In [1]:
```
#Load CSV using pandas
import pandas as pd
```

In [ ]:
```
#question One: Show the steps you would take to clean data for Modelling?

i. Identify Outliers and remove them (groupimg them in sandbox for future use if needed
ii. Handling missing value and replaced with Nan
iii. Identifying duplicates in the data and delete it
iv. Mark empty value as missing value and mark missing value as Nan
v. Identify the columns that contain a single value and delete or replace it with Nan
vi. Check columns with values less than 5% and delete it
vii. Missing data imputation using statistical methods (mean, median, mode, constant, e
```

In [2]:
```
filename = 'Churn_Train.csv'
filepath = 'C:/Users/User/Documents/Workspace/'
data = pd.read_csv(filepath + filename)
```

In [3]:
```
data
```

Out[3]:

	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages
0	NV	125.0	area_code_510	no	no	0.0
1	HI	108.0	area_code_415	no	no	0.0
2	DC	82.0	area_code_415	no	no	0.0
3	HI	NaN	area_code_408	no	yes	30.0
4	OH	83.0	area_code_415	no	no	0.0
...	...	...	...	...	...	...
3328	OH	144.0	area_code_415	no	yes	18.0
3329	LA	69.0	area_code_415	no	yes	37.0
3330	SD	NaN	area_code_415	no	no	0.0
3331	NY	39.0	area_code_408	no	no	0.0
3332	WI	41.0	area_code_408	no	no	0.0

3333 rows × 20 columns

In [4]:
```
data.describe()
```

Out[4]:

	account_length	number_vmail_messages	total_day_minutes	total_day_calls	total_day_charge	tot
count	2832.000000	3133.000000	3133.000000	3133.000000	3133.000000	
mean	97.321328	7.332589	418.947048	100.331631	30.628455	

	account_length	number_vmail_messages	total_day_minutes	total_day_calls	total_day_charge	tot
std	47.874422	13.756056	626.315020	20.039364	9.275752	
min	-209.000000	-10.000000	0.000000	0.000000	0.000000	
25%	72.000000	0.000000	149.300000	87.000000	24.450000	
50%	100.000000	0.000000	190.500000	101.000000	30.650000	
75%	127.000000	16.000000	237.800000	114.000000	36.840000	
max	243.000000	51.000000	2185.100000	165.000000	59.640000	

In [5]:
```
data.isna().sum()
```

Out[5]:
```
state                          0
account_length                 501
area_code                      0
international_plan             0
voice_mail_plan                0
number_vmail_messages          200
total_day_minutes              200
total_day_calls                200
total_day_charge               200
total_eve_minutes              301
total_eve_calls                200
total_eve_charge               200
total_night_minutes            200
total_night_calls              0
total_night_charge             200
total_intl_minutes             200
total_intl_calls               301
total_intl_charge              200
number_customer_service_calls  200
churn                          0
dtype: int64
```

In [ ]:
```
```

In [7]:
```
# Question Two: Describe the data, provide an overview of the data, including data expl

data
```

Out[7]:

	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages
0	NV	125.0	area_code_510	no	no	0.0
1	HI	108.0	area_code_415	no	no	0.0
2	DC	82.0	area_code_415	no	no	0.0
3	HI	NaN	area_code_408	no	yes	30.0
4	OH	83.0	area_code_415	no	no	0.0
...	...	...	...	...	...	...
3328	OH	144.0	area_code_415	no	yes	18.0

	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages
3329	LA	69.0	area_code_415	no	yes	37.0
3330	SD	NaN	area_code_415	no	no	0.0
3331	NY	39.0	area_code_408	no	no	0.0
3332	WI	41.0	area_code_408	no	no	0.0

3333 rows × 20 columns

In [8]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 20 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   state                           3333 non-null   object
 1   account_length                  2832 non-null   float64
 2   area_code                       3333 non-null   object
 3   international_plan              3333 non-null   object
 4   voice_mail_plan                 3333 non-null   object
 5   number_vmail_messages           3133 non-null   float64
 6   total_day_minutes               3133 non-null   float64
 7   total_day_calls                 3133 non-null   float64
 8   total_day_charge                3133 non-null   float64
 9   total_eve_minutes               3032 non-null   float64
 10  total_eve_calls                 3133 non-null   float64
 11  total_eve_charge                3133 non-null   float64
 12  total_night_minutes             3133 non-null   float64
 13  total_night_calls               3333 non-null   int64
 14  total_night_charge              3133 non-null   float64
 15  total_intl_minutes              3133 non-null   float64
 16  total_intl_calls                3032 non-null   float64
 17  total_intl_charge               3133 non-null   float64
 18  number_customer_service_calls   3133 non-null   float64
 19  churn                           3333 non-null   object
dtypes: float64(14), int64(1), object(5)
memory usage: 520.9+ KB
```

In [9]:
```
data.dtypes
```

Out[9]:
```
state                           object
account_length                 float64
area_code                       object
international_plan              object
voice_mail_plan                 object
number_vmail_messages          float64
total_day_minutes              float64
total_day_calls                float64
total_day_charge               float64
total_eve_minutes              float64
total_eve_calls                float64
total_eve_charge               float64
total_night_minutes            float64
total_night_calls                int64
total_night_charge             float64
```

```
total_intl_minutes              float64
total_intl_calls                float64
total_intl_charge               float64
number_customer_service_calls   float64
churn                            object
dtype: object
```

In [10]:
```
data.columns
```

Out[10]:
```
Index(['state', 'account_length', 'area_code', 'international_plan',
       'voice_mail_plan', 'number_vmail_messages', 'total_day_minutes',
       'total_day_calls', 'total_day_charge', 'total_eve_minutes',
       'total_eve_calls', 'total_eve_charge', 'total_night_minutes',
       'total_night_calls', 'total_night_charge', 'total_intl_minutes',
       'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls', 'churn'],
      dtype='object')
```

In [11]:
```
data.describe()
```

Out[11]:

	account_length	number_vmail_messages	total_day_minutes	total_day_calls	total_day_charge	tot
count	2832.000000	3133.000000	3133.000000	3133.000000	3133.000000	
mean	97.321328	7.332589	418.947048	100.331631	30.628455	
std	47.874422	13.756056	626.315020	20.039364	9.275752	
min	-209.000000	-10.000000	0.000000	0.000000	0.000000	
25%	72.000000	0.000000	149.300000	87.000000	24.450000	
50%	100.000000	0.000000	190.500000	101.000000	30.650000	
75%	127.000000	16.000000	237.800000	114.000000	36.840000	
max	243.000000	51.000000	2185.100000	165.000000	59.640000	

In [12]:
```
data.shape
```

Out[12]:
```
(3333, 20)
```

In [13]:
```
data.head(15)
```

Out[13]:

	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages	t
0	NV	125.0	area_code_510	no	no	0.0	
1	HI	108.0	area_code_415	no	no	0.0	
2	DC	82.0	area_code_415	no	no	0.0	
3	HI	NaN	area_code_408	no	yes	30.0	
4	OH	83.0	area_code_415	no	no	0.0	
5	MO	89.0	area_code_415	no	no	0.0	

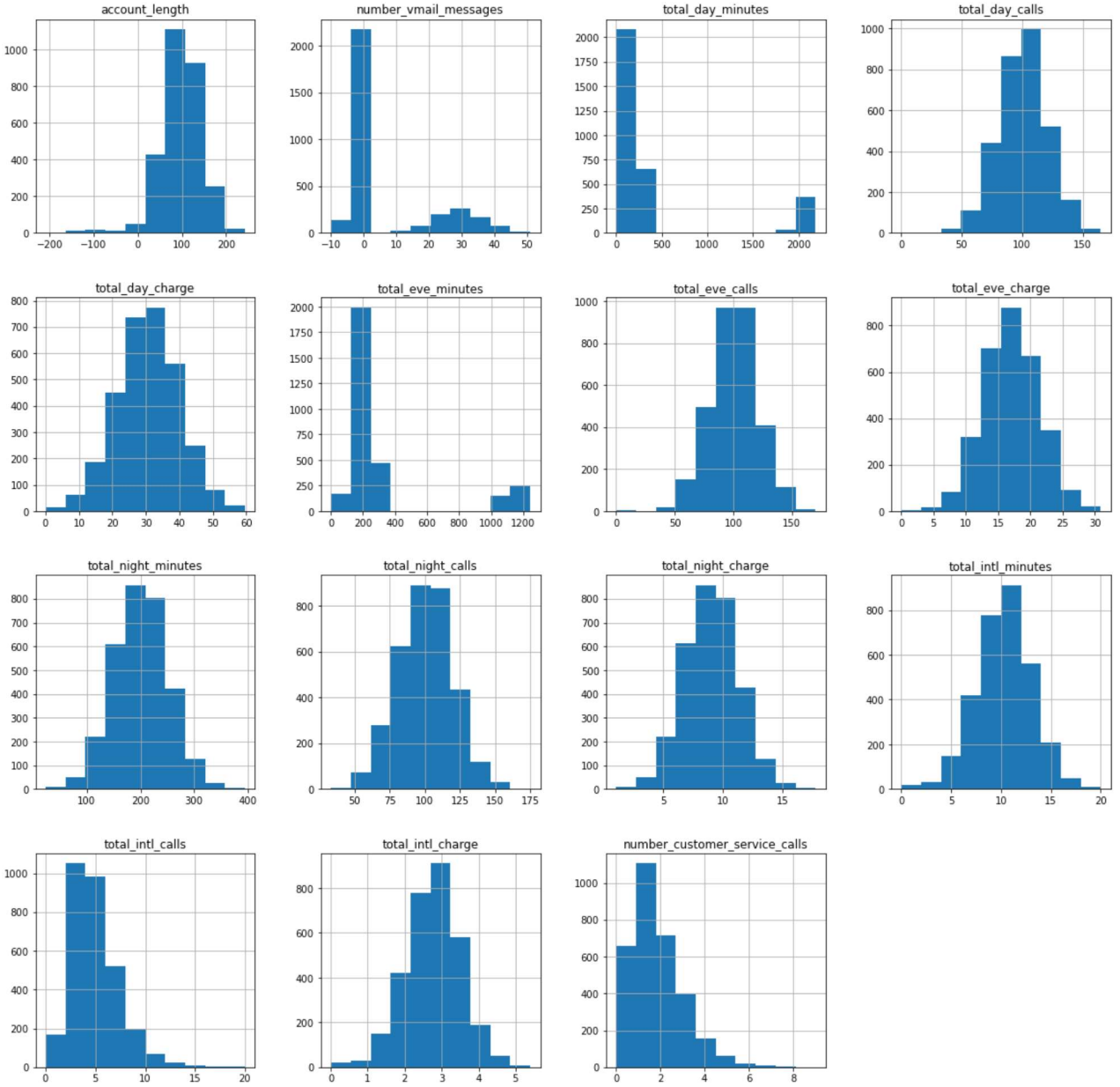
	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages	t
6	NC	135.0	area_code_415	no	no	0.0	
7	PA	28.0	area_code_415	no	no	0.0	
8	IA	86.0	area_code_408	no	no	0.0	
9	IN	65.0	area_code_415	no	no	0.0	
10	DE	125.0	area_code_408	no	no	0.0	
11	MI	NaN	area_code_415	yes	no	NaN	
12	ID	NaN	area_code_415	no	yes	32.0	
13	ID	NaN	area_code_510	no	no	0.0	
14	KY	106.0	area_code_510	no	yes	9.0	

In [14]:

```
# Explore and visualize the data using hist() plot to gain insight

import matplotlib.pyplot as plt
from matplotlib import pyplot

data.hist()
plt.gcf().set_size_inches(20, 20)
pyplot.show()
```



In [ ]:

In [15]:
```
#Question Three: Identify missing values and implement your preferred treatment for the

data.isna().sum() #Missing Values
```

Out[15]:
```
state                           0
account_length                501
area_code                       0
international_plan              0
voice_mail_plan                 0
number_vmail_messages         200
total_day_minutes             200
total_day_calls               200
total_day_charge              200
total_eve_minutes             301
total_eve_calls               200
total_eve_charge              200
```

```
total_night_minutes              200
total_night_calls                  0
total_night_charge               200
total_intl_minutes               200
total_intl_calls                 301
total_intl_charge                200
number_customer_service_calls    200
churn                              0
dtype: int64
```

In [16]:
```
from numpy import nan

df = data.replace(0, nan)
df.isna().sum()
```

Out[16]:
```
state                              0
account_length                   501
area_code                          0
international_plan                 0
voice_mail_plan                    0
number_vmail_messages           2309
total_day_minutes                202
total_day_calls                  202
total_day_charge                 202
total_eve_minutes                302
total_eve_calls                  201
total_eve_charge                 201
total_night_minutes              200
total_night_calls                  0
total_night_charge               200
total_intl_minutes               218
total_intl_calls                 318
total_intl_charge                218
number_customer_service_calls    860
churn                              0
dtype: int64
```

In [22]:
```
#Treating missing value (fill missing value with mean)

dfmean = data.fillna(data.mean())
print(dfmean.isna().sum())
```

```
state                            0
account_length                   0
area_code                        0
international_plan               0
voice_mail_plan                  0
number_vmail_messages            0
total_day_minutes                0
total_day_calls                  0
total_day_charge                 0
total_eve_minutes                0
total_eve_calls                  0
total_eve_charge                 0
total_night_minutes              0
total_night_calls                0
total_night_charge               0
total_intl_minutes               0
total_intl_calls                 0
total_intl_charge                0
number_customer_service_calls    0
```

```
churn                           0
dtype: int64
```

In [25]:
```
dataset = dfmean
```

In [ ]:
```

```

In [41]:
```
#Question 4: Conduct data transformation on the data where required (Rescaling, Standar

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

df = dataset
df['state'] =le.fit_transform(df['state'])
df['area_code'] =le.fit_transform(df['area_code'])
df['international_plan'] =le.fit_transform(df['international_plan'])
df['voice_mail_plan'] =le.fit_transform(df['voice_mail_plan'])
df['churn'] =le.fit_transform(df['churn'])
```

In [31]:
```
dataset.isna().sum()
```

Out[31]:
```
state                           0
account_length                  0
area_code                       0
international_plan              0
voice_mail_plan                 0
number_vmail_messages           0
total_day_minutes               0
total_day_calls                 0
total_day_charge                0
total_eve_minutes               0
total_eve_calls                 0
total_eve_charge                0
total_night_minutes             0
total_night_calls               0
total_night_charge              0
total_intl_minutes              0
total_intl_calls                0
total_intl_charge               0
number_customer_service_calls   0
churn                           0
dtype: int64
```

In [32]:
```
# data transformation usingscikit learn MinMaxScaler to Rescale

from numpy import set_printoptions
from sklearn.preprocessing import MinMaxScaler

filename = 'churn_Train.csv'
rescale_data = dataset

x = rescale_data.iloc[:, 0:19]
y = rescale_data.iloc[:, 19]

model = MinMaxScaler(feature_range= (0, 1))
rescaledx = model.fit_transform(x)
```

```
set_printoptions(precision= 3)
print(rescaledx[0:5, :])
```

```
[[0.66  0.739 1.    0.    0.    0.164 0.921 0.6   0.481 0.89  0.629 0.483
  0.592 0.415 0.592 0.545 0.35  0.544 0.    ]
 [0.22  0.701 0.5   0.    0.    0.164 0.133 0.6   0.831 0.178 0.547 0.608
  0.554 0.542 0.554 0.7   0.45  0.7   0.222]
 [0.14  0.644 0.5   0.    0.    0.164 0.137 0.661 0.856 0.145 0.588 0.498
  0.664 0.282 0.664 0.585 0.2   0.585 0.    ]
 [0.22  0.678 0.    0.    1.    0.656 0.05  0.43  0.314 0.147 0.635 0.501
  0.432 0.387 0.432 0.55  0.4   0.55  0.222]
 [0.7   0.646 0.5   0.    0.    0.164 0.154 0.727 0.962 0.183 0.682 0.625
  0.352 0.57  0.352 0.79  0.35  0.791 0.    ]]
```

In [33]:

```
rescaledx_data = pd.DataFrame(rescaledx)

rescaledx_data.columns = ['state', 'account_length', 'area_code', 'international_plan',
       'voice_mail_plan', 'number_vmail_messages', 'total_day_minutes',
       'total_day_calls', 'total_day_charge', 'total_eve_minutes',
       'total_eve_calls', 'total_eve_charge', 'total_night_minutes',
       'total_night_calls', 'total_night_charge', 'total_intl_minutes',
       'total_intl_calls', 'total_intl_charge',
       'number_customer_service_calls',]

rescaledx_data['churn'] = rescale_data['churn']
rescaledx_data
rescaledx_data.describe()
```

Out[33]:

	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_mes
count	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.0(
mean	0.521188	0.677702	0.500300	0.096910	0.276628	0.2
std	0.296498	0.097630	0.354824	0.295879	0.447398	0.2
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.0(
25%	0.280000	0.632743	0.000000	0.000000	0.000000	0.1(
50%	0.520000	0.677702	0.500000	0.000000	0.000000	0.1(
75%	0.780000	0.732301	1.000000	0.000000	1.000000	0.2
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.0(

In [34]:

```
rescaledx_data.isna().sum()
```

Out[34]:

```
state                          0
account_length                 0
area_code                      0
international_plan             0
voice_mail_plan                0
number_vmail_messages          0
total_day_minutes              0
total_day_calls                0
total_day_charge               0
total_eve_minutes              0
```

```
total_eve_calls                 0
total_eve_charge                0
total_night_minutes             0
total_night_calls               0
total_night_charge              0
total_intl_minutes              0
total_intl_calls                0
total_intl_charge               0
number_customer_service_calls   0
churn                           0
dtype: int64
```

In [39]:

```
# Converting back to categorical data

df = rescaledx_data
data['state'] = df['state'].astype('category')
data['area_code'] = df['area_code'].astype('category')
data['international_plan'] = df['international_plan'].astype('category')
data['voice_mail_plan'] = df['voice_mail_plan'].astype('category')
data['churn'] = df['churn'].astype('category')

data
```

Out[39]:

	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages	to
0	0.66	125.0	1.0	0.0	0.0	0.0	
1	0.22	108.0	0.5	0.0	0.0	0.0	
2	0.14	82.0	0.5	0.0	0.0	0.0	
3	0.22	NaN	0.0	0.0	1.0	30.0	
4	0.70	83.0	0.5	0.0	0.0	0.0	
...	...	...	...	...	...	...	
3328	0.70	144.0	0.5	0.0	1.0	18.0	
3329	0.36	69.0	0.5	0.0	1.0	37.0	
3330	0.82	NaN	0.5	0.0	0.0	0.0	
3331	0.68	39.0	0.0	0.0	0.0	0.0	
3332	0.96	41.0	0.0	0.0	0.0	0.0	

3333 rows × 20 columns

In [34]:

```
# Standardize data: to standardize a differing mean and standard deviation

from sklearn.preprocessing import StandardScaler
from numpy import set_printoptions

filename = 'Churn_Train.csv'
df_data = rescaledx_data

x = df_data.iloc[:, : 19]
y = df_data.iloc[:, 19]
```

```
scaler = StandardScaler().fit(x)
standardx = scaler.transform(x)

set_printoptions()
print(standardx[0:5, :])
```

```
[[ 4.682e-01  5.783e-01  1.409e+00 -3.276e-01 -6.184e-01 -5.331e-01
   2.546e+00 -6.646e-02 -2.122e-01  2.447e+00  3.454e-01 -5.015e-01
   8.350e-01 -4.144e-01  8.356e-01  2.400e-01  1.030e+00  2.354e-01
  -1.187e+00]
 [-1.016e+00  2.231e-01 -8.457e-04 -3.276e-01 -6.184e-01 -5.331e-01
  -2.034e-01 -6.646e-02  2.042e+00 -3.223e-01 -3.583e-01  3.972e-01
   5.553e-01  5.056e-01  5.535e-01  1.345e+00  1.845e+00  1.345e+00
   3.334e-01]
 [-1.286e+00 -3.201e-01 -8.457e-04 -3.276e-01 -6.184e-01 -5.331e-01
  -1.895e-01  4.326e-01  2.202e+00 -4.476e-01 -6.465e-03 -3.944e-01
   1.366e+00 -1.385e+00  1.364e+00  5.252e-01 -1.916e-01  5.259e-01
  -1.187e+00]
 [-1.016e+00        nan -1.410e+00 -3.276e-01  1.617e+00  1.648e+00
  -4.929e-01 -1.464e+00 -1.281e+00 -4.432e-01  3.956e-01 -3.688e-01
  -3.451e-01 -6.188e-01 -3.455e-01  2.757e-01  1.438e+00  2.750e-01
   3.334e-01]
 [ 6.032e-01 -2.992e-01 -8.457e-04 -3.276e-01 -6.184e-01 -5.331e-01
  -1.302e-01  9.816e-01  2.882e+00 -3.026e-01  7.977e-01  5.230e-01
  -9.380e-01  7.100e-01 -9.361e-01  1.987e+00  1.030e+00  1.992e+00
  -1.187e+00]]
```

In [56]:

```
rescaledx_data
```

Out[56]:

	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages	to
0	0.66	0.738938	1.0	0.0	0.0	0.163934	
1	0.22	0.701327	0.5	0.0	0.0	0.163934	
2	0.14	0.643805	0.5	0.0	0.0	0.163934	
3	0.22	NaN	0.0	0.0	1.0	0.655738	
4	0.70	0.646018	0.5	0.0	0.0	0.163934	
...	...	...	...	...	...	...	
3328	0.70	0.780973	0.5	0.0	1.0	0.459016	
3329	0.36	0.615044	0.5	0.0	1.0	0.770492	
3330	0.82	NaN	0.5	0.0	0.0	0.163934	
3331	0.68	0.548673	0.0	0.0	0.0	0.163934	
3332	0.96	0.553097	0.0	0.0	0.0	0.163934	

3333 rows × 20 columns

In [74]:

```
rescaledx_data.isna().sum()
```

Out[74]:

```
state                     0
account_length          501
area_code                 0
```

```
international_plan                0
voice_mail_plan                   0
number_vmail_messages           200
total_day_minutes               200
total_day_calls                 200
total_day_charge                200
total_eve_minutes               301
total_eve_calls                 200
total_eve_charge                200
total_night_minutes             200
total_night_calls                 0
total_night_charge              200
total_intl_minutes              200
total_intl_calls                301
total_intl_charge               200
number_customer_service_calls   200
churn                             0
dtype: int64
```

In [36]:

```
# Normalizing - rescaling each observation (record) to have a length of 1

from sklearn.preprocessing import Normalizer

filename = 'Churn_Train.csv'
dframe = rescaledx_data

x = dframe.iloc[:, : 19]
y = dframe.iloc[:, 19]

scaler = Normalizer().fit(x)
normalizedx = scaler.transform(x)


set_printoptions(precision = 5)
print(normalizedx[0:5, :])
```

```
[[0.25967 0.29072 0.39344 0.      0.      0.0645  0.36252 0.23606 0.18907
  0.35024 0.24763 0.19004 0.23291 0.16347 0.23305 0.21442 0.1377  0.2142
  0.     ]
 [0.1012  0.3226  0.23    0.      0.      0.07541 0.06139 0.27599 0.38232
  0.08174 0.25164 0.27963 0.25486 0.24943 0.25488 0.32199 0.207   0.32199
  0.10222]
 [0.06831 0.31414 0.24397 0.      0.      0.07999 0.06706 0.32234 0.41767
  0.07098 0.28703 0.24295 0.32403 0.13745 0.32404 0.28545 0.09759 0.28554
  0.     ]
 [0.10686 0.32918 0.      0.      0.48573 0.31851 0.02452 0.20901 0.15271
  0.07121 0.30858 0.24357 0.20981 0.18813 0.20991 0.26715 0.19429 0.26715
  0.10794]
 [0.29857 0.27554 0.21326 0.      0.      0.06992 0.06586 0.3102  0.41022
  0.07795 0.29104 0.26673 0.14994 0.2433  0.15016 0.33695 0.14928 0.33727
  0.     ]]
```

In [58]:

```
Dataset = dfmean
```

In [38]:

```
normalizedx_data = pd.DataFrame(normalizedx)

normalizedx_data.columns = ['state', 'account_length', 'area_code', 'international_plan
       'voice_mail_plan', 'number_vmail_messages', 'total_day_minutes',
       'total_day_calls', 'total_day_charge', 'total_eve_minutes',
       'total_eve_calls', 'total_eve_charge', 'total_night_minutes',
```

```
        'total_night_calls', 'total_night_charge', 'total_intl_minutes',
        'total_intl_calls', 'total_intl_charge',
        'number_customer_service_calls',]

normalizedx_data['churn'] = dframe['churn']
normalizedx_data
normalizedx_data.describe()
```

Out[38]:

	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_mes
count	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.0(
mean	0.240532	0.318442	0.228406	0.041068	0.117149	0.1
std	0.134670	0.056607	0.158561	0.126040	0.190464	0.0
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.0(
25%	0.129283	0.284903	0.000000	0.000000	0.000000	0.0
50%	0.243554	0.317838	0.236836	0.000000	0.000000	0.0
75%	0.349074	0.353620	0.345525	0.000000	0.372822	0.1
max	0.547723	0.512247	0.576140	0.566120	0.546589	0.4

In [ ]:

In [45]:

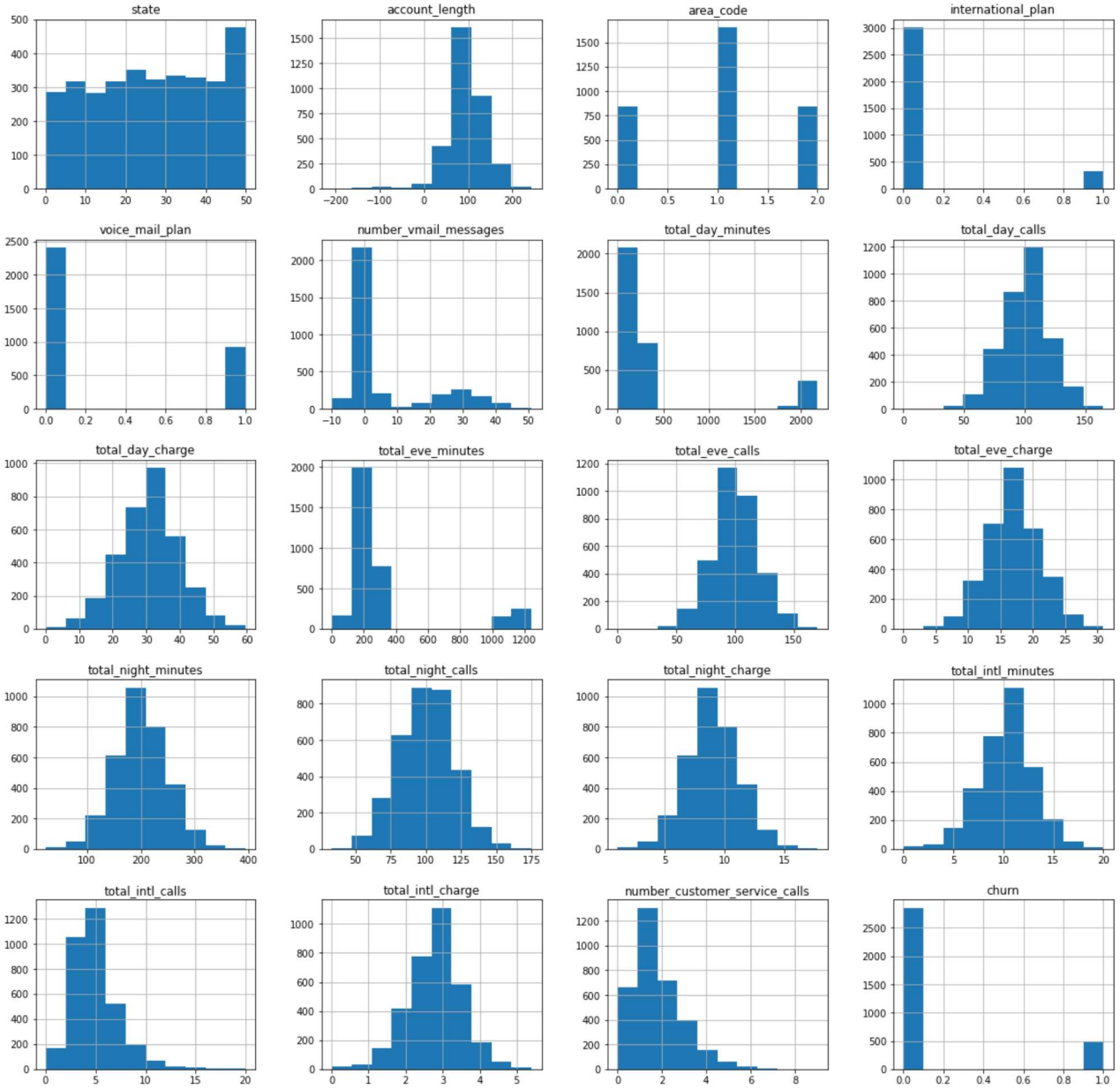
```
# Question 5: Include charts in your report

import matplotlib.pyplot as plt

dataset.hist(),'Churn_Train.csv'

plt.gcf().set_size_inches(20, 20)

plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: