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
Introduction to Data Science and Machine Learning
Data Science and Machine Learning

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..
The objective of data science is to provide you with actionable insights to support decision making....
Data Science and Machine Learning

How?

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

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

Cluster Analysis
Discover natural groupings within the data points

Association Rule Discovery
Extract frequent patterns present in the data

Time Series Analysis
Analysis of temporal data to forecast future values

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

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

Collaborative Filtering
Filtering of information using techniques involving collaboration viewpoints
From Experimental Data Science to Operational Machine Learning
Data Science Activities

Experimentation vs. Operationalization

Data Analysis & Experimentation

- Interactive
- Easy to perform
- Rich Visualizations

Exploratory Data Analysis

- Collect Data
- Blend
- Prepare
- Learning Dataset

ML Experiment

- Algorithm Selection
- Parameter Tuning
- Training & Testing
- Model

Report of Visuals & Findings

Decision!
Data Science Activities

Experimentation vs. Operationalization

Operational ML Pipelines
- Pipelined (ETL Integration)
- Scalable
- Apps Integration
Microsoft Advanced Analytics Technologies
Microsoft Advanced Analytics

Cortana Intelligence Suite

https://gallery.cortanaintelligence.com/
Microsoft Advanced Analytics

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
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
Real-time Predictions

Azure Machine Learning

App

Event Hub

Stream Analytics

Power BI

Send data points

Consume messages

Send Results (Input, Output)

Send Input

Receive Output
Azure Machine Learning

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
- Statistical Functions
  - Apply Math Operation
  - Compute Elementary Statistic...
  - 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 Selection...
  - Fisher Linear Discriminant A...
  - Permutation Feature Import...
  - SMOTE
- Text Analytics
  - Feature Hashing
    - Named Entity Recognition
    - Score Vowpal Wabbit Versio...
    - Score Vowpal Wabbit Versio...
    - Score Vowpal Wabbit Versio...
    - Score 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 Recon...
    - Train Model
    - Tune Model Hyperpar...
Algorithms Cheat Sheet

ANOMALY DETECTION
- One-class SVM: >100 features, aggressive boundary
- PCA-based anomaly detection: Fast training

CLUSTERING
- K-means: Discovering structure

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
- One-vs-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
- Two-class decision forest: Accuracy, fast training
- Two-class boosted decision tree: Accuracy, fast training, large memory footprint
- Two-class decision jungle: Accuracy, small memory footprint
- Two-class locally deep SVM: >100 features
- Two-class neural network: Accuracy, long training time
Azure Machine Learning

ML Studio

[Diagram of a workflow process involving data splitting, feature selection, model training, and prediction]

© Hitachi Consulting Corporation 2017. All rights reserved.
Azure Machine Learning

Web Service

Sample Code

```csharp
using System;
using System.Collections.Generic;
using System.IO;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;

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

Web service consumption options

- **Primary Key**: 1nuKjLz2o5fP-8P6uwvG5hrcnpf13Mep6olokQ9WtIyDA6Byu35P8u0dgbk9Ln6C2UKXaAmRyj//
- **Secondary Key**: 2PTfN4KpmqgU6B09PFm1zQ9WmllUX-UK6oW3l0m6fR-N07u6zmlrK0Pq8v9f/5sQD3QxhC
- **Request-Response**: https://europewest.services.azure.com/subscriptions/0330851672914445/b3b23b82b827a6373/services/75183f6e-76cc-440c-ae5f-c1d202a9e75/services/44072f75-6f83-45d7-83d6-75f76444af4f/0/1.0/invocation
- **Batch Requests**: https://europewest.services.azure.com/subscriptions/0330851672914445/b3b23b82b827a6373/services/75183f6e-76cc-440c-ae5f-c1d202a9e75/services/44072f75-6f83-45d7-83d6-75f76444af4f/0/1.0/invocation

Input 9
Output 1

Test Request Response

Scoring Labels: 11.7567170744107

© Hitachi Consulting Corporation 2017. All rights reserved.
Stream Analytics Integration
library(AzureML)
ws <- workspace(id = "1f2b56bcf5fe4f3f9c32ee437f0bfa6995ba1d620f198d07",
auth = "VsvOdAo+nOBzPxphao2Z6fddY57T6QHpdy4M4P7W",
api_endpoint = "https://europeweststudio.azureml.net",
management_endpoint = "https://europewest.azuremange")

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)
}

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

### Comparison

<table>
<thead>
<tr>
<th></th>
<th>CRAN</th>
<th>MRO</th>
<th>MRS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data size</strong></td>
<td>In-memory</td>
<td>In-memory</td>
<td>In-memory &amp; disk</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>Single threaded</td>
<td>Multi-threaded</td>
<td>Multi-threaded, parallel processing 1:N servers</td>
</tr>
<tr>
<td><strong>Support</strong></td>
<td>Community</td>
<td>Community</td>
<td>Community + Commercial</td>
</tr>
<tr>
<td><strong>Functionality</strong></td>
<td>7500+ innovative analytic packages</td>
<td>7500+ innovative analytic packages</td>
<td>7500+ innovative packages + commercial parallel high-speed functions</td>
</tr>
<tr>
<td><strong>Licence</strong></td>
<td>Open Source</td>
<td>Open Source</td>
<td>Commercial license.</td>
</tr>
</tbody>
</table>
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

Microsoft R Server – ScaleR Example

Check Environment

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

Load XDF

```r
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

```r
xDataStep(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)
}

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

Build Predictive Model

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

Perform Prediction

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

Microsoft R Server – ScaleR Functionality

Data Preparation
- Data import – Delimited, Fixed, SAS, SPSS, ODBC
- 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.
- Used Microsoft ScaleR libraries

**Limitations**

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

---

**Process Data** → **Train R Model** → **Serialize Store Models** → **Maintain Models**

**Data Sources** → **ETL Using SSIS**

**EXECUTE sp_execute_external_script**

**Prediction Pipeline**

**Process Data** → **Load Model** → **Perform Prediction** → **Store Results**

**Training Pipeline**
**T-SQL Script**

### Build and Save Model

```sql
DECLARE @model varbinary(max);
EXEC sp_execute_external_script
    @language = 'N''R''',
    @script = N'
    -- Begin Learn Model Script
    model <- lm(Output ~ Input, data = inputData);
    print(summary(model))
    modelbin <- serialize(model, NULL);
    -- End Learn Model Script
    SELECT
        Input, Output
    FROM
        demo.Data;
    , @input_data_1 = N'

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

### Model Summary

<table>
<thead>
<tr>
<th>Results</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>STDOUT message(s) from external script:</td>
<td></td>
</tr>
<tr>
<td>Call:</td>
<td>lm(formula = Output ~ Input, data = inputData)</td>
</tr>
<tr>
<td>Residuals:</td>
<td>Min 1Q Median 3Q Max</td>
</tr>
<tr>
<td>-56.190 -13.946 -1.876 10.000 62.790</td>
<td></td>
</tr>
<tr>
<td>Coefficients:</td>
<td>Estimate Std. Error t value Pr(&gt;</td>
</tr>
<tr>
<td>(Intercept)</td>
<td>-1.0582 5.8035 -0.187 0.852</td>
</tr>
<tr>
<td>Input</td>
<td>1.9462 0.1158 17.794 2.2e-14 ***</td>
</tr>
<tr>
<td>Signif. codes:</td>
<td>0 *** 0.001 *** 0.01 ** 0.05 * 0.1 1</td>
</tr>
<tr>
<td>Residual standard error: 20.02 on 45 degrees of freedom</td>
<td></td>
</tr>
<tr>
<td>Multiple R-squared: 0.8594, Adjusted R-squared: 0.8564</td>
<td></td>
</tr>
<tr>
<td>F-statistic 31.6 on 1 and 48 DF, p-value: &lt; 2.2e-16</td>
<td></td>
</tr>
</tbody>
</table>

### Prediction

```sql
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''',
    @script = N'
    -- Begin Predict
    model <- unserialize(as.raw(model));
    output <- predict(mod, InputDataSet);
    InputDataSet$Output = output
data_output = InputDataSet
    print(data_output)
    -- End Predict
    , @input_data_1 = N'
    , @params = N'

SELECT @input_data_1 = NSELECT @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));
```
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

Azure SQL DW/DB → Build Model → SQL Server Analysis Services

DMX Query → Result

Batch Scoring → Explore/Interpret Model

Build Model → Retrain Model

Online Apps
SQL Server Analysis Services

Overview

- Decision Trees
- 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

Key Influencers Report for "Purchased Bike"
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

Setup a Cognitive Services API

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

More about Microsoft Cognitive Services
Documentation
Pricing
Supplemental Terms of Use for Microsoft Azure
Previews

Put intelligence APIs to work

Microsoft Cognitive Services let you build apps with powerful algorithms using just a few lines of code. Choose from devices and platforms such as iOS, Android, and Windows, keep improving, and add...
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

First-time Installation

<table>
<thead>
<tr>
<th>STATUS</th>
<th>JOB NAME</th>
<th>AUS</th>
<th>LANGUAGE</th>
<th>DURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparing</td>
<td>Install U-SQL Extensions - RegisterAll.usql</td>
<td>1</td>
<td>U-SQL</td>
<td>Just Now</td>
</tr>
</tbody>
</table>
Azure Data Lake Analytics

U-SQL Script

```sql
1 REFERENCE ASSEMBLY [TextCommon];
2 REFERENCE ASSEMBLY [TextSentiment];
3 REFERENCE ASSEMBLY [TextKeyPhrase];
4
5 DECLARE @input_file string = "sqlbits/input-data/book-reviews-sample.csv";
6 DECLARE @output_file string = "sqlbits/output-results/reviews-sentiment.tsv";
7
8 @input_data =
9   EXTRACT Score decimal,
10      Text string
11   FROM @input_file
12   USING Extractors.Csv();
13
14 @sentiment =
15   PROCESS @input_data
16   PRODUCE Score,
17      Text,
18      Sentiment string,
19      Conf double
20   READEONLY Score,
21      Text
22   USING new Cognition.Text.SentimentAnalyzer(true);
23
24 OUTPUT @sentiment
```
Azure Data Lake Analytics

Execution & Output

dla-reviews-sentiment

Job Summary

Preparing Queued Running Finalizing

Progress 0/0

Job Details

Resubmit Refresh Duplicate Script Cancel Job

- 28s 18s 2min 5s
- Succeeded
- 2min 51s
- the_flame_head@hotmail.com
- 3/23/2017, 2:34:13 PM

Show more...

Input Output

NAME

- book-reviews-sample.csv 2.12 MB

Output

- it is very well written. It covers a topic th... Negative -0.5889580594049439...
- vorites; that's not something I can say ab... Positive 0.611548231721756
- result in our stars is that our sun contains l... Negative -0.533393141312441...
- it defies its genre in all the best ways pos... Positive 0.560975730692288
- anger among the Nerdfighter community... Negative -0.5570531354396917...
- stochastic things that you will ever do... Negative -0.5218295244893119
- begin with the author’s assurance of bei... Positive 0.53763207041451855
- as one. I wanted very much to like it and f... Negative -0.537812193725128...
- for several months. I was vaguely aware... Negative -0.589140697376104...
- her and YA lit lover, so I was expecting gr... Positive 0.5158341165808971
- 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
- nention 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
- Suitable for Big Data - **Batch Model Training and Scoring**
- Scheduled & Automated Using Azure Data Factory

Limitations

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

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

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.feature import StringIndexer, VectorIndexer
from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

# Set up spark and sql contexts
sparkContext = SparkContext('spark://headnodehost:27017', 'pyspark')
sqlContext = SQLContext(sparkContext)

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

# Load the data stored in edis 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[1]
print(treeModel)
```

```
# 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)" % accuracy)
```
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

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
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