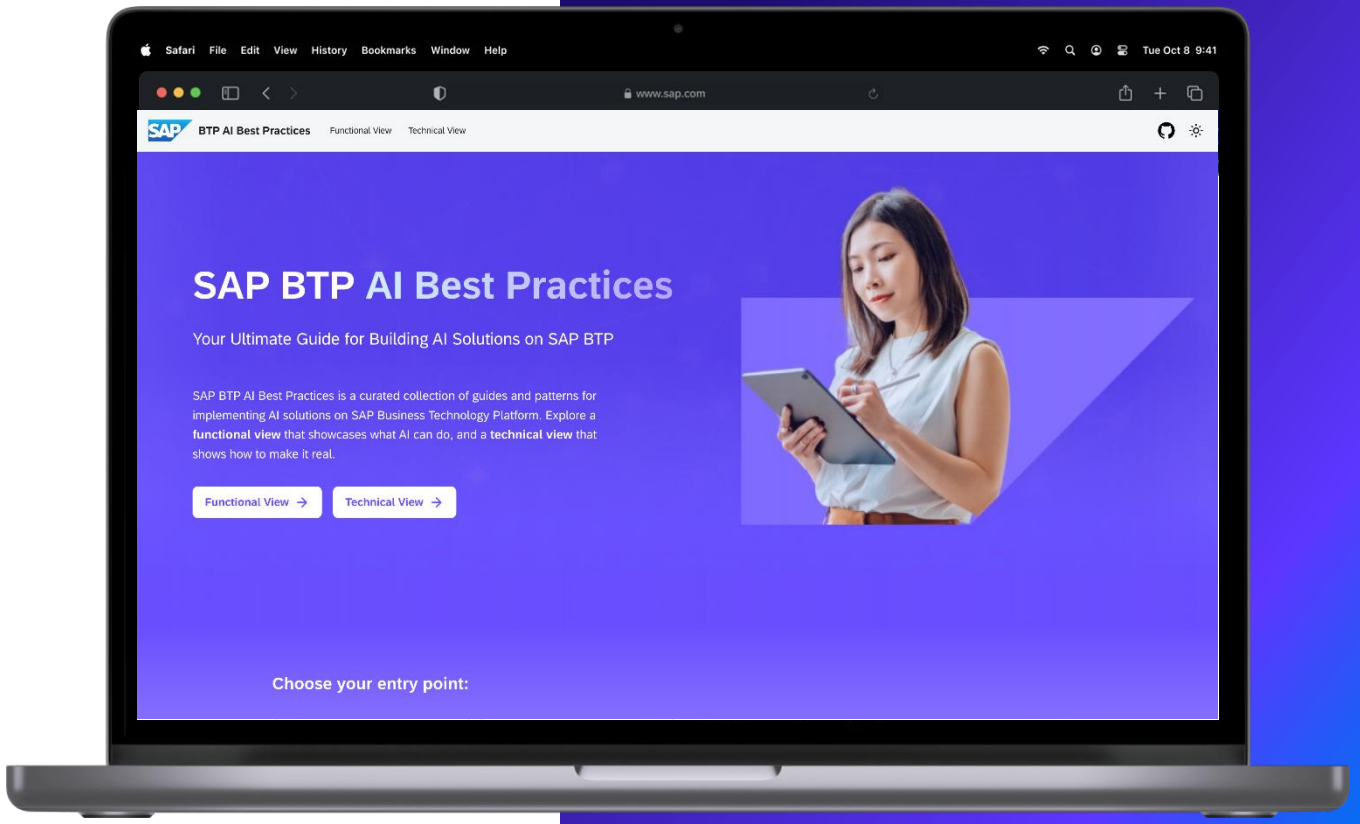
SAP BTP AI Best Practices

Clustering

Clustering aims to partition a dataset into subsets (clusters), such that data points within the same cluster exhibit high intra-cluster similarity, while points in different clusters exhibit low inter-cluster similarity.



BTP AI Services Center of Excellence

16.09.2025

Steps

1 Overview

2 Pre-requisites

3 Key Choices and Guidelines

4 Implementation

Clustering

Simple approach to explain phenomena.

Clustering aims to partition a dataset into subsets (clusters), such that data points within the same cluster exhibit high intra-cluster similarity, while points in different clusters exhibit low inter-cluster similarity.

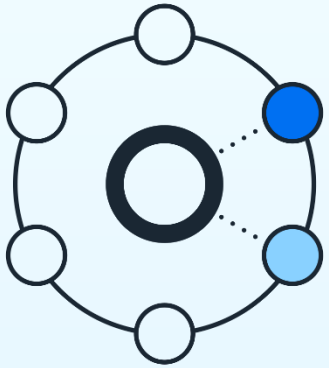
In the SAP ecosystem, this involves leveraging tools within SAP HANA ML (PAL, *hana_ml*) to partition dataset into groups that share similar characteristics

Expected Outcome

- The outcome of a clustering process is a set of groupings (called clusters) where each data point is assigned to one cluster based on similarity. The goal is to discover natural groupings in data without using labels.
- It helps to identify patterns, segment data, or detect anomalies

Key Benefits

Why use SAP HANA ML for Clustering?



Algorithm Interchangeability

Easily switch between algorithms (K-Means, DBSCAN, Agglomerate Hierarchical Clustering, Spectral clustering) to best suit your task.

Out-of-the-box Features

Supercharge your development with built-in capabilities Data Preprocessing Algorithms among many others.

Security & SAP Ecosystem

It's fully integrated into the SAP Ecosystem, leveraging the best of SAP technologies.

Pre-requisites

Supported Environments

- SAP HANA Platform 2.0 SPS 04 or higher
- SAP Hana Cloud (recommended for easier management)
- SAP Datasphere
- SAP Hana express edition (for development and testing)

Required Components

- Application Function Library (AFL) containing PAL and APL
- Script Server enabled for ML algorithms
- Required user authorizations and roles for PAL/APL

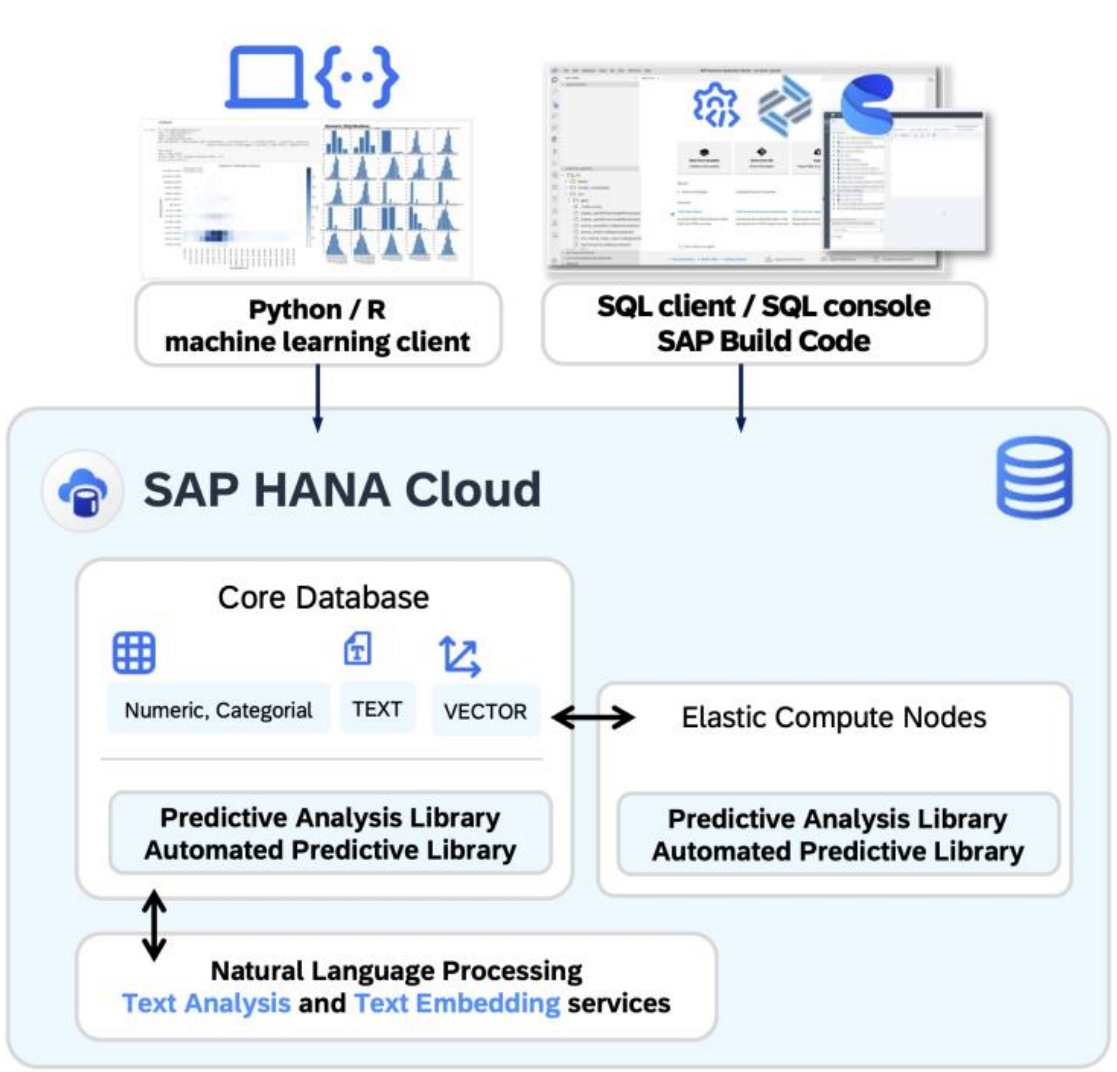
SAP HANA Platform

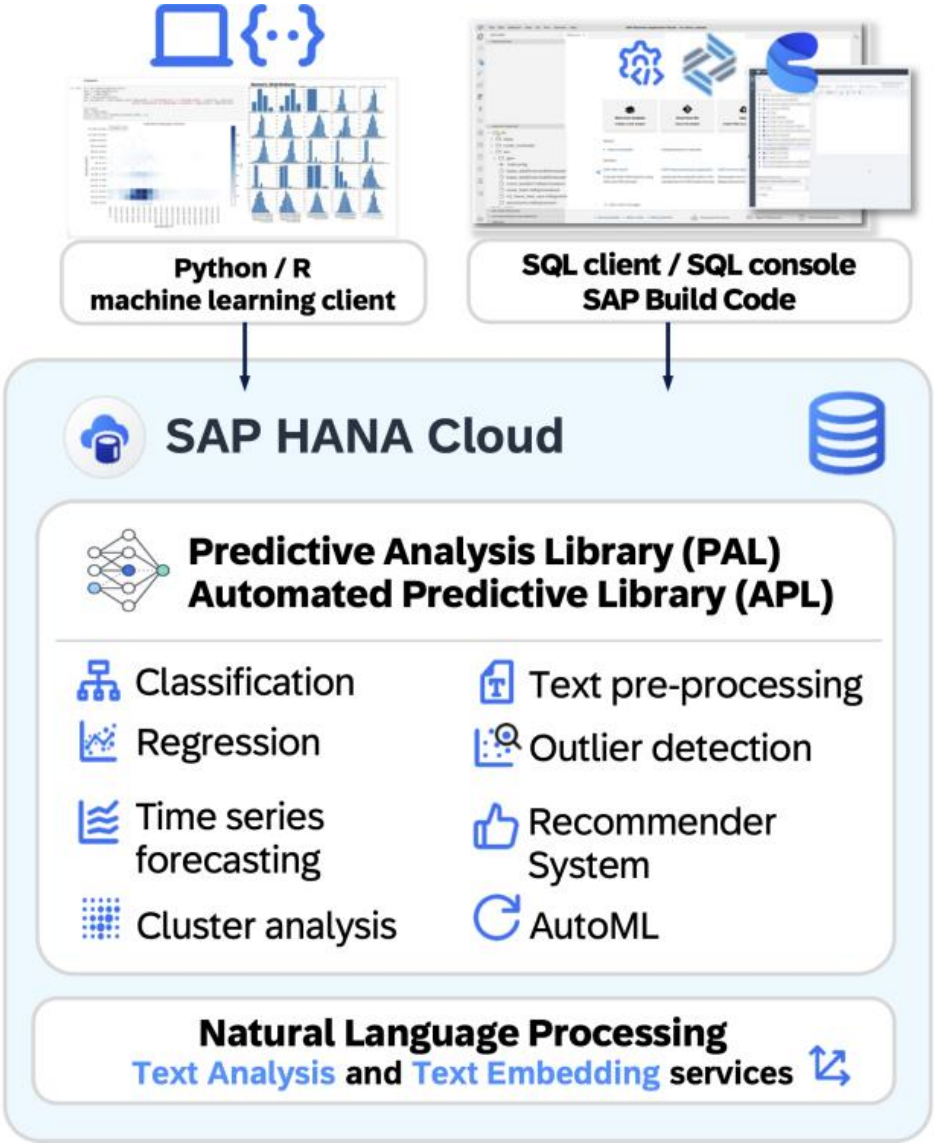
- SAP HANA is an in-memory database that enables real-time analytics and applications
- The HANA ML libraries (PAL, APL) provide native in-database functions for predictive analysis and Machine Learning.

Script Server

- Auxiliary SAP HANA process responsible for executing application function libraries
- Critical component that must be enabled for PAL and APL functionality
- Serves as execution environment for Machine Learning procedures

High-level reference architecture





Key Choices and Guidelines

Decisions that impact the performance and utility of the application

Data Preparation

Clear Null Policy

- Define a strategy for **handling missing values** (e.g., imputation, deletion).
- **Why**: Essential as many PAL procedures abort with Nulls, ensures data integrity
- **HANA ML Tool**: Use the Imputer (PAL) for various imputation strategies

Feature Importance Analysis

- Identify and select the **most relevant features** to distinguish anomalies
- **Why**: Reduces noise, combats the “curse of dimensionality”, improves model accuracy and interpretability
- **HANA ML Tool**: Use *FeatureSelection* or *permutation_importance* (PAL)

Dimensionality Reduction

- **Reduce features** using techniques like PCA while preserving key data variance
- **Why**: Reduces noise, combats the “curse of dimensionality”, improves model accuracy
- **HANA ML Tool**: Use *PCA* (PAL) for principal component analysis

Key Choices and Guidelines

Decisions that impact the performance and utility of the application

Algorithm Selection

Choosing the right clustering technique in SAP HANA PAL depends on several key factors related to both the characteristics of your data and the business problem you're trying to address

- K-Means algorithm partitions n observations or records into k clusters in which each observation belongs to the cluster with the nearest center.
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based data clustering algorithm. It finds a number of clusters starting from the estimated density distribution of corresponding nodes.
- Agglomerate Hierarchical Clustering is a widely used clustering method which can find natural groups within a set of data. The idea is to group the data into a hierarchy or a binary tree of the subgroups. A hierarchical clustering can be either agglomerate or divisive, depending on the method of hierarchical decomposition.
- Spectral clustering is an algorithm evolved from graph theory and has been widely used in clustering. Its main idea is to treat all data as points in space, which can be connected by edges. The edge weight between two points farther away is low, while the edge weight between two points closer is high.

Key Choices and Guidelines

Decisions that impact the performance and utility of the application

Model Evaluation

Model performance metrics are crucial for evaluating and comparing the effectiveness of machine learning models. They provide a quantitative measure of how well a model is performing, helping to identify areas for improvement and guide model selection. Common metrics specific to clustering tasks include the following:

- **Slight Silhouette:** Silhouette refers to a method used to validate the cluster of data. The complex of Silhouette is O(N2), where N represents the number of records. When N is very large, the silhouette costs a long time. Considering the efficiency, PAL provides a light version of silhouette called slight silhouette, where –1 stands for the poor clustering result, and 1 stands for the good result.

Implementation

Programming Model Selection Guidelines

Data Science Workflows

Utilize the **Python** hana_ml library for a streamlined, intuitive experience aligned with standard data science practices, including convenient data manipulation and integration with machine learning workflows

Alternative Approaches

Use **SQLScript** to directly call PAL procedures when tight integration with SAP HANA artifacts is needed, or the R interface via the external **SAP HANA R client**.

Python

SDK

- hana_ml

Reference Code

- SAP BTP AI Best Practices - Sample Code

Learning Journeys

- Developing AI Models with the Python Machine Learning Client for SAP HANA
- Outlier Detection by Clustering using Python Machine Learning Client for SAP HANA
- Model Storage with Python Machine Learning Client for SAP HANA

SQLScript

SDK

- SAP HANA Predictive Analysis Library (PAL)

Reference Code

- SAP BTP AI Best Practices - Sample Code

Learning Journeys

- SAP HANA PAL quick start

Code Sample

Python – Clustering

KMeans Training

Train a Kmeans on the Training Data

```
from hana_ml.algorithms.pal import clustering

features = ['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']
kmeans = clustering.KMeans(thread_ratio=0.2,
                           n_clusters=3,
                           distance_level='euclidean',
                           max_iter=100,
                           tol=1.0E-6,
                           category_weights=0.5,
                           normalization='min_max')
trained_km_model = kmeans.fit(data=df_iris_train, key = 'Id', features=features)
trained_df=trained_km_model.labels_.collect()
print(trained_km_model.labels_.collect())
```

[70] ✓ 4.4s

```
...       Id  CLUSTER_ID  DISTANCE  SLIGHT_SILHOUETTE
0      1           1  0.057713           0.930591
1      2           1  0.184429           0.763131
2      3           1  0.125252           0.848985
3      4           1  0.175024           0.783514
4      5           1  0.092754           0.892042
..   ...         ...       ...                ...
115  146           2  0.137110           0.710681
116  147           0  0.273207           0.150578
```

Contributors

Kute, Rajeshwari

Robledo, Francisco

Marques, Luis

Thank you