# Used Car Price Prediction

Using data from German used car sales platform AutoScout24 we want to predict the potential asking price of a users car given its technical features (make, model, age, milage, …)

## Basics

### Business Plan:

We are a small used car sales platform operating purely via online sales.
Since new cars are expensive (and hard to get right now), we want to leverage new technologies to make our used car platform exciting.
This is why we want to integrate a car price prediction ML-component into our website.
Offering this a standalone app, users are then prompted to sell their car on our platform.
If their car was sold at or under 90% of the predicted price, they get half of that difference (actual sale price minus predicted price) as cash back.
Either way, the final sales price is used as feedback.

### Requirements:

- web app for desktop and mobile
- simple UI providing the user with an easy to use form to enter the cars features
- proactive form auto completion (e.g. given the model „SLK“, it suggests the make „Mercedes“)
- prediction is explained (e.g. you cars price is high because of its an „Aston Martin“ or you cars price is low because it has high milage)
- prediction is tied to referral code for feedback and money back (de facto user account)
- scraping for new data every day
- after sale has been made, feedback (final sale price) is submitted
- FEEDBACK: no standard deviation, but scrpaed data as „ground truth“ (80% of data is right) or wirkaufendeinauto.de as reference for feedback

- OPTIONAL: uploading picture of car, automatically fill in make, model and color (https://eyedea.cz/products/make-and-model-recognition-mmr)

## Requirements Specification

### Context specification:

Define the context in which the AI-component is deployed and operates

- standard web application
- Backend - handling user requests, saving requests, referral handling, provides feedback interface
- Frontend - for user interaction, input form, image upload, prediction presentation, sell options
- Model served as authenticated REST API
- Feedback, predictions + referral saved on relational DB
- prediction only for complete set of features
- prediction possible if at least 80% of input params exist in data

## Data specification:
Define requirements on the used data to ensure correct/desired functional and non-functional behavior of the system

- data origin:
  - Training: scraped historic data from AutoScout24, plus real time scraped data every day
  - Inference: data provided by user in web form, has monetary intent to give correct information
- data quality: since AutoScout24 needs all values to be filled in, the quality should be at least decent, duplicates may occur, for inference see above
  - ASSUMPTION: listed price (label) corresponds to actual sale price
- data properties: bias towards German car brands, no privacy related data is scraped, per row label structure prevents data leakage

## Model specification:
Define model type, parameters, metrics, dimensions, adaptability, portability.

- supervised, regression/gradient boost model
  - ideally with explainability
- batch learning - retrain every week + every 10 feedbacks
- success metrics:
  - deviation from actual sales price (feedback)

## Specification of metrics for real-time monitoring of deployed models:
Enable a continuous improvement and early data/model shift detection

- success metric: sales on the online car sales platform (referred from the AI app)
  - rational: if the predicted price satisfies the user, he/she will go on and sell the car (having the cash back in mind)
- success metric: money lost on false predictions vs. sale commissions

- improvement can only happen through finished sales, therefore giving the correct car sale price
- possible shift: dataset leans towards traditional gas engines, prediction on EVs/hybrids may be influenced by that

## Human factor specification:
Define how humans react on, for example, automated decisions

- users are presented the predicted price, having two options:
  - continuing with that price, listing the vehicle, being eligible for cash back on lower (less 90%) actual sale price
  - omitting the predicted price, using their own price, NOT being eligible for cash back (but still resulting in feedback for the model)
- if the prediction seems totally wrong to the user, it is possible too give direct feedback (e.g., „Report technical error“)

## Ethical requirements:
Define what ethical decisions are made by the system and how are ethical aspects will be addressed.

- only things/cars are rated using purely technical information
  - no involvement of human related info whatsoever
- compensation for deviating predictions clearly communicated with customer

## Non-functional / quality requirements:

Define requirements on quality attributes (explainability, legal req.)

**Accountability** - AI system need to be accountable for its decisions, actions, and effects on stakeholders

- insured by compensation for false prediction

**Controllability** - Degree to which an external agent can intervene in the AI's processes

- possible though scraped data (agent being AutoScout24)
- sale creation should NOT be possible for cars with estimated value over 200.000€ (market to volatile)

**Explainability** - Ability of explaining how an AI system has reached to a certain output

- given by using regression/gradient boosting ML algorithms

**Interpretability** - Degree to which humans can reason on the results of an AI system

- users with basic car knowledge can easily interpret the predicted price
- especially on the basis of the price they paid for the car

**Reliability** - Consistency of AI output (unlike reliably of a SW system, focusing on up-time)

- prediction reliable
- model/make image recognition may be prone to down time (because of limited API calls)

**Robustness** - Degree to which a system is keeping functioning when an error occurs during execution

- the user should NOT be able to list a car for extremely wrong prediction (safeguard like: 100 < price < 1000000)

## Hardware requirements:

Specify the HW systems the AI component trains and inferences on.

- model inference on edge device, served as REST API
- model training in data center/edge device