



Metody i narzędzia *Big Data*

Uczenie maszynowe w ujęciu całościowym





Model wyrusza w świat

MLOps

środowisko uczenia maszynowego traktujemy systemowo

Czym jest uczenie maszynowe

1. Wydobywanie wzorców z danych = „uczenie” = dopasowanie modelu do danych
2. Prognozowanie z nowych danych

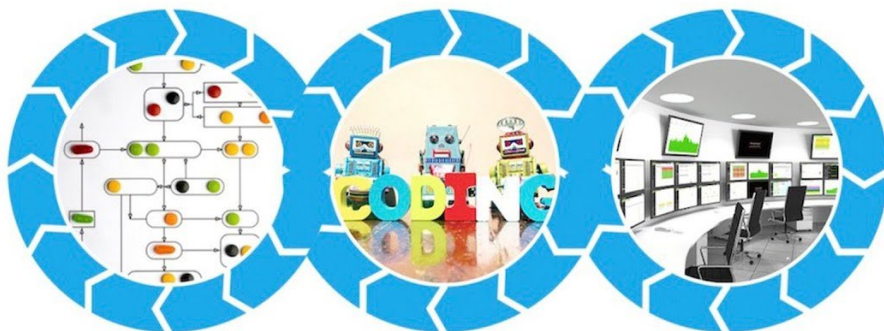
DevOps (ang. **Developments and Operations**)

Rozwój i Eksploatacja

Eksploatacja = stosowanie w produkcji

- wdrożenie
- monitorowanie
- utrzymanie

MLOPS = ML + Dev + Ops



ML

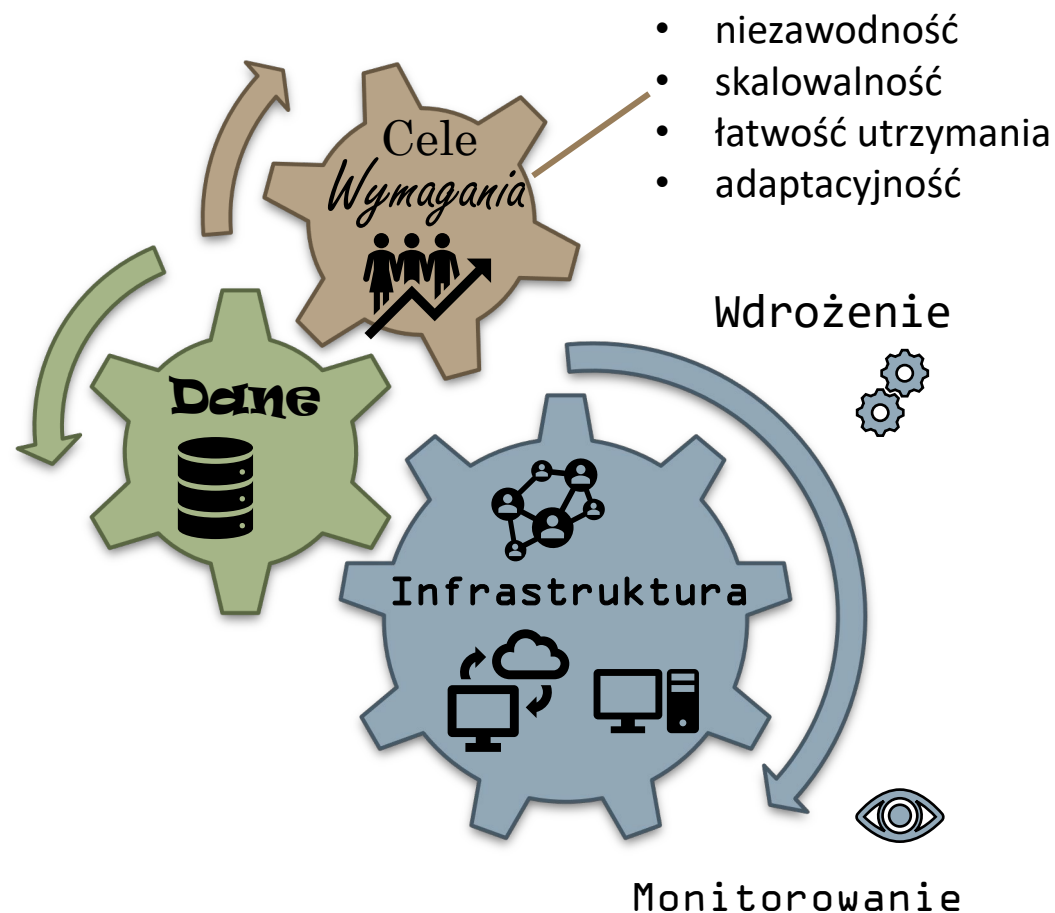
Experiment, Train & Tune
ML to find
Optimal Model

Dev

Develop CI / CD
Pipelines to deploy
Models

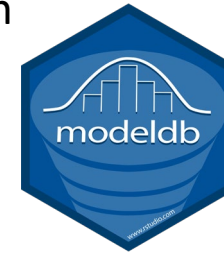
Ops

Monitor & Manage
Models
at scale



Różnice MLOps a DevOps

- Tradycyjna inżynieria oprogramowania: kod i **dane** są oddzielone
- System uczenia maszynowego = kod + **dane** + *artefakty*
- Ulepszać algorytmy czy ulepszać dane: testowanie i wersjonowanie danych



Środowisko badawcze a środowisko produkcyjne

- Ewaluacja modelu
 - kalibracja modelu (*próg pewności*)
 - testy **niezmienności** (*zgoda na kredyt*)
 - ewaluacja na **wycinkach** (*podzbiór mniejszościowy, krytyczny*)



środowiska

projektowe

testowe

produkcyjne

Eksportowanie modelu = serializacja



Testowanie modelu w ...

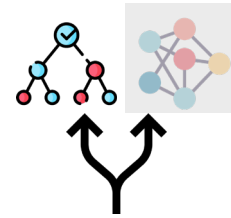
Testowanie modelu w „produkcji”

- Model kandydujący



- Kopia rozwiązania (ang. *shadow deployment*)

- podwaja się liczba prognoz
- niskie ryzyko wpadki



- Testy A/B

- Dane są losowo kierowane do modelu A lub B

- Testy kanarkowe (ang. *canary release*)

- stopniowe wprowadzanie modelu do niewielkiej grupy użytkowników (na ogół mniej krytycznej)



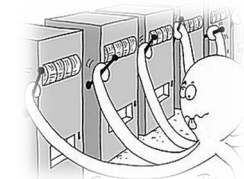
- Eksperymenty przeplatane (ang. *interleaving experiments*)

- użytkownik otrzymuje prognozy/rekomendacje z obu modeli i wybiera jedną




- Algorytmy bandyty

- eksploracja / eksploatacja poprzez randomizowane kierowanie żądań do modeli i pomiar jakości



Ważne w przemyśle, nie w nauce

- Wydajność nowej technologii **może** poprawić się z czasem
 - nieudane wdrożenie lub zostać **w tyle** za konkurencją
- Różni interesariusze mają **różne wymagania** do jednego systemu
- Dane są niechlujne i **zmieniają się**
- Magia  dla klienta i dla inwestorów
- Możliwa jest **awaria** bez sygnału o błędzie
- System nie jest ukończony, ale trzeba go **udostępnić**
- **Wsparcie** społeczności
- Koszty ulepszeń
- Interpretowalność a wydajność
- Modele zespołowe (ang. *ensemble models*) nie takie dobre
- Czas obliczeń a **czas inżynierów**

Julia Evans

ABOUT
TALKS
PROJECTS
TWITTER
MASTODON
GITHUB

FAVORITES ★ ZINES ★ RSS

Machine learning isn't Kaggle competitions

• MACHINELEARNING •

I write about strace and kernel programming on this blog, but at work I actually mostly work on machine learning, and it's about time I started writing about it! Disclaimer: I work on a data analysis / engineering team at a tech company, so that's where I'm coming from.

Lauren Oakden-Rayner

Medical AI Researcher / Radiologist

ABOUT ME EDUCATION MY MOOCS (WITH REVIEWS) BLOG



AI competitions don't produce useful models

SEPTEMBER 19, 2019 - LAURENOAKDENRAYNER

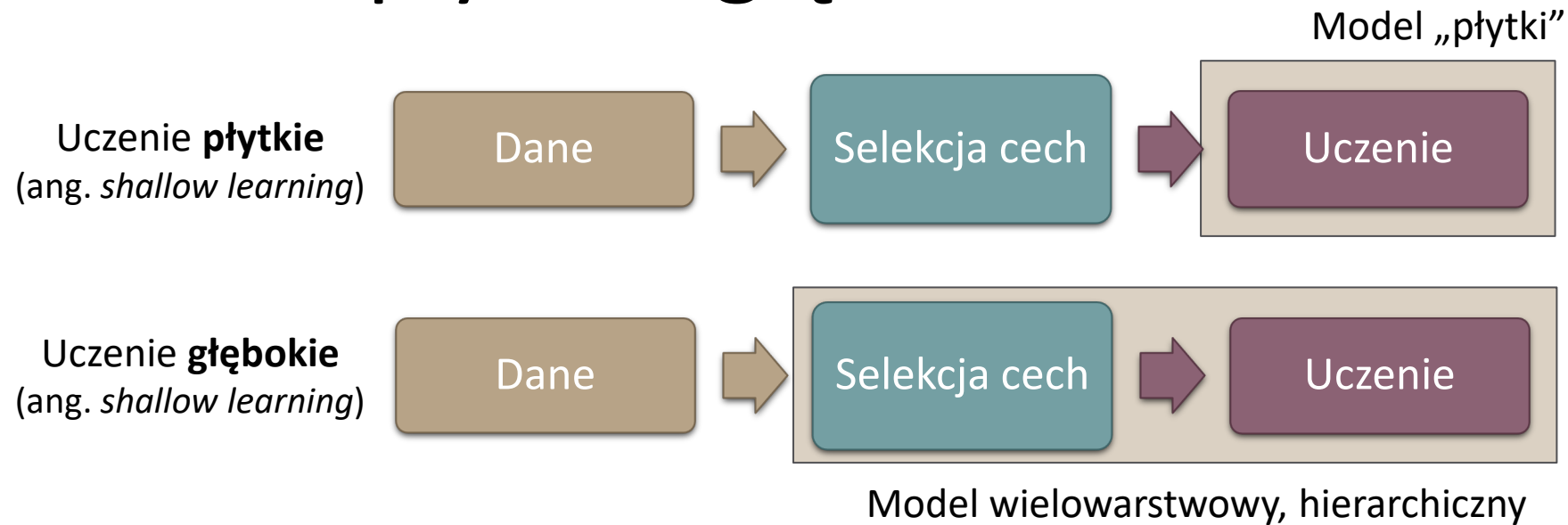
A huge new CT brain dataset was released today, with the goal of training

Archives

July 2022 (1)

March 2022 (1)

Uczenie płytkie i głębokie



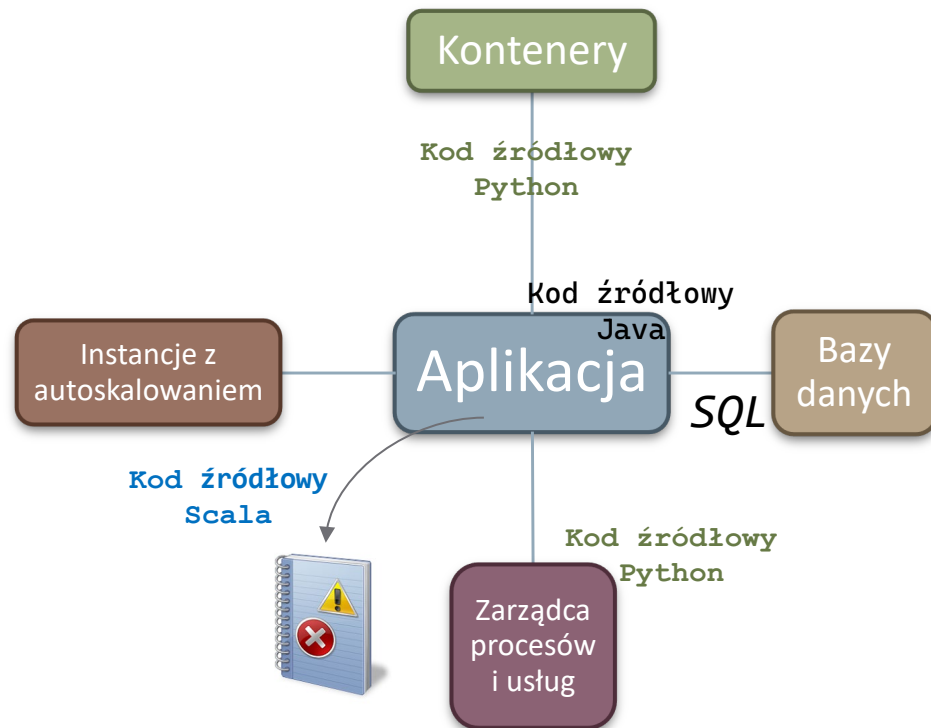
- Optymalizacja sprzętu na potrzeby popularnych modeli
- Uczenie transferowe
- Osadzenia
- W środowisku produkcyjnym zmieniają się kategorie/etykiety

Kiedyś to było...

1 aplikacja = 1 usługa

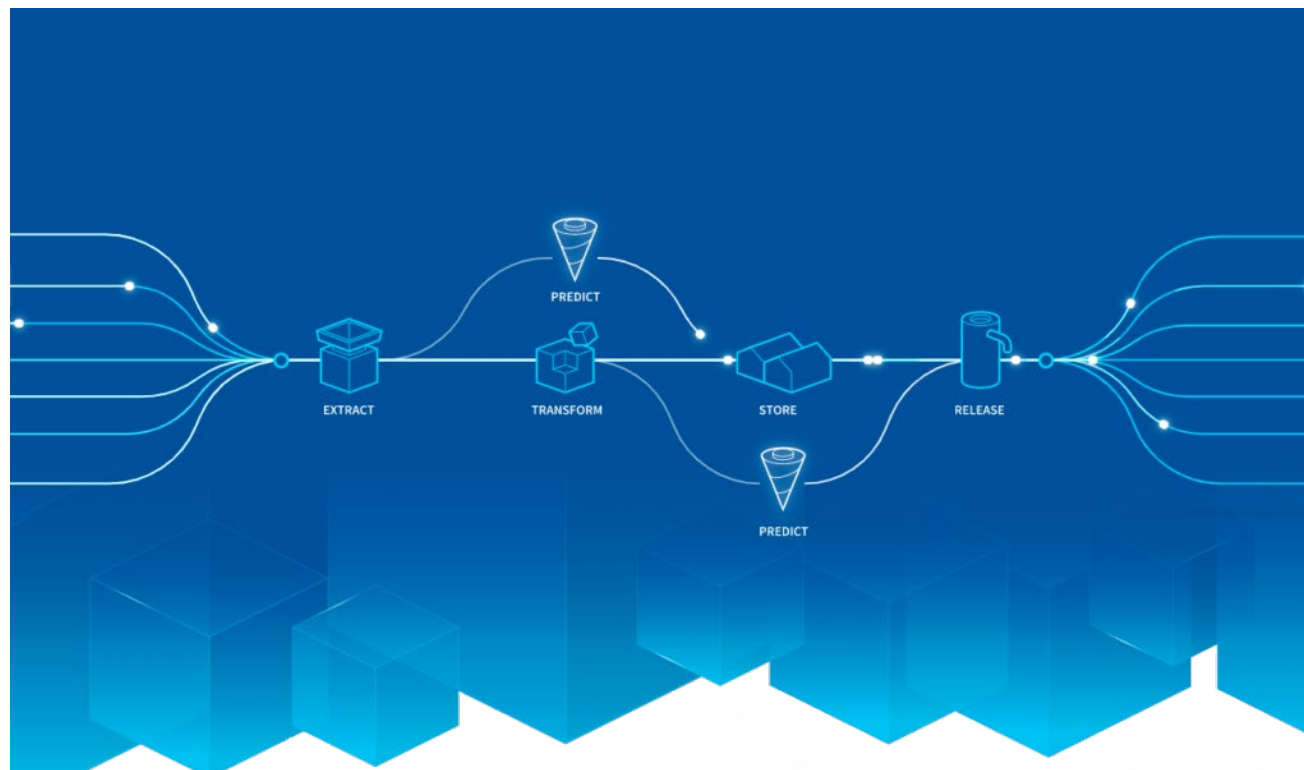


A dziś, to panie ...



• Uczenie ciągłe

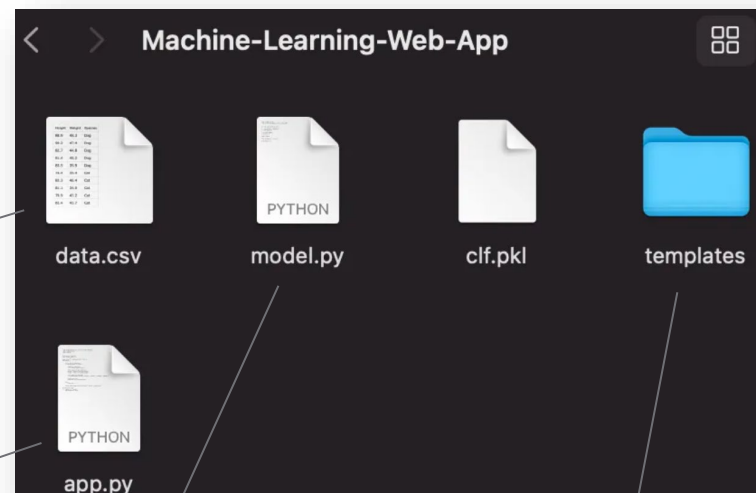
- cron, zarządca procesów i orkiestratory
- aktualizacja modelu na podstawie mikropaczek/mikrowsadów
- iteracja modelu, iteracja danych



Platformy uczenia maszynowego

Pierwsze kroki

Height	Weight	Species
88.9	48.3	Dog
90.2	47.4	Dog
82.7	44.8	Dog
81.4	48.2	Dog
83.5	39.9	Dog
76.4	35.4	Cat
82.3	40.4	Cat
81.1	39.9	Cat



```
from flask import Flask, request, render_template
import pandas as pd
import joblib

# Declare a Flask app
app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def main():

    # If a form is submitted
    if request.method == "POST":

        # Unpickle classifier
        clf = joblib.load("clf.pkl")

        # Get values through input bars
        height = request.form.get("height")
        weight = request.form.get("weight")

        # Put inputs to dataframe
        X = pd.DataFrame([[height, weight]], columns = ["Height", "Weight"])

        # Get prediction
        prediction = clf.predict(X)[0]
```

```
import pandas as pd
from sklearn.naive_bayes import GaussianNB
import joblib

df = pd.read_csv("data.csv")

X = df[["Height", "Weight"]]
y = df["Species"]

clf = GaussianNB()
clf.fit(X, y)

joblib.dump(clf, "clf.pkl")
```

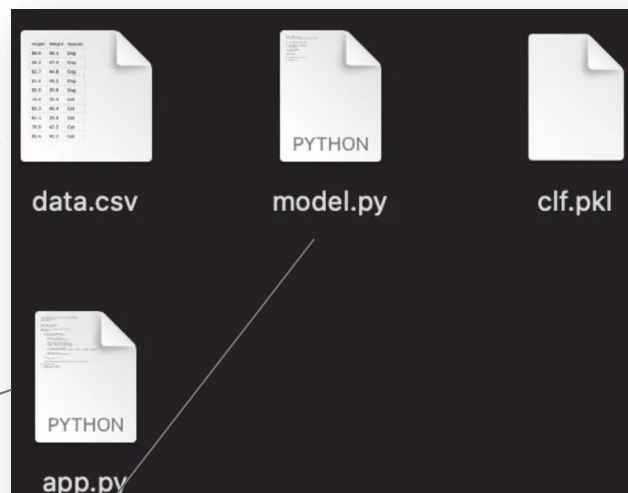
Height:

Weight:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Your Machine Learning App</title>
  </head>
  <body>
    <form name="form", method="POST", style="text-align: center;">
      <br>
      Height: <input type="number" name="height", placeholder="Enter height in cm" required/>
      <br><br>
      Weight: <input type="number" name="weight", placeholder="Enter weight in kg" required/>
      <br><br>
      <button value="Submit">Run</button>
    </form>
    <p style="text-align: center;">{{ output }}</p>
  </body>
</html>
```

Pierwsze kroki

Height	Weight	Species
88.9	48.3	Dog
90.2	47.4	Dog
82.7	44.8	Dog
81.4	48.2	Dog
83.5	39.9	Dog
76.4	35.4	Cat
82.3	40.4	Cat
81.1	39.9	Cat



```
import streamlit as st
import pandas as pd
import joblib

# Title
st.header("Streamlit Machine Learning App")

# Input bar 1
height = st.number_input("Enter Height")

# Input bar 2
weight = st.number_input("Enter Weight")

# Dropdown input
eyes = st.selectbox("Select Eye Colour", ("Blue", "Brown"))

# If button is pressed
if st.button("Submit"):

    # Unpickle classifier
    clf = joblib.load("clf.pkl")
```

```
import pandas as pd
from sklearn.naive_bayes import GaussianNB

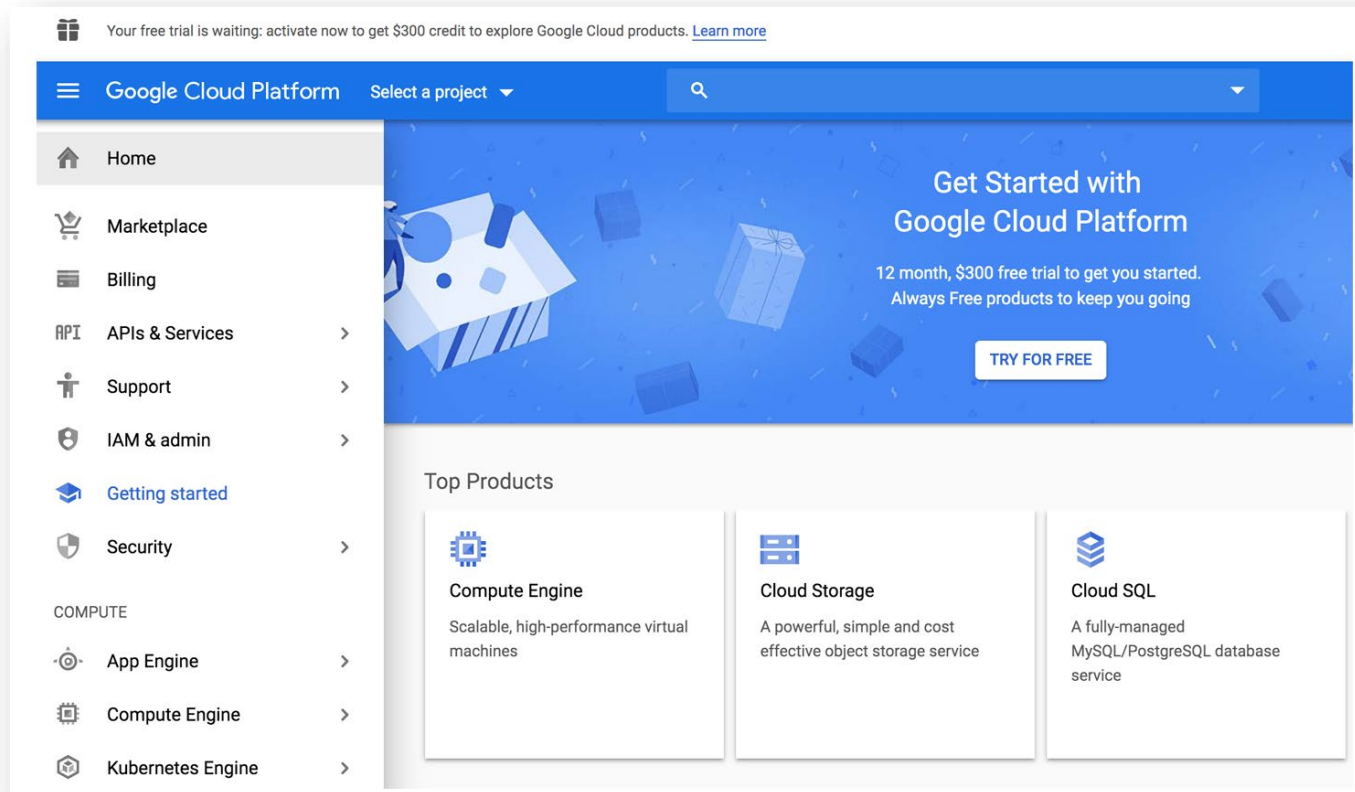
df = pd.read_csv("data.csv")

X = df[["Height", "Weight"]]
y = df["Species"]

clf = GaussianNB()
clf.fit(X, y)
```

The screenshot shows the user interface of the Streamlit Machine Learning App. It has a dark theme. At the top, it says "Streamlit Machine Learning App". Below that, there are three input fields: "Enter Height" with a value of 0.00, "Enter Weight" with a value of 0.00, and "Select Eye Colour" with a dropdown menu showing "Blue". At the bottom, there is a "Submit" button.

Platforma uczenia maszynowego



← Create version

To create a new version of your model, make necessary adjustments to your saved model file before exporting and store your exported model in Cloud Storage. [Learn more](#)

Name
v0001

Name cannot be changed, is case sensitive, must start with a letter, and may only contain letters, numbers, and underscores. 5 / 128

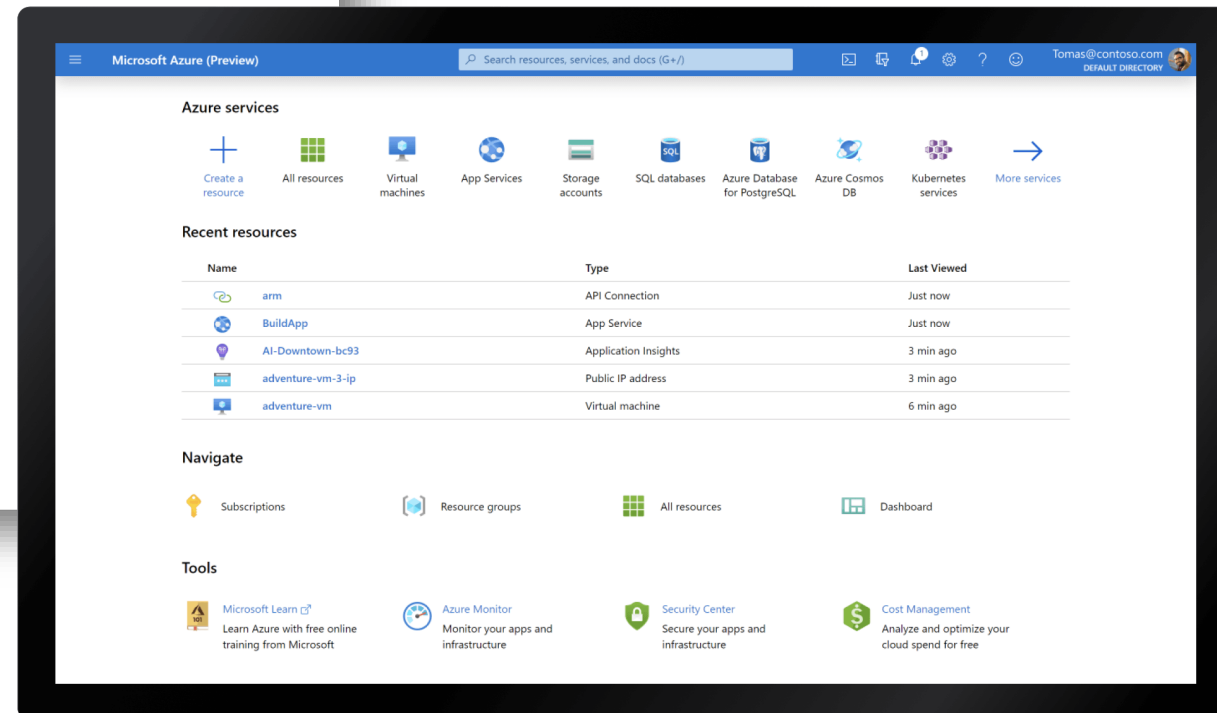
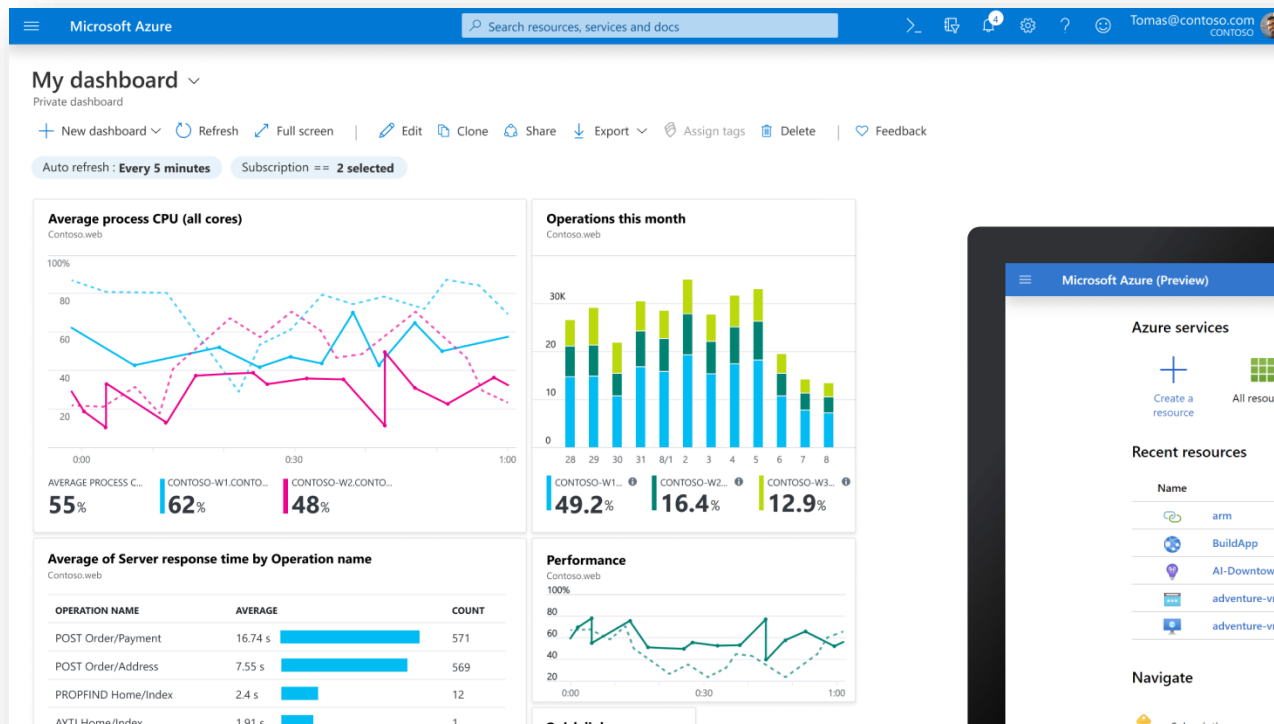
Description
Dense net with 2 layers (100, 10 units)

Python version
3.5

Select the Python version you used to train the model

Framework
TensorFlow

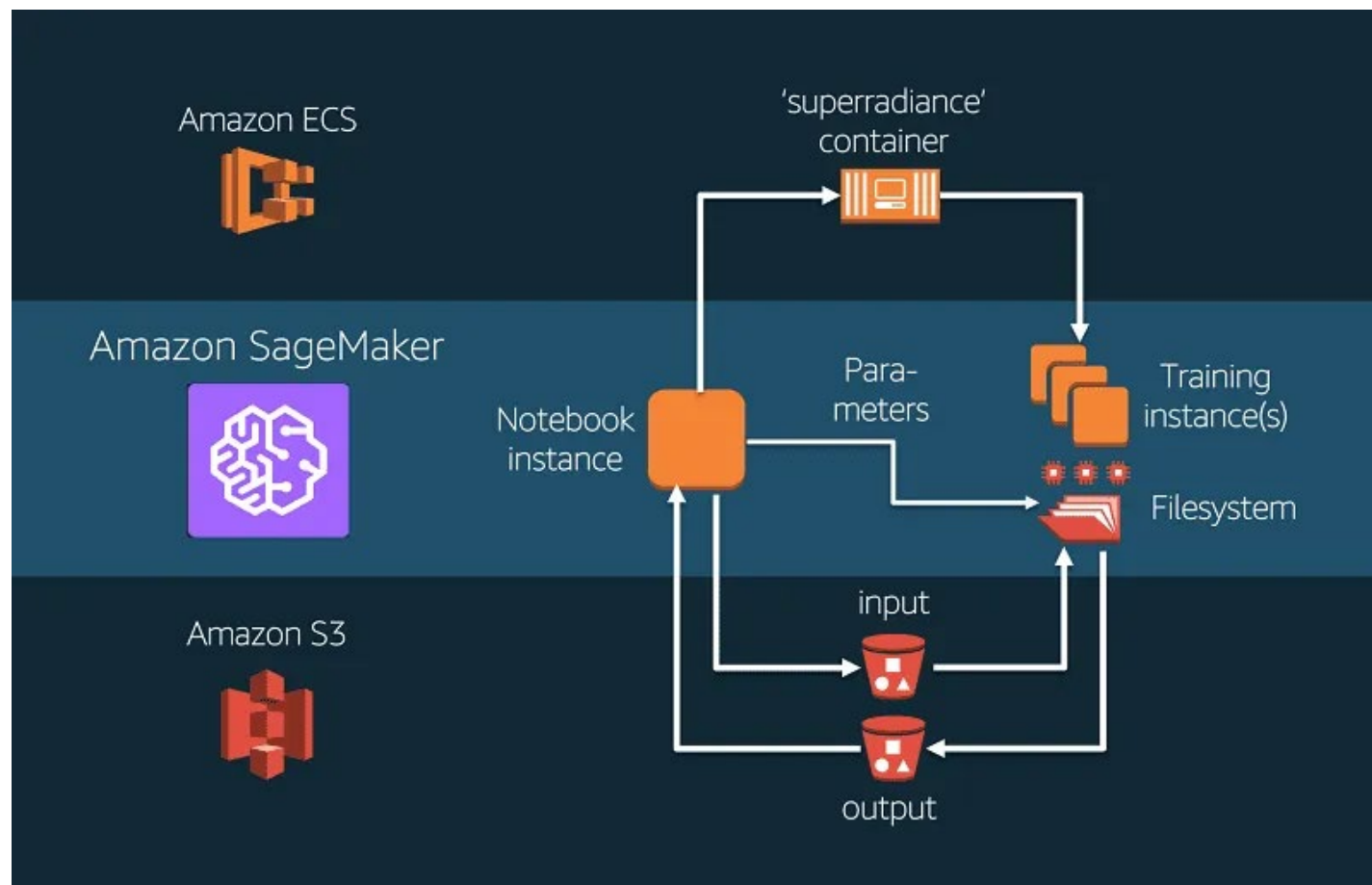
Platforma uczenia maszynowego



Platforma uczenia maszynowego



Platforma uczenia maszynowego

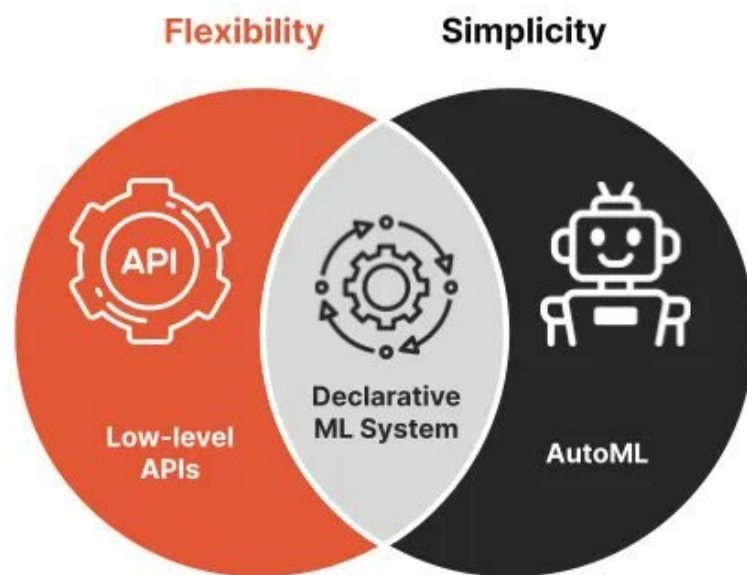


Platforma uczenia maszynowego



Platforma uczenia maszynowego

DECLARATIVE MACHINE LEARNING SYSTEMS



Higher abstraction provides flexibility, automation, ease of use



Opens the door for **non experts** to use the power of ML



We **pioneered** them since 2017 with **Ludwig** (Uber) and **Overton** (Apple) ACM CACM

Beyond Interactive: Notebook Innovation at Netflix



Netflix Technology Blog · Follow

Published in Netflix TechBlog · 13 min read · Aug 16, 2018



32

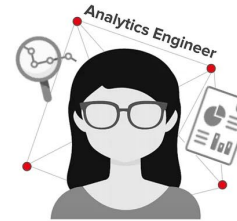


By [Michelle Ufford](#), [M Pacer](#), [Matthew Seal](#), and [Kyle Kelley](#)

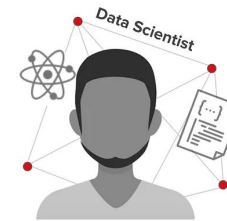
Notebooks have rapidly grown in popularity among data scientists to become the de facto standard for quick prototyping and exploratory analysis. At Netflix, we're pushing the boundaries even further, reimagining what a notebook can be, who can use it, and what they can do with it. And we're making big investments to help make this vision a reality.

Our users have found a surprising number of uses for these templates. Some of the most common ones are:

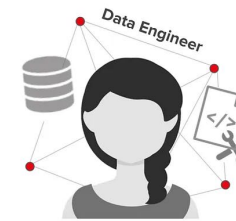
- **Data Scientist:** run an experiment with different coefficients and summarize the results
- **Data Engineer:** execute a collection of data quality audits as part of the deployment process
- **Data Analyst:** share prepared queries and visualizations to enable a stakeholder to explore more deeply than Tableau allows
- **Software Engineer:** email the results of a troubleshooting script each time there's a failure



tools: Sublime, Atom, Tableau
languages: SparkSQL, Presto, Python



tools: Jupyter, RStudio, PyCharm
languages: Python, Presto, R, PySpark



tools: IntelliJ, PyCharm, Sublime
languages: Scala, Spark, Python, SQL

nteract is a next-gen React-based UI for Jupyter notebooks. It provides a simple, intuitive interface and offers several improvements over the classic Jupyter UI, such as inline cell toolbars, drag and droppable cells, and a built-in data explorer.

Papermill is a library for parameterizing, executing, and analyzing Jupyter notebooks. With it, you can spawn multiple notebooks with different parameter sets and execute them concurrently. Papermill can also help collect and summarize metrics from a collection of notebooks.

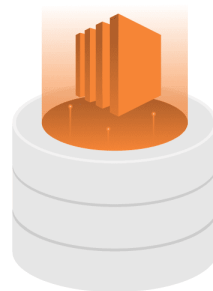
Commuter is a lightweight, vertically-scalable service for viewing and sharing notebooks. It provides a Jupyter-compatible version of the contents API and makes it trivial to read notebooks stored locally or on Amazon S3. It also offers a directory explorer for finding and sharing notebooks.

Titus is a container management platform that provides scalable and reliable container execution and cloud-native integration with Amazon AWS. Titus was built internally at Netflix and is used in production to power Netflix streaming, recommendation, and content systems.

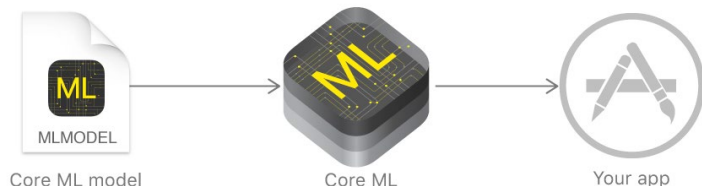
Platforma uczenia maszynowego

- MOC OBLICZENIOWA
- Chmura obliczeniowa

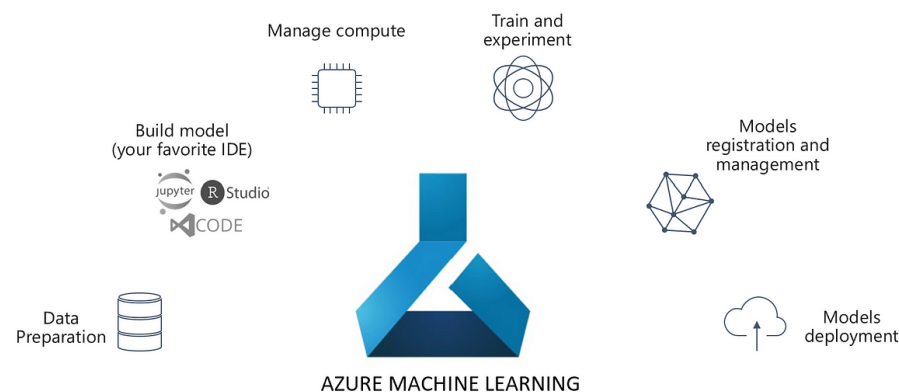
Amazon
EC2



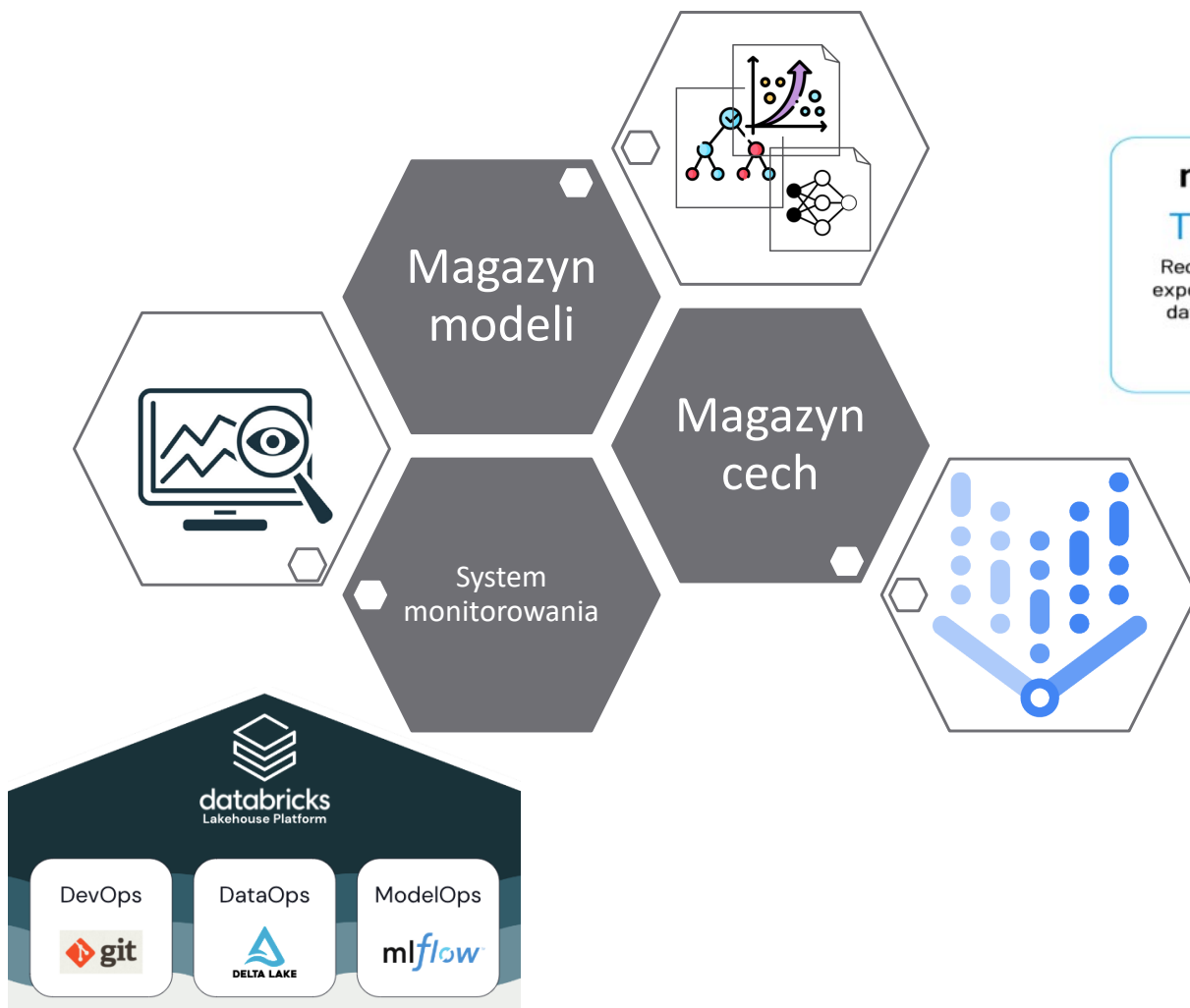
Google Cloud Platform



Spark MLlib



Platforma uczenia maszynowego



MLflow Components

mlflow Tracking

Record and query experiments: code, data, config, and results

mlflow Projects

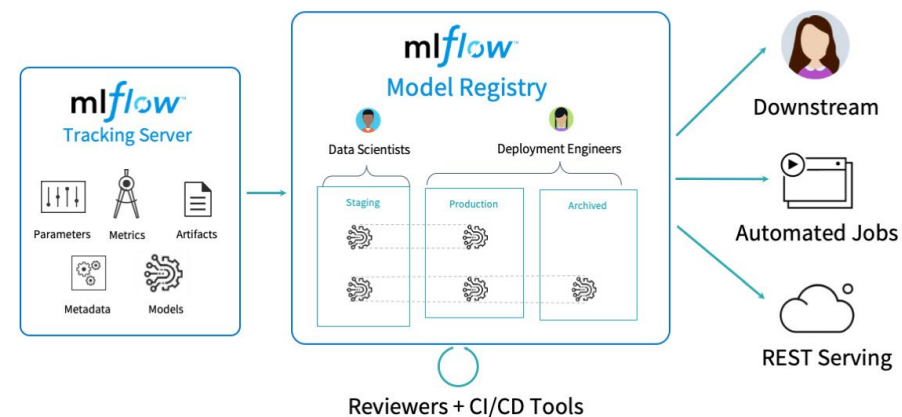
Package data science code in a format that enables reproducible runs on any platform

mlflow Models

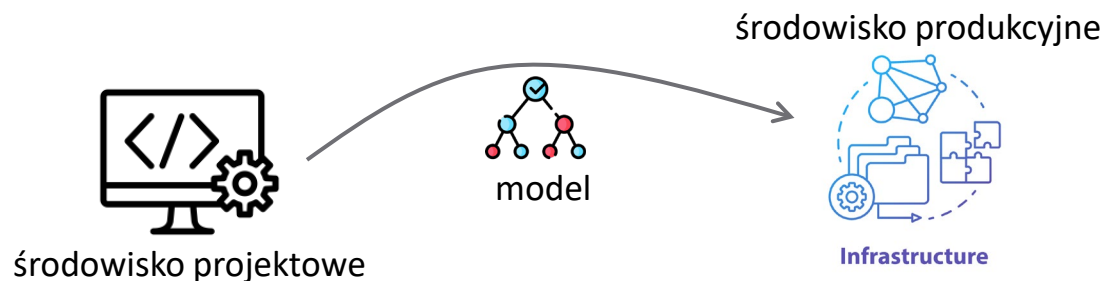
Deploy machine learning models in diverse serving environments

mlflow Model Registry

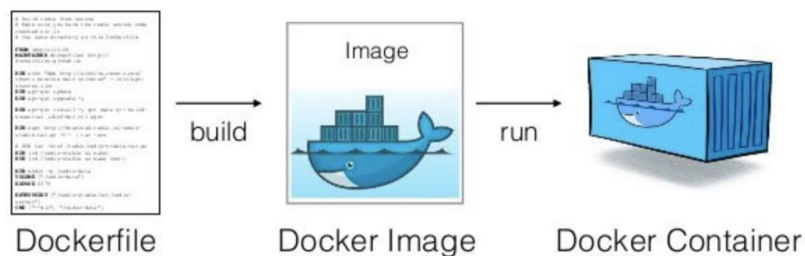
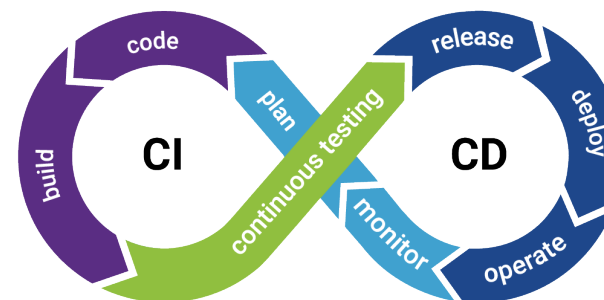
Store, annotate and manage models in a central repository



Platforma uczenia maszynowego

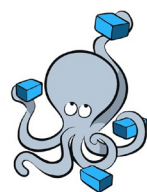


CI/CD – ciągła integracja, ciągłe wdrażanie



Autoskalowanie:
nowa instancja → nowe środowisko

każda nowa instancja to kontener Dockera



docker
Compose

Orkiestrator kontenerów

1 komputer

wiele komputerów

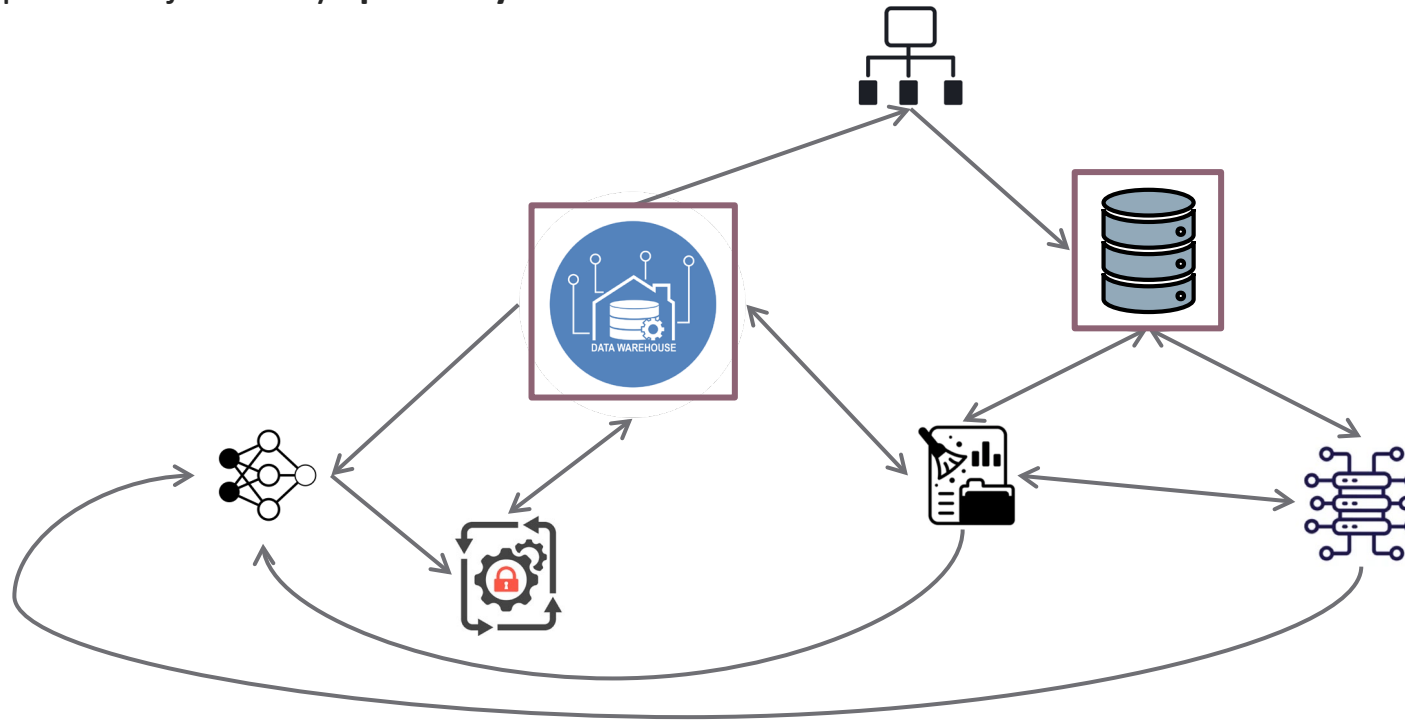


kubernetes



Przepływ danych pomiędzy procesami

- Tryb przekazywania przez bazy danych
 - konieczny dostęp procesów do tych samych baz danych
 - odczyt z bazy i zapis do niej może być **powolny**

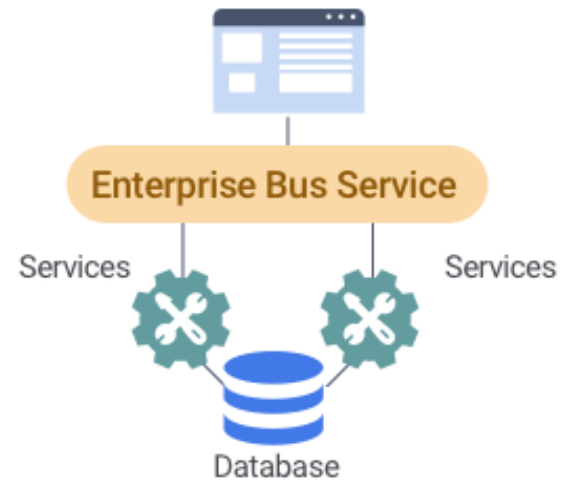


Przepływ danych pomiędzy procesami

- Tryb sterowany żądaniem / przekazywanie danych przez usługi
 - Do usługi można uzyskać dostęp zdalnie
 - Synchroniczne: usługa następuje żądania

SOA vs Microservices

Service Oriented Architecture



Vs

Microservices



Przepływ danych pomiędzy procesami

- Tryb sterowany zdarzeniami / przekazywanie przez połączenia w czasie rzeczywistym

- Pośrednik (ang. **broker**) koordynuje przekazywanie danych między usługami

- Typ połączeń **pubsub** (ang. *publish-subscribe*)



- Typ połączeń **kolejka komunikatów**



Przepływ danych pomiędzy procesami

- **Cron** – planowanie uruchomienia powtarzalnych zadań o wyznaczonym z góry czasie
- **Zarządcy** procesów = cron + zależności między zadaniami oparte o DAG + przydział zasobów do okresowych zadań
- **Orkiestratory** – rezerwacja zasobów (instancje, klastry) przydzielonych przez zarządcę do zadań wynikających z żądań



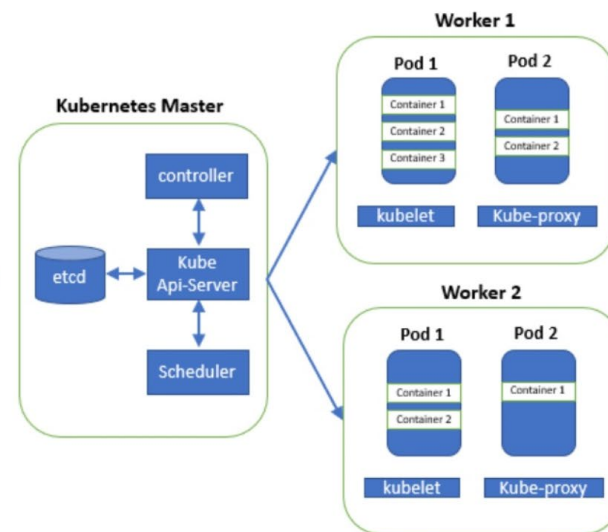
データ基盤に
ArgoWorkflowsを
導入した話



Apache
Airflow



Kubectl / APIs / Dashboard

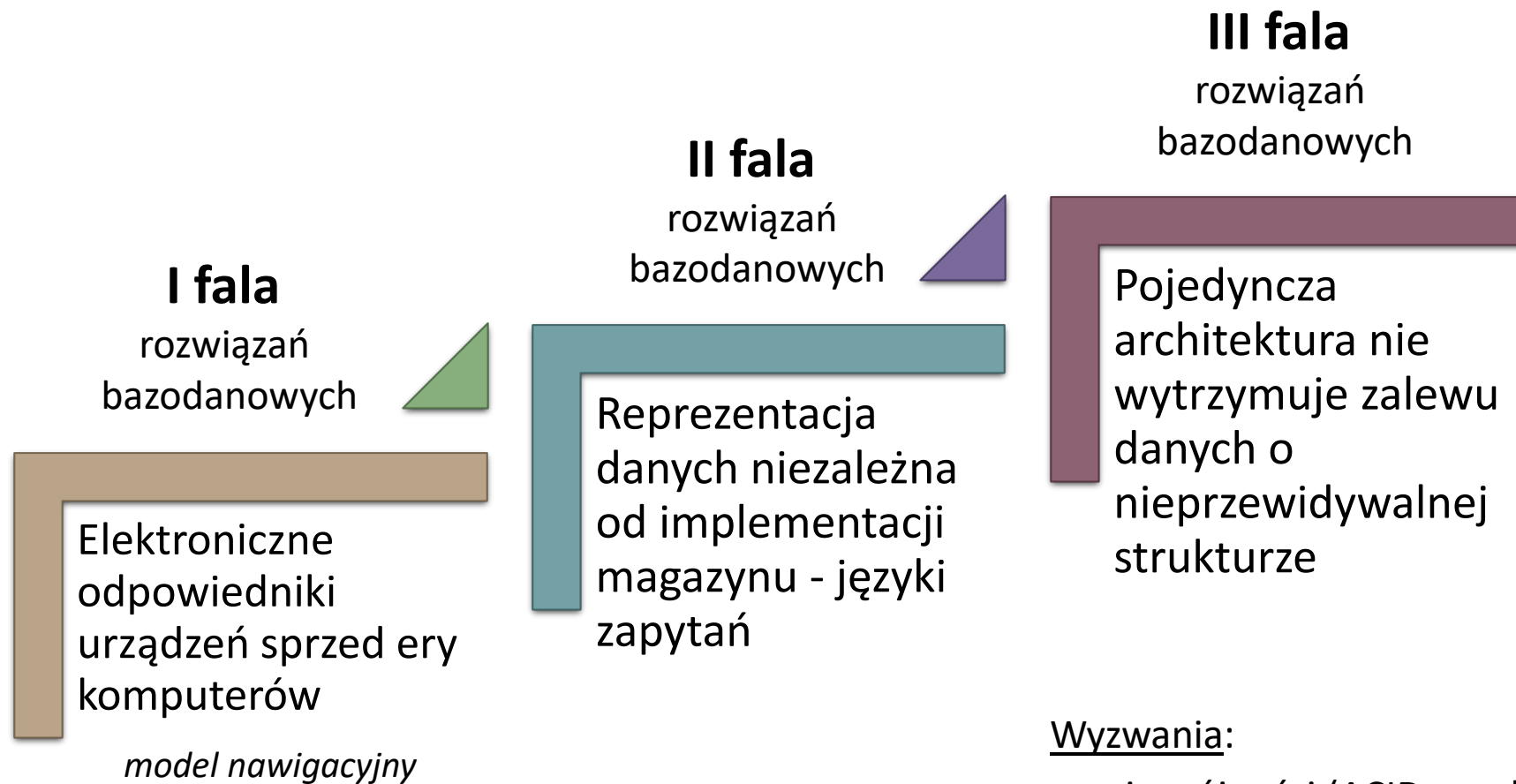


Kubeflow



PREFECT

Coordinating the world's dataflow.



Wyzwania:

- niespójności (ACID, model łańcucha bloków)
- powrót modelu nawigacyjnego (bardziej dla programisty niż „widok” dla analityka)
- trudności z integracją z narzędziami BI
- przekleństwo wyboru

Modele danych

- Model relacyjny (1970) Edgar F. Codd
 - Dane są zorganizowane w relacje (zbiory krotek = wiersze tabeli)
 - Kolejność wierszy i kolumn nie ma znaczenia
 - Normalizacja rozrzuca dane do wielu tabel
 - Język zapytań SQL: deklaratywny
- Model dokumentowy
 - Dane są dostarczone w dokumentach o ustalonym formacie, np. JSON, XML, BSON
 - Relacje występują rzadko
 - Zbiór dokumentów = tabela, dokument = wiersz tabeli
 - Dokumenty w jednej relacji mogą mieć różne schematy
 - Aplikacja zewnętrzna odpowiada za obsługę struktur dokumentów
- Model Grafowy
 - najważniejsze są relacje pomiędzy elementami
 - węzły mogą mieć różne typy



Modele danych

- Model **relacyjny** (1970) Edgar F. Codd
- Model **dokumentowy**
- Model **grafowy**
- Model obiektowy
- Model **kolumnowy**
- Model krotek klucz-wartość
- Modele dla pamięci podręcznych
- Model rozproszony
- Model strumieniowy
- Model transakcyjny



<https://hostingdata.co.uk/nosql-database/>



You

what is data megastructure?



ChatGPT

A data megastructure typically refers to a large-scale, highly organized framework or system designed to handle vast amounts of data efficiently. It often involves various components such as databases, data warehouses, data lakes, and other infrastructure elements. These structures are crucial for organizations dealing with massive volumes of data, such as internet companies, financial institutions, research organizations, and government agencies.

Data megastructures are designed to store, manage, process, and analyze huge datasets in a way that enables easy access, scalability, security, and reliability. They often incorporate advanced technologies like cloud computing, distributed computing, big data analytics, and artificial intelligence to handle the complexities of large-scale data management.

The term "megastructure" implies a scale beyond traditional data management systems, suggesting a massive, integrated architecture capable of handling petabytes or even exabytes of data from various sources and in various formats.



Megastruktury danych

Dane nieustrukturyzowane

- **Szybki dostęp** do danych
- Obsługa schematu przerzucona na **aplikacje**, które używają danych
- Repozytorium na nieprzetworzone dane to **jezioro danych** (ang. *data lake*)



Dane ustrukturyzowane

- Konieczność trzymania się zadanego z góry **schematu**
- Zmiana schematu → wsteczna aktualizacja danych
- Gdy dane pochodzą z wielu źródeł, trudno wymóc spójny schemat
- **Łatwo wyszukiwać i analizować** dane
- Repozytorium w gotowych do użycia formatach to **hurtownia danych**



DATA LAKE

vs

DATA WAREHOUSE



Raw

Data Lakes contain unstructured, semi structured and structured data with minimal processing. It can be used to contain unconventional data such as log and sensor data

Large

Data Lakes contain vast amounts of data in the order of petabytes. Since the data can be in any form or size, large amounts of unstructured data can be stored indefinitely and can be transformed when in use only

Undefined

Data in data lakes can be used for a wide variety of applications, such as Machine Learning, Streaming analytics, and AI



Refined

Data Warehouses contain highly structured data that is cleaned, pre-processed and refined. This data is stored for very specific use cases such as BI.

Smaller

Data Warehouses contain less data in the order of terabytes. In order to maintain data cleanliness and health of the warehouse, Data must be processed before ingestion and periodic purging of data is necessary

Relational

Data Warehouses contain historic and relational data, such as transaction systems, operations etc



The Databricks Lakehouse Platform

Unify your data warehousing and AI use cases on a single platform

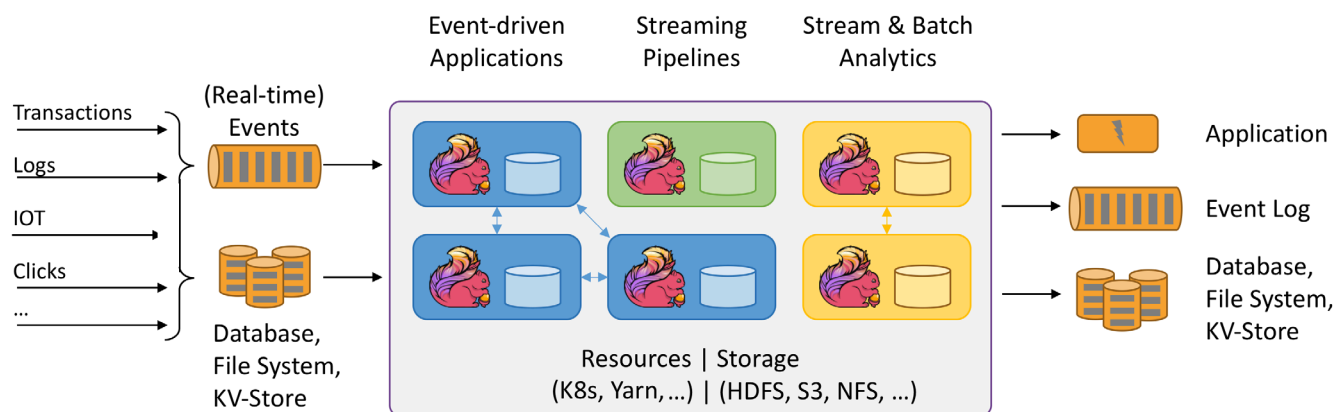
Przetwarzanie wsadowe i strumieniowe

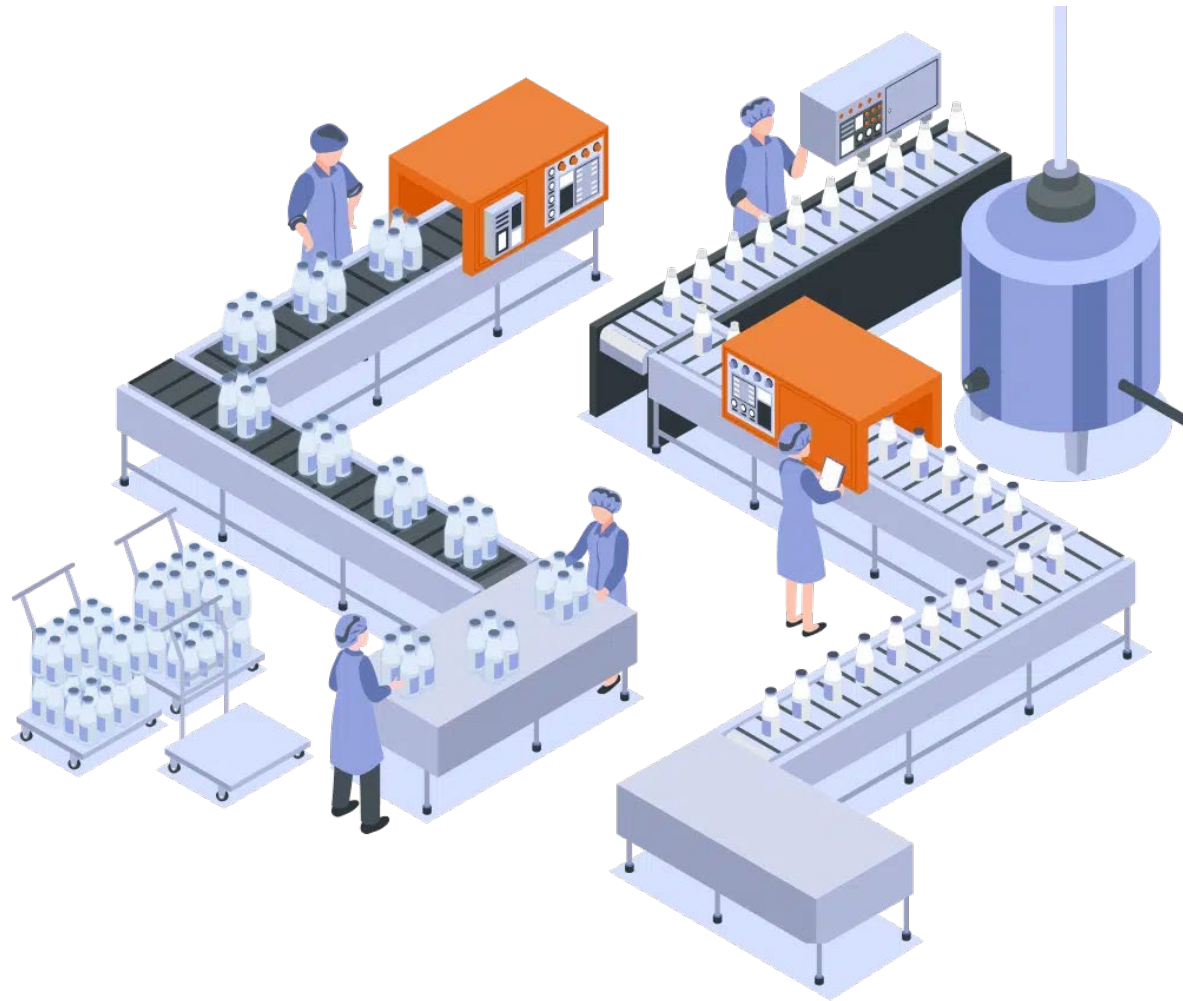
- Wsadowe



- Strumieniowe

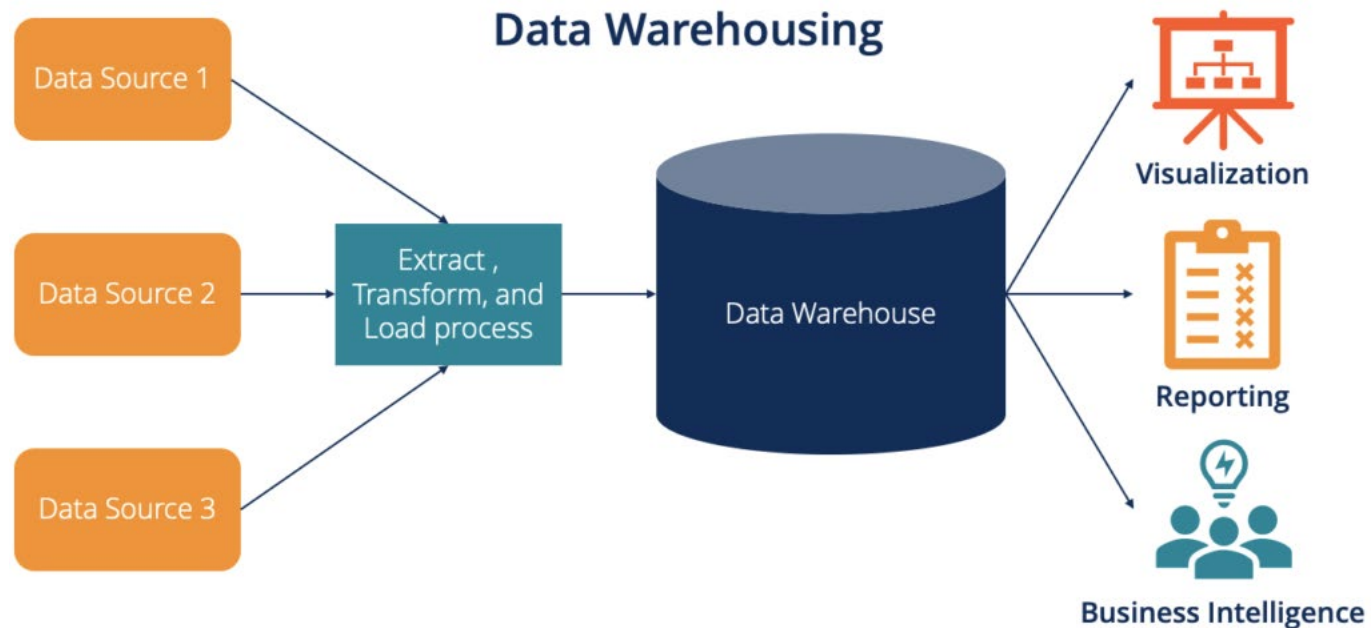
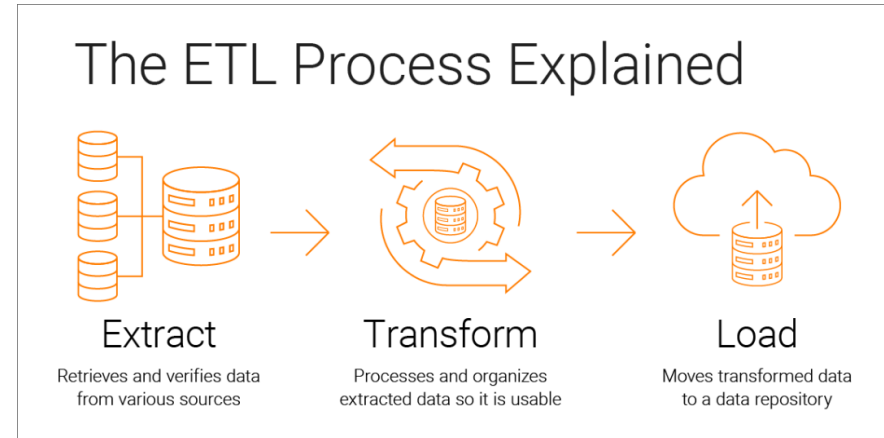
- niskie opóźnienia bo dane są brane w locie, nie z bazy





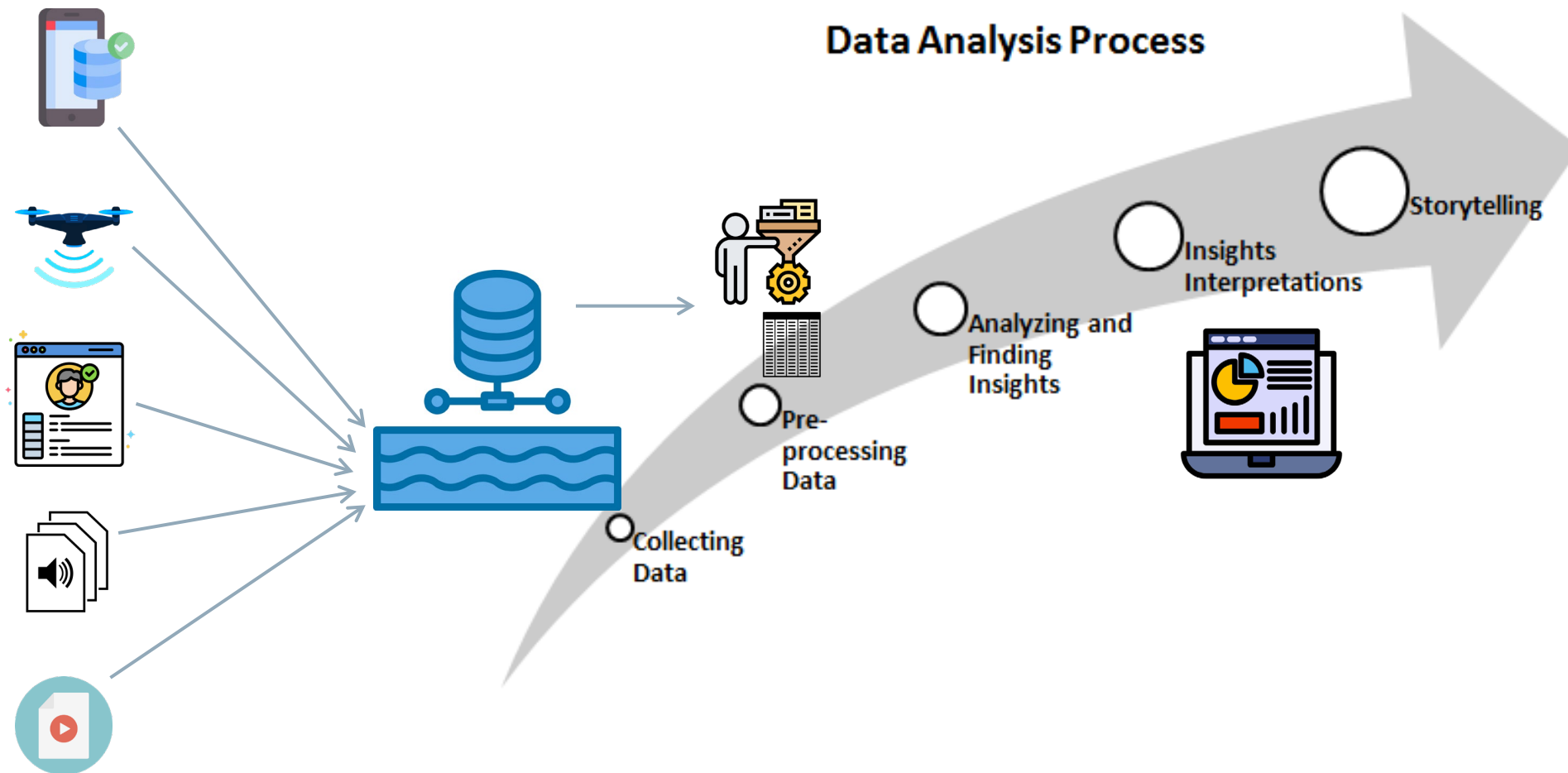
Procesy

Proces ETL



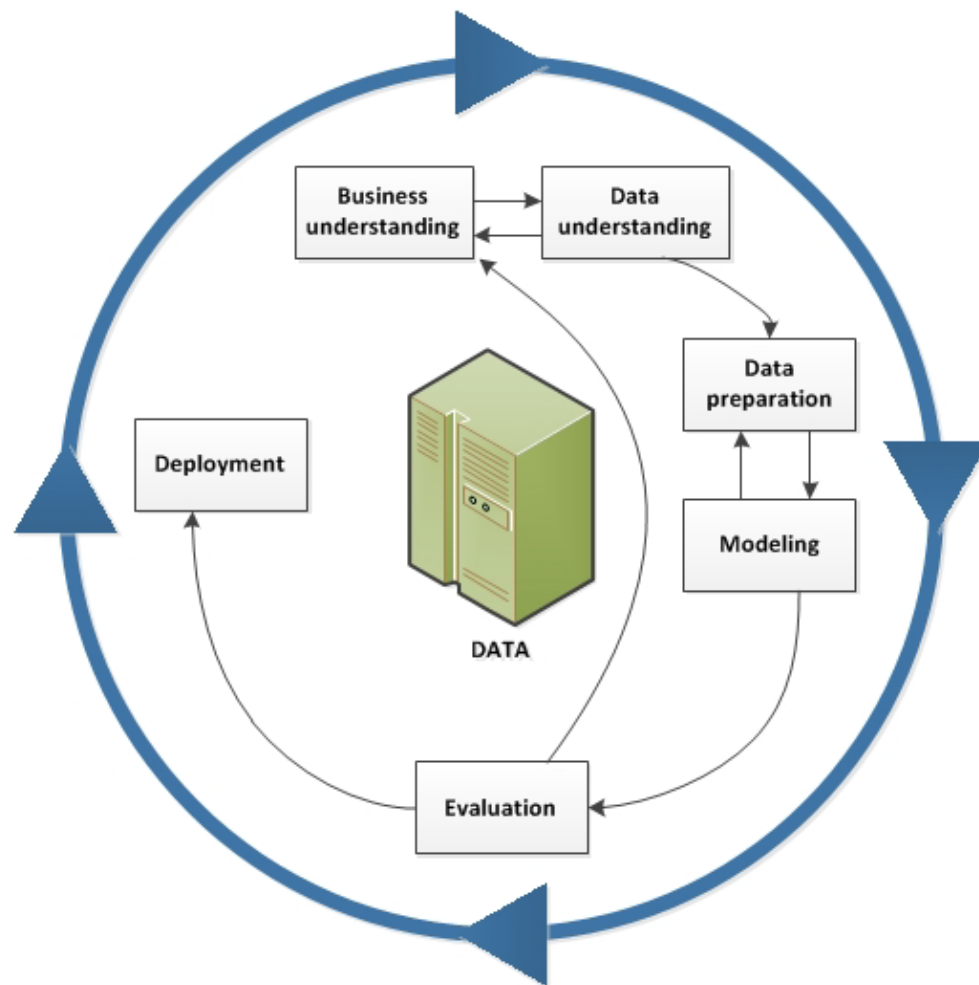
Magazyny danych

Standardowy proces analizy danych



Międzybranżowy proces drążenia danych CRISP-DM

ang. ***C**Ross-**I**ndy**S**try **P**rocess for **D**ata **M**ining*



The
End.