10

April 30, 2023

0.1 Group Name: Go Bear!

0.2 Name: Xiaoke Song

0.3 Email: xiaokesong57@gmail.com

0.4 Country: born in China, college in the US

0.5 College: UC Berkeley

0.6 Specialization: Data Science

[1]:
```
import pandas as pd
```

[2]:
```
df = pd.read_csv('cust_seg.csv')
df.head(5)
```

```
/var/folders/_0/nmpfpzw134n12j0c0z6jtrw80000gn/T/ipykernel_69684/3036801543.py:1
: DtypeWarning: Columns (16) have mixed types. Specify dtype option on import or
set low_memory=False.
  df = pd.read_csv('cust_seg.csv')
```

[2]:

	Unnamed: 0	fecha_dato	ncodpers	ind_empleado	pais_residencia	sexo	age	\
0	0	2015-01-28	1375586	N	ES	H	35	
1	1	2015-01-28	1050611	N	ES	V	23	
2	2	2015-01-28	1050612	N	ES	V	23	
3	3	2015-01-28	1050613	N	ES	H	22	
4	4	2015-01-28	1050614	N	ES	V	23	

	fecha_alta	ind_nuevo	antiguedad	…	ind_hip_fin_ult1	ind_plan_fin_ult1	\
0	2015-01-12	0.0	6	…	0	0	
1	2012-08-10	0.0	35	…	0	0	
2	2012-08-10	0.0	35	…	0	0	
3	2012-08-10	0.0	35	…	0	0	
4	2012-08-10	0.0	35	…	0	0	

	ind_pres_fin_ult1	ind_reca_fin_ult1	ind_tjcr_fin_ult1	ind_valo_fin_ult1	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	

```
3                   0                   0                   0                   0
4                   0                   0                   0                   0

  ind_viv_fin_ult1 ind_nomina_ult1 ind_nom_pens_ult1  ind_recibo_ult1
0                0            0.0               0.0                0
1                0            0.0               0.0                0
2                0            0.0               0.0                0
3                0            0.0               0.0                0
4                0            0.0               0.0                0

[5 rows x 48 columns]
```

Problem description: Customer Segmation ____ "XYZ bank wants to roll out Christmas offers to their customers. But Bank does not want to roll out same offer to all customers instead they want to roll out personalized offer to particular set of customers. If they manually start understanding the category of customer then this will be not efficient and also they will not be able to uncover the hidden pattern in the data (pattern which group certain kind of customer in one category). Bank approached ABC analytics company to solve their problem. Bank also shared information with ABC analytics that they don't want more than 5 group as this will be inefficient for their campaign."

0.7 EDA:

[5]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Check the number of rows and columns
print(df.shape)
```

```
(1000000, 48)
```

[7]:
```
# Check the data types of the columns
print(df.dtypes)
```

```
Unnamed: 0                int64
fecha_dato               object
ncodpers                  int64
ind_empleado             object
pais_residencia          object
sexo                     object
age                      object
fecha_alta               object
ind_nuevo               float64
antiguedad               object
indrel                  float64
ult_fec_cli_1t           object
indrel_1mes             float64
```

```
tiprel_1mes               object
indresi                   object
indext                    object
conyuemp                  object
canal_entrada             object
indfall                   object
tipodom                  float64
cod_prov                 float64
nomprov                   object
ind_actividad_cliente    float64
renta                    float64
ind_ahor_fin_ult1          int64
ind_aval_fin_ult1          int64
ind_cco_fin_ult1           int64
ind_cder_fin_ult1          int64
ind_cno_fin_ult1           int64
ind_ctju_fin_ult1          int64
ind_ctma_fin_ult1          int64
ind_ctop_fin_ult1          int64
ind_ctpp_fin_ult1          int64
ind_deco_fin_ult1          int64
ind_deme_fin_ult1          int64
ind_dela_fin_ult1          int64
ind_ecue_fin_ult1          int64
ind_fond_fin_ult1          int64
ind_hip_fin_ult1           int64
ind_plan_fin_ult1          int64
ind_pres_fin_ult1          int64
ind_reca_fin_ult1          int64
ind_tjcr_fin_ult1          int64
ind_valo_fin_ult1          int64
ind_viv_fin_ult1           int64
ind_nomina_ult1          float64
ind_nom_pens_ult1        float64
ind_recibo_ult1            int64
dtype: object
```

[8]:
```
# Check for missing values
print(df.isnull().sum())
```

```
Unnamed: 0                   0
fecha_dato                   0
ncodpers                     0
ind_empleado             10782
pais_residencia          10782
sexo                     10786
age                          0
fecha_alta               10782
```

```
ind_nuevo                 10782
antiguedad                    0
indrel                    10782
ult_fec_cli_1t           998899
indrel_1mes               10782
tiprel_1mes               10782
indresi                   10782
indext                    10782
conyuemp                 999822
canal_entrada             10861
indfall                   10782
tipodom                   10782
cod_prov                  17734
nomprov                   17734
ind_actividad_cliente     10782
renta                    175183
ind_ahor_fin_ult1             0
ind_aval_fin_ult1             0
ind_cco_fin_ult1              0
ind_cder_fin_ult1             0
ind_cno_fin_ult1              0
ind_ctju_fin_ult1             0
ind_ctma_fin_ult1             0
ind_ctop_fin_ult1             0
ind_ctpp_fin_ult1             0
ind_deco_fin_ult1             0
ind_deme_fin_ult1             0
ind_dela_fin_ult1             0
ind_ecue_fin_ult1             0
ind_fond_fin_ult1             0
ind_hip_fin_ult1              0
ind_plan_fin_ult1             0
ind_pres_fin_ult1             0
ind_reca_fin_ult1             0
ind_tjcr_fin_ult1             0
ind_valo_fin_ult1             0
ind_viv_fin_ult1              0
ind_nomina_ult1            5402
ind_nom_pens_ult1          5402
ind_recibo_ult1               0
dtype: int64
```

[9]:
```
# Check for duplicated rows
print(df.duplicated().sum())
```

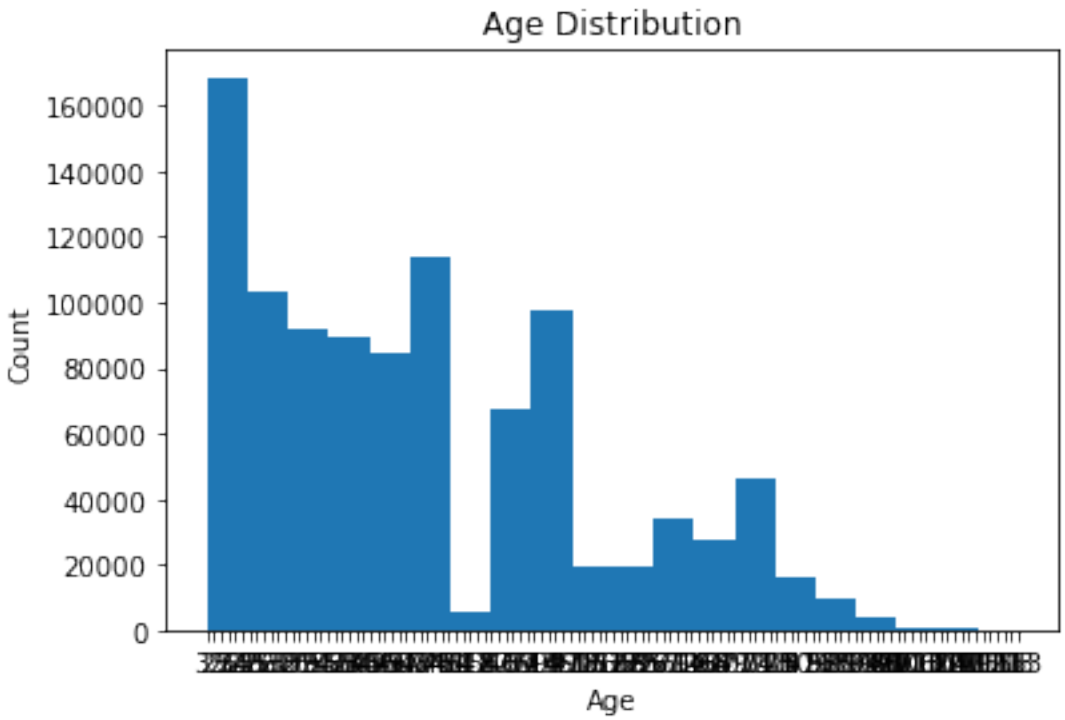
```
0
```

[10]:
```
# Check the summary statistics of the numerical columns
print(df.describe())
```

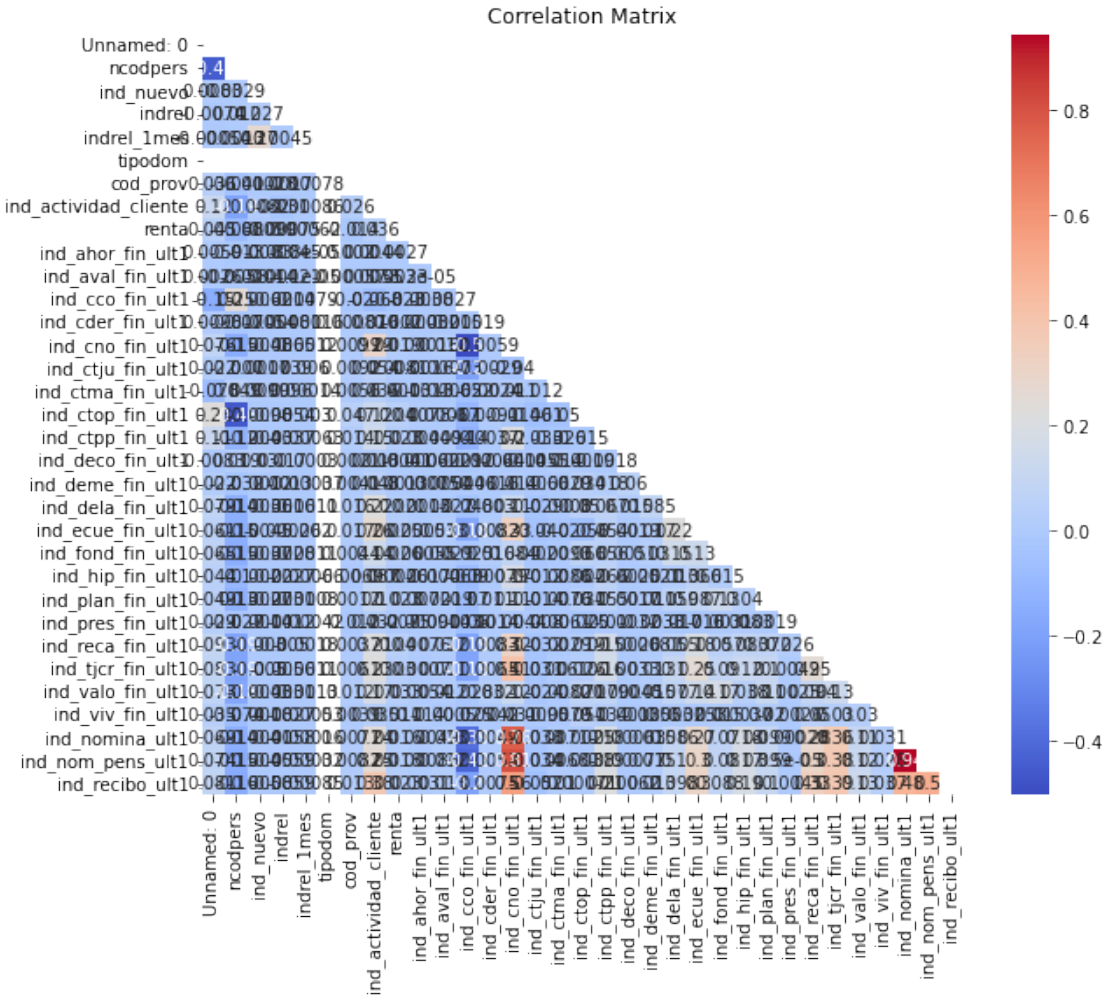
```
          Unnamed: 0      ncodpers      ind_nuevo         indrel  \
count  1000000.000000  1.000000e+06  989218.000000  989218.000000
mean    499999.500000  6.905967e+05       0.000489       1.109074
std     288675.278933  4.044084e+05       0.022114       3.267624
min          0.000000  1.588900e+04       0.000000       1.000000
25%     249999.750000  3.364110e+05       0.000000       1.000000
50%     499999.500000  6.644760e+05       0.000000       1.000000
75%     749999.250000  1.074511e+06       0.000000       1.000000
max     999999.000000  1.379131e+06       1.000000      99.000000

         indrel_1mes   tipodom       cod_prov  ind_actividad_cliente  \
count  989218.000000  989218.0  982266.000000          989218.000000
mean        1.000085       1.0      26.852131               0.564971
std         0.012954       0.0      12.422924               0.495761
min         1.000000       1.0       1.000000               0.000000
25%         1.000000       1.0      18.000000               0.000000
50%         1.000000       1.0      28.000000               1.000000
75%         1.000000       1.0      33.000000               1.000000
max         3.000000       1.0      52.000000               1.000000

              renta  ind_ahor_fin_ult1  …  ind_hip_fin_ult1  \
count  8.248170e+05     1000000.000000  …    1000000.000000
mean   1.396462e+05           0.000177  …          0.009982
std    2.389858e+05           0.013303  …          0.099410
min    1.202730e+03           0.000000  …          0.000000
25%    7.157184e+04           0.000000  …          0.000000
50%    1.066519e+05           0.000000  …          0.000000
75%    1.634325e+05           0.000000  …          0.000000
max    2.889440e+07           1.000000  …          1.000000

       ind_plan_fin_ult1  ind_pres_fin_ult1  ind_reca_fin_ult1  \
count     1000000.000000     1000000.000000     1000000.000000
mean            0.014553           0.004661           0.072581
std             0.119755           0.068112           0.259448
min             0.000000           0.000000           0.000000
25%             0.000000           0.000000           0.000000
50%             0.000000           0.000000           0.000000
75%             0.000000           0.000000           0.000000
max             1.000000           1.000000           1.000000

       ind_tjcr_fin_ult1  ind_valo_fin_ult1  ind_viv_fin_ult1  \
count     1000000.000000     1000000.000000    1000000.000000
mean            0.066084           0.039378          0.006442
std             0.248429           0.194493          0.080003
```

```
min             0.000000          0.000000          0.000000
25%             0.000000          0.000000          0.000000
50%             0.000000          0.000000          0.000000
75%             0.000000          0.000000          0.000000
max             1.000000          1.000000          1.000000

       ind_nomina_ult1  ind_nom_pens_ult1  ind_recibo_ult1
count    994598.000000      994598.000000   1000000.000000
mean          0.071629           0.079543         0.166275
std           0.257873           0.270584         0.372327
min           0.000000           0.000000         0.000000
25%           0.000000           0.000000         0.000000
50%           0.000000           0.000000         0.000000
75%           0.000000           0.000000         0.000000
max           1.000000           1.000000         1.000000

[8 rows x 33 columns]
```

[16]:
```
# Create a histogram of the 'age' column
plt.hist(df['age'], bins=20)
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Age Distribution')
plt.show()
```



[17]:
```
# Compute the correlation matrix
corr = df.corr()

# Create a heatmap using seaborn
plt.figure(figsize=(10,8))
sns.heatmap(corr, cmap='coolwarm', annot=True, square=True, mask=np.triu(corr))
plt.title('Correlation Matrix')
plt.show()
```



[13]:
```
# Check the frequency distribution of categorical columns
print(df['sexo'].value_counts())
```

```
V    562000
H    427214
Name: sexo, dtype: int64
```

0.8 Final Recommendation:

Based on the EDA performed, we can make the following recommendations for the company: 1. The dataset contains missing values, outliers and categorical variables that need to be preprocessed before analysis. The missing values can be filled with appropriate values such as mean, median or mode depending on the distribution of the data. Outliers can be removed or handled using appropriate techniques such as winsorization or transformations. Categorical variables can be encoded using one-hot encoding or label encoding techniques. 2. The distribution of the target variable (y) indicates that the dataset is imbalanced, with a higher proportion of negative outcomes than positive outcomes. This could potentially impact the model's performance and should be considered during model training. 3. The correlation heatmap shows that some of the numerical variables are strongly correlated with each other, which may lead to multicollinearity issues during model training. Feature selection or dimensionality reduction techniques can be used to reduce the number of features and improve model performance. 4. The box plot of age distribution by gender shows that there are some outliers in the data. These outliers could potentially impact the model's performance and should be handled accordingly. 5. The histogram of numerical variables shows that most of the variables have a skewed distribution. Transformations such as log, square root or box-cox transformations can be used to reduce the skewness and improve the model's performance. 6. The scatter plot matrix shows that there is no strong correlation between the numerical variables and the target variable (y). This indicates that a simple linear regression model may not perform well on this dataset, and more complex models such as decision trees, random forests or neural networks may need to be explored. 7. Finally, it is recommended to use cross-validation techniques such as K-fold or stratified K-fold to evaluate the performance of the models on the imbalanced dataset. This will ensure that the model is able to generalize well on new unseen data.

[ ]: