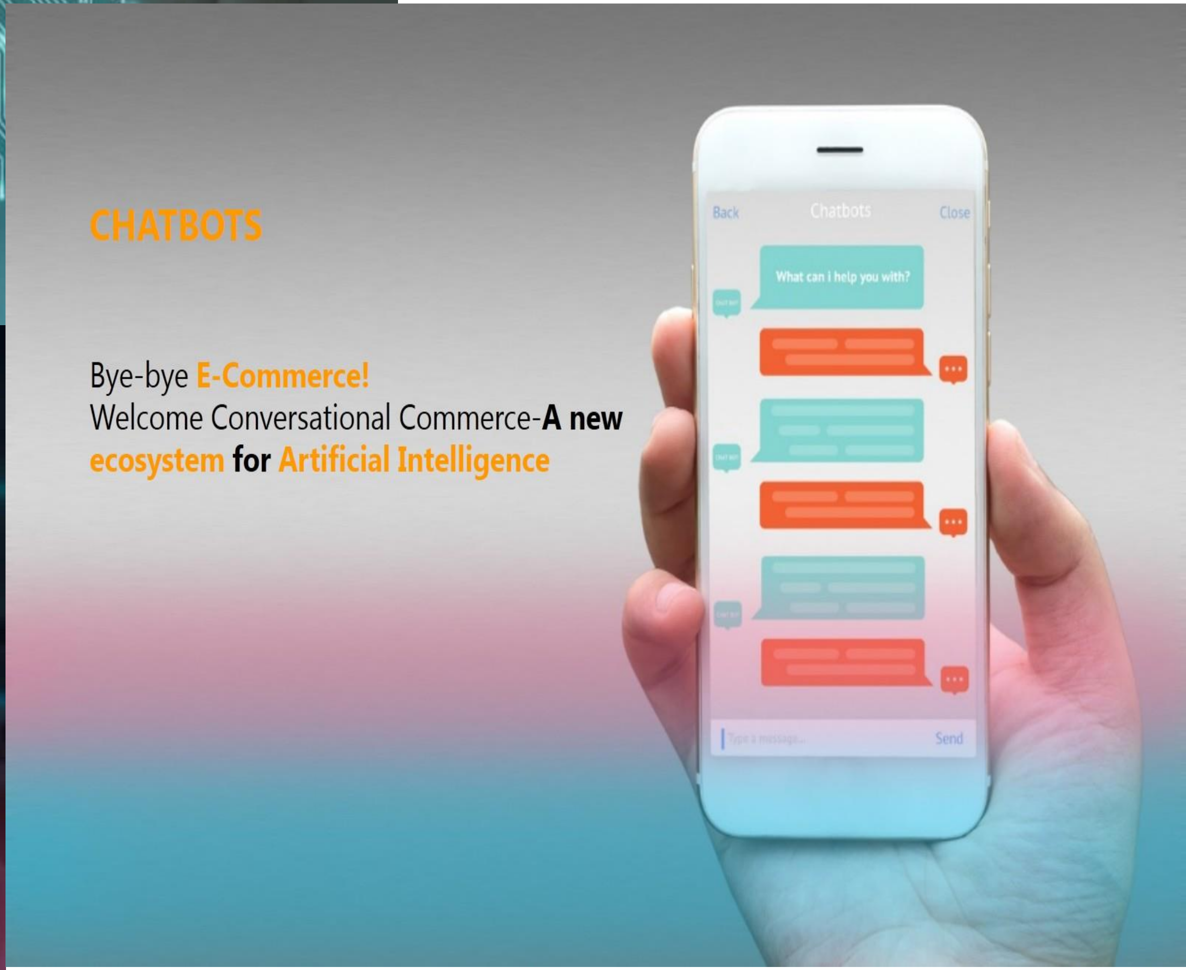
Microsoft

Azure Machine Learning

Level – 300.

Srinag

A Boutique Company that provides Training and Consulting services on Data + AI + IoT Workloads



JUMP START NINJA
THE GROWTH HACKER FOR YOUR BUSINESS

www.jumpstartninja.com

About Me

Advanced Analytics Specialist.

Certified from MIT on 'Advanced Analytics and Big Data Challenges'. Technology Enthusiast and Data Specialist.

Certifications

- Microsoft Certified Power BI Developer.
- Microsoft Certified Azure Machine Learning Developer from Microsoft Virtual Academy
- Certified on Practical Data Analytics with Microsoft Cortana Intelligence Suite.

Sri Nag Sashank

Experience

7.5 years of total experience as Data Specialist and Data Engineer.

Role

Currently working as Azure Data Specialist and Enterprise Trainer on Azure Data Analytics , Cortana Intelligence Suite and Cognitive Sciences

Training

Expert **facilitator and corporate behavioral trainer** with 5 years of experience in designing programs, content development , Data Analytics and Machine Learning.

- Hosted many webinars and corporate workshops on Azure Data Analytics & Machine Learning.

Agenda

- Microsoft Cortana Intelligence Suite
- Introduction to Machine Learning
- Data Preparation in Azure ML
- Classification
- Confusion Matrix
- Ensemble Learning Introduction
- Support Vector Machines
- Hyper Parameter Tuning

Microsoft

Cortana Intelligence Suite

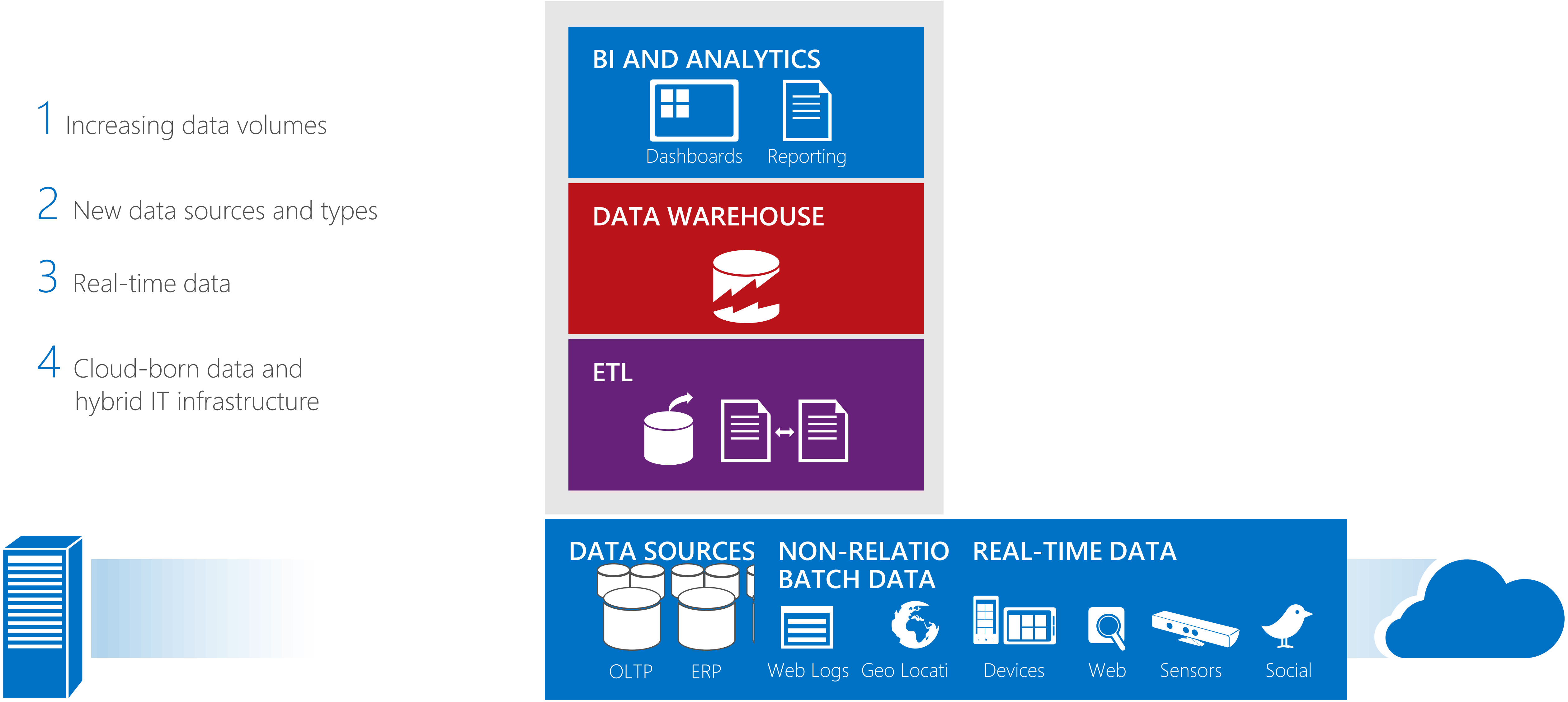
Microsoft Vision on Data

1 Increasing data volumes

2 New data sources and types

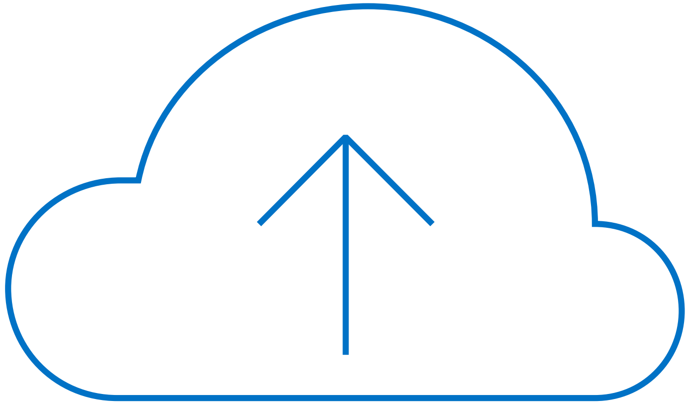
3 Real-time data

4 Cloud-born data and hybrid IT infrastructure

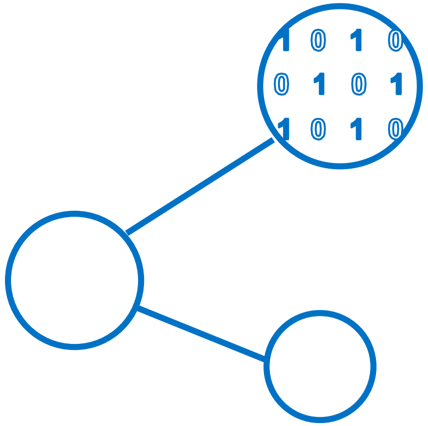


Insights is a journey

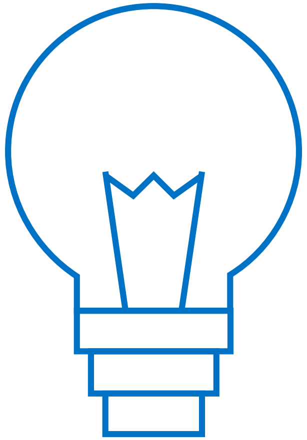
Convergence accelerates digital transformation



Cloud

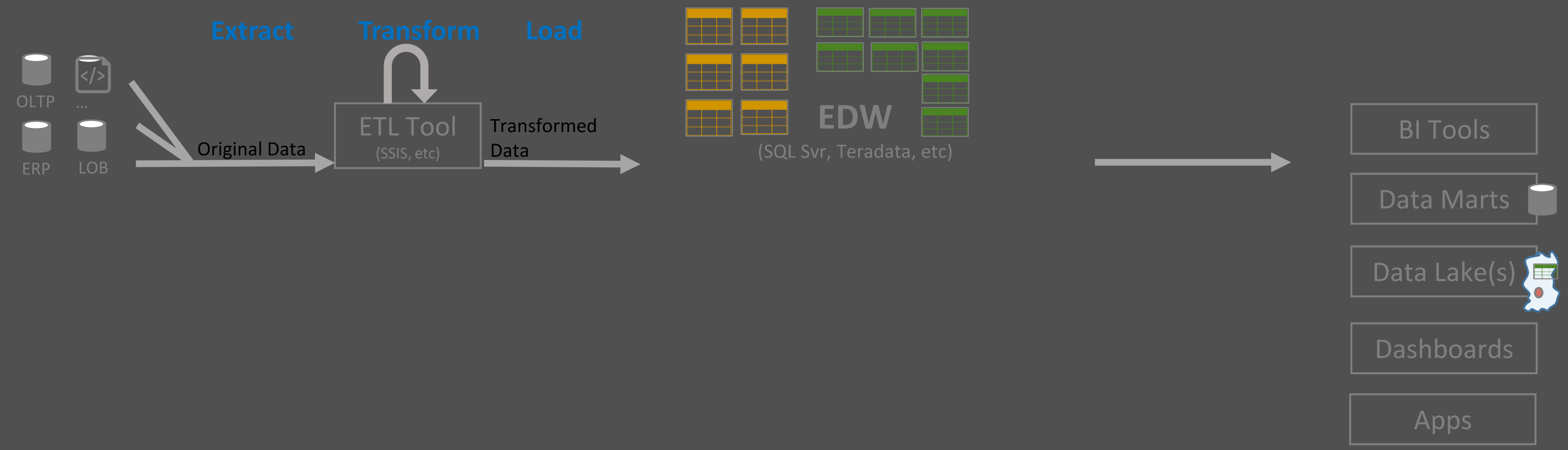


Data



Intelligence

Evolving Approaches to Analytics
EDWs to Data Lakes

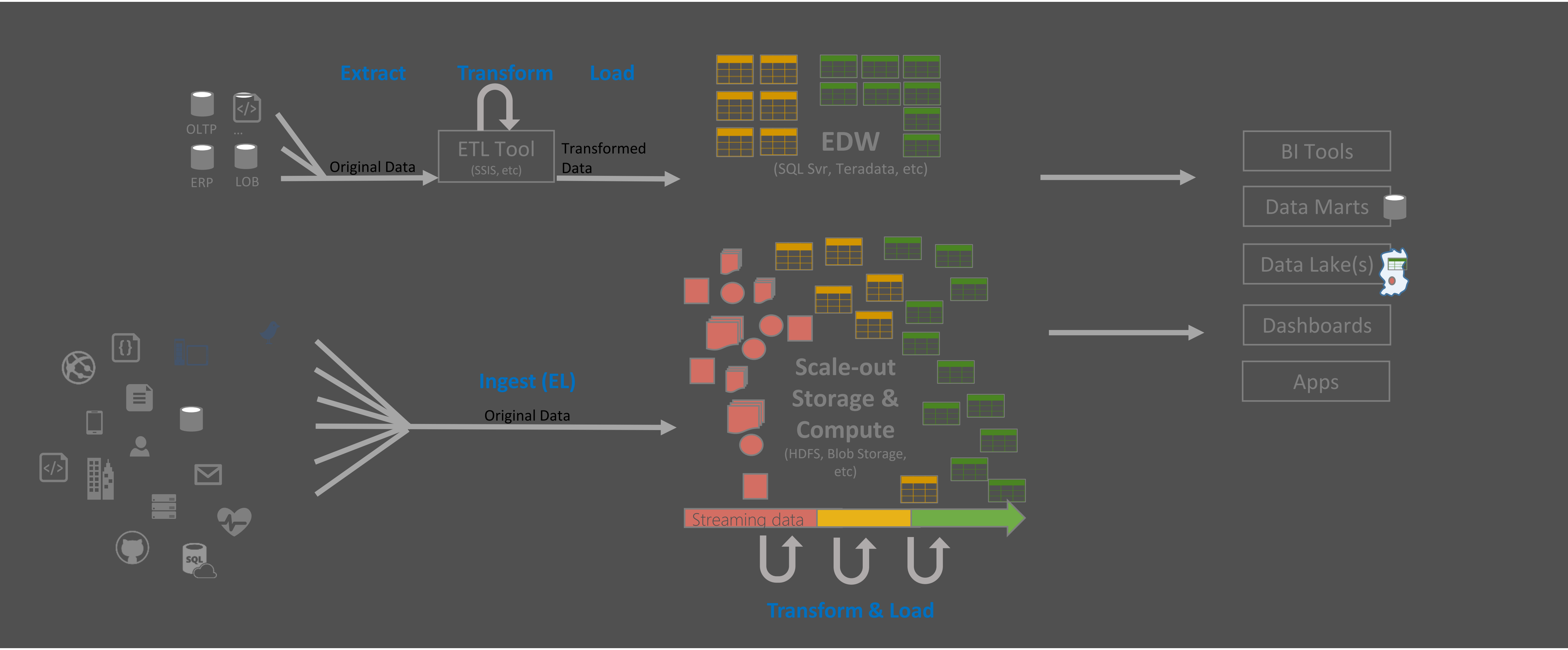


Evolving Approaches to Analytics

EDWs to Data Lakes

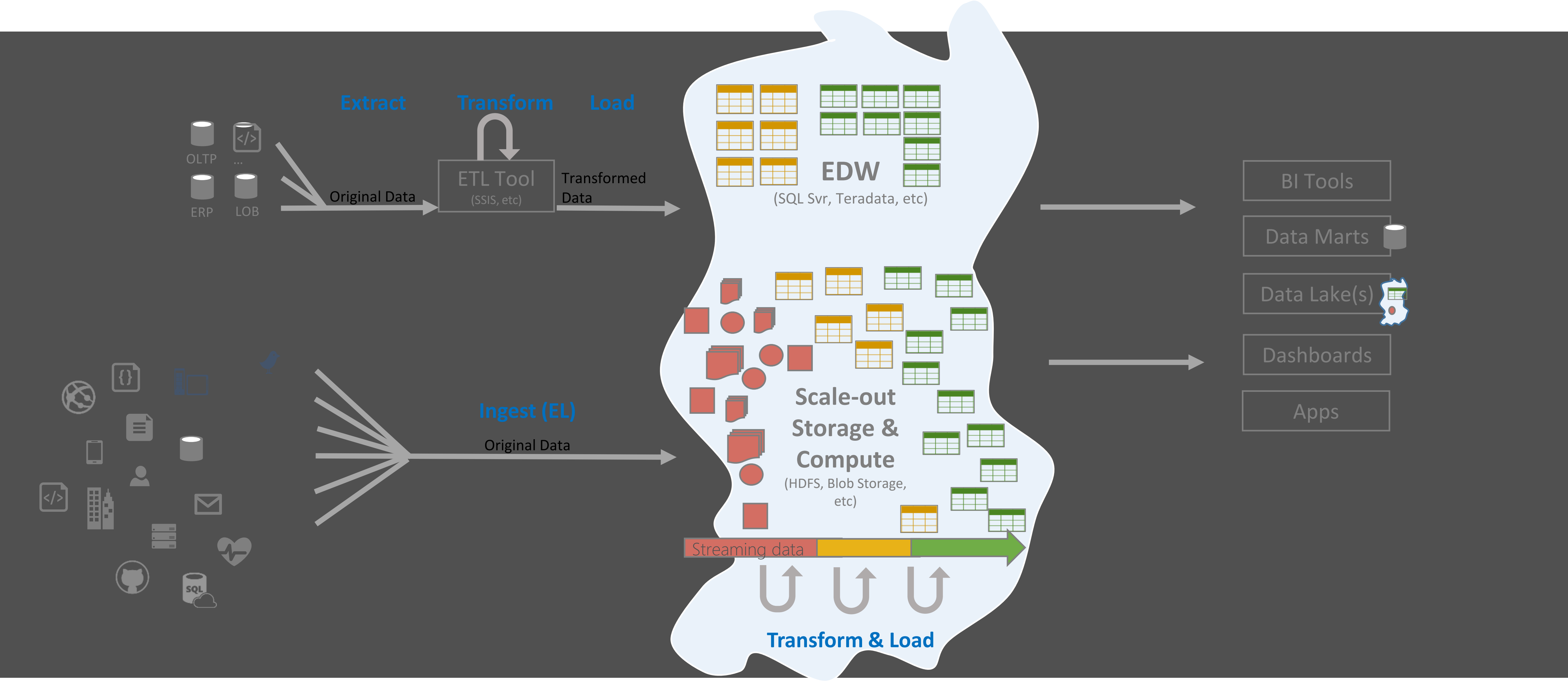
Evolving Approaches to Analytics

EDWs to Data Lakes

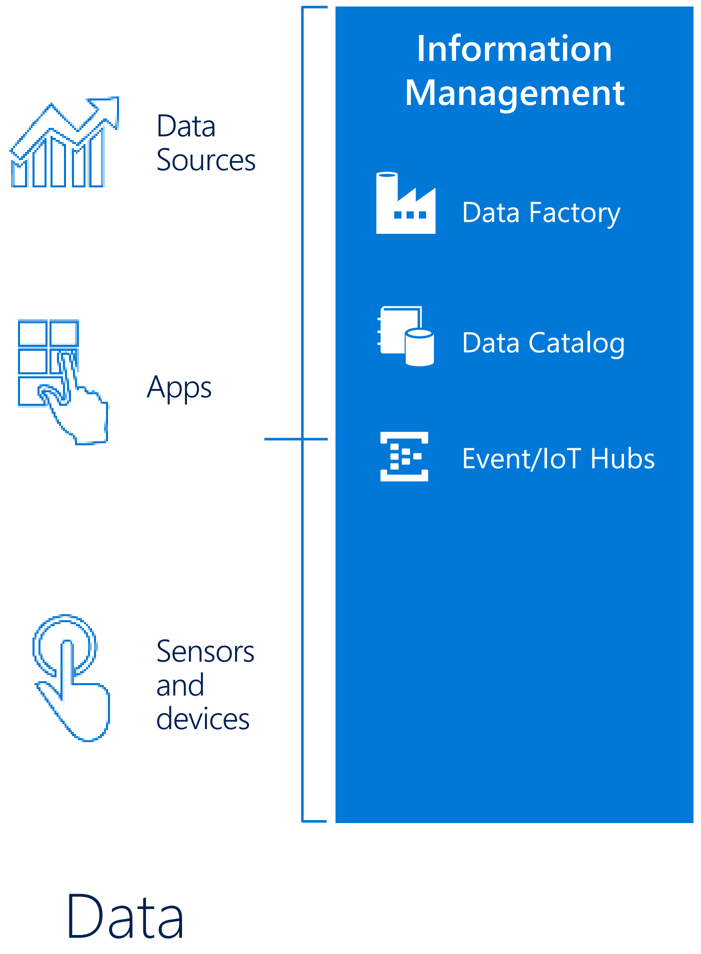


Evolving Approaches to Analytics

EDWs to Data Lakes



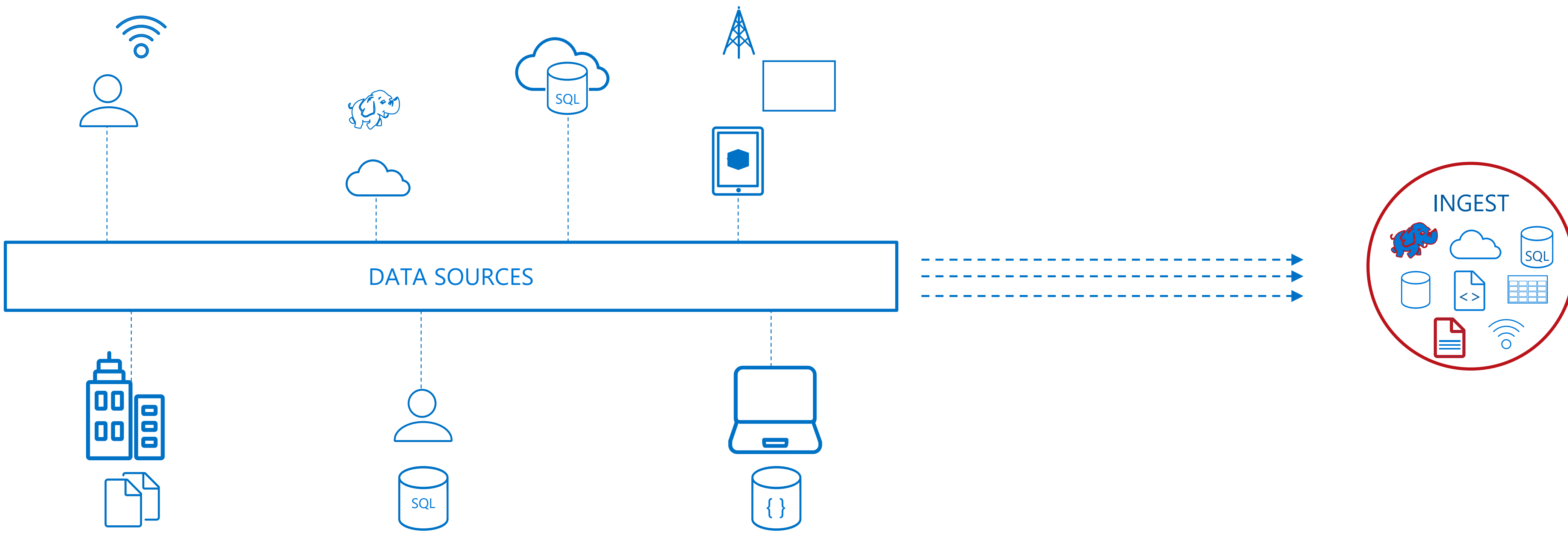
Information Management



Compose and orchestrate data services at scale

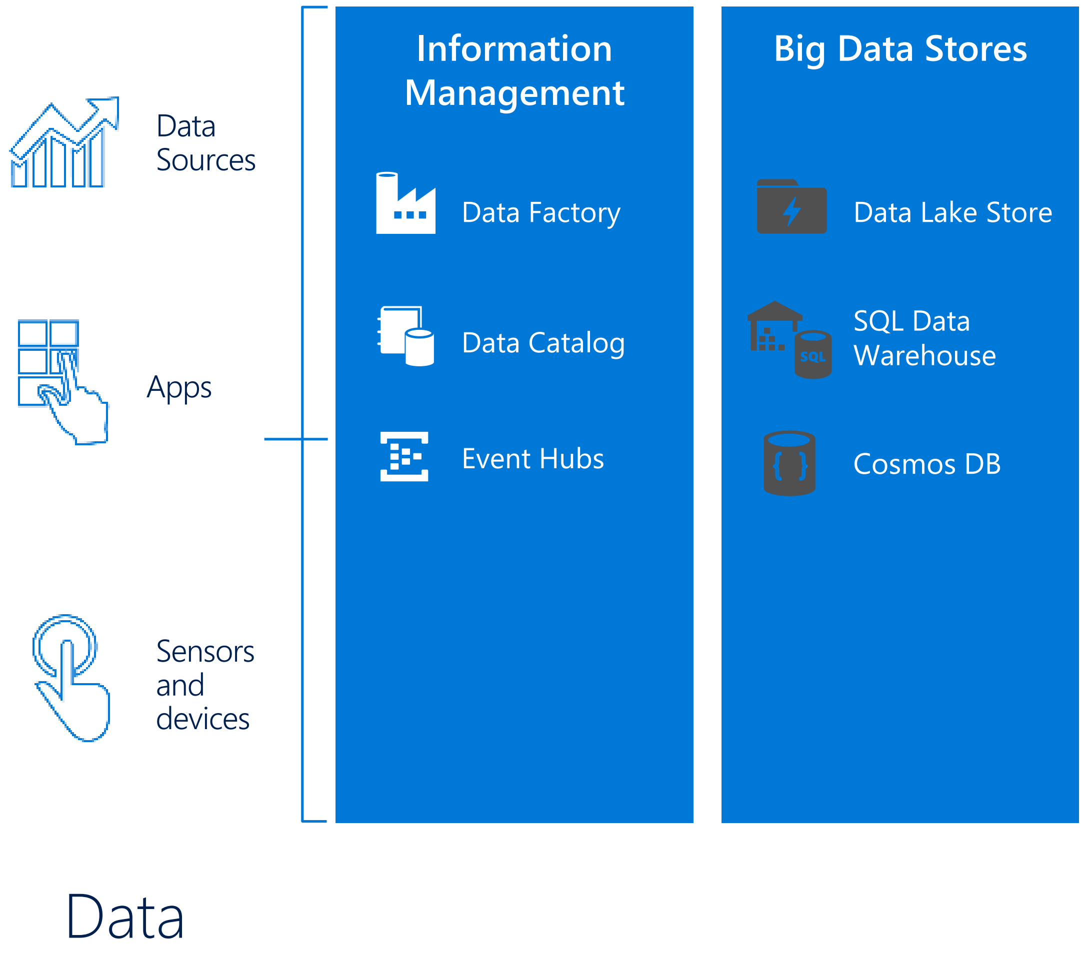
Information Management

- Data Factory
- Data Catalog
- Event /IoT Hubs

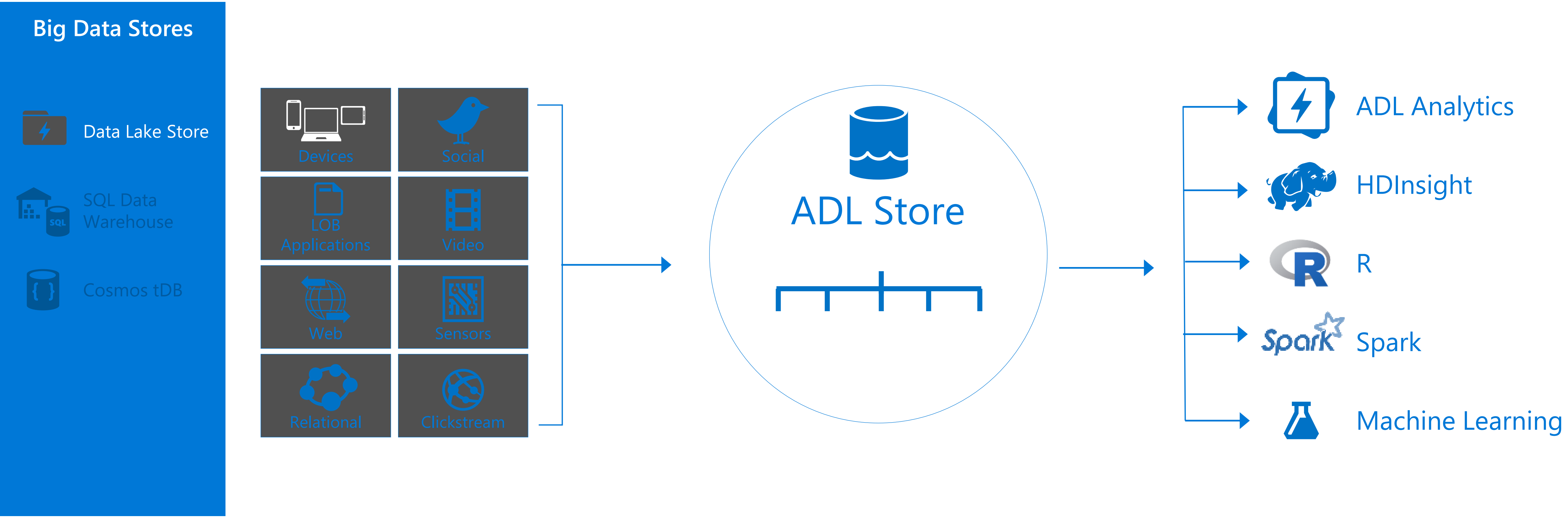


- Create, schedule, orchestrate, and manage data pipelines
- Visualize data lineage
- Connect to on-premises and cloud data sources
- Monitor data pipeline health
- Automate cloud resource management
- Move relational data for Hadoop processing
- Transform with Hive, pig, or custom code

Big Data Stores

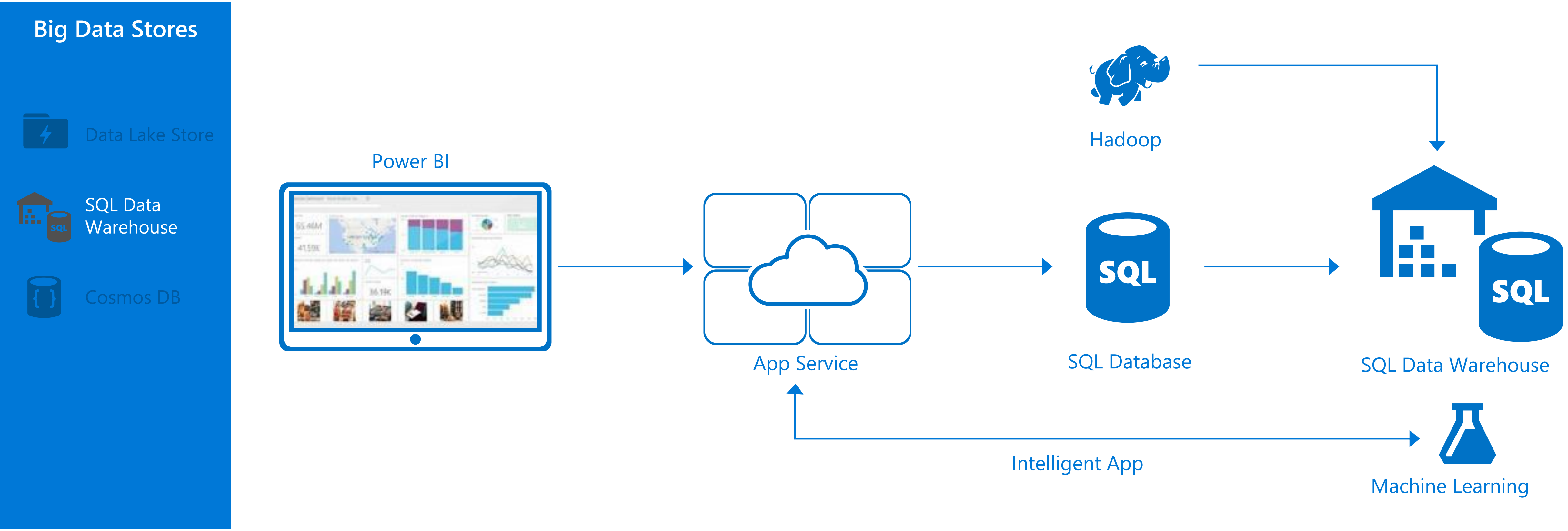


A hyper-scale repository for big data analytics workloads



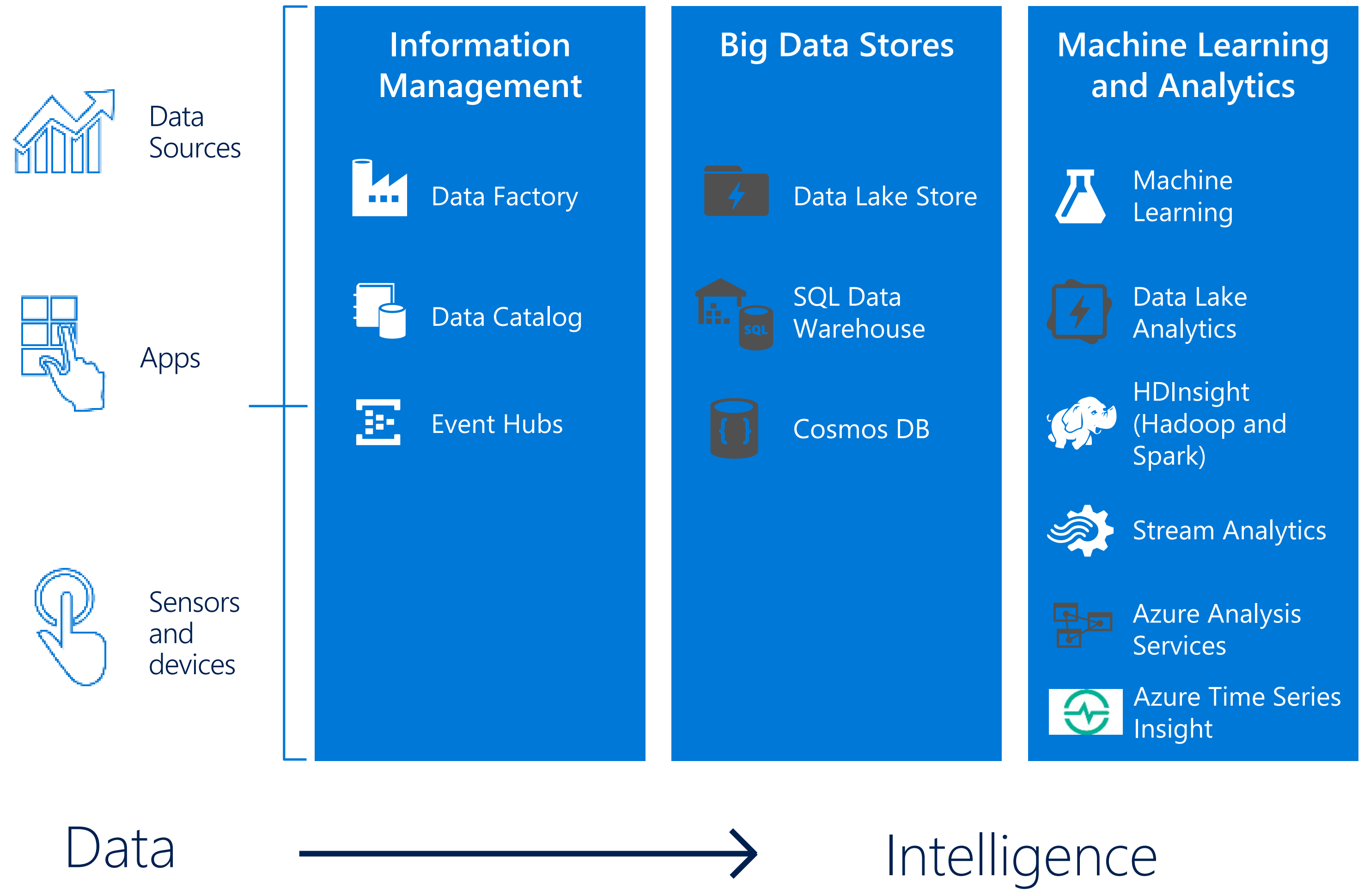
- A Hadoop Distributed File System for the cloud
- No fixed limits on file size
- No fixed limits on account size
- Unstructured and structured data in their native format
- Massive throughput to increase analytic performance
- High durability, availability, and reliability
- Azure Active Directory access control

Elastic data warehouse as a service with enterprise-class features



- Petabyte scale with massively parallel processing
- Independent scaling of compute and storage—in seconds
- Transact-SQL queries across relational and non-relational data
- Full enterprise-class SQL Server experience
- Works seamlessly with Power BI, Machine Learning, HDInsight, and Data Factory

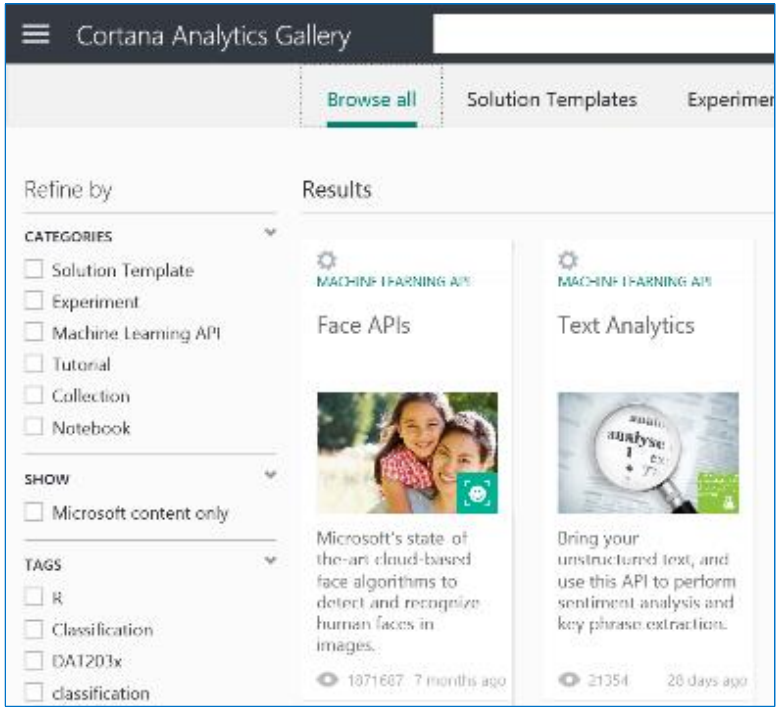
Machine Learning and Analytics

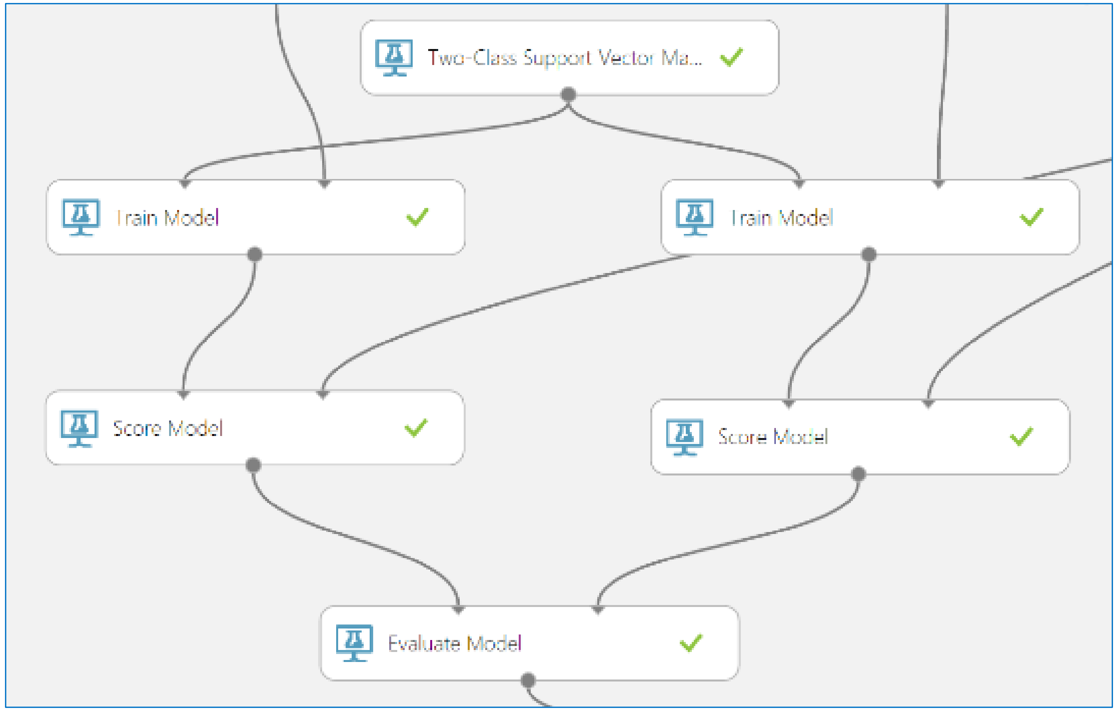


Easily build, deploy, and share predictive analytics solutions

Machine Learning and Analytics

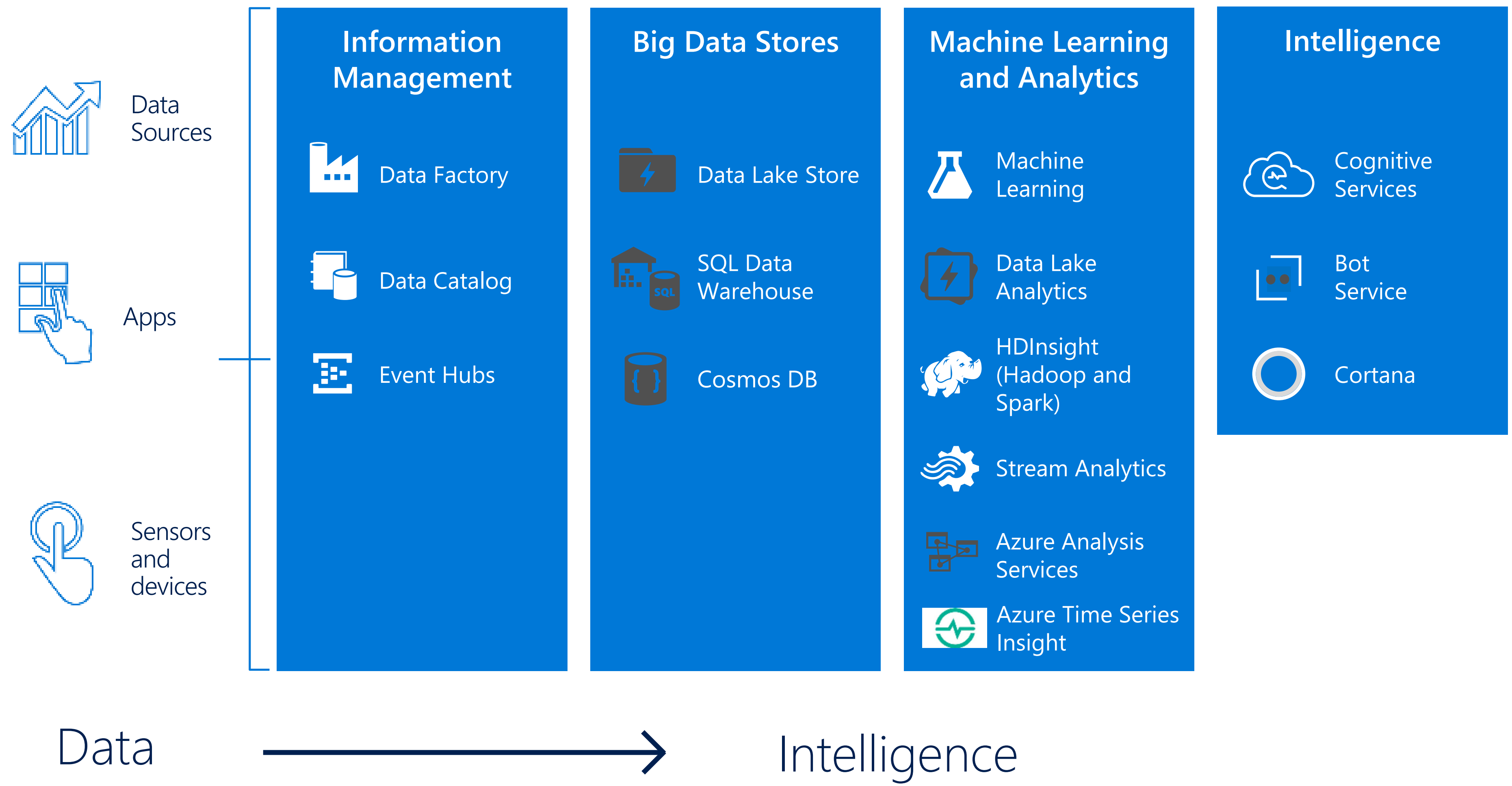
- Machine Learning
- Data Lake Analytics
- HDInsight (Hadoop and Spark)
- Stream Analytics
- Azure Analysis Services
- Azure Time Series Insight





- Simple, scalable, cutting edge. A fully managed cloud service that enables you to easily build, deploy, and share predictive analytics solutions.
- Deploy in minutes. Azure Machine Learning means business. You can deploy your model into production as a web service that can be called from any device, anywhere and that can use any data source.
- Publish, share, monetize. Share your solution with the world in the Gallery or on the Azure Marketplace.

Intelligence



Build applications that understand people

Intelligence

- Cognitive Services
- Bot Service
- Cortana

Vision	Speech	Language	Knowledge	Search
Computer Vision	Speaker Recognition	Text Analytics	Academic	Bing Web Search API
Content Moderator	Bing Speech	Bing Spell Check	Entity Linking	Bing Image Search API
Face	Custom Speech Service	Web Language Model	Knowledge Exploration	Bing Video Search API
Emotion		Linguistic Analysis	Recommendations	Bing News Search API
Video		Language Understanding	QnA Maker	Bing Auto Suggest API
		Translator		

- Faces, images, emotion recognition and video intelligence
- Spoken language processing, speaker recognition, custom speech recognition
- Natural language processing, sentiment and topics analysis, spelling errors
- Complex tasks processing, knowledge exploration, intelligent recommendations
- Bing engine capabilities for Web, Autosuggest, Image, Video and News

Get things done in more helpful, proactive and natural ways

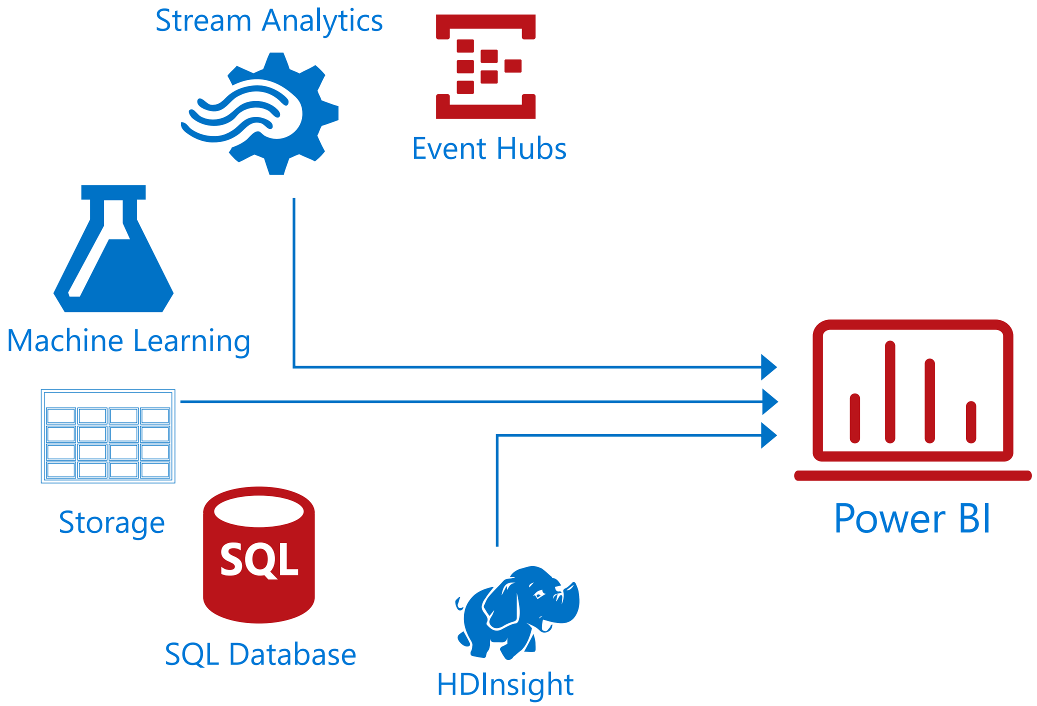
Intelligence

- Cognitive Services
- Bot Service
- Cortana

Here are some of the things I can help you with…	Cortana for Consumers (today)	With the Cortana Intelligence
Answers	Public reference data answers – *"How far is it from Los Angeles to San Francisco?"*	Answers from organizational data in Power BI *"What were our biggest deals that closed last month?"*
Predictions	Event predictions – *"Who do you think is going to win the Germany Italy game?"*	Integration with prediction solutions *"Which of our customers are most likely to churn in the next quarter?"*
Monitoring & Alerts	Flight status, traffic conditions, changes in weather, …	Monitoring KPIs and preemptive alerting *"Alert me if this customer ever has a 90% chance of churn in the next 30 days"*
Task Completion	Setting reminders, scheduling meetings, getting directions, …	Line of business process integration *Assistance with expense report submission on-time within policy*

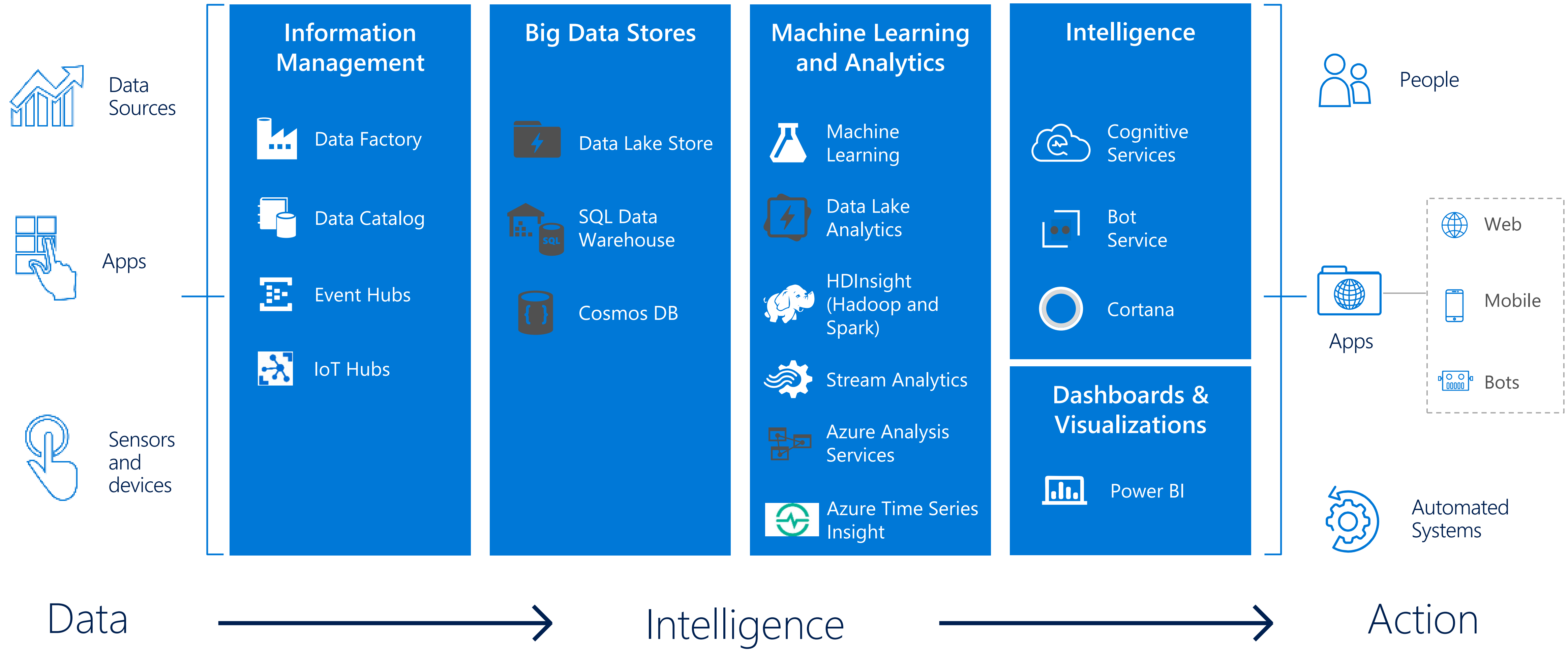
Dashboards & Visualizations

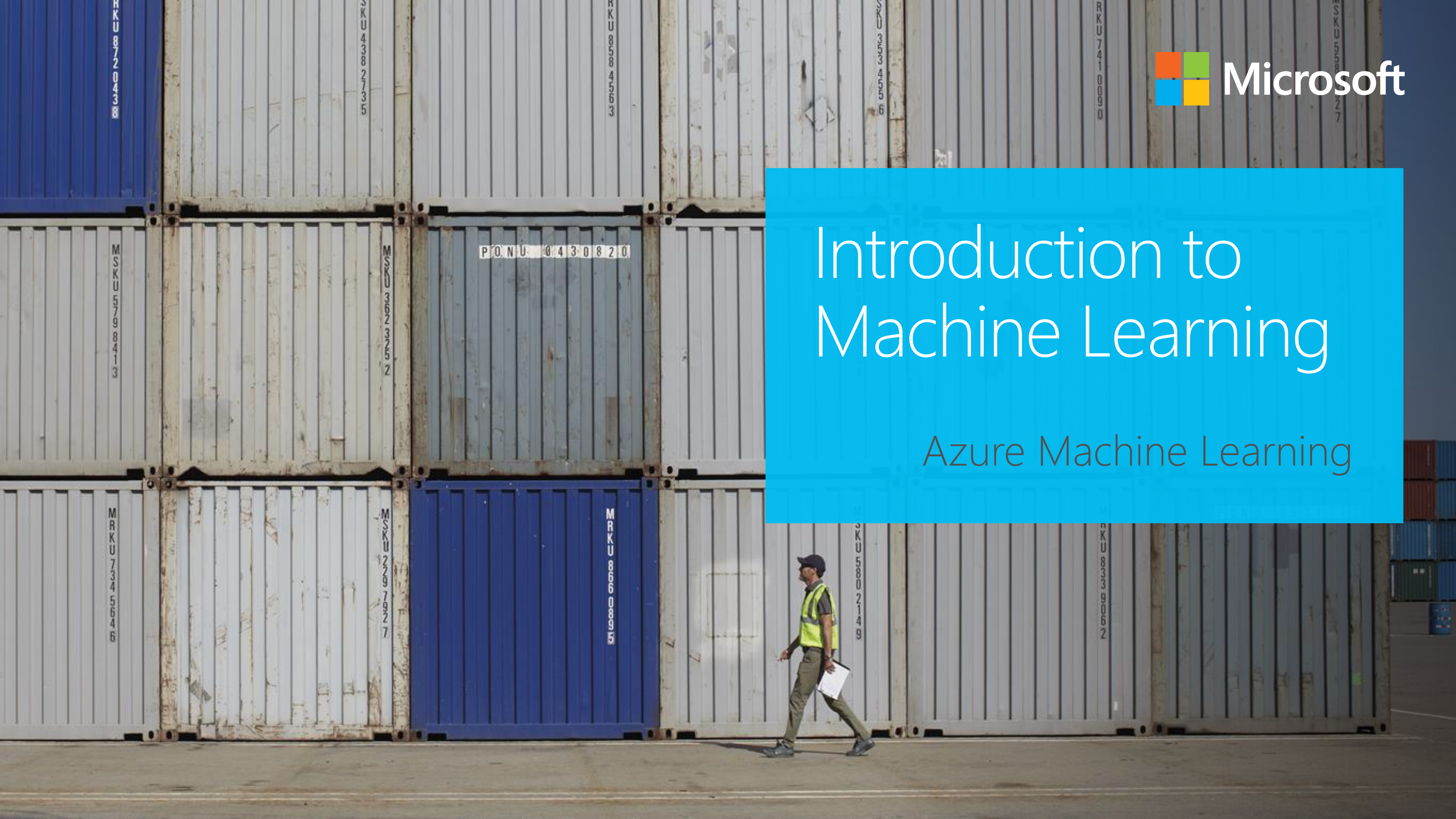
Power BI



- Analytics for everyone, even non-data experts
- Your whole business on one dashboard
- Create stunning, interactive reports
- Drive consistent analysis across your organization
- Embed visuals in your applications
- Get real-time alerts when things change

Transform data into intelligent action





Microsoft

Introduction to Machine Learning

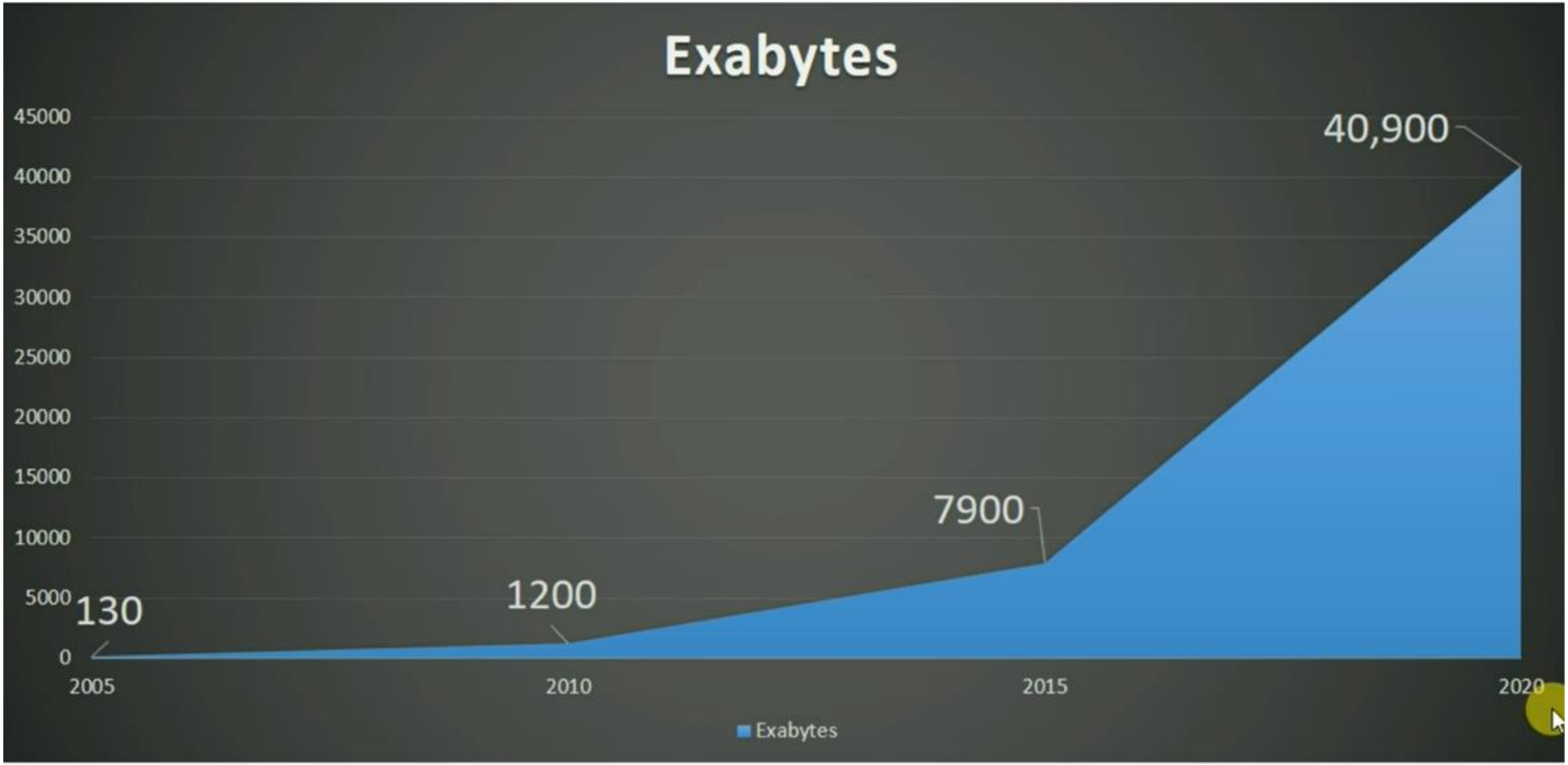
Azure Machine Learning

Why Machine Learning is the Future?

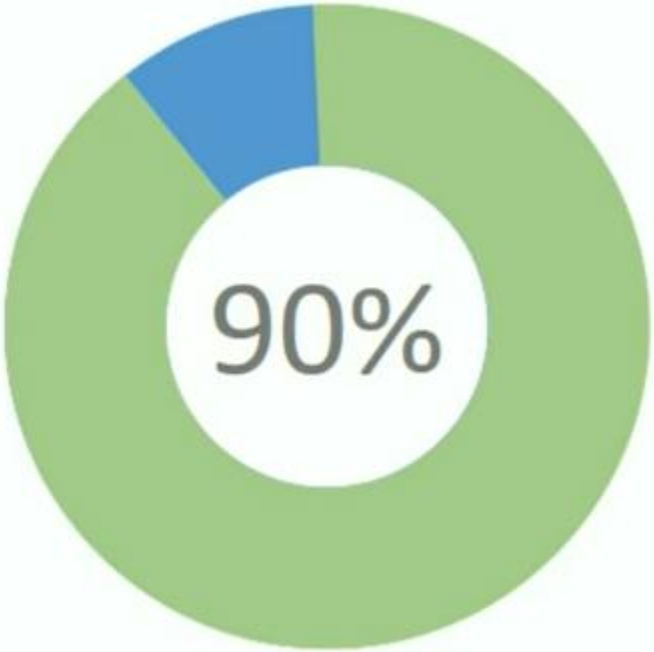
Measurement of Data

1 TB	=	1000 GB
1 PB	=	1000 TB
1 EB	=	1000 PB

Growth of Data

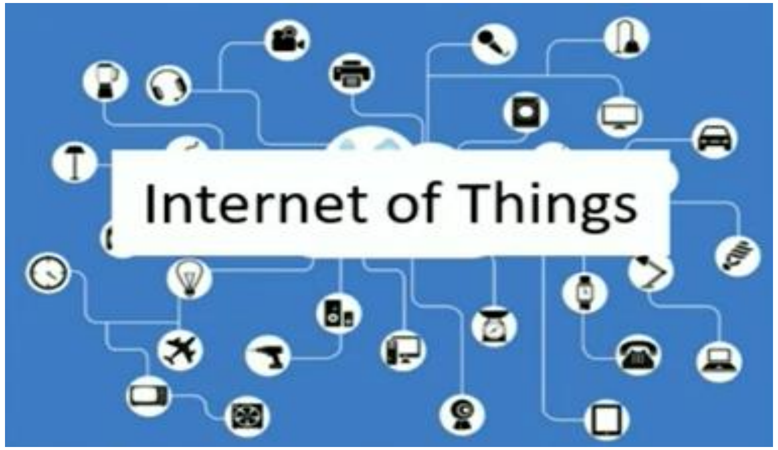


Statistics on Data Growth



90% of today's data has been created in last two years alone

Statistics on Data Growth



Heard on the Streets

- IDC Futurescape - Two-thirds of Global 2000 Enterprises CEOs will centre their corporate strategy on digital transformation including machine learning (ML) solutions.

- Harvard Business Review - Data Scientist: The Sexiest Job of the 21st Century

- McKinsey Report - 45 percent of work activities could potentially be automated by currently demonstrated technologies; machine learning can be an enabling technology for the automation of 80 percent of those activities.

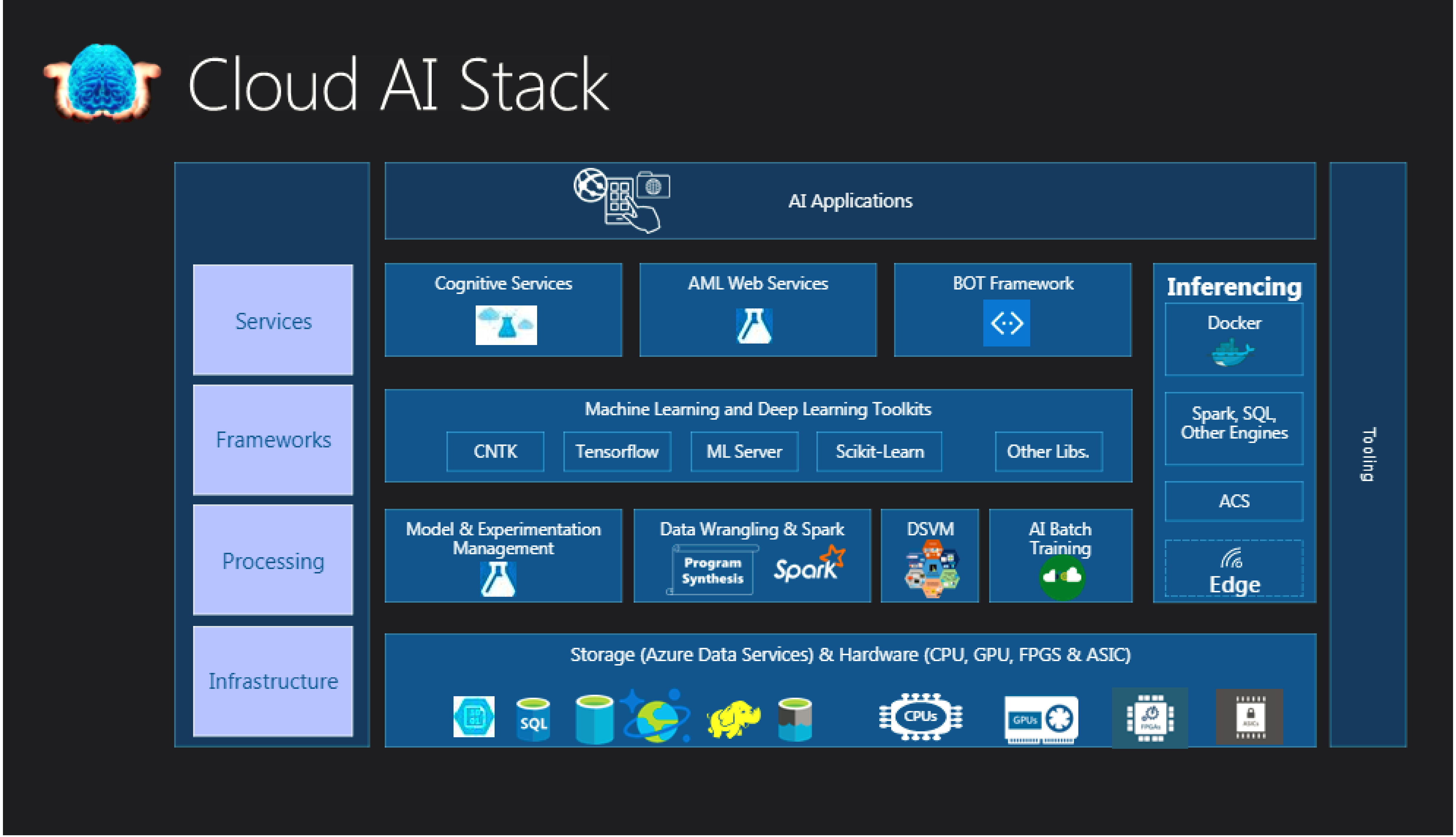
- Microsoft CEO Satya Nadella - called out machine learning and the big data that powers it as a key development in his memo to Microsoft last July.

Agenda – Day1

1. Introduction to Machine Learning
2. How can we leverage Azure Machine Learning
3. Data Preparation in AML Studio
4. Classification
5. Regression
6. Hyper tuning Parameters in AML Studio
7. Deployment of Models using AML Studio.

Agenda – Day2

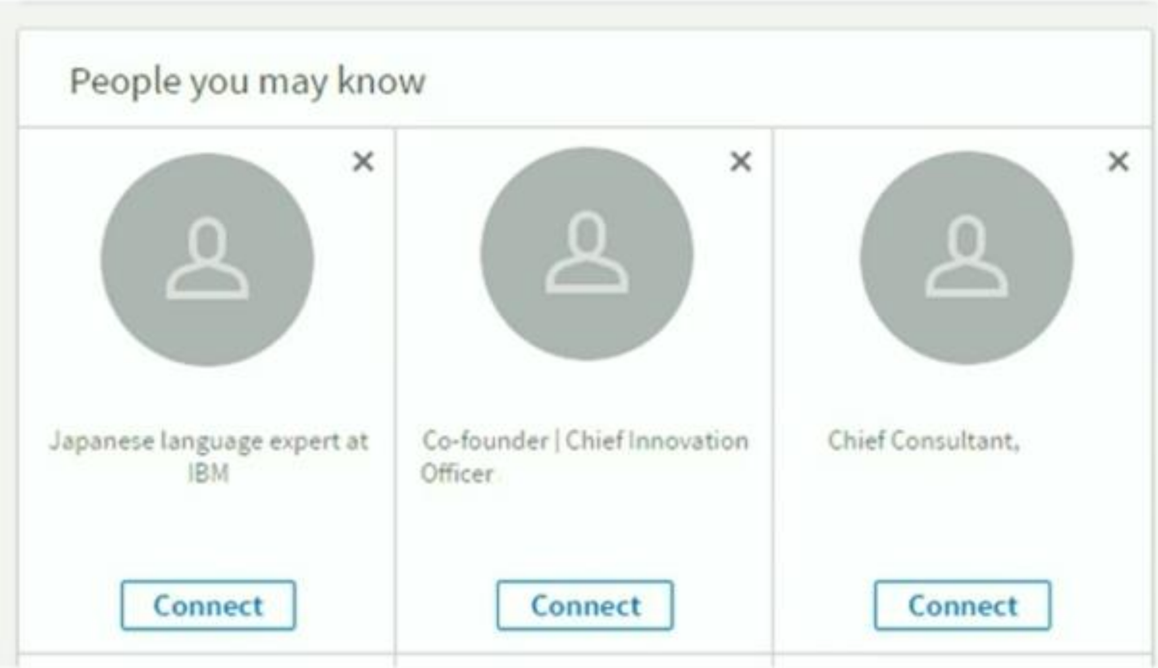
1. Clustering
2. Recommendation Engine in AML Studio
3. Azure Machine Learning Services
4. Deep Learning & Image Classification
5. AML Work Bench Advanced Data Preparation
6. Building Models in AML WB
7. Deploying Locally & in Dockers.
8. Creating Webservices in Model Management.
9. Demo on Image Classification in AML Workbench.


Cloud AI Stack
Services
Frameworks
Processing
Infrastructure
AI Applications
Cognitive Services
AML Web Services
BOT Framework
Inferencing
Docker
Spark, SQL, Other Engines
ACS
Edge
Machine Learning and Deep Learning Toolkits
CNTK
Tensorflow
ML Server
Scikit-Learn
Other Libs.
Model & Experimentation Management
Data Wrangling & Spark
Program Synthesis
Spark
DSVM
AI Batch Training
Storage (Azure Data Services) & Hardware (CPU, GPU, FPGS & ASIC)
SQL
CPUs
GPUs
Tooling

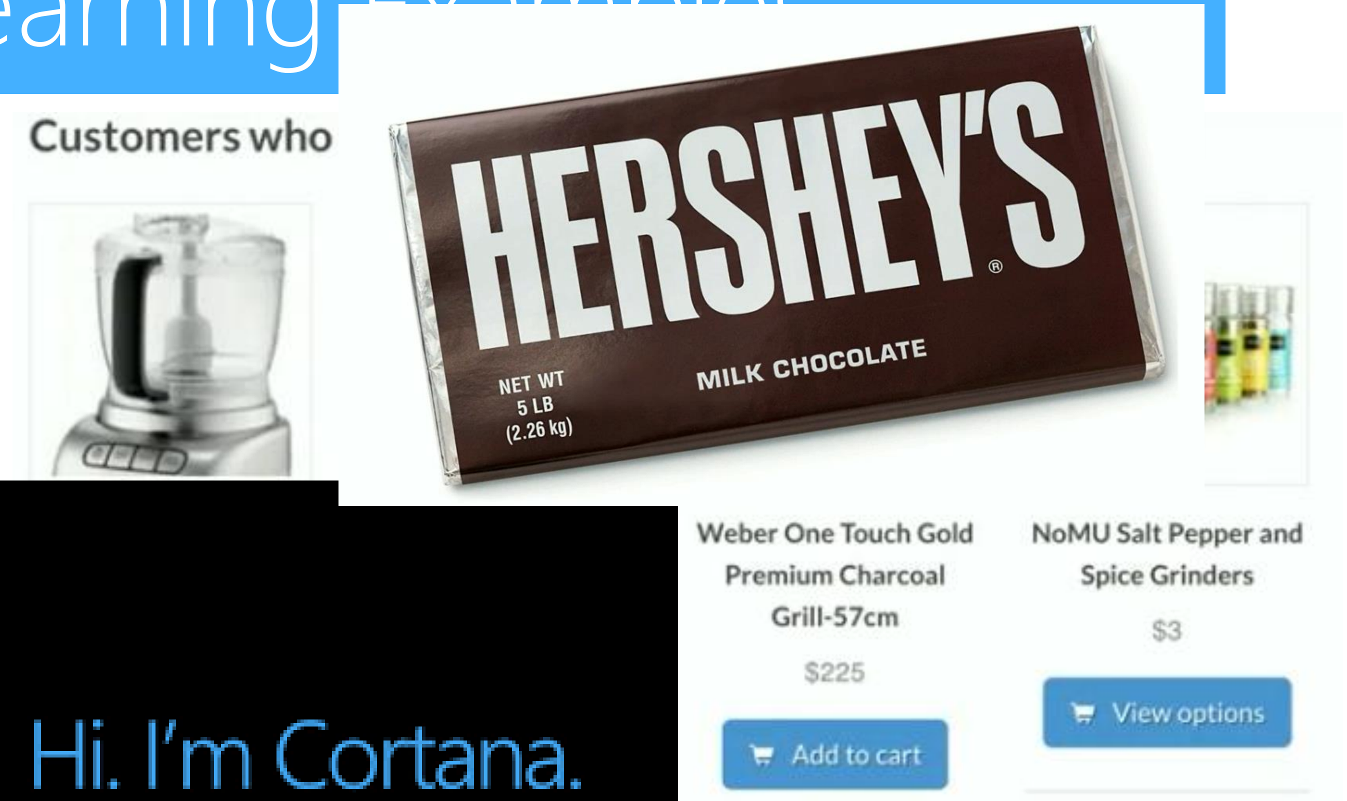
Machine Learning in Real World

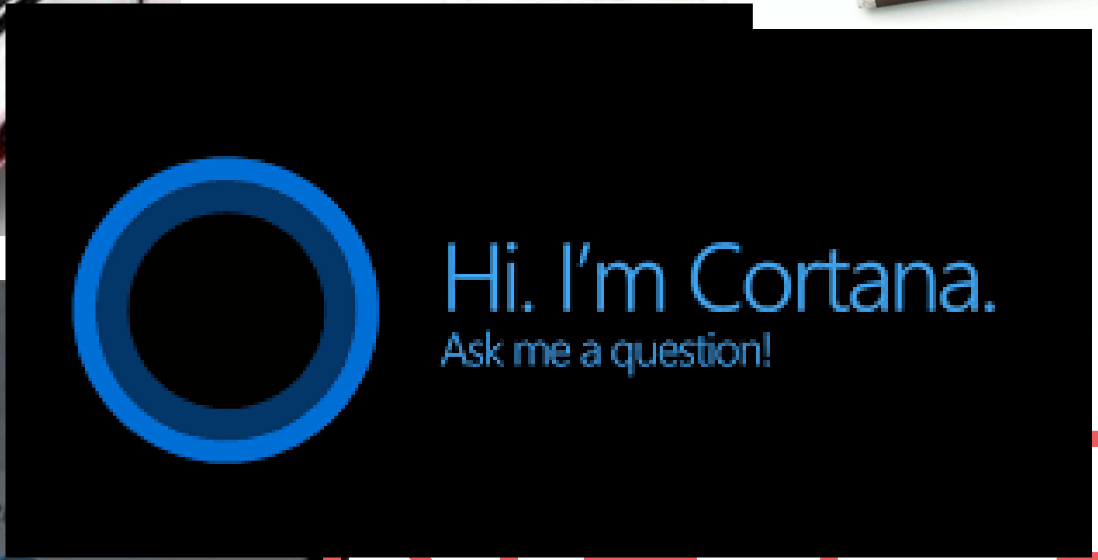
Linkedin

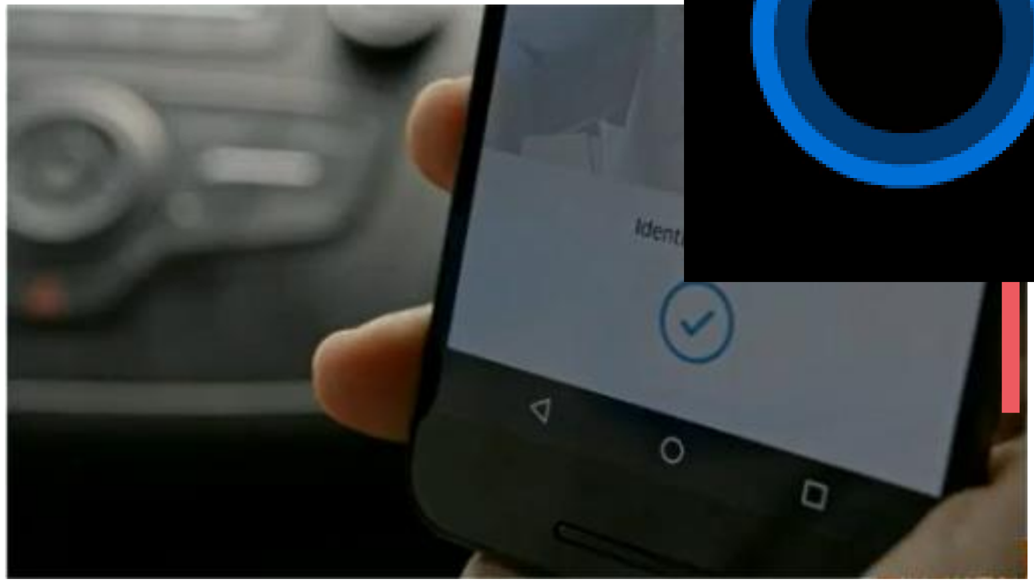
- "People You May Know" ads achieved a click- through rate 30% higher
- They generated millions of new page views.
- Thanks to this one feature, Linkedin's growth trajectory shifted significantly upward.



Few More Machine Learning Examples







Movie Recommendations

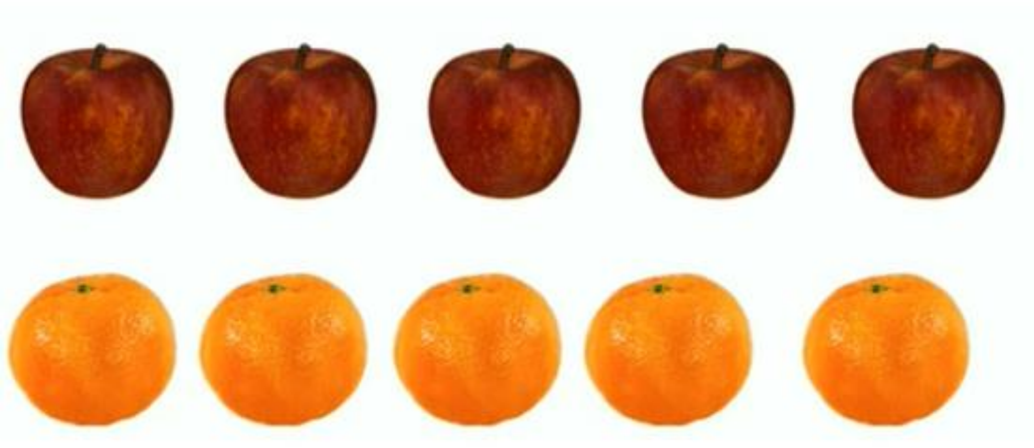
Why Azure ML?

- Drag and Drop interface and no Programming required
- Large variety of algorithm as modules
- From experiment to production API in minutes
- Supports R and Python to bring in your existing code
- Flexibility of data storage; supports variety of data storage options
- Large number of pre-built APIs available as a service

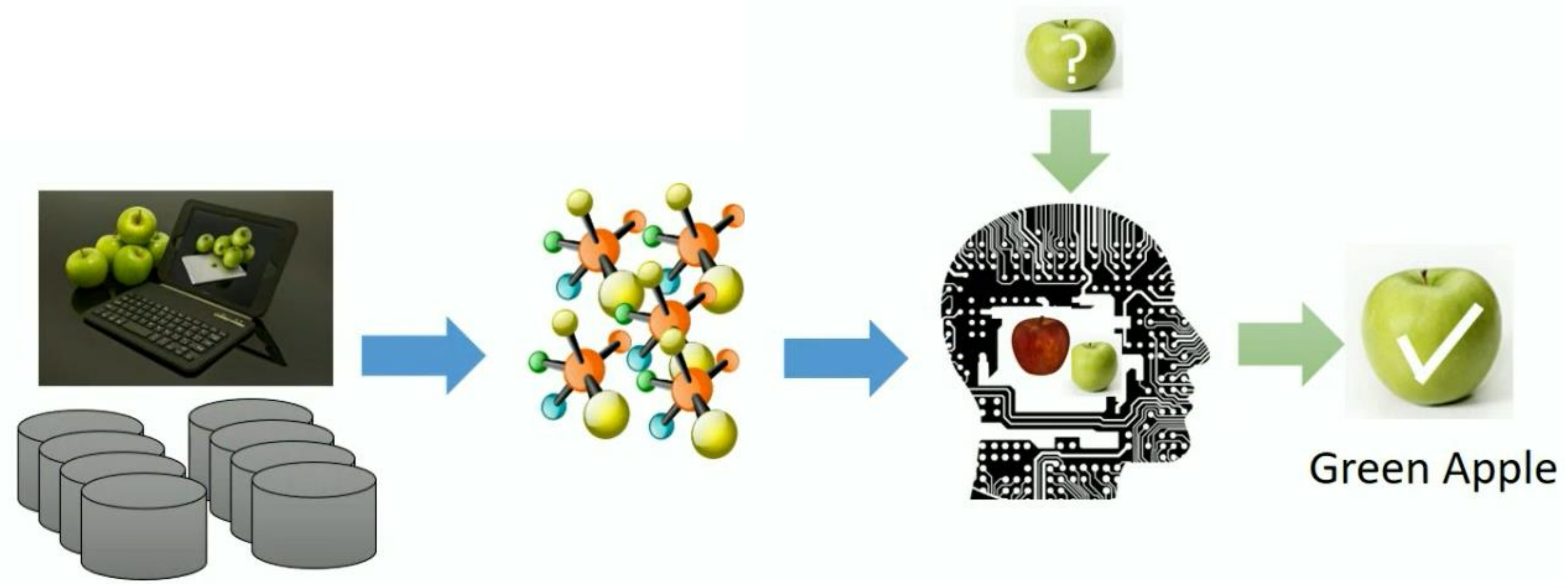
What is Machine Learning?

Machine Learning

- Machine learning is the subfield of computer science that gives "computers the ability to learn without being explicitly programmed".



How Machines Learn?

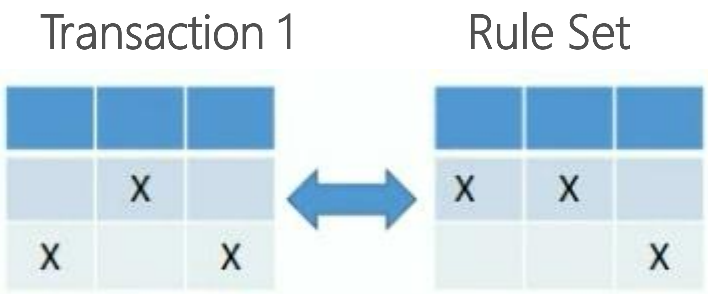


Historic Data

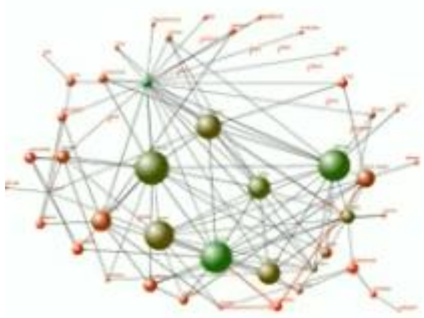
Learning Algorithms

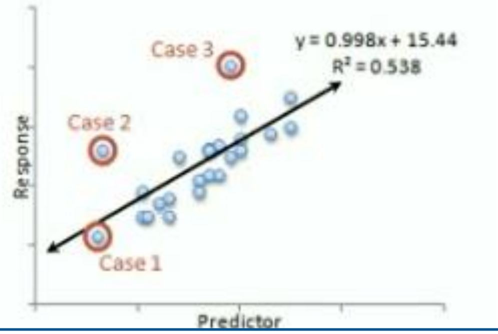
Intelligent Model

Futuristic practices and models



Data Match/Mismatch
triggers rule





Various other predictive techniques

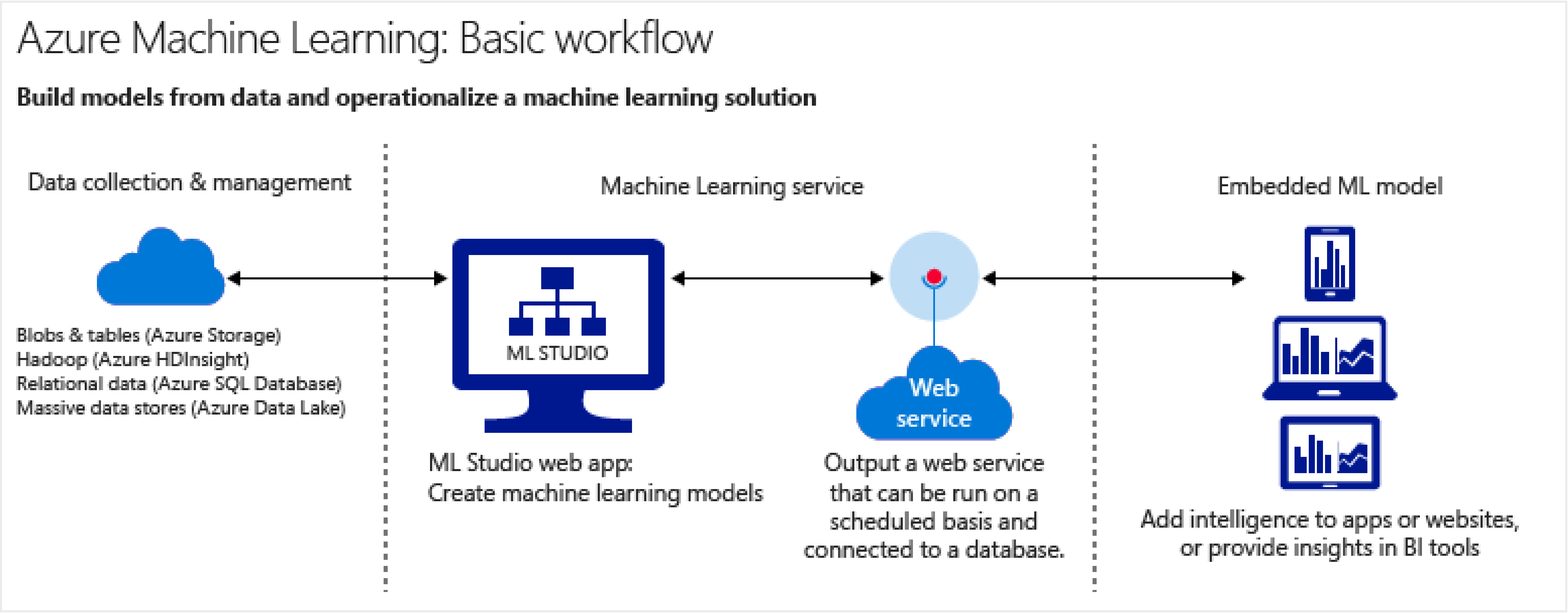
Traditional Approach

- Looking for data matches and rules

Leading or Future Models

- Spotting likelihood using predictive modelling
- Creating a self-learning algorithm that adapts to the trends

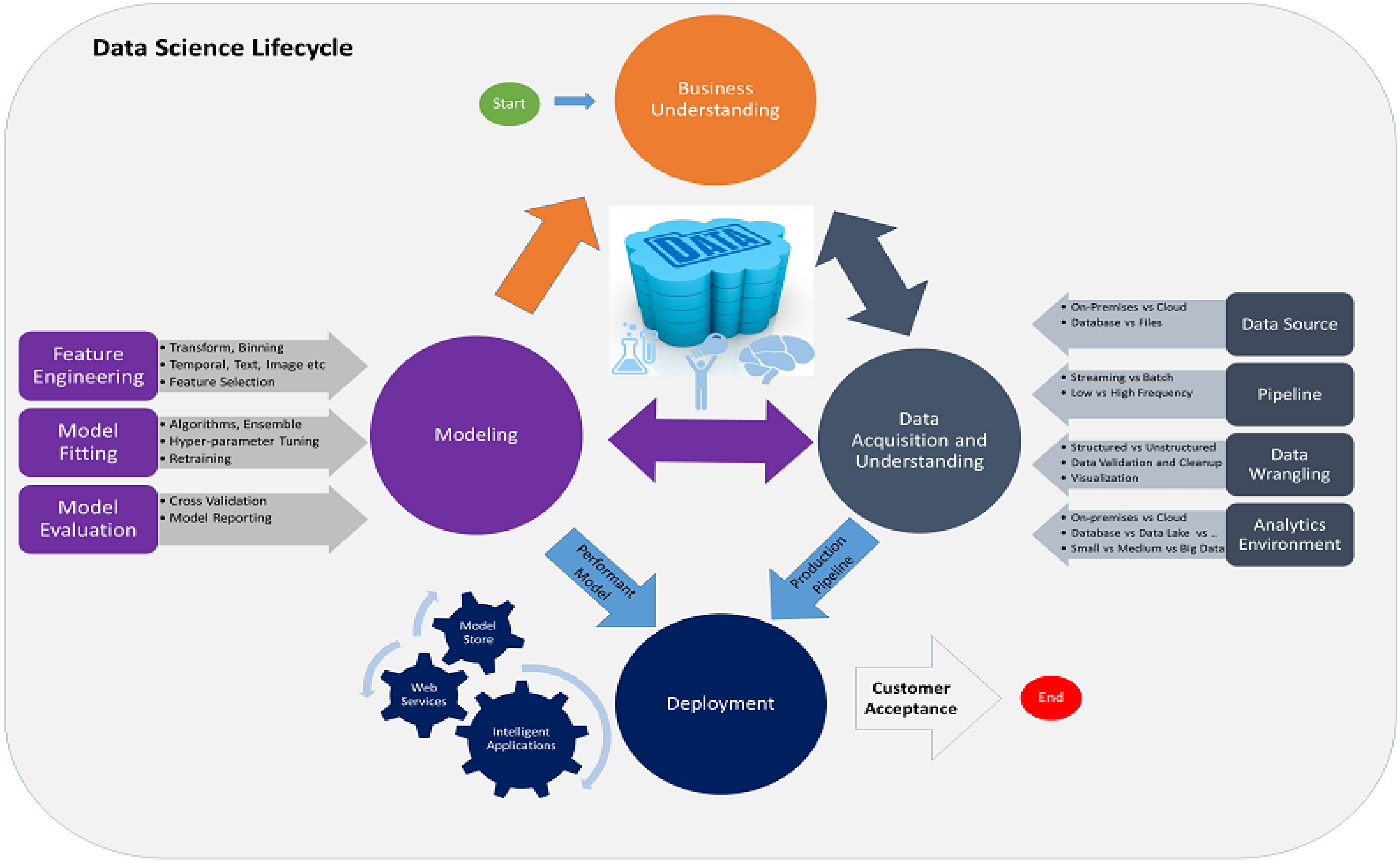
What is Machine Learning in the Microsoft Azure cloud?



Data Science Process

The TDSP lifecycle is composed of five major stages that are executed iteratively. These include:

- **1. Business Understanding**
- **2. Data Acquisition and Understanding**
- **3. Modeling**
- **4. Deployment**
- **5. Customer Acceptance**



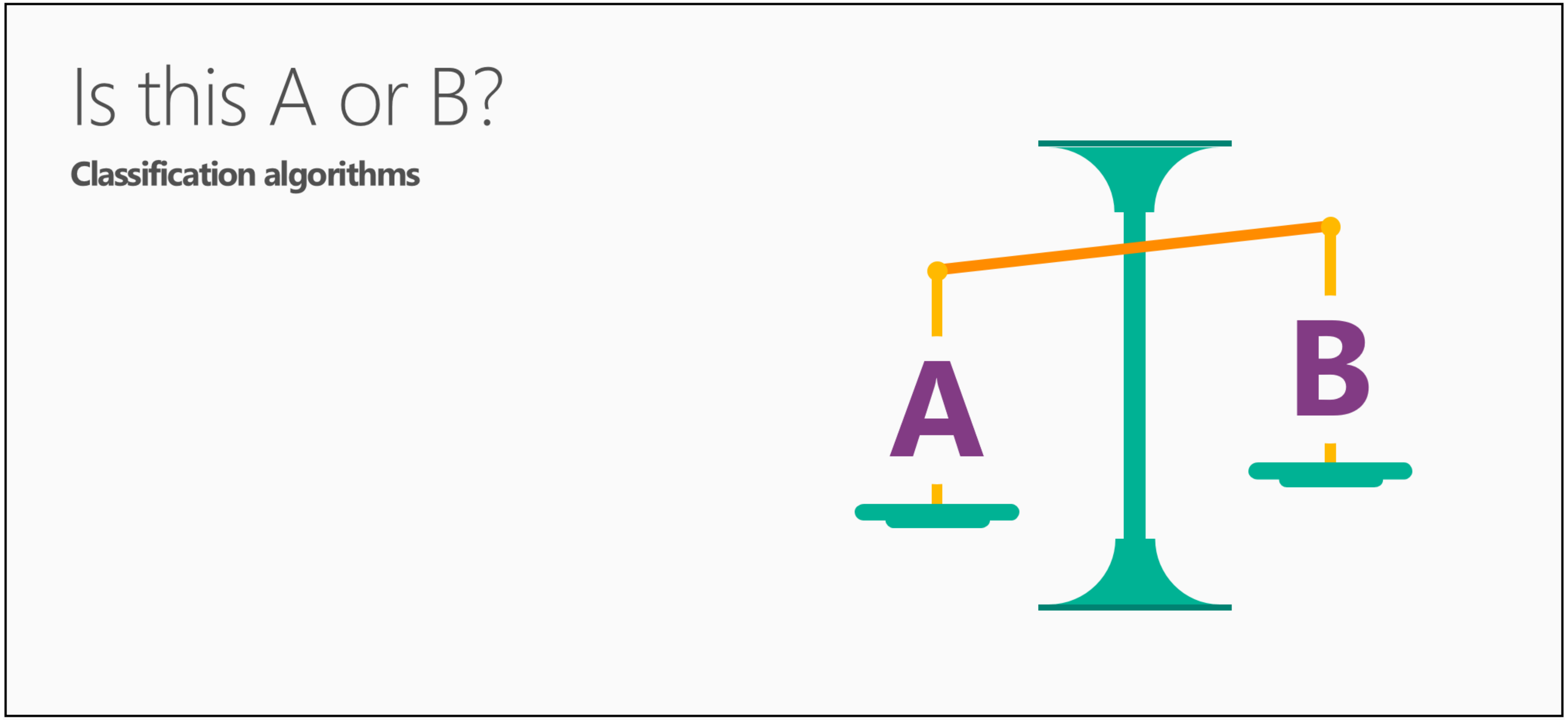
The 5 questions data science answers

It might surprise you, but *there are only five questions that data science answers*:

- **Is this A or B?**
- **Is this weird?**
- **How much – or – How many?**
- **How is this organized?**
- **What should I do next?**

Each one of these questions is answered by a separate family of machine learning methods, called algorithms.

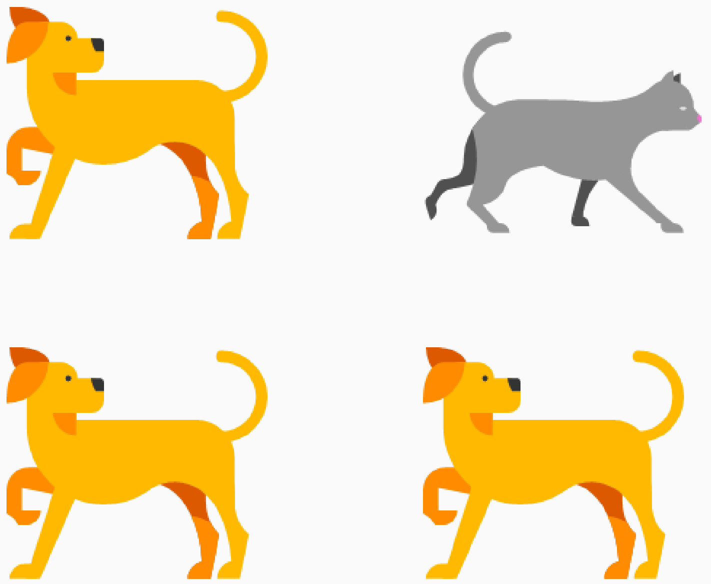
Question 1: Is this A or B? uses classification algorithms



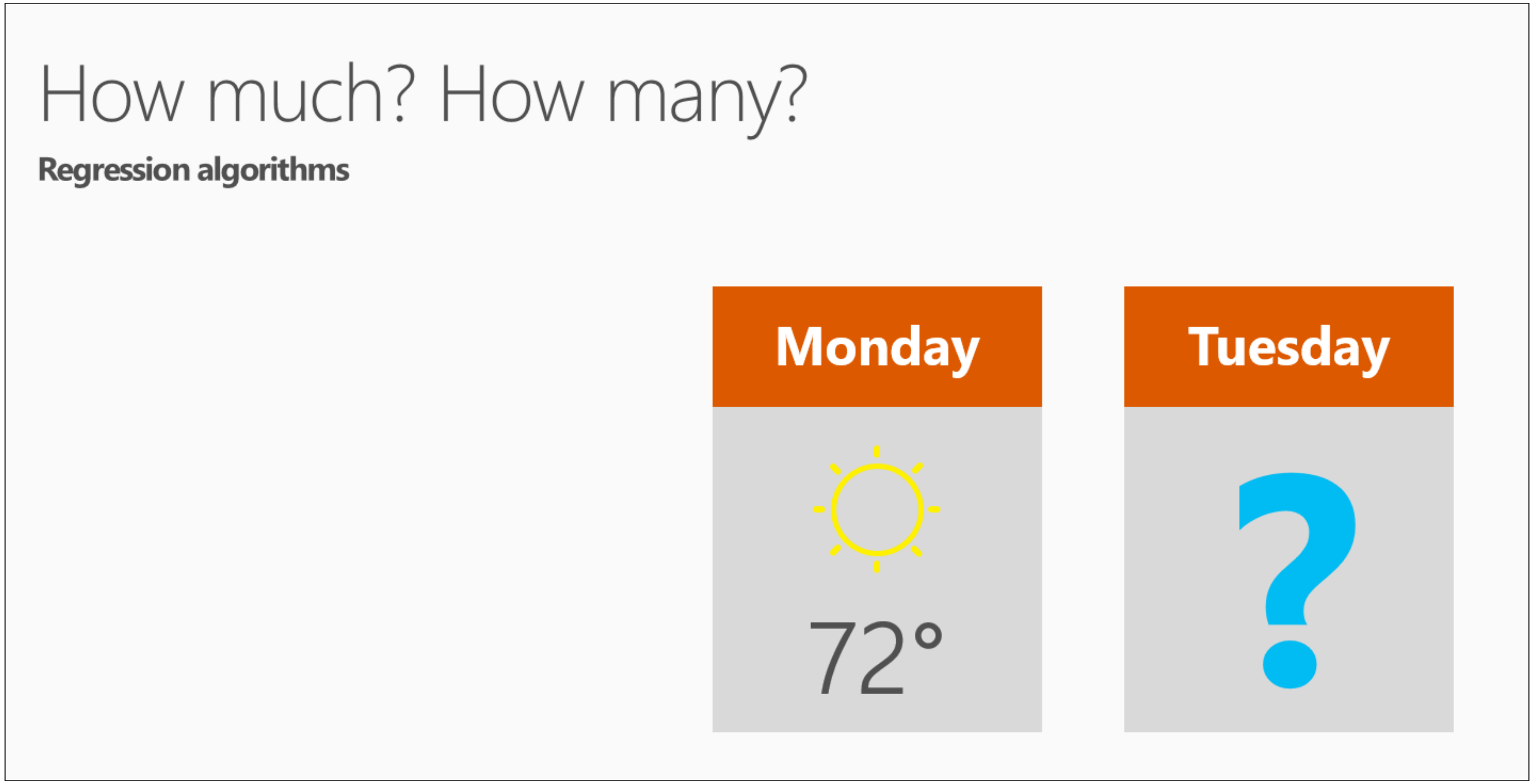
Question 2: Is this weird? uses anomaly detection algorithms

Is this weird?

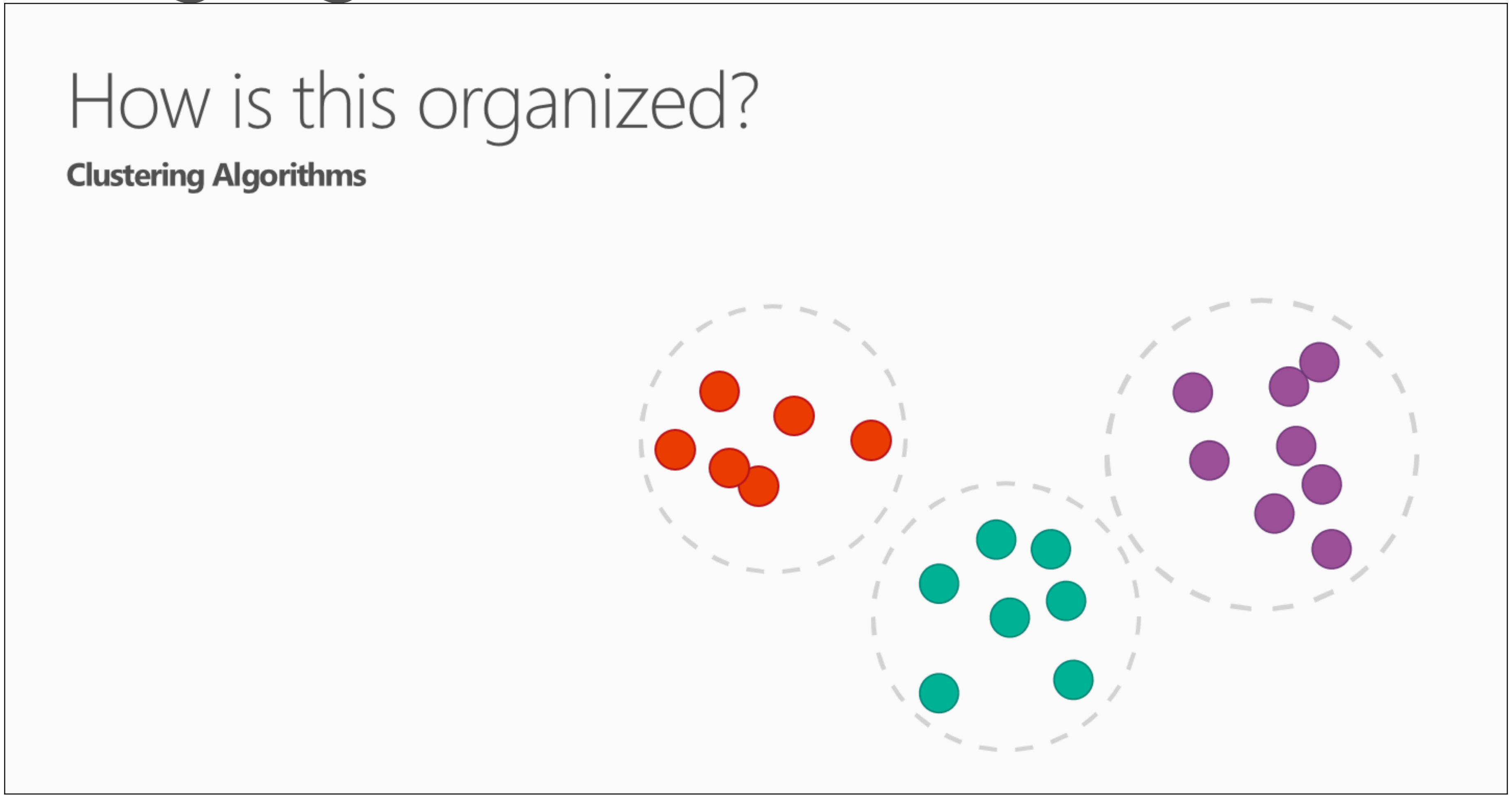
Anomaly detection algorithms



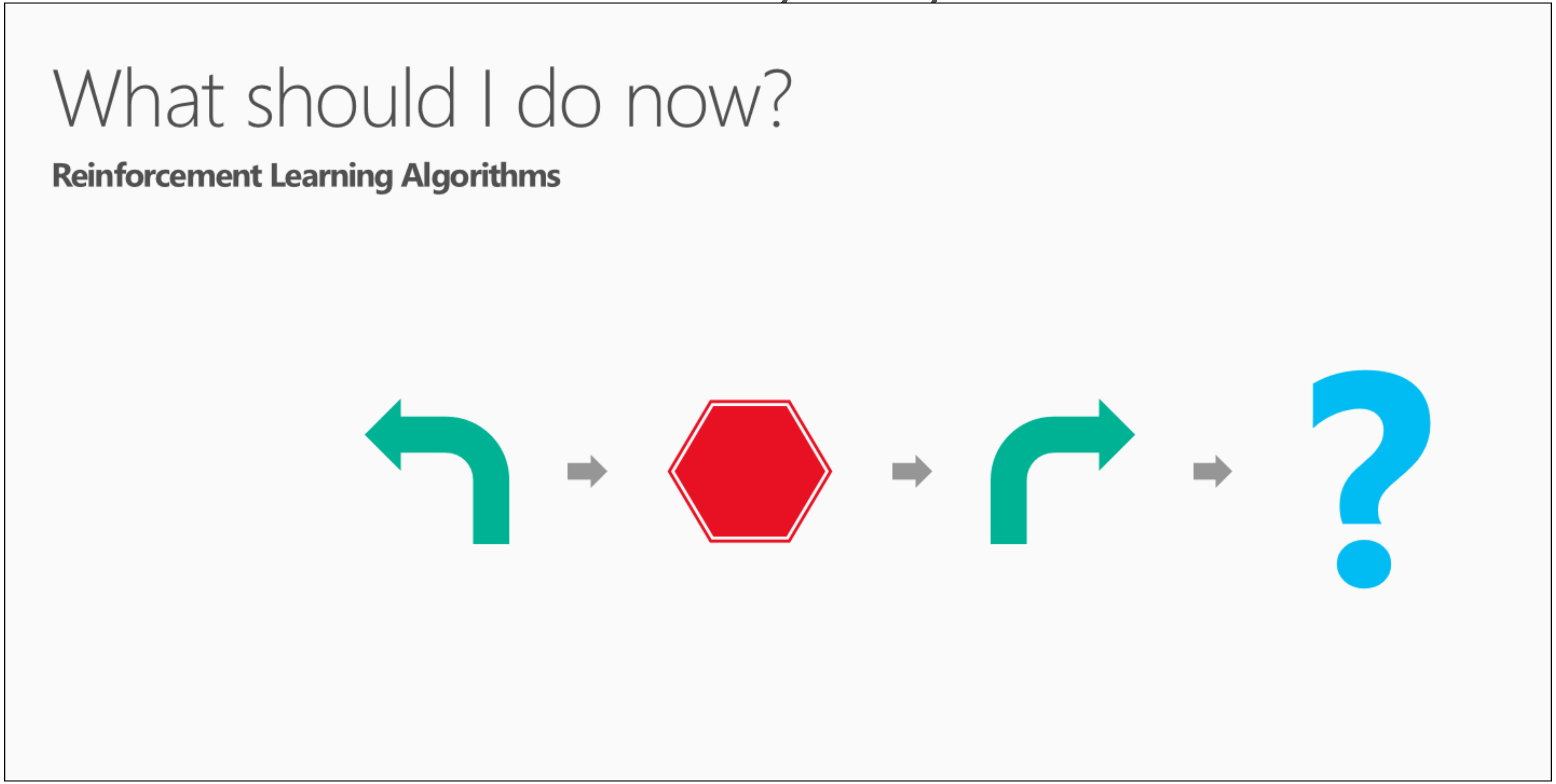
Question 3: How much? or How many? uses regression algorithms



Question 4: How is this organized? uses clustering algorithms

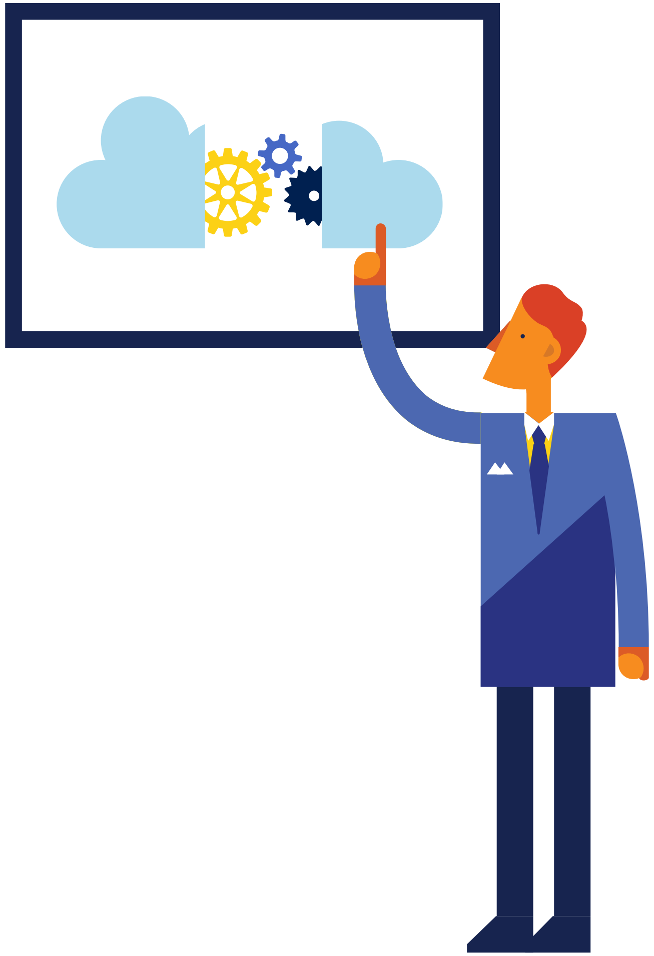


Question 5: What should I do now? uses reinforcement learning algorithms



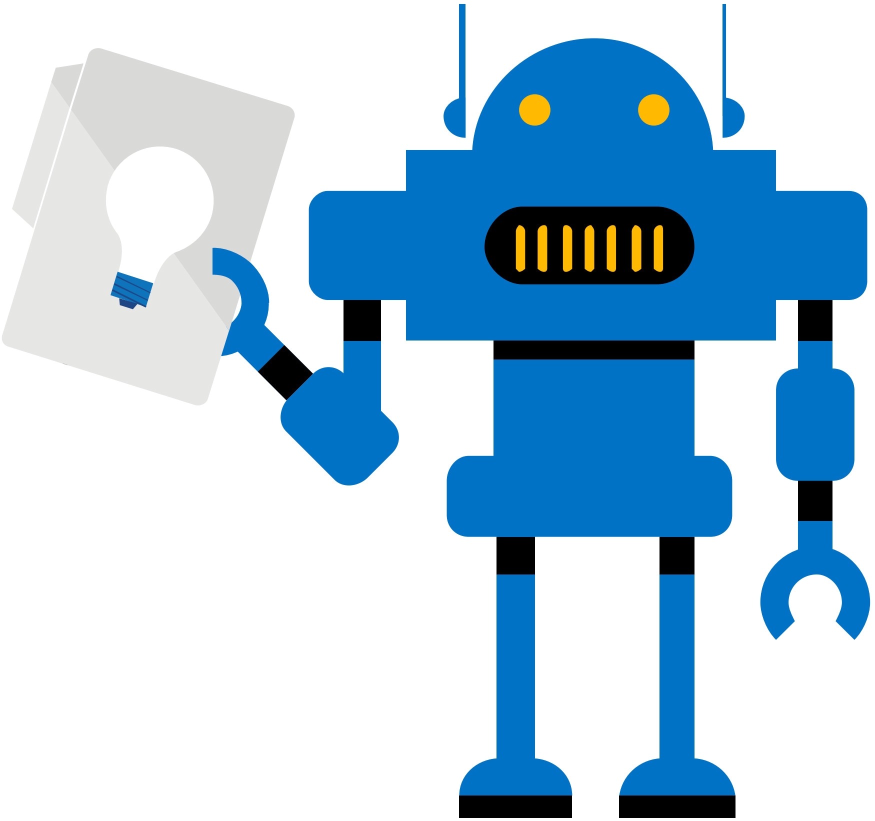
Supervised learning

1. Purpose
2. Process: Data set and test set
3. Components
4. Computation and recommendations
5. Verification
6. Selection



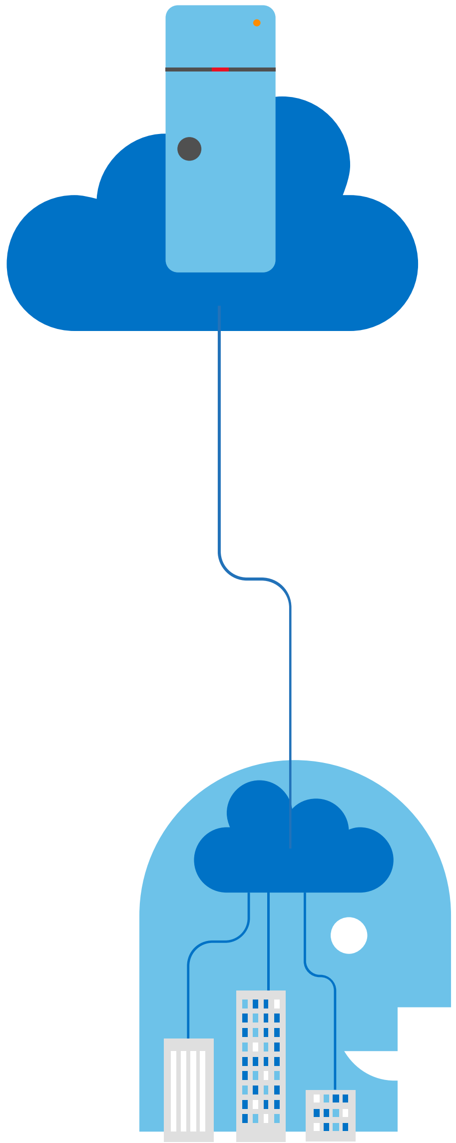
Unsupervised learning

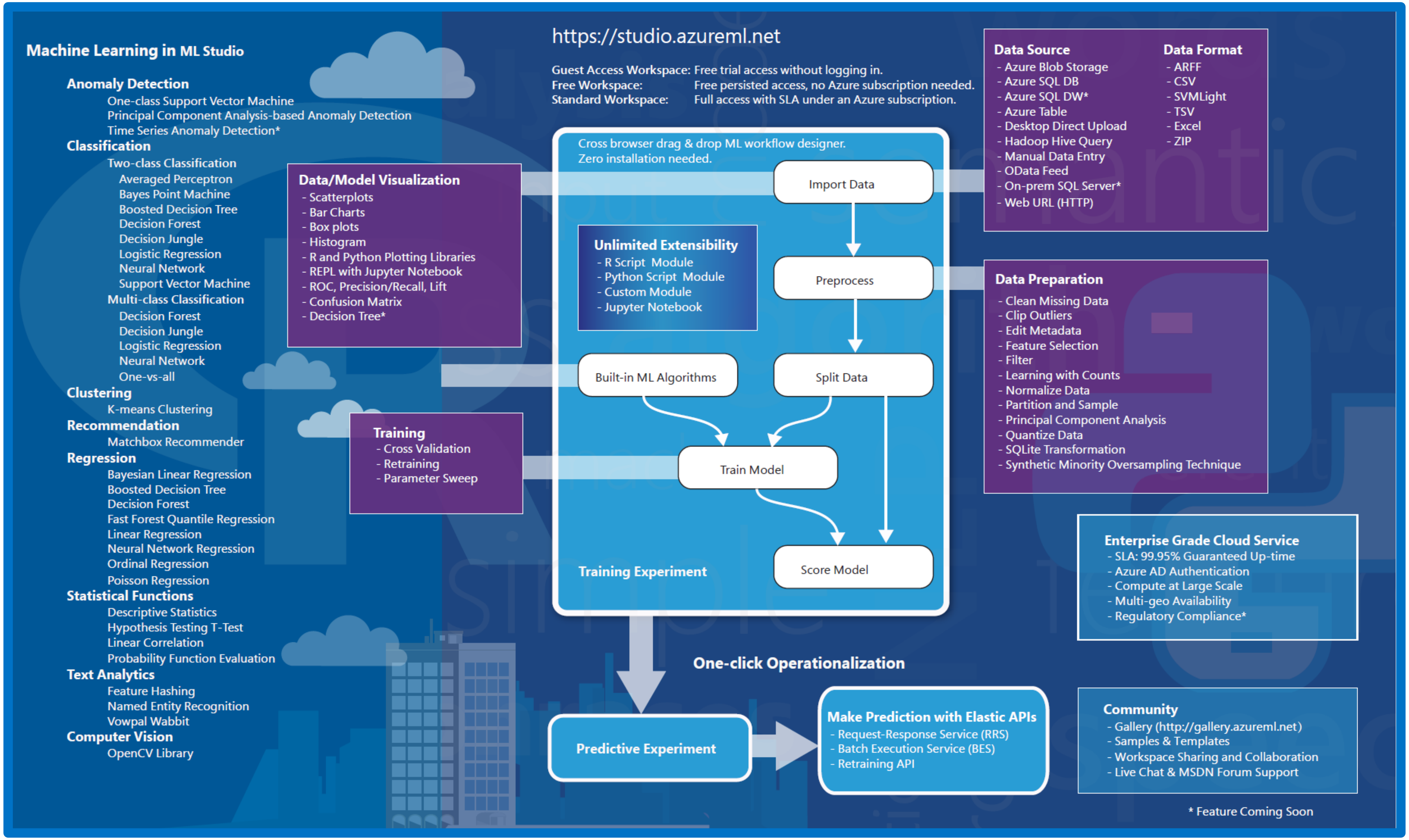
1. Purpose
2. Process
3. Components
4. Computation and recommendation



Azure Machine Learning

- Why Azure Machine Learning?
- Getting started
  - Setting up a Microsoft Azure account
  - Setting up a storage account
  - Setting up an Azure Machine Learning workspace
- Advantages





Machine Learning in ML Studio
Anomaly Detection
One-class Support Vector Machine
Principal Component Analysis-based Anomaly Detection
Time Series Anomaly Detection*
Classification
Two-class Classification
Averaged Perceptron
Bayes Point Machine
Boosted Decision Tree
Decision Forest
Decision Jungle
Logistic Regression
Neural Network
Support Vector Machine
Multi-class Classification
Decision Forest
Decision Jungle
Logistic Regression
Neural Network
One-vs-all
Clustering
K-means Clustering
Recommendation
Matchbox Recommender
Regression
Bayesian Linear Regression
Boosted Decision Tree
Decision Forest
Fast Forest Quantile Regression
Linear Regression
Neural Network Regression
Ordinal Regression
Poisson Regression
Statistical Functions
Descriptive Statistics
Hypothesis Testing T-Test
Linear Correlation
Probability Function Evaluation
Text Analytics
Feature Hashing
Named Entity Recognition
Vowpal Wabbit
Computer Vision
OpenCV Library
https://studio.azureml.net
Guest Access Workspace: Free trial access without logging in.
Free Workspace: Free persisted access, no Azure subscription needed.
Standard Workspace: Full access with SLA under an Azure subscription.
Data Source
- Azure Blob Storage
- Azure SQL DB
- Azure SQL DW*
- Azure Table
- Desktop Direct Upload
- Hadoop Hive Query
- Manual Data Entry
- OData Feed
- On-prem SQL Server*
- Web URL (HTTP)
Data Format
- ARFF
- CSV
- SVMLight
- TSV
- Excel
- ZIP
Data/Model Visualization
- Scatterplots
- Bar Charts
- Box plots
- Histogram
- R and Python Plotting Libraries
- REPL with Jupyter Notebook
- ROC, Precision/Recall, Lift
- Confusion Matrix
- Decision Tree*
Cross browser drag & drop ML workflow designer.
Zero installation needed.
Import Data
Unlimited Extensibility
- R Script Module
- Python Script Module
- Custom Module
- Jupyter Notebook
Preprocess
Built-in ML Algorithms
Split Data
Train Model
Score Model
Training Experiment
Training
- Cross Validation
- Retraining
- Parameter Sweep
Data Preparation
- Clean Missing Data
- Clip Outliers
- Edit Metadata
- Feature Selection
- Filter
- Learning with Counts
- Normalize Data
- Partition and Sample
- Principal Component Analysis
- Quantize Data
- SQLite Transformation
- Synthetic Minority Oversampling Technique
Enterprise Grade Cloud Service
- SLA: 99.95% Guaranteed Up-time
- Azure AD Authentication
- Compute at Large Scale
- Multi-geo Availability
- Regulatory Compliance*
One-click Operationalization
Predictive Experiment
Make Prediction with Elastic APIs
- Request-Response Service (RRS)
- Batch Execution Service (BES)
- Retraining API
Community
- Gallery (http://gallery.azureml.net)
- Samples & Templates
- Workspace Sharing and Collaboration
- Live Chat & MSDN Forum Support
* Feature Coming Soon

The algorithms clubbed

- Classification: Predict what class case belongs to

  Scenarios: Churn analysis, fraud detection, speech recognition

  Algorithms: Boosted Decision Tree, Decision Forest, Logistic Regression,…

- Regression: Predict numerical outcomes

  Scenarios: Stock prices prediction, sales forecasts, quality control

  Algorithms: Bayesian Linear, Linear Regression, Decision Forest

- Clustering: Discover natural groupings of cases

  Scenarios: Customer segmentation, pattern recognition

  Algorithms: K-means

- Anomaly Detection: Explore unusual patterns

  Scenarios: Network intrusion, fraud detection

  Algorithms: One class support vector machine ,PCA based anomaly detection

Understanding Data, Variables/Features

Understanding The Variables Using a Dataset

Loan_ID	Gender	Married	Dependents	Self_Employed	Income	LoanAmt	Term	CreditHistory	Property_Area	Status
LP001002	Male	No	0	No	$5,849.00		60	1	Urban	Y
LP001003	Male	Yes	1	No	$4,583.00	$128.00	120	1	Rural	N
LP001005	Male	Yes	0	Yes	$3,000.00	$66.00	60	1	Urban	Y
LP001006	Male	Yes	2	No	$2,583.00	$120.00	60	1	Urban	Y

Types of Variables

- Predictor/ Independent
  - Gender
  - Married
  - Dependents
  - Self _ Employed
  - Income
  - Loan Amt
  - Term
  - Credit History
  - Property Area
- Target/Dependent
  - Status

Data Type

- Character/String
  - Gender
  - Married
  - Self-Employed
  - Property Area
  - Status
- Numeric
  - Dependents
  - Income
  - Loan Amt
  - Term
  - Credit History

Category

- Categorical
  - Gender
  - Married
  - Credit History
  - Self-Employed
  - Property Area
  - Status
- Continuous
  - Dependents
  - Income
  - Loan Amt
  - Term

Types of Models

Types of Models

- Classification
- Regression
- Clustering
- Anomaly Detection

Classification

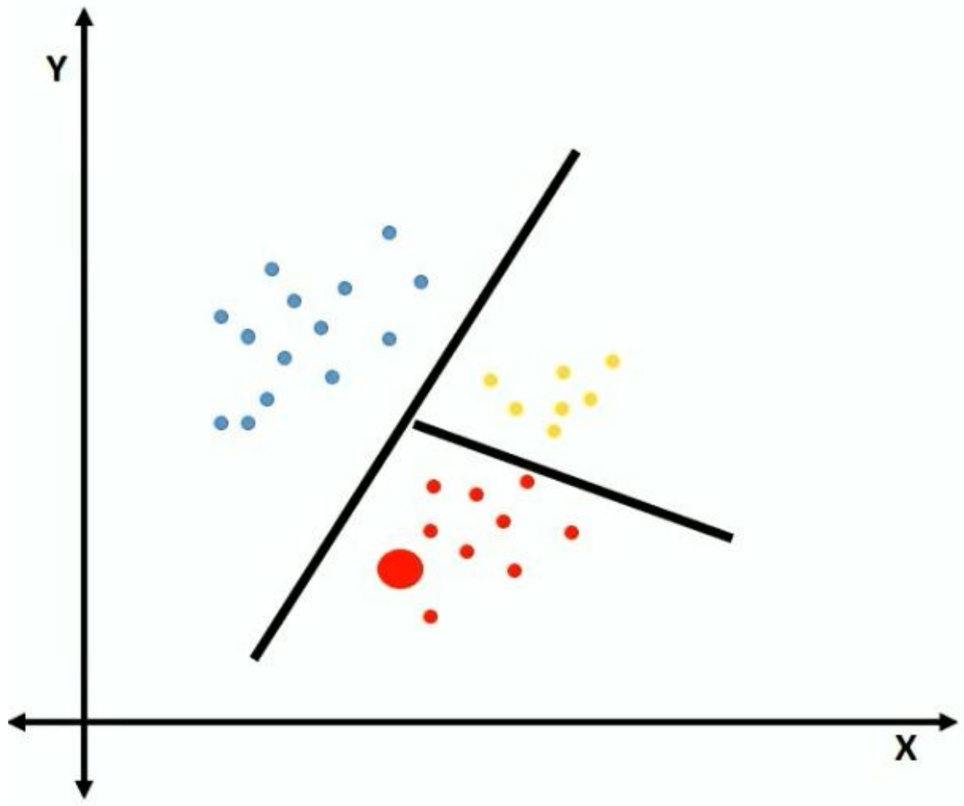
- Classification is identifying to which set of categories a new observation belongs, on the basis of a training set of data containing observations whose category membership is known.

- Binary/ Two-Class Classification — Either/Or, Yes or No type

- Multi-Class Classification — One of the many alternatives

Some examples could be

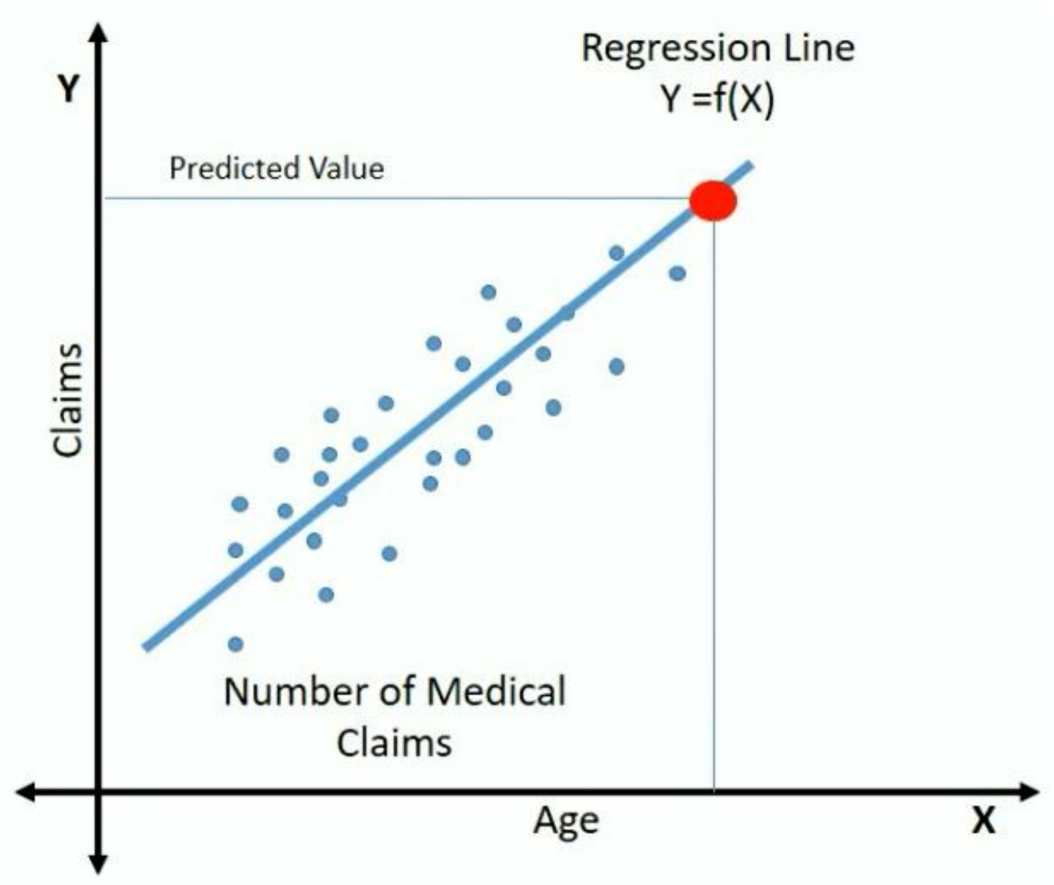
- Assigning a given email into "spam" or "non-spam" classes Or Primary, Social or Promotional emails
- Will this customer default on loan repayment?
- Will this customer buy my product?



Predicting the value for categorical variable

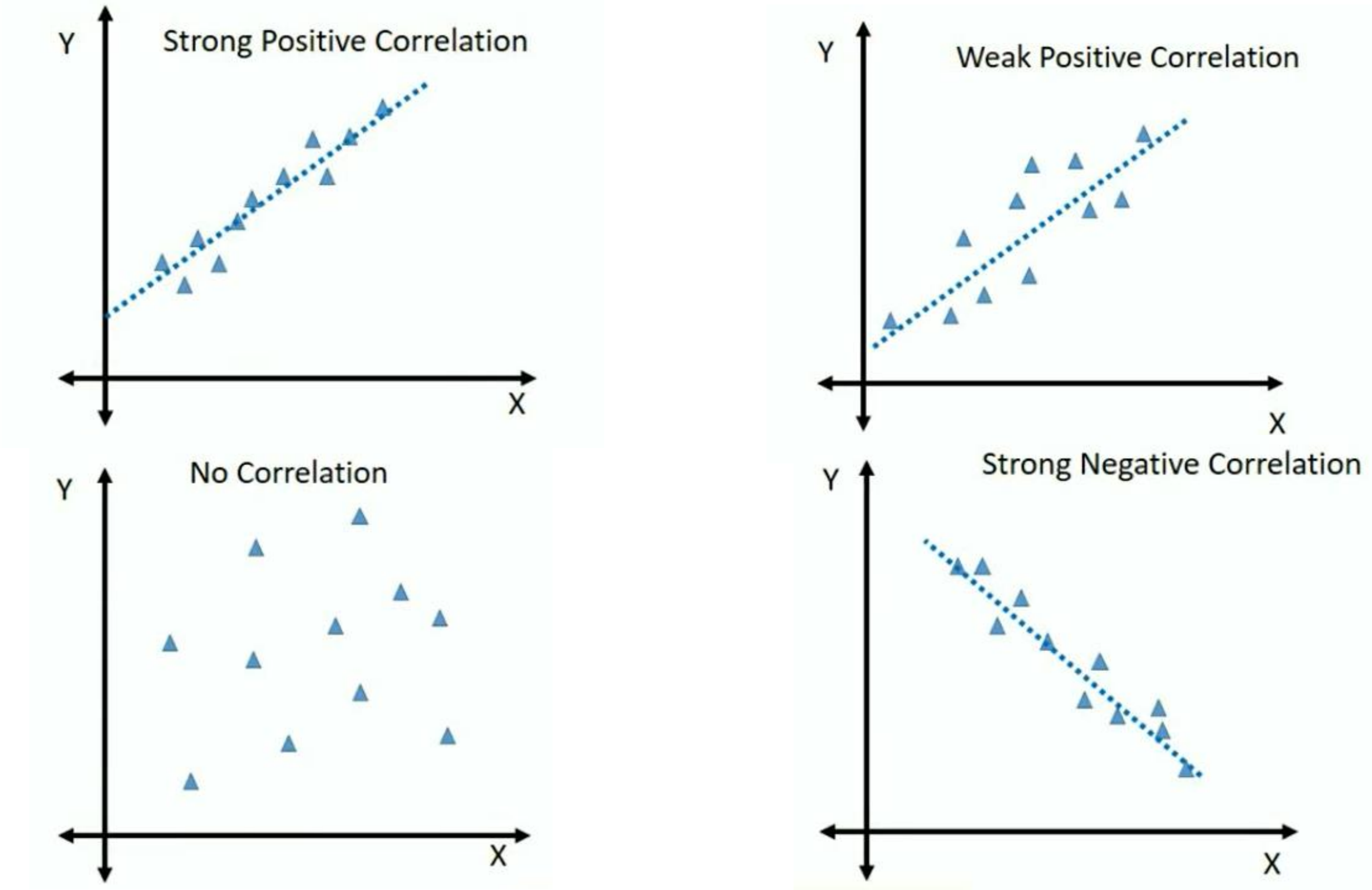
Regression Analysis

- Regression Analysis is a statistical process for estimating the relationships among variables where the predictor is a continuous variable
- The focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors')
- One of the most common methods used in Machine Learning
- In certain circumstances, it can also be used to infer causal relationships between dependent and independent variables.



Examples
- Predicting the future sale of products
- Computing fair price of the product or service

Casual Relationship?

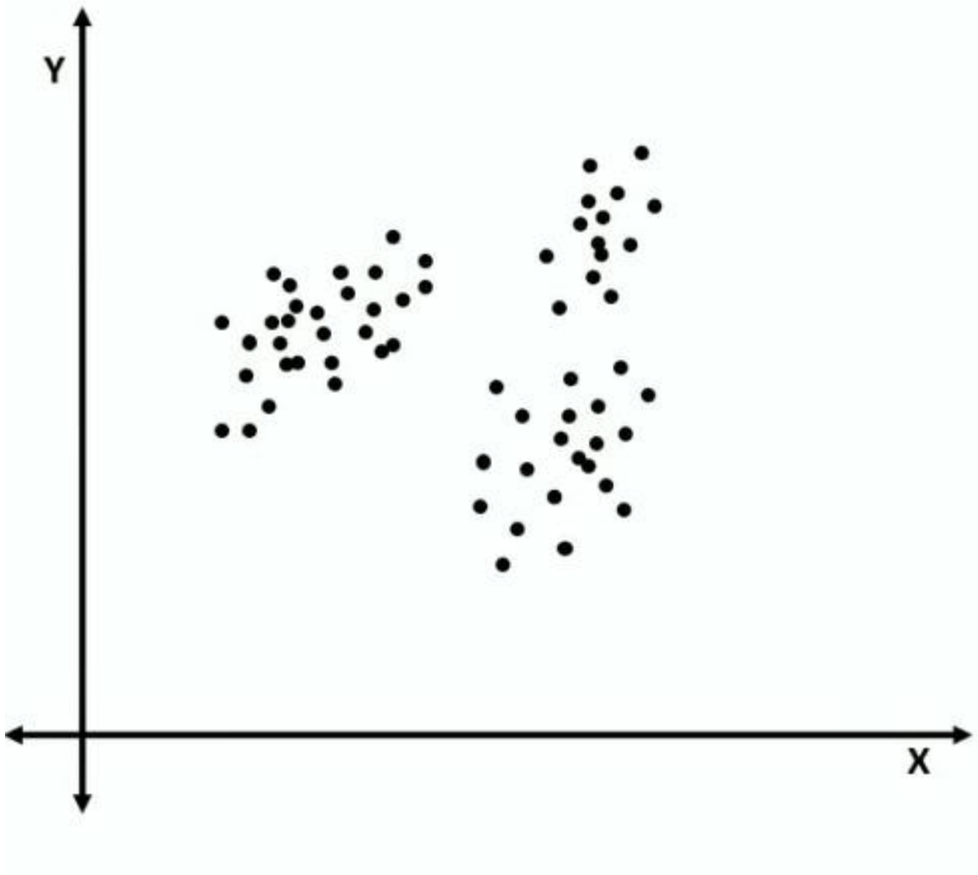

Strong Positive Correlation
Weak Positive Correlation
No Correlation
Strong Negative Correlation
Y
X

Clustering or Cluster Analysis

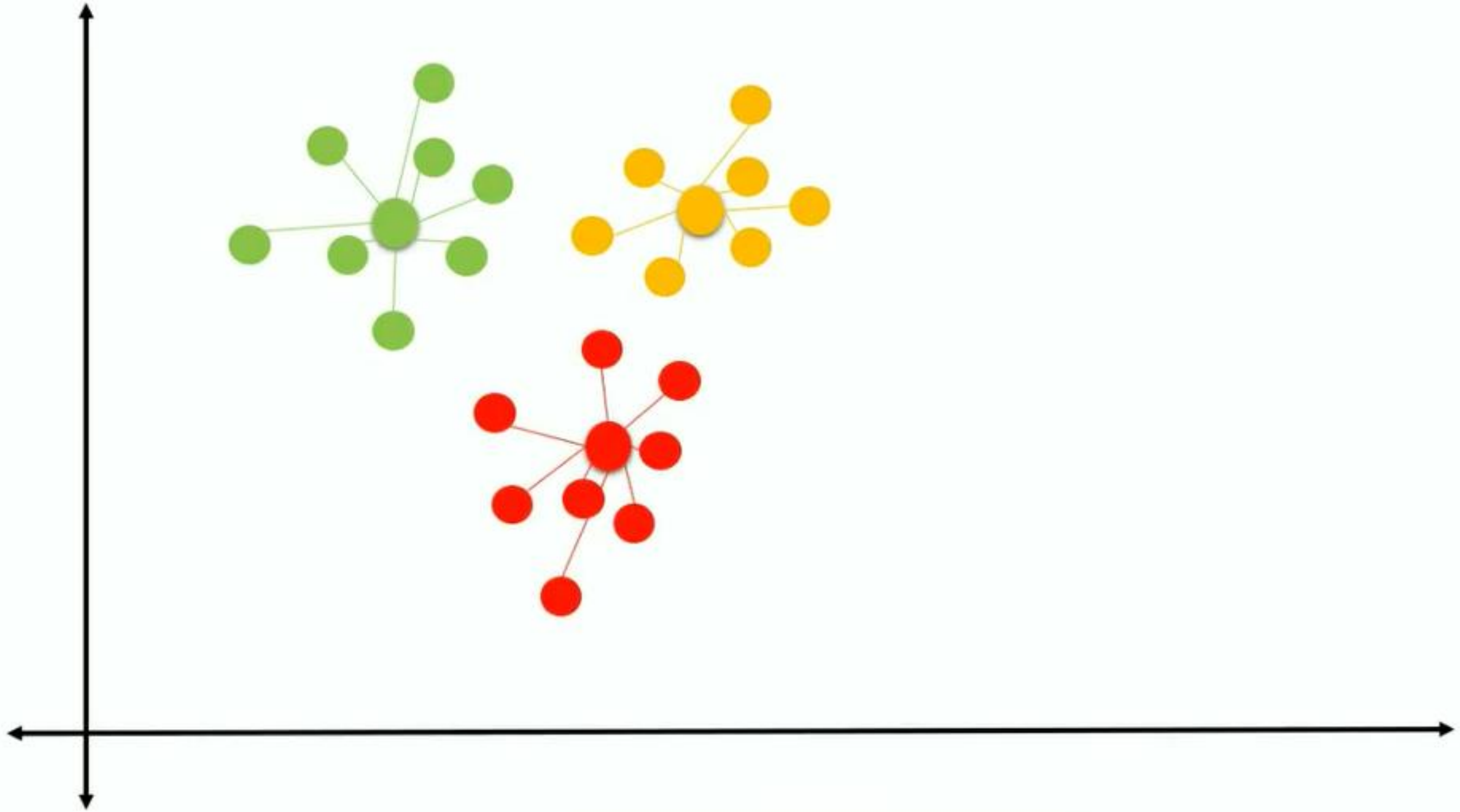
- Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters).

- Unsupervised Learning model

- Customers who make lot of long-distance calls and don't have a job. Who are they?



Cluster Formation

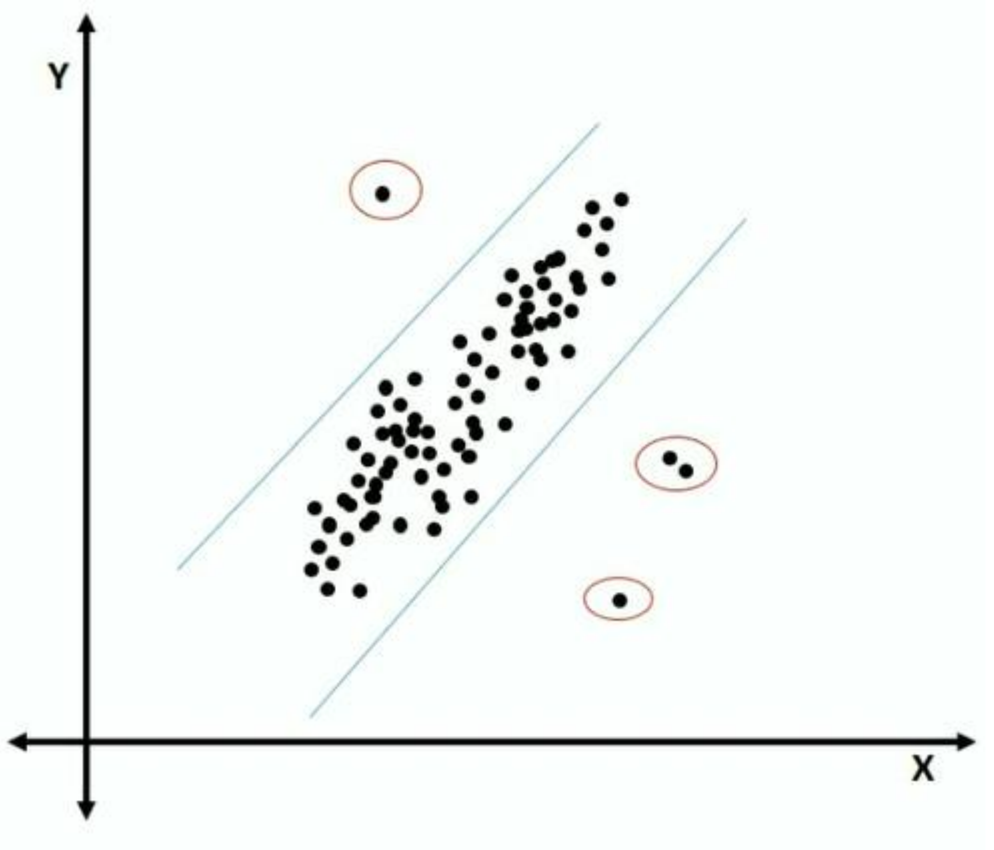


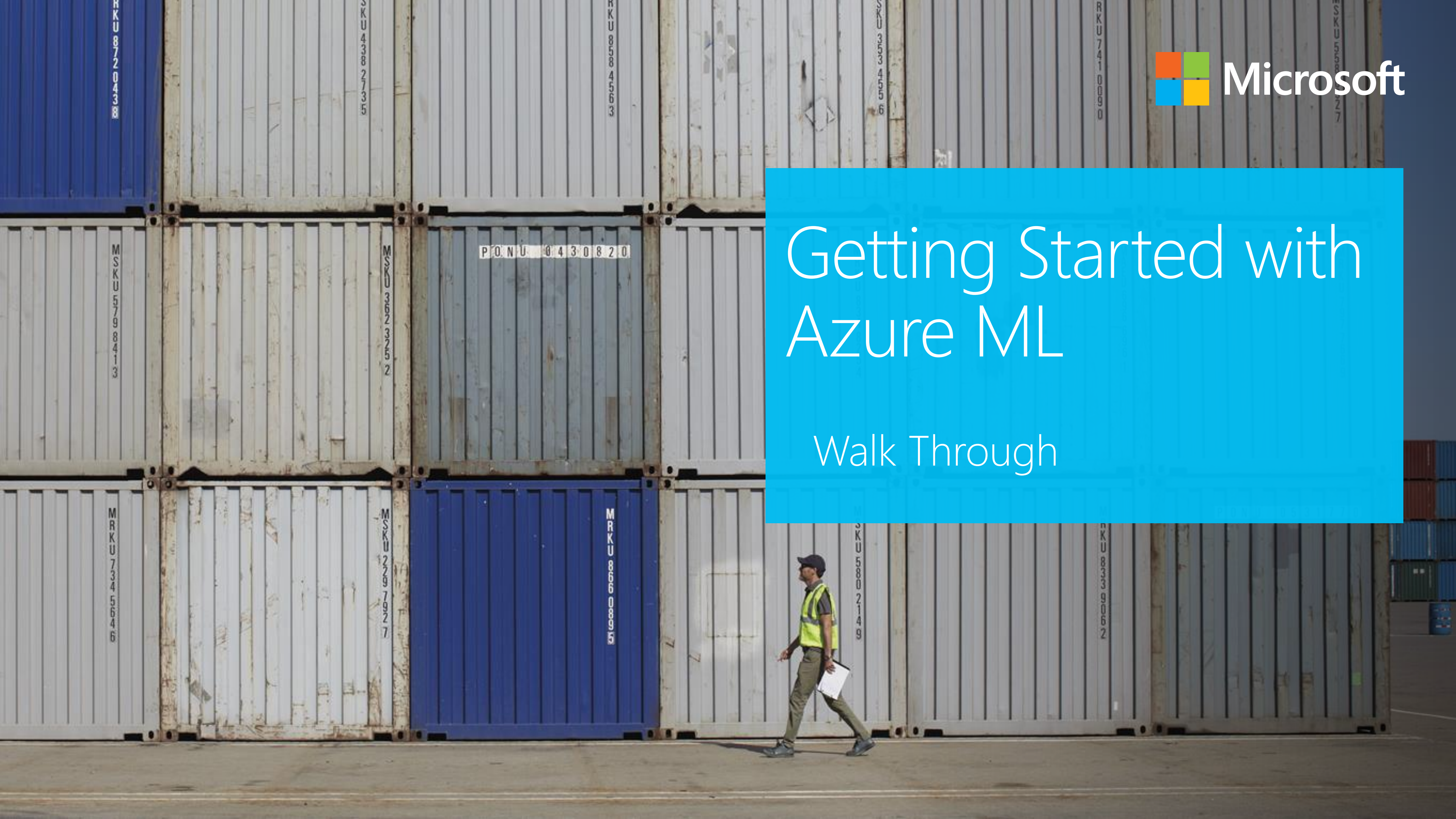
Anomaly Detection

- Anomaly detection (also outlier detection) is the Identification of items, events or observations which do not conform to an expected pattern or other items in a dataset.

- Typically the anomalous items will translate to some kind of problem such as
    - Bank fraud
    - Credit Card Fraud
    - Structural defect
    - Medical problems

- Anomalies are also referred to as outliers, novelties, noise, deviations and exceptions.





Microsoft

Getting Started with Azure ML

Walk Through

Covered– Basics of Machine Learning

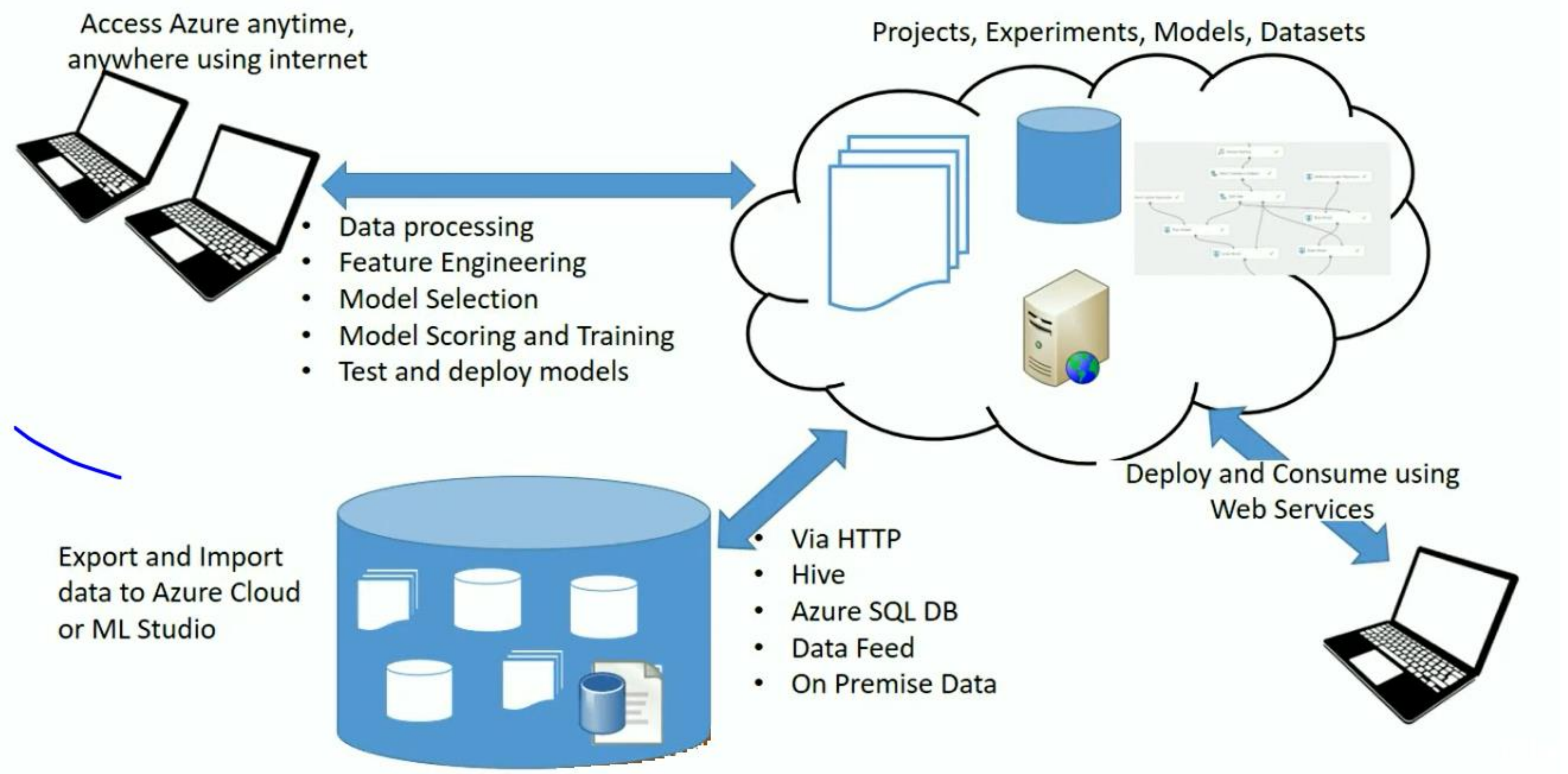
- Why Machine Learning is the Future?
- What is Machine Learning?
- Understanding the data
- Common Machine Learning Terms
- Types of Machine Learning Models

What is Microsoft Azure ML?

Azure Machine Learning

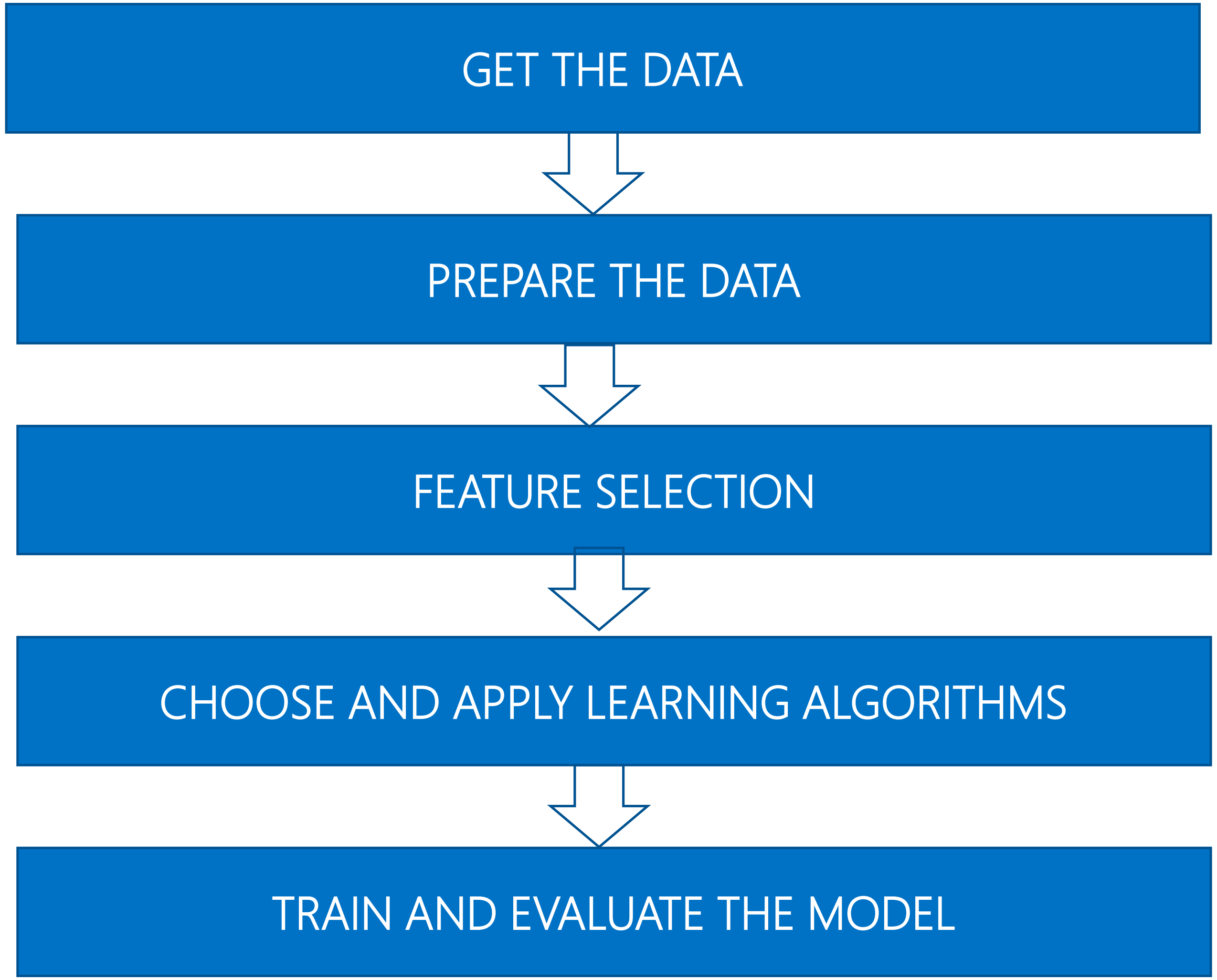
- Cloud based predictive analytics service
- Provides tools to create complete machine learning solution in the cloud
- Quick model creation and deployment using Azure ML Studio
- Allows Models to be deployed as web services
- Provides a large library of Pre-Built Machine learning algorithms and Modules
- Allows for extending your models with custom built R and Python code

Azure ML Studio Overview



Creating Azure ML Account

Workflow of Azure ML



Workflow of Azure ML

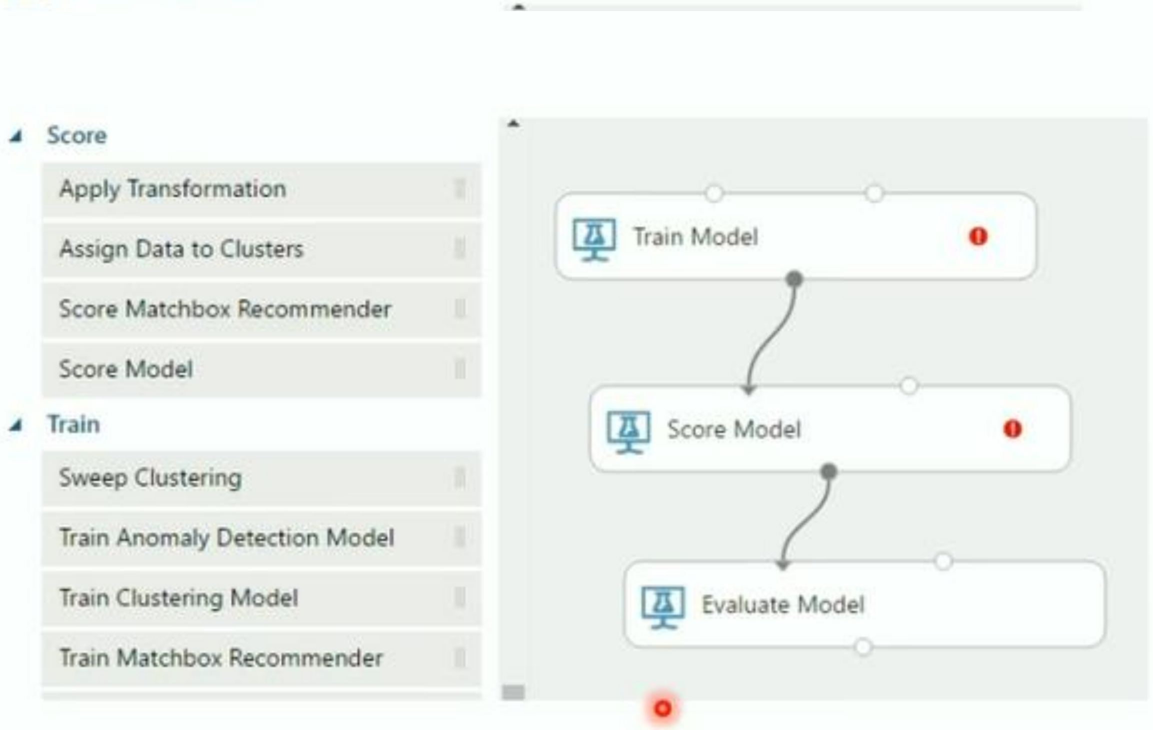
GET THE DATA

PREPARE THE DATA

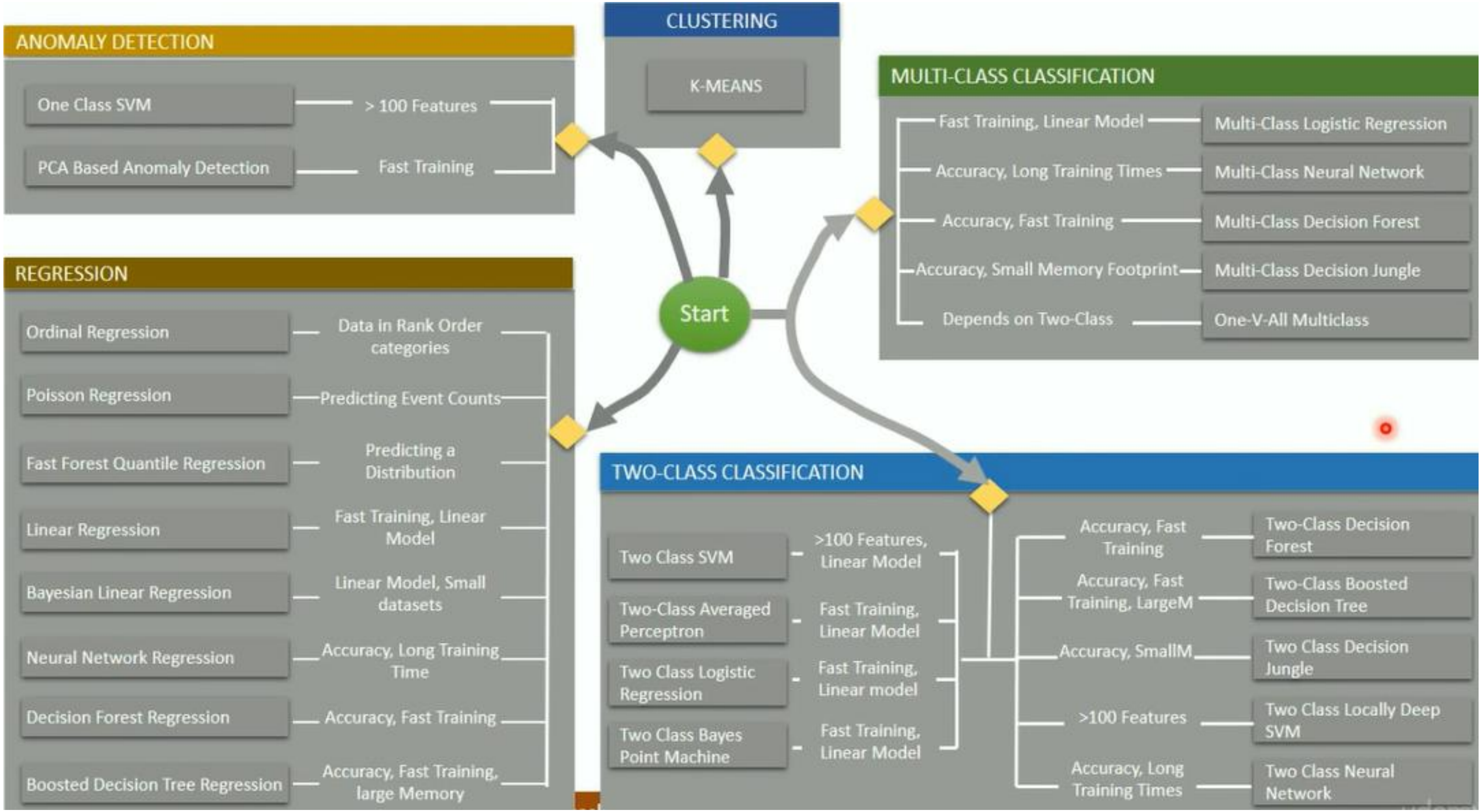
Feature Selection

CHOOSE AND APPLY LEARNING ALGORITHMS

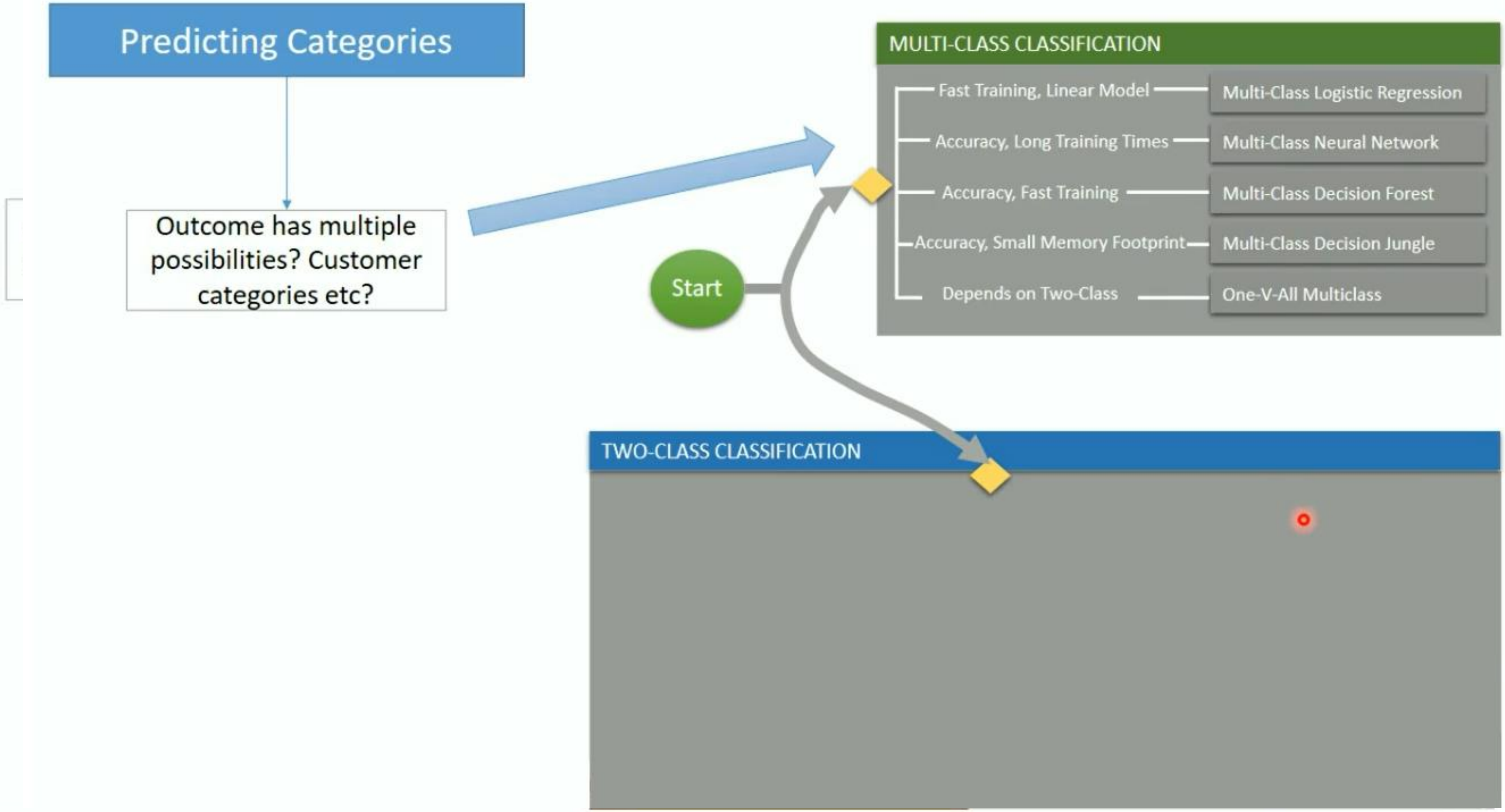
TRAIN AND EVALUATE THE MODEL



Azure ML Algorithm CheatSheet



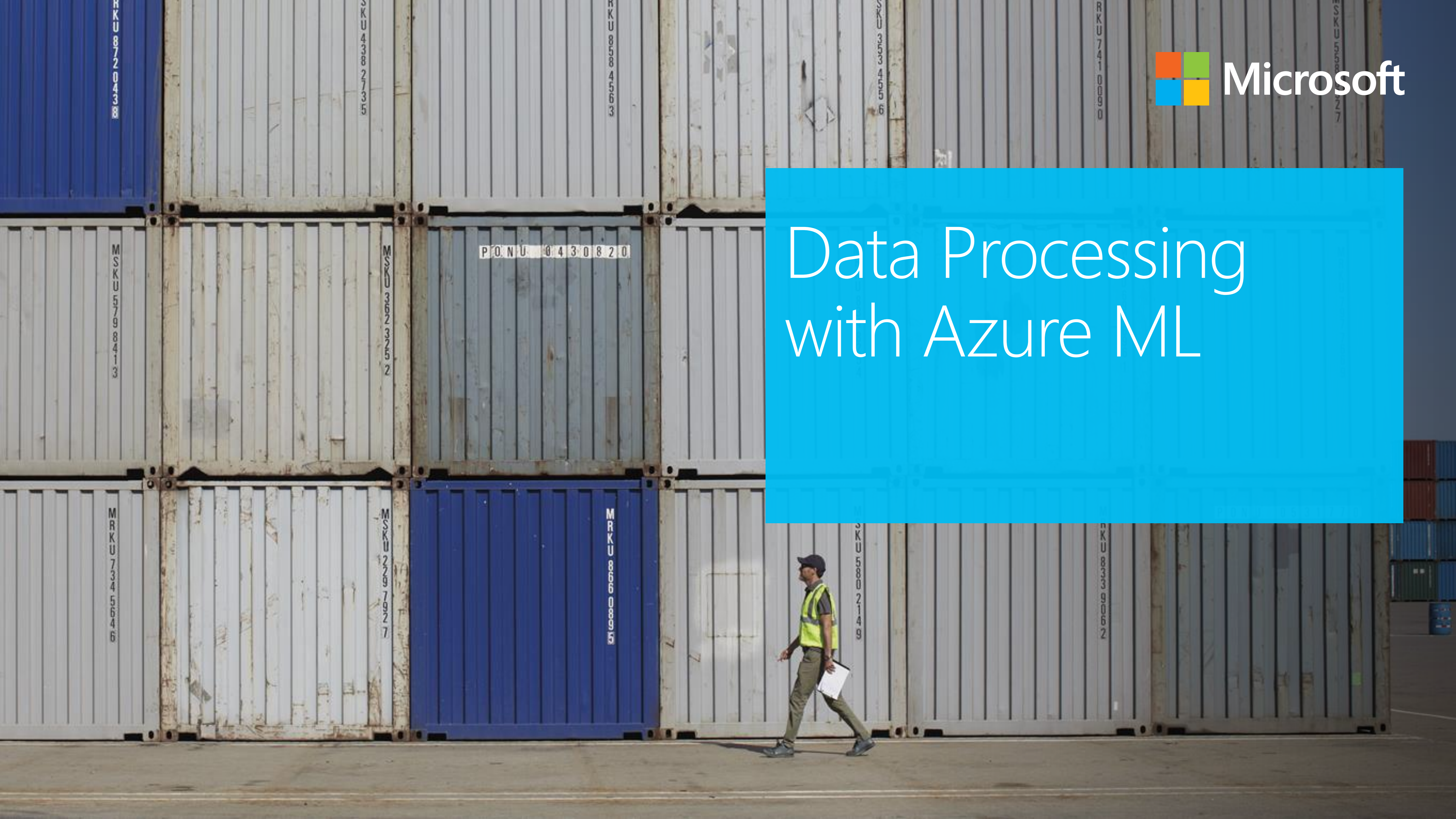
ANOMALY DETECTION
One Class SVM
> 100 Features
PCA Based Anomaly Detection
Fast Training
CLUSTERING
K-MEANS
MULTI-CLASS CLASSIFICATION
Fast Training, Linear Model
Multi-Class Logistic Regression
Accuracy, Long Training Times
Multi-Class Neural Network
Accuracy, Fast Training
Multi-Class Decision Forest
Accuracy, Small Memory Footprint
Multi-Class Decision Jungle
Depends on Two-Class
One-V-All Multiclass
Start
REGRESSION
Ordinal Regression
Data in Rank Order 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 datasets
Neural Network Regression
Accuracy, Long Training Time
Decision Forest Regression
Accuracy, Fast Training
Boosted Decision Tree Regression
Accuracy, Fast Training, large Memory
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
Accuracy, Fast Training
Two-Class Decision Forest
Accuracy, Fast Training, LargeM
Two-Class Boosted Decision Tree
Accuracy, SmallM
Two Class Decision Jungle
>100 Features
Two Class Locally Deep SVM
Accuracy, Long Training Times
Two Class Neural Network



Predicting Categories
Outcome has multiple possibilities? Customer categories etc?
Start
MULTI-CLASS CLASSIFICATION
Fast Training, Linear Model
Multi-Class Logistic Regression
Accuracy, Long Training Times
Multi-Class Neural Network
Accuracy, Fast Training
Multi-Class Decision Forest
Accuracy, Small Memory Footprint
Multi-Class Decision Jungle
Depends on Two-Class
One-V-All Multiclass
TWO-CLASS CLASSIFICATION

Microsoft

Data Processing with Azure ML



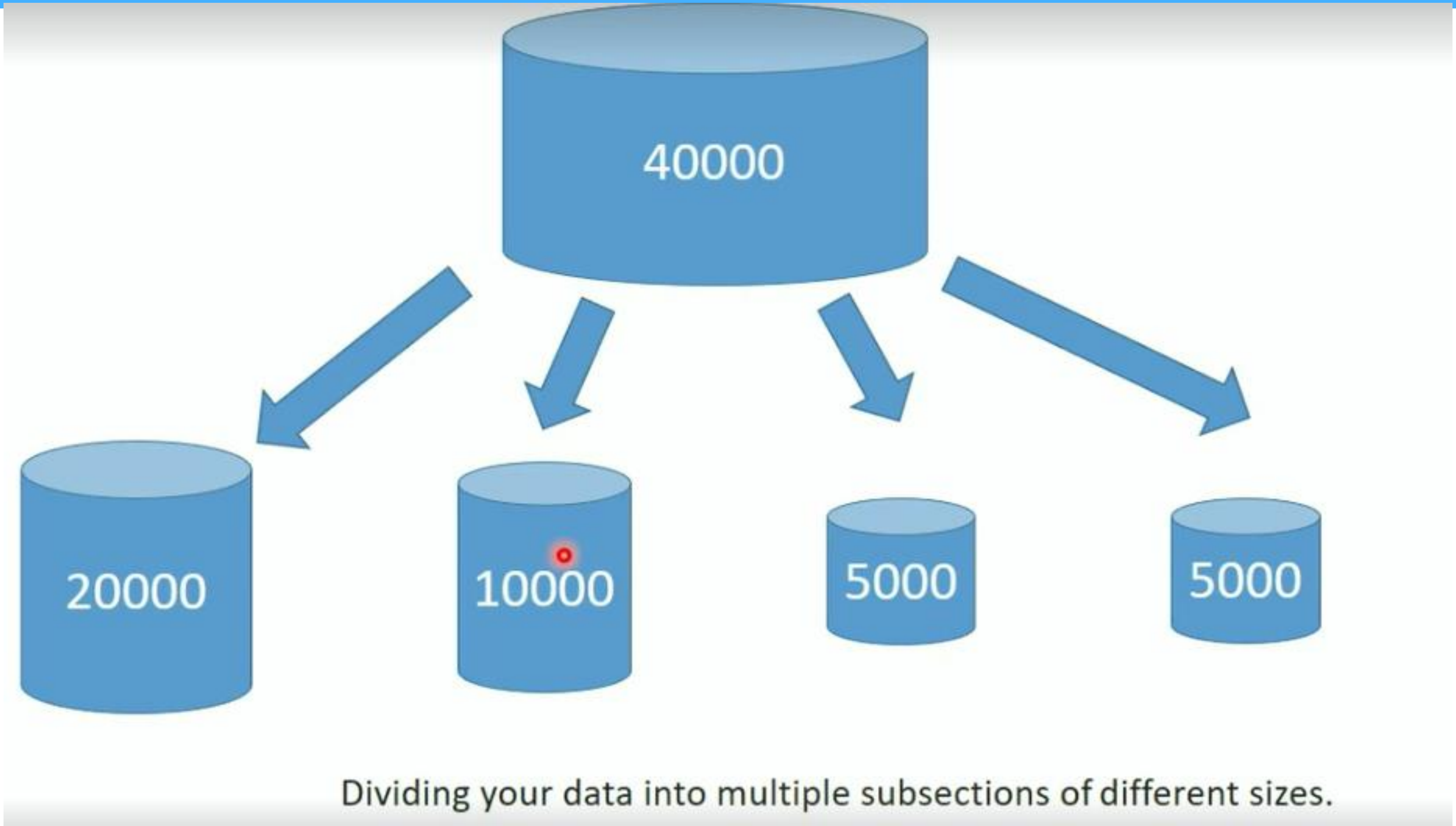
Data Input Output to Azure Workspace

- Enter Data Manually
- Upload a Dataset
- Data Format Conversion
- Import Data
- Unpacked Zipped Dataset

Lets make the hands dirty

Demo -1. Data Preparation in Azure ML

Sample and Split Data



Why to Partition Data

- The input dataset is huge and have too many observations
- Partition the data to create train and test datasets
- Create different bins of data for cross validation of your results
- Create a random sample of observations or a more balanced dataset for a particular column

Demo on Partitioning

Without Stratification

Rating	Number of Observations
Rating 3	20
Rating 5	5
Total	25

Sampling rate 0.4 or 40% — No Stratification

Rating	Number of Observations
Rating 3	10
Rating 5	0
Total	10

Rating	Number of Observations
Rating 3	5
Rating 5	5
Total	10

Rating	Number of Observations
Rating 3	7
Rating 5	3
Total	10

With Stratification

Rating	Number of Observations
Rating 3	20 x 0.4
Rating 5	5 x 0.4
Total	25

Sampling rate 0.4 or 40% — With Stratification

Rating	Number of Observations
Rating 3	8
Rating 5	2
Total	10

Classification in Azure ML

Logistic Regression

What is Logistics Regression?

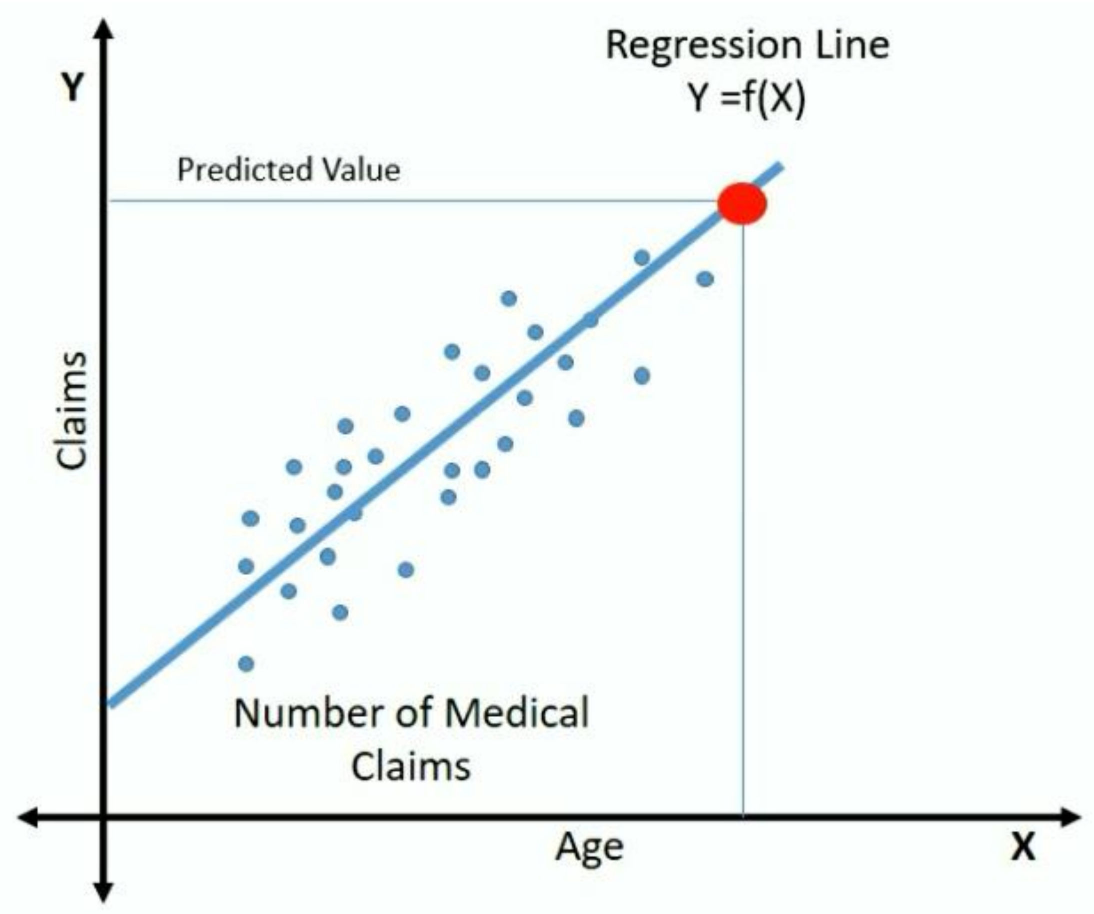
- Used to predict the probability of an outcome
- Can be binary - Yes/No or Multiple
- Supervised learning method
- Must provide a dataset that already contains the outcomes to train the model.

Understanding the Logistic Regression

Y = f(x)

Y = b0 + b1X

No of claims = 18 + b1(age)

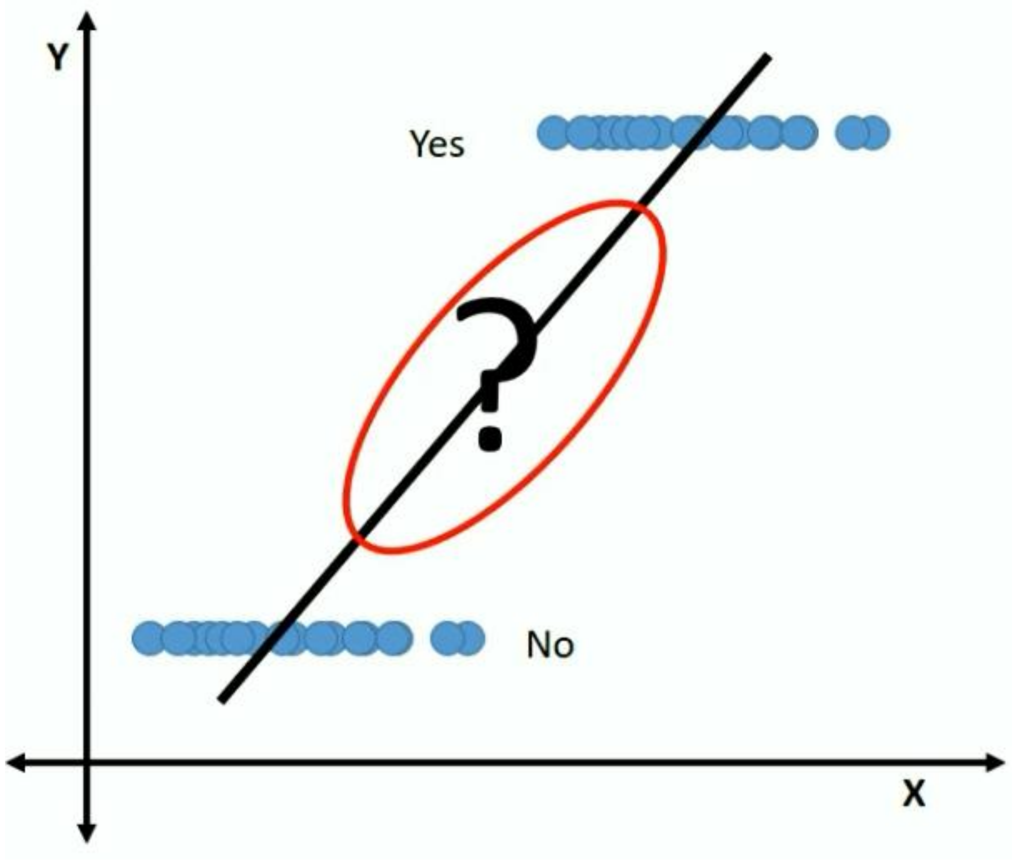


Simple Linear Regression

Logistic Regression?

- Outcome is categorical

- What is the probability of this customer buying this product?



Logistic Regression?

Probability needs to satisfy two basic conditions

- Always positive i.e. > 0
- Always less than or equal to 1

$$Y = b0 + b1\,X \xrightarrow{\text{To make it positive}} \exp(Y) = \exp\,(b0 + b1X)$$

To make it less than 1 divide it with a number greater than P

Logistic Regression?

$$\frac{\exp(Y)}{\exp(Y) + 1} = \frac{\exp(b0 + b1X)}{\exp(b0 + b1X) + 1}$$

$$P = \frac{e^{y}}{e^{(y)} + 1}$$ probability of success

Logistic Regression?

$$P = \frac{e^y}{e^{(y)} + 1}$$ probability of success

$$Q = 1 - P = 1 - \frac{e^y}{e^{(y)} + 1} = \frac{e^{(y)} + 1 - e^y}{e^{(y)}+1} = \frac{1}{e^{(y)} + 1}$$

Probability of Failure

Logistic Regression?

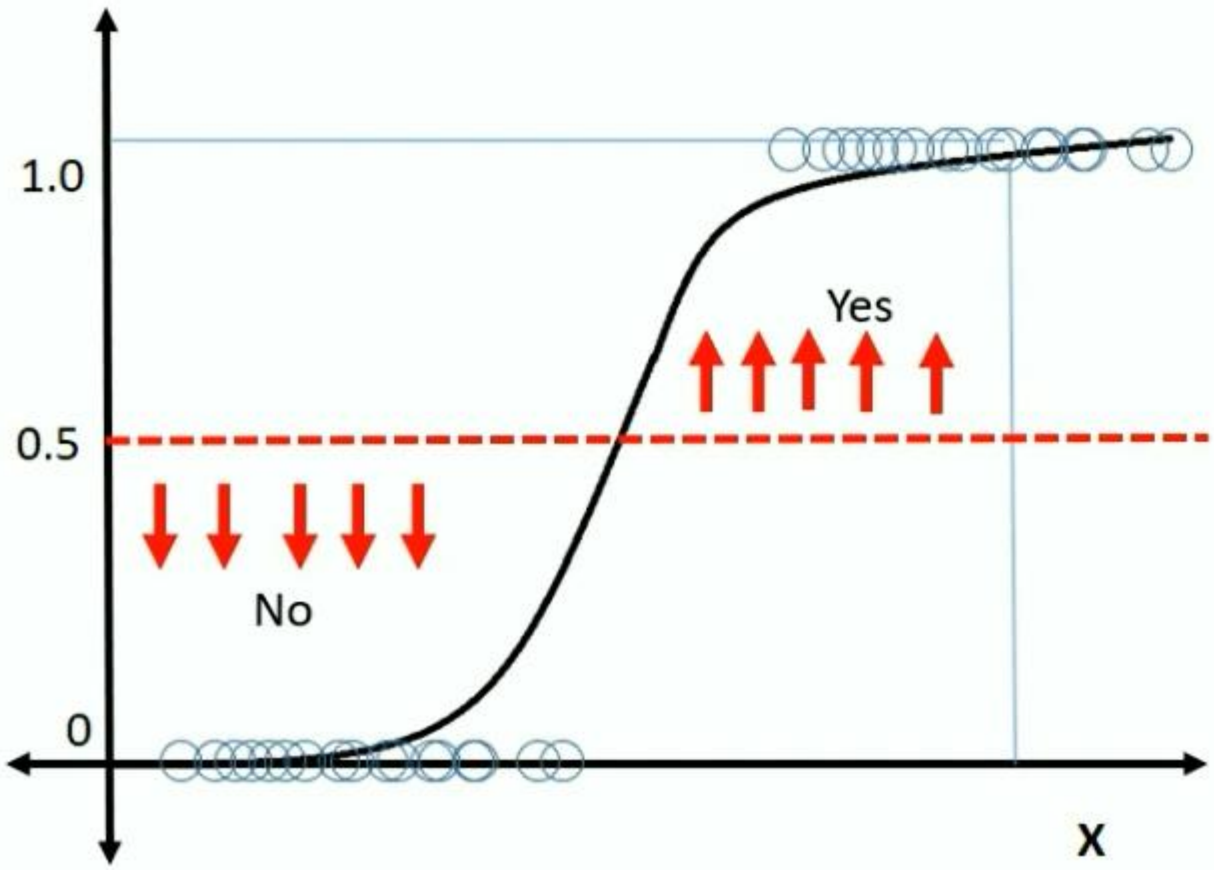
$$\frac{P}{1-P} = e^Y$$

$$\text{Log}\ \frac{P}{1-P} = Y$$

$$\text{Log}\ \frac{P}{1-P} = b0 + b1X$$

Plotting Logistic Regression?

$$\text{Log} \frac{P}{1 - P} = b0 + b1X$$



Threshold Value of 50% probability
> 0.5 Success
< 0.5 Failure

Lets make the hands dirty

Demo -2. Loan Eligibility in Azure ML

Problem Statement

- Automate loan eligibility process
- Identify customers whose loan will be approved

Loan_ID	Gender	Married	Dependents	Self_Employed	Income	LoanAmt	Term	CreditHistory	Property_Area	Status
LP001002	Male	No	0	No	$5,849.00		60	1	Urban	Y
LP001003	Male	Yes	1	No	$4,583.00	$128.00	120	1	Rural	N
LP001005	Male	Yes	0	Yes	$3,000.00	$66.00	60	1	Urban	Y
LP001006	Male	Yes	2	No	$2,583.00	$120.00	60	1	Urban	Y

Logistic Regression in Azure ML

What are Hyperparameters?

Parameters to Logistic Regression

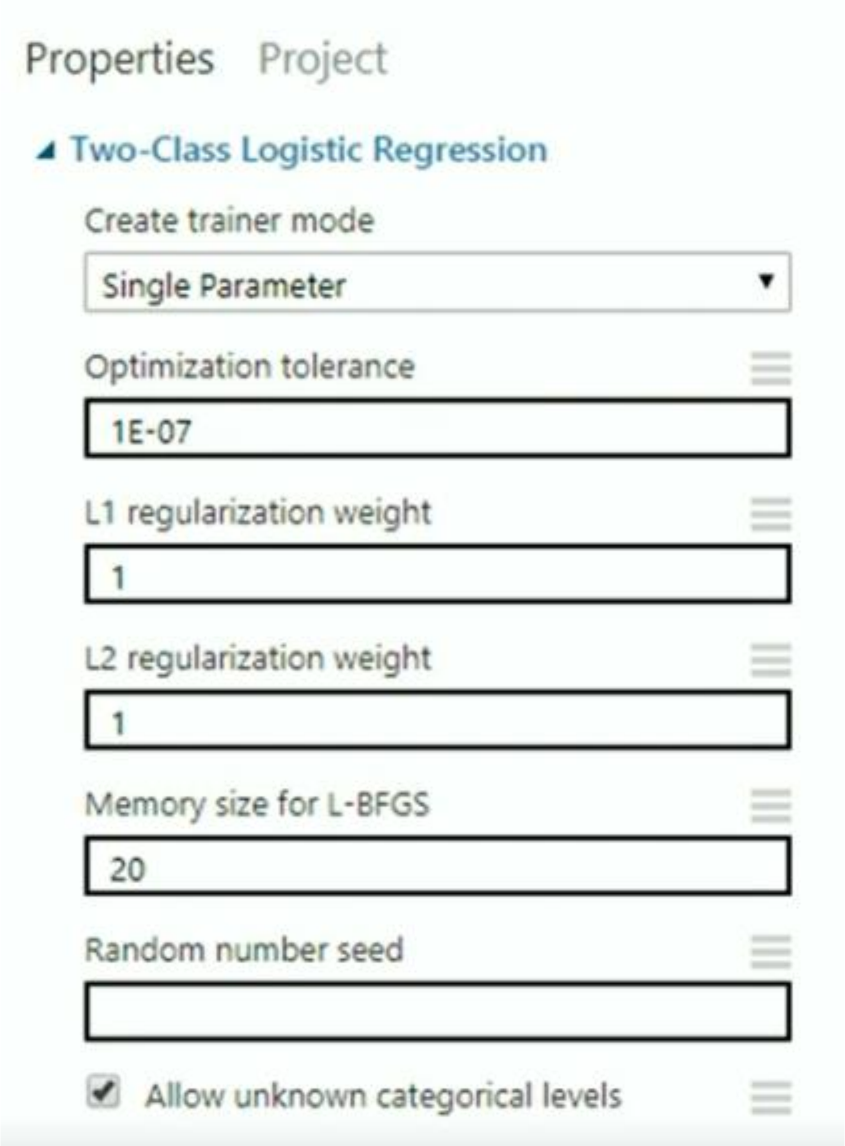
- Create Trainer Mode
  - Single Parameter - Provide specific set of values
  - Parameter Range - specify multiple values and get the optimum set for given configuration

- Optimization Tolerance - Threshold Value to stop the model iterations on trained dataset

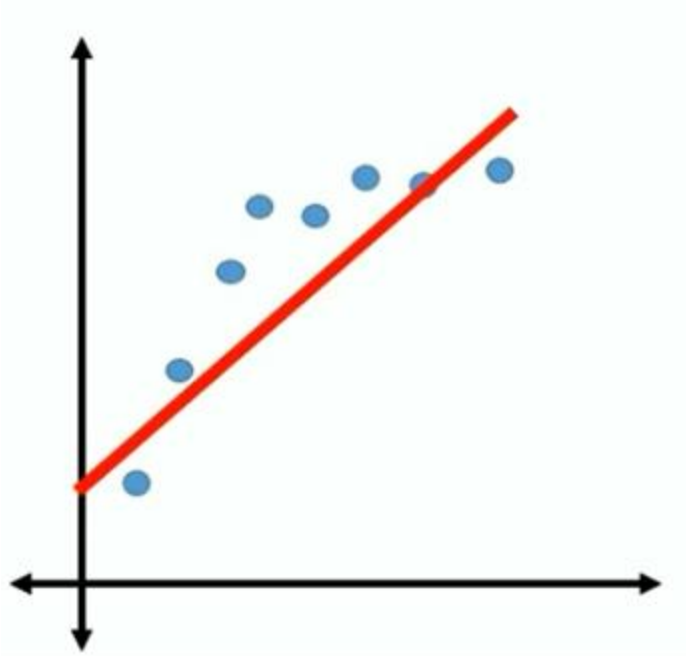
- Memory Size for L-BFGS - Amount of memory to use for next steps and direction

- Random Number Seed - Random integer number that is used for reproducing the same results

- Allow Unknown Categorical Levels - Creates an additional "Unknown" level

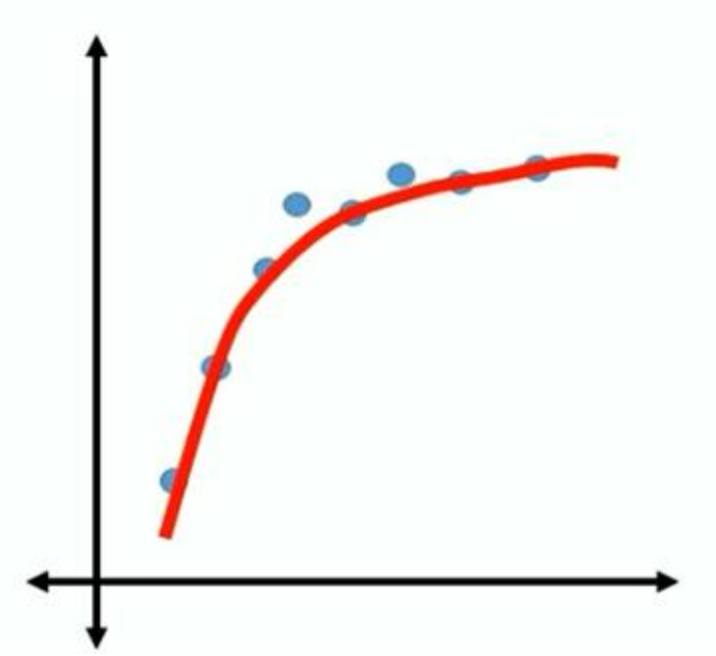


Regularization Weight



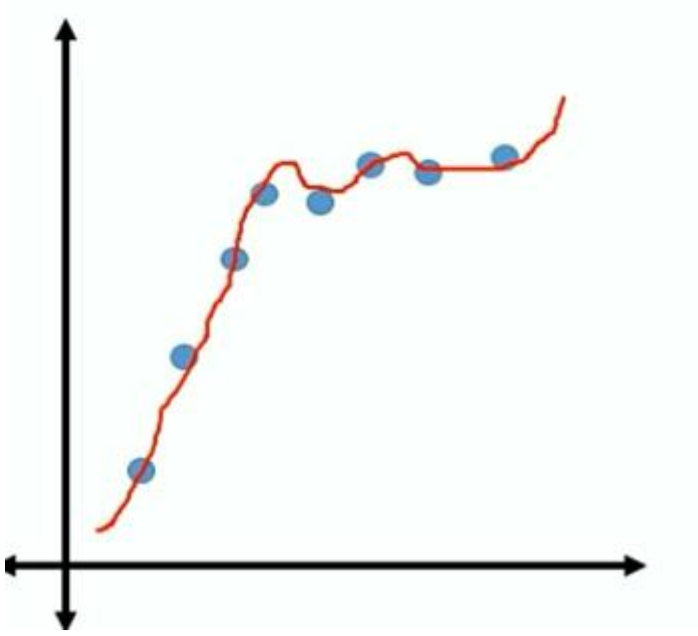
$b0 + b1X$

Under-fit



$b0 + b1X1^{2}$

Right



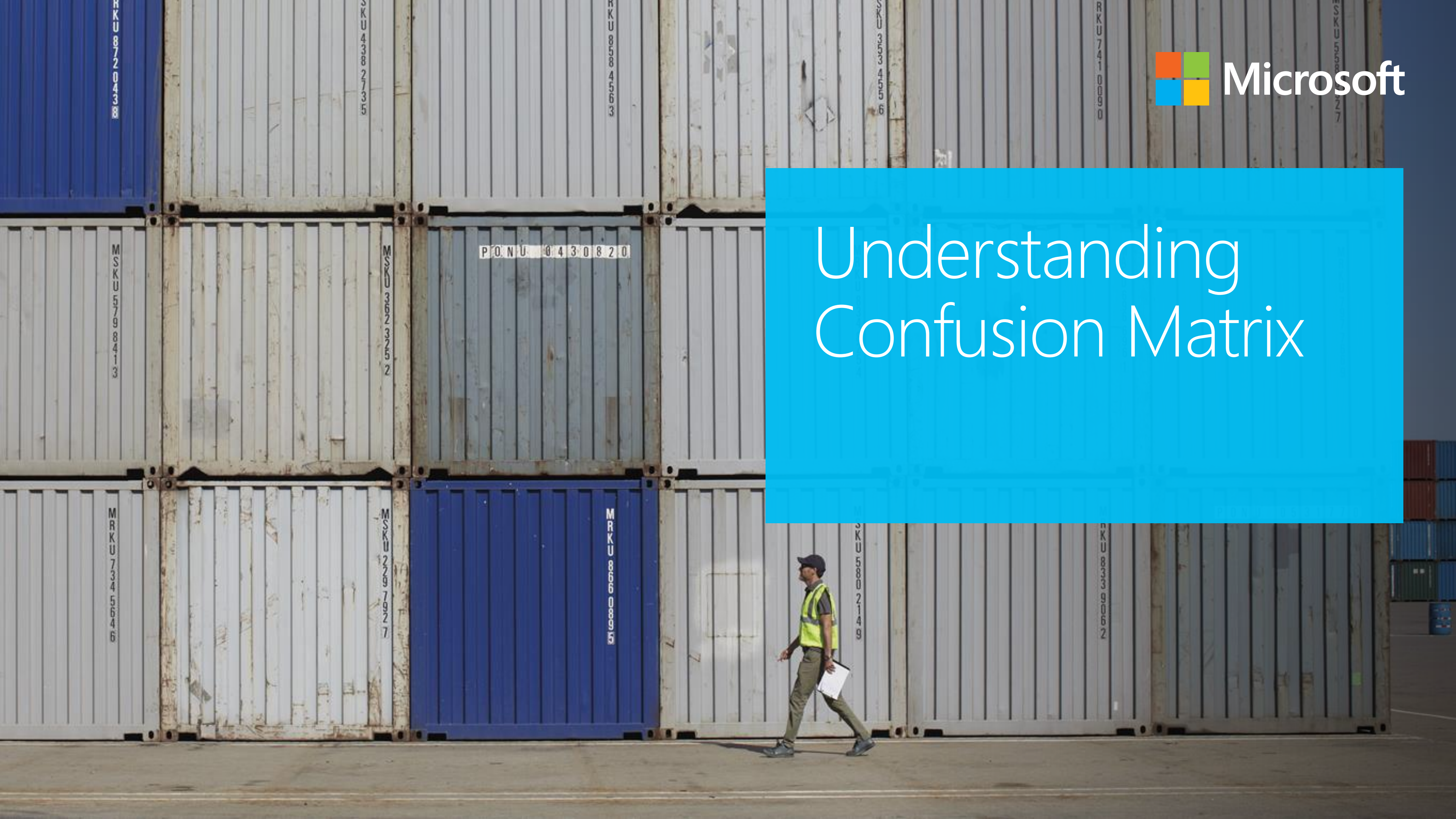
$b0 + b1X1^{2} + b2X2^{3}$

Over Fit

What if the effect of such weights is reduced significantly
Or reduced to zero

Regularization Weight

- L2 (Ridge) shrinks all the coefficient by the same proportions but eliminates none
- L1(Lasso) can shrink some coefficients to zero, performing variable selection.
- Both L1 and L2 regularization prevents overfitting by shrinking (imposing a penalty) on the coefficients.
- L2 penalizes one big weight more than many small weights.
- With L2, you tend to end up with many small weights, while with L1, you tend to end up with larger weights, but more zeros.



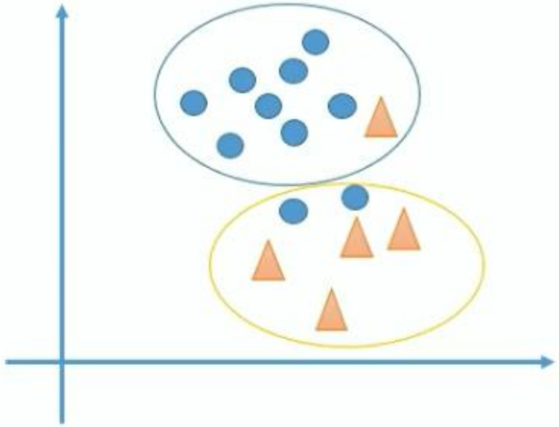
Microsoft

Understanding Confusion Matrix

Prediction Outcome

	Predicted Positives	Predicted Negatives
Actual Positives	True Positives	False Negatives
Actual Negative	False Positives	True Negatives

	Predicted Positives	Predicted Negatives	
Actual Positives	8	2	10
Actual Negative	1	4	5
	9	6	



Accuracy – Proportions of total number of correct results

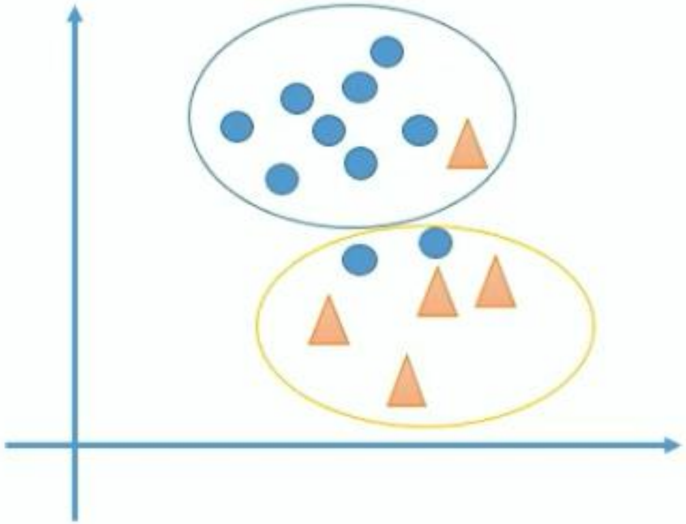
$$\text{Accuracy} = (8 + 4) / 15 = 0.8 \text{ or } 80\%$$

- **Accuracy:** Overall, how often is the classifier correct?

Prediction Outcome

	Predicted Positives	Predicted Negatives
Actual Positives	True Positives	False Negatives
Actual Negative	False Positives	True Negatives

	Predicted Positives	Predicted Negatives	
Actual Positives	8	2	10
Actual Negative	1	4	5
	9	6	



Precision – Proportion of correct positive results out of all predicted positive results

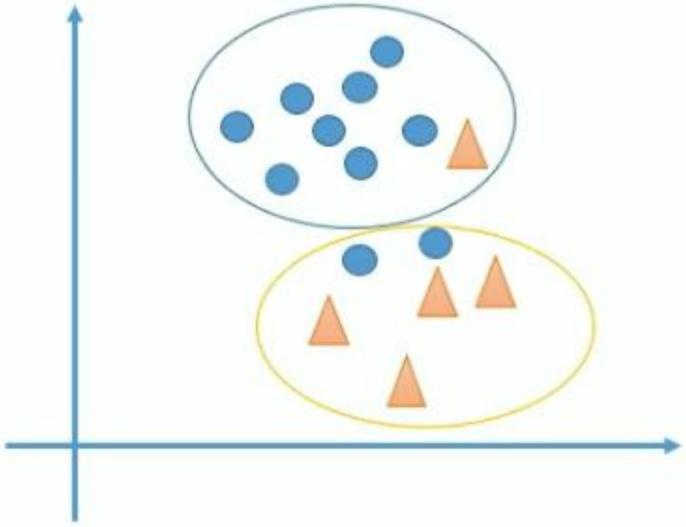
$$\text{Precision} = 8 / 9 = 0.889 \text{ or } 88.9\%$$

- **Precision:** When it predicts yes, how often is it correct?

Prediction Outcome

	Predicted Positives	Predicted Negatives
Actual Positives	True Positives	False Negatives
Actual Negative	False Positives	True Negatives

	Predicted Positives	Predicted Negatives	
Actual Positives	8	2	10
Actual Negative	1	4	5
	9	6	



Recall – Proportion of actual positive cases

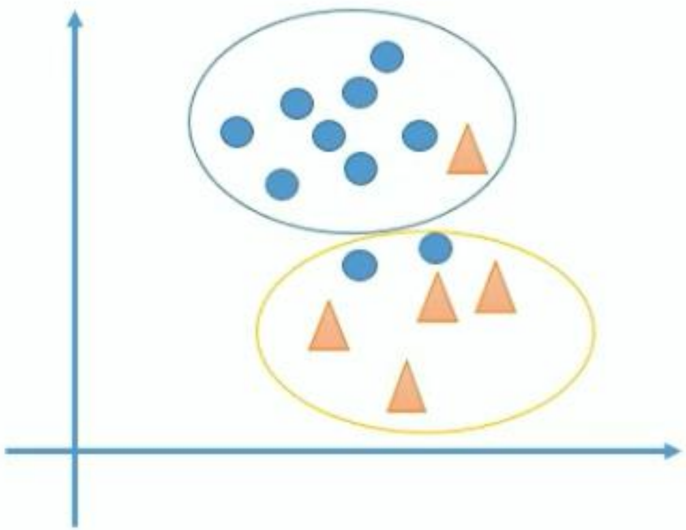
$$\text{Recall} = 8 / (8 + 2) = 0.8 \text{ or } 80\%$$

True Positive Rate: When it's actually yes, how often does it predict yes?
also known as "Sensitivity" or "Recall"

Prediction Outcome

	Predicted Positives	Predicted Negatives
Actual Positives	True Positives	False Negatives
Actual Negative	False Positives	True Negatives

	Predicted Positives	Predicted Negatives	
Actual Positives	8	2	10
Actual Negative	1	4	5
	9	6	



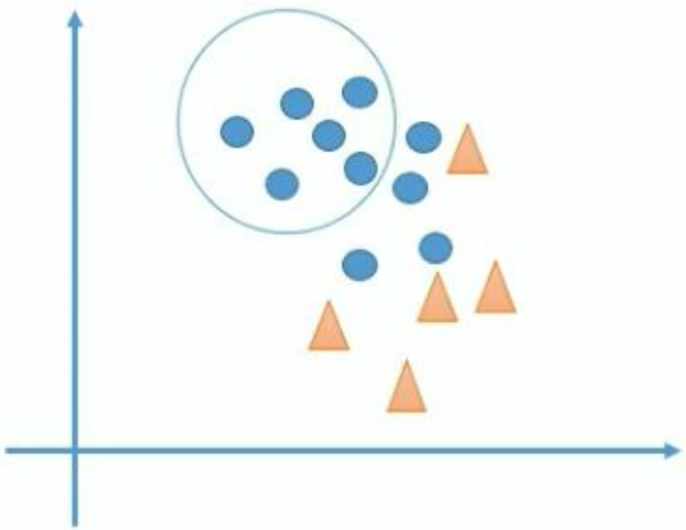
F1-Score – Weighted Average (Harmonic Mean) of Precision and Recall

F1Score = 2 * Precision * Recall / (Precision + Recall) = 0.84

Prediction Outcome

	Predicted Positives	Predicted Negatives
Actual Positives	True Positives	False Negatives
Actual Negative	False Positives	True Negatives

	Predicted Positives	Predicted Negatives	
Actual Positives	6	4	10
Actual Negative	0	5	5
	6	6	



In the Previous example

Precision = 6 / 6 = 1 or 100%

Recall = 6 / (6 + 4) = 0.6 or 60%

Precision = 0.889

Recall = 0.8

Average = 0.8

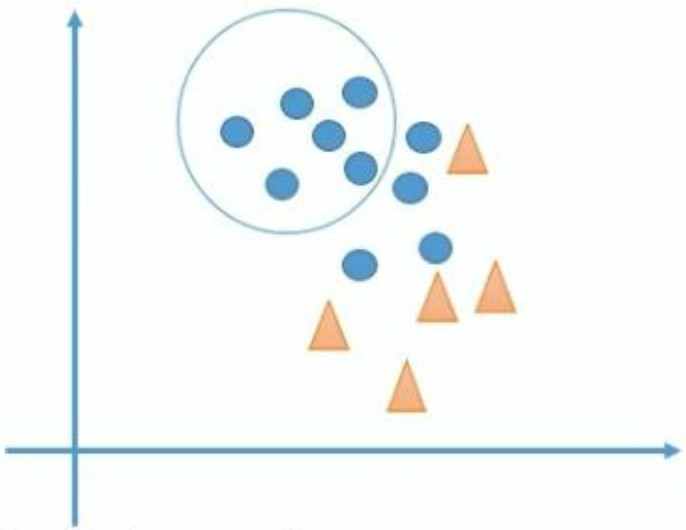
May lead to false interpretation

Average = 0.84

Prediction Outcome

	Predicted Positives	Predicted Negatives
Actual Positives	True Positives	False Negatives
Actual Negative	False Positives	True Negatives

	Predicted Positives	Predicted Negatives	
Actual Positives	6	4	10
Actual Negative	0	5	5
	6	6	



In the first example

Precision = 6 / 6 = 1 or 100%

Recall = 6 / (8 + 2) = 0.6 or 60%

F1Score = 0.75

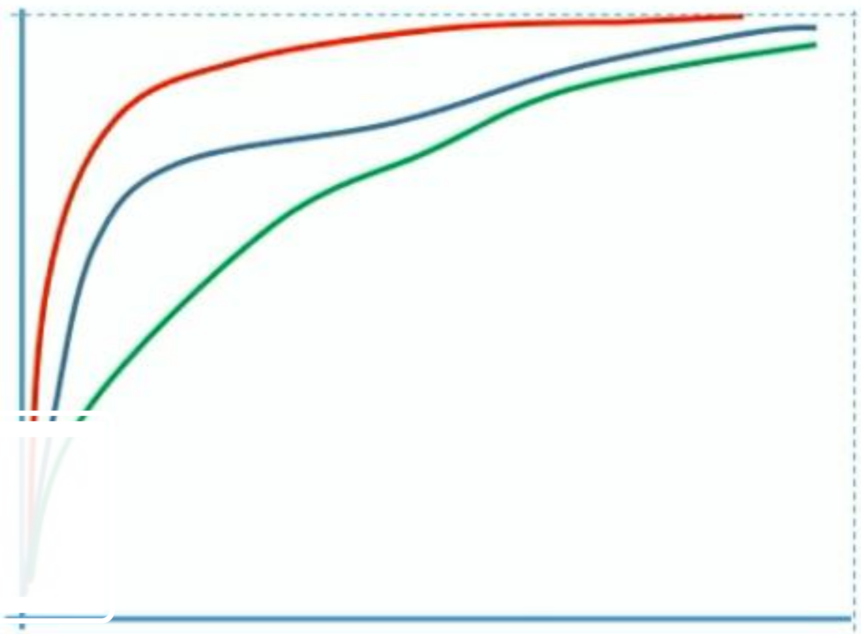
Precision = 0.889

Recall = 0.8

F1Score = 0.84

AUC ROC

True Positive Rate
TP / (TP + FN)



False Positive Rate
FP / (FP + TN)

AUC – Area Under the Curve

ROC – Receiver Operating Characteristics

First used during World War II for the analysis of radar signals

Following the attack on Pearl Harbor in 1941, the United States army began new research to increase the prediction of correctly detected Japanese aircraft from their radar signals.

For this purposes they measured the ability of radar receiver operators to make these important distinctions, which was called the Receiver Operating Characteristics

Provides a single number that lets you compare models of different types.

004 - Logistic Regression - Understanding the results - Excel

File | Home | Insert | Page Layout | Formulas | Data | Review | View | Add-Ins | Team | Tell me what you want to do... | Sign in | Share

H17 | f_x | 0.5

FPR = FP/(FP + TN)

Threshold	0.05		
TP	FN		
1019	33	TPR	0.96863
FP	TN	FPR	0.43342
3463	4527		

Threshold	0.03		
TP	FN		
1042	10	TPR	0.99049
FP	TN	FPR	0.64431
5148	2842		

Threshold	0.01		
TP	FN		
1051	1	TPR	0.99905
FP	TN	FPR	0.91277
7293	697		

Threshold	0.1		
TP	FN		
913	139	TPR	0.86787
FP	TN	FPR	0.19312
1543	6447		

Threshold	0.2		
TP	FN		
692	360	TPR	0.65779
FP	TN	FPR	0.08811
704	7286		

Threshold	0.31		
TP	FN		
524	528	TPR	0.4981
FP	TN	FPR	0.04781
382	7608		

Threshold	0.4		
TP	FN		
404	648	TPR	0.38403
FP	TN	FPR	0.03179
254	7736		

Threshold	0.5		
TP	FN		
317	735	TPR	0.30133
FP	TN	FPR	0.0219
175	7815		

Threshold	0.6		
TP	FN		
249	803	TPR	0.23669
FP	TN	FPR	0.01452
116	7874		

Threshold	0.7		
TP	FN		
179	873	TPR	0.17015
FP	TN	FPR	0.00914
73	7917		

Threshold	0.8		
TP	FN		
122	930	TPR	0.11597
FP	TN	FPR	0.00526
42	7948		

Threshold	0.9		
TP	FN		
60	992	TPR	0.05703
FP	TN	FPR	0.00275
22	7968		

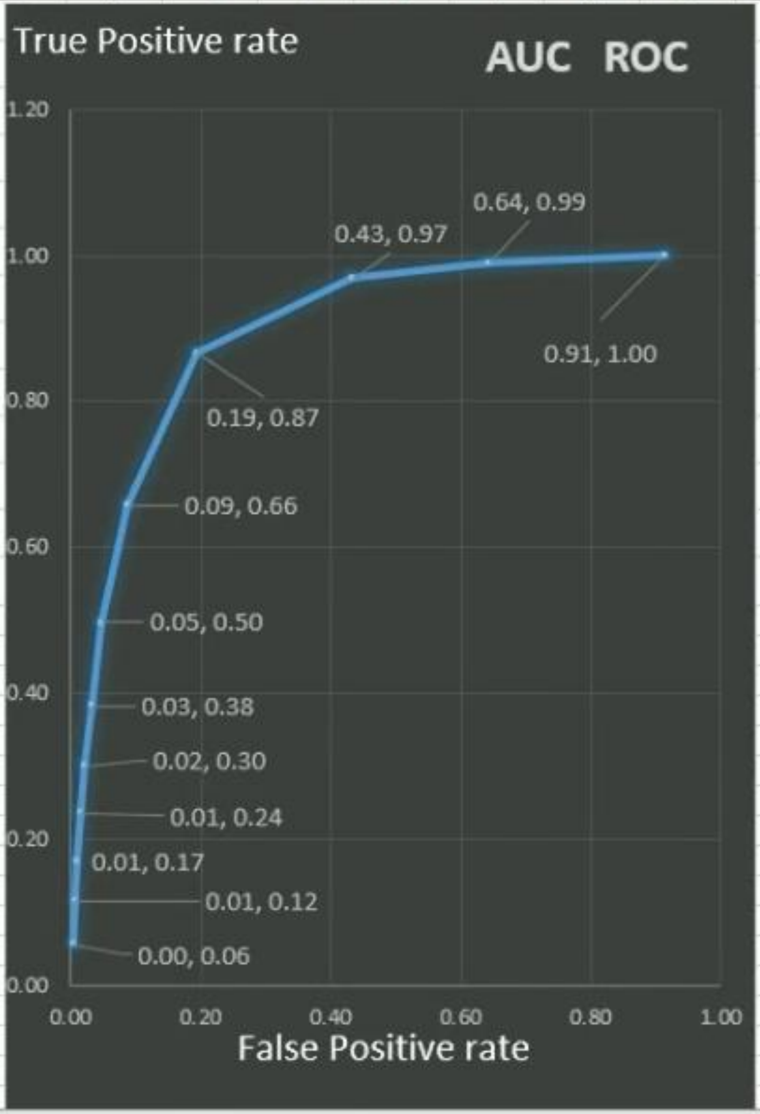
Row: 3

Data | AUC Plot

Ready | 120%

004 - Logistic Regression - Understanding the results - Excel

Threshold	FPR	TPR
0.9	0.00	0.06
0.8	0.01	0.12
0.7	0.01	0.17
0.6	0.01	0.24
0.5	0.02	0.30
0.4	0.03	0.38
0.3	0.05	0.50
0.2	0.09	0.66
0.1	0.19	0.87
0.05	0.43	0.97
0.03	0.64	0.99
0.01	0.91	1.00



Data | AUC Plot

Ready

Impact Analysis

Multiple Model Analysis

- Scenario 1 - Impact of Split Percentage
- **Scenario 2 - Impact of Stratification**
- Scenario 3 - Low L1 (0.0001) and High L2 (1)
- Scenario 4 - High L1 (1) and Low L2 (0.0001)
- Scenario 5 - High L1 (1) and High L2 (1)
- Scenario 6 - Low L1 (0.0001) and Low L2 (0.0001)

Multiclass Logistic Regression

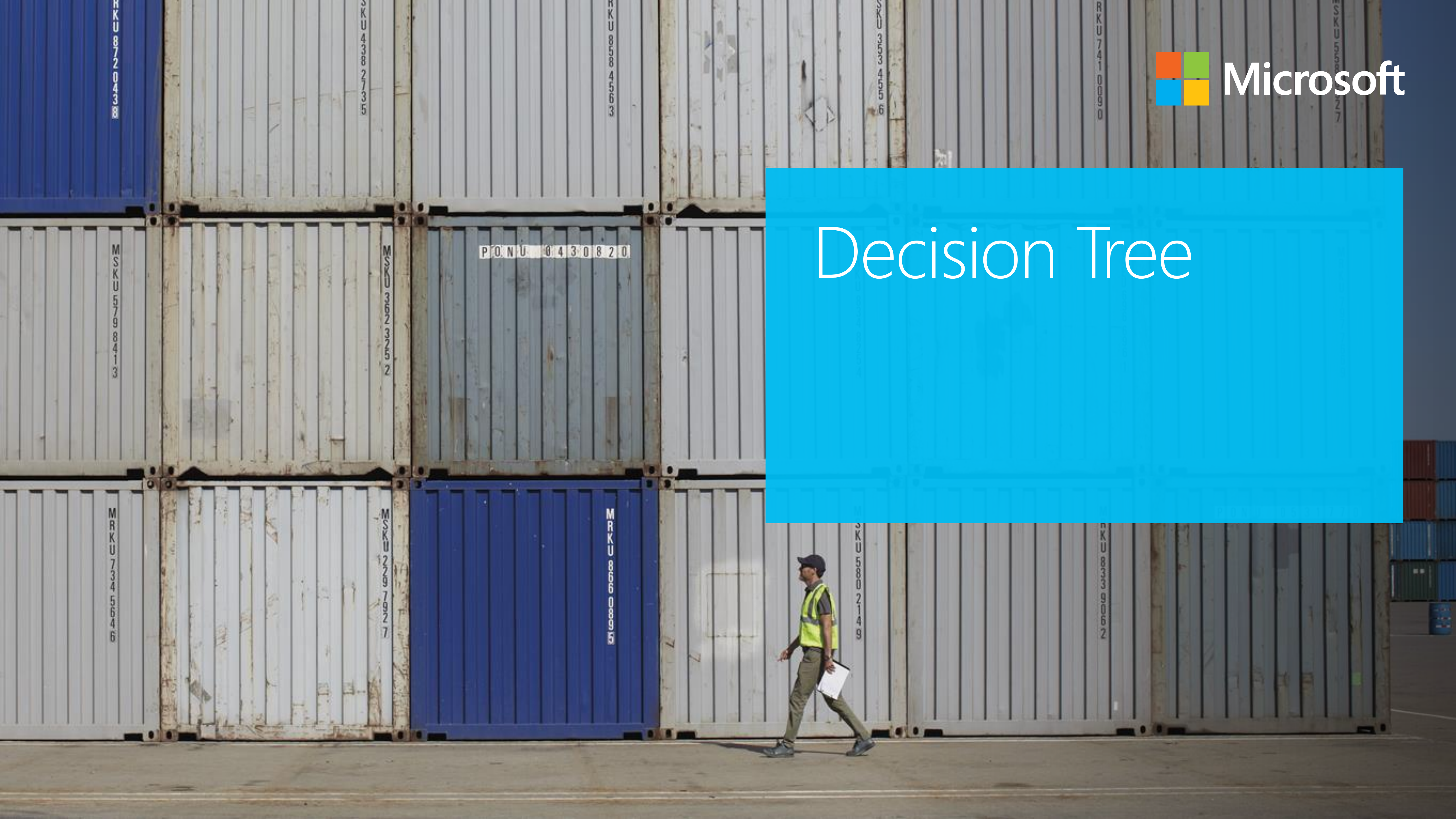
Wine Quality Prediction

- Fixed and Volatile Acidity
- Citric acid
- Residual sugar
- Chlorides
- Free and Total Sulphur dioxide
- Density
- pH
- Sulphates
- Alcohol Content

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.
Modelling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553,2009

Lets make the hands dirty

Demo -3. Wine Quality in Azure ML



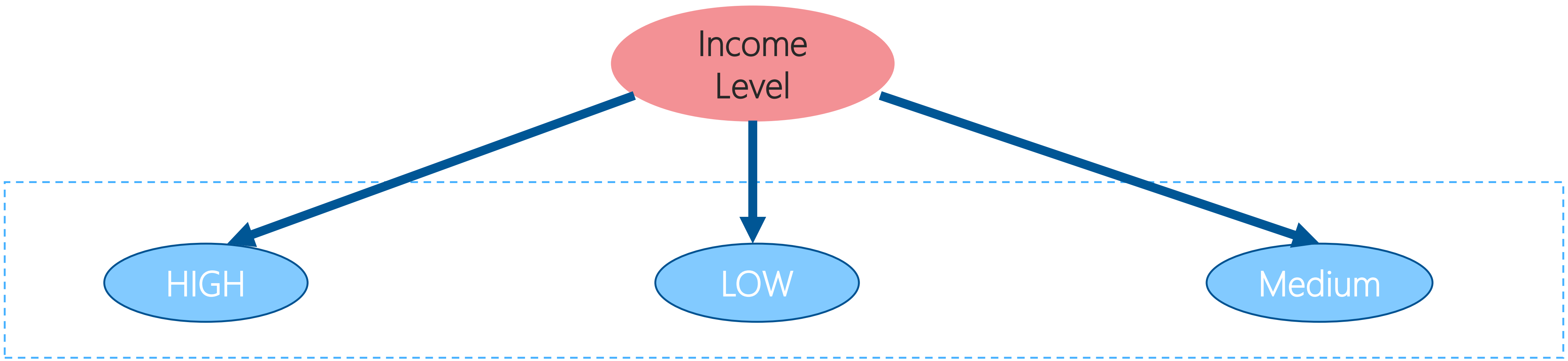
Microsoft

Decision Tree

What is Decision Tree?

- Supervised learning method
- Decision support tool that uses a tree-like graph or model of decisions and their possible consequences
- Various variations such as Boosted Decision Tree, Decision Forest, Decision Jungle
- Can be used for categorical as well as continuous variables

Loan ID	Income Level	Credit Score	Employment	Approved?
L1	Medium	Low	Self-Employed	No
L2	High	Low	Self-Employed	Yes
L3	High	High	Salaried	Yes
L4	Medium	Low	Salaried	Yes
L5	Low	High	Salaried	No
L6	Low	Low	Self-Employed	No
L7	High	Low	Salaried	Yes
L8	Medium	Low	Self-Employed	No
L9	High	High	Self-Employed	Yes
L10	Medium	High	Self-Employed	Yes
L11	High	Low	Salaried	Yes
L12	Medium	High	Salaried	Yes
L13	Medium	High	Self-Employed	Yes



LID	IL	CS	ET	Status
L2	High	Low	SE	Yes
L3	High	High	Salaried	Yes
L7	High	Low	Salaried	Yes
L9	High	High	SE	Yes
L11	High	Low	Salaried	Yes

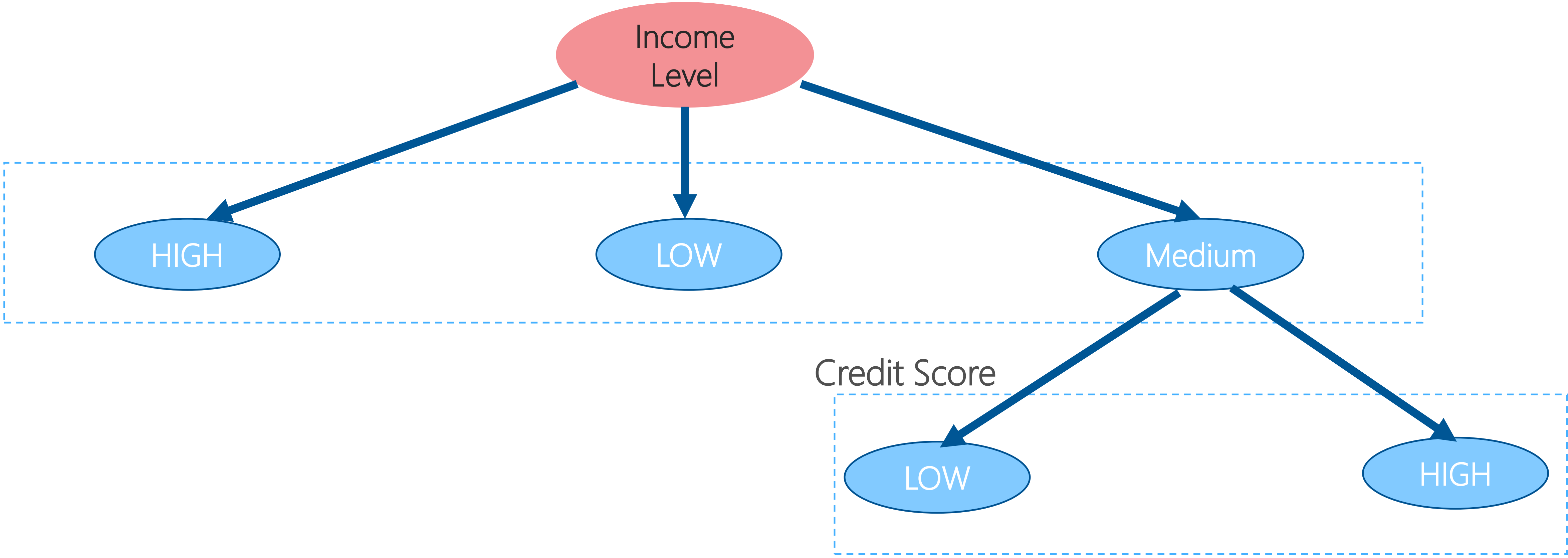
Pure Subset

LID	IL	CS	ET	Status
L5	Low	High	Salaried	No
L6	Low	Low	SE	No
L14	Low	Low	SE	No
L15	Low	High	SE	No

Pure Subset

LID	IL	CS	ET	Status
L1	Medium	Low	SE	No
L4	Medium	Low	Salaried	Yes
L8	Medium	Low	SE	No
L10	Medium	High	SE	Yes
L12	Medium	High	Salaried	Yes
L13	Medium	High	SE	Yes

Split Further

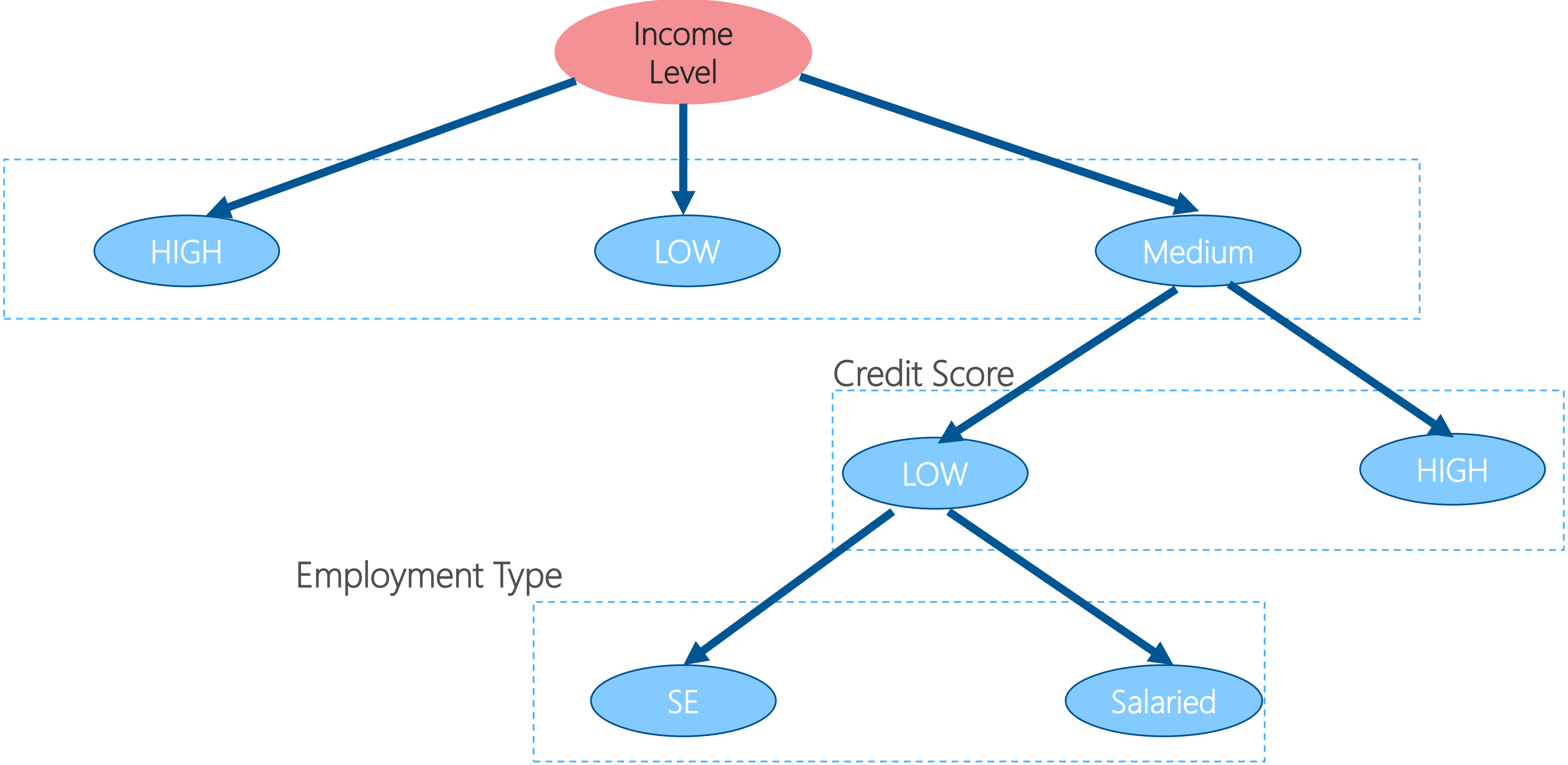


LID	IL	CS	ET	Status
L1	Medium	Low	SE	No
L4	Medium	Low	Salaried	Yes
L8	Medium	Low	SE	No

Split Further

LID	IL	CS	ET	Status
L10	Medium	High	SE	Yes
L12	Medium	High	Salaried	Yes
L13	Medium	High	SE	Yes

Pure Subset



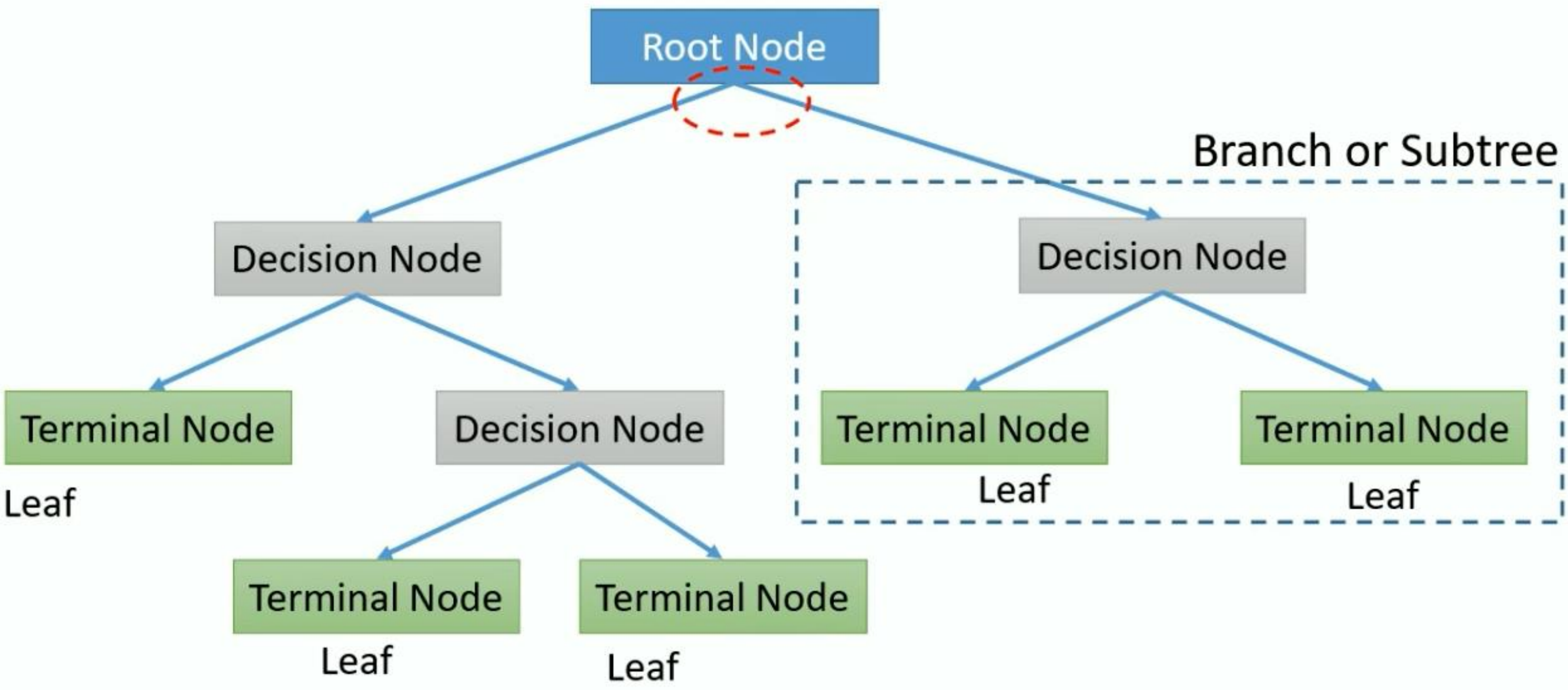
LID	IL	CS	ET	Status
L1	Medium	Low	SE	No
L8	Medium	Low	SE	No

Pure Subset

LID	IL	CS	ET	Status
L4	Medium	Low	Salaried	Yes

Pure Subset

Decision Tree Terms



Ensemble learning

Everyday Ensemble Learning

Decision?

Is this price fair?

Appreciation of price?

Construction Quality?

Location appropriate?

Neighbourhood?

Decision?

Broker or real estate portal to check fair price, price appreciation

Friend or colleague who stays nearby or stayed in the neighbourhood

Inspection by an architect for quality checks and structural defects.

Decision?

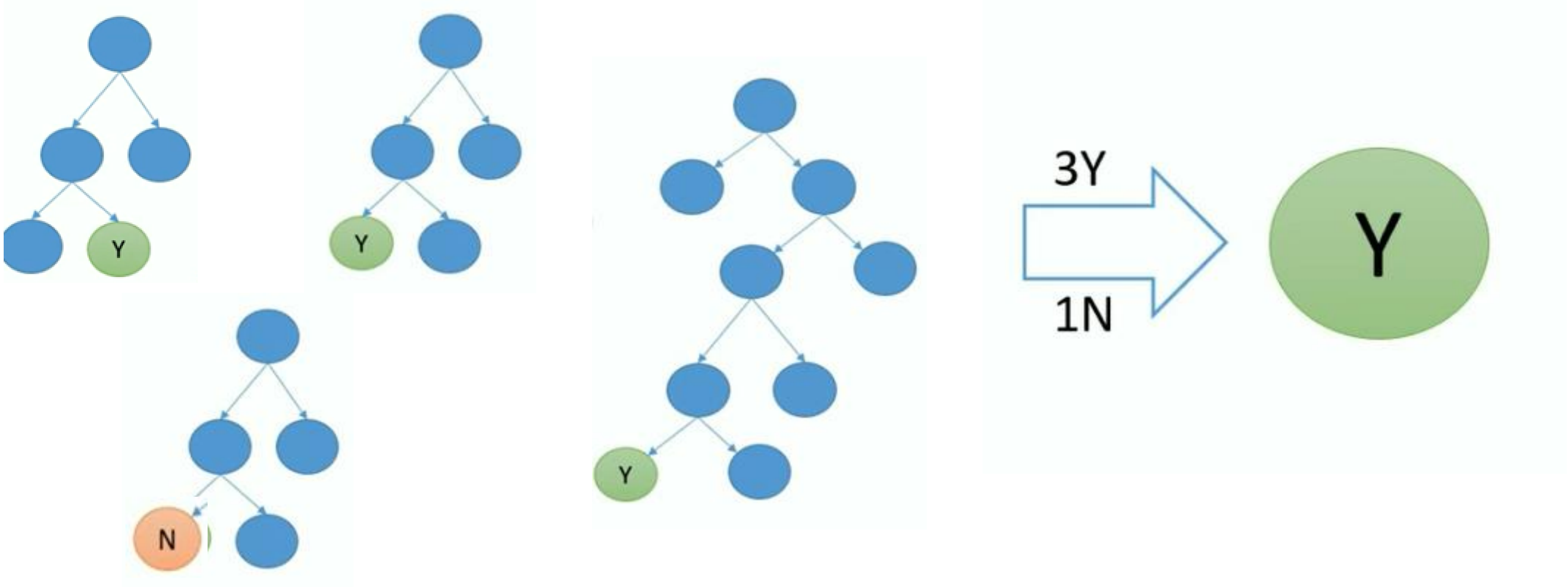

Is this price fair?
Appreciation of price?
Construction Quality?
Majority
Weighted Average
Neighbourhood?
Location appropriate?

Ensemble Learning

- All algorithms have errors
- Collective wisdom is higher than the individual intelligence
- Generate a group of base learners and combined result gives higher accuracy
- Different base learners can use different,
  - Parameters
  - Sequence
  - Training sets etc
- Two major Ensemble Learning Methods
  - Bagging
  - Boosting

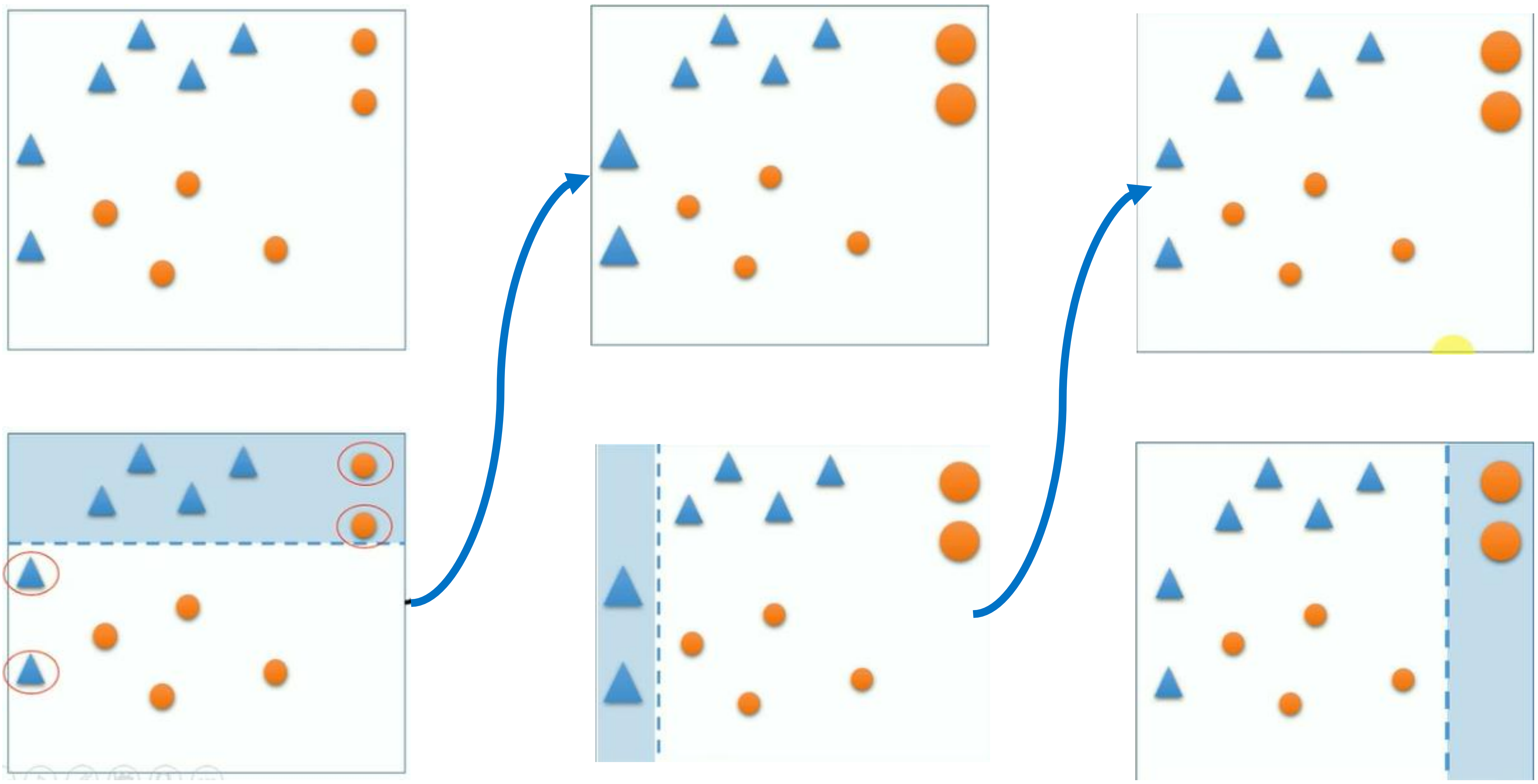
Bagging

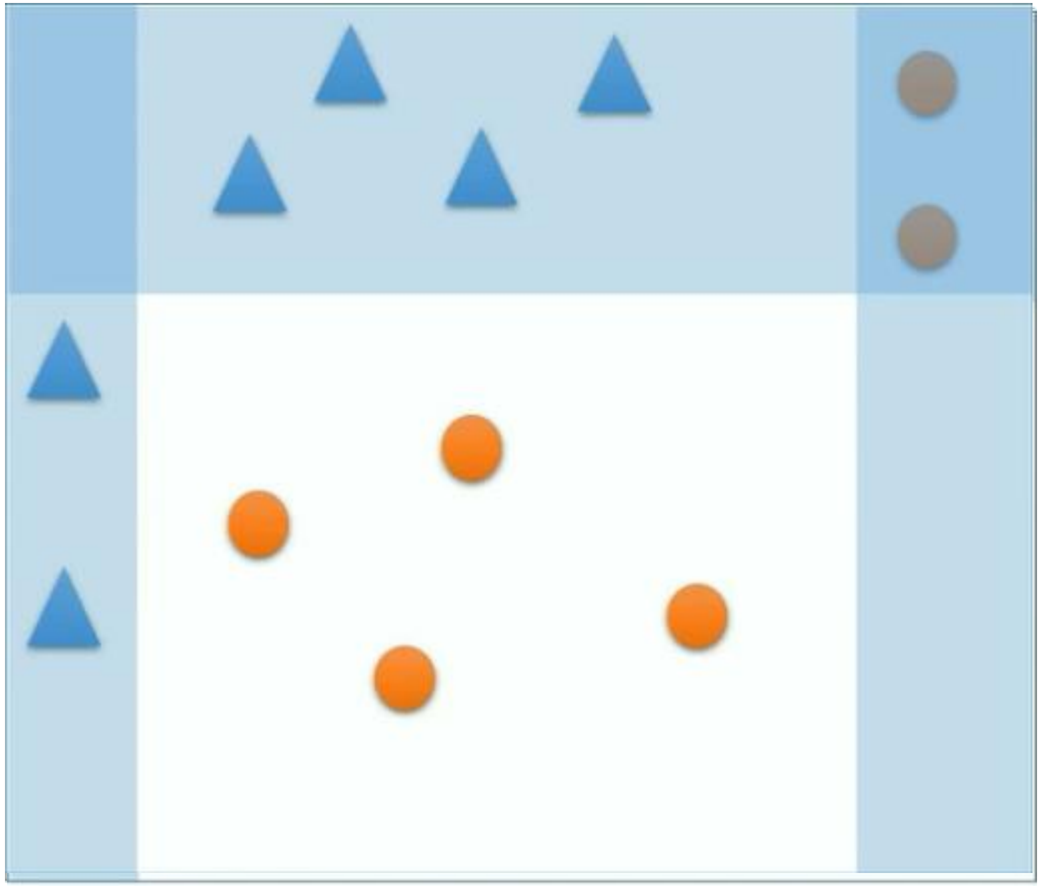
- Various models are built in parallel
- All models vote to give the final prediction



Boosting

- Train the Decision Tree in a sequence
- Learn from the previous tree by focussing on incorrect observations
- Build new model with higher weight for incorrect observations from previous sequence





Lets make the hands dirty

Demo -4. Two Class Boosted Decision Tree

Bank Telemarketing

- Goal is to predict if the client will subscribe to a product or not

Number of instances — 45, 211

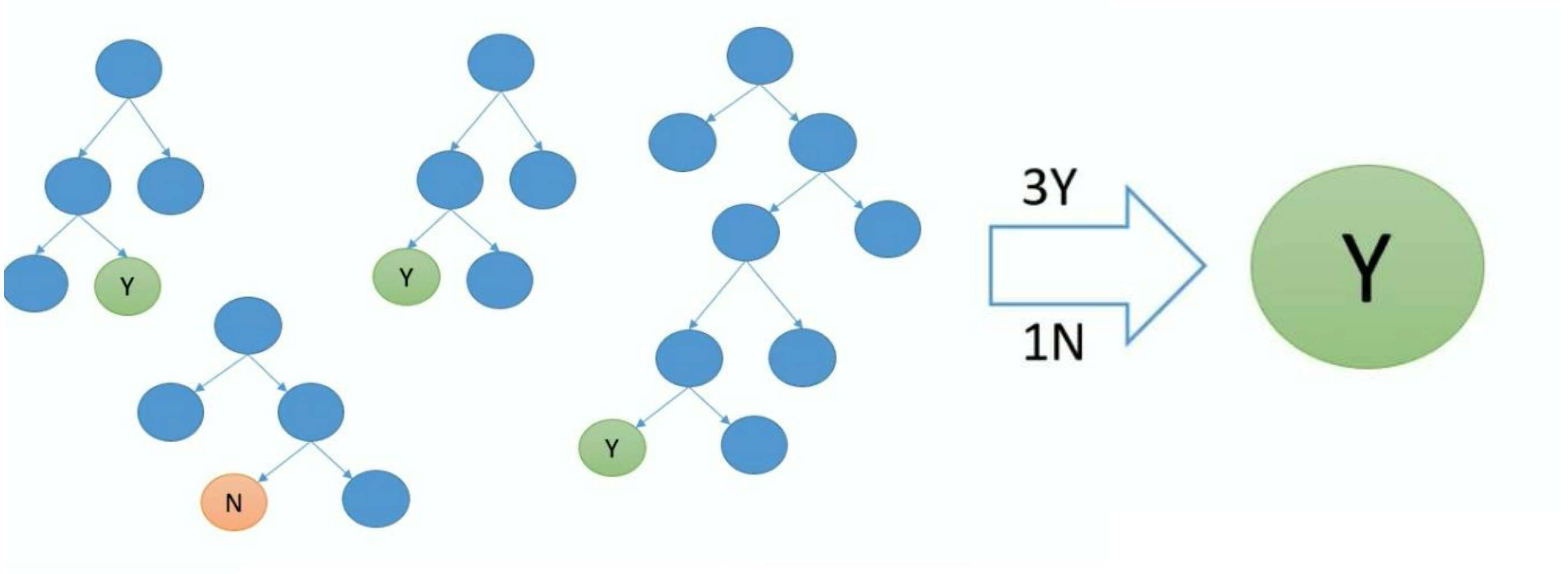
1. Age
2. Job Type
3. Marital Status
4. Education Level
5. Credit Default?
6. Housing Loan?
7. Personal Loan
8. Contacted Type
9. Contacted Month
10. Last Contacted day
11. Contact Duration
12. Campaign Type
13. P-Days
14. Previous
15. P-Outcome
16. Emp-Var-Rate
17. Consumer Price Index
18. Consumer Confidence Index
19. Euribor 3 Month Rate
20. Number of employees
21. Subscribed?

Two Class Boosted Decision Tree?

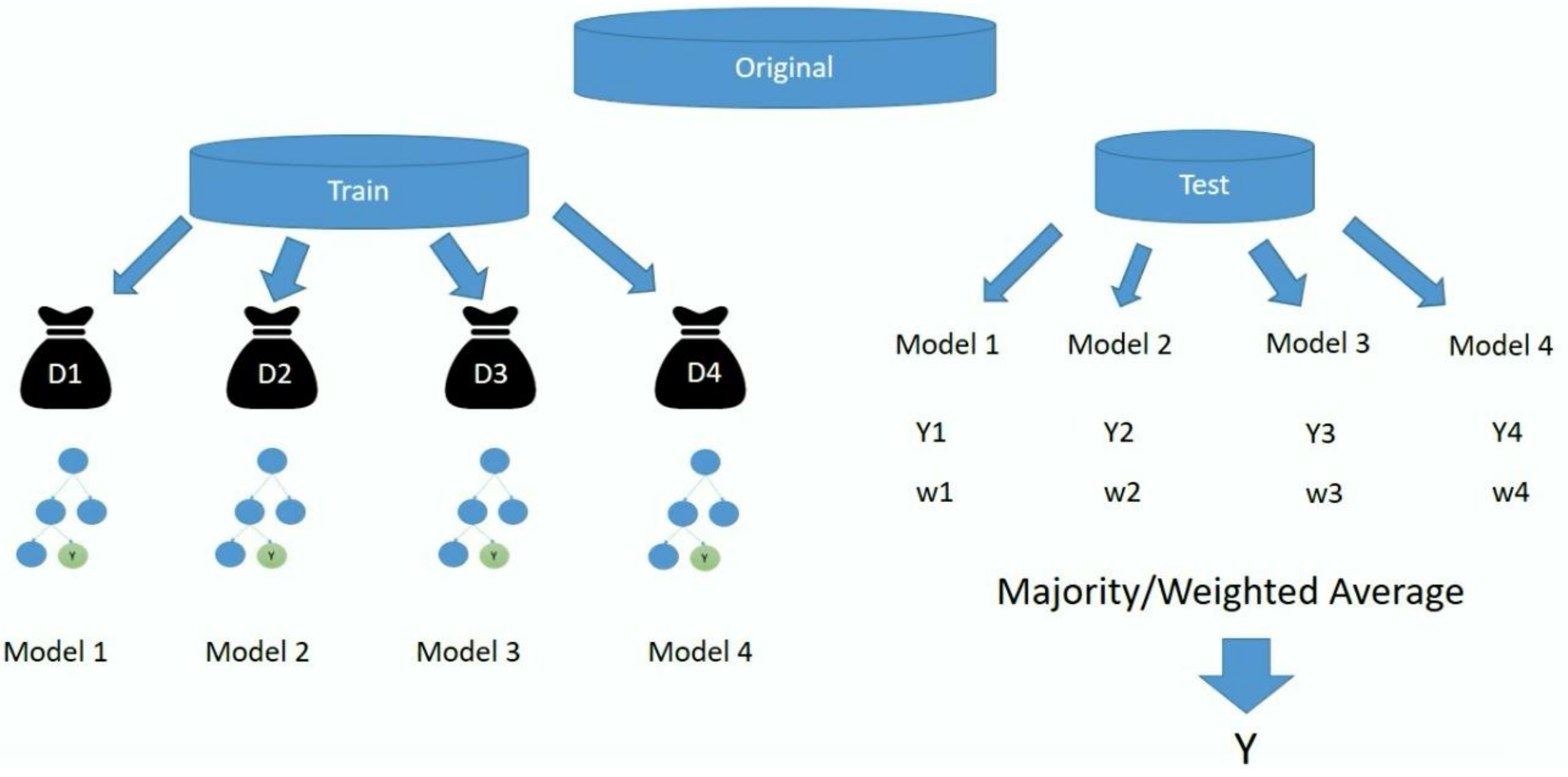
- Machine learning model based on the boosted decision trees algorithm
- Based on ensemble learning method
- Among the easiest methods to get top performance
- One of the more memory-intensive learners

Bagging

- Various models are built in parallel
- All models vote to give the final prediction



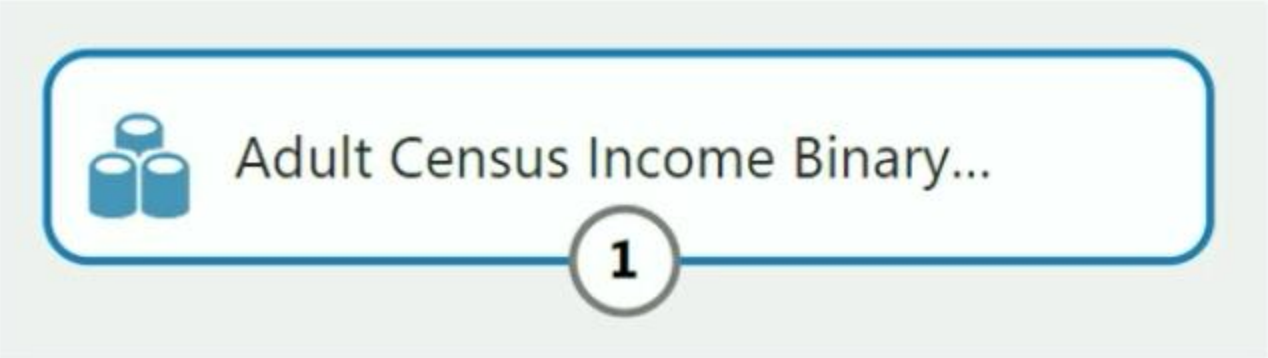
Bagging



Original
Train
Test
D1
D2
D3
D4
Y
Model 1
Model 2
Model 3
Model 4
Model 1
Model 2
Model 3
Model 4
Y1
Y2
Y3
Y4
w1
w2
w3
w4
Majority/Weighted Average
Y

Adult Census Data

Problem statement: Predict whether income exceeds $50K/yr based on census data



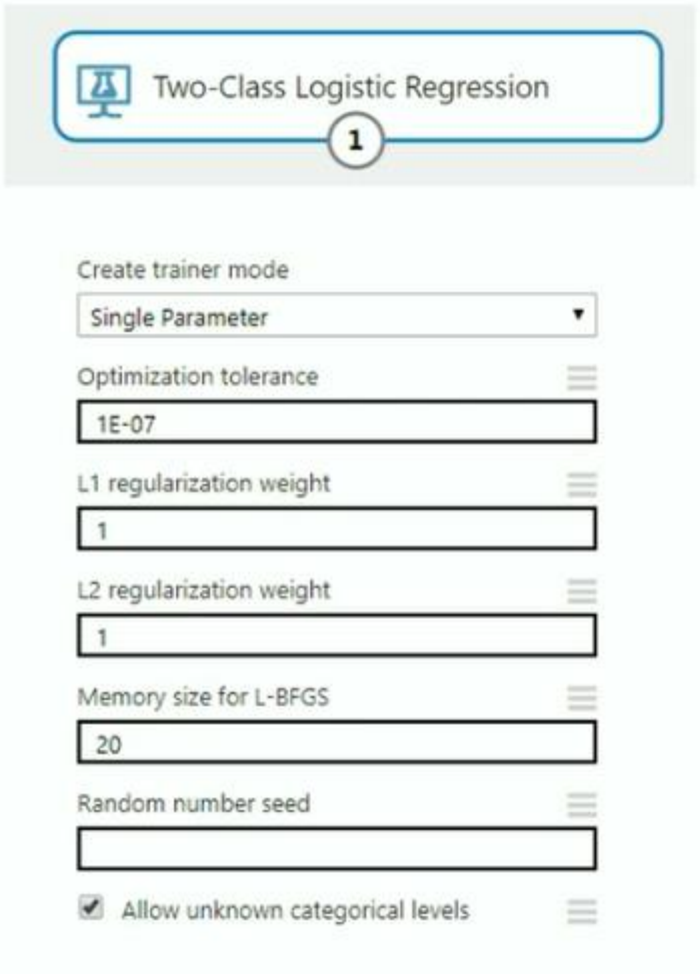
1. Age
2. Workclass
3. Fnlwgt
4. Education
5. Education-Num
6. Marital Status
7. Occupation
8. Relationship
9. Race
10. Sex
11. Capital Gains
12. Capital Losses
13. Hours per week
14. Native Country
15. Income

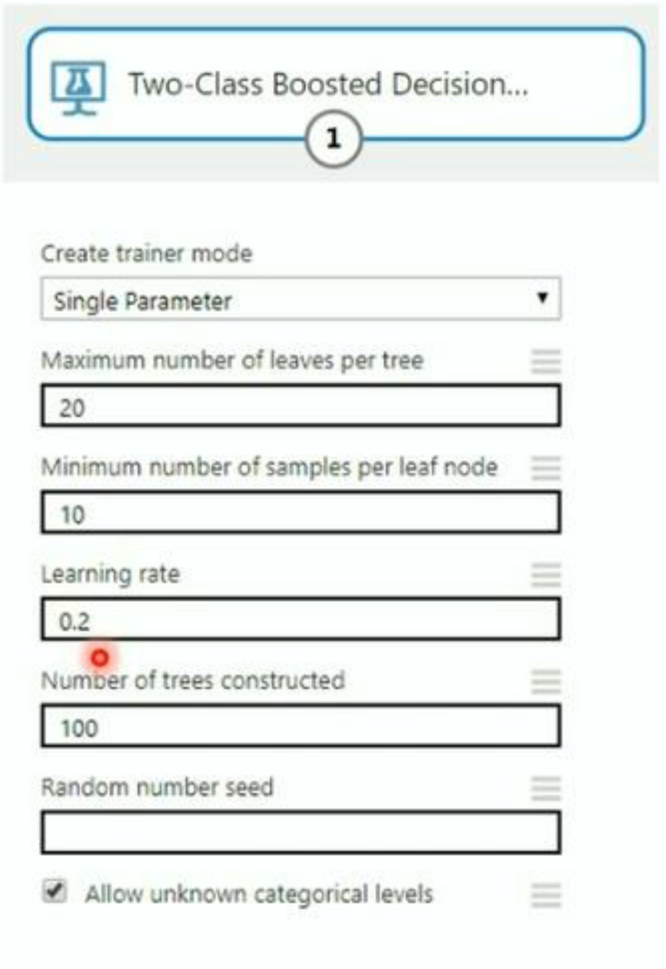
Demo -5. Decision Forest

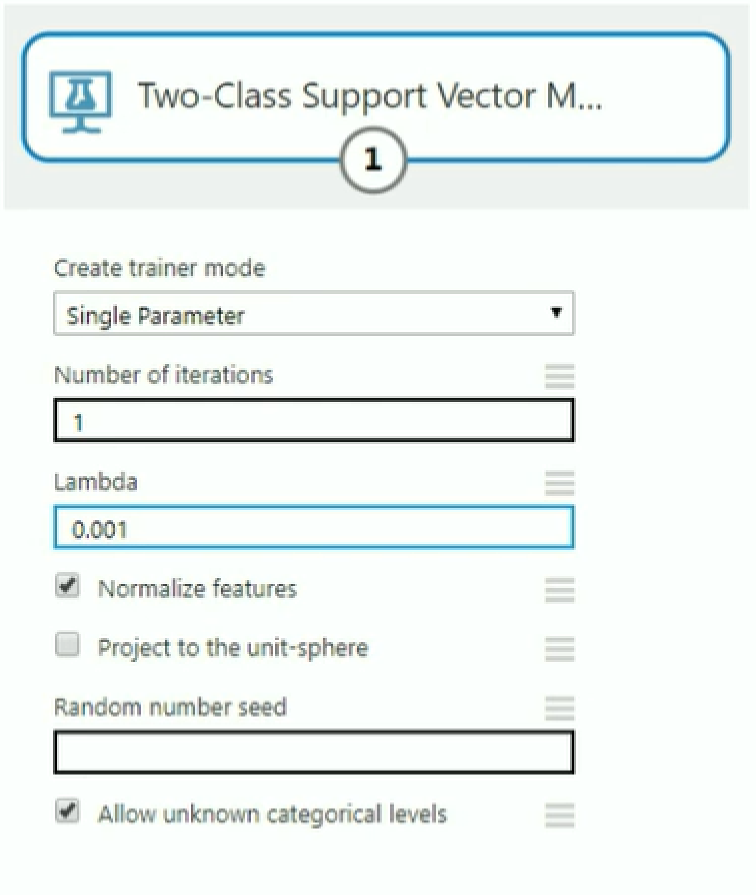
Demo -6. Support Vector Machine
&
Evaluate 2 Models

Tune Model Hyperparameters

Models and Parameters







What are Hyperparameters?

Model parameters?

- Decision Trees
  - Maximum number of leaves per tree
  - Minimum number of samples per leaf node
  - Learning rate
  - Number of trees to construct

- Logistic regression
  - Optimization tolerance
  - L1 regularization weight
  - L2 regularization weight
  - Memory size for L-BFGS

Tune Model Hyperparameters?

- Helps in determining the best possible combination of hyperparameters

- Also known as hyperparameter optimization

- Performance metric to measure
    - Accuracy (Overall, how often is the classifier correct?)
    - Precision (When it predicts yes, how often is it correct?)
    - Recall (When it's actually yes, how often does it predict yes?)
    - AUC
    - F1Score

Parameter Sweeping Modes

- Random Grid
- Entire Grid
- Random Sweep

What is a Grid?

- Cartesian Product of Parameters
- Parameter 1→ 1, 2, 3
- Parameter 2→ A, B, C, D

Parameter 1 →

↓ Parameter 2

	1	2	3
A	A, 1	A, 2	A, 3
B	B, 1	B, 2	B, 3
C	C, 1	C, 2	C, 3
D	D, 1	D, 2	D, 3

Random Grid

Parameter 1 →

↓ Parameter 2

	1	2	3
A	A, 1	A, 2	A, 3
B	B, 1	B, 2	B, 3
C	C, 1	C, 2	C, 3
D	D, 1	D, 2	D, 3

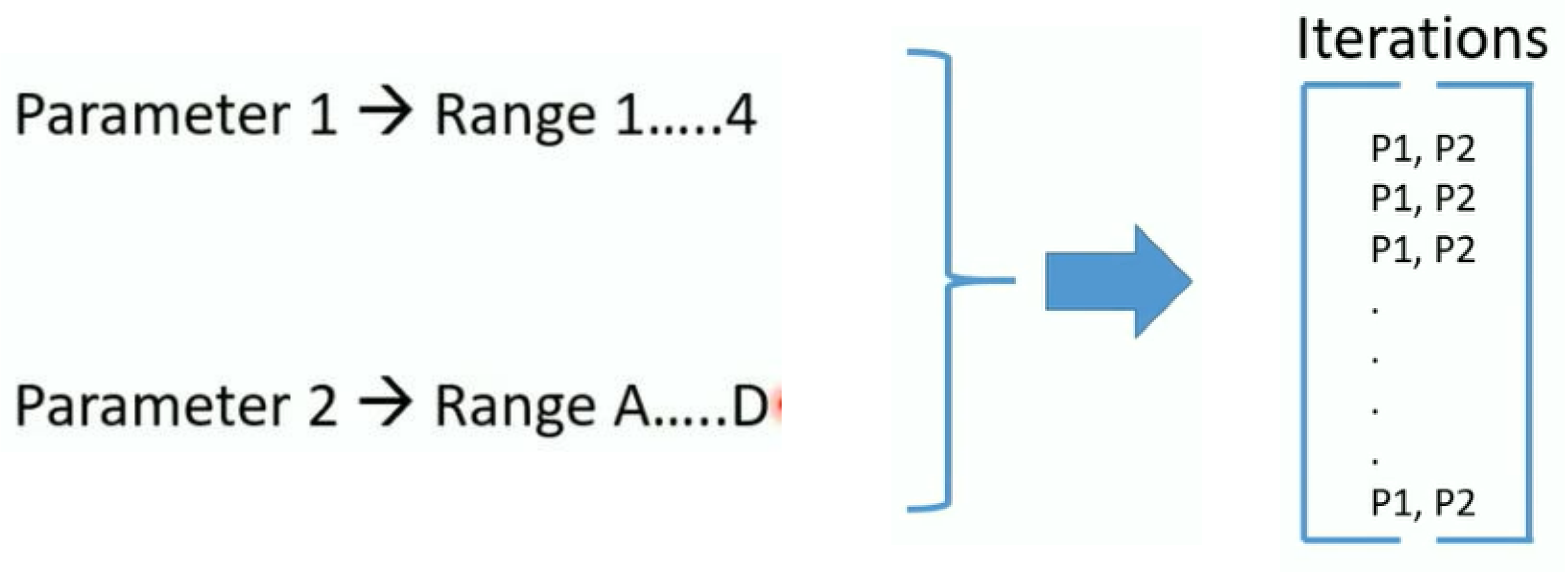
Entire Grid

Parameter 1 →

↓ Parameter 2

	1	2	3
A	A, 1	A, 2	A, 3
B	B, 1	B, 2	B, 3
C	C, 1	C, 2	C, 3
D	D, 1	D, 2	D, 3

Random Sweep

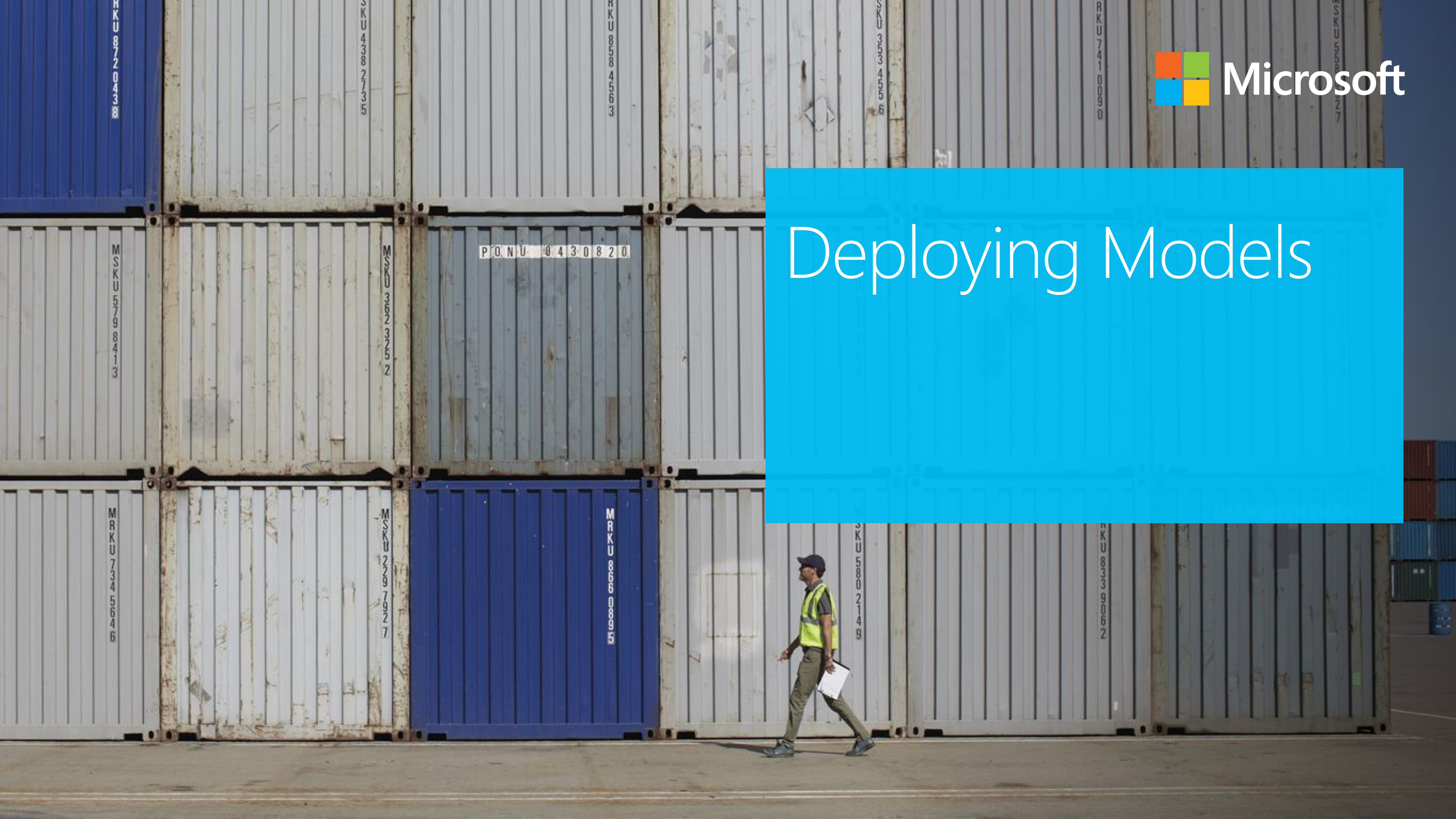


DEMOS

Tuning Hyper Parameters

Microsoft

Deploying Models



I have a Model. Now What?

- Develop but how to deploy?
- Model language is not supported
- Difficult to deploy in the current architecture
- Tedious Environment set-up
- Many more…

Steps for Azure ML webservices Deployment

Creating a Training Experiment

↓

Convert to predictive experiment

↓

Deploy it as a web-service

↓

Consume the web-service

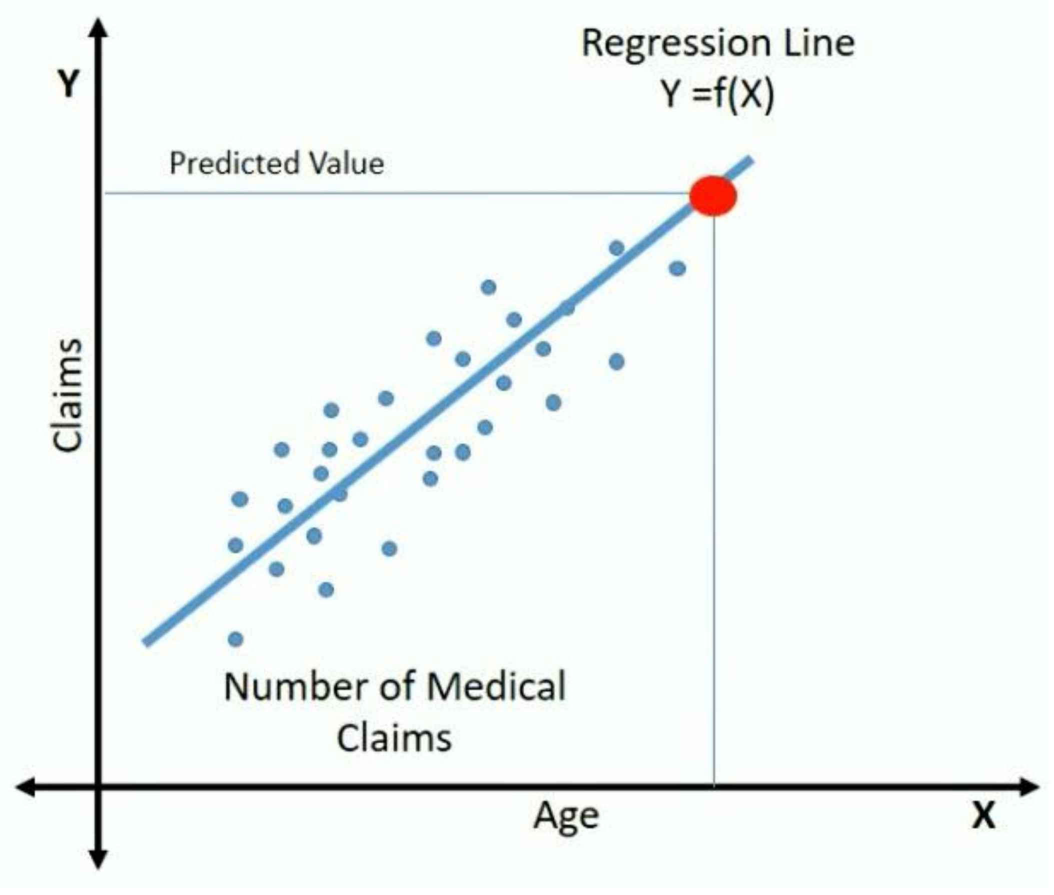
DEMOS
Deploying & Creating Webservice

Microsoft

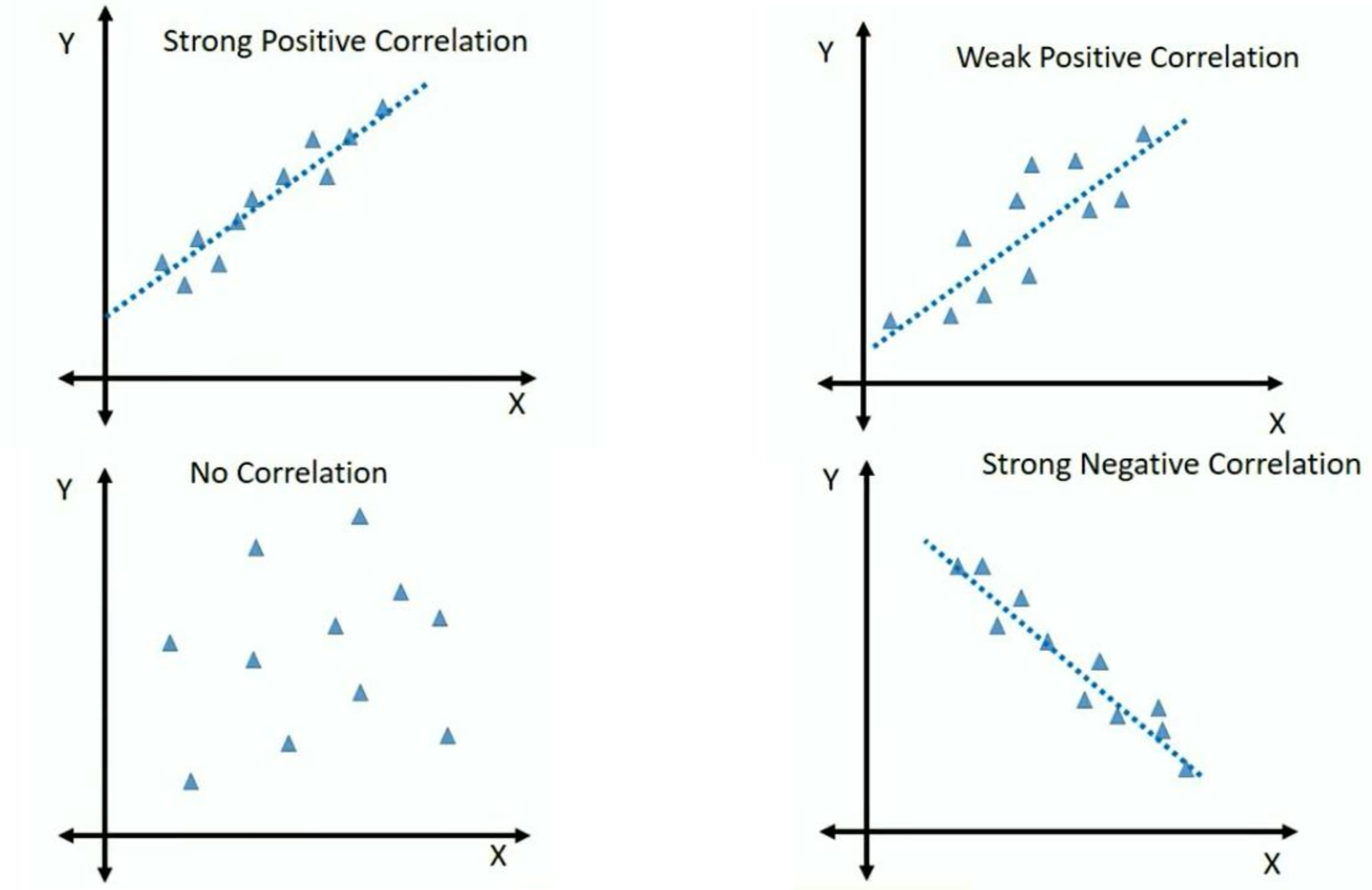
Regression in AML

Regression Analysis

- Statistical process for estimating the relationships among variables
- Relationship between a dependent variable and one or more independent variables (or 'predictors')
- The predictor is a continuous variable
- Can also be used to infer causal relationships between dependent and independent variables.



Casual Relationship?


Strong Positive Correlation
Y
X
Weak Positive Correlation
Y
X
No Correlation
Y
X
Strong Negative Correlation
Y
X

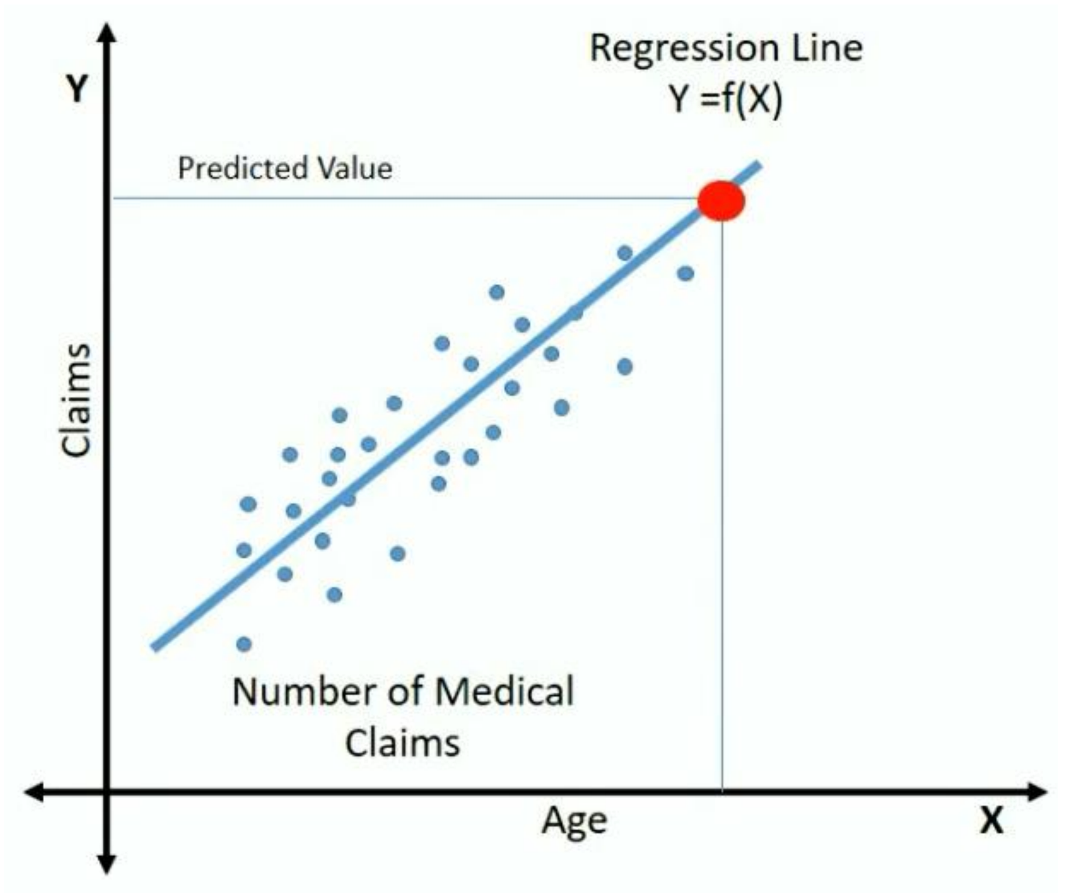
Linear Regression

Simple Regression

$$Y = B0 + B1X$$

Multivariate linear regression.

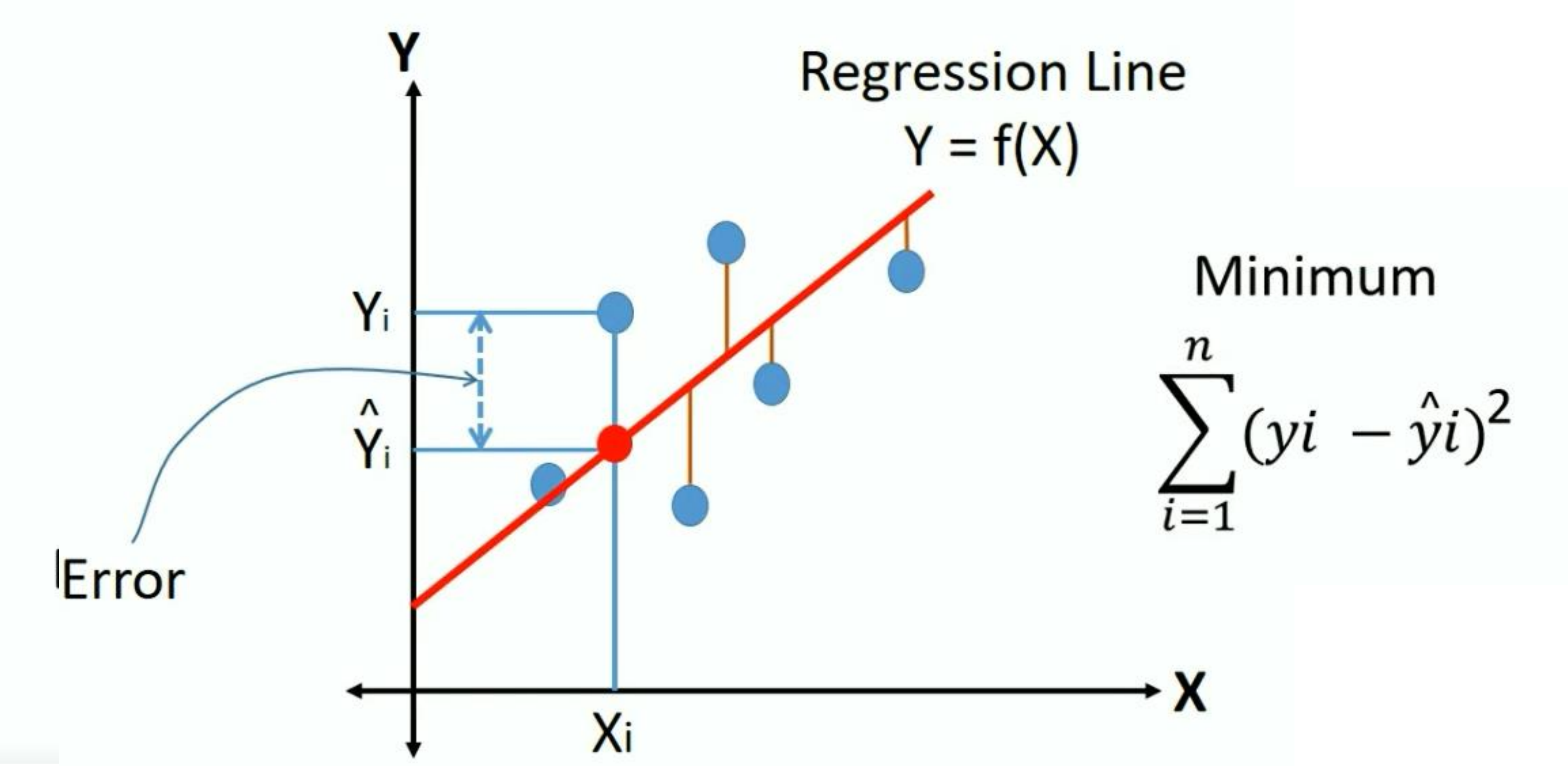
$$Y = B0 + B1X1 + B2X2 + B3X3 \ldots\ldots BnXn$$



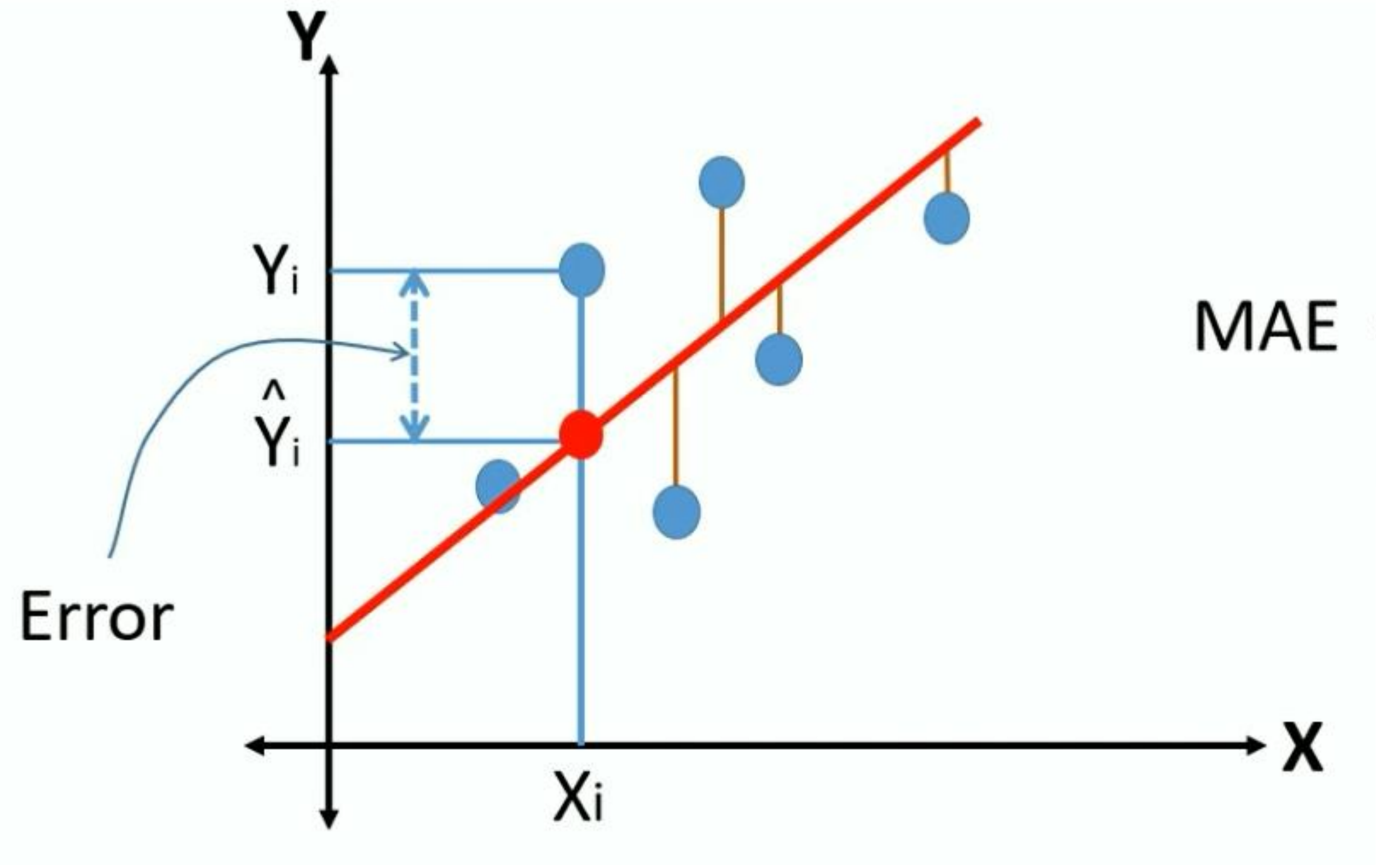
Simple Linear Regression

Common Regression Terms

Ordinary Least Square



Mean Absolute Error



$$MAE = \frac{1}{n}\sum_{i=1}^{n} |yi - \hat{y}i|$$

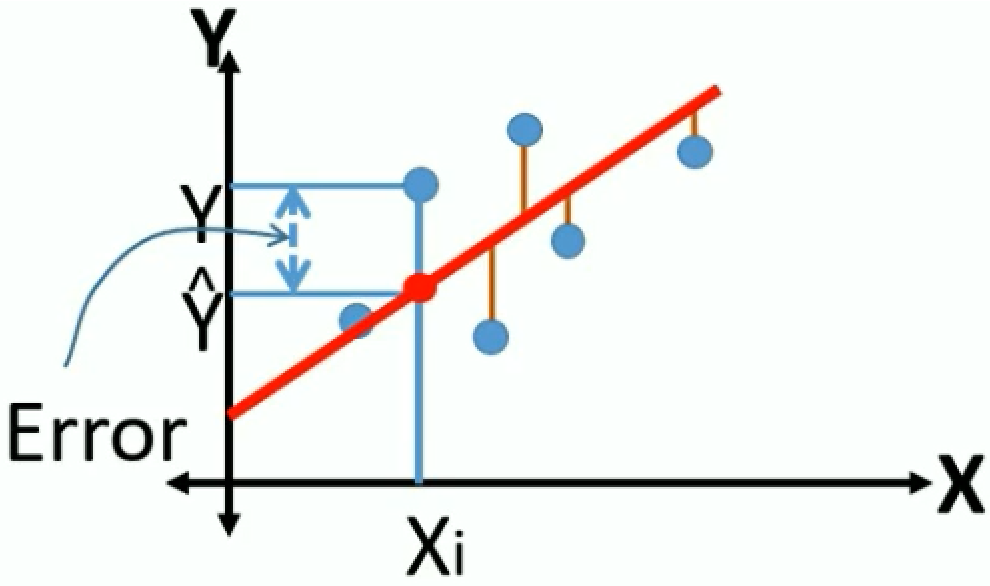
Mean absolute error (MAE) is a quantity used to measure how close forecasts or predictions are to the eventual outcomes.

Root Mean Square Error

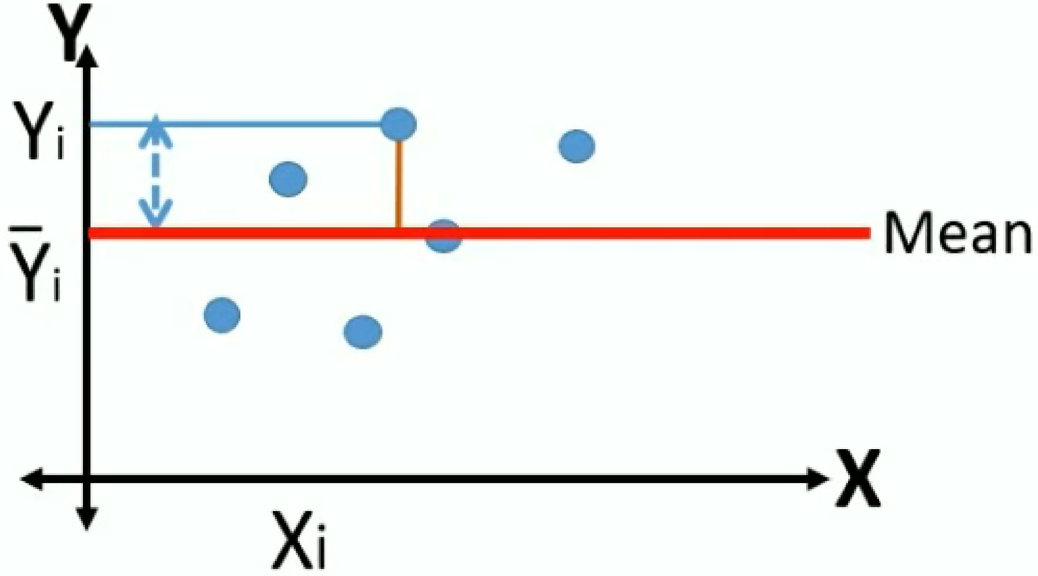
$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2}$$

- Very commonly used and makes for an excellent general purpose error metric for numerical predictions.
- Compared to the similar Mean Absolute Error, RMSE amplifies and severely punishes large errors.

Relative Absolute Error

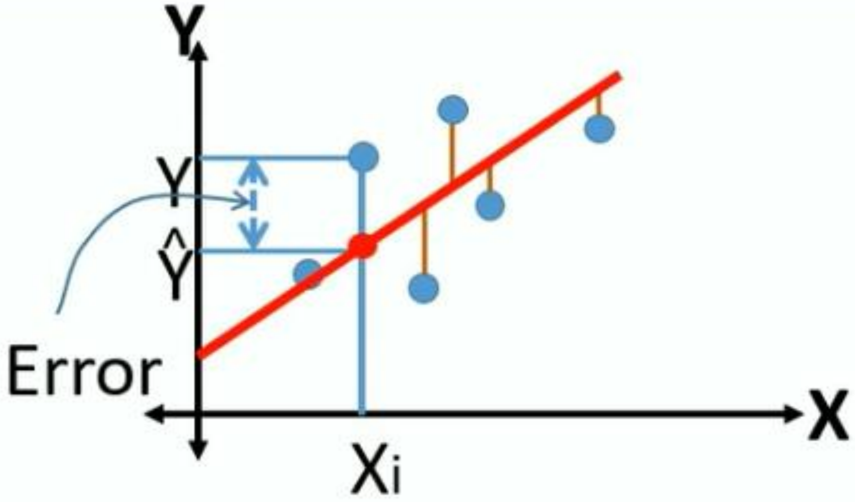


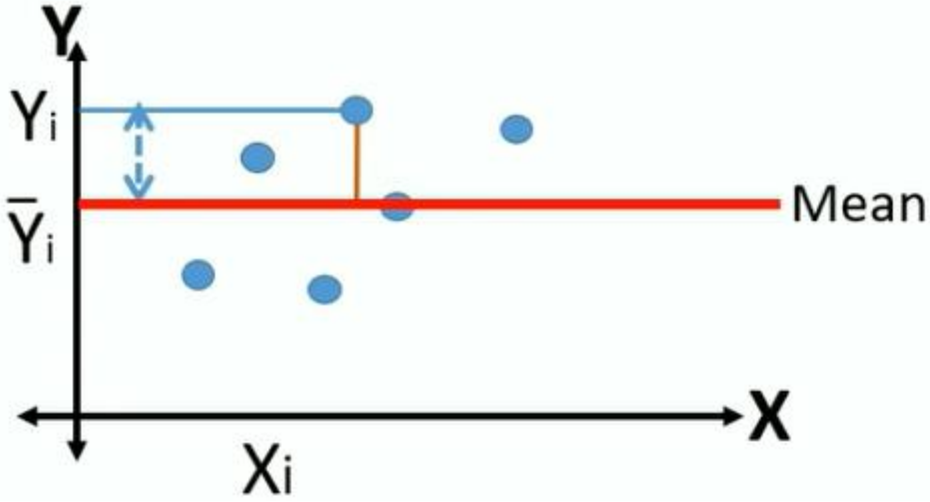
$$\sum_{i=1}^{n} |yi - \hat{y}i|$$



$$\sum_{i=1}^{n} |yi - \bar{y}i|$$

Relative Absolute Error





$$RAE = \frac{\sum_{i=1}^{n} |yi - \hat{y}i|}{\sum_{i=1}^{n} |yi - \bar{y}i|}$$

Business Problem

Build a model to predict the price of the vehicle based on the available historic data

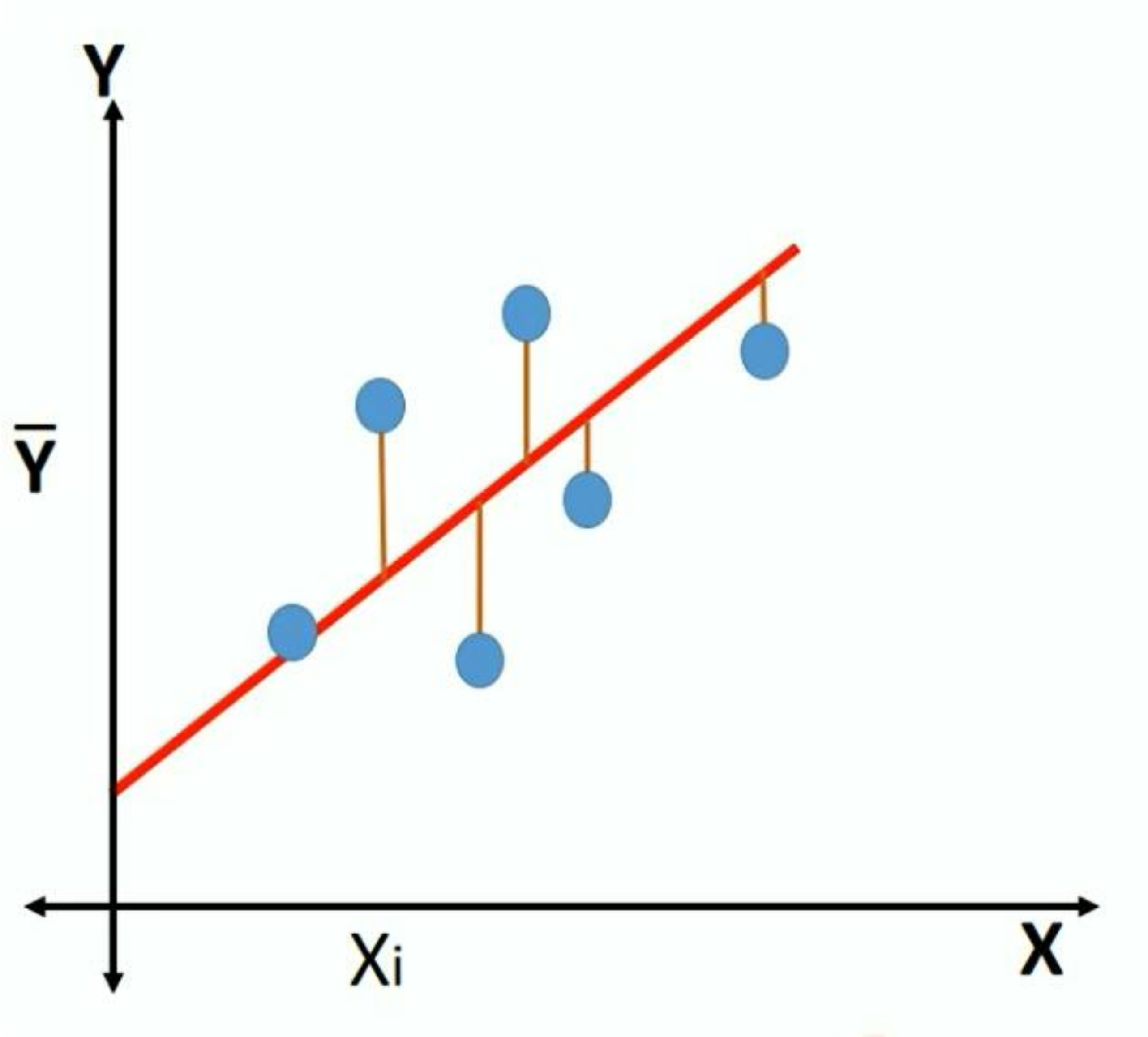
DEMOS

Linear Regression for Automobile Data

Coefficient of Determination or R Squared

Coefficient of Determination

How much (what %) of variation in Y is described by the variation in X?



R-Square with an Example

Hrs Studied (X)	Marks (Y)
0	40
2	52
3	53
4	55
4	56
5	72
6	71
6	88
7	56
7	74
8	89
9	67
9	89
5.38	66.31
Mean	

$Y = 41.8 + 4.55X$

Predicted Marks $\hat{Y}$
41.80
50.90
55.45
60.00
60.00
64.55
69.10
69.10
73.65
73.65
78.20
82.75
82.75

$(Y - \bar{Y})$^2	$(\hat{Y} - \bar{Y})$^2
692.22	600.74
204.78	237.47
177.16	117.94
127.92	39.82
106.30	39.82
32.38	3.10
22.00	7.78
470.46	7.78
106.30	53.88
59.14	53.88
514.84	141.37
0.48	270.27
514.84	270.27
3028.77	1844.12
SST	SSR

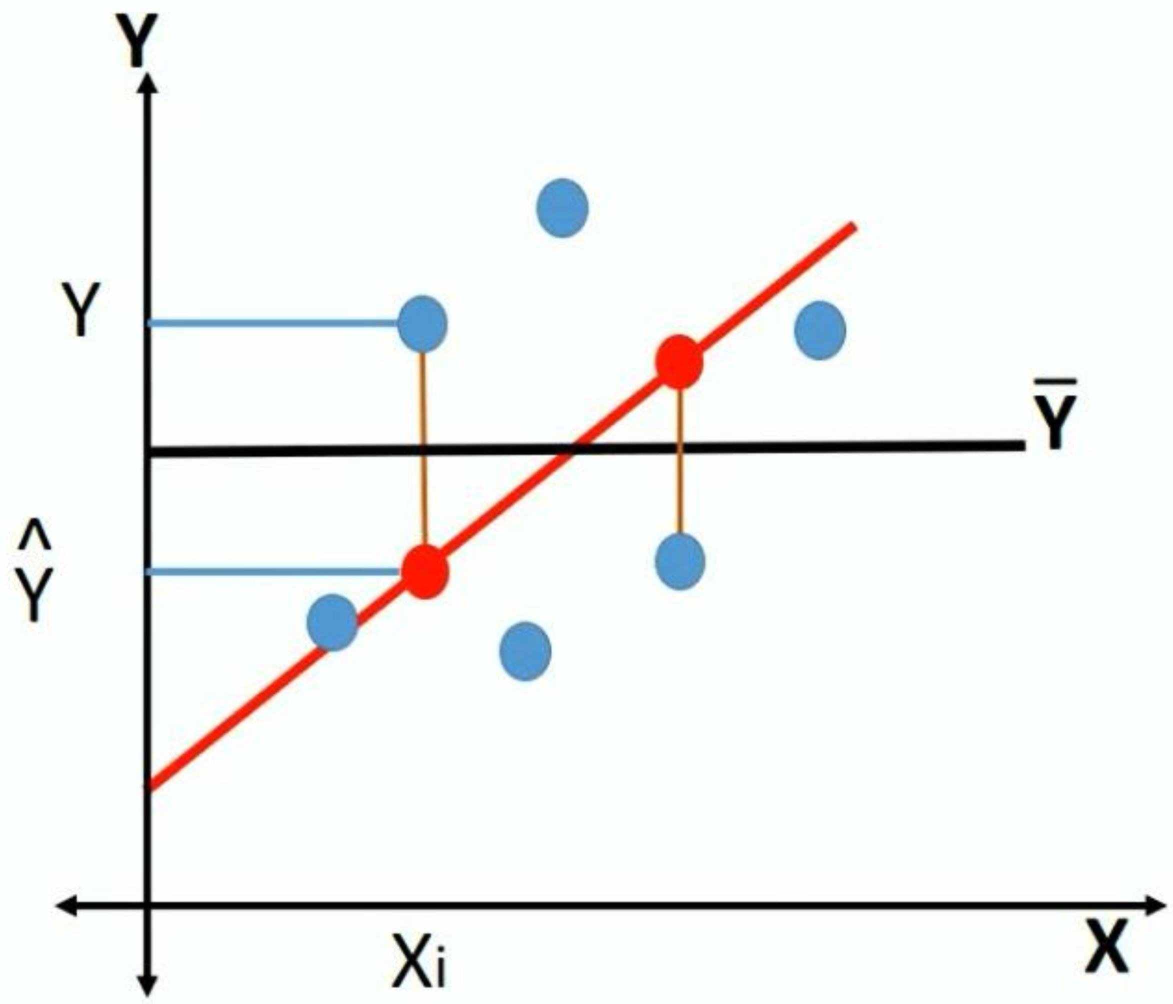
Coefficient of Determination

Sum of Squares Due to Regression

$$SSR = \sum_{i=1}^{n} \left(\boxed{\hat{y}i} - \bar{yi}\right)^2$$

Total Sum of Squares

$$SST = \sum_{i=1}^{n} \left(\boxed{yi} - \bar{yi}\right)^2$$



Coefficient of Determination

$$R^2 = SSR/SST = 1844.12/3028.77$$
$$= 0.60886$$

Higher the value → Variation in Y is explained by variation in X.

Gradient Descent

Hypothesis

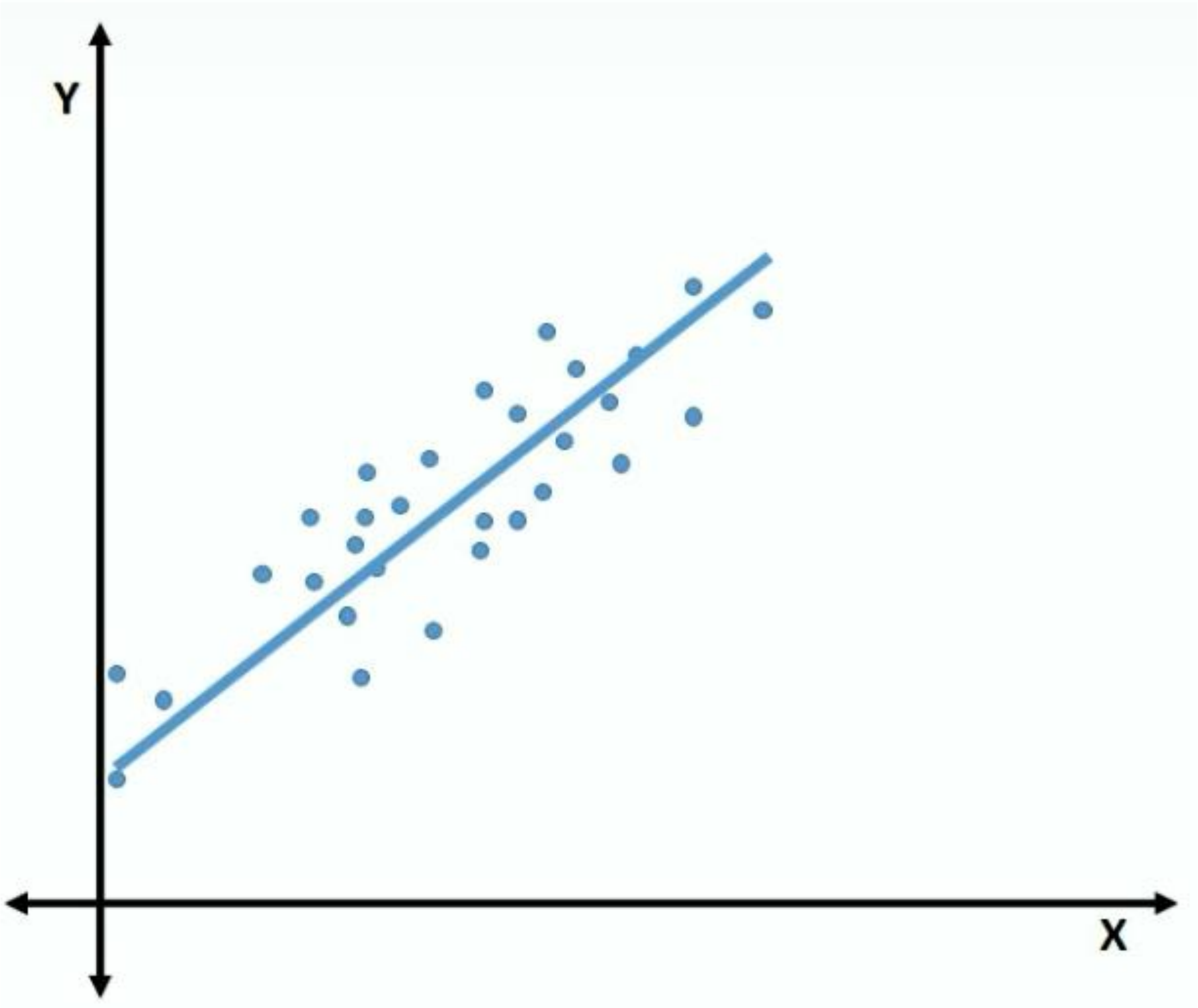
"Proposed explanation made on the basis of limited evidence as a starting point for further investigation"

$$h(x) = b0 + b1x$$

Find out value of b0 and b1 such that

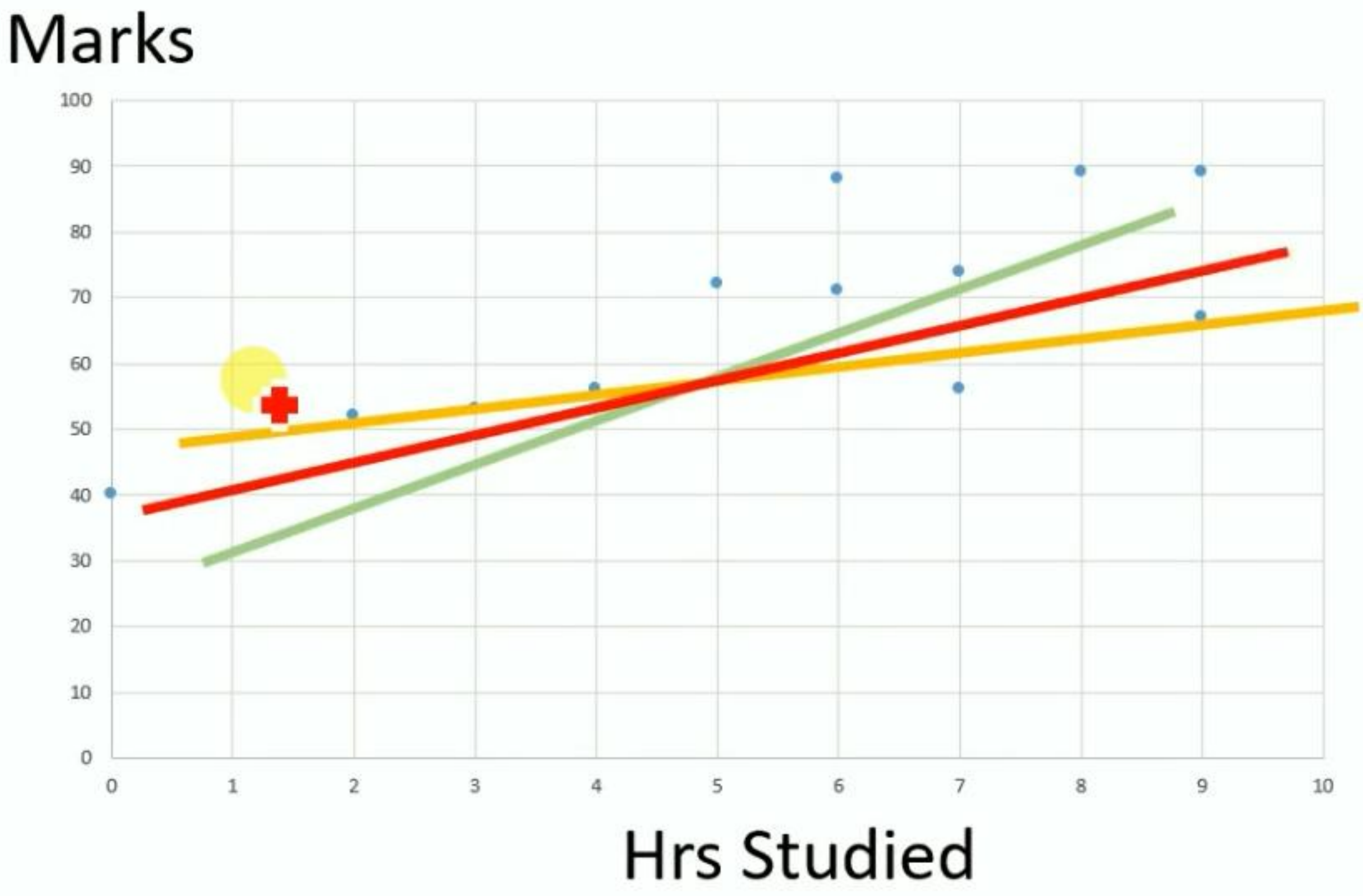
$$Y \sim N\ h(x)$$

for the given observations



Example of Linear Regression

Hrs Studied	Marks
0	40
2	52
3	53
4	55
4	56
5	72
6	71
6	88
7	56
7	74
8	89
9	67
9	89



Cost Function

Hypothesis: $h(x) = b0 + b1x$

Hrs Studied	Marks	b0 = 0; b1 = 1 Marks Predicted
0	40	0
2	52	2
3	53	3
4	55	4
4	56	4
5	72	5
6	71	6
6	88	6
7	56	7
7	74	7
8	89	8
9	67	9
9	89	9

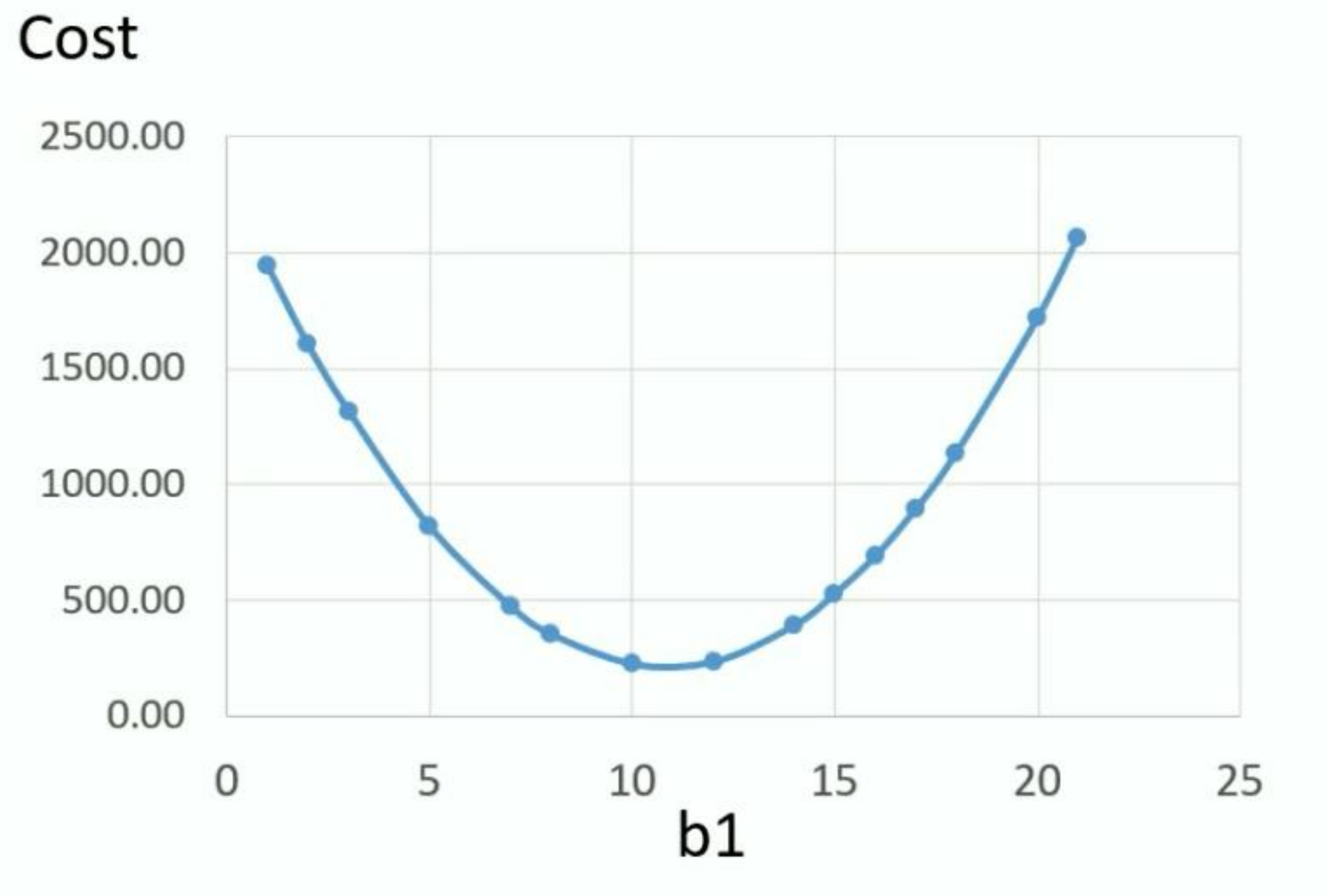
$(Yi - \hat{Yi})^2$
1600
2500
2500
2601
2704
4489
4225
6724
2401
4489
6561
3364
6400

$$\frac{1}{2n}\sum_{i=1}^{n}(yi - \hat{y}i)^2$$

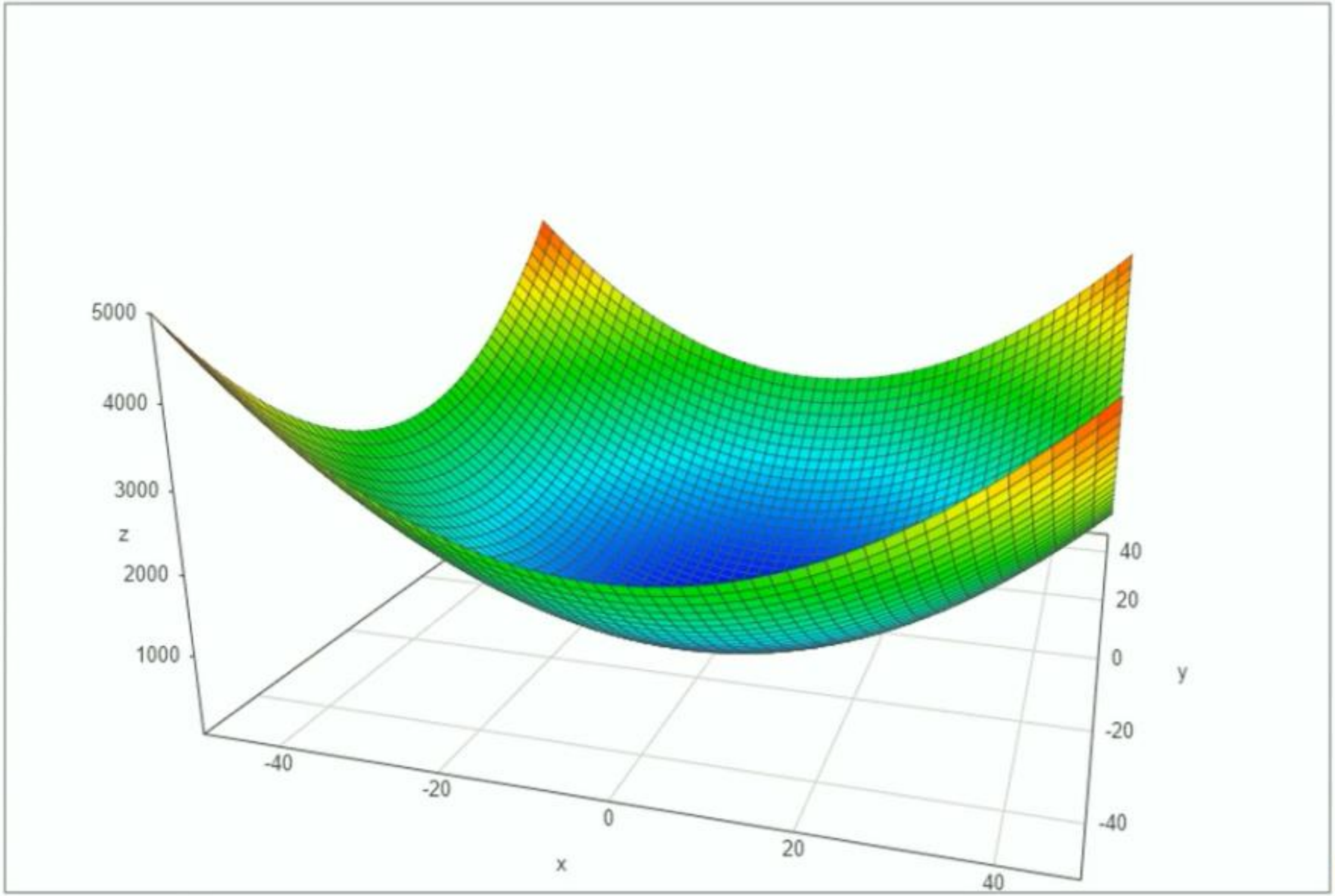
b0	b1	Cost
0	1	1944.538

Cost Function Plot

b0	b1	cost
0	1	1944.54
0	2	1610.08
0	3	1311.46
0	5	821.77
0	7	475.46
0	8	356.08
0	10	224.85
0	12	237.00
0	14	392.54
0	15	524.08
0	16	691.46
0	17	894.69
0	18	1133.77
0	20	1719.46
0	21	2066.08



Cost Function with b0 and b1

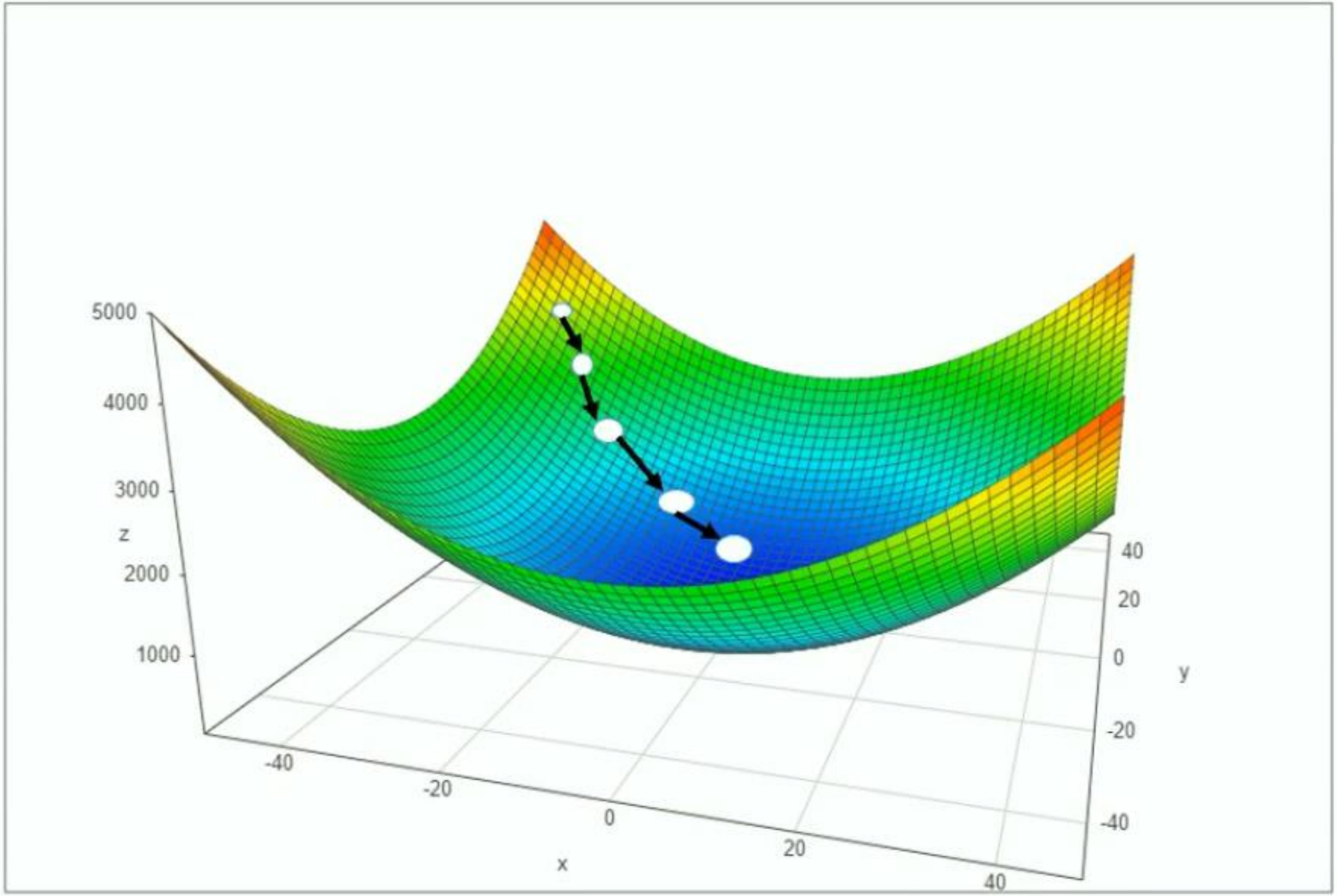


X axis – b1

Y axis – b0

Z = C(b0,b1)

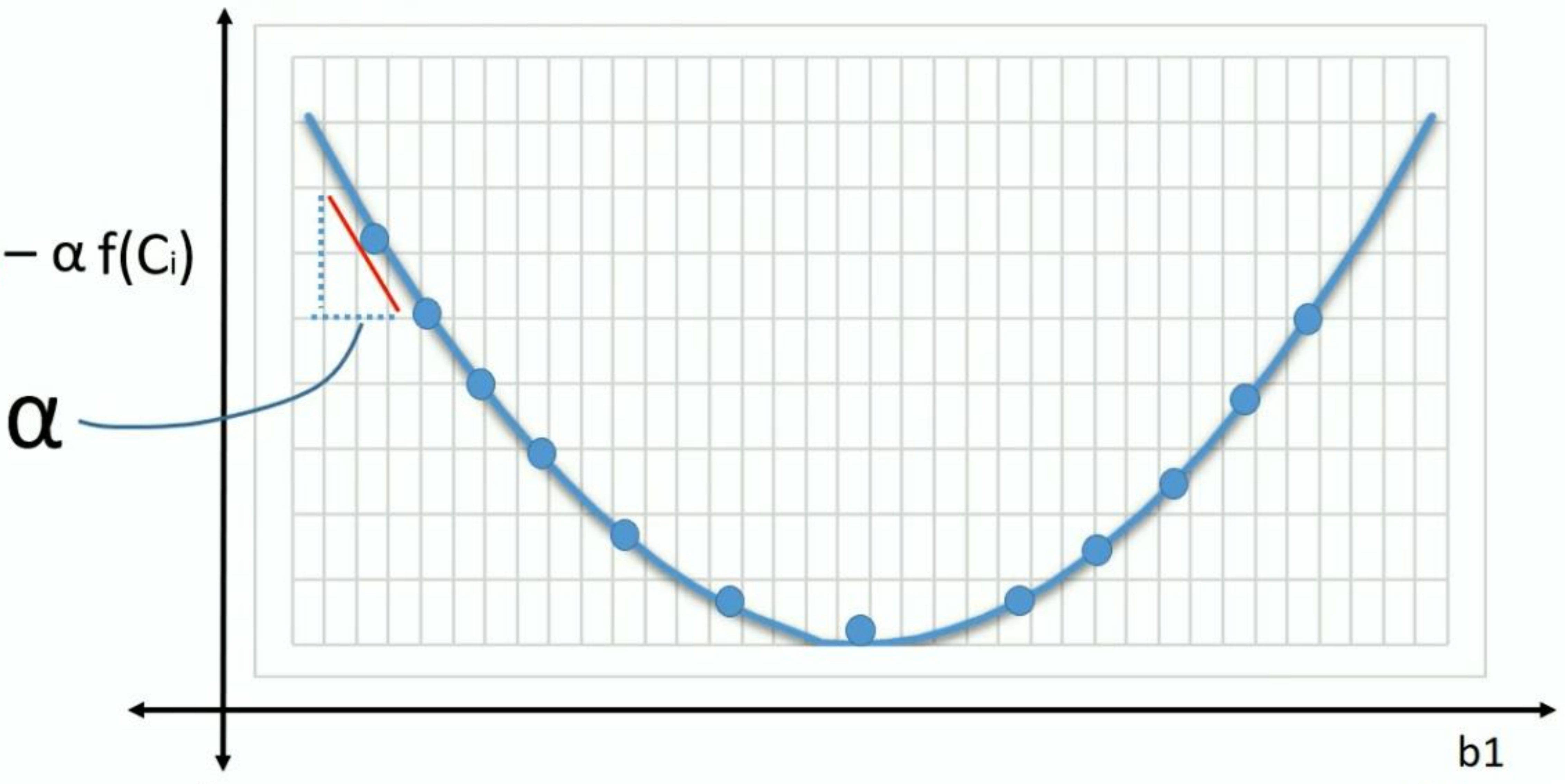
Gradient Descent



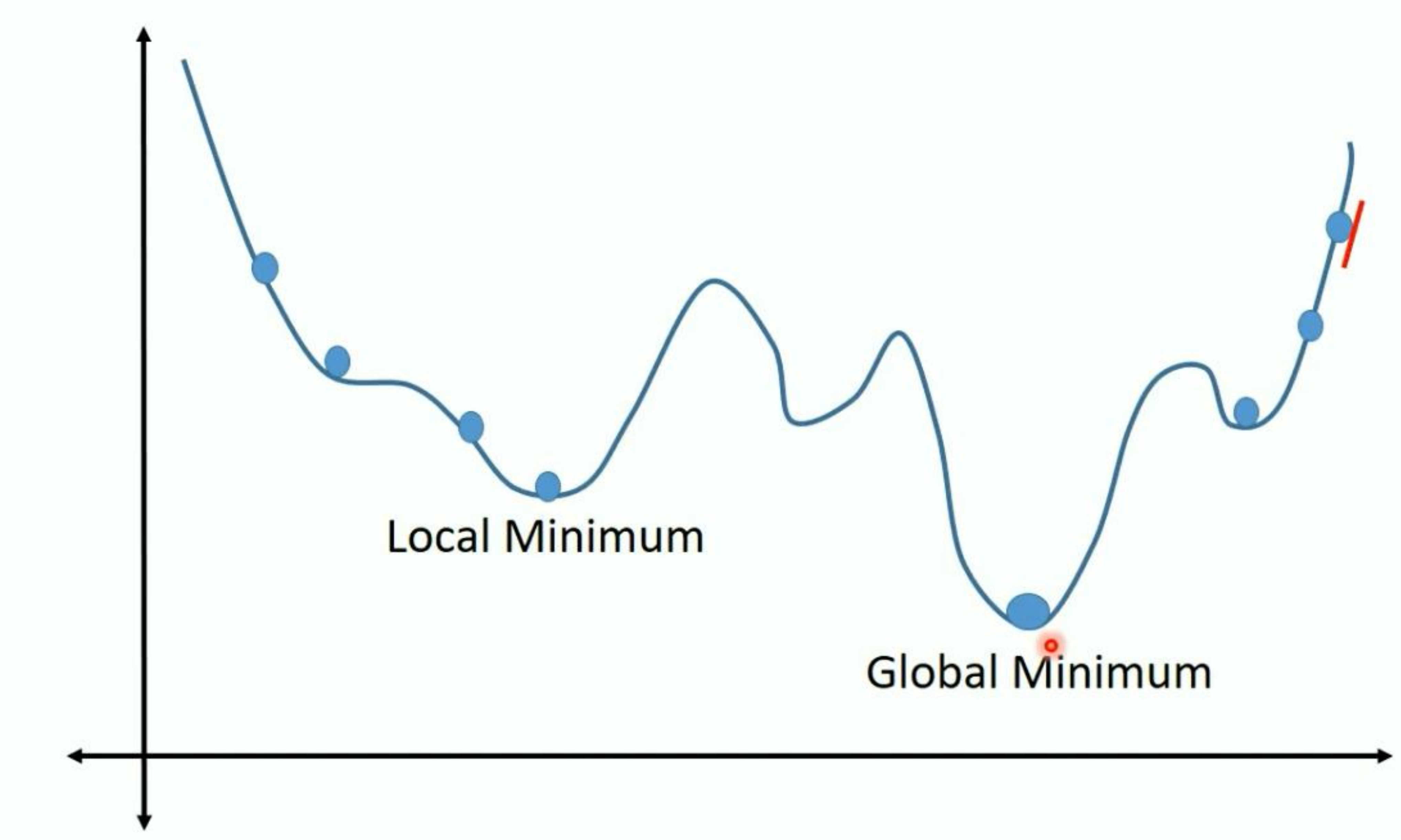
Gradient Descent

Cost Function: C(b1)

$$b_j := b_i - \alpha f(C_i)$$



Gradient Descent



Batch Gradient Descent

X1	X2	....	Xn

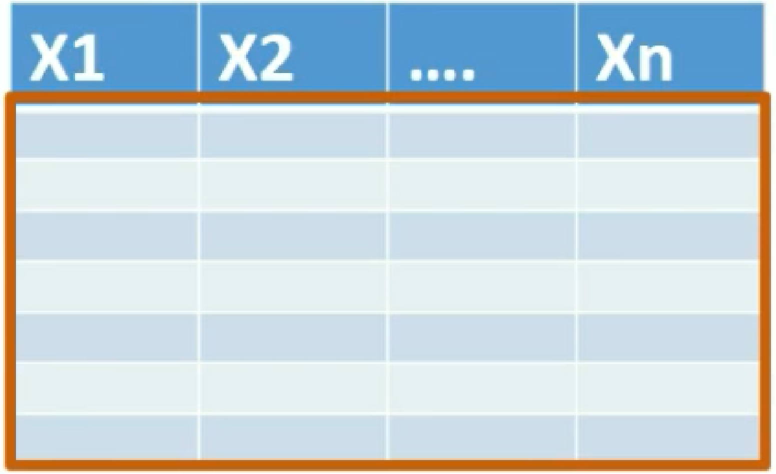
$$b_j := b_j - \alpha\ f(C_i)$$

Does it for
number of examples
number of features
learning rate

Sum of All before taking one step (epoch)
Long time to reach the bottom

Batch Gradient Descent

Batch Vs Stochastic Gradient Descent

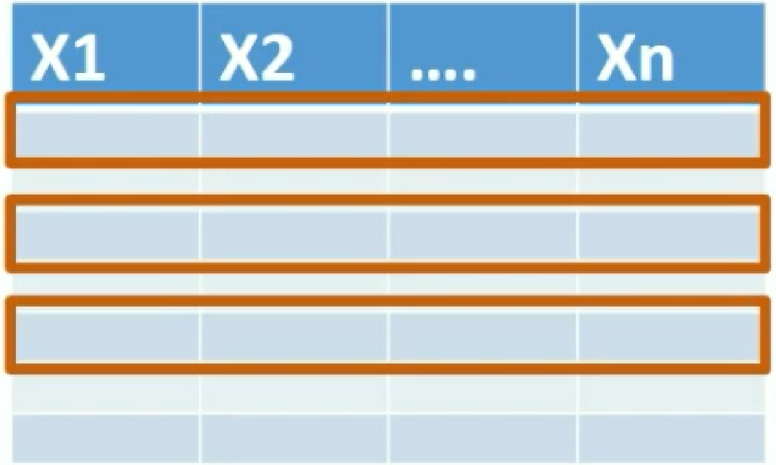


$b_j := b_j - \alpha\, f(C_i)$

Does it for

- number of examples
- number of features
- learning rate

Batch Gradient Descent

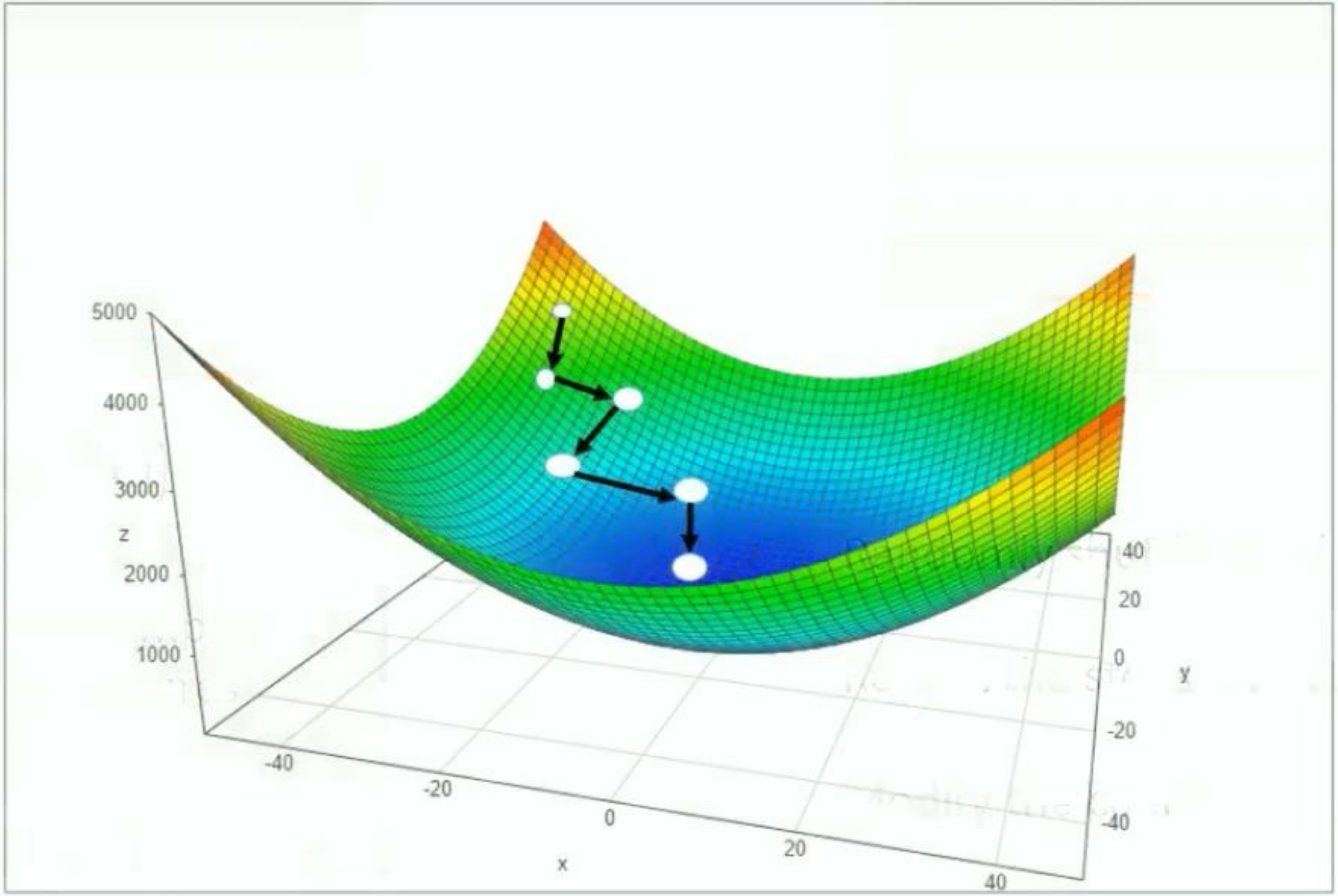


Randomly shuffle the dataset

Repeat the steps for every example

Modify the coefficient at every step

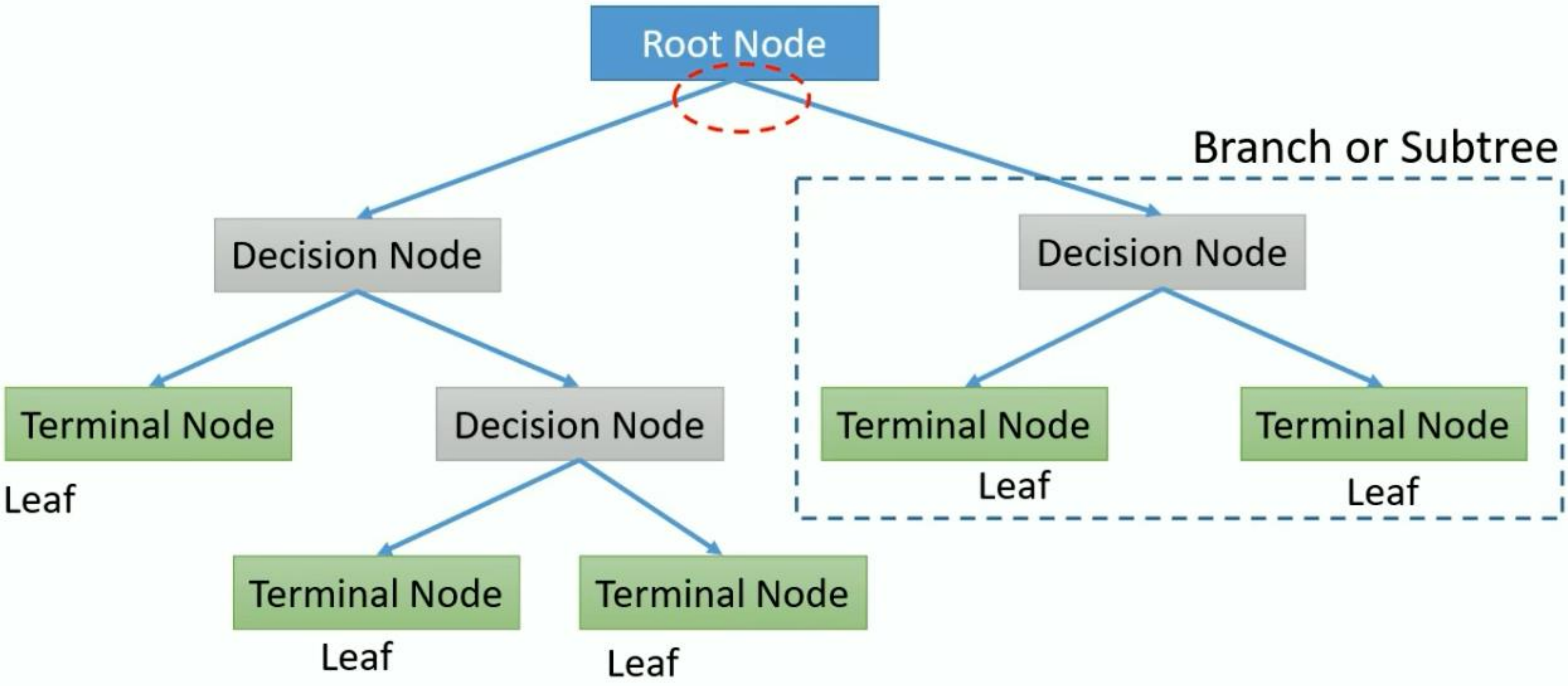
Stochastic Gradient Descent

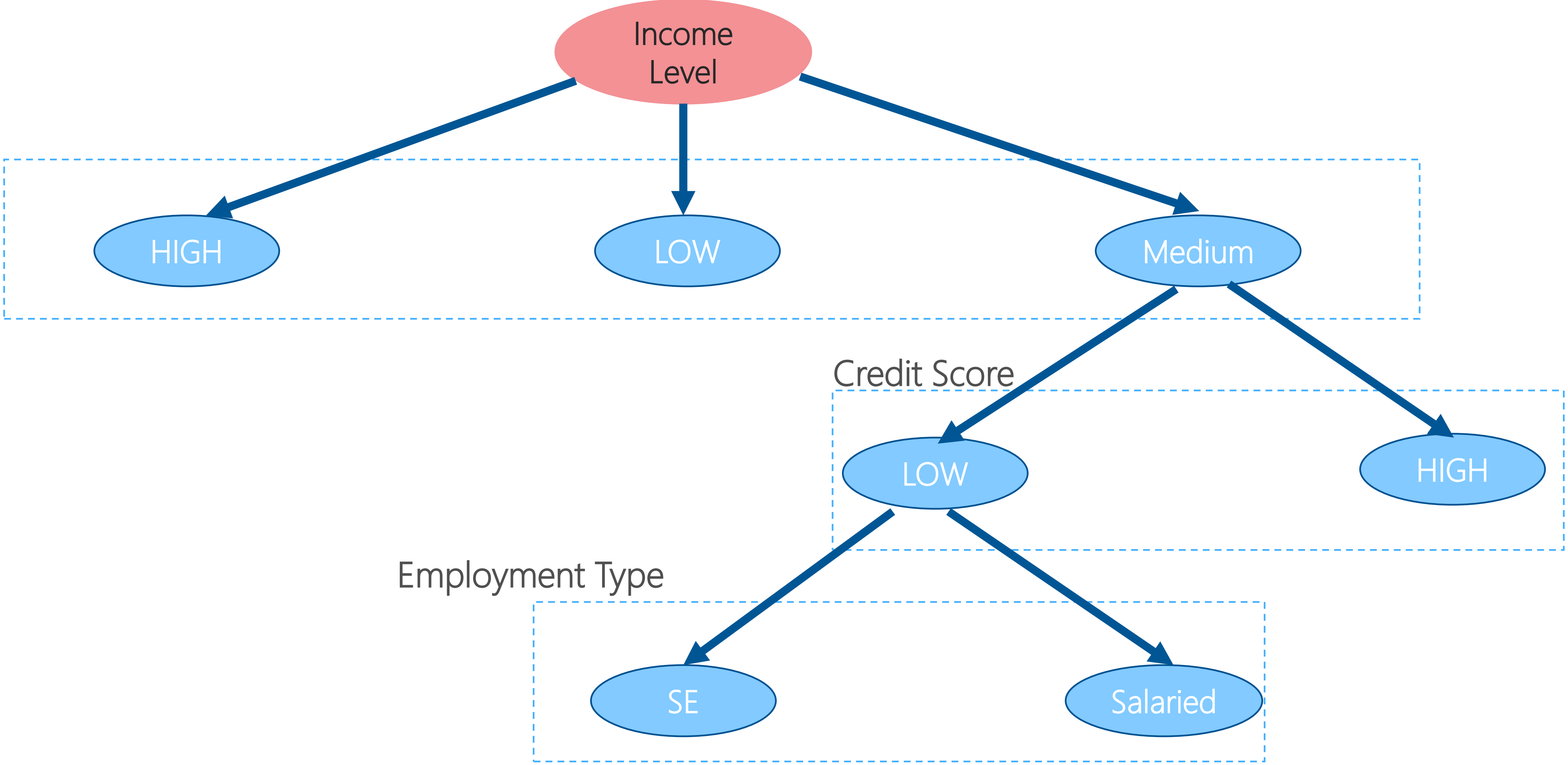


DEMO
Online Gradient Descent in Linear Regression

Regression Using Decision Tree?

Decision Tree Terms





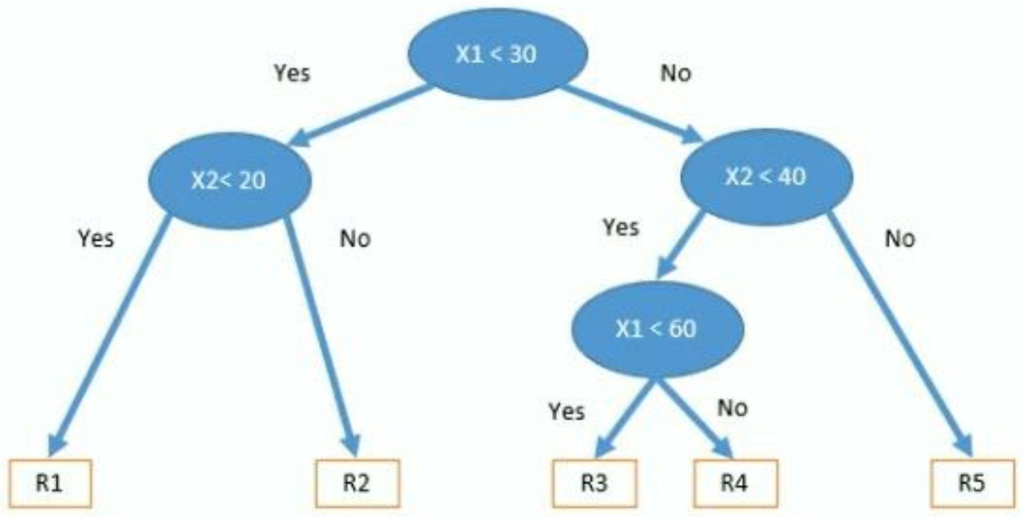
LID	IL	CS	ET	Status
L1	Medium	Low	SE	No
L8	Medium	Low	SE	No

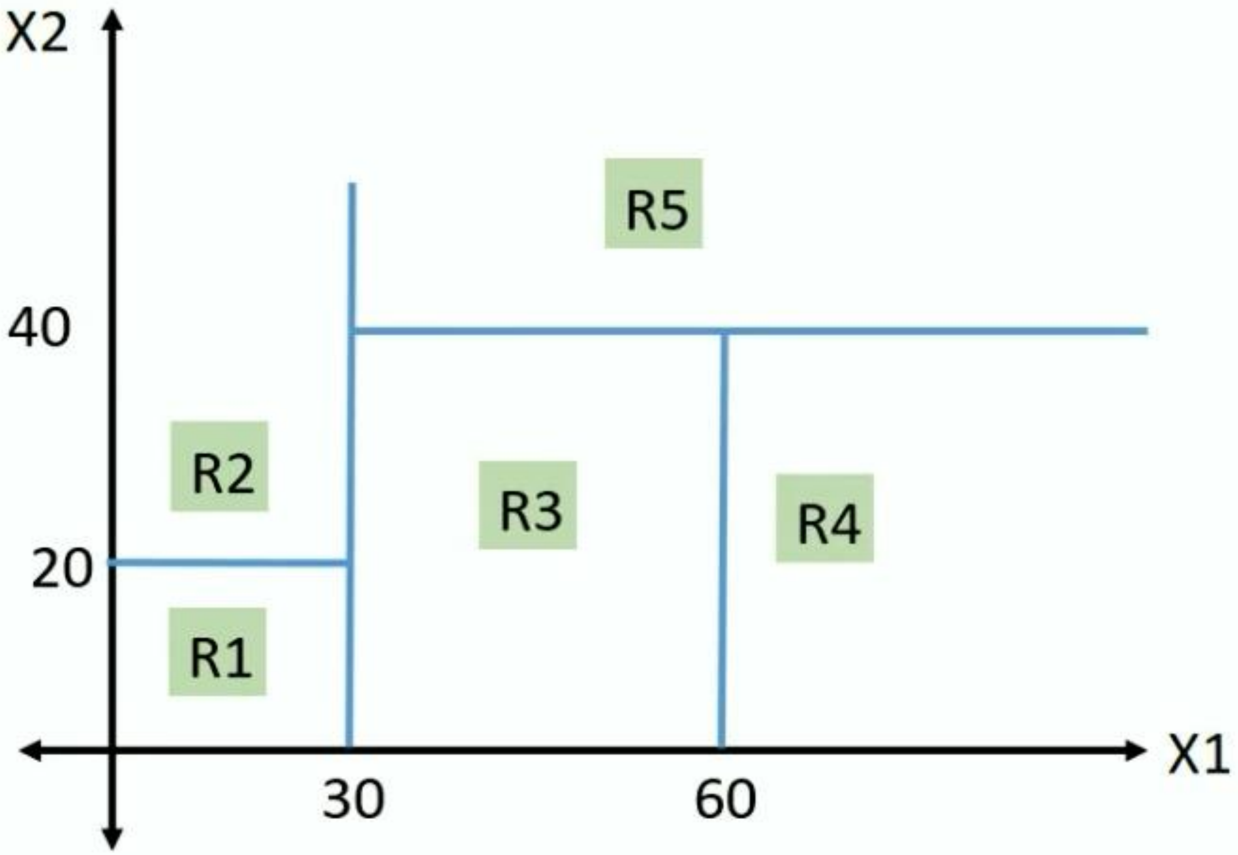
Pure Subset

LID	IL	CS	ET	Status
L4	Medium	Low	Salaried	Yes

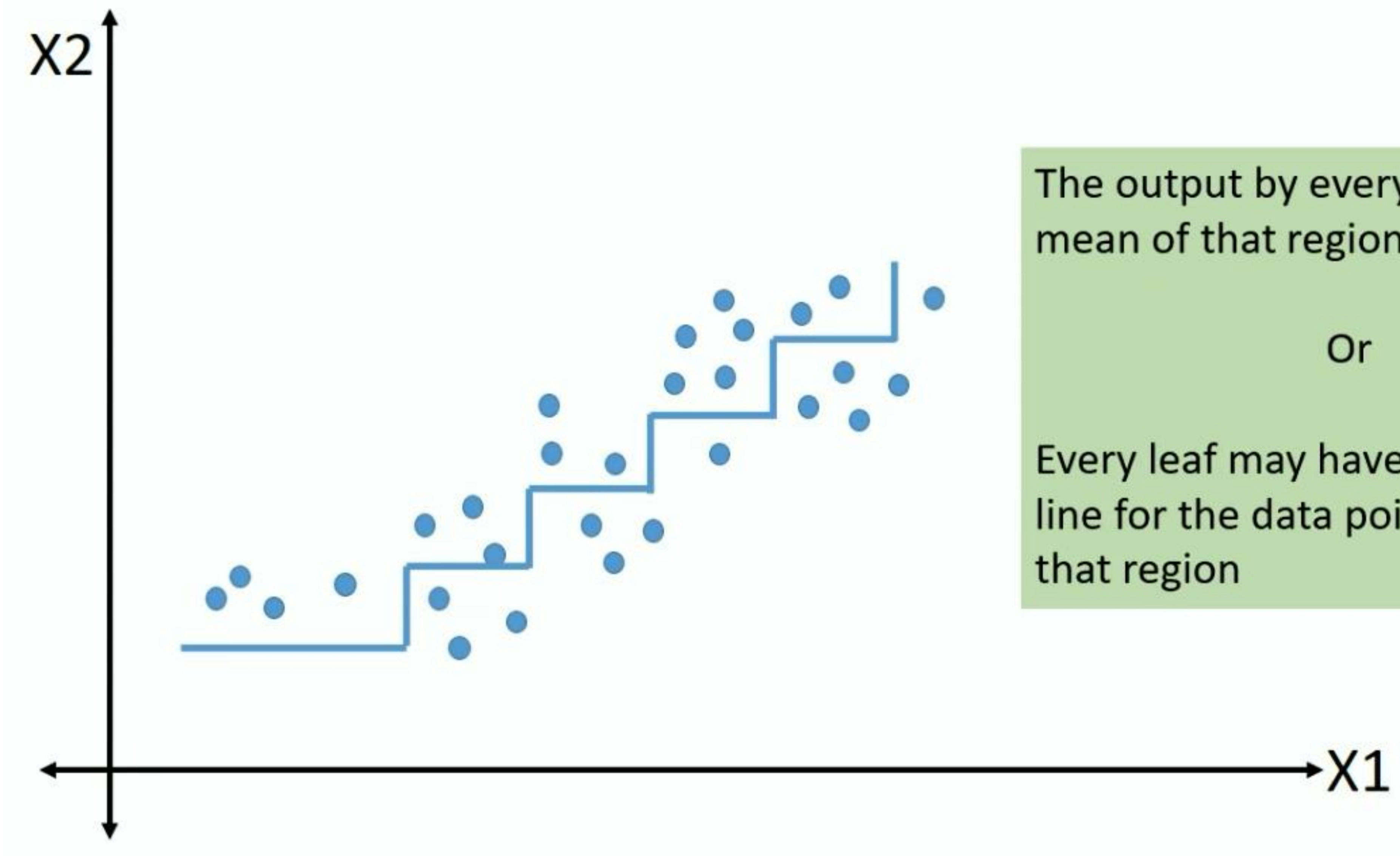
Pure Subset

Decision Tree Regression





Decision Tree Regression



The output by every region is the mean of that region

Or

Every leaf may have a regression line for the data points within that region

Boosted Decision Tree Regression

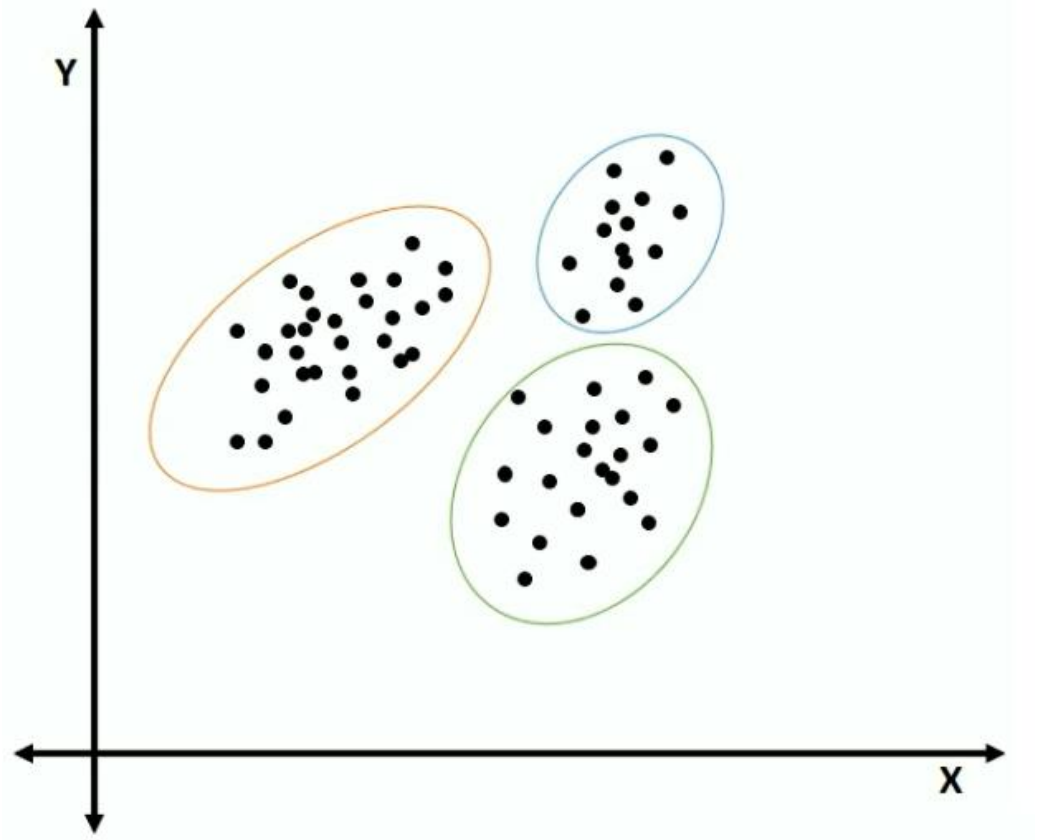
- MART gradient boosting algorithm.
- Builds each regression tree in a step-wise fashion
- Predefined loss function to measure the error in each step

DEMO
Boosted Decision Tree Regression

Cluster Analysis

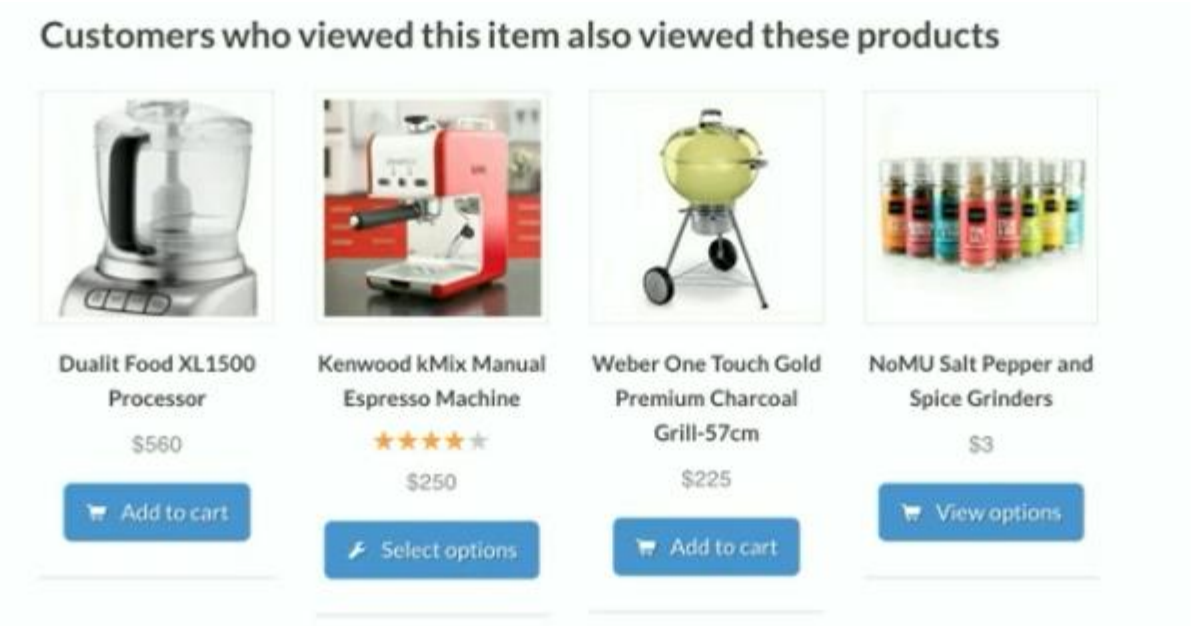
Clustering or Cluster Analysis

- Clustering is the task of grouping a set of objects
- Unsupervised Learning model
- Discovering distinct groups in customer databases
- Used for creating strategies to adopt for certain segments



Examples of Clustering

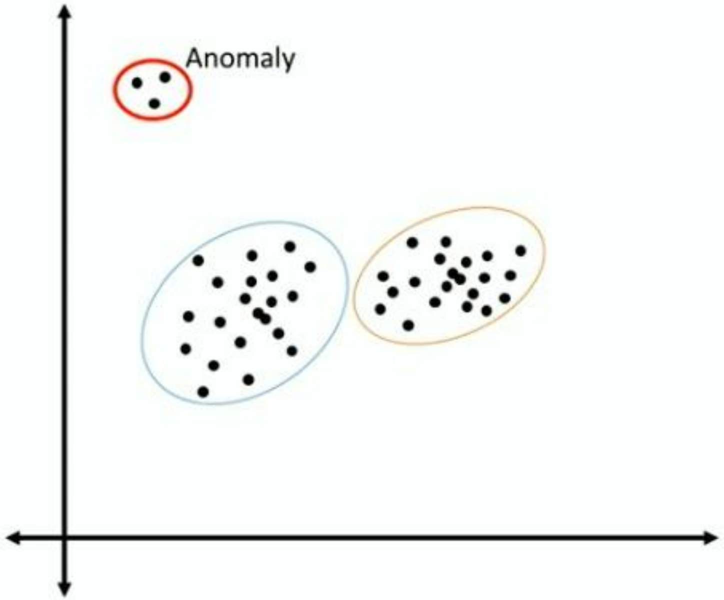
- Recommendation engines
- Market Segmentation
- Social Network Analysis



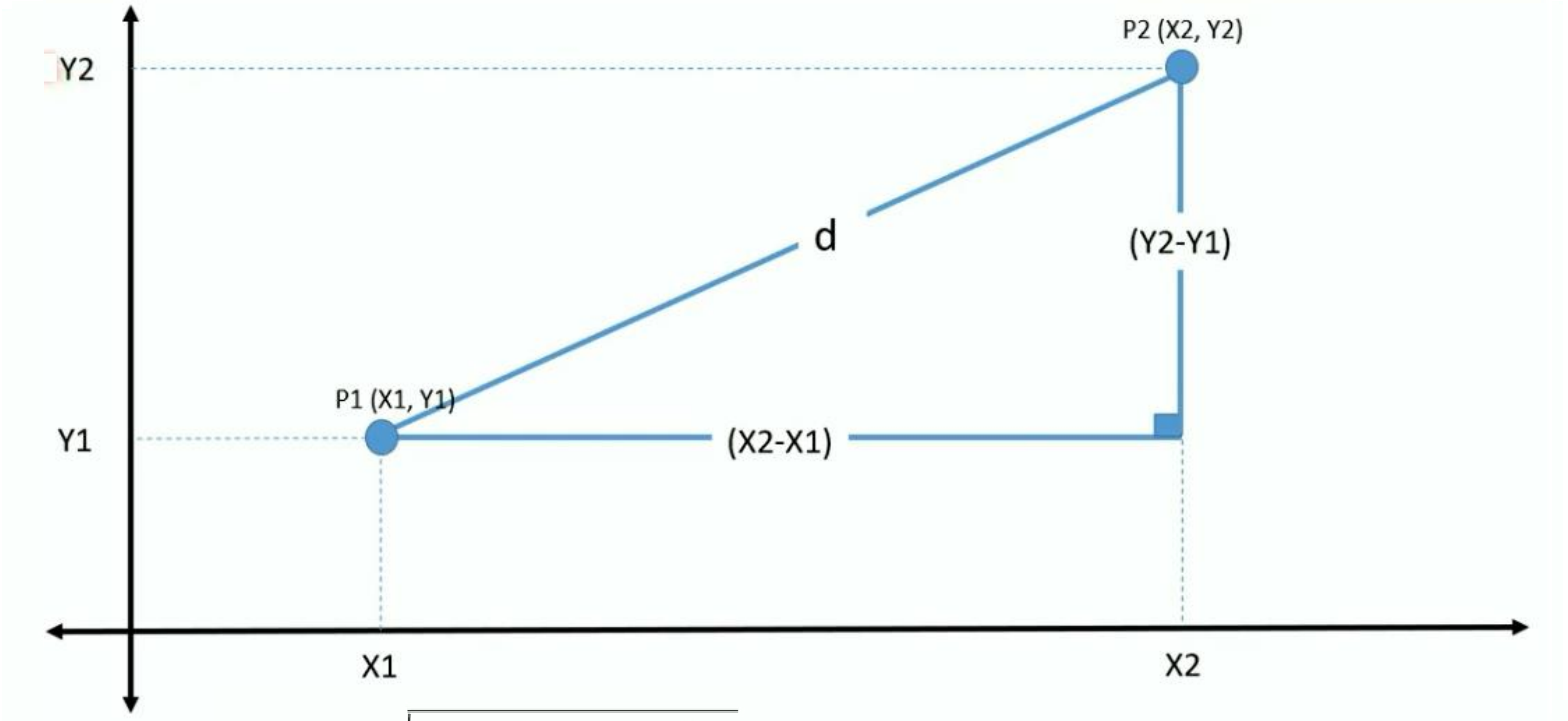
Examples of Clustering

- Medical /Health Science
- Image Segmentation

- Anomaly detection



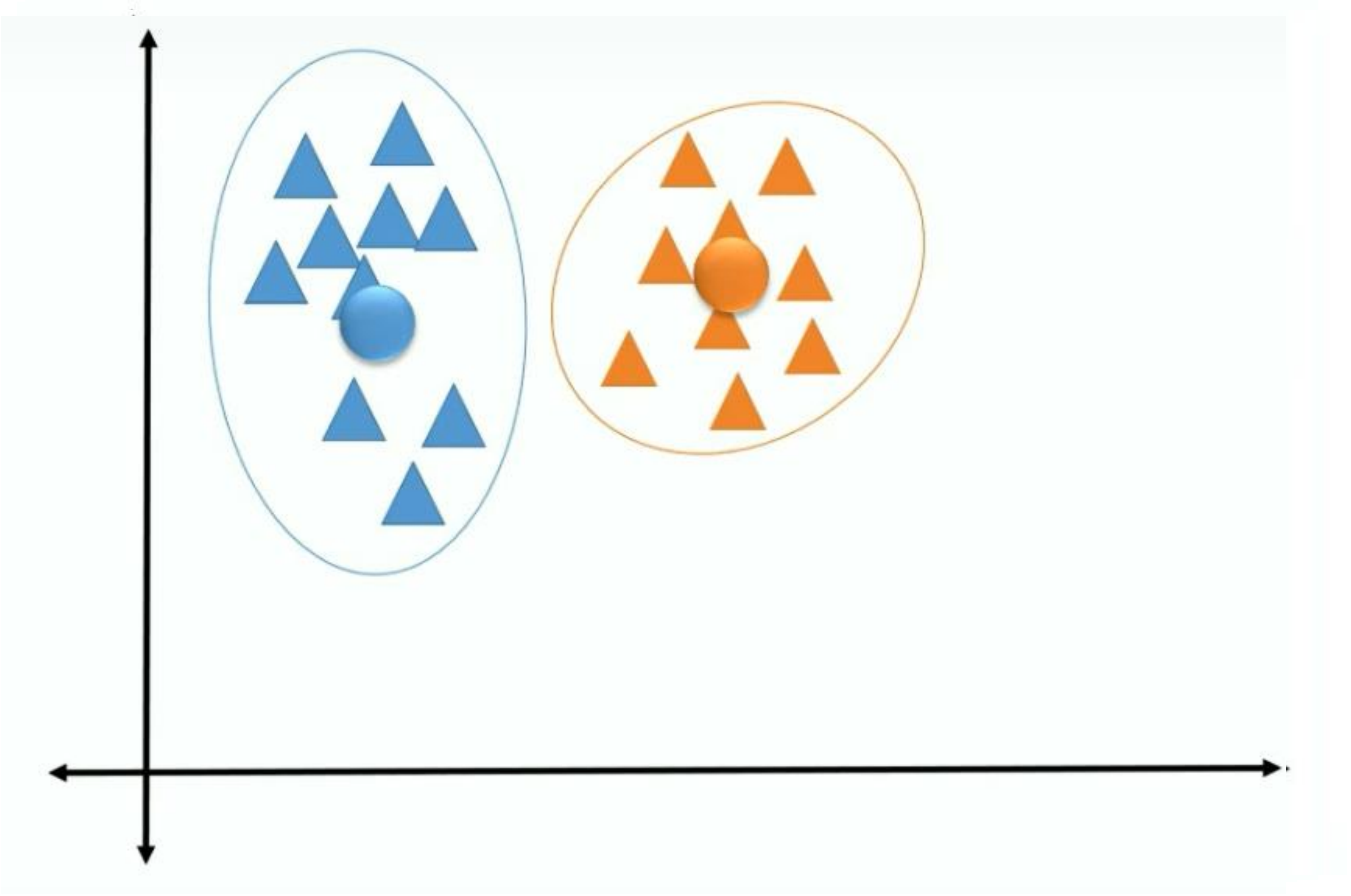
Euclidean Distance



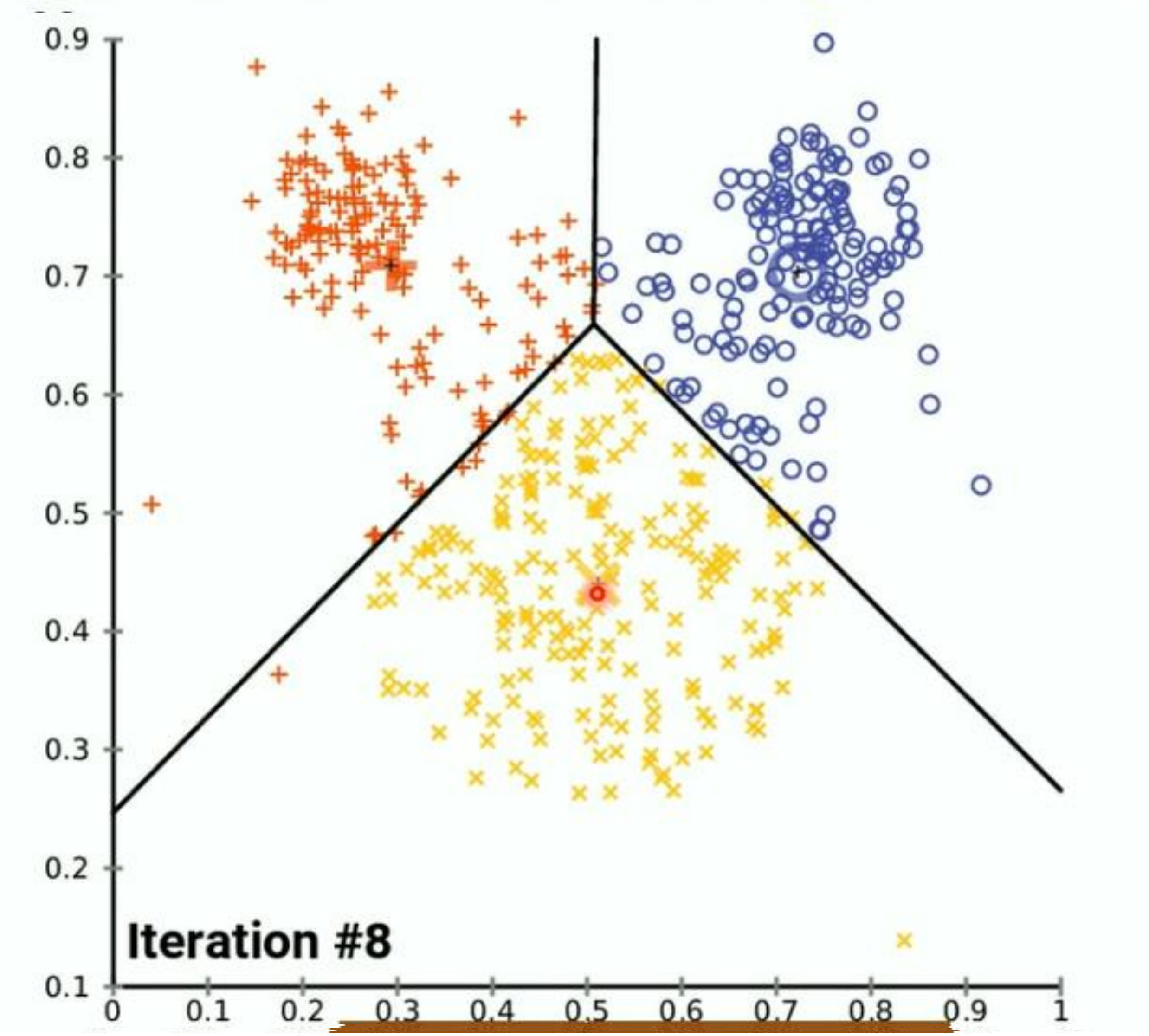
$$d = \sqrt{(X2-X1)^2 + (Y2-Y1)^2}$$

K-Means algorithm uses Euclidean Distance for clustering

How Clusters are formed?



https://en.wikipedia.org/wiki/K-means_clustering User: Chire



Good Cluster Analysis

- Observations in the same group share similar characteristics

- Clusters have proportionate number observations

Cluster Initialization

- Random - Random placement of data points into clusters
- First N or Forgy Method - First Data points at Random
- K-Means++ - Default method and an improvement over finding the initial means
- K-Means++ Fast — Optimised for faster clustering
- Evenly
- Use Label Column

DEMO
Clustering using K-Means

Microsoft

Recommendation

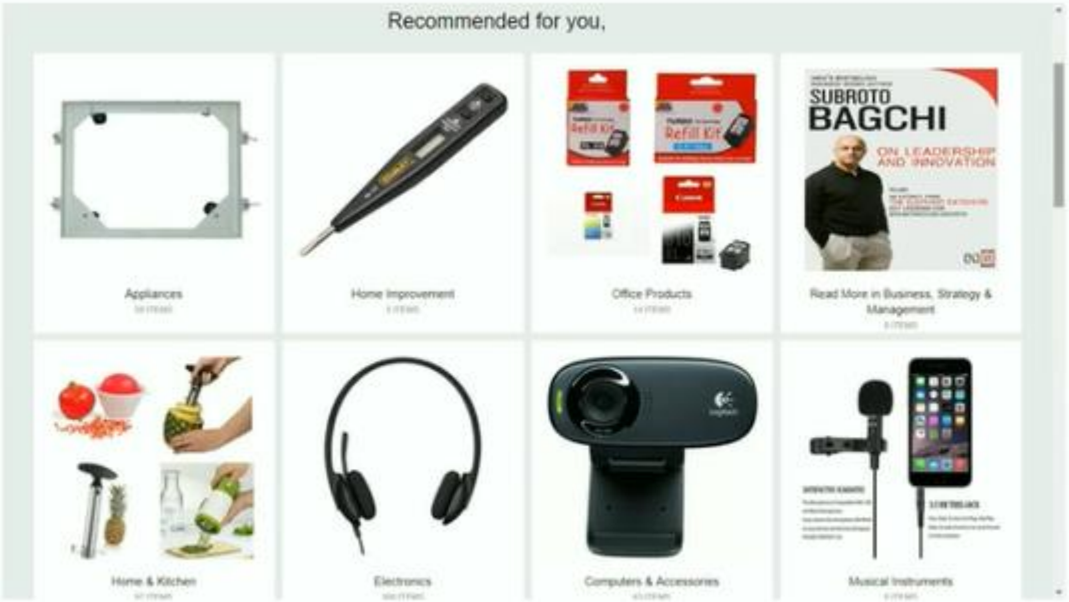
AzureML Matchbox Recommender

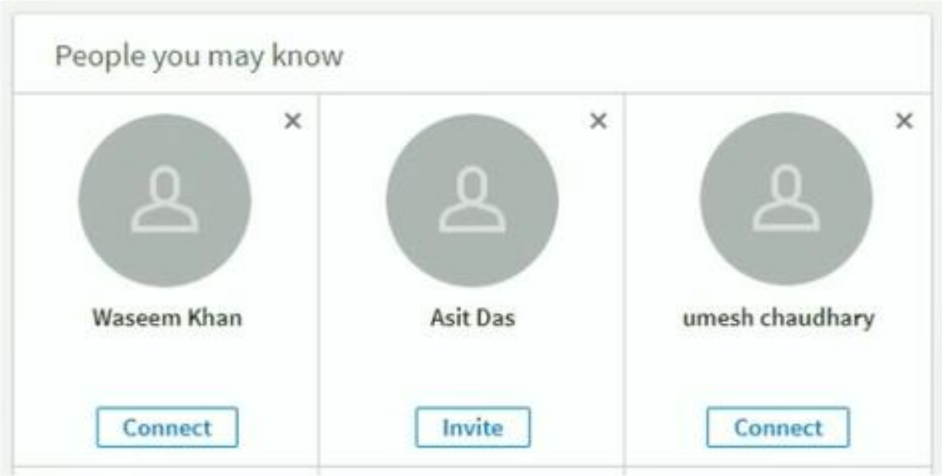
Recommendation System

- What is a Recommendation System?
- Types of Recommendation Systems
    - Collaborative Filtering
    - Content Based Filtering
- How a recommendation system works?

What is Recommendation System?

"A recommender system or a recommendation system (platform or engine) seeks to predict the "rating" or "preference" that a user would give to an item."





Everyday Recommendations

History of preferences or behaviour

Shopping

Restaurant

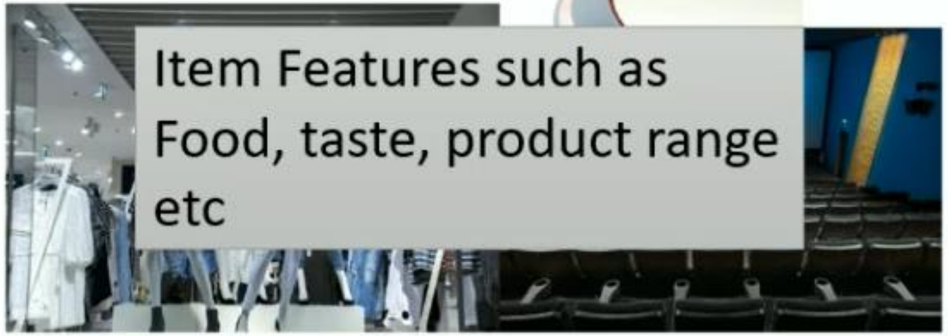
Your Features such as Likes, Dislikes, Taste, Style

e

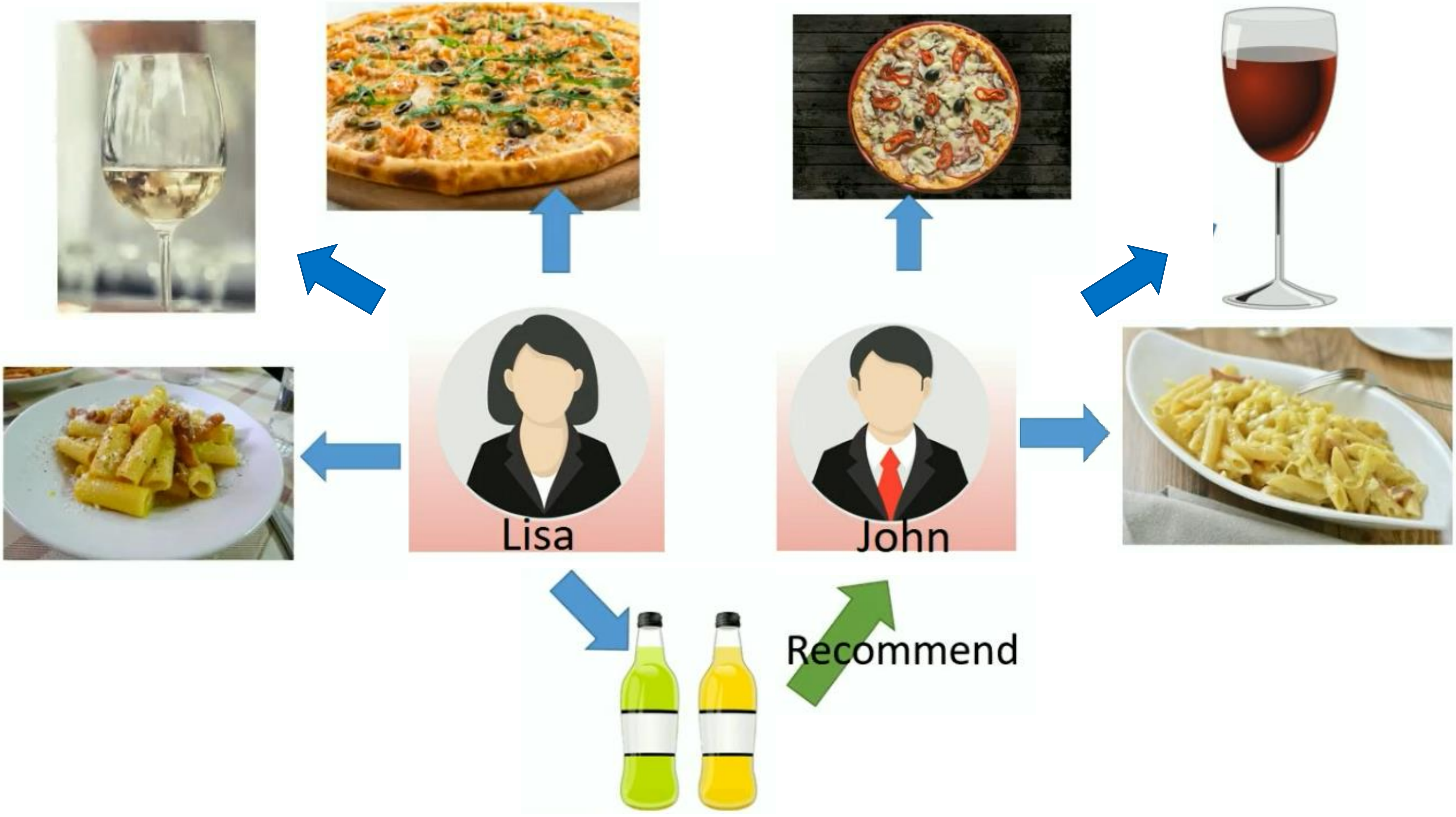
Books

Travel

Friends, Family, Colleagues, Professors



Item Features such as Food, taste, product range etc

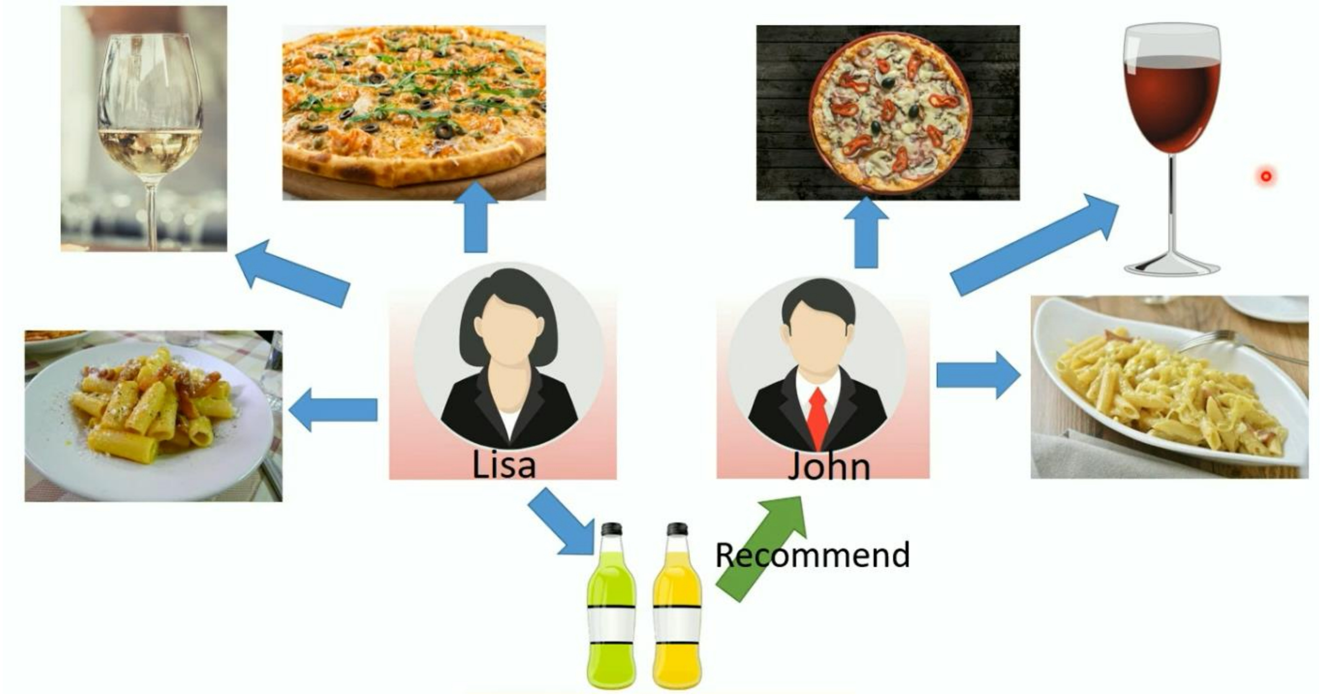

Lisa
John
Recommend

Types of Recommendations Systems

- Collaborative Filtering
- Content Based Filtering
- Hybrid - Combination of the Collaborative and Content Based
- Popularity based - Most bought, most watched, most downloaded, most heard

Collaborative Filtering

- Analyse User behaviour, Activities and preferences
- Recommend based on similarity to other user
- People who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past.



People who buy "x" also buy "y".

Content based Filtering

Business Problem

A travel booking website books thousands of hotel rooms every day.

With such a huge growth, the task at hand is to improve the booking per customer by showing the best hotels as per the taste and preferences of the user.

Sample Set

- Hotel 1 -Good Gym, Small Pool
- Hotel 2 - Amazing Gym, No Pool
- Hotel 3 - Great Gym, Small Pool
- Hotel 4- Basic Gym, Large Pool
- Hotel 5-No Gym, Large and Beautifully designed pool

- John - Prefers Gym over swimming pool
- Kavin - Likes Gym more than pool
- Bill - Needs a pool with basic Gym
- Frans - A pool is a must compared to Gym

Item – Feature Matrix

Hotel	Gym	Pool
Hotel 1	0.8	0.2
Hotel 2	1	0
Hotel 3	0.9	0.1
Hotel 4	0.1	0.9
Hotel 5	0	1

User – Feature Matrix

User	Gym	Pool
John	0.9	0.1
Kavin	0.8	0.2
Bill	0.3	0.7
Frans	0	1

Creating Feature Vector

Hotel	Gym	Pool
Hotel 1	0.8	0.2
Hotel 2	1	0
Hotel 3	0.9	0.1
Hotel 4	0.1	0.9
Hotel 5	0	1

Hotel	Feature Vector
H1	[0.8, 0.2]
H2	[1, 0]
H3	[0.9, 0.1]
H4	[0.1, 0.9]
H5	[0, 1]

User	Gym	Pool
John	0.9	0.1
Kavin	0.8	0.2
Bill	0.3	0.7
Frans	0	1

User	Feature Vector
U1	[0.9, 0.1]
U2	[0.8, 0.2]
U3	[0.3, 0.7]
U4	[0, 1]

Creating Feature Vector

Hotel	Feature Vector
H1	[0.8, 0.2]
H2	[1, 0]
H3	[0.9, 0.1]
H4	[0.1, 0.9]
H5	[0, 1]

User	Feature Vector
U1	[0.9, 0.1]
U2	[0.8, 0.2]
U3	[0.3, 0.7]
U4	[0, 1]

- Which Hotels should be recommended to John (U1)?
- MAX (Uj * Hi)

MAX (U1*H1, U1*H2, U1*H3, U1*H4, U1*H5)

MAX (0.74, 0.9, 0.82, 0.18, 0.1)

Recommended Hotels in the order of preference

H2, H3, H1

Collaborative Filtering

Workflow of CF, (Wikipedia)

- A user expresses his or her preferences by rating items.
- Ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.
- The system matches this user's ratings against other users' and finds the people with most "similar" tastes.
- With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user (presumably the absence of rating is often considered as the unfamiliarity of an item)

User	Hotel 1	Hotel 2	Hotel 3	Hotel 4	Hotel 5
John	4	5	?	?	1
Kavin	?	5	4.5	?	1
Bill	2	3	?	5	4
Frans	1	?	?	?	5

Collaborative Filtering

User	Hotel 1	Hotel 2	Hotel 3	Hotel 4	Hotel 5
John	4	5	?	?	1
Kavin	?	5	4.5	?	1
Bill	2	3	?	5	4
Frans	1	?	?	?	5

Matchbox Recommender

Recommendation Data Issues

- Cold Start
- Same Scale Judgement

Cold Start

- Content based filtering constructs user profile before system can recommend
- Collaborative Filtering needs like-minded users for recommendations
- Completely new profile or item
- Can be tackled using Hybrid recommendation

Same Scale judgement

- A user rates all the items on a same scale
- An item is rated on the same scale by all users

Recommender Split

- Fraction of training only users - Fraction of users assigned only to the training dataset. The rows would never be used to test the model.
- Fraction of test user ratings for training - Portion of the user ratings that can be used for training.
- Fraction of cold users - Cold users are users that the system has not previously encountered
- Fraction of cold items - Cold items are items that the system has not previously encountered.
- Fraction of ignored users - Specify the percentage of users that should be ignored.
- Fraction of ignored items - Specify the percentage of items to ignore.
- Remove occasionally produced cold items

DEMOS

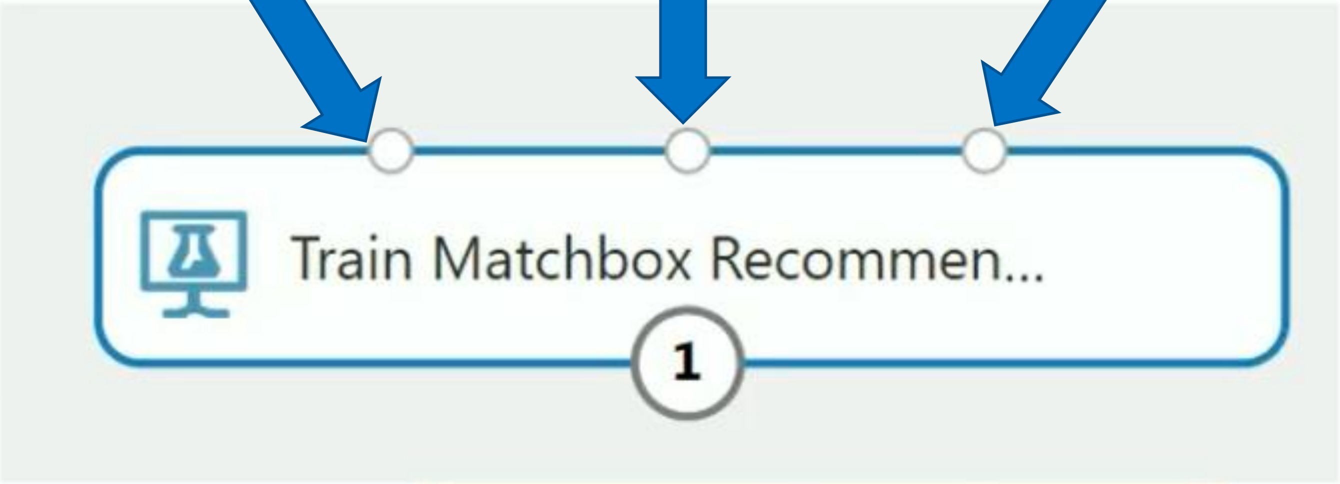
Matchbox Recommendation System

What is Matchbox Recommendation System

- Recommendation system developed by Microsoft Research
- Based on a the Hybrid Approach of Content and Collaborative filtering
- Uses Train Matchbox Recommender and Score Matchbox Recommender Modules
- Train Matchbox Recommender reads a dataset of user-item-rating triples and, optionally, some user and item features.

Configuring Train Matchbox Recommender

- User-Item-Rating Data - Prepare the data used for training, containing **user-item-rating** triples
  - First Column – User Identifier
  - Second Column – Item Identifier
  - Third Column – Rating for the user-item pair
- User-Feature – userID and the userfeatures
- Item-Feature – itemID and the Itemfeatures

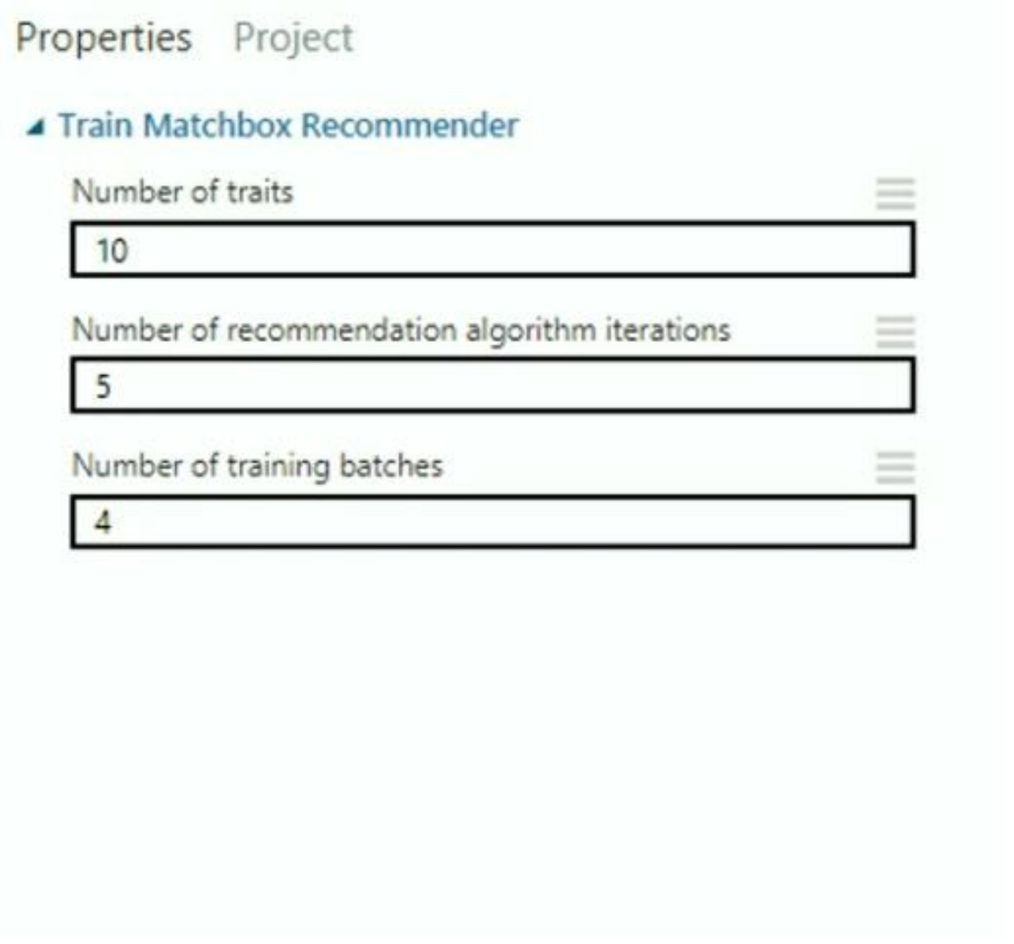


Parameters to Train Matchbox Recommender

- **Number of Traits**
  - How many traits to learn for each user and item
  - Each feature is associated with a latent “trait” vector

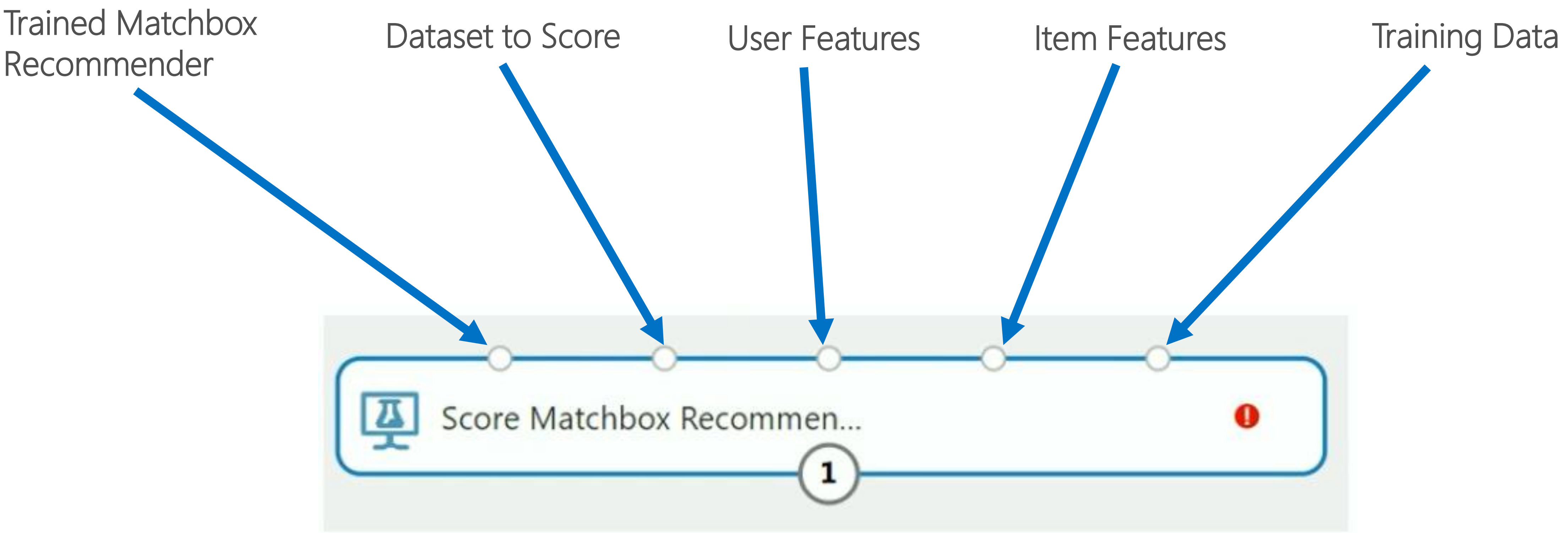
- **Number of recommendation algorithm iterations**
  - how many times the algorithm should process the input data

- **Number of training batches**
  - Number of batches for dividing the data during training



Score Matchbox Recommender

Input to Score Matchbox Recommender



Types of Prediction

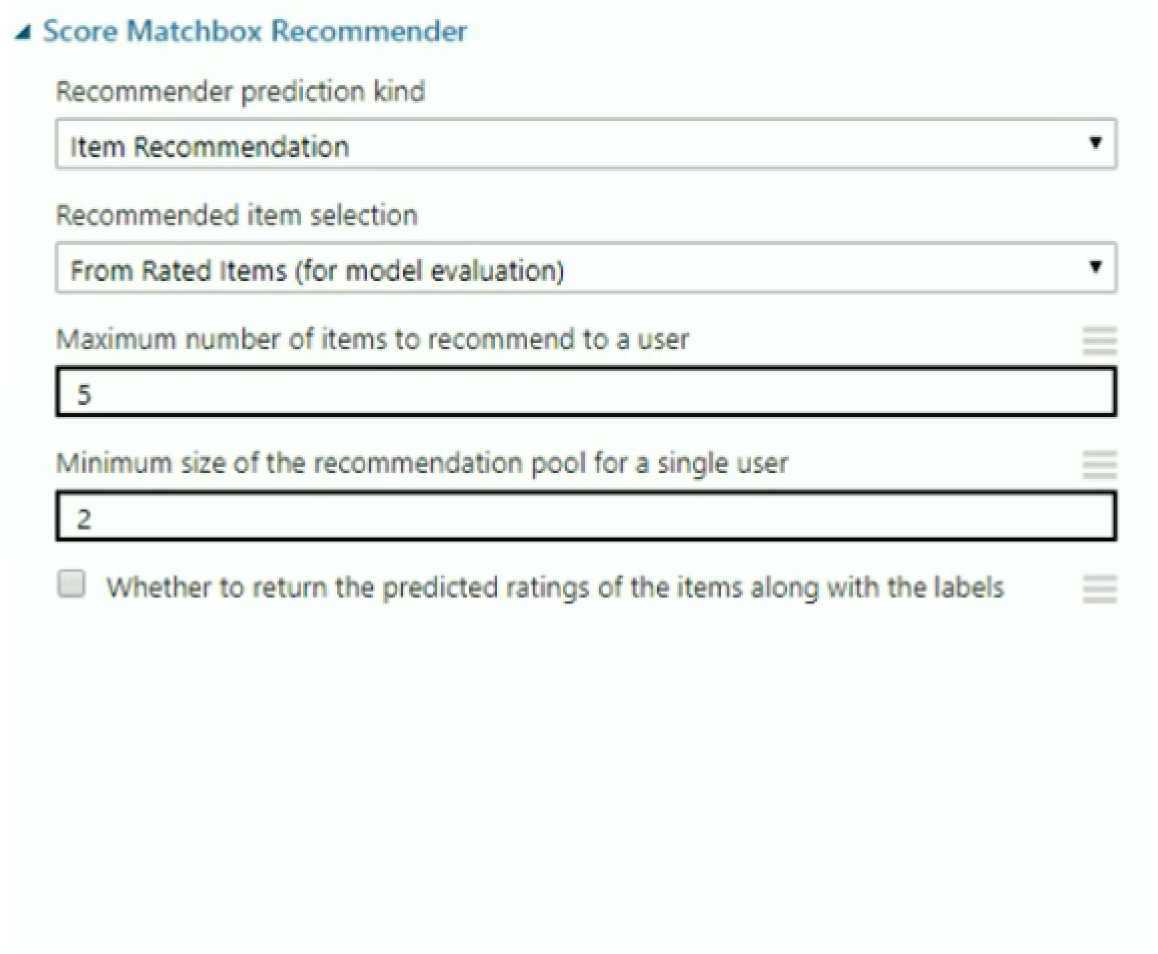
- Predict Ratings
- Item Recommendation
- Related Users
- Related Items

Ratings of Prediction

- Rating predictions by a user
- Input data must contain User and Item
- Does not require any parameter

Item Recommendation

- Users and Items as input
- Uses its knowledge about existing items and users
- Generates a list of items that will appeal to each user



Related Users and Items

- Can be used for "People like you" predictions
- Generate recommendations for users based on items that have already been rated

Understanding the Recommender Result

- For Item Recommendations
- Normalized Discounted Cumulative Gain (NDCG)

Ranking Quality

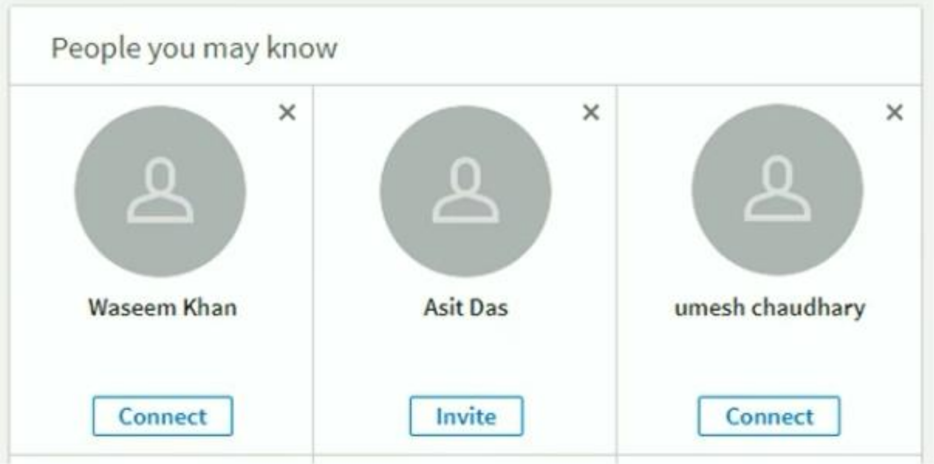
Search Result

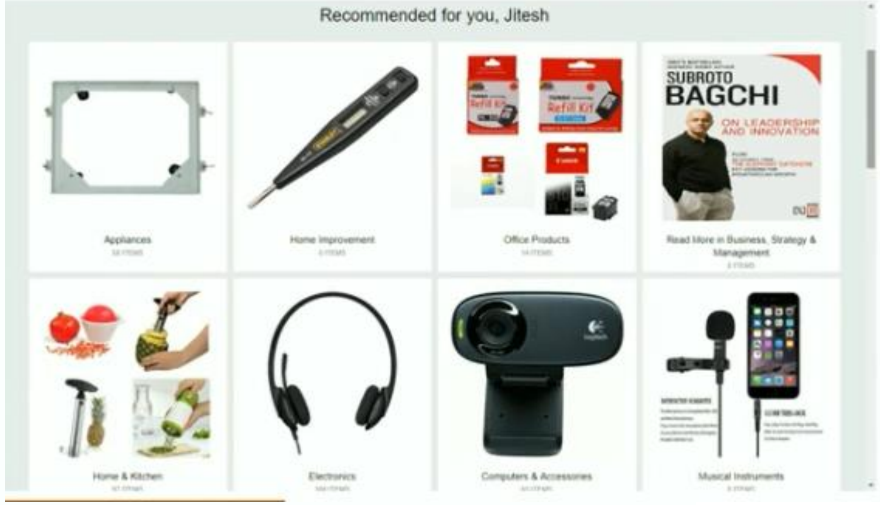
What is machine learning? - Definition from WhatIs.com
whatis.techtarget.com › Topics › AppDev › Programming ▾
Jun 24, 2017 - **Machine learning** is the field of study that gives computers the ability to learn without being explicitly programmed.

The 10 Algorithms Machine Learning Engineers Need to Know
www.kdnuggets.com/2016/08/10-algorithms-machine-learning-engineers.html ▾
Aug 8, 2016 - It is no doubt that the sub-field of **machine learning** / artificial intelligence has increasingly gained more popularity in the past couple of years.

Machine Learning | SAP - SAP.com
https://www.sap.com/india/trends/machine-learning.html ▾
Discover how AI, **machine learning**, and deep learning are powering a new breed of software that uses Big Data to drive radical changes to business.

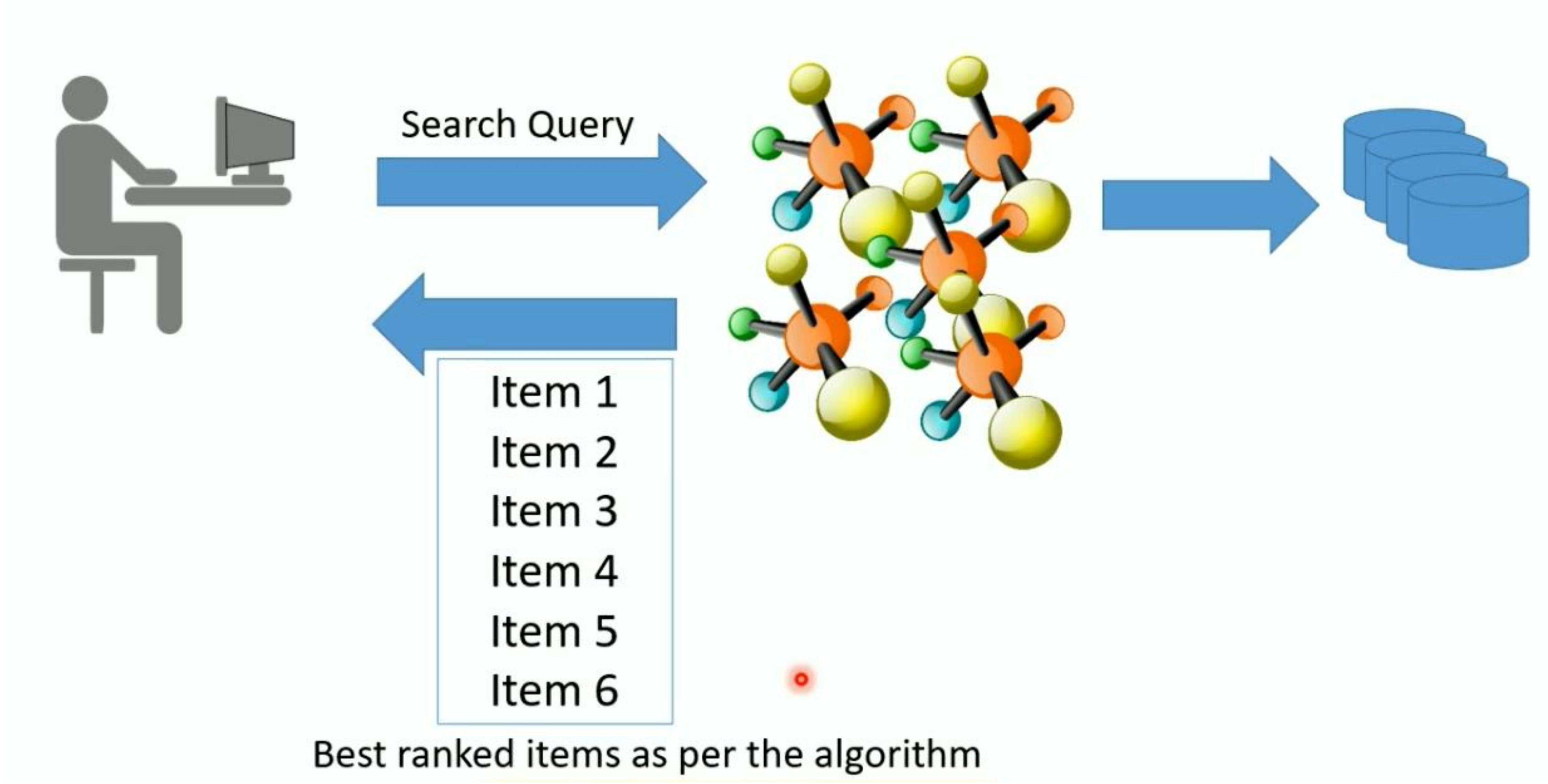
Machine Learning | edX
https://www.edx.org/course/machine-learning-columbiax-csmm-102x-0 ▾
Master the essentials of **machine learning** and algorithms to help improve learning from data without human intervention.





The highest ranked item should result in highest gain.

Ranking Quality



Discounted Cumulative Gain

Ranked items by the algorithms → I1, I2, I3, I4, I5, I6

Gain perceived by the user → 4, 3, 4, 0, 1, 2

$$CG = 4 + 3 + 4 + 0 + 1 + 2 = 14$$

i	rel_i	$\log_2(i+1)$	$\frac{rel_i}{\log_2(i+1)}$
1	4	1	4.00
2	3	1.585	1.89
3	4	2	2
4	0	2.322	0
5	1	2.585	0.39
6	2	2.81	0.71
		DCG	**8.99**

Ideal Discounted Cumulative Gain

Ideal Ranking by algorithm → I1, I3, I2, I6, I5, I4

Ideal Gain perceived by the user → 4, 4, 3, 2, 1, 0

CG = 4 + 4 + 3 + 2 + 1 + 0
= 14

i	rel_i	$\log_2 (i+1)$	$\dfrac{rel_i}{\log_2 (i+1)}$
1	4	1	4.00
2	4	1.585	2.52
3	3	2	1.5
4	2	2.322	0.86
5	1	2.585	0.38
6	0	2.81	0
		IDCG	**9.27**

Normalised DCG

DCG = 8.99 IDCG = 9.27

$$\text{NDCG} = \frac{\text{DCG}}{\text{IDCG}} = \frac{8.99}{9.27} = 0.9697$$

Highly relevant documents are more useful than marginally relevant documents, which are in turn more useful than non-relevant documents

Understanding the Result

- Normalized Discounted Cumulative Gain (NDCG)

Computer Vision Tasks

Image Classification

- Is there a deer in the image?

Object detection

- Where in the image is the deer?

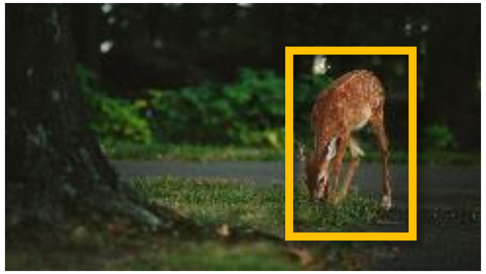


Image segmentation

- Where exactly is the deer? What pixels?

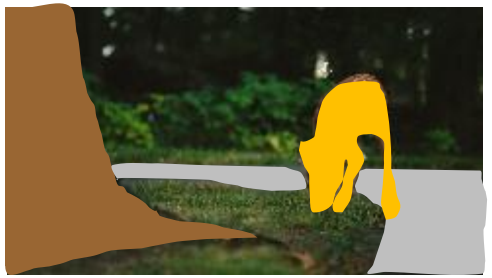


Image Similarity

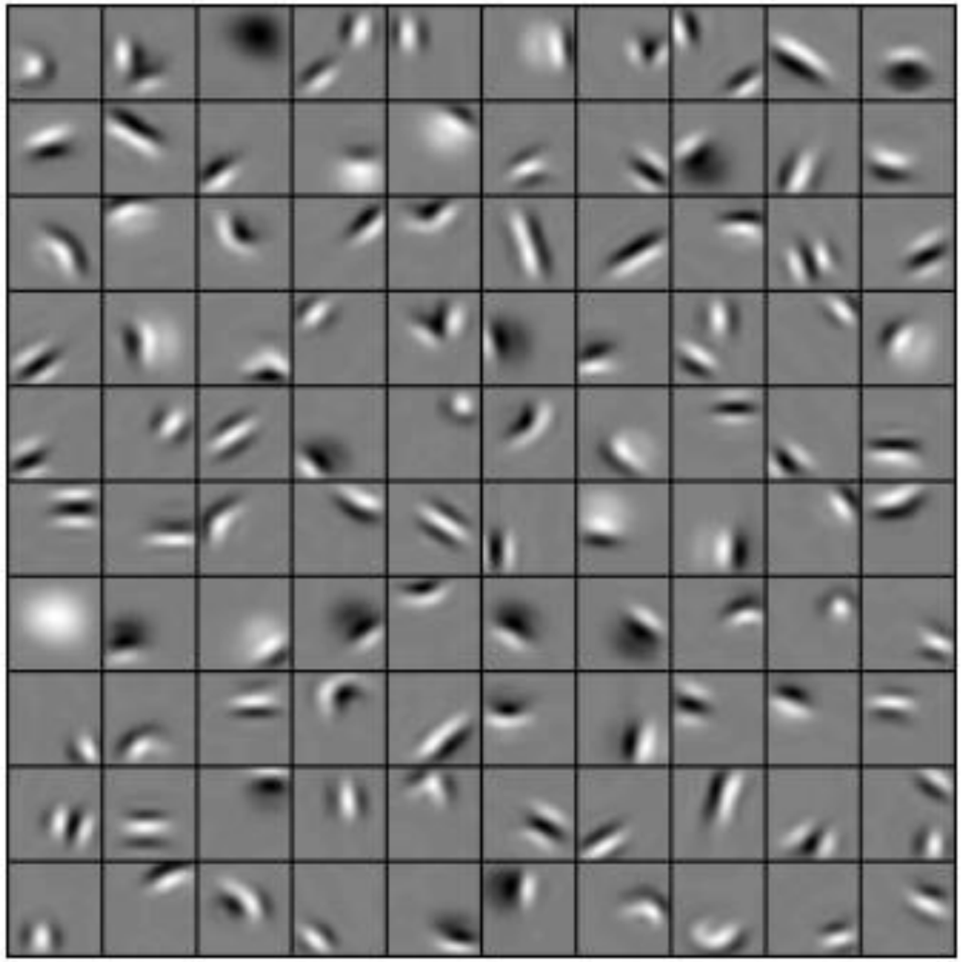
- Which images are similar to the query image?

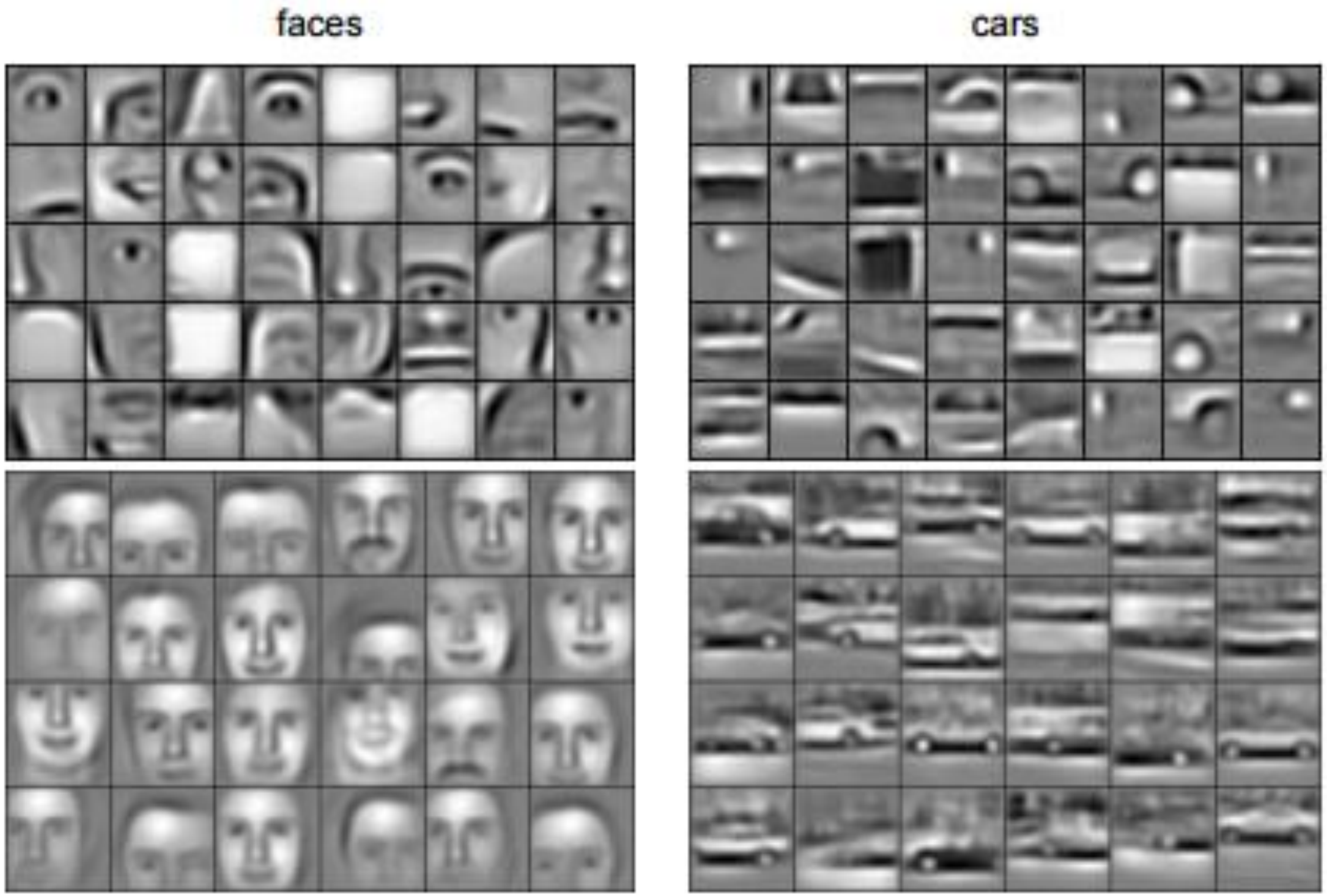
Before Deep Learning

- Researchers took a traditional machine learning approach
  - Manual creation of a variety of different visual feature extractors
  - Followed by traditional ML classifiers

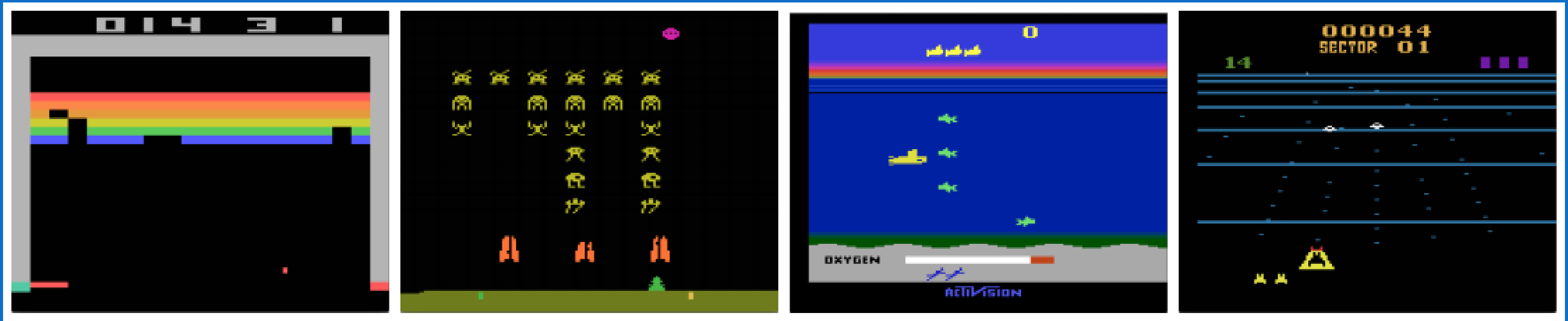
- Example: HoG Detectors
  - Histogram of oriented gradients (HoG) features
  - Sliding window detector
  - SVM Classifier
  - Very fast OpenCV implementation (<100ms)

How it Works:
Convolutional Neural Networks

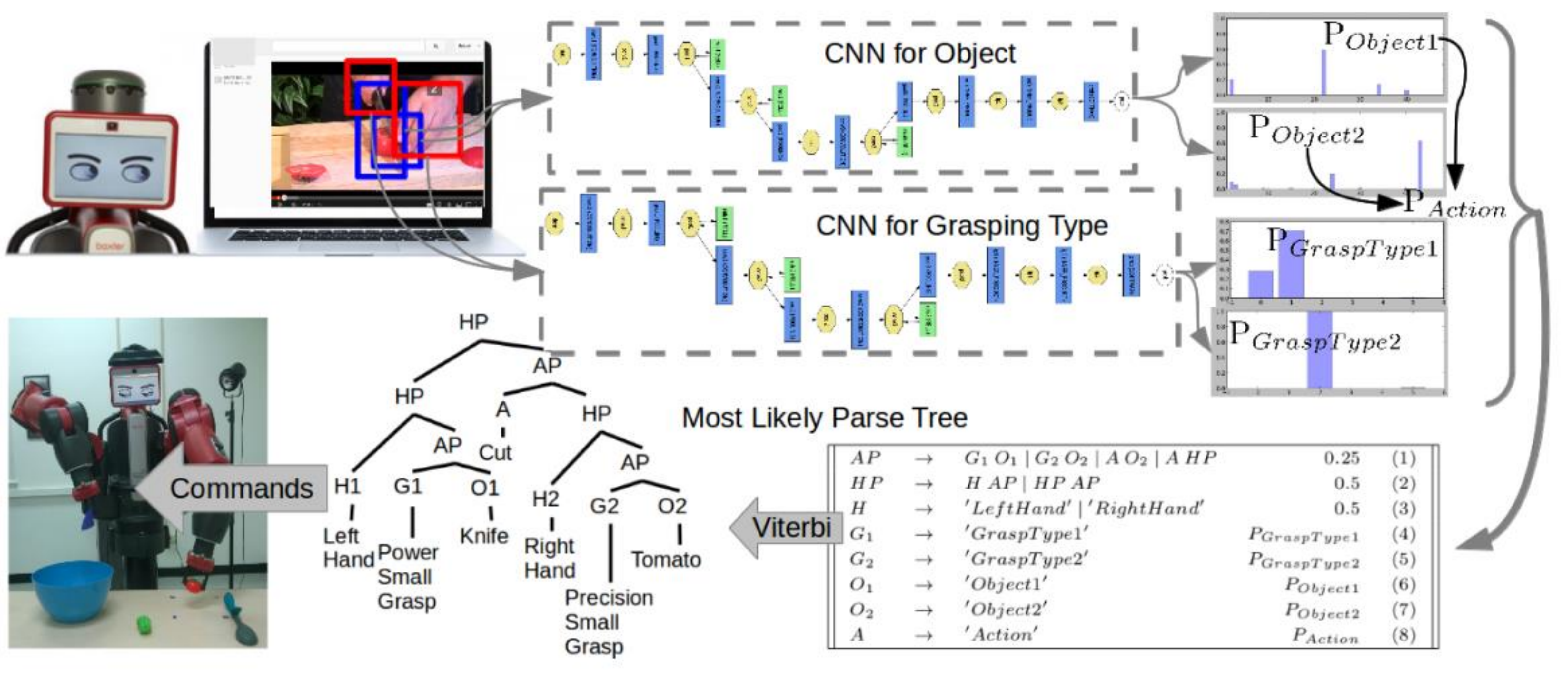




[Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations Honglak Lee, Roger Grosse, Rajesh Ranganath, Andrew Y. Ng]



[Playing Atari with Deep Reinforcement Learning.
Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller]



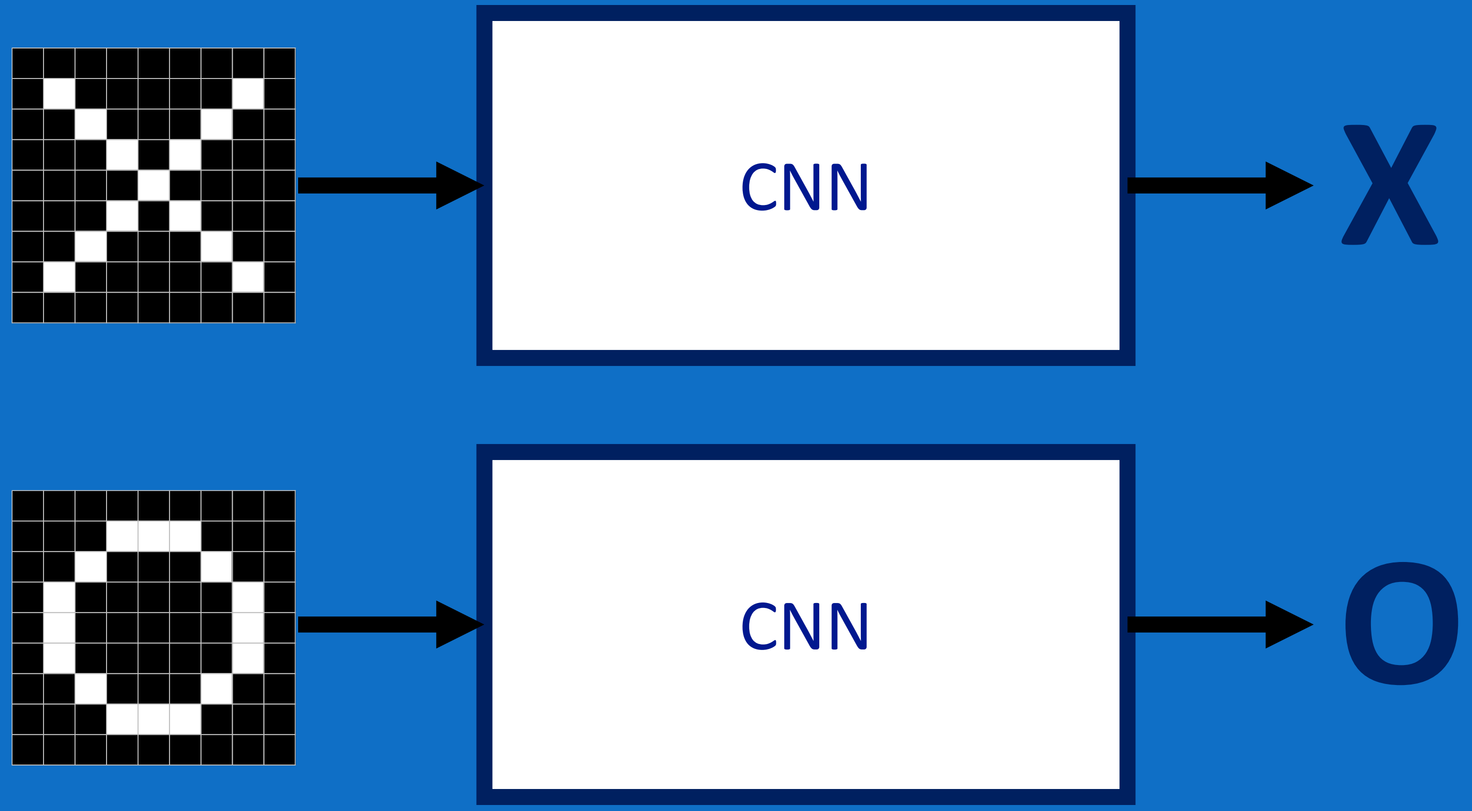
Robot Learning ManipulationAction Plans by “Watching” Unconstrained Videos from the World Wide Web.
Yezhou Yang, Cornelia Fermuller, Yiannis Aloimonos

A toy ConvNet: X’s and O’s

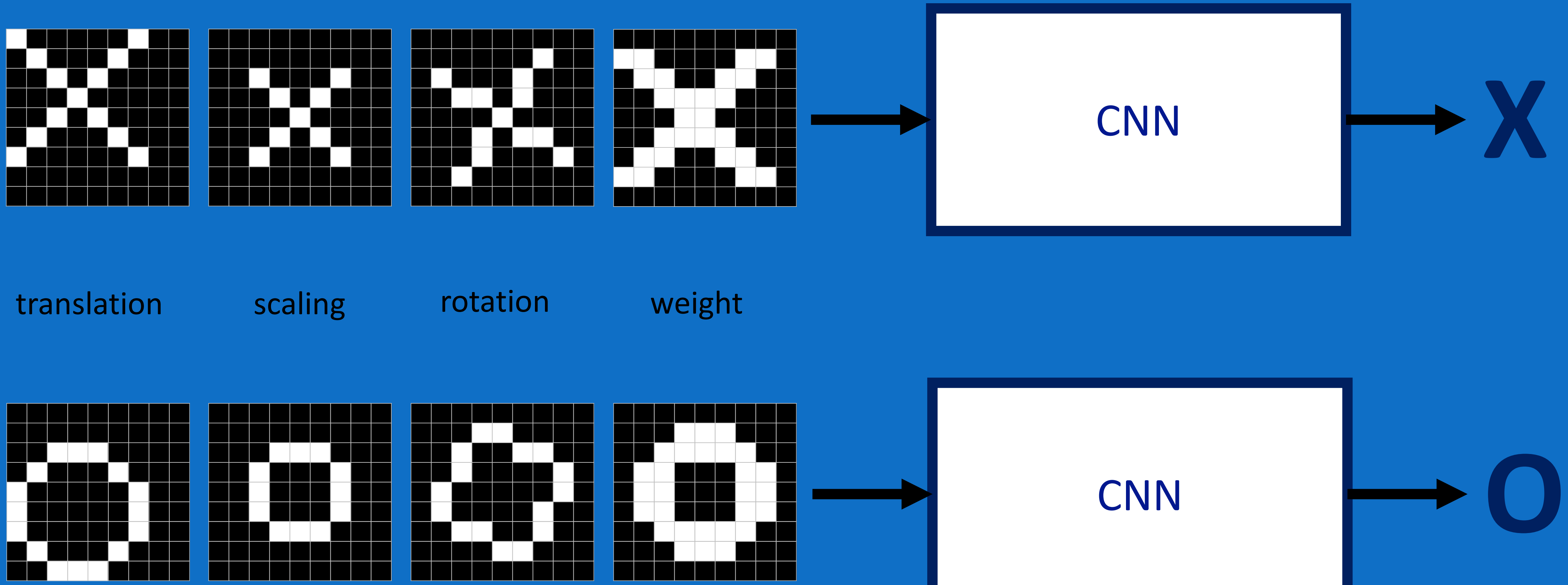
Says whether a picture is of an X or an O

A two-dimensional array of pixels

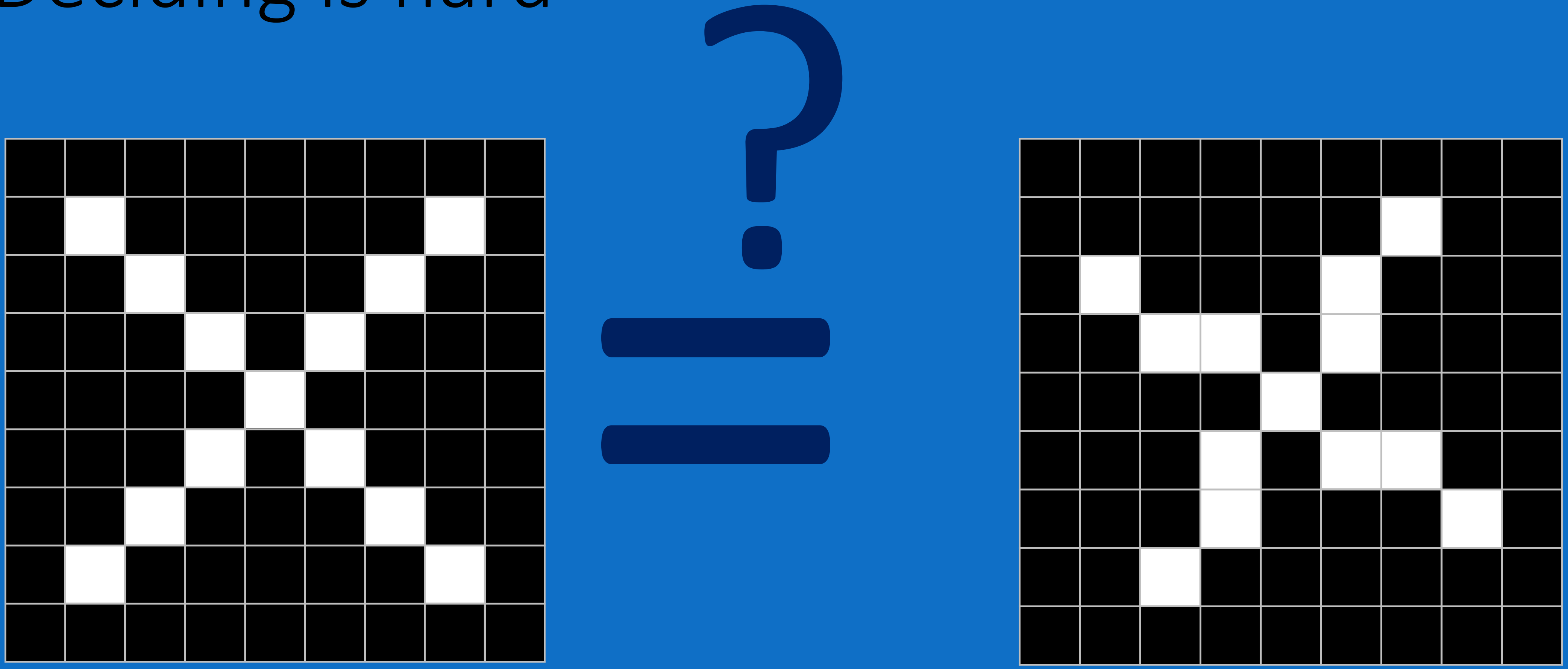
For example



Trickier cases



Deciding is hard



What computers see

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

$\stackrel{?}{=}$

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	1	-1	-1	-1
-1	-1	1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	1	-1	-1
-1	-1	-1	1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

What computers see

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	X	-1	-1	-1	-1	X	X	-1
-1	X	X	-1	-1	X	X	-1	-1
-1	-1	X	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	X	-1	-1
-1	-1	X	X	-1	-1	X	X	-1
-1	X	X	-1	-1	-1	-1	X	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

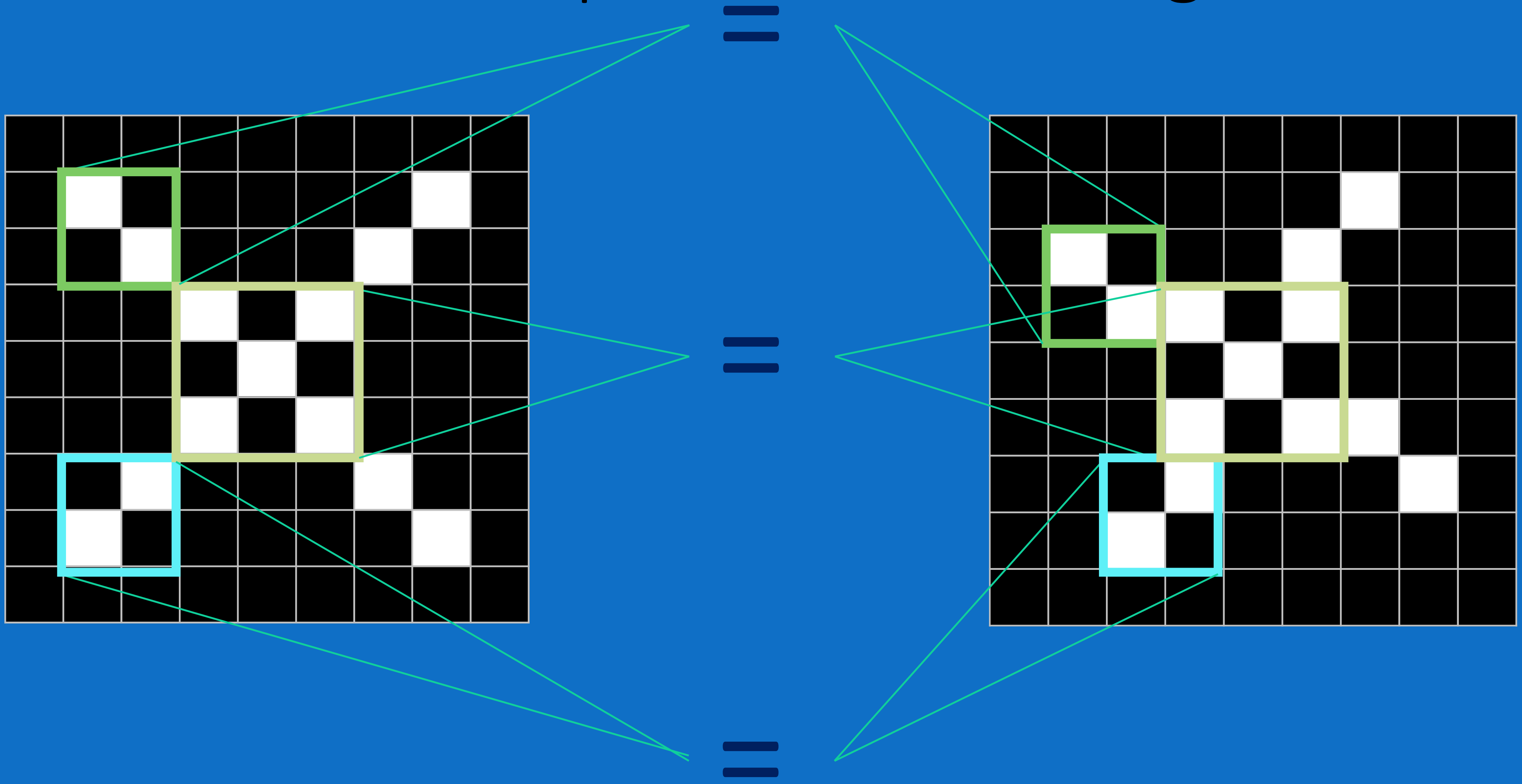
Computers are literal

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

$\neq$

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	1	-1	-1	-1
-1	-1	1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	1	-1	-1
-1	-1	-1	1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

ConvNets match pieces of the image

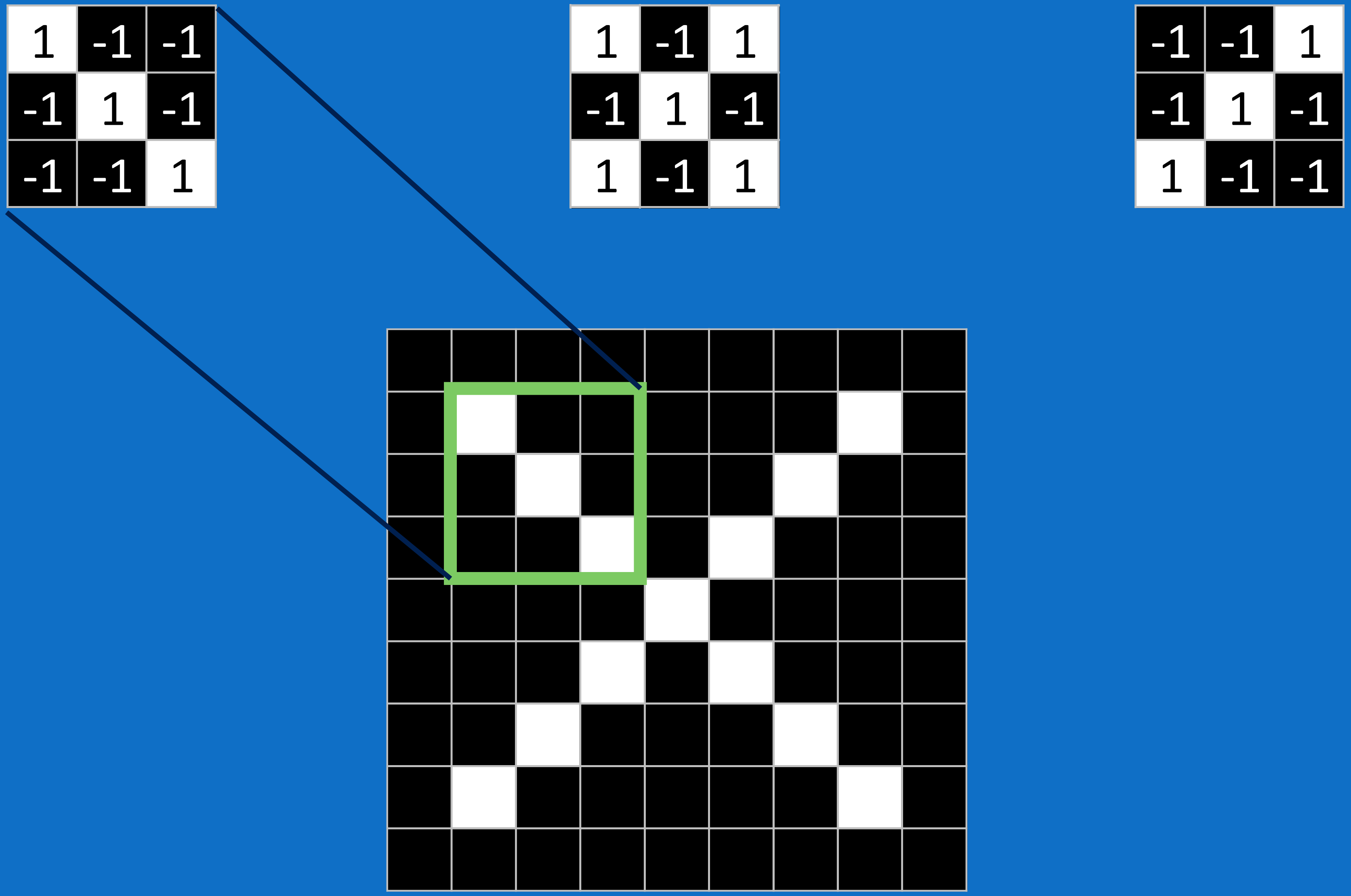


Features match pieces of the image

1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

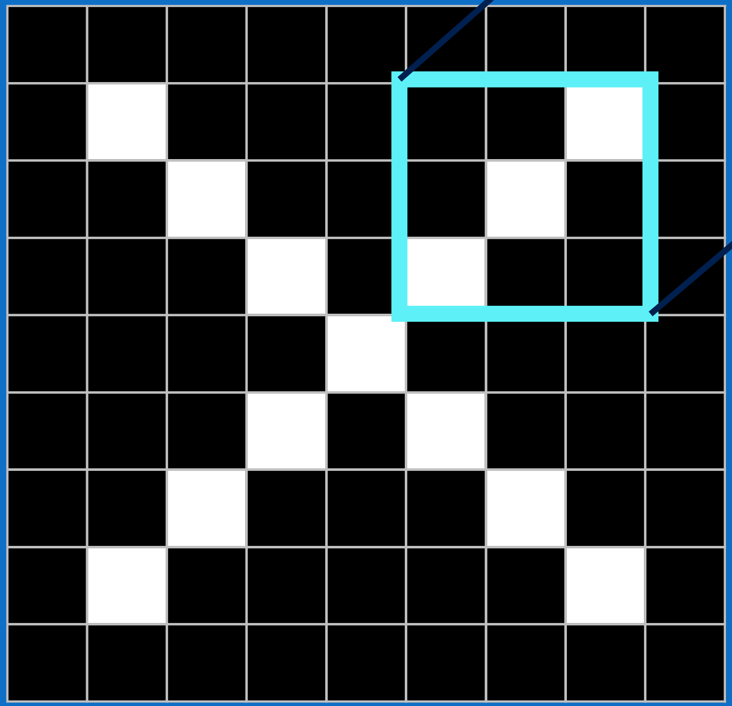
-1	-1	1
-1	1	-1
1	-1	-1


1 -1 -1
-1 1 -1
-1 -1 1
1 -1 1
-1 1 -1
1 -1 1
-1 -1 1
-1 1 -1
1 -1 -1

1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

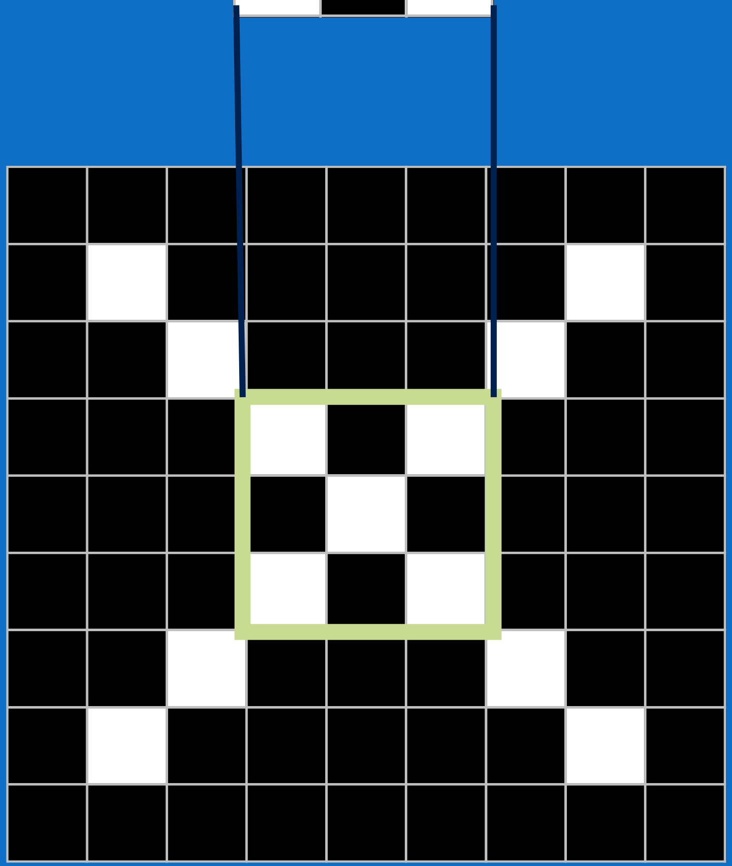
-1	-1	1
-1	1	-1
1	-1	-1



1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

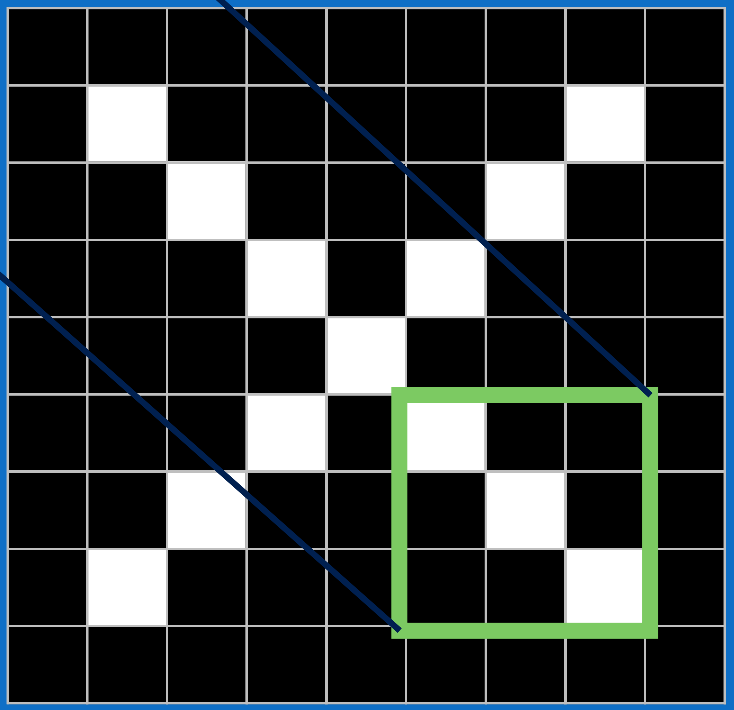
-1	-1	1
-1	1	-1
1	-1	-1



1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

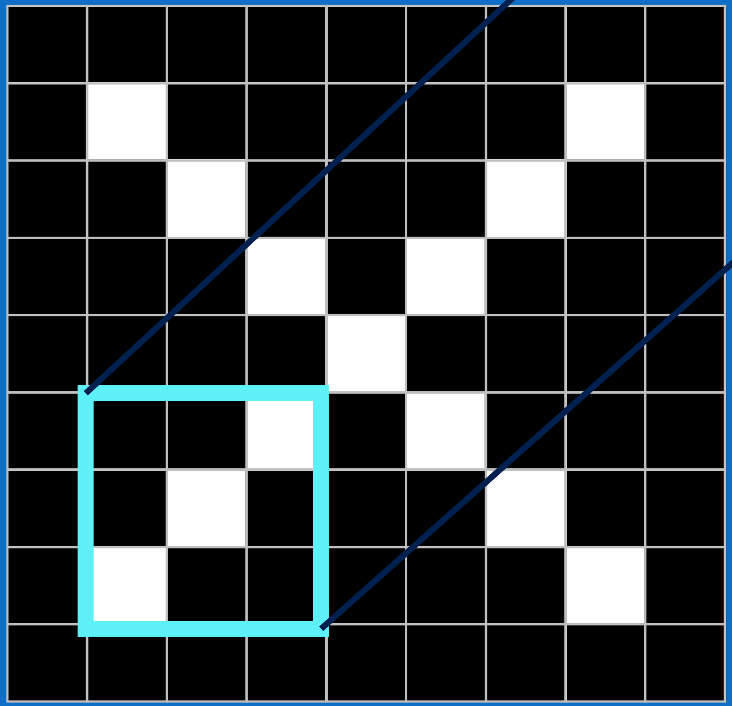
-1	-1	1
-1	1	-1
1	-1	-1



1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

-1	-1	1
-1	1	-1
1	-1	-1



Filtering: The math behind the match

1	-1	-1
-1	1	-1
-1	-1	1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

DON'T PANIC

Filtering: The math behind the match

1. Line up the feature and the image patch.
2. Multiply each image pixel by the corresponding feature pixel.
3. Add them up.
4. Divide by the total number of pixels in the feature.

Filtering: The math behind the match

1	-1	-1
-1	1	-1
-1	-1	1

$\boxed{1} \times \boxed{1} = 1$

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Filtering: The math behind the match

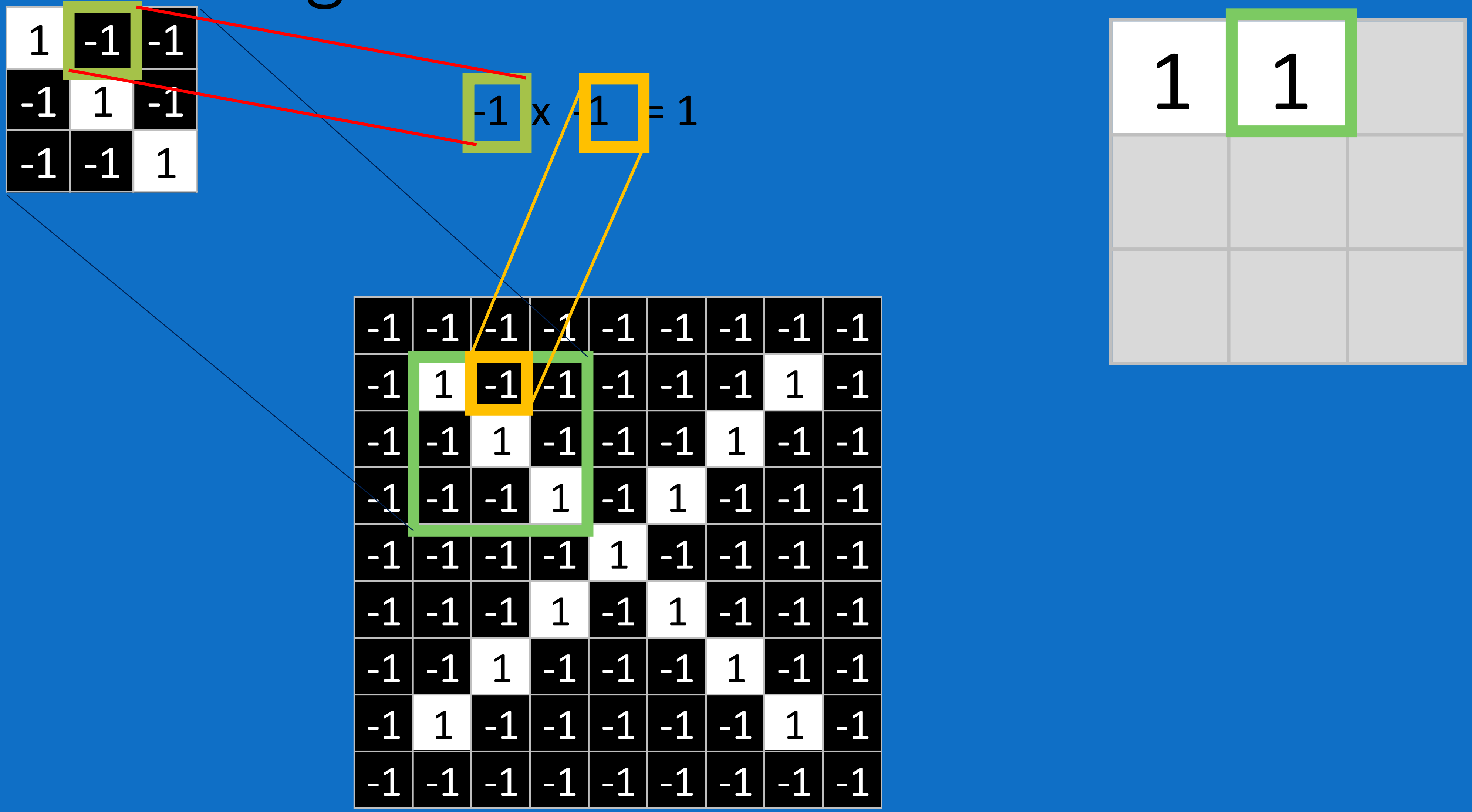
1	-1	-1
-1	1	-1
-1	-1	1

1 x 1 = 1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

1		

Filtering: The math behind the match



Filtering: The math behind the match

1	-1	-1
-1	1	-1
-1	-1	1

$-1 \times -1 = 1$

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

1	1	1

Filtering: The math behind the match

1	-1	-1
-1	1	-1
-1	-1	1

-1 x -1 = 1

1	1	1
1		

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Filtering: The math behind the match

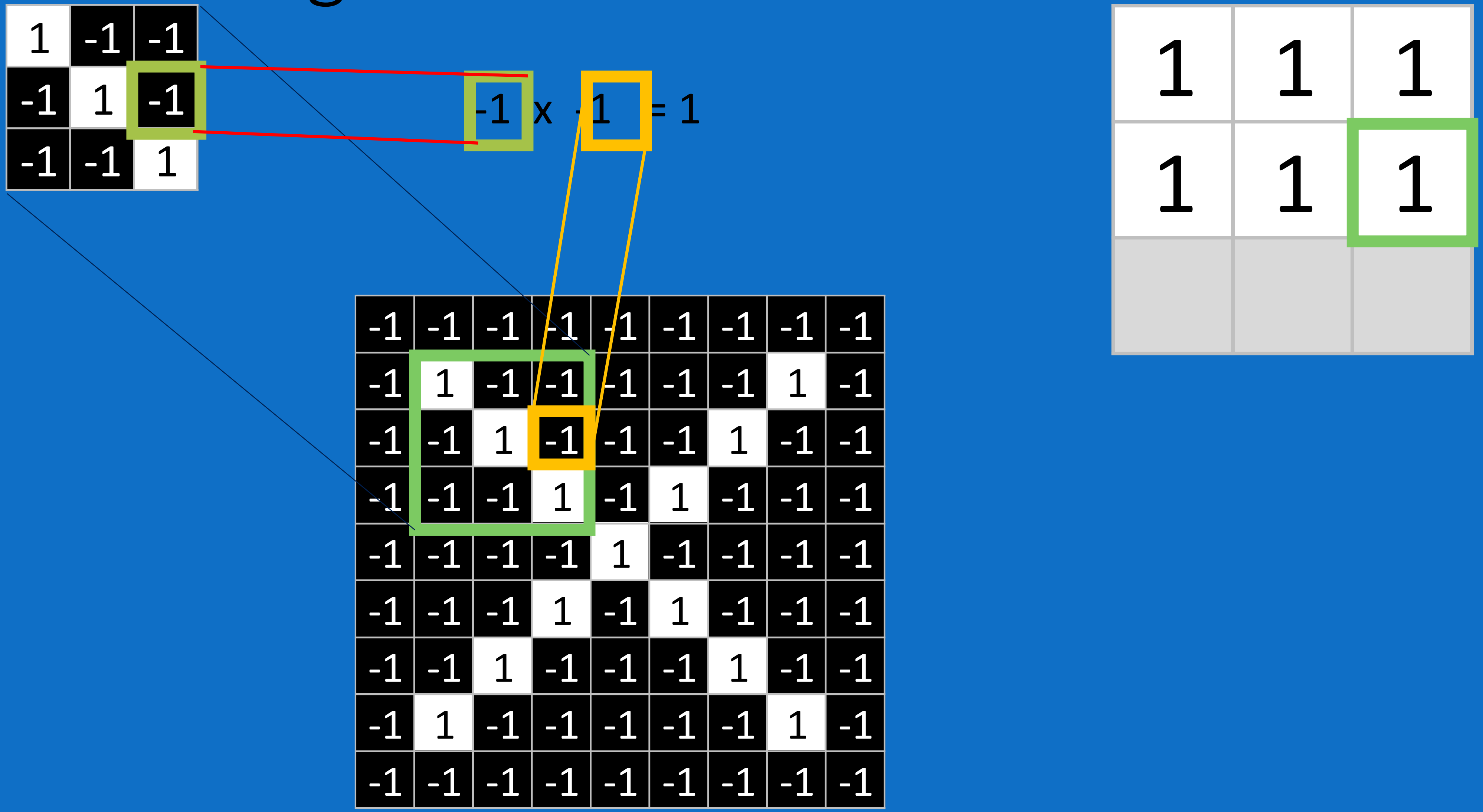
1	-1	-1
-1	1	-1
-1	-1	1

$1 \times 1 = 1$

1	1	1
1	1	

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Filtering: The math behind the match



Filtering: The math behind the match

1	-1	-1
-1	1	-1
-1	-1	1

-1 x -1 = 1

1	1	1
1	1	1
1		

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Filtering: The math behind the match

1	-1	-1
-1	1	-1
-1	-1	1

$-1 \times -1 = 1$

1	1	1
1	1	1
1	1	

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Filtering: The math behind the match

1	-1	-1
-1	1	-1
-1	-1	1

1 x 1 = 1

1	1	1
1	1	1
1	1	1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Filtering: The math behind the match

1	-1	-1
-1	1	-1
-1	-1	1

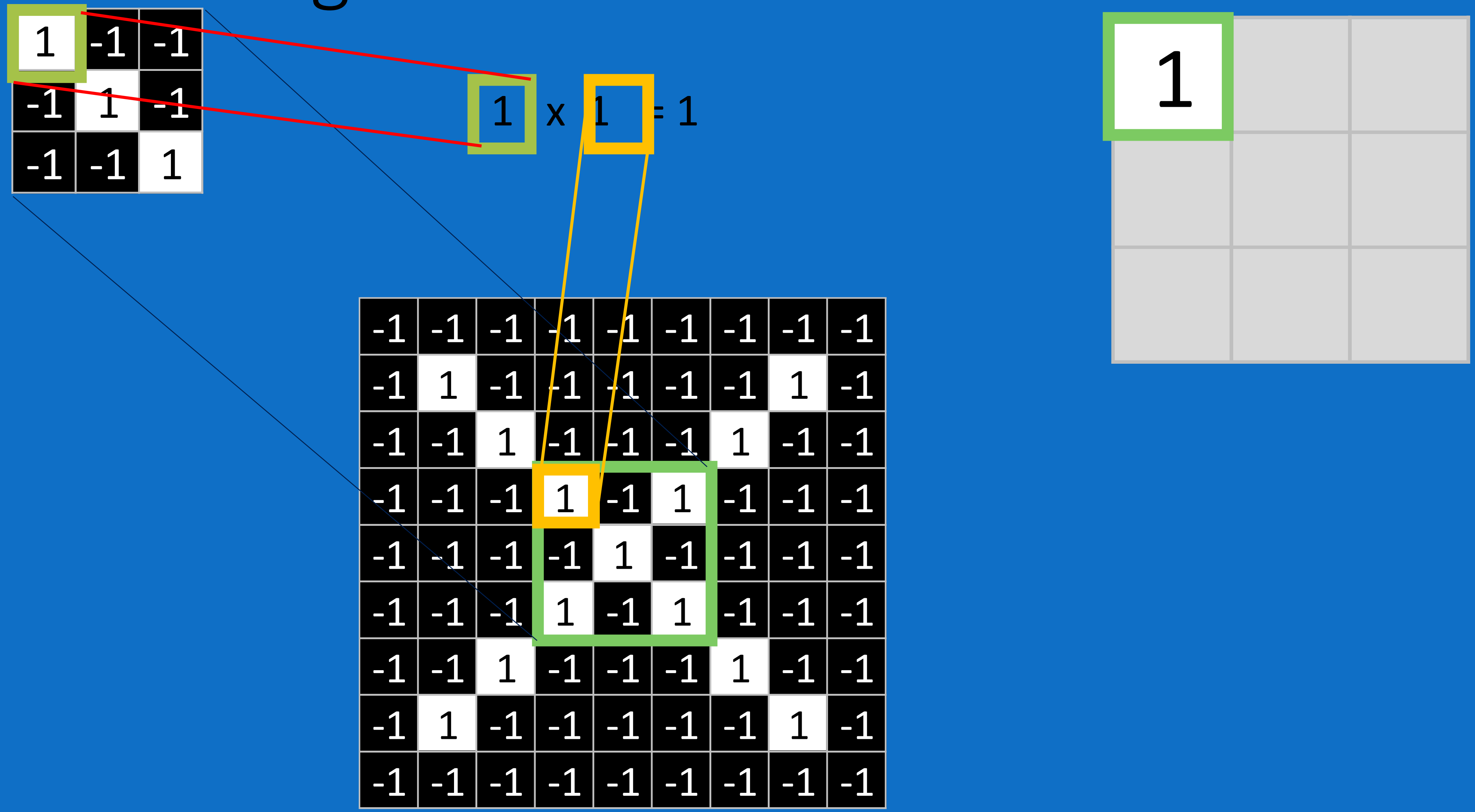
1	1	1
1	1	1
1	1	1

$$\frac{1+1+1+1+1+1+1+1+1}{9} = \boxed{1}$$

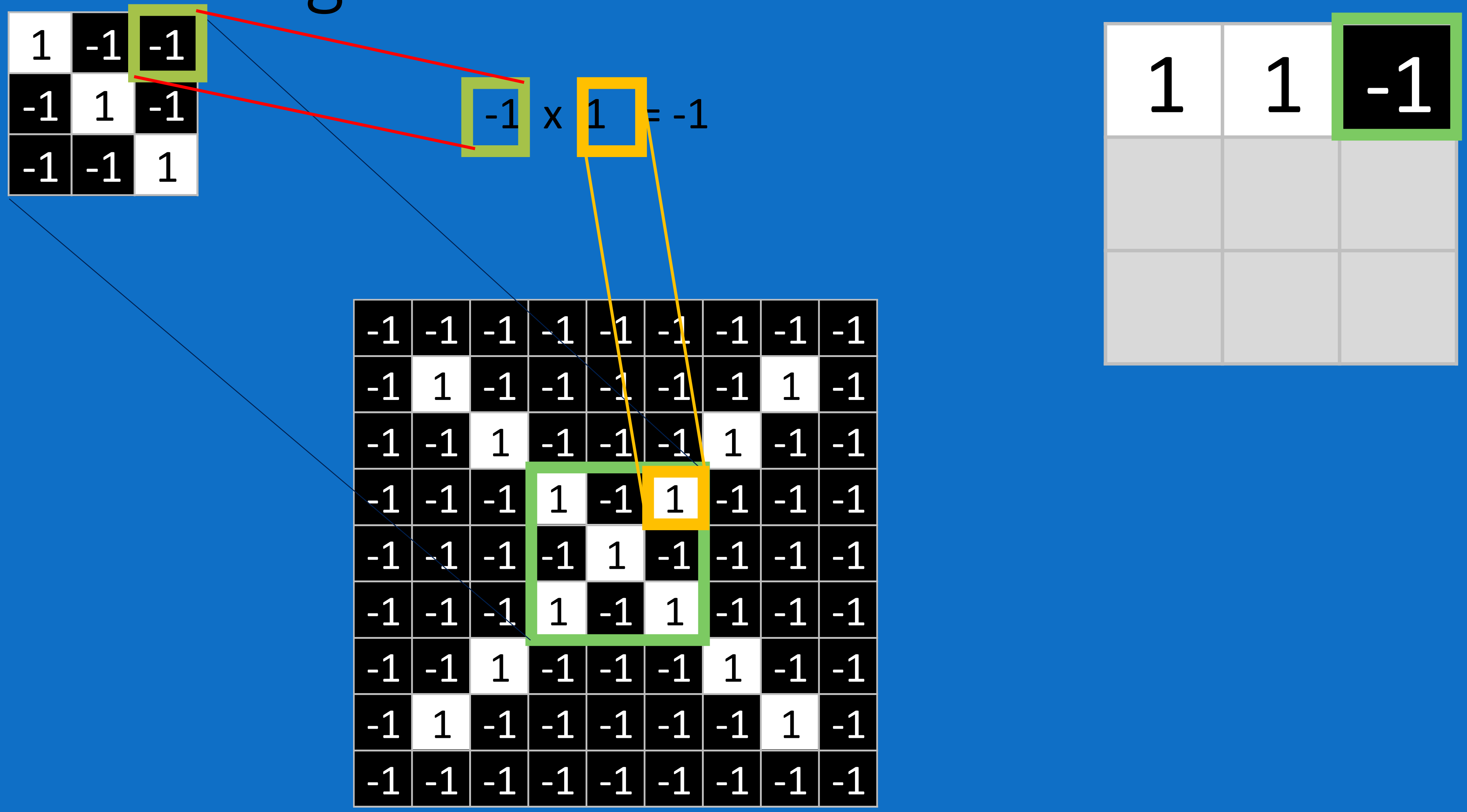
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

		1						

Filtering: The math behind the match



Filtering: The math behind the match



Filtering: The math behind the match

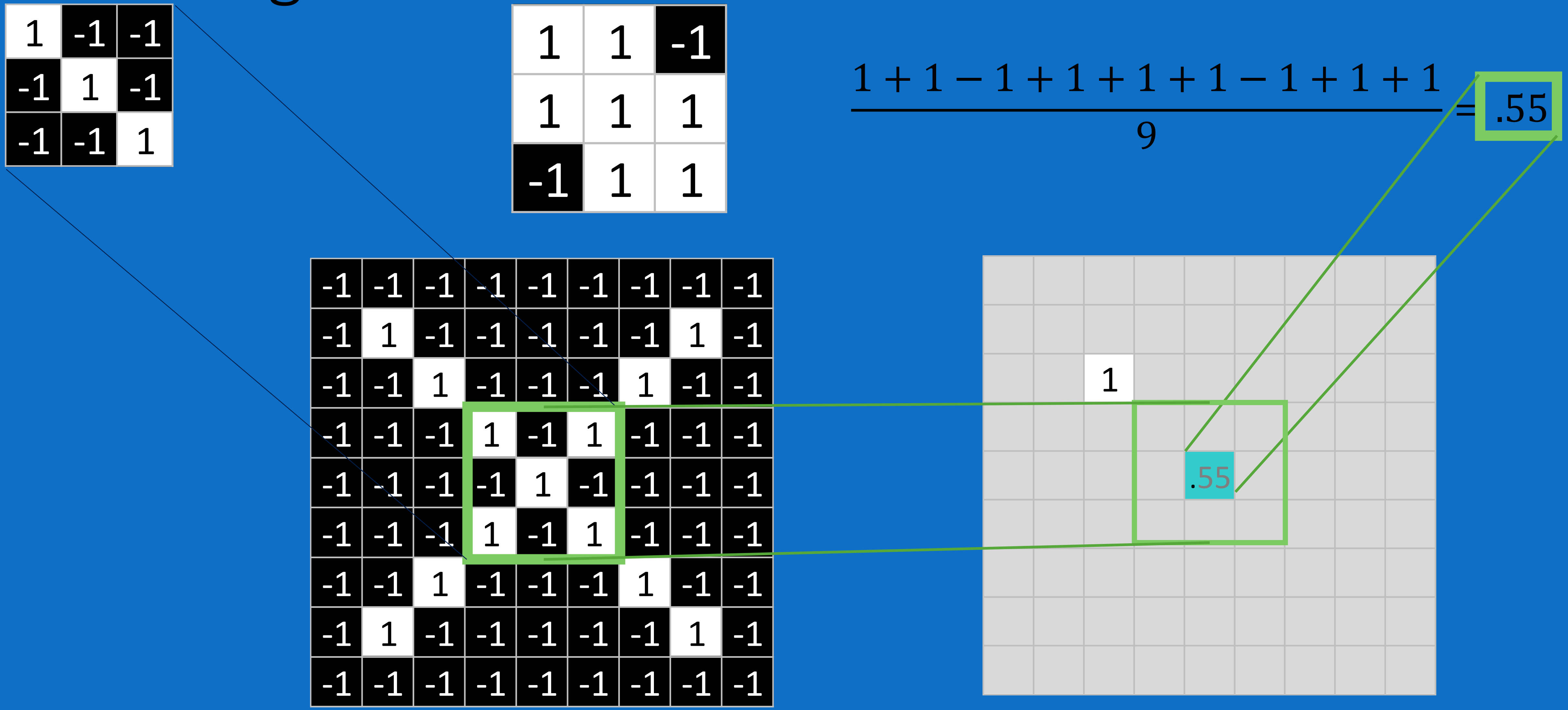
1	-1	-1
-1	1	-1
-1	-1	1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

1	1	-1
1	1	1
-1	1	1

Filtering: The math behind the match

$$\frac{1+1-1+1+1+1-1+1+1}{9} = .55$$



Convolution: Trying every possible match

1	-1	-1
-1	1	-1
-1	-1	1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

Convolution: Trying every possible match

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

$\otimes$

1	-1	-1
-1	1	-1
-1	-1	1

$=$

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

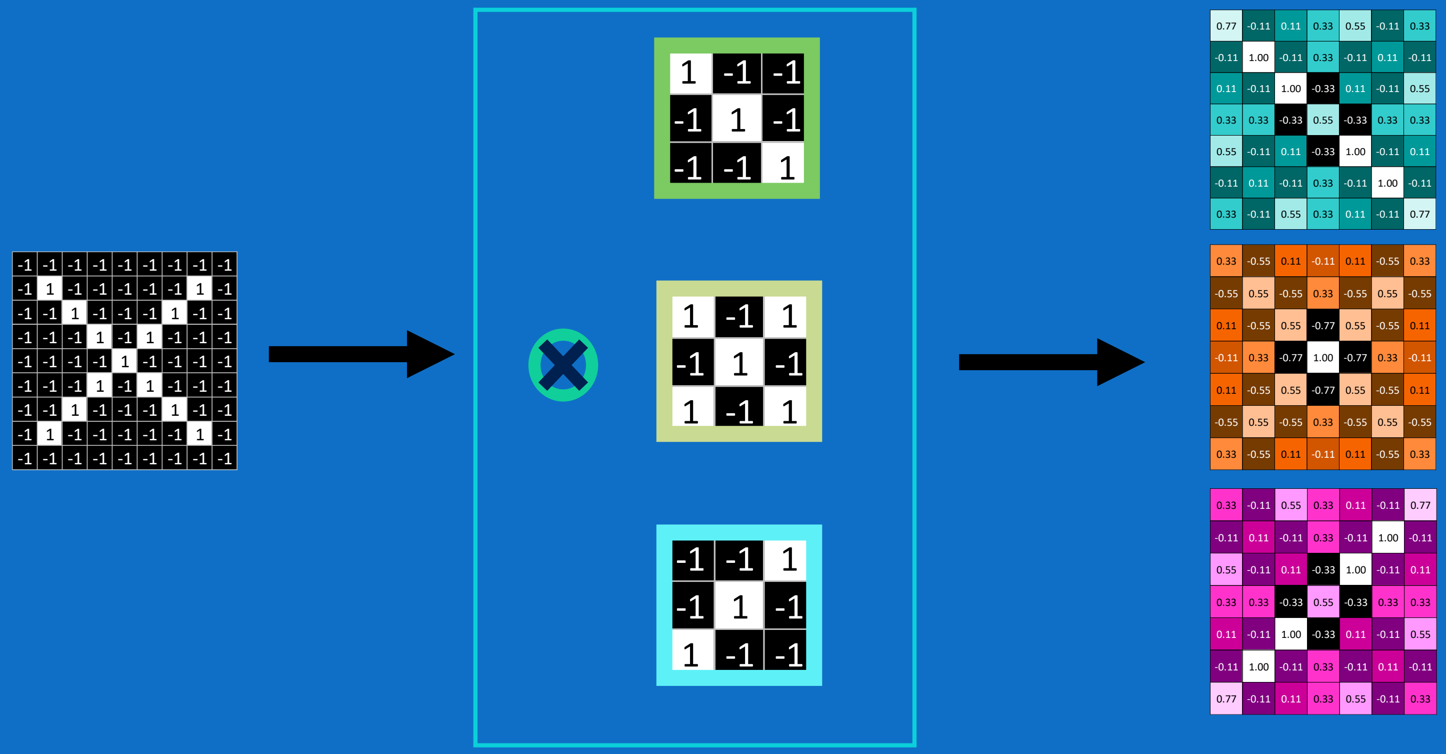
$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 0.77 & -0.11 & 0.11 & 0.33 & 0.55 & -0.11 & 0.33 \\ -0.11 & 1.00 & -0.11 & 0.33 & -0.11 & 0.11 & -0.11 \\ 0.11 & -0.11 & 1.00 & -0.33 & 0.11 & -0.11 & 0.55 \\ 0.33 & 0.33 & -0.33 & 0.55 & -0.33 & 0.33 & 0.33 \\ 0.55 & -0.11 & 0.11 & -0.33 & 1.00 & -0.11 & 0.11 \\ -0.11 & 0.11 & -0.11 & 0.33 & -0.11 & 1.00 & -0.11 \\ 0.33 & -0.11 & 0.55 & 0.33 & 0.11 & -0.11 & 0.77 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 0.33 & -0.55 & 0.11 & -0.11 & 0.11 & -0.55 & 0.33 \\ -0.55 & 0.55 & -0.55 & 0.33 & -0.55 & 0.55 & -0.55 \\ 0.11 & -0.55 & 0.55 & -0.77 & 0.55 & -0.55 & 0.11 \\ -0.11 & 0.33 & -0.77 & 1.00 & -0.77 & 0.33 & -0.11 \\ 0.11 & -0.55 & 0.55 & -0.77 & 0.55 & -0.55 & 0.11 \\ -0.55 & 0.55 & -0.55 & 0.33 & -0.55 & 0.55 & -0.55 \\ 0.33 & -0.55 & 0.11 & -0.11 & 0.11 & -0.55 & 0.33 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \otimes \begin{bmatrix} -1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 0.33 & -0.11 & 0.55 & 0.33 & 0.11 & -0.11 & 0.77 \\ -0.11 & 0.11 & -0.11 & 0.33 & -0.11 & 1.00 & -0.11 \\ 0.55 & -0.11 & 0.11 & -0.33 & 1.00 & -0.11 & 0.11 \\ 0.33 & 0.33 & -0.33 & 0.55 & -0.33 & 0.33 & 0.33 \\ 0.11 & -0.11 & 1.00 & -0.33 & 0.11 & -0.11 & 0.55 \\ -0.11 & 1.00 & -0.11 & 0.33 & -0.11 & 0.11 & -0.11 \\ 0.77 & -0.11 & 0.11 & 0.33 & 0.55 & -0.11 & 0.33 \end{bmatrix}$$

Convolution layer

One image becomes a stack of filtered images



Convolution layer

One image becomes a stack of filtered images

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.11	0.33	-0.77	1.00	-0.77	0.33	-0.11
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33

0.33	-0.11	0.55	0.33	0.11	-0.11	0.77
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33

Pooling: Shrinking the image stack

1. Pick a window size (usually 2 or 3).
2. Pick a stride (usually 2).
3. Walk your window across your filtered images.
4. From each window, take the maximum value.

Pooling

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

maximum

1.00			

Pooling

maximum

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

1.00	0.33		

Pooling

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

maximum

1.00	0.33	0.55	

Pooling

maximum

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

1.00	0.33	0.55	0.33

Pooling

maximum

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

1.00	0.33	0.55	0.33
0.33			

Pooling

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

max pooling

1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.11	0.33	-0.77	1.00	-0.77	0.33	-0.11
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33

0.55	0.33	0.55	0.33
0.33	1.00	0.55	0.11
0.55	0.55	0.55	0.11
0.33	0.11	0.11	0.33

0.33	-0.11	0.55	0.33	0.11	-0.11	0.77
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33

0.33	0.55	1.00	0.77
0.55	0.55	1.00	0.33
1.00	1.00	0.11	0.55
0.77	0.33	0.55	0.33

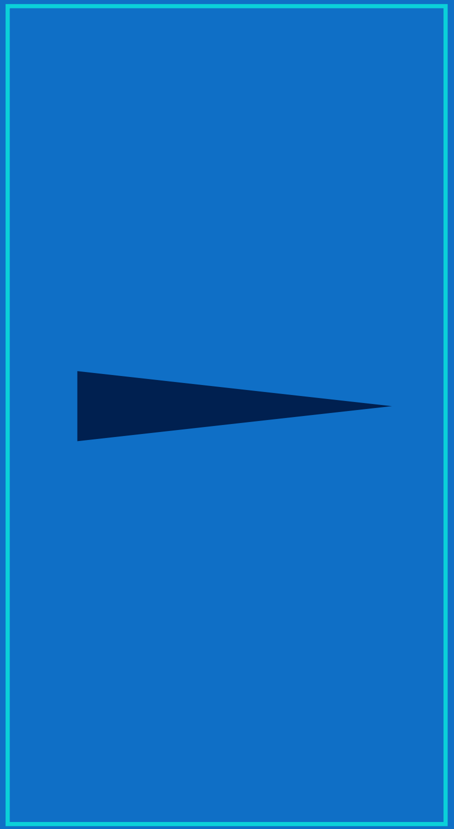
Pooling layer

A stack of images becomes a stack of smaller images.

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.11	0.33	-0.77	1.00	-0.77	0.33	-0.11
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33

0.33	-0.11	0.55	0.33	0.11	-0.11	0.77
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33



1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

0.55	0.33	0.55	0.33
0.33	1.00	0.55	0.11
0.55	0.55	0.55	0.11
0.33	0.11	0.11	0.33

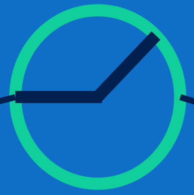
0.33	0.55	1.00	0.77
0.55	0.55	1.00	0.33
1.00	1.00	0.11	0.55
0.77	0.33	0.55	0.33

Normalization

Keep the math from breaking by tweaking each of the values just a bit.

Change everything negative to zero.

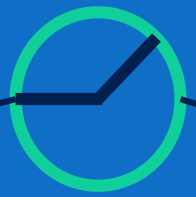
Rectified Linear Units (ReLUs)



0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.77						

Rectified Linear Units (ReLUs)



0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.77	0					

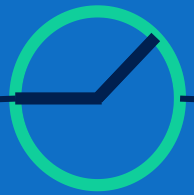
Rectified Linear Units (ReLUs)

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.77	0	0.11	0.33	0.55	0	0.33

Rectified Linear Units (ReLUs)

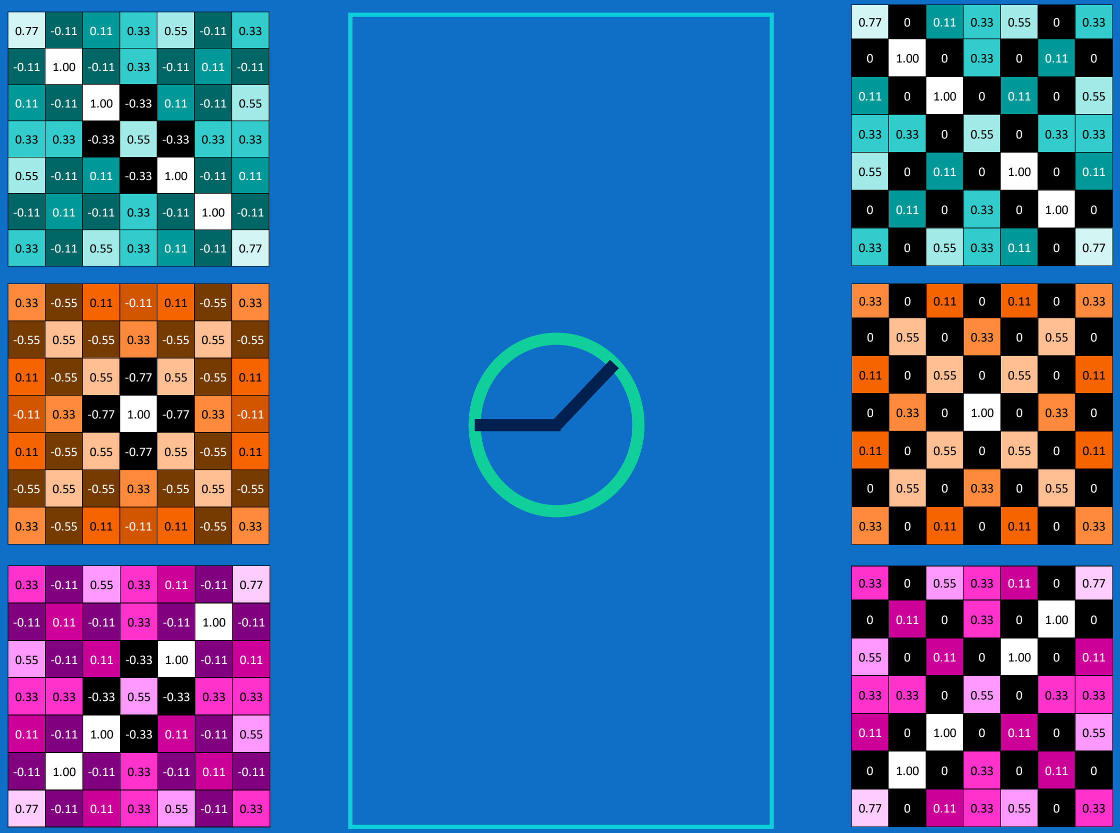
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77



0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	0.77

ReLU layer

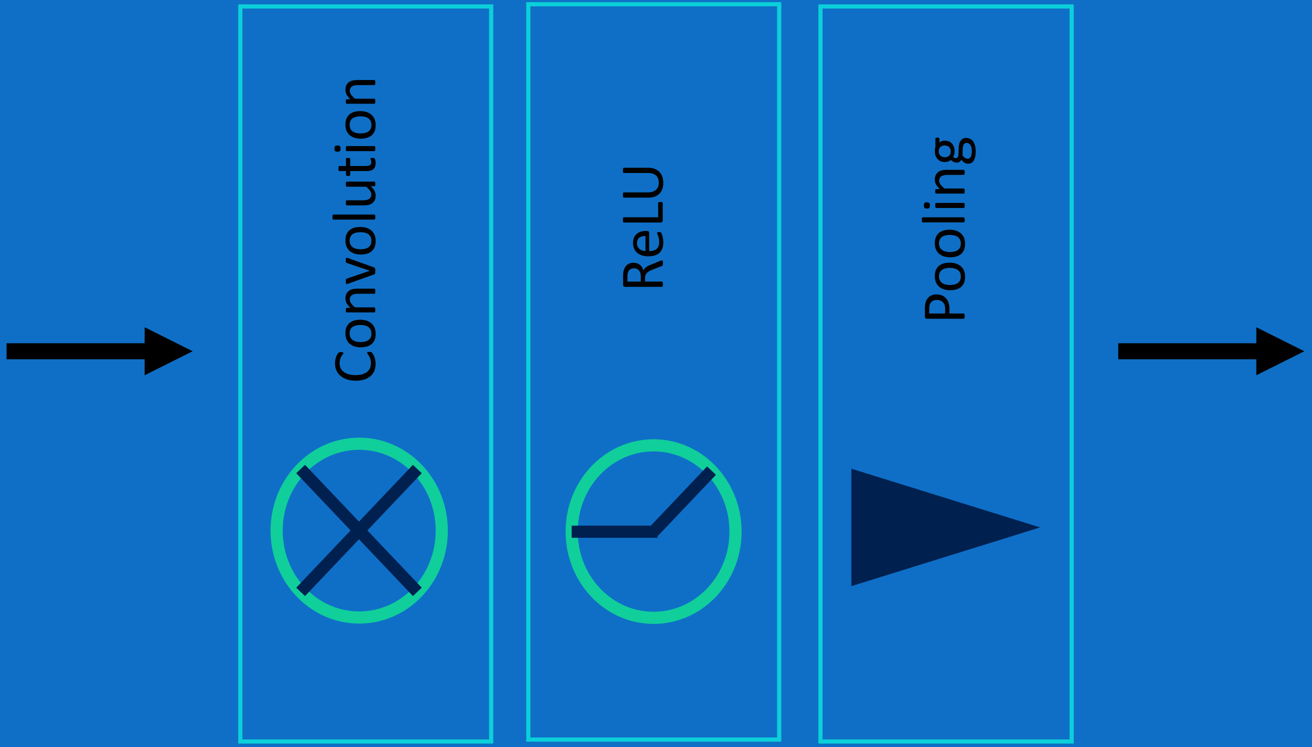
A stack of images becomes a stack of images with no negative values.



Layers get stacked

The output of one becomes the input of the next.

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1



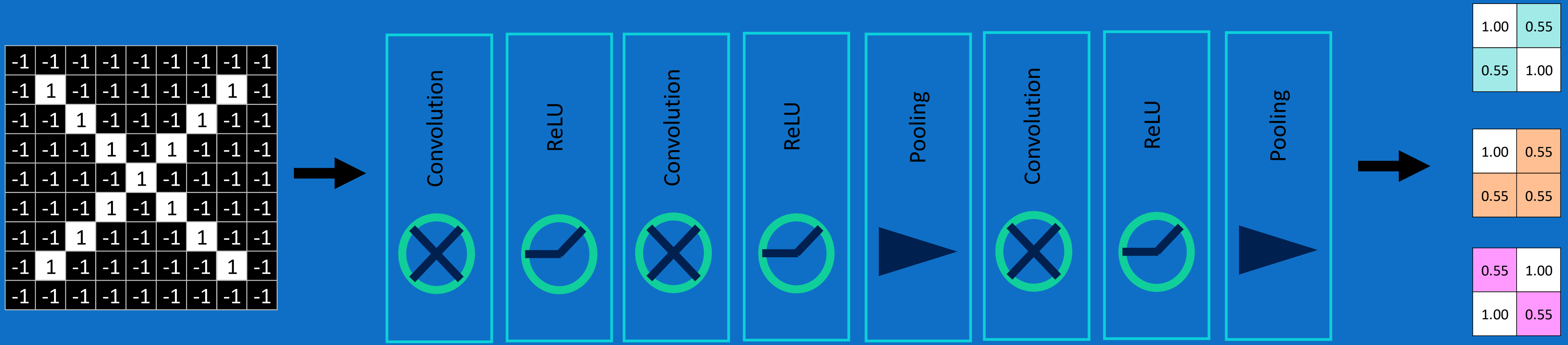
1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

0.55	0.33	0.55	0.33
0.33	1.00	0.55	0.11
0.55	0.55	0.55	0.11
0.33	0.11	0.11	0.33

0.33	0.55	1.00	0.77
0.55	0.55	1.00	0.33
1.00	1.00	0.11	0.55
0.77	0.33	0.55	0.33

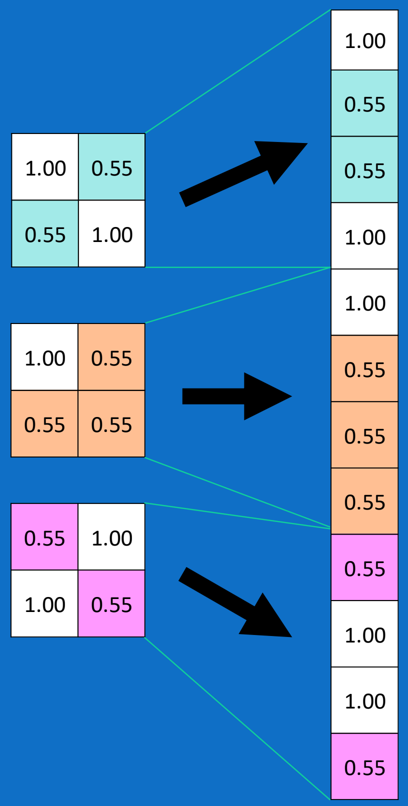
Deep stacking

Layers can be repeated several (or many) times.



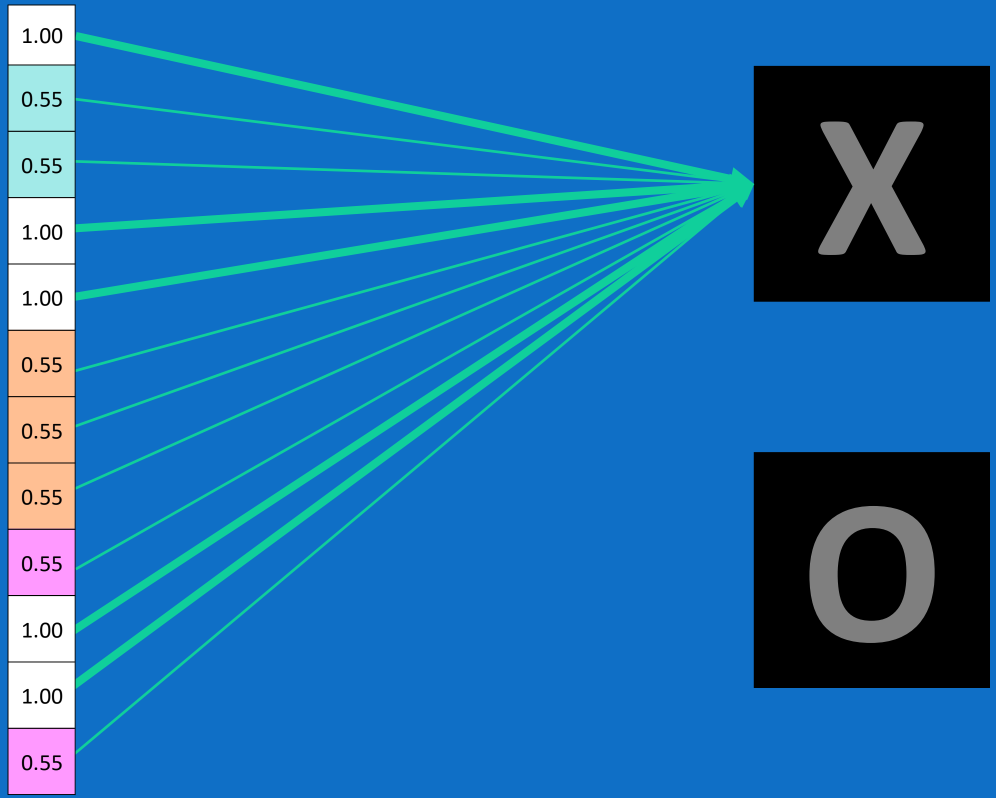
Fully connected layer

Every value gets a vote



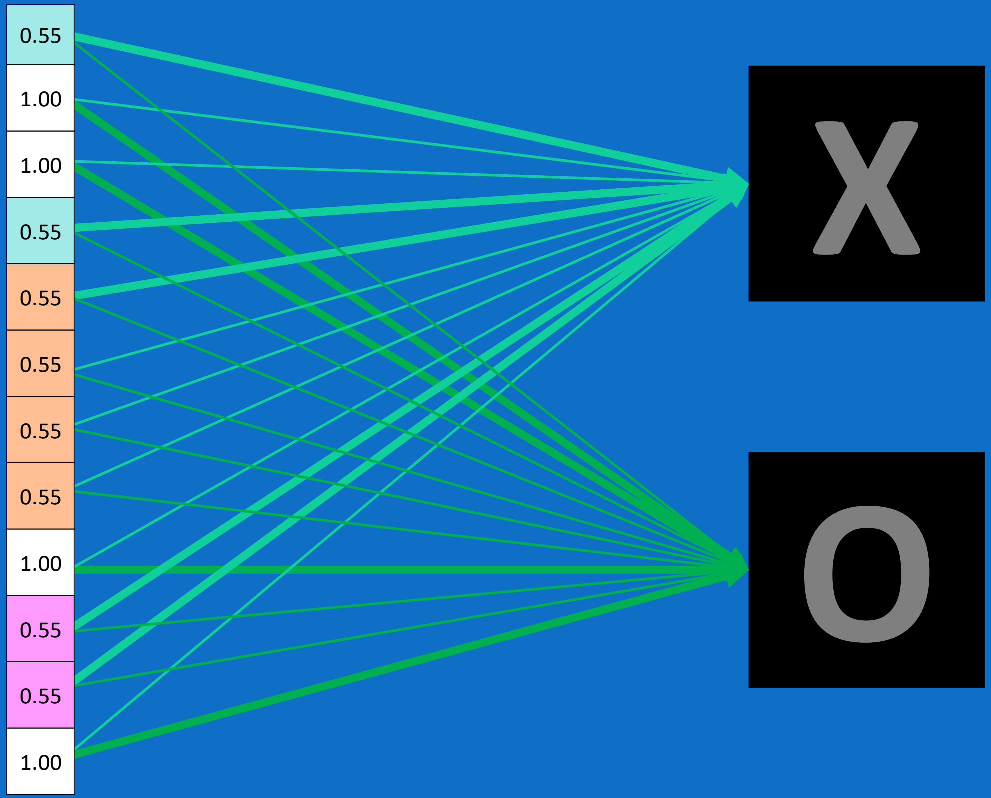
Fully connected layer

Vote depends on how strongly a value predicts X or O



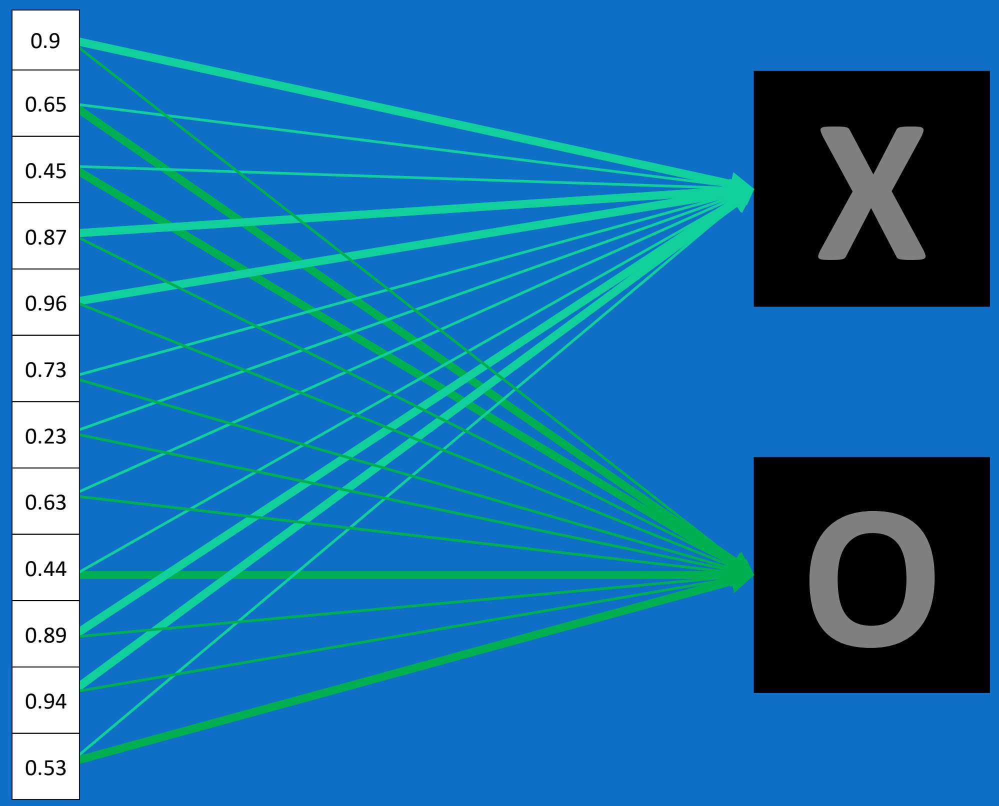
Fully connected layer

Vote depends on how strongly a value predicts X or O



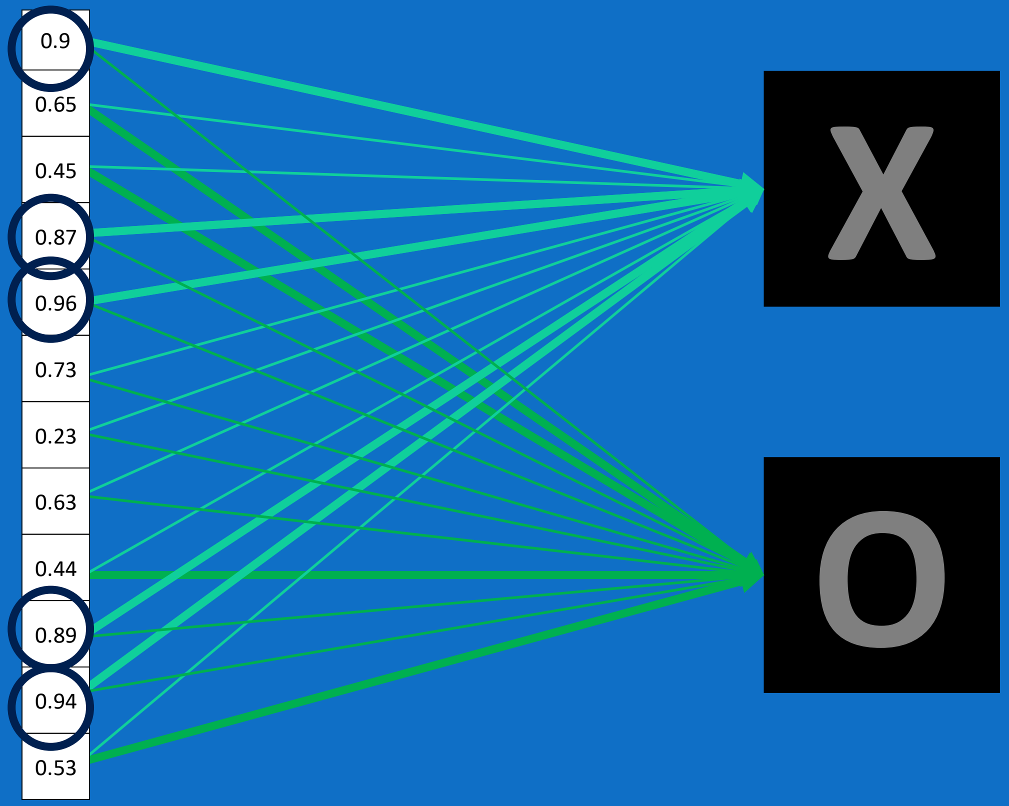
Fully connected layer

Future values vote on X or O



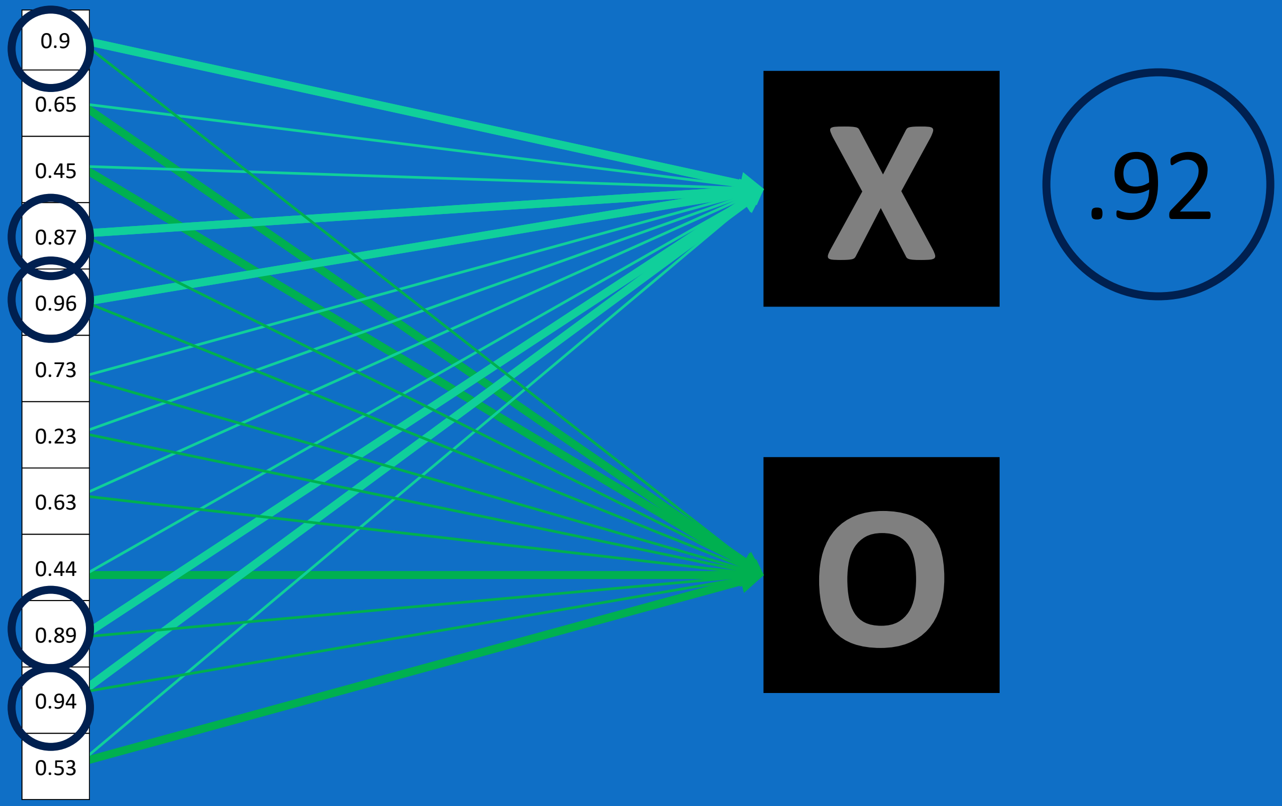
Fully connected layer

Future values vote on X or O



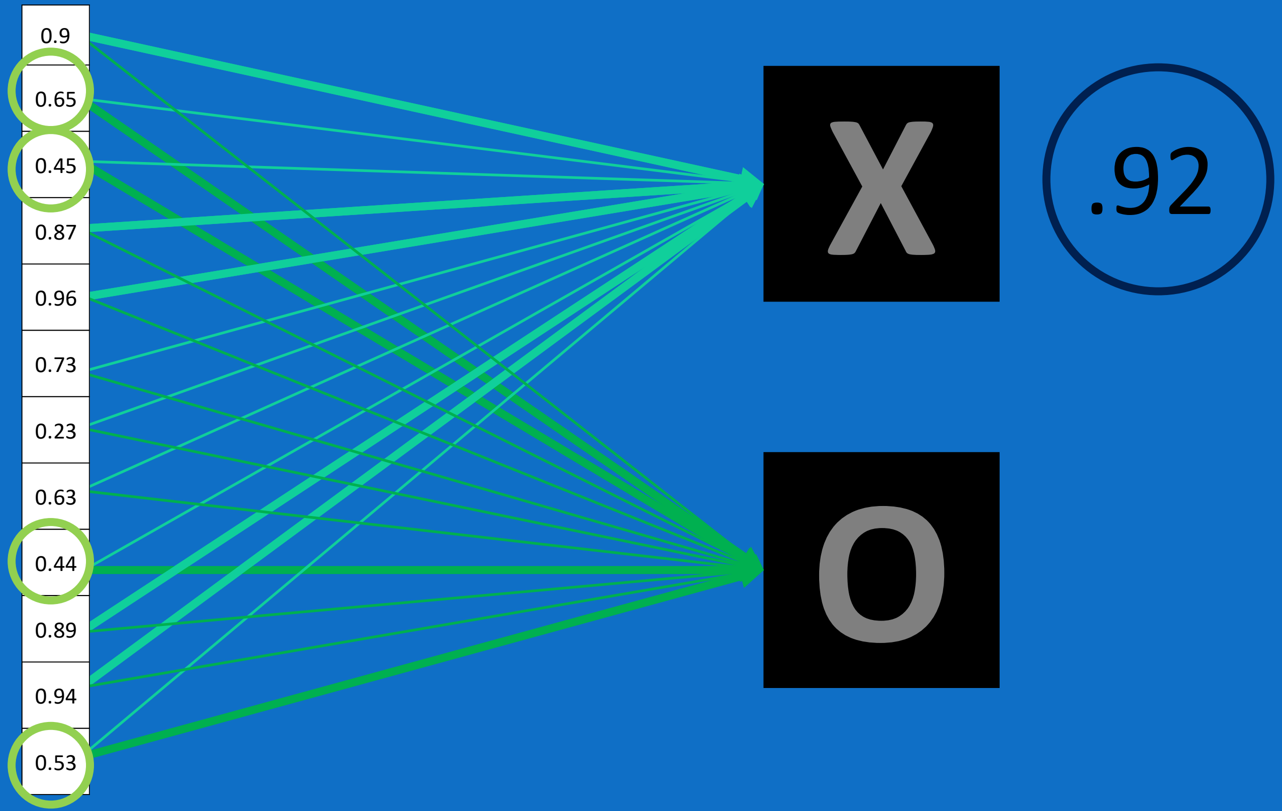
Fully connected layer

Future values vote on X or O



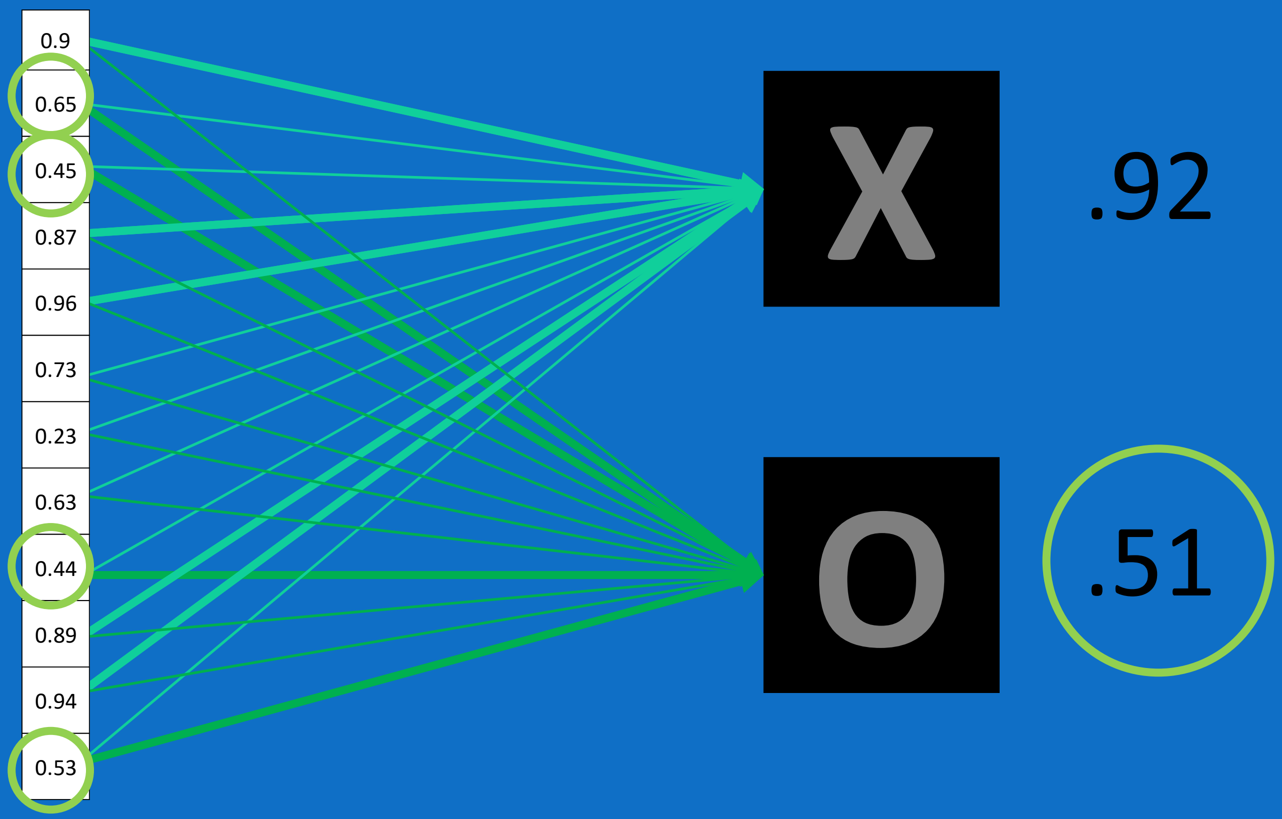
Fully connected layer

Future values vote on X or O



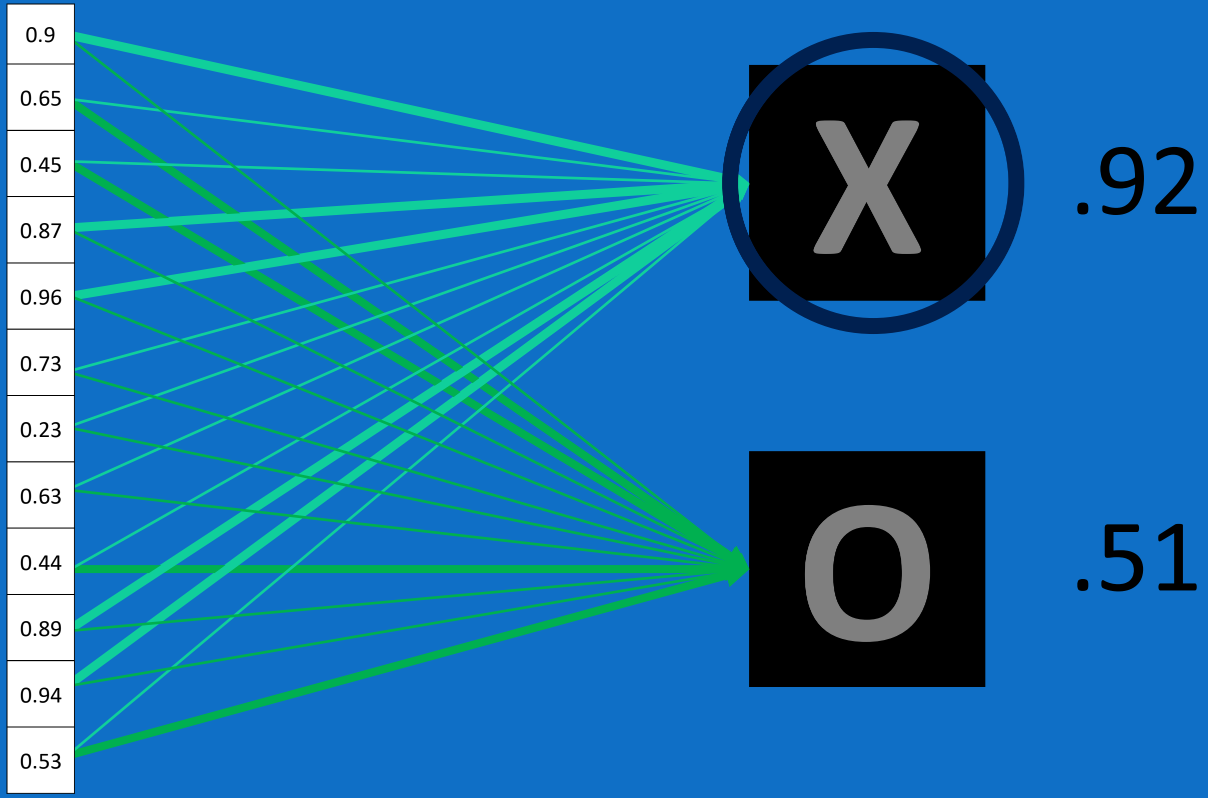
Fully connected layer

Future values vote on X or O



Fully connected layer

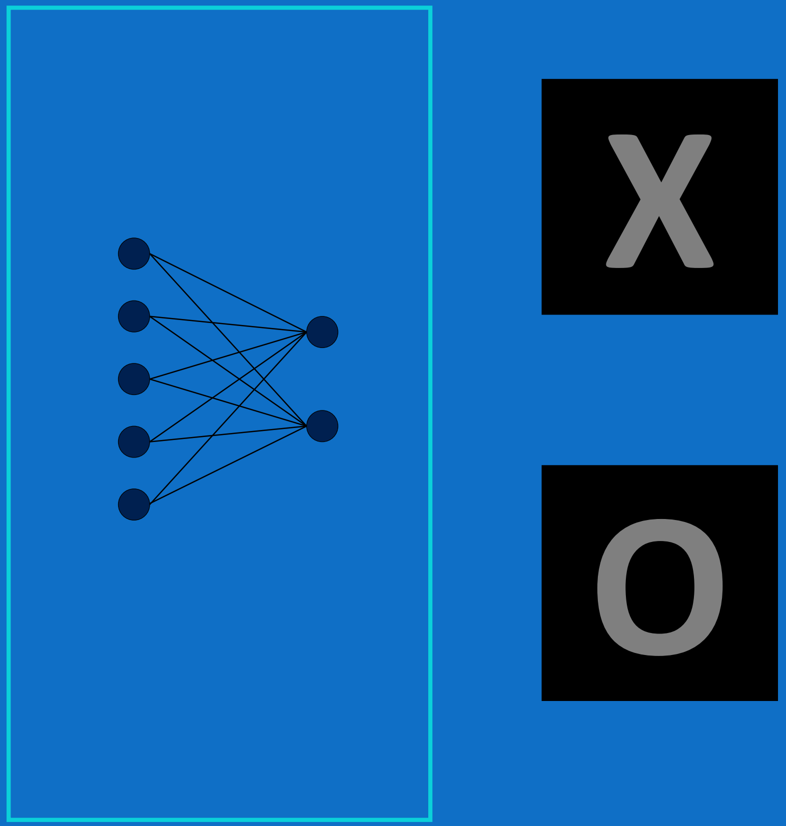
Future values vote on X or O



Fully connected layer

A list of feature values becomes a list of votes.

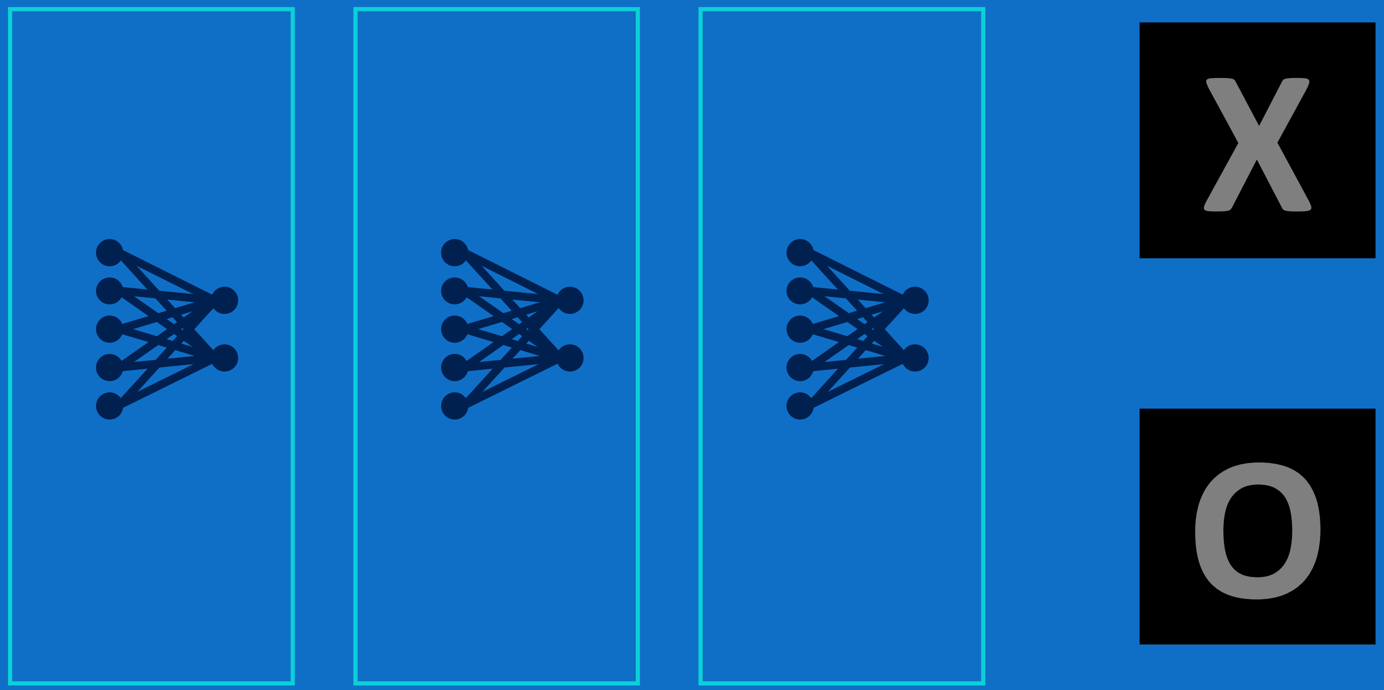
0.9
0.65
0.45
0.87
0.96
0.73
0.23
0.63
0.44
0.89
0.94
0.53



Fully connected layer

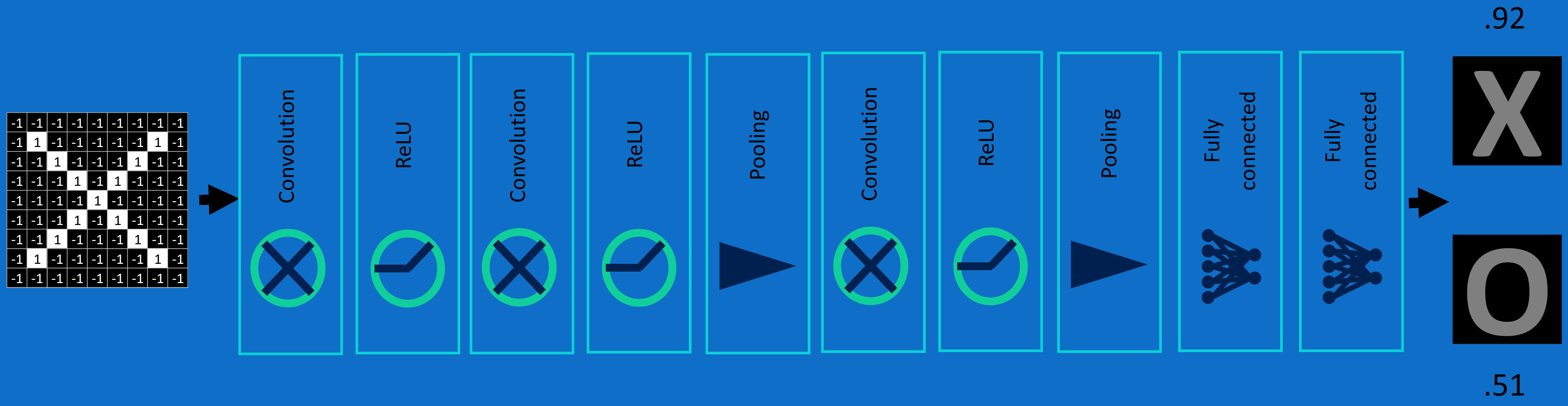
These can also be stacked.

0.9
0.65
0.45
0.87
0.96
0.73
0.23
0.63
0.44
0.89
0.94
0.53



Putting it all together

A set of pixels becomes a set of votes.



Learning

Q: Where do all the magic numbers come from?

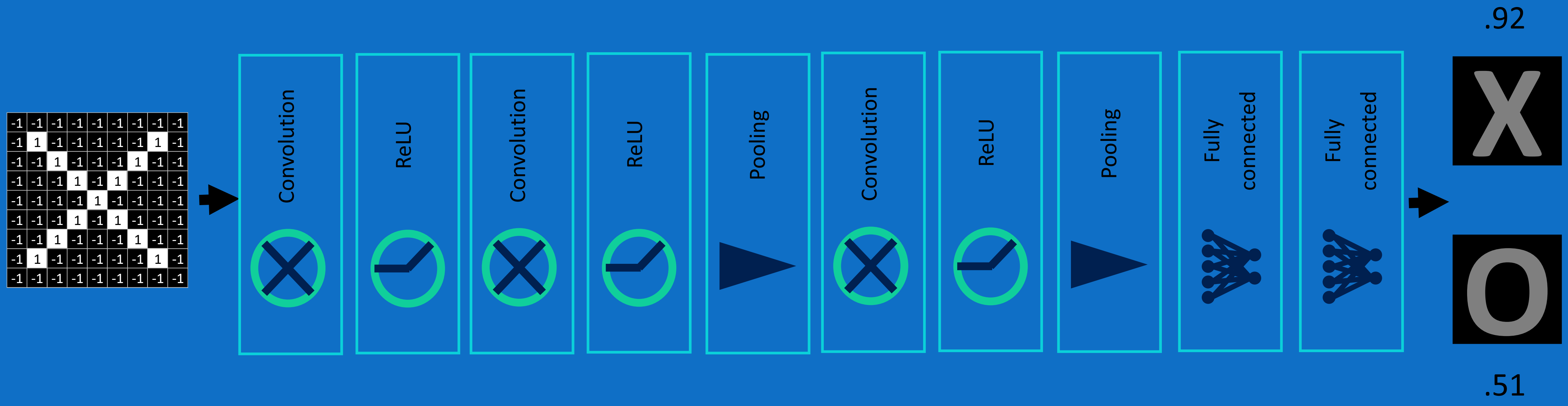
Features in convolutional layers

Voting weights in fully connected layers

A: Backpropagation

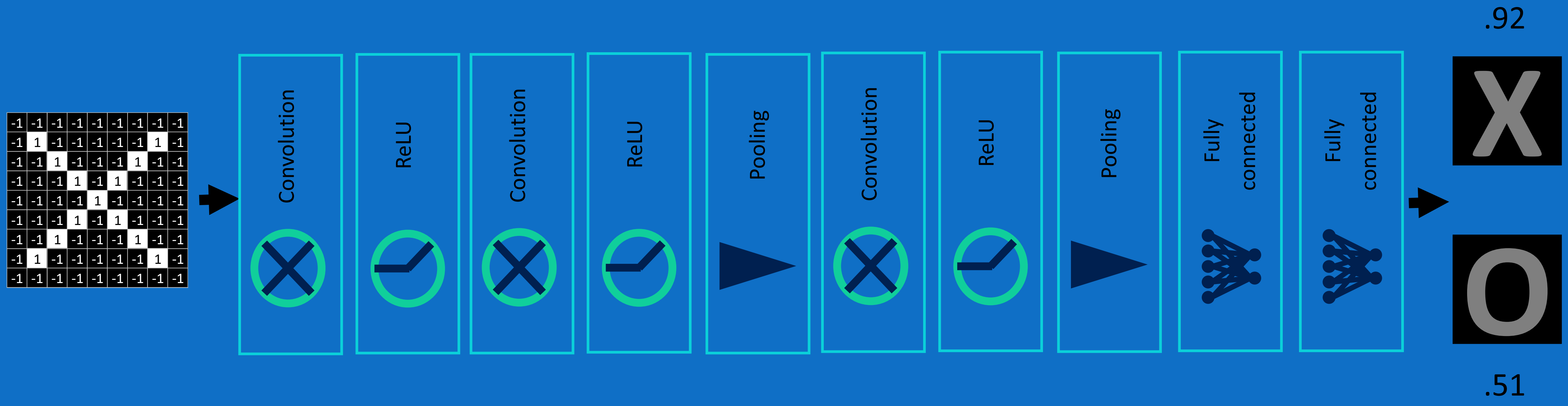
Backprop

Error = right answer – actual answer



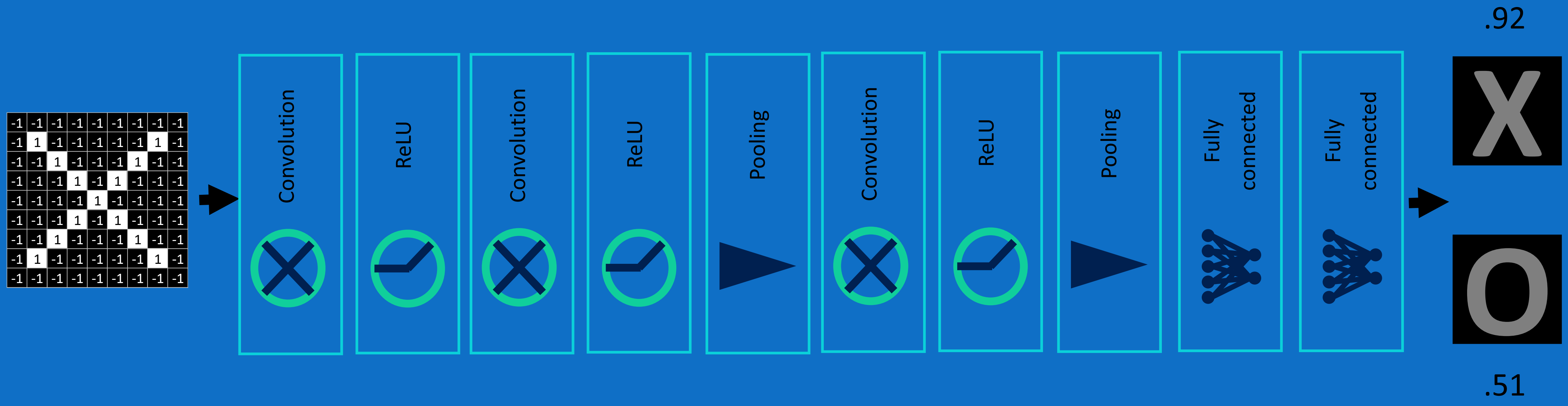
Backprop

	Right answer	Actual answer	Error
X	1		
O			



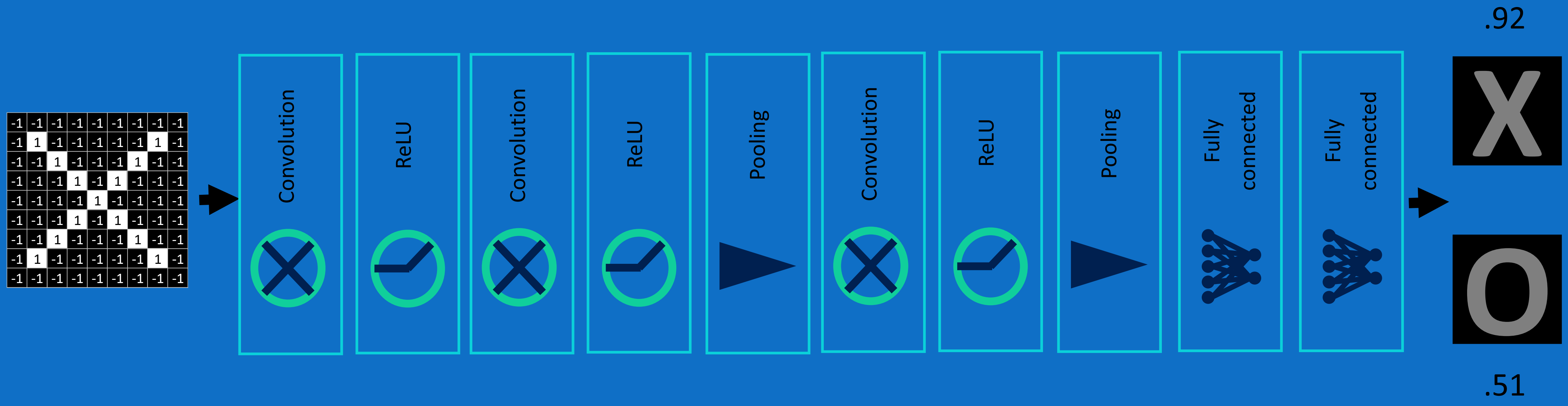
Backprop

	Right answer	Actual answer	Error
X	1	0.92	
O			



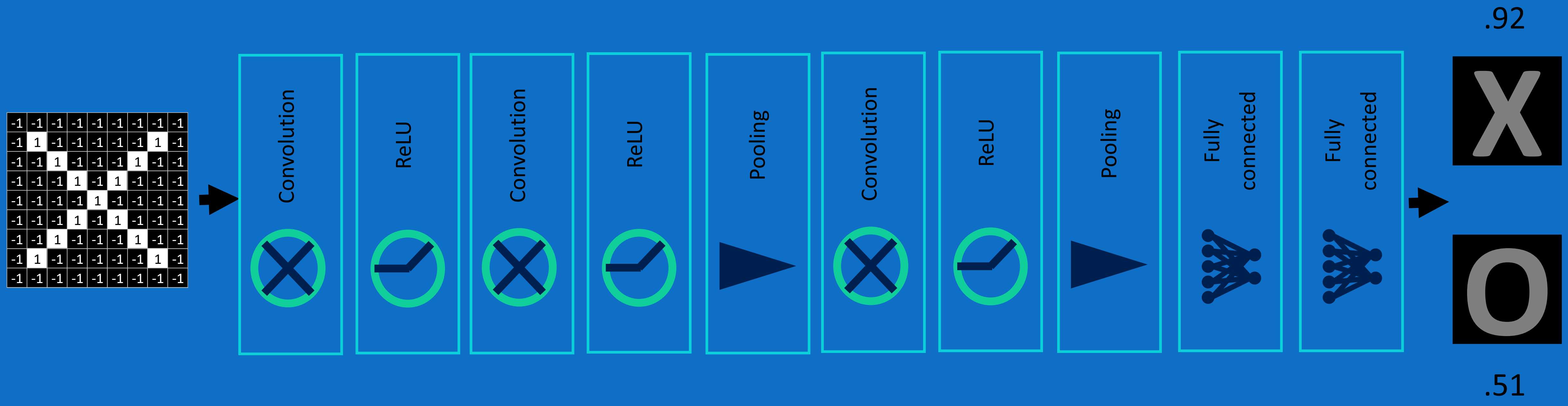
Backprop

	Right answer	Actual answer	Error
X	1	0.92	0.08
O			



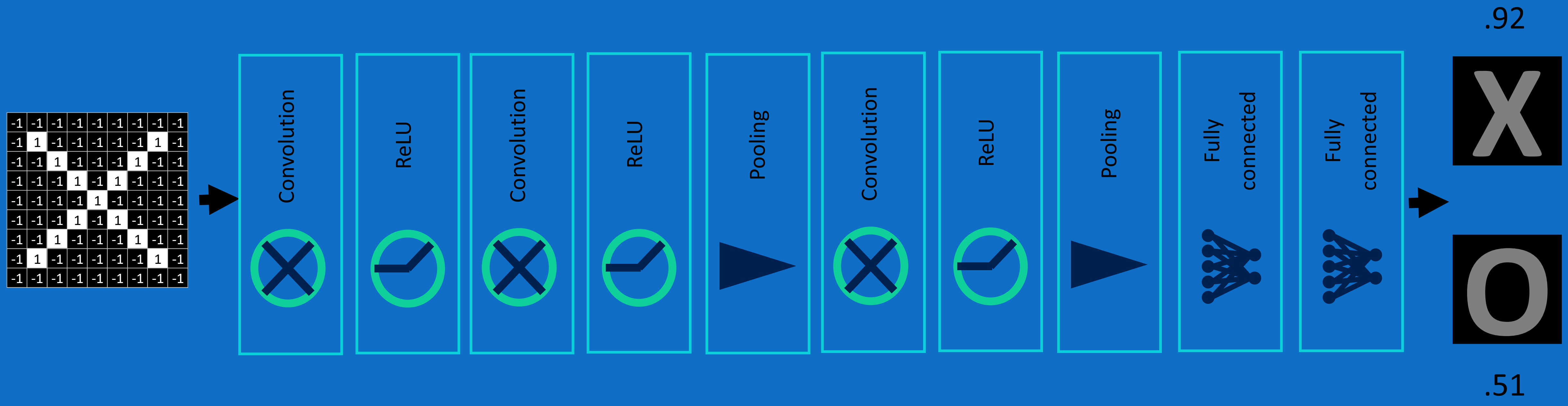
Backprop

	Right answer	Actual answer	Error
X	1	0.92	0.08
O	0	0.51	0.49



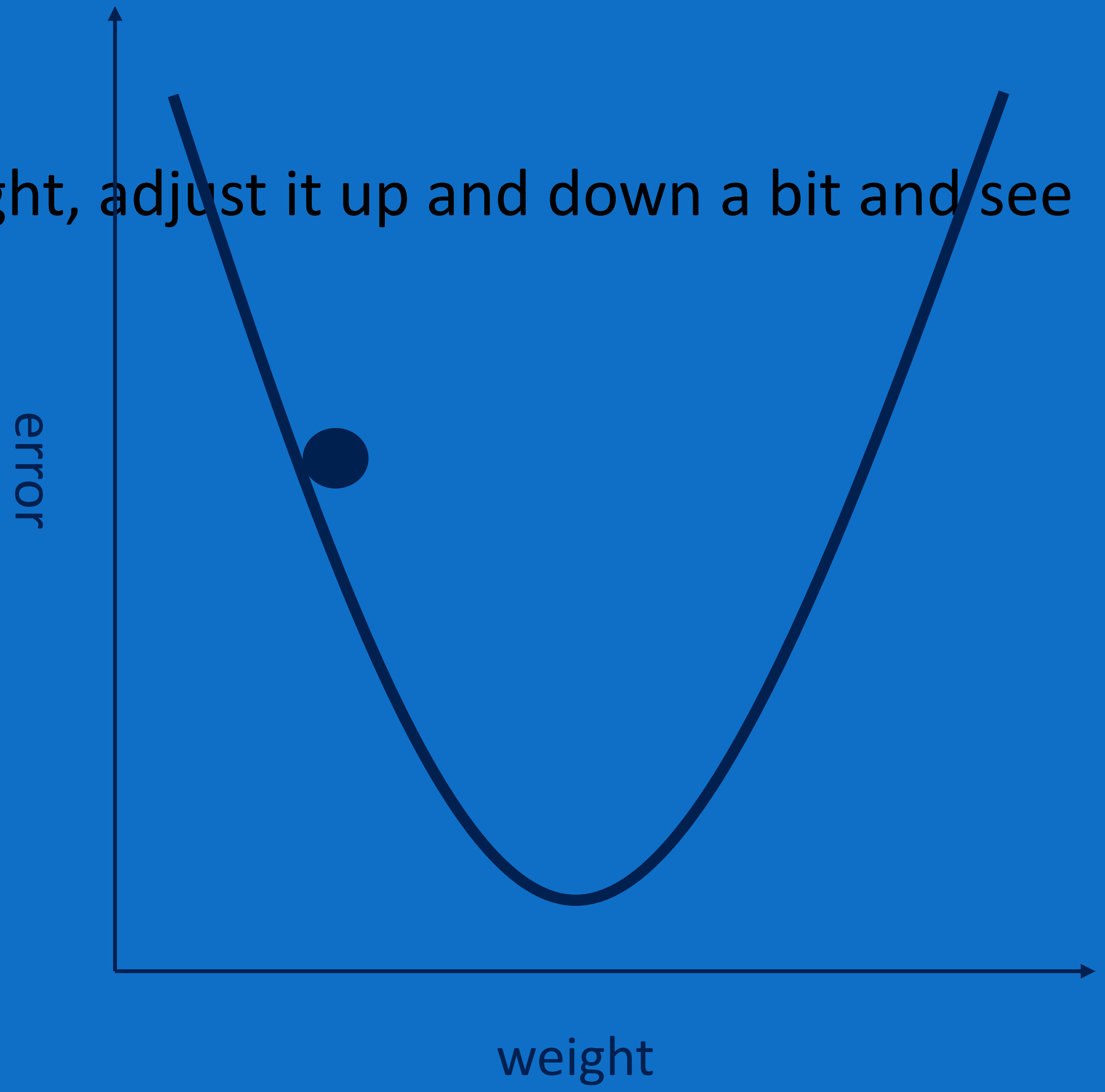
Backprop

	Right answer	Actual answer	Error
X	1	0.92	0.08
O	0	0.51	0.49
		Total	0.57



Gradient descent

For each feature pixel and voting weight, adjust it up and down a bit and see how the error changes.



Gradient descent

For each feature pixel and voting weight, adjust it up and down a bit and see how the error changes.

Hyperparameters (knobs)

- Convolution
  - Number of features
  - Size of features
- Pooling
  - Window size
  - Window stride
- Fully Connected
  - Number of neurons

Architecture

How many of each type of layer?

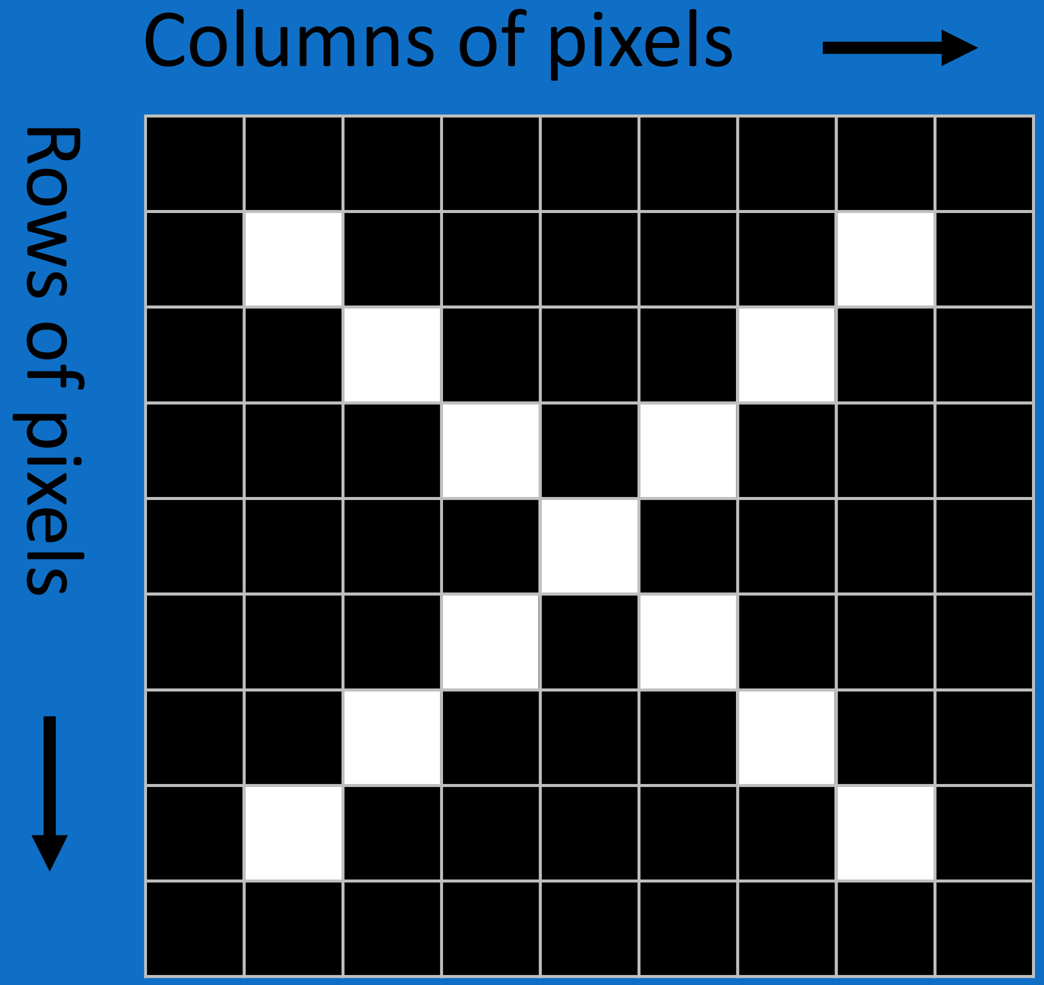
In what order?

Not just images

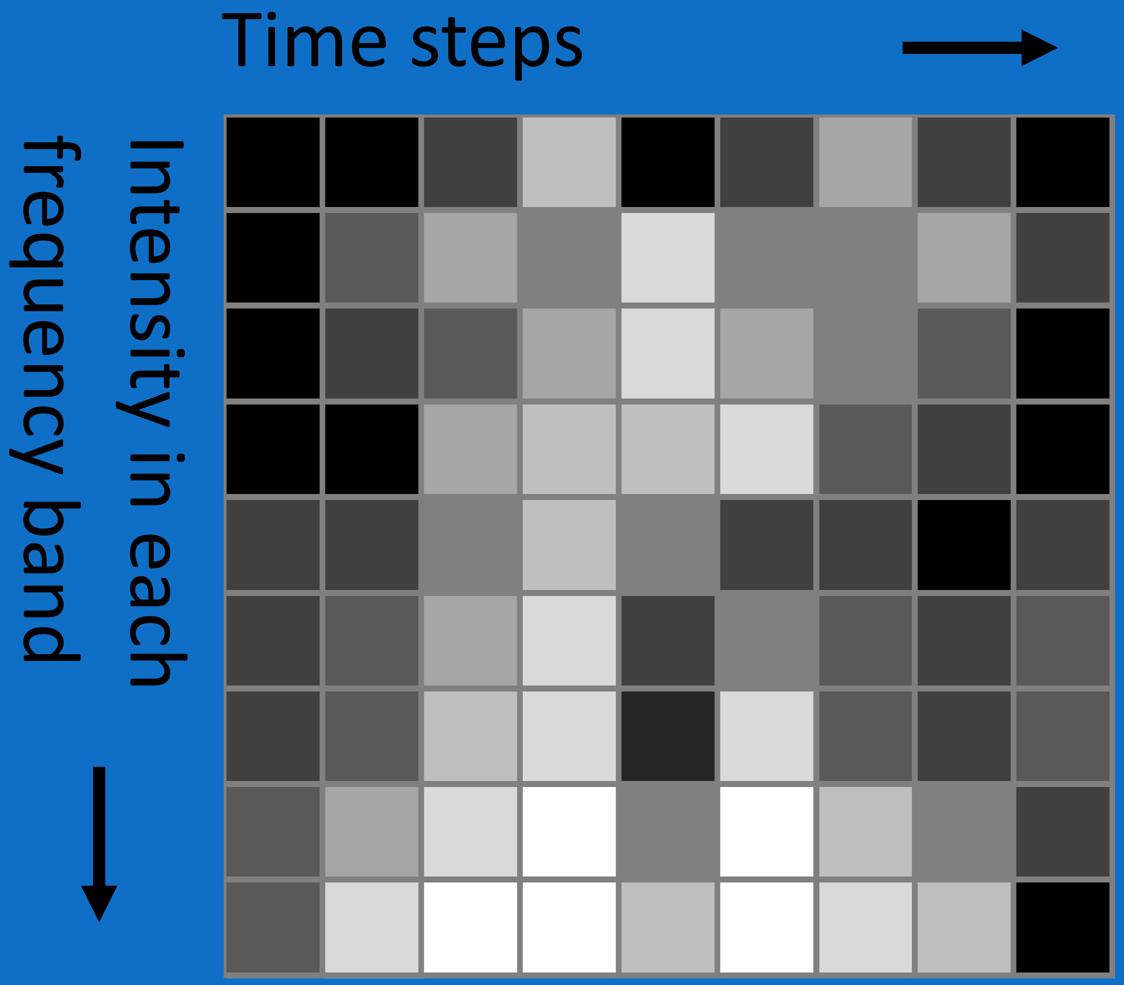
Any 2D (or 3D) data.

Things closer together are more closely related than things far away.

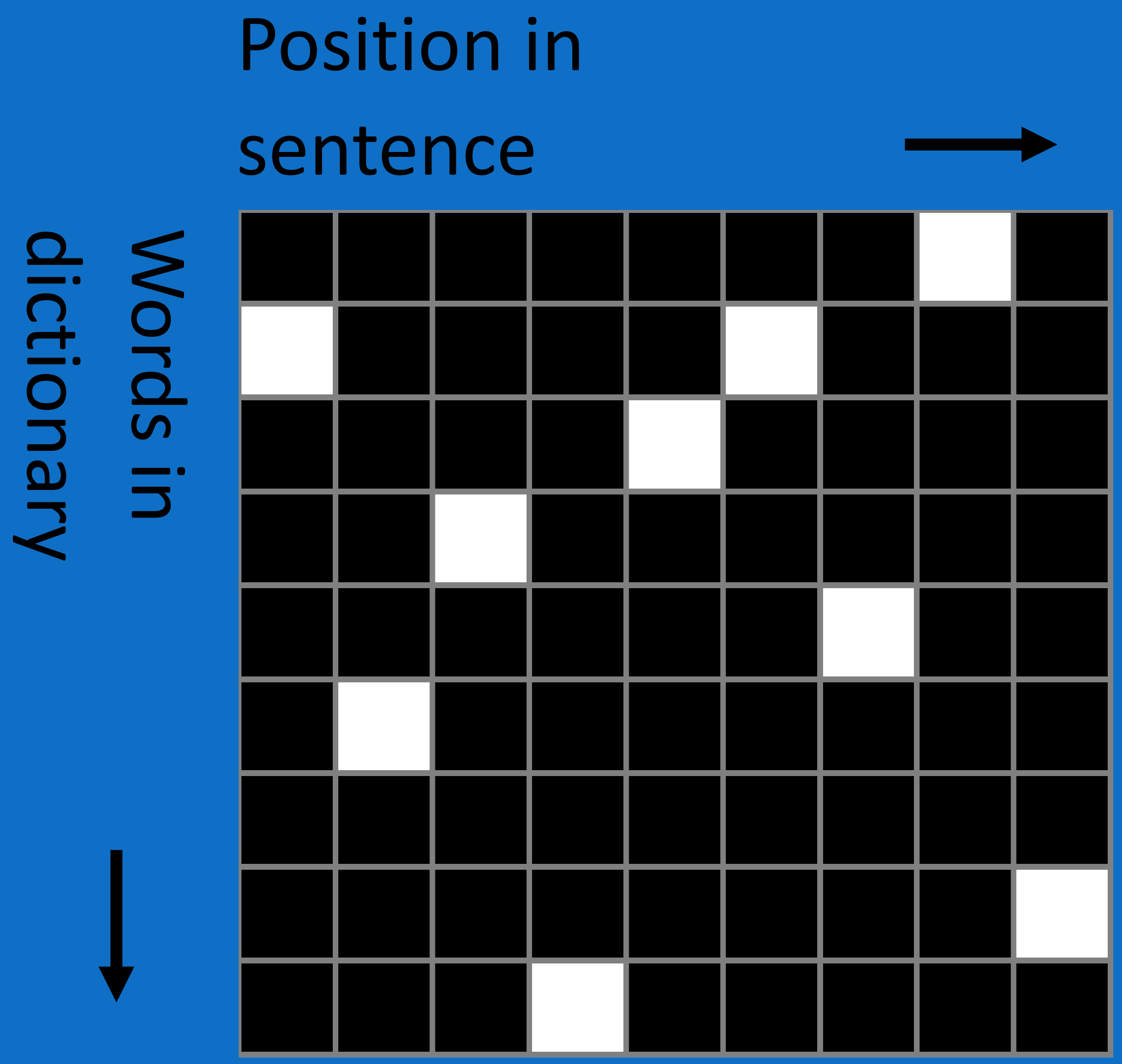
Images

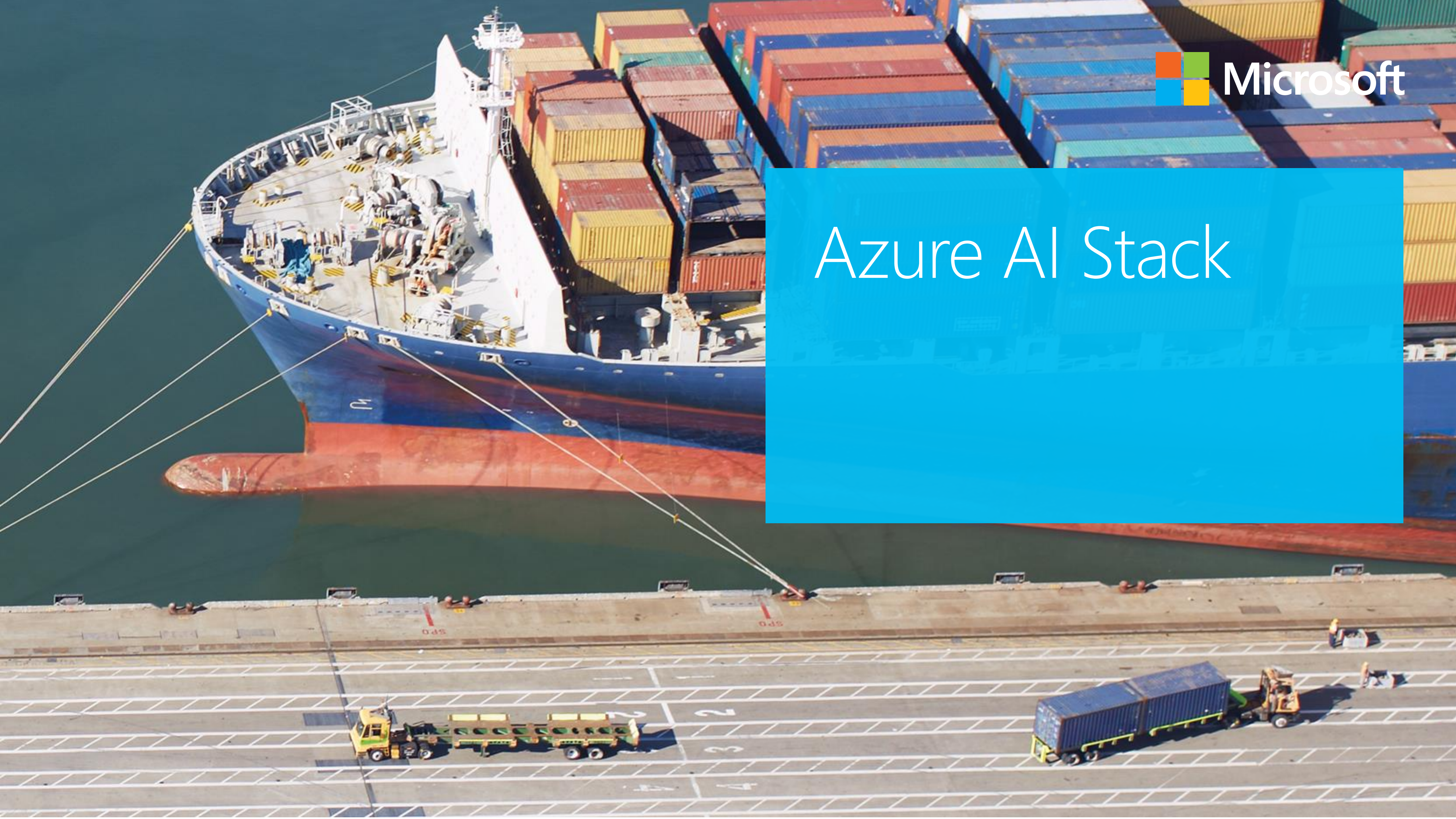


Sound



Text

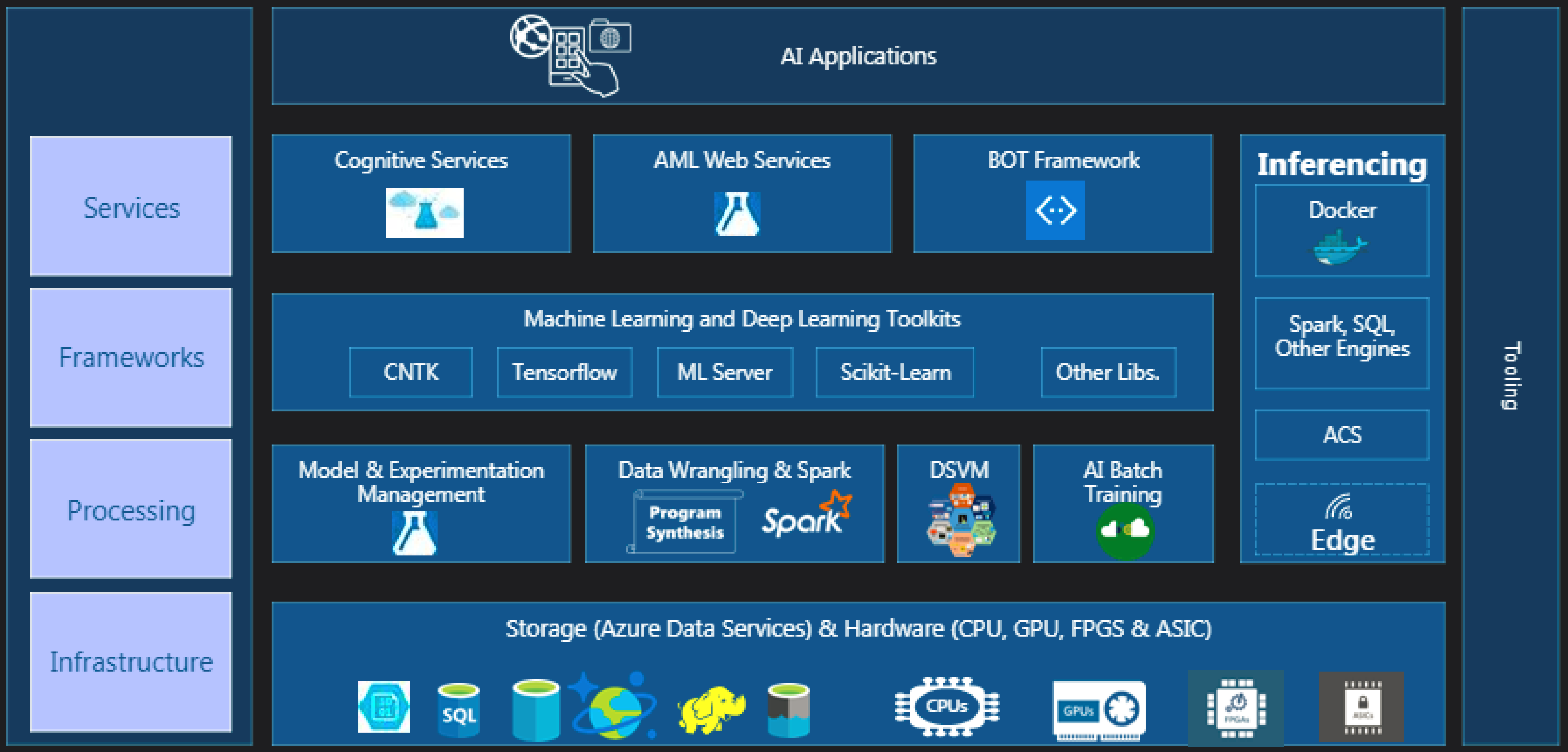




Microsoft

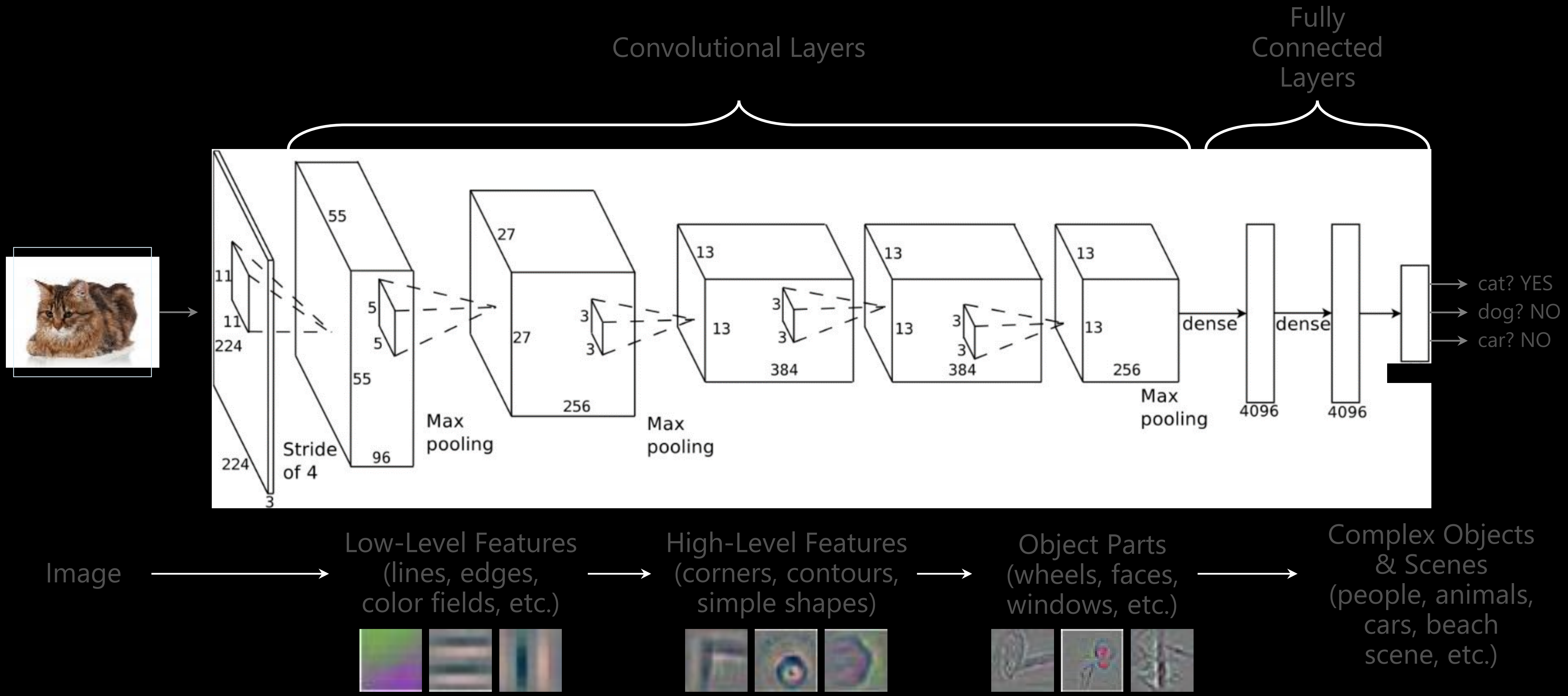
Azure AI Stack

Cloud AI Stack



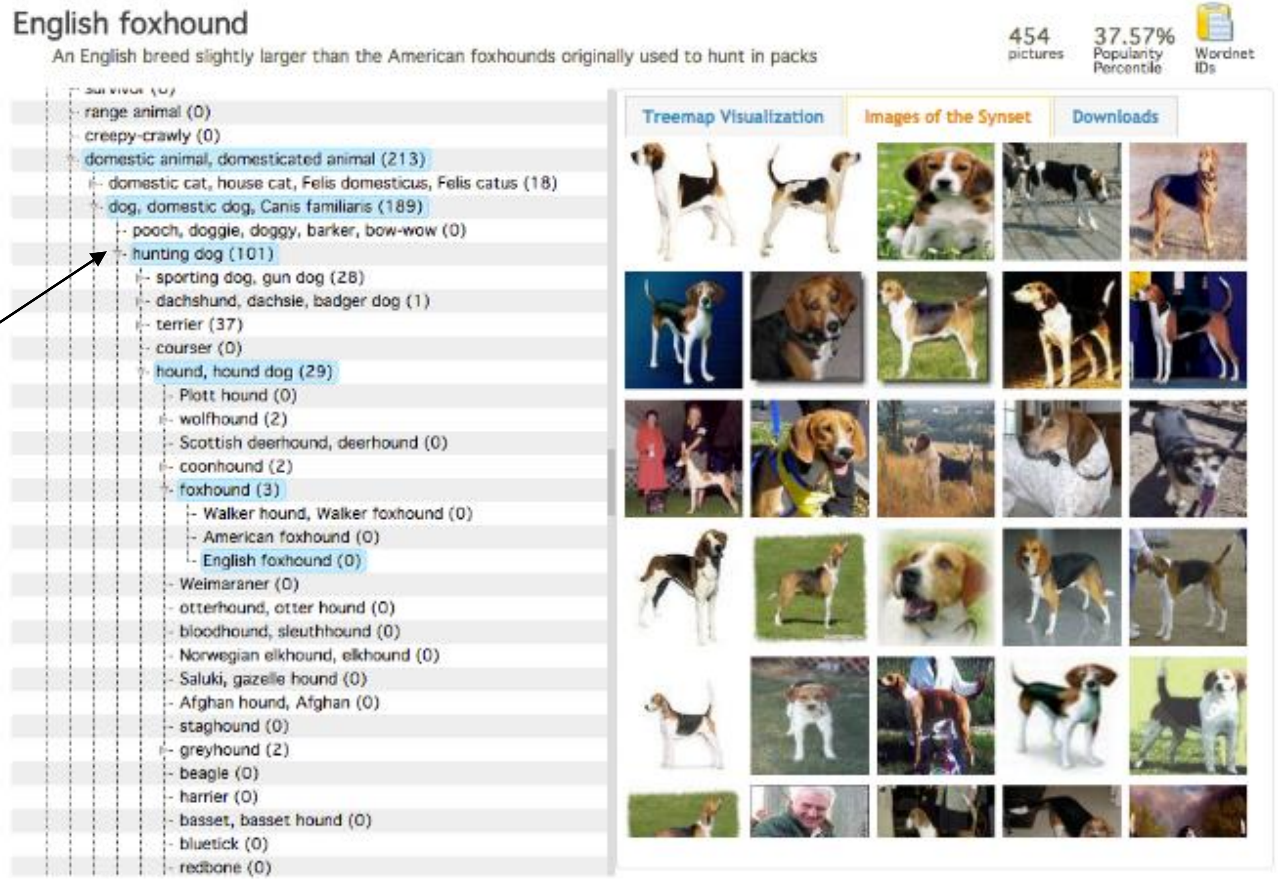
Deep Learning for Image Classification

Deep Neural Network for Computer Vision



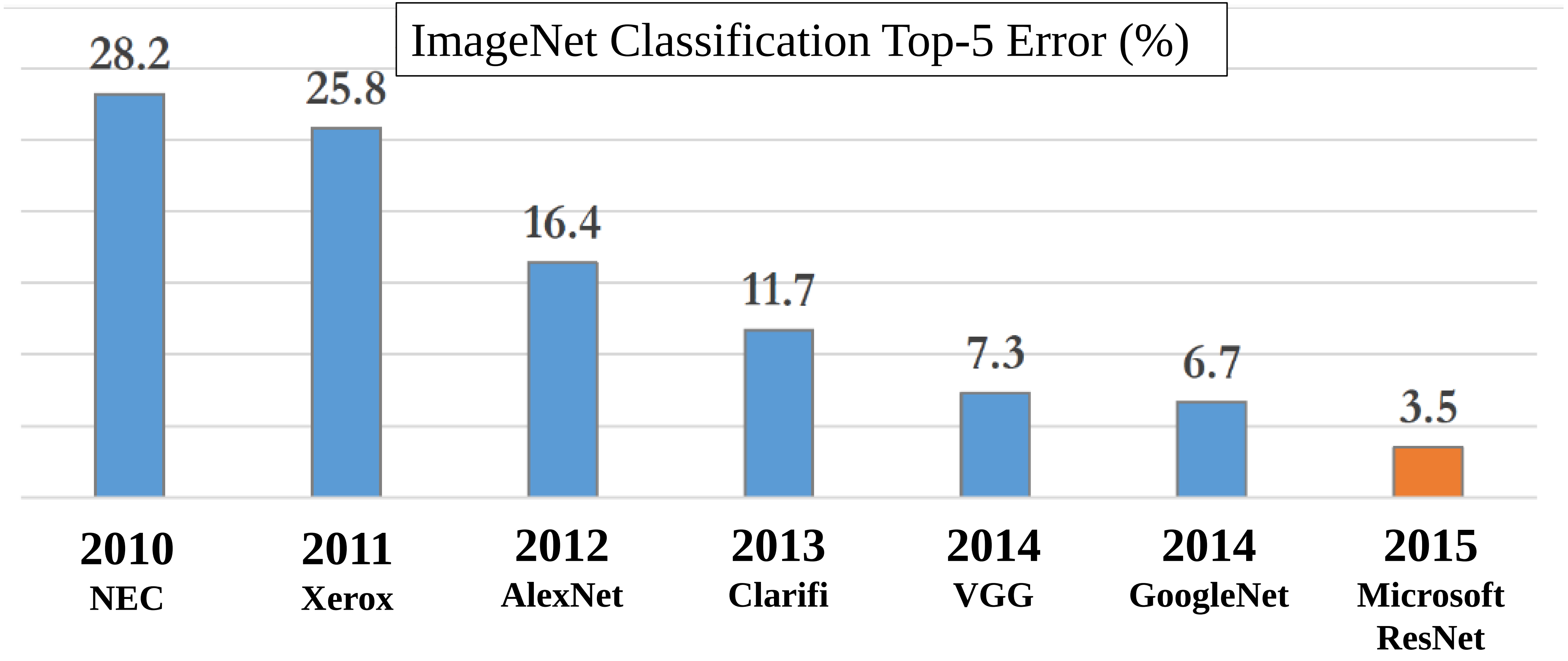
ImageNet dataset

- Research dataset with >10 million images
- Image annotated with labels from on ontology (>22K labels)
- Generic images covering an extremely wide ranges of labels



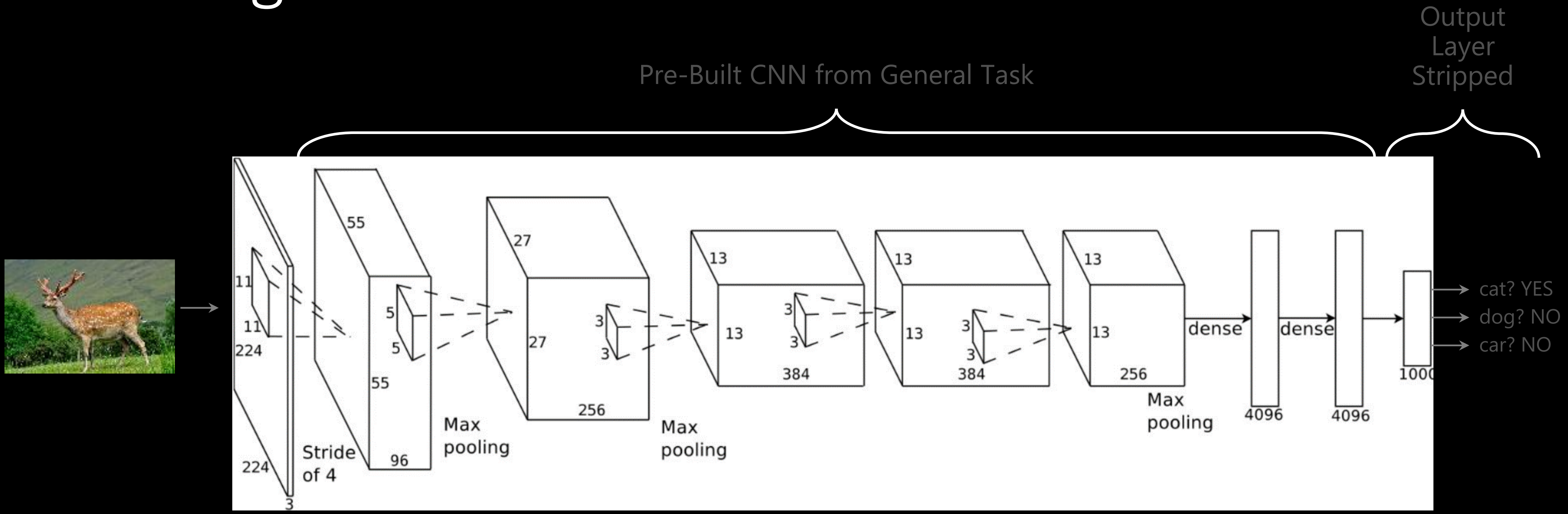
ImageNet taxonomy with example images for “English foxhound”

Image Net ILSVRC Competition



Customizing Deep Learning Model

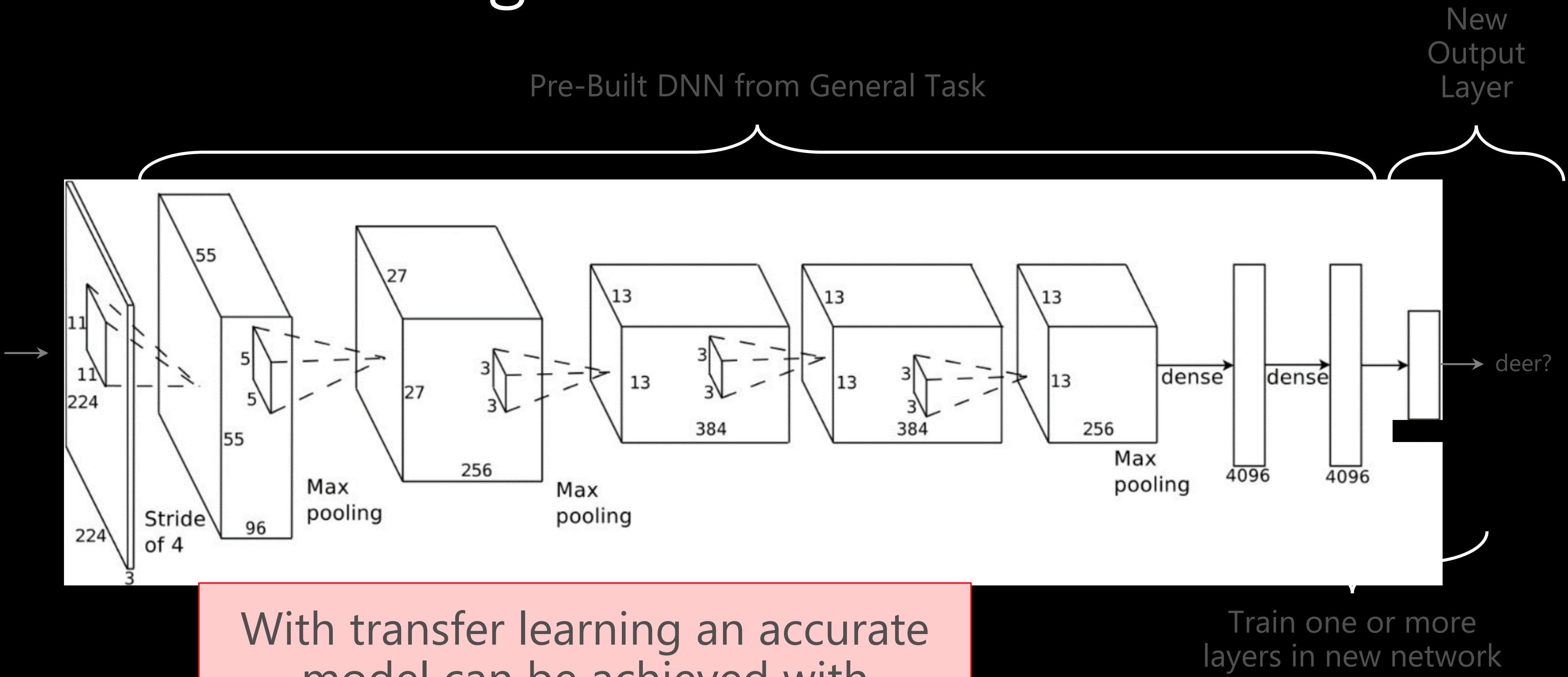
Using a Pre-Trained CNN as a Featurizer



Outputs of penultimate layer of ImageNet Trained CNN provide excellent general purpose image features

Generic Image Features

Transfer Learning

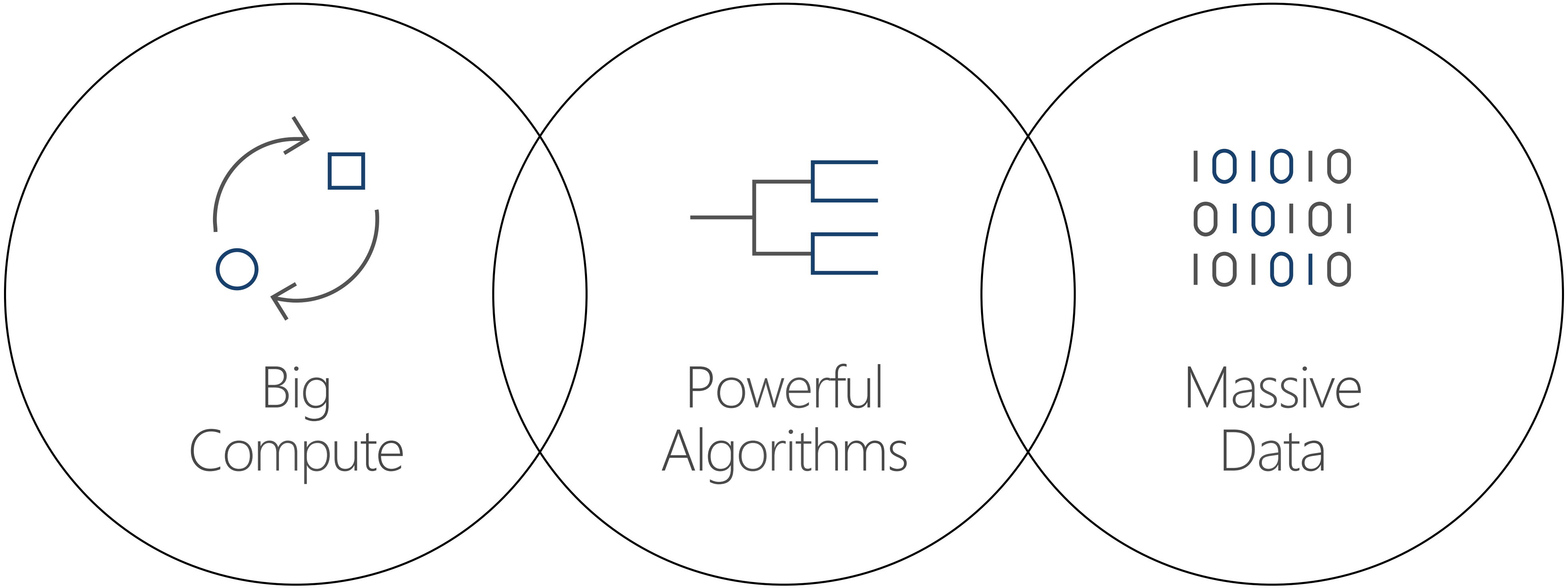


With transfer learning an accurate model can be achieved with thousands (or less) of labeled examples instead of millions

Microsoft AI Platform

Batch AI Training
Azure Machine Learning
Data Science Virtual Machine

Recipe for AI Innovation



Batch AI Training

Easy Deployment and Flexibility

Managed service that enables data scientists to easily train and test deep learning and other AI models

High Performance Training

Work with clusters of GPUs to run experiments in parallel and at scale to reduce training time

Any Framework and Tools

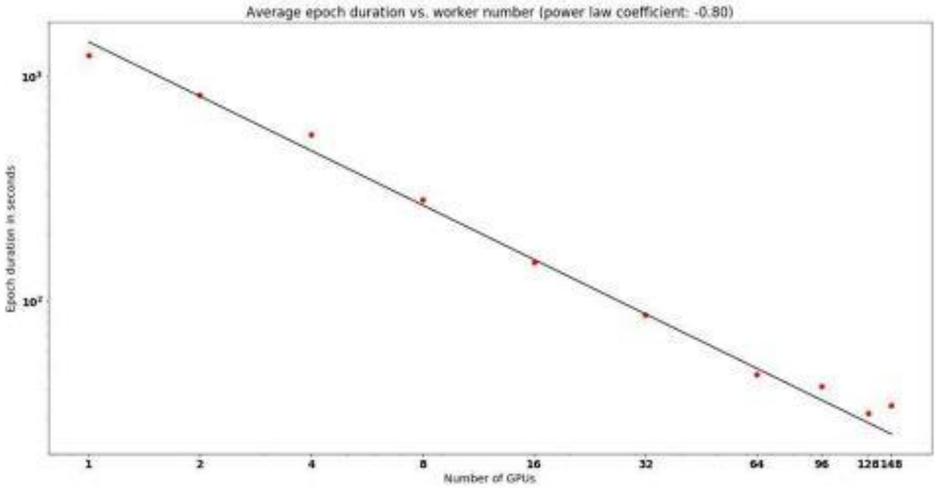
Do your AI with the scale and tools of Microsoft

Near linear scale up on 1.6 TB Aerial Images

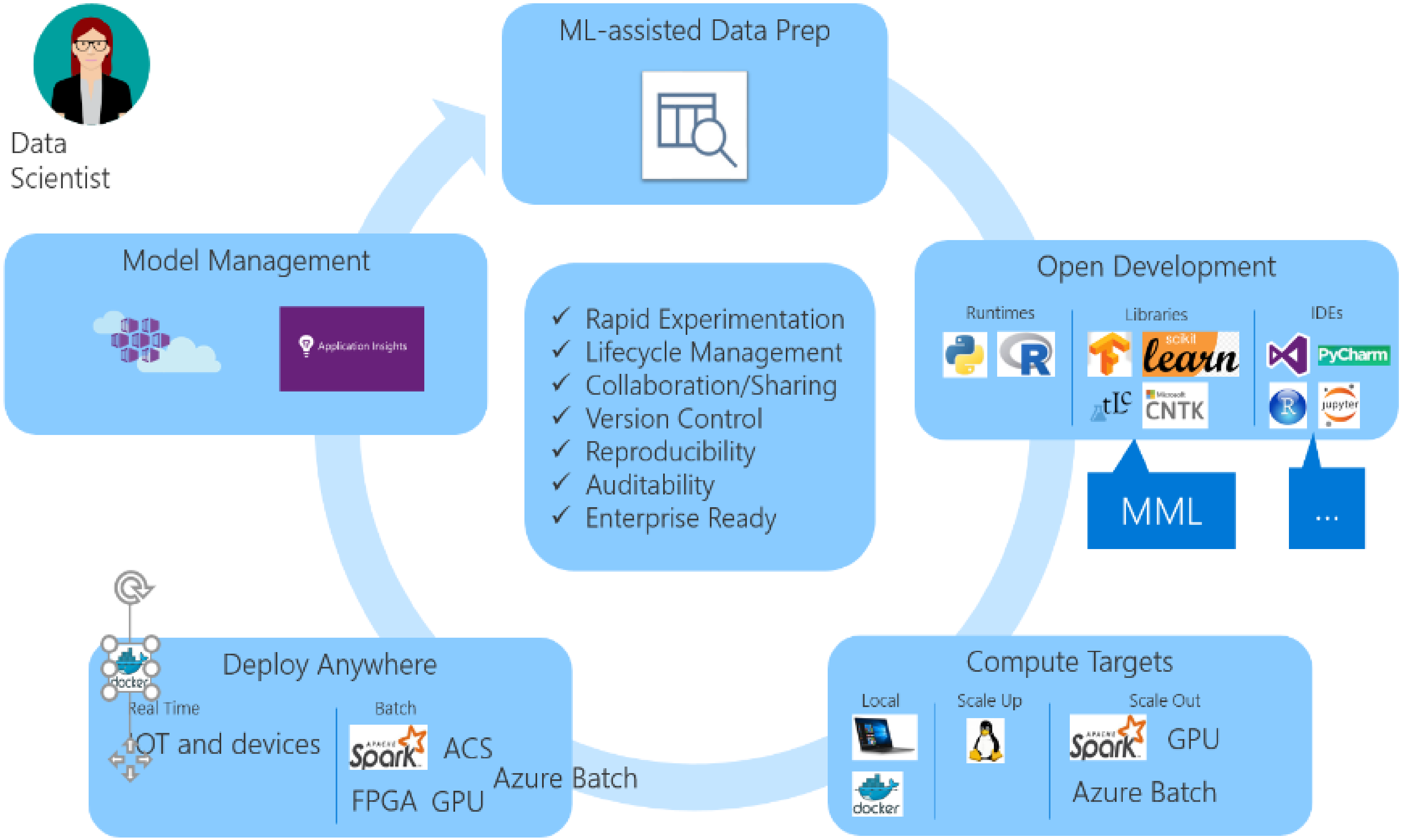
Training time decreases from

1925 seconds/epoch to

32 seconds/epoch.

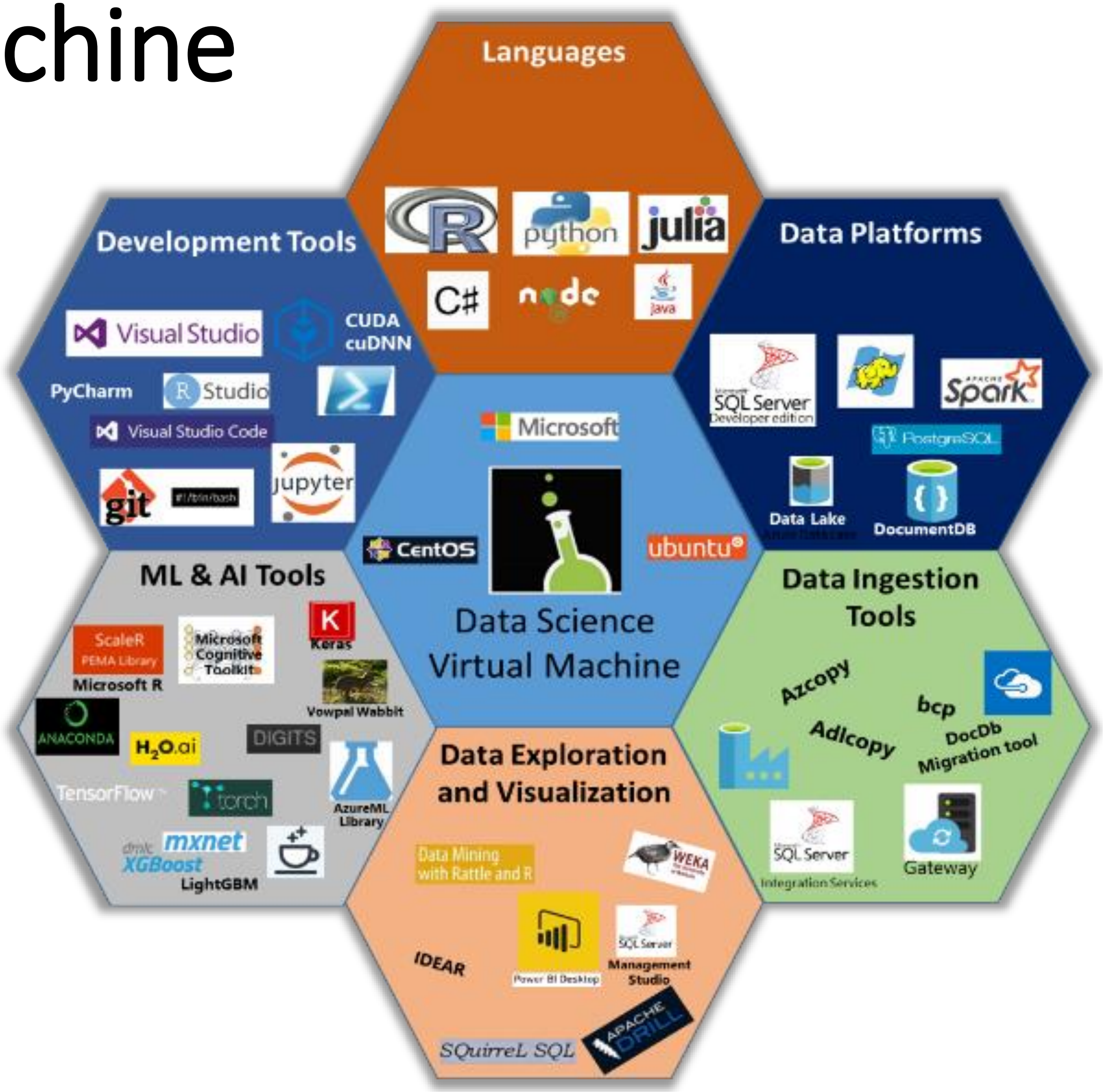


Azure Machine Learning



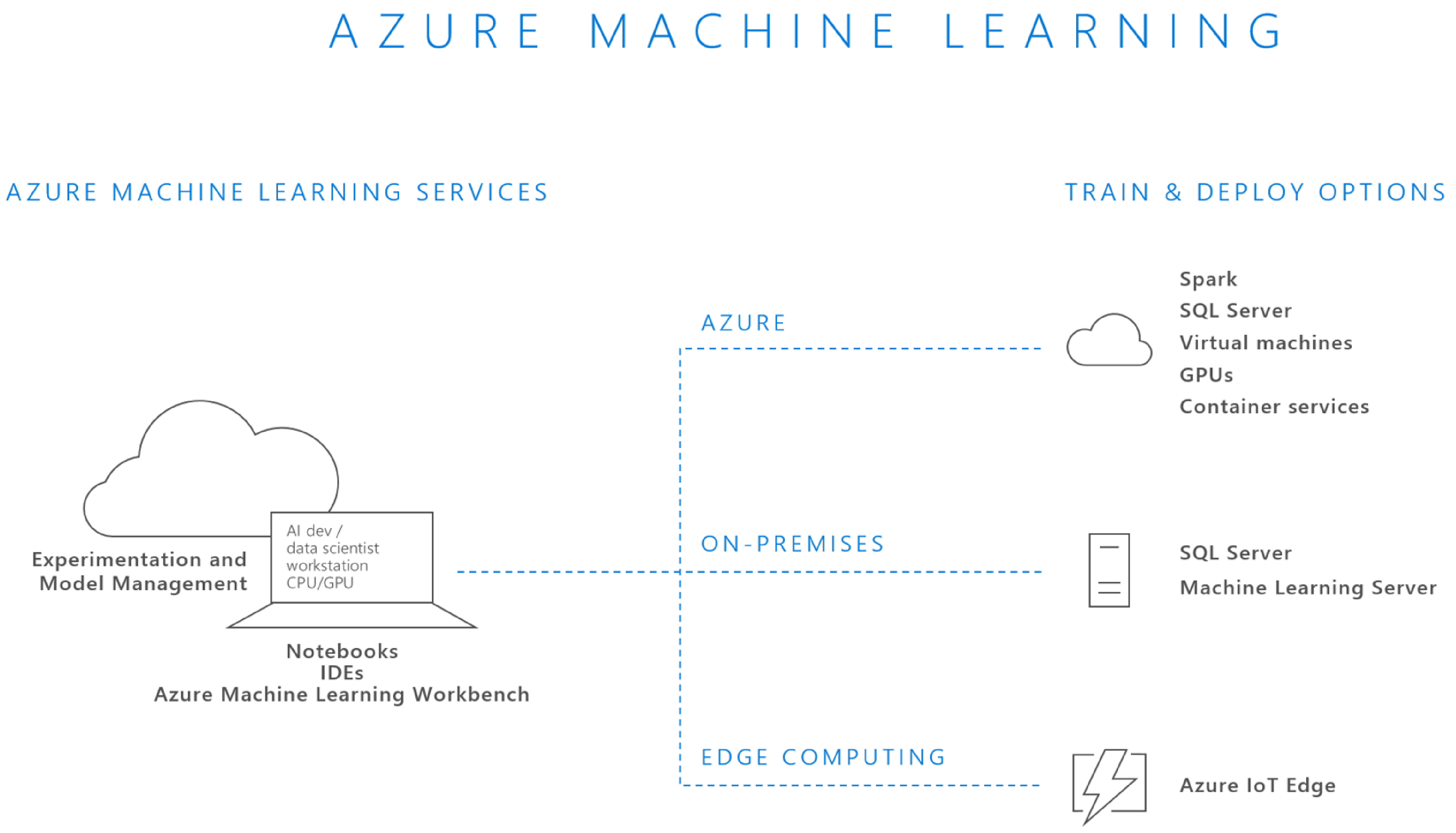
Data Science Virtual Machine

Running our Image Processing Pipeline in a Data Science Virtual Machine (DSVM)
with Deep Learning products pre-stalled



Components of Azure Machine Learning

- Azure Machine Learning Workbench
- Azure Machine Learning Experimentation Service
- Azure Machine Learning Model Management Service
- Microsoft Machine Learning Libraries for Apache Spark (MMLSpark Library)
- Visual Studio Code Tools for AI



AZURE MACHINE LEARNING
AZURE MACHINE LEARNING SERVICES
TRAIN & DEPLOY OPTIONS
Experimentation and Model Management
AI dev / data scientist workstation CPU/GPU
Notebooks
IDEs
Azure Machine Learning Workbench
AZURE
Spark
SQL Server
Virtual machines
GPUs
Container services
ON-PREMISES
SQL Server
Machine Learning Server
EDGE COMPUTING
Azure IoT Edge

Azure Machine Learning Workbench

- Azure Machine Learning Workbench is a desktop application plus command-line tools
- It allows you to manage machine learning solutions through the entire data science life cycle:
- Data ingestion and preparation
- Model development and experiment management
- Model deployment in various target environments

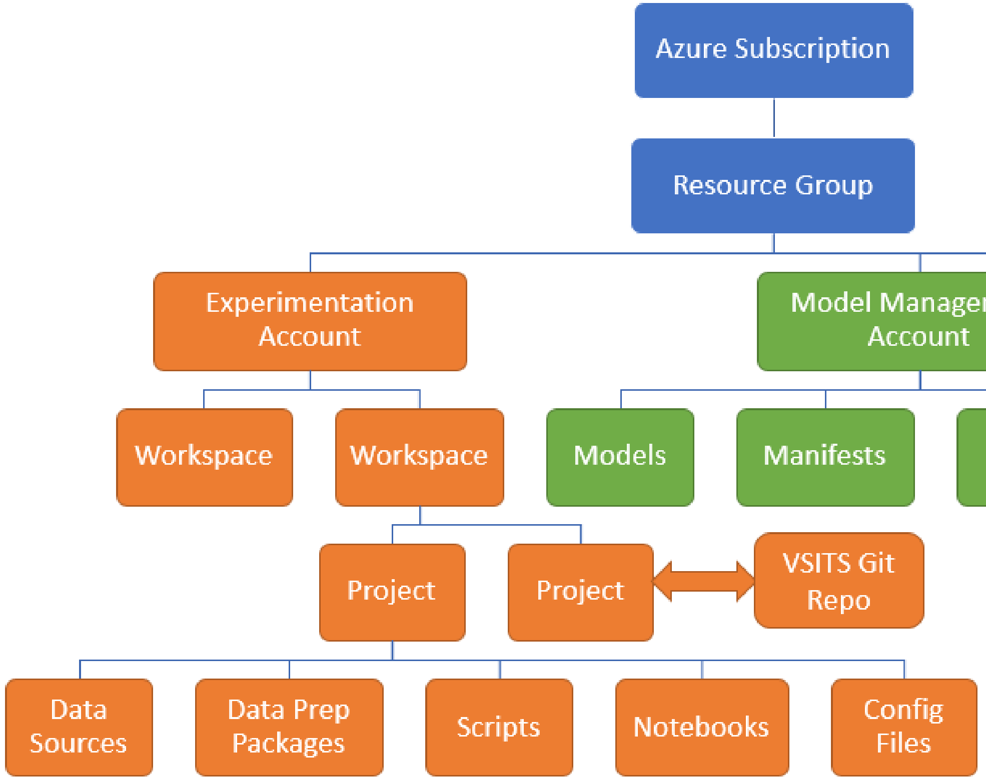
Azure Machine Learning Experimentation Service

- The Experimentation Service handles the execution of machine learning experiments.
- It also supports the Workbench by providing project management, Git integration, access control, roaming, and sharing.
- Through easy configuration, you can execute your experiments across a range of compute environment options:
  - Local native
  - Local Docker container
  - Docker container on a remote VM
  - Scale out Spark cluster in Azure

Azure Machine Learning Model Management Service

- Model Management Service allows data scientists and dev-ops teams to deploy predictive models into a wide variety of environments.
- Model versions and lineage are tracked from training runs to deployments. Models are stored, registered, and managed in the cloud
- Using simple CLI commands, you can containerize your model, scoring scripts and dependencies into Docker images. These images are registered in your own Docker registry hosted in Azure (Azure Container Registry). They can be reliably deployed to the following targets:
  - Local machines
  - On-premises servers
  - The cloud
  - IoT edge devices

Azure Machine Learning Services



Offering from Microsoft on AI

- Besides Azure Machine Learning, there are a wide variety of options in Azure to build, deploy, and manage machine learning models.
    - Microsoft Machine Learning Services in SQL Server
    - Microsoft Machine Learning Server
    - Data Science Virtual Machine
    - Spark MLLib in HDInsight
    - Batch AI Training Service
    - Microsoft Cognitive Toolkit
    - Microsoft Cognitive Services

Image Classification Demo

- Training, evaluation and deployment to local or the cloud.
- All steps running within AML Workbench.
- Jupyter notebooks provided to visualize results, deploy as Rest API, and to perform load-testing.

Datasets:

Clothing texture images

Azure Machine Learning Service Demo

1. Prepare data
2. Build Models
3. Deploy Model
4. Advanced Analytics

Demo on Image Classification using CNTK

Image Classification Demo

Clothing texture dataset:

- Collected 1716 images from Bing, and annotated as one of 11 textures (striped, dotted, etc.)

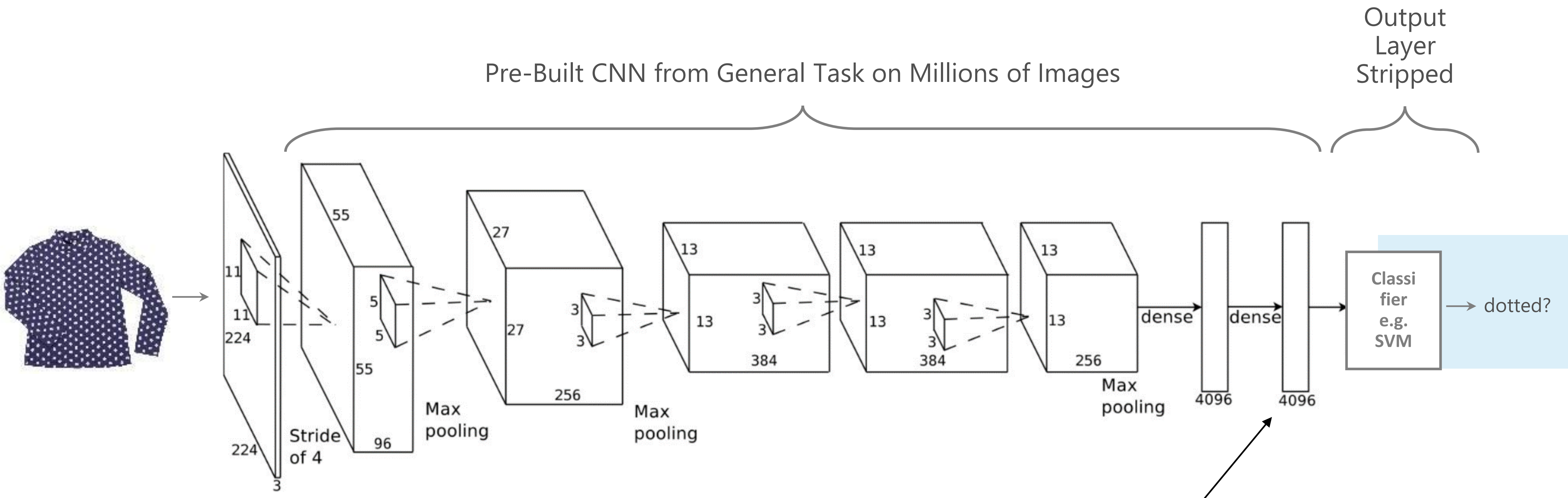
Striped

Dotted

Argyle

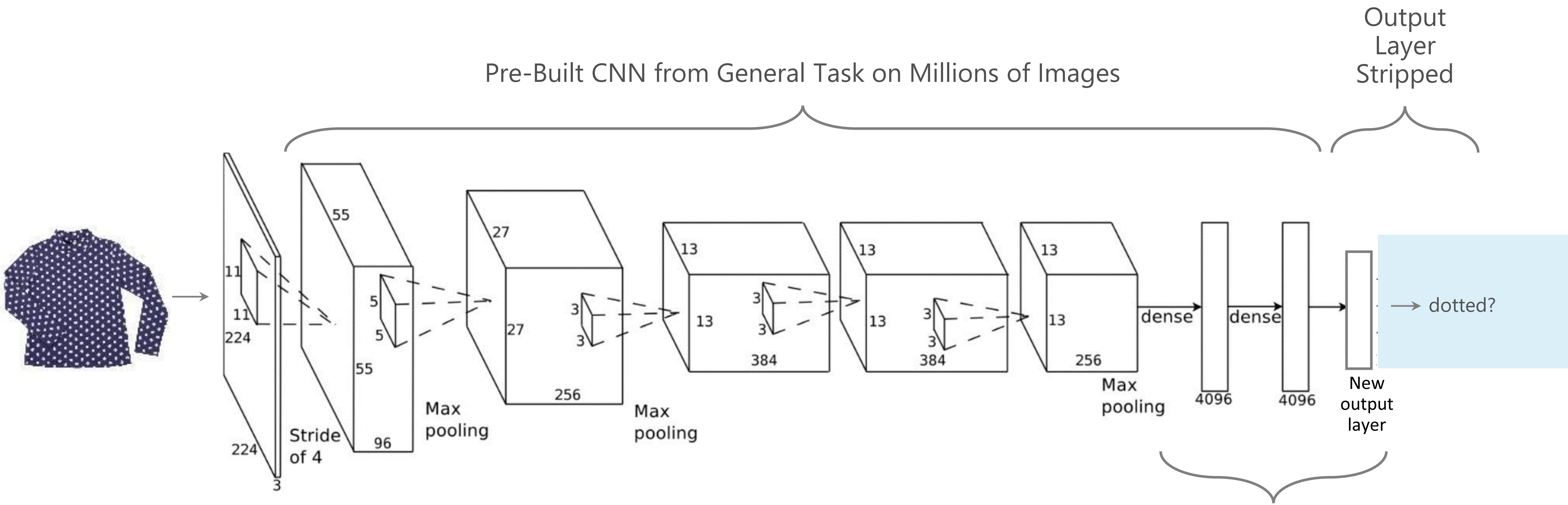
Can we apply transfer learning to accurately classify clothing texture?

Using a Pre-Trained CNN as a **Featurizer**



Outputs of penultimate layer of ImageNet Trained CNN provide excellent general purpose image features

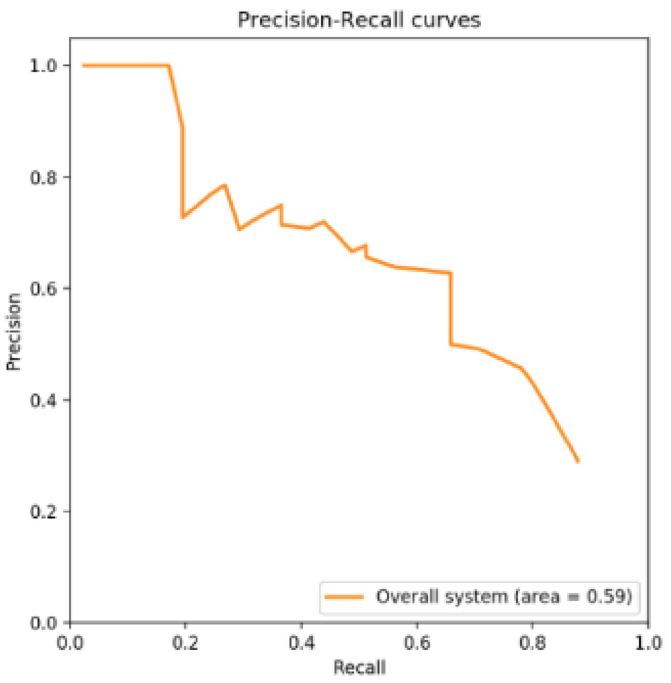
Using a Pre-Trained CNN and **Finetune**



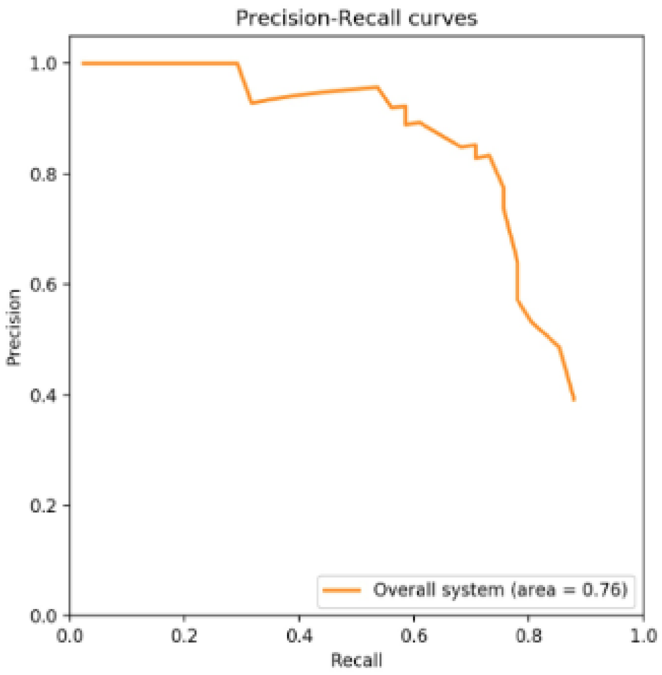
Using a pre-trained DNN, an accurate model can be achieved with thousands (or less) of labeled examples instead of millions

Transfer Learning Results - Texture Dataset

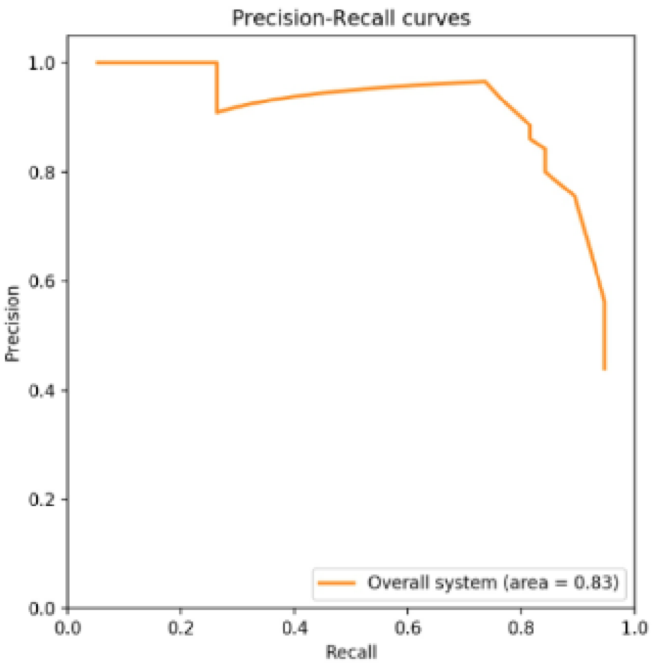
DNN featurization
Input Image Size: 224x224 pixels
Area Under Curve: 0.59
Classification Accuracy: **69.0%**



Fine-tuning (full CNN)
Input Image Size: 224x224 pixels
Area Under Curve: 0.76
Classification Accuracy: **77.4%**



Fine-tuning (full CNN)
Input Image Size: 896x886 pixels
Area Under Curve: 0.83
Classification Accuracy: **88.2%**



Deployment as Rest API

Easy deployment using AML Workbench.

Cluster management using Kubernetes.

Notebook provided to do measure average response time

- ➢ Experiment: #requests: 100, #concurrent requests: 10.

Number of replicas	Average round trip response time
1	536.99ms
10	81.88ms

Deep Learning using Azure Machine Learning Services.

- Bird Detection : Link
- Skin Cancer Detection : Link
- Shirts Classification : Link
- Aerial Image Classification : Link
- Document Collection Analysis : Link
- AI Real world Usecases : Link
- Energy Demand Forecasting : Link

Thank you