Gruppinlämning

En maskininlärningsapp

Python för maskininlärning
HT 2021

Introduktion

Ni kommer att arbeta i grupper om 3 eller 4 personer. Som grupp ska ni samarbeta och sätta ihop allt ni lärt er under kursens gång och skapa en maskininlärningsapplikation.

Målet med uppgiften är att ni ska leverera ett GitHub repository med väldokumenterad kod, som är lätt för en användare att köra. Varje grupp kommer att interagera med en maskininlärningsmodell via ett API, och ni ska använda ett bibliotek som heter streamlit för att kunna bygga appen och enkelt visualisera interaktionen med maskininlärningsmodellen. Data som användaren laddar upp ska även sparas i en enkel databas (SQLite).

GitHub repot kommer att lämnas in gruppvis. I era commits måste det synas att alla har varit med och arbetat på koden som ni har lämnat in. I slutet av projektet ska även varje gruppmedlem lämna in en rapport om gruppuppgiften.

Presentationer och code reviews

Den sista dagen på kursen kommer varje grupp att presentera sitt arbete. I presentationen ska det ingå en kort demo av hur appen fungerar, samt en snabb genomgång av vilka val som gjordes och motivering bakom dessa. Om ni stötte på oväntade problem får ni gärna ta upp dem och berätta hur ni löste dem – det är bra om de andra grupperna kan lära sig även av era misstag.

Varje grupp kommer att genomföra code review på en annan grupps kod och ge feedback på dokumentering, kommentering, clean coding och hur lätt det var att använda programmet. Denna code review kommer att genomföras efter varje grupps slutpresentation. Exempel: Grupp 1 presenterar sitt arbete. Grupp 4, som har gjort code review på Grupp 1, berättar om sin feedback när Grupp 1 har presenterat klart. Sedan presenterar Grupp 2 sitt arbete osv.

Varje grupp får max 15 minuter för presentation av sitt projekt, följt av max 10 minuters code review av en annan grupp.

Krav på programmet

För att ni ska få godkänt på inlämningsuppgiften måste ert program uppfylla följande krav:

- Det ska levereras i ett GitHub repository
    - Det ska finnas en tydlig README för hur man startar appen lokalt på sin dator
    - Detta innebär att filer som t ex requirements.txt är självklara
- Användaren ska kunna ladda upp egen data i appen
    - Den datan ska sedan sparas i databasen så att användaren kan gå tillbaka och välja samma data igen, utan att behöva ladda upp den igen
- Appen ska interagera med minst en maskininlärningsmodell via ett API
    - Vi kommer att tillhandahålla en server med instruktioner så att ni kan välja att interagera med 4 olika modeller
- Ni ska använda biblioteket Streamlit för att sätta upp en sida som användaren kan interagera via
- Ni ska använda docstrings, clean coding och i mån av behov även kommentarer i koden – kom ihåg att en annan grupp ska kunna förstå koden!
- **Tips**: innan ni lämnar in ert repo, klona ner det i en ny mapp, gör en helt ny conda environment och följ era egna instruktioner - se till att det går att köra!!

Krav på individuell rapport

I de individuella rapporterna, som ska vara mellan 1 till 3 sidor lång, vill vi att du ska ha med följande:

- Ett kort intro till vad din grupp gjorde för projekt och vilken/vilka ML-modeller ni interagerade med
- Ett UML-diagram över hur din grupps kod är strukturerad, som du ska förklara och motivera (motivera olika val av mappar, funktioner, design, klasser mm.). Du ska ha gjort ditt UML-diagram själv.
- En förklaring till hur ni satte upp er databas samt hur den fungerar. I denna förklaring ska vi kunna se att du förstår hur en relationsdatabas fungerar, så motivera din förklaring väl.
- En förklaring av hur ett RESTful API fungerar, hur ni använder det i er kod och varför ni gör de requests ni gör. I denna förklaring ska vi kunna se att du förstår hur ett RESTful API fungerar.
- Code review (se nedan) med en kort introduktion till vad den andra gruppens projekt var.

Krav på code review

En code review kommer i detta fall att bestå av följande steg:

- Klona ner repot och följa instruktionerna i README för att få det att fungera
- Gå igenom hur applikationen fungerar och testa den
- Gå igenom koden och försöka förstå hur den är uppbyggd och om det är intuitivt, till exempel kan du ta hänsyn till följande:
    - Har klasserna/funktionerna/variablerna tydliga namn?
    - Följa PEP8?
    - Är koden lätt att förstå? Är koden uppdelad i flera tydliga, mindre funktioner eller är det bara en lång, komplicerad kodklump?
    - Är det för mycket kommentarer? Är det för lite? Finns docstrings? Stämmer de överens med vad som står i koden?
    - Finns det mycket bortkommenterad kod eller kod som aldrig kallas på?

Vi vill att du i din individuella rapport ska reflektera över följande punkter:

- Var det lätt att starta repot? Skulle något kunna göras bättre eller tydligare i README? Hur?
- När du gick igenom koden, kunde du förstå ungefär hur skaparna hade tänkt? Hade du lätt att titta igenom koden och se och förstå ungefär hur saker skedde? Vad hade du tyckt borde vara annorlunda, om något? Vad var svårt att förstå och vad var övertydligt?
- Hur har den andra gruppen tänkt när de byggt sitt repo jämfört med din grupp? Har ni tagit liknande approaches för att lösa olika problem eller har den andra gruppen tänkt helt annorlunda? Vad tror du det beror på och vad tror du är rätt? Finns det tydliga rätt/fel?
- Vad har den andra gruppen gjort på ett bra sätt? Gjorde din grupp likadant? Varför/Varför inte?

Vi vill att ni i er code review tänker på följande:

- Arbeta med konstruktiv kritik – ni är inte här för att såga någon vid fotknölarna! Detta innebär även att beröm är välkommet.
- Ge generell feedback, gå inte in på detaljnivå (”Filen main.py kunde haft mer kommentarer för att förtydliga alla steg” istället för ”på rad 23 i utils.py har ni ett konstigt variabelnamn”)
- Se om gruppen har följt konventioner när det gäller namngivning
- Se om gruppen har en tydlig filstruktur och ge feedback på detta samt deras README

Bedömning består av

När ni har lämnat in hela projektet och presenterat kommer er bedömning att bestå av en sammanvägd bedömning av samtliga delmoment i grupparbetet, det vill säga gruppens repository, presentationen, den individuella rapporten samt er code review. Observera att betyget VG endast kan bedömas från den individuella rapporten.