.00	11.00	521.64			
4	2.00	0.00	0.00			
5	15.00	15.00	364.00			
	Contract_Category	STRATEGIC_ACCOUNTS	TERRITORY_REGION	Tenure	Cluster_Id	\
0	No Contract	0	MW	77.00	3	
1	FSMA	0	MC	1,879.00	3	
3	FSMA	0	NW	1,974.00	3	
4	No Contract	0	NE	1,164.00	2	
5	FSMA	0	NW	1,963.00	3	
	churn_prob_BGNBD	Churned_365	Churned_BGNBD			
0	0.05	0	0			
1	0.02	0	0			
3	0.00	0	0			
4	0.01	0	0			
5	0.05	0	0			

```
[30]: potential_churn = potential_churn
      →[['CUSTOMER_SITE_ID','Recency','Frequency','Tenure','Avg_Trxn_Amt','Avg_Margin','Num_of_Trxns']]
      potential_churn['Total_Trxn_Amt'] = potential_churn['Avg_Trxn_Amt'] *_
      →potential_churn['Num_of_Trxns']
      potential_churn['Total_Margin'] = potential_churn['Avg_Margin'] *_
      →potential_churn['Num_of_Trxns']
      temp = potential_churn['Cluster_Id']
      potential_churn.drop('Cluster_Id',axis = 1,inplace = True)
      potential_churn['Cluster_Id'] = temp

      potential_churn.head()
```

```
[30]: CUSTOMER_SITE_ID Recency Frequency Tenure Avg_Trxn_Amt Avg_Margin \
8096 8386636 259 13.50 367.00 4,420.03 3,916.73
2649 113342 358 20.04 1,981.00 5,706.04 2,762.46
7563 8336697 360 22.47 742.00 2,581.83 2,313.56
1103 34151 128 8.15 1,971.00 1,882.71 1,506.53
5613 695136 251 19.85 1,978.00 1,277.68 1,204.66
```

	Num_of_Trxns	Total_Trxn_Amt	Total_Margin	Cluster_Id
8096	9	39,780.25	35,250.53	3
2649	82	467,895.58	226,521.70	2
7563	18	46,472.98	41,644.00	3
1103	227	427,374.75	341,982.00	3
5613	88	112,436.25	106,009.82	3

```
[31]: cluster_segment = pd.DataFrame(potential_churn.
    ↪groupby(['Cluster_Id'])['Total_Margin'].mean())
cluster_segment['rank(1-top_priority)'] = cluster_segment['Total_Margin'].
    ↪rank(ascending=False)
cluster_segment = cluster_segment.reset_index()
cluster_segment = cluster_segment.drop('Total_Margin',axis = 1)
cluster_segment
```

```
[31]: Cluster_Id rank(1-top_priority)
0 0 4.00
1 1 3.00
2 2 1.00
3 3 2.00
```

```
[32]: potential_churn = potential_churn.merge(cluster_segment, on='Cluster_Id', ↪
    ↪how='left')
potential_churn = potential_churn.sort_values(["rank(1-top_priority)", ↪
    ↪"Total_Margin"], ascending = (True, False))
potential_churn
```

```
[32]: CUSTOMER_SITE_ID Recency Frequency Tenure Avg_Trxn_Amt Avg_Margin \
1 113342 358 20.04 1,981.00 5,706.04 2,762.46
13 71033 289 25.62 1,665.00 3,324.54 763.45
18 89579 357 26.95 1,617.00 2,491.63 683.60
3 34151 128 8.15 1,971.00 1,882.71 1,506.53
21 36080 317 6.81 1,978.00 843.69 657.77
7 29598 273 12.07 1,963.00 1,145.61 1,063.22
8 198762 240 11.68 1,694.00 1,093.82 951.75
4 695136 251 19.85 1,978.00 1,277.68 1,204.66
51 735 161 5.83 1,974.00 381.78 331.30
53 5768 80 5.88 1,963.00 427.33 319.99
26 591188 127 10.88 1,954.00 1,151.01 592.99
17 285085 359 11.69 1,961.00 930.08 707.12
```

24	156142	161	12.45	1,978.00	789.53	640.34
57	41379	234	6.12	1,971.00	383.77	304.03
14	8380485	50	3.89	478.00	927.76	756.56
15	66176	294	15.74	1,684.00	3,792.80	752.68
27	61702	274	11.77	1,933.00	696.11	541.73
37	71036	342	11.04	1,976.00	570.58	452.62
23	965128	181	15.06	1,702.00	748.52	640.76
54	119877	107	9.73	1,975.00	442.01	317.78
42	66173	309	10.86	1,981.00	2,072.45	395.55
33	66930	178	14.56	1,954.00	521.46	478.71
31	8067	282	14.65	1,981.00	663.95	487.29
9	4951301	343	16.46	1,281.00	1,007.05	924.43
46	106774	317	12.20	1,659.00	1,735.02	373.97
30	164660	280	18.25	1,977.00	621.02	524.35
74	239131	352	7.70	1,617.00	1,020.93	213.00
80	234623	107	7.95	1,975.00	229.97	185.58
20	5028416	219	16.35	1,282.00	710.25	658.27
82	49864	147	7.89	1,830.00	236.26	180.83
65	5007459	132	7.33	1,304.00	314.88	260.18
78	61621	301	7.52	1,970.00	221.46	187.17
2	8336697	360	22.47	742.00	2,581.83	2,313.56
39	587950	196	18.27	1,950.00	631.10	424.67
38	5181	358	17.01	1,957.00	531.14	428.65
41	66732	227	18.83	1,978.00	541.18	421.35
29	122126	272	23.18	1,692.00	626.01	526.83
0	8386636	259	13.50	367.00	4,420.03	3,916.73
19	178057	237	19.54	1,253.00	797.73	662.90
62	500441	258	13.65	1,978.00	343.45	275.42
35	121513	261	22.88	1,954.00	577.20	464.39
52	802123	345	13.39	1,764.00	389.24	320.78
69	686056	251	12.95	1,960.00	275.97	239.87
10	8351627	205	14.44	667.00	1,016.62	910.02
61	625051	255	16.18	1,970.00	344.21	278.98
43	80843	359	21.84	1,616.00	1,364.28	389.55
34	190209	309	28.28	1,949.00	587.23	475.08
44	8299019	135	11.82	974.00	798.48	378.94
63	569708	233	17.65	1,927.00	322.51	274.65
58	268451	352	18.35	1,967.00	363.87	298.02
68	63817	351	15.67	1,630.00	283.14	241.52
79	9949	337	12.30	1,623.00	276.27	185.84
72	88594	338	14.11	1,876.00	1,144.59	220.91
49	69705	310	24.40	1,969.00	501.29	350.95
70	305128	252	17.76	1,705.00	527.20	239.25
56	487283	259	24.23	1,955.00	372.46	306.56
123	66316	294	6.49	1,975.00	120.14	82.01
81	5609	287	14.95	1,976.00	249.57	183.66
12	7928899	244	20.12	767.00	873.00	766.15

85	58126	217	13.48	1,969.00	187.01	157.15
103	1242869	237	8.85	1,680.00	156.05	122.32
48	91449	321	29.89	1,935.00	448.86	351.06
66	188215	273	23.62	1,974.00	293.99	257.34
121	68690	260	8.11	1,971.00	130.88	87.74
73	780212	202	19.10	1,768.00	257.81	216.89
55	11996	342	28.16	1,919.00	463.41	308.96
32	8373919	260	7.29	515.00	607.64	481.84
59	4361	315	28.77	1,955.00	401.04	294.43
50	566543	324	32.48	1,948.00	381.04	333.85
5	8375156	282	18.57	542.00	1,366.38	1,098.54
75	4604	302	21.62	1,967.00	243.07	205.53
22	553616	342	24.87	914.00	809.07	656.00
89	552009	279	18.00	1,971.00	206.17	150.47
6	8363484	332	19.83	570.00	1,254.83	1,089.66
122	13297	190	10.82	1,964.00	108.02	84.33
99	181321	178	17.30	1,977.00	156.76	124.95
67	798414	316	28.53	1,771.00	300.13	250.48
83	374322	253	25.21	1,967.00	252.47	175.06
90	67020	297	22.69	1,976.00	183.20	147.62
45	8334196	281	17.71	777.00	460.79	376.29
101	9986	352	23.03	1,941.00	211.09	124.65
25	8399007	213	15.80	371.00	763.32	629.41
125	3717904	289	12.98	1,379.00	84.77	70.80
28	475966	296	20.29	438.00	697.36	538.60
111	8353055	251	12.78	711.00	127.46	106.47
77	8338560	314	28.65	801.00	224.95	193.78
94	8359077	267	23.93	626.00	164.07	138.87
76	8332063	363	18.70	550.00	226.90	195.42
100	8342047	349	27.25	785.00	227.23	124.67
84	736080	286	4.71	319.00	187.31	163.40
87	8371489	119	4.29	149.00	451.80	155.71
96	8406486	252	10.00	312.00	145.63	129.08
110	236758	196	8.50	51.00	123.81	107.90
98	367748	359	22.41	1,928.00	148.74	125.71
91	4962	293	29.28	1,962.00	191.33	147.02
116	50950	321	26.35	1,981.00	117.34	95.69
126	33594	318	18.29	1,964.00	92.52	66.90
108	5752	325	32.74	1,340.00	133.31	110.77
127	8388585	330	14.67	374.00	76.14	62.58
16	108324	196	0.25	197.00	901.63	728.86
36	8356501	248	0.67	252.00	537.89	457.20
11	2001	253	0.50	1.00	1,087.59	894.32
40	5865094	198	0.25	199.00	446.87	423.51
60	8408327	290	0.33	291.00	338.91	291.24
47	8372377	275	0.50	276.00	441.52	373.64
71	8367934	364	0.33	365.00	251.95	230.77

64	719986	265	0.50	266.00	331.32	265.67
120	254034	139	3.80	19.00	107.02	88.46
117	756036	167	0.25	168.00	112.02	94.41
86	8327225	349	0.50	350.00	187.87	156.25
88	8398140	342	0.50	343.00	183.72	152.91
92	8411336	272	0.50	273.00	171.59	144.61
93	8408277	273	0.50	274.00	192.02	142.18
95	8383551	352	0.50	353.00	147.15	129.44
97	8383046	317	0.50	318.00	148.03	126.33
102	8400436	311	0.50	312.00	143.05	123.21
104	8400785	346	0.50	347.00	145.50	120.21
105	8406737	303	0.50	304.00	137.48	117.62
106	8360353	253	0.50	254.00	141.13	113.08
107	8405935	314	0.50	315.00	132.18	112.71
109	488874	323	0.50	324.00	123.20	109.33
112	8407463	251	0.50	252.00	112.86	100.07
128	8406452	253	0.17	254.00	73.73	42.82
113	8411153	274	0.50	275.00	112.86	98.73
114	8410073	262	0.50	263.00	114.97	98.34
115	8412601	253	0.50	254.00	113.73	97.53
118	8401711	339	0.50	340.00	106.48	92.35
119	8410802	279	0.50	280.00	117.71	92.32
124	8402083	346	2.50	351.00	92.50	77.33
129	690630	286	0.50	287.00	49.38	37.92

Num_of_Trxns	Total_Trxn_Amt	Total_Margin	Cluster_Id	\
1	82	467,895.58	226,521.70	2
13	66	219,419.96	50,387.95	2
18	61	151,989.26	41,699.77	2
3	227	427,374.75	341,982.00	3
21	245	206,705.11	161,154.53	3
7	141	161,530.36	149,913.90	3
8	146	159,697.09	138,954.94	3
4	88	112,436.25	106,009.82	3
51	312	119,114.60	103,366.09	3
53	321	137,173.23	102,715.47	3
26	169	194,519.90	100,215.71	3
17	138	128,351.00	97,583.16	3
24	147	116,060.87	94,129.42	3
57	285	109,374.87	86,648.70	3
14	111	102,981.41	83,978.46	3
15	108	409,622.75	81,289.47	3
27	142	98,847.93	76,925.87	3
37	149	85,016.84	67,440.17	3
23	102	76,349.53	65,357.24	3
54	193	85,308.63	61,330.87	3
42	155	321,229.40	61,310.04	3

33	123	64,139.27	58,881.49	3
31	117	77,682.20	57,013.10	3
9	58	58,408.89	53,616.74	3
46	137	237,697.65	51,233.61	3
30	94	58,375.50	49,288.61	3
74	211	215,416.89	44,942.58	3
80	236	54,273.08	43,797.26	3
20	66	46,876.51	43,445.78	3
82	233	55,047.44	42,133.84	3
65	161	50,696.13	41,889.60	3
78	223	49,385.70	41,738.77	3
2	18	46,472.98	41,644.00	3
39	97	61,216.75	41,193.38	3
38	95	50,458.58	40,721.91	3
41	94	50,870.69	39,606.50	3
29	74	46,324.88	38,985.76	3
0	9	39,780.25	35,250.53	3
19	53	42,279.59	35,133.70	3
62	127	43,617.67	34,978.89	3
35	75	43,290.12	34,829.15	3
52	107	41,648.26	34,323.60	3
69	133	36,703.86	31,902.45	3
10	33	33,548.47	30,030.76	3
61	107	36,830.71	29,851.25	3
43	75	102,320.72	29,215.91	3
34	59	34,646.63	28,029.90	3
44	72	57,490.68	27,283.80	3
63	97	31,283.01	26,640.67	3
58	89	32,384.85	26,523.37	3
68	105	29,730.15	25,360.00	3
79	133	36,744.08	24,716.52	3
72	110	125,904.98	24,299.74	3
49	69	34,588.68	24,215.32	3
70	97	51,138.33	23,207.19	3
56	71	26,444.53	21,766.03	3
123	260	31,236.48	21,322.66	3
81	114	28,451.04	20,937.63	3
12	27	23,571.04	20,685.98	3
85	131	24,497.93	20,586.14	3
103	164	25,592.73	20,060.25	3
48	55	24,687.26	19,308.29	3
66	73	21,461.10	18,785.48	3
121	212	27,746.66	18,601.08	3
73	83	21,398.17	18,002.02	3
55	57	26,414.47	17,610.83	3
32	36	21,874.94	17,346.13	3
59	58	23,260.33	17,077.11	3

50	51	19,433.04	17,026.19	3
5	15	20,495.73	16,478.03	3
75	78	18,959.82	16,030.96	3
22	24	19,417.61	15,743.95	3
89	95	19,586.29	14,294.84	3
6	13	16,312.77	14,165.63	3
122	165	17,823.20	13,914.06	3
99	105	16,460.05	13,119.59	3
67	52	15,606.72	13,024.73	3
83	69	17,420.47	12,079.27	3
90	75	13,740.12	11,071.70	3
45	29	13,362.92	10,912.42	3
101	70	14,775.96	8,725.35	3
25	11	8,396.56	6,923.46	3
125	85	7,205.69	6,018.06	3
28	8	5,578.86	4,308.83	3
111	37	4,716.05	3,939.54	3
77	18	4,049.04	3,488.01	3
94	16	2,625.07	2,221.96	3
76	11	2,495.88	2,149.58	3
100	17	3,862.84	2,119.35	3
84	8	1,498.50	1,307.23	3
87	8	3,614.36	1,245.70	3
96	7	1,019.40	903.54	3
110	7	866.70	755.30	3
98	71	10,560.88	8,925.25	1
91	58	11,097.21	8,526.90	1
116	64	7,509.82	6,124.36	1
126	91	8,419.12	6,087.81	1
108	32	4,266.04	3,544.61	1
127	4	304.56	250.32	1
16	5	4,508.16	3,644.32	0
36	7	3,765.23	3,200.43	0
11	3	3,262.76	2,682.97	0
40	5	2,234.36	2,117.55	0
60	4	1,355.64	1,164.97	0
47	3	1,324.56	1,120.93	0
71	4	1,007.79	923.08	0
64	3	993.96	797.00	0
120	6	642.11	530.73	0
117	5	560.08	472.04	0
86	3	563.60	468.74	0
88	3	551.16	458.73	0
92	3	514.78	433.83	0
93	3	576.06	426.54	0
95	3	441.45	388.32	0
97	3	444.08	379.00	0

102	3	429.14	369.62	0
104	3	436.51	360.64	0
105	3	412.45	352.87	0
106	3	423.38	339.25	0
107	3	396.55	338.14	0
109	3	369.60	327.99	0
112	3	338.57	300.21	0
128	7	516.14	299.71	0
113	3	338.57	296.20	0
114	3	344.92	295.03	0
115	3	341.19	292.60	0
118	3	319.43	277.06	0
119	3	353.12	276.95	0
124	3	277.50	231.99	0
129	3	148.15	113.76	0

rank(1-top_priority)

1	1.00
13	1.00
18	1.00
3	2.00
21	2.00
7	2.00
8	2.00
4	2.00
51	2.00
53	2.00
26	2.00
17	2.00
24	2.00
57	2.00
14	2.00
15	2.00
27	2.00
37	2.00
23	2.00
54	2.00
42	2.00
33	2.00
31	2.00
9	2.00
46	2.00
30	2.00
74	2.00
80	2.00
20	2.00
82	2.00

65	2.00
78	2.00
2	2.00
39	2.00
38	2.00
41	2.00
29	2.00
0	2.00
19	2.00
62	2.00
35	2.00
52	2.00
69	2.00
10	2.00
61	2.00
43	2.00
34	2.00
44	2.00
63	2.00
58	2.00
68	2.00
79	2.00
72	2.00
49	2.00
70	2.00
56	2.00
123	2.00
81	2.00
12	2.00
85	2.00
103	2.00
48	2.00
66	2.00
121	2.00
73	2.00
55	2.00
32	2.00
59	2.00
50	2.00
5	2.00
75	2.00
22	2.00
89	2.00
6	2.00
122	2.00
99	2.00
67	2.00

83	2.00
90	2.00
45	2.00
101	2.00
25	2.00
125	2.00
28	2.00
111	2.00
77	2.00
94	2.00
76	2.00
100	2.00
84	2.00
87	2.00
96	2.00
110	2.00
98	3.00
91	3.00
116	3.00
126	3.00
108	3.00
127	3.00
16	4.00
36	4.00
11	4.00
40	4.00
60	4.00
47	4.00
71	4.00
64	4.00
120	4.00
117	4.00
86	4.00
88	4.00
92	4.00
93	4.00
95	4.00
97	4.00
102	4.00
104	4.00
105	4.00
106	4.00
107	4.00
109	4.00
112	4.00
128	4.00
113	4.00

```
114          4.00
115          4.00
118          4.00
119          4.00
124          4.00
129          4.00
```

```
[33]: potential_churn.groupby("rank(1-top_priority)")['Total Margin'].mean()
```

```
[33]: rank(1-top_priority)
1.00    106,203.14
2.00     41,114.32
3.00      5,576.54
4.00      763.91
Name: Total Margin, dtype: float64
```

```
[34]: potential_churn.drop(['Avg_Trxn_Amt','Avg_Margin'],axis = 1,inplace = True)
```

```
[35]: # removing rows with 0 Frequency if any
```

```
potential_churn = potential_churn.astype(int)
potential_churn = potential_churn[potential_churn["Frequency"] > 0]
potential_churn
```

```
[35]:   CUSTOMER_SITE_ID  Recency  Frequency  Tenure  Num_of_Trxns \
1           113342       358        20    1981            82
13          71033        289        25    1665            66
18          89579        357        26    1617            61
3           34151       128         8    1971           227
21          36080       317         6    1978           245
7           29598       273        12    1963           141
8           198762      240        11    1694           146
4           695136      251        19    1978            88
51          735         161         5    1974           312
53          5768        80          5    1963           321
26          591188      127        10    1954           169
17          285085      359        11    1961           138
24          156142      161        12    1978           147
57          41379       234         6    1971           285
14          8380485      50          3    478            111
15          66176        294        15    1684            108
27          61702        274        11    1933           142
37          71036        342        11    1976           149
23          965128       181        15    1702            102
54          119877       107         9    1975            193
42          66173         309        10    1981           155
33          66930        178        14    1954            123
```

31	8067	282	14	1981	117
9	4951301	343	16	1281	58
46	106774	317	12	1659	137
30	164660	280	18	1977	94
74	239131	352	7	1617	211
80	234623	107	7	1975	236
20	5028416	219	16	1282	66
82	49864	147	7	1830	233
65	5007459	132	7	1304	161
78	61621	301	7	1970	223
2	8336697	360	22	742	18
39	587950	196	18	1950	97
38	5181	358	17	1957	95
41	66732	227	18	1978	94
29	122126	272	23	1692	74
0	8386636	259	13	367	9
19	178057	237	19	1253	53
62	500441	258	13	1978	127
35	121513	261	22	1954	75
52	802123	345	13	1764	107
69	686056	251	12	1960	133
10	8351627	205	14	667	33
61	625051	255	16	1970	107
43	80843	359	21	1616	75
34	190209	309	28	1949	59
44	8299019	135	11	974	72
63	569708	233	17	1927	97
58	268451	352	18	1967	89
68	63817	351	15	1630	105
79	9949	337	12	1623	133
72	88594	338	14	1876	110
49	69705	310	24	1969	69
70	305128	252	17	1705	97
56	487283	259	24	1955	71
123	66316	294	6	1975	260
81	5609	287	14	1976	114
12	7928899	244	20	767	27
85	58126	217	13	1969	131
103	1242869	237	8	1680	164
48	91449	321	29	1935	55
66	188215	273	23	1974	73
121	68690	260	8	1971	212
73	780212	202	19	1768	83
55	11996	342	28	1919	57
32	8373919	260	7	515	36
59	4361	315	28	1955	58
50	566543	324	32	1948	51

5	8375156	282	18	542	15
75	4604	302	21	1967	78
22	553616	342	24	914	24
89	552009	279	18	1971	95
6	8363484	332	19	570	13
122	13297	190	10	1964	165
99	181321	178	17	1977	105
67	798414	316	28	1771	52
83	374322	253	25	1967	69
90	67020	297	22	1976	75
45	8334196	281	17	777	29
101	9986	352	23	1941	70
25	8399007	213	15	371	11
125	3717904	289	12	1379	85
28	475966	296	20	438	8
111	8353055	251	12	711	37
77	8338560	314	28	801	18
94	8359077	267	23	626	16
76	8332063	363	18	550	11
100	8342047	349	27	785	17
84	736080	286	4	319	8
87	8371489	119	4	149	8
96	8406486	252	10	312	7
110	236758	196	8	51	7
98	367748	359	22	1928	71
91	4962	293	29	1962	58
116	50950	321	26	1981	64
126	33594	318	18	1964	91
108	5752	325	32	1340	32
127	8388585	330	14	374	4
120	254034	139	3	19	6
124	8402083	346	2	351	3

	Total_Trxn_Amt	Total_Margin	Cluster_Id	rank(1-top_priority)
1	467895	226521	2	1
13	219419	50387	2	1
18	151989	41699	2	1
3	427374	341982	3	2
21	206705	161154	3	2
7	161530	149913	3	2
8	159697	138954	3	2
4	112436	106009	3	2
51	119114	103366	3	2
53	137173	102715	3	2
26	194519	100215	3	2
17	128350	97583	3	2
24	116060	94129	3	2

57	109374	86648	3	2
14	102981	83978	3	2
15	409622	81289	3	2
27	98847	76925	3	2
37	85016	67440	3	2
23	76349	65357	3	2
54	85308	61330	3	2
42	321229	61310	3	2
33	64139	58881	3	2
31	77682	57013	3	2
9	58408	53616	3	2
46	237697	51233	3	2
30	58375	49288	3	2
74	215416	44942	3	2
80	54273	43797	3	2
20	46876	43445	3	2
82	55047	42133	3	2
65	50696	41889	3	2
78	49385	41738	3	2
2	46472	41644	3	2
39	61216	41193	3	2
38	50458	40721	3	2
41	50870	39606	3	2
29	46324	38985	3	2
0	39780	35250	3	2
19	42279	35133	3	2
62	43617	34978	3	2
35	43290	34829	3	2
52	41648	34323	3	2
69	36703	31902	3	2
10	33548	30030	3	2
61	36830	29851	3	2
43	102320	29215	3	2
34	34646	28029	3	2
44	57490	27283	3	2
63	31283	26640	3	2
58	32384	26523	3	2
68	29730	25359	3	2
79	36744	24716	3	2
72	125904	24299	3	2
49	34588	24215	3	2
70	51138	23207	3	2
56	26444	21766	3	2
123	31236	21322	3	2
81	28451	20937	3	2
12	23571	20685	3	2
85	24497	20586	3	2

103	25592	20060	3	2
48	24687	19308	3	2
66	21461	18785	3	2
121	27746	18601	3	2
73	21398	18002	3	2
55	26414	17610	3	2
32	21874	17346	3	2
59	23260	17077	3	2
50	19433	17026	3	2
5	20495	16478	3	2
75	18959	16030	3	2
22	19417	15743	3	2
89	19586	14294	3	2
6	16312	14165	3	2
122	17823	13914	3	2
99	16460	13119	3	2
67	15606	13024	3	2
83	17420	12079	3	2
90	13740	11071	3	2
45	13362	10912	3	2
101	14775	8725	3	2
25	8396	6923	3	2
125	7205	6018	3	2
28	5578	4308	3	2
111	4716	3939	3	2
77	4049	3488	3	2
94	2625	2221	3	2
76	2495	2149	3	2
100	3862	2119	3	2
84	1498	1307	3	2
87	3614	1245	3	2
96	1019	903	3	2
110	866	755	3	2
98	10560	8925	1	3
91	11097	8526	1	3
116	7509	6124	1	3
126	8419	6087	1	3
108	4266	3544	1	3
127	304	250	1	3
120	642	530	0	4
124	277	231	0	4

```
[36]: potential_churn.to_csv('potential_churn_customers.csv', index = False)
```

Part 7 - Analysis Dataset Creation

February 12, 2021

```
[1]: # import required libraries for dataframe and visualization
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime as dt
from lifelines import KaplanMeierFitter

# Ensuring all rows and columns are visible
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
pd.set_option('float_format', '{:.2f}'.format)
```

```
[2]: # Reading the data on which analysis needs to be done

df = pd.read_csv('cleaned_data.csv', index_col = 0)
df.head()
```

```
[2]:   Site_Level_Price_Index CUSTOMER_ID CUSTOMER_SITE_ID      SHORT_VERTICAL \
0           0.80       117841        609636  FRUIT & VEGETABLE
1           0.91       113032        578406    AERO/AUTO
2           0.80       117841        609636  FRUIT & VEGETABLE
3           0.91       113032        578406    AERO/AUTO
4           0.80       117841        609636  FRUIT & VEGETABLE

      POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE SUPPLIES_SEGMENTATION \
0         97301     END USER      Industrial          S
1         32539     END USER      Industrial          S
2         97301     END USER      Industrial          S
3         32539     END USER      Industrial          S
4         97301     END USER      Industrial          S

SUPPLIES_DECLINE_REASON DUNS_NUMBER      TRX_DATE  TRX_AMT_USD Margin \
0  Over Stocked / Timing     78842640  2016-02-05     207.72  188.59
1                  None     43202248  2016-12-16     214.79  198.94
2  Over Stocked / Timing     78842640  2016-08-26     207.72  188.59
3                  None     43202248  2016-10-19     429.58  397.89
4  Over Stocked / Timing     78842640  2016-08-19     623.16  565.77
```

	SALES_CHANNEL	QUANTITY	ORDER_TYPE	ORDER_NUM	PRODUCT_FAMILY	\
0	Online	1	STANDARD DOMESTIC	511,759,211.00		CIJ
1	Esker	1	STANDARD DOMESTIC	511,868,043.00		CIJ
2	Online	1	STANDARD DOMESTIC	511,823,154.00		CIJ
3	Esker	2	STANDARD DOMESTIC	511,849,315.00		CIJ
4	Online	3	STANDARD DOMESTIC	511,823,154.00		CIJ

	PRODUCT_MODEL	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\
0	MAKE-UP	11.00	3.00	8.00	
1	VALUE PACK	21.00	5.00	16.00	
2	MAKE-UP	11.00	3.00	8.00	
3	VALUE PACK	21.00	5.00	16.00	
4	MAKE-UP	11.00	3.00	8.00	

	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	Max_Case_Origin	\
0	Call	18.00	1.97	unknown	
1	Email	58.00	1.00	Email - VTI NACC	
2	Call	18.00	1.97	unknown	
3	Email	58.00	1.00	Email - VTI NACC	
4	Call	18.00	1.97	unknown	

	Max_Case_Reason	Num_of_Active_Install_Bases	Total_Contracts	\
0	unknown	6.00	0.00	
1	Customer Experience	4.00	4.00	
2	unknown	6.00	0.00	
3	Customer Experience	4.00	4.00	
4	unknown	6.00	0.00	

	Contract_length	Contract_Category	Num_of_Inactive_Install_Bases	\
0	0.00	No Contract	0.00	
1	1,011.25	FSMA	0.00	
2	0.00	No Contract	0.00	
3	1,011.25	FSMA	0.00	
4	0.00	No Contract	0.00	

	STRATEGIC_ACCOUNTS	TERRITORY_REGION	TRX_YEAR	
0	0	NW	2016	
1	0	SE	2016	
2	0	NW	2016	
3	0	SE	2016	
4	0	NW	2016	

[3]: # set Transaction Date to datetime
df['TRX_DATE'] = pd.to_datetime(df['TRX_DATE'])

[4]: df.shape

[4]: (380401, 35)

```
[5]: # read variables dataset
variables = pd.read_csv('variables_with_churn.csv',index_col = 0 )
variables.head()
```

```
[5]:    CUSTOMER_SITE_ID  Recency  Frequency  Num_of_Trxns  Avg_Trxn_Amt \
0              24        24      17.67          4       3,102.11
1              90        50      18.86         98       233.90
2             111       415      72.85         21       220.75
3             114        6      11.38        174       145.43
4             126       49      61.94         19       835.78

   Avg_Margin  Avg_Quantity Mode_of_Product_Family Mode_of_Product_Model \
0     2,901.24        12.00           LCM                  INK
1      172.39        11.37           CIJ                MAKE-UP
2     188.43        1.05            CIJ                MAKE-UP
3     103.28        4.43            CIJ                MAKE-UP
4      605.25        8.84           TIJ                  INK

   Types_of_Product_Family  Types_of_Product_Model \
0                      2                  2
1                      1                  3
2                      1                  2
3                      3                  4
4                      1                  1

   Most_Frequent_Sales_Channel  Most_Frequent_Order_Type  Avg_Price_Index \
0                     Copy           STANDARD DOMESTIC            0.79
1                     EDI                 EDI            1.35
2                     EDI                 EDI            1.28
3                     EDI                 EDI            1.25
4                     EDI                 EDI            0.75

   SHORT_VERTICAL  POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE \
0      GRAPHICS      60085    END USER    Industrial
1  CHEMICALS      65802    END USER    Industrial
2      GRAPHICS      56073    END USER    Industrial
3  PHARMA & MEDICAL    92614    END USER    Industrial
4  PHARMA & MEDICAL      8822    END USER    Industrial

   SUPPLIES_SEGMENTATION  SUPPLIES_DECLINE_REASON DUNS_NUMBER \
0                      S                   None  144782380
1                      S                   None  43937895
2                      S                   None  119130057
3                      M                   None  84160407
4                      S  Over Stocked / Timing  36781508
```

```

      Total_SVC_Incidents  Total_Repeat_Calls  Total_FTF_Calls  \
0                  13.00           7.00          6.00
1                  57.00          13.00         44.00
2                  1.00           0.00          1.00
3                  57.00          14.00         43.00
4                  1.00           0.00          1.00

      Most_Frequent_Interaction_Type  Total_Visits  Total_Cases  \
0                      Call        81.00       1.97
1                      Call        53.00       3.03
2                      Call        22.00       3.03
3                      Call        70.00       1.00
4                      Call        13.00       1.00

      Num_of_Active_Install_Bases  Total_Contracts  Contract_length  \
0                  5.00            0.00            0.00
1                  6.00            6.00          1,003.00
2                  2.00            0.00            0.00
3                 15.00            11.00          521.64
4                  2.00            0.00            0.00

      Contract_Category  STRATEGIC_ACCOUNTS  TERRITORY_REGION  Tenure  Cluster_Id  \
0      No Contract             0              MW    77.00        3
1          FSMA                0              MC  1,879.00        3
2      No Contract             0              MW  1,457.00        1
3          FSMA                0              NW  1,974.00        3
4      No Contract             0              NE  1,164.00        2

      churn_prob_BGNBD  Churned_365  Churned_BGNBD
0            0.05            0            0
1            0.02            0            0
2            0.49            1            1
3            0.00            0            0
4            0.01            0            0

```

[6]: variables.groupby('Churned_365')[['Tenure']].mean()

[6]: Churned_365
0 1,521.21
1 620.54
Name: Tenure, dtype: float64

[7]: variables.shape

[7]: (8367, 38)

```
[8]: df.shape
```

```
[8]: (380401, 35)
```

```
[9]: variables.head()
```

```
[9]:    CUSTOMER_SITE_ID  Recency  Frequency  Num_of_Trxns  Avg_Trxn_Amt  \
0              24        24      17.67          4     3,102.11
1              90        50      18.86         98     233.90
2             111       415      72.85         21     220.75
3             114        6      11.38        174     145.43
4             126       49      61.94         19     835.78

   Avg_Margin  Avg_Quantity Mode_of_Product_Family Mode_of_Product_Model  \
0    2,901.24        12.00           LCM                  INK
1     172.39        11.37           CIJ                MAKE-UP
2     188.43        1.05            CIJ                MAKE-UP
3     103.28        4.43            CIJ                MAKE-UP
4     605.25        8.84            TIJ                  INK

   Types_of_Product_Family  Types_of_Product_Model  \
0                      2                      2
1                      1                      3
2                      1                      2
3                      3                      4
4                      1                      1

   Most_Frequent_Sales_Channel Most_Frequent_Order_Type  Avg_Price_Index  \
0                   Copy           STANDARD DOMESTIC        0.79
1                   EDI            EDI                 1.35
2                   EDI            EDI                 1.28
3                   EDI            EDI                 1.25
4                   EDI            EDI                 0.75

   SHORT_VERTICAL  POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE  \
0      GRAPHICS      60085    END USER    Industrial
1  CHEMICALS      65802    END USER    Industrial
2      GRAPHICS      56073    END USER    Industrial
3  PHARMA & MEDICAL  92614    END USER    Industrial
4  PHARMA & MEDICAL      8822    END USER    Industrial

   SUPPLIES_SEGMENTATION SUPPLIES_DECLINE_REASON DUNS_NUMBER  \
0                      S                    None  144782380
1                      S                    None  43937895
2                      S                    None  119130057
3                      M                    None  84160407
4                      S  Over Stocked / Timing  36781508
```

```

      Total_SVC_Incidents  Total_Repeat_Calls  Total_FTF_Calls  \
0                  13.00            7.00            6.00
1                  57.00           13.00          44.00
2                  1.00            0.00            1.00
3                  57.00           14.00          43.00
4                  1.00            0.00            1.00

      Most_Frequent_Interaction_Type  Total_Visits  Total_Cases  \
0                      Call        81.00         1.97
1                      Call        53.00         3.03
2                      Call        22.00         3.03
3                      Call        70.00         1.00
4                      Call        13.00         1.00

      Num_of_Active_Install_Bases  Total_Contracts  Contract_length  \
0                  5.00            0.00            0.00
1                  6.00            6.00          1,003.00
2                  2.00            0.00            0.00
3                 15.00            11.00          521.64
4                  2.00            0.00            0.00

      Contract_Category  STRATEGIC_ACCOUNTS  TERRITORY_REGION  Tenure  Cluster_Id  \
0       No Contract            0             MW    77.00         3
1          FSMA                0             MC  1,879.00         3
2       No Contract            0             MW  1,457.00         1
3          FSMA                0             NW  1,974.00         3
4       No Contract            0             NE  1,164.00         2

      churn_prob_BGNBD  Churned_365  Churned_BGNBD
0            0.05            0            0
1            0.02            0            0
2            0.49            1            1
3            0.00            0            0
4            0.01            0            0

```

```
[10]: # merge into a big dataset
df_final = pd.merge(df, variables, on='CUSTOMER_SITE_ID')
df_final.head()
```

```
[10]:  Site_Level_Price_Index  CUSTOMER_ID  CUSTOMER_SITE_ID  SHORT_VERTICAL_x  \
0                  0.80        117841        609636  FRUIT & VEGETABLE
1                  0.80        117841        609636  FRUIT & VEGETABLE
2                  0.80        117841        609636  FRUIT & VEGETABLE
3                  0.80        117841        609636  FRUIT & VEGETABLE
4                  0.63        117841        609636  FRUIT & VEGETABLE
```

	POSTAL_CODE_x	CUSTOMER_CLASS_x	TERRITORY_TYPE_x	SUPPLIES_SEGMENTATION_x	\
0	97301	END USER	Industrial		S
1	97301	END USER	Industrial		S
2	97301	END USER	Industrial		S
3	97301	END USER	Industrial		S
4	97301	END USER	Industrial		S

	SUPPLIES_DECLINE_REASON_x	DUNS_NUMBER_x	TRX_DATE	TRX_AMT_USD	Margin	\
0	Over Stocked / Timing	78842640	2016-02-05	207.72	188.59	
1	Over Stocked / Timing	78842640	2016-08-26	207.72	188.59	
2	Over Stocked / Timing	78842640	2016-08-19	623.16	565.77	
3	Over Stocked / Timing	78842640	2016-02-10	207.72	188.59	
4	Over Stocked / Timing	78842640	2016-10-03	1,102.72	993.27	

	SALES_CHANNEL	QUANTITY	ORDER_TYPE	ORDER_NUM	PRODUCT_FAMILY	\
0	Online	1	STANDARD DOMESTIC	511,759,211.00		CIJ
1	Online	1	STANDARD DOMESTIC	511,823,154.00		CIJ
2	Online	3	STANDARD DOMESTIC	511,823,154.00		CIJ
3	Online	1	STANDARD DOMESTIC	511,759,211.00		CIJ
4	Online	8	STANDARD DOMESTIC	511,843,348.00		CIJ

	PRODUCT_MODEL	Total_SVC_Incidents_x	Total_Repeat_Calls_x	\
0	MAKE-UP	11.00	3.00	
1	MAKE-UP	11.00	3.00	
2	MAKE-UP	11.00	3.00	
3	MAKE-UP	11.00	3.00	
4	MAKE-UP	11.00	3.00	

	Total_FTF_Calls_x	Most_Frequent_Interaction_Type_x	Total_Visits_x	\
0	8.00	Call	18.00	
1	8.00	Call	18.00	
2	8.00	Call	18.00	
3	8.00	Call	18.00	
4	8.00	Call	18.00	

	Total_Cases_x	Max_Case_Origin	Max_Case_Reason	\
0	1.97	unknown	unknown	
1	1.97	unknown	unknown	
2	1.97	unknown	unknown	
3	1.97	unknown	unknown	
4	1.97	unknown	unknown	

	Num_of_Active_Install_Bases_x	Total_Contracts_x	Contract_length_x	\
0	6.00	0.00	0.00	
1	6.00	0.00	0.00	
2	6.00	0.00	0.00	
3	6.00	0.00	0.00	

4		6.00	0.00	0.00		
0	Contract_Category_x	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS_x	\		
0	No Contract	0.00	0			
1	No Contract	0.00	0			
2	No Contract	0.00	0			
3	No Contract	0.00	0			
4	No Contract	0.00	0			
0	TERRITORY_REGION_x	TRX_YEAR	Recency	Frequency	Num_of_Trxns	\
0	NW	2016	477	32.24	47	
1	NW	2016	477	32.24	47	
2	NW	2016	477	32.24	47	
3	NW	2016	477	32.24	47	
4	NW	2016	477	32.24	47	
0	Avg_Trxn_Amt	Avg_Margin	Avg_Quantity	Mode_of_Product_Family	\	
0	455.66	389.53	4.57	CIJ		
1	455.66	389.53	4.57	CIJ		
2	455.66	389.53	4.57	CIJ		
3	455.66	389.53	4.57	CIJ		
4	455.66	389.53	4.57	CIJ		
0	Mode_of_Product_Model	Types_of_Product_Family	Types_of_Product_Model	\		
0	MAKE-UP		1	3		
1	MAKE-UP		1	3		
2	MAKE-UP		1	3		
3	MAKE-UP		1	3		
4	MAKE-UP		1	3		
0	Most_Frequent_Sales_Channel	Most_Frequent_Order_Type	Avg_Price_Index	\		
0	Online	STANDARD DOMESTIC	0.80			
1	Online	STANDARD DOMESTIC	0.80			
2	Online	STANDARD DOMESTIC	0.80			
3	Online	STANDARD DOMESTIC	0.80			
4	Online	STANDARD DOMESTIC	0.80			
0	SHORT_VERTICAL_y	POSTAL_CODE_y	CUSTOMER_CLASS_y	TERRITORY_TYPE_y	\	
0	FRUIT & VEGETABLE	97301	END USER	Industrial		
1	FRUIT & VEGETABLE	97301	END USER	Industrial		
2	FRUIT & VEGETABLE	97301	END USER	Industrial		
3	FRUIT & VEGETABLE	97301	END USER	Industrial		
4	FRUIT & VEGETABLE	97301	END USER	Industrial		
0	SUPPLIES_SEGMENTATION_y	SUPPLIES_DECLINE_REASON_y	DUNS_NUMBER_y	\		
0	S	Over Stocked / Timing	78842640			
1	S	Over Stocked / Timing	78842640			

```

2           S      Over Stocked / Timing      78842640
3           S      Over Stocked / Timing      78842640
4           S      Over Stocked / Timing      78842640

  Total_SVC_Incidents_y  Total_Repeat_Calls_y  Total_FTF_Calls_y \
0            11.00          3.00             8.00
1            11.00          3.00             8.00
2            11.00          3.00             8.00
3            11.00          3.00             8.00
4            11.00          3.00             8.00

  Most_Frequent_Interaction_Type_y  Total_Visits_y  Total_Cases_y \
0                  Call        18.00         1.97
1                  Call        18.00         1.97
2                  Call        18.00         1.97
3                  Call        18.00         1.97
4                  Call        18.00         1.97

  Num_of_Active_Install_Bases_y  Total_Contracts_y  Contract_length_y \
0                6.00          0.00            0.00
1                6.00          0.00            0.00
2                6.00          0.00            0.00
3                6.00          0.00            0.00
4                6.00          0.00            0.00

  Contract_Category_y  STRATEGIC_ACCOUNTS_y  TERRITORY_REGION_y  Tenure \
0    No Contract              0            NW 1,483.00
1    No Contract              0            NW 1,483.00
2    No Contract              0            NW 1,483.00
3    No Contract              0            NW 1,483.00
4    No Contract              0            NW 1,483.00

  Cluster_Id  churn_prob_BGNBD  Churned_365  Churned_BGNBD
0          3            1.00          1            1
1          3            1.00          1            1
2          3            1.00          1            1
3          3            1.00          1            1
4          3            1.00          1            1

```

```
[11]: df_final.columns = df_final.columns.str.replace('_x', '', regex = True)
```

```
[12]: df_final.head()
```

```
[12]:   Site_Level_Price_Index  CUSTOMER_ID  CUSTOMER_SITE_ID  SHORT_VERTICAL \
0            0.80          117841          609636  FRUIT & VEGETABLE
1            0.80          117841          609636  FRUIT & VEGETABLE
2            0.80          117841          609636  FRUIT & VEGETABLE
```

3	0.80	117841	609636	FRUIT & VEGETABLE
4	0.63	117841	609636	FRUIT & VEGETABLE
POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE SUPPLIES_SEGMENTATION \				
0	97301	END USER	Industrial	S
1	97301	END USER	Industrial	S
2	97301	END USER	Industrial	S
3	97301	END USER	Industrial	S
4	97301	END USER	Industrial	S
SUPPLIES_DECLINE_REASON DUNS_NUMBER TRX_DATE TRX_AMT_USD Margin \				
0	Over Stocked / Timing	78842640	2016-02-05	207.72 188.59
1	Over Stocked / Timing	78842640	2016-08-26	207.72 188.59
2	Over Stocked / Timing	78842640	2016-08-19	623.16 565.77
3	Over Stocked / Timing	78842640	2016-02-10	207.72 188.59
4	Over Stocked / Timing	78842640	2016-10-03	1,102.72 993.27
SALES_CHANNEL QUANTITY ORDER_TYPE ORDER_NUM PRODUCT_FAMILY \				
0	Online	1	STANDARD DOMESTIC	511,759,211.00 CIJ
1	Online	1	STANDARD DOMESTIC	511,823,154.00 CIJ
2	Online	3	STANDARD DOMESTIC	511,823,154.00 CIJ
3	Online	1	STANDARD DOMESTIC	511,759,211.00 CIJ
4	Online	8	STANDARD DOMESTIC	511,843,348.00 CIJ
PRODUCT_MODEL Total_SVC_Incidents Total_Repeat_Calls Total_FTF_Calls \				
0	MAKE-UP	11.00	3.00	8.00
1	MAKE-UP	11.00	3.00	8.00
2	MAKE-UP	11.00	3.00	8.00
3	MAKE-UP	11.00	3.00	8.00
4	MAKE-UP	11.00	3.00	8.00
Most_Frequent_Interaction_Type Total_Visits Total_Cases Max_Case_Origin \				
0	Call	18.00	1.97	unknown
1	Call	18.00	1.97	unknown
2	Call	18.00	1.97	unknown
3	Call	18.00	1.97	unknown
4	Call	18.00	1.97	unknown
Max_Case_Reason Num_of_Active_Install_Bases Total_Contracts \				
0	unknown	6.00	0.00	
1	unknown	6.00	0.00	
2	unknown	6.00	0.00	
3	unknown	6.00	0.00	
4	unknown	6.00	0.00	
Contract_length Contract_Category Num_of_Inactive_Install_Bases \				
0	0.00	No Contract	0.00	

1	0.00	No Contract		0.00		
2	0.00	No Contract		0.00		
3	0.00	No Contract		0.00		
4	0.00	No Contract		0.00		
0	STRATEGIC_ACCOUNTS	TERRITORY_REGION	TRX_YEAR	Recency	Frequency	\
1	0	NW	2016	477	32.24	
2	0	NW	2016	477	32.24	
3	0	NW	2016	477	32.24	
4	0	NW	2016	477	32.24	
0	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	Avg_Quantity	\	
1	47	455.66	389.53	4.57		
2	47	455.66	389.53	4.57		
3	47	455.66	389.53	4.57		
4	47	455.66	389.53	4.57		
0	Mode_of_Product_Family	Mode_of_Product_Model	Types_of_Product_Family	\		
1	CIJ	MAKE-UP		1		
2	CIJ	MAKE-UP		1		
3	CIJ	MAKE-UP		1		
4	CIJ	MAKE-UP		1		
0	Types_of_Product_Model	Most_Frequent_Sales_Channel	\			
1	3	Online				
2	3	Online				
3	3	Online				
4	3	Online				
0	Most_Frequent_Order_Type	Avg_Price_Index	SHORT_VERTICAL_y	POSTAL_CODE_y	\	
1	STANDARD DOMESTIC	0.80	FRUIT & VEGETABLE	97301		
2	STANDARD DOMESTIC	0.80	FRUIT & VEGETABLE	97301		
3	STANDARD DOMESTIC	0.80	FRUIT & VEGETABLE	97301		
4	STANDARD DOMESTIC	0.80	FRUIT & VEGETABLE	97301		
0	CUSTOMER_CLASS_y	TERRITORY_TYPE_y	SUPPLIES_SEGMENTATION_y	\		
1	END USER	Industrial		S		
2	END USER	Industrial		S		
3	END USER	Industrial		S		
4	END USER	Industrial		S		

```

SUPPLIES_DECLINE_REASON_y DUNS_NUMBER_y Total_SVC_Incidents_y \
0 Over Stocked / Timing 78842640 11.00
1 Over Stocked / Timing 78842640 11.00
2 Over Stocked / Timing 78842640 11.00
3 Over Stocked / Timing 78842640 11.00
4 Over Stocked / Timing 78842640 11.00

Total_Repeat_Calls_y Total_FTF_Calls_y Most_Frequent_Interaction_Type_y \
0 3.00 8.00 Call
1 3.00 8.00 Call
2 3.00 8.00 Call
3 3.00 8.00 Call
4 3.00 8.00 Call

Total_Visits_y Total_Cases_y Num_of_Active_Install_Bases_y \
0 18.00 1.97 6.00
1 18.00 1.97 6.00
2 18.00 1.97 6.00
3 18.00 1.97 6.00
4 18.00 1.97 6.00

Total_Contracts_y Contract_length_y Contract_Category_y \
0 0.00 0.00 No Contract
1 0.00 0.00 No Contract
2 0.00 0.00 No Contract
3 0.00 0.00 No Contract
4 0.00 0.00 No Contract

STRATEGIC_ACCOUNTS_y TERRITORY_REGION_y Tenure Cluster_Id \
0 0 NW 1,483.00 3
1 0 NW 1,483.00 3
2 0 NW 1,483.00 3
3 0 NW 1,483.00 3
4 0 NW 1,483.00 3

churn_prob_BGNBD Churned_365 Churned_BGNBD
0 1.00 1 1
1 1.00 1 1
2 1.00 1 1
3 1.00 1 1
4 1.00 1 1

```

```
[13]: df_final.drop(['SHORT_VERTICAL_y',
                  'POSTAL_CODE_y', 'CUSTOMER_CLASS_y', 'TERRITORY_TYPE_y',
                  'SUPPLIES_SEGMENTATION_y', 'SUPPLIES_DECLINE_REASON_y',
                  'DUNS_NUMBER_y', 'Total_SVC_Incidents_y', 'Total_Repeat_Calls_y',
                  'Total_FTF_Calls_y', 'Most_Frequent_Interaction_Type_y',
```

```
'Total_Visits_y', 'Total_Cases_y',
'Num_of_Active_Install_Bases_y', 'Total_Contracts_y',
'Contract_length_y', 'Contract_Category_y',
'STRATEGIC_ACCOUNTS_y', 'TERRITORY_REGION_y'], axis = 1,inplace = True)
```

[14]: df_final.head()

	Site_Level_Price_Index	CUSTOMER_ID	CUSTOMER_SITE_ID	SHORT_VERTICAL	\	
0	0.80	117841	609636	FRUIT & VEGETABLE		
1	0.80	117841	609636	FRUIT & VEGETABLE		
2	0.80	117841	609636	FRUIT & VEGETABLE		
3	0.80	117841	609636	FRUIT & VEGETABLE		
4	0.63	117841	609636	FRUIT & VEGETABLE		
	POSTAL_CODE	CUSTOMER_CLASS	TERRITORY_TYPE	SUPPLIES_SEGMENTATION	\	
0	97301	END USER	Industrial	S		
1	97301	END USER	Industrial	S		
2	97301	END USER	Industrial	S		
3	97301	END USER	Industrial	S		
4	97301	END USER	Industrial	S		
	SUPPLIES_DECLINE_REASON	DUNS_NUMBER	TRX_DATE	TRX_AMT_USD	Margin	\
0	Over Stocked / Timing	78842640	2016-02-05	207.72	188.59	
1	Over Stocked / Timing	78842640	2016-08-26	207.72	188.59	
2	Over Stocked / Timing	78842640	2016-08-19	623.16	565.77	
3	Over Stocked / Timing	78842640	2016-02-10	207.72	188.59	
4	Over Stocked / Timing	78842640	2016-10-03	1,102.72	993.27	
	SALES_CHANNEL	QUANTITY	ORDER_TYPE	ORDER_NUM	PRODUCT_FAMILY	\
0	Online	1	STANDARD DOMESTIC	511,759,211.00	CIJ	
1	Online	1	STANDARD DOMESTIC	511,823,154.00	CIJ	
2	Online	3	STANDARD DOMESTIC	511,823,154.00	CIJ	
3	Online	1	STANDARD DOMESTIC	511,759,211.00	CIJ	
4	Online	8	STANDARD DOMESTIC	511,843,348.00	CIJ	
	PRODUCT_MODEL	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\	
0	MAKE-UP	11.00	3.00	8.00		
1	MAKE-UP	11.00	3.00	8.00		
2	MAKE-UP	11.00	3.00	8.00		
3	MAKE-UP	11.00	3.00	8.00		
4	MAKE-UP	11.00	3.00	8.00		
	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	Max_Case-Origin	\	
0	Call	18.00	1.97	unknown		
1	Call	18.00	1.97	unknown		
2	Call	18.00	1.97	unknown		
3	Call	18.00	1.97	unknown		

4	Call	18.00	1.97	unknown		
	Max_Case_Reason	Num_of_Active_Install_Bases	Total_Contracts	\		
0	unknown	6.00	0.00			
1	unknown	6.00	0.00			
2	unknown	6.00	0.00			
3	unknown	6.00	0.00			
4	unknown	6.00	0.00			
	Contract_length	Contract_Category	Num_of_Inactive_Install_Bases	\		
0	0.00	No Contract	0.00			
1	0.00	No Contract	0.00			
2	0.00	No Contract	0.00			
3	0.00	No Contract	0.00			
4	0.00	No Contract	0.00			
	STRATEGIC_ACCOUNTS	TERRITORY_REGION	TRX_YEAR	Recency	Frequency	\
0	0	NW	2016	477	32.24	
1	0	NW	2016	477	32.24	
2	0	NW	2016	477	32.24	
3	0	NW	2016	477	32.24	
4	0	NW	2016	477	32.24	
	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	Avg_Quantity	\	
0	47	455.66	389.53	4.57		
1	47	455.66	389.53	4.57		
2	47	455.66	389.53	4.57		
3	47	455.66	389.53	4.57		
4	47	455.66	389.53	4.57		
	Mode_of_Product_Family	Mode_of_Product_Model	Types_of_Product_Family	\		
0	CIJ	MAKE-UP	1			
1	CIJ	MAKE-UP	1			
2	CIJ	MAKE-UP	1			
3	CIJ	MAKE-UP	1			
4	CIJ	MAKE-UP	1			
	Types_of_Product_Model	Most_Frequent_Sales_Channel	\			
0	3	Online				
1	3	Online				
2	3	Online				
3	3	Online				
4	3	Online				
	Most_Frequent_Order_Type	Avg_Price_Index	Tenure	Cluster_Id	\	
0	STANDARD DOMESTIC	0.80	1,483.00	3		
1	STANDARD DOMESTIC	0.80	1,483.00	3		

```

2      STANDARD DOMESTIC          0.80 1,483.00      3
3      STANDARD DOMESTIC          0.80 1,483.00      3
4      STANDARD DOMESTIC          0.80 1,483.00      3

      churn_prob_BGNBD  Churned_365  Churned_BGNBD
0            1.00           1             1
1            1.00           1             1
2            1.00           1             1
3            1.00           1             1
4            1.00           1             1

[15]: df_final.sort_values('CUSTOMER_SITE_ID', ascending = True, inplace=True)

[16]: df_final.reset_index(inplace=True)

[17]: df_final = df_final.iloc[:,1:]

[18]: df_final.head()

[18]:   Site_Level_Price_Index  CUSTOMER_ID  CUSTOMER_SITE_ID  SHORT_VERTICAL \
0            0.79           6482              24    GRAPHICS
1            0.79           6482              24    GRAPHICS
2            0.79           6482              24    GRAPHICS
3            0.79           6482              24    GRAPHICS
4            1.40           37                90  CHEMICALS

      POSTAL_CODE  CUSTOMER_CLASS  TERRITORY_TYPE  SUPPLIES_SEGMENTATION \
0        60085      END USER      Industrial                  S
1        60085      END USER      Industrial                  S
2        60085      END USER      Industrial                  S
3        60085      END USER      Industrial                  S
4       65802      END USER      Industrial                  S

      SUPPLIES_DECLINE_REASON  DUNS_NUMBER  TRX_DATE  TRX_AMT_USD  Margin \
0            None        144782380 2020-03-20     1,855.74 1,381.20
1            None        144782380 2020-04-15      552.68  537.02
2            None        144782380 2020-05-04     5,000.00 4,843.37
3            None        144782380 2020-05-11     5,000.00 4,843.37
4            None        43937895 2017-01-03      213.80  129.59

      SALES_CHANNEL  QUANTITY  ORDER_TYPE  ORDER_NUM  PRODUCT_FAMILY \
0      Esker          6  STANDARD DOMESTIC  512,254,720.00      LASER
1  SFDC_CPQ          2 EQUIPMENT DOMESTIC  512,260,527.00      LCM
2    Online          20  STANDARD DOMESTIC  512,267,173.00      LCM
3     Copy          20  STANDARD DOMESTIC  512,268,803.00      LCM
4      EDI          20               EDI  511,872,093.00      CIJ

```

	PRODUCT_MODEL	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\	
0	FUME EXTRACTION	13.00	7.00	6.00		
1	INK	13.00	7.00	6.00		
2	INK	13.00	7.00	6.00		
3	INK	13.00	7.00	6.00		
4	MAKE-UP	57.00	13.00	44.00		
	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	Max_Case-Origin	\	
0	Call	81.00	1.97	unknown		
1	Call	81.00	1.97	unknown		
2	Call	81.00	1.97	unknown		
3	Call	81.00	1.97	unknown		
4	Call	53.00	3.03	unknown		
	Max_Case_Reason	Num_of_Active_Install_Bases	Total_Contracts	\		
0	unknown	5.00	0.00			
1	unknown	5.00	0.00			
2	unknown	5.00	0.00			
3	unknown	5.00	0.00			
4	unknown	6.00	6.00			
	Contract_length	Contract_Category	Num_of_Inactive_Install_Bases	\		
0	0.00	No Contract	0.00			
1	0.00	No Contract	0.00			
2	0.00	No Contract	0.00			
3	0.00	No Contract	0.00			
4	1,003.00	FSMA	0.00			
	STRATEGIC_ACCOUNTS	TERRITORY_REGION	TRX_YEAR	Recency	Frequency	\
0	0	MW	2020	24	17.67	
1	0	MW	2020	24	17.67	
2	0	MW	2020	24	17.67	
3	0	MW	2020	24	17.67	
4	0	MC	2017	50	18.86	
	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	Avg_Quantity	\	
0	4	3,102.11	2,901.24	12.00		
1	4	3,102.11	2,901.24	12.00		
2	4	3,102.11	2,901.24	12.00		
3	4	3,102.11	2,901.24	12.00		
4	98	233.90	172.39	11.37		
	Mode_of_Product_Family	Mode_of_Product_Model	Types_of_Product_Family	\		
0	LCM	INK	2			
1	LCM	INK	2			
2	LCM	INK	2			
3	LCM	INK	2			

4

CIJ

MAKE-UP

1

```
Types_of_Product_Model  Most_Frequent_Sales_Channel  \
0                      2                           Copy
1                      2                           Copy
2                      2                           Copy
3                      2                           Copy
4                      3                           EDI

Most_Frequent_Order_Type  Avg_Price_Index  Tenure  Cluster_Id  \
0      STANDARD DOMESTIC       0.79    77.00      3
1      STANDARD DOMESTIC       0.79    77.00      3
2      STANDARD DOMESTIC       0.79    77.00      3
3      STANDARD DOMESTIC       0.79    77.00      3
4                  EDI           1.35 1,879.00      3

churn_prob_BGNBD  Churned_365  Churned_BGNBD
0                0.05        0        0
1                0.05        0        0
2                0.05        0        0
3                0.05        0        0
4                0.02        0        0
```

[19]: df_final.shape

[19]: (380401, 53)

[20]: df_final.to_csv('analysis.csv')

Part 8 - Cox Regression Model

February 12, 2021

In this notebook, the dataset is analyzed for customer churn irritants for top four product families CIJ, TTO, LCM and TIJ using Cox Regression Model.

[1]: *#Importing the necessary libraries*

```
import pandas as pd
import sklearn
import lifelines
from sklearn.preprocessing import StandardScaler
```

[2]: *# Ensuring all rows and columns are visible*

```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
pd.set_option('float_format', '{:.2f}'.format)
```

[3]: *#Reading the dataset*

```
var = pd.read_csv('analysis.csv', index_col = 0)
```

[4]: *var.head()*

```
Site_Level_Price_Index CUSTOMER_ID CUSTOMER_SITE_ID SHORT_VERTICAL \
0 0.79 6482 24 GRAPHICS
1 0.79 6482 24 GRAPHICS
2 0.79 6482 24 GRAPHICS
3 0.79 6482 24 GRAPHICS
4 1.40 37 90 CHEMICALS

POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE SUPPLIES_SEGMENTATION \
0 60085 END USER Industrial S
1 60085 END USER Industrial S
2 60085 END USER Industrial S
3 60085 END USER Industrial S
4 65802 END USER Industrial S

SUPPLIES_DECLINE_REASON DUNS_NUMBER TRX_DATE TRX_AMT_USD Margin \
0 None 144782380 2020-03-20 1,855.74 1,381.20
1 None 144782380 2020-04-15 552.68 537.02
2 None 144782380 2020-05-04 5,000.00 4,843.37
3 None 144782380 2020-05-11 5,000.00 4,843.37
```

4	None	43937895	2017-01-03	213.80	129.59
0	SALES_CHANNEL	QUANTITY	ORDER_TYPE	ORDER_NUM	PRODUCT_FAMILY \ LASER
0	Esker	6	STANDARD DOMESTIC	512,254,720.00	LASER
1	SFDC_CPQ	2	EQUIPMENT DOMESTIC	512,260,527.00	LCM
2	Online	20	STANDARD DOMESTIC	512,267,173.00	LCM
3	Copy	20	STANDARD DOMESTIC	512,268,803.00	LCM
4	EDI	20	EDI	511,872,093.00	CIJ
0	PRODUCT_MODEL	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\ 6.00
0	FUME EXTRACTION	13.00	7.00	6.00	
1	INK	13.00	7.00	6.00	
2	INK	13.00	7.00	6.00	
3	INK	13.00	7.00	6.00	
4	MAKE-UP	57.00	13.00	44.00	
0	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	Max_Case_Origin	\ unknown
0	Call	81.00	1.97	unknown	
1	Call	81.00	1.97	unknown	
2	Call	81.00	1.97	unknown	
3	Call	81.00	1.97	unknown	
4	Call	53.00	3.03	unknown	
0	Max_Case_Reason	Num_of_Active_Install_Bases	Total_Contracts		\ 0.00
0	unknown	5.00	0.00		
1	unknown	5.00	0.00		
2	unknown	5.00	0.00		
3	unknown	5.00	0.00		
4	unknown	6.00	6.00		
0	Contract_length	Contract_Category	Num_of_Inactive_Install_Bases		\ 0.00
0	0.00	No Contract			
1	0.00	No Contract			
2	0.00	No Contract			
3	0.00	No Contract			
4	1,003.00	FSMA			
0	STRATEGIC_ACCOUNTS	TERRITORY_REGION	TRX_YEAR	Recency	Frequency \ 17.67
0	0	MW	2020	24	17.67
1	0	MW	2020	24	17.67
2	0	MW	2020	24	17.67
3	0	MW	2020	24	17.67
4	0	MC	2017	50	18.86
0	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	Avg_Quantity	\ 12.00
0	4	3,102.11	2,901.24	12.00	
1	4	3,102.11	2,901.24	12.00	

2	4	3,102.11	2,901.24	12.00
3	4	3,102.11	2,901.24	12.00
4	98	233.90	172.39	11.37
Mode_of_Product_Family Mode_of_Product_Model Types_of_Product_Family \				
0	LCM	INK		2
1	LCM	INK		2
2	LCM	INK		2
3	LCM	INK		2
4	CIJ	MAKE-UP		1
Types_of_Product_Model Most_Frequent_Sales_Channel \				
0	2	Copy		
1	2	Copy		
2	2	Copy		
3	2	Copy		
4	3	EDI		
Most_Frequent_Order_Type Avg_Price_Index Tenure Cluster_Id \				
0	STANDARD DOMESTIC	0.79	77.00	3
1	STANDARD DOMESTIC	0.79	77.00	3
2	STANDARD DOMESTIC	0.79	77.00	3
3	STANDARD DOMESTIC	0.79	77.00	3
4	EDI	1.35	1,879.00	3
churn_prob_BGNBD Churned_365 Churned_BGNBD				
0	0.05	0	0	
1	0.05	0	0	
2	0.05	0	0	
3	0.05	0	0	
4	0.02	0	0	

```
[5]: #var['Mode_Product_Family_Overall']
var['PRODUCT_FAMILY'].unique()
```

```
[5]: array(['LASER', 'LCM', 'CIJ', 'TTO', 'TIJ', 'GRAPHICS BA', 'BINARY ARRAY',
       'LPA', 'RAW MATERIAL', 'GRAPHICS'], dtype=object)
```

```
[6]: var['PRODUCT_FAMILY'].value_counts()
```

CIJ	314474
LCM	26654
TTO	21202
GRAPHICS BA	7268
TIJ	6027
LPA	3182
BINARY ARRAY	841

```

LASER           359
GRAPHICS       272
RAW MATERIAL   122
Name: PRODUCT_FAMILY, dtype: int64

```

```
[7]: #Drop unnecessary columns - columns with high cardinality, redundancy/
      ↪collinearity
var.drop(columns=['CUSTOMER_SITE_ID', \
                  'DUNS_NUMBER', 'TRX_YEAR', 'POSTAL_CODE', \
                  'ORDER_TYPE', \
                  'Mode_of_Product_Family', 'Mode_of_Product_Model', \
                  'TRX_DATE', 'ORDER_NUM', 'Cluster_Id', 'CUSTOMER_ID', \
                  'churn_prob_BGNBD', 'Churned_BGNBD', 'Recency', 'Avg_Trxn_Amt' \
], inplace=True)
```

```
[8]: var.shape
```

```
[8]: (380401, 38)
```

```
[9]: var.head()
```

```
[9]: Site_Level_Price_Index SHORT_VERTICAL CUSTOMER_CLASS TERRITORY_TYPE \
0          0.79      GRAPHICS      END USER    Industrial
1          0.79      GRAPHICS      END USER    Industrial
2          0.79      GRAPHICS      END USER    Industrial
3          0.79      GRAPHICS      END USER    Industrial
4          1.40      CHEMICALS     END USER    Industrial

SUPPLIES_SEGMENTATION SUPPLIES_DECLINE_REASONs TRX_AMT_USD Margin \
0                   S             None    1,855.74 1,381.20
1                   S             None     552.68  537.02
2                   S             None    5,000.00 4,843.37
3                   S             None    5,000.00 4,843.37
4                   S             None     213.80  129.59

SALES_CHANNEL QUANTITY PRODUCT_FAMILY PRODUCT_MODEL \
0      Esker        6      LASER  FUME EXTRACTION
1    SFDC_CPQ       2      LCM      INK
2     Online       20      LCM      INK
3      Copy        20      LCM      INK
4      EDI         20      CIJ      MAKE-UP

Total_SVC_Incidents Total_Repeat_Calls Total_FTF_Calls \
0            13.00          7.00          6.00
1            13.00          7.00          6.00
2            13.00          7.00          6.00
3            13.00          7.00          6.00
```

4	57.00	13.00	44.00
0	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases
0	Call	81.00	1.97
1	Call	81.00	1.97
2	Call	81.00	1.97
3	Call	81.00	1.97
4	Call	53.00	3.03
0	Max_Case_Reason	Num_of_Active_Install_Bases	Total_Contracts
0	unknown	5.00	0.00
1	unknown	5.00	0.00
2	unknown	5.00	0.00
3	unknown	5.00	0.00
4	unknown	6.00	6.00
0	Contract_length	Contract_Category	Num_of_Inactive_Install_Bases
0	0.00	No Contract	0.00
1	0.00	No Contract	0.00
2	0.00	No Contract	0.00
3	0.00	No Contract	0.00
4	1,003.00	FSMA	0.00
0	STRATEGIC_ACCOUNTS	TERRITORY_REGION	Frequency
0	0	MW	17.67
1	0	MW	17.67
2	0	MW	17.67
3	0	MW	17.67
4	0	MC	18.86
4			98
4			172.39
0	Avg_Quantity	Types_of_Product_Family	Types_of_Product_Model
0	12.00	2	2
1	12.00	2	2
2	12.00	2	2
3	12.00	2	2
4	11.37	1	3
0	Most_Frequent_Sales_Channel	Most_Frequent_Order_Type	Avg_Price_Index
0	Copy	STANDARD DOMESTIC	0.79
1	Copy	STANDARD DOMESTIC	0.79
2	Copy	STANDARD DOMESTIC	0.79
3	Copy	STANDARD DOMESTIC	0.79
4	EDI		1.35
0	Tenure	Churned_365	
0	77.00	0	
1	77.00	0	

```

2    77.00      0
3    77.00      0
4 1,879.00      0

```

```
[10]: var = pd.get_dummies(var,columns = [
    →['SHORT_VERTICAL','CUSTOMER_CLASS','TERRITORY_TYPE','SUPPLIES_SEGMENTATION',
     →'SUPPLIES_DECLINE_REASON', 'SALES_CHANNEL', 'PRODUCT_FAMILY', 'PRODUCT_MODEL',
     →'Most_Frequent_Interaction_Type', 'Max_Case_Origin', 'Max_Case_Reason',
     →'Contract_Category', 'TERRITORY_REGION', 'Most_Frequent_Sales_Channel',
     →'Most_Frequent_Order_Type'], 
    →drop_first=True)
```

```
[11]: var.shape
```

```
[11]: (380401, 157)
```

```
[12]: # drop_duplicates
var = var.drop_duplicates()
var.shape
```

```
[12]: (235924, 157)
```

```
[13]: #Filter dataset on the four major product families - CIJ, TTO, LCM, TIJ
```

```
[14]: cij = var[var['PRODUCT_FAMILY_CIJ']==1].copy()
```

```
[15]: tto = var[var['PRODUCT_FAMILY_TTO']==1].copy()
```

```
[16]: lcm = var[var['PRODUCT_FAMILY_LCM']==1].copy()
```

```
[17]: tij = var[var['PRODUCT_FAMILY_TIJ']==1].copy()
```

1 Model building for CIJ

```
[19]: cij.reset_index(inplace=True,drop=True)
```

```
[20]: cij.head()
```

	Site_Level_Price_Index	TRX_AMT_USD	Margin	QUANTITY	Total_SVC_Incidents
0	1.40	213.80	129.59	20	57.00
1	1.40	222.40	138.19	20	57.00
2	1.40	66.72	41.46	6	57.00
3	1.40	44.48	27.64	4	57.00

4	1.40	33.36	20.73	3	57.00
0	Total_Repeat_Calls	Total_FTF_Calls	Total_Visits	Total_Cases	\
1	13.00	44.00	53.00	3.03	
2	13.00	44.00	53.00	3.03	
3	13.00	44.00	53.00	3.03	
4	13.00	44.00	53.00	3.03	
0	Num_of_Active_Install_Bases	Total_Contracts	Contract_length	\	
1	6.00	6.00	1,003.00		
2	6.00	6.00	1,003.00		
3	6.00	6.00	1,003.00		
4	6.00	6.00	1,003.00		
0	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	Frequency	Num_of_Trxns	\
1	0.00	0	18.86	98	
2	0.00	0	18.86	98	
3	0.00	0	18.86	98	
4	0.00	0	18.86	98	
0	Avg_Margin	Avg_Quantity	Types_of_Product_Family	Types_of_Product_Model	\
1	172.39	11.37	1	3	
2	172.39	11.37	1	3	
3	172.39	11.37	1	3	
4	172.39	11.37	1	3	
0	Avg_Price_Index	Tenure	Churned_365	\	
1	1.35	1,879.00	0		
2	1.35	1,879.00	0		
3	1.35	1,879.00	0		
4	1.35	1,879.00	0		
0	SHORT_VERTICAL_BAKED_GOODS & CEREALS	SHORT_VERTICAL_BEVERAGE	\		
1	0	0			
2	0	0			
3	0	0			
4	0	0			
0	SHORT_VERTICAL_BUILDING_MATERIALS	SHORT_VERTICAL_CANDY & CONFECTION	\		
1	0	0			

SUPPLIES_SEGMENTATION_M	SUPPLIES_SEGMENTATION_S	\
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1
SUPPLIES_SEGMENTATION_Unclass	SUPPLIES_SEGMENTATION_XL	\
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
SUPPLIES_DECLINE_REASON_Financial Distress/Credit Hold	\	
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASON_Migration to 1000 Line/TIJ/TTO/LCM/LPA	\	
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASON_Migration to Lasers	\	
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASON_Moved Equipment	\	
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASON_No More Coding Requirement	\	
0	0	
1	0	
2	0	

3		0	
4		0	
	SUPPLIES_DECLINE_REASON_None	SUPPLIES_DECLINE_REASON_Off Brand \	
0	1	0	
1	1	0	
2	1	0	
3	1	0	
4	1	0	
	SUPPLIES_DECLINE_REASON_Over Stocked / Timing \		
0		0	
1		0	
2		0	
3		0	
4		0	
	SUPPLIES_DECLINE_REASON_Pricing / Discounting \		
0		0	
1		0	
2		0	
3		0	
4		0	
	SUPPLIES_DECLINE_REASON_Printing/EQ downtime Issues \		
0		0	
1		0	
2		0	
3		0	
4		0	
	SUPPLIES_DECLINE_REASON_Production / Code Reduction \		
0		0	
1		0	
2		0	
3		0	
4		0	
	SUPPLIES_DECLINE_REASON_Production Down (timing) \		
0		0	
1		0	
2		0	
3		0	
4		0	
	SUPPLIES_DECLINE_REASON_Project Based \		
0		0	

1		0	
2		0	
3		0	
4		0	
	SUPPLIES_DECLINE_REASONs_Recent Regain/Win-back \		
0		0	
1		0	
2		0	
3		0	
4		0	
	SUPPLIES_DECLINE_REASONs_Seasonal Producer \		
0		0	
1		0	
2		0	
3		0	
4		0	
	SUPPLIES_DECLINE_REASONs_Served by Authorized Distributor \		
0		0	
1		0	
2		0	
3		0	
4		0	
	SUPPLIES_DECLINE_REASONs_Site Closed \		
0		0	
1		0	
2		0	
3		0	
4		0	
	SUPPLIES_DECLINE_REASONs_VJ Operations Issues SALES_CHANNEL_Copy \		
0		0	0
1		0	0
2		0	0
3		0	0
4		0	0
	SALES_CHANNEL_ECOMM PO IMPORT SALES_CHANNEL_EDI SALES_CHANNEL_Esker \		
0		0	1 0
1		0	1 0
2		0	0 0
3		0	0 0
4		0	0 0

	SALES_CHANNEL_IStore Account	SALES_CHANNEL_OCC	SALES_CHANNEL_Online	\	
0	0	0	0		
1	0	0	0		
2	0	0	1		
3	0	0	1		
4	0	0	0		
	SALES_CHANNEL_SFDC_CPQ	SALES_CHANNEL_Service	Billing	PRODUCT_FAMILY_CIJ	\
0	0		0	1	
1	0		0	1	
2	0		0	1	
3	0		0	1	
4	0		0	1	
	PRODUCT_FAMILY_GRAPHICS	PRODUCT_FAMILY_GRAPHICS	BA	PRODUCT_FAMILY_LASER	\
0	0		0	0	
1	0		0	0	
2	0		0	0	
3	0		0	0	
4	0		0	0	
	PRODUCT_FAMILY_LCM	PRODUCT_FAMILY_LPA	PRODUCT_FAMILY_RAW_MATERIAL	\	
0	0	0		0	
1	0	0		0	
2	0	0		0	
3	0	0		0	
4	0	0		0	
	PRODUCT_FAMILY_TIJ	PRODUCT_FAMILY_TTO	PRODUCT_MODEL_CLEANING SOLUTION	\	
0	0	0		0	
1	0	0		0	
2	0	0		0	
3	0	0		0	
4	0	0		0	
	PRODUCT_MODEL_FUME_EXTRACTION	PRODUCT_MODEL_INK	PRODUCT_MODEL_LABELS	\	
0	0	0		0	
1	0	0		0	
2	0	0		0	
3	0	0		0	
4	0	0		0	
	PRODUCT_MODEL_MAKE-UP	PRODUCT_MODEL_PACKAGING	PRODUCT_MODEL_RIBBONS	\	
0	1	0	0		
1	1	0	0		
2	1	0	0		
3	1	0	0		

	4	1	0	0
PRODUCT_MODEL_SOLVENT	0	0	0	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
Most_Frequent_Interaction_Type_Callback	\	0	0	
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
Most_Frequent_Interaction_Type_Contact_Customer	\	0	0	
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
Most_Frequent_Interaction_Type_Customer_Meeting	\	0	0	
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
Most_Frequent_Interaction_Type_Dial	Most_Frequent_Interaction_Type_Email	\	0	
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
Most_Frequent_Interaction_Type_Make_Qualified_Sales_Call	\	0	0	
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
Most_Frequent_Interaction_Type_Meeting	\	0	0	
0	0	0	0	
1	0	0	0	

2		0
3		0
4		0
	Most_Frequent_Interaction_Type_Other \	
0		0
1		0
2		0
3		0
4		0
	Most_Frequent_Interaction_Type_TS Task \	
0		0
1		0
2		0
3		0
4		0
	Max_Case-Origin_CX Survey Detractor	Max_Case-Origin_Email \
0		0
1		0
2		0
3		0
4		0
	Max_Case-Origin_Email - VTI CC Sales Escalations \	
0		0
1		0
2		0
3		0
4		0
	Max_Case-Origin_Email - VTI NACC	Max_Case-Origin_Email/Fax - VTI CS \
0		0
1		0
2		0
3		0
4		0
	Max_Case-Origin_FS Survey Followup	Max_Case-Origin_Install Complete \
0		0
1		0
2		0
3		0
4		0
	Max_Case-Origin_Phone	Max_Case-Origin_TS Survey Followup \

0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Origin_unknown	Max_Case_Reason_CX: Customer Care \
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0
	Max_Case_Reason_CX: Field Sales	Max_Case_Reason_CX: Field Service \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Reason_CX: Manufacturing	Max_Case_Reason_CX: Other Team \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Reason_CX: Tech Support	Max_Case_Reason_Customer Experience \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Reason_unknown	Contract_Category_Full Care \
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0
	Contract_Category_No Contract	Contract_Category_Supportive \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0

	Contract_Category_WFC	TERRITORY_REGION_MW	TERRITORY_REGION_NE	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
	TERRITORY_REGION_NW	TERRITORY_REGION_SC	TERRITORY_REGION_SE	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
	Most_Frequent_Sales_Channel_Copy	Most_Frequent_Sales_Channel_EDI	\	
0	0	1		
1	0	1		
2	0	1		
3	0	1		
4	0	1		
	Most_Frequent_Sales_Channel_Esker	\		
0	0			
1	0			
2	0			
3	0			
4	0			
	Most_Frequent_Sales_Channel_IStore_Account	\		
0	0			
1	0			
2	0			
3	0			
4	0			
	Most_Frequent_Sales_Channel_OCC	Most_Frequent_Sales_Channel_Online	\	
0	0	0		
1	0	0		
2	0	0		
3	0	0		
4	0	0		
	Most_Frequent_Sales_Channel_SFDC_CPQ	\		
0	0			
1	0			
2	0			

```
3 0  
4 0  
  
Most_Frequent_Sales_Channel_Service Billing \\  
0 0  
1 0  
2 0  
3 0  
4 0  
  
Most_Frequent_Order_Type_BILL ONLY \\  
0 0  
1 0  
2 0  
3 0  
4 0  
  
Most_Frequent_Order_Type_DEMO EQUIPMENT ACCEPT \\  
0 0  
1 0  
2 0  
3 0  
4 0  
  
Most_Frequent_Order_Type_EDI Most_Frequent_Order_Type_EQUIPMENT DOMESTIC \\  
0 1 0  
1 1 0  
2 1 0  
3 1 0  
4 1 0  
  
Most_Frequent_Order_Type_SERVICE \\  
0 0  
1 0  
2 0  
3 0  
4 0  
  
Most_Frequent_Order_Type_STANDARD DOMESTIC \\  
0 0  
1 0  
2 0  
3 0  
4 0  
  
Most_Frequent_Order_Type_STANDARD INTERNATIONAL \\  
0 0
```

```

1          0
2          0
3          0
4          0

    Most_Frequent_Order_Type_US FULL CARE EQPT DOMESTIC \
0          0
1          0
2          0
3          0
4          0

    Most_Frequent_Order_Type_US FULL CARE INTERNATIONAL \
0          0
1          0
2          0
3          0
4          0

    Most_Frequent_Order_Type_WEB ORDER
0          0
1          0
2          0
3          0
4          0

```

[21]: cijChurned = cij[['Churned_365']]

[22]: cijChurned.head()

[22]: Churned_365

0	0
1	0
2	0
3	0
4	0

[23]: cij.drop(columns=['Churned_365'],axis=1,inplace=True)

[24]: *#Scaling*
Instantiate
scaler = StandardScaler()

fit_transform
cij_scaled = scaler.fit_transform(cij)

```
[25]: cij_scaled = pd.DataFrame(cij_scaled)
cij_scaled.columns = cij.columns
cij_scaled.head()
```

	Site_Level_Price_Index	TRX_AMT_USD	Margin	QUANTITY	Total_SVC_Incidents	\
0	1.07	-0.23	-0.28	0.36		0.01
1	1.07	-0.22	-0.27	0.36		0.01
2	1.07	-0.38	-0.39	-0.28		0.01
3	1.07	-0.40	-0.40	-0.37		0.01
4	1.07	-0.41	-0.41	-0.41		0.01

	Total_Repeat_Calls	Total_FTF_Calls	Total_Visits	Total_Cases	\
0	-0.12	0.07	0.00	1.37	
1	-0.12	0.07	0.00	1.37	
2	-0.12	0.07	0.00	1.37	
3	-0.12	0.07	0.00	1.37	
4	-0.12	0.07	0.00	1.37	

	Num_of_Active_Install_Bases	Total_Contracts	Contract_length	\
0	-0.21	-0.17	0.47	
1	-0.21	-0.17	0.47	
2	-0.21	-0.17	0.47	
3	-0.21	-0.17	0.47	
4	-0.21	-0.17	0.47	

	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	Frequency	Num_of_Trxns	\
0	-0.12	-0.49	-0.36	-0.20	
1	-0.12	-0.49	-0.36	-0.20	
2	-0.12	-0.49	-0.36	-0.20	
3	-0.12	-0.49	-0.36	-0.20	
4	-0.12	-0.49	-0.36	-0.20	

	Avg_Margin	Avg_Quantity	Types_of_Product_Family	Types_of_Product_Model	\
0	-0.39	-0.05		-0.61	-0.46
1	-0.39	-0.05		-0.61	-0.46
2	-0.39	-0.05		-0.61	-0.46
3	-0.39	-0.05		-0.61	-0.46
4	-0.39	-0.05		-0.61	-0.46

	Avg_Price_Index	Tenure	SHORT_VERTICAL_BAKED_GOODS & CEREALS	\
0	1.21	0.36		-0.19
1	1.21	0.36		-0.19
2	1.21	0.36		-0.19
3	1.21	0.36		-0.19
4	1.21	0.36		-0.19

	SHORT_VERTICAL_BEVERAGE	SHORT_VERTICAL_BUILDING_MATERIALS	\
--	-------------------------	-----------------------------------	---

0	-0.32	-0.21
1	-0.32	-0.21
2	-0.32	-0.21
3	-0.32	-0.21
4	-0.32	-0.21
0	SHORT_VERTICAL_CANDY & CONFECTION	SHORT_VERTICAL_CHEMICALS \
1	-0.11	5.25
2	-0.11	5.25
3	-0.11	5.25
4	-0.11	5.25
0	SHORT_VERTICAL_COSMETICS / PERSONAL CARE	SHORT_VERTICAL_DAIRY & EGGS \
1	-0.22	-0.28
2	-0.22	-0.28
3	-0.22	-0.28
4	-0.22	-0.28
0	SHORT_VERTICAL_DISTRIBUTOR	SHORT_VERTICAL_ELECTRICAL / ELECTRONICS \
1	-0.04	-0.16
2	-0.04	-0.16
3	-0.04	-0.16
4	-0.04	-0.16
0	SHORT_VERTICAL_EXTRUSION / WIRE & CABLE	SHORT_VERTICAL_FISH & SEAFOOD \
1	-0.16	-0.06
2	-0.16	-0.06
3	-0.16	-0.06
4	-0.16	-0.06
0	SHORT_VERTICAL_FROZEN PREPARED MEALS	SHORT_VERTICAL_FRUIT & VEGETABLE \
1	-0.09	-0.21
2	-0.09	-0.21
3	-0.09	-0.21
4	-0.09	-0.21
0	SHORT_VERTICAL_GRAPHICS	SHORT_VERTICAL_INDUSTRIAL EQUIPMENT \
1	-0.27	-0.19
2	-0.27	-0.19
3	-0.27	-0.19
4	-0.27	-0.19

	SHORT_VERTICAL_MEAT & POULTRY	SHORT_VERTICAL_OEM-INDUSTRIAL EQUIPMENT	\	
0	-0.22	-0.07		
1	-0.22	-0.07		
2	-0.22	-0.07		
3	-0.22	-0.07		
4	-0.22	-0.07		
	SHORT_VERTICAL_OTHER	SHORT_VERTICAL_OTHER FOOD	\	
0	-0.33	-0.33		
1	-0.33	-0.33		
2	-0.33	-0.33		
3	-0.33	-0.33		
4	-0.33	-0.33		
	SHORT_VERTICAL_PACKAGING MATERIALS	SHORT_VERTICAL_PET FOOD & ANIMAL FEED	\	
0	-0.11	-0.12		
1	-0.11	-0.12		
2	-0.11	-0.12		
3	-0.11	-0.12		
4	-0.11	-0.12		
	SHORT_VERTICAL_PHARMA & MEDICAL	SHORT_VERTICAL_POSTAL	\	
0	-0.26	-0.02		
1	-0.26	-0.02		
2	-0.26	-0.02		
3	-0.26	-0.02		
4	-0.26	-0.02		
	SHORT_VERTICAL_SALTY SNACKS	SHORT_VERTICAL_TEXTILE	\	
0	-0.08	-0.08		
1	-0.08	-0.08		
2	-0.08	-0.08		
3	-0.08	-0.08		
4	-0.08	-0.08		
	SHORT_VERTICAL_TOBACCO	SHORT_VERTICAL_UNKNOWN	CUSTOMER_CLASS_END USER	\
0	-0.05	-0.18	0.11	
1	-0.05	-0.18	0.11	
2	-0.05	-0.18	0.11	
3	-0.05	-0.18	0.11	
4	-0.05	-0.18	0.11	
	CUSTOMER_CLASS_OEM	TERRITORY_TYPE_Industrial	TERRITORY_TYPE_Postal	\
0	-0.09	0.16	-0.05	
1	-0.09	0.16	-0.05	
2	-0.09	0.16	-0.05	

3	-0.09	0.16	-0.05
4	-0.09	0.16	-0.05
	SUPPLIES_SEGMENTATION_M	SUPPLIES_SEGMENTATION_S	\
0	-0.43	1.15	
1	-0.43	1.15	
2	-0.43	1.15	
3	-0.43	1.15	
4	-0.43	1.15	
	SUPPLIES_SEGMENTATION_Unclass	SUPPLIES_SEGMENTATION_XL	\
0	-0.01	-0.37	
1	-0.01	-0.37	
2	-0.01	-0.37	
3	-0.01	-0.37	
4	-0.01	-0.37	
	SUPPLIES_DECLINE_REASON_Financial Distress/Credit Hold	\	
0		-0.02	
1		-0.02	
2		-0.02	
3		-0.02	
4		-0.02	
	SUPPLIES_DECLINE_REASON_Migration to 1000 Line/TIJ/TTO/LCM/LPA	\	
0		-0.33	
1		-0.33	
2		-0.33	
3		-0.33	
4		-0.33	
	SUPPLIES_DECLINE_REASON_Migration to Lasers	\	
0		-0.09	
1		-0.09	
2		-0.09	
3		-0.09	
4		-0.09	
	SUPPLIES_DECLINE_REASON_Moved Equipment	\	
0		-0.06	
1		-0.06	
2		-0.06	
3		-0.06	
4		-0.06	
	SUPPLIES_DECLINE_REASON_No More Coding Requirement	\	
0		-0.05	

1		-0.05	
2		-0.05	
3		-0.05	
4		-0.05	
	SUPPLIES_DECLINE_REASONS_None	SUPPLIES_DECLINE_REASONS_Off	Brand \
0	0.76		-0.14
1	0.76		-0.14
2	0.76		-0.14
3	0.76		-0.14
4	0.76		-0.14
	SUPPLIES_DECLINE_REASONS_Over Stocked / Timing \		
0		-0.29	
1		-0.29	
2		-0.29	
3		-0.29	
4		-0.29	
	SUPPLIES_DECLINE_REASONS_Pricing / Discounting \		
0		-0.01	
1		-0.01	
2		-0.01	
3		-0.01	
4		-0.01	
	SUPPLIES_DECLINE_REASONS_Printing/EQ downtime Issues \		
0		-0.03	
1		-0.03	
2		-0.03	
3		-0.03	
4		-0.03	
	SUPPLIES_DECLINE_REASONS_Production / Code Reduction \		
0		-0.23	
1		-0.23	
2		-0.23	
3		-0.23	
4		-0.23	
	SUPPLIES_DECLINE_REASONS_Production Down (timing) \		
0		-0.15	
1		-0.15	
2		-0.15	
3		-0.15	
4		-0.15	

SUPPLIES_DECLINE_REASONs_Project Based \			
0	-0.04		
1	-0.04		
2	-0.04		
3	-0.04		
4	-0.04		
SUPPLIES_DECLINE_REASONs_Recent Regain/Win-back \			
0	-0.09		
1	-0.09		
2	-0.09		
3	-0.09		
4	-0.09		
SUPPLIES_DECLINE_REASONs_Seasonal Producer \			
0	-0.08		
1	-0.08		
2	-0.08		
3	-0.08		
4	-0.08		
SUPPLIES_DECLINE_REASONs_Served by Authorized Distributor \			
0	-0.09		
1	-0.09		
2	-0.09		
3	-0.09		
4	-0.09		
SUPPLIES_DECLINE_REASONs_Site Closed \			
0	-0.11		
1	-0.11		
2	-0.11		
3	-0.11		
4	-0.11		
SUPPLIES_DECLINE_REASONs_VJ Operations Issues SALES_CHANNEL_Copy \			
0	-0.02	-0.17	
1	-0.02	-0.17	
2	-0.02	-0.17	
3	-0.02	-0.17	
4	-0.02	-0.17	
SALES_CHANNEL_ECOMM PO IMPORT SALES_CHANNEL_EDI SALES_CHANNEL_Esker \			
0	-0.01	9.07	-0.58
1	-0.01	9.07	-0.58
2	-0.01	-0.11	-0.58
3	-0.01	-0.11	-0.58

4		-0.01	-0.11	-0.58
0	SALES_CHANNEL_IStore Account	SALES_CHANNEL_OCC	SALES_CHANNEL_Online	\ -1.22
1		-0.13	-0.23	-1.22
2		-0.13	-0.23	0.82
3		-0.13	-0.23	0.82
4		-0.13	-0.23	-1.22
0	SALES_CHANNEL_SFDC_CPQ	SALES_CHANNEL_Service	Billing	PRODUCT_FAMILY_CIJ \ 0.00
1		-0.09	0.00	0.00
2		-0.09	0.00	0.00
3		-0.09	0.00	0.00
4		-0.09	0.00	0.00
0	PRODUCT_FAMILY_GRAPHICS	PRODUCT_FAMILY_GRAPHICS	BA	PRODUCT_FAMILY_LASER \ 0.00
1		0.00	0.00	0.00
2		0.00	0.00	0.00
3		0.00	0.00	0.00
4		0.00	0.00	0.00
0	PRODUCT_FAMILY_LCM	PRODUCT_FAMILY_LPA	PRODUCT_FAMILY_RAW MATERIAL	\ 0.00
1		0.00	0.00	0.00
2		0.00	0.00	0.00
3		0.00	0.00	0.00
4		0.00	0.00	0.00
0	PRODUCT_FAMILY_TIJ	PRODUCT_FAMILY_TTO	PRODUCT_MODEL_CLEANING SOLUTION	\ -0.46
1		0.00	0.00	-0.46
2		0.00	0.00	-0.46
3		0.00	0.00	-0.46
4		0.00	0.00	-0.46
0	PRODUCT_MODEL_FUME_EXTRACTION	PRODUCT_MODEL_INK	PRODUCT_MODEL_LABELS	\ 0.00
1		0.00	-0.65	0.00
2		0.00	-0.65	0.00
3		0.00	-0.65	0.00
4		0.00	-0.65	0.00
0	PRODUCT_MODEL_MAKE-UP	PRODUCT_MODEL_PACKAGING	PRODUCT_MODEL_RIBBONS	\ 0.00
1		0.98	0.00	0.00
		0.98	0.00	0.00

2	0.98	0.00	0.00
3	0.98	0.00	0.00
4	0.98	0.00	0.00
	PRODUCT_MODEL_SOLVENT	PRODUCT_MODEL_VALUE	PACK \
0	0.00	-0.15	
1	0.00	-0.15	
2	0.00	-0.15	
3	0.00	-0.15	
4	0.00	-0.15	
	Most_Frequent_Interaction_Type_Callback	\	
0		-0.01	
1		-0.01	
2		-0.01	
3		-0.01	
4		-0.01	
	Most_Frequent_Interaction_Type_Contact	Customer \	
0		-0.01	
1		-0.01	
2		-0.01	
3		-0.01	
4		-0.01	
	Most_Frequent_Interaction_Type_Customer	Meeting \	
0		-0.04	
1		-0.04	
2		-0.04	
3		-0.04	
4		-0.04	
	Most_Frequent_Interaction_Type_Dial	Most_Frequent_Interaction_Type_Email \	
0	-0.01	-0.34	
1	-0.01	-0.34	
2	-0.01	-0.34	
3	-0.01	-0.34	
4	-0.01	-0.34	
	Most_Frequent_Interaction_Type_Make	Qualified Sales Call \	
0		0.00	
1		0.00	
2		0.00	
3		0.00	
4		0.00	
	Most_Frequent_Interaction_Type_Meeting	\	

0		-0.08
1		-0.08
2		-0.08
3		-0.08
4		-0.08
	Most_Frequent_Interaction_Type_Other \	
0		-0.28
1		-0.28
2		-0.28
3		-0.28
4		-0.28
	Most_Frequent_Interaction_Type_TS Task \	
0		-0.01
1		-0.01
2		-0.01
3		-0.01
4		-0.01
	Max_Case-Origin_CX Survey Detractor Max_Case-Origin_Email \	
0		-0.04
1		-0.04
2		-0.04
3		-0.04
4		-0.04
	Max_Case-Origin_Email - VTI CC Sales Escalations \	
0		-0.00
1		-0.00
2		-0.00
3		-0.00
4		-0.00
	Max_Case-Origin_Email - VTI NACC Max_Case-Origin_Email/Fax - VTI CS \	
0		-0.42
1		-0.42
2		-0.42
3		-0.42
4		-0.42
	Max_Case-Origin_FS Survey Followup Max_Case-Origin_Install Complete \	
0		-0.07
1		-0.07
2		-0.07
3		-0.07
4		-0.07

	Max_Case_Origin_Phone	Max_Case_Origin_TS	Survey	Followup	\
0	-0.34			-0.12	
1	-0.34			-0.12	
2	-0.34			-0.12	
3	-0.34			-0.12	
4	-0.34			-0.12	
	Max_Case_Origin_unknown	Max_Case_Reason_CX: Customer Care			\
0	0.68			-0.18	
1	0.68			-0.18	
2	0.68			-0.18	
3	0.68			-0.18	
4	0.68			-0.18	
	Max_Case_Reason_CX: Field Sales	Max_Case_Reason_CX: Field Service			\
0		-0.03		-0.09	
1		-0.03		-0.09	
2		-0.03		-0.09	
3		-0.03		-0.09	
4		-0.03		-0.09	
	Max_Case_Reason_CX: Manufacturing	Max_Case_Reason_CX: Other Team			\
0		-0.02		-0.03	
1		-0.02		-0.03	
2		-0.02		-0.03	
3		-0.02		-0.03	
4		-0.02		-0.03	
	Max_Case_Reason_CX: Tech Support	Max_Case_Reason_Customer Experience			\
0		-0.13		-0.59	
1		-0.13		-0.59	
2		-0.13		-0.59	
3		-0.13		-0.59	
4		-0.13		-0.59	
	Max_Case_Reason_unknown	Contract_Category_Full Care			\
0	0.68		-0.28		
1	0.68		-0.28		
2	0.68		-0.28		
3	0.68		-0.28		
4	0.68		-0.28		
	Contract_Category_No Contract	Contract_Category_Supportive			\
0		-0.60		-0.04	
1		-0.60		-0.04	
2		-0.60		-0.04	

3	-0.60	-0.04		
4	-0.60	-0.04		
	Contract_Category_WFC	TERRITORY_REGION_MW	TERRITORY_REGION_NE	\
0	-0.02	-0.41	-0.52	
1	-0.02	-0.41	-0.52	
2	-0.02	-0.41	-0.52	
3	-0.02	-0.41	-0.52	
4	-0.02	-0.41	-0.52	
	TERRITORY_REGION_NW	TERRITORY_REGION_SC	TERRITORY_REGION_SE	\
0	-0.49	-0.37	-0.43	
1	-0.49	-0.37	-0.43	
2	-0.49	-0.37	-0.43	
3	-0.49	-0.37	-0.43	
4	-0.49	-0.37	-0.43	
	Most_Frequent_Sales_Channel_Copy	Most_Frequent_Sales_Channel_EDI	\	
0	-0.04	7.32		
1	-0.04	7.32		
2	-0.04	7.32		
3	-0.04	7.32		
4	-0.04	7.32		
	Most_Frequent_Sales_Channel_Esker	\		
0	-0.65			
1	-0.65			
2	-0.65			
3	-0.65			
4	-0.65			
	Most_Frequent_Sales_Channel_IStore	Account	\	
0		-0.13		
1		-0.13		
2		-0.13		
3		-0.13		
4		-0.13		
	Most_Frequent_Sales_Channel_OCC	Most_Frequent_Sales_Channel_Online	\	
0	-0.14	-1.31		
1	-0.14	-1.31		
2	-0.14	-1.31		
3	-0.14	-1.31		
4	-0.14	-1.31		
	Most_Frequent_Sales_Channel_SFDC_CPQ	\		
0	-0.05			

1		-0.05
2		-0.05
3		-0.05
4		-0.05
	Most_Frequent_Sales_Channel_Service Billing \	
0		0.00
1		0.00
2		0.00
3		0.00
4		0.00
	Most_Frequent_Order_Type_BILL ONLY \	
0		-0.01
1		-0.01
2		-0.01
3		-0.01
4		-0.01
	Most_Frequent_Order_Type_DEMO EQUIPMENT ACCEPT \	
0		-0.01
1		-0.01
2		-0.01
3		-0.01
4		-0.01
	Most_Frequent_Order_Type_EDI Most_Frequent_Order_Type_EQUIPMENT DOMESTIC \	
0	7.39	-0.10
1	7.39	-0.10
2	7.39	-0.10
3	7.39	-0.10
4	7.39	-0.10
	Most_Frequent_Order_Type_SERVICE \	
0	0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
	Most_Frequent_Order_Type_STANDARD DOMESTIC \	
0		-3.60
1		-3.60
2		-3.60
3		-3.60
4		-3.60

```

    Most_Frequent_Order_Type_STANDARD INTERNATIONAL \
0                           -0.06
1                           -0.06
2                           -0.06
3                           -0.06
4                           -0.06

    Most_Frequent_Order_Type_US FULL CARE EQPT DOMESTIC \
0                           -0.02
1                           -0.02
2                           -0.02
3                           -0.02
4                           -0.02

    Most_Frequent_Order_Type_US FULL CARE INTERNATIONAL \
0                           -0.00
1                           -0.00
2                           -0.00
3                           -0.00
4                           -0.00

    Most_Frequent_Order_Type_WEB ORDER
0                           -0.19
1                           -0.19
2                           -0.19
3                           -0.19
4                           -0.19

```

[26]: cij_final = pd.concat([cij_scaled,cijChurned],axis=1,sort=False)

[27]: cij_final.head()

	Site_Level_Price_Index	TRX_AMT_USD	Margin	QUANTITY	Total_SVC_Incidents
0	1.07	-0.23	-0.28	0.36	0.01
1	1.07	-0.22	-0.27	0.36	0.01
2	1.07	-0.38	-0.39	-0.28	0.01
3	1.07	-0.40	-0.40	-0.37	0.01
4	1.07	-0.41	-0.41	-0.41	0.01

	Total_Repeat_Calls	Total_FTF_Calls	Total_Visits	Total_Cases
0	-0.12	0.07	0.00	1.37
1	-0.12	0.07	0.00	1.37
2	-0.12	0.07	0.00	1.37
3	-0.12	0.07	0.00	1.37
4	-0.12	0.07	0.00	1.37

	Num_of_Active_Install_Bases	Total_Contracts	Contract_length
--	-----------------------------	-----------------	-----------------

0	-0.21	-0.17	0.47
1	-0.21	-0.17	0.47
2	-0.21	-0.17	0.47
3	-0.21	-0.17	0.47
4	-0.21	-0.17	0.47
0	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	Frequency
1	-0.12	-0.49	-0.36
2	-0.12	-0.49	-0.36
3	-0.12	-0.49	-0.36
4	-0.12	-0.49	-0.36
0	Avg_Margin	Avg_Quantity	Types_of_Product_Family
1	-0.39	-0.05	-0.61
2	-0.39	-0.05	-0.61
3	-0.39	-0.05	-0.61
4	-0.39	-0.05	-0.61
0	Avg_Price_Index	Tenure	SHORT_VERTICAL_BAKED GOODS & CEREALS
1	1.21	0.36	-0.19
2	1.21	0.36	-0.19
3	1.21	0.36	-0.19
4	1.21	0.36	-0.19
0	SHORT_VERTICAL_BEVERAGE	SHORT_VERTICAL_BUILDING MATERIALS	\
1	-0.32	-0.21	
2	-0.32	-0.21	
3	-0.32	-0.21	
4	-0.32	-0.21	
0	SHORT_VERTICAL_CANDY & CONFECTION	SHORT_VERTICAL_CHEMICALS	\
1	-0.11	5.25	
2	-0.11	5.25	
3	-0.11	5.25	
4	-0.11	5.25	
0	SHORT_VERTICAL_COSMETICS / PERSONAL CARE	SHORT_VERTICAL_DAIRY & EGGS	\
1	-0.22	-0.28	
2	-0.22	-0.28	
3	-0.22	-0.28	
4	-0.22	-0.28	

	SHORT_VERTICAL_DISTRIBUTOR	SHORT_VERTICAL_ELECTRICAL / ELECTRONICS \	
0	-0.04	-0.16	
1	-0.04	-0.16	
2	-0.04	-0.16	
3	-0.04	-0.16	
4	-0.04	-0.16	
	SHORT_VERTICAL_EXTRUSION / WIRE & CABLE	SHORT_VERTICAL_FISH & SEAFOOD \	
0	-0.16	-0.06	
1	-0.16	-0.06	
2	-0.16	-0.06	
3	-0.16	-0.06	
4	-0.16	-0.06	
	SHORT_VERTICAL_FROZEN PREPARED MEALS	SHORT_VERTICAL_FRUIT & VEGETABLE \	
0	-0.09	-0.21	
1	-0.09	-0.21	
2	-0.09	-0.21	
3	-0.09	-0.21	
4	-0.09	-0.21	
	SHORT_VERTICAL_GRAPHICS	SHORT_VERTICAL_INDUSTRIAL EQUIPMENT \	
0	-0.27	-0.19	
1	-0.27	-0.19	
2	-0.27	-0.19	
3	-0.27	-0.19	
4	-0.27	-0.19	
	SHORT_VERTICAL_MEAT & POULTRY	SHORT_VERTICAL_OEM-INDUSTRIAL EQUIPMENT \	
0	-0.22	-0.07	
1	-0.22	-0.07	
2	-0.22	-0.07	
3	-0.22	-0.07	
4	-0.22	-0.07	
	SHORT_VERTICAL_OTHER	SHORT_VERTICAL_OTHER FOOD \	
0	-0.33	-0.33	
1	-0.33	-0.33	
2	-0.33	-0.33	
3	-0.33	-0.33	
4	-0.33	-0.33	
	SHORT_VERTICAL_PACKAGING MATERIALS	SHORT_VERTICAL_PET FOOD & ANIMAL FEED \	
0	-0.11	-0.12	
1	-0.11	-0.12	
2	-0.11	-0.12	

3	-0.11	-0.12		
4	-0.11	-0.12		
0	SHORT_VERTICAL_PHARMA & MEDICAL	SHORT_VERTICAL_POSTAL \		
1	-0.26	-0.02		
2	-0.26	-0.02		
3	-0.26	-0.02		
4	-0.26	-0.02		
0	SHORT_VERTICAL_SALTY SNACKS	SHORT_VERTICAL_TEXTILE \		
1	-0.08	-0.08		
2	-0.08	-0.08		
3	-0.08	-0.08		
4	-0.08	-0.08		
0	SHORT_VERTICAL_TOBACCO	SHORT_VERTICAL_UNKNOWN	CUSTOMER_CLASS_END	USER \
1	-0.05	-0.18	0.11	
2	-0.05	-0.18	0.11	
3	-0.05	-0.18	0.11	
4	-0.05	-0.18	0.11	
0	CUSTOMER_CLASS_OEM	TERRITORY_TYPE_Industrial	TERRITORY_TYPE_Postal	\
1	-0.09	0.16	-0.05	
2	-0.09	0.16	-0.05	
3	-0.09	0.16	-0.05	
4	-0.09	0.16	-0.05	
0	SUPPLIES_SEGMENTATION_M	SUPPLIES_SEGMENTATION_S	\	
1	-0.43	1.15		
2	-0.43	1.15		
3	-0.43	1.15		
4	-0.43	1.15		
0	SUPPLIES_SEGMENTATION_Unclass	SUPPLIES_SEGMENTATION_XL	\	
1	-0.01	-0.37		
2	-0.01	-0.37		
3	-0.01	-0.37		
4	-0.01	-0.37		
0	SUPPLIES_DECLINE_REASON_Financial Distress/Credit Hold	\		
	-0.02			

1		-0.02
2		-0.02
3		-0.02
4		-0.02
	SUPPLIES_DECLINE_REASONs_Migration to 1000 Line/TIJ/TTO/LCM/LPA \	
0		-0.33
1		-0.33
2		-0.33
3		-0.33
4		-0.33
	SUPPLIES_DECLINE_REASONs_Migration to Lasers \	
0		-0.09
1		-0.09
2		-0.09
3		-0.09
4		-0.09
	SUPPLIES_DECLINE_REASONs_Moved Equipment \	
0		-0.06
1		-0.06
2		-0.06
3		-0.06
4		-0.06
	SUPPLIES_DECLINE_REASONs_No More Coding Requirement \	
0		-0.05
1		-0.05
2		-0.05
3		-0.05
4		-0.05
	SUPPLIES_DECLINE_REASONs_None SUPPLIES_DECLINE_REASONs_Off Brand \	
0	0.76	-0.14
1	0.76	-0.14
2	0.76	-0.14
3	0.76	-0.14
4	0.76	-0.14
	SUPPLIES_DECLINE_REASONs_Over Stocked / Timing \	
0		-0.29
1		-0.29
2		-0.29
3		-0.29
4		-0.29

```

SUPPLIES_DECLINE_REASONSPricing / Discounting \
0 -0.01
1 -0.01
2 -0.01
3 -0.01
4 -0.01

SUPPLIES_DECLINE_REASONSPrinting/EQ downtime Issues \
0 -0.03
1 -0.03
2 -0.03
3 -0.03
4 -0.03

SUPPLIES_DECLINE_REASONSProduction / Code Reduction \
0 -0.23
1 -0.23
2 -0.23
3 -0.23
4 -0.23

SUPPLIES_DECLINE_REASONSProduction Down (timing) \
0 -0.15
1 -0.15
2 -0.15
3 -0.15
4 -0.15

SUPPLIES_DECLINE_REASONSProject Based \
0 -0.04
1 -0.04
2 -0.04
3 -0.04
4 -0.04

SUPPLIES_DECLINE_REASONSPrecent Regain/Win-back \
0 -0.09
1 -0.09
2 -0.09
3 -0.09
4 -0.09

SUPPLIES_DECLINE_REASONSSeasonal Producer \
0 -0.08
1 -0.08
2 -0.08
3 -0.08

```

4		-0.08		
	SUPPLIES_DECLINE_REASON_Served by Authorized Distributor \			
0		-0.09		
1		-0.09		
2		-0.09		
3		-0.09		
4		-0.09		
	SUPPLIES_DECLINE_REASON_Site Closed \			
0		-0.11		
1		-0.11		
2		-0.11		
3		-0.11		
4		-0.11		
	SUPPLIES_DECLINE_REASON_VJ Operations Issues SALES_CHANNEL_Copy \			
0		-0.02		-0.17
1		-0.02		-0.17
2		-0.02		-0.17
3		-0.02		-0.17
4		-0.02		-0.17
	SALES_CHANNEL_ECOMM PO IMPORT SALES_CHANNEL_EDI SALES_CHANNEL_Esker \			
0		-0.01	9.07	-0.58
1		-0.01	9.07	-0.58
2		-0.01	-0.11	-0.58
3		-0.01	-0.11	-0.58
4		-0.01	-0.11	-0.58
	SALES_CHANNEL_IStore Account SALES_CHANNEL_OCC SALES_CHANNEL_Online \			
0		-0.13	-0.23	-1.22
1		-0.13	-0.23	-1.22
2		-0.13	-0.23	0.82
3		-0.13	-0.23	0.82
4		-0.13	-0.23	-1.22
	SALES_CHANNEL_SFDC_CPQ SALES_CHANNEL_Service Billing PRODUCT_FAMILY_CIJ \			
0		-0.09	0.00	0.00
1		-0.09	0.00	0.00
2		-0.09	0.00	0.00
3		-0.09	0.00	0.00
4		-0.09	0.00	0.00
	PRODUCT_FAMILY_GRAPHICS PRODUCT_FAMILY_GRAPHICS BA PRODUCT_FAMILY_LASER \			
0		0.00	0.00	0.00
1		0.00	0.00	0.00

2	0.00	0.00	0.00	
3	0.00	0.00	0.00	
4	0.00	0.00	0.00	
	PRODUCT_FAMILY_LCM	PRODUCT_FAMILY_LPA	PRODUCT_FAMILY_RAW MATERIAL	\
0	0.00	0.00	0.00	
1	0.00	0.00	0.00	
2	0.00	0.00	0.00	
3	0.00	0.00	0.00	
4	0.00	0.00	0.00	
	PRODUCT_FAMILY_TIJ	PRODUCT_FAMILY_TTO	PRODUCT_MODEL_CLEANING SOLUTION	\
0	0.00	0.00	-0.46	
1	0.00	0.00	-0.46	
2	0.00	0.00	-0.46	
3	0.00	0.00	-0.46	
4	0.00	0.00	-0.46	
	PRODUCT_MODEL_FUME EXTRACTION	PRODUCT_MODEL_INK	PRODUCT_MODEL_LABELS	\
0	0.00	-0.65	0.00	
1	0.00	-0.65	0.00	
2	0.00	-0.65	0.00	
3	0.00	-0.65	0.00	
4	0.00	-0.65	0.00	
	PRODUCT_MODEL_MAKE-UP	PRODUCT_MODEL_PACKAGING	PRODUCT_MODEL_RIBBONS	\
0	0.98	0.00	0.00	
1	0.98	0.00	0.00	
2	0.98	0.00	0.00	
3	0.98	0.00	0.00	
4	0.98	0.00	0.00	
	PRODUCT_MODEL_SOLVENT	PRODUCT_MODEL_VALUE PACK	\	
0	0.00	-0.15		
1	0.00	-0.15		
2	0.00	-0.15		
3	0.00	-0.15		
4	0.00	-0.15		
	Most_Frequent_Interaction_Type_Callback	\		
0		-0.01		
1		-0.01		
2		-0.01		
3		-0.01		
4		-0.01		
	Most_Frequent_Interaction_Type_Contact Customer	\		

0		-0.01
1		-0.01
2		-0.01
3		-0.01
4		-0.01
	Most_Frequent_Interaction_Type_Customer Meeting \	
0		-0.04
1		-0.04
2		-0.04
3		-0.04
4		-0.04
	Most_Frequent_Interaction_Type_Dial Most_Frequent_Interaction_Type_Email \	
0	-0.01	-0.34
1	-0.01	-0.34
2	-0.01	-0.34
3	-0.01	-0.34
4	-0.01	-0.34
	Most_Frequent_Interaction_Type_Make Qualified Sales Call \	
0	0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
	Most_Frequent_Interaction_Type_Meeting \	
0	-0.08	
1	-0.08	
2	-0.08	
3	-0.08	
4	-0.08	
	Most_Frequent_Interaction_Type_Other \	
0	-0.28	
1	-0.28	
2	-0.28	
3	-0.28	
4	-0.28	
	Most_Frequent_Interaction_Type_TS Task \	
0	-0.01	
1	-0.01	
2	-0.01	
3	-0.01	
4	-0.01	

	Max_Case_Origin_CX Survey Detractor	Max_Case_Origin_Email \
0	-0.04	-0.02
1	-0.04	-0.02
2	-0.04	-0.02
3	-0.04	-0.02
4	-0.04	-0.02
	Max_Case_Origin_Email - VTI CC Sales Escalations \	
0	-0.00	
1	-0.00	
2	-0.00	
3	-0.00	
4	-0.00	
	Max_Case_Origin_Email - VTI NACC	Max_Case_Origin_Email/Fax - VTI CS \
0	-0.42	-0.05
1	-0.42	-0.05
2	-0.42	-0.05
3	-0.42	-0.05
4	-0.42	-0.05
	Max_Case_Origin_FS Survey Followup	Max_Case_Origin_Install Complete \
0	-0.07	-0.04
1	-0.07	-0.04
2	-0.07	-0.04
3	-0.07	-0.04
4	-0.07	-0.04
	Max_Case_Origin_Phone	Max_Case_Origin_TS Survey Followup \
0	-0.34	-0.12
1	-0.34	-0.12
2	-0.34	-0.12
3	-0.34	-0.12
4	-0.34	-0.12
	Max_Case_Origin_unknown	Max_Case_Reason_CX: Customer Care \
0	0.68	-0.18
1	0.68	-0.18
2	0.68	-0.18
3	0.68	-0.18
4	0.68	-0.18
	Max_Case_Reason_CX: Field Sales	Max_Case_Reason_CX: Field Service \
0	-0.03	-0.09
1	-0.03	-0.09
2	-0.03	-0.09

3	-0.03	-0.09	
4	-0.03	-0.09	
		\	
0	-0.02	-0.03	
1	-0.02	-0.03	
2	-0.02	-0.03	
3	-0.02	-0.03	
4	-0.02	-0.03	
		\	
0	-0.13	-0.59	
1	-0.13	-0.59	
2	-0.13	-0.59	
3	-0.13	-0.59	
4	-0.13	-0.59	
		\	
0	0.68	-0.28	
1	0.68	-0.28	
2	0.68	-0.28	
3	0.68	-0.28	
4	0.68	-0.28	
		\	
0	-0.60	-0.04	
1	-0.60	-0.04	
2	-0.60	-0.04	
3	-0.60	-0.04	
4	-0.60	-0.04	
		\	
0	-0.02	-0.41	-0.52
1	-0.02	-0.41	-0.52
2	-0.02	-0.41	-0.52
3	-0.02	-0.41	-0.52
4	-0.02	-0.41	-0.52
			\
0	-0.49	-0.37	-0.43
1	-0.49	-0.37	-0.43
2	-0.49	-0.37	-0.43
3	-0.49	-0.37	-0.43
4	-0.49	-0.37	-0.43
			\
0	Most_Frequent_Sales_Channel_Copy	Most_Frequent_Sales_Channel_EDI	\
	-0.04	7.32	

1	-0.04	7.32
2	-0.04	7.32
3	-0.04	7.32
4	-0.04	7.32
	Most_Frequent_Sales_Channel_Esker \	
0	-0.65	
1	-0.65	
2	-0.65	
3	-0.65	
4	-0.65	
	Most_Frequent_Sales_Channel_IStore Account \	
0	-0.13	
1	-0.13	
2	-0.13	
3	-0.13	
4	-0.13	
	Most_Frequent_Sales_Channel_OCC Most_Frequent_Sales_Channel_Online \	
0	-0.14	-1.31
1	-0.14	-1.31
2	-0.14	-1.31
3	-0.14	-1.31
4	-0.14	-1.31
	Most_Frequent_Sales_Channel_SFDC_CPQ \	
0	-0.05	
1	-0.05	
2	-0.05	
3	-0.05	
4	-0.05	
	Most_Frequent_Sales_Channel_Service Billing \	
0	0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
	Most_Frequent_Order_Type_BILL ONLY \	
0	-0.01	
1	-0.01	
2	-0.01	
3	-0.01	
4	-0.01	

```

    Most_Frequent_Order_Type_DEMO EQUIPMENT ACCEPT \
0                                -0.01
1                                -0.01
2                                -0.01
3                                -0.01
4                                -0.01

    Most_Frequent_Order_Type_EDI  Most_Frequent_Order_Type_EQUIPMENT DOMESTIC \
0                                7.39                               -0.10
1                                7.39                               -0.10
2                                7.39                               -0.10
3                                7.39                               -0.10
4                                7.39                               -0.10

    Most_Frequent_Order_Type_SERVICE \
0                                0.00
1                                0.00
2                                0.00
3                                0.00
4                                0.00

    Most_Frequent_Order_Type_STANDARD DOMESTIC \
0                                -3.60
1                                -3.60
2                                -3.60
3                                -3.60
4                                -3.60

    Most_Frequent_Order_Type_STANDARD INTERNATIONAL \
0                                -0.06
1                                -0.06
2                                -0.06
3                                -0.06
4                                -0.06

    Most_Frequent_Order_Type_US FULL CARE EQPT DOMESTIC \
0                                -0.02
1                                -0.02
2                                -0.02
3                                -0.02
4                                -0.02

    Most_Frequent_Order_Type_US FULL CARE INTERNATIONAL \
0                                -0.00
1                                -0.00
2                                -0.00
3                                -0.00

```

4

-0.00

	Most_Frequent_Order_Type_WEB ORDER	Churned_365
0	-0.19	0
1	-0.19	0
2	-0.19	0
3	-0.19	0
4	-0.19	0

```
[28]: cij_final = cij_final.drop(columns=['SALES_CHANNEL_Service Billing',  
→'PRODUCT_FAMILY_CIJ', 'PRODUCT_FAMILY_GRAPHICS', 'PRODUCT_FAMILY_GRAPHICS  
→BA', 'PRODUCT_FAMILY_LASER', 'PRODUCT_FAMILY_LCM', 'PRODUCT_FAMILY_LPA',  
→'PRODUCT_FAMILY_RAW MATERIAL', 'PRODUCT_FAMILY_TIJ', 'PRODUCT_FAMILY_TTO',  
→'PRODUCT_MODEL_FUME EXTRACTION', 'PRODUCT_MODEL_LABELS',  
→'PRODUCT_MODEL_PACKAGING', 'PRODUCT_MODEL_RIBBONS', 'PRODUCT_MODEL_SOLVENT',  
→'Most_Frequent_Interaction_Type_Make Qualified Sales Call',  
→'Most_Frequent_Sales_Channel_Service Billing',  
→'Most_Frequent_Order_Type_SERVICE'],axis=1)
```

```
[29]: cij_final.shape
```

[29]: (196766, 139)

```
[30]: cph = lifelines.CoxPHFitter(penalizer=0.00001)  
cph.fit(cij_final, step_size=0.05, duration_col='Tenure',  
→event_col='Churned_365', show_progress=False)  
cph.print_summary()
```

covariate	coef	exp(coef)	se(coef)	coef lower 95%
Site_Level_Price_Index	0.00	1.00	0.01	-0.05
TRX_AMT_USD	0.36	1.43	0.05	0.20
Margin	-0.40	0.67	0.06	-0.51
QUANTITY	0.01	1.01	0.01	-0.02
Total_SVC_Incidents	0.11	1.11	0.56	-1.00
Total_Repeat_Calls	0.23	1.26	0.20	-0.10
Total_FTF_Calls	0.04	1.04	0.39	-0.73
Total_Visits	-0.89	0.41	0.03	-0.93
Total_Cases	0.21	1.23	0.01	0.18
Num_of_Active_Install_Bases	-0.11	0.90	0.03	-0.10
Total_Contracts	-0.07	0.93	0.03	-0.13
Contract_length	-0.44	0.65	0.02	-0.47
Num_of_Inactive_Install_Bases	-0.07	0.93	0.01	-0.10
STRATEGIC_ACCOUNTS	0.18	1.20	0.01	0.10
Frequency	-0.40	0.67	0.01	-0.41
Num_of_Trxns	-2.06	0.13	0.05	-2.13
Avg_Margin	0.12	1.13	0.02	0.09
Avg_Quantity	-0.00	1.00	0.02	-0.04
Types_of_Product_Family	0.23	1.26	0.01	0.20
Types_of_Product_Model	-0.45	0.64	0.01	-0.47
Avg_Price_Index	-0.01	0.99	0.01	-0.04
SHORT_VERTICAL_BAKED GOODS & CEREALS	0.06	1.07	0.01	0.05
SHORT_VERTICAL_BEVERAGE	-0.01	0.99	0.01	-0.03
SHORT_VERTICAL_BUILDING MATERIALS	0.05	1.06	0.01	0.04
SHORT_VERTICAL_CANDY & CONFECTION	-0.03	0.97	0.01	-0.05
SHORT_VERTICAL_CHEMICALS	-0.01	0.99	0.01	-0.02
SHORT_VERTICAL_COSMETICS / PERSONAL CARE	0.02	1.02	0.01	-0.00
SHORT_VERTICAL_DAIRY & EGGS	0.01	1.01	0.01	-0.01
SHORT_VERTICAL_DISTRIBUTOR	-0.21	0.81	0.02	-0.24
SHORT_VERTICAL_ELECTRICAL / ELECTRONICS	0.02	1.02	0.01	0.00
SHORT_VERTICAL_EXTRUSION / WIRE & CABLE	0.03	1.03	0.01	0.01
SHORT_VERTICAL_FISH & SEAFOOD	0.01	1.01	0.01	0.00
SHORT_VERTICAL_FROZEN PREPARED MEALS	0.03	1.03	0.01	0.02
SHORT_VERTICAL_FRUIT & VEGETABLE	-0.08	0.92	0.01	-0.10
SHORT_VERTICAL_GRAPHICS	0.05	1.05	0.01	0.03
SHORT_VERTICAL_INDUSTRIAL EQUIPMENT	-0.00	1.00	0.01	-0.02
SHORT_VERTICAL_MEAT & POULTRY	0.01	1.01	0.01	-0.01
SHORT_VERTICAL_OEM-INDUSTRIAL EQUIPMENT	0.02	1.02	0.01	0.00
SHORT_VERTICAL_OTHER	-0.02	0.98	0.01	-0.04
SHORT_VERTICAL_OTHER FOOD	-0.02	0.98	0.01	-0.05
SHORT_VERTICAL_PACKAGING MATERIALS	0.06	1.06	0.01	0.04
SHORT_VERTICAL_PET FOOD & ANIMAL FEED	0.01	1.01	0.01	-0.00
SHORT_VERTICAL_PHARMA & MEDICAL	0.03	1.03	0.01	0.01
SHORT_VERTICAL_POSTAL	-0.29	0.75	0.19	-0.67
SHORT_VERTICAL_SALTY SNACKS	0.04	1.04	0.01	0.03
SHORT_VERTICAL_TEXTILE	-0.01	0.99	0.01	-0.03
SHORT_VERTICAL_TOBACCO	0.02	1.02	0.01	0.01
SHORT_VERTICAL_UNKNOWN	0.09	1.10	0.01	0.08
CUSTOMER_CLASS_END USER	-0.60	0.55	0.01	-0.63
CUSTOMER_CLASS_OEM	-0.50	0.61	0.01	-0.55
TERRITORY_TYPE_Industrial	0.03	1.04	0.01	0.02

```
[31]: cij_final_table = cph.summary[cph.summary['p']<0.005].
    ↪sort_values(by='exp(coef)', ascending=False).head(10)
```

2 Model building for TTO

```
[33]: tto.reset_index(inplace=True, drop=True)
```

```
[34]: tto.head()
```

	Site_Level_Price_Index	TRX_AMT_USD	Margin	QUANTITY	\
0	0.73	1,492.40	1,196.86	5	
1	0.73	596.96	478.74	2	
2	0.73	298.48	239.37	1	
3	0.74	2,984.80	2,396.56	10	
4	2.19	235.41	169.06	1	

	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	Total_Visits	\
0	57.00	14.00	43.00	70.00	
1	57.00	14.00	43.00	70.00	
2	57.00	14.00	43.00	70.00	
3	57.00	14.00	43.00	70.00	
4	57.00	14.00	43.00	70.00	

	Total_Cases	Num_of_Active_Install_Bases	Total_Contracts	Contract_length	\
0	1.00	15.00	11.00	521.64	
1	1.00	15.00	11.00	521.64	
2	1.00	15.00	11.00	521.64	
3	1.00	15.00	11.00	521.64	
4	1.00	15.00	11.00	521.64	

	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	Frequency	Num_of_Trxns	\
0	0.00	0	11.38	174	
1	0.00	0	11.38	174	
2	0.00	0	11.38	174	
3	0.00	0	11.38	174	
4	0.00	0	11.38	174	

	Avg_Margin	Avg_Quantity	Types_of_Product_Family	Types_of_Product_Model	\
0	103.28	4.43	3	4	
1	103.28	4.43	3	4	
2	103.28	4.43	3	4	
3	103.28	4.43	3	4	
4	103.28	4.43	3	4	

	Avg_Price_Index	Tenure	Churned_365	\
0	1.25	1,974.00	0	

1	1.25	1,974.00	0
2	1.25	1,974.00	0
3	1.25	1,974.00	0
4	1.25	1,974.00	0
0	SHORT_VERTICAL_BAKED GOODS & CEREALS	SHORT_VERTICAL_BEVERAGE	\
1	0	0	
2	0	0	
3	0	0	
4	0	0	
0	SHORT_VERTICAL_BUILDING MATERIALS	SHORT_VERTICAL_CANDY & CONFECTION	\
1	0	0	
2	0	0	
3	0	0	
4	0	0	
0	SHORT_VERTICAL_CHEMICALS	SHORT_VERTICAL_COSMETICS / PERSONAL CARE	\
1	0	0	
2	0	0	
3	0	0	
4	0	0	
0	SHORT_VERTICAL_DAIRY & EGGS	SHORT_VERTICAL_DISTRIBUTOR	\
1	0	0	
2	0	0	
3	0	0	
4	0	0	
0	SHORT_VERTICAL_ELECTRICAL / ELECTRONICS	\	
1	0		
2	0		
3	0		
4	0		
0	SHORT_VERTICAL_EXTRUSION / WIRE & CABLE	SHORT_VERTICAL_FISH & SEAFOOD	\
1	0	0	
2	0	0	
3	0	0	
4	0	0	

	SHORT_VERTICAL_FROZEN PREPARED MEALS	SHORT_VERTICAL_FRUIT & VEGETABLE	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_GRAPHICS	SHORT_VERTICAL_INDUSTRIAL EQUIPMENT	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_MEAT & POULTRY	SHORT_VERTICAL_OEM-INDUSTRIAL EQUIPMENT	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_OTHER	SHORT_VERTICAL_OTHER FOOD	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_PACKAGING MATERIALS	SHORT_VERTICAL_PET FOOD & ANIMAL FEED	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_PHARMA & MEDICAL	SHORT_VERTICAL_POSTAL	\
0	1	0	
1	1	0	
2	1	0	
3	1	0	
4	1	0	
	SHORT_VERTICAL_SALTY SNACKS	SHORT_VERTICAL_TEXTILE	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	

4	0	0	
0	0	0	1
1	0	0	1
2	0	0	1
3	0	0	1
4	0	0	1
CUSTOMER_CLASS_OEM	TERRITORY_TYPE_Industrial	TERRITORY_TYPE_Postal	\
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
SUPPLIES_SEGMENTATION_M	SUPPLIES_SEGMENTATION_S	\	
0	1	0	
1	1	0	
2	1	0	
3	1	0	
4	1	0	
SUPPLIES_SEGMENTATION_Unclass	SUPPLIES_SEGMENTATION_XL	\	
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
SUPPLIES_DECLINE_REASON_Financial Distress/Credit Hold	\		
0	0		
1	0		
2	0		
3	0		
4	0		
SUPPLIES_DECLINE_REASON_Migration to 1000 Line/TIJ/TTO/LCM/LPA	\		
0	0		
1	0		
2	0		
3	0		
4	0		
SUPPLIES_DECLINE_REASON_Migration to Lasers	\		
0	0		
1	0		

2		0
3		0
4		0
SUPPLIES_DECLINE_REASONs_Moved Equipment \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_No More Coding Requirement \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_None SUPPLIES_DECLINE_REASONs_Off Brand \		
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0
SUPPLIES_DECLINE_REASONs_Over Stocked / Timing \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_Pricing / Discounting \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_Printing/EQ downtime Issues \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_Production / Code Reduction \		

0	0
1	0
2	0
3	0
4	0
SUPPLIES_DECLINE_REASON_Production Down (timing) \	
0	0
1	0
2	0
3	0
4	0
SUPPLIES_DECLINE_REASON_Project Based \	
0	0
1	0
2	0
3	0
4	0
SUPPLIES_DECLINE_REASON_Recent Regain/Win-back \	
0	0
1	0
2	0
3	0
4	0
SUPPLIES_DECLINE_REASON_Seasonal Producer \	
0	0
1	0
2	0
3	0
4	0
SUPPLIES_DECLINE_REASON_Served by Authorized Distributor \	
0	0
1	0
2	0
3	0
4	0
SUPPLIES_DECLINE_REASON_Site Closed \	
0	0
1	0
2	0
3	0
4	0

SUPPLIES_DECLINE_REASON_VJ	Operations	Issues	SALES_CHANNEL_Copy	\
0		0	0	
1		0	0	
2		0	0	
3		0	0	
4		0	0	
SALES_CHANNEL_ECOMM	PO IMPORT	SALES_CHANNEL_EDI	SALES_CHANNEL_Esker	\
0	0	1	0	
1	0	1	0	
2	0	1	0	
3	0	1	0	
4	0	0	0	
SALES_CHANNEL_IStore	Account	SALES_CHANNEL_OCC	SALES_CHANNEL_Online	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
SALES_CHANNEL_SFDC_CPQ	SALES_CHANNEL_Service	Billing	PRODUCT_FAMILY_CIJ	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
PRODUCT_FAMILY_GRAPHICS	PRODUCT_FAMILY_GRAPHICS	BA	PRODUCT_FAMILY_LASER	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
PRODUCT_FAMILY_LCM	PRODUCT_FAMILY_LPA	PRODUCT_FAMILY_RAW MATERIAL		\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
PRODUCT_FAMILY_TIJ	PRODUCT_FAMILY_TTO	PRODUCT_MODEL_CLEANING SOLUTION		\
0	0	1	0	
1	0	1	0	
2	0	1	0	

3	0	1	0	
4	0	1	0	
0	PRODUCT_MODEL_FUME_EXTRACTION	PRODUCT_MODEL_INK	PRODUCT_MODEL_LABELS	\
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
0	PRODUCT_MODEL_MAKE-UP	PRODUCT_MODEL_PACKAGING	PRODUCT_MODEL_RIBBONS	\
1	0	0	1	
2	0	0	1	
3	0	0	1	
4	0	0	1	
0	PRODUCT_MODEL_SOLVENT	PRODUCT_MODEL_VALUE_PACK	\	
1	0	0		
2	0	0		
3	0	0		
4	0	0		
0	Most_Frequent_Interaction_Type_Callback	\		
1	0			
2	0			
3	0			
4	0			
0	Most_Frequent_Interaction_Type_Contact_Customer	\		
1	0			
2	0			
3	0			
4	0			
0	Most_Frequent_Interaction_Type_Customer_Meeting	\		
1	0			
2	0			
3	0			
4	0			
0	Most_Frequent_Interaction_Type_Dial	Most_Frequent_Interaction_Type_Email	\	
	0	0		

1	0	0
2	0	0
3	0	0
4	0	0
0	Most_Frequent_Interaction_Type_Make Qualified Sales Call \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Most_Frequent_Interaction_Type_Meeting \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Most_Frequent_Interaction_Type_Other \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Most_Frequent_Interaction_Type_TS Task \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Max_Case_Origin_CX Survey Detractor \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Max_Case_Origin_Email - VTI CC Sales Escalations \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0

	Max_Case_Origin_Email - VTI NACC	Max_Case_Origin_Email/Fax - VTI CS \
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0
	Max_Case_Origin_FS Survey Followup	Max_Case_Origin_Install Complete \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Origin_Phone	Max_Case_Origin_TS Survey Followup \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Origin_unknown	Max_Case_Reason_CX: Customer Care \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Reason_CX: Field Sales	Max_Case_Reason_CX: Field Service \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Reason_CX: Manufacturing	Max_Case_Reason_CX: Other Team \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Reason_CX: Tech Support	Max_Case_Reason_Customer Experience \
0	0	1
1	0	1
2	0	1
3	0	1

4

0

1

	Max_Case_Reason_unknown	Contract_Category_Full	Care	\
0	0		0	
1	0		0	
2	0		0	
3	0		0	
4	0		0	

	Contract_Category_No	Contract	Contract_Category_Supportive	\
0		0		0
1		0		0
2		0		0
3		0		0
4		0		0

	Contract_Category_WFC	TERRITORY_REGION_MW	TERRITORY_REGION_NE	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	TERRITORY_REGION_NW	TERRITORY_REGION_SC	TERRITORY_REGION_SE	\
0	1	0	0	
1	1	0	0	
2	1	0	0	
3	1	0	0	
4	1	0	0	

	Most_Frequent_Sales_Channel_Copy	Most_Frequent_Sales_Channel_EDI	\
0	0	1	
1	0	1	
2	0	1	
3	0	1	
4	0	1	

	Most_Frequent_Sales_Channel_Esker	\
0	0	
1	0	
2	0	
3	0	
4	0	

	Most_Frequent_Sales_Channel_IStore	Account	\
0		0	
1		0	

2		0
3		0
4		0
	Most_Frequent_Sales_Channel_OCC	Most_Frequent_Sales_Channel_Online \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Most_Frequent_Sales_Channel_SFDC_CPQ \	
0	0	
1	0	
2	0	
3	0	
4	0	
	Most_Frequent_Sales_Channel_Service Billing \	
0		0
1		0
2		0
3		0
4		0
	Most_Frequent_Order_Type_BILL ONLY \	
0	0	
1	0	
2	0	
3	0	
4	0	
	Most_Frequent_Order_Type_DEMO EQUIPMENT ACCEPT \	
0		0
1		0
2		0
3		0
4		0
	Most_Frequent_Order_Type_EDI	Most_Frequent_Order_Type_EQUIPMENT DOMESTIC \
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0
	Most_Frequent_Order_Type_SERVICE \	

```

0          0
1          0
2          0
3          0
4          0

Most_Frequent_Order_Type_STANDARD DOMESTIC \
0          0
1          0
2          0
3          0
4          0

Most_Frequent_Order_Type_STANDARD INTERNATIONAL \
0          0
1          0
2          0
3          0
4          0

Most_Frequent_Order_Type_US FULL CARE EQPT DOMESTIC \
0          0
1          0
2          0
3          0
4          0

Most_Frequent_Order_Type_US FULL CARE INTERNATIONAL \
0          0
1          0
2          0
3          0
4          0

Most_Frequent_Order_Type_WEB ORDER
0          0
1          0
2          0
3          0
4          0

```

[35]: tto.shape

[35]: (12346, 157)

[36]: ttoChurned = tto[['Churned_365']]

```
[37]: tto.drop(columns=['Churned_365'],axis=1,inplace=True)

[38]: #Scaling
# Instantiate
scaler = StandardScaler()

# fit_transform
tto_scaled = scaler.fit_transform(tto)

[39]: tto_scaled = pd.DataFrame(tto_scaled)
tto_scaled.columns = tto.columns
tto_scaled.head()

[39]:   Site_Level_Price_Index  TRX_AMT_USD  Margin  QUANTITY  Total_SVC_Incidents \
0           -0.67          -0.21  -0.07     -0.17        0.06
1           -0.67          -0.53  -0.45     -0.40        0.06
2           -0.67          -0.64  -0.58     -0.47        0.06
3           -0.64          0.34   0.58      0.20        0.06
4            3.91          -0.67  -0.62     -0.47        0.06

   Total_Repeat_Calls  Total_FTF_Calls  Total_Visits  Total_Cases \
0            0.03          0.07       -0.03      -1.16
1            0.03          0.07       -0.03      -1.16
2            0.03          0.07       -0.03      -1.16
3            0.03          0.07       -0.03      -1.16
4            0.03          0.07       -0.03      -1.16

   Num_of_Active_Install_Bases  Total_Contracts  Contract_length \
0             -0.08           0.11        -0.02
1             -0.08           0.11        -0.02
2             -0.08           0.11        -0.02
3             -0.08           0.11        -0.02
4             -0.08           0.11        -0.02

   Num_of_Inactive_Install_Bases  STRATEGIC_ACCOUNTS  Frequency  Num_of_Trxns \
0              -0.23           -0.53        -0.49        1.04
1              -0.23           -0.53        -0.49        1.04
2              -0.23           -0.53        -0.49        1.04
3              -0.23           -0.53        -0.49        1.04
4              -0.23           -0.53        -0.49        1.04

   Avg_Margin  Avg_Quantity  Types_of_Product_Family  Types_of_Product_Model \
0         -0.83        -0.50                  1.32          0.90
1         -0.83        -0.50                  1.32          0.90
2         -0.83        -0.50                  1.32          0.90
3         -0.83        -0.50                  1.32          0.90
4         -0.83        -0.50                  1.32          0.90
```

	Avg_Price_Index	Tenure	SHORT_VERTICAL_BAKED GOODS & CEREALS \	
0	1.28	0.66		-0.30
1	1.28	0.66		-0.30
2	1.28	0.66		-0.30
3	1.28	0.66		-0.30
4	1.28	0.66		-0.30
	SHORT_VERTICAL_BEVERAGE	SHORT_VERTICAL_BUILDING MATERIALS \		
0		-0.09		-0.16
1		-0.09		-0.16
2		-0.09		-0.16
3		-0.09		-0.16
4		-0.09		-0.16
	SHORT_VERTICAL_CANDY & CONFECTION	SHORT_VERTICAL_CHEMICALS \		
0		-0.24		-0.12
1		-0.24		-0.12
2		-0.24		-0.12
3		-0.24		-0.12
4		-0.24		-0.12
	SHORT_VERTICAL_COSMETICS / PERSONAL CARE	SHORT_VERTICAL_DAIRY & EGGS \		
0		-0.08		-0.20
1		-0.08		-0.20
2		-0.08		-0.20
3		-0.08		-0.20
4		-0.08		-0.20
	SHORT_VERTICAL_DISTRIBUTOR	SHORT_VERTICAL_ELECTRICAL / ELECTRONICS \		
0		-0.03		-0.02
1		-0.03		-0.02
2		-0.03		-0.02
3		-0.03		-0.02
4		-0.03		-0.02
	SHORT_VERTICAL_EXTRUSION / WIRE & CABLE	SHORT_VERTICAL_FISH & SEAFOOD \		
0		-0.07		-0.11
1		-0.07		-0.11
2		-0.07		-0.11
3		-0.07		-0.11
4		-0.07		-0.11
	SHORT_VERTICAL_FROZEN PREPARED MEALS	SHORT_VERTICAL_FRUIT & VEGETABLE \		
0		-0.14		-0.27
1		-0.14		-0.27
2		-0.14		-0.27

3		-0.14	-0.27	
4		-0.14	-0.27	
	SHORT_VERTICAL_GRAPHICS	SHORT_VERTICAL_INDUSTRIAL_EQUIPMENT	\	
0	-0.21	-0.09		
1	-0.21	-0.09		
2	-0.21	-0.09		
3	-0.21	-0.09		
4	-0.21	-0.09		
	SHORT_VERTICAL_MEAT & POULTRY	SHORT_VERTICAL_OEM-INDUSTRIAL_EQUIPMENT	\	
0	-0.30	-0.13		
1	-0.30	-0.13		
2	-0.30	-0.13		
3	-0.30	-0.13		
4	-0.30	-0.13		
	SHORT_VERTICAL_OTHER	SHORT_VERTICAL_OTHER_FOOD	\	
0	-0.24	-0.50		
1	-0.24	-0.50		
2	-0.24	-0.50		
3	-0.24	-0.50		
4	-0.24	-0.50		
	SHORT_VERTICAL_PACKAGING_MATERIALS	SHORT_VERTICAL_PET_FOOD & ANIMAL_FEED	\	
0	-0.06	-0.06		
1	-0.06	-0.06		
2	-0.06	-0.06		
3	-0.06	-0.06		
4	-0.06	-0.06		
	SHORT_VERTICAL_PHARMA & MEDICAL	SHORT_VERTICAL_POSTAL	\	
0	3.07	0.00		
1	3.07	0.00		
2	3.07	0.00		
3	3.07	0.00		
4	3.07	0.00		
	SHORT_VERTICAL_SALTY_SNACKS	SHORT_VERTICAL_TEXTILE	\	
0	-0.37	-0.03		
1	-0.37	-0.03		
2	-0.37	-0.03		
3	-0.37	-0.03		
4	-0.37	-0.03		
	SHORT_VERTICAL_TOBACCO	SHORT_VERTICAL_UNKNOWN	CUSTOMER_CLASS_END_USER	\
0	-0.03	-0.18	0.17	

1	-0.03	-0.18	0.17
2	-0.03	-0.18	0.17
3	-0.03	-0.18	0.17
4	-0.03	-0.18	0.17
CUSTOMER_CLASS_OEM TERRITORY_TYPE_Industrial TERRITORY_TYPE_Postal \			
0	-0.16	0.03	0.00
1	-0.16	0.03	0.00
2	-0.16	0.03	0.00
3	-0.16	0.03	0.00
4	-0.16	0.03	0.00
SUPPLIES_SEGMENTATION_M SUPPLIES_SEGMENTATION_S \			
0	2.69	-0.65	
1	2.69	-0.65	
2	2.69	-0.65	
3	2.69	-0.65	
4	2.69	-0.65	
SUPPLIES_SEGMENTATION_Unclass SUPPLIES_SEGMENTATION_XL \			
0	-0.02	-0.61	
1	-0.02	-0.61	
2	-0.02	-0.61	
3	-0.02	-0.61	
4	-0.02	-0.61	
SUPPLIES_DECLINE_REASON_Financial Distress/Credit Hold \			
0		0.00	
1		0.00	
2		0.00	
3		0.00	
4		0.00	
SUPPLIES_DECLINE_REASON_Migration to 1000 Line/TIJ/TTO/LCM/LPA \			
0		-0.12	
1		-0.12	
2		-0.12	
3		-0.12	
4		-0.12	
SUPPLIES_DECLINE_REASON_Migration to Lasers \			
0		-0.07	
1		-0.07	
2		-0.07	
3		-0.07	
4		-0.07	

SUPPLIES_DECLINE_REASONs_Moved Equipment \		
0	-0.08	
1	-0.08	
2	-0.08	
3	-0.08	
4	-0.08	
SUPPLIES_DECLINE_REASONs_No More Coding Requirement \		
0	-0.02	
1	-0.02	
2	-0.02	
3	-0.02	
4	-0.02	
SUPPLIES_DECLINE_REASONs_None SUPPLIES_DECLINE_REASONs_Off Brand \		
0	0.89	-0.33
1	0.89	-0.33
2	0.89	-0.33
3	0.89	-0.33
4	0.89	-0.33
SUPPLIES_DECLINE_REASONs_Over Stocked / Timing \		
0	-0.42	
1	-0.42	
2	-0.42	
3	-0.42	
4	-0.42	
SUPPLIES_DECLINE_REASONs_Pricing / Discounting \		
0	-0.04	
1	-0.04	
2	-0.04	
3	-0.04	
4	-0.04	
SUPPLIES_DECLINE_REASONs_Printing/EQ downtime Issues \		
0	-0.07	
1	-0.07	
2	-0.07	
3	-0.07	
4	-0.07	
SUPPLIES_DECLINE_REASONs_Production / Code Reduction \		
0	-0.23	
1	-0.23	
2	-0.23	
3	-0.23	

4		-0.23	
SUPPLIES_DECLINE_REASON_Production Down (timing) \			
0		-0.14	
1		-0.14	
2		-0.14	
3		-0.14	
4		-0.14	
SUPPLIES_DECLINE_REASON_Project Based \			
0		-0.04	
1		-0.04	
2		-0.04	
3		-0.04	
4		-0.04	
SUPPLIES_DECLINE_REASON_Recent Regain/Win-back \			
0		-0.16	
1		-0.16	
2		-0.16	
3		-0.16	
4		-0.16	
SUPPLIES_DECLINE_REASON_Seasonal Producer \			
0		-0.12	
1		-0.12	
2		-0.12	
3		-0.12	
4		-0.12	
SUPPLIES_DECLINE_REASON_Served by Authorized Distributor \			
0		-0.08	
1		-0.08	
2		-0.08	
3		-0.08	
4		-0.08	
SUPPLIES_DECLINE_REASON_Site Closed \			
0		-0.15	
1		-0.15	
2		-0.15	
3		-0.15	
4		-0.15	
SUPPLIES_DECLINE_REASON_VJ Operations Issues SALES_CHANNEL_Copy \			
0		-0.03	-0.15
1		-0.03	-0.15

2			-0.03	-0.15	
3			-0.03	-0.15	
4			-0.03	-0.15	
	SALES_CHANNEL_ECOMM	PO IMPORT	SALES_CHANNEL_EDI	SALES_CHANNEL_Esker	\
0		0.00	11.24	-0.64	
1		0.00	11.24	-0.64	
2		0.00	11.24	-0.64	
3		0.00	11.24	-0.64	
4		0.00	-0.09	-0.64	
	SALES_CHANNEL_IStore	Account	SALES_CHANNEL_OCC	SALES_CHANNEL_Online	\
0		-0.08	-0.20	-1.24	
1		-0.08	-0.20	-1.24	
2		-0.08	-0.20	-1.24	
3		-0.08	-0.20	-1.24	
4		-0.08	-0.20	-1.24	
	SALES_CHANNEL_SFDC_CPQ	SALES_CHANNEL_Service	Billing	PRODUCT_FAMILY_CIJ	\
0		-0.08	0.00	0.00	
1		-0.08	0.00	0.00	
2		-0.08	0.00	0.00	
3		-0.08	0.00	0.00	
4		-0.08	0.00	0.00	
	PRODUCT_FAMILY_GRAPHICS	PRODUCT_FAMILY_GRAPHICS	BA	PRODUCT_FAMILY_LASER	\
0		0.00	0.00	0.00	
1		0.00	0.00	0.00	
2		0.00	0.00	0.00	
3		0.00	0.00	0.00	
4		0.00	0.00	0.00	
	PRODUCT_FAMILY_LCM	PRODUCT_FAMILY_LPA	PRODUCT_FAMILY_RAW MATERIAL		\
0		0.00	0.00	0.00	
1		0.00	0.00	0.00	
2		0.00	0.00	0.00	
3		0.00	0.00	0.00	
4		0.00	0.00	0.00	
	PRODUCT_FAMILY_TIJ	PRODUCT_FAMILY_TTO	PRODUCT_MODEL_CLEANING SOLUTION		\
0		0.00	0.00	0.00	
1		0.00	0.00	0.00	
2		0.00	0.00	0.00	
3		0.00	0.00	0.00	
4		0.00	0.00	0.00	
	PRODUCT_MODEL_FUME EXTRACTION	PRODUCT_MODEL_INK	PRODUCT_MODEL_LABELS		\

0	0.00	0.00	0.00
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.00	0.00	0.00
	PRODUCT_MODEL_MAKE-UP	PRODUCT_MODEL_PACKAGING	PRODUCT_MODEL_RIBBONS
0	0.00	0.00	0.00
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.00	0.00	0.00
	PRODUCT_MODEL_SOLVENT	PRODUCT_MODEL_VALUE	PACK
0	0.00	0.00	
1	0.00	0.00	
2	0.00	0.00	
3	0.00	0.00	
4	0.00	0.00	
	Most_Frequent_Interaction_Type_Callback		\
0	0.00		
1	0.00		
2	0.00		
3	0.00		
4	0.00		
	Most_Frequent_Interaction_Type_Contact_Customer		\
0	0.00		
1	0.00		
2	0.00		
3	0.00		
4	0.00		
	Most_Frequent_Interaction_Type_Customer_Meeting		\
0	-0.06		
1	-0.06		
2	-0.06		
3	-0.06		
4	-0.06		
	Most_Frequent_Interaction_Type_Dial	Most_Frequent_Interaction_Type_Email	\
0	0.00	-0.21	
1	0.00	-0.21	
2	0.00	-0.21	
3	0.00	-0.21	
4	0.00	-0.21	

```

    Most_Frequent_Interaction_Type_Make Qualified Sales Call  \
0                           0.00
1                           0.00
2                           0.00
3                           0.00
4                           0.00

    Most_Frequent_Interaction_Type_Meeting  \
0                     -0.10
1                     -0.10
2                     -0.10
3                     -0.10
4                     -0.10

    Most_Frequent_Interaction_Type_Other  \
0                     -0.20
1                     -0.20
2                     -0.20
3                     -0.20
4                     -0.20

    Most_Frequent_Interaction_Type_TS Task  \
0                           0.00
1                           0.00
2                           0.00
3                           0.00
4                           0.00

    Max_Case_Origin_CX Survey Detractor  Max_Case_Origin_Email  \
0                           0.00           -0.05
1                           0.00           -0.05
2                           0.00           -0.05
3                           0.00           -0.05
4                           0.00           -0.05

    Max_Case_Origin_Email - VTI CC Sales Escalations  \
0                           0.00
1                           0.00
2                           0.00
3                           0.00
4                           0.00

    Max_Case_Origin_Email - VTI NACC  Max_Case_Origin_Email/Fax - VTI CS  \
0                     2.43           -0.08
1                     2.43           -0.08
2                     2.43           -0.08

```

3	2.43	-0.08
4	2.43	-0.08
		\
0	-0.06	-0.03
1	-0.06	-0.03
2	-0.06	-0.03
3	-0.06	-0.03
4	-0.06	-0.03
		\
0	-0.35	-0.10
1	-0.35	-0.10
2	-0.35	-0.10
3	-0.35	-0.10
4	-0.35	-0.10
		\
0	-1.46	-0.19
1	-1.46	-0.19
2	-1.46	-0.19
3	-1.46	-0.19
4	-1.46	-0.19
		\
0	0.00	-0.12
1	0.00	-0.12
2	0.00	-0.12
3	0.00	-0.12
4	0.00	-0.12
		\
0	0.00	0.00
1	0.00	0.00
2	0.00	0.00
3	0.00	0.00
4	0.00	0.00
		\
0	-0.12	1.74
1	-0.12	1.74
2	-0.12	1.74
3	-0.12	1.74
4	-0.12	1.74
		\
0	-1.46	-0.18

1	-1.46	-0.18		
2	-1.46	-0.18		
3	-1.46	-0.18		
4	-1.46	-0.18		
	Contract_Category_No	Contract	Contract_Category_Supportive	\
0		-0.88	0.00	
1		-0.88	0.00	
2		-0.88	0.00	
3		-0.88	0.00	
4		-0.88	0.00	
	Contract_Category_WFC	TERRITORY_REGION_MW	TERRITORY_REGION_NE	\
0	0.00	-0.37	-0.50	
1	0.00	-0.37	-0.50	
2	0.00	-0.37	-0.50	
3	0.00	-0.37	-0.50	
4	0.00	-0.37	-0.50	
	TERRITORY_REGION_NW	TERRITORY_REGION_SC	TERRITORY_REGION_SE	\
0	1.82	-0.42	-0.47	
1	1.82	-0.42	-0.47	
2	1.82	-0.42	-0.47	
3	1.82	-0.42	-0.47	
4	1.82	-0.42	-0.47	
	Most_Frequent_Sales_Channel_Copy	Most_Frequent_Sales_Channel_EDI	\	
0		-0.04	9.81	
1		-0.04	9.81	
2		-0.04	9.81	
3		-0.04	9.81	
4		-0.04	9.81	
	Most_Frequent_Sales_Channel_Esker	\		
0		-0.66		
1		-0.66		
2		-0.66		
3		-0.66		
4		-0.66		
	Most_Frequent_Sales_Channel_IStore	Account	\	
0		-0.04		
1		-0.04		
2		-0.04		
3		-0.04		
4		-0.04		

	Most_Frequent_Sales_Channel_OCC	Most_Frequent_Sales_Channel_Online	\
0	-0.14	-1.34	
1	-0.14	-1.34	
2	-0.14	-1.34	
3	-0.14	-1.34	
4	-0.14	-1.34	
	Most_Frequent_Sales_Channel_SFDC_CPQ	\	
0	-0.05		
1	-0.05		
2	-0.05		
3	-0.05		
4	-0.05		
	Most_Frequent_Sales_Channel_Service_Billing	\	
0	0.00		
1	0.00		
2	0.00		
3	0.00		
4	0.00		
	Most_Frequent_Order_Type_BILL_ONLY	\	
0	-0.01		
1	-0.01		
2	-0.01		
3	-0.01		
4	-0.01		
	Most_Frequent_Order_Type_DEMO_EQUIPMENT_ACCEPT	\	
0	-0.01		
1	-0.01		
2	-0.01		
3	-0.01		
4	-0.01		
	Most_Frequent_Order_Type_EDI	Most_Frequent_Order_Type_EQUIPMENT_DOMESTIC	\
0	9.81	-0.11	
1	9.81	-0.11	
2	9.81	-0.11	
3	9.81	-0.11	
4	9.81	-0.11	
	Most_Frequent_Order_Type_SERVICE	\	
0	0.00		
1	0.00		
2	0.00		
3	0.00		

```

4          0.00

  Most_Frequent_Order_Type_STANDARD DOMESTIC \
0           -4.56
1           -4.56
2           -4.56
3           -4.56
4           -4.56

  Most_Frequent_Order_Type_STANDARD INTERNATIONAL \
0            -0.05
1            -0.05
2            -0.05
3            -0.05
4            -0.05

  Most_Frequent_Order_Type_US FULL CARE EQPT DOMESTIC \
0            -0.02
1            -0.02
2            -0.02
3            -0.02
4            -0.02

  Most_Frequent_Order_Type_US FULL CARE INTERNATIONAL \
0            0.00
1            0.00
2            0.00
3            0.00
4            0.00

  Most_Frequent_Order_Type_WEB ORDER
0            -0.13
1            -0.13
2            -0.13
3            -0.13
4            -0.13

```

[40]: tto_final = pd.concat([tto_scaled,ttoChurned],axis=1,sort=False)

[41]:

```
tto_final = tto_final.drop(columns=['SHORT_VERTICAL_POSTAL',  
→'TERRITORY_TYPE_Postal', 'SUPPLIES_DECLINE_REASON_Financial Distress/Credit  
→Hold', 'SALES_CHANNEL_ECOMM PO IMPORT', 'SALES_CHANNEL_Service Billing',  
→'PRODUCT_FAMILY_CIJ', 'PRODUCT_FAMILY_GRAPHICS', 'PRODUCT_FAMILY_GRAPHICS  
→BA', 'PRODUCT_FAMILY_LASER', 'PRODUCT_FAMILY_LCM', 'PRODUCT_FAMILY_LPA',  
→'PRODUCT_FAMILY_RAW MATERIAL', 'PRODUCT_FAMILY_TIJ', 'PRODUCT_FAMILY_TTO',  
→'PRODUCT_MODEL_CLEANING SOLUTION', 'PRODUCT_MODEL_FUME EXTRACTION',  
→'PRODUCT_MODEL_INK', 'PRODUCT_MODEL_LABELS', 'PRODUCT_MODEL_MAKE-UP',  
→'PRODUCT_MODEL_PACKAGING', 'PRODUCT_MODEL_RIBBONS', 'PRODUCT_MODEL_SOLVENT',  
→'PRODUCT_MODEL_VALUE PACK', 'Most_Frequent_Interaction_Type_Callback',  
→'Most_Frequent_Interaction_Type_Contact Customer',  
→'Most_Frequent_Interaction_Type_Dial', 'Most_Frequent_Interaction_Type_Make  
→Qualified Sales Call', 'Most_Frequent_Interaction_Type_TS Task',  
→'Max_Case-Origin_CX Survey Detractor', 'Max_Case-Origin_Email - VTI CC Sales  
→Escalations', 'Max_Case_Reason_CX: Field Sales', 'Max_Case_Reason_CX:  
→Manufacturing', 'Max_Case_Reason_CX: Other Team',  
→'Contract_Category_Supportive', 'Contract_Category_WFC',  
→'Most_Frequent_Sales_Channel_Service Billing',  
→'Most_Frequent_Order_Type_SERVICE', 'Most_Frequent_Order_Type_US FULL CARE  
→INTERNATIONAL'], axis=1)
```

```
[42]: tto_final.shape
```

```
[42]: (12346, 119)
```

```
[43]: cph1 = lifelines.CoxPHFitter(penalizer=0.000001)  
cph1.fit(tto_final, step_size=0.05, duration_col='Tenure',  
→event_col='Churned_365', show_progress=False)  
cph1.print_summary()
```

covariate	coef	exp(coef)	se(coef)	coef lower 95%
Site_Level_Price_Index	-0.07	0.93	0.04	-0.11
TRX_AMT_USD	0.90	2.46	0.13	0.63
Margin	-0.74	0.48	0.12	-0.90
QUANTITY	-0.28	0.75	0.07	-0.42
Total_SVC_Incidents	0.00	1.00	7.06	-13.84
Total_Repeat_Calls	-0.16	0.85	2.19	-4.40
Total_FTF_Calls	0.07	1.07	5.13	-9.99
Total_Visits	-0.08	0.92	0.06	-0.11
Total_Cases	0.23	1.26	0.03	0.17
Num_of_Active_Install_Bases	-0.13	0.88	0.05	-0.23
Total_Contracts	0.07	1.08	0.07	-0.07
Contract_length	-0.19	0.82	0.08	-0.33
Num_of_Inactive_Install_Bases	-0.02	0.98	0.04	-0.10
STRATEGIC_ACCOUNTS	0.44	1.55	0.05	0.38
Frequency	-0.08	0.93	0.03	-0.11
Num_of_Trxns	-0.30	0.74	0.07	-0.44
Avg_Margin	-0.15	0.86	0.07	-0.29
Avg_Quantity	0.28	1.32	0.07	0.18
Types_of_Product_Family	-0.18	0.84	0.07	-0.32
Types_of_Product_Model	-0.31	0.73	0.08	-0.40
Avg_Price_Index	-0.07	0.93	0.05	-0.10
SHORT_VERTICAL_BAKED GOODS & CEREALS	0.14	1.15	0.10	-0.00
SHORT_VERTICAL_BEVERAGE	-0.47	0.62	3.28	-6.85
SHORT_VERTICAL_BUILDING MATERIALS	0.16	1.18	0.08	-0.00
SHORT_VERTICAL_CANDY & CONFECTION	0.30	1.35	0.09	0.13
SHORT_VERTICAL_CHEMICALS	0.03	1.03	0.06	-0.08
SHORT_VERTICAL_COSMETICS / PERSONAL CARE	0.04	1.04	0.07	-0.09
SHORT_VERTICAL_DAIRY & EGGS	0.05	1.05	0.08	-0.10
SHORT_VERTICAL_DISTRIBUTOR	-0.00	1.00	6.93	-13.58
SHORT_VERTICAL_ELECTRICAL / ELECTRONICS	-0.15	0.86	2.79	-5.63
SHORT_VERTICAL_EXTRUSION / WIRE & CABLE	0.16	1.18	0.05	0.07
SHORT_VERTICAL_FISH & SEAFOOD	-0.26	0.77	0.12	-0.48
SHORT_VERTICAL_FROZEN PREPARED MEALS	0.41	1.51	0.06	0.30
SHORT_VERTICAL_FRUIT & VEGETABLE	0.29	1.33	0.09	0.10
SHORT_VERTICAL_GRAPHICS	0.14	1.15	0.07	-0.01
SHORT_VERTICAL_INDUSTRIAL EQUIPMENT	0.18	1.20	0.04	0.10
SHORT_VERTICAL_MEAT & POULTRY	0.03	1.03	0.10	-0.11
SHORT_VERTICAL_OEM-INDUSTRIAL EQUIPMENT	-0.20	0.81	0.06	-0.33
SHORT_VERTICAL_OTHER	0.09	1.10	0.09	-0.08
SHORT_VERTICAL_OTHER FOOD	0.35	1.42	0.14	0.07
SHORT_VERTICAL_PACKAGING MATERIALS	0.10	1.11	0.05	0.00
SHORT_VERTICAL_PET FOOD & ANIMAL FEED	0.18	1.20	0.03	0.11
SHORT_VERTICAL_PHARMA & MEDICAL	0.27	1.32	0.11	0.07
SHORT_VERTICAL_SALTY SNACKS	0.32	1.38	0.12	0.09
SHORT_VERTICAL_TEXTILE	0.07	1.08	0.02	0.03
SHORT_VERTICAL_TOBACCO	-0.15	0.86	3.19	-6.40
SHORT_VERTICAL_UNKNOWN	0.49	1.63	0.07	0.30
CUSTOMER_CLASS_END USER	0.18	1.20	0.09	0.00
CUSTOMER_CLASS_OEM	0.43	1.53	0.09	0.24
TERRITORY_TYPE_Industrial	-0.06	0.95	0.03	-0.11
SUPPLIES_SEGMENTATION_M	0.12	1.14	0.05	0.00

```
[44]: tto_final_table = cph1.summary[cph1.summary['p']<0.005].
      ↪sort_values(by='exp(coef)', ascending=False).head(10)
```

3 Model building for LCM product family

```
[46]: lcm.reset_index(inplace=True,drop=True)
```

```
[47]: lcm.head()
```

	Site_Level_Price_Index	TRX_AMT_USD	Margin	QUANTITY	\
0	0.79	552.68	537.02	2	
1	0.79	5,000.00	4,843.37	20	
2	0.79	5,000.00	4,843.37	20	
3	0.86	782.12	754.50	4	
4	1.06	880.00	232.91	4	

	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	Total_Visits	\
0	13.00	7.00	6.00	81.00	
1	13.00	7.00	6.00	81.00	
2	13.00	7.00	6.00	81.00	
3	154.00	26.00	128.00	123.00	
4	50.00	7.00	43.00	99.00	

	Total_Cases	Num_of_Active_Install_Bases	Total_Contracts	Contract_length	\
0	1.97	5.00	0.00	0.00	
1	1.97	5.00	0.00	0.00	
2	1.97	5.00	0.00	0.00	
3	1.00	34.00	42.00	1,266.79	
4	1.39	15.00	12.00	1,095.00	

	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	Frequency	Num_of_Trxns	\
0	0.00	0	17.67	4	
1	0.00	0	17.67	4	
2	0.00	0	17.67	4	
3	10.00	0	8.97	218	
4	0.00	0	10.28	190	

	Avg_Margin	Avg_Quantity	Types_of_Product_Family	Types_of_Product_Model	\
0	2,901.24	12.00	2	2	
1	2,901.24	12.00	2	2	
2	2,901.24	12.00	2	2	
3	673.77	37.66	3	5	
4	972.92	10.89	3	3	

	Avg_Price_Index	Tenure	Churned_365	\
0	0.79	77.00	0	

1	0.79	77.00	0
2	0.79	77.00	0
3	1.00	1,961.00	0
4	1.14	1,957.00	0
0	SHORT_VERTICAL_BAKED GOODS & CEREALS	SHORT_VERTICAL_BEVERAGE	\
1	0	0	0
2	0	0	0
3	0	1	
4	0	0	
0	SHORT_VERTICAL_BUILDING MATERIALS	SHORT_VERTICAL_CANDY & CONFECTION	\
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	
0	SHORT_VERTICAL_CHEMICALS	SHORT_VERTICAL_COSMETICS / PERSONAL CARE	\
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	
0	SHORT_VERTICAL_DAIRY & EGGS	SHORT_VERTICAL_DISTRIBUTOR	\
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	
0	SHORT_VERTICAL_ELECTRICAL / ELECTRONICS	\	
1	0		
2	0		
3	0		
4	0		
0	SHORT_VERTICAL_EXTRUSION / WIRE & CABLE	SHORT_VERTICAL_FISH & SEAFOOD	\
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	

	SHORT_VERTICAL_FROZEN PREPARED MEALS	SHORT_VERTICAL_FRUIT & VEGETABLE	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_GRAPHICS	SHORT_VERTICAL_INDUSTRIAL EQUIPMENT	\
0	1	0	
1	1	0	
2	1	0	
3	0	0	
4	1	0	
	SHORT_VERTICAL_MEAT & POULTRY	SHORT_VERTICAL_OEM-INDUSTRIAL EQUIPMENT	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_OTHER	SHORT_VERTICAL_OTHER FOOD	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_PACKAGING MATERIALS	SHORT_VERTICAL_PET FOOD & ANIMAL FEED	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_PHARMA & MEDICAL	SHORT_VERTICAL_POSTAL	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_SALTY SNACKS	SHORT_VERTICAL_TEXTILE	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	

4	0	0	
0	0	0	1
1	0	0	1
2	0	0	1
3	0	0	1
4	0	0	1
CUSTOMER_CLASS_OEM	TERRITORY_TYPE_Industrial	TERRITORY_TYPE_Postal	\
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
SUPPLIES_SEGMENTATION_M	SUPPLIES_SEGMENTATION_S	\	
0	0	1	
1	0	1	
2	0	1	
3	0	0	
4	0	0	
SUPPLIES_SEGMENTATION_Unclass	SUPPLIES_SEGMENTATION_XL	\	
0	0	0	
1	0	0	
2	0	0	
3	0	1	
4	0	1	
SUPPLIES_DECLINE_REASON_Financial Distress/Credit Hold	\		
0	0		
1	0		
2	0		
3	0		
4	0		
SUPPLIES_DECLINE_REASON_Migration to 1000 Line/TIJ/TTO/LCM/LPA	\		
0	0		
1	0		
2	0		
3	0		
4	0		
SUPPLIES_DECLINE_REASON_Migration to Lasers	\		
0	0		
1	0		

2		0
3		0
4		0
SUPPLIES_DECLINE_REASONs_Moved Equipment \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_No More Coding Requirement \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_None SUPPLIES_DECLINE_REASONs_Off Brand \		
0	1	0
1	1	0
2	1	0
3	0	0
4	0	0
SUPPLIES_DECLINE_REASONs_Over Stocked / Timing \		
0	0	
1	0	
2	0	
3	1	
4	1	
SUPPLIES_DECLINE_REASONs_Pricing / Discounting \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_Printing/EQ downtime Issues \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_Production / Code Reduction \		

0	0
1	0
2	0
3	0
4	0

SUPPLIES_DECLINE_REASON_Production Down (timing) \	
0	0
1	0
2	0
3	0
4	0

SUPPLIES_DECLINE_REASON_Project Based \	
0	0
1	0
2	0
3	0
4	0

SUPPLIES_DECLINE_REASON_Recent Regain/Win-back \	
0	0
1	0
2	0
3	0
4	0

SUPPLIES_DECLINE_REASON_Seasonal Producer \	
0	0
1	0
2	0
3	0
4	0

SUPPLIES_DECLINE_REASON_Served by Authorized Distributor \	
0	0
1	0
2	0
3	0
4	0

SUPPLIES_DECLINE_REASON_Site Closed \	
0	0
1	0
2	0
3	0
4	0

	SUPPLIES_DECLINE_REASON_VJ	Operations	Issues	SALES_CHANNEL_Copy	\
0			0	0	
1			0	0	
2			0	1	
3			0	0	
4			0	0	
	SALES_CHANNEL_ECOMM	PO IMPORT	SALES_CHANNEL_EDI	SALES_CHANNEL_Esker	\
0		0	0	0	
1		0	0	0	
2		0	0	0	
3		0	0	0	
4		0	0	0	
	SALES_CHANNEL_IStore	Account	SALES_CHANNEL_OCC	SALES_CHANNEL_Online	\
0		0	0	0	
1		0	0	1	
2		0	0	0	
3		0	0	1	
4		0	0	1	
	SALES_CHANNEL_SFDC_CPQ	SALES_CHANNEL_Service	Billing	PRODUCT_FAMILY_CIJ	\
0		1	0	0	
1		0	0	0	
2		0	0	0	
3		0	0	0	
4		0	0	0	
	PRODUCT_FAMILY_GRAPHICS	PRODUCT_FAMILY_GRAPHICS	BA	PRODUCT_FAMILY_LASER	\
0		0	0	0	
1		0	0	0	
2		0	0	0	
3		0	0	0	
4		0	0	0	
	PRODUCT_FAMILY_LCM	PRODUCT_FAMILY_LPA	PRODUCT_FAMILY_RAW MATERIAL		\
0		1	0	0	
1		1	0	0	
2		1	0	0	
3		1	0	0	
4		1	0	0	
	PRODUCT_FAMILY_TIJ	PRODUCT_FAMILY_TTO	PRODUCT_MODEL_CLEANING SOLUTION		\
0		0	0	0	
1		0	0	0	
2		0	0	0	

3	0	0	0
4	0	0	0
	PRODUCT_MODEL_FUME_EXTRACTION	PRODUCT_MODEL_INK	PRODUCT_MODEL_LABELS
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
	PRODUCT_MODEL_MAKE-UP	PRODUCT_MODEL_PACKAGING	PRODUCT_MODEL_RIBBONS
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
	PRODUCT_MODEL_SOLVENT	PRODUCT_MODEL_VALUE_PACK	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	Most_Frequent_Interaction_Type_Callback	\	
0	0		
1	0		
2	0		
3	0		
4	0		
	Most_Frequent_Interaction_Type_Contact_Customer	\	
0	0		
1	0		
2	0		
3	0		
4	0		
	Most_Frequent_Interaction_Type_Customer_Meeting	\	
0	0		
1	0		
2	0		
3	0		
4	0		
	Most_Frequent_Interaction_Type_Dial	Most_Frequent_Interaction_Type_Email	\
0	0	0	

1	0	0
2	0	0
3	0	0
4	0	0
0	Most_Frequent_Interaction_Type_Make Qualified Sales Call \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Most_Frequent_Interaction_Type_Meeting \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Most_Frequent_Interaction_Type_Other \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Most_Frequent_Interaction_Type_TS Task \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Max_Case_Origin_CX Survey Detractor \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Max_Case_Origin_Email - VTI CC Sales Escalations \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0

	Max_Case_Origin_Email - VTI NACC	Max_Case_Origin_Email/Fax - VTI CS \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Origin_FS Survey Followup	Max_Case_Origin_Install Complete \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Origin_Phone	Max_Case_Origin_TS Survey Followup \
0	0	0
1	0	0
2	0	0
3	0	1
4	0	0
	Max_Case_Origin_unknown	Max_Case_Reason_CX: Customer Care \
0	1	0
1	1	0
2	1	0
3	0	0
4	1	0
	Max_Case_Reason_CX: Field Sales	Max_Case_Reason_CX: Field Service \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Reason_CX: Manufacturing	Max_Case_Reason_CX: Other Team \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Max_Case_Reason_CX: Tech Support	Max_Case_Reason_Customer Experience \
0	0	0
1	0	0
2	0	0
3	1	0

4	0	0	
0	Max_Case_Reason_unknown	Contract_Category_Full	Care \
1	1		0
2	1		0
3	0		0
4	1		0
0	Contract_Category_No	Contract	Contract_Category_Supportive \
1		1	0
2		1	0
3		1	0
4		0	0
0	Contract_Category_WFC	TERRITORY_REGION_MW	TERRITORY_REGION_NE \
1	0	1	0
2	0	1	0
3	0	0	1
4	0	0	0
0	TERRITORY_REGION_NW	TERRITORY_REGION_SC	TERRITORY_REGION_SE \
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	1
0	Most_Frequent_Sales_Channel_Copy	Most_Frequent_Sales_Channel_EDI	\
1	1		0
2	1		0
3	0		0
4	0		0
0	Most_Frequent_Sales_Channel_Esker	\	
1	0		
2	0		
3	0		
4	0		
0	Most_Frequent_Sales_Channel_IStore	Account	\
1		0	
		0	

2		0
3		0
4		0
	Most_Frequent_Sales_Channel_OCC	Most_Frequent_Sales_Channel_Online \
0	0	0
1	0	0
2	0	0
3	0	1
4	0	1
	Most_Frequent_Sales_Channel_SFDC_CPQ \	
0	0	
1	0	
2	0	
3	0	
4	0	
	Most_Frequent_Sales_Channel_Service Billing \	
0		0
1		0
2		0
3		0
4		0
	Most_Frequent_Order_Type_BILL ONLY \	
0	0	
1	0	
2	0	
3	0	
4	0	
	Most_Frequent_Order_Type_DEMO EQUIPMENT ACCEPT \	
0		0
1		0
2		0
3		0
4		0
	Most_Frequent_Order_Type_EDI	Most_Frequent_Order_Type_EQUIPMENT DOMESTIC \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Most_Frequent_Order_Type_SERVICE \	

```

0          0
1          0
2          0
3          0
4          0

    Most_Frequent_Order_Type_STANDARD DOMESTIC \
0          1
1          1
2          1
3          1
4          1

    Most_Frequent_Order_Type_STANDARD INTERNATIONAL \
0          0
1          0
2          0
3          0
4          0

    Most_Frequent_Order_Type_US FULL CARE EQPT DOMESTIC \
0          0
1          0
2          0
3          0
4          0

    Most_Frequent_Order_Type_US FULL CARE INTERNATIONAL \
0          0
1          0
2          0
3          0
4          0

    Most_Frequent_Order_Type_WEB ORDER
0          0
1          0
2          0
3          0
4          0

```

[48]: `lcmChurned = lcm[['Churned_365']]`

[49]: `lcm.drop(columns=['Churned_365'],axis=1,inplace=True)`

[50]: `#Scaling
Instantiate`

```

scaler = StandardScaler()

# fit_transform
lcm_scaled = scaler.fit_transform(lcm)

[51]: lcm_scaled = pd.DataFrame(lcm_scaled)
lcm_scaled.columns = lcm.columns
lcm_scaled.head()

```

	Site_Level_Price_Index	TRX_AMT_USD	Margin	QUANTITY	Total_SVC_Incidents	\
0	-0.49	-0.26	-0.25	-0.33		-0.58
1	-0.49	0.58	0.63	0.31		-0.58
2	-0.49	0.58	0.63	0.31		-0.58
3	-0.33	-0.21	-0.20	-0.26		0.96
4	0.15	-0.19	-0.31	-0.26		-0.18

	Total_Repeat_Calls	Total_FTF_Calls	Total_Visits	Total_Cases	\
0	-0.32	-0.67	0.19	0.09	
1	-0.32	-0.67	0.19	0.09	
2	-0.32	-0.67	0.19	0.09	
3	0.29	1.22	0.80	-0.97	
4	-0.32	-0.10	0.45	-0.55	

	Num_of_Active_Install_Bases	Total_Contracts	Contract_length	\
0	-0.36	-0.60	-0.91	
1	-0.36	-0.60	-0.91	
2	-0.36	-0.60	-0.91	
3	0.02	1.51	1.17	
4	-0.23	0.00	0.89	

	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	Frequency	Num_of_Trxns	\
0	-0.26	-0.51	-0.32	-0.84	
1	-0.26	-0.51	-0.32	-0.84	
2	-0.26	-0.51	-0.32	-0.84	
3	-0.10	-0.51	-0.43	0.39	
4	-0.26	-0.51	-0.41	0.23	

	Avg_Margin	Avg_Quantity	Types_of_Product_Family	Types_of_Product_Model	\
0	0.91	-0.09	-0.11		-0.94
1	0.91	-0.09	-0.11		-0.94
2	0.91	-0.09	-0.11		-0.94
3	-0.25	1.08	1.05		1.36
4	-0.09	-0.14	1.05		-0.18

	Avg_Price_Index	Tenure	SHORT_VERTICAL_BAKED_GOODS & CEREALS	\
0	-0.71	-3.54		-0.33
1	-0.71	-3.54		-0.33

2	-0.71	-3.54	-0.33
3	-0.06	0.53	-0.33
4	0.40	0.52	-0.33
0	SHORT_VERTICAL_BEVERAGE	SHORT_VERTICAL_BUILDING MATERIALS	\
1	-0.24	-0.14	
2	-0.24	-0.14	
3	-0.24	-0.14	
4	4.19	-0.14	
0	-0.24	-0.14	
0	SHORT_VERTICAL_CANDY & CONFECTION	SHORT_VERTICAL_CHEMICALS	\
1	-0.15	-0.18	
2	-0.15	-0.18	
3	-0.15	-0.18	
4	-0.15	-0.18	
0	SHORT_VERTICAL_COSMETICS / PERSONAL CARE	SHORT_VERTICAL_DAIRY & EGGS	\
1	-0.22	-0.32	
2	-0.22	-0.32	
3	-0.22	-0.32	
4	-0.22	-0.32	
0	SHORT_VERTICAL_DISTRIBUTOR	SHORT_VERTICAL_ELECTRICAL / ELECTRONICS	\
1	-0.06	-0.04	
2	-0.06	-0.04	
3	-0.06	-0.04	
4	-0.06	-0.04	
0	SHORT_VERTICAL_EXTRUSION / WIRE & CABLE	SHORT_VERTICAL_FISH & SEAFOOD	\
1	-0.03	-0.08	
2	-0.03	-0.08	
3	-0.03	-0.08	
4	-0.03	-0.08	
0	SHORT_VERTICAL_FROZEN PREPARED MEALS	SHORT_VERTICAL_FRUIT & VEGETABLE	\
1	-0.15	-0.29	
2	-0.15	-0.29	
3	-0.15	-0.29	
4	-0.15	-0.29	
0	SHORT_VERTICAL_GRAPHICS	SHORT_VERTICAL_INDUSTRIAL EQUIPMENT	\

0	3.47	-0.31
1	3.47	-0.31
2	3.47	-0.31
3	-0.29	-0.31
4	3.47	-0.31
0	SHORT_VERTICAL_MEAT & POULTRY	SHORT_VERTICAL_OEM-INDUSTRIAL EQUIPMENT \
1	-0.21	-0.08
2	-0.21	-0.08
3	-0.21	-0.08
4	-0.21	-0.08
0	SHORT_VERTICAL_OTHER	SHORT_VERTICAL_OTHER FOOD \
1	-0.24	-0.43
2	-0.24	-0.43
3	-0.24	-0.43
4	-0.24	-0.43
0	SHORT_VERTICAL_PACKAGING MATERIALS	SHORT_VERTICAL_PET FOOD & ANIMAL FEED \
1	-0.07	-0.09
2	-0.07	-0.09
3	-0.07	-0.09
4	-0.07	-0.09
0	SHORT_VERTICAL_PHARMA & MEDICAL	SHORT_VERTICAL_POSTAL \
1	-0.18	-0.06
2	-0.18	-0.06
3	-0.18	-0.06
4	-0.18	-0.06
0	SHORT_VERTICAL_SALTY SNACKS	SHORT_VERTICAL_TEXTILE \
1	-0.13	-0.07
2	-0.13	-0.07
3	-0.13	-0.07
4	-0.13	-0.07
0	SHORT_VERTICAL_TOBACCO	SHORT_VERTICAL_UNKNOWN CUSTOMER_CLASS_END USER \
1	-0.04	-0.14 0.33
2	-0.04	-0.14 0.33
3	-0.04	-0.14 0.33
4	-0.04	-0.14 0.33

	CUSTOMER_CLASS_OEM	TERRITORY_TYPE_Industrial	TERRITORY_TYPE_Postal	\
0	-0.09	0.17	-0.06	
1	-0.09	0.17	-0.06	
2	-0.09	0.17	-0.06	
3	-0.09	0.17	-0.06	
4	-0.09	0.17	-0.06	
	SUPPLIES_SEGMENTATION_M	SUPPLIES_SEGMENTATION_S	\	
0	-0.36	1.32		
1	-0.36	1.32		
2	-0.36	1.32		
3	-0.36	-0.76		
4	-0.36	-0.76		
	SUPPLIES_SEGMENTATION_Unclass	SUPPLIES_SEGMENTATION_XL	\	
0	0.00	-0.57		
1	0.00	-0.57		
2	0.00	-0.57		
3	0.00	1.75		
4	0.00	1.75		
	SUPPLIES_DECLINE_REASON_Financial	Distress/Credit Hold	\	
0	0.00			
1	0.00			
2	0.00			
3	0.00			
4	0.00			
	SUPPLIES_DECLINE_REASON_Migration_to_1000	Line/TIJ/TTO/LCM/LPA	\	
0	-0.24			
1	-0.24			
2	-0.24			
3	-0.24			
4	-0.24			
	SUPPLIES_DECLINE_REASON_Migration_to_Lasers	\		
0	-0.10			
1	-0.10			
2	-0.10			
3	-0.10			
4	-0.10			
	SUPPLIES_DECLINE_REASON_Moved_Equipment	\		
0	-0.08			
1	-0.08			
2	-0.08			

3		-0.08
4		-0.08
	SUPPLIES_DECLINE_REASONs_No More Coding Requirement \	
0		-0.06
1		-0.06
2		-0.06
3		-0.06
4		-0.06
	SUPPLIES_DECLINE_REASONs_None SUPPLIES_DECLINE_REASONs_Off Brand \	
0	0.87	-0.17
1	0.87	-0.17
2	0.87	-0.17
3	-1.15	-0.17
4	-1.15	-0.17
	SUPPLIES_DECLINE_REASONs_Over Stocked / Timing \	
0		-0.33
1		-0.33
2		-0.33
3		3.02
4		3.02
	SUPPLIES_DECLINE_REASONs_Pricing / Discounting \	
0	0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
	SUPPLIES_DECLINE_REASONs_Printing/EQ downtime Issues \	
0		-0.03
1		-0.03
2		-0.03
3		-0.03
4		-0.03
	SUPPLIES_DECLINE_REASONs_Production / Code Reduction \	
0		-0.26
1		-0.26
2		-0.26
3		-0.26
4		-0.26
	SUPPLIES_DECLINE_REASONs_Production Down (timing) \	
0		-0.18

1		-0.18	
2		-0.18	
3		-0.18	
4		-0.18	
	SUPPLIES_DECLINE_REASONs_Project Based \		
0		-0.03	
1		-0.03	
2		-0.03	
3		-0.03	
4		-0.03	
	SUPPLIES_DECLINE_REASONs_Recent Regain/Win-back \		
0		-0.16	
1		-0.16	
2		-0.16	
3		-0.16	
4		-0.16	
	SUPPLIES_DECLINE_REASONs_Seasonal Producer \		
0		-0.13	
1		-0.13	
2		-0.13	
3		-0.13	
4		-0.13	
	SUPPLIES_DECLINE_REASONs_Served by Authorized Distributor \		
0		-0.13	
1		-0.13	
2		-0.13	
3		-0.13	
4		-0.13	
	SUPPLIES_DECLINE_REASONs_Site Closed \		
0		-0.12	
1		-0.12	
2		-0.12	
3		-0.12	
4		-0.12	
	SUPPLIES_DECLINE_REASONs_VJ Operations Issues SALES_CHANNEL_Copy \		
0		-0.02	-0.16
1		-0.02	-0.16
2		-0.02	6.44
3		-0.02	-0.16
4		-0.02	-0.16

	SALES_CHANNEL_ECOMM	PO_IMPORT	SALES_CHANNEL_EDI	SALES_CHANNEL_Esker	\
0		-0.01	-0.09	-0.57	
1		-0.01	-0.09	-0.57	
2		-0.01	-0.09	-0.57	
3		-0.01	-0.09	-0.57	
4		-0.01	-0.09	-0.57	
	SALES_CHANNEL_IStore	Account	SALES_CHANNEL_OCC	SALES_CHANNEL_Online	\
0		-0.11	-0.22	-1.26	
1		-0.11	-0.22	0.80	
2		-0.11	-0.22	-1.26	
3		-0.11	-0.22	0.80	
4		-0.11	-0.22	0.80	
	SALES_CHANNEL_SFDC_CPQ	SALES_CHANNEL_Service	Billing	PRODUCT_FAMILY_CIJ	\
0		10.13	-0.01	0.00	
1		-0.10	-0.01	0.00	
2		-0.10	-0.01	0.00	
3		-0.10	-0.01	0.00	
4		-0.10	-0.01	0.00	
	PRODUCT_FAMILY_GRAPHICS	PRODUCT_FAMILY_GRAPHICS	BA	PRODUCT_FAMILY_LASER	\
0		0.00	0.00	0.00	
1		0.00	0.00	0.00	
2		0.00	0.00	0.00	
3		0.00	0.00	0.00	
4		0.00	0.00	0.00	
	PRODUCT_FAMILY_LCM	PRODUCT_FAMILY_LPA	PRODUCT_FAMILY_RAW_MATERIAL		\
0		0.00	0.00	0.00	
1		0.00	0.00	0.00	
2		0.00	0.00	0.00	
3		0.00	0.00	0.00	
4		0.00	0.00	0.00	
	PRODUCT_FAMILY_TIJ	PRODUCT_FAMILY_TTO	PRODUCT_MODEL_CLEANING_SOLUTION		\
0		0.00	0.00	0.00	
1		0.00	0.00	0.00	
2		0.00	0.00	0.00	
3		0.00	0.00	0.00	
4		0.00	0.00	0.00	
	PRODUCT_MODEL_FUME_EXTRACTION	PRODUCT_MODEL_INK	PRODUCT_MODEL_LABELS		\
0		0.00	0.39	0.00	
1		0.00	0.39	0.00	
2		0.00	0.39	0.00	
3		0.00	0.39	0.00	

4	0.00	0.39	0.00
	PRODUCT_MODEL_MAKE-UP	PRODUCT_MODEL_PACKAGING	PRODUCT_MODEL_RIBBONS \
0	0.00	0.00	0.00
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.00	0.00	0.00
	PRODUCT_MODEL_SOLVENT	PRODUCT_MODEL_VALUE_PACK \	
0	-0.39	0.00	
1	-0.39	0.00	
2	-0.39	0.00	
3	-0.39	0.00	
4	-0.39	0.00	
	Most_Frequent_Interaction_Type_Callback \		
0		0.00	
1		0.00	
2		0.00	
3		0.00	
4		0.00	
	Most_Frequent_Interaction_Type_Contact Customer \		
0		-0.01	
1		-0.01	
2		-0.01	
3		-0.01	
4		-0.01	
	Most_Frequent_Interaction_Type_Customer Meeting \		
0		-0.04	
1		-0.04	
2		-0.04	
3		-0.04	
4		-0.04	
	Most_Frequent_Interaction_Type_Dial Most_Frequent_Interaction_Type_Email \		
0	0.00		-0.28
1	0.00		-0.28
2	0.00		-0.28
3	0.00		-0.28
4	0.00		-0.28
	Most_Frequent_Interaction_Type_Make Qualified Sales Call \		
0		-0.01	
1		-0.01	

2		-0.01
3		-0.01
4		-0.01
	Most_Frequent_Interaction_Type_Meeting \	
0		-0.07
1		-0.07
2		-0.07
3		-0.07
4		-0.07
	Most_Frequent_Interaction_Type_Other \	
0		-0.26
1		-0.26
2		-0.26
3		-0.26
4		-0.26
	Most_Frequent_Interaction_Type_TS Task \	
0		0.00
1		0.00
2		0.00
3		0.00
4		0.00
	Max_Case-Origin_CX Survey Detractor Max_Case-Origin_Email \	
0		-0.03
1		-0.03
2		-0.03
3		-0.03
4		-0.03
	Max_Case-Origin_Email - VTI CC Sales Escalations \	
0		0.00
1		0.00
2		0.00
3		0.00
4		0.00
	Max_Case-Origin_Email - VTI NACC Max_Case-Origin_Email/Fax - VTI CS \	
0		-0.44
1		-0.44
2		-0.44
3		-0.44
4		-0.44
	Max_Case-Origin_FS Survey Followup Max_Case-Origin_Install Complete \	

0		-0.07	-0.03
1		-0.07	-0.03
2		-0.07	-0.03
3		-0.07	-0.03
4		-0.07	-0.03
	Max_Case_Origin_Phone	Max_Case_Origin_TS	Survey Followup \
0		-0.39	-0.15
1		-0.39	-0.15
2		-0.39	-0.15
3		-0.39	6.63
4		-0.39	-0.15
	Max_Case_Origin_unknown	Max_Case_Reason_CX: Customer Care \	
0		0.73	-0.14
1		0.73	-0.14
2		0.73	-0.14
3		-1.37	-0.14
4		0.73	-0.14
	Max_Case_Reason_CX: Field Sales	Max_Case_Reason_CX: Field Service \	
0		-0.02	-0.09
1		-0.02	-0.09
2		-0.02	-0.09
3		-0.02	-0.09
4		-0.02	-0.09
	Max_Case_Reason_CX: Manufacturing	Max_Case_Reason_CX: Other Team \	
0		0.00	-0.02
1		0.00	-0.02
2		0.00	-0.02
3		0.00	-0.02
4		0.00	-0.02
	Max_Case_Reason_CX: Tech Support	Max_Case_Reason_Customer Experience \	
0		-0.17	-0.64
1		-0.17	-0.64
2		-0.17	-0.64
3		5.97	-0.64
4		-0.17	-0.64
	Max_Case_Reason_unknown	Contract_Category_Full Care \	
0		0.73	-0.27
1		0.73	-0.27
2		0.73	-0.27
3		-1.37	-0.27
4		0.73	-0.27

	Contract_Category_No	Contract	Contract_Category_Supportive	\
0		1.07	-0.04	
1		1.07	-0.04	
2		1.07	-0.04	
3		-0.93	-0.04	
4		-0.93	-0.04	
	Contract_Category_WFC	TERRITORY_REGION_MW	TERRITORY_REGION_NE	\
0		0.00	2.06	-0.53
1		0.00	2.06	-0.53
2		0.00	2.06	-0.53
3		0.00	-0.49	1.90
4		0.00	-0.49	-0.53
	TERRITORY_REGION_NW	TERRITORY_REGION_SC	TERRITORY_REGION_SE	\
0		-0.46	-0.33	-0.45
1		-0.46	-0.33	-0.45
2		-0.46	-0.33	-0.45
3		-0.46	-0.33	-0.45
4		-0.46	-0.33	2.23
	Most_Frequent_Sales_Channel_Copy	Most_Frequent_Sales_Channel_EDI	\	
0		25.52	-0.10	
1		25.52	-0.10	
2		25.52	-0.10	
3		-0.04	-0.10	
4		-0.04	-0.10	
	Most_Frequent_Sales_Channel_Esker	\		
0		-0.65		
1		-0.65		
2		-0.65		
3		-0.65		
4		-0.65		
	Most_Frequent_Sales_Channel_IStore	Account	\	
0		-0.12		
1		-0.12		
2		-0.12		
3		-0.12		
4		-0.12		
	Most_Frequent_Sales_Channel_OCC	Most_Frequent_Sales_Channel_Online	\	
0		-0.11	-1.36	
1		-0.11	-1.36	
2		-0.11	-1.36	

3	-0.11	0.74
4	-0.11	0.74
0	Most_Frequent_Sales_Channel_SFDC_CPQ \ -0.04	
1	-0.04	
2	-0.04	
3	-0.04	
4	-0.04	
0	Most_Frequent_Sales_Channel_Service Billing \ 0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
0	Most_Frequent_Order_Type_BILL ONLY \ -0.02	
1	-0.02	
2	-0.02	
3	-0.02	
4	-0.02	
0	Most_Frequent_Order_Type_DEMO EQUIPMENT ACCEPT \ 0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
0	Most_Frequent_Order_Type_EDI Most_Frequent_Order_Type_EQUIPMENT DOMESTIC \ -0.10 -0.13	
1	-0.10	-0.13
2	-0.10	-0.13
3	-0.10	-0.13
4	-0.10	-0.13
0	Most_Frequent_Order_Type_SERVICE \ 0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
0	Most_Frequent_Order_Type_STANDARD DOMESTIC \ 0.40	

```

1                      0.40
2                      0.40
3                      0.40
4                      0.40

    Most_Frequent_Order_Type_STANDARD INTERNATIONAL \
0                         -0.07
1                         -0.07
2                         -0.07
3                         -0.07
4                         -0.07

    Most_Frequent_Order_Type_US FULL CARE EQPT DOMESTIC \
0                         0.00
1                         0.00
2                         0.00
3                         0.00
4                         0.00

    Most_Frequent_Order_Type_US FULL CARE INTERNATIONAL \
0                         0.00
1                         0.00
2                         0.00
3                         0.00
4                         0.00

    Most_Frequent_Order_Type_WEB ORDER
0                     -0.16
1                     -0.16
2                     -0.16
3                     -0.16
4                     -0.16

```

[52]: `lcm_final = pd.concat([lcm_scaled,lcmChurned],axis=1,sort=False)`

[53]:

```
lcm_final = lcm_final.drop(['SUPPLIES_SEGMENTATION_Unclass',  
    ↳ 'SUPPLIES_DECLINE_REASON_Financial Distress/Credit Hold',  
    ↳ 'SUPPLIES_DECLINE_REASON_Pricing / Discounting', 'PRODUCT_FAMILY_CIJ',  
    ↳ 'PRODUCT_FAMILY_GRAPHICS', 'PRODUCT_FAMILY_GRAPHICS_BA',  
    ↳ 'PRODUCT_FAMILY_LASER', 'PRODUCT_FAMILY_LCM', 'PRODUCT_FAMILY_LPA',  
    ↳ 'PRODUCT_FAMILY_RAW_MATERIAL', 'PRODUCT_FAMILY_TIJ', 'PRODUCT_FAMILY_TTO',  
    ↳ 'PRODUCT_MODEL_CLEANING_SOLUTION', 'PRODUCT_MODEL_FUME_EXTRACTION',  
    ↳ 'PRODUCT_MODEL_LABELS', 'PRODUCT_MODEL_MAKE_UP', 'PRODUCT_MODEL_PACKAGING',  
    ↳ 'PRODUCT_MODEL_RIBBONS', 'PRODUCT_MODEL_VALUE_PACK',  
    ↳ 'Most_Frequent_Interaction_Type_Callback',  
    ↳ 'Most_Frequent_Interaction_Type_Dial', 'Most_Frequent_Interaction_Type_TS  
    ↳ Task', 'Max_Case_Origin_Email - VTI CC Sales Escalations',  
    ↳ 'Max_Case_Reason_CX: Manufacturing', 'Contract_Category_WFC',  
    ↳ 'Most_Frequent_Sales_Channel_Service_Billing',  
    ↳ 'Most_Frequent_Order_Type_DEMO_EQUIPMENT_ACCEPT',  
    ↳ 'Most_Frequent_Order_Type_SERVICE', 'Most_Frequent_Order_Type_US_FULL_CARE  
    ↳ EQPT_DOMESTIC', 'Most_Frequent_Order_Type_US_FULL_CARE  
    ↳ INTERNATIONAL'], axis=1)
```

```
[54]: lcm_final.shape
```

```
[54]: (15658, 127)
```

```
[55]: cph2 = lifelines.CoxPHFitter(penalizer=0.00001)  
cph2.fit(lcm_final, step_size=0.05, duration_col='Tenure',  
    ↳ event_col='Churned_365', show_progress=False)  
cph2.print_summary()
```

covariate		coef	exp(coef)	se(coef)	coef lower	95%
Site_Level_Price_Index		-0.05	0.95	0.05	-0.14	
TRX_AMT_USD		1.16	3.19	0.89	-0.59	
Margin		-0.85	0.43	0.81	-2.43	
QUANTITY		-0.10	0.90	0.10	-0.31	
Total_SVC_Incidents		0.22	1.24	1.99	-3.68	
Total_Repeat_Calls		-0.03	0.98	0.72	-1.43	
Total_FTF_Calls		0.32	1.38	1.41	-2.44	
Total_Visits		-0.77	0.46	0.11	-0.99	
Total_Cases		-0.29	0.75	0.04	-0.36	
Num_of_Active_Install_Bases		-1.22	0.29	0.21	-1.64	
Total_Contracts		0.37	1.45	0.15	0.07	
Contract_length		-0.67	0.51	0.09	-0.85	
Num_of_Inactive_Install_Bases		0.01	1.01	0.07	-0.11	
STRATEGIC_ACCOUNTS		0.42	1.53	0.04	0.34	
Frequency		-0.20	0.82	0.03	-0.25	
Num_of_Trxns		-2.41	0.09	0.17	-2.75	
Avg_Margin		-0.55	0.58	0.09	-0.72	
Avg_Quantity		0.70	2.02	0.10	0.50	
Types_of_Product_Family		0.04	1.04	0.06	-0.07	
Types_of_Product_Model		-0.20	0.82	0.06	-0.31	
Avg_Price_Index		-0.06	0.94	0.05	-0.16	
SHORT_VERTICAL_BAKED GOODS & CEREALS		0.23	1.26	0.07	0.08	
SHORT_VERTICAL_BEVERAGE		-0.04	0.96	0.07	-0.17	
SHORT_VERTICAL_BUILDING MATERIALS		-0.10	0.90	0.04	-0.18	
SHORT_VERTICAL_CANDY & CONFECTION		0.11	1.12	0.05	0.01	
SHORT_VERTICAL_CHEMICALS		0.13	1.14	0.05	0.04	
SHORT_VERTICAL_COSMETICS / PERSONAL CARE		0.22	1.24	0.06	0.10	
SHORT_VERTICAL_DAIRY & EGGS		-0.01	0.99	0.08	-0.16	
SHORT_VERTICAL_DISTRIBUTOR		0.15	1.16	0.03	0.09	
SHORT_VERTICAL_ELECTRICAL / ELECTRONICS		0.06	1.06	0.02	0.02	
SHORT_VERTICAL_EXTRUSION / WIRE & CABLE		0.01	1.01	0.04	-0.06	
SHORT_VERTICAL_FISH & SEAFOOD		-0.02	0.98	0.04	-0.09	
SHORT_VERTICAL_FROZEN PREPARED MEALS		0.03	1.03	0.05	-0.06	
SHORT_VERTICAL_FRUIT & VEGETABLE		-0.11	0.90	0.07	-0.24	
SHORT_VERTICAL_GRAPHICS		0.06	1.07	0.07	-0.07	
SHORT_VERTICAL_INDUSTRIAL EQUIPMENT		0.14	1.15	0.09	-0.04	
SHORT_VERTICAL_MEAT & POULTRY		0.02	1.02	0.06	-0.09	
SHORT_VERTICAL_OEM-INDUSTRIAL EQUIPMENT		0.03	1.03	0.03	-0.03	
SHORT_VERTICAL_OTHER		0.12	1.13	0.06	0.01	
SHORT_VERTICAL_OTHER FOOD		0.12	1.13	0.09	-0.05	
SHORT_VERTICAL_PACKAGING MATERIALS		-0.45	0.64	0.90	-2.22	
SHORT_VERTICAL_PET FOOD & ANIMAL FEED		-0.08	0.93	0.03	-0.14	
SHORT_VERTICAL_PHARMA & MEDICAL		0.06	1.06	0.05	-0.04	
SHORT_VERTICAL_POSTAL		-1.01	0.36	0.63	-2.26	
SHORT_VERTICAL_SALTY SNACKS		0.23	1.26	0.04	0.16	
SHORT_VERTICAL_TEXTILE		0.20	1.22	0.02	0.15	
SHORT_VERTICAL_TOBACCO		-0.31	0.73	0.88	-2.03	
SHORT_VERTICAL_UNKNOWN	102	0.17	1.19	0.04	0.10	
CUSTOMER_CLASS_END USER		-0.90	0.41	0.13	-1.16	
CUSTOMER_CLASS_OEM		-0.21	0.81	0.05	-0.31	
TERRITORY_TYPE_Industrial		0.05	0.95	0.03	0.11	

```
[56]: lcm_final_table = cph2.summary[cph2.summary['p']<0.005].
    ↪sort_values(by='exp(coef)', ascending=False).head(10)
```

4 Model building for TIJ

```
[58]: tij.reset_index(inplace=True, drop=True)
```

```
[59]: tij.head()
```

	Site_Level_Price_Index	TRX_AMT_USD	Margin	QUANTITY	Total_SVC_Incidents	\
0	0.80	246.57	146.87	3	57.00	
1	0.80	209.58	109.88	3	57.00	
2	0.75	823.20	575.52	8	1.00	
3	0.75	1,029.00	719.40	10	1.00	
4	0.75	926.10	647.46	9	1.00	

	Total_Repeat_Calls	Total_FTF_Calls	Total_Visits	Total_Cases	\
0	14.00	43.00	70.00	1.00	
1	14.00	43.00	70.00	1.00	
2	0.00	1.00	13.00	1.00	
3	0.00	1.00	13.00	1.00	
4	0.00	1.00	13.00	1.00	

	Num_of_Active_Install_Bases	Total_Contracts	Contract_length	\
0	15.00	11.00	521.64	
1	15.00	11.00	521.64	
2	2.00	0.00	0.00	
3	2.00	0.00	0.00	
4	2.00	0.00	0.00	

	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	Frequency	Num_of_Trxns	\
0	0.00	0	11.38	174	
1	0.00	0	11.38	174	
2	0.00	0	61.94	19	
3	0.00	0	61.94	19	
4	0.00	0	61.94	19	

	Avg_Margin	Avg_Quantity	Types_of_Product_Family	Types_of_Product_Model	\
0	103.28	4.43	3	4	
1	103.28	4.43	3	4	
2	605.25	8.84	1	1	
3	605.25	8.84	1	1	
4	605.25	8.84	1	1	

	Avg_Price_Index	Tenure	Churned_365	\
0	1.25	1,974.00	0	

1	1.25	1,974.00	0
2	0.75	1,164.00	0
3	0.75	1,164.00	0
4	0.75	1,164.00	0
0	SHORT_VERTICAL_BAKED GOODS & CEREALS	SHORT_VERTICAL_BEVERAGE	\
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
0	SHORT_VERTICAL_BUILDING MATERIALS	SHORT_VERTICAL_CANDY & CONFECTION	\
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
0	SHORT_VERTICAL_CHEMICALS	SHORT_VERTICAL_COSMETICS / PERSONAL CARE	\
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
0	SHORT_VERTICAL_DAIRY & EGGS	SHORT_VERTICAL_DISTRIBUTOR	\
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
0	SHORT_VERTICAL_ELECTRICAL / ELECTRONICS	\	
1	0		
2	0		
3	0		
4	0		
0	SHORT_VERTICAL_EXTRUSION / WIRE & CABLE	SHORT_VERTICAL_FISH & SEAFOOD	\
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

	SHORT_VERTICAL_FROZEN PREPARED MEALS	SHORT_VERTICAL_FRUIT & VEGETABLE	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_GRAPHICS	SHORT_VERTICAL_INDUSTRIAL EQUIPMENT	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_MEAT & POULTRY	SHORT_VERTICAL_OEM-INDUSTRIAL EQUIPMENT	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_OTHER	SHORT_VERTICAL_OTHER FOOD	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_PACKAGING MATERIALS	SHORT_VERTICAL_PET FOOD & ANIMAL FEED	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	SHORT_VERTICAL_PHARMA & MEDICAL	SHORT_VERTICAL_POSTAL	\
0	1	0	
1	1	0	
2	1	0	
3	1	0	
4	1	0	
	SHORT_VERTICAL_SALTY SNACKS	SHORT_VERTICAL_TEXTILE	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	

4	0	0	
0	0	0	1
1	0	0	1
2	0	0	1
3	0	0	1
4	0	0	1
CUSTOMER_CLASS_OEM	TERRITORY_TYPE_Industrial	TERRITORY_TYPE_Postal	\
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
SUPPLIES_SEGMENTATION_M	SUPPLIES_SEGMENTATION_S	\	
0	1	0	
1	1	0	
2	0	1	
3	0	1	
4	0	1	
SUPPLIES_SEGMENTATION_Unclass	SUPPLIES_SEGMENTATION_XL	\	
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
SUPPLIES_DECLINE_REASON_Financial Distress/Credit Hold	\		
0	0		
1	0		
2	0		
3	0		
4	0		
SUPPLIES_DECLINE_REASON_Migration to 1000 Line/TIJ/TTO/LCM/LPA	\		
0	0		
1	0		
2	0		
3	0		
4	0		
SUPPLIES_DECLINE_REASON_Migration to Lasers	\		
0	0		
1	0		

2		0
3		0
4		0
SUPPLIES_DECLINE_REASONs_Moved Equipment \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_No More Coding Requirement \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_None SUPPLIES_DECLINE_REASONs_Off Brand \		
0	1	0
1	1	0
2	0	0
3	0	0
4	0	0
SUPPLIES_DECLINE_REASONs_Over Stocked / Timing \		
0	0	
1	0	
2	1	
3	1	
4	1	
SUPPLIES_DECLINE_REASONs_Pricing / Discounting \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_Printing/EQ downtime Issues \		
0	0	
1	0	
2	0	
3	0	
4	0	
SUPPLIES_DECLINE_REASONs_Production / Code Reduction \		

0	0
1	0
2	0
3	0
4	0
SUPPLIES_DECLINE_REASONs_Production Down (timing) \	
0	0
1	0
2	0
3	0
4	0
SUPPLIES_DECLINE_REASONs_Project Based \	
0	0
1	0
2	0
3	0
4	0
SUPPLIES_DECLINE_REASONs_Recent Regain/Win-back \	
0	0
1	0
2	0
3	0
4	0
SUPPLIES_DECLINE_REASONs_Seasonal Producer \	
0	0
1	0
2	0
3	0
4	0
SUPPLIES_DECLINE_REASONs_Served by Authorized Distributor \	
0	0
1	0
2	0
3	0
4	0
SUPPLIES_DECLINE_REASONs_Site Closed \	
0	0
1	0
2	0
3	0
4	0

	SUPPLIES_DECLINE_REASON_VJ	Operations	Issues	SALES_CHANNEL_Copy	\
0			0	0	
1			0	0	
2			0	0	
3			0	0	
4			0	0	
	SALES_CHANNEL_ECOMM	PO IMPORT	SALES_CHANNEL_EDI	SALES_CHANNEL_Esker	\
0		0	1	0	
1		0	0	0	
2		0	1	0	
3		0	0	0	
4		0	0	0	
	SALES_CHANNEL_IStore	Account	SALES_CHANNEL_OCC	SALES_CHANNEL_Online	\
0		0	0	0	
1		0	0	0	
2		0	0	0	
3		0	0	1	
4		0	0	1	
	SALES_CHANNEL_SFDC_CPQ	SALES_CHANNEL_Service	Billing	PRODUCT_FAMILY_CIJ	\
0		0	0	0	
1		0	0	0	
2		0	0	0	
3		0	0	0	
4		0	0	0	
	PRODUCT_FAMILY_GRAPHICS	PRODUCT_FAMILY_GRAPHICS	BA	PRODUCT_FAMILY_LASER	\
0		0	0	0	
1		0	0	0	
2		0	0	0	
3		0	0	0	
4		0	0	0	
	PRODUCT_FAMILY_LCM	PRODUCT_FAMILY_LPA	PRODUCT_FAMILY_RAW MATERIAL		\
0		0	0	0	
1		0	0	0	
2		0	0	0	
3		0	0	0	
4		0	0	0	
	PRODUCT_FAMILY_TIJ	PRODUCT_FAMILY_TTO	PRODUCT_MODEL_CLEANING SOLUTION		\
0		1	0	0	
1		1	0	0	
2		1	0	0	

3	1	0	0
4	1	0	0
PRODUCT_MODEL_FUME EXTRACTION PRODUCT_MODEL_INK PRODUCT_MODEL_LABELS \			
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
PRODUCT_MODEL_MAKE-UP PRODUCT_MODEL_PACKAGING PRODUCT_MODEL_RIBBONS \			
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
PRODUCT_MODEL_SOLVENT PRODUCT_MODEL_VALUE PACK \			
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
Most_Frequent_Interaction_Type_Callback \			
0	0		
1	0		
2	0		
3	0		
4	0		
Most_Frequent_Interaction_Type_Contact Customer \			
0	0		
1	0		
2	0		
3	0		
4	0		
Most_Frequent_Interaction_Type_Customer Meeting \			
0	0		
1	0		
2	0		
3	0		
4	0		
Most_Frequent_Interaction_Type_Dial Most_Frequent_Interaction_Type_Email \			
0	0	0	

1	0	0
2	0	0
3	0	0
4	0	0
0	Most_Frequent_Interaction_Type_Make Qualified Sales Call \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Most_Frequent_Interaction_Type_Meeting \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Most_Frequent_Interaction_Type_Other \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Most_Frequent_Interaction_Type_TS Task \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Max_Case_Origin_CX Survey Detractor \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0
0	Max_Case_Origin_Email - VTI CC Sales Escalations \ 0	0
1	0	0
2	0	0
3	0	0
4	0	0

	Max_Case_Origin_Email - VTI NACC	Max_Case_Origin_Email/Fax - VTI CS \	
0	1	0	
1	1	0	
2	0	0	
3	0	0	
4	0	0	
	Max_Case_Origin_FS Survey Followup	Max_Case_Origin_Install Complete \	
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	Max_Case_Origin_Phone	Max_Case_Origin_TS Survey Followup \	
0	0	0	
1	0	0	
2	1	0	
3	1	0	
4	1	0	
	Max_Case_Origin_unknown	Max_Case_Reason_CX: Customer Care \	
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	Max_Case_Reason_CX: Field Sales	Max_Case_Reason_CX: Field Service \	
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	Max_Case_Reason_CX: Manufacturing	Max_Case_Reason_CX: Other Team \	
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
	Max_Case_Reason_CX: Tech Support	Max_Case_Reason_Customer Experience \	
0	0	1	
1	0	1	
2	0	1	
3	0	1	

4

0

1

	Max_Case_Reason_unknown	Contract_Category_Full	Care	\
0	0		0	
1	0		0	
2	0		0	
3	0		0	
4	0		0	

	Contract_Category_No	Contract	Contract_Category_Supportive	\
0		0		0
1		0		0
2		1		0
3		1		0
4		1		0

	Contract_Category_WFC	TERRITORY_REGION_MW	TERRITORY_REGION_NE	\
0	0	0		0
1	0	0		0
2	0	0		1
3	0	0		1
4	0	0		1

	TERRITORY_REGION_NW	TERRITORY_REGION_SC	TERRITORY_REGION_SE	\
0	1	0		0
1	1	0		0
2	0	0		0
3	0	0		0
4	0	0		0

	Most_Frequent_Sales_Channel_Copy	Most_Frequent_Sales_Channel_EDI	\
0	0		1
1	0		1
2	0		1
3	0		1
4	0		1

	Most_Frequent_Sales_Channel_Esker	\
0	0	
1	0	
2	0	
3	0	
4	0	

	Most_Frequent_Sales_Channel_IStore	Account	\
0		0	
1		0	

2		0
3		0
4		0
	Most_Frequent_Sales_Channel_OCC	Most_Frequent_Sales_Channel_Online \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	Most_Frequent_Sales_Channel_SFDC_CPQ \	
0	0	
1	0	
2	0	
3	0	
4	0	
	Most_Frequent_Sales_Channel_Service Billing \	
0		0
1		0
2		0
3		0
4		0
	Most_Frequent_Order_Type_BILL ONLY \	
0	0	
1	0	
2	0	
3	0	
4	0	
	Most_Frequent_Order_Type_DEMO EQUIPMENT ACCEPT \	
0		0
1		0
2		0
3		0
4		0
	Most_Frequent_Order_Type_EDI	Most_Frequent_Order_Type_EQUIPMENT DOMESTIC \
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0
	Most_Frequent_Order_Type_SERVICE \	

```

0          0
1          0
2          0
3          0
4          0

    Most_Frequent_Order_Type_STANDARD DOMESTIC \
0          0
1          0
2          0
3          0
4          0

    Most_Frequent_Order_Type_STANDARD INTERNATIONAL \
0          0
1          0
2          0
3          0
4          0

    Most_Frequent_Order_Type_US FULL CARE EQPT DOMESTIC \
0          0
1          0
2          0
3          0
4          0

    Most_Frequent_Order_Type_US FULL CARE INTERNATIONAL \
0          0
1          0
2          0
3          0
4          0

    Most_Frequent_Order_Type_WEB ORDER
0          0
1          0
2          0
3          0
4          0

```

[60]: tijChurned = tij[['Churned_365']]

[61]: tij.drop(columns=['Churned_365'],axis=1,inplace=True)

[62]: #Scaling
Instantiate

```

scaler = StandardScaler()

# fit_transform
tij_scaled = scaler.fit_transform(tij)

[63]: tij_scaled = pd.DataFrame(tij_scaled)
tij_scaled.columns = tij.columns
tij_scaled.head()

```

	Site_Level_Price_Index	TRX_AMT_USD	Margin	QUANTITY	Total_SVC_Incidents	\
0	-0.47	-0.55	-0.54	-0.48	0.22	
1	-0.47	-0.57	-0.56	-0.48	0.22	
2	-0.62	-0.30	-0.23	-0.38	-0.66	
3	-0.62	-0.21	-0.13	-0.34	-0.66	
4	-0.62	-0.25	-0.18	-0.36	-0.66	

	Total_Repeat_Calls	Total_FTF_Calls	Total_Visits	Total_Cases	\
0	0.10	0.26	0.03	-0.90	
1	0.10	0.26	0.03	-0.90	
2	-0.60	-0.68	-0.84	-0.90	
3	-0.60	-0.68	-0.84	-0.90	
4	-0.60	-0.68	-0.84	-0.90	

	Num_of_Active_Install_Bases	Total_Contracts	Contract_length	\
0	0.26	0.45	0.26	
1	0.26	0.45	0.26	
2	-0.44	-0.48	-0.72	
3	-0.44	-0.48	-0.72	
4	-0.44	-0.48	-0.72	

	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	Frequency	Num_of_Trxns	\
0	-0.22	-0.51	-0.58	0.65	
1	-0.22	-0.51	-0.58	0.65	
2	-0.22	-0.51	0.05	-0.62	
3	-0.22	-0.51	0.05	-0.62	
4	-0.22	-0.51	0.05	-0.62	

	Avg_Margin	Avg_Quantity	Types_of_Product_Family	Types_of_Product_Model	\
0	-0.68	-0.63	1.20	1.39	
1	-0.68	-0.63	1.20	1.39	
2	-0.21	-0.48	-1.06	-0.98	
3	-0.21	-0.48	-1.06	-0.98	
4	-0.21	-0.48	-1.06	-0.98	

	Avg_Price_Index	Tenure	SHORT_VERTICAL_BAKED_GOODS & CEREALS	\
0	0.91	0.75	-0.13	
1	0.91	0.75	-0.13	

2	-0.80	-0.73	-0.13
3	-0.80	-0.73	-0.13
4	-0.80	-0.73	-0.13
0	SHORT_VERTICAL_BEVERAGE	SHORT_VERTICAL_BUILDING MATERIALS	\
1	-0.16	-0.17	
2	-0.16	-0.17	
3	-0.16	-0.17	
4	-0.16	-0.17	
0	SHORT_VERTICAL_CANDY & CONFECTION	SHORT_VERTICAL_CHEMICALS	\
1	-0.08	-0.07	
2	-0.08	-0.07	
3	-0.08	-0.07	
4	-0.08	-0.07	
0	SHORT_VERTICAL_COSMETICS / PERSONAL CARE	SHORT_VERTICAL_DAIRY & EGGS	\
1	-0.09	-0.33	
2	-0.09	-0.33	
3	-0.09	-0.33	
4	-0.09	-0.33	
0	SHORT_VERTICAL_DISTRIBUTOR	SHORT_VERTICAL_ELECTRICAL / ELECTRONICS	\
1	0.00	-0.11	
2	0.00	-0.11	
3	0.00	-0.11	
4	0.00	-0.11	
0	SHORT_VERTICAL_EXTRUSION / WIRE & CABLE	SHORT_VERTICAL_FISH & SEAFOOD	\
1	0.00	-0.03	
2	0.00	-0.03	
3	0.00	-0.03	
4	0.00	-0.03	
0	SHORT_VERTICAL_FROZEN PREPARED MEALS	SHORT_VERTICAL_FRUIT & VEGETABLE	\
1	-0.05	-0.05	
2	-0.05	-0.05	
3	-0.05	-0.05	
4	-0.05	-0.05	
	SHORT_VERTICAL_GRAPHICS	SHORT_VERTICAL_INDUSTRIAL EQUIPMENT	\

0	-0.59	-0.13
1	-0.59	-0.13
2	-0.59	-0.13
3	-0.59	-0.13
4	-0.59	-0.13
0	SHORT_VERTICAL_MEAT & POULTRY	SHORT_VERTICAL_OEM-INDUSTRIAL EQUIPMENT \
1	-0.17	-0.16
2	-0.17	-0.16
3	-0.17	-0.16
4	-0.17	-0.16
0	SHORT_VERTICAL_OTHER	SHORT_VERTICAL_OTHER FOOD \
1	-0.27	-0.22
2	-0.27	-0.22
3	-0.27	-0.22
4	-0.27	-0.22
0	SHORT_VERTICAL_PACKAGING MATERIALS	SHORT_VERTICAL_PET FOOD & ANIMAL FEED \
1	-0.03	-0.05
2	-0.03	-0.05
3	-0.03	-0.05
4	-0.03	-0.05
0	SHORT_VERTICAL_PHARMA & MEDICAL	SHORT_VERTICAL_POSTAL \
1	1.71	0.00
2	1.71	0.00
3	1.71	0.00
4	1.71	0.00
0	SHORT_VERTICAL_SALTY SNACKS	SHORT_VERTICAL_TEXTILE \
1	0.00	-0.09
2	0.00	-0.09
3	0.00	-0.09
4	0.00	-0.09
0	SHORT_VERTICAL_TOBACCO	SHORT_VERTICAL_UNKNOWN CUSTOMER_CLASS_END USER \
1	-0.24	-0.15 0.24
2	-0.24	-0.15 0.24
3	-0.24	-0.15 0.24
4	-0.24	-0.15 0.24

	CUSTOMER_CLASS_OEM	TERRITORY_TYPE_Industrial	TERRITORY_TYPE_Postal	\
0	-0.23	0.55	-0.02	
1	-0.23	0.55	-0.02	
2	-0.23	0.55	-0.02	
3	-0.23	0.55	-0.02	
4	-0.23	0.55	-0.02	
	SUPPLIES_SEGMENTATION_M	SUPPLIES_SEGMENTATION_S	\	
0	3.04	-0.86		
1	3.04	-0.86		
2	-0.33	1.17		
3	-0.33	1.17		
4	-0.33	1.17		
	SUPPLIES_SEGMENTATION_Unclass	SUPPLIES_SEGMENTATION_XL	\	
0	0.00	-0.59		
1	0.00	-0.59		
2	0.00	-0.59		
3	0.00	-0.59		
4	0.00	-0.59		
	SUPPLIES_DECLINE_REASON_Financial	Distress/Credit Hold	\	
0	0.00			
1	0.00			
2	0.00			
3	0.00			
4	0.00			
	SUPPLIES_DECLINE_REASON_Migration_to_1000	Line/TIJ/TTO/LCM/LPA	\	
0		-0.11		
1		-0.11		
2		-0.11		
3		-0.11		
4		-0.11		
	SUPPLIES_DECLINE_REASON_Migration_to_Lasers	\		
0		-0.11		
1		-0.11		
2		-0.11		
3		-0.11		
4		-0.11		
	SUPPLIES_DECLINE_REASON_Moved_Equipment	\		
0		-0.08		
1		-0.08		
2		-0.08		

3		-0.08
4		-0.08
SUPPLIES_DECLINE_REASON_No More Coding Requirement \		
0	0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
SUPPLIES_DECLINE_REASON_None SUPPLIES_DECLINE_REASON_Off Brand \		
0	0.84	-0.23
1	0.84	-0.23
2	-1.19	-0.23
3	-1.19	-0.23
4	-1.19	-0.23
SUPPLIES_DECLINE_REASON_Over Stocked / Timing \		
0	-0.44	
1	-0.44	
2	2.28	
3	2.28	
4	2.28	
SUPPLIES_DECLINE_REASON_Pricing / Discounting \		
0	0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
SUPPLIES_DECLINE_REASON_Printing/EQ downtime Issues \		
0	0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
SUPPLIES_DECLINE_REASON_Production / Code Reduction \		
0	-0.25	
1	-0.25	
2	-0.25	
3	-0.25	
4	-0.25	
SUPPLIES_DECLINE_REASON_Production Down (timing) \		
0	-0.17	

1		-0.17
2		-0.17
3		-0.17
4		-0.17
	SUPPLIES_DECLINE_REASONs_Project Based \	
0		-0.04
1		-0.04
2		-0.04
3		-0.04
4		-0.04
	SUPPLIES_DECLINE_REASONs_Recent Regain/Win-back \	
0		-0.11
1		-0.11
2		-0.11
3		-0.11
4		-0.11
	SUPPLIES_DECLINE_REASONs_Seasonal Producer \	
0		0.00
1		0.00
2		0.00
3		0.00
4		0.00
	SUPPLIES_DECLINE_REASONs_Served by Authorized Distributor \	
0		-0.08
1		-0.08
2		-0.08
3		-0.08
4		-0.08
	SUPPLIES_DECLINE_REASONs_Site Closed \	
0		-0.11
1		-0.11
2		-0.11
3		-0.11
4		-0.11
	SUPPLIES_DECLINE_REASONs_VJ Operations Issues SALES_CHANNEL_Copy \	
0		0.00 -0.16
1		0.00 -0.16
2		0.00 -0.16
3		0.00 -0.16
4		0.00 -0.16

	SALES_CHANNEL_ECOMM	PO IMPORT	SALES_CHANNEL_EDI	SALES_CHANNEL_Esker	\
0		0.00	10.06	-0.61	
1		0.00	-0.10	-0.61	
2		0.00	10.06	-0.61	
3		0.00	-0.10	-0.61	
4		0.00	-0.10	-0.61	
	SALES_CHANNEL_IStore	Account	SALES_CHANNEL_OCC	SALES_CHANNEL_Online	\
0		-0.09	-0.22	-1.15	
1		-0.09	-0.22	-1.15	
2		-0.09	-0.22	-1.15	
3		-0.09	-0.22	0.87	
4		-0.09	-0.22	0.87	
	SALES_CHANNEL_SFDC_CPQ	SALES_CHANNEL_Service	Billing	PRODUCT_FAMILY_CIJ	\
0		-0.10	0.00	0.00	
1		-0.10	0.00	0.00	
2		-0.10	0.00	0.00	
3		-0.10	0.00	0.00	
4		-0.10	0.00	0.00	
	PRODUCT_FAMILY_GRAPHICS	PRODUCT_FAMILY_GRAPHICS	BA	PRODUCT_FAMILY_LASER	\
0		0.00	0.00	0.00	
1		0.00	0.00	0.00	
2		0.00	0.00	0.00	
3		0.00	0.00	0.00	
4		0.00	0.00	0.00	
	PRODUCT_FAMILY_LCM	PRODUCT_FAMILY_LPA	PRODUCT_FAMILY_RAW_MATERIAL		\
0		0.00	0.00	0.00	
1		0.00	0.00	0.00	
2		0.00	0.00	0.00	
3		0.00	0.00	0.00	
4		0.00	0.00	0.00	
	PRODUCT_FAMILY_TIJ	PRODUCT_FAMILY_TTO	PRODUCT_MODEL_CLEANING SOLUTION		\
0		0.00	0.00	0.00	
1		0.00	0.00	0.00	
2		0.00	0.00	0.00	
3		0.00	0.00	0.00	
4		0.00	0.00	0.00	
	PRODUCT_MODEL_FUME_EXTRACTION	PRODUCT_MODEL_INK	PRODUCT_MODEL_LABELS		\
0		0.00	0.00	0.00	
1		0.00	0.00	0.00	
2		0.00	0.00	0.00	
3		0.00	0.00	0.00	

4	0.00	0.00	0.00
	PRODUCT_MODEL_MAKE-UP	PRODUCT_MODEL_PACKAGING	PRODUCT_MODEL_RIBBONS \
0	0.00	0.00	0.00
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.00	0.00	0.00
	PRODUCT_MODEL_SOLVENT	PRODUCT_MODEL_VALUE_PACK \	
0	0.00	0.00	
1	0.00	0.00	
2	0.00	0.00	
3	0.00	0.00	
4	0.00	0.00	
	Most_Frequent_Interaction_Type_Callback \		
0		0.00	
1		0.00	
2		0.00	
3		0.00	
4		0.00	
	Most_Frequent_Interaction_Type_Contact Customer \		
0		0.00	
1		0.00	
2		0.00	
3		0.00	
4		0.00	
	Most_Frequent_Interaction_Type_Customer Meeting \		
0		-0.04	
1		-0.04	
2		-0.04	
3		-0.04	
4		-0.04	
	Most_Frequent_Interaction_Type_Dial Most_Frequent_Interaction_Type_Email \		
0	0.00		-0.14
1	0.00		-0.14
2	0.00		-0.14
3	0.00		-0.14
4	0.00		-0.14
	Most_Frequent_Interaction_Type_Make Qualified Sales Call \		
0		0.00	
1		0.00	

2		0.00
3		0.00
4		0.00
	Most_Frequent_Interaction_Type_Meeting \	
0		-0.06
1		-0.06
2		-0.06
3		-0.06
4		-0.06
	Most_Frequent_Interaction_Type_Other \	
0		-0.26
1		-0.26
2		-0.26
3		-0.26
4		-0.26
	Most_Frequent_Interaction_Type_TS Task \	
0		0.00
1		0.00
2		0.00
3		0.00
4		0.00
	Max_Case-Origin_CX Survey Detractor Max_Case-Origin_Email \	
0		0.00
1		0.00
2		0.00
3		0.00
4		0.00
	Max_Case-Origin_Email - VTI CC Sales Escalations \	
0		0.00
1		0.00
2		0.00
3		0.00
4		0.00
	Max_Case-Origin_Email - VTI NACC Max_Case-Origin_Email/Fax - VTI CS \	
0		2.06
1		2.06
2		-0.49
3		-0.49
4		-0.49
	Max_Case-Origin_FS Survey Followup Max_Case-Origin_Install Complete \	

0		0.00	-0.02
1		0.00	-0.02
2		0.00	-0.02
3		0.00	-0.02
4		0.00	-0.02
	Max_Case_Origin_Phone	Max_Case_Origin_TS Survey Followup	\
0	-0.30	-0.11	
1	-0.30	-0.11	
2	3.30	-0.11	
3	3.30	-0.11	
4	3.30	-0.11	
	Max_Case_Origin_unknown	Max_Case_Reason_CX: Customer Care	\
0	-1.46	-0.16	
1	-1.46	-0.16	
2	-1.46	-0.16	
3	-1.46	-0.16	
4	-1.46	-0.16	
	Max_Case_Reason_CX: Field Sales	Max_Case_Reason_CX: Field Service	\
0	-0.02	-0.07	
1	-0.02	-0.07	
2	-0.02	-0.07	
3	-0.02	-0.07	
4	-0.02	-0.07	
	Max_Case_Reason_CX: Manufacturing	Max_Case_Reason_CX: Other Team	\
0	0.00	0.00	
1	0.00	0.00	
2	0.00	0.00	
3	0.00	0.00	
4	0.00	0.00	
	Max_Case_Reason_CX: Tech Support	Max_Case_Reason_Customer Experience	\
0	-0.13	1.63	
1	-0.13	1.63	
2	-0.13	1.63	
3	-0.13	1.63	
4	-0.13	1.63	
	Max_Case_Reason_unknown	Contract_Category_Full Care	\
0	-1.46	-0.18	
1	-1.46	-0.18	
2	-1.46	-0.18	
3	-1.46	-0.18	
4	-1.46	-0.18	

	Contract_Category_No	Contract	Contract_Category_Supportive	\
0		-1.12	0.00	
1		-1.12	0.00	
2		0.89	0.00	
3		0.89	0.00	
4		0.89	0.00	
	Contract_Category_WFC	TERRITORY_REGION_MW	TERRITORY_REGION_NE	\
0		0.00	-0.64	-0.50
1		0.00	-0.64	-0.50
2		0.00	-0.64	1.99
3		0.00	-0.64	1.99
4		0.00	-0.64	1.99
	TERRITORY_REGION_NW	TERRITORY_REGION_SC	TERRITORY_REGION_SE	\
0		2.58	-0.34	-0.42
1		2.58	-0.34	-0.42
2		-0.39	-0.34	-0.42
3		-0.39	-0.34	-0.42
4		-0.39	-0.34	-0.42
	Most_Frequent_Sales_Channel_Copy	Most_Frequent_Sales_Channel_EDI	\	
0		-0.07	9.42	
1		-0.07	9.42	
2		-0.07	9.42	
3		-0.07	9.42	
4		-0.07	9.42	
	Most_Frequent_Sales_Channel_Esker	\		
0		-0.65		
1		-0.65		
2		-0.65		
3		-0.65		
4		-0.65		
	Most_Frequent_Sales_Channel_IStore	Account	\	
0		-0.09		
1		-0.09		
2		-0.09		
3		-0.09		
4		-0.09		
	Most_Frequent_Sales_Channel_OCC	Most_Frequent_Sales_Channel_Online	\	
0		-0.15	-1.27	
1		-0.15	-1.27	
2		-0.15	-1.27	

3	-0.15	-1.27
4	-0.15	-1.27
0	Most_Frequent_Sales_Channel_SFDC_CPQ \ -0.03	
1	-0.03	
2	-0.03	
3	-0.03	
4	-0.03	
0	Most_Frequent_Sales_Channel_Service Billing \ 0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
0	Most_Frequent_Order_Type_BILL ONLY \ 0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
0	Most_Frequent_Order_Type_DEMO EQUIPMENT ACCEPT \ 0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
0	Most_Frequent_Order_Type_EDI Most_Frequent_Order_Type_EQUIPMENT DOMESTIC \ 9.23 -0.18	
1	9.23 -0.18	
2	9.23 -0.18	
3	9.23 -0.18	
4	9.23 -0.18	
0	Most_Frequent_Order_Type_SERVICE \ 0.00	
1	0.00	
2	0.00	
3	0.00	
4	0.00	
0	Most_Frequent_Order_Type_STANDARD DOMESTIC \ -3.48	

```

1           -3.48
2           -3.48
3           -3.48
4           -3.48

    Most_Frequent_Order_Type_STANDARD INTERNATIONAL \
0                  0.00
1                  0.00
2                  0.00
3                  0.00
4                  0.00

    Most_Frequent_Order_Type_US FULL CARE EQPT DOMESTIC \
0                  0.00
1                  0.00
2                  0.00
3                  0.00
4                  0.00

    Most_Frequent_Order_Type_US FULL CARE INTERNATIONAL \
0                  0.00
1                  0.00
2                  0.00
3                  0.00
4                  0.00

    Most_Frequent_Order_Type_WEB ORDER
0                 -0.18
1                 -0.18
2                 -0.18
3                 -0.18
4                 -0.18

```

[64]: tij_final = pd.concat([tij_scaled,tijChurned],axis=1,sort=False)

[65]:

```

tij_final = tij_final.drop(['SHORT_VERTICAL_DISTRIBUTOR', u
→'SHORT_VERTICAL_EXTRUSION / WIRE & CABLE', 'SHORT_VERTICAL_POSTAL', u
→'SHORT_VERTICAL_SALTY_SNACKS', 'SUPPLIES_SEGMENTATION_Unclass', u
→'SUPPLIES_DECLINE_REASON_Financial Distress/Credit Hold', u
→'SUPPLIES_DECLINE_REASON_No More Coding Requirement', u
→'SUPPLIES_DECLINE_REASON_Pricing / Discounting', u
→'SUPPLIES_DECLINE_REASON_Printing/EQ downtime Issues', u
→'SUPPLIES_DECLINE_REASON_Seasonal Producer', 'SUPPLIES_DECLINE_REASON_VJ u
→Operations Issues', 'SALES_CHANNEL_ECOMM PO IMPORT', 'SALES_CHANNEL_Service u
→Billing', 'PRODUCT_FAMILY_CIJ', 'PRODUCT_FAMILY_GRAPHICS', u
→'PRODUCT_FAMILY_GRAPHICS_BA', 'PRODUCT_FAMILY_LASER', 'PRODUCT_FAMILY_LCM', u
→'PRODUCT_FAMILY_LPA', 'PRODUCT_FAMILY_RAW_MATERIAL', 'PRODUCT_FAMILY_TIJ', u
→'PRODUCT_FAMILY_TTO', 'PRODUCT_MODEL_CLEANING SOLUTION', 'PRODUCT_MODEL_FUME u
→EXTRACTION', 'PRODUCT_MODEL_INK', 'PRODUCT_MODEL_LABELS', u
→'PRODUCT_MODEL_MAKE-UP', 'PRODUCT_MODEL_PACKAGING', 'PRODUCT_MODEL_RIBBONS', u
→'PRODUCT_MODEL_SOLVENT', 'PRODUCT_MODEL_VALUE_PACK', u
→'Most_Frequent_Interaction_Type_Callback', u
→'Most_Frequent_Interaction_Type_Contact_Customer', u
→'Most_Frequent_Interaction_Type_Dial', 'Most_Frequent_Interaction_Type_Make u
→Qualified Sales Call', 'Most_Frequent_Interaction_Type_TS_Task', u
→'Max_Case_Origin_CX_Survey_Detractor', 'Max_Case_Origin_Email', u
→'Max_Case_Origin_Email - VTI_CC_Sales_Escalations', 'Max_Case_Origin_FS u
→Survey_Followup', 'Max_Case_Reason_CX_Manufacturing', 'Max_Case_Reason_CX: u
→Other Team', 'Contract_Category_Supportive', 'Contract_Category_WFC', u
→'Most_Frequent_Sales_Channel_Service_Billing', u
→'Most_Frequent_Order_Type_BILL_ONLY', 'Most_Frequent_Order_Type_DEMO u
→EQUIPMENT_ACCEPT', 'Most_Frequent_Order_Type_SERVICE', u
→'Most_Frequent_Order_Type_STANDARD INTERNATIONAL', u
→'Most_Frequent_Order_Type_US_FULL CARE_EQPT_DOMESTIC', u
→'Most_Frequent_Order_Type_US_FULL CARE INTERNATIONAL'
], axis=1)

```

[66]: tij_final.shape

[66]: (4395, 106)

[67]: cph3 = lifelines.CoxPHFitter(penalizer=0.1)
cph3.fit(tij_final, step_size=0.05, duration_col='Tenure', u
→event_col='Churned_365', show_progress=False)
cph3.print_summary()

covariate		coef	exp(coef)	se(coef)	coef lower	95%
Site_Level_Price_Index		-0.03	0.97	0.04	-0.11	
TRX_AMT_USD		-0.03	0.97	0.04	-0.12	
Margin		-0.04	0.96	0.04	-0.12	
QUANTITY		-0.01	0.99	0.04	-0.10	
Total_SVC_Incidents		0.02	1.02	0.04	-0.07	
Total_Repeat_Calls		0.05	1.05	0.04	-0.04	
Total_FTF_Calls		-0.00	1.00	0.04	-0.09	
Total_Visits		-0.13	0.88	0.04	-0.21	
Total_Cases		0.03	1.03	0.03	-0.02	
Num_of_Active_Install_Bases		-0.17	0.84	0.04	-0.26	
Total_Contracts		-0.09	0.91	0.05	-0.18	
Contract_length		-0.05	0.95	0.04	-0.13	
Num_of_Inactive_Install_Bases		-0.03	0.97	0.04	-0.12	
STRATEGIC_ACCOUNTS		-0.01	0.99	0.04	-0.09	
Frequency		-0.12	0.88	0.03	-0.18	
Num_of_Trxns		-0.18	0.83	0.04	-0.27	
Avg_Margin		-0.11	0.90	0.04	-0.19	
Avg_Quantity		-0.05	0.95	0.04	-0.13	
Types_of_Product_Family		-0.09	0.91	0.04	-0.17	
Types_of_Product_Model		-0.08	0.92	0.04	-0.16	
Avg_Price_Index		-0.04	0.96	0.04	-0.12	
SHORT_VERTICAL_BAKED GOODS & CEREALS		-0.02	0.98	0.04	-0.11	
SHORT_VERTICAL_BEVERAGE		-0.11	0.89	0.04	-0.18	
SHORT_VERTICAL_BUILDING MATERIALS		0.08	1.08	0.03	0.02	
SHORT_VERTICAL_CANDY & CONFECTION		0.08	1.08	0.03	0.02	
SHORT_VERTICAL_CHEMICALS		-0.04	0.96	0.04	-0.11	
SHORT_VERTICAL_COSMETICS / PERSONAL CARE		0.09	1.10	0.03	0.03	
SHORT_VERTICAL_DAIRY & EGGS		-0.09	0.91	0.04	-0.18	
SHORT_VERTICAL_ELECTRICAL / ELECTRONICS		0.05	1.05	0.03	-0.00	
SHORT_VERTICAL_FISH & SEAFOOD		-0.01	0.99	0.04	-0.09	
SHORT_VERTICAL_FROZEN PREPARED MEALS		-0.01	0.99	0.05	-0.09	
SHORT_VERTICAL_FRUIT & VEGETABLE		0.04	1.04	0.02	-0.00	
SHORT_VERTICAL_GRAPHICS		0.01	1.01	0.04	-0.07	
SHORT_VERTICAL_INDUSTRIAL EQUIPMENT		0.03	1.03	0.03	-0.03	
SHORT_VERTICAL_MEAT & POULTRY		0.09	1.09	0.03	0.02	
SHORT_VERTICAL_OEM-INDUSTRIAL EQUIPMENT		-0.01	0.99	0.03	-0.08	
SHORT_VERTICAL_OTHER		-0.07	0.93	0.04	-0.14	
SHORT_VERTICAL_OTHER FOOD		-0.00	1.00	0.04	-0.08	
SHORT_VERTICAL_PACKAGING MATERIALS		0.07	1.07	0.02	0.04	
SHORT_VERTICAL_PET FOOD & ANIMAL FEED		-0.01	0.99	0.04	-0.10	
SHORT_VERTICAL_PHARMA & MEDICAL		-0.02	0.98	0.04	-0.10	
SHORT_VERTICAL_TEXTILE		0.04	1.04	0.02	0.00	
SHORT_VERTICAL_TOBACCO		0.01	1.01	0.04	-0.07	
SHORT_VERTICAL_UNKNOWN		0.14	1.15	0.03	0.08	
CUSTOMER_CLASS_END USER		0.07	1.07	0.04	-0.01	
CUSTOMER_CLASS_OEM		-0.05	0.95	0.04	-0.12	
TERRITORY_TYPE_Industrial	130	0.02	1.02	0.04	-0.06	
TERRITORY_TYPE_Postal		0.05	1.05	0.02	0.01	
SUPPLIES_SEGMENTATION_M		-0.06	0.94	0.04	-0.13	
SUPPLIES_SEGMENTATION_S		0.32	1.38	0.04	0.24	
SUPPLIES_SEGMENTATION_XI		0.12	0.89	0.04	0.20	

```
[68]: tij_final_table = cph3.summary[cph3.summary['p']<0.005].  
      ↪sort_values(by='exp(coef)', ascending=False).head(10)
```

5 Exporting Results

```
[69]: cij_final_table.to_csv("cij_significant_factors.csv")  
tto_final_table.to_csv("tto_significant_factors.csv")  
lcm_final_table.to_csv("lcm_significant_factors.csv")  
tij_final_table.to_csv("tij_significant_factors.csv")
```

Part 9.2 - Survival Analysis Churn - All Transactions

February 12, 2021

0.1 Data Preprocessing

0.1.1 1. Import Dataset and relevant Python libraries

```
[1]: # import required libraries for dataframe and visualization
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from lifelines import KaplanMeierFitter
# Ensuring all rows and columns are visible
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
pd.set_option('float_format', '{:,.2f}'.format)

# Reading the data on which analysis needs to be done

df = pd.read_csv('analysis.csv')
df = df.iloc[:,1:]
df.head()
```

```
[1]: Site_Level_Price_Index CUSTOMER_ID CUSTOMER_SITE_ID SHORT_VERTICAL \
```

```
0 0.79 6482 24 GRAPHICS
1 0.79 6482 24 GRAPHICS
2 0.79 6482 24 GRAPHICS
3 0.79 6482 24 GRAPHICS
4 1.40 37 90 CHEMICALS
```

```
POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE SUPPLIES_SEGMENTATION \
```

```
0 60085 END USER Industrial S
1 60085 END USER Industrial S
2 60085 END USER Industrial S
3 60085 END USER Industrial S
4 65802 END USER Industrial S
```

```
SUPPLIES_DECLINE_REASON DUNS_NUMBER TRX_DATE TRX_AMT_USD Margin \
```

```
0 None 144782380 2020-03-20 1,855.74 1,381.20
1 None 144782380 2020-04-15 552.68 537.02
```

2		None	144782380	2020-05-04	5,000.00	4,843.37
3		None	144782380	2020-05-11	5,000.00	4,843.37
4		None	43937895	2017-01-03	213.80	129.59
	SALES_CHANNEL	QUANTITY	ORDER_TYPE	ORDER_NUM	PRODUCT_FAMILY	\
0	Esker	6	STANDARD DOMESTIC	512,254,720.00	LASER	
1	SFDC_CPQ	2	EQUIPMENT DOMESTIC	512,260,527.00	LCM	
2	Online	20	STANDARD DOMESTIC	512,267,173.00	LCM	
3	Copy	20	STANDARD DOMESTIC	512,268,803.00	LCM	
4	EDI	20	EDI	511,872,093.00	CIJ	
	PRODUCT_MODEL	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\	
0	FUME EXTRACTION	13.00	7.00	6.00		
1	INK	13.00	7.00	6.00		
2	INK	13.00	7.00	6.00		
3	INK	13.00	7.00	6.00		
4	MAKE-UP	57.00	13.00	44.00		
	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	Max_Case-Origin	\	
0	Call	81.00	1.97	unknown		
1	Call	81.00	1.97	unknown		
2	Call	81.00	1.97	unknown		
3	Call	81.00	1.97	unknown		
4	Call	53.00	3.03	unknown		
	Max_Case_Reason	Num_of_Active_Install_Bases	Total_Contracts	\		
0	unknown	5.00	0.00			
1	unknown	5.00	0.00			
2	unknown	5.00	0.00			
3	unknown	5.00	0.00			
4	unknown	6.00	6.00			
	Contract_length	Contract_Category	Num_of_Inactive_Install_Bases	\		
0	0.00	No Contract	0.00			
1	0.00	No Contract	0.00			
2	0.00	No Contract	0.00			
3	0.00	No Contract	0.00			
4	1,003.00	FSMA	0.00			
	STRATEGIC_ACCOUNTS	TERRITORY_REGION	TRX_YEAR	Recency	Frequency	\
0	0	MW	2020	24	17.67	
1	0	MW	2020	24	17.67	
2	0	MW	2020	24	17.67	
3	0	MW	2020	24	17.67	
4	0	MC	2017	50	18.86	
	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	Avg_Quantity	\	

```

0          4    3,102.11   2,901.24      12.00
1          4    3,102.11   2,901.24      12.00
2          4    3,102.11   2,901.24      12.00
3          4    3,102.11   2,901.24      12.00
4         98     233.90    172.39      11.37

Mode_of_Product_Family Mode_of_Product_Model  Types_of_Product_Family \
0                      LCM                  INK                2
1                      LCM                  INK                2
2                      LCM                  INK                2
3                      LCM                  INK                2
4                     CIJ      MAKE-UP            1

Types_of_Product_Model Most_Frequent_Sales_Channel \
0                  2             Copy
1                  2             Copy
2                  2             Copy
3                  2             Copy
4                  3             EDI

Most_Frequent_Order_Type Avg_Price_Index  Tenure  Cluster_Id \
0      STANDARD DOMESTIC       0.79    77.00      3
1      STANDARD DOMESTIC       0.79    77.00      3
2      STANDARD DOMESTIC       0.79    77.00      3
3      STANDARD DOMESTIC       0.79    77.00      3
4                 EDI        1.35  1,879.00      3

churn_prob_BGNBD  Churned_365  Churned_BGNBD
0            0.05        0        0
1            0.05        0        0
2            0.05        0        0
3            0.05        0        0
4            0.02        0        0

```

```
[2]: df['Churned_BGNBD'].value_counts()
```

```
[2]: 0    316089
1    64312
Name: Churned_BGNBD, dtype: int64
```

0.1.2 2. Divide dataset into different product dataframes

```
[3]: # set date columns to datetime
df['TRX_DATE'] = pd.to_datetime(df['TRX_DATE'])
```

```
[4]: CIJ = df[df['PRODUCT_FAMILY'] == 'CIJ']
TTO = df[df['PRODUCT_FAMILY'] == 'TTO']
LCM = df[df['PRODUCT_FAMILY'] == 'LCM']
TIJ = df[df['PRODUCT_FAMILY'] == 'TIJ']

[5]: # set now to max transaction date + 1
from datetime import timedelta
NOW = max(df.TRX_DATE) + timedelta(days=1)

[6]: CIJ['TRX_DATE'] = pd.to_datetime(CIJ['TRX_DATE'])
TTO['TRX_DATE'] = pd.to_datetime(TTO['TRX_DATE'])
LCM['TRX_DATE'] = pd.to_datetime(LCM['TRX_DATE'])
TIJ['TRX_DATE'] = pd.to_datetime(TIJ['TRX_DATE'])

/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    """Entry point for launching an IPython kernel.
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

See the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
This is separate from the ipykernel package so we can avoid doing imports
until
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
after removing the cwd from sys.path.
```

CIJ_Final Creation

[7]: CIJ.head()

```
[7]:   Site_Level_Price_Index CUSTOMER_ID CUSTOMER_SITE_ID SHORT_VERTICAL \
4           1.40          37            90      CHEMICALS
5           1.40          37            90      CHEMICALS
6           1.40          37            90      CHEMICALS
7           1.40          37            90      CHEMICALS
8           1.40          37            90      CHEMICALS

    POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE SUPPLIES_SEGMENTATION \
4       65802     END USER     Industrial             S
5       65802     END USER     Industrial             S
6       65802     END USER     Industrial             S
7       65802     END USER     Industrial             S
8       65802     END USER     Industrial             S

    SUPPLIES_DECLINE_REASON DUNS_NUMBER    TRX_DATE    TRX_AMT_USD Margin \
4           None    43937895 2017-01-03      213.80  129.59
5           None    43937895 2017-09-01      222.40  138.19
6           None    43937895 2017-08-29      66.72   41.46
7           None    43937895 2017-08-29      44.48   27.64
8           None    43937895 2017-05-26      33.36   20.73

    SALES_CHANNEL QUANTITY        ORDER_TYPE ORDER_NUM PRODUCT_FAMILY \
4         EDI        20          EDI 511,872,093.00      CIJ
5         EDI        20          EDI 511,955,407.00      CIJ
6      Online        6  STANDARD DOMESTIC 511,954,133.00      CIJ
7      Online        4  STANDARD DOMESTIC 511,954,133.00      CIJ
8  BigMachine        3 EQUIPMENT DOMESTIC 511,920,130.00      CIJ

    PRODUCT_MODEL Total_SVC_Incidents Total_Repeat_Calls Total_FTF_Calls \
4      MAKE-UP          57.00            13.00          44.00
5      MAKE-UP          57.00            13.00          44.00
6      MAKE-UP          57.00            13.00          44.00
7      MAKE-UP          57.00            13.00          44.00
8      MAKE-UP          57.00            13.00          44.00

    Most_Frequent_Interaction_Type Total_Visits Total_Cases Max_Case_Origin \
4                 Call        53.00      3.03      unknown
5                 Call        53.00      3.03      unknown
6                 Call        53.00      3.03      unknown
7                 Call        53.00      3.03      unknown
8                 Call        53.00      3.03      unknown

    Max_Case_Reason Num_of_Active_Install_Bases Total_Contracts \

```

4	unknown		6.00		6.00	
5	unknown		6.00		6.00	
6	unknown		6.00		6.00	
7	unknown		6.00		6.00	
8	unknown		6.00		6.00	
	Contract_length	Contract_Category	Num_of_Inactive_Install_Bases	\		
4	1,003.00	FSMA		0.00		
5	1,003.00	FSMA		0.00		
6	1,003.00	FSMA		0.00		
7	1,003.00	FSMA		0.00		
8	1,003.00	FSMA		0.00		
	STRATEGIC_ACCOUNTS	TERRITORY_REGION	TRX_YEAR	Recency	Frequency	\
4	0	MC	2017	50	18.86	
5	0	MC	2017	50	18.86	
6	0	MC	2017	50	18.86	
7	0	MC	2017	50	18.86	
8	0	MC	2017	50	18.86	
	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	Avg_Quantity	\	
4	98	233.90	172.39	11.37		
5	98	233.90	172.39	11.37		
6	98	233.90	172.39	11.37		
7	98	233.90	172.39	11.37		
8	98	233.90	172.39	11.37		
	Mode_of_Product_Family	Mode_of_Product_Model	Types_of_Product_Family	\		
4	CIJ	MAKE-UP		1		
5	CIJ	MAKE-UP		1		
6	CIJ	MAKE-UP		1		
7	CIJ	MAKE-UP		1		
8	CIJ	MAKE-UP		1		
	Types_of_Product_Model	Most_Frequent_Sales_Channel	\			
4	3	EDI				
5	3	EDI				
6	3	EDI				
7	3	EDI				
8	3	EDI				
	Most_Frequent_Order_Type	Avg_Price_Index	Tenure	Cluster_Id	\	
4	EDI	1.35	1,879.00	3		
5	EDI	1.35	1,879.00	3		
6	EDI	1.35	1,879.00	3		
7	EDI	1.35	1,879.00	3		
8	EDI	1.35	1,879.00	3		

```

churn_prob_BGNBD Churned_365 Churned_BGNBD
4           0.02        0        0
5           0.02        0        0
6           0.02        0        0
7           0.02        0        0
8           0.02        0        0

```

```
[14]: # calculate CIJ recency
CIJvar = CIJ.groupby('CUSTOMER_SITE_ID').agg({'TRX_DATE': lambda x: (NOW - x.
    .max()).days}).reset_index()
CIJvar['TRX_DATE'] = CIJvar['TRX_DATE'].astype(int)
CIJvar.rename(columns = {'TRX_DATE': 'CIJ_Recency'}, inplace = True)
CIJvar.head(10)
```

```

[14]:   CUSTOMER_SITE_ID  CIJ_Recency
0            90          50
1           111         415
2           114          6
3           141          80
4           158          78
5           247          17
6           287          64
7           352         294
8           367          15
9           380          37

```

```
[15]: CIJvar.shape
```

```
[15]: (6831, 2)
```

```
[16]: # calculate CIJ tenure
CIJvar1 = CIJ.groupby('CUSTOMER_SITE_ID').agg({'TRX_DATE': lambda x: (x.max() -
    x.min()).days + 1}).reset_index()
CIJvar1['TRX_DATE'] = CIJvar1['TRX_DATE'].astype(int)
CIJvar1.rename(columns = {'TRX_DATE': 'CIJ_Tenure'}, inplace = True)
```

```
[17]: CIJvar1.head()
```

```

[17]:   CUSTOMER_SITE_ID  CIJ_Tenure
0            90         1829
1           111         1457
2           114         1968
3           141         1883
4           158         1896

```

```
[18]: CIJvar1.shape
```

[18]: (6831, 2)

[19]: CIJvar2 = pd.merge(CIJvar, CIJvar1, on='CUSTOMER_SITE_ID')

[20]: CIJvar2.head()

[20]:

	CUSTOMER_SITE_ID	CIJ_Recency	CIJ_Tenure
0	90	50	1829
1	111	415	1457
2	114	6	1968
3	141	80	1883
4	158	78	1896

[21]: CIJ_Final = pd.merge(CIJ, CIJvar2, on='CUSTOMER_SITE_ID')

[22]: CIJ_Final.head()

[22]:

	Site_Level_Price_Index	CUSTOMER_ID	CUSTOMER_SITE_ID	SHORT_VERTICAL	\
0	1.40	37	90	CHEMICALS	
1	1.40	37	90	CHEMICALS	
2	1.40	37	90	CHEMICALS	
3	1.40	37	90	CHEMICALS	
4	1.40	37	90	CHEMICALS	

	POSTAL_CODE	CUSTOMER_CLASS	TERRITORY_TYPE	SUPPLIES_SEGMENTATION	\
0	65802	END USER	Industrial	S	
1	65802	END USER	Industrial	S	
2	65802	END USER	Industrial	S	
3	65802	END USER	Industrial	S	
4	65802	END USER	Industrial	S	

	SUPPLIES_DECLINE_REASON	DUNS_NUMBER	TRX_DATE	TRX_AMT_USD	Margin	\
0	None	43937895	2017-01-03	213.80	129.59	
1	None	43937895	2017-09-01	222.40	138.19	
2	None	43937895	2017-08-29	66.72	41.46	
3	None	43937895	2017-08-29	44.48	27.64	
4	None	43937895	2017-05-26	33.36	20.73	

	SALES_CHANNEL	QUANTITY	ORDER_TYPE	ORDER_NUM	PRODUCT_FAMILY	\
0	EDI	20	EDI	511,872,093.00	CIJ	
1	EDI	20	EDI	511,955,407.00	CIJ	
2	Online	6	STANDARD DOMESTIC	511,954,133.00	CIJ	
3	Online	4	STANDARD DOMESTIC	511,954,133.00	CIJ	
4	BigMachine	3	EQUIPMENT DOMESTIC	511,920,130.00	CIJ	

	PRODUCT_MODEL	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\
0	MAKE-UP	57.00	13.00	44.00	

1	MAKE-UP	57.00	13.00	44.00	
2	MAKE-UP	57.00	13.00	44.00	
3	MAKE-UP	57.00	13.00	44.00	
4	MAKE-UP	57.00	13.00	44.00	
0	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	Max_Case-Origin \	
1	Call	53.00	3.03	unknown	
2	Call	53.00	3.03	unknown	
3	Call	53.00	3.03	unknown	
4	Call	53.00	3.03	unknown	
0	Max_Case_Reason	Num_of_Active_Install_Bases	Total_Contracts	\	
1	unknown	6.00	6.00		
2	unknown	6.00	6.00		
3	unknown	6.00	6.00		
4	unknown	6.00	6.00		
0	Contract_length	Contract_Category	Num_of_Inactive_Install_Bases	\	
1	1,003.00	FSMA	0.00		
2	1,003.00	FSMA	0.00		
3	1,003.00	FSMA	0.00		
4	1,003.00	FSMA	0.00		
0	STRATEGIC_ACCOUNTS	TERRITORY_REGION	TRX_YEAR	Recency	Frequency \
1	0	MC	2017	50	18.86
2	0	MC	2017	50	18.86
3	0	MC	2017	50	18.86
4	0	MC	2017	50	18.86
0	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	Avg_Quantity	\
1	98	233.90	172.39	11.37	
2	98	233.90	172.39	11.37	
3	98	233.90	172.39	11.37	
4	98	233.90	172.39	11.37	
0	Mode_of_Product_Family	Mode_of_Product_Model	Types_of_Product_Family	\	
1	CIJ	MAKE-UP	1		
2	CIJ	MAKE-UP	1		
3	CIJ	MAKE-UP	1		
4	CIJ	MAKE-UP	1		

```

Types_of_Product_Model Most_Frequent_Sales_Channel \
0 3 EDI
1 3 EDI
2 3 EDI
3 3 EDI
4 3 EDI

Most_Frequent_Order_Type Avg_Price_Index Tenure Cluster_Id \
0 EDI 1.35 1,879.00 3
1 EDI 1.35 1,879.00 3
2 EDI 1.35 1,879.00 3
3 EDI 1.35 1,879.00 3
4 EDI 1.35 1,879.00 3

churn_prob_BGNBD Churned_365 Churned_BGNBD CIJ_Recency CIJ_Tenure
0 0.02 0 0 50 1829
1 0.02 0 0 50 1829
2 0.02 0 0 50 1829
3 0.02 0 0 50 1829
4 0.02 0 0 50 1829

```

[23]: CIJ_Final[['CUSTOMER_SITE_ID', 'Tenure', 'CIJ_Tenure', 'Churned_365']].head()

[23]: CUSTOMER_SITE_ID Tenure CIJ_Tenure Churned_365

	CUSTOMER_SITE_ID	Tenure	CIJ_Tenure	Churned_365
0	90	1,879.00	1829	0
1	90	1,879.00	1829	0
2	90	1,879.00	1829	0
3	90	1,879.00	1829	0
4	90	1,879.00	1829	0

[24]: # update CIJ tenure

```

CIJ_Final.loc[CIJ_Final['Churned_BGNBD'] == 0, 'New Tenure'] = ↪
    CIJ_Final['CIJ_Recency'] + CIJ_Final['CIJ_Tenure']
CIJ_Final.loc[CIJ_Final['Churned_BGNBD'] == 1, 'New Tenure'] = ↪
    CIJ_Final['CIJ_Tenure']
CIJ_Final.drop('CIJ_Tenure', inplace=True, axis=1)
CIJ_Final.rename(columns = {'New Tenure':'CIJ_Tenure'}, inplace = True)

```

[25]: CIJ_Final[['CUSTOMER_SITE_ID', 'CIJ_Tenure', 'Tenure', 'Churned_BGNBD']].head()

[25]: CUSTOMER_SITE_ID CIJ_Tenure Tenure Churned_BGNBD

	CUSTOMER_SITE_ID	CIJ_Tenure	Tenure	Churned_BGNBD
0	90	1,879.00	1,879.00	0
1	90	1,879.00	1,879.00	0
2	90	1,879.00	1,879.00	0
3	90	1,879.00	1,879.00	0
4	90	1,879.00	1,879.00	0

[31]: CIJ_Final.shape

[31]: (314474, 56)

TTO_Final Creation

[32]: TTO.head()

```
Site_Level_Price_Index  CUSTOMER_ID  CUSTOMER_SITE_ID  SHORT_VERTICAL \
181          0.73        37            114  PHARMA & MEDICAL
183          0.73        37            114  PHARMA & MEDICAL
194          0.73        37            114  PHARMA & MEDICAL
203          0.74        37            114  PHARMA & MEDICAL
296          2.19        37            114  PHARMA & MEDICAL

POSTAL_CODE  CUSTOMER_CLASS  TERRITORY_TYPE  SUPPLIES_SEGMENTATION \
181         92614      END USER    Industrial           M
183         92614      END USER    Industrial           M
194         92614      END USER    Industrial           M
203         92614      END USER    Industrial           M
296         92614      END USER    Industrial           M

SUPPLIES_DECLINE_REASON  DUNS_NUMBER  TRX_DATE  TRX_AMT_USD  Margin \
181             None  84160407  2019-08-07   1,492.40  1,196.86
183             None  84160407  2019-03-01    596.96   478.74
194             None  84160407  2019-01-23    298.48   239.37
203             None  84160407  2020-03-25   2,984.80  2,396.56
296             None  84160407  2015-07-21    235.41   169.06

SALES_CHANNEL  QUANTITY  ORDER_TYPE  ORDER_NUM  PRODUCT_FAMILY \
181          EDI       5          EDI  512,186,579.00        TTO
183          EDI       2          EDI  512,135,829.00        TTO
194          EDI       1          EDI  512,122,913.00        TTO
203          EDI      10          EDI  512,256,620.00        TTO
296  BigMachine       1 EQUIPMENT DOMESTIC  511,687,013.00        TTO

PRODUCT_MODEL  Total_SVC_Incidents  Total_Repeat_Calls  Total_FTF_Calls \
181      RIBBONS          57.00            14.00          43.00
183      RIBBONS          57.00            14.00          43.00
194      RIBBONS          57.00            14.00          43.00
203      RIBBONS          57.00            14.00          43.00
296      RIBBONS          57.00            14.00          43.00

Most_Frequent_Interaction_Type  Total_Visits  Total_Cases \
181              Call        70.00        1.00
183              Call        70.00        1.00
194              Call        70.00        1.00
```

203		Call	70.00	1.00			
296		Call	70.00	1.00			
	Max_Case_Origin	Max_Case_Reason	Num_of_Active_Install_Bases		\		
181	Email - VTI NACC	Customer Experience		15.00			
183	Email - VTI NACC	Customer Experience		15.00			
194	Email - VTI NACC	Customer Experience		15.00			
203	Email - VTI NACC	Customer Experience		15.00			
296	Email - VTI NACC	Customer Experience		15.00			
	Total_Contracts	Contract_length	Contract_Category		\		
181	11.00	521.64	FSMA				
183	11.00	521.64	FSMA				
194	11.00	521.64	FSMA				
203	11.00	521.64	FSMA				
296	11.00	521.64	FSMA				
	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	TERRITORY_REGION		\		
181	0.00	0	NW				
183	0.00	0	NW				
194	0.00	0	NW				
203	0.00	0	NW				
296	0.00	0	NW				
	TRX_YEAR	Recency	Frequency	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	\
181	2019	6	11.38	174	145.43	103.28	
183	2019	6	11.38	174	145.43	103.28	
194	2019	6	11.38	174	145.43	103.28	
203	2020	6	11.38	174	145.43	103.28	
296	2015	6	11.38	174	145.43	103.28	
	Avg_Quantity	Mode_of_Product_Family	Mode_of_Product_Model		\		
181	4.43	CIJ	MAKE-UP				
183	4.43	CIJ	MAKE-UP				
194	4.43	CIJ	MAKE-UP				
203	4.43	CIJ	MAKE-UP				
296	4.43	CIJ	MAKE-UP				
	Types_of_Product_Family	Types_of_Product_Model		\			
181	3	4					
183	3	4					
194	3	4					
203	3	4					
296	3	4					
	Most_Frequent_Sales_Channel	Most_Frequent_Order_Type	Avg_Price_Index	\			
181	EDI	EDI	1.25				

183		EDI		1.25	
194		EDI		1.25	
203		EDI		1.25	
296		EDI		1.25	
	Tenure	Cluster_Id	churn_prob_BGNBD	Churned_365	Churned_BGNBD
181	1,974.00	3	0.00	0	0
183	1,974.00	3	0.00	0	0
194	1,974.00	3	0.00	0	0
203	1,974.00	3	0.00	0	0
296	1,974.00	3	0.00	0	0

```
[33]: # calculate CIJ recency
TTOvar = TTO.groupby('CUSTOMER_SITE_ID').agg({'TRX_DATE': lambda x: (NOW - x.
    .max()).days}).reset_index()
TTOvar['TRX_DATE'] = TTOvar['TRX_DATE'].astype(int)
TTOvar.rename(columns = {'TRX_DATE': 'TTO_Recency'}, inplace = True)
TTOvar.head(10)
```

```
[33]: CUSTOMER_SITE_ID  TTO_Recency
0           114          71
1           946          71
2           948          23
3          1009         409
4          1313         742
5          1337        1590
6          1439          94
7          1529          51
8          1830         617
9          2371         806
```

```
[34]: TTOvar.shape
```

```
[34]: (1220, 2)
```

```
[35]: # calculate CIJ tenure
TTOvar1 = TTO.groupby('CUSTOMER_SITE_ID').agg({'TRX_DATE': lambda x: (x.max() -
    x.min()).days + 1}).reset_index()
TTOvar1['TRX_DATE'] = TTOvar1['TRX_DATE'].astype(int)
TTOvar1.rename(columns = {'TRX_DATE': 'TTO_Tenure'}, inplace = True)
```

```
[36]: TTOvar1.head()
```

```
[36]: CUSTOMER_SITE_ID  TTO_Tenure
0           114         1710
1           946          433
2           948         1925
```

```
3          1009        246
4          1313        737
```

```
[37]: TT0var1.shape
```

```
[37]: (1220, 2)
```

```
[38]: TT0var2 = pd.merge(TT0var,TT0var1,on='CUSTOMER_SITE_ID')
```

```
[39]: TT0var2.head()
```

```
[39]:   CUSTOMER_SITE_ID  TTO_Recency  TTO_Tenure
0            114           71         1710
1            946           71          433
2            948           23         1925
3            1009          409          246
4            1313          742          737
```

```
[40]: TT0_Final = pd.merge(TT0, TT0var2, on='CUSTOMER_SITE_ID')
```

```
[41]: TT0_Final.head()
```

```
[41]:   Site_Level_Price_Index  CUSTOMER_ID  CUSTOMER_SITE_ID      SHORT_VERTICAL \
0                  0.73           37          114  PHARMA & MEDICAL
1                  0.73           37          114  PHARMA & MEDICAL
2                  0.73           37          114  PHARMA & MEDICAL
3                  0.74           37          114  PHARMA & MEDICAL
4                  2.19           37          114  PHARMA & MEDICAL
```

```
  POSTAL_CODE CUSTOMER_CLASS TERRITORY_TYPE SUPPLIES_SEGMENTATION \
0      92614      END USER    Industrial             M
1      92614      END USER    Industrial             M
2      92614      END USER    Industrial             M
3      92614      END USER    Industrial             M
4      92614      END USER    Industrial             M
```

```
  SUPPLIES_DECLINE_REASON DUNS_NUMBER    TRX_DATE  TRX_AMT_USD  Margin \
0            None     84160407  2019-08-07    1,492.40  1,196.86
1            None     84160407  2019-03-01     596.96   478.74
2            None     84160407  2019-01-23     298.48   239.37
3            None     84160407  2020-03-25    2,984.80  2,396.56
4            None     84160407  2015-07-21     235.41   169.06
```

```
  SALES_CHANNEL QUANTITY      ORDER_TYPE  ORDER_NUM PRODUCT_FAMILY \
0       EDI          5           EDI  512,186,579.00        TTO
1       EDI          2           EDI  512,135,829.00        TTO
2       EDI          1           EDI  512,122,913.00        TTO
```

3	EDI	10	EDI	512,256,620.00	TT0	
4	BigMachine	1	EQUIPMENT DOMESTIC	511,687,013.00	TT0	
0	RIBBONS	57.00	14.00	43.00		
1	RIBBONS	57.00	14.00	43.00		
2	RIBBONS	57.00	14.00	43.00		
3	RIBBONS	57.00	14.00	43.00		
4	RIBBONS	57.00	14.00	43.00		
0	Call	70.00	1.00	Email - VTI NACC		
1	Call	70.00	1.00	Email - VTI NACC		
2	Call	70.00	1.00	Email - VTI NACC		
3	Call	70.00	1.00	Email - VTI NACC		
4	Call	70.00	1.00	Email - VTI NACC		
0	Customer Experience	15.00	11.00			
1	Customer Experience	15.00	11.00			
2	Customer Experience	15.00	11.00			
3	Customer Experience	15.00	11.00			
4	Customer Experience	15.00	11.00			
0	521.64	FSMA	0.00			
1	521.64	FSMA	0.00			
2	521.64	FSMA	0.00			
3	521.64	FSMA	0.00			
4	521.64	FSMA	0.00			
0	STRATEGIC_ACCOUNTS	TERRITORY_REGION	TRX_YEAR	Recency	Frequency	\
1	0	NW	2019	6	11.38	
2	0	NW	2019	6	11.38	
3	0	NW	2019	6	11.38	
4	0	NW	2020	6	11.38	
0	0	NW	2015	6	11.38	
0	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	Avg_Quantity	\	
1	174	145.43	103.28	4.43		
2	174	145.43	103.28	4.43		
3	174	145.43	103.28	4.43		
4	174	145.43	103.28	4.43		
0	Mode_of_Product_Family	Mode_of_Product_Model	Types_of_Product_Family	\		
	CIJ	MAKE-UP	3			

```

1             CIJ          MAKE-UP      3
2             CIJ          MAKE-UP      3
3             CIJ          MAKE-UP      3
4             CIJ          MAKE-UP      3

    Types_of_Product_Model Most_Frequent_Sales_Channel \
0                  4           EDI
1                  4           EDI
2                  4           EDI
3                  4           EDI
4                  4           EDI

    Most_Frequent_Order_Type Avg_Price_Index   Tenure Cluster_Id \
0                 EDI        1.25 1,974.00      3
1                 EDI        1.25 1,974.00      3
2                 EDI        1.25 1,974.00      3
3                 EDI        1.25 1,974.00      3
4                 EDI        1.25 1,974.00      3

    churn_prob_BGNBD Churned_365 Churned_BGNBD TTO_Recency TTO_Tenure
0         0.00            0            0          71        1710
1         0.00            0            0          71        1710
2         0.00            0            0          71        1710
3         0.00            0            0          71        1710
4         0.00            0            0          71        1710

```

```
[42]: TTO_Final[['CUSTOMER_SITE_ID', 'Tenure', 'TTO_Tenure', 'Churned_BGNBD']].head()
```

```
[42]: CUSTOMER_SITE_ID  Tenure  TTO_Tenure  Churned_BGNBD
0           114 1,974.00      1710          0
1           114 1,974.00      1710          0
2           114 1,974.00      1710          0
3           114 1,974.00      1710          0
4           114 1,974.00      1710          0
```

```
[43]: # update TTO tenure
TTO_Final.loc[TTO_Final['Churned_BGNBD'] == 0, 'New Tenure'] = ↴
    ↪TTO_Final['TTO_Recency'] + TTO_Final['TTO_Tenure']
TTO_Final.loc[TTO_Final['Churned_BGNBD'] == 1, 'New Tenure'] = ↴
    ↪TTO_Final['TTO_Tenure']
TTO_Final.drop('TTO_Tenure', inplace=True, axis=1)
TTO_Final.rename(columns = {'New Tenure': 'TTO_Tenure'}, inplace = True)
```

```
[44]: TTO_Final[['CUSTOMER_SITE_ID', 'TTO_Tenure', 'Tenure', 'Churned_BGNBD']].head()
```

```
[44]: CUSTOMER_SITE_ID  TTO_Tenure  Tenure  Churned_BGNBD
0           114 1,781.00 1,974.00          0
```

```

1           114    1,781.00 1,974.00      0
2           114    1,781.00 1,974.00      0
3           114    1,781.00 1,974.00      0
4           114    1,781.00 1,974.00      0

```

[45]: TTO_Final = TTO_Final.reset_index()

[46]: TTO_Final[['CUSTOMER_SITE_ID', 'TTO_Tenure', 'Tenure', 'Churned_BGNBD']].head()

```

[46]:   CUSTOMER_SITE_ID  TTO_Tenure    Tenure  Churned_BGNBD
0           114    1,781.00 1,974.00      0
1           114    1,781.00 1,974.00      0
2           114    1,781.00 1,974.00      0
3           114    1,781.00 1,974.00      0
4           114    1,781.00 1,974.00      0

```

[47]: TTO_Final.shape

[47]: (21202, 56)

LCM_Final Creation

[48]: LCM.head()

```

[48]:   Site_Level_Price_Index  CUSTOMER_ID  CUSTOMER_SITE_ID  SHORT_VERTICAL \
1           0.79          6482            24      GRAPHICS
2           0.79          6482            24      GRAPHICS
3           0.79          6482            24      GRAPHICS
877          0.86          125             367     BEVERAGE
878          0.86          125             367     BEVERAGE

  POSTAL_CODE  CUSTOMER_CLASS  TERRITORY_TYPE  SUPPLIES_SEGMENTATION \
1        60085      END USER      Industrial           S
2        60085      END USER      Industrial           S
3        60085      END USER      Industrial           S
877      12302      END USER      Industrial          XL
878      12302      END USER      Industrial          XL

  SUPPLIES_DECLINE_REASON  DUNS_NUMBER  TRX_DATE  TRX_AMT_USD  Margin \
1           None       144782380 2020-04-15      552.68  537.02
2           None       144782380 2020-05-04     5,000.00 4,843.37
3           None       144782380 2020-05-11     5,000.00 4,843.37
877  Over Stocked / Timing      37366788 2015-05-22      782.12  754.50
878  Over Stocked / Timing      37366788 2015-02-23      782.12  754.50

  SALES_CHANNEL  QUANTITY      ORDER_TYPE  ORDER_NUM PRODUCT_FAMILY \
1      SFDC_CPQ        2 EQUIPMENT DOMESTIC  512,260,527.00           LCM

```

2	Online	20	STANDARD DOMESTIC	512,267,173.00	LCM		
3	Copy	20	STANDARD DOMESTIC	512,268,803.00	LCM		
877	Online	4	STANDARD DOMESTIC	511,570,308.00	LCM		
878	Online	4	STANDARD DOMESTIC	511,570,308.00	LCM		
1	PRODUCT_MODEL	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\		
2	INK	13.00	7.00	6.00			
3	INK	13.00	7.00	6.00			
877	INK	154.00	26.00	128.00			
878	INK	154.00	26.00	128.00			
1	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	\			
2	Call	81.00	1.97				
3	Call	81.00	1.97				
877	Call	123.00	1.00				
878	Call	123.00	1.00				
1	Max_Case-Origin	Max_Case_Reason	Num_of_Active_Install_Bases	\			
2	unknown	unknown	5.00				
3	unknown	unknown	5.00				
877	TS Survey Followup	CX: Tech Support	34.00				
878	TS Survey Followup	CX: Tech Support	34.00				
1	Total_Contracts	Contract_length	Contract_Category	\			
2	0.00	0.00	No Contract				
3	0.00	0.00	No Contract				
877	42.00	1,266.79	FSMA				
878	42.00	1,266.79	FSMA				
1	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	TERRITORY_REGION	\			
2	0.00	0	MW				
3	0.00	0	MW				
877	10.00	0	NE				
878	10.00	0	NE				
1	TRX_YEAR	Recency	Frequency	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	\
2	2020	24	17.67	4	3,102.11	2,901.24	
3	2020	24	17.67	4	3,102.11	2,901.24	
877	2015	15	8.97	218	873.03	673.77	
878	2015	15	8.97	218	873.03	673.77	

	Avg_Quantity	Mode_of_Product_Family	Mode_of_Product_Model	\
1	12.00	LCM	INK	
2	12.00	LCM	INK	
3	12.00	LCM	INK	
877	37.66	CIJ	MAKE-UP	
878	37.66	CIJ	MAKE-UP	

	Types_of_Product_Family	Types_of_Product_Model	\
1	2	2	
2	2	2	
3	2	2	
877	3	5	
878	3	5	

	Most_Frequent_Sales_Channel	Most_Frequent_Order_Type	Avg_Price_Index	\
1	Copy	STANDARD DOMESTIC	0.79	
2	Copy	STANDARD DOMESTIC	0.79	
3	Copy	STANDARD DOMESTIC	0.79	
877	Online	STANDARD DOMESTIC	1.00	
878	Online	STANDARD DOMESTIC	1.00	

	Tenure	Cluster_Id	churn_prob_BGNBD	Churned_365	Churned_BGNBD
1	77.00	3	0.05	0	0
2	77.00	3	0.05	0	0
3	77.00	3	0.05	0	0
877	1,961.00	3	0.00	0	0
878	1,961.00	3	0.00	0	0

```
[49]: # calculate CIJ recency
LCMvar = LCM.groupby('CUSTOMER_SITE_ID').agg({'TRX_DATE': lambda x: (NOW - x.
    .max()).days}).reset_index()
LCMvar['TRX_DATE'] = LCMvar['TRX_DATE'].astype(int)
LCMvar.rename(columns = {'TRX_DATE': 'LCM_Recency'}, inplace = True)
LCMvar.head(10)
```

```
[49]: CUSTOMER_SITE_ID  LCM_Recency
0                  24        24
1                 367      1749
2                 425       147
3                 426      1168
4                 456      1536
5                 481       854
6                 682        15
7                 905        37
8                 946      238
9                 980      902
```

```
[50]: LCMvar.shape
```

```
[50]: (1486, 2)
```

```
[51]: # calculate LCM tenure
LCMvar1 = LCM.groupby('CUSTOMER_SITE_ID').agg({'TRX_DATE': lambda x: (x.max() - x.min()).days + 1}).reset_index()
LCMvar1['TRX_DATE'] = LCMvar1['TRX_DATE'].astype(int)
LCMvar1.rename(columns = {'TRX_DATE': 'LCM_Tenure'}, inplace = True)
```

```
[52]: LCMvar1.head()
```

```
[52]:   CUSTOMER_SITE_ID  LCM_Tenure
0                 24        27
1                367       212
2                425      1278
3                426         1
4                456       405
```

```
[53]: LCMvar1.shape
```

```
[53]: (1486, 2)
```

```
[54]: LCMvar2 = pd.merge(LCMvar, LCMvar1, on='CUSTOMER_SITE_ID')
```

```
[55]: LCMvar2.head()
```

```
[55]:   CUSTOMER_SITE_ID  LCM_Recency  LCM_Tenure
0                 24          24        27
1                367        1749       212
2                425         147      1278
3                426        1168         1
4                456        1536       405
```

```
[56]: LCM_Final = pd.merge(LCM, LCMvar2, on='CUSTOMER_SITE_ID')
```

```
[57]: LCM_Final.head()
```

```
[57]:   Site_Level_Price_Index  CUSTOMER_ID  CUSTOMER_SITE_ID  SHORT_VERTICAL \
0                  0.79        6482             24    GRAPHICS
1                  0.79        6482             24    GRAPHICS
2                  0.79        6482             24    GRAPHICS
3                  0.86        125              367  BEVERAGE
4                  0.86        125              367  BEVERAGE

  POSTAL_CODE  CUSTOMER_CLASS  TERRITORY_TYPE  SUPPLIES_SEGMENTATION \
0        60085        END USER        Industrial           S
```

1	60085	END USER	Industrial	S
2	60085	END USER	Industrial	S
3	12302	END USER	Industrial	XL
4	12302	END USER	Industrial	XL
SUPPLIES_DECLINE_REASONS DUNS_NUMBER TRX_DATE TRX_AMT_USD Margin \				
0	None	144782380	2020-04-15	552.68 537.02
1	None	144782380	2020-05-04	5,000.00 4,843.37
2	None	144782380	2020-05-11	5,000.00 4,843.37
3	Over Stocked / Timing	37366788	2015-05-22	782.12 754.50
4	Over Stocked / Timing	37366788	2015-02-23	782.12 754.50
SALES_CHANNEL QUANTITY ORDER_TYPE ORDER_NUM PRODUCT_FAMILY \				
0	SFDC_CPQ	2	EQUIPMENT DOMESTIC	512,260,527.00 LCM
1	Online	20	STANDARD DOMESTIC	512,267,173.00 LCM
2	Copy	20	STANDARD DOMESTIC	512,268,803.00 LCM
3	Online	4	STANDARD DOMESTIC	511,570,308.00 LCM
4	Online	4	STANDARD DOMESTIC	511,570,308.00 LCM
PRODUCT_MODEL Total_SVC_Incidents Total_Repeat_Calls Total_FTF_Calls \				
0	INK	13.00	7.00	6.00
1	INK	13.00	7.00	6.00
2	INK	13.00	7.00	6.00
3	INK	154.00	26.00	128.00
4	INK	154.00	26.00	128.00
Most_Frequent_Interaction_Type Total_Visits Total_Cases \				
0	Call	81.00	1.97	
1	Call	81.00	1.97	
2	Call	81.00	1.97	
3	Call	123.00	1.00	
4	Call	123.00	1.00	
Max_Case_Origin Max_Case_Reason Num_of_Active_Install_Bases \				
0	unknown	unknown	5.00	
1	unknown	unknown	5.00	
2	unknown	unknown	5.00	
3	TS Survey Followup	CX: Tech Support	34.00	
4	TS Survey Followup	CX: Tech Support	34.00	
Total_Contracts Contract_length Contract_Category \				
0	0.00	0.00	No Contract	
1	0.00	0.00	No Contract	
2	0.00	0.00	No Contract	
3	42.00	1,266.79	FSMA	
4	42.00	1,266.79	FSMA	

	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	TERRITORY_REGION	\			
0	0.00	0	MW				
1	0.00	0	MW				
2	0.00	0	MW				
3	10.00	0	NE				
4	10.00	0	NE				
	TRX_YEAR	Recency	Frequency	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	\
0	2020	24	17.67	4	3,102.11	2,901.24	
1	2020	24	17.67	4	3,102.11	2,901.24	
2	2020	24	17.67	4	3,102.11	2,901.24	
3	2015	15	8.97	218	873.03	673.77	
4	2015	15	8.97	218	873.03	673.77	
	Avg_Quantity	Mode_of_Product_Family	Mode_of_Product_Model	\			
0	12.00	LCM	INK				
1	12.00	LCM	INK				
2	12.00	LCM	INK				
3	37.66	CIJ	MAKE-UP				
4	37.66	CIJ	MAKE-UP				
	Types_of_Product_Family	Types_of_Product_Model	\				
0	2	2					
1	2	2					
2	2	2					
3	3	5					
4	3	5					
	Most_Frequent_Sales_Channel	Most_Frequent_Order_Type	Avg_Price_Index	\			
0	Copy	STANDARD DOMESTIC	0.79				
1	Copy	STANDARD DOMESTIC	0.79				
2	Copy	STANDARD DOMESTIC	0.79				
3	Online	STANDARD DOMESTIC	1.00				
4	Online	STANDARD DOMESTIC	1.00				
	Tenure	Cluster_Id	churn_prob_BGNBD	Churned_365	Churned_BGNBD	\	
0	77.00	3	0.05	0	0		
1	77.00	3	0.05	0	0		
2	77.00	3	0.05	0	0		
3	1,961.00	3	0.00	0	0		
4	1,961.00	3	0.00	0	0		
	LCM_Recency	LCM_Tenure					
0	24	27					
1	24	27					
2	24	27					
3	1749	212					

```
4          1749         212
```

```
[58]: LCM_Final[['CUSTOMER_SITE_ID', 'Tenure', 'LCM_Tenure', 'Churned_BGNBD']].head()
```

```
[58]:   CUSTOMER_SITE_ID  Tenure  LCM_Tenure  Churned_BGNBD
 0            24    77.00      27            0
 1            24    77.00      27            0
 2            24    77.00      27            0
 3           367  1,961.00     212            0
 4           367  1,961.00     212            0
```

```
[59]: # update CIJ tenure
LCM_Final.loc[LCM_Final['Churned_BGNBD'] == 0, 'New Tenure'] = ↴
    LCM_Final['LCM_Recency'] + LCM_Final['LCM_Tenure']
LCM_Final.loc[LCM_Final['Churned_BGNBD'] == 1, 'New Tenure'] = ↴
    LCM_Final['LCM_Tenure']
LCM_Final.drop('LCM_Tenure', inplace=True, axis=1)
LCM_Final.rename(columns = {'New Tenure':'LCM_Tenure'}, inplace = True)
```

```
[60]: LCM_Final[['CUSTOMER_SITE_ID', 'LCM_Tenure', 'Tenure', 'Churned_BGNBD']].head()
```

```
[60]:   CUSTOMER_SITE_ID  LCM_Tenure  Tenure  Churned_BGNBD
 0            24    51.00    77.00            0
 1            24    51.00    77.00            0
 2            24    51.00    77.00            0
 3           367  1,961.00  1,961.00            0
 4           367  1,961.00  1,961.00            0
```

```
[61]: LCM_Final = LCM_Final.reset_index()
```

```
[62]: LCM_Final[['CUSTOMER_SITE_ID', 'LCM_Tenure', 'Tenure', 'Churned_BGNBD']].head()
```

```
[62]:   CUSTOMER_SITE_ID  LCM_Tenure  Tenure  Churned_BGNBD
 0            24    51.00    77.00            0
 1            24    51.00    77.00            0
 2            24    51.00    77.00            0
 3           367  1,961.00  1,961.00            0
 4           367  1,961.00  1,961.00            0
```

```
[63]: LCM_Final.shape
```

```
[63]: (26654, 56)
```

TIJ_Final Creation

```
[64]: TIJ.head()
```

[64] :	Site_Level_Price_Index	CUSTOMER_ID	CUSTOMER_SITE_ID	SHORT_VERTICAL	\
212	0.80	37	114	PHARMA & MEDICAL	
213	0.80	37	114	PHARMA & MEDICAL	
297	0.75	37	126	PHARMA & MEDICAL	
298	0.75	37	126	PHARMA & MEDICAL	
299	0.75	37	126	PHARMA & MEDICAL	
	POSTAL_CODE	CUSTOMER_CLASS	TERRITORY_TYPE	SUPPLIES_SEGMENTATION	\
212	92614	END USER	Industrial	M	
213	92614	END USER	Industrial	M	
297	8822	END USER	Industrial	S	
298	8822	END USER	Industrial	S	
299	8822	END USER	Industrial	S	
	SUPPLIES_DECLINE_REASON	DUNS_NUMBER	TRX_DATE	TRX_AMT_USD	Margin \
212	None	84160407	2017-09-07	246.57	146.87
213	None	84160407	2017-04-12	209.58	109.88
297	Over Stocked / Timing	36781508	2017-03-29	823.20	575.52
298	Over Stocked / Timing	36781508	2017-08-17	1,029.00	719.40
299	Over Stocked / Timing	36781508	2017-06-23	926.10	647.46
	SALES_CHANNEL	QUANTITY	ORDER_TYPE	ORDER_NUM	PRODUCT_FAMILY \
212	EDI	3	EDI	511,957,157.00	TIJ
213	BigMachine	3	EQUIPMENT DOMESTIC	511,904,741.00	TIJ
297	EDI	8	EDI	511,900,892.00	TIJ
298	Online	10	STANDARD DOMESTIC	511,945,707.00	TIJ
299	Online	9	STANDARD DOMESTIC	511,930,964.00	TIJ
	PRODUCT_MODEL	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\
212	INK	57.00	14.00	43.00	
213	INK	57.00	14.00	43.00	
297	INK	1.00	0.00	1.00	
298	INK	1.00	0.00	1.00	
299	INK	1.00	0.00	1.00	
	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases		\
212	Call	70.00	1.00		
213	Call	70.00	1.00		
297	Call	13.00	1.00		
298	Call	13.00	1.00		
299	Call	13.00	1.00		
	Max_Case_Origin	Max_Case_Reason	Num_of_Active_Install_Bases		\
212	Email - VTI NACC	Customer Experience	15.00		
213	Email - VTI NACC	Customer Experience	15.00		
297	Phone	Customer Experience	2.00		
298	Phone	Customer Experience	2.00		

299	Phone	Customer Experience	2.00				
	Total_Contracts	Contract_length	Contract_Category	\			
212	11.00	521.64	FSMA				
213	11.00	521.64	FSMA				
297	0.00	0.00	No Contract				
298	0.00	0.00	No Contract				
299	0.00	0.00	No Contract				
	Num_of_Inactive_Install_Bases	STRATEGIC_ACCOUNTS	TERRITORY_REGION	\			
212	0.00	0	NW				
213	0.00	0	NW				
297	0.00	0	NE				
298	0.00	0	NE				
299	0.00	0	NE				
	TRX_YEAR	Recency	Frequency	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	\
212	2017	6	11.38	174	145.43	103.28	
213	2017	6	11.38	174	145.43	103.28	
297	2017	49	61.94	19	835.78	605.25	
298	2017	49	61.94	19	835.78	605.25	
299	2017	49	61.94	19	835.78	605.25	
	Avg_Quantity	Mode_of_Product_Family	Mode_of_Product_Model	\			
212	4.43	CIJ	MAKE-UP				
213	4.43	CIJ	MAKE-UP				
297	8.84	TIJ	INK				
298	8.84	TIJ	INK				
299	8.84	TIJ	INK				
	Types_of_Product_Family	Types_of_Product_Model	\				
212	3	4					
213	3	4					
297	1	1					
298	1	1					
299	1	1					
	Most_Frequent_Sales_Channel	Most_Frequent_Order_Type	Avg_Price_Index	\			
212	EDI	EDI	1.25				
213	EDI	EDI	1.25				
297	EDI	EDI	0.75				
298	EDI	EDI	0.75				
299	EDI	EDI	0.75				
	Tenure	Cluster_Id	churn_prob_BGNBD	Churned_365	Churned_BGNBD		
212	1,974.00	3	0.00	0	0		
213	1,974.00	3	0.00	0	0		

```

297 1,164.00      2          0.01      0          0
298 1,164.00      2          0.01      0          0
299 1,164.00      2          0.01      0          0

```

```
[65]: # calculate TIJ recency
TIJvar = TIJ.groupby('CUSTOMER_SITE_ID').agg({'TRX_DATE': lambda x: (NOW - x.
    .max()).days}).reset_index()
TIJvar['TRX_DATE'] = TIJvar['TRX_DATE'].astype(int)
TIJvar.rename(columns = {'TRX_DATE': 'TIJ_Recency'}, inplace = True)
TIJvar.head(10)
```

```
[65]:   CUSTOMER_SITE_ID  TIJ_Recency
0           114        1001
1           126         49
2           335       1952
3           352         195
4           425         237
5           426         295
6           427         465
7           441         16
8           487        440
9           673        870
```

```
[66]: TIJvar.shape
```

```
[66]: (509, 2)
```

```
[67]: # calculate TIJ tenure
TIJvar1 = TIJ.groupby('CUSTOMER_SITE_ID').agg({'TRX_DATE': lambda x: (x.max() -
    x.min()).days + 1}).reset_index()
TIJvar1['TRX_DATE'] = TIJvar1['TRX_DATE'].astype(int)
TIJvar1.rename(columns = {'TRX_DATE': 'TIJ_Tenure'}, inplace = True)
```

```
[68]: TIJvar1.head()
```

```
[68]:   CUSTOMER_SITE_ID  TIJ_Tenure
0           114         149
1           126       1115
2           335         29
3           352       920
4           425      1541
```

```
[69]: TIJvar1.shape
```

```
[69]: (509, 2)
```

```
[70]: TIJvar2 = pd.merge(TIJvar, TIJvar1, on='CUSTOMER_SITE_ID')
```

```
[71]: TIJvar2.head()
```

	CUSTOMER_SITE_ID	TIJ_Recency	TIJ_Tenure
0	114	1001	149
1	126	49	1115
2	335	1952	29
3	352	195	920
4	425	237	1541

```
[72]: TIJ_Final = pd.merge(TIJ, TIJvar2, on='CUSTOMER_SITE_ID')
```

```
[73]: TIJ_Final.shape
```

```
[73]: (6027, 55)
```

```
[74]: TIJ_Final.head()
```

	Site_Level_Price_Index	CUSTOMER_ID	CUSTOMER_SITE_ID	SHORT_VERTICAL	\
0	0.80	37	114	PHARMA & MEDICAL	
1	0.80	37	114	PHARMA & MEDICAL	
2	0.75	37	126	PHARMA & MEDICAL	
3	0.75	37	126	PHARMA & MEDICAL	
4	0.75	37	126	PHARMA & MEDICAL	

	POSTAL_CODE	CUSTOMER_CLASS	TERRITORY_TYPE	SUPPLIES_SEGMENTATION	\
0	92614	END USER	Industrial	M	
1	92614	END USER	Industrial	M	
2	8822	END USER	Industrial	S	
3	8822	END USER	Industrial	S	
4	8822	END USER	Industrial	S	

	SUPPLIES_DECLINE_REASON	DUNS_NUMBER	TRX_DATE	TRX_AMT_USD	Margin	\
0	None	84160407	2017-09-07	246.57	146.87	
1	None	84160407	2017-04-12	209.58	109.88	
2	Over Stocked / Timing	36781508	2017-03-29	823.20	575.52	
3	Over Stocked / Timing	36781508	2017-08-17	1,029.00	719.40	
4	Over Stocked / Timing	36781508	2017-06-23	926.10	647.46	

	SALES_CHANNEL	QUANTITY	ORDER_TYPE	ORDER_NUM	PRODUCT_FAMILY	\
0	EDI	3	EDI	511,957,157.00	TIJ	
1	BigMachine	3	EQUIPMENT DOMESTIC	511,904,741.00	TIJ	
2	EDI	8	EDI	511,900,892.00	TIJ	
3	Online	10	STANDARD DOMESTIC	511,945,707.00	TIJ	
4	Online	9	STANDARD DOMESTIC	511,930,964.00	TIJ	

	PRODUCT_MODEL	Total_SVC_Incidents	Total_Repeat_Calls	Total_FTF_Calls	\
0	INK	57.00	14.00	43.00	

1	INK	57.00	14.00	43.00	
2	INK	1.00	0.00	1.00	
3	INK	1.00	0.00	1.00	
4	INK	1.00	0.00	1.00	
	Most_Frequent_Interaction_Type	Total_Visits	Total_Cases	Max_Case-Origin \	
0	Call	70.00	1.00	Email - VTI NACC	
1	Call	70.00	1.00	Email - VTI NACC	
2	Call	13.00	1.00	Phone	
3	Call	13.00	1.00	Phone	
4	Call	13.00	1.00	Phone	
	Max_Case_Reason	Num_of_Active_Install_Bases	Total_Contracts	\	
0	Customer Experience	15.00	11.00		
1	Customer Experience	15.00	11.00		
2	Customer Experience	2.00	0.00		
3	Customer Experience	2.00	0.00		
4	Customer Experience	2.00	0.00		
	Contract_length	Contract_Category	Num_of_Inactive_Install_Bases	\	
0	521.64	FSMA	0.00		
1	521.64	FSMA	0.00		
2	0.00	No Contract	0.00		
3	0.00	No Contract	0.00		
4	0.00	No Contract	0.00		
	STRATEGIC_ACCOUNTS	TERRITORY_REGION	TRX_YEAR	Recency	Frequency \
0	0	NW	2017	6	11.38
1	0	NW	2017	6	11.38
2	0	NE	2017	49	61.94
3	0	NE	2017	49	61.94
4	0	NE	2017	49	61.94
	Num_of_Trxns	Avg_Trxn_Amt	Avg_Margin	Avg_Quantity	\
0	174	145.43	103.28	4.43	
1	174	145.43	103.28	4.43	
2	19	835.78	605.25	8.84	
3	19	835.78	605.25	8.84	
4	19	835.78	605.25	8.84	
	Mode_of_Product_Family	Mode_of_Product_Model	Types_of_Product_Family		\
0	CIJ	MAKE-UP		3	
1	CIJ	MAKE-UP		3	
2	TIJ	INK		1	
3	TIJ	INK		1	
4	TIJ	INK		1	

```

Types_of_Product_Model Most_Frequent_Sales_Channel \
0 4 EDI
1 4 EDI
2 1 EDI
3 1 EDI
4 1 EDI

Most_Frequent_Order_Type Avg_Price_Index Tenure Cluster_Id \
0 EDI 1.25 1,974.00 3
1 EDI 1.25 1,974.00 3
2 EDI 0.75 1,164.00 2
3 EDI 0.75 1,164.00 2
4 EDI 0.75 1,164.00 2

churn_prob_BGNBD Churned_365 Churned_BGNBD TIJ_Recency TIJ_Tenure
0 0.00 0 0 1001 149
1 0.00 0 0 1001 149
2 0.01 0 0 49 1115
3 0.01 0 0 49 1115
4 0.01 0 0 49 1115

```

```
[75]: TIJ_Final[['CUSTOMER_SITE_ID', 'Tenure', 'TIJ_Tenure', 'Churned_BGNBD', 'TIJ_Recency']]  
      ↪head()
```

```
[75]: CUSTOMER_SITE_ID Tenure TIJ_Tenure Churned_BGNBD TIJ_Recency
0 114 1,974.00 149 0 1001
1 114 1,974.00 149 0 1001
2 126 1,164.00 1115 0 49
3 126 1,164.00 1115 0 49
4 126 1,164.00 1115 0 49
```

```
[76]: # update TIJ tenure
TIJ_Final.loc[TIJ_Final['Churned_BGNBD'] == 0, 'New Tenure'] =  
      ↪TIJ_Final['TIJ_Recency'] + TIJ_Final['TIJ_Tenure']
TIJ_Final.loc[TIJ_Final['Churned_BGNBD'] == 1, 'New Tenure'] =  
      ↪TIJ_Final['TIJ_Tenure']
TIJ_Final.drop('TIJ_Tenure', inplace=True, axis=1)
TIJ_Final.rename(columns = {'New Tenure':'TIJ_Tenure'}, inplace = True)
```

```
[77]: TIJ_Final[['CUSTOMER_SITE_ID', 'TIJ_Tenure', 'Tenure', 'Churned_BGNBD']].head()
```

```
[77]: CUSTOMER_SITE_ID TIJ_Tenure Tenure Churned_BGNBD
0 114 1,150.00 1,974.00 0
1 114 1,150.00 1,974.00 0
2 126 1,164.00 1,164.00 0
3 126 1,164.00 1,164.00 0
4 126 1,164.00 1,164.00 0
```

```
[78]: TIJ_Final = TIJ_Final.reset_index()

[79]: TIJ_Final[['CUSTOMER_SITE_ID', 'TIJ_Tenure', 'Tenure', 'Churned_BGNBD']].head()

[79]:
   CUSTOMER_SITE_ID  TIJ_Tenure    Tenure  Churned_BGNBD
0                 114  1,150.00  1,974.00          0
1                 114  1,150.00  1,974.00          0
2                 126  1,164.00  1,164.00          0
3                 126  1,164.00  1,164.00          0
4                 126  1,164.00  1,164.00          0

[80]: TIJ_Final.shape

[80]: (6027, 56)
```

1 SHORT VERTICAL

1.1 CIJ

```
[82]: CIJ_Final_SV = CIJ_Final[CIJ_Final['SHORT_VERTICAL'] != 'UNKNOWN']

[86]: kmf1 = KaplanMeierFitter()

plt.figure(figsize = (25,20))
duration = CIJ_Final_SV['CIJ_Tenure']
observed = CIJ_Final_SV['Churned_BGNBD']

# Set the order that the positions will be plotted
positions_1 = ['CHEMICALS', 'GRAPHICS', 'PHARMA & MEDICAL', 'OTHER',
              'ELECTRICAL / ELECTRONICS', 'EXTRUSION / WIRE & CABLE',
              'OEM-INDUSTRIAL EQUIPMENT', 'BEVERAGE', 'DAIRY & EGGS',
              'COSMETICS / PERSONAL CARE']
positions_2 = ['BAKED GOODS & CEREALS', 'PET FOOD & ANIMAL FEED',
              'MEAT & POULTRY', 'FRUIT & VEGETABLE', 'INDUSTRIAL EQUIPMENT',
              'BUILDING MATERIALS', 'FROZEN PREPARED MEALS',
              'PACKAGING MATERIALS', 'TOBACCO', 'POSTAL']
positions_3 = ['AERO/AUTO', 'OTHER FOOD', 'CANDY & CONFECTION', 'TEXTILE',
              'SALTY SNACKS', 'FISH & SEAFOOD',
              'DISTRIBUTOR']

ax1 = plt.subplot(331)
ax2 = plt.subplot(332, sharey = ax1)
ax3 = plt.subplot(333, sharey = ax1)
ax1.title.set_text('1')
ax2.title.set_text('2')
ax3.title.set_text('3')
```

```

for pos in positions_1:

    idx = CIJ_Final_SV['SHORT_VERTICAL'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax1)
    ax1.grid(linestyle='dotted')
    ax1.legend()

for pos in positions_2:

    idx = CIJ_Final_SV['SHORT_VERTICAL'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax2)
    ax2.grid(linestyle='dotted')
    ax2.legend()

for pos in positions_3:

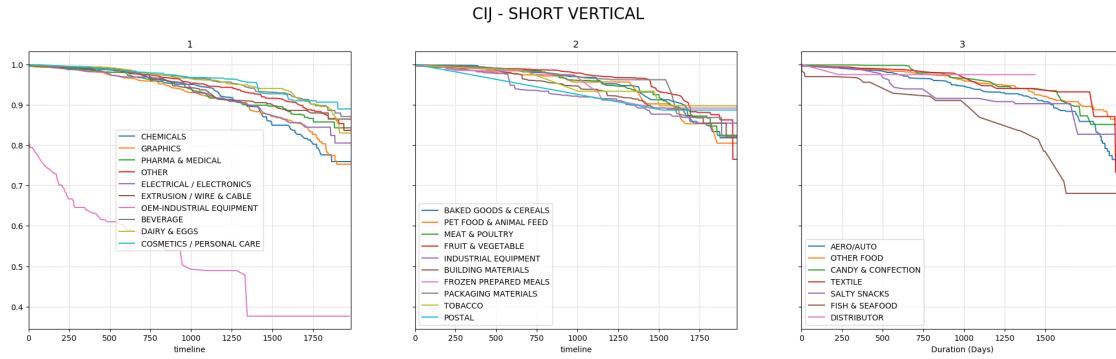
    idx = CIJ_Final_SV['SHORT_VERTICAL'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax3)
    ax3.grid(linestyle='dotted')
    ax3.legend()

plt.suptitle('CIJ - SHORT VERTICAL', fontsize = 20)
plt.subplots_adjust(top=0.94)
plt.xticks(np.arange(0,1750,250))
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.style.use('default')
plt.savefig('CIJ SHORT VERTICAL.png',dpi = 100)
plt.show()

```



1.2 TTO

```
[87]: TTO_Final_SV = TTO_Final[TTO_Final['SHORT_VERTICAL'] != 'UNKNOWN']
```

```
[92]: kmf1 = KaplanMeierFitter()
```

```
plt.figure(figsize = (25,20))
duration = TTO_Final_SV['TTO_Tenure']
observed = TTO_Final_SV['Churned_BGNBD']

# Set the order that the positions will be plotted
positions_1 = ['PHARMA & MEDICAL', 'GRAPHICS', 'OTHER FOOD',
    'MEAT & POULTRY', 'AERO/AUTO', 'FRUIT & VEGETABLE',
    'COSMETICS / PERSONAL CARE', 'OEM-INDUSTRIAL EQUIPMENT',
    'DAIRY & EGGS', 'CANDY & CONFECTION']

positions_2 = ['BAKED GOODS & CEREALS',
    'FROZEN PREPARED MEALS', 'CHEMICALS', 'BUILDING MATERIALS',
    'INDUSTRIAL EQUIPMENT', 'OTHER', 'ELECTRICAL / ELECTRONICS',
    'PET FOOD & ANIMAL FEED', 'FISH & SEAFOOD']

position_3 = ['EXTRUSION / WIRE & CABLE', 'TEXTILE', 'TOBACCO', 'BEVERAGE',
    'PACKAGING MATERIALS', 'DISTRIBUTOR']

ax1 = plt.subplot(331)
ax2 = plt.subplot(332, sharey = ax1)
ax3 = plt.subplot(333, sharey = ax1)
ax1.title.set_text('SHORT VERTICAL 1')
ax2.title.set_text('SHORT VERTICAL 2')
ax3.title.set_text('SHORT VERTICAL 3')

for pos in positions_1:
```

```

idx = TTO_Final_SV['SHORT_VERTICAL'] == pos

kmf1.fit(duration[idx], observed[idx], label = pos)

kmf1.survival_function_.plot(ax=ax1)
ax1.grid(linestyle='dotted')
ax1.legend()

for pos in positions_2:

    idx = TTO_Final_SV['SHORT_VERTICAL'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax2)
    ax2.grid(linestyle='dotted')
    ax2.legend()

for pos in positions_3:

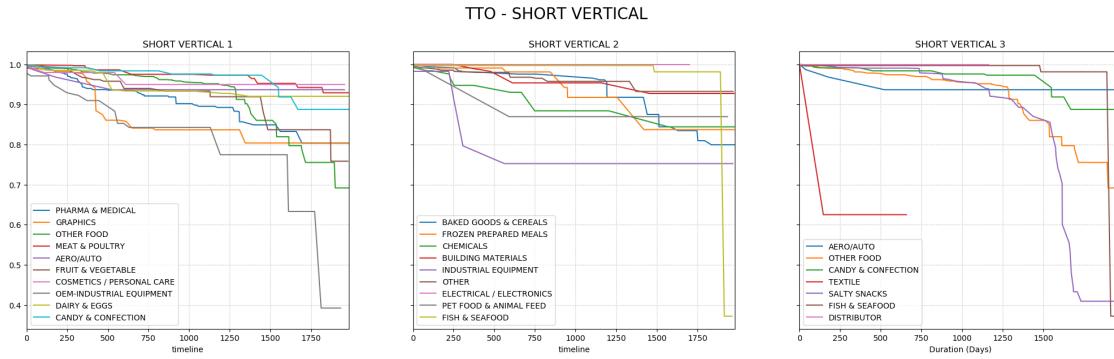
    idx = TTO_Final_SV['SHORT_VERTICAL'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax3)
    ax3.grid(linestyle='dotted')
    ax3.legend()

plt.suptitle('TTO - SHORT VERTICAL', fontsize = 20)
plt.subplots_adjust(top=0.94)
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.style.use('default')
plt.xticks(np.arange(0,1750,250))
plt.savefig('TTO SHORT VERTICAL.png', dpi=100)
plt.show()

```



1.3 LCM

```
[93]: LCM_Final_SV = LCM_Final[LCM_Final['SHORT_VERTICAL'] != 'UNKNOWN']
```

```
[95]: kmf1 = KaplanMeierFitter()
```

```
plt.figure(figsize = (25,20))
duration = LCM_Final_SV['LCM_Tenure']
observed = LCM_Final_SV['Churned_BGNBD']

# Set the order that the positions will be plotted
positions_1 = ['GRAPHICS', 'BEVERAGE', 'OTHER', 'OTHER FOOD', 'PHARMA & MEDICAL',
              'CANDY & CONFECTION', 'BAKED GOODS & CEREALS', 'DAIRY & EGGS',
              'MEAT & POULTRY', 'FRUIT & VEGETABLE', 'AERO/AUTO',
              'EXTRUSION / WIRE & CABLE']

positions_2 = ['COSMETICS / PERSONAL CARE',
              'FROZEN PREPARED MEALS', 'ELECTRICAL / ELECTRONICS',
              'BUILDING MATERIALS', 'PET FOOD & ANIMAL FEED', 'CHEMICALS',
              'INDUSTRIAL EQUIPMENT', 'SALTY SNACKS']

positions_3 = ['OEM-INDUSTRIAL EQUIPMENT',
              'POSTAL', 'PACKAGING MATERIALS', 'TEXTILE', 'TOBACCO',
              'FISH & SEAFOOD', 'DISTRIBUTOR']

ax1 = plt.subplot(331)
ax2 = plt.subplot(332, sharey = ax1)
ax3 = plt.subplot(333, sharey = ax1)
ax1.title.set_text('SHORT VERTICAL 1')
ax2.title.set_text('SHORT VERTICAL 2')
ax3.title.set_text('SHORT VERTICAL 3')
```

```

for pos in positions_1:

    idx = LCM_Final_SV['SHORT_VERTICAL'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax1)
    ax1.legend()
    ax1.grid(linestyle='dotted')


for pos in positions_2:

    idx = LCM_Final_SV['SHORT_VERTICAL'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax2)
    ax2.legend()
    ax2.grid(linestyle='dotted')


for pos in positions_3:

    idx = LCM_Final_SV['SHORT_VERTICAL'] == pos

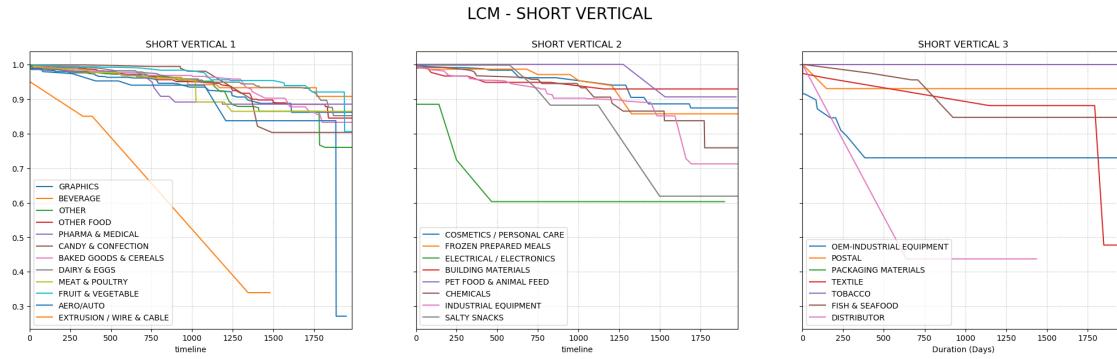
    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax3)
    ax3.legend()
    ax3.grid(linestyle='dotted')


plt.suptitle('LCM - SHORT VERTICAL', fontsize = 20)
plt.subplots_adjust(top=0.94)
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')

plt.style.use('default')
plt.savefig('LCM SHORT VERTICAL.png', dpi=100)
plt.show()

```



1.4 TIJ

```
[96]: TIJ_Final_SV = TIJ_Final[TIJ_Final['SHORT_VERTICAL'] != 'UNKNOWN']
```

```
[97]: kmf1 = KaplanMeierFitter()
```

```
plt.figure(figsize = (25,20))
duration = TIJ_Final_SV['TIJ_Tenure']
observed = TIJ_Final_SV['Churned_BGNBD']

# Set the order that the positions will be plotted
positions_1 = ['PHARMA & MEDICAL', 'GRAPHICS', 'OEM-INDUSTRIAL EQUIPMENT',
    'COSMETICS / PERSONAL CARE', 'FROZEN PREPARED MEALS', 'OTHER',
    'ELECTRICAL / ELECTRONICS', 'OTHER FOOD', 'DAIRY & EGGS']

positions_2 = ['BEVERAGE', 'INDUSTRIAL EQUIPMENT', 'TOBACCO', 'CHEMICALS',
    'TEXTILE', 'MEAT & POULTRY', 'BAKED GOODS & CEREALS',
    'BUILDING MATERIALS', 'CANDY & CONFECTION']

positions_3 = [ 'AERO/AUTO',
    'FRUIT & VEGETABLE', 'PET FOOD & ANIMAL FEED', 'FISH & SEAFOOD',
    'PACKAGING MATERIALS']

ax1 = plt.subplot(331)
ax2 = plt.subplot(332, sharey = ax1)
ax3 = plt.subplot(333, sharey = ax1)
ax1.title.set_text('SHORT VERTICAL 1')
ax2.title.set_text('SHORT VERTICAL 2')
ax3.title.set_text('SHORT VERTICAL 3')

for pos in positions_1:
```

```

idx = TIJ_Final_SV['SHORT_VERTICAL'] == pos

kmf1.fit(duration[idx], observed[idx], label = pos)

kmf1.survival_function_.plot(ax=ax1)
ax1.legend()
ax1.grid(linestyle='dotted')


for pos in positions_2:

    idx = TIJ_Final_SV['SHORT_VERTICAL'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax2)
    ax2.legend()
    ax2.grid(linestyle='dotted')


for pos in positions_3:

    idx = TIJ_Final_SV['SHORT_VERTICAL'] == pos

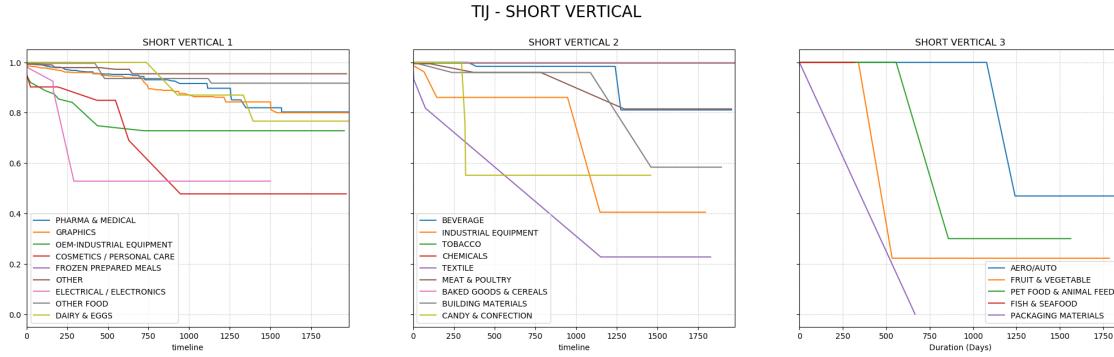
    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax3)
    ax3.legend()
    ax3.grid(linestyle='dotted')


plt.suptitle('TIJ - SHORT VERTICAL', fontsize = 20)
plt.subplots_adjust(top=0.94)
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')

plt.style.use('default')
plt.savefig('TIJ SHORT VERTICAL.png', dpi=100)
plt.show()

```



2 SUPPLIES_SEGMENTATION

2.1 CIJ

```
[98]: CIJ_Final_SS = CIJ_Final[CIJ_Final['SUPPLIES_SEGMENTATION'] != 'Unclass']
```

```
[99]: kmf1 = KaplanMeierFitter()
plt.figure(figsize = (8,6))

duration = CIJ_Final_SS['CIJ_Tenure']
observed = CIJ_Final_SS['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = CIJ_Final_SS['SUPPLIES_SEGMENTATION'].unique()

ax = plt.subplot()

for pos in positions:

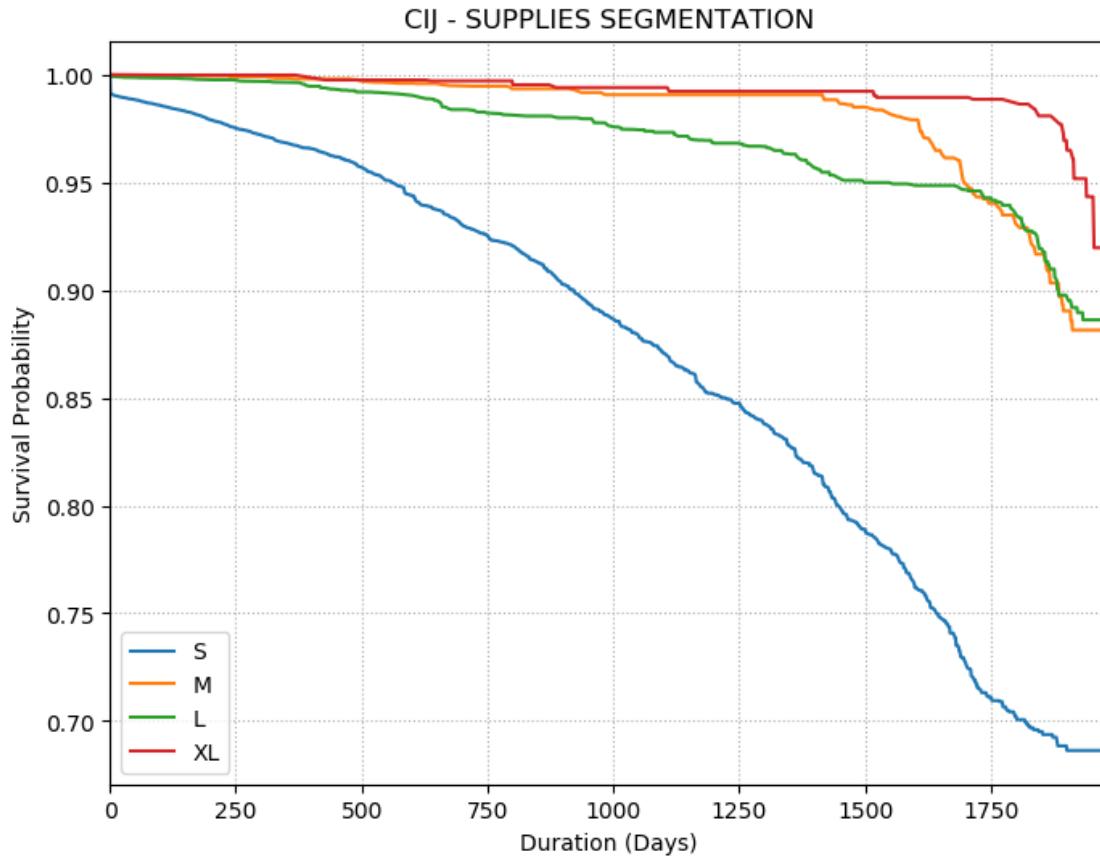
    idx = CIJ_Final_SS['SUPPLIES_SEGMENTATION'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('CIJ - SUPPLIES SEGMENTATION')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
```

```
plt.show()
```



2.2 TTO

```
[100]: TTO_Final_SS = TTO_Final[TTO_Final['SUPPLIES_SEGMENTATION']!='Unclass']
```

```
[101]: kmf1 = KaplanMeierFitter()
plt.figure(figsize = (8,6))
duration = TTO_Final_SS['TTO_Tenure']
observed = TTO_Final_SS['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TTO_Final_SS['SUPPLIES_SEGMENTATION'].unique()

ax = plt.subplot()

for pos in positions:
```

```

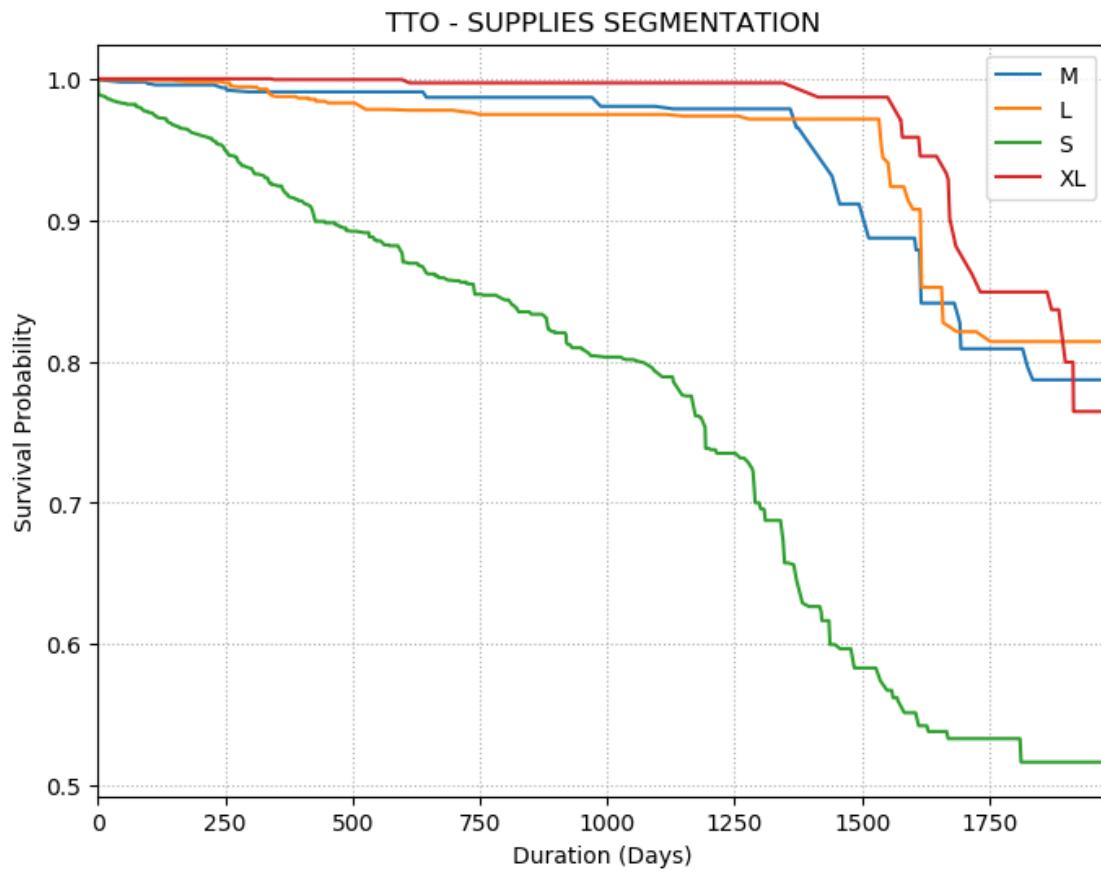
idx = TTO_Final_SS['SUPPLIES_SEGMENTATION'] == pos

kmf1.fit(duration[idx], observed[idx], label = pos)

kmf1.survival_function_.plot(ax=ax)
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('TTO - SUPPLIES SEGMENTATION')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()

```



2.3 LCM

```
[102]: LCM_Final_SS = LCM_Final[LCM_Final['SUPPLIES_SEGMENTATION']!='Unclass']
```

```
[103]: kmf1 = KaplanMeierFitter()
plt.figure(figsize = (8,6))

duration = LCM_Final_SS['LCM_Tenure']
observed = LCM_Final_SS['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = LCM_Final_SS['SUPPLIES_SEGMENTATION'].unique()
ax = plt.subplot()

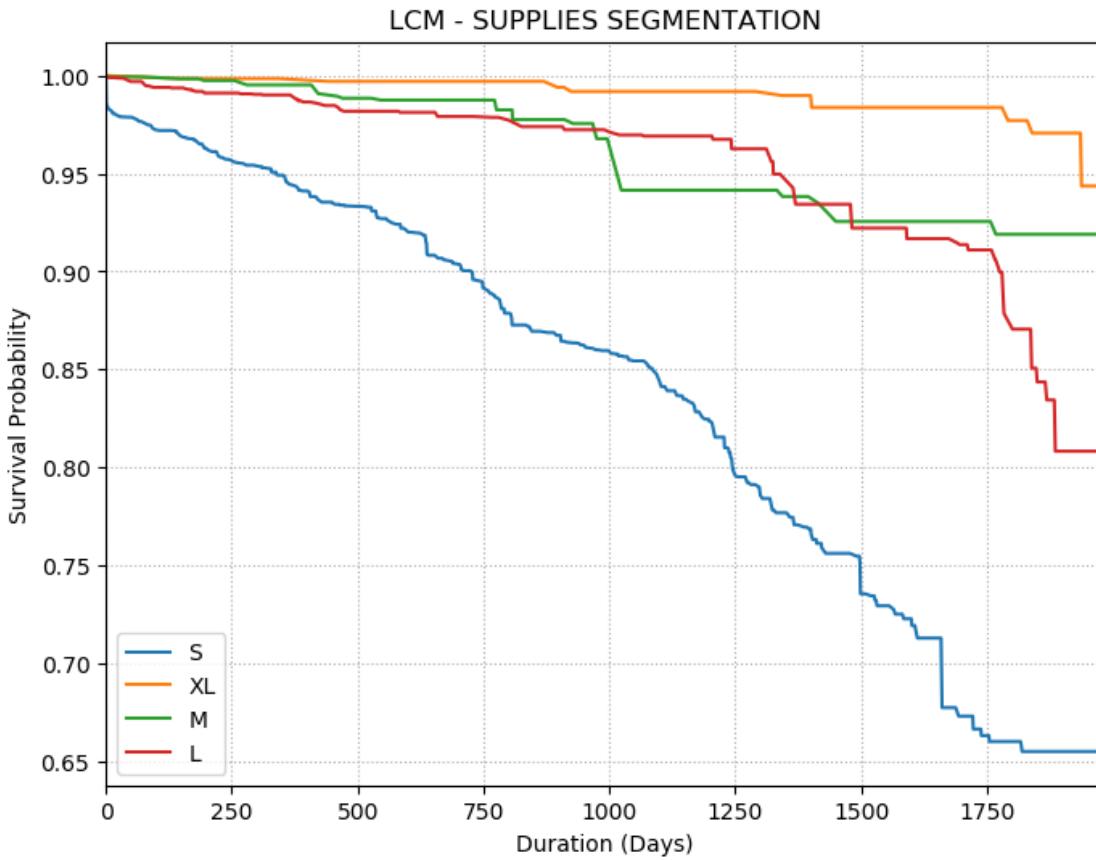
for pos in positions:

    idx = LCM_Final_SS['SUPPLIES_SEGMENTATION'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('LCM - SUPPLIES SEGMENTATION')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()
```



2.4 TIJ

```
[104]: TIJ_Final_SS = TIJ_Final[TIJ_Final['SUPPLIES_SEGMENTATION']!='Unclass']
```

```
[105]: kmf1 = KaplanMeierFitter()
plt.figure(figsize = (8,6))

duration = TIJ_Final_SS['TIJ_Tenure']
observed = TIJ_Final_SS['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TIJ_Final_SS['SUPPLIES_SEGMENTATION'].unique()
ax = plt.subplot()

for pos in positions:

    idx = TIJ_Final_SS['SUPPLIES_SEGMENTATION'] == pos
```

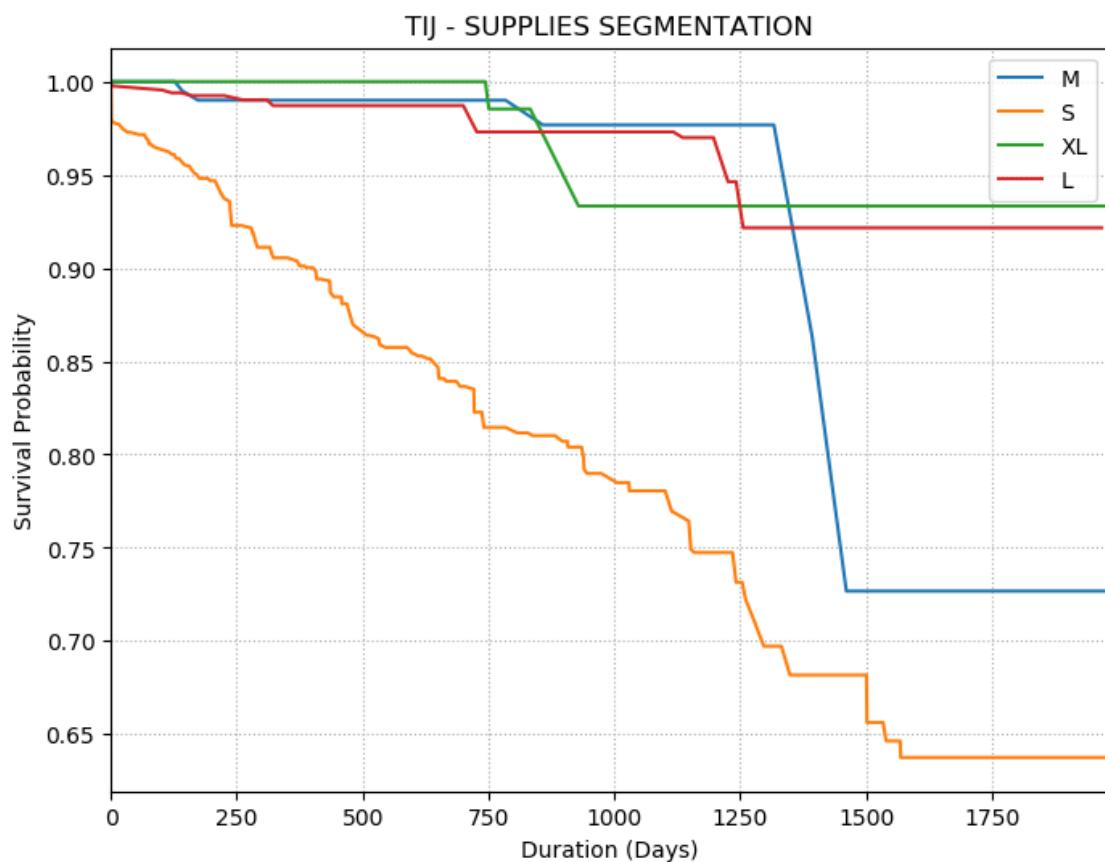
```

kmf1.fit(duration[idx], observed[idx], label = pos)

kmf1.survival_function_.plot(ax=ax)
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('TIJ - SUPPLIES SEGMENTATION')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()

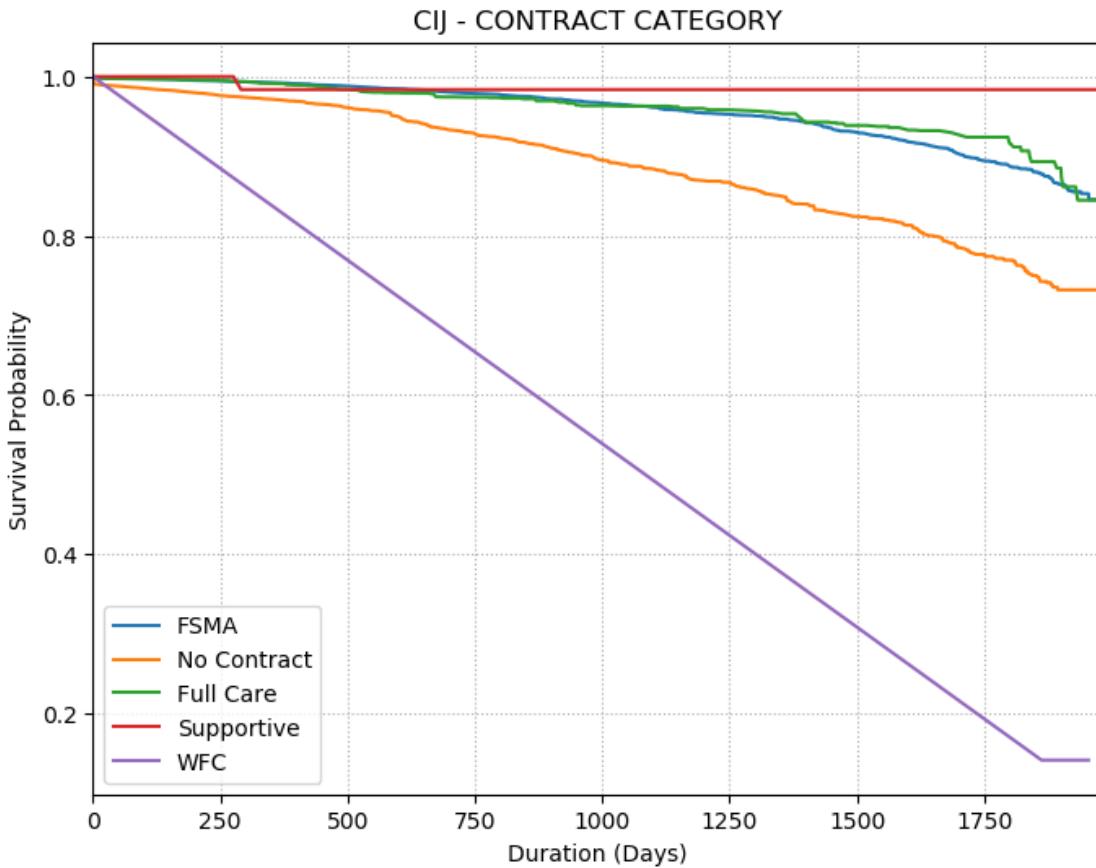
```



3 Contract Category

3.1 CIJ

```
[106]: kmf1 = KaplanMeierFitter()  
plt.figure(figsize = (8,6))  
  
duration = CIJ_Final['CIJ_Tenure']  
observed = CIJ_Final['Churned_BGNBD']  
  
# Set the order that the positions will be plotted  
positions = CIJ_Final['Contract_Category'].unique()  
  
ax = plt.subplot()  
  
  
for pos in positions:  
  
    idx = CIJ_Final['Contract_Category'] == pos  
  
    kmf1.fit(duration[idx], observed[idx], label = pos)  
  
    kmf1.survival_function_.plot(ax=ax)  
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))  
  
ax.legend()  
plt.title('CIJ - CONTRACT CATEGORY')  
plt.xlabel('Duration (Days)')  
plt.ylabel('Survival Probability')  
plt.grid(linestyle='dotted')  
plt.style.use('default')  
plt.show()
```



3.2 TTO

```
[107]: kmf1 = KaplanMeierFitter()
plt.figure(figsize = (8,6))

duration = TTO_Final['TTO_Tenure']
observed = TTO_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TTO_Final['Contract_Category'].unique()

ax = plt.subplot()

for pos in positions:

    idx = TTO_Final['Contract_Category'] == pos

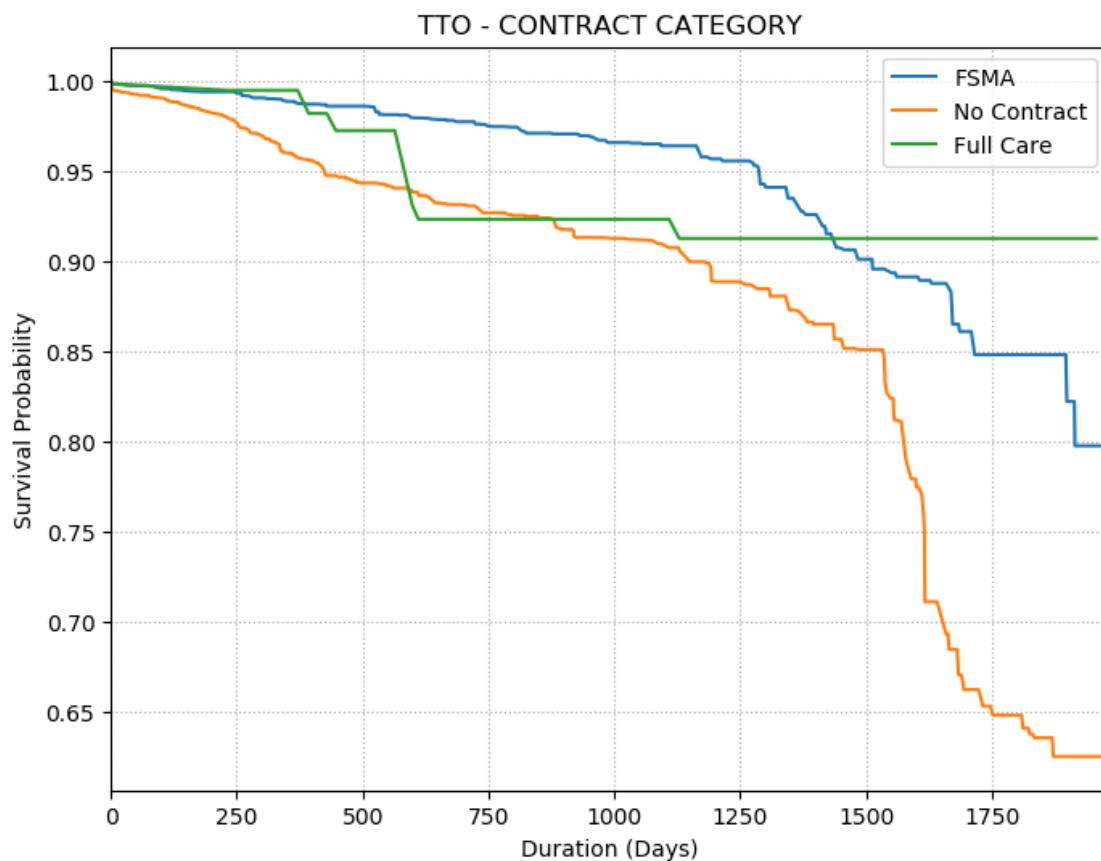
    kmf1.fit(duration[idx], observed[idx], label = pos)
```

```

kmf1.survival_function_.plot(ax=ax)
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('TTO - CONTRACT CATEGORY')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()

```



3.3 LCM

```

[108]: kmf1 = KaplanMeierFitter()
plt.figure(figsize = (8,6))

duration = LCM_Final['LCM_Tenure']
observed = LCM_Final['Churned_BGNBD']

```

```

# Set the order that the positions will be plotted
positions = LCM_Final['Contract_Category'].unique()

ax = plt.subplot()

for pos in positions:

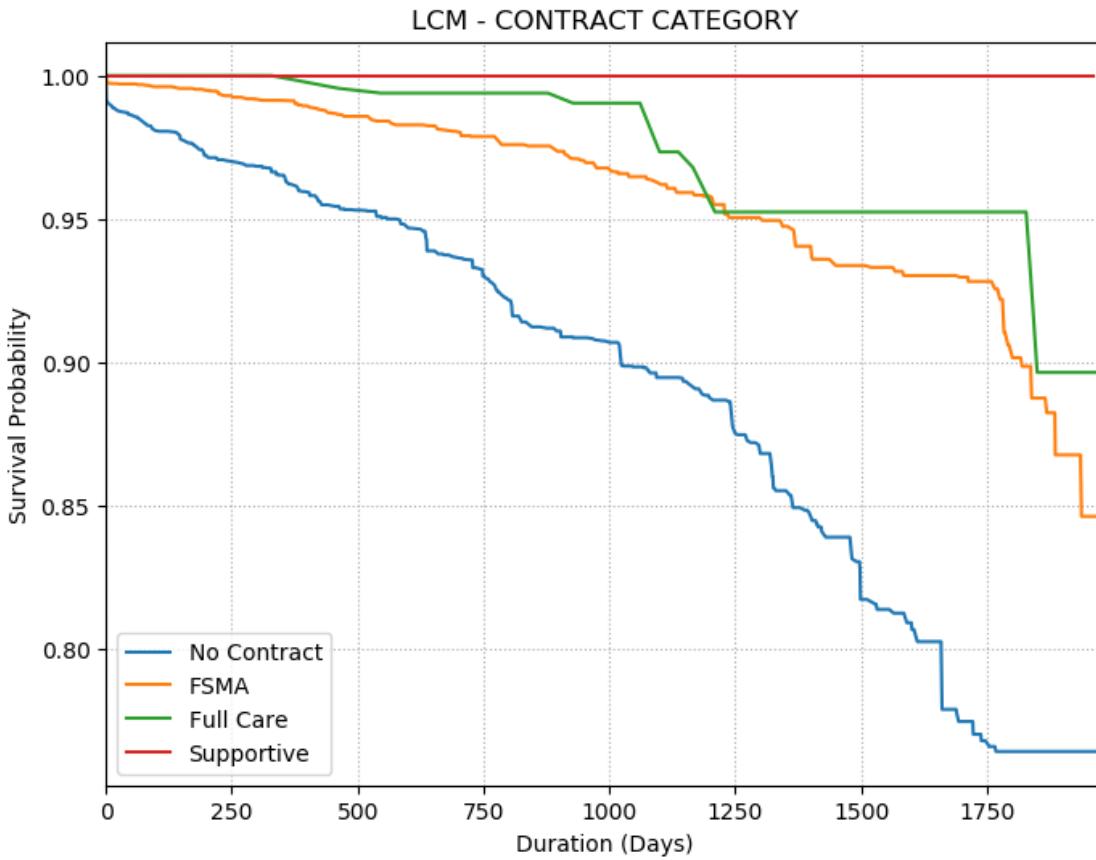
    idx = LCM_Final['Contract_Category'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('LCM - CONTRACT CATEGORY')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()

```



3.4 TIJ

```
[109]: kmf1 = KaplanMeierFitter()
plt.figure(figsize = (8,6))

duration = TIJ_Final['TIJ_Tenure']
observed = TIJ_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TIJ_Final['Contract_Category'].unique()

ax = plt.subplot()

for pos in positions:

    idx = TIJ_Final['Contract_Category'] == pos

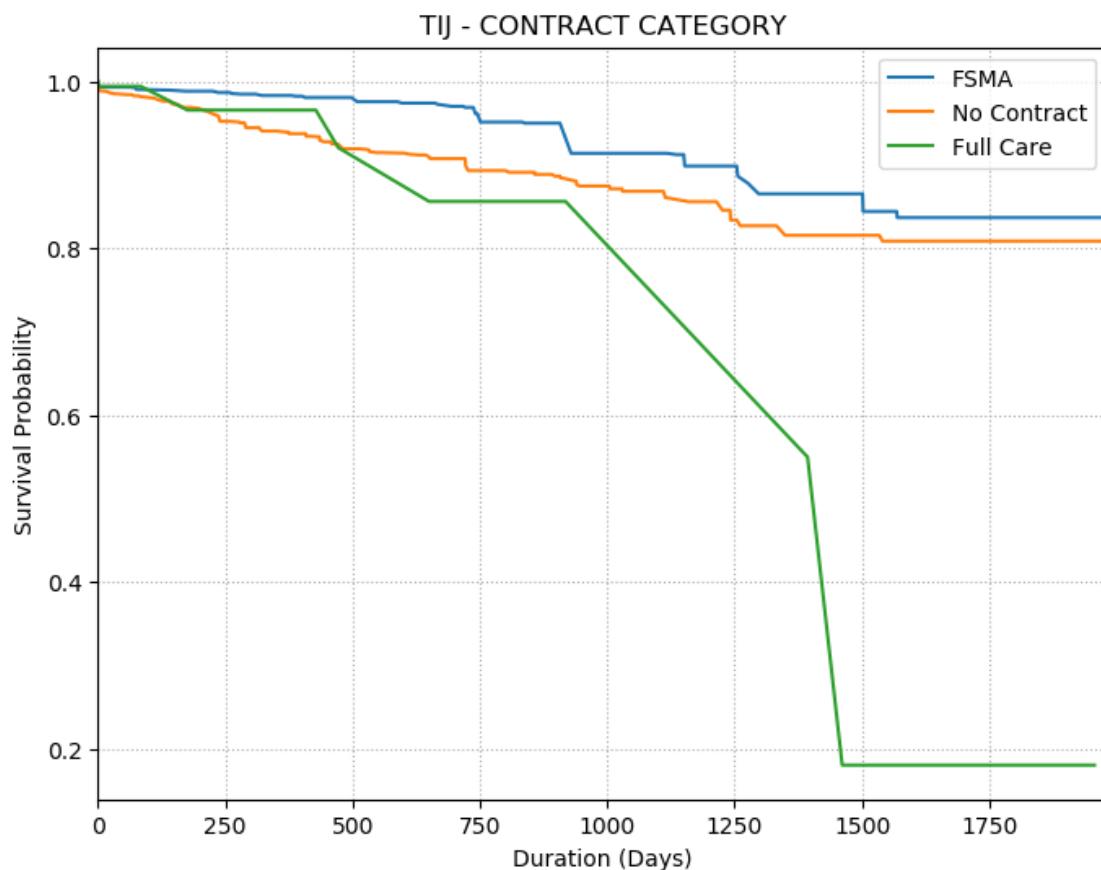
    kmf1.fit(duration[idx], observed[idx], label = pos)
```

```

kmf1.survival_function_.plot(ax=ax)
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('TIJ - CONTRACT CATEGORY')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()

```



4 Strategic Account

4.1 CIJ

```
[110]: kmf1 = KaplanMeierFitter()

duration = CIJ_Final['CIJ_Tenure']
observed = CIJ_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = CIJ_Final['STRATEGIC_ACCOUNTS'].unique()
ax = plt.subplot()

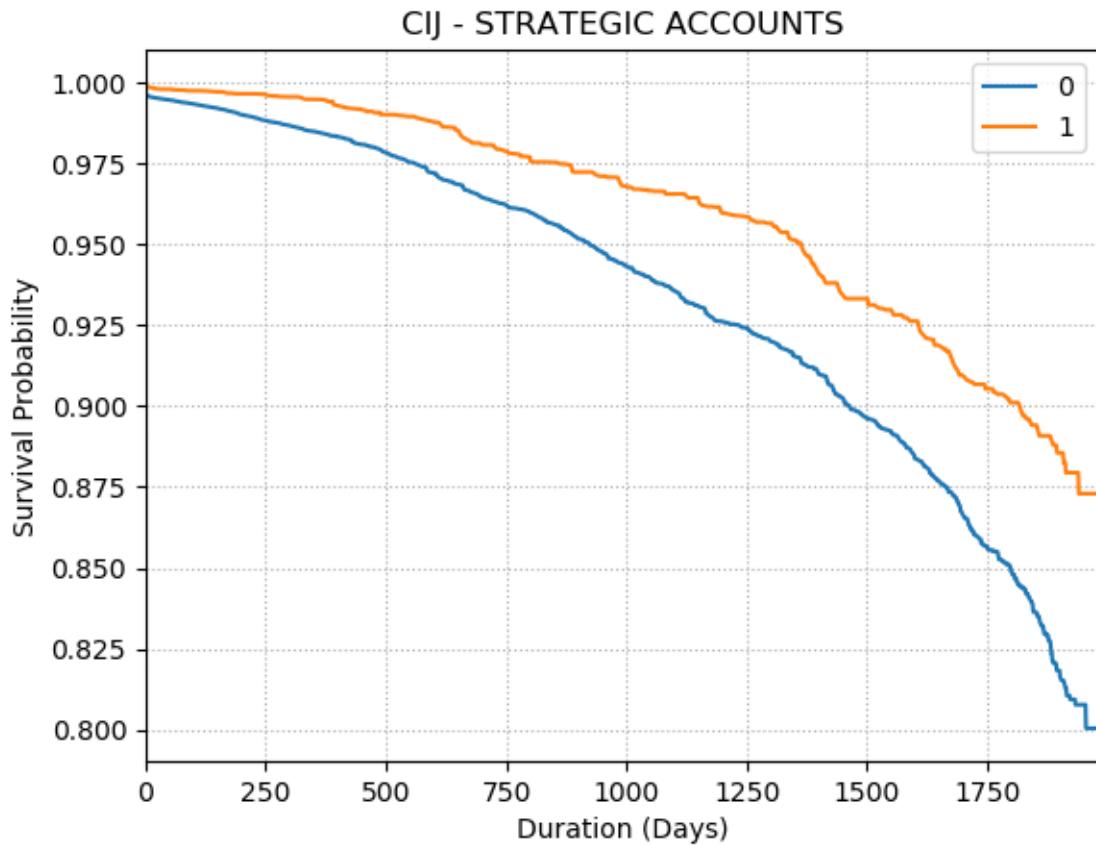
for pos in positions:

    idx = CIJ_Final['STRATEGIC_ACCOUNTS'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('CIJ - STRATEGIC ACCOUNTS')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()
```



4.2 TTO

```
[111]: kmf1 = KaplanMeierFitter()

duration = TTO_Final['TTO_Tenure']
observed = TTO_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TTO_Final['STRATEGIC_ACCOUNTS'].unique()
ax = plt.subplot()

for pos in positions:

    idx = TTO_Final['STRATEGIC_ACCOUNTS'] == pos

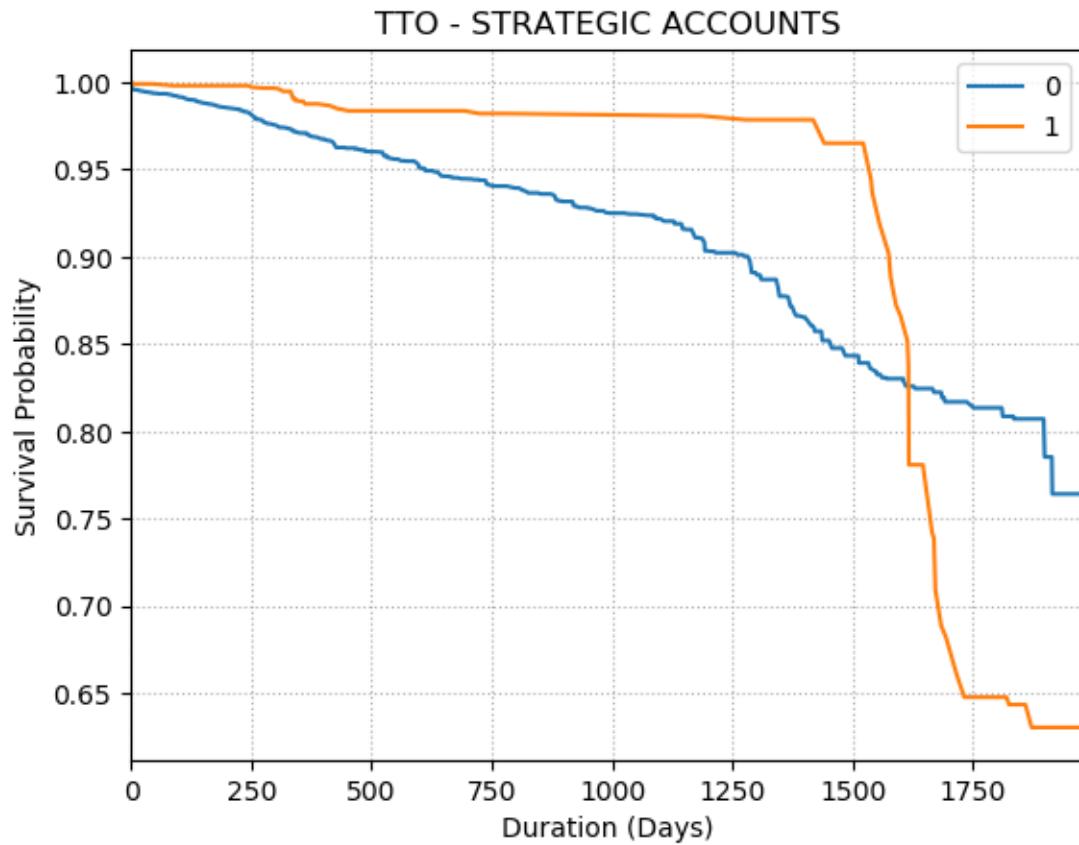
    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
```

```

ax.legend()
plt.title('TTO - STRATEGIC ACCOUNTS')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()

```



4.3 LCM

```

[112]: kmf1 = KaplanMeierFitter()

duration = LCM_Final['LCM_Tenure']
observed = LCM_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = LCM_Final['STRATEGIC_ACCOUNTS'].unique()
ax = plt.subplot()

```

```

for pos in positions:

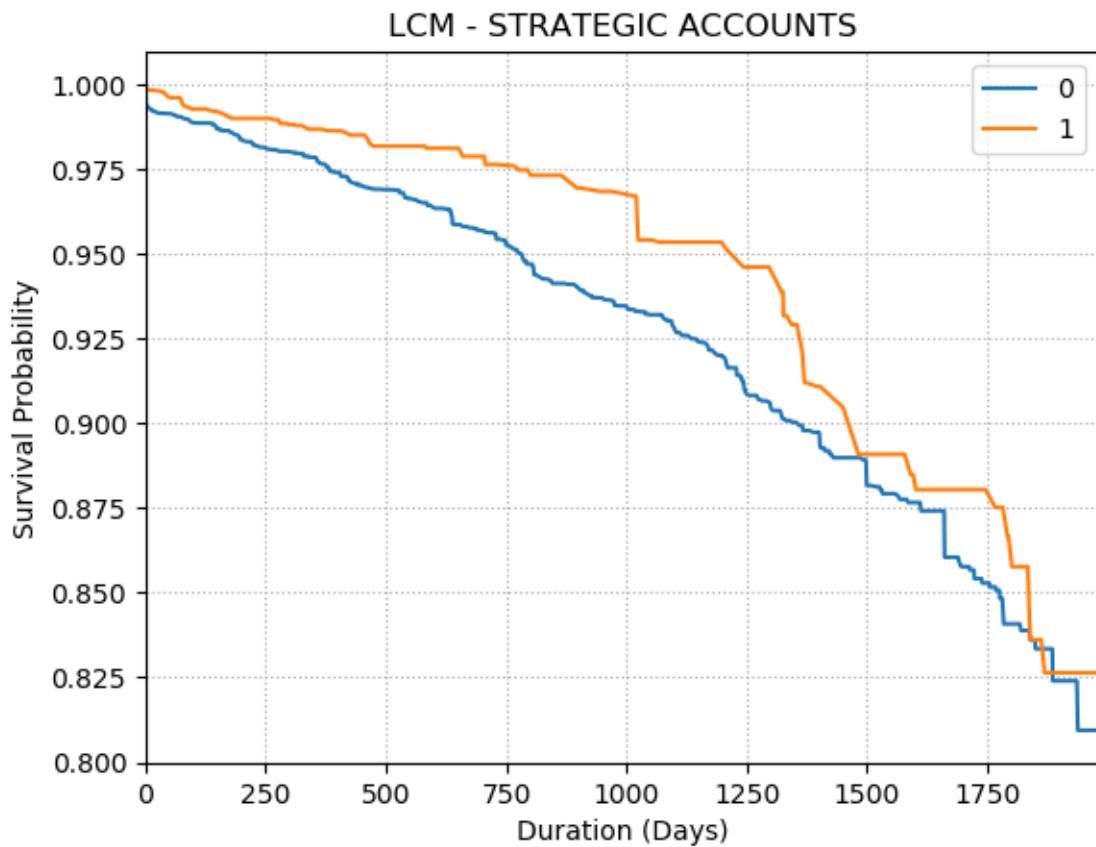
    idx = LCM_Final['STRATEGIC_ACCOUNTS'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('LCM - STRATEGIC ACCOUNTS')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()

```



4.4 TIJ

```
[113]: kmf1 = KaplanMeierFitter()

duration = TIJ_Final['TIJ_Tenure']
observed = TIJ_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TIJ_Final['STRATEGIC_ACCOUNTS'].unique()
ax = plt.subplot()

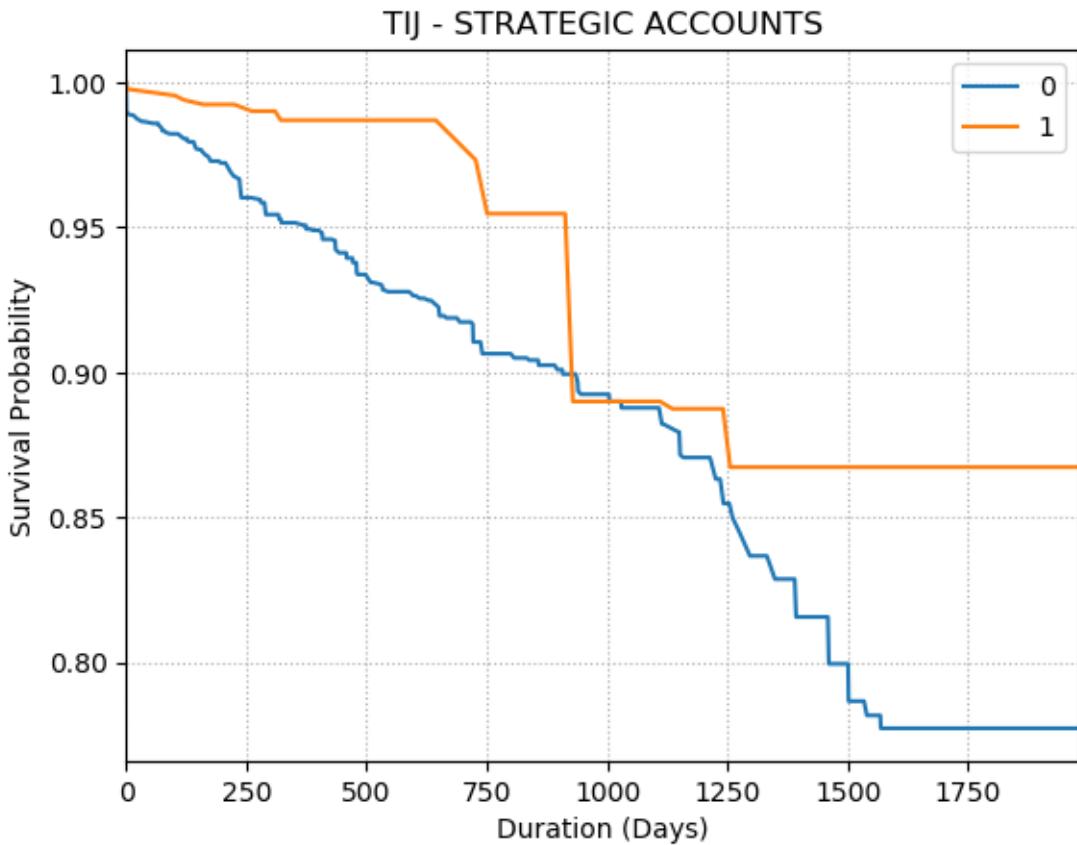
for pos in positions:

    idx = TIJ_Final['STRATEGIC_ACCOUNTS'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('TIJ - STRATEGIC ACCOUNTS')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()
```



5 Territory Region

5.1 CIJ

```
[114]: kmf1 = KaplanMeierFitter()

duration = CIJ_Final['CIJ_Tenure']
observed = CIJ_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = CIJ_Final['TERRITORY_REGION'].unique()
ax = plt.subplot()

for pos in positions:

    idx = CIJ_Final['TERRITORY_REGION'] == pos

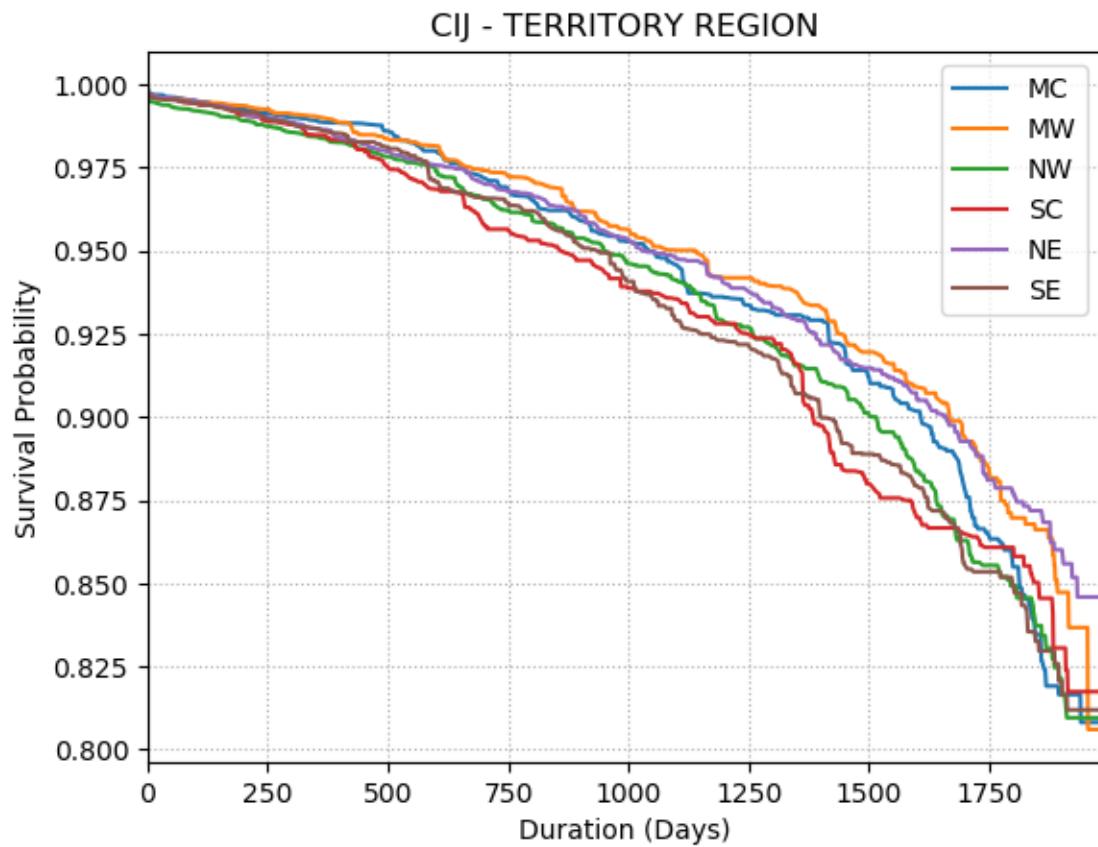
    kmf1.fit(duration[idx], observed[idx], label = pos)
```

```

kmf1.survival_function_.plot(ax=ax)
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('CIJ - TERRITORY REGION')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()

```



5.2 TTO

```

[115]: kmf1 = KaplanMeierFitter()

duration = TTO_Final['TTO_Tenure']
observed = TTO_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TTO_Final['TERRITORY_REGION'].unique()

```

```

ax = plt.subplot()

for pos in positions:

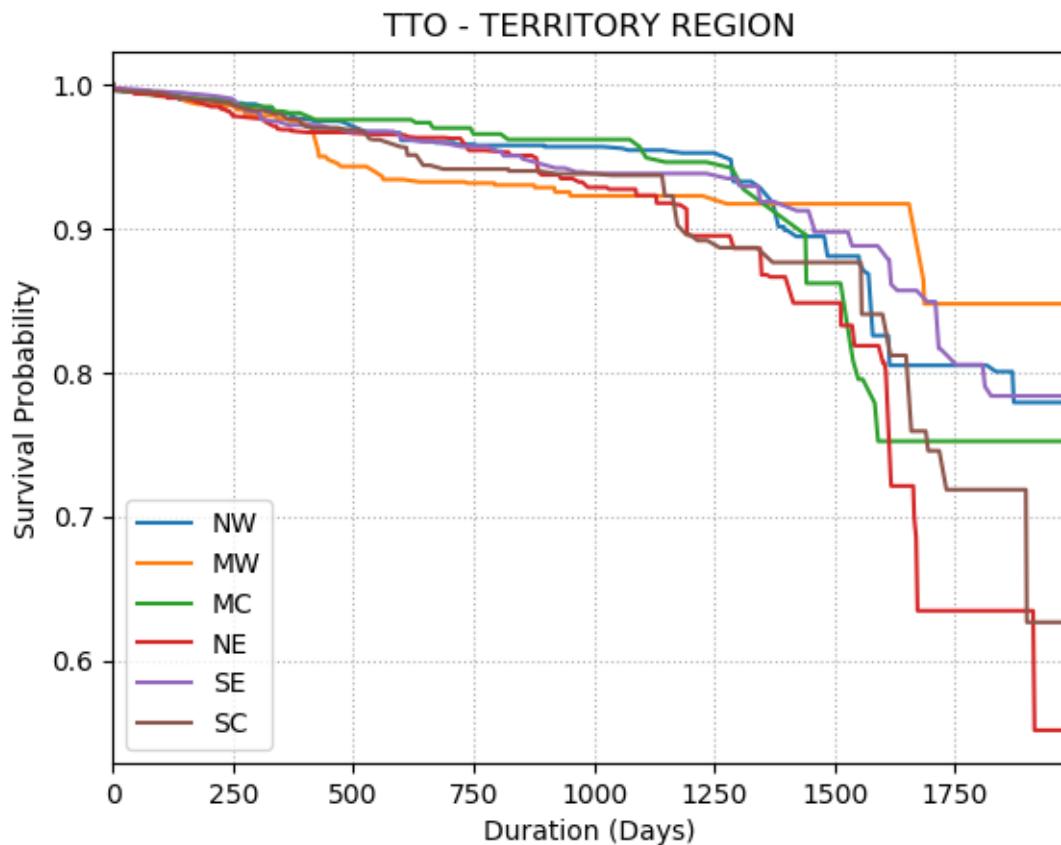
    idx = TTO_Final['TERRITORY_REGION'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('TTO - TERRITORY REGION')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()

```



5.3 LCM

```
[116]: kmf1 = KaplanMeierFitter()

duration = LCM_Final['LCM_Tenure']
observed = LCM_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = LCM_Final['TERRITORY_REGION'].unique()
ax = plt.subplot()

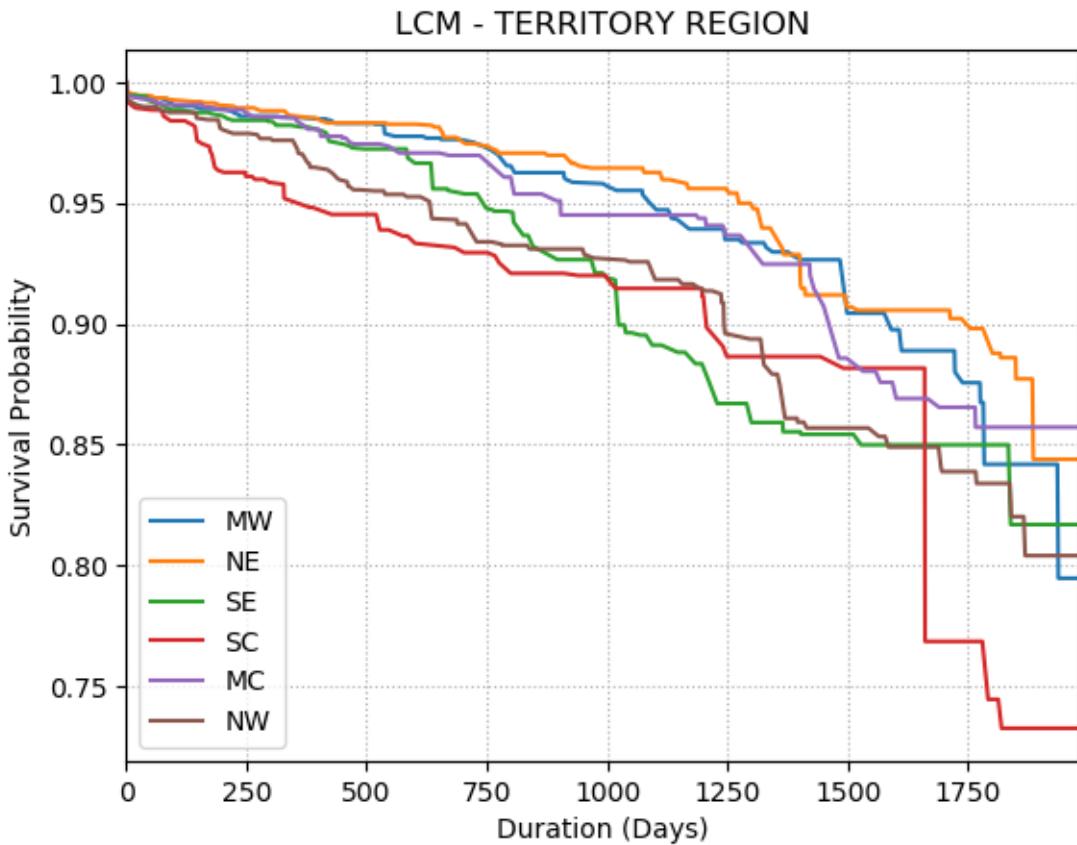
for pos in positions:

    idx = LCM_Final['TERRITORY_REGION'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('LCM - TERRITORY REGION')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()
```



5.4 TIJ

```
[117]: kmf1 = KaplanMeierFitter()

duration = TIJ_Final['TIJ_Tenure']
observed = TIJ_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TIJ_Final['TERRITORY_REGION'].unique()
ax = plt.subplot()

for pos in positions:

    idx = TIJ_Final['TERRITORY_REGION'] == pos

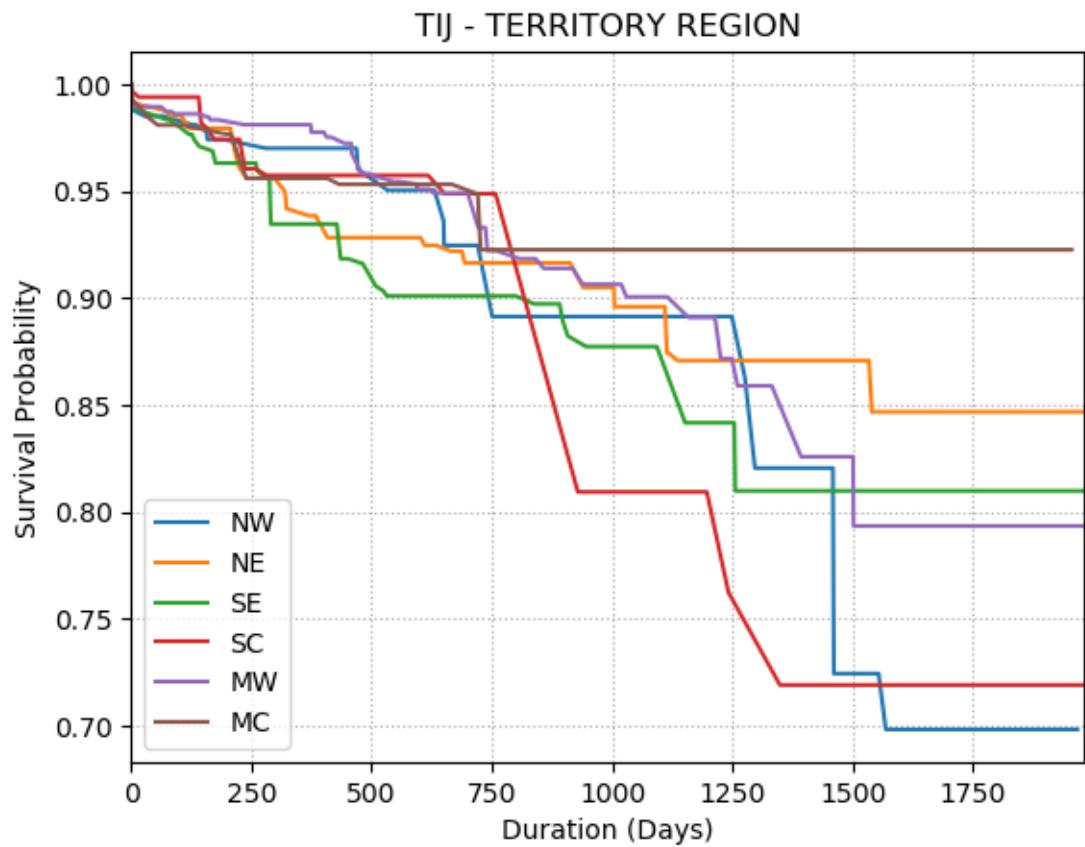
    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
```

```

ax.legend()
plt.title('TIJ - TERRITORY REGION')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()

```



6 Churn

6.1 CIJ

```

[118]: kmf1 = KaplanMeierFitter()

duration = CIJ_Final['CIJ_Tenure']
observed = CIJ_Final['Churned_BGNBD']

```

```

# Set the order that the positions will be plotted
positions = CIJ_Final['Churned_BGNBD'].unique()
ax = plt.subplot()

for pos in positions:

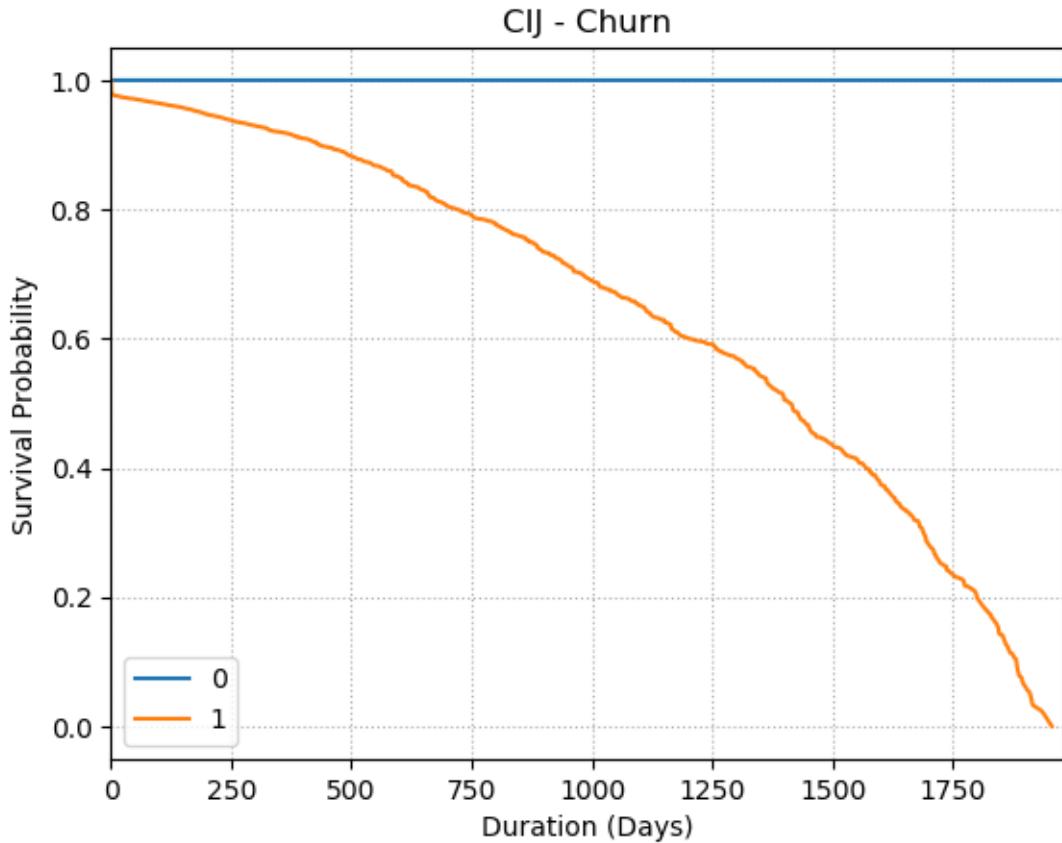
    idx = CIJ_Final['Churned_BGNBD'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

plt.title('CIJ - Churn')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
ax.legend()
plt.show()

```



6.2 TTO

```
[119]: kmf1 = KaplanMeierFitter()

duration = TTO_Final['TTO_Tenure']
observed = TTO_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TTO_Final['Churned_BGNBD'].unique()
ax = plt.subplot()

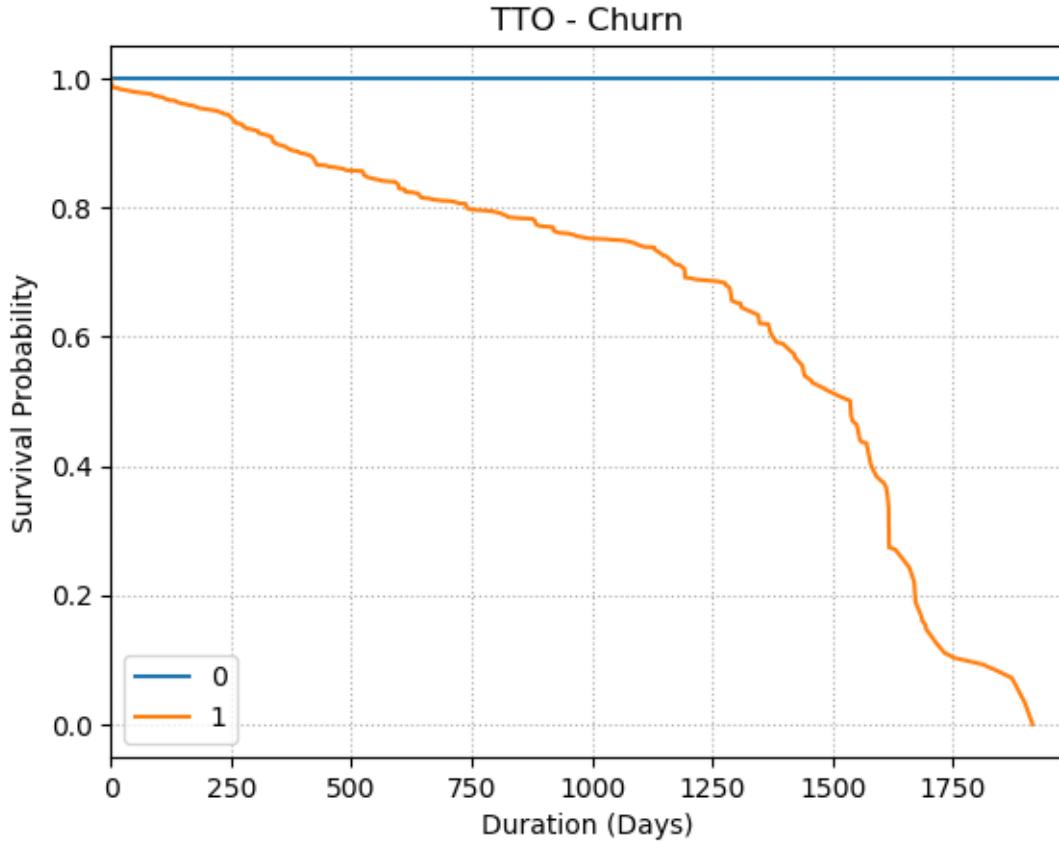
for pos in positions:

    idx = TTO_Final['Churned_BGNBD'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

plt.title('TTO - Churn')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
ax.legend()
plt.show()
```



6.3 LCM

```
[120]: kmf1 = KaplanMeierFitter()

duration = LCM_Final['LCM_Tenure']
observed = LCM_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = LCM_Final['Churned_BGNBD'].unique()
ax = plt.subplot()

for pos in positions:

    idx = LCM_Final['Churned_BGNBD'] == pos

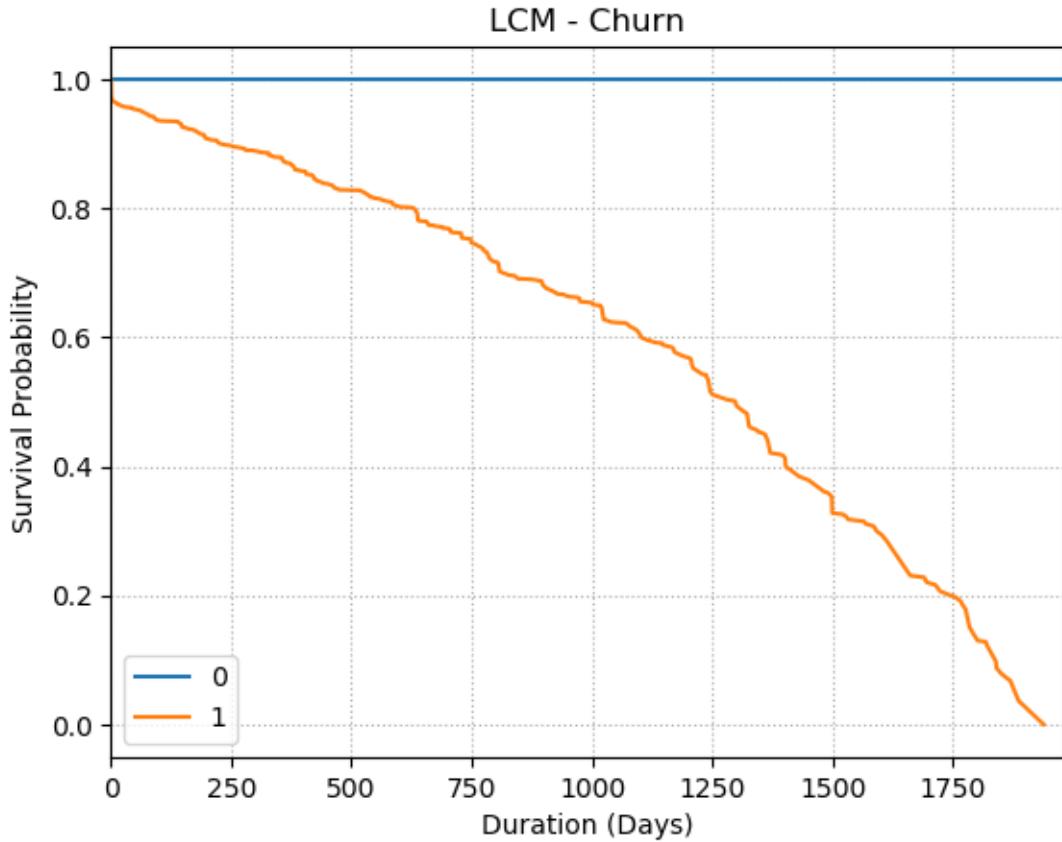
    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
```

```

plt.title('LCM - Churn')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
ax.legend()
plt.show()

```



6.4 TIJ

```

[121]: kmf1 = KaplanMeierFitter()

duration = TIJ_Final['TIJ_Tenure']
observed = TIJ_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TIJ_Final['Churned_BGNBD'].unique()
ax = plt.subplot()

```

```

for pos in positions:

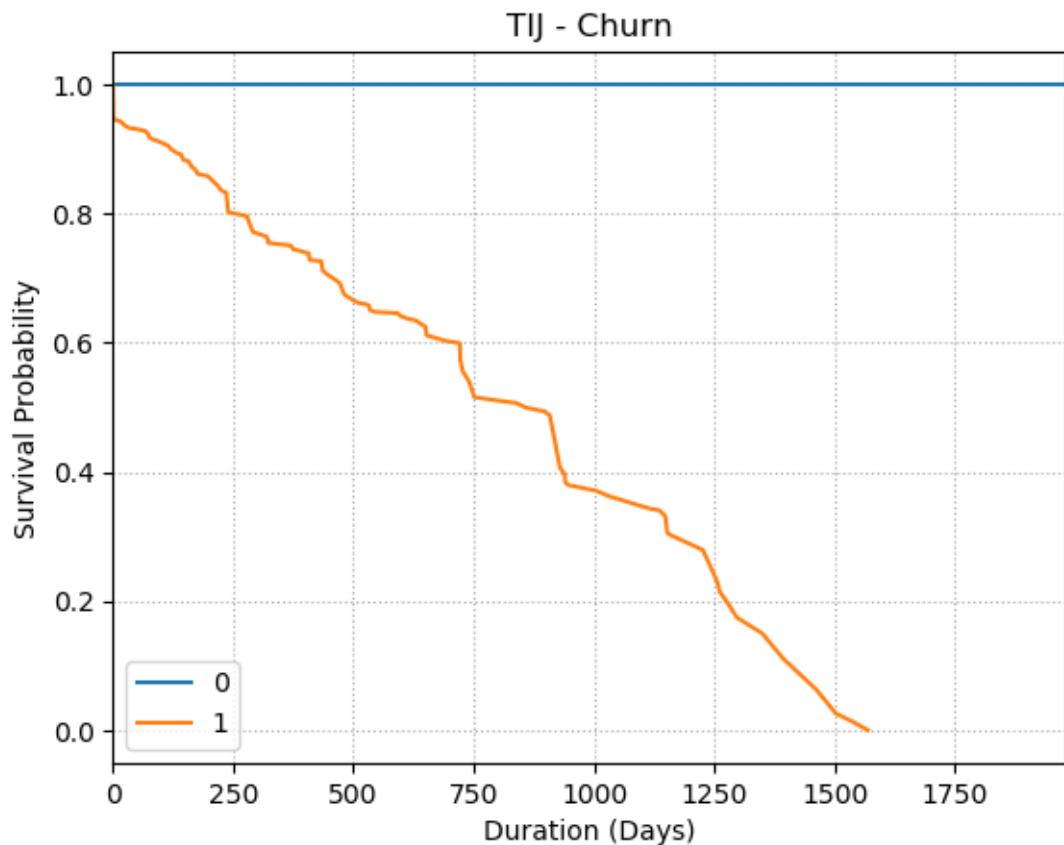
    idx = TIJ_Final['Churned_BGNBD'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

plt.title('TIJ - Churn')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
ax.legend()
plt.show()

```



7 Cluster

```
[36]: CIJ.dropna(inplace = True)
TTO.dropna(inplace = True)
LCM.dropna(inplace = True)
TIJ.dropna(inplace = True)
```

```
/Users/andrewchuang/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

"""Entry point for launching an IPython kernel.

```
/Users/andrewchuang/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
/Users/andrewchuang/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

This is separate from the ipykernel package so we can avoid doing imports until

```
/Users/andrewchuang/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
after removing the cwd from sys.path.

7.1 CIJ

```
[122]: kmf1 = KaplanMeierFitter()

duration = CIJ_Final['CIJ_Tenure']
observed = CIJ_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = CIJ_Final['Cluster_Id'].unique()
ax = plt.subplot()
```

```

for pos in positions:

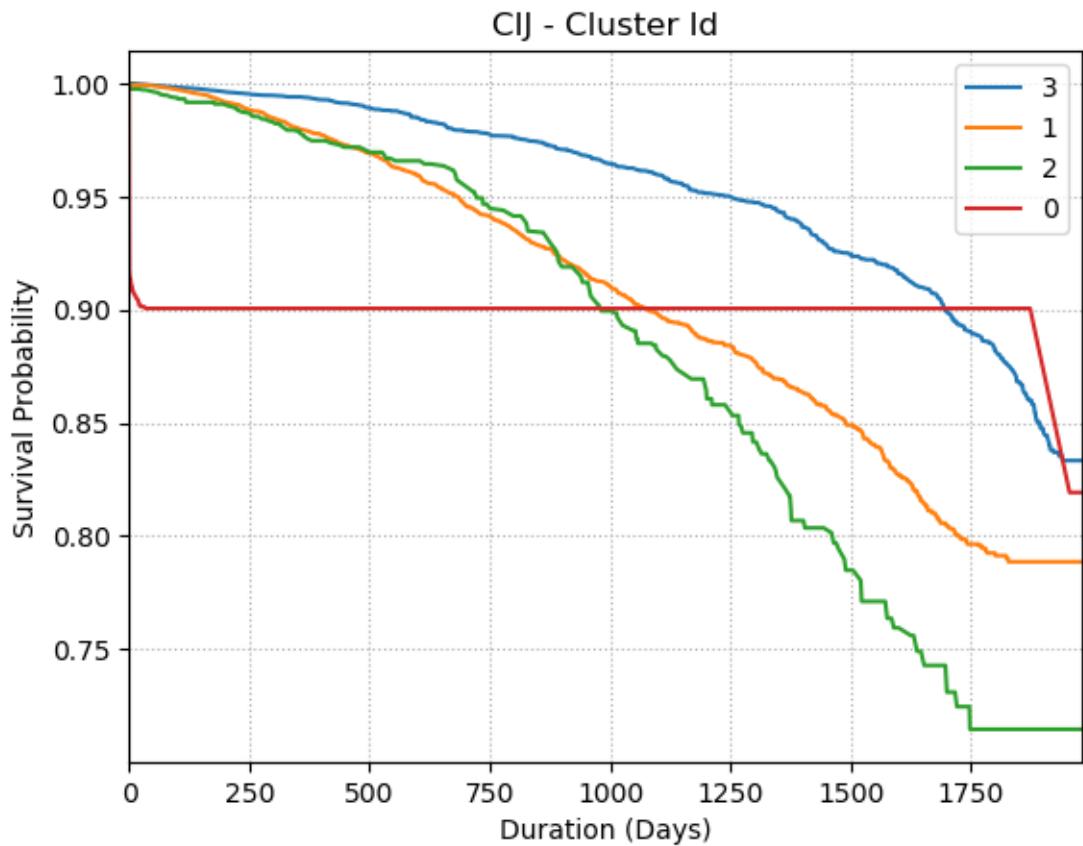
    idx = CIJ_Final['Cluster_Id'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

plt.title('CIJ - Cluster Id')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
ax.legend()
plt.show()

```



7.2 TTO

```
[123]: kmf1 = KaplanMeierFitter()

duration = TTO_Final['TTO_Tenure']
observed = TTO_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TTO_Final['Cluster_Id'].unique()
ax = plt.subplot()

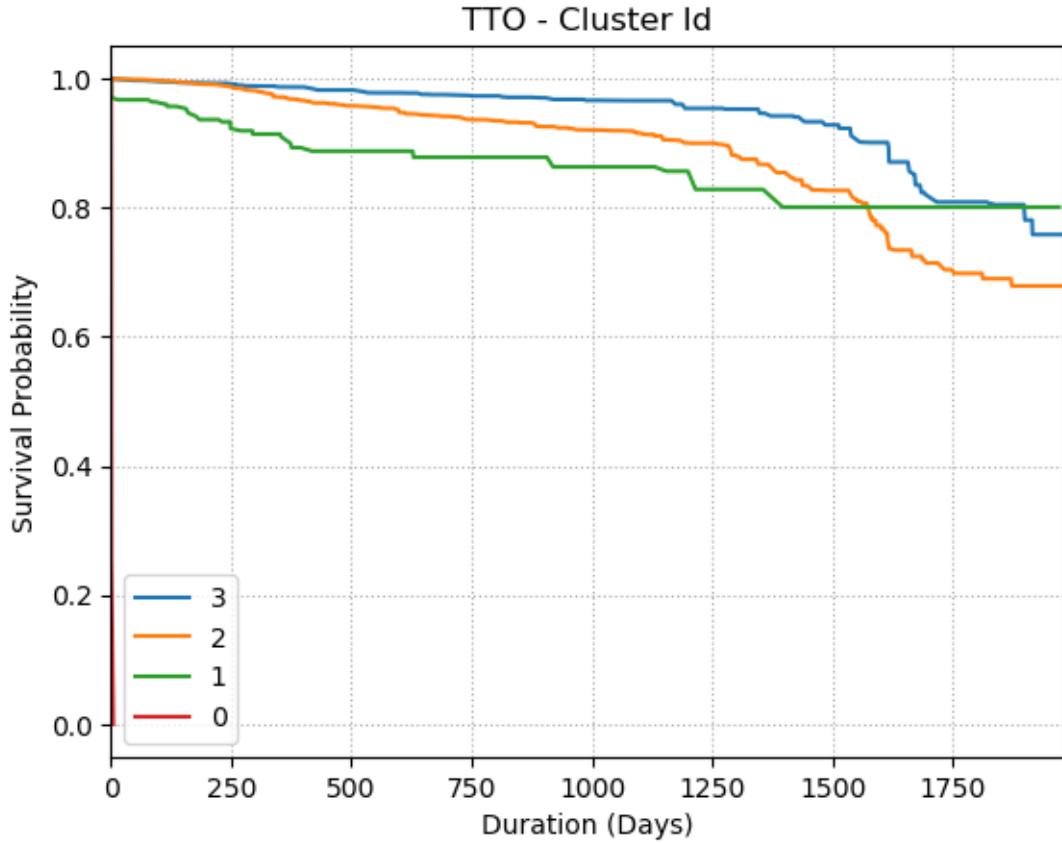
for pos in positions:

    idx = TTO_Final['Cluster_Id'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

plt.title('TTO - Cluster Id')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
ax.legend()
plt.show()
```



7.3 LCM

```
[124]: kmf1 = KaplanMeierFitter()

duration = LCM_Final['LCM_Tenure']
observed = LCM_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = LCM_Final['Cluster_Id'].unique()
ax = plt.subplot()

for pos in positions:

    idx = LCM_Final['Cluster_Id'] == pos

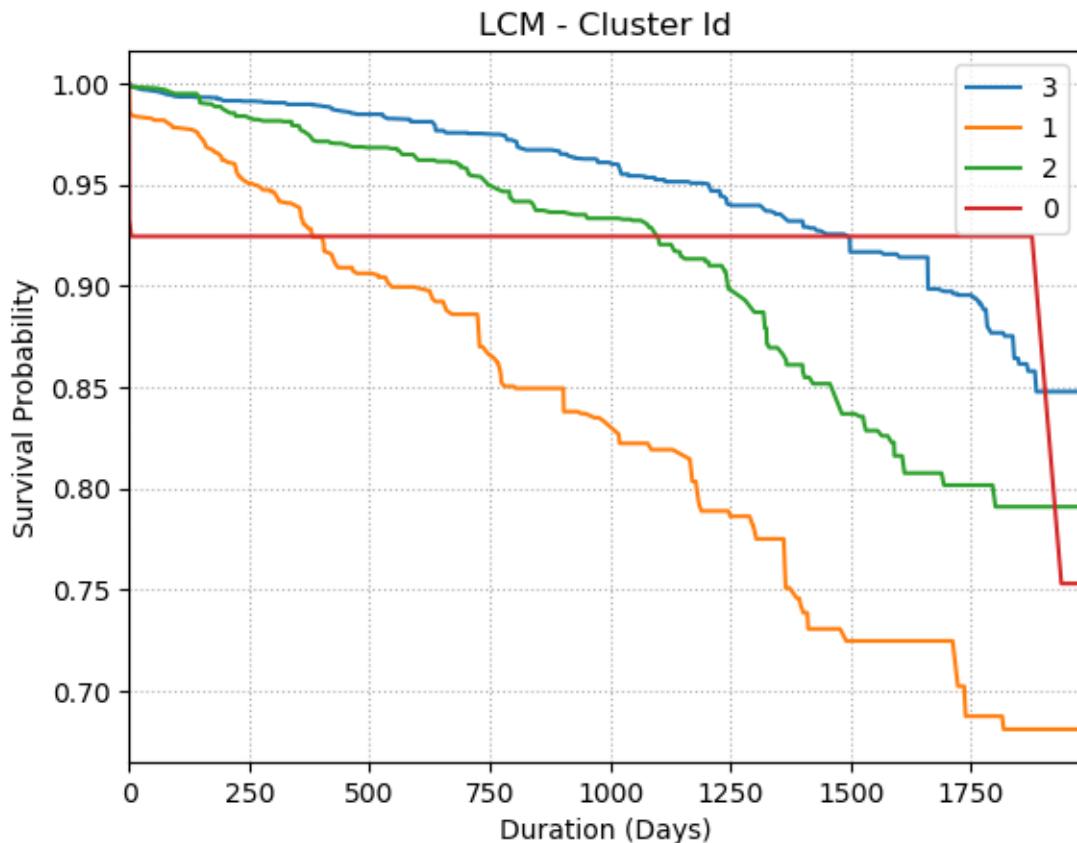
    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
```

```

plt.title('LCM - Cluster Id')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
ax.legend()
plt.show()

```



7.4 TIJ

```

[127]: kmf1 = KaplanMeierFitter()

duration = TIJ_Final['TIJ_Tenure']
observed = TIJ_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TIJ_Final['Cluster_Id'].unique()
ax = plt.subplot()

```

```

for pos in positions:

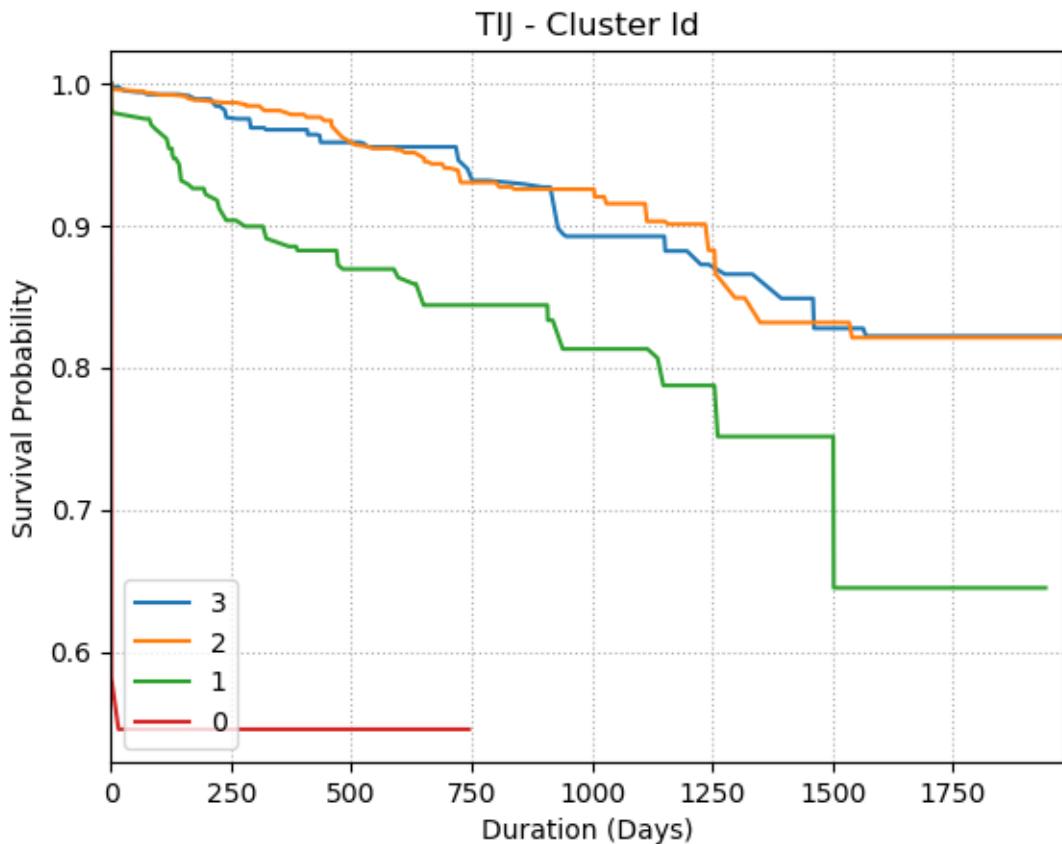
    idx = TIJ_Final['Cluster_Id'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

plt.title('TIJ - Cluster Id')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
ax.legend(loc = 'lower left')
plt.show()

```



8 SALES_CHANNEL

8.1 CIJ

```
[128]: kmf1 = KaplanMeierFitter()

duration = CIJ_Final['CIJ_Tenure']
observed = CIJ_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = CIJ_Final['SALES_CHANNEL'].unique()

ax = plt.subplot()

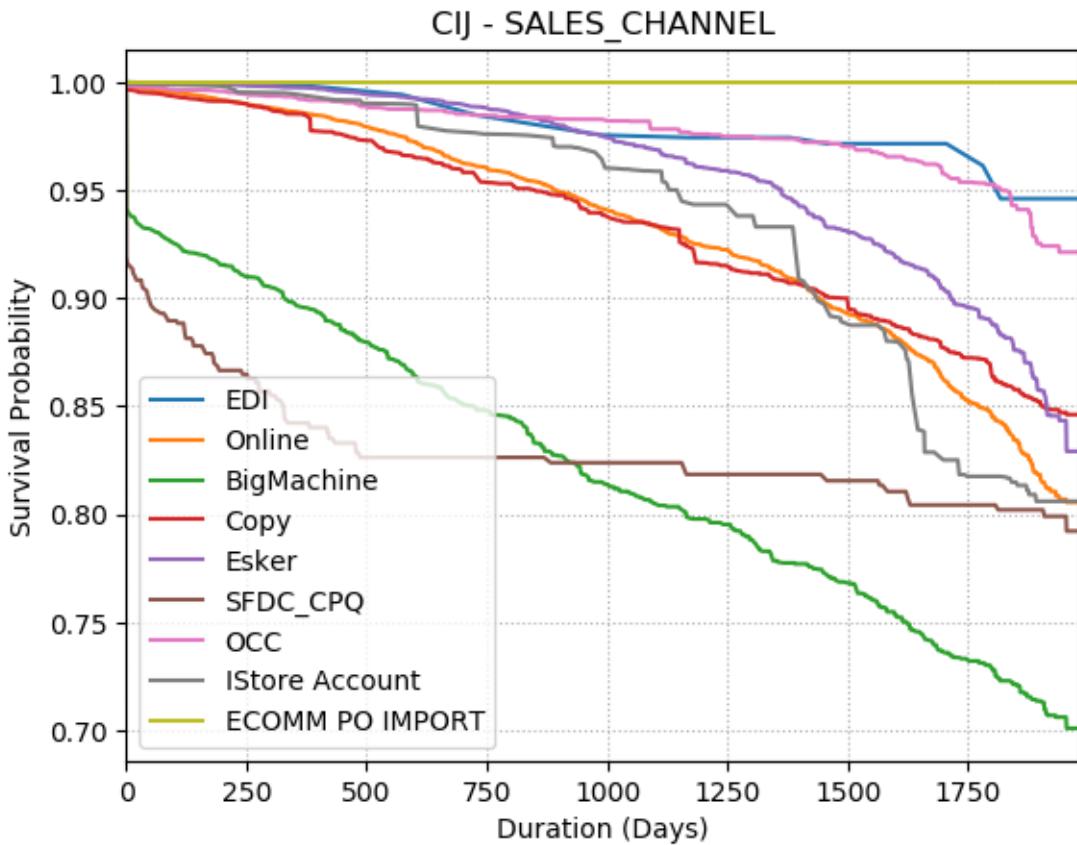
for pos in positions:

    idx = CIJ_Final['SALES_CHANNEL'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('CIJ - SALES_CHANNEL')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()
```



8.2 TTO

```
[129]: kmf1 = KaplanMeierFitter()

duration = TTO_Final['TTO_Tenure']
observed = TTO_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TTO_Final['SALES_CHANNEL'].unique()

ax = plt.subplot()

for pos in positions:

    idx = TTO_Final['SALES_CHANNEL'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

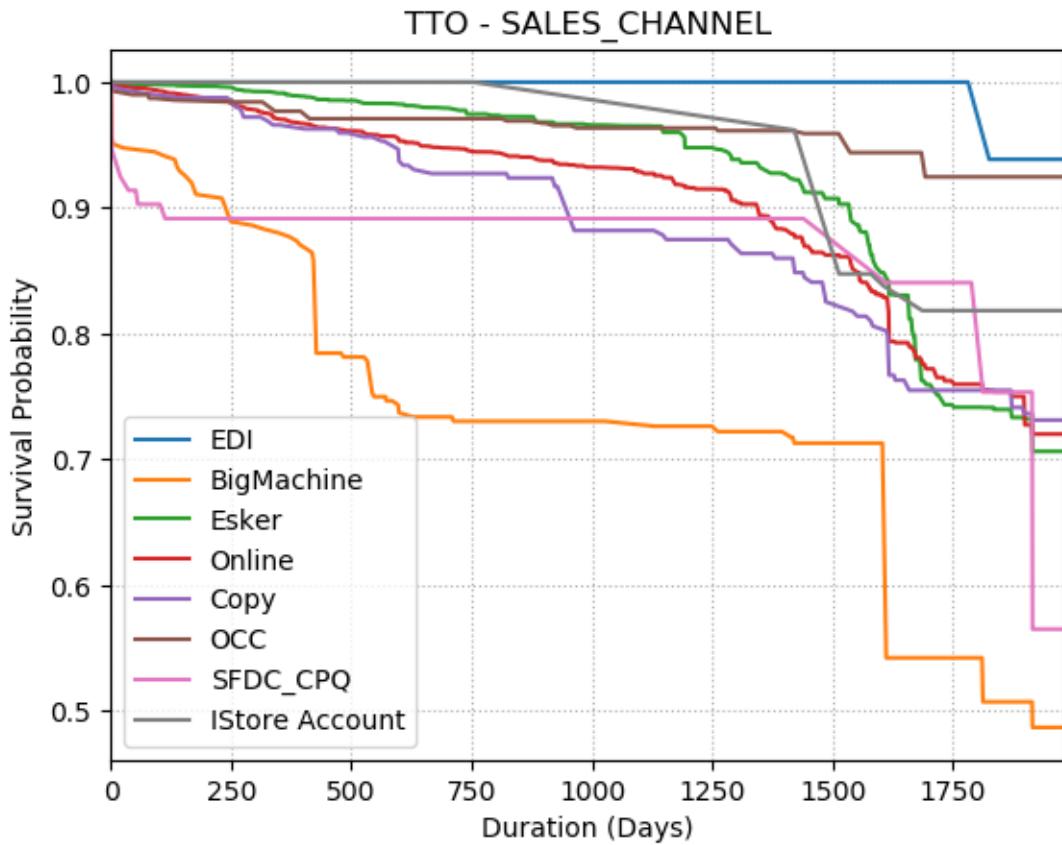
    kmf1.survival_function_.plot(ax=ax)
```

```

ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('TTO - SALES_CHANNEL')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()

```



8.3 LCM

```
[43]: LCM['SALES_CHANNEL'].value_counts()
```

[43]: Online	16080
Esker	7388
OCC	1129
BigMachine	654
Copy	520
EDI	405

```
IStore Account      310
SFDC_CPQ           162
ECOMM PO IMPORT     5
Service Billing      1
Name: SALES_CHANNEL, dtype: int64
```

```
[130]: kmf1 = KaplanMeierFitter()

duration = LCM_Final['LCM_Tenure']
observed = LCM_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = LCM_Final['SALES_CHANNEL'].unique()

ax = plt.subplot()

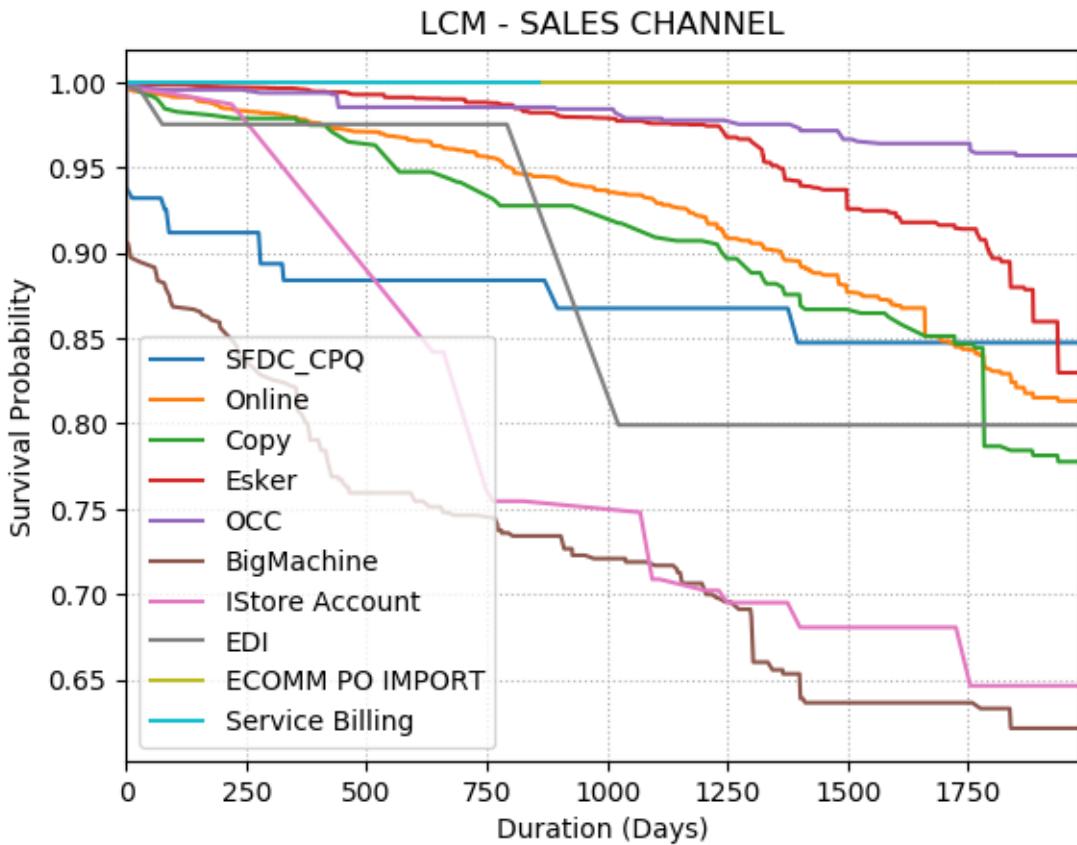
for pos in positions:

    idx = LCM_Final['SALES_CHANNEL'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('LCM - SALES CHANNEL')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()
```



8.4 TIJ

```
[131]: kmf1 = KaplanMeierFitter()

duration = TIJ_Final['TIJ_Tenure']
observed = TIJ_Final['Churned_BGNBD']

# Set the order that the positions will be plotted
positions = TIJ_Final['SALES_CHANNEL'].unique()

ax = plt.subplot()

for pos in positions:

    idx = TIJ_Final['SALES_CHANNEL'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

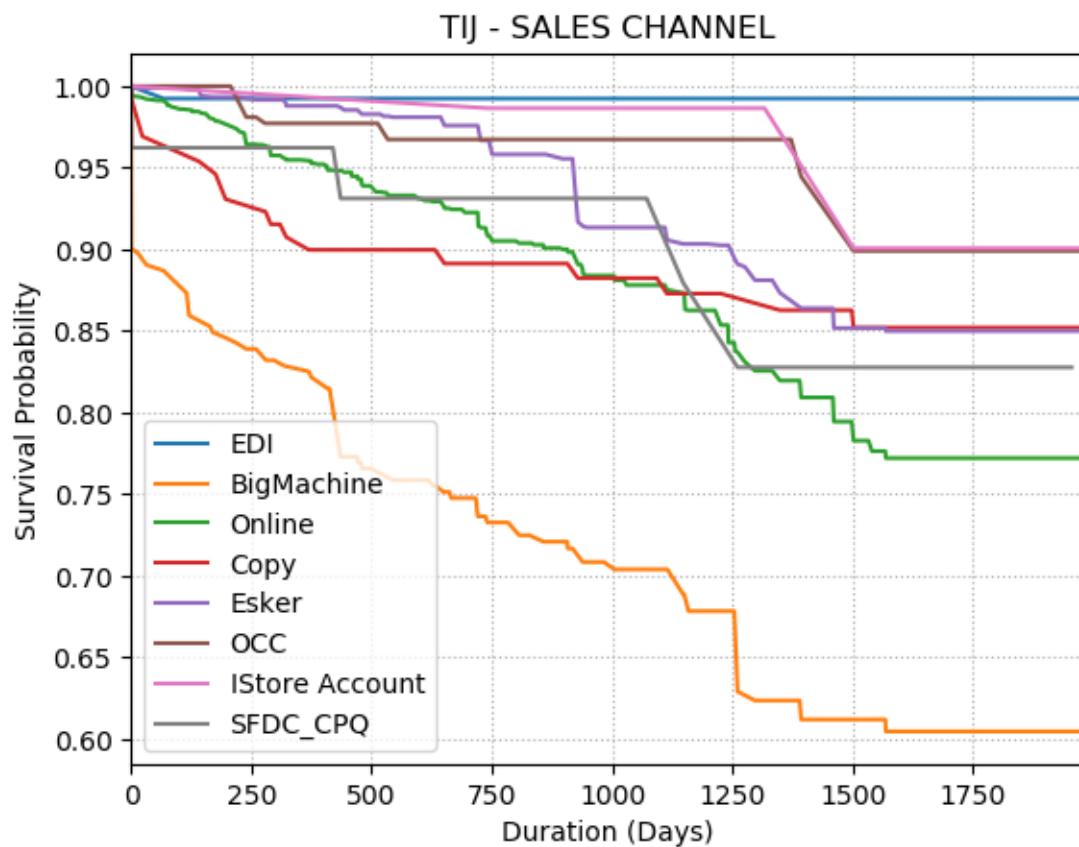
    kmf1.survival_function_.plot(ax=ax)
```

```

ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

ax.legend()
plt.title('TIJ - SALES CHANNEL')
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.grid(linestyle='dotted')
plt.style.use('default')
plt.show()

```



9 SUPPLIES_DECLINE_REASON

9.1 CIJ

```
[132]: CIJ_Final_SDR = CIJ_Final[CIJ_Final['SUPPLIES_DECLINE_REASON'] != 'None']
```

```
[133]: CIJ_Final_SDR['SUPPLIES_DECLINE_REASON'].unique()
```

```
[133]: array(['Migration to 1000 Line/TIJ/TTO/LCM/LPA', 'Over Stocked / Timing',
   'Off Brand', 'Production Down (timing)', 'Site Closed',
   'AP Competitive Displacement', 'Migration to Lasers',
   'Production / Code Reduction', 'Moved Equipment',
   'Recent Regain/Win-back', 'Served by Authorized Distributor',
   'No More Coding Requirement', 'Seasonal Producer', 'Project Based',
   'Printing/EQ downtime Issues', 'VJ Operations Issues',
   'Pricing / Discounting', 'Financial Distress/Credit Hold'],
  dtype=object)
```

```
[134]: kmf1 = KaplanMeierFitter()

plt.figure(figsize = (15,10))
duration = CIJ_Final_SDR['CIJ_Tenure']
observed = CIJ_Final_SDR['Churned_BGNBD']

# Set the order that the positions will be plotted
positions_1 = ['Migration to 1000 Line/TIJ/TTO/LCM/LPA',
   'Over Stocked / Timing', 'Off Brand', 'Production Down (timing)',
   'Site Closed', 'AP Competitive Displacement',
   'Migration to Lasers', 'Production / Code Reduction',
   'Moved Equipment', 'Recent Regain/Win-back']

positions_2 = ['Served by Authorized Distributor', 'No More Coding Requirement',
   'Seasonal Producer', 'Project Based',
   'Printing/EQ downtime Issues', 'VJ Operations Issues',
   'Pricing / Discounting', 'Financial Distress/Credit Hold']

ax1 = plt.subplot(221)
ax2 = plt.subplot(222, sharey = ax1)

ax1.title.set_text('1')
ax2.title.set_text('2')

for pos in positions_1:

    idx = CIJ_Final_SDR['SUPPLIES_DECLINE_REASON'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax1)
    ax1.grid(linestyle='dotted')
    ax1.legend()
```

```

for pos in positions_2:

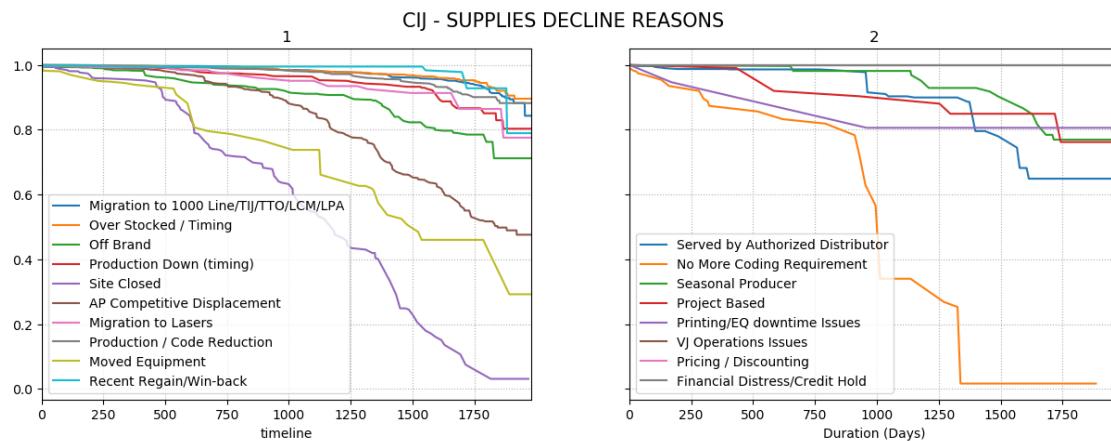
    idx = CIJ_Final_SDR['SUPPLIES_DECLINE_REASONS'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax2)
    ax2.grid(linestyle='dotted')
    ax2.legend()

plt.suptitle('CIJ - SUPPLIES DECLINE REASONS', fontsize = 15)
plt.subplots_adjust(top=0.94)
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.style.use('default')
plt.show()

```



9.2 TTO

```
[135]: TTO_Final_SDR = TTO_Final[TTO_Final['SUPPLIES_DECLINE_REASONS'] != 'None']
```

```
[136]: kmf1 = KaplanMeierFitter()
```

```

plt.figure(figsize = (15,10))
duration = TTO_Final_SDR['TTO_Tenure']
observed = TTO_Final_SDR['Churned_BGNBD']

# Set the order that the positions will be plotted
positions_1 = ['AP Competitive Displacement', 'Off Brand',
              'Over Stocked / Timing', 'Migration to 1000 Line/TIJ/TTO/LCM/LPA',
              'Served by Authorized Distributor', 'No More Coding Requirement',
              'Seasonal Producer', 'Project Based',
              'Printing/EQ downtime Issues', 'VJ Operations Issues',
              'Pricing / Discounting', 'Financial Distress/Credit Hold']

```

```

'Production / Code Reduction', 'Production Down (timing)',
'Site Closed', 'Project Based', 'Served by Authorized Distributor',
'Migration to Lasers']

positions_2 = ['Recent Regain/Win-back',
    'VJ Operations Issues', 'Seasonal Producer',
    'Printing/EQ downtime Issues', 'Moved Equipment',
    'Pricing / Discounting', 'No More Coding Requirement']

ax1 = plt.subplot(221)
ax2 = plt.subplot(222, sharey = ax1)

ax1.title.set_text('1')
ax2.title.set_text('2')

for pos in positions_1:

    idx = TTO_Final_SDR['SUPPLIES_DECLINE_REASON'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax1)
    ax1.grid(linestyle='dotted')
    ax1.legend()

for pos in positions_2:

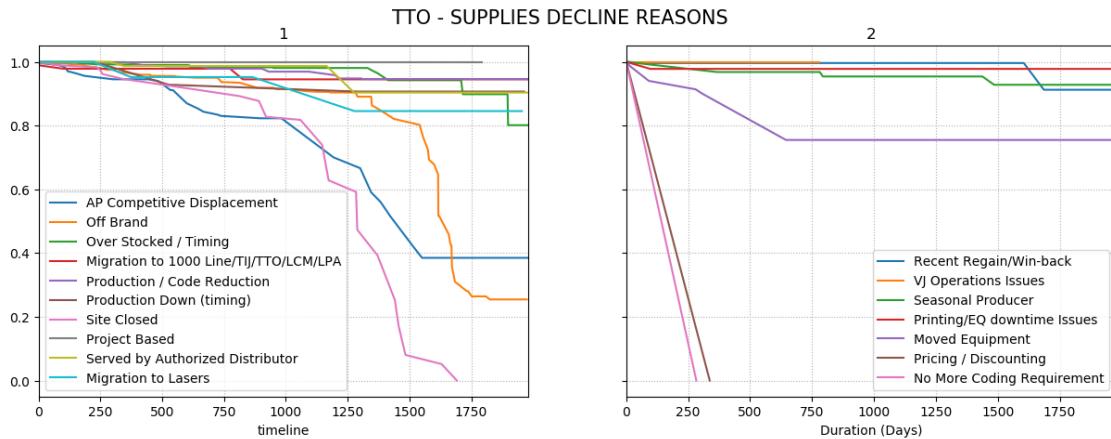
    idx = TTO_Final_SDR['SUPPLIES_DECLINE_REASON'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax2)
    ax2.grid(linestyle='dotted')
    ax2.legend()

plt.suptitle('TTO - SUPPLIES DECLINE REASONS', fontsize = 15)
plt.subplots_adjust(top=0.94)
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.style.use('default')
plt.show()

```



9.3 LCM

```
[137]: LCM_Final_SDR = LCM_Final[LCM_Final['SUPPLIES_DECLINE_REASONS'] != 'None']
```

```
[138]: kmf1 = KaplanMeierFitter()

plt.figure(figsize = (15,10))
duration = LCM_Final_SDR['LCM_Tenure']
observed = LCM_Final_SDR['Churned_BGNBD']

# Set the order that the positions will be plotted
positions_1 = ['Over Stocked / Timing', 'AP Competitive Displacement',
    'Production / Code Reduction',
    'Migration to 1000 Line/TIJ/TTO/LCM/LPA', 'Off Brand',
    'Production Down (timing)', 'Moved Equipment',
    'Served by Authorized Distributor', 'Recent Regain/Win-back']

positions_2 = ['Site Closed', 'Migration to Lasers', 'Project Based',
    'Seasonal Producer', 'Printing/EQ downtime Issues',
    'No More Coding Requirement', 'VJ Operations Issues']

ax1 = plt.subplot(221)
ax2 = plt.subplot(222, sharey = ax1)

ax1.title.set_text('1')
ax2.title.set_text('2')

for pos in positions_1:
```

```

idx = LCM_Final_SDR['SUPPLIES_DECLINE_REASONS'] == pos

kmf1.fit(duration[idx], observed[idx], label = pos)

kmf1.survival_function_.plot(ax=ax1)
ax1.grid(linestyle='dotted')
ax1.legend()

for pos in positions_2:

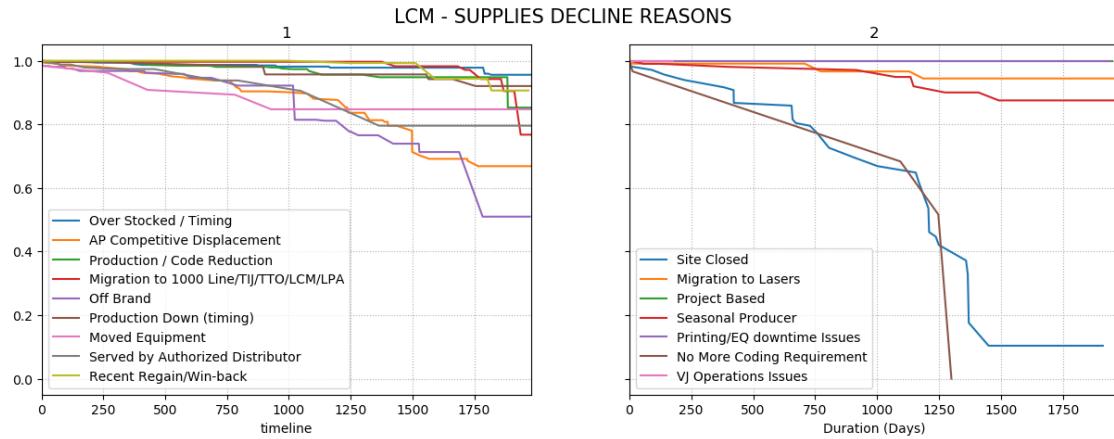
    idx = LCM_Final_SDR['SUPPLIES_DECLINE_REASONS'] == pos

    kmf1.fit(duration[idx], observed[idx], label = pos)

    kmf1.survival_function_.plot(ax=ax2)
    ax2.grid(linestyle='dotted')
    ax2.legend()

plt.suptitle('LCM - SUPPLIES DECLINE REASONS', fontsize = 15)
plt.subplots_adjust(top=0.94)
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.style.use('default')
plt.show()

```



9.4 TIJ

```
[139]: TIJ_Final_SDR = TIJ_Final[TIJ_Final['SUPPLIES_DECLINE_REASONS'] != 'None']
```

```
[140]: kmf1 = KaplanMeierFitter()

plt.figure(figsize = (15,10))
duration = TIJ_Final_SDR['TIJ_Tenure']
observed = TIJ_Final_SDR['Churned_BGNBD']

# Set the order that the positions will be plotted
positions_1 = ['Over Stocked / Timing', 'Production Down (timing)',  

    'AP Competitive Displacement', 'Off Brand',  

    'Recent Regain/Win-back', 'Production / Code Reduction',  

    'Site Closed', 'Migration to Lasers']

positions_2 = ['Served by Authorized Distributor',  

    'Migration to 1000 Line/TIJ/TTO/LCM/LPA', 'Moved Equipment',  

    'Project Based']

ax1 = plt.subplot(221)
ax2 = plt.subplot(222, sharey = ax1)

ax1.title.set_text('1')
ax2.title.set_text('2')

for pos in positions_1:  

    idx = TIJ_Final_SDR['SUPPLIES_DECLINE_REASON'] == pos  

    kmf1.fit(duration[idx], observed[idx], label = pos)  

    kmf1.survival_function_.plot(ax=ax1)
    ax1.grid(linestyle='dotted')
    ax1.legend()  

for pos in positions_2:  

    idx = TIJ_Final_SDR['SUPPLIES_DECLINE_REASON'] == pos  

    kmf1.fit(duration[idx], observed[idx], label = pos)  

    kmf1.survival_function_.plot(ax=ax2)
    ax2.grid(linestyle='dotted')
    ax2.legend()  

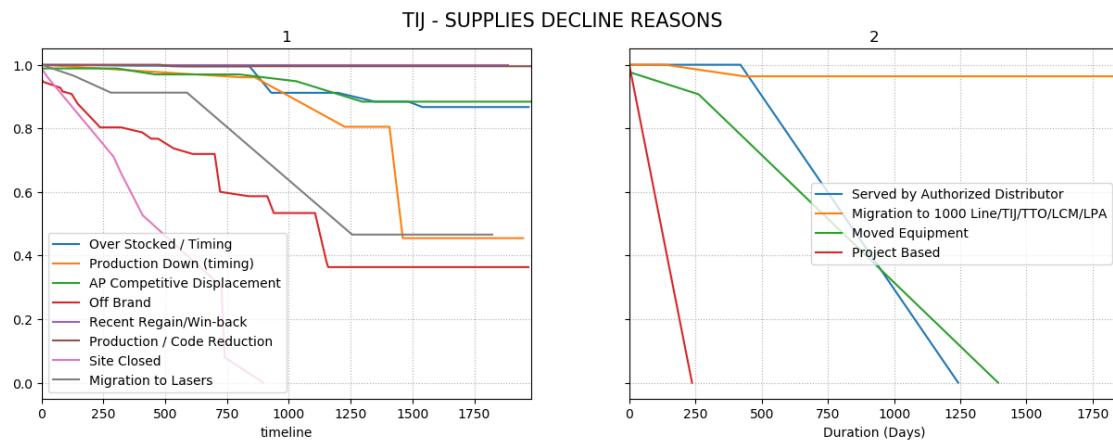
  

plt.suptitle('TIJ - SUPPLIES DECLINE REASONS', fontsize = 15)
```

```

plt.subplots_adjust(top=0.94)
plt.xlabel('Duration (Days)')
plt.ylabel('Survival Probability')
plt.style.use('default')
plt.show()

```



[]: