Machine Learning and Neural Networks

Student Name : Dhanesh Kanakaraj

Student ID : 23056970

Assignment : Individual assignment: Machine learning tutorial

GitHub Link : https://github.com/kdhanesh619/Machine-Learning---Individual-Assignment

Principal Component Analysis on Customer Segmentation Dataset

Introduction

In today's world, data is everywhere. Companies collect information about what we buy, when we shop, and even how we browse websites. But just collecting data isn't enough. We need to understand it. And that can be a real challenge when the data has many columns, features, or variables.

This is where **Principal Component Analysis (PCA)** becomes helpful. PCA is a technique used in machine learning to make big datasets smaller without losing what's important. It helps us spot patterns, see clusters, and understand the bigger picture.

In this tutorial, I'll show you how PCA works using a real-world dataset about customers. This dataset contains lots of information about how customers behave what they buy, how often, and how much they spend. We'll apply PCA to reduce the number of features, and then we'll visualize the results to see what insights we can uncover.

Whether you're a beginner in machine learning or just someone curious about how we make sense of data, this tutorial will walk you through PCA in a simple and visual way.

What is PCA and Why Should We Use It?

Let's imagine you're working with a spreadsheet that has dozens of columns each one holding different information about your customers. There's data about their income, how many kids they have, how recently they've shopped, and how much they spend on things like wine, meat, or gold. It's a lot!

Now imagine trying to make sense of all of this at once. It's like trying to understand a conversation where ten people are talking at the same time. Confusing, right?

That's exactly where **Principal Component Analysis (PCA)** comes in.

So, What is PCA?

PCA is a mathematical technique that helps us simplify large datasets. It reduces the number of features while still keeping the most important parts of the data. It does this by creating new features (called *principal components*) that capture the key differences and patterns in the data.

These components are not random they're created in a smart way:

- The **first component** captures the biggest variation in the data.
- The **second component** captures the next biggest variation (but in a different direction).
- And it goes on like this.

By using just the first two or three components, we can still understand most of what's going on in the dataset while working with fewer variables.

Why Is This Useful?

PCA is especially useful when:

- You have too many features, and it's hard to visualize or process them.
- Some features are similar or correlated, and you want to simplify them.
- You want to clean up your data before using it in a machine learning model.

It's like decluttering a messy room you don't throw everything away, you just keep what's truly valuable.

About the Dataset

For this tutorial, I used a real-world dataset focused on **customer behaviour**. Imagine you're working in the marketing team of a company that wants to better understand its customers. What kind of people are buying what? Who's spending more? Are there groups of similar customers?

This dataset gives us a peek into exactly that. Each row represents a customer, and the columns give us information like:

- **Year of birth**
- **Education level**
- **Marital status**
- **Income**
- **How many kids or teenagers live in their home**
- **How much they spend on different products** (like wine, fruits, meat, etc.)
- **How recently they made a purchase**
- **How often they shop through different channels** (web, catalogue, store)
- **Whether they accepted marketing campaigns**

In total, there are **29 columns** (or features) and over **2,000 customer records**. That's a lot of data which is why PCA is such a great tool here.

Why This Dataset is a Good Fit for PCA

This customer segmentation dataset is perfect for PCA because:

- It has **many numerical features** that might be correlated (like spending habits).
- It includes **behavioural patterns** that can be grouped or simplified.
- It gives us a chance to **visualize** customer data in 2D or 3D space after reducing its dimensions.
- And importantly it's **realistic**, not overly cleaned or perfect, which makes the analysis more meaningful.

Our goal is to simplify this data, uncover hidden patterns, and explore what we can learn from a visual and analytical perspective.

Preparing the Data for PCA

Before we can use PCA, we need to prepare the data properly. Think of it like cooking a good meal you need to clean, cut, and prep the ingredients before you start cooking.

Our dataset was rich but a little messy. Here's how I cleaned and prepared it:

1. Dropping Unnecessary Columns

Some columns, like the customer ID or the date when they joined, didn't add much value for our PCA. These columns were more like labels or metadata. So, I removed:

- ID – just a number, not useful for analysis
- Dt_Customer – the join date, not in a format that helps with PCA
- Z_CostContact and Z_Revenue – these had the same values for all rows (no variation)

2. Handling Missing Values

The dataset had a few missing values in the Income column. Since we needed complete rows for PCA to work correctly, I removed any rows that had missing data.

3. Encoding Categorical Data

Next, I had to deal with columns like Education and Marital_Status, which contain text values like "Graduation" or "Single". PCA can't work directly with text it needs numbers. So I used something called **one-hot encoding**, which turns these text values into separate columns with 1s and 0s.

For example, the Marital_Status column turned into several columns like:

- Marital_Status_Married
- Marital_Status_Single
- Marital_Status_Divorced ...and so on.

Each customer would have a 1 in the column that matched their status, and 0s in the rest.

4. Scaling the Data

Finally, before applying PCA, I scaled all the numeric values so that they were on the same scale. Why? Because PCA looks at variance, and we don't want one feature (like income) to dominate the others just because it has larger numbers.

I used **Standard Scaler**, which transforms all features so that they have a mean of 0 and a standard deviation of 1.

Summary

At the end of this step, I had a clean, numeric, and standardized dataset ready to be analysed with PCA. This step was important to make sure PCA worked correctly and gave meaningful results.

Applying PCA and Visualizing the Results

Once the dataset was cleaned and standardized, I applied PCA using the scikit-learn library. The goal was to reduce the number of features while preserving the most important patterns in the data.

To understand how much information each principal component carried, I generated both a **scree plot** and a **cumulative explained variance plot**. These showed that just the first five components captured most of the dataset's variance, confirming that we could work with fewer dimensions without losing much meaning.

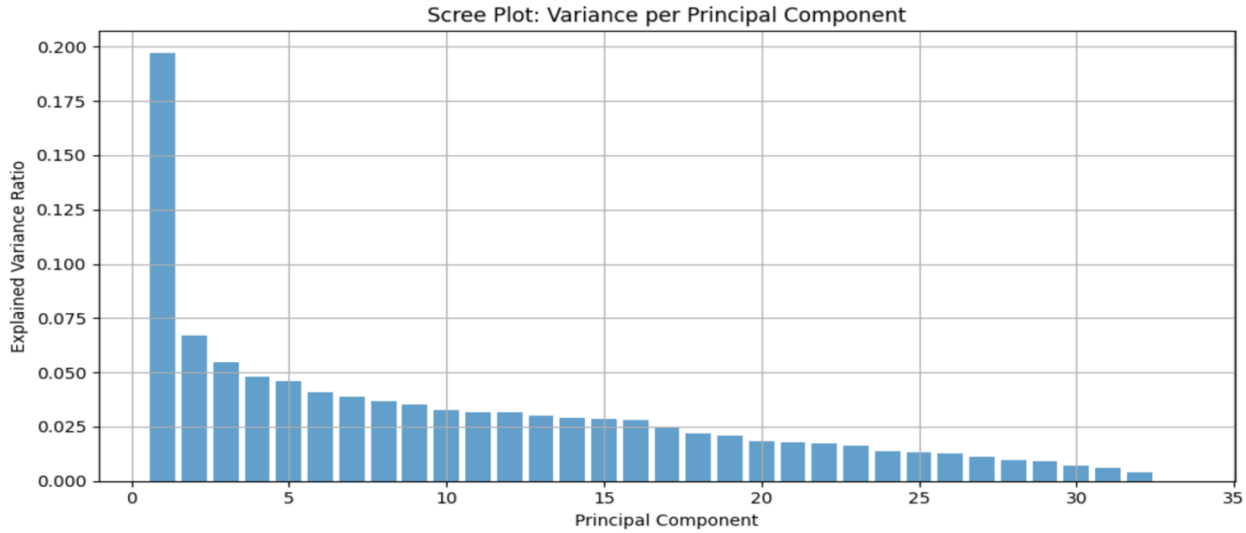

Figure 1: Scree Plot – Variance explained by each principal component

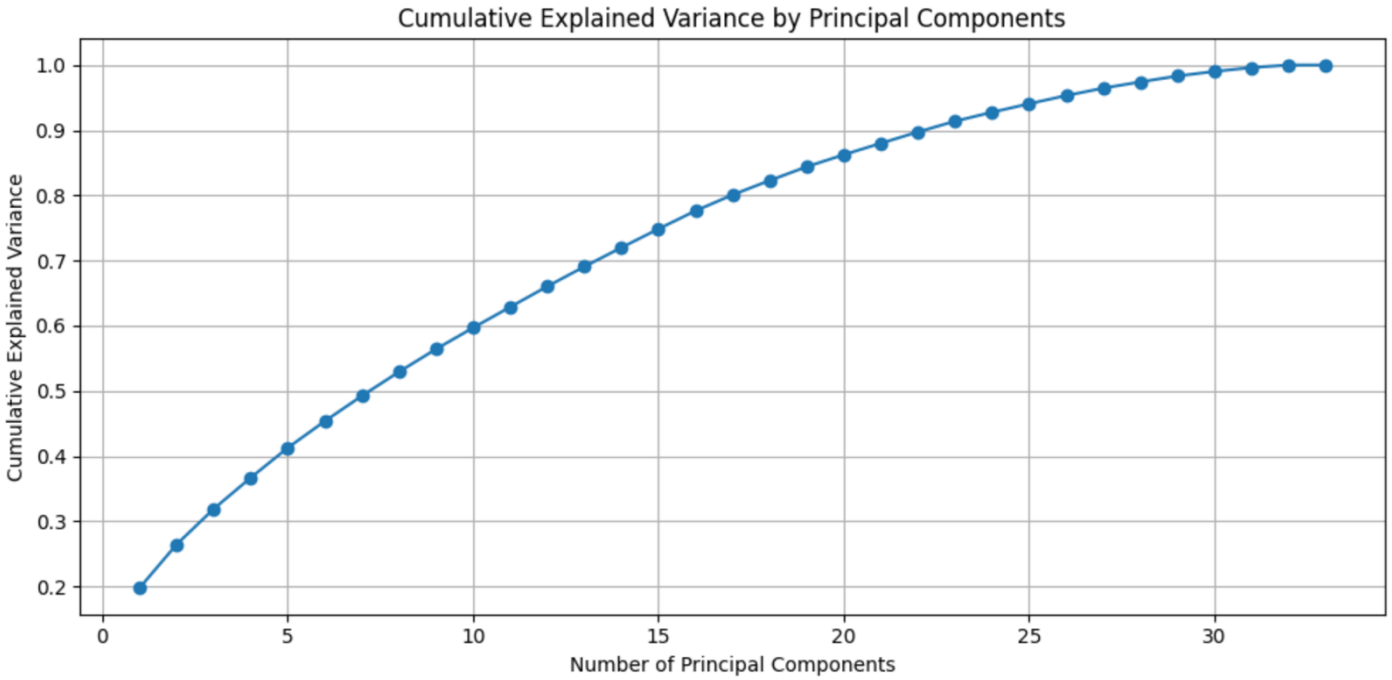

Figure 2: Cumulative Explained Variance – How much information is retained by top components.

I also used a **PCA loadings plot** to see which features influenced the components the most. This helped interpret what kind of customer behaviour each component was capturing for example, high contributions from features like MntWines, Income, and Recency.

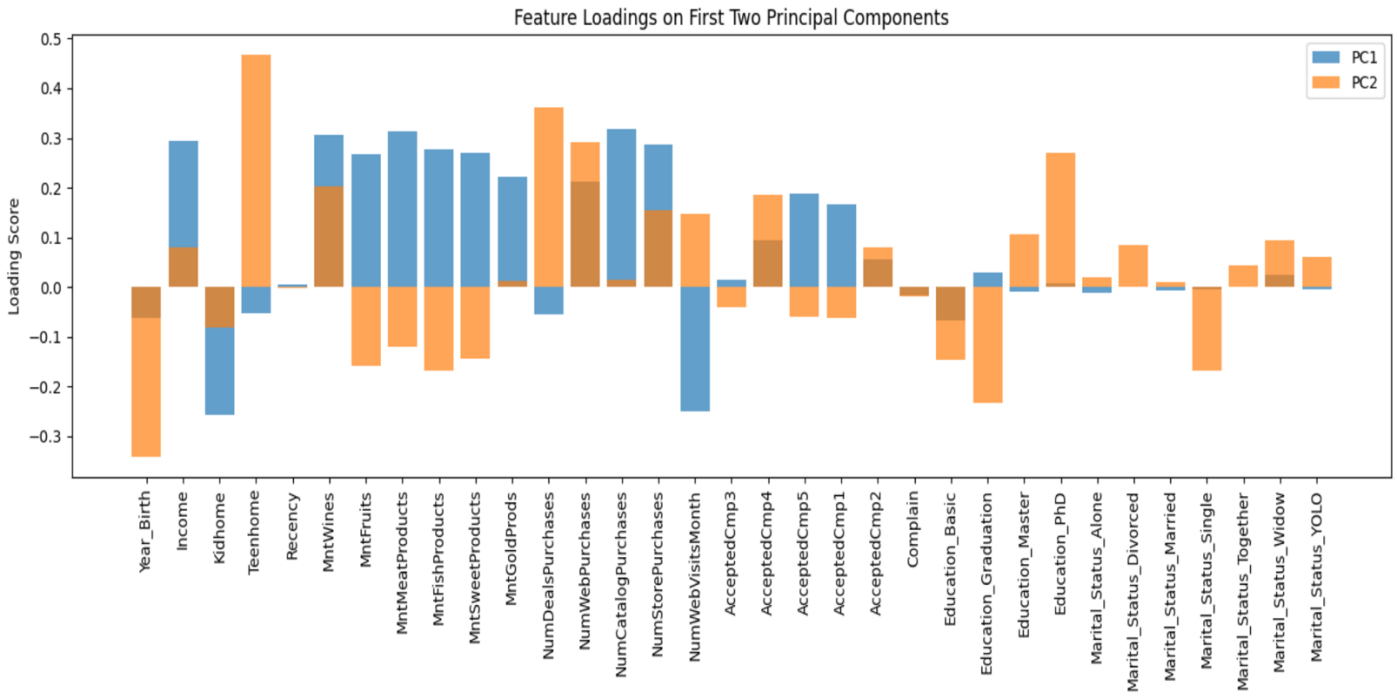

Figure 4: PCA Loadings – Feature contributions to the first two components

To make the results more visual, I created a **2D scatter plot** using the first two principal components. Each point represents a customer, and I used the Response variable (whether they accepted a marketing campaign) to colour the points. While the separation wasn't perfect, the plot revealed visible clusters and patterns, helping to identify different customer types.

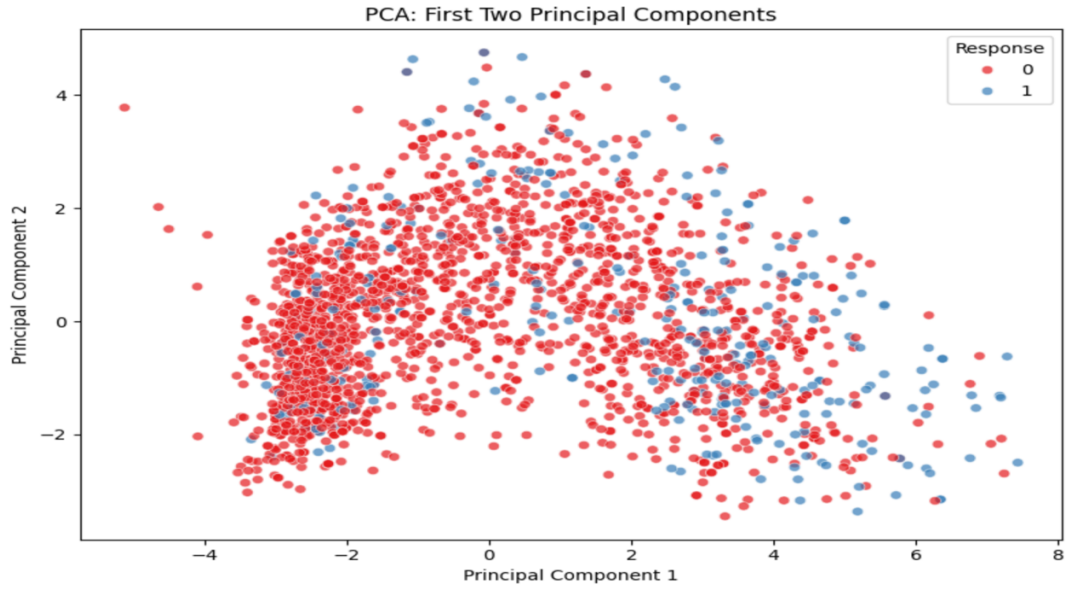

Figure 5: 2D PCA Scatter Plot with customer responses as colour categories

Additionally, I created a **correlation heatmap** to see how different features in the dataset were related to each other. This helped me notice that some features, especially the ones about how much customers spend on products like wine, meat, and sweets, were linked. For example, people who spent more on wine often also spent more on meat. These patterns showed that some parts of the data were saying similar things. That's why using PCA made sense—it helped to **combine these related features** into simpler ones, so the dataset could still tell the same story but with fewer columns.

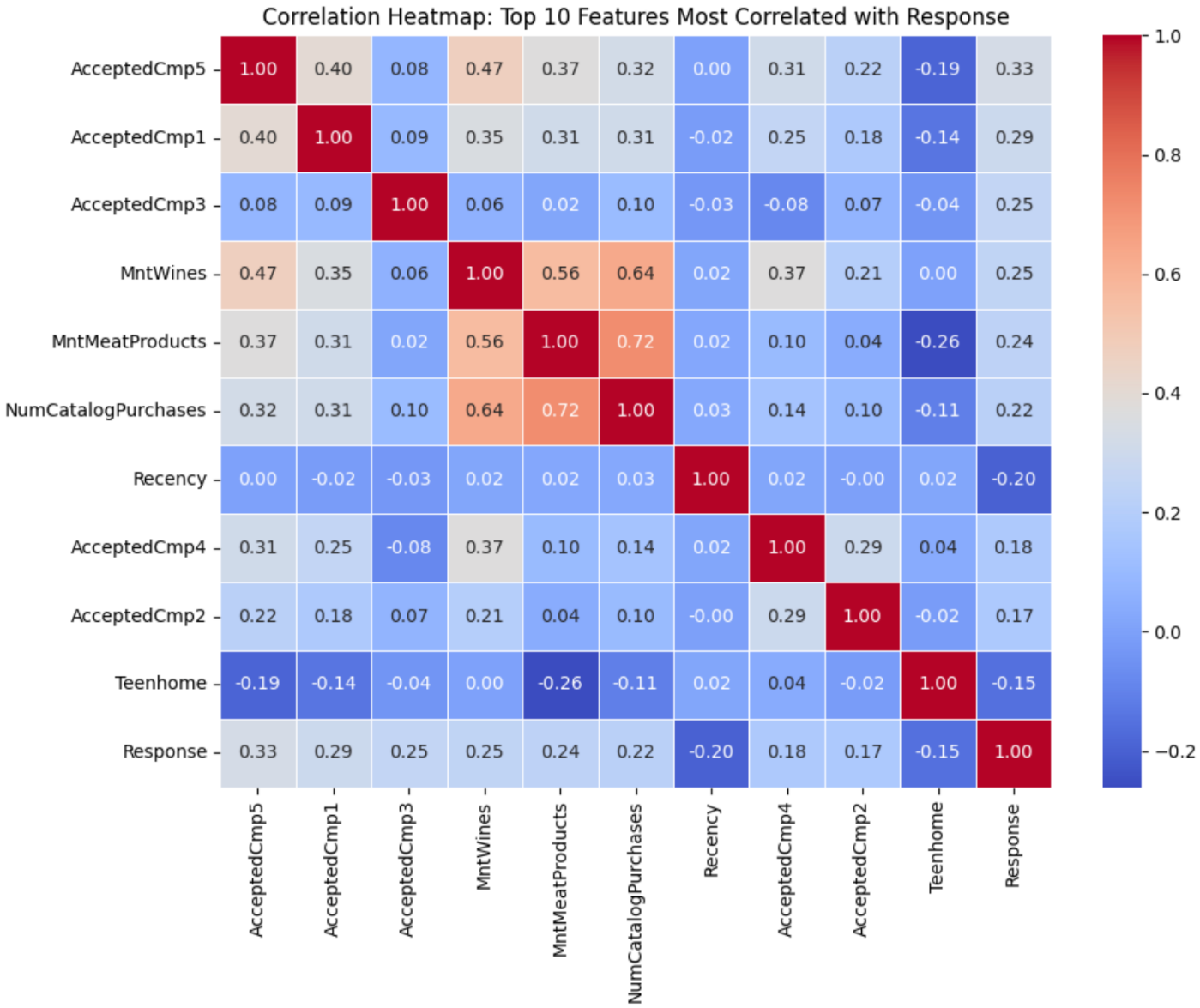

Figure 3: Correlation Heatmap showing relationships between original features

Insights and Takeaways

After applying PCA and visualizing the results, I had a much better understanding of the dataset and how different customers behaved. Here are some of the key insights and lessons I took away from this project:

1. PCA Helps Reveal Structure in the Data

One of the biggest benefits of PCA is how it **simplifies complex data**. The original dataset had more than 30 columns, and it was hard to see any patterns just by looking at them. But after reducing it to just 2 components, it became easier to spot **clusters and similarities** between customers.

This shows how PCA can be a great tool in **customer segmentation** and **pattern discovery**, especially when working with large, high-dimensional data.

2. Dimensionality Reduction Without Losing Much

Even though we reduced the number of features drastically, we didn't lose much information. The **explained variance plot** showed that the first few principal components held most of the important details. This means PCA allowed us to **keep the core of the dataset** while removing redundancy.

This can be especially helpful when preparing data for machine learning models, which often perform better with fewer, more meaningful features.

3. Visualization Matters

The 2D scatter plot turned numbers into something we could actually **see and interpret**. By colouring the points using the Response variable, we could start asking meaningful questions like:

- Are certain types of customers more likely to respond to campaigns?
- Do high-spending customers cluster in a specific region of the plot?

Even without doing full machine learning, PCA gave us a **visual way to think about customer behaviour**.

Final Thoughts

PCA turned out to be more than just a mathematical technique it was a practical, visual, and powerful way to explore and simplify data. It helped clean up noise, reveal hidden patterns, and prepare the dataset for further analysis.

Whether you're working in marketing, finance, or healthcare, PCA is a great tool to have in your data toolbox especially when you're faced with too many features and not enough clarity.

Conclusion

Working with this customer dataset showed me just how powerful PCA can be when exploring high-dimensional data. At first, the dataset felt overwhelming, with so many different features to look at. But PCA helped cut through the noise by finding the most important patterns hidden within the data.

By cleaning and preparing the dataset, applying PCA, and visualizing the results in just two dimensions, I was able to uncover groupings of customers and understand the structure of the data in a new, simpler way. This process not only helped make the data more manageable it also made it more meaningful.

PCA is an essential tool for data analysts and scientists because it:

- Reduces complexity
- Helps with visualization
- Reveals hidden relationships
- And prepares data for better performance in machine learning models

Through this project, I've learned how to take a real dataset, apply a machine learning technique, and turn the results into insights that could actually be used in a real-world business setting. And that's what makes PCA so valuable not just the math behind it, but the clarity it brings to messy, complicated data.

References

Here are some helpful sources I used to understand and apply PCA:

1. **Scikit-learn documentation on PCA**
   https://scikit-learn.org/stable/modules/decomposition.html#pca
2. **Towards Data Science - A Friendly Introduction to PCA**
   https://towardsdatascience.com/a-step-by-step-explanation-of-principal-component-analysis-3829fdf49f5e
3. **Machine Learning Mastery – Principal Component Analysis**
   https://machinelearningmastery.com/principal-components-analysis-for-dimensionality-reduction-in-python/
4. **Dataset Source: [Kaggle – Customer Segmentation Data]**
   https://www.kaggle.com/datasets/vishakhdapat/customer-segmentation-clustering/data