ORACLE

What's new and what's next in Oracle Machine Learning

New features and roadmap

Move the Algorithms – Not the Data

Mark Hornick, Senior Director
Sherry LaMonica, Consulting MTS
Oracle Machine Learning Product Management

June 2023

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Agenda

Component overview
New features
Roadmap

Oracle Machine Learning family of components

OML Component	Oracle Autonomous Database (19c, 21c)	Oracle Database (19c, 21c)	Oracle DBCS	Oracle Exadata CS/CI/C@C
OML4SQL API Build ML models and score data with no data movement	✓ ADB-S, ADB-D, ADB C@C, ADB Dedicated Region	✓	✓	✓
OML4Py API Leverage the database as a high-performance compute engine from Python with in-database ML	✓ ADB-S, ADB Dedicated Region	✓	✓	✓
OML4R API Leverage the database as a high-performance compute engine from R with in-database ML	✓ ADB-S, ADB Dedicated Region	✓	✓	✓
OML Notebooks SQL, PL/SQL, Python, R, conda, and markdown interpreters	✓ ADB-S, ADB Dedicated Region			
OML AutoML UI No-code automated modeling interface	✓ ADB-S, ADB Dedicated Region			
OML Services RESTful model management, deployment, monitoring	✓ ADB-S, ADB Dedicated Region			
Oracle Data Miner SQL Developer extension with a drag-n-drop interface for creating ML methodologies	✓ ADB-S, ADB-D, ADB C@C, ADB Dedicated Region	✓	✓	✓

Poll Questions

1) Which OML APIs have you tried?
(check all that apply)

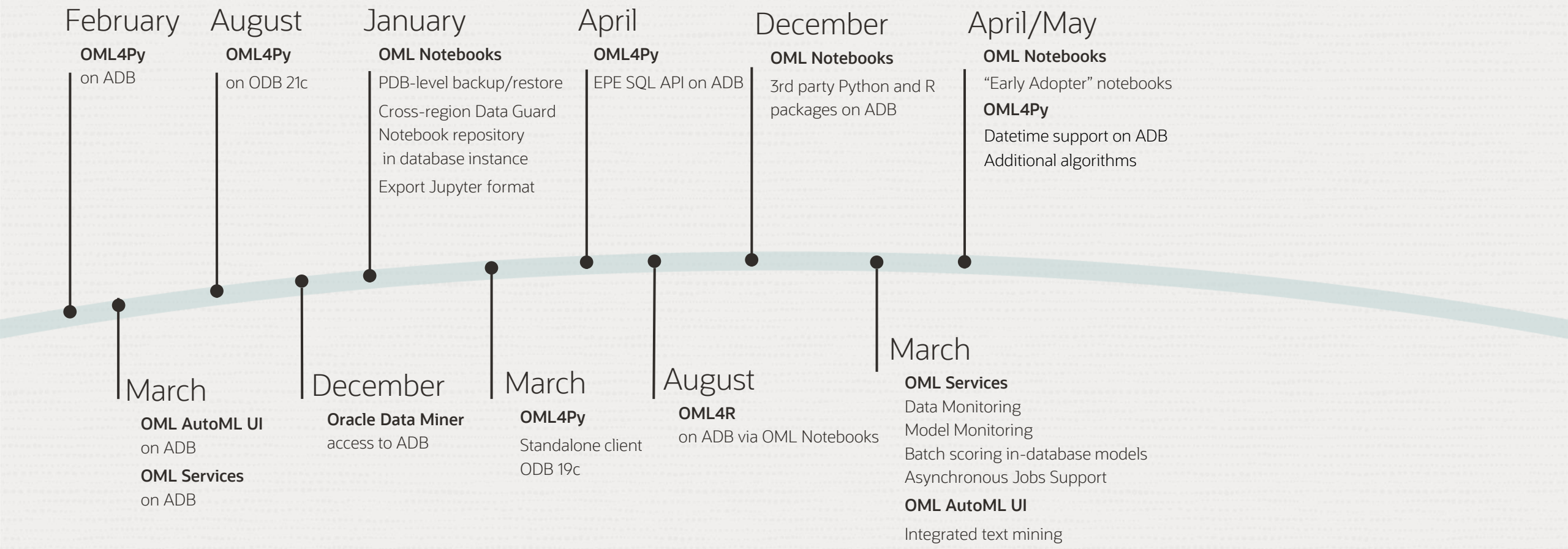
- OML4SQL
- OML4Py
- OML4R
- OML Services REST API

2) Which OML user interfaces have you tried? (check all that apply)

- OML Notebooks
- OML AutoML UI
- Oracle Data Miner
- OML Models

OML New Features

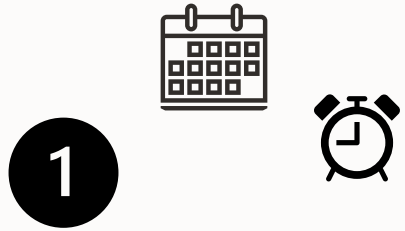
Oracle Machine Learning release timeline



OML4Py 2.0

New data types to enable greater Python-based exploration and manipulation of database data

- **Datetime** - represents a table column of TIMESTAMP, with DATE
- **Timedelta** - represents a table column of difference between two dates or times
- **Timezone** - represents a table column of time zone data
- **Integer** - represents a table column of INTEGER data



In-database algorithms now exposed through the OML4Py API

- **Exponential Smoothing** (ESM) – ML algorithm used for forecasting univariate time series data
- **Non-negative Matrix Factorization** (NMF) - feature extraction algorithm often used where many attributes have individually weak predictability. Producing linear combinations of attributes can increase data signal, and in text analysis, uncover topics or themes
- **Extreme Gradient Boosting** (XGBoost) - a highly efficient, scalable machine learning algorithm for regression and classification that leverages the open-source package

Blog: Announcing OML4Py 2.0

OML4Py 2.0 date-related examples

```
%script

describe SH.SALES
```

```
Name          Null?    Type
------------- -------- ------------
PROD_ID NOT NULL NUMBER
CUST_ID NOT NULL NUMBER
TIME_ID  NOT NULL DATE
CHANNEL_ID NOT NULL NUMBER
PROMO_ID NOT NULL NUMBER
QUANTITY_SOLD NOT NULL NUMBER(10,2)
AMOUNT_SOLD NOT NULL NUMBER(10,2)
```

```
%python

SALES = oml.sync(schema="SH", table="SALES")
z.show(SALES.head())
```

Get a proxy object that contains a DATE column

settings

PROD_ID	CUST_ID	TIME_ID	CHANNEL_ID	PROMO_ID	QUANTITY_SOLD	AMOUNT_SOLD
33	1006	2001-02-25 00:00:00	2	999	1.0	46.5
33	1427	2001-02-25 00:00:00	2	999	1.0	46.5
33	1524	2001-02-25 00:00:00	2	999	1.0	46.5

```
%python

pd.set_option('display.max_columns', 50)
pd.set_option('display.width', 1000)
SALES.describe(include='all')
```

Compute column statistics

	PROD_ID	CUST_ID	TIME_ID	CHANNEL_ID	PROMO_ID
count	918843.000000	918843.000000	918843	918843.000000	918843.000000
unique	NaN	NaN	1460	NaN	NaN
top	NaN	NaN	2001-10-18 00:00:00	NaN	NaN
freq	NaN	NaN	2940	NaN	NaN
mean	78.183945	7289.807720	NaN	2.861603	976.396093
std	49.008014	8948.653221	NaN	0.686874	121.829887
min	13.000000	2.000000	NaN	2.000000	33.000000
25%	31.000000	2383.000000	NaN	2.000000	999.000000
50%	48.000000	4927.000000	NaN	3.000000	999.000000
75%	127.000000	9163.000000	NaN	3.000000	999.000000
max	148.000000	101000.000000	NaN	9.000000	999.000000

```
%python

date = SALES['TIME_ID']
SALES2 = SALES.concat({'YEAR': date.year, 'MONTH': date.month})
SALES2.head()
```

Extract YEAR and MONTH features from TIME_ID

	PROD_ID	CUST_ID	TIME_ID	CHANNEL_ID	PROMO_ID	QUANTITY_SOLD	AMOUNT_SOLD	YEAR	MONTH
0	33	1006	2001-02-25	2	999	1.0	46.5	2001	2
1	33	1427	2001-02-25	2	999	1.0	46.5	2001	2
2	33	1524	2001-02-25	2	999	1.0	46.5	2001	2
3	33	1547	2001-02-25	2	999	1.0	46.5	2001	2
4	33	1672	2001-02-25	2	999	1.0	46.5	2001	2

OML Notebooks Early Adopter

Enhancing user experience for ADB-based notebooks

"Early Adopter" highlights

- Faster notebook loading time
- Oracle look and feel
- Richer visualization
  Text, Line Chart, Area Chart, Bar Chart, Pyramid Chart, Pie Chart, Donut Chart, Funnel Chart, Tag Cloud, TreeMap Diagram, Sunburst Diagram, Scatter Plot, Box Plot
- Paragraph comments
- Paragraph dependencies
- Service name selection (H/M/L) dropdown
- Zeppelin and Jupyter layouts
- On-page notebook versioning, history viewing, and version comparison

OML Notebooks functionality

- Easily convert existing notebooks
- Save notebooks as personal and/or shared template
- Schedule notebooks to run
- Use workspace permissions
- Both existing and early adopter notebooks are supported during the introduction period

Blog: **Announcing next generation OML Notebooks on Oracle Autonomous Database**

OML Notebooks EA

Demonstration using OML4Py notebooks

Custom third-party packages on ADB via OML Notebooks

Expanding ADB as a platform for data science and machine learning

Support 3rd party Python and R package installation and conda environment creation

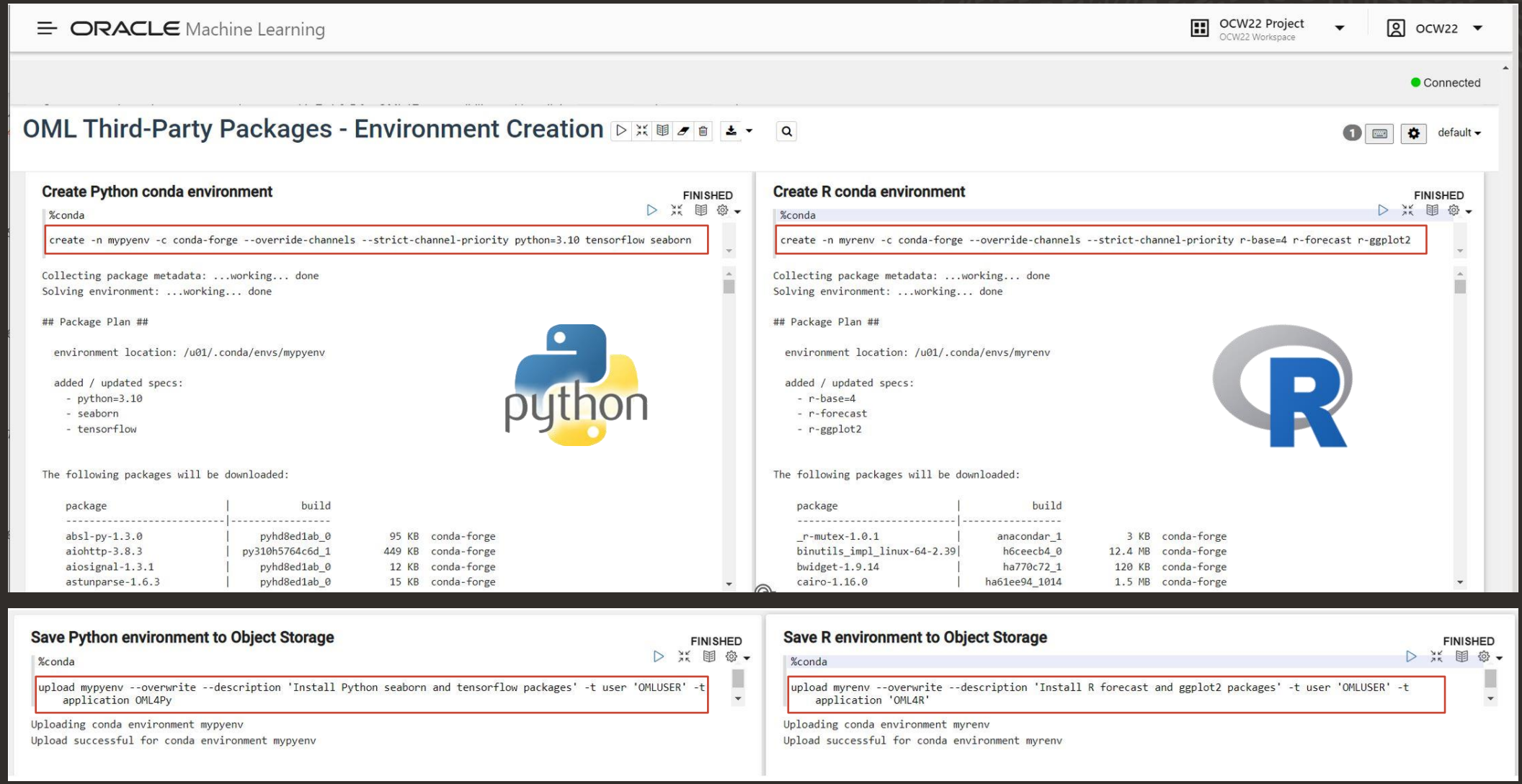
- **conda** - open-source package and environment management system
- Admins install third-party packages and manage conda environments
- Users download and activate conda environments from Object Storage
- Environments run in a separate container for security

Use with Python and R interpreters in OML Notebooks

Use with embedded execution in OML4Py and OML4R

Blog: Announcing custom third-party Python and R packages for use on Autonomous Database
OML Office Hours: Creating and using conda environments with third-party Python and R packages on ADB

Conda environment creation via OML Notebooks on ADB



Conda environment usage via OML Notebooks on ADB

Download and activate the 'mypyenv' environment

```
%conda
download mypyenv
activate mypyenv
```


```
Downloading conda environment mypyenv
Download successful for conda environment mypyenv
```

Download and activate the 'myrenv' environment

```
%conda
download myrenv
activate myrenv
```

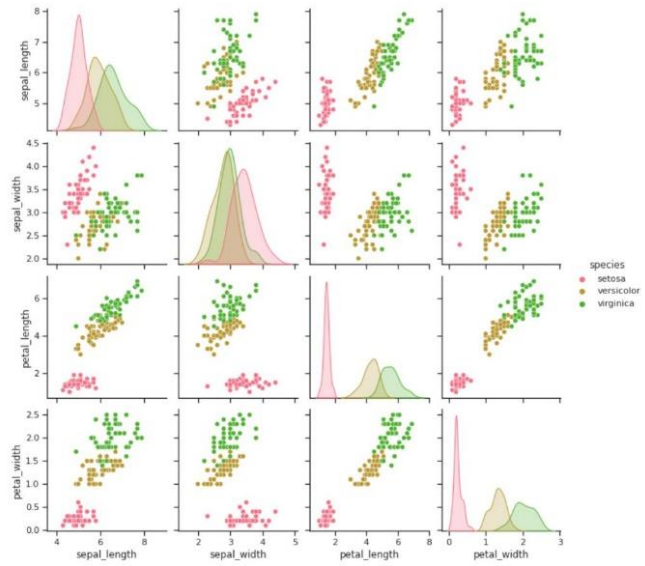

```
Downloading conda environment myrenv
Download successful for conda environment myrenv
```

ORACLE Machine Learning

Plot the pairwise relationships in the iris dataset

```
%python
sns.set(style="ticks")
sns.set_palette("husl")
sns.pairplot(df.iloc[:,0:6], hue="species")
```

```
<seaborn.axisgrid.PairGrid object at 0x7f15d4f42500>
```



ORACLE Machine Learning

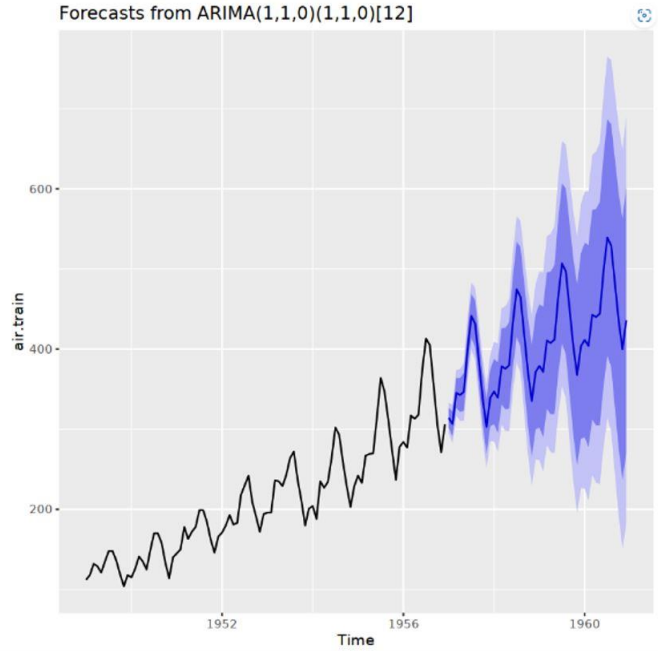
Create multi-step forecasts for each day

FINISHED

```
%r
accuracy(air.model)
air.multi.forecast <- air.model %>% forecast(h = n.test)
air.multi.forecast %>% autoplot()
```

A matrix: 1 x 7 of type dbl

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.442465	8.834477	6.351386	0.1376787	2.870884	0.2174955	0.003856404



OML Services Data and Model Monitoring REST API

Expanding data/model lifecycle support

Data Monitoring

Data drift detection

Track and report on significant changes in the statistical properties of data

Maintain data quality standards

Analyze data when it passes a threshold

Perform action when data passes threshold

Indicate when data drift has occurred

Model Monitoring

Model concept and quality drift detection

Track and report on, e.g., changes in prediction distribution and model accuracy, which may signal the need to rebuild a model or investigate causes

Supports in-database models

Indicate when model drift has occurred

Blog: Announcing OML Monitoring on Autonomous Database
OML Office Hours: Feature Highlight: Data and Model Monitoring in Autonomous Database

OML Services Batch Scoring REST API

Expanding ADB as a platform for data science and machine learning

- In-database scoring of multiple records in a table or view
- Invoked asynchronously since job may be longer running, i.e., not real-time
- Support for regression, classification, clustering, and feature extraction
- Enabled through REST API

```
$ curl -X POST "${omlservice}/omlmod/v1/jobs"
       -H "Authorization: Bearer ${token}"
       --data '{
        "jobSchedule":{
        "jobStartDate":"2023-04-10T10:15:00Z",
        "jobEndDate":"2023-04-15T10:15:00Z",
        "repeatInterval":"FREQ=DAILY",
        "maxRuns": "15"
         }...
```

Token variable

Job start and end dates

Job frequency

Maximum number of runs within the job time period

```
"jobProperties": {
  "jobname": "MY_BATCH_SCORING_JOB",
  "jobType": "BATCH_SCORING",
  "modelId": "35a97c7b-0ff4-4940-96f5-bfb29f64d223",
  "inputData": "CUSTOMERS360",
  "outputData": "CLASS_PRED1",
  "supplementalColumnNames":["CUST_ID","AFFINITY_CARD"],
  "jobServiceLevel": "MEDIUM",
  "topN": 2,
  "topNDetails": 2,
  "recompute": "true"}
```

Job name and type

Model ID

Input/output data

Columns from input data to identify output rows

topN probabilities and prediction details

Replace output table

Blog: [Announcing OML Services Batch Scoring on Autonomous Database]

OML Office Hours: [OML Services Batch Scoring on Autonomous Database]

OML AutoML UI

No-code user interface for automated machine learning

Integrated text mining

Text columns of type VARCHAR2 with length > 4000 and CLOB are treated as unstructured text

Automatically performs tokenization for combination with structured data

Optimized algorithm settings for high cardinality classification targets

Improve overall performance and scalability

Poll Question

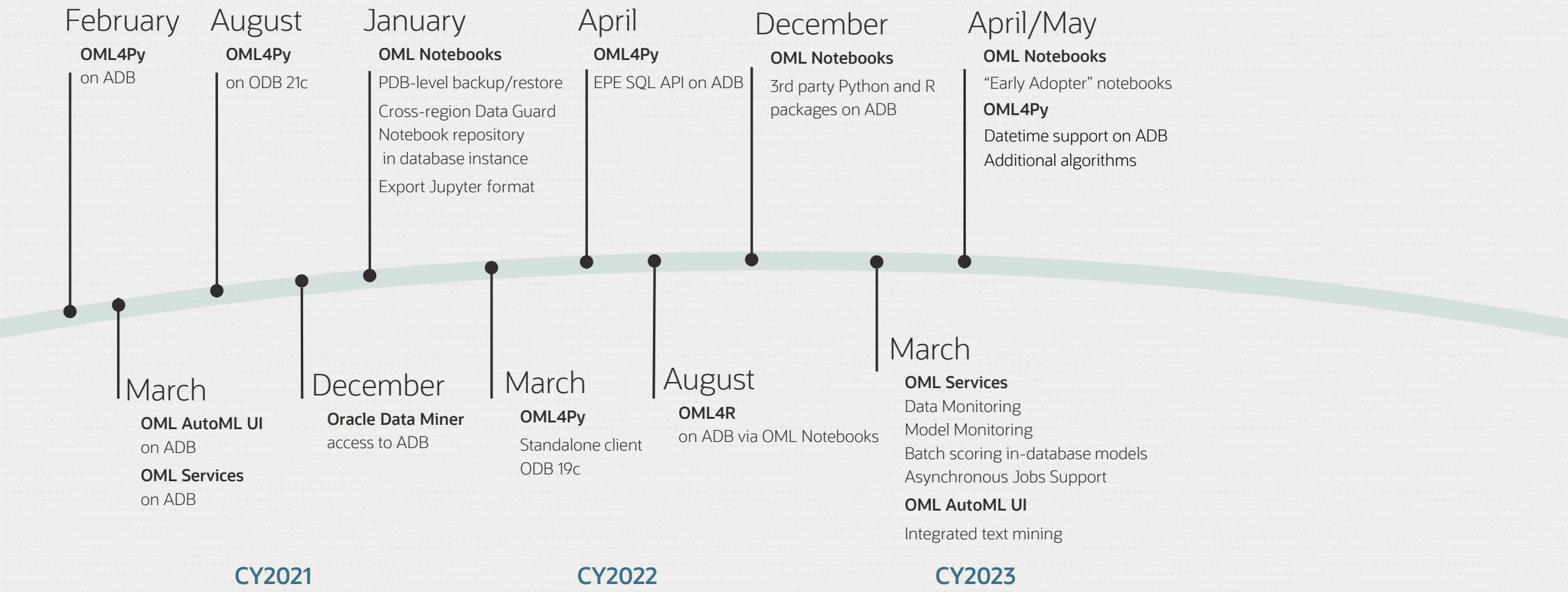
3) Which of these recent new features are you most likely to try or have already tried?

(select all that apply)

- OML Notebooks support for 3rd-party packages on ADB with OML4Py and OML4R
- OML4Py DATE and TIMESTAMP support
- OML Notebooks Early Adopter user interface
- OML Services Monitoring
- OML Services Batch Scoring for in-database models

OML Roadmap

Oracle Machine Learning release timeline



Oracle Machine Learning release timeline

January

OML Notebooks

PDB-level backup/restore

Cross-region Data Guard

Notebook repository in database instance

Export Jupyter format

March

OML4Py

Standalone client ODB 19c

April

OML4Py

EPE SQL API on ADB

August

OML4R

on ADB via OML Notebooks

December

OML Notebooks

3rd party Python and R packages on ADB

March

OML Services

Data Monitoring

Model Monitoring

Batch scoring in-database models

Asynchronous Jobs Support

OML AutoML UI

Integrated text mining

April/May

OML Notebooks

"Early Adopter" notebooks

OML4Py

Datetime support on ADB

Additional algorithms

CY2022

Oracle Database 23c

Survival Analysis via XGBoost

ESA with doc2vec

New GLM link functions

Monotone/interaction constraints XGBoost

Data query persisted with model

BOOLEAN data type support

4000 columns per table

High-cardinality feature optimizations

EM anomaly detection

Time Series model search

Multiple Time Series

OML4Py

OML4Py v2.0 for ODB

Updated standalone client on ODB 19c

Deploy in-DB models to OML Services

OML4R

OML4R v2.0 on ODB

Standalone client for ADB/ODC

OML Services

Data and model monitoring UI

OML AutoML UI

Time Series forecasting

CY2023

OML Notebooks

Support use of **GPU** compute

Full cutover to enhanced interface

OML4Py

Extended AutoML pipeline

ONNX model support

Deploy in-DB models to OML Services

Support use of **GPU** compute on ADB

OML Services

Bias and fairness monitoring

Monitoring for ONNX-format models

OML AutoML UI

Experiment scheduling

Anomaly detection

CY2024

OML Services Monitoring

Expanding data/model lifecycle support

Bias and fairness monitoring

Assess both data and models for possible bias and fairness concerns

Help ensure against putting certain groups or individuals at a disadvantage

Notify users when bias or fairness issues arise

No-code user interface for monitoring

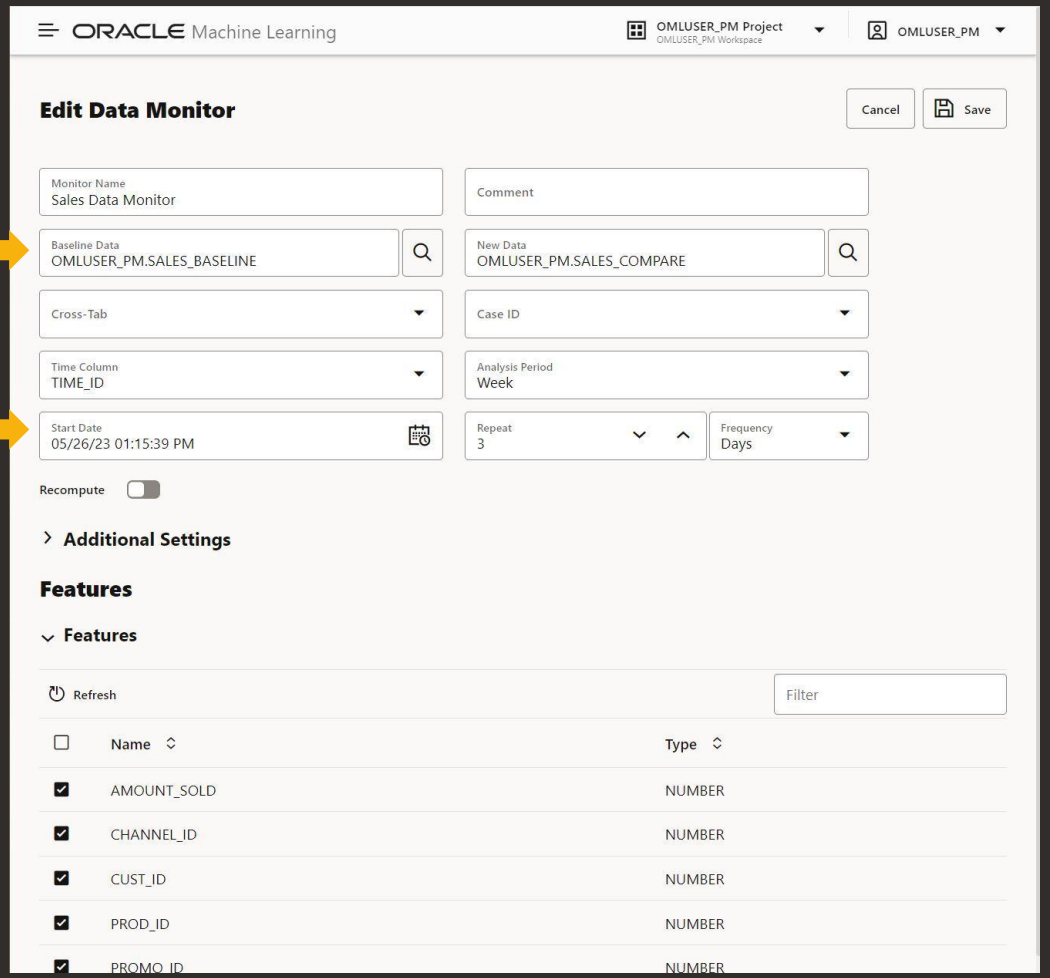
Define and schedule *data* monitoring jobs (phase 1)

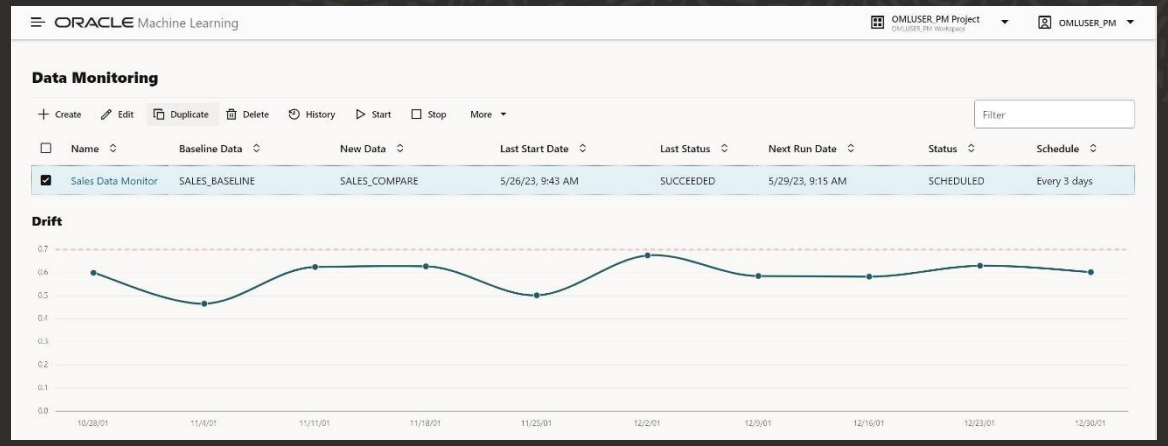
Define and schedule *model* monitoring jobs (phase 2)

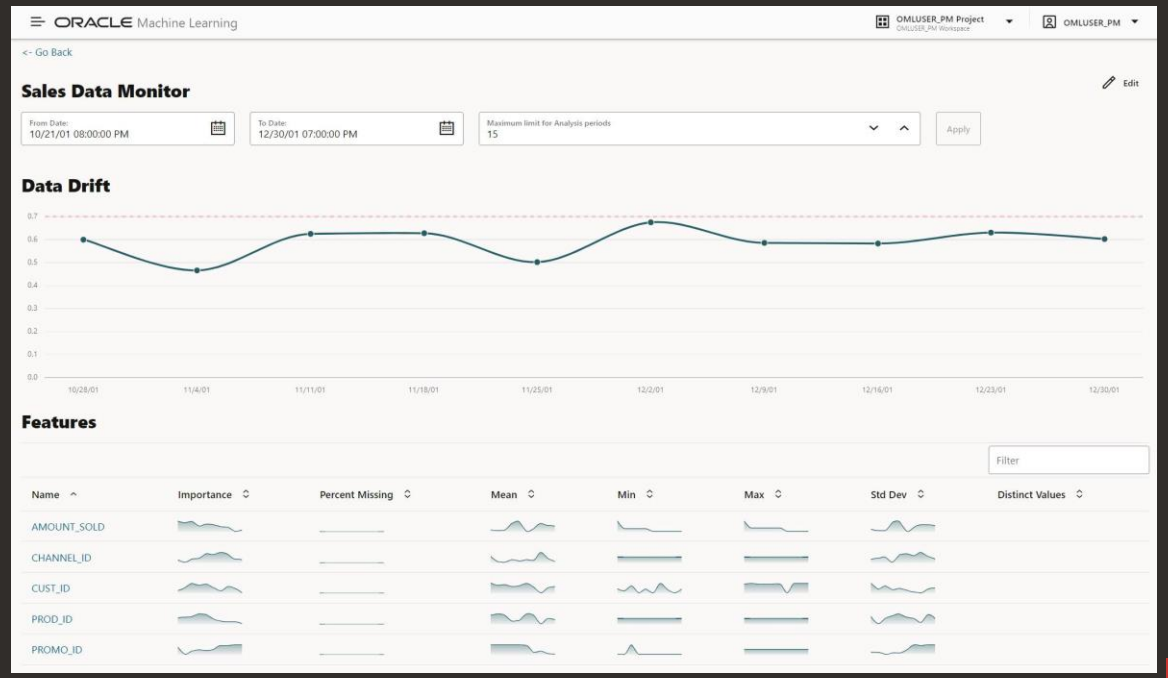
Visualize monitoring results

Be notified of drift

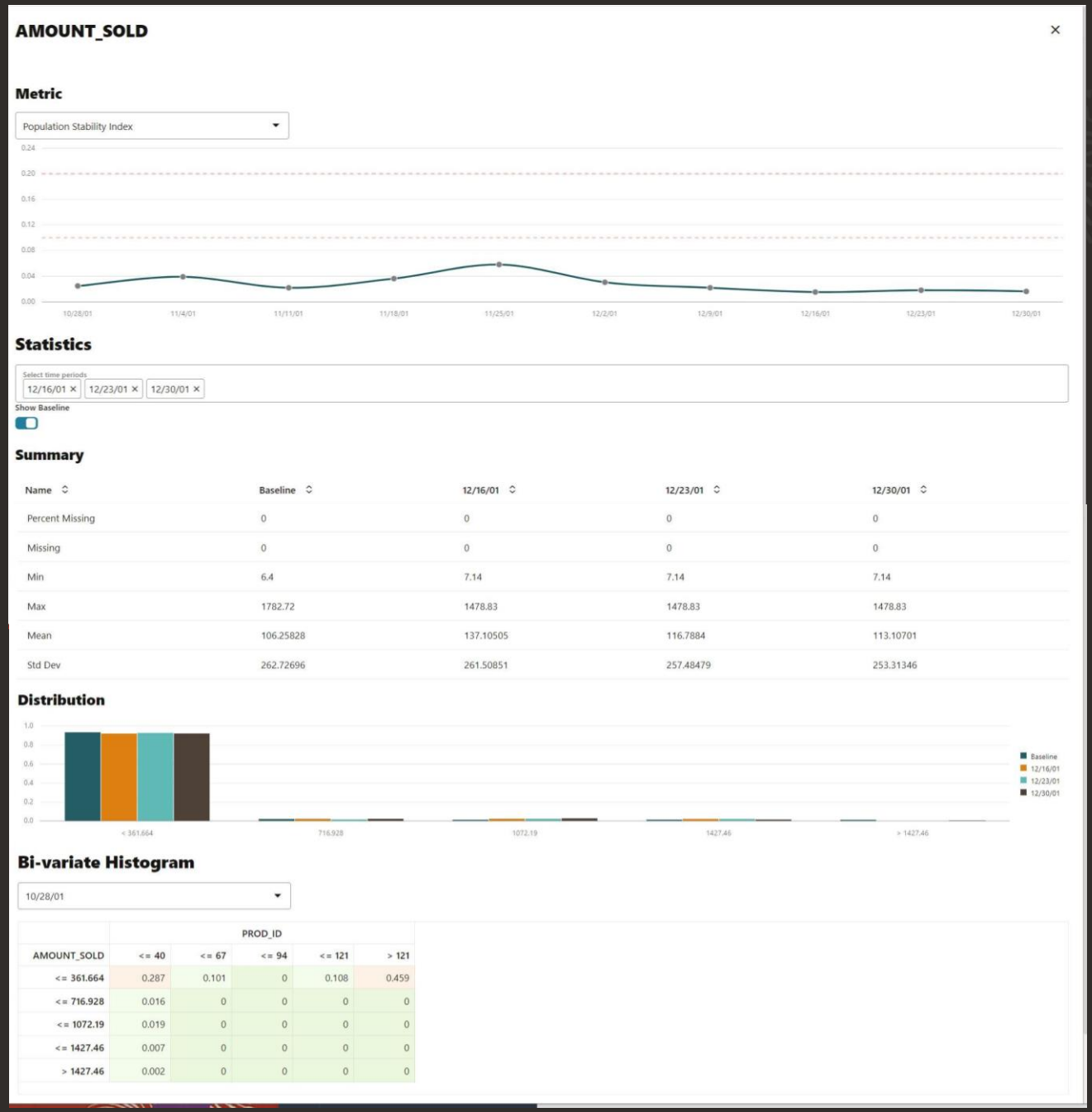
OML Data Monitoring no-code user interface







OML Data Monitoring



OML4Py

Expanding ADB as a platform for data science and machine learning

Extended AutoML pipeline

Match OML AutoML UI functionality for programmatic access

Include model quality metrics in AutoML result

In-database computation of model quality metrics, including algorithm-specific metrics

Deploy in-database models to OML Services

Enable programmatic deployment of in-database models

ONNX model support

Enable programmatic export of Python models to ONNX-format and deployment to OML Services

Additional algorithms

Exposed additional in-database algorithms, CUR Decomposition, O-Cluster, MSET

OML enhancements for Oracle Database 23c

- BOOLEAN data type support
- 4K columns per table
- Improved data prep for high cardinality categorical features
- Outlier detection using EM clustering
- Lineage: data query persisted with model

- Survival Analysis via XGBoost
- ESA with doc2vec
- New GLM link functions
- Monotone/interaction constraints via XGBoost
- Automated Time Series model search
- Multiple Time Series

Enhanced in-database ML algorithms

Explicit Semantic Analysis (ESA)
Text analytics feature extraction

Enriched text features for algorithm-integrated text processing

Dense projections with embeddings for doc2vec

Enhances use of, e.g., call center rep notes on customers or physician notes on patients along with other structured data

Extreme Gradient Boosting (XGBoost)
Classification and regression

Expand in-database XGBoost algorithm to support survival analysis

Add feature interaction constraints and monotonic constraints to limit variable interactions

Increase model accuracy when predicting, e.g., equipment failures and healthcare outcomes

Generalized Linear Model (GLM)
Classification

Adding link functions: probit, cloglog, cauchit

Enhanced support for binary targets and predictors with varying properties

Exponential Smoothing (ESM)
Time series forecasting

Automates hyperparameter search to produce better forecasting models without manual or exhaustive search

Enables non-expert users while increasing data scientist productivity

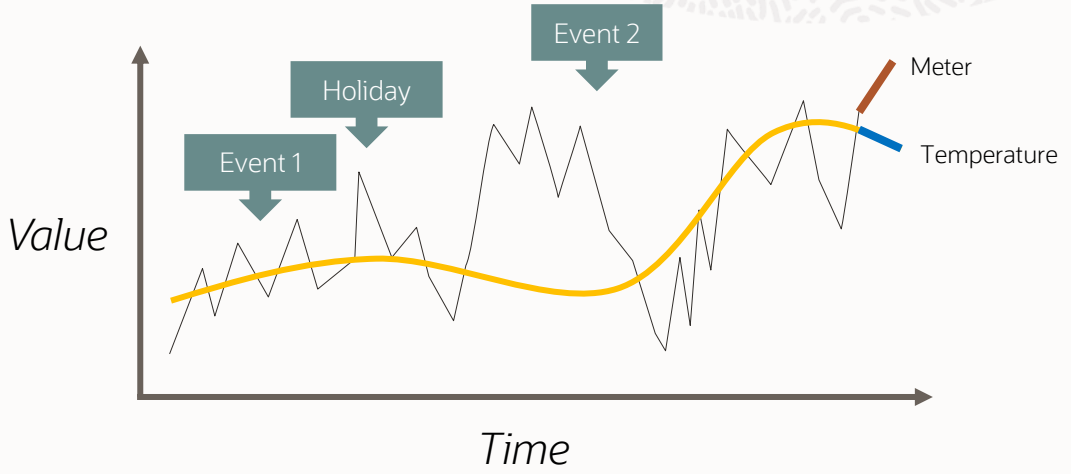
Multiple time series forecasting

Enabling Time Series Regression through component functionality

Supports conveniently generating backcasts and forecasts on one or more time series

"Target" time series also has confidence bounds

Result used as input to other ML algorithms, e.g., to support time series regression using XGBoost, where data set can be augmented categorical and numeric variables



Timestamp	Events	Holiday	...	Temp	DM$Temp	DM$Meter	Meter
<ts-1>	0	0		120	*Backcasts*	*Backcasts*	332
<ts-2>	0	1		122	...	...	347
<ts-3>	1	0		119	...	...	352
...	...	...		...	...	...	...
<ts-n+1>	Manual	Manual			*Forecasts*	*Forecasts*	Predicted target value via regression

Multiple time series forecasting

Code example

```
DECLARE
  v_setlst DBMS_DATA_MINING.SETTING_LIST;
BEGIN
  v_setlst('ALGO_NAME')             := 'ALGO_EXPONENTIAL_SMOOTHING';
  v_setlst('EXSM_INTERVAL')         := 'EXSM_INTERVAL_DAY';
  v_setlst('EXSM_MODEL')            := 'EXSM_ADDWINTERS_DAMPED';
  v_setlst('EXSM_SEASONALITY')      := '7';
  v_setlst('EXSM_PREDICTION_STEP') := '1';
  v_setlst('EXSM_SERIES_LIST')      := 'Temp';

  DBMS_DATA_MINING.CREATE_MODEL2(
         MODEL_NAME           => 'MTS_MODEL',
         MINING_FUNCTION      => 'TIME_SERIES',
         DATA_QUERY           => 'SELECT * FROM METER_READINGS',
         CASE_ID_COLUMN_NAME => 'Timestamp',
         TARGET_COLUMN_NAME   => 'Meter',
         SET_LIST             => v_setlst);
END;
```

OML AutoML UI

No-code user interface for automated machine learning

Experiment scheduling

Schedule experiments to run using the same jobs interface supporting OML Notebooks

Time series forecasting

Introduce automated time series machine learning technique

GPU support

Increase set of AI/ML use cases supported by ADB

Allocate GPU compute resources in OML Notebooks for OML4Py

Seamlessly use the Python interpreter for Python packages supporting GPU compute resources

Use GPU compute with embedded Python execution

On Autonomous Database, specify GPU compute resources when running user-defined Python functions with EPE.

Poll Question

4) Which of these roadmap items are you most looking forward to?

(check all that apply)

- OML AutoML experiment scheduling
- Multiple Time Series enabling Time Series Regression
- Automated Time Series search
- OML Data Monitoring no-code user interface
- GPU compute on OML Notebooks and OML4Py

For more information…

OML Webpage
https://oracle.com/machine-learning

OML Blog
https://bit.ly/omlblogs

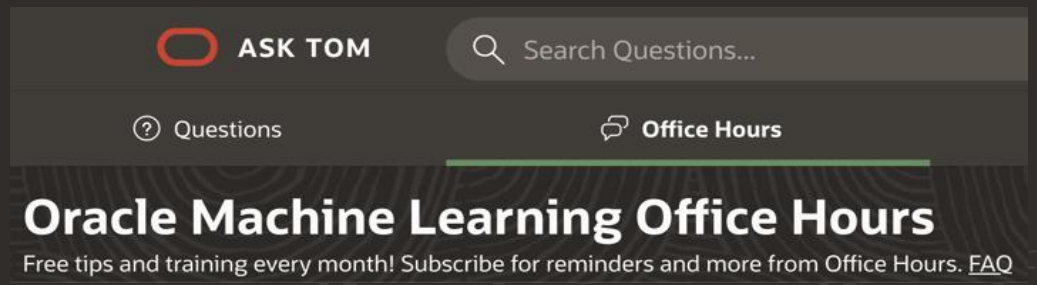
OML GitHub Repository
https://bit.ly/omlgithub

OML Office Hours
https://bit.ly/omlofficehours

Try on Oracle LiveLabs

Overview: https://bit.ly/omlfundamentalshol
OML4Py: https://bit.ly/oml4pyhol
All OML: https://bit.ly/omllivelabs

OML Documentation
https://docs.oracle.com/en/database/oracle/machine-learning



Q&A

Thank you

Mark Hornick

mark.hornick@oracle.com

@MarkHornick

MarkHornick

Group: Oracle Machine Learning