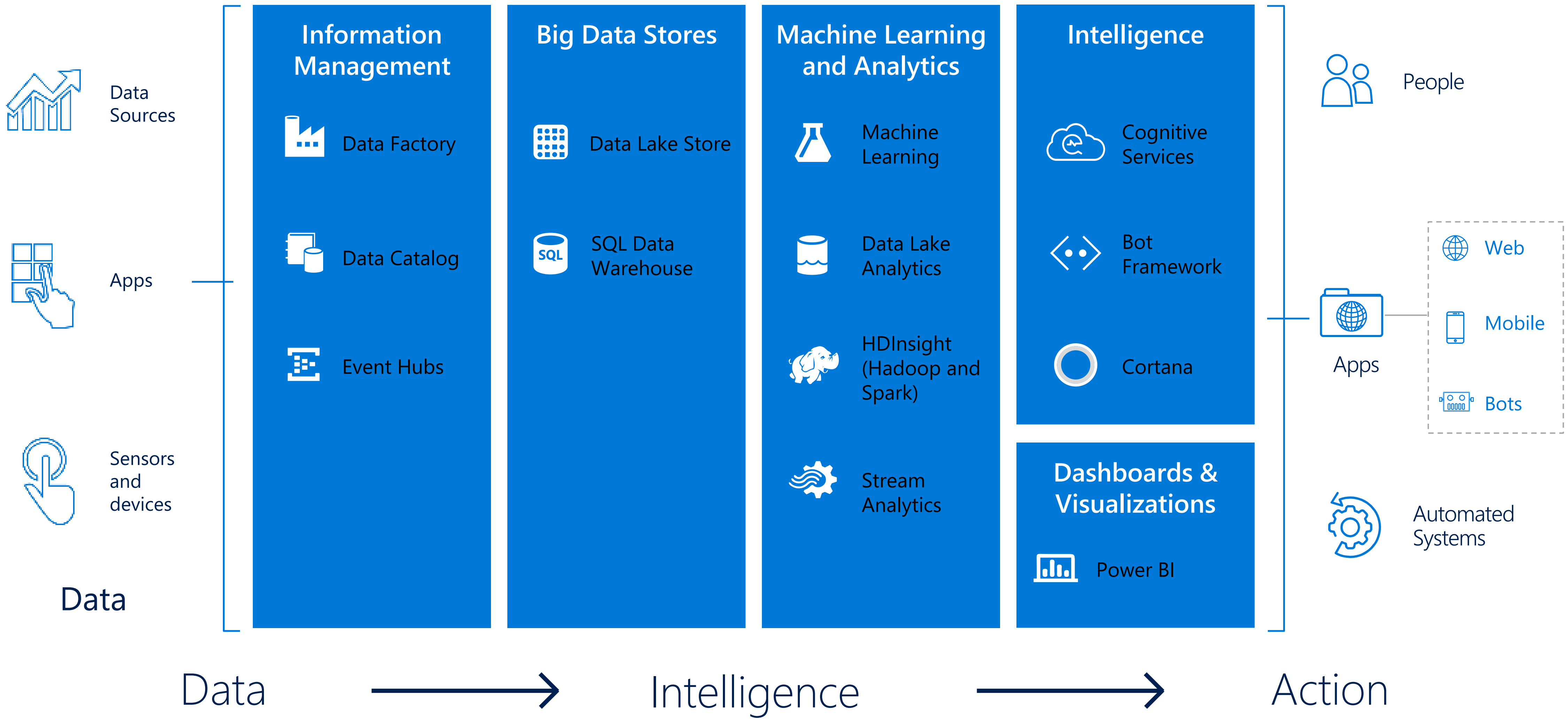
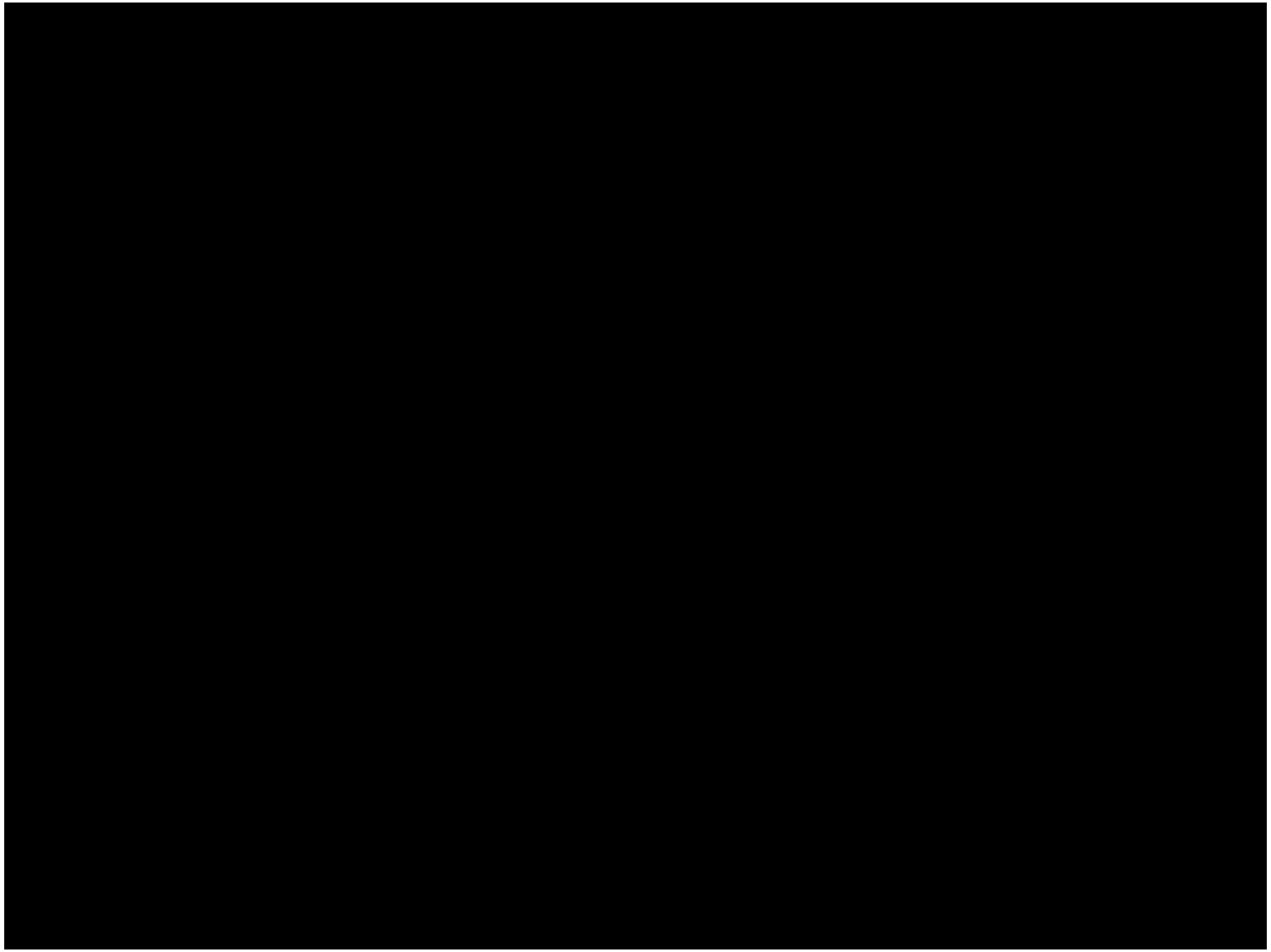
- https://github.com/milanleecal/rosetta20170321

Transform data into intelligent action



Seeing AI – Cognitive Services



Microsoft Cognitive Services

Give your apps a human side

Vision

From faces to feelings, allow your apps to understand images and video

Speech

Hear and speak to your users by filtering noise, identifying speakers, and understanding intent

Language

Process text and learn how to recognize what users want

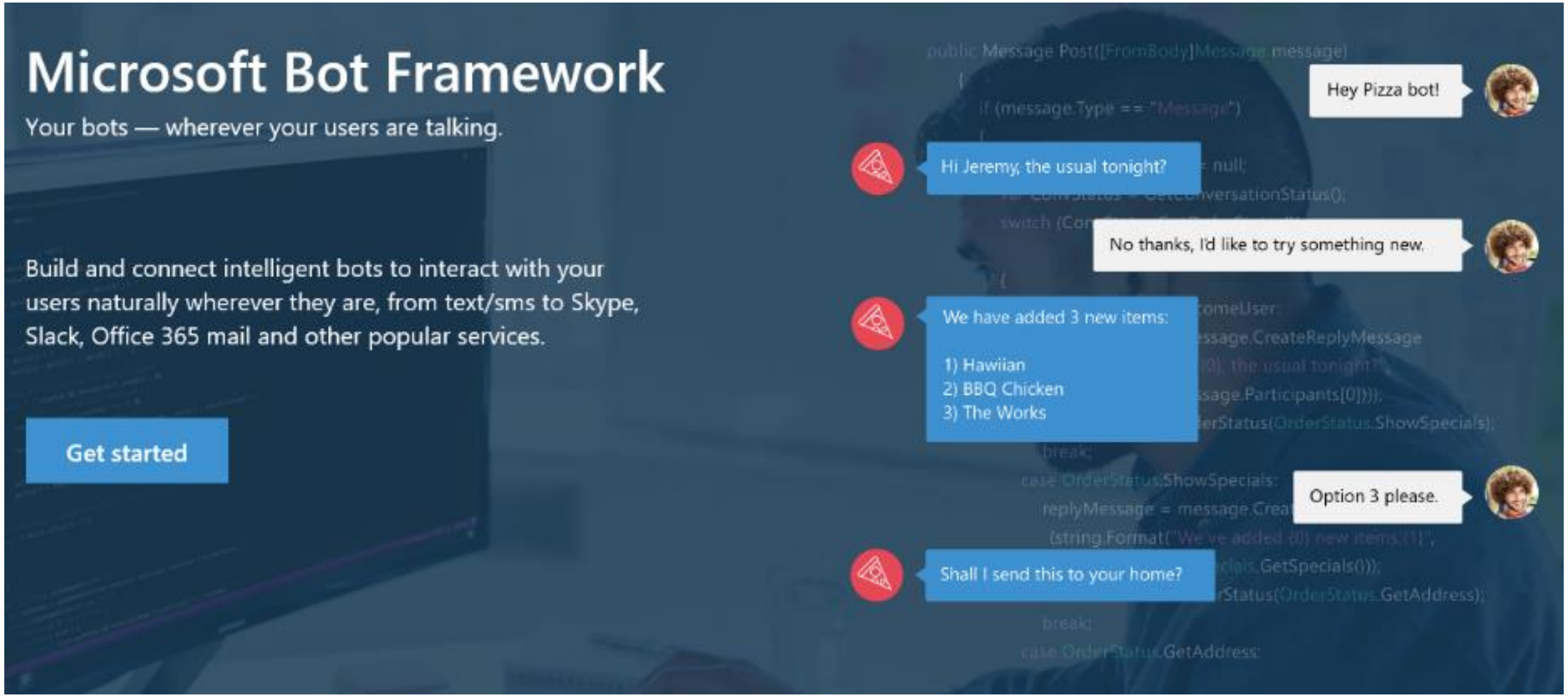
Knowledge

Tap into rich knowledge amassed from the web, academia, or your own data

Search

Access billions of web pages, images, videos, and news with the power of Bing APIs

Your bots – wherever your users converse



- Bot Connector Service: A service to register your bot, configure channels and publish to the Bot Directory. Connect your bot(s) seamlessly to text/sms, Office 365 mail, Skype, Slack, Twitter, and more.
- Bot Builder SDK: An open source SDK hosted on GitHub. Everything you need to build great dialogs within your Node.js or C# bot
- Bot Directory: A public directory of bots registered through the Bot Connector Service. Discover, try, and add bots to conversation experiences

Microsoft & Machine Learning

18 years of realizing innovation

1999	2004	2005	2008	2010	2012	2014
Computers work on users behalf, filtering junk email	Microsoft search engine built with machine learning	SQL Server enables data mining	Bing Maps ships with ML traffic-prediction service	Microsoft Kinect can watch users gestures	Successful, real-time, speech-to-speech translation	Microsoft launches Azure Machine Learning

John Platt,
Distinguished scientist at
Microsoft Research

"Machine learning is pervasive throughout Microsoft products."

Microsoft Azure Machine Learning

Built for a cloud-first, mobile-first world

Fully managed

No software to install, no hardware to manage, and one portal to view and update.

Integrated

Simple drag, drop and connect interface for Data Science. No need for programming for common tasks.

Best in Class Algorithms + R + Python

Built-in collection of best of breed algorithms. Support for R and Python.

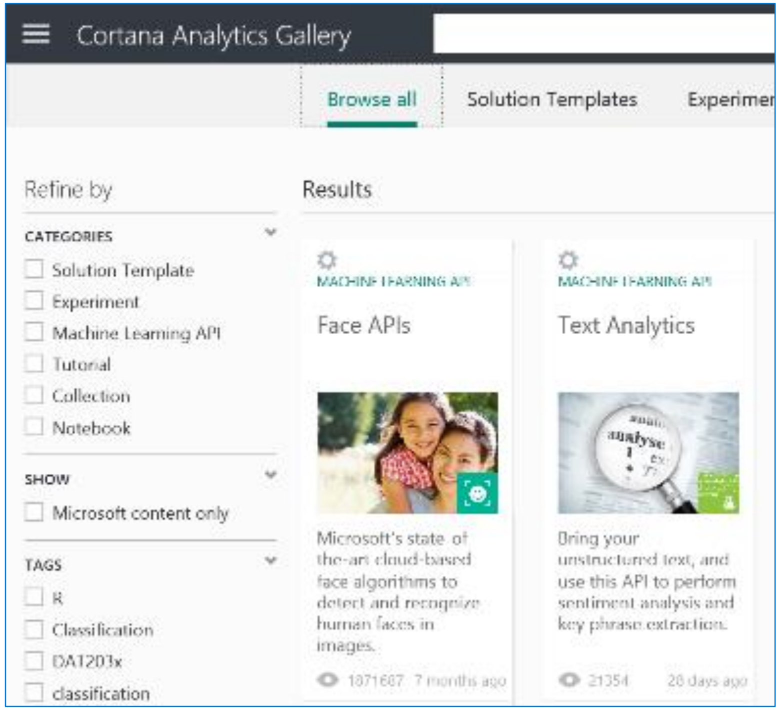
Deploy in Seconds

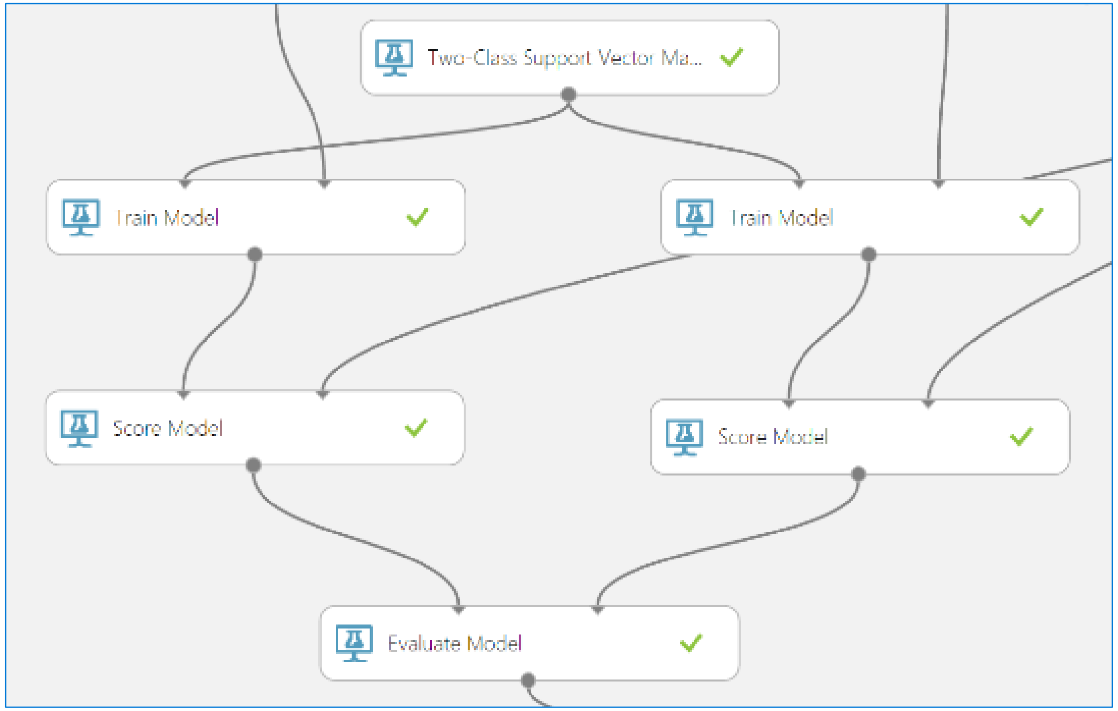
Operationalize models with a single click. Retrain models programmatically.

Easily build, deploy, and share predictive analytics solutions

Machine Learning and Analytics

- Machine Learning
- Data Lake Analytics
- HDInsight (Hadoop and Spark)
- Stream Analytics

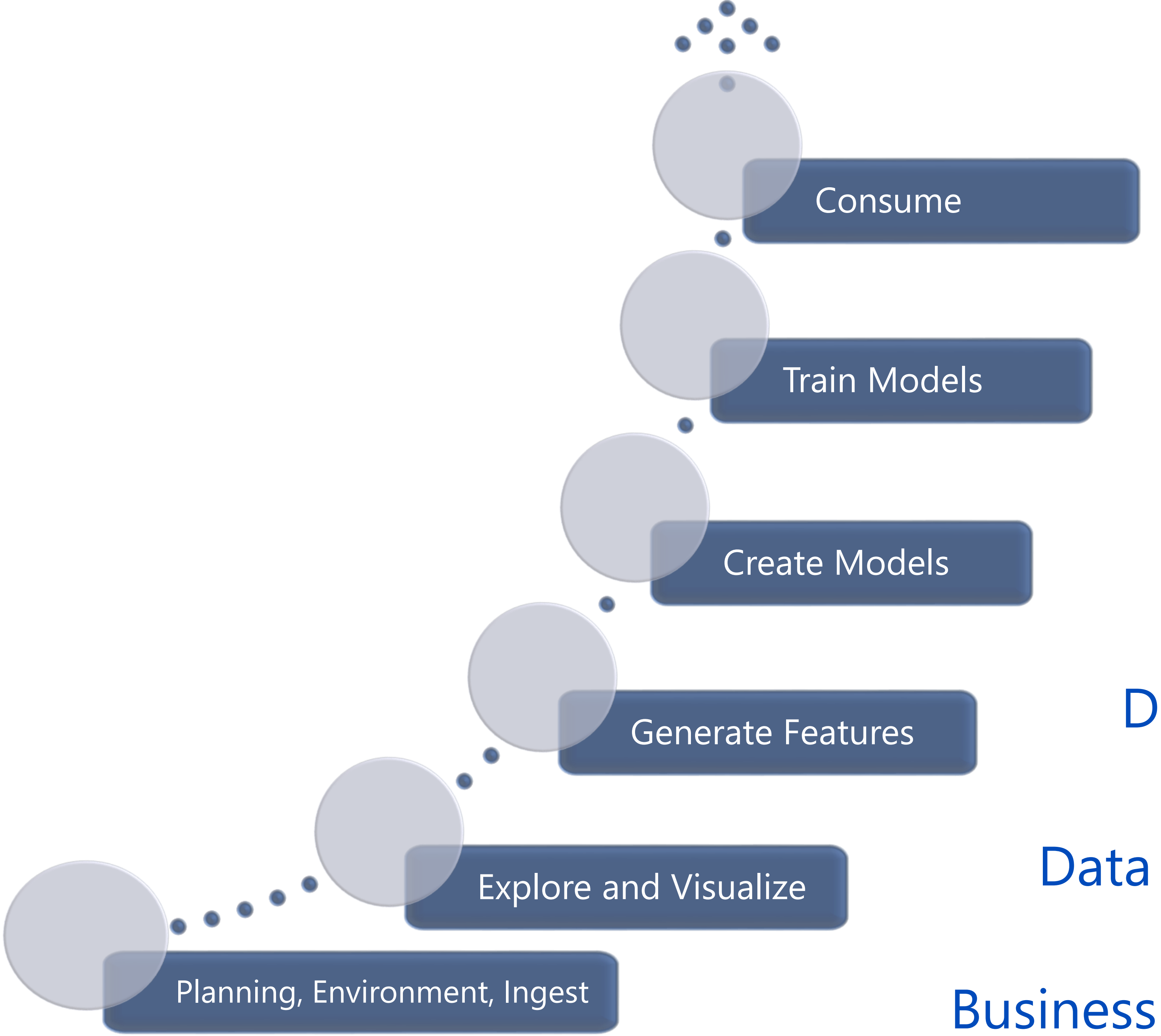




- Simple, scalable, cutting edge. A fully managed cloud service that enables you to easily build, deploy, and share predictive analytics solutions.
- Deploy in minutes. Azure Machine Learning means business. You can deploy your model into production as a web service that can be called from any device, anywhere and that can use any data source.
- Publish, share, monetize. Share your solution with the world in the Gallery or on the Azure Marketplace.

Data Science Process

The cross-industry process for data mining (CRISP-DM)



Deploying

Evaluating

Modeling

Data Preparation

Data Understanding

Business Understanding

Everyday examples of predictive analytics

- Product recommendation – “customers who bought this item also bought”
- Mortgage applications – credit worthiness
- Pattern recognition – speech recognition on your smart phone, character recognition on postal mails, facial recognition on security systems
- Web search page result – display sequences to render on page
- Predictive Maintenance – used on things we can monitor: planes, elevators, cars, data centers, etc.
- Healthcare – determine patient outcomes and future care

Getting Started with Azure Machine Learning (First time setting up)

- Set up a Microsoft Azure Account
- One stop shop for Azure services -> portal.azure.com
- Set up an AzureML Workspace -> under Intelligence + analytics, choose Machine Learning Workspace
- Give a workspace name
- Set up a new Resource Group
- Set up a new Storage Account
- Set up a new Web Service Plan
- Launch AzureML Studio

Accessing Azure Machine Learning (Return visits)

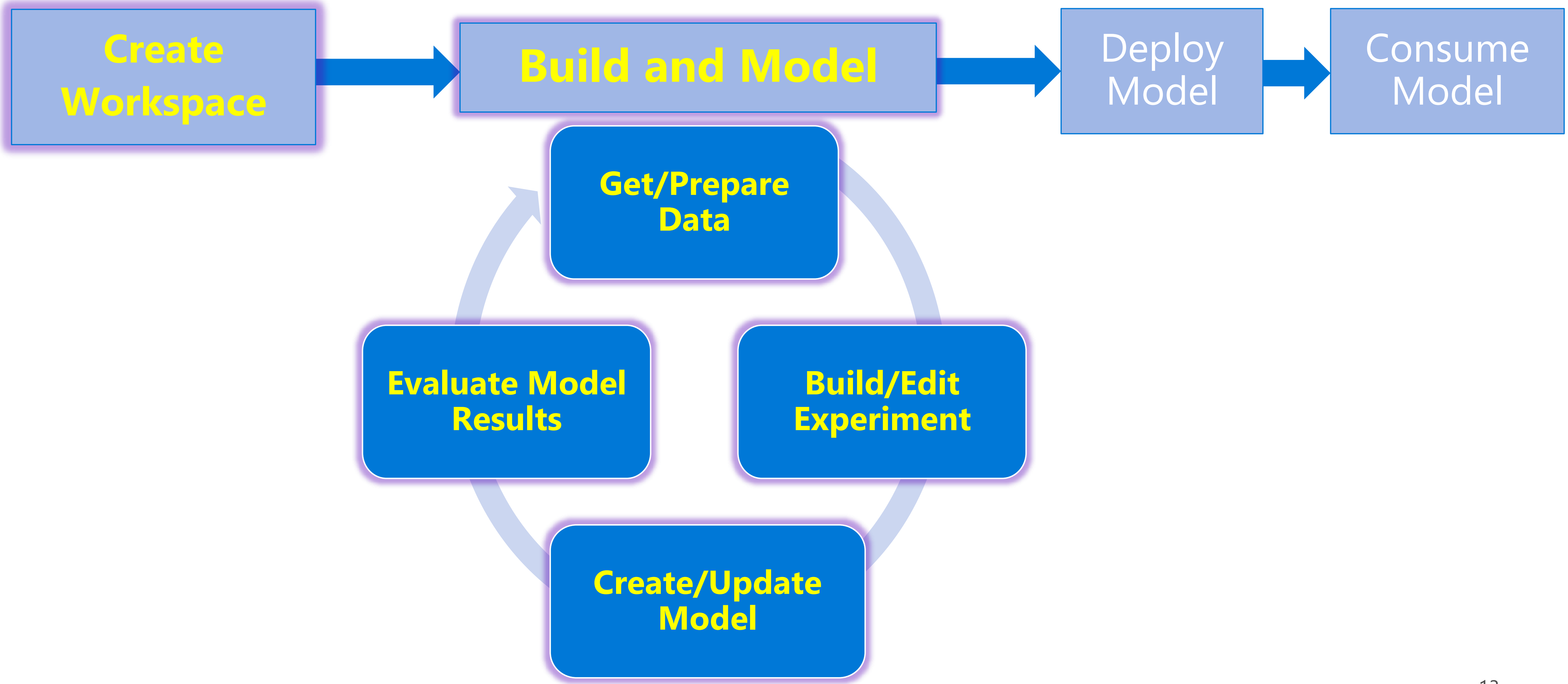
Option 1

- Log in to Azure portal -> portal.microsoft.com
- Select your AzureML workspace on the dashboard
- Launch AzureML Studio

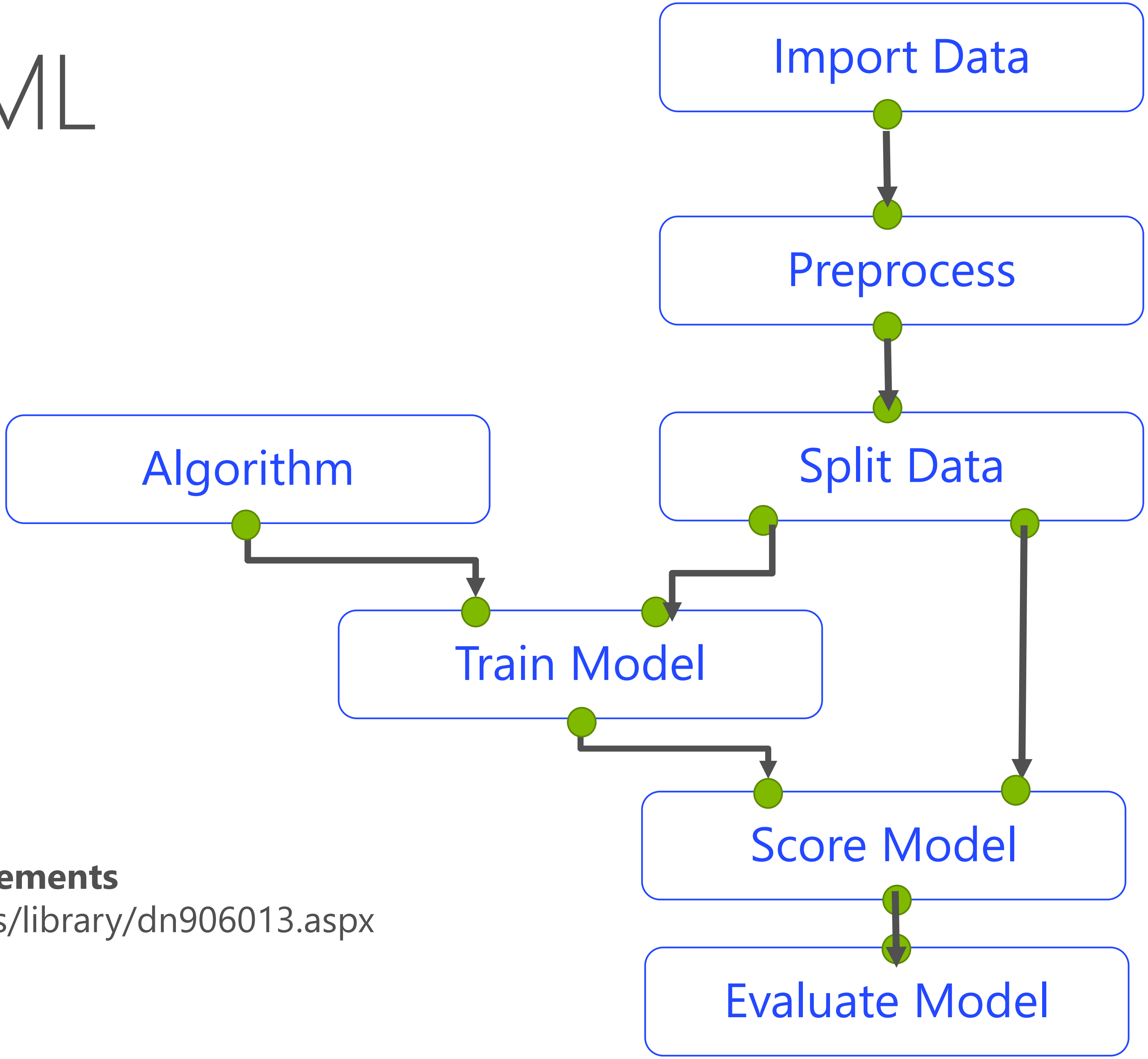
Option 2

- Log in directly to AzureML Studio -> studio.azureml.net

Creating an Experiment



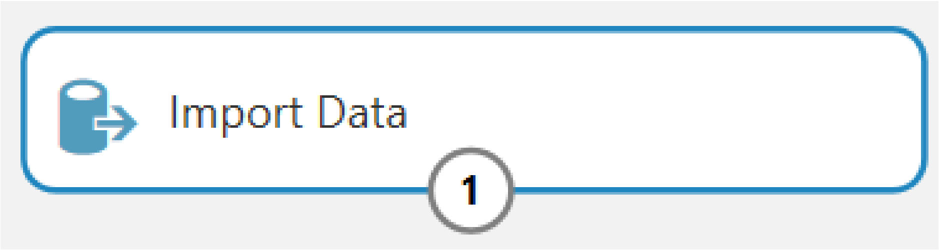
Basic Azure ML Elements



Reference to all the AzureML Elements
https://msdn.microsoft.com/en-us/library/dn906013.aspx

Import Data

- Read data from:
  - Web URL
  - Hive query
  - Azure SQL database
  - Azure table
  - Azure blob storage

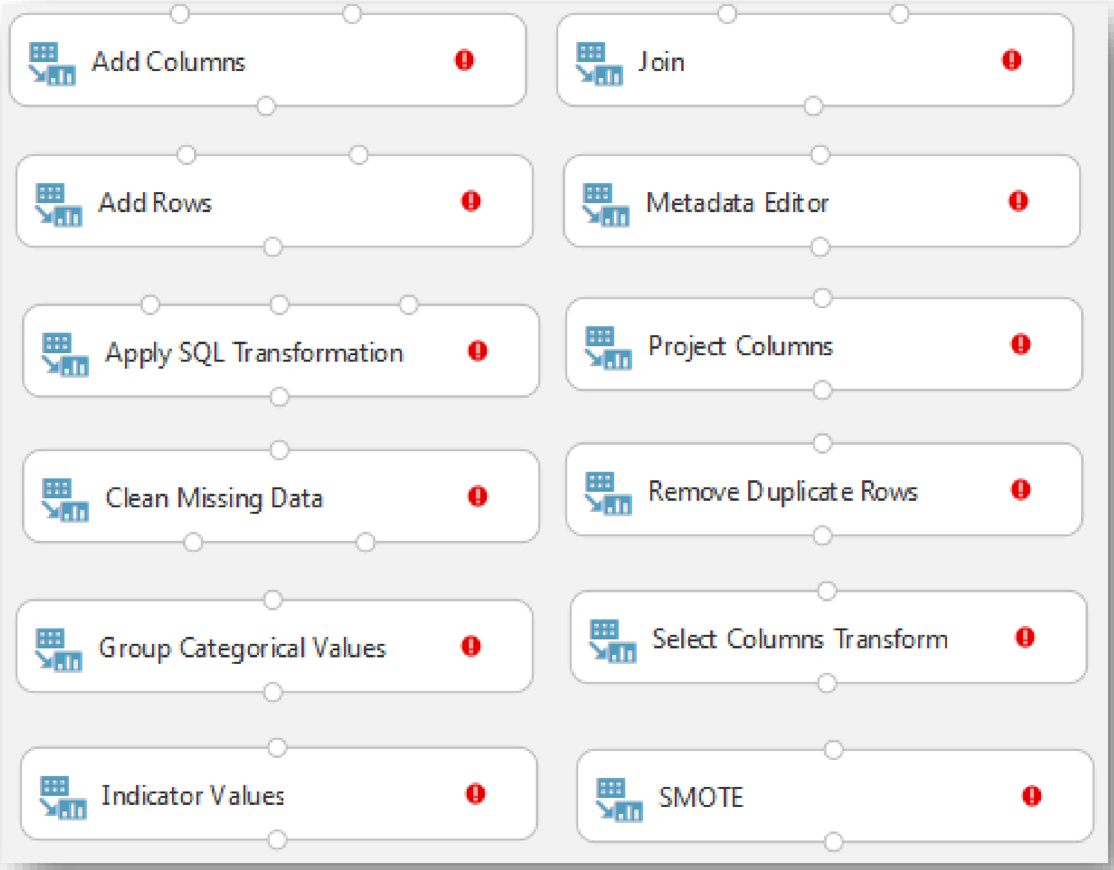


- Load a collection of images from blob storage for use in image classification tasks

Preprocess Data

Prepare data for Machine Learning

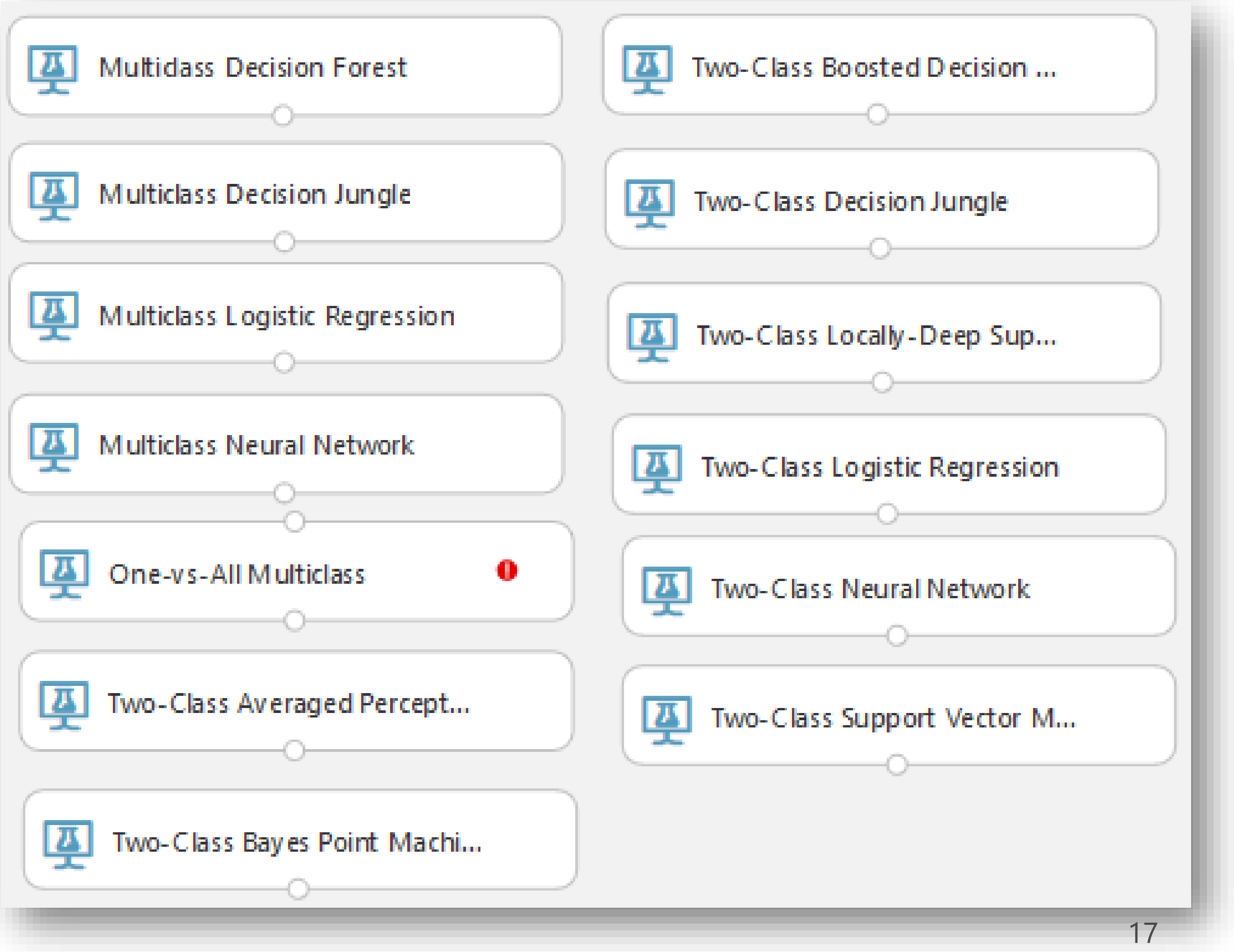
- Merging datasets
- Grouping and summarizing data
- Converting values to another type
- Checking for missing values and replacing them with appropriate values
- Flagging columns as features (for example, labels)

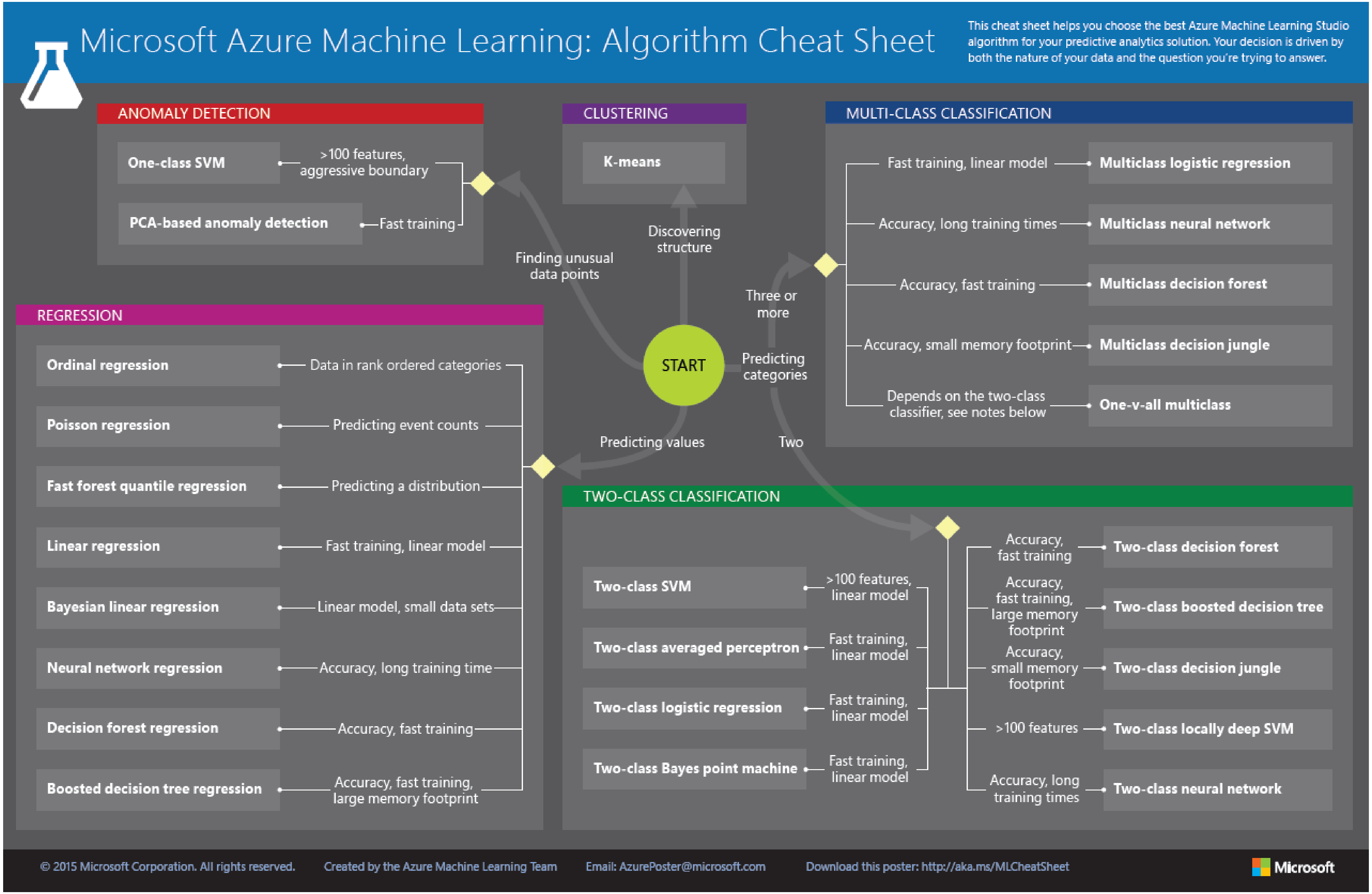


Choosing a Model

Initialize Model – Classification

- Predict the class or category for data




Microsoft Azure Machine Learning: Algorithm Cheat Sheet
This cheat sheet helps you choose the best Azure Machine Learning Studio algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.
ANOMALY DETECTION
One-class SVM
>100 features, aggressive boundary
PCA-based anomaly detection
Fast training
CLUSTERING
K-means
Discovering structure
Finding unusual data points
START
Predicting categories
Three or more
Two
Predicting values
MULTI-CLASS CLASSIFICATION
Fast training, linear model
Multiclass logistic regression
Accuracy, long training times
Multiclass neural network
Accuracy, fast training
Multiclass decision forest
Accuracy, small memory footprint
Multiclass decision jungle
Depends on the two-class classifier, see notes below
One-v-all multiclass
REGRESSION
Ordinal regression
Data in rank ordered categories
Poisson regression
Predicting event counts
Fast forest quantile regression
Predicting a distribution
Linear regression
Fast training, linear model
Bayesian linear regression
Linear model, small data sets
Neural network regression
Accuracy, long training time
Decision forest regression
Accuracy, fast training
Boosted decision tree regression
Accuracy, fast training, large memory footprint
TWO-CLASS CLASSIFICATION
Two-class SVM
>100 features, linear model
Two-class averaged perceptron
Fast training, linear model
Two-class logistic regression
Fast training, linear model
Two-class Bayes point machine
Fast training, linear model
Accuracy, fast training
Two-class decision forest
Accuracy, fast training, large memory footprint
Two-class boosted decision tree
Accuracy, small memory footprint
Two-class decision jungle
>100 features
Two-class locally deep SVM
Accuracy, long training times
Two-class neural network
© 2015 Microsoft Corporation. All rights reserved.
Created by the Azure Machine Learning Team
Email: AzurePoster@microsoft.com
Download this poster: http://aka.ms/MLCheatSheet
Microsoft

Machine Learning Algorithms

Split into two main categories:

- Supervised learning
  - Labels provided
  - Predicting the future
  - Learn from known past examples to predict future

- Unsupervised learning
  - Labels not provided
  - Understanding the past and making sense of data
  - Learning the structure of data

Supervised learning – example

Data Input Features (Vectors)

Known Outcomes

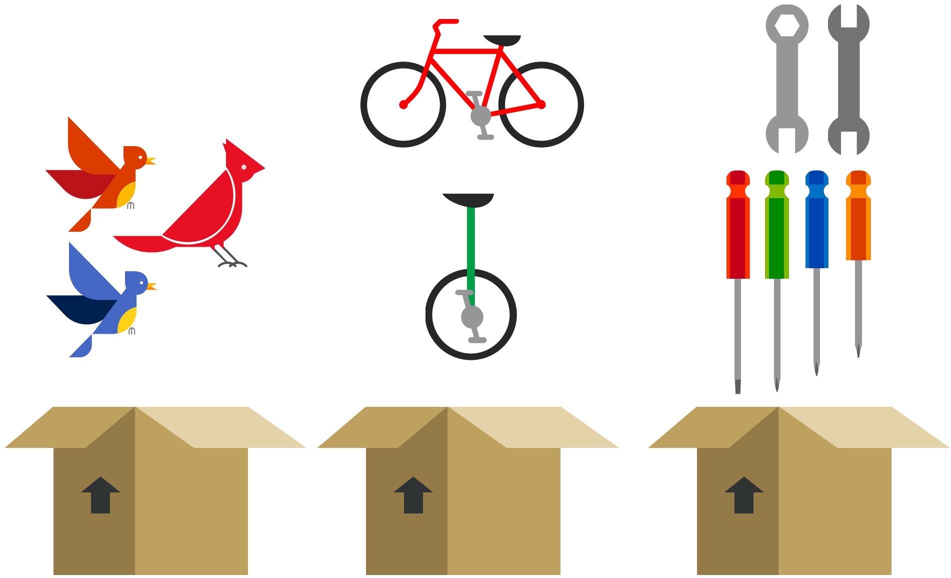
age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	hours-per-week	income
39	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Not-in-family	White	Male	60	<=50K
38	State-gov	Doctorate	16	Married-civ-spouse	Prof-specialty	Husband	White	Male	45	>50K
38	Private	Some-college	10	Divorced	Exec-managerial	Not-in-family	White	Female	50	<=50K
38	Private	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	Black	Male	40	<=50K
66	Private	11th	7	Married-civ-spouse	Craft-repair	Husband	White	Male	20	<=50K
26	Private	Bachelors	13	Married-civ-spouse	Sales	Wife	Black	Female	40	>50K
50	Private	9th	5	Divorced	Transport-moving	Not-in-family	White	Male	50	<=50K
53	Private	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White	Male	40	<=50K
28	Private	HS-grad	9	Never-married	Transport-moving	Unmarried	White	Male	55	<=50K
28	Private	HS-grad	9	Never-married	Exec-managerial	Not-in-family	White	Male	40	<=50K

Machine Learning Capabilities

SUPERVISED

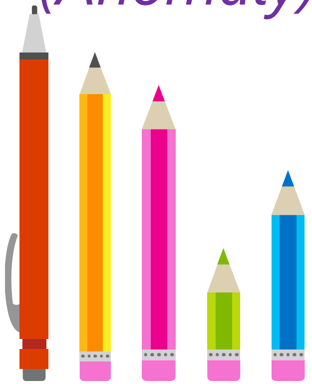
UNSUPERVISED

Which category
(Classification)

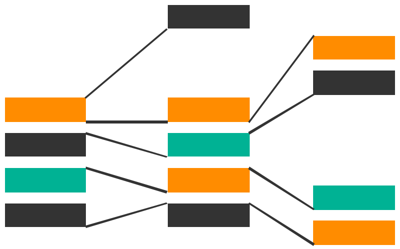


How much/many
(Regression)

Is it odd
(Anomaly)



Which group
(Clustering, Recommender)

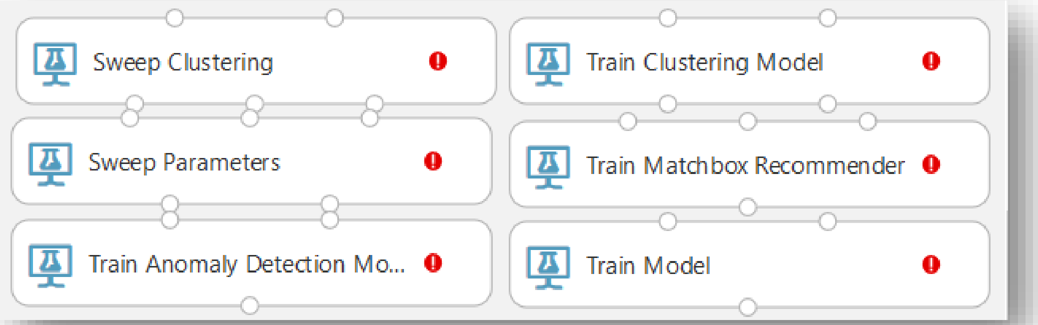


Testing the Model

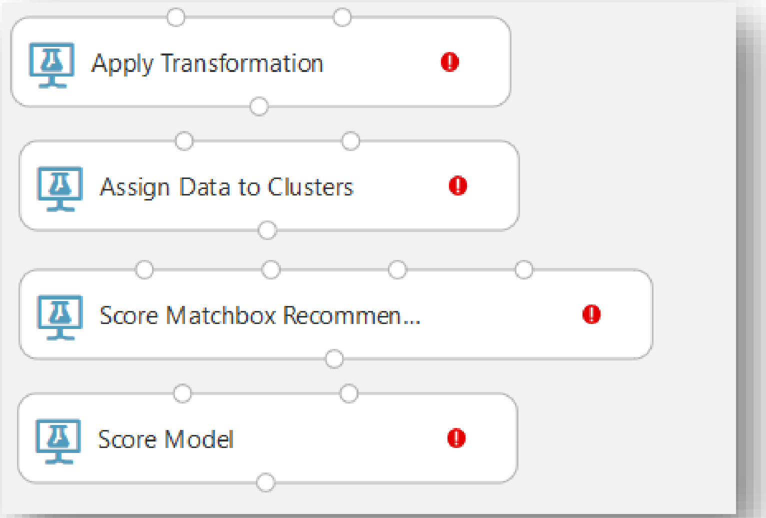
Apply a trained model to the test data to get:

- Estimates of projected demand, volume, or other numeric quantity, for regression models
- Cluster assignments
- A predicted class or outcome, for classification models
- Probability scores associated with these outputs

Train



Score



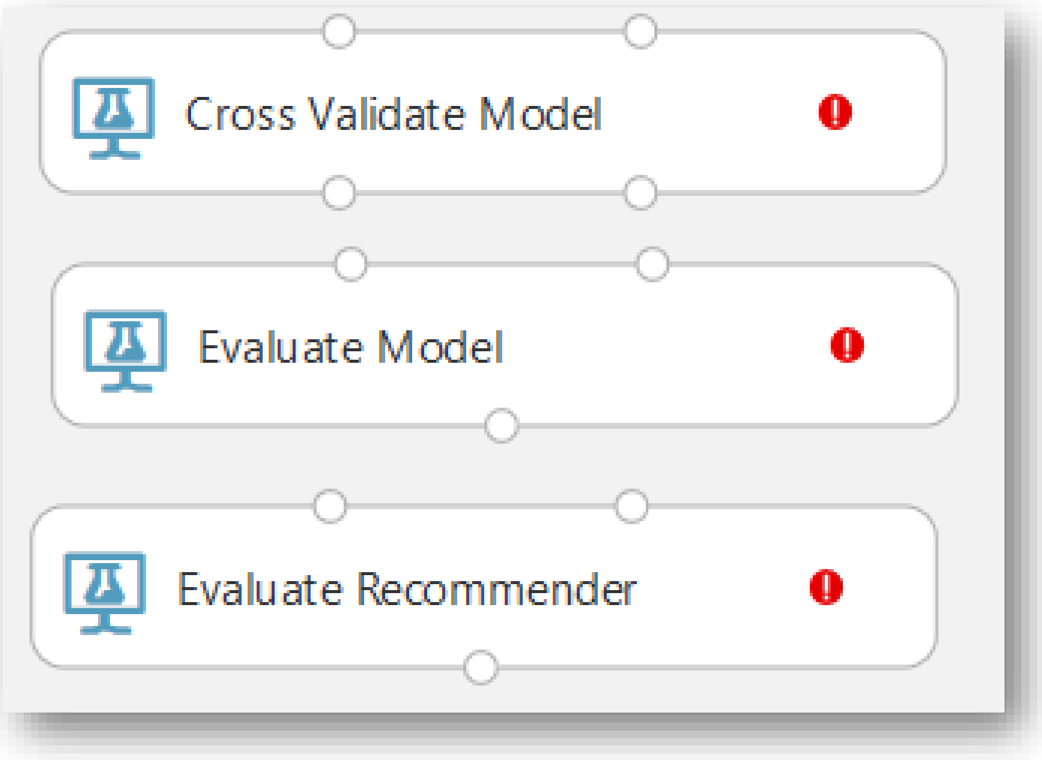
Evaluating the Model

Metrics for Classification Models

- Accuracy
- Recall
- Precision
- F-Score
- AUC
- Average Log Loss
- Training Log Loss

Metrics for Regression Models

- Mean absolute error (MAE)
- Root mean squared error (RMSE)
- Relative absolute error (RAE)
- Relative squared error (RSE)
- Coefficient of determination



Azure Machine Learning – Titanic

Predict likelihood of survival.

Data – passenger and boarding details

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05		S

view as

Data Dictionary

Data Dictionary

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Variable Notes

pclass: A proxy for socio-economic status (SES)
1st = Upper
2nd = Middle
3rd = Lower

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way...
Sibling = brother, sister, stepbrother, stepsister
Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way...
Parent = mother, father
Child = daughter, son, stepdaughter, stepson
Some children travelled only with a nanny, therefore parch=0 for them.

Uploading data – option 1 of 2

- In AML Studio, in lower left hand corner, click New -> Dataset -> From Local File. Then browse for the dataset.
- The data will appear in the workspace under “My Datasets”.
- With this option, the data persists only at the Workspace level.

Uploading data – option 2 of 2

- Open Microsoft Azure Storage Explorer from your computer. You will need to download and install it.
  - Install Azure Storage Explorer - Here
- Click on the top left "Connect to Azure Storage" button
- Select "Use a storage account name and key"
- Obtain account name and key from the Azure portal
  - Go to the storage account
  - In the left hand menu, go to "Access Keys"
  - Copy the storage account name, paste it in the Microsoft Azure Storage Explorer
  - Copy the key1, paste it in the Microsoft Azure Storage Explorer
  - Then click "Connect"
- In the Storage Explorer, you should now see your storage. Create a new blob container. You can now upload or drag files into the interface.
- With this option, your data persists in the storage account permanently.

Building a classification model (part 1 of 2)

- Data Input and Output -> Import Data
  - Azure Blob Storage; Public or SAS; Copy the URL from the blob properties of train.csv → https://publicisstorage.blob.core.windows.net/titanic/train.csv
  - File format: CSV
  - Check "URL file has header row" and check "Use cached results"
- Data Transformation -> Manipulation -> Select Columns in dataset
  - Select all columns except for Name, Cabin, Ticket.
- Data Transformation -> Manipulation -> Edit Metadata
  - Select PassengerID -> Under Fields -> Clear feature
- Data Transformation -> Manipulation -> Edit Metadata
  - Select Survived, Sex, Embark -> Under Categorical -> Make Categorical
- Data Transformation -> Manipulation -> Edit Metadata
  - Select Survived -> Under Fields -> Label

Building a classification model (part 2 of 2)

- Data Transformation -> Sample and Split -> Split Data
  - Set training sample to be 0.7. Put in a non-zero random seed. Set Stratified Split to True, and select Survived.
- Machine Learning -> Initialize Model -> Classification ->Two-Class Logistic Regression
  - Leave parameters as defaults
- Machine Learning -> Train -> Train Model
  - Select Survived as column name
  - Connect the machine learning algorithm to port 1. Connect the training dataset to port 2.
- Machine Learning -> Score -> Score Model
  - Connect trained model to port 1. Connect the test dataset to port 2.
- Machine Learning -> Evaluate -> Evaluate Model
- Run the experiment
- Save the experiment

Evaluating classification models (part 1 of 2)

	Event=Positive	Event=Negative	Events
Predicted Positive	True Positive(TP)	False Positive(FP)	TP+FP
Predicted Negative	False Negative(FN)	True Negative(TN)	FN+TN
Observations	TP+FN	FP+TN	Te

Accuracy
- Ratio of correctly predicted observations
- Accuracy=(TP+TN)/Te
- Good for even distribution of data say 50/50

Precision
- What percentage of Predicted observation were correct
- Precision=(TP)/(TP+FP)
- Good for uneven distribution

Recall
- What percentage of positive events were correctly predicted(sensitivity)
- Recall=TP/(TP+FN)
- Good for uneven distribution

F1 Score
- Weighted average of Precision and Recall
- F1 Score=2*(Recall*Precision)/(Recall + Precision)
- Better way to show uneven distribution

Data Set	Events=Positive	Events=Negative	Events
Predicted Positive	2894	994	3888
Predicted Negative	643	11750	12393
Observations	3537	12744	16281

Total Events=16281

Accuracy=(2894+1750)/16281=~90%

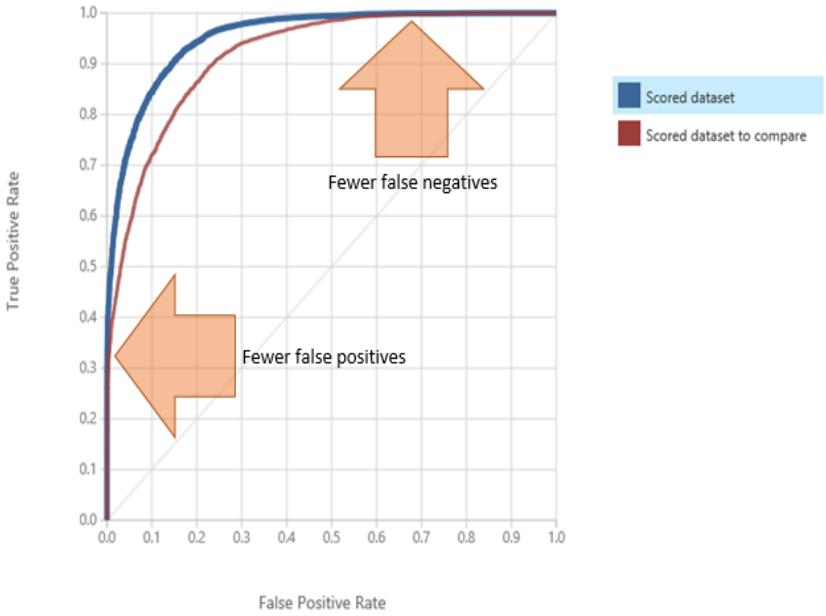
Precision=2894/3537=~82%

Recall=2894/3888=~74%

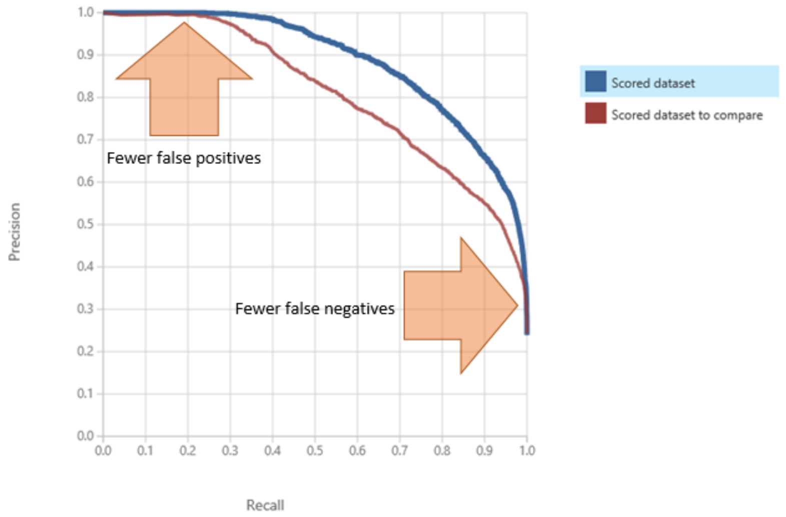
F1 Score= 2*(0.74*0.82)/(0.74+0.82)=~78%

Evaluating classification models (part 2 of 2)

ROC- Receiver operating characteristics



Precision / Recall Plot



AUC- Area under the curve

- Between 0 and 1, ideally should be close to 1

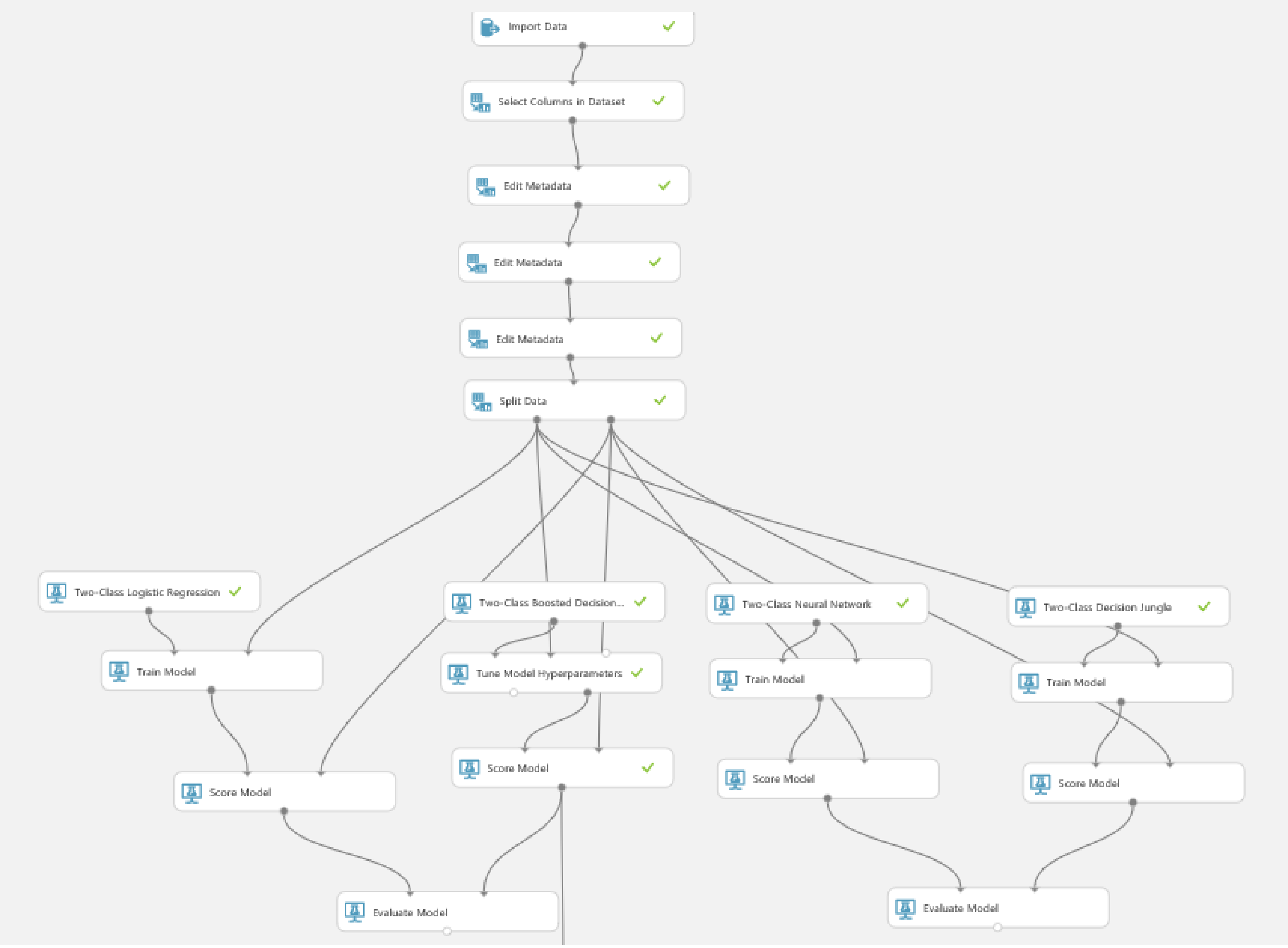
Threshold

- Optimize based on cost of False positive vs False negatives

Refining a classification model

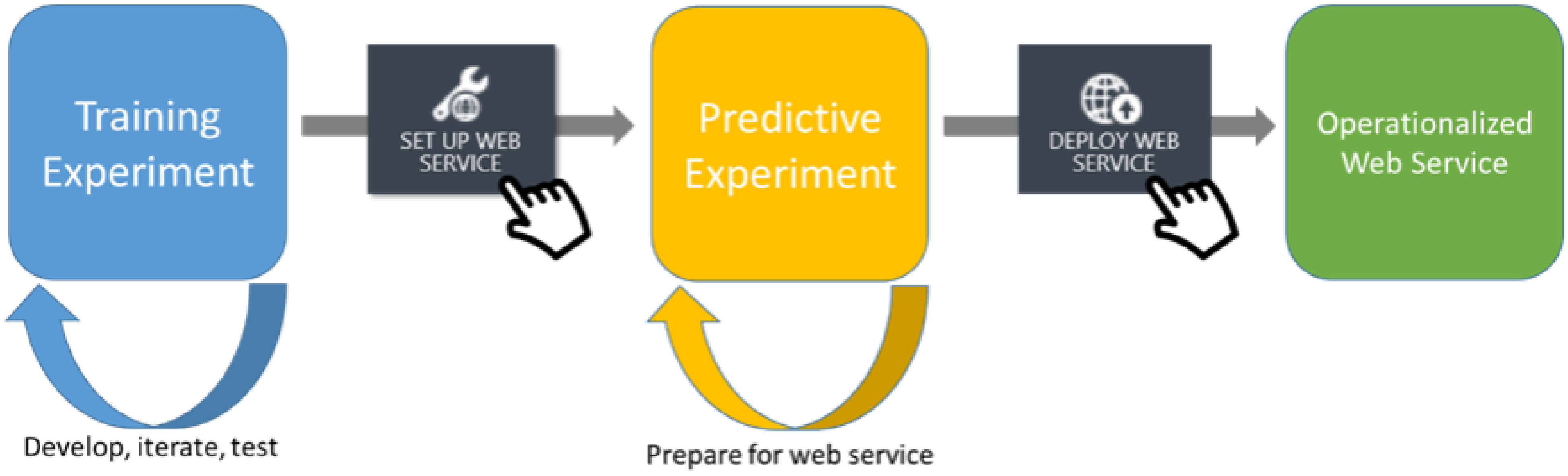
- Machine Learning -> Initialize Model -> Classification ->Two-Class Boosted Decision Tree
  - Create Trainer Mode: Parameter Range
- Machine Learning -> Train -> Tune Model Hyperparameters
  - Specify parameter sweeping: Random sweep
  - Maximum number of runs on random sweep: 10
  - Metric for measuring performance for classification: AUC
- Machine Learning -> Score -> Score Model
- Machine Learning -> Evaluate -> Evaluate Model
  - Connect both models to one module
- Run the experiment
- Save the experiment
- Try adding two additional machine learning algorithms for two-class classification. What is your final champion model?

Sample experiment in AzureML



Operationalize your model as a web service

Here are the stages that a typical solution follows as you develop and deploy it using AzureML

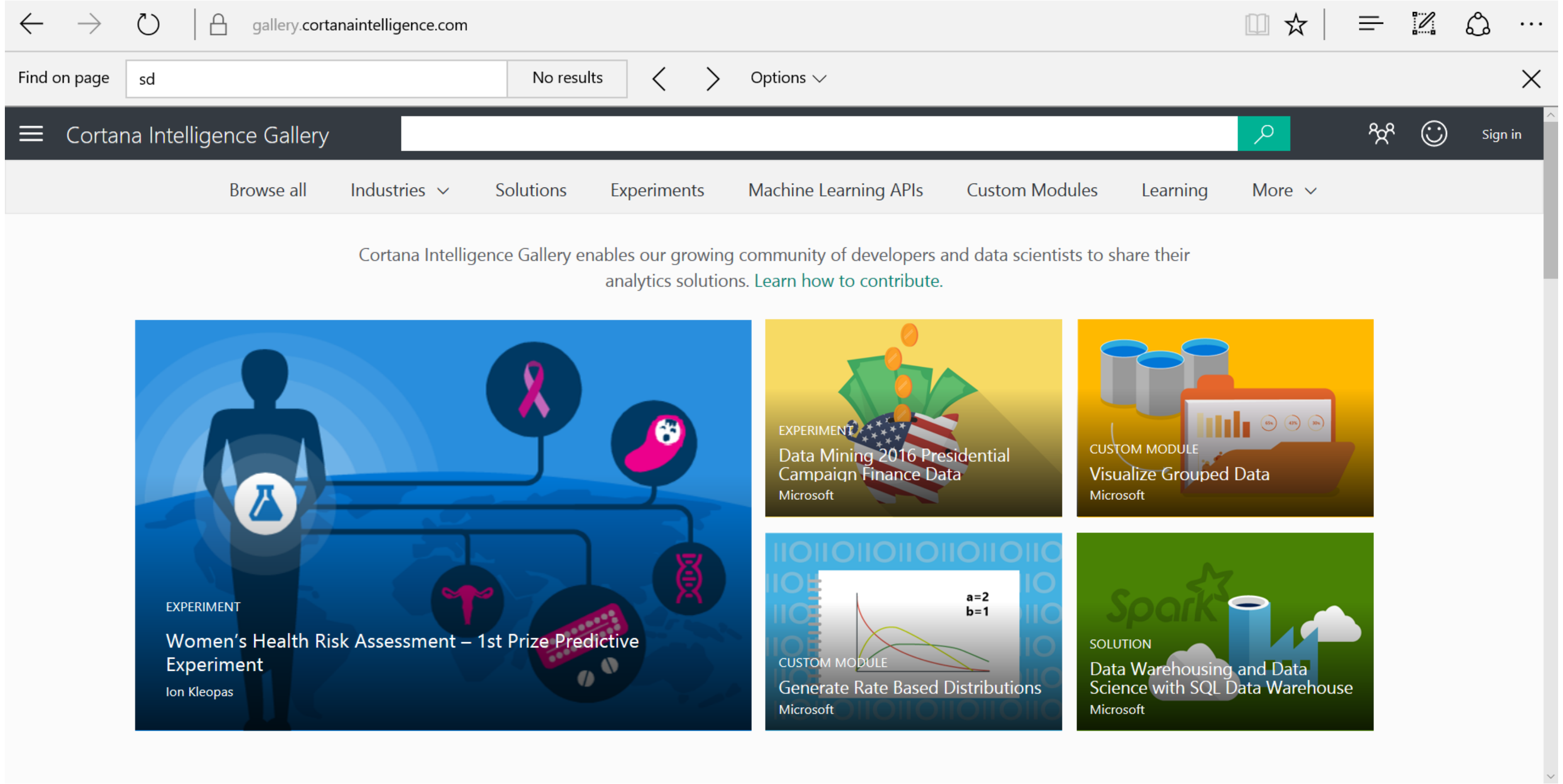


Operationalize Model

- Highlight “Train Model” for your champion model
- Click on Set up web service -> Predictive Web Service
- Run
- Deploy Web Service (Classic)
- Test the web service using the Request/Response API
- To embed a web service into a web app, you will need the API key and the API Host endpoint URL

Cortana Intelligence Gallery

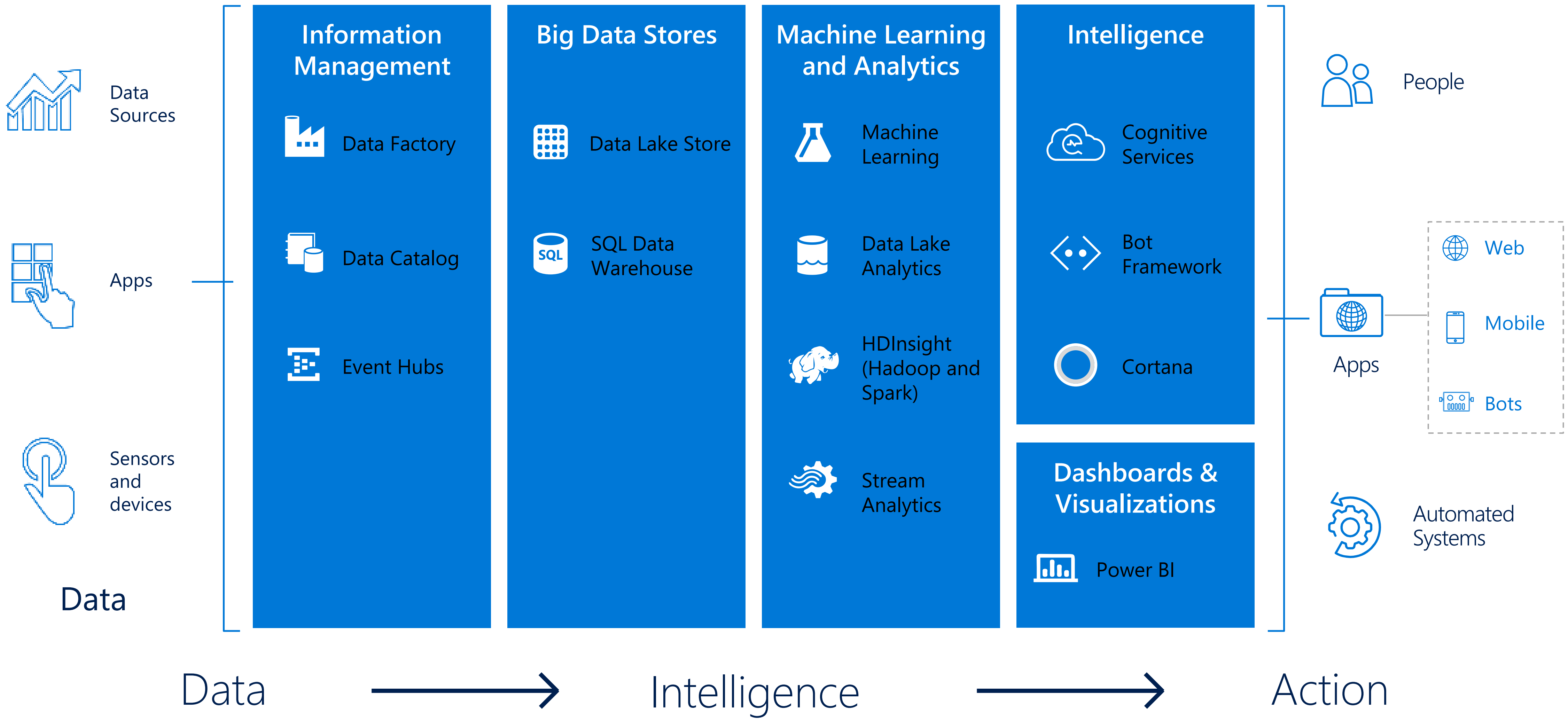
- Preconfigured machine learning templates and solutions -> https://gallery.cortanaintelligence.com/



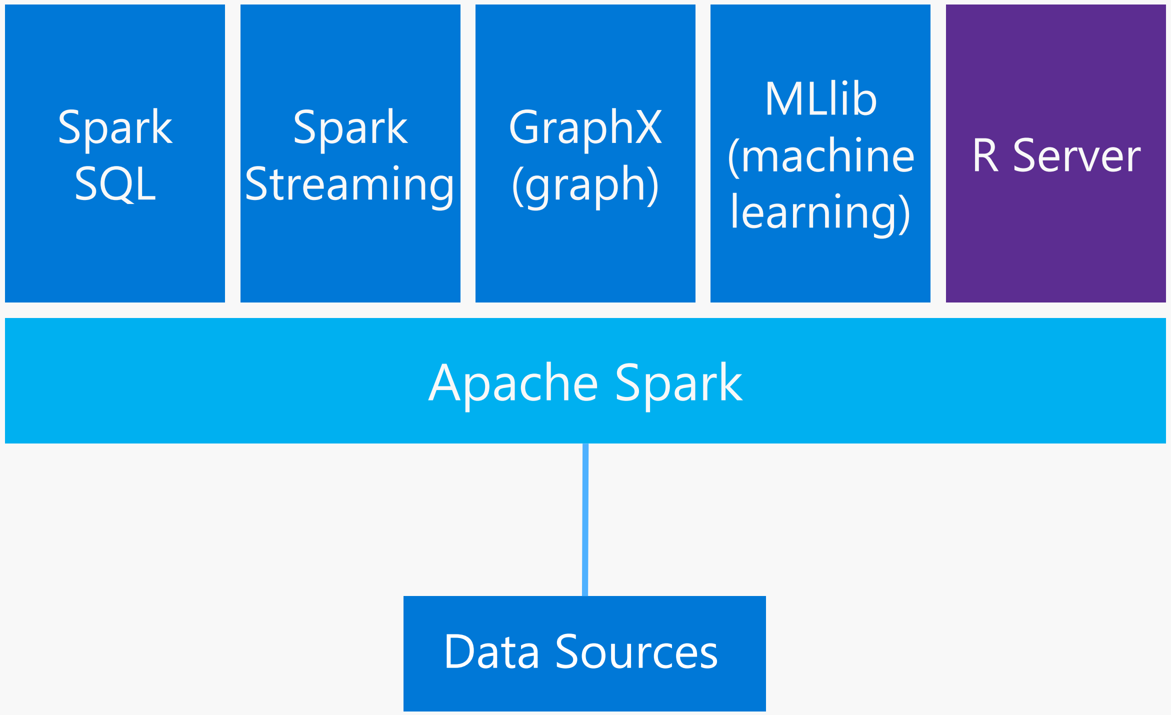
Export data for later use

- Data Input and Output -> Export Data
  - Azure Blob Storage
  - Authentication Type - Account
  - Put in storage name (obtained from your storage service)
  - Put in account key (obtained from your storage service)
  - Put in container and blob name (obtained from your storage service) – put it in this format container/blob.csv

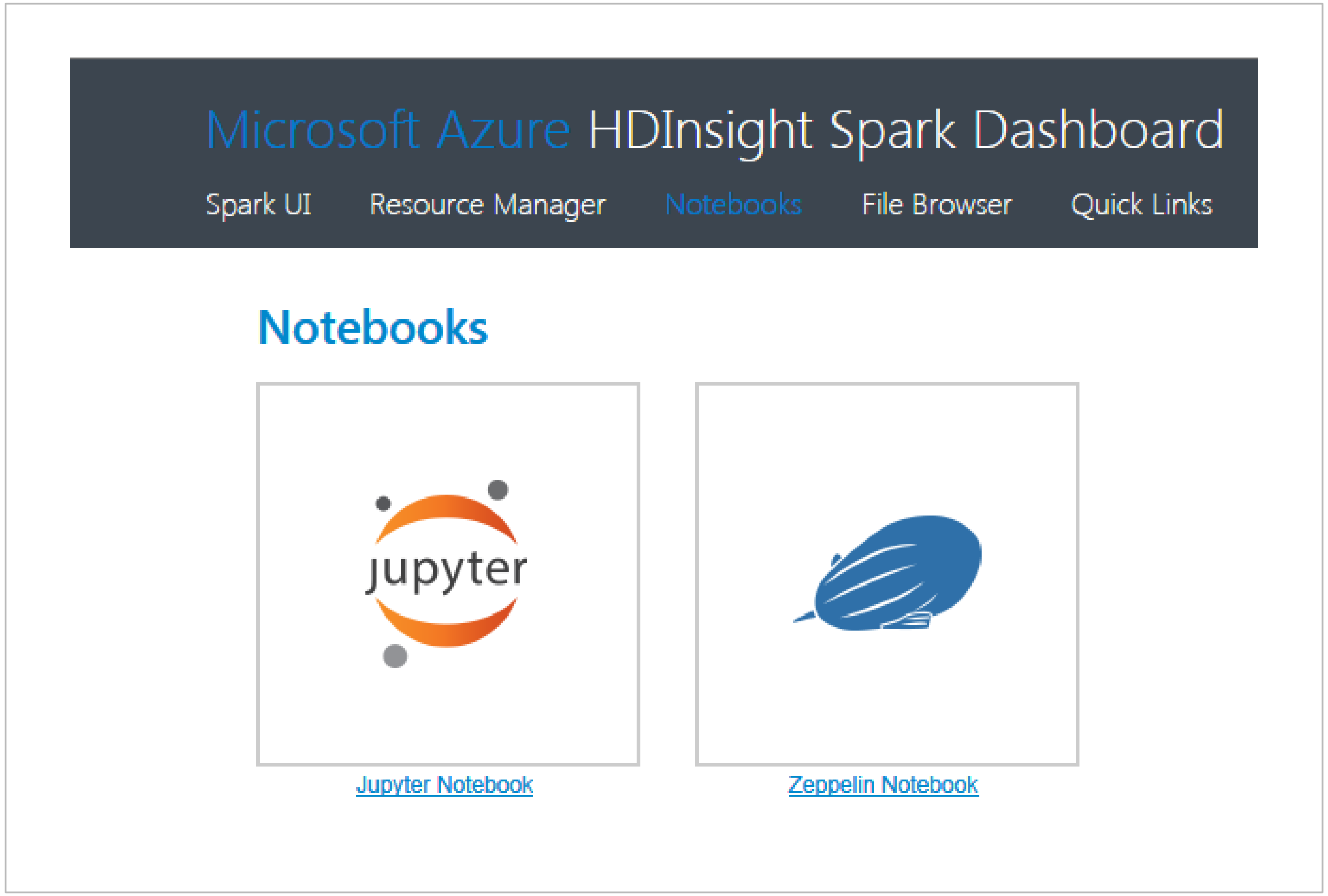
Transform data into intelligent action



HDInsight - Spark as a platform



Notebooks: Jupyter and Zeppelin



- HDInsight Spark supports two notebooks: Jupyter and Zeppelin
- The links to notebooks are available on the Dashboard
- The notebooks are pre-installed and can be launched by just clicking on the links. No other setup is required such as launching any background services. HDInsight Spark will start the Jupytr Kernels or Zeppelin Servers
- The notebooks belong to a subscription and are persistent across clusters. They will not be any or all Spark cluster are shutdown.

Machine Learning with Anaconda

- Anaconda is a free, enterprise-ready Python distribution for large-scale data processing, predictive analytics, and scientific computing
- It include 330+ of the most popular Python packages for science, math, engineering, data analysis
- Anaconda is installed as part of HDInsight Spark. The libraries will be updated at a regular cadence

R Server: scale-out R, Enterprise Class!

100% compatible with open source R

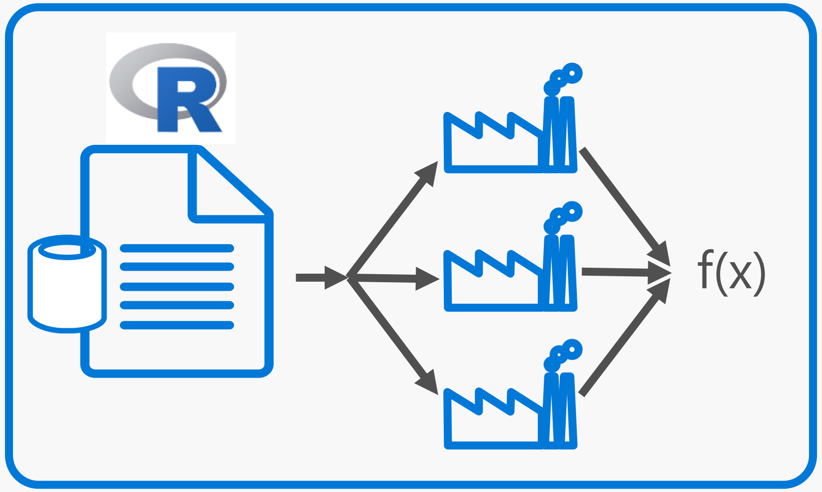
Full ecosystem access: any code/package that works today with R will work in R Server

Wide range of scalable and distributed R functions

Examples: rxDataStep(), rxSummary(), rxGlm(), rxDForest(), rxPredict()

Ability to parallelize R functions

Ideal for parameter sweeps, simulation, scoring, transformations



ScaleR Package - Big Data Functions

Data Step

- Data import – Delimited, Fixed, SAS, SPSS, OBDC
- Variable creation & transformation
- Recode variables
- Factor variables
- Missing value handling
- Sort, Merge, Split
- Aggregate by category (means, sums)

Descriptive Statistics

- Min / Max, Mean, Median (approx.)
- Quantiles (approx.)
- Standard Deviation
- Variance
- Correlation
- Covariance
- Sum of Squares (cross product matrix for set variables)
- Pairwise Cross tabs
- Risk Ratio & Odds Ratio
- Cross-Tabulation of Data (standard tables & long form)
- Marginal Summaries of Cross Tabulations

Statistical Tests

- Chi Square Test
- Kendall Rank Correlation
- Fisher's Exact Test
- Student's t-Test

Sampling

- Subsample (observations & variables)
- Random Sampling

Predictive Models

- Sum of Squares (cross product matrix for set variables)
- Quantiles (approx.)
- Generalized Linear Models (GLM) exponential family distributions: binomial, Gaussian, inverse Gaussian, Poisson, Tweedie. Standard link functions: cauchit, identity, log, logit, probit. User defined distributions & link functions.
- Covariance & Correlation Matrices
- Logistic Regression
- Classification & Regression Trees
- Predictions/scoring for models
- Residuals for all models

Variable Selection

- Stepwise Regression

Simulation

- Simulation (e.g. Monte Carlo)
- Parallel Random Number Generation

Cluster Analysis

- K-Means

Classification

- Decision Trees
- Decision Forests
- Gradient Boosted Decision Trees
- Naïve Bayes

Custom Development

- rxDataStep
- rxExec
- PEMA-R API Custom Algorithms

New package - MML

Learner	Task				Scalability			
	Binary	Multi	Regression	Other	# cols	# rows	# CPUs	Nodes
Fast Linear	✓		✓		~1Bil.	∞	mult-proc	1
Fast Tree	✓		✓		~30k	RAM-bound	mult-proc	1
Fast Forest	✓		✓		~30k	RAM-bound	mult-proc	1
Logistic Regression	✓	✓			~1Bil.	∞	single proc	1
						RAM-bound	mult-proc	1
Neural Network	✓	✓	✓		~10k	∞	mult-proc GPU	1
OneClassSvm				anomaly	~1k	RAM-bound	single proc	1

New package - MML

Featurizer	Function	Description
Text	featurizeText()	Includes: • Language detection • Word and character tokenization (multi-lang) • Stop-word removal, text normalization • Word and Char N-grams • TF, IDV, TF-IDF
Categorical	Categorical() categoricalHash()	Categorical indicator vector Categorical hash indicator vector
Feature Selection	Count-based Mutual information	Filter out spurious features to increase accuracy and reduce speed

Open source R

```
mydata <-   read.csv("http://www.ats.ucla.edu/stat/data/binary.csv")

mylogit <-     glm(admit ~ gre + gpa + rank, data = mydata,
                   family = "binomial")
```

R Server

Switch functions

```
mydata <- RxTextData("/data/binary.csv", fileSystem = hdfsFS)

mylogit <- rxLogit(admit ~ gre + gpa + rank, data = mydata)
```

R Server parallelized by Spark

Switch compute context

```
rxSetComputeContext( RxSpark(…) )

mydata <- RxTextData("/data/binary.csv", fileSystem = hdfsFS)

mylogit <- rxLogit(admit ~ gre + gpa + rank, data = mydata)
```

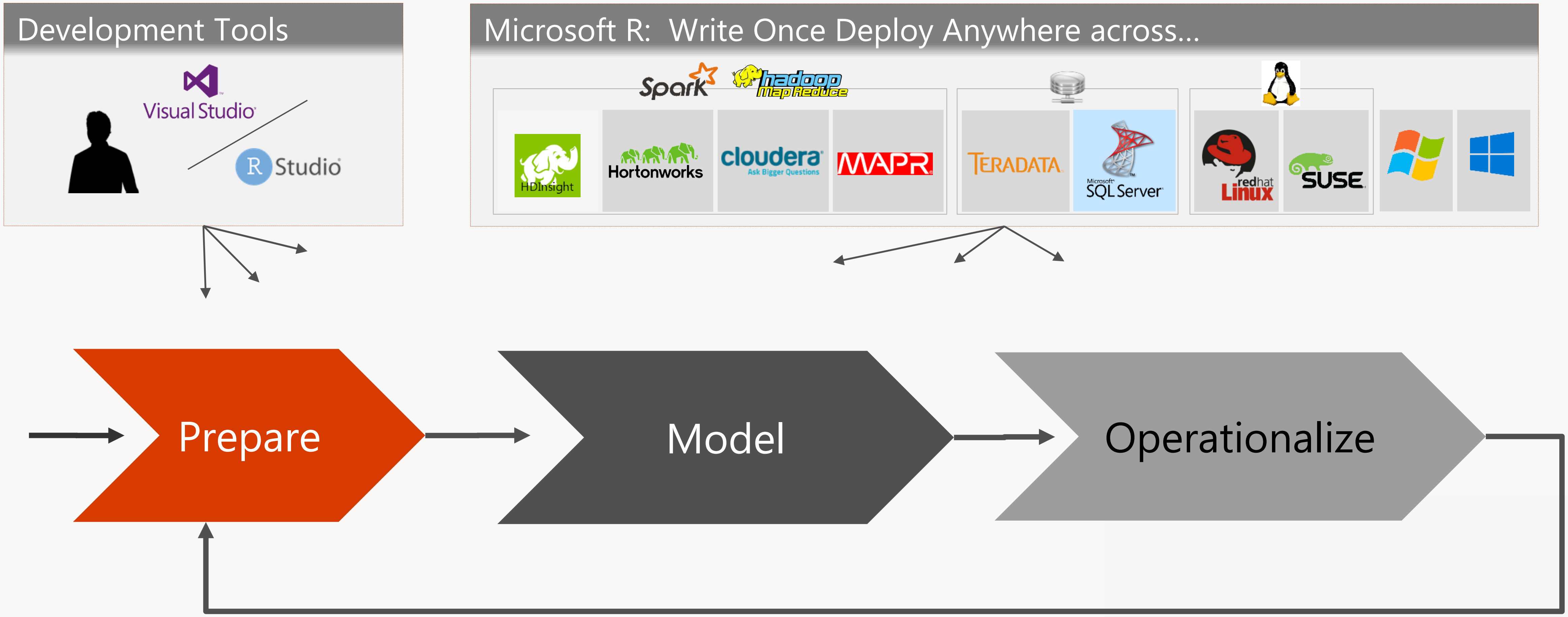
R Server parallelized by Map-Reduce

Switch compute context

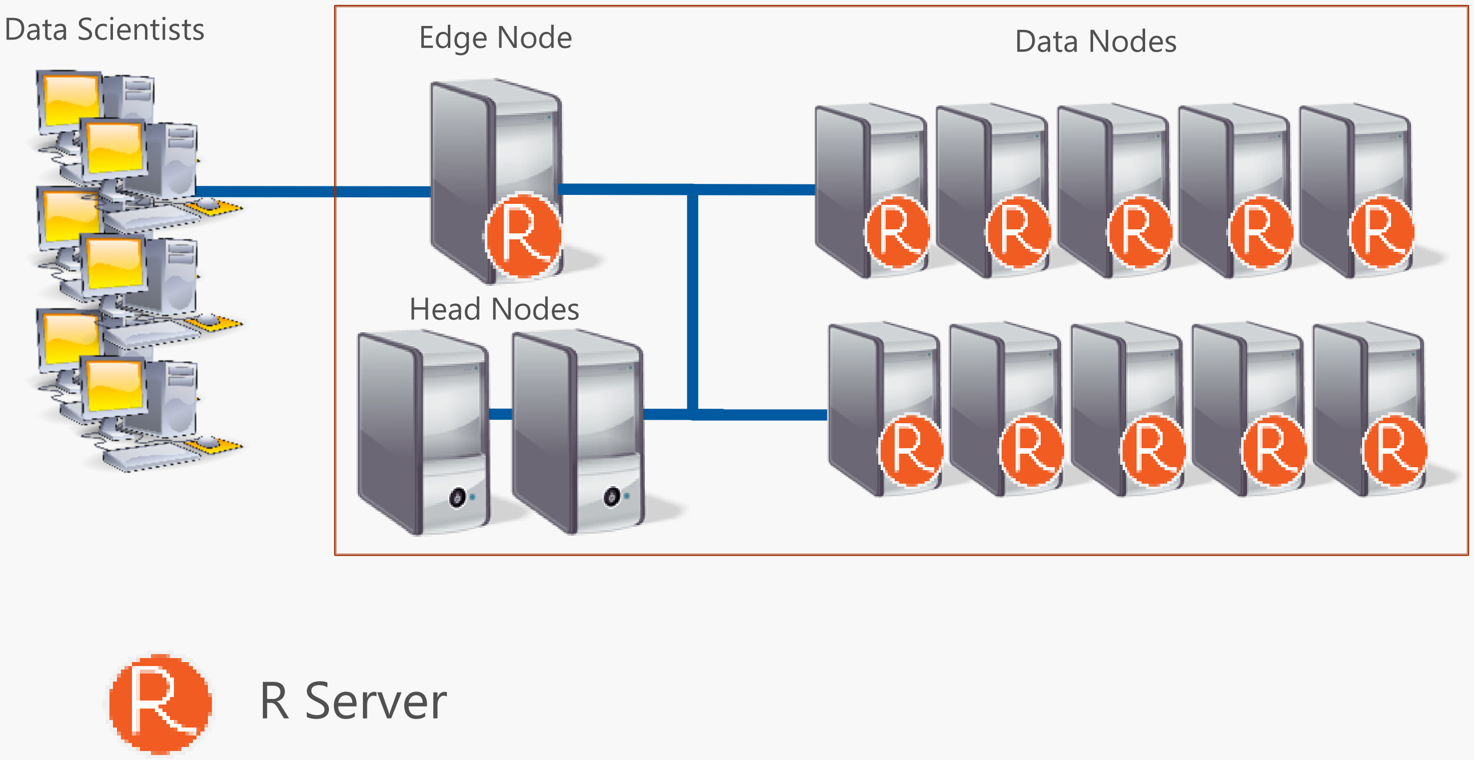
```
rxSetComputeContext( RxHadoopMR(…) )

mydata <- RxTextData("/data/binary.csv", fileSystem = hdfsFS)

mylogit <- rxLogit(admit ~ gre + gpa + rank, data = mydata)
```

Note: New Spark compute context is 3-7x faster than Map reduce

End-to-End Analytics for Hybrid Environments



R Server on HDInsight



Create Azure Data Lake Storage

- In Azure portal, create Azure Data Lake Store
- Name: datalakestoremilan
- Resource Group: datalakestoremilanresource
- Location: Central US
- Pin to dashboard

Create HDInsight cluster for analytics

1. Configure Basics
- On Azure Portal, add HDInsight
- Cluster name: hdimilan
- Cluster type: R Server
- Leave admin and SSH user name as default
- Provide a password (please remember the password). Check "Use same password for cluster".
- Resource group name: hdimilanresource

2. Set Storage Settings
- Primary storage type: Data Lake Store
- Root path: /hdimilan/
- Create new service principal
- Service principal name: servprin
- Leave default start and expiration dates
- Provide password
- Access: select the data lake store, and click Run.

3. Confirm configuration
- Edit Cluster Size. Change number of worker nodes to 2. Check that the price decreases.

Deploy R model to Azure ML Web Service

- Once cluster is deployed, click on R Server Dashboard, then R Studio Server
- First prompt, enter admin username and password
- Second prompt, enter SSH username and password
- Upload the script “Deploy Azure ML web service.R”
- Replace wsID and wsAuth values
  - In Azure ML workspace, go to Settings tab, get the WS ID under Names tab. Get the WS authorization under Authorization Tokens tab