TITANIC

Håkon Mydland, Georg Taklo Kvalsund og Even Torbjørnsen - ML gruppe 30

BESKRIV PROBLEMET

SCOPE

- Prosjektets mål var å finne om personene vil overleve eller dø under den historiske hendelser når Titanic synker.
- Løsningen som er laget skal kunne brukes til å avgjøre om personene ville overlevd eller dø når Titanic sank. Det finnes mange forskjellig løsninger på kaggle, men ingen i produksjon. Dette er mer et øvingssett for å øve maskinlæring. Oppgaven kunne mest sannsynlig blitt utført manuelt, men det er mye data å gå gjennom, og dermed regler for koden. Kvinner og barn var viktige å få reddet, og dermed vil det være flest menn som ikke kunne kommet gjennom.
- En business metric vil være hvordan systemet vil sikre bedriften inntekt, øke inntekt eller redusere kostnadene. Problemet og løsningen vil vi skal gå inn for å evaluere antall døde, og endre for å få antallet ned.

METRIKKER

- Etter ulykke så etterforsker ulykken, og ved Titanics tilfelle vil det bli stort erstatningsansvar for antall døde. Derfor vil gode modeller kan modellere hvem, som døde, der vi kan se at en lavere klasse døde oftere. Dette kan gi en evaluere og forbedre deler av evakuerings systemet til båten. Dermed vil færre døde, og mindre erstatningsansvar.
- Vi ønsker gjerne at modellen skal løse det problemet, som er definert. Derfor vil vi sjekke presisjon, nøyaktighet og at modell og system fungerer. Vi brukte F1 score hos PyCaret, som ga en god score på testsettet. Dermed vil modellen gi et bilde av sannsynligheten for om du hadde overlevd. Dermed kan vi fokusere på de høyeste verdiene for død. Verdien vi vurderer etter er erstatning etter ulykker.

DATA

Dataen som brukes ligger klart i kaggle competitionen, der det er en train.csv og en test.csv. Dataen inneholder passasjerid, alder, kjønn, billettklasse, billettnummer, passasjerpris, lugarnummer, avreisehavn, tallet på foreldre/barn, tallet på søsken/ektefeller, og til slutt labelen som sier om de overlevde eller ikke. Sistnevnte er en verdi på 0 eller 1, der 1 tilsvarer «overlevd» og 0 tilsvarer «ikke overlevd».

Dataen inneholder en god blanding av numeriske verdier, desimalverdier og tekstverdier, noe som må bli tatt hensyn til senere. Vi ser også at 20% av aldrende mangler i dataen, og siden vi ikke ønsker å droppe denne egenskapen, må vi prøve å fylle inn disse manglende verdiene.
Vi ser at kolonnen lugarnummer mangler i 78% av tilfellene i testsettet, og 77% av tilfellene i treningssettet. Gitt en så stor mangel blir det her vanskelig å fylle inn verdier, og man burde kanskje heller vurdere å droppe denne featuren.

I et slikt prosjekt kan det også ofte være lurt å bruke «feature engineering» for å lage nye features av de som allerede finnes. Det kan eksempelvis være interessant å vite snittalder per kjønn, eller forholdet mellom kjønn og billettklasse.

MODELLERING

Siden vi snakker om et klassifikasjonsproblem (enten har passasjeren overlevd, eller så har den ikke overlevd), var det naturlig å teste ulike klassifikasjonsmodeller. I første omgang testet vi de modellene vi hadde kjennskap til fra tidligere:

- Logistic Regression
- K-Nearest Neighbours
- Decision Tree Classification
- Random Forest Classification

Etter å ha fått kunnskap til pycaret, gjorde vi litt endringer i notebooken og fant de 6 beste modellene:

- Gradient Boosting Classifier

- CatBoost Classifier
- Ada Boost Classifier
- Quadratic Discriminant Analysis
- GaussianNB
- Random Forest Classifier

Vi observerte at de to listene var ganske annerledes, så det var fornuftig å bruke pycaret slik at vi fikk testet modeller vi kanskje ikke ville testet til vanlig. Resultatet ble forsåvidt likt som tidligere, ettersom det viste seg at random forest var den beste modellen.

For å kunne vurdere om maskineri er nødvendig, er det gjerne en baseline modell. Det er denne som må slås, og forbedres for at maskinlæring skal være nyttig. En baseline modell vil være den modellen som ble kodet tidligere etter klare regler, som deler inn i overlevelse eller ikke etter standard regler. Derfor vil maskinlære være relevant hvis den kan slå baselinemodellen.

Vi har i koden ikke sjekket noe spesifikt på feature importance, men viktigheten av at den ikke er avhengig av enkelte features, kan gis ved at en er dominerende. Hvis det er nok for modellen å vite kjønn og alder, så blir disse så dominerende at modellen gir feil bilde av enkelte hendelser.

DEPLOYMENT

Modellen deployes ved hjelp av en flask.applikasjon. Slik kan brukeren fylle inn verdier som blir sendt videre
som data inn i modellen, og du får en prediksjon på om personen overlevde Titanic eller ikke. I utgangspunktet skulle applikasjonen lastes opp til heroku, slik at den var tilgjengelig i skyen, og ikke bare lokalt. Etter mye prøving klarte vi dessverre ikke dette. Det finnes flere alternativer til deployment, men vi hadde ikke kunnskap eller tid til å teste/finne ut av dette.

REFERANSER

Kaggle competition:

https://www.kaggle.com/competitions/titanic/

Flask:

https://palletsprojects.com/p/flask/