Customer segmentation using data science

Introduction :

1. **Data Preparation**
2. **Exploring the content of variables**
3. **Insight on product categories**
4. **Customer categories**
5. **Classifying customers**

1. Data Preparation :

Data preparation is the process of preparing raw data so that it is suitable for further processing and analysis. Key steps include collecting, cleaning, and labeling raw data into a form suitable for machine learning (ML) algorithms and then exploring and visualizing the data.

2. Exploring the content of variables :

Exploring variable relationships can be made easier through the **use of visual tools, such as charts, graphs, or plots**. These tools can provide insight into the distribution, shape, and variation of your variables, as well as the direction, strength, and form of their relationships.

3. Insight on product categories :

Category insights are pieces of knowledge that allow brands to develop a deeper understanding of the dynamics and relationships between other brands in their product/service category, as well as how the category serves its consumers.

4. Customer categories :

Customer categories Examples :

1.Gender.
2.Age.
3.Occupation.
4.Marital Status.
5.Household Income.
6.Location.
7.Preferred Language.
8.Transportation.

5.Classifying customers :

Types of Customers

Five Main Types of Customers. In the retail industry, customers can be segmented into five main types: ...

Loyal Customers. ...

Impulse Customers. ...

Discount Customers. ...

Need-Based Customers. ...

Wandering Customers. ...

Related Readings.

Common customer classifications used include gender, age, geographic location, and income. Organizations use the classifications to determine who is using their services and buying their products. They also use these classifications to determine and target programs, such as marketing campaigns.

1. Data preparation

Input :

```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import datetime, nltk, warnings
import matplotlib.cm as cm
import itertools
from pathlib import Path
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples,
silhouette_score
from sklearn import preprocessing, model_selection,
metrics, feature_selection
from sklearn.model_selection import GridSearchCV,
learning_curve
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
```

```
 from sklearn import neighbors, linear_model, svm,
tree,
ensemble
from wordcloud import WordCloud, STOPWORDS
from sklearn.ensemble import AdaBoostClassifier
from sklearn.decomposition import PCA
from IPython.display import display, HTML
import plotly.graph_objs as go
from plotly.offline import init_notebook_mode,iplot
init_notebook_mode(connected=True)
warnings.filterwarnings("ignore")
plt.rcParams["patch.force_edgecolor"] = True
plt.style.use('fivethirtyeight')
mpl.rc('patch', edgecolor = 'dimgray', linewidth=1)
%matplotlib inline

#_____________________
# read the datafile
   df_initial =
   pd.read_csv('../input/data.csv',encoding="ISO-
8859-1",
dtype={'CustomerID': str,'InvoiceID': str})
print('Dataframe dimensions:', df_initial.shape)
#_______
```

```
df_initial['InvoiceDate'] = pd.to_datetime(df_initial['
InvoiceDate'])
  #_____
_____

_____
# gives some infos on columns types and numer of null
values

  tab_info=pd.DataFrame(df_initial.dtypes).T.rename
(index
={0:'column type'})
  tab_info=tab_info.append(pd.DataFrame(df_initial.i
snull()
.sum()).T.rename(index={0:'null values (nb)'}))
  tab_info=tab_info.append(pd.DataFrame(df_initial.i
snull()
.sum()/df_initial.shape[0]*100).T.
rename(index={0:'null values (%)'}))
display(tab_info)
#_____
# show first lines
display(df_initial[:5])
```

Dataframe dimensions: (541909, 8)

Output :

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom

2. Exploring the content of variable

Input :

```
df_check = df_initial[(df_initial['Quantity'] < 0) &
(df_
initial['Description'] != 'Discount')][
['CustomerID','Quantity','StockCode',
'Description','UnitPrice']]
for index, col in df_check.iterrows():
  if df_initial[(df_initial['CustomerID'] == col[0]) &
(df_
initial['Quantity'] == -col[1])
```

```
& (df_initial['Description'] == col[2])].shape[0] =
= 0:
print(index, df_check.loc[index])
print(15*'-'+'>'+' HYPOTHESIS NOT FULFILLED')
break
```

Output :

```
154 CustomerID 15311
Quantity -1
StockCode 35004C
Description SET OF 3 COLOURED FLYING DUCKS
UnitPrice 4.65
Name: 154, dtype: object
```

3.Insight on product categories

Input :

```
matrix = X.as_matrix()
for n_clusters in range(3,10):
kmeans = KMeans(init='k-means++', n_clusters = n_
clusters, n_init=30)
kmeans.fit(matrix)
clusters = kmeans.predict(matrix)
silhouette_avg = silhouette_score(matrix, clusters)
```

```
print("For n_clusters =", n_clusters, "The average
silhouette_score is :", silhouette_avg)
```

Output :

```
For n_clusters = 3 The average silhouette_score is : 0.
1007168175806424 8
For n_clusters = 4 The average silhouette_score is : 0.
1220823976115394 4
For n_clusters = 5 The average silhouette_score is : 0.
1470081849157512
For n_clusters = 6 The average silhouette_score is : 0.
1438984147242635 4
For n_clusters = 7 The average silhouette_score is : 0.
1521222011014401 7
For n_clusters = 8 The average silhouette_score is : 0.
1558201267218184
For n_clusters = 9 The average silhouette_score is : 0.
1165617340911786 2
```

4. Customer categories

Input :

```
corresp = dict()for key, val in zip (liste_produits, clusters):
  corresp[key] = val
  #_____
  _____df_cleaned['categ_
  product'] = df_cleaned.loc[:, 'Description'].map(corresp)

  for i in range(5):
   col = 'categ_{}'.format(i)
  df_temp = df_cleaned[df_cleaned['categ_product'] == i]
  price_temp = df_temp['UnitPrice'] * (df_temp['Quantity']
  - df_temp['QuantityCanceled'])
  price_temp = price_temp.apply(lambda x:x if x > 0 else
0)
  df_cleaned.loc[:, col] = price_temp
  df_cleaned[col].fillna(0, inplace =
  True)#_____
  _____
  _____
  _____df_clea ned[['InvoiceNo', 'Description',
  'categ_product', 'categ_0', 'categ_1', 'categ_2',
  'categ_3','categ_4']][:5]
```

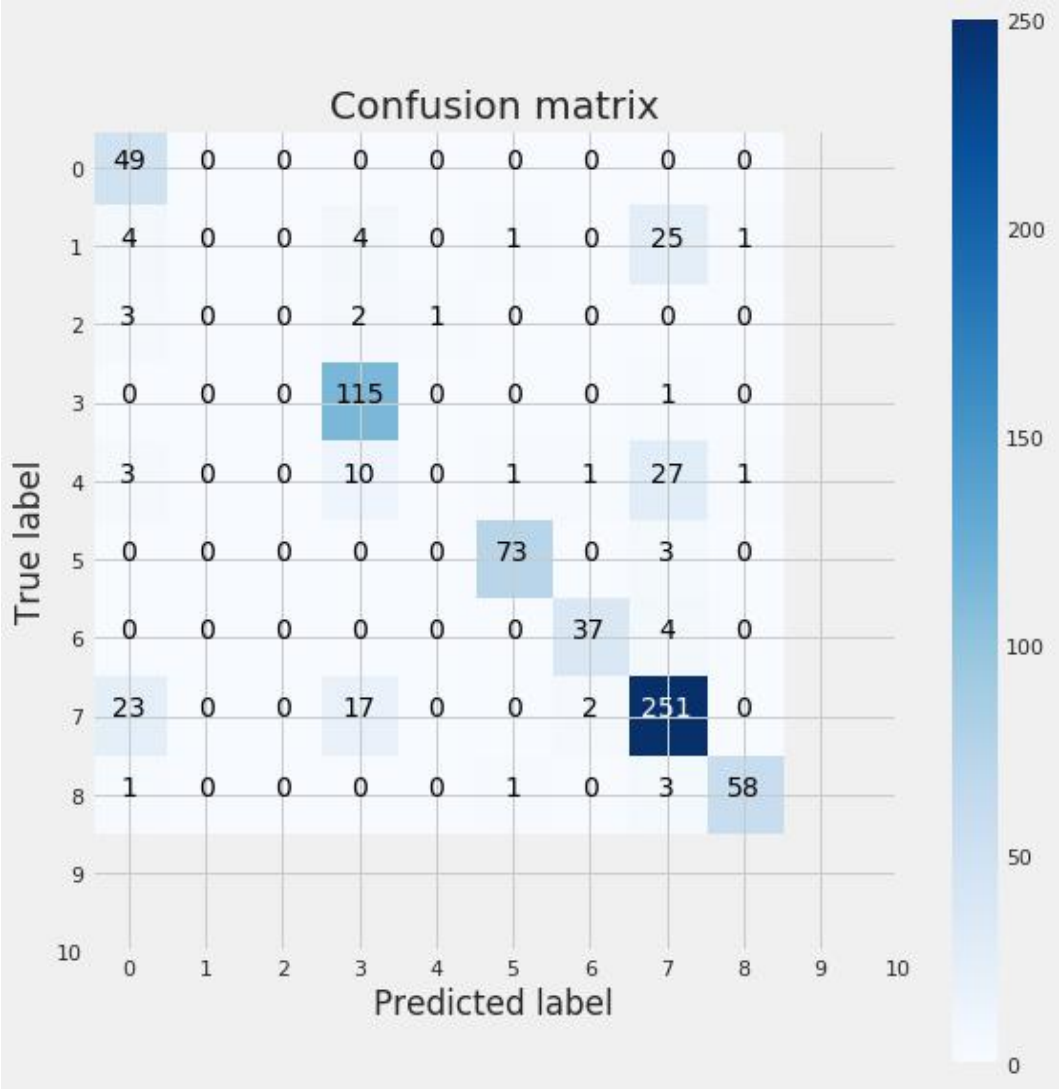
5. Classification of customers :

Input :

```
def plot_confusion_matrix(cm, classes,
normalize=False, title='Confusion matrix',
cmap=plt.cm.Blues):
if normalize:
cm = cm.astype('float') / cm.sum(axis=1)[:,
np.newaxis]
print("Normalized confusion matrix")
else:
print('Confusion matrix, without normalization')

#_____
_____
plt.imshow(cm, interpolation='nearest', cmap=cmap)
plt.title(title)
plt.colorbar()
tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation=0)
plt.yticks(tick_marks, classes)
#_____
_____
```

```
fmt = '.2f' if normalize else 'd'

thresh = cm.max() / 2.

for i, j in itertools.product(range(cm.shape[0]),
  range(cm.shape[1])):

  plt.text(j, i, format(cm[i, j], fmt),
      horizontalalignment="center",
           color="white" if cm[i, j] > thresh else "black")

#_____________________________________________

_______
 plt.tight_layout()
 plt.ylabel('True label')
 plt.xlabel('Predicted label')
```

Output :


Confusion matrix
True label
Predicted label
0 1 2 3 4 5 6 7 8 9 10
49 0 0 0 0 0 0 0 0
4 0 0 4 0 1 0 25 1
3 0 0 2 1 0 0 0 0
0 0 0 115 0 0 0 1 0
3 0 0 10 0 1 1 27 1
0 0 0 0 0 73 0 3 0
0 0 0 0 0 0 37 4 0
23 0 0 17 0 0 2 251 0
1 0 0 0 0 1 0 3 58
250
200
150
100
50
0

7. Conclusion :

The work described in this notebook is based on a database providing details on purchases made on an E-commerce platform over a period of one year. Each entry in the dataset describes the purchase of a product, by a particular customer and at a given date. In total, approximately ～～4000 clients appear in the database. Given the available information, I decided to develop a classifier that allows to anticipate the type of purchase that a customer will make, as well as the number of visits that he will make during a year, and this from its first visit to the E-commerce site.

The first stage of this work consisted in describing the different products sold by the site, which was the subject of a first classification. There, I grouped the different products into 5 main categories of goods. In a second step, I performed a classification of the customers by analyzing their consumption

habits over a period of 10 months. I have classified clients into 11 major categories based on the type of products they usually buy, the number of visits they make and the amount they spent during the 10 months. Once these categories established, I finally trained several classifiers whose objective is to be able to classify consumers in one of these 11 categories and this from their first purchase. For this, the classifier is based on 5 variables which are:

Thank you

Customer Segmentation Using data science

customer Customer segmentation is the process of dividing a base into groups of individuals that are similar in certain ways relevant to marketing, such as age, gender, interests, and spending habits. It enables companies to target specific groups with tailored promotions, products, or services that are most likely to resonate with them. Machine learning has become a popular tool for automating the process of customer segmentation, providing a more efficient and effective way to identify patterns and relationships within customer data.

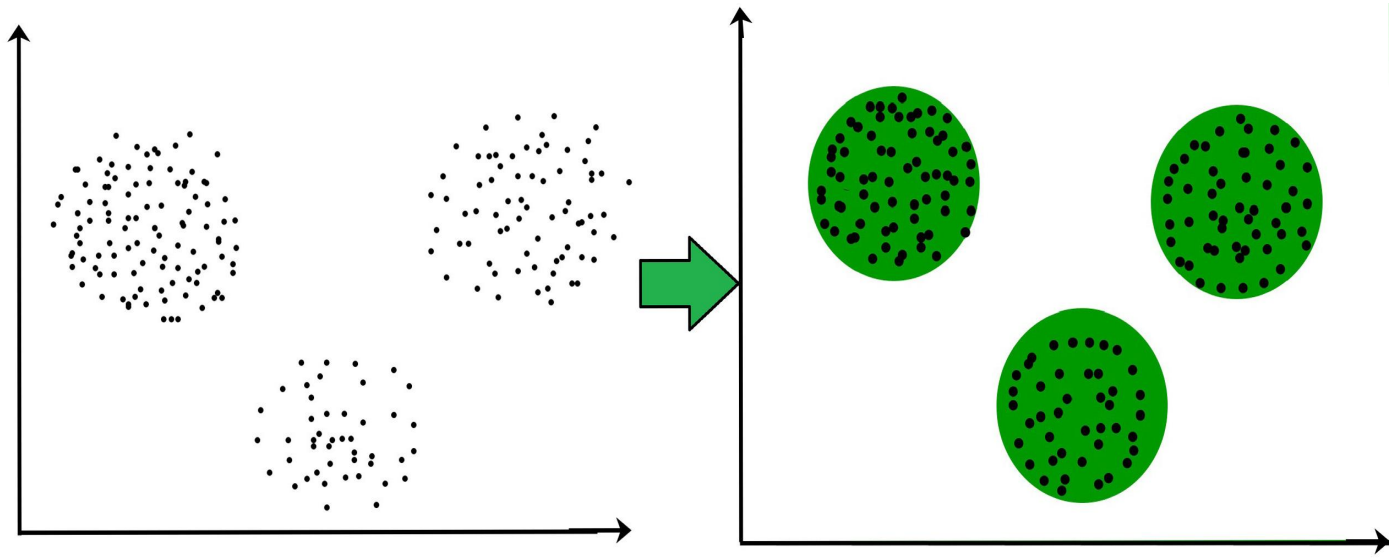
Different methods for using machine learning :

1. Clustering algorithms
2. Decision trees
3. Neural networks
4. Association rule learning

1. Clustering algorithms :

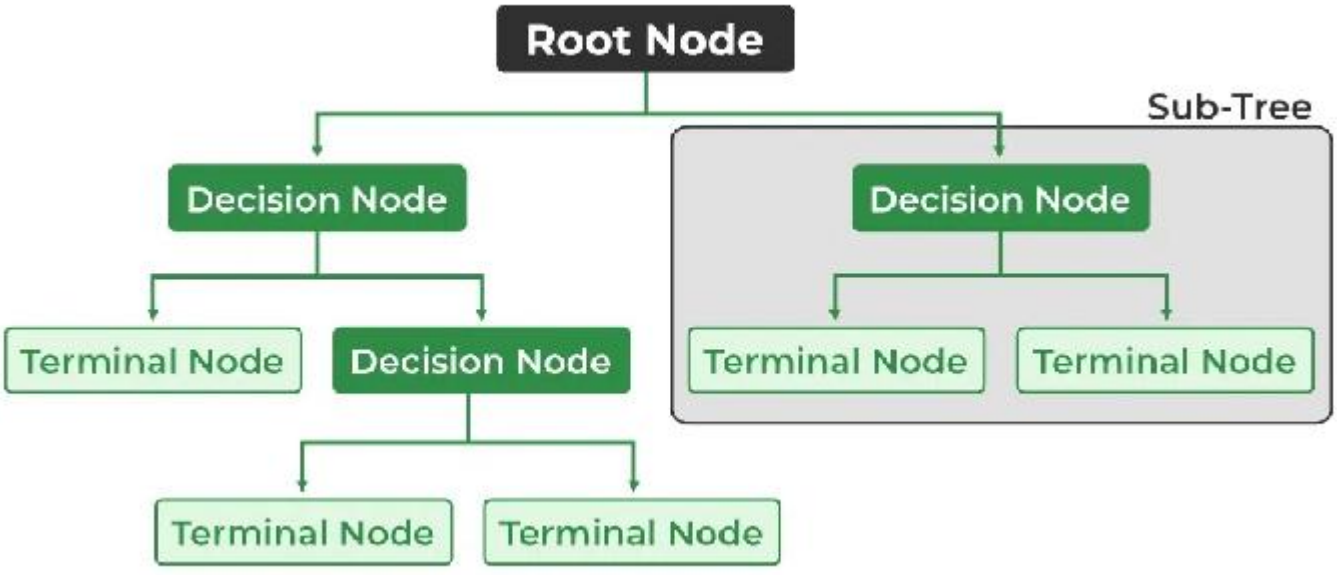
Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

For example The data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, and we can identify that there are 3 clusters in the below picture



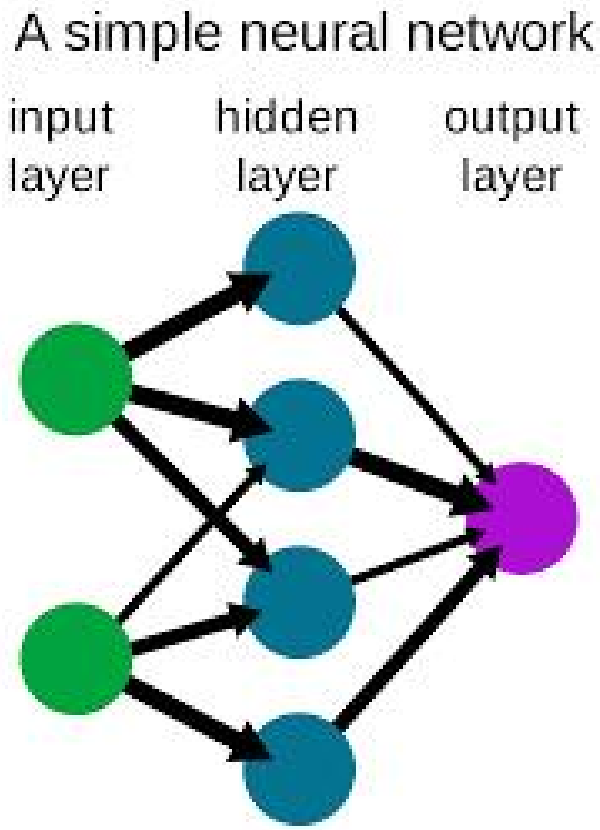
2. Decision trees :

A decision tree is a flowchart-like tree structure where each internal node denotes the feature, branches denote the rules and the leaf nodes denote the result of the algorithm. It is a versatile supervised data science algorithm, which is used for both classification and regression problems. It is one of the very powerful algorithms. And it is also used in Random Forest to train on different subsets of training data, which makes random forest one of the most powerful algorithms in data science.



3. Neural networks :

A neural network is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain.

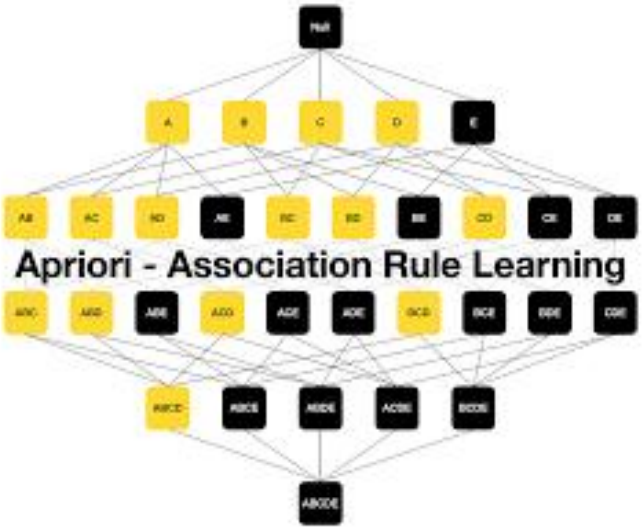


4. Association rule learning :

Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently a itemset occurs in a transaction. A typical example is a Market Based Analysis.

Market Based Analysis is one of the key techniques used by large relations to show associations between items.It allows retailers to identify relationships between the items that people buy together frequently.

Given a set of transactions, we can find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.



Clustering algorithms program :

Example Input :

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

# Generate random data for demonstration
np.random.seed(0)
X = np.random.rand(100, 2)

    # Create clusters by adding random offsets to the
    data points
cluster1 = 2 * X + 1
cluster2 = 2 * X - 1
data = np.vstack((cluster1, cluster2))

# Define the number of clusters
k = 2
```

Customer Segmentation Using data Science :

Introduction :

Customer segmentation in AI refers to the process of categorizing a company's customers into distinct groups based on various characteristics and behaviors using artificial intelligence techniques. These characteristics can include demographics, purchasing habits, online behavior, and more. The goal of customer segmentation is to better understand and target different customer groups with tailored marketing strategies, product recommendations, and services to enhance customer satisfaction and ultimately drive business growth. AI helps automate and refine this process by

analyzing large amounts of data to identify meaningful customer segments.

Types of AI in Costomer Segmentation :

Artificial Intelligence (AI) can be applied to customer segmentation in various ways to enhance the accuracy and effectiveness of the process. Here are some common types of AI techniques used for customer segmentation:

1. **Clustering Algorithms :** AI algorithms like K-Means, hierarchical clustering, or DBSCAN can group customers based on their similarities in terms of purchasing behavior, demographics, or other features.

2. Classification Algorithms : Algorithms such as decision trees, random forests, or support vector machines can be used for predictive customer segmentation. These algorithms categorize customers into predefined segments based on historical data.

3. Neural Networks : Deep learning models, particularly neural networks, can analyze complex data and uncover hidden patterns in customer behavior. Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) can be used for sequential and image-based data, respectively.

4. Natural Language Processing (NLP) :

NLP techniques are employed to analyze customer reviews, feedback, or social media interactions, allowing for sentiment-based segmentation and understanding customer opinions.

5. Recommendation Systems :

Collaborative filtering and content-based recommendation systems use AI to segment customers based on their preferences and past interactions, enabling personalized recommendations.

6. **Anomaly Detection :** AI can identify unusual behavior or fraud by detecting anomalies in customer transactions, which

can be a form of segmentation for risk management.

7. Dimensionality Reduction : Techniques like Principal Component Analysis (PCA) or t-Distributed Stochastic Neighbor Embedding (t-SNE) can reduce the dimensionality of data while retaining its structure, which can aid in customer segmentation.

8. Time-Series Analysis : For businesses with temporal data, AI techniques can segment customers based on their historical behaviors and changes over time.

9. Reinforcement Learning : In some cases, AI models can learn to segment customers by interacting with them and optimizing their segmentation based on business goals.

10. Hybrid Models : Many companies use a combination of these AI techniques to perform customer segmentation, as different approaches may be suitable for different aspects of the business.