

ORACLE

Oracle Data Science Platform

Accelerate research

Peter Szegedi

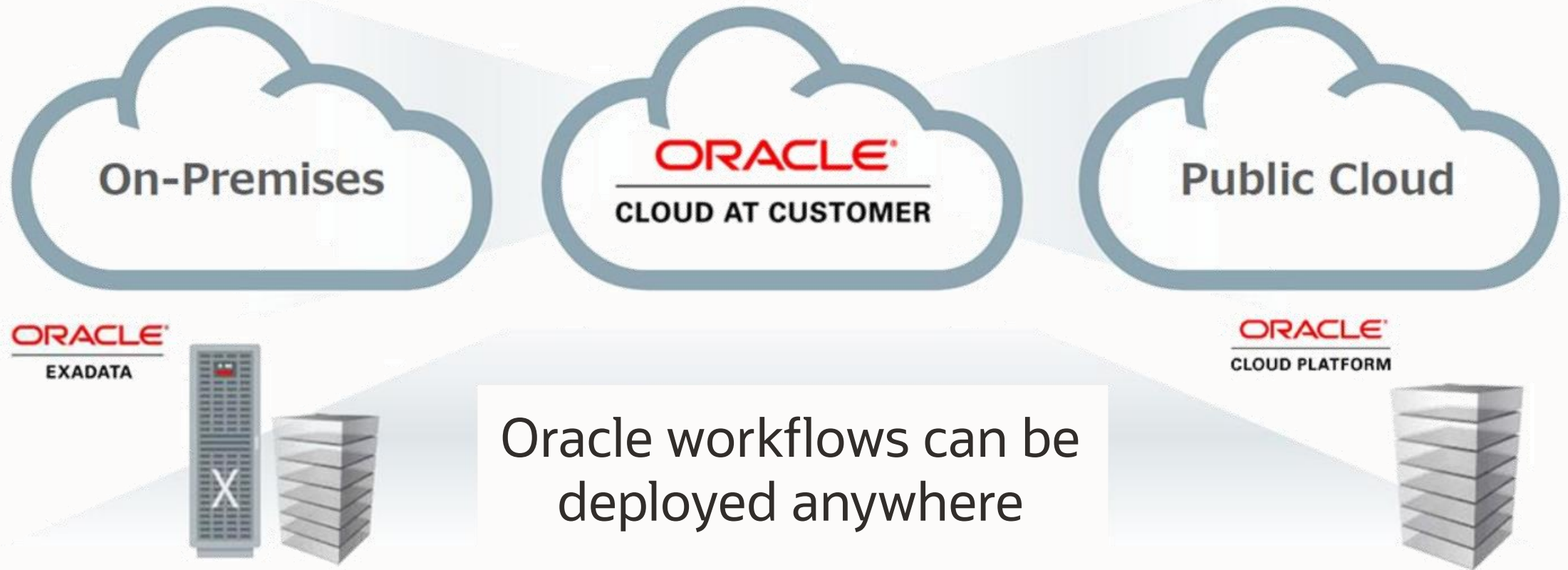
Oracle EMEA

Safe harbor statement

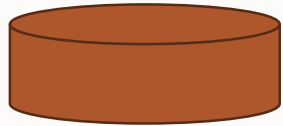
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

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.

On-premise, hybrid and cloud!

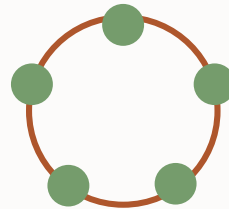


Challenges of Data Science



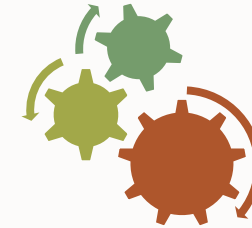
Access to Data

- On-prem/cloud/hybrid data management
- Finding data
- Combining data



Complex lifecycle to build models

- Multiple roles involved
- Reuse of past work
- Access to tools and infrastructure
- Access to flexible compute power



Operationalizing ML

- Packaging models for consumption
- Monitoring models
- Updating models

Oracle Cloud Infrastructure Data Science

Support for Python and open source

Accelerated

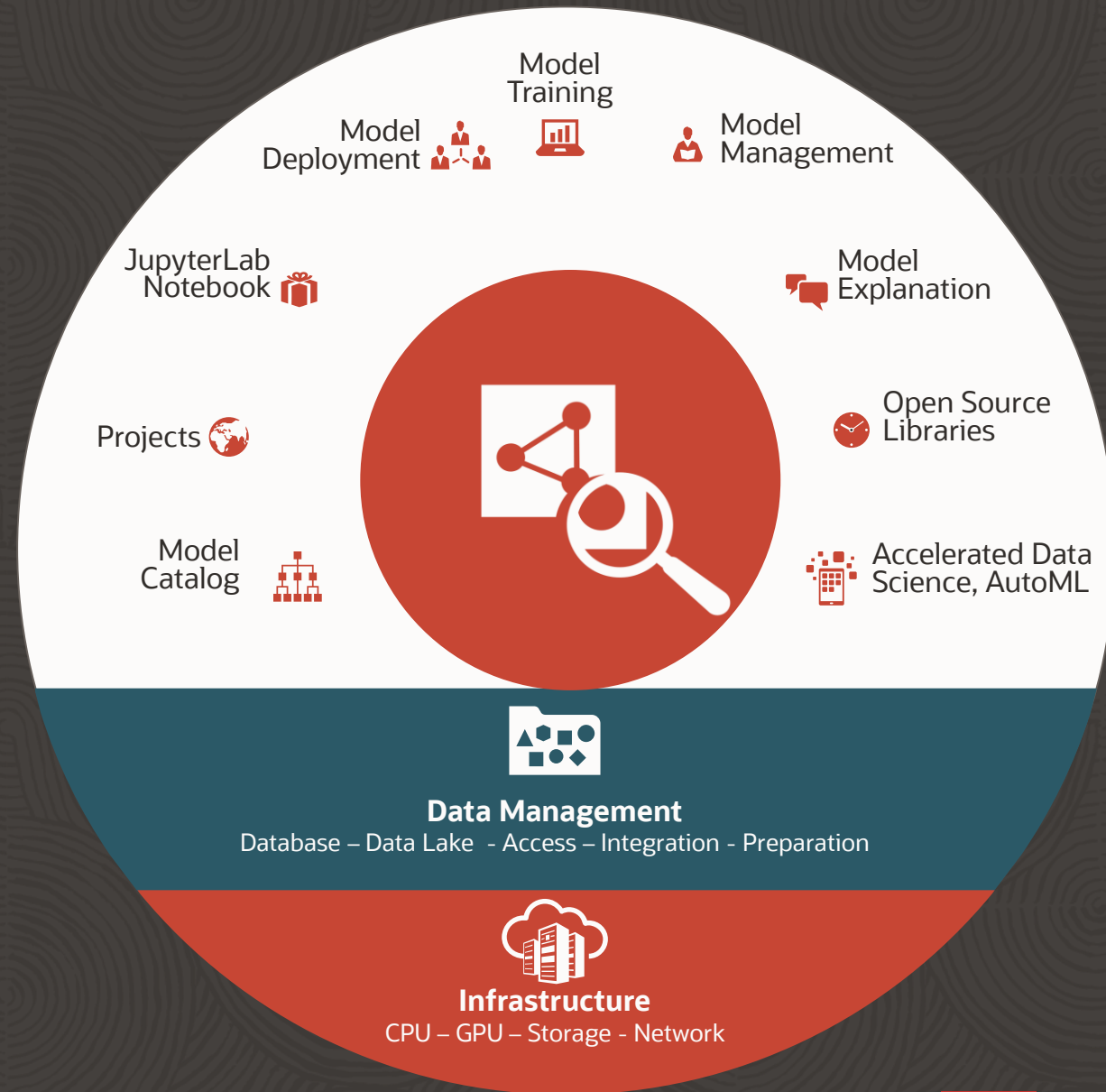
Allow data scientists to work the way they want to, and provide access to automated workflows, the best of open source, and a streamlined approach to building models.

Collaborative

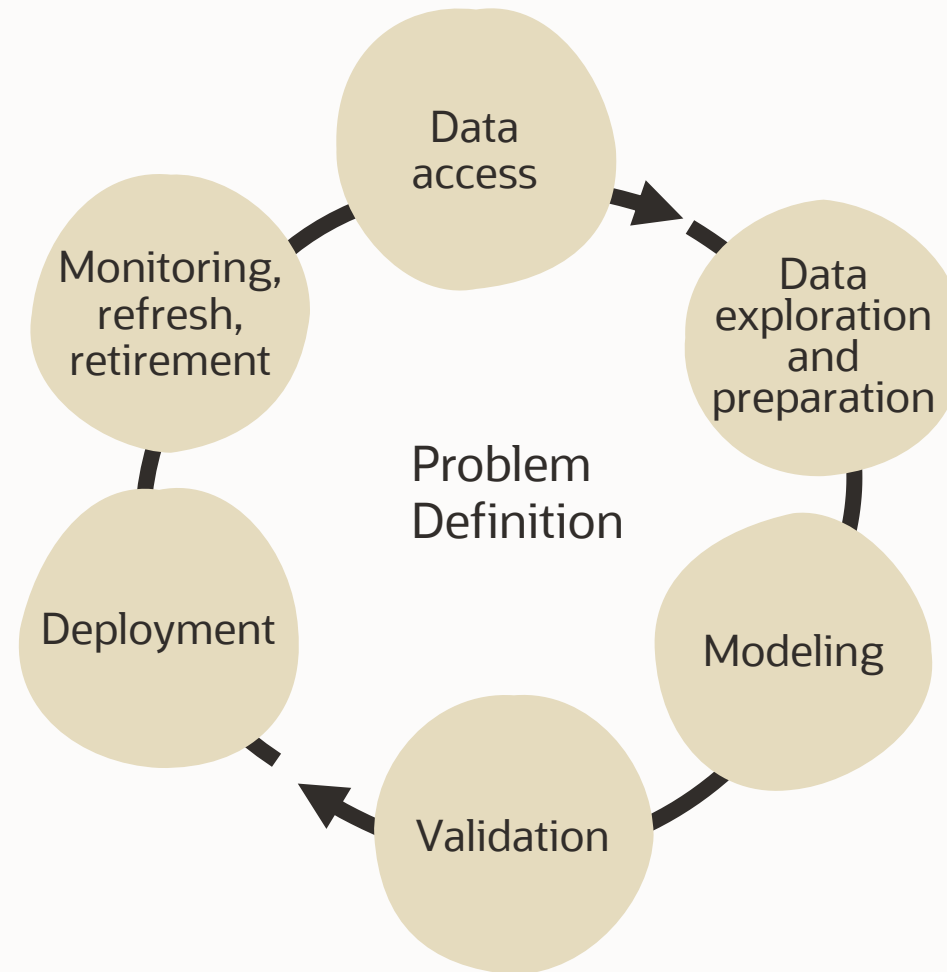
Enable data science teams to work together with ways to share and reproduce models in a structured, secure way for enterprise-grade results.

Enterprise-Grade

Provide a fully managed platform built to meet the needs of the modern enterprise

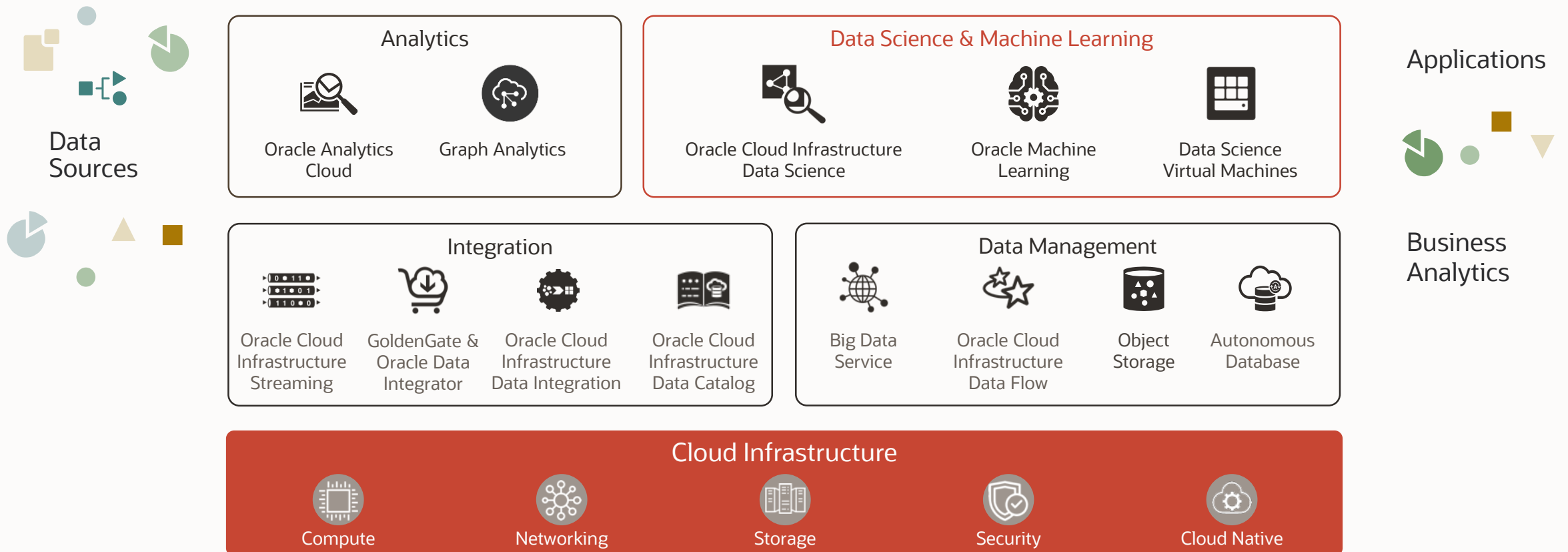


Data Science Project Lifecycle



Oracle Data Science Platform





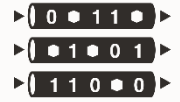



Machine learning supported by data sources, ingest, management and analytics







Data Access

Configurable networking and built-in Python connectors make data access flexible and easy

- Data Source-Agnostic
 - Oracle Cloud, other clouds, on-premises
- Data Format-Agnostic
 - Structured, unstructured, semi-structured
 - CSV, TSV, Parquet, libsvm, json, Excel, HDF5, SQL, xml, apache server log files (clf, log), arff, etc.

Oracle	Other
 Autonomous Database	 elasticsearch
 Oracle Object Storage	 S3
 Streaming	 Azure Blob Storage
 MySQL™	 Google Cloud Storage

Other Databases

 PostgreSQL	 Microsoft SQL Server
 mongoDB	 SQLite

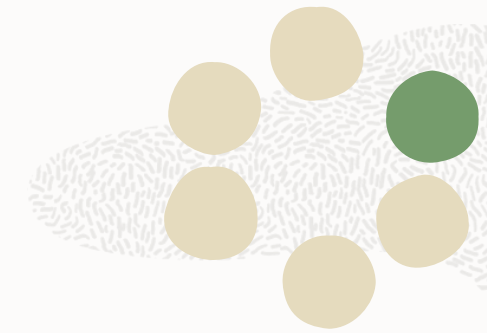


Data Transformation

Use your favorite open source tools

Perform custom data transformations

- Data imputation, normalization, encoding, etc. with pandas
- Distributed transformations and lazy executions with Dask
- Transformations on multi-dimensional arrays with numpy, optimizations done through numba for fast algorithms
- Advanced data processing and statistical analysis with Scipy
- Text processing with NLTK



Data Visualization

Use your favorite open source tools

—
Create custom visualizations

- Custom static charts with matplotlib
- Interactive charts with plotly and bokeh
- Out-of-the-box stylish charts and graphs templates with seaborn



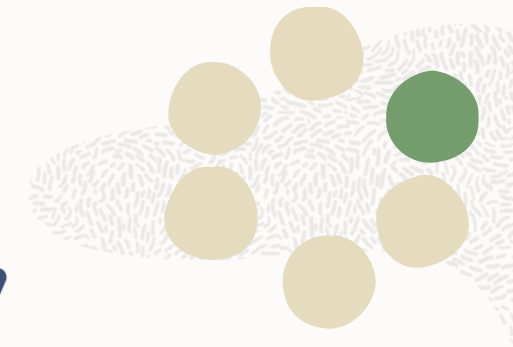
plotly

seaborn

matplotlib



bokeh



Model Training and Tuning

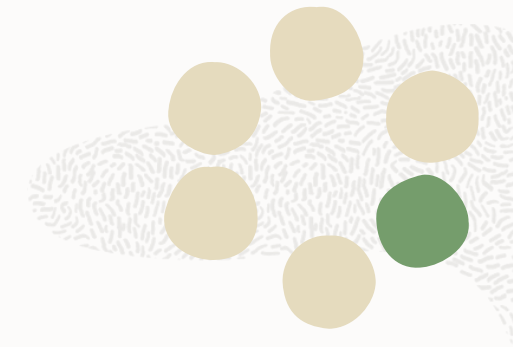
Use your favorite open source tools

- Use pre-installed machine learning and deep learning frameworks, including specialized libraries for NLP, time series, Bayesian techniques, anomaly detection, and more.
- Or, install any additional Python package
- Data scientists can select the amount of cloud computing resources they need to accomplish their modeling workloads: CPU and GPU VMs.



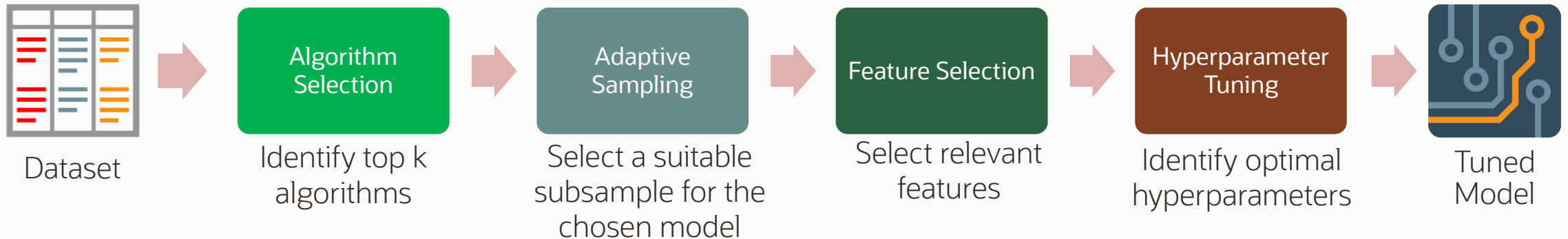
Model Training and Tuning

Take advantage of Oracle's Accelerated Data Science library (ADS)



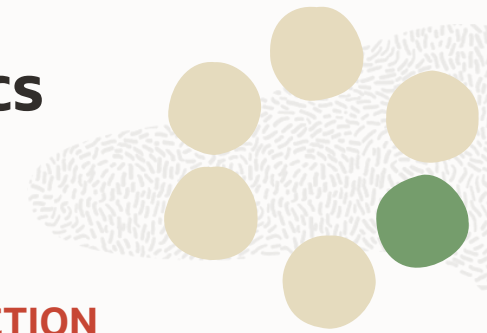
ADS offers Oracle's AutoML engine, developed over years of R&D in Oracle Labs

- Automated algorithm selection, data and feature selection, and hyperparameter tuning
- Optimized for data scientist expertise and time, runtime, and model performance



Oracle Machine Learning in-database Algorithms and Analytics

On Autonomous Database accessible in SQL and Python**



CLASSIFICATION

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

CLUSTERING

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

ANOMALY DETECTION

One-Class SVM
MSET-SPRT *

REGRESSION

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

TIME SERIES *

Forecasting - Exponential Smoothing
Includes popular models e.g. Holt-Winters with trends, seasonality, irregularity, missing data

ATTRIBUTE IMPORTANCE

Minimum Description Length
Principal Component Analysis *
Unsupervised Pair-wise KL Div *
CUR Decomposition *

ASSOCIATION RULES

A priori/ market basket

STATISTICAL FUNCTIONS

min, max, median, stdev, Pearson/
Kendall/Spearman correlation
Others: t-test, F-test,, Chi-Sq,
ANOVA, etc. *

FEATURE EXTRACTION

Principal Comp Analysis (PCA)
Non-negative Matrix Factorization
Singular Value Decomposition
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) *

SQL ANALYTICS *

SQL Windows
SQL Patterns
SQL Aggregates

* Available in SQL API only

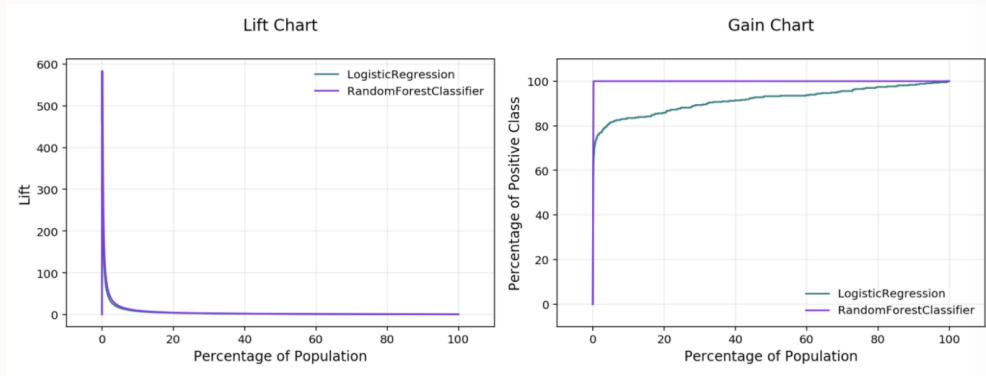
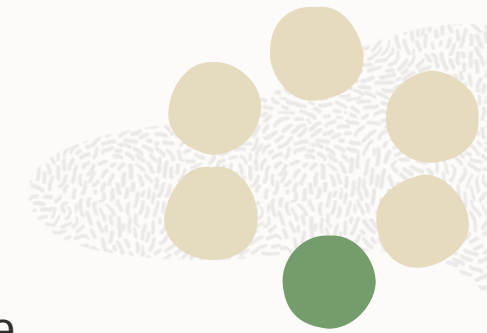
**future



Model Validation

Take advantage of Oracle's Accelerated Data Science library (ADS)

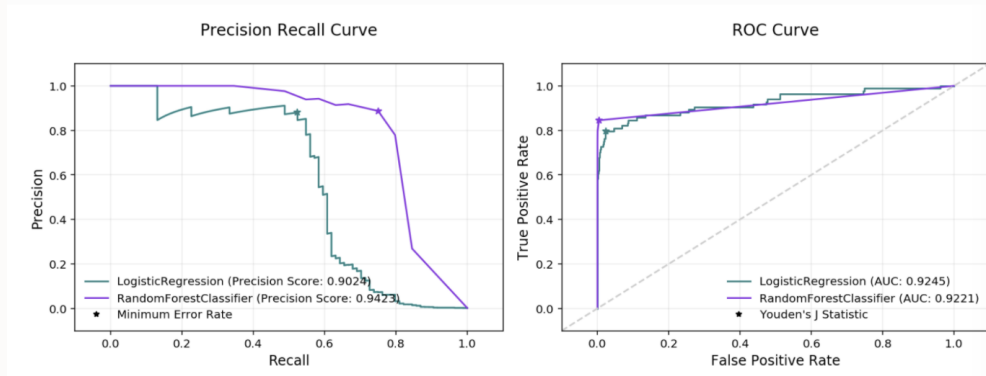
ADS Evaluator helps data scientists understand their models' accuracy and performance



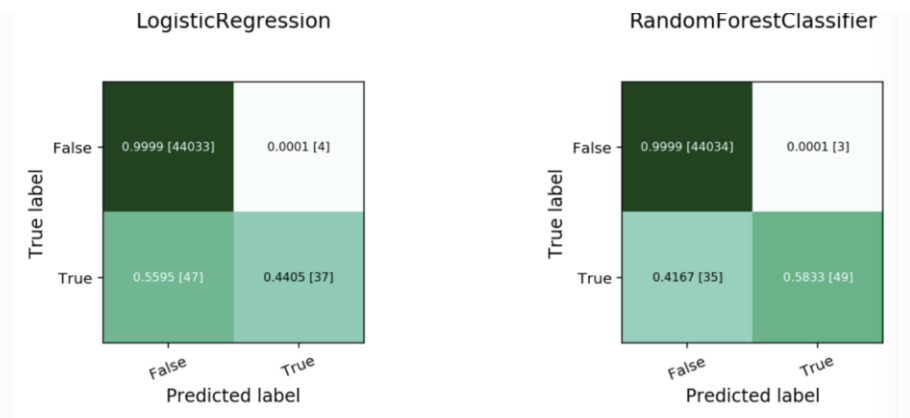
Lift & Gain Chart

Evaluation Metrics (testing data):

	LogisticRegression	RandomForestClassifier
accuracy	0.9988	0.9991
hamming_loss	0.001156	0.0008839
kappa_score_	0.5915	0.7268
precision	0.9024	0.8814
recall	0.4405	0.619
f1	0.592	0.7273
auc	0.9245	0.9042



PR & ROC Curves



Normalized Confusion Matrix

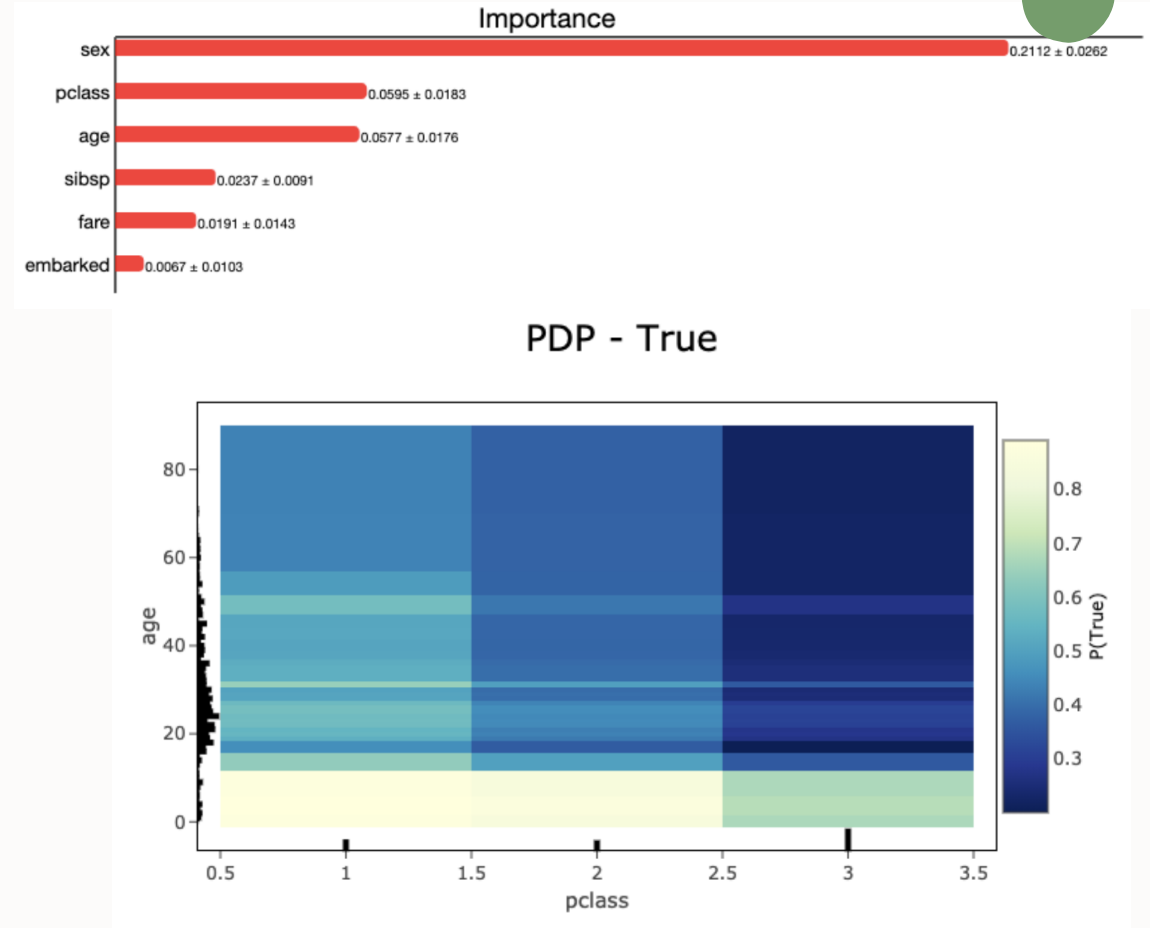


Model Validation

Take advantage of Oracle's Accelerated Data Science library (ADS)

ADS offers Oracle's MLX for Model Explanation, developed over years of R&D in Oracle Labs

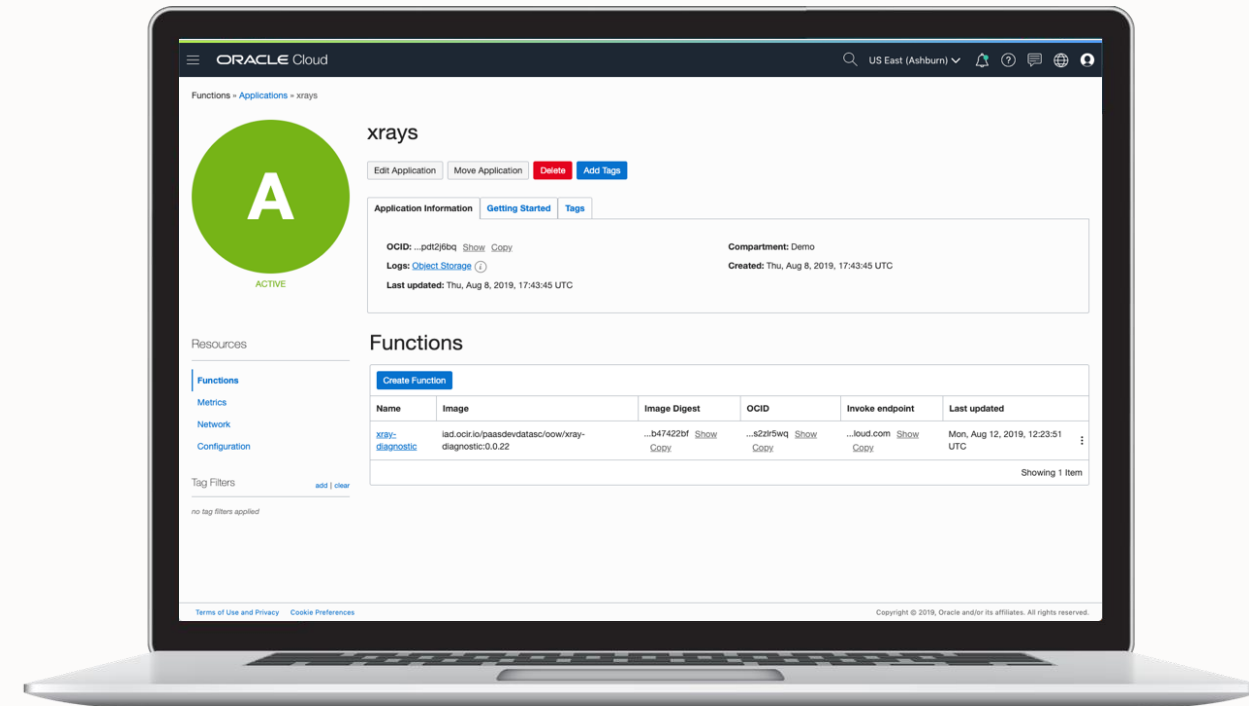
- Automated model-agnostic explanations improve understanding and trust, address regulatory needs, and increase speed of ML adoption
- Global explanations help explain the overall behavior of a model and local explanations explain specific model predictions



Deployment

Deploy models using Oracle Functions

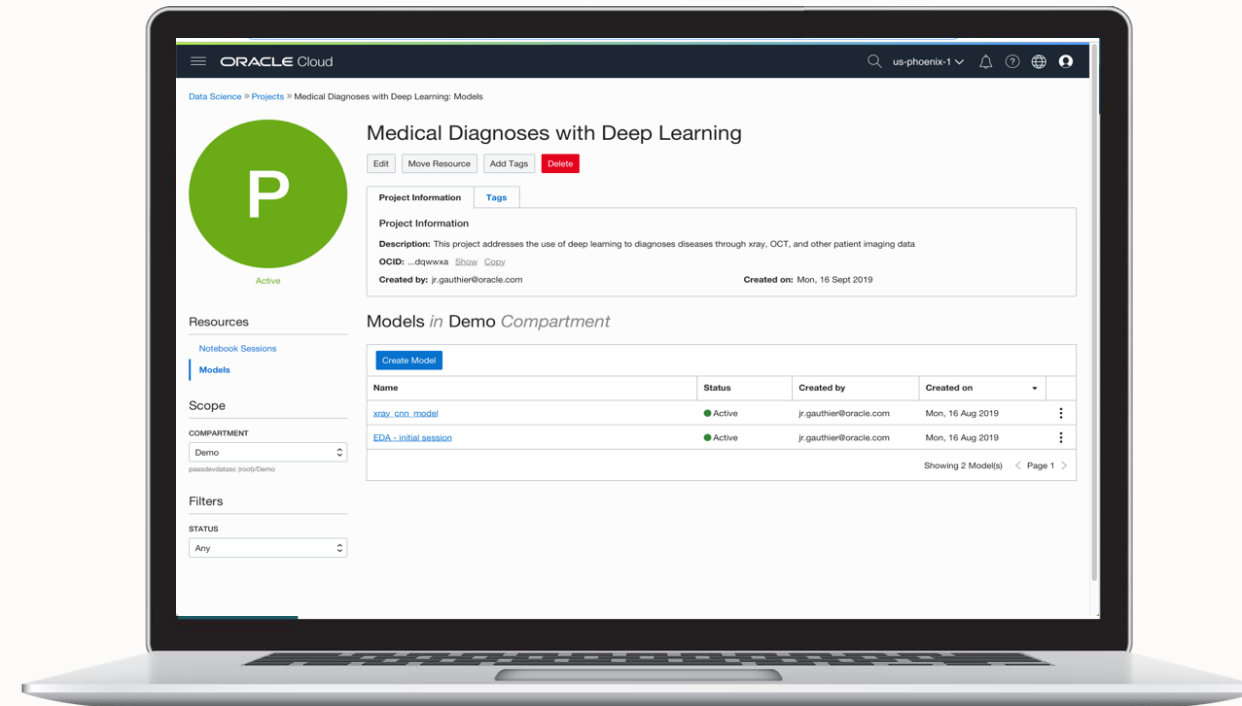
- One of many possible deployment targets
- Fully managed, highly scalable, on-demand, functions-as-a-service platform
- Invoke machine learning functions from any application



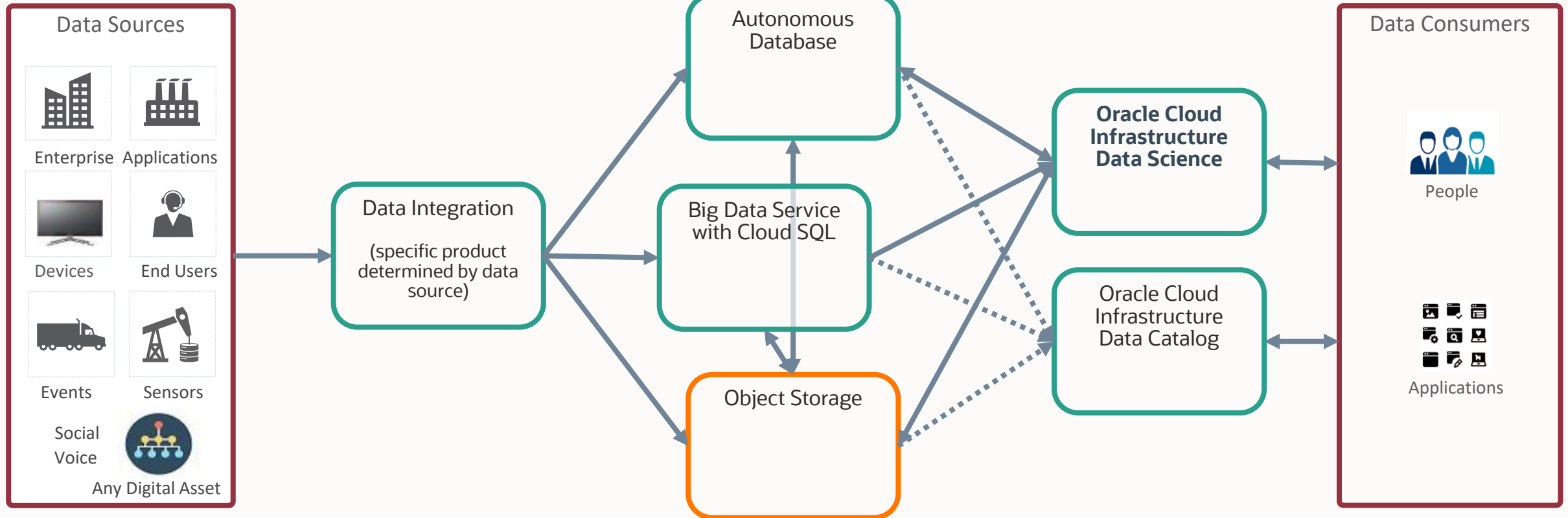
Model Management

The Model Catalog fosters collaboration and ensures model auditability and reproducibility amongst the data science team

- Track model metadata to answer where did this model come from?
- Save model artifacts in serviced-managed object storage to reproduce compute environment and model state
- Load models (your own or your teammates') for testing, validation, and deployment
- Deploy to Oracle Functions using the model artifact pre-written Functions files



Solution Pattern: ML with Oracle Cloud Infrastructure Data Science



Sum it up

- Cloud lock-in vs Lock into vendor's stack
- Incorporate open-source tools and processes
- Bring your data, bring your model
- Share your model, share your environment!

ORACLE

DEMO

Peter Szegedi
Oracle EMEA

Data Science Platform on OCI

The screenshot displays the Oracle Cloud console interface for a 'Notebook-test' session. At the top, the Oracle Cloud header includes the logo, 'Applications >' menu, a search bar, and the region 'Germany Central (Frankfurt)'. The breadcrumb trail shows 'Data Science » Projects » odsc_test: Notebook Sessions » Notebook-test'. The main content area features a large green circle with a white 'N' and the word 'ACTIVE' below it. To the right, there are action buttons: 'Open', 'Edit', 'Deactivate', 'Terminate', and 'More Actions'. Below these are two tabs: 'Notebook Session Information' (selected) and 'Tags'. The information section lists: OCID: ...wd2ty5bq, Created By: ...acle.com, Block Storage Size (in GB): 50 GB, Subnet: Private Subnet-DataScienceVCN, Created On: Wed, Jan 20, 2021, 10:04:34 UTC, Compute Instance Shape: VM.Standard.E2.2, and VCN: DataScienceVCN. A 'Resources' sidebar on the left has 'Metrics' selected. The 'Metrics' section contains an information box stating that metrics are only available for newly created or activated sessions. Below this are filters for 'START TIME' (Jan 20, 2021 3:43:18 PM), 'END TIME' (Jan 20, 2021 4:43:18 PM), and 'QUICK SELECTS' (Last hour). Two charts are shown: 'CPU UTILIZATION' and 'MEMORY UTILIZATION', both with a 1-minute interval and Mean statistic. The charts are currently empty.



Notebook session

The screenshot displays the Oracle Cloud Notebook interface. At the top, the header shows "ORACLE Cloud" and "Notebook-test" with a "Sign Out" button. Below the header is a menu bar with options: File, Edit, View, Run, Kernel, Tabs, Settings, and Help. The interface is divided into two main sections. On the left is a file explorer with a sidebar containing icons for home, refresh, and a hamburger menu. The main area of the file explorer shows a directory structure with a table of files:

Name	Last Modified
/	
conda	6 hours ago
Damaged boxes.ipynb	6 hours ago

On the right is the "Launcher" section, which is organized into three categories:

- Notebook**: Contains four options: Environment Explorer, Python 3, Notebook Examples, and Python [conda env:root] *.
- Console**: Contains two options: Python 3 and Python [conda env:root] *.
- Other**: Contains four options: Terminal, Text File, Markdown File, and Show Contextual Help.

At the bottom of the interface, there is a status bar showing "0" and "1" next to a gear icon, and the word "Launcher" in the bottom right corner.



Code Environments

The screenshot shows the Oracle Cloud Infrastructure Data Science Conda Environment Explorer interface. The top navigation bar includes the Oracle Cloud logo, a 'Notebook-test' link, and a 'Sign Out' button. Below the navigation bar is a menu with options: File, Edit, View, Run, Kernel, Tabs, Settings, and Help. The main content area is titled 'Oracle Cloud Infrastructure Data Science Conda Environment Explorer' and contains a descriptive paragraph about the service. Below the text are four tabs: 'All Conda Environments', 'Data Science Conda Environments' (which is selected), 'Installed Conda Environments', and 'Published Conda Environments'. There is also a refresh button. The interface displays three environment cards, each with an Oracle logo, a title, a version number (v1.0), and a description. The first card is 'CLASSIC CPU NOTEBOOK SESSION KERNEL', the second is 'ORACLE DATABASE', and the third is 'CLASSIC GPU NOTEBOOK SESSION KERNEL'. Each card includes details about the Python version, CPU/GPU, creation time, and size, as well as instructions on how to use the environment and a 'Notebook Examples launcher button'. The bottom of the interface shows a status bar with '0', '1', and 'Environment Explorer'.



Example of damaged box detection

The screenshot displays an Oracle Cloud Notebook interface. The top navigation bar includes 'ORACLE Cloud', 'Notebook-test', and a 'Sign Out' button. Below the navigation bar is a menu with 'File', 'Edit', 'View', 'Run', 'Kernel', 'Tabs', 'Settings', and 'Help'. The left sidebar shows a file explorer with a 'conda' folder and a 'Damaged boxes.ipynb' file, both last modified 6 hours ago. The main workspace shows a Python 3 kernel with a script named 'Damaged boxes.ipynb'. The script iterates through 'train' and 'test' sets, displaying images of boxes. The output shows four plots: 'Set: train, Not Damaged' (a standard brown cardboard box), 'Set: test, Not Damaged' (another standard brown cardboard box), 'Set: train, Damaged' (a box with a small tear), and 'Set: test, Damaged' (two boxes, one with a significant tear and another with a large hole).

```
for i, _set in enumerate(['train', 'test']):
    set_path = DATADIR+_set
    ax[i].imshow(plt.imread(set_path+'/damaged_no'+os.listdir(set_path+'/damaged_no')[0]), cmap='gray')
    ax[i].set_title('Set: {}, Not Damaged'.format(_set))
    ax[i+2].imshow(plt.imread(set_path+'/damaged_yes'+os.listdir(set_path+'/damaged_yes')[0]), cmap='gray')
    ax[i+2].set_title('Set: {}, Damaged'.format(_set))
```

Set: train, Not Damaged

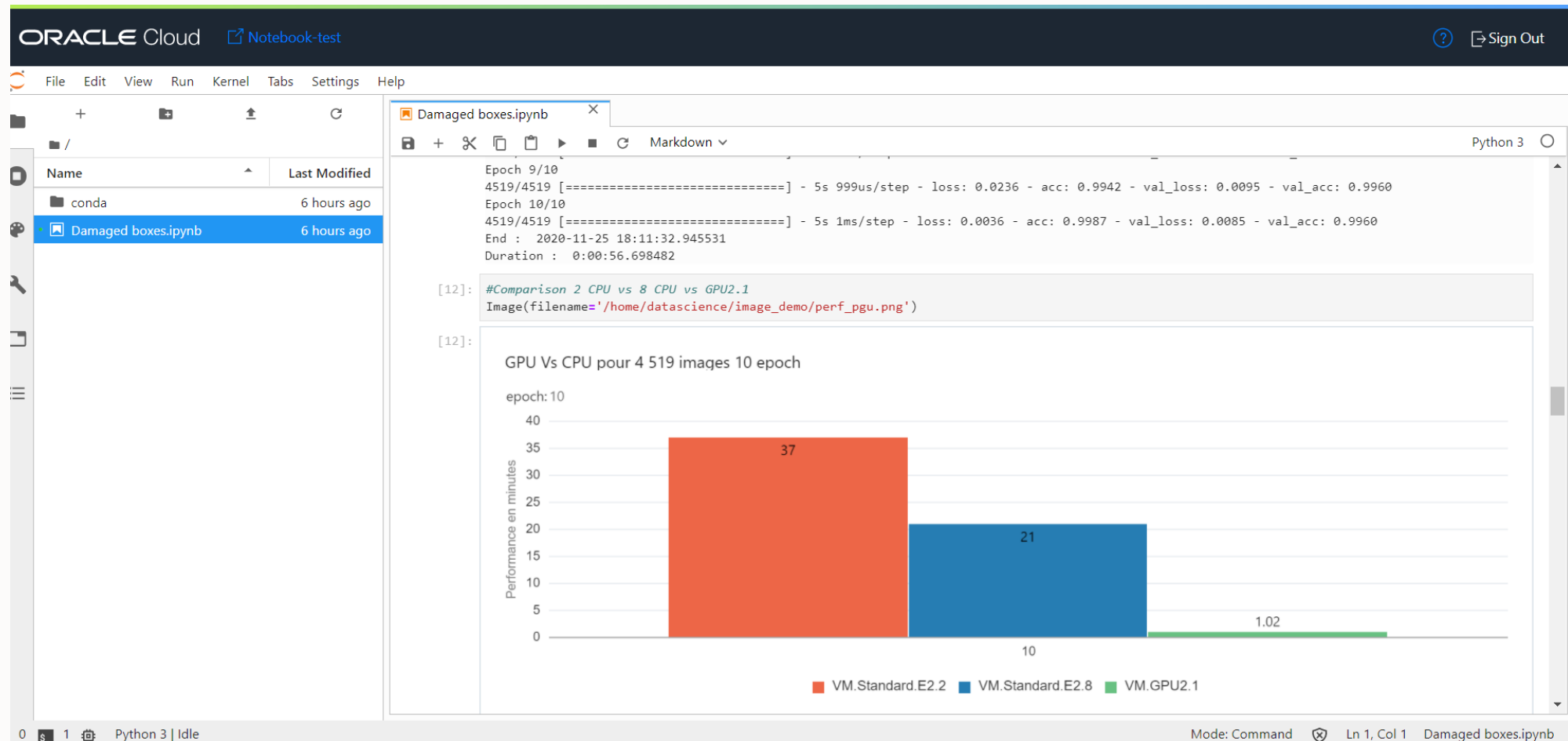
Set: test, Not Damaged

Set: train, Damaged

Set: test, Damaged



CPU vs GPU performance comparison



Model deployment

The screenshot displays an Oracle Cloud Jupyter Notebook interface. The notebook is titled "Damaged boxes.ipynb" and is running Python 3. The code in the notebook cell is as follows:

```
plt.xticks(range(2))
plt.yticks([])
thisplot = plt.bar(range(1), prediction, color="#888888")
plt.ylim([0,1])
predicted_label = np.argmax(prediction)
thisplot[predicted_label].set_color('red')
```

```
plt.figure(figsize=(20,10))
for i in range(2):
    # image
    plt.subplot(2, 4, 2*i+1)
    plot_image(preds[i], images[i])
    # bar chart
    plt.subplot(2, 4, 2*i+2)
    plot_value_array(preds[i])
plt.show()
```

The output of the code shows two rows of plots. The first row corresponds to the first image (a damaged box) and the second row to the second image (a healthy box). Each row contains an image plot and a bar chart plot. The bar chart plots show the predicted probability for each class (0 and 1). The predicted label is highlighted in red in the bar chart.

Image	Class 0 Probability	Class 1 Probability	Predicted Label
Damaged box (FRAGILE)	0.23	0.77	1
Healthy box	0.99	0.01	0

The status bar at the bottom of the notebook shows "Mode: Command", "Ln 1, Col 1", and "Damaged boxes.ipynb".

