Oracle Warehouse Management Cloud

AIML Guide

Release 26A

Oracle Warehouse Management Cloud
AIML Guide

Release 26A

G47895-01

Author: Oracle WMS Cloud Product Team

Contents

Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

Get Help in the Applications

Some application pages have help icons (?) to give you access to contextual help. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons. If the page has contextual help, help icons will appear.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at *Oracle University*.

Join Our Community

Use *Cloud Customer Connect* to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest *ideas* for product enhancements, and watch events.

Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to *oracle_fusion_applications_help_ww_grp@oracle.com*.

Thanks for helping us improve our user assistance!

1 Oracle WMS Machine Learning

WMS Machine Learning

Accurately predicting how long it takes to process an order, from start to finish, is a growing challenge in today's fast-paced, data-driven environment. Oracle WMS now includes integrated AI and machine learning (ML) capabilities that help solve this problem by analyzing your historical order data and generating smart, reliable predictions.

We currently support machine learning models for:

- **Predicting Order Cycle Time, Waiting Time, and Processing Time**
- **Intelligent Cycle Counting**
- **Market Basket Analysis**

These capabilities help you optimize planning, reduce operational delays, and better manage customer expectations by using insights drawn directly from your own data.

Predicting Order Cycle Time, Processing Time, and Waiting Time

In modern supply chain environments, **predictability is power**. Warehouses often operate with limited visibility into how long an order will take to move from receipt to shipment. This lack of foresight can create planning inefficiencies, staffing issues, customer dissatisfaction, and missed service level agreements (SLAs).

By leveraging **AI/ML-driven predictions for Order Cycle Time, Processing Time, and Waiting Time**, businesses can make smarter, data-backed decisions that drive operational efficiency and customer satisfaction.

Use Case Scenario

A high-volume distribution center is preparing for a seasonal surge in order volume. Operations managers need to forecast how long orders will take to complete, not just on average, but across different order types, times of day, and warehouse zones.

Using Oracle WMS's predictive capabilities:

- **Order Cycle Time** predictions help determine when specific orders are likely to leave the facility, which enables accurate delivery promises and staffing adjustments.
- **Processing Time** forecasts highlight potential internal delays (e.g., picking, packing) and allow managers to preemptively redistribute labor to keep workflows balanced.
- **Waiting Time** predictions identify bottlenecks, such as staging or dock congestion, before they occur, providing the ability to adjust load schedules or sequence orders more efficiently.

Business Value

- **Proactive Labor Planning**: Allocate resources to high-delay areas before issues arise.
- **Customer Experience**: Provide accurate delivery estimates and improve on-time performance.
- **Throughput Optimization**: Identify and eliminate workflow bottlenecks in real-time.
- **SLA & Carrier Compliance**: Minimize late shipments by predicting orders at risk and taking action earlier.
- **Cost Reduction**: Avoid overstaffing and overtime by aligning labor with actual processing needs.

By embedding predictive intelligence into core WMS workflows, businesses can shift from reactive firefighting to proactive decision-making.

Intelligent Cycle Counting

Maintaining accurate inventory is foundational to successful warehouse operations, however, traditional cycle counting methods are time-consuming, labor-intensive, and often lack strategic prioritization. Many warehouses rely on fixed schedules, ABC classifications, or random sampling, which can lead to overcounting low-risk items and undercounting high-risk or high-impact SKUs.

Intelligent Cycle Counting, powered by AI/ML in Oracle WMS, transforms this process by using data-driven predictions to determine **which items are most likely to be inaccurate** AND prioritizes those for counting.

Use Case Scenario

A warehouse handles thousands of SKUs with varying turnover rates, locations, and movement histories. Rather than using static ABC logic, the team enables Intelligent Cycle Counting to dynamically identify inventory at higher risk of inaccuracy such as:

- Items with frequent adjustments or discrepancies
- SKUs recently involved in multiple picks or returns
- Inventory stored in high-traffic or hard-to-reach locations

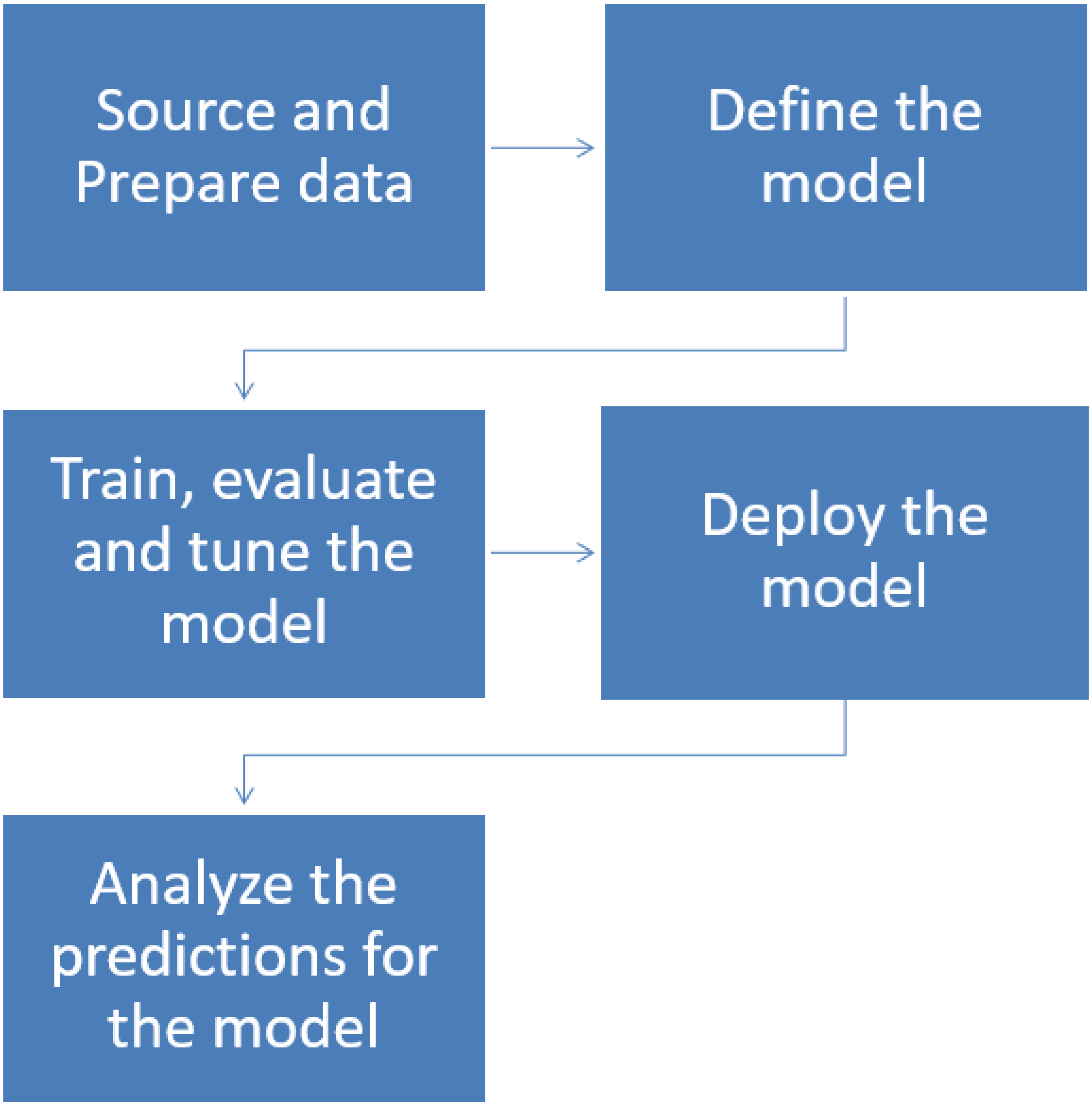
By running a scheduled AI/ML job, the WMS will recommend a prioritized list of items to countbased on real-time inventory behavior, historical accuracy trends, and movement data.

Business Value

- **Higher Accuracy with Less Effort**: Count fewer items while improving overall inventory accuracy.
- **Data-Driven Prioritization**: Focus counting efforts where risk is highest, not just based on item value.
- **Labor Optimization**: Reduce unnecessary cycle counts, freeing up staff for other critical tasks.
- **Shrinkage Reduction**: Catch and correct inventory issues early
- **Audit Confidence**: Strengthen audit readiness with continuous, intelligent verification.

Intelligent Cycle Counting shifts inventory management from **scheduled routines** to **smart, predictive actions**, helping businesses stay lean, accurate, and responsive without adding manual overhead.

Machine Learning Process Flow



Source and Prepare data

Define the model

Train, evaluate and tune the model

Deploy the model

Analyze the predictions for the model

Types of Algorithms Available in WMS

Oracle WMS currently supports a variety of **machine learning algorithms**, each designed to handle specific types of prediction problems. While each algorithm requires different parameters and follows a unique training methodology, many of them can be applied across multiple supported metrics.

These algorithms form the foundation for predictive capabilities such as forecasting **Order Cycle Time**, **Waiting Time**, **Processing Time**, and powering features like **Intelligent Cycle Counting** and **Market Basket Analysis**.

Random Forest

Random Forest is a widely used algorithm for solving both **regression** and **classification** problems (see glossary for definitions). It operates by building multiple decision trees and combining their outputs for improved accuracy and robustness.

Key Parameters:

- `n_estimators`

  The number of decision trees in the model. More trees can increase accuracy but also add processing time.

  - **Default**: 100
  - **Allowed Range**: 1–200

- `max_depth`

  Specifies how deep each decision tree can grow. Deeper trees can capture more complexity in the data but also risk overfitting.

  - **Allowed Range**: 1–20
  - > **Note:** No default value is set. Entering a value outside the range will trigger an error.

- `max_depth`

  - `This value should not be changed.`

    Training Tip: Run several iterations to train and evaluate the model. This helps fine-tune parameters and improves overall accuracy by capturing variations in results.

Feed Forward Neural Networks

Feed Forward Neural Networks are inspired by the human brain and excel at tasks that involve **pattern recognition** such as identifying images, text, or sounds. In WMS, they are typically used for **classification** and **clustering** tasks.

Key Parameters:

- `hidden_layer_sizes`

  Determines the architecture of the network—the number of hidden layers and the number of neurons (nodes) in each.

- `max_iter`

  Indicates the number of training iterations, or **epochs**.

Epoch: One complete pass through the entire training dataset.

Gradient Boosting

Gradient Boosting is a powerful and flexible algorithm that can be used for both **classification** and **regression**. It builds models in a sequential way, where each new model corrects errors made by the previous one.

Key Parameters:

- `max_iter`

  The number of boosting stages or epochs to run.

- `max_depth`

  Controls the depth of individual decision trees (same behavior as in Random Forest).

- `learning_rate`

  Determines how quickly the model adapts to the data.
  - A **high learning rate** may overshoot the optimal solution.
  - A **low learning rate** may take longer but offers more control and stability.

> **Note:** Results will vary based on the model and parameters given.

Creating a Machine Learning Model in WMS

To build and use a machine learning model in WMS, follow the structured process below:

Step 1: Create a Scheduled Job

Initiate the process by scheduling a job to prepare the necessary training data.

Step 2: Review the Training Data

Use the **AIML Training Data** screen to review and validate the dataset generated by the scheduled job.

Step 3: Define the Model

Configure your model using parameters and filters. Assign the model to an **AI/ML Training Template**, where you can also set a default template for reuse. This can be done in AIML Training Template UI.

Step 4: Run Predictions

Execute the model using the **AIML Prediction Run UI** to generate predictions. The system will capture and display the results from this run.

Step 5: Deploy and Refine the Model

Deploy the model to begin using it in production. You can also retrain and adjust the model as needed based on results and performance insights.

Available AI/ML Screens in WMS

- **AIML Training Data**: View the structured training dataset prepared via scheduled job.
- **AIML Training Template**: Create, configure, and manage training templates; set defaults for ease of reuse.
- **AIML Training Date Rule Hdr: Create custom seasonality based date ranged as need for you warehouse needs. This only applies for Market Basket Analysis.**
- **AIML Models UI**: View models created from your training templates, including parameters and status.
- **AIML Model's Training Logs UI: View relevant logging information related to model creation.**
- **AIML Prediction Run UI**: Run models and view detailed prediction results.
- **AIML Predictive Dashboard**: Visualize recent prediction outcomes (currently supports Order Cycle Time only).

Step One: Create a Scheduled Job

Each metric is configured to work with specific AI/ML algorithms and defines the type of input data required, as well as the expected output.

Currently, we support AI/ML functionality for the following metrics:

- **Order Cycle Time**
- **Order Waiting Time**
- **Order Picking Time**
- **Intelligent Cycle Count**

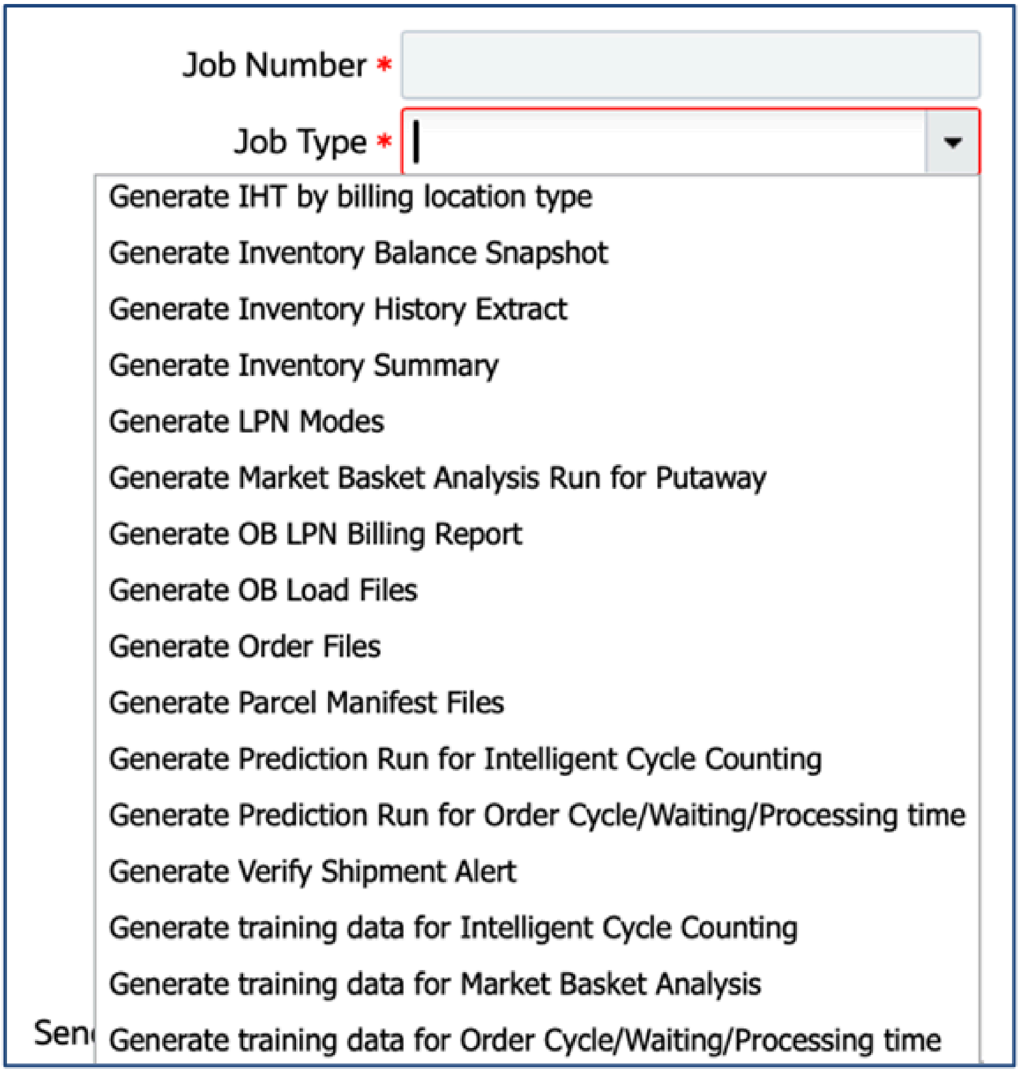
Using the Scheduled Jobs UI

You can schedule the generation of training data for your AI/ML models directly through the **Scheduled Jobs UI**.

In the **CRUD pane**, users will find a dropdown menu where they can select the appropriate job type. To initiate the data generation process, choose:

"Generate Training Data for [Metric]"

This action creates a scheduled job that prepares the data needed for model training.



Job Number *
Job Type *
Generate IHT by billing location type
Generate Inventory Balance Snapshot
Generate Inventory History Extract
Generate Inventory Summary
Generate LPN Modes
Generate Market Basket Analysis Run for Putaway
Generate OB LPN Billing Report
Generate OB Load Files
Generate Order Files
Generate Parcel Manifest Files
Generate Prediction Run for Intelligent Cycle Counting
Generate Prediction Run for Order Cycle/Waiting/Processing time
Generate Verify Shipment Alert
Generate training data for Intelligent Cycle Counting
Generate training data for Market Basket Analysis
Generate training data for Order Cycle/Waiting/Processing time
Sen

Selecting one of the above job types from the dropdown will expand additional **Job Parameters**, which must be configured before scheduling the job:

- **Username** *(mandatory)*: Enter a valid WMS login. This identifies the user associated with the job.
- **From Date**: The starting date for the data to be included in training.
- **To Date**: The ending date for the data range.
- **Past Number of Days**: Alternatively, you can specify a rolling window of past days to pull data from (e.g., last 30 days).

> **Note:** Either the **From/To Date range** (mm/dd/yy) or the **Past Number of Days** (must be in format -x) must be provided. These values determine the time window used to generate training data for the selected metric.

> **Note:** Each of the jobs will not be enabled by default. If you attempt to save without any value for username, you will receive an error message.

The output (model) of the job (different for each metric) will be captured in the **AIML Models UI**. One for each job run.

- Each record thus generated is defined by a unique composite key (business key).
- The job will add, or update records based on the key.

For every run of the scheduler, the job will be logged under Scheduled Job Detail UI. Logs should be provided for each job run. If the job fails, logs will be updated accordingly with the Failed job status.

Automating AIML Metric Prediction Runs

Users will have the ability to generate this metric through a scheduled job. This will predict the number of orders during a user specified time. Users can configure the scheduled job in Scheduled Jobs UI and will create the data required to build the model.

Generating Prediction Runs

The above job type is available in the CRUD panes of the Scheduled Job UI.

- Selection of the above job type from the drop down will further expand the Job parameters:

- ***Username***
- ***Training Run Number***
- ***Orders*** -

-
  - ***Wave number -*** (optional)
  - **Order Type(s)**

The result of the model is available in Prediction Run UI when a user clicks on the corresponding details button in Prediction Run UI. By selecting the run number users will be able to see the results of the specific run. Output results will be stored separately per KPI/Metric and per prediction run and will be different for each metric.

The output data will also record if it was the result of a "default" template for that KPI/Metric.

> **Note:** The detail drill-down from the Prediction Run UI will show the data in a data grid.

Step Two: Viewing the Training Data

The process to create one of these models is defined below:

- **Step One:** Create a Scheduled Job.
- **Step Two:** View the Training Data.
- **Step Three:** Define the model with parameters/filters and select it in the AIML Training Template.
- **Step Four:** Run the prediction and view results in the AIML Prediction Run UI.
- **Step Five:** Deploy, retrain models, and interpret results.

These screens allow you to view training data:

- **AIML Training Data –** Allows users to view the prepared data generated by a scheduled job.
- **AIML Training Template –** Where users define templates and optionally set a default.
- **AIML Models UI –** Displays all models built from training templates.
- **AIML Prediction Run UI –** Shows the results from prediction executions.
- **AIML Predictive Dashboard –** Displays the latest prediction results (currently available for Order Cycle Time only).

When the job runs, it processes the **Username** and selected **AIML Template**, and upon successful completion, generates a model. This job also triggers the **Train** action on the corresponding AIML Training Template defined in the job parameters, allowing users to generate models on a recurring basis.

The output (model) from each job run is captured in the **AIML Models UI**, where a new model is logged for every execution.

A new screen called **AIML Training Data** allows users to view the results of the scheduled job for:

- **Order Cycle Time**
- **Order Waiting Time**
- **Order Processing Time**
- **Intelligent Cycle Count**
- **Market Basket Analysis**

Users can select one of these metrics from a dropdown. Once selected, a data grid appears showing the relevant training data. This data represents the output of the "Generate Training Data for..." job.

Each metric will have its own unique data grid:

- Switching the dropdown will load a new grid with fields specific to the selected metric.
- Each sub-screen supports **search functionality**.
- Users can **delete** records but cannot **create or edit** them.
- **CSV download** is available.
- Users can also **filter by date range** using “From Date” and “To Date”.

Two new sub-screens—**Intelligent Cycle Count** and **Order Cycle/Waiting/Processing Time**—are now available in the **AIML Prediction Run Details UI**. This is where the final prediction results are stored and displayed.

Step Three: Define Model with Parameters and Filters

A new UI screen, **AIML Training Template**, has been introduced. This screen allows users to configure the parameters and details required to generate an AI/ML model for a specific metric.

The following fields are **mandatory** when creating a training template:

- **Template Name**
- **AIML Metric**
- **Date Range**
- **AIML Algorithm**

> **Note:**
> - The template name **cannot contain a forward slash (/)**.
> - Once saved, the **template name cannot be edited**.

In this screen, users can **train and generate AI/ML models** for future predictions. Similar to the **Wave Template UI**, the **AIML Training Template** screen allows users to configure specific parameters and settings required to generate a model for each supported metric.

The following KPIs are available for model creation:

- **Order Cycle Time** (regression problem)
- **Order Waiting Time** (regression problem)
- **Order Picking Time** (regression problem)
- **Intelligent Cycle Count**

Action Buttons Available in the Training Template UI

- **Train Template**

  This button is enabled **only** for metrics that support model generation (e.g., **Order Cycle Time** and **Order Picking Time**).

Once the template is saved, users can invoke the **'Train Template'** action button. Upon clicking, the system will prompt the user to confirm whether they wish to proceed. If confirmed, the model training process begins, and the resulting model is captured and displayed in the **AI/ML Models UI**.

The **'Predict'** action button is available only for specific metrics that do **not** require a trained model. For these KPIs, selecting **Predict** will directly generate output based on the available data—no model creation is necessary.

Users can define the **date range** for which they want to generate AI/ML KPI data. However, if the selected range does not contain sufficient data, the system will return the following error:

"Insufficient Training Data to Generate Model"

This screen supports full **CRUD operations**. When editing an existing template, all fields except the **Template Name** are editable. Once the required fields are properly configured with valid values, the template can be saved successfully.

Step Four: Predict and Execute Capturing Results in AIML Prediction Run UI

Prediction Run UI

In Prediction Run UI, each prediction run is defined by the training run that has been used and other parameters that will vary by the metric. For example, Order Cycle Time, it is the order filter criteria.

Each time a metric invokes the 'Predict' button, the results will be stored here. Additionally, each scheduled job run for the job type ' ***Generate Prediction Run*** ' will result in an entry here.

The following fields are in the Prediction Run UI:

Field Name:	Description:
Prediction Run Nbr	System generated number on invoking Predict action or on running a 'Generate Prediction Run' job.
Training Run Nbr	Reference to the training Run from the AIML Training Models
Prediction Parameters	Reference to the parameters used to execute the model against new data to generate predictions
Default flag	Copied from training run; there can be multiple runs for same metric with this enabled. the latest one is the relevant one.
Status	created, training, completed, failed, cancelled

There is a drill down detail button available when a user selects any given record. This shows the output data for that run. It is specific to the metric of the run that the user selects. The output will be one of the following KPIs:

- Intelligent Cycle Count
- Order Cycle time
- Order Waiting Time
- Order processing Time

Users are also able to use search/filters for Prediction Run Number, Training Run Number, Status and Default Flag. Each drill down screen also supports CSV download and appropriate filters as needed.

Order Cycle Time, Order Waiting Time, Order Processing Time, and Intelligent Cycle Count

When selecting a record in the AI/ML Models screen for any of the metrics, a user must click the ‘Predict’ button to carry out the prediction. A popup will be displayed to further finetune the models.

Note: If no field is populated, then a user will be shown an error.

- On clicking 'OK', the execution i.e. the prediction for the respective metrics will commence for the set of orders selected. (On 'Cancel', the action is cancelled.)
- On completing the execution, a record is generated in the Prediction Run (Inquiry) UI which records the parameters and details used to run that prediction.
- Also, the execution will generate the output i.e. the *Order Cycle Time or Order Waiting Time or Order Processing Time* depending on the metric of the record selected for prediction, viewed in a separate UI screen ex: '*AIML Order Cycle Wait Processing Time*

AIML Models Inquiry

Every training template execution, whether manual or scheduled will result in an entry here. For metrics that produce a model, the model generated will be stored within this screen.

The following fields are supported within this UI:

Field Name	Description
Facility	User's context facility
Training Run Number	System generated number on running a AIML training template
AIML Training Template	name of the training template ran for the specific run number

Field Name	Description
Training Start Date	Start date of the training data
Training End Date	End date of the training data
AIML Algorithm	Algorithm for which the model was generated
Training Filters	Copied from template
Algorithm Parameters	Copied from template
Default Flag	Same as from template
Status	Created, training, completed, tailed, cancelled
Message Text	• Shows the latest message from the training run • Will be a reference number or link • updated when training completes. some metrics won’t have a model (like basket analysis) • This should be copied back to the template "Last Trained Model"
AIML Model Reference	Shows the performance of the model
Performance Metrics	Mean absolute error, r^2
Create Timestamp	Date and time of the model/record. Creation
Create User	User who generated the model/record (one who ran the AI/ML training template)
Metric	Metric being training, comes from template

Action Buttons in AIML Models Inquiry:

- - - **Logs**
      - **Predict**
      - **Prediction Run**

Each of the drill down screens supports CSV download and appropriate filters as needed.

Step Five: Interpreting Results

After successfully creating and viewing your models in AIML Models UI, it is important to note that your results may not yet be optimal. It could potentially take several iterations of the template to generate better results. Oftentimes it may be best to change the type of algorithm along with the types of parameters. Your results could also vary based on the type of industry you are in.

2 AIML Predictive Fulfillment Dashboard

AIML Predictive Fulfillment Dashboard

The **AI/ML Predictive Dashboard** allows you to make well informed decisions based on past order data to view future Order Cycle/Processing/Waiting Time predictions. This information comes from the most recent prediction from your model in AIML Prediction Run.

The predictive dashboard displays the average order cycle time in hours. We have included various KPIs that gives insight on how your orders have and will move out of the warehouse. By clicking on the order status, it will further filter and display the order type by status.

Your orders above and below your set target threshold will be shown in a datagrid that displays the following information:

- Order Number
- Status, Order Type
- Predicted Time
- Expected Shipping Date
- Required Shipping Date

The ratio of orders above and below your target threshold will also be shown in a pie chart above the data grid.

Steps to enable:

- Navigate to the UI and click on the username dropdown.
- Select Redwood UI Prototype.
- All the data should be there.

Please note that only the Default Template will display data in the Predictive Dashboard. If no default template is configured, the predictive dashboard will not return any results.

You can access the AI/ML Predictive Fulfillment Dashboard by clicking the on the username.

Oracle Market Basket Analysis

We've introduced **Market Basket Analysis (MBA)** in Oracle Warehouse Management to help you gain deeper insights into your inventory and customer buying patterns. This AI-powered feature identifies **items frequently ordered together**, enabling smarter decisions when it comes to **inventory placement and putaway strategies**.

By understanding these patterns, you can optimize slotting, reduce picker travel time, and increase overall warehouse efficiency.

What It Does

Market Basket Analysis leverages **Oracle's advanced AI/ML algorithms** to analyze historical order data. It detects relationships between products and groups them into **"frequent item sets"** based on how often they are ordered together and the likelihood that this trend will continue.

The result is a clear, actionable report that helps you:

- Identify product pairings or groupings.
- Improve inventory layout by slotting complementary items together.
- Streamline picking operations based on actual order behavior.

There are **two key metrics** when running a Market Basket Analysis, support and confidence. The system ranks items based on support and then confidence, this means that if supports are equal between items, then they will be based on confidence after.

Support measures how frequently an itemset appears in the dataset. It's calculated as the proportion of transactions that contain a specific item or combination of items out of all transactions.

Confidence indicates the likelihood that item B is purchased when item A is purchased. It's calculated as the ratio of transactions containing both A and B to the number of transactions containing just A.

- Create a scheduled job to generate the training data for Market Basket Analysis in the **Scheduled Jobs** UI.

> **Note:** You must enter a valid username, or else you'll receive an error.

- Navigate to the AI/ML Training Data to verify that the training data contains the correct data.
- Create a Market Basket Analysis Training Template in the **AI/ML Training Template** UI

> **Note:** To run the Market Basket Analysis, you must select the "Apirori" algorithm.

- After creating your Market Basket template with all of the mandatory fields completed, click **Train Template**.
- Navigate to the **AI/ML Model** UI and select the most recent Market Basket Analysis entry. In the "Model Reference" field, click on the hyperlink.
- You can now view the results of your successful Market Basket Analysis.

> **Note:** *association_max_rule_length* should only be changed if users want to see associations past an A & B association. For example, if *association_max_rule_length* = 4, users will see relationships between items A/B/C/D, however this may impact performance.

Inbound and Outbound Performance Metrics

You can access Inbound and Outbound Performance metrics via tabs in **Redwood Desktop Experience → Dashboard.**

From the **Inbound Performance Metric** tab you can view:

- LPNs Received in the last 7 days
- Open IB Shipments by Status

- Today's Appointments
- Last 7 days average dock to stock cycle time

From the **Outbound Performance Metric** tab you can view:

- Open Tasks
- Orders to be Shipped Today
- Outbound LPNs by Status
- Order Fill Rate

Accessing the Performance Metrics

1. To access the Inbound and Outbound Performance Metrics, click the user drop-down menu in WMS, and select the **Try the new Redwood Experience** option. A new tab will open up with the new redwood experience. The Home page gives you quick access to useful links like User Guides and Release Content.
2. Click the **Dashboard** tab to access the available Performance Metrics.

3 GenAI Features

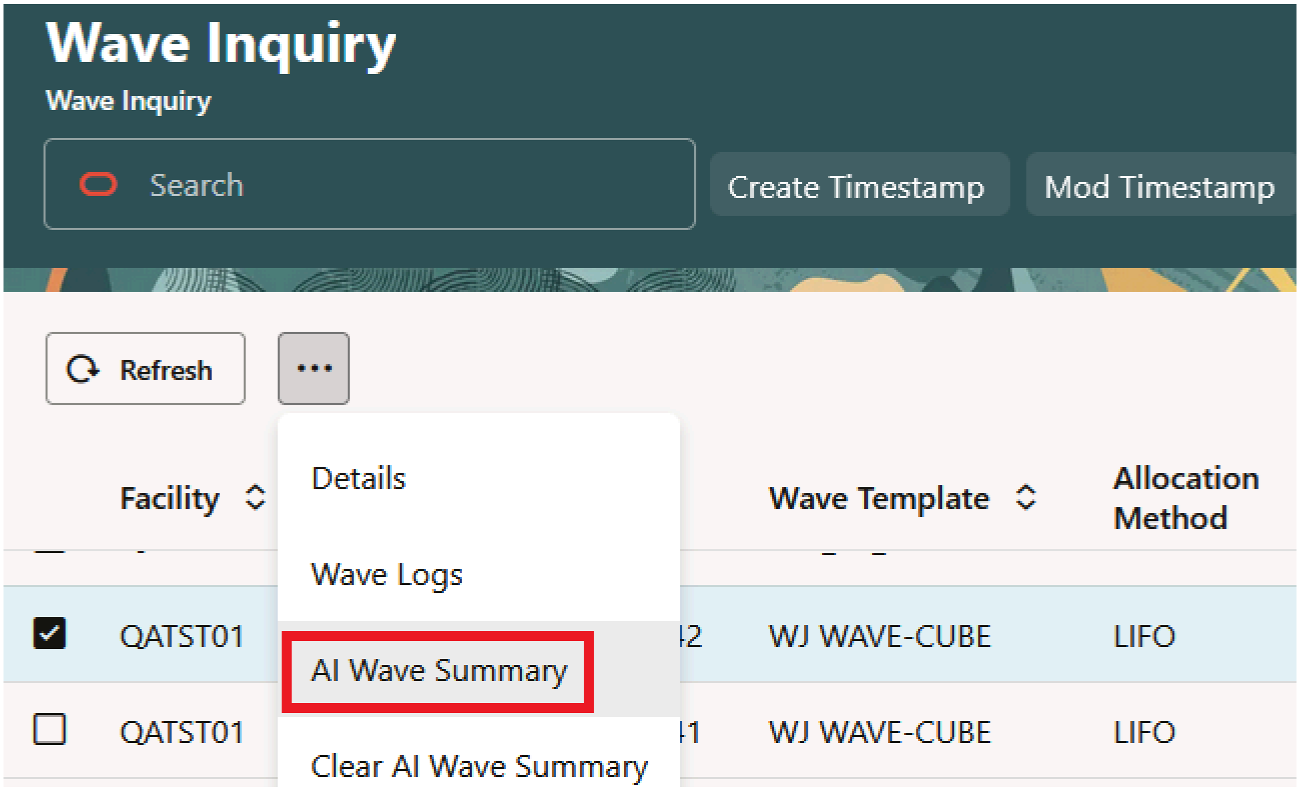
Intelligent Wave Summary

In the **Wave Inquiry Summary** UI, you have a comprehensive and efficient way to review and analyze waves, bringing advanced AI capabilities to your warehouse management system.

AI Wave Summary Button

The**AI Wave Summary** button, in the wave inquiry header, provides quick access to valuable insights.

Click this button to open up a pop-up window, displaying an AI-generated summary of the wave execution, including key details and potential issues. The button is enabled only for specific wave statuses: 'Completed', 'Completed, Not Fully Allocated', 'Completed, Nothing Allocated', and 'Failed', ensuring relevant information is readily available.



	Facility	Wave Template	Allocation Method
☑	QATST01	WJ WAVE-CUBE	LIFO
□	QATST01	WJ WAVE-CUBE	LIFO

Customized Wave Summaries

Different pop-up summaries are generated based on wave types and statuses, providing tailored insights. Summaries for 'Completed' and 'Failed' statuses offer a clear understanding of the wave execution for:

- Picking Waves and Wave-based Replenishment
- Replenishment Waves
- Lean Time Replenishment Waves

Note:

- AI Summary features are not supported for all languages that the WMS supports. For languages not supported by AI, the summary will default to English.
- AI Summary features are enabled by default in North America and Europe geographies. Customers elsewhere will have to raise an SR if they want this feature as this requires authorization of data communication to one of the two geographies

Wave Inquiry and Wave Inquiry Detail UI

From the **Wave Inquiry** and **Wave Inquiry Detail** UIs:

- You can view all records related to a wave run, ensuring a comprehensive overview.
- Action buttons, such as 'Substitute Item', are readily accessible, allowing users to take immediate action.
- Advanced search capabilities enable users to filter and find specific wave inquiries efficiently.

Permission-Based Access

- Group Permission for 'Wave Inquiry / AI Wave Summary' determines user access.
- By default, this permission is disabled, allowing administrators to grant access to specific user groups.

Wave Inquiry UI - Replenishment Wave

The **Wave Inquiry** UI for **Replenishment Wave** includes the following new fields:

- **Total Wave Time:** This field will display the overall duration of the wave, providing a clear understanding of the time taken for the entire process.
- **Task Creation Time:** Warehouse managers can now pinpoint the exact moment when tasks are generated, allowing for better task management and resource allocation.
- **Allocation Time:** Knowing the allocation timing is crucial for optimizing inventory management. This field will reveal when the allocation process occurs, helping you identify potential bottlenecks.
- **Number of Tasks Created:** A clear indication of the workload generated by each wave, enabling better workforce planning.
- **Total Quantity Allocated:** Provides insight into the volume of inventory allocated, aiding in stock management and ensuring efficient resource utilization.

Search by Replenishment Trigger Modes

You can search by Replenishment trigger modes via the following search fields in the Wave Inquiry UI:

- **Minimum Capacity:** Quickly find waves triggered by minimum capacity thresholds, ensuring you can replenish stock promptly.
- **Reactive Replenishment:** Identify waves generated through reactive replenishment strategies, allowing for immediate analysis and response.
- **Percentage of Max:** Locate waves based on percentage-based replenishment triggers, enabling precise inventory management.
- **Order Based Replenishment:** Focus on waves created for order fulfillment, helping you prioritize customer orders efficiently.
- **Movement Request Based Replenishment:** Search for waves related to movement requests, providing a comprehensive view of inventory adjustments.

This search feature is also available as an optional column in the **Wave Inquiry** UI, ensuring a clutter-free interface. You can enable and customize your search criteria, adding this field when needed for tailored analysis.

Clear AI Wave Summary Button

From the **Wave Inquiry UI**, the 'Clear AI Summary' button allows you to manually clear the wave summary information stored in the database, offering the ability to refresh and update wave details whenever needed.

The **Clear AI Wave Summary** button is useful for:

- **Wave Summary Generation:** After a wave run, the system automatically generates a summary, including order selection, sequencing, and allocation strategies.
- **Wave Failure and Resolution:** If a wave fails, the manager can troubleshoot and resume the wave.
- **Clearing Summary Data:** Upon resuming a failed wave, the 'Clear AI Summary' button removes the previous summary data from the database, reflecting the wave's new 'Completed' status.
- **Regenerating Summary:** Users can click the 'AI Wave Summary' button to create an updated summary, providing a fresh analysis of the wave's performance.

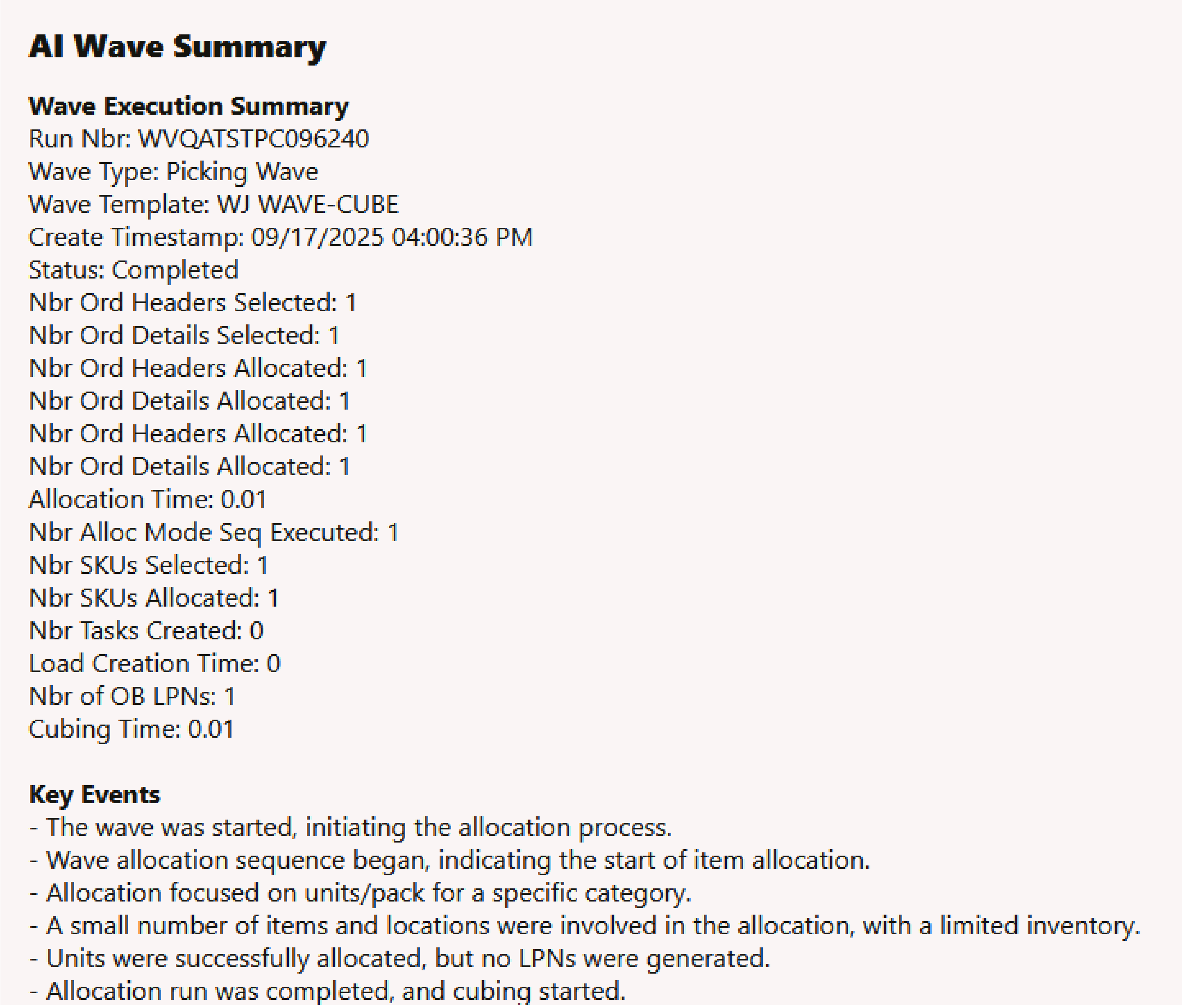
Steps to Enable

To enable AI Wave Summary in the Wave Inquiry UI:

1. Enable the Group Permission '**Wave Inquiry / AI Wave Summary.**'

To access **AI Wave Summary**:

1. From Redwood Desktop, click Ask Oracle.
2. Select the record(s) and click the **More Actions** elipses button (...).
3. Click **AI Wave Summary**.
4. An AI Summary pop-up will appear with the AI Wave Summary:



AI Wave Summary

Wave Execution Summary
Run Nbr: WVQATSTPC096240
Wave Type: Picking Wave
Wave Template: WJ WAVE-CUBE
Create Timestamp: 09/17/2025 04:00:36 PM
Status: Completed
Nbr Ord Headers Selected: 1
Nbr Ord Details Selected: 1
Nbr Ord Headers Allocated: 1
Nbr Ord Details Allocated: 1
Nbr Ord Headers Allocated: 1
Nbr Ord Details Allocated: 1
Allocation Time: 0.01
Nbr Alloc Mode Seq Executed: 1
Nbr SKUs Selected: 1
Nbr SKUs Allocated: 1
Nbr Tasks Created: 0
Load Creation Time: 0
Nbr of OB LPNs: 1
Cubing Time: 0.01

Key Events
- The wave was started, initiating the allocation process.
- Wave allocation sequence began, indicating the start of item allocation.
- Allocation focused on units/pack for a specific category.
- A small number of items and locations were involved in the allocation, with a limited inventory.
- Units were successfully allocated, but no LPNs were generated.
- Allocation run was completed, and cubing started.

4 Index

Index

Classification

Classification is the process of predicting which category (or “class”) a data point belongs to, based on patterns in the data. In machine learning, models are trained to recognize similarities among data points so they can accurately classify new, unseen inputs.

Example: Predicting whether an order is "on time" or "delayed."

Regression

Regression is a statistical technique used to model the relationship between a **dependent variable** and one or more **independent variables**. In machine learning, regression is used to predict continuous values.

Example: Estimating how many hours it will take to fulfill an order based on its size, contents, and location.

Tuple

A tuple is an **ordered collection** of elements, often used to represent a single, structured record of related values. Tuples are commonly used in programming and machine learning to organize data inputs.

Example: A tuple might look like this: (OrderID, ItemCount, ShipDate).