AI Summit 발표 자료

대규모 데이터의 처리를 위한 In-DB Machine Learning과 AutoML

M/L의 쉬운 활용을 통해 Citizen Data Scientist 지원

장성우 전무

Cloud Engineering, Oracle Korea

Dec 9, 2021

Overall Message

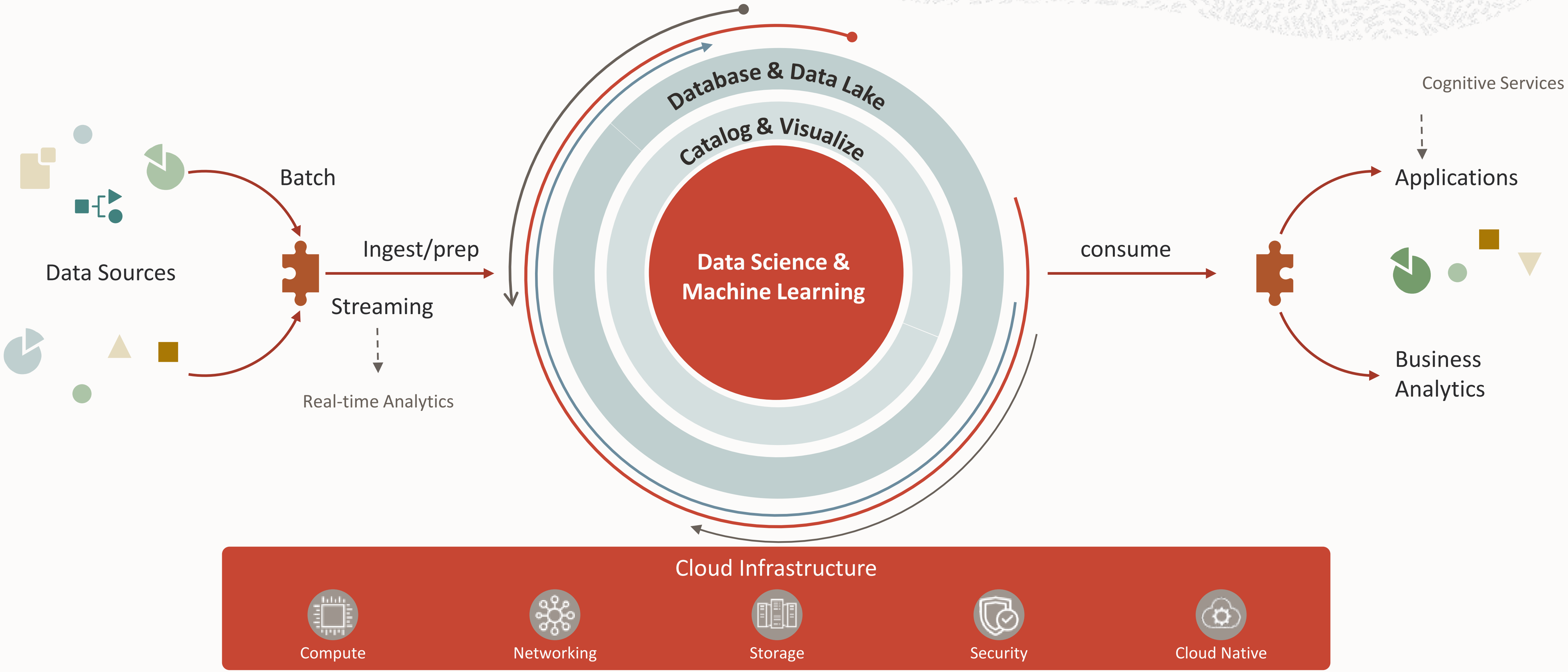
- Data Scientist가 접하는 어려움
  - ✓ 대규모 데이터의 빠른 처리를 통한 생산성 향상 필요
  - ✓ 수많은 데이터 관리 작업과 복잡한 M/L Process 수반
- In-DB Machine Learning
  - ✓ 대규모 데이터의 이동 없이 DB 안에서 M/L 기반 분석/예측 작업 가능
  - ✓ 알고리즘의 병렬 수행 지원을 통한 성능 향상
- AutoML
  - ✓ 예측력/성능이 좋은 M/L 알고리즘을 자동으로 선택
- Autonomous DB
  - ✓ Data Scientist는 모델에만 집중하고 나머지 DB 관리 작업은 ADB가 자율적으로 수행
- Oracle Machine Learning : In-DB M/L + AutoML + Autonomous DB
  - ✓ M/L의 쉬운 활용을 통해 Citizen Data Scientist 지원

Agenda

- Machine Learning Overview
- In-Database Machine Learning
  - Supporting Algorithms
  - Performance
  - Interface Method for Python, R and SQL
- AutoML
  - Methodology
  - Performance Improvement
- Autonomous DB
  - Architecture
  - Auto-Scaling
- Summaries : Oracle Machine Learning

Data Scientist의 Machine Learning Process

End-2-End M/L 개발 및 배포/활용 과정에서 효과적인 대규모 데이터 관리 방안이 중요해짐



Factors Affecting Machine Learning Performance

Data volume – the most obvious factor is the amount of data involved

Data movement and loading

- related to data volume is the performance impact of moving data from one environment to another.
- this time needs to be considered when comparing machine learning tools and processes.

Algorithm implementation

- algorithms implemented in a non-parallel or single-threaded manner result in poor performance even when run on multi-processor hardware
- enabling parallelism is often fundamental for improving performance and scalability.

Choice of algorithm – different algorithms can have vastly different computational requirements and results

Concurrent users – # of data scientists working on the system

Load on the system – # of workloads on the system

데이터의 거대화에 따른 HPC 시장의 성장

- **데이터의 거대화는 불가피**
  - 과거 20년 동안 빅데이터 시장의 폭발적인 성장
  - 초거대 AI의 출현
  - 클라우드의 성장 및 일반화

- **HPC(High Performance Computing) 시장의 성장**
  - AI와 Big Data에 따른 HPC 시장의 성장 확대
  - Top500 SC의 성능 증대 → 엑사스케일로 발전 중
  - HPC 수요에 대응하기 위한 민관 투자 증가
  - "국가초고성능컴퓨팅 혁신 전략" 수립 중

"거대 규모 데이터의 빠른 처리를 위한 HPC 시장의 성장 및 확대가 당분간 지속될 것으로 예측됨"

□ (시 장) **고성능컴퓨터 시장은 '19년 278억 달러에서 '24년 429억 달러로 지속 확대 전망**(연평균 9.0% 성장, Hyperion Research, '20)

○ 데이터 집약적 컴퓨팅의 확산으로 **스토리지 시장 급성장** 중이며, **슈퍼컴퓨터급 서버의 비중이 지속 확대**('19년 41.8% → '24년 47.7%)

○ 또한, 산·학·연, 정부기관 각 **수요주체별 수요도 안정적으로 증가 중**

< 고성능컴퓨터 매출 전망('19~'24) >

□ (기 술) 세계 주요국은 차세대 초고성능컴퓨팅 기술로서 인간의 두뇌 처리 능력과 비견되는 **엑사스케일급 시스템**(1초당 10^{18}번 연산)* **기술개발 경쟁 중**

* 현재 가장 빠른 일반PC(200GF) 5백만대 이상의 성능

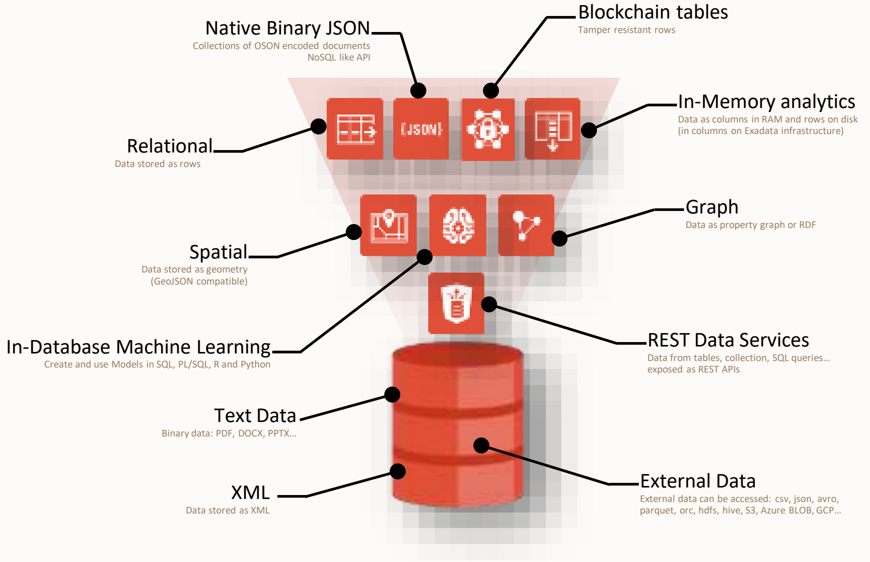
○ **중앙처리장치 성능은** 반도체 집적화('무어의 법칙') 한계로 **둔화 상태**

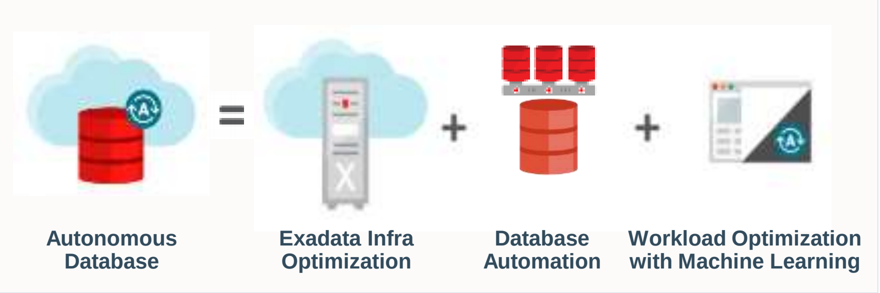
○ 이기종 멀티코어, 차세대 메모리 구조 등을 통해 **연산성능·집적도 100배↑ 및 전력효율 3배↑, 높은 확장성·유연성 구현**을 위한 기술 개발 중

출처 : "국가초고성능컴퓨팅 혁신전략, 2021.5"

대규모 데이터의 빠르고 안정적인 처리를 위한 Oracle Database 기술

- **단일 엔진 DBMS**
  - MVCC를 지원하는 DB kernel 기반 위에 지속적으로 새로운 기능을 추가
  - 지원 기능별 Version만 상이
- **Converged Database**
  - Relational, JSON, Spatial, Graph, Blockchain, XML, Text, LOB 등의 모든 타입의 데이터 모델 지원
- **병렬처리를 위한 RAC(Real Application Cluster)**
  - H/W 추가(scale-out)를 통한 선형적인 성능 향상 지원
- **Extreme Performance를 위한 Exadata DB Machine**
  - DB 서버와 Storage 서버를 초고속 네트워크로 연결
  - 27.6M OLTP Read IOPS 지원(X9M, 8K IO 기준)
  - Rack 연결을 통한 수십 PB DB 운영 지원
- **Autonomous DB**
  - DB 내에 M/L을 적용하여 auto-management 제공
  - Citizen Data Scientist의 기반 제공





"성능과 안정성의 기반 위에 Emerging Technology를 지속적으로 지원"

* MVCC : Multi-Version Concurrency Control

Machine Learning Performance 개선을 위한 3가지 방안

Data Movement and Loading Algorithm Implementation	In-Database Machine Learning
Choice of Algorithm	AutoML
Concurrent User Load on the system	Autonomous DB

Machine Learning 업무 적용의 현재

현업 담당자는 D/S 조직에 분석 모델을 요청하고, D/S 조직은 모델을 작성/제공하는 과정에서

긴 시간 소요 및 Communication Overhead 발생

현업 담당자가 직접 분석 모델을 만들고 적용하여 분석의 실시간성 및 업무의 효율성을 높일 필요가 있음

➔ Citizen Data Scientist의 필요 이유

Citizen Data Scientist 확산의 전제 조건

현업 → Learning Data Analytics and M/L Providing Automation Tools → Citizen Data Scientist

- ✓ 스스로 데이터 관리/분석
- ✓ 실시간 분석 기반의 통찰력을 통한 비즈니스 기여

- **Machine Learning Process의 간소화 필요**
  - 현업 담당자는 데이터 혹은 시스템 관리의 부담 없이
  - 업무에 대한 지식을 기반으로 분석 대상과 목표만 정의하면
  - 쉬운 UI 혹은 Low-Code 환경을 통해
  - M/L 알고리즘의 선택 및 모델 생성 작업을 시스템이 자동적으로 지원해 줄 수 있어야함

- **지원 방안 : In-DB M/L + AutoML + Autonomous DB**

In-DB Machine Learning

M/L Algorithms Operate on Data in Database

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

DB의 성능과 안정성 기반 위에

DB 내 대규모 데이터에 대한 M/L 알고리즘 적용을 지원하여

데이터 이동 최소화 및 병렬 처리를 통한 성능 향상 제공

Machine Learning Algorithms and Analytics in Oracle Database

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

Anomaly Detection
- One-Class SVM
- *MSET-SPRT**

Clustering
- Hierarchical K-Means
- Hierarchical O-Cluster
- Expectation Maximization (EM)

Time Series
- Forecasting - Exponential Smoothing
- Includes popular models e.g. Holt-Winters with trends, seasonality, irregular time series

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

Attribute Importance
- Minimum Description Length
- Principal Component Analysis (PCA)
- Unsupervised Pairwise KL Divergence
- CUR decomposition for row & AI

Association Rules
- A priori

Predictive Queries
- Predict, cluster, detect, features

SQL Analytics
- SQL Windows
- SQL Patterns
- SQL Aggregates

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

Row Importance
- CUR Decomposition

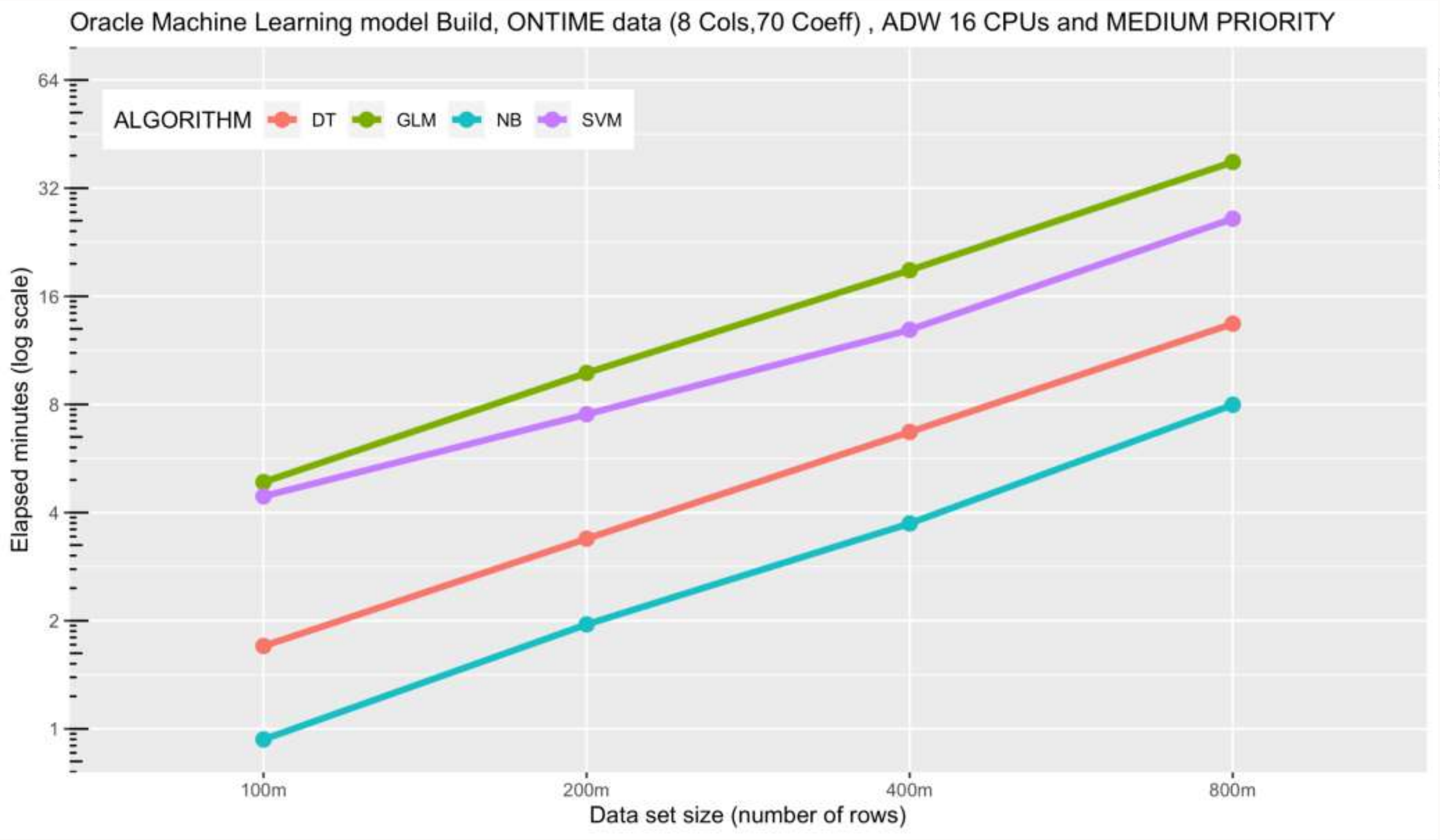
Ranking
- *XGBoost**

Text Mining Support
- Algorithms support text columns
- Tokenization and theme extraction
- Explicit Semantic Analysis (ESA)

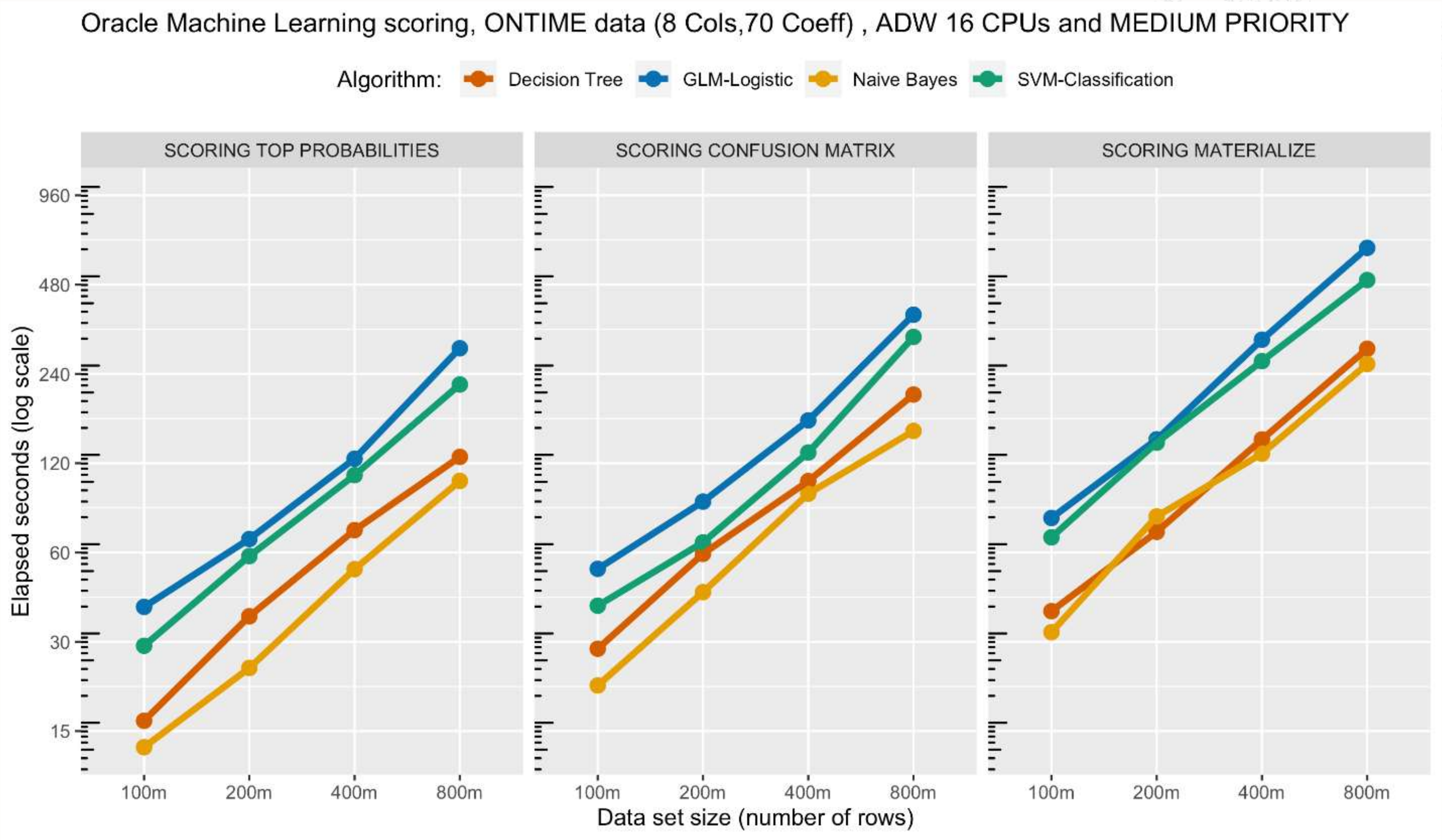
Statistical Functions
- min, max, median, stdev, t-test, F-test, Pearson's, Chi-Sq, ANOVA, etc.

Includes support for Partitioned Models, Transactional data and aggregations

** New in Oracle DB 21c*

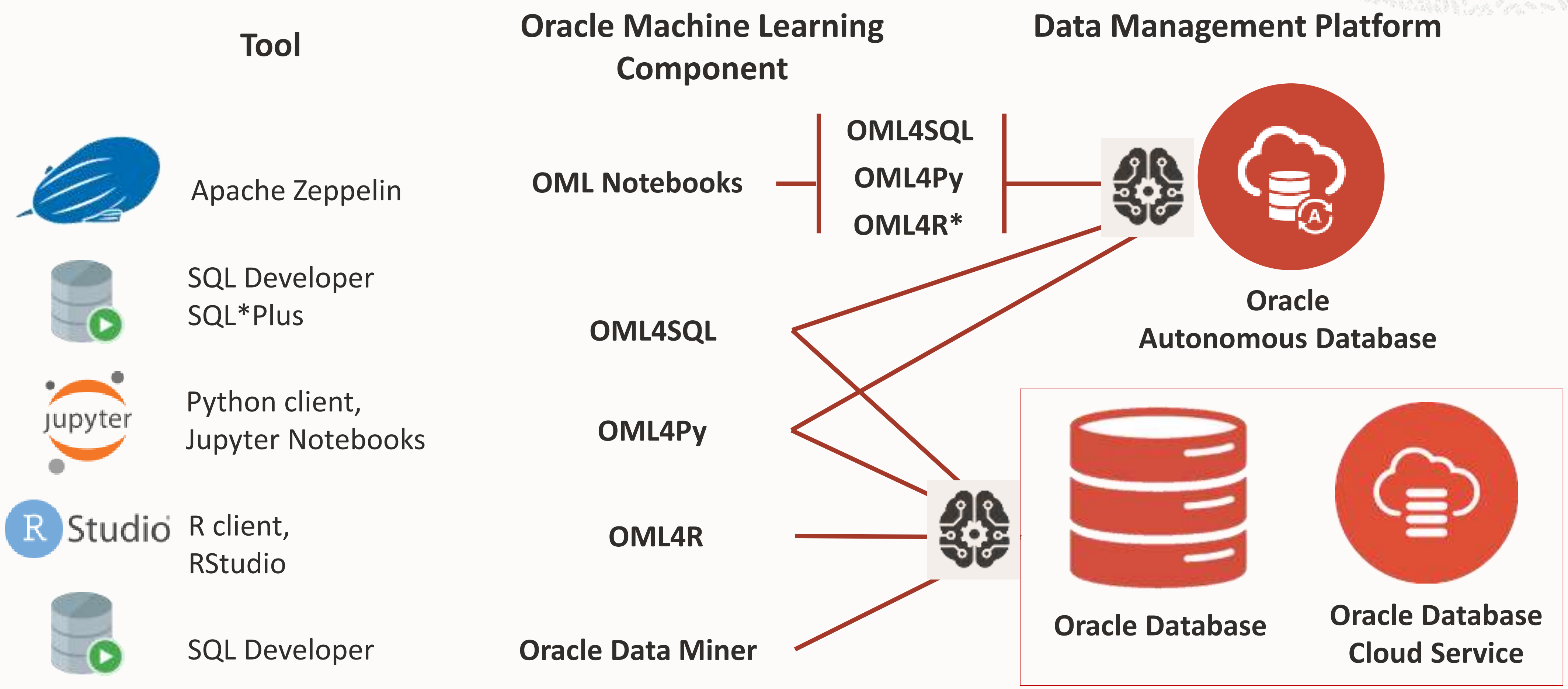


https://blogs.oracle.com/machinelearning/machine-learning-performance-on-autonomous-database



https://blogs.oracle.com/machinelearning/machine-learning-scoring-performance-on-autonomous-database

Oracle Machine Learning interfaces to Oracle Database



Oracle Machine Learning Notebooks

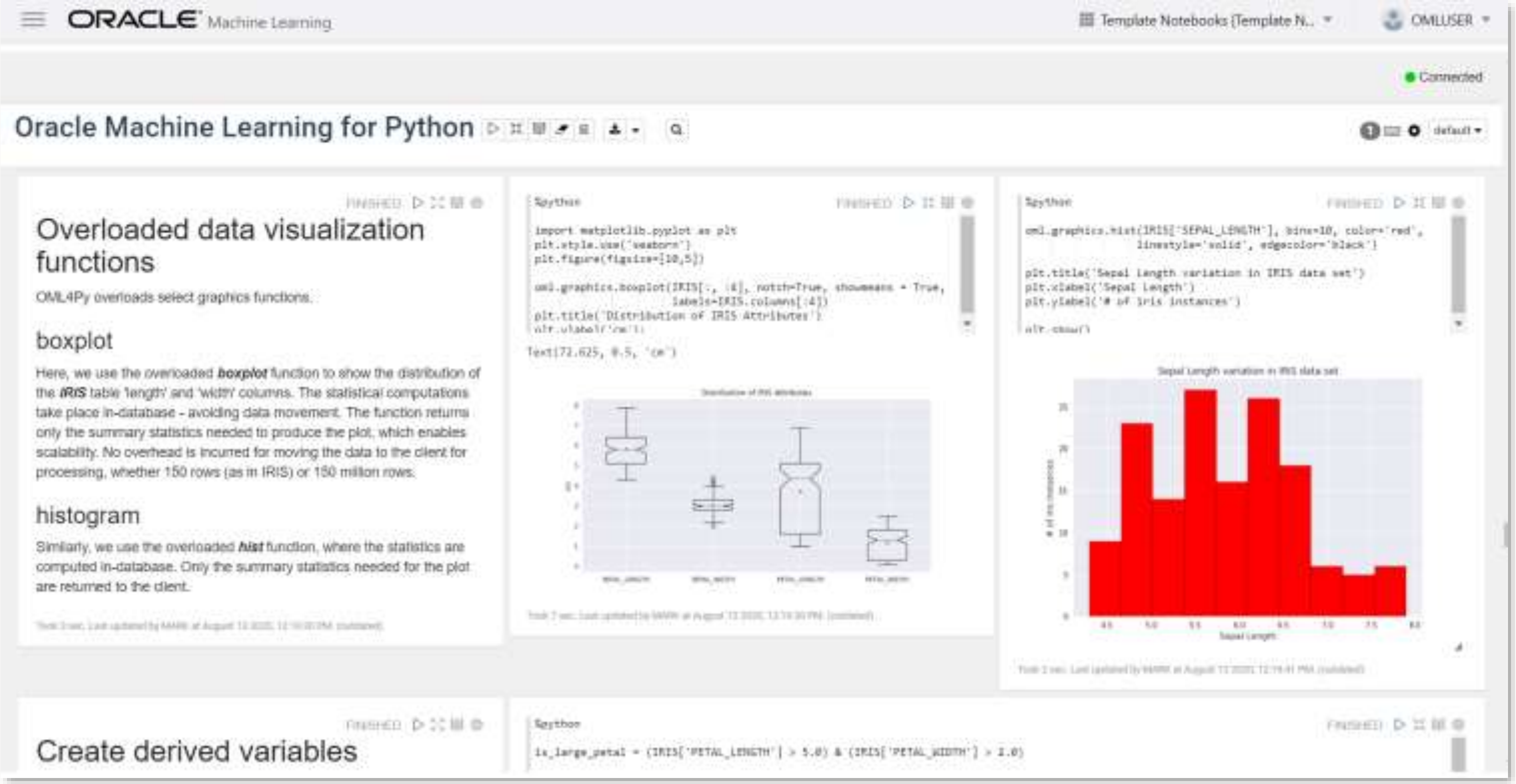
Autonomous Database as a Data Science Platform

Collaborative UI

- Based on Apache Zeppelin
- Supports data scientists, data analysts, application developers, and DBAs with SQL and Python
- Easy notebook sharing
- Scheduling, versioning, access control

Included with Autonomous Database

- Automatically provisioned and managed
- In-database algorithms and analytics functions
- Explore and prepare, build and evaluate models, score data, deploy solutions



Oracle Machine Learning for R and Python

Empower data scientists with open source environments

Transparency layer

- Leverage proxy objects so data remains in database
- Overload native functions translating functionality to SQL
- Use familiar R / Python syntax on database data

Parallel, distributed algorithms

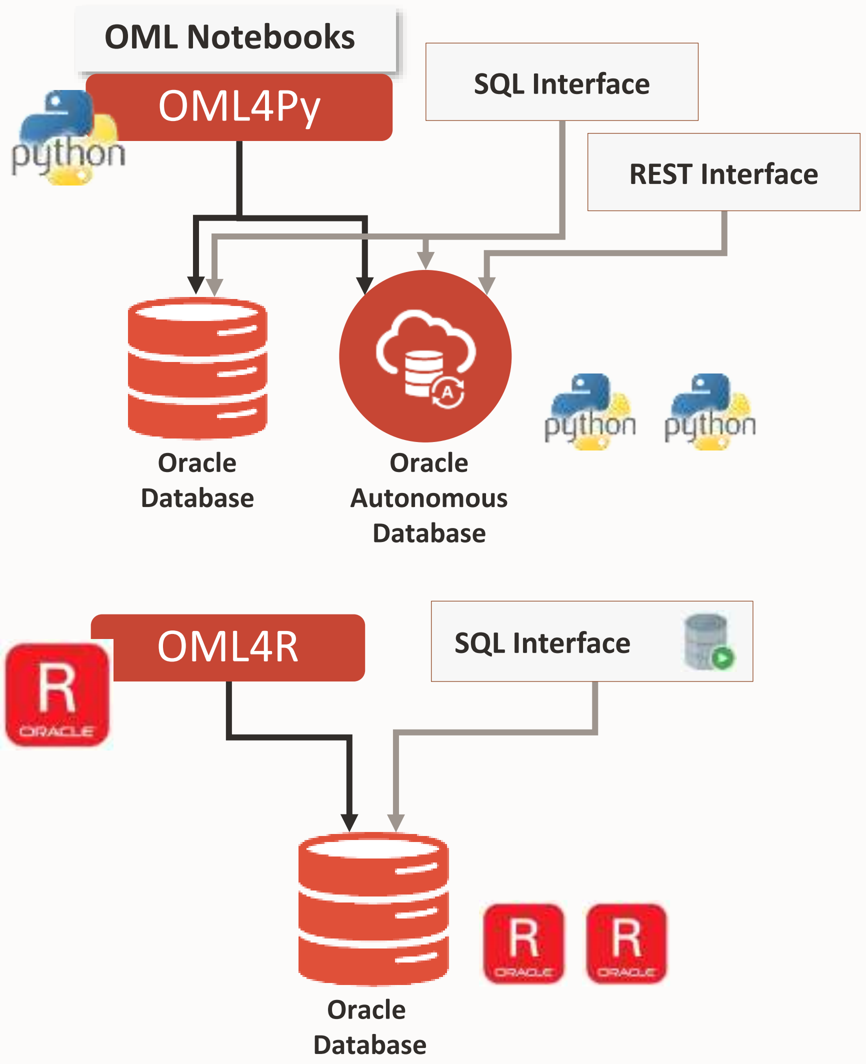
- Scalability and performance
- Exposes in-database algorithms available from OML4SQL

Embedded execution

- Manage and invoke R or Python scripts in Oracle Database
- Data-parallel, task-parallel, and non-parallel execution
- Use open source packages to augment functionality

OML4Py also includes AutoML and MLX

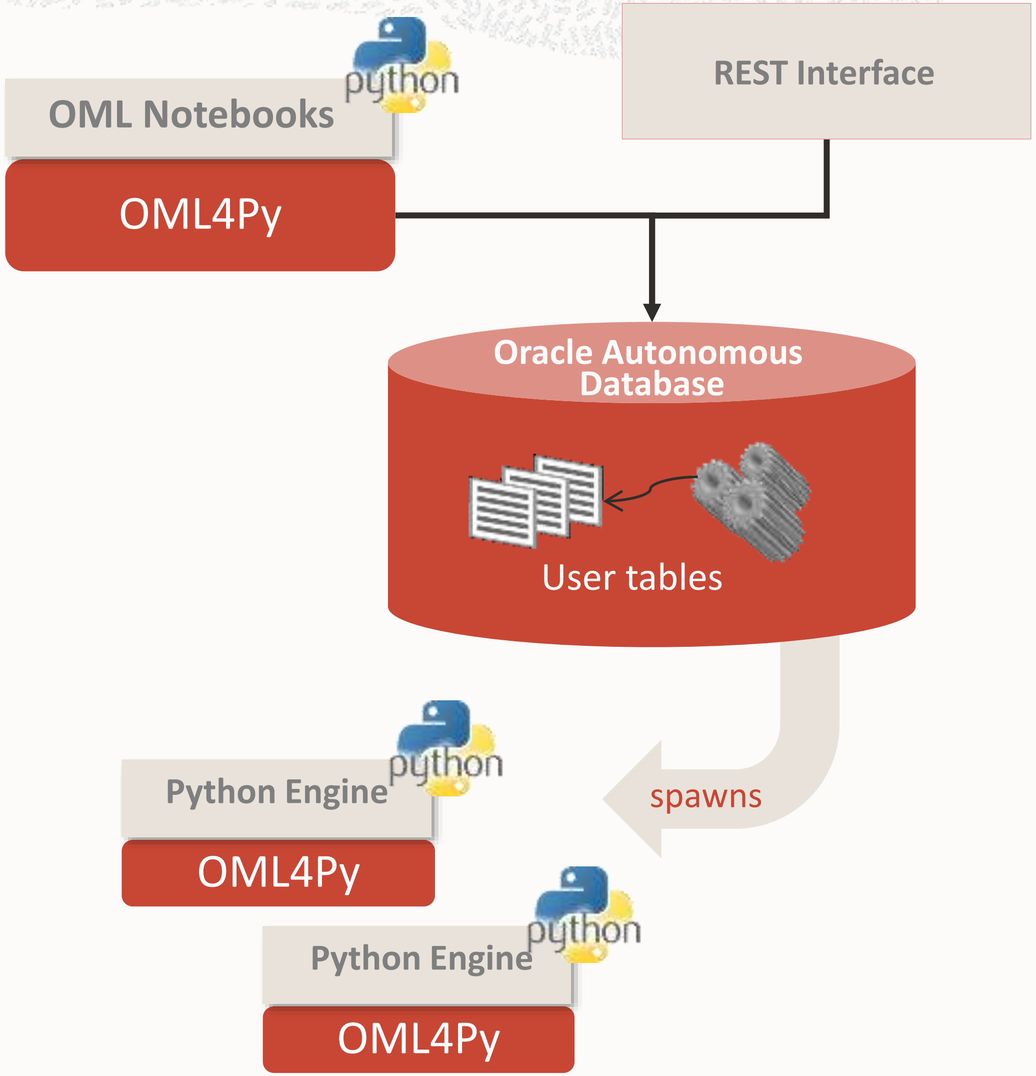
- Automated algorithm selection, feature selection, model tuning
- Algorithm-agnostic model explainability (MLX) for feature ranking



Embedded Execution

Example of parallel partitioned data flow using third party package using OML4Py

```
# user-defined function using sklearn
def build_lm(dat):
  from sklearn import linear_model
  lm = linear_model.LinearRegression()
  X = dat[['PETAL_WIDTH']]
  y = dat[['PETAL_LENGTH']]
  lm.fit(X, y)
  return lm
# select column(s) for partitioning data
index = oml.DataFrame(IRIS['SPECIES'])
# invoke function in parallel on IRIS table
mods = oml.group_apply(IRIS, index,
                       func=build_lm,
                       parallel=2)
mods.pull().items()
```



Oracle Machine Learning for SQL

Empower SQL users with immediate access to ML included with Oracle Database and Oracle Autonomous Database

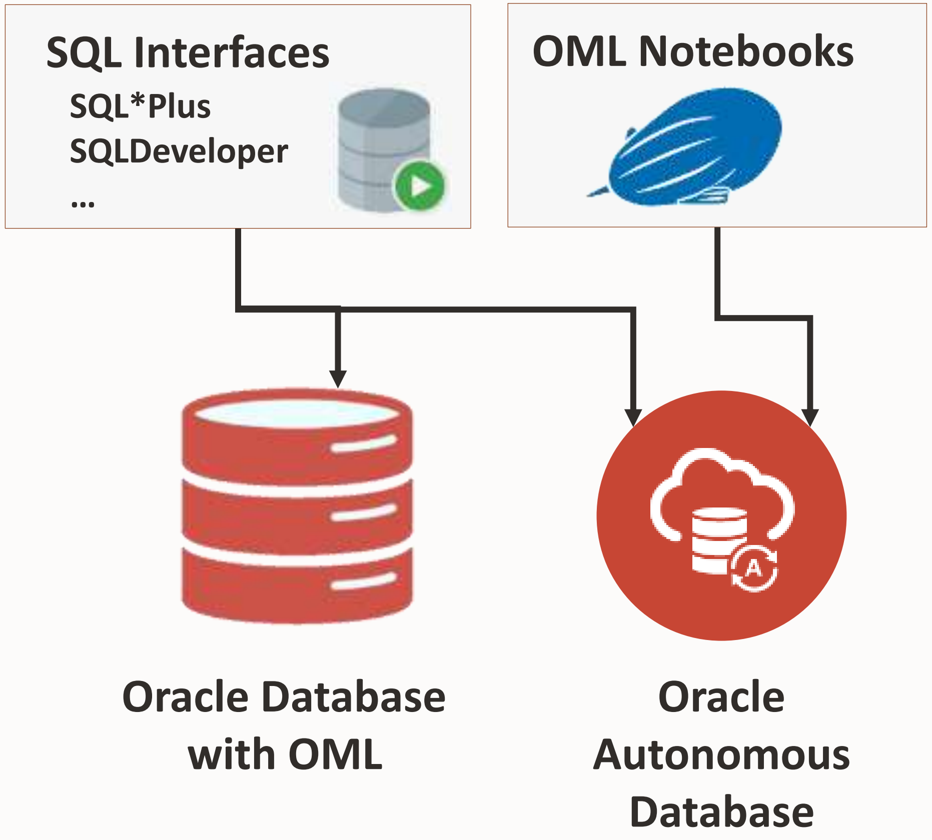
In-database, parallelized, distributed algorithms

- No extracting data to separate ML engine
- Fast and scalable
- Batch and real-time scoring at scale that leverages Exadata storage-tier function pushdown
- Algorithm-specific automatic data preparation
- Explanatory prediction details

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



Intuitive SQL API—OML4SQL

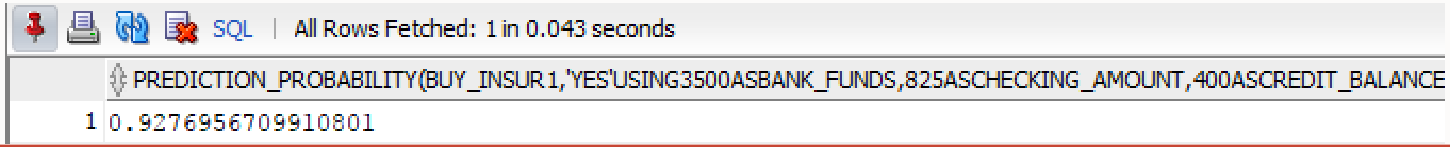
OML to Predict Customer Behavior

-- 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_TRVL_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 which customers are likely to buy

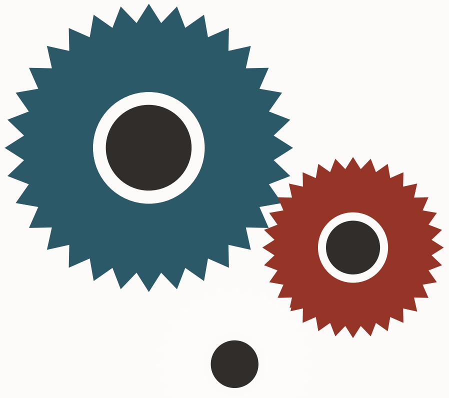
```
SELECT prediction_probability(BUY_TRVL_INSUR, 'Yes'
         USING 3500 as bank_funds, 37 as age, 'Married' as marital_status, 2 as num_previous_cruises)
FROM dual;
```



Automation for Machine Learning Modeling

Pain points with traditional ML

- Rapid model development cycles
- Machine learning process can be complex and iterative
- Advanced knowledge of ML algorithms is normally required to get high quality models
- Developing machine learning models is often time intensive to manually explore space of algorithms and hyperparameters



Solution

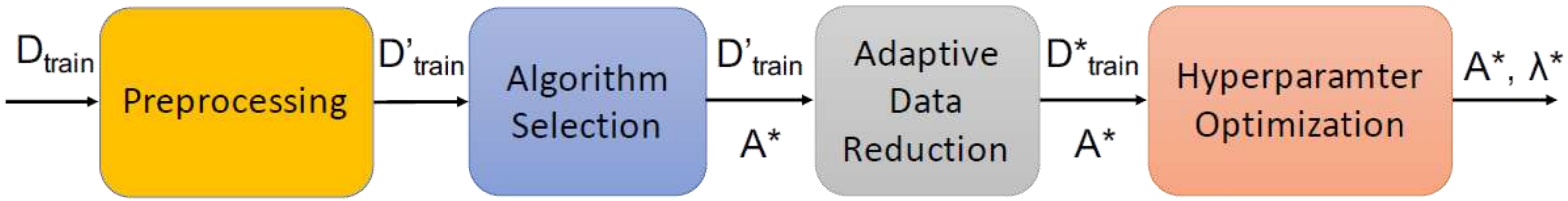
- Introduce an efficient ML pipeline to address rapid model development cycles
- Eliminate repetitive modeling tasks to increase user productivity

to reduce compute time

AutoML

A novel, iteration-free machine learning modeling pipeline

- Designed to provide accurate models in a shorter runtime
- Achieved by eliminating the need to repeatedly iterate over various pipeline configurations
- Each pipeline stage makes decisions based on meta-learned proxy models
- Proxy models predict candidate pipeline configuration performance before building the full final model
- Builds and tunes only the best candidate pipeline
- Achieves better results in a fraction of the time compared to state-of-the-art open source AutoML tools



Pipeline stages

Oracle AutoML methodology supports in-database and open source algorithms

Oracle AutoML Methodology

Oracle Machine Learning
in-database algorithms via OML4Py

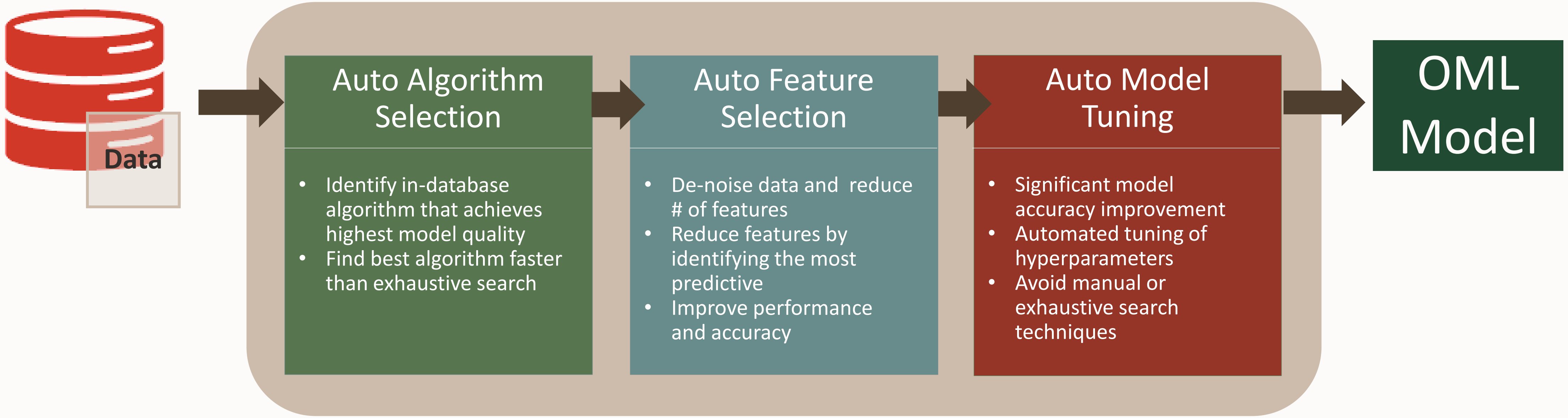
Proxy Models for in-database algorithms

OCI Data Science
open source Python algorithms

Proxy Models for open source algorithms

AutoML – *new* with OML4Py

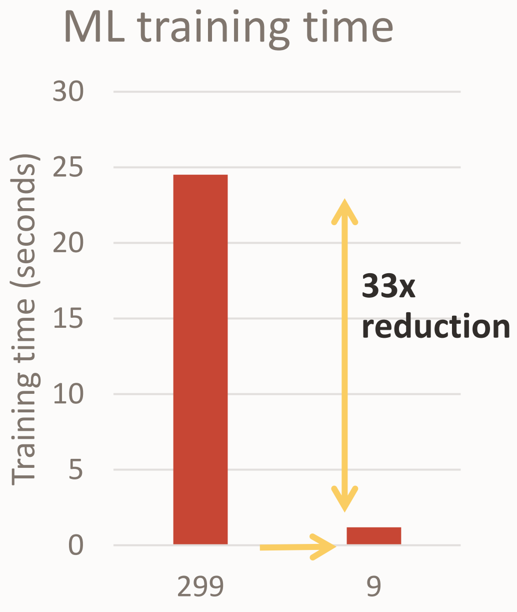
Increase data scientist productivity – reduce overall compute time

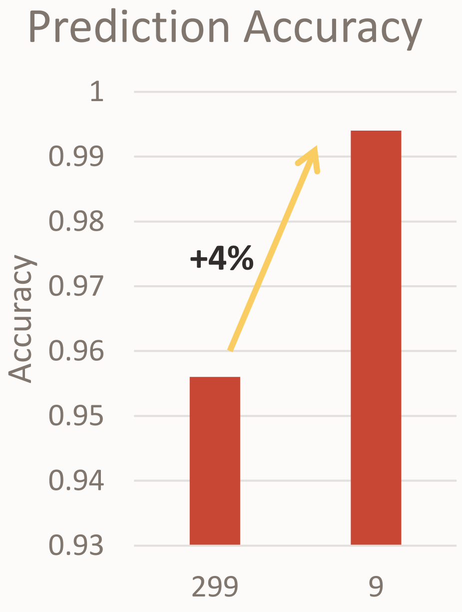


Enables non-expert users to leverage Machine Learning

Improve performance and accuracy with AutoML

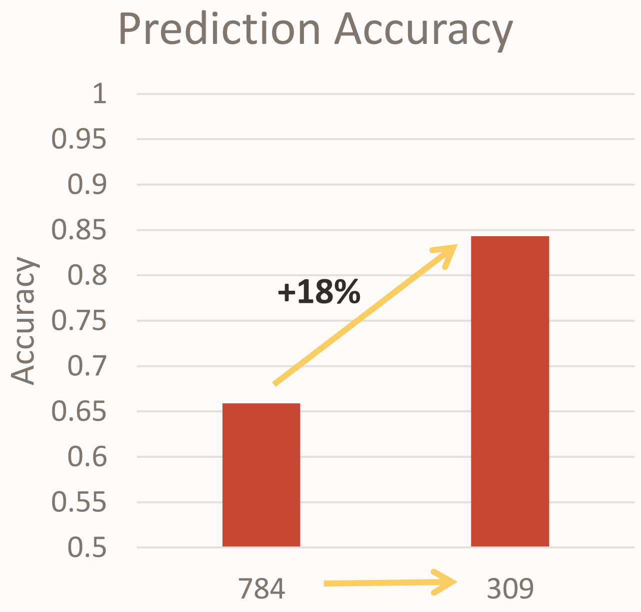
Reduce # features by identifying most relevant





97% reduction

OpenML dataset 312 with 1925 rows, 299 columns



60% reduction

1.3X time reduction to build SVM Gaussian model

OpenML dataset 40996 with 56K rows, 784 columns

Model Evaluation

Algorithms automatically compute standard metrics

```
%python

rf_mod = oml.rf(tree_term_max_depth = '4')
rf_mod = rf_mod.fit(IRIS_train_X, IRIS_train_y)

rf_mod
```

```
Algorithm Name: Random Forest

Mining Function: CLASSIFICATION

Target: Species

Settings:
                     setting name            setting value
0                       ALGO_NAME        ALGO_RANDOM_FOREST
1               CLAS_MAX_SUP_BINS                        32
2           CLAS_WEIGHTS_BALANCED                       OFF
3                    ODMS_DETAILS               ODMS_ENABLE
4    ODMS_MISSING_VALUE_TREATMENT   ODMS_MISSING_VALUE_AUTO
5                ODMS_RANDOM_SEED                         0
6                   ODMS_SAMPLING     ODMS_SAMPLING_DISABLE
7                       PREP_AUTO                        ON
8                  RFOR_NUM_TREES                        20
9             RFOR_SAMPLING_RATIO                        .5
10           TREE_IMPURITY_METRIC        TREE_IMPURITY_GINI
11             TREE_TERM_MAX_DEPTH                         4
12           TREE_TERM_MINPCT_NODE                       .05
13          TREE_TERM_MINPCT_SPLIT                        .1
14           TREE_TERM_MINREC_NODE                        10
15          TREE_TERM_MINREC_SPLIT                        20

Computed Settings:
  setting name setting value
0    RFOR_MTRY             2

Global Statistics:
   AVG_DEPTH  AVG_NODECOUNT  MAX_DEPTH  MAX_NODECOUNT  MIN_DEPTH  MIN_NODECOUNT  NUM_ROWS
0       3.25            5.5        4.0            6.0        4.0            4.0     104.0
```

View GLM model summary statistics

FINISHED

```
%sql

SELECT NAME, NUMERIC_VALUE, STRING_VALUE
  FROM DM$VGGLMR_SH_REGR_SAMPLE
 ORDER BY NAME;
```

settings

NAME	NUMERIC_VALUE	STRING_VALUE
ADJUSTED_R_SQUARE	0.62276742240863769	
AIC	813.68188071606562	
COEFF_VAR	28.62024439675282	
CONVERGED		YES
CORRECTED_TOTAL_DF	2713	
CORRECTED_TOT_SS	9617.5416359616866	
DEPENDENT_MEAN	4.0405305821665438	
ERROR_DF	2689	
ERROR_MEAN_SQUARE	1.3372834579528474	
ERROR_SUM_SQUARES	3595.9552184352069	
F_VALUE	187.61873750213547	
GMSEP	1.3497206234953807	
HOCKING_SP	0.00049750128644079146	
J_P	1.3496018391056923	
MODEL_DF	24	
MODEL_F_P_VALUE	0	
MODEL_MEAN_SQUARE	250.89943406360331	
MODEL_SUM_SQUARES	6021.5864175264796	
NUM_PARAMS	25	
NUM_ROWS	2714	
ROOT_MEAN_SQ	1.1564097275416043	

Model Explanation
Prediction Details

Identify the features or predictors that most influence a given prediction

Important where model transparency necessary to justify action

Useful during model evaluation, as well as for end-users to understand

Influential predictors with actual value and weight

CUST_ID	PRED_YRS_RES	LOWER_BOUND	UPPER_BOUND	FIRST_ATTRIBUTE	SECOND_ATTRIBUTE	THIRD_ATTRIBUTE	FOURTH_ATTRIBU...
100002	4.219	4.1	4.4	"Y_BOX_GAMES" actualValue="0" weight=".031"	"CUST_YEAR_OF_BIRTH" actualValue="1962" weight=".013"	"AFFINITY_CARD" actualValue="0" weight="-.956"	
100003	4.392	4.3	4.5	"Y_BOX_GAMES" actualValue="0" weight=".026"	"CUST_YEAR_OF_BIRTH" actualValue="1969" weight=".007"	"AFFINITY_CARD" actualValue="0" weight="-.967"	
100005	5.273	5.2	5.4	"Y_BOX_GAMES" actualValue="0" weight="-.047"	"CUST_YEAR_OF_BIRTH" actualValue="1957" weight="-.151"	"AFFINITY_CARD" actualValue="1" weight="-.157"	"OCCUPATION" actualValue="Crafts" weight="-.644"
100009	2.22	2	2.4	"AFFINITY_CARD"	"CUST_YEAR_OF_BIR	"Y_BOX_GAMES"	

OML AutoML UI

Enhance data scientist productivity and enable non-expert data professionals

Accelerate new ML projects

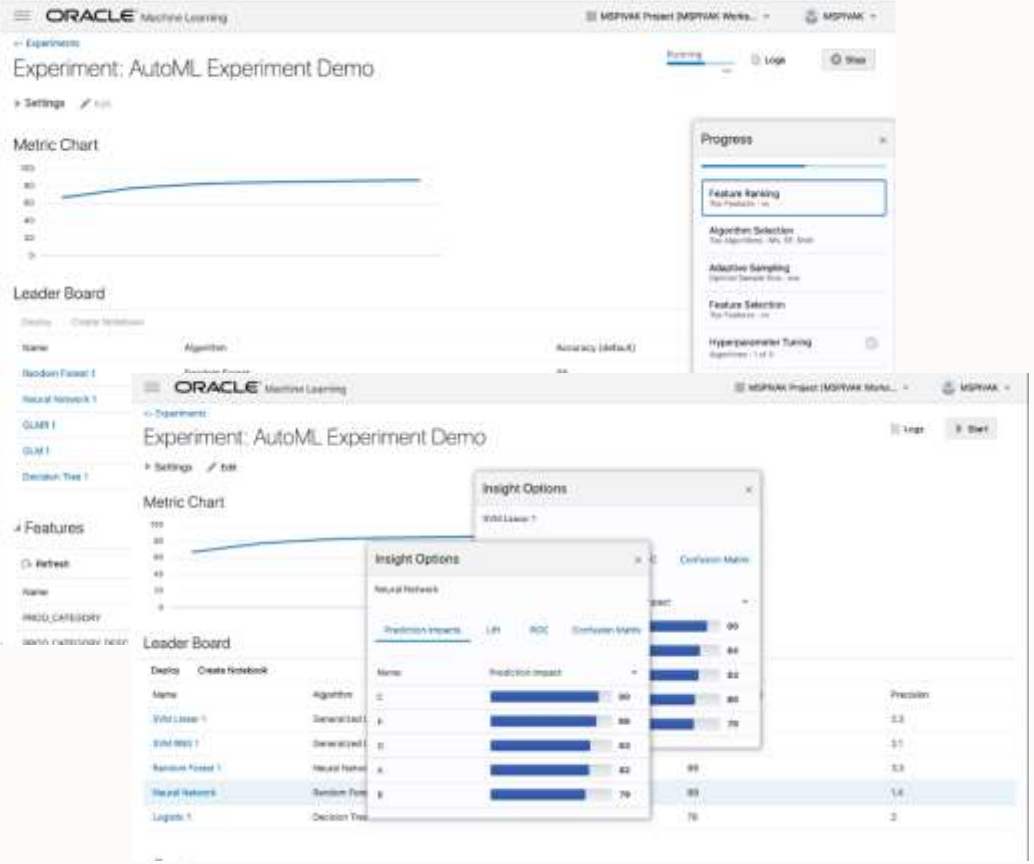
Automate repetitive and time-consuming tasks

Generate editable notebooks for selected models

Deploy models as REST endpoints

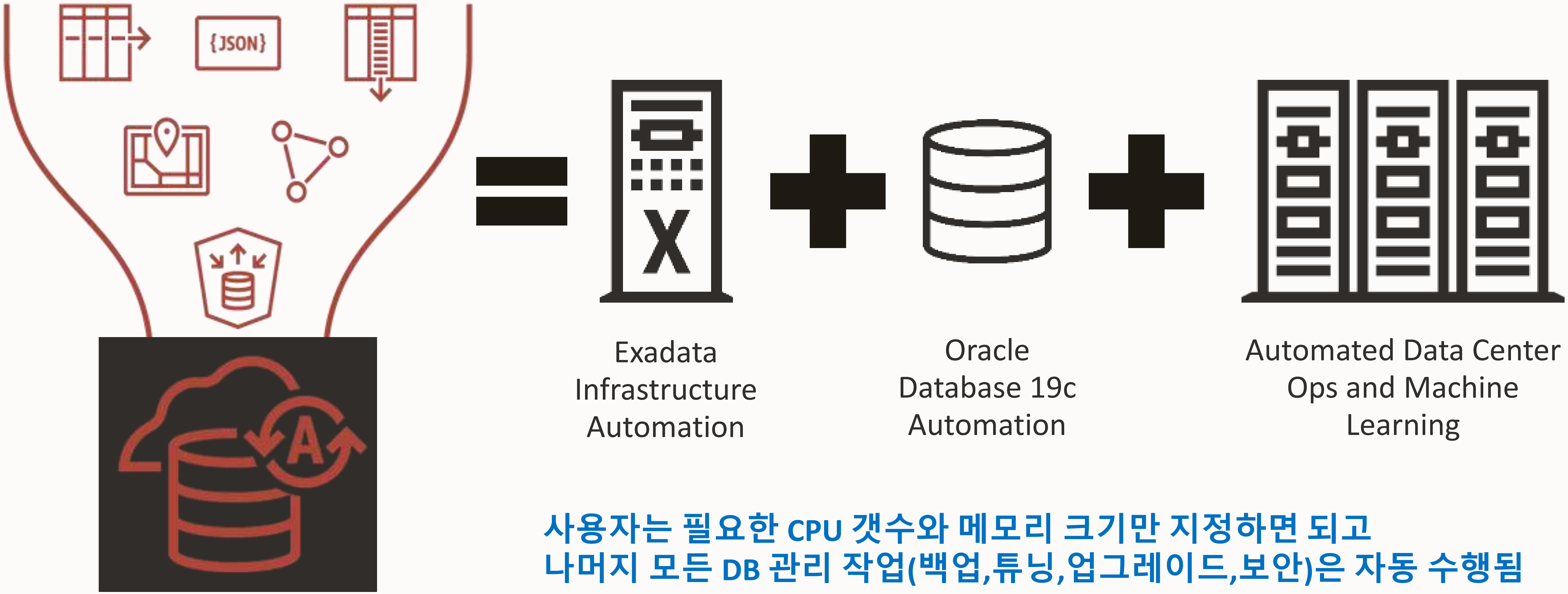
Featuring

- Monitor experiment progress
- Customize selection quality metric and metrics display
- Even faster data scoring performance for streaming and real-time applications



Oracle Autonomous Database is a Fully Managed Cloud Service

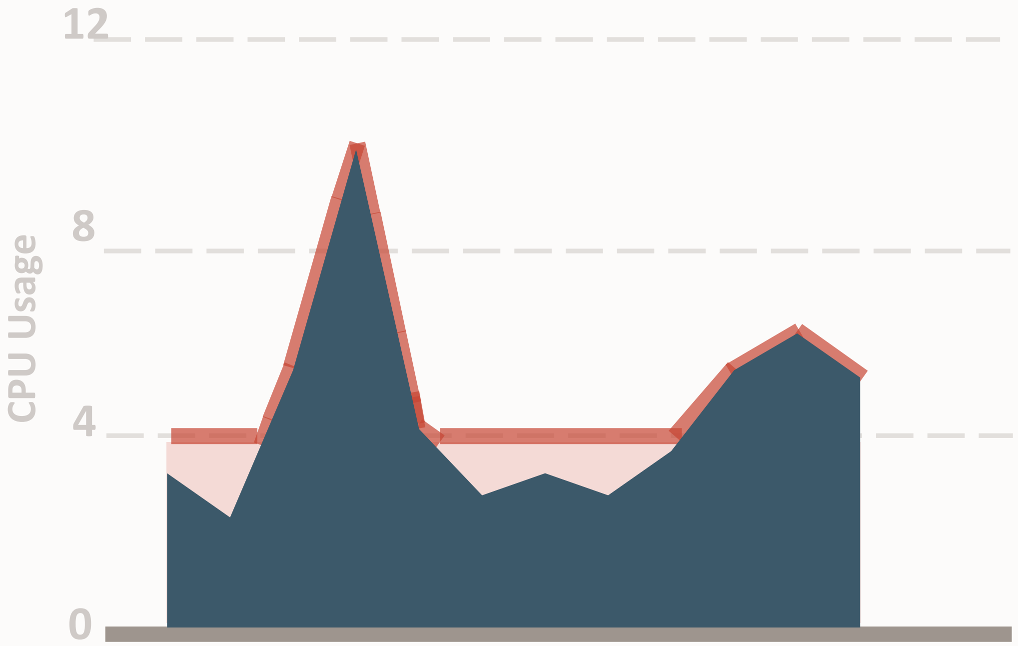
With automation across all components driven by AI and ML



사용자는 필요한 CPU 갯수와 메모리 크기만 지정하면 되고
나머지 모든 DB 관리 작업(백업,튜닝,업그레이드,보안)은 자동 수행됨

Auto Scaling in Oracle Autonomous Database

Automatically scales CPU/IO resources up to 3x when needed by workload



Dynamic Auto-Scale

Automatically scale up to 3X

Scaling with zero downtime

All scaling occurs online, while the application continuously runs.

Up to 3x better throughput for CPU/IO intensive workloads

사용자가 지정한 CPU 갯수와 메모리 크기의 최대 3배까지 자동적으로 Scale-up 되어짐

현업 사용자도 쉽게 M/L을 수행할 수 있는 기술적 토대 지원

Citizen Data Scientist로의 초대

현업 → Learning Data Analytics and M/L Providing Automation Tools → Citizen Data Scientist

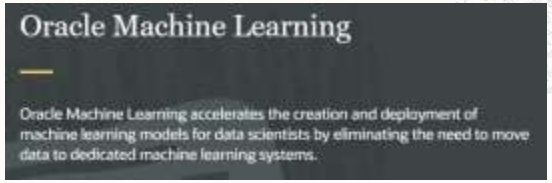
- ✓ 스스로 데이터 관리/분석
- ✓ 실시간 분석 기반의 통찰력을 통한 비즈니스 기여

- **필요한 작업 : "Just Do It"**
  - 데이터 분석과 Machine Learning의 기본 이해
  - Domain Knowledge를 기반으로 분석 대상과 목표를 정의
  - 편리한/익숙한 클라우드 벤더 선택 (데이터 관리의 부담이 최소화 되는 환경으로)
  - 업무 데이터 수집 및 적재
  - In-DB + AutoML의 **"Hands-on"** 수행
  - 분석 결과의 공유 및 확산

Summary

- Citizen Data Scientist의 필요성
  - ✓ 현업과 D/S 조직의 분리에 따른 Long Lead Time 및 Communication Overhead 최소화
  - ✓ 업에 대한 통찰력과 실시간 분석을 통한 비즈니스 기여 증대
- Machine Learning Process 간소화의 도구들
  - In-DB Machine Learning
    - ✓ 대규모 데이터의 이동 없이 DB 안에서 M/L 기반 분석/예측 작업 가능
    - ✓ 알고리즘의 병렬 수행 지원을 통한 성능 향상
  - AutoML
    - ✓ 예측력/성능이 좋은 M/L 알고리즘을 자동으로 선택
  - Autonomous DB
    - ✓ Data Scientist는 모델에만 집중하고 나머지 DB 관리 작업은 ADB가 자율적으로 수행
- Oracle Machine Learning : In-DB M/L + AutoML + Autonomous DB
  - ✓ M/L의 쉬운 활용 여건을 제공하여 Citizen Data Scientist 지원

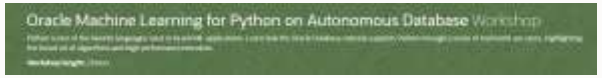
Helpful Links



ORACLE MACHINE LEARNING ON O.COM

https://www.oracle.com/machine-learning

OML TUTORIALS



OML LiveLab: https://apexapps.oracle.com/pls/apex/dbpm/r/livelabs/view-workshop?p180_id=560

OML4Py LiveLab: https://apexapps.oracle.com/pls/apex/dbpm/r/livelabs/view-workshop?wid=786

Interactive tour: https://docs.oracle.com/en/cloud/paas/autonomous-database/oml-tour

OML OFFICE HOURS

https://asktom.oracle.com/pls/apex/asktom.search?office=6801#sessionss

ORACLE ANALYTICS CLOUD

https://www.oracle.com/solutions/business-analytics/data-visualization/examples.html

OML4PY

- OML4Py (2m video)
- OML4Py Introduction (17m video)
- OML4Py Technical Brief
- OML4Py User's Guide
- Blog: Introducing OML4Py
- GitHub Repository with Python notebooks

ORACLE AUTOML UI

- Oracle Machine Learning AutoML UI (2m video)
- Oracle Machine Learning Demonstration (6m video)
- OML AutoML UI Technical Brief
- Blog: Introducing Oracle Machine Learning AutoML UI

OML SERVICES

- Oracle Machine Learning Services (2m video)
- OML Services Technical Brief
- Oracle Machine Learning Services Documentation
- Blog: Introducing Oracle Machine Learning Services
- GitHub Repository with OML Services examples

감사합니다

SUNGWOO.CHANG@ORACLE.COM