AI INTRODUCTION

- AI trend in business and strategy
- AI day-to-day Development
- Machine Learning
- Deep Learning

ABOUT ME

- Past Projects will be showed on the seminar day.

Please call me Nam

namth4@topica.edu.vn

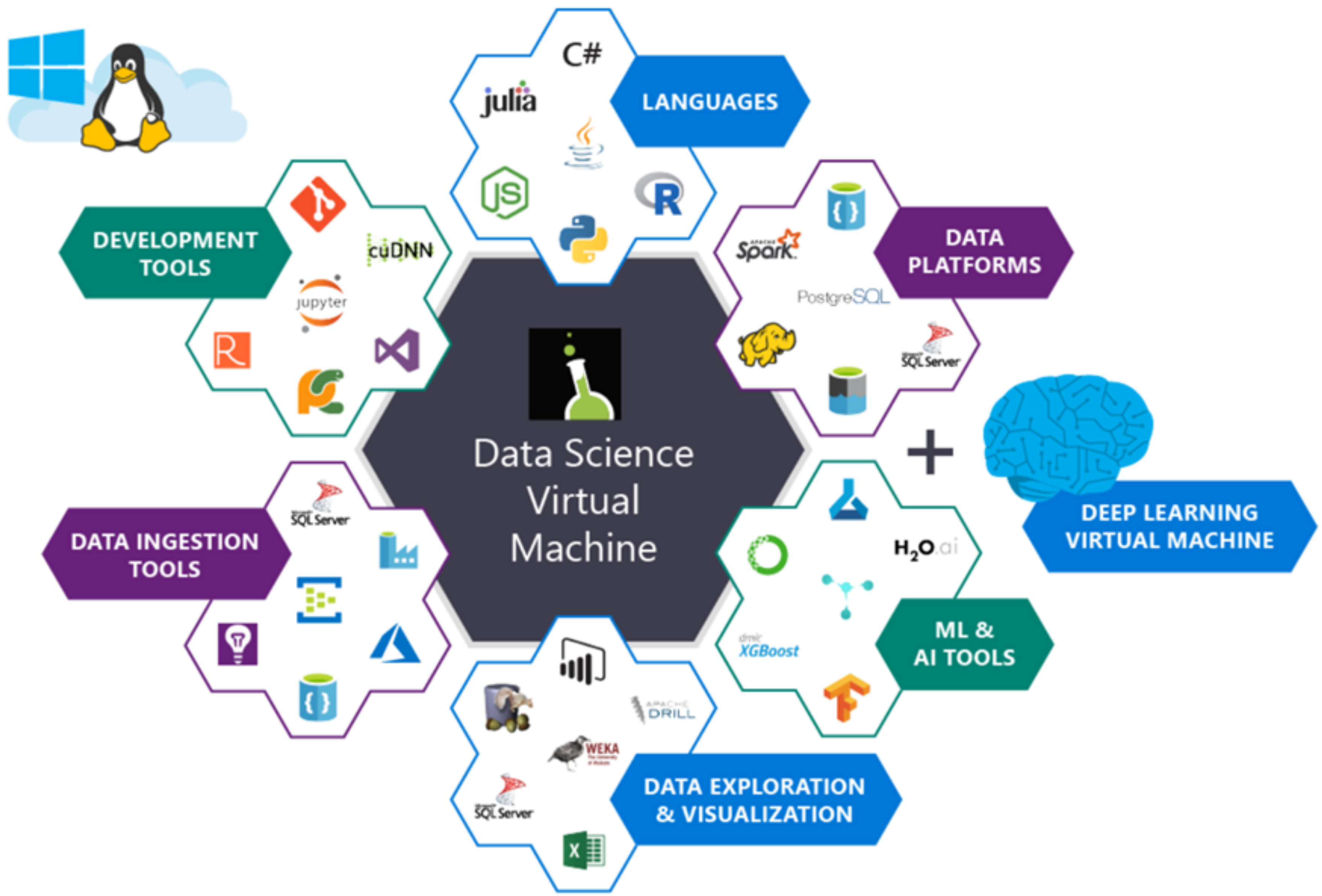
AI IN BUSINESS

- Because the content of this section is strictly confidential, so please come to the seminar, I can share with you in confidence.

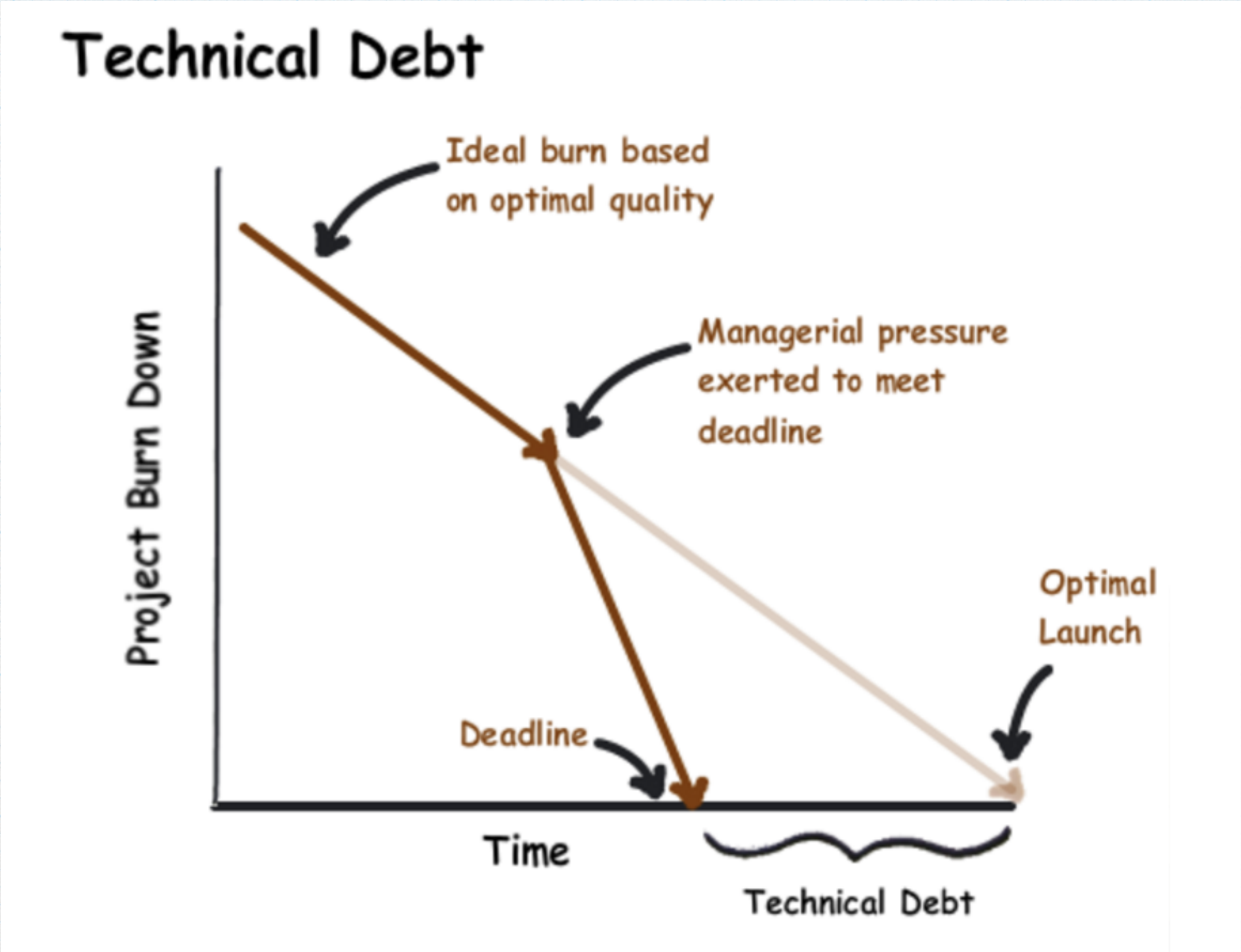
AI DEVELOPMENT

Pre-Configured environments in the cloud for Data Science and AI Development

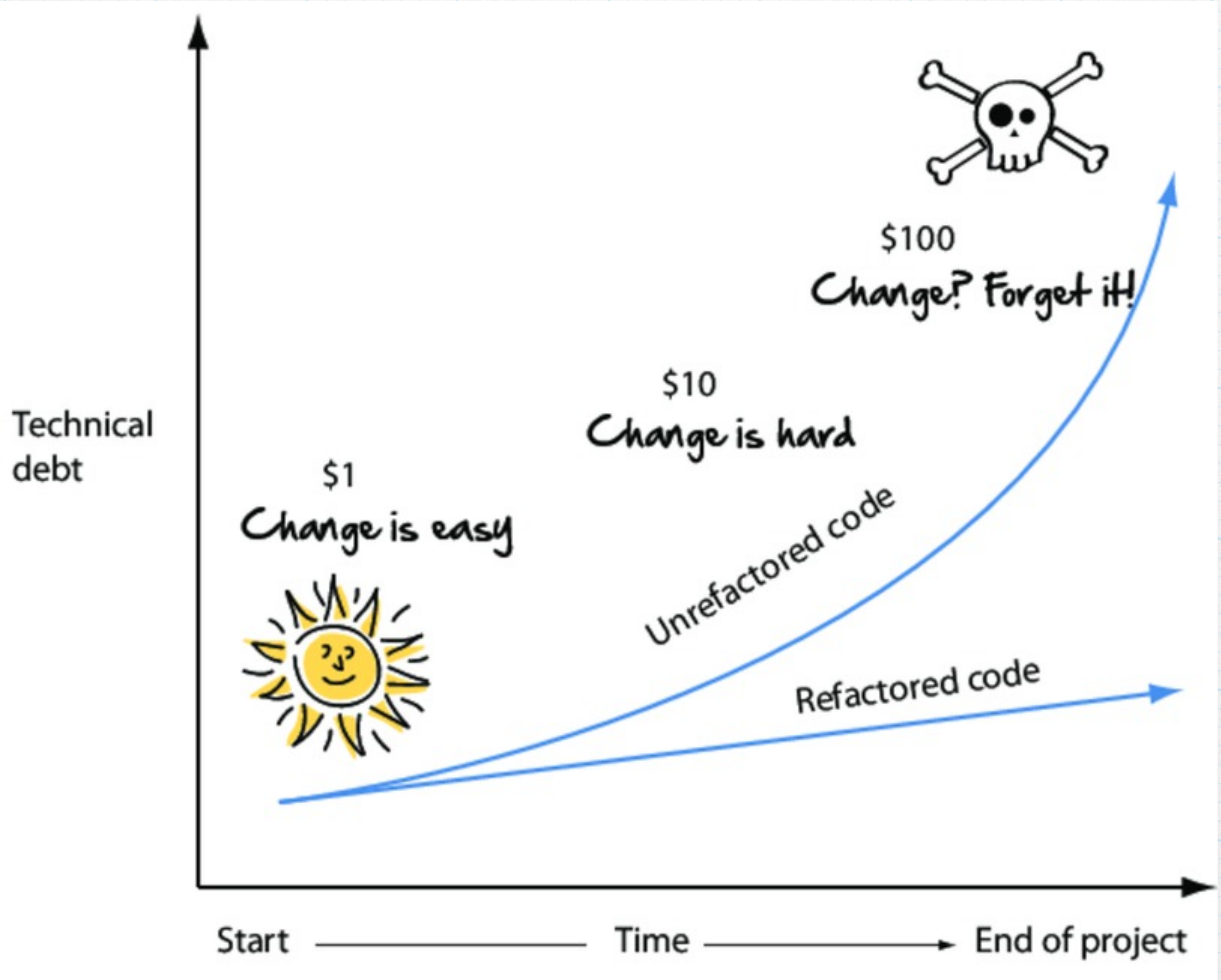
DSVMs are Azure Virtual Machine images, pre-installed, configured and tested with several popular tools that are commonly used for data analytics, machine learning and AI training.



Technical Debt

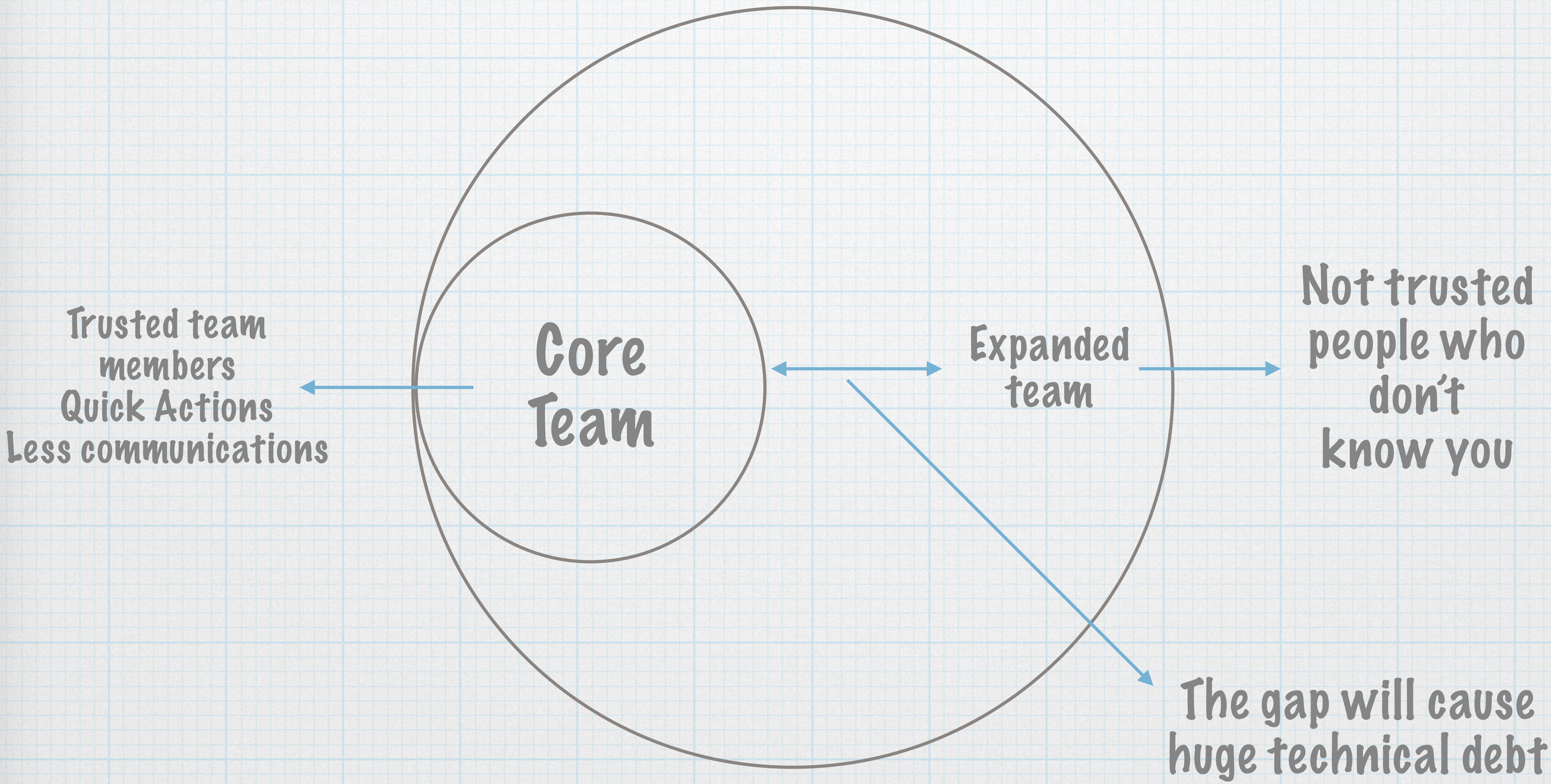


Technical Debt

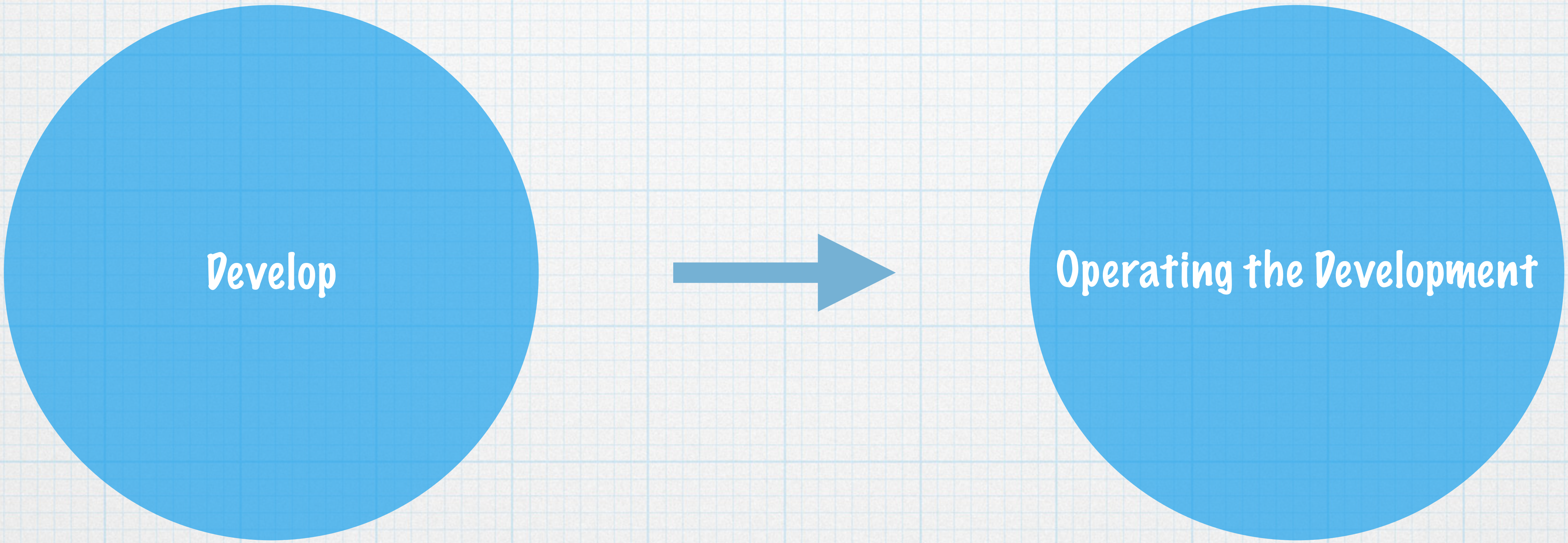


Engineer team expansion

Expansion of the team even makes the technical debt bigger



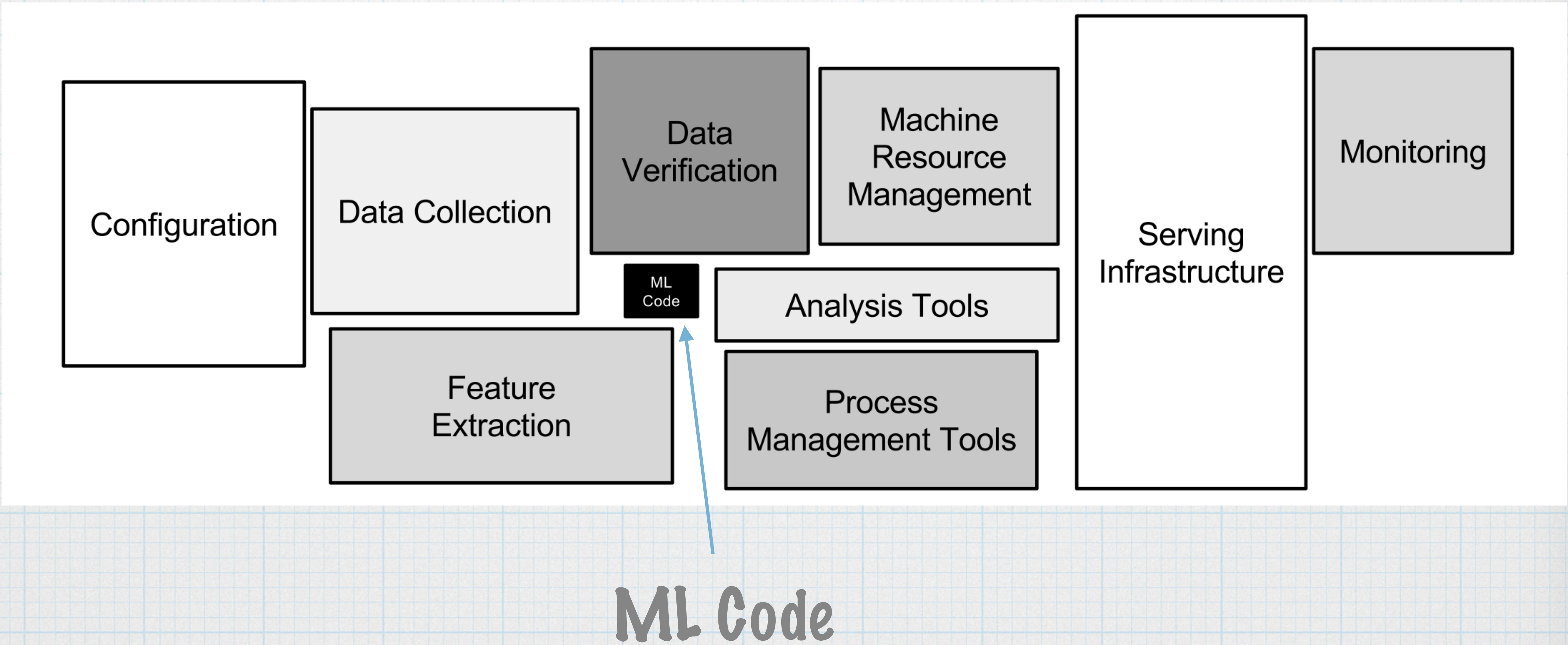
Mindset must change



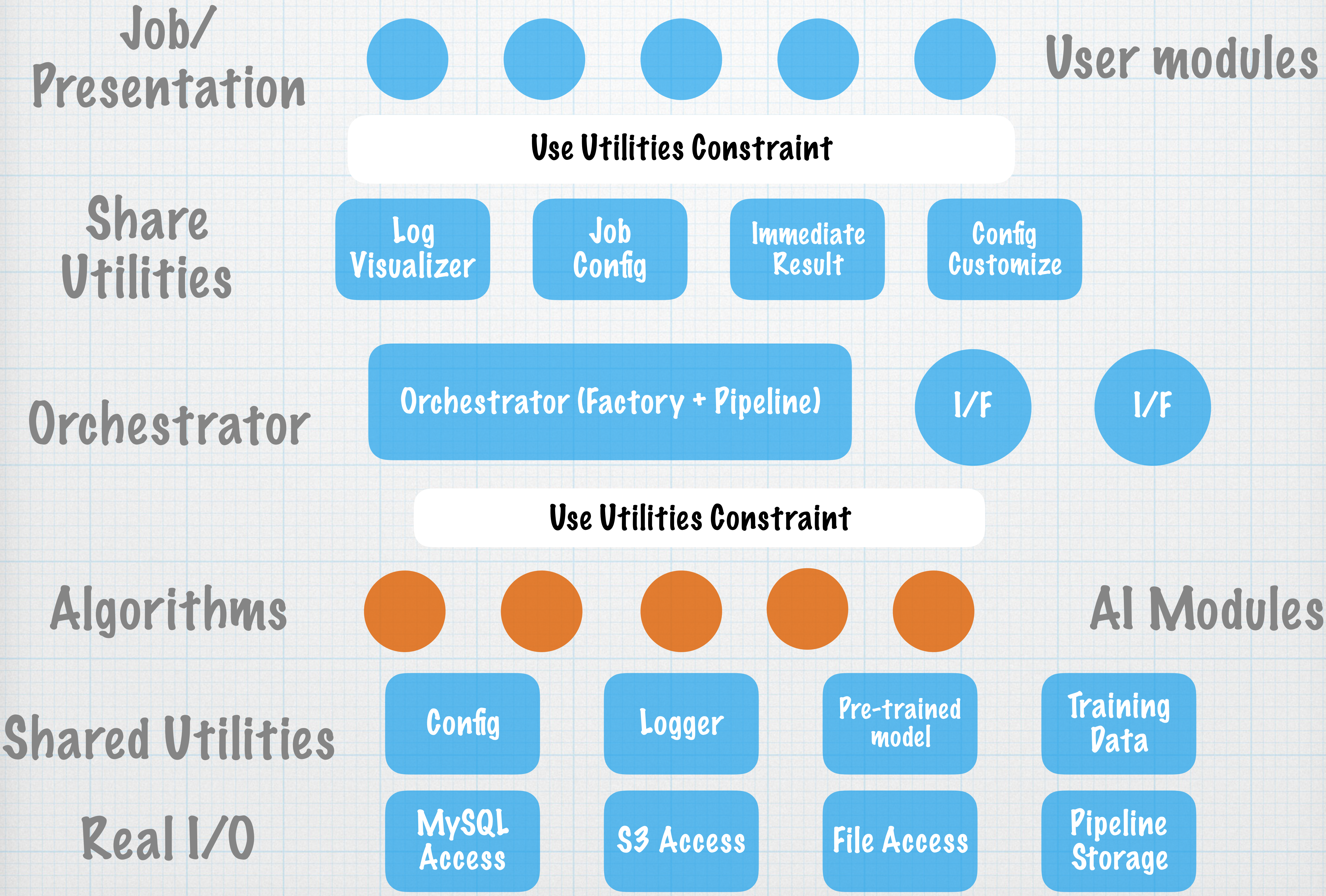
Before scaling the team
(Seed or Series A)

AI Technical Debt

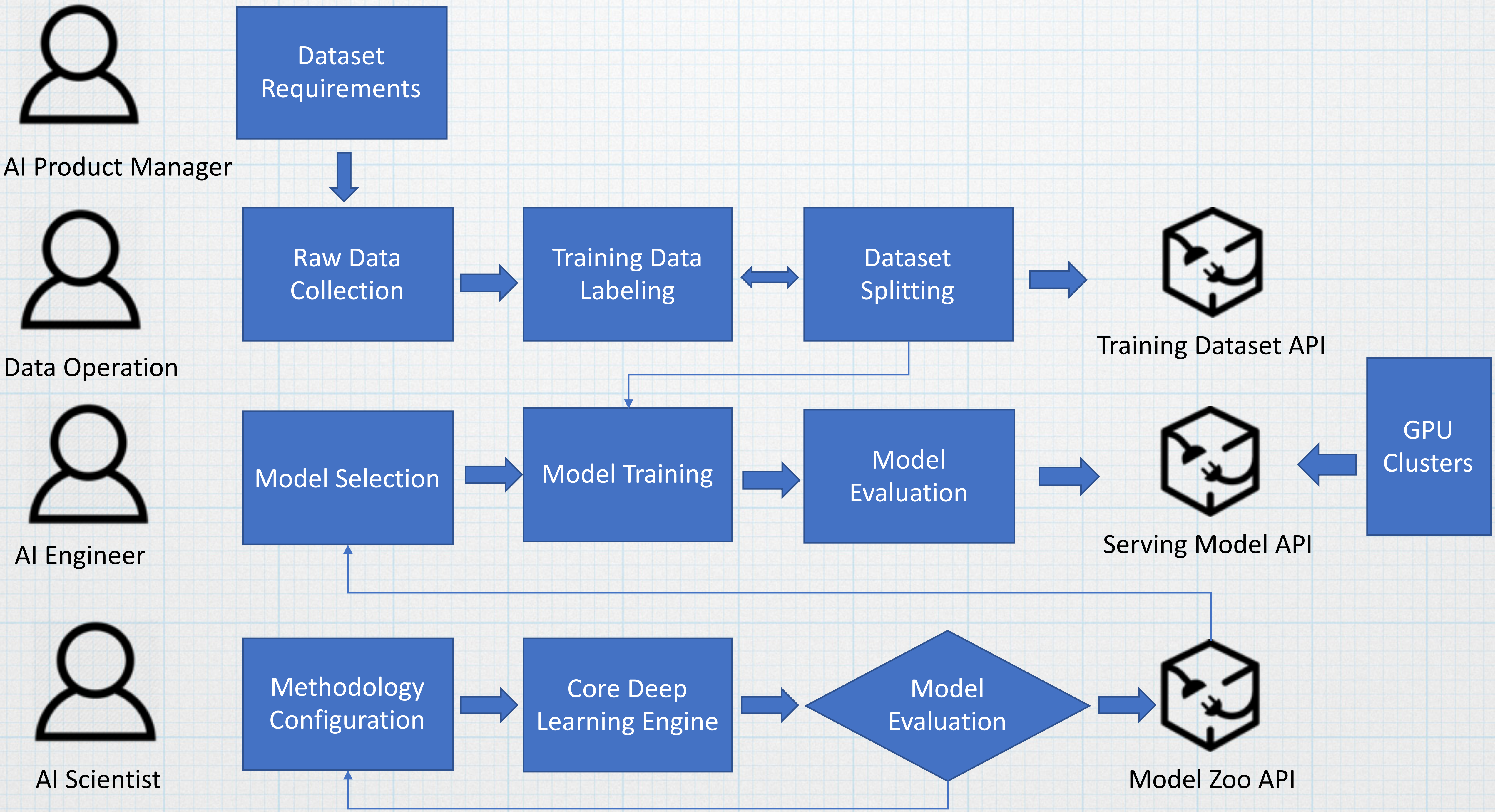
AI code usually 98% composed of other systems than Machine Learning and the module dependency will turn into hidden technical debt



Layered Architecture

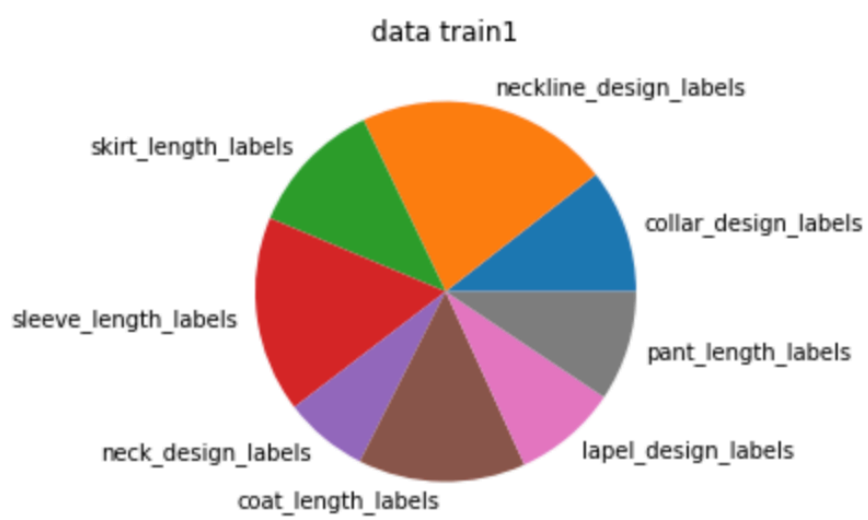


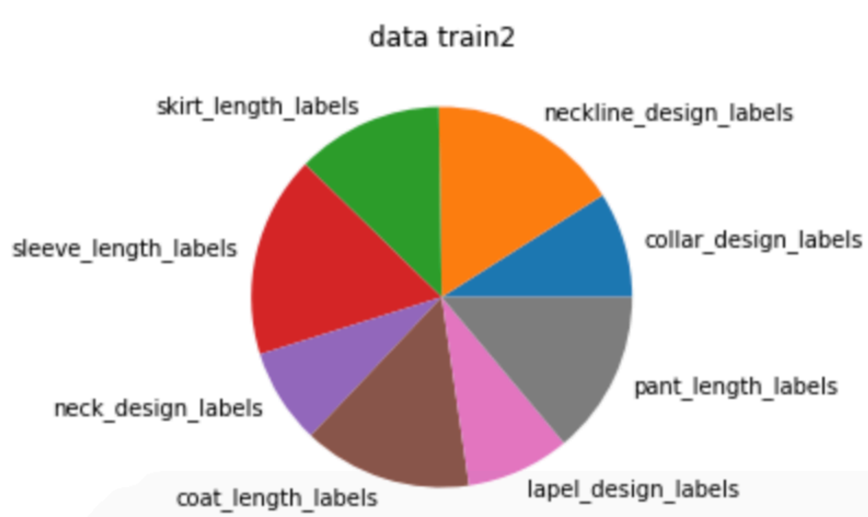
Roles in AI Development

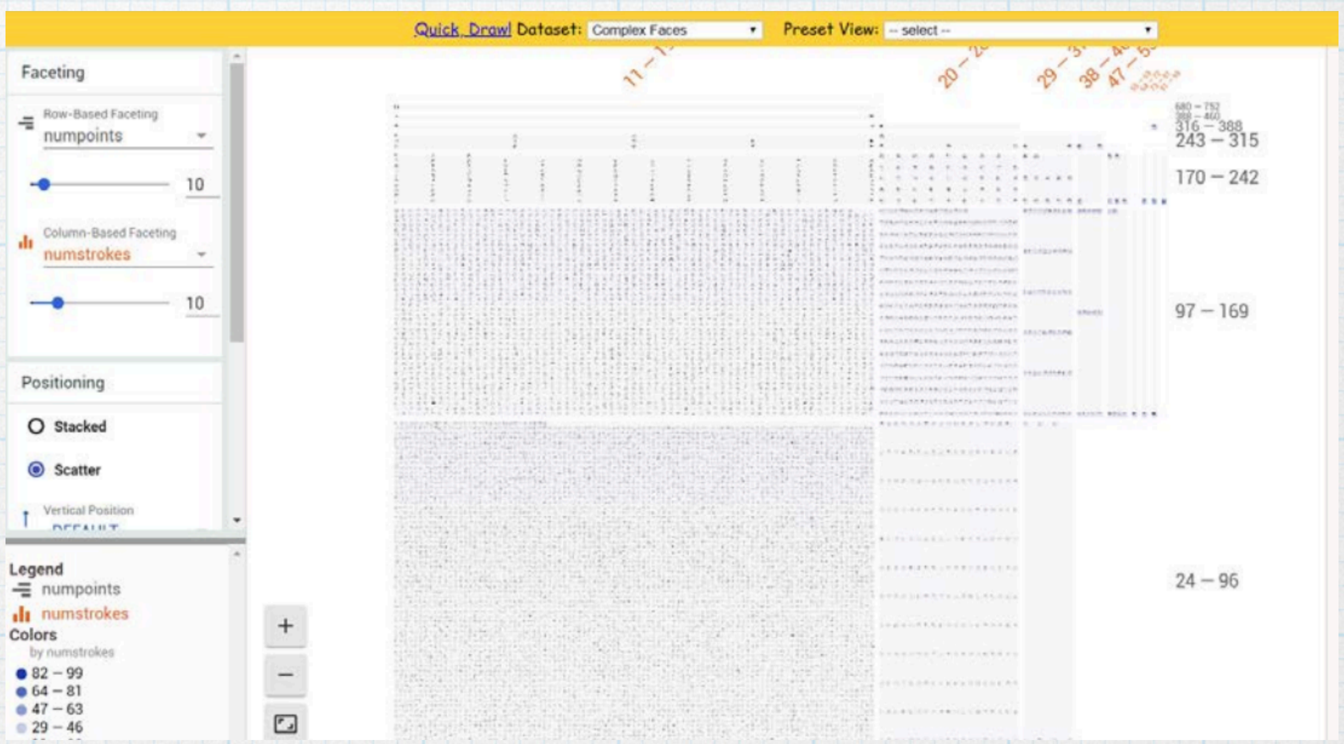


Data Analysis

```
'pant_length_labels': 14003,
'skirt_length_labels': 12555,
'collar_design_labels': 9058,
'lapel_design_labels': 8876,
'neck_design_labels': 8154})
```

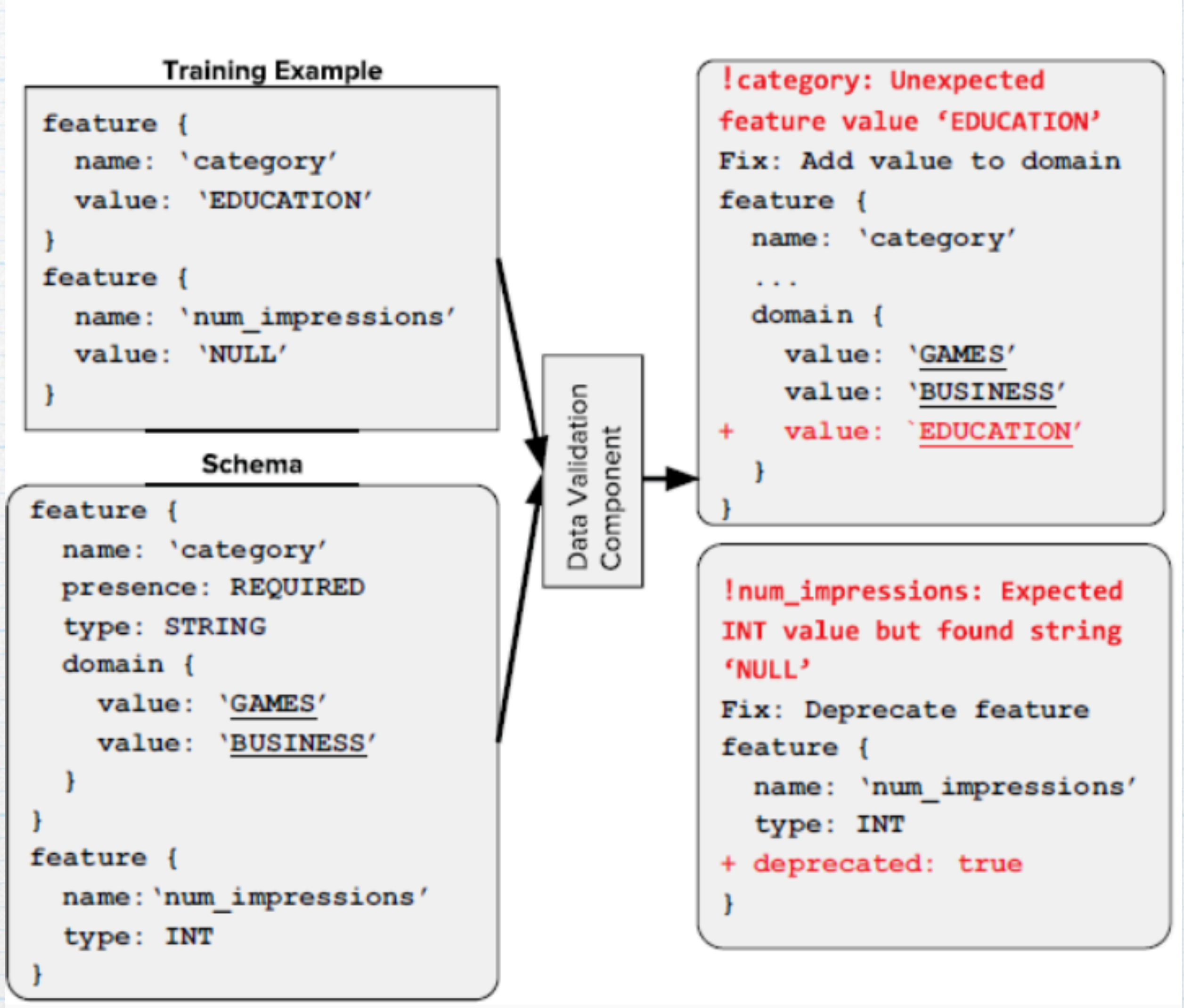






Data Validation

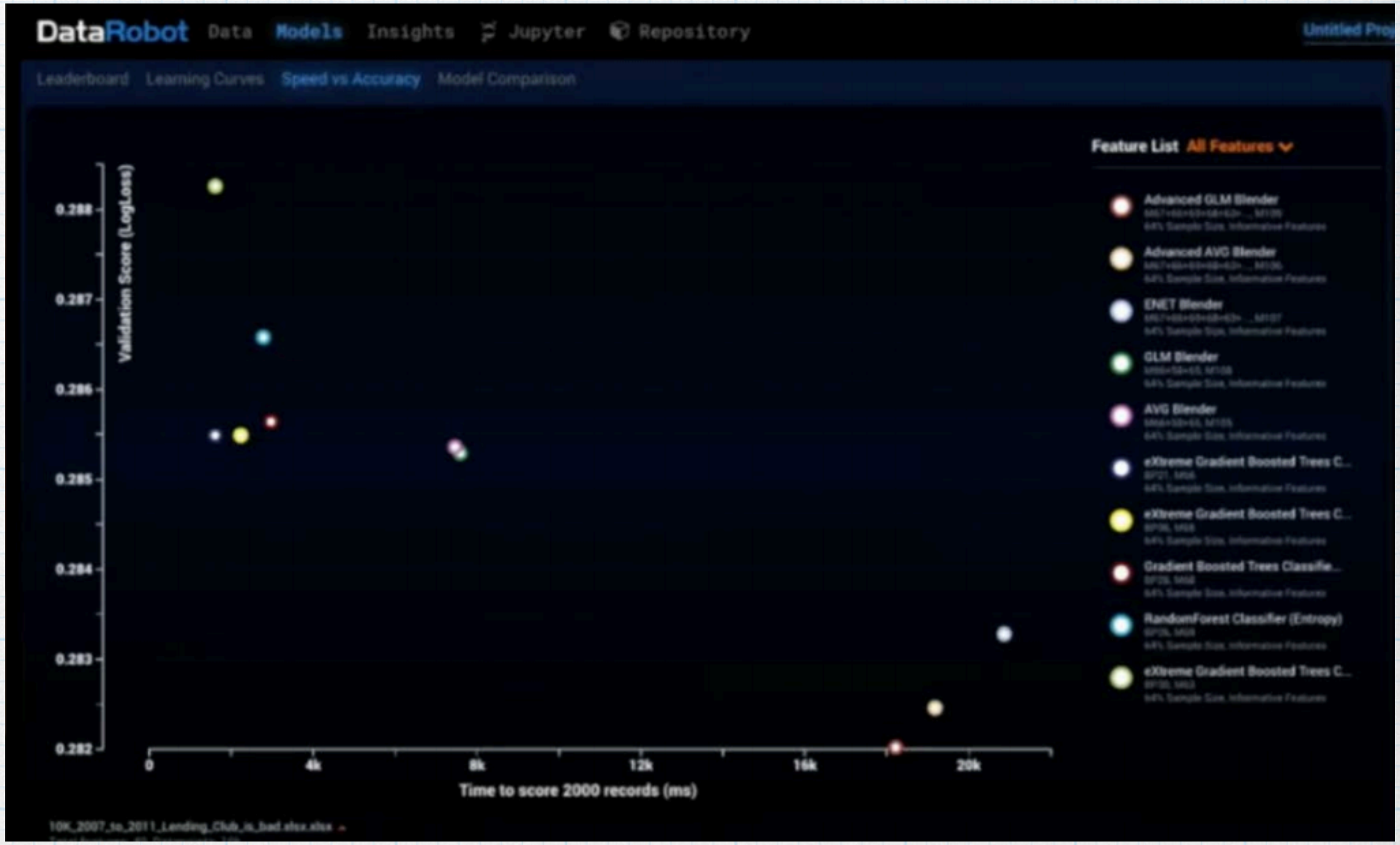
- Mis-coding issue
- Duplication , empty
- Missing of feature min/max value
- auto-recommendation/alerting
- etc



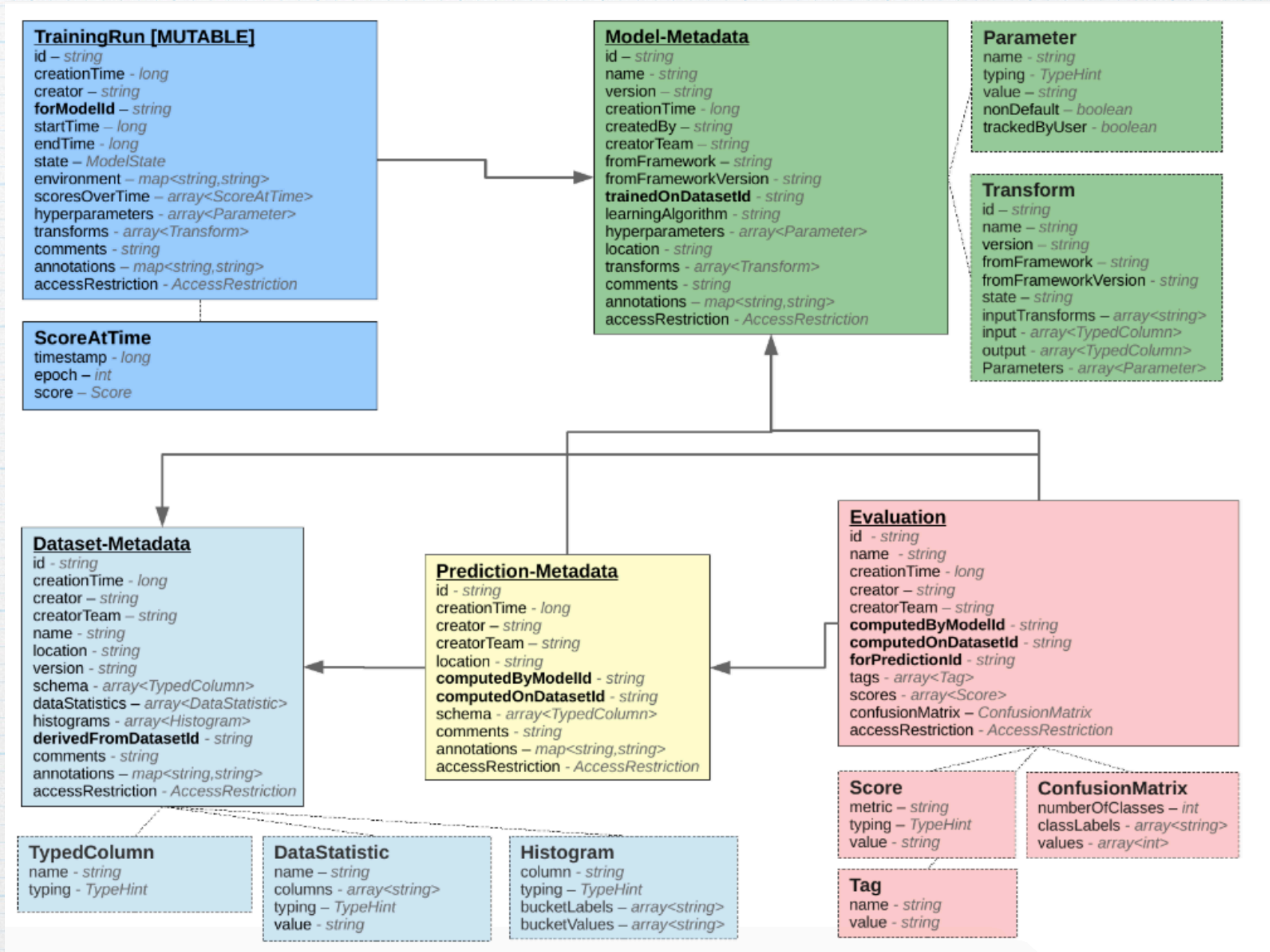
TFX: A TensorFlow-Based Production-Scale Machine Learning Platform D. Baylor etc KDD, 2017.

Model Validation - Safety

+) Taxonomy check
+) Latency check
+) Memory profiling check
+) Pre/post release monitoring and alerting



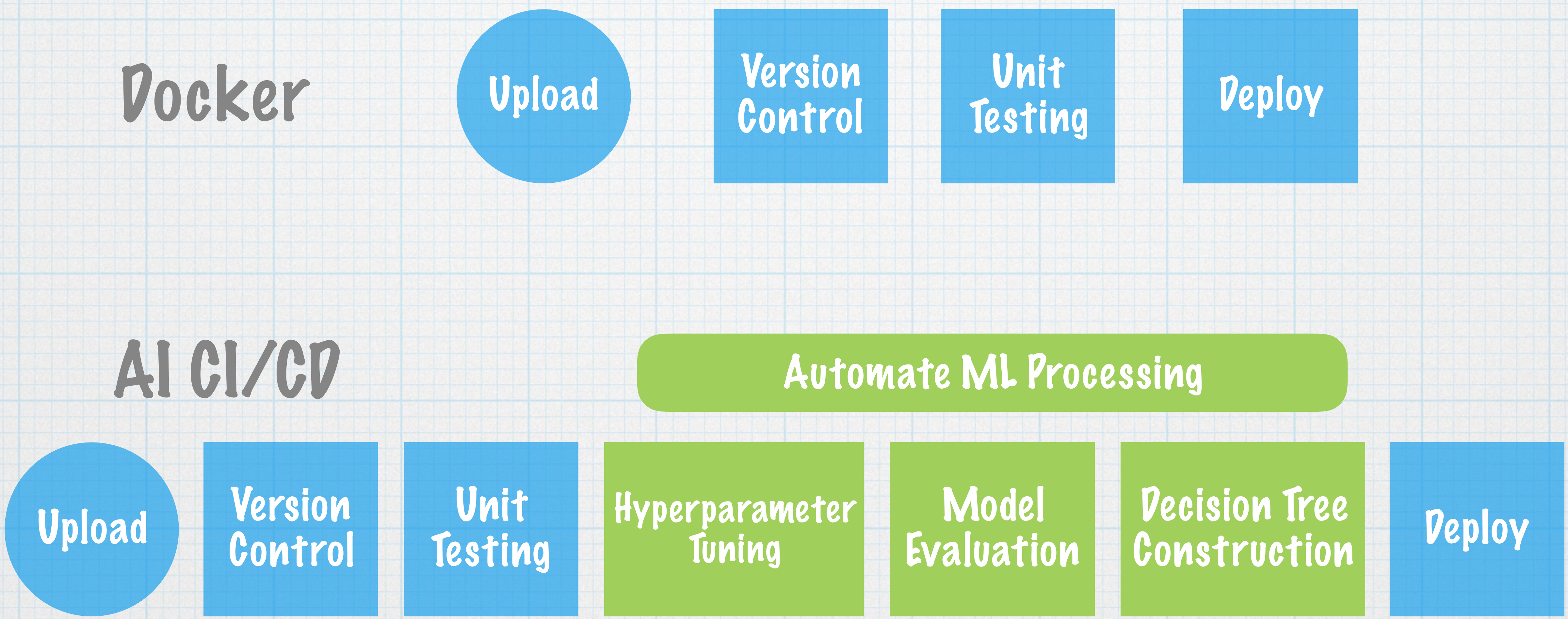
Model lifecycle Management



+) Model related metadata
+) Evaluation related metadata
+) Training Dataset metadata
+) Algorithm related metadata
+) Training task metadata
+) Training environment metadata

Schelter, Sebastian et al. "Automatically Tracking Metadata and Provenance of Machine Learning Experiments." (NIPS 2017).

Extend to automate the test



Example: Document Scanner Case

Image Processing

- Filetype conversion
- Binarization
- Classification
- Cropping Area

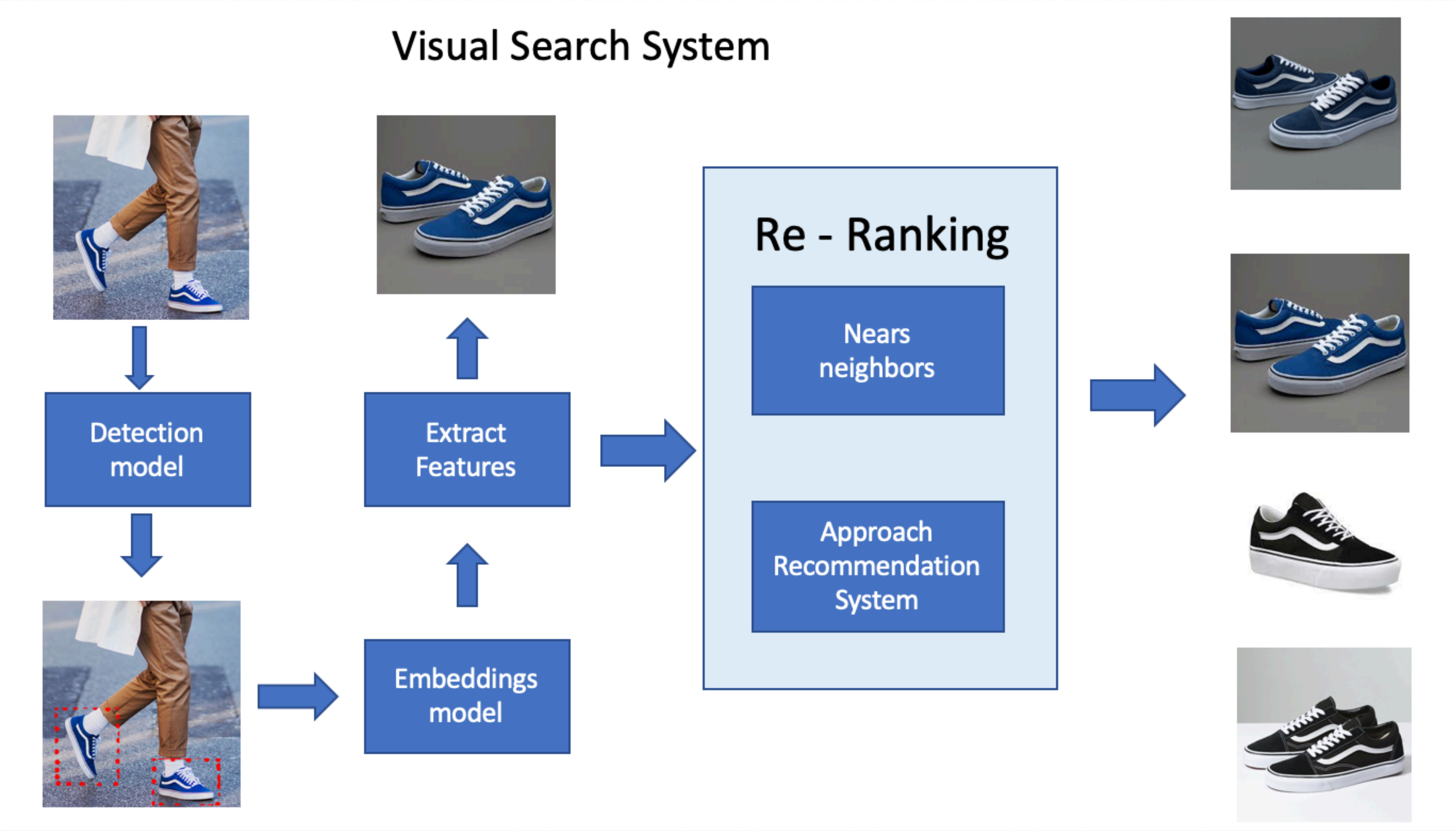
AI Modules

- Area Extraction
- Text Prediction
- Auto Correcting
- Text format validation

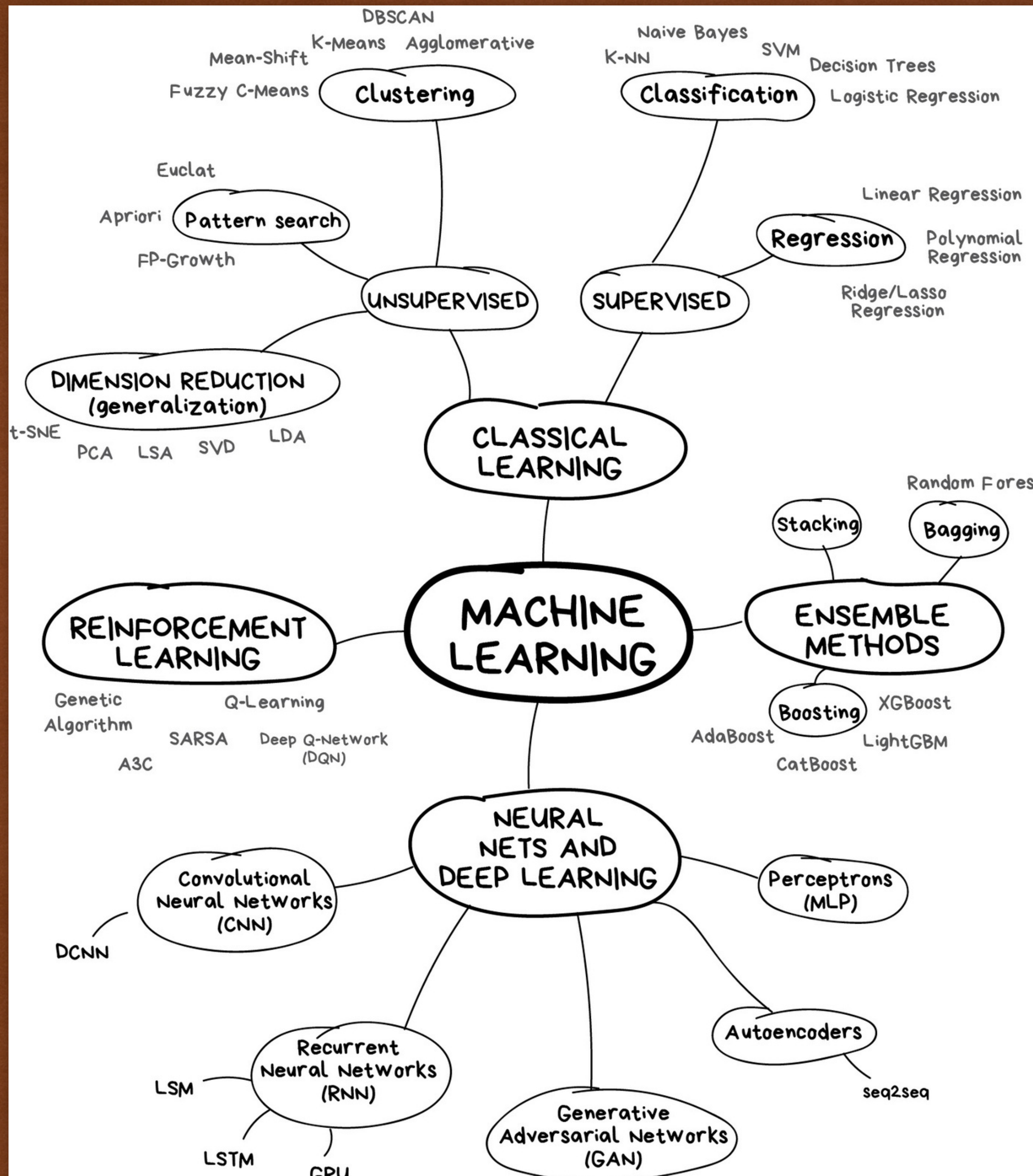
Front-end

- Anonymization
- Access level control
- Human Correction
- Other system integration

Example: MatarStars Visual Search System



MACHINE LEARNING



MACHINE LEARNING

1. Supervised Learning
    1.1 Linear Regression
        1.1.1 Simple Linear Regression
        1.1.2 Multiple Linear Regression
        1.1.3 Polynomial Regression
        1.1.4 Cost function
        1.1.5 Linear Regression and Cost Function code along
    1.2 Classification
        1.2.1 Logistic Regression
        1.2.2 K Nearest Neighbors
        1.2.3 SVM
        1.2.4 Naive Bayes
        1.3.5 Classification code along
2. Unsupervised Learning
3. Semi Supervised Learning
4. Reinforcement Learning
5. Overfitting
6. Cross Validation

MACHINE LEARNING

1. Supervised Learning
    1.1 Linear Regression
        1.1.1 Simple Linear Regression
        1.1.2 Multiple Linear Regression
        1.1.3 Polynomial Regression
        1.1.4 Cost function
        1.1.5 Linear Regression and Cost Function code along
    1.2 Classification
        1.2.1 Logistic Regression
        1.2.2 K Nearest Neighbors
        1.2.3 SVM
        1.2.4 Naive Bayes
        1.3.5 Classification code along
2. Unsupervised Learning
3. Semi Supervised Learning
4. Reinforcement Learning
5. Overfitting
6. Cross Validation

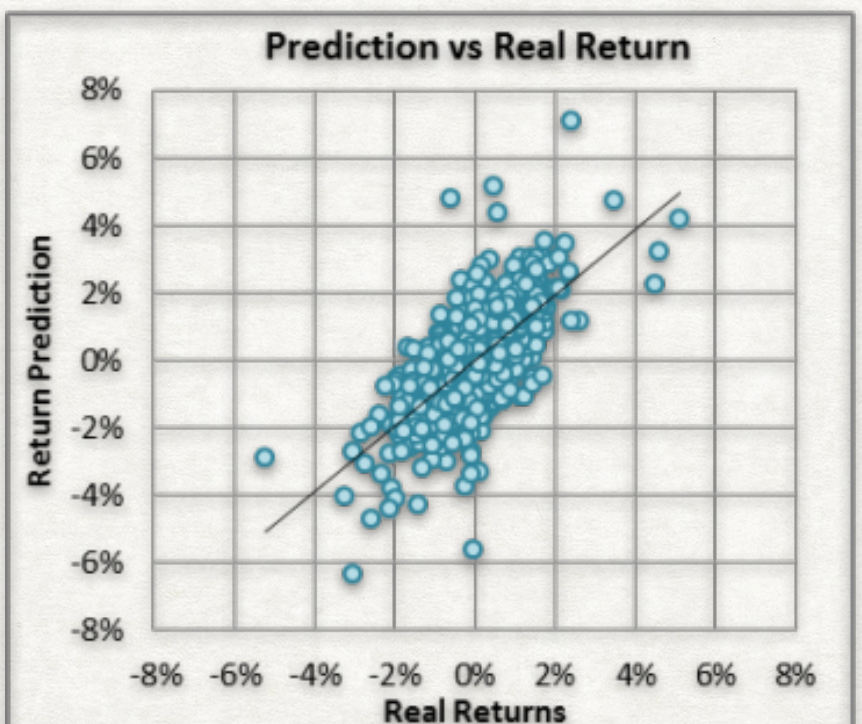
1. SUPERVISED LEARNING

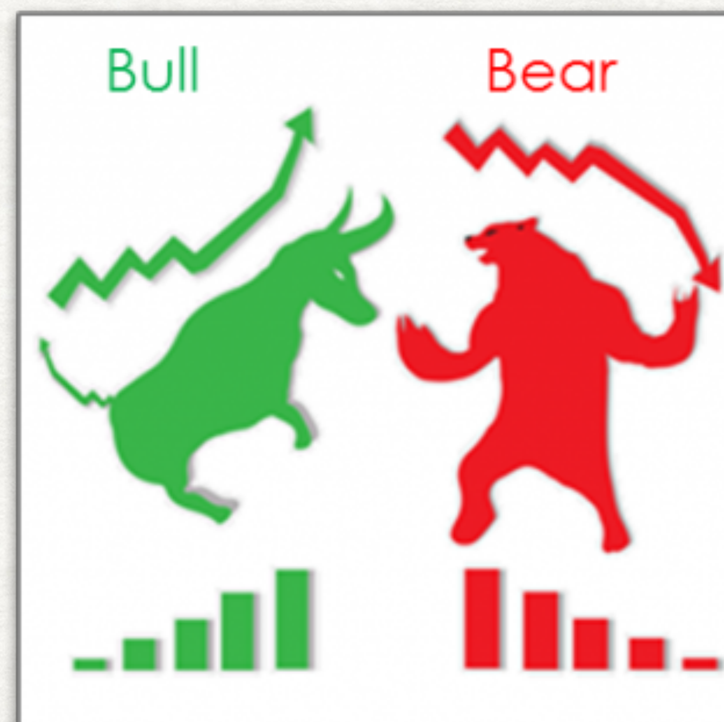
If you're learning a task under supervision, someone is present judging whether you're getting the right answer. Similarly, in supervised learning, that means having a full set of labeled data while training an algorithm.

Classification example: Bull or Bear

Regression example: An algorithm that predicts the price of an apartment in San Francisco based on square footage, location and proximity to public transport.

Regression **vs** **Classification**





1.1 LINEAR REGRESSION

Linear regression is probably one of the most important and widely used regression techniques. It's among the simplest regression methods. One of its main advantages is the ease of interpreting results.

Typically, you need regression to answer whether and how some phenomenon influences the other or **how several variables are related**. For example, you can use it to determine *if*and *to what extent* the experience or gender impact salaries.

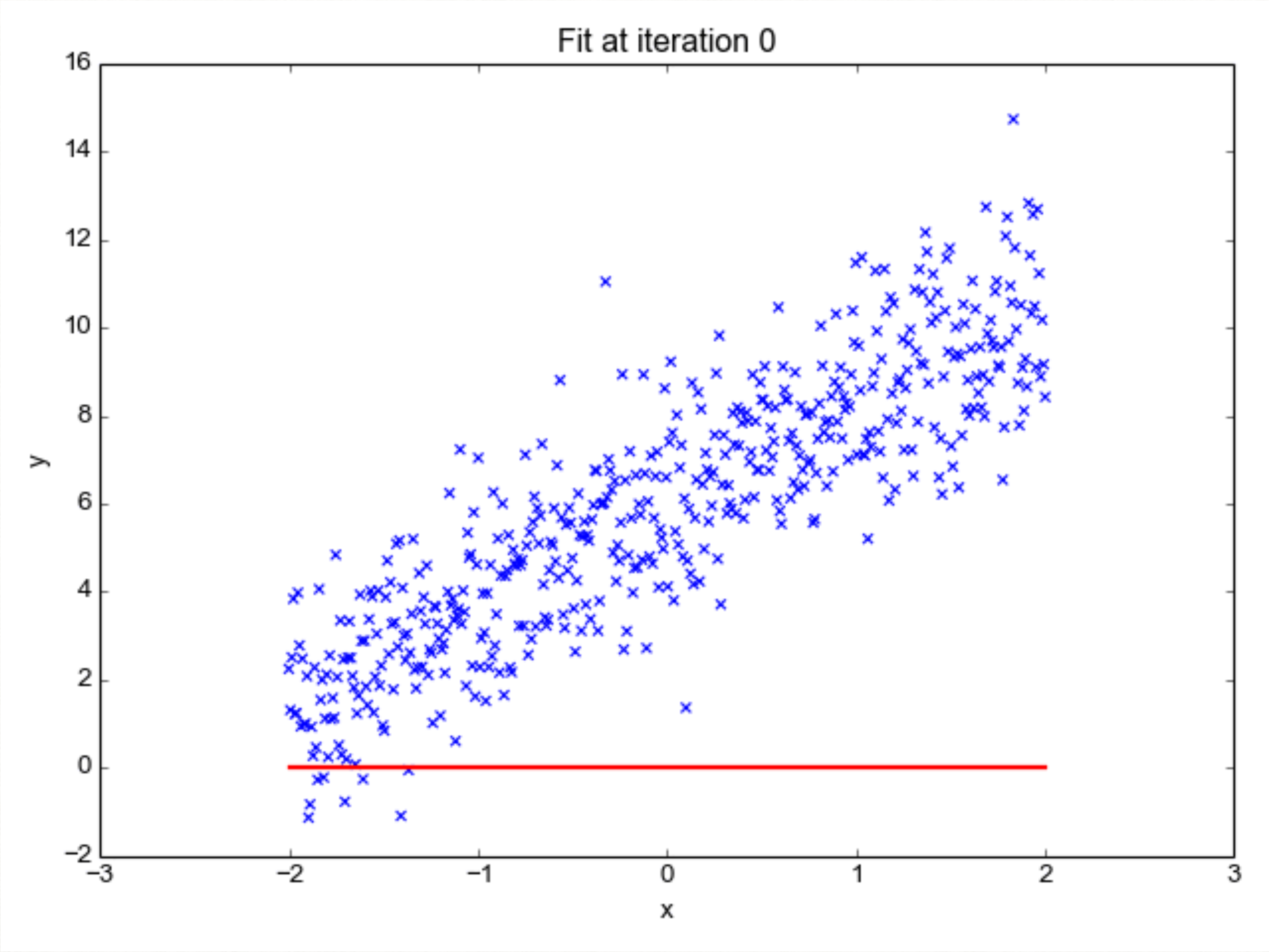
Regression is also useful when you want **to forecast a response** using a new set of predictors. For example, you could try to predict electricity consumption of a household for the next hour given the outdoor temperature, time of day, and number of residents in that household.

Regression is used in many different fields: economy, computer science, social sciences, and so on. Its importance rises every day with the availability of large amounts of data and increased awareness of the practical value of data.

1.1.1 SIMPLE LINEAR REGRESSION

Simple or single-variate linear regression is the simplest case of linear regression with a single independent variable, $\mathbf{x} = x$.

The following figure illustrates simple linear regression:

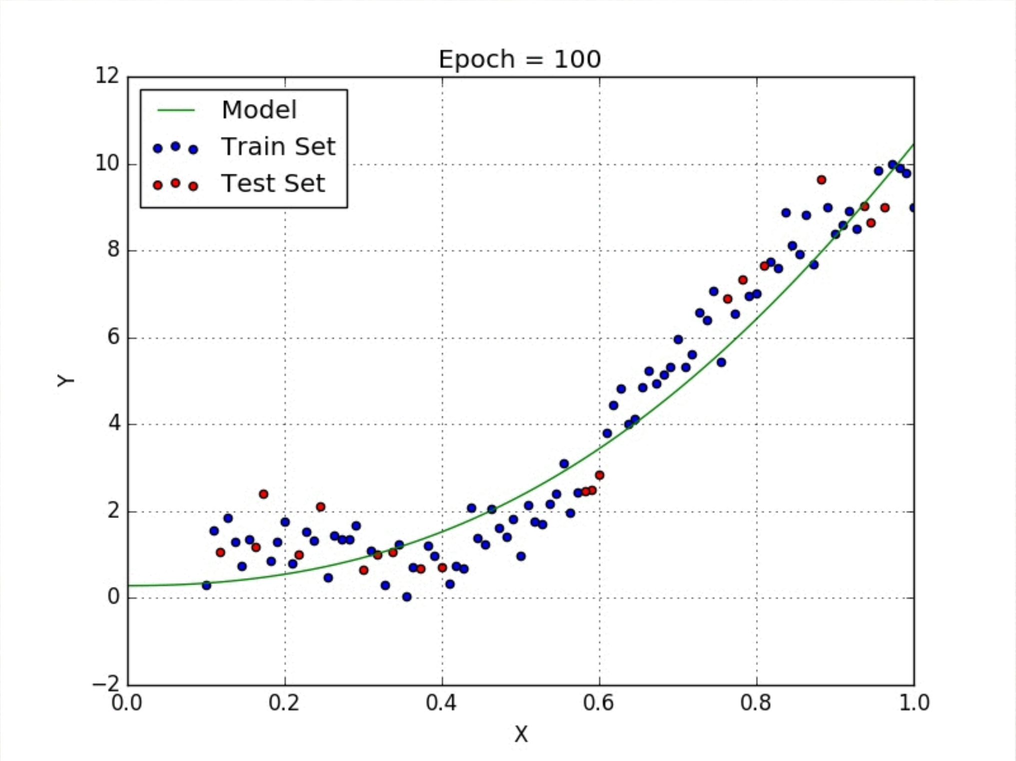


1.1.2 MULTIPLE LINEAR REGRESSION

Multiple or multivariate linear regression is a case of linear regression with two or more independent variables.

If there are just two independent variables, the estimated regression function is $f(x_1, x_2) = b_0 + b_1x_1 + b_2x_2$. It represents a regression plane in a three-dimensional space. The goal of regression is to determine the values of the weights b_0, b_1, and b_2 such that this plane is as close as possible to the actual responses and yield the minimal SSR.

The case of more than two independent variables is similar, but more general. The estimated regression function is $f(x_1, \ldots, x_r) = b_0 + b_1x_1 + \cdots + b_rx_r$, and there are $r + 1$ weights to be determined when the number of inputs is r.

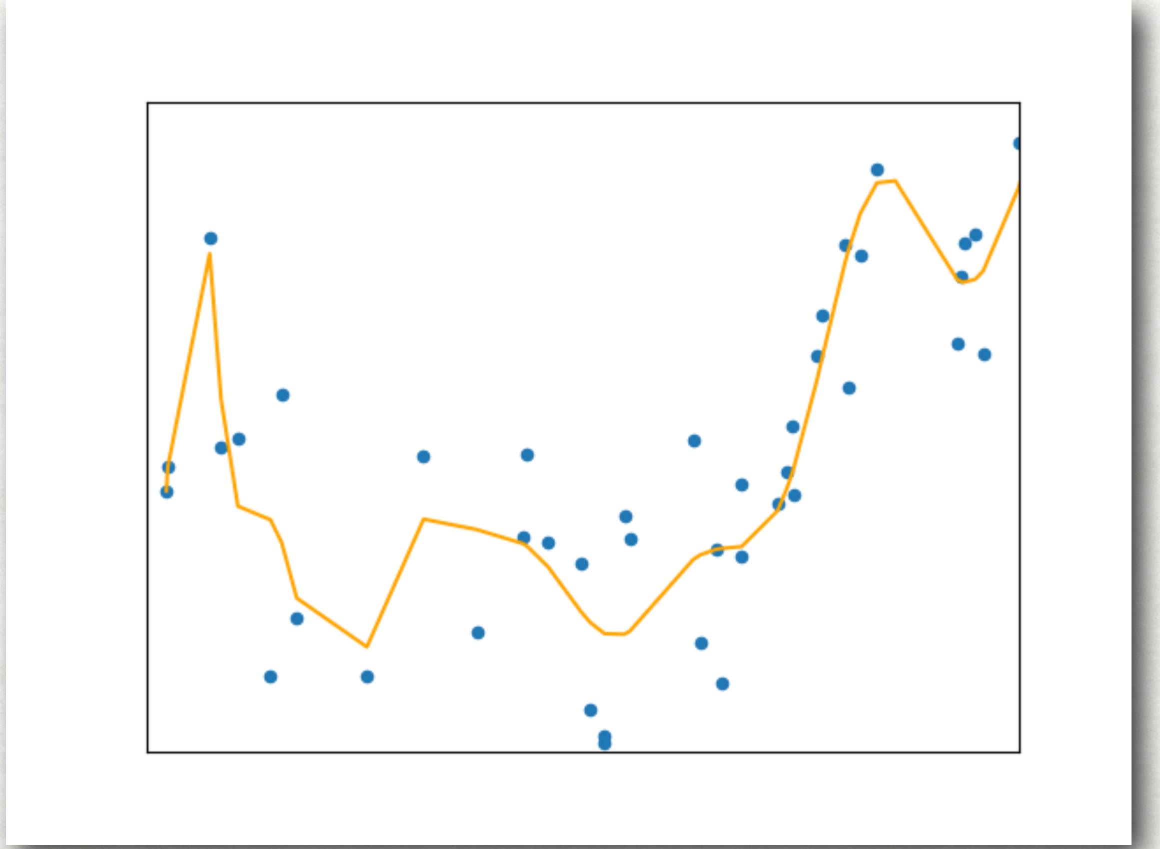


1.1.3 POLYNOMIAL REGRESSION

You can regard polynomial regression as a generalized case of linear regression. You assume the polynomial dependence between the output and inputs and, consequently, the polynomial estimated regression function.

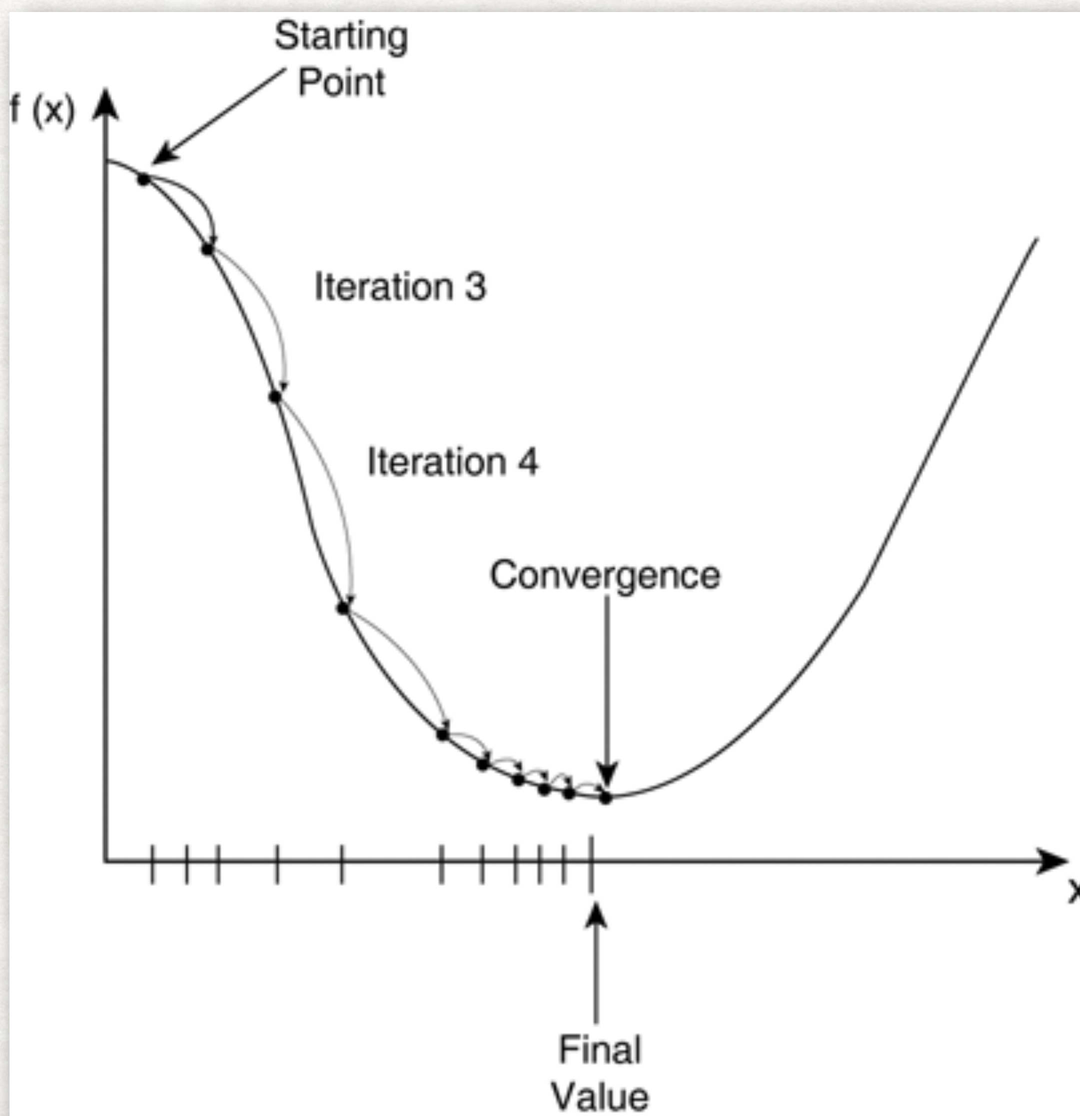
In other words, in addition to linear terms like b_1x_1, your regression function f can include non-linear terms such as $b_2x_1^2$, $b_3x_1^3$, or even $b_4x_1x_2$, $b_5x_1^2x_2$, and so on.

The simplest example of polynomial regression has a single independent variable, and the estimated regression function is a polynomial of degree 2: $f(x) = b_0 + b_1x + b_2x^2$.



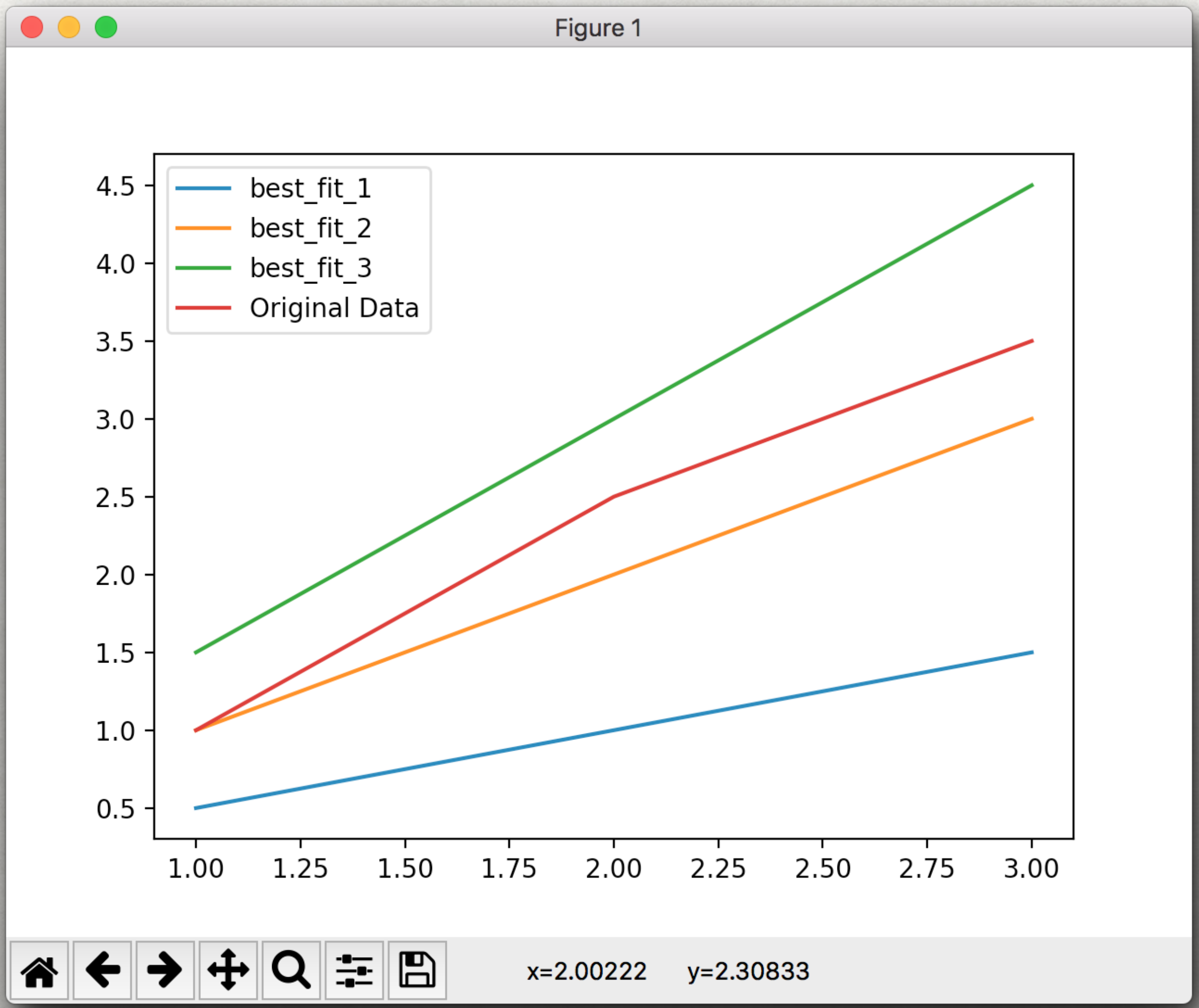
1.1.4 COST FUNCTION

Cost function is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event.



1.1.5. COST FUNCTION

X	y	best_fit_1	best_fit_2	best_fit_3
1.00	1.00	0.50	1.00	1.50
2.00	2.50	1.00	2.00	3.00
3.00	3.50	1.50	3.00	4.00



1.1.5 COST FUNCTION

Remember a cost function *maps event or values of one or more variables onto a real number*. In this case, the event we are finding the cost of is the **difference between estimated values**, or the difference between the *hypothesis* and the real values — the actual data we are trying to fit a line to.

$$\frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Squared Error cost function

1.1.5. COST FUNCTION

For X = 1. The hypothesis value is 0.50 , and the actual yvalue is 1.00 . So we get (0.50— 1.00)^2, which is 0.25, and so on, for X = 2 we get 2.25 and 4.00.

X	y	best_fit_1
1.00	1.00	0.50
2.00	2.50	1.00
3.00	3.50	1.50

$$\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Squared Error cost function

In conclusion we get: 1/6 * (0.25 + 2.25 + 4.00) = 1.083

1.1.5. COST FUNCTION

.

best_fit_1: 1.083

best_fit_2: 0.083

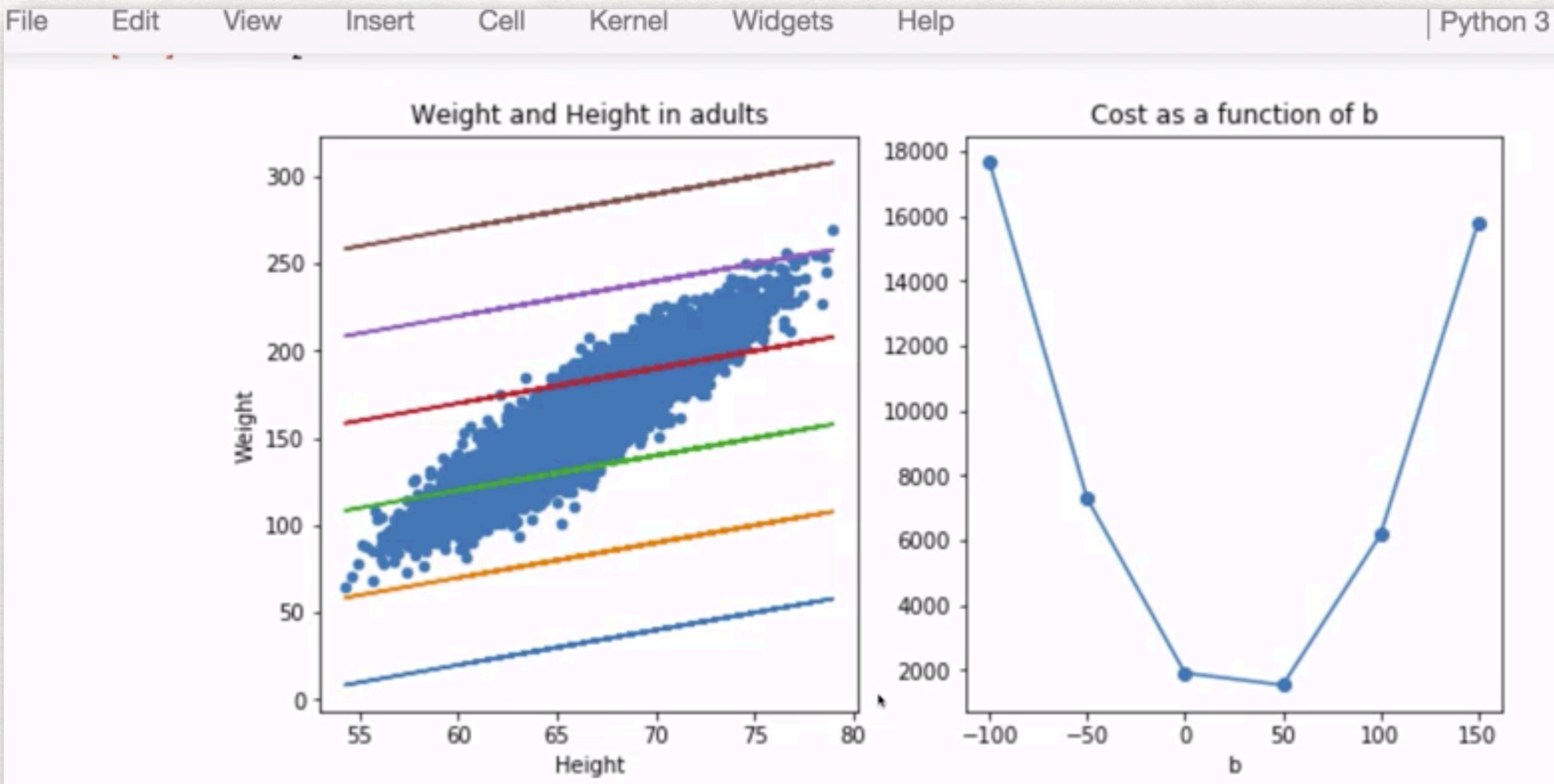
best_fit_3: 0.25

Best_fit_2 is the best model

X	y	best_fit_1	best_fit_2	best_fit_3
1.00	1.00	0.50	1.00	1.50
2.00	2.50	1.00	2.00	3.00
3.00	3.50	1.50	3.00	4.00

1.1.6 COST FUNCTION AND LINEAR REGRESSION CODE ALONG

Will be presented on the seminar day

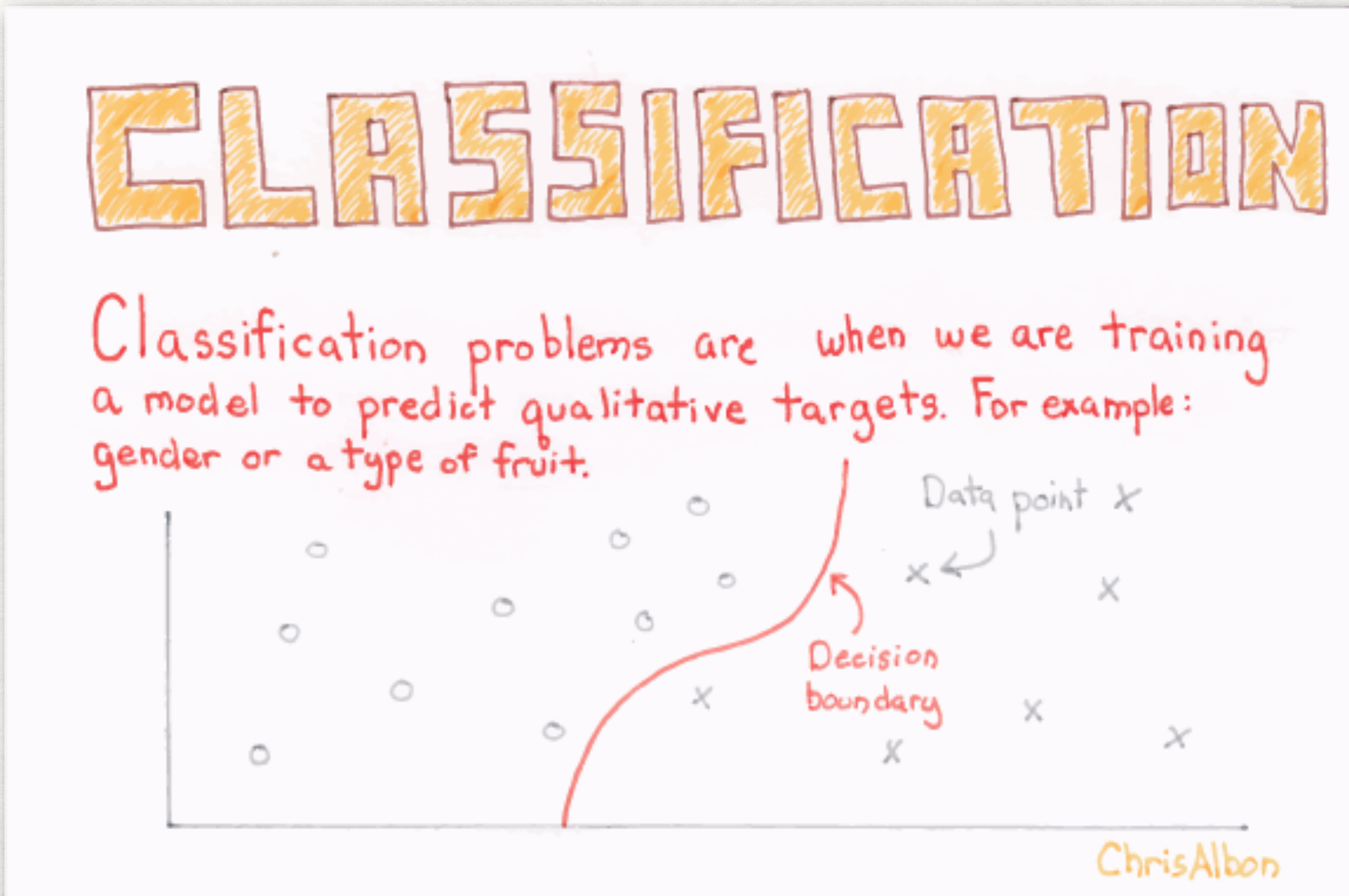


MACHINE LEARNING

1. Supervised Learning
   - **1.1 Linear Regression**
     - 1.1.1 Simple Linear Regression
     - 1.1.2 Multiple Linear Regression
     - 1.1.3 Polynomial Regression
     - 1.1.4 Cost function
     - 1.1.5 Linear Regression and Cost Function code along
   - 1.2 Classification
     - 1.2.1 Logistic Regression
     - 1.2.2 K Nearest Neighbors
     - 1.2.3 SVM
     - 1.2.4 Naive Bayes
     - 1.3.5 Classification code along
2. Unsupervised Learning
3. Semi Supervised Learning
4. Reinforcement Learning
5. Overfitting
6. Cross Validation

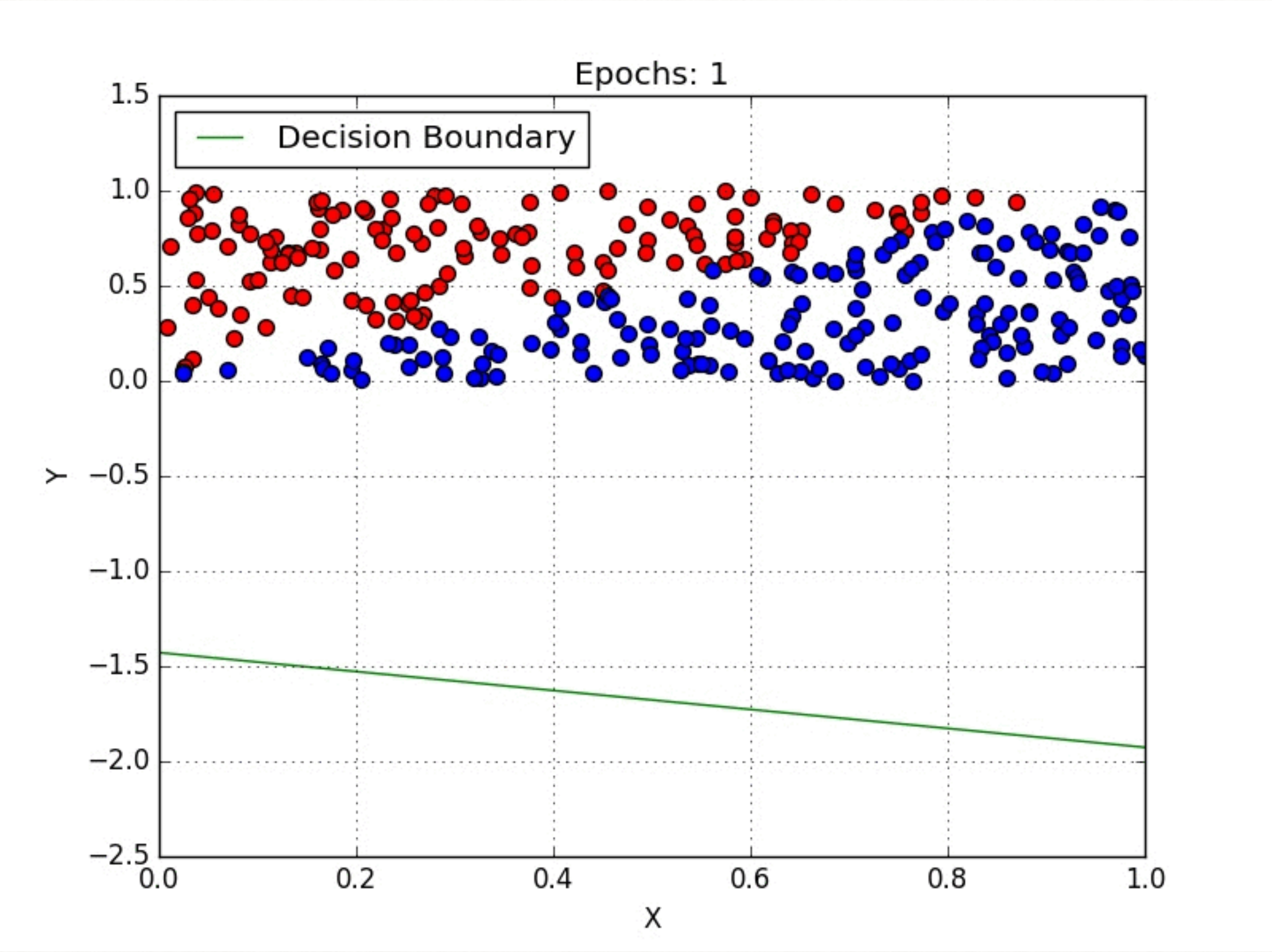
1.2 CLASSIFICATION

Classification is a technique for determining class the dependent belongs to based on the one or more independent variables



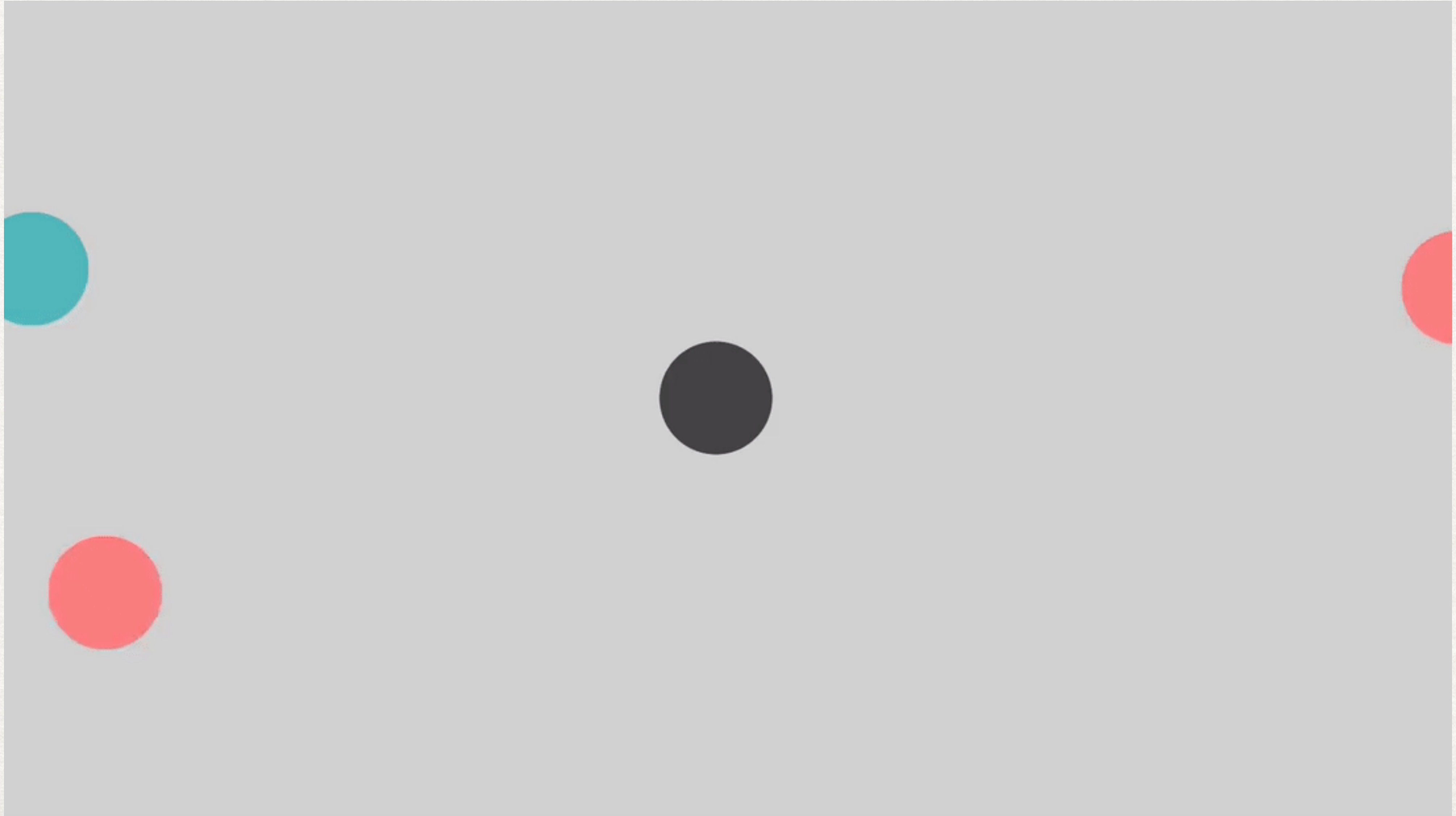
1.2.1 LOGISTIC REGRESSION

Logistic regression is kind of like linear regression but is used when the dependent variable is not a number, but something else (like a Yes/No response).



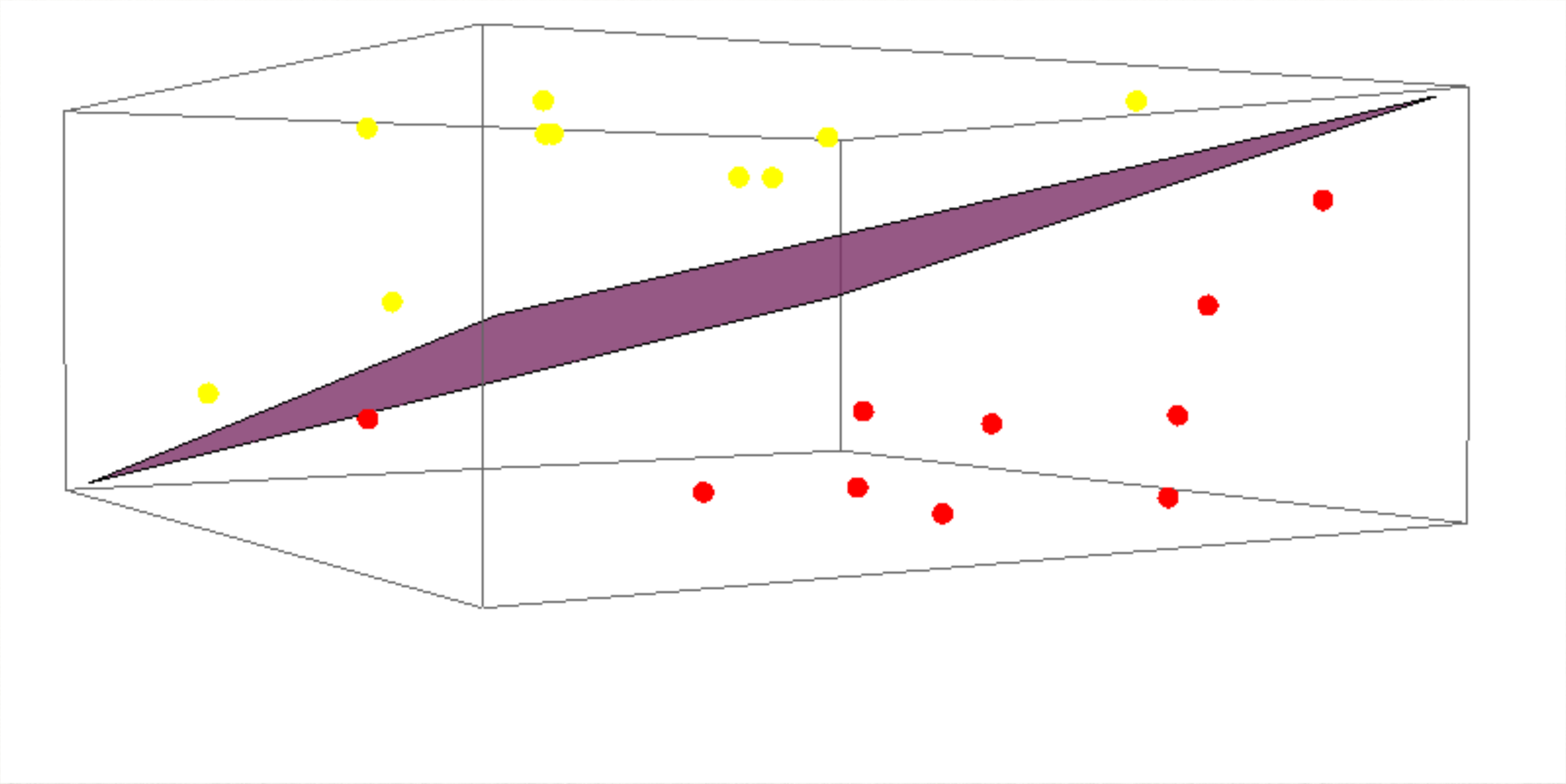
1.2.2 K-NEAREST NEIGHBORS

K-NN algorithm is one of the simplest classification algorithm and it is used to identify the data points that are separated into several classes to predict the classification of a new sample point. K-NN is a **non-parametric,** lazy learning algorithm. It classifies new cases based on a similarity measure (e.g. distance functions).



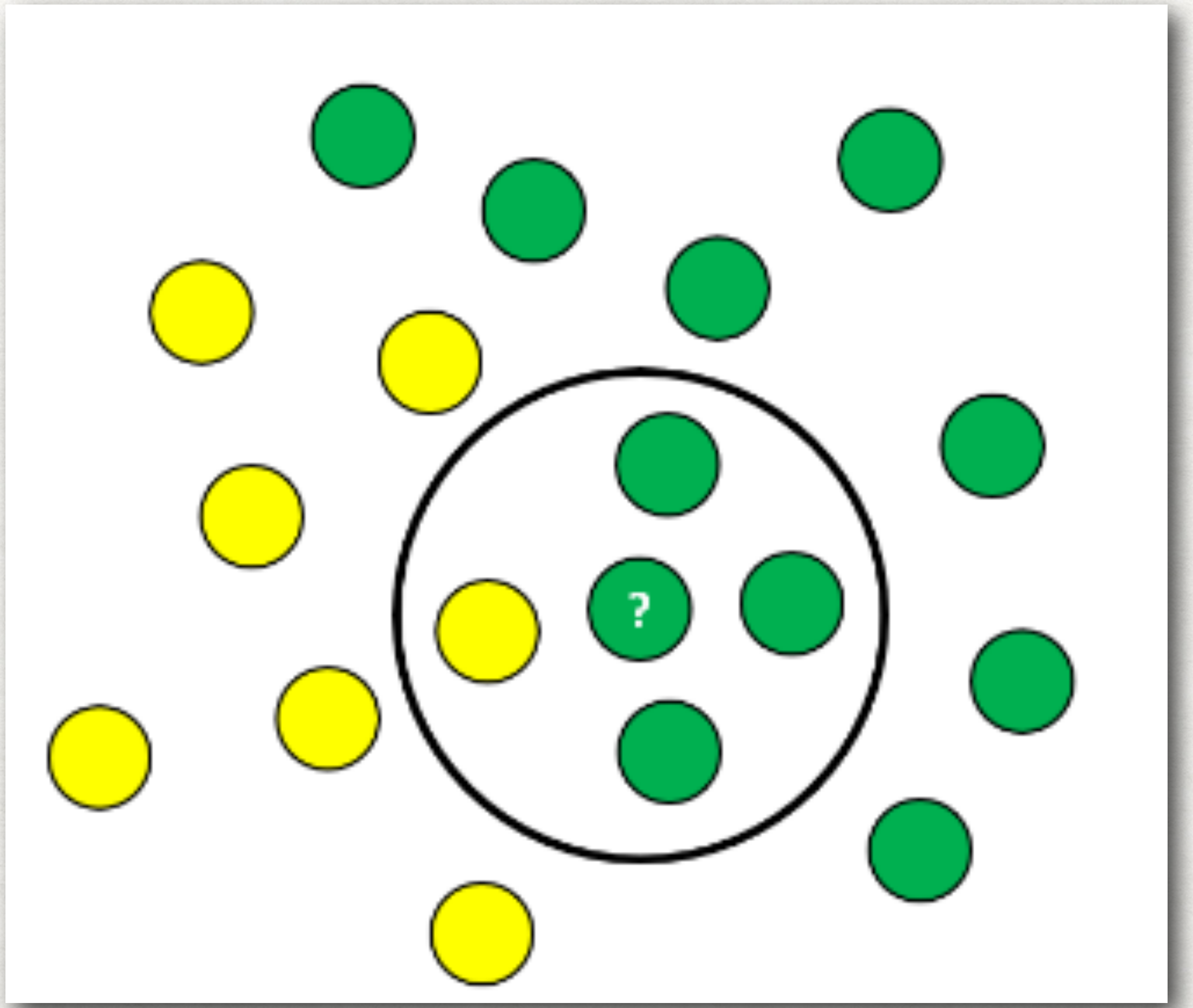
1.2.3 SUPPORT VECTOR MACHINE

Support Vector is used for both regression and Classification. It is based on the concept of decision planes that define decision boundaries. A decision plane(hyperplane) is one that separates between a set of objects having different class memberships.



1.2.4 NAIVE BAYES

Naive Bayes classifier is based on Bayes' theorem with the independence assumptions between predictors i.e it assumes the presence of a feature in a class is unrelated to any other feature. Even if these features depend on each other or upon the existence of the other features, all of these properties independently. Thus, the name Naive Bayes.



1.2.5 CLASSIFICATION CODE ALONG

Will be presented on the seminar day

File Edit View Insert Cell Kernel Widgets Help | Python 3

```
In [75]: y_pred = model.predict(X)
         y_class_pred = y_pred > 0.5

In [76]: from sklearn.metrics import accuracy_score

In [77]: print("The accuracy score is {:0.3f}".format(accuracy_score(y, y_class_pred

         The accuracy score is 0.790
```

Train/Test split

```
In [79]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

In [ ]: params = model.get_weights()
        params = [np.zeros(w.shape) for w in params]
        model.set_weights(params)

In [ ]: print("The accuracy score is {:0.3f}".format(accuracy_score(y, model.predic
```

MACHINE LEARNING

1. Supervised Learning

   1.1 Linear Regression

      1.1.1 Simple Linear Regression

      1.1.2 Multiple Linear Regression

      1.1.3 Polynomial Regression

      1.1.4 Cost function

      1.1.5 Linear Regression and Cost Function code along

   1.2 Classification

      1.2.1 Logistic Regression

      1.2.2 K Nearest Neighbors

      1.2.3 SVM

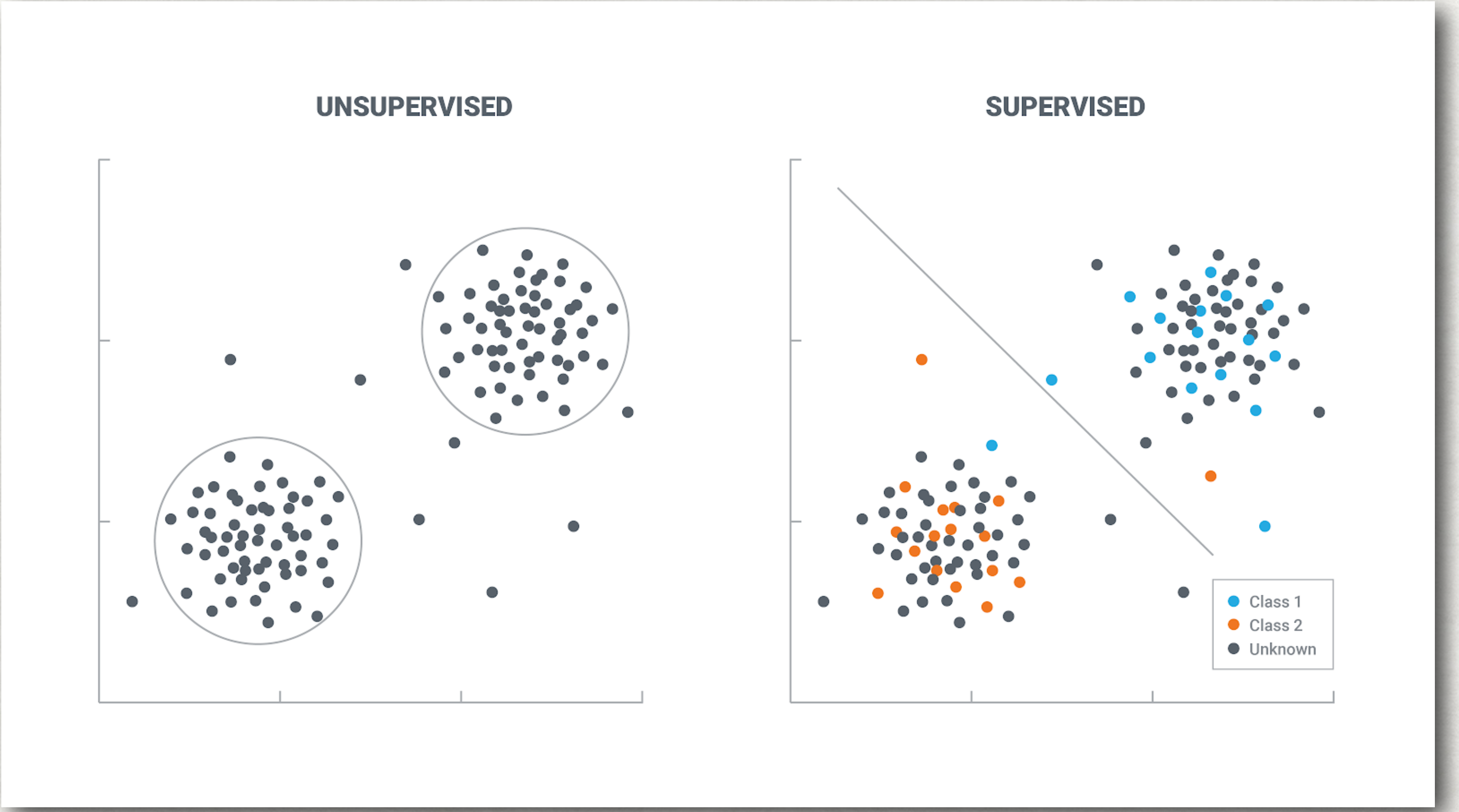
      1.2.4 Naive Bayes

      1.3.5 Classification code along

2. Unsupervised Learning
3. Semi Supervised Learning
4. Reinforcement Learning
5. Overfitting
6. Cross Validation

2.UNSUPERVISED LEARNING

In unsupervised learning, a deep learning model is handed a dataset without explicit instructions on what to do with it. The training dataset is a collection of examples without a specific desired outcome or correct answer.
The neural network then attempts to automatically find structure in the data by extracting useful features and analyzing its structure.



2. UNSUPERVISED LEARNING

- **Clustering**: Without being an expert ornithologist, it's possible to look at a collection of bird photos and separate them roughly by species, relying on cues like feather color, size or beak shape. That's how the most common application for unsupervised learning, clustering, works: the deep learning model looks for training data that are similar to each other and groups them together.

- **Anomaly detection**: Banks detect fraudulent transactions by looking for unusual patterns in customer's purchasing behavior. For instance, if the same credit card is used in California and Denmark within the same day, that's cause for suspicion. Similarly, unsupervised learning can be used to flag outliers in a dataset.

- **Association**: Fill an online shopping cart with diapers, applesauce and sippy cups and the site just may recommend that you add a bib and a baby monitor to your order. This is an example of association, where certain features of a data sample correlate with other features. By looking at a couple key attributes of a data point, an unsupervised learning model can predict the other attributes with which they're commonly associated.

- **Autoencoders**: Autoencoders take input data, compress it into a code, then try to recreate the input data from that summarized code. While a neat deep learning trick, there are fewer real-world cases where a simple autocoder is useful. But add a layer of complexity and the possibilities multiply: by using both noisy and clean versions of an image during training, autoencoders can remove noise from visual data like images, video or medical scans to improve picture quality.

3. SEMI-SUPERVISED LEARNING

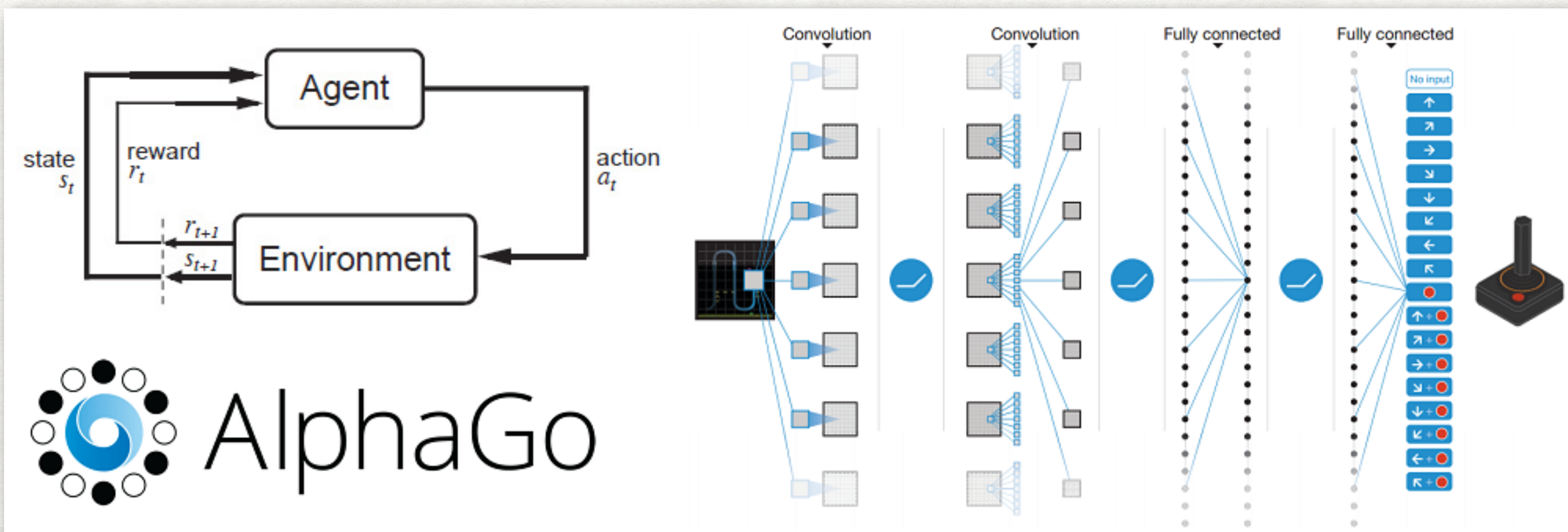
Semi-supervised learning is, for the most part, just what it sounds like: a training dataset with both labeled and unlabeled data. This method is particularly useful when extracting relevant features from the data is difficult, and labeling examples is a time-intensive task for experts.

Semi-supervised learning is especially useful for medical images, where a small amount of labeled data can lead to a significant improvement in accuracy.

4. REINFORCEMENT LEARNING

Reinforcement learning operates on the same principle — and actually, video games are a common test environment for this kind of research.

In this kind of machine learning, AI agents are attempting to find the optimal way to accomplish a particular goal, or improve performance on a specific task. As the agent takes action that goes toward the goal, it receives a reward. The overall aim: predict the best next step to take to earn the biggest final reward.



MACHINE LEARNING

1. Supervised Learning
    1.1 Linear Regression
        1.1.1 Simple Linear Regression
        1.1.2 Multiple Linear Regression
        1.1.3 Polynomial Regression
        1.1.4 Cost function
        1.1.5 Linear Regression and Cost Function code along
    1.2 Classification
        1.2.1 Logistic Regression
        1.2.2 K Nearest Neighbors
        1.2.3 SVM
        1.2.4 Naive Bayes
        1.3.5 Classification code along
2. Unsupervised Learning
3. Semi Supervised Learning
4. Reinforcement Learning
5. Overfitting
6. Cross Validation
7. ML EXERCISE AND SOLUTION

5. OVERFITTING

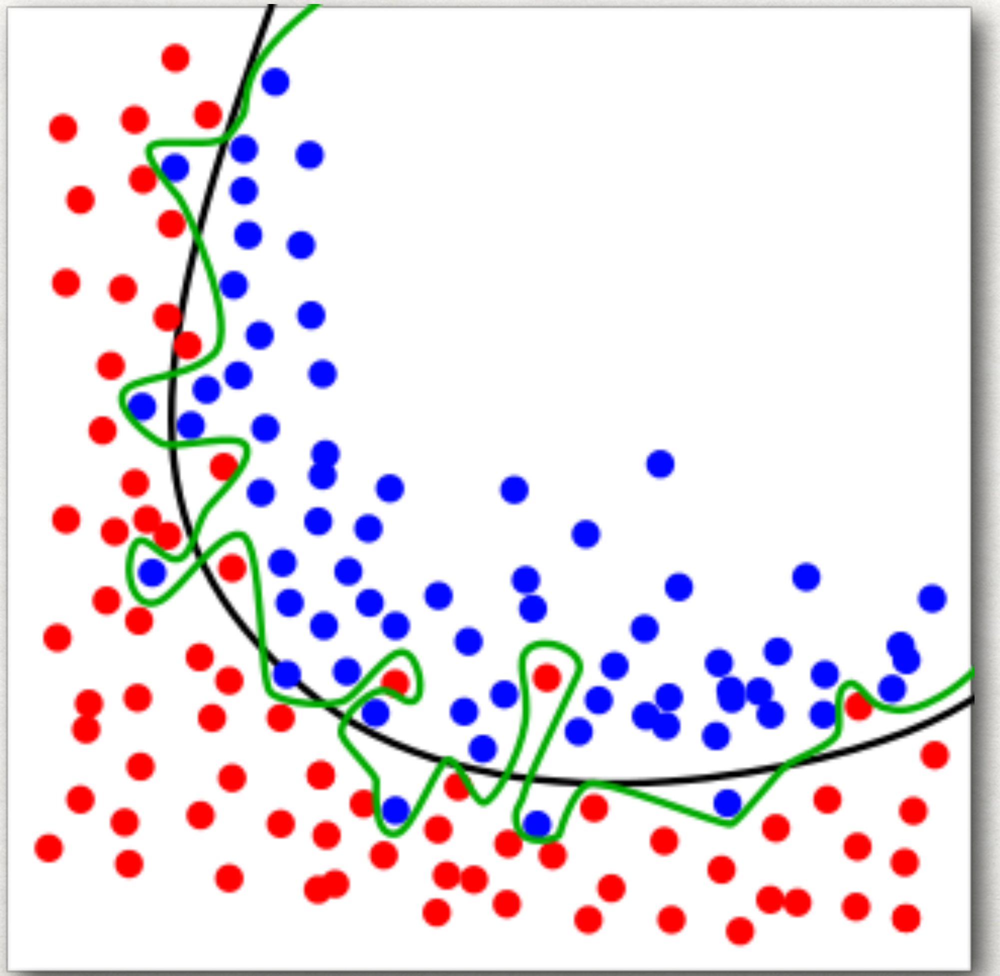
Let's say we want to predict if a student will land a job interview based on her resume.

Now, assume we train a model from a dataset of 10,000 resumes and their outcomes.

Next, we try the model out on the original dataset, and it predicts outcomes with 99% accuracy… wow!

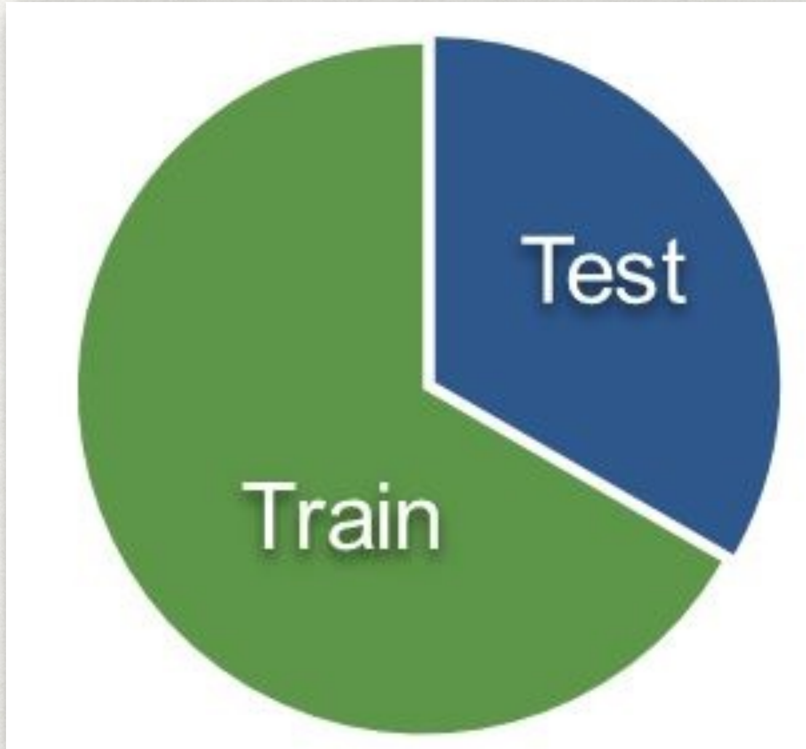
But now comes the bad news.

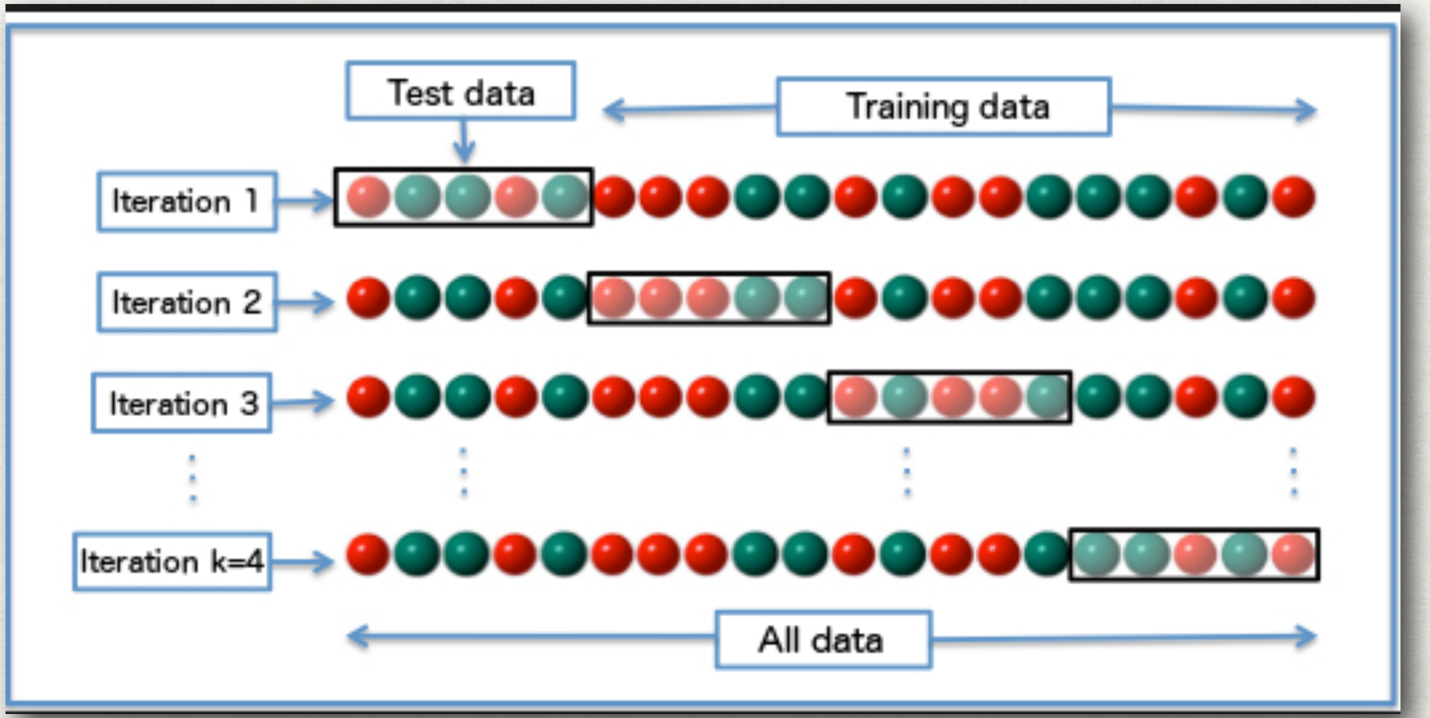
When we run the model on a new ("unseen") dataset of resumes, we only get 50% accuracy… uh-oh!



6. CROSS VALIDATION

To evaluate the performance of any machine learning model we need to test it on some unseen data. Based on the models performance on unseen data we can say weather our model is Under-fitting/Over-fitting/Well generalised. Cross validation (CV) is one of the technique used to test the effectiveness of a machine learning models, it is also a re-sampling procedure used to evaluate a model if we have a limited data.





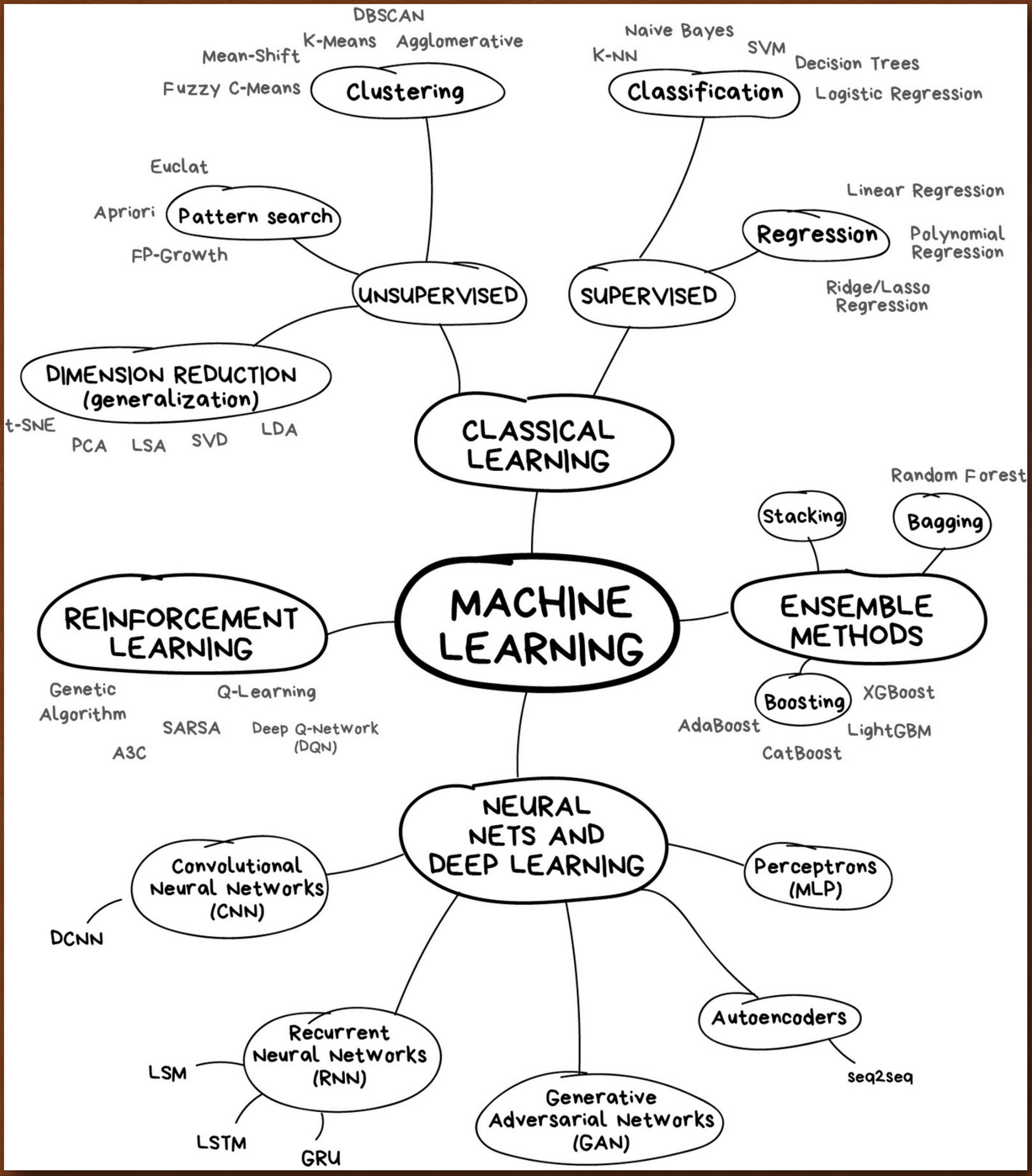
- Train_Test Split approach.

K-Folds Cross Validation:

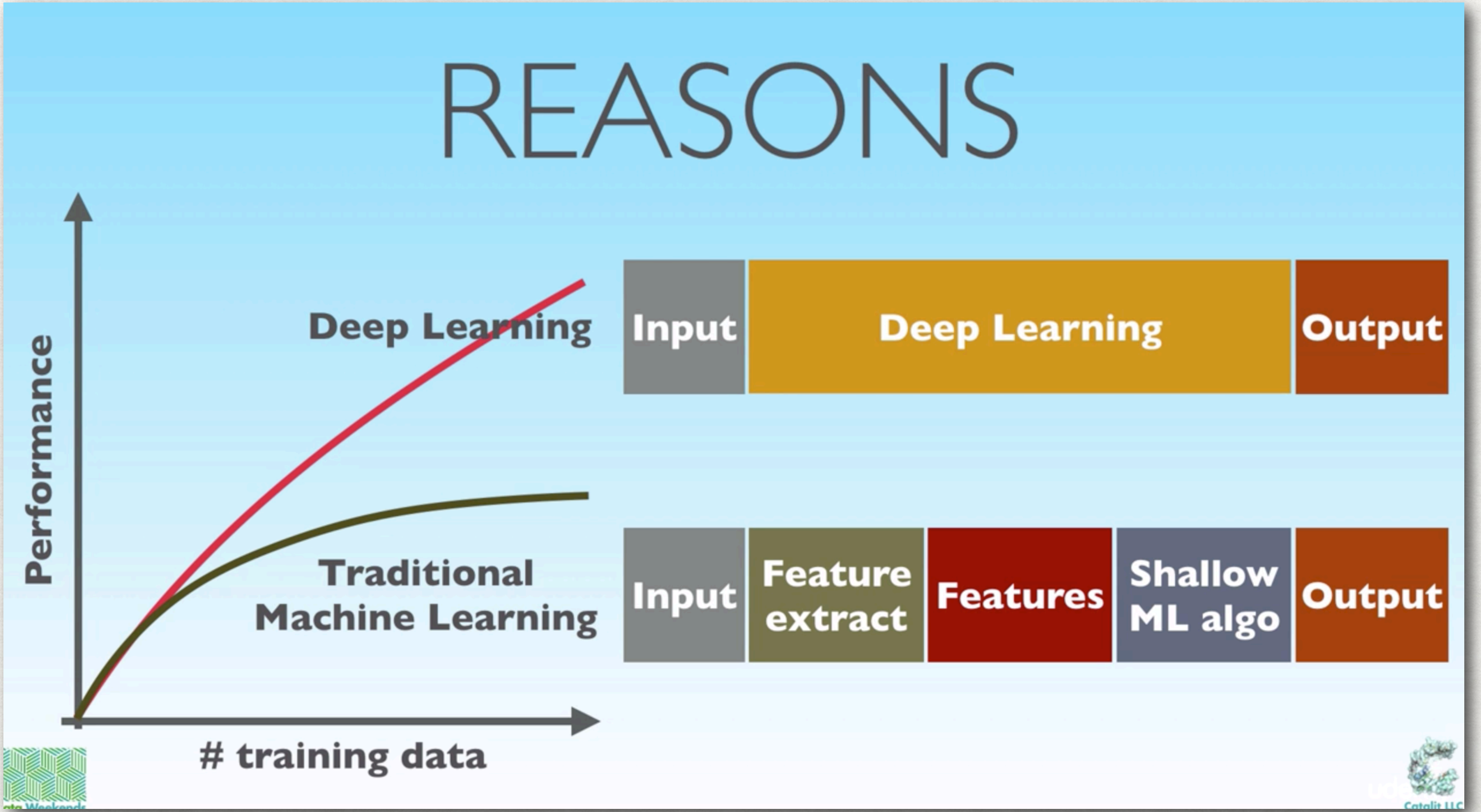
12. ML EXERCISE AND SOLUTION

Will be showed on Seminar day

DEEP LEARNING



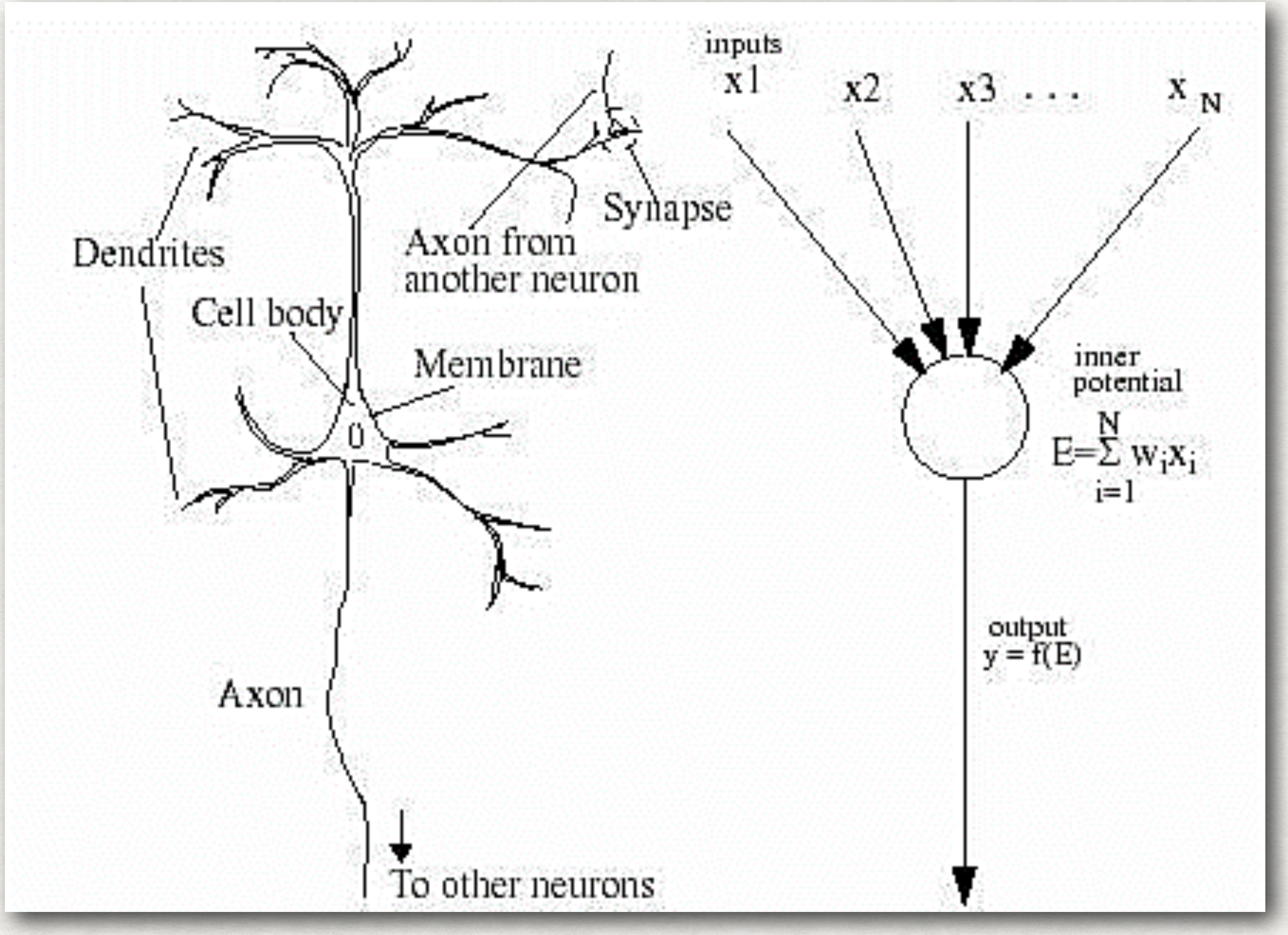
1. ML VS DL



REASONS
Performance
Deep Learning
Traditional
Machine Learning
training data
Input
Deep Learning
Output
Input
Feature
extract
Features
Shallow
ML algo
Output

2. NEURAL NETWORKS

Artificial neural networks recreate the structure of human neurons to process information resulting in much more accurate results than previously used regression models.



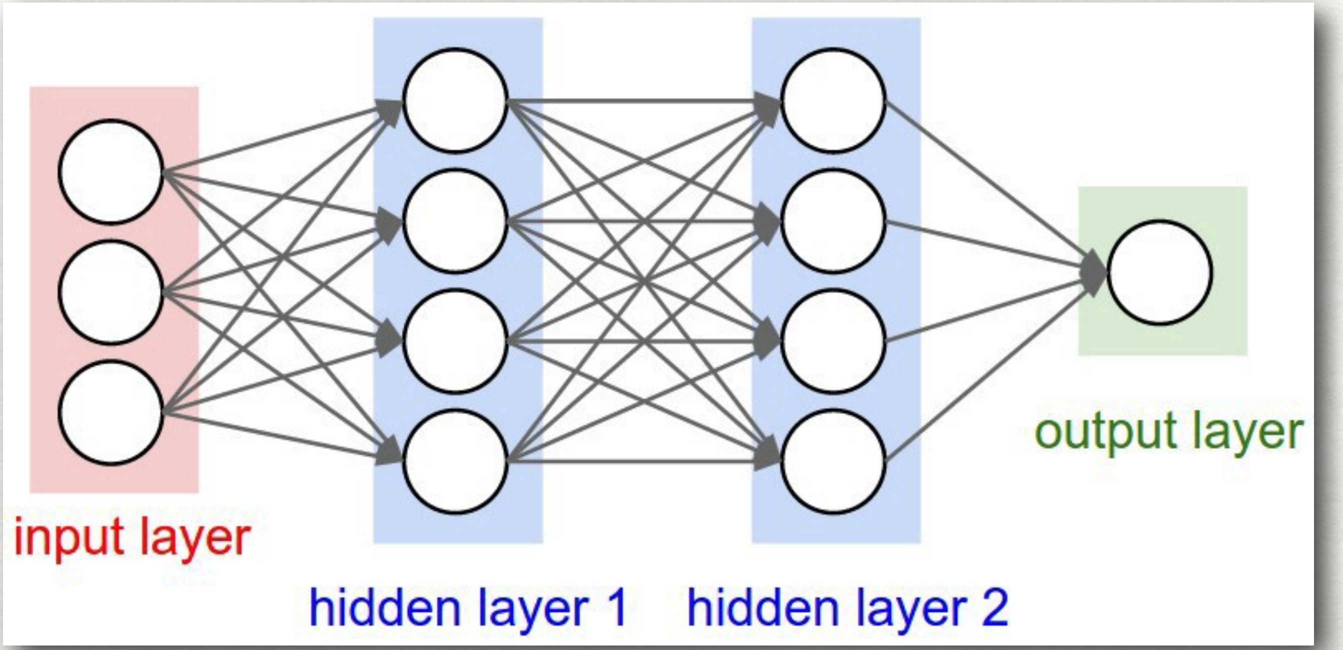
2.1 THE PARTS OF A NEURAL NETWORK

A neural network is made up of 3 main parts:

+) Input layer: The layer that inputs information for the neural network to process

+) Hidden layer: These layers do all the processing for neural networks. Generally speaking, the more hidden layers you have, the more accurate the neural network will be.

+) Output layer: This layer simply brings together the information from the last hidden layer of the network to output all the information you need from the program.



2.2 NEURAL NETWORKS CODE ALONG

Will be presented on Seminar Day

File Edit View Insert Cell Kernel Widgets Help | Python 3

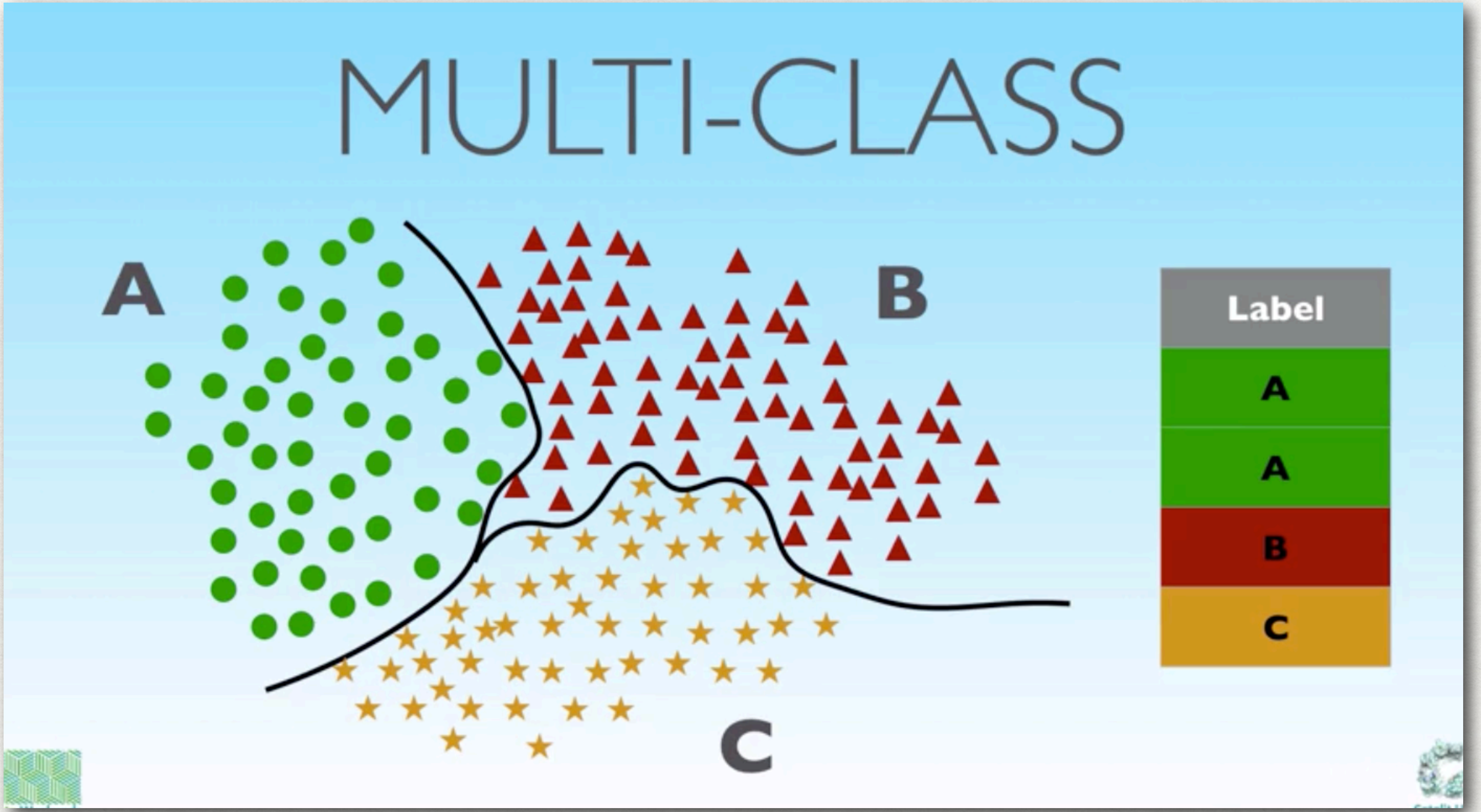
```
from keras.optimizers import SGD, Adam
```

```
Using TensorFlow backend.
```

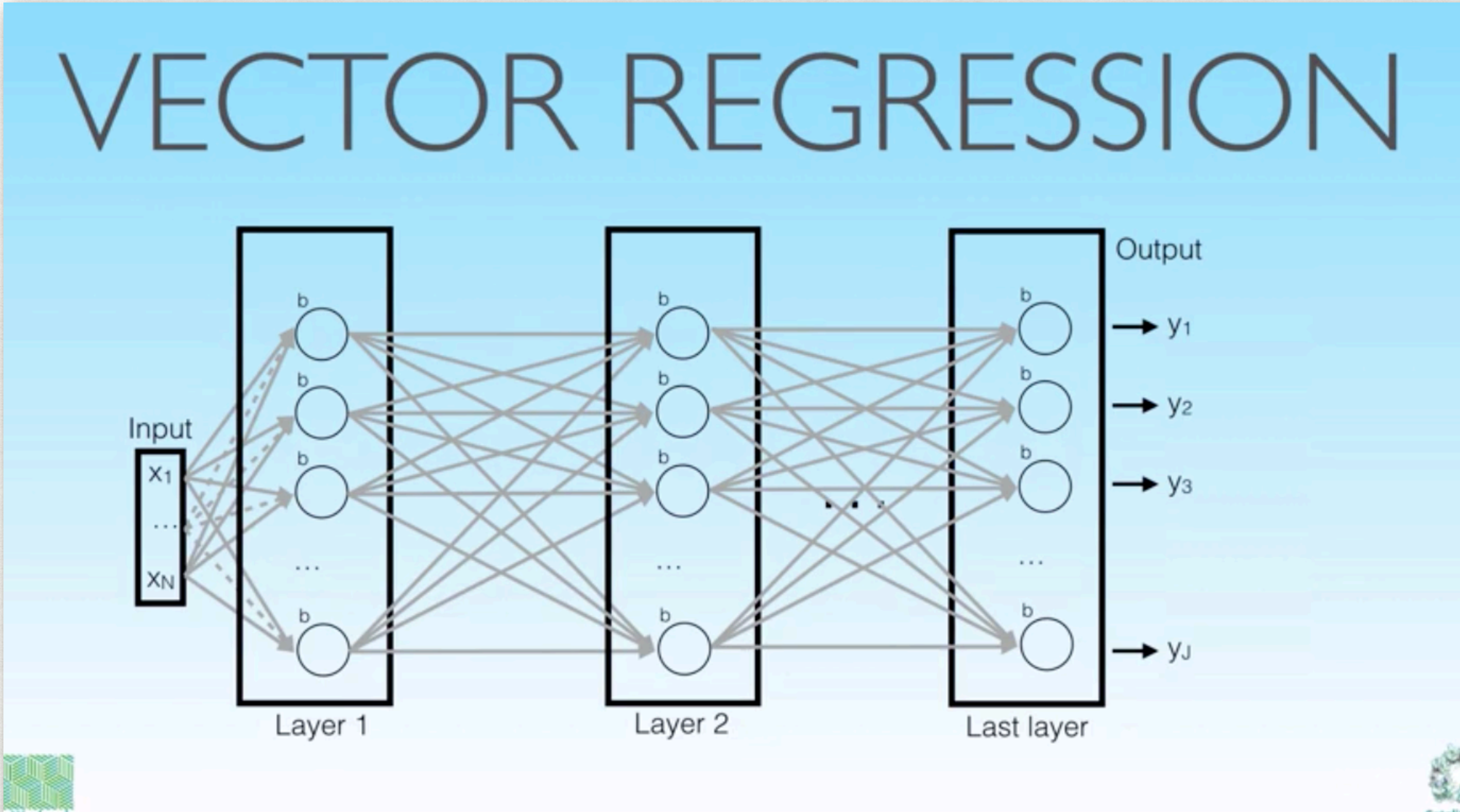
Shallow Model

```
In [7]: model = Sequential()
        model.add(Dense(1, input_shape=(2,), activation='sigmoid'))
        model.compile(Adam(lr=0.05), 'binary_crossentropy', metrics=['accuracy'])

In [8]: model.fit(X_train, y_train, epochs=200, verbose=0)

Out[8]: <keras.callbacks.History at 0x11a3a45c0>

In [9]: results = model.evaluate(X_test, y_test)

         32/300 [==>...........................] - ETA: 0s

In [ ]: print("The Accuracy score on the Train set is:\t{:0.3f}".format(results[1]))
```

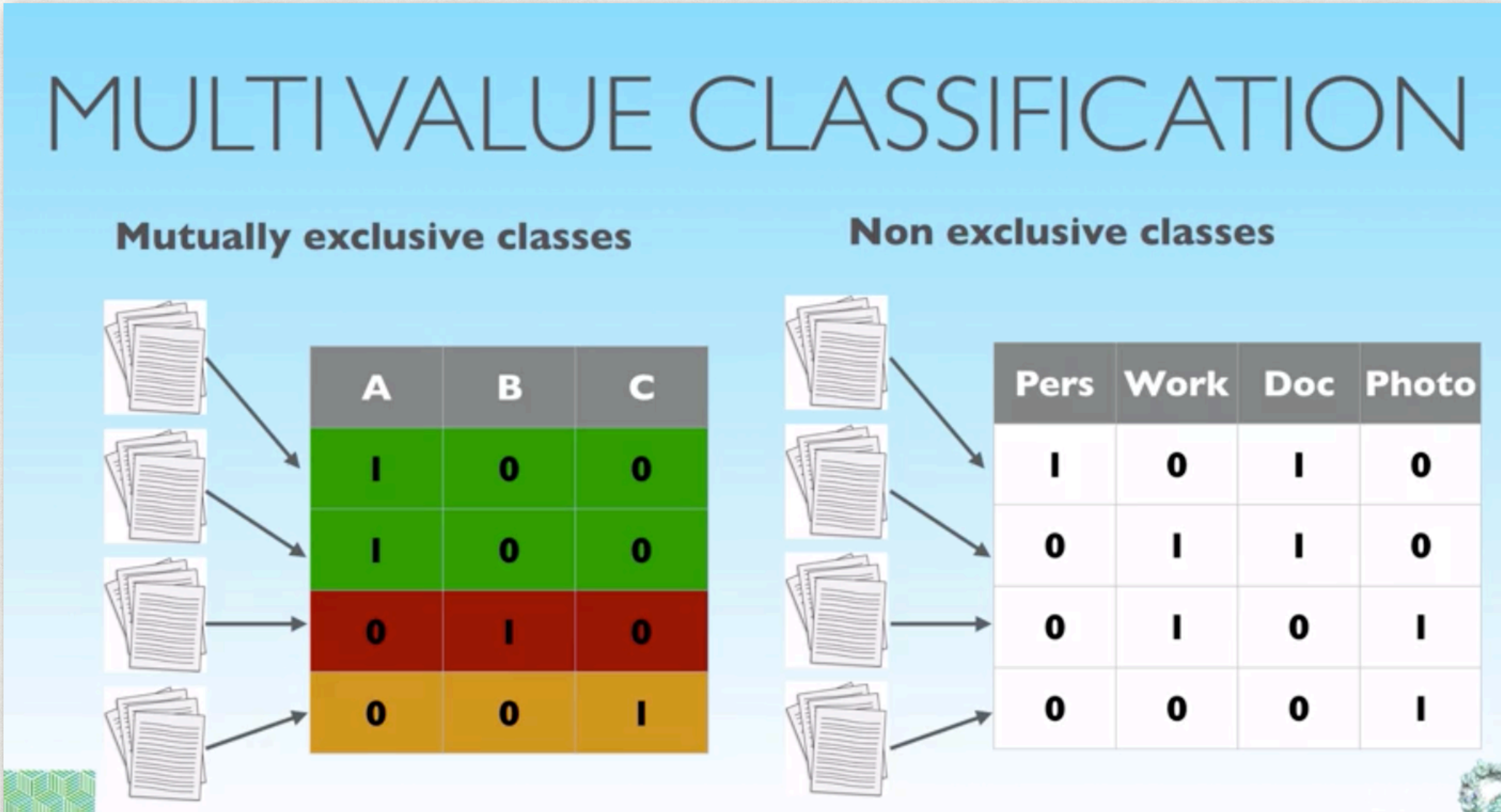
3. MULTIPLE OUTPUTS



3. MULTIPLE OUTPUTS



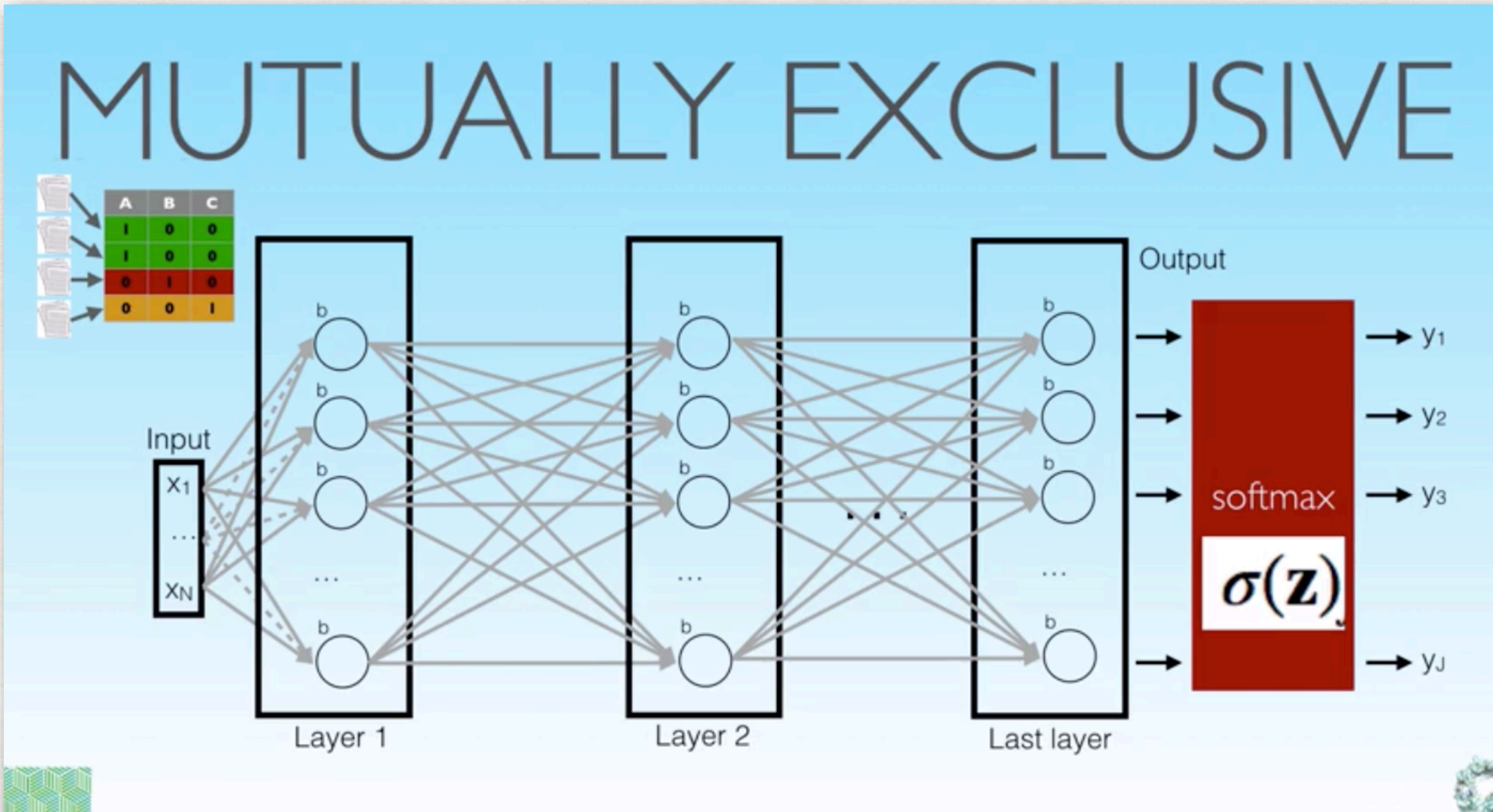
3. MULTIPLE OUTPUTS



3. MULTIPLE OUTPUTS

SOFTMAX

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for j = 1, …, K.}$$



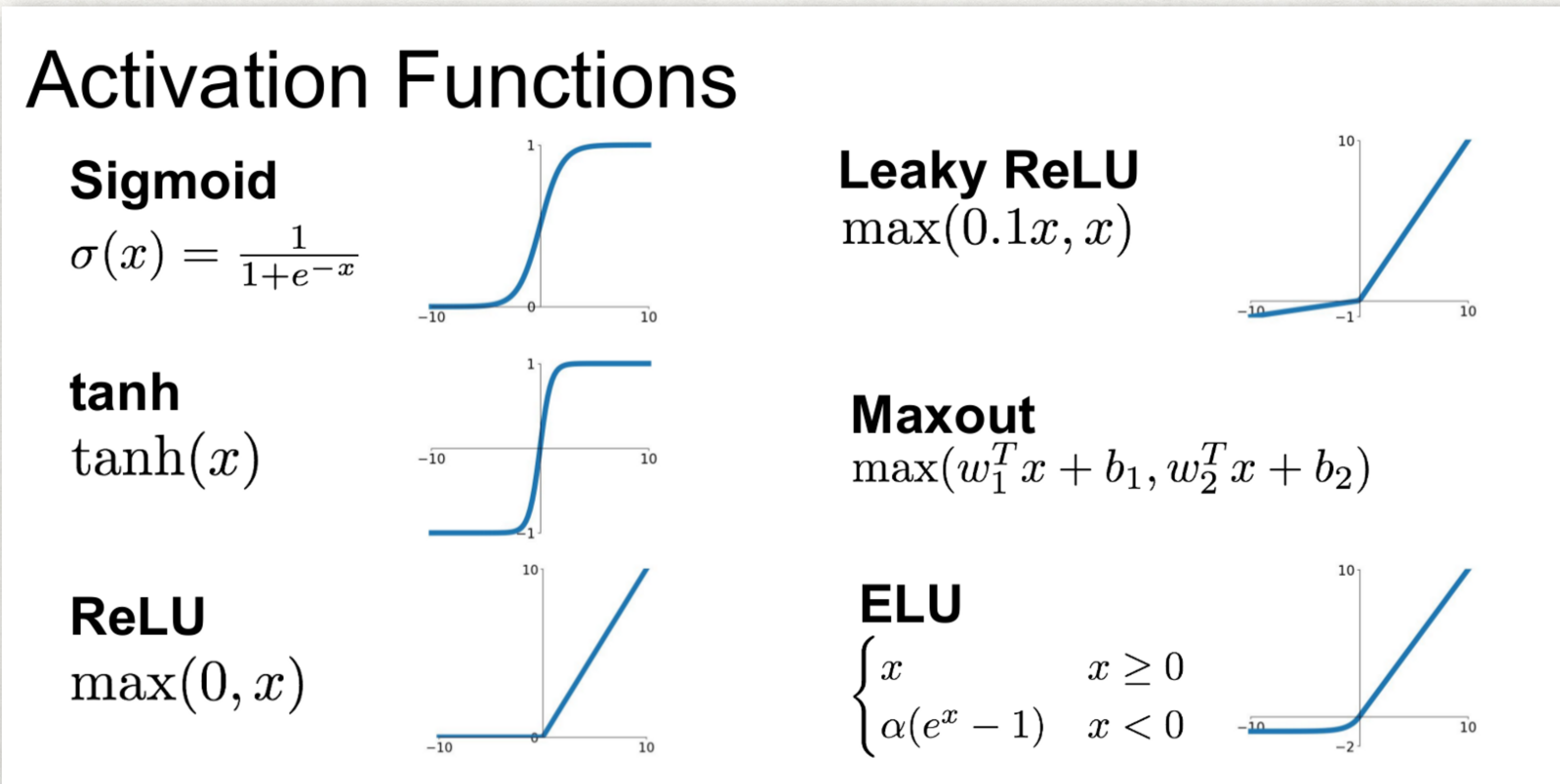
3. MUTICLASS CLASSIFICATION CODE ALONG

Will be showed on seminar day

```
File    Edit    View    Insert    Cell    Kernel    Widgets    Help        Python 3

In [29]: X_train, X_test, y_train, y_test = train_test_split(X.values, y_cat,
                                                             test_size=0.2)

In [30]: model = Sequential()
         model.add(Dense(3, input_shape=(4,), activation='softmax'))
         model.compile(Adam(lr=0.1),
                       loss='categorical_crossentropy',
                       metrics=['accuracy'])

In [31]: model.fit(X_train, y_train, epochs=20, validation_split=0.1)

         Train on 108 samples, validate on 12 samples
         Epoch 1/20
         108/108 [==============================] - 0s - loss: 4.0228 - acc: 0.287
         0 - val_loss: 1.6317 - val_acc: 0.0000e+00
         Epoch 2/20
         108/108 [==============================] - 0s - loss: 2.0102 - acc: 0.203
         7 - val_loss: 1.1606 - val_acc: 0.5833
         Epoch 3/20
         108/108 [==============================] - 0s - loss: 1.6086 - acc: 0.500
         0 - val_loss: 0.7879 - val_acc: 0.7500
         Epoch 4/20
```

4. ACTIVATION FUNCTION

The activation function takes into account the interaction effects in different parameters and does a transformation after which it gets to decide which neuron *passes forward* the value into the next layer.



5. EXERCISE AND SOLUTION CODE ALONG

Will be showed on seminar day

END TO END A REAL DEEP LEARNING PROJECT AND DEMO

Will be showed in confidence on Seminar Day

HOME WORK