INFO 180 – Introduksjon til kunstig intelligens

Oblig-oppgåve 2 – 30.sep – 4.okt 2024 – Maskinlæring

OBLIGATORISK! Denne oppgåva må gjennomførast og godkjennast for å få gå opp til eksamen.

Innleveringsfrist på Mitt UiB: 17. oktober 14:00

Maskinlæring er eit stort felt innan kunstig intelligens, og er eit felt som har massevis av ressursar på veven. Det fins òg programpakkar som har ferdig implementerte versjonar av maskinlærings-algoritmar. Algoritmane er altså ferdig implementert, ein treng berre å tilpasse data, ein del viktige læringsparametrar og køyre algoritmane og evaluere dei.

I denne oppgåva skal de jobbe med eit datasett med eigenskapar for studentar ved UiB og i kor stor grad dei vil vere ok å ha med på fest. Datasettet er tilgjengeleg på Mitt UiB. Ein kan tenke seg at dataa er samla inn av organisasjonen NoPartyKillers som registrerer på veven erfaringar folk har hatt med ulike festdeltakarar (sjølvsagt heilt ulovleg, men …). De skal eigentleg sjekke kor gode tre maskinlærings-algoritmar er på dette datasettet. Oppgåva vert denne gongen i stor grad å finne ut av maskinlærings-verktøy og -metodar ved å bruke ressursar på veven.

Datasettet har følgande kolonnar:

- Gender: male eller female
- Age: < 20, 20-24, 24-30, > 30
- Study: socsci, mathsci, med, hum – kva fakultet studenten kjem frå
- Activity: favorittfritidsaktivitet blant 5 mulige: outdoor life, gaming, sport, music, cooking.
- Music: musikkpreferanse blant 6 mulige: rock, soul/rb, hiphop, jazz, classic, folk
- Is dancer: dancer eller not dancer. Om personen dansar på festar
- Ok guest: ok eller not ok - om gjesten fungerte ok på festen. Det er denne de skal predikere

Dei to algoritmane «k-næraste naboar» og «logistisk regresjon» er gjennomgått på forelesinga. De skal og prøve ut algoritmen «avgjerdstre» (decision tree).

Før du kan gjere noko må du installere dei rette programpakkane i dykkar python-oppsett. Du treng i alle fall pandas, numpy, og sklearn. Innlesing av data kan for eksempel gjerast med pandas sin read_csv()-metode. Eg reknar med de kan finne ut av dette. De kan finne masse nyttig informasjon om dette på nettsida

https://machinelearningmastery.com/machine-learning-in-python-step-by-step/

Alle data her er kategoriske, så til kvar av algoritmane må det gjerast nokre førebuande steg (preprosessering) på datasettet:

- **K-næraste nabo:** Sidan algoritmane i sklearn ikkje vil ha tekstverdiar, er det fornuftig å bruke indikatorvariable (one-hot-encoding) for kvar kategori. Dette gjer de best med pandas sin

get_dummies()-metode. Etter dette må de gå gjennom heile datasettet og erstatte alle verdiar i data-settet med 0 eller 1. Bruk for eksempel sklearn sin OrdinalEncoder. De skal bruke sklearn sin K-nearest neighbour-klasse til læringa:
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
De treng ikkje tenke på anna enn val av k, dvs. tal på naboar som skal brukast i klassifiseringa som parameter her. Om de ikkje vel k, vert den sett til 5. Prøv med versjonar der k = 3, 5, 11 eller 17.

- **Logistisk regresjon:** Her må de legge til rette ein litt annan versjon av datasettet. Logistisk regresjon fungerer nemleg dårleg når det er multi-kolinearitet mellom data. Det skjer når du brukar indikatorvariable sidan den siste av dei nye kolonnene vert automatisk bestemt av dei andre. Så du må då sette ein parameter i get_dummies() som heiter drop_first. Sjekk dokumentasjon på veven for å finne ut av dette.
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Standard-oppsettet for logistisk regresjon har noko som heiter regularisering innebygd. Regularisering vert styrt med ein parameter penalty (penalty='l2'). Prøv også logistisk regresjon med penalty=None
- **Avgjerdstre:** Dette er ein algoritme som lærer ein trestruktur (7.3.1 i læreboka).
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
De skal bruke same versjonen av datasettet som til k-næraste nabo. Det er to vanlege kriterier ein brukar for å bygge tre, det eine heiter gini det andre entropy. Prøv algoritmen med begge desse kriteria. (criterion-parametren til DecisionTreeClassifier).

Hugs at de før de køyrer sjølve maskinlæringsalgoritmen må dele det preprosesserte datasettet opp i eit treningssett og eit testsett – de skal bruke 80% av datasettet til trening og 20% til testing. Oppdelinga gjerast med
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
Det fins eksempel på korleis dette er gjort på tutorial-nettsidene som er nemnde i teksten ovanfor.

De skal måle kor gode dei tre algoritmane og deira variantar er ved å køyre prediksjon på både treningssettet og testsettet og samanlikne med den korrekte versjonen, dvs. klassifiseringa på treningssettet/testsettet. De skal bruke korrektheit (accuracy) som mål for både treningssett og testsett. Du skal i tillegg vise ei forvekslingsmatrise på testsettet. Skriv ut presisjon i klassifisering av gjester som 'ok'. Skriv til slutt ut forskjell på korrektheit for treningssett og testsett (viser grad av overtilpassing). Sjå seksjon 6 på nettsida
https://machinelearningmastery.com/machine-learning-in-python-step-by-step/

Programmet skal lage utskrift for kvar versjon (4 for kNN, 2 for logistisk regresjon, 2 for avgjerdstre). Kva for modell fungerer best om du vil ha færrast mogleg party killers på festen din (du vil at modellen skal vere presis i å klassifisere nokon som ok)?

Godkjenning: De skal enten

- vise køyringa av programmet dykkar på laben eller
- levere inn ei python-fil og ein pdf med utskrift av køyringa på Mitt UiB