ORACLE

DatabaseWorld @ CloudWorld

Oracle Autonomous Database as a Platform for Data Science and Machine Learning

DIG3519

Mark Hornick
Senior Director
Oracle Machine Learning
September 2023

Oracle Autonomous Database

Using the Cloud to eliminate the complexity of data management

Oracle Autonomous Database = **Complete Infrastructure Automation** + **Complete Database Automation** + **Automated Data Center Operations**

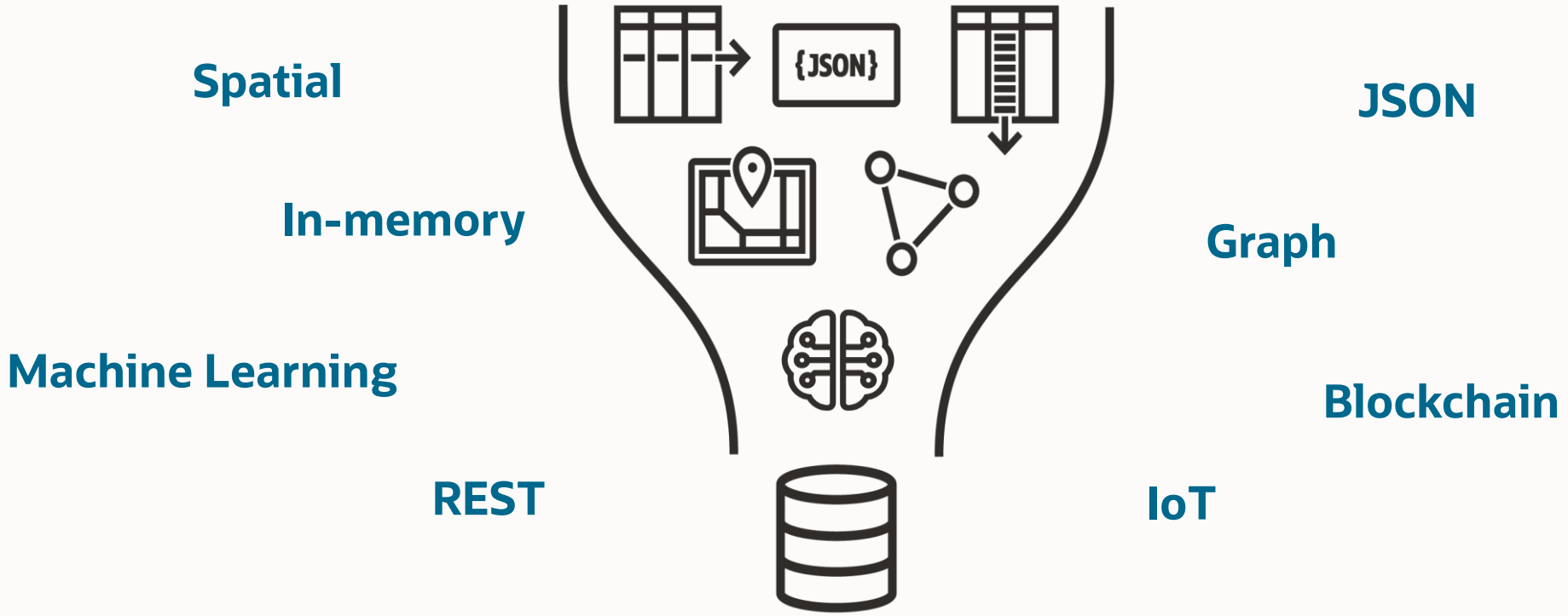
Autonomous Database

Oracle Database reimagined for the Cloud

- Completely automating the full database management lifecycle
- Supporting mission-critical databases
- Enabling you to innovate more, pay less, and ensure data security

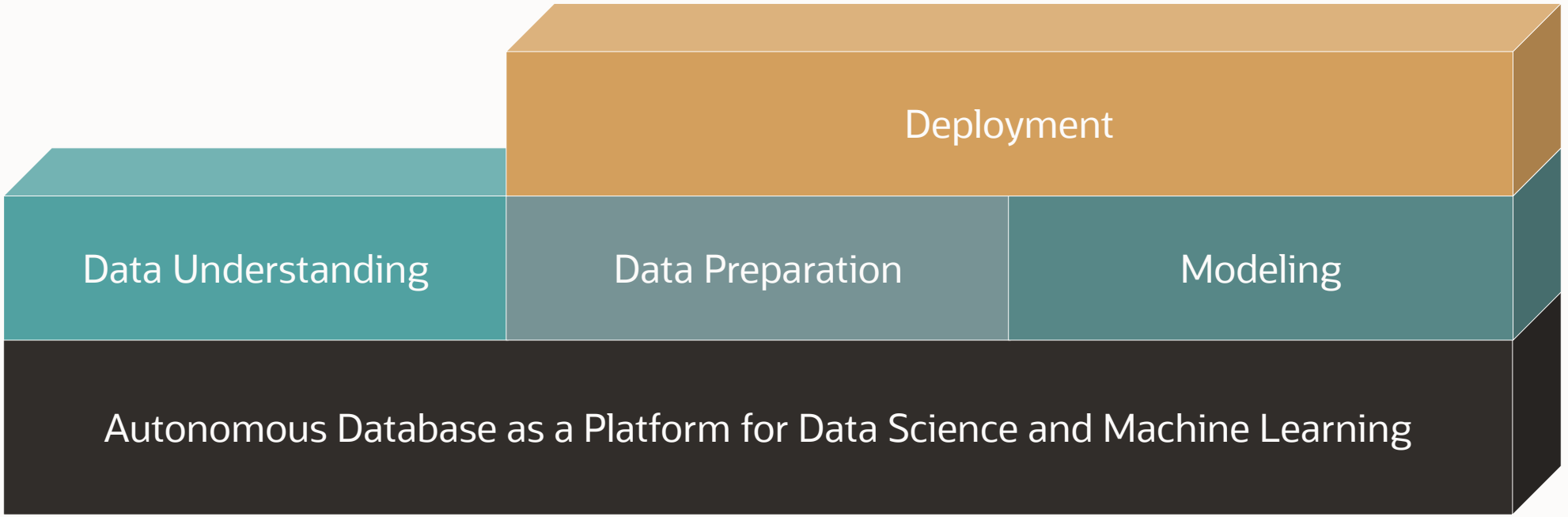
Oracle's *Converged* Database

Instead of single-use proprietary databases run converged, open Database

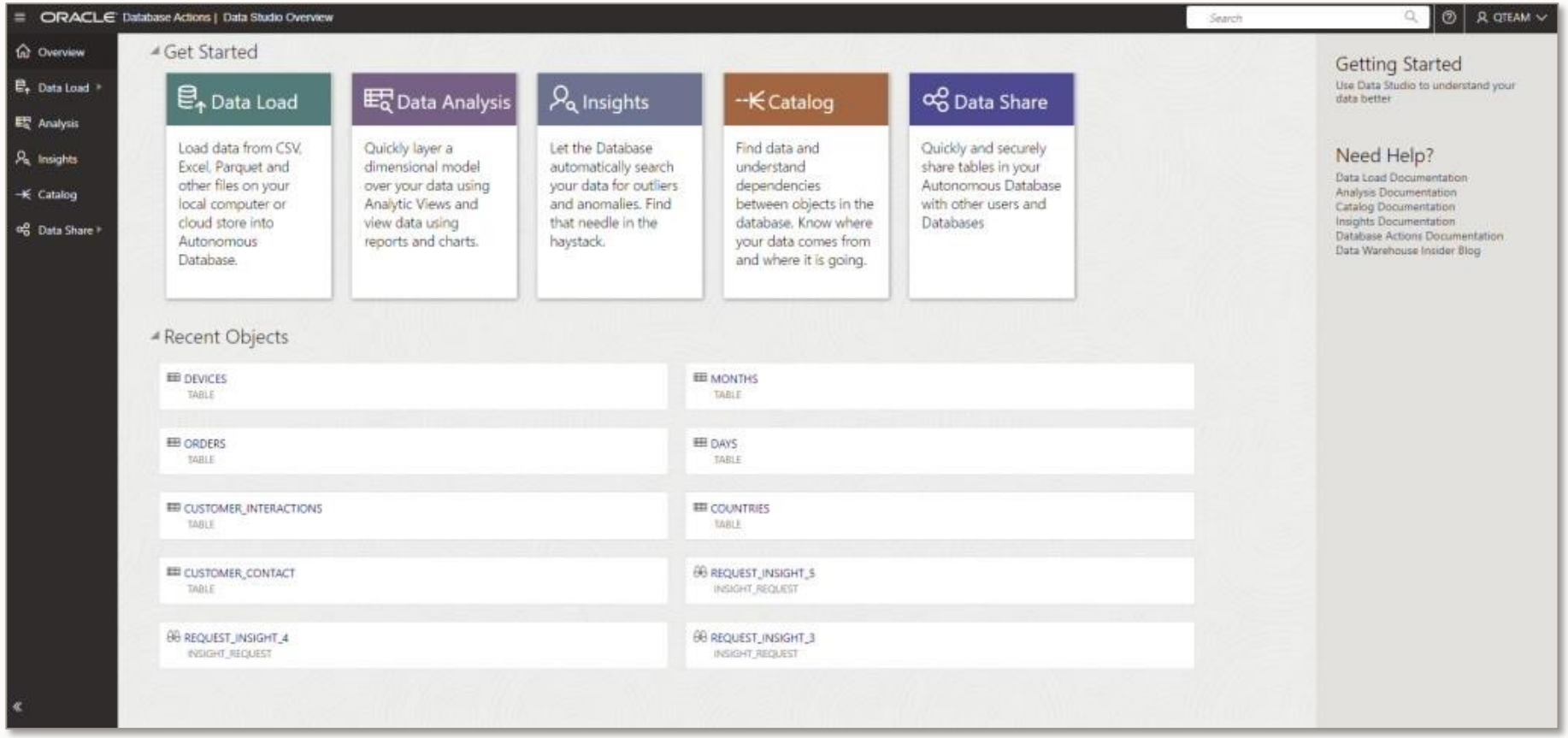


All modern data types, analytics, and the latest development paradigms built into one product at no additional cost

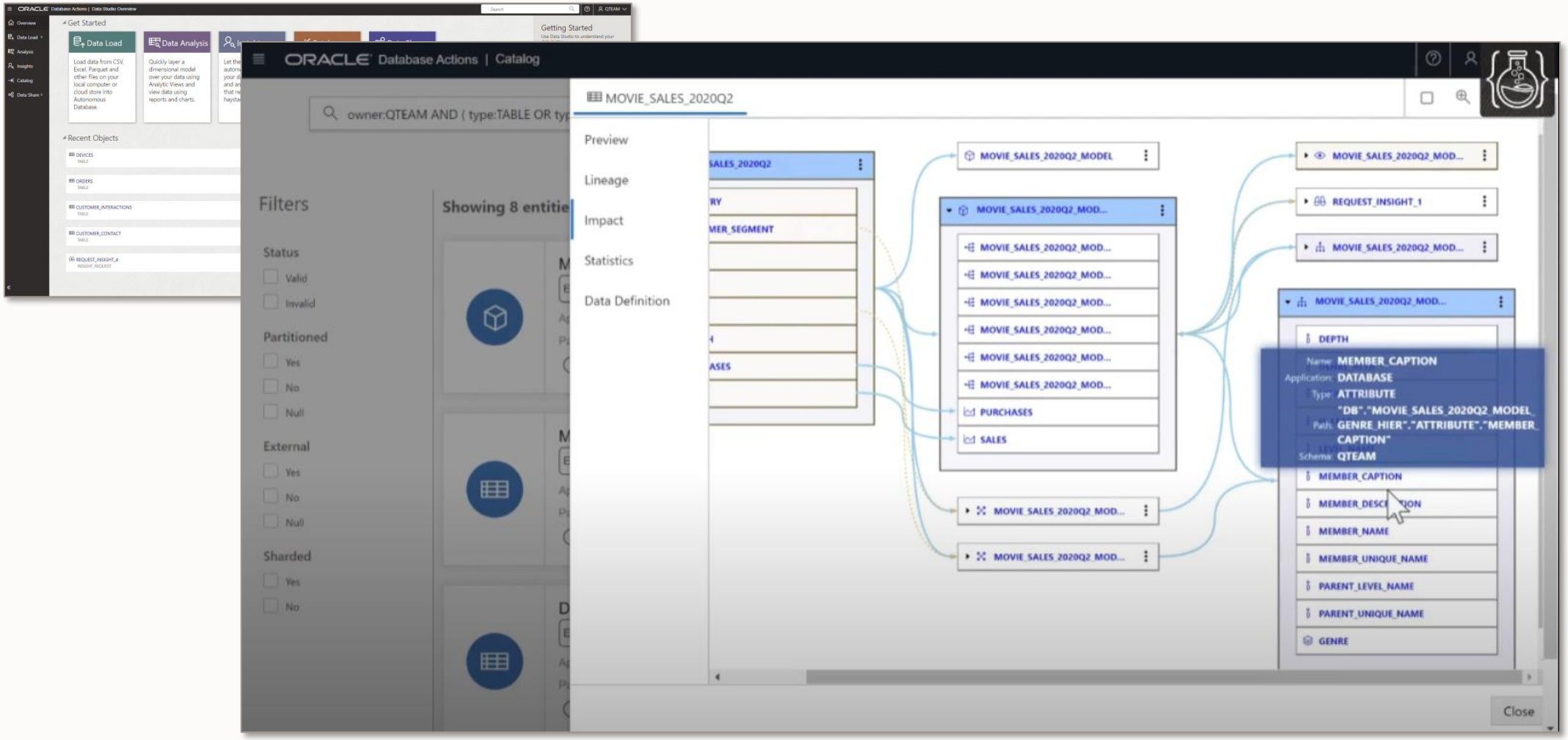
Autonomous Database as a Platform for Data Science and Machine Learning



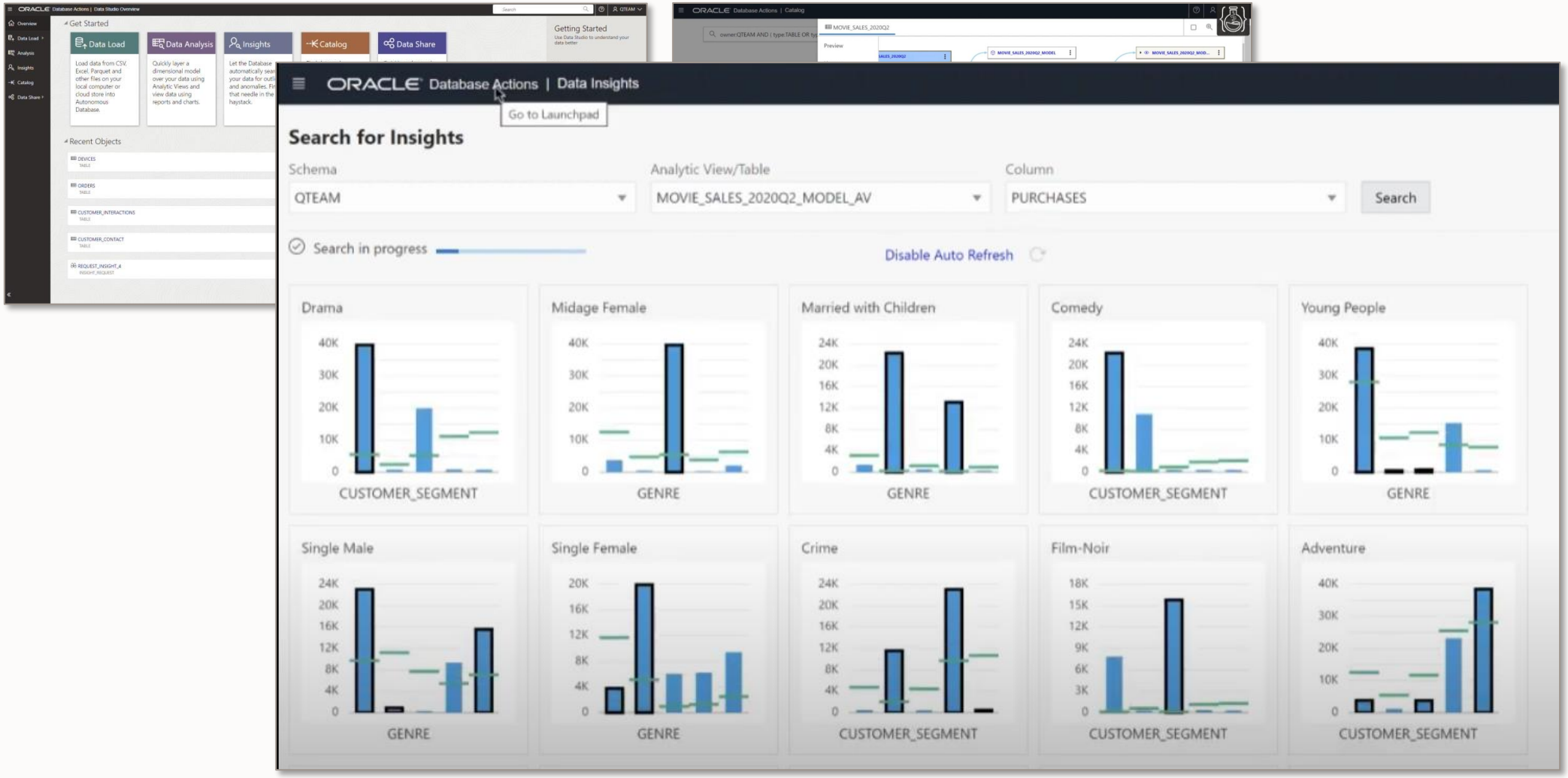
Data Studio Tools



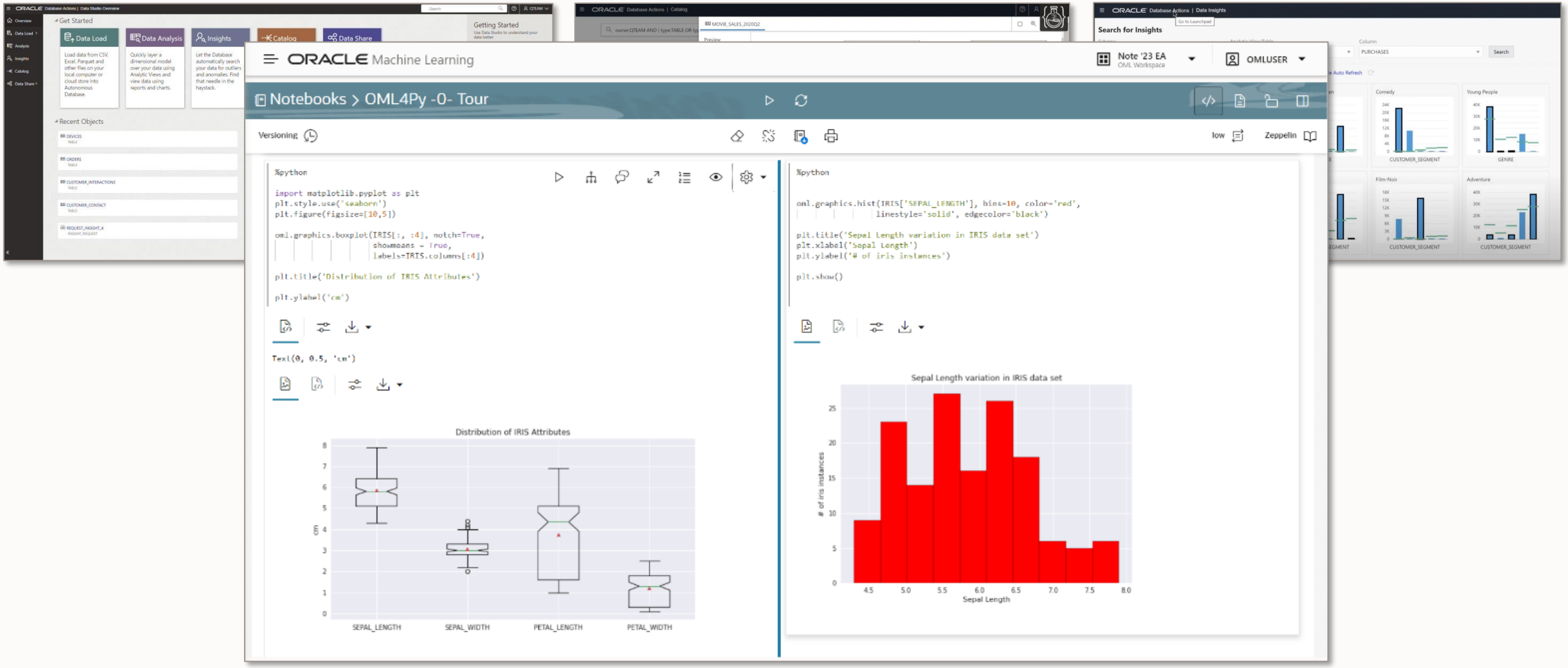
Catalog



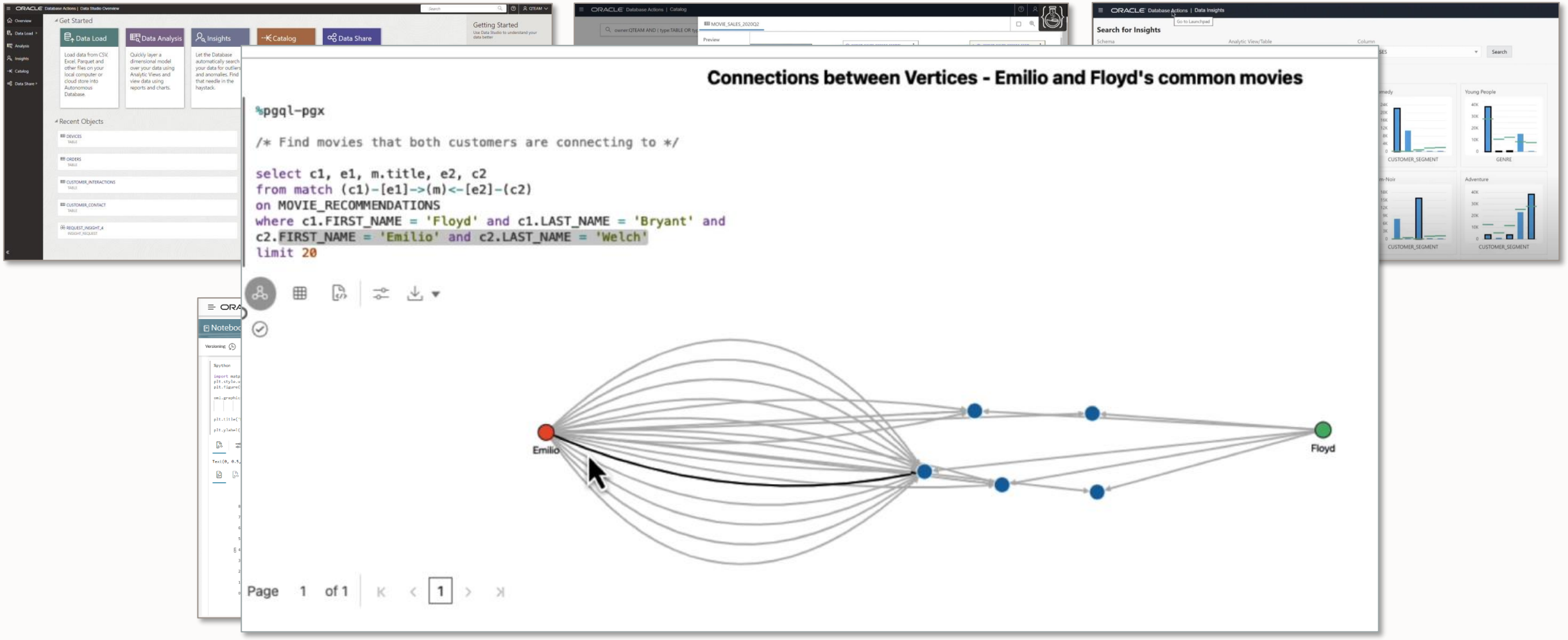
Insights



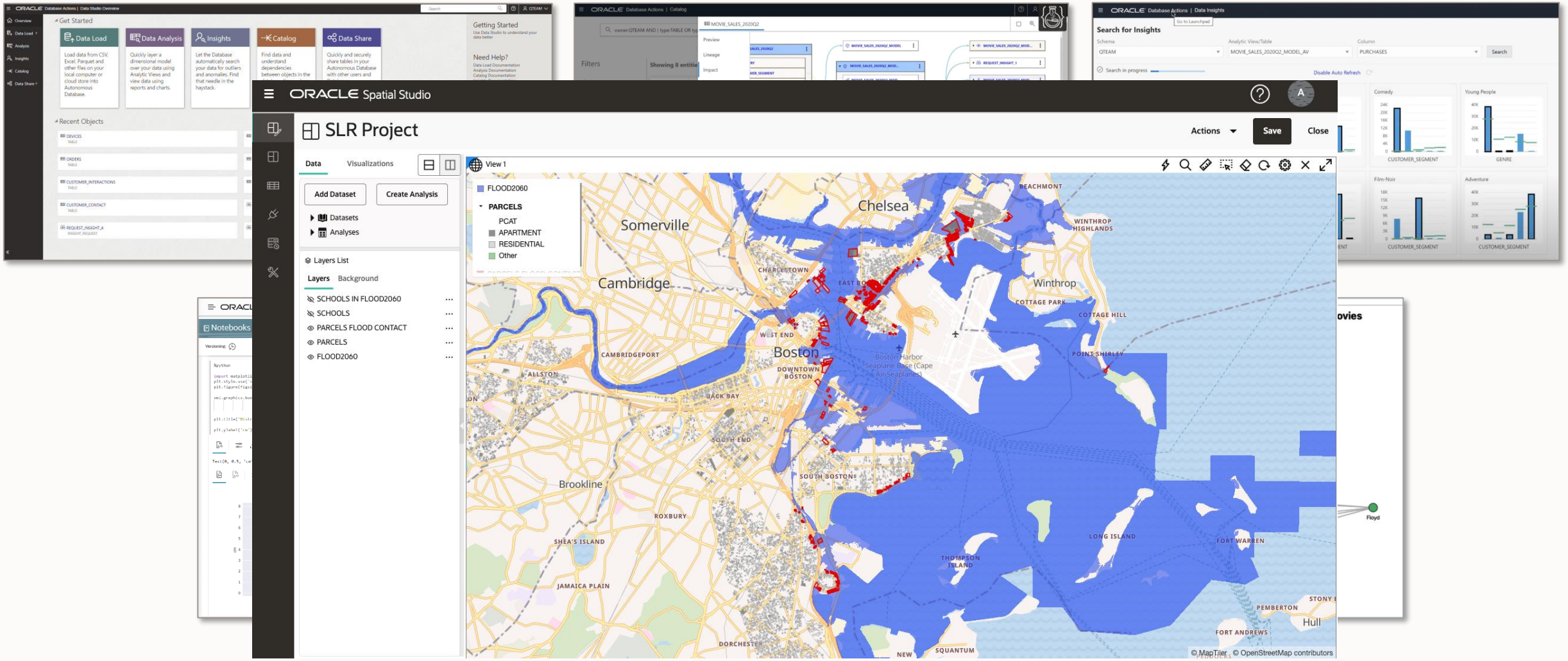
Machine Learning Notebooks



Graph Studio

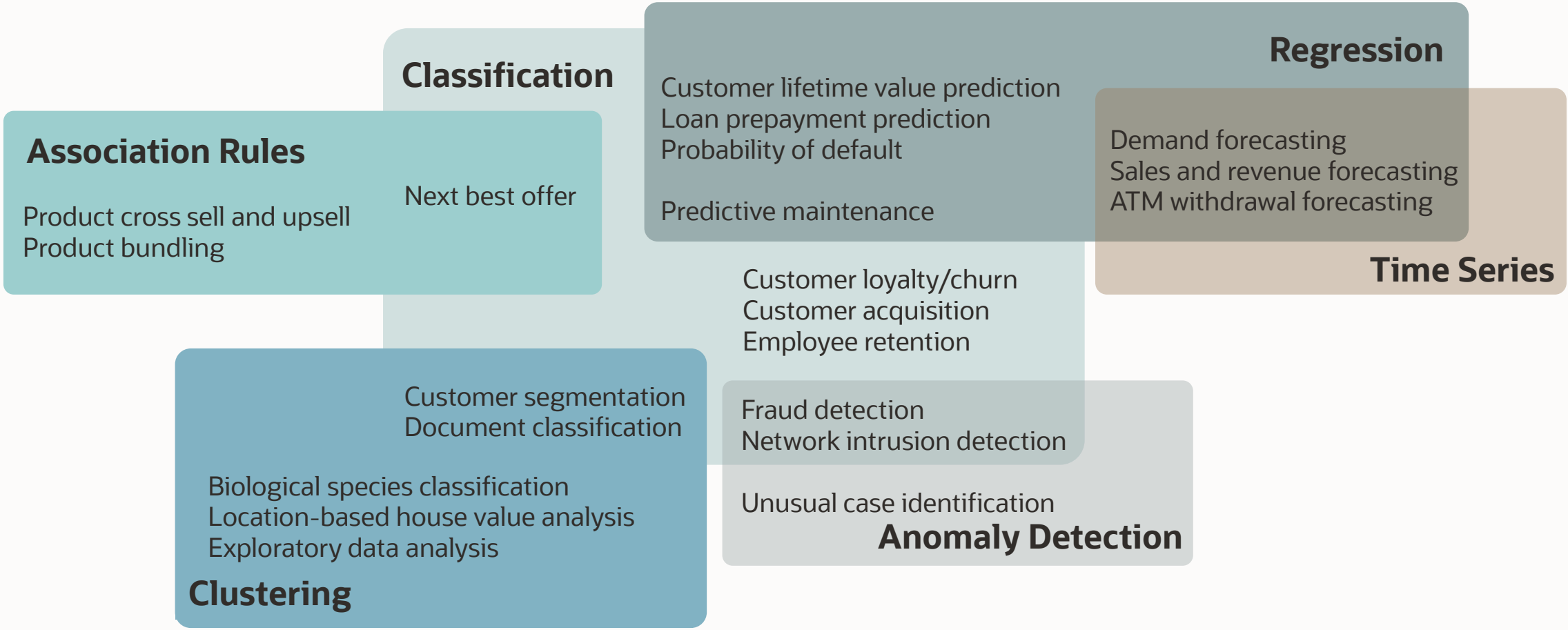


Spatial Studio



Use cases and machine learning techniques

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



Oracle Machine Learning In-Database Algorithms

Address a wide range of business problems

Classification

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

Clustering

Hierarchical K-Means
Hierarchical O-Cluster
Expectation Maximization

Row Importance

CUR Decomposition

Ranking

XGBoost

Regression

Generalized Linear Model (GLM)
Neural Network
Support Vector Machine (SVM)
Stepwise Linear regression
XGBoost

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 for row & AI

- OML Algorithm Cheat Sheet
- Algorithm Documentation
- OML Performance on ADB

Time Series

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

Anomaly Detection

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

Association Rules

A priori

Survival Analysis

XGBoost

Includes support for partitioned models,
integrated text mining,
automatic data preparation

Oracle Machine Learning In-Database Algorithms

Empower users with ML included in Oracle Autonomous Database

In-database, parallelized, distributed algorithms

ML models as first-class database objects

Faster time-to-market through immediate solution deployment

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

Example - 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_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;
```

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

Example - 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_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;
```

```
ore.sync(table="CUSTOMERS")
settings = list(model_name="BUY_TRAVEL_INSUR")
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_INSUR')
```

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

Example - 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_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;
```

```
ore.sync(table="CUSTOMERS")
settings = list(model_name="BUY_TRAVEL_INSUR")
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_INSUR')
```

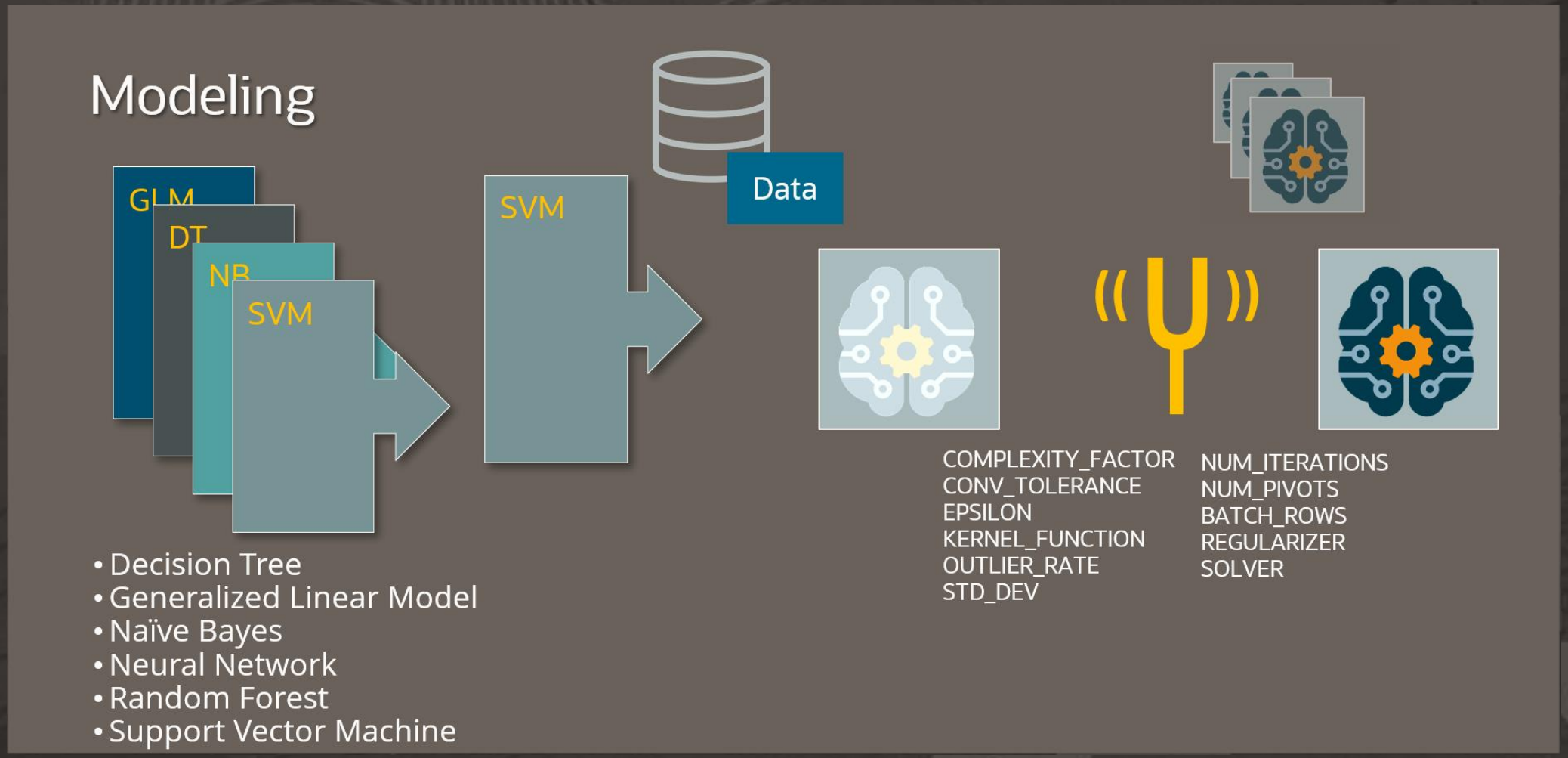
Apply a machine learning model to predict probability that individual customer is likely to buy

```
SELECT prediction_probability(BUY_TRAVEL_INSUR, '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_INSUR 1,'YES'USING3
1	0.9276956709910801

Machine Learning Modeling Process



Automated Machine Learning (AutoML)

Simplify the modeling process

Eliminate repetitive tasks of model building and evaluation

Increase user **productivity**

Enable **non-experts** to produce ML models

Apply ML to the ML process to reduce search space and compute

Build in-database models using Python and skip the details

OML4Py AutoML API simplifies the modeling process

Algorithm Selection

```
%python

as_wine_cl = automl.AlgorithmSelection(mining_function='classification',
                                       score_metric='accuracy', parallel=2)

wine_alg_ranking_cl = as_wine_cl.select(WINE_X_cl, WINE_y_cl, k=4)

print("Ranked algorithms:\n", wine_alg_ranking_cl)

selected_wine_alg_cl = next(iter(dict(wine_alg_ranking_cl).keys()))
print("Best algorithm: ", selected_wine_alg_cl)
```

```
Ranked algorithms:
 [('svm_gaussian', 0.9825515947467167), ('svm_linear', 0.9727954971857411), ('nn', 0.967081
2114714554), ('rf', 0.9495470383275262)]
Best algorithm:  svm_gaussian
```

Build in-database models using Python and skip the details

OML4Py AutoML API simplifies the modeling process

Algorithm Selection

```
%python

as_wine_cl = automl.AlgorithmSelection(mining_function='classification',
                                       score_metric='accuracy', parallel=2)

wine_alg_ranking_cl = as_wine_cl.select(WINE_X_cl, WINE_y_cl, k=4)

print("Ranked algorithms:\n", w

selected_wine_alg_cl = next(ite
print("Best algorithm: ", selec
```

```
Ranked algorithms:
 [('svm_gaussian', 0.98255159474
2114714554), ('rf', 0.949547038
Best algorithm:  svm_gaussian
```

Feature Selection

```
%python

fs_wine_cl = automl.FeatureSelection(mining_function = 'classification',
                                     score_metric = 'accuracy', parallel=2)

selected_wine_features_cl = fs_wine_cl.reduce(selected_wine_alg_cl,
                                              WINE_X_cl, WINE_y_cl)

WINE_X_reduced_cl = WINE_X_cl[:,selected_wine_features_cl]

print("Selected columns:", WINE_X_reduced_cl.columns)
print("Number of columns:")
"{}  reduced to {}".format(len(WINE_X_cl.columns), len(selected_wine_features_cl))
```

FINISHED

```
Selected columns: ['alcohol', 'ash', 'alcalinity_of_ash', 'flavanoids', 'nonflavanoid_pheno
ls', 'color_intensity', 'hue', 'od280/od315_of_diluted_wines', 'proline']
Number of columns:
'13  reduced to 9'
```

Build in-database models using Python and skip the details

OML4Py AutoML API simplifies the modeling process

Algorithm Selection

```
%python

as_wine_cl = automl.AlgorithmSelection(mining_function='classification',
                                       score_metric='accuracy', parallel=2)

wine_alg_ranking_cl = as_wine_cl.select(WINE_X_cl, WINE_y_cl, k=4)

print("Ranked algorithms:\n", w

selected_wine_alg_cl = next(ite
print("Best algorithm: ", selec
```

```
Ranked algorithms:
 [('svm_gaussian', 0.98255159474
2114714554), ('rf', 0.949547038
Best algorithm:  svm_gaussian
```

Feature Selection

```
%python

fs_wine_cl = automl.FeatureSelection(mining_function = 'classification',
                                     score_metric = 'accuracy', parallel=2)

selected_wine_features_cl = fs_wine_cl.reduce(selected_wine_alg_cl,
                                              WINE_X_cl, WINE_y_cl)

WINE_X_reduced_cl = WINE_X

print("Selected columns:",
print("Number of columns:"
"{}  reduced to {}".format
```

FINISHED

```
Selected columns: ['alcohol
ls', 'color_intensity', 'hu
Number of columns:
'13  reduced to 9'
```

Model Tuning

```
%python


mt_wine_cl = automl.ModelTuning(mining_function = 'classification', parallel=2)

results_cl = mt_wine_cl.tune(selected_wine_alg_cl, WINE_X_reduced_cl, WINE_y_cl)
tuned_model_cl = results_cl['best_model']
tuned_model_cl


Algorithm Name: Support Vector Machine

Mining Function: CLASSIFICATION
```

FINISHED

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

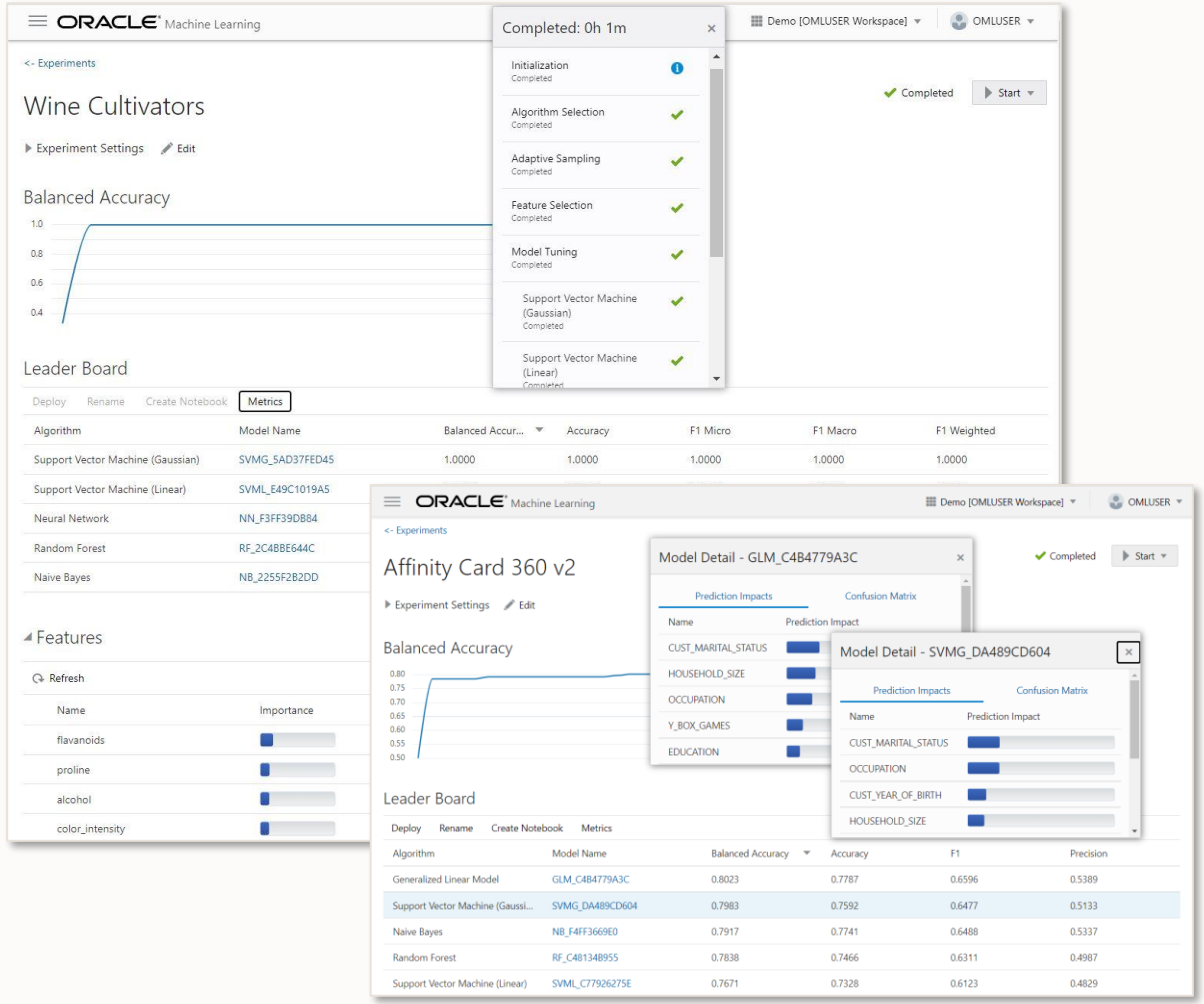
Start automated build and compare of multiple models with model quality metrics

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

Rename models to easily recognize models in model repository

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

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



Use ML models from REST endpoints with OML Services

MLOps with ease of application integration

Model Management		Deployment		Monitoring		Cognitive Text	
Store/ Organize	Version/ Compare	Real-time Scoring	Batch Scoring	Data Monitoring	Model Monitoring	Topics and keyword extraction	Text summary/ similarity

Lightweight scoring using REST endpoints

Real-time data scoring for streaming and other applications

Singleton, small batch, and full batch scoring

Supports classification, regression, clustering and feature extraction models

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

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

Simplify Python and R solution deployment

Data scientists and R/Python users develop solutions

Manage and invoke Python or R user-defined functions from the database environment

Use third-party packages to augment database functionality

Invoke from SQL and REST

No need to worry about starting, stopping, or managing Python or R engines explicitly

Invoke user-defined functions in a data-parallel, task-parallel, and non-parallel manner

Top 10 enterprise requirements for data science and machine learning platform

- Scalability
- Security
- Integration
- Collaboration
- Model Mngt and MLOps
- Explainability
- AutoML
- Governance
- Flexibility
- Cost-effectiveness

For more information…

Webpages

https://oracle.com/machine-learning

https://www.oracle.com/database/spatial

https://www.oracle.com/database/graph

OML Blog

https://bit.ly/omlblogs

OML GitHub Repository

https://bit.ly/omlgithub

OML Office Hours

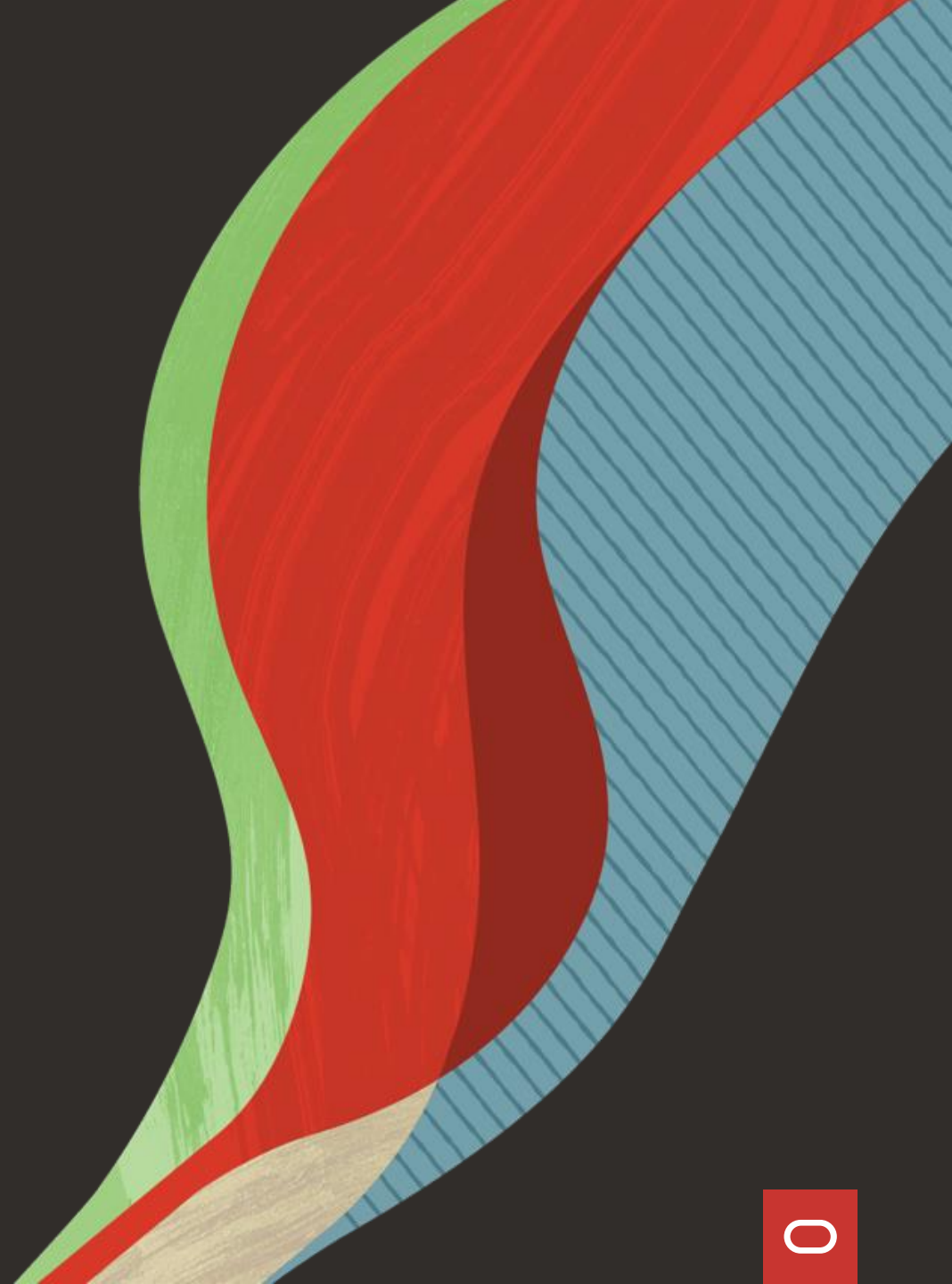
https://bit.ly/omlofficehours

Try on Oracle LiveLabs

Overview: https://bit.ly/omlfundamentalshol
OML4Py: https://bit.ly/oml4pyhol

OML Documentation

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



ORACLE
DatabaseWorld

Thank you

Mark Hornick

mark.hornick@oracle.com

@MarkHornick

MarkHornick

Group: Oracle Machine Learning