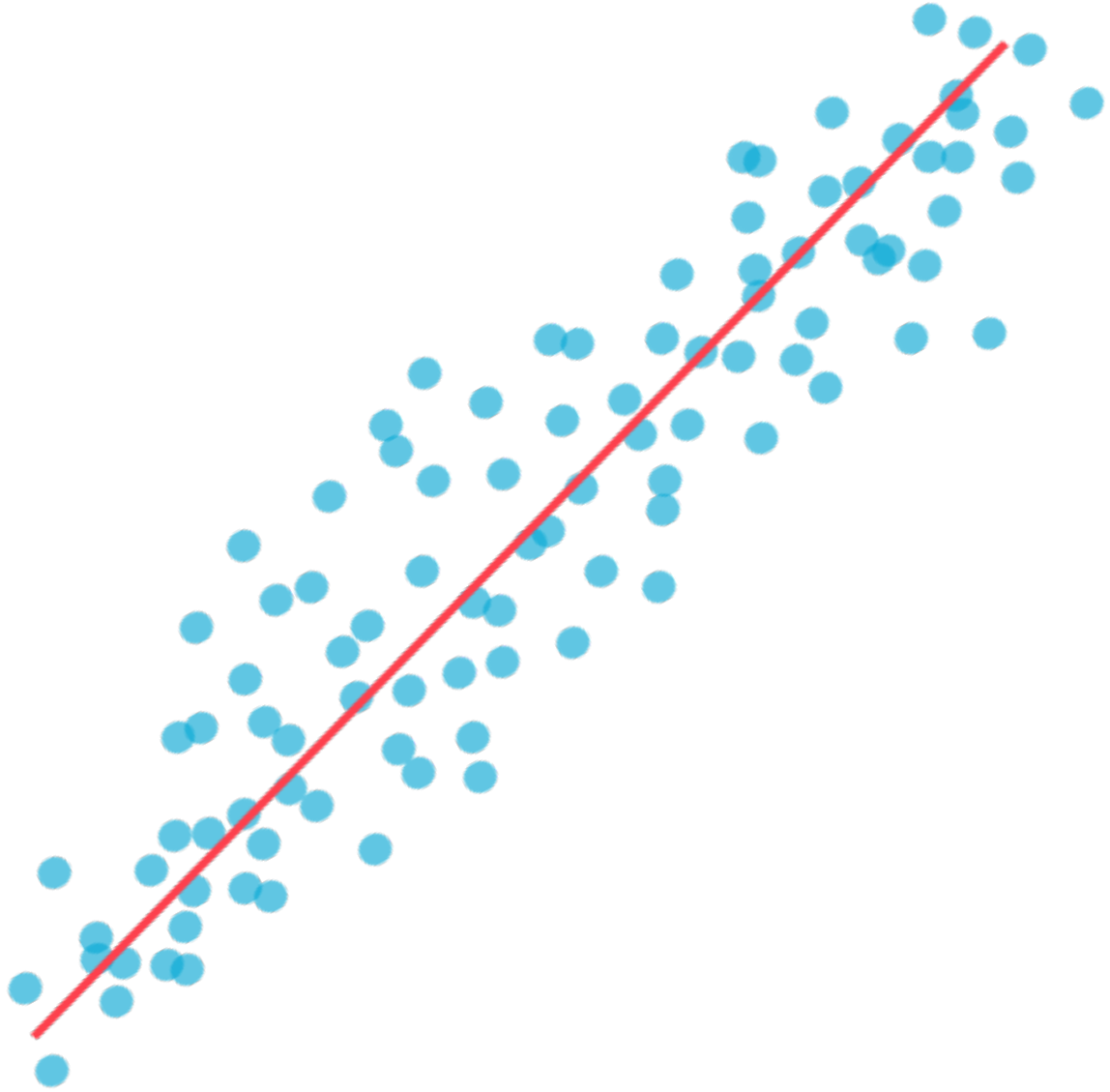


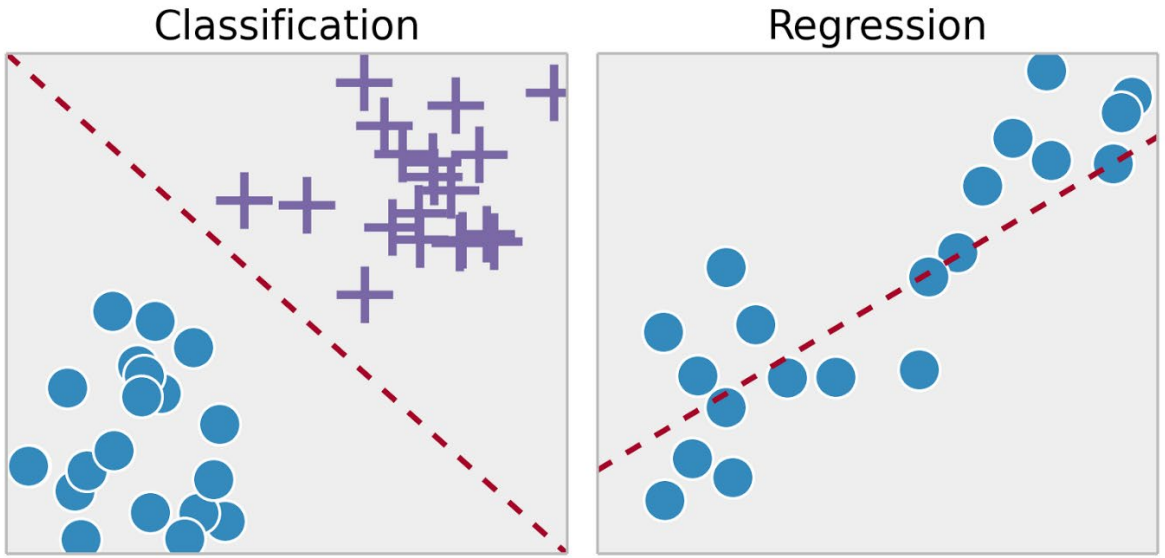
”

Supervised Machine Learning

NOSERYOUNG

Supervised ML

- **Idea**: given a dataset and its corresponding desired output, determine the best algorithm & parameters to predict the output from the data
- **Use cases:**
  - Classification
  - Regression / Prediction
- **Common Methods**:
  - Decision Trees
  - Support Vector Machines
  - **Neural Networks**



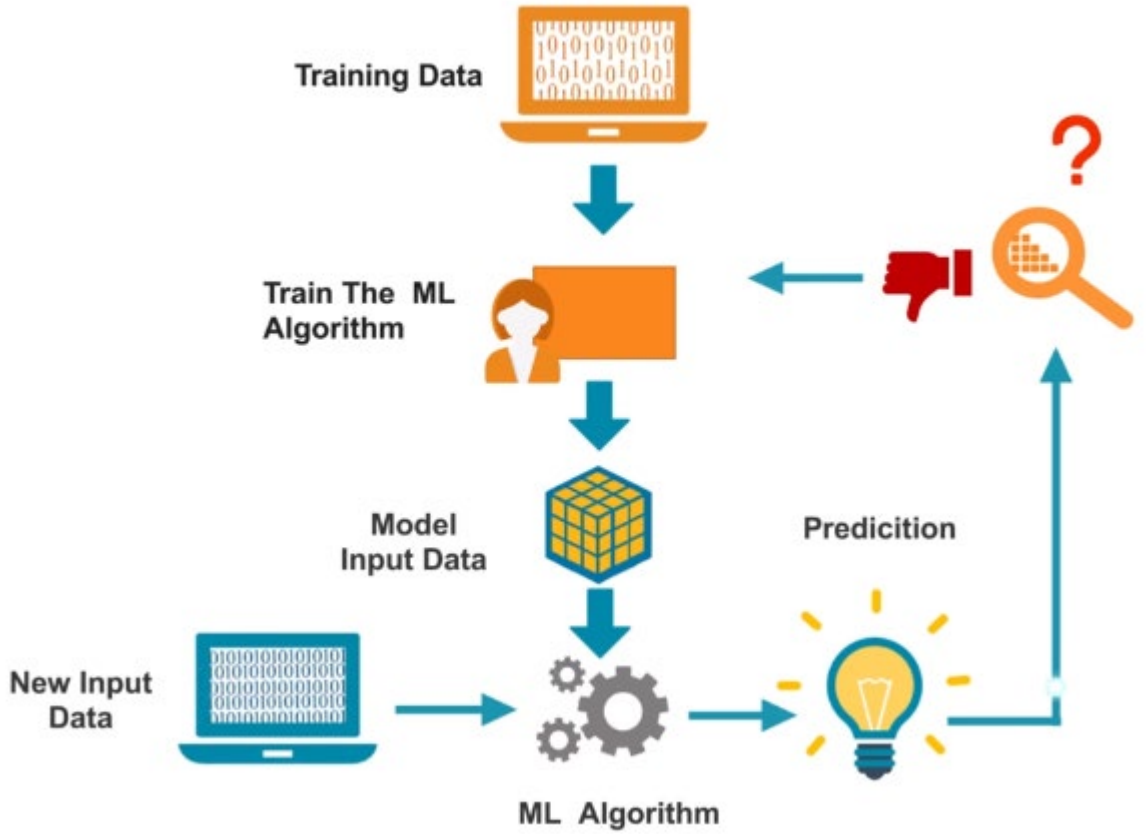
NOSERYOUNG

How does “supervision” work?

- **Intuition:** 1 Algorithm **builds models**, 1 Algorithm **scores the models** and choses the best.
  - ➔ The builder Algorithm tweaks the best model and repeats
- **Building**: Depends on specific method, generally tweaking of model parameters.
- **Scoring**: Calculate a predefined **cost function / score**
  - E.g. Sum of Squared Errors (SSE)
- How Machnes Learn [CGPGrey]

Typical Supervised ML Workflow

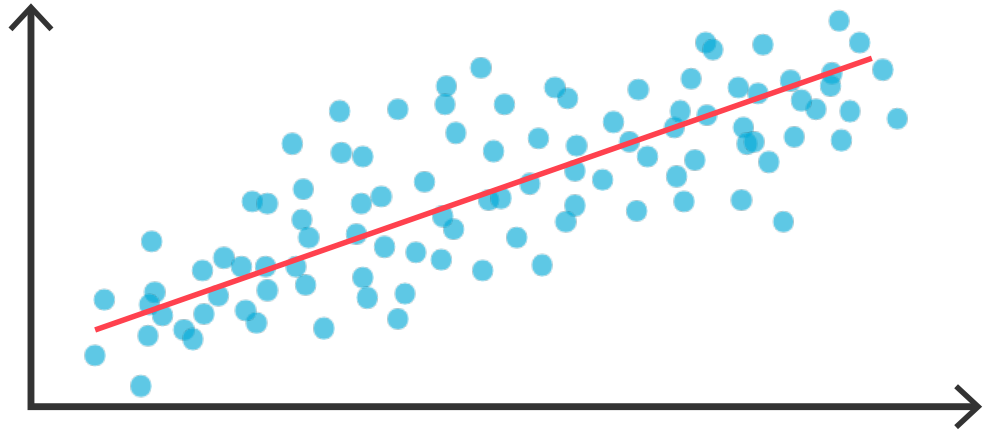
1. Define Goal
2. Get Data
3. Prepare Data
4. Create & Train A Model
5. Evaluate & Improve
6. Make Predictions

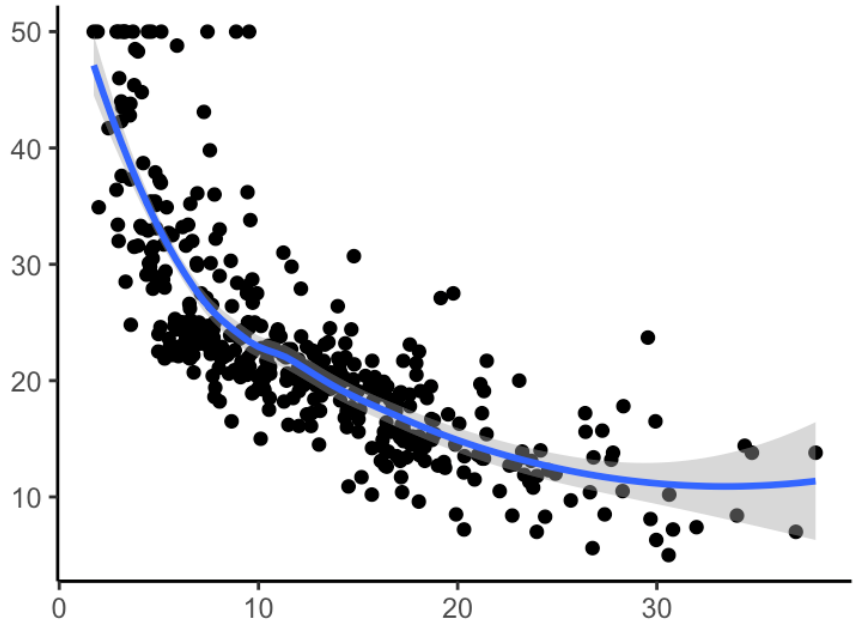


NOSERYOUNG

Regression

- **Idea:** Given a set of input variables, predict a numerical output variable
- **Use cases:**
  - Prediction in Marketing, Medicine, Finance etc.
- **Method:**
  - Find line that minimizes error between prediction and real values



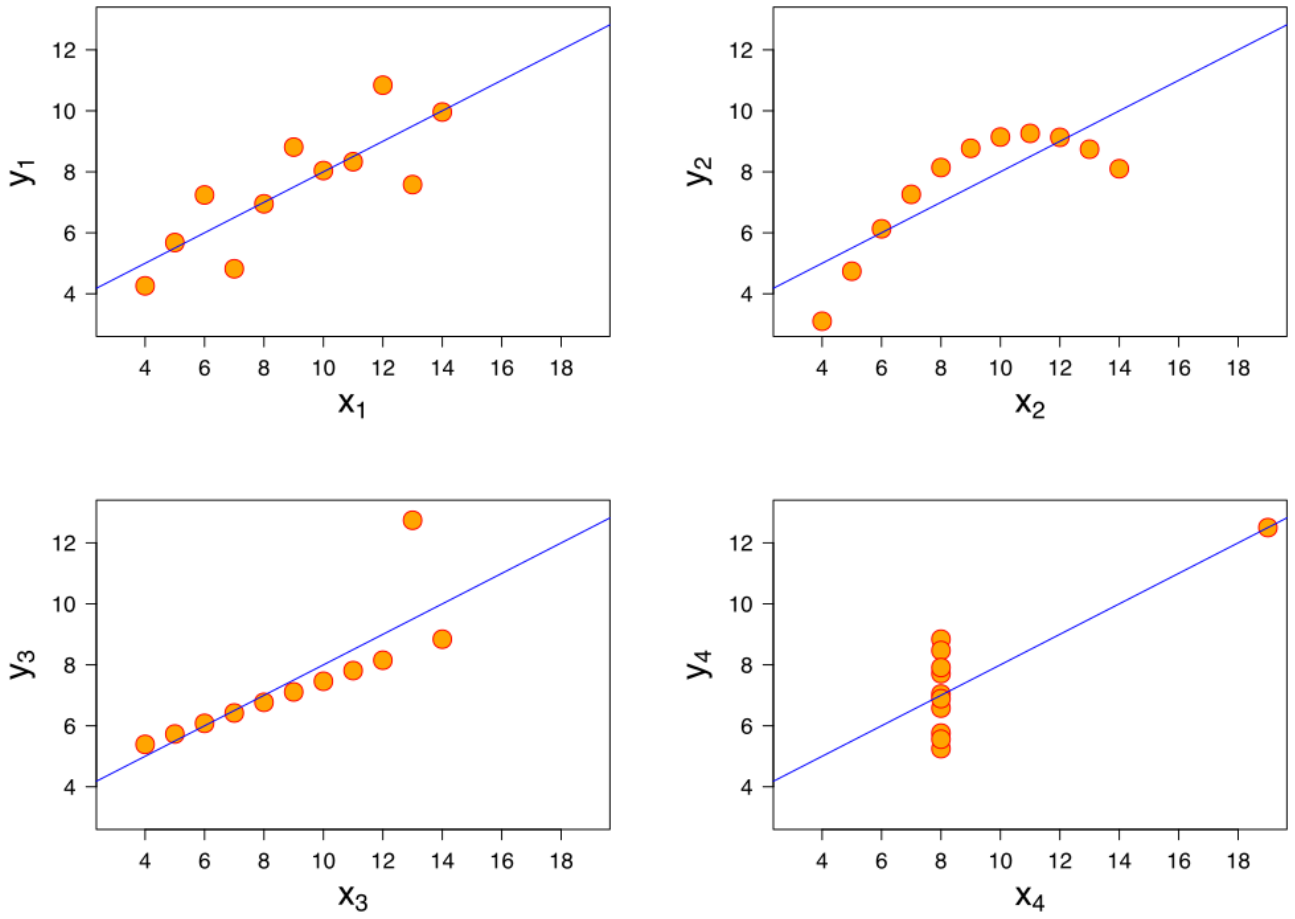


NOSERYOUNG

Regression

Problems

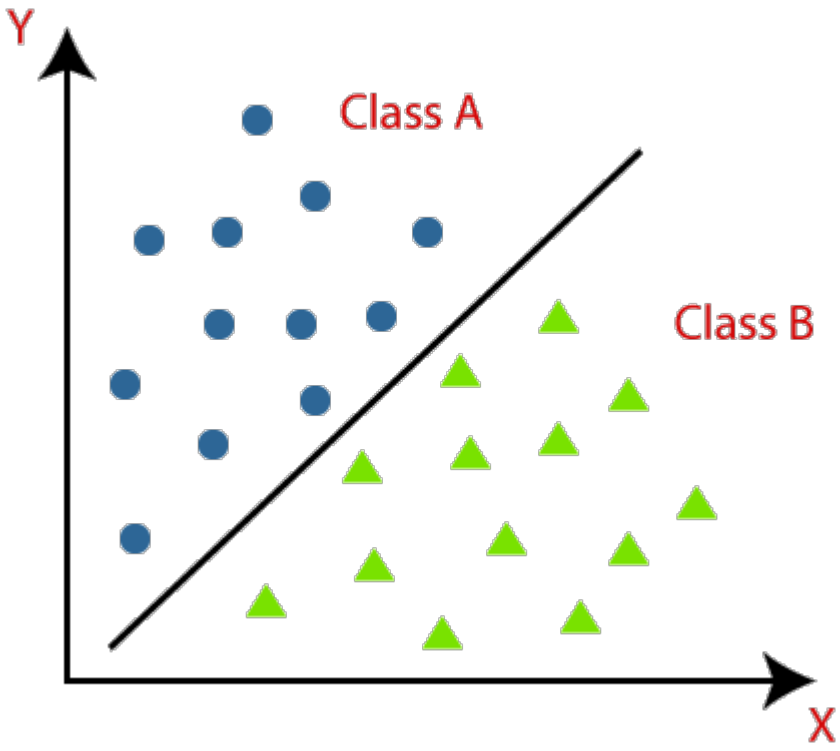
- Can be susceptible to outliers (this can be overcome)
- Can be susceptible to over- / underfitting



NOSERYOUNG

Classification

- **Idea:** given a set of input variables, predict a class for each datapoint
- **Use cases:**
  - Image-/ Speech-recognition
  - Medical prognosis
  - Customer Segmentation
  - Spam detection

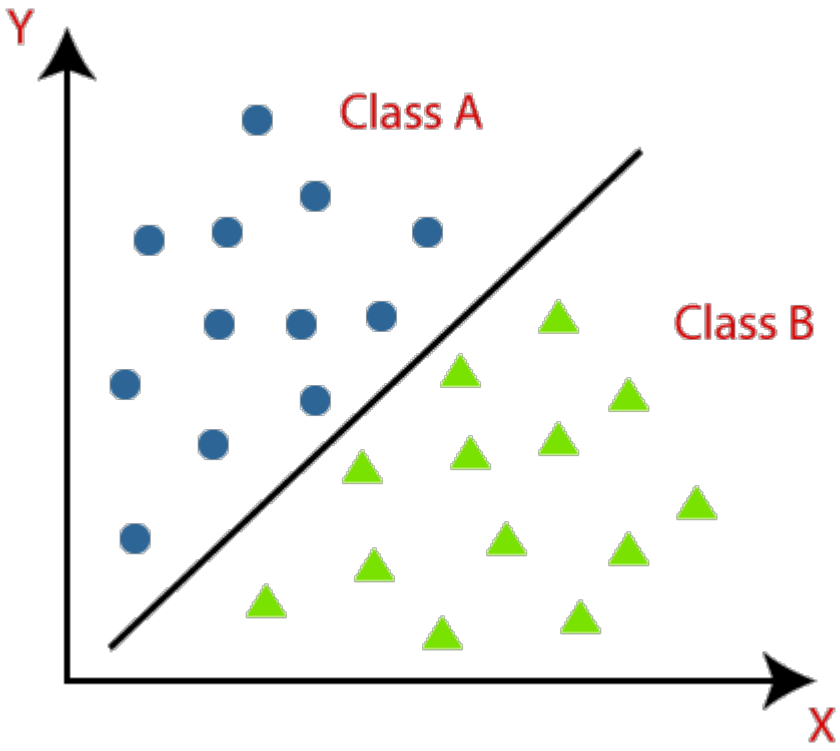


NOSERYOUNG

Classification

Method

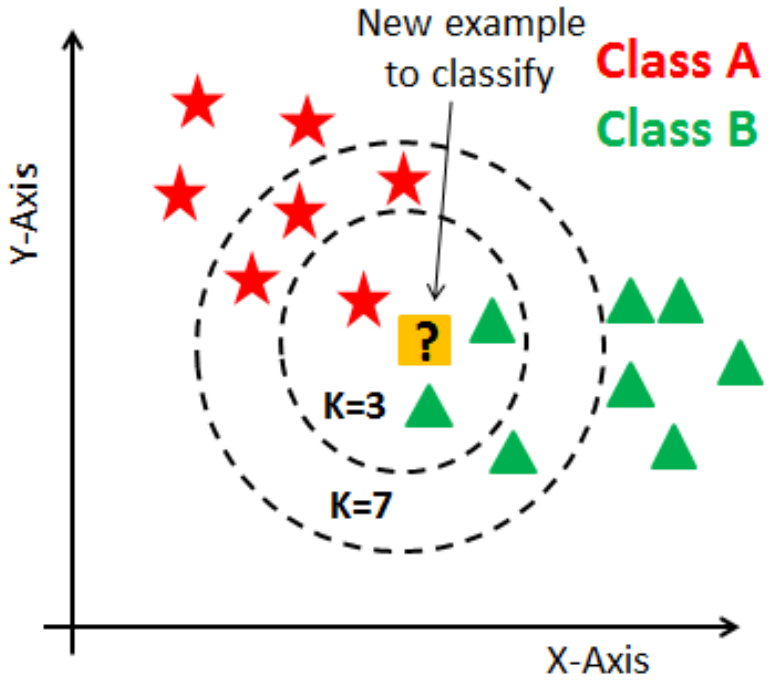
- In general, the aim is to minimize the number and severity of wrong assignments
- **Sleection of classification Algorithms:**
  - K-nearest neighbours
  - Decision Trees
  - Support Vector Machines
  - Neural Networks



NOSERYOUNG

K-nearest-neighbour

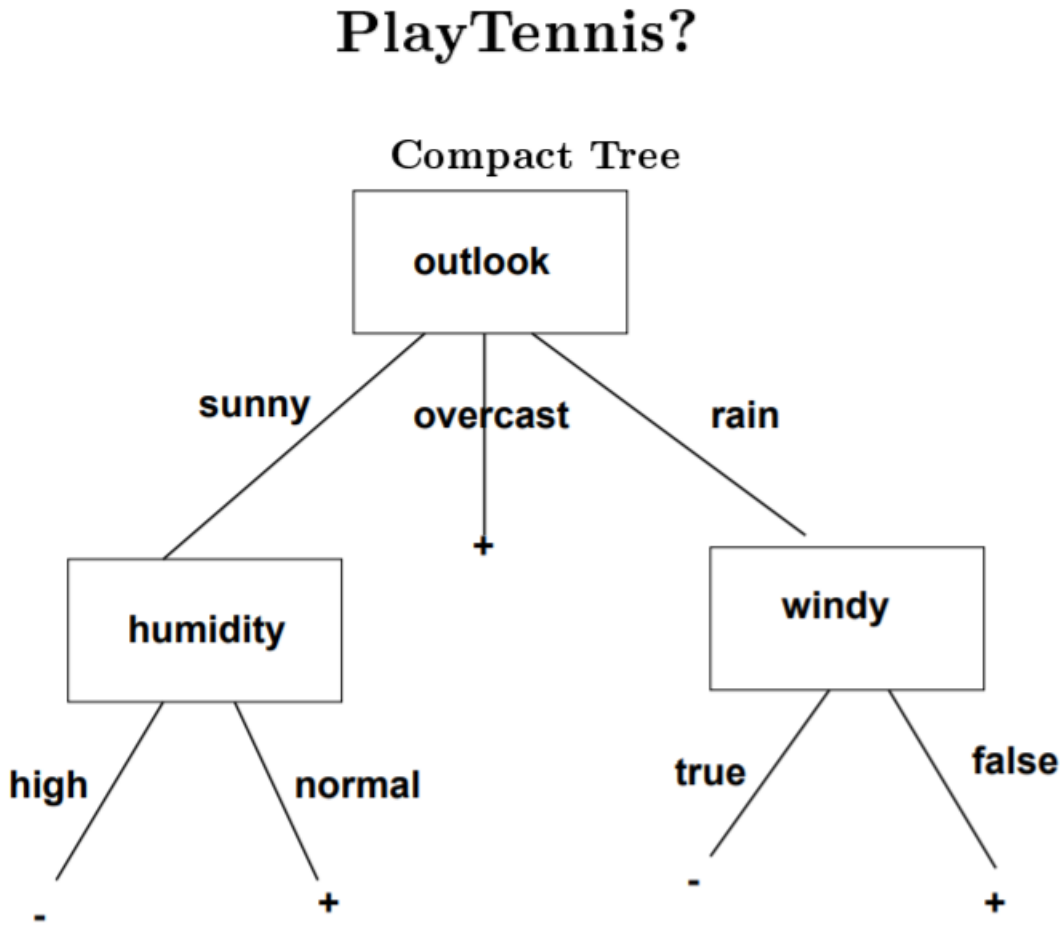
- **Idea**: Classify a point as the majority class of its k nearest neighbours.
- **Pros**: No training, simple, easy to incorporate more data
- **Con**: cannot handle very large, very high dimensional or imbalanced datasets. Sensitive to outliers



NOSERYOUNG

Decision Trees

- **Idea**: find a combination of binary decisions that best classify all datapoints into output classes
- **Pros**: less preporocessing required Easy to interpret and may give additional insights
- **Con**: unstable results ➔ ensamble methods (random forest)

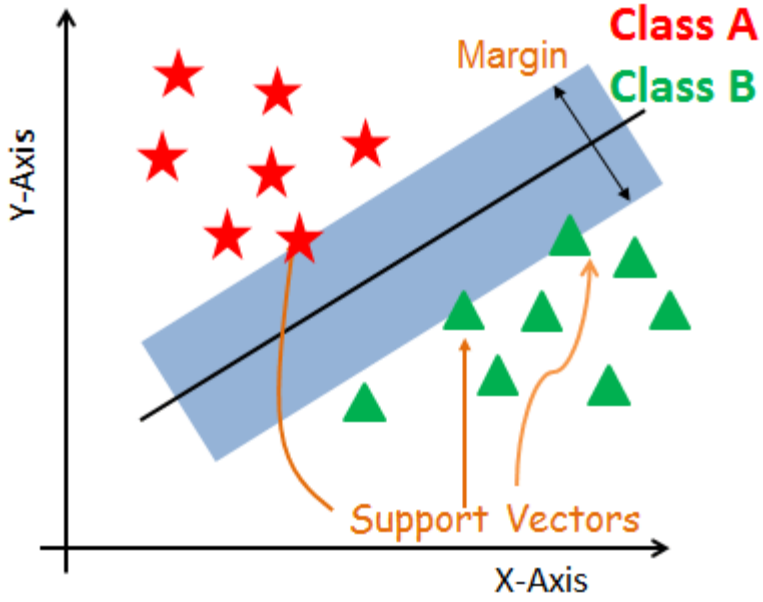


NOSERYOUNG

Support Vector Machines

- **Idea**: find the hyperplane (Line in 2D) that best separates the classes (largest margin)

- Pros: handles highdimensional data well
- Cons: problems with noisy / overlapping data.



NOSERYOUNG

Problems

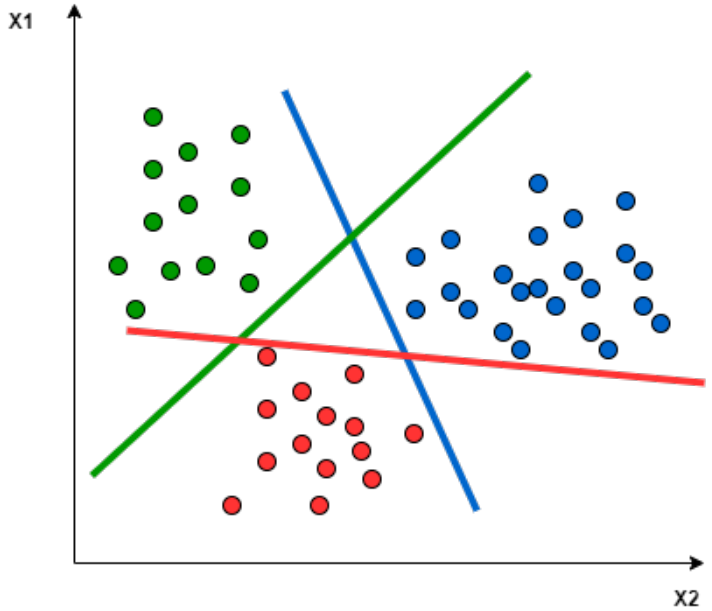
Poor class separation

NOSERYOUNG

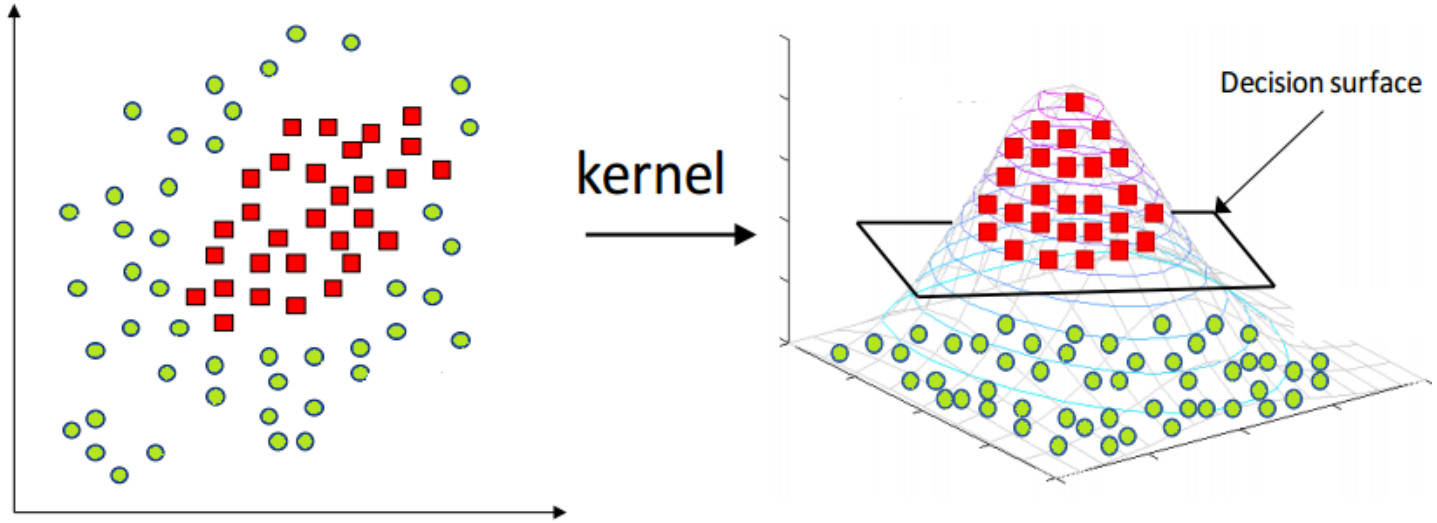
Problems

Advanced

Multi-Class Classification



Feature Transformation



NOSERYOUNG

Hands-On

Part 4

1. Implement the K-Nearest Neighbours Algorithm for a set of random 2D datapoints (use the ‘`sklearn make_blobs`’ function to get a random dataset with underlying clusters
2. Use seaborn.lmplot to perform a linear regression
3. Use sklearn.SVC to **train** a SCV classifier and plot the data with your trained classifier
4. Use your trained classifier to predict the classes of new datapoints.

NOSERYOUNG