

Developing and deploying AI solutions on the cloud using Team Data Science Process (TDSP) and Azure Machine Learning (AML)

Debraj GuhaThakurta
Microsoft, AI & Research
Cloud AI Platform
Algorithms and Data Science Group

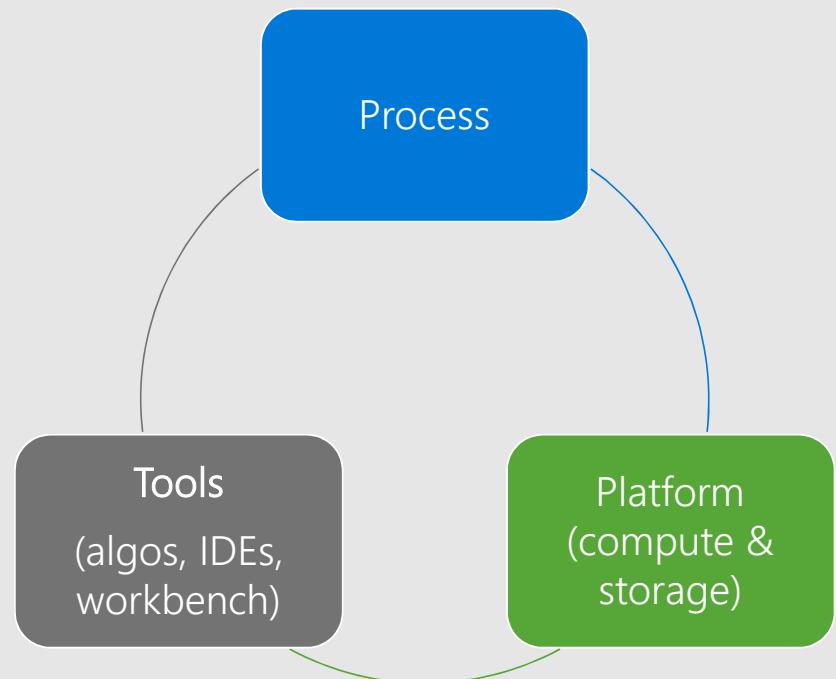
Global Artificial
Intelligence Conference
January 18, 2018, Santa Clara, CA



Deck available @: <https://aka.ms/tdsp-presentations>

Agenda

- Team Data Science Process (TDSP)
 - Principle and objective
 - Key components
 - Adoption and use
- Azure Machine Learning (AML)
 - Key features
 - Developing and deploying AI solutions
- Using TDSP with AML (process + tools)
- Summary

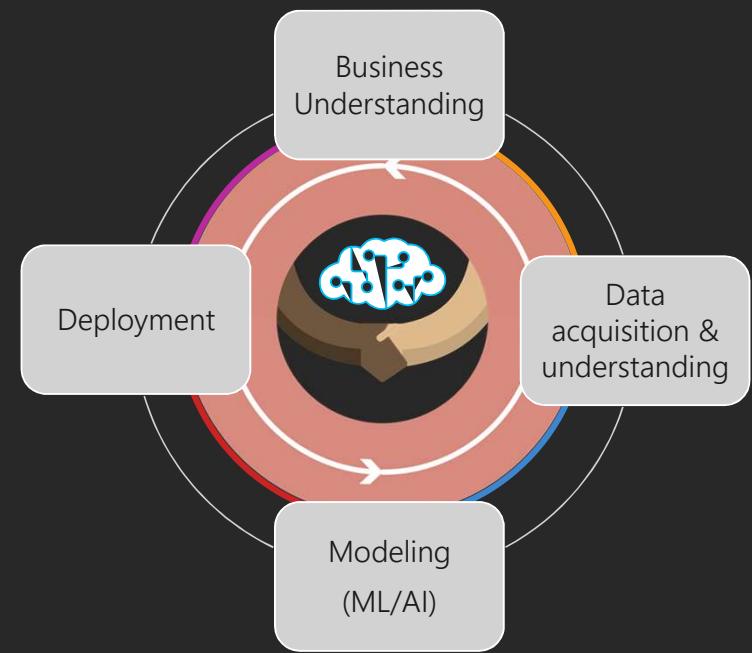


Team Data Science Process (TDSP)

Agile and iterative process to develop,
deploy and manage AI applications in
the cloud

<https://aka.ms/tdsp>

tdsp-feedback@microsoft.com



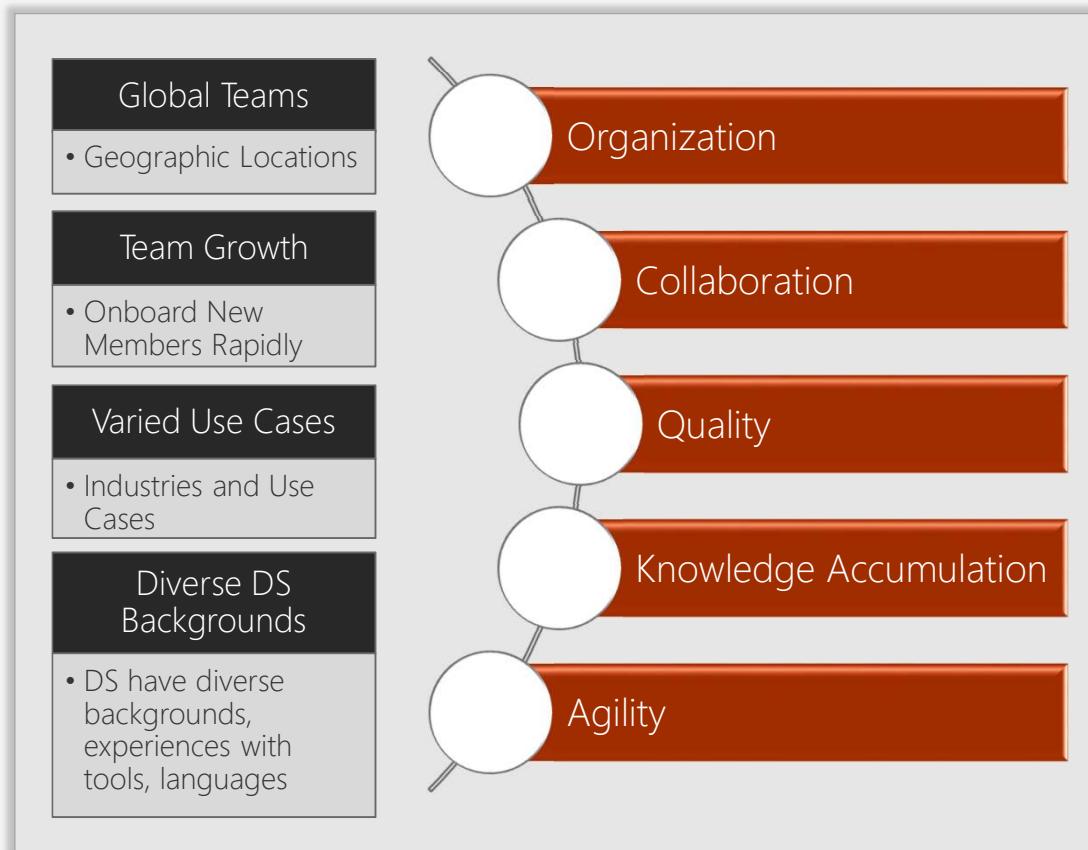
The opportunity and challenge of data science in enterprises

Opportunity: 17% had a well-developed Predictive/Prescriptive Analytics program in place, while 80% planned on implementing such a program within five years – Dataversity 2015 Survey

Challenge: Only 27% of the big data projects are regarded as successful – CapGenimi 2014

Tools & data platforms have matured -
Still a major gap in executing on the potential

One reason: Process challenge in Data Science



"Intelligent" application (ML/AI) development has unique complexity not always encountered in other Software Development scenarios

Why is a process useful for Data Science?

A process is a detailed sequence of activities necessary to perform specific business tasks

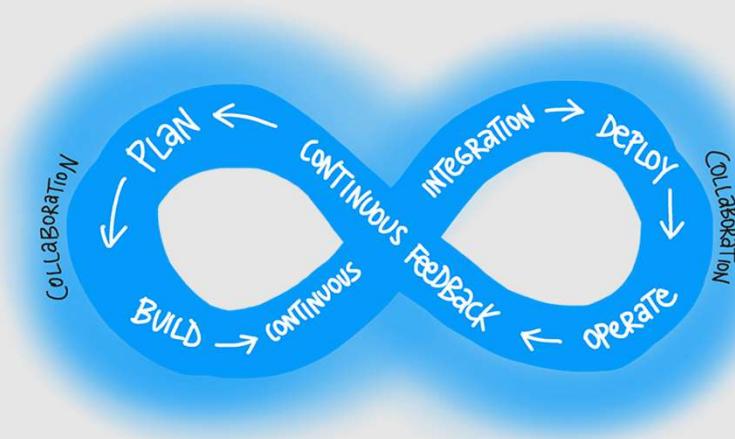
It is used to standardize procedures and establish best practices

Technology and tools are changing rapidly. A standardized process can provide continuity and stability of work-flow.

- Based on discussions with Luis Morinigo, Dir. IoT, NewSignature

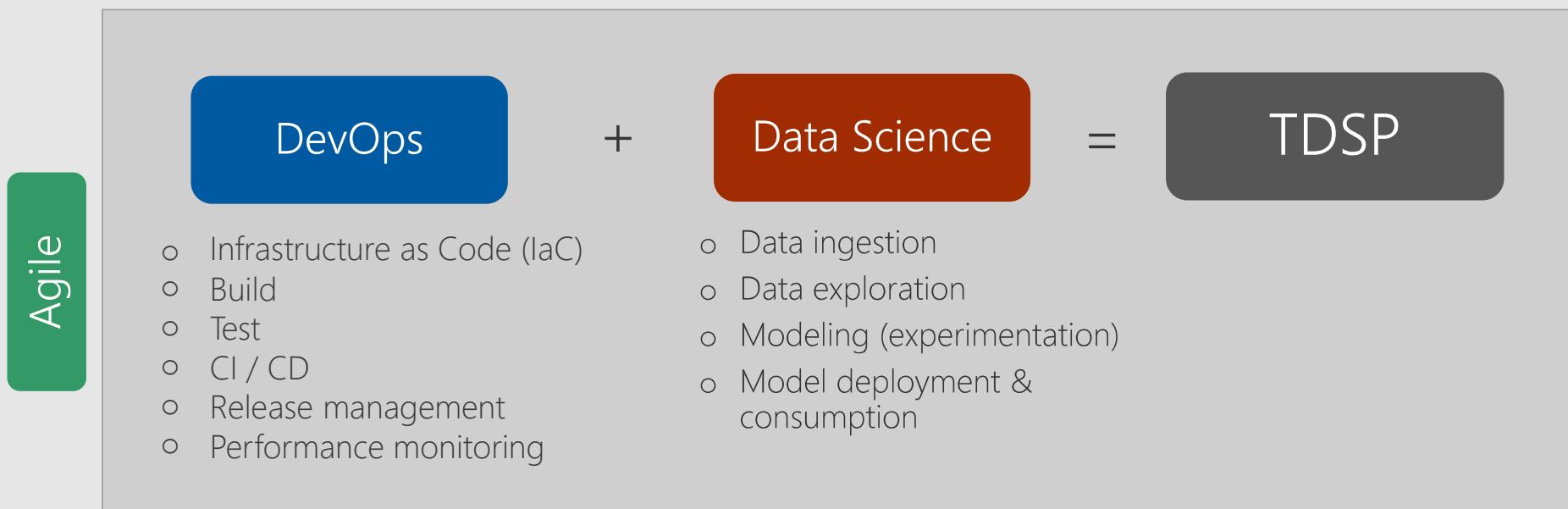
Data Science can borrow processes from DevOps

- Integrated Software Development & Operations (DevOps) has had much more time to mature, standardize, build in efficiency and develop best practices
- Data Science has unique complexity – but can learn standardized processes from DevOps



TDSP objective

Integrate DevOps with **data science workflows** to improve collaboration, quality, robustness and efficiency in data science projects



TDSP features for data science teams

Standardized Data Science Lifecycle

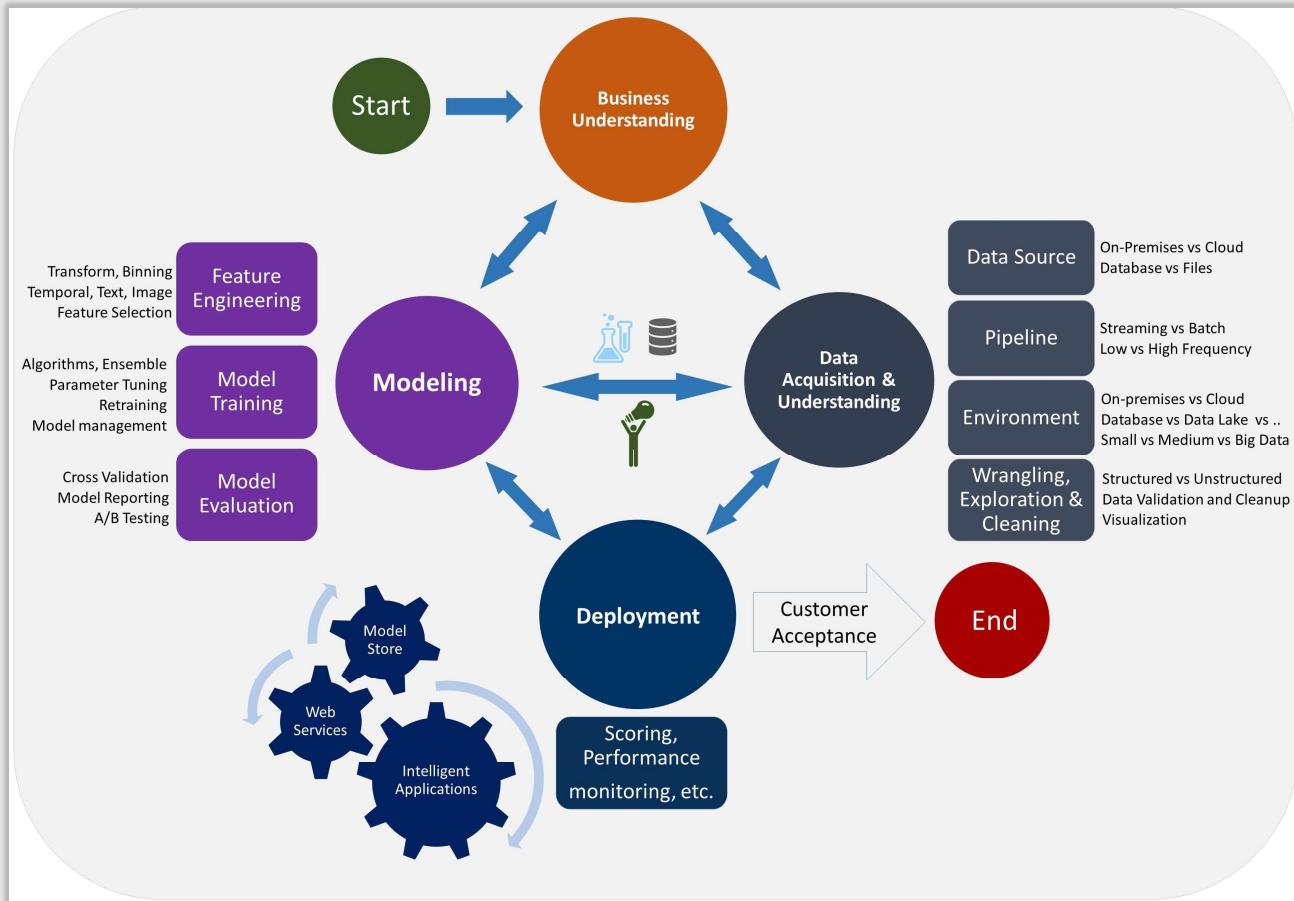
Infrastructure & Toolkits

Project Structure, Templates & Roles

Project Execution (incl. DevOps components)

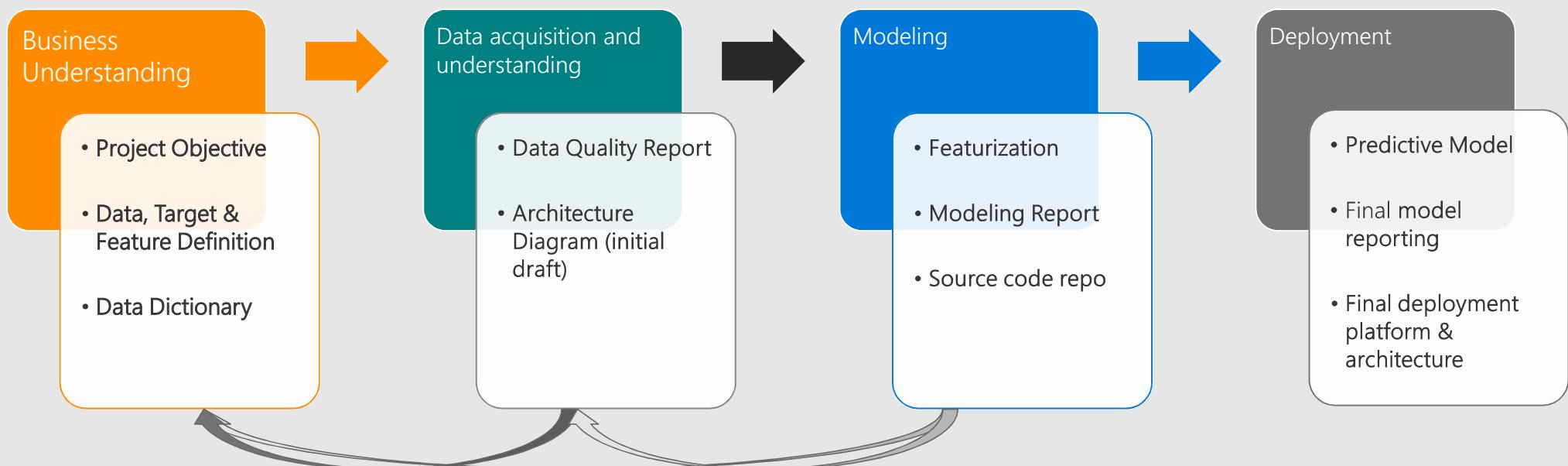
Re-usable Data Science Utilities

Data Science lifecycle



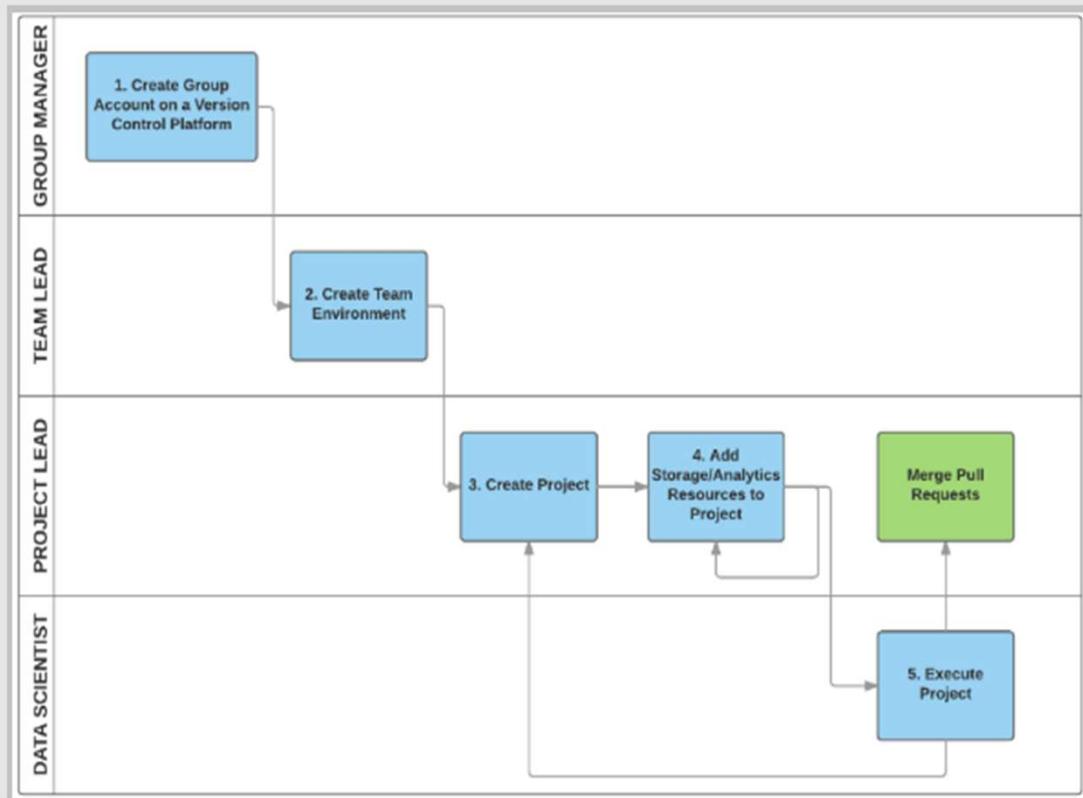
- Primary stages:
 - Business Understanding
 - Data Acquisition and Understanding
 - Modeling
 - Deployment

TDSP lifecycle stages can be integrated with specific deliverables & checkpoints



Project roles & tasks

Personas: account managers, PMs, DSs, SWEs, architects



- Governance and Project Management

- Team lead

- Git template repo & sever management, access control

- Project lead

- Business understanding, create project, work-items

- AI Developers

- Data scientist

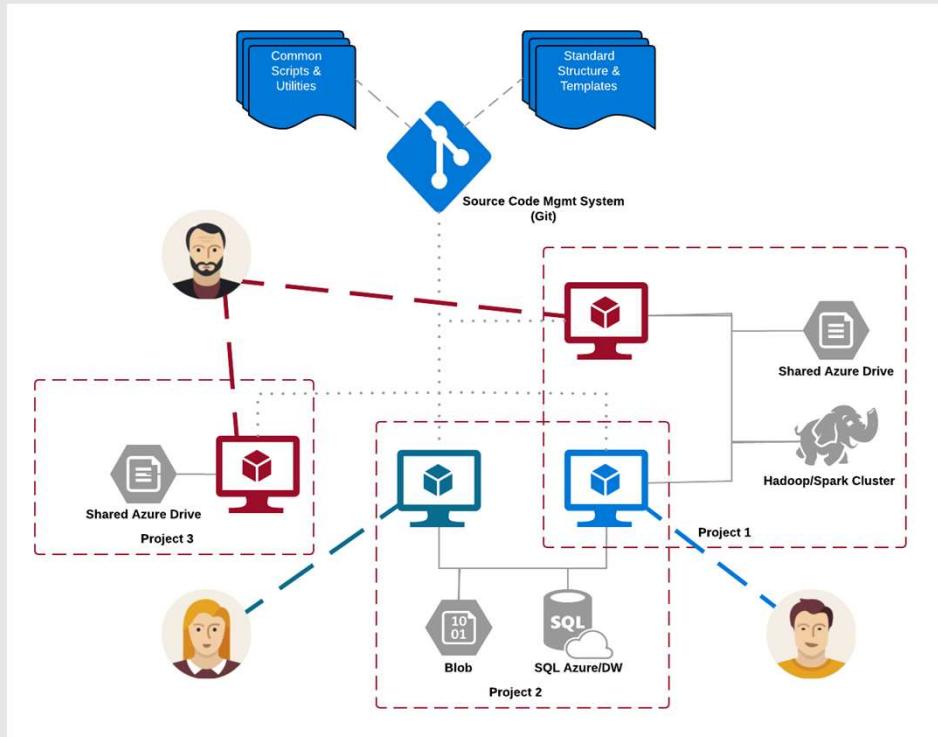
- Modeling, exploratory analysis

- SWE, data engineers, or architect

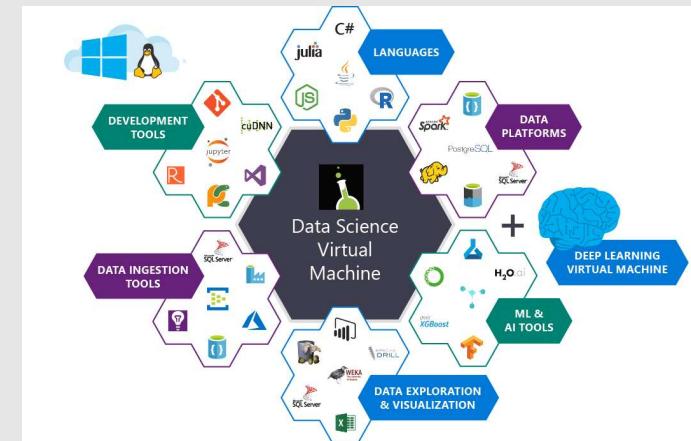
- Data ingestion, deployment, solution architecture, integration into business applications

NOTE: Specific roles depend on organizations

Shared and distributed infrastructure & toolkits

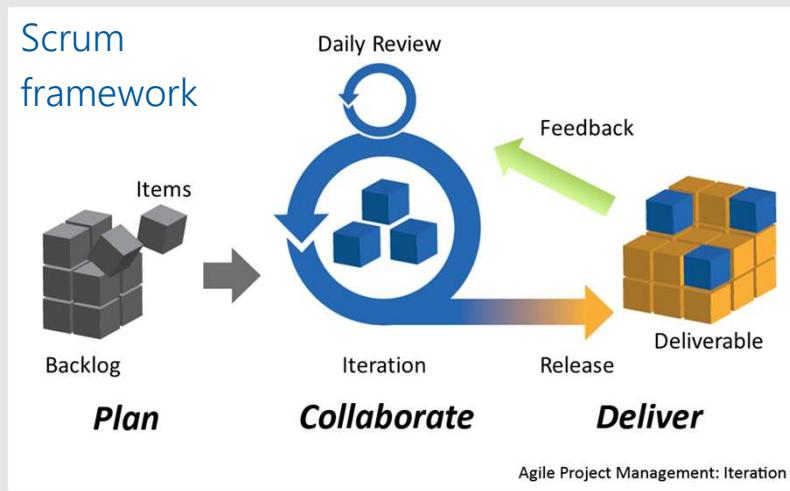


- Project artifacts & code stored in central git (version control) repositories.
- Data typically stored in cloud stores, such as blob or database
- Virtual machines (VMs), or clusters are disposable compute, added to projects as needed
- Many-to-many relationship between data scientists, compute and projects



Agile work planning and execution template

- Use Agile work planning & execution template (data science specific)
 - DS Projects: e.g. "Fraud Detection for Customer ABC"
 - DS Stages: correspond to the stages in TDSP lifecycle.
 - DS Stories: correspond to the life-cycle sub-stages.
 - DS Tasks: Tasks are assignable code or document work items to complete a specific data science story.



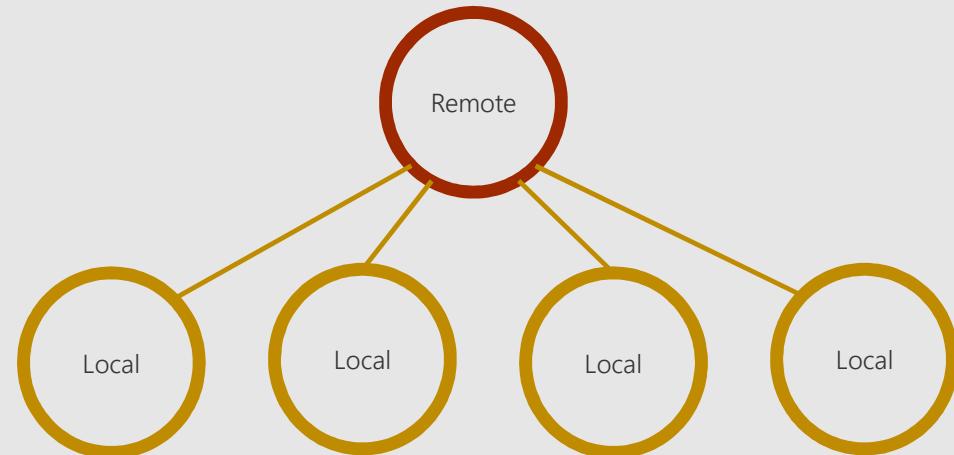
<https://commons.wikimedia.org>

Work Item Type	Title	State
Data Science...	Fraud_Detection_CompanyABC	... New
Business Und...	Define Objectives	Active
Data Science...	identify business variables	Active
Data Science...	Define success metrics	Active
Data Science...	* define accuracy	New
Data Aquisition	Ingest Data	New
Data Aquisition	Explore First Derm data	New
Data Science...	✓ Apply IDEAR to check data quality	New
Data Science...	* Find missing data	New
Data Science...	* Find numerical data distribution	New
Modeling	Feature Engineering	New
Data Science...	✓ Compute various features	New
Data Science...	* Compute categorical features	New
Data Science...	* Compute text features	New
Deployment	Operationalize the model	New
Data Science...	✓ Deploy model as a web services	New
Data Science...	* set up web service with azure cloud services	New

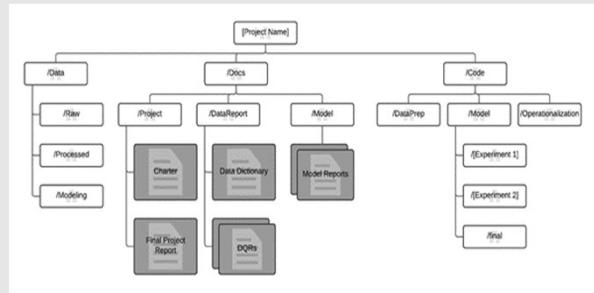


Collaborative development guidelines

- Version control and review
 - Git is a Version Control System
 - Each repo contains the full change history
 - Used in a distributed way with a single remote repo and several local repos (on local machine or a VM)



TDSP git Template



Create a branch

Name

Enter your branch name

Based on

Utilities

master

Work items to link

Search work items by ID or title

261 This is a second task put in iteration 2
Updated 8/30/2016, New

Integrating DevOps in projects

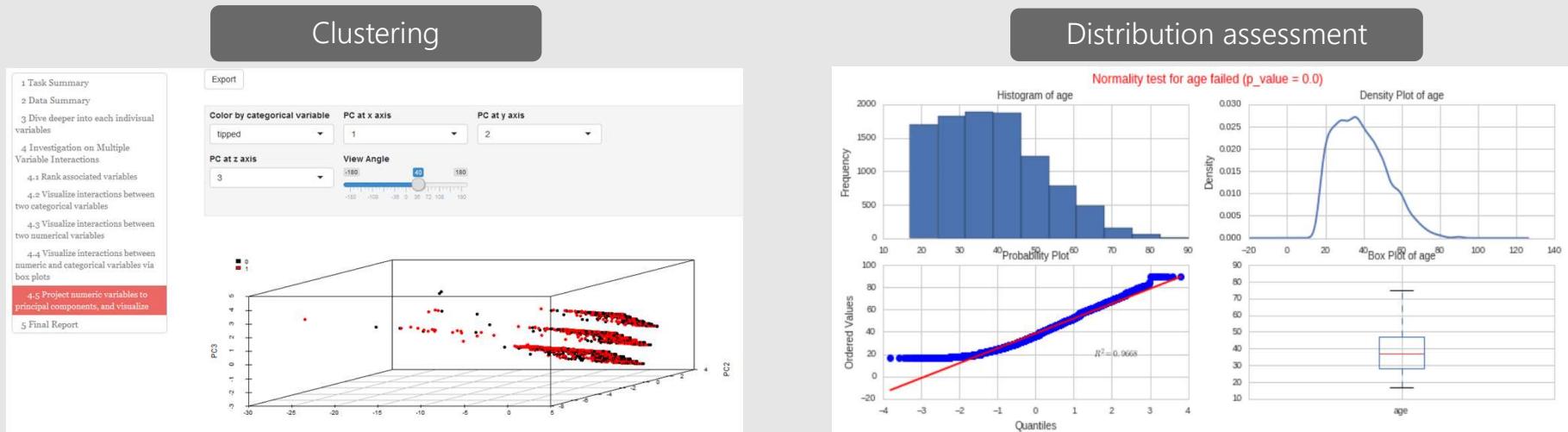
Create new build definition	
Select a template	
Build	Deployment Custom
 Android	Build your Android projects, run tests, sign and align Android App Package files. This template requires the Android SDK to be installed on the build agent.
 Ant	Build your Java projects and run tests with Apache Ant. This template requires Ant to be installed on the build agent.
 Gradle	Build your Java projects and run tests with Gradle using a Gradle wrapper script.
 Maven	Build your Java projects and run tests with Apache Maven. This template requires Maven to be installed on the build agent.
 Universal Windows Platform	Build Universal Windows Platform applications using Visual Studio. This template requires that Visual Studio and the Universal templates are installed.
 Visual Studio	
 Empty	Start with a definition that has no steps.

The screenshot shows the Microsoft Test Lab interface. At the top, there's a ribbon with icons for Deploy, Chef, and Chef Knife. Below the ribbon, there are two sections: 'Chef' and 'Chef Knife'. The 'Chef' section has a 'Deploy to Chef environments by editing environment attributes' link and an 'Add' button. The 'Chef Knife' section has a 'Run Scripts with knife commands on your chef workstation' link and an 'Add' button. Below these sections is a navigation bar with tabs: Fabrikam, Home, Code, Work, Build & Release, and Test. The 'Test' tab is highlighted with a blue background. Under the 'Test' tab, there are several sub-tabs: Test Plans (which is selected and highlighted with an orange border), Parameters, Configurations, Runs, Machines, and Load test. To the right of the sub-tabs, there's a message: 'You can't create test cases without a test plan'. Below this message, there are 'Tests' and 'Charts' buttons. On the left side, there's a sidebar with a 'Select a test plan' dropdown and a list of suite types: 'Test plan' (selected and highlighted with an orange border), 'Static suite', 'Requirement-based suite', 'Query-based suite', and 'Shared steps'. There are also icons for creating new suites and deleting existing ones.

Re-usable data science utilities: Analytics

Interactive data exploration and reporting – IDEAR (Python, R, MRS)

- Data quality assessment
- Getting business insights from the data
- Association between variables
- Generating standardized data quality reports automatically



<https://github.com/Azure/Azure-TDSP-Utilities>

Re-usable data science utilities: Analytics - modeling

Automated modeling and reporting AMAR (R)

o.1 Introduction

- o.2 Specify YAML parameter file for input data and modeling
- o.3 Input data, and splitting data into train/test
- o.4 Model training
- o.5 Model evaluation: Compare model accuracies of different algorithms, and examine variable importance
- o.6 Summary

Automated Model training: Regression

Team Data Science Process by Microsoft
September 23, 2016

0.1 Introduction

This R Markdown performs **exploratory** model training and evaluation for **regression** tasks using the **Caret package**, which has convenient functions for resampling, hyper-parameter sweeping, and model accuracy comparison. The user can use Caret with R machine learning packages (such as, **glmnet**, **RandomForest**, **xgboost**, etc.). We use these three algorithms with limited paraUsers can customize this template to create their own model training and evaluation process for linear regression tasks.

Predicted vs. Actual (multiple algorithms)

o.1 Introduction

o.2 Specify YAML parameter file for input data and modeling

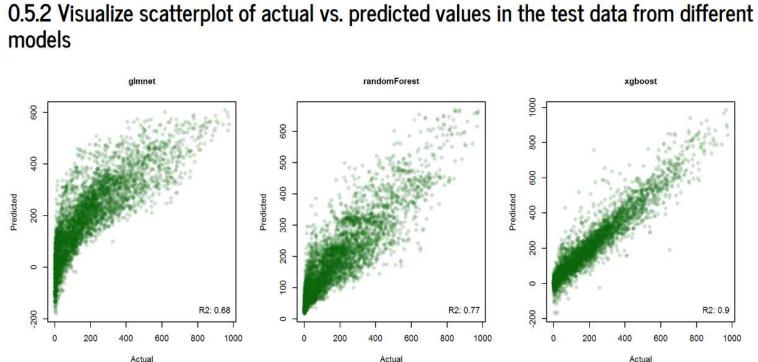
o.3 Input data, and splitting data into train/test

o.4 Model training

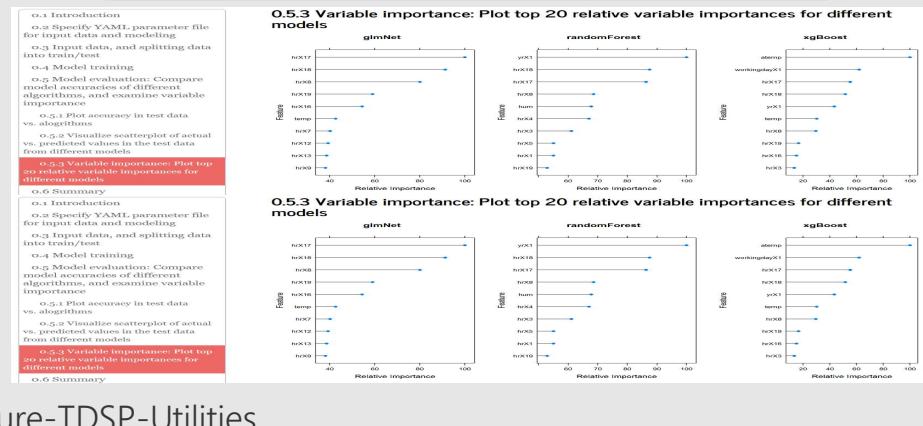
o.5 Model evaluation: Compare model accuracies of different algorithms, and examine variable importance

o.5.1 Plot accuracy in test data vs. algorithms

o.5.2 Visualize scatterplot of actual vs. predicted values in the test data from different models

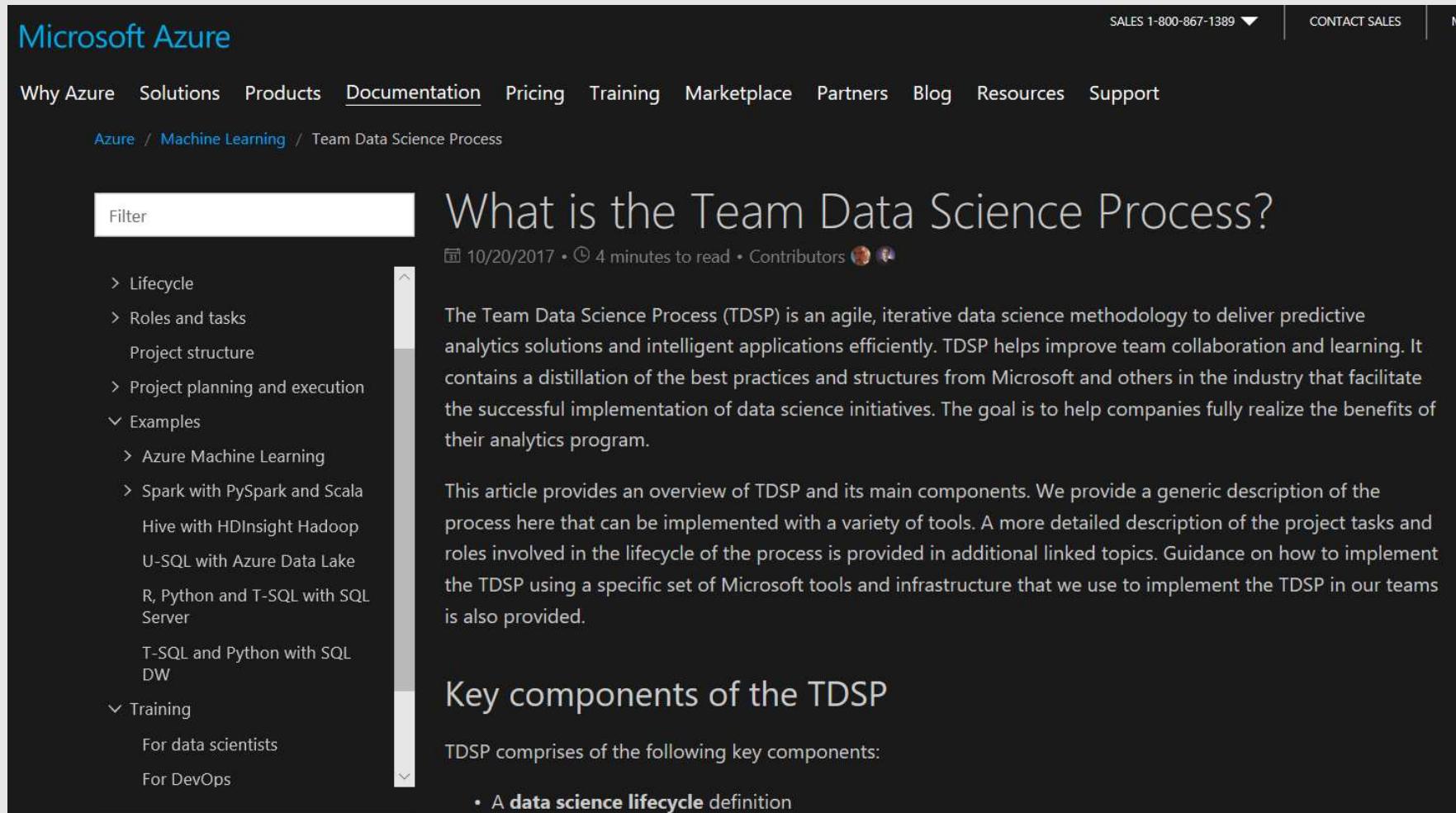


Feature Importance (multiple algorithms)



<https://github.com/Azure/Azure-TDSP-Utilities>

TDSP documentation: <https://aka.ms/tdsp>



The screenshot shows the Microsoft Azure documentation page for the Team Data Science Process (TDSP). The page has a dark background with white text. At the top, there's a navigation bar with links for Why Azure, Solutions, Products, Documentation (which is underlined), Pricing, Training, Marketplace, Partners, Blog, Resources, and Support. Below the navigation bar, the breadcrumb trail shows 'Azure / Machine Learning / Team Data Science Process'. On the left side, there's a sidebar with a 'Filter' input field and a list of topics under 'Lifecycle', 'Examples', and 'Training'. The main content area features a large heading 'What is the Team Data Science Process?' followed by a timestamp ('10/20/2017'), a reading time indicator ('4 minutes to read'), and a 'Contributors' section with two profile icons. The main text describes TDSP as an agile, iterative methodology for delivering predictive analytics solutions. It highlights its purpose of improving team collaboration and learning through best practices and structures. Below this, another text block provides an overview of TDSP components and its implementation. A section titled 'Key components of the TDSP' lists a single bullet point: 'A **data science lifecycle** definition'.

Microsoft Azure

SALES 1-800-867-1389 ▾ CONTACT SALES

Why Azure Solutions Products Documentation Pricing Training Marketplace Partners Blog Resources Support

Azure / Machine Learning / Team Data Science Process

Filter

> Lifecycle

> Roles and tasks

Project structure

> Project planning and execution

Examples

> Azure Machine Learning

> Spark with PySpark and Scala

Hive with HDInsight Hadoop

U-SQL with Azure Data Lake

R, Python and T-SQL with SQL Server

T-SQL and Python with SQL DW

Training

For data scientists

For DevOps

What is the Team Data Science Process?

10/20/2017 • 4 minutes to read • Contributors

The Team Data Science Process (TDSP) is an agile, iterative data science methodology to deliver predictive analytics solutions and intelligent applications efficiently. TDSP helps improve team collaboration and learning. It contains a distillation of the best practices and structures from Microsoft and others in the industry that facilitate the successful implementation of data science initiatives. The goal is to help companies fully realize the benefits of their analytics program.

This article provides an overview of TDSP and its main components. We provide a generic description of the process here that can be implemented with a variety of tools. A more detailed description of the project tasks and roles involved in the lifecycle of the process is provided in additional linked topics. Guidance on how to implement the TDSP using a specific set of Microsoft tools and infrastructure that we use to implement the TDSP in our teams is also provided.

Key components of the TDSP

TDSP comprises of the following key components:

- A **data science lifecycle** definition

TDSP trainings

[Filter](#)

Team Data Science Process for data scientists

11/21/2017 • 8 minutes to read • Contributors

This article provides guidance to a set of objectives that are typically used to implement comprehensive data science solutions with Azure technologies. You are guided through:

- understanding an analytics workload
- using the Team Data Science Process
- using Azure Machine Learning
- the foundations of data transfer and storage
- providing data source documentation
- using tools for analytics processing

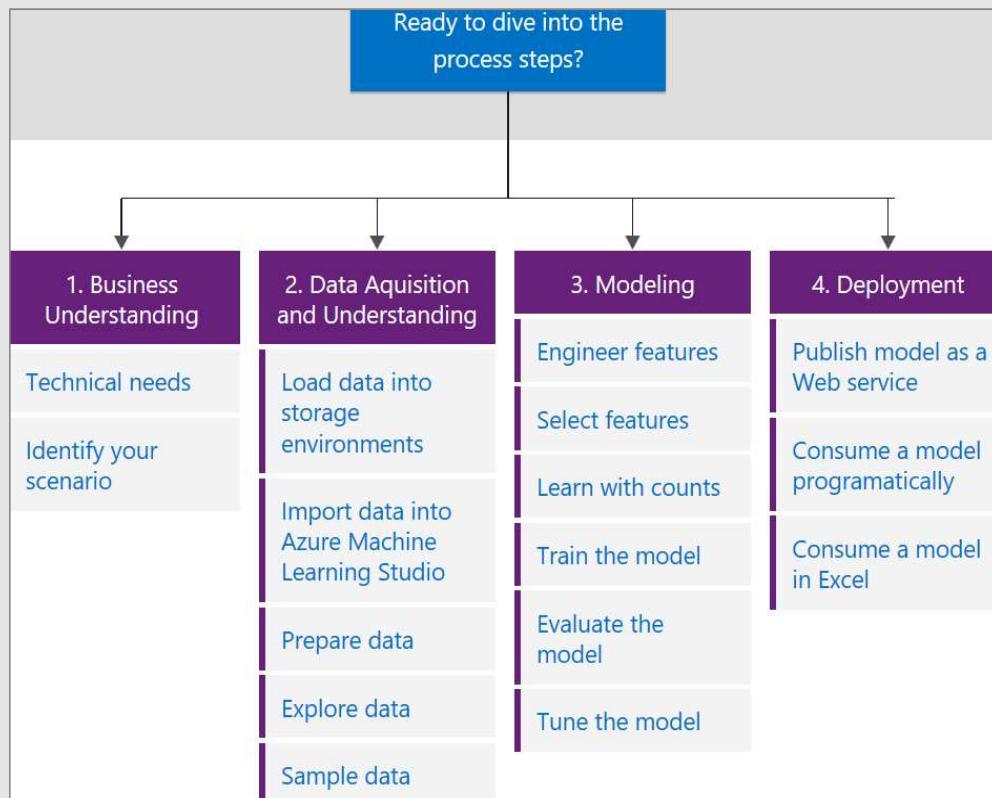
These training materials are related to the Team Data Science Process (TDSP) and Microsoft and open-source software and toolkits, which are helpful for envisioning, executing and delivering data science solutions.

Lesson Path

You can use the items in the following table to guide your own self-study. Read the *Description* column to follow the path, click on the *Topic* links for study references, and check your skills using the *Knowledge Check* column.

Objective	Topic	Description	Knowledge Check
Understand the processes for developing analytic projects	An introduction to the Team Data Science Process	We begin by covering an overview of the Team Data Science Process – the TDSP. This process guides you through each step of an	Review and download the TDSP Project Structure artifacts to your local machine for your project.

TDSP in action: E2E worked-out samples



- Azure Machine Learning
- Azure HDInsight Spark
- Azure HDInsight Hadoop
- SQL-server with R and Python
- Azure SQL data warehouse
- Azure Data Lake

Adoption: How to stage (if needed)

- Data science teams may stage adoption as follows

Level 1

- One git repository per project
- Standard directory structure
- Standardized templates like charter, exit reports
- Planning and tracking of work items

Level 2

- Customize templates to fit team needs
- Create shared team utility repo (like IDEAR, AMAR)

Level 3

- Develop process to graduate code from projects to the shared team utility repo
- Develop E2E worked-out templates
- Use mature work planning and tracking system (e.g. Agile)

Level 4

- Link git branch with work items
- Code review
- Manage and version model and data assets
- Develop automated testing framework
- Develop automated CI/CD

Adoption: Customers

- Microsoft internal
 - Microsoft consulting services (MCS)
 - AI & R Cloud Platform: Algorithm and data sciences team
 - Windows Devices DS team
 - Technology Evangelism and Delivery DS team
- External partners
 - *New Signature*
 - *BlueGranite*

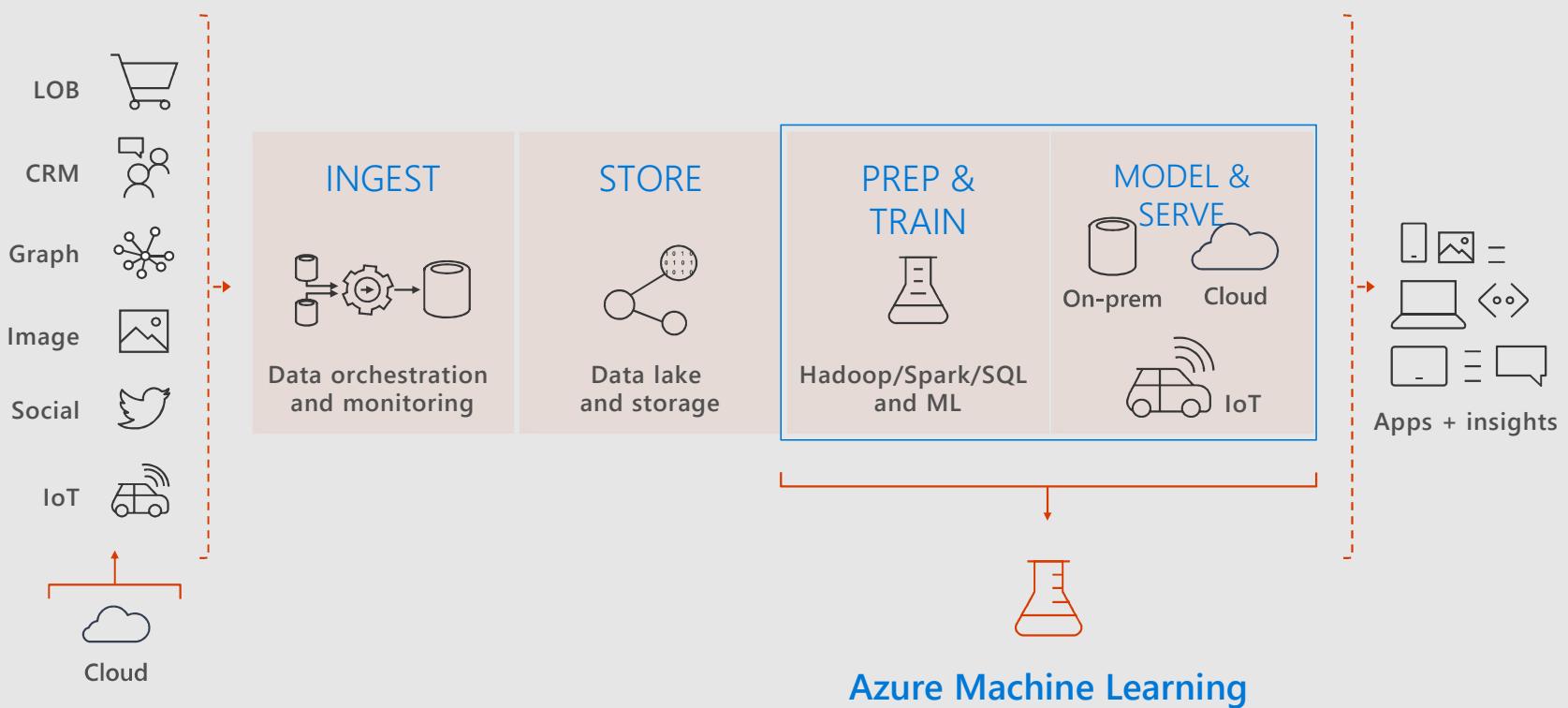


Azure Machine Learning (AML)

<https://docs.microsoft.com/azure/machine-learning/preview>

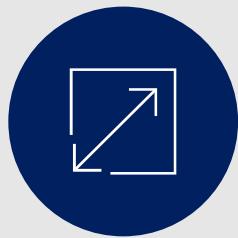


Azure Machine Learning

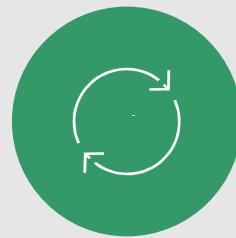


Bring AI everywhere

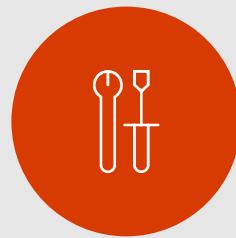
Benefit from the fastest AI developer cloud



Build, deploy, and
manage at scale



Boost productivity with
agile development



Build with the tools and
platforms you know

Build, deploy, and manage at scale



- **Build and deploy everywhere** – cloud, on-premises, edge, and in-data
- **Deploy in minutes** with data driven management and retraining of all your models
- **Prototype locally then scale up** and out with VMs, Spark clusters, and GPUs
- Real time, high through-put insights everywhere, including Excel integration





Manage models

Deployment and management of models as HTTP services

Container-based hosting of real time and batch processing

Management and monitoring through Azure (e.g., AppInsights)

First class support for SparkML, Python, CNTK, TF, TLC, R, extensible to support others (Caffe, MXnet)

Service authoring in Python and .NET Core

```
# -n app name
# -f scoring script file name
# -m dependencies, in this case it is the pickled model file
# -r type of model, in this case it is the scikit-learn model
$ az ml service create realtime -n irisapp -f iris_score.py -m model.pkl -r scikit-py

$ az ml service run realtime -n irisapp -d '{"input": "[[1.0, 2.2, 2.3, 2.7]]"}'
2
```



Deploy everywhere



DOCKER

- Single node deployment
(cloud/on-prem)
- Azure Container Service
- Azure IoT Edge
- Spark clusters

Boost productivity with agile development



- Spend more time modeling and less time prepping with built-in intelligent data wrangling
- Increase collaboration and sharing with notebooks and Git
- Avoid losing anything with [version control](#) and [reproducibility](#)
- Know the best performing models with [metrics](#), [lineage](#), [run history](#), [asset management](#), and more



Build with tools and platforms you know



- Choose between visual drag-and-drop or code-first authoring
- Use [your favorite IDEs](#)
- Call Azure Machine Learning services directly in VS Code*
- Build on any framework or library with the most popular languages
- Train quicker and easier with industry-leading Spark and GPUs



*More IDEs coming soon

Use what you want

Use your favorite IDE

Leverage all types of platforms and tools

USE ANY FRAMEWORK OR LIBRARY



USE ANY TOOL



USE THE MOST POPULAR INNOVATIONS



Using TDSP with AML

Using TDSP with Azure Machine Learning

The screenshot shows the Azure Machine Learning Workbench interface. On the left, there's a sidebar with a 'How To' section where 'Use TDSP in Azure ML' is highlighted. The main area displays a 'Create New Project' dialog. The dialog includes fields for 'Project name' (set to 'TDSP_Template'), 'Project directory' (set to 'C:\Users\remoteuser\Desktop\IgniteDemos'), 'Project description (optional)' (set to 'TDSP Template Project'), 'Git repository' (set to '<Your EMPTY Git Repo>'), and 'Workspace' (set to 'IgniteDemos'). Below these fields is a 'Gallery' section with a search bar containing 'TDSP'. A red arrow points from the search bar to the text 'TDSP'. At the bottom right of the dialog is a large blue 'Create' button, which is also highlighted with a red box.

Filter

- Team Data Science Process Documentation
- Overview
- > Lifecycle
- > Roles and tasks
- Project structure
- Project execution
- > Tutorials
- < How To
 - Use TDSP in Azure ML
- > Set up data science environments
- > Analyze business needs
- > Acquire and understand data
- > Develop models
- > Deploy models
- > Related
- > Resources

Azure Machine Learning Workbench

Create New Project

Project name: TDSP_Template

Project directory: C:\Users\remoteuser\Desktop\IgniteDemos

Project description (optional): TDSP Template Project

Git repository: <Your EMPTY Git Repo>

Workspace: IgniteDemos

TDSP

Create Cancel

Comments Edit Share Theme Dark Branch release-ignite-aml-v2

In this article

- What Is Team Data Science Process?
- Why Should You Use TDSP Structure and Templates?
- Things To Note Before Creating A New Project
- Instantiating TDSP Structure and Templates From the Azure Machine Learning Template Gallery
- Examine The TDSP Project Structure
- Using The TDSP Structure and Templates
- Documenting Your Project
- Next Steps

TDSP – AML worked-out samples in AI

- Unstructured data (NLP) modeling with deep-learning
 - Sentiment classification using supervised word-embeddings
 - Biomedical (PubMed) Named entity recognition using LSTM
- Structured data
 - Income classification
- Public Github repositories for each sample

The screenshot shows a GitHub repository page for a project named "aml_config". The top navigation bar indicates 60 commits, 1 branch, 0 releases, 3 contributors, and an MIT license. The repository has a single branch named "master". The commit history lists several commits by user "wguo123":

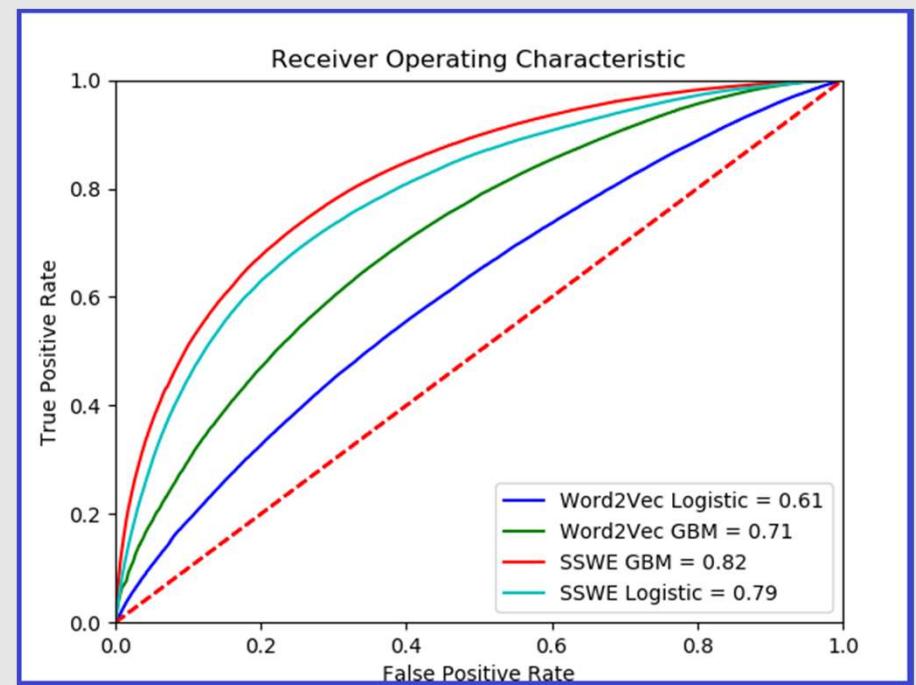
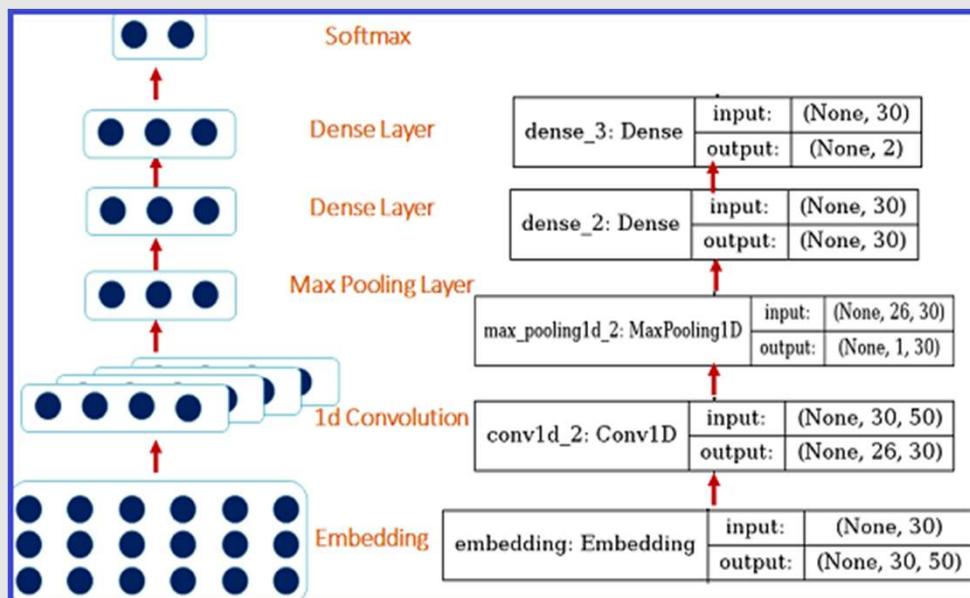
- Update readme (26 days ago)
- revise docs, structures, etc (a month ago)
- modified some wording (a month ago)
- added blog post docx (a month ago)
- Initial commit (.gitignore, LICENSE) (3 months ago)
- Initial commit (README.md) (3 months ago)
- Update readme (26 days ago)

A large text box at the bottom contains the following text:

Use word embeddings to predict Twitter sentiment following Team Data Science Process

<https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/walkthroughs>

Sample: Sentiment-specific word embeddings (SSWE) improves classification of sentiments



<https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/predict-twitter-sentiment>

<http://www.aclweb.org/anthology/P14-1146>

Summary

- A well-defined data science process is critical to bridge the gap between opportunities and challenges, and put AI models in production to impact businesses
- A combination of appropriate process (e.g. TDSP), tools (e.g. AML) and data-platforms is important for efficient development and deployment of AI solutions
 - TDSP – process
 - AML – tool
- A process such as TDSP can be used with a variety of tools and data platforms
- Resources are available to use TDSP and AML together, along with other data platforms on Azure to efficiently build and deploy AI solutions on the cloud

Thank you!

<https://aka.ms/tdsp>
tdsp-feedback@microsoft.com

Deck available @: <https://aka.ms/tdsp-presentations>