

# General Framework for Classification at the Top

---

**Václav Mácha**

29/06/2023

Czech Technical University in Prague  
Faculty of Nuclear Sciences and Physical Engineering  
Department of Mathematics

# Motivation

---

# Binary classification

- Two group of samples:
  - negative samples with label  $y = 0$ ,
  - positive samples with label  $y = 1$ .

# Binary classification

- Two group of samples:
  - negative samples with label  $y = 0$ ,
  - positive samples with label  $y = 1$ .
- Classifier usually has two parts:
  - model  $f$  maps a sample  $x$  to its classification score  $s \in \mathbb{R}$ ,
  - decision threshold  $t \in \mathbb{R}$  decides whether a sample is classified as positive or not

$$\hat{y} = \begin{cases} 1 & \text{if } s \geq t, \\ 0 & \text{otherwise.} \end{cases}$$

- Usually, the model returns the probability that the sample is from the positive class and the threshold is 0.5.

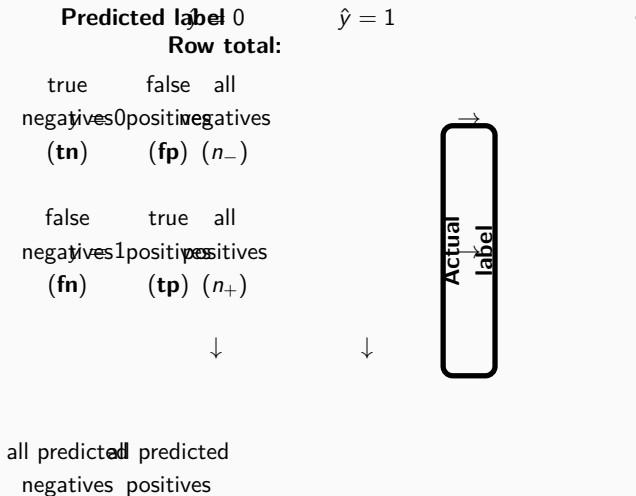
# Binary classification

- Two group of samples:
  - negative samples with label  $y = 0$ ,
  - positive samples with label  $y = 1$ .
- Classifier usually has two parts:
  - model  $f$  maps a sample  $x$  to its classification score  $s \in \mathbb{R}$ ,
  - decision threshold  $t \in \mathbb{R}$  decides whether a sample is classified as positive or not

$$\hat{y} = \begin{cases} 1 & \text{if } s \geq t, \\ 0 & \text{otherwise.} \end{cases}$$

- Usually, the model returns the probability that the sample is from the positive class and the threshold is 0.5.
- **Goal:** find classifier which predicts labels for unknown samples with the lowest possible error.

# Confusion matrix



# Binary Classification

- General form of binary classification

$$\begin{aligned} \underset{\mathbf{w}, t}{\text{minimize}} \quad & \frac{1}{n_-} \sum_{i \in \mathcal{I}_-} \mathbb{1}_{[s_i \geq t]} + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} \mathbb{1}_{[s_i < t]} \\ \text{subject to} \quad & s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I}, \end{aligned}$$

- $\mathcal{I} = \mathcal{I}_- \cup \mathcal{I}_+$  is a set of indices of all sample where

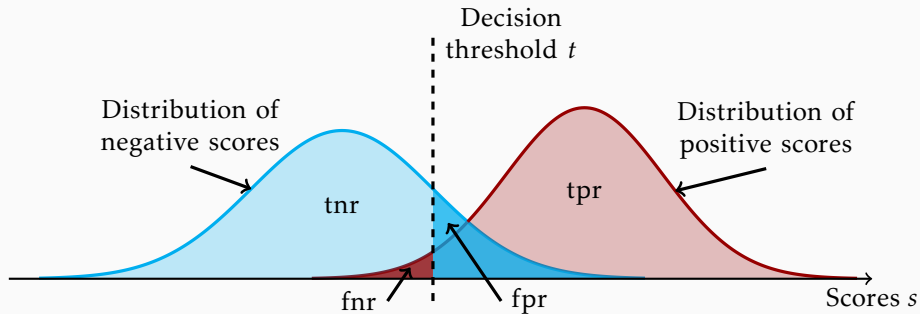
$$\mathcal{I}_- = \{i \mid i \in \{1, 2, \dots, n\} \wedge y_i = 0\},$$

$$\mathcal{I}_+ = \{i \mid i \in \{1, 2, \dots, n\} \wedge y_i = 1\},$$

- $\mathbb{1}_{[\cdot]}$  is Iverson function defined by

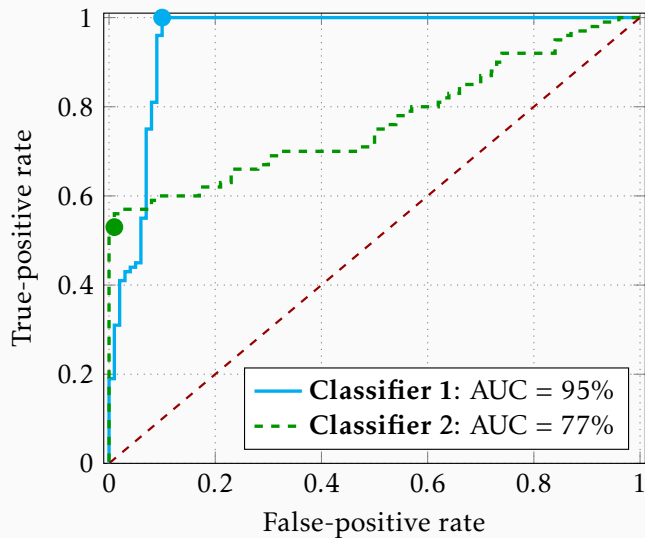
$$\mathbb{1}_{[x]} = \begin{cases} 0 & \text{if } x \text{ is false,} \\ 1 & \text{if } x \text{ is true.} \end{cases}$$

# ROC curves





# ROC curves



## Classifier 1 is better ... or not?

- If the goal is to classify all samples with the lowest possible error then **Classifier 1** is better.

## Classifier 1 is better ... or not?

- If the goal is to classify all samples with the lowest possible error then **Classifier 1** is better.
- What if some samples are more relevant than others?
  - Search engines: relevant results should be on the first few pages.
  - Malware detection: false-alarms are disruptive to the user.
  - Expensive post-processing: development of new drugs.

## Classifier 1 is better ... or not?

- If the goal is to classify all samples with the lowest possible error then **Classifier 1** is better.
- What if some samples are more relevant than others?
  - Search engines: relevant results should be on the first few pages.
  - Malware detection: false-alarms are disruptive to the user.
  - Expensive post-processing: development of new drugs.
- In such cases, **Classifier 2** may be better.

## Classifier 1 is better ... or not?

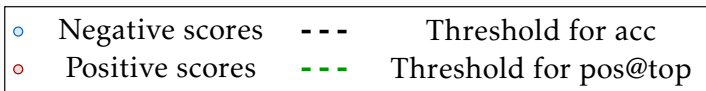
- If the goal is to classify all samples with the lowest possible error then **Classifier 1** is better.
- What if some samples are more relevant than others?
  - Search engines: relevant results should be on the first few pages.
  - Malware detection: false-alarms are disruptive to the user.
  - Expensive post-processing: development of new drugs.
- In such cases, **Classifier 2** may be better.
- **Classifier 1** maximizes accuracy

$$\text{acc}(\mathbf{s}) = \frac{1}{n} \sum_{i \in \mathcal{I}} \mathbb{1}_{[y_i = \hat{y}_i]}.$$

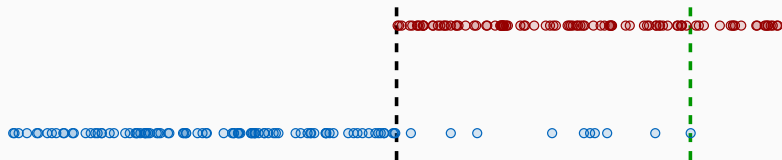
- **Classifier 2** maximizes the number of positive samples at the top

$$\text{pos@top}(\mathbf{s}) = \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} \mathbb{1}_{[s_i \geq \max_{j \in \mathcal{I}_-} s_j]}.$$

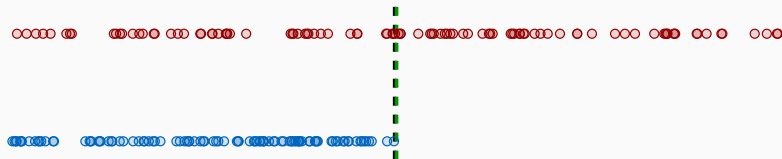
# ROC curves



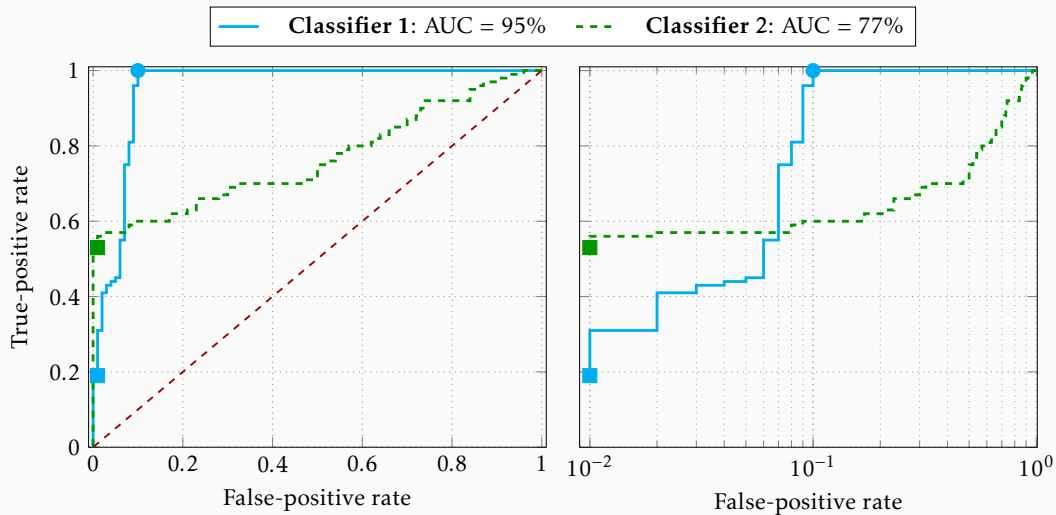
**Classifier 1**  
acc 95%  
pos@top 19%



**Classifier 2**  
acc 76%  
pos@top 53%



# ROC curves



## Classification at the Top

---



## Problem formulation

- **Goal:** classify correctly only the most relevant samples.

## Problem formulation

- **Goal:** classify correctly only the most relevant samples.
- The most relevant samples are samples with the highest scores.

## Problem formulation

- **Goal:** classify correctly only the most relevant samples.
- The most relevant samples are samples with the highest scores.
- General formulation

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{n_-} \sum_{i \in \mathcal{I}_-} \mathbb{1}_{[s_i \geq t]} + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} \mathbb{1}_{[s_i < t]} \\ \text{subject to} \quad & s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I}, \\ & t = G(\mathbf{s}, \mathbf{y}), \end{aligned}$$

where threshold  $t$  is a function of all scores.

# Problem formulation

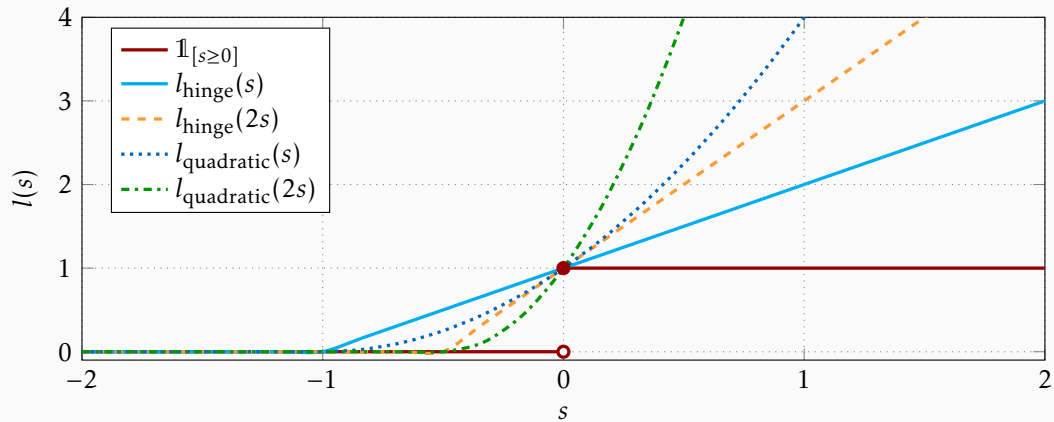
- **Goal:** classify correctly only the most relevant samples.
- The most relevant samples are samples with the highest scores.
- General formulation

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{n_-} \sum_{i \in \mathcal{I}_-} \mathbb{1}_{[s_i \geq t]} + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} \mathbb{1}_{[s_i < t]} \\ \text{subject to} \quad & s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I}, \\ & t = G(\mathbf{s}, \mathbf{y}), \end{aligned}$$

where threshold  $t$  is a function of all scores.

- Difficult problem: constrained, discontinuous, non-convex, and non-decomposable.

# ROC curves



# Surrogate approximation

- General surrogate formulation

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} && \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ & \text{subject to} && s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I}, \\ & && t = G(\mathbf{s}, \mathbf{y}). \end{aligned}$$

# Surrogate approximation

- General surrogate formulation

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} && \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ & \text{subject to} && s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I}, \\ & && t = G(\mathbf{s}, \mathbf{y}). \end{aligned}$$

- TopPush* maximizes the number of positive samples at the top

$$t = \max_{j \in \mathcal{I}_-} s_j.$$

# Surrogate approximation

- General surrogate formulation

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} && \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ & \text{subject to} && s_i = f(\mathbf{x}_i; \mathbf{w}), \quad i \in \mathcal{I}, \\ & && t = G(\mathbf{s}, \mathbf{y}). \end{aligned}$$

- TopPush* maximizes the number of positive samples at the top

$$t = \max_{j \in \mathcal{I}_-} s_j.$$

- Pat&Mat-NP* maximizes true-positive rate with fixed false-positive rate

$$t \text{ solves } \frac{1}{n_-} \sum_{i \in \mathcal{I}_-} l(\vartheta(s_i - t)) = \tau.$$



## Classification at the Top: Linear Model

---

- Linear model  $f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \mathbf{x}$ .
- General surrogate formulation with linear model

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ \text{subject to} \quad & s_i = \mathbf{w}^\top \mathbf{x}_i, \quad i \in \mathcal{I}, \\ & t = G(\mathbf{s}, \mathbf{y}). \end{aligned}$$

- Properties that we are interested in:
  - Convexity of the objective function.
  - Robustness to outliers.

## Theorem

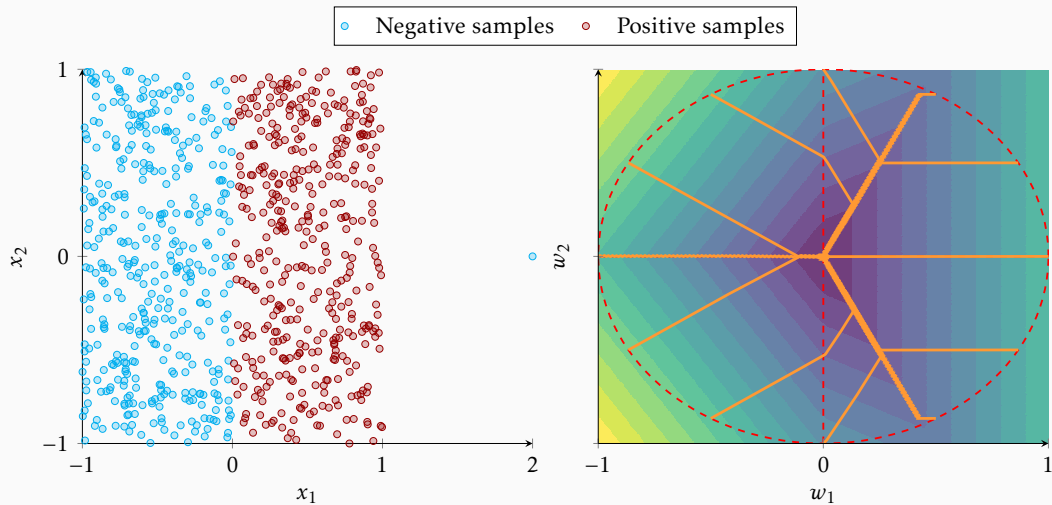
If the threshold  $t$  is a convex function of the weights  $\mathbf{w}$ , then function

$$L(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i)$$

is convex.

- Both formulations *TopPush* and *Pat&Mat-NP* have convex thresholds.
- Both formulations are convex and continuous.

## When convexity is not enough...



# How to solve it?

- Using gradient descent

$$\mathbf{w}^{k+1} \leftarrow \mathbf{w}^k - \alpha^k \cdot \nabla L(\mathbf{w}^k),$$

where  $\alpha^k > 0$  is a learning rate, and  $\nabla L(\mathbf{w}^k)$  is a gradient of the objective function

$$\nabla L(\mathbf{w}) = \mathbf{w} + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l'(t(\mathbf{w}) - f(\mathbf{x}_i; \mathbf{w})) (\nabla t(\mathbf{w}) - \nabla f(\mathbf{x}_i; \mathbf{w})).$$

## **Classification at the Top: Non-linear Model**

---

- General surrogate formulation with non-linear model

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ \text{subject to} \quad & s_i = \mathbf{w}^\top \mathbf{x}_i, \quad i \in \mathcal{I}, \\ & t = G(\mathbf{s}, \mathbf{y}). \end{aligned}$$

- General surrogate formulation with non-linear model

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ \text{subject to} \quad & s_i = \mathbf{w}^\top \mathbf{x}_i, \quad i \in \mathcal{I}, \\ & t = G(\mathbf{s}, \mathbf{y}). \end{aligned}$$

- Disadvantages:
  - Objective function is not convex.
  - Non-linear models are usually large and expensive to train.



- General surrogate formulation with non-linear model

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ \text{subject to} \quad & s_i = \mathbf{w}^\top \mathbf{x}_i, \quad i \in \mathcal{I}, \\ & t = G(\mathbf{s}, \mathbf{y}). \end{aligned}$$

- Disadvantages:
  - Objective function is not convex.
  - Non-linear models are usually large and expensive to train.
- What to do if the dataset is too large to fit in memory?

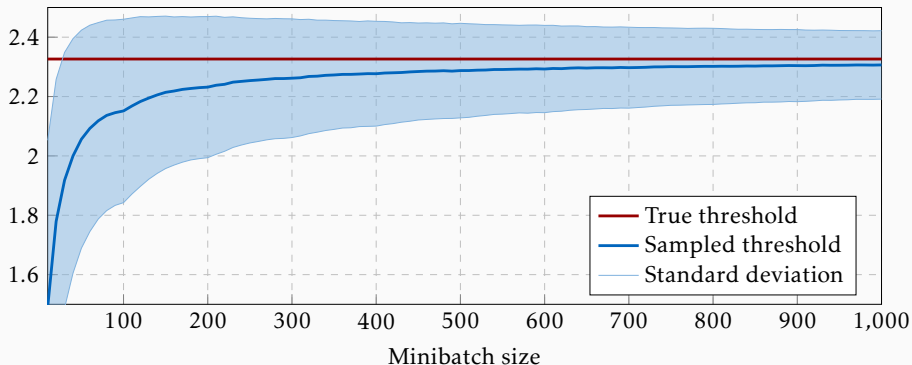
- General surrogate formulation with non-linear model

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{n_+} \sum_{i \in \mathcal{I}_+} l(t - s_i) \\ \text{subject to} \quad & s_i = \mathbf{w}^\top \mathbf{x}_i, \quad i \in \mathcal{I}, \\ & t = G(\mathbf{s}, \mathbf{y}). \end{aligned}$$

- Disadvantages:
  - Objective function is not convex.
  - Non-linear models are usually large and expensive to train.
- What to do if the dataset is too large to fit in memory?
- Stochastic gradient descent: the gradient is computed only on a small subset of all data called minibatch.

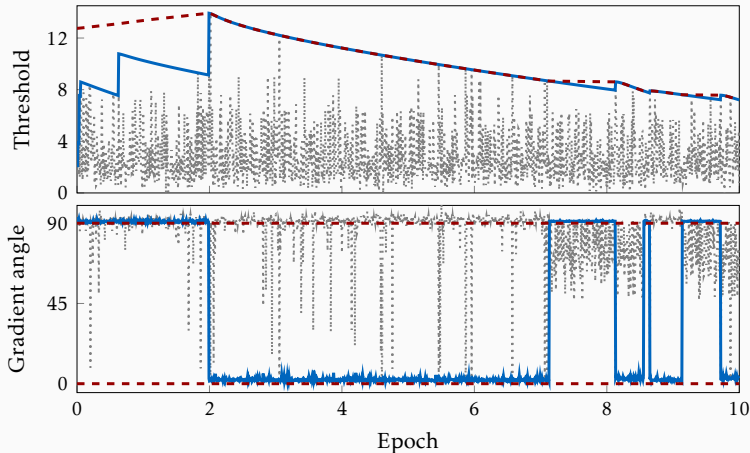
# Issues when passing to mini-batches

- Problems:
  - The threshold is a function of all scores  $\rightarrow$  the loss function is non-decomposable.
  - As a result, stochastic gradient descent provides a biased gradient estimate.



# How to reduce bias?

- Increase size of minibatch  $\rightarrow$  *Pat&Mat-NP*.
- Add threshold from last minibatch  $\rightarrow$  *DeepTopPush*.

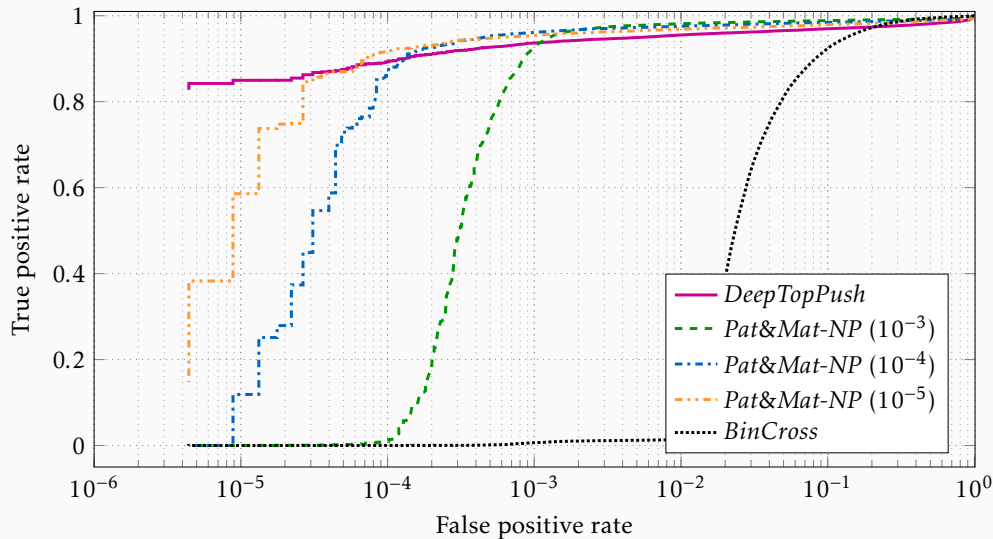


**How does it work?**

---

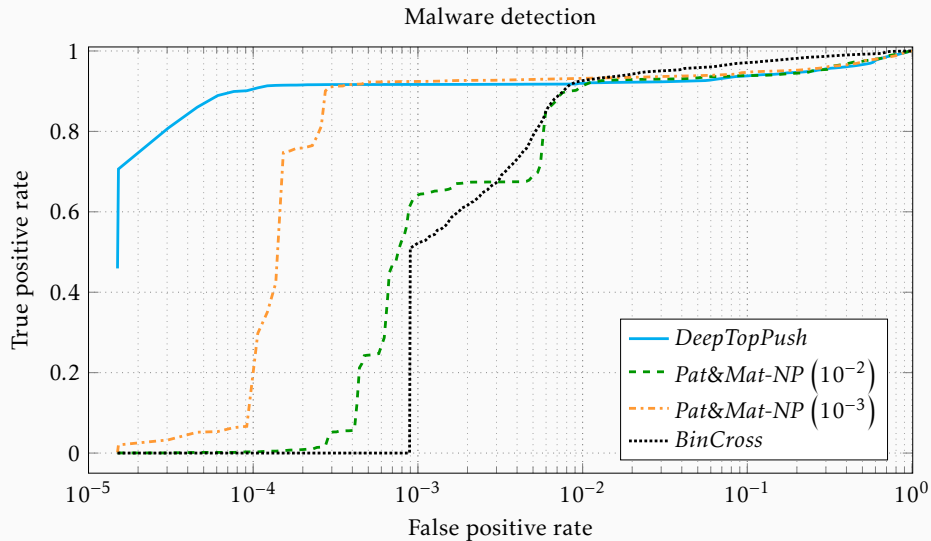
- Dataset: 205 579 samples, 9% of samples are positive.
- Each sample consists of 22 510 features.
- Only linear model.
- **Goals:**
  - Maximize the true-positive rate at extremely low levels of the false-positive rate.

# Steganography



- Dataset: 6 580 166 samples, 87% of samples are positive.
- Hierarchical data structure:
  - Each sample is a JSON file, which may consist of other JSON files.
  - Each sample is of a different size (from 1 KB to 2.5 MB).
  - *DeepTopPush* and *Pat&Mat-NP* used as an extension for hierarchical multi-instance learning (HMIL).
- **Goals:**
  - Maximize the true-positive rate at extremely low levels of the false-positive rate.
  - The false-positive rate must be as low as possible to avoid disruptive false alarms for the end-user.





**Thank you for your attention.**