TMA Innovation Center

Data Science Level 1

-- Session 3--

Basic Machine learning

Hung Nguyen

TIC Data Team Lead

Data Science Process: Recap

Business Understanding	Data Preparation	Data Munging	Model Training	Model Evaluation	Model Deployment	Model Tracking
Determine	Identify	Impute	Train	Evaluate	Deploy	Monitor
Understand	Collect	Transform	Assess	Peer Review	Document	Maintain
Map	Assess	Reduce	Select	Present		Test
	Vectorize					

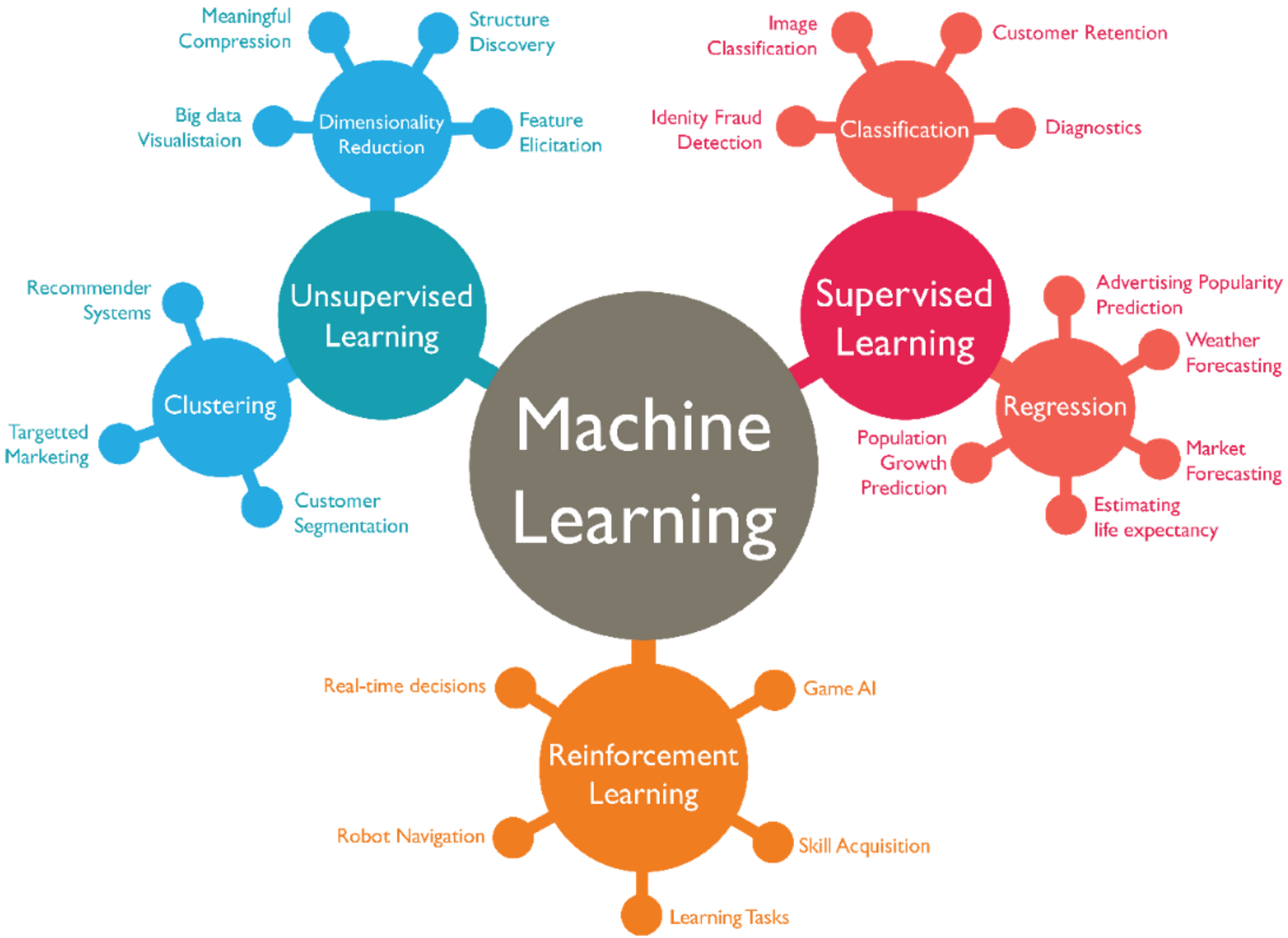
DISCUSS COLLATE WRANGLE PERFORM COMMUNICATE EXECUTE TRACK

What is machine learning?

Machine learning = **learn** from examples and **experience**, **without** being explicitly **programmed**

Why Machine Learning?

- Some tasks cannot be defined well
- Find hidden relationships/correlations within **large amount** of data
- Human designing cannot work well as desired
- **Reduce** computational **time**


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Market Forecasting
Estimating life expectancy
Population Growth Prediction
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

TMA Innovation Center

Data Science Level 1

-- Session 3--

Basic ML Algorithms

- Linear Regression
- Logistic Regression
- K-Means clustering
- kNN
- Decision Tree
- SVM
- Naive Bayes
- Random Forest
- Dimensionality Reduction Algorithms
- Gradient Boosting algorithms

Linear regression

Find

$$\hat{Y} = W \cdot X + b$$

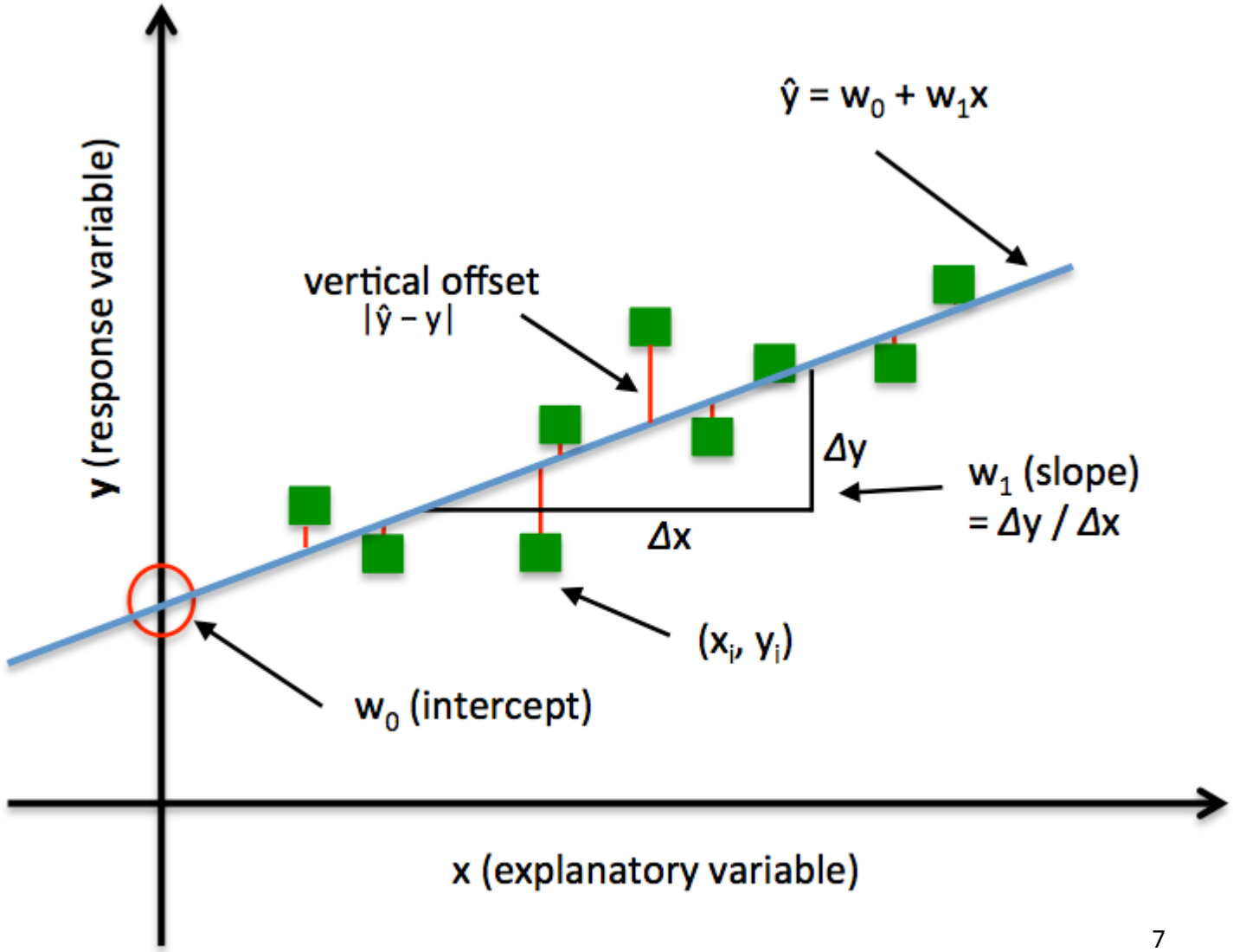
that minimizes

$$\sum (Y - \hat{Y})^2$$

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()

model.fit(X, y)
```



Regression model evaluation

- Root mean squared error (RMSE)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2}$$

- Mean squared error (MSE)

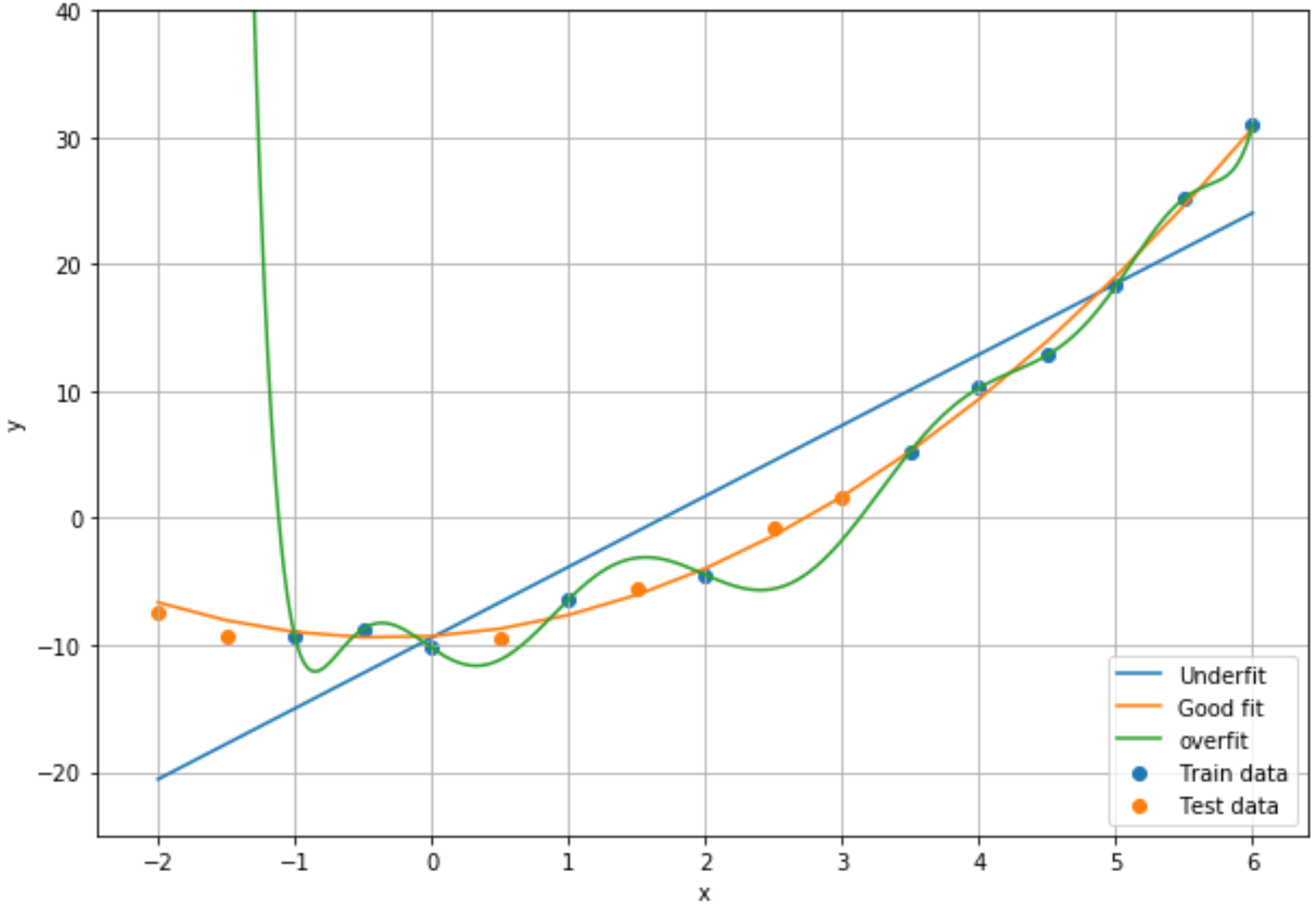
$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2$$

- Mean absolute error (MAE)

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left\|Y_i - \hat{Y}_i\right\|$$

The **smaller error**, the **better model**

Overfit & underfit



		Error on test set	
		small	**large**
Error on train set	**small**	**good fit**	overfit
	large	You're lucky!	underfit

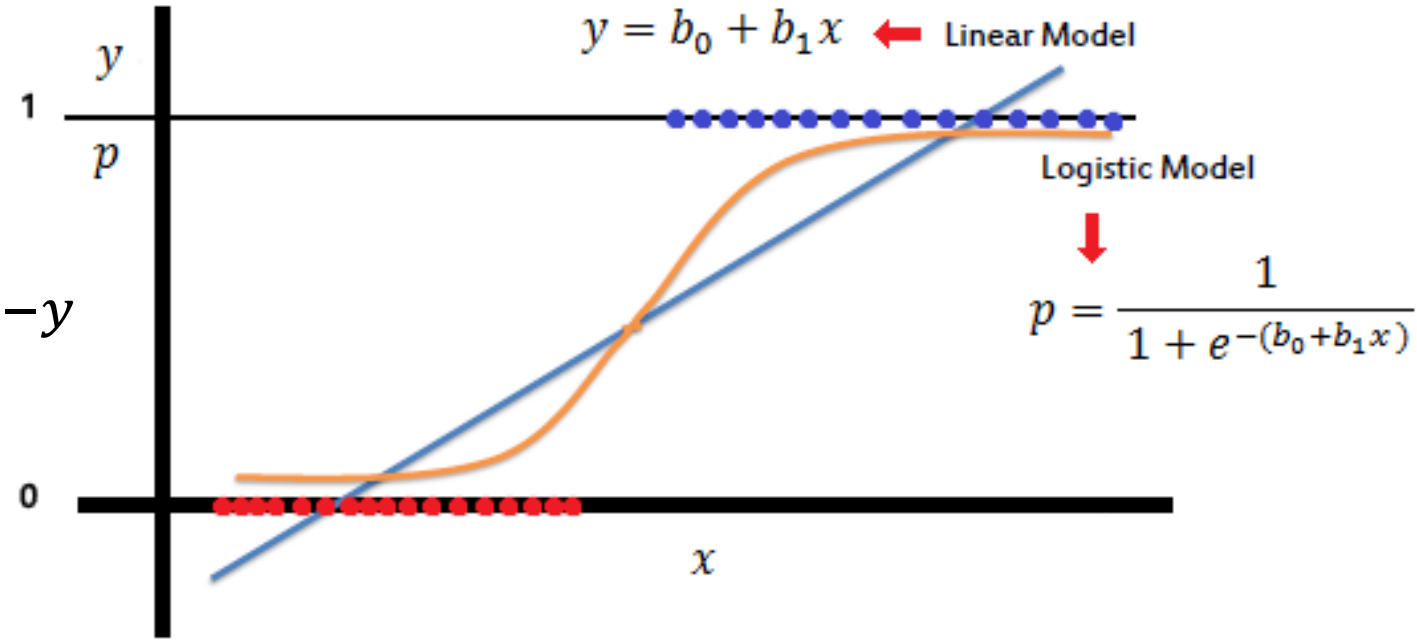
Logistic regression

Find parameters θ

that maximize **likelihood**

$$\Pr(y|X;\theta) = h_\theta(X)^y\left(1 - h_\theta(X)\right)^{1-y}$$

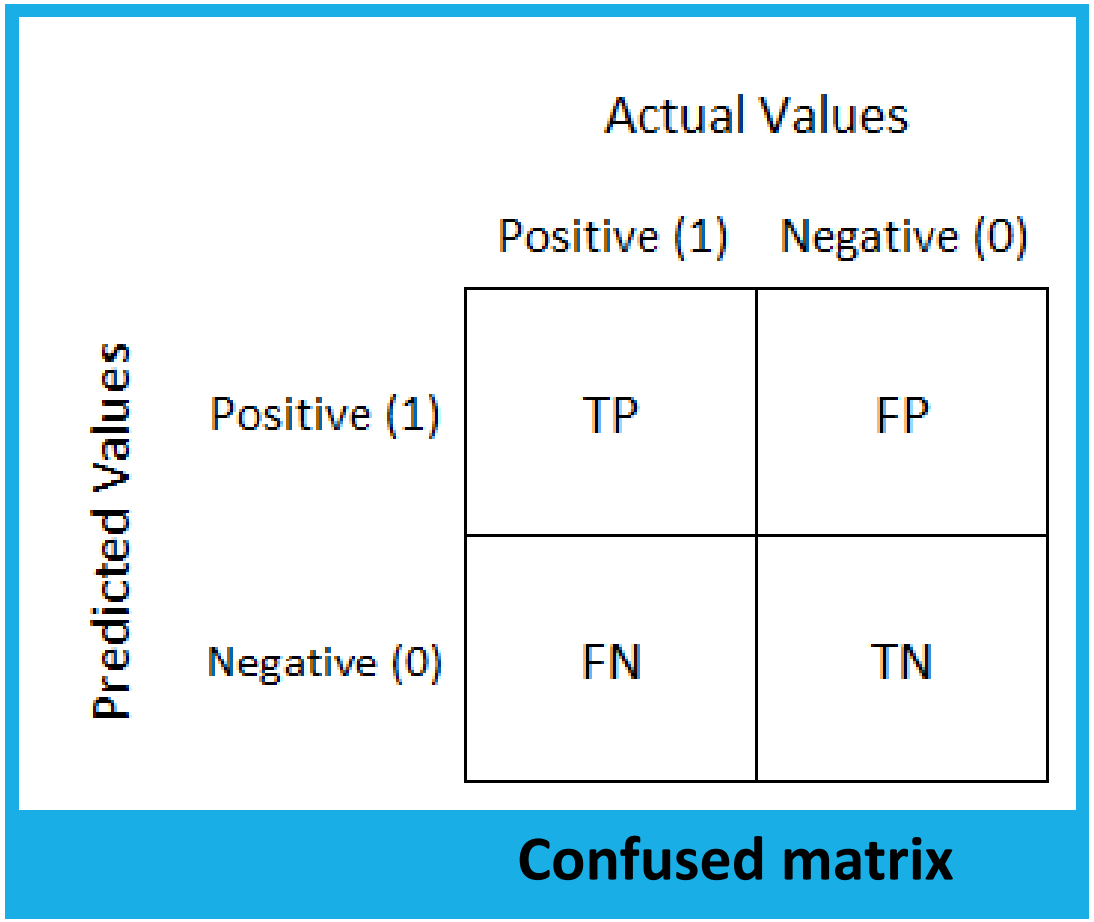
$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}} = Pr(Y = 1|X;\theta)$$




```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X,y)
```

Classification model evaluation

A lot of metrics to consider



Confused matrix

$$\boldsymbol{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

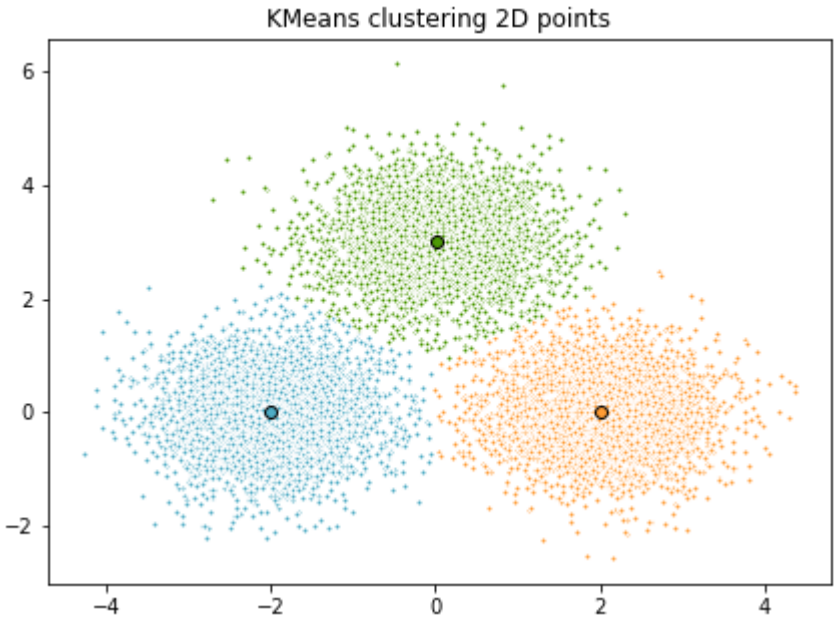
$$\boldsymbol{Precision} = \frac{TP}{TP + FP}$$

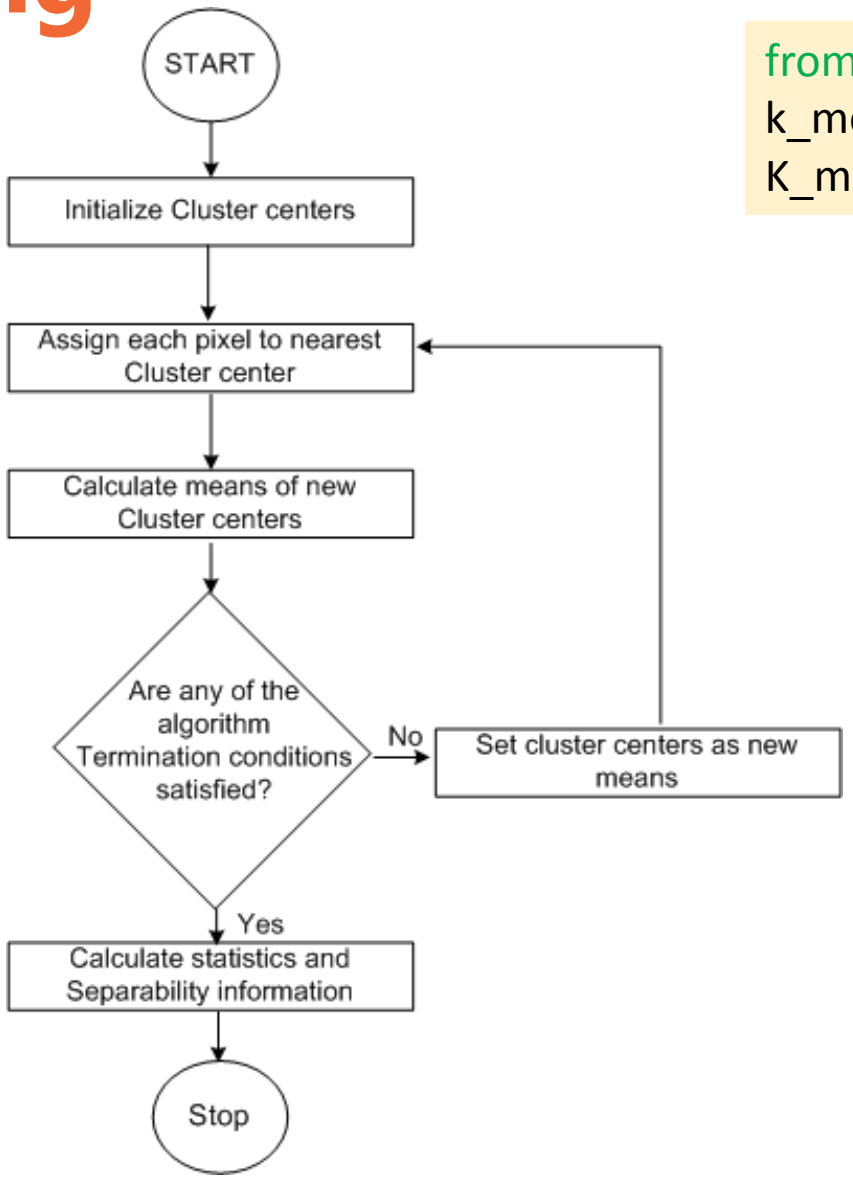
$$\boldsymbol{Recall} = \frac{TP}{TP + FN}$$

$$\boldsymbol{F1} = 2\frac{\boldsymbol{Precision} \times \boldsymbol{Recall}}{\boldsymbol{Precision} + \boldsymbol{Recall}}$$

$$\boldsymbol{ROC}$$

k-Means clustering





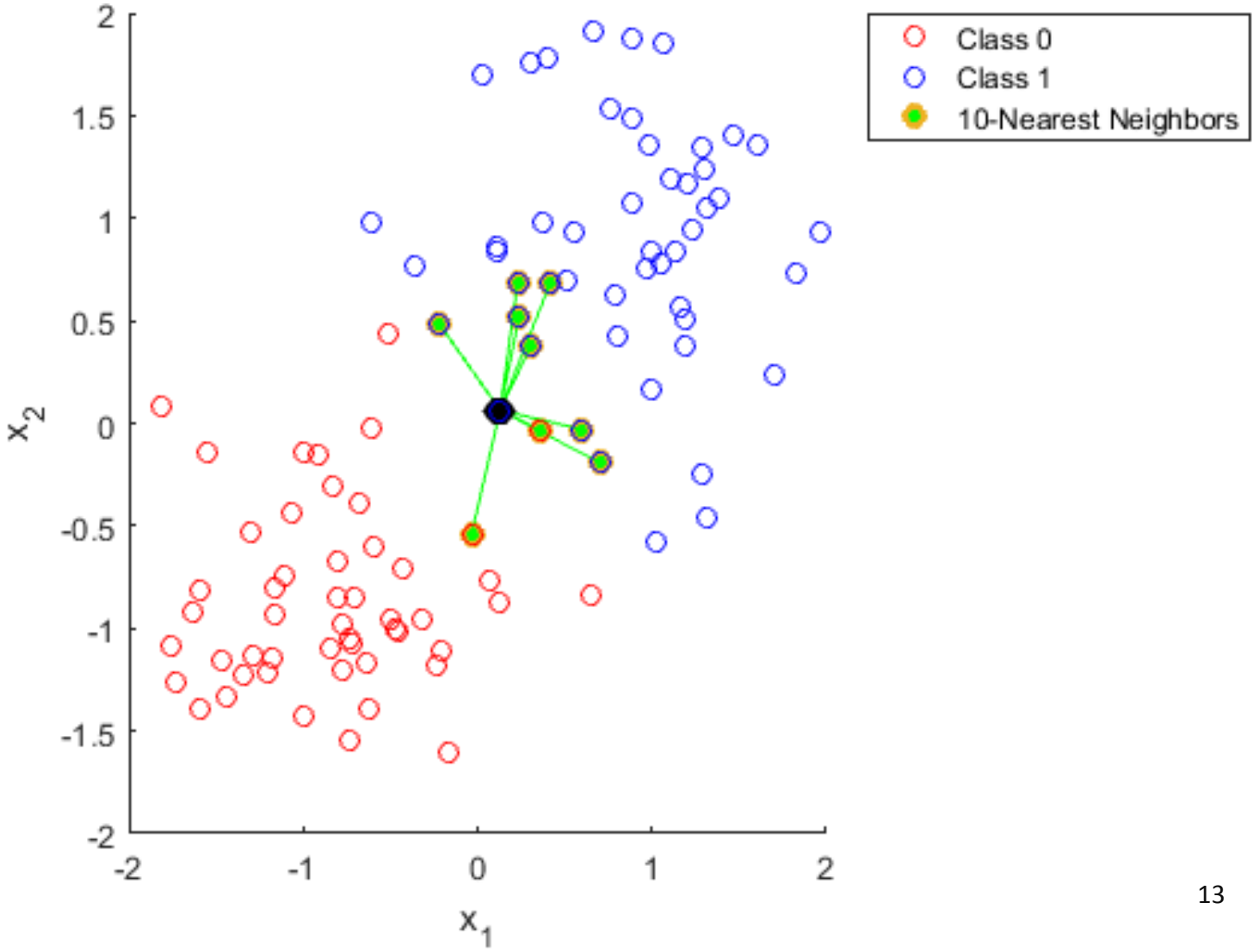

```
from sklearn.cluster import Kmeans
k_means = Kmeans()
K_means.fit(X)
```

K nearest neighbors: regression & classification

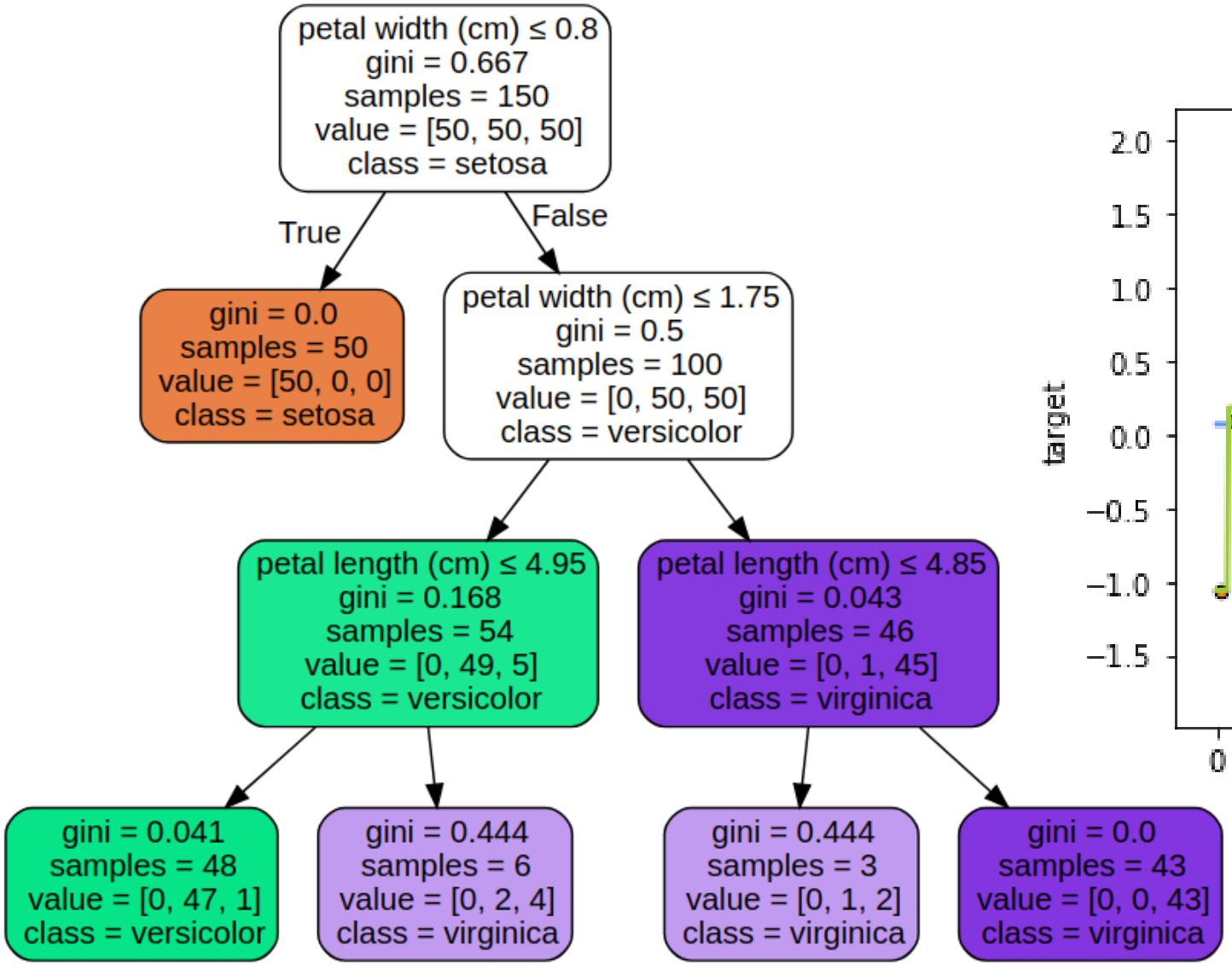
Predicted label of new point

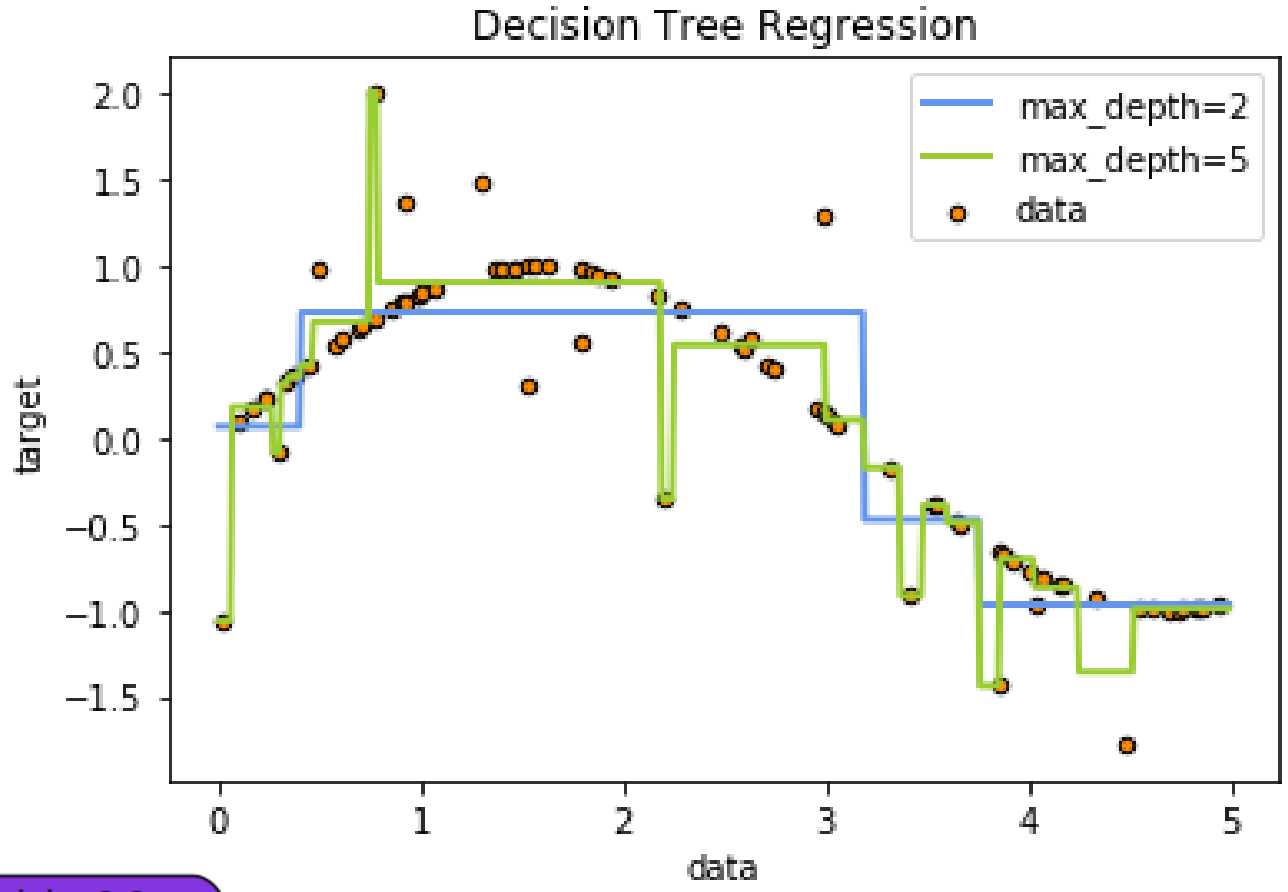
is based on its k nearest points

- If **classification** – **voting**
- If **regression** – taking **averages**



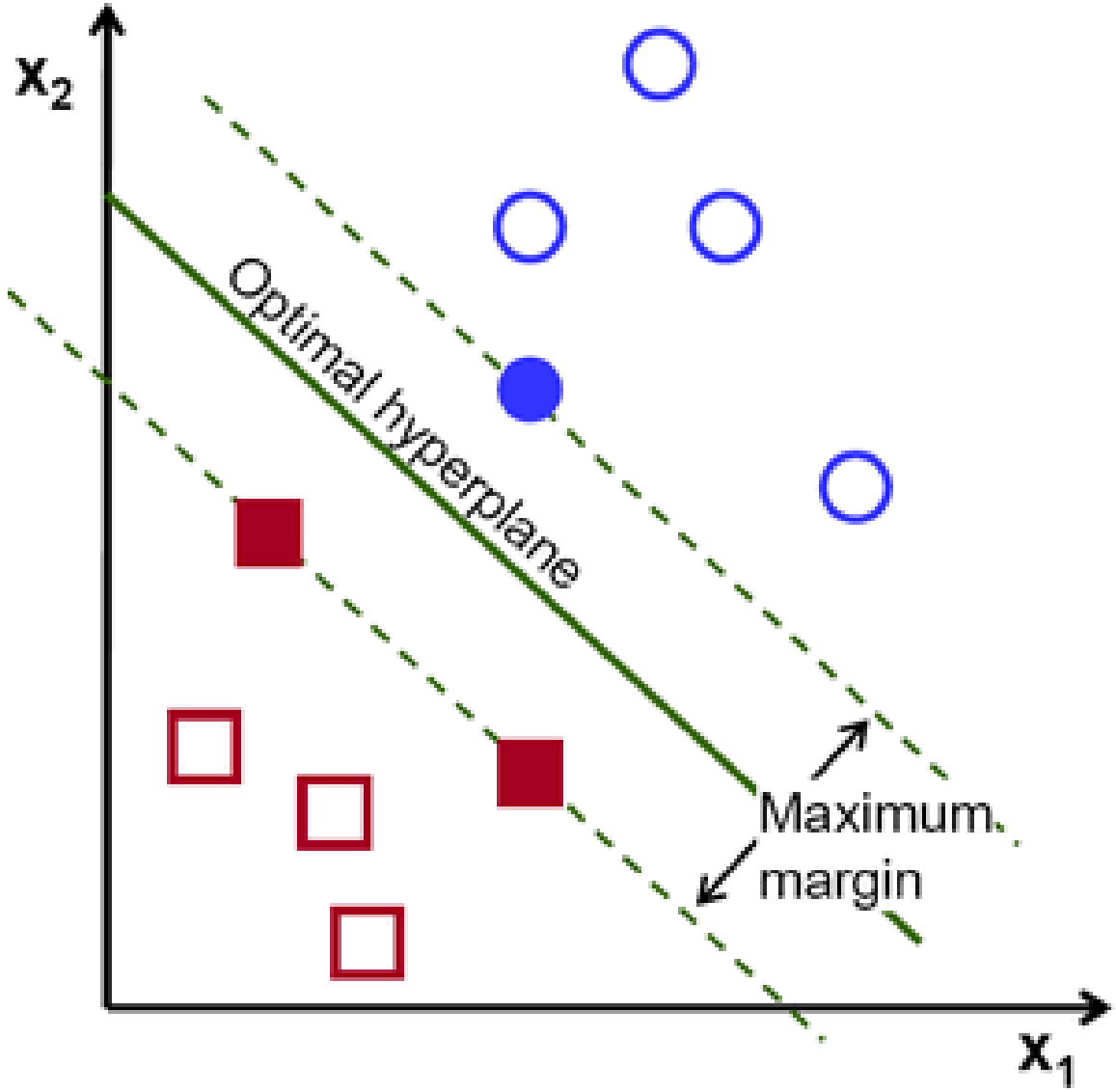
Decision tree: regression & classification





Support vector machine - classification

Find some line (hyperplane) that splits the data between the two differently classified groups of data, such that the **distances from the closest point** in each of the two groups will **be farthest** away.



GAUSSIAN NAIVE BAYES CLASSIFIER

"Gaussian" because this is a normal distribution

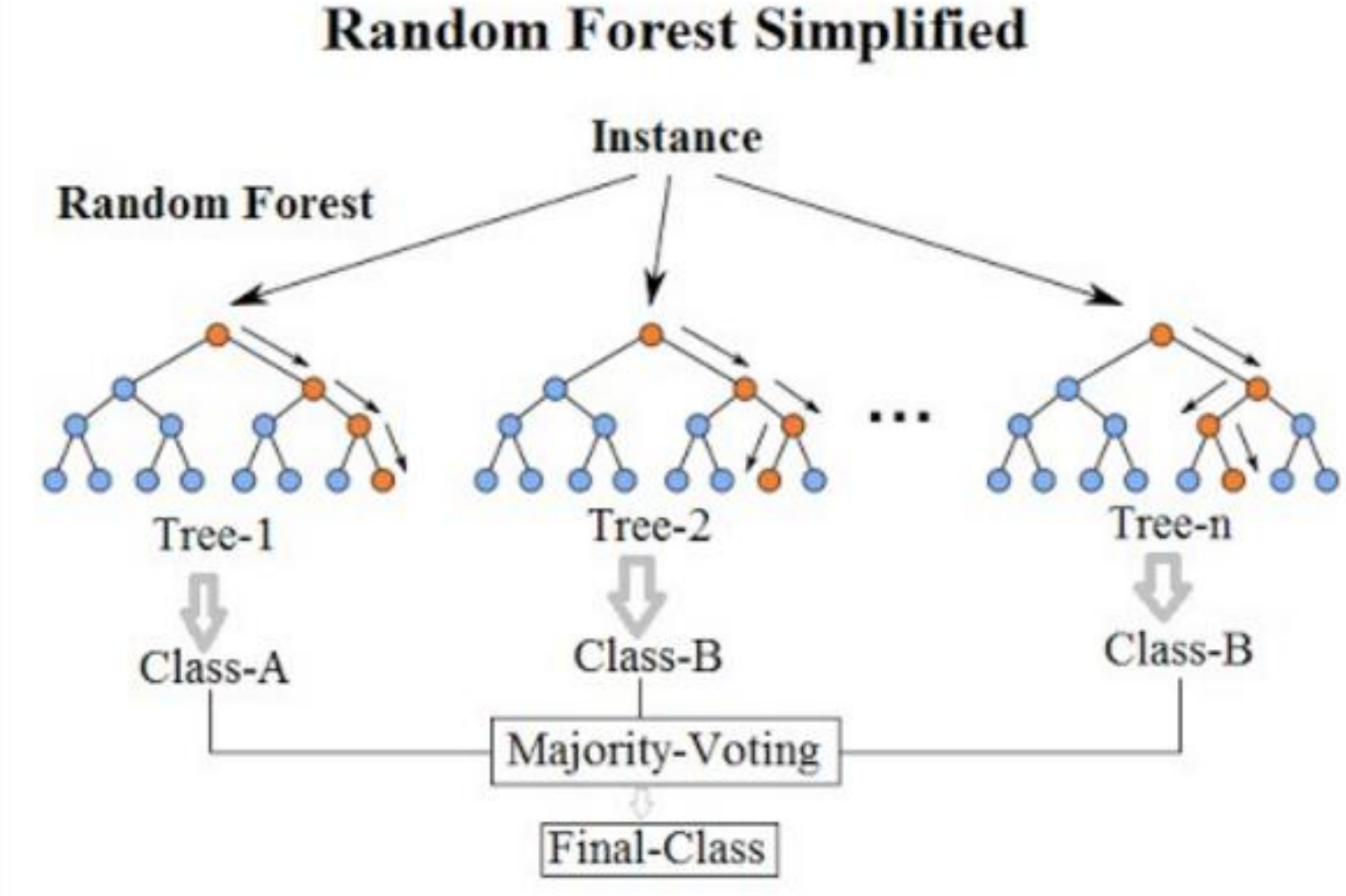
This is our prior belief

$$P(\text{class} \mid \text{data}) = \frac{P(\text{data} \mid \text{class}) \times P(\text{class})}{P(\text{data})}$$

We don't calculate this in naive bayes classifiers

ChrisAlbon

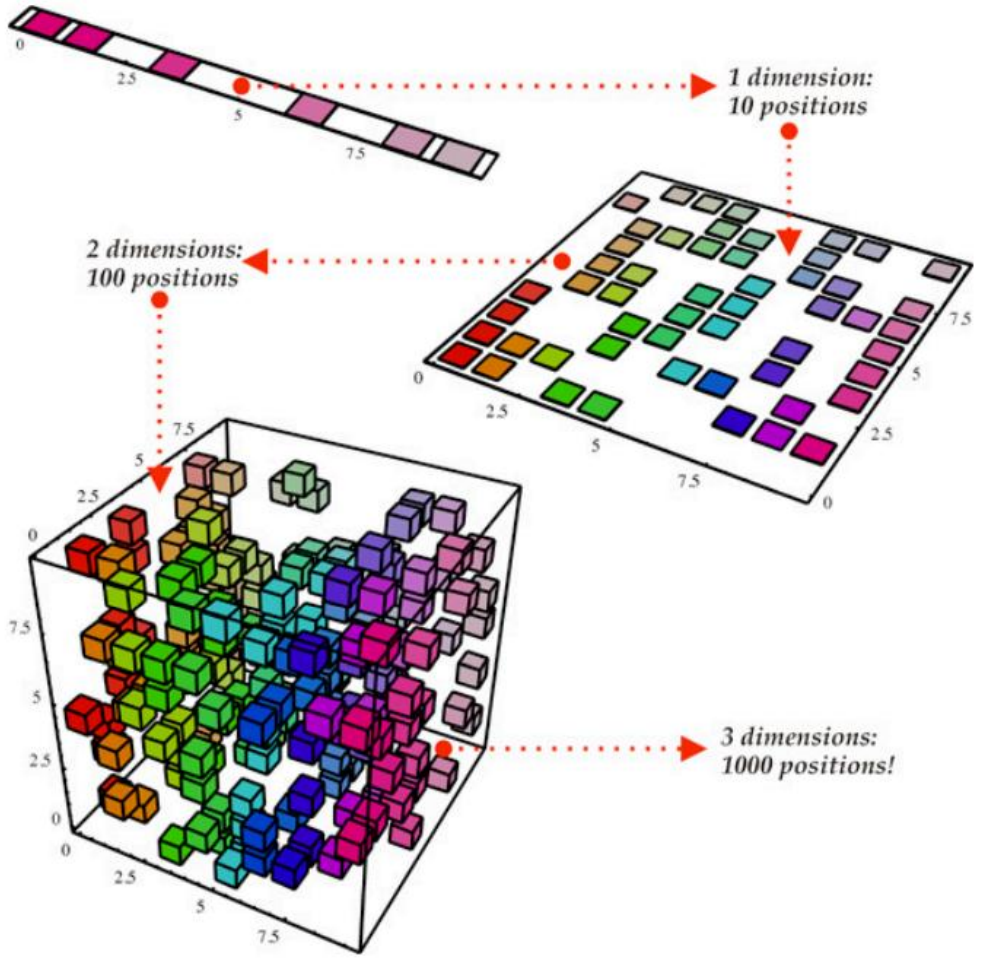
Random forest – an ensemble of decision trees



Dimensionality Reduction Algorithms

Reduce (usually by PCA) dimension

of input data to speed up the model



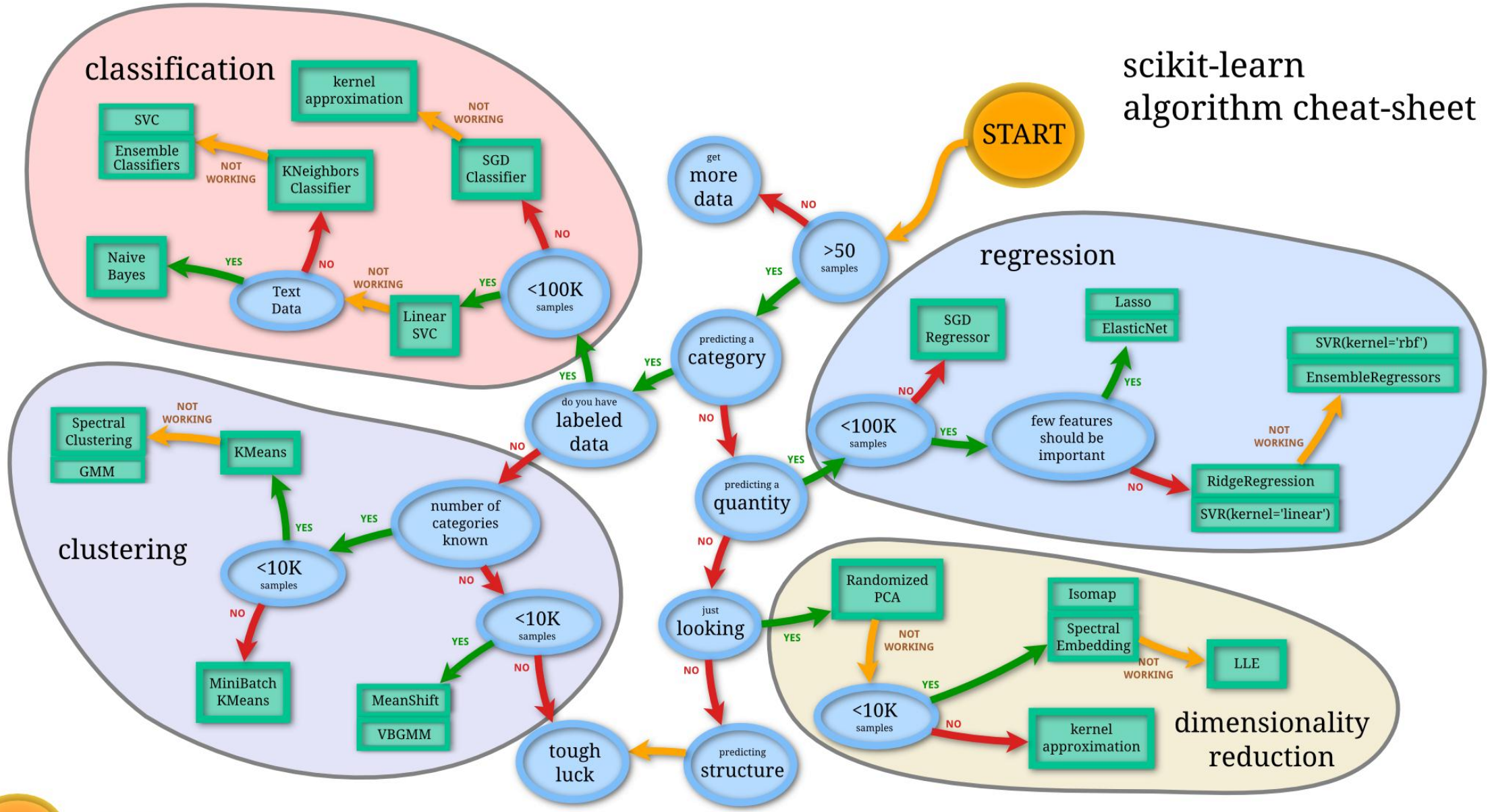
Gradient Boosting

Idea

- make a prediction with **higher prediction power** from **plenty** of data
- an **ensemble** of learning algorithms which **combines** the prediction of several **base estimators**

Framework

- GBM, AdaBoost (sklearn)
- XGBoost
- LightGBM



https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

THANK YOU!

Vietnam:	**84-2839-951-059**
North America:	**+1 844 224 4188**
Australia:	**+61 414 734 277**
Japan:	**+81 364 324 994**

Website: **http://tma-innovation.center**

Email: **innovation@tma.com.vn**