The machine learning process

From ideation to deployment with Azure Machine Learning

Francesca Lazzeri, PhD – Senior Machine Learning Scientist Lead

@frlazzeri
aka.ms/AzureMLrepo
medium.com/@francescalazzeri

Microsoft

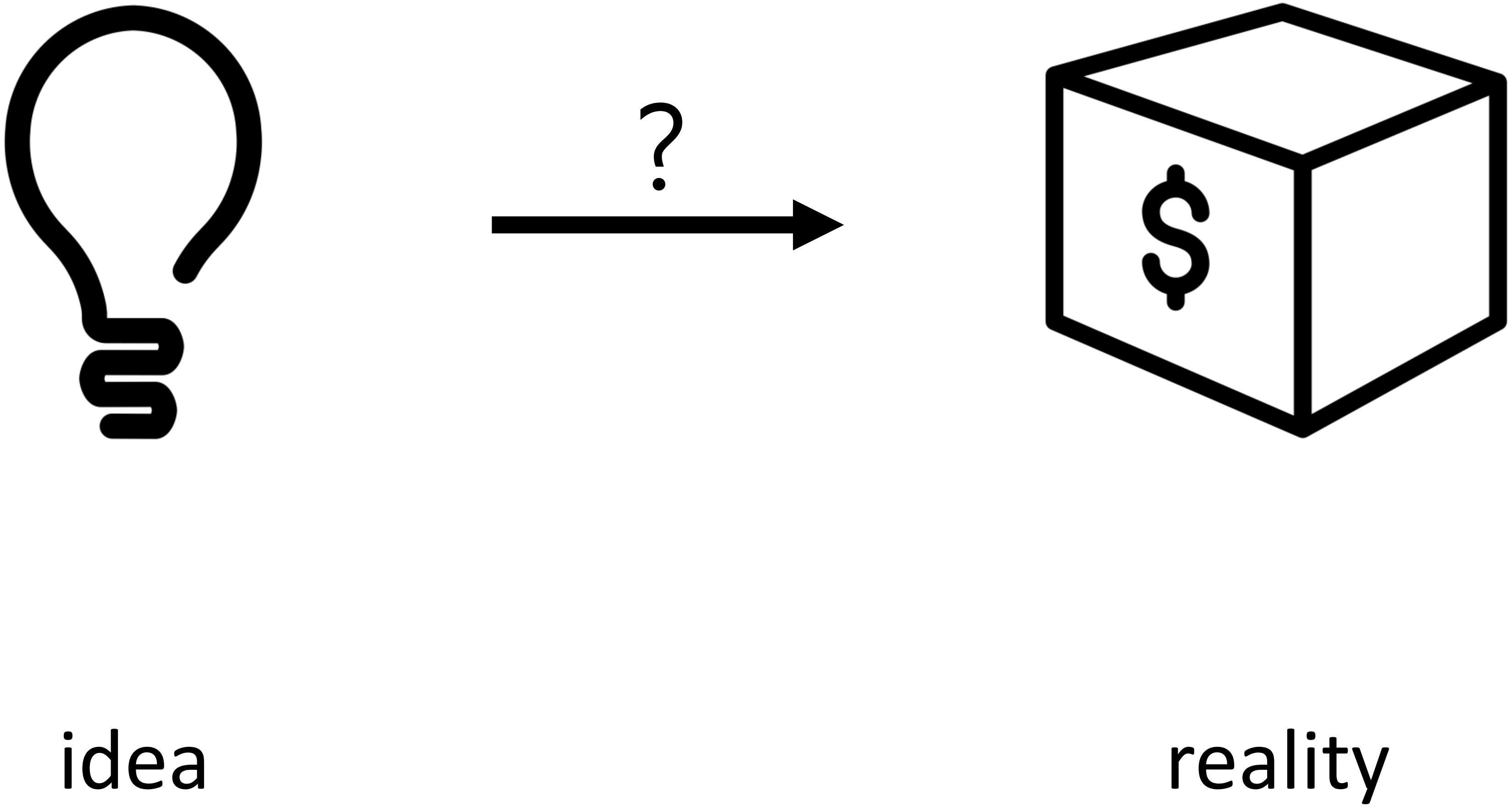
HDSI | Harvard Data Science Initiative

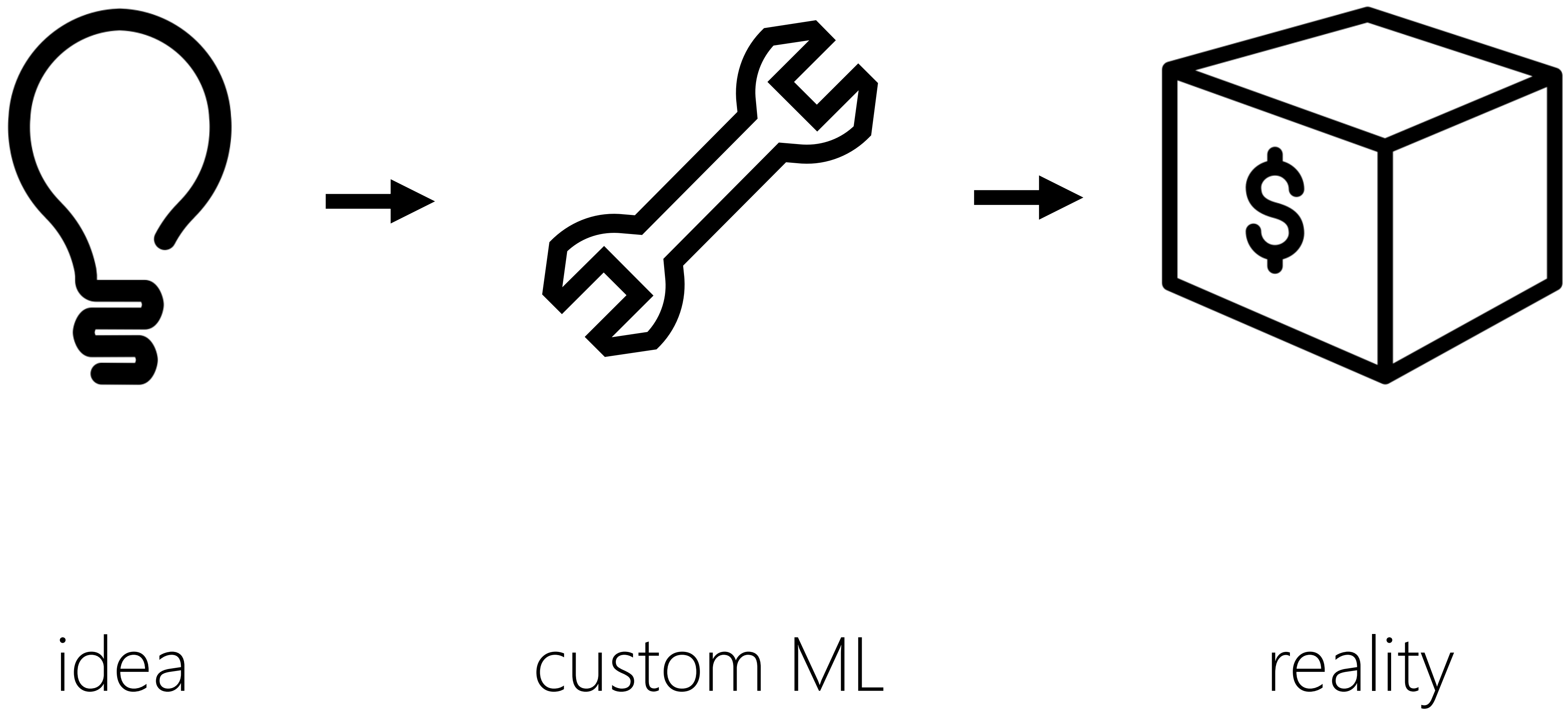
Useful Resources:

@frlazzeri

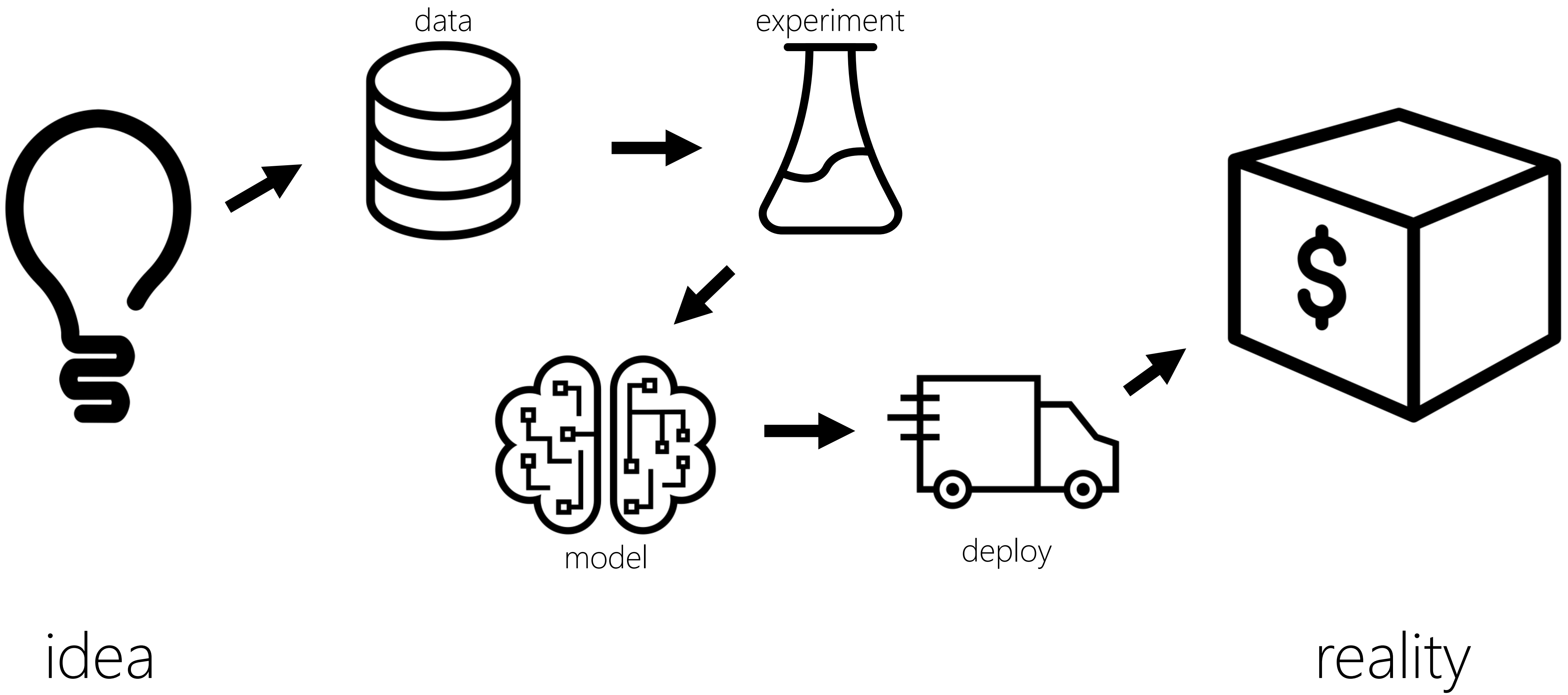
- ✓ AzureML GitHub: aka.ms/AzureMLrepo
- ✓ Algorithm Cheat Sheet: aka.ms/AlgorithmCheatSheet
- ✓ Deep Learning VS Machine Learning: aka.ms/DeepLearningVSMachineLearning
- ✓ Automated Machine Learning Documentation: aka.ms/AutomatedMLDocs
- ✓ Model Interpretability with Azure ML Service: aka.ms/AzureMLModelInterpretability
- ✓ Azure Machine Learning Service: aka.ms/AzureMLservice
- ✓ Azure Machine Learning Designer: aka.ms/AzureMLdesigner
- ✓ Get started with Azure ML: aka.ms/GetStartedAzureML
- ✓ Azure Notebooks: aka.ms/AzureNB

@frlazzeri




idea
custom ML
reality

@frlazzeri



There are many jobs & tools involved in production ML

@frlazzeri

Data Scientist

Azure Machine Learning
GitHub
TensorFlow, PyTorch, sklearn
Azure Compute – CPU/GPU/FPGA

IT / Ops

Data Analyst

Business Owner

& many more…

Data Engineer

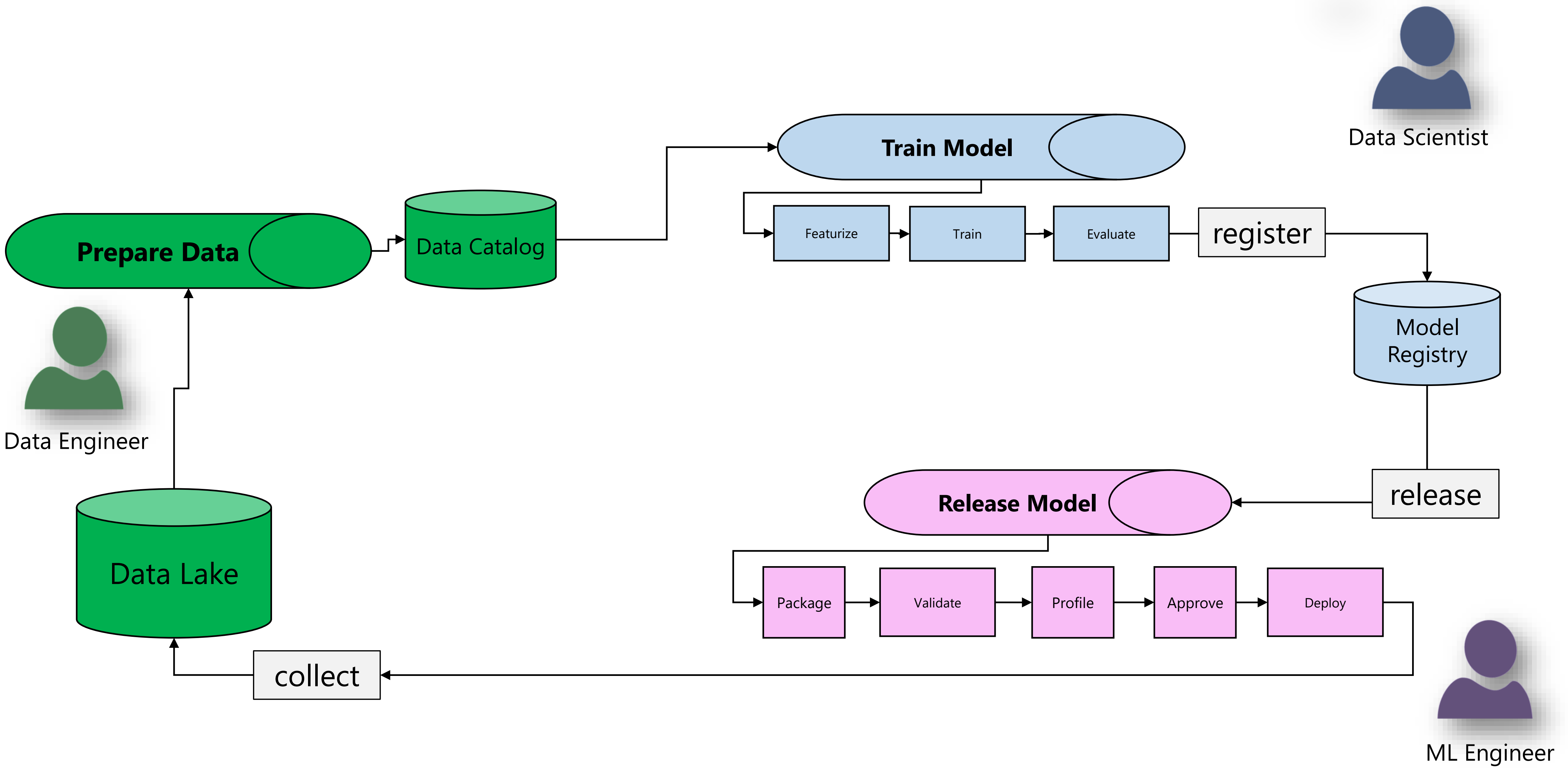
Azure Data Lake
Azure Data Factory
Azure DataBricks
Azure SQL

Azure DevOps
GitHub
Azure Kubernetes Service
Azure IoT Edge
Azure Monitor

ML Engineer

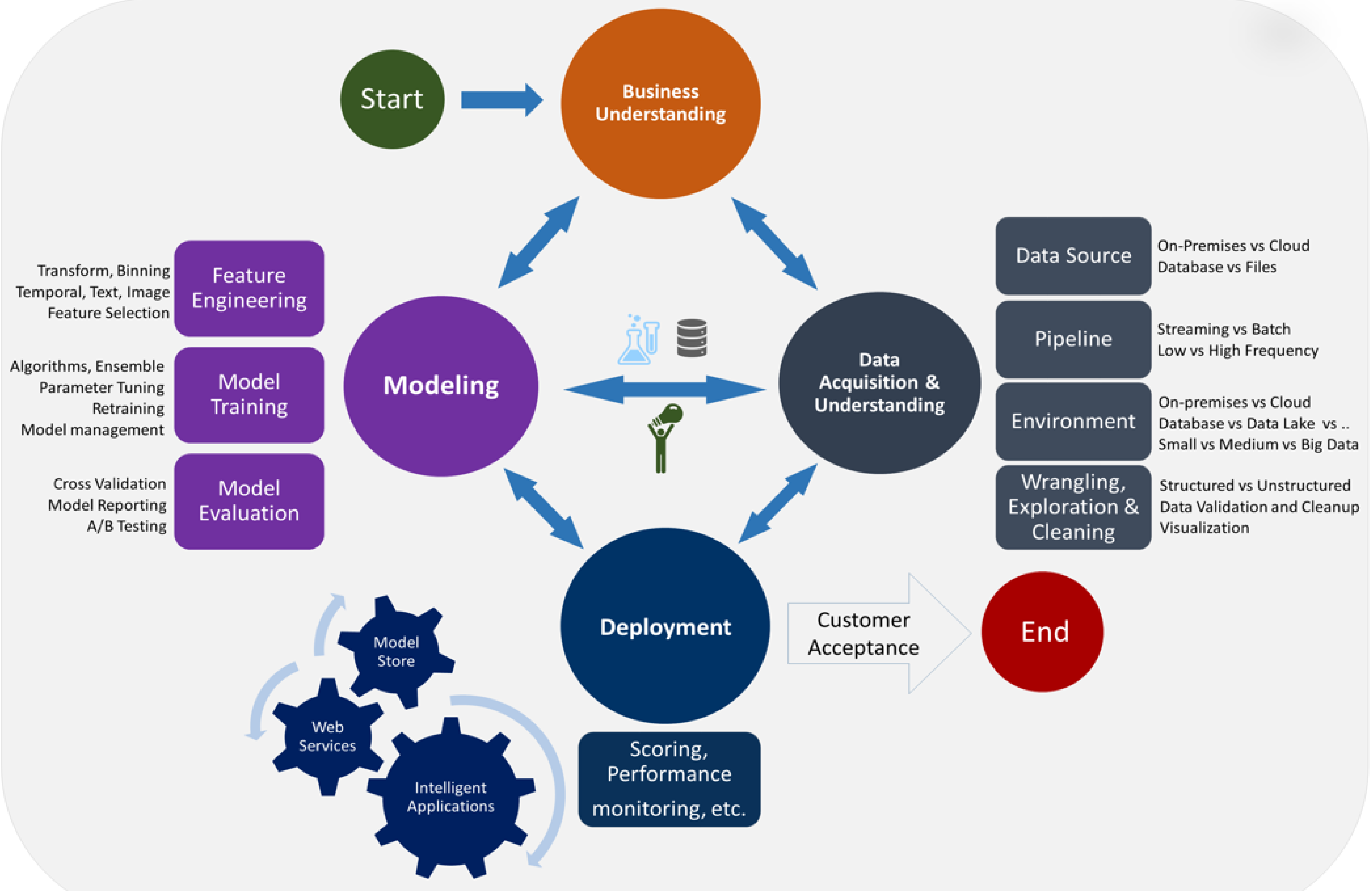
E2E Machine Learnin process

@frlazzeri



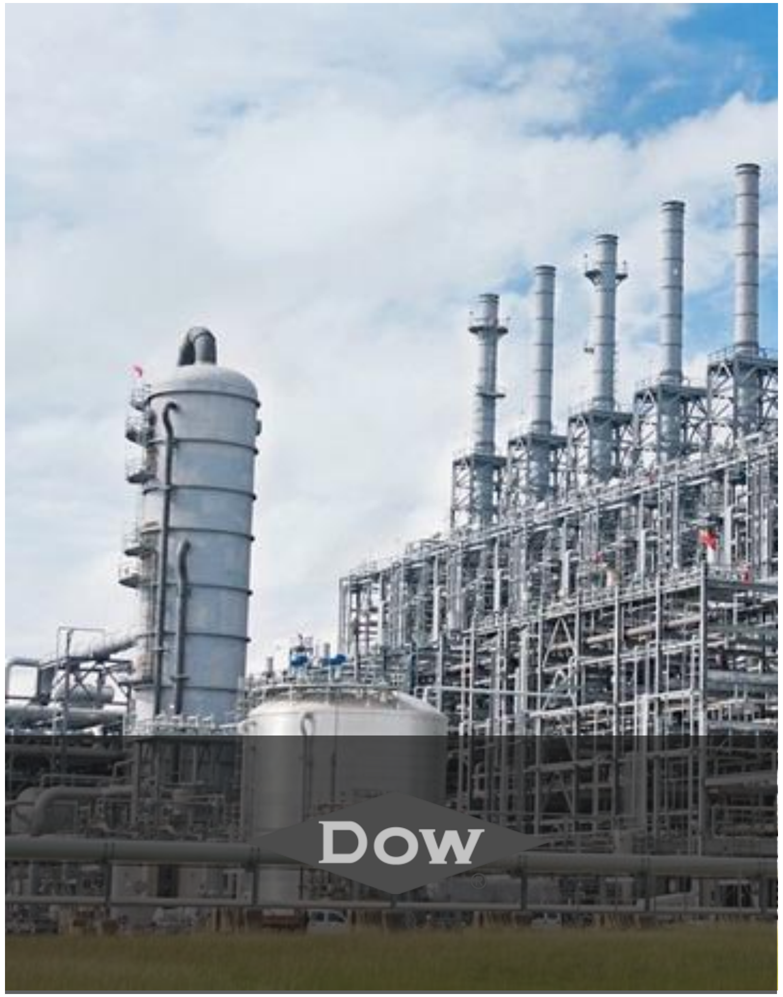
The Machine Learning Process

@frlazzeri



@frlazzeri

Example - Inventory Optimization

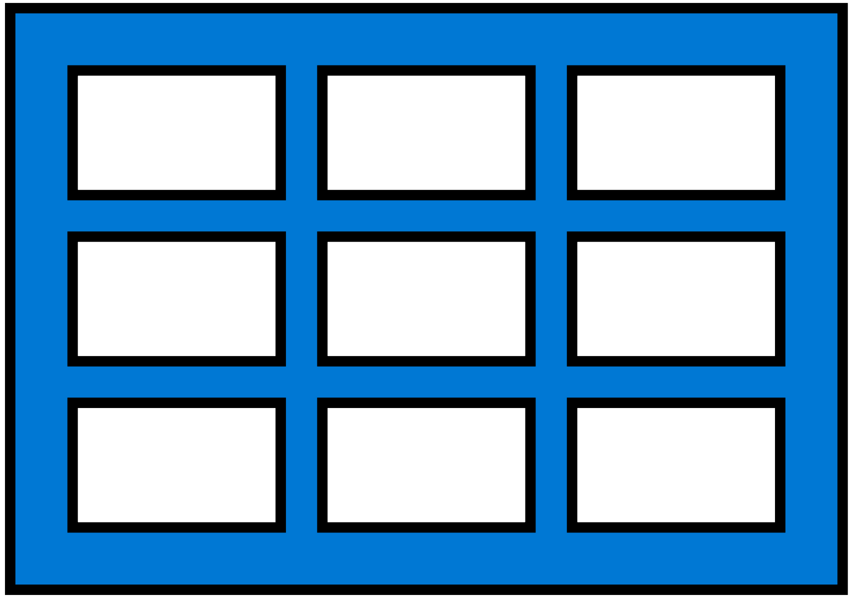


Inventory Optimization

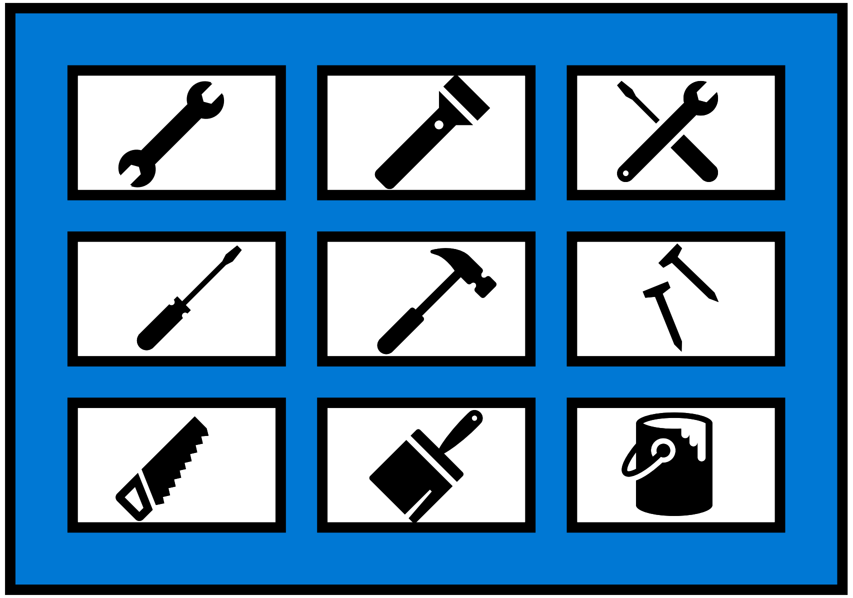
@frlazzeri

Balancing capital investment constraints or objectives and service-level goals over a large assortment of stock-keeping units (SKUs) while taking demand and supply volatility into account.

Store 1



Store 2



Business Understanding

- ✓ Ask the right questions
- ✓ Define Performance Metrics
- ✓ Understand what to do with your data

Understand what to do with your data

aka.ms/AlgorithmCheatSheet

What do you want to do?

Extract information from text

Text Analytics

Derives high-quality information from text

Answers questions like: What info is in this text?

Extract N-Gram Features from Text	Creates a dictionary of n-grams from a column of free text
Feature Hashing	Converts text data to integer encoded features using the Vowpal Wabbit library
Preprocess Text	Performs cleaning operations on text, like removal of stop-words, case normalization
Word2Vector	Converts words to values for use in NLP tasks, like recommender, named entity recognition, machine translation

Predict values

Regression

Makes forecasts by estimating the relationship between values

Answers questions like: How much or how many?

Fast Forest Quantile Regression	Predicts a distribution
Poisson Regression	Predicts event counts
Linear Regression	Fast training, linear model
Bayesian Linear Regression	Linear model, small data sets
Decision Forest Regression	Accurate, fast training times
Neural Network Regression	Accurate, long training times
Boosted Decision Tree Regression	Accurate, fast training times, large memory footprint

Generate recommendations

Recommenders

Predicts what someone will be interested in

Answers the question: What will they be interested in?

SVD Recommender	Collaborative filtering, better performance with lower cost by reducing dimensionality

Discover structure

Clustering

Separates similar data points into intuitive groups

Answers questions like: How is this organized?

K-Means	Unsupervised learning

Find unusual occurrences

Anomaly Detection

Identifies and predicts rare or unusual data points

Answers the question: Is this weird?

One Class SVM	Under 100 features, aggressive boundary
PCA-Based Anomaly Detection	Fast training times

Predict between several categories

Multiclass Classification

Answers complex questions with multiple possible answers

Answers questions like: Is this A or B or C or D?

Multiclass Logistic Regression	Fast training times, linear model
Multiclass Neural Network	Accuracy, long training times
Multiclass Decision Forest	Accuracy, fast training times
One-vs-All Multiclass	Depends on the two-class classifier
Multiclass Boosted Decision Tree	Non-parametric, fast training times and scalable

Predict between two categories

Two-Class Classification

Answers simple two-choice questions, like yes or no, true or false

Answers questions like: Is this A or B?

Two-Class Support Vector Machine	Under 100 features, linear model
Two-Class Averaged Perceptron	Fast training, linear model
Two-Class Decision Forest	Accurate, fast training
Two-Class Logistic Regression	Fast training, linear model
Two-Class Boosted Decision Tree	Accurate, fast training, large memory footprint
Two-Class Neural Network	Accurate, long training times

Classify images

Image Classification

Classifies images with popular networks

Answers questions like: What does this image represent?

DenseNet	High accuracy, better efficiency

Business Understanding

@frlazzeri

Inventory optimization

Demand plans
Forecasts
Sales history
Trends
Local events/weather patterns

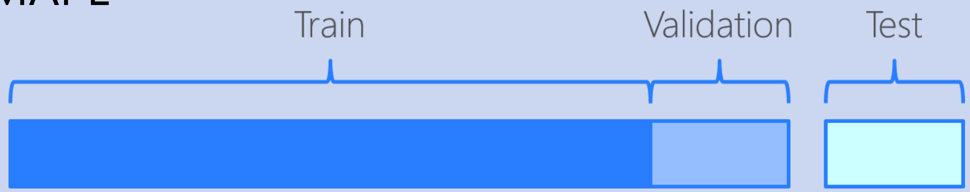
Data-driven stock, inventory, ordering

Predict inventory positions and distribution

Fraud detection

Market basket analysis

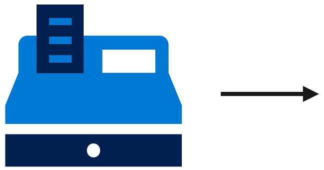
Omni-channel shopping experience with machine learning

Ask the right questions	What are the forecasted sales quantities per item per store for the next 4 weeks? Using forecasting models such as determining reorder points and economic order quantities can help ensure optimal inventory control.
Define Performance Metrics	Evaluation metric: MAPE $\frac{100}{N}\sum_{i=1}^{N}\left\lvert\frac{y_i-\hat{y}_i}{y_i}\right\rvert$ 
Understand what to do with your data	Regression – Time Series Forecasting approach

Data Acquisition & Understanding

- ✓ Data Architecture
- ✓ Data preprocessing
- ✓ Feature Engineering

Data



POS data

Stores

Information about the 45 stores, indicating the type and size of store

Features

Contains additional data related to the store, department, and regional activity for the given dates.

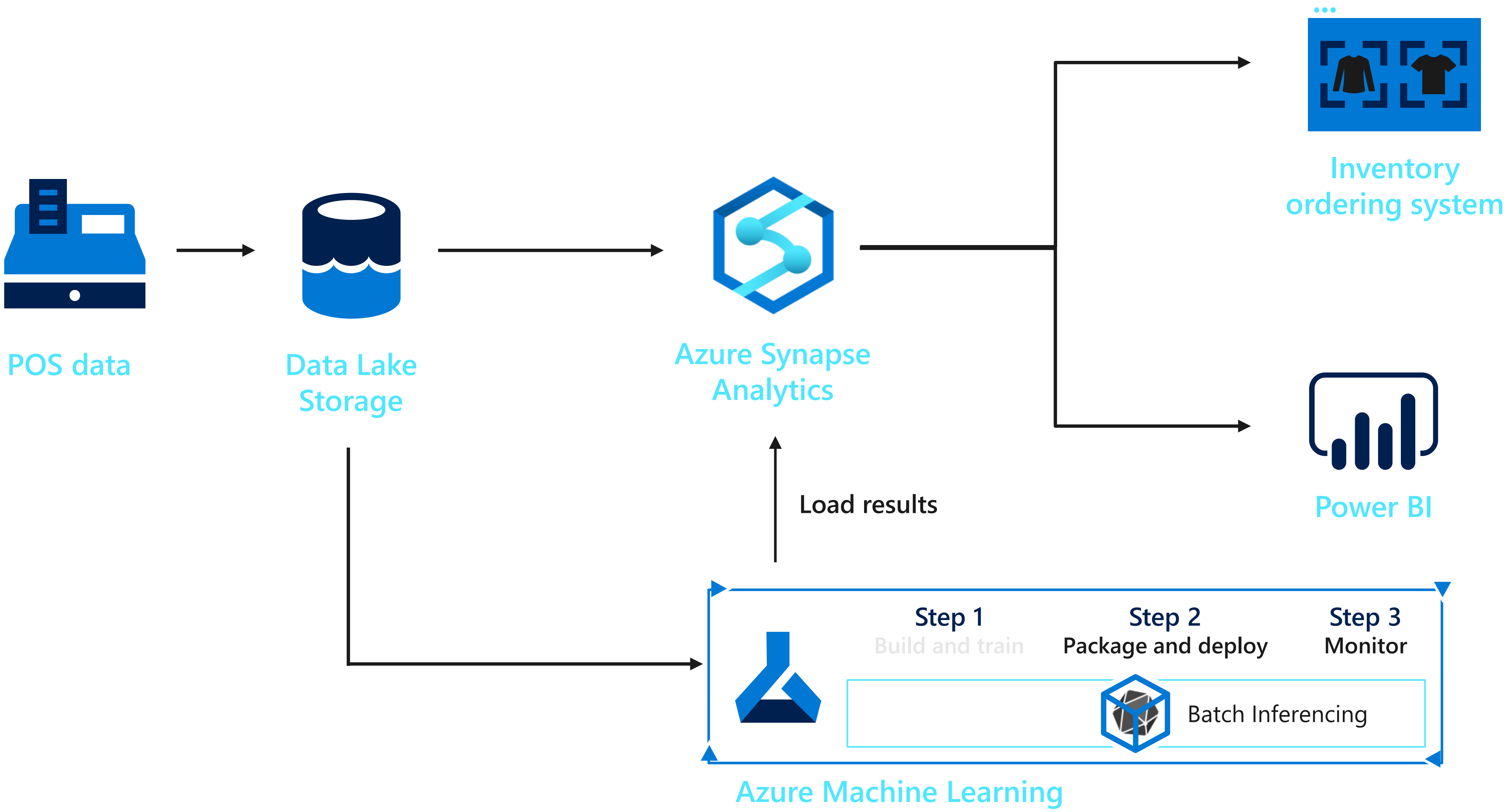
- Store - the store number
- Date - the week
- Temperature - average temperature in the region
- Fuel_Price - cost of fuel in the region
- MarkDown1-5 - anonymized data related to promotional markdowns.
- RDPI - Real Disposable Personal Income
- Unemployment - the unemployment rate

Sales

Historical weekly sales data, which covers 3 years:

- Store - the store number
- Dept - the department number
- Date - the week
- Weekly_Sales - sales for the given department in the given store

Inventory management architecture



Find Features and Process Data

@frlazzeri

Find and validate if enough data is available

Gather external data to improve model performance

Add future dates to dataset to get prediction

Find and Validate Data

- Get store id, item id, time, and quantities of items sold
- Validate there is data for at least a year
- Fill in missing gaps in data

Starting Dataset

ID1	ID2	time	value
1	2	11/13/2010 12:00:00 AM	130
1	2	11/20/2010 12:00:00 AM	222
1	2	11/27/2010 12:00:00 AM	166
1	2	12/4/2010 12:00:00 AM	174
1	2	12/11/2010 12:00:00 AM	236
1	2	12/18/2010 12:00:00 AM	350
1	2	12/25/2010 12:00:00 AM	216
1	2	1/1/2011 12:00:00 AM	230
1	2	1/8/2011 12:00:00 AM	268
1	2	1/15/2011 12:00:00 AM	332
1	2	1/22/2011 12:00:00 AM	280
1	2	1/29/2011 12:00:00 AM	214
1	2	2/5/2011 12:00:00 AM	220
1	2	2/12/2011 12:00:00 AM	276

Add RDPI Index to data

Add external data to improve model performance

Resulting Dataset

ID1	ID2	Time	Value	RDPI
1	2	9/21/2019 20:07	168	11753.2
1	2	9/14/2019 20:07	126	11753.2
1	2	9/7/2019 20:07	208	11753.2
1	2	8/31/2019 20:07	210	11753.2
1	2	8/24/2019 20:07	160	11696.6
1	2	8/17/2019 20:07	176	11696.6
1	2	8/10/2019 20:07	170	11696.6
1	2	8/3/2019 20:07	212	11696.6
1	2	7/27/2019 20:07	164	11725.6
1	2	7/20/2019 20:07	174	11725.6
1	2	7/13/2019 20:07	238	11725.6
1	2	7/6/2019 20:07	178	11725.6
1	2	6/29/2019 20:07	160	11725.6
1	2	6/22/2019 20:07	144	11713
1	2	6/15/2019 20:07	268	11713
1	2	6/8/2019 20:07	222	11713

Feature Engineering: Create Data Features

@frlazzeri

- Date and Time features: year, month, week of month, etc.
- Season/Holiday features: New Year, U.S. Labor Day, U.S. Black Friday, and Christmas
- Fourier features to capture seasonality
- Lag features: these are values at prior time steps (weeks)

Demo

We are using C# but you can do these steps in any language!

Process Data: Add Weeks to Predict

@frlazzeri

Add default values for the number of weeks to predict to the forecasting data collection.

C#

```
var forcastingDataItem = new ForecastingData
{
        ID1 = storeID1,
        ID2 = itemID2,
        Time = latestDate,
        Value = 0,
        RDPI = latestRdpi,
        DatesInWeek = new List<DateTime>()
};
```

Create Time Features: Extract Concept of Time

@frlazzeri

Date and Time features: year, month, week of month, etc.

C#

```
foreach (var item in forecastData)
{
        item.Year = item.time.Year;
        item.Month = item.time.Month;
        item.WeekOfMonth = Convert.ToInt32(Math.Ceiling(item.time.Day / 7.0));
        item.WeekOfYear = Convert.ToInt32(Math.Ceiling(item.Time.DayOfYear / 7.0));
}
```

Create Holiday Features: Extract Holidays from Time

Season/Holiday features: New Year, U.S. Labor Day, U.S. Black Friday, and Christmas.

C#

```
// 4th Friday in November
item.IsBlackFriday = item.DatesInWeek.Any(date => date.Month == 11 && date.DayOfWeek == DayOfWeek.Friday &&
date.Day > 22 && date.Day < 29);
// 1st Monday in September
item.IsUsLaborDay = item.DatesInWeek.Any(date => date.Month == 9 && date.DayOfWeek == DayOfWeek.Monday &&
date.Day < 8);
// 25th of December
item.IsChristmasDay = item.DatesInWeek.Any(date => date.Month == 12 && date.Day == 25);
// 1st of January
item.IsUsNewYearsDay = item.DatesInWeek.Any(date => date.Month == 1 && date.Day == 1);
```

Create Fourier Features: Capture Up/Down Pattern

Fourier features to capture seasonality

C#

```
var seasonality = 52;
foreach (var item in forecastData)
{
        item.FreqCos1 = Math.Cos(item.WeekOfYear * 2 * Math.PI * 1 / seasonality);
        item.FreqSin1 = Math.Sin(item.WeekOfYear * 2 * Math.PI * 1 / seasonality);
        ...
        item.FreqCos4 = Math.Cos(item.WeekOfYear * 2 * Math.PI * 4 / seasonality);
        item.FreqSin4 = Math.Sin(item.WeekOfYear * 2 * Math.PI * 4 / seasonality);
}
```

@frlazzeri

Create Lag Features: Capture Prior Weeks to Current

Lag features: these are values at prior time steps

C#

```
for (int i = forecastData.Count-1; i >= forecastData.Count; i--)
{
        forecastData[i].Lag1 = forecastData[i - 1].value;
        forecastData[i].Lag2 = forecastData[i - 2].value;
        ...
        forecastData[i].Lag26 = forecastData[i - 26].value;
}
```

Resulting Dataset:

@frlazzeri

ID1	ID2	time	value	RDPI	year	month	weekofmonth	weekofyear	USNewYearsDay	USLaborDay	USThanksgivingDay	...	FreqCos1	FreqSin1	FreqCos2	FreqSin2	...	lag1	lag2	lag3
1	2	2013-01-05T00:00:00Z	5.308268	11638.5	2013	1	1	1	True	False	False		0.464723	0.885456	-0.568065	0.822984		4.941642	5.123964	5.023881
1	2	2013-01-12T00:00:00Z	5.365976	11709.1	2013	1	2	2	False	False	False		0.354605	0.935016	-0.748511	0.663123		4.941642	5.123964	5.023881
1	2	2013-01-19T00:00:00Z	5.214936	11709.1	2013	1	3	3	False	False	False		0.239316	0.970942	-0.885456	0.464723		4.941642	5.123964	5.023881
1	2	2013-01-26T00:00:00Z	5.049856	11709.1	2013	1	4	4	False	False	False		0.120537	0.992709	-0.970942	0.239316		4.941642	5.123964	5.023881
1	2	2013-02-02T00:00:00Z	4.912655	11709.1	2013	2	1	5	False	False	False		0	1	-1	0		4.941642	5.123964	5.023881
1	2	2013-02-09T00:00:00Z	5.337538	11877.2	2013	2	2	6	False	False	False		-0.120537	0.992709	-0.970942	-0.239316		4.941642	5.123964	5.023881
1	2	2013-02-16T00:00:00Z	5.298317	11877.2	2013	2	3	7	False	False	False		-0.239316	0.970942	-0.885456	-0.464723		4.941642	5.123964	5.023881
1	2	2013-02-23T00:00:00Z	5.214936	11877.2	2013	2	4	8	False	False	False		-0.354605	0.935016	-0.748511	-0.663123		4.941642	5.123964	5.023881
1	2	2013-03-02T00:00:00Z	5.087596	11877.2	2013	3	1	9	False	False	False		-0.464723	0.885456	-0.568065	-0.822984		4.941642	5.123964	5.023881
1	2	2013-03-09T00:00:00Z	5.384495	12214.1	2013	3	2	10	False	False	False		-0.568065	0.822984	-0.354605	-0.935016		4.941642	5.123964	5.023881

Resulting Dataset:

@frlazzeri

ID1	ID2	time	value	RDPI	year	month	weekofmonth	weekofy
1	2	2013-01-05T00:00:00Z	5.308268	11638.5	2013	1	1	1
1	2	2013-01-12T00:00:00Z	5.365976	11709.1	2013	1	2	2
1	2	2013-01-19T00:00:00Z	5.214936	11709.1	2013	1	3	3
1	2	2013-01-26T00:00:00Z	5.049856	11709.1	2013	1	4	4
1	2	2013-02-02T00:00:00Z	4.912655	11709.1	2013	2	1	5
1	2	2013-02-09T00:00:00Z	5.337538	11877.2	2013	2	2	6
1	2	2013-02-16T00:00:00Z	5.298317	11877.2	2013	2	3	7
1	2	2013-02-23T00:00:00Z	5.214936	11877.2	2013	2	4	8
1	2	2013-03-02T00:00:00Z	5.087596	11877.2	2013	3	1	9
1	2	2013-03-09T00:00:00Z	5.384495	12214.1	2013	3	2	10

Modeling

Selecting the “Right” Algorithm to Train Your Model

Azure Machine Learning

@frlazzeri

Authoring tools: Automated ML, Azure Machine designer, Notebooks

Assets: Datasets, Experiments, ML Workflow Pipelines, Models, Deployments

Management: Compute, Datastores, Workspaces

Deploying

Operationalize Your Model

@frlazzeri

Deployment with Azure Machine Learning

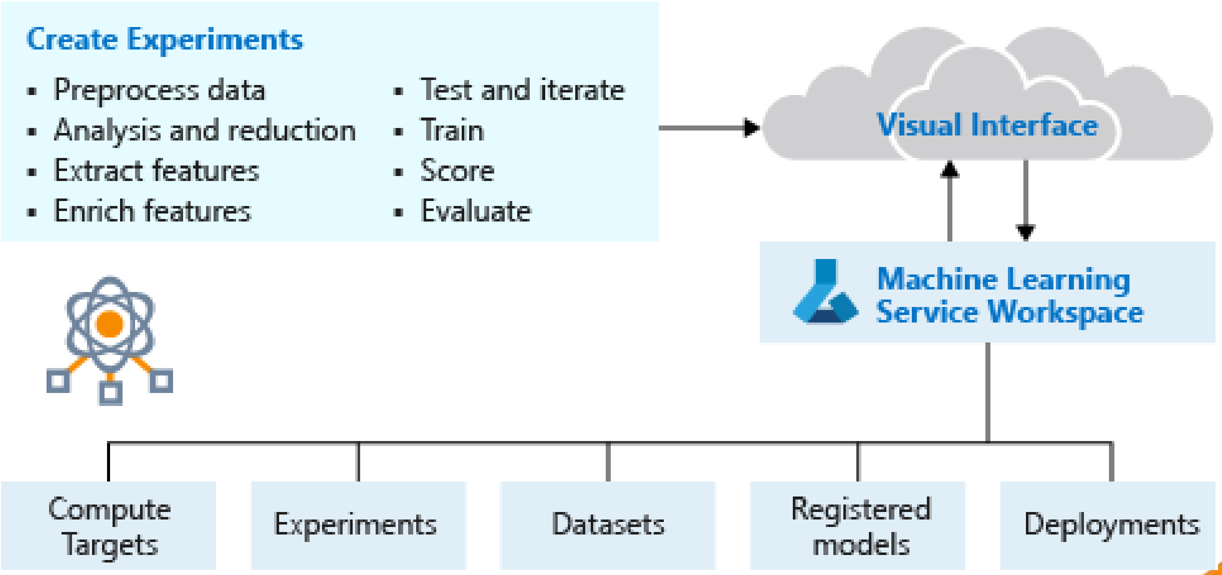
Making models available to external customers and/or other teams and stakeholders in your company.

Other teams in your company can use them, send data to them and get their predictions, which are in turn populated back into the company systems to increase training data quality and quantity.

Companies will start building and deploying higher numbers of machine learning models in production and master robust and repeatable ways to move models from development environments into business operations systems.

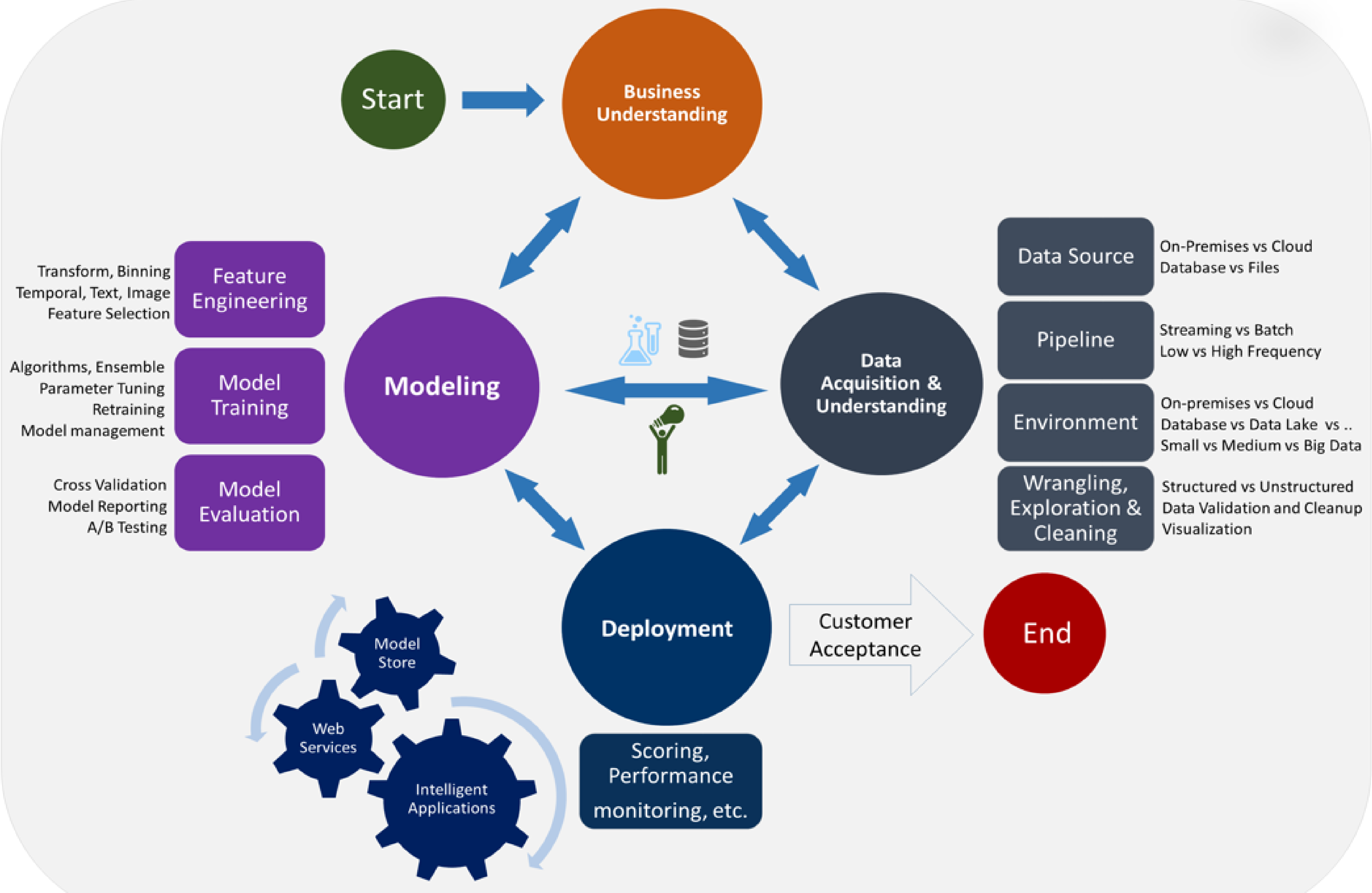
@frlazzeri

Demo: Build, Test and Deploy Your Model



The Machine Learning Process

@frlazzeri



Useful Resources:

@frlazzeri

- ✓ AzureML GitHub: aka.ms/AzureMLrepo
- ✓ Algorithm Cheat Sheet: aka.ms/AlgorithmCheatSheet
- ✓ Deep Learning VS Machine Learning: aka.ms/DeepLearningVSMachineLearning
- ✓ Automated Machine Learning Documentation: aka.ms/AutomatedMLDocs
- ✓ Model Interpretability with Azure ML Service: aka.ms/AzureMLModelInterpretability
- ✓ Azure Machine Learning Service: aka.ms/AzureMLservice
- ✓ Azure Machine Learning Designer: aka.ms/AzureMLdesigner
- ✓ Get started with Azure ML: aka.ms/GetStartedAzureML
- ✓ Azure Notebooks: aka.ms/AzureNB

Coming next…

@frlazzeri

ODSC East • Boston

April 16th, 2020 • 14:00 PM Boston Hynes • Convention Center

Training and Operationalizing Interpretable Machine Learning Models

https://opendatascience.com/training-and-operationalizing-interpretable-machine-learning-models/

Training and Operationalizing Interpretable Machine Learning Models

Abstract:

AI offers companies the possibility to transform their operations: from AI applications able to predict and schedule equipment's maintenance, to intelligent R&D applications able to estimate the success of future drugs, until HR AI-powered tools able to enhance the hiring process and employee retention strategy. However, in order to be able to leverage this opportunity, companies have to learn how to successfully build, train, test, and push hundreds of machine learning models in production, and to move models from development to their production environment in ways that are robust, explainable, and repeatable.

Nowadays data scientists and developers have a much easier experience when building AI-based solutions through the availability and accessibility of data and open-source machine learning frameworks. However, this process becomes a lot more complex when they need to think about model deployment and pick the best strategy to scale up to a production-grade system.

FRANCESCA LAZZERI, PHD

Thank You

Francesca Lazzeri, PhD – Senior Machine Learning Scientist Lead

@frlazzeri
aka.ms/AzureMLrepo
medium.com/@francescalazzeri

Microsoft

HDSI | Harvard Data Science Initiative