MACHINE LEARNING DESIGN, DEMYSTIFIED

SATURN 2018 Tutorial | May 8 | Plano

Carnegie Mellon University
Software Engineering Institute

softserve

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM18-0886

INTRODUCTIONS

Rick Kazman

Professor, University of Hawaii
Research Scientist, SEI

Serge Haziyev

Head of Intelligent Enterprise, SoftServe

Iurii Milovanov

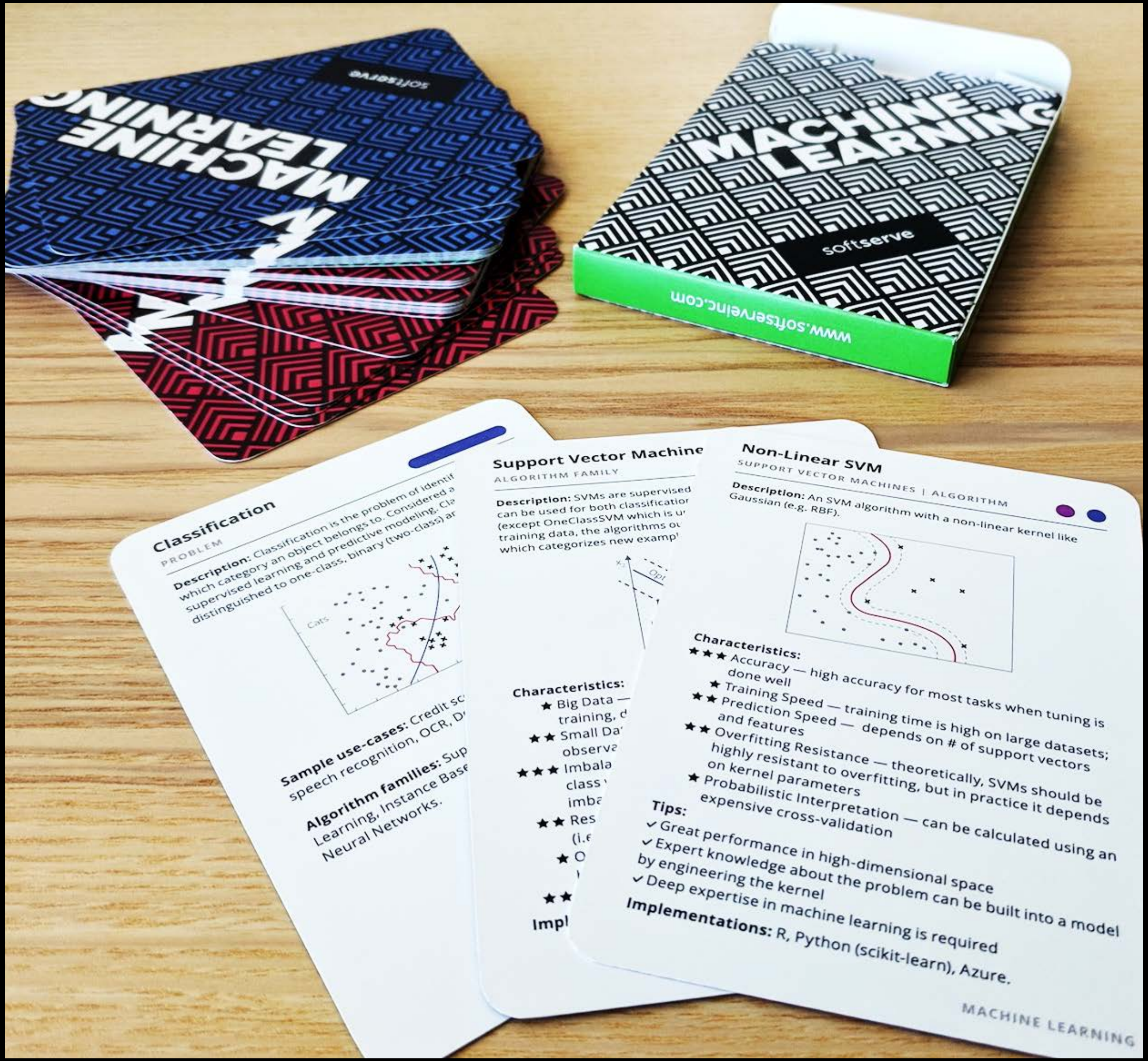
Data Science Practice Leader, SoftServe

Carnegie Mellon University
Software Engineering Institute

softserve

AGENDA

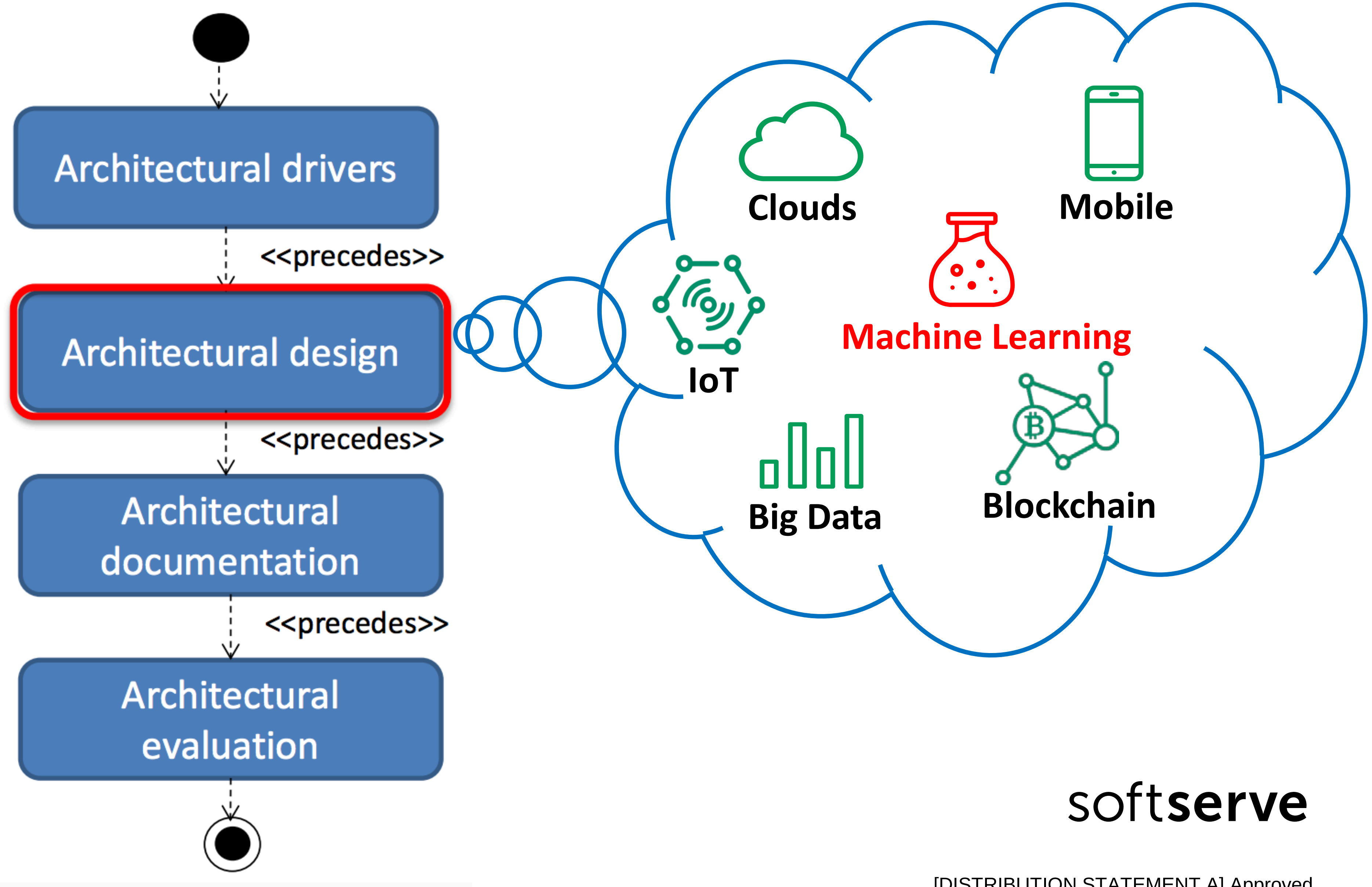
- A Bit of Background
- Game ☺
- Prototyping
- Summary & QA



Carnegie Mellon University
Software Engineering Institute

softserve

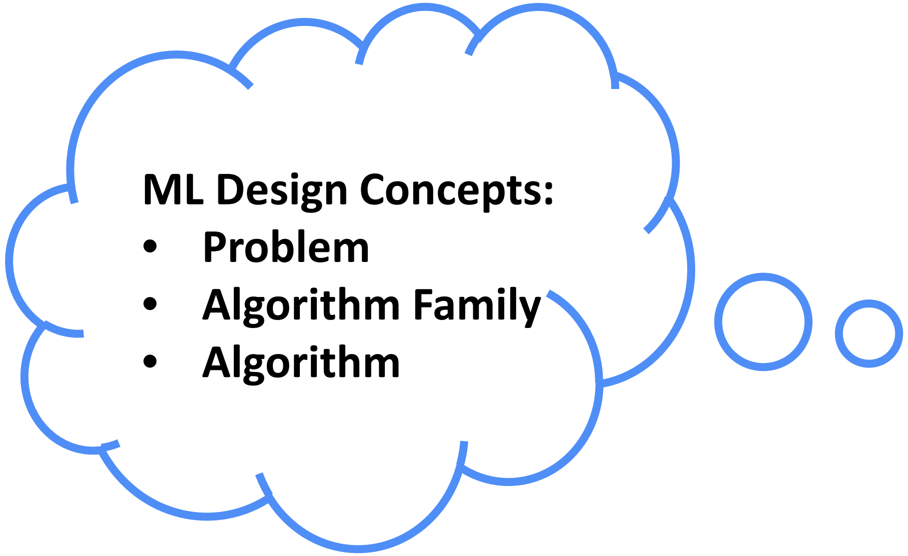
MOTIVATION

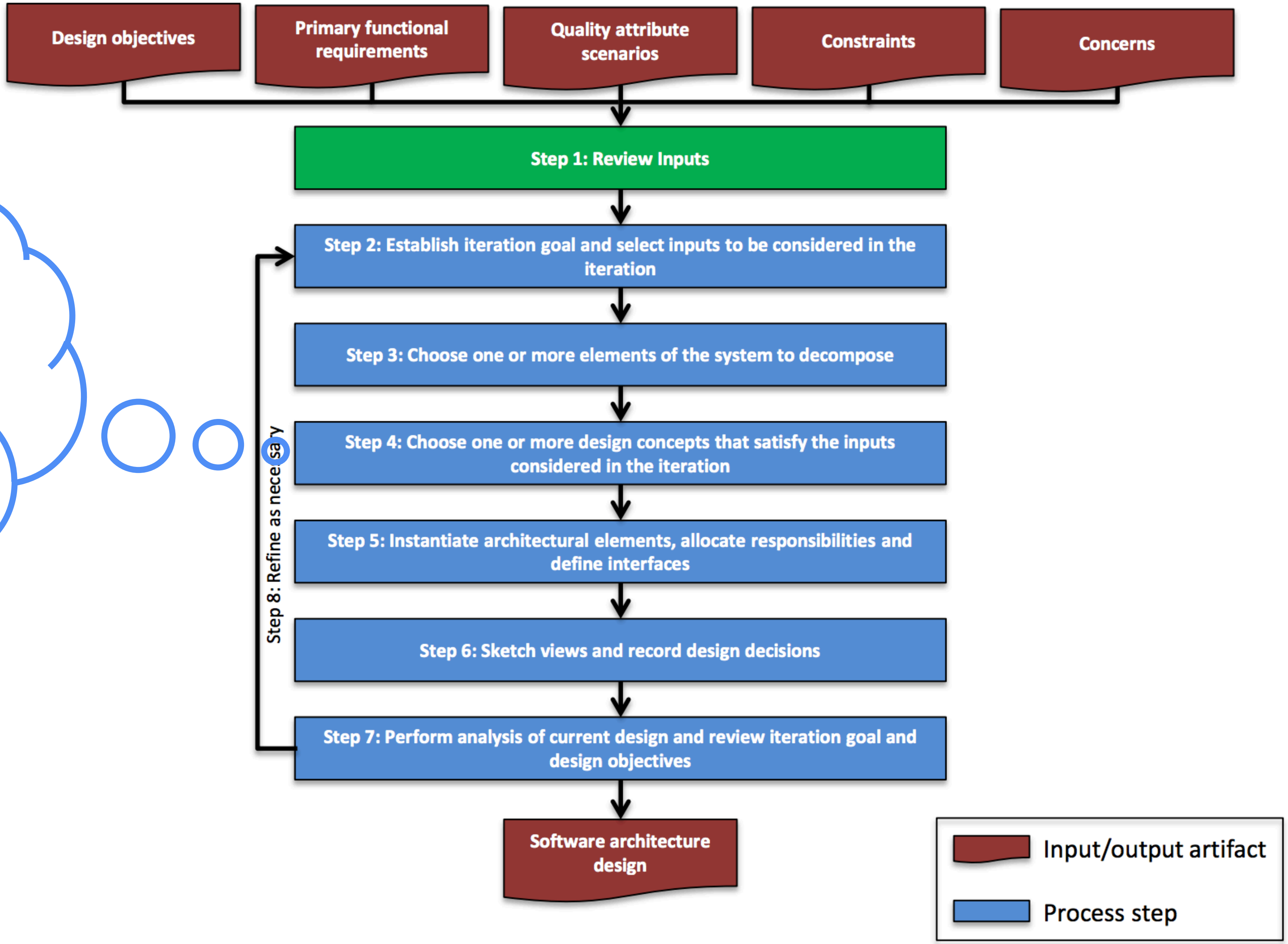


Carnegie Mellon University
Software Engineering Institute

softserve

ADD 3.0





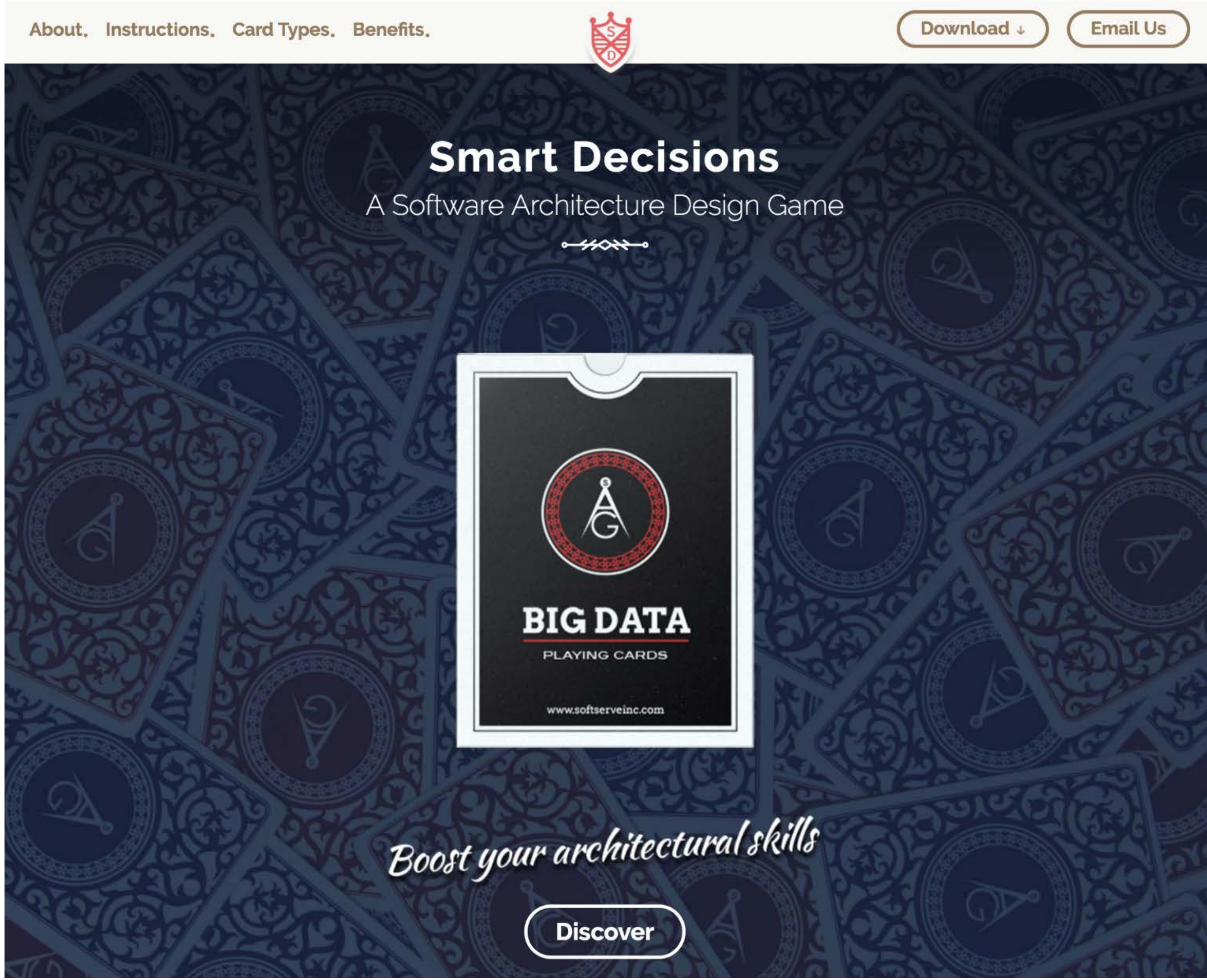
Carnegie Mellon University
Software Engineering Institute

SMART DECISIONS GAME

First presented at SATURN 2015

A fun, lightweight way to introduce architectural design and ADD

Available at:
http://smartdecisionsgame.com/



softserve

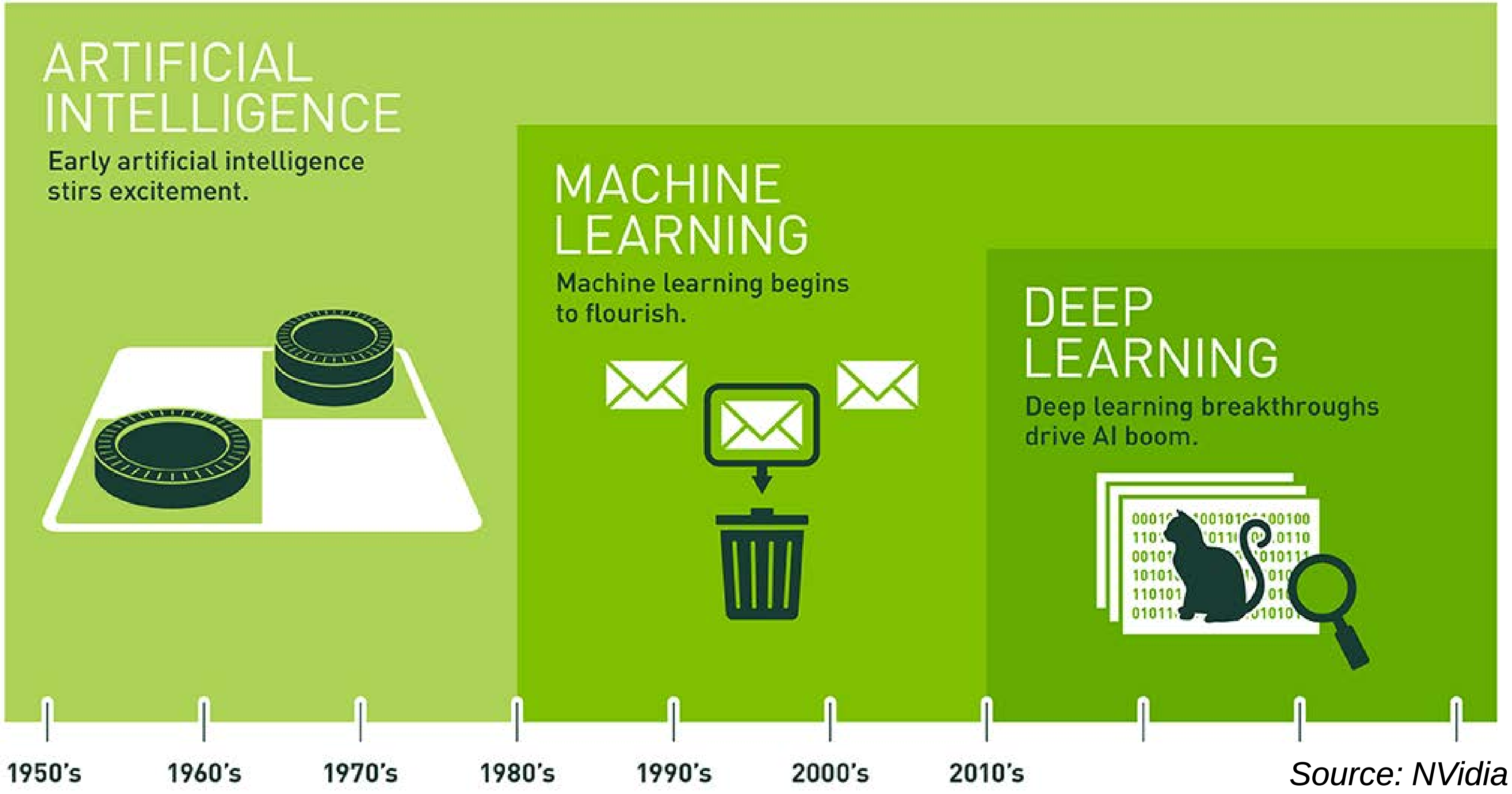
Carnegie Mellon University
Software Engineering Institute

SHORT QUIZ ☺

What’s the name of this company in AI field?

10x increase over the past 2 years!

AI PROGRESS SINCE 1950s

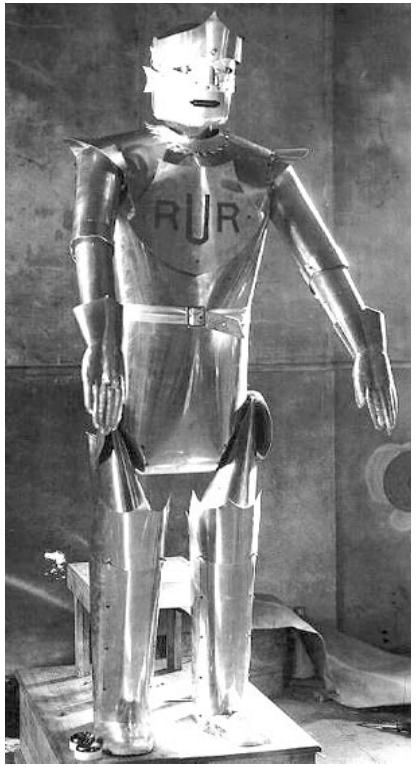


Source: NVidia

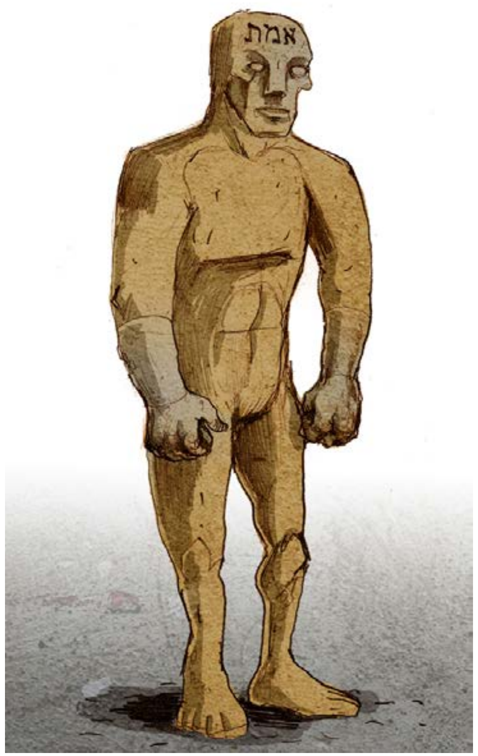
Carnegie Mellon University
Software Engineering Institute

softserve

MYTHS AND FICTION ABOUT ARTIFICIAL BEINGS



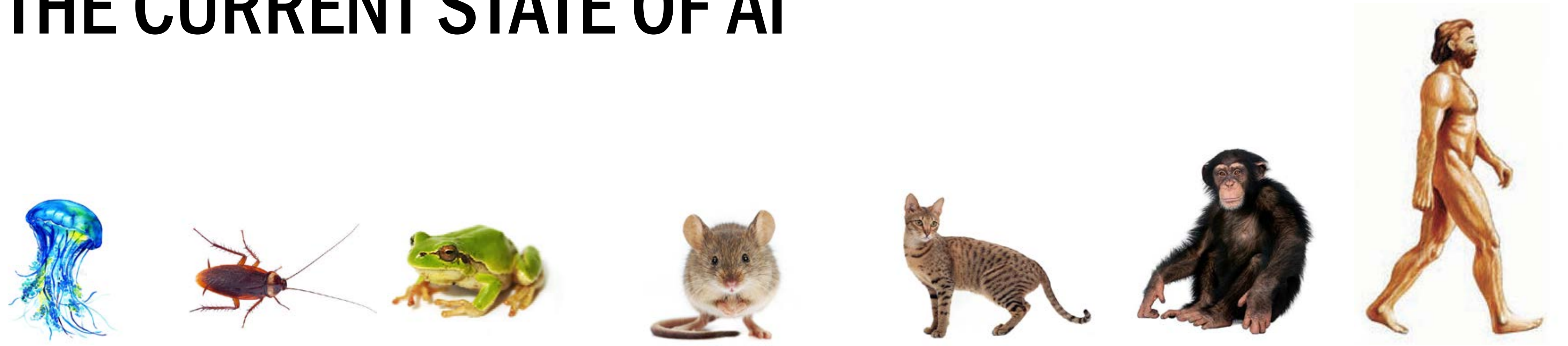
R.U.R. (Karel Čapek)
1921



Golem (Bible)
~1000 BC

Sumerian Anunnaki creating the first man
~2300 BC

THE CURRENT STATE OF AI



5
K

1M

16M

71M

760M

22B

86B

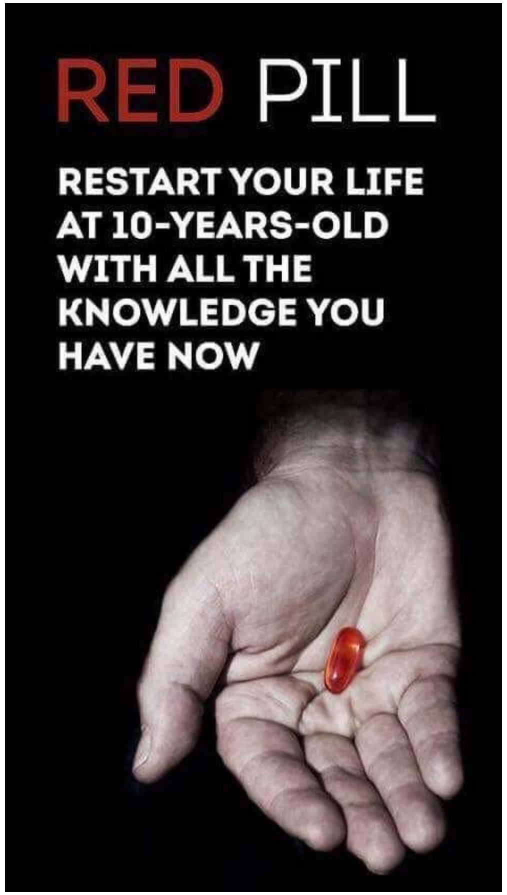
Number of neurons

Carnegie Mellon University
Software Engineering Institute

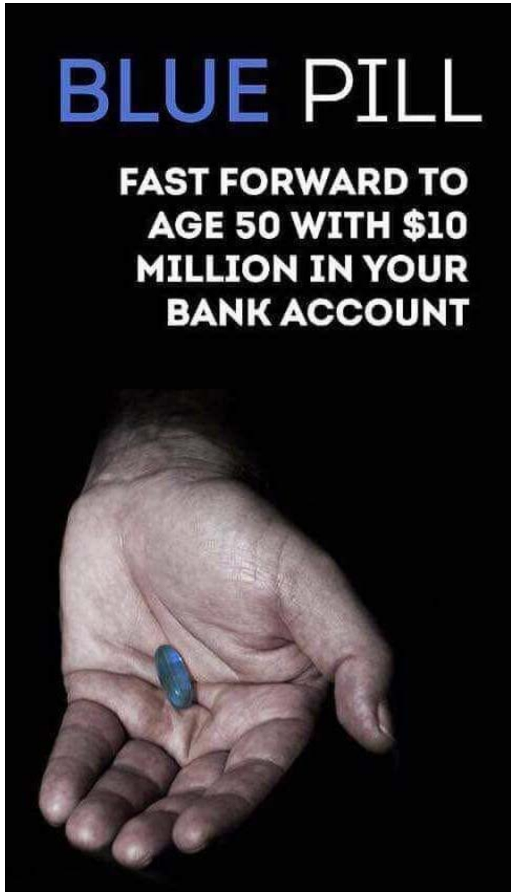
softserve

GAME CHALLENGE OVERVIEW

Business Use Case



Banner A



Banner B

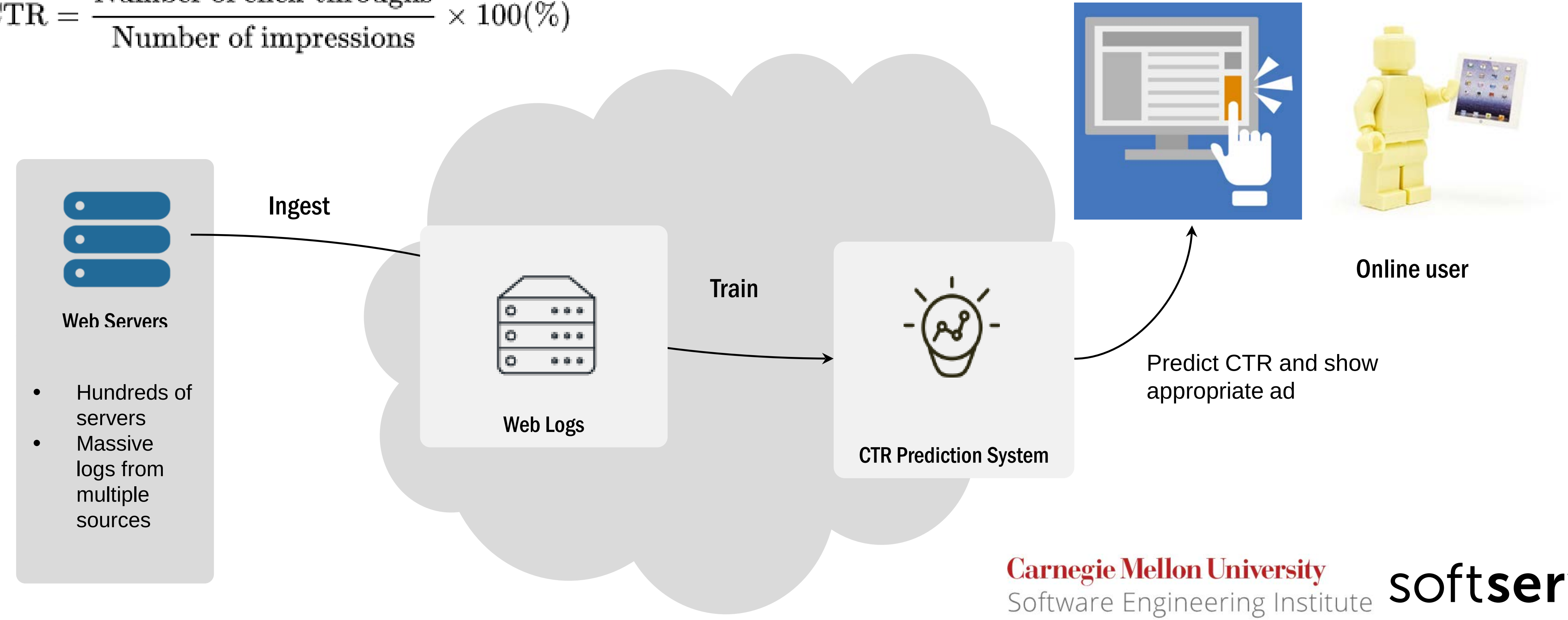
Which ad will the user choose?

Carnegie Mellon University
Software Engineering Institute

softserve

GAME CHALLENGE OVERVIEW
Marketecture Diagram

$$\mathrm{CTR} = \frac{\text{Number of click-throughs}}{\text{Number of impressions}} \times 100(\%)$$



Carnegie Mellon University
Software Engineering Institute
softserve

WHY DO WE NEED **MACHINE LEARNING?**

Carnegie Mellon University
Software Engineering Institute

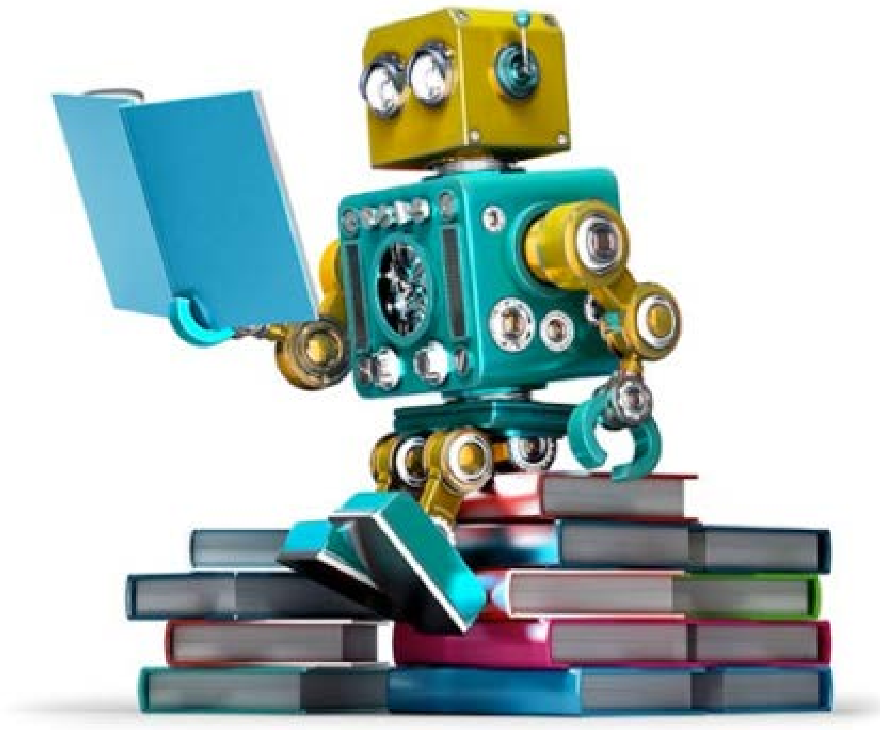
softserve

WHY NOT JUST CODING?!

Most of the today's AI problems:

- Deal with an infinite problem space – think about how many words are there in the English language
- Poorly defined – we still do not know how our brain solves problems

Therefore, traditional rule-based hand-coding for such problems suffers a 'complexity collapse' and is not feasible



Carnegie Mellon University
Software Engineering Institute

softserve

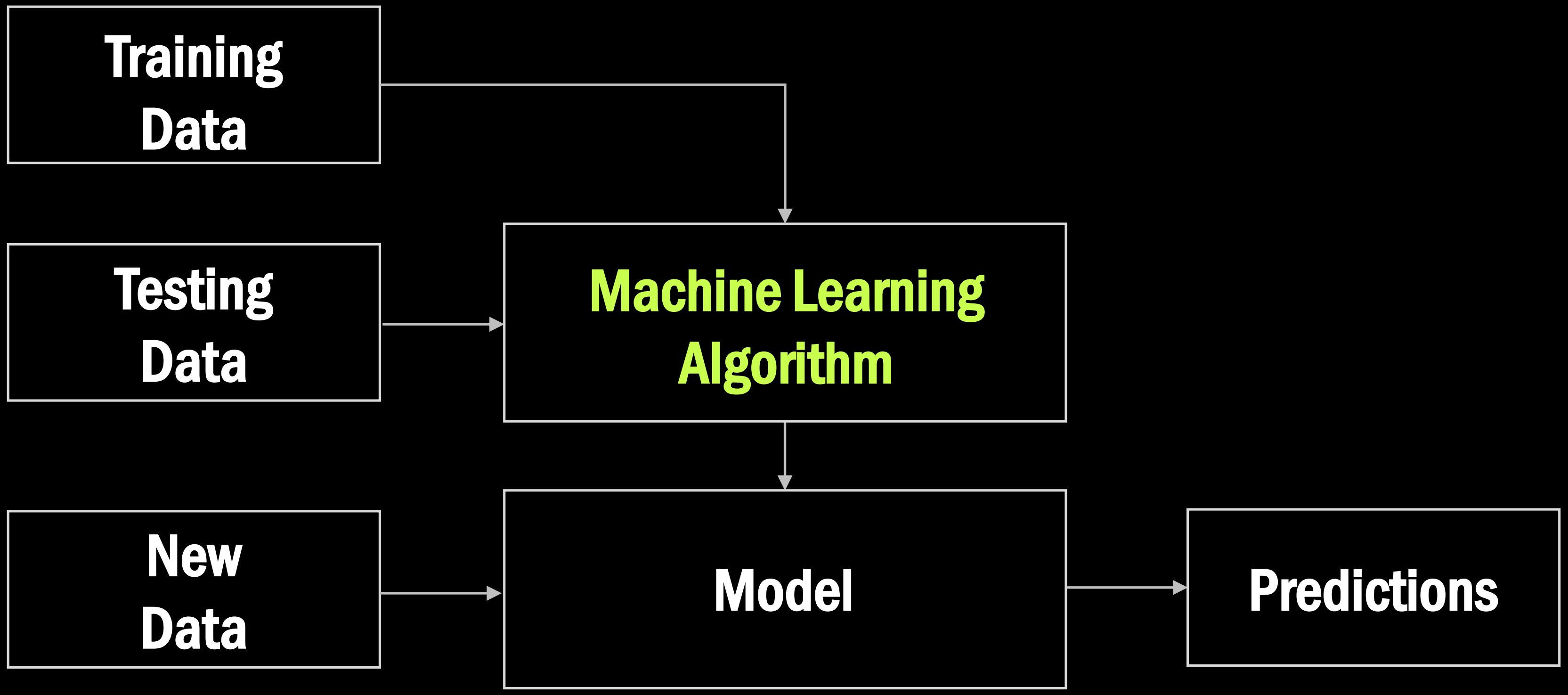
MACHINE LEARNING APPROACH

Instead of writing a program by hand, we use a set of examples to train the algorithm

Developer writes code

Algorithm "writes code"

ML BUILDING BLOCKS



Carnegie Mellon University
Software Engineering Institute

softserve

TYPES OF LEARNING

Carnegie Mellon University
Software Engineering Institute

softserve

SUPERVISED LEARNING

- **Input examples** and corresponding **ground truth outputs** are provided
- The goal is to learn general rules that map a new example to the predicted output

SUPERVISED LEARNING

Input Data

Output Data

Example: Given a set of **house features** along with corresponding **house prices**, predict a price for a new house based on its features (e.g. size, location, etc.)

UNSUPERVISED LEARNING

- Only **input examples** are provided
- No explicit information about ground truth
- The algorithm tries to discover the internal structure of the data based on some prior knowledge about desired outcome

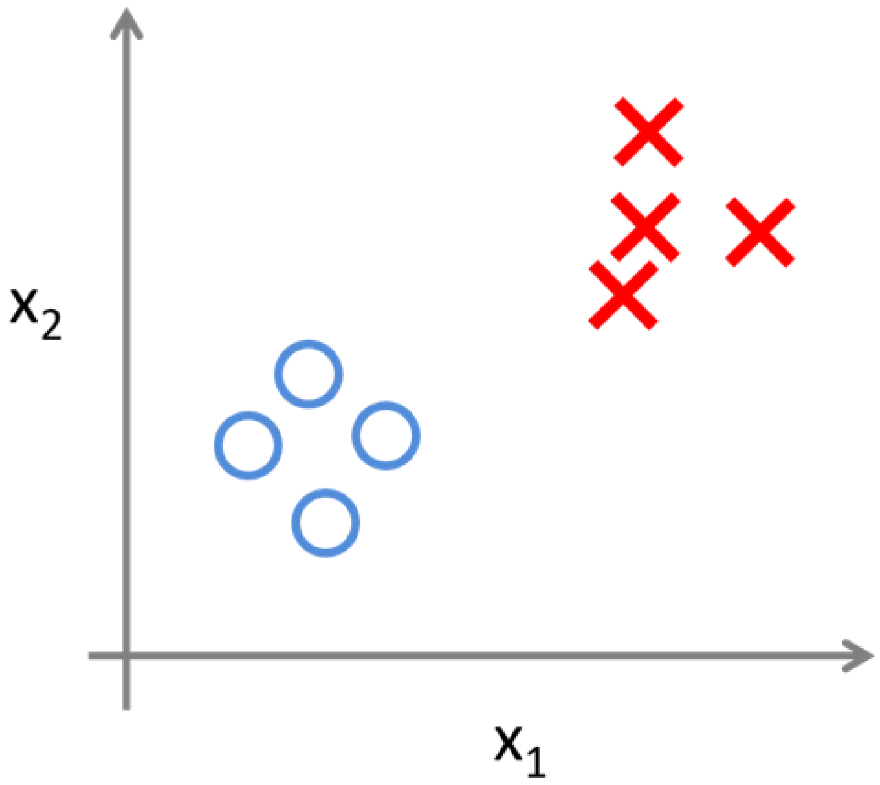
UNSUPERVISED LEARNING

Input Data

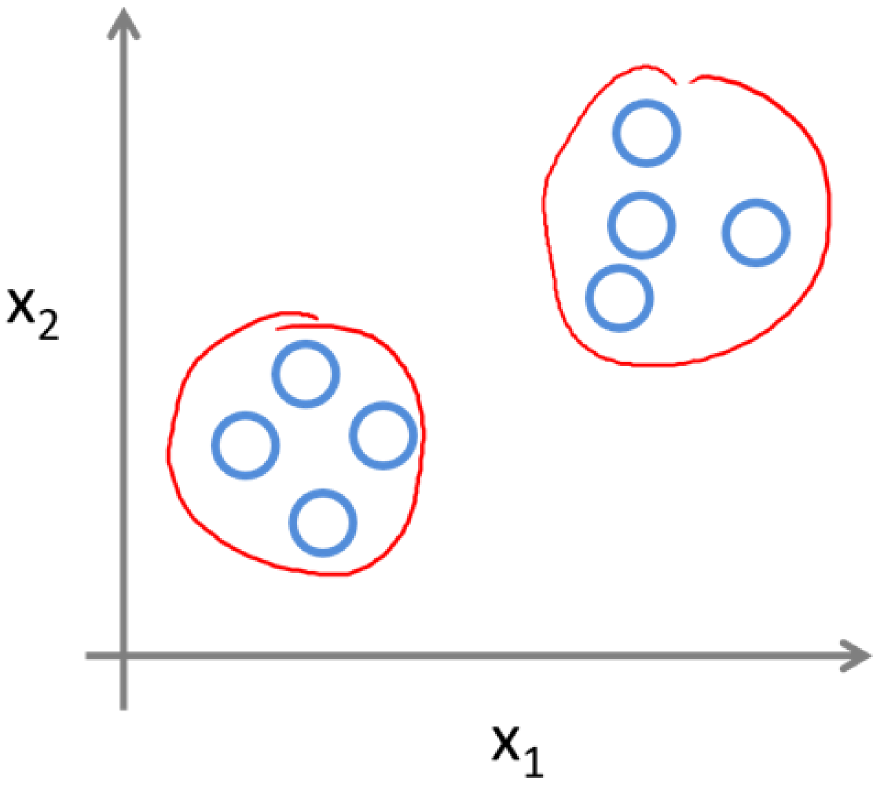
Example: Given a set of **customer transactions** discover what would be the best way to group them into clusters based on **customer similarity**

Output Preferences

SUPERVISED LEARNING



UNSUPERVISED LEARNING

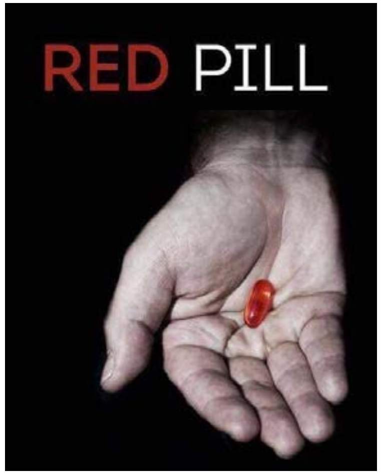


ITERATION 1:
What type of learning best fits a given use case?

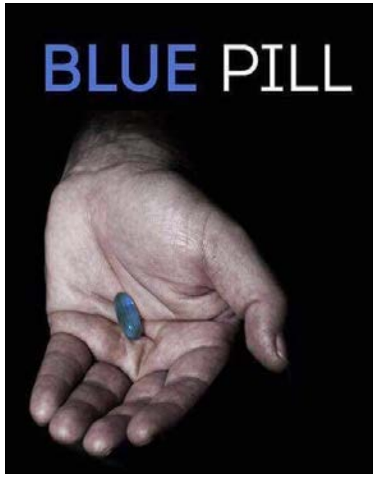
Select from: supervised or unsupervised

ITERATION 1:
Supervised or Unsupervised Learning?

Historical Dataset



Banner A



Banner B

Logs:

User X Features | Banner A | Click 0
User Y Features | Banner B | Click 0
User X Features | Banner B | Click 1
…

Train
Input: Logs

ML Algorithm

Predict
Input: User Features
Output: Most preferred banner to show

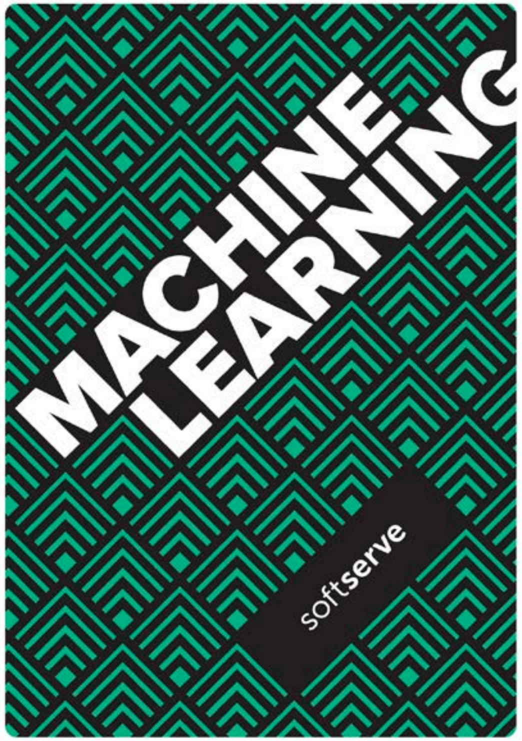
MACHINE LEARNING CARDS

Carnegie Mellon University
Software Engineering Institute

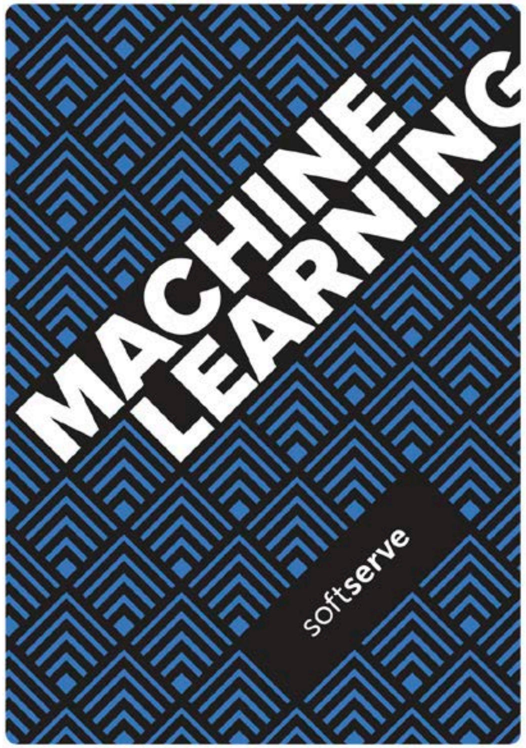
softserve

MACHINE LEARNING CARDS

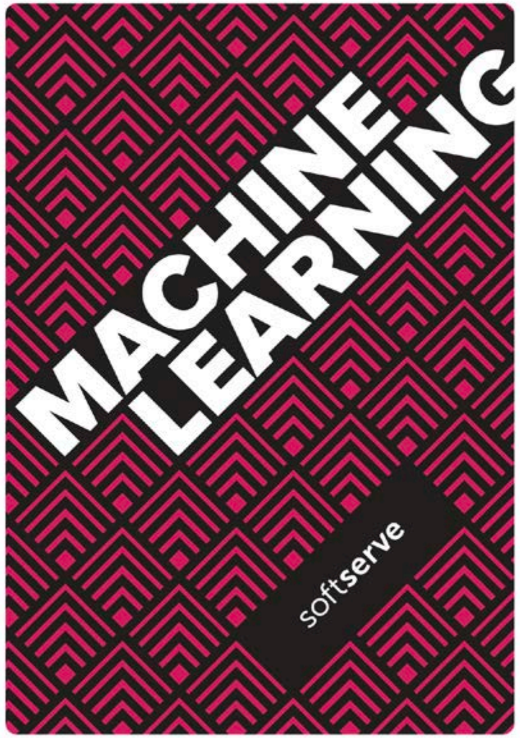
ITERATION 2:
PROBLEM TYPE



ITERATION 3a:
ALGORITHM FAMILY

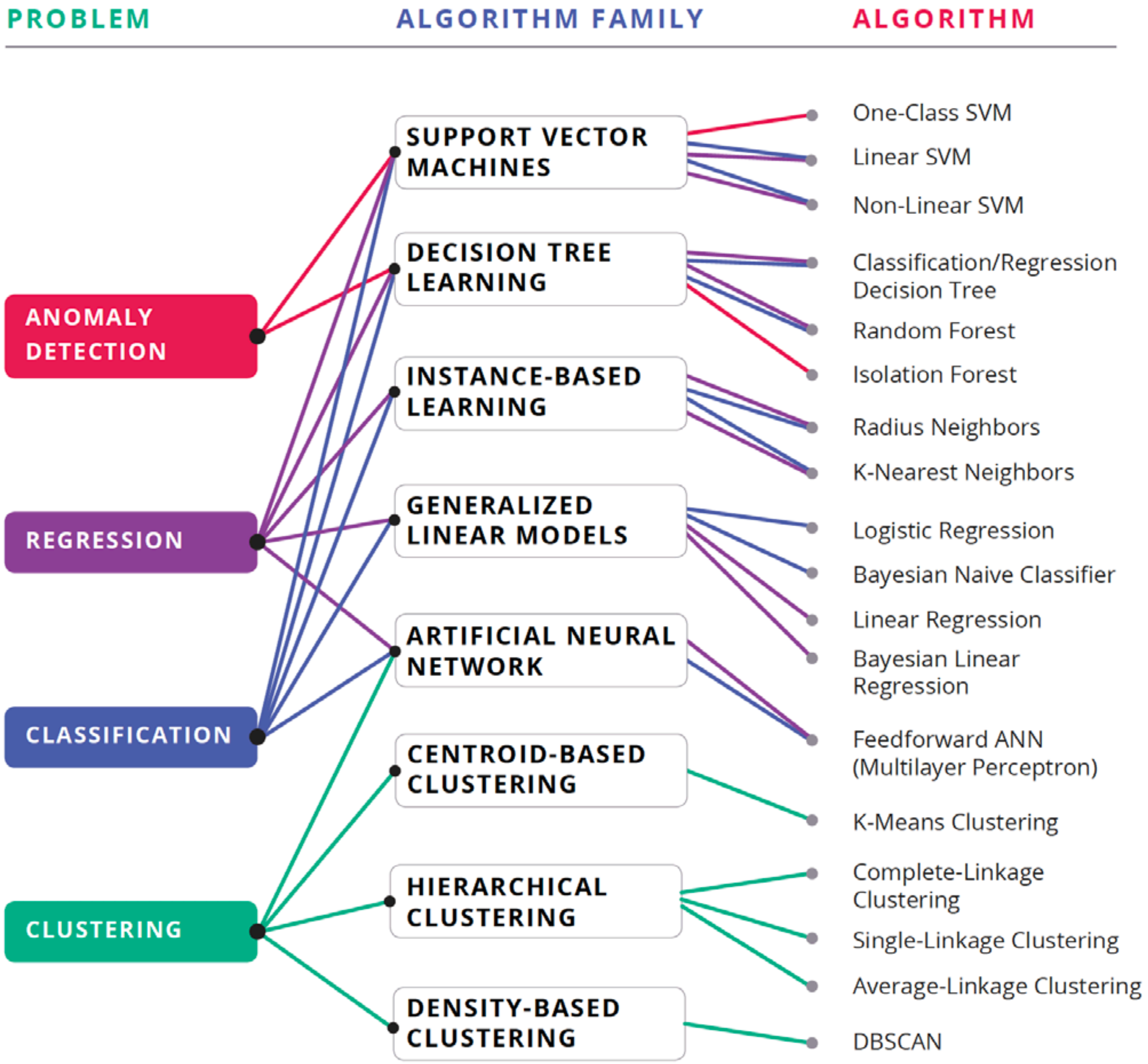


ITERATION 3b:
ML ALGORITHM

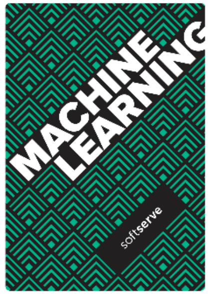


Carnegie Mellon University
Software Engineering Institute

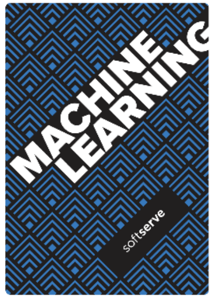
softserve



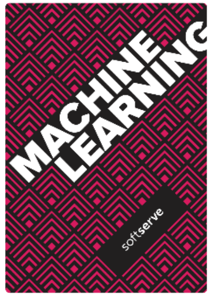
Legend:



- Problem cards



- Algorithm Family cards



- Algorithm cards

Carnegie Mellon University
Software Engineering Institute

softserve

PROBLEM TYPES

Carnegie Mellon University
Software Engineering Institute

softserve

CLASSIFICATION

Key Highlights:

- Identifies which category an object belongs to
- Supervised learning problem

Examples:

- Detect fraudulent transactions (one-class)
- Categorize emails by spam or not spam (binary)
- Categorize articles based on their topic (multi-class)
- Detect objects on the image (multi-label)

REGRESSION

Key Highlights:

- Predict a continuous value associated with an object
- Supervised learning problem

Examples:

- Predict stock prices from market data
- Score a credit application based on historical data
- Estimate demand for a given product

CLUSTERING

Key Highlights:

- Group similar objects into clusters
- Unsupervised learning problem

Examples:

- Discover audiences to target on social networks
- Group checking data based on GEO-proximity
- Detect common topics in corporate knowledge base

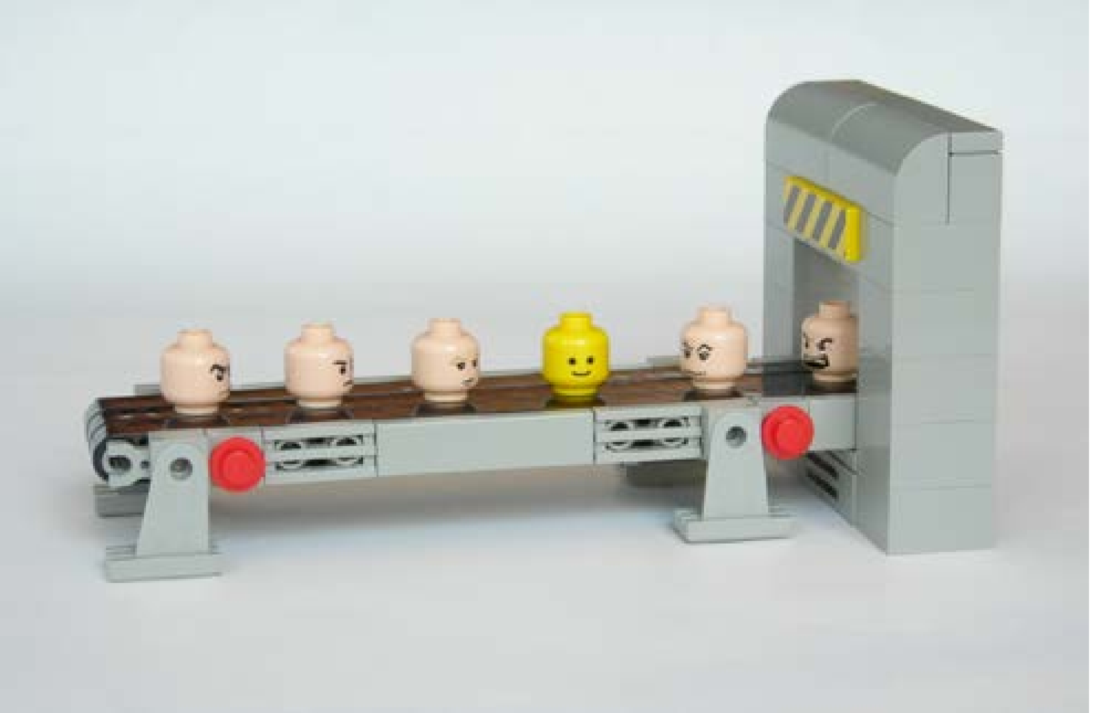
ANOMALY DETECTION

Key Highlights:

- Identify observations that do not conform to an expected pattern
- Addresses both supervised and unsupervised learning

Examples:

- Identify fraudulent transactions or abnormal customer behavior
- In manufacturing, detect physical parts that are likely to fail in the near future

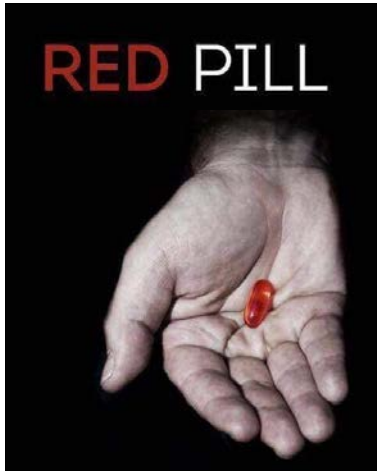


ITERATION 2:
What type of problem best fits a given use case?

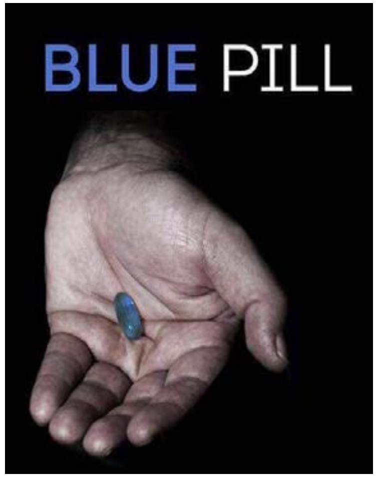
Select problem card from: classification, regression, clustering or anomaly detection

ITERATION 2:
What type of problem?

Historical Dataset



Banner A



Banner B

Logs:

User X Features | Banner A | Click 0
User Y Features | Banner B | Click 0
User X Features | Banner B | Click 1
…

Train
Input: Logs

ML Algorithm

Predict
Input: User Features
Output: Most preferred banner to show

Carnegie Mellon University
Software Engineering Institute

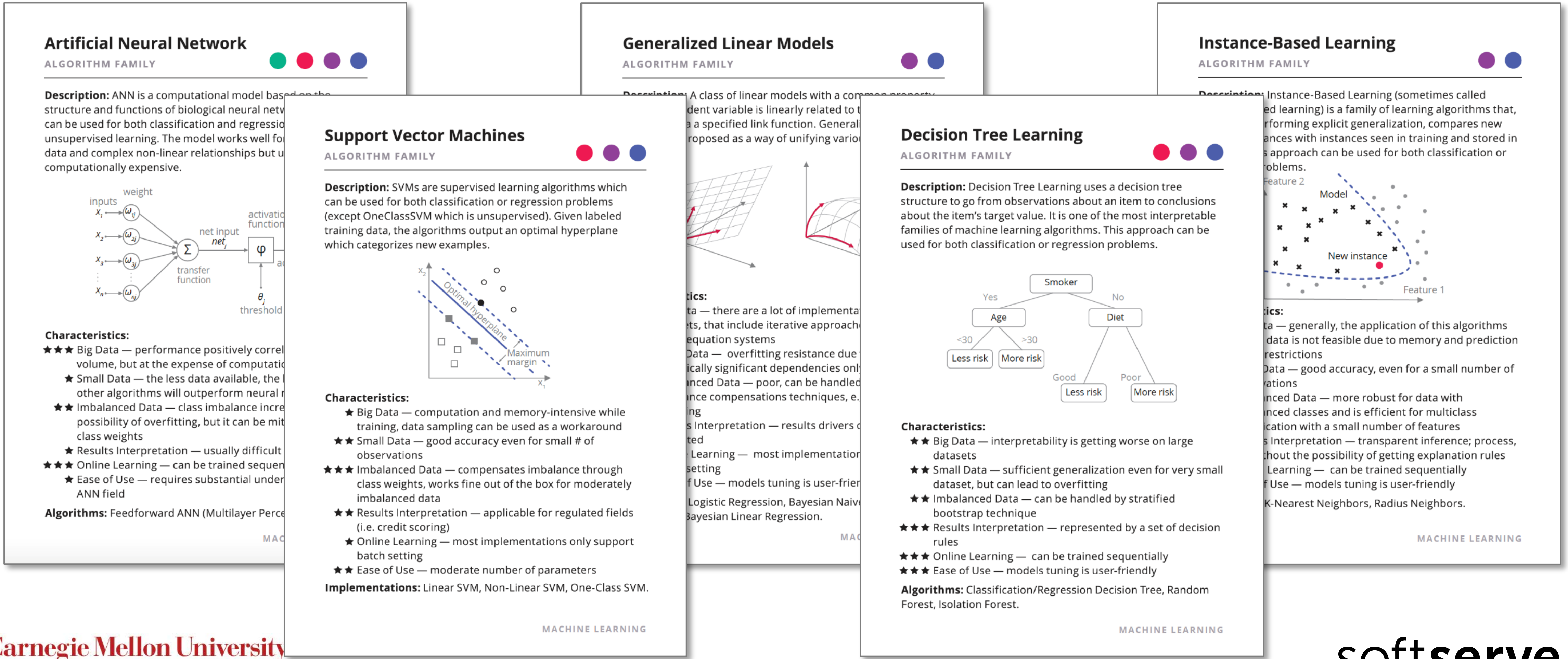
softserve

FAMILIES AND ALGORITHMS

Carnegie Mellon University
Software Engineering Institute

softserve

CLASSIFICATION FAMILIES


Artificial Neural Network

ALGORITHM FAMILY

Description: ANN is a computational model based on the structure and functions of biological neural netw… can be used for both classification and regressio… unsupervised learning. The model works well fo… data and complex non-linear relationships but u… computationally expensive.

Characteristics:

★★★ Big Data — performance positively correl… volume, but at the expense of computati…

★ Small Data — the less data available, the … other algorithms will outperform neural …

★★ Imbalanced Data — class imbalance incre… possibility of overfitting, but it can be mit… class weights

★ Results Interpretation — usually difficult

★★★ Online Learning — can be trained sequen…

★ Ease of Use — requires substantial under… ANN field

Algorithms: Feedforward ANN (Multilayer Perce…

Generalized Linear Models

ALGORITHM FAMILY

Instance-Based Learning

ALGORITHM FAMILY

Description: Instance-Based Learning (sometimes called …ed learning) is a family of learning algorithms that, …rforming explicit generalization, compares new …ances with instances seen in training and stored in …s approach can be used for both classification or …oblems.

…K-Nearest Neighbors, Radius Neighbors.

MACHINE LEARNING

Support Vector Machines

ALGORITHM FAMILY

Description: SVMs are supervised learning algorithms which can be used for both classification or regression problems (except OneClassSVM which is unsupervised). Given labeled training data, the algorithms output an optimal hyperplane which categorizes new examples.

Characteristics:

★ Big Data — computation and memory-intensive while training, data sampling can be used as a workaround

★★ Small Data — good accuracy even for small # of observations

★★★ Imbalanced Data — compensates imbalance through class weights, works fine out of the box for moderately imbalanced data

★★ Results Interpretation — applicable for regulated fields (i.e. credit scoring)

★ Online Learning — most implementations only support batch setting

★★ Ease of Use — moderate number of parameters

Implementations: Linear SVM, Non-Linear SVM, One-Class SVM.

MACHINE LEARNING

Decision Tree Learning

ALGORITHM FAMILY

Description: Decision Tree Learning uses a decision tree structure to go from observations about an item to conclusions about the item's target value. It is one of the most interpretable families of machine learning algorithms. This approach can be used for both classification or regression problems.

Characteristics:

★★ Big Data — interpretability is getting worse on large datasets

★★ Small Data — sufficient generalization even for very small dataset, but can lead to overfitting

★★ Imbalanced Data — can be handled by stratified bootstrap technique

★★★ Results Interpretation — represented by a set of decision rules

★★★ Online Learning — can be trained sequentially

★★★ Ease of Use — models tuning is user-friendly

Algorithms: Classification/Regression Decision Tree, Random Forest, Isolation Forest.

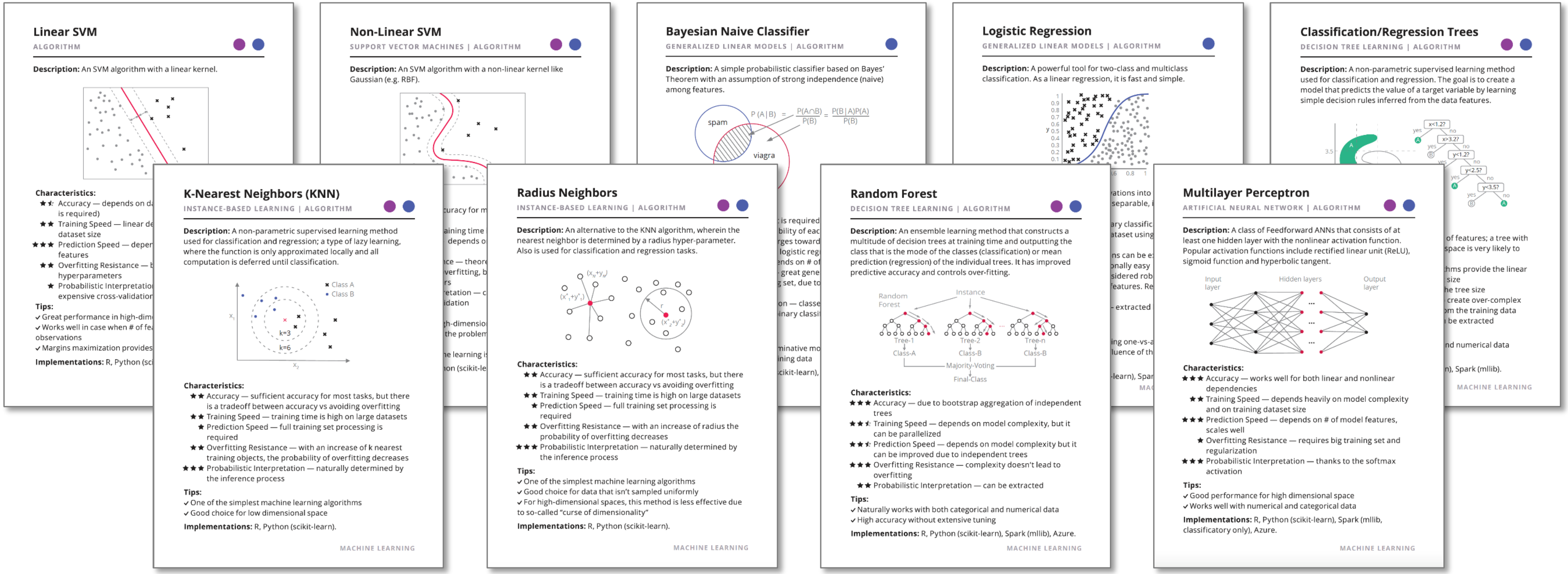
MACHINE LEARNING

Carnegie Mellon University
Software Engineering Institute

softserve

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

CLASSIFICATION ALGORITHMS



Linear SVM
ALGORITHM

Description: An SVM algorithm with a linear kernel.

Non-Linear SVM
SUPPORT VECTOR MACHINES | ALGORITHM

Description: An SVM algorithm with a non-linear kernel like Gaussian (e.g. RBF).

Bayesian Naive Classifier
GENERALIZED LINEAR MODELS | ALGORITHM

Description: A simple probabilistic classifier based on Bayes' Theorem with an assumption of strong independence (naive) among features.

Logistic Regression
GENERALIZED LINEAR MODELS | ALGORITHM

Description: A powerful tool for two-class and multiclass classification. As a linear regression, it is fast and simple.

Classification/Regression Trees
DECISION TREE LEARNING | ALGORITHM

Description: A non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

K-Nearest Neighbors (KNN)
INSTANCE-BASED LEARNING | ALGORITHM

Description: A non-parametric supervised learning method used for classification and regression; a type of lazy learning, where the function is only approximated locally and all computation is deferred until classification.

Characteristics:
- ★★ Accuracy — sufficient accuracy for most tasks, but there is a tradeoff between accuracy vs avoiding overfitting
- ★★ Training Speed — training time is high on large datasets
- ★ Prediction Speed — full training set processing is required
- ★★ Overfitting Resistance — with an increase of k nearest training objects, the probability of overfitting decreases
- ★★★ Probabilistic Interpretation — naturally determined by the inference process

Tips:
- ✓ One of the simplest machine learning algorithms
- ✓ Good choice for low dimensional space

Implementations: R, Python (scikit-learn).

MACHINE LEARNING

Radius Neighbors
INSTANCE-BASED LEARNING | ALGORITHM

Description: An alternative to the KNN algorithm, wherein the nearest neighbor is determined by a radius hyper-parameter. Also is used for classification and regression tasks.

Characteristics:
- ★★ Accuracy — sufficient accuracy for most tasks, but there is a tradeoff between accuracy vs avoiding overfitting
- ★★ Training Speed — training time is high on large datasets
- ★ Prediction Speed — full training set processing is required
- ★★ Overfitting Resistance — with an increase of radius the probability of overfitting decreases
- ★★★ Probabilistic Interpretation — naturally determined by the inference process

Tips:
- ✓ One of the simplest machine learning algorithms
- ✓ Good choice for data that isn't sampled uniformly
- ✓ For high-dimensional spaces, this method is less effective due to so-called "curse of dimensionality"

Implementations: R, Python (scikit-learn).

MACHINE LEARNING

Random Forest
DECISION TREE LEARNING | ALGORITHM

Description: An ensemble learning method that constructs a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It has improved predictive accuracy and controls over-fitting.

Characteristics:
- ★★★ Accuracy — due to bootstrap aggregation of independent trees
- ★★½ Training Speed — depends on model complexity, but it can be parallelized
- ★★½ Prediction Speed — depends on model complexity but it can be improved due to independent trees
- ★★★ Overfitting Resistance — complexity doesn't lead to overfitting
- ★★ Probabilistic Interpretation — can be extracted

Tips:
- ✓ Naturally works with both categorical and numerical data
- ✓ High accuracy without extensive tuning

Implementations: R, Python (scikit-learn), Spark (mllib), Azure.

MACHINE LEARNING

Multilayer Perceptron
ARTIFICIAL NEURAL NETWORK | ALGORITHM

Description: A class of Feedforward ANNs that consists of at least one hidden layer with the nonlinear activation function. Popular activation functions include rectified linear unit (ReLU), sigmoid function and hyperbolic tangent.

Characteristics:
- ★★★ Accuracy — works well for both linear and nonlinear dependencies
- ★★ Training Speed — depends heavily on model complexity and on training dataset size
- ★★★ Prediction Speed — depends on # of model features, scales well
- ★ Overfitting Resistance — requires big training set and regularization
- ★★★ Probabilistic Interpretation — thanks to the softmax activation

Tips:
- ✓ Good performance for high dimensional space
- ✓ Works well with numerical and categorical data

Implementations: R, Python (scikit-learn), Spark (mllib, classificatory only), Azure.

MACHINE LEARNING

Carnegie Mellon University
Software Engineering Institute

softserve

DECISION DRIVERS

Carnegie Mellon University
Software Engineering Institute

softserve

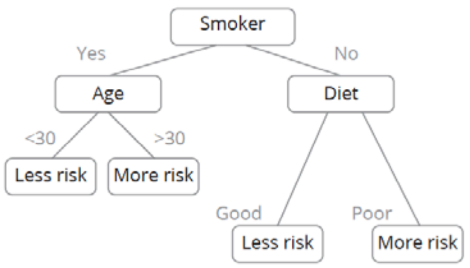
FAMILY DRIVERS

- **Big Data** – scalability and ability to leverage from new data
- Small Data – ability to learn from a few examples
- **Imbalanced Data** – ability to distinguish rare events
- Results Interpretation – human-friendly results
- Online Learning – ability to continuously train from new data
- **Ease of Use** – number of parameters to manually tune

Decision Tree Learning

ALGORITHM FAMILY

Description: Decision Tree Learning uses a decision tree structure to go from observations about an item to conclusions about the item's target value. It is one of the most interpretable families of machine learning algorithms. This approach can be used for both classification or regression problems.



Characteristics:

★★ Big Data — interpretability is getting worse on large datasets
★★ Small Data — sufficient generalization even for very small dataset, but can lead to overfitting
★★ Imbalanced Data — can be handled by stratified bootstrap technique
★★★ Results Interpretation — represented by a set of decision rules
★★★ Online Learning — can be trained sequentially
★★★ Ease of Use — models tuning is user-friendly

Algorithms: Classification/Regression Decision Tree, Random Forest, Isolation Forest.

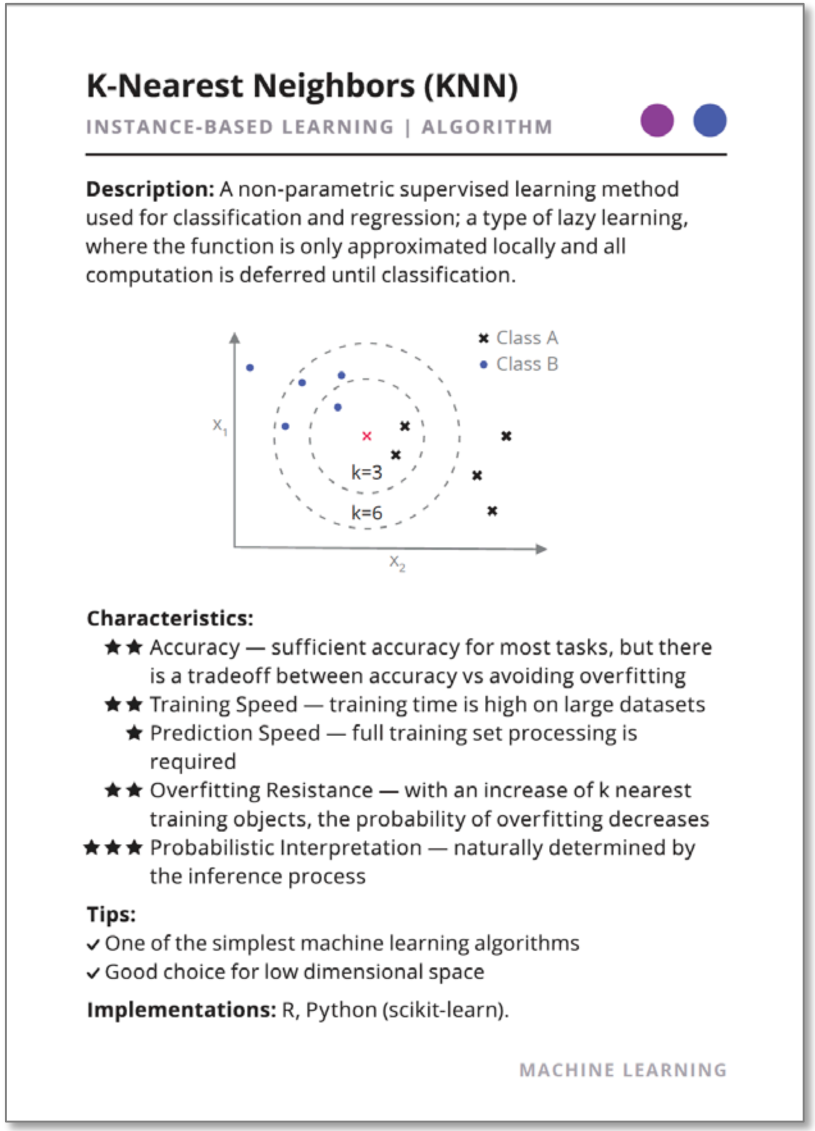
MACHINE LEARNING

Carnegie Mellon University
Software Engineering Institute

softserve

ALGORITHM DRIVERS

- **Accuracy** – ability to solve complex problems
- **Training Speed** – training runtime performance
- **Prediction Speed** – production runtime performance
- Overfitting Resistance – ability to generalize to new data
- Probabilistic Interpretation – return results as probabilities



K-Nearest Neighbors (KNN)

INSTANCE-BASED LEARNING | ALGORITHM

Description: A non-parametric supervised learning method used for classification and regression; a type of lazy learning, where the function is only approximated locally and all computation is deferred until classification.

Characteristics:

- ★★ Accuracy — sufficient accuracy for most tasks, but there is a tradeoff between accuracy vs avoiding overfitting
- ★★ Training Speed — training time is high on large datasets
- ★ Prediction Speed — full training set processing is required
- ★★ Overfitting Resistance — with an increase of k nearest training objects, the probability of overfitting decreases
- ★★★ Probabilistic Interpretation — naturally determined by the inference process

Tips:

- ✓ One of the simplest machine learning algorithms
- ✓ Good choice for low dimensional space

Implementations: R, Python (scikit-learn).

MACHINE LEARNING

Carnegie Mellon University
Software Engineering Institute

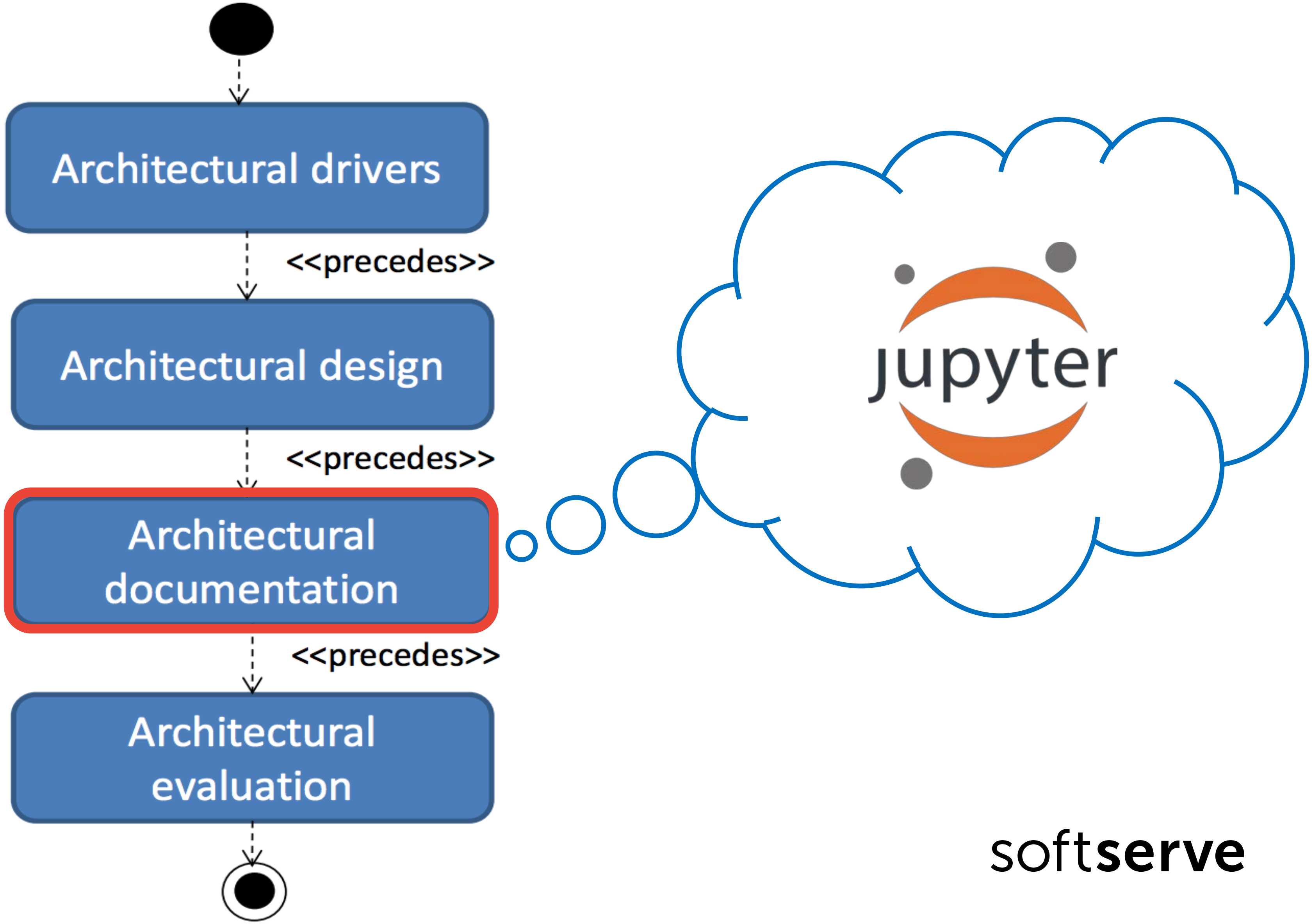
softserve

ITERATION 3:

Select a family and an algorithm card that would best fit a given use case

Family Key Drivers: Big Data, Imbalanced Data, Ease of Use
Algorithm Key Drivers: Accuracy, Training and Prediction Speed

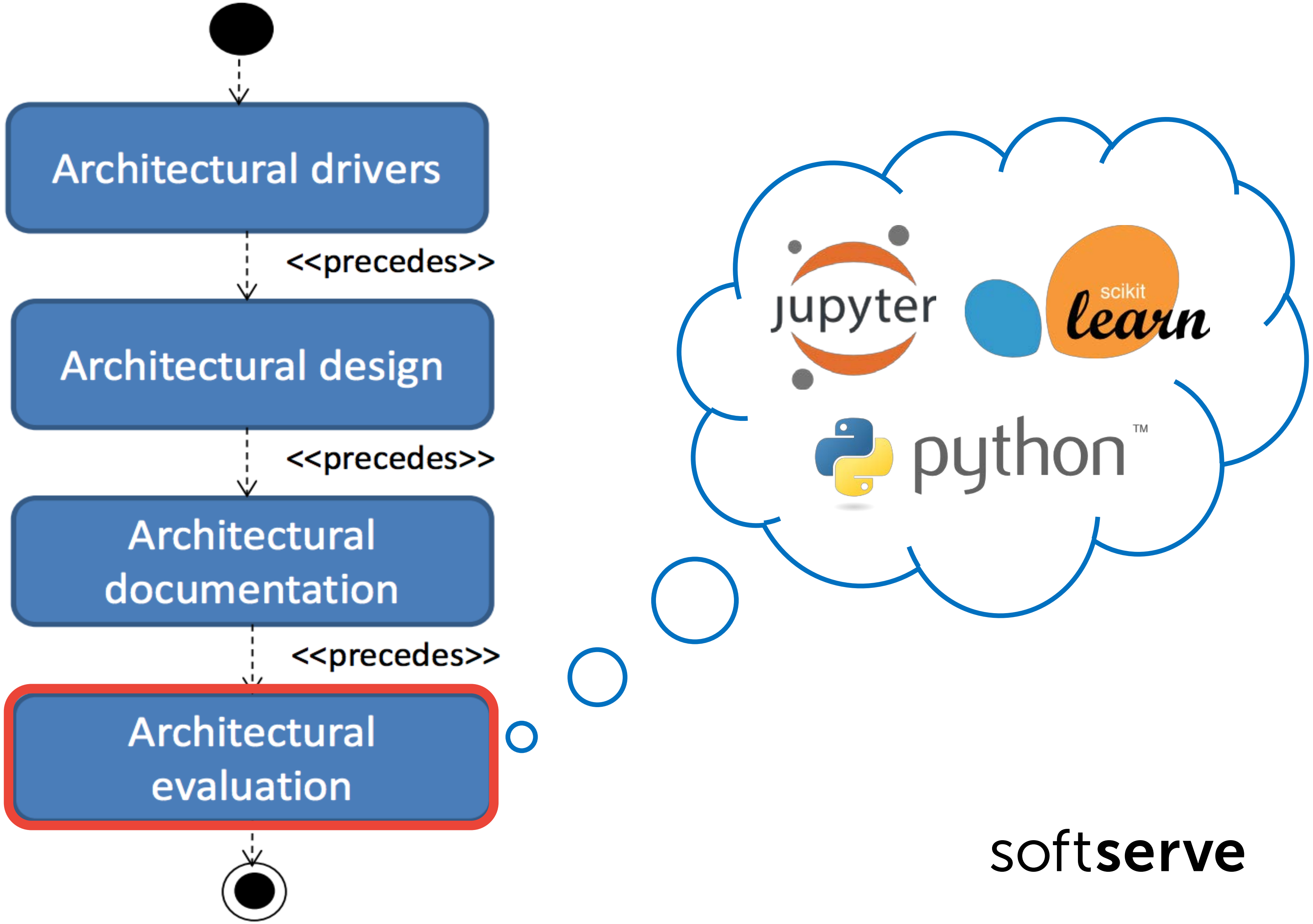
DESIGN PROCESS



Carnegie Mellon University
Software Engineering Institute

softserve

DESIGN PROCESS

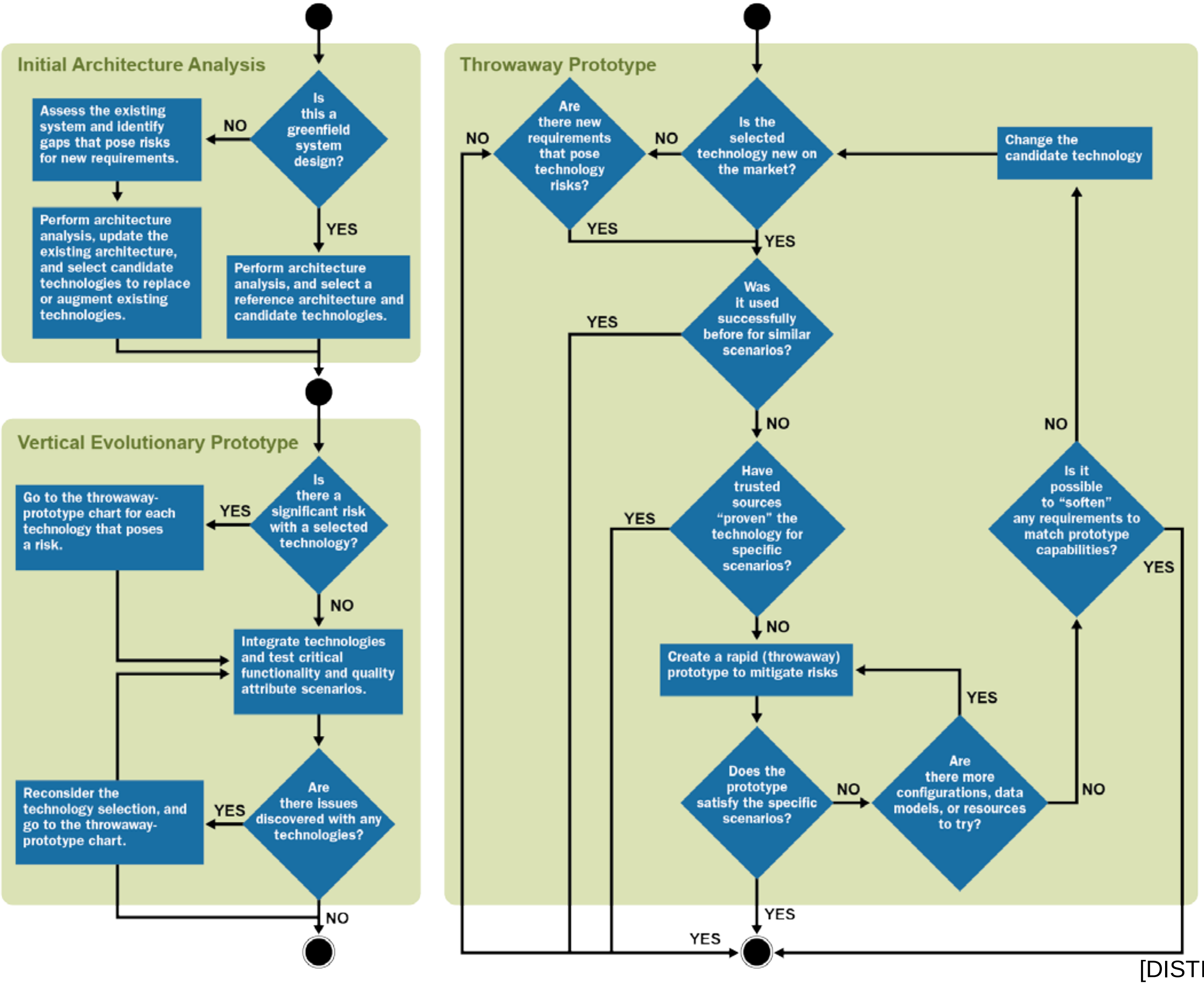


PROTOTYPING AND EVALUATION SESSION

Carnegie Mellon University
Software Engineering Institute

softserve

PROTOTYPING FOR EVALUATION



Carnegie Mellon University
Software Engineering Institute

softserve

RESULTS SUMMARY

Algorithm name	Training Time	Prediction Time	Tuning Time	Initial Accuracy	Final Accuracy
Random Forest	2.61	0.47	94.44	81.61%	**83.05%**
KNeighbors	0.41	**44.29**	84.27	80.57%	**83.05%**
Logistic Regression	0.12	0.05	**45.94**	**82.93%**	82.93%
MLP	0.80	0.08	164.04	**66.25%**	82.90%
SVM	**177.78**	54.87	**973.73**	82.83%	82.83%
Linear SVM	5.93	0.04	82.91	82.69%	82.69%
Decision Trees	0.03	**0.005**	52.97	73.16%	82.36%
Naive Bayes	**0.02**	0.01	0	78.46%	**78.46%**

Carnegie Mellon University
Software Engineering Institute

softserve

KEY TAKEAWAYS

- Machine Learning solution design is an iterative process
- ADD principles help make ML design decisions in a systematic way
- ML Cards aim to select candidate algorithms from a wide variety of alternatives
- Prototyping is necessary to validate design decisions

QUESTIONS? WE'VE GOT THE ANSWERS.

Carnegie Mellon University
Software Engineering Institute

softserve