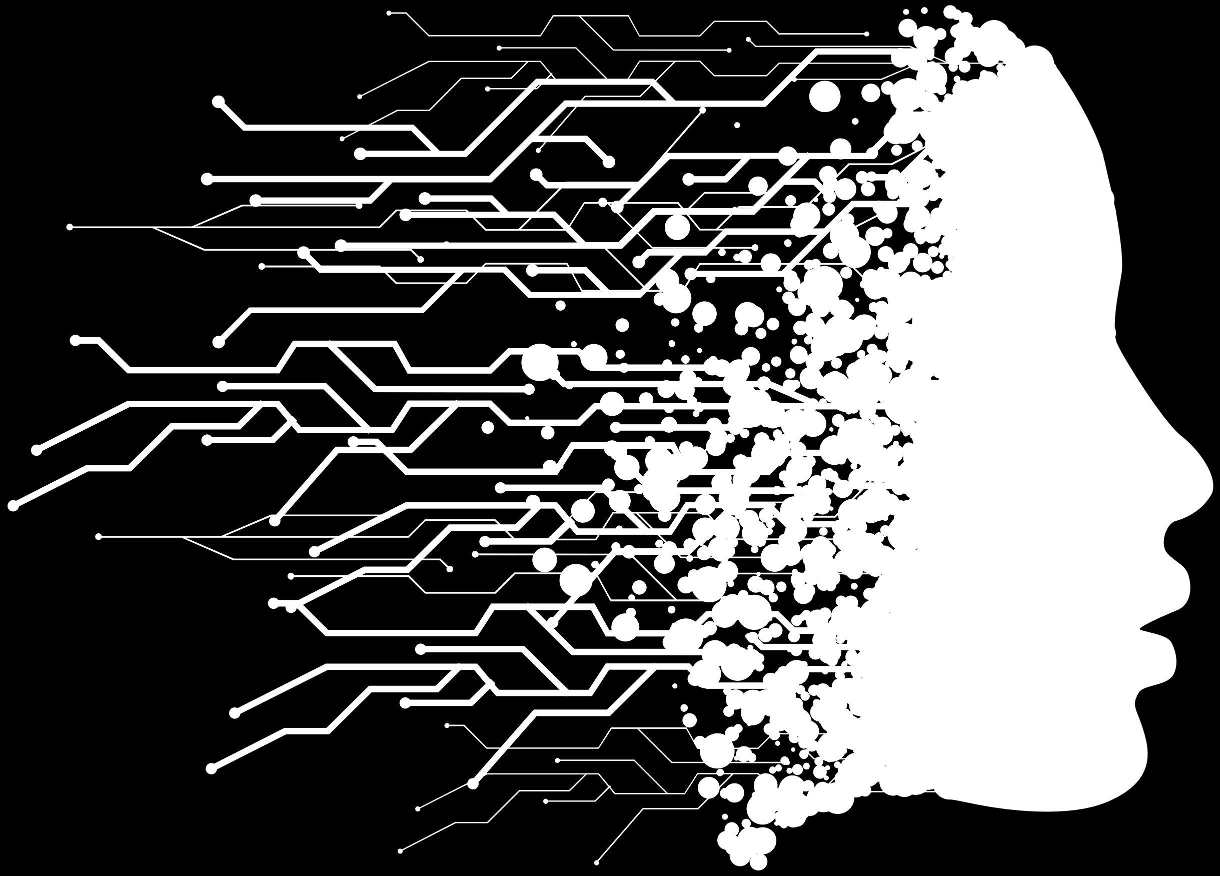


Machine Learning With Spark (BD127)

We will be starting soon

Day 3

Welcome to Day 3 Spark Machine Learning

Please perform PRE-WORK

1. Access the virtual Lab using link https://html.inspiredvlabs.com Use the username TEKMD190-XX (replace XX with your number) and password **TEKBD127!23**

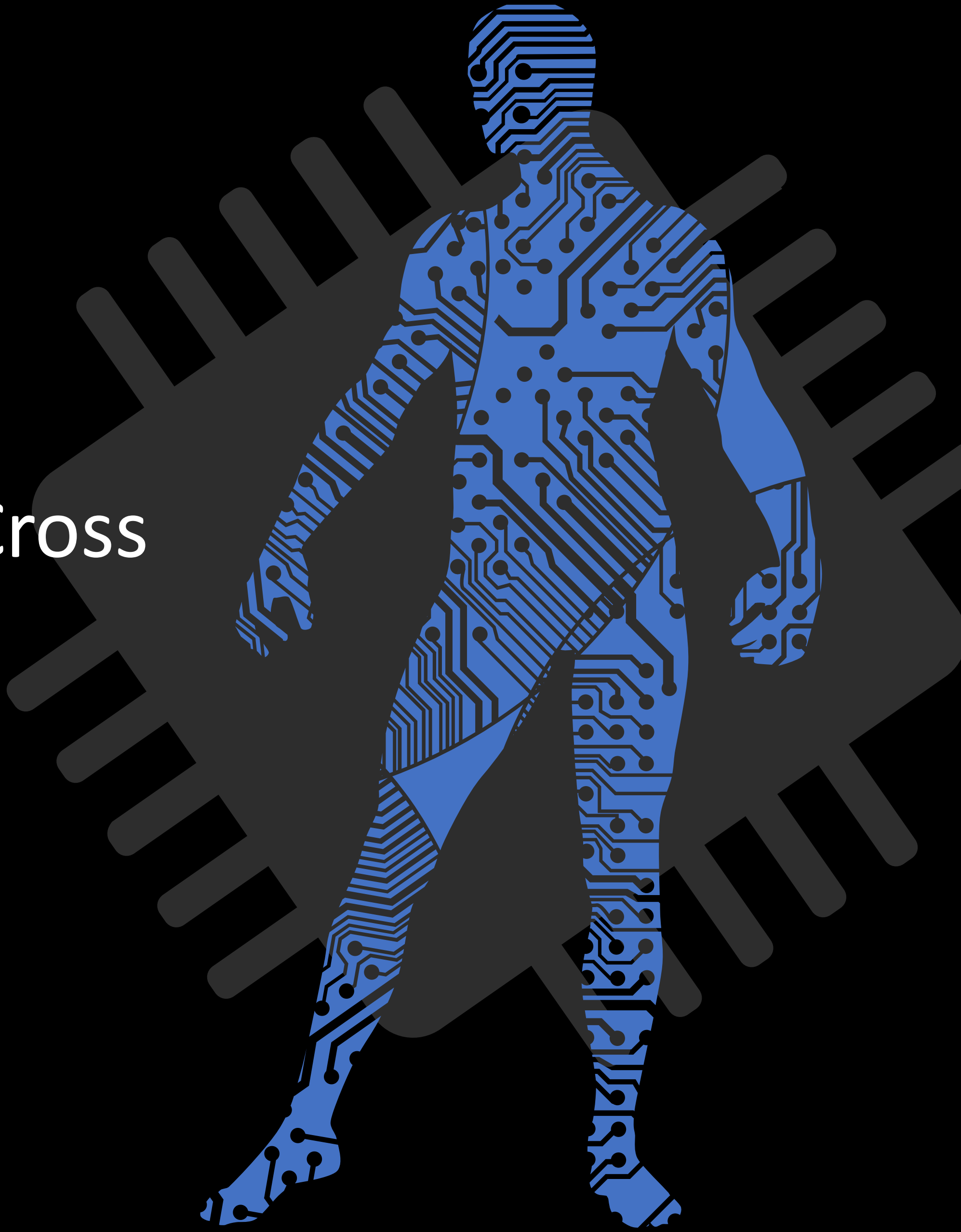
https://tinyurl.com/bdtSparkML

We will be starting soon

Last Name	First Name	Login Id
ALCEDO MORENO	ALVARO	TEKMD190-01
BOGADAPATI	BHAVANI	TEKMD190-02
BOOSTANI	ANOUSH	TEKMD190-03
FRENCH	CHRIS	TEKMD190-04
FRINO	MASSIMILIANO	TEKMD190-05
KULKARNI	ALPESH	TEKMD190-06
MA	CUONG	TEKMD190-07
MADAGANI	SRINIVASA	TEKMD190-08
MATULIS	STEPHEN	TEKMD190-09
MIAO	HUALING	TEKMD190-10
MILLER	KENT	TEKMD190-11
OBRIEN	CHARLES	TEKMD190-12
SELVARAJ	RAJESH KHANNA	TEKMD190-13
VINCENT	SWAROOP	TEKMD190-14
YOUSSEF	MINA	TEKMD190-15

Agenda – Day 3

- Recap of Day 2
- Complete project (Param Grid and Cross validation)
- Extra Credit
- Unsupervised Learning
- Feature Extraction and Elimination
- Spark, ML Environment set up
- Idea Clinic

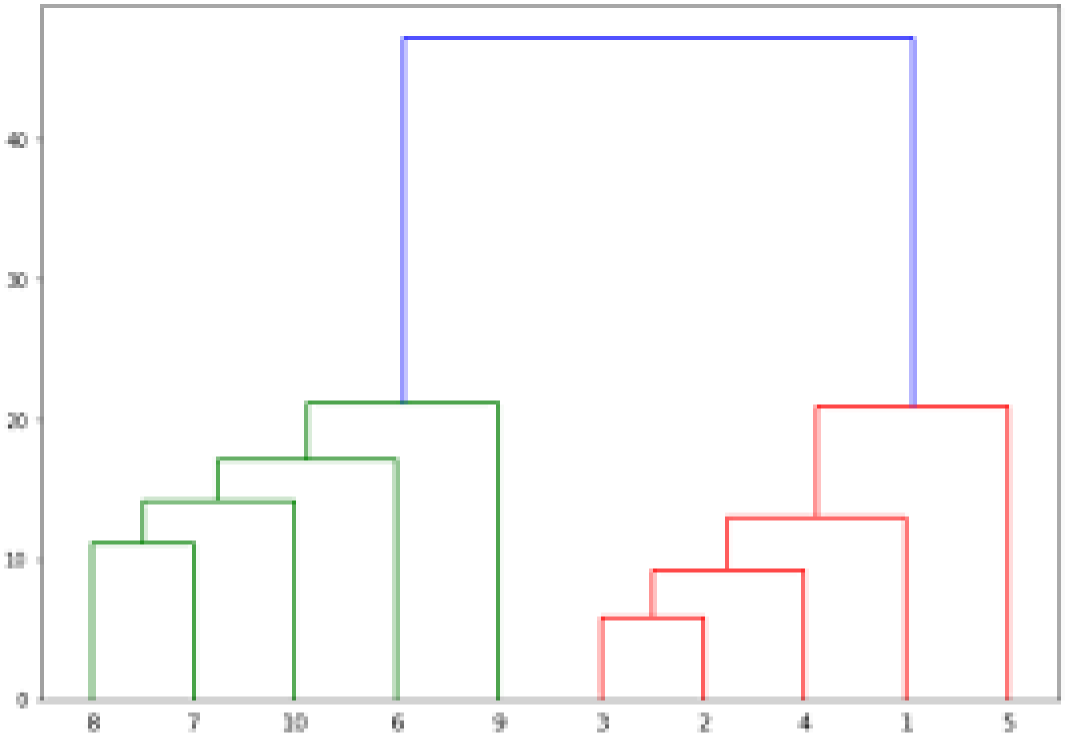


Hierarchical Clustering

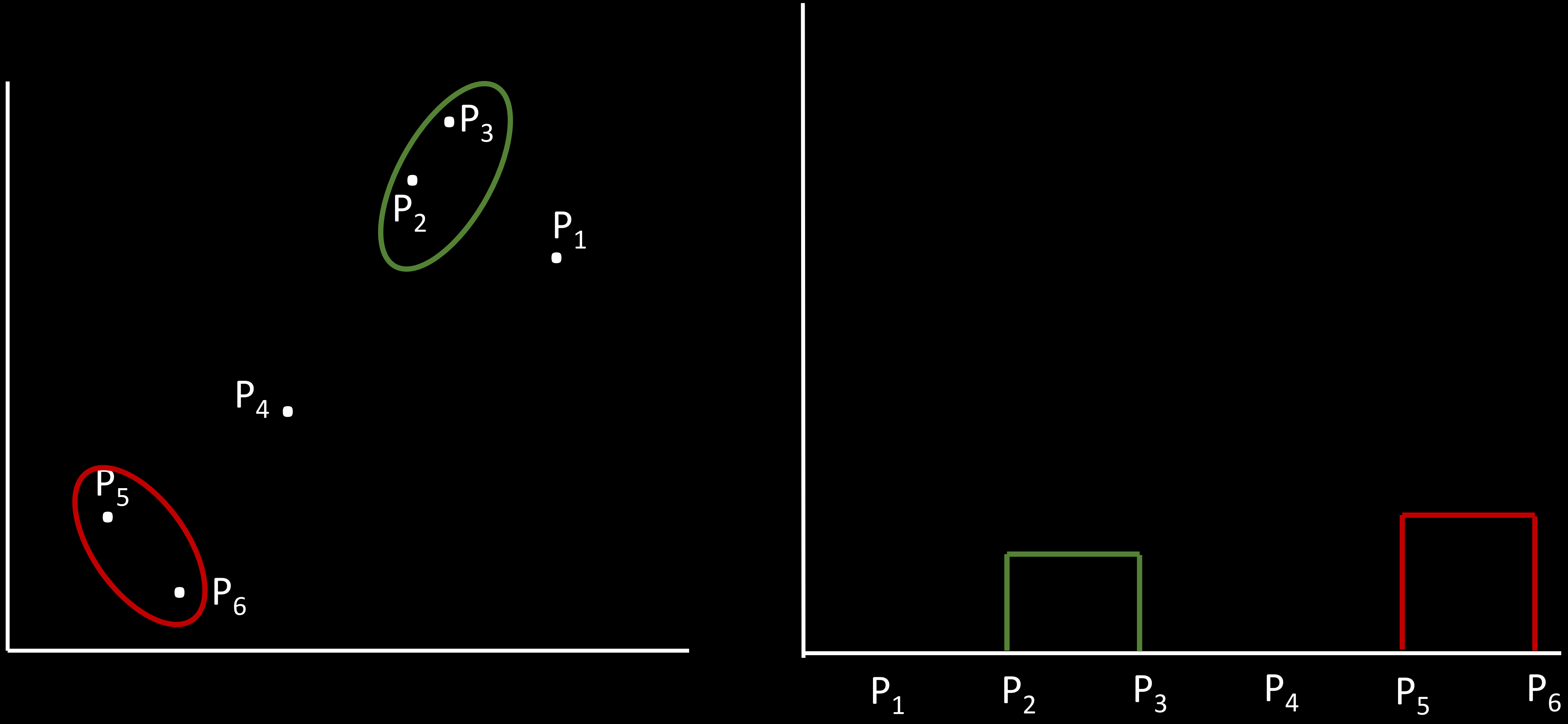
Hierarchical Clustering

- Hierarchical Clustering is like K-Means clustering except the processing is different
- Two types:
  - Agglomerative – bottom up approach
  - Divisive – top down approach (Bisecting K-Means)

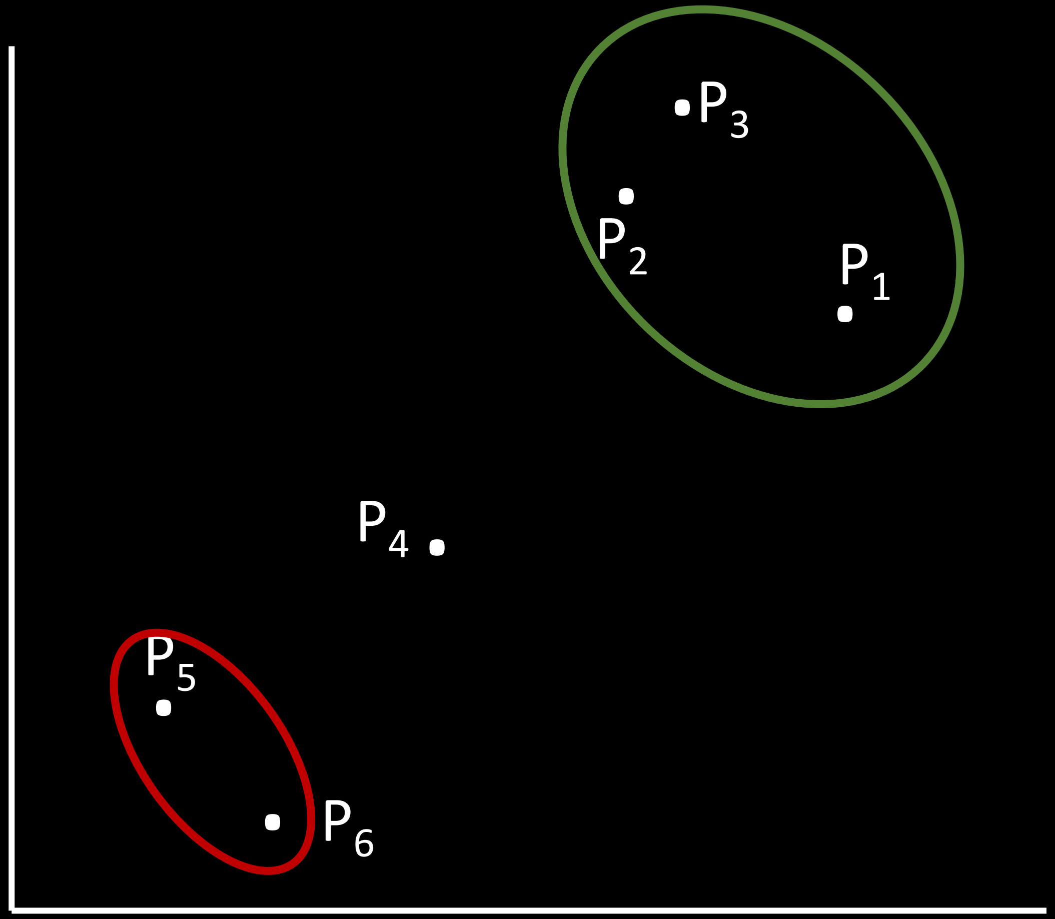
https://en.wikipedia.org/wiki/Dendrogram

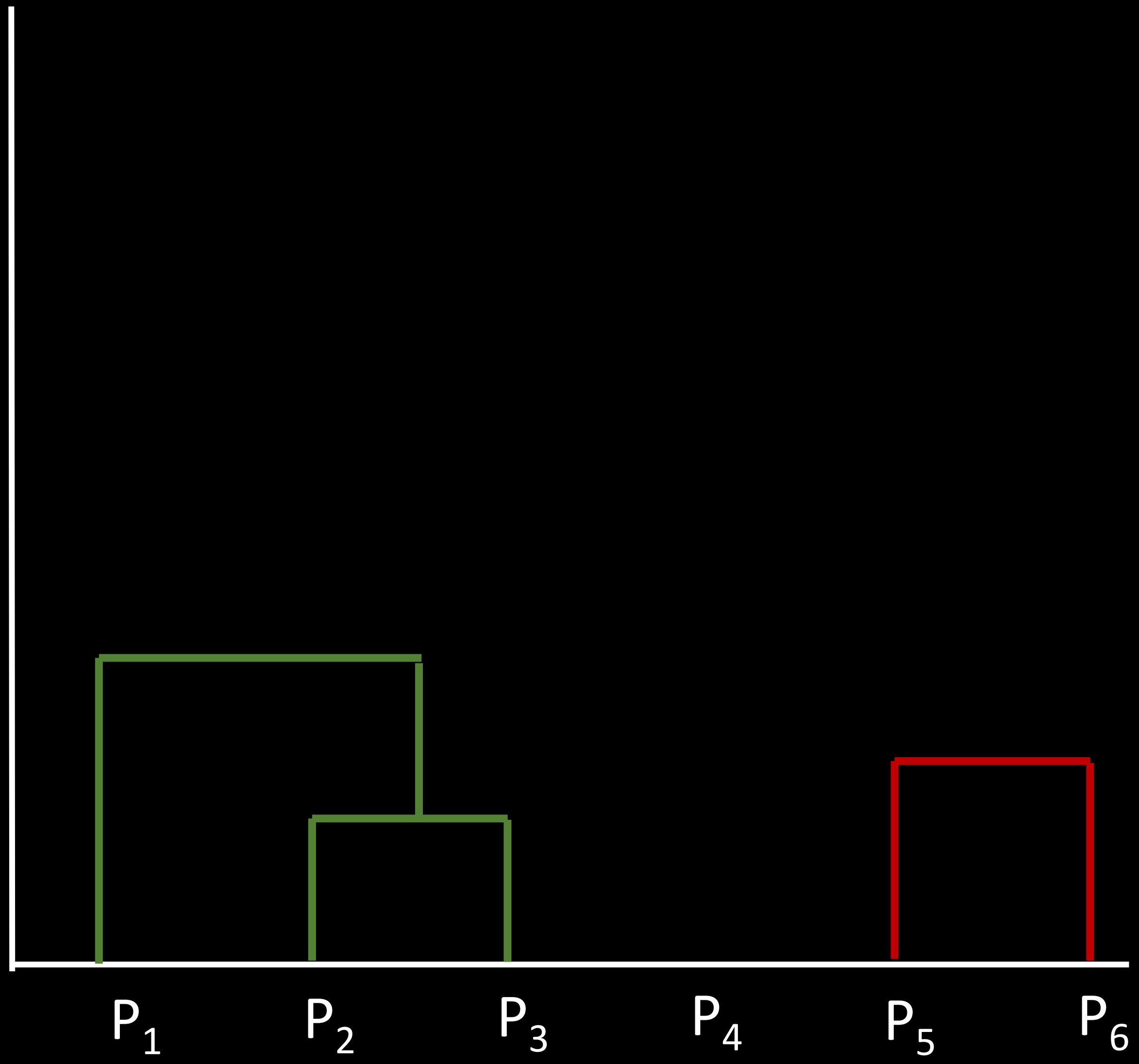


Hierarchical Clustering

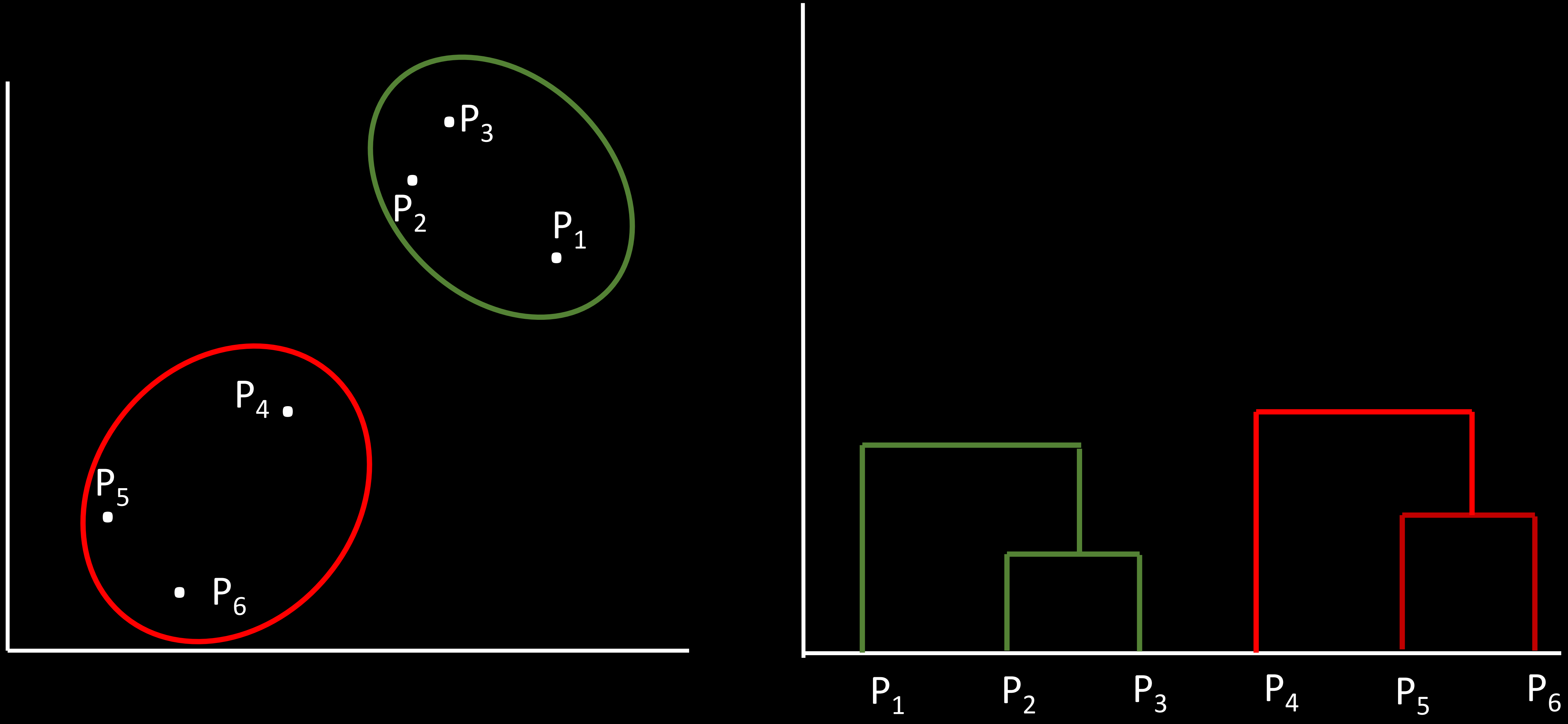


Hierarchical Clustering

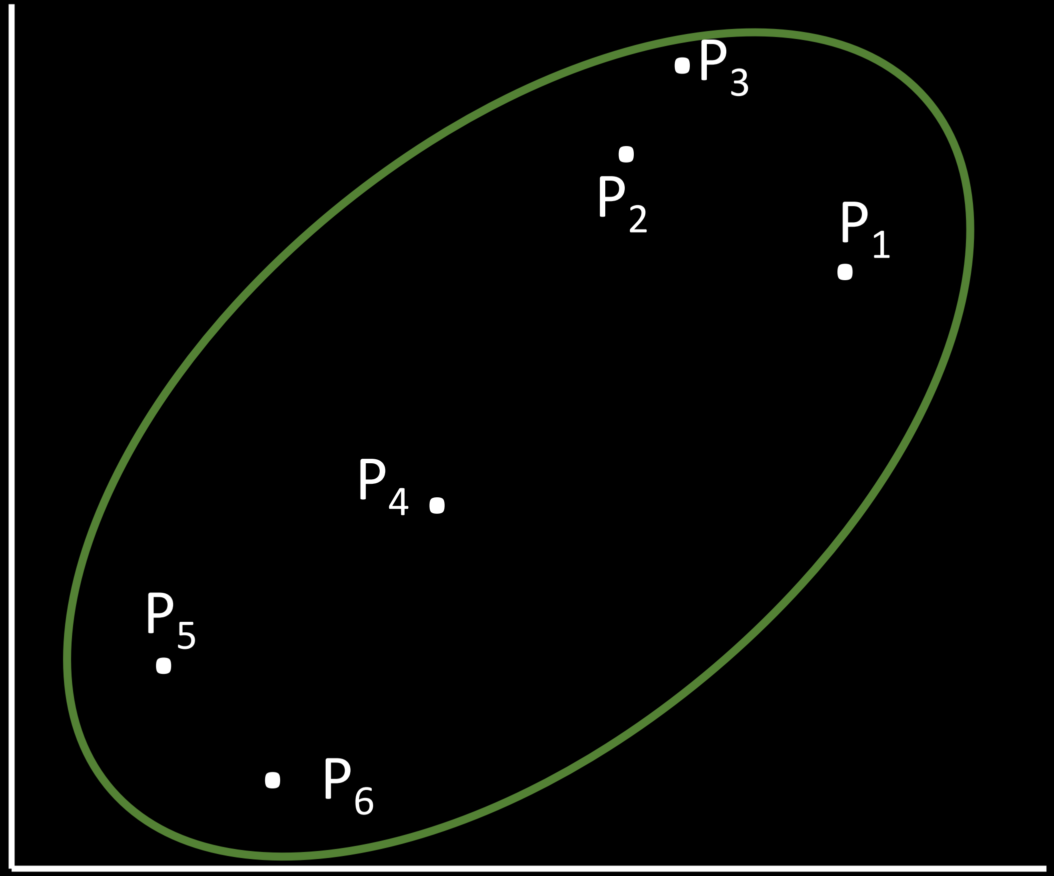


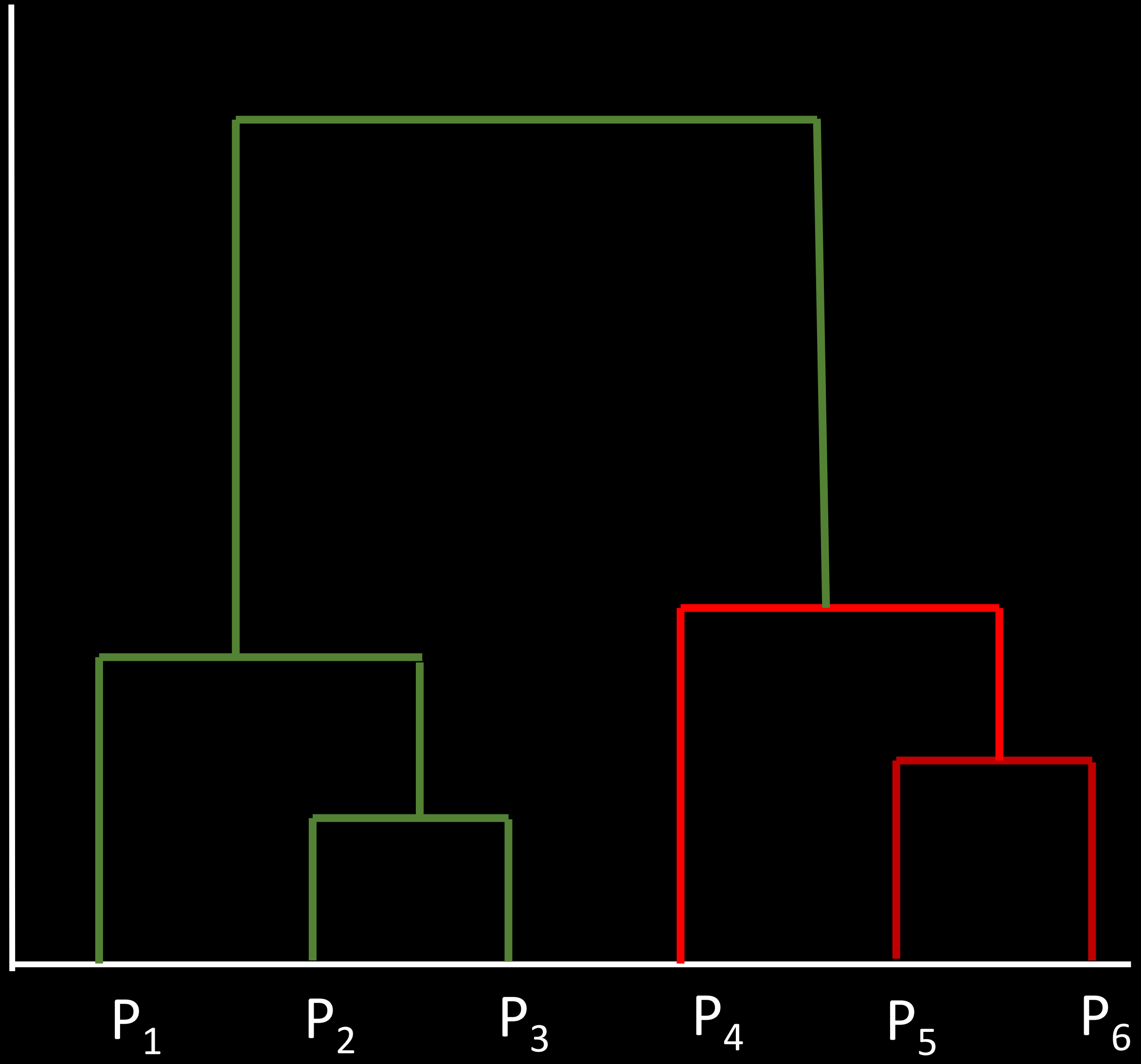


Hierarchical Clustering

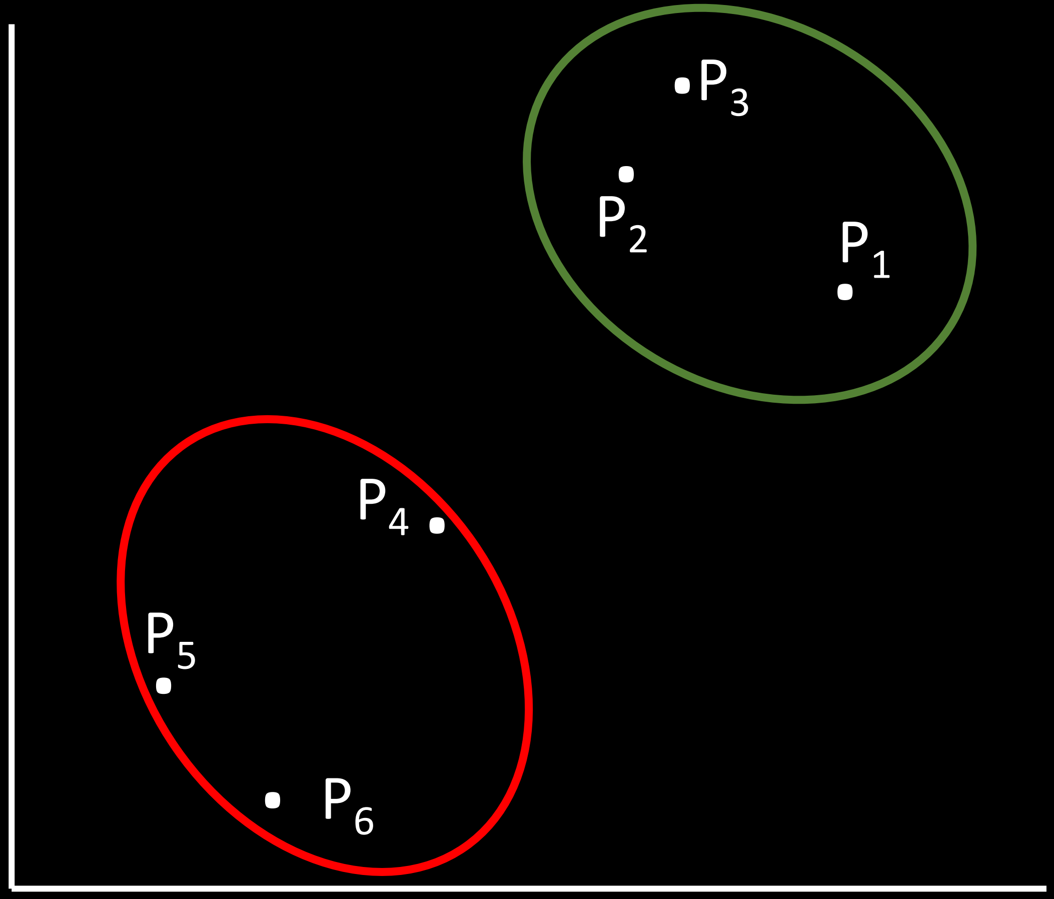


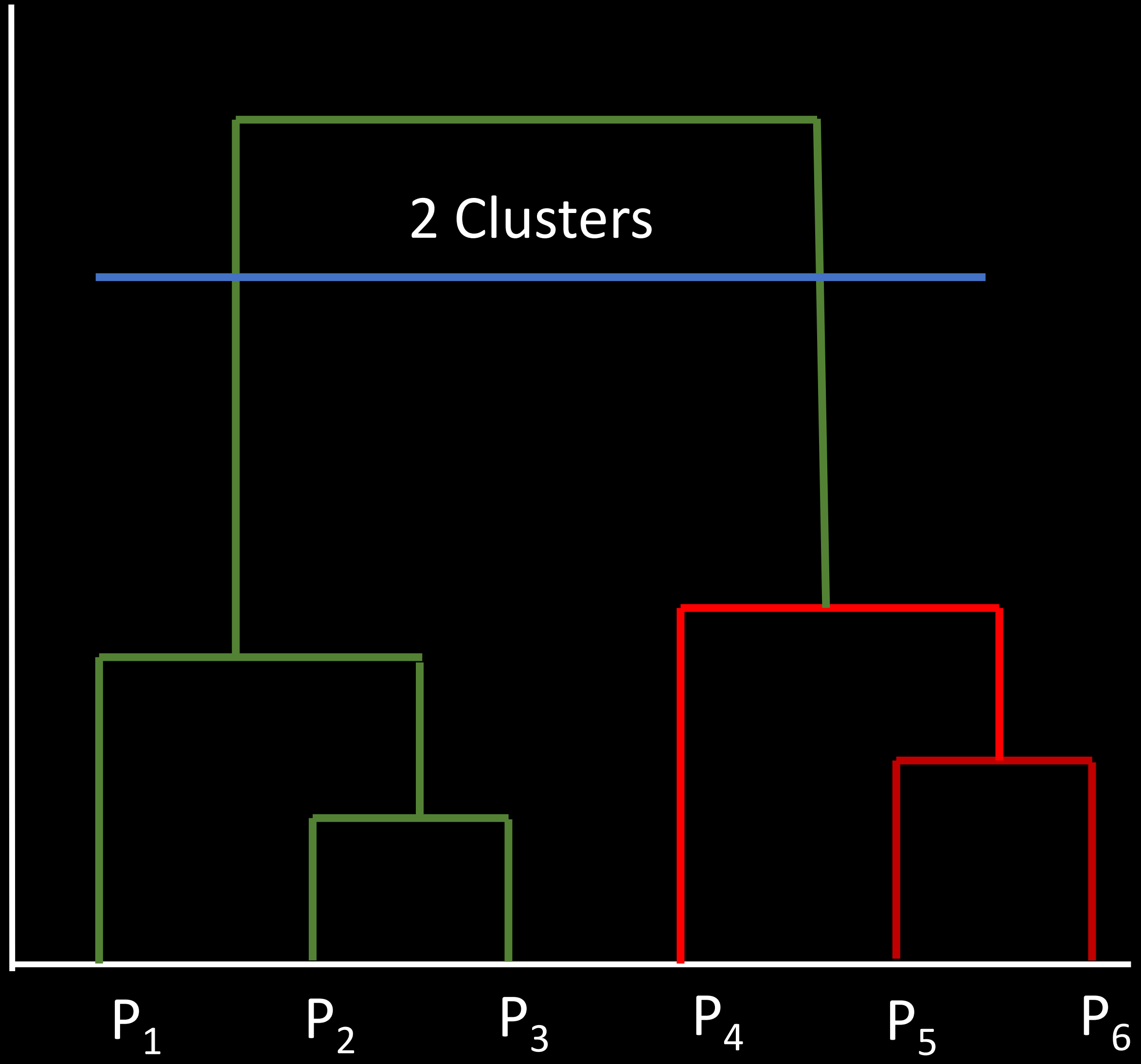
Hierarchical Clustering



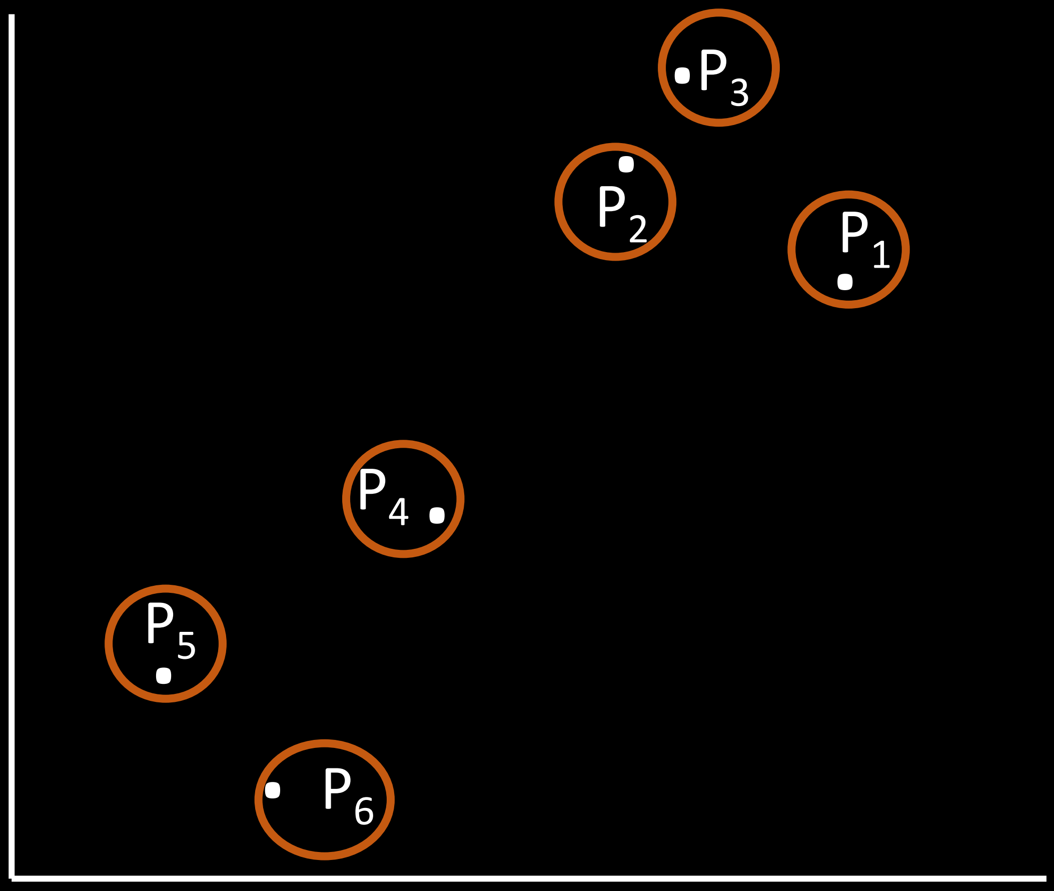


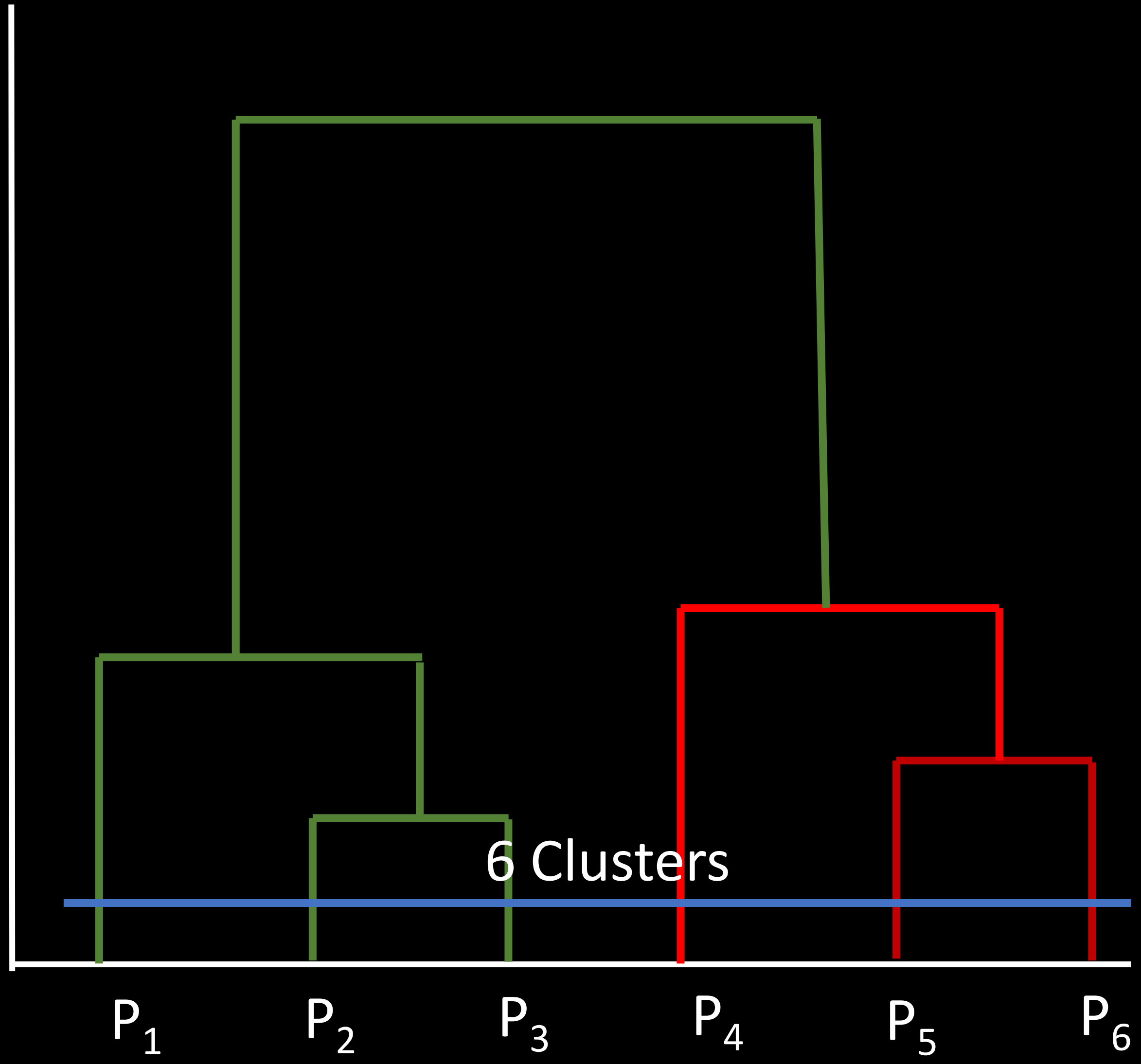
Hierarchical Clustering – Put it together



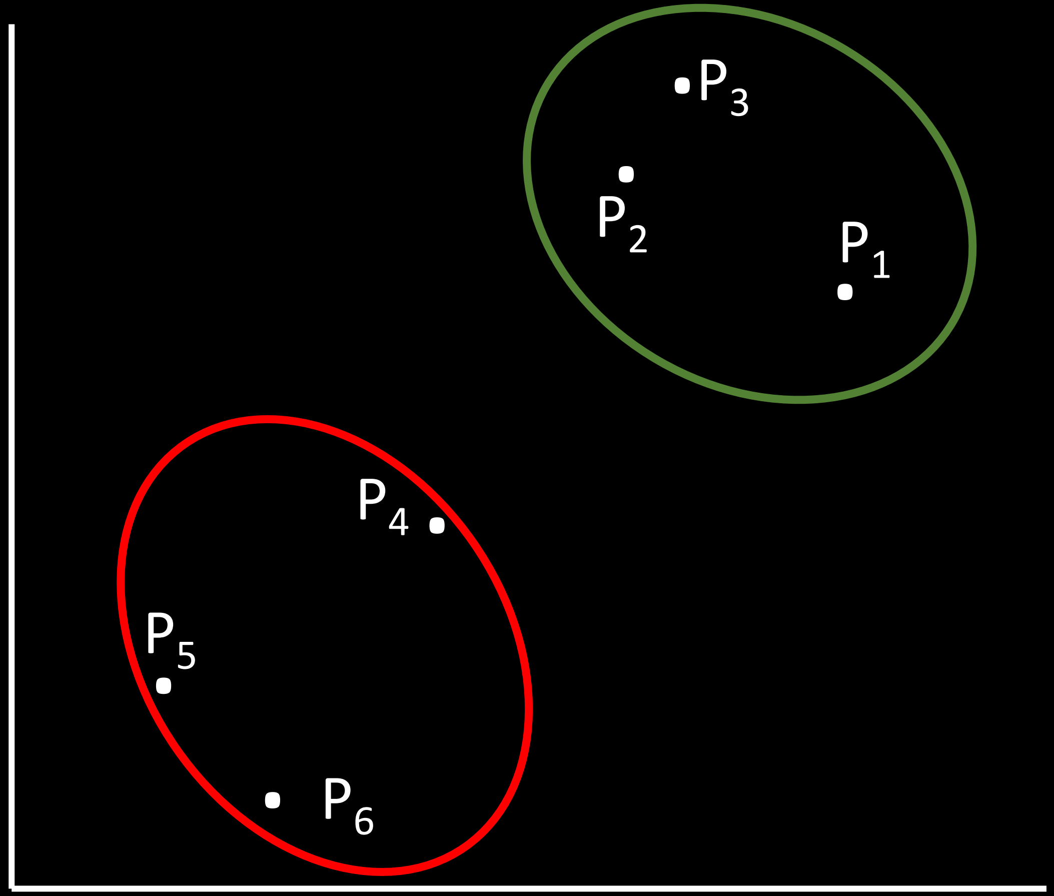


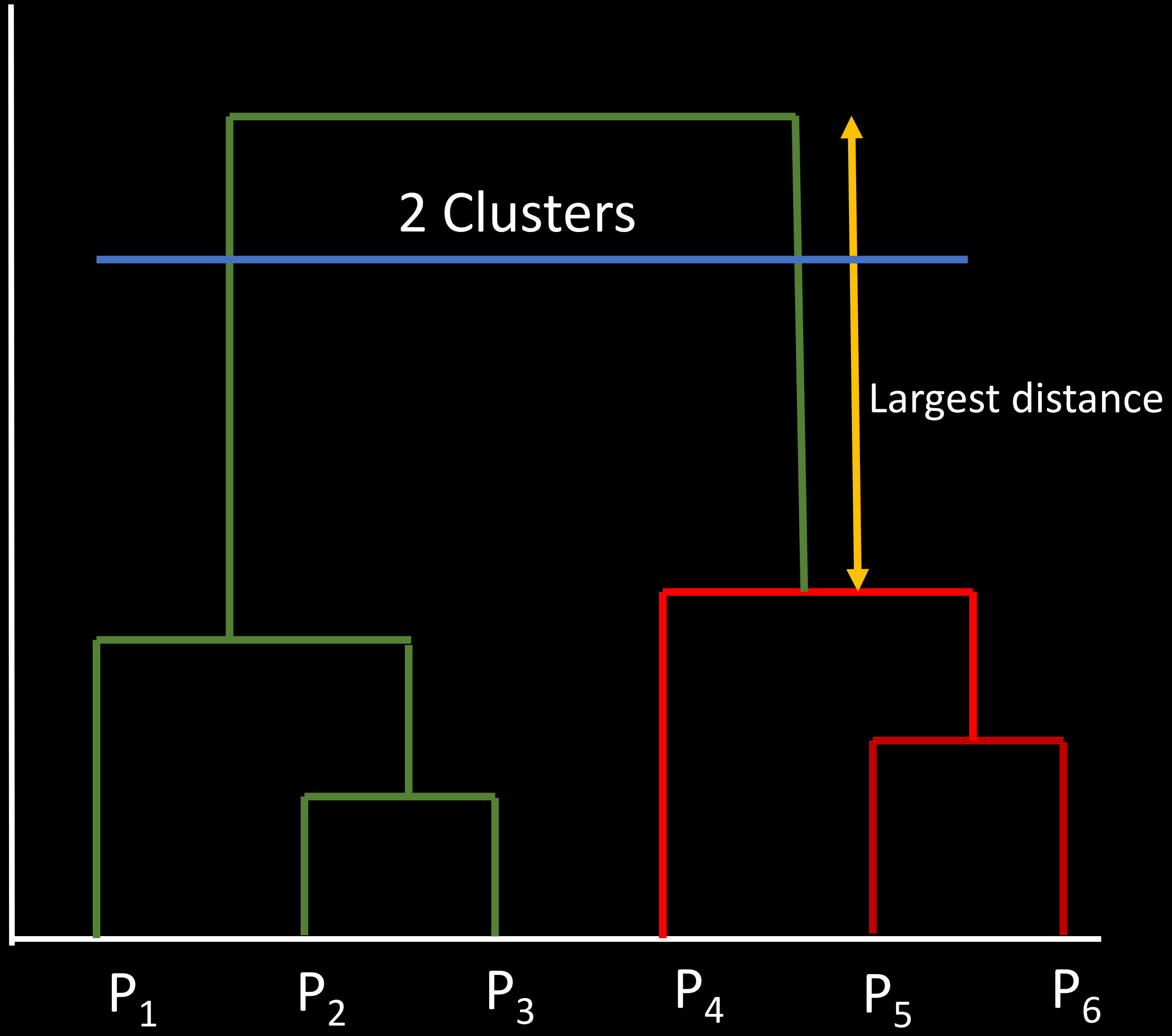
Hierarchical Clustering – Put it together





Hierarchical Clustering – Optimal



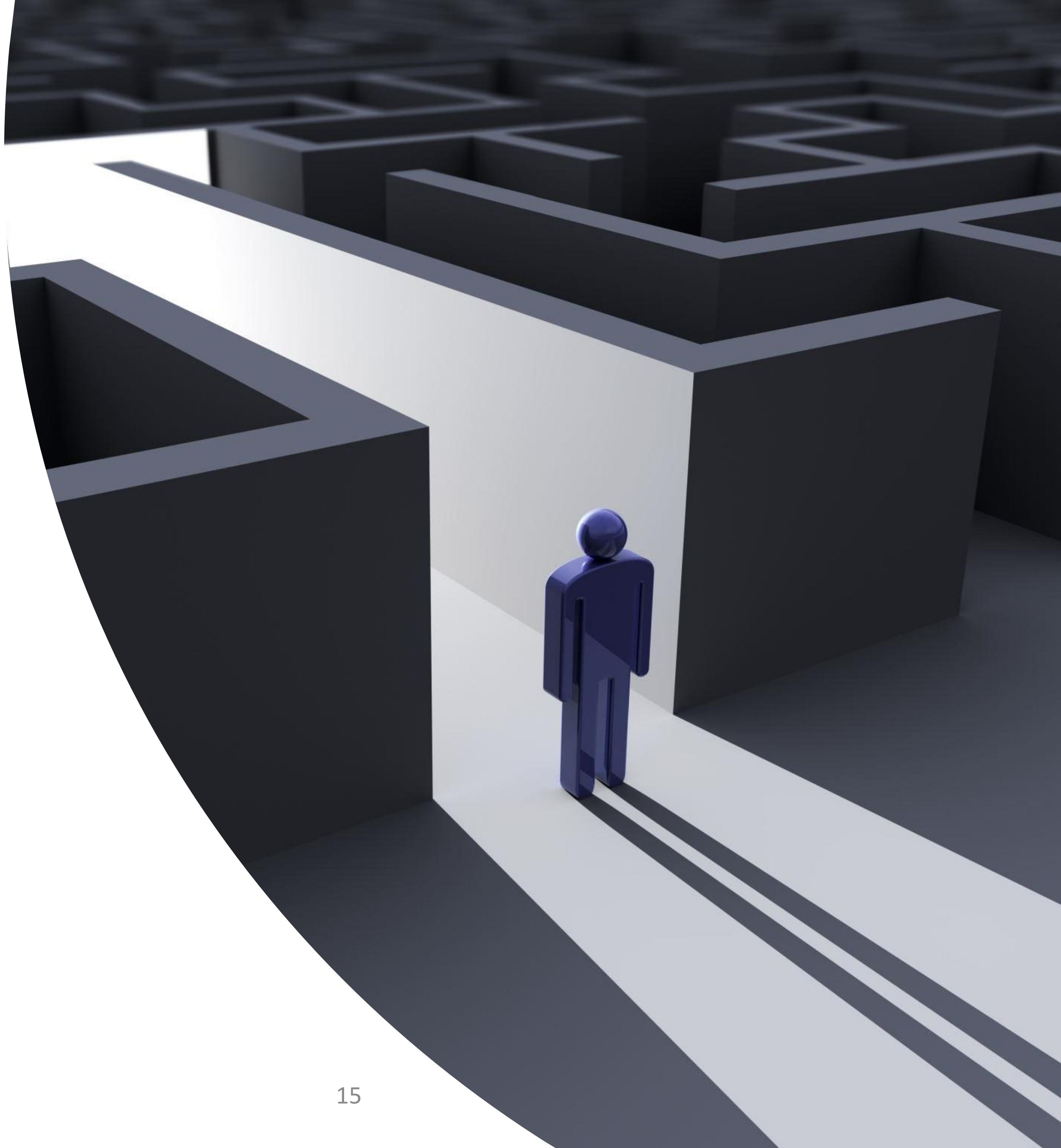


Clustering Comparison

Clustering	Pros	Cons
K-Means	Simple to understand, easily adaptable, works well on both small and large datasets, fast, efficient	Need to choose number of clusters
Hierarchical Clustering	Optimal number of clusters can be obtained from the model itself, practical visualization with dendrogram	Not appropriate for large datasets

Dimensionality Reduction

- Say we have a dataset with lots of features
- Can we understand the relationship between each of these features?
- But we want to focus on some of them
- That’s when Dimensionality Reduction can help
- Two ways of achieving Dimensionality Reduction
  - Feature Elimination
  - Feature Extraction



Feature Elimination

- Its easy, examine the data and drop features deemed not important
- Drawback is that there was no information gain from the dropped features e.g. dropping ‘Gender’ from the Social Advertising dataset
- However, by dropping the ‘User Id’ feature, there was no loss of information

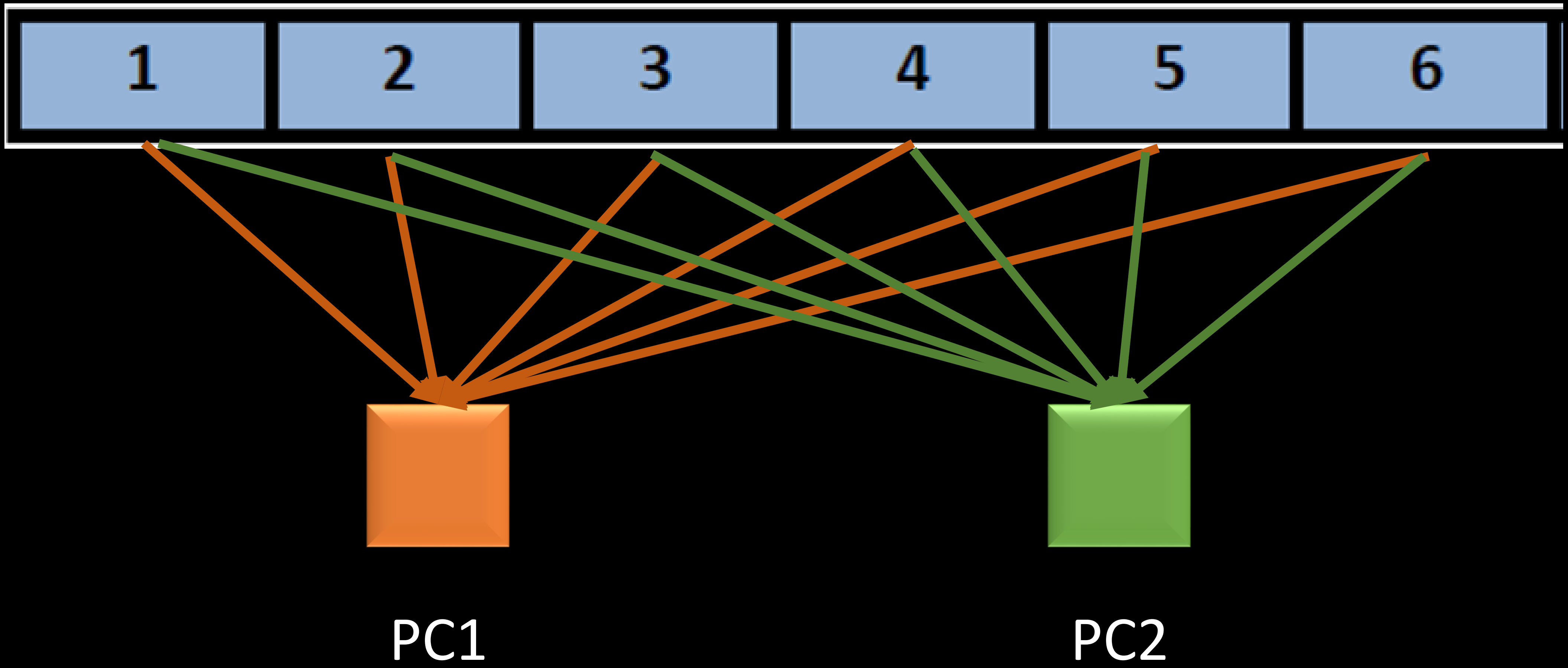
Feature Extraction

- Feature extraction does not have the drawback of Feature Elimination
- Create 10 new independent features from existing 10 features
- Each “new” feature is some combination of the 10 “old” features
- Order these “new” features by how well do they predict the dependent variable
- WHERE IS DEMSIONALITY REDUCTION?

Feature Extraction

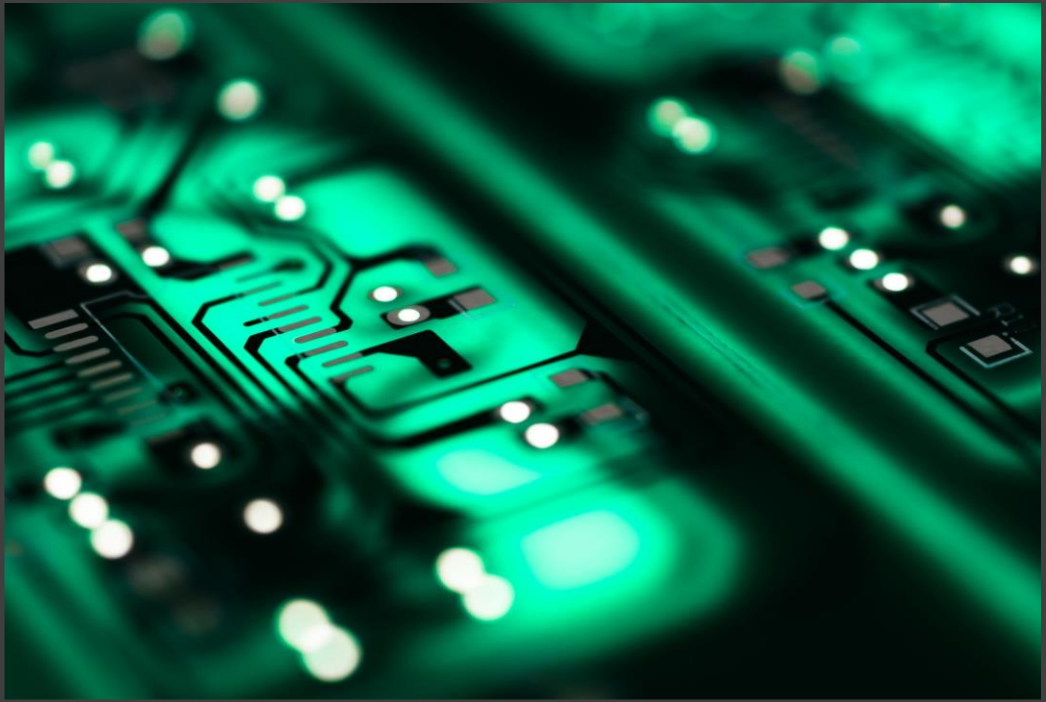
- Keep the most important “new” features and drop the least important features
- Since we have created each “new” features by using the “old” features – we have not lost much information from the “old” features
- Principle Component Analysis (PCA) helps with feature extraction

Feature Extraction



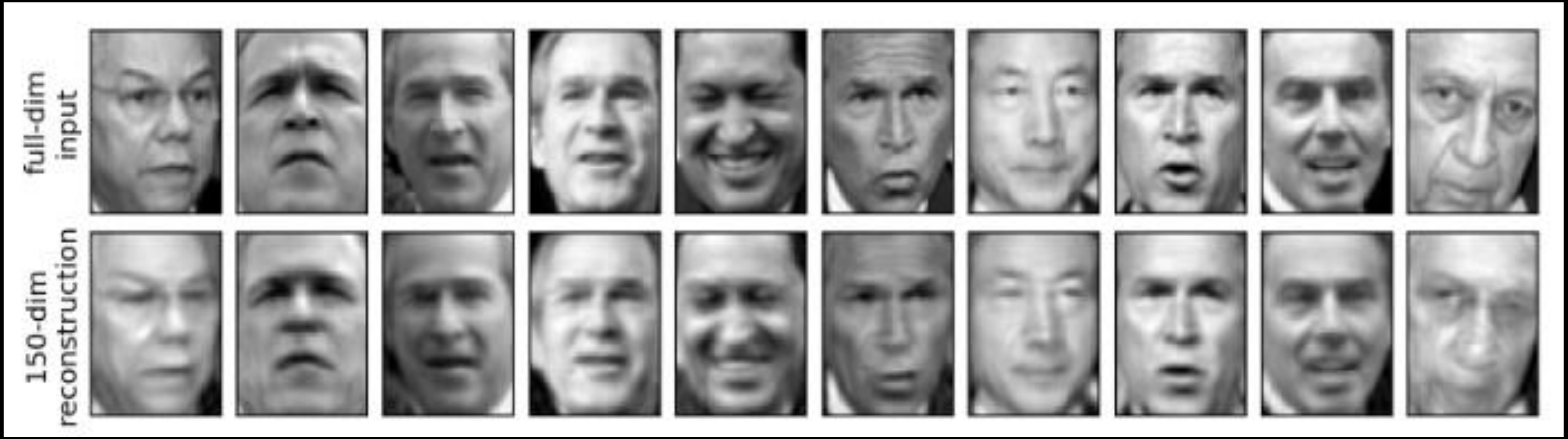
Principle Component Analysis

Principle Component Analysis

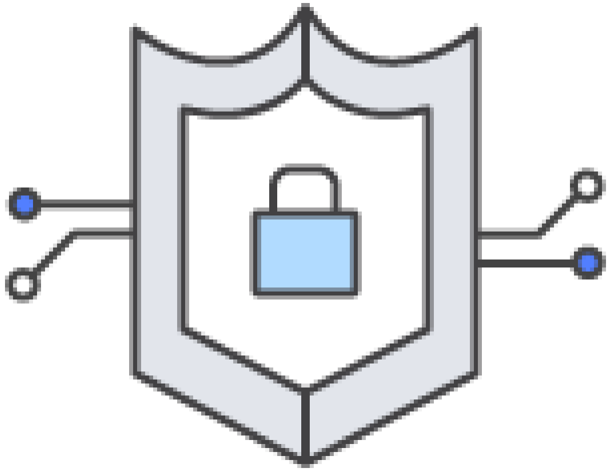


- PCA is a method of compressing a lot of data into something that captures the essence of the data
- "New" features are independent of one another
- When to use PCA?
  1. Want to remove features, but are unable to identify them
  2. Want to ensure that features are independent of one another
  3. You will not be able to interpret the "new" features

PCA Faces Example



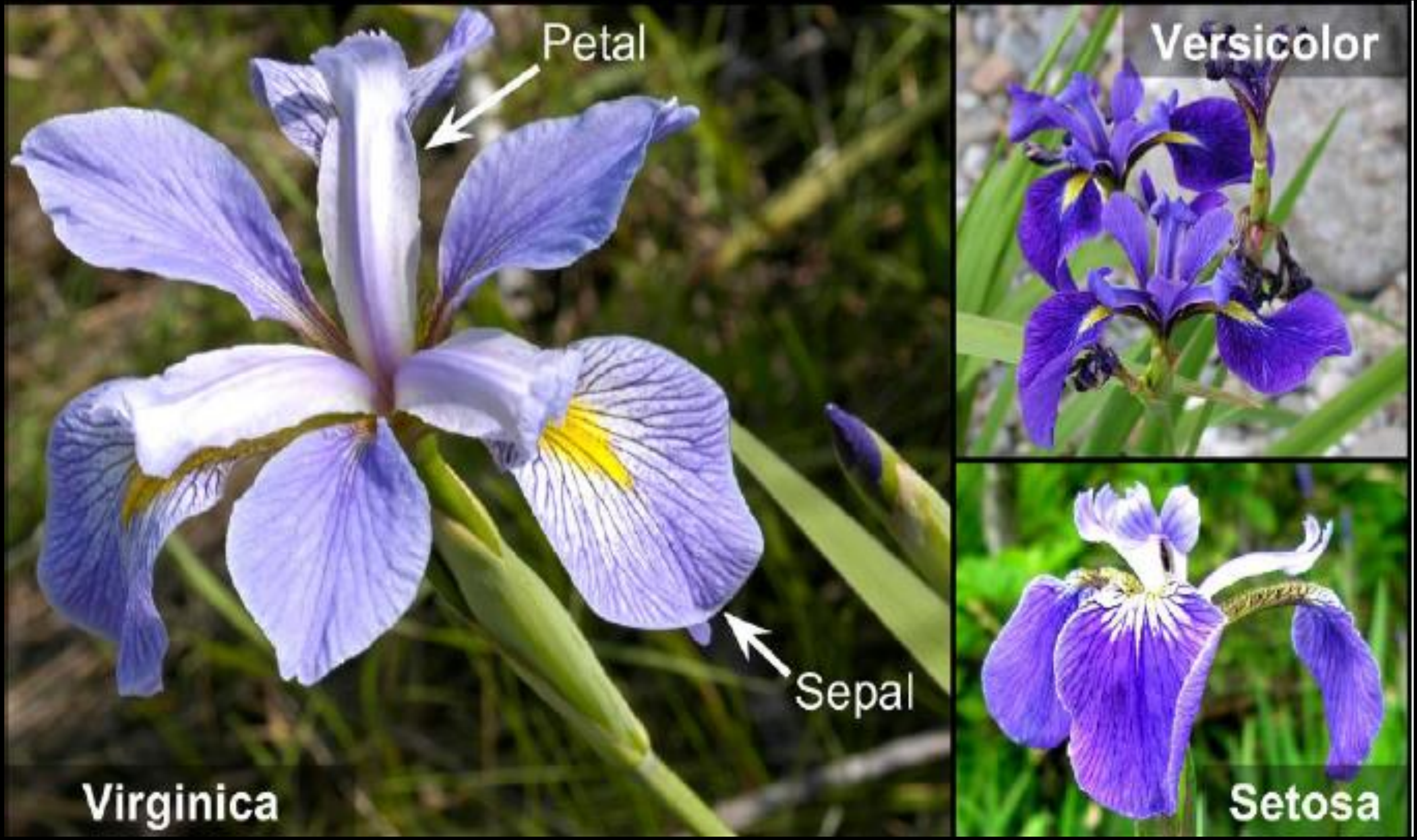
PCA Demo



- Open file ‘SparkExamples/PCA on Iris Spark’ using Jupyter
- Use PCA to create 2 features on the Iris dataset

Iris Dataset

- Sepal and Petal widths to identify types of iris flowers
- Attribute Information:
    - sepal length in cm
    - sepal width in cm
    - petal length in cm
    - petal width in cm
    - Class/type:
        - Iris Setosa
        - Iris Versicolor
        - Iris Virginica
- Number of Samples: 150



PCA Lab

Assignment

- Open notebook – “SparkLab/PCA Wine Lab”
- Use the wine dataset and use PCA to create new features (start with PCA as 2 features)
- Implement code (Marked in Red)
- Apply Logistic Regression to make predictions on the PCA features

Use case: Wine Customer Segmentation

- Dataset – Wine Dataset contains 13 features
- There are 3 Customer Segments (target variable)
- Dataset has following features:

Alcohol
Malic acid
Ash
Alcalinity of ash
Magnesium
Total phenols
Flavanoids
Nonflavanoid phenols
Proanthocyanins
Color intensity
Hue
OD280/OD315 of diluted wines
Proline
Customer_Segment

- Dataset - Wine.csv
- Solution:
  - Spark Lab Solution/PCA Wine Solution

Extra Credits

Extra Credits

- There are 3 datasets:
  1. Churn Modelling
  2. Credit Card Application
  3. Heart Disease
- Your Work
  - Pick anyone of these datasets
  - Apply the techniques that you have learnt related to Feature Cleaning, Feature selection, etc.
  - Choose 1 or more algorithms and make predictions
  - Tomorrow afternoon we will review it with the class
- Description of datasets in following slides

Use case: Financial Customer Churn Data

- Dataset – Customer information with a financial institution
- If doing classification with SVM then it can be applied to most of the datasets where logistic regression is used
- Dataset has following features:

RowNumber: Dataset row number
CustomerId: Customer Id
Surname: Last name of the person
CreditScore: Credit Score of the person
Geography: Country of residence
Gender: Person's Gender
AGE: Age of the person
Tenure: How long has the person owned the card
Balance: Outstanding balance
NumOfProducts: Number of products owned by the person with company
HasCrCard: Person has credit card
IsActiveMember: Is the person active member of the company
EstimatedSalary: Estimated salary of the person
Exited: Did the person stay or leave

- Dataset - Churn_Modelling.csv

Use Case: Credit Card Application

- Based on 15 features (not named), predict whether a new customer's credit card application should be approved
- Predicting a "class" – 1 or 0
- Dataset - Credit_Card_Applications.csv

Use Case: Heart Disease

- Based on 13 features predict if a patient will have heart disease
- Features include age, gender, chest pain, resting blood pressure, cholesterol, fasting blood sugar, resting ECG, max heart rate, exercise induced angina, etc.
- Predicting a "target" – 1 or 0
- Dataset - heart.csv

Feature Extraction and Elimination

Persistence

Persistence

Persistence can be applied to Models, Pipelines, Scalers

Use the save and load generic functions

Can be saved in one language and loaded in another language

- Save the model using python
- In production load the model in Java code

In production must apply the same pipelines, scalers that were used on the training and test dataset

- New data coming in must be subject to the same processing

Must use appropriate “Model” class during load

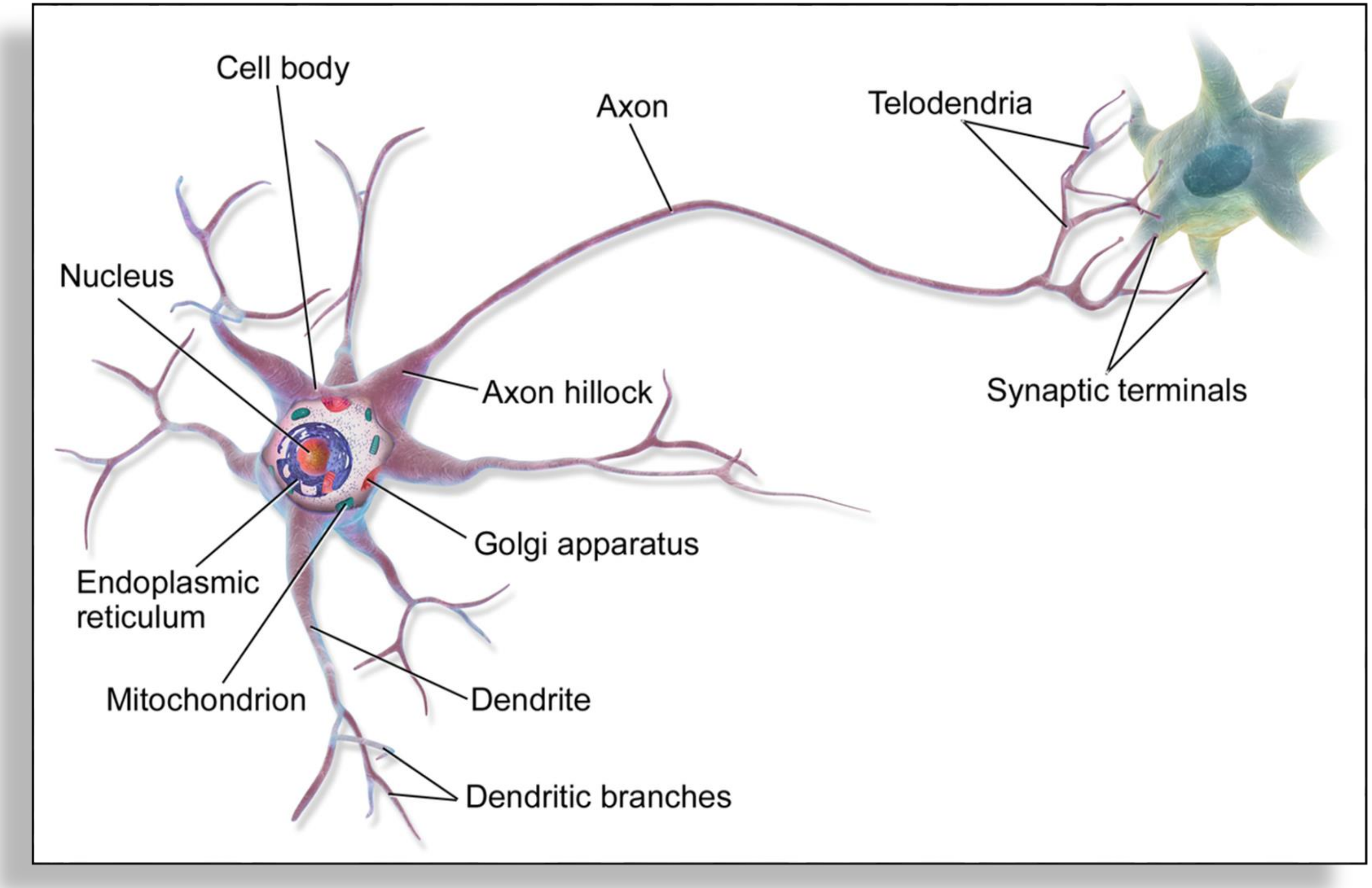
- StandardScalerModel to load a StandardScaler model that was saved

Pipeline and Model Persistence

- Open file ‘SparkExamples/Spark-Model-Persistence’ using Jupyter
- Use the same “agent.csv” used in project
- Save the pipeline and the Logistic Regression model that were created
- Open file ‘SparkExamples/LoadPipeLineModel’ using Jupyter
- Will use “agent-0.csv” – new data
- Will apply pipeline and model to make predictions
- Write results to .csv file

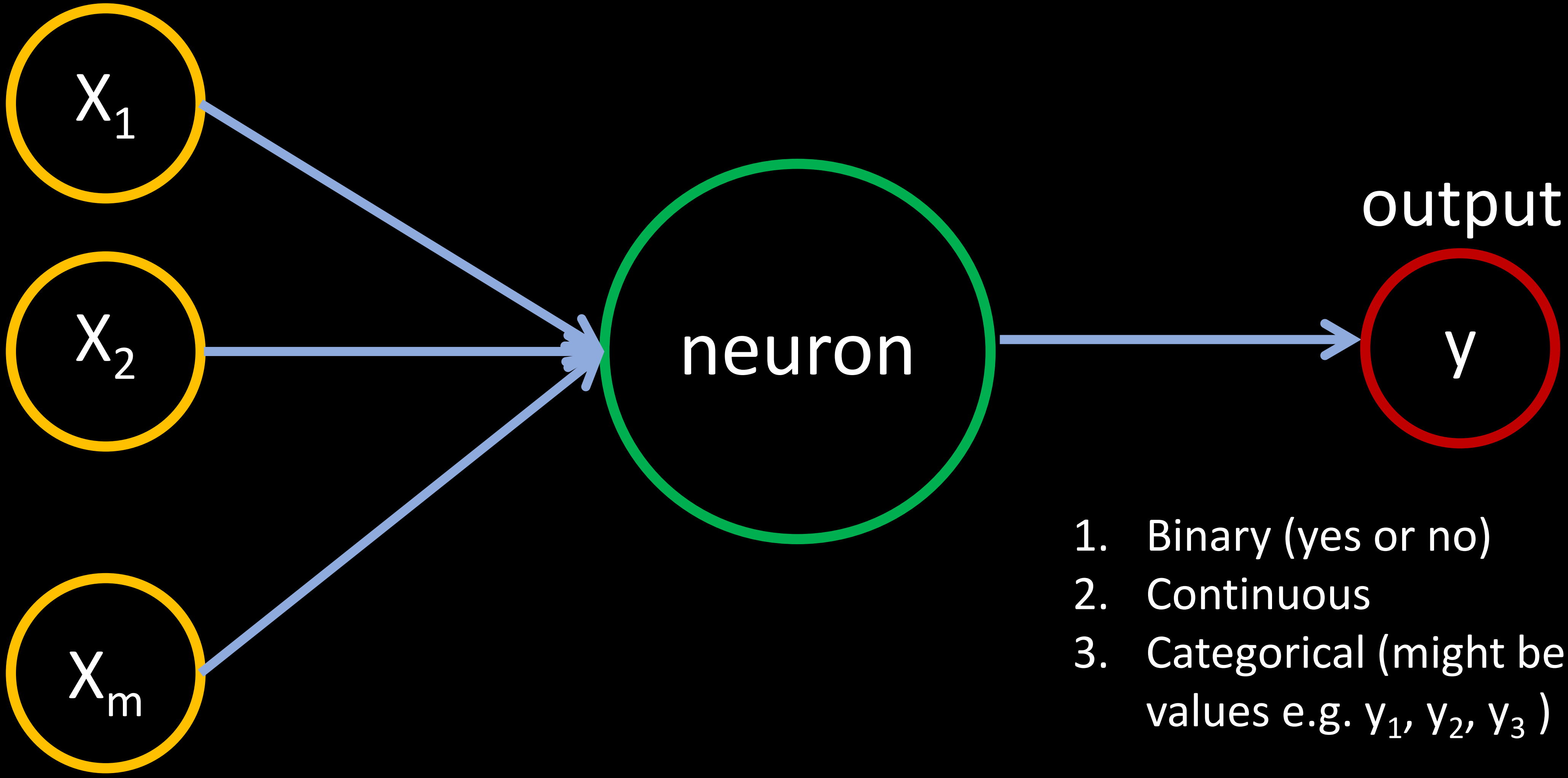
Neural Networks

Neuron



From Wikipedia

What are Neural Networks?



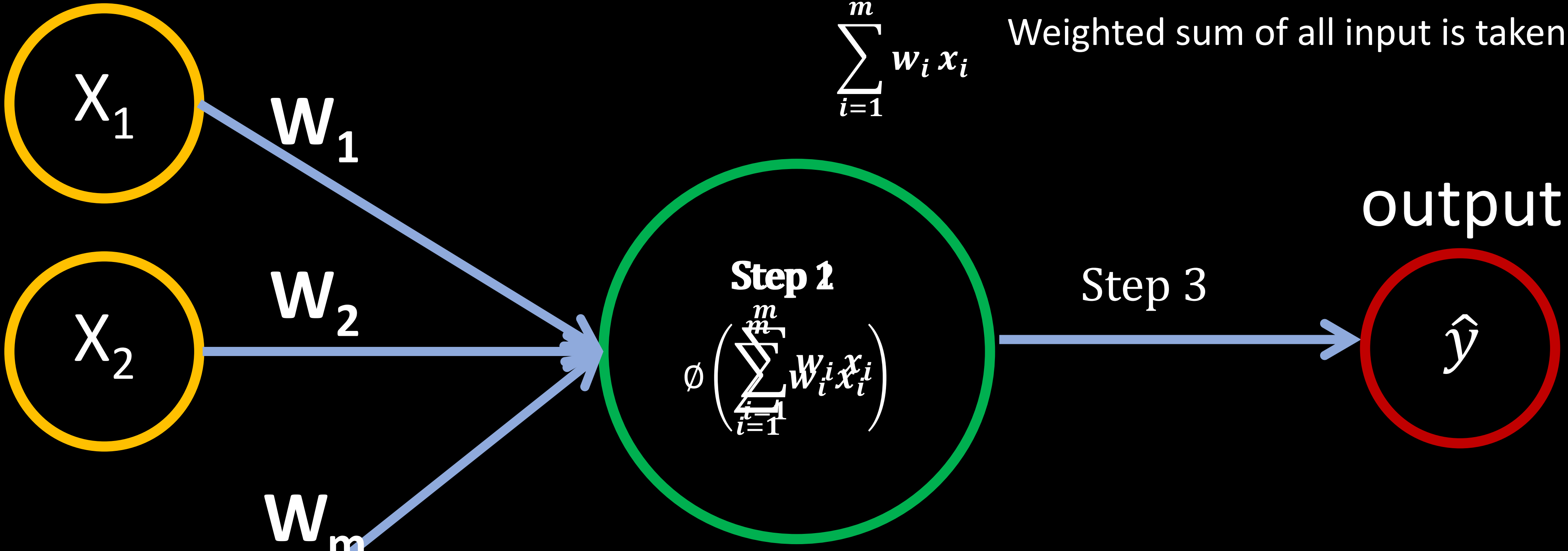
1. Binary (yes or no)
2. Continuous
3. Categorical (might be multiple values e.g. y_1, y_2, y_3)

Neural Network – in action

Step 1

$$\sum_{i=1}^{m} w_i\, x_i$$

Weighted sum of all input is taken

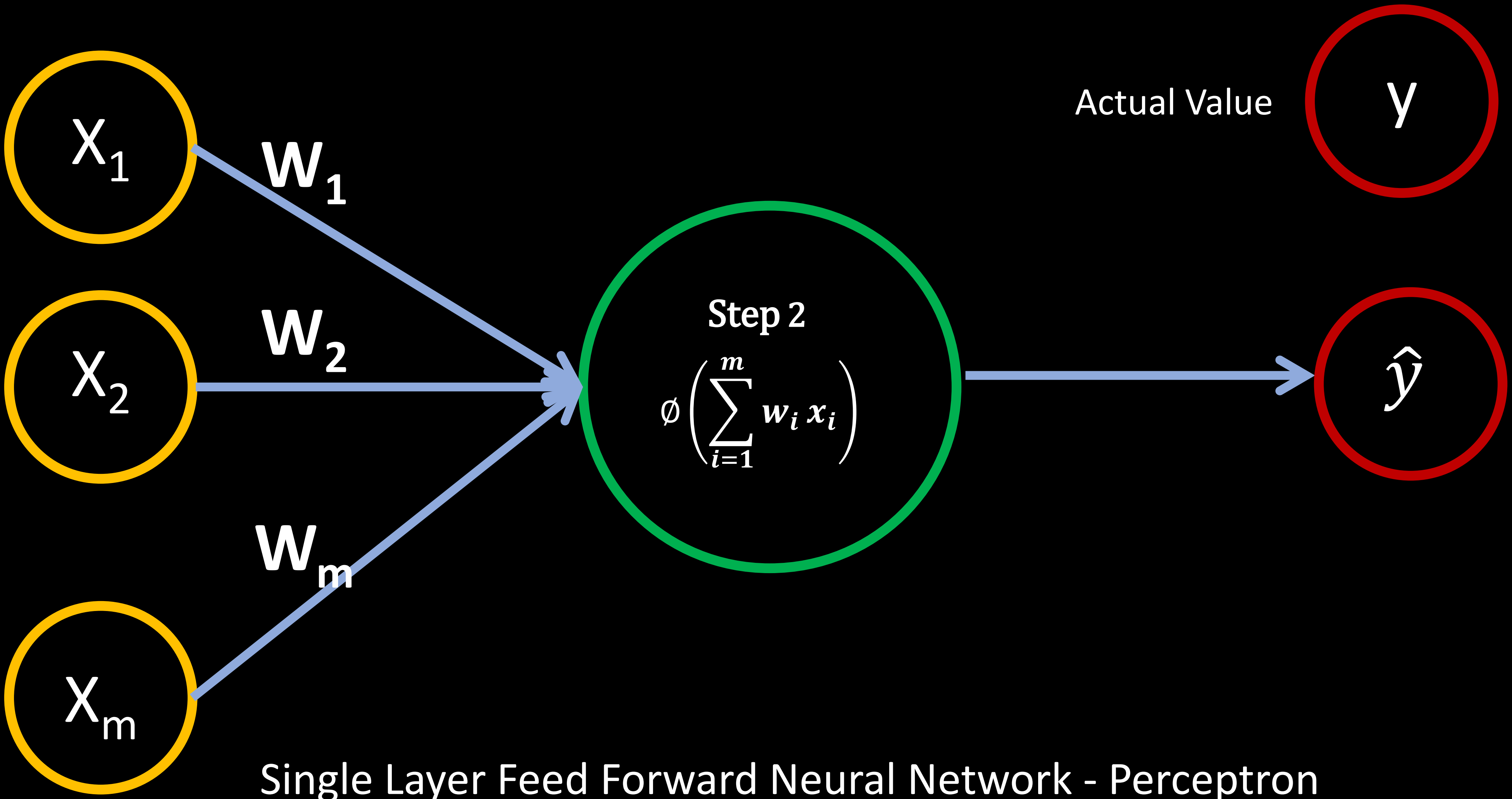


Step 2

$$\emptyset\left(\sum_{i=1}^{m} w_i\, x_i\right)$$

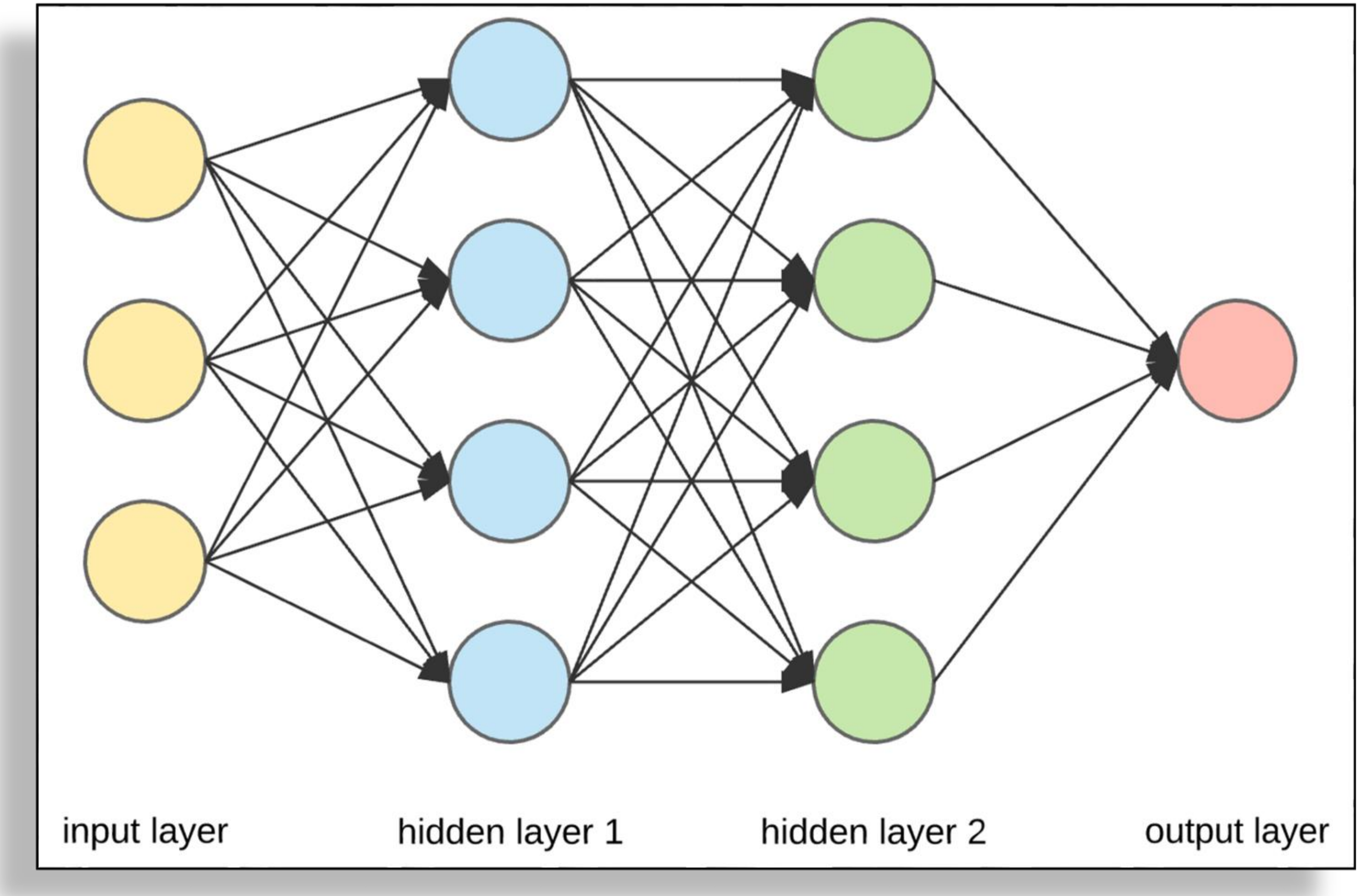
Activation function is applied to the sum

Single Layer Perceptron

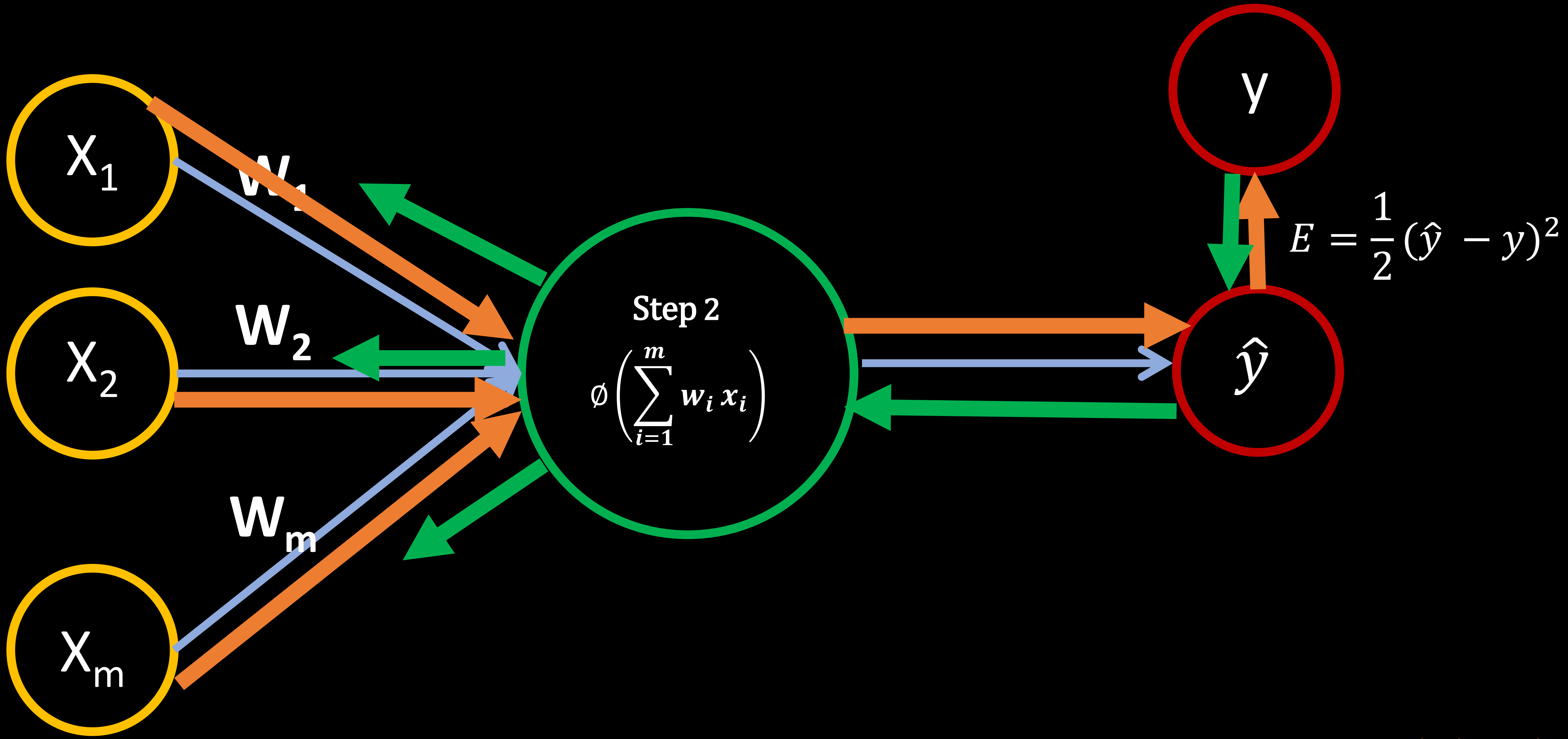


Single Layer Feed Forward Neural Network - Perceptron

Multi Layer Perceptron

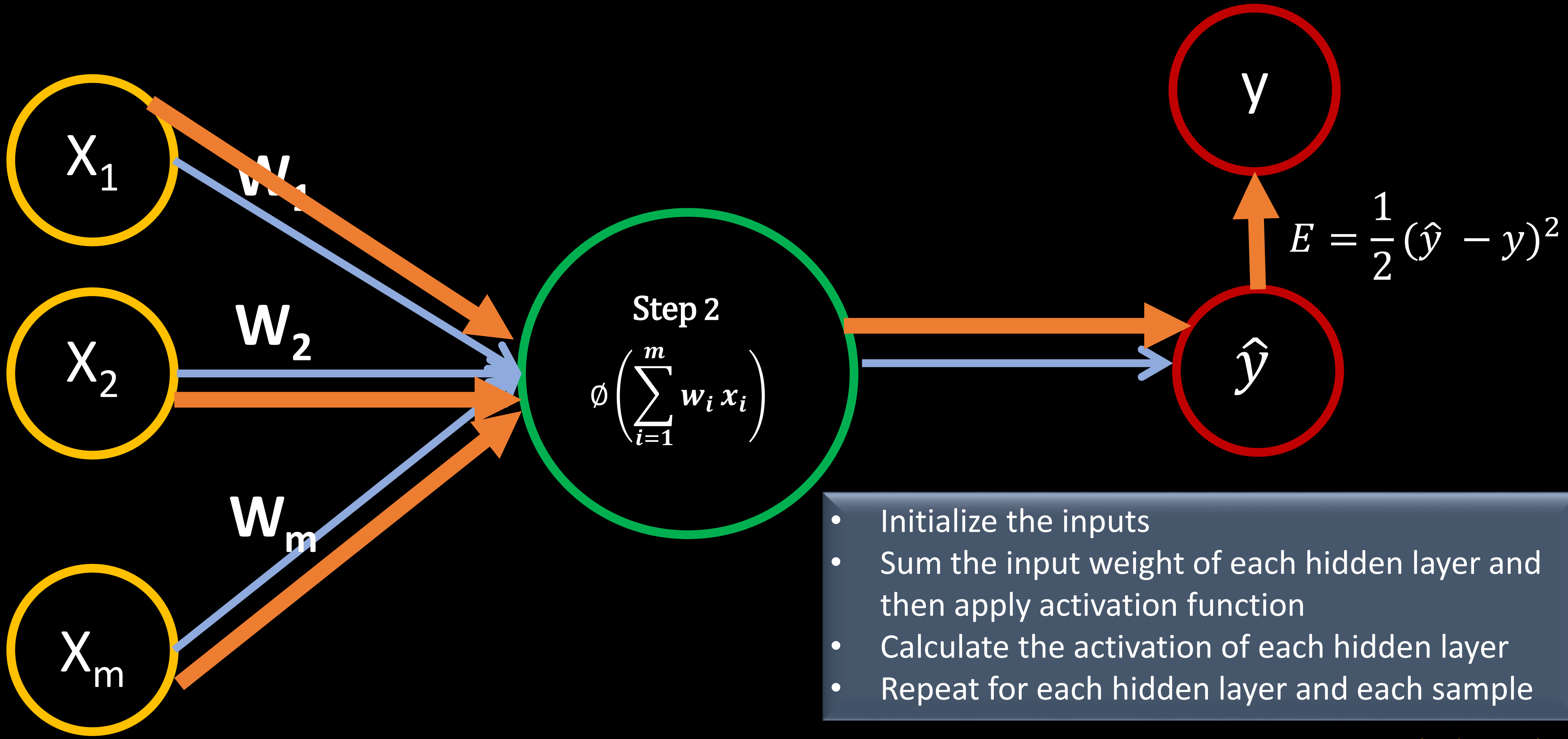


Neural Networks - Learning



$$E = \frac{1}{2}(\hat{y} - y)^2$$

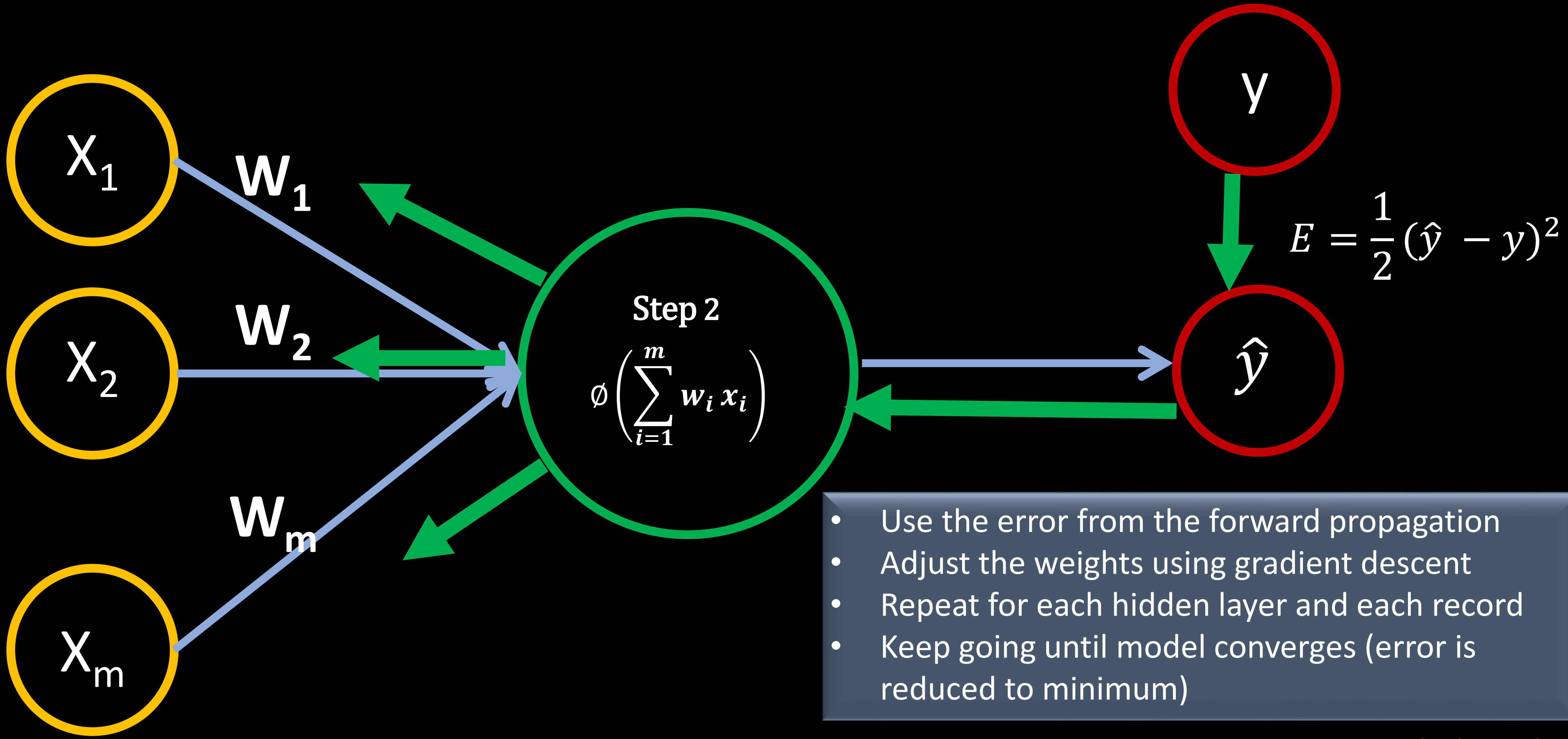
Neural Networks – Feed Forward



X1
X2
Xm
W1
W2
Wm
Step 2
$\emptyset\left(\sum_{i=1}^{m} w_i\, x_i\right)$
$\hat{y}$
y
$E = \frac{1}{2}(\hat{y} - y)^2$

- Initialize the inputs
- Sum the input weight of each hidden layer and then apply activation function
- Calculate the activation of each hidden layer
- Repeat for each hidden layer and each sample

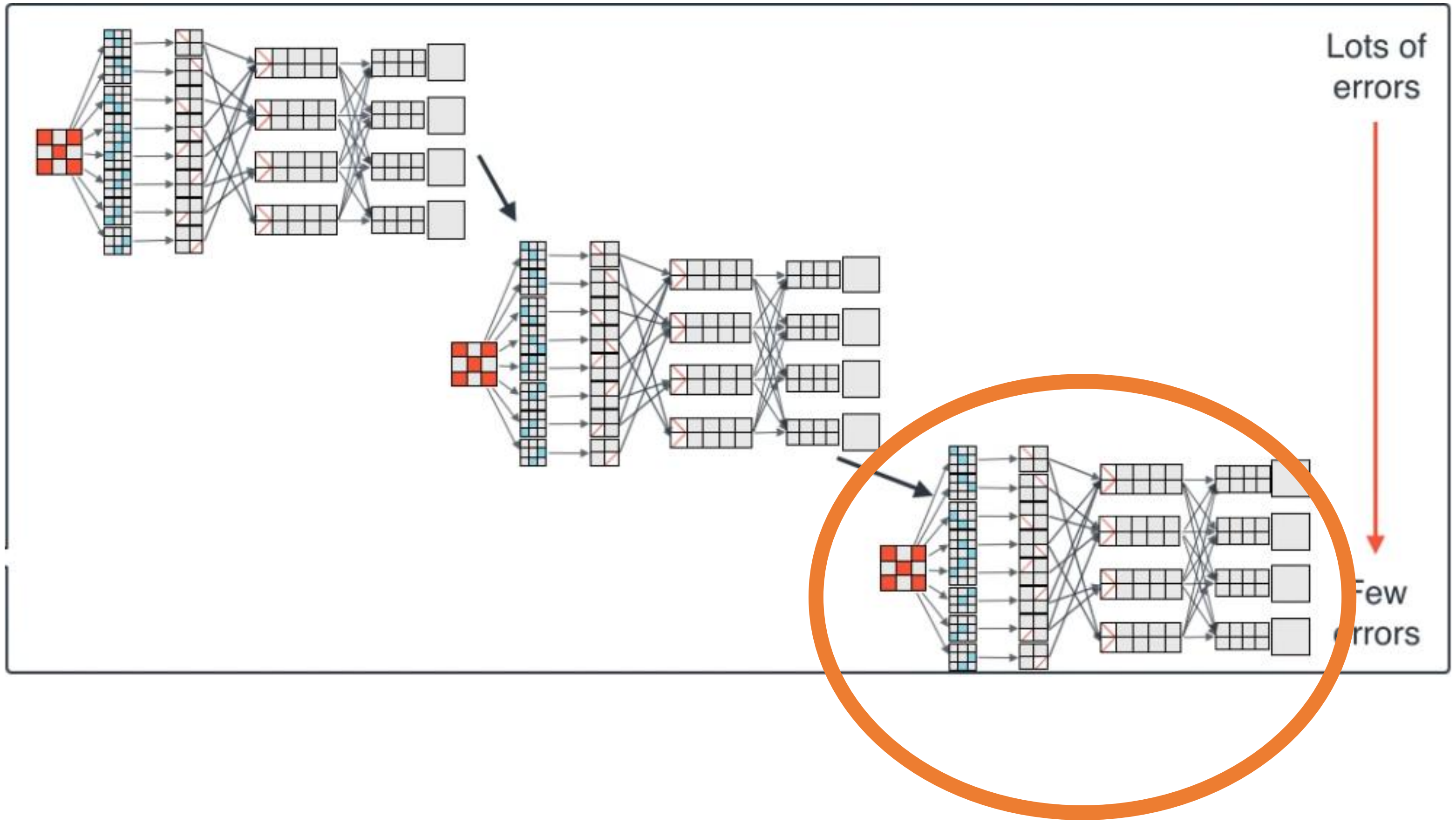
Neural Networks – Back Propagation

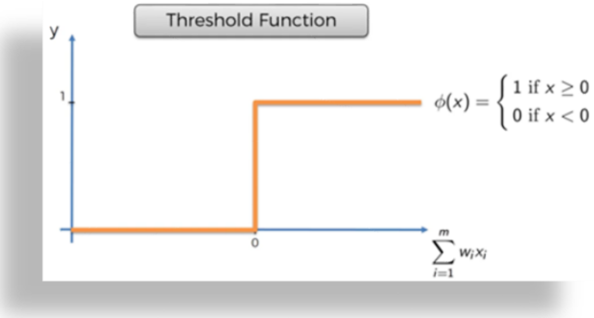


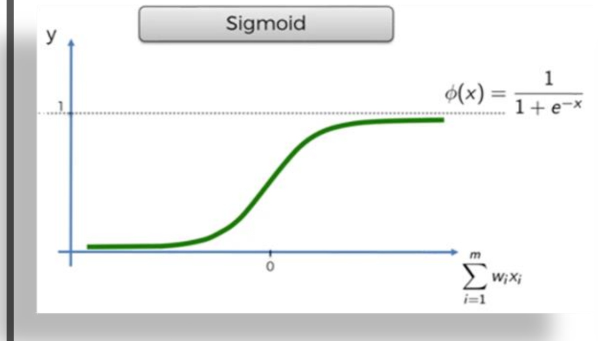
X_1
X_2
X_m
W_1
W_2
W_m
Step 2
$\emptyset\left(\sum_{i=1}^{m} w_i\, x_i\right)$
y
$\hat{y}$
$E = \frac{1}{2}(\hat{y} - y)^2$

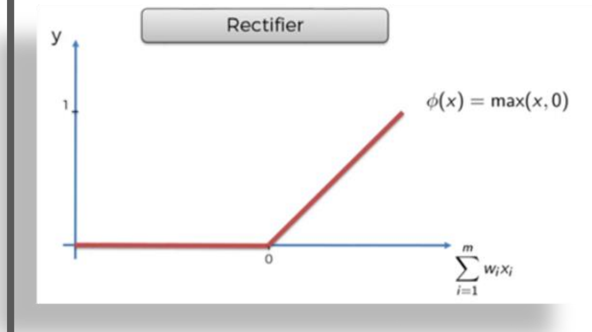
- Use the error from the forward propagation
- Adjust the weights using gradient descent
- Repeat for each hidden layer and each record
- Keep going until model converges (error is reduced to minimum)

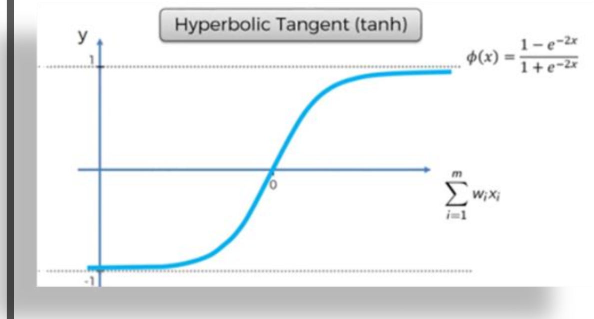
Gradient Descent











Neural Networks – Activation Functions

Neural Networks – MLP Demo

Neural Networks - MLP Demo

- Open file ‘SparkExamples/MLP Classifier Iris’ using Jupyter
- Will use Multi Layer Perceptron Classifier to predict different types of flowers from Iris dataset
- Important hyper parameter is defining the layers

Neural Networks - Hands-on

Assignment

- Open notebook – “SparkLab/MLP Classifier Wine”
- Use wine dataset
- Implement code (Marked in Red)
- Apply Multilayer Perceptron Classifier to classify the customer segments

Use case: Wine Customer Segmentation

- Dataset – Wine Dataset contains 13 features
- There are 3 Customer Segments (target variable)
- Dataset has following features:

Alcohol
Malic acid
Ash
Alcalinity of ash
Magnesium
Total phenols
Flavanoids
Nonflavanoid phenols
Proanthocyanins
Color intensity
Hue
OD280/OD315 of diluted wines
Proline
Customer_Segment

- Dataset - Wine.csv
- Spark Lab/MLP Classifier Wine
- Solution:
  - Spark Lab Solution/MLP Classifier Wine Solution

Spark and Machine Learning Environment Set up

Setting Up Environment

- Following slides contain instructions to install Spark and Machine Learning environment on MAC and Windows
- NOTE – Spark will run in standalone mode (not distributed)
- Consists of 3 steps:
    1. Install Anaconda (Jupyter notebook)
    2. Install Spark
    3. Install “findSpark” package

Setting Up Environment - Anaconda

- https://www.anaconda.com/distribution/ - use the link to download the distribution for your machine (Windows or MAC).
- https://docs.anaconda.com/anaconda/install/ - follow the instructions for the machine type (“Installing on Windows”, or “Installing on macOS”)

Setting Up Environment – Spark on MAC

- Follow the instructions in the following link to install Spark on MAC
    - JAVA, Spark, Scala
    - Do not install python since it is already installed by Anaconda in the pervious step
    - Make sure to set up environmental variables in the “bashrc” file as defined in the instructions

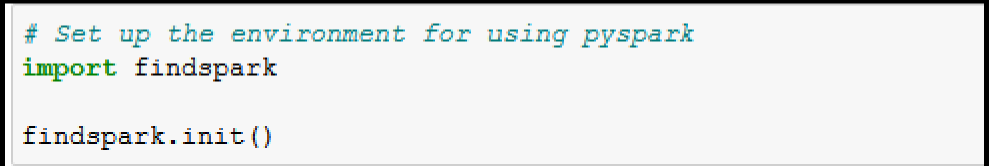
- https://www.tutorialkart.com/apache-spark/how-to-install-spark-on-mac-os/

Setting Up Environment – Spark on Windows

- Follow the instructions in the following link to install Spark on Windows
  - Spark, JAVA and Winutils will be installed
  - Do not install python since it is already installed by Anaconda in the pervious step
  - Make sure to set up environmental variables correctly

- https://www.knowledgehut.com/blog/big-data/how-to-install-apache-spark-on-windows

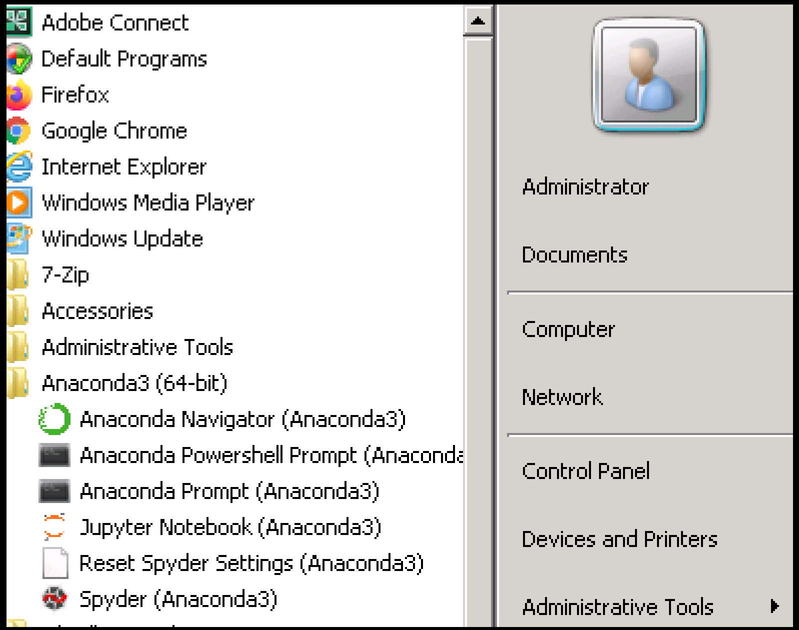
“findSpark” installation



```
# Set up the environment for using pyspark
import findspark

findspark.init()
```

- Need to install “findSpark” utility to set up environment variables correctly so that Jupyter notebook can execute Spark code
- Open “Anaconda Prompt”
- At the prompt type:
  - conda install –c conda-forge findspark



Jupyter Notebook with Spark

Information in the .bashrc file for Linux
Spark runs locally

```
function snotebook()
{
SPARK_HOME=/usr/local/spark/spark-2.4.2-bin-hadoop2.7
export PATH=$SPARK_HOME/bin:$PATH

export PYSPARK_DRIVER_PYTHON=jupyter
export PYSPARK_DRIVER_PYTHON_OPTS='notebook'

$SPARK_HOME/bin/pyspark --master local[2]
}
```

Spark ML Industry Uses Cases

Spark Industry Usage

Usage	Purpose
91%	Use it because of performance gains over MapReduce
77%	Ease of use
71%	Ease of deployment
64%	Leverage advanced analytics
52%	For real-time streaming

Un-named Financial Institution

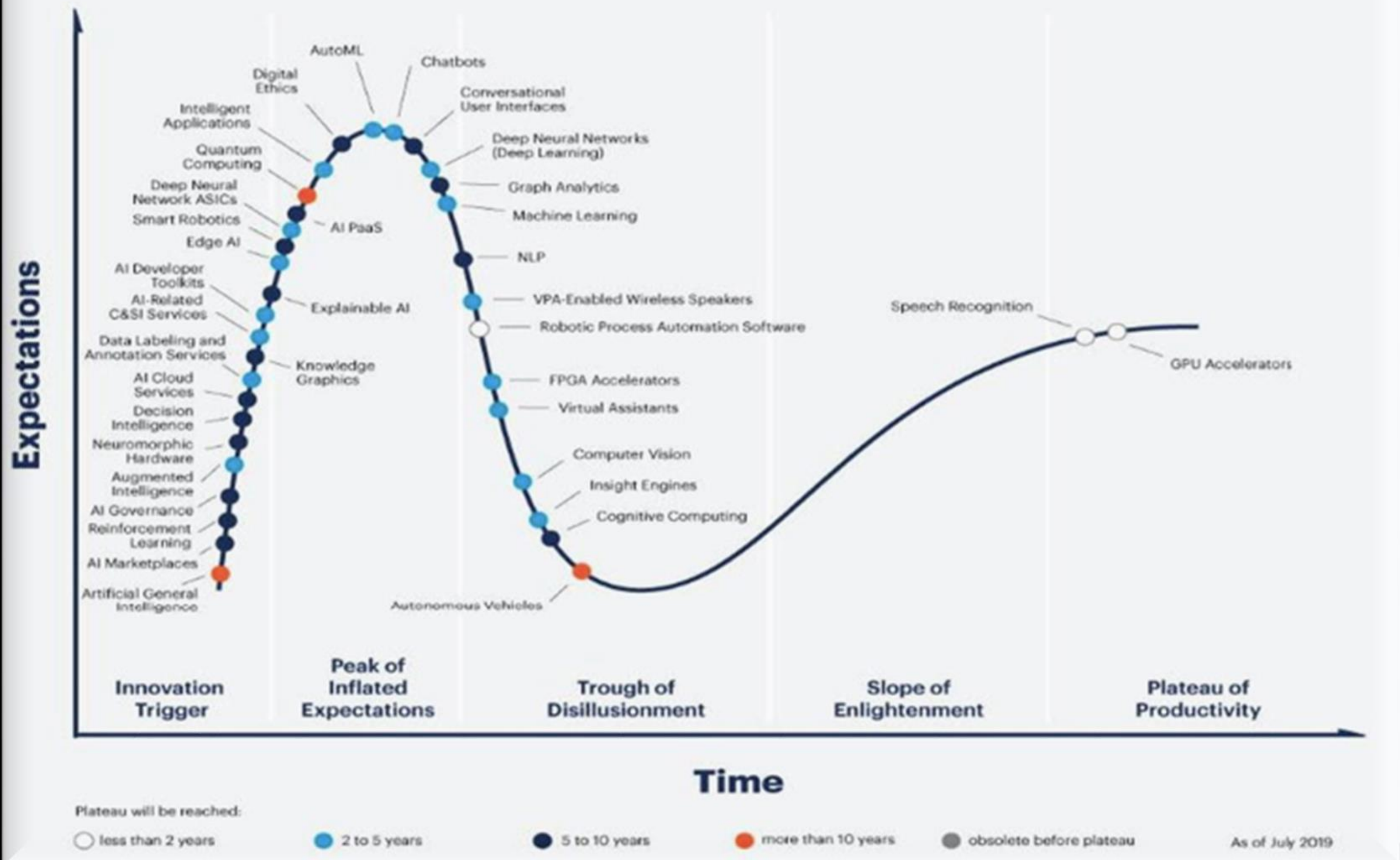
- Provides retail banking and brokerage operations
- Have reduced customer churn by 25%
- Want 360-degree view of their clients – corporations or individuals
- Spark is used as an underlying layer for creating consolidated view of the customer
- Uses machine learning with Spark to automate analytics

- Biggest e-commerce giant in the world
- Runs one of the largest Spark job in the world – analyze hundreds of petabytes of data
- Millions of vendors and users interact with their platform, each generating a complex graph on which the company uses Spark ML

- Uses Spark for personalizing news and targeted advertising
- 15,000 lines of C++ ML code
- With Scala ML, they did it in 120 lines of code
- Ready for production in 30 minutes of training on one hundred million datasets

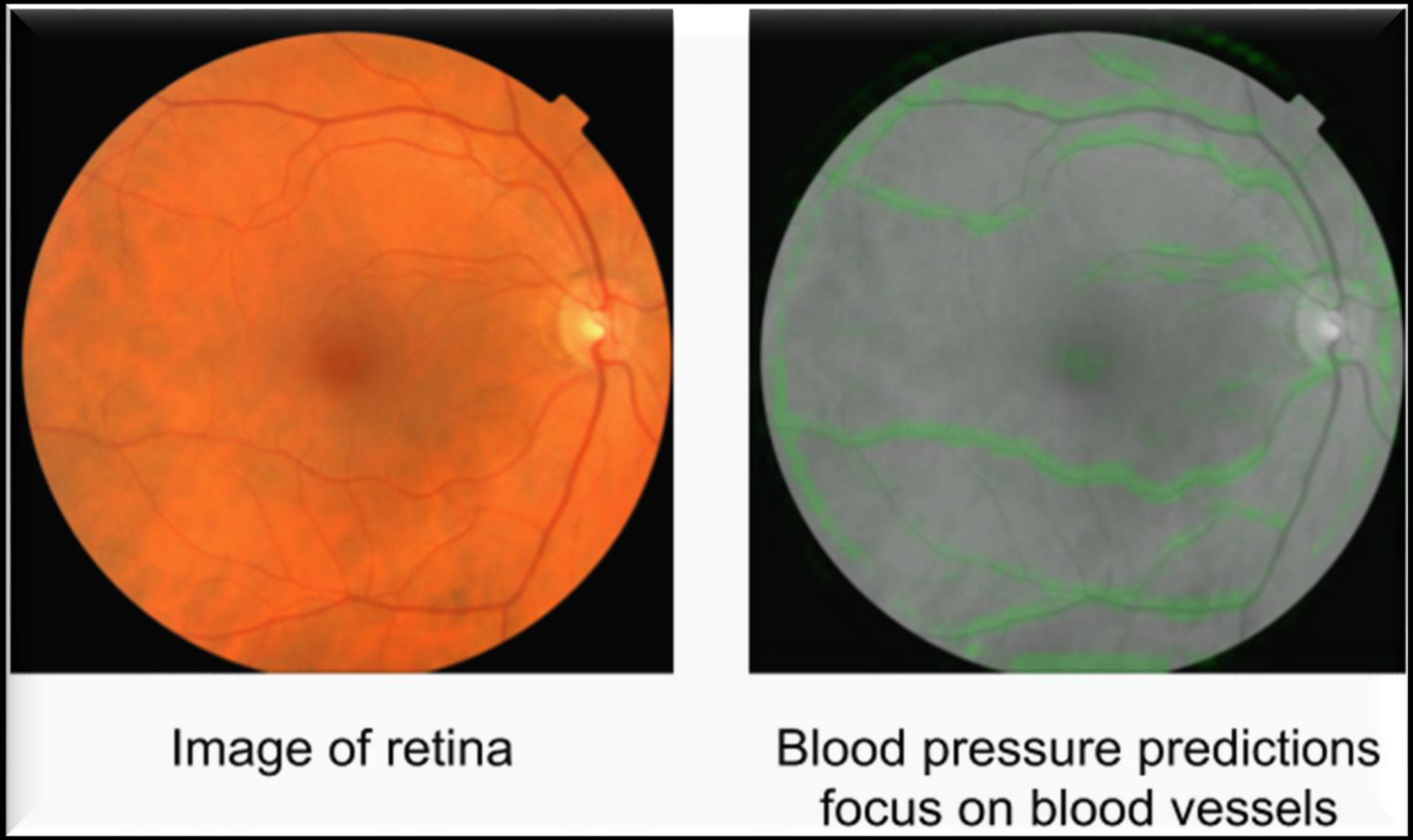
- Online real time reservation service with 31,000 restaurants and 15 million diners per month
- Uses Spark to train its recommendation algorithms and NLP for restaurant reviews
- Has reduced the run time of its ML algorithm from few weeks to just few hours

Gartner Hype Cycle for Artificial Intelligence, 2019



Health Care

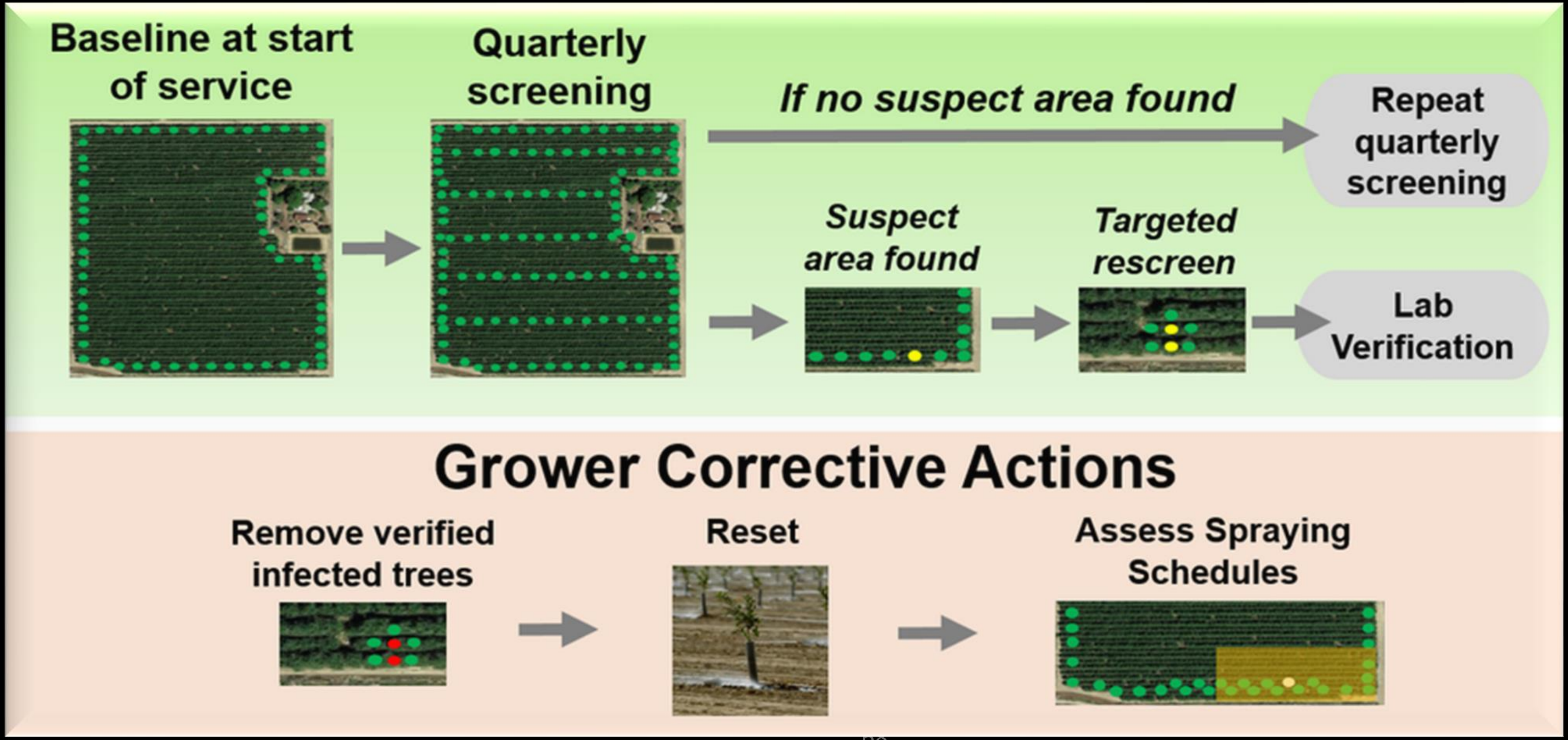
Reference
https://ai.googleblog.com/2018/02/assessing-cardiovascular-risk-factors.html



Agriculture – Disease Screening & Monitoring

Reference

http://www.croptix.solutions/solution.html



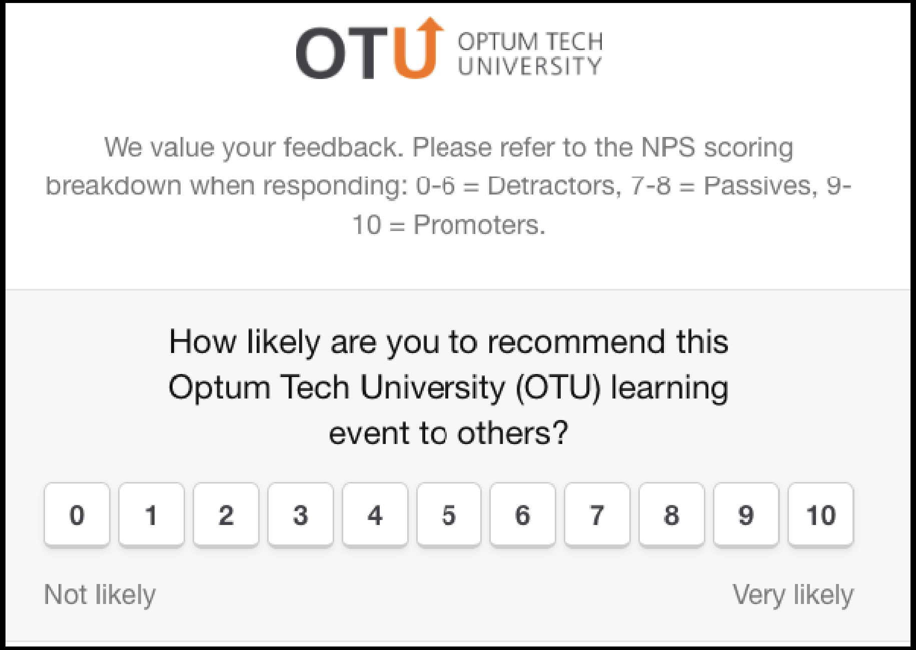
Baseline at start of service
Quarterly screening
If no suspect area found
Repeat quarterly screening
Suspect area found
Targeted rescreen
Lab Verification
Grower Corrective Actions
Remove verified infected trees
Reset
Assess Spraying Schedules

Ride Share – Uber and Lyft

- Machine Learning
  - Supply & Demand
  - Route Optimization
  - Rush Hour Pool
  - Uber Eats

Meet on the corner of Wilshire Blvd and State St

https://delighted.com/e/en-x-optum/s/N8SMRs4pheXlkcV6gkubpsPI/26sxtIx9



OTU OPTUM TECH UNIVERSITY

We value your feedback. Please refer to the NPS scoring breakdown when responding: 0-6 = Detractors, 7-8 = Passives, 9-10 = Promoters.

How likely are you to recommend this Optum Tech University (OTU) learning event to others?

The Optum Tech University (OTU) team would like your feedback regarding your learning experience in this class. Input from participants about their experience helps to continually enhance the value and effectiveness of courses like this.

Thank you!

Your Use Cases – Ideas clinic

Google Dataset Search

toolbox.google.com/datasetsearch

Kaggle

kaggle.com/datasets

ProPublica Data Store

propublica.org/datastore

World Bank Open Data

data.worldbank.org

Next steps

- Make a custom plan for yourself to continue this journey
- Improve some of your skills (Stats, Python, ML, Domain, etc.)
- Take some additional courses
- Try a Kaggle.com competition
- Work on a personal project to improve understanding

Blog and Glossary

Blog Example

- http://www.neural.cz/dataset-exploration-boston-house-pricing.html

Machine Learning Glossary

- https://developers.google.com/machine-learning/glossary

Useful Data Science and Machine Learning books for beginners to intermediate:

Book	Author
Spark: The Definitive Guide: Big Data Processing Made Simple	Bill Chambers and Matei Zaharia
Advanced Analytics with Spark: Patterns for Learning from Data at Scale	Sandy Ryza
An Introduction to Statistical Learning	Robert Tibshirani and Trevor Hastie
Python Machine Learning	Sebastian Raschka and Vahid Mirjalili
The Hundred-Page Machine Learning Book	Andriy Burkov

Recommendations Systems

Recommendation System

- Recommendation system consists of a set of web applications that involve predicting user responses to options
- Helps user find content that is to their liking
- Examples
  - Offering customers things that they might like to buy based on their past purchases
  - Recommend online new articles based on reader's interests

Types of Recommendation Systems

Collaborative Filtering

Recommends items based on interests of a group of users

Two types – User based and Item based

Content-based Filtering

Recommends items of similar content

Hybrid Content-based Collaborative Filtering

It is combination of the above two approaches to overcome the disadvantages of each approach

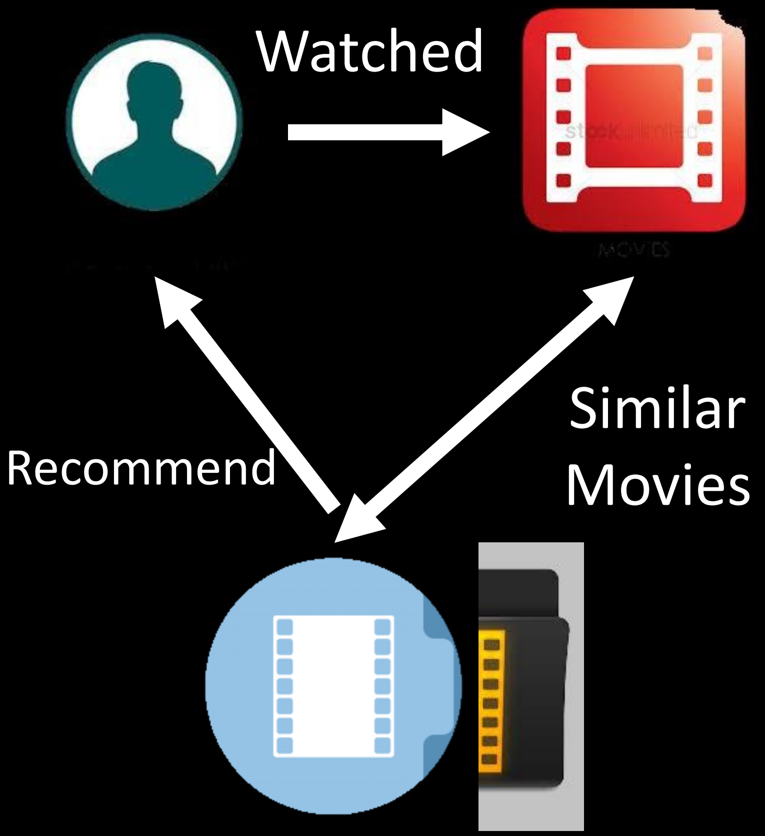
Collaborative Filtering – User Based (Pearson Correlation)

- Create user to item rating matrix
- Also make user-to-user correlations (find highly correlated users)
- Recommend items preferred to those users
- Pros:
  - No item knowledge is required
- Cons:
  - Cannot always rely on user preferences – we humans change
  - Sparsity Problem – if there are a lot of items to be recommended then the user/rating matrix will be spare – we might not have rating for each item
  - Popularity bias – might tend to recommend most popular items

Collaborative Filtering – Item Based

- Create user-item rating matrix
- Also make item to item correlations (find highly correlated items)
- Recommend items with highest correlations
- Pros
  - No knowledge of item features is required
  - Scalability improves since items with correlations are maintained (unless Amazon – thousands of items)
- Cons
  - New user cold start problem
  - New item cold start problem

Content Based Filtering



Watched

Similar Movies

Recommend

Profile Vector (X) – User ratings – movies liked and dis-liked

Item Vector (Y) – Information related to the movies e.g. genre, cast, length, etc.

Cosine Similarity – measure of similarity between two non-zero vectors. Values are between -1 and 1. Based on these cosine values movies are sorted in descending order

Algorithm recommends items that are of the same type, cannot recommend movie which was never rated

$$\cos(\theta) = \frac{X \, . Y}{||X|| \, ||Y||}$$

Thank You

Reference slides

Algorithms Summary

Regression Algorithms - Comparison

Regression Model	Pros	Cons
Linear Regression	Works on any size of data set, gives information about feature relevance	Assume Linear relationship between features
Polynomial Regression	Works on any size of data set, works well with non-linear relations	Need to select right polynomial degree for good results
Super Vector Regressor (SVR)	Works well on non-linear data set, not biased towards outliers	Must apply feature scaling, difficult to understand
Decision Tree Regressor	No need to feature scale, works with both linear and non-linear data sets	Poor results on very small data set
Random Forest Regressor	Powerful and accurate, good performance on many problems including non-linear	Need to choose right number of decision trees

Classification Algorithms - Comparison

Classification Model	Pros	Cons
Logistic Regression	Probabilistic approach, gives information about statistical significance of features	
K-Nearest Neighbor (KNN)	Simple to understand, fast and efficient	Need to choose right number of neighbors
Super Vector Machine (SVM)	Good performance, not biased towards outliers	Not good for non-linear problems, not efficient for large number of features
Decision Tree Classification	No need to feature scale, works with both linear and non-linear data sets	Poor results on very small data set
Random Forest Classification	Powerful and accurate, good performance on many problems including non-linear	Need to choose right number of trees