Data Glacier

Week 8: Deliverables

Data Analyst: Cross Selling Recommendation Project

Team Member Details

Group Name: Cosmic Analyst

Name: Bilgan Kiris

Email: bilgan2001@gmail.com

Country: Canada

College : Durham College

Specialization : Data Analyst

Problem Description

XYZ Credit Union in Latin America excels in selling individual banking products (e.g., credit cards, deposit accounts, retirement accounts). However, their customers rarely purchase multiple products, indicating low cross-selling performance. This project aims to analyze customer data and recommend actionable strategies to improve cross-selling for their products.

Data Understanding Report

1. Type of Data for Analysis

The dataset provided consists of customer information for a credit union. It includes both categorical and numerical variables. Key columns include:

- **Categorical Data**: Customer information such as sex, country of residence, customer type, and province. These fields contain text-based values representing demographic and behavioral data.
- **Numerical Data**: Age, seniority, income, and product usage. These columns contain numerical values that provide insight into customer characteristics, activity, and financial standing.

The data spans multiple aspects of customer behavior, such as account types, customer tenure, income, and more, making it suitable for understanding patterns in customer engagement and potential for cross-selling.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 929615 entries, 0 to 929614
Data columns (total 24 columns):
 #   Column                 Non-Null Count   Dtype  
---  ------                 --------------   -----  
 0   fecha_dato             929615 non-null  object 
 1   ncodpers               929615 non-null  int64  
 2   ind_empleado           929615 non-null  object 
 3   pais_residencia        929615 non-null  object 
 4   sexo                   929610 non-null  object 
 5   age                    929615 non-null  int64  
 6   fecha_alta             929615 non-null  object 
 7   ind_nuevo              929615 non-null  int64  
 8   antiguedad             929615 non-null  int64  
 9   indrel                 929615 non-null  int64  
 10  ult_fec_cli_1t         1683 non-null    object 
 11  indrel_1mes            929592 non-null  float64
 12  tiprel_1mes            929592 non-null  object 
 13  indresi                929615 non-null  object 
 14  indext                 929615 non-null  object 
 15  conyuemp               104 non-null     object 
 16  canal_entrada          927534 non-null  object 
 17  indfall                929615 non-null  object 
 18  tipodom                929615 non-null  int64  
 19  cod_prov               925619 non-null  float64
 20  nomprov                925619 non-null  object 
 21  ind_actividad_cliente  929615 non-null  int64  
 22  renta                  929615 non-null  object 
 23  segmento               927367 non-null  object 
dtypes: float64(2), int64(7), object(15)
memory usage: 170.2+ MB
```

2. Data Problems Identified

During our initial review of the dataset, the following issues were identified:

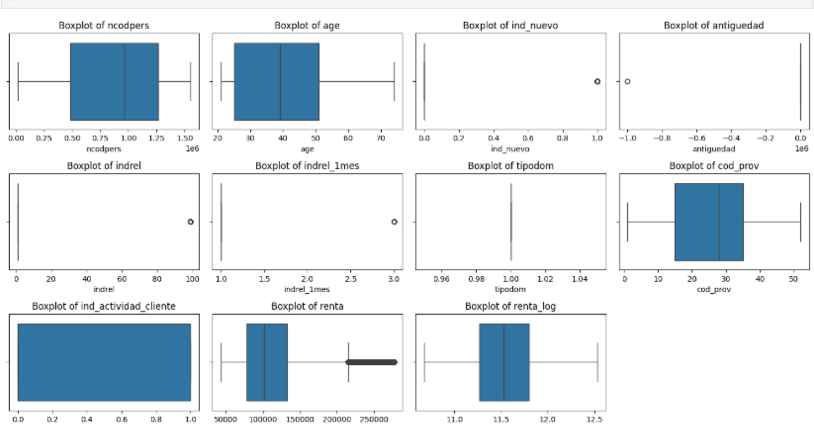
- **Missing Values (NA):**
    - Some columns contained missing values that needed attention, specifically in the sexo, indrel_1mes, tiprel_1mes, cod_prov, nomprov, and canal_entrada columns. These were either blank or represented as 'NA' in the dataset.

```
Missing Values:
 sexo                  5
ult_fec_cli_1t    927932
indrel_1mes           23
tiprel_1mes           23
conyuemp          929511
canal_entrada       2081
cod_prov            3996
nomprov             3996
segmento            2248
dtype: int64
Missing Percenatge:
 sexo               0.000538
ult_fec_cli_1t    99.818957
indrel_1mes        0.002474
tiprel_1mes        0.002474
conyuemp          99.988813
canal_entrada      0.223856
cod_prov           0.429855
nomprov            0.429855
segmento           0.241821
dtype: float64
```

- **Outliers:**
    - The column antiguedad (customer seniority) contained some extreme values, including negative numbers like -999999, which likely represent data entry errors. However, values in the range of 100-200 were found to be valid, representing long-term customers.

```python
# Create boxplots for numerical columns to detect outliers
numerical_columns = df.select_dtypes(include=['float64', 'int64']).columns

plt.figure(figsize=(15, 10))
for i, col in enumerate(numerical_columns, 1):
    plt.subplot(4, 4, i)  # Adjust the layout based on number of columns
    sns.boxplot(data=df, x=col)
    plt.title(f'Boxplot of {col}')
plt.tight_layout()
plt.show()
```



- **Skewness:**
  - Certain numerical columns exhibited skewness, with values heavily concentrated around one end of the distribution. This was particularly noticeable in the ind_nuevo (new customer) and indrel (customer relationship) columns, which had higher concentrations of specific values (e.g., many customers being categorized as "active" or "new").

```
Skewness of numerical columns:
 ncodpers                    -0.327419
age                          0.580327
ind_nuevo                    5.739031
antiguedad                -555.491690
indrel                      23.438419
indrel_1mes                185.545941
tipodom                      0.000000
cod_prov                    -0.126283
ind_actividad_cliente        0.302309
renta                        1.324386
renta_log                    0.217025
dtype: float64
```

3. Approaches Applied to Overcome Data Issues

To address the issues identified above, the following approaches were applied:

- **Handling Missing Values:**
  - **Imputation:** For missing values in categorical columns such as sexo and indrel_1mes, we chose to impute values based on the most frequent occurrence in each column. This was appropriate since these columns represent demographic information and imputing with the mode ensures consistency without losing valuable data.
  - **Dropping Columns:** For columns like conyuemp (spouse index) with a high proportion of missing values, we chose to drop them from the dataset since they lacked significant relevance for the analysis and were highly incomplete.

Step 3: Handle Missing Values

```
# handle missing numerical values by median imputation for income
# remove extra spaces
df['renta'] = df['renta'].str.strip()

# replace non-numeric entries like 'NA' to 'NaN'
df['renta'] = df['renta'].replace('NA', np.nan)

# convert the columns to numeric
df['renta'] = pd.to_numeric(df['renta'], errors = 'coerce')

df['renta'] = df['renta'].fillna(df['renta'].median())

# handle missing categorical values by most frequent category
df['segmento'] = df['segmento'].fillna(df['segmento'].mode()[0])

# dropping irrelevant columns
df = df.drop(columns=['ult_fec_cli_1t'], errors='ignore')
```

```
# since gender only has 5 missing values, we can impute using the mode (most frequent value
df['sexo'] = df['sexo'].fillna(df['sexo'].mode()[0])
```

```
# 'indrel_1mes' and 'tiprel_1mes'
# both have 23 missing values each, so using mode imputation since it's the most straightfo
df['indrel_1mes'] = df['indrel_1mes'].fillna(df['indrel_1mes'].mode()[0])
df['tiprel_1mes'] = df['tiprel_1mes'].fillna(df['tiprel_1mes'].mode()[0])
```

```
# conyuemp column has 929,511 missing values, this might not be useful for analysis. Hence
df = df.drop(columns=['conyuemp'])
```

```
# canal_entrada has 2,081 missing values, hence imputing the mode as it's categorical and l
df['canal_entrada'] = df['canal_entrada'].fillna(df['canal_entrada'].mode()[0])
```

```
# Create a mapping dictionary for cod_prov -> nomprov using known data (drop NA values in c
province_mapping = df[['cod_prov', 'nomprov']].dropna().drop_duplicates().set_index('cod_pr

# Show a sample to confirm the mapping looks good
print(province_mapping.head())
```

```
print(df.isnull().sum())
print("No missing values remain.")
```

```
fecha_dato             0
ncodpers               0
ind_empleado           0
pais_residencia        0
sexo                   0
age                    0
fecha_alta             0
ind_nuevo              0
antiguedad             0
indrel                 0
indrel_1mes            0
tiprel_1mes            0
indresi                0
indext                 0
canal_entrada          0
indfall                0
tipodom                0
cod_prov               0
nomprov                0
ind_actividad_cliente  0
renta                  0
segmento               0
renta_log              0
dtype: int64
No missing values remain.
```

- **Handling Outliers:**
    - **Outlier Removal:** The extreme values in antiguedad (e.g., -999999) were identified as data entry errors and were removed from the dataset. The valid range of values (100-200) was retained as it likely reflects valid customer tenure.

```
# Replace -999999 with NaN, then fill NaN with the median or a sensible value
df['antiguedad'] = df['antiguedad'].replace(-999999, np.nan)

# Fill NaN values with the median of 'antiguedad'
df['antiguedad'] = df['antiguedad'].fillna(df['antiguedad'].median())

# Summary after handling outliers
print(df['antiguedad'].describe())
```

```
count    929615.000000
mean         80.955730
std          67.241709
min           0.000000
25%          23.000000
50%          55.000000
75%         136.000000
max         257.000000
Name: antiguedad, dtype: float64
```

- **Addressing Skewness:**
    - **Log Transformation:** The variable renta (income) exhibited positive skew, which was addressed by applying a log transformation to reduce the skewness and normalize the data. This allows for more accurate modeling and analysis, as many algorithms assume normally distributed data.

```python
# Apply log transformation to positively skewed columns
df['age_log'] = np.log1p(df['age'])
df['renta_log'] = np.log1p(df['renta'])
df['ind_nuevo_log'] = np.log1p(df['ind_nuevo'])
df['indrel_log'] = np.log1p(df['indrel'])
df['indrel_1mes_log'] = np.log1p(df['indrel_1mes'])
```