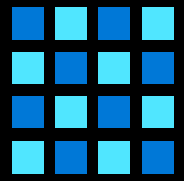
Agenda

- Big data fundamentals
  - Big data scenarios
  - Azure
  - Hadoop technology stack
  - Pig and Hive
  - HDInsight
  - Spark
  - Databricks
  - SQL Server 2019 big data clusters
- Understanding Machine Learning

Big data scenarios

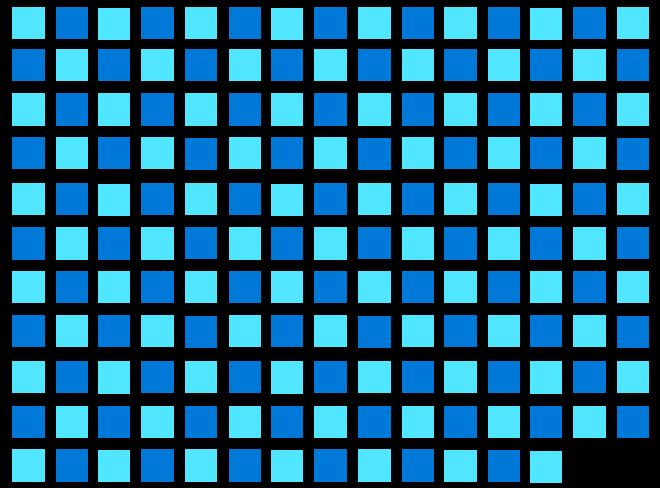
In 2016



16.1 ZBs

of data was generated

In 2025



163 ZBs

of data will be generated

*IDC White Paper, Data Age 2025: The Evolution of Data to Life-Critical

The 4 Vs of Big Data and AI

Velocity

No ETL pipelines!

Immediate access to data through data virtualization

Scale-out compute for faster queries

Spark for scale-out data prep, query, ML

Variety

Access all types of data – structured and unstructured in one system

Access data from many data systems using data virtualization

Veracity

Minimize data errors in ETL pipelines by working with the data from the source

Data is real time when you use data virtualization

Volume

Scalable storage in HDFS

Feed more data to your AI through data virtualization

Organizations that transform data into insights outperform the competition

What do these organizations do differently?

Integrate data without ETL

37% of leaders dynamically update data models

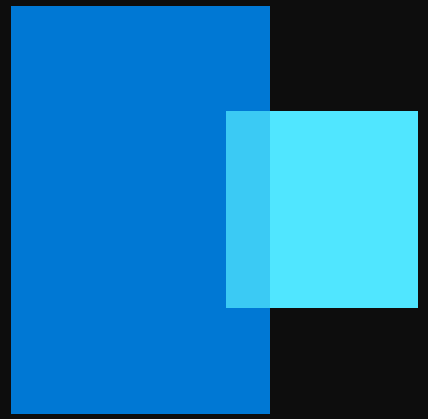
Combine data in a central data store

Leaders combine structured and unstructured data in a data lake 8X as often

Perform predictive analytics

74% of leaders use predictive models

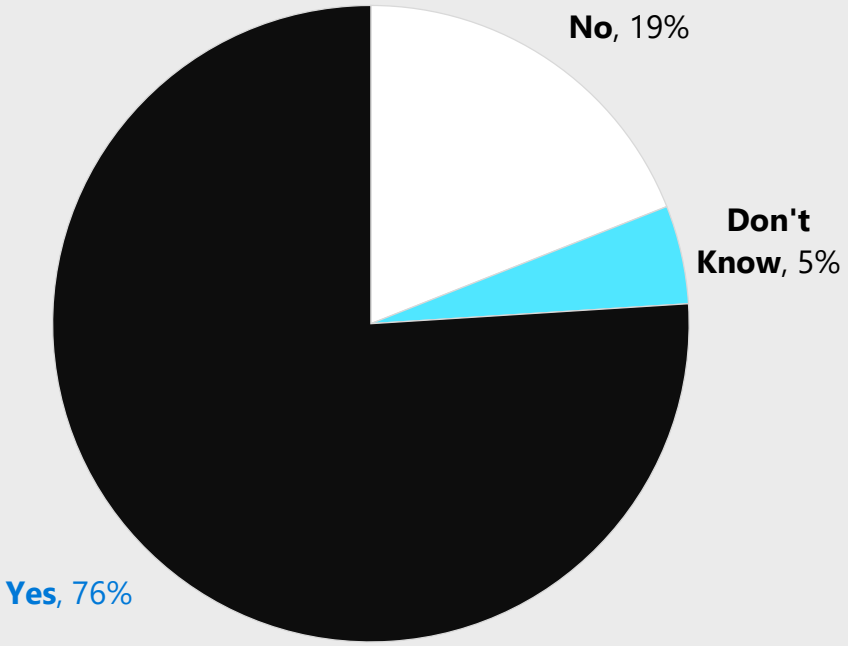
Source: Keystone Strategy interviews Oct 2015 - Mar 2016



Integrating all data

Data movement is a barrier to faster insights

- **Costs** → Duplicated storage costs; Engineering effort to build and maintain data pipelines
- **Speed** → Delays in integrate data before it can be used; Increased data latency
- **Security** → Increased attack surface area; Inconsistent security models
- **Quality** → Data quality issues can be created by ETL pipelines
- **Compliance** → Increased governance issues

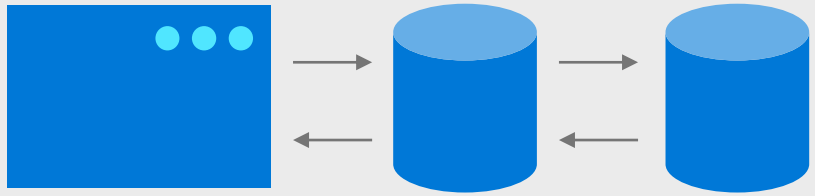


3/4 of respondents say that
untimely data has inhibited business opportunities

*IDC 3rd Platform Information Management Requirements Survey, Oct 2016

Data virtualization creates solutions

- **Costs** → Lower storage costs; Less dev time spent on integration
- **Speed** → Rapid iterations and prototypes; Timely data
- **Security** → Smaller attach surface area; Consistent security model
- **Quality** → Fresh and accurate data
- **Compliance** → Easier data governance



Data virtualization integrates data from disparate sources, locations and formats, **without replicating or moving the data**, to create a single "virtual" data fabric

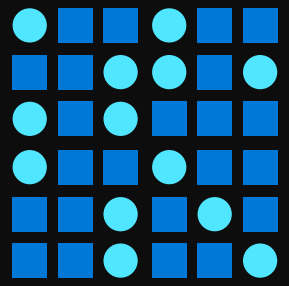
Managing all data

Big Data leads to big problems

Complex scale-out deployment

Time-consuming patching and upgrades

Cumbersome security management



Analyzing all data

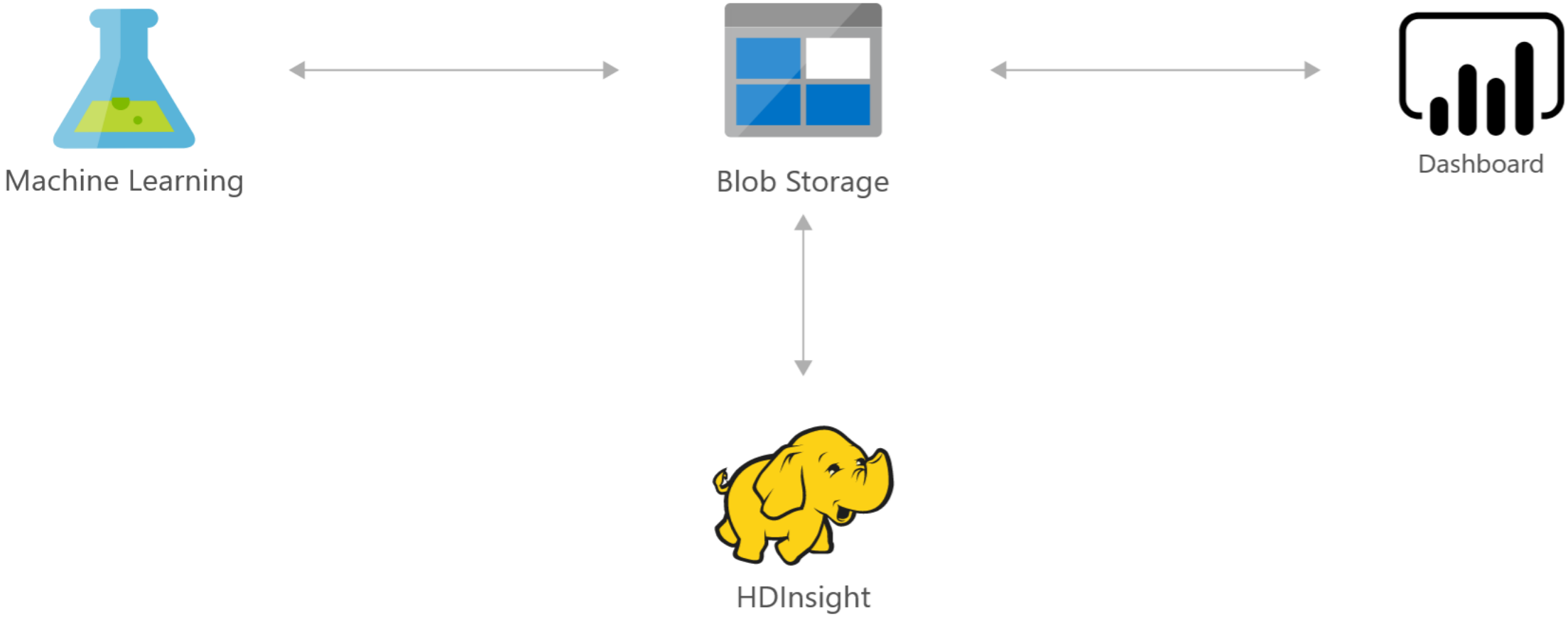
Accessing and analyzing data is difficult

Developers struggle to access insights from Big Data

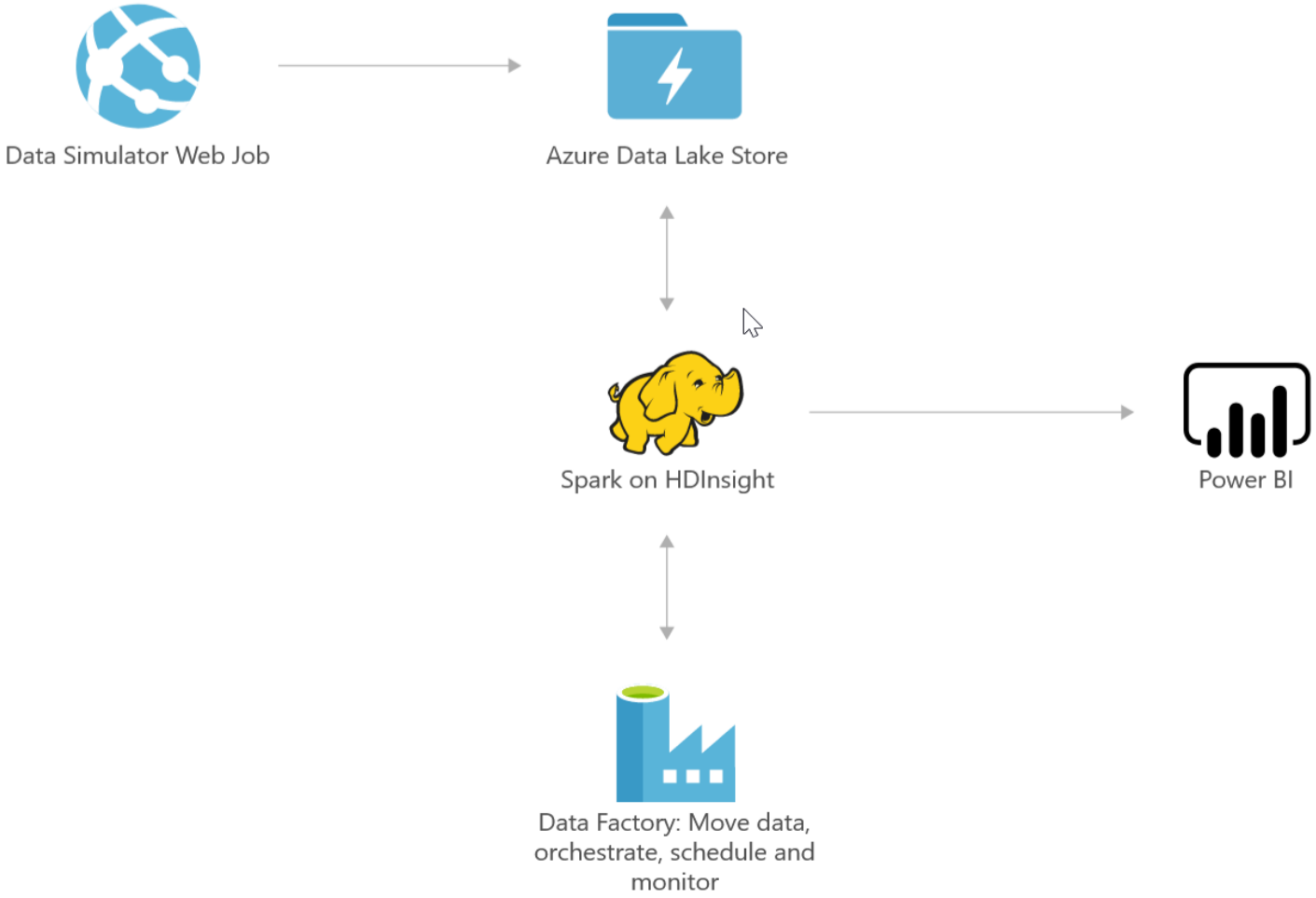
Data science is siloed from operational data

Lengthy time to train and operationalize models

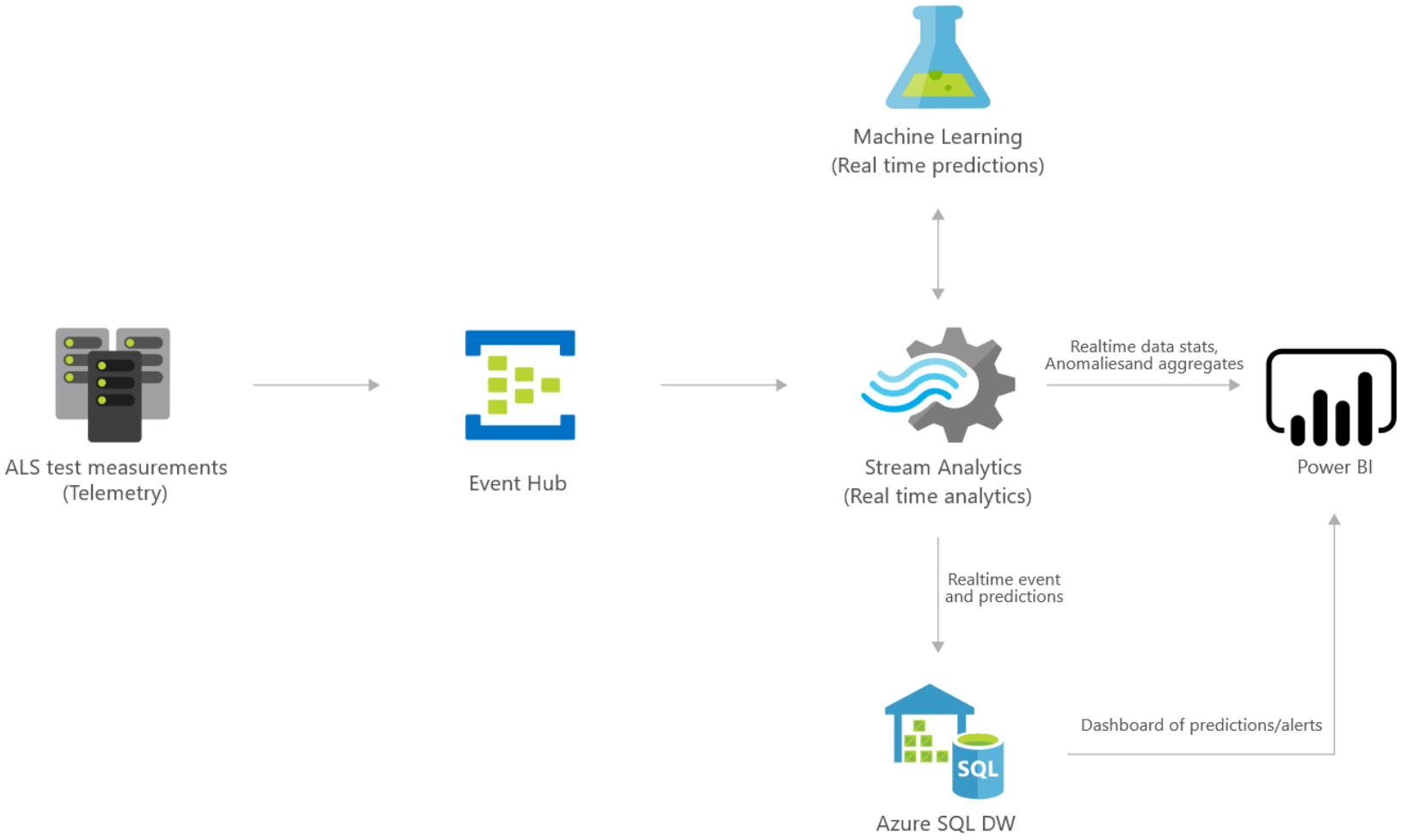
Predictive marketing campaigns with machine learning and Spark



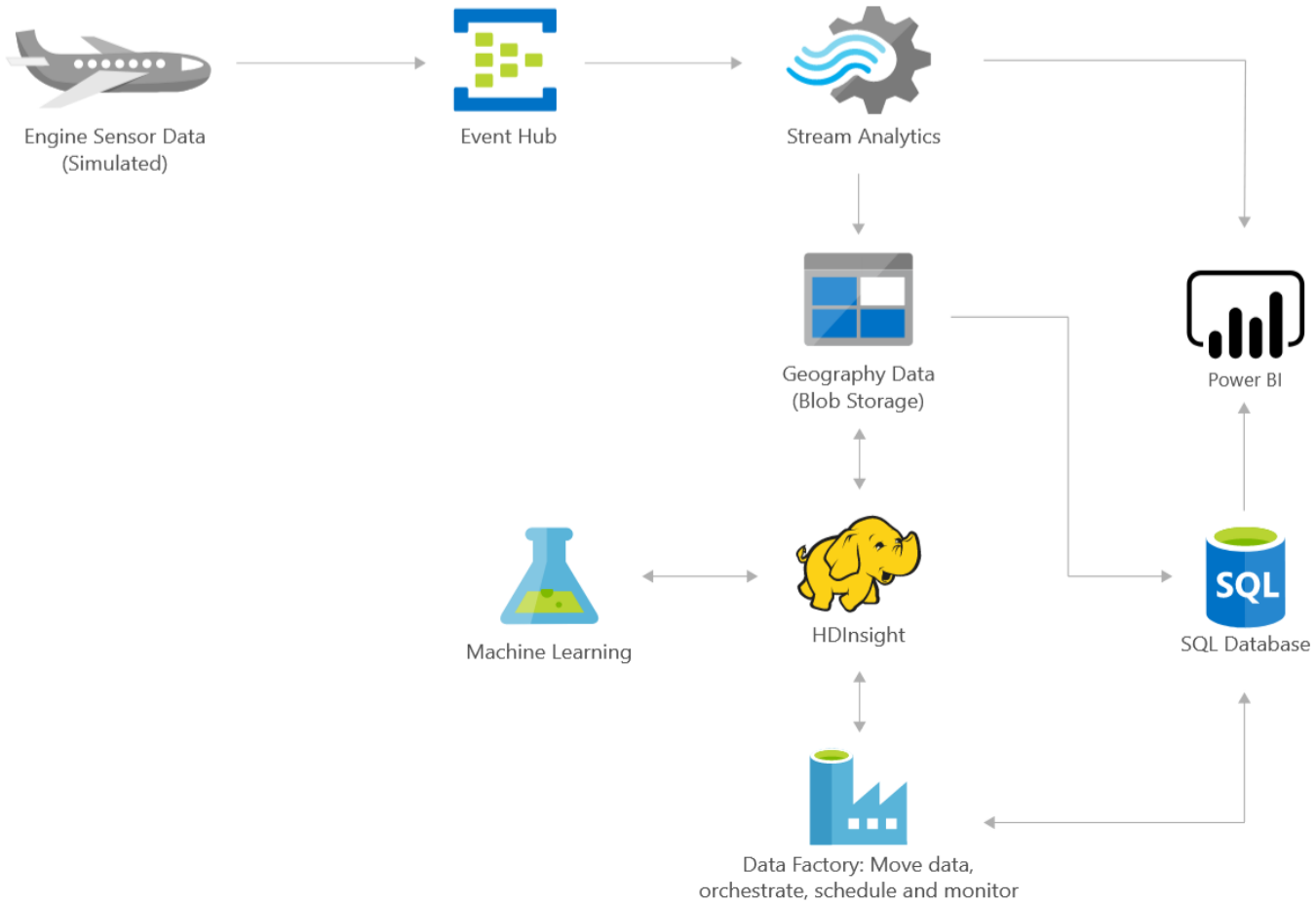
Demand forecasting and price optimization for marketing



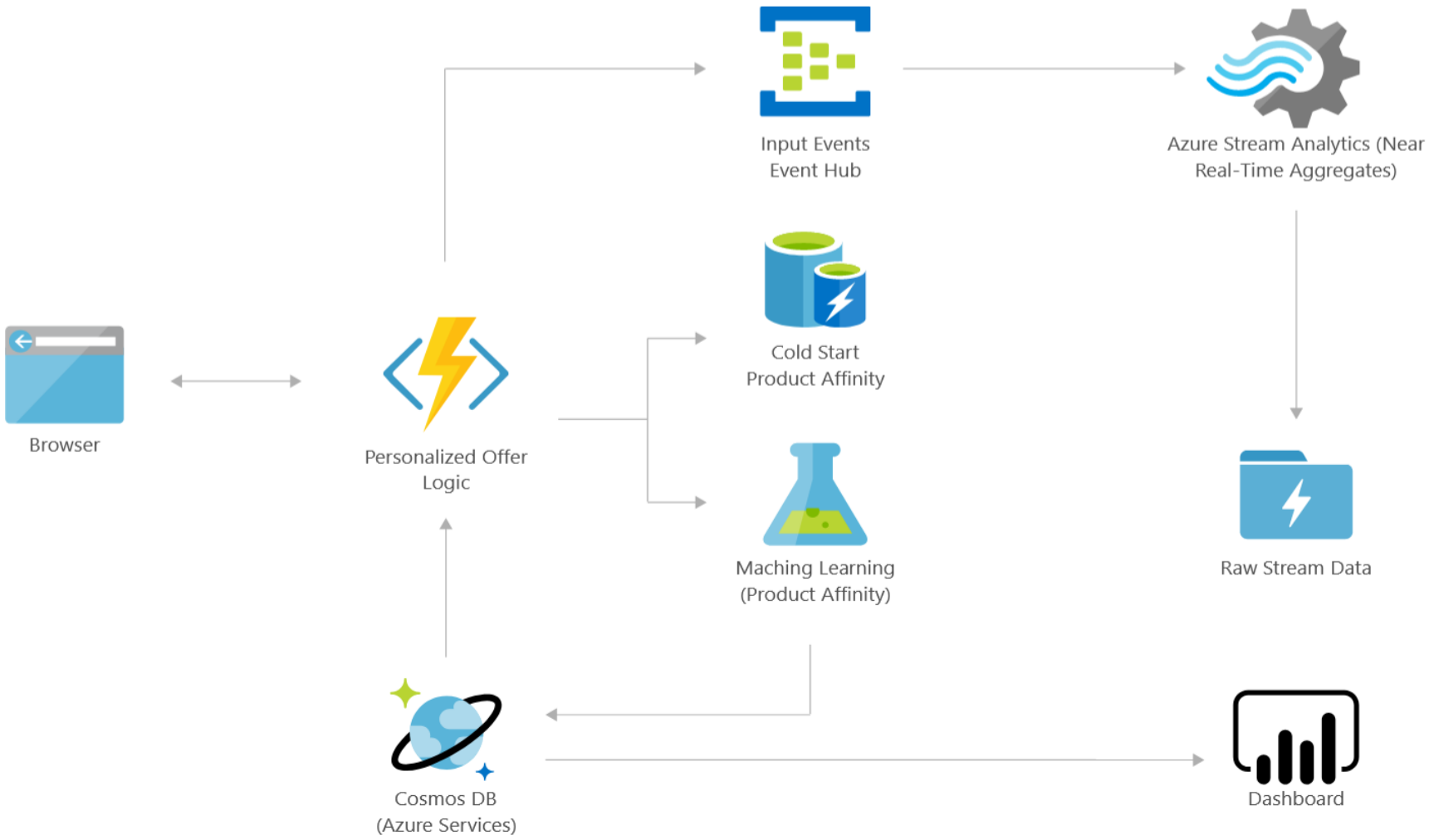
Defect prevention with predictive maintenance



Aircraft engine monitoring for predictive maintenance in aerospace



Personalized marketing solutions



Azure

Azure

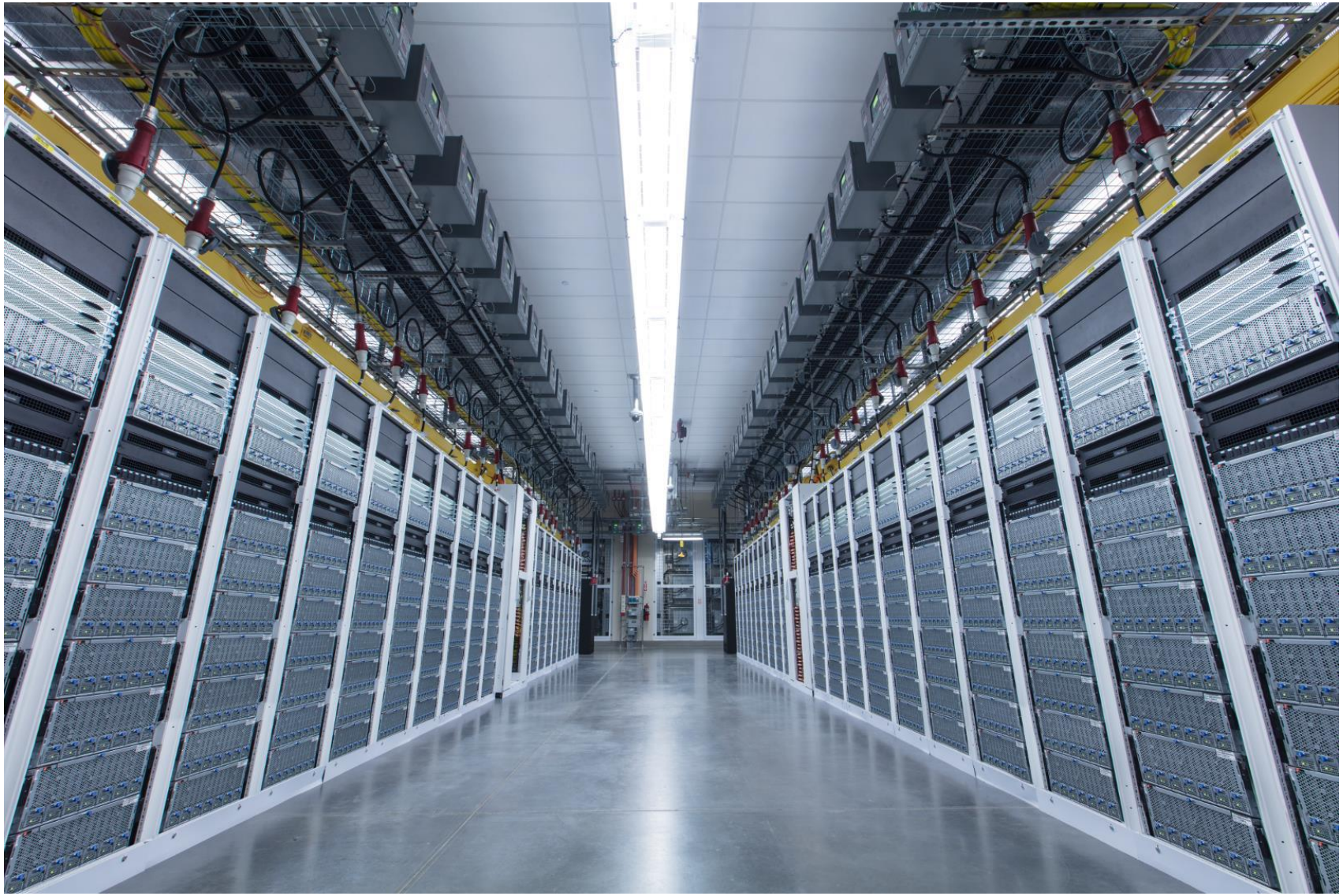

54 regions worldwide
140 available in 140 countries
West US 2
West Central US
West US
US Gov Arizona
South Central US
US Gov Texas
US Gov Iowa
Central US
North Central US
US DoD East
East US
East US 2
US Gov Virginia
US DoD Central
Canada East
Canada Central
Norway West
Norway East
West Europe
Germany West Central
UK South
Germany North
North Europe
Germany Northeast
UK West
Germany Central
France Central
Switzerland North
France South
Switzerland West
China North
China North 2
Korea Central
Korea South
Japan East
Japan West
China East 2
China East
East Asia
UAE North
UAE Central
West India
Central India
South India
Southeast Asia
Brazil South
South Africa North
South Africa West
Australia East
Australia Central
Australia Southeast
Australia Central 2
Available region
Announced region
Availability Zones
* Two Azure Government Secret region locations undisclosed

Your Data Lives Here (Amsterdam NL)

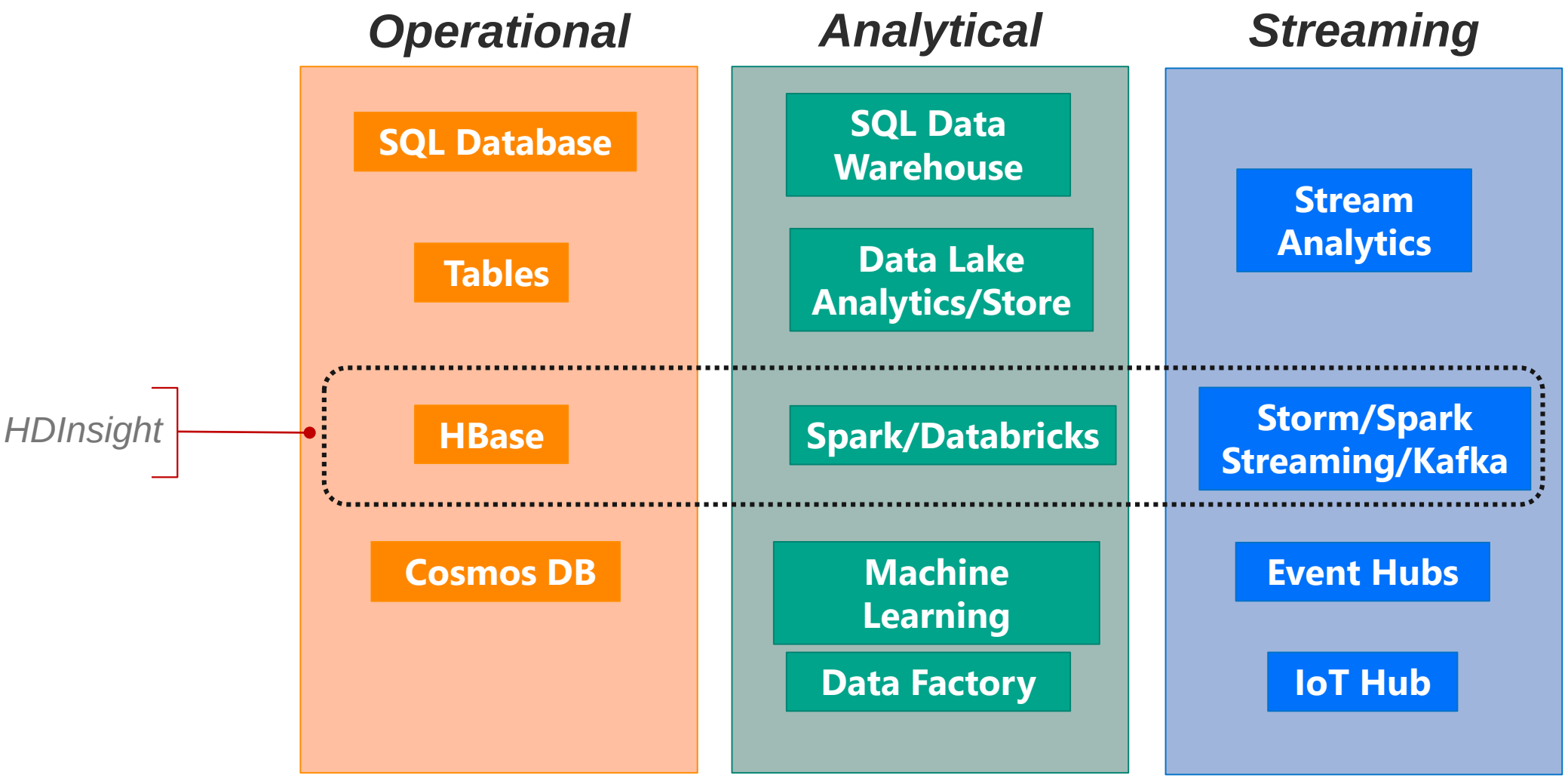
Your Data Lives Here (Cheyenne WY)

Your Data Lives Here (Quincy WY)

What it Looks Inside

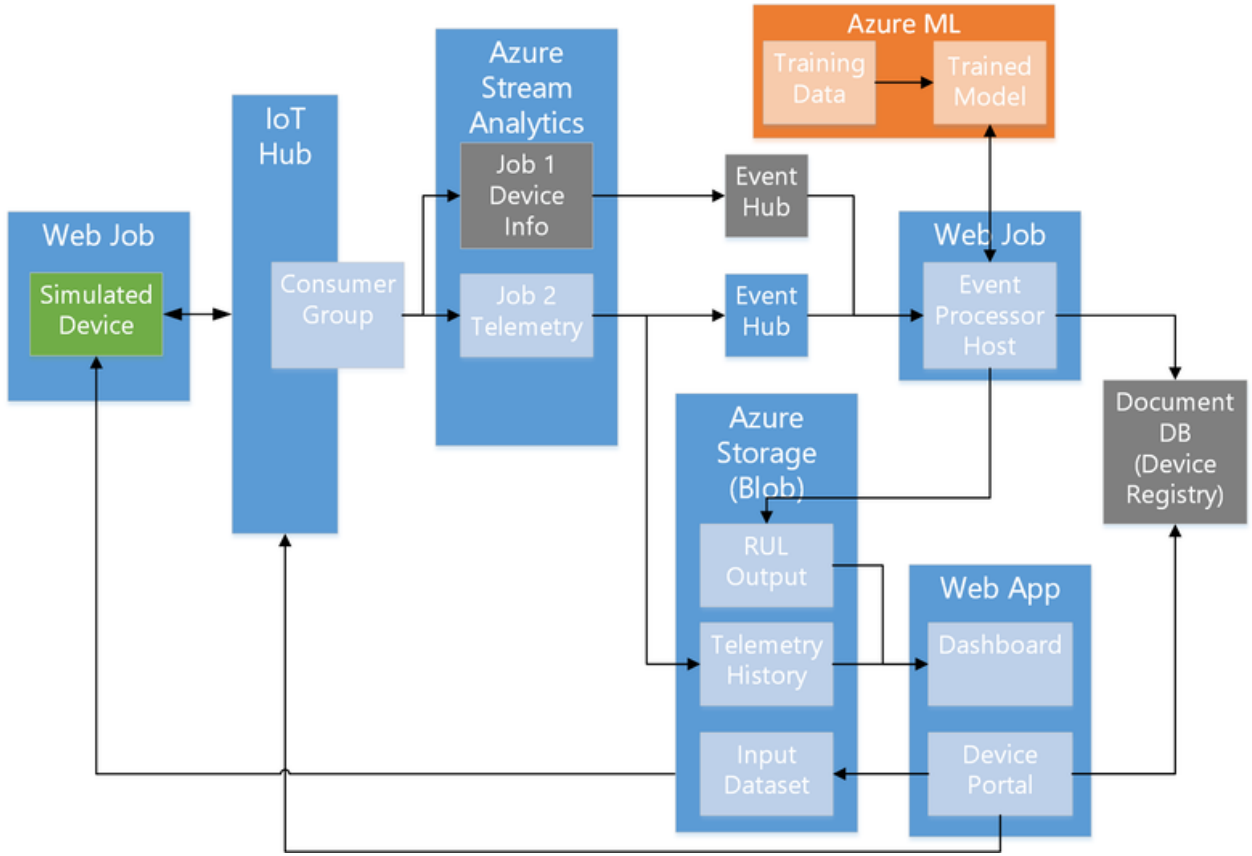


The Azure Data Platform



Lab: Predictive Maintenance solution accelerator

- https://www.azureiotsolutions.com/Accelerators#description/predictive-maintenance



Hadoop technology stack

RDBMS vs. Hadoop

RDBMS require a schema to be applied when the data is written:

- The data is transformed to accommodate the schema
- Some information hidden in the data may be lost at write-time

Hadoop/HDInsight applies a schema only when the data is read:

- The schema doesn't change the structure of the underlying data
- The data is stored in its original (raw) format so that all hidden information is retained

RDBMS perform query processing in a central location:

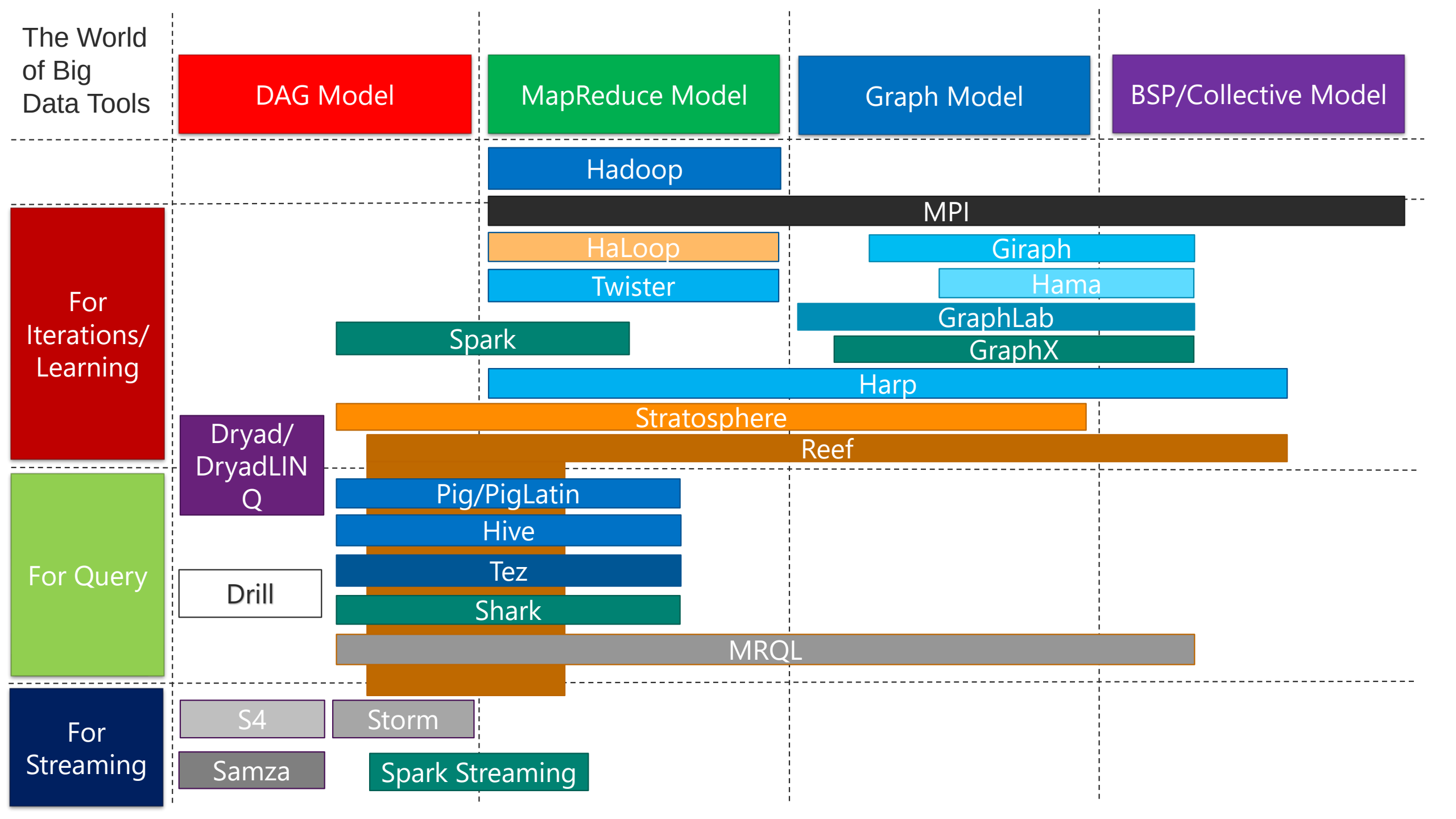
- Data is moved from storage to a central location for processing
- More central processing capacity is required to move data and execute the query

Hadoop performs query processing at each storage node:

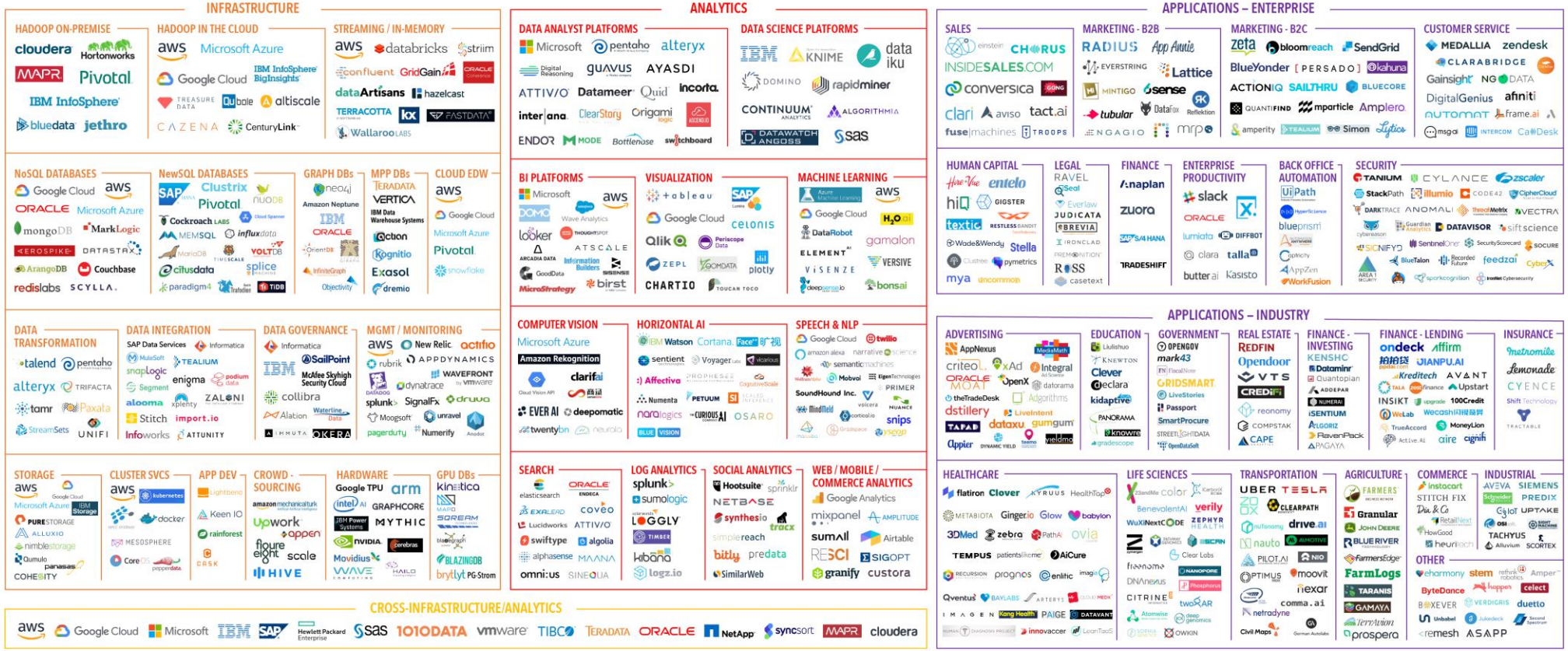
- Data doesn't need to be moved across the network for processing
- Only a fraction of the central processing capacity is required to execute the query

RDBMS vs. Hadoop

Feature	Relational Database	Hadoop / HDInsight
Data Types and Formats	Structured	Semi-Structured or Unstructured
Data Integrity	High: Transactional Updates	Low: Eventually Consistent
Schema	Static: Required on Write	Dynamic: Optional on Read & Write
Read and Write Pattern	Fully Repeatable Read/Write	Write Once; Repeatable Read
Storage Volume	Gigabytes to Petabytes	Terabytes, Petabytes and Beyond
Scalability	Scale Up with More Powerful HW	Scale Out with Additional Servers
Data Processing Distribution	Limited or None	Distributed Across Cluster
Economics	Expensive Hardware & Software	Commodity Hardware & Open Source Software


The World of Big Data Tools
DAG Model
MapReduce Model
Graph Model
BSP/Collective Model
Hadoop
MPI
HaLoop
Giraph
Twister
Hama
GraphLab
Spark
GraphX
Harp
Stratosphere
Reef
For Iterations/ Learning
Dryad/ DryadLINQ
Pig/PigLatin
Hive
Tez
Drill
Shark
MRQL
For Query
S4
Storm
Samza
Spark Streaming
For Streaming

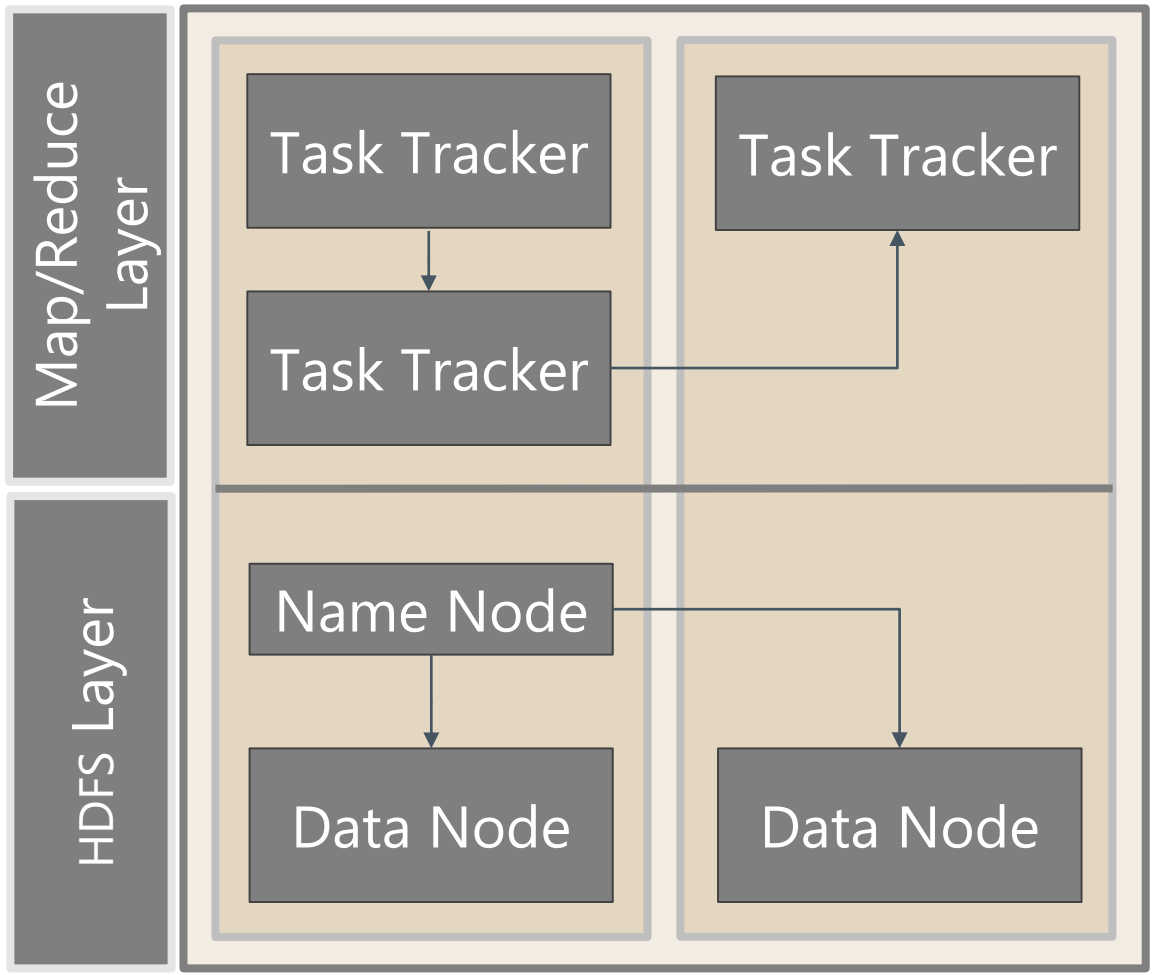
BIG DATA & AI LANDSCAPE 2018



FIRSTMARK
EARLY STAGE VENTURE CAPITAL

What is Hadoop and how does it work?

- Implements a Divide and Conquer Algorithm to Achieve Greater Parallelism
- Hadoop Distributed Architecture:
  - Storage Layer (HDFS)
  - Programming Layer (Map/Reduce)



Schema on Read vs. Schema on Write

- Traditionally we have always brought the data to the schema and code
- Hadoop sends the schema and the code to the data
- We don't have to pay the cost or live with the limitations of moving the data: IOPs, Network traffic, etc.

Key Features of MapReduce Model

- Designed for clouds
  - Large clusters of commodity machines
- Designed for big data
  - Support from local disks based distributed file system (GFS / HDFS)
  - Disk based intermediate data transfer in Shuffling
- MapReduce programming model
  - Computation pattern: Map tasks and Reduce tasks
- Data abstraction
  - KeyValue pairs

How MapReduce works?

Invoice	Date	Amount
1001	01-01-2016	$100.00
1002	01-01-2016	$95.00
1003	01-02-2016	$100.00
1003	01-03-2016	$75.00
1004	01-03-2016	$50.00

Map

Split data into Key/Value pairs

Key	Value
01-01-2016	{$100.00, $95.00}
01-02-2016	{$100.00}
01-03-2016	{$75.00, $50.00}

Reduce

Key	Value
01-01-2016	∑ = $195.00

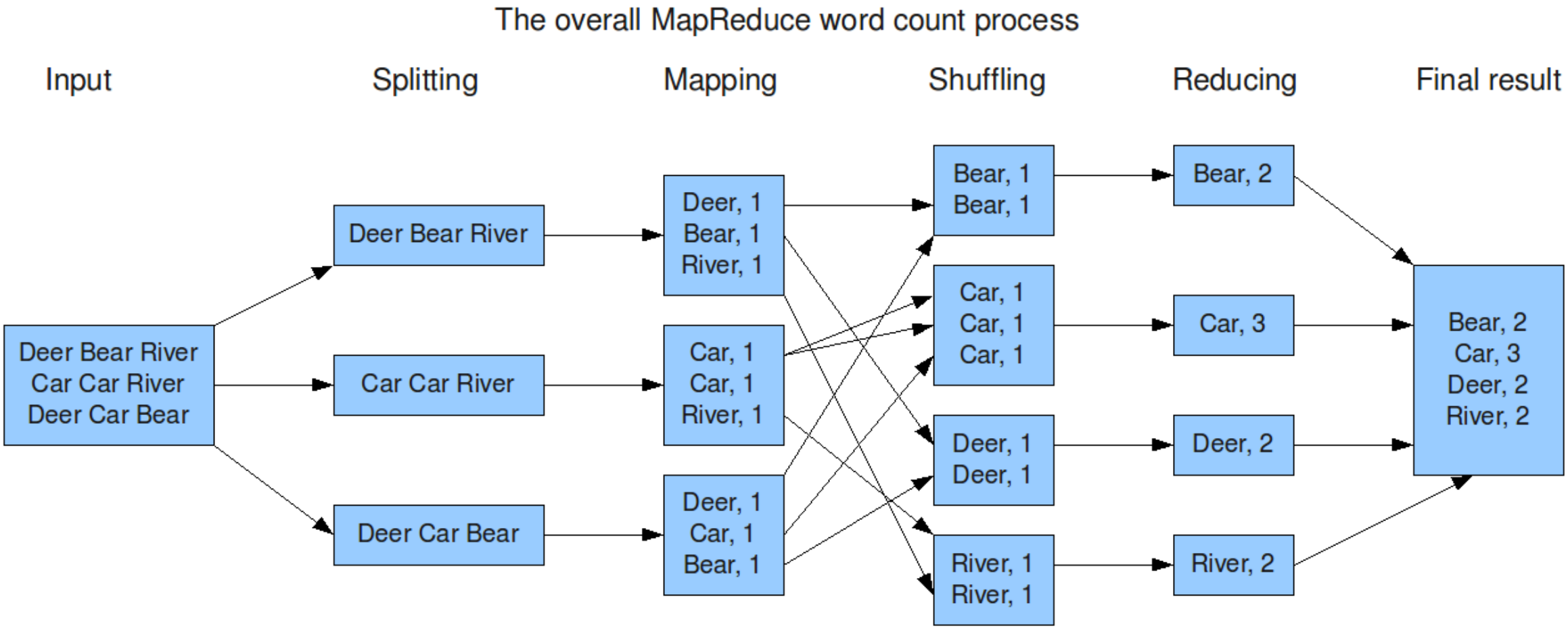
Key	Value
01-02-2016	∑ = $100.00

Key	Value
01-03-2016	∑ = $125.00

Output

Key	Value
01-01-2016	$195.00
01-02-2016	$100.00
01-03-2016	$125.00

“Hello World” of MapReduce



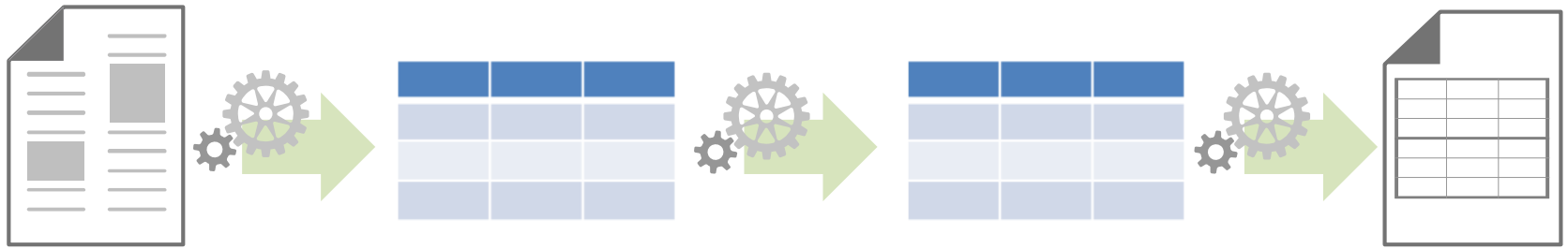
Pig and Hive

Query on Big Data

- Query with procedural language
  - Google Sawzall (2003)
    - Rob Pike et al. Interpreting the Data: Parallel Analysis with Sawzall. Special Issue on Grids and Worldwide Computing Programming Models and Infrastructure 2003.
  - Apache Pig (2006)
    - Christopher Olston et al. Pig Latin: A Not-So-Foreign Language for Data Processing. SIGMOD 2008.
    - https://pig.apache.org/

Pig

- Pig performs a series of transformations to data relations based on Pig Latin statements
- Relations are loaded using schema on read semantics to project table structure at runtime
- You can run Pig Latin statements interactively in the Grunt shell, or save a script file and run them as a batch



What kinds of things can I do with Pig?

```
2013-06-01,12
2013-06-01,14
2013-06-01,16
2013-06-02,9
2013-06-02,12
2013-06-02,9
...
```

- -- Load comma-delimited source data
- Readings = LOAD '/weather/data.txt' USING PigStorage(',') AS (date:chararray, temp:long);
- -- Group the tuples by date
- GroupedReadings = GROUP Readings BY date;
- -- Get the average temp value for each date grouping
- GroupedAvgs = FOREACH GroupedReadings GENERATE group, AVG(Readings.temp) AS avgtemp;
- -- Ungroup the dates with the average temp
- AvgWeather = FOREACH GroupedAvgs GENERATE FLATTEN(group) as date, avgtemp;
- -- Sort the results by date
- SortedResults = ORDER AvgWeather BY date ASC;
- -- Save the results in the /weather/summary folder
- STORE SortedResults INTO '/weather/summary';


```
2013-06-01  14.00
2013-06-02  10.00
```

Common Pig Latin operations

- `LOAD`
- `FILTER`
- `FOR EACH … GENERATE`
- `ORDER`
- `JOIN`
- `GROUP`
- `FLATTEN`
- `LIMIT`
- `DUMP`
- `STORE`

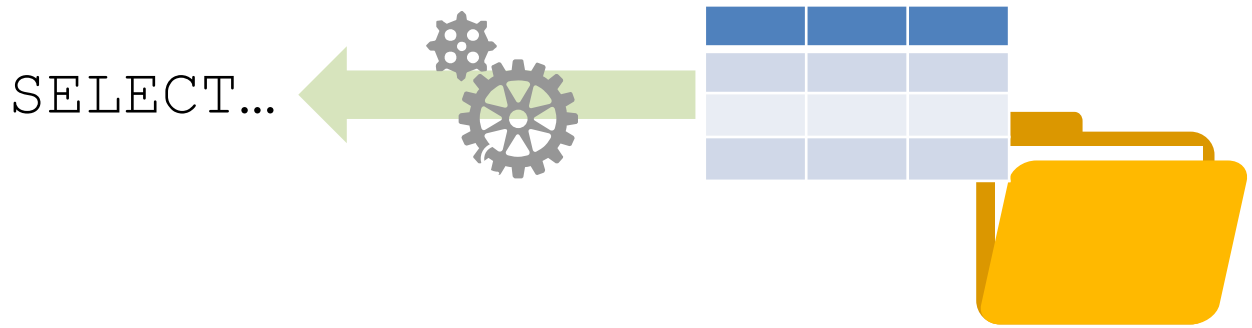
How do I run a Pig script?

- Save a Pig Latin script file
- Run the script using Pig
- Consume the results using any Azure storage client
  - For example, Excel or Power BI
  - Default output does not include schema – just data


```
pig wasb:///scripts/myscript.pig
```

Hive

- A metadata service that projects tabular schemas over folders
- Enables the contents of folders to be queried as tables, using SQL-like query semantics
- Queries are translated into jobs
  - Execution engine can be Tez or MapReduce



Query on Big Data

- Other Tools for Query
  - Apache Tez (2013)
    - http://tez.incubator.apache.org/
    - To build complex DAG of tasks for Apache Pig and Apache Hive
    - On top of YARN
  - Dremel (2010) Apache Drill (2012)
    - Sergey Melnik et al. Dremel: Interactive Analysis of Web-Scale Datasets. VLDB 2010.
    - http://incubator.apache.org/drill/index.html
    - System for interactive query

How do I create and load Hive tables?

- Use the CREATE TABLE HiveQL statement
  - Defines schema metadata to be projected onto data in a folder when the table is queried (not when it is created)
- Specify file format and file location
  - Defaults to textfile format in the <database>/<table_name> folder
    - Default database is in /hive/warehouse
    - Create additional databases using CREATE DATABASE
- Create internal or external tables
  - Internal tables manage the lifetime of the underlying folders
  - External tables are managed independently from folders

How do I create and load Hive tables?

```
CREATE TABLE table1
(col1 STRING,
 col2 INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ' ';
```

Internal table (folders deleted when table is dropped)

Default location (/hive/warehouse/table1)

```
CREATE TABLE table2
(col1 STRING,
 col2 INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ' '
STORED AS TEXTFILE LOCATION '/data/table2';
```

Stored in a custom folder (but still internal, so the folder is deleted when table is dropped)

```
CREATE EXTERNAL TABLE table3
(col1 STRING,
 col2 INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ' '
STORED AS TEXTFILE LOCATION '/data/table3';
```

External table (folders and files are left intact in Azure Blob Store when the table is dropped)

Data loading

- Save data files in table folders (or create table on existing files)

```
PUT myfile.txt /data/table1
```

- Use the LOAD statement

```
LOAD DATA [LOCAL] INPATH '/data/source' INTO TABLE MyTable;
```

- Use the INSERT statement

```
INSERT INTO TABLE Table2
SELECT Col1, UPPER(Col2),
FROM Table1;
```

- Use a CREATE TABLE AS SELECT (CTAS) statement

```
CREATE TABLE Table3
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE LOCATION '/data/summarytable'
AS SELECT Col1, SUM(Col2) As Total
FROM Table1 GROUP BY Col1;
```

How do I query Hive tables?

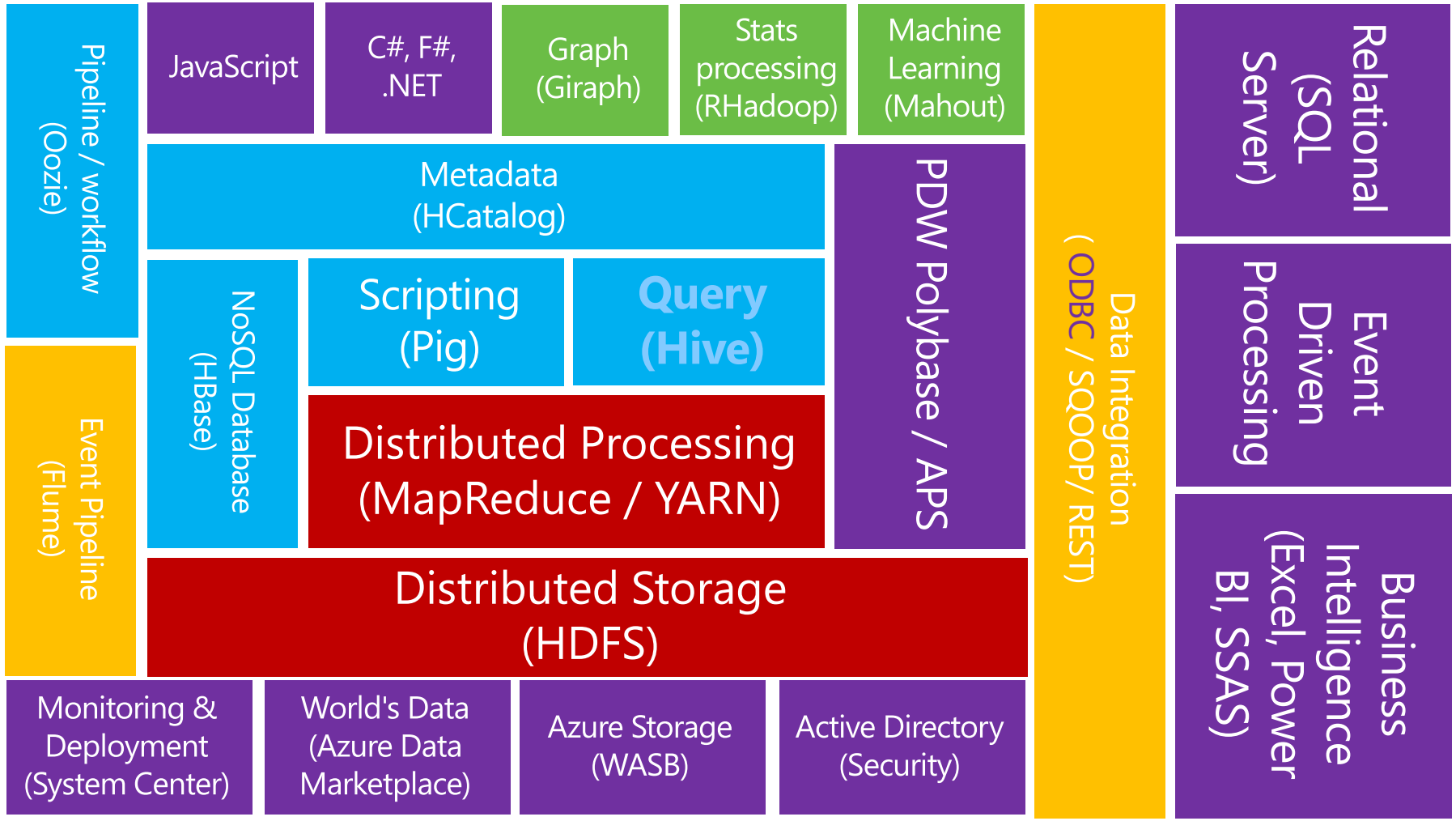
- Query data using the SELECT HiveQL statement

```
SELECT Col1, SUM(Col2) AS TotalCol2
FROM MyTable
WHERE Col3 = 'ABC' AND Col4 < 10
GROUP BY Col1
ORDER BY Col4;
```

- Hive translates the query into jobs and applies the table schema to the underlying data files

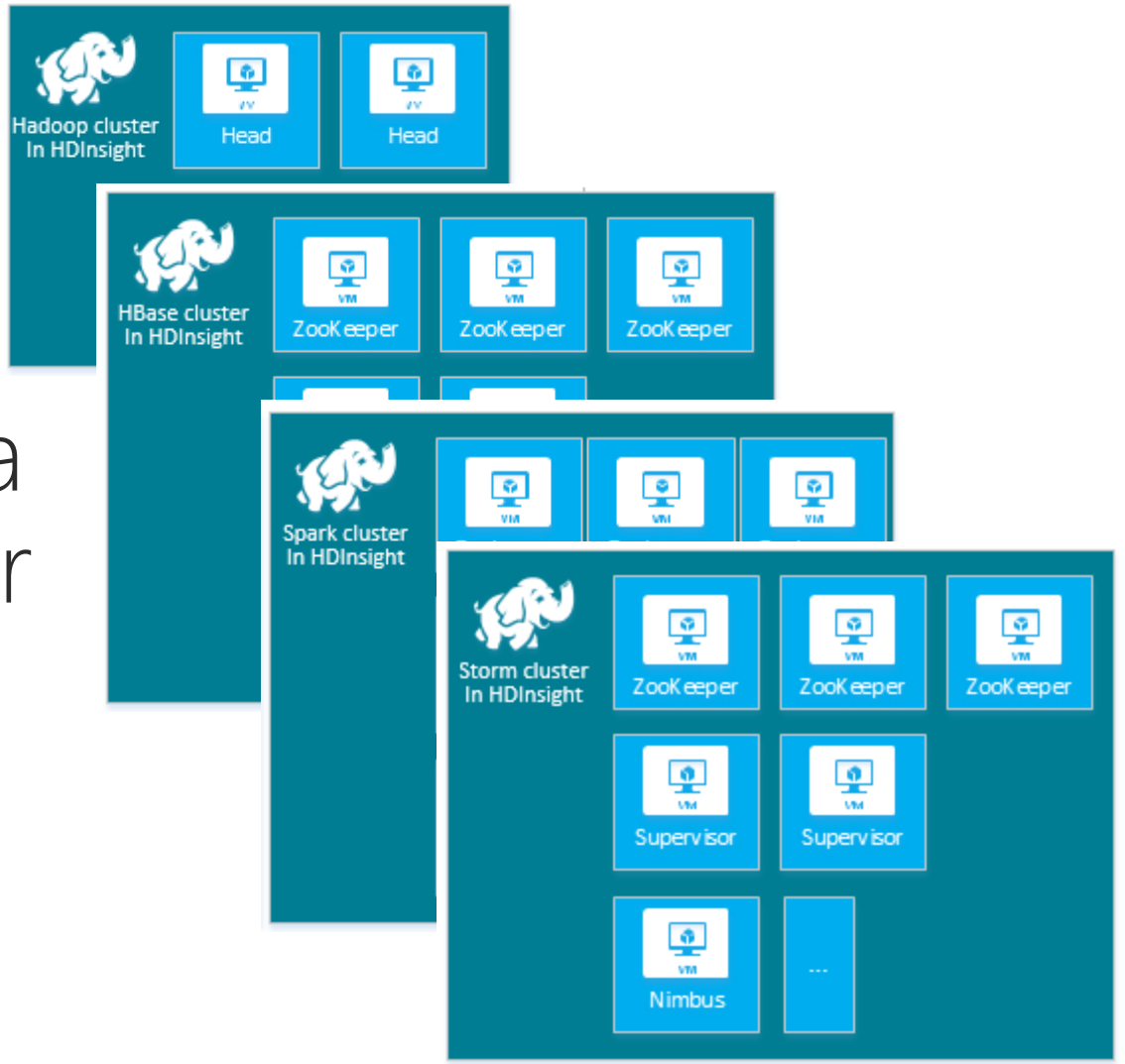
HDInsight

HDInsight ecosystem

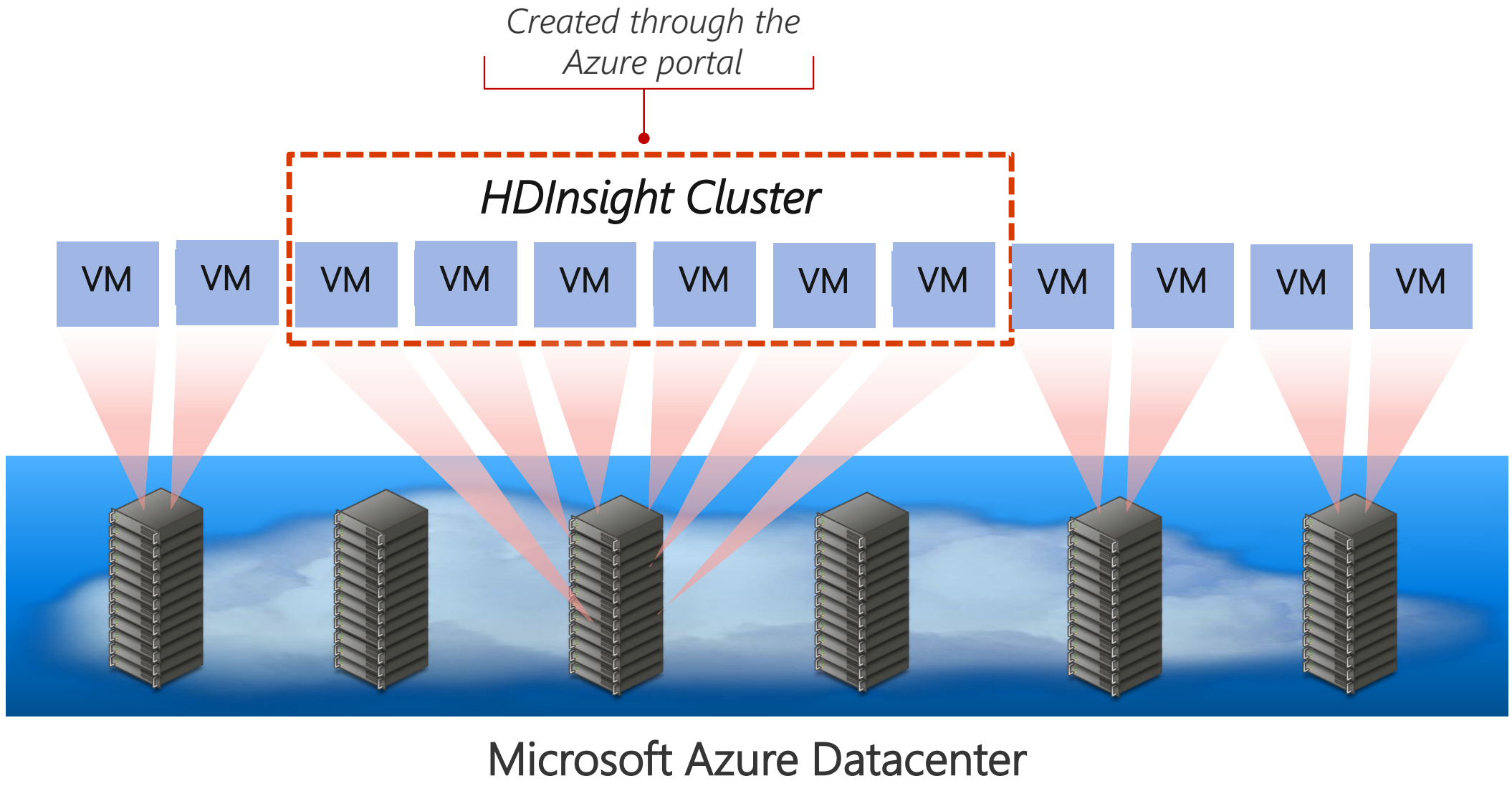


HDInsight: Hadoop on request

- Hortonworks HDP on Azure VMs, Linux or Windows
  - How: PowerShell, ARM Templates, C# SDK,Azure CL
- Azure Storage or Azure Data Lake provides the HDFS layer
  - Inexpensive, stored independently from cluster
- Azure SQL Database stores metadata

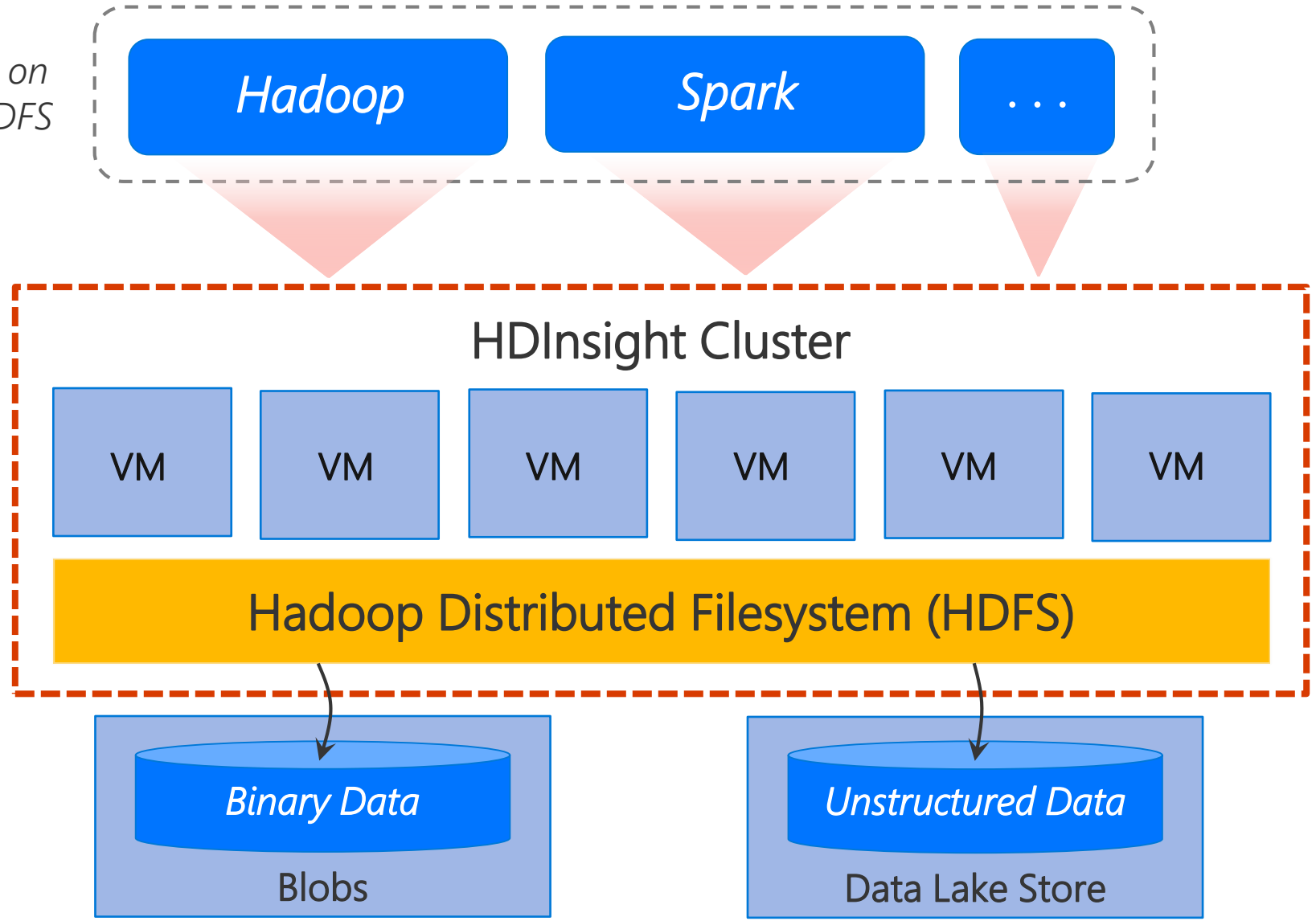


HDInsight clusters

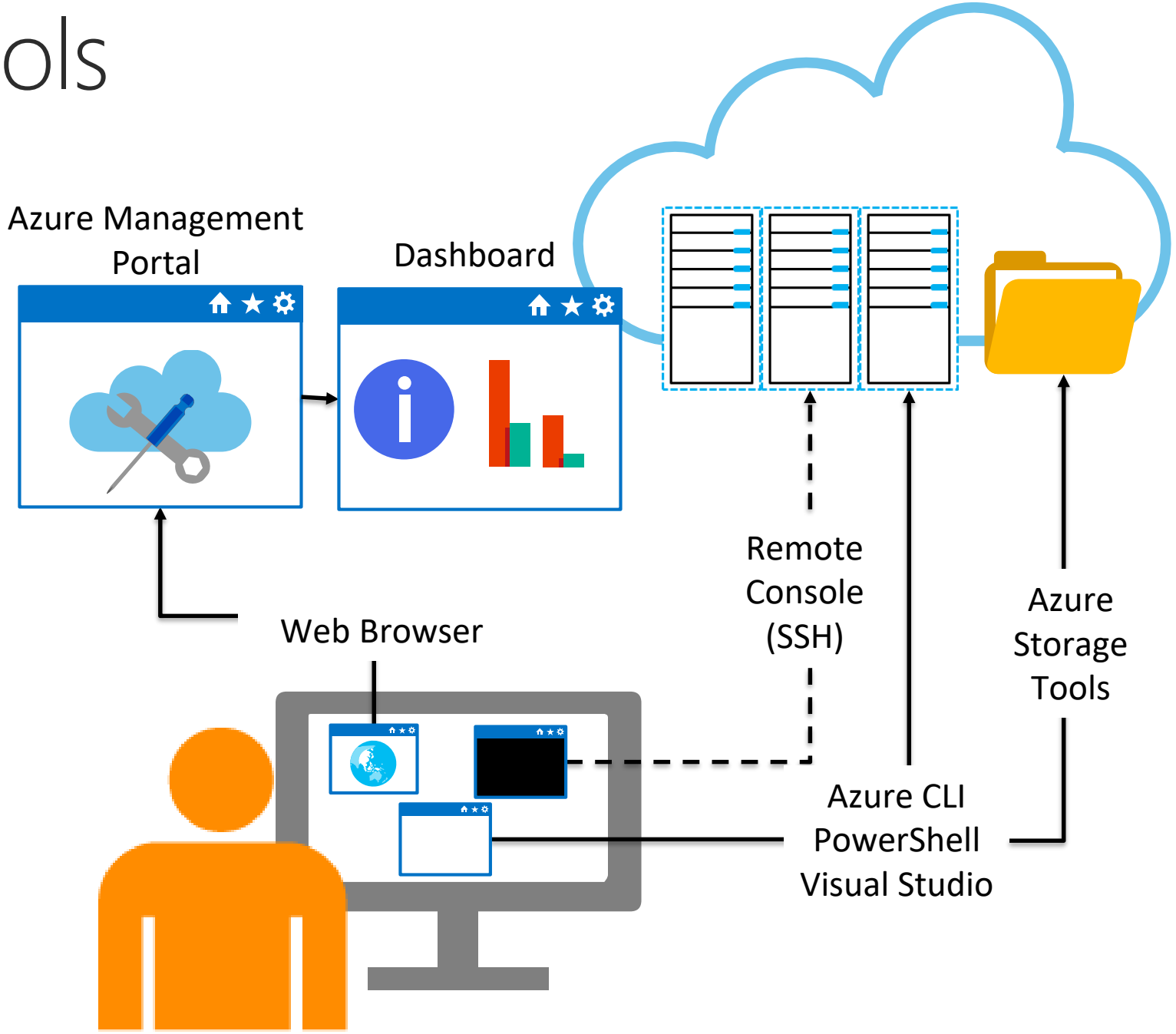


HDInsight technologies

All open source, run on a cluster, and use HDFS



Client tools

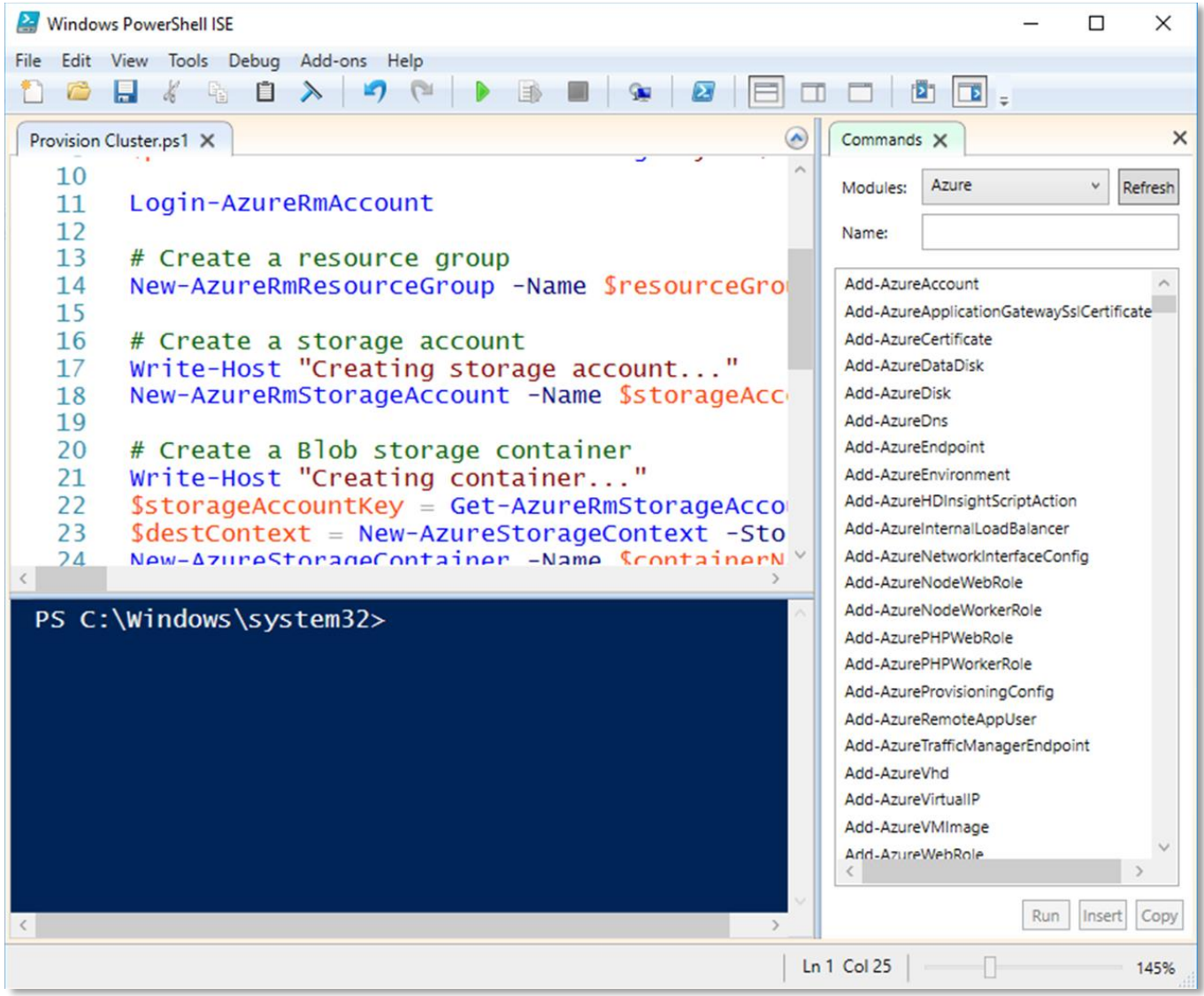


HDFS (Hadoop Distributed Files System)

- Fault Tolerant
  - Data is distributed across each Data Node in the cluster (like RAID 5)
  - 3 copies of the data is stored in case of storage failures
  - Data faults can be quickly detected and repaired due to data redundancy
- High Throughput
  - Favors batch over interactive operations to support streaming large datasets
  - Data files are written once and then closed; never to be updated.
  - Supports data locality - HDFS facilitates moving the application code (query) to the data rather than moving the data to the application (schema on read)

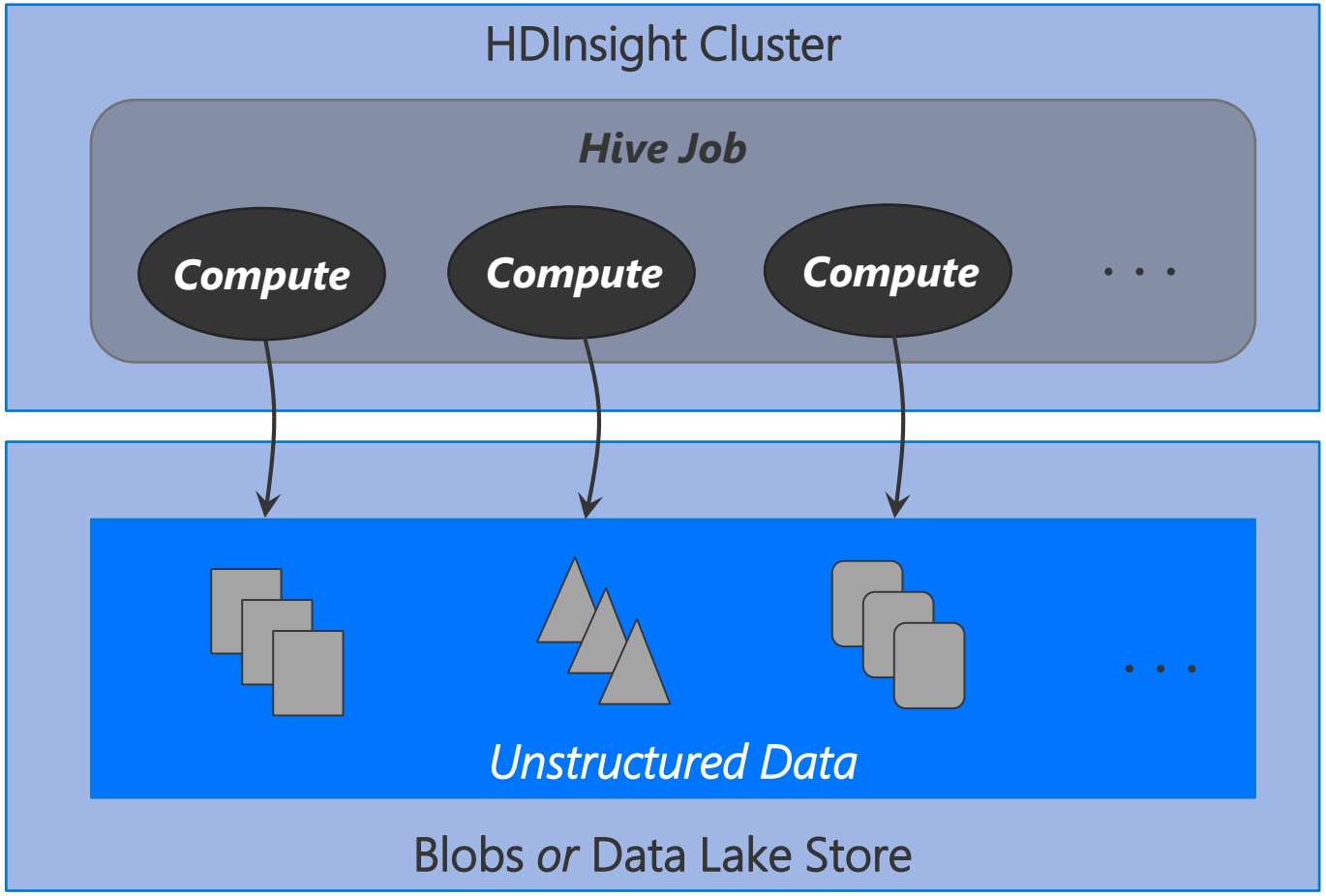
How do I Run a MapReduce Job?

- The Azure PowerShell module includes cmdlets to work with Azure services, including HDInsight
- Use PowerShell to:
  - Provision HDInsight clusters
  - Upload/download files
  - Submit jobs
  - Manage cluster resources



Lab: Processing Big Data with Hadoop in Azure HDInsight

- Lab 1 - Getting Started with HDInsight
- Lab 2 - Processing Big Data with Hive
- Lab 3 - Beyond Hive - Pig and Python



Spark

What is Apache Spark

Apache Spark emerged to provide a parallel processing framework that supports in-memory processing to boost the performance of big-data analytical applications on massive volumes of data.

Interactive Data Analysis
Used by business analysts or data engineers to analyze and prepare data.

Streaming Analytics
Ingest data from technologies such as Kafka and Flume to ingest data in real-time.

Machine Learning
Contains a number of libraries that enables a Data Scientist to perform Machine Learning.

Why use Azure Databricks

Azure Databricks is a wrapper around Apache Spark that simplifies the provisioning and configuration of a Spark cluster in a GUI interface

.

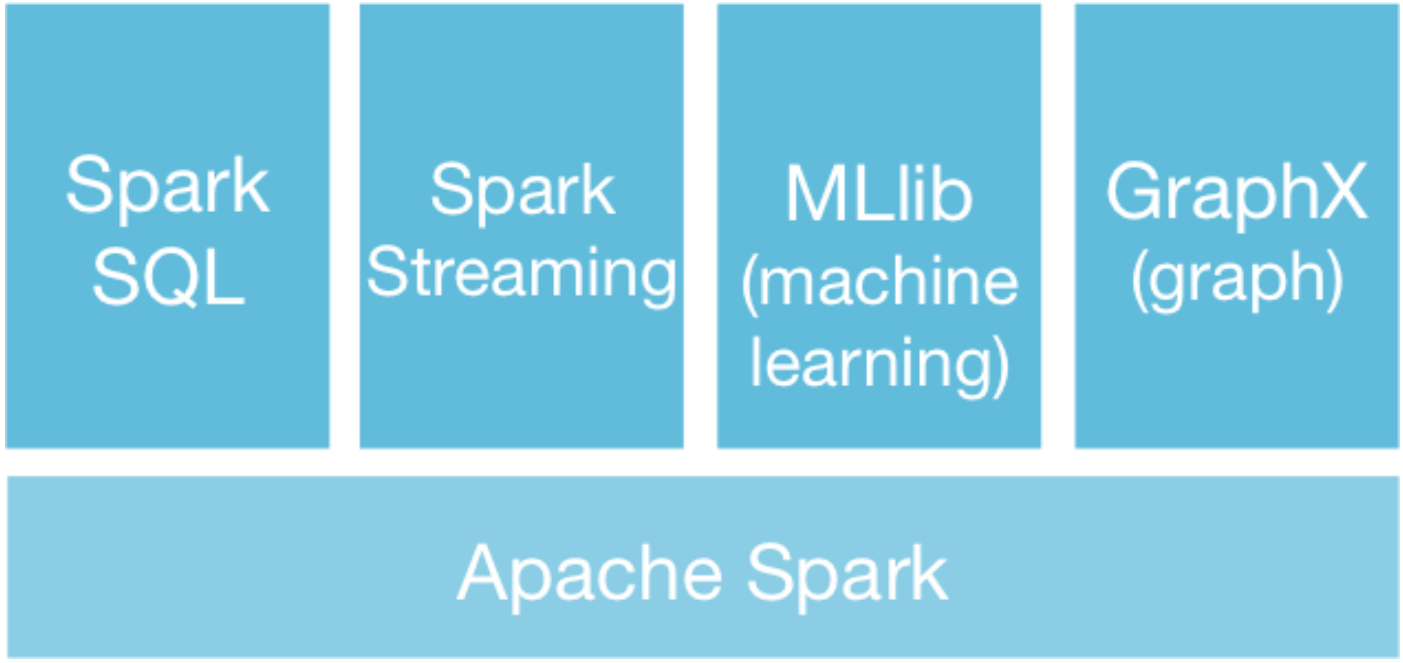
Azure Databricks components.

- Spark SQL and DataFrames
- Streaming
- Mlib
- GraphX
- Spark Core API

Apache Spark – An Unified Framework

An unified, open source, parallel, data processing framework for Big Data Analytics

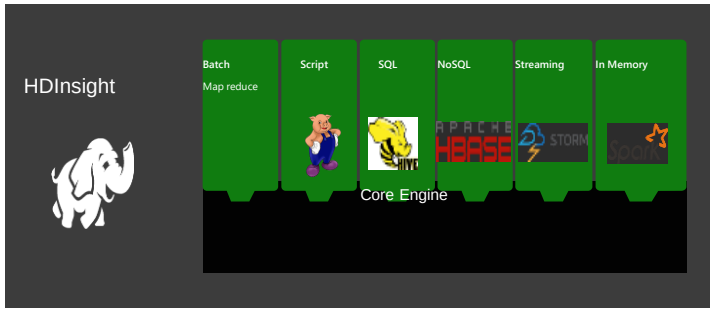
- Spark Unifies:
  - Batch Processing
  - Real-time processing
  - Stream Analytics
  - Machine Learning
  - Interactive SQL

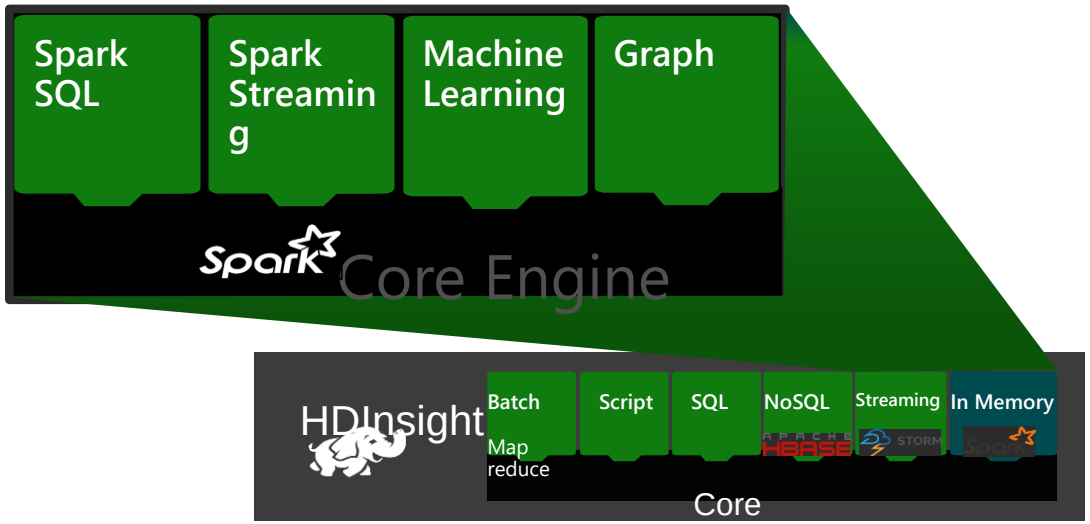


https://spark.apache.org

Spark for In-Memory Computation

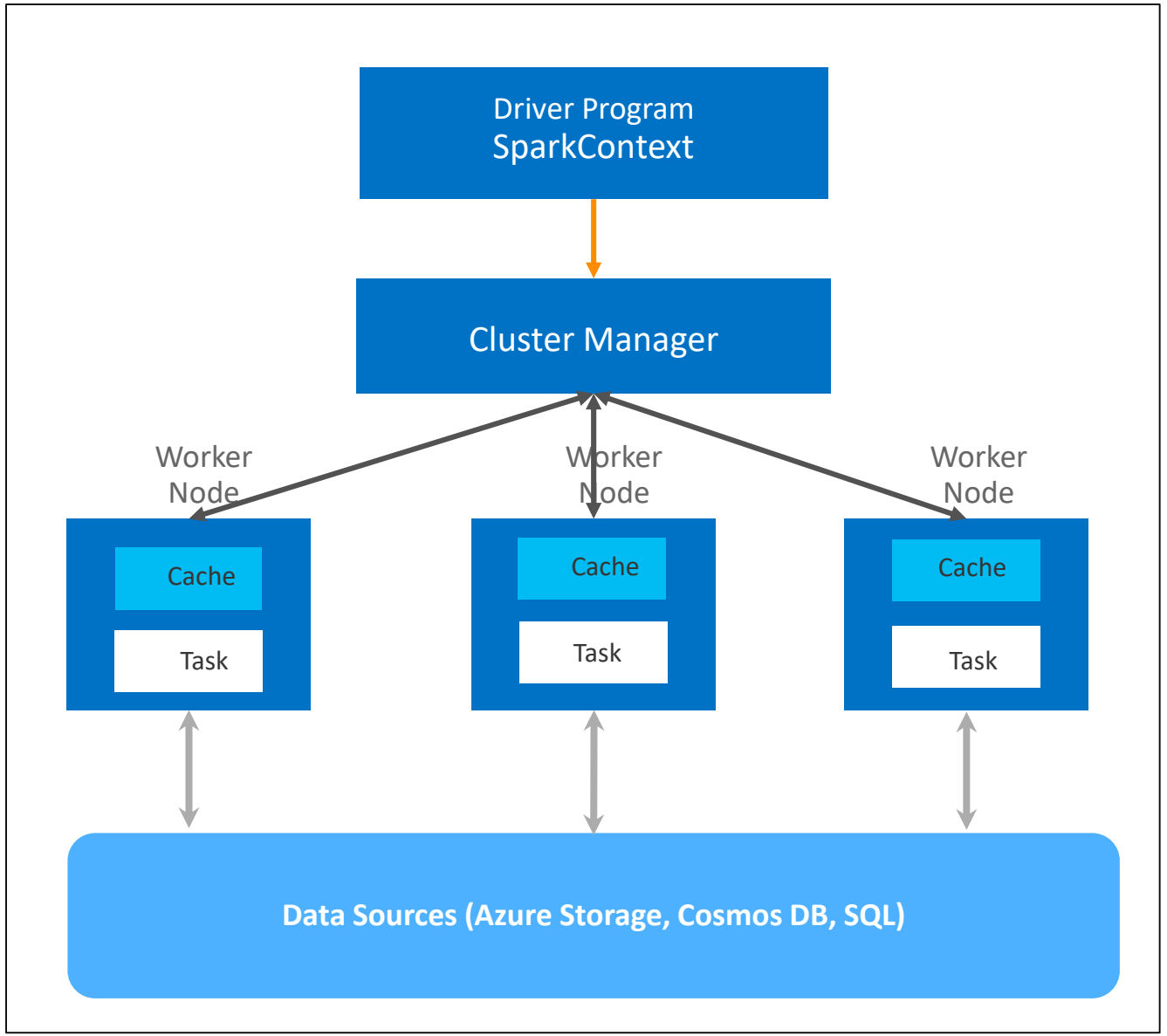
- Single execution model for multiple tasks (SQL queries, Streaming, Machine Learning, and Graph)
- Processing up to 100x faster performance
- Developer friendly (Java, Python, Scala)
- BI tool of choice (Power BI, Tabelau, Qlik, SAP)
- Notebook experience (Jupyter/iPython, Zeppelin)





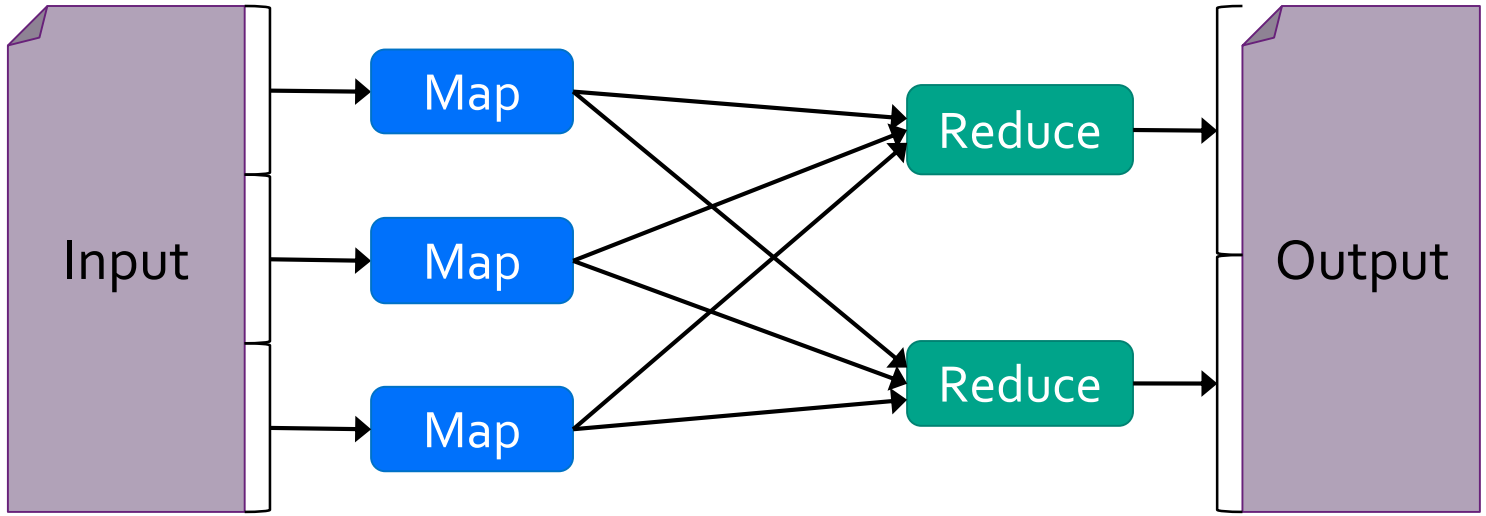
Spark Architecture & Dataframes

- 'Driver' runs the user's 'main' function and executes the various parallel operations on the worker nodes
- To take advantage of Spark – you use Dataframes as the data structure
- Once your Data is in the DataFrame – Spark can parallelize operations
- The Dataframes support both batch and streaming data
- The results of the operations are collected by the driver



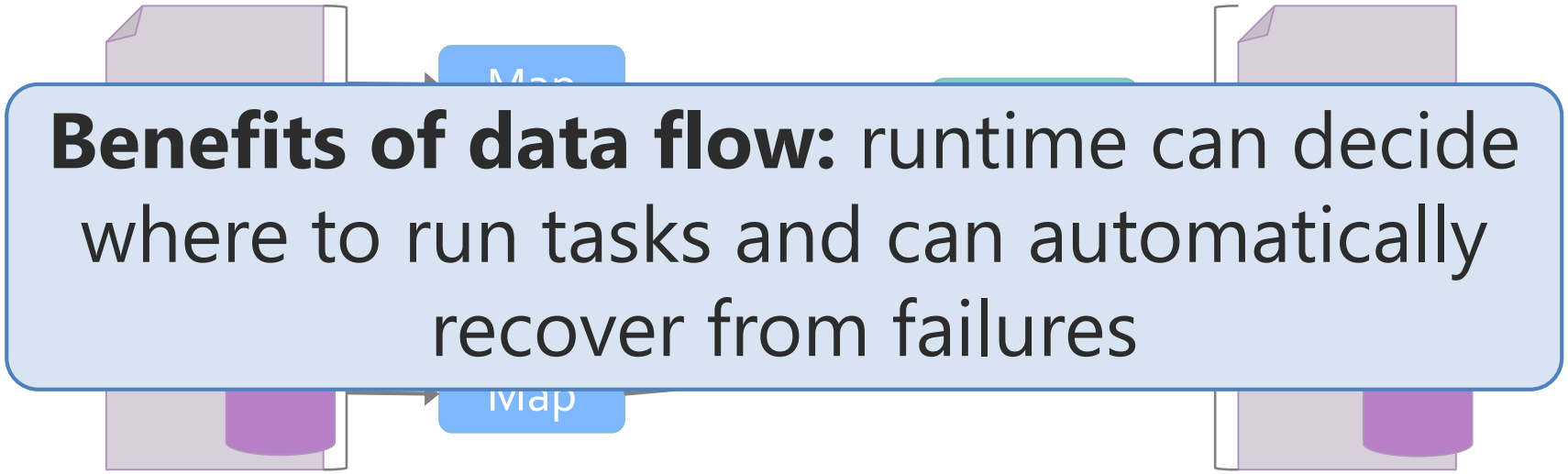
Motivation

- Most traditional cluster programming models are based on acyclic data flow from stable storage to stable storage



Motivation

- Most traditional cluster programming models are based on acyclic data flow from stable storage to stable storage



Motivation

- Acyclic data flow is inefficient for applications that repeatedly reuse a working set of data:
  - Iterative algorithms (machine learning, graphs)
  - Interactive data mining tools (R, Excel, Python)
- With traditional frameworks, apps reload data from stable storage on each query

Solution: Resilient Distributed Datasets (RDDs)

- Allow apps to keep working sets in memory for efficient reuse
- Retain the attractive properties of MapReduce
  - Fault tolerance, data locality, scalability
- Support a wide range of applications

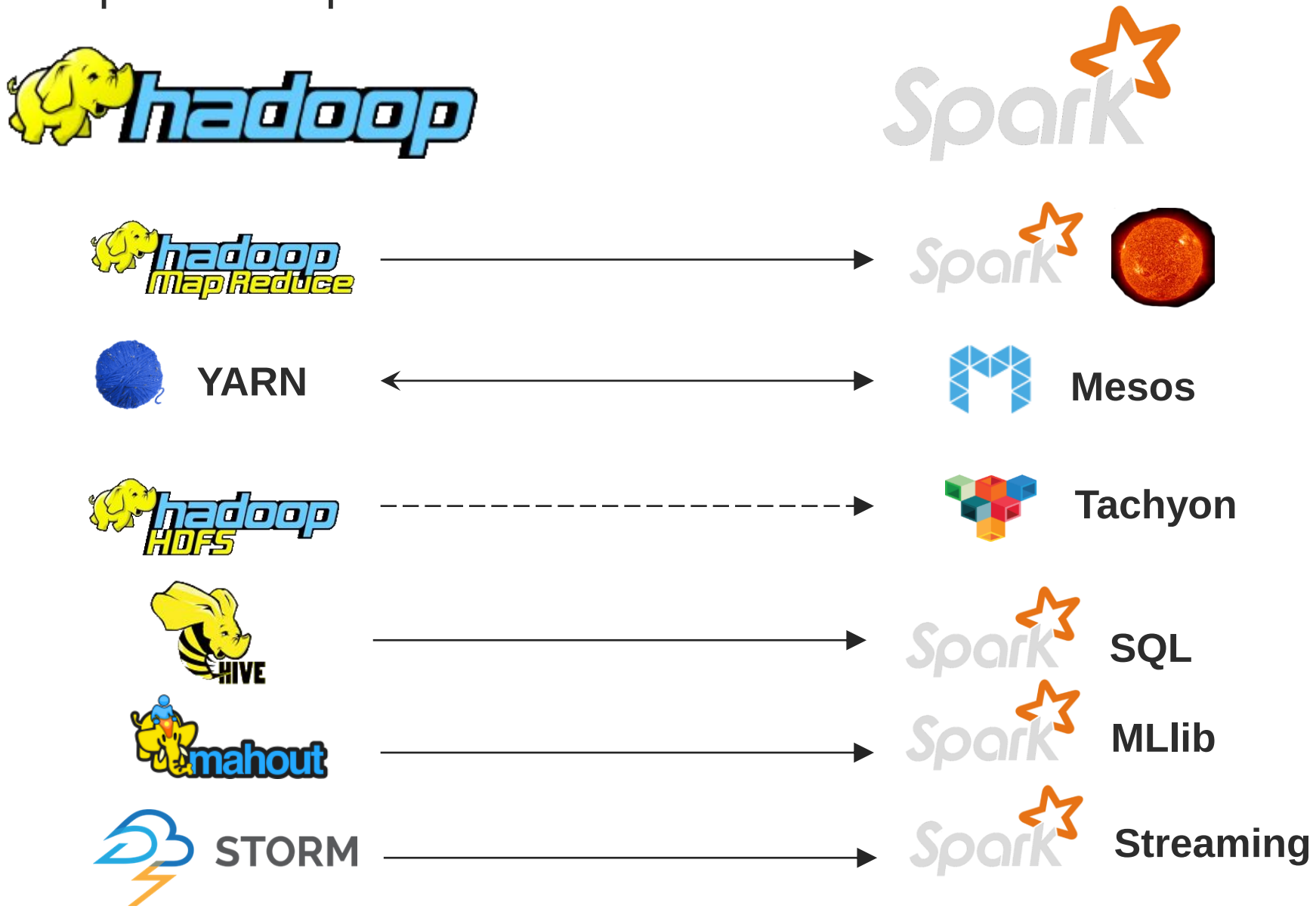
Programming Model

- Resilient distributed datasets (RDDs)
  - Immutable, partitioned collections of objects
  - Created through parallel transformations (map, filter, groupBy, join, …) on data in stable storage
  - Can be cached for efficient reuse
- Actions on RDDs
  - Count, reduce, collect, save, …

Spark operations

Transformations (define a new RDD)	map filter sample groupByKey reduceByKey sortByKey	flatMap union join cogroup cross mapValues
Actions (return a result to driver program)	collect reduce count save lookupKey	

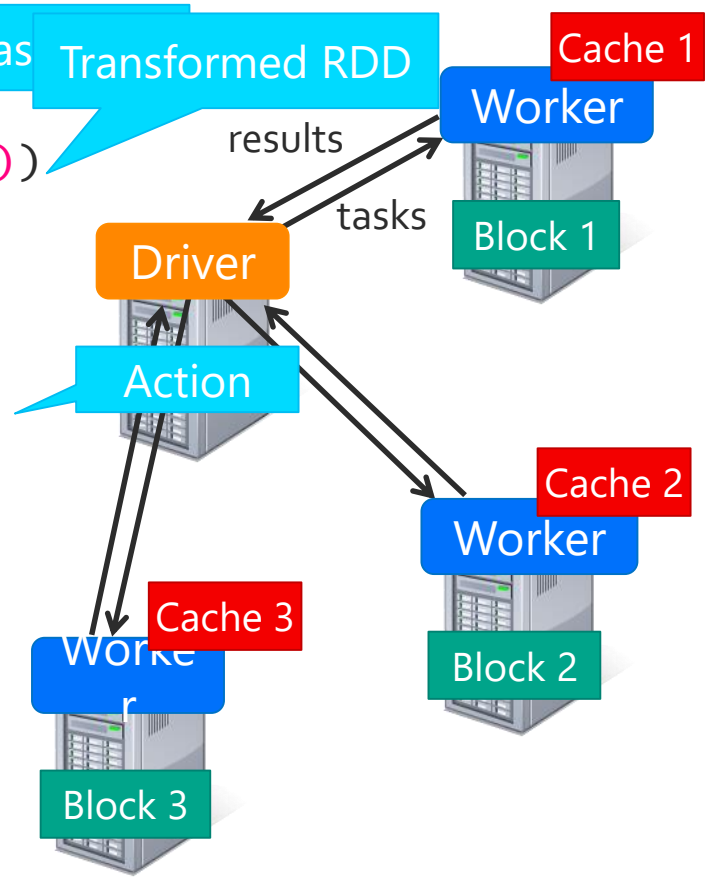
Hadoop vs Spark



Example: Log Mining

Load error messages from a log into memory, then interactively search for various patterns

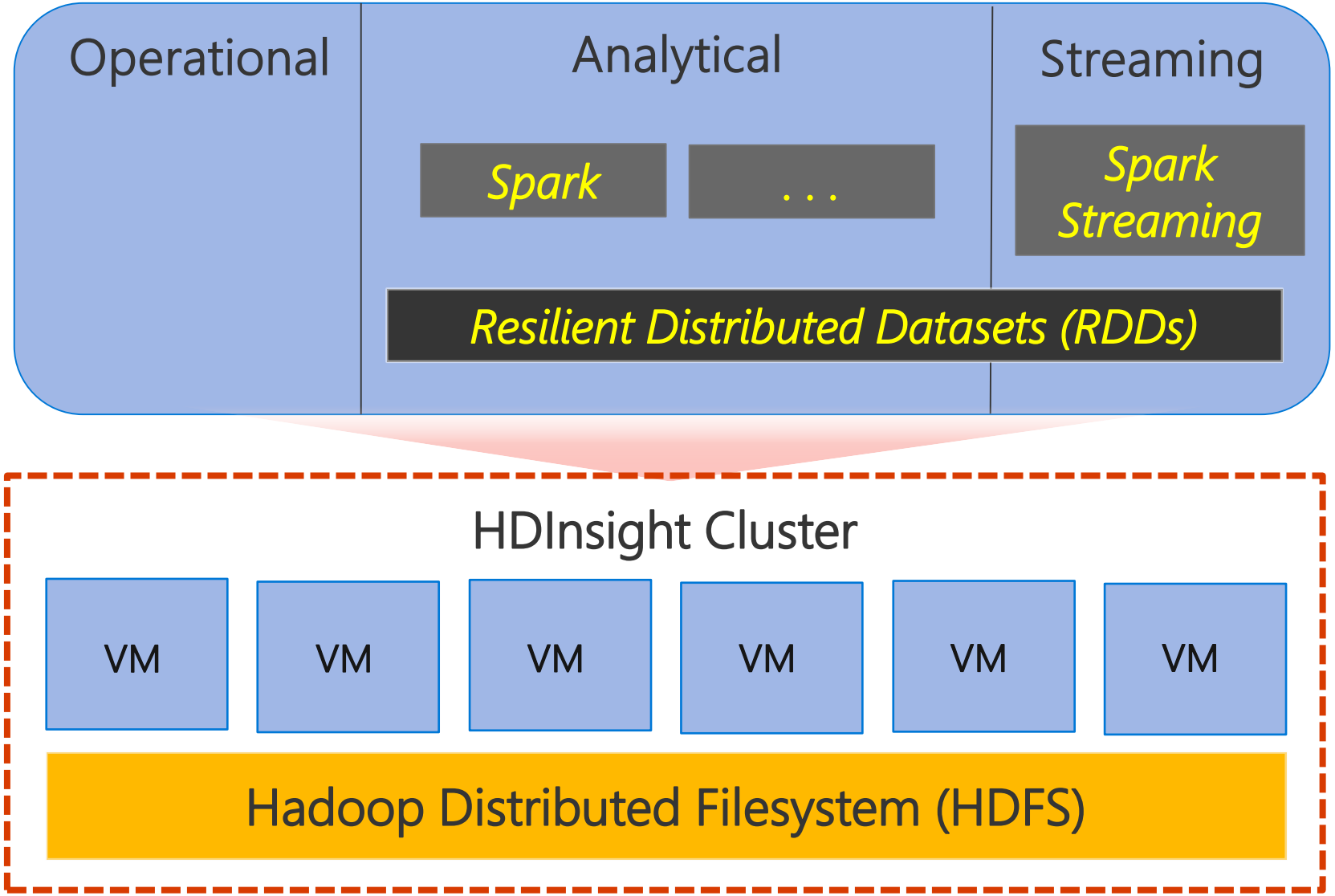
```
lines = spark.textFile(“hdfs://...”)
errors = lines.filter(_.startsWith(“ERROR”))
messages = errors.map(_.split(‘\t’)(2))
cachedMsgs = messages.cache()

cachedMsgs.filter(_.contains(“foo”)).count
cachedMsgs.filter(_.contains(“bar”)).count
. . .
```



Result: scaled to 1 TB data in 5-7 sec
(vs 170 sec for on-disk data)

HDInsight Spark

Spark has more mindshare than Hadoop today



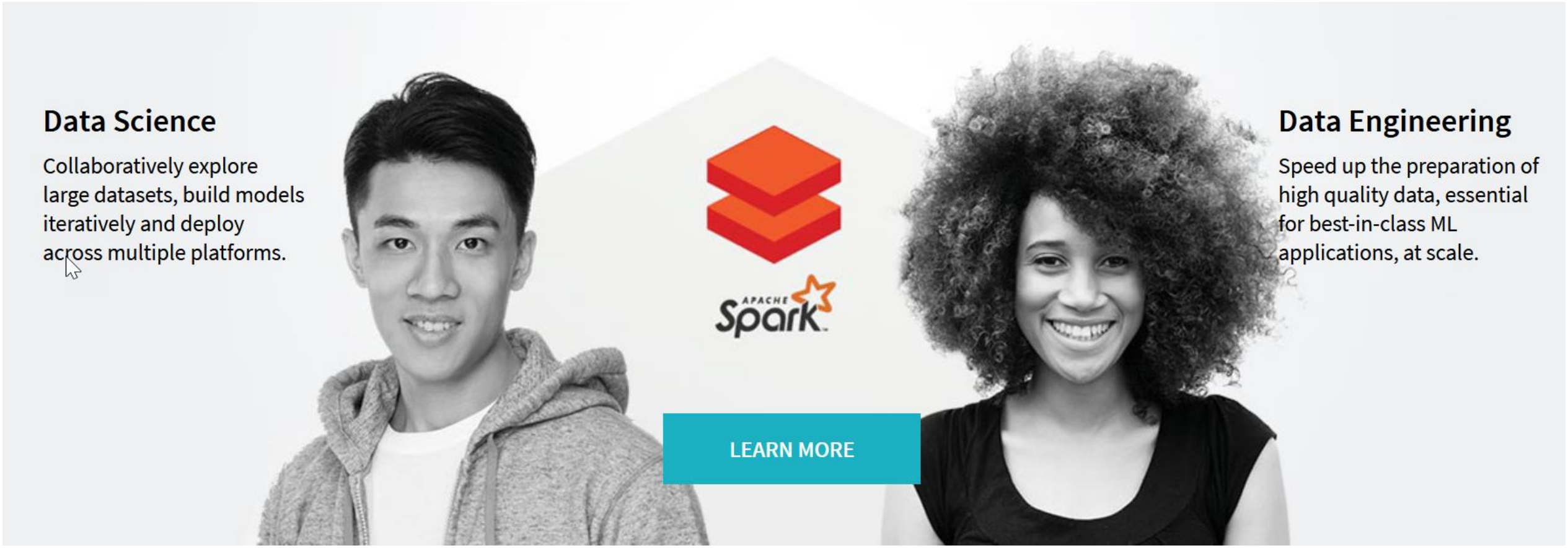
Lab: Exploring Data with Spark

- Provisioning an HDInsight Spark Cluster
- Exploring Data

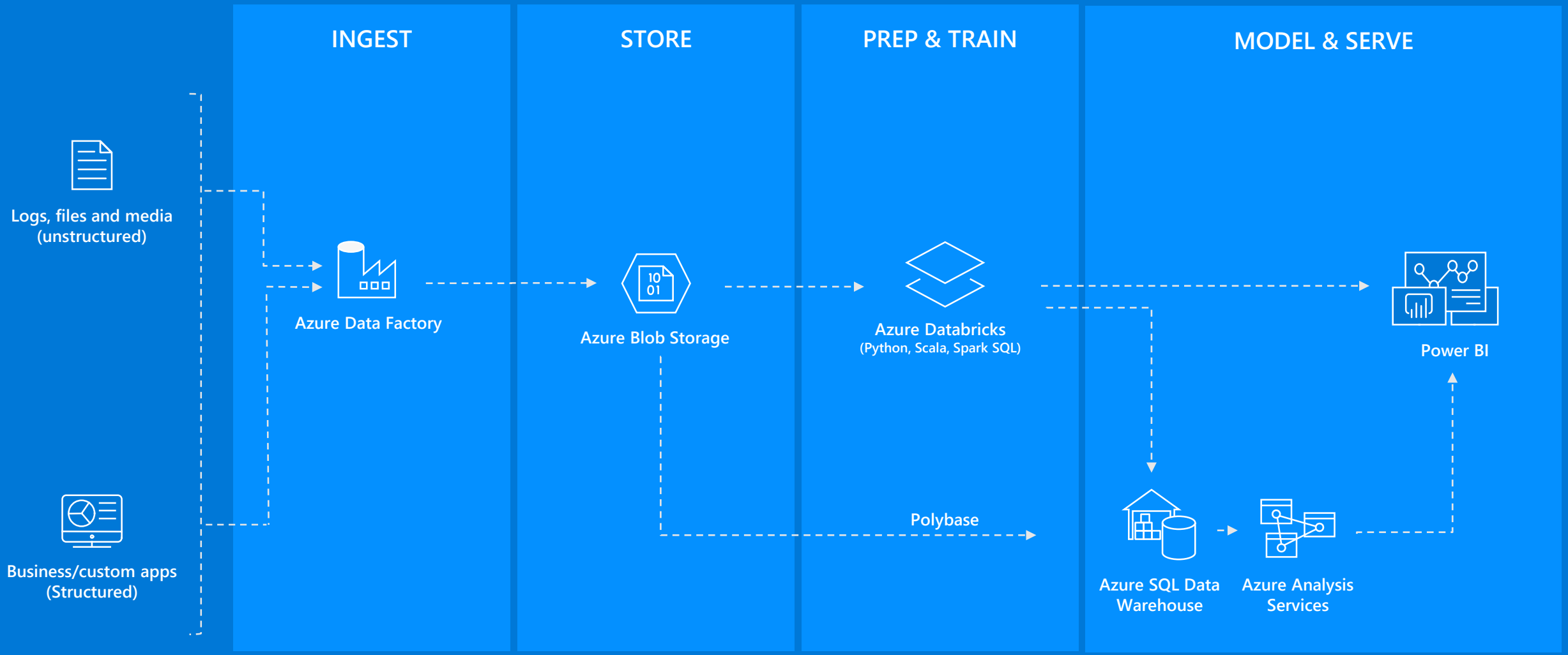
Databricks

Databricks: making big data simple

Unifying Data Science and Engineering

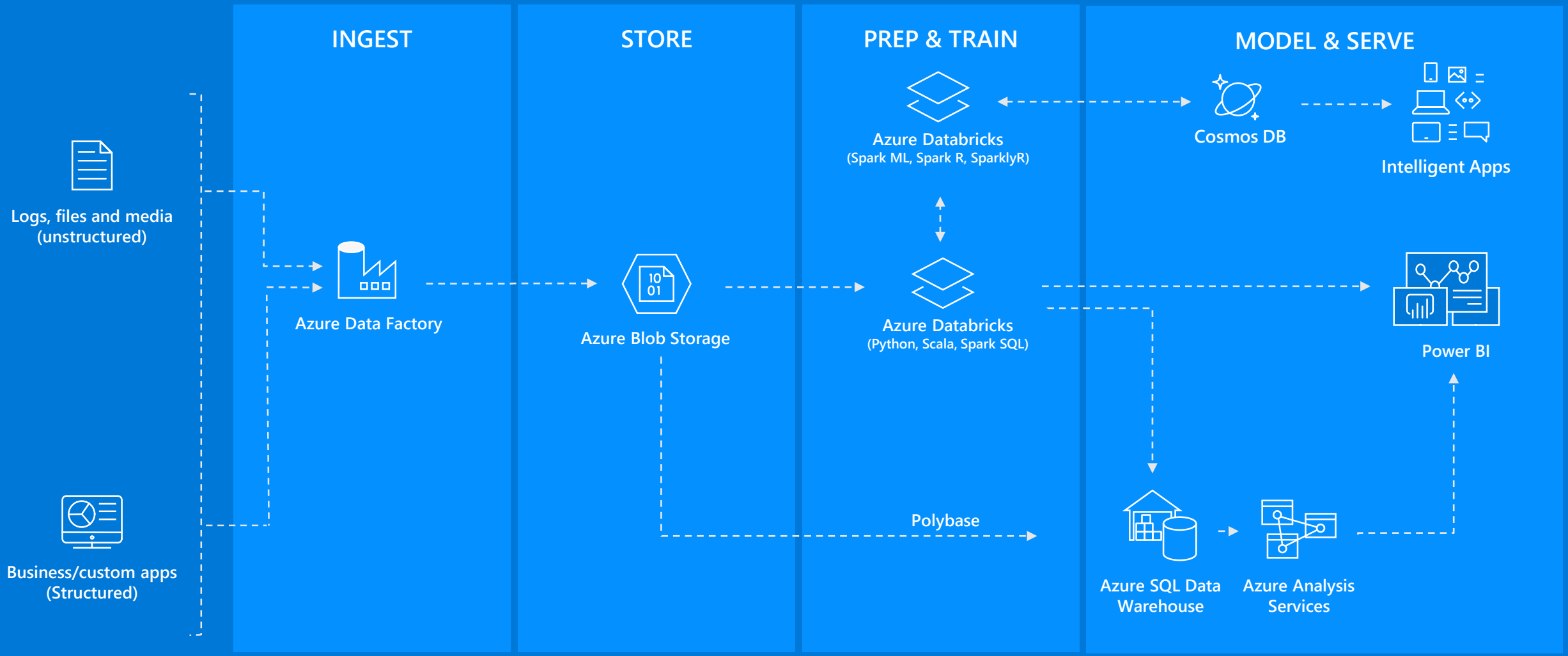


Modern Data Warehouse



Azure also supports other Big Data services like Azure HDInsight and Azure Data Lake to allow customers to tailor the above architecture to meet their unique needs.

Advanced analytics



Azure also supports other Big Data services like Azure HDInsight and Azure Data Lake to allow customers to tailor the above architecture to meet their unique needs.

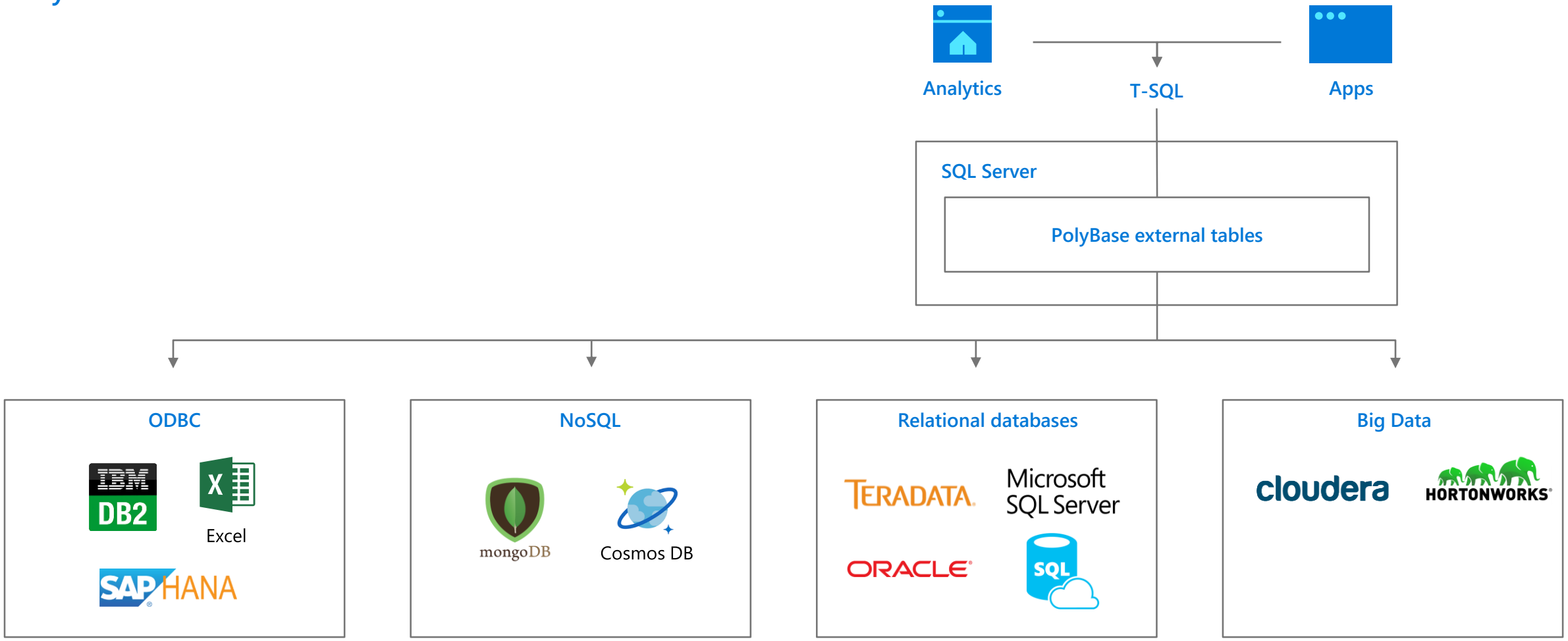
Azure Databricks

- First party service on Azure
  - Unlike with other clouds, it is not an Azure Marketplace or a 3rd party hosted service
- Azure Databricks is integrated seamlessly with Azure services:
  - Azure Portal: Service an be launched directly from Azure Portal
  - Azure Storage Services: Directly access data in Azure Blob Storage and Azure Data Lake Store
  - Azure Active Directory: For user authentication, eliminating the need to maintain two separate sets of uses in Databricks and Azure.
  - Azure SQL DW and Azure Cosmos DB: Enables you to combine structured and unstructured data for analytics
  - Apache Kafka for HDInsight: Enables you to use Kafka as a streaming data source or sink
- Azure Billing: You get a single bill from Azure
- Azure Power BI: For rich data visualization

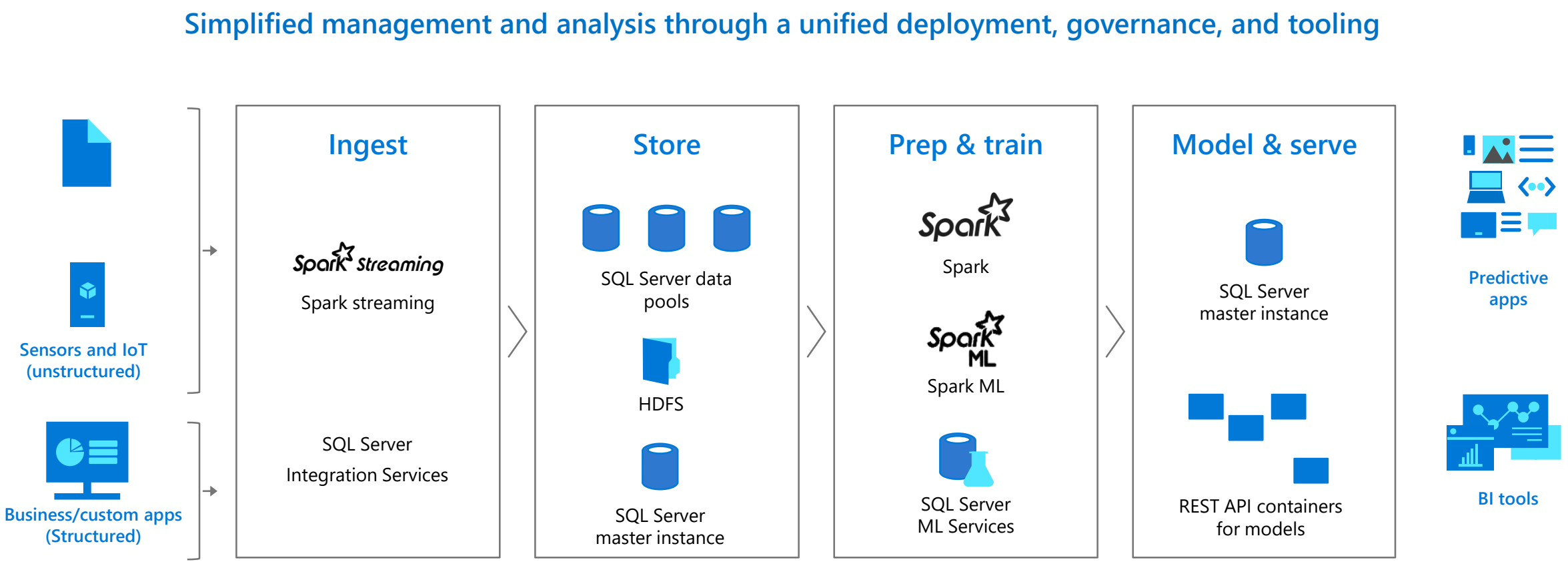
SQL Server 2019 big data clusters

SQL Server is the hub for integrating data

Easily combine across relational and non-relational data stores

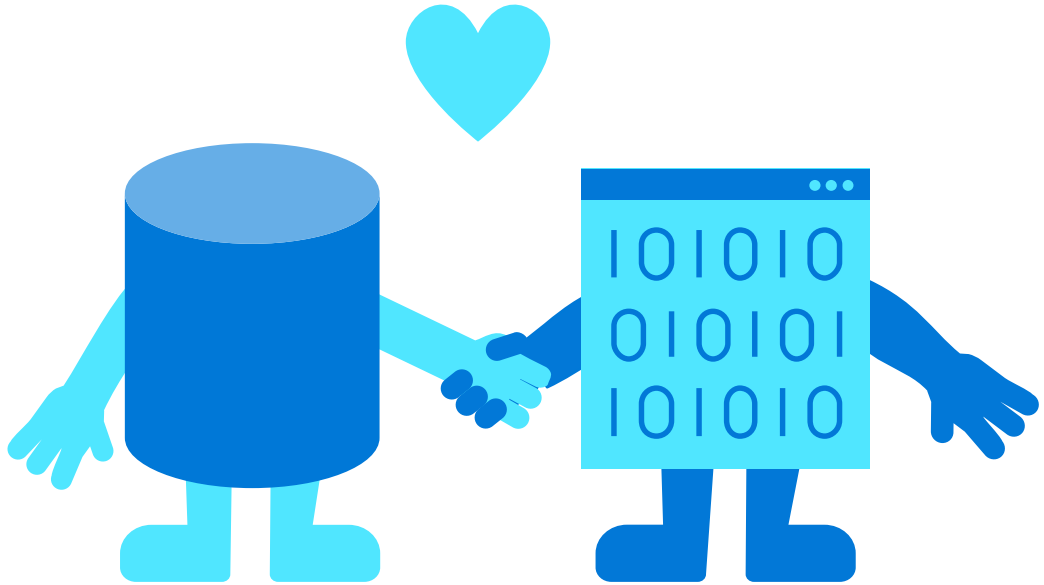


Integrate structured and unstructured data



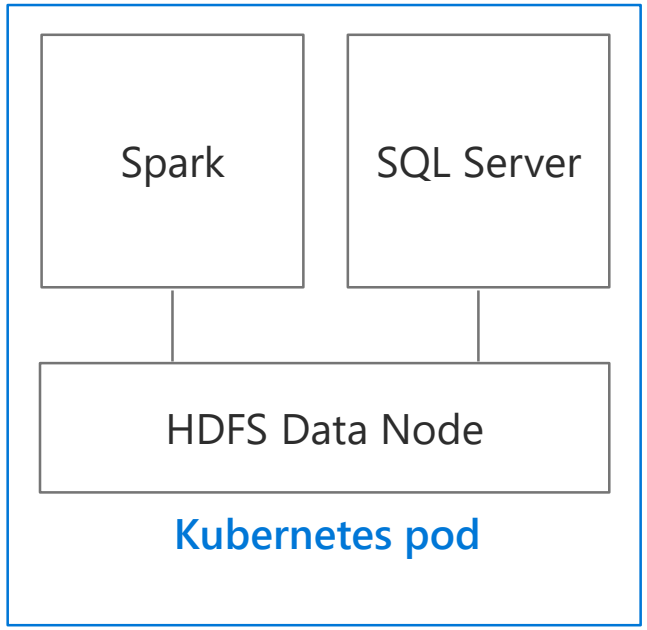
Easily deploy and manage a SQL Server + Big Data cluster

- Easily deploy and manage a Big Data cluster using Microsoft's Kubernetes-based Big Data solution built-in to SQL Server
- Hadoop Distributed File System (HDFS) storage, SQL Server relational engine, and Spark analytics are deployed as containers on Kubernetes in one easy-to manage package



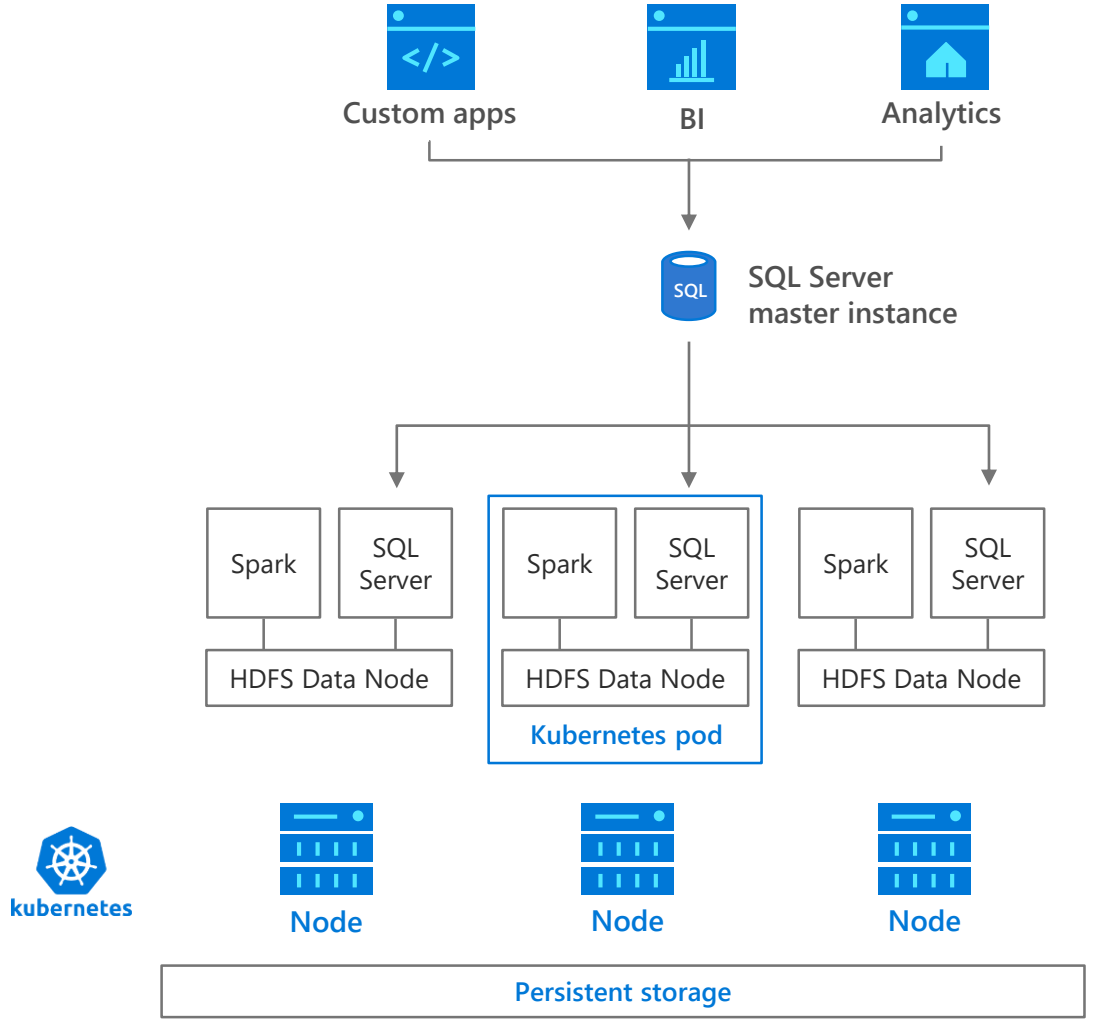
Simplified deployment with containers & Kubernetes

- A container is a standardized unit of software that includes everything needed to run it
- Kubernetes is a container hosting platform
- Benefits of containers and Kubernetes:
  - Fast to deploy
  - Self-contained – no installation required
  - Upgrades are easy because - just upload a new image
  - Scalable, multi-tenant, designed for elasticity



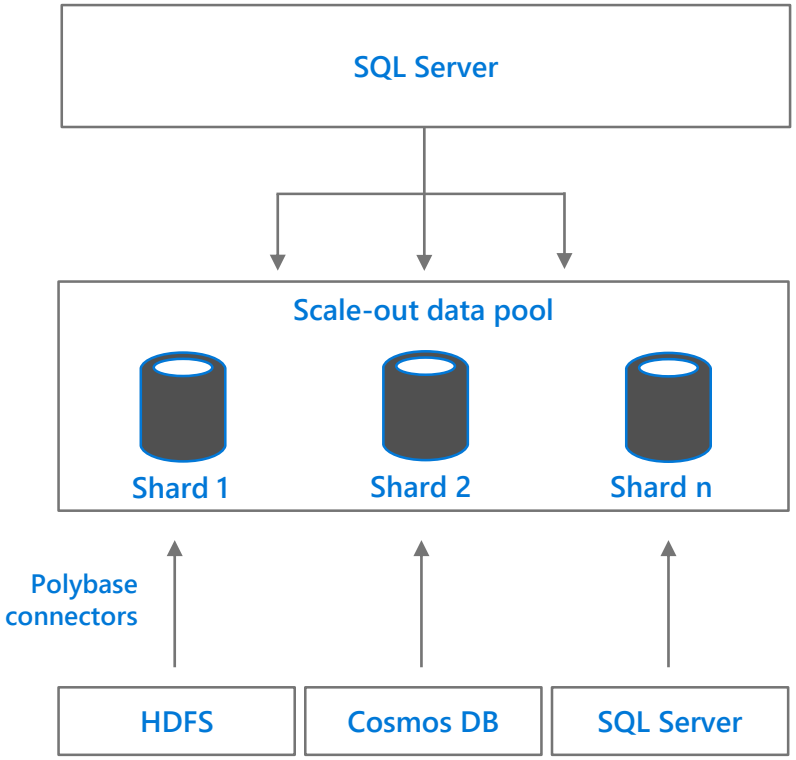
Scale Big Data on demand

- SQL Server can now read directly from HDFS files
- Elastically scale compute and storage using HDFS-based storage pools with SQL Server and Spark built in
- Apps, BI, and analytics access Big Data through the SQL Server master instance

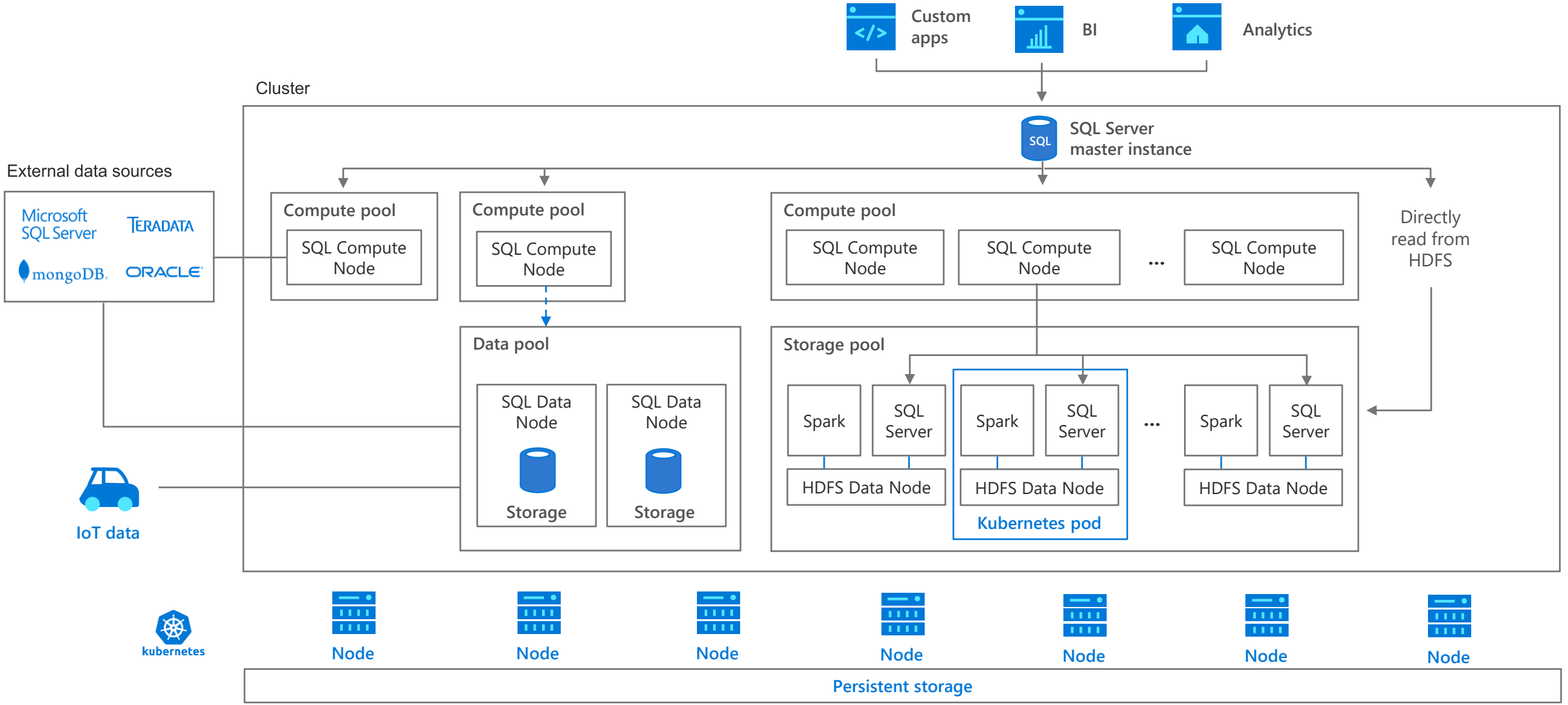


Increase performance for data virtualization

- Scale-out data pools combine and cache data from many sources for fast querying
- Scenario:
  - A global car manufacturing company wants to join data from across multiple sources including HDFS, SQL Server, and Cosmos DB
- Solution
  - Query data in relational and non-relational data stores with new PolyBase connectors
  - Create a scale-out data pool cache of combined data
  - Expose the datasets as a shared data source, without writing code to move and integrate data

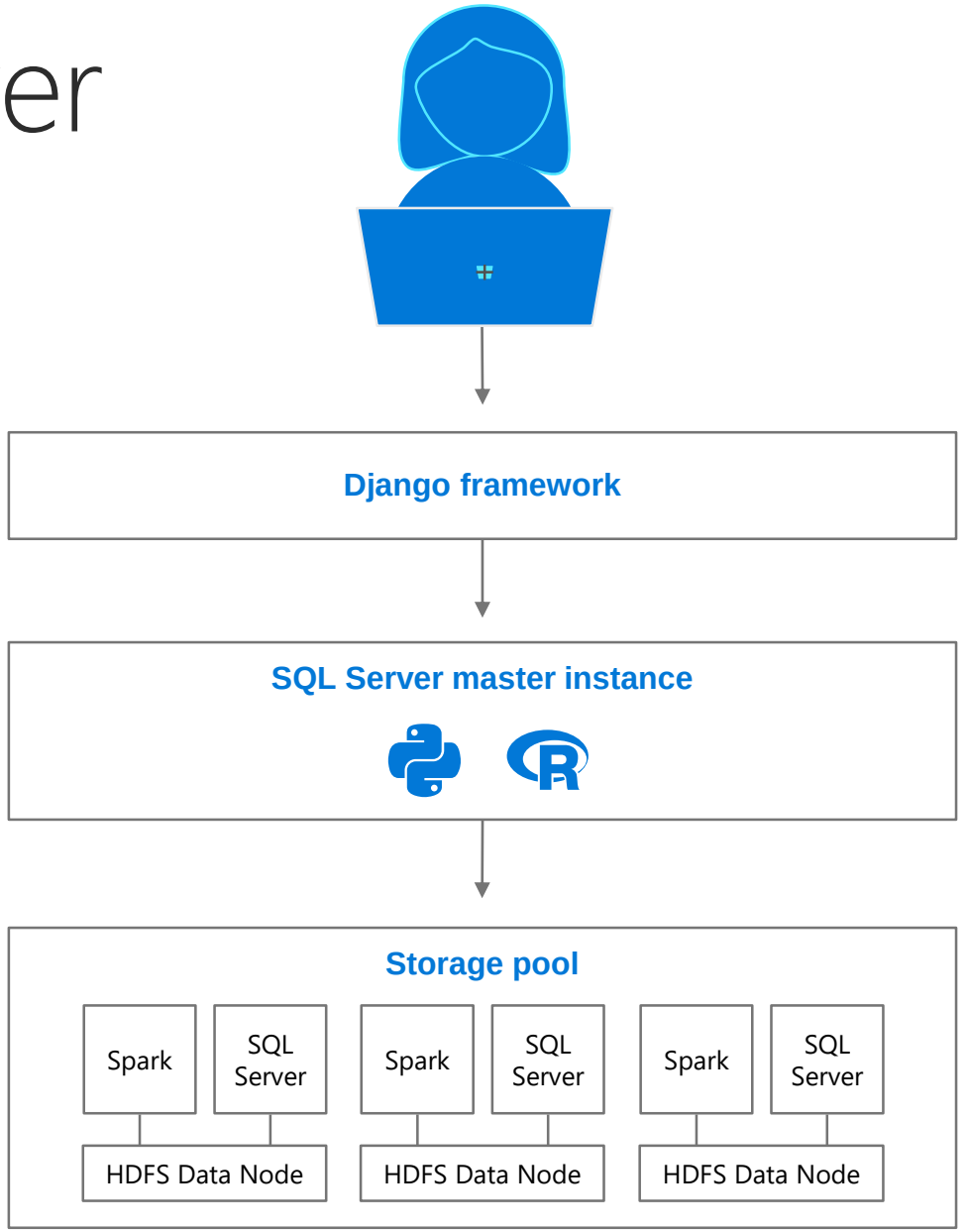


Architecture



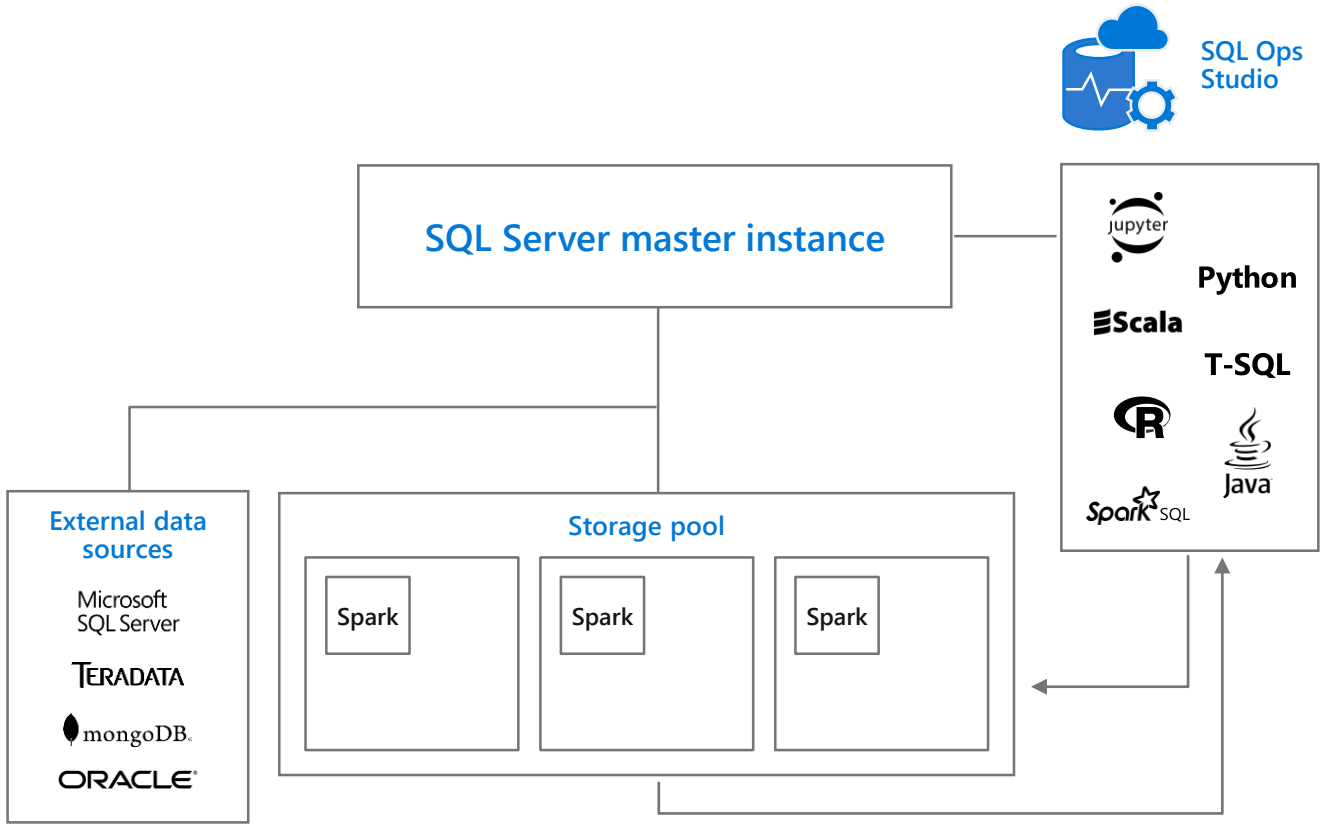
Build more intelligent apps with ML Services built into SQL Server

- Access relational and non-relational data using familiar T-SQL commands and development frameworks
- Enrich apps with data from other sources like Oracle database, Mongo DB
- Build intelligent applications with access to unstructured, high volume, and high velocity data
- Train R and Python models against Big Data stored in Hadoop and score your application data without ever leaving SQL Server
- Apply easy to use tools like Azure Data Studio and Visual Studio Code



Analyze data with a unified platform

- Data scientists can use familiar tools to analyze structured and unstructured data
  - Use Azure Data Studio notebooks run a Spark job over structured and unstructured data
  - Spark jobs can access data in SQL Server through JDBC, Tedious, etc.
  - Queries can be access data from other sources like Oracle Database and Mongo DB via external tables
  - The Spark job returns the data to the notebook



Understanding Machine Learning

Machine Learning represents a growing opportunity

Global business value derived from **AI in 2022** will reach

"Forecast: The Business Value of Artificial Intelligence, Worldwide, 2017-2025", Gartner, April 2018.

Helping you innovate across your business

Marketing	Sales	Service	Finance	Operations	Workforce
Product Recommendation	Lead Scoring	Intelligent chatbots	Financial Forecasting	**Predictive Maintenance**	Employee Insights
Customer Insights	Sales Insights	Virtual Assistants	Cash Flow Forecasting	**Demand Forecasting**	HR Insights
Churn Analytics	Dynamic Pricing	Waiting line optimization	Risk Management	Quality Assurance	Resource Planning

Understanding Machine Learning

Name	*Amount*	*Fraudulent*
Smith	$2,600.45	No
Potter	$2,294.58	Yes
Peters	$1,003.30	Yes
Adams	$8,488.32	No

What's the pattern for fraudulent transactions?

Understanding Machine Learning

Name	*Amount*	*Origin*	*Transaction*	*Amount*	*Fraudulent*
Smith	$2,600.45	USA	USA	22	No
Potter	$2,294.58	USA	RUS	29	Yes
Peters	$1,003.30	USA	RUS	25	Yes
Adams	$8,488.32	FRA	USA	64	No
Pali	$200.12	AUS	JAP	58	No
Jones	$3,250.11	USA	RUS	43	No
Hanford	$8,156.20	USA	RUS	27	Yes
Marx	$7,475.11	UK	GER	32	No
Norse	$540.00	USA	RUS	27	No
Edson	$7,475.11	USA	RUS	20	Yes

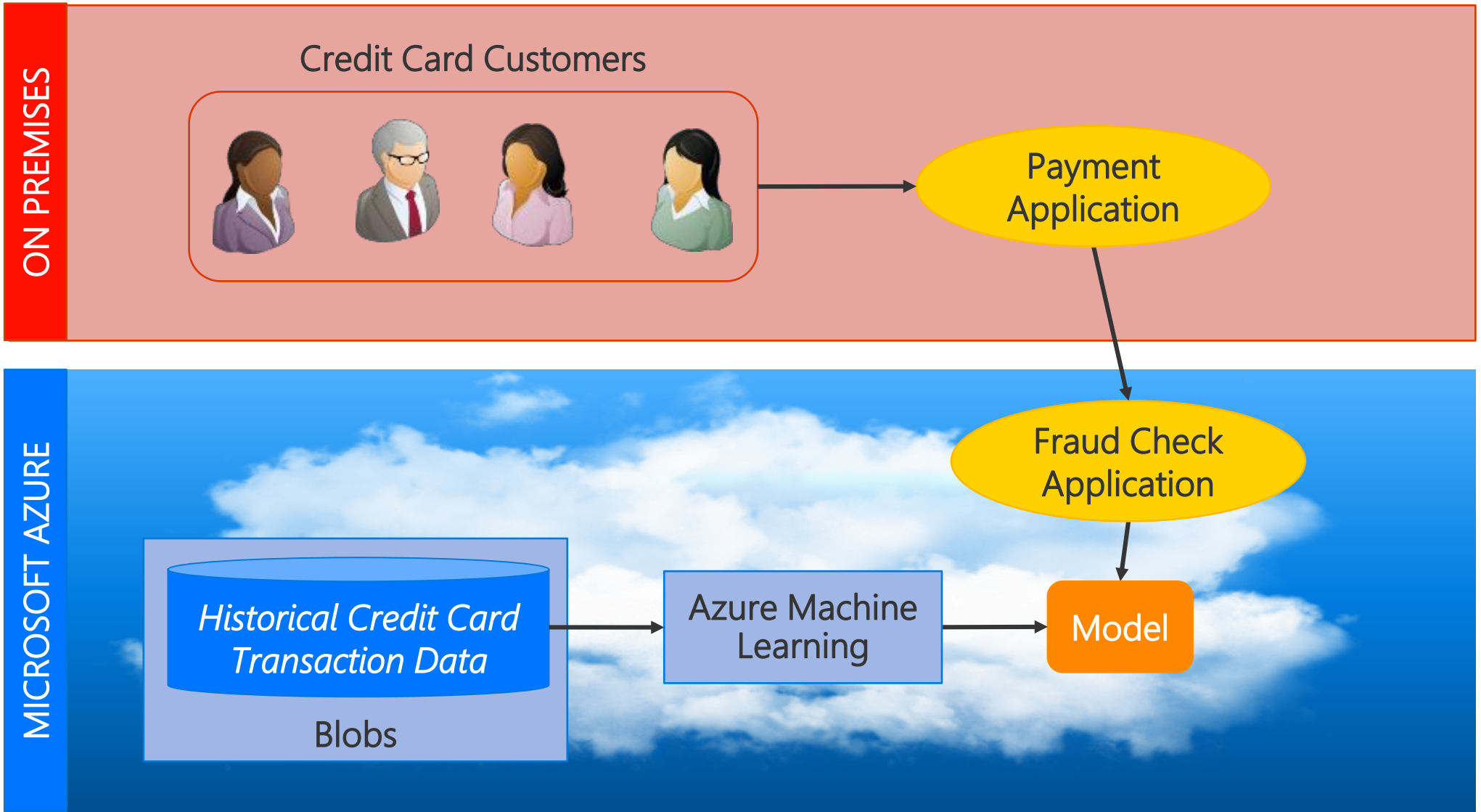
What's the pattern for fraudulent transactions?

Machine Learning

- Some problems can not be solved algorithmically
- Attempts algorithmic solutions to many other problems would be not only very complicated, but also highly ineffective
- Instead of writing a program by hand for each specific task, we collect lots of examples that specify the correct output for a given input
- A machine learning algorithm then takes these examples and produces a program that does the job

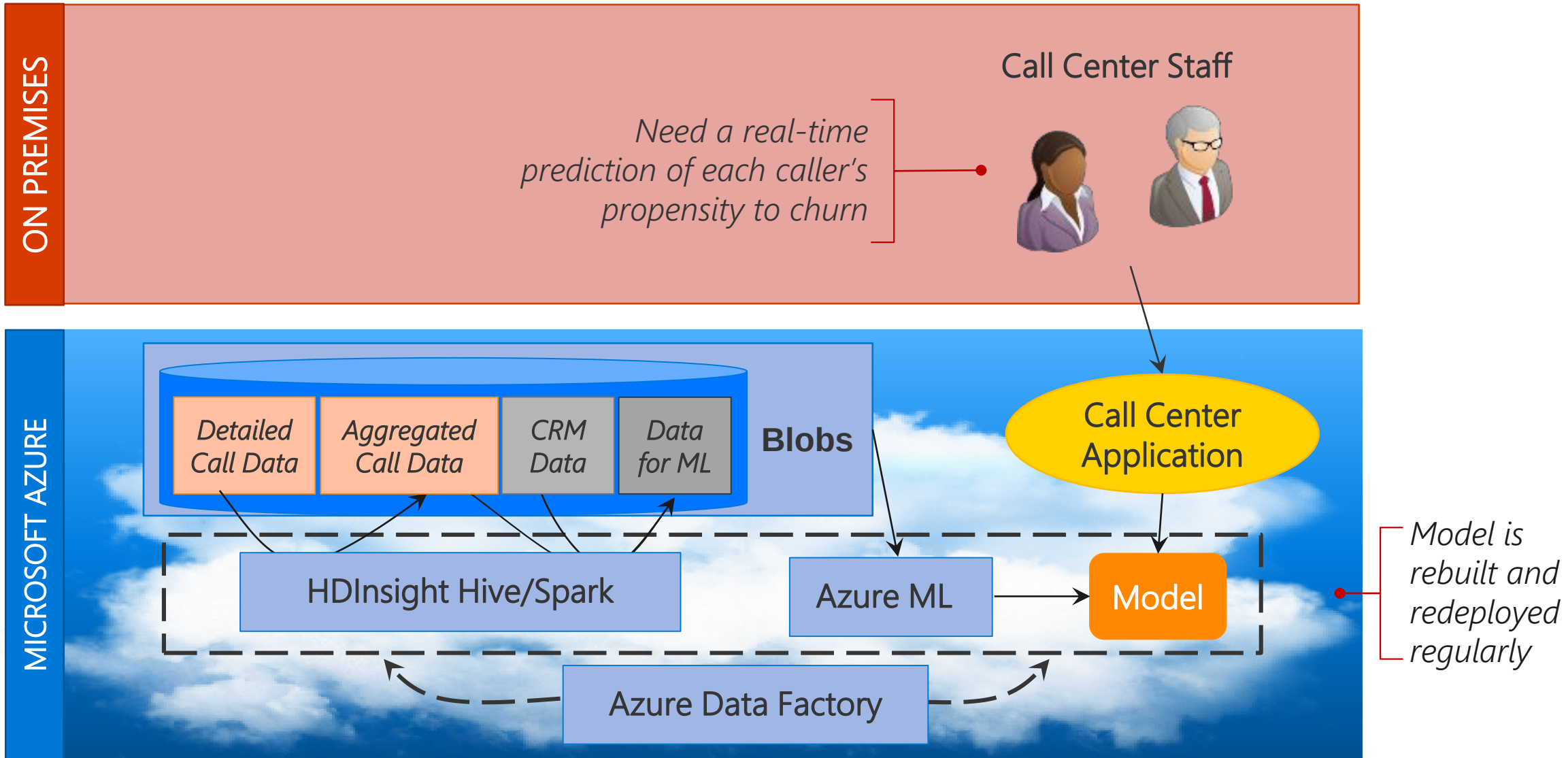
Using Historical Data to Make Better Decisions

Scenario

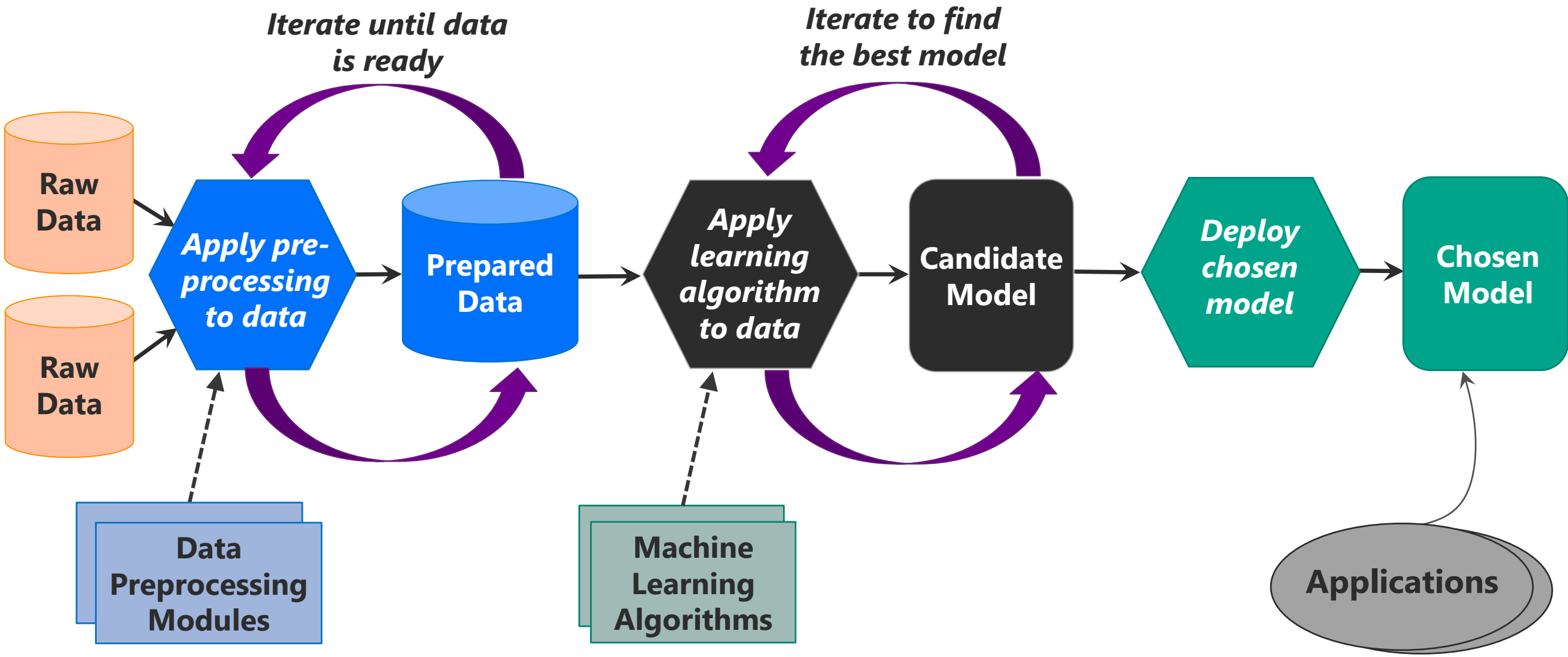


Anticipating Customer Behavior

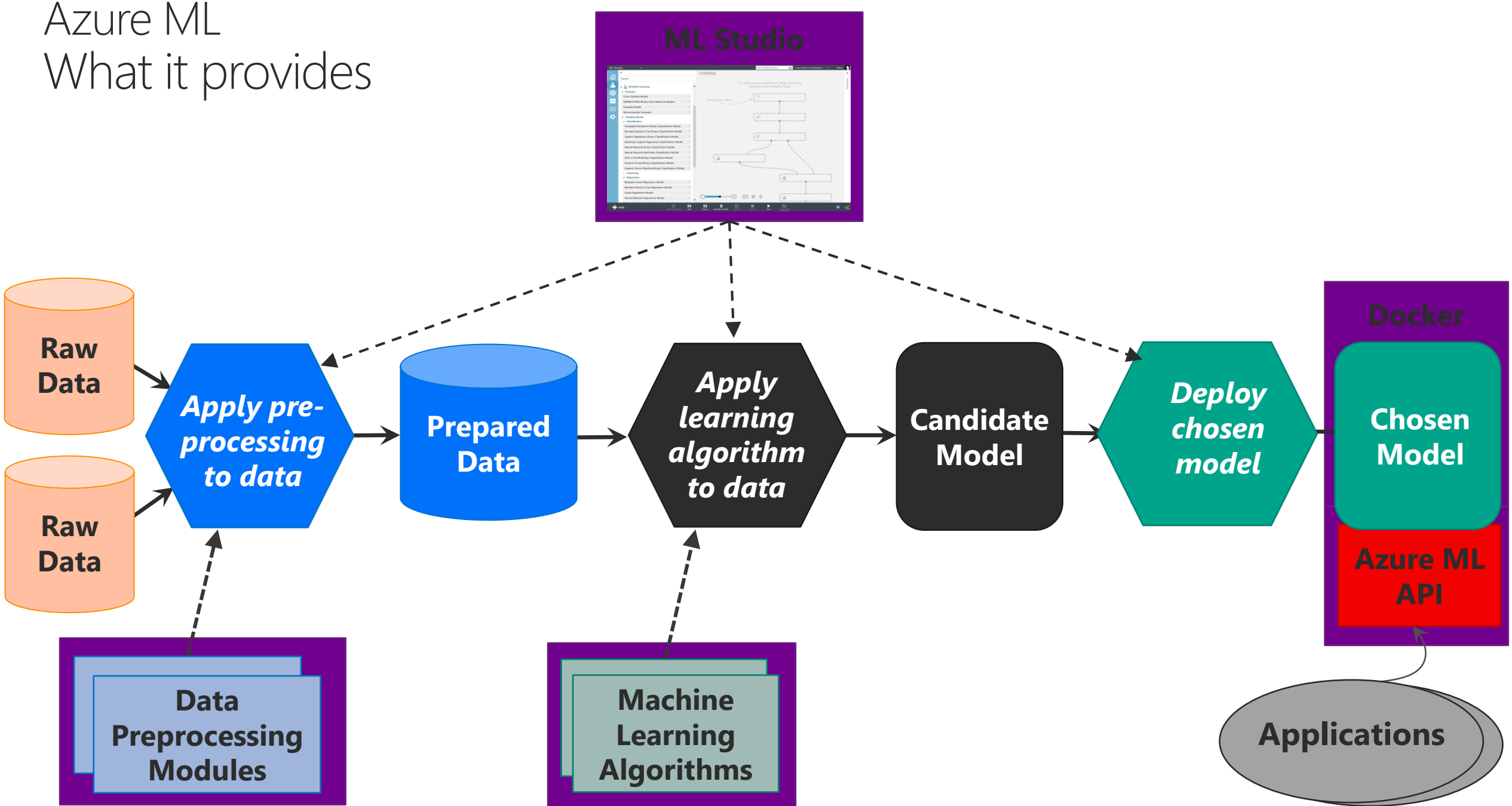
Scenario



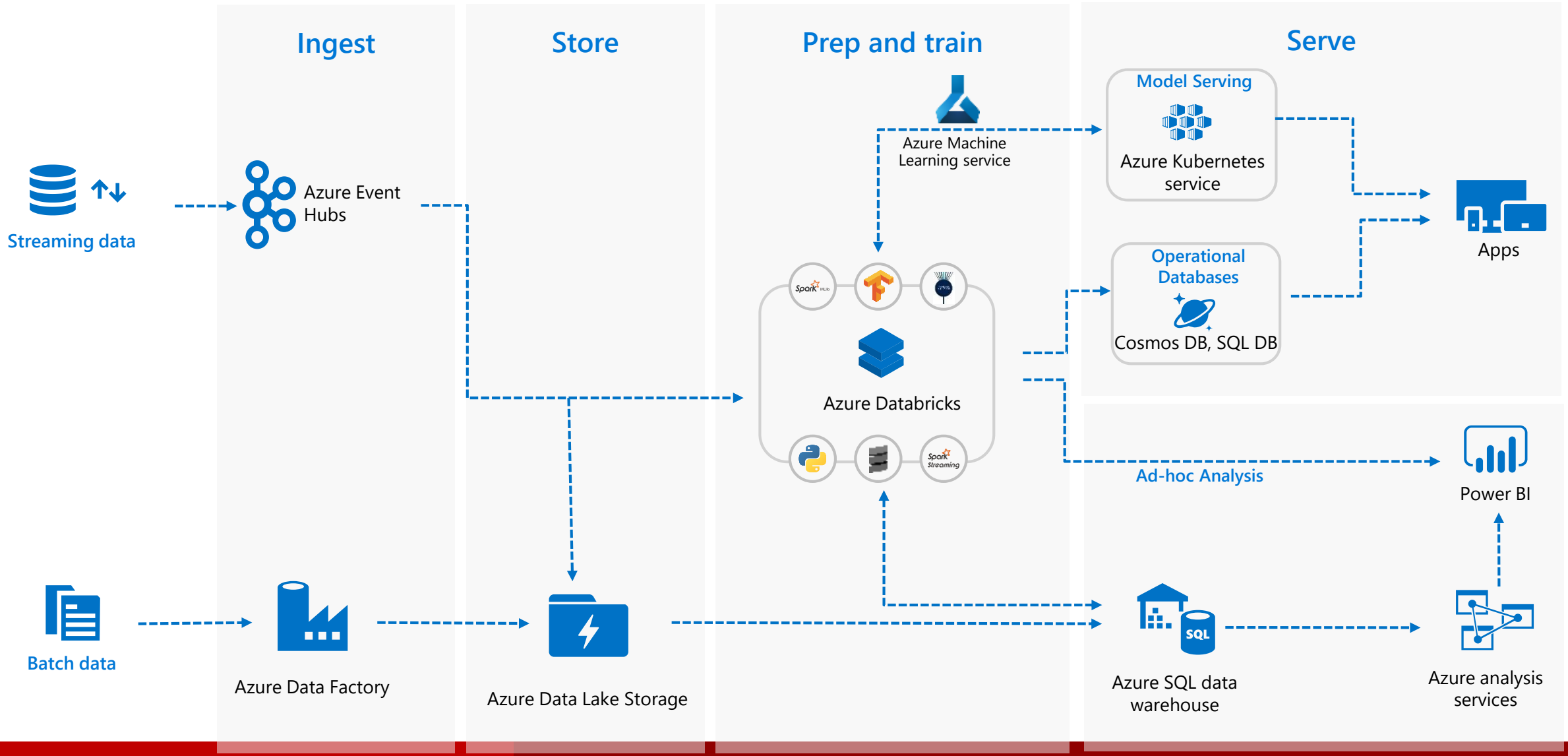
Machine Learning
Illustrating the process



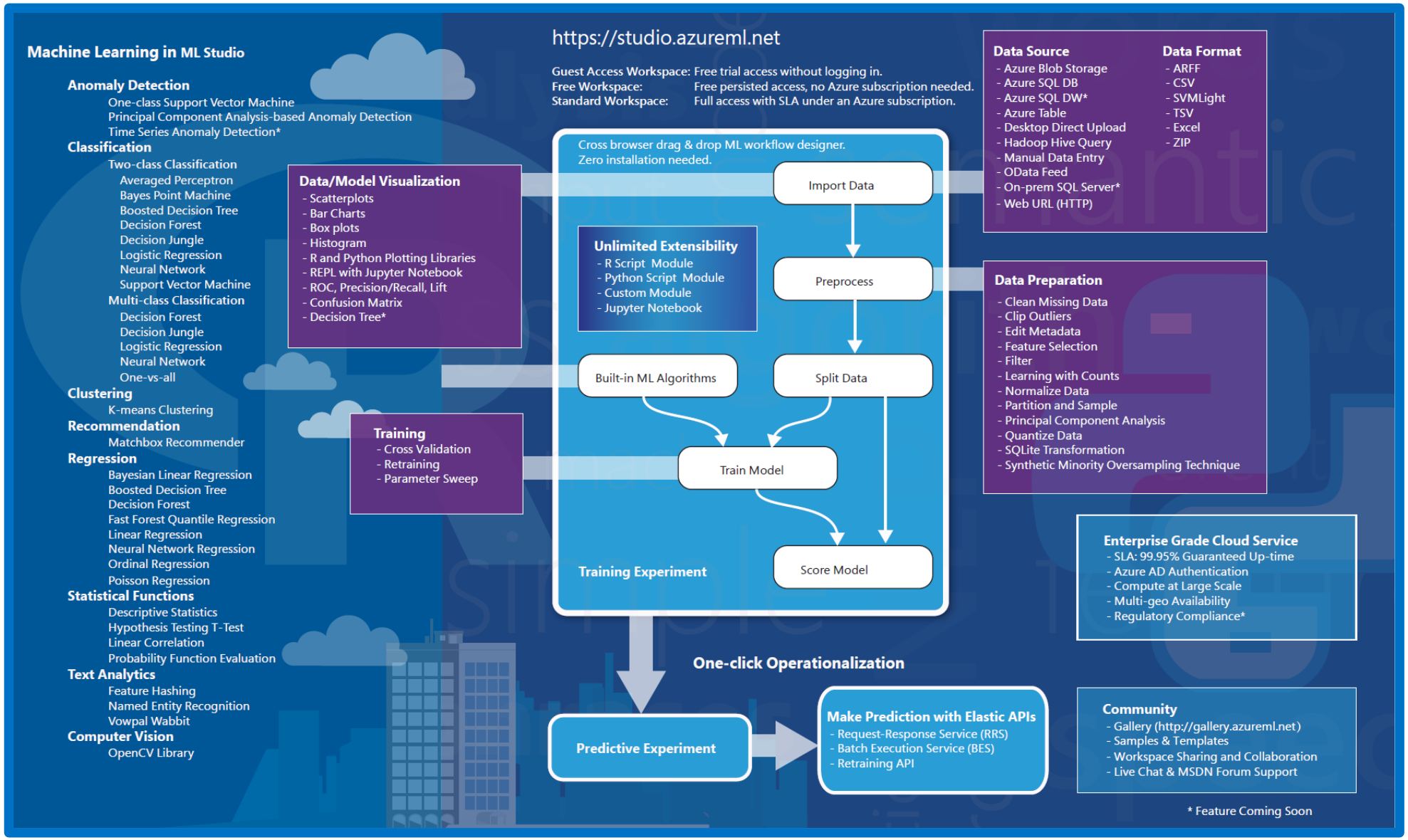
Azure ML
What it provides



Recommended architecture to build e2e ML solutions



Azure Machine Learning Studio



Designing a solution using Azure Machine Learning

Getting Data In and Out of Azure ML

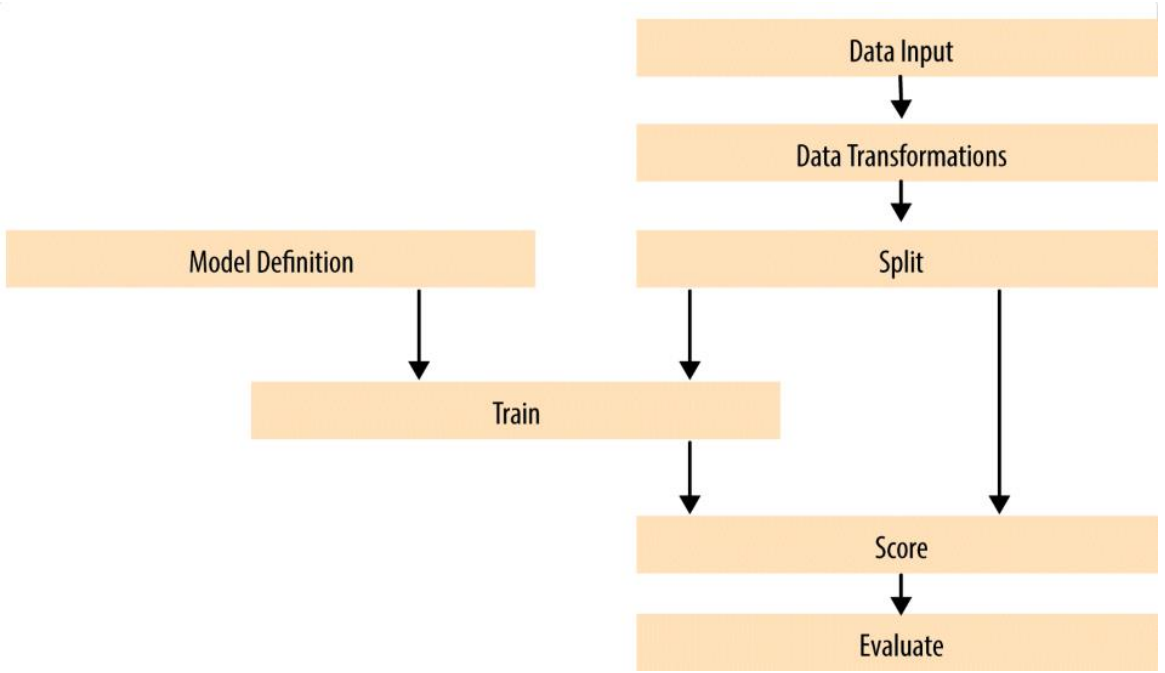
- Azure ML supports several data I/ O options, including:
  - Web services
  - HTTP connections
  - Azure SQL tables
  - Azure Blob storage
  - CSV files
  - SQL Server connections (available only in PRO)
- Data I/O at scale is supported by the AzureML Reader and Writer modules
  - Reader and Writer modules provide an interface with Cortana data storage component

Modules

- Azure ML provides a wide range of modules for data transformation, machine learning, and model evaluation
- Most native Azure ML modules are computationally efficient and scalable
- As a general rule, these native modules should be your first choice
- In the Azure ML Studio, input ports are located above module icons, and output ports are located below module icons

Workflow

- Data input can come from a variety of interfaces, including web services, HTTP connections, Azure SQL, and Hive Query
- Transformations of the data can be performed using a combination of native Azure ML modules and the R language
- A Model Definition module defines the model type and properties
  - On the left hand pane of the Studio you will see numerous choices for models
  - The parameters of the model are set in the properties pane
  - R model training and scoring scripts can be provided in a Create R Model module
- The Training module trains the model
- Training of the model is scored in the Score module and performance summary statistics are computed in the Evaluate module



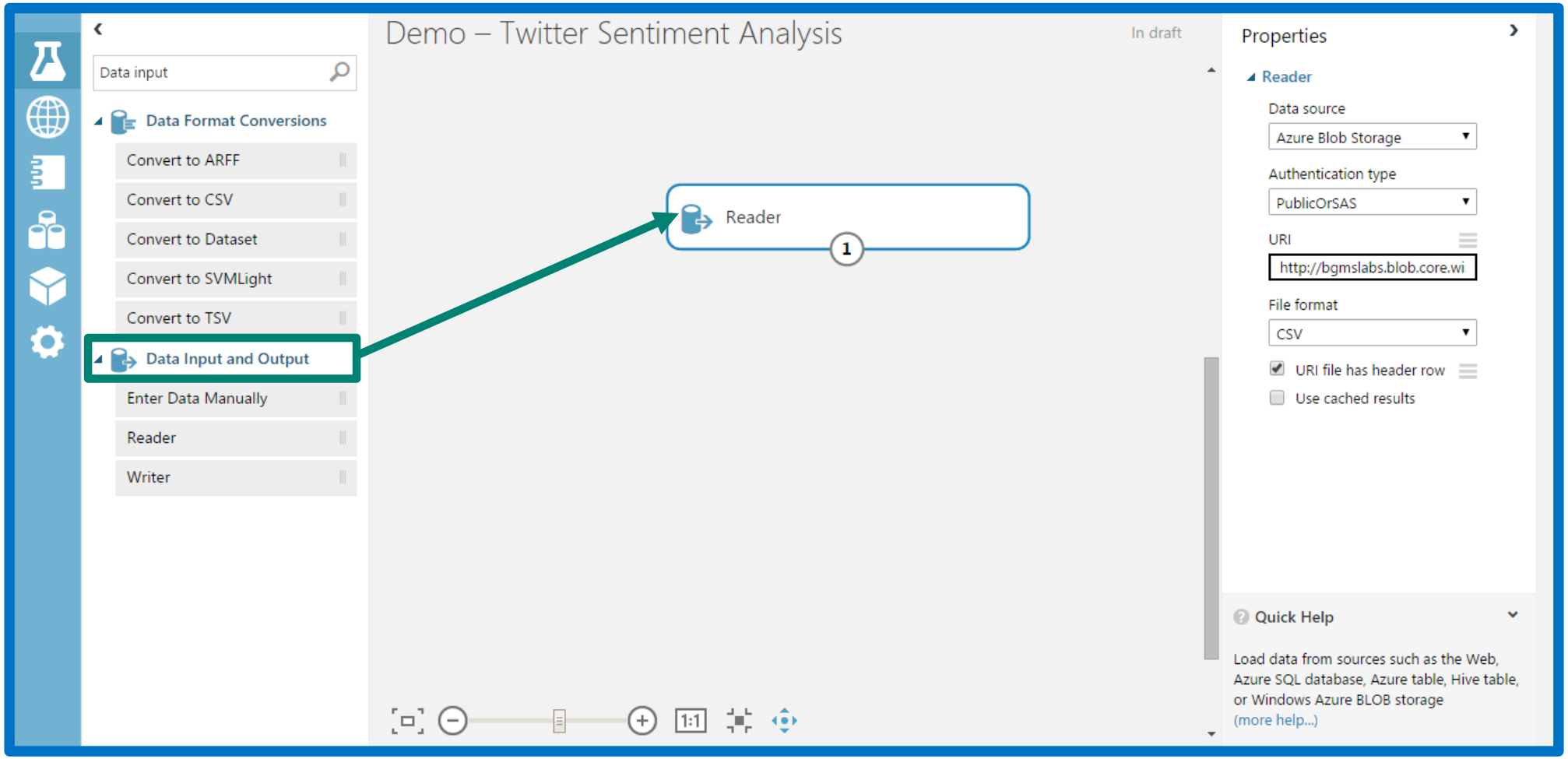
Predictive Web Services

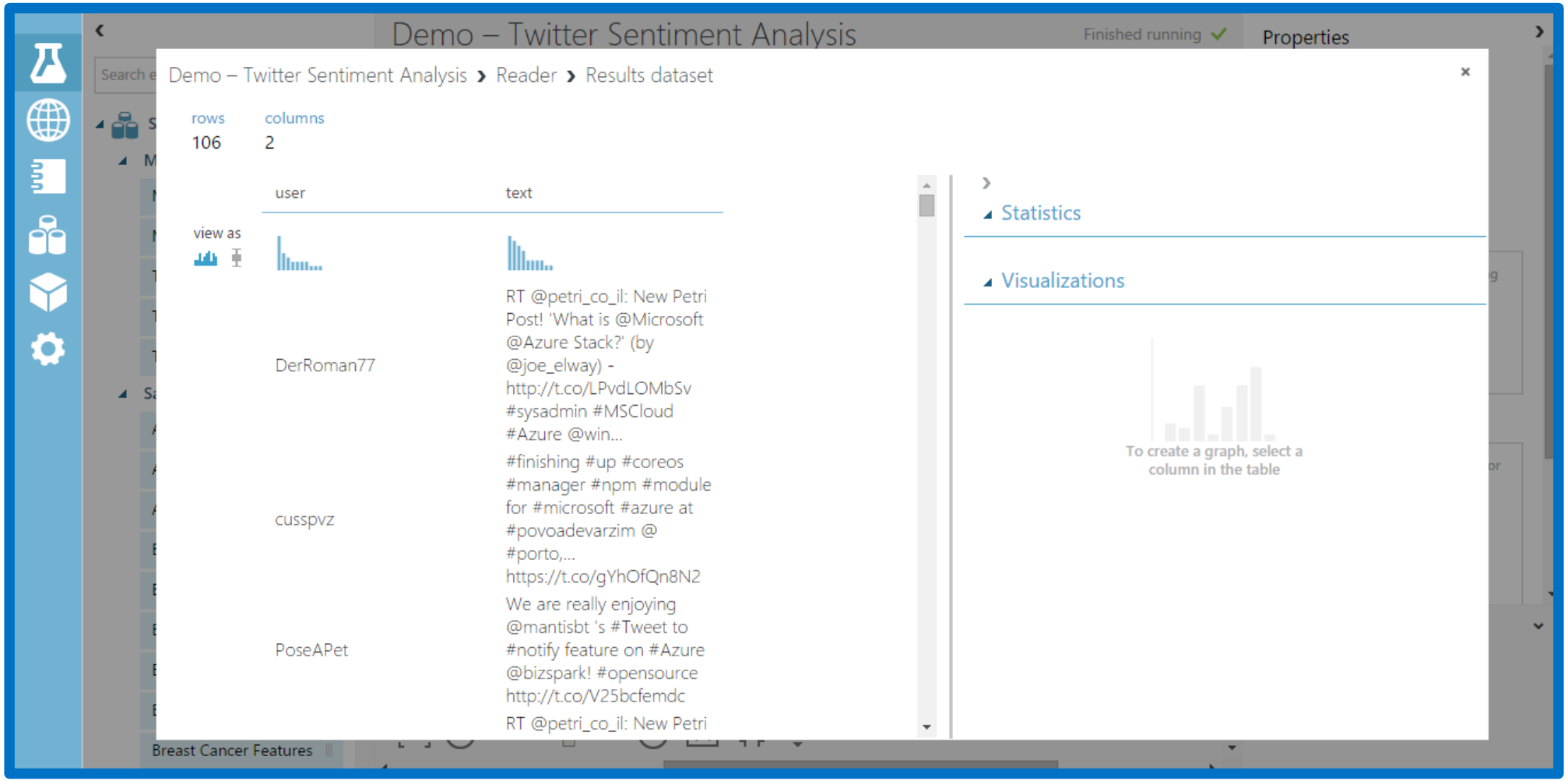
- Predictive Web Services are available in the *WEB SERVICES* section
  - In the *Configuration* tab, you can specify the name and description of Web services and descriptions of the individual parameters
  - Much more options can be found in *DASHBOARD* tab
  - You can also test your Web service

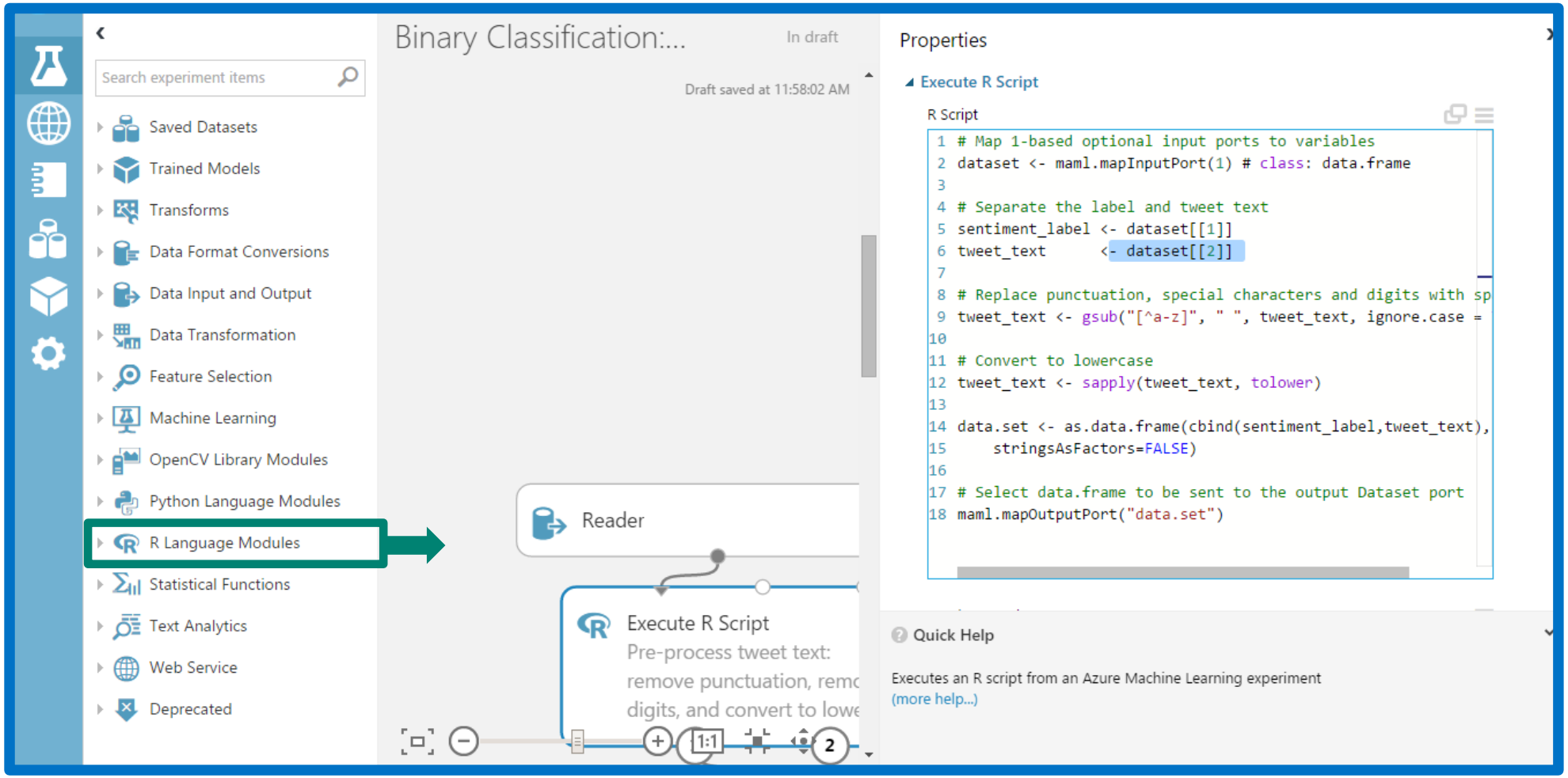
Ad-hoc predictive queries

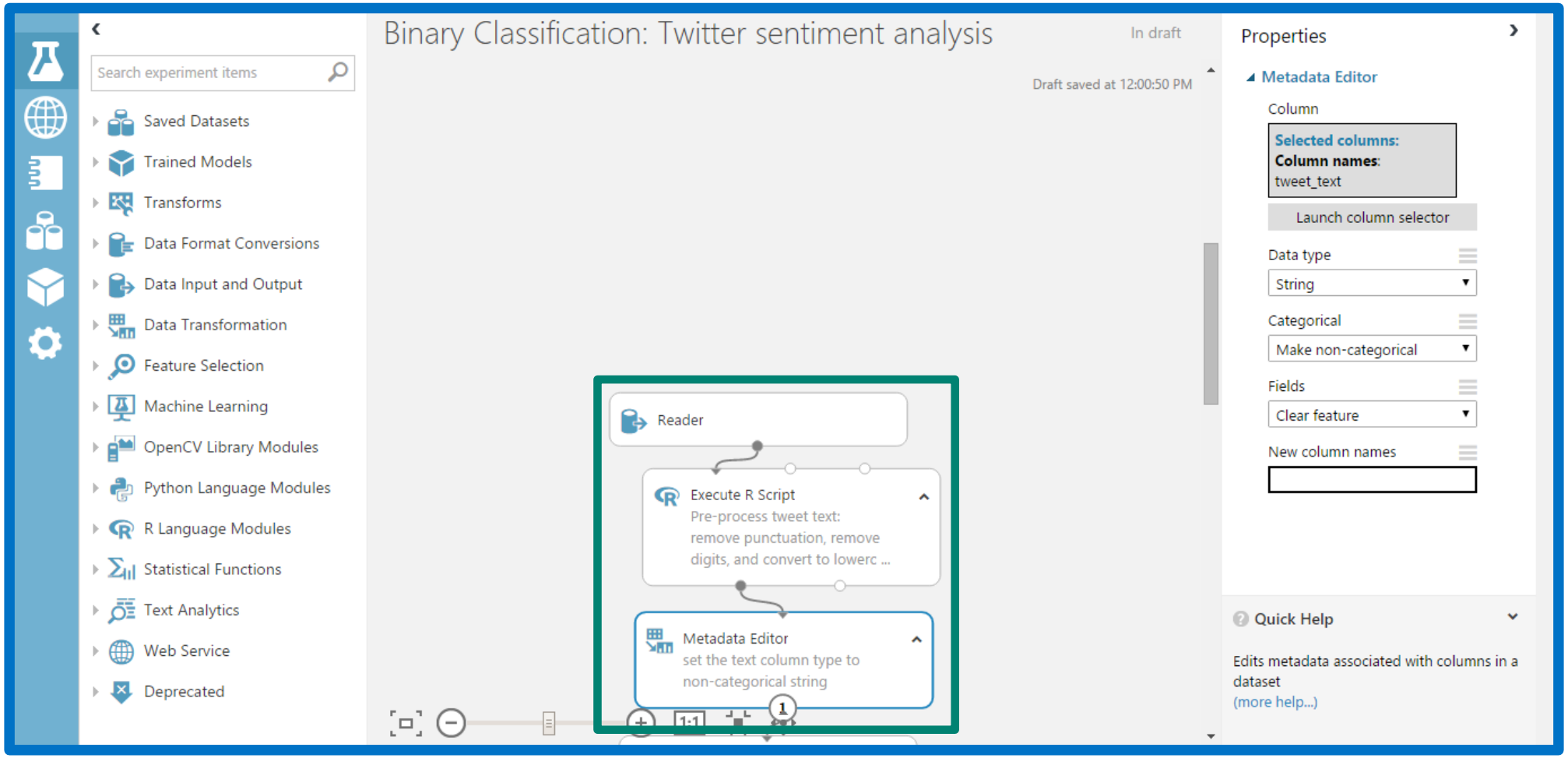
- Ad-hoc predictive queries are executed one by one
- When you click the *REQUEST / RESPONSE* link, you will see detailed documentation of this interface contains, among other thigs, sample code in C#, Python and R, and ready-to-use Web application template

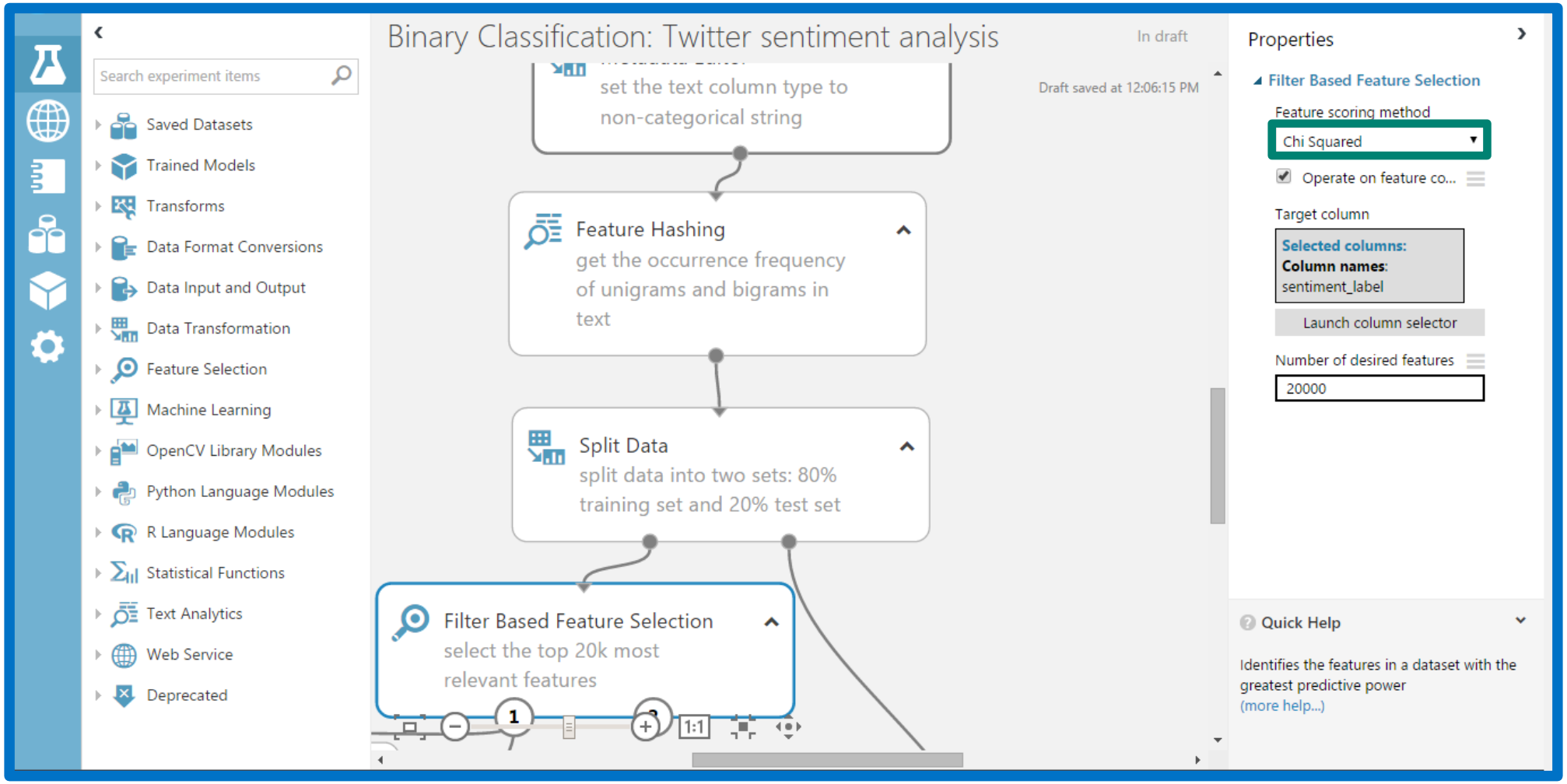
Demo
Deploying an API: Twitter sentiment analysis

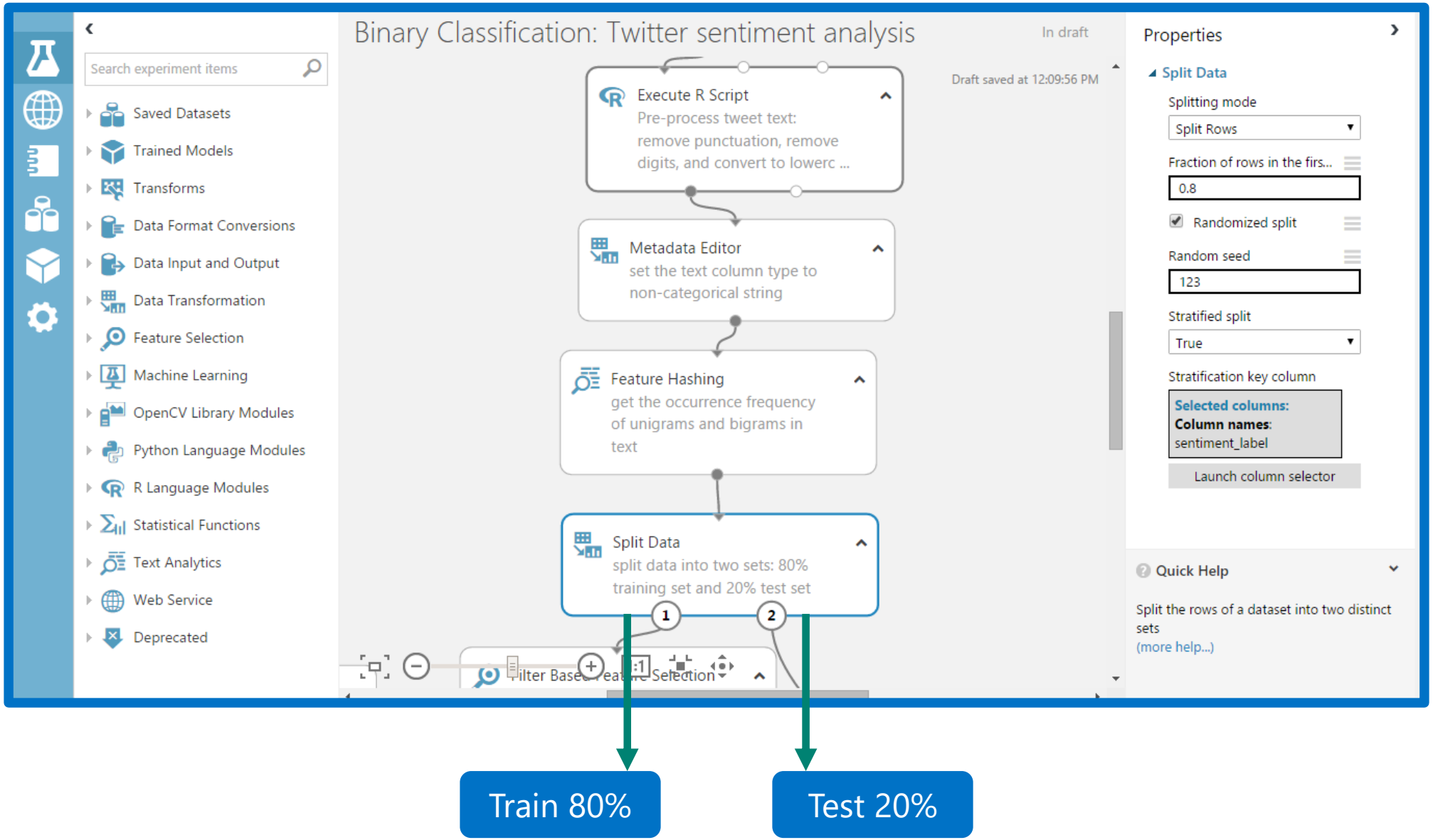

Demo – Twitter Sentiment Analysis
In draft
Data input
Data Format Conversions
Convert to ARFF
Convert to CSV
Convert to Dataset
Convert to SVMLight
Convert to TSV
Data Input and Output
Enter Data Manually
Reader
Writer
Reader
1
Properties
Reader
Data source
Azure Blob Storage
Authentication type
PublicOrSAS
URI
http://bgmslabs.blob.core.wi
File format
CSV
URI file has header row
Use cached results
Quick Help
Load data from sources such as the Web, Azure SQL database, Azure table, Hive table, or Windows Azure BLOB storage
(more help...)

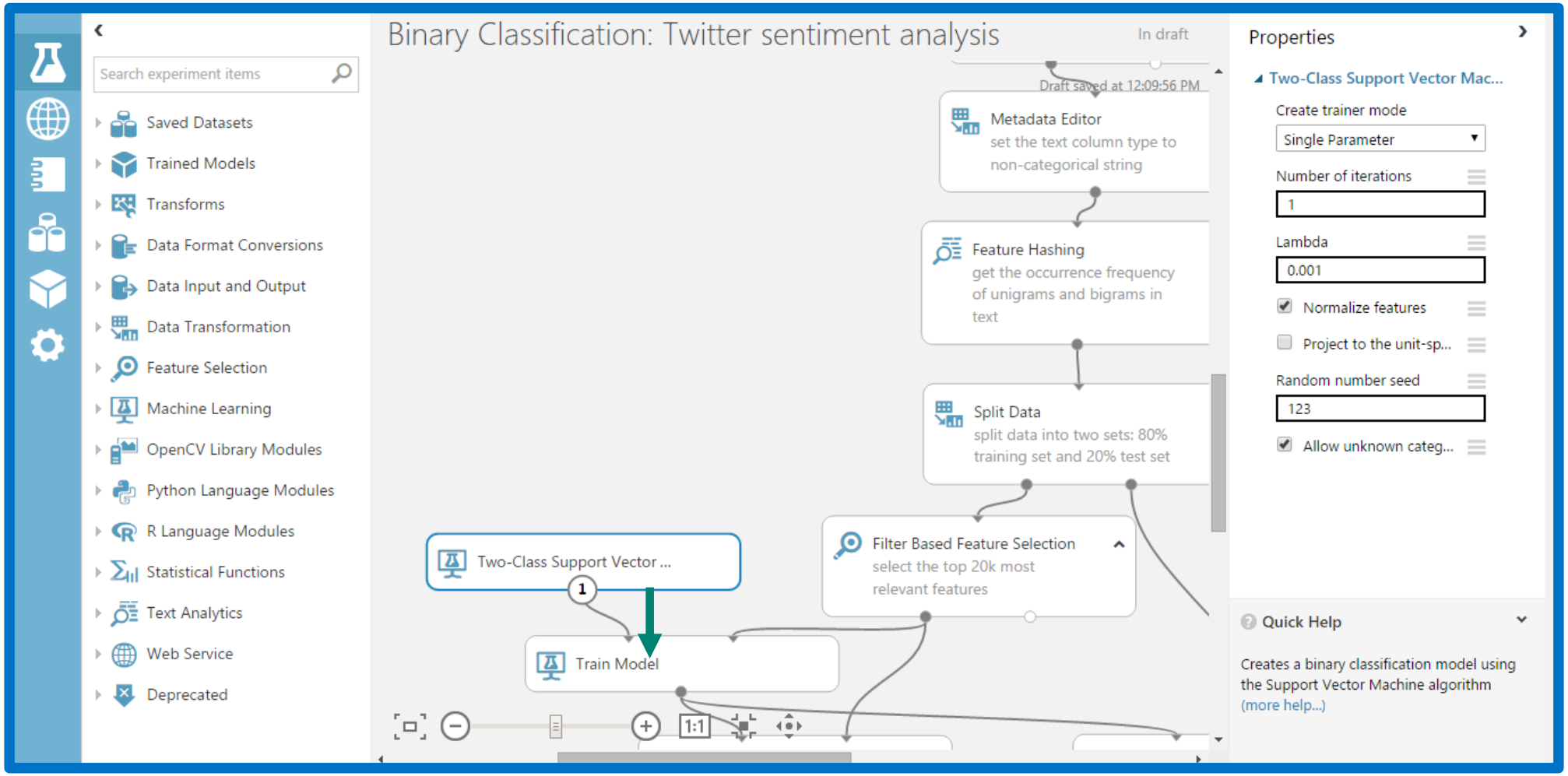

Demo – Twitter Sentiment Analysis
Finished running
Properties
Demo – Twitter Sentiment Analysis › Reader › Results dataset
rows
106
columns
2
user
text
view as
DerRoman77
RT @petri_co_il: New Petri Post! 'What is @Microsoft @Azure Stack?' (by @joe_elway) - http://t.co/LPvdLOMbSv #sysadmin #MSCloud #Azure @win...
cusspvz
#finishing #up #coreos #manager #npm #module for #microsoft #azure at #povoadevarzim @ #porto,... https://t.co/gYhOfQn8N2
PoseAPet
We are really enjoying @mantisbt 's #Tweet to #notify feature on #Azure @bizspark! #opensource http://t.co/V25bcfemdc
RT @petri_co_il: New Petri
Statistics
Visualizations
To create a graph, select a column in the table
Breast Cancer Features

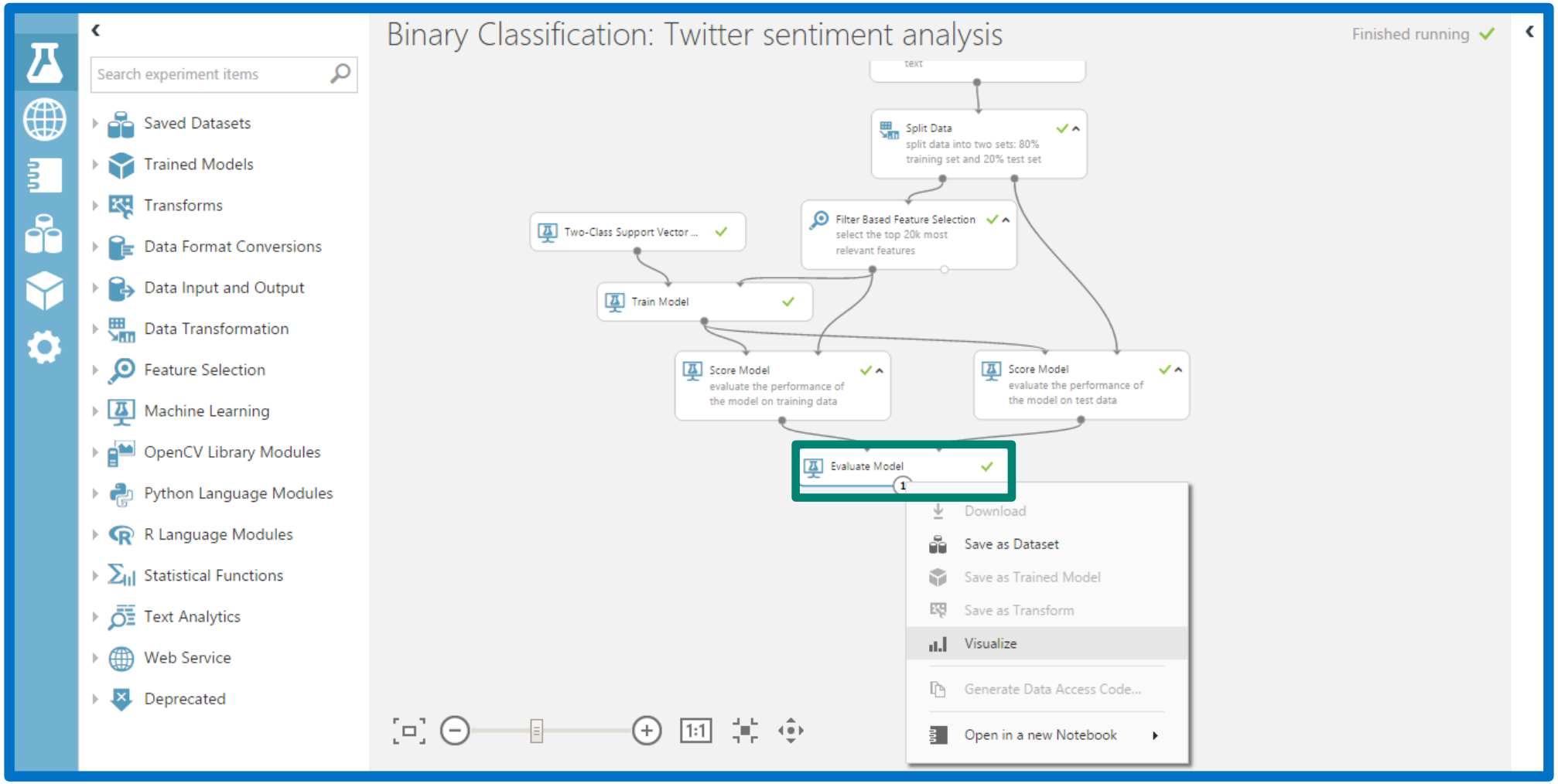

Search experiment items
Saved Datasets
Trained Models
Transforms
Data Format Conversions
Data Input and Output
Data Transformation
Feature Selection
Machine Learning
OpenCV Library Modules
Python Language Modules
R Language Modules
Statistical Functions
Text Analytics
Web Service
Deprecated
Binary Classification:...
In draft
Draft saved at 11:58:02 AM
Reader
Execute R Script
Pre-process tweet text:
remove punctuation, remo
digits, and convert to lowe
Properties
Execute R Script
R Script
1 # Map 1-based optional input ports to variables
2 dataset <- maml.mapInputPort(1) # class: data.frame
3
4 # Separate the label and tweet text
5 sentiment_label <- dataset[[1]]
6 tweet_text <- dataset[[2]]
7
8 # Replace punctuation, special characters and digits with sp
9 tweet_text <- gsub("[^a-z]", " ", tweet_text, ignore.case =
10
11 # Convert to lowercase
12 tweet_text <- sapply(tweet_text, tolower)
13
14 data.set <- as.data.frame(cbind(sentiment_label,tweet_text),
15 stringsAsFactors=FALSE)
16
17 # Select data.frame to be sent to the output Dataset port
18 maml.mapOutputPort("data.set")
Quick Help
Executes an R script from an Azure Machine Learning experiment
(more help...)

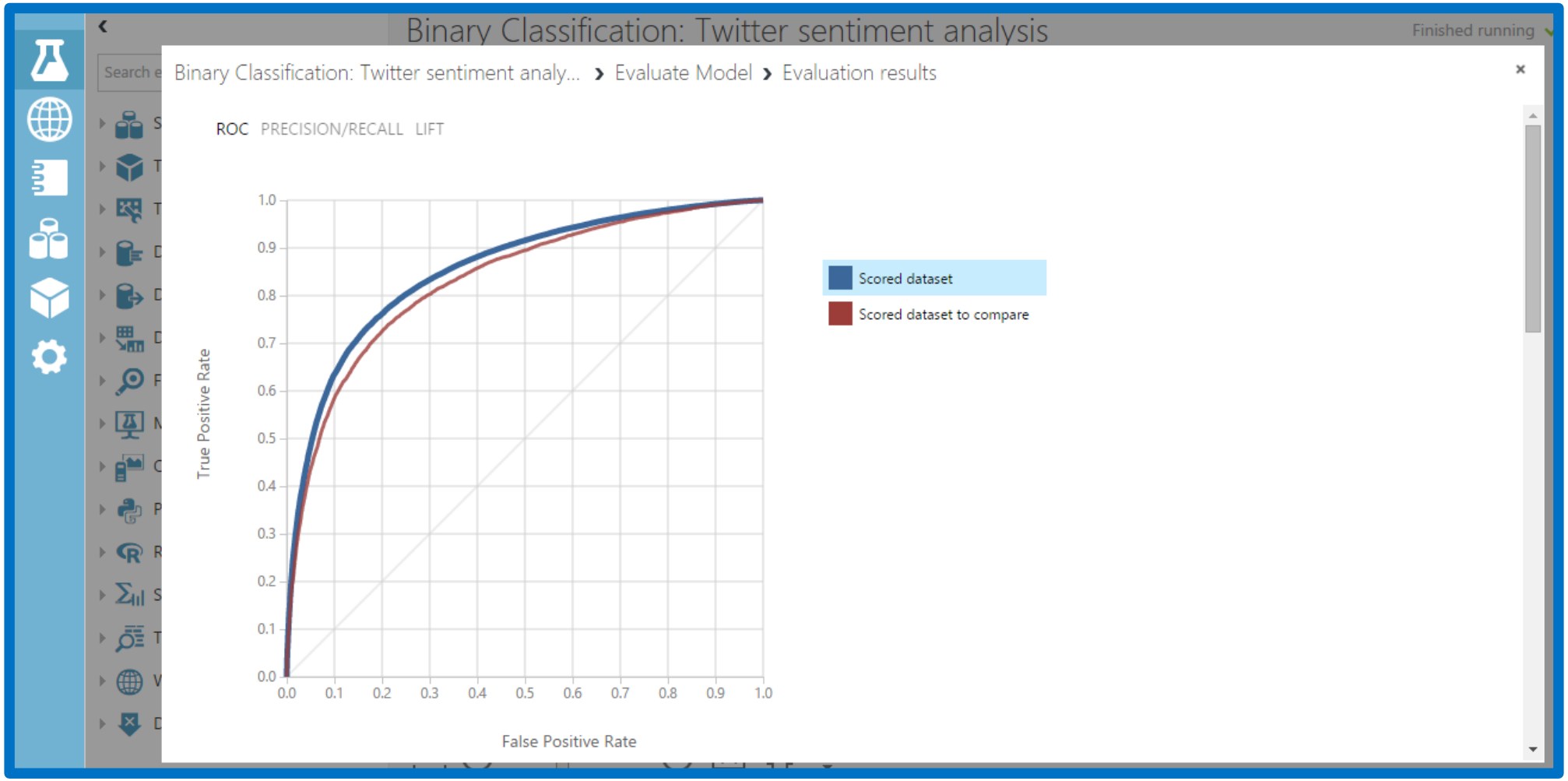

Binary Classification: Twitter sentiment analysis
In draft
Draft saved at 12:00:50 PM
Search experiment items
Saved Datasets
Trained Models
Transforms
Data Format Conversions
Data Input and Output
Data Transformation
Feature Selection
Machine Learning
OpenCV Library Modules
Python Language Modules
R Language Modules
Statistical Functions
Text Analytics
Web Service
Deprecated
Reader
Execute R Script
Pre-process tweet text: remove punctuation, remove digits, and convert to lowerc ...
Metadata Editor
set the text column type to non-categorical string
Properties
Metadata Editor
Column
Selected columns:
Column names:
tweet_text
Launch column selector
Data type
String
Categorical
Make non-categorical
Fields
Clear feature
New column names
Quick Help
Edits metadata associated with columns in a dataset
(more help...)

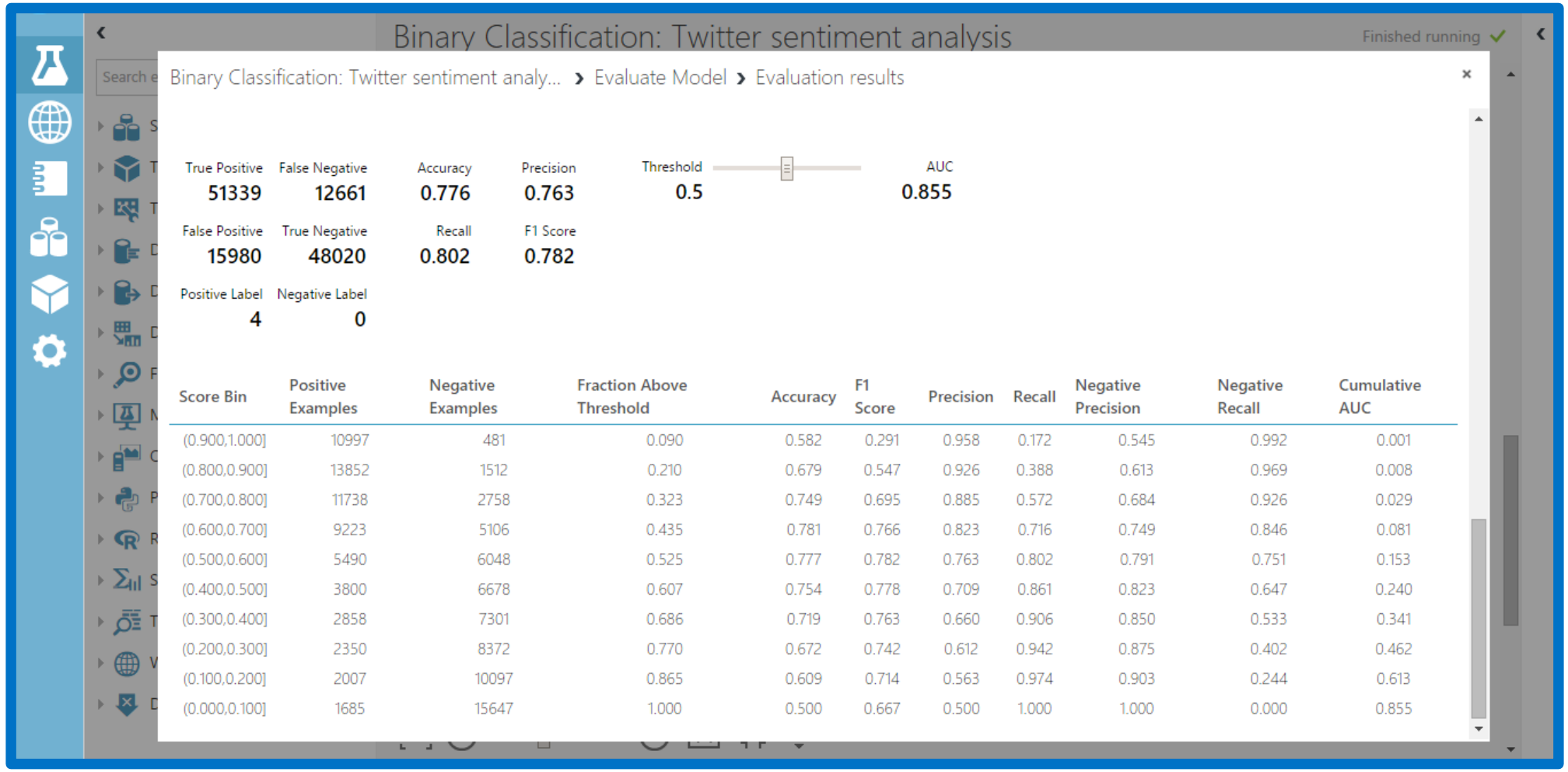

Search experiment items
Saved Datasets
Trained Models
Transforms
Data Format Conversions
Data Input and Output
Data Transformation
Feature Selection
Machine Learning
OpenCV Library Modules
Python Language Modules
R Language Modules
Statistical Functions
Text Analytics
Web Service
Deprecated
Binary Classification: Twitter sentiment analysis
In draft
Draft saved at 12:06:15 PM
set the text column type to non-categorical string
Feature Hashing
get the occurrence frequency of unigrams and bigrams in text
Split Data
split data into two sets: 80% training set and 20% test set
Filter Based Feature Selection
select the top 20k most relevant features
Properties
Filter Based Feature Selection
Feature scoring method
Chi Squared
Operate on feature co...
Target column
Selected columns:
Column names:
sentiment_label
Launch column selector
Number of desired features
20000
Quick Help
Identifies the features in a dataset with the greatest predictive power
(more help...)


Binary Classification: Twitter sentiment analysis
In draft
Draft saved at 12:09:56 PM
Search experiment items
Saved Datasets
Trained Models
Transforms
Data Format Conversions
Data Input and Output
Data Transformation
Feature Selection
Machine Learning
OpenCV Library Modules
Python Language Modules
R Language Modules
Statistical Functions
Text Analytics
Web Service
Deprecated
Execute R Script
Pre-process tweet text: remove punctuation, remove digits, and convert to lowerc ...
Metadata Editor
set the text column type to non-categorical string
Feature Hashing
get the occurrence frequency of unigrams and bigrams in text
Split Data
split data into two sets: 80% training set and 20% test set
1
2
Properties
Split Data
Splitting mode
Split Rows
Fraction of rows in the firs...
0.8
Randomized split
Random seed
123
Stratified split
True
Stratification key column
Selected columns:
Column names:
sentiment_label
Launch column selector
Quick Help
Split the rows of a dataset into two distinct sets
(more help...)
Train 80%
Test 20%

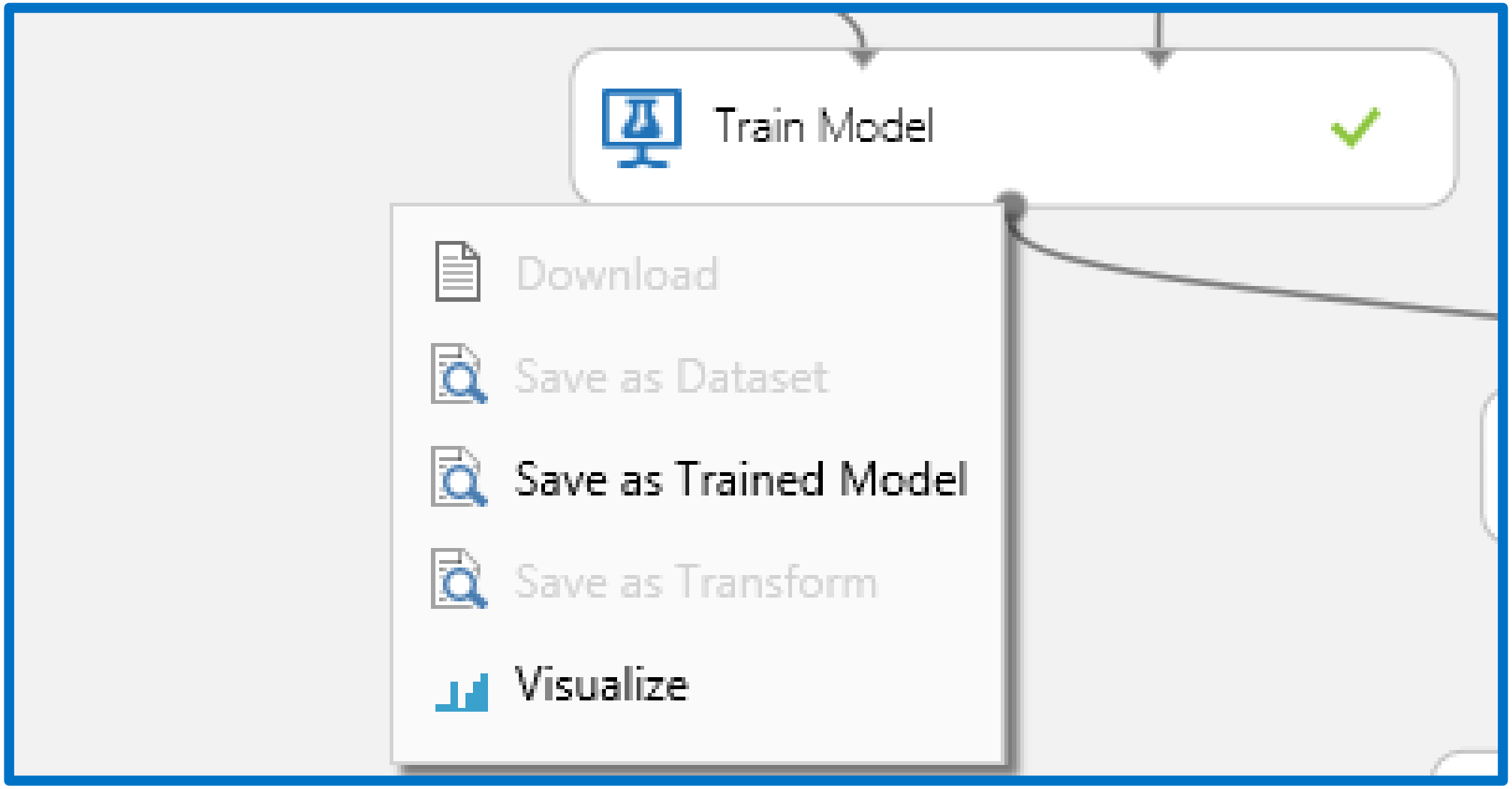

Binary Classification: Twitter sentiment analysis
In draft
Search experiment items
Saved Datasets
Trained Models
Transforms
Data Format Conversions
Data Input and Output
Data Transformation
Feature Selection
Machine Learning
OpenCV Library Modules
Python Language Modules
R Language Modules
Statistical Functions
Text Analytics
Web Service
Deprecated
Draft saved at 12:09:56 PM
Metadata Editor
set the text column type to non-categorical string
Feature Hashing
get the occurrence frequency of unigrams and bigrams in text
Split Data
split data into two sets: 80% training set and 20% test set
Filter Based Feature Selection
select the top 20k most relevant features
Two-Class Support Vector ...
1
Train Model
1:1
Properties
Two-Class Support Vector Mac...
Create trainer mode
Single Parameter
Number of iterations
1
Lambda
0.001
Normalize features
Project to the unit-sp...
Random number seed
123
Allow unknown categ...
Quick Help
Creates a binary classification model using the Support Vector Machine algorithm
(more help...)

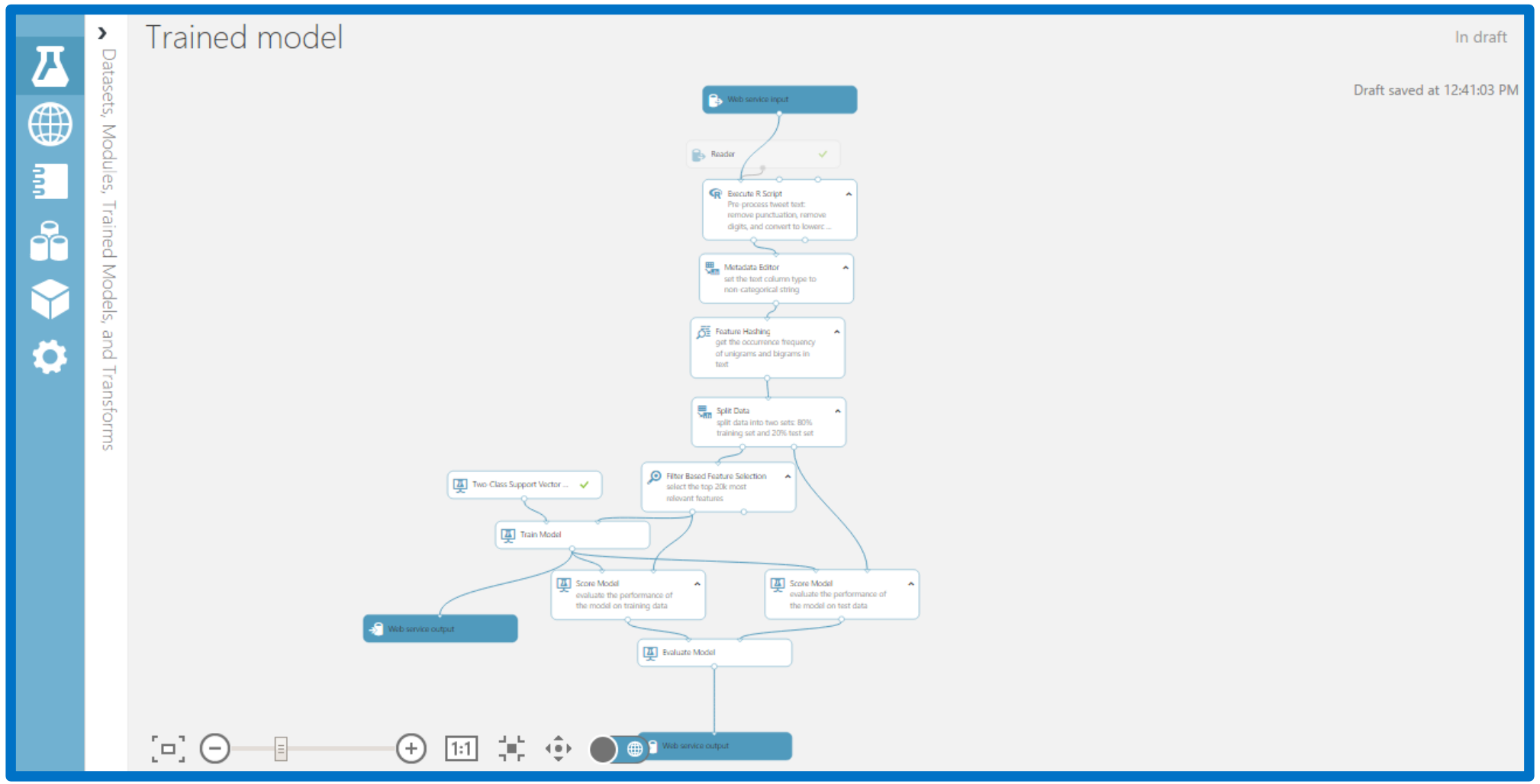

Binary Classification: Twitter sentiment analysis
Finished running
Search experiment items
Saved Datasets
Trained Models
Transforms
Data Format Conversions
Data Input and Output
Data Transformation
Feature Selection
Machine Learning
OpenCV Library Modules
Python Language Modules
R Language Modules
Statistical Functions
Text Analytics
Web Service
Deprecated
Split Data
split data into two sets: 80% training set and 20% test set
Two-Class Support Vector ...
Filter Based Feature Selection
select the top 20k most relevant features
Train Model
Score Model
evaluate the performance of the model on training data
Score Model
evaluate the performance of the model on test data
Evaluate Model
Download
Save as Dataset
Save as Trained Model
Save as Transform
Visualize
Generate Data Access Code...
Open in a new Notebook
1:1

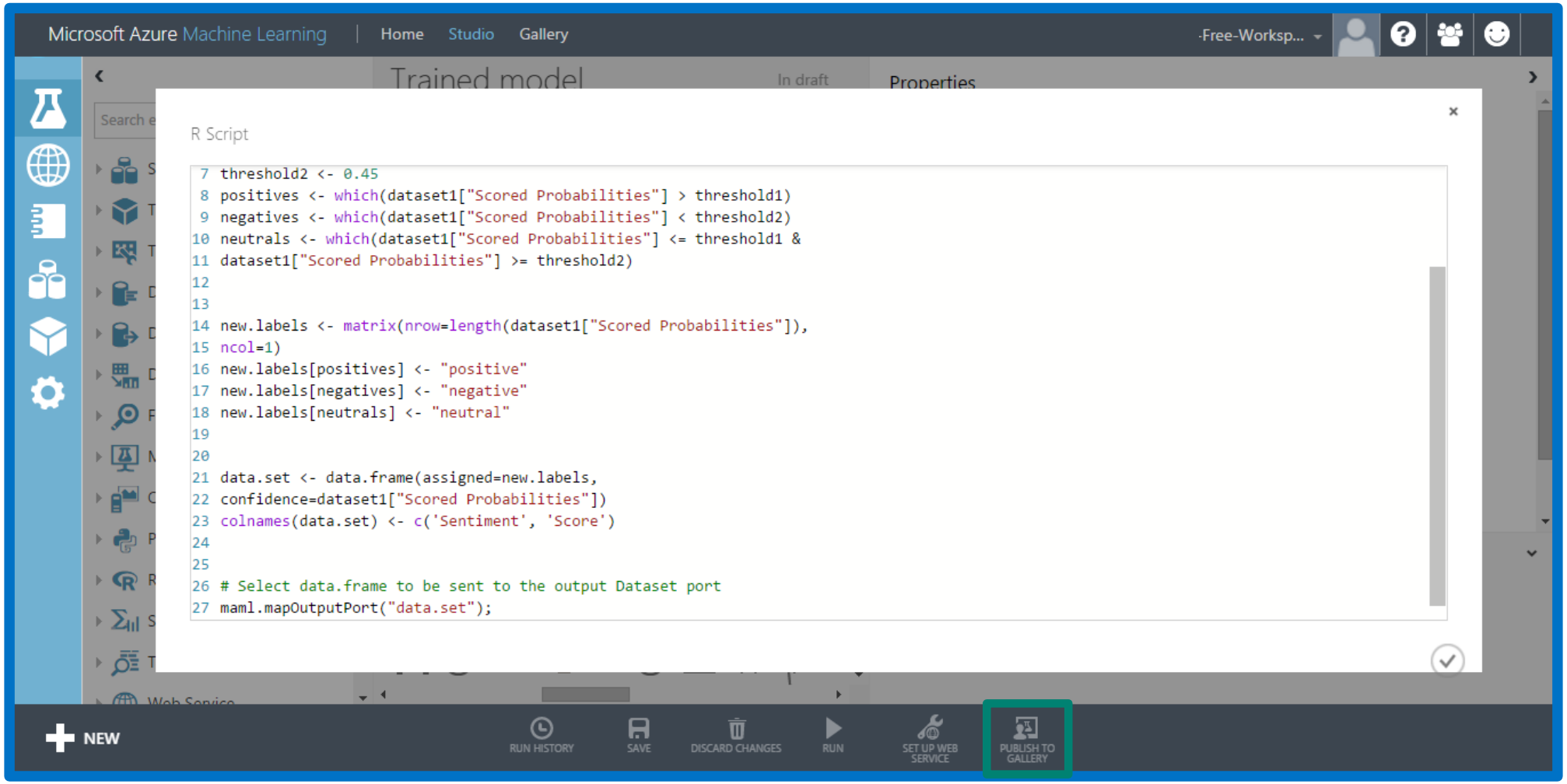

Binary Classification: Twitter sentiment analysis
Finished running
Binary Classification: Twitter sentiment analy... › Evaluate Model › Evaluation results
ROC PRECISION/RECALL LIFT
True Positive Rate
False Positive Rate
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Scored dataset
Scored dataset to compare



Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	10997	481	0.090	0.582	0.291	0.958	0.172	0.545	0.992	0.001
(0.800,0.900]	13852	1512	0.210	0.679	0.547	0.926	0.388	0.613	0.969	0.008
(0.700,0.800]	11738	2758	0.323	0.749	0.695	0.885	0.572	0.684	0.926	0.029
(0.600,0.700]	9223	5106	0.435	0.781	0.766	0.823	0.716	0.749	0.846	0.081
(0.500,0.600]	5490	6048	0.525	0.777	0.782	0.763	0.802	0.791	0.751	0.153
(0.400,0.500]	3800	6678	0.607	0.754	0.778	0.709	0.861	0.823	0.647	0.240
(0.300,0.400]	2858	7301	0.686	0.719	0.763	0.660	0.906	0.850	0.533	0.341
(0.200,0.300]	2350	8372	0.770	0.672	0.742	0.612	0.942	0.875	0.402	0.462
(0.100,0.200]	2007	10097	0.865	0.609	0.714	0.563	0.974	0.903	0.244	0.613
(0.000,0.100]	1685	15647	1.000	0.500	0.667	0.500	1.000	1.000	0.000	0.855


Train Model
Download
Save as Dataset
Save as Trained Model
Save as Transform
Visualize


Trained model
In draft
Draft saved at 12:41:03 PM
Datasets, Modules, Trained Models, and Transforms
Web service input
Reader
Execute R Script
Pre process tweet text: remove punctuation, remove digits, and convert to lowerc ...
Metadata Editor
set the text column type to non-categorical string
Feature Hashing
get the occurrence frequency of unigrams and bigrams in text
Split Data
split data into two sets: 80% training set and 20% test set
Two-Class Support Vector ...
Filter Based Feature Selection
select the top 20k most relevant features
Train Model
Score Model
evaluate the performance of the model on training data
Score Model
evaluate the performance of the model on test data
Web service output
Evaluate Model
1:1
Web service output


Microsoft Azure Machine Learning
Home
Studio
Gallery
Free-Worksp...
Trained model
In draft
Properties
R Script
threshold2 <- 0.45
positives <- which(dataset1["Scored Probabilities"] > threshold1)
negatives <- which(dataset1["Scored Probabilities"] < threshold2)
neutrals <- which(dataset1["Scored Probabilities"] <= threshold1 &
dataset1["Scored Probabilities"] >= threshold2)
new.labels <- matrix(nrow=length(dataset1["Scored Probabilities"]),
ncol=1)
new.labels[positives] <- "positive"
new.labels[negatives] <- "negative"
new.labels[neutrals] <- "neutral"
data.set <- data.frame(assigned=new.labels,
confidence=dataset1["Scored Probabilities"])
colnames(data.set) <- c('Sentiment', 'Score')
# Select data.frame to be sent to the output Dataset port
maml.mapOutputPort("data.set");
NEW
RUN HISTORY
SAVE
DISCARD CHANGES
RUN
SET UP WEB SERVICE
PUBLISH TO GALLERY

Questions

- Machine Learning practitioner
- Over 25 years of professional experience
- Artificial Intelligence MVP & MCT
- Microsoft Certified Solutions Expert
  - Data Management and Analytics
  - Cloud Platform and Infrastructure
  - Business Intelligence
- Microsoft Certified Solutions Developer
  - Azure Solution Architect