Business Analytics – TP2

Customer segmentation

Maxime Manderlier
Fabian Lecron

1 Introduction

You are business analysts and data scientists working for a Portuguese banking institution. Your mission is to leverage customer data to create meaningful marketing segmentation. This segmentation will help the business tailor its direct marketing campaigns (conducted via phone calls) and improve campaign efficiency.
The dataset contains information about clients contacted during past marketing campaigns. Your task is to explore and interpret the data to create clusters that can guide marketing strategies.

1.1 Dataset description

The dataset consists of the following attributes:

Attribute	**Description**
`age`	Age of the client
`job`	Type of job (e.g., admin, technician, blue-collar)
`marital`	Marital status (e.g., single, married, divorced)
`education`	Level of education (e.g., high school, university degree)
`default`	Has credit in default? (`yes`, `no`, `unknown`)
`balance`	Average yearly balance, in euros, for the client
`housing`	Has a housing loan? (`yes`, `no`, `unknown`)
`loan`	Has a personal loan? (`yes`, `no`, `unknown`)
`contact`	Contact communication type (`cellular` or `telephone`)
`month`	Last contact month of year (e.g., Jan, Feb)
`day_of_week`	Last contact day of the week
`duration`	Duration of the last contact in seconds
`campaign`	Number of contacts performed during this campaign
`pdays`	Number of days since the client was last contacted (-1 means not previously contacted)
`previous`	Number of contacts performed before this campaign
`poutcome`	Outcome of the previous marketing campaign (`success`, `failure`, `nonexistent`)

Table 1: Description of dataset attributes

2 Dataset preparation and initial clustering with KMeans

2.1 Load and understand the dataset

To load the dataset and remove the target column y, you can choose one of the following approaches:

2.1.1 Using the `datasets` library

```
from datasets import load_dataset

# Load the CSV file using the datasets library
dataset = load_dataset("csv",
                       data_files="/data/datasets/TP2/bank-additional-full.csv",
                       delimiter=";")

# Drop the column 'y'
dataset = dataset.remove_columns("y")
```

2.1.2 Using `pandas`

```
import pandas as pd

# Load the dataset with pandas
dataset = pd.read_csv("/data/datasets/TP2/bank-additional-full.csv", sep=";")

# Drop the column 'y'
dataset = dataset.drop(columns=["y"])
```

Note: Both methods are valid; use the one that fits your preferences. You can also switch between them based on your needs, as they are compatible.

2.2 Preprocess the data

Before clustering, preprocess the dataset to ensure it is ready for analysis. Consider the following steps:

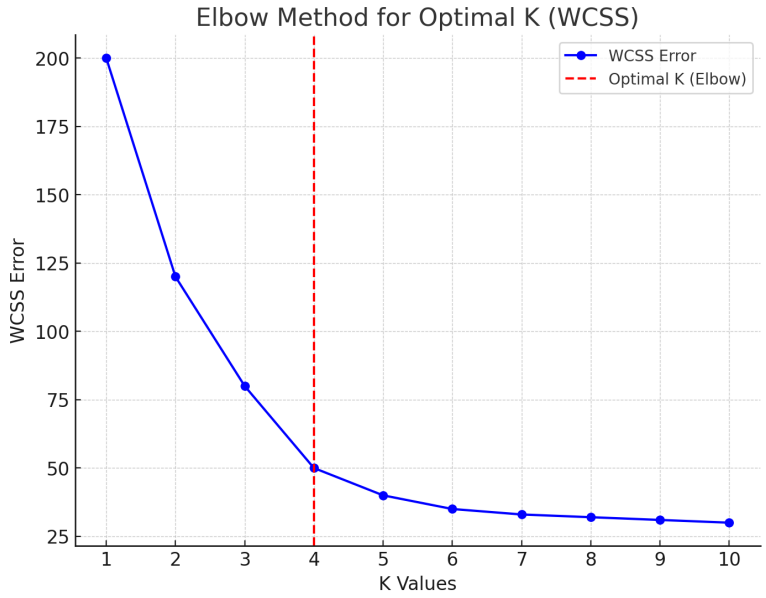
- Check for **duplicates**: Are there any rows repeated in the dataset? Remove them if necessary.
- Handle **outliers**: Are there extreme values in the numerical attributes that could distort clustering results?
- Address **missing values**: How should null or unknown values be managed? For example, you can impute them with a mean, median, or mode, or drop them if appropriate.
- **Encode categorical variables**: Transform categorical data into numerical format using techniques such as one-hot encoding or label encoding.

- **Scale numerical variables**: Normalize or standardize numerical data to ensure all attributes have comparable scales.
- Explore **correlations or redundancy**: Are some variables highly correlated or redundant? You may decide to drop certain variables.

2.3 Tasks for KMeans clustering

Perform the following tasks with KMeans:

- Determine the optimal number of clusters using the **elbow method**.



- Visualize the clusters (use dimensionality reduction techniques if necessary).
- Calculate and interpret the **silhouette coefficient** to evaluate clustering quality.
- Identify the variables that influence cluster assignments the most.
- Provide a business-level interpretation of the clusters.

3 Addressing limitations of KMeans

3.1 K-Modes: a clustering method for categorical data

KMeans is effective for clustering numerical data but struggles with categorical attributes because it relies on Euclidean distance, which is not meaningful for non-numerical data.

To address this limitation, **K-Modes** is specifically designed for clustering categorical data. It works by:

- Using a dissimilarity measure based on the matching of categories rather than Euclidean distance.
- Updating cluster centroids using the mode (most frequent value) instead of the mean.

Challenge: While K-Modes handles categorical data effectively, it does not account for numerical attributes. Reflect on how you could combine the advantages of KMeans for numerical data and K-Modes for categorical data into a single clustering algorithm. Research and implement a solution to cluster datasets with mixed data types.

- Perform the same tasks as for KMeans:
  - Determine the optimal number of clusters using an equivalent elbow method.
  - Visualize the clusters.
  - Calculate and interpret the silhouette coefficient.
  - Identify influential variables for cluster assignments.
  - Provide a business interpretation of the clusters.

4 Enhancing clustering with an embedding model (LLM-based)

4.1 Compute embeddings with an embedding model

To enhance clustering, use an embedding model to generate embeddings for each row in the dataset. These embeddings can represent categorical and text-based attributes more effectively in a continuous space.

Steps:

- Preprocess the dataset, ensuring all textual and categorical columns are properly formatted.
- Use the embedding model to compute embeddings for each data row.
- Choose one of the available local models from `./data/models/`:
  - `nomic-embed-text-v1.5`
  - `nomic-embed-text-v2-moe`
  - `bge-m3`

```
from transformers import AutoTokenizer, AutoModel

# Load the model and tokenizer
model_name = [MODEL]
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModel.from_pretrained(model_name, device_map="auto").to("cuda")
```

- Perform the same tasks as for KMeans:
  - Determine the optimal number of clusters.
  - Visualize the clusters.
  - Calculate and interpret the silhouette coefficient.
  - Identify influential variables for cluster assignments.
  - Provide a business interpretation of the clusters.

5 Final business interpretation and strategy generation

After comparing the results from all clustering methods, select the best-performing model. Then:

- Use the clusters to interpret customer segments in detail.
- Describe the key characteristics of each segment.
- Propose a marketing strategy tailored to each customer segment.
- Explain how these strategies can improve marketing campaigns and business outcomes.

Instructions: For this section, use a **large language model (LLM)** to generate the explanations and strategies, **as you did in TP1**. Follow the same procedure to interact with the model and produce the outputs.

Tips:

- Provide clear and concise descriptions of the clusters as input to the LLM.
- Generate detailed outputs that include actionable marketing strategies tailored to each customer segment.
- Ensure that the generated strategies align with the business objectives of improving marketing campaigns and overall outcomes.