Laboration maskininlärning AI22

Syftet med den här labben är att använda verktygen ni lärt er i maskininlärningen till att få kännedom och tillämpa på olika slags problem som ni kan stöta på i näringslivet.

Notera att de algoritmerna och modellerna vi bygger upp här kommer vara enkla och i näringslivet är det inte ovanligt att man kombinerar flertals modeller i sina lösningar. Poängen med den här labben är att få förståelse av hur man kan angripa olika slags problem mha maskininlärning.

Delmoment

Den här labben är uppdelad i två delmoment:

1. Rekommender system (obligatorisk)
2. Disease prediction (bonus)

För frågor där ni behöver skriva text:

Korta frågor kan ni besvara med hjälp av en kommentar eller i markdown om ni kör jupyter notebook. Kör ni i Pythonskript, skriv en markdown vid sidan av och besvara på frågorna.

---

1. Recommender system

När du tittar på Youtube, beställer mat online, köper böcker online, lyssnar på Spotify, använder LinkedIn så får du ständigt rekommendationer för nya videoklipp, maträtter mm. Det som ligger bakom dessa är en typ av recommender system.

1.0 - Uppvärmning

Börja med att kolla på denna youtube-video och följ efter i kod för att skapa ett enkelt recommender system för filmer med hjälp av KNN. Datasetet som används i videon är från movielens small som består av 100,000 ratings på 9000 filmer och 600 användare.

Lämna inte in denna delen, utan gör den för egen skull.

1.1 - Movielens full - EDA

Nu ska du ladda ned datasetet **ml-latest.zip** under sektionen "recommended for education and development". Läs in dataseten "movies.csv" och "ratings.csv" besvara på följande frågorna nedan.

a) Gör en EDA för att förstå datasetet. Inkludera olika slags plots. Begränsa dig inte till frågorna nedan, utan försök undersöka fler aspekter av datan.

b) Vilka är de 10 filmerna med flest ratings?

c) Beräkna den genomsnittliga ratingen för dessa 10 filmerna med flest ratings.

d) Gör en plot över årtal och antalet filmer representerade i datasetet.

e) Gör en plot över antalet ratings mot movieId.

f) Beräkna genomsnittliga ratings för de top 10 filmerna med flest ratings. Gör ett stapeldiagram över dessa.

1.2 Skapa gles matris

Likt i videon i uppgift 1.0 skapade du en pivottabell av dataframet med index: "movieId", columns: "userId" och values: "ratings". Denna pivottabell är dock "dyr" att skapa och förmodligen kommer inte din dator att klara av skapa den om du inte filtrerar bort viss data. Fundera ut ett lämpligt sätt att filtrera ditt dataset, pröva dig fram och motivera.

Skapa en gles (sparse) matris av denna pivottabell mha `scipy.sparse.csc_matrix()`. Vill du använda dig av all data går det också att lösa, men du behöver lösa hur du skapar den glesa matrisen utan pandas pivot-tabell.

1.3 Recommender system

Skapa ett recommender system med KNN och låt systemet ta input från användaren och skriva ut top 5 rekommenderade filmerna, baserat på användarens sökquery. Observera att det finns ett **logiskt fel** i videon som gör att rekommendationerna inte blir så bra, försök hitta felet och åtgärda det.

a) Beskriv med ord hur ditt system fungerar.

b) Leta online och läs vidare om rekommenderarsystem och beskriv kort hur dem fungerar. Glöm inte källhänvisa.

---

2. Disease prediction (bonus)

I det här momentet kommer vi jobba med ett dataset med data för hjärt-kärlsjukdom. Börja med att ladda ned datasetet från Kaggle och läs på vad de olika features betyder. Notera att detta dataset innehåller många felaktigheter, exempelvis finns negativa blodtryck och blodtryck som är omöjligt höga.

I uppgifterna nedan är ett arbetssätt för data scientist för att undersöka, rengöra data och testa olika modeller.

2.0 - EDA uppvärmning

Använd pandas, matplotlib och seaborn för att besvara på följande frågor för datasetet:

a) Hur många är positiva för hjärt-kärlsjukdom och hur många är negativa?

b) Hur stor andel har normala, över normala och långt över normala kolesterolvärden? Rita ett tårtdiagram.

c) Hur ser åldersfördelningen ut? Rita ett histogram.

d) Hur stor andel röker?

e) Hur ser viktfördelningen ut? Rita lämpligt diagram.

f) Hur ser längdfördelningen ut? Rita lämpligt diagram.

g) Hur stor andel av kvinnor respektive män har hjärt-kärlsjukdom? Rita lämpligt diagram

---

2.1.0 - Feature engineering BMI

Skapa en feature för BMI (Body Mass Index), läs på om formeln på wikipedia.

a) Släng de samples med orimliga BMIer och outliers. Notera att detta kan vara svårt att avgöra i vilket range av BMIer som vi ska spara. Beskriv hur du kommer fram till gränserna.

b) Skapa en kategorisk BMI-feature med kategorierna: normal range, overweight, obese (class I), obese (class II), obese (class III).

2.1.1 - Feature engineering blodtryck

Släng bort samples med orimliga blodtryck och outliers. Likt uppgift 2.1.0 är det inte trivialt att sätta gränserna. Skapa en feature för blodtryckskategorier enligt tabellen i denna artikel. Beskriv hur du kommer fram till gränserna.

2.2.0 - Visualisera andel sjukdomar

Skapa barplots med en feature mot andelen positiva för hjärt-kärl sjukdom. Exempelvis blodtryckskategorier mot andel positiva, BMI kategori mot andel positiva mm. Gör dessa plots i en figur med flera subplots.

2.2.1 - Visualisera korrelation

Skapa en heatmap av korrelationer och se om du hittar features som är starkt korrelerade, dvs nära 1 eller features som är starkt negativt korrelerade, dvs nära -1. Kan du förklara varför de kan vara korrelerade?

2.2.2 - Ta bort korrelerade features

Ta bort de features som bidrar till extra korrelationen, ex om 2 features är nära korrelerade så ta bort 1 av dem.

2.3 - Välja modell

Välj 3-5 maskininlärningsmodeller, gärna så olika som möjligt och gör följande:

- train|validation|test split
- skala datasetet med feature standardization eller normalization
- definiera hyperparametrar (param_grids) att testa för varje modell
- använda `GridSearchCV()` och välja lämplig evalueringsmetric
- gör prediction på valideringsdata

- beräkna och spara evaluation score för ditt valda metric
- checka bästa parametrarna för respektive modell

2.4 Ensemble

Använd `VotingClassifier()` på datasetet som du valt och lägg in de bästa parametrarna för respektive modell.

2.5 Evalueringar

Gör confusion matrices och classification reports för 2.4 och 2.5.

2.6 "Deploy" - spara modell

Börja med att plocka ut 100 slumpmässigt valda rader från ditt dataset. Exportera dessa 100 samples i **test_samples.csv**. Därefter tar du den bästa modellen och träna på all data vi har förutom de 100 datapunkterna du plockade ut. Spara därefter modellen i en .pkl-fil med hjälp av `joblib.dump()`. För modellen kan du behöva använda argumentet compress för att komprimera om filstorleken för stor.

2.7 Ladda modellen

Skapa ett nytt skript: **production_model.py**, ladda in **test_samples.csv** och din modell. Använd `joblib.load()` för att ladda in en .pkl-fil. Gör prediction på de 100 datapunkterna och exportera en fil "prediction.csv" som ska innehålla kolumnerna med ifyllda värden:

- probability class 0
- probability class 1
- prediction

Bedömning

Om du har fått någon kodsnutt från någon annan, LLM som ChatGPT eller hittat i någon sida är det **viktigt** att du källhänvisar. Skriv en kommentar bredvid koden som du har tagit.

Godkänt

- löst uppgift 1 på ett korrekt sätt
- koden är kommenterad med relevanta kommentarer
- har beskrivit och motiverat val av parametrar, modeller mm
- dataanalysen och datavisualiseringen är gjord på korrekt sätt
- variabelnamnen är bra valda
- gjort flera relevanta git commits

Väl Godkänt

Uppfyllt allt för godkänt samt:

- löst båda uppgifterna
- koden är tydlig och enkel att följa
- koden är välstrukturerad med funktioner och/eller OOP
- dataanalysen och datavisualiseringen är väl genomtänkt
- du motiverar koden väl med datavetenskapligt korrekt språk
- du motiverar väl dina val av parametrar, modeller mm