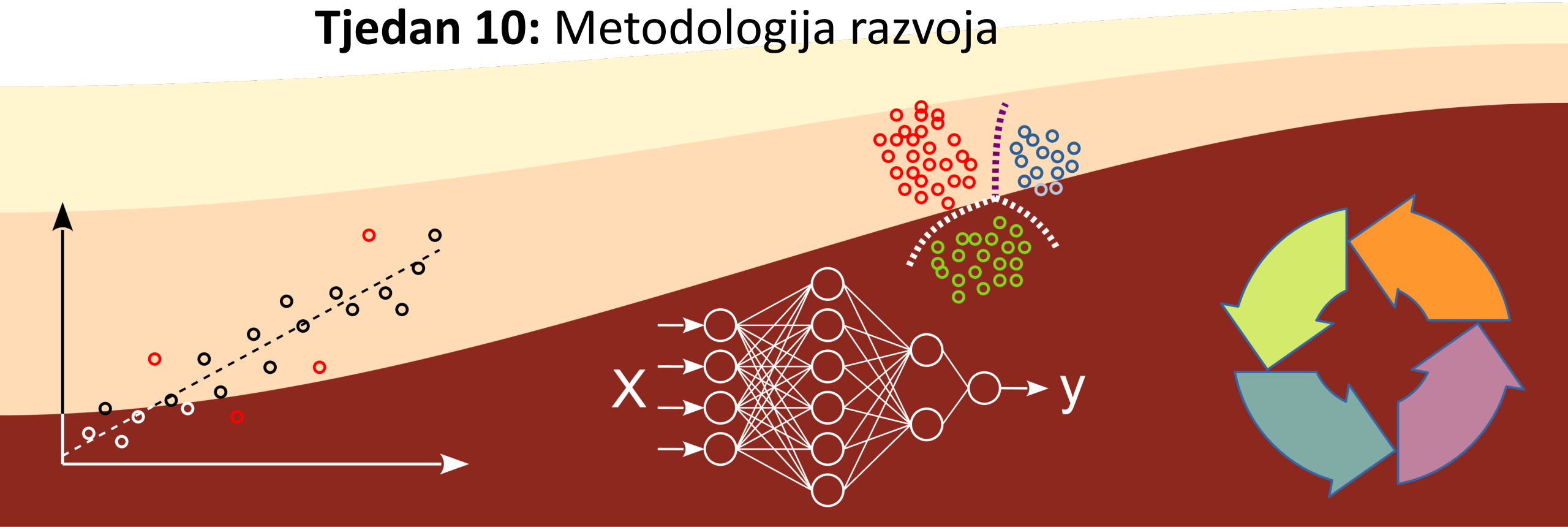
Arhitektura i Razvoj Inteligentnih Sustava

Tjedan 10: Metodologija razvoja



Creative Commons

- slobodno smijete:
  - dijeliti — umnožavati, distribuirati i javnosti priopćavati djelo
  - prerađivati djelo
- pod sljedećim uvjetima:
  - imenovanje: morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
  - nekomercijalno: ovo djelo ne smijete koristiti u komercijalne svrhe.
  - dijeli pod istim uvjetima: ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.

Općenito

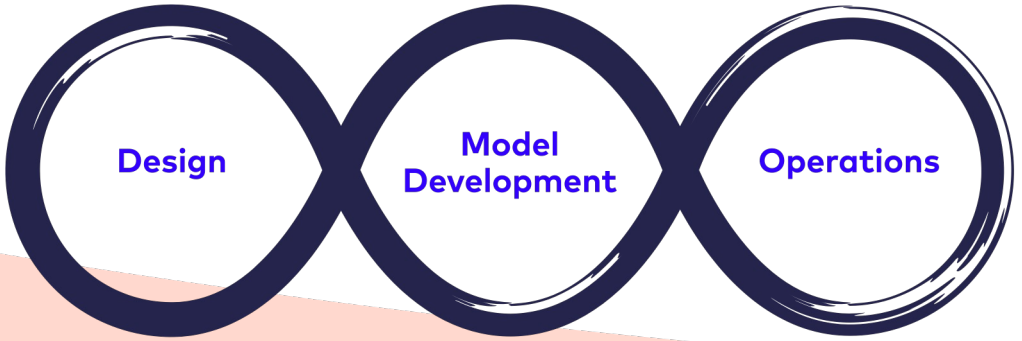
- Metodologija je skup propisa i dobrih praksi
- Mi se fokusiramo na metodologije za dizajn i razvoj informacijskih sustava – u našem slučaju inteligentnih sustava
- Postoje razni tipovi metodologija za dizajn i razvoj inf. sustava
  - Klasične – *Waterfall*
  - Iterativne – *Unified Process*
  - Agilne – *eXtreme Programming*, *scrum, kanban, DevOps (?)*
  - Disciplinirana agilna – kombinacija iterativne metodologije s agilnom
    - Osnovne faze se preuzimaju iz iterativnih metodologija – da, ne preskačemo osnovni dizajn inf. sustava !!
    - Unutar faza/iteracija imamo brze sprintove sve do dok ne dostignemo dovoljnu razinu zadovoljstva klijenta za tu iteraciju

Općenito

- Metodologije često definiraju
  - Sudionike u procesu dizajniranja i razvoja
  - Faze i aktivnosti – na visokom nivou faze, na niskom nivou pojedini zadaci
  - Isporuke – što se traži kao ulaz u neku aktivnost, a što se isporučuje
- Metodologije u sebi sadrže ukupno iskustvo s različitih projekata
  - Recimo *Unified Process* je izrazito glomazna i opsežna metodologija
  - Ne moramo na svim projektima raditi sve i svašta
  - Metodologija se prilagodi našem projektu – njegovoj veličini i trajanju
  - Napravimo radionicu s klijentom u kojoj definiramo što i kako radimo – to se zove *methodology tailoring session*

Agilna metodologija + DevOps

- DevOps dodaje značajnu podršku agilnim metodologijama
- Proširuje standardne agilne metodologije s pakiranjem, instalacijom (*deployment*) i monitoringom
  - Sve se ujedinjuje u procese – cjevovode
  - Integracijski (CI – *continuous integration*) i instalacijski/monitoring (*CD – continuous delivery*) cjevovodi
- Na tim se principima temelji i MLOps – *Machine Learning Operations*

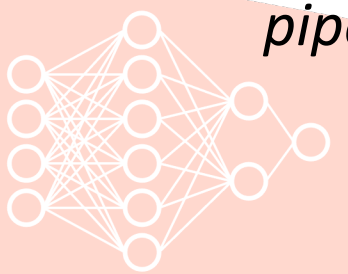


MLOps – Dizajn (1)

- Uloge: poslovni analitičar, IT arhitekt
- Koji je to poslovni problem koji pokušavamo riješiti ili poboljšati?
  - Koji je to slučaj korištenja?
  - Prikupljanje korisničkih zahtjeva: funkcionalni i nefunkcionalni
- Utvrđivanje trenutnog stanja
  - Utvrđivanje okoline: arhitektura i dizajn postojećih informacijskih sustava
  - Utvrđivanje izvora podataka: baze podataka, sheme, podatkovni modeli
- Dizajn
  - Servisni model: sučelja inteligentnih servisa
  - Podatkovni model: logički i fizički modeli
  - Infrastruktura: dizajn infrastrukture uz osvrt na uklapanje u postojeću okolinu
  - Ažuriranje procesa: dizajn cjevovoda

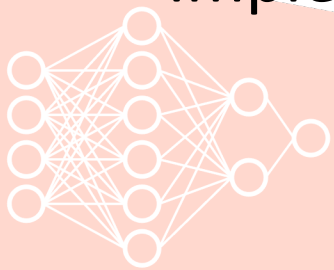
MLOps – Izrada modela (1)

- Uloga: podatkovni inženjer (*data engineer*, *data scientist*)
- Analiza ulaznih podataka
  - Analiza postojećih shema i modela podataka
  - Selekcija značajki
  - Označavanje podataka (*labeling*) – često je potrebna domenska ekspertiza – sjetimo se medicinskih slučajeva korištenja
- Implementacija postupaka obrade podataka
  - Izrada međuspremnika podataka
  - Implementacija postupaka za čišćenje, selekciju značajki i transformaciju podataka
  - Implementacija podatkovnog dijela cjevovoda za učenje (*data engineering pipeline*)



MLOps – Izrada modela (2)

- Uloga: inženjer za strojno učenje (*ML engineer*)
- Analiza prostora značajki
- Komunikacija s podatkovnim inženjerom oko obrade podataka
- Selekcija algoritma za strojno učenje
- Izrada modela
  - Učenje modela
  - Evaluacija i testiranje modela
  - Izrada postupaka za pakiranje modela i popratnih kodova
- Implementacija dijela cjevovoda za strojno učenje



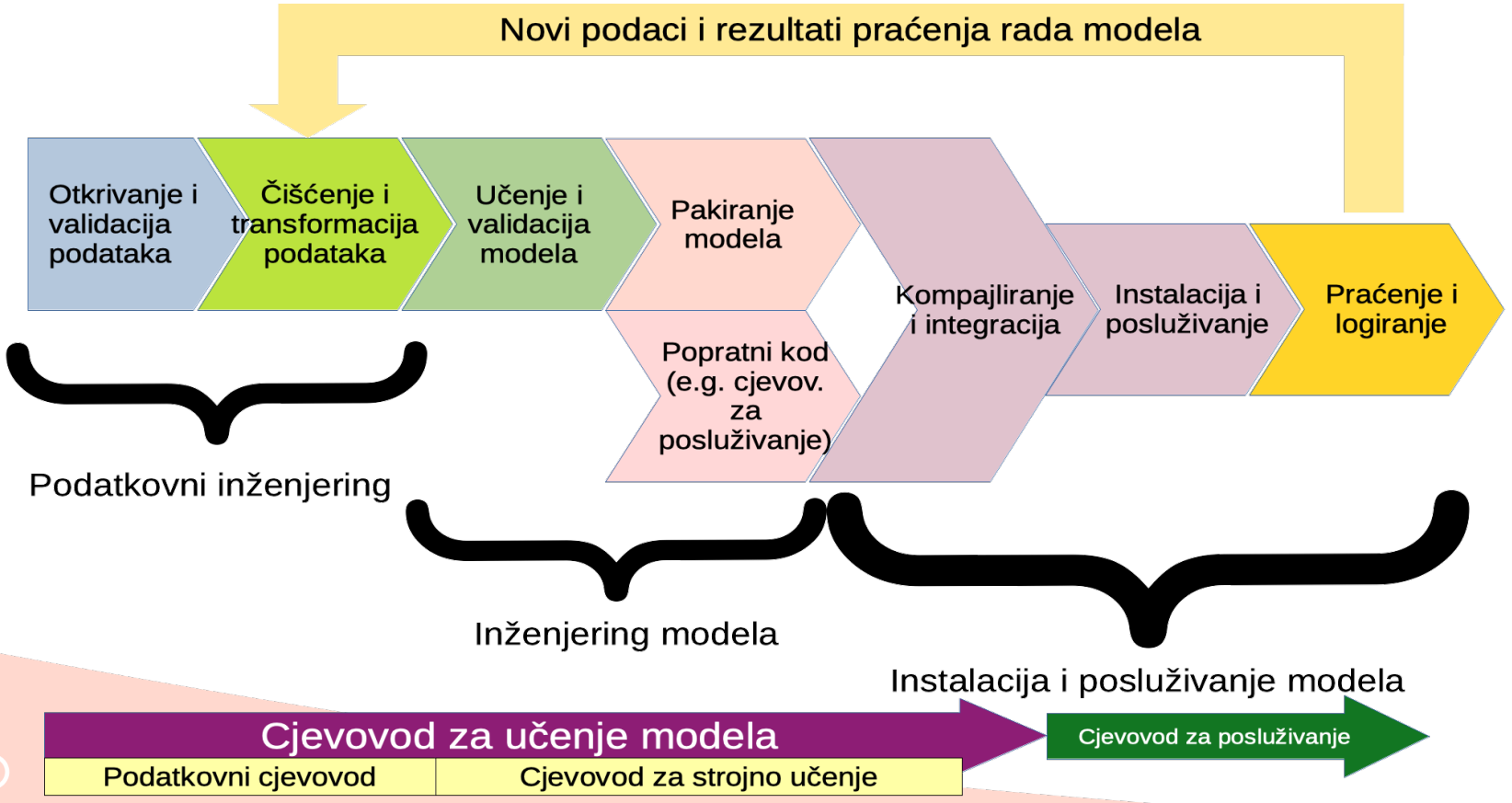
MLOps – Instalacija i posluživanje modela

- Uloga: inženjer za infrastrukturu i DevOps
  - Izrada postupaka za instalaciju modela u cjevovodu za učenje modela
  - Izrada cjevovoda za posluživanje
  - Servisna integracija sustava
    - Praćenje rada modela i podatkovne povratne veze

- Ne zaboraviti da netko treba instalirati i konfigurirati infrastrukturu da bismo uopće mogli raditi

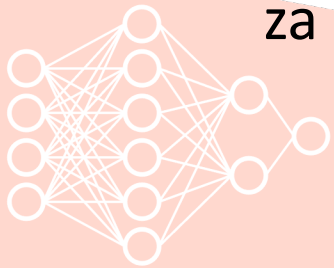
MLOps

- Osnovne faze MLOps-a se mogu sažeti u slijed aktivnosti



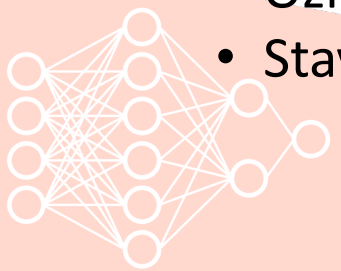
MLOps osnovne faze

1. Otkrivanje i validacija podataka
   - Gledamo izvore naših ulaznih podataka – to su često drugih inf. sustavi
   - Kakva je struktura podataka? Kakvi su sami podaci?
   - Pokušavamo vidjeti statistiku tih podataka
   - Selekcija značajki
2. Čišćenje i transformacija podataka
   - Implementiramo postupak čišćenja i transformacije podataka
   - Takvi transformirani podaci mogu biti upisani u posebna skladišta podataka koja se zatim koriste za učenje, evaluaciju i testiranje modela
   - Podaci koji ulaze u ove postupke mogu doći i kao rezultat izvođenja cjevovoda za posluživanje modela (*serving pipeline*)



MLOps osnovne faze

3. Učenje i validacija modela
   - Pročišćene podatke dijelimo na podskupove za učenje, evaluacija i testiranje našeg modela
   - Učenje modela – sjetite se svih detalja o paralelizaciji tog učenja
   - Evaluacija, testiranje – ukoliko nemamo dobre rezultate, tada imamo nekoliko opcija
     - Promjena parametara, algoritma, ...
4. Pakiranje modela
   - Jednom kad smo zadovoljni rezultatom, pakiramo model (*.pkl za scikit-learn, *.pth za PyTorch) i sve popratne modele – model za objašnjenje odluka (*explainer*), *drift* detektori, transformacijski modeli *(scaler)*
   - Dodajemo opisnike sučelja, potrebne biblioteke i module, instalacijske opisnike za poslužitelj modela koji koristimo
   - Označavamo verzije
   - Stavljamo sve skupa u repozitorij modela koji koristimo (AWS – s3 – minIO, GS, ...)



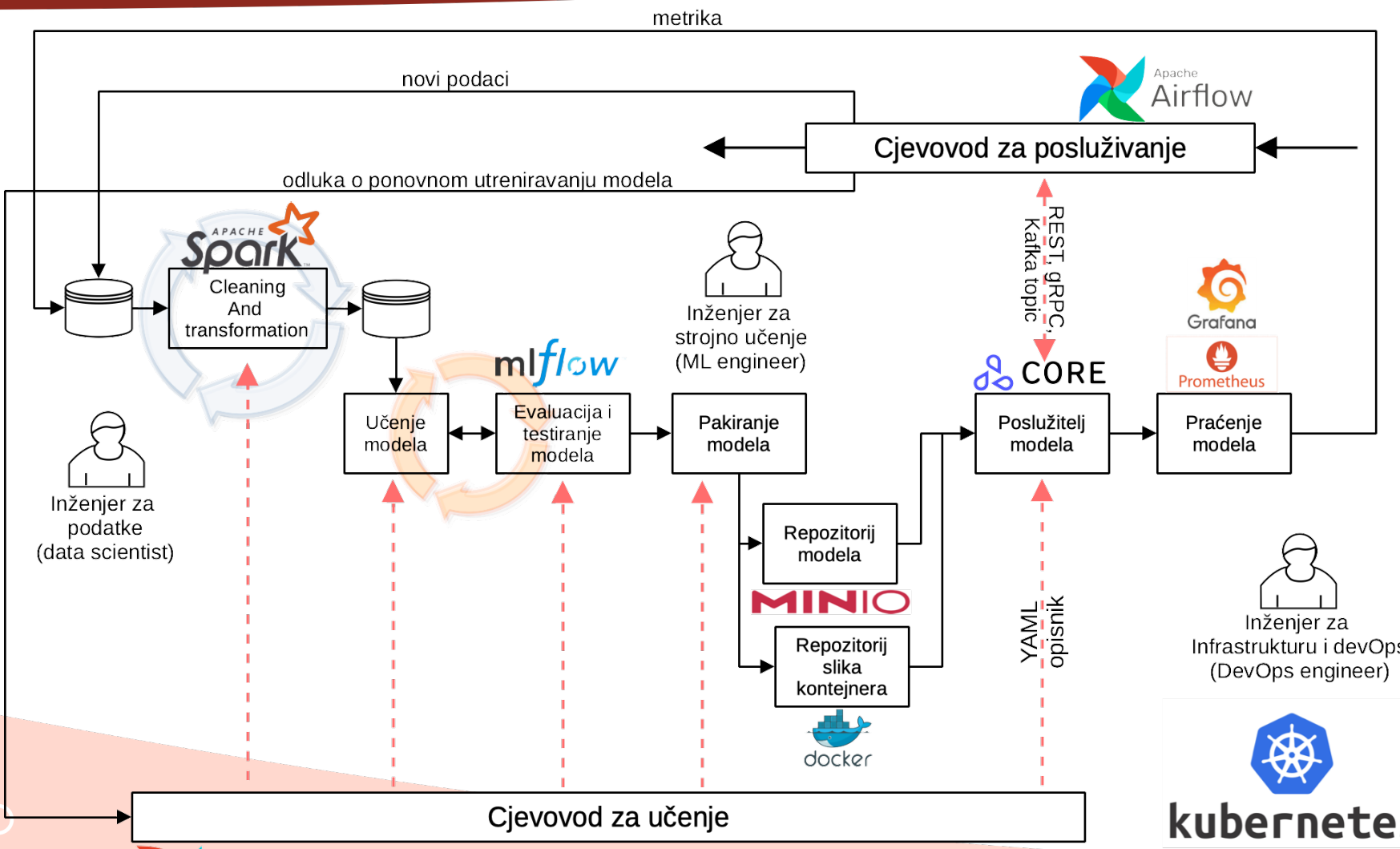
MLOps osnovne faze

5. Popratni kod
   - Recimo sve promjene na cjevovodu za posluživanje
6. Prevođenje (*build*) i integracija
7. Instalacija na poslužitelj(e) (*deployment*)
   - Određenim postupkom sve pakete instaliramo na odabrane poslužitelje
   - Model instaliramo na ML poslužitelj
     - Ujedno instaliramo i lokalni cjevovod koji povezuje sve modele u jednu cjelinu – sjetimo se da se popratni modeli mogu ulančavati s osnovnim modelom (*inference*)
   - Cjevovod za posluživanje (*serving pipeline*)
     - Recimo instaliramo novu verziju cjevovoda za posluživanje na Apache Airflow – python DAG

MLOps osnovne faze

7. Posluživanje modela
   - Vanjski sustavi pozivaju cjevovod za posluživanje
   - Taj cjevovod zatim poziva servis kroz koji je model izložen – REST, gRCP, tokovi (Kafka)
   - Unutarnji cjevovodi ulančavaju modele i vraćaju rezultate
   - Informacijski sustavi koji su pozvali model zatim još dodatno obrade te rezultate
     - Svi ti rezultati imaju za posljedicu generiranje novih podataka koji se zatim mogu koristiti za ponovno učenje modela – ponovno pokretanje cjevovoda za učenje
8. Praćenje modela
   - Pratimo funkcijske rezultate rada modela
     - Rezultati modela za objašnjenje odluka i drift detektora
     - Pratimo podatke koji su posljedica rezultata rad modela
     - U trenutku kada model nije precizan i ne radi dobro – želimo ga ponovno naučiti korištenjem novih podataka
   - Pratimo nefunkcijske rezultate rada modela
     - Ako performanse nisu zadovoljavajuće
       - Promjena složenosti modela i prostora značajki
       - Promjena na infrastrukturi

Open source *stack* za MLOps



MLOps *stack* – Obrada podataka

- Apache Spark *cluster*
  - Skalabilnost – mogućnost proširivanja
  - map/reduce način rada
  - Instalacija ETL postupaka na cluster
  - Asinkrono ili vremensko pokretanje postupaka
  - Pozivanje postupka iz cjevovoda za učenje (podatkovni dio)

MLOps *stack* – Izrada modela

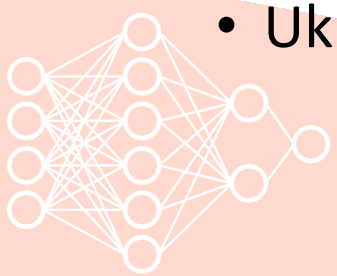
- MLflow
  - Repozitorij eksperimenata i modela
  - Modeli se čuvaju u repozitoriju artefakata (recimo minIO)
  - Suradnja između inženjera za strojno učenje na jednom projektu i problemu
  - Metrika i automatska evaluacija modela
  - Opisnici za ovisne biblioteke i module, te sučelje modela

MLOps *stack* – Instalacija i posluživanje modela

- minIO
  - s3 repozitorij modela i artefakata
  - Mjesto gdje poslužitelji modela mogu dohvatiti model i njegove opisnike
- DockerHub
  - Za sve prilagođene (*custom*) modele koji se instaliraju zajedno s serverskim bibliotekama
- Seldon MLServer
  - Jednostavni tanki server koji nudi samo višedretveni pristup za skalabilnost
  - Nema sučelja za upravljanje serverom
  - Može se upakirati u *docker* kontejner i staviti na DockerHub

MLOps *stack* – Instalacija i posluživanje modela

- Seldon Core (*open source*)
  - Server za posluživanje modela za strojno učenje koji se instalira u Kubernetes cluster
  - Skalabilan i elastičan
  - Jednostavno komandno sučelje (CLI) za upravljanje modelom – YAML
  - Cjevovodi za ulančavanje modela – kroz Kafku
  - Modeli izloženi kroz REST, gRPC i Kafka sučelja
  - Ima servis za metriku – izlaže i metriku Kubernetes clustera vezano uz posluživanje modela
- Seldon Deploy (*commercial*)
  - Uključuje sučelja za upravljanje serverom – komandno (CLI) i web



MLOps *stack* – Praćenje rada modela

- Poslužitelji modela izlažu servise za metriku prema open standardima
- Prometheus
  - Baza vremenskih serija (*time-series database*)
  - Spaja se na servis za metriku na poslužitelju za modele
  - Uzima definirane metrike i sprema ih u svoju bazu
  - Sučelje koje omogućava upite nad metrikama
- Grafana
  - Vizualna nadzorna ploča (*dashboard*)
  - Spaja se na Prometheus kao izvor podataka

MLOps *stack* – Cjevovodi

- Apache Airflow
  - Procesna orijentacija
  - DAG-ovi koji predstavljaju procese spremaju se u repozitorij od kud ih Apache Airflow dohvaća
  - Korisničko sučelje koje omogućava upravljanje procesima

- Infrastruktura – Kubernetes cluster
  - Skalabilan
  - Lako se nadograđuje
  - Ima jednostavno komandno sučelje – iz čega se mogu raditi vrhunske instalacijske i konfiguracijske skripte – doslovno isprogramirate postavljanje okoline i instalaciju modela