Member-only story

Mastering Customer Segmentation with Clustering Algorithms

Ritesh Gupta · Follow
Published in Python in Plain English · 5 min read · Nov 7, 2024

How to Use K-Means, DBSCAN, and More to Understand and Engage Your Audience

Customer segmentation is a simple tool that a business organization can use to understand their target group, customize the marketing, and improve the level of customer satisfaction. Unsupervised learning techniques can be used for clustering for meaningful grouping of customers concerning trends in their behavior. Therefore, this article shall support guidance through customer segmentation by well-popular algorithms such as K-Means, DBSCAN, and Hierarchical Clustering with some Customer datasets concerning Purchase History, Browsing Patterns, and Demographics.

Photo by Anastasia Shuraeva: https://www.pexels.com/photo/woman-in-white-robe-holding-white-paper-5720634/

Why Customer Segmentation? 💡

Customer segmentation allows companies to divide their customers into distinct groups, each characterized by unique behaviors and preferences. By understanding these segments, businesses can:

- **Personalize Marketing Campaigns** 🎯
- **Improve Customer Retention** 🔄
- **Identify High-Value Customers** 💰
- **Enhance Product Recommendations** 🛍️

For example, an e-commerce platform might use segmentation to send tailored product recommendations, offering discounts to value-conscious segments, or launching exclusive campaigns for high-spending customers.

Data for Customer Segmentation 📈

To create effective segments, we need to gather relevant data. Here are some common features used in customer segmentation:

Purchase History: How often, what products are purchased, average order value.

Browsing Patterns: Time spent on site, number of pages viewed, keywords used.

Demographics: Age, sex, area.

Customer Engagement: Email open rates, response to promotions.

These data points provide us with a complete view of customer behavior and enable us to form clusters based on different purchasing and engagement patterns.

Choosing the Right Clustering Algorithm 🤖

All this would depend upon the kind and complexity of the data; thus, here are some popular clustering algorithms presented in brief:

K-Means Clustering : Simple, efficient, and powerful for well-separated clusters. Requires K to be input in advance.

DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise. It identifies clusters of any shape and size and can identify outliers well, especially for noisy patterns.

Hierarchical Clustering : It generates a hierarchy of clusters and is particularly helpful for exploratory analysis in visualizing the data structure.

We will take one by one with the aim of cutting customers based on their behavioral information.

1. K-Means Clustering for Customer Segmentation 📍

Step 1: Preprocessing the Data 🔄

Standardize the data to ensure features contribute equally to the distance calculation. Use techniques like Min-Max scaling or Z-score normalization.

Step 2: Choosing the Optimal K with the Elbow Method 📐

K-Means requires specifying the number of clusters, K. The **Elbow Method** is used to find the ideal number by plotting the sum of squared distances (inertia) and identifying the “elbow” point where adding more clusters provides diminishing returns.

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Assume 'data' is your preprocessed DataFrame of customer features
inertia = []
```

```
K_range = range(1, 11)
for K in K_range:
    kmeans = KMeans(n_clusters=K, random_state=0).fit(data)
    inertia.append(kmeans.inertia_)

plt.plot(K_range, inertia, 'bo-')
plt.xlabel('Number of clusters K')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal K')
plt.show()
```

Step 3: Segmenting Customers and Analyzing Results 📊

After selecting K, apply K-Means to generate customer clusters. Visualize the clusters using a scatter plot (if feasible) and interpret the characteristics of each segment.

Example:

- **Cluster 1:** Frequent shoppers, high average order value 💸
- **Cluster 2:** Browsers with low purchase frequency 👀
- **Cluster 3:** Value-conscious customers who respond to discounts 💲

Targeted Marketing:

- **Cluster 1:** Offer exclusive previews of new products.
- **Cluster 2:** Encourage purchases with limited-time offers.
- **Cluster 3:** Send promotions with discounts.

2. DBSCAN for Customer Segmentation 🌀

DBSCAN is well-suited for customer data with natural clusters and noise (e.g., one-time visitors or inactive customers).

Step 1: Set Epsilon (ε) and Minimum Samples Parameters 📏

DBSCAN requires two parameters:

- **Epsilon (ε):** The maximum distance between points to be considered part of the same neighborhood.
- **Minimum Samples:** The minimum number of points to form a dense cluster.

Step 2: Identifying Clusters and Noise 🌐

DBSCAN groups customers into clusters based on density and labels outliers as **noise**.

```
from sklearn.cluster import DBSCAN

# Fit DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=5).fit(data)
data['Cluster'] = dbscan.labels_

# Visualize
plt.scatter(data['Feature1'], data['Feature2'], c=data['Cluster'], cmap='viridis
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('DBSCAN Clustering of Customers')
plt.show()
```

Example:

- **Core Cluster 1:** Engaged users with regular purchases 🛒
- **Core Cluster 2:** Rare buyers but high purchase value 💵
- **Noise:** Inactive or low-engagement users ⛔

Targeted Marketing:

- **Core Cluster 1:** Retention campaigns with loyalty rewards.
- **Core Cluster 2:** Personalized recommendations based on high-value items.
- **Noise:** Re-engagement emails with introductory offers.

3. Hierarchical Clustering for Customer Segmentation 🌳

Hierarchical clustering builds a tree (dendrogram) that helps us explore different segmentation levels without a predefined number of clusters.

Step 1: Create a Dendrogram

Visualize hierarchical relationships among customers using a dendrogram, which shows how clusters merge at various thresholds.

```
from scipy.cluster.hierarchy import dendrogram, linkage

linked = linkage(data, 'ward')
plt.figure(figsize=(10, 7))
dendrogram(linked)
```

```
plt.title('Customer Dendrogram')
plt.show()
```

Step 2: Choose Clusters Based on the Dendrogram 📏

Identify meaningful clusters by cutting the dendrogram at a height that balances segment detail with interpretability.

Example:

- **Cluster 1:** High spenders, frequent purchases 🏆
- **Cluster 2:** Bargain hunters, infrequent but discount-sensitive 🏷️
- **Cluster 3:** Occasional buyers, low engagement 📉

Targeted Marketing:

- **Cluster 1:** VIP loyalty programs with exclusive benefits.
- **Cluster 2:** Targeted promotions with discount codes.
- **Cluster 3:** Send reminders and incentives to drive repeat purchases.

Interpreting and Utilizing Clusters for Marketing 🚀

After forming clusters, interpret the segments based on their characteristics. Summarize each segment's preferences and tailor marketing strategies accordingly:

- **High-Value Customers (Cluster 1):** Provide exclusive offers and product launches.
- **Occasional Shoppers (Cluster 3):** Send targeted discounts and reminders to increase purchase frequency.
- **Discount Shoppers (Cluster 2):** Offer promotions during sales seasons to maximize engagement.

Real-World Example 📍

A **fashion retail company** segmented its customers using behavioral clustering. They found four clusters:

- **Fashionistas** (Cluster 1): Regularly buy trendy, high-end items 👠
- **Budget Shoppers** (Cluster 2): Purchase primarily during sales 💲

- **Gift Shoppers** (Cluster 3): Buy around holidays and special occasions 🎁
- **Infrequent Buyers** (Cluster 4): Low engagement, few purchases 📉

Based on these insights, the retailer ran targeted campaigns:

- **Fashionistas** received new arrival notifications and loyalty perks.
- **Budget Shoppers** got discount codes during holiday sales.
- **Gift Shoppers** were targeted with gift card promotions and holiday deals.
- **Infrequent Buyers** received re-engagement emails with discounts to encourage purchases.

Conclusion 🏁

One of the strongest approaches to understanding and leveraging customer behavior for targeted marketing is customer segmentation with the help of clustering. Enabling the business to create actionable segments through algorithms, it may use K-Means, DBSCAN, or hierarchical clustering, so businesses can personify the strategy to meet the diverse needs of the customers. Through these techniques, customer satisfaction and ROI are improved because marketing efforts are aligned with specific audience segments.

All set to get deeper? Explore your data and segment your customers for deeper insight and better marketing results! Ready to get deeper?

In Plain English 🚀

*Thank you for being a part of the **In Plain English** community! Before you go:*

- Be sure to **clap** and **follow** the writer 👏
- Follow us: **X** | **LinkedIn** | **YouTube** | **Discord** | **Newsletter** | **Podcast**
- **Create a free AI-powered blog on Differ.**
- More content at **PlainEnglish.io**

Published in Python in Plain English

31K Followers · Last published 11 hours ago

New Python content every day. Follow to join our 3.5M+ monthly readers.

Follow

Written by Ritesh Gupta

3.6K Followers · 28 Following

Data Scientist, I write Article on Machine Learning| Deep Learning| NLP | Open CV | AI Lover ❤️

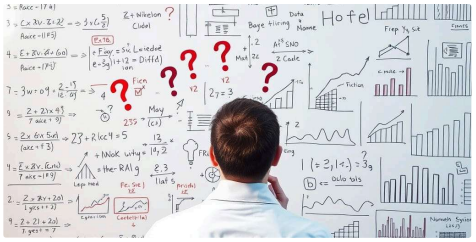
Follow

No responses yet

What are your thoughts?

Respond

More from Ritesh Gupta and Python in Plain English

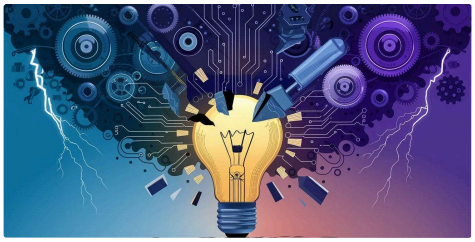


Ritesh Gupta

Can You Handle These 25 Toughest Data Science Interview Questions?

The role of a Data Scientist demands a unique blend of skills, including statistics, machine...

Sep 25 211 7



In Python in Plain English by Abdur Rahman

5 Overrated Python Libraries (And What You Should Use Instead)

Traditional Devs, Look Away—This One's Not for You!

Nov 3 2.3K 20

In Python in Plain English by Abdur Rahman

8 Uncommon but Extremely Useful Python Libraries for 2025

You'll regret not knowing this earlier

Nov 5 817 7

In Python in Plain English by Ritesh Gupta

7 GitHub Repos to Transform You into a Pro ML/AI Engineer

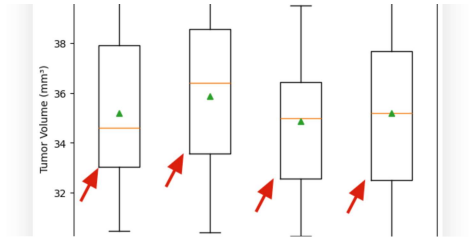
Hands-On Guides, Tools, and Frameworks to Fast-Track Your AI Journey

Nov 5 143 1

See all from Ritesh Gupta

See all from Python in Plain English

Recommended from Medium

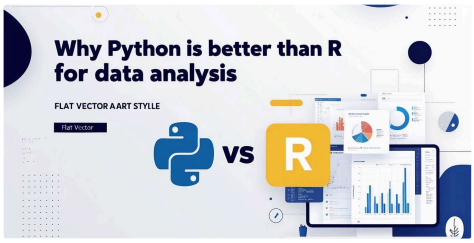


In Towards Data Science by Leo Anello

Statistical Analysis Using Python: Insights from Cancer Treatment...

Step-by-step exploration of statistical methods, data visualization, and regression...

3d ago 447 4



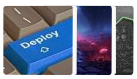
In Python in Plain English by Mayur Koshti

Why Python is Better than R for Data Analysis

Python and R for data analysis would involve breaking down each relevant aspect, feature...

Nov 15 34 7

Lists



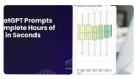
Predictive Modeling w/ Python

20 stories · 1691 saves

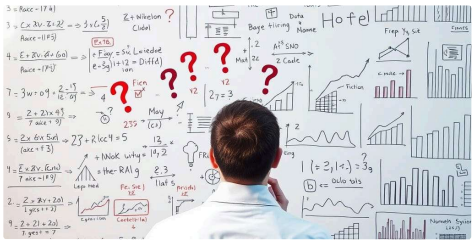
Practical Guides to Machine Learning

10 stories · 2057 saves

Natural Language Processing
1839 stories · 1461 saves



ChatGPT prompts
50 stories · 2295 saves

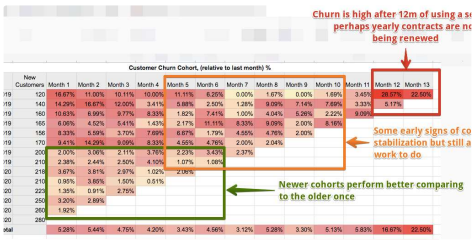


Ritesh Gupta

Can You Handle These 25 Toughest Data Science Interview Questions?

The role of a Data Scientist demands a unique blend of skills, including statistics, machine...

Sep 25 · 211 · 7

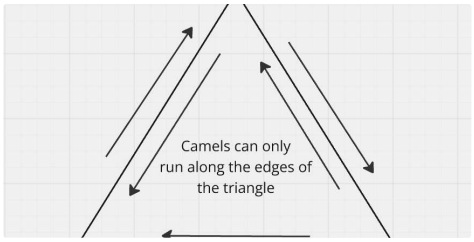


In Stackademic by Muhammad Muhsi Sidik

Data Visualization for Customer Retention: Unlocking Insights wit...

Customer retention is the lifeblood of any business striving for long-term success....

Nov 20 · 7



Lucas Samba

3 Probability Questions I was asked in Walmart Data Scientist Interview

Recently I got an opportunity to interview at Walmart for Data Scientist—3 position. All...

Aug 23 · 1.1K · 32

In DataDrivenInvestor by Bex T.

Firecrawl: How to Scrape Entire Websites With a Single Command...

And turn them into LLM-ready data

4d ago · 294 · 3

See more recommendations