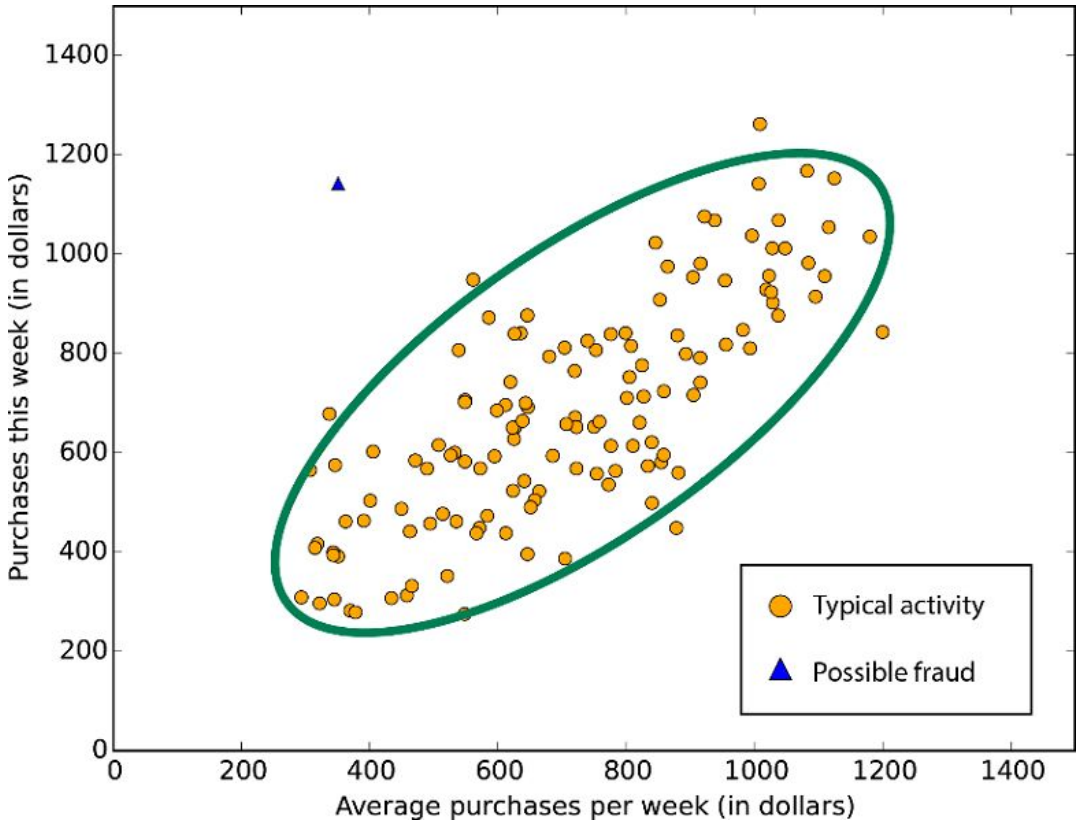
Practical AI: classification practicum

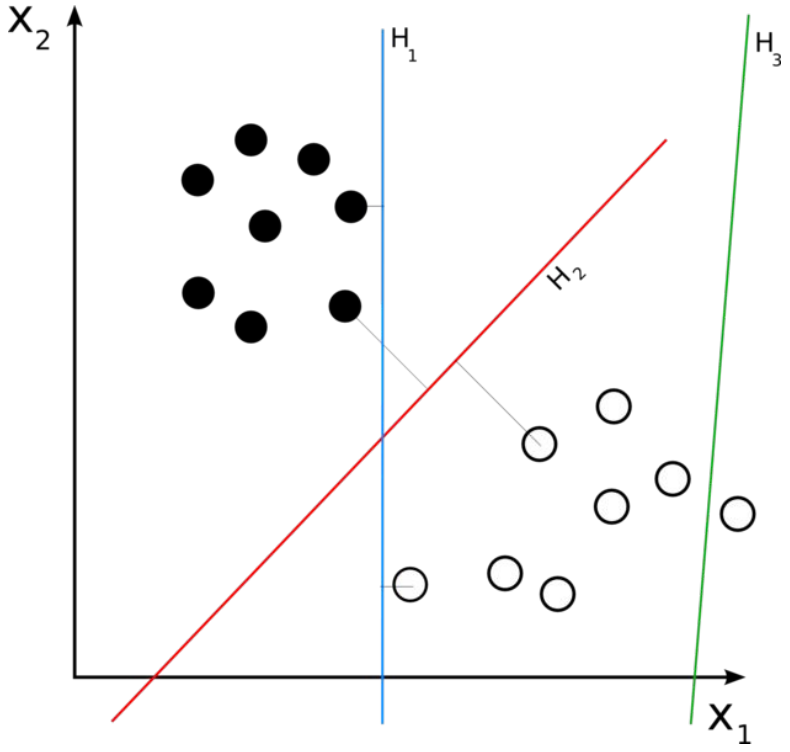
Stanislav Protasov for
Harbour.Space University

Agenda

- One-class, binary classification
- Measuring quality:
  - accuracy,
  - precision,
  - recall,
  - F1,
  - ROC/AuC
- Linear model
  - Logistic Regression
- SVM
- ANN
- Multiclass approaches
  - One-vs-all
  - kNN classifier
  - ANN classifier

One-class and binary classification





How to measure

Accuracy - `success / total`

Precision - we care about **correct detections** (how many of detections are correct). `successfully detected / total detections`

Recall = TPR - FOMO :) We care about **missing** (we are penalized for missing). `successfully detected / total positive`

FPR - “panic rate”. How many of predictions made are done for “panic”.
`false positive / total detections`

F_1-score is an integral metric for precision-recall (harmonic mean).

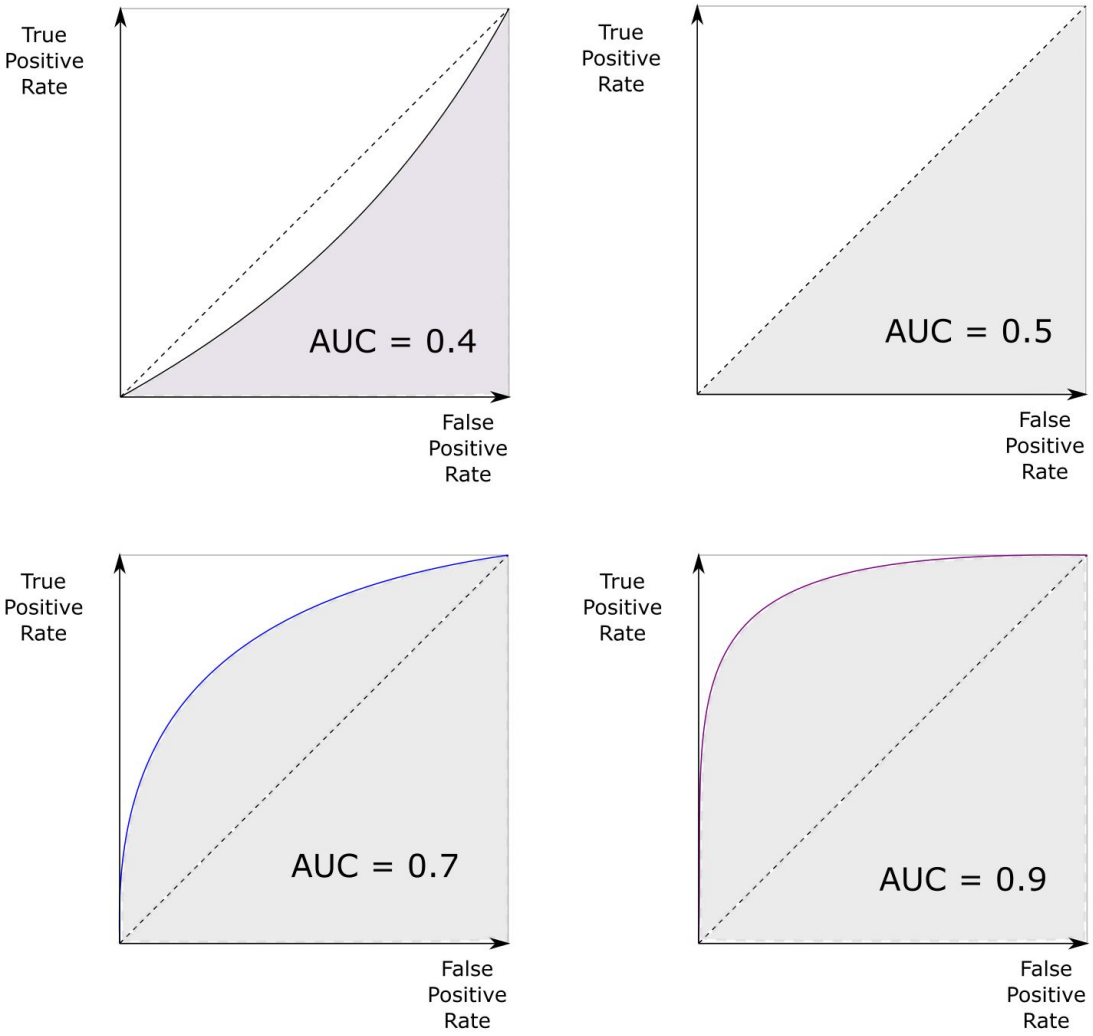
F_1 = 2 * Pr * Re / (Pr + Re), lays in [0..1]

ROC, AuC

ROC - receiver operating characteristic.
A plot **TPR** vs **FPR**.

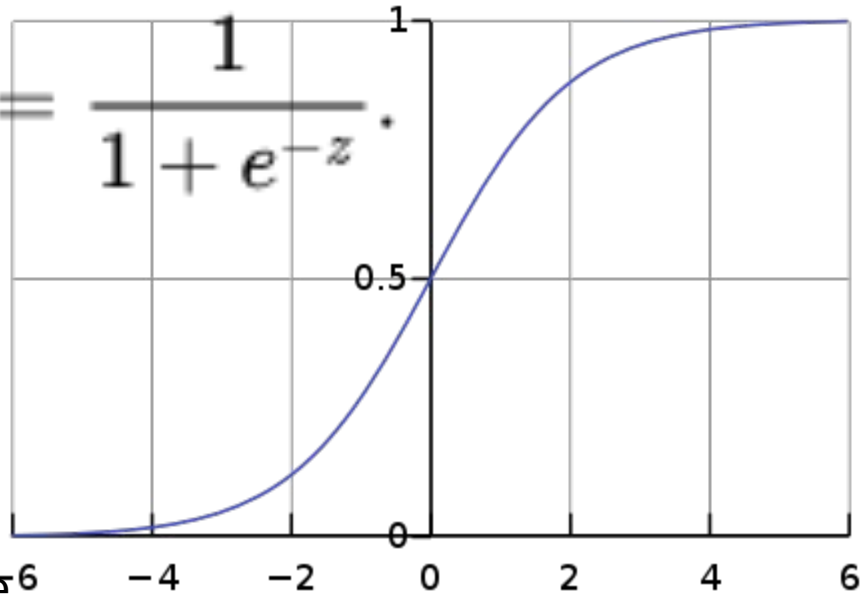
Idea: if we **tune a border** of classifier to bring **more true positive** detections, **how many false positives** it will bring along with?

Area under Curve - single number to describe ROC.



Logistic Regression

$$f(z) = \frac{1}{1 + e^{-z}}.$$



$$\mathbb{P}\{y = 1 \mid x\} = f(z),$$
$$\mathbb{P}\{y = 0 \mid x\} = 1 - f(z) = 1 - f(\theta^T x).$$

P stands for probability that even X is true.

f(z) stands for prediction of probability.

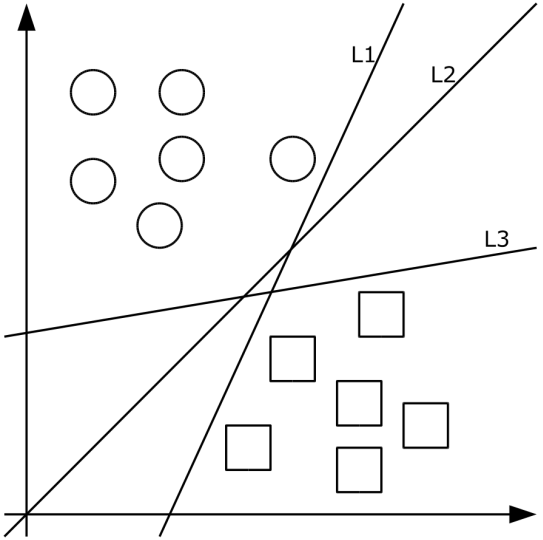
$$\mathbb{P}\{y \mid x\} = f(\theta^T x)^y (1 - f(\theta^T x))^{1-y}, \quad y \in \{0, 1\}.$$
$$\hat{\theta} = \operatorname{argmax}_\theta L(\theta) = \operatorname{argmax}_\theta \prod_{i=1}^{m} \mathbb{P}\{y = y^{(i)} \mid x = x^{(i)}\}.$$

We can, again train Logistic Regression with different algorithms.

LR with X-val, LR with SGD, just LR

Andrew Ng is cool:
https://see.stanford.edu/materials/aimlcs229/cs229-notes1.pdf

Support vector machine

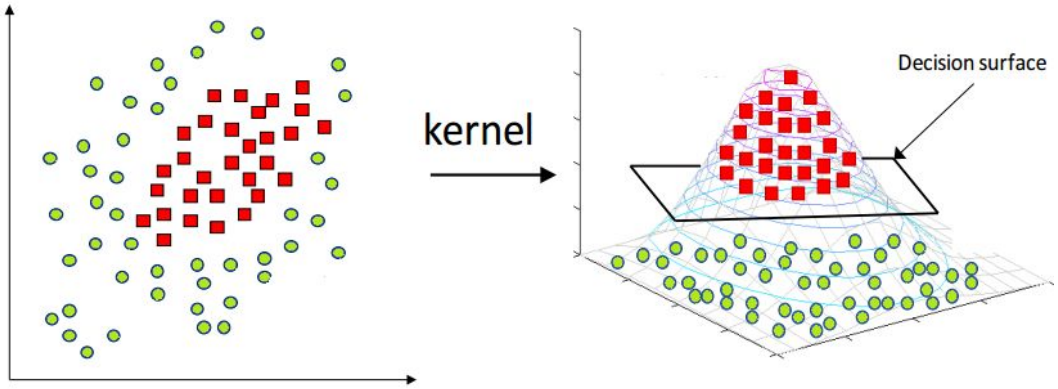


General idea: classifier has better generalization power if it preserves biggest possible GAP between classes

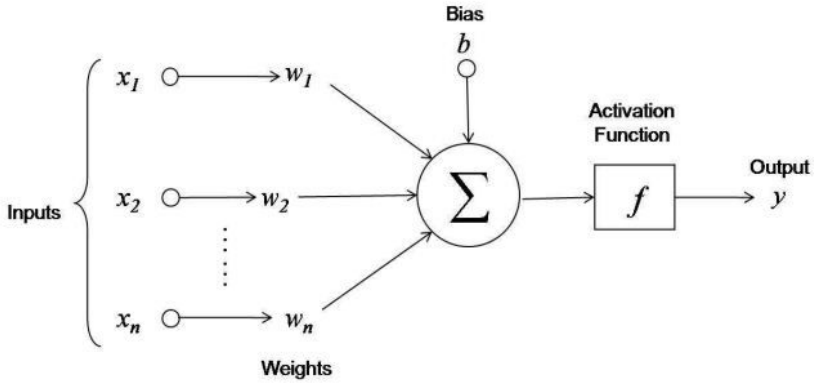
SVM itself is well known for involving **kernel trick**.

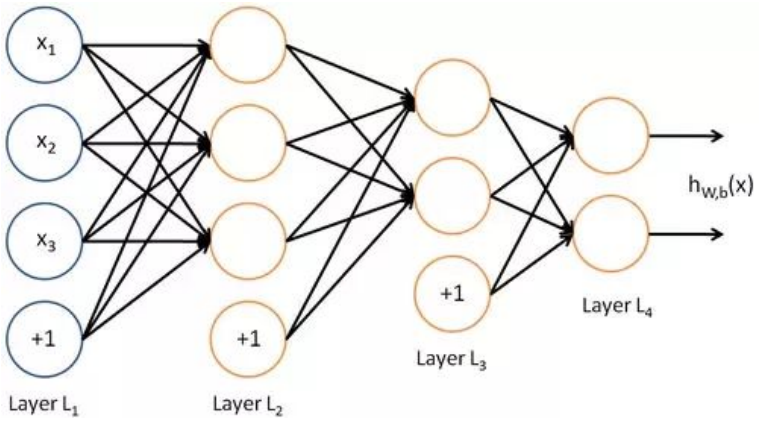
Idea: we have raw data, and instead of **computing non-linear features** we can introduce KERNEL - a **non-linear** function to compute sample **distance**. Thus, there IS another space, but it is IMPLICIT.

Then model builds borders for a class by summing kernelized distances.



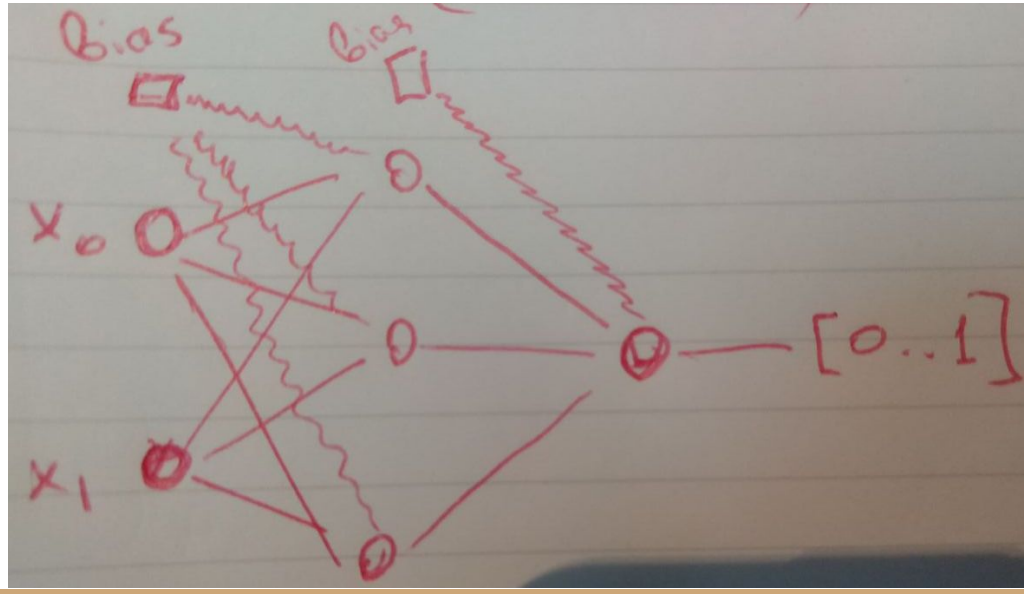
ANN: whiteboard time





Lab #1. ANN classifier lab

1. Consider and run (?) an example on cats-vs-dogs classifier
   a. Study metrics. Do you understand them all?
   b. Study parameters of MLP Classifier
2. Implement MLP Classifier for XOR function with given topology:
   a. Generate dataset
   b. Use ReLU activation function
   c. Train
   d. Measure accuracy, study quality report



Detectors

General idea: for a stream of data run unary of binary classifier.

For image:

- Build image pyramid
- Run for each patch of accepted size a classifier
- Where detects, create bounding box

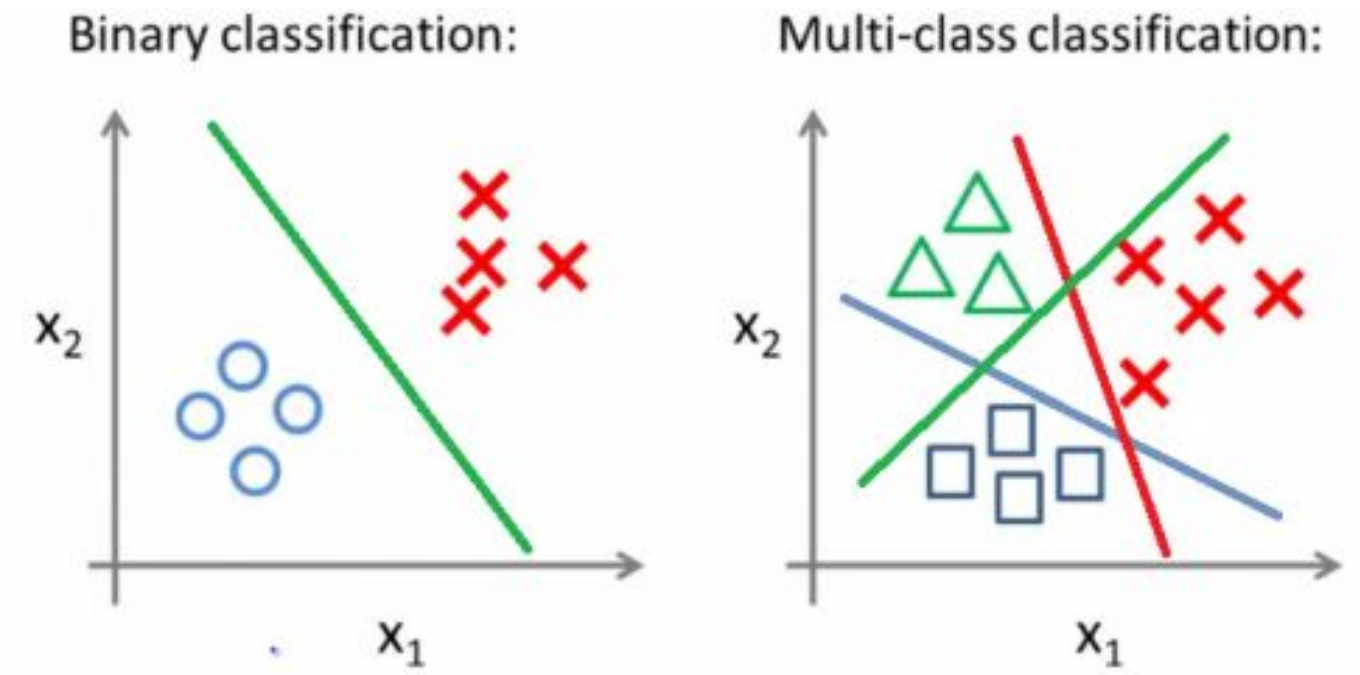
Run this example to understand the idea. Try different scale factor

Multiclass classifers

One-vs-all

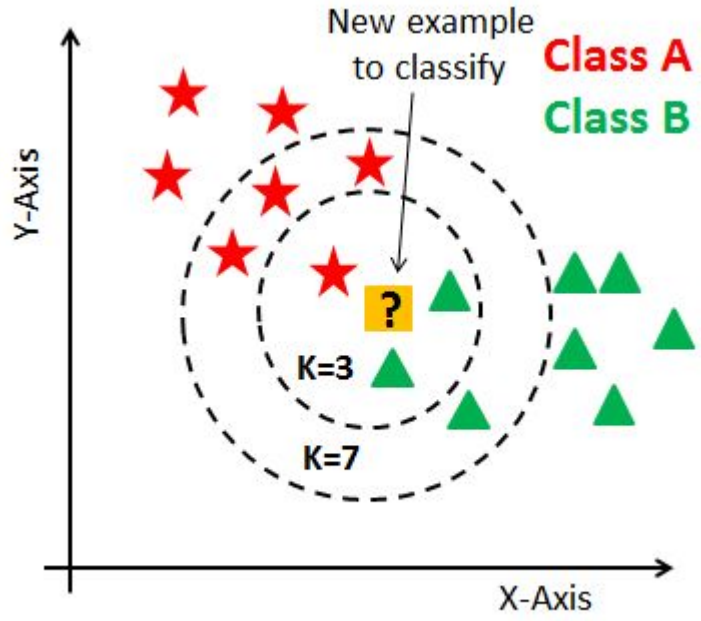
Use any binary classifier.

E.g. this is how it works for LogisticRegression.



kNN classifier

Distance function is important. Result will depend on data normalization. Use “pyfunc” if you want to use your own.



Lab #2

Study example with grid search for kNN.

Train and save the best **ANN classifier**.

Homework

Recognize hand-written number

https://github.com/hsu-ai-course/hsu.ai/tree/master/homeworks/12