# Cars dataset report

**Team members:** Martin Maikov and Joosep Tiger Tilgar

Link to project repository

# Task 2

## Identifying business goals

### Background

At the moment the main conclusions regarding cars performance is based solely on technical specifications and car reviewers interpretation of those metrics, but there are other more interesting ways of approaching this matter. For example, out in the world there are up to hundreds of thousands of different generations of different car models from many different brands and a variety of makers. Therefore we have a lot of close to indifferent metrics to use to find interesting patterns and extract valuable data which can't be gathered just by looking at the technical specifications themselves, but rather by analysing it with machine learning algorithms.

### Business goals

Understanding of general trends in cars.

### Business success criteria

- Find at least three different unseen correlations in the data that can be used for further prediction training.
- Get certain attributes predicting accuracy higher than 75%.

- Be able to distinguish at least a handful of different trends in the car industry.

# Assessing situation

## Inventory of resources

The main resource of the project will be Jupyter notebooks containing our analysis and expect re-executable research steps. Also of importance are the parsed CSV datasets containing all the raw data for future use, and most importantly, cleaned CSV data files containing normalized values suitable for immediate use for machine learning algorithms or models. We also can't forget human resources, which include two University of Tartu undergraduate computer science students, who do not have a lot of prior data science knowledge. Main machines used for the project will be two laptops, because the computational load of this project doesn't require any fancy or powerful machine hardware.

## Requirements, assumptions and constraints

- All data used must be available on the open web and there can't be any existing datasets or cleaned data: everything must be gathered by ourselves.
- Vehicle prices are not taken into consideration due to time limitations, that car market browsing requires and how subjective the prices can be because of vehicle condition and inflation.
- Technical metrics must be comparable (except names) and normalized for easier processing.

## Risks and contingencies

There is a wide variety of technical specifications and units of measurements, even among indifferent metrics. Therefore, after scraping the data, it is often hard to normalize values over multiple sources and their ways of marking down specifications. Therefore there will be some compromises made in the parsing process and invalid or incomplete data might not be included or will be dropped in latter stages. One of the most common differences is the length of technical specifications list and what's included, what is left out or marked irrelevant. It is also

possible for sources to interpret names of vehicle generations in their own ways, for example by classifying models based on engine power output or size, which might not be official differentiator, or shoving many different generation releases under one single year etc. As we all know, technology is prone to becoming a relic of the past quite quickly, especially in today's world, where in a few years the principles observed currently might be thrown off and correlations will be rendered obsolete by advancements in autonomous systems and transition to new designs adapted to electric vehicles or things we can't foresee as of now.

## Terminology

**Make** - automobile manufacturer/brand (e.g., Toyota, BMW, Volkswagen)
**Model** - specific model line (e.g., Corolla, 3 Series, Golf)
**Generation** - distinct design/technical iteration of a model (e.g., Golf Mk7, BMW 3 Series G20, Toyota Corolla E170)
**Model year** - the year the vehicle was first sold/introduced (e.g., 2019)
**Facelift or LCI** - mid-cycle refresh of a generation with minor technical or visual changes
**Bodystyle** - body type (sedan, hatchback, wagon/estate, SUV, coupe, convertible, etc.)
**Drivetrain** - FWD (Front-Wheel Drive), RWD (Rear-Wheel Drive), AWD/4WD (All-Wheel Drive)
**Powertrain** - combination of engine type + transmission (e.g., ICE + manual, BEV, Mild Hybrid)
**Displacement** - engine capacity in litres or cubic centimetres (e.g., 2.0 L, 1998 $cm^3$)
**Forced induction** - turbocharged, supercharged, twin-charged, or naturally aspirated (NA)
**Power output** - maximum engine power in kW or PS (metric horsepower)
**Torque** - maximum engine torque in Nm
**Specific power** - power per litre of displacement (kW/L or PS/L) – key downsizing indicator
**Power-to-weight ratio** - power divided by kerb weight (kW/kg or PS/tonne)
**Curb weight** - vehicle weight with standard equipment, full tank, no passengers (EU: usually with 75 kg driver included)
**Acceleration 0–100 km/h** - acceleration time from standstill to 100 km/h
**Top speed** - manufacturer-declared maximum speed
**Combined fuel consumption** - official fuel/energy consumption (WLTP or NEDC) in L/100 km or kWh/100 km
**$CO_2$ emissions** - grams of $CO_2$ per kilometre (WLTP combined)
**Euro NCAP rating** - safety rating year and overall stars (if available for that generation)

**Platform** - underlying chassis/architecture shared across models or brands (e.g., MQB, TNGA, CLAR)
**Downsizing** - manufacturer strategy of reducing engine displacement while maintaining or increasing power via forced induction
**Feature** (in machine learning context) - individual measurable technical property used as input variable (e.g., “engine_power_kw”, “kerb_weight_kg”)
**Target variable** - variable we predict in regression tasks (e.g., “combined_fuel_consumption_l_100km_next_generation”)
**Observational unit or record** - one specific vehicle variant (one row in the dataset

### Costs and benefits

The project has no major cost besides invested human hours, which won’t be paid and are for educational learning. The only measurable cost is the cost of electricity, in total approximately 70 hours of run time power usage (for two lighter laptops).

## Defining data mining goals

### Data-mining goals

The first goal is to compare a set of machine learning models performance on the gathered and normalized data in order to test different approaches.

The second goal is to find possible correlations and ways of predicting certain technical metrics of any car and possibly comparing them to predictions of average human participants.

The third goal is to identify trends among car manufacturers and in their ways of designing vehicles.

### Data-mining success criteria

- Surprise findings among model generations, brand ethics etc.
- Less than 10% error for predicted numerical features.

# Task 3

## Gathering data

### Data requirements outline

There are no data requirements, we use all of the data we can get, since there is no set goal requiring us to analyse fuel consumption. We'll do that, but if data doesn't exist, then we don't do that.

### Data availability verification

All the data we intended to use was easily available and accessible at any time from anywhere, although one had stricter server request policy. Using standard BeautifulSoup parsing tools resulted in 403 errors and therefore required browser emulation with PlayWright toolset instead. As explained in the following subsection, we ended up not using the page that had this constraint. Due to realizing how long it takes to normalize the columns data, we decided to narrow our focus to only two sites, Auto-Data and Carsdirectory, because they were the cleanest of all three and also it was way easier to split the work among two sites rather than three ones.

### Data selection criteria

In this current project we are using four most well-rounded online consumer cars data websites that we know of:

1) https://www.auto-data.net/en/
2) https://www.cars-directory.net/
3) https://www.thecarspec.net/
4) https://www.autoevolution.com/cars/#letterR

All of these sites have generally layered their car data in specific tree structure, making it easier to parse. The first tier consists of car brands, the second tier is models inside one brand, furthermore, generations of one model and finally the technical specifications for every independent generation release. Unfortunately the fourth page couldn't be scraped effectively because of stricter request policy (whole scrape takes well over an hour) and messier sub-page structures and mismatched generation names with non-uniform DOM-tree element classnames and misleading redirections. Also in the case of Auto-Data websites, there were also generations without any content, resulting in None values, but missing parameters were overally a common occurrence.

## Describing data

The first dataset (auto-data.net) contains ~50 000 rows and is gathered through website scraping on the auto-data.net website using Python, requests and beautifulsoup tools. It contains all the technical parameters that each car on there has, which is suitable for our goals.

The second dataset (cars-directory.net) contains ~38 000 rows of data and is gathered by web scraping via Python scripts. It contains technical specifications about every generation of brand models, which are mentioned on the site, making it suitable for this project.

The third dataset (thecarspec.net) contains ~37 000 rows and is gathered through website scraping on the thecarspec.net website using Python. It also contains all the technical parameters that each car on there has, which is suitable for our goals.

## Exploring data

The first dataset contains lots of duplicate columns that need to be merged together. The reason for duplicate columns is that there are a lot of different standards or ways to measure it. Fuel consumption is often measured separately for city and highway driving. CO2 emissions have NEDC, WLTP measuring standards, etc. It also contains a lot of computable attributes: power per litre, torque per litre and others. There are also empty or useless columns: Engine oil specification, which have no values in it.

The second dataset was the most compact and concise out of all three. Despite its lower column count, it doesn't contain any duplicate rows and it maintains a high fill rate throughout all the columns, making it easier to work with.

The third dataset was the messiest of all three, consisting of about 150 rows and half of it was made of mostly empty rows with less than 1% fill rate. Therefore it had a bit less mostly full rows than the previous ones and that's why it didn't make it to the top 2.

## Verifying data quality

The first dataset doesn't seem to have quality problems.

The second dataset doesn't seem to have any problems as well.

The third dataset has quite a few quality problems because of very few random data points in certain rows, therefore needing more cleanup, but stricter scraping policy could have solved that problem.

# Task 4

# Project plan

## Tasks and work distribution:

| nr | Task | Martin | Joosep |
|---|---|---|---|
| 1. | Identifying our goals and specifying the boundaries of the main area of interest. | 1h | 1h |
| 2. | Looking for sources of data and concentrating the data mining focus on a smaller set of data sources. (Searching for web-databases using google) | 1h | 1h |
| 3. | Searching for data: building web scrapers to collect data and translate it into usable format. (Python and CSV) | 4h | 8h |
| 4. | Completing the project overview report. | 1h | 4h |
| 5. | Cleaning the dataset and normalizing attributes. (Jupyter notebook, Pandas dataframe tools and Python) | 5h | 1h |
| 6. | Identifying different ways of processing cleaned data. Choosing appropriate machine learning algorithms for experimentation. (Jupyter notebook) | 4h | 4h |
| 7. | Experimenting with different classification strategies and testing different machine learning models and measuring their performances. | 2h | 2h |
| 8. | Choosing the best algorithms and training them to find correlations in the data. | 2h | 2h |
| 9. | Optimizing models based on results adding normalization algorithms to prevent overlearning. | 3h | 3h |

| | | | |
|---|---|---|---|
| 10. | Finding the points of interest and correlations in the data. Charting our findings. (Python, Pandas, Matplotlib etc) | 2h | 2h |
| 11. | Documenting our findings and making appealing charts to visualize our findings. (Seaborn, Plotnine and Matplotlib) | 2h | 2h |
| 12. | Creating a project poster for project presentation (Canva) | 0,5h | 0,5h |