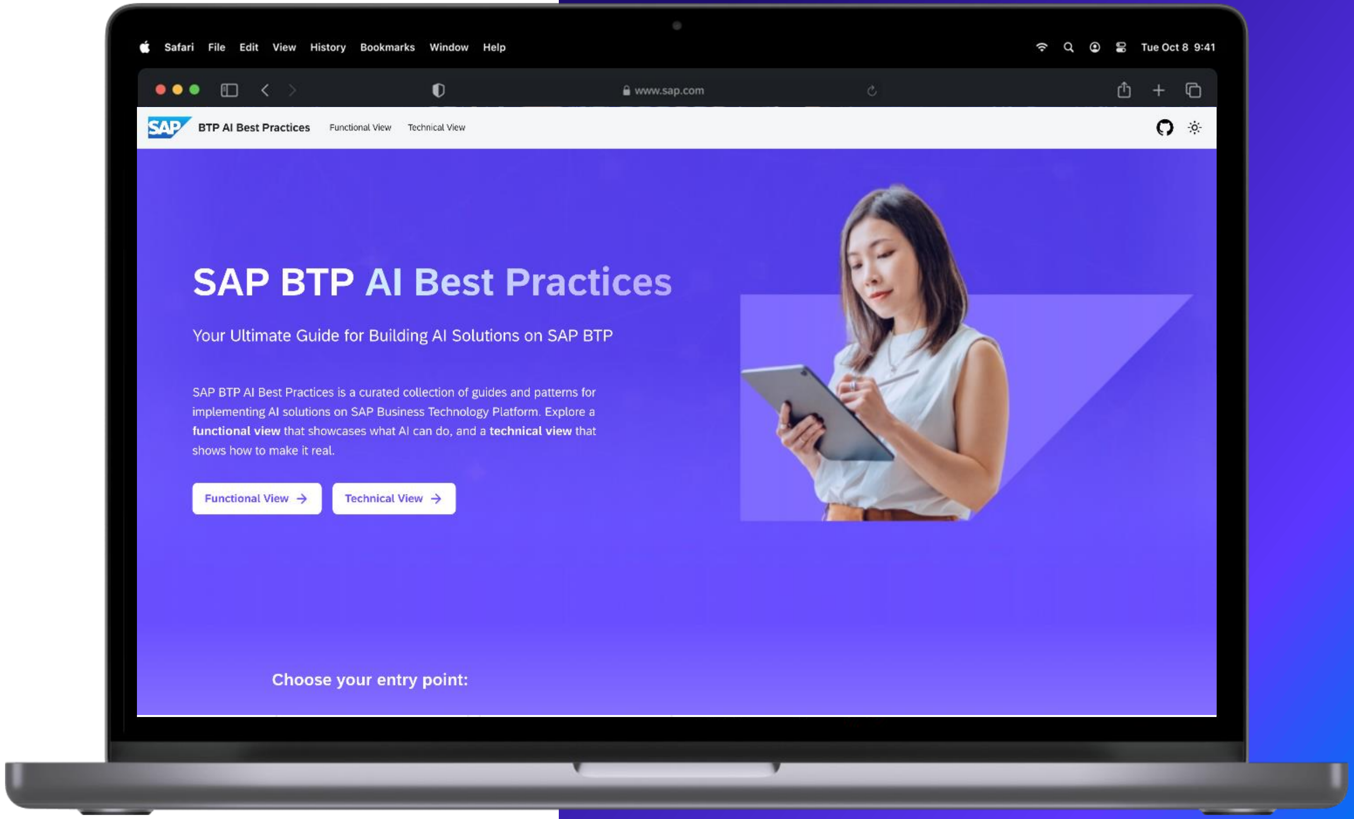
SAP BTP AI Best Practices

Time Series Forecasting

Time series forecasting is the process of analyzing time-ordered data points to predict future values. Time series data is a sequence of observations collected at successive points in time, typically at uniform intervals (e.g., hourly temperature readings, monthly sales).



BTP AI Services Center of Excellence

30.07.2025

Steps

1. Overview
2. Pre-requisites
3. Key Choices and Guidelines
4. Implementation

Time Series Forecasting

Simple approach to explain phenomena.

Time series forecasting is the process of using historical time-ordered data to predict future values. It involves identifying patterns like trends, seasonality, and cycles in data collected at regular intervals (e.g., daily, monthly).

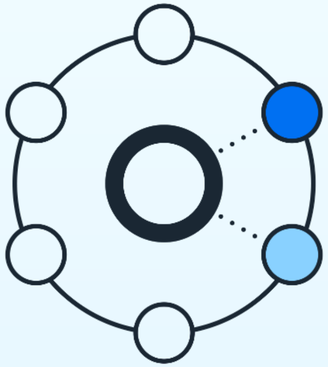
In the SAP ecosystem, this involves leveraging tools within SAP HANA ML (PAL, *hana_ml*) to predict future values using historical time-ordered data

Expected Outcome

- Ability to predict future values based on past observations.
- Support to identify patterns like trends, seasonality, and cycles in data collected at regular intervals
- Optimization of business processes operations by enabling data-driven planning and resource allocation

Key Benefits

Why use SAP HANA ML for Time Series Forecasting?



Algorithm Interchangeability

Easily switch between algorithms (ARIMA, Additive Model Time Series Analysis, Auto Exponential Smoothing, Bayesian Change Point Detection) to best suit your task.

Out-of-the-box Features

Supercharge your development with built-in capabilities for Seasonality Test, Stationarity Test 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

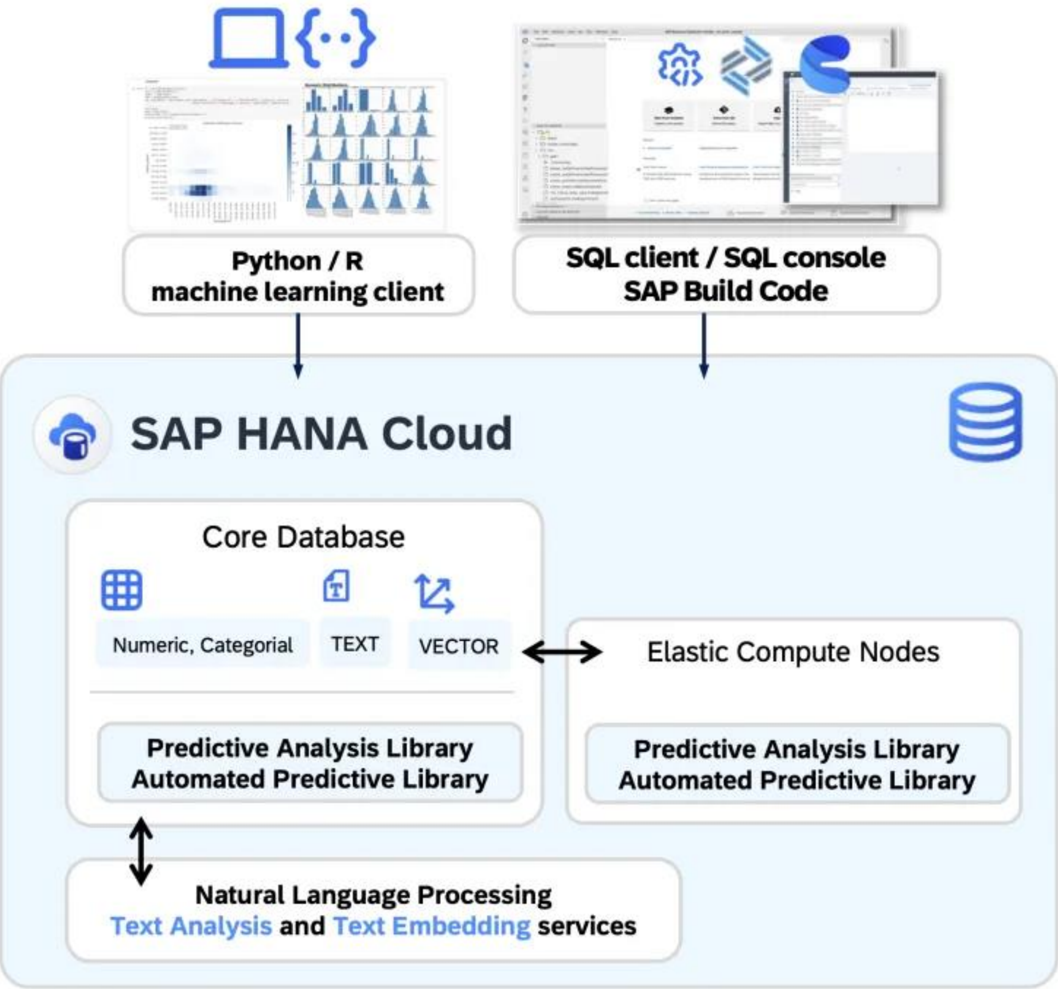
SAP HANA Platform

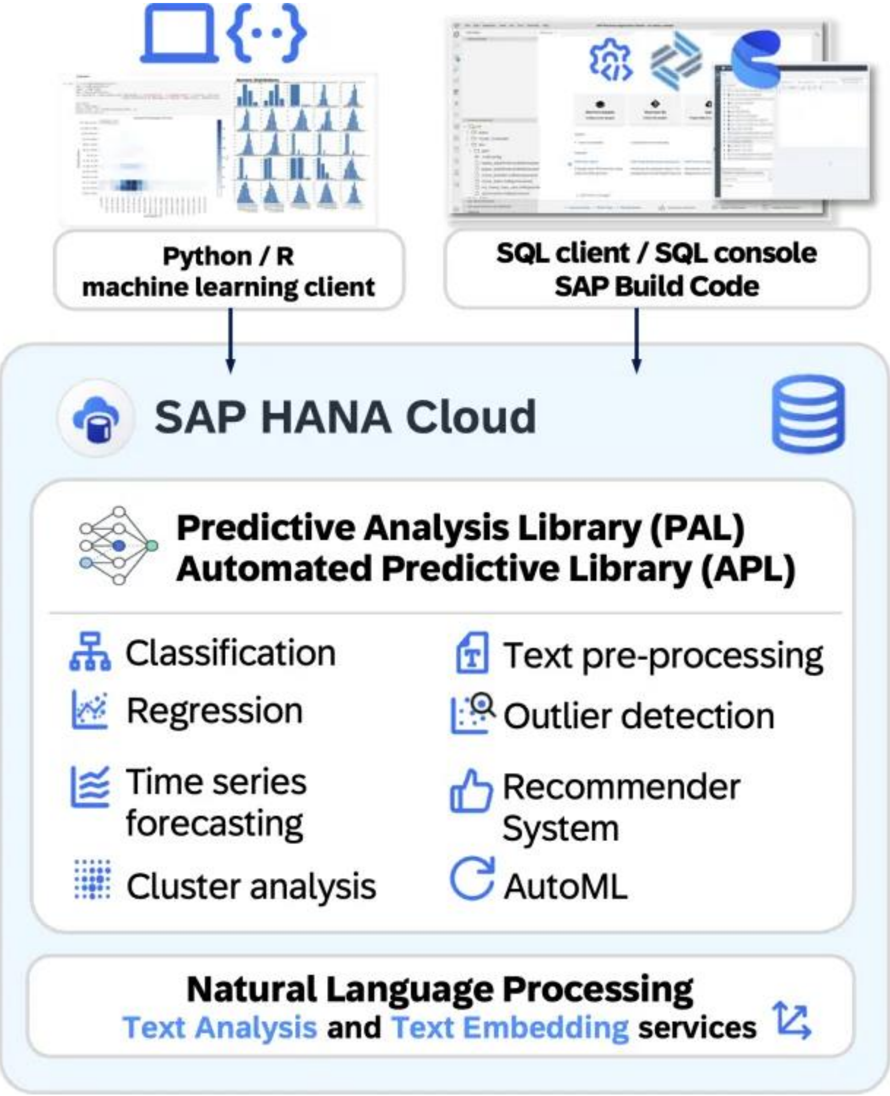
- SAP HANA is an in-memory database that enables real-time analytics and applications
- The HANA ML libraries (PAL, APL) provide native in-database functions for predictive analysis and Machine Learning.

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

Consistent Time Intervals

Time series models benefit greatly from evenly spaced data points. Check for and correct any missing timestamps or irregular intervals using techniques like interpolation or backfilling.

Resampling

You may need to resample or aggregate the data depending on the desired forecasting granularity (e.g., daily, weekly).

Stationarity

Many models (e.g., ARIMA) assume stationarity. Apply transformations like differencing or log transformations if the statistical properties of your series change over time.

Key Choices and Guidelines

Decisions that impact the performance and utility of the application

Algorithm Selection

Choosing the right time series forecasting model 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

- **Additive Model Time Series Analysis** uses an additive model to forecast time series data. It handles data with strong seasonal effects and is robust to changes in historical trends.
- **ARIMA (Auto Regressive Integrated Moving Average)** algorithm in SAP HANA PAL is a powerful statistical method used for forecasting univariate time series data.
- **Bayesian Change Point Detection** is a time series analysis technique used to detect points where the statistical properties of the data change, such as shifts in mean or variance.
- **Auto Exponential Smoothing** automatically calculates optimal parameters for Single, Double, and Triple Exponential Smoothing models. It forecasts based on these parameters by exploring the full parameter space and evaluating quality using MSE or MAPE, comparing historic and forecast values.
- **Massive, data-parallel (aka segmented) time series forecasting** independently apply Predictive Analysis Library (PAL) functions to subsets of data identified by a grouping column and Single call invoking massive data-parallel ML processing
- **AutoML for Time Series** AutoML framework implementation greatly help users in probing multiple algorithms and parameter variations for a given scenario for selecting the best possible algorithm and model.

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 Time Series Forecasting tasks are the following:

- **Expected MAPE (Mean Absolute Percentage Error):** The Expected MAPE evaluates the prediction error of a forecasting model over the forecast horizon. It is calculated as the mean of the absolute differences between actual and forecasted values, expressed as a percentage of the actual values. The lower the MAPE the better the model.

- **Expected MAE (Mean Absolute Error):** The Expected MAE evaluates the prediction error of a forecasting model over the forecast horizon. It is calculated as the mean of the absolute differences between actual and forecasted values, expressed in the same units as the actual values. A lower MAE indicates better model performance, and an Expected MAE of zero signifies a perfect forecast.

- **Expected MASE (Mean Absolute Scaled Error):** The Expected MASE (Mean Absolute Scaled Error) is the evaluation of the error made when using the predictive model to estimate the future values of the target, whatever the horizon. The lower the MASE, the better the model performance

- **Expected RMSE (Root Mean Squared Error):** The Expected RMSE evaluates the prediction error of a forecasting model over the forecast horizon. It is calculated as the square root of the mean of the squared differences between actual and forecasted values, expressed in the same units as the actual values. A lower RMSE indicates better model performance, with a value of zero representing a perfect forecast.

- **Expected R^2 (Coefficient of Determination):** The Expected R^2 evaluates the proportion of variation in the target variable that is explained by the predictive model over the forecast horizon. It measures how well the model captures the variability of the data, with a value of 1 indicating a perfect explanation of the variation. However, R^2 does not measure the accuracy or goodness of fit—a model can have a high R^2 while still being imprecise in predicting actual values.

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

- Identification of Seasonality in Time Series with Python Machine Learning Client for SAP HANA
- A Multivariate Time Series Modeling and Forecasting Guide with Python Machine Learning Client for SAP HANA
- Anomaly Detection in Time-Series using Seasonal Decomposition in 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 – Time Series Forecasting

```
# Create an HANA Dataframe for the actual series
from hana_ml import dataframe as hd
conn = hd.ConnectionContext(userkey='MLMDA_KEY')
series_in = conn.table('<table_name>', schema='<schema_name>')

# Preview Data
series_in.head(5).collect()

# Fit & Predict with APL
from hana_ml.algorithms.apl.time_series import AutoTimeSeries
apl_model = AutoTimeSeries(time_column_name= '<time_column>', target= '<target_column>', horizon= <forecast_horizon>)
series_out = apl_model.fit_predict(data = series_in, build_report=True)
df = series_out.select(series_out.columns[0:5]).collect()
dict = {'ACTUAL': 'Actual',
        'PREDICTED_1': 'Forecast',
        'LOWER_INT_95PCT': 'Lower Limit',
        'UPPER_INT_95PCT': 'Upper Limit' }
df.rename(columns=dict, inplace=True)
df

# Generate Report
apl_model.generate_html_report('TimeSeries_Report')
apl_model.generate_notebook_iframe_report()
```

Contributors

 Kute, Rajeshwari

RF Robledo, Francisco

 Marques, Luis

Thank you