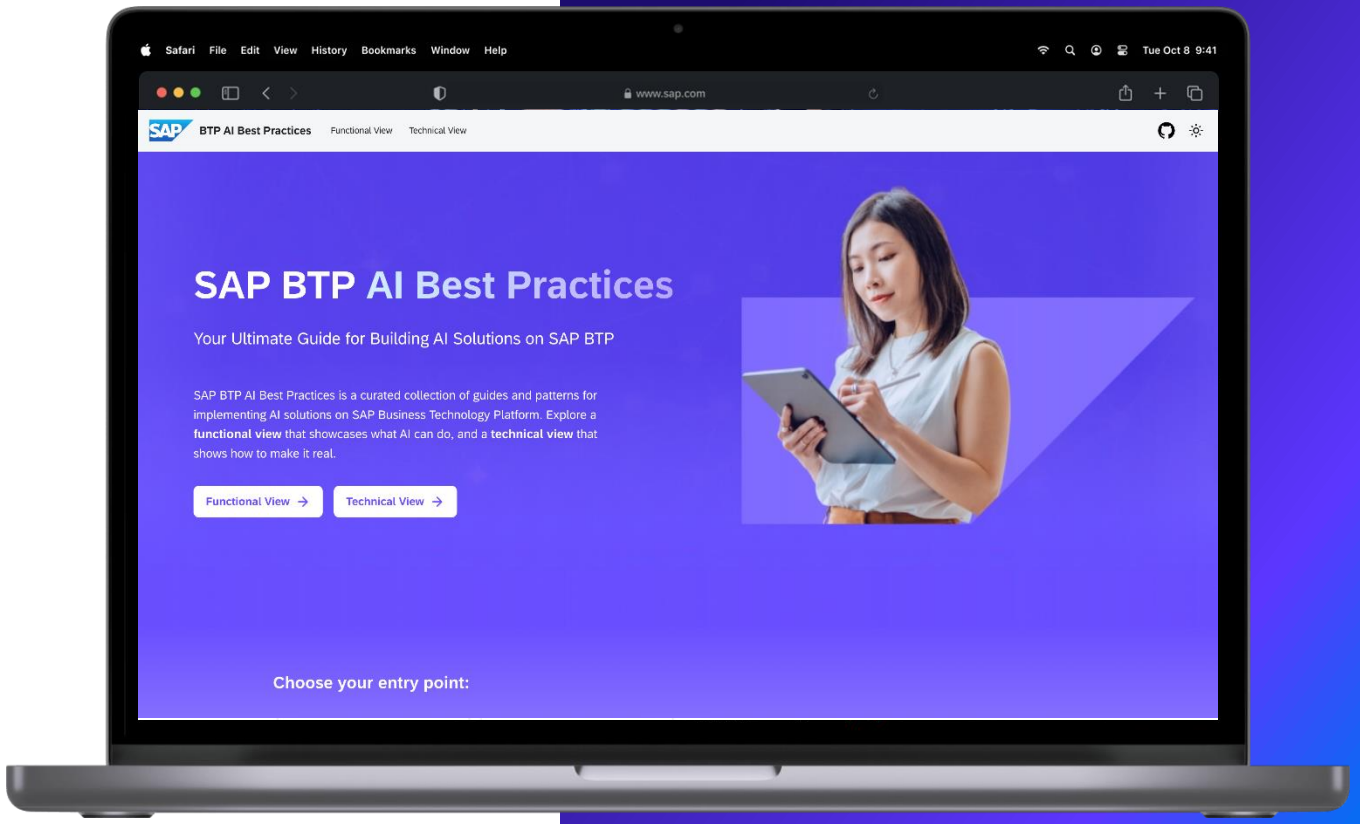
# SAP BTP AI Best Practices

## Regression

A powerful and simple approach to effectively deliver predictions quickly, well-suited for online settings, and allowing for a straightforward understanding of how the relationship between variables can be used to explain phenomena.




**BTP AI Services Center of Excellence**

**30.07.2025**

# Steps

1 Overview

2 Pre-requisites

3 Key Choices and Guidelines

4 Implementation

# Regression

Simple approach to explain phenomena.

Regression is the process of finding the best fit through a set of data points, essentially summarizing the relationship between two variables through **SAP HANA ML**

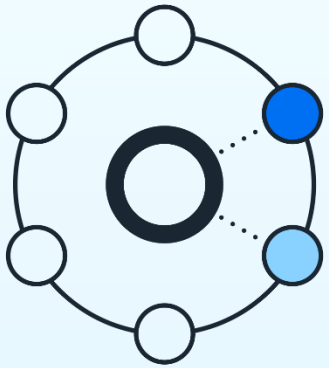
In the SAP ecosystem, this involves leveraging tools within SAP HANA ML (PAL, *hana_ml*) to uncover relationships between variables

## Expected Outcome

- To predict the value of the dependent variable based on the independent variable(s).
- Allowing for the identification of which independent variables are most influential in predicting the dependent variable.
- Optimize business processes by helping to identify factors that influence key performance indicators (KPIs), test hypotheses, and find optimal input values.

# Key Benefits

Why use SAP HANA ML for Regression?



## Algorithm Interchangeability

Easily switch between algorithms (Linear Regression, Online Linear Regression, Hybrid Gradient Boosting Regressor, RDT Regressor) to best suit your task.

## Out-of-the-box Features

Supercharge your development with built-in capabilities for Generalized linear models (GLM), decision tree-based modelling, 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

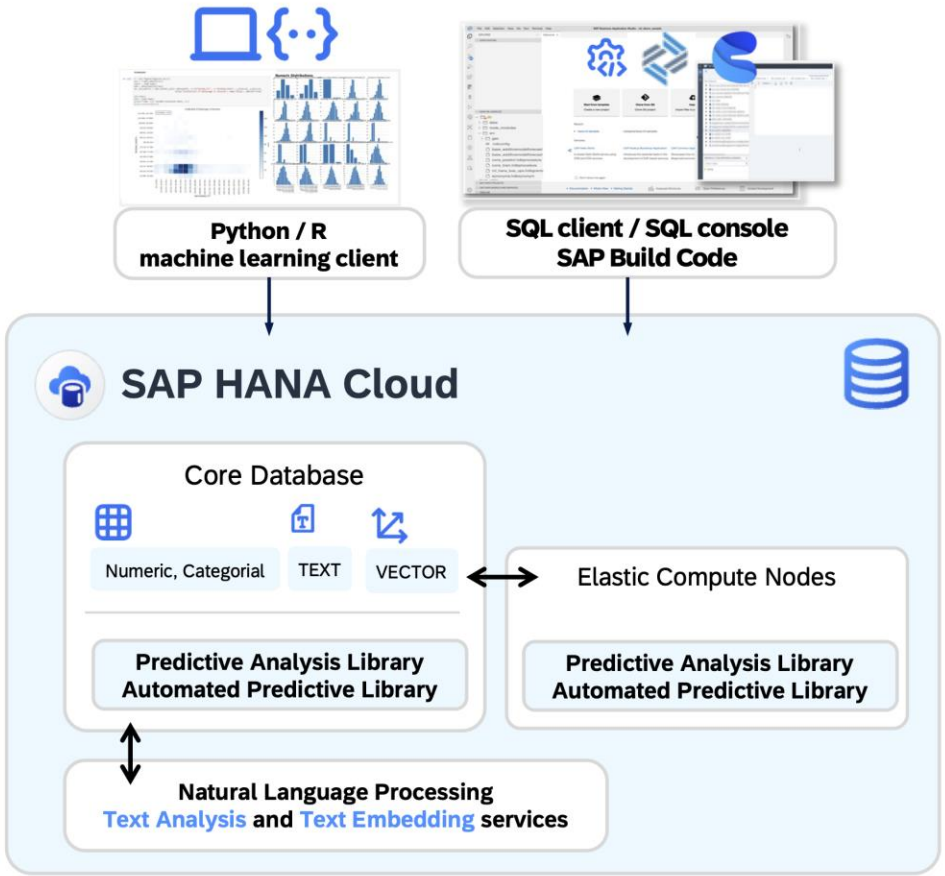
## SAP HANA Platform

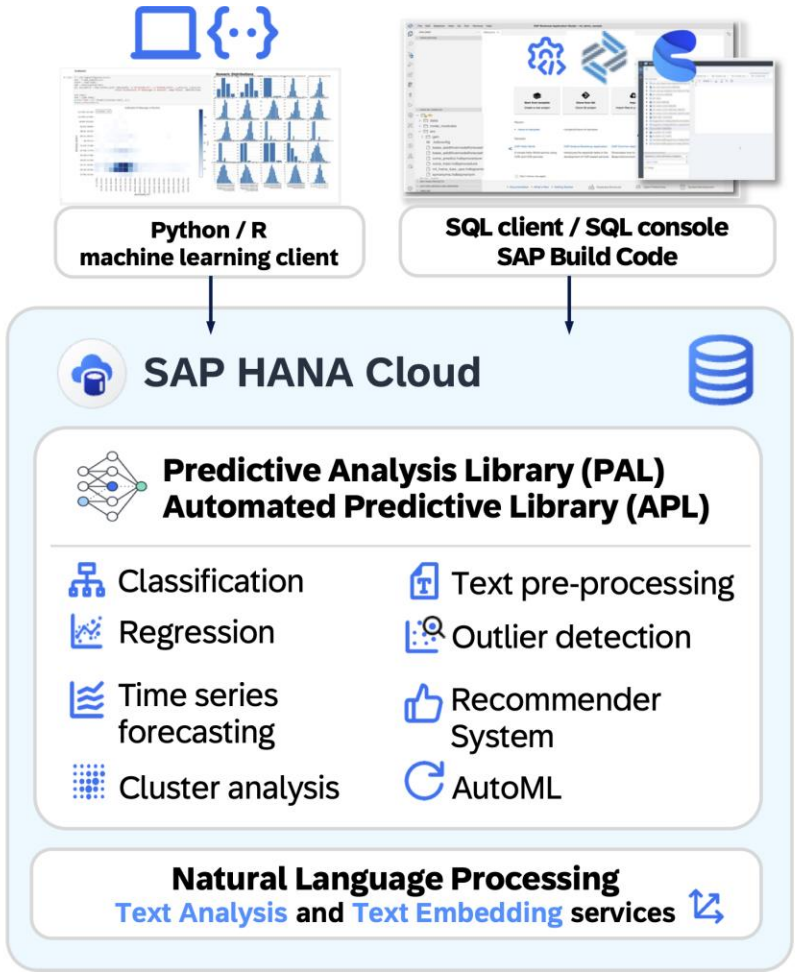
- SAP HANA is an in-memory database that enables real-time analytics and applications
- The HANA ML libraries (PAL, APL) provide native in-database functions for predictive analysis and Machine Learning.

## 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

## 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 regression tasks are the following:

  - **R-squared** is also known as **coefficient of determination and** denoted as **$R^2$** . It is basically the proportion of the variation in the response variable that is predictable from the independent variable(s). That is, the larger the value of $R^2$, the more variability is explained by the model. Typically, $R^2$ values vary from 0 to 1.

  - **The Root Mean Squared Error** is the square root of the **Mean Squared Error**. The **Mean Squared Error** is computed by taking the mean of the squared differences between each actual value and its corresponding predicted value. A **lower RMSE** value indicates better model performance, meaning the model's predictions are closer to the true values (ground truth).

  - When making decisions based on the forecast of regression models, **confidence intervals** may also be meaningful. A confidence interval is the mean of your estimate plus and minus the variation in that estimate; this is the range of values you expect your estimate to fall between, within a certain level of confidence. Additionally, PAL supports both confidence and prediction intervals for models like **Generalized Linear Models** (**GLM**), and **Multiple Linear Regression** (**MLR**), allowing users to assess not only the expected outcome, but also the uncertainty around individual predictions.

# Key Choices and Guidelines



Decisions that impact the performance and utility of the application

## Relevance

- Select independent variables that are logically related to the dependent variable and can contribute to the prediction. For example, if you're predicting sales, you might consider factors like advertising spend, market price, and competitor activity. Consider domain expertise when selecting features, as some features may be more important than others from a business or scientific perspective

# Key Choices and Guidelines



Decisions that impact the performance and utility of the application

## Data Quality

- Ensure independent variables are accurate, complete, and free from errors or outliers that could distort the analysis. Data cleaning, outlier removal, and normalization are crucial steps in preparing your data. Ensure sufficient data points, specially considering the number of independent variables and the noise level of your data. Validate imputation strategies and model fit using techniques like cross-validation and residual analysis.

# Key Choices and Guidelines



Decisions that impact the performance and utility of the application

## Feature Selection

- Consider techniques like feature importance analysis or model-based selection to identify the most relevant predictors. The former provides insights into which features are most influential in a model's output, allowing for better understanding of the data and model performance; several methods exist, including permutation importance, SHapley Additive exPlanations (SHAP) values, and built-in importance scores from specific models. For instance, **SHAP values** offer a way to understand feature contributions by considering all possible combinations of features and calculating their marginal contributions.

# 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.

## Python

**SDK**

- Hana_ml

**Reference Code**

- SAP BTP AI Best Practices - Sample Code

**Learning Journeys**

- ML- Linear Regression definition, implementation scenarios in HANA
- Introduction to Linear Regression with SAP HANA Studio and Automated Predictive Library (APL)

## SQLScript

**SDK**

- SAP HANA Predictive Analysis Library (PAL)

**Learning Journeys**

- SAP HANA PAL quick start
- SAP HANA PAL Multiple Linear Regression

# Code Sample

Python – Regression

```
%%time

from hana_ml.algorithms.pal.trees import HybridGradientBoostingRegressor

hgr = HybridGradientBoostingRegressor(
        n_estimators = 20, split_threshold=0.75,
        split_method = 'exact', learning_rate=0.3,
        max_depth=2,
        resampling_method = 'cv', fold_num=5,
        evaluation_metric = 'rmse', ref_metric=['mae'] )

hgr.fit(train_hdf, features=['ID', 'MedInc', 'HouseAge', 'AveRooms',
                             'AveBedrms', 'Population', 'AveOccup',
                             'Latitude', 'Longitude'], label='Target')
```

```
CPU times: total: 15.6 ms
Wall time: 3.34 s
```

# Contributors

PS Pacheco-Sanchez, Sergio

 Rzhaksynskyi, Andrii

RF Robledo, Francisco

 Marques, Luis

# Thank you