Data Glacier Data Scientist Internship

Batch:LISUM39

Week9: Deliverables

Project: Bank Customer Segmentation

Group name: Apple Analytics

Name: Madoka Fujii

Email address: mdkfji@gmail.com

Country: United States

Company: Omdena

Specialization: Data Analytics

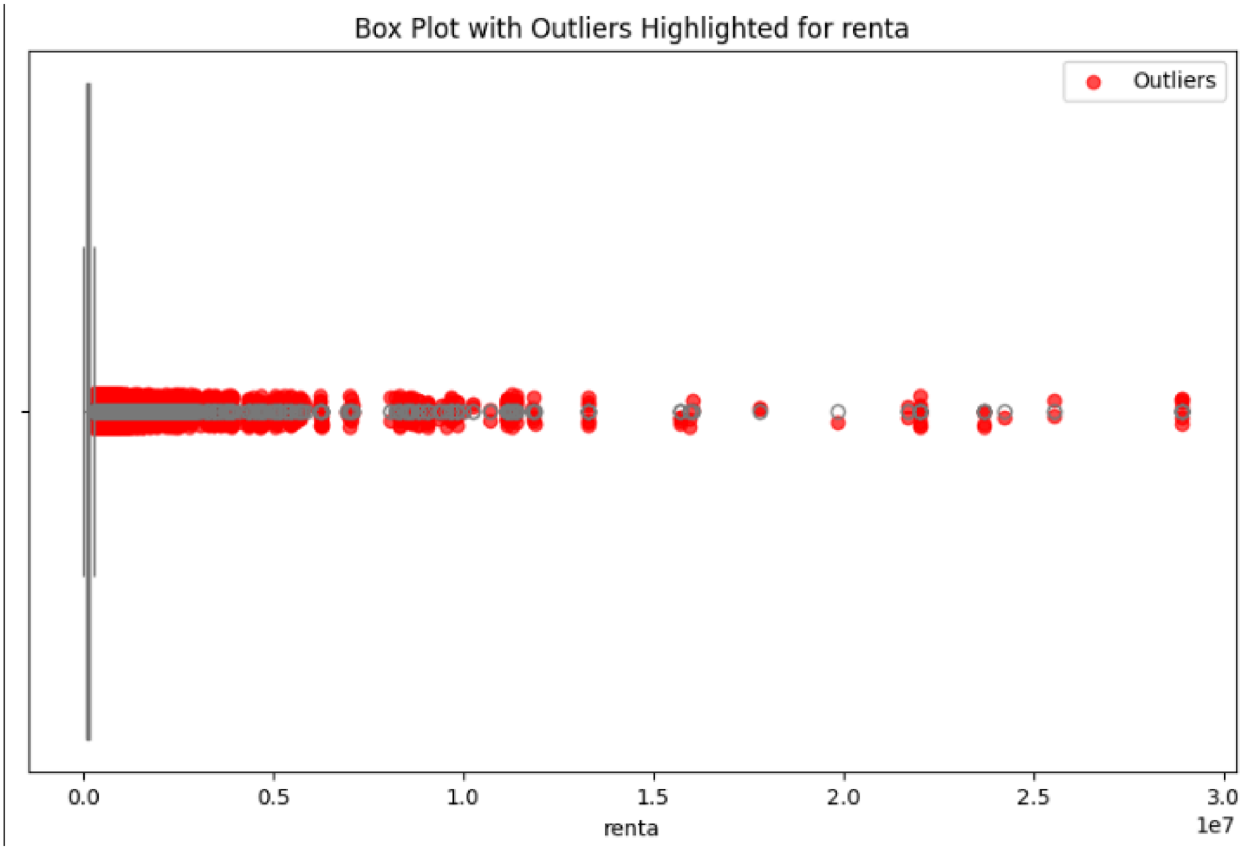
Problem Description:

XYZ Bank plans to enhance its marketing campaign as Christmas offers for its customers. However, instead of offering the same deal to all customers as generic, the bank wants to provide personalized offers to specific customer groups to fit their preferences. Identifying customer categories manually would be inefficient and fail to uncover hidden patterns in the data that could inform better segmentation. To address this, the bank has sought the assistance of ABC Analytics. Additionally, the bank has specified that customer segmentation should result in no more than 5 groups to ensure the campaign's efficiency.

Data Cleaning:

We discussed Data Understanding on week 8. Based on that, on week 9, we clean up the data for the next step.

1, Regarding the variable named 'renta', I examined the **outliers** and skewness in the visualizations below in week 8. It appears that the 'renta' variable, which represents the gross income of households, contains many outliers above the median and mean. These outliers correspond to high-income customers. Retaining these outliers may be beneficial for identifying specific trends within the high-income group, which could be useful for targeted campaigns. Therefore, while I am checking for outliers, I do not plan to omit them.




```
[29] # check for outliers
     q1 = df['renta'].quantile(0.25)
     q3 = df['renta'].quantile(0.75)

     iqr = q3-q1
     threshold = 1.5

     upper_limit = q3 + threshold * iqr
     lower_limit = q1 - threshold * iqr
     print(f'Outliers of renta \nIQR: {iqr} \nUpperlimit: {round(upper_limit,3)} \nLower limit: {round(lower_limit,3)}')
```

```
Outliers of renta
IQR: 91860.63
Upperlimit: 301223.415
Lower limit: -66219.105
```

2, Dropped 2 columns because of too many missing values as 99%

Since the columns of "conyuemp" and "ult_fec_cli_1t" have 99% of missing values, we drop them. It is not useful and appropriate to fill in the missing values with 99%.

```
[31] df.drop(columns=["conyuemp"], inplace = True, axis = 1)
     df.drop(columns=["ult_fec_cli_1t"], inplace = True, axis = 1)
```

3, Make sure dropped them

```
[32] null_agg_percent(df)
```

	Column_Name	aggregate	percent
0	renta	175183	0.175183
1	nomprov	17734	0.017734
2	cod_prov	17734	0.017734
3	canal_entrada	10861	0.010861
4	sexo	10786	0.010786
5	indrel_1mes	10782	0.010782
6	ind_actividad_cliente	10782	0.010782
7	tipodom	10782	0.010782
8	indfall	10782	0.010782
9	indext	10782	0.010782
10	indresi	10782	0.010782
11	tiprel_1mes	10782	0.010782
12	indrel	10782	0.010782
13	ind_nuevo	10782	0.010782
14	fecha_alta	10782	0.010782
15	pais_residencia	10782	0.010782
16	ind_empleado	10782	0.010782
17	ind_nomina_ult1	5402	0.005402
18	ind_nom_pens_ult1	5402	0.005402
19	ind_pres_fin_ult1	0	0.000000
20	ind_ecue_fin_ult1	0	0.000000
21	ind_fond_fin_ult1	0	0.000000
22	ind_hip_fin_ult1	0	0.000000
23	ind_plan_fin_ult1	0	0.000000

```
[33] df.shape
(1000000, 46)
```

4, Check how many percent missing values are shared in the dataset. There are 0.81% of missing values in the dataset which we can fill in because of not a lot.

```
[36] percent_of_missing_value = round(100*((missing_values)/total_counts),2)
     print(f'the percent of missing values in the dataset is {percent_of_missing_value}%')

     the percent of missing values in the dataset is 0.81%
```

5, Fill in the column 'Renta' with the **imputation method with median**.

Regarding **Renta**, the median and mean are almost same points and extremely skewed to right so applying **imputation with median**.

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

6, Regarding numerical values, apply **KNN imputation method**.

Apply **KNN Imputer** to numerical values

Nearest Neighbor Imputation is a powerful technique that relies on the similarity between data points. It can provide more accurate imputations than simpler methods like mean or median imputation, especially when relationships between features are complex.

```
[38] from sklearn.impute import KNNImputer

     # Initialize KNNImputer with k=2 neighbors
     imputer = KNNImputer(n_neighbors=2)

     # Impute missing values
     numeric_df = df.select_dtypes(include=[float, int]) #KNN impute only can apply for numerical values
     imputed_data = imputer.fit_transform(numeric_df)

     print(imputed_data)
```

7, Regarding Categorical value, first of all, convert the 'object' type to 'category' type.

Impute Categorical missing values

```
[44] cols = df.select_dtypes(['object']).columns.tolist()
     print(cols)
```

```
[45] for i in cols:
       df[i] = df[i].astype('category')
```

8, Impute them with **the imputation method with mode**.

```
[50] for column in cat_cols:
       mode = df[column].mode()[0]
       df[column] = df[column].fillna(value=mode)
```

9, Make sure all of the missing values are imputed and the ‘0’ missing value.

```
[51] df.isnull().sum()
```

	0
Unnamed: 0	0
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
ind_ahor_fin_ult1	0
ind_aval_fin_ult1	0
ind_cco_fin_ult1	0

10, For reader-friendly and part of EDA, rename each variable.

Project life cycle along with deadline:

Project weeks	Deadline	Lifecycle
Week7	Dec 19, 2024	Problem statement, Pre-process
Week8	Dec 26, 2024	Data process, understanding
Week9	**Jan 02, 2025**	**Data Cleaning, Merge, Review**
Week10	Jan 09, 2025	EDA, Final recommendation
Week11	Jan 16, 2025	EDA presentation for business users
Week12	Jan 23, 2025	Model Selection and Model Building/Dashboard
Week13	Jan 30, 2025	Final Project Report and Code

Tabular data details: cust_seg.csv.zip:

Total number of observations	1000000
Total number of files	1
Total number of features	48
Base format of the file	csv.zip
Size of the data	19MB