HCL MaxAI Workbench User Guide

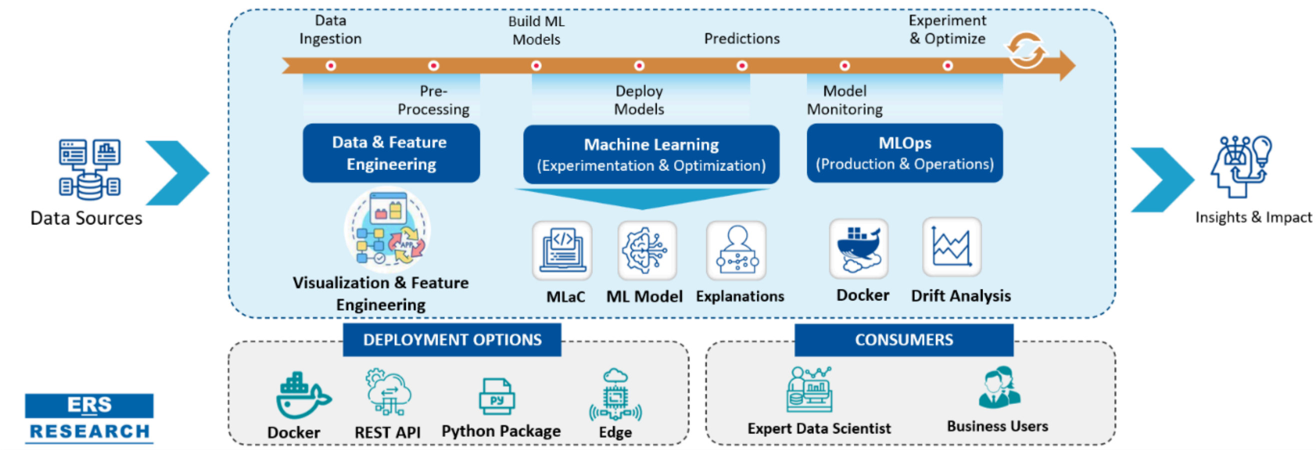
Contents

Chapter 1. Introduction

MaxAI Workbench is an AI life cycle management platform for applying machine learning to real-world problems. MaxAI Workbench encompasses the complete pipeline from raw data ingestion to a deployable machine learning model with a less-code/no-code approach.

It includes the following sub-processes:

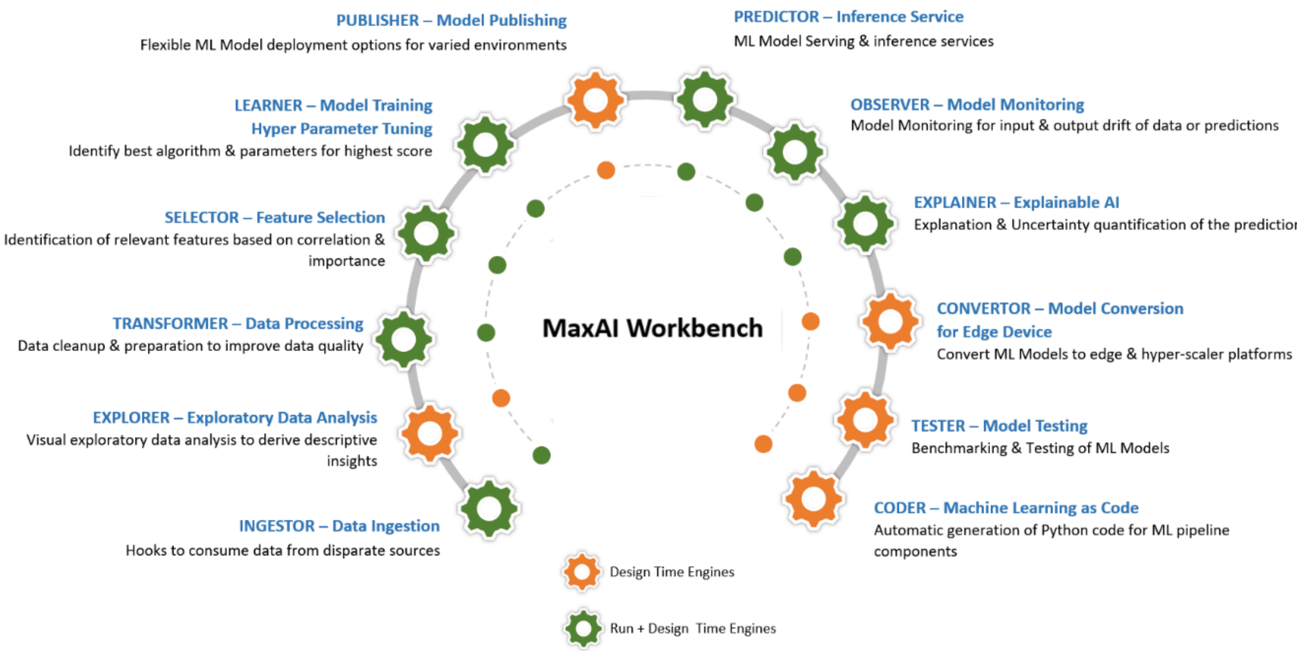
AutoML	Enable business user-generated ML models with no code approach
MLOPs	Model Versioning, Deployment & Drift Monitoring
MLac	Machine Learning as Code for automatic pipeline code generation
Explainability	Higher accuracy predictions with supported explanations and model confidence



In MaxAI Workbench, all these processes are automated. When you give input data and the use case that needs to be solved, MaxAI Workbench runs through all the stages of data analytics cycle and generates the best deployment model.

MaxAI Workbench Engines

MaxAI Workbench has a multi-stage pipeline as depicted below in the figure.



A brief explanation of each stage follows:

INGESTOR - Data Ingestion	In this stage, the dataset is uploaded in MaxAI Workbench GUI from disparate sources.
EXPLORER - Exploratory Data Analysis	This shows the details about the nature of data, the relation between features, model statistics, and performance information to derive descriptive insights.
TRANSFORMER - Data Processing	In this stage data-cleaning, data preparation, and outlier detectionares did automatically to improve the quality of data for better model accuracy.
SELECTOR - Feature Selection	The statistical analysis takes place to identify the relevant features for model training and remove the unimportant features based on correlation and importance.
LEARNER - Model Training Hyper Parameter Tuning	This stage trains configured models and selects the best parameters based on hyperparameter tuning (using which the models are trained). A broad spectrum of algorithms is supported.
PREDICTOR - Inference service	ML Model serving and inference services.
OBSERVER – Model Monitoring	It monitors the model for input and output drift of data or predictions.
CODER- Machine Learning as Code	It creates Python code automatically for ML pipeline components.

Accessing MaxAI Workbench

To access MaxAI Workbench, complete the following steps:

1. Log in to Unica.
2. Navigate to **MaxAI > MaxAI Workbench**.

   You can also acess MaxAI Workbench by:

   - Typing MaxAI Workbench or just Workbench in the Global Search bar and selecting the appropriate result that appears in the dynamic dropdown box.
   - Customizing the Favorites menu to include MaxAI Workbench and selecting the menu.

Chapter 2. Model Management

Creating Usecases

To create usecases, complete the following steps:

1. Select .
2. Click **+ Use Case**.

   Result

   The **NEW USE CASE** dialog appears.
3. Provide values for **Use Case Name** and **Description**. If required, you can retain the default **Use Case Name**.

   Note: Use case name does not support any special characters or space.

4. Click **Submit**.

   Result

   The **Use Case Name** appears in the **Use Case List** page.

Computing Infrastructure

Users can select the local machine as the compute infrastructure for training.

1. Select the **Compute Infrastructure** menu.
2. Select `Local`.

Viewing Existing Models

To view the existing models, complete the following steps:

1. Click .

   Result

   The existing use cases and their training status can be seen.
2. You can launch Trained models for Retaining, Prediction, Trusted AI parameters and Drift Analysis.
3. For multiple version of use case, click the use case name to view different model versions that were trained by Model Retraining. For example, by clicking `AI00053_lanquos`, you can view different versions under that Use Case.
4. You can perform the following operations on existing models:

Import	Import models that can be used for retraining.
Edit	Edit the model name and description.
Logs	View and download the model training logs for the model.

Delete	Delete the model.
Export	Exports the use case code which you can import into other workbench using the **Import** operation.

Publish Models

Trained model can be published in the form of following packages:

- MLaC
- Python Package
- Model Container
- Retrained Model

MLaC: Machine Learning as Code generates code automatically based on ML Operations performed during various ML Lifecycle. Using MLaC, expert data scientists can have better control over experimentation, optimization & deployment of ML Models.

MLaC has the following unique component:

1. **Generate Code**: Generates the code for each pipeline component of the selected use case i.e., data transformations, feature engineering, model training and prediction. This code can further be used to retrain the model on the latest data and can also be used to create a container for each component. The user will find the code in the local system through path: `C:\Users\yashaswini.ragi\AppData\Local\HCLT\AION\target\AION_375_1\publish\MLaC` .

Python Package: Python Package creates a WHL File of a trained model so that the user can further use that file in testing and prediction. If the user wants to consume the trained model for prediction in some system where MaxAI Workbench is not there, then either the user can install a python package or use the docker images.

Python package can be consumed by completing the following the steps:

1. Click on the python package icon from the packages tab and as a result whl file will get download.
2. Open the command prompt and type the command: `python -m pip install whl_file_path` and press `Enter`.

Model container: The model container creates a docker image of the trained model so that user can use that image anywhere, pull the image, and perform testing. If the user wants to consume the trained model for prediction purpose in some system where MaxAI Workbench is not there, then either the user can install it as python package or use the docker images.

There are two cases to create a docker container:

1. When the Docker is installed in the machine user can click on the docker and it gets created.
2. In case Docker is not installed, running MaxAI Workbench will throw an error. For that user can copy the mentioned path file and put in other machine where the docker should be installed and can build the image there also using docker build command.

The model container also supports CORS (Cross-Origin Resource Sharing) protocol to mitigate the default security policy, Same-Origin Policy (SOP) used by the browsers to protect resources. The same-origin policy is a critical security mechanism that restricts a documents or script loaded by one origin can interact with a resource from another origin.

Command to enable CORS: `python aion_service.py ip 0.0.0.0 -p 80 -cors “http://localhost:5000”`

Model Retraining

- After the Model got trained go to the Home.
- From the Train column click on the Model Re-training icon to re-train the model withing Re-Configuring.

  For Re-Configuring and training click on the Retrain icon.
- Click On EDA if required or click on Next.
- Set the Basic and Advance configuration if required.
- Click Train Model tab to train the model.
- Model will get Re-trained successfully and user can see the results.

Configuring System Settings

To configure and change system settings, complete the following steps:

1. In the left panel, click the Settings icon.
2. User can view the settings and configure them if required.
3. Configurations for the following integration entities are available:

    a. **Cloud Storage Settings**

        i. AWS S3 Bucket

    Credentials need to be given for accessing AWS by clicking on add bucket and data hence data can be uploaded from the cloud in GUI.

Chapter 3. Model Training

The Model Training tab is used to train the model based on the data input file.

Data Upload

Data upload tab is used to load the dataset in the GUI to train the model and get insight of it.

While uploading the dataset, data size and ram size need to be checked in two cases:

1. If file size is greater than the RAM size, file will not be uploaded for further process. File size should not be more than 80% of RAM size.
2. If the file size is greater than 50% of RAM size than the alert will be shown and proceed for the next step.

Uploading Data File

To upload the data file, complete the following steps:

1. Create use case.
2. Click the **Data Upload** tab.
3. File types like Delimited (CSV, TSV, and Excel) files, Parquet files, and Avro files can be uploaded (maximum 30 MB size) from local, NiFi, AWS S3 Bucket, and Graviton. Credentials might be required to ingest data form all the cloud storages.
4. As we **Submit**, the file will be uploaded successfully.
   Graviton: It enables implementation of the modern data platform using polyglot data technologies, open-source data frameworks and cloud-native services. It facilitates the development and delivery of optimal data products and efficient management of data product lifecycle from development to production.

Uploading a Document

This option is used to upload a document.

About this task

To upload a file, complete the following steps:

1. Click on the **Data Upload** tab.
2. Click **Unstructure data**.
    - Select the type of document you want to upload (TXT, LOG, PDF).
    - Place the files in a folder and specify the path in the **Folder Path** field.

   Note: Currently MaxAI Workbench does not support the trianing of use cases using unstructured data.

3. Click **Submit**.
   Result
   The files in the folder will be uploaded successfully.

Uploading a Python Script

The python script needs to be created by the user, which can be converted into a data file.

About this task

To upload a python script, complete the following steps:

1. Click the **Data Upload** tab.
2. Click **Custom Python Script**.
3. Specify the python script path.
4. Click **Submit**.

   Result

   The file will be uploaded successfully.

Uploading External Data Sources

Users can also access data from external databases for model training.

About this task

For uploading external data sources, complete the following steps:

1. Click the **Data Upload** tab.
2. Click **External Data Sources**.
3. Select different SQL databases and data warehouses to ingest data. For example, SQLite, Microsoft SQL, PostgreSQL, MySQL, Oracle, Google BigQuery, Redshift, Snowflake, Actian.
4. To access data from the database, you need to provide essential credentials like Database Name, Username, Host Port, etc.
5. If a single table contains all the data, you can simply put the table name and fetch the table as Data Frame. Otherwise, the user can perform join and selection based on the condition to merge more than one table and finally fetch the data.

   Note: Currently, this option is to upload data from some external databases and some data warehouses like-SQLite, PostgreSQL, etc. Users can insert a single table as a database. Also, some basic SQL operation like join, and selection based on where is also possible on multiple tables. Please contact Unica Admin to enable this option.

Uploading SKLearn Dataset

To upload a standard sklearn data set, complete the following steps:

1. Click on the **Data Upload** tab.
2. Click **SKLearn Data Sources**.
3. Select a dataset.

4. Click **Submit**.

   Result

   The dataset will be uploaded successfully.

Using EDA

EDA stands for Exploratory Data Analysis where user can analyze and investigate data sets and summarize their main characteristics, employing data visualization methods.

About this task

1. After uploading the data file, you can view the dataset of top 10 records.
2. To see the remaining details, click the **EDA** button.
3. On clicking the **EDA** button, a window will pop up that enables configuring the features and the sample of data that needed to be selected for EDA.
4. Select features and data subsampling size for exploratory data analysis.
5. Finally, click **Next**.
   - **Data Overview**: The summary statistics of the features.
   - **Data Distribution**: It shows the data distribution type for each feature of a dataset.

     For numerical data, select the numerical feature and click on **Show Data Distribution** to see the distribution of data.

     For text data, select the text feature and click on the **Show Data Distribution**. The word cloud is generated.

     More details can be seen by clicking the **?** icon on the right of the **Data Distribution** tab.
   - **Feature Importance**: The relevant features depending on the variance ratio can be viewed in the bar chart. The higher the variance ratio more relevant the feature is.
   - **Correlation Analysis**: The strength of relationships varying from 0 to 1 among various features. A value closer to 1 shows that the features are highly correlated and a value close to 0 shows that the features are less correlated.

     Example: The correlation value of feature row record_dateTime and feature column id is 1 which shows that these two features are highly correlated and considering both the feature is irrelevant.
   - **Unsupervised Clustering**: The unsupervised clustering of the data can be viewed.
   - **Data Deep Dive**: It offers an interface to inspect the correlation between data points for all the features of a dataset. Each item in the visualization represents the data point of an ingested dataset. Clicking on an individual item shows the key pairs that represent the features of that record whose values may be strings or numbers.

     Note: If the data ingestion is huge, Data Deep Dive may not be responsive, or data may fail to load.
   - **Pair Graph**: The variation of two selected features can be compared.

- **Fairness Metrics**: The bias identification of data can be viewed by the fairness metrics. There are three metrics- Statistical Parity Difference, Disparate Impact and Theil Index to check for unwanted bias in the dataset. Both are measures of discrimination (i.e., deviation from fairness).
- **Statistical Parity**: Measures the difference that the majority and protected classes receive a favorable outcome. This measure must be equal to 0 to be fair.
- **Disparate Impact**: Compares the proportion of individuals that receive a favorable outcome for two groups, a majority group, and a minority group. This measure must be equal to 1 to be fair.

  Select the feature (Categorical) and target feature and then click **Show Metrics** to analyze the potential bias between the selected feature and the target variable.
- **Theil Index**: Measures inequality among individuals in the group for a classification problem type. Ideally it should be zero for perfect fairness and biasness increase with increase in value greater than zero.

  Note: The acceptable threshold is highlighted in green. A class is considered biased if the metrics score is outside the threshold range.

6. Click **Download Reports** to download a summarized report of the EDA. The downloaded report is of Excel format.

Basic Configuration

This section provides the option to select more than one algorithm based on problem type and also provides the option to discard the unnecessary features from the dataset.

- Problem Description
- Model features
- Filter and Grouping

Problem Description

- Select the Problem Type. The following problem types are supported:
  1. `Classification`
  2. `Regression`
  3. `Anomaly Detection`
  4. `Time Series Anomaly Detection`
  5. `Clustering`
  6. `Time Series Forecasting`
  7. `Recommender System`

  Note: By default, classification is selected.

- Select the Scoring Criteria.

  Based on the selected Problem Type, select one of the following scoring criteria:

Problem Type	Scoring Criteria
Classification	Accuracy Precision Recall F1_Score ROC_AUC
Regression	R-Squared Mean Absolute Error Mean Squared Error Root Mean Sqaured Error
Time Series Forecasting	Mean Squared Error R-Squared Root Mean Sqaured Error Mean Absolute Error

- Based on the selected Problem Type, select one of the following Algorithms:

Problem Type	Algorithm
Classification	Extreme Gradient Boosting (XGBoost) Light Gradient Boosting (LightGBM) Categorical Boosting (CatBoost) Bagging (Ensemble) Random Forest
Regression	Extreme Gradient Boosting (XGBoost) Light Gradient Boosting (LightGBM) Categorical Boosting (CatBoost) Bagging (Ensemble)

Problem Type	Algorithm
	Random Forest
Time Series Forecasting	VAR - Vector Autoregressive LSTM - Long Short-Term Memory ARIMA - Autoregressive Integrated Moving Average Prophet MLP - Multi Layer Perceptron
Time Series Anomaly Detection	AutoEncoder DBScan
Clustering	KMEANS DBScan
Anomaly Detection	IsolationForest oneclassSVM DBScan
Recommender System	ItemRating

Model Features

This section provides the option to select features that will be used for training, options to change the data type and transform each selected feature.

Note: The target feature is not considered a training feature.

- In this section, the user can see the list of features with prefilled default values.
- Type of feature can be selected as date, index, categorical, numerical, or text.
- Six fill method is supported with the default value for numerical being Median and for Categorical is Mode.
- User can select different Encoding techniques for different features.
- Outlier and Normalization can be selected from the list of given options.
- Finally mention the Target Features (for supervised learning).

Row Filtering and Grouping

This section is used to filter certain rows from the dataset based on the requirement. If filtering is not required, this section can be skipped.

Clicking on the **Next** button will save the configuration and moves it to **Advanced Configuration** Tab.

Updating Advanced Configuration

To update the Advanced Configuration, complete the following steps:

1. Followed by data uploading and basic configuration click the **Advance Configuration** tab.
2. Click on plus (+) icon to expand the section and view the detailed configuration for that section.
3. By default, the fields are pre-filled. Change the fields if required.
4. Click **Next**.
   Result
   The advance configuration details are saved for training.

Training the Model

Model training tab gives the information about the Problem type, Selected algorithms, number of rows and Training Features, Training Data, Data balancing and Number of iterations.

About this task

The data balancing is applicable only for classification problem. The percentage of test data can be changed accordingly by the user. Number of iterations reflects the size of hyper-parameter space selected by the user.

1. Click **Train Model** tab to train the model.
2. Training can be aborted, if required, by clicking **Stop Training** tab.
3. Model training can also be performed by using API. For more information, see Fairness Metrics on page 18.

Details of the Trained Model

After successful training, the trained model details can be viewed.

Trained Model Info

It gives an information about- Training status, Deployment location of a model, problem type, best model (in case multiple algorithms is selected), Score of a best model, feature used, and the reports can be downloaded by clicking on **Download Reports** tab.

Evaluated Models

Evaluated models gives an information about the model performance based on testing score, Confidence Score and Feature Engineering Method. For classification and regression performance of seclected model is compared with the basic model such as logistic regression and linear regression respectively.

Example: When user selects Extreme Gradient Boosting (XGBoost) model for classification, its performance will be compared with Logistic regression.

Training Data Evaluation

It gives an information about training data.

Test Data Evaluation

It gives an information about test data.

Performance

Performance tab shows the graph of True Positive Rate vs False Positive Rate and Precision vs Recall, giving an idea about the model performance.

Performance Leaderboard

Performance leaderboard gives an idea about the best performing algorithm when multiple algorithms are selected based on its best score. It supports only machine learning algorithms for classification and regression problem types.

To view the leaderboard, click on the **Burger** icon from the right corner of the model training window. The Leaderboard appers.

Chapter 4. Model Prediction

Model prediction predicts the input data based on the pattern of training data.

Predict Instance

To execute Single Instance prediction, complete the following steps:

1. Select
2. In the **Predict** column, select .
3. In the **Predict Instance** tab, if required, change the values of the fields.
4. To obtain the Prediction result, click **Predict**.

Predict Batch

To execute prediction from the file, complete the following steps:

1. Select
2. In the **Predict** column, select .
3. Select the **Predict Batch** tab.
4. Upload the dataset.
5. To view the prediction result for the selected file, click **Submit**.

What to do next

You can also perform Prediction using API. For more information, see *MaxAI Workbench REST API Guide*.

Chapter 5. Trusted AI

This feature is available for classification and regression problem types.

Trusted AI provides an integrated approach to evaluate, quantify, and monitor the impact and trustworthiness of the models.

Fairness Metrics

Fairness metrics are a set of measures that enables the user to detect the presence of bias in model.

Theil Index	Measures inequality among individuals in the group for a classification problem type. Value of Theil Index is always greater than zero. Ideally it should be zero for perfect fairness and biasness increase with increase in value higher than zero.
Equal Opportunity Difference	Measures the difference in true positive rate between privileged and unprivileged group of protected features for a classification problem type. Ideally it should be zero to be fair. A value less than zero illustrates a higher benefit for the privileged group and positive value is a sign of higher benefit for the unprivileged group. Statistical Parity Difference measure the difference in the rate of favorable outcomes between privileged and unprivileged groups. This measure must be equal to 0 to be fair.

Viewing Fairness Metrics

To view the fairness metrics, complete the following steps:

1. Select 
2. In the **Predict** column, select .
3. From the **Check Bias For** dropdown box, select the protected feature. When training the model, the Target feature is automatically selected.
4. Select the graph type.

5. To view the graph, click the **Show Metric** tab.

> **Note:** The acceptable threshold is highlighted in green. A class is considered biased if metrics-score is outside the threshold range.

Performance

Model performance is an assessment of the model's ability to perform a task accurately not only with training data but also in real-time with runtime data when the model is deployed.

Matthews correlation coefficient (MCC) is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 an average random prediction and -1 an inverse prediction.

Transparency

Transparency helps users to understand the input data, algorithm and expected output of an ML model.

Brier Score measures the mean of squared difference between the predicted probability between the model and the ground truth values of a target feature.

Explainability

Explainability tab gives an explanation about Prediction and Model.

Xplain Prediction

About this task

Xplain Prediction explain the details about prediction concerning its Precision, confidence, reason, along with force plot and decision plot.

1. Select
2. In the **Predict** column, select .
3. Select the **Explainability** tab.
4. Select the **Xplain Prediction** tab.
5. Click the **Xplain** button.

   Result

   The detailed result appears.

Xplain Model

About this task

Xplain Model gives the insights of input data and explain the details of its important features.

1. Select
2. In the **Predict** column, select .
3. Select the **Explainability** tab.
4. Select the **Xplain Model** tab.
5. Click the **Get Model Details** button.
   Result
   The details of the model explanation appears. Scroll for detailed information.

Robustness

Robustness of AI/ML models is analyzed by Sensitivity Analysis examining the extent to which the results are affected by changes in input data.

There are two options to perform sensitivity analysis to understand the influencing factors. They are:

- First Order Sensitivity Analysis
- Total Order Sensitivity Analysis

First Order Sensitivity Analysis

First Order Sensitivity Analysis quantifies the impact of individual input variables on the output of a model by systematically varying one variable at a time while keeping others constant. It helps to understand the most influential factors of a model.

Total Order Sensitivity Analysis

Total Order Sensitivity Analysis evaluates the influence of individual input variables as well as their interactions on the output of a model. It measures the relative importance of the variable considering both their direct and indirect impact, providing understanding of their influence on the model prediction. It helps user to understand the most critical factors of a model.

Chapter 6. Model Monitoring

This feature is available for classification and regression problem types.

Model Monitoring monitors ML models for changes in data distribution. This ensures that the data is maintaining an acceptable level of variation.

You can use MaxAI Workbench to perform two types of drifts. They are:

- Input drift
- Performance drift

Accessing Model Monitoring

To access Model Monitoring, complete the following steps:

1. Select 
2. In the **Monitor** column, select the **Model Drift** icon.

Input Drift

Use Input Drift to understand the data distribution between the new dataset and the old dataset that are used to train the model. If the data maintains an acceptable level of variation, no further training is required to reuse the model. However, for an unacceptable range of variation in data distribution, you must retrain the model with the new dataset.

About this task

To monitor the training data distribution, complete the following steps:

1. Click **Input**.
2. Enter the current data file path.
3. Click **Submit**.
   Result
   The distribution details are shown.
4. Click **New data distribution**.
   Result
   The graph is shown.

Performance Drift

Use Performance drift to compare data distribution between the output data results and the label of input data providing details about the model performance.

About this task

To monitor the training data distribution, complete the following steps:

1. Click **Performance**.
2. Enter the current data file path from Local or URL.
3. Click **Submit**.

Chapter 7. Utility

For Data Generator and BYOM (Bring Your Own Model), select the **Utility** tab.

Data Generator

Data generation tool is used to generating artificial datasets with configurable size and complexity. The tool currently supports classification, regression and timeseries problem type.

Click **Data** > **Data generator** > **Fill the configuration based on problem type** which is available in the following sections:

Configuration of Classification data

number_samples	Number of samples (records) in dummy dataset (default is 100)
number_numerical_features	Number of numerical features in dummy dataset (default is 25)
number_categorical_features	Number of categorical features in dummy dataset (default is 2)
number_text_features	Number of text features in dummy dataset (default is 2)
missing_proportion	Create records with missing values based on a predefined proportion. For example: if there are 100 records in the dataset, and missing proportion is 0.1, number of missing records will be 100*0.1=10 (default is 0.1)
number_informative	Number of informative features. (Default is 20, i.e out of 25 total number features, only 20 features are useful of the model, and the other 5 are uninformative or redundant.)
number_class	Number of classes for classification problem (default is 2)
weights	The proportions of samples assigned to each class. If None, then classes are balanced. (Default is [0.4, 0.6] with number_class=2).
shift	Shift features by the specified value (default is 0.0). This value can be a float, or an array with a size equals to number of numerical features.
value_range_dict	Set the range of values of different features based on features id (default value is {“0”: [1,2]}, meaning that feature with id 0 has the range between 1 and 2. Users can set with multiple feature ids if needed.

Configuration of Regression data

number_samples	Number of samples (records) in dummy dataset (default=100)
number_numerical_features	Number of numerical features in dummy dataset (default=25)
number_categorical_features	Number of categorical features in dummy dataset (default=2)
number_text_feature	Number of text features in dummy dataset (default is 2)
missing_proportion	Create records with missing values based on a predefined proportion. (Default is 0.1 meaning that if there are 100 records in the dataset, and the missing proportion is 0.1, the number of records with missing value will be 100*0.1=10.)
number_informative	Number of informative features. (Default is 20, i.e out of 25 total number features, only 20 features are useful of the model, and the other 5 are uninformative or redundant.)
number_target	Number of target features
bias	The bias term (or offset/y-intercept) in the underlying linear model (default is 0.0, meaning that no bias term is added up).
noise	The standard deviation of the gaussian noise applied to the dataset (default is 0.0, meaning that there is no noise applied to the dataset).
value_range_dict	Set the range of values of different features based on features id (default value is {“0”: [1,2]}, meaning that feature with id 0 has the range between 1 and 2. Users can set up multiple feature ids if needed.

Configuration of Time series data

number_samples	Number of samples (records) in dummy dataset (default=100)
number_numerical_features	Number of numerical features in dummy dataset (default=25)
number_categorical_features	Number of categorical features in dummy dataset (default=2)
number_text_feature	Number of text features in dummy dataset (default is 2)

missing_proportion	Create records with missing values based on a predefined proportion. (Default is 0.1 meaning that if there are 100 records in the dataset, and the missing proportion is 0.1, the number of records with missing value will be 100*0.1=10.)
number_informative	Number of informative features. (Default is 20, i.e out of 25 total number features, only 20 features are useful of the model, and the other 5 are uninformative or redundant.)
number_target	Number of target features
bias	The bias term (or offset/y-intercept) in the underlying linear model (default is 0.0, meaning that no bias term is added up).
noise	The standard deviation of the gaussian noise applied to the dataset (default is 0.0, meaning that there is no noise applied to the dataset).
value_range_dict	Set the range of values of different features based on features id (default value is {“0”: [1,2]}, meaning that feature with id 0 has the range between 1 and 2. Users can set up multiple feature ids if needed.
Univariate	Set to True if the user wants to generate univariate time series data. Otherwise, set to False to generate multivariate time series data.
Start Time	Left bound for generating dates.
End Time	Right bound for generating dates.

BYOM (Bring Your Own Model)

You can bring your own trained model into MaxAI Workbench using `PKL` and `SAV` file types. This section provides step-by-step instructions to use the MaxAI Workbench web interface for generating information of your trained ML model.

Baselining and Testing and ML Model

1. Access the MaxAI Workbench Webpage and click MLTest.

   This starts the process baseline.

2. If you have downloaded the ML model file and the dataset file, copy-and-paste the files to the MaxAI Workbench user context.

Note: MaxAI Workbench does not identify any other user context path or root context path.

Example: `/home/aionuser/mountpoint/`

3. Provide the Model Path.

   Enter the path to your trained ML model file. The model should be saved in either joblib or pickle format.

   Example Model Path:

   `/home/aionuser/mountpoint/HCLT/data/AB-Test-2.joblib`

4. Provide the Data Path.

   Enter the path to your data file. The data can be either preprocessed (transformed) or raw.

   Example Data Path:

   `/home/aionuser/mountpoint/HCLT/data/stars.csv`

5. Select the Training Features.

   Select the training features used for training the model. Ensure that the features you select match those used during the model training to align the data correctly with the model's expectations.

6. Select the Target Feature.

   Specify the target feature in your dataset that the model aims to predict.

7. Submit the Form.

   After filling in the required information, click the **Generate Baseline** button. The system will process the inputs and generate the baseline report.

8. View and Save the Results.

   The results will be displayed on the page, including metrics such as the model name, parameters, model performance, and time latency. If needed, you can save the results to a file for future reference.

Chapter 8. Reference Models

In version 25.1.0, MaxAI Workbench provides reference models for use with products like Unica Campaign, Unica Journey, HCL CDP, etc.

The models are:

- Next Best Channel (NBC)
- Send Time Optimization (STO)
- Churn Prediction
- Buying Propensity

Next Best Channel (NBC) Model

The Next Best Channel (NBC) model identifies the most effective communication channel for each customer based on demographics, banking behavior, device preferences, and previous marketing interactions. It helps optimize customer engagement and improve conversion rates across channels like WhatsApp, Email, SMS, IVR, Push Notifications, and more.

MaxAI helps interpret the model's predictions. This includes:

SHAP (SHapley Additive exPlanations)	To identify how each feature contributed to the predicted channel.
Force and Waterfall Plots	For individual predictions – showing positive and negative feature impact.
Bar Charts of Feature Importance	To understand which features globally influence channel selection across customers.

These visualizations are available via Single Prediction Explanation Report (HTML) and Batch Prediction Explanation Report (PDF).

Key Features

- **Channel Affinity Prediction**

  AI-driven model analyzes demographics, banking behavior, device preferences, and campaign history.

- **Dynamic Channel Selection**

  Automatically selects from Email, PUSH, RCS, SMS, and WhatsApp.

- **Personalized Engagement**

  Tailors outreach strategy per customer.

This model will be used as is with HCL CDP. For more details related to integration and using the models, see the chapter *MaxAI Integration* and its sub-topics, within the *HCL CDP Integrations Guide*.

Send Time Optimization (STO) Model

The Send Time Optimization (STO) Model enhances customer engagement and boost conversion rates by predicting the optimal time and channel to send marketing communications for each customer. By analyzing customer demographics, behaviors, device preferences, and historical engagement patterns, the model personalizes marketing strategies for improving open rates, click-through rates, conversions, and marketing ROI.

Key Features

- **Time-based Personalization**

  Predicts the optimal send time for each user based on historical engagement.
- **Performance-driven**

  Improves open rates, click-throughs, and conversions.
- **Automated Scheduling**

  Outbound messages are sent at the time users are most likely to engage.

This model will be used as is with HCL CDP. For more details related to integration and using the models, see the chapter *MaxAI Integration* and its sub-topics, within the *HCL CDP Integrations Guide*.

Churn Prediction Model

A churn prediction model identifies the likelihood that a customer will stop using a company's products or services within a given period. The objective is to proactively detect customers at high risk of churn so that timely retention actions can be taken.

This model leverages customer demographics, product holdings, transaction patterns, engagement behaviour, service interactions, and historical activity data. The target variable (target) is binary:

1 indicates the customer has churned (or is likely to churn), and

0 indicates the customer remains active.

The Churn Prediction model aims to forecast the likelihood of an existing customer discontinuing or reducing their relationship with the bank or financial institution. By accurately identifying customers at risk of churn, the model enables proactive retention efforts, strengthens customer loyalty, and minimizes revenue loss.

Fr example, assume a bank wanting to retain every valuable customer before the customer decides to leave. Using a churn prediction model, it can pinpoint who is most likely to churn and why, allowing the bank to take timely, personalized actions such as targeted offers, service improvements, or engagement campaigns. This not only reduces attrition but also enhances customer satisfaction and long-term profitability.

Buying Propensity Model

A product propensity model predicts the likelihood that a customer, existing or new, will adopt or purchase a specific product in the near future. Predicting which product, a customer is likely to take next.

This model leverages customer demographics, product holdings, behavioural data, and historical campaign interactions. The target variable (target) is binary:

1 indicates conversion/engagement.

0 indicates no conversion.

The Product Propensity Model aims to predict the likelihood of an existing or prospective customer purchasing or adopting a specific banking or financial product (for example, credit card, personal loan, savings account upgrade, insurance). This helps prioritize offers, personalize campaigns, and optimize sales channel efficiency.

Example: If a bank wants to offer the perfect product to each customer, using a propensity model, it can predict who is most likely to respond, ensuring every campaign reaches the right person. This not only boosts conversions but also creates a personalized experience for every customer.

Features

Customer Profile (Demographics and Static Attributes)

Demographics and profile attributes often influence product adoption behaviour. For example, younger customers may prefer digital products, while older customers may prefer traditional loans. Geography and device can also affect accessibility and channel effectiveness.

Product Ownership / Portfolio

Existing product portfolio is highly predictive of future product adoption. Cross-sell/up-sell opportunities depend on what the customer already holds or has closed in the past (e.g., customers with savings accounts are more likely to adopt credit cards).

Engagement & Campaign Exposure

Measures historical exposure and responsiveness to marketing campaigns. Customers with higher campaign exposure and recent interactions are more likely to show interest in new offers.

Channel Engagement & Response Behaviour

Captures channel-level effectiveness and customer preferences. For example, customers with high WhatsApp accept rate are strong candidates for future WhatsApp-driven offers. Balancing open and accept rates helps identify not just engagement but also intent.