

Maskininlärning

Deadline: 31:a mars 17:00

Denna uppgift är en möjlighet för den studerande att redovisa att denna besitter förmågan att använda python i syfte av att utveckla program som använder maskininlärning.

Uppgiften är tredelad men redovisas som en uppgift.

1. Välja ett maskininlärningsproblem

Studerande har valt ett dataset som denna formulerat ett problem till. Tanken är att studerande ska försöka finna en algoritm som är välanpassad för att lösa problemet samt dokumentera problemet under pågående utveckling.

2. Optimera lösningen

Studerande förväntas utföra *någon* optimering på datasetet för att hypotetisk producera ett bättre svar. Formuleringen är hypotetisk då slumpen i somliga fall kan skapa ett sämre resultat även om optimeringen är välgrundad.

- Här bör det även framgå varför en optimering görs genom användandet av grafer. Exempelvis så kan ett linjärt beroende påvisas av ett scatterplot av två features.

3. Leverera lösningen via ett REST API

Studerande förväntas skapa ett *enklare* REST API där användare kan skicka in "ny" data och få fram en förutsägelse baserat på den maskininlärningen som gjordes i steg 1 och 2.

– *Viktigt* –

Ett "bra" resultat är relativt datasetet. Om sannolikheten att en människa kan göra en god chansningen utan användningen av en maskin så är tanken att maskininlärningen ska producera en bättre "chansning".

Form och bedömning

All bedömning sker på kvalité av lösning och dokumentation i koden. Dokumentation i koden förväntas vara på en nivå där den förklarar *varför* koden har valts som lösning för ett problem. Betoning på *varför* (**inte** hur) då koden förväntas vara självdokumenterad genom väl valda variabler och funktionsnamn. Exempel nedanför:

```
"<code that visualizes feature relationships with linear separability>"  
  
# Dataset:et består av oberoende variabler där de till största del går att separera linjärt  
"<code for linear regression>"
```

All kod skickas som en länk till ett github repo via LMZ:en. Studerande väljer själv i vilket format som inlämningen sker i, men förslagsvis så lämnas maskininlärningen **med** optimeringen in som en ipynb fil (notebook) och REST API:et som en py fil.

Övrigt

Då prognoserna i REST API:et skapas som resultat av en inlärd AI modell så kommer denna att behöva exporteras ut för att sedan användas i REST API:et. Som lösningsförslag till detta ges joblib av sklearn https://scikit-learn.org/stable/model_persistence.html.

Fusk och Plagiat

Medan det är okej att hämta inspiration och förklaringen från nätet så ska dessa ej användas in inlämnad uppgifter. Syftet med att dokumentera kod är för att studerande ska resonera kring kodens syfte och vad den är tänkt att lösa. I samband med dokumentationen förväntas studerande anpassa koden efter problemets karaktär och på så vis undvika plagiat.

Om koden ej anpassas eller endast mindre förändringar sker (exempelvis utbyte av variabelnamn) så bör en länk bifogas där det tydligt framgår vad som studerande har skapat och vad som är taget från en annan källa. Vidare så bör även studerande undersöka om källan tillåter kopierande av kod, exempelvis så är ett olisencerat github repo copyrightskyddade och bör ej användas i en lösning.

Inlämning och bedömning

Inlämning sker via awesome. Godkända format är en länk till ett github repo. Innan länken skickas in är det viktigt, och förväntas det, att studerande verifierar att länken är tillgänglig för bedömande lärare.

Ingen skriftlig inlämning, men kod förväntas vara tydligt strukturerad och dokumenterad med syftet att förklarar varför en lösning används.

Betygskriterier

Betyget på kursen sätts utifrån kodkvalité och dokumentation, vid behov kan ett handledningssamtal bokas in. Plagierad kod, inklusive AI genererad kod, kommer att leda till ett underkänt betyg och att UL uppmärksammas om misstänkt fusk.

Godkänt

Koden innehåller 3 ovannämnda delar med dokumentation som till större delar förklarar varför vald/skriven kod/algorithm används för att lösa tillhörande del. Någon graf används för att förklarar val av algorithm, alternativt optimering. Studerande lämnar även ett python script med ett REST API som använder vald ML-lösning.

Väl godkänt

Kriterier för Godkänt är uppfyllda samt att dokumentation är välanpassad och beskrivande för lösningen. Vidare ska även ett par grafer användas för att beskriva problemet och optimeringar som har utförts på dataset:et.