

Projektidé

Modulärisering av AI-modell

(AI-toolchain)

*Mattias Festin*

*Emelie Chandni Jutvik*

*2024-05-28*

Syfte [3](#_heading=h.30j0zll)

Nytta [3](#_heading=h.1fob9te)

Värde [3](#_heading=h.3znysh7)

Befintlig kodbas [4](#_heading=h.2et92p0)

Paketering [6](#_heading=h.tyjcwt)

**Syfte**

* + Utveckla ett verktyg (toolchain) som förenklar inlärningsprocess och arbetsflöde för att implementera AI-modeller i Decerno:s projekt.
  + Kunna erbjuda en specialistlösning som skiljer Decerno från andra aktörer på marknaden.
  + Kunna erbjuda anställda på Decerno att vara delaktig i utvecklingen av en teknikmodul som ligger i framkant.
  + Verktyget fungerar som kvalitetssäkring av framtagen AI-modell då utvecklingsstegen i toolchain:en ställer krav på arbetsflödet samt data.

**Nytta**

* + Fler utvecklare inom Decerno får kunskap om AI
  + Fler utvecklare inom Decerno kan implementera AI-modeller i sina projekt
  + Utvecklingstiden för att implementera AI minskas
  + Möjligheten till att bygga kundspecifika lösningar
  + Möjligheten till att kapsla in lösningen och kunna säkerställa att kundens data inte lämnar deras system

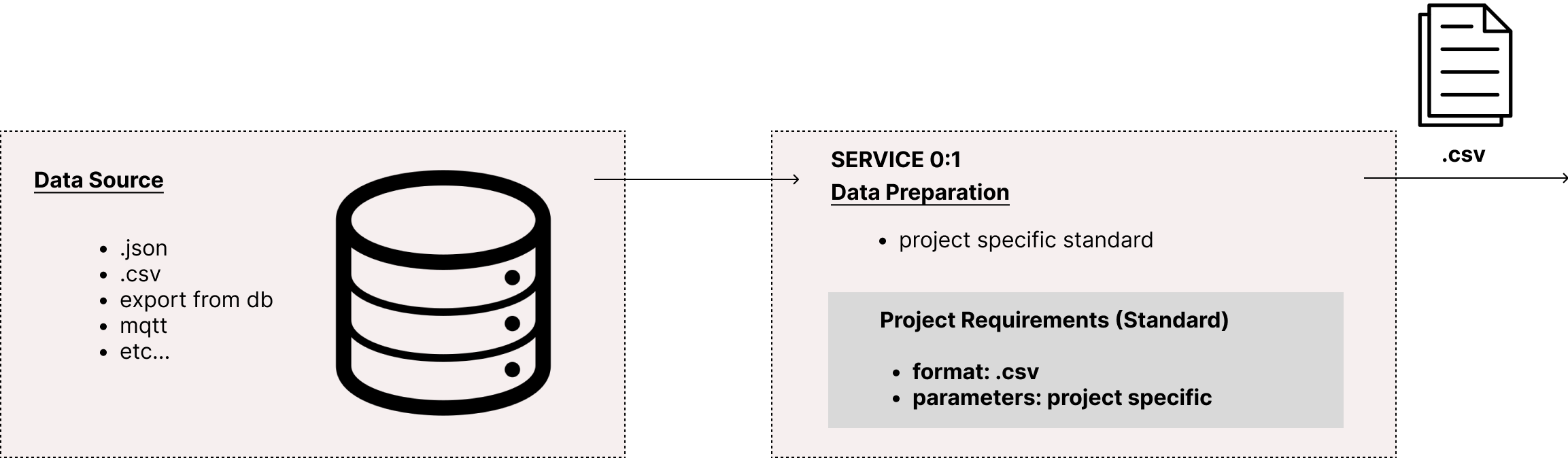
**Värde**

* + En modul/toolchain som kan tillämpas inom flera projekt möjliggör att kostnader för vidareutveckling och förädling kan fördelas mellan projekten
  + Produkten kan användas som säljargument och visa upp Decerno som ett företag som utvecklat en unik toolchain som förenklar vägen från idé till realisering av AI-modell

**Befintlig kodbas**

## Modul 1

Hanterar olika typer av indata-format och levererar en csv-fil med relevant data för beräkningen.



## Modul 2

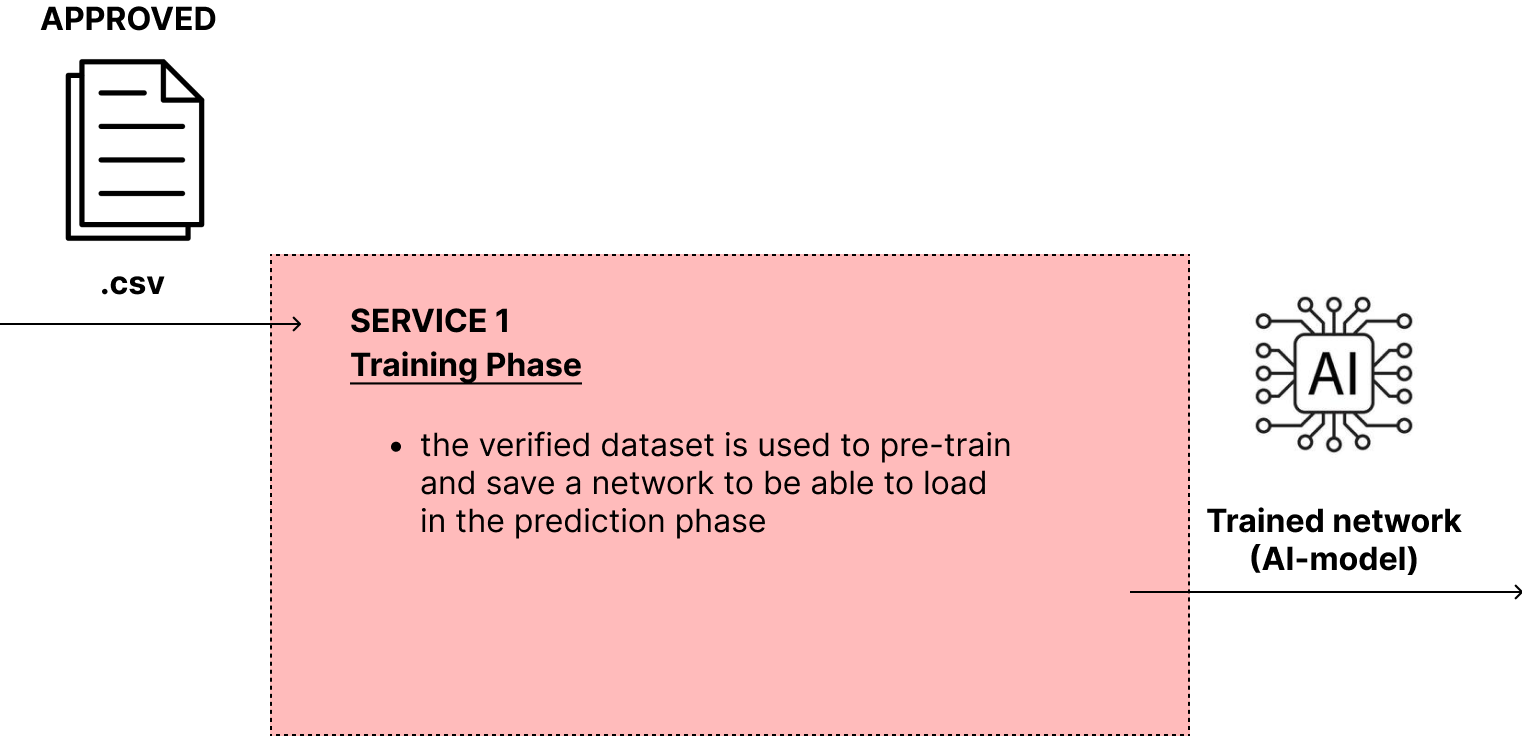
Läser csv-filen från modul 1 och förbereder denna data för att kunna läsas av ett neuralt nätverk. Den förberedda datan och nätverket genomgår ett valideringstest:

* + - Godkänd validering —> data och modell är redo för att skickas till modul 3
    - Icke godkänd validering —> modellen behöver justeras och körs genom valideringstest tills dess att önskat resultat uppnås



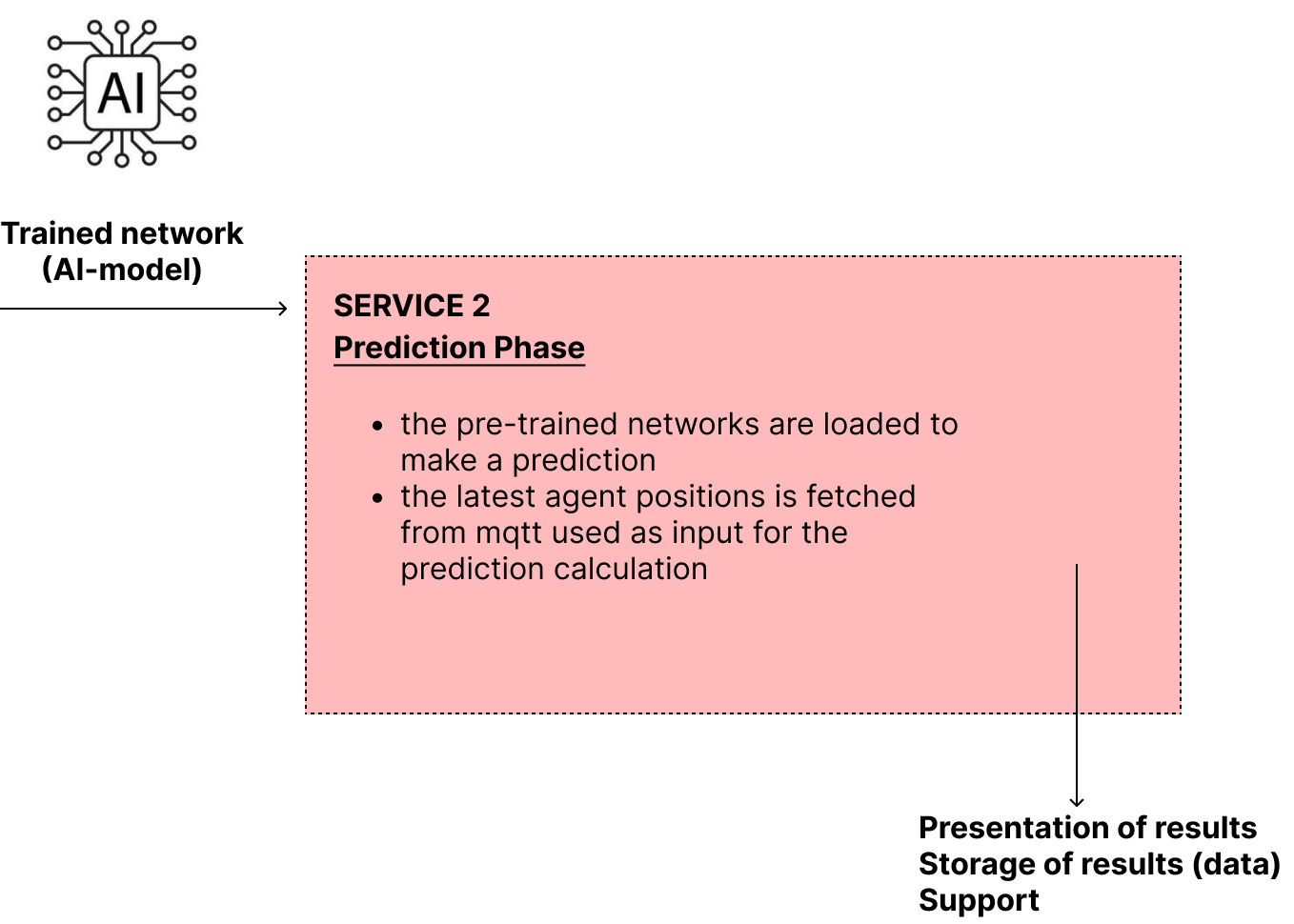
## Modul 3

Det neurala nätverket tränas och sparas.



## Modul 4

Det sparade/tränade neurala nätverket (från modul 3) nyttjas för att göra en prediktion. Resultatet skickas i form av rådata. Presentationen av resultatet och jämförelsen med verklig data görs lokalt. Tanken är att denna data kan skickas till ett lagringsutrymme som vidare kan läsas in i BI-rapport, webb-applikation, o. dyl.



**Paketering**

## MVP (Minimum value product)

* + Hantering av olika typer av indata-filer (.json, .csv till att börja med)
  + Användargränssnitt som hanterar:
    - * Uppladdning av indata-fil
      * Val av neuralt nätverk
      * Specifikation av in-parametrar som ska användas för beräkningen (string)
      * Specifikation av ut-parameter (string)
      * Inställning av hyper-parametrar
      * Presentation av verifieringstest (plot)
      * Möjligheten att starta träning av ett nätverk
      * Output —> tränad modell

## Vidareutveckling

* + Utveckla ett prediktionssteg (modul 4) där vi automatiserar flödet för att hämta indata och göra en prediktion i realtid
  + Stegvis automatisering av manuella funktioner som sker via användargränssnittet i MVP:
    - * Val av neuralt nätverk
      * Val av in-parametrar till modellen
      * Hyperparameter-justering
      * Verifiering av modellen
      * Start av träningsfasen
  + Utveckla ett test där modellen själv kan avgöra minsta möjliga datamängd som krävs för att göra en god nog prediktion
  + Implementation av flera olika typer av nätverk (idag befintliga lösningen finns ANN, RNN , Transformer)
  + Tillägg av hantering av flera olika datakällor (text, bilder, osv. I befintlig lösning hanteras endast indata i form av rådata.)