ML car value finder

Omfang:

Målet med dette prosjektet er å utvikle en modell for å sette verdi på brukte biler. Det skal være et verktøy som folk kan
bruke når de ikke har så mye peiling på biler, eller bare en kjapp taksting av bilen. Måten dette blir gjort idag er å gå på finn, søke den bilen man har, også se hvilke priser de blir solgt for. Dette krever erfaring og tid. Ellers så blir det å dra til bilforhandlere, men da også forvente mindre pris.

Dette produktet vil bli en suksess i markedet fordi vi egentlig gjør det akkurat samme, bare at maskinlæringen gjør det fortere og mer effektivt. Modellen kan undersøker store mengder data på kort tid og kan lett finne mønster som vanlig folk ikke hadde sett.

Hvorfor blir det suksess på markedet?

Det er stor etterspørsel etter automatisk pris til bil. Som sagt vil det også spare tid, og en avansert analyse blir tilgjengelig for alle.

Hvem er dette for?

- Det er for privatpersoner som ønsker å selge eller kjøpe
- Bilforhandlere som vil ha rask pris vurdering
- Bil interesserte som vil finne verdi på bil.

Rett å slett alle som har tilgang til en bil vil finne nytte av dette produktet.

Business objective & impact:

Automatisering på vurdering av bil og fjerning av prisusikkerhet vil få andre til å bruke den mer og mer når de bli vant til det. Dette er fordi med kjappe detalje klikk så har du alt du trenger.

Vi kan tjene på modellen slik:

- Modellen blir integrert i andre apps eller nettsider for komersiell bruk.
- Kan har reklamer på siden
- Kan også integrere noen sponsorer der vi gir en ca pris en viss forhandler hadde gitt for den bilen.
- En viss forhandler som selger bilen med annonse til det.
- Kan også bli integrert i andre apps eller nettsider for kommersiell bruk.

Lønnsomhet metrics:

Vi kan måle lønnsomhet i prosjektet slikt:

Brukervekst → Antall brukere som bruker systemet → viser behovet

Feedback → rating fra brukere → Nyttighet og treffsikkerhet på modell

Korrekt gjetting → To knapper om bruker synes pris virker riktig → viser “trust” i praksis

Samme besøkere → hvor ofte en bruker kommer tilbake → viser nytte og tillit til modellen.

Ressurser:

Det er ikke mye vi trenger, for modellen trenger vi:

- En dataset med informasjon om bruktbiler og deres priser.
- Python, scikit-learn og streamlit
- Maskinvare for trening av modellen.

For nettstedet trenger vi:

- Domene
- Hosting
- Vedlikehold

Dette prosjektet kan bli gjor av en person, men når det skal videre til forhandlere eller integreres i andres apper behøves det flere i teamet, som selgere osv.

Metrikker:

Kvadratisk avvik = mean squared error
Treffsikkerhet
Latency throughput
Hvordan vi ikke bruker accuracy fordi det ikke fins rett og galt, men “mer riktig”

Kjappe definisjoner:
Lav MSE = små feil (dette er bra)
Høy R^2 (nære 1 som mulig) = modellen forklarer prisen godt

Første fase i treningen ble det slik:
MSE ca 1.9e9
R^2 ca 0.27
Det viser at modellen kan det grunnleggende, men den er way off.

Andre fase:

Mean Squared Error: 1.77e9
R2 ca 0.329

I andre fase endret jeg slik at vi brukte både motor og bil model input variabler for å forbedre prediksjoner.

Sammenligning med business mål:
Bedre MSE og R^2 betyr mer nøyaktig estimat og dette fører til at brukere stoler mer på systemet, de bruker det flere ganger, og selvfølgelig at det brukes mindre tid på å finne pris.

Data:

Kilde:
Datasettet er av bruktbiler, egenskaper til bilen og hvor mye den ble solgt for. Kilden for datasettet er Kaggle.
Det innholder flere type data:
Numeriske → pris, årsmodell, km stand, motorstørrelse
Kategoriske → drivstoff, girkasse, merke / modell og farge
Boolean → om den har vært i ulykke

Hele datasettet var omtrent 188 000 biler før rensing. Det var en CSV fil.

Rensing:
- Først å fremst fjernet jeg duplikater.
- Så fjernet jeg verider som var ekstreme som en bil over 1 million kroner. Dette var fordi det bare fjernet treffsikkerheten på andre bilen mer enn den faktisk hjalp til.
- Deretter sørget jeg for at det som står som tall er formatert som tall og samme med tekst
- Det var verdier som mangler og de har jeg enten fjernet eller fylt med “unknown”
- Sist men ikke minst gjorde jeg one-hot encoding, dvs å gjøre kategorier til nummer.

Labels:
Label i prosjektet er pris. Det gjør prosjektet til supervised regression.

Etikk og personvern:
Det er ingen personlig data, bare info om biler. Dette fører til at det er ingen GDPR-risiko og ingen personopplysninger.

Data represent:
Data ble representert som Pandas dataframe og senere → x (features) y (target)
Train / test 80 / 20.

Dvs at vi brukte 80% for å trene modellen og 20% for å teste

Modellering:

Jeg skal teste modeller med regresjon og predicte prisen på en bruktbil basert på historiske data. Vår target variable er “pris” og den er kontinuerlig, derfor passer regresjon best.

Jeg started med lineær regresjon. Den var rask og ga meg en god baseline for å hvordan prisen hengte sammen med alle variabler. Jeg kunne også se hvilke variabel som endret prisen mest. Hvis jeg merker at jeg trenger mer “korrekte” svar, vil jeg bruke Random Forest Regressor istede.

Baseline:
For baseline sammenligner jeg mot et par ting. Først er det gjennomsnittet i hele datasettet. Uansett hvilken bil så bør den kunne hvertfall slå den prisen.

Jeg måler også effektiviteten til modellen ved å bruke MAE, RMSE og R*2. MAE er gjennomsnittlig pris feil, RMSE straffer store feil mye hardere og R^2 sier om hvor god prediction det ble.

Feil-prediksjoner:
Jeg ser på feil prediksjoner for å forbedre modellen. Jeg ser hvor modellen bommer for å finne mønstre. Kanskje den strever med dyre merker, eller når bilen har høy km stand. Dette viser hvor den trenger forbedring og hva jeg må jobbe med.

Feature importance blir brukt for å se hva som endrer prisen mest. I lineær regresjon er det koeffisienten. Hvis noen bil egenskaper ikke påvirker prisen som forventet kan jeg forbedre det eller se om kanskje modellen tolker noe av de feil.

Deployment:

Det blir en webapp hvor brukere skriver info til en bil, også for dem et estimat med en gang av modellen. Det er forslag som blir gitt med tanke på historiske data.

Monitorering gjør jeg ved å lage en logg over modellens prediksjoner og faktisk pris over tid. Om det begynner og bli mye feil så vil det merkes. Jeg må også mate modellen med mer data, og kanskje bytte til en avansert modell hvis jeg ser effektiviteten svikter.

Referanser:

Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (3rd ed.). O'Reilly Media. URL: https://www.ingramacademic.com/9781098125974/hands-on-machine-learning-with-scikit-learn-keras-and-tensorflow/

Kaggle. (n.d.). *Playground Series S4E9 — Regression of Used Car Prices* [Competition page]. Retrieved November 2, 2025, from https://www.kaggle.com/competitions/playground-series-s4e9/

ChatGPT. (2025, October 25). Explanation of regression models in machine learning. OpenAI.

Chatgpt ble brukt for å lage hele app siden, altså hele webapp så vi får brukt modellen i nettleseren. Det ble også brukt for å forklare hvordan jeg kunne trene modellen, men det skrev jeg selv. Den ble også brukt til å kommentere å vurdere min rapport til å se hva som kunne forbedres.