ML assignment 2: Project work

Kristin Håberg og Marie Levang

Problemet

Omfang/Scope

Modellen vil gi studenter mulighet til å undersøke hvordan egne vaner kan påvirke resultatene de oppnår på skolen. Det kan ha en positiv økonomisk betydning for samfunnet, samtidig som det er tidsbesparende for studentene, om det blir nedgang i antall studenter som ender opp med å droppe ut. Ifølge en rapport fra regjeringen fra 2010, kan manglende gjennomføring av videregående opplæring føre til økt behov for støtte fra ulike trygde- og stønadsordninger (Falch et al., 2009). Målet med modellen er å gi studentene en indikator på om deres vaner har negativ innvirkning for deres prestasjon, og slik gi dem mulighet til å endre negative vaner tidlig nok til å utgjøre en forskjell. Om vi klarer å redusere antall studenter som dropper ut vil dette hjelpe med at tenåringer eller unge voksne ikke faller fra i videregående skole, og kanskje vil ha ekstra behov for støtteordninger fra staten. I tillegg til å kunne redusere antall personer som trenger støtteordninger, vil det også være tidsbesparende for studentene om de får endret vanene sine tidsnok til å ikke stryke. De vil da kunne begynne å studere eller komme seg ut i jobb tidligere, enn om de måtte tatt opp igjen semester.

Løsningen er tenkt som en nettside hvor studenter kan legge inn sine arbeidsvaner og andre faktorer som kan ha innvirkning på resultater, slik at de kan følge med på om deres vaner vil ha en negativ innvirkning på resultatene de kan oppnå. Basert på parameterne som er fylt inn, vil nettsiden gi tilbakemelding på om det er sannsynlig at studenten vil lykkes eller ikke.

I dag blir dette problemet løst gjennom at studenter får tilbakemeldinger i form av delresultater i løpet av semestrene, dette gir studentene mulighet til å vite hvordan de ligger an før semesterslutt og endelig karakter. I tillegg vil studenter ofte sammenligne seg selv med medstudenter, venner og familie. Begge disse faktorene gjør at studenter kan justere egne vaner om de ser at det skulle være behov for det.

Modellen skal gi ut et svar på om det er sannsynlig at studenten står eller stryker. Vi valgte derfor å benytte en binary classification modell. Siden det er lite raske forandringer i data, vil det også være naturlig å gå for supervised og offline trening av modellen. Datasettet vi har brukt for modellen er ikke veldig stort, og det kan diskuteres om det er ideelt å bruke til dette. For å få en bedre modell burde nok data blitt samlet inn av oss for å bli brukt til akkurat det formålet vi vil bruke modellen til. Dataen er i tillegg hentet fra to skoler i

Portugal, så det vil nok være faktorer her som gjør det vanskeligere å predikere nøyaktig basert på norske studenters vaner.

Vi anser det som en god løsning å bruke maskinlæring for å løse dette problemet, da det ikke finnes noen god løsning for dette problemet i dag. Det er noen alternative løsninger som kunne blitt brukt, blant annet en enkel applikasjon der brukeren legger inn litt informasjon, og man har store kodeblokker med forskjellige looper som genererer et svar basert på hva som er lagt inn av informasjon. Dette ville blitt et tungt program å både produsere og vedlikeholde, da det ville inneholdt mye unødvendig kode, som må ta stilling til alle forskjellige kombinasjoner som kan legges inn. En annen alternativ løsning er at noen rundt studenten gir beskjed om at dens vaner kan ha negativ innvirkning på resultater. Denne løsningen er ikke veldig gunstig, da dette mest sannsynlig er noe veldig mange vil ha høy terskel for å gjøre.

Vi lagde en liten manuell løsning basert på kun parameteren for study time. Denne løsningen predikerte 331 riktig og 188 galt på om studenten ville stryke eller stå, altså gav den ca. 64% rett svar. Denne løsningen alene var 11 kodelinjer, og med tanke på at det er 16 andre parameter som skal inn i modellen, ville denne løsningen blitt unødvendig stor. Vi vil også si at å bruke maskinlæring for denne løsningen vil være mer treffsikker enn den manuelle løsningen ville vært.

Ideelt sett er det ønskelig at færre studenter skal stryke ved å bruke denne modellen. De kan sjekke hvordan de ligger an og endre vaner tidlig nok til at det kan utgjøre en forskjell. Dette kan også bidra med å få ned strykprosenten i skolen generelt, og samtidig redusere antall personer som faller fra.

Metrikker

Vårt mål for løsningen er 20 % lavere strykprosent blant videregåendeelever som tar løsningen i bruk. For å oppnå dette målet anser vi precision som en viktigere metrikk enn recall. Dersom en student ligger nær grensen mellom stryk og ståkarakter, tenker vi at det er bedre om modellen forteller enkelte elever som ville ha stått at de står i fare for å stryke, enn at den forteller elever som vil stryke at de ligger an til å bestå. Vi vil derfor forsøke å optimalisere for precision i vurdering av modellen.

Data

For å gjennomføre et slikt prosjekt, vil det være behov for data som omhandler elevers arbeidsvaner og resultater, og som er relevante for situasjonen modellen skal brukes i. Så vidt vi kunne se, var ikke dette noe vi hadde tilgang til per i dag for Norge. Dette er også data som det vil være personvernshensyn knyttet til, og det vil nok kanskje være tvilsomt

om en “mest for gøy” type applikasjon som dette vil bli regnet som et tilstrekkelig godt bruksområde for å benytte slike data. Det skal imidlertid ikke samles inn mer personopplysninger enn det som er nødvendig. Til vårt bruksområde trenger vi få demografiske opplysninger, men heller opplysninger om hvordan elevene forholder seg til skolen og skolearbeidet. Som labels vil vi ha behov for akademiske resultater, som for Norge kan være i form av en tallkarakter fra 1-6 i et spesifikt fag, eller eventuelt en boolsk verdi som sier om studenten har strøket i ett eller flere fag i inneværende semester. Det mest nærliggende vil være å samle inn slike data gjennom en undersøkelse som studentene svarer på selv. Det vil da være en risiko for at enkelte studenter velger å pynte på svarene de responderer, slik at garantert korrekte labels kun kan oppnås gjennom skolenes offisielle systemer. Vi anser likevel ikke risikoen for dette som veldig stor siden det ville ha blitt benyttet anonyme undersøkelser.

I mangel av reelle data har vi testet ut systemet på et datasett tilgjengelig via UCI Machine Learning Repository med data om resultater og studievaner hos elever på to portugisiske skoler (Cortez, 2008). Datasettet som omhandlet resultater i faget portugisisk ble benyttet til predikeringen da dette hadde flest oppføringer, men det finnes ingen garanti for at dette har overføringsverdi til norske elever og fag.

Vi identifiserte også noen konkrete problemer med datasettet. Hvilken skole elevene gikk på var en feature i datasettet som nødvendigvis måtte droppes, siden dette ikke har relevans i noen andre sammenhenger. Det viste seg imidlertid at skole var ganske sterkt korrelert med elevenes karakterer. De to skolene det var snakk om befinner seg i en liten landsby og i en mindre by, og vi så også at det var korrelasjon mellom skole og andre variabler, blant annet om eleven bor i by eller landlig, foreldrenes utdannelse, og elevens reisevei. Vi ser ingen annen løsning enn å droppe skolevariabelen og håpe at å beholde disse andre variablene til en viss grad kan veie opp for dette.

Et annet identifisert problem er at en ganske stor andel respondenter oppgir å ikke ha internett hjemme, noe som ikke er vanlig i Norge, og som også ser ut til å ha i alle fall noe påvirkning på elevens resultater. Det vil også være andre forskjeller mellom Norge og Portugal som kan ha innvirkning på resultatene, men dette har ikke blitt undersøkt i forbindelse med dette prosjektet.

Det var også flere andre features i datasettet som ble droppet fra prediksjonen, hovedsakelig fordi de hadde lav korrelasjon med målvariabelen. Dette gjaldt kjønn, alder, om eleven hadde gått på “Nursery school”, om eleven var i et romantisk forhold, familiestørrelse, paid classes, guardian, samt om foreldrene var skilte.

Videre var de fleste av variablene kategoriske, slik at encoding var nødvendig. Flere av variablene fulgte en skala fra 1 til 5 og var allerede lagt inn med disse tallverdiene som da kunne benyttes direkte. Mors jobb, fars jobb, og grunn for å velge skolen lå inne som forhåndsdefinerte tekststrenger, og ble derfor omdannet ved hjelp av one hot encoding.

Når vi etterpå studerte korrelasjonene til de ulike variabelene, så vi at det noen ganger kun var noen av features-ene som ble dannet fra de kategoriske variablene som hadde betydning. Dette gjaldt særlig for foreldrenes jobber, hvor det eneste som i praksis hadde betydning var dersom foreldrene var lærere eller var hjemmeværende. De resterende jobbene ble derfor droppet. I tillegg oppdaget vi at det ble oppnådd høyere korrelasjon ved å slå sammen variablene for hver forelder med én felles variabel. Vi erstattet derfor separate features for mor og far med features for høyeste utdanningsnivå for foreldre, om minst én forelder er lærer, og om minst én forelder er hjemmeværende.

Siden vi endte opp med å benytte Random Forest, og det ikke var ekstremt store forskjeller i tallverdiene til de ulike variablene, ble det ikke benyttet skalering. Det var to numeriske variabler i datasettet: Failures og absences. For absences var det verdier i datasettet mellom 0 og 93, hvor de aller fleste var i den nedre delen av skalaen. De øverste verdiene ble klassifisert som outliers. Vi valgte å benytte clipping på denne variabelen på verdien 25, da vi antar at det i liten grad vil være reelle forskjeller på en elev som har 25 dager fravær og en elev som har 40 dager.

Datasettet inneholdt ingen manglende verdier. Etter preprosessering endte vi opp med følgende sett features:

#	Column	Non-Null Count	Dtype
0	address	519 non-null	int64
1	traveltime	519 non-null	int64
2	studytime	519 non-null	int64
3	failures	519 non-null	int64
4	higher	519 non-null	int64
5	famrel	519 non-null	int64
6	freetime	519 non-null	int64
7	goout	519 non-null	int64
8	Dalc	519 non-null	int64
9	Walc	519 non-null	int64
10	absences	519 non-null	int64
11	reason_course	519 non-null	int64
12	reason_home	519 non-null	int64
13	reason_other	519 non-null	int64
14	reason_reputation	519 non-null	int64
15	parent_maxEdu	519 non-null	int64
16	parent_teacher	519 non-null	int64
17	parent_at_home	519 non-null	int64

Modellering

Som et første og grunnleggende forsøk prøvde vi å predikere med en Random Forest-modell med max depth = 3. Dette ga Accuracy score på 0.81, Precision score på 0.79 og

Recall score på 0.98. Vi benyttet deretter 10-fold cross validation for å undersøke videre, og fikk da lignende verdier: Precision scores mellom 0.70 og 0.94 (gjennomsnitt 0.80 med standardavvik 0.07), og Recall scores mellom 0.74 og 0.97 (gjennomsnitt 0.90 med standardavvik 0.06).

Det virket derfor som vi hadde en relativt godt fungerende modell fra start, men siden Precision var definert som viktigere enn recall i denne applikasjonen ønsket vi å optimalisere videre mot dette. Det ble også laget en Confusion Matrix, som bekreftet at modellen genererte en ganske stor andel falske positive, noe som ikke er ønskelig.

Siden vi har et lite datasett som gir kort treningstid, forsøkte vi et Grid search med precision som scoring-parameter. Resultatet av dette var RandomForestClassifier (max_features='log2', min_samples_split=5), og denne modellen ble forsøkt uten at det ga nevneverdige endringer i resultat.

Vi forsøkte videre andre modeller, nærmere bestemt SGDClassifier og VotingClassifier, men oppnådde lignende resultater også med disse. Siden det fremdeles var Random Forest som ga best resultat bestemte vi oss for å utforske denne modellen videre.

Da vi plottet feature importance, så vi at modellen la svært stor vekt på tidligere strykkarakterer, som er en variabel som er sterkt negativt korrelert med å stryke på nytt. Det var også tre variabler som knapt ble brukt av modellen i det hele tatt: Health, Extra-curricular activities og Extra school support. Disse hadde også lav korrelasjon med målvariabelen, og var variabler som vi anså som noe problematiske med tanke på personvern og overførbarhet. De ble derfor droppet fra modellen, noe som ga en svært svak forbedring, men fortsatt med lavere precision enn recall. Den nye feature importance-grafen var også mer balansert: tidligere strykkarakterer hadde noe lavere importance score, og alle de øvrige variablene ble benyttet av modellen i større eller mindre grad.

Vi sjekket deretter Confidence scores for radene generelt og falske positive spesielt ved hjelp av predict_proba(). Dette viste at confidence lå rundt 0.5-0.6 for mange av de falske positive. Vi bestemte oss derfor for å øke threshold for modellen til 0.65. Testing med validation set viste at denne modifiseringen ga bedre resultater:

Accuracy: 0.798
Precision: 0.84
Recall: 0.875

og de endelige resultatene ved testing med test-settet var:

Accuracy: 0.78
Precision: 0.88
Recall: 0.82

Deployment

Modellen skal settes i drift ved hjelp av et user-interface gjennom gradio som er koblet sammen med maskinlærings modellen. Her er tanken at studenter skal svare på relevante spørsmål for så å få ut en prediksjon på om det er sannsynlig at de vil stryke eller stå basert på svarene de har gitt. Prediksjonene skal brukes for å gi studenten en indikasjon på om de bør forbedre vanene sine eller ikke, og minske sannsynligheten for at studenten stryker.

For monitorering og vedlikehold bør det skrives kode for å sjekke ytelsen til systemet jevnlig. Siden dette er prediksjoner som blir gitt i forhold til hvilke vaner som er dårlige for studenter, anser vi de som generelle og at de mest sannsynlig ikke vil forandre seg mye over kort tid. For å forbedre systemet burde det også bli hentet inn mer data til datasettet, og muligens data som er mer relevant for å gi prediksjoner til norske studenter.

Referanser

Falch, T., A.B. Johannesen, og B. Strøm (2009). Kostnader av frafall i videregående opplæring Trondheim: Senter for økonomisk forskning AS

Cortez, P. (2008). Student Performance [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5TG7T.

Kodeassistenter: ChatGPT har blitt brukt til å generere enkelte kodeavsnitt, dette er markert med kommentar. ChatGPT og Gemini innebygget i colab er også benyttet til feilsøking/feilretting av kode.