Database Driven Machine Learning

Doug Hood
@ScalableDBDoug
Consulting Member of Technical Staff + Cloud Product Manager

October 25, 2018

Safe Harbor Statement

The preceding 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, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

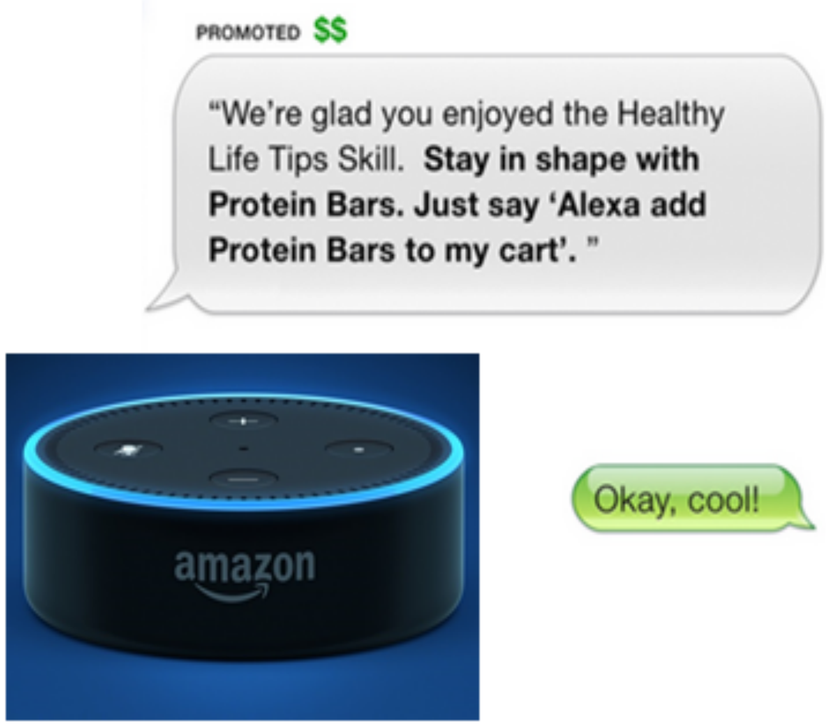
Agenda

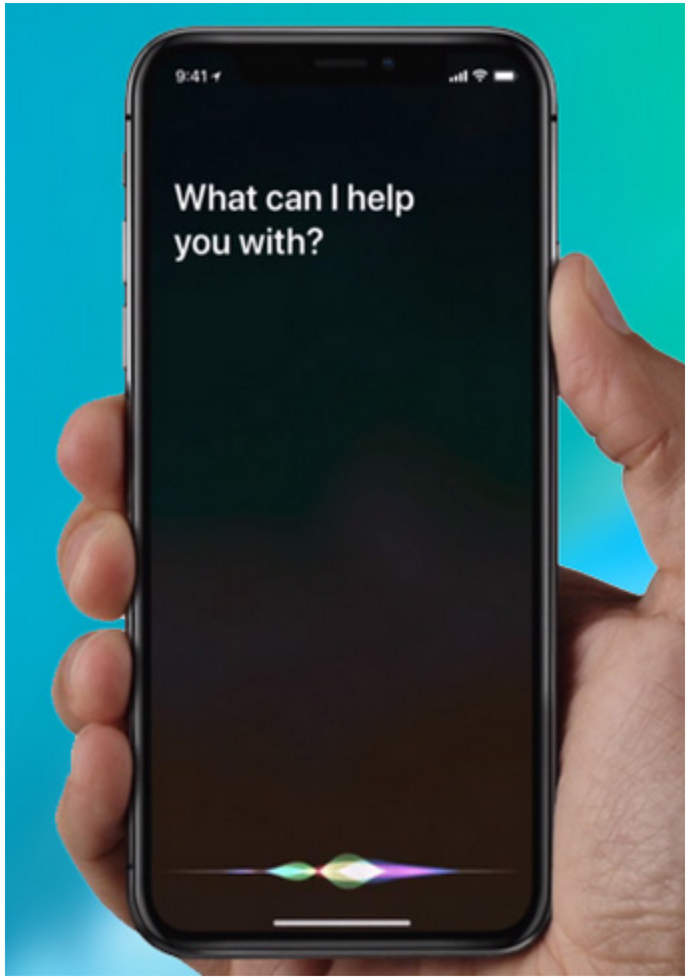
1. Introduction to Machine Learning
2. Machine Learning using internal database algorithms
3. Machine Learning using external database algorithms
4. Machine Learning for Autonomous DB
5. Summary and Q & A

ORACLE

Voice recognition and Natural Language Processing

Siri, Alexa, Cortana and Google Assistant are arguing again …





Computer vision – needs feature extraction

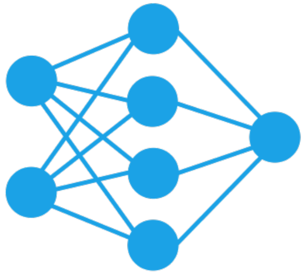
Detected vehicles with heatmaps and thresholding

Computer Vision & Self driving cars – are they ready?

Governor Doug Ducey tells Uber crash raises concerns about its ability to safely test technology

▲ Transport safety investigators examine a self-driving Uber vehicle involved in a fatal accident in Tempe, Arizona. Photograph: Reuters

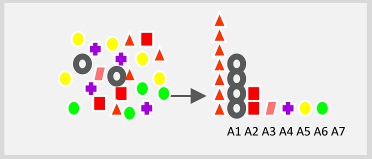
What is Machine Learning?

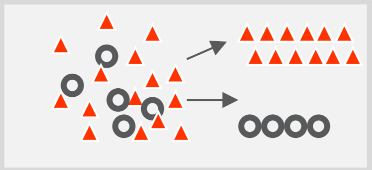


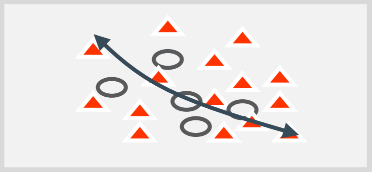
- Ask ten different Machine Language **researchers** or **practitioners** and you will get **ten different answers**
- **Machine learning** is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" (e.g., **progressively improve performance on a specific task**) with **data**, without being explicitly programmed [*Samuel, A, 1959*]
- **Ads** are "perhaps by far the most lucrative application of AI [and] machine learning in the industry" [Cosley, J, 2017]

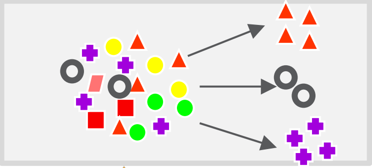
Some other uses for Machine Learning

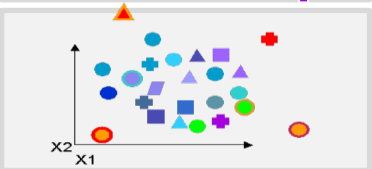
- Identify most important factor *(Attribute Importance)*
- Predict customer behavior *(Classification)*
- Predict or estimate a value *(Regression)*
- Find profiles of targeted people or items *(Decision Trees)*
- Segment a population *(Clustering)*
- Find fraudulent or “rare events” *(Anomaly Detection)*
- Determine co-occurring items in a “baskets” *(Associations)*

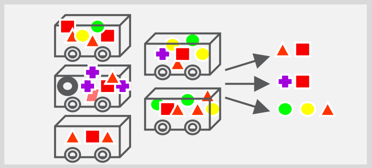




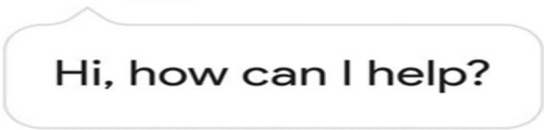




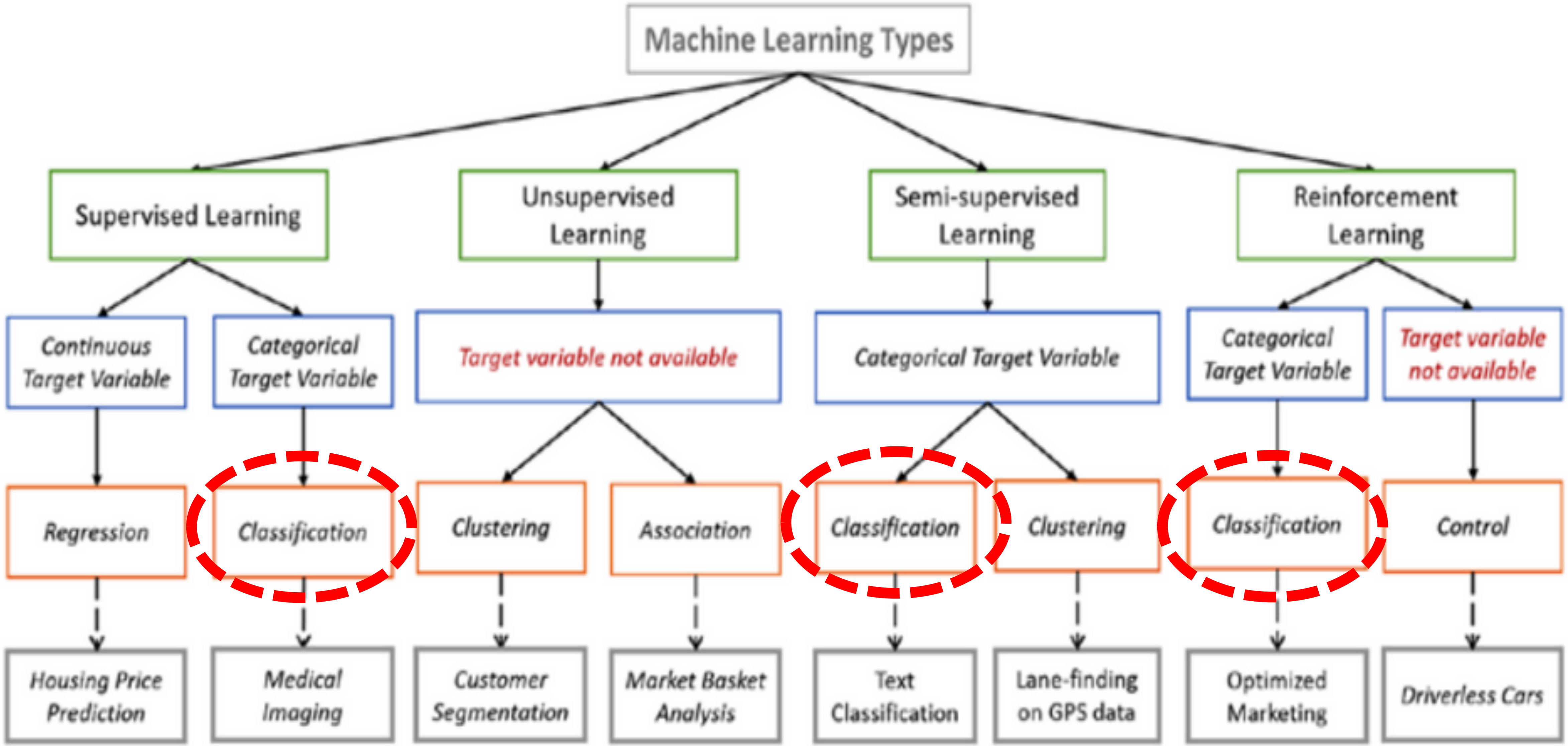




Examples of Machine Learning making or saving $$$

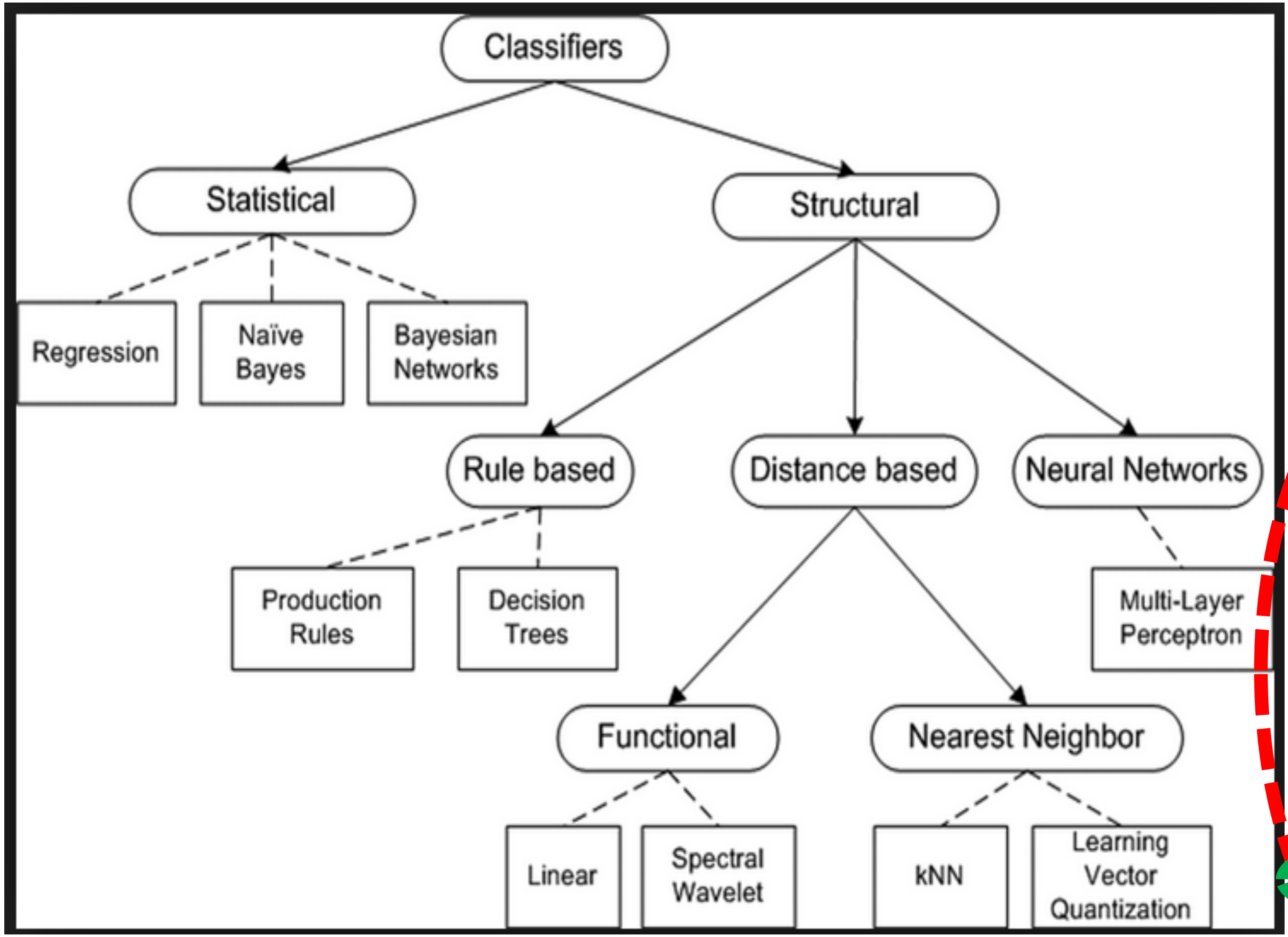
Name	Description	Examples
Classification	Predict customer behavior / targeted ads	Google f 阿里巴巴 Alibaba.com
Regression	Predict or estimate a value	FINANCIAL SERVICES
Voice Recognition and NLP & Chatbots	Automatically engage customers more cheaply	
Anomaly Detection	Find fraudulent or rare events	FINANCIAL SERVICES
Recommenders	Given a product, suggest other related products	amazon.com NETFLIX
Associations	Determine co-occurring items in a basket	
Clustering	Segment data and give probability that value will be in cluster	Cambridge Analytica

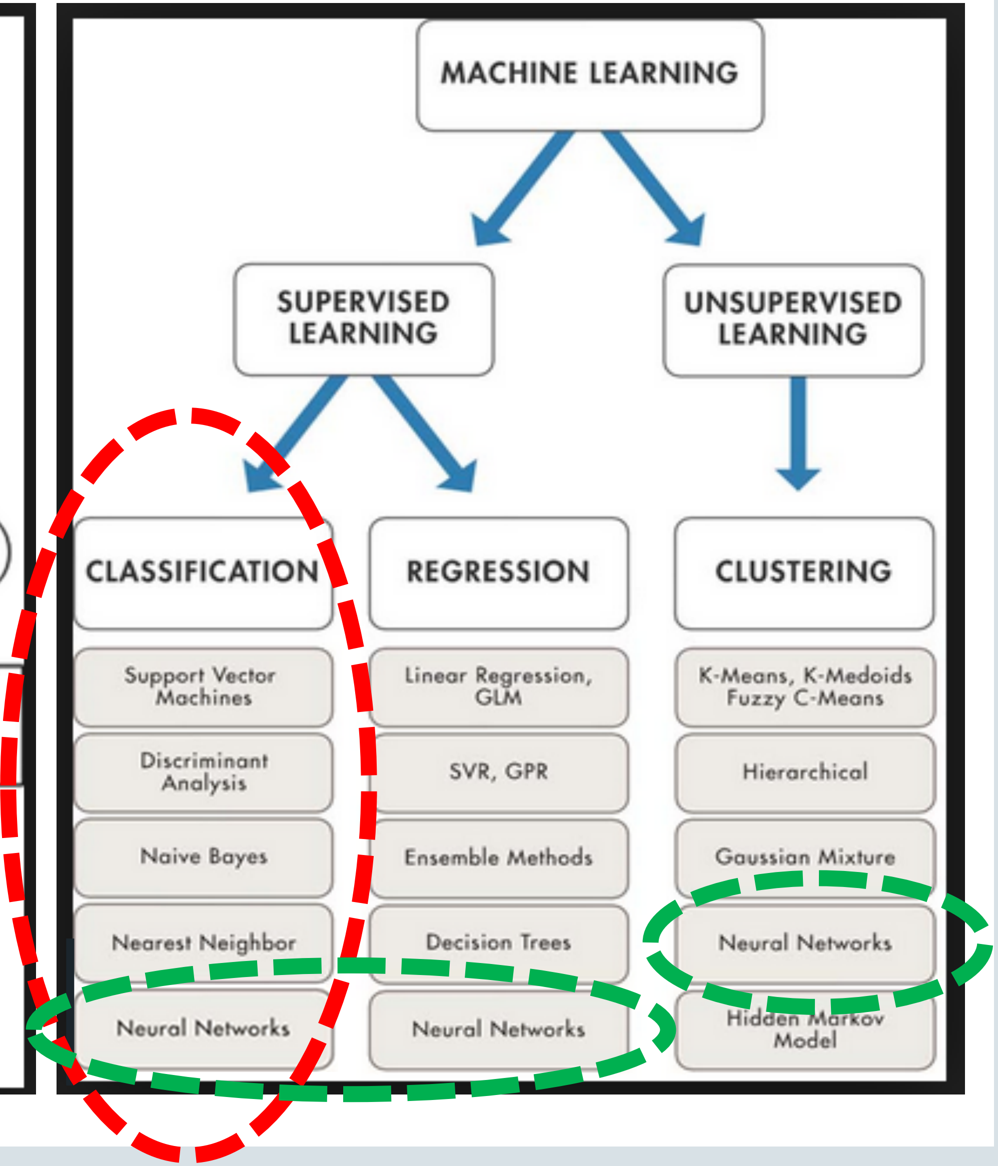
The four main types of Machine Learning



ORACLE

Machine Learning Algorithms

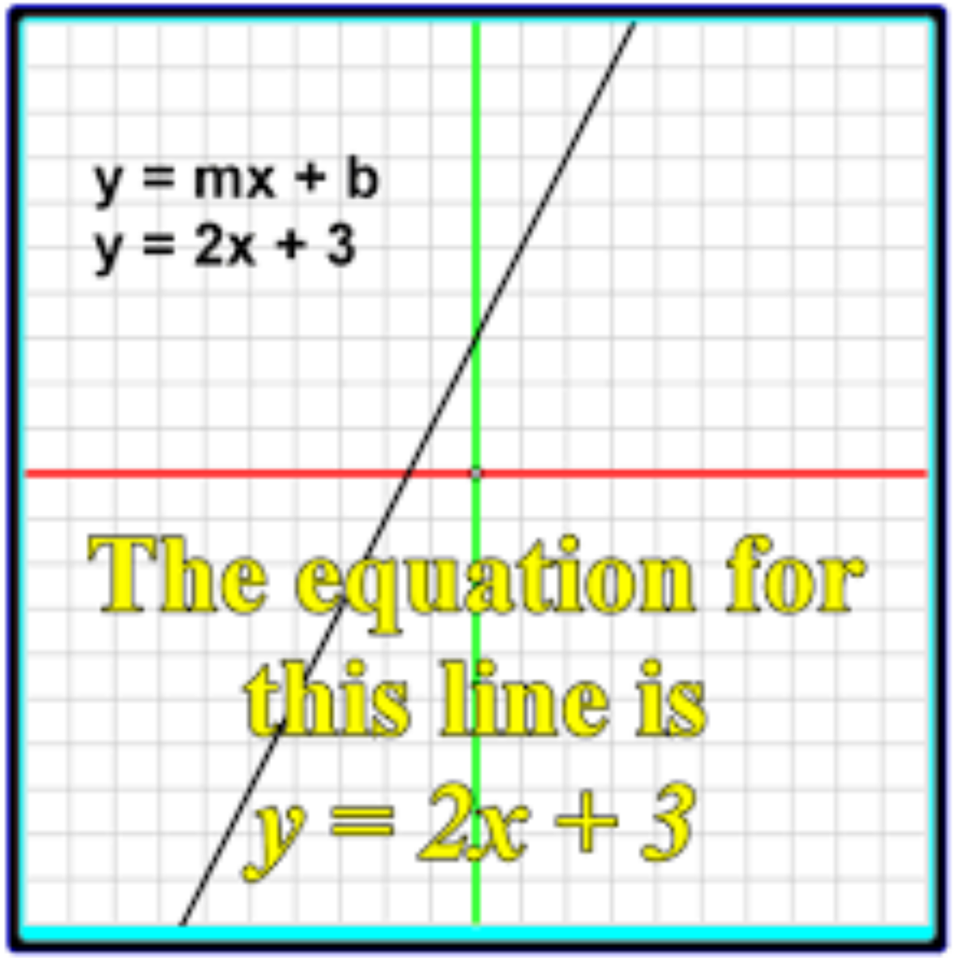




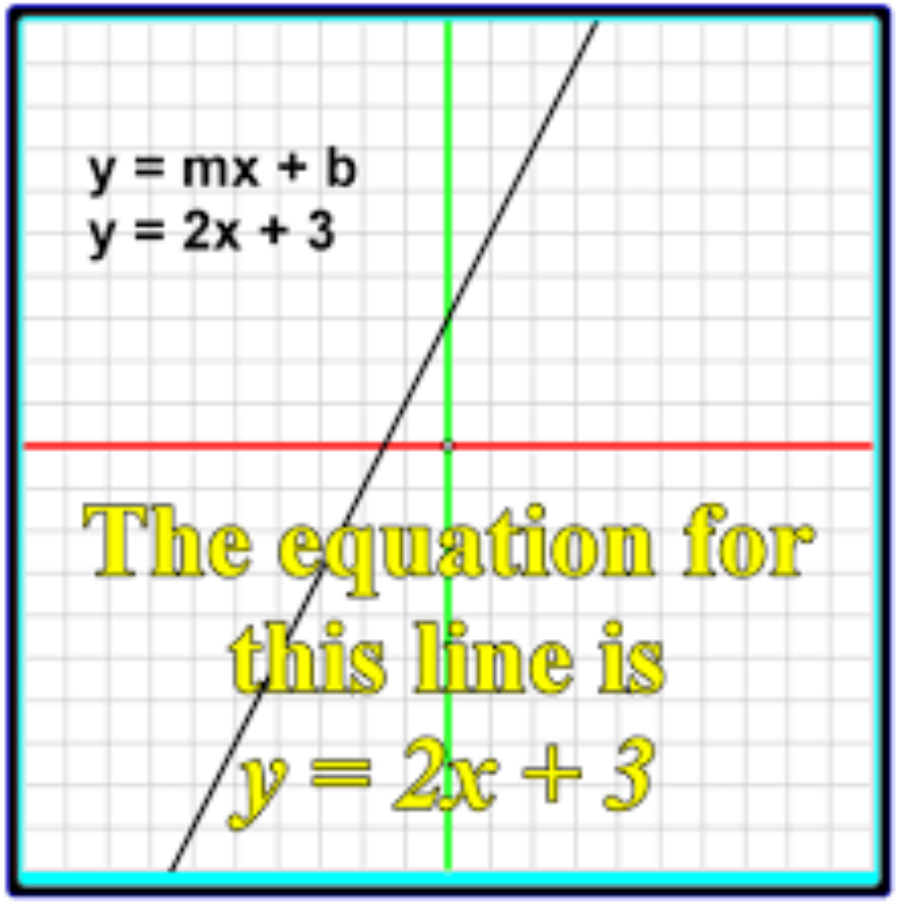
Popular Machine Learning Languages & Libraries

Language	Libraries	Features
python	119	Computer Vision, NLP, general purpose ML, data analysis and visualization, deep learning, reinforcement learning and Kaggle competition source code
R	90	Regression analysis, feature selection, classification, decision trees, rule based models, gradient boosting, inference and prediction, time series forecasting, random forests, deep learning, Bayesian and Gaussian models
Java	57	NLP, probabilistic ML, speech recognition, data analysis and visualization, deep learning, classification and evaluation
C++	46	Computer vision, NLP, speech recognition, sequence analysis, gesture detection, deep learning, gradient boosting and CUDA
JS	43	NLP, data analysis and visualization, deep learning, clustering, decision trees, Bayesian and Gaussian models, topic modeling, K-means, SVM, regression
	26	NLP, pattern recognition, Bayesian Classification, decision trees, deep learning, data analysis and visualization, facial recognition, image classification
.NET	16	Computer vision, NLP, deep learning, Bayesian inference, data analysis and visualization

The simplest Machine Learning Model?



The simplest Machine Learning Model?



- Calculate **M** and **B**
- Given X determine Y
- Can be used for prediction

- **Y = 2X + 3** is the **model**
- M and B are **parameters [derived from data]**
- Use linear **regression** to determine the **model**

Most models are much **more complex**
Most models are a **bunch of numbers**
Hyperparameters are set by the **data scientist**

- Not determined by the data

Why is Machine Learning hard?

Theorem 1: For any algorithms a_1 and a_2, at iteration step m

$$\sum_f P(d_m^y | f, m, a_1) = \sum_f P(d_m^y | f, m, a_2),$$

Why is Machine Learning hard?

Theorem 1: For any algorithms a_1 and a_2, at iteration step m

$$\sum_f P(d_m^y | f, m, a_1) = \sum_f P(d_m^y | f, m, a_2),$$

(Wolpert and Macredy, 1997)

- No ML algorithm is better than all others for all datasets and problem domains
- Some algorithms are better than others for specific datasets and problem domains
- Each algorithm's model needs meaningful [hyper] parameters

You need **EXPERIENCE** to choose the relevant/best algorithm + [hyper] parameters + weights for your datasets and problem domain

A fool with a tool is still a fool, Grady Booch

To succeed with Machine Learning you need

A well defined problem

Things change

Lots of clean, normalized data

Productize the solution
(effective, efficient & robust)

Actionable insights

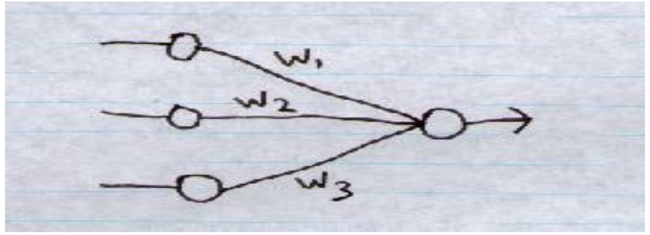
Find the best attributes

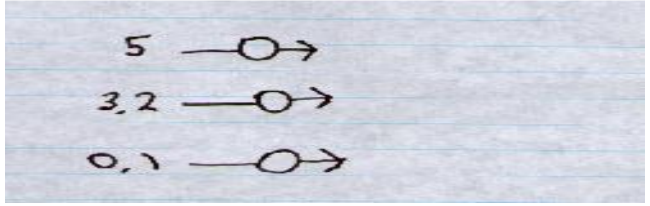
Time to train the model

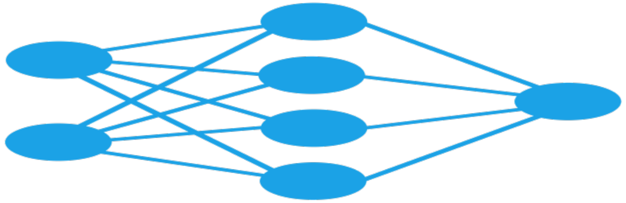
‘Best’ algorithm & [hyper] parameters

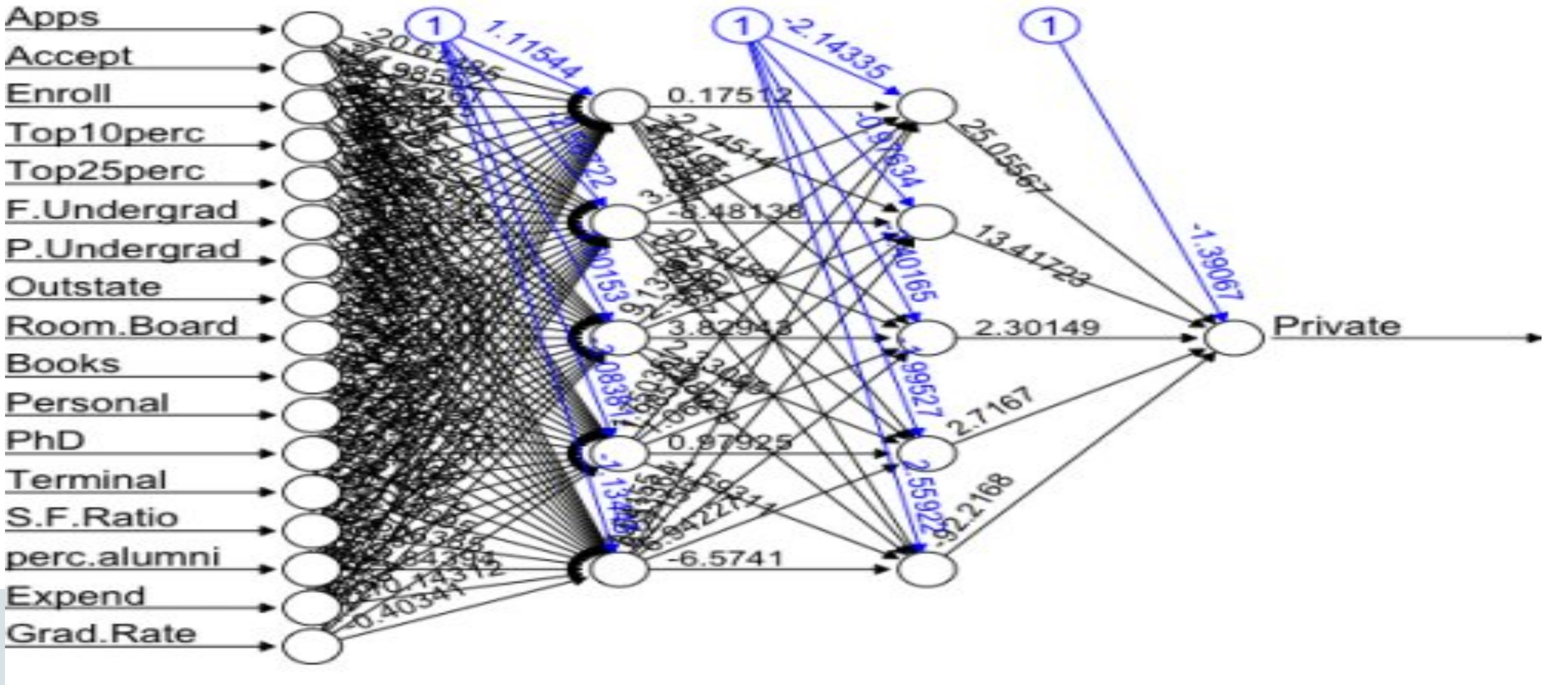
Neural Networks in 60 seconds

- An old concept (1943, 1949)
- Hot in the 1990s and now with GPUs and the Cloud
- Use perceptrons to mimic biological neurons
- A set of inputs
- Usually a single output
- Choose the number of perceptrons per layer
- Choose the number of hidden layers
- Train sample inputs to desired outputs
- Training creates weights
- Can take a long time to train





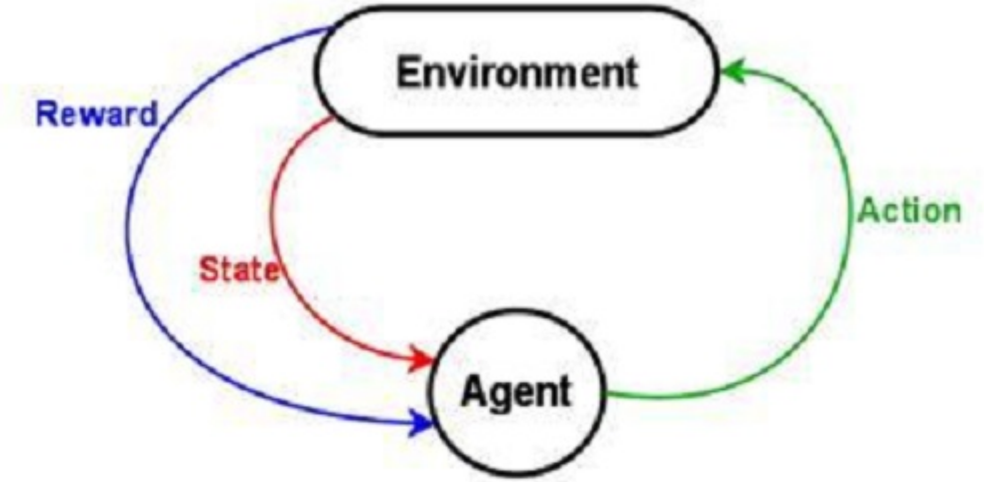


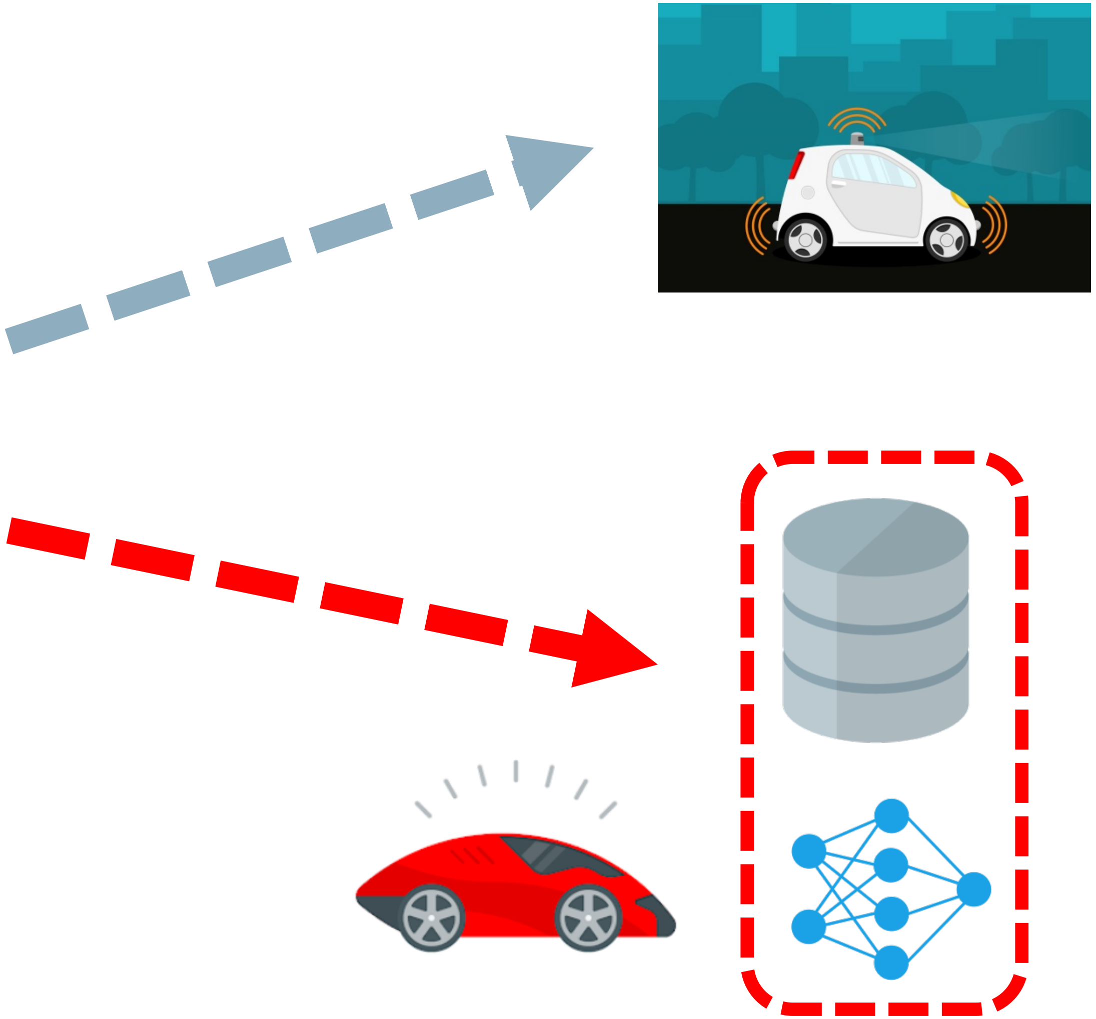


Reinforcement Learning in the Real World

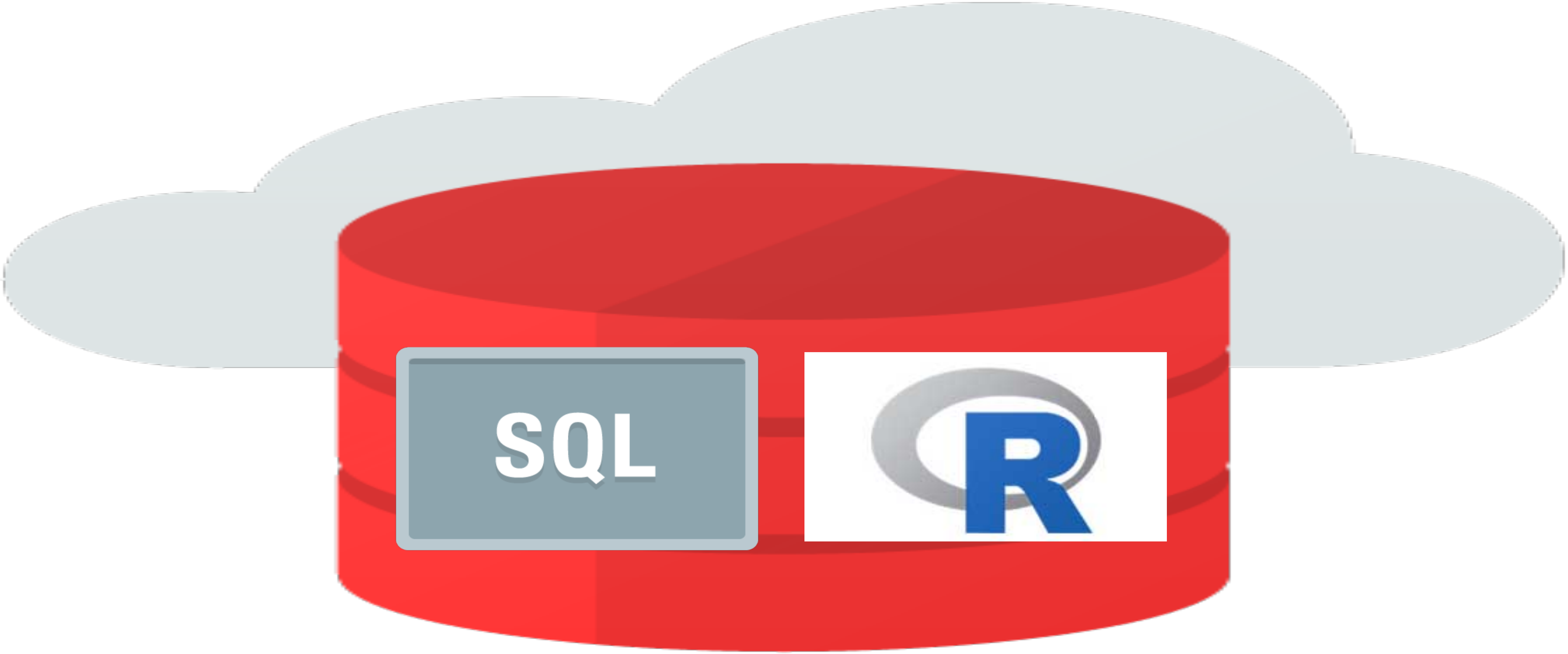
- Reward good behavior
- Punish bad behavior

Reinforcement Learning for Machine Learning





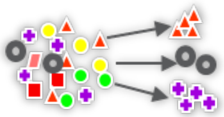
Machine Learning using internal database algorithms



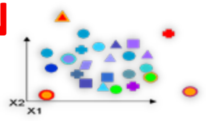
Oracle's Machine Learning & Adv. Analytics Algorithms

Classification


- Naïve Bayes
- Logistic Regression (GLM)
- Decision Tree
- Random Forest
- Neural Network
- Support Vector Machine

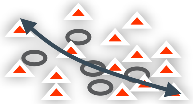
Clustering


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

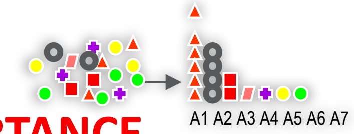
Anomaly Detection


- One-Class SVM

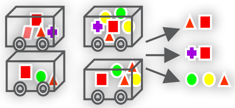
Time Series
- Holt-Winters, Regular & Irregular, with and w/o trends & seasonal
- Single, Double Exp Smoothing

Regression


- Linear Model
- Generalized Linear Model
- Support Vector Machine (SVM)
- Stepwise Linear regression
- Neural Network
- LASSO

Attribute Importance


- Minimum Description Length
- Principal Comp Analysis (PCA)
- Unsupervised Pair-wise KL Div
- CUR decomposition for row & AI

Association Rules


- A priori/ market basket

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)

Text Mining Support
- Algorithms support text type
- Tokenization and theme extraction
- Explicit Semantic Analysis (ESA) for document similarity

Statistical Functions


- Basic statistics: min, max, median, stdev, t-test, F-test, Pearson's, Chi-Sq, ANOVA, etc.

R Packages
- CRAN R Algorithm Packages through Embedded R Execution
- Spark MLlib algorithm integration

Exportable ML Models
- C and Java code for deployment

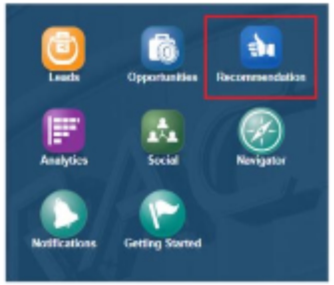
ORACLE

Enabling “Predictive” Enterprise Applications

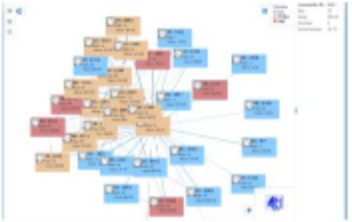
Oracle Applications Using Oracle Advanced Analytics—Partial List

- **Oracle HCM Cloud**
  - Employee turnover and performance prediction and *“What if?”* analysis

- **Oracle Sales Cloud**
  - Prediction of sales opportunities, what to sell, amount, timing, etc.

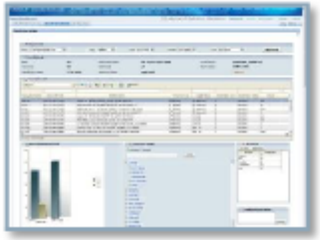


- **Oracle Industry Data Models**
  - **Communications Data Model** churn prediction, segmentation, profiling, etc.
  - **Retail Data Model** loyalty and market basket analysis
  - **Airline Data Model** analysis frequent flyers, loyalty, etc.
  - **Utilities Data Model** customer churn, cross-sell, loyalty, etc.



- **Oracle Retail GBU Cloud Services**
  - Market Basket Analysis Insights
  - Customer Insights & Clustering

- **Oracle Customer Support**
  - Predictive Incident Monitoring (PIM)
- **Oracle Spend Classification**
  - Real-time and batch flagging of noncompliance and anomalies in expense submissions



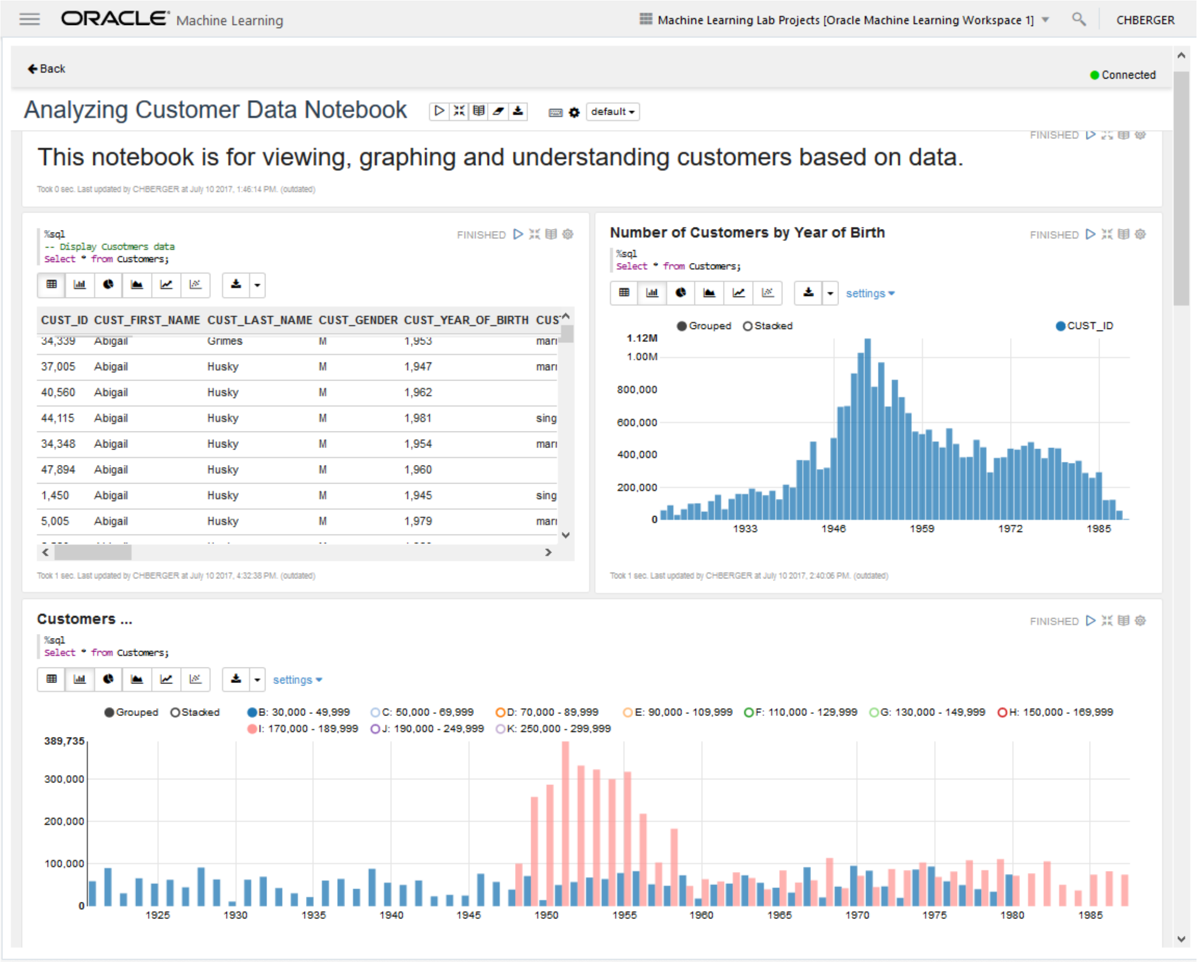
- **Oracle FinServ Analytic Applications**
  - Customer Insight, Enterprise Risk Management, Enterprise Performance, Financial Crime and Compliance
- **Oracle Adaptive Access Manager**
  - Real-time security and fraud analytics

Oracle Machine Learning

Machine Learning Notebook for Autonomous Data Warehouse Cloud

Key Features

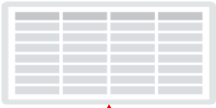
- Collaborative UI for data scientists
  - Packaged with Autonomous Data Warehouse Cloud (V1)
  - Easy access to shared notebooks, templates, permissions, scheduler, etc.
  - SQL ML algorithms API (V1)
  - Supports deployment of ML analytics



The Oracle Advanced Analytics Approach / Method

1.

2.



3.

Create & run the model

DBMS_DATA_MINING.CREATE_MODEL()

- **Model name**
- **Settings table name**
- **Your data table/view name**
- **Case id column name**
- Xform list
- Target column name
- **Mining Function**
- …

4.

Analyze the results

Mining Function Value	Algorithms
ASSOCIATION	Apriori
ATTRIBUTE_IMPORTANCE	Min Description Length & CUR Matrix Decomposition
CLASSIFICATION	Naive Bayes, Neural Network, Decision Tree, Logistic Regression, SVM & Explicit Semantic Analysis
CLUSTERING	K-Means, O-Cluster & Expectation Maximization
FEATURE_EXTRACTION	Non negative matrix factorization, singular value decomposition & explicit semantic analysis
REGRESSION	Support Vector Machine , Linear Regression & Neural Network
TIME_SERIES	Exponential Smoothing

ORACLE

Fraud Prediction Demo

Automated In-DB Analytical Methodology

```
create table CLAIMS_SET (setting_name varchar2(30), setting_value varchar2(4000));
insert into CLAIMS_SET values ('ALGO_NAME','ALGO_SUPPORT_VECTOR_MACHINES');
insert into CLAIMS_SET values ('PREP_AUTO','ON');
commit;
```

```
begin
dbms_data_mining.create_model('CLAIMSMODEL', 'CLASSIFICATION',
  'CLAIMS', 'POLICYNUMBER', null, 'CLAIMS_SET');
end;
/
```

```
-- Top 5 most suspicious fraud policy holder claims
select * from
(select POLICYNUMBER, round(prob_fraud*100,2) percent_fraud,
     rank() over (order by prob_fraud desc) rnk from
(select POLICYNUMBER, prediction_probability(CLAIMSMODEL, '0' using *) prob_fraud
from CLAIMS
where PASTNUMBEROFCLAIMS in ('2to4', 'morethan4')))
where rnk <= 5
order by percent_fraud desc;
```

Script Output × Query Result ×

SQL | All Rows Fetched: 5 in 0.064 seconds

	POLICYNUMBER	PERCENT_FRAUD	RNK
1	654	61.87	1
2	11068	57.37	2
3	7435	55.47	3
4	3599	55.4	4
5	14877	55.37	5

Automated Monthly "Application"

Create
View CLAIMS2_30
As
Select * from CLAIMS2
Where mydate > SYSDATE – 30

Time measure: set timing on;

ORACLE

Neural Net Classification Demo [New in 18.1]

```
create table NN_SET (setting_name varchar2(30), setting_value varchar2(4000));
insert into NN_SET values ('ALGO_NAME','ALGO_NEURAL_NETWORK');
insert into NN_SET values ('NNET_NODES_PER_LAYER', '50');
insert into NN_SET values ('NNET_HIDDEN_LAYERS', '2');
insert into NN_SET values ('NNET_ITERATIONS', '300');
insert into NN_SET values ('NNET_TOLERANCE', '0.0001');
insert into NN_SET values ('ODMS_RANDOM_SEED', '15');
commit;
```

```
begin
dbms_data_mining.create_model('NNMODEL', dbms_data_mining.classification,
  'mining_data_build_v', 'ID', 'affinity_card', 'NN_SET');
end;
/
```

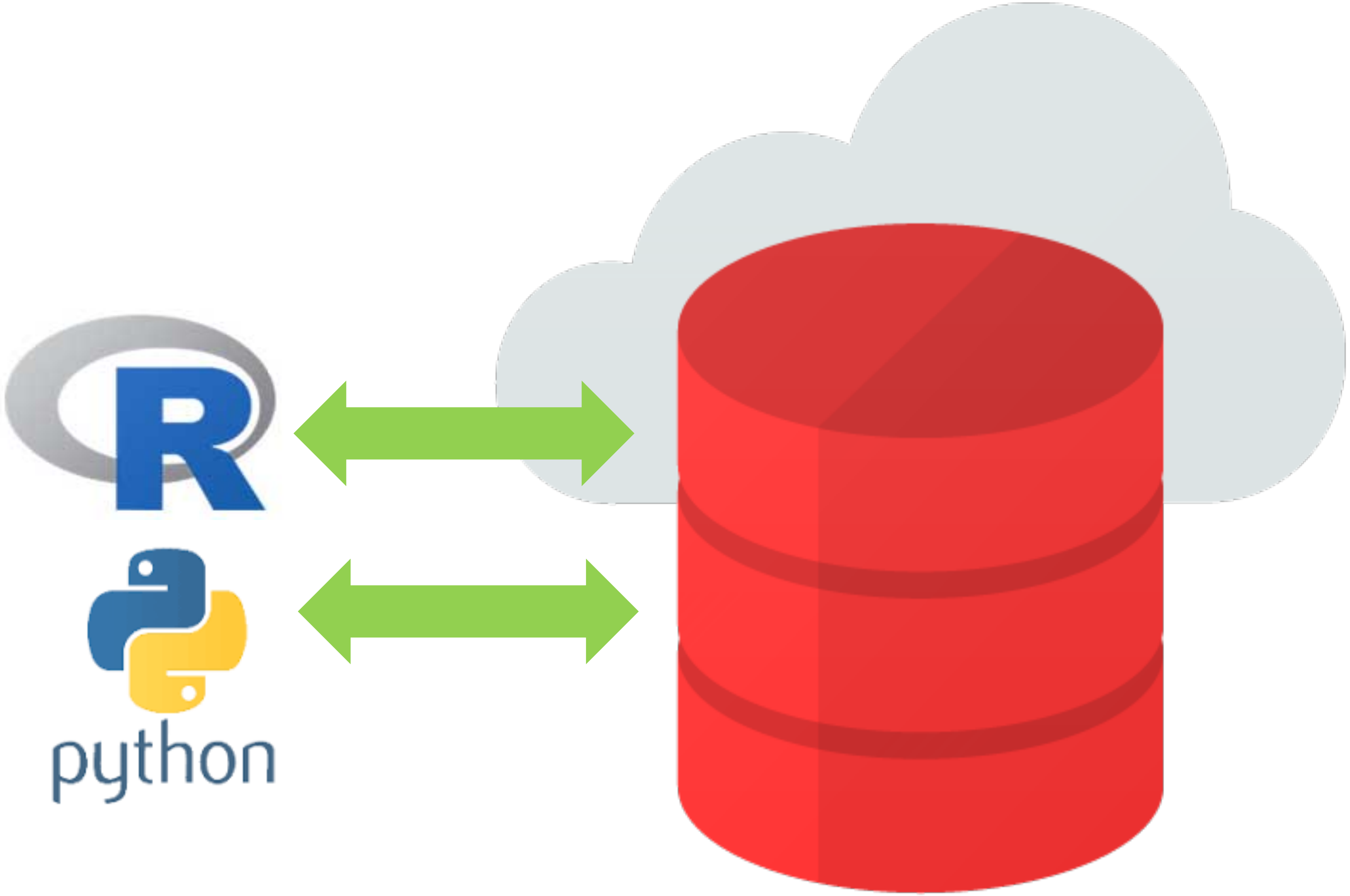
```
-- Scoring
SELECT id, affinity_card, prediction(tmnn2_01 using *) pred,
prediction_probability(nnmodel using *) prob
FROM mining_data_build_v
WHERE id between 101501 and 101509 order by id;

        ID AFFINITY_CARD       PRED       PROB
---------- ------------- ---------- ----------
    101501             0          0     .68547
    101502             0          0     .80765
    101503             0          0    1.00000
    101504             1          0     .60190
    101505             1          1     .53430
    101506             0          0     .80574
```

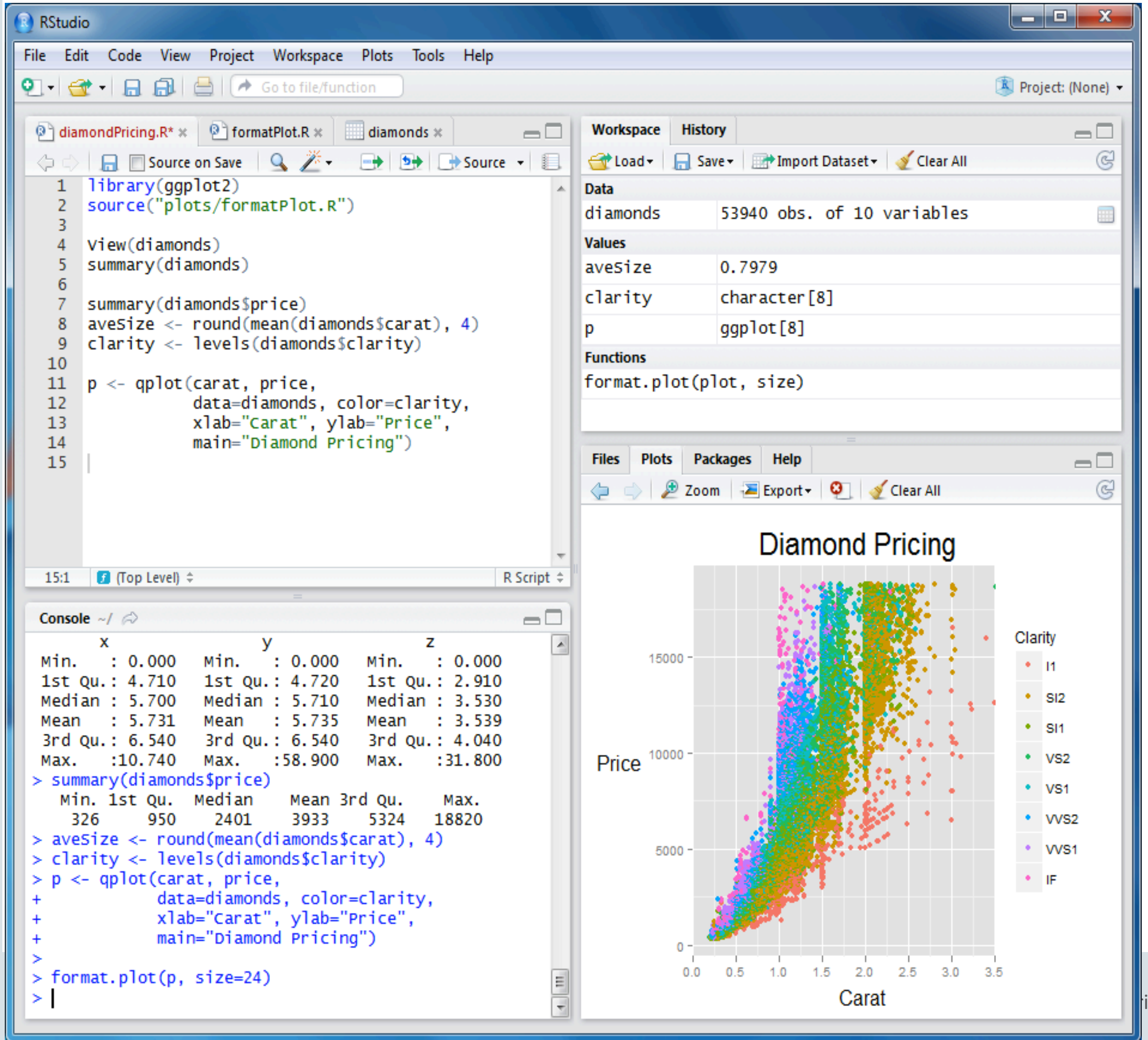
```
-- Accuracy
SELECT SUM(correct)/COUNT(*) AS accuracy
FROM (SELECT DECODE(affinity_card,
PREDICTION(nnmodel USING *), 1, 0) AS correct
FROM mining_data_build_v);

ACCURACY
--------
    .757
```

```
-- Weights
SELECT layer, idx_from,
       attribute_name as attr_name, idx_to, weight
FROM DM$VAnnmmodel
ORDER BY layer, idx_to, idx_from ASC nulls FIRST;

LAYER IDX_FROM ATTR_NAME      IDX_TO       WEIGHT
----- -------- -------------- ---------- -------------
    0        0 AGE                     0   -1.25821459
    0        1 ANNUAL_INCOME           0     .15279913
    0        2 YRS_RESIDENCE           0    1.42988670
    0                                  1   16.38790321
    0        0 AGE                     1   13.97867584
    0        1 ANNUAL_INCOME           1    -.82504904
    0        2 YRS_RESIDENCE           1     .01078100
    0                                  2     .42324463
    0        0 AGE                     2  -27.43167877
    0        1 ANNUAL_INCOME           2    -.20815386
    0        2 YRS_RESIDENCE           2  -33.85304642
```

ORACLE

Machine Learning using external database algorithms





In **data science**, the open source language R is popular for **statistical analysis**, **charting** and **machine learning**.

R is a very powerful **scripting** language

R has an interface for **DB drivers**

Using Oracle R with the Oracle Database

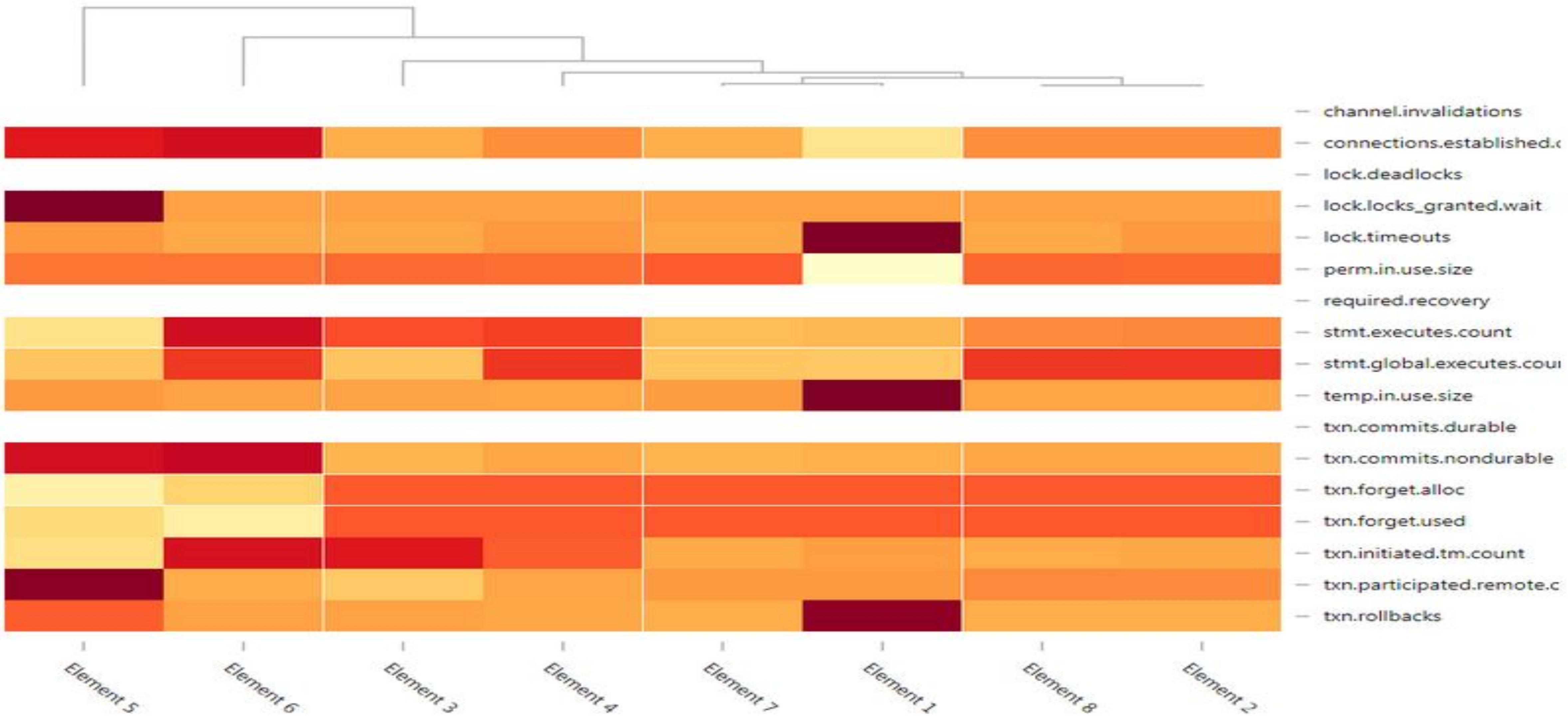
- DB Driver for Oracle
- Written in OCI

Using Oracle R with TimesTen Scaleout

- DB Driver for Oracle
- Written in OCI

- Oracle TimesTen Scaleout **supports OCI**
- Oracle Client 11.2.0.4+ **supports both** Oracle + TimesTen
- The **ROracle** OCI DB **driver works** with TimesTen Scaleout **unchanged**
  - Just change the **connect string** (tnsnames.ora or easy connect)

Using R Oracle to find hot spots in a clustered RDBMS



ORACLE

Connecting to Oracle or TimesTen Scaleout with R Oracle

```
# load the ROracle package
# load reshape2 to pivot tables
# load d3heatmap to create an interactive heat map visualization
library ("ROracle")
library ("reshape2")
library ("d3heatmap")


# Use the Oracle OCI SQL driver
drv <- dbDriver ("Oracle")

ttdb <- dbConnect (drv,
                   username = "myUsername"
                   password = "myPassword"
                   dbname = "myDB"
```

```
# query the GV$MONITOR and GV$TTSTATS_ELEMENT_METRICS global views for
# current performance metrics on each node
heat.kpi <- dbGetQuery (ttdb, "
  WITH BEGIN_DATE AS (
    SELECT
      MAX (TO_DATE (TIME_OF_1ST_CONNECT,
        'DY MON DD HH24:MI:SS YYYY')) AS BEGIN_DATE
    FROM GV$MONITOR
  ),
  MONITOR_STATS AS (
    SELECT
      'Element ' || ELEMENTID AS ELEMENTID,
      'perm.in.use.size' AS METRIC_NAME,
      PERM_IN_USE_SIZE  AS METRIC_VALE
    FROM GV$MONITOR
    UNION
    SELECT
      'Element ' || ELEMENTID AS ELEMENTID,
      'temp.in.use.size' AS METRIC_NAME,
      TEMP_IN_USE_SIZE AS METRIC_VALUE
    FROM GV$MONITOR
    UNION
    SELECT
      'Element ' || ELEMENTID AS ELEMENTID,
      'required.recovery' AS METRIC_NAME,
      REQUIRED_RECOVERY AS METRIC_VALUE
    FROM GV$MONITOR
  )
  SELECT /*+ TT_PartialResult (1) */
    'Element ' || ELEMENTID AS ELEMENTID,
    TRIM (METRIC_NAME) AS METRIC_NAME,
    MAX (METRIC_VALUE) - MIN (METRIC_VALUE) AS METRIC_VALUE
  FROM GV$TTSTATS_ELEMENT_METRICS
  WHERE COLLECTED_AT > (SELECT BEGIN_DATE FROM BEGIN_DATE) AND
    TRIM (METRIC_NAME) IN ('connections.established.count',
      'stmt.executes.count', 'stmt.global.executes.count',
      'lock.timeouts', 'lock.deadlocks', 'lock.locks.granted.wait',
      'txn.commits.durable', 'txn.commits.nondurable', 'txn.rollbacks',
      'channel.invalidations', 'txn.forget.used', 'txn.forget.alloc',
      'txn.initiated.tm.count', 'txn.participated.remote.count')
  GROUP BY ELEMENTID, TRIM (METRIC_NAME)
  UNION
  SELECT * FROM MONITOR_STATS
")
# disconnect from Velocity Scale
dbDisconnect (ttdb)
```

dbGetQuery function

Pass in a **SQL statement**

- WITH clause
- UNIONs
- String IN set
- GROUP BY
- UNION

Disconnect

Using R Oracle to create a heatmap

```
# pivot rows to columns
heat.kpi.pivot <- recast (heat.kpi, METRIC_NAME ~ ELEMENTID)

# convert the data frame to a matrix
heat.kpi.matrix <- as.matrix (heat.kpi.pivot)

# remove the METRIC_NAME column
heat.kpi.matrix <- heat.kpi.matrix[,-1]

# convert the matrix data to the numeric type
heat.kpi.matrix <- apply (heat.kpi.matrix, 2, as.numeric)

# set the row names
rownames (heat.kpi.matrix) <- heat.kpi.pivot$METRIC_NAME
```

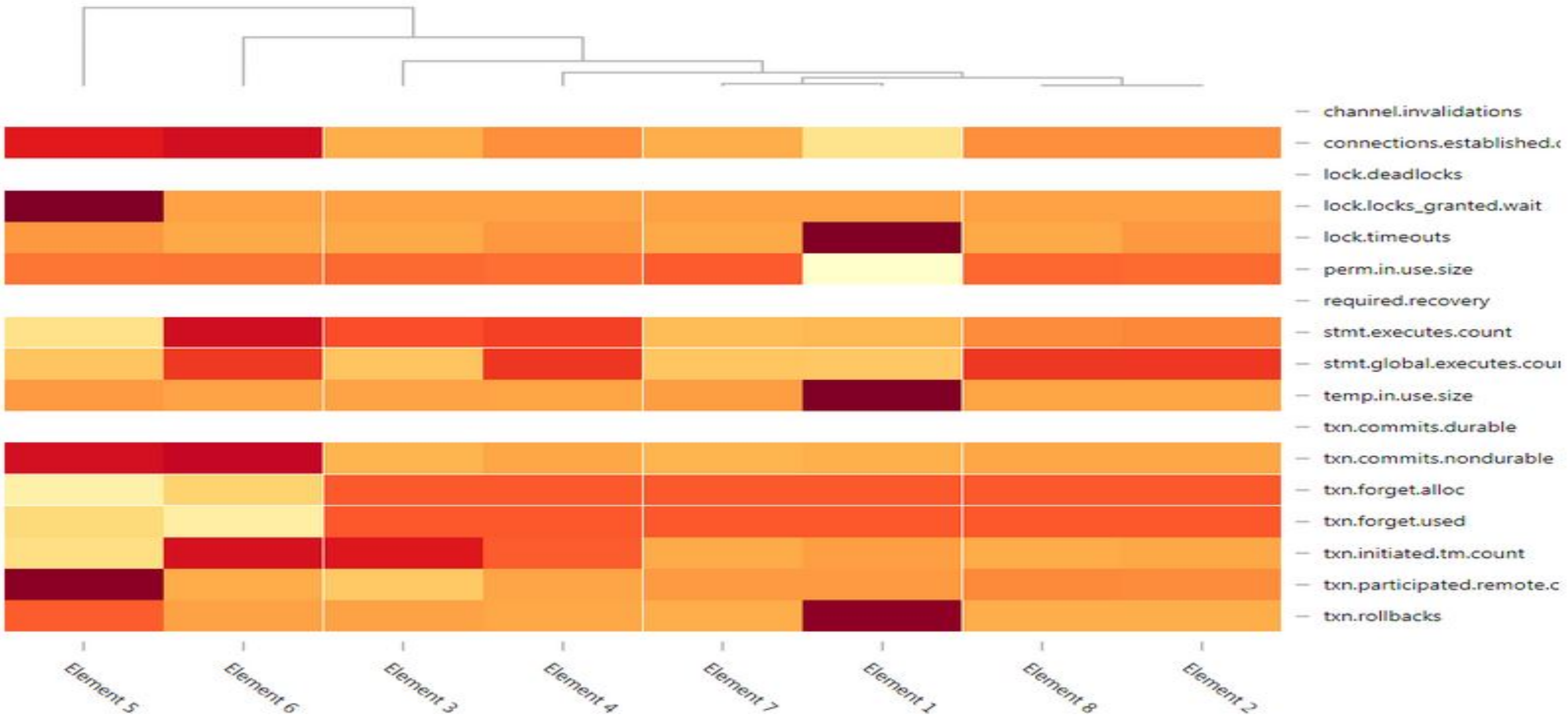
DB independent Code

Just R

```
# generate the heat map
d3heatmap (heat.kpi.matrix, scale="row",
           xaxis_font_size = "7pt", yaxis_font_size = "7pt",
           dendrogram = "column", color = "YlOrRd", show_grid=TRUE)
```

ORACLE

Using R Oracle to find hot spots in TimesTen Scaleout

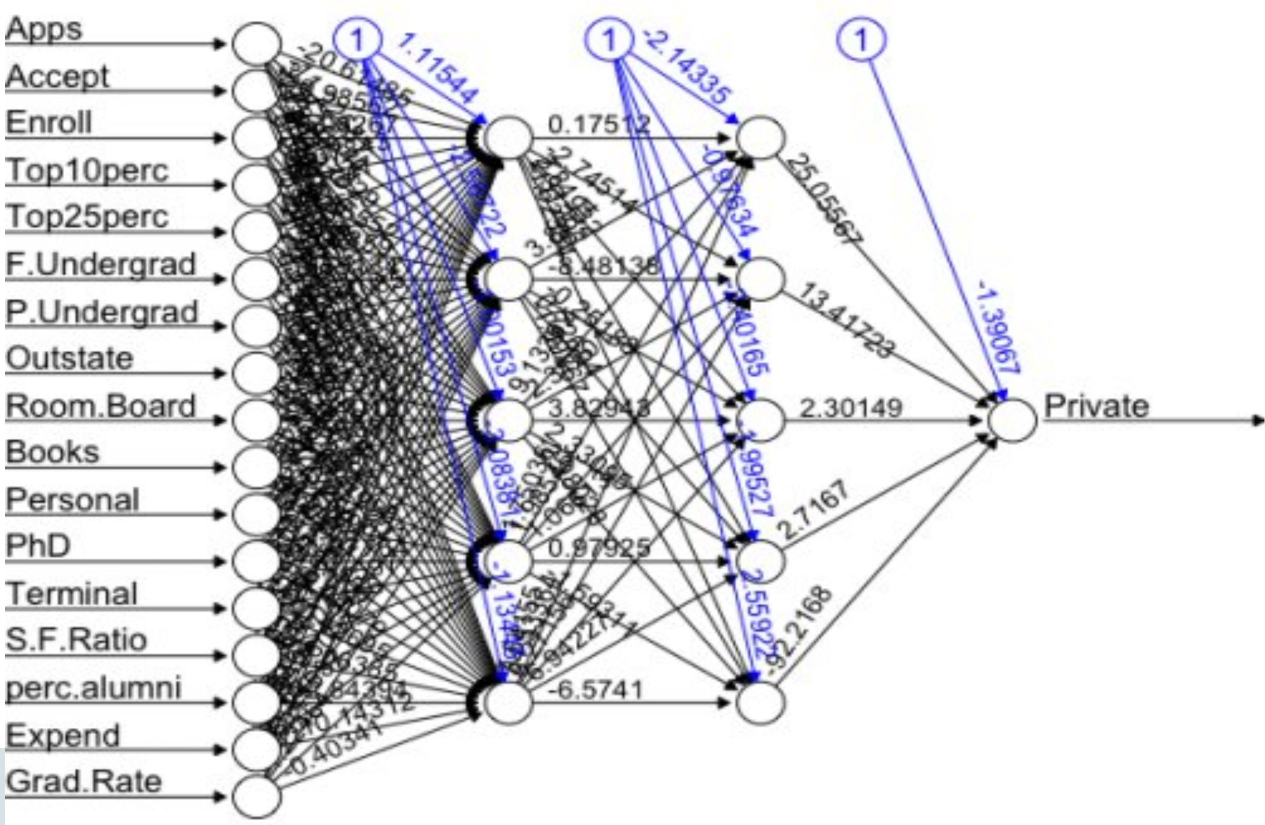


ORACLE

A simple Neural Net in R

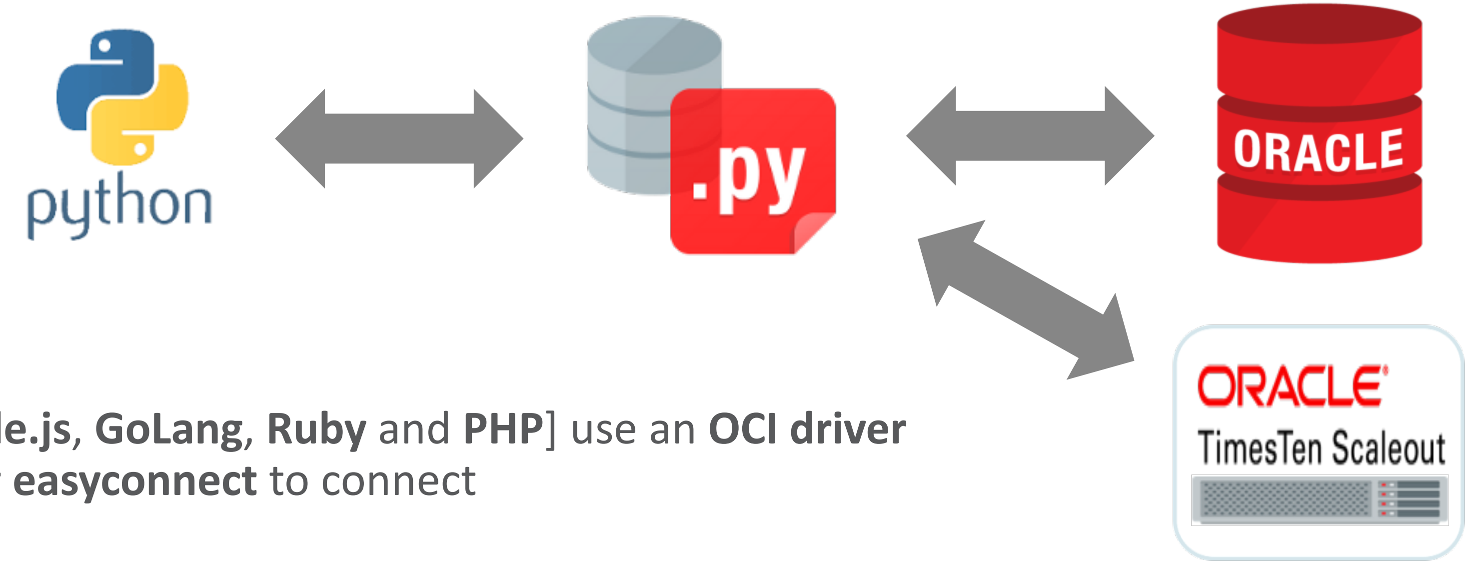
```
library(ISLR)
library(neuralnet)
library(caTools)

# Create Vector of Column Max and Min Values
maxs <- apply(College[,2:18], 2, max)
mins <- apply(College[,2:18], 2, min)

# Use scale() and convert the matrix to a data frame
scaled.data <- as.data.frame(scale(College[,2:18],
center = mins, scale = maxs - mins))

# Convert Private column from Yes/No to 1/0
Private = as.numeric(College$Private)-1
data = cbind(Private,scaled.data)

set.seed(101)
# Create Split (any column is fine)
split = sample.split(data$Private, SplitRatio = 0.70)

# Split based off of split Boolean Vector
train = subset(data, split == TRUE)
test = subset(data, split == FALSE)

feats <- names(scaled.data)
```

```
# Concatenate strings
f <- paste(feats,collapse=' + ')
f <- paste('Private ~',f)

f <- as.formula(f)

nn <- neuralnet(f,train,hidden=c(10,10,10),linear.output=FALSE)
predicted.nn.values <- compute(nn,test[2:18])

print(head(predicted.nn.values$net.result))
plot(nn)
```



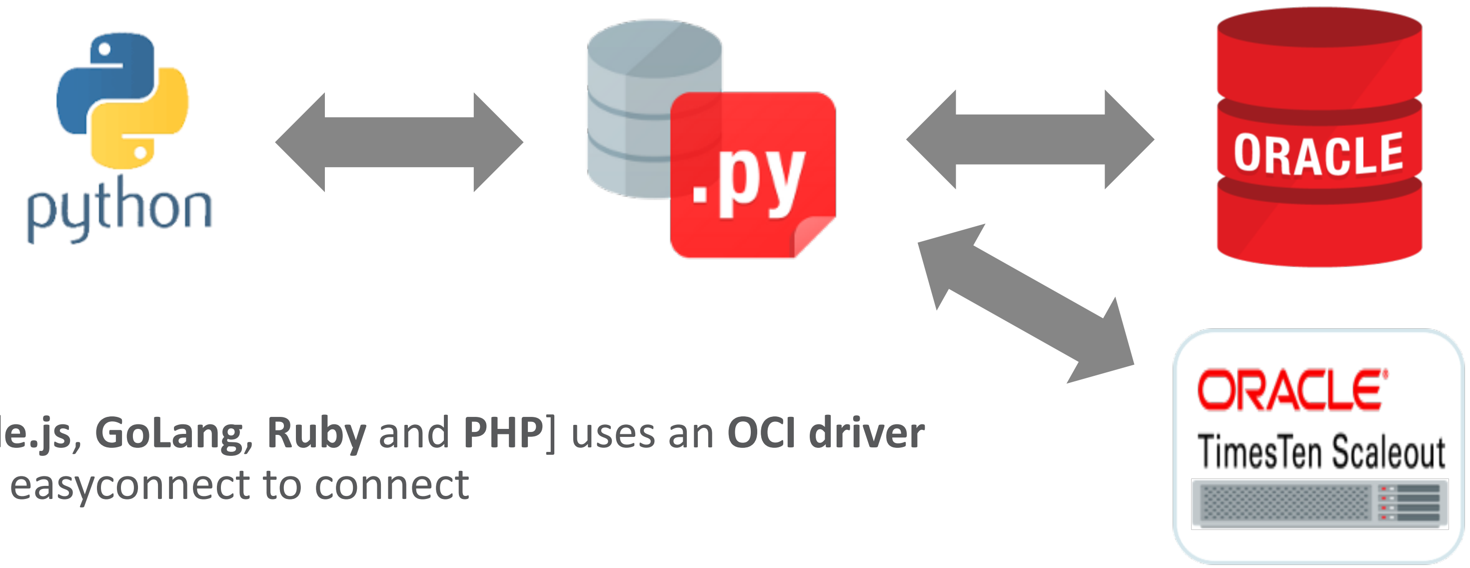
Using Oracle cx_Python with Oracle DB

Using Oracle cx_Python with TimesTen Scaleout



Python [and **Node.js**, **GoLang**, **Ruby** and **PHP**] use an **OCI driver**
Use **tnsnames** or **easyconnect** to connect

Using Oracle cx_Python with TimesTen Scaleout



Python [and **Node.js**, **GoLang**, **Ruby** and **PHP**] uses an **OCI driver**
Use tnsnames or easyconnect to connect

tnsnames.ora :

sampledb_1122 =(DESCRIPTION=(CONNECT_DATA = (SERVICE_NAME = **sampledb_1122**)(SERVER = **timesten_direct**)))
sampledbCS_1122 =(DESCRIPTION=(CONNECT_DATA = (SERVICE_NAME = **sampledbCS_1122**)(SERVER = **timesten_client**)))

TimesTen ODBC DSN

Client/Server or Direct Linked

Connecting to Oracle or TimesTen Scaleout with cx_Oracle

```
from __future__ import print_function

import cx_Oracle

connection = cx_Oracle.connect("tpch", "tpch", "exampledb")

cursor = connection.cursor()

cursor.execute("""
    SELECT N_NATIONKEY, N_NAME, N_REGIONKEY, N_COMMENT
    FROM nation
    where N_REGIONKEY = :rk""",
    rk = 1)

for N_NATIONKEY, N_NAME, N_REGIONKEY, N_COMMENT in cursor:
    print(N_NATIONKEY, N_NAME, N_REGIONKEY, N_COMMENT)

cursor.close
connection.close
```

ORACLE

A simple Neural Net in Python using Tensorflow – **Part 1**

```
"""Convolutional Neural Network Estimator for MNIST, built with tf.layers."""

from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import numpy as np
import tensorflow as tf

tf.logging.set_verbosity(tf.logging.INFO)

def cnn_model_fn(features, labels, mode):
  """Model function for CNN."""
  # Input Layer
  # Reshape X to 4-D tensor: [batch_size, width, height, channels]
  # MNIST images are 28x28 pixels, and have one color channel
  input_layer = tf.reshape(features["x"], [-1, 28, 28, 1])
```

ORACLE

A simple Neural Net in Python using Tensorflow – **Part 8**

```
# Train the model
  train_input_fn = tf.estimator.inputs.numpy_input_fn(
      x={"x": train_data},
      y=train_labels,
      batch_size=100,
      num_epochs=None,
      shuffle=True)
  mnist_classifier.train(
      input_fn=train_input_fn,
      steps=20000,
      hooks=[logging_hook])

  # Evaluate the model and print results
  eval_input_fn = tf.estimator.inputs.numpy_input_fn(
      x={"x": eval_data},
      y=eval_labels,
      num_epochs=1,
      shuffle=False)
  eval_results = mnist_classifier.evaluate(input_fn=eval_input_fn)
  print(eval_results)

if __name__ == "__main__":
tf.app.run()
```

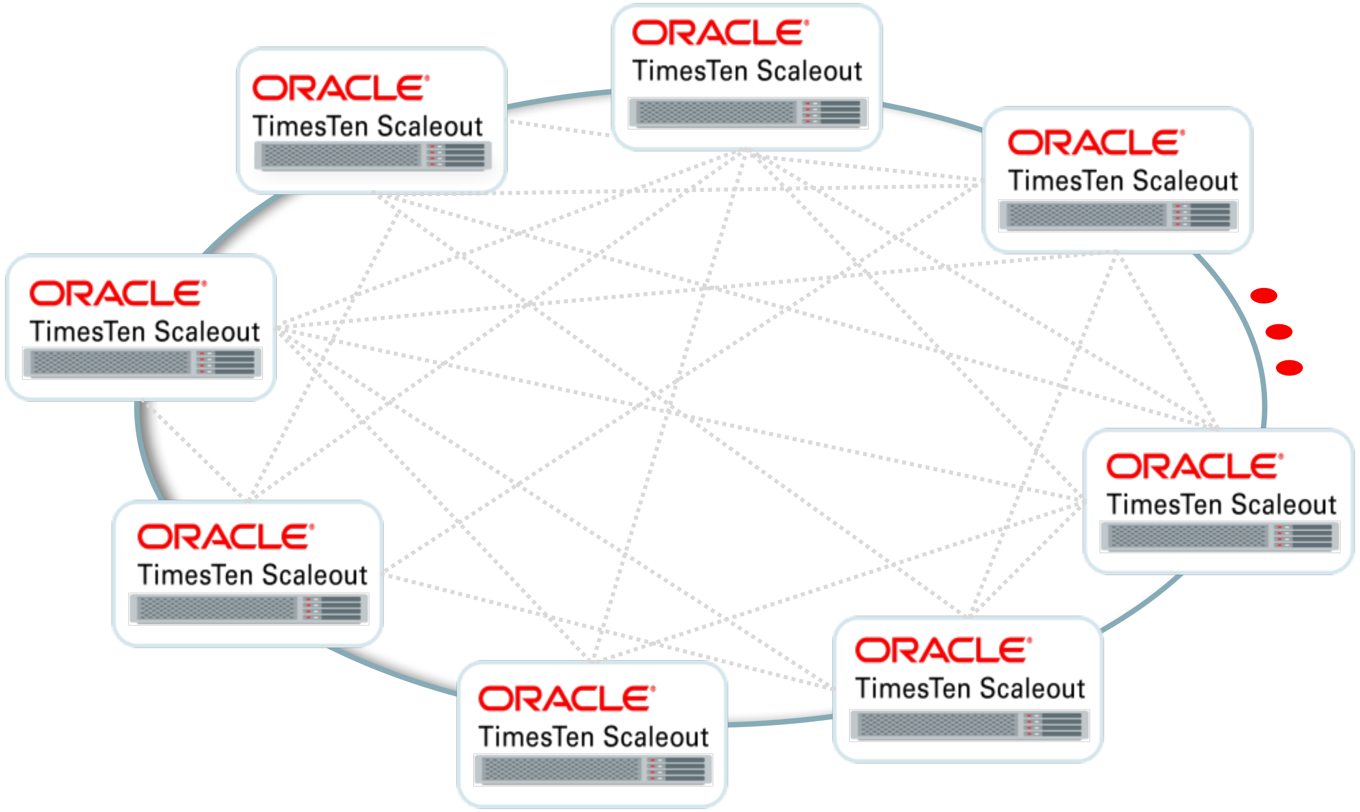
TimesTen Autonomous Database Cloud Service

World's Fastest OLTP Database

Provisioning

Automatic Patching

Automatic Upgrades



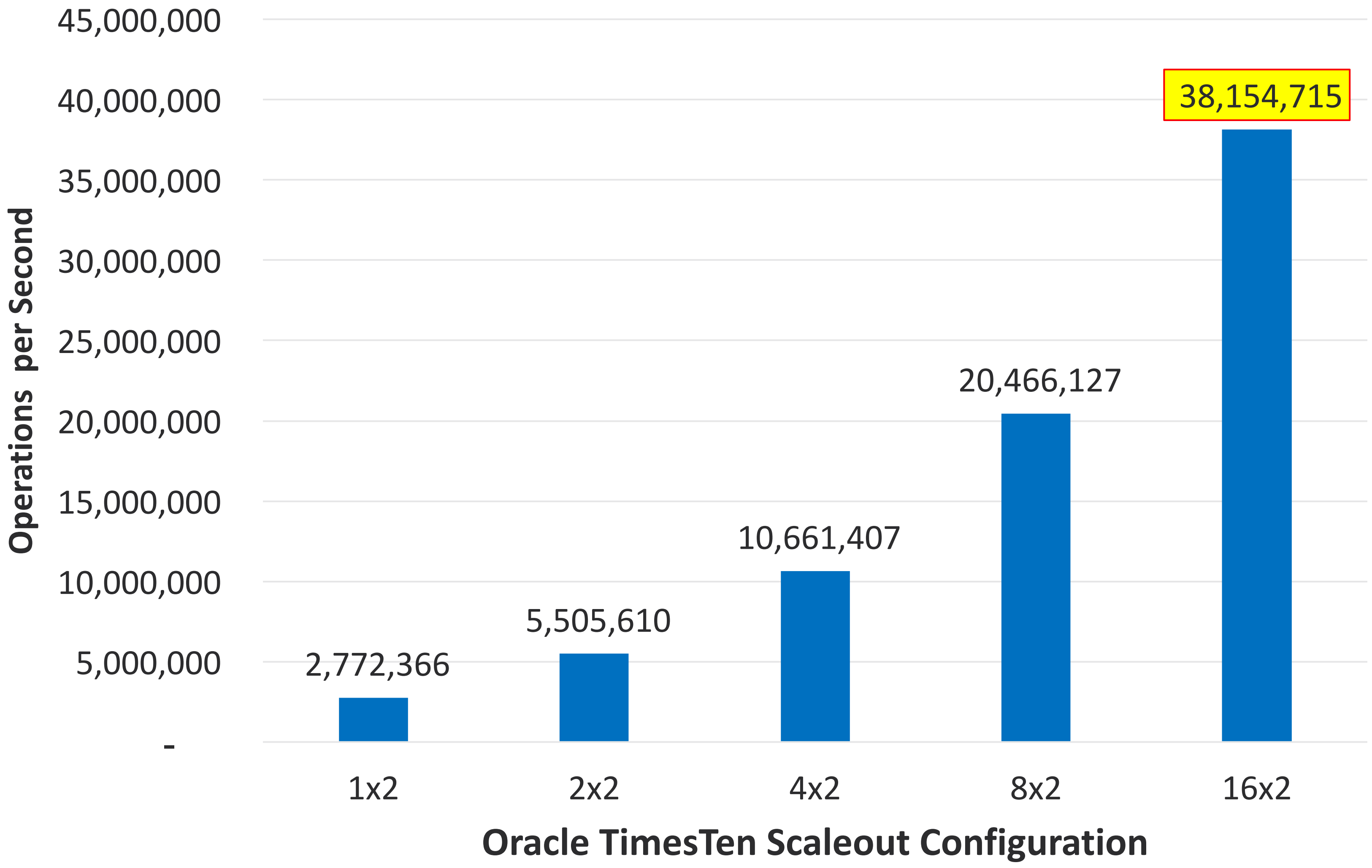
Automatic Backup

Self Tuning

Self Diagnostics

YCSB Workload B (95% Read 5% Update): **38 Million** Txns/Sec

Reminder: The best YCSB-B result found in our survey was **1.6 Million** Ops/Sec



YCSB version 0.15.0

- 1KB record (100-byte x 10 Fields)
- 100M records / Replica Set
- Uniform Distribution

TimesTen Scaleout

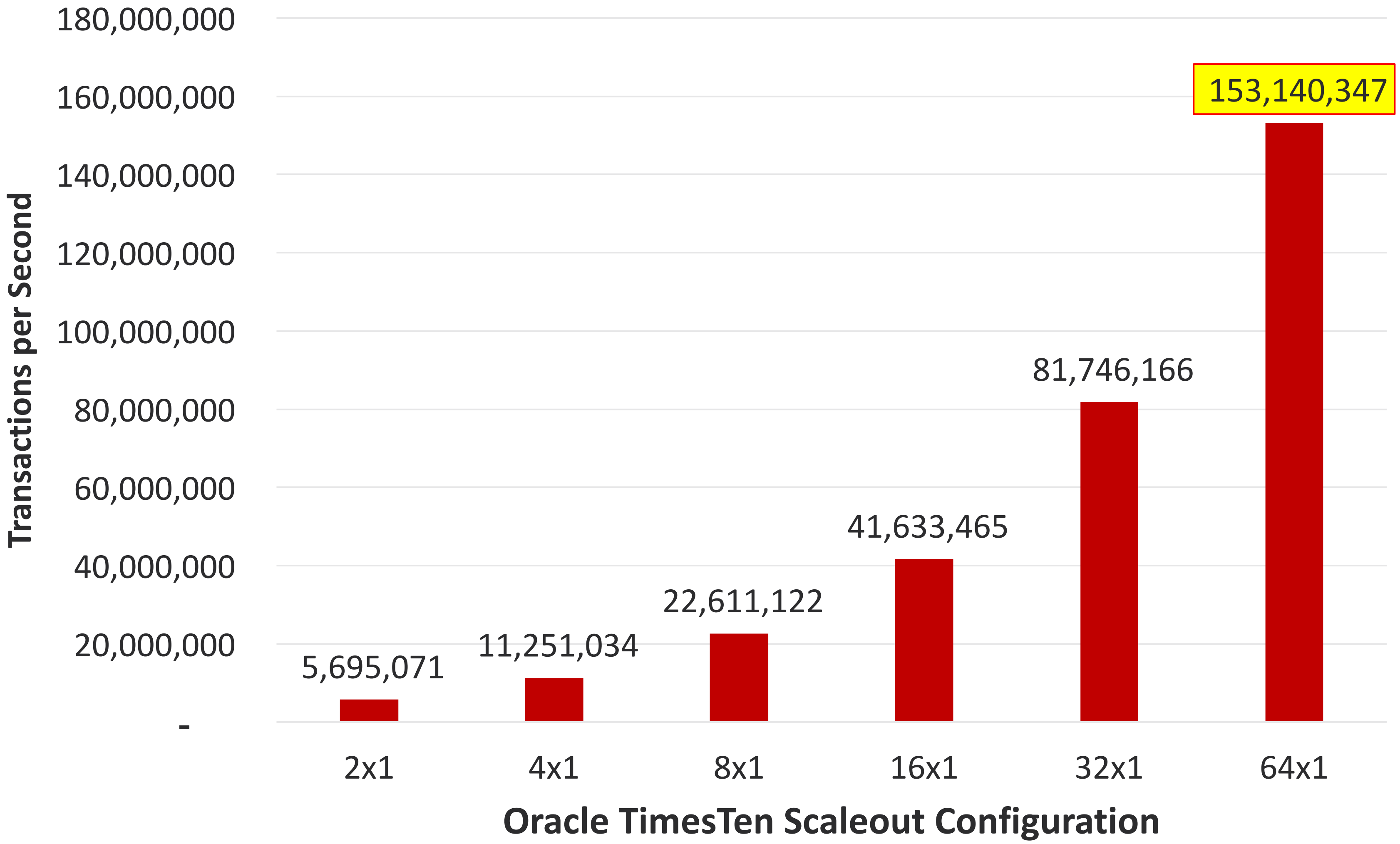
- 1 to 16 replica sets
- 2 synchronous replicas per replica set

Oracle Cloud Infrastructure

- 32 * BM.DenseIO2.52

ORACLE

TPTBM 80% Read 20% Update: **153 Million** Transactions/Sec



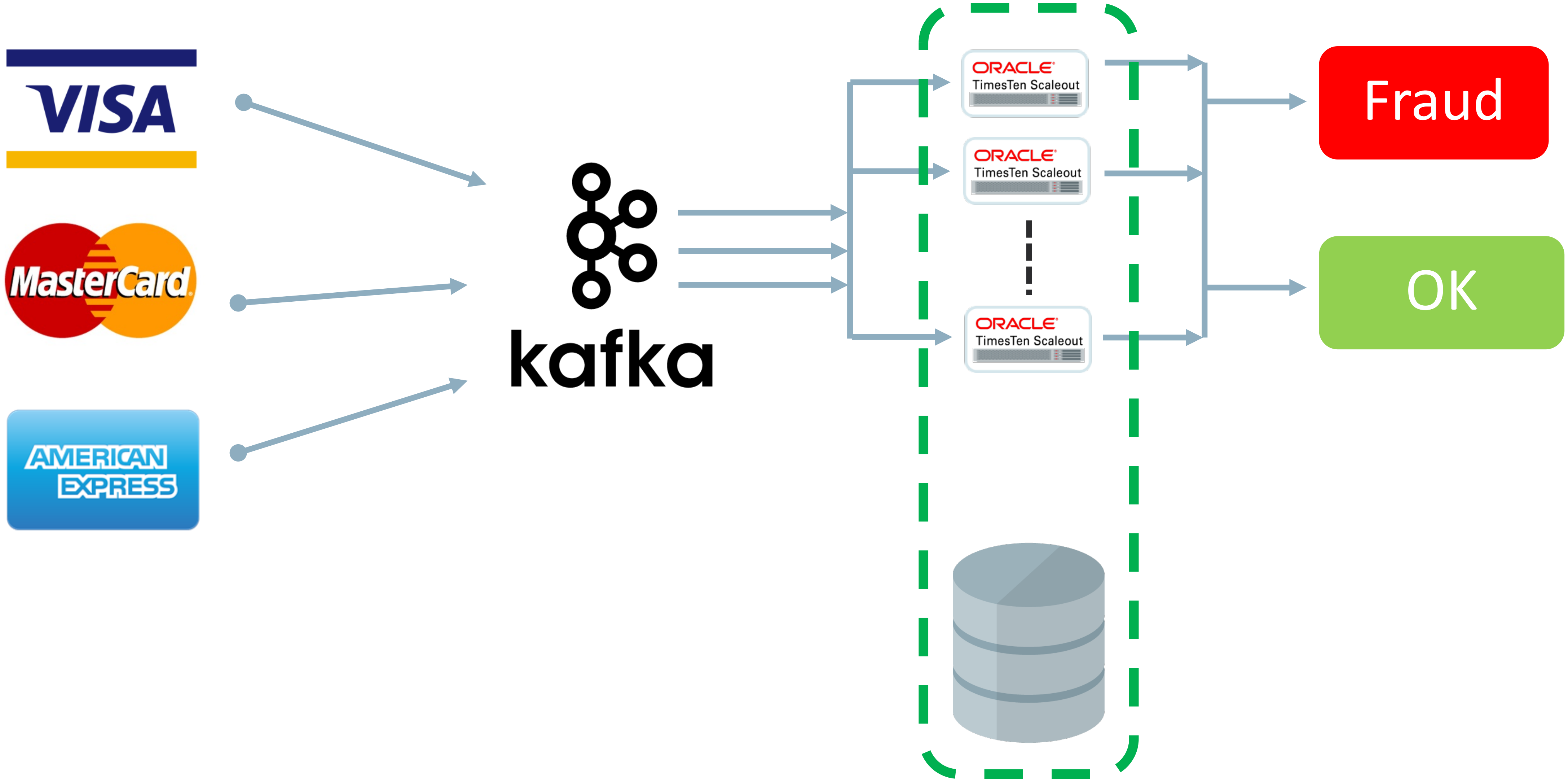
TPTBM Configuration
- 128-byte record
- 100M records / Replica Set
- Uniform Distribution

TimesTen Scaleout
- 1 to 64 replica sets
- 1 replica per replica set

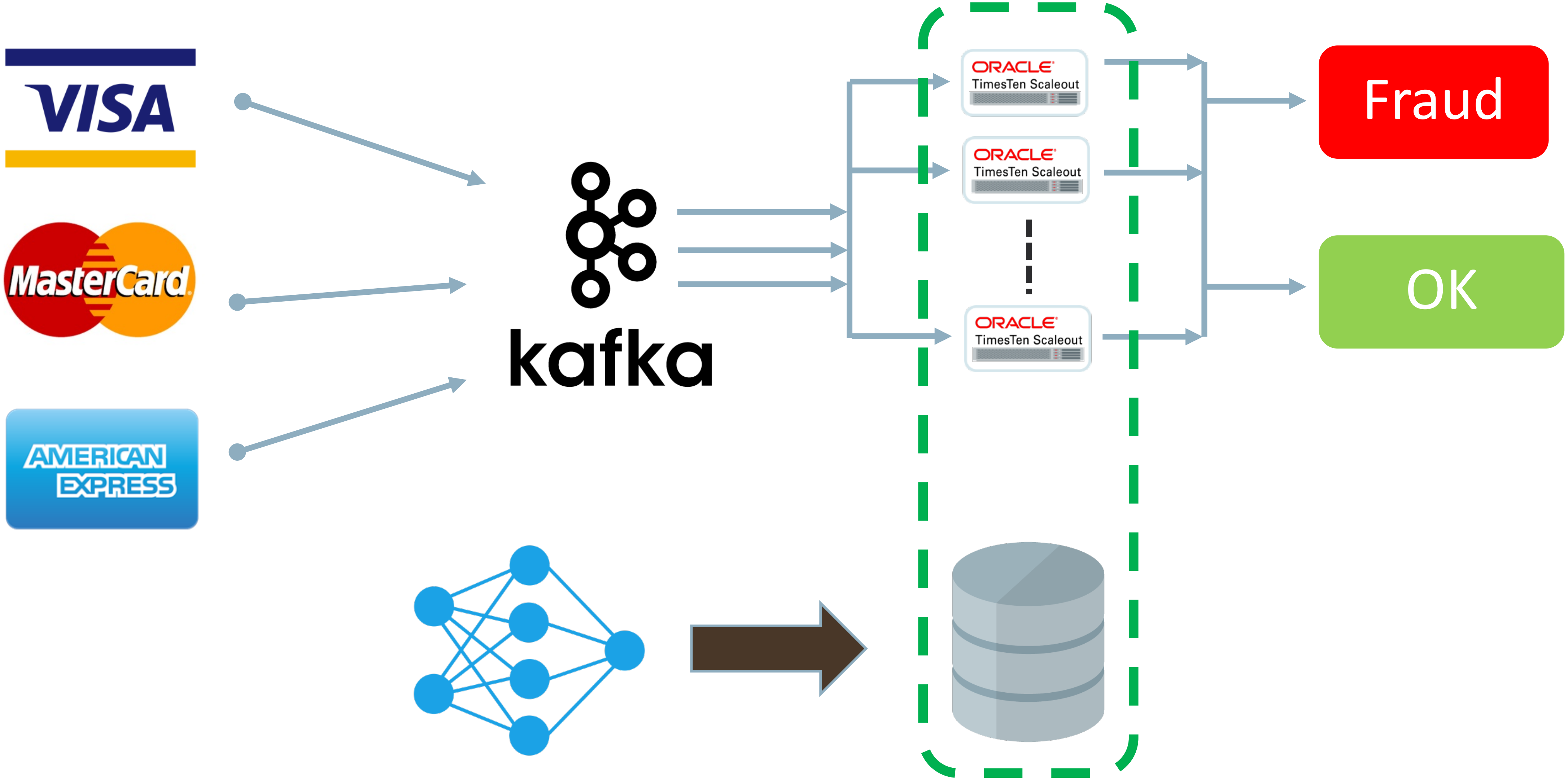
Oracle Cloud Infrastructure
- 32 * BM.DenseIO2.52
- Two TimesTen instances per compute node

ORACLE

An **IoT pipeline** for Real Time Credit Card Fraud Detection

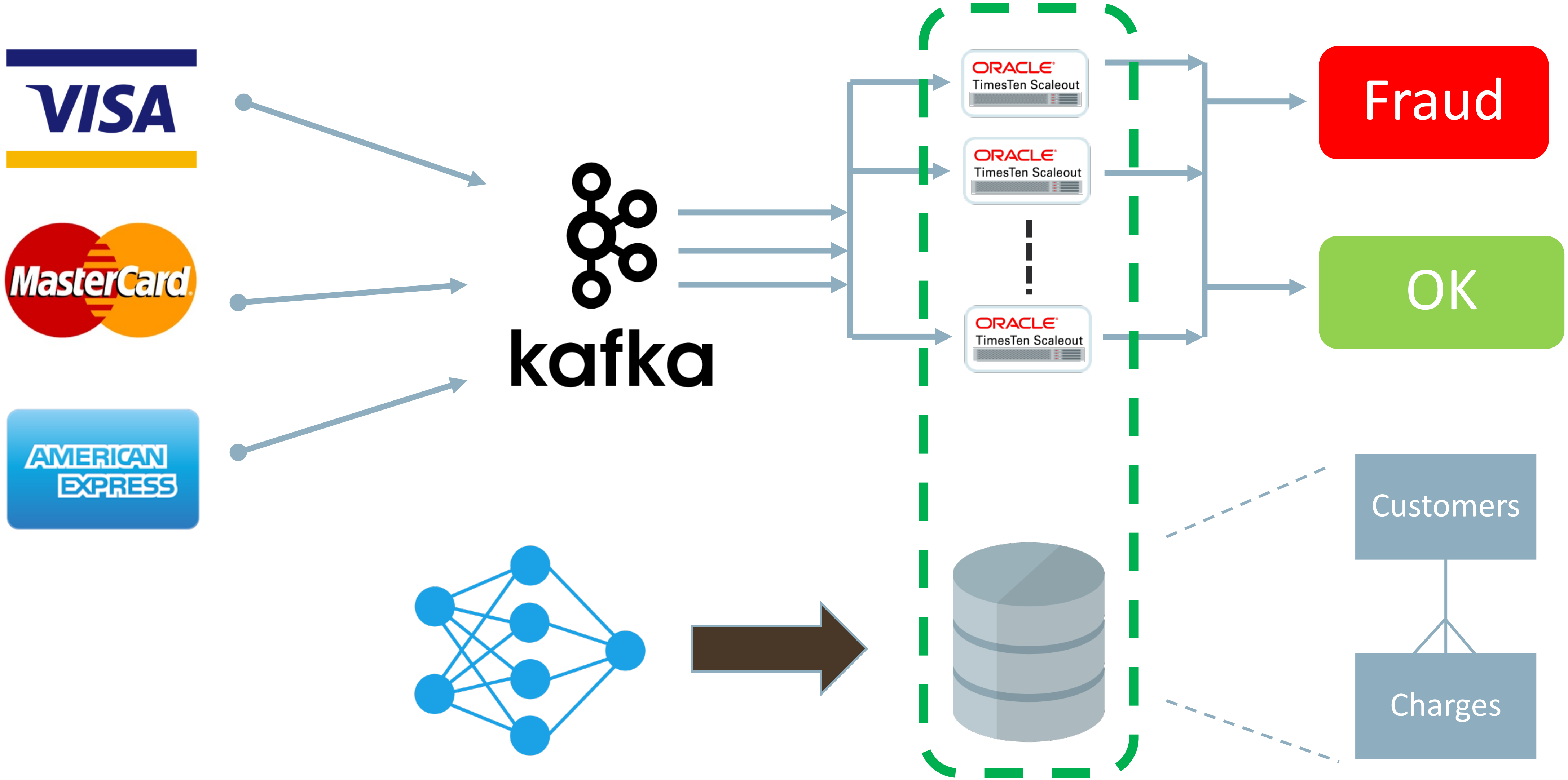


An **IoT pipeline** for Real Time Credit Card Fraud Detection

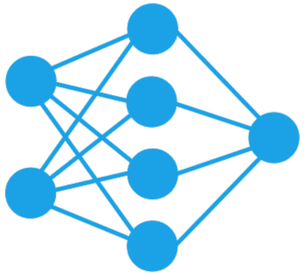


ORACLE

An **IoT pipeline** for Real Time Credit Card Fraud Detection



Machine Learning for Self Tuning OLTP DB

- What are the actionable outcomes for OTLP workloads?
- Which algorithms should be used?
- What mode should the DB run in?
- How effective is the ML?
- How efficient is the ML?

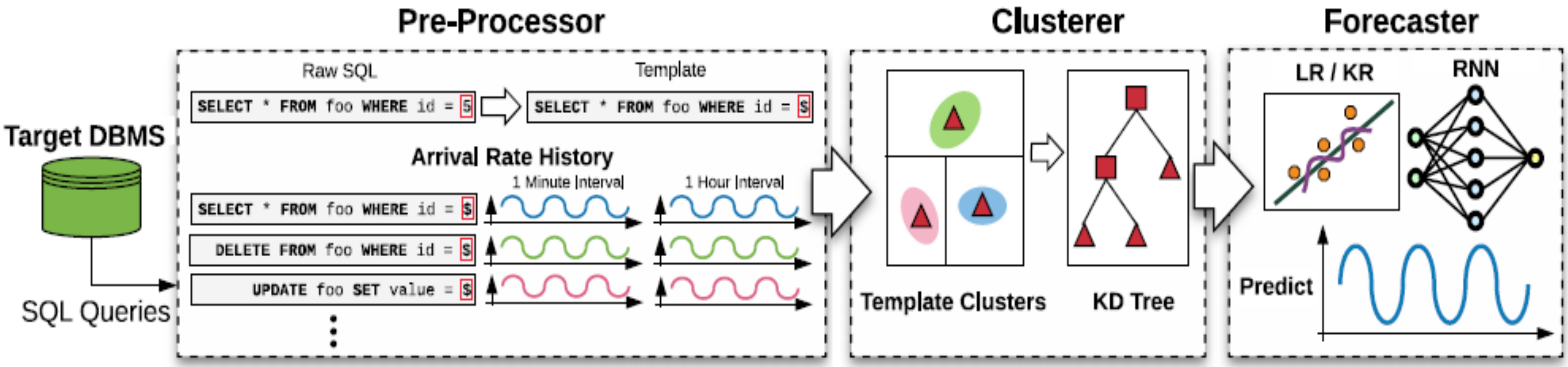
Which would you prefer?

- **Predictive Algorithm**
  - Makes recommendations on adding/removing indexes **once every hour**
  - Requires a **GPU cluster** for the TensorFlow Neural Nets
  - **Needs 25 days of *workload specific* training data, assumes future like past**
  - About 90% accurate

- **Reactive Algorithm**
  - Makes recommendations on adding/removing indexes **once every five minutes**
  - Requires **one [part time] background CPU thread**
  - **No training required**
  - About 90% accurate

ORACLE

Query based workload forecasting for self driving DBs (CMU)



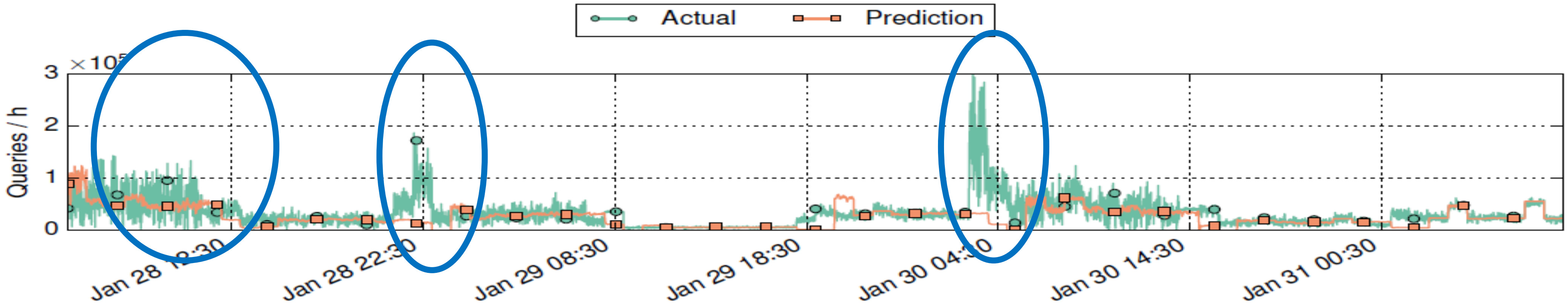


Figure 17: Prediction Results – Actual vs. predicted query arrival rates for a synthetic noisy workload.

TimesTen Self Tuning Database

- Do NOT try to predict the future
  - Instead 'tune frequently'
- Inputs
  - SQL workloads, schemas & stats
  - Only tune the 'slow SQL'
  - Nodes & error logs
- Outputs [advise or do]
  - Create/drop global/local indexes & MVs
  - Add/remove SQL hints
  - Change Distribution Keys
  - Add or remove nodes
  - Reboot or evict nodes

ORACLE

Summary

- Machine Learning is complex and covers many areas
- The statistical, data mining and ML features in the Oracle DB make many ML algorithms simple to implement
  - Just call SQL / PLSQL functions
- Python and R are the languages with the most ML algorithms
- Using Python and R for ML with Oracle + TimesTen is possible
- Self tuning TimesTen using ML is in development

ORACLE