Business understanding

Identifying your business goals

Our project is a part of a competition with a defined goal and ruleset so the CRISP-DM process isn't entirely applicable and relevant for our purposes.

Background

Self-driving cars are an incredibly fast growing market that has really skyrocketed within the last few years. Remarkably many car manufacturers such as Honda, Toyota and Tesla have come out with models with different levels of autonomy. These cars use many sensors (radars, lidars, sonars) to safely navigate traffic. Despite all efforts, Honda released the first legally approved level three autonomous vehicle (meaning the driver, under some conditions, doesn't have to pay attention to traffic) in 2021. The biggest problem, besides the incredible complexity of technology required for a safe self-driving car, is the opinion of the public. A study conducted in Cambridge, UK reported that "users were less accepting of high autonomy levels and displayed significantly lower intention to use highly autonomous vehicles." Therefore the future of self-driving cars might be in jeopardy if people don't warm up to the concept and if tragic accidents involving self-driving cars continue getting extensive media coverage.
Fortunately, our project is just a small-scale, gamelike dip into the world of self driving and neural networks.
This project is for the DeltaX Self Driving Competition, where there are two sub-goals for the toy car for which we will build a self-driving model for: obstacle avoidance and route completion, the details of which can be found on this page. We are using the DonkeyCar, which is a remote controlled car with a Raspberry Pi 4 and a small camera

mounted on it. There’s a python environment and an app developed for communicating with the car and gathering data via a designated wireless network.

Business goals

Not exactly relevant for our project, despite there being cash prizes for doing well in the competition. Our goal is mostly just to take part in the competition.

Business success criteria

Not relevant for our project as we don’t want to set rigid goals in terms of how we compare to other teams.

Assessing your situation

Inventory of resources

We have a DonkeyCar set, complete with all the necessary accessories (batteries, a controller etc) for the project.

Requirements, assumptions, and constraints

The interesting thing with requirements is that we have two due dates: one for this course in December and the competition in January. This might actually be good, because if we meet our goals by December, we’ll have a whole month for making improvements to be extra competitive.

The main assumption we will be making in our model, is that it will only be driving in the toy town, with known wooden walls and 90 degree turns. As this is the only training data we will use, the model will most likely be effectively useless outside the toy town.

Risks and contingencies

The most significant risk/contingency is the functionality of the hardware - if that works we’ll be able to gather data and test models on the car.

Terminology

No special terminology is needed to understand our project that hasn't already been defined within the scope of this course.

Defining your data-mining goals

Data-mining goals

Find and use an appropriate machine learning algorithm that could accurately predict DonkeyCar steering metrics based on the camera image. The goal for the outcome of the data-mining process is simple, as we are taking part in a competition - it's to do as well as possible. That and getting all the valuable experience of working with software, hardware, data mining, model building.
Our goal will be to test different models on different layouts of the toy town to find one that works best.

Data-mining success criteria

We don't plan on setting a goal in terms of placement in the competitions, as the other teams are obviously very competitive, too. The main success criteria for us will be to: develop a model for our DonkeyCar that manages to drive around for a minute, with average speed, in the toy city, without hitting an obstacle.

Data Understanding

Gathering Data

Data requirements

The data must include images that were captured while driving and information about the car's throttle and steering angle. The information about the steering angle and throttle must be connected to a corresponding image via timestamp, because in the end the image and throttle information are used to predict the steering angle.

Data availability and selection criteria

The data was gathered by driving a remote controlled car in a toy town. The car is equipped with a camera that records images and a RaspberryPi computer which is used to communicate driving data such as images, throttle and steering angle values between the car and a host PC. Every driving session creates a new ‘tub’ in the project folder which consists of images and .json files which will later be used during training.

Describing and Exploring the Data

The .json files consist of six key/value pairs, the keys are: index, session_id, timestamp, cam/image_array, angle, throttle. The key cam/image_array refers to the filename of the corresponding image with the same timestamp, session_id is constant therefore irrelevant from training perspective, angle is a real value within [-1,1] and throttle is a real value from (0,1]. One session consists of around 20,000 .json files (and corresponding images).
The gathered data is sufficient for analysis and training, because it contains the predictable value ‘angle’ and the values used to predict the steering angle: ‘throttle’ and the corresponding image.

Upon reviewing the data no inconsistencies were found. No images were missing or corrupted. The throttle and steering angle values were from expected intervals and the evaluations were plausible.

Verifying Data Quality

The dataset is of appropriate size, no missing values were detected, images are uncorrupted. Some problems might arise concerning the modest picture quality and the monotonous colour palette. The reason for this is that later on it could be hard for the algorithm to detect edges of the toy town walls, since the corridor walls, the toy town walls and the floor are of similar colour. To overcome this difficulty we might implement some picture editing like cropping to make the walls more easily detectable for the algorithm.

Project Plan

Part of the self-driving competition is **gathering the data yourself** by driving around the track. All team members can participate in gathering the data. We also need to **get the car to work reliably** as we have had some issues.
We have already tested the data collection and everything seems to be working. After that we need to **research models for self-driving.** Once we have a better understanding of the necessary model we will probably need to **preprocess the data** and after that **implement the model(s)**. Once we have some models we have to **start testing** them out. To our knowledge there is no other way to test the models other than experimenting with them on the track. The final step is to **take the best model and participate in the competition**.

Tasks/Names	Hans Kristjan	Karl-Ingmar	Hendrik
Setting up the DonkeyCar	4h	4h	4h

and the necessary Python environment.			
Gathering data	1.5h	1.5h	1.5h
Sort out some mechanical issues with the car	1h	2h.	1h
Research image recognition algorithms	3h	3h	3h
Research what models are used for self-driving	3h	3h	3h
preprocessing the data	3h	3h	3h
Implement models	5h	5h	5h
Testing/evaluating models	5h	5h	5h
Work on presentation	3h	3h	3h

s, texts, reports, documentation, etc			
Participate in the competition	3h	3h	3h