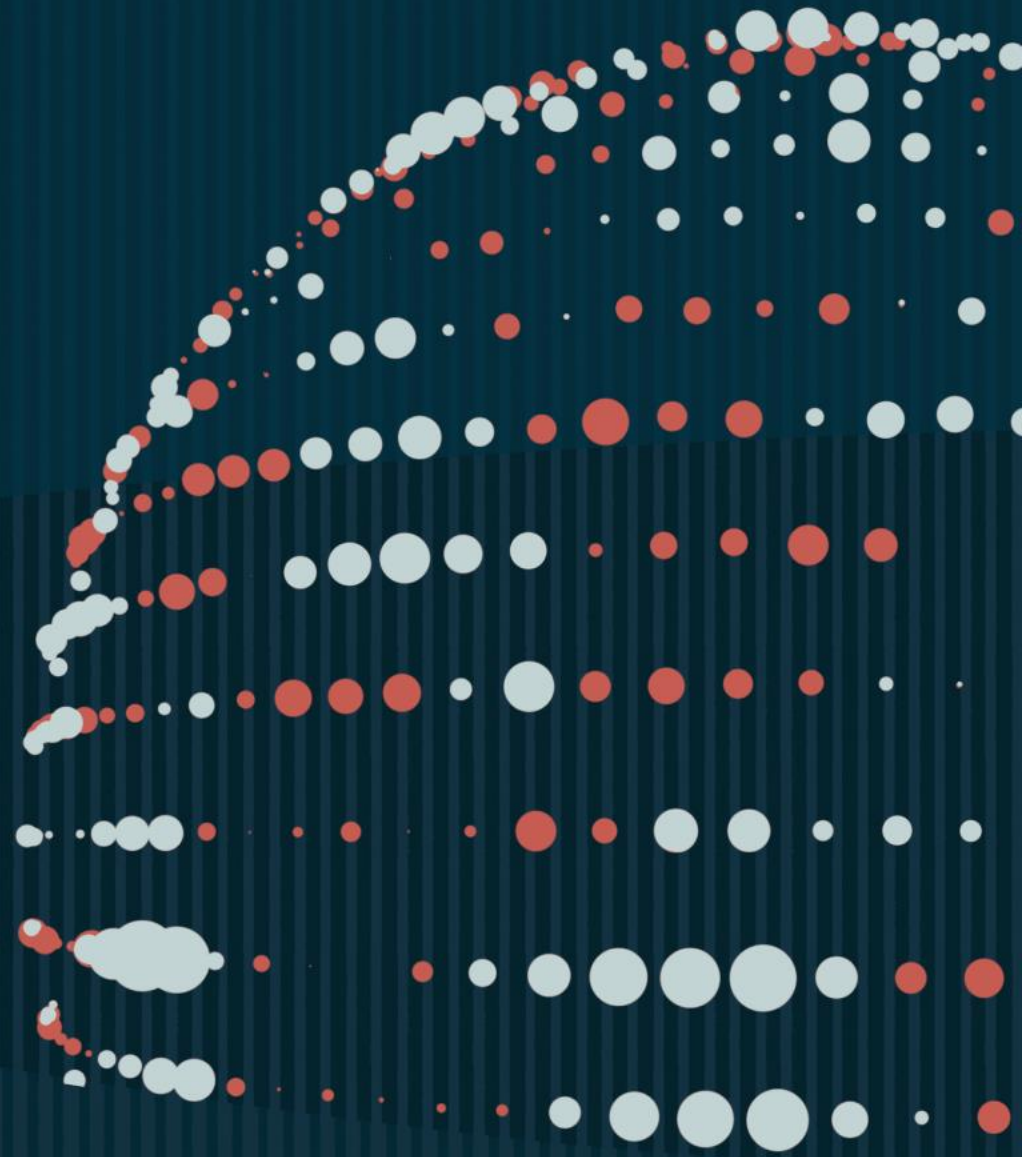
Getting started with Oracle Machine Learning on Autonomous Database

Move the Algorithms – Not the Data

Mark Hornick
Senior Director
Product Management
Machine Learning and AI

Sherry LaMonica
Consulting MTS
Product Management

January 31, 2025



Poll Question #1

Which of these products or features have you used? (select all that apply)

- Oracle Database
- Oracle Autonomous Database
- Oracle Machine Learning for SQL, R, or Python
- Oracle Machine Learning Notebooks
- Oracle Machine Learning AutoML UI
- Oracle Machine Learning Data or Model Monitoring
- Oracle Machine Learning Services

Oracle Autonomous Database

Using the Cloud to eliminate the complexity of data management

Oracle Autonomous Database = **Mission-Critical Database – Oracle Database** + **High Performance Engineered System: Oracle Exadata** + **Automation, Best Practices, AI and Data Tools**

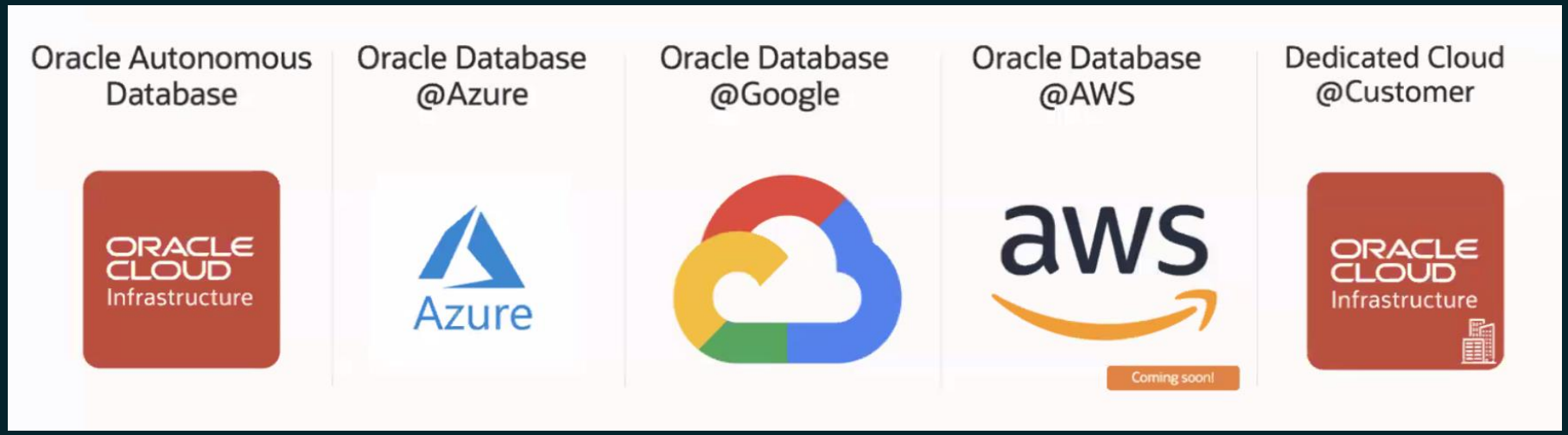
Autonomous Database

Oracle Database reimagined for the Cloud

- Automate the database management lifecycle
- Support mission-critical databases
- Enable you to innovate more, pay less, and support data security

Deployment choices for your cloud and data requirements

Autonomous Database runs where you need it



A platform for DS and ML

Robust, scalable, and capable of supporting end-to-end workflows

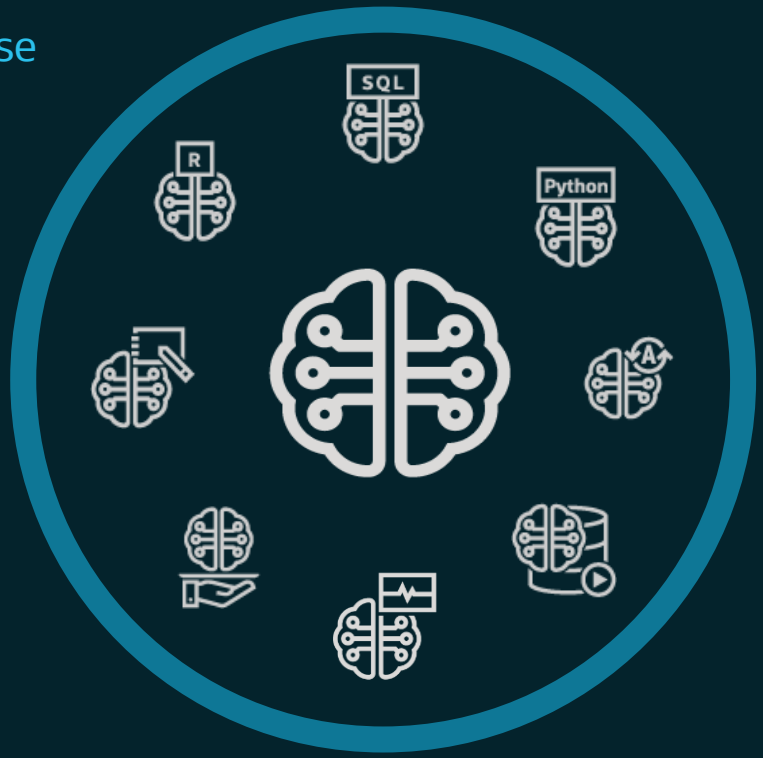
- Data Management
- Development Tools
- Machine Learning/Graph/Spatial
- Semantic Similarity Search
- Generative AI
- Monitoring
- Security and Compliance
- Infrastructure and Scalability
- Visualization and Reporting

Oracle Machine Learning

Develop data-driven business solutions using technology built into your database

Machine learning algorithms in the database kernel software

- Eliminate data movement
- Simplify solution architectures
- Speed solution development and deployment for SQL, R, Python, REST
- Augment applications and dashboards quickly and easily



Over 30 scalable in-database ML algorithms using SQL, R, and Python APIs, and automated no-code user interfaces, including AutoML

ML functionality in Oracle's converged database with spatial, graph, blockchain, AI Vector Search, JSON

Easily use with OAS/OAC, Oracle APEX, OCI DS, ODI, and others

Oracle Machine Learning database features included with **Oracle Database** and **Oracle Autonomous Database**

Evolution of in-database Machine Learning

Java API for in-database algorithms
Oracle Database 9iR2 - 2002

DM4J User Interface
Oracle Database 10R2 - 2005

DBMS_DATA_MINING
Oracle Database 10gR1 - 2004

Oracle Advanced Analytics Option
Oracle Database – 2011

Oracle Data Miner
SQL Developer Extension 11gR2 – 2010

OML for Python 1.0
ADB Serverless – 2021

OML Services
ADB Serverless – 2021

OML for Python 1.0
Oracle Database 2022

OML4R
ADB Serverless 2022

OML for Python 2.0
ADB Serverless – 2023

OML4R 2.0
ADB Serverless & ODB – 2023

OML Monitoring UI
ADB Serverless – 2023/24

OML Notebooks with GPUs
ADB Serverless – 2024

OML Notebooks 2.0
ADB Serverless - 2023

Oracle Data Miner
ADB - 2022

OML AutoML UI
ADB Serverless - 2021

OML Notebooks
ADB Serverless - 2022

Oracle R Enterprise 1.0
Oracle Database 11gR2 - 2011

DBMS_DATA_MINING
Oracle Database 11gR2 - 2009

DBMS_DATA_MINING
Oracle Database 11gR1 - 2007

Thinking Machines Corporation

Acquired in 1999

Machine learning horizontal use cases

Customers

- Segmentation
- Loyalty
- Lifetime value
- Retention/churn
- Acquisition
- Lead generation and prioritization
- Factors of best customers
- Invoice anomaly detection
- Fraud detection

Products

- Next best offer
- Cross-sell
- Upsell
- Product bundling
- Demand forecasting
- Inventory forecasting
- Revenue prediction

Equipment

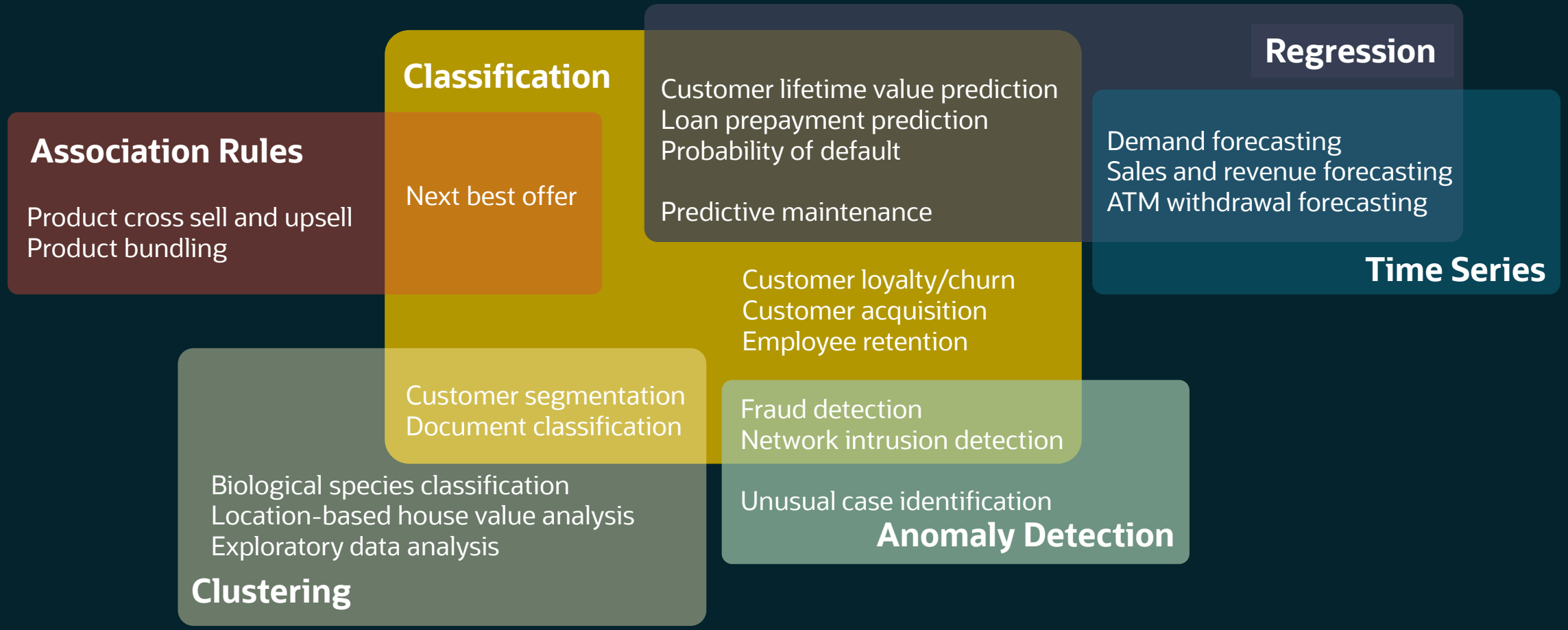
- Predictive maintenance
- Root cause analysis
- Demand forecasting
- Usage and sensor anomaly detection
- Failure risk assessment

Employees

- Attributes of best employees
- Retention
- Hiring
- Training needs analysis
- Survey feedback sentiment analysis
- Predicting hiring needs

Use cases and machine learning techniques

Address business problems that impact customers, products, operations, and employees



Oracle Machine Learning in-database ML algorithms

Classification

Decision Tree
Explicit Semantic Analysis
Logistic Regression (GLM)
Naïve Bayes
Neural Network
Random Forest
Support Vector Machine (SVM)
XGBoost (21c)

Clustering

Hierarchical K-Means
Hierarchical O-Cluster
Expectation Maximization

Row Importance

CUR Decomposition

Ranking

XGBoost (21c)

Regression

Generalized Linear Model (GLM)
Neural Network
Support Vector Machine (SVM)
Linear regression w/feature selection/generation
XGBoost (21c)

Feature Extraction

Principal Comp Analysis (PCA)
Non-negative Matrix Factorization
Singular Value Decomposition (SVD)
Explicit Semantic Analysis (ESA)

Attribute Importance

Minimum Description Length
Random Forest
Unsupervised Pairwise KL Divergence
CUR Decomposition

Time Series

Exponential Smoothing
Multiple Time Series (23ai) for TS Regression
Includes popular models
e.g. Holt-Winters with trends,
seasonality, irregular time series

Anomaly Detection

One-Class SVM
MSET-SPRT (21c)
Expectation Maximization (23ai)

Association Rules

A priori

Survival Analysis

XGBoost

Plus…
integrated text mining, explanatory prediction details, partitioned models, automatic data preparation

Oracle Machine Learning in-database ML algorithms

In-database, parallelized, distributed algorithms

- No extracting data to separate ML engines
- Scalability across cluster nodes
- Optimized memory utilization
- Scoring uses Exadata storage-tier function push-down optimizations
- Batch/real-time scoring

ML models as first-class database objects

- Access control per model
- Audit user actions
- Export / import models across databases
- Ease of backup, recovery, and security

Faster time-to-market through immediate solution deployment

- Use prediction operators in SQL queries
- Make predictions from R and Python
- Deploy models on ADB-S using no-code and REST interfaces

Oracle Machine Learning family of components

OML Component	Autonomous Database Serverless | DR	Autonomous Database Dedicated | C@C	Oracle Database on premises, DBDS, CS, CI, C@C
OML4SQL API Build ML models and score data with no data movement using SQL and PL/SQL	✓	✓	✓
OML4Py API Leverage the database as a high-performance compute engine from Python with in-database ML	✓		✓
OML4R API Leverage the database as a high-performance compute engine from R with in-database ML	✓		✓
OML Notebooks SQL, PL/SQL, Python, R, conda, and markdown interpreters	✓		
OML AutoML UI No-code automated modeling interface	✓		
OML Monitoring No-code user interface for monitoring changes in data and in-database ML model quality	✓		
OML Services RESTful model management, deployment, monitoring, data bias detection	✓		
Oracle Data Miner SQL Developer extension with a drag-n-drop interface for creating ML methodologies	✓	✓	✓

Demonstration
Accessing the OML UI on ADB

OML Notebooks on ADB Serverless

Collaborative notebook environment

Rich visualization

Text, Line Chart, Area Chart, Bar Chart, Pyramid Chart, Pie Chart, Donut Chart, Funnel Chart, Tag Cloud, TreeMap, Sunburst, Scatter Plot, Box Plot

Paragraph comments and runtime dependencies

Zeppelin and Jupyter layouts, with import/export

On-page notebook versioning, history viewing, and version comparison

Use third-party R and Python packages via conda environments

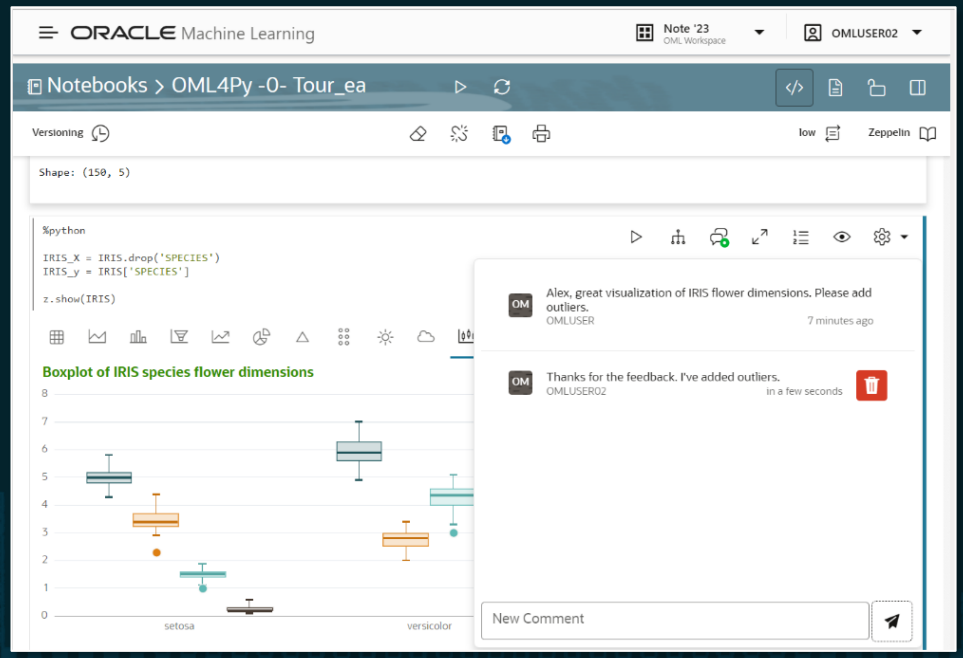
Leverage GPUs in OML Notebooks with Python interpreter and GPU-enabled packages

Access over 150 example template notebooks

Save notebooks as personal/shared template

Schedule notebooks to run with email notifications

Manage workspace user permissions



Oracle Machine Learning for SQL

Empower data scientists and SQL users with native SQL and PL/SQL ML functionality

Build and evaluate ML models using in-database algorithms

DBMS_DATA_MINING package supports model building

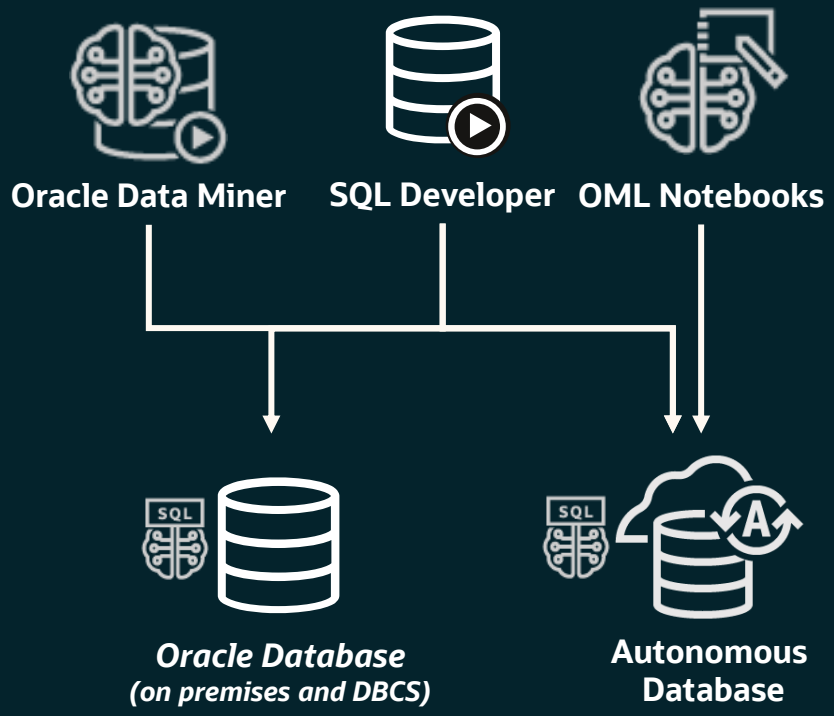
Models immediately available for scoring using SQL queries with integrated SQL operators

Model detail views provide inside look at model performance, metadata, and patterns – as applicable per algorithm

Export/import models across database instances and to OML Services

Use dense vector embeddings as input to multiple in-database algorithms

Import ONNX-format models, including text and image transformers, for native use within SQL queries



OML4SQL – model building

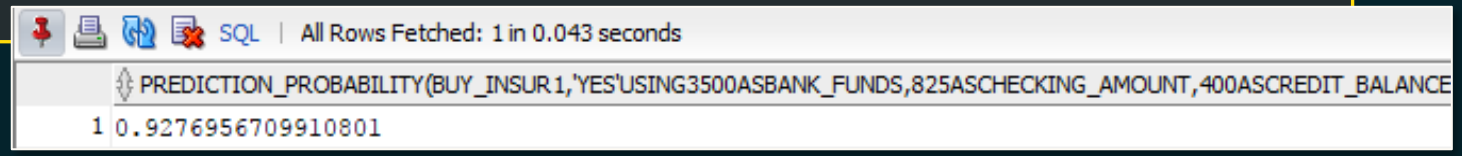
Example – Predict Customer Behavior

Build a machine learning model to determine which customers are likely buy travel insurance

```
DECLARE
    v_setlst DBMS_DATA_MINING.SETTING_LIST;
BEGIN
    v_setlst('ALGO_NAME')  := 'ALGO_SUPPORT_VECTOR_MACHINES';
    V_setlst('PREP_AUTO')  := 'ON';
    DBMS_DATA_MINING.CREATE_MODEL2(
        MODEL_NAME          => 'BUY_TRAVEL_INSUR',
        MINING_FUNCTION     => 'CLASSIFICATION',
        DATA_QUERY          => 'select * from CUSTOMERS',
        SET_LIST            => v_setlst,
        CASE_ID_COLUMN_NAME => 'CUST_ID',
        TARGET_COLUMN_NAME  => 'BUY_TRAVEL_INSURANCE');
END;
```

Apply a machine learning model to predict probability that individual customer is likely to buy

```
SELECT prediction_probability(BUY_TRAVEL_INSURANCE, 'Yes'
  USING 3500 as bank_funds, 1987 as cust_year_of_birth, 'Married' as marital_status, 2 as num_previous_cruises)
FROM dual;
```



SQL API – explanatory prediction details

Which predictors most contribute to individual prediction results

Generate predictions and extract prediction details from XML table

```
SELECT CUST_ID, round(BUY_INSURANCE_PRED,3) BUY_INSURANCE_PRED,
       RTRIM(TRIM(SUBSTR(OUTPRED."Attribute1",17,100)),'rank="1"/>') FIRST_ATTRIBUTE,
       RTRIM(TRIM(SUBSTR(OUTPRED."Attribute2",17,100)),'rank="2"/>') SECOND_ATTRIBUTE
FROM (SELECT CUST_ID, PREDICTION(BUY_TRAVEL_INSUR USING *) BUY_INSURANCE_PRED,
             PREDICTION_DETAILS(BUY_TRAVEL_INSUR USING *) PD
      FROM TEST_DATA
      ORDER BY CUST_ID) OUT,
      XMLTABLE('/Details' PASSING OUT.PD COLUMNS
                "Attribute1" XMLType PATH 'Attribute[1]’,
                "Attribute2" XMLType PATH 'Attribute[2]’)  OUTPRED;
```

View results

CUST_ID	BUY_INSURANCE_PRED	FIRST_ATTRIBUTE	SECOND_ATTRIBUTE
100005	1	"CUST_YEAR_OF_BIRTH" actualValue="1957" weight="2.496"	"YRS_RESIDENCE" actualValue="5" weight="1.696"
100009	0	"CUST_YEAR_OF_BIRTH" actualValue="1974" weight="3.291"	"Y_BOX_GAMES" actualValue="1" weight="2.209"
100010	0	"CUST_YEAR_OF_BIRTH" actualValue="1975" weight="2.357"	"YRS_RESIDENCE" actualValue="3" weight="1.338"
100012	1	"CUST_YEAR_OF_BIRTH" actualValue="1968" weight="1.144"	"YRS_RESIDENCE" actualValue="4" weight=".355"

SQL API – partitioned models

Single model for scoring comprised of models each built on a data partition

Build a partitioned model on column gender

```
DECLARE
    v_setlst DBMS_DATA_MINING.SETTING_LIST;
BEGIN
    v_setlst('ALGO_NAME')             := 'ALGO_SUPPORT_VECTOR_MACHINES';
    v_setlst('SVMS_KERNEL_FUNCTION')  :='SVMS_LINEAR';      -- choose linear kernel for coefficients
    v_setlst('ODMS_PARTITION_COLUMNS'):='CUST_GENDER';      -- choose partition column - gender

    DBMS_DATA_MINING.CREATE_MODEL2(MODEL_NAME          => 'SVM_MOD_PARTITIONED',
                                   MINING_FUNCTION     => 'REGRESSION',
                                   DATA_QUERY          => 'SELECT * FROM CUSTOMERS_DEMO',
                                   SET_LIST            => v_setlst,
                                   CASE_ID_COLUMN_NAME => 'CUST_ID',
                                   TARGET_COLUMN_NAME  => 'YRS_RESIDENCE');
END;
```

View partitioned sub-models

```
%sql

SELECT *
FROM ALL_MINING_MODEL_PARTITIONS
WHERE MODEL_NAME = 'SVM_MOD_PARTITIONED';
```

settings

OWNER	MODEL_NAME	PARTITION_NAME	POSITION	COLUMN_NAME	COLUMN_V
OMLUSER	SVM_MOD_PARTITIONED	M	1	CUST_GENDER	M
OMLUSER	SVM_MOD_PARTITIONED	F	1	CUST_GENDER	F

Oracle Machine Learning for Python and R

Empower data scientists and Python/R users with open-source environments

Oracle Database as HPC engine via proxy objects and function overloading

In-database machine learning algorithms from Python/R

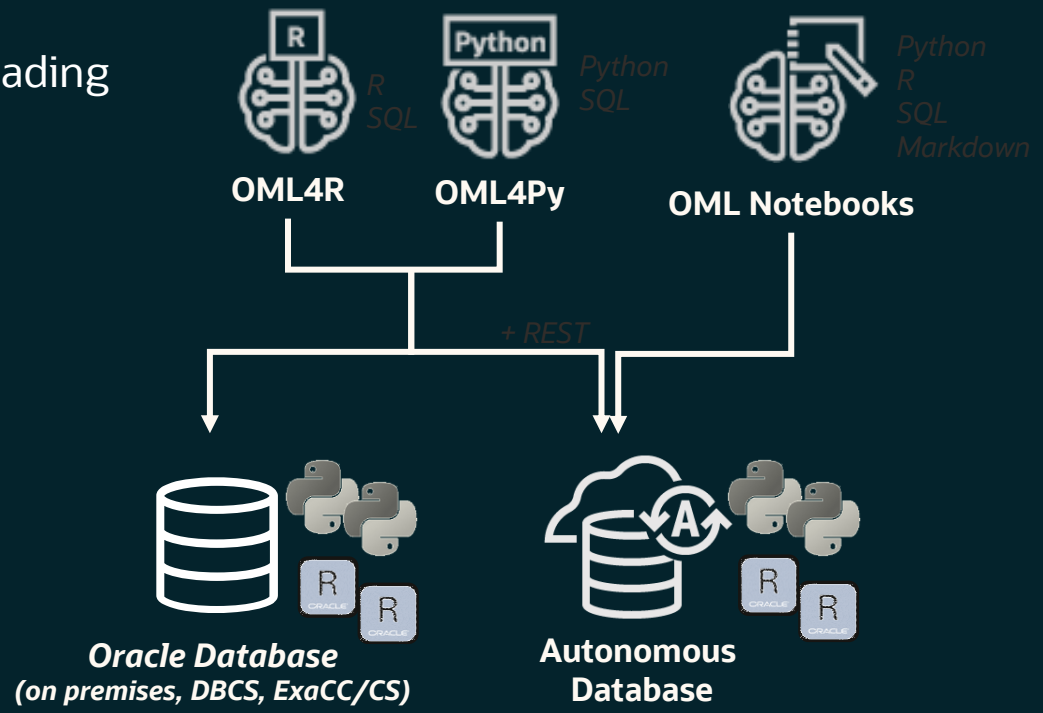
OML4Py automated machine learning

Manage Python/R user-defined functions/objects in the database

Integrate results into applications and dashboards via SQL or REST

Augment functionality with third-party Python and R packages

No need to provision Python or R engines for solution deployment



Build in-database models from OML APIs: SQL, R, Python

Build a machine learning model to determine which customers are likely to buy travel insurance

PL/SQL

```
DECLARE
  v_setlst DBMS_DATA_MINING.SETTING_LIST;
BEGIN
  v_setlst('ALGO_NAME')  := 'ALGO_SUPPORT_VECTOR_MACHINES';
  v_setlst('PREP_AUTO')  := 'ON';
  DBMS_DATA_MINING.CREATE_MODEL2(
        MODEL_NAME          => 'BUY_TRAVEL_INSURANCE',
        MINING_FUNCTION     => 'CLASSIFICATION',
        DATA_QUERY          => 'select * from CUSTOMERS',
        SET_LIST            => v_setlst,
        CASE_ID_COLUMN_NAME => 'CUST_ID',
        TARGET_COLUMN_NAME  => 'BUY_TRAVEL_INSURANCE');
END;
```

```
ore.sync(table="CUSTOMERS")
settings = list(model_name="BUY_TRAVEL_INSURANCE")
svm.mod <- ore.odmSVM(BUY_TRAVEL_INSURANCE~.,CUSTOMERS,
                      odm.settigs = settings)
```

```
CUSTOMERS = oml.sync(table="CUSTOMERS")
X = CUSTOMERS.drop("BUY_TRAVEL_INSURANCE")
y = CUSTOMERS["BUY_TRAVEL_INSURANCE"]
svm_mod = svm()
svm_mod = svm_mod.fit(X, y,
                      model_name = 'BUY_TRAVEL_INSURANCE')
```

Apply a machine learning model to predict probability that individual customer is likely to buy

```
SELECT prediction_probability(BUY_TRAVEL_INSURANCE, 'Yes'
  USING 98400 as income, 45 as age, 'Married' as marital_status, 2 as num_previous_cruises)
FROM dual;
```

SQL | All Rows Fetched: 1 in 0.043 seconds

	PREDICTION_PROBABILITY(BUY_INSUR1,'YES'USING:
1	0.9276956709910801

Datastore for Python and R object persistence

Python: **oml.ds.save**() **oml.ds.load**()

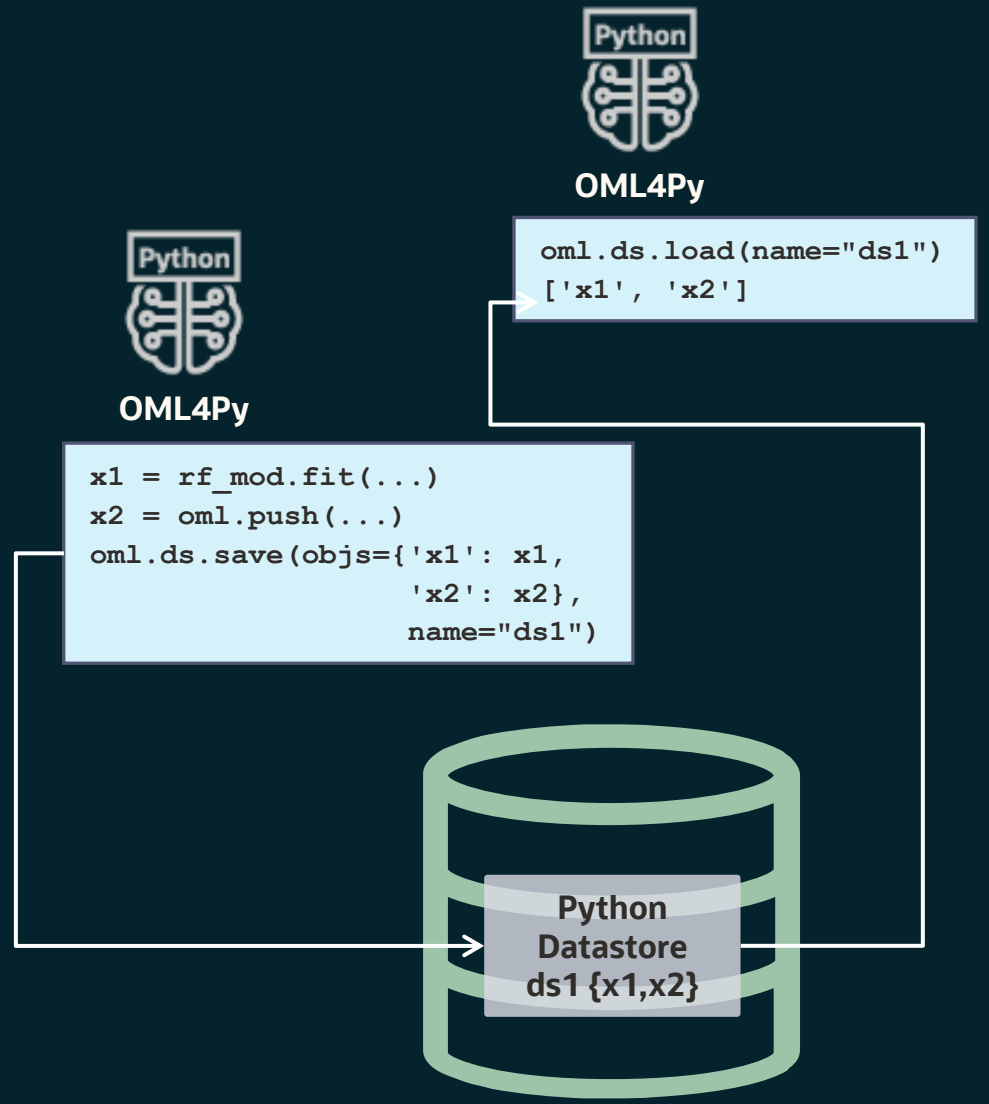
R: **ore.save**() **ore.load**()

Provide database storage to save/load objects

- Python and OML4Py objects across Python sessions
- R and OML4R objects across R sessions

Use cases

- Persist R and Python objects to reuse or share
- Pass arguments to user-defined Python and R functions for embedded execution, especially non-scalar objects with SQL and REST invocation
- Preserve OML4Py and OML4R proxy objects across sessions



Demonstration
Using OML Notebooks

Automated Machine Learning (AutoML) simplifies the modeling process

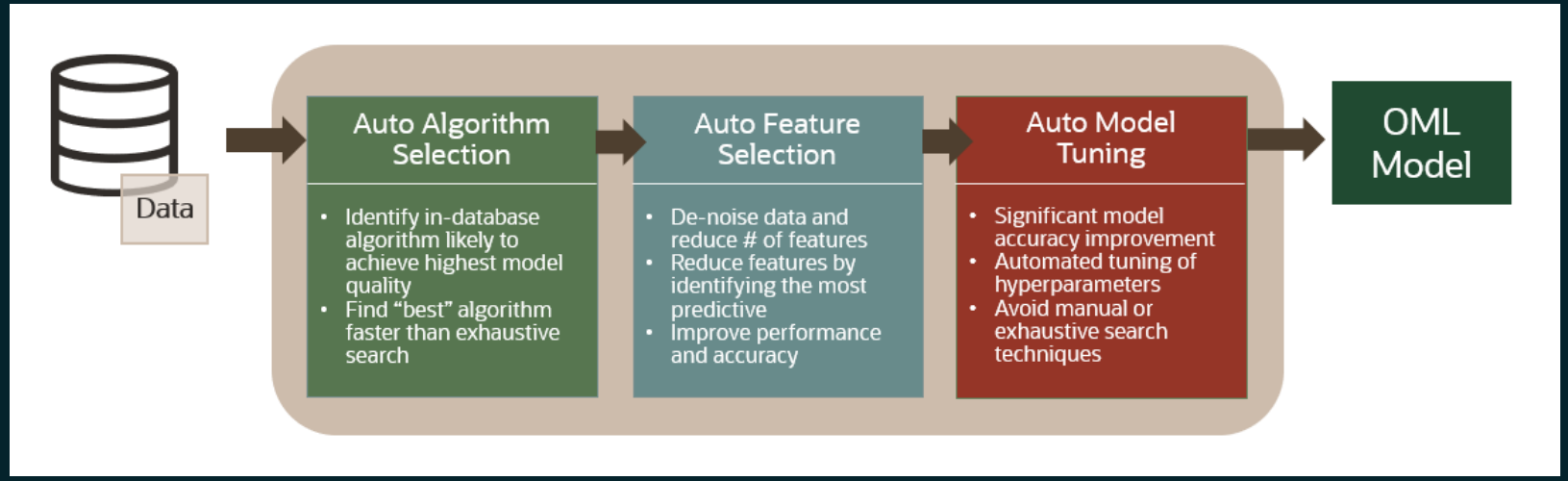
Enhance data scientist productivity and help non-experts produce ML models

Eliminate repetitive and time-consuming tasks of model building and evaluation

No need to understand how to tune each algorithm's specific hyperparameters

Reduce compute time and cost

- Apply ML to the ML process
- Reduce algorithm and hyperparameters search space



Click your way to an ML model

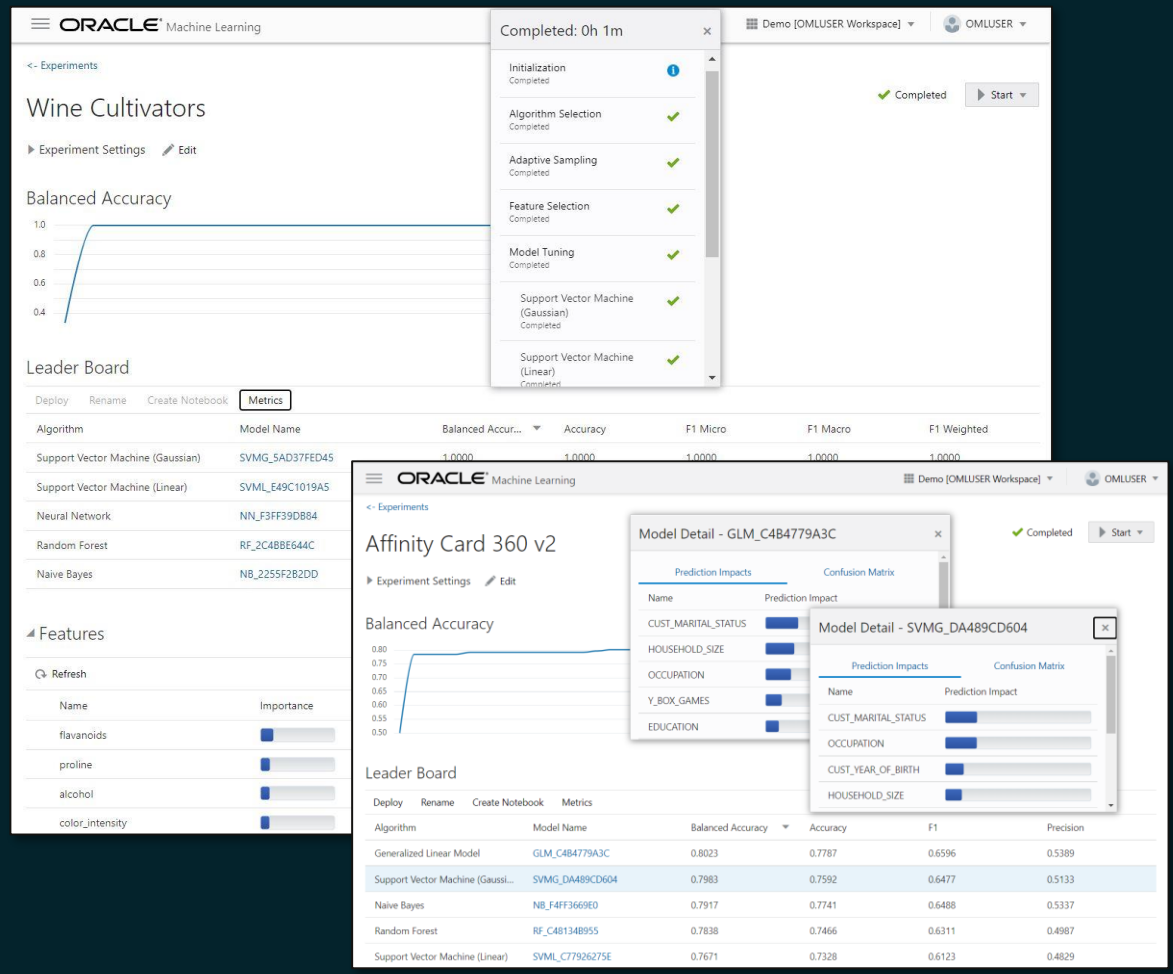
OML AutoML UI accelerates model building with a no-code interface

Select the prepared data table and the column you want to predict

Start automated build and compare of multiple models with model quality metrics

Generate editable notebooks for desired models with AutoML-selected hyperparameter values

Deploy models immediately using SQL or deploy to OML Services as REST endpoints



Enhance data scientist productivity and help non-experts produce ML models

Demonstration

OML AutoML UI to generate a notebook

OML Services for MLOps support on ADB Serverless

Manage and use ML models from REST endpoints for ease of application integration

Model Management		Deployment		Monitoring		Bias Detection
Store/ Organize	Version/ Compare	Real-time Scoring	Batch Scoring	Data Monitoring	Model Monitoring	Data Bias Detection

Lightweight scoring using REST endpoints

Real-time data scoring for streaming and other applications

Singleton, small batch, and full batch scoring

Data and model monitoring

Deploy in-database (native format) and third-party (ONNX format) models

Supports classification, regression, clustering and feature extraction models

Pay only for actual scoring compute – no separate VM provisioning or management

BYOM: Import embedding models to your 23ai database

Load text transformer models as first-class database objects for use with AI Vector Search

OML4Py oml.util package

- Generate portable ONNX model from a **Hugging Face** repository transformer
- Save model to database for use with in-database **ONNX Runtime**

```
import oml
from oml.utils import EmbeddingModel

em = EmbeddingModel(model_name="sentence-transformers/all-MiniLM-L6-v2")

em.export2db("sentence-transformer")
# or
em.export2file("sentence-transformer")
```

OML4SQL DBMS_DATA_MINING package

- Get pre-created ONNX-format bundle from Object Storage
- Import model to database for use with in-database **ONNX Runtime**


```
DECLARE
  model_source BLOB := NULL;
BEGIN
  model_source := DBMS_CLOUD.get_object(
    credential_name => 'OBJ_STORE_CRED',
    object_uri      =>
      'https://objectstorage…bucketname/o/sentence-transformer-bundle.onnx');

  DBMS_DATA_MINING.import_onnx_model(
    model_name => "sentence-transformer",
    model_data => model_source,
    metadata   => JSON('{ function : "embedding" }'));
END;
```

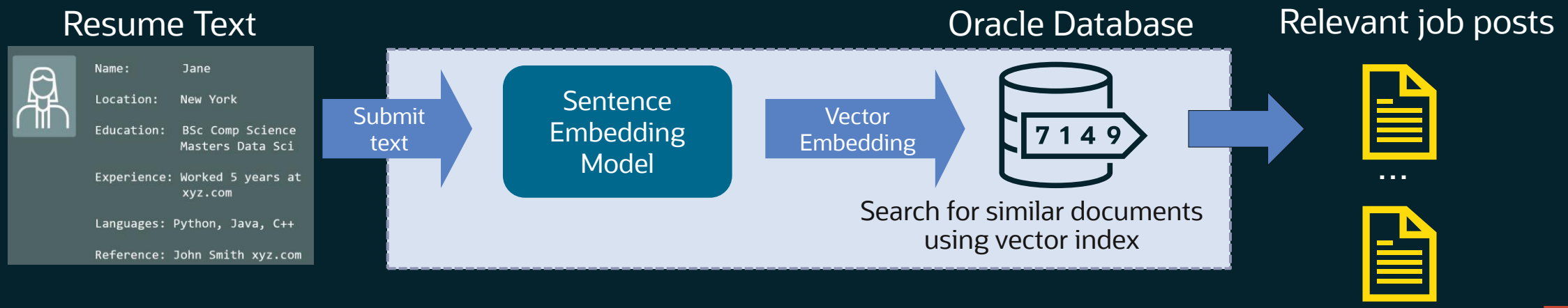
Document similarity search using embeddings in query

Get top 10 job posts for a software engineer in New York ordered by relevance of the resume

```
CREATE TABLE job_post_vector_table AS
  SELECT id, job_title, location, job_post,
    VECTOR_EMBEDDING(sentence_transformer
                     USING job_post) AS embedding
  FROM job_posts_table;
```

```
CREATE VECTOR INDEX vector_idx ON
  job_post_vector_table (embedding)
  ORGANIZATION INMEMORY NEIGHBOR GRAPH;
```

```
SELECT id, job_post FROM job_post_vector_table t
WHERE t.job_title = 'Software Engineer'
AND   t.location = 'New York'
ORDER BY VECTOR_DISTANCE(
           embedding,
           VECTOR_EMBEDDING(sentence_transformer
                            USING :resume)) ASC
FETCH APPROX FIRST 10 ROWS ONLY;
```



BYOM: Import ONNX-format ML models to your 23ai database

Load traditional ML models as first-class database objects for in-database prediction

Supports machine learning techniques **classification**, **regression**, and **clustering**

Make predictions using in-database **ONNX Runtime**

Imported ONNX-format models behave like **native in-database** models

Use **same prediction operators** as for other in-database models

```
DECLARE
  model_source BLOB := NULL;
BEGIN
  model_source := DBMS_CLOUD.get_object(
    credential_name => 'OBJ_STORE_CRED',
    object_uri      => 'https://objectstorage…bucketname/o/classification-churn-bundle.onnx');

  DBMS_DATA_MINING.import_onnx_model(
    model_name => "churn_model",
    model_data => model_source,
    metadata   => JSON('{ function : "classification" }'));
END;
SELECT case_id, PREDICTION(churn_model using *) from CUSTOMERS
```

OML4Py Spatial AI

Spatial machine learning algorithms

Points

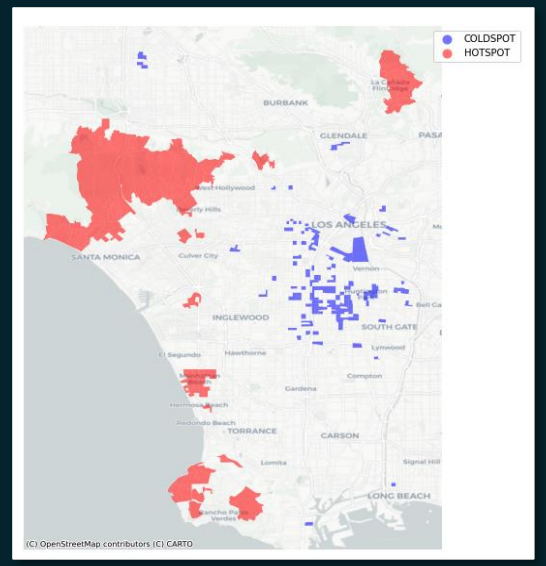
Lines

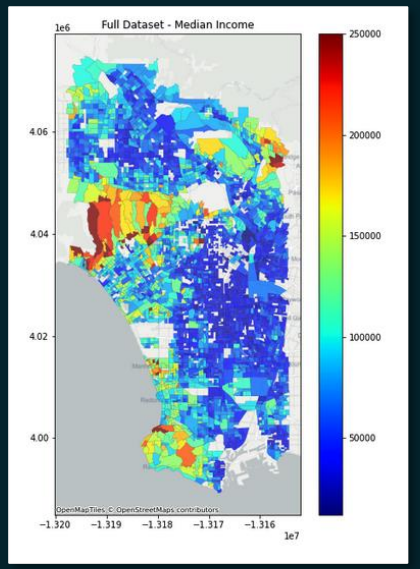
Polygons

Where are crimes clustered?
Where are the unemployment hotspots?

What is the predicted impact of a highway project on traffic safety?

What is the predicted solar energy adoption for a planned housing subdivision?





OML4Py Spatial AI Algorithms

Classification

Logistic SLX classifier
GWR Classifier
Geographical Classifier

Clustering

LISA Hotspot
DBSCAN with Regionalization
Agglomerative with Regionalization

Anomaly Detection

LocalOutlierFactor

Feature Engineering

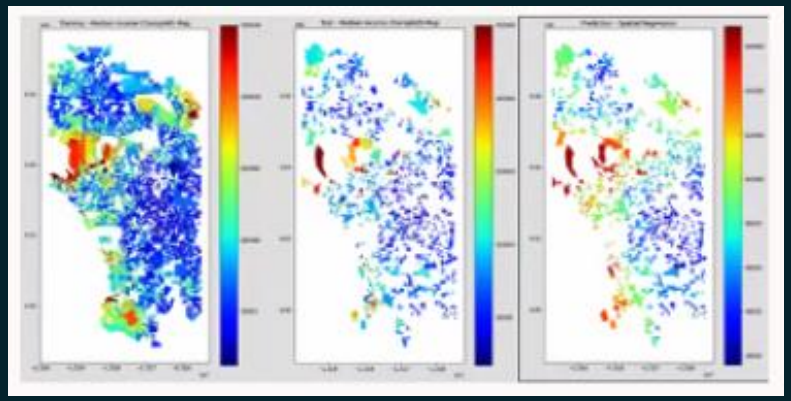
Spatial Lag Transformer
Categorical Lab Transformer
Spatial Coordinates Transformer
Spatial Imputer

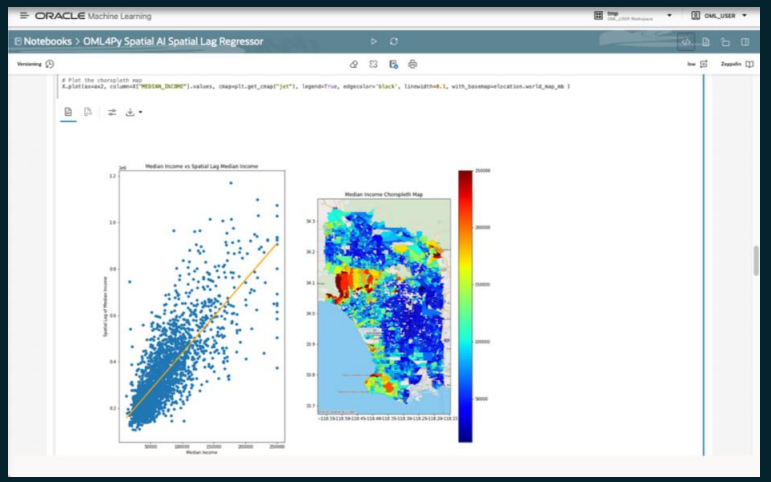
Regression

Spatial Cross-regressive Model (SLX)
Spatial Lag Model (SAR)
Spatial Error Model (SEM)
Geographical Regressor (GR)
Geographic Weighted Regression (GWR)
Spatial Regimes (OLS Regimes)
Spatial Fixed Effects (SFE)

Adaptive Spatial Modeling

Automatically search and evaluate different algorithms to find model for a specific application
Used when one does not know which regression algorithm to use or which may be better





Poll Question #2

Which topics would you most like to see in upcoming OML Office Hours? (select up to 3)

- OML4SQL
- OML4R
- OML4Py
- OML Notebooks
- OML AutoML UI
- OML Data Monitoring
- OML Model Monitoring
- OML Services
- OML Services Data Bias Detection
- Deeper dive on “Bring Your Own Model”

Oracle Machine Learning

Benefits summary

Simpler solution architecture and management
using Oracle's converged database

Multiple language APIs
for SQL, Python, R, REST

Eliminate data movement
for database data

Bring Your Own Model
using ONNX-format, R, and Python

Automated Machine Learning (AutoML)
through Python and, on ADB, a no-code UI

Data and model governance
through Oracle-enabled security

Simple pricing structure - ML capabilities
in core product at **no additional cost**

Scalable and high-performance modeling
and scoring, with **elastic scaling** on ADB

Flexible development, test, and deployment
in cloud, on-premises, and hybrid environments

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