

# Developing and deploying NLP services on the cloud using Azure ML and the Team Data Science Process

---

Debraj GuhaThakurta  
Cloud & AI  
Algorithms and Data Science Group

Global Artificial  
Intelligence Conference  
April 28, 2018, Seattle, WA

Credits: Mohamed Abdel-Hady, Wei Guo, Raza Khan, Akshay Mehra, Zoran Dzunic



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

# Global Big Data Conference

GLOBAL ARTIFICIAL INTELLIGENCE  
CONFERENCE

Seattle

APRIL 27<sup>th</sup>, 28<sup>th</sup>, 29<sup>th</sup> 2018



Washington State  
Convention Center  
at convention place



[www.globalbigdataconference.com](http://www.globalbigdataconference.com)

Twitter : @bigdataconf  
#GAIC

# Agenda

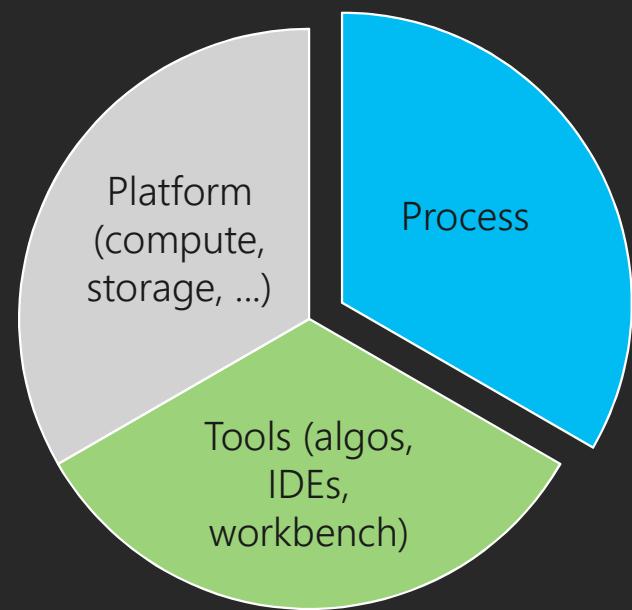
- Team Data Science Process (TDSP)
  - Principle and objective
  - Key components
  - Adoption and use
- Azure Machine Learning (AML)
  - Key features
  - Developing and deploying AI solutions
- Using AML and TDSP to develop & deploy NLP services

# Team Data Science Process (TDSP)

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

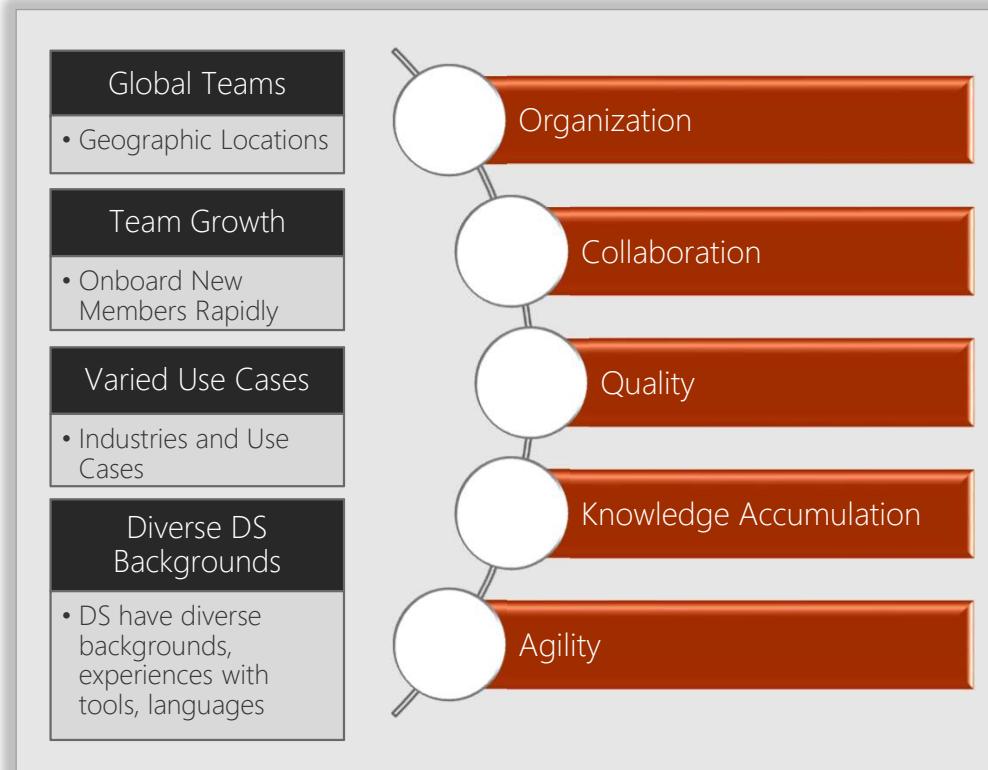
<https://aka.ms/tdsp>

tdsp-feedback@microsoft.com



# Process challenge in Data Science

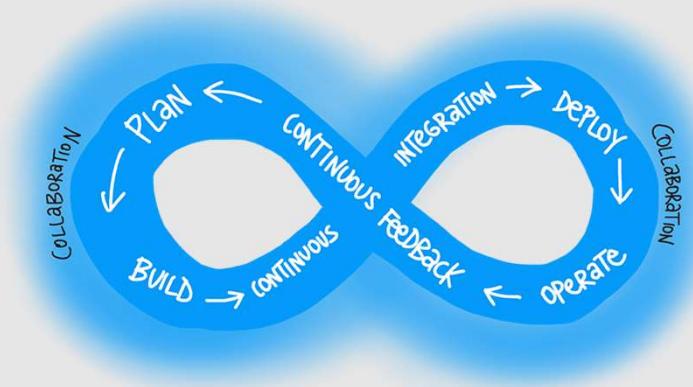
5



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

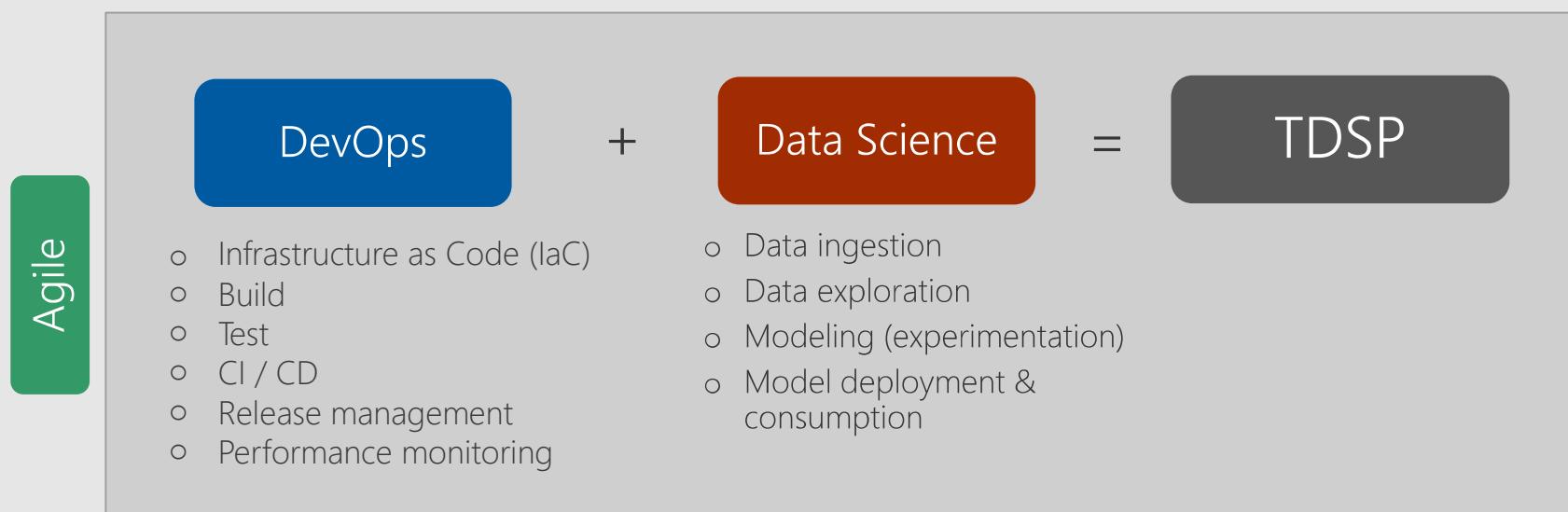
# Data Science can borrow processes from DevOps

- 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

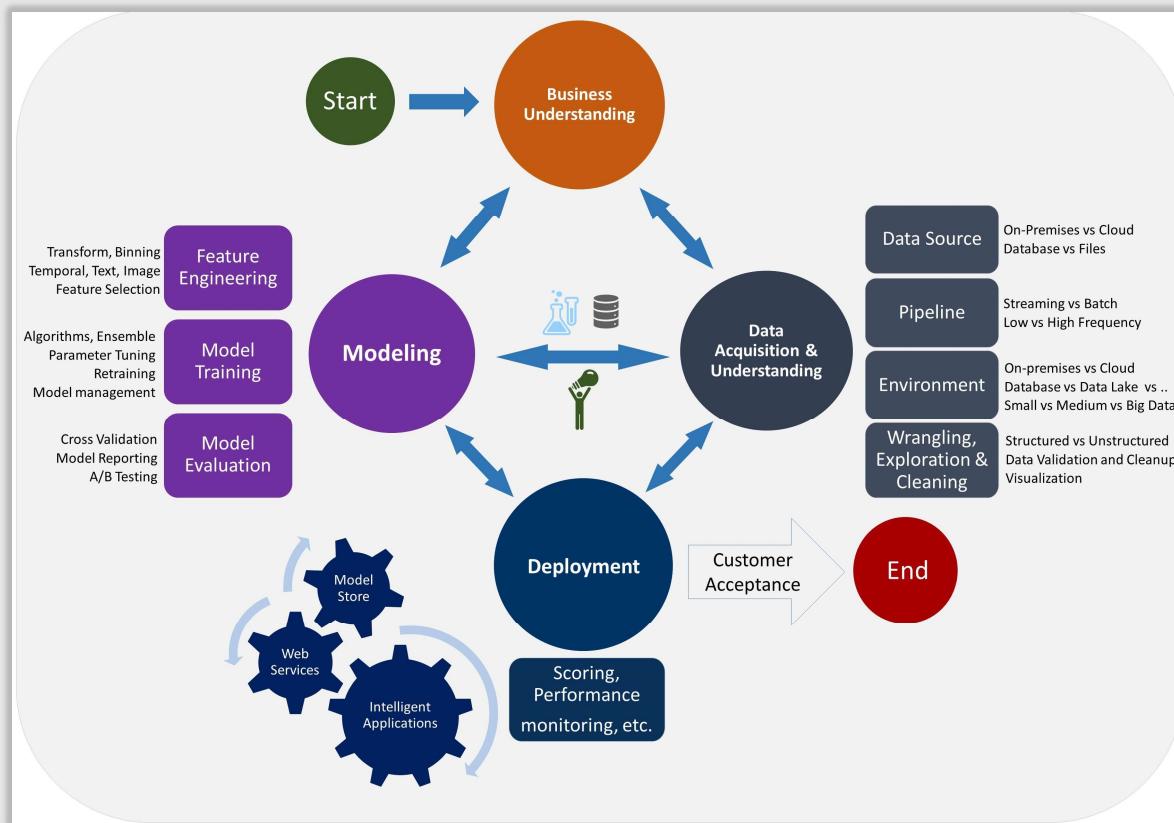


# TDSP objective

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

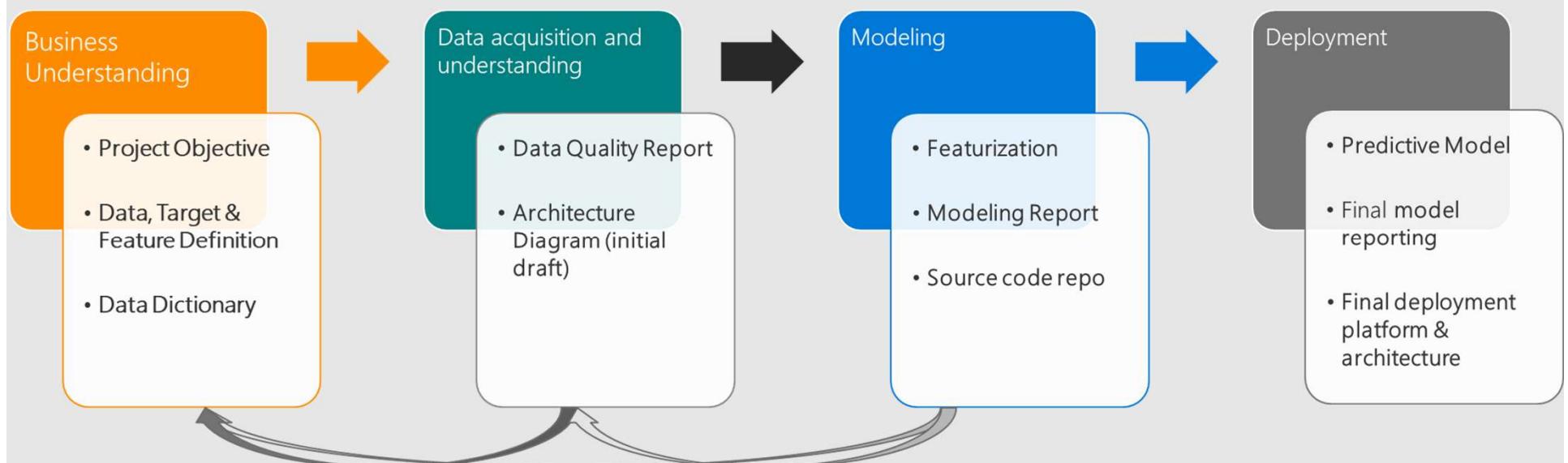


# Data Science lifecycle

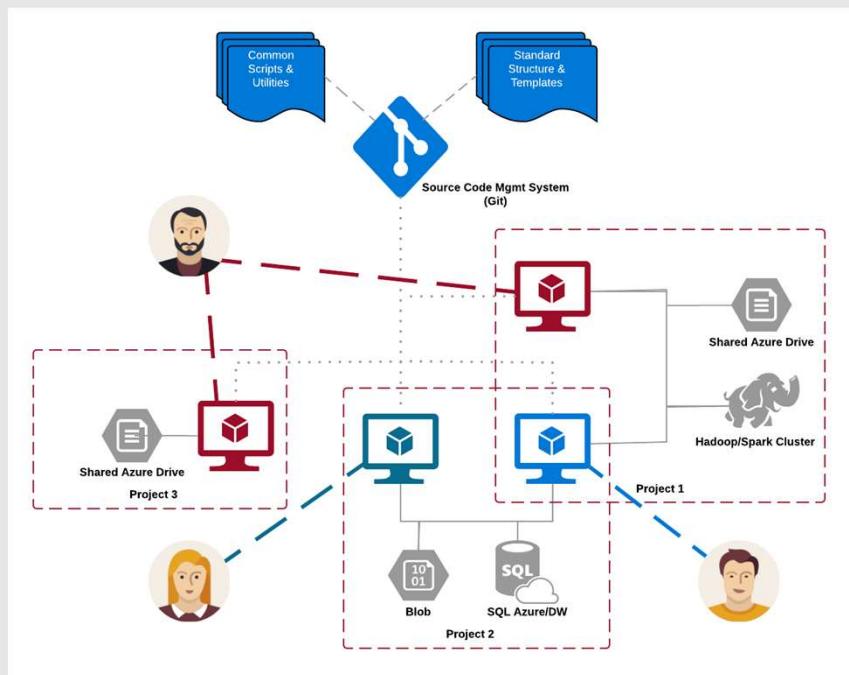


- Primary stages:
  - Business Understanding
  - Data Acquisition and Understanding
  - Modeling
  - Deployment

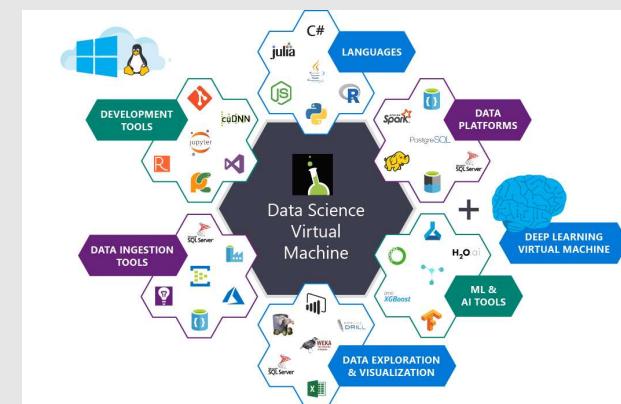
TDSP lifecycle stages can be integrated with specific deliverables & checkpoints – Templates available



# Shared and distributed infrastructure & toolkits

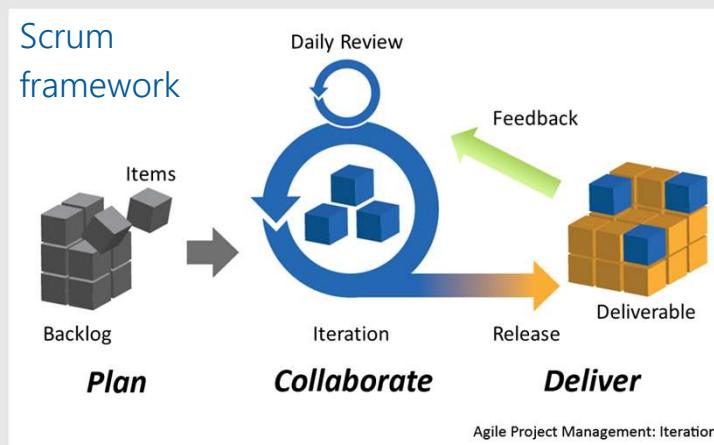


- Project artifacts & code stored in central git (version control) repositories.
- Data typically stored in cloud stores, such as blob or database
- Virtual machines (VMs), or clusters are disposable compute, added to projects as needed
- Many-to-many relationship between data scientists, compute and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



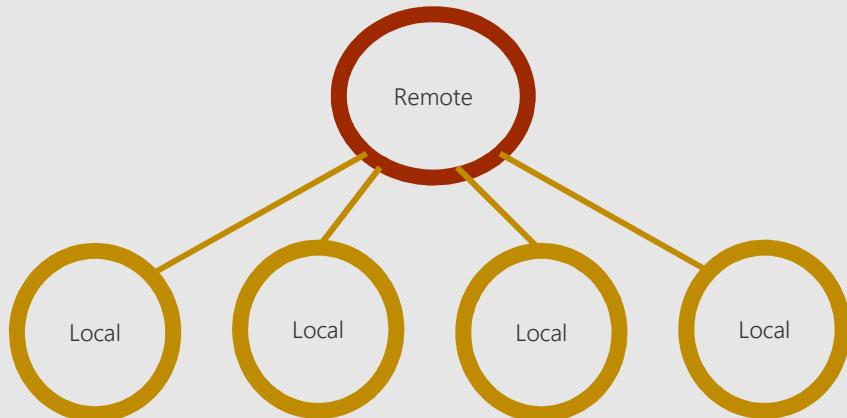
<https://commons.wikimedia.org>

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

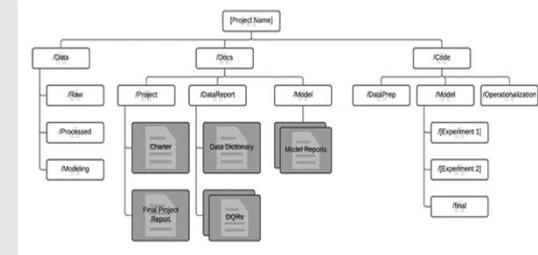


# Collaborative development guidelines

- 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



## Integrated Agile planning & code development

Order	Work Item Type	Title	State
+	1	Data Science... <input checked="" type="checkbox"/> Fraud_Detection_CompanyABC	... ● New
		Business Und... <input checked="" type="checkbox"/> Define Objectives	● Active
		Data Science... <input checked="" type="checkbox"/> identify business variables	● Active
		Data Science... <input checked="" type="checkbox"/> Define success metrics	● Active
		Data Science... <input checked="" type="checkbox"/> define accuracy	● 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 on the build agent.
- Visual Studio**
- Empty**  
Start with a definition that has no steps.

> Next Cancel

Deploy a website to Microsoft Azure

Chef Deploy to Chef environments by editing environment attributes

Chef Knife Run Scripts with knife commands on your chef workstation

Fabrikam Home Code Work Build & Release Test

Test Plans Parameters Configurations Runs Machines Load test

Select a test plan

+ Test plan Static suite Requirement-based suite Query-based suite Shared steps

You can't create test cases without a test plan

Tests Charts

New Add existing X

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

The screenshot shows a 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 to indicate it's the current page), Pricing, Training, Marketplace, Partners, Blog, Resources, and Support. There are also links for SALES 1-800-867-1389, CONTACT SALES, and a Microsoft logo. Below the navigation, the breadcrumb trail shows 'Azure / Machine Learning / Team Data Science Process'. On the left, there's a sidebar with a 'Filter' button and a list of topics under 'Lifecycle', 'Examples', 'Training', and 'For DevOps'. The main content area starts with the title 'What is the Team Data Science Process?' followed by a timestamp (10/20/2017), a reading time (4 minutes), and contributor icons. The text explains what TDSP is: '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.' Below this, another text block describes the overview of TDSP components and its implementation with various tools. The section 'Key components of the TDSP' is introduced, followed by a list of components: 'TDSP comprises of the following key components:' and a bullet point: '• 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

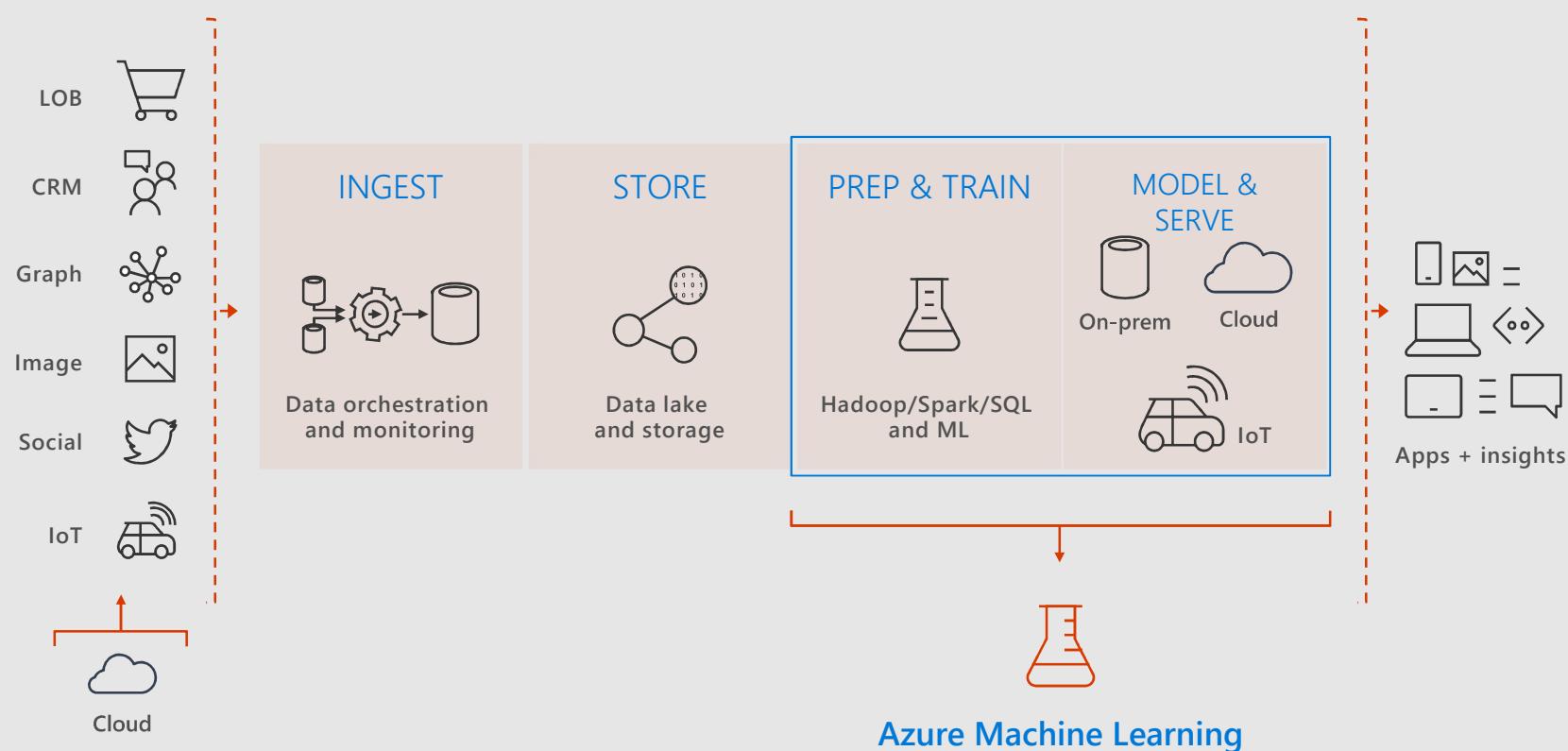
You can use the items in the following table to guide your own self-study. Read the *Description* column to follow the path, click on the *Topic* links for study references, and check your skills using the *Knowledge Check* column.

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

# Azure Machine Learning

<https://docs.microsoft.com/en-us/azure/machine-learning/service/>

# Azure Machine Learning



# Deploy everywhere



DOCKER

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

# Use what you want

Use your favorite IDE

Leverage all types of platforms and tools/libraries

USE ANY FRAMEWORK OR LIBRARY



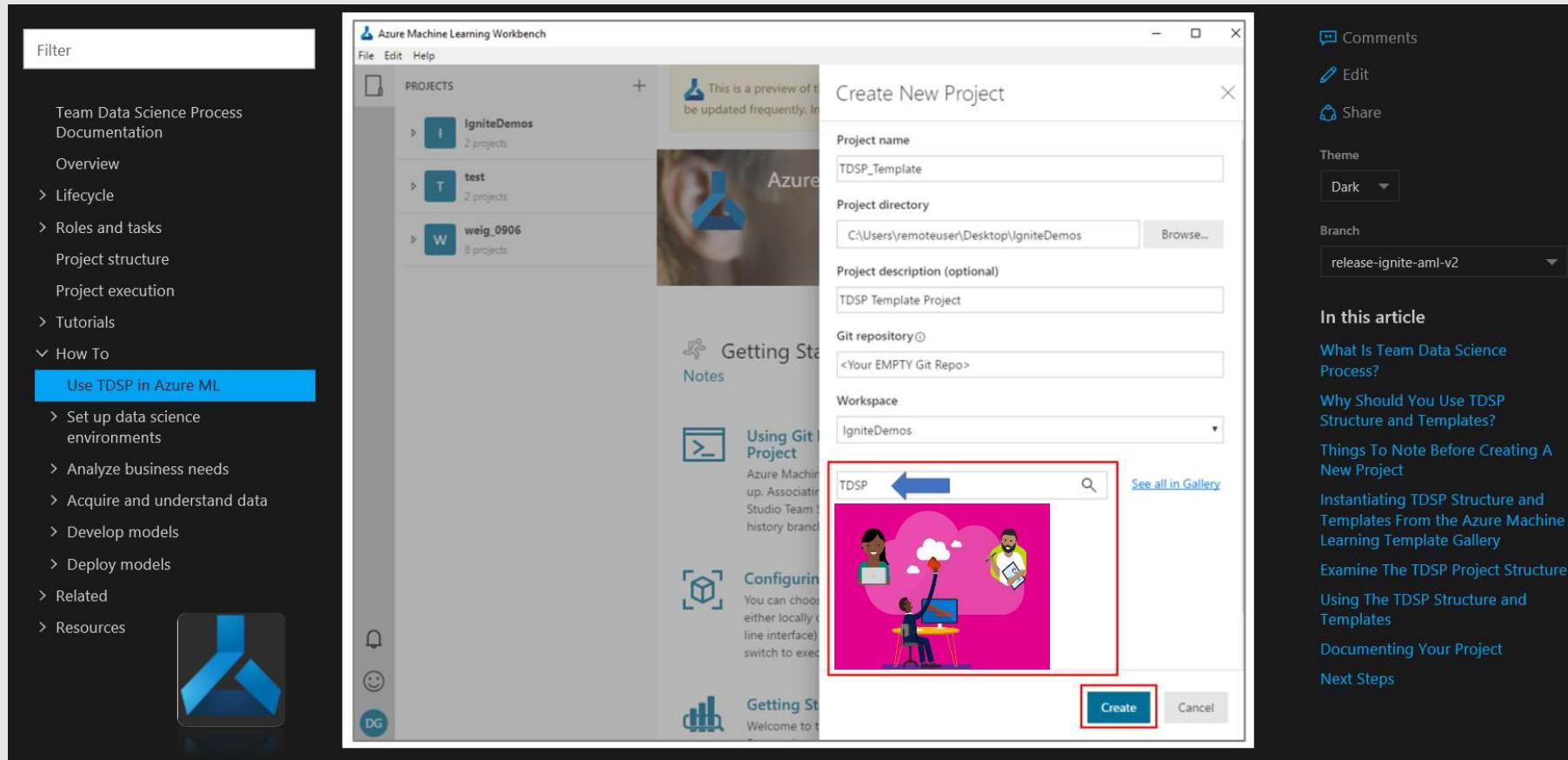
USE ANY TOOL



USE THE MOST POPULAR INNOVATIONS



# Using TDSP with Azure Machine Learning



# TDSP – AML worked-out samples in AI

21

- 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 related to sentiment analysis. At the top, it displays statistics: 60 commits, 1 branch, 0 releases, 3 contributors, and an MIT license. Below this, a commit history table lists several commits made by user 'wguo123'. The commits include updates to README files, configuration files, and documentation. The last commit was made 26 days ago. At the bottom of the page, there is a text box containing 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>

# Developing & deploying NLP services on Azure

# *Sentiment classification*

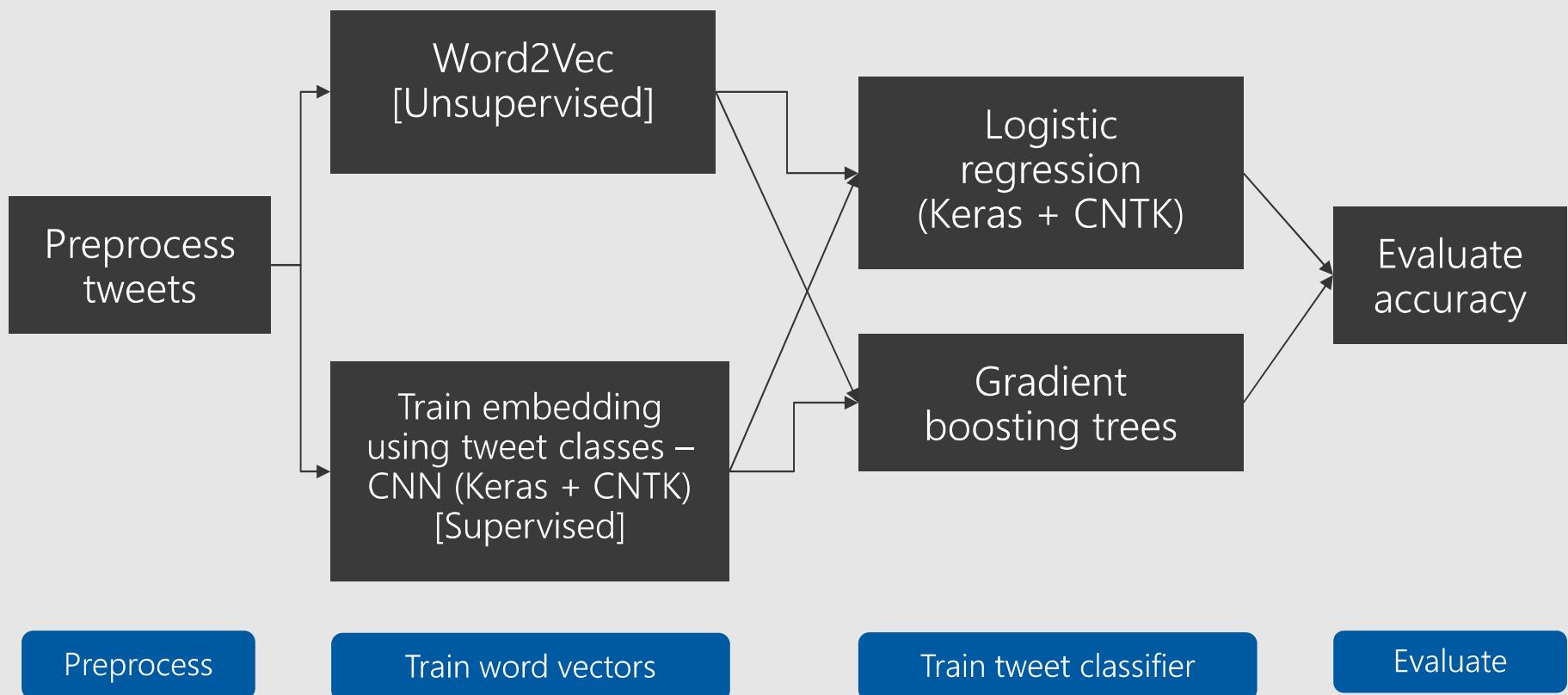
<https://github.com/Azure/MachineLearningSamples-TwitterSentimentPrediction>

# Use case introduction: Twitter sentiment prediction

- Predict Twitter sentiment polarity using word embedding and DL for classification
- Objective & scientific question
  - Demonstrate how to develop and deploy sentiment classification service on Azure
  - Can supervised word embeddings improve accuracy of classification accuracy of sentiment class prediction?
- Dataset
  - “sentiment140”
  - 1.3 million tweets for training
  - 320,000 for testing
  - Classes - only two, positive and negative (neutral tweets were removed for this exercise)

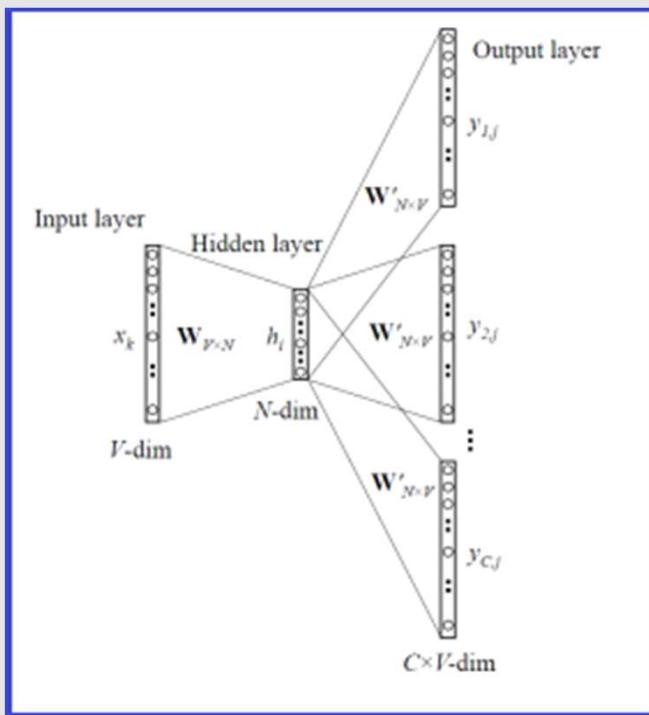
# Analysis strategy

- Train word embeddings in 2 different ways
  - Unsupervised: Word2Vec
  - Supervised: Using CNN to generate classification based embeddings
- Train classifier (GBT, Logistic regression using Keras)
- Evaluate accuracy on test set

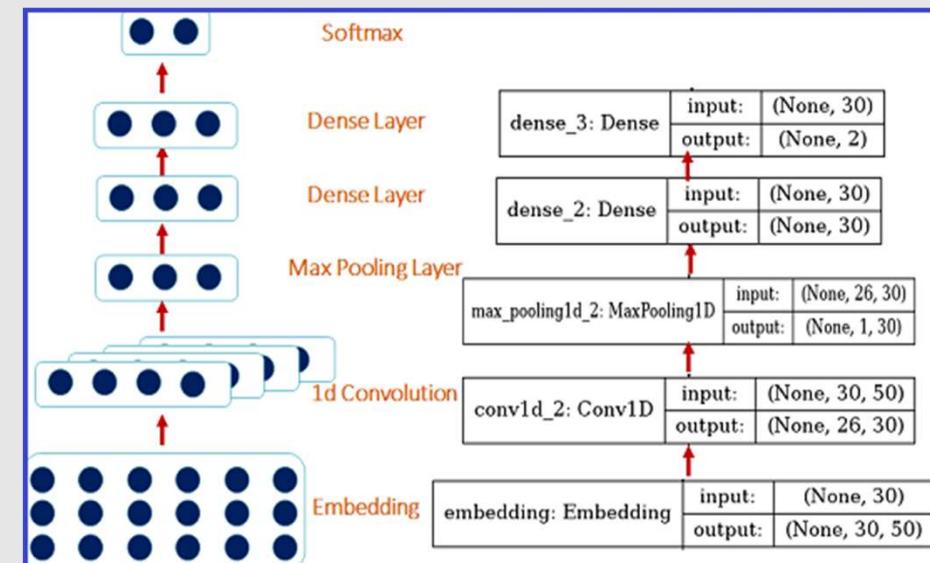


# Word embedding generation

Word2Vec: skip-gram



Supervised word embeddings using CNN



# Supervised embedding generation using CNN

27

- code

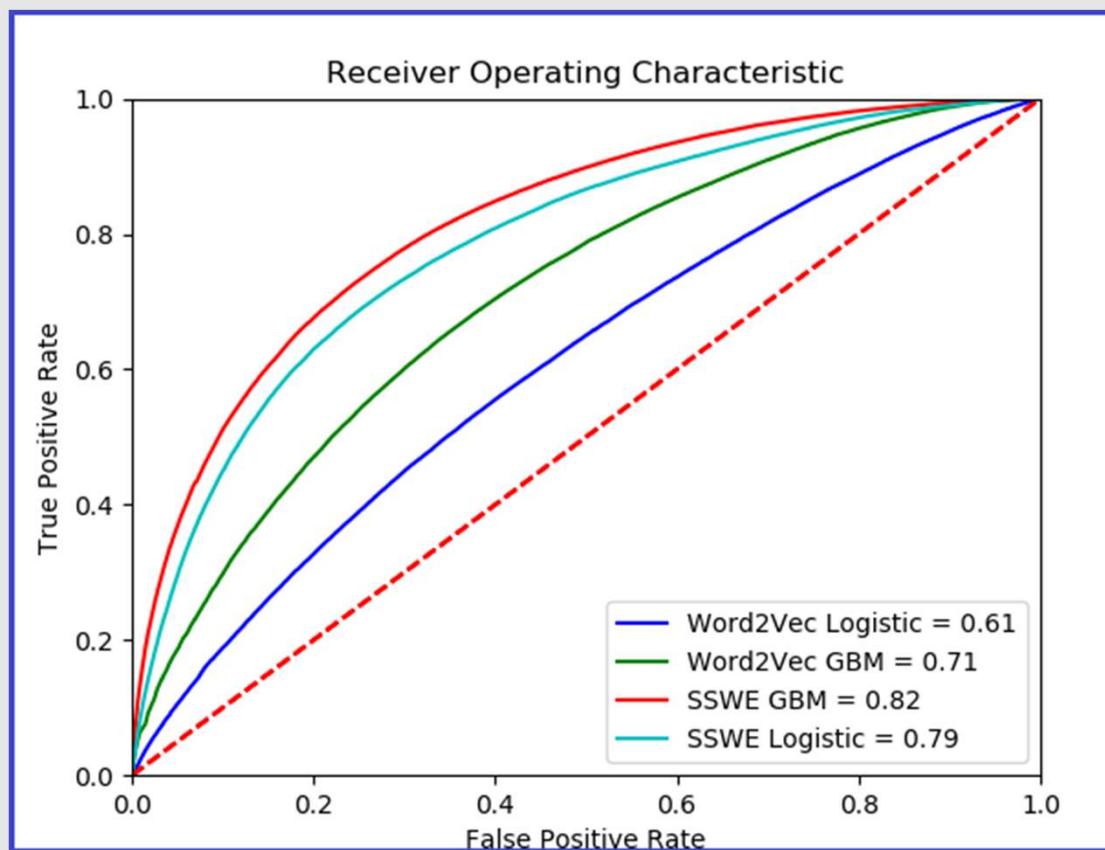
```
# Model Instantiation
print ('Initializing the model')
mcp = ModelCheckpoint('./model_chkpoint', monitor="val_acc", save_best_only=True, save_weights_only=False)

#Creating network
model = Sequential()
model.add(Embedding(len(word_index)+2,
                    embedding_dim,
                    input_length=max_sequence_length,
                    trainable=trainable, name='embedding'))
model.add(Convolution1D(no_filters, filter_size, activation='relu'))
model.add(MaxPooling1D(max_sequence_length - filter_size))
model.add(Flatten())
model.add(Dense(no_filters, activation='tanh'))
model.add(Dense(len(labels[0])), activation='softmax'))

optim=optimizers.Adam(lr=0.1, )
model.compile(loss='categorical_crossentropy',
              optimizer=optim,
              metrics=['acc'])
model.summary()
```

# Model evaluation: Sentiment-specific word embeddings (SSWE) improves classification of sentiments

28



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

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

# Model deployment

- The operationalization environment provisions Docker and Kubernetes in the cluster to manage the web-service deployment

```
az ml service create realtime --model-file (model file name) -f (scoring script name) -n (your-new-acctname) -s (web service schema json file) -r (compute environment, python or PySpark, etc) -d (dependency files)
```

The screenshot shows the Kubernetes UI interface. At the top left is the 'kubernetes' logo. To its right is a search bar with a magnifying glass icon and the word 'Search'. Below the header, a blue navigation bar contains the text 'Workloads > Pods'. On the left side, there's a sidebar with options: 'Cluster', 'Namespaces', 'Nodes', and 'Persistent Volumes'. The 'Namespaces' option is currently selected. The main content area is titled 'Pods'. It features a table with three columns: 'Name', 'Status', and 'Restarts'. A single row is visible in the table, corresponding to the selected namespace. The pod name is 'weigtwitter0913v13-723221451-l772r', its status is 'Running', and it has '0' restarts.

Name	Status	Restarts
weigtwitter0913v13-723221451-l772r	Running	0

# *Named entity recognition (NER)*

<https://github.com/Azure/MachineLearningSamples-BiomedicalEntityExtraction>

# Use case introduction: Biomedical named entity recognition using LSTMs

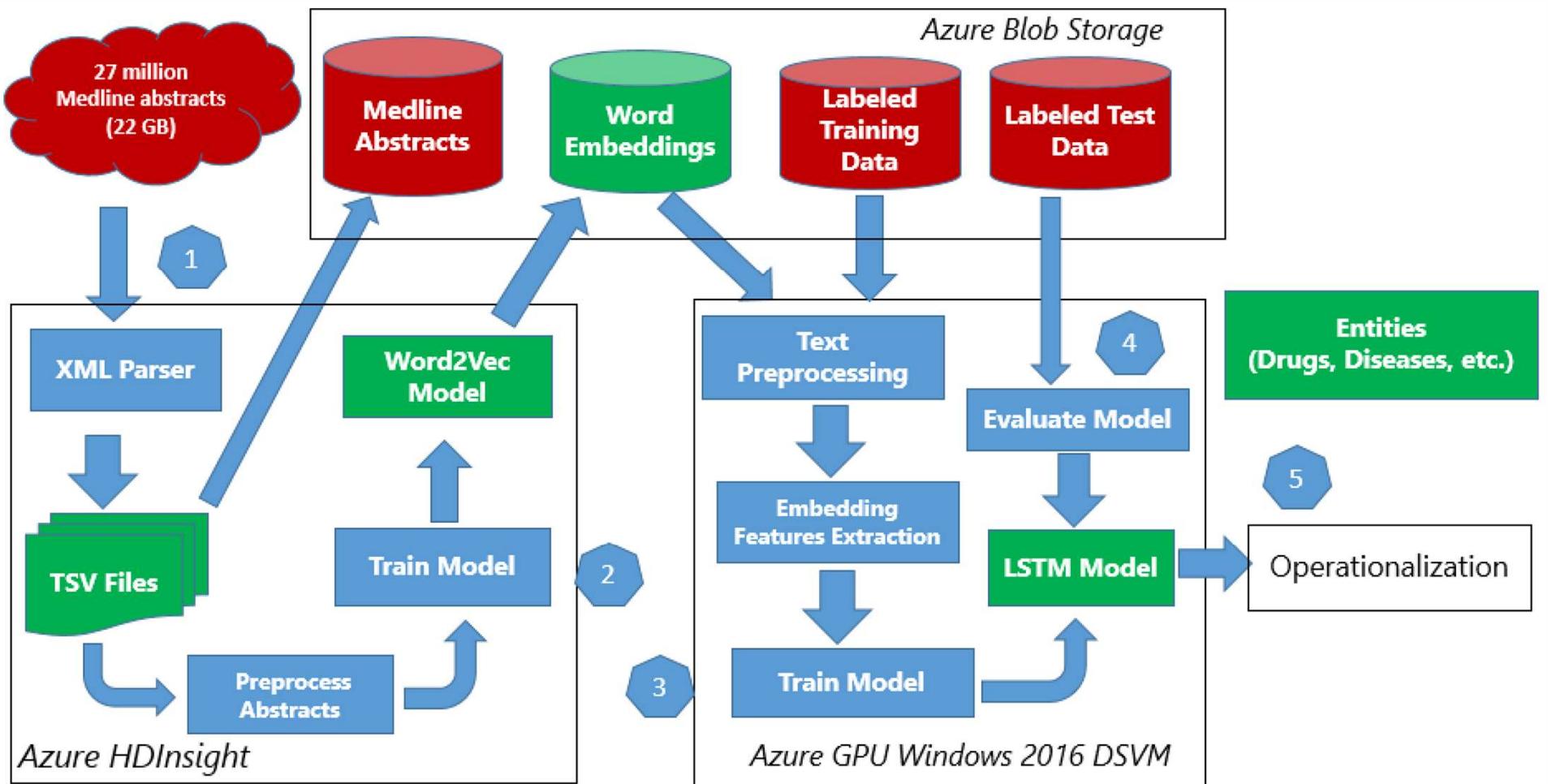
- Identify entities drugs, chemicals & diseases from medical abstracts (PubMed) or medical documents such as electronic health records (EHRs)
  - E.g. Create structured information from unstructured data
- Objective & question
  - Train a scalable model for NER using word vectors and DL architecture
  - Can we improve NER accuracy using domain-specific (biomedical) datasets for word embedding, rather than a generic dataset (e.g. Google news articles)?
- Datasets
  - Word embeddings: Entire set of PubMed abstracts, ~30 MIL
  - Drugs and disease - labeled data-set
  - Biological entities (protein, DNA, RNA etc.) – labeled data

# Labeled entity data-sets

```
Famotidine-associated    B-Chemical  
delirium      B-Disease  
.      0  
  
A      0  
series 0  
of     0  
six    0  
cases   0  
.      0  
  
Famotidine    B-Chemical  
is      0  
a      0  
histamine  0  
H2-receptor 0  
antagonist  0  
used    0  
in      0  
inpatient 0  
settings  0  
for     0  
prevention 0  
of      0  
stress   0  
ulcers   B-Disease  
and    0  
is      0  
showing 0  
increasing 0  
popularity 0  
because 0  
of      0  
its    0  
low    0  
cost   0  
.      0
```

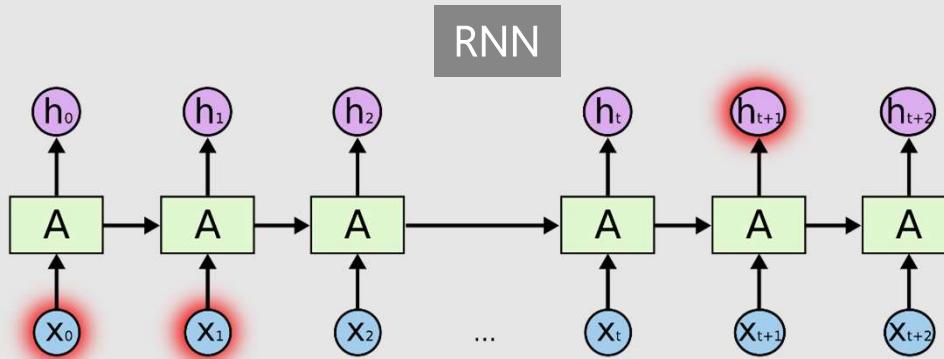
- Training: Typically labeled data containing
  - 500-2,000 abstracts
  - 8,000-20,000 sentences
- Test: Typically labeled data
  - 200-400 abstracts
  - 600-4,000 sentences

# Architecture: Data pipeline and experimental setup



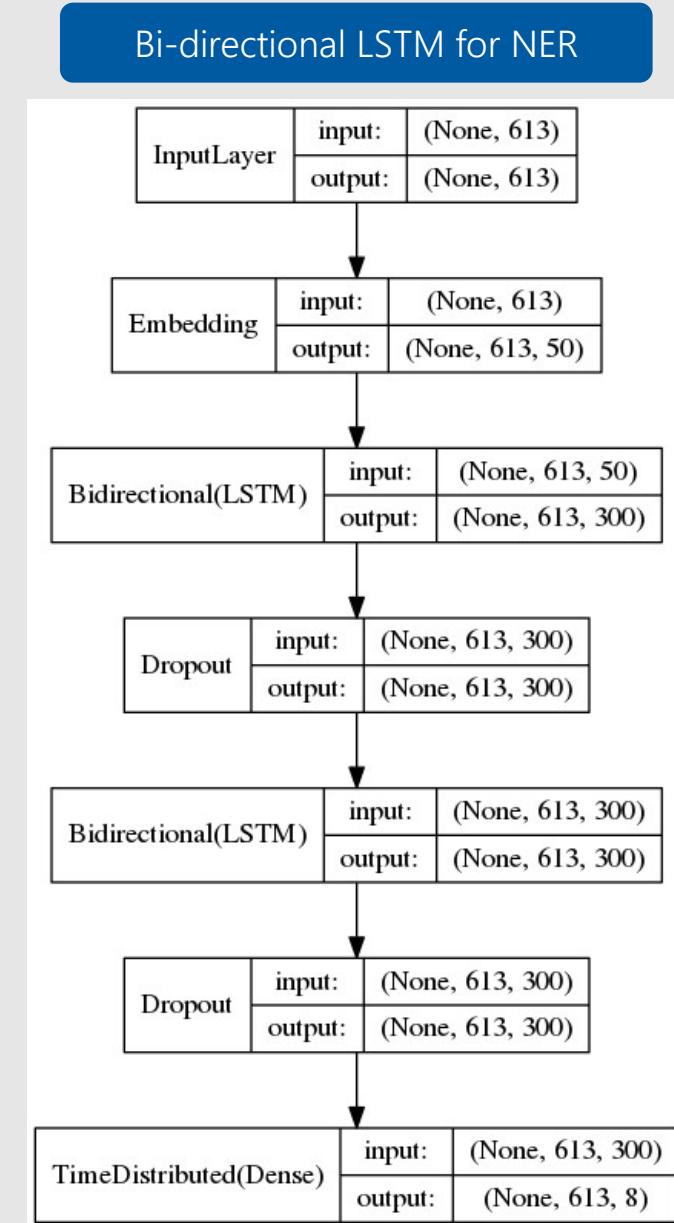
# Word2Vec & Bi-directional LSTM architecture

- Word2Vec training:
  - Spark MLlib implementation
  - Azure HDInsight Spark cluster
  - 11 nodes
- LSTM training:
  - GPU – DSVM (data science VM)



LSTM - Long Short Term Memory networks -  
special kind of RNN, capable of learning long-term dependencies (details not shown)

<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>



# Bi-directional LSTM for NER - code

```
self.model = Sequential()
self.model.add(Embedding(self.wordvecs.shape[0], self.wordvecs.shape[1], \
                        input_length = train_X.shape[1], \
                        weights = [self.wordvecs], trainable = False))

for i in range(0, num_layers):
    if network_type == 'unidirectional':
        # uni-directional LSTM
        self.model.add(LSTM(num_hidden_units, return_sequences = True))
    else:
        # bi-directional LSTM
        self.model.add(Bidirectional(LSTM(num_hidden_units, return_sequences = True)))

    self.model.add(Dropout(dropout))

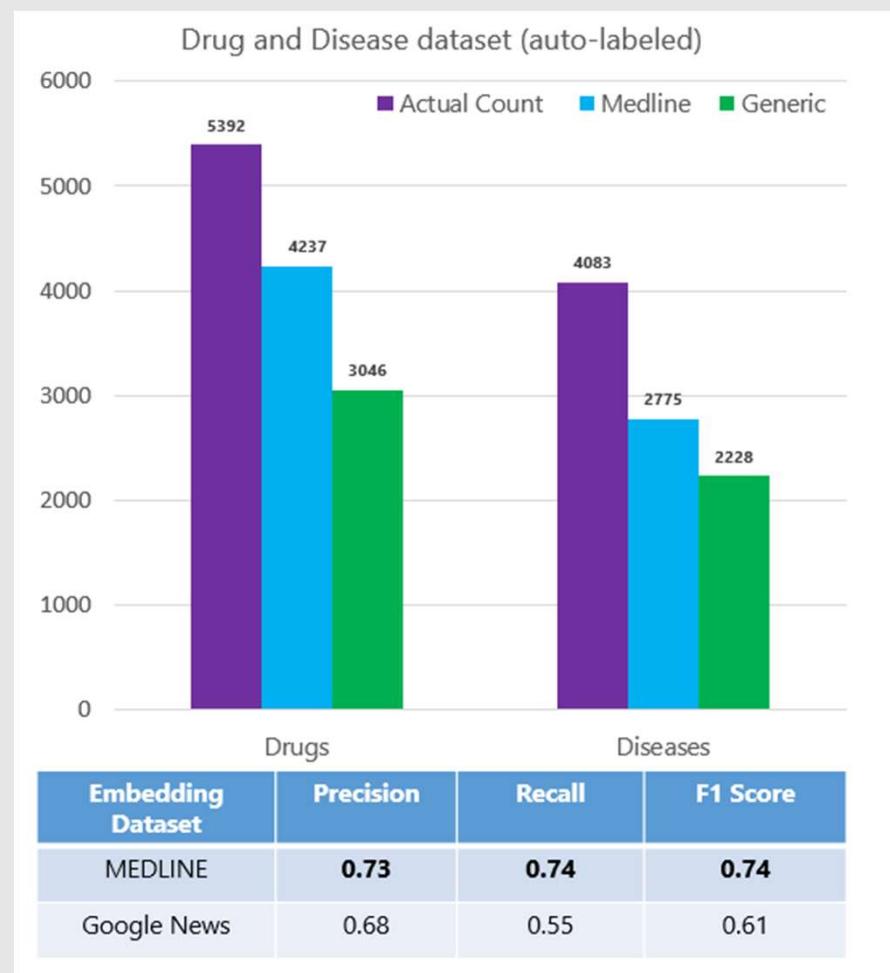
self.model.add(TimeDistributed(Dense(train_Y.shape[2], activation='softmax')))

self.model.compile(loss='categorical_crossentropy', optimizer='adam')
print(self.model.summary())

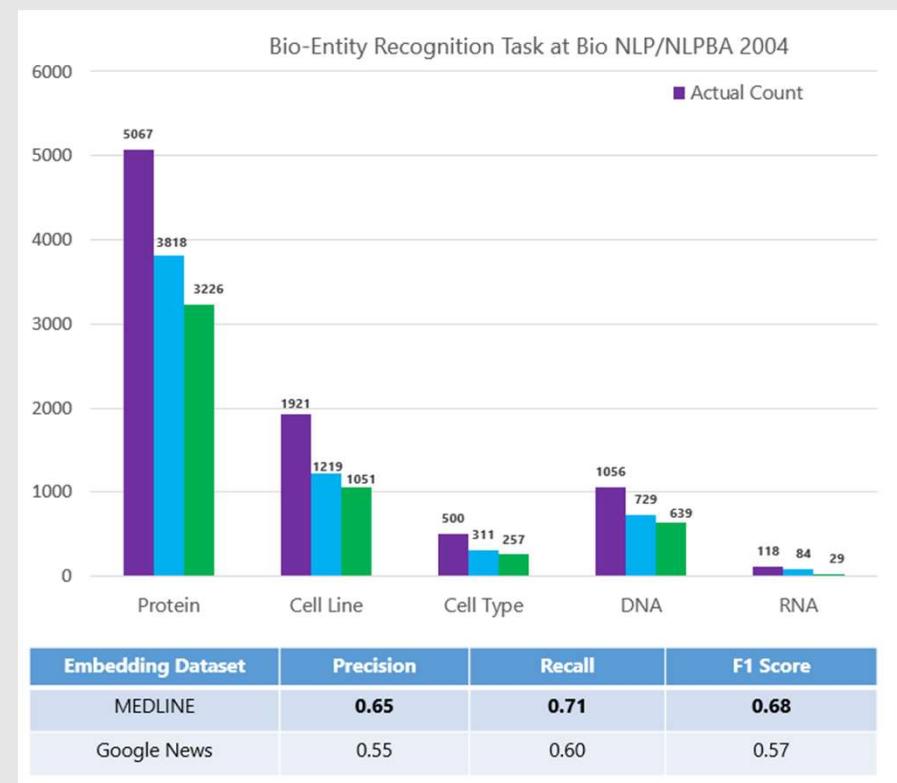
self.model.fit(train_X, train_Y, epochs = num_epochs, batch_size = batch_size)
```

# Biomedical NER model evaluation

Doman-specific embedding improve performance of NER identification



Semeval 2013 - Task 9.1 (Drug Recognition)



BioCreative V CDR task

# Detection of entities – example with biomedical abstract from PubMed: Web service

## Entity Extractor

Baricitinib, Methotrexate, or Baricitinib Plus Methotrexate in Patients with Early Rheumatoid Arthritis Who Had Received Limited or No Treatment with Disease-Modifying-Anti-Rheumatic-Drugs (DMARDs): Phase 3 Trial Results.

Keywords: Janus kinase (JAK), methotrexate (MTX) and rheumatoid arthritis (RA) and Clinical research.

In 2 completed phase 3 studies, baricitinib (bari) improved disease activity with a satisfactory safety profile in patients (pts) with moderately-to-severely active RA who were inadequate responders to either conventional synthetic or biologic2DMARDs. This abstract reports results from a phase 3 study of bari administered as monotherapy or in combination with methotrexate (MTX) to pts with early active RA who had limited or no prior treatment with DMARDs. MTX monotherapy was the active comparator.

Bcl-3 protein was also highly expressed in early burst-forming-unit-erythroid (BFU-E)-derived erythroid precursors (day 7) and decreased during maturation (days 10 and 14), suggesting that Bcl-3 is involved in normal erythroid proliferation. In these hematopoietic cells, Bcl-3 was hyperphosphorylated. GM-CSF and Epo modulated the subcellular localization of Bcl-3. Upon stimulation of TF-1 cells with GM-CSF or Epo, the nuclear translocation of Bcl-3 was dramatically enhanced. Overexpression of Bcl-3 in TF-1 cells by transient transfection along with the NF-kappaB factors p50 or p52 resulted in significant induction of an human immunodeficiency virus-type 1 ( HIV-1 ) kappaB-TATA-luciferase reporter plasmid, demonstrating that Bcl-3 has a positive role in transactivation of kappaB-containing genes in erythroid cells. Stimulation with GM-CSF enhanced c-myb mRNA expression in these cells. Bcl-3 in nuclear extracts of TF-1 cells bound to a kappaB enhancer in the c-myb promoter together with NF-kappaB2/p52 and this binding activity was enhanced by GM-CSF stimulation. Furthermore, cotransfection of Bcl-3 with p52 or p50 in TF-1 cells resulted in significant activation of a c-myb kappaB-TATA-luciferase reporter plasmid. Selective Serotonin Reuptake Inhibitors ( SSRIs ) : SSRIs ( e.g., fluoxetine, fluvoxamine, paroxetine, sertraline ) have been rarely reported to cause weakness, hyperreflexia, and incoordination when coadministered with 5-HT1 agonists. If concomitant treatment with AXERT and an SSRI is clinically warranted, appropriate observation of the patient is advised.

[See Results](#)

## Entity Extractor Output

[See Results](#)

## Entity Extractor Output

Key: Disease, Drug or Chemical, DNA, RNA, Protein, Cell Line, Cell Type

Baricitinib, Methotrexate, or Baricitinib Plus Methotrexate in Patients with Early Rheumatoid Arthritis Who Had Received Limited or No Treatment with Disease-Modifying-Anti-Rheumatic-Drugs (DMARDs) : Phase 3 Trial Results.

Keywords : Janus kinase (JAK), methotrexate (MTX) and rheumatoid arthritis (RA) and Clinical research.

In 2 completed phase 3 studies, baricitinib (bari) improved disease activity with a satisfactory safety profile in patients (pts) with moderately-to-severely active RA who were inadequate responders to either conventional synthetic or biologic2DMARDs.

This abstract reports results from a phase 3 study of bari administered as monotherapy or in combination with methotrexate (MTX) to pts with early active RA who had limited or no prior treatment with DMARDs.

MTX monotherapy was the active comparator.

# Summary

- TDSP (Team Data Science Process) provides a standardized process for development & deployment of AI services on Azure
- Resources are available to use TDSP and AML together, along with data platforms on Azure to develop and deploy AI solutions
- Common DL use cases are provided as examples in open GitHub repos, with experiments on how to improve performance

# Thank you!

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

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