



Rise of the Machines

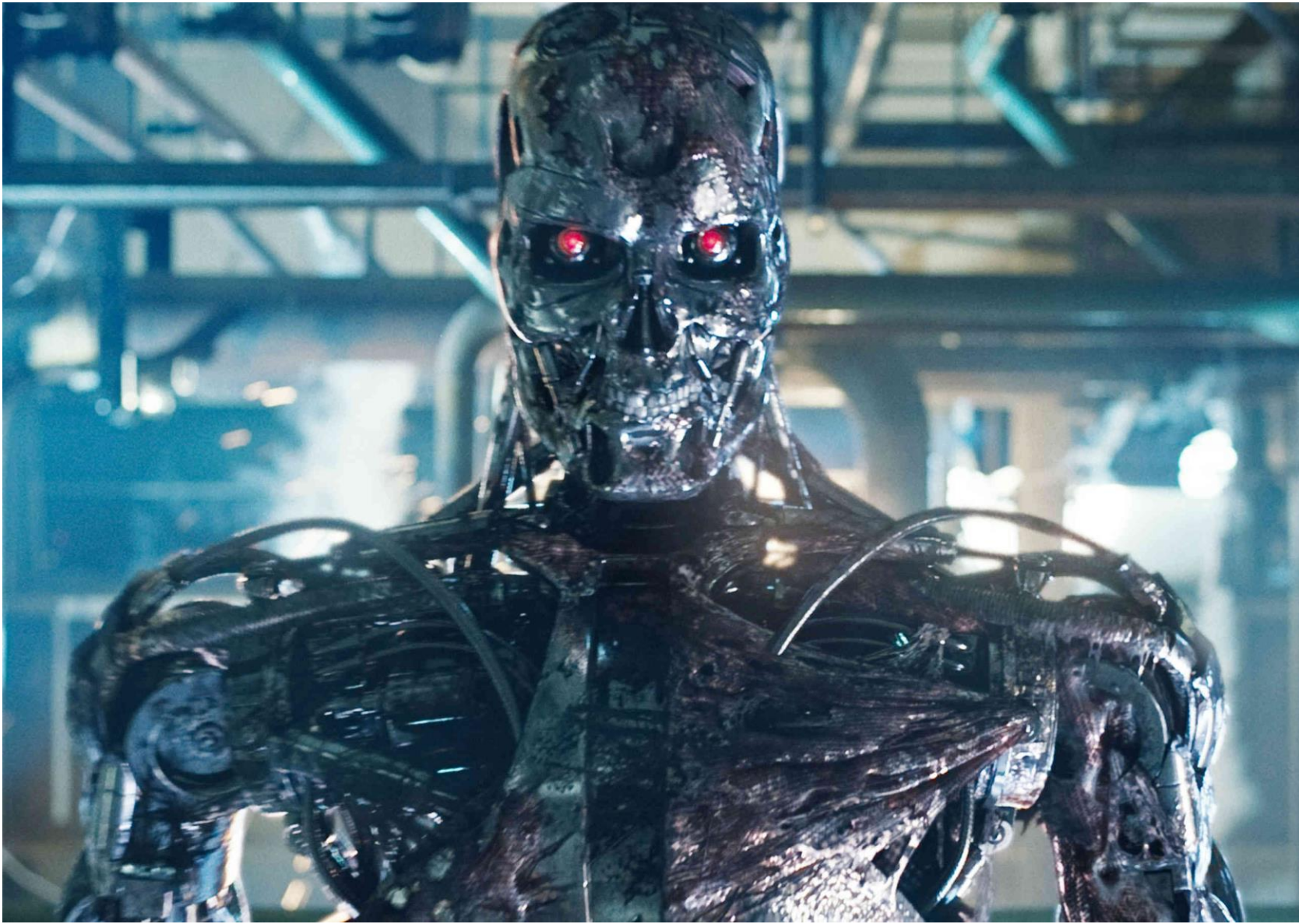
Uczenie Maszynowe w Praktyce

Tomasz Szumlak

Seminarium WFiIS AGH
22/06/2018, Kraków

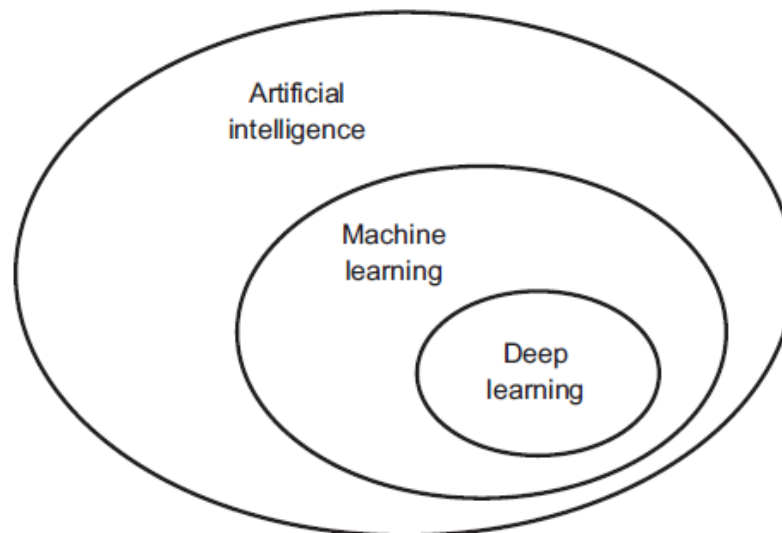
Outline

- ❑ Dlaczego teraz...?
- ❑ Przygarść historii
- ❑ LHCb
- ❑ Spin-off



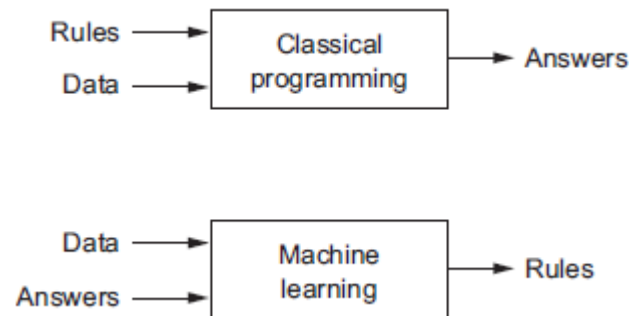
Setting the scene

- ❑ **Uczenie Maszynowe (Machine Learning)**
 - ❑ To nie jest nowy pomysł: lata 50 XX wieku
 - ❑ Część dyscypliny wiedzy związanej z poszukiwaniem możliwości stworzenia sztucznej inteligencji: „czy możemy zautomatyzować zadania wymagające intelektu (ludzkiego)”
 - ❑ Symboliczna AI – dostarczamy „zasad” posługiwania się wiedzą, przy pewnej liczbie zasad osiągamy „masę krytyczną” czyli AI



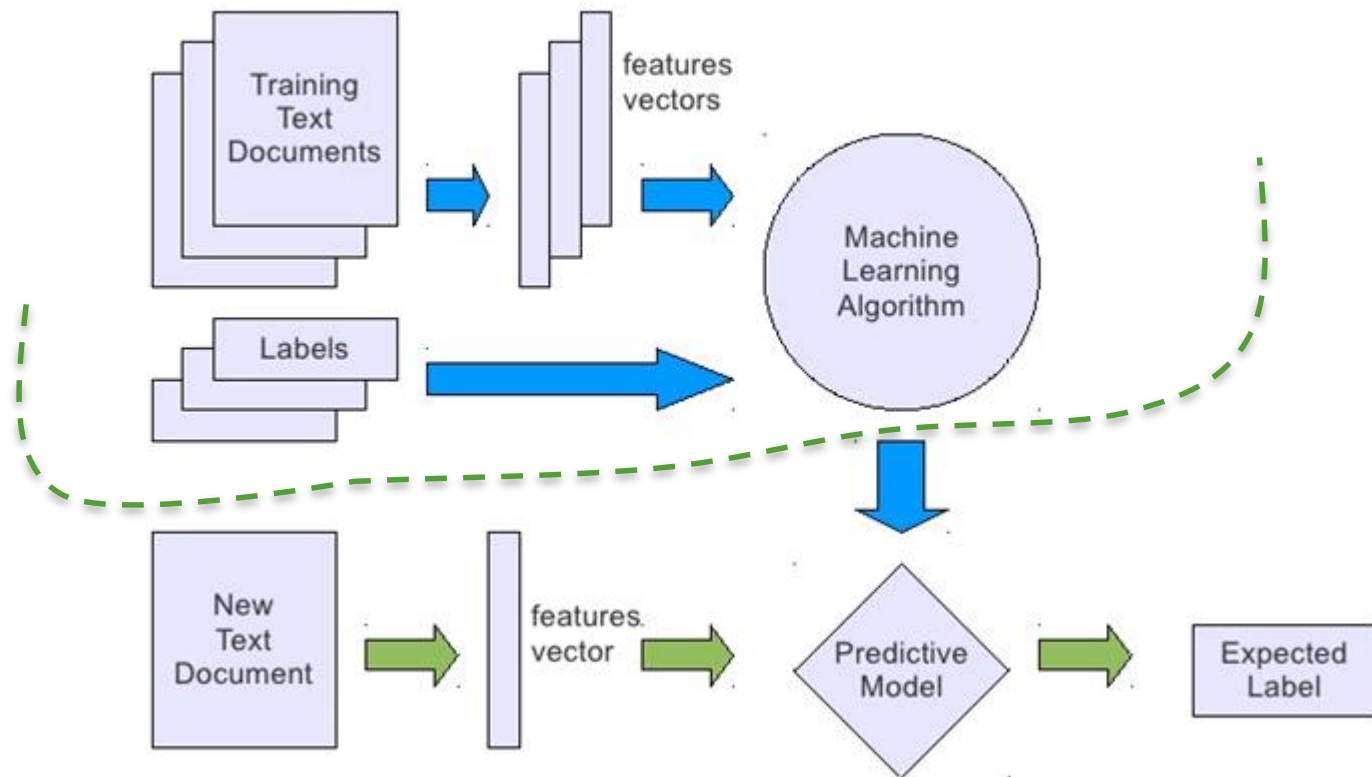
Setting the scene

- ❑ Uczenie Maszynowe „in a nutshell”
- ❑ Klasyczne programowanie:
 - ❑ Dane + **Algorytm („zasady”)** → Odpowiedzi
- ❑ UM odwraca trochę porządek:
 - ❑ Dane + Odpowiedzi = **Algorytm („zasady”)**
 - ❑ **Poza fazą kodowania, mamy dodatkowo fazę treningu!**



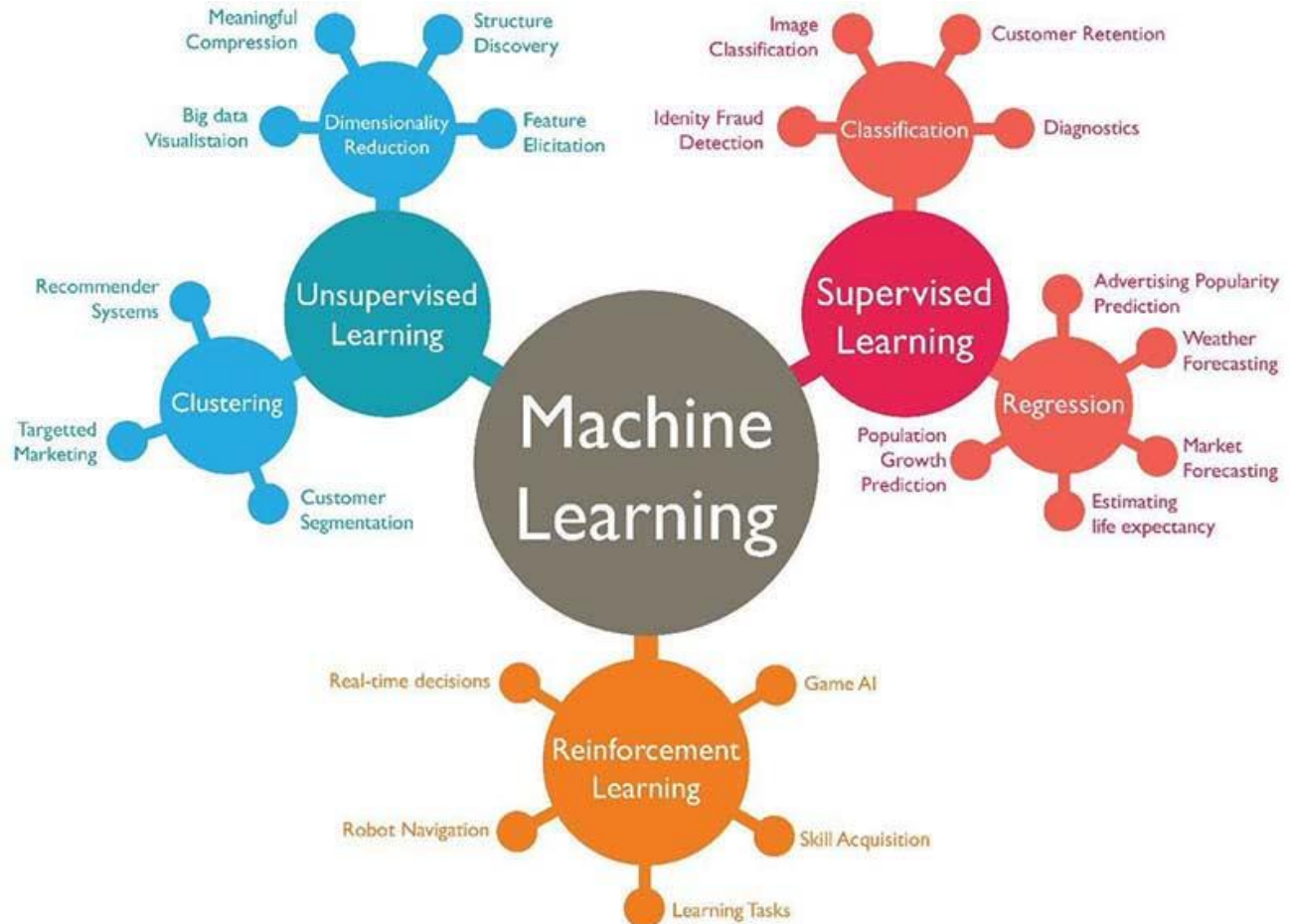
Setting the scene

- Przygotowanie i użytkowanie oprogramowania



Setting the scene

□ Podział technik uczenia maszynowego



Dlaczego teraz...?



- Moim zdaniem masę krytyczną stworzyły:

Dlaczego teraz...?

- Moim zdaniem masę krytyczną stworzyły:



Dlaczego teraz...?

- Moim zdaniem masę krytyczną stworzyły:



Dlaczego teraz...?

- **Moim zdaniem** masę krytyczną stworzyły:



Caffe2

Microsoft CNTK

TensorFlow

Chainer

PYTORCH

dmic mxnet

**Eksperyment LHCb generuje
ok 1 PB/miesiąc**

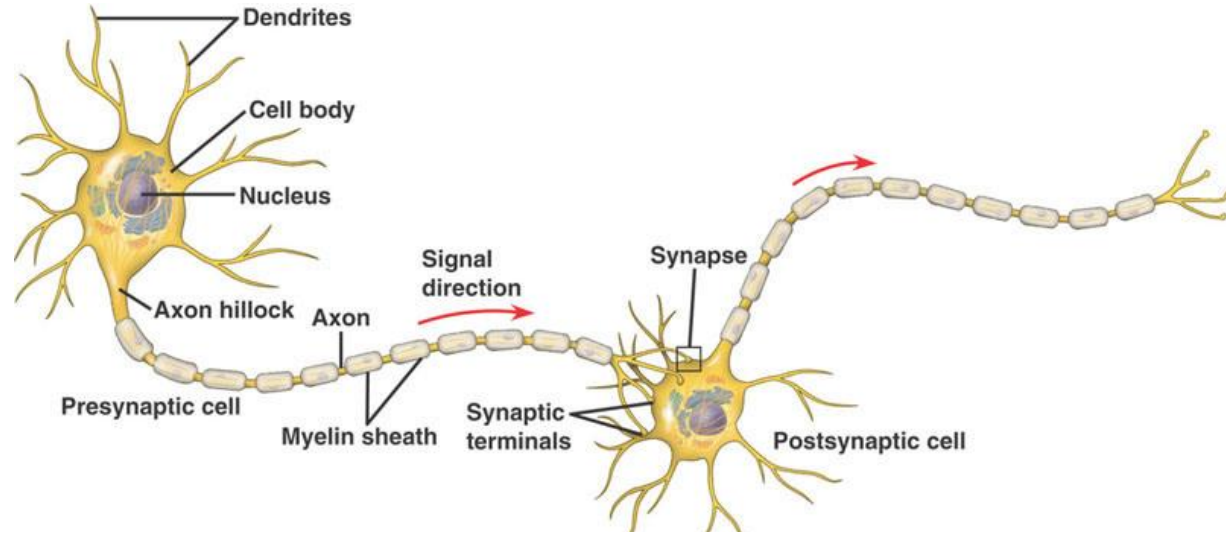
Historia – perceptron

- ❑ Przygoda zaczyna się w **1943** kiedy to McCulloch-Pitts proponują **model neuronu**
- ❑ Używają do tego **komórki nerwowej** i opisu bazującego **cyfrowej bramce logicznej** z binarnym wyjściem – taki **perceptron** (sztuczny neuron) dostaje na wejściu wiele sygnałów, **dokonuje ich „asocjacji”** i jeżeli sygnał otrzymany przekracza odpowiednio dobrany próg perceptron jest w stanie wyprodukować **sygnał wyjściowy**

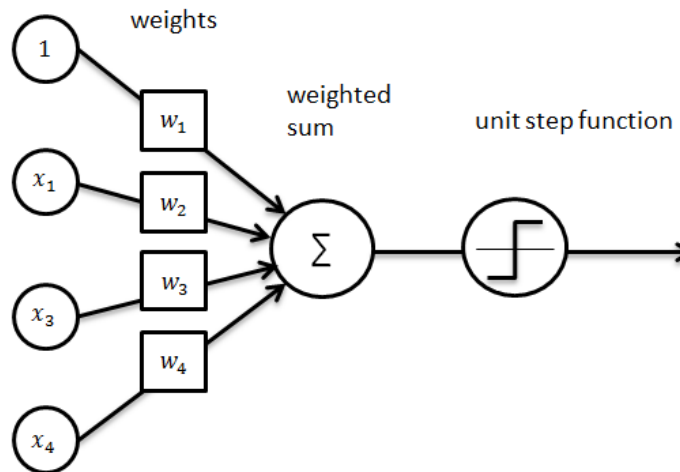
Historia – perceptron

- ❑ Przygoda zaczyna się w **1943** kiedy to McCulloch-Pitts proponują **model neuronu**
- ❑ Używają do tego **komórki nerwowej** i opisu bazującego **cyfrowej bramce logicznej** z binarnym wyjściem – taki **perceptron** (sztuczny neuron) dostaje na wejściu wiele sygnałów, **dokonuje ich „asocjacji”** i jeżeli sygnał otrzymany przekracza odpowiednio dobrany próg perceptron jest w stanie wyprodukować **sygnał wyjściowy**
- ❑ Badania nad tego typu strukturami doprowadziły do opracowania metody uczenia (1957) przez **Rosenblatt’a**
- ❑ Zasada uczenia Rosenblatt’a pozwala na automatyczne dobranie najlepszych wag, które kontrolują odpowiedź perceptronu

Perceptron model



inputs




Zasada Rosenblatt'a

- ❑ W tym podejściu, używamy nauczania nadzorowanego do **optymalizacji (trenowania)** wag
- ❑ Wagi kombinowane są z danymi wejściowymi (cechami) i biorąc pod uwagę otrzymany wynik perceptron podejmuje decyzję o odpowiedzi (fire/not fire)
- ❑ Tak wytrenowany perceptron może następnie podejmować decyzje w odniesieniu do przynależności danych do klas (klasyfikacja)
- ❑ Na przykład, bardzo często (analiza danych HEP) spotykamy się z problemem klasyfikacji binarnej, czyli mamy dwie klasy przypadków **negatywne** -1 (szum) oraz **pozytywne** $+1$ (sygnał)
- ❑ Zachowanie perceptronu kontroluje **funkcja aktywacji**: $\phi(z)$

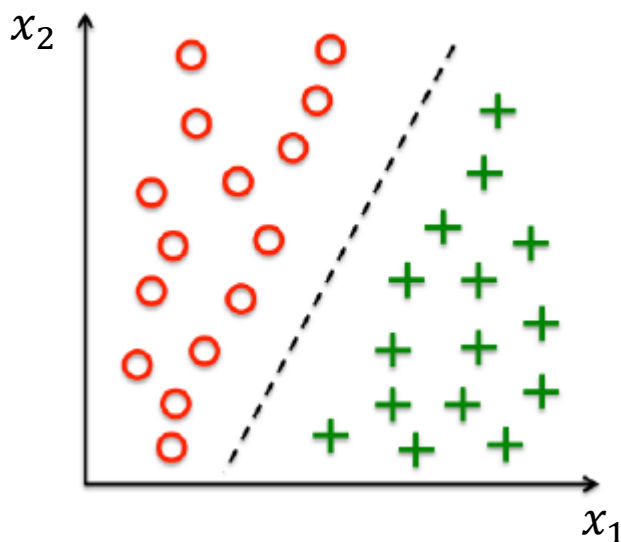
$$\phi(z) = \begin{cases} +1 & \text{if } z \geq \theta \\ -1 & \text{if } z < \theta \end{cases}$$

Próg decyzji



Klasyfikacja Binarna

- Typowy problem przy analizie danych: mamy 30 przypadków, założmy że połowa to szum (klasa „negatywna”) a połowa sygnał (klasa „pozytywna”)



- Dla prostoty przyjmijmy, że przypadki są opisywane przez dwie zmienne kinematyczne (x_1, x_2)
 - Jeżeli spróbujemy selekcji liniowej (użyjemy każdej zmiennej oddzielnie) wyniki będą **słabe!**
 - Odpowiednie rozkłady ID pokrywają się
- Nasz algorytm **wielomodalny** musi nauczyć się zasady separacji przypadków, tak żeby każdy nowy przypadek zaklasyfikować do jednej z dwóch klas (szum/sygnał) używając zmiennych x_1, x_2
 - Geometrycznie, wytrenowana zasada klasyfikacji to hiperpowierzchnia

Redukcja wymiarów

- ❑ Funkcja skokowa Heaviside'a jest jednym z najbardziej popularnych typów funkcji aktywacji (jest wiele innych). Jako argument podajemy funkcji skokowej kombinację liniową zmiennych oraz wag

$$\vec{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix}, \vec{x}^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_k^{(i)} \end{bmatrix}$$

- ❑ Zapisując to dla wygody jako wektory wyznaczamy z:

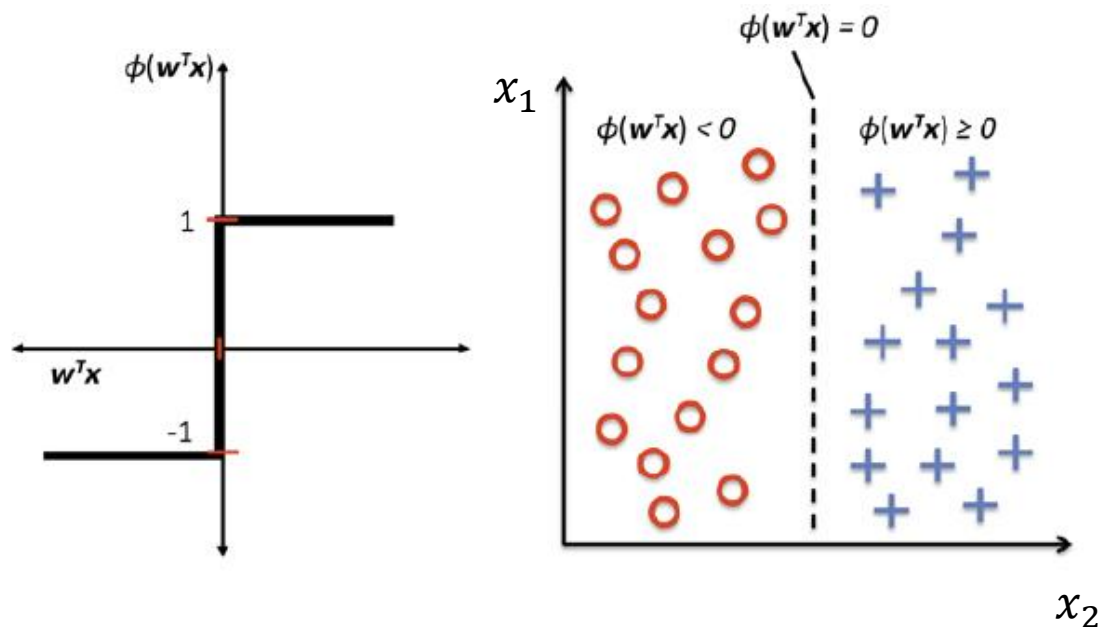
$$z^{(i)} = w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_k x_k^{(i)} = \sum_{j=1}^{j=k} w_j x_j^{(i)} = \vec{w}^T \vec{x}^{(i)}$$

- ❑ Można również przenieść na lewą stronę równania próg aktywacji aby dostać:

$$z^{(i)} = w_0 x_0^{(i)} + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_k x_k^{(i)}, w_0 = \theta, x_0^{(i)} = 1$$

And the Magic...

- ❑ Należy tutaj zwrócić uwagę na dwie rzeczy:
 - ❑ Skomplikowana struktura danych wejściowych została zamieniona na informację binarną (+1/-1)
 - ❑ W jaki sposób możemy dokonać na tej podstawie klasyfikacji?



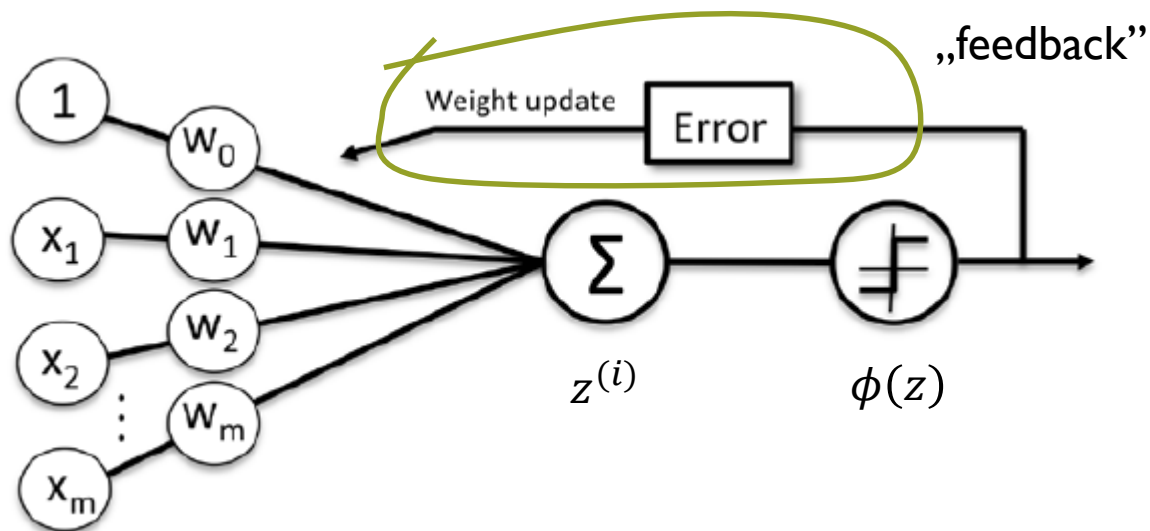
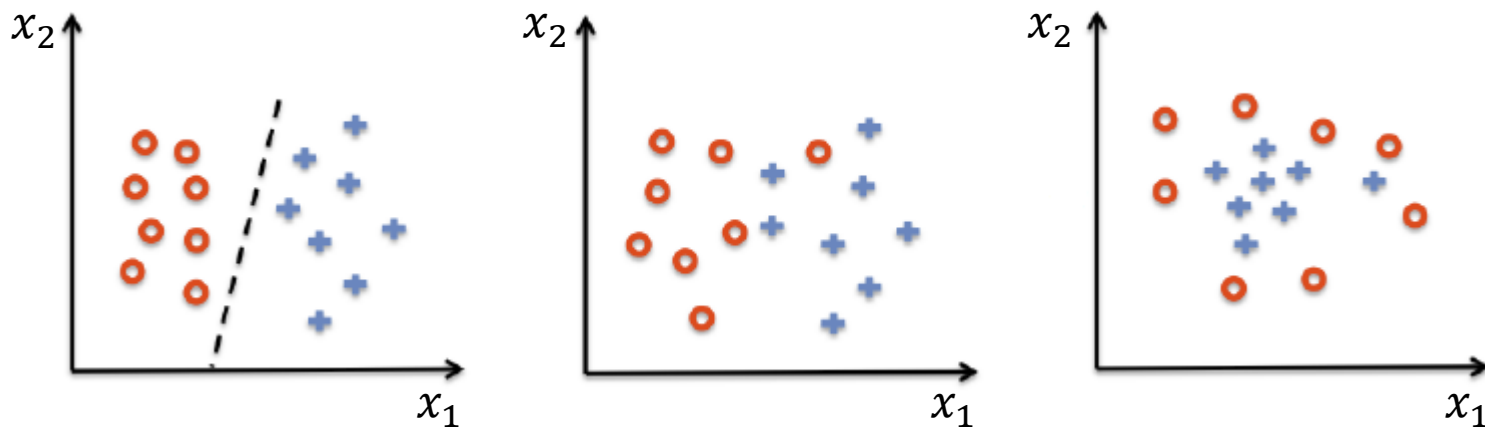
Trening

- ❑ Implementacja treningu dla perceptronu:
 - ❑ Inicjalizacja wag (zwykle „0” na początek)
 - ❑ Dla każdego przypadku $\vec{x}^{(i)}$ wykonaj:
 - ❑ Wyznacz odpowiedź neuronu (klasa) $\tilde{y}^{(i)}$, przy pomocy naszej funkcji aktywacji
 - ❑ Odśwież wartości wag w zależności od otrzymanej klasy
- ❑ Modyfikacja wag:

$$w_j = w_j + \Delta w_j$$
$$\Delta w_j = \eta \cdot (y^{(i)} - \tilde{y}^{(i)}) \cdot x_j^{(i)}$$

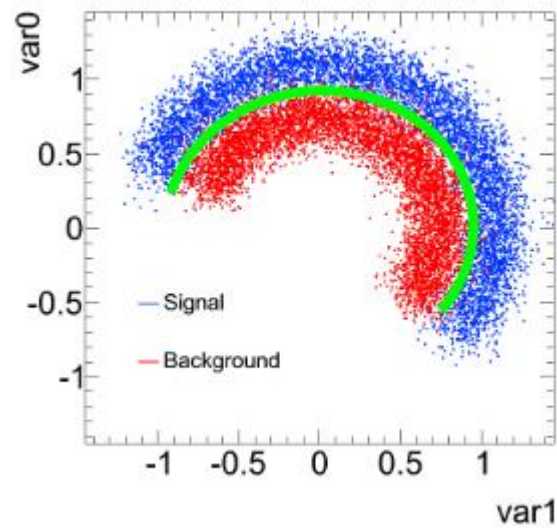
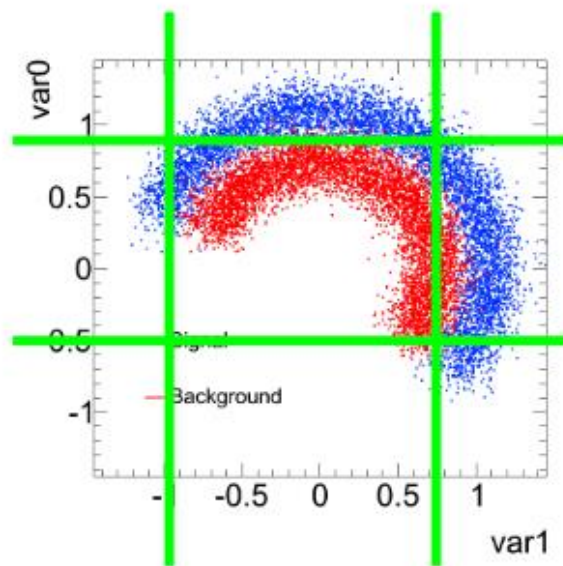
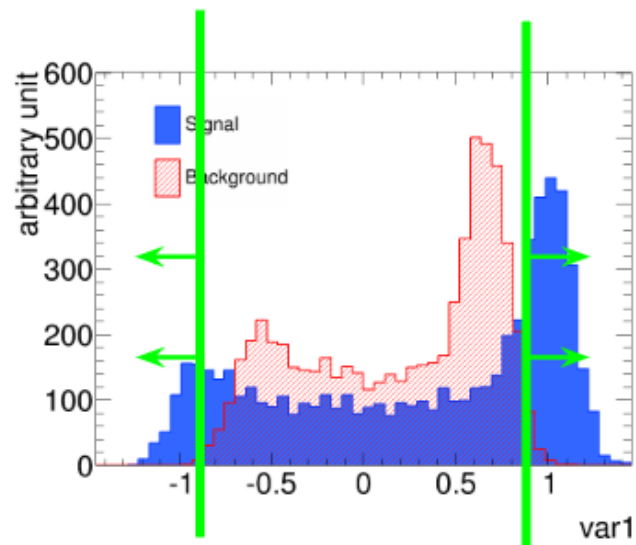
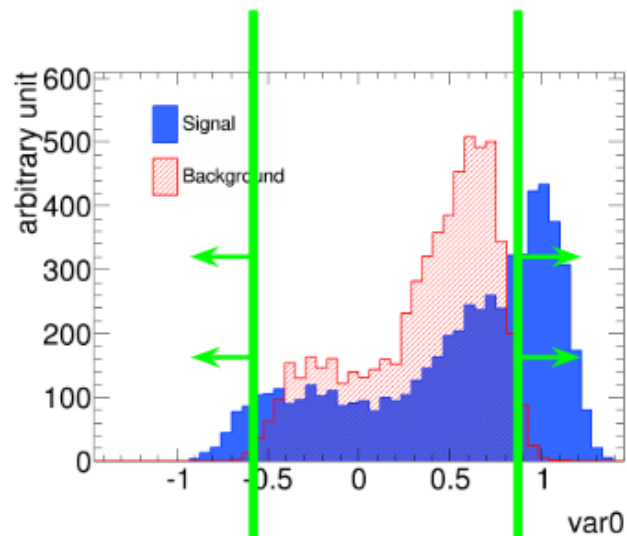
- ❑ Drugie równanie nazywamy **zasadą uczenia neronu**, parametr η gradient (szybkość) uczenia (formalnie jest to parametr z przedziału $[0,1]$)

Trening





Real Life...



Trees

- ❑ **Drzewo klasyfikujące (classification tree):**
 - ❑ Lista „pytań”, które reprezentują **węzły drzewa**
 - ❑ W każdym węźle możemy mieć tylko **dwie odpowiedzi** (zasada ta pracuje bardzo dobrze zarówno dla zmiennych **ciągłych** jak **dyskretnych** – istotne dla HEP)
 - ❑ Węzły „czują się nawzajem”, lub innymi słowy: pytania zależą od uprzednio uzyskanych odpowiedzi
 - ❑ Ostateczna odpowiedź nazywana jest liściem i oznacza klasyfikację do danej klasy przypadków
- ❑ Drzewa są szczególnie ulubione przez HEP...
 - ❑ W każdym węźle mamy pełną kontrolę nad tym co się dzieje i jesteśmy w stanie łatwo zinterpretować wyniki (inaczej niż w przypadku NN)
 - ❑ Wydaje się, że im więcej zmiennych tym lepiej dla drzewka...
 - ❑ Procedura trenowania jest bardzo szybka

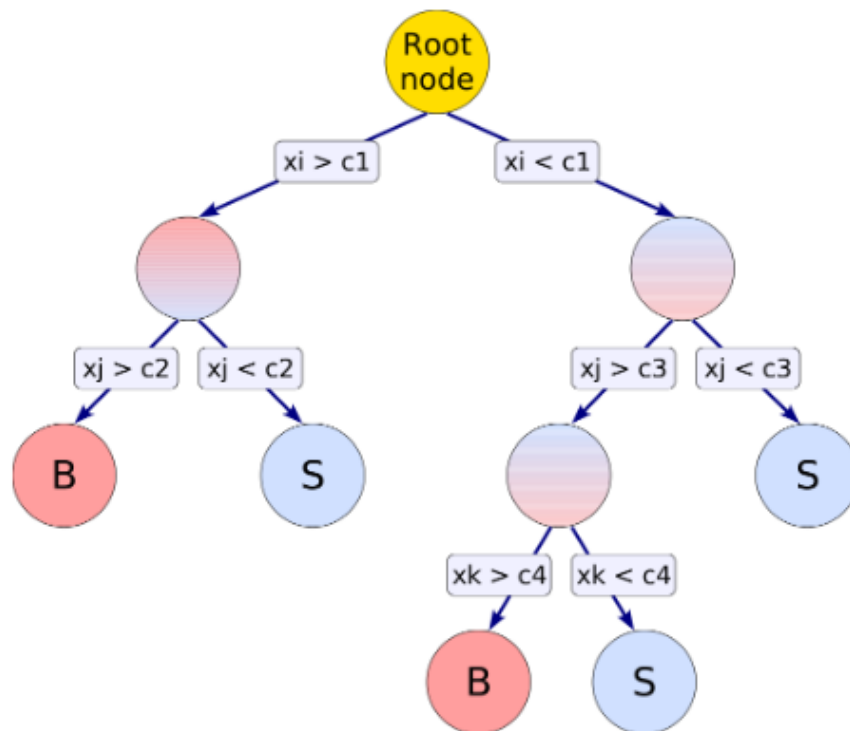
Trees

- ❑ Nothing is perfect, however...
 - ❑ Zmiana własności danych wejściowych może zupełnie popsuć jakość klasyfikacji
 - ❑ Żeby poprawić sytuację zamiast jednego drzewa wykorzystujemy **lasy**... (grupa drzew)
- ❑ Na początku, może wydawać się to trochę nie intuicyjne, jak użycie wielu podobnych drzew może poprawić klasyfikację? Magic?
- ❑ Załóżmy, że mamy wytrenowane 3 drzewa, które charakteryzują się taką samą jakością klasyfikacji (separation power), która wynosi 0.7
- ❑ Lasy głosują w sprawie decyzji, więc 2/3 drzew musi dokonać klasyfikacji poprawnej, tak więc prawd. Błędnej decyzji wynosi teraz

$$p_{miss} = \binom{3}{2} 0.3^2 0.7 + \binom{3}{3} 0.3^3 = 0.216$$

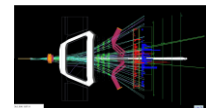
Trees

- Schemat drzewka wygląda tak...

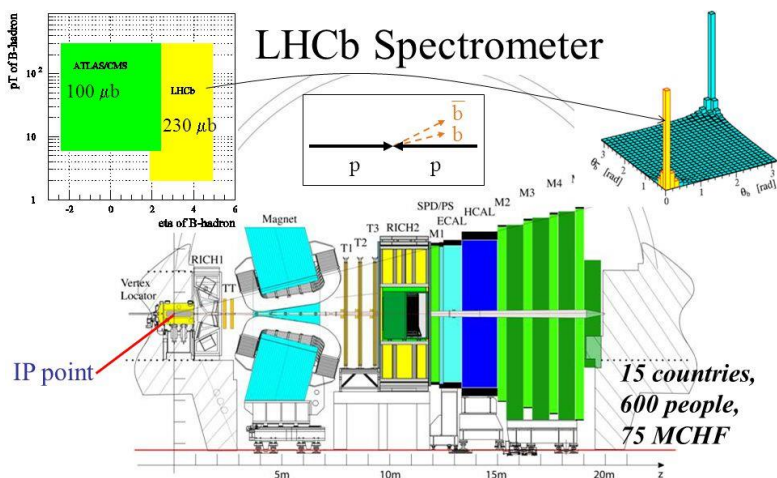


Boosting (Wzmacnianie)

- ❑ Ostatnią z rzeczy historycznych, które stanowią o sile metod uczenia maszynowego jest technika wzmacniania decyzji
- ❑ Pytanie – czy słabi uczniowie mogą połączyć siły i zostać mędrcem (Kearns i Valiant)
- ❑ Matematyczny dowód – Schapire (1990)
 - ❑ Można pokazać, że klasyfikatory, które dają odpowiedzi tylko nieco lepsze niż klasyfikacja losowa mogą „współpracować” ze sobą tak, żeby dać klasyfikator o **arbitralnie wysokiej jakości klasyfikacji**
 - ❑ **Boosting**
- ❑ Trenujemy słabych uczniów a następnie łączymy ich ze sobą
- ❑ Większość obecnie prowadzonych analiz fizycznych w LHCb (i pewnie nie tylko...) bazuje na technice BDT (Boosted Decision Trees)



Banda LHCb-AGH

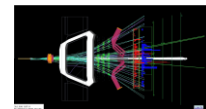


Good mass and proptime resolution: VELO + tracking system
Hadron identification: RICH system

L0 Lepton and Hadron p_T trigger: Calorimeter and muon system

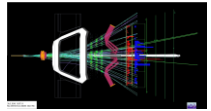
- Agnieszka Obłąkowska
- Bartek Rachwał
- Adam Dendek
- Maciej Majewski
- Wojciech Krupa
- Paweł Kopciewicz
- Artur Kucia

- Klasyfikacji śladów długożyciowych w czasie rzeczywistym (tryger)
- Analizy rzadkich rozpadów hadronowych: $B_S^0 \rightarrow D_S^+ K^{*-}$ oraz $B_S^0 \rightarrow D_S^{*+} K^{*-}$
- Autonomiczny system do oceny jakości danych detektora wierzchołka
- Modelowanie funkcji opisującej kształt odpowiedzi układów odczytu front-end

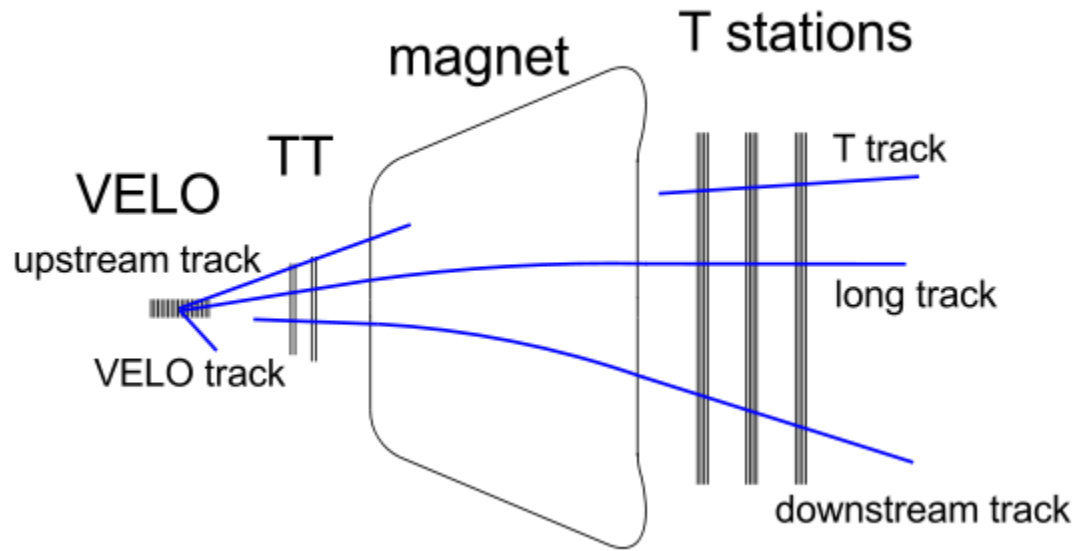


Tryger LHCb (A.D., A.K., T.S.)

- ❑ Detektor LHCb dokonuje pełnego odczytu wszystkich swoich kanałów z częstotliwością 1.1 MHz
- ❑ Farma, która przetwarza dane w czasie rzeczywistym ma około **250 ms** na podjęcie decyzji czy dany przypadek jest interesujący czy nie (błędnej decyzji nie da się naprawić!)
- ❑ W tym czasie musimy między innymi zrekonstruować wszystkie ślady jakie rejestruje detektor
- ❑ Typowy przypadek zawiera ~ 350 cząstek (rozkład Poisson'a)
- ❑ Duża gęstość „hitów” stanowi trudny problem dla rekonstrukcji
- ❑ Metody klasyczne (jakość dopasowania modelu śladu) nie wystarczają

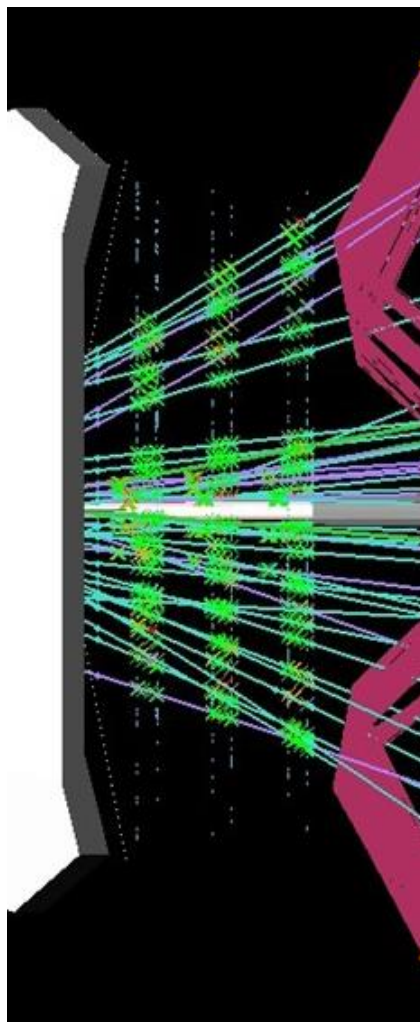
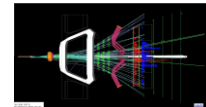


Tryger LHCb

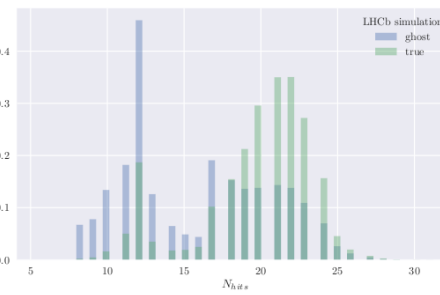
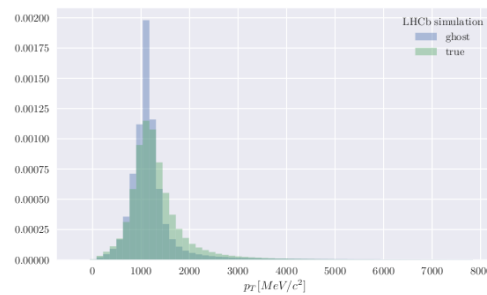
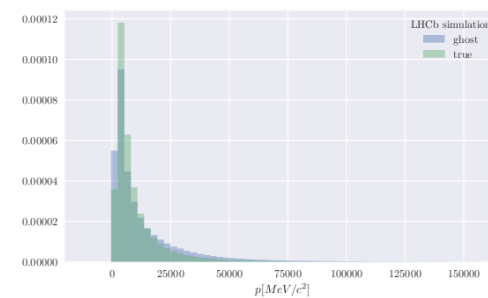
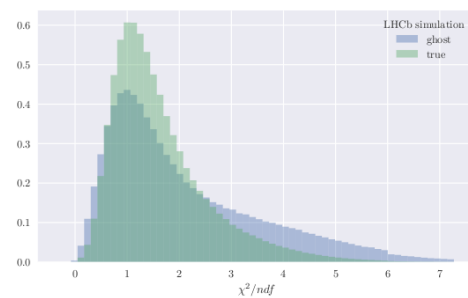


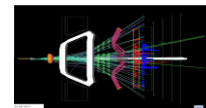
$$\vec{x} = \begin{pmatrix} x \\ y \\ t_x \\ t_y \\ q/p \end{pmatrix} \text{ with } t_x = \frac{\partial x}{\partial z} \text{ and } t_y = \frac{\partial y}{\partial z}$$

Tryger LHCb



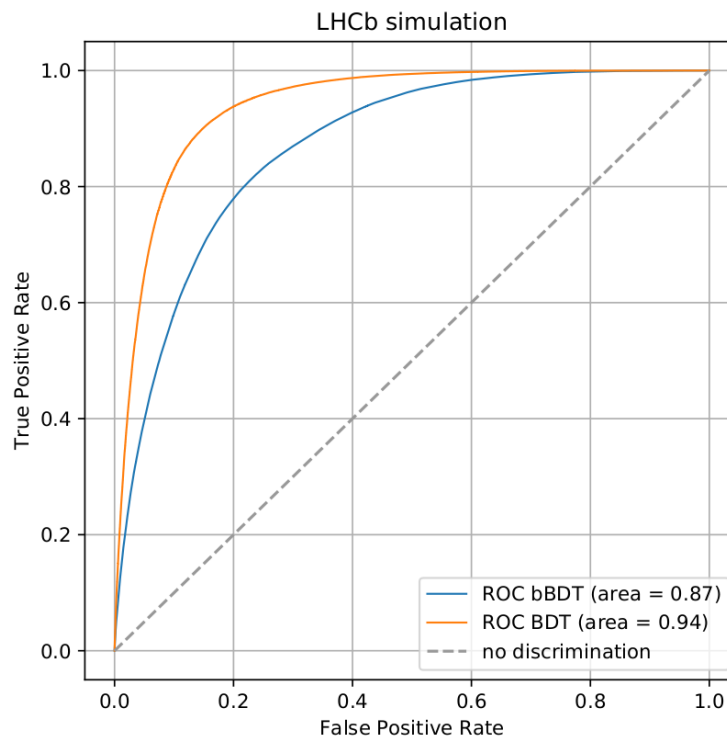
- ❑ Trudny problem kombinatoryczny
- ❑ Duże prawd. rekonstrukcji śladów fałszywych „ghost”

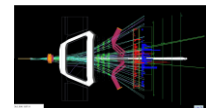




Tryger LHCb

- ❑ Adam i Artur zaatakowali problem używając BDT oraz sieci neuronowej
- ❑ Ostateczny wybór padł na bBDT, który od początku 2016 roku działa w systemie wyzwalania przypadku LHCb



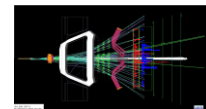


Tryger LHCb

- ❑ Adam i Artur zaatakowali problem używając BDT oraz sieci neuronowej
- ❑ Ostateczny wybór padł na bBDT, który od początku 2016 roku działa w systemie wyzwalań przypadku LHCb

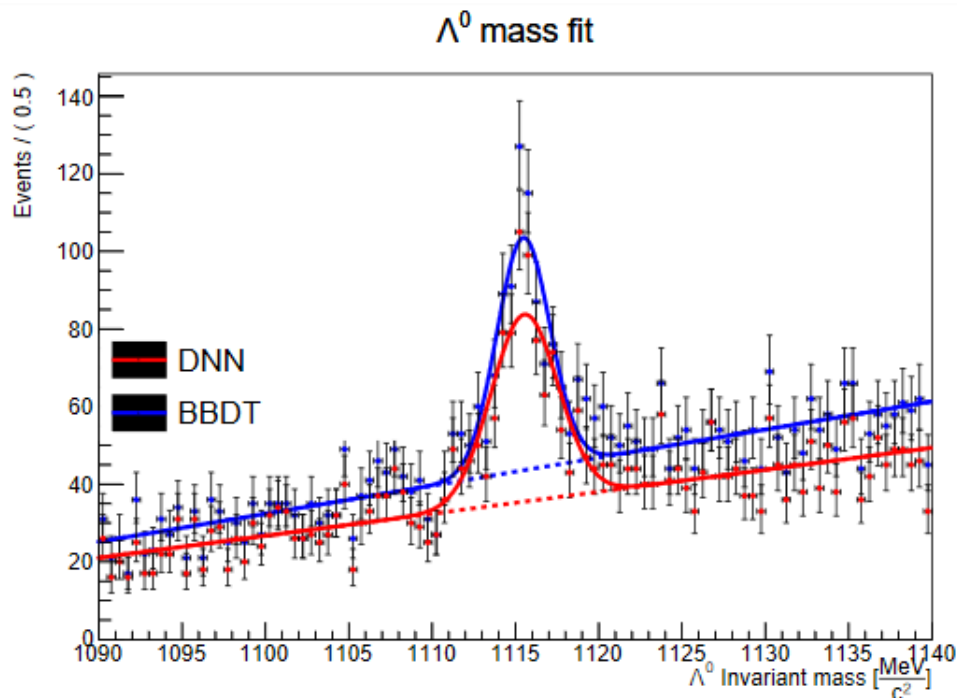
parameter	new model	BBDT	parameter	new model	BBDT
ϵ_{rec}	74.1%	75.0%	ϵ_{rec}	73.1%	74.1%
$\epsilon_{rec,p>5GeV}$	80.5%	81.4%	$\epsilon_{rec,p>5GeV}$	79.9%	80.8%
<i>ghost rate</i>	23.2%	30.3%	<i>ghost rate</i>	24.2%	31.3%

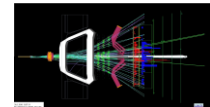
parameter	new model [ms]	BBDT [ms]
CPU time	1.822	1.681



Tryger LHCb

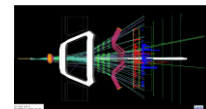
- ❑ Adam i Artur zaatakowali problem używając BDT oraz sieci neuronowej
- ❑ Ostateczny wybór padł na bBDT, który od początku 2016 roku działa w systemie wyzwalania przypadku LHCb



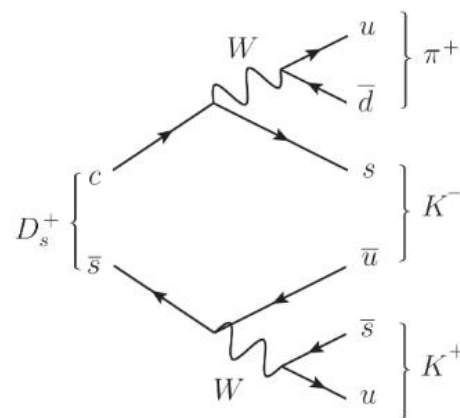
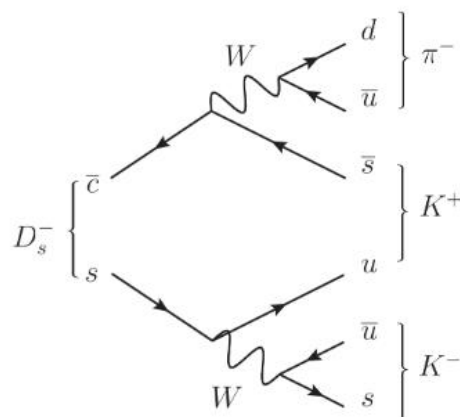
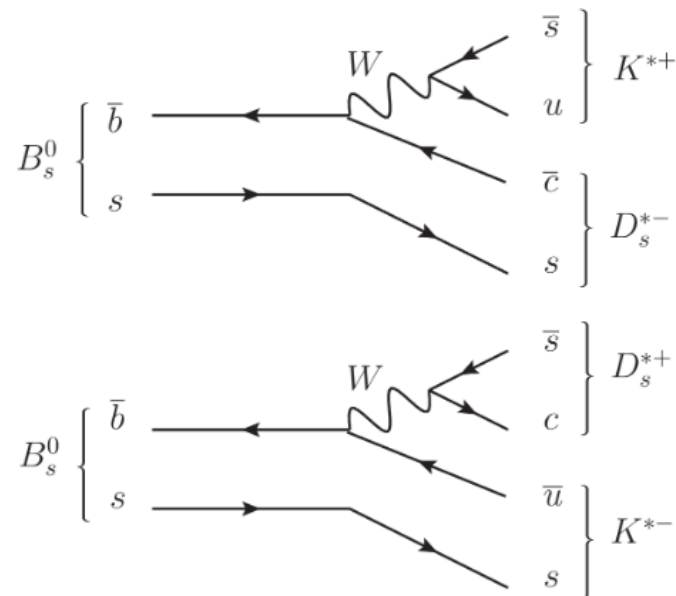
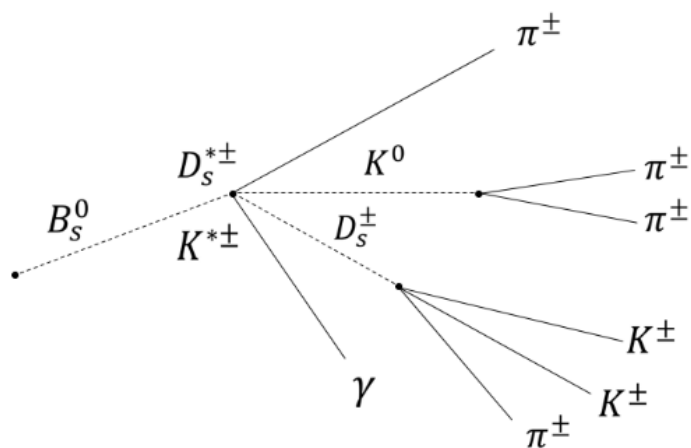


New Way

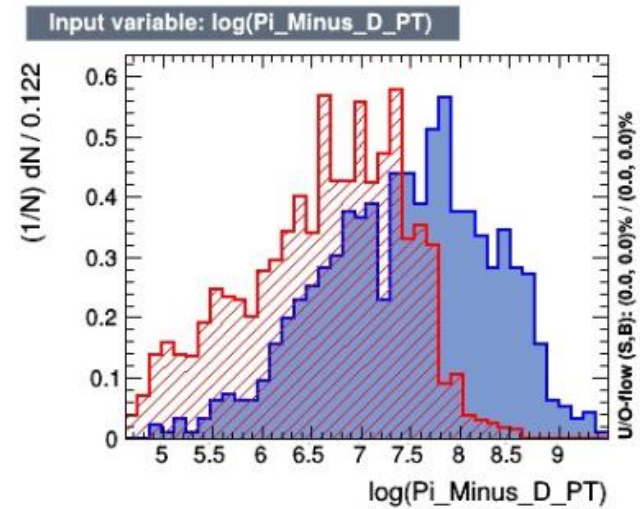
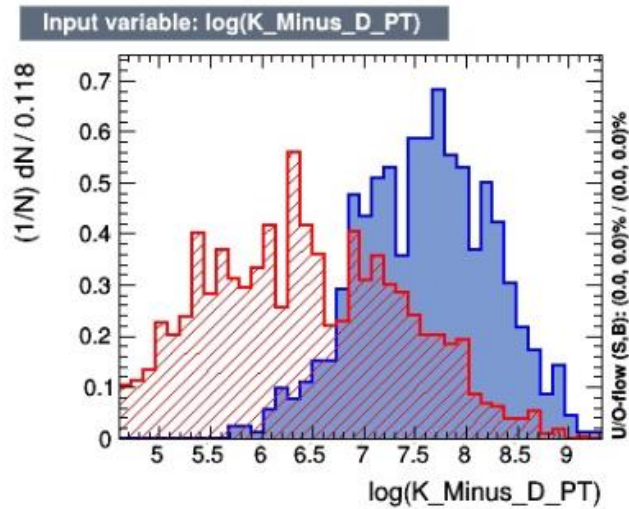
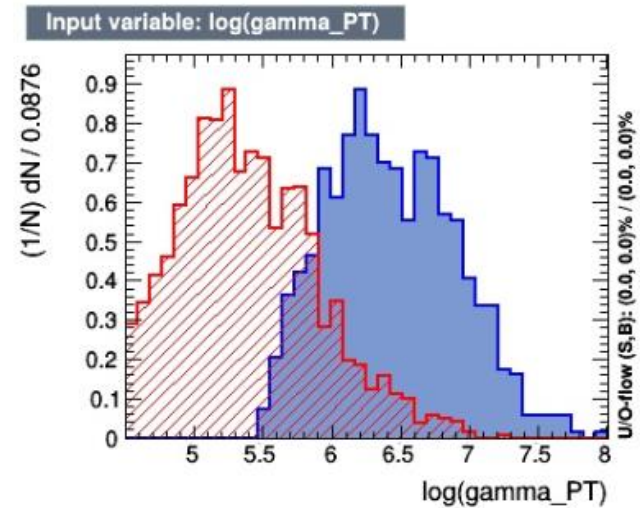
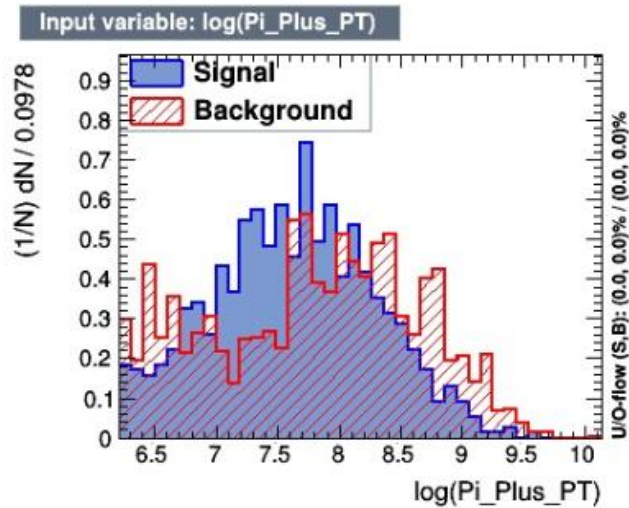
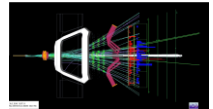
- ❑ Z uwagi na to, że modernizacja detektora LHCb jest w pełni rozkwitu – połączyliśmy siły z grupą **prof. Kulczyckiego (Szymon Łukasik, Grzegorz Gołaszewski)** aby stworzyć jeszcze bardziej wyrafinowaną aplikację do klasyfikacji śladów długożyciowych
- ❑ Podejmiemy próbę stworzenia systemu, który będzie się uczyć bez nadzoru – korzystając z danych rzeczywistych
- ❑ Zupełnie nowe podejście w LHCb!
- ❑ Problemem może być szybkość – zmodernizowany tryger będzie posiadać budżet czasowy wynoszący około **13 ms** na przetworzenie przypadku i wypracowanie decyzji...
- ❑ Jednocześnie pracujemy nad bBDT oraz NN



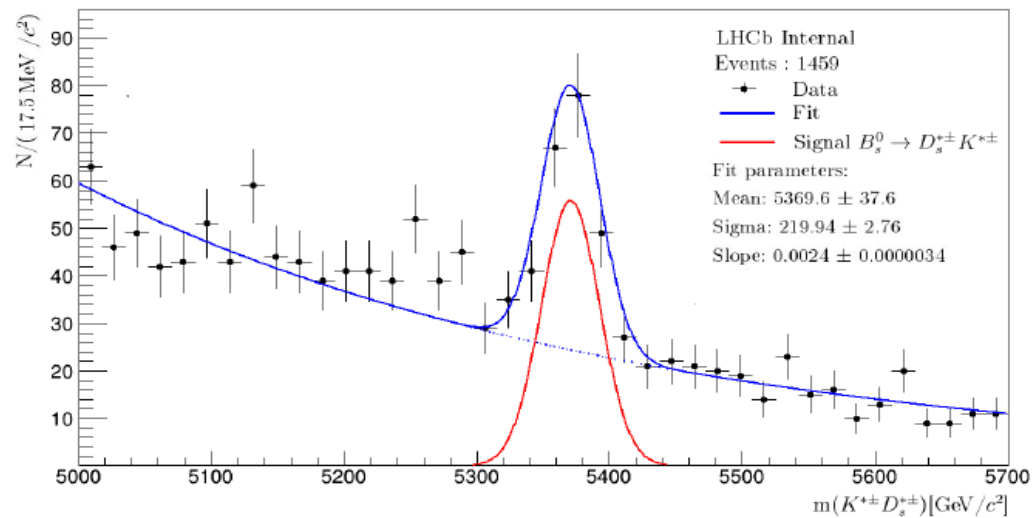
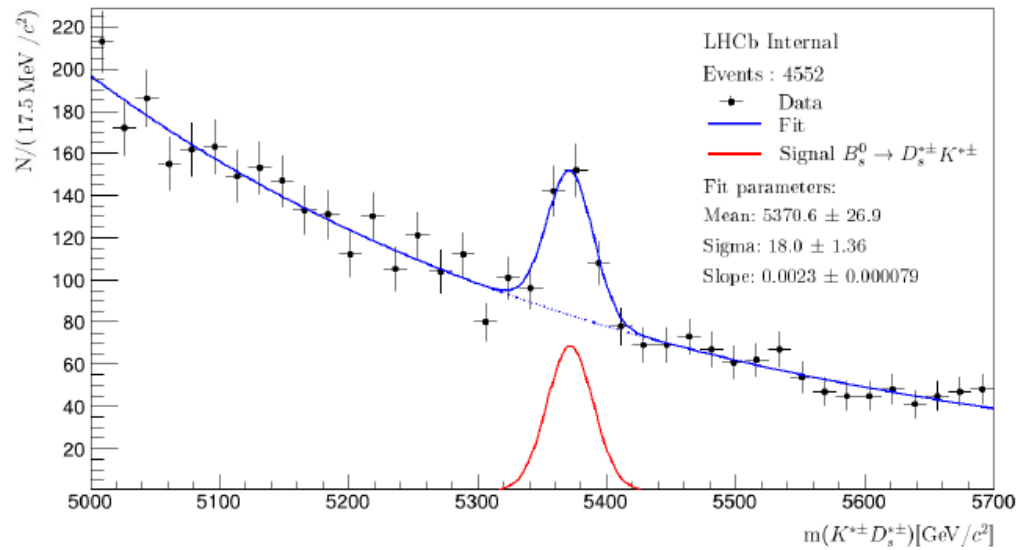
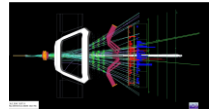
Otwarty Powab (A.O.M., W.K., T.S.)



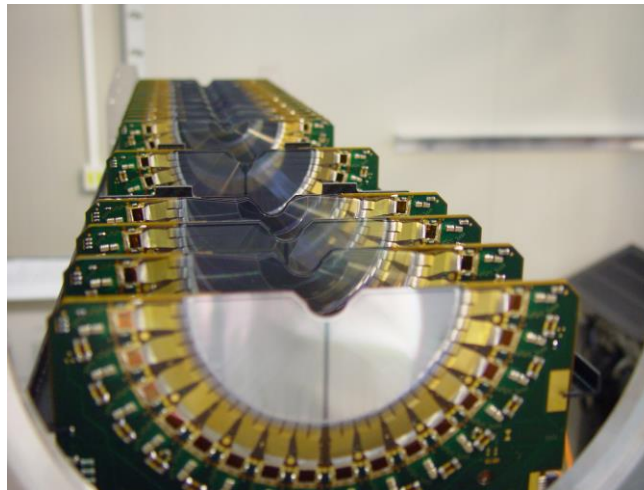
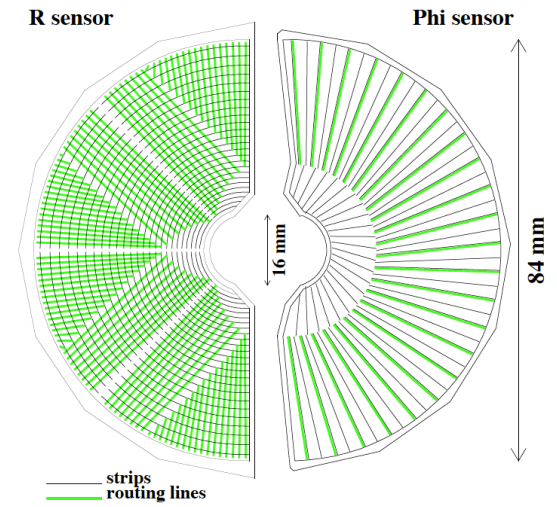
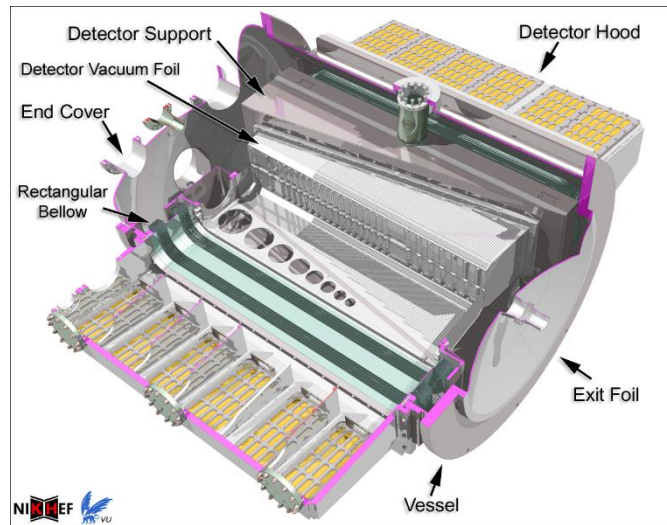
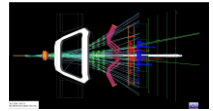
Otwarty Powab

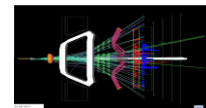


Otwarty Powab



VELO (M.M., T.S.)

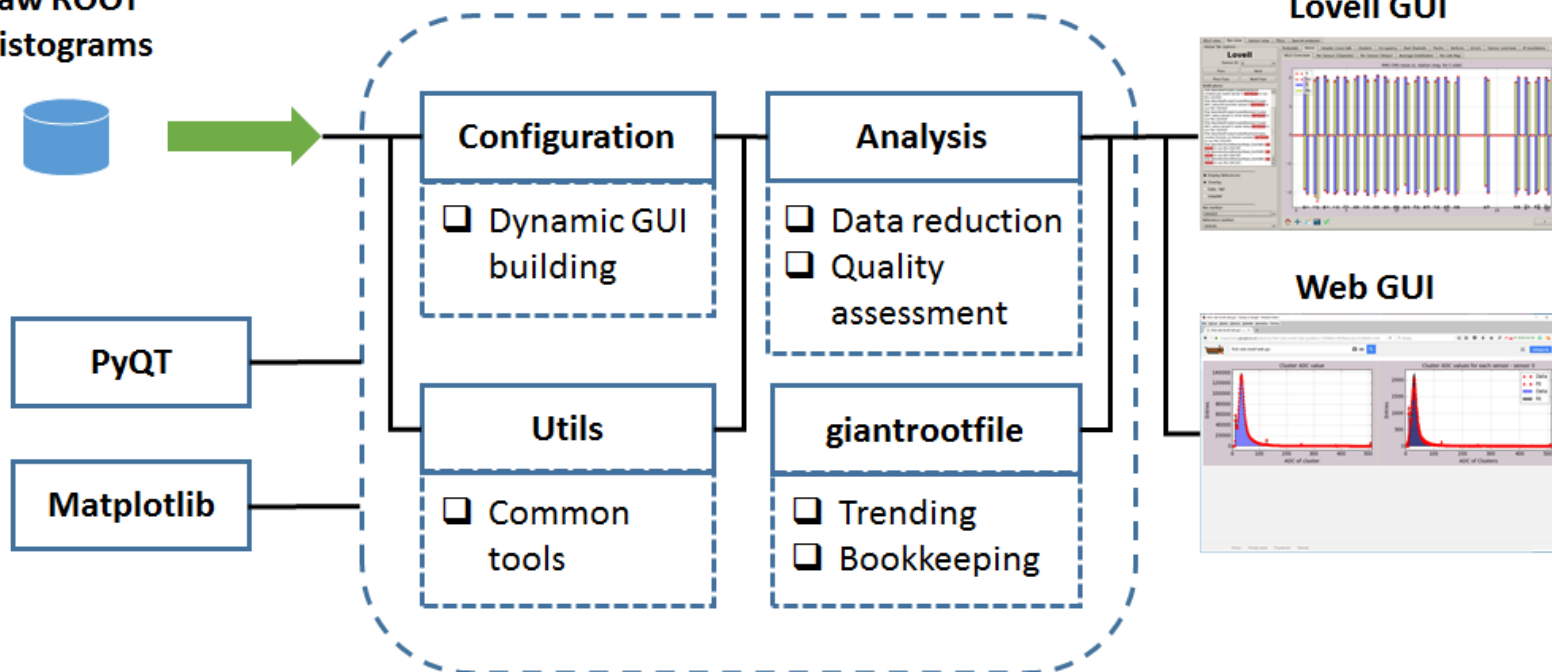


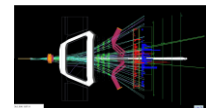


VELO (M.M.,T.S.)

- ❑ Rozbudowana platforma do analizy danych produkowanych przez detektor VELO
- ❑ 180 000 kanałów do monitorowania

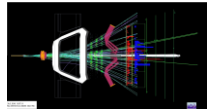
Raw ROOT
Histograms



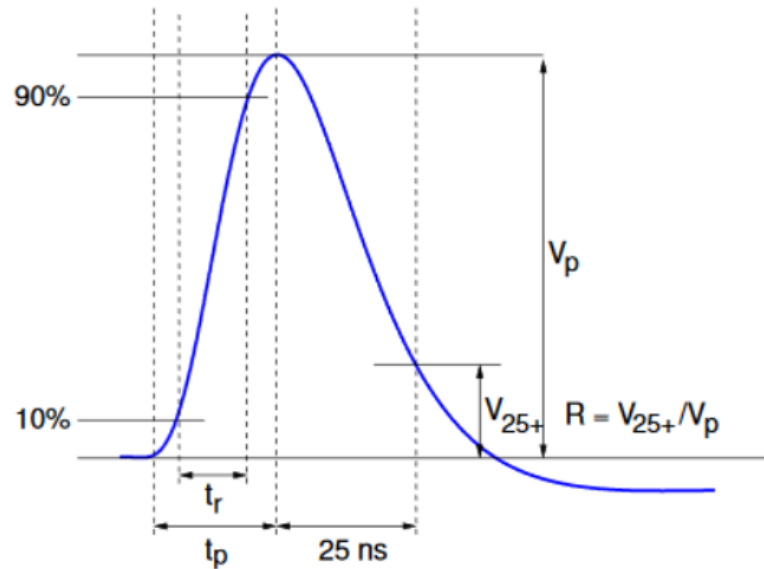
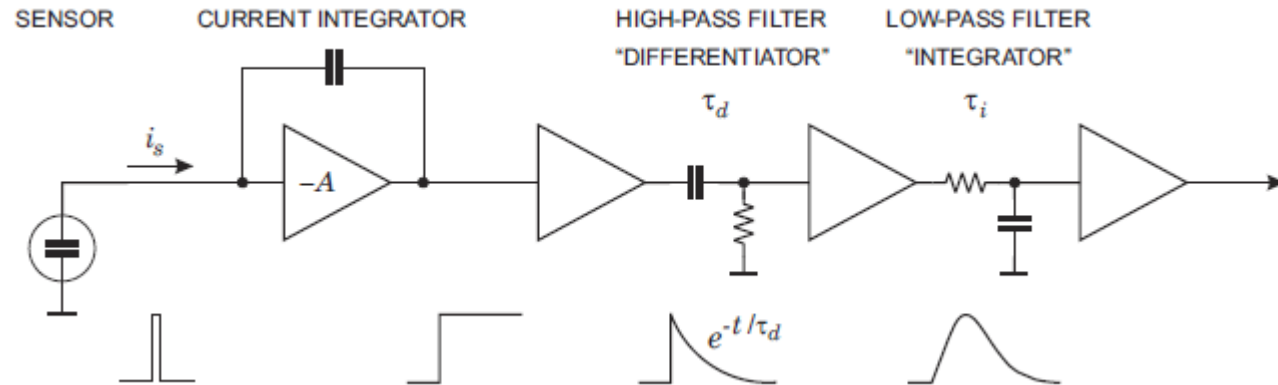


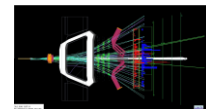
VELO (M.M., T.S.)

- ❑ Korzystając z metod uczenia maszynowego – system autonomiczny, który jest w stanie sam podjąć decyzję co do jakości danych – wyeliminowanie operatora
- ❑ Grant preludeum
- ❑ Intensywne prace nad redukcją danych
- ❑ Celem jest stworzenie systemu dla zmodernizowanego detektora (64 miliony kanałów pomiarowych, ekstremalne zniszczenia radiacyjne)
- ❑ Używamy istniejącego systemu jako test-bed, **próba** uruchomienia modułu z inteligencją obliczeniową **po wakacjach** (trzy miesiące zbierania danych)



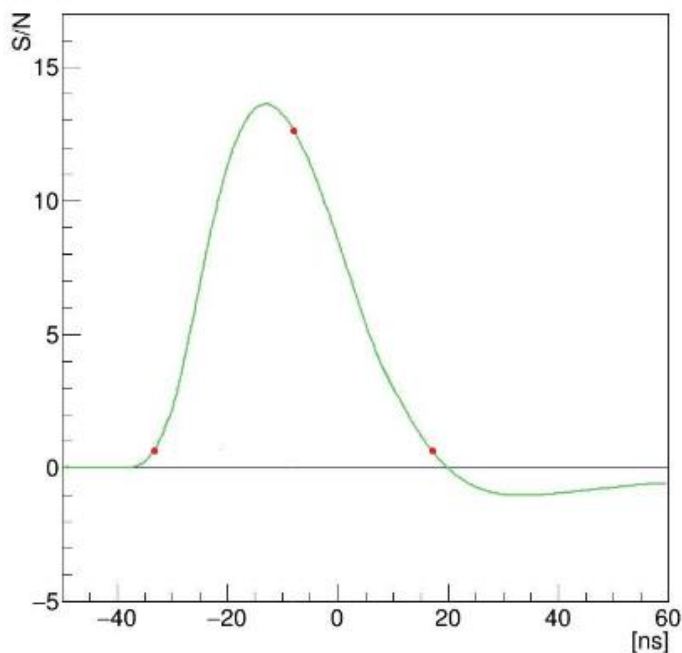
Pulse shape (P.K.)

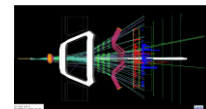




Pulse shape (P.K.)

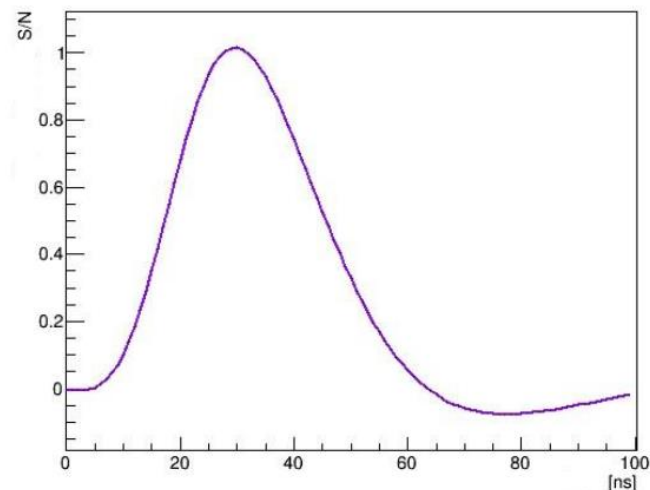
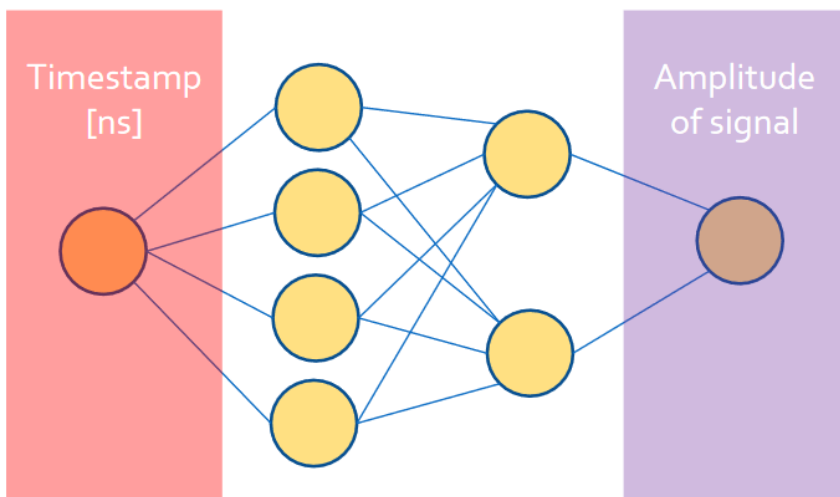
- ❑ Dokładny opis odpowiedzi układu front-end jest potrzebny do symulacji odpowiedzi detektora
- ❑ Zwykle stosuje się albo binowanie albo dopasowanie przy użyciu wielomianów wyższego stopnia w przedziałach





Pulse shape (P.K.)

- ❑ Dokładny opis odpowiedzi układu front-end jest potrzebny do symulacji odpowiedzi detektora
- ❑ Zwykle stosuje się albo binowanie albo dopasowanie przy użyciu wielomianów wyższego stopnia w przedziałach



Spin-off(s)

- ❑ Grupa entuzjastycznie nastawionych magistrantów chciałaby pracować nad autonomicznym pojazdem
- ❑ Dwie prace (rozpoznawanie znaków oraz orientacja na drodze – Artur Róg, Jakub Kolybacz) została zauważona przez NVIDIA Polska
- ❑ Dołączyli: Mateusz Karpik, Kamila Kalecińska
- ❑ Możliwa współpraca – ale negocjacje „idą powoli”
- ❑ Nowy przedmiot dotyczący głębokiego uczenia wg planu NVIDIA (uczestnicy mogą mieć lepsze szanse w aplikowaniu o staż w NV-Polska)