Aksel Østmoe

TDT4171 Assignment 4

Task 1 A

Her har jeg valgt å fokusere på følgende kategoriske variabler. De variablene som ble utelatt var enten ettersom de er kontinuerlige, eller fordi jeg anså disse som irrelevante for klassifiseringen av datasettet. At jeg anså variablene som irrelevant selv om den kan anses som kategorisk, gjelder attributtet «Embarked».
Dermed vil de attributtene som brukes i byggingen av treet være:

- Sex: Ettersom kvinner og barn ofte ble reddet først fra skip anser jeg dette som relevant for hvem som overlevde
- Pclass: klasse på reisen vil jeg anta kan ha noe å si for pasasjerenes prioritet og hvor de befant seg på båten, som vil være relevant
- Parch: Hvor mange foreldre eller barn om bør vil være aktuelt, igjen på grunn av at kvinner og barn, og også familier, vil bli prioritert
- SipSp: Vil også på grunn av familie om bord på samme måte som punktet over, være relevant.

SipSp og Parch har såpass få verdier at jeg anså de som kategoriske.

Når jeg kjører koden i Task1a.py med disse variablene, finner jeg en accuracy på 0.8657074340527577

Task 1 B

Ettersom jeg ikke behøver å se på Age i denne oppgaven (fra Piazza), fokuserer jeg på Fare som eneste kontinuerlige variabel. En delvis implementasjon av denne oppgaven, med ferdig metode for splitting, kan finnes i Task1b.py

Task 1 C

Ettersom jeg ikke har fått noe endelig svar på oppgave 1 B blir det vanskelig å sammenlikne de to accuracyene. Jeg tenker at dersom jeg hadde hatt en fungerende implementasjon på oppgave 1 B, burde denne være mer presis enn det min accuracy i oppgave 1 A. Dette ettersom man har mulighet til å bruke flere attributter for å trene modellen til en mer presis tilnærming.

For å forbedre accuracyen til algoritmen kan man prøve å unngå overfitting, altså at algoritmen tilpasser seg mønstre i datasettet som ikke ville vært gjeldene i større og mer reele datasett. Dette kan unngås ved early stopping. Dette implementeres ved å se om å sjekke om å legge til attributter i treet skaper en økt accuracy i test-dataen, og dersom det ikke gjør det, så tas det ikke med. Dette vil være med på å minske risikoen for overfitting, men kan samtidig føre til at noen attributter i treet som skaper bedre accuracy, og ikke grunnet overfitting, vil bli tatt bort ettersom de for akkurat det datasettet vi har som testdata ikke skaper høyere accuracy.

Et annet tiltak som kan gjøres for å øke accuracy er å bruke bootstrapping. Da vil man lage tilfeldige subsett av dataen, og lage trær for disse. Så lager man en samlet accuracy basert på et gjennomsnitt av de man får på de ulike datasettene. Dermed kan man ha større tiltro til at accuracyen er riktig, og ikke offer for overfitting.

Aksel Østmoe

Task 2

De er flere ulike forskjellige måter man kan behandle manglende variabler i et dataset. Som det kom frem av diskusjonen på Piazza ble man oppfordret til å komme med egne forslag til hvordan man kan løse en slik situasjon på en hensiktsmessig måte. Per nå gjør algoritmene som har blitt implementert i øvingen ingenting dersom et attributt ikke har en tilordnet verdi. For eksempel kan man prøve å skape en tilnærming til hva verdien kan være, som baserer seg på de andre verdiene for dette attributtet. Et eksempel på en slik fremgangsmåte kommer frem av oppgave 18.9 i læreboken. I tillegg vil andre måter å skape en tilnærming til dette være å ta utgangspunkt i de andre eksempelverdiene for dette attributtet på dette punktet i learning-treet, og ta et gjennomsnitt av disse. Dette gjennomsnittet vil være et vektet gjennomsnitt dersom attributtet er en kategorisk variabel, og bør rundes av til nærmeste kategoriske variabel for å ikke påvirke hvordan treet blir utformet.

På samme måte som PLURAL-VALUE returnerer den klassifiseringen som oppstår oftest eksempelverdiene, kan man bruke typetall for å finne den verdien for et gitt attributt som forekommer oftest. Dette vil være best å bruke for kategoriske variabler, som tar en av et sett med predefinerte verdier. Dette vil derfor også være en måte å tilegne en verdi dersom de aktuelle attributtene ikke har tallverdier, men også inneholder bokstaver og tegn.

Jeg ser for meg at disse tiltakene kan implementeres i treningen på datasettet, hvor det kan implementeres en hjelpefunksjon som sjekker om det er noen tomme verdier i det eksempeldatasettet som jobber med i løpet av byggingen av treet. Dersom dette er tilfellet, kan det gjennomføres en beregning enten på et gjennomsnitt eller typetall som da kan plasseres i datasettet på den tomme plassen.