Data Glacier

Week 9: 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.

1. Data Review and Preprocessing

- **Dataset Overview**: We began by reviewing the dataset, which contains multiple columns with mixed data types, including numerical and categorical values.
- **Handling Missing Values:** The dataset was checked for missing values, and since there were no missing values, no imputation was needed.
- **Outlier Detection and Handling:**
    - We performed outlier detection using Z-scores for the renta column. Although extreme outliers were identified, no rows were removed as none exceeded the threshold for removal based on Z-score criteria (-3 to 3).
- **Skewness Check:** Skewness was assessed for the numerical columns, and it was observed that certain features, like antiguedad and indrel, showed skewness, which may indicate the need for further transformation (e.g., log transformation) to make them more normally distributed.

2. Feature Engineering

- **Log Transformation:** We applied a log transformation to some columns like renta, age, indrel, and indrel_1mes to reduce skewness and make the data more normally distributed, which is beneficial for many statistical models.
- **Segment Encoding**: We cleaned the segmento column, simplifying it by removing numeric prefixes and ensuring consistent categories (PARTICULARES, TOP, and UNIVERSITARIO).

3. Data Transformation for Analysis

- **Standardization:** Z-scores were used to standardize the renta column and identify extreme outliers. However, no rows were removed since the Z-scores stayed within acceptable bounds.
- **Categorical Encoding**: Categorical columns like ind_empleado, pais_residencia, and sexo were encoded appropriately for model building, ensuring that the data is ready for further analysis.

```
import re
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
# Remove the numeric prefix using regex
df['segmento_clean'] = df['segmento'].str.extract(r'[-\s](.*)$')[0]
print(df['segmento_clean'].value_counts())
```

```
segmento_clean
- PARTICULARES     547626
- UNIVERSITARIO    346028
- TOP               35961
Name: count, dtype: int64
```

```
# Converting the cleaned categories into numerical values using label encoding
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
df['segmento_encoded'] = label_encoder.fit_transform(df['segmento_clean'])
print(label_encoder.classes_)  # To see the mapping
```

```
['- PARTICULARES' '- TOP' '- UNIVERSITARIO']
```

```
# after cleaning and encoding, analyze the distribution of customer segments
print(df['segmento_clean'].value_counts())
```

```
segmento_clean
- PARTICULARES     547626
- UNIVERSITARIO    346028
- TOP               35961
Name: count, dtype: int64
```

[36]:
```
df.groupby('segmento_clean')['renta'].mean()
df.groupby('segmento_clean')['age'].mean()
```

[36]:
```
segmento_clean
- PARTICULARES     48.690535
- TOP              54.648063
- UNIVERSITARIO    24.469627
Name: age, dtype: float64
```

```python
# Applying Standardization
# using z-scores to identify and handle extreme outliers
from scipy.stats import zscore

df['renta_zscore'] = zscore(df['renta'])
df_outliers_removed = df[(df['renta_zscore'] > -3) & (df['renta_zscore'] < 3)]
```

```python
print("Original dataset size:", len(df))
print("Filtered dataset size:", len(df_outliers_removed))
print("Number of rows removed:", len(df) - len(df_outliers_removed))
```

```
Original dataset size: 929615
Filtered dataset size: 929615
Number of rows removed: 0
```

```python
# Check the range of z-scores
print("Z-scores range:", df['renta_zscore'].min(), "to", df['renta_zscore'].max())
```

```
Z-scores range: -1.2283899617598286 to 2.7176503648128776
```