Data Community

SQL Day

PLATINUM SPONSOR

TECHNOLOGY INNOVATION DATA KNOWLEDGE tidk

STRATEGIC PARTNER

Microsoft

GOLD SPONSORS

CLOUDS ON MARS

Future Processing

SPYROSOFT

SILVER SPONSOR

dbWatch DATABASE CONTROL

elitmind think bright

VEEAM

BRONZE SPONSOR

Największa księgarnia IT w Polsce

NOVATECH

www.novatech.com.pl

Python i uczenie maszynowe na dużą skalę w Apache Spark

Tomasz Cieplak

Data Community

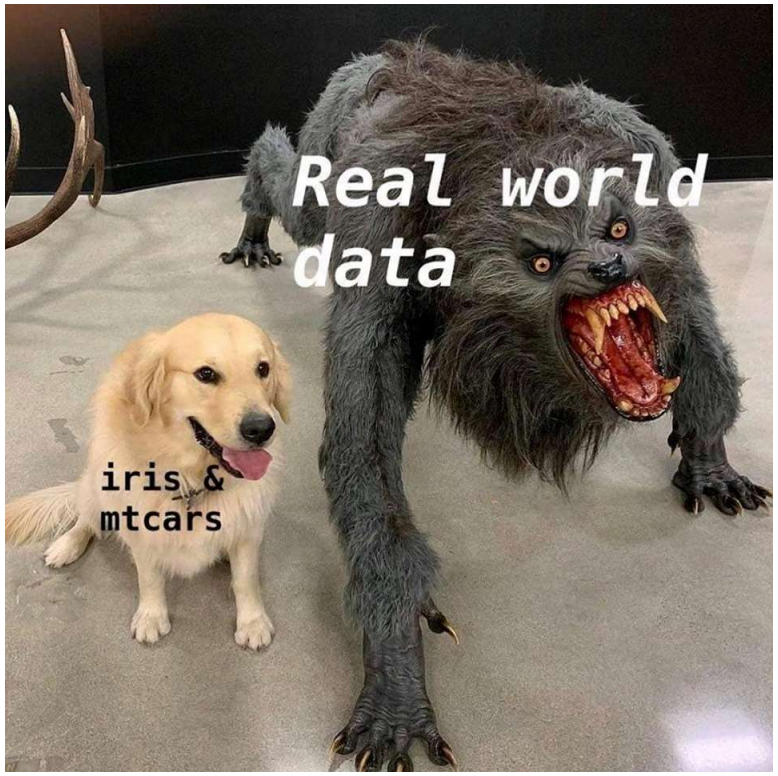
SQL Day

Agenda

- Podstawowe pytanie: Ale po co? (Scikit-Learn, TensorFlow, Keras)
- Jeśli znamy poprzednią odpowiedź, to kolejne pytanie: Jak?
  - Proste modele uczenia maszynowego (Scikit-Learn)
  - Uczenie głębokie (TensorFlow, Keras, DeepLearning Pipeline)
- Co dalej? (MLFlow)

- … Generalnie Apache Spark … PySpark …

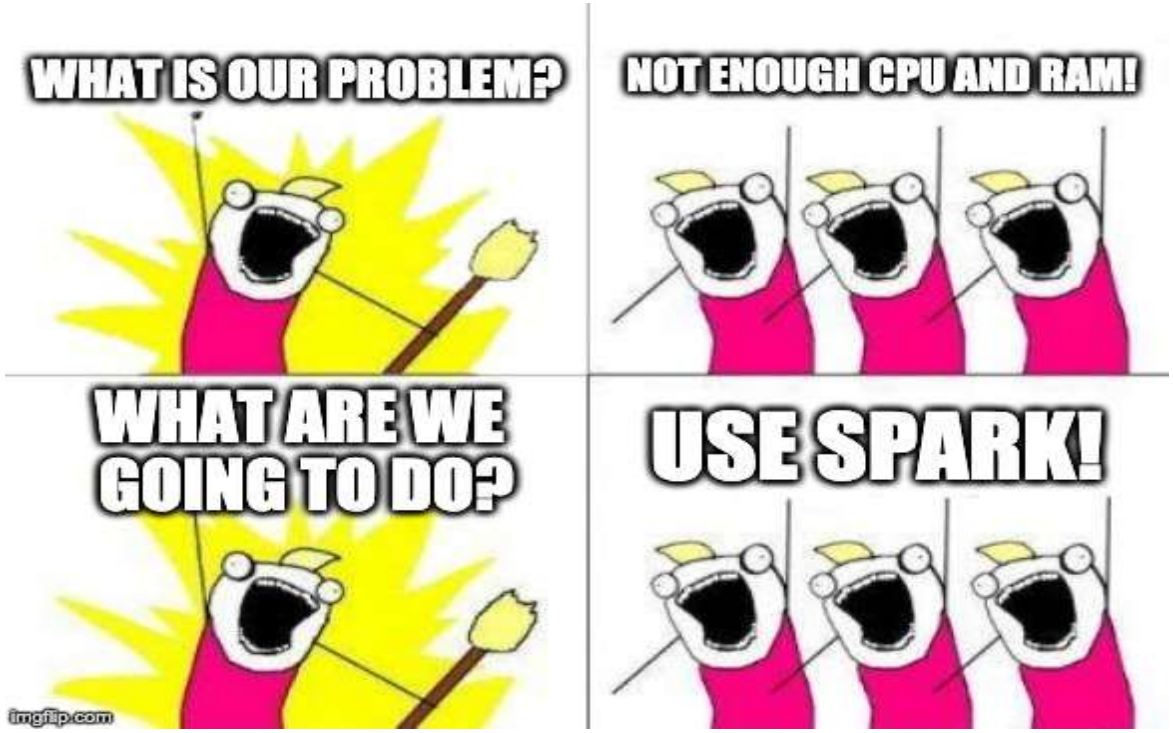
Witamy w prawdziwym świecie



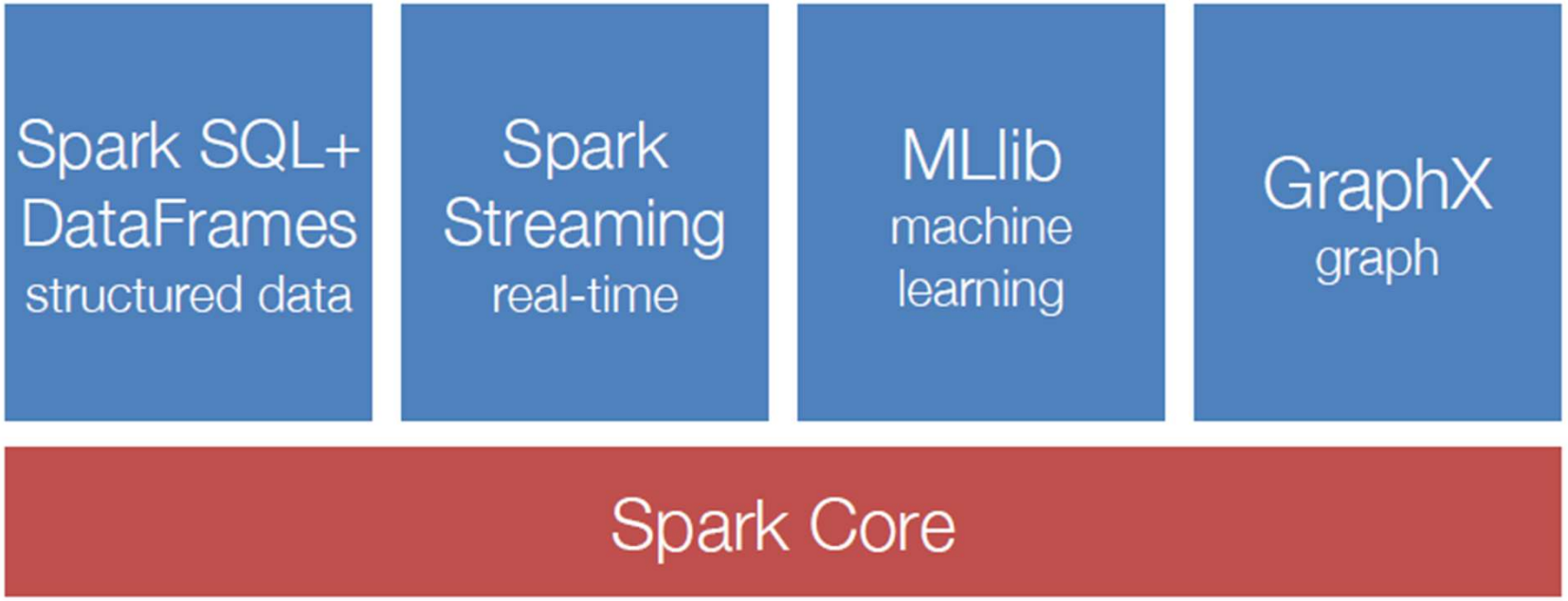
https://twitter.com/AdamGruer/status/1122271095225118720?s=20

Poniedziałek rano…

- Mamy za dużo danych… (?)
- Nasze biblioteki języka Python (Pandas, Numpy, Scikit Learn) nie są skalowalne i wykorzystują tylko jeden CPU (serio?)



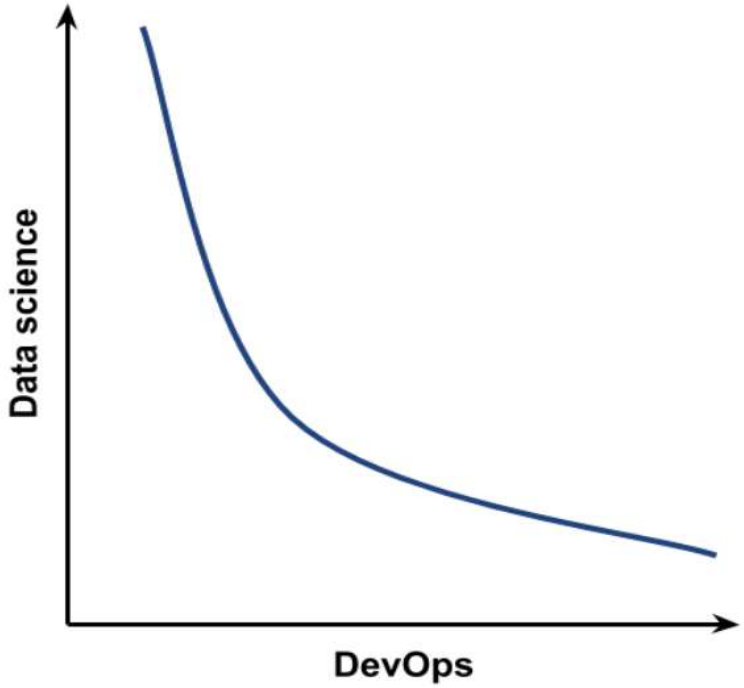
Ekosystem Apache Spark

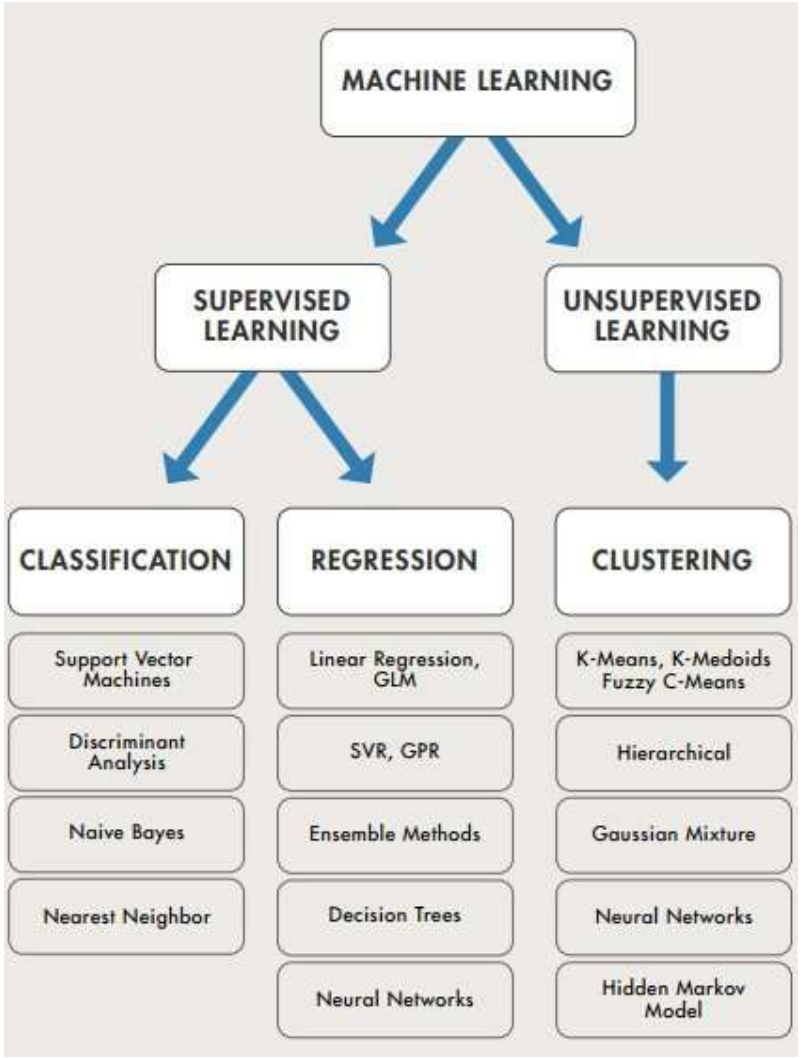


Na pewno? Dlaczego? Masz dowód?

STOP

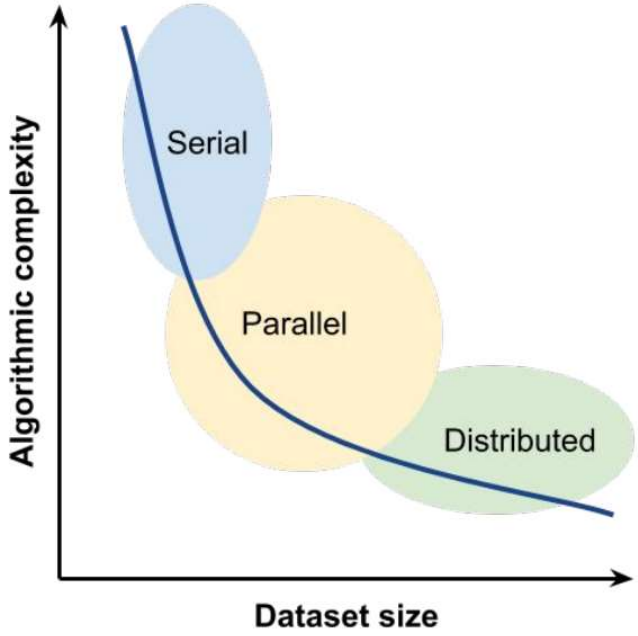
Poszukiwanie kompromisu

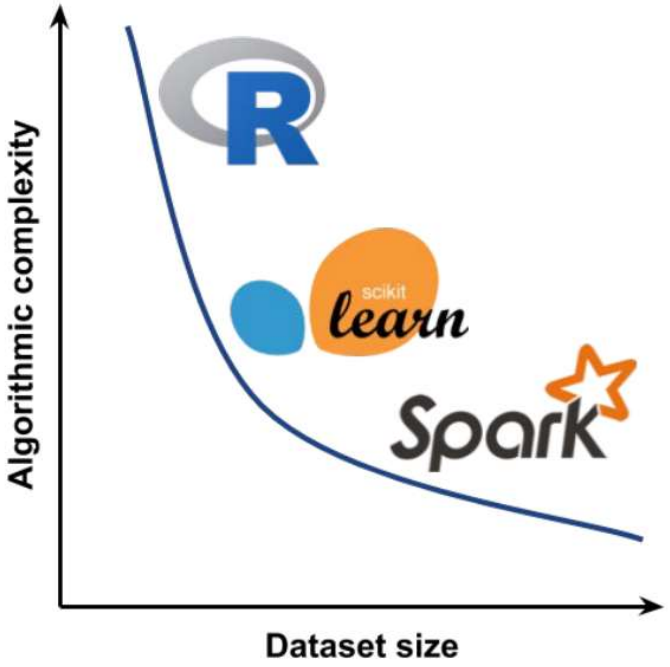




Lepiej jak jest więcej…

"More data beats better algorithms"





Skala jest kluczowym czynnikiem efektywnej nauki danych i sztucznej inteligencji

https://www.figure-eight.com/more-data-beats-better-algorithms/

To sprawdzamy – przypadek Scikit Learn

- Scikit-learn używa biblioteki `joblib` do zrównoleglania zadań na jednej maszynie
- Pozwala to trenować większość estymatorów (ale tylko, które akceptują parametr `n_jobs`) przy użyciu wielu rdzeni CPU
- `n_jobs = -1`

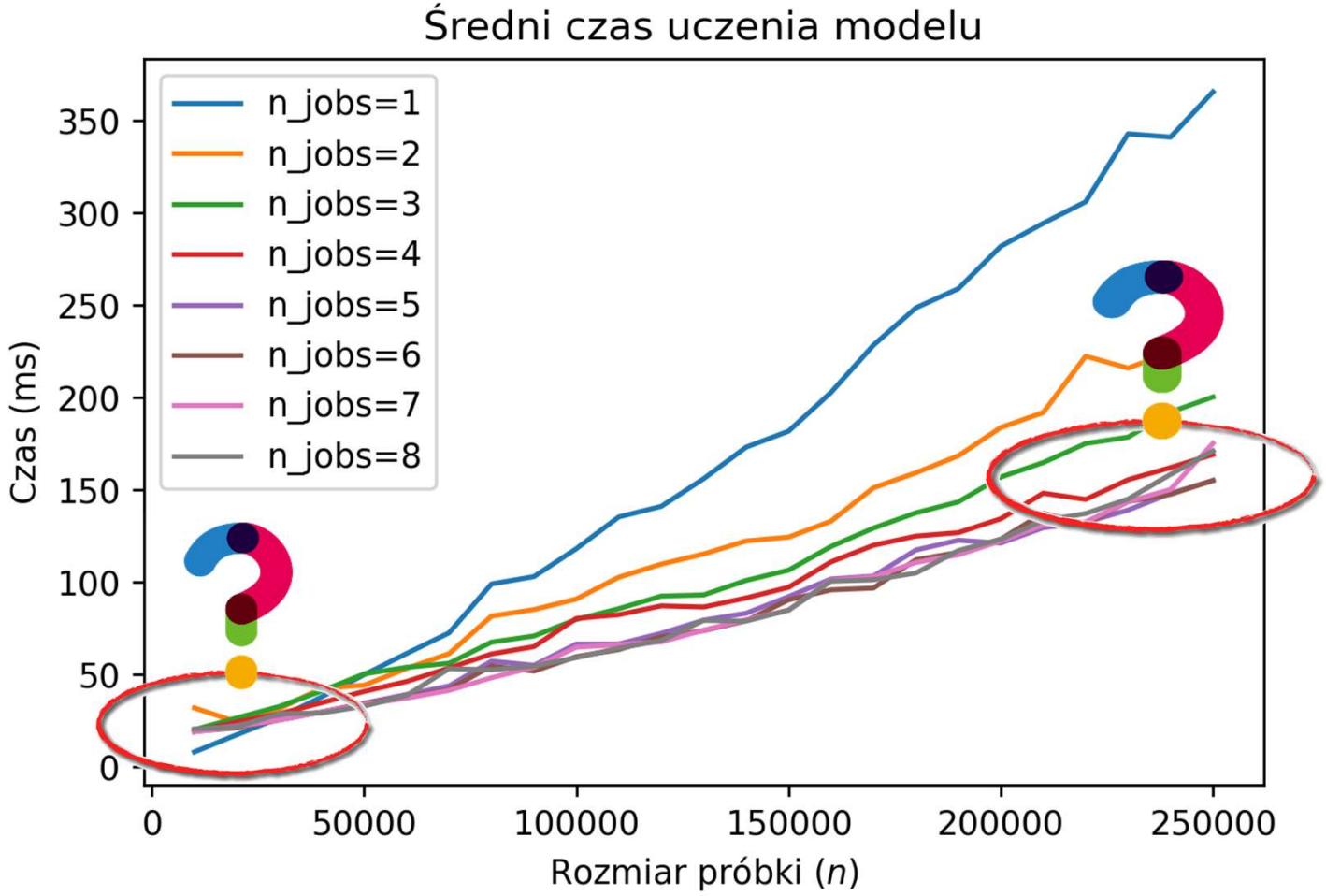
  lub

- ```
  import psutil
  cores = psutil.cpu_count()
  n_jobs = cores
  ```
  ```

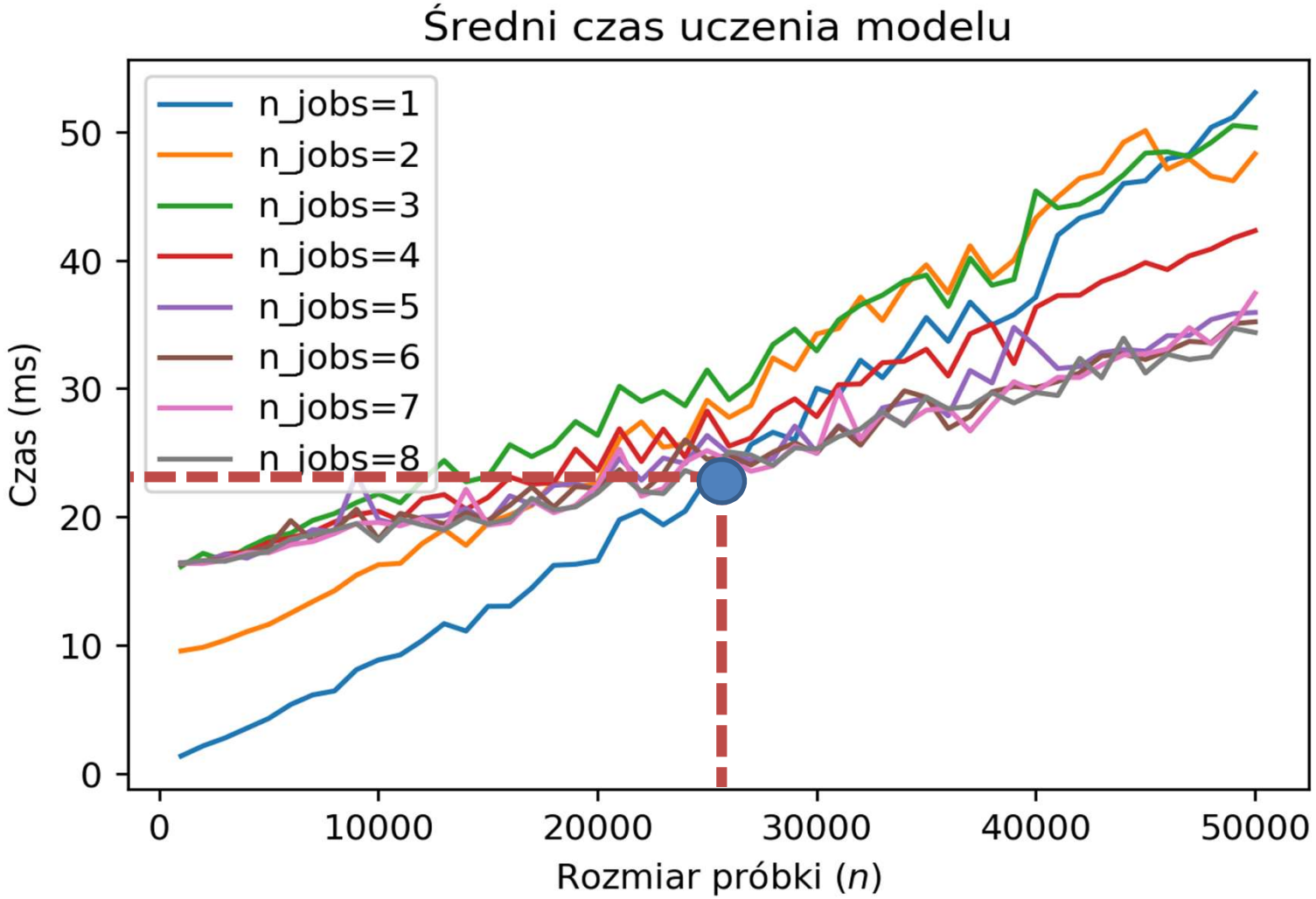
„ale tylko, które akceptują parametr n_jobs”

ANALIZA – PRZYPADEK SCIKIT LEARN

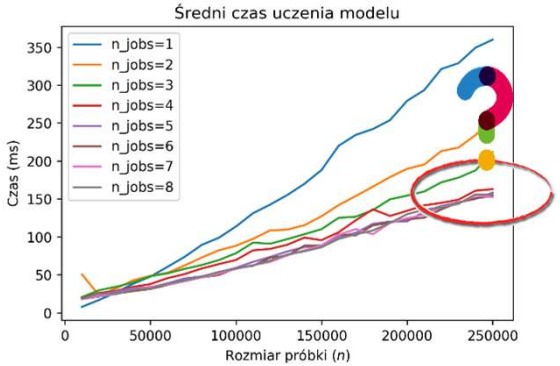
Generalnie jest dobrze, ale…

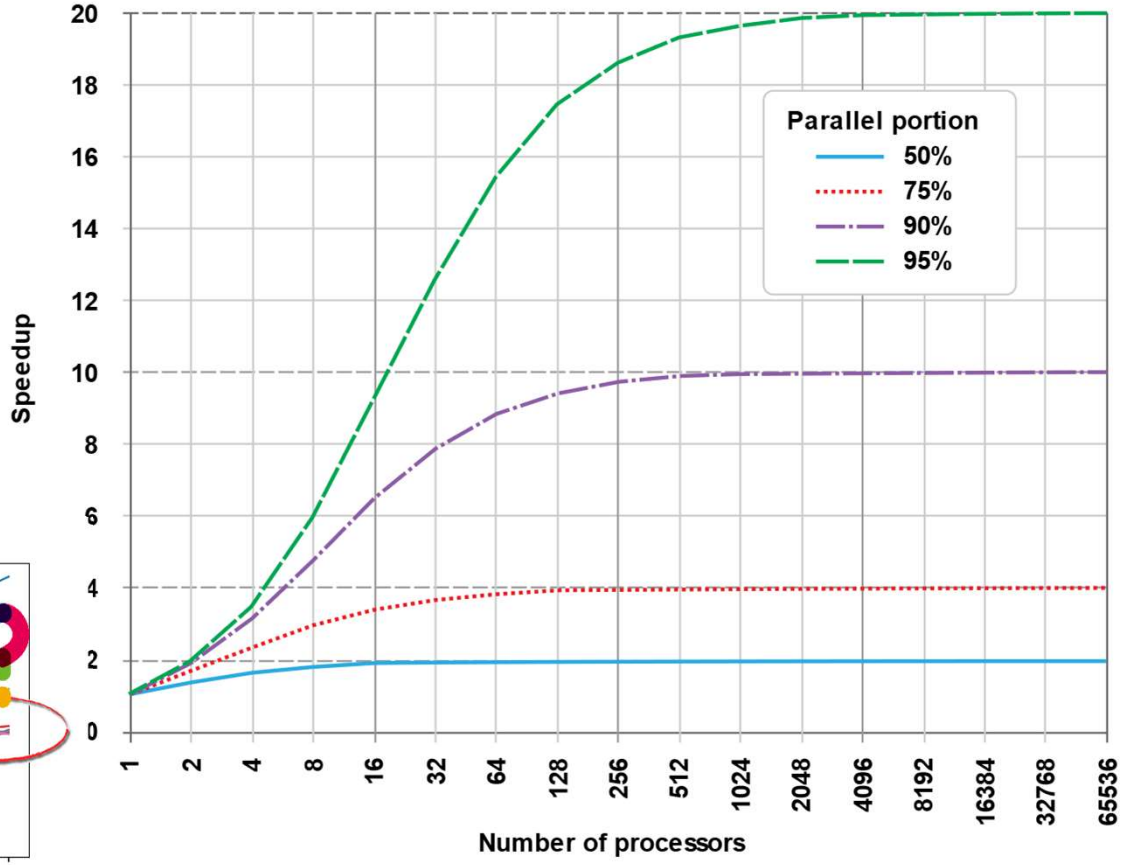


Wielkość analizowanego zbioru danych



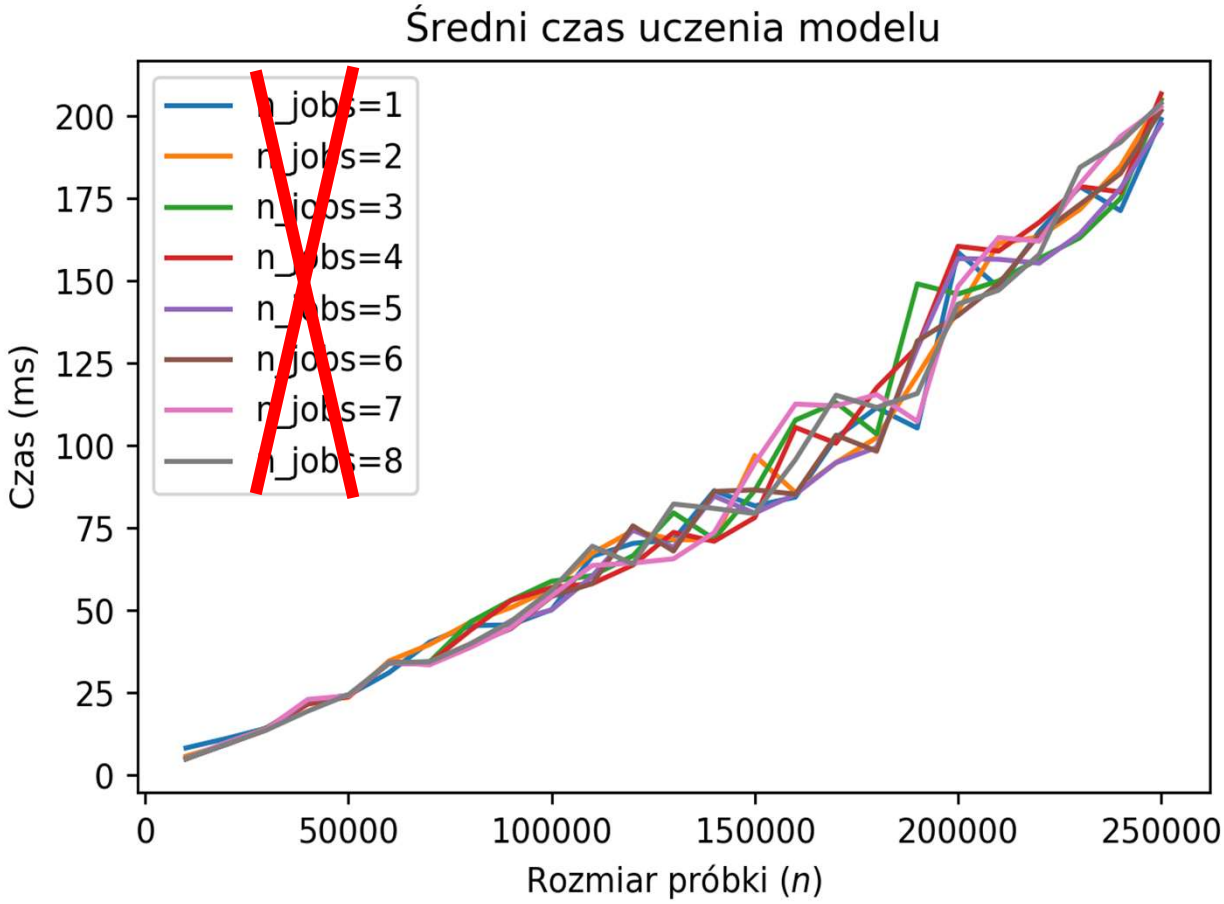
Ograniczenia – prawo Amdahlsa



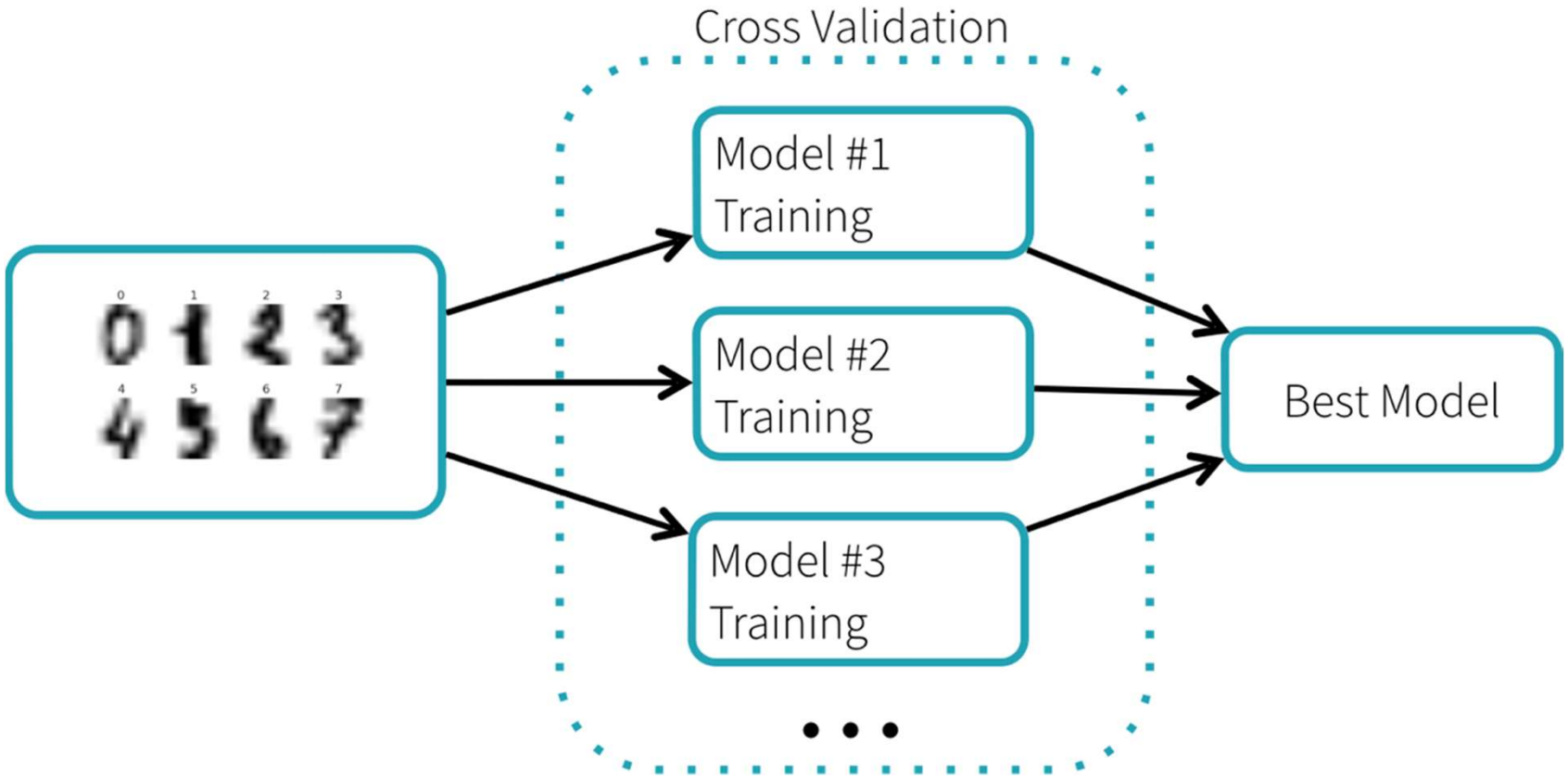


Scikit Learn – a te które nie akceptują parametru n_jobs?

- SVC
- AdaBoostClassifier
- …



Zastosowanie przeszukiwania siatki i walidacji krzyżowej



https://databricks.com/blog/2016/02/08/auto-scaling-scikit-learn-with-apache-spark.html

Więc kiedy Spark?

- Mamy dużo danych, które nie mieszczą się w pamięci RAM pojedynczej maszyny
- Stosujemy narzędzia, które skalują zadania na wiele maszyn przy zastosowaniu określonych metod
- Zastosowanie produkcyjne – wydajność, niezawodność procesu, duża skalowalność

Scikit-learn w Spark - uwaga, na jakiej maszynie (klastra) jest uruchamiany kod

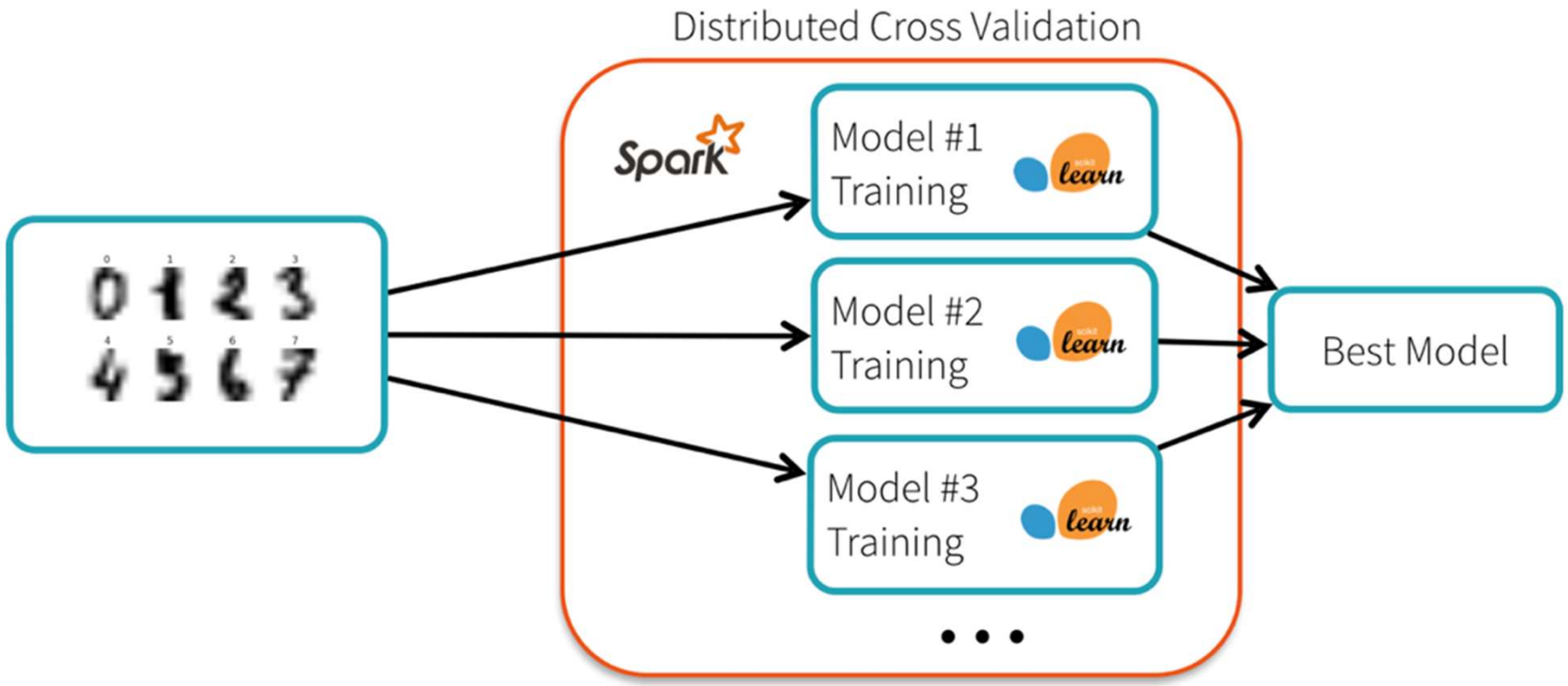
Spark-sklearn – rozproszenie zadań

Apache Spark Deep Learning Pipelines

Inne: ONNX, MLFlow, TensorFrames

SCENARIUSZE

Integracja Scikit-learn i Apache Spark



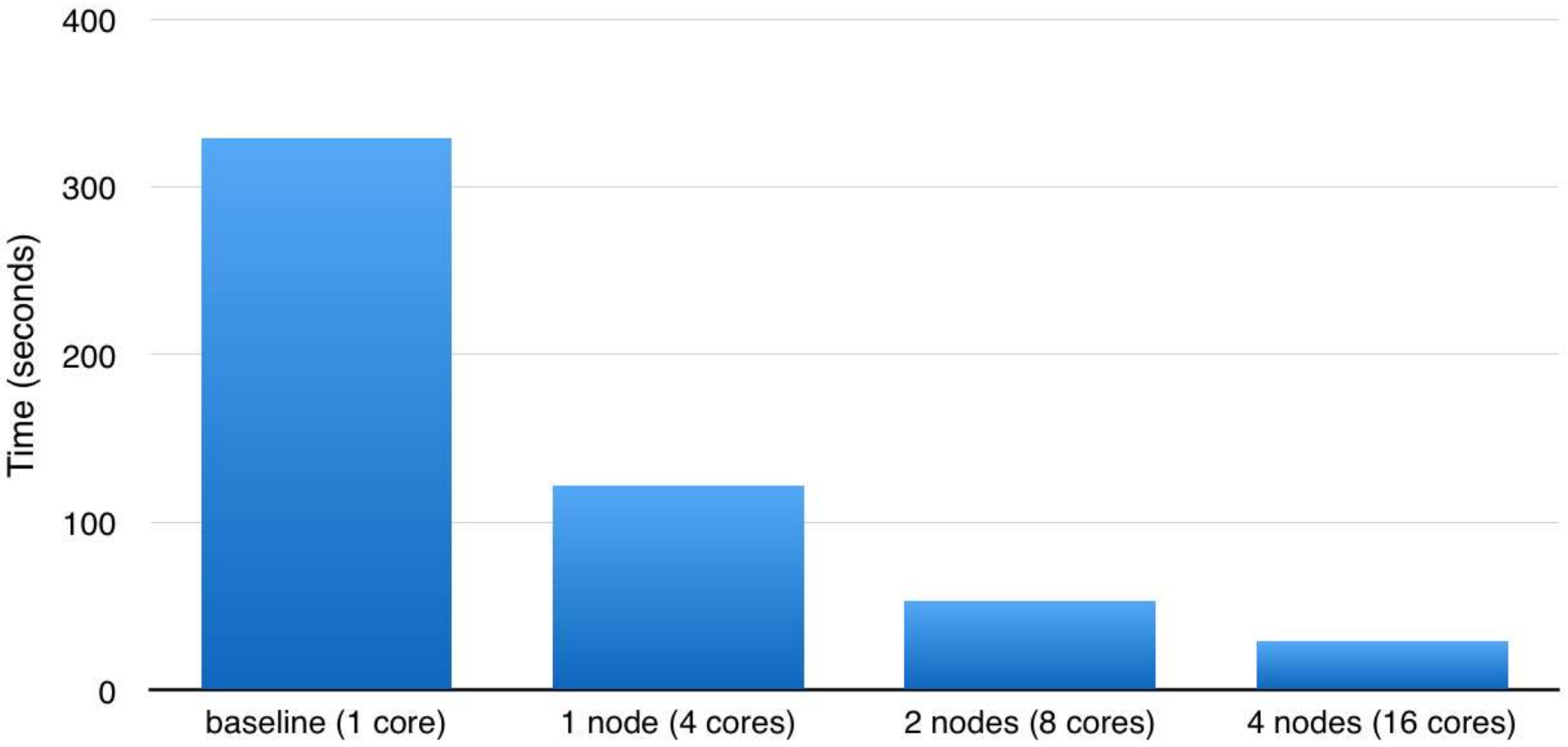
- uczenie i ocena wielu modeli scikit-learn jednocześnie (GridSearchCV)
- rozproszona implementacja analogiczna do wielordzeniowego zrównoleglania domyślnie zawartego w scikit-learn

SPARK - SKLEARN

Spark-SKLearn

DEMO

Spark-sklearn (wyniki orientacyjne)



https://databricks.com/blog/2016/02/08/auto-scaling-scikit-learn-with-apache-spark.html

Spark-sklearn – a jednak ograniczenia…

- Zalecane jest stosowanie biblioteki do „małej” ilości danych.
- Głównym celem stosowania biblioteki jest poszukiwanie parametrów za pomocą technik walidacji krzyżowej.
- Jednak dla dużych zbiorów danych, zalecane jest stosowanie natywnych dla Sparka metod uczenia maszynowego…
- Czyli `spark.ml` lub `spark.mllib`
- `I tutaj jest nadzieja…`

class spark_sklearn.Converter(sc)

- Klasa służąca do konwersji między modelami scikit-learn i Spark ML
- Jednak…
- `pyspark.ml.classification.LogisticRegressionModel` ⇔ `sklearn.linear_model.LogisticRegression`
  (tylko klasyfikacja binarna, bez wieloklasowej)
- `pyspark.ml.regression.LinearRegressionModel` ⇔ `sklearn.linear_model.LinearRegression`

SPARK ML I SPARK MLLIB

Biblioteka MLlib

MLlib obejmuje następujące klasy algorytmów i funkcji:

- Klasyfikacja - regresja logistyczna, naiwny klasyfikator bayesowski, drzewa decyzyjne i losowe lasy
- Regresja - uogólniona regresja liniowa i analiza przeżycia
- Rekomendacja - naprzemienne najmniejsze kwadraty (ALS)
- Klastrowanie - K-średnie i mieszanki Gaussa (GMM)
- Modelowanie tematyczne - alokacja ukryta Dirichleta (LDA)
- Częste zestawy przedmiotów, reguły asocjacji i eksploracja wzorów sekwencyjnych
- Rozproszona algebra liniowa - rozkład wartości pojedynczych (SVD), analiza głównych składowych (PCA)
- Statystyki - statystyki podsumowujące, testowanie hipotez, standaryzacja, normalizacja i wiele innych.

Czym to się różni

Spark ML

- spark.ml udostępnia API wyższego poziomu wbudowane w DataFrames służące do tworzenia potoków ML

Spark MLlib

- spark.mllib zawiera starsze API zbudowane na RDD

Podstawowym API uczenia maszynowego dla Sparka jest teraz interfejs API DataFrame w pakiecie spark.ml

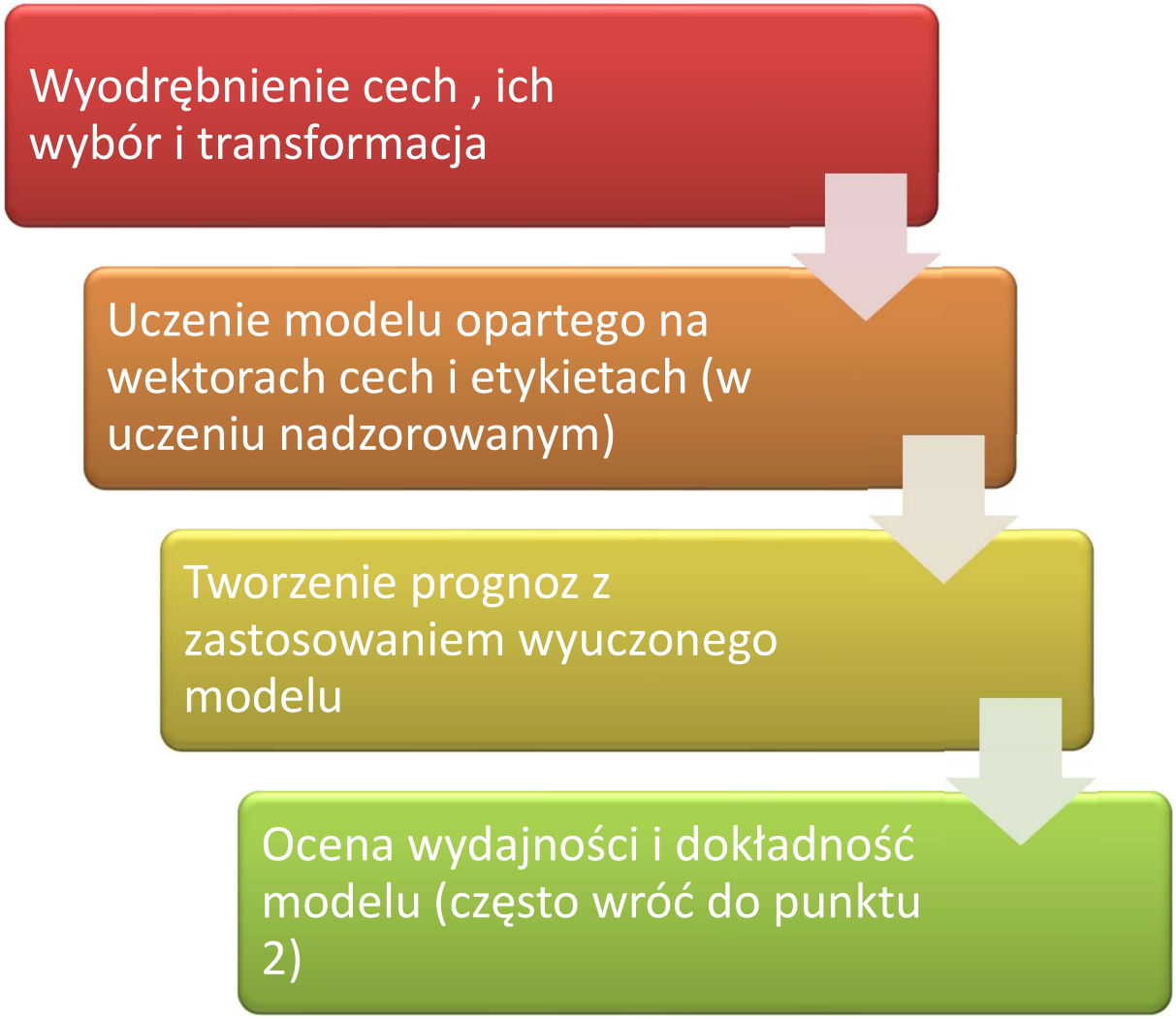
MLlib nadal będzie obsługiwał interfejs API oparty na RDD w spark.mllib z wdrażanymi poprawkami.

Jednak …

Uczenie maszynowe opiera się na uruchamianiu sekwencji algorytmów do przetwarzania i uczenia na podstawie danych.

Pipeline (potok)

Potok uczenia maszynowego w Apache Spark



Właściwości komponentów potoku

Transformer.transform() i Estimator.fit() obydwa komponenty są bezstanowe. W przyszłości stanowe algorytmy mogą być wspierane poprzez alternatywne koncepcje.

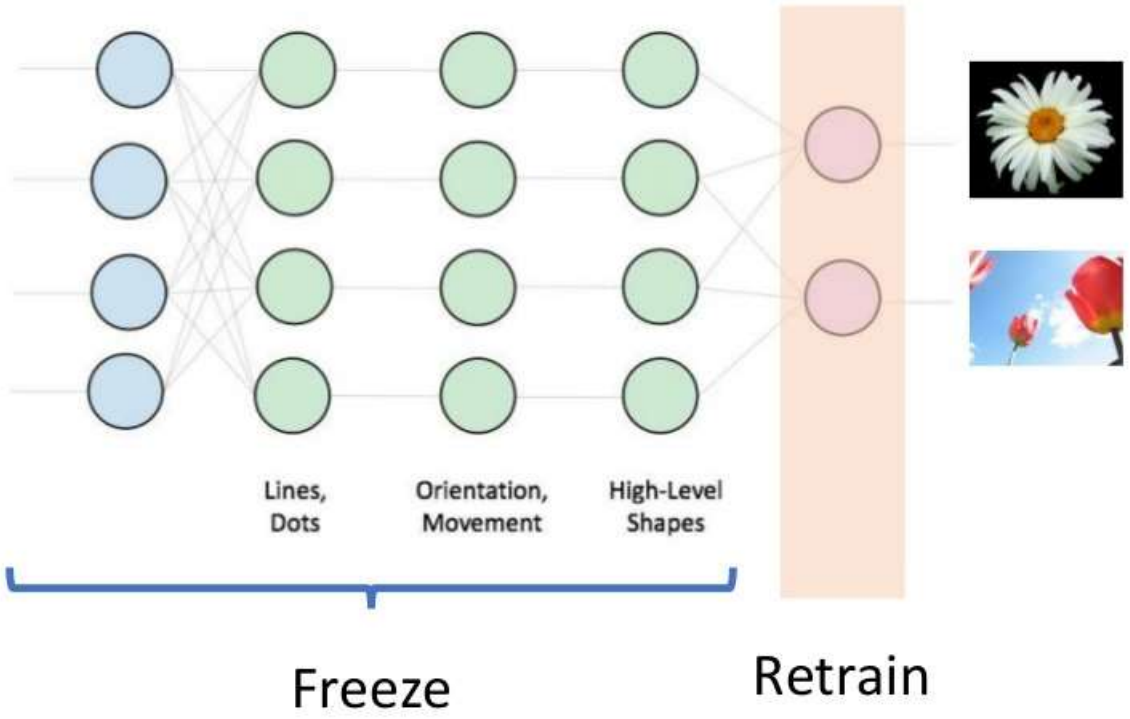
Każde wystąpienie transformera lub estymatora ma unikalny identyfikator, który jest przydatny w określaniu parametrów

SparkDL

Zadania realizowane przez bibliotekę DeepLearning Pipeline:

- ładowanie obrazów,
- stosowanie wstępnie wyuczonych modeli jako transformatorów w potoku Spark ML (Transfer learning)
- stosowanie modeli głębokiego uczenia w skali
- rozproszone dostrajanie hiperparametrów modeli
- wdrażanie modeli za pomocą Spark SQL

Transfer Learning (transfer poznania)

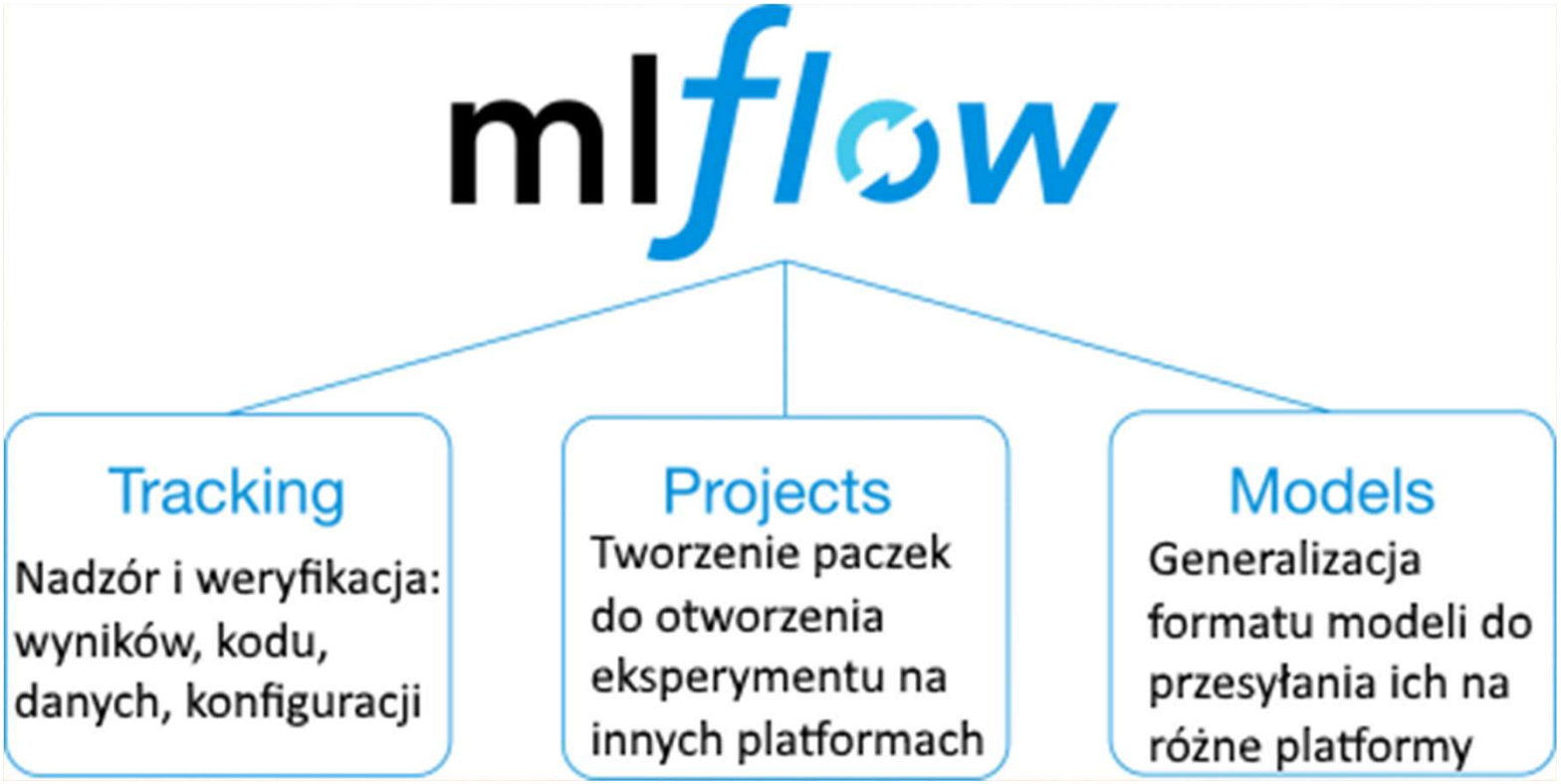


https://www.alexkalinin.com

SparkDL

DEMO

MLFlow


mlflow
Tracking
Nadzór i weryfikacja: wyników, kodu, danych, konfiguracji
Projects
Tworzenie paczek do otworzenia eksperymentu na innych platformach
Models
Generalizacja formatu modeli do przesyłania ich na różne platformy

MLFlow

- Języki Programowania
  - Python, R, Java
- Biblioteki uczenia maszynowego
  - TensorFlow, Keras, Spark ML, Scikit-Learn, PyTorch, mleap i inne
- Źródła danych
  - Amazon S3, Azure Storage, MS SQL i inne
- Systemy wdrożeniowe:
  - Docker, Apache Spark, Azure Machine Learning, Amazon SageMaker, ONNx, kubernetes, ONNX

Podsumowanie

- Zwróć uwagę z jakimi danymi masz do czynienia, zapoznaj się z nimi (CRISP-DM)
- Odpowiedz sobie na pytanie, jakiego typu narzędzia tworzące modele powinieneś wykorzystać
- Apache Spark ML to bogaty zbiór algorytmów uczenia maszynowego
- Jednak, istnieje wiele metod pozwalających wykorzystać zaawansowane biblioteki uczenia maszynowego (Scikit-Learn) lub ogólniej, sztucznej inteligencji (TensorFlow, Keras)

Materiały z prezentacji dostępne pod adresem:

https://github.com/tomyc/sqlday_2019

Data Community

SQL Day

PLATINUM SPONSOR

TECHNOLOGY INNOVATION DATA KNOWLEDGE tidk

STRATEGIC PARTNER

Microsoft

GOLD SPONSORS

CLOUDS ON MARS

Future Processing

SPYROSOFT

SILVER SPONSOR

dbWatch DATABASE CONTROL

elitmind think bright

VEEAM

BRONZE SPONSOR

Największa księgarnia IT w Polsce

NOVATECH

www.novatech.com.pl