Operational Machine Learning

Using Microsoft Technologies for Applied Data Science

Khalid M. Salama, Ph.D.
Business Insights & Analytics
Hitachi Consulting UK

Outline

- Introduction to Data Science
- From Experimental Data Science to Operational Machine Learning
- MS Technologies for Data Science & Advanced Analytics
- Demos & Screenshots
- Concluding Remarks

Hitachi Consulting

HITACHI
Inspire the Next

Introduction to Data Science and Machine Learning

Data Science and Machine Learning

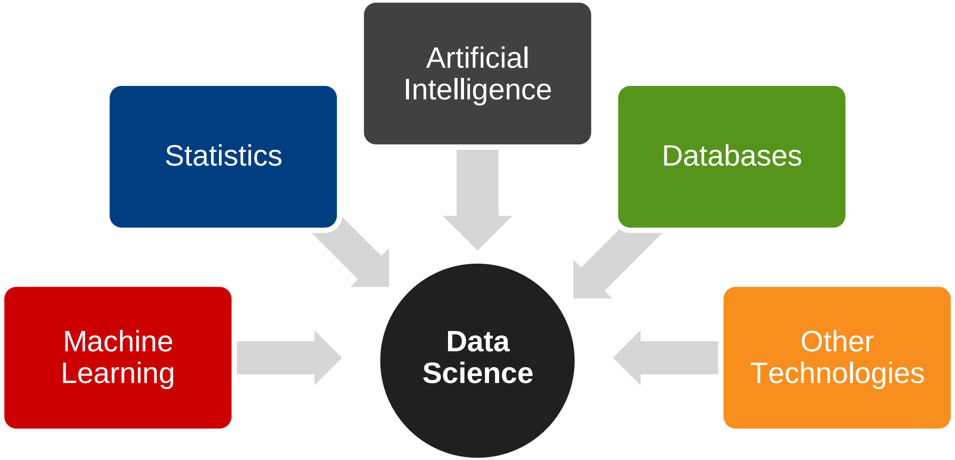
HITACHI
Inspire the Next

What?

“Data mining, an interdisciplinary subfield of computer science, is the computational process of **automatic discovering interesting and useful patterns in large data sets”**

Other Related Technologies:

- Visualization
- Big Data
- High Performance Computing
- Cloud Computing
- Others..

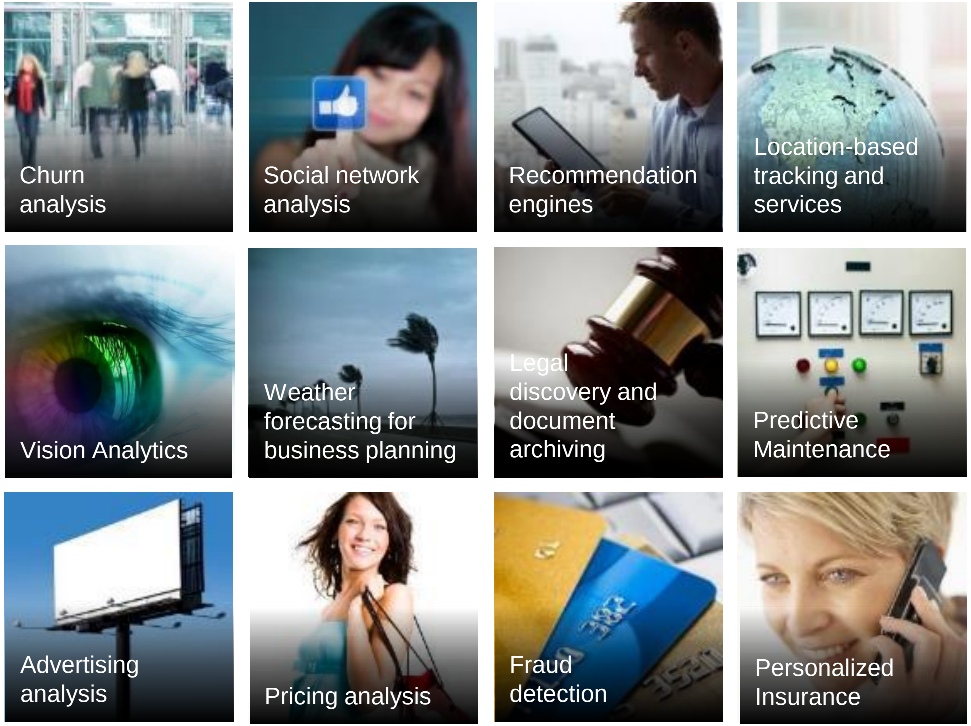


Data Science and Machine Learning

HITACHI
Inspire the Next

Why?

The objective of data science is to provide you with actionable insights to support decision making….



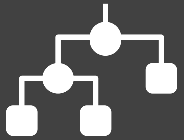
Data Science and Machine Learning

HITACHI
Inspire the Next

How?

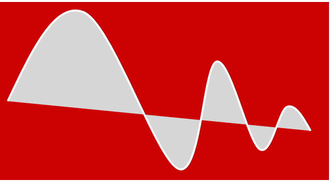
Classification Learning

Build a model that can predict the target class of an input case



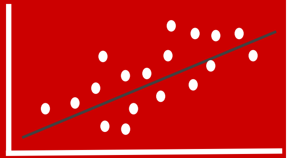
Time Series Analysis

Analysis of temporal data to forecast future values



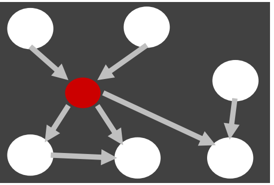
Regression Modeling

Build a model that can estimate the response value given an input case



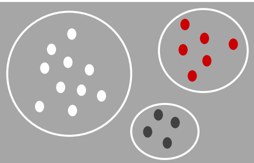
Probabilistic Modeling

Compute the probability of an event to occur given a set of conditions



Cluster Analysis

Discover natural groupings within the data points



Similarity Analysis

Identify similar cases to a given input case based on the input features

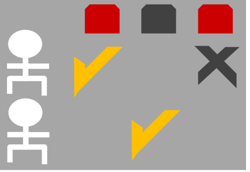
Association Rule Discovery

Extract frequent patterns present in the data

```
IF .. AND .. AND .. THEN A
ELSE IF .. AND .. THEN C
ELSE IF .. AND .. THEN B
..
..
ELSE C
```

Collaborative Filtering

Filtering of information using techniques involving collaboration viewpoints



Hitachi Consulting

HITACHI
Inspire the Next

From Experimental Data Science to Operational Machine Learning

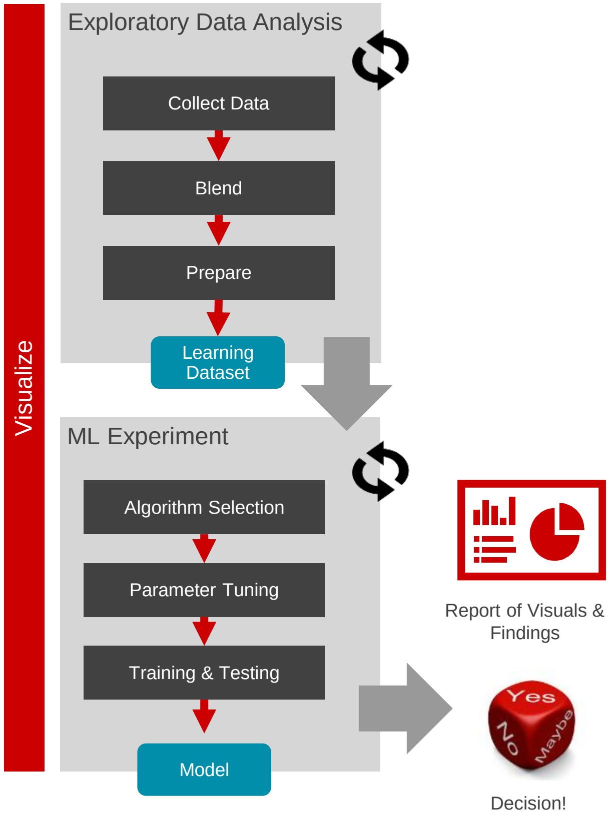
Data Science Activities

HITACHI
Inspire the Next

Experimentation vs. Operationalization

Data Analysis & Experimentation

- Interactive
- Easy to perform
- Rich Visualizations



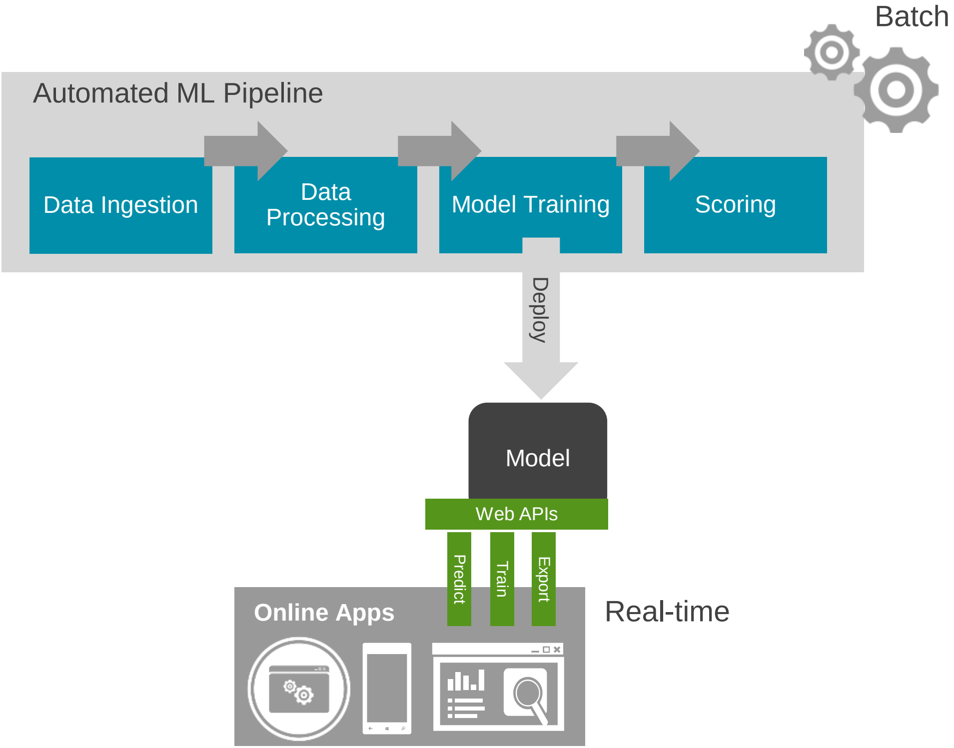
Data Science Activities

HITACHI
Inspire the Next

Experimentation vs. Operationalization

Operational ML Pipelines

- Pipelined (ETL Integration)
- Scalable
- Apps Integration



Hitachi Consulting

HITACHI
Inspire the Next

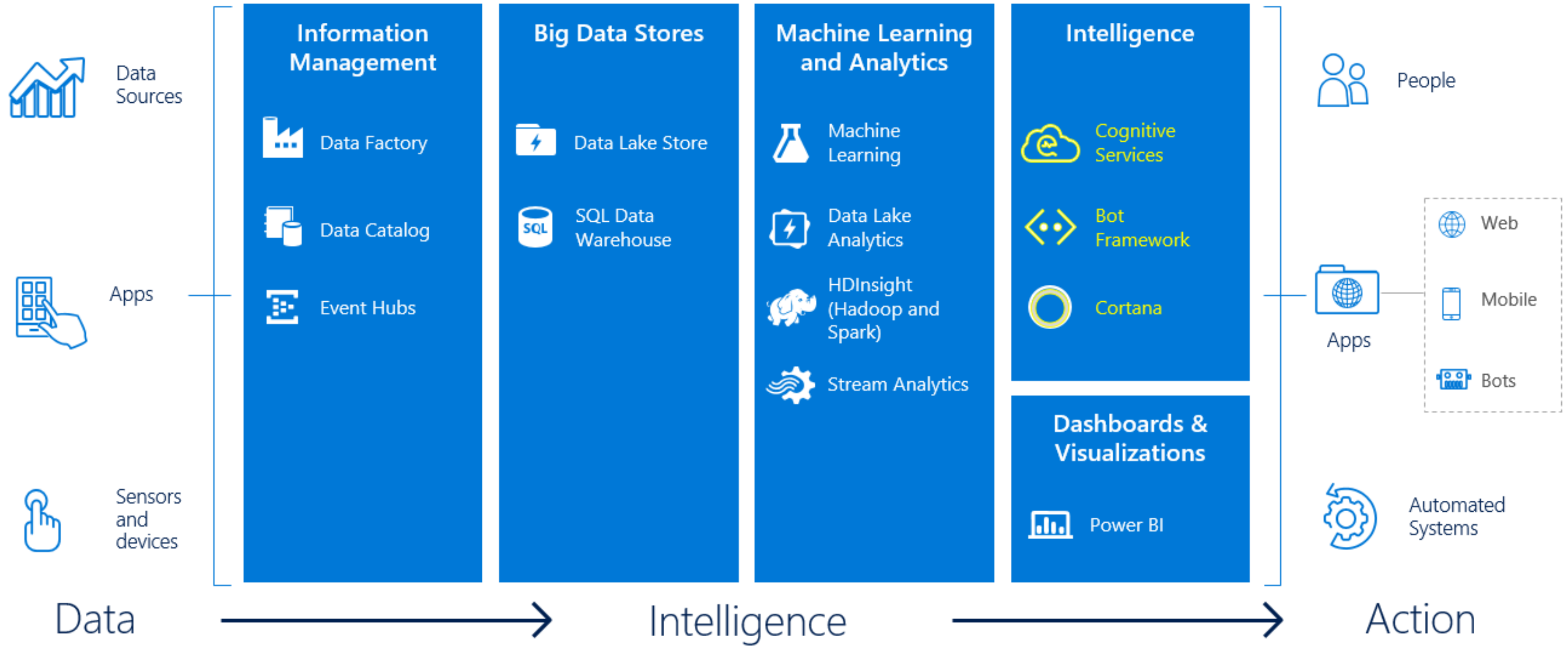
Microsoft Advanced Analytics Technologies

Microsoft Advanced Analytics

HITACHI
Inspire the Next

Cortana Intelligence Suite

https://gallery.cortanaintelligence.com/



Microsoft Advanced Analytics

HITACHI
Inspire the Next

Data Science, Machine Learning, & Intelligence

Azure Machine Learning

Microsoft R Server – SQL Server R Services

Data Mining – SQL Server Analysis Services

Spark ML – Azure HDInsight

Cognitive Features – Azure Data Lake Analytics

Azure Cognitive Services

Microsoft Bot Framework

Microsoft Azure Machine Learning

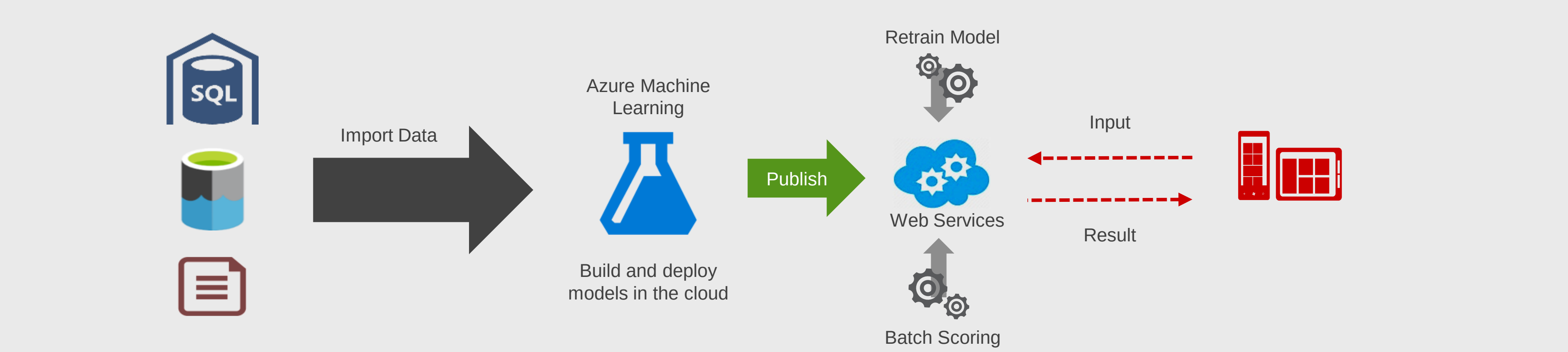
Azure Machine Learning

MS Cloud-native Data Science

- Cloud-based Machine Learning Services
- Interactive Data Science Studio
- Rich built-in functionality
- Imports data from everywhere
- Easy to **develop and productionize** – Web Services
- Extensible via R and Python scripts

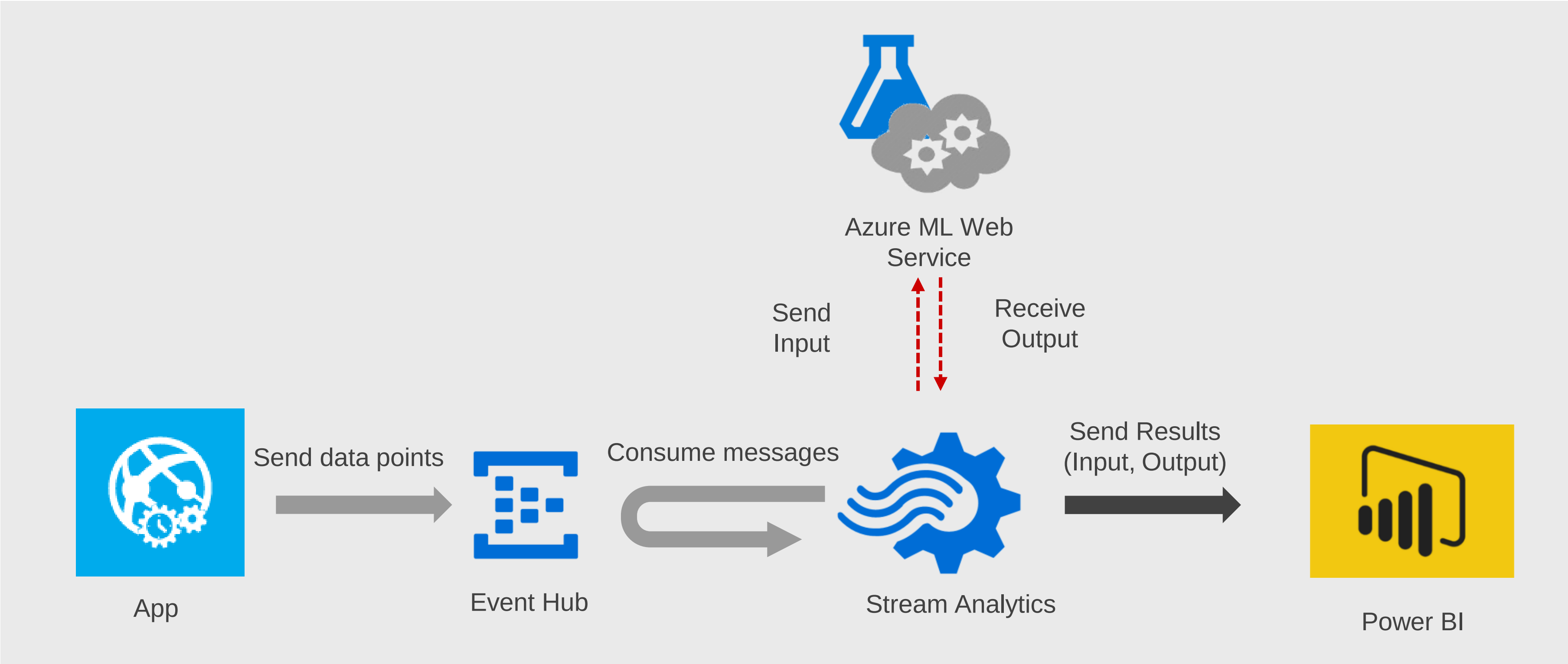
Limitations

- Only Cloud-based (Data Regulations)
- Scalability – Maximum dataset size = 10GB
- Microsoft R Open is not supported, yet
- No Source Control



Azure Machine Learning

Real-time Predictions



Azure Machine Learning

HITACHI
Inspire the Next

Built-in Features

- Saved Datasets
- 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

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
- Export Data
- Import Data
- Unpack Zipped Datasets

Filter
- Apply Filter
- FIR Filter
- IIR Filter
- Median Filter
- Moving Average Filter
- Threshold Filter
- User Defined Filter

Data Transformation
- Filter
- Learning with Counts
- Manipulation
- Sample and Split
- Scale and Reduce

Manipulation
- Add Columns
- Add Rows
- Apply SQL Transformat...
- Clean Missing Data
- Convert to Indicator Va..
- Edit Metadata
- Group Categorical Valu..
- Join Data
- Remove Duplicate Rows
- Select Columns in Data..
- Select Columns Transfo..
- SMOTE

Statistical Functions
- Apply Math Operation
- Compute Elementary Statist...
- Compute Linear Correlation
- Evaluate Probability Function
- Replace Discrete Values
- Summarize Data
- Test Hypothesis using t-Test

Python Language Modules
- Execute Python Script

R Language Modules
- Create R Model
- Execute R Script

Feature Selection
- Filter Based Feature Selectio..
- Fisher Linear Discriminant A...
- Permutation Feature Import...

Text Analytics
- Feature Hashing
- Named Entity Recognition
- Score Vowpal Wabbit Versio..
- Score Vowpal Wabbit Versio..
- Score Vowpal Wabbit Versio..
- Train Vowpal Wabbit Versio...
- Train Vowpal Wabbit Versio...
- Train Vowpal Wabbit Versio...

Sample and Split
- Partition and Sample
- Split Data

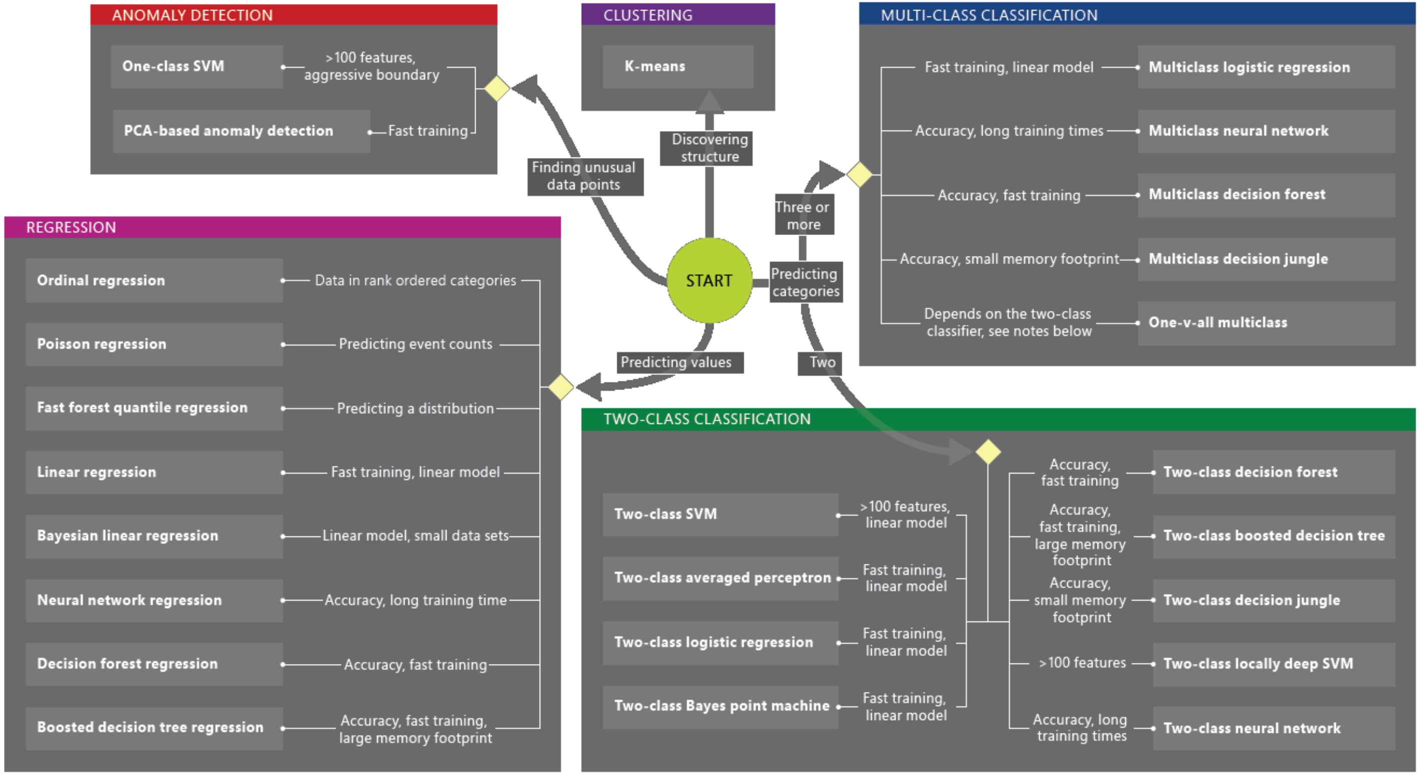
Scale and Reduce
- Clip Values
- Group Data into Bins
- Normalize Data
- Principal Component A..

Machine Learning
- Evaluate
  - Cross Validate Model
  - Evaluate Model
  - Evaluate Recommend...
- Initialize Model
  - Anomaly Detection
  - Classification
  - Clustering
  - Regression
- Score
  - Apply Transformation
  - Assign Data to Clusters
  - Score Matchbox Reco...
  - Score Model
- Train
  - Sweep Clustering
  - Train Anomaly Detect...
  - Train Clustering Model
  - Train Matchbox Reco...
  - Train Model
  - Tune Model Hyperpa...

Azure Machine Learning

HITACHI
Inspire the Next

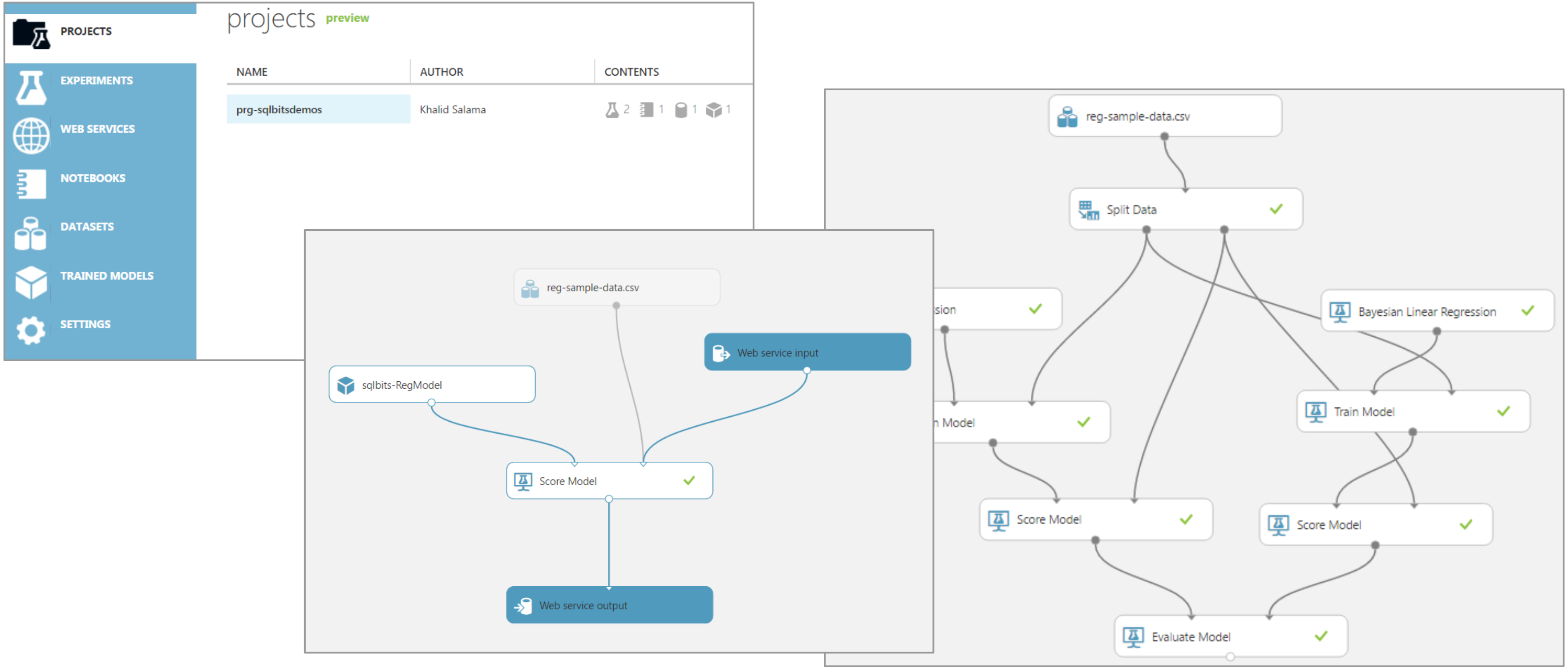
Algorithms Cheat Sheet



Azure Machine Learning

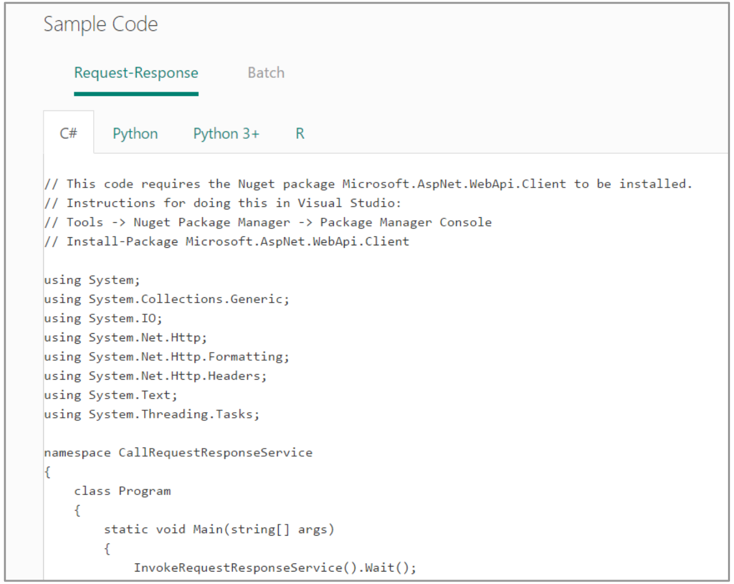
HITACHI
Inspire the Next

ML Studio

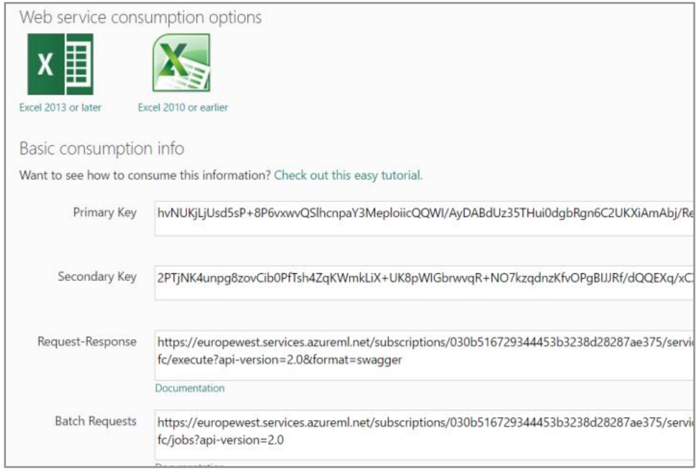


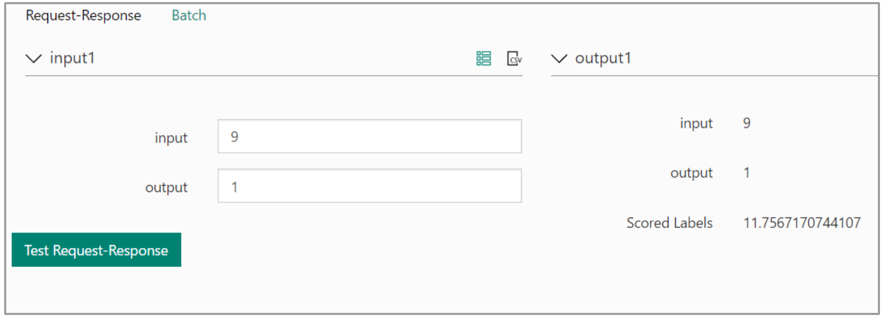
Azure Machine Learning

Web Service



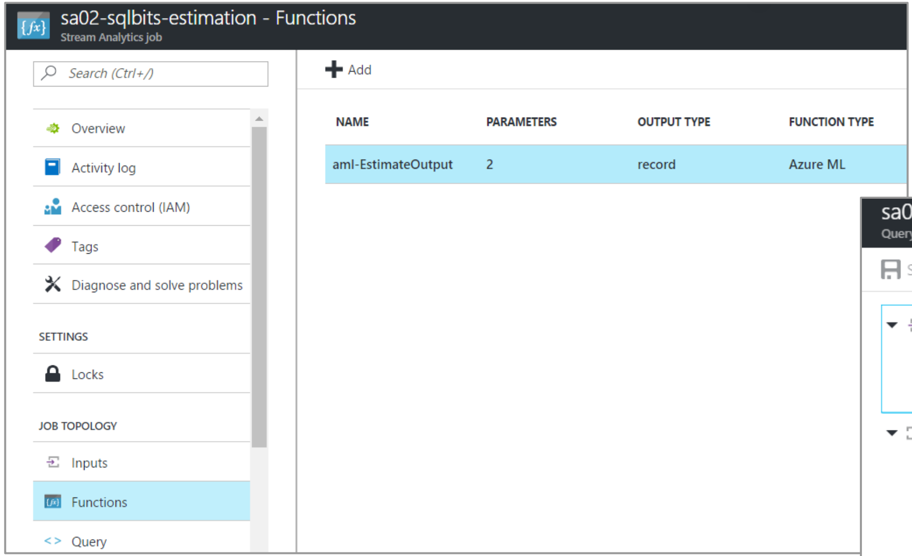

```
// This code requires the Nuget package Microsoft.AspNet.WebApi.Client to be installed.
// Instructions for doing this in Visual Studio:
// Tools -> Nuget Package Manager -> Package Manager Console
// Install-Package Microsoft.AspNet.WebApi.Client

using System;
using System.Collections.Generic;
using System.IO;
using System.Net.Http;
using System.Net.Http.Formatting;
using System.Net.Http.Headers;
using System.Text;
using System.Threading.Tasks;

namespace CallRequestResponseService
{
    class Program
    {
        static void Main(string[] args)
        {
            InvokeRequestResponseService().Wait();
```

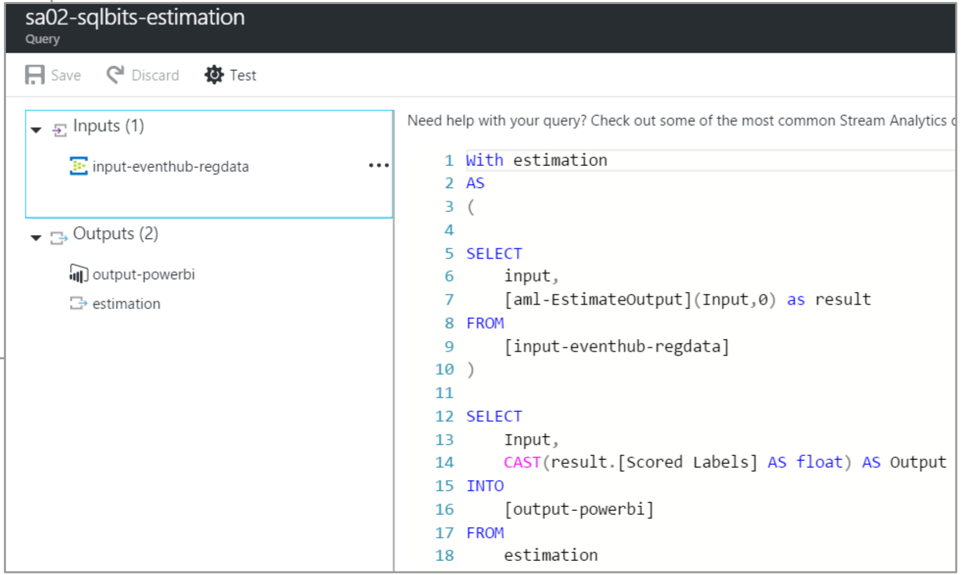




Azure Machine Learning

Stream Analytics Integration





Azure Machine Learning

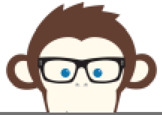
AzureML R Library

```r
library(AzureML)
ws <- workspace(id = "1f2b56bcf5fe4f3f9c32ee4371
auth = "VsvDbAo+noBzPxphaoZZfdyY57T6QHDpdyM4P7W(
api_endpoint = "https://europewest.studio.azurer
management_endpoint = "https://europewest.manage

head(experiments(ws))

data_file = "C:/Master/data.csv"
data = read.csv(data_file, header = TRUE)
model = lm(output ~ input, data = data)

test = data.frame(input = c(1, 10, 100))
predict(model, test)

predictOutput = function(input) {

    data = data.frame(c(input))
    colnames(data) = c("input")
    output = predict(model, data)
    return(output)
}
```

```r
api <- publishWebService(ws, fun = predictOutput, name = "aml-predictOutput",
                         inputSchema = list(input = "numeric"),
                         outputSchema = list(output = "numeric")
)
```

Microsoft R Server

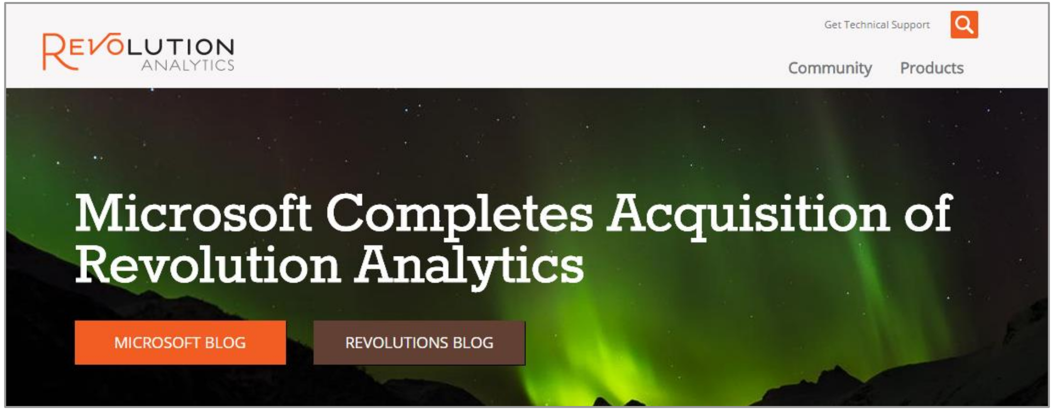
Microsoft R Server



R in Microsoft World

Microsoft R Open (MRO)

- Based on latest Open Source R (3.2.2.) - Built, tested, and distributed by Microsoft
- More **efficient and multi-threaded** computation
- **Enhanced by Intel Math Kernel Library** (MKL) to speed up linear algebra functions
- Compatible with all R-related software



Microsoft R Server

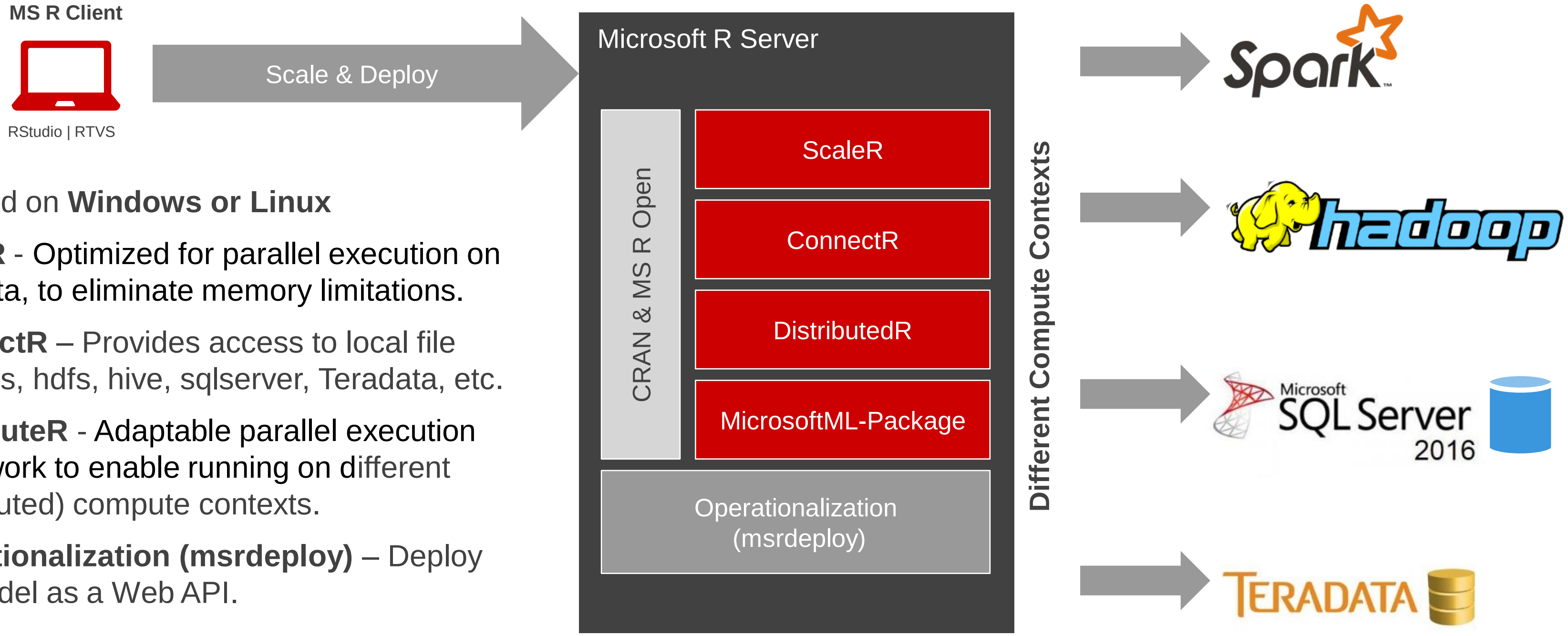
HITACHI
Inspire the Next

Comparison

	CRAN	MRO	MRS
Data size	In-memory	In-memory	**In-memory & disk**
Efficiency	Single threaded	**Multi-threaded**	Multi-threaded, **parallel processing 1:N servers**
Support	Community	Community	Community + Commercial
Functionality	7500+ innovative analytic packages	7500+ innovative analytic packages	7500+ innovative packages + commercial parallel high-speed functions
Licence	Open Source	Open Source	Commercial license.

Microsoft R Server

Components and Compute Contexts



- Installed on **Windows or Linux**
- **ScaleR** - Optimized for parallel execution on Big Data, to eliminate memory limitations.
- **ConnectR** – Provides access to local file systems, hdfs, hive, sqlserver, Teradata, etc.
- **DistributeR** - Adaptable parallel execution framework to enable running on different (distributed) compute contexts.
- **Operationalization (msrdeploy)** – Deploy the model as a Web API.

Microsoft R Server

HITACHI
Inspire the Next

Microsoft R Server – ScaleR Example

Check Environment

```
Revo.version
Revo.home()
rxGetComputeContext()
#rxSetComputeContext()
```

Load XDF

```
titanic_xdf = "data/titanic.xdf"
rxImport(titanic_csv, titanic_xdf, colClasses = col_classes, overwrite = TRUE)
titanic_xdata <- RxXdfData(titanic_xdf)
rxGetInfo(titanic_xdata, getVarInfo = TRUE, numRows = 1)
rxSummary( ~ Survived, titanic_xdata)
```

Prepare Data – Process XDF

```
rxDataStep(titanic_xdata, titanic_xdata,
  transforms = list(
    Survived = factor(Survived, levels = 0:1, labels = c('No', 'Yes')),
    FareToAgeRatio = Fare/Age
    ),
  overwrite = TRUE)
```

```
prepare_data <- function(data) {
    age_mean = mean(data$Age, na.rm = TRUE)
    data$Age[is.na(data$Age)] <- age_mean
    return(data)
}

rxDataStep(titanic_xdata, titanic_xdata,
           transformFunc = prepare_data,
           overwrite = TRUE)
```

Build Predictive Model

```
rx_decision_tree <- rxDTree(Survived ~ Age + Sex + Fare + Pclass,
                            data = titanic_xdata, pruneCp = "auto",
                            reportProgress = 0)
```

Perform Prediction

```
test_data = data.frame(Age = c(30,20), Sex = c("male","female"))
predictions = rxPredict(rx_decision_tree, test_data)
head(predictions)
```

Microsoft R Server – ScaleR Functionality

Data Preparation

- 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)
- Multiple Linear Regression
- 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

Combination

- rxDataStep
- rxExec
- PEMA-R API Custom Algorithms

SQL Server (in-database) R Services

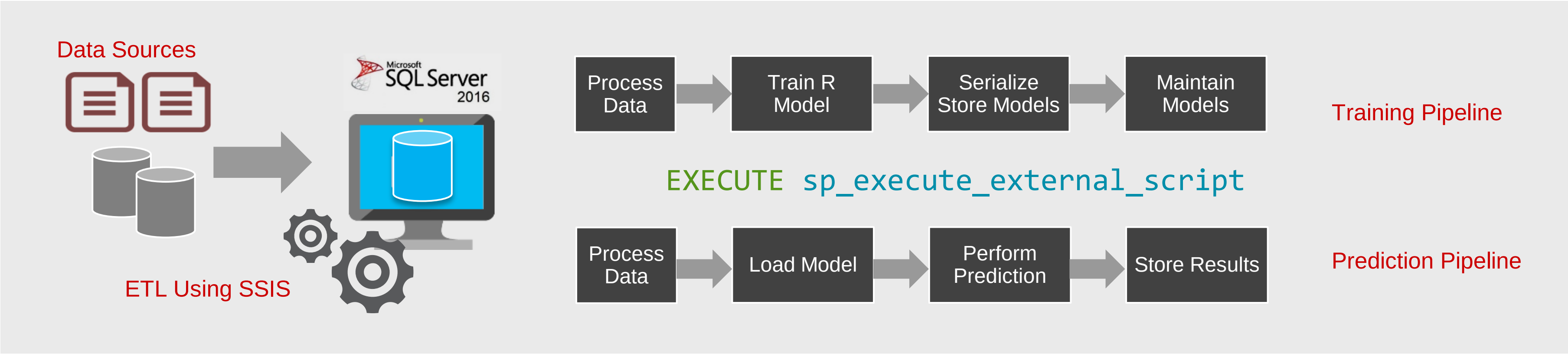
SQL Server R Services

In-database Analytics

- R Services (in-database) – Keep your analytics close to the data
- T-SQL Script – Can be **encapsulated in Stored Procedures**
- Models are **built, trained, saved** as part of the ETL process (SSIS)
- Used for **batch** prediction (as part of the ETL process)
- Visual Studio SQL Database Project, Source Controlled, etc.
- Uses Microsoft ScaleR libraries

Limitations

- Not supported in Azure SQL DB/DW, yet
- Not suitable for Interactive Data Science
- Only R, no python, yet.



SQL Server R Services

HITACHI
Inspire the Next

T-SQL Script

Configure

```
Exec sp_configure  'external scripts enabled', 1
Reconfigure  with override;
```

Build and Save Model

```
DECLARE @model varbinary(max);

EXEC sp_execute_external_script
     @language = N'R'

-- Begin Learn Model Script
     ,@script =
N'
    model <- lm(Output ~ Input, data = inputData);
    print(summary(model))
    modelbin <- serialize(model, NULL);
'
-- End Learn Model Script

     , @input_data_1 =
N'
    SELECT
        Input,
        Output
    FROM
        demo.Data;
'
      , @input_data_1_name = N'inputData'
      , @params = N'@modelbin varbinary(max) OUTPUT'
      , @modelbin = @model OUTPUT;

INSERT INTO demo.Models (Name,Model,ModifiedDate)
SELECT 'regModel-demo-v2',@model,GETDATE()
```

Model Summary

10 %

Results | Messages

```
STDOUT message(s) from external script:

Call:
lm(formula = Output ~ Input, data = inputData)

Residuals:
    Min      1Q  Median      3Q     Max
-56.193 -13.545  -1.576  10.008  62.790

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.0852     5.8035  -0.187    0.852
Input         1.9662     0.1105  17.794   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 20.82 on 48 degrees of freedom
Multiple R-squared:  0.8684,	Adjusted R-squared:  0.8656
F-statistic: 316.6 on 1 and 48 DF,  p-value: < 2.2e-16
```

Prediction

```
DECLARE @input_in FLOAT = 20;
DECLARE @model_in VARBINARY(MAX);

SELECT  @model_in = Model
FROM    demo.Models
WHERE   ModifiedDate IN (Select MAX(ModifiedDate) FROM demo.Models);

EXEC sp_execute_external_script
     @language = N'R'

-- Begin Predict
     ,@script =
N'
    mod <- unserialize(as.raw(model));
    output <- predict(mod, InputDataSet);

    InputDataSet$output = output
    data_output = InputDataSet

    print(data_output)
'
-- End Predict

    ,@input_data_1 = N'SELECT @Input AS Input;'
    ,@output_data_1_name = N'data_output'
    ,@params = N'@model varbinary(max),
                 @input float'
                 ,@model = @model_in
                 ,@input = @input_in
    WITH RESULT SETS (([Input] FLOAT, [Output] FLOAT));
```

Prediction Output

10 %

Results | Messages

	Input	Output
1	20	38.2391599966004

Microsoft Analysis Services Data Mining

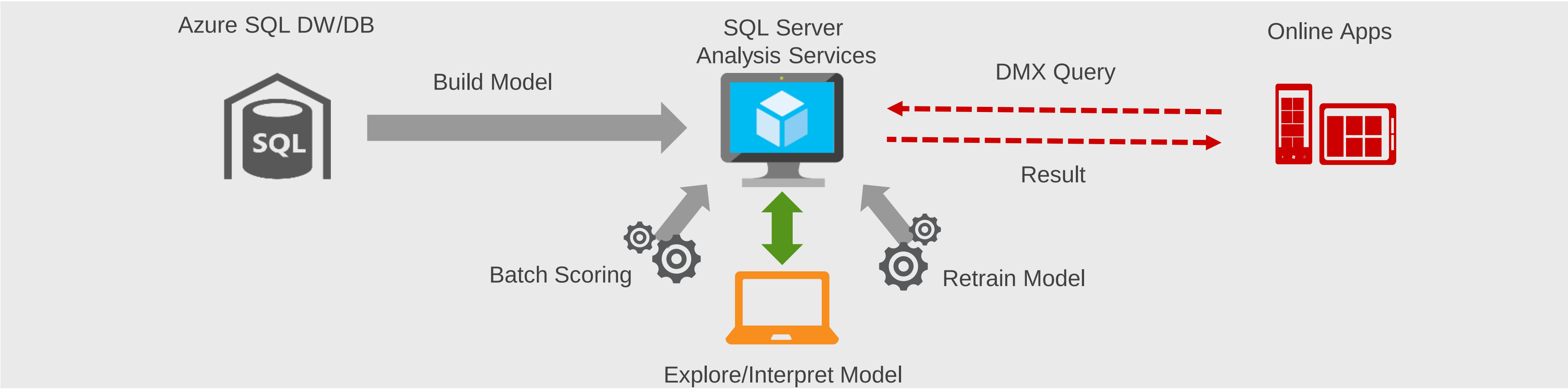
SQL Server Analysis Services

Data Mining

- Process data from many OLEDB and ODBC data sources
- Easy to **build, interpret, deploy, and productionize**
- SSIS Support – Tasks to Train & Predict
- **Interactive Visuals for model interpretation**
- Excel Integration – Data Mining Add-in

Limitations

- Limited Extensibility
- Limited Algorithms & Functionalities
- No Azure PaaS Service



SQL Server Analysis Services

HITACHI
Inspire the Next

Overview

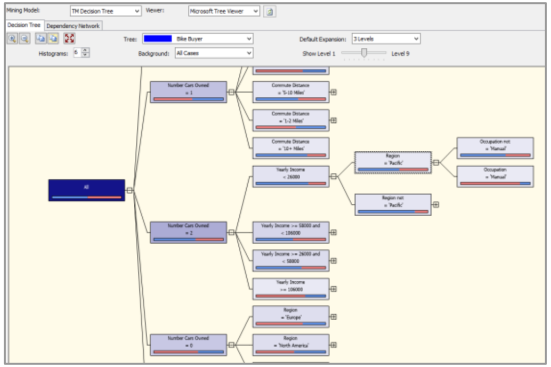
Microsoft SQL Server 2016

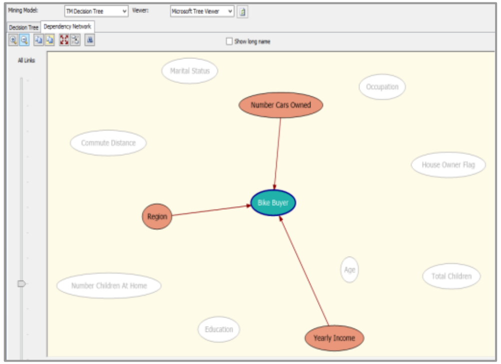
Data Source View → Mining Structure → Mining Algorithm → Mining Model

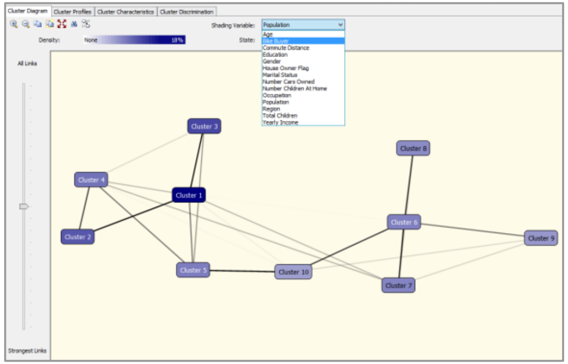
- Decision Tress
- Naïve-Bayes
- Linear Regression
- Neural Networks
- Association Rules
- Clustering
- Sequence Clustering
- Time Series

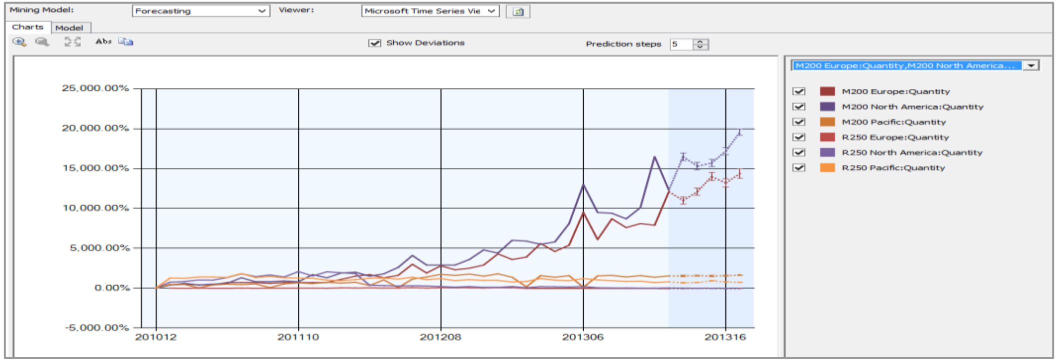
SQL Server Analysis Services

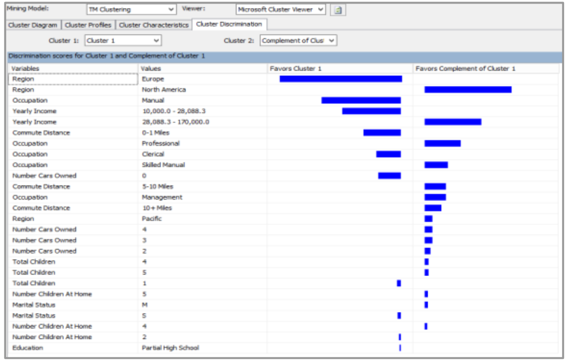
Visualizing Models





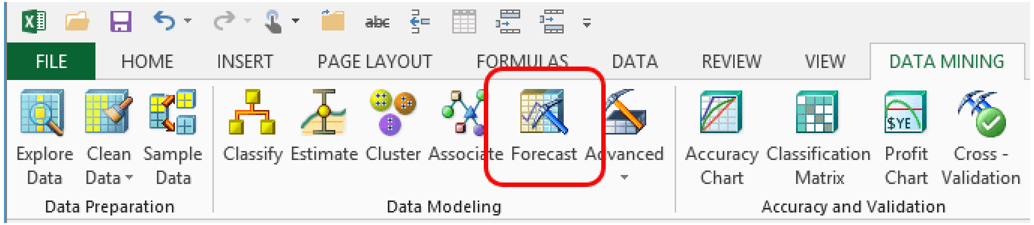


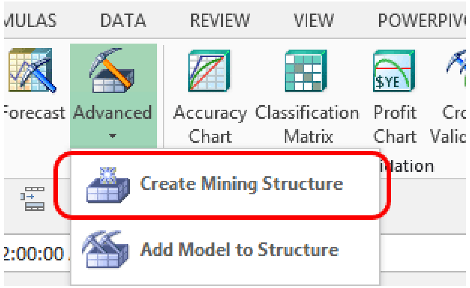


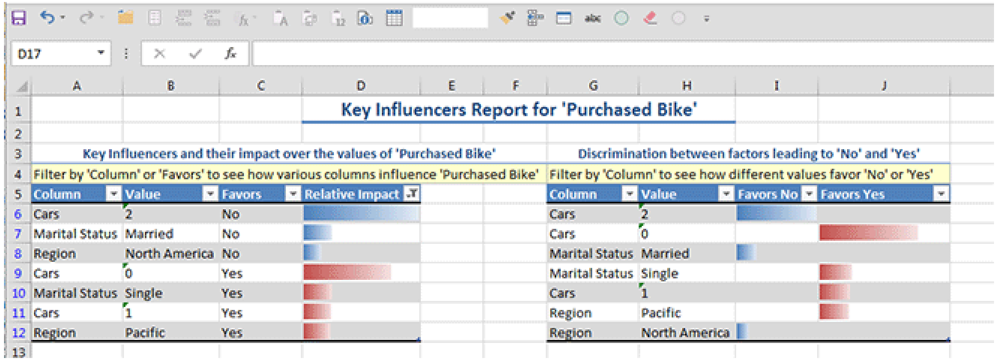


SQL Server Analysis Services

Excel Data Mining Add-in







Hitachi Consulting

HITACHI
Inspire the Next

Azure Cognitive Services

Azure Cognitive Services

Ready-to-use Intelligence

Language

Allow your apps to process natural language, evaluate sentiment and topics, and learn how to recognise what users want.

Language Understanding Intelligent Service

Teach your apps to understand commands from your users

Text Analytics API

Easily evaluate sentiment and topics to understand what users want

Web Language Model API

Use the power of predictive language models trained on web-scale data

Bing Spell Check API

Detecting and correcting spelling mistakes in your app

Speech

Processing spoken language in your applications

Bing Speech API

Convert speech to text and back again to understand user intent

Speaker Recognition API

Use speech to identify and authenticate individual speakers

Vision

State-of-the-art image processing algorithms to build more personalised apps by returning smart insights such as faces, images and emotion recognition.

Face API

Detect, analyse, organise and tag faces in photos

Emotion API

Personalise user experiences with emotion recognition

Search

Make your apps, web pages and other experiences smarter and more engaging with the Bing Search APIs.

Bing Search APIs

Search, image, video and news APIs for your apps

Bing Autosuggest API

Give your app intelligent autosuggest options for searches

Knowledge

Map complex information and data in order to solve tasks such as intelligent recommendations and semantic search.

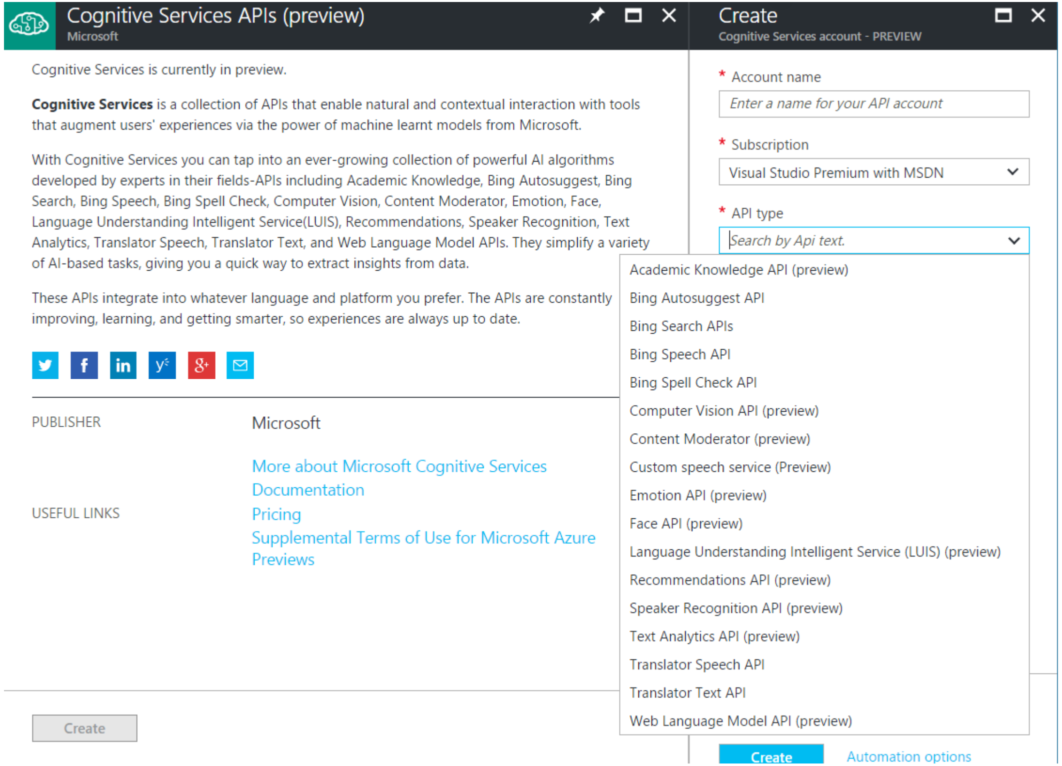
Recommendations API

Predict and recommend items that your customers want

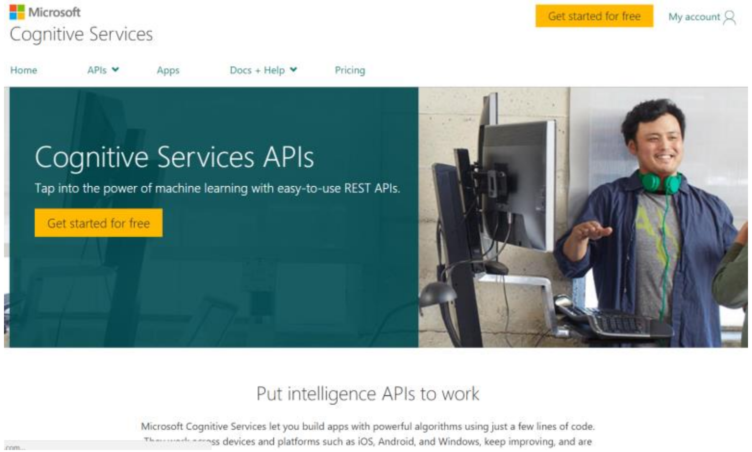
Azure Cognitive Services

HITACHI
Inspire the Next

Setup a Cognitive Services API



https://www.microsoft.com/cognitive-services/



Hitachi Consulting

HITACHI
Inspire the Next

Cognitive Features in Azure Data Lake Analytics

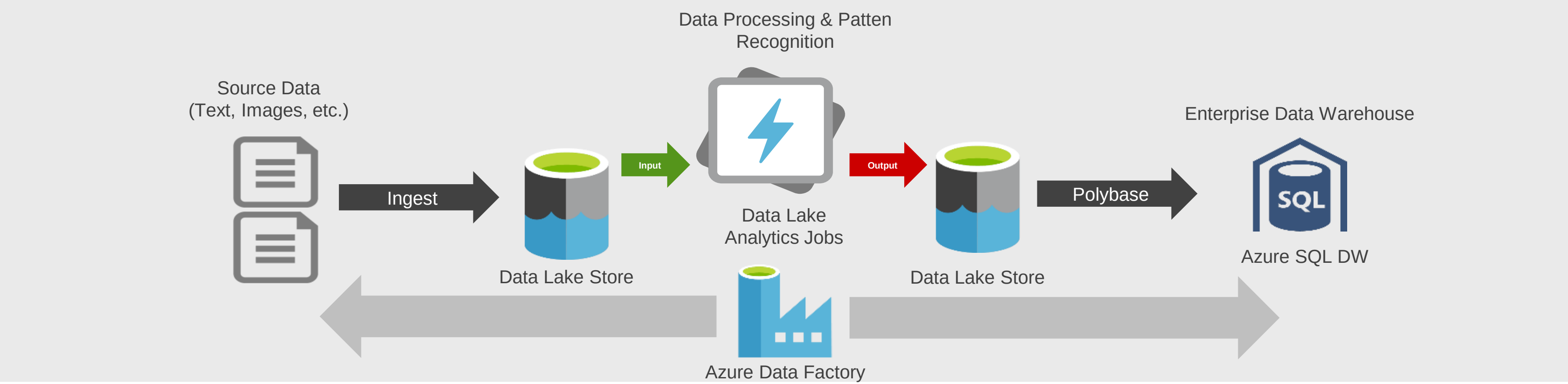
Azure Data Lake Analytics

Cognitive Features

- Pre-built intelligence – Text & Image Analysis
- **Integrated** with your **data processing pipelines** (DLA)
- Used for **batch** recognition (not singleton real-time)
- Scheduled & Automated using Azure Data Factory
- R & Python Extensions!
- **Scalable** – Suitable for Big Data

Limitations

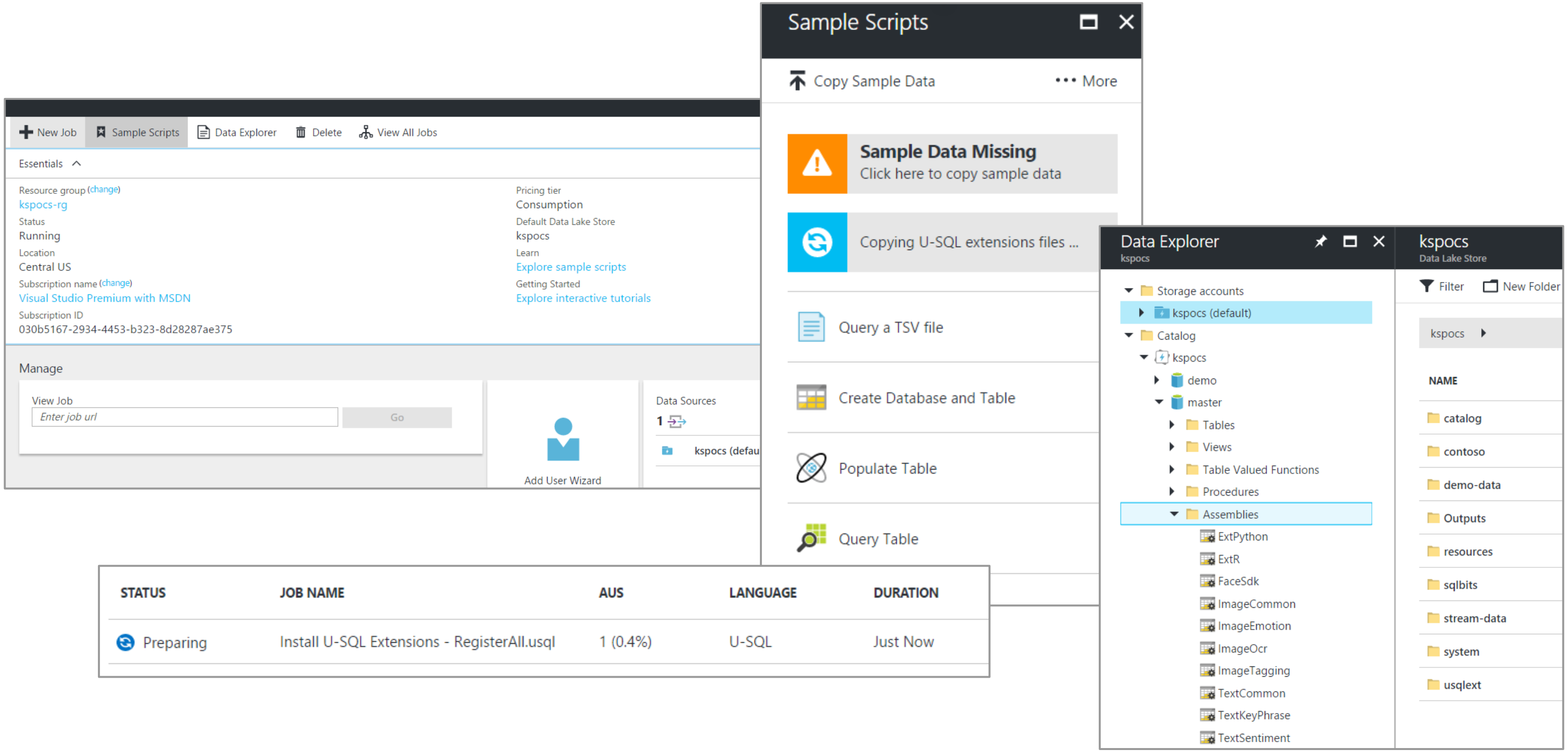
- Limited Features
- Not suitable for real-time scoring



Azure Data Lake Analytics

HITACHI
Inspire the Next

First-time Installation



Azure Data Lake Analytics

HITACHI
Inspire the Next

U-SQL Script

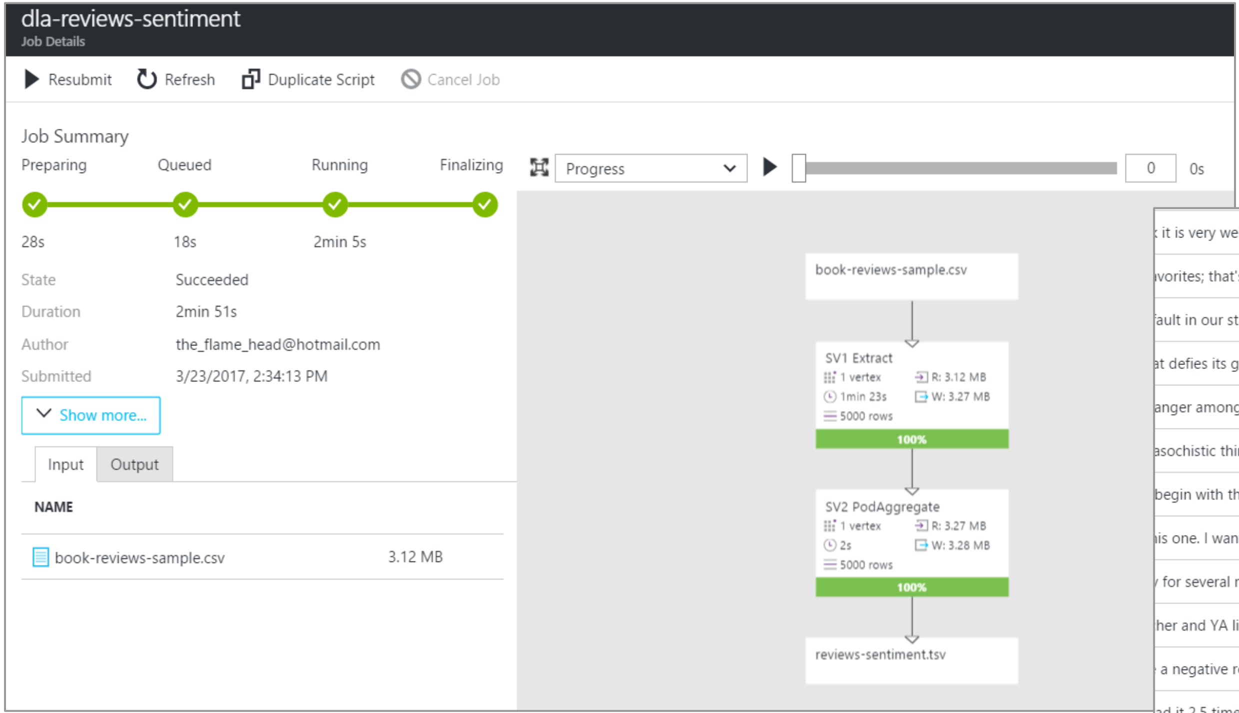
Submit Job | Data Explorer | Open File | Save As

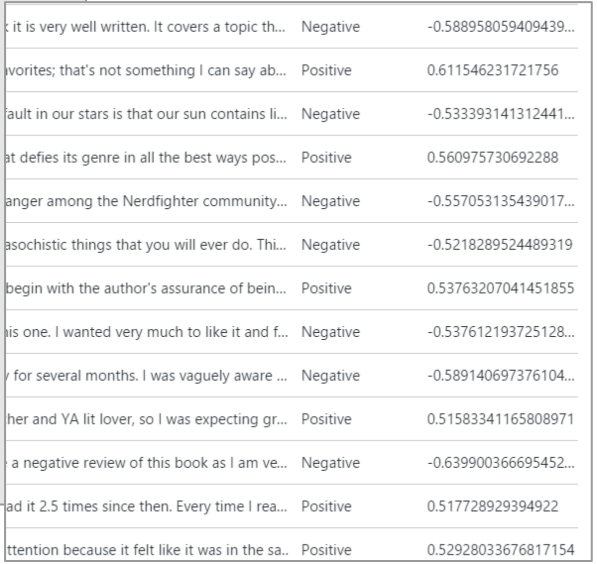
* Job Name: dla-reviews-sentiment
Priority: 1000
AUs: 1
Estimated Cost: 0.03 USD/minute

```
REFERENCE ASSEMBLY [TextCommon];
REFERENCE ASSEMBLY [TextSentiment];
REFERENCE ASSEMBLY [TextKeyPhrase];

DECLARE @input_file string = "sqlbits/input-data/book-reviews-sample.csv";
DECLARE @output_file string = "sqlbits/output-results/reviews-sentiment.tsv";

@input_data =
    EXTRACT Score decimal,
            Text string
    FROM @input_file
    USING Extractors.Csv();

@sentiment =
    PROCESS @input_data
    PRODUCE Score,
            Text,
            Sentiment string,
            Conf double
    READONLY Score,
             Text
    USING new Cognition.Text.SentimentAnalyzer(true);


OUTPUT @sentiment
```

Azure Data Lake Analytics

Execution & Output





: it is very well written. It covers a topic th...	Negative	-0.588958059409439...
ıvorites; that's not something I can say ab...	Positive	0.611546231721756
ault in our stars is that our sun contains li...	Negative	-0.533393141312441...
at defies its genre in all the best ways pos...	Positive	0.560975730692288
anger among the Nerdfighter community...	Negative	-0.557053135439017...
asochistic things that you will ever do. Thi...	Negative	-0.5218289524489319
begin with the author's assurance of bein...	Positive	0.53763207041451855
ıis one. I wanted very much to like it and f...	Negative	-0.537612193725128...
ı for several months. I was vaguely aware ...	Negative	-0.589140697376104...
her and YA lit lover, so I was expecting gr...	Positive	0.51583341165808971
a negative review of this book as I am ve...	Negative	-0.639900366695452...
ad it 2.5 times since then. Every time I rea...	Positive	0.517728929394922
ttention because it felt like it was in the sa..	Positive	0.52928033676817154

Spark ML on HDInsight

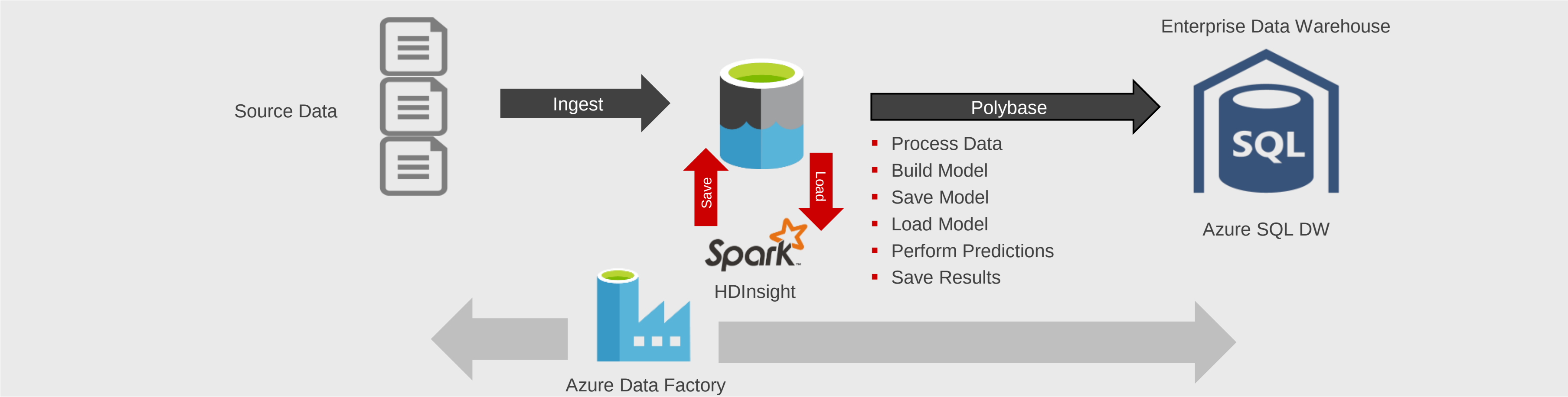
Spark ML on HDInsight

Scalable ML for Big Data

- Rich Spark ML Libraries
- **Scalable**, distributed, in-memory
- Extensible – Python, R, Java, Scala
- Suitable for Big Data - **Batch Model Training and Scoring**
- **Spark Streaming** for Real-time predictions
- Scheduled & Automated Using Azure Data Factory

Limitations

- Expensive to keep it up & running
- Slow to spin-up

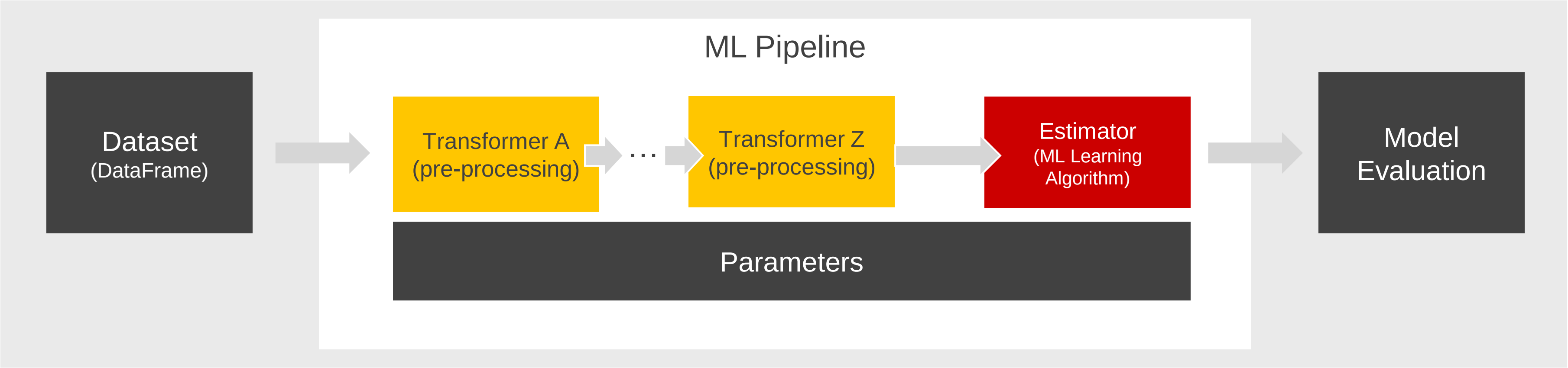


Spark ML on HDInsight

Spark ML Pipelines

Spark ML standardizes APIs for machine learning algorithms to make it easier to combine multiple task into a single pipeline, or workflow.

- **Transformers** – used for data pre-processing. Input: DataFrame - Output:DataFrame
- **Estimators** – ML algorithm used to build a predictive model. Input: DataFrame - Output: Model.
- **Parameters** – Configurations for Transformers and Estimators
- **Pipeline** – Chains Transformers and Estimators



Spark ML on HDInsight

HITACHI
Inspire the Next

Spark ML Functionality

Transformers

Text Feature Extraction

- TF-IDF (HashingTF and IDF)
- Word2Vec
- CountVectorizer
- Tokenizer
- StopWordsRemover
- n-gram

Feature Selection

- VectorSlicer
- RFormula
- ChiSqSelector

Dimensionality Reduction

- PCA

Features Vector Preparation

- VectorAssembler
- VectorIndexer
- StringIndexer
- IndexToString

Feature Type Conversion

- Binarizer
- Discrete Cosine Transform (DCT)
- OneHotEncoder
- Bucketizer
- QuantileDiscretizer

Feature Scaling

- Normalizer
- StandardScaler
- MinMaxScaler

Feature Construction

- SQLTransformer
- ElementwiseProduct
- PolynomialExpansion

Estimators (supervised)

Classification

- Decision Trees – Ensembles
- Naïve-Bayes
- SVM

Regression

- Linear Regression
- SVM

Other (Unsupervised)

- Clustering
- Collaborative Filtering
- Frequent Pattern Mining

Spark ML on HDInsight

HITACHI
Inspire the Next

Spark ML - Example

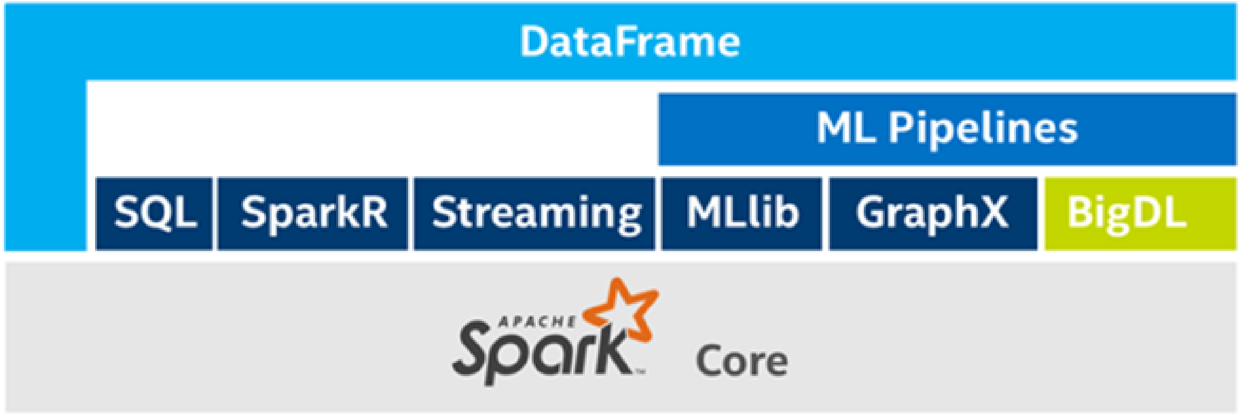
```python
from pyspark import SparkContext
from pyspark.sql import SQLContext
from pyspark.sql.types import *
from pyspark.ml import Pipeline
from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.ml.feature import StringIndexer, VectorIndexer
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

# Set up spark and sql contexts
sparkContext = SparkContext('spark://headnodehost:7077', 'pyspark')
sqlContext = SQLContext(sparkContext)

data_path = "adls:///ml_data/sample_data.txt"

# Load the data stored in adls as RDD.
data_text = sc.textFile(data_path)

# parse data
data_parsed = data_text.map(lambda line: line.split('|')).filter(lambda row: row[0] =! 'NULL')

data_frame = sqlContext.CreateDataFrame(data_parsed,[])

# Index labels, adding metadata to the label column.
labelIndexer = StringIndexer(inputCol="label", outputCol="indexedLabel").fit(data_frame)

# Automatically identify categorical features, and index them.
featureIndexer = VectorIndexer(inputCol="features", outputCol="indexedFeatures", maxCategories=4).fit(data_frame)

# Split the data into training and test sets (30% held out for testing)
(trainingData, testData) = data_frame.randomSplit([0.7, 0.3])

# Train a DecisionTree model.
dt = DecisionTreeClassifier(labelCol="indexedLabel", featuresCol="indexedFeatures")

# Chain indexers and tree in a Pipeline
pipeline = Pipeline(stages=[labelIndexer, featureIndexer, dt])

# Train model.  This also runs the indexers.
model = pipeline.fit(trainingData)

# Summary only.
treeModel = model.stages[2]
print(treeModel)
```

```python
model_path =  "adls:///ml_models/sample_model.mlm"

# Save Model.
model.save(model_path)

# Load Model.
model = DecisionTreeClassifier.load(model_path)

# Make predictions.
predictions = model.transform(testData)

# Select (prediction, true label) and compute test error.
evaluator = MulticlassClassificationEvaluator(labelCol="indexedLabel", predictionCol="prediction", metricName="accuracy")
accuracy = evaluator.evaluate(predictions)

# Print Accuracy
print("Test Error = %g " % (1.0 - accuracy))
```

Spark ML on HDInsight

BigDL – Intel's Distributed Deep Learning Library

https://azure.microsoft.com/en-us/blog/use-bigdl-on-hdinsight-spark-for-distributed-deep-learning/



Concluding Remarks

HITACHI
Inspire the Next

Interactive Data Science Studio

- Azure ML

Extensibility

- Spark on HDI
- Azure ML
- Microsoft R Server

Built-in Features

- Azure ML
- Spark on HDI

Rich Model Interpretability

- SSAS Data Mining
- Microsoft R Server

Pre-built Intelligence

- Azure Cognitive Services
- Azure Data Lake Analytics

ML Pipelining

- Spark on HDI
- Azure Data Lake Analytics
- SQL Server R Services
- Data Mining SSAS

Integration with Operational Apps

- Azure ML
- Azure Cognitive Services
- Microsoft R Operationalization

Scalability (Big Data)

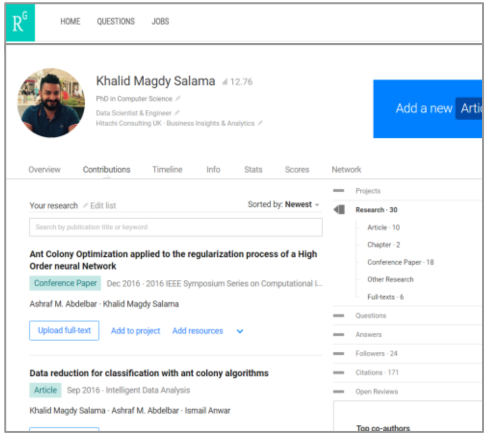
- Microsoft R Server
- Spark on HDI

My Background

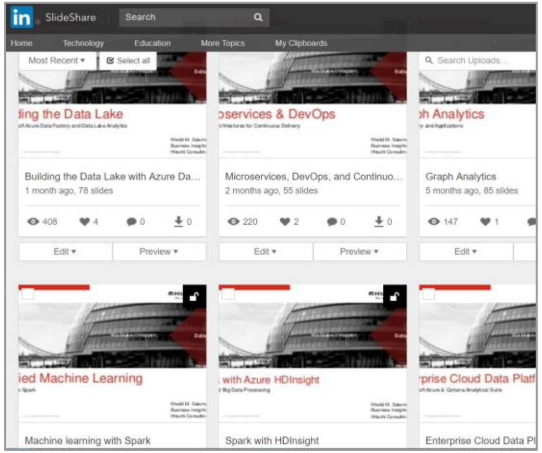
HITACHI
Inspire the Next

Applying Computational Intelligence in Data Mining

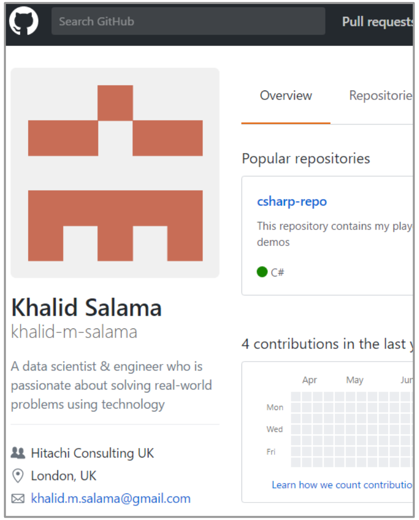
- Honorary Research Fellow, School of Computing , University of Kent.
- Ph.D. Computer Science, University of Kent, Canterbury, UK.
- 28+ published journal and conference papers in the fields of AI and ML



https://www.researchgate.net/profile/Khalid_Salama



https://www.linkedin.com/in/khalid-salama-24403144/



https://github.com/khalid-m-salama/sqlbits-2017

HITACHI
Inspire the Next