# Lots of Data, Little of it Relevant: Imbalanced Datasets

**L. Gjeltema**

$\pi \; day \; - \; 3, twothousandseventeen$

# A motivating example from the industry of industrial machine manufacturing

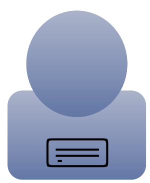
- A company is manufacturing expensive industrial machines for their clients.
- Machines don‘t break very often:
  - ▶ a **lot of sensor data of machines** that are **working**
  - ▶ very **few data points of machines** that are about to **fail**
- They want to know **when** one of these **machines will break in production**.

# A motivating example from the industry of industrial machine manufacturing

Based on past data, can we predict when a machine is going to break in production?

Why does the manufacturer care?

*Questions:*



Service contracts, maintenance intervals, replacement parts,...

? Why would we like to know when a machine may break?

? What other questions could we answer with that data?

# Abstract the business problem to a classification problem with imbalanced data

Machine failure is our target activity we want to predict.

▽▼

We have lots of machine data when machines work well, but only little data about failing machines – our data is imbalanced.

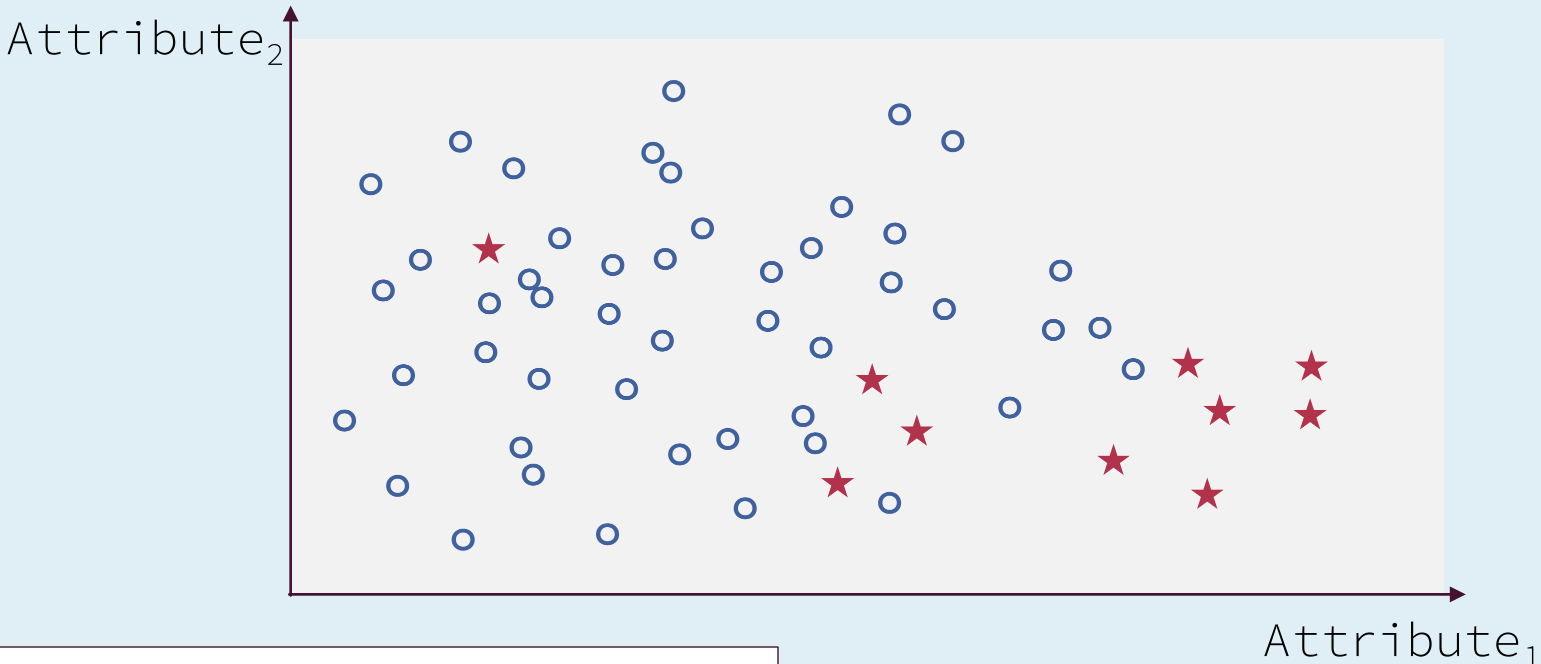
▽▼

Our algorithms won't get enough of these minority 'failure' signals in the data to learn the machine behavior – we have to amplify these signals.

# Approaches to tackle imbalanced data

- **Data-based sampling**
  - Stratified undersampling and oversampling
  - Synthetic generation of new minority data points
  - Tomek links, Cluster-based samples, …
- **Modification of existing algorithms**
  - Cost-sensitive learning
  - 1-class learner
- **Data preparation with ensemble algorithms**
  - Bagging, Boosting, …

# SMOTE:
# Synthetic minority oversampling technique

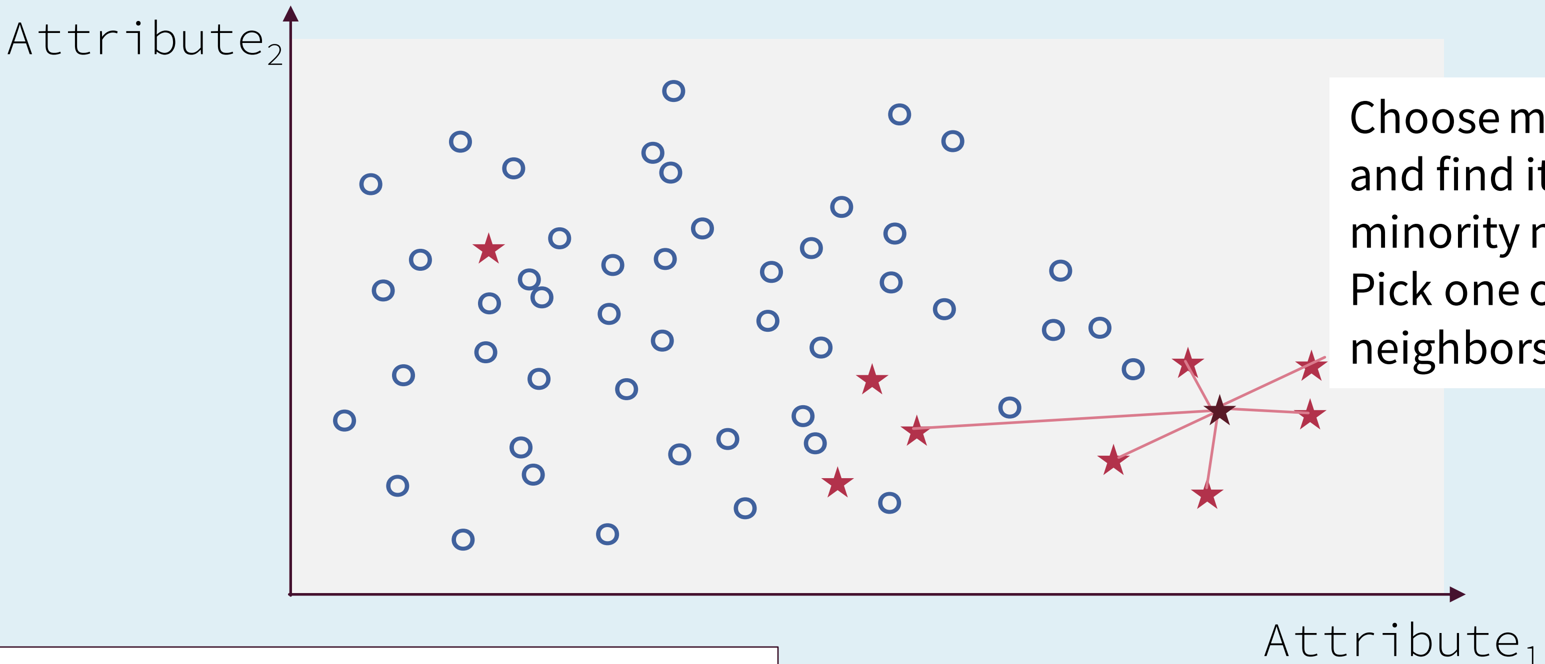



*Legend:*

- Majority (machine working)
- Minority (machine failure)
- Synthetic minority data

Chawla et al: SMOTE: Synthetic Minority Over-Sampling Technique. *J. Artificial Intelligence Research*, 2002.

# SMOTE:
# Synthetic minority oversampling technique

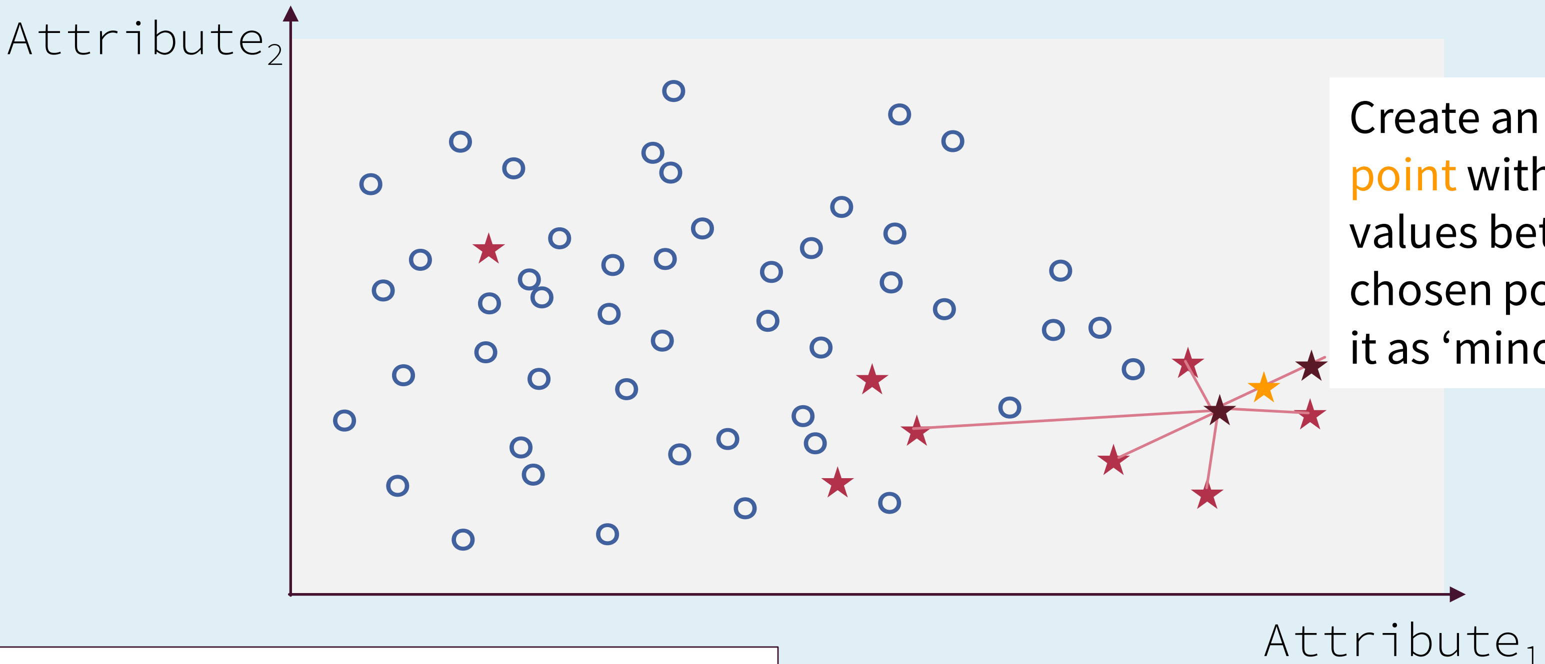



Choose minority point and find its nearest k minority neighbors; Pick one of these neighbors.

*Legend:*

- Majority (machine working)
- Minority (machine failure)
- Synthetic minority data

# SMOTE:
# Synthetic minority oversampling technique

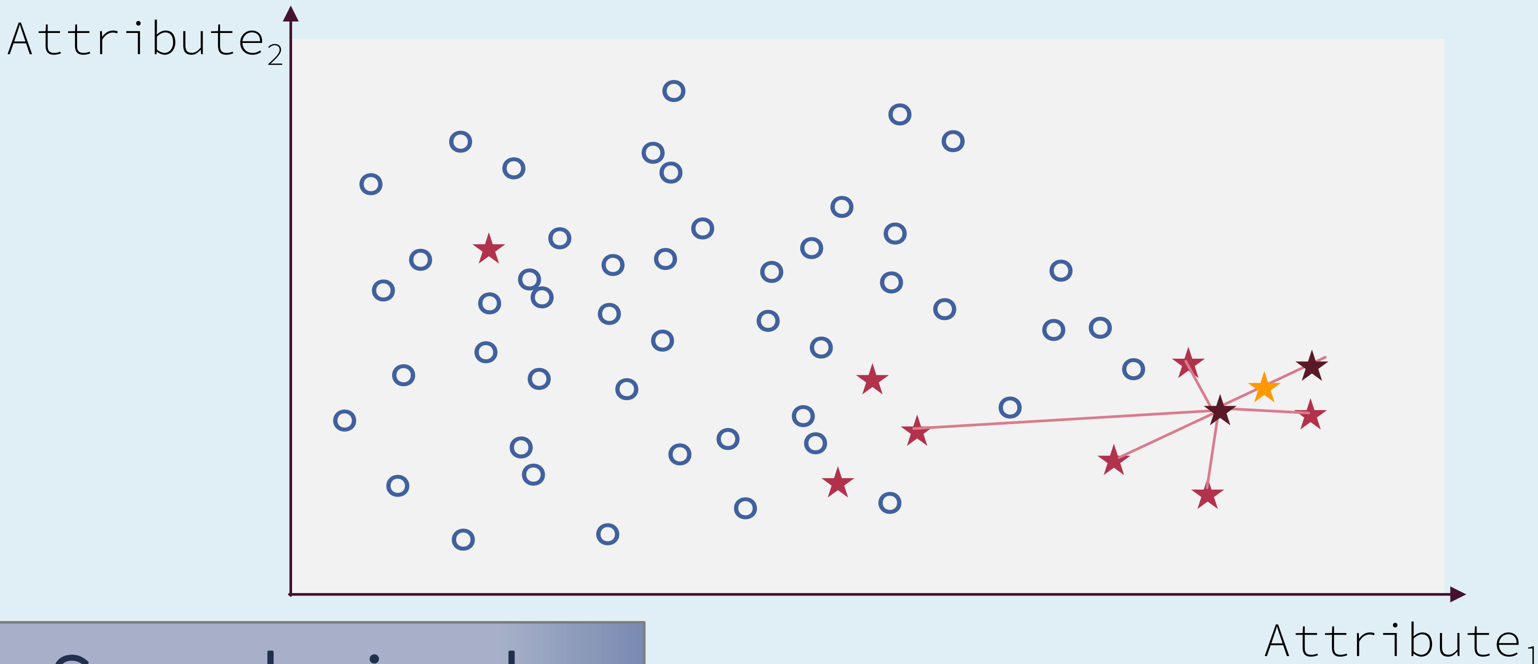



Create an artificial data point with attribute values between the two chosen points and label it as 'minority'.

*Legend:*

- ○ Majority (machine working)
- ★ Minority (machine failure)
- ★ Synthetic minority data

# SMOTE: Synthetic minority oversampling technique




Sounds simple. What‘s the catch?

Overgeneralization; variance; how to deal with categorical data; ...

# SMOTE:
# Expansions and alternatives

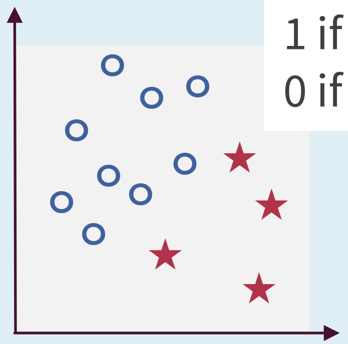
- Modified SMOTE [safe|border|noise]
- SPIDER [local oversampling of minority; filtering out difficult examples from majority]

He, Garcia: Learning from Imbalanced Data. IEEE, 2009

# Cost insensitive learning (not what we want)

**Same misclassification cost for all data points**

Every data point 'costs' us
1 if misclassified,
0 if properly classified



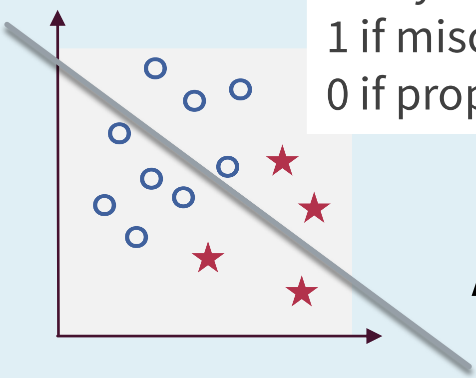
*Legend:*

Majority (machine working)
Minority (machine failure)

# **Cost insensitive learning** (not what we want)

**Same misclassification cost for all data points**

Every data point 'costs' us
1 if misclassified,
0 if properly classified



Above the line? declare machine as failing

Below the line?  declare machine as working

*Legend:*
○ Majority (machine working)
★ Minority (machine failure)

# Cost sensitive learning

~~**Same misclassification cost for all data points**~~

**Different misclassification cost**

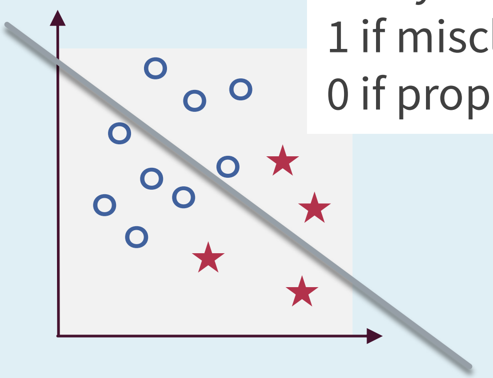
Many algorithms have the same misclassification costs for all data points.

Cost-sensitive learning algorithms assign a higher penalty to misclassifying data from the minority group.

McCarthy et al: Does Cost-Sensitive Learning Beat Sampling for Classifying Rare Classes? *UBDM*, 2005.

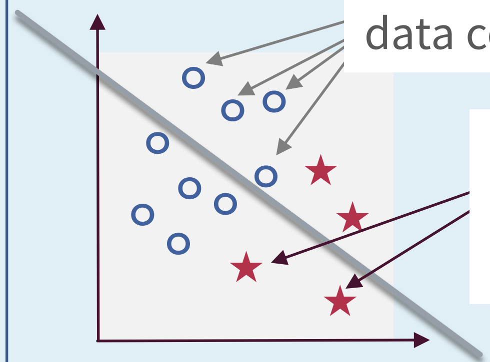
# Cost sensitive learning

## Same misclassification cost for all data points



Every data point ‘costs’ us
1 if misclassified,
0 if properly classified

## Different misclassification cost



Misclassified majority data costs 1 each

Misclassified minority data costs 3 each

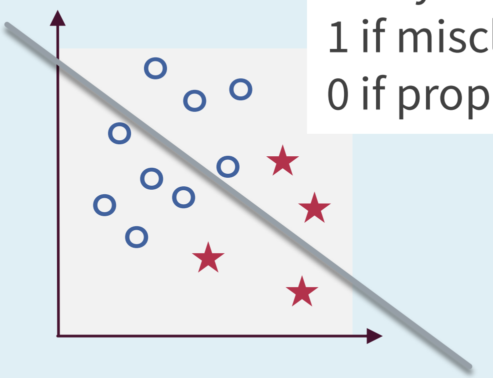
*Legend:*
- ○ Majority (machine working)
- ★ Minority (machine failure)

McCarthy et al: Does Cost-Sensitive Learning Beat Sampling for Classifying Rare Classes? *UBDM*, 2005.

# Cost sensitive learning

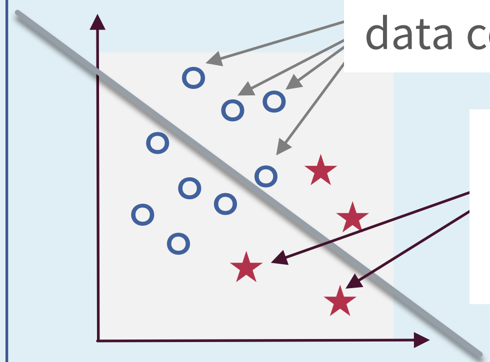
## Same misclassification cost for all data points

Every data point ‘costs’ us
1 if misclassified,
0 if properly classified



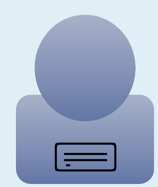
## Different misclassification cost

Misclassified majority data costs 1 each



Misclassified minority data costs 3 each

What is the overall cost of each approach in this example ?



Left side = 6
Right side = 10

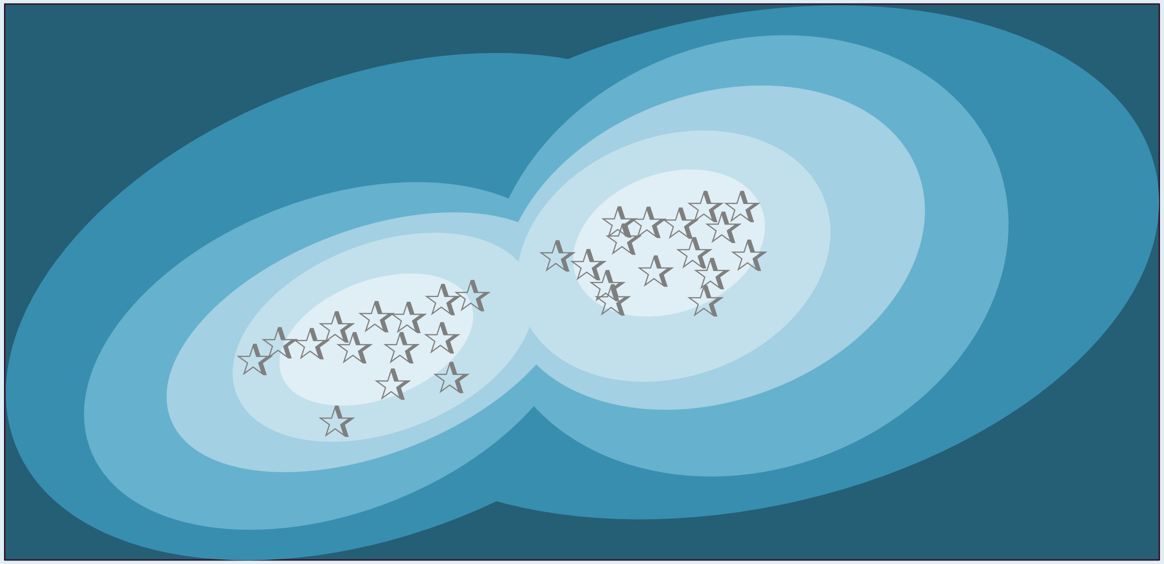
*Legend:*

- o Majority (machine working)
- ★ Minority (machine failure)

McCarthy et al: Does Cost-Sensitive Learning Beat Sampling for Classifying Rare Classes? *UBDM*, 2005.

# 1-class learner

We only feed the minority data into the model for training



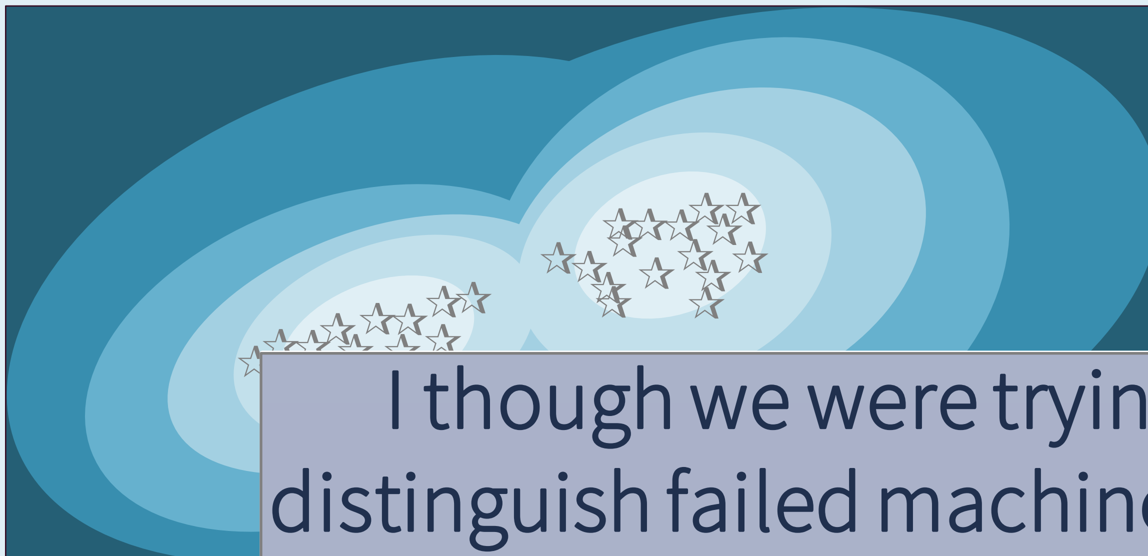
*Legend:*

☆ Training data (only machine failures)

Schölkopf et al: Estimating the support of a high-dimensional distribution. *Neural computation*, 2001.

# 1-class learner

We only feed the minority data into the model for training



I though we were trying to distinguish failed machines from working machines!?

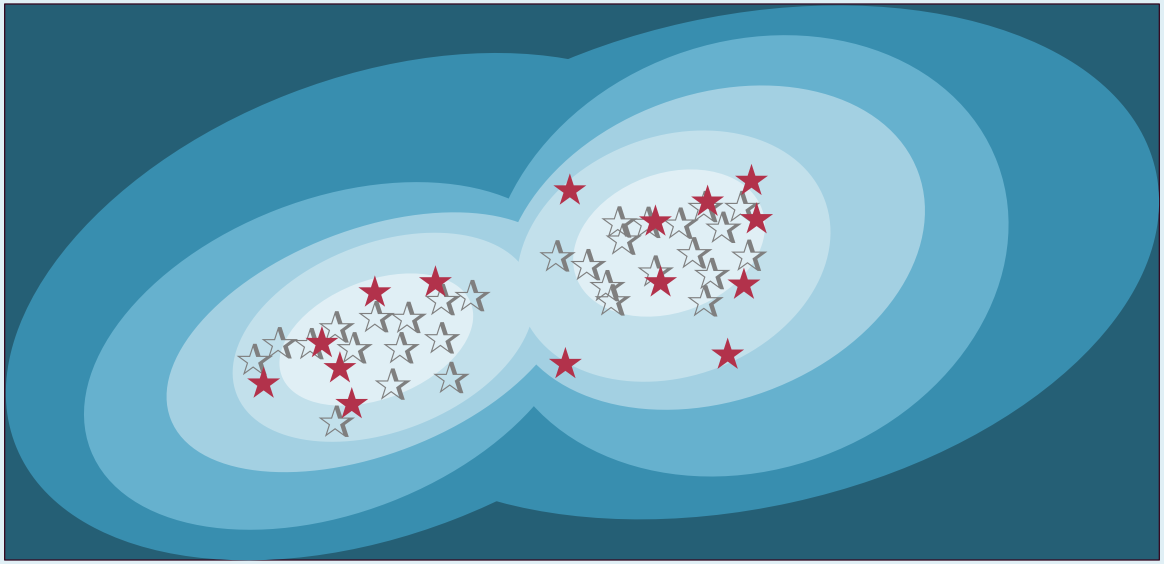
*Legend:*

☆ Training data (only machine failures)

Schölkopf et al: Estimating the support of a high-dimensional distribution. *Neural computation*, 2001.

# 1-class learner

New data points on the inner contours are declared ‘machine failure’



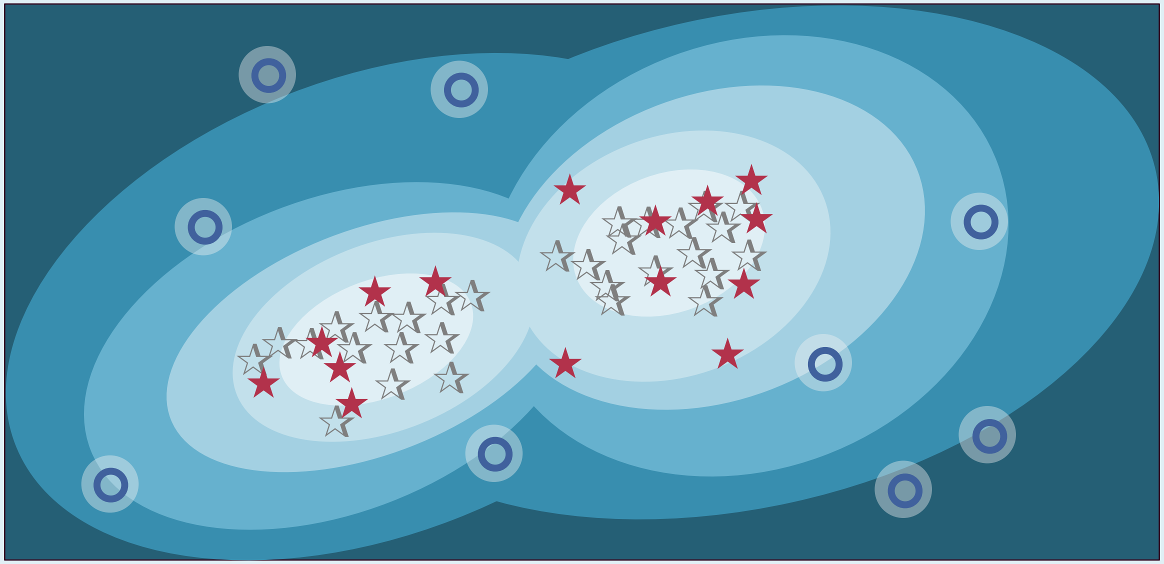
*Legend:*

☆ Training data (only machine failures)

★ New data (we predict: failure)

# 1-class learner

New data points on outer contours are declared 'working'



*Legend:*

☆ Training data (only machine failures)

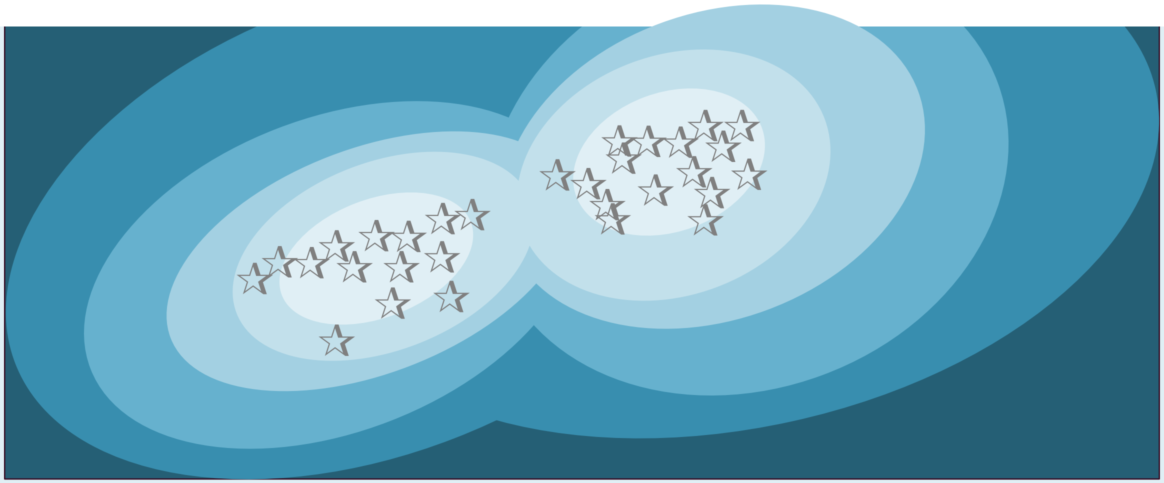
★ New data (we predict: failure)

○ New data (we predict: ok)

# 1-class learner

Another option, used in novelty detection:

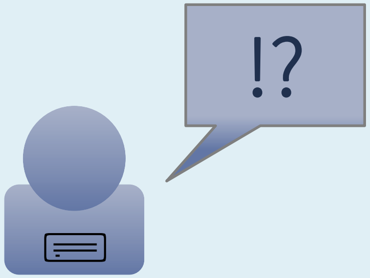
We only feed the majority data into the model for training



*Legend:*

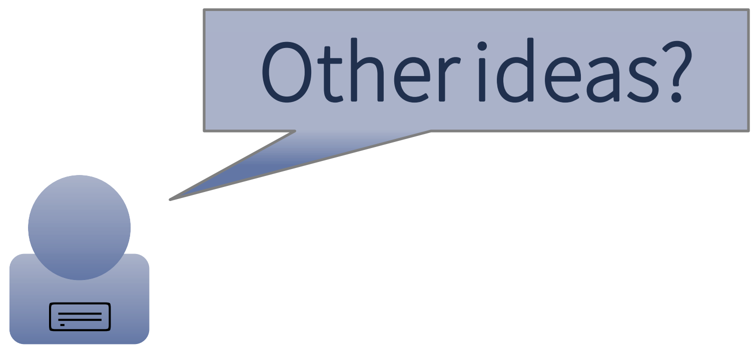
☆ Training data (only working machines)

Narrowly define what "normal" behavior is.
Use that as baseline for novelty detection.



# Other options if all else is lost

- **Data preparation with ensembles (bagging/boosting)**
- **Pool data with your competitor**
- **Be patient**
  - More minority data points may come in as machines break
  - Online learning



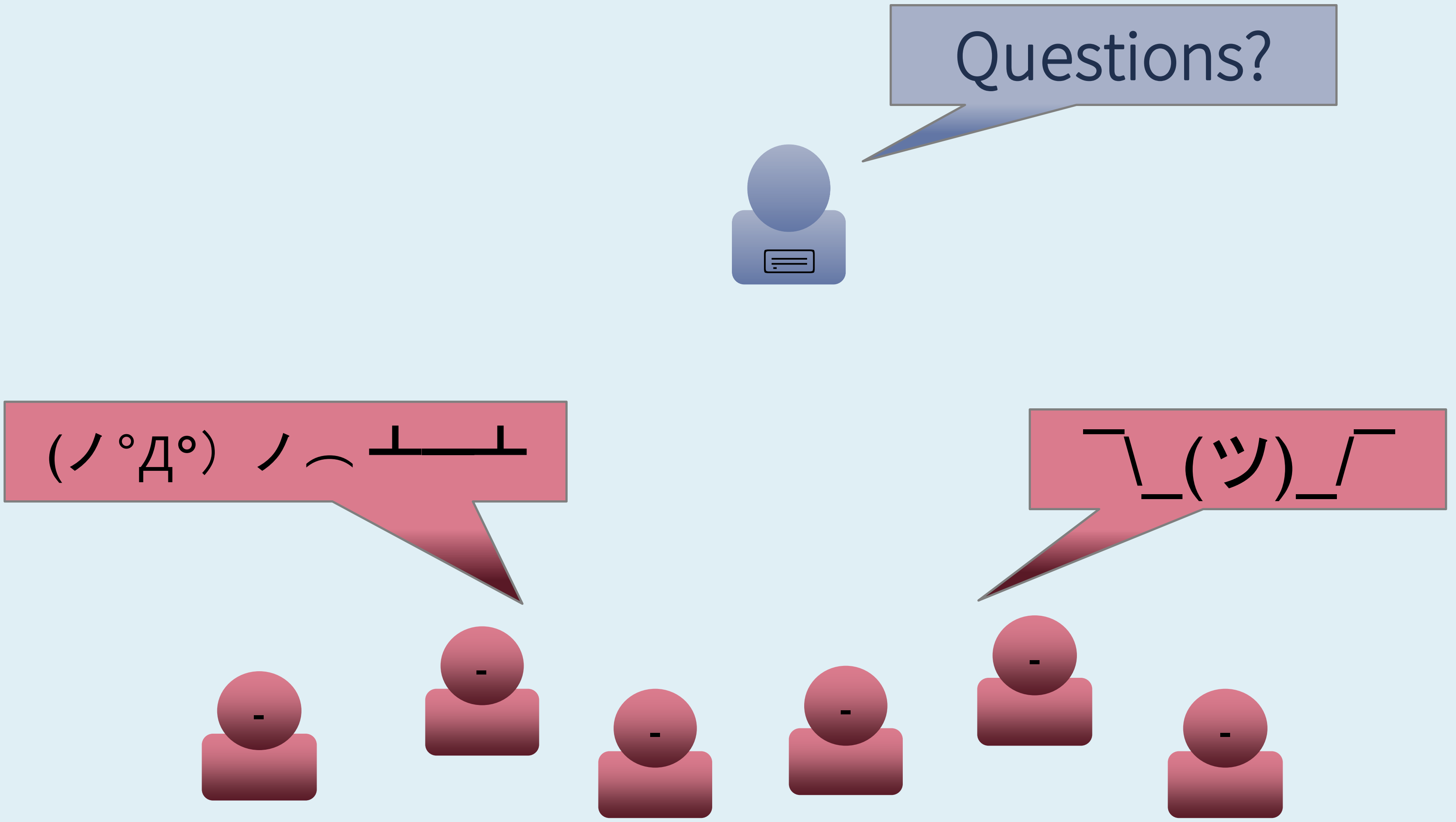
# Other options if all else is lost

- **Break some machines?**


From “Office Space”
Credit: Mike Judge Film Co/ Twentieth Century Fox/Alamy

# Literature


- Blagus, Lusa: Class prediction for high-dimensional class-imbalanced data. BMC Bioinformatics, 2010.
- Chawla et al: SMOTE: Synthetic Minority Over-Sampling Technique. J. Artificial Intelligence Research, 2002.
- Galar et al: A Review on Ensembles for the Class Imbalance Problem - Bagging Boosting and Hybrid-based Approaches. IEEE, 2011.
- He, Garcia: Learning from Imbalanced Data. IEEE, 2009
- Lin, Chen: Class-imbalanced classifiers for high-dimensional data. Briefings in Bioinformatics, 2012.
- Lopez et al: An insight into classification with imbalanced data Empirical results and current trends on using data intrinsic characteristics. Information Sciences, 2013.
- McCarthy et al: Does Cost-Sensitive Learning Beat Sampling for Classifying Rare Classes? UBDM , 2005.
- Schölkopf et al: Estimating the support of a high-dimensional distribution. Neural computation, 2001.


Questions?
(ノ°Д°）ノ︵┻━┻
¯_(ツ)_/¯