UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Artur Kreegipuu

Creation of Materials to Teach Data Science via Self-Driving

Bachelor's Thesis (9 ECTS)

Supervisor: Ardi Tampuu, PhD

Tartu 2024

Creation of Materials to Teach Data Science via Self-Driving

Abstract:

Data science projects often contain pitfalls that students might understand theoretically but rarely experience in practice. Such crucial problems are vividly demonstrable on self-driving toy cars. Students can witness how common data science mistakes during the training and deployment process impact the performance of self-driving neural networks. This thesis focused on creating practical study materials to help students understand, detect, and prevent common pitfalls in data science using self-driving toy cars. Practical tasks that involve gathering data, training and deploying self-driving models, highlighting various machine learning pitfalls and limitations of artificial intelligence, were developed for the educational materials. Practical tasks were tested in various different lighting conditions and expected outcomes were filmed. Feedback from two machine learning experts was collected about the study materials.

Keywords:

machine learning, neural networks, self-driving neural networks

CERCS: P176 Artificial intelligence, S270 Pedagogy and didactics

Andmeteaduslike õppematerjalide loomine isejuhtimise abil

Lühikokkuvõte:

Andmeteaduse projektides esineb tihti kitsaskohti, millega õpilased võivad olla teoreetiliselt tuttavad, kuid puuduvad praktiliselt kogemused. Sageli esinevaid probleeme saab demonstreerida isejuhtivate mudelautode abil. Praktiliste ülesannete kaudu saavad õpilased kogeda, kuidas isejuhtivate närvivõrkude arendamise etapis tehtavad andmeteaduslikud vead mõjutavad isejuhtiva mudeli sooritusvõimet. Lõputöö raames loodi isejuhtivate mudelautode abil praktilised õppematerjalid eesmärgiga panna õpilased mõistma, ära tundma ja ennetama andmeteaduses laialdaselt levinud probleeme. Selleks loodi probleeme demonstreerivad praktilised ülesanded, mille käigus tuleb õpilastel koguda andmeid, treenida isejuhtivaid närvivõrke ja katsetada loodud mudeleid rajal. Kõik ülesanded lahendati testimise eesmärgil erinevates valgusoludes ja iga ülesande eeldatud tulemist filmiti video. Loodud õppematerjalidele koguti tagasisidet kahelt masinõppe eksperdilt.

Võtmesõnad:

masinõpe, tehisnärvivõrgud, isejuhtivad närvivõrgud

CERCS: P176 Tehisintellekt , S270 Pedagoogika ja didaktika

Table of Contents

Introduction 5

1 Background 6

  1.1 Self-driving Toy Cars as a Learning Tool 6

  1.2 Backward Design Method 7

  1.3 Donkey Car 8

    1.3.1 Autonomous Driving with Donkey Car 8

    1.3.2 Use of Donkey Car in the University of Tartu 9

2 Methodology 11

  2.1 Selecting Data Science Pitfalls to Demonstrate 11

  2.2 Creating Study Materials Using the Backward Design Method 14

  2.3 Data Gathering and the Track Setup 16

  2.4 Model Training 20

  2.5 Model Testing 22

3 Results 24

  3.1 Study Materials 24

    3.1.1 Qualitative Description 24

    3.1.2 Quantitative Description 28

  3.2 Expert Feedback 28

4 Discussion 30

  4.1 Limitations and Future Work 31

List of References 33

Appendix 36

  I. List of Data Science Pitfalls Generated by ChatGPT 4 36

  II. Created Study Materials 38

  III. Gathered Data Description 39

  IV. Model Testing Results 40

  V. Expert Feedback 42

  VI. License 45

Introduction

Data science contains many pitfalls that are frequently mentioned in various study materials, such as Andrew Ng's *Machine Learning Yearning* [1] and Tom M. Mitchell's *Machine Learning* [2], but are rarely experienced by the students. Encountering these problems in the real-world data science projects can slow down the work progress [3] due to the lack of transparency in modern machine learning algorithms, making it challenging to debug incorrect outputs [4]. Therefore, understanding common problems and following good practices are essential for conducting successful machine learning projects. The primary objective of this thesis is to develop study materials that address common failure points in data science by using self-driving toy cars, which provide students with a practical and engaging way to experience some pitfalls first-hand. The study materials are intended for students with a basic background in computer science but without deep knowledge of data science and machine learning. The materials are designed for individuals interested in learning about data science and understanding why artificial intelligence might encounter failures. It is important to note that the created study materials do not serve as an independent and comprehensive course on common problems in data science. Instead, they are designed to be complemented by additional theoretical materials, which were not developed within this thesis, to cover topics not addressed in the practical study materials.

The thesis is divided into 4 chapters. The first chapter provides some pedagogical background about why learning through experiences is beneficial and describes the backward design method used for creating the study materials as part of the thesis. It also gives an overview of the central platform used in the study materials, called Donkey Car. The second chapter describes the methods used in developing the study materials. The third chapter presents the study material together with its analysis and feedback received from experts in the domain of data science. The fourth chapter discusses strengths, limitations, and future development options of the created study materials.

This thesis benefitted from the assistance of ChatGPT 3.5 [5]. This chatbot was utilised solely to correct grammar mistakes and clarify sentence wording. It was instructed not to generate any new knowledge, and this directive was also specified in the prompts. Linguistical suggestions generated by the ChatGPT, were considered, carefully reviewed, and adjusted as necessary before incorporation into the thesis.

1 Background

This chapter provides a description of important concepts used in creating study materials using self-driving toy cars. It discusses the advantages of studying through practical experiences and introduces the method of backward design, which was employed in developing the study materials. Additionally, the chapter introduces the platform utilised for integrating self-driving technology into the created study materials.

1.1 Self-driving Toy Cars as a Learning Tool

Successful learning is not merely about memorising facts but rather about developing a deeper understanding along with the ability to apply new knowledge. According to David Kolb's [6] experiential learning theory, learners acquire new knowledge more effectively when they engage not only in passive information reception through reading or listening but also in practical experience. Similarly, Hao et al. [7] argue that a practical approach to learning and teaching has a significantly positive impact on knowledge acquisition, particularly in the context of computer science education. Hao et al.'s study revealed that the learning environment plays a crucial role in how new information is conveyed to students. The research indicated that students studying computer science who experienced active learning and teaching methods, such as answering short quizzes before lectures, achieved greater academic success compared to students who solely relied on conventional learning methods, such as reading textbooks independently and passively listening to lectures. This underscores the rationale behind this work, which aims to create educational materials for illustrating common problems in data science, via hands-on tasks using self-driving toy cars.

Self-driving toy cars present an excellent opportunity to be used in practical learning materials. When a learner witnesses a self-driving toy car deviate from the safe path or collide with an obstacle due to a well-known mistake made during model creation or deployment, it provides a stronger learning experience compared to simply reading about these issues in a textbook or visualising them on charts or computer screens.

Additionally, the use of self-driving small cars for educational purposes is important to prepare people for fully autonomous driving in the near future due to recent fast developments in the field of autonomous driving, as emphasised by Li et al. [8]. In their article, the use of self-driving toy cars in educational materials is also emphasised as this will shape the advancements that will be made in the field of autonomous driving in the near future.

1.2 Backward Design Method

In the development of the educational study materials presented in this work, reliance is placed on the backward design (BD) instructional framework. This chapter provides a summary of how Wiggins and McTighe describe the BD framework and its core stages in their book *Understanding by Design* [9]. The authors assert that a key characteristic of BD is to start with the desired learning outcomes when designing educational materials and methodologies.

The initial stage involves defining the objectives of the lesson or course. When establishing the objectives that students are expected to achieve, considerations must be made for the allocated teaching time and other possible requirements, such as fulfilling national curriculum standards. In setting objectives, it is recommended to consider the following questions:

- What should students briefly acquaint themselves with or encounter to some extent?
- What essential knowledge and skills should students possess?
- What are the core concepts that form the nucleus of acquired knowledge and should be retained by students long after the course or lesson concludes?

Once the objectives are clearly defined, it is easier to provide students with tasks that genuinely contribute to achieving the learning objectives.

The second stage of BD involves determining how to assess whether students have achieved the set objectives. The material developer must clarify how to determine that a student has achieved the learning objectives and what to consider as appropriate evidence. Confirming students' purposeful acquisition of knowledge can be accomplished through assessment and organisation of homework assignments, quizzes, group work, or other similar activities.

The third and final step in creating a study material using the BD framework involves determining teaching methods. Within this stage, answers must be found to the following questions:

- What skills and prior knowledge should students possess to achieve the learning objectives?
- What activities will help students achieve the objectives?
- How should these activities be guided?
- What educational materials are best suited to achieve the objectives identified in the initial stage?

Wiggins and McTighe note that while BD is primarily intended for use in multi-day courses, the principles of this framework can be adapted for creating individual lessons or workshops, as is the case with the educational material being developed in this bachelor's thesis.

1.3 Donkey Car

This thesis uses the Donkey Car[1] platform, an open-source self-driving software for small-scale cars, to exemplify prevalent challenges in the domain of data science. The 1:10 scale Donkey Car S1 platform equipped with a Raspberry Pi 4B[2], MM1 remote control board, and a single frontal wide-angle Raspberry Pi camera is employed in this thesis. This software and hardware combination is designed to be an affordable and accessible way for enthusiasts to experiment with autonomous driving technology. The open-source code base provided by Donkey Car is supported by a Python framework compatible with leading machine learning frameworks like Keras[3] and Tensorflow[4]. This simplifies tasks such as gathering and cleaning training data, training neural networks, analysing and describing the results, even for novices within the field of autonomous driving technology. Moreover, Donkey Car has an inclusive and supportive community on Discord, further enhancing its appeal to newcomers in the domain.

Donkey Car‘s throttle and steering can be controlled via a physical gamepad, a mobile app "Donkey Car Controller" (available on both Android and iOS), or a web-based interface. The Logitech Wireless F710 Gamepad, used in the thesis, is among the compatible physical gamepads listed in the Donkey Car documentation [10].

1.3.1 Autonomous Driving with Donkey Car

Achieving autonomous driving with the Donkey Car involves utilising behavioural cloning, a machine learning technique used in training a machine to replicate human decision-making, as described by Sammut [11]. Training data has to be gathered by manually driving the car around the designated training environment. This training data comprises numerous frames, each containing a camera image paired with simultaneously recorded throttle value and steering angle value. Throttle values range between -1 and 1, where -1 indicates moving backwards and 1 indicates moving forwards at maximum speed. Steering angle values range between -1 and 1, where negative values indicate steering to the left and positive values

---

[1] https://www.donkeycar.com/

[2] https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[3] https://keras.io/

[4] https://www.tensorflow.org/

indicate steering to the right. With the collected dataset, a model can be trained to predict both throttle values and steering angles based on input images. By deploying this model on the Donkey Car, a self-driving vehicle capable of emulating the behaviours learned from the training data is created. In essence, behavioural cloning leverages machine learning to teach the car how to drive autonomously by mimicking human driving patterns.

To accomplish autonomous driving, artificial neural networks are employed. The following paragraph describing neural networks is based on the book by Goodfellow and Bengio [12]. Neural networks consist of interconnected nodes, or neurons, organised into layers. Through a process called training, the connection strengths between neurons are modified, allowing the neural networks to learn complex patterns and relationships within the training data. Each neuron processes input values received from other neurons to generate an output. Inputs are scaled by the corresponding connection strengths, combined through weighted summing and applying a non-linear activation function to the sum. During training, the network refines these weights, i.e., connection strengths, by comparing its output to the desired output, using a loss function to evaluate the disparity, and adjusting the weights in a way to minimise this disparity (i.e., minimise the loss).

A trained neural network deployed onto the toy car processes real-time camera inputs to make instantaneous decisions about steering and throttle control. Through this integration, the Donkey Car is empowered to navigate its environment autonomously.

1.3.2 Use of Donkey Car in the University of Tartu

The University of Tartu Autonomous Driving Lab has acquired 14 Donkey Cars for educational and PR purposes. In education, it is not feasible to allow beginner-level students the use of an expensive real-sized vehicle. However, the experiences of dealing with hardware, collecting data in the real world and deploying solutions to the real world are considered beneficial. Such experiences provide a foundation to achieve success in subsequent projects with the real vehicle [8].

The Donkey Car platform has served as a central tool in various contexts. It has been used for competitions like the annual ADL Minicar Challenge[5] organised by the Institute of Computer Science of the University of Tartu which also serves the purpose of teaching the

---

[5] https://adl.cs.ut.ee/teaching/minicar-challenge

competitors how real data science projects are conducted. Additionally, every semester, the vehicles are made available for course projects in various courses, including Autonomous Vehicles Project, Machine Learning, Intelligent Transportation Systems, Introduction to Data Science and Neural Networks.

Moreover, Donkey Car has been instrumental in numerous scientific inquiries, with several master's thesis utilising the platform to explore pertinent research questions. For instance, Roosild [13] used the Donkey Car platform to point out the potential pitfalls of autonomous driving systems when confronted with speeds divergent from those encountered during training, shedding light on deployment challenges in real-world scenarios. Uduste [14] delved into the effect of delays in the autonomous driving systems using the Donkey Car platform and proposed a novel method called frameshift to counter it. Camara [15] experimented with the Donkey Car to investigate if and how adversarial defence methods could improve the generalisation performance of deep-neural-network-controlled automatic driving systems. His findings underscored the feasibility of improving system robustness under certain conditions. Demonstrating the viability of this platform for internationally recognized research, a scientific manuscript [16] was published based on the works of Roosild and Uduste.

2 Methodology

This chapter provides an overview of methods used to create the study materials. Subsequently, it describes the process of choosing the concepts to demonstrate (the learning outcomes) in the study material, the process of creating the study material as well as how the practical tasks were created. Notice that the creation and verification of practical tasks involves numerous repetitions of training and testing the self-driving models.

2.1 Selecting Data Science Pitfalls to Demonstrate

The goal of this work is to develop practical materials that put in clear evidence some of the common failure points in data science projects. According to the widely-used CRISP-DM [17] framework, the life cycle of a data science project comprises multiple phases. At a coarse-grain view, these phases include defining the problem that needs solving, translating it into a data science question, acquiring data, visualising and verifying data quality, cleaning and preparing the data for modelling, model selection and creation, model evaluation, model deployment, model performance monitoring. The process is not linear and it may require cycling back to earlier stages to incorporate lessons learned in subsequent steps. In many of these steps, inexperienced practitioners may stumble into common mistakes or pitfalls.

An initial list of potential pitfalls in data science to address in the study materials was generated by ChatGPT 4 by asking it to generate a list of common failure points of machine learning solutions, where things can and often go wrong. Full prompt and response of ChatGPT 4 [18] can be found in Appendix 1. This source of information was selected because the many blog posts on the topic all seemed to offer a partial (e.g. only data quality) or biased view (usually skewed towards business applications). Scientific sources search yielded no comprehensive list across the entire life cycle of a project. The thesis supervisor, an expert in the field, validated the GPT-generated list as comprehensive and correct.

As a second step, we determined which of the problems would be most relevant and vividly demonstrable with self-driving toy cars. Expert opinion of the supervisor was used together with the observations from the experiences with the Donkey Car documented in the form of blog posts by self-driving enthusiasts [19-22]. As a result, the initial list of 20 common points of failure in data science generated by ChatGPT 4 [18] was reduced to 13, and later further 2 were discarded. The process of narrowing down the scope is shown in Figure 1.

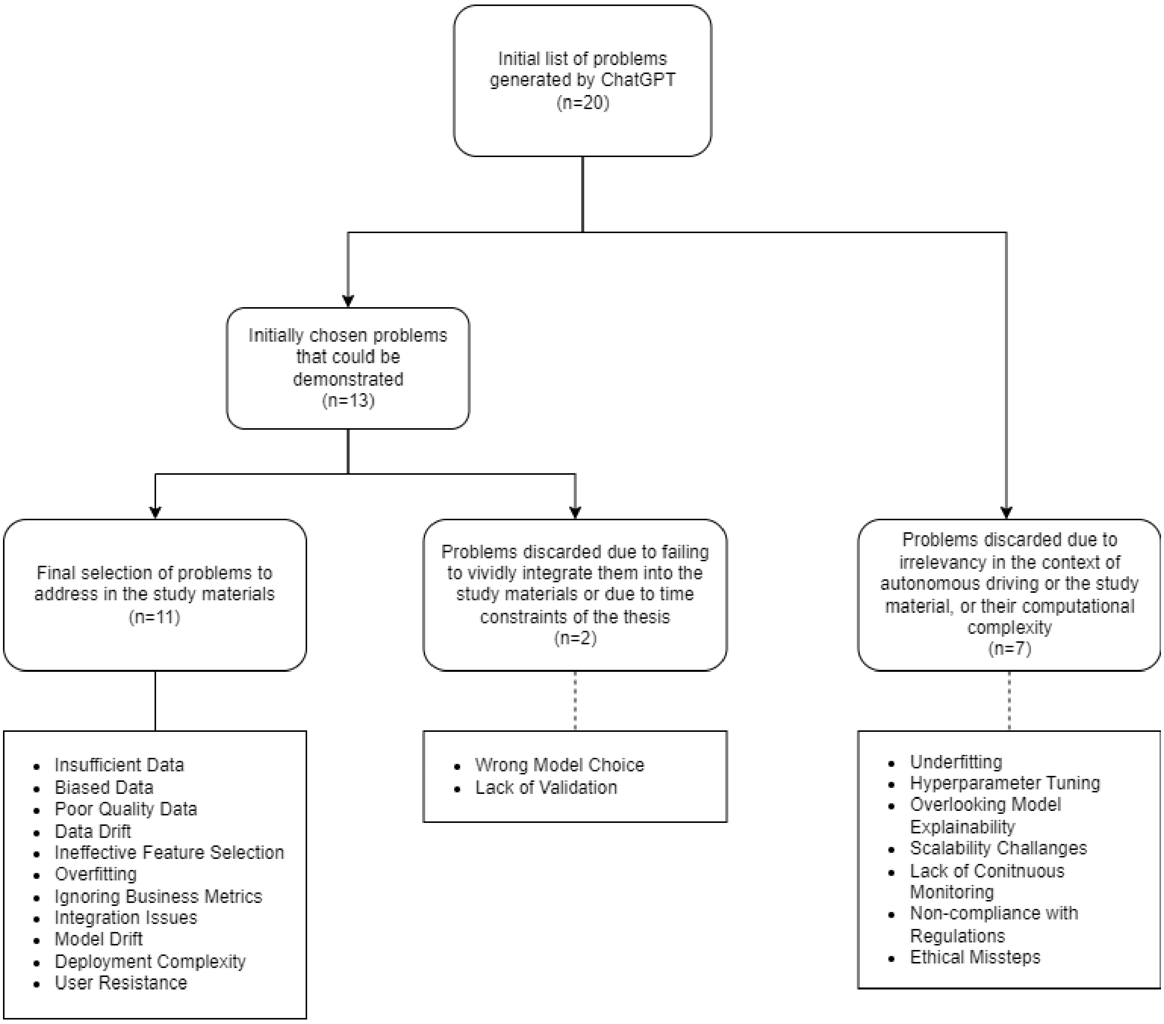

Initial list of problems generated by ChatGPT (n=20)
Initially chosen problems that could be demonstrated (n=13)
Final selection of problems to address in the study materials (n=11)
Problems discarded due to failing to vividly integrate them into the study materials or due to time constraints of the thesis (n=2)
Problems discarded due to irrelevancy in the context of autonomous driving or the study material, or their computational complexity (n=7)
Insufficient Data
Biased Data
Poor Quality Data
Data Drift
Ineffective Feature Selection
Overfitting
Ignoring Business Metrics
Integration Issues
Model Drift
Deployment Complexity
User Resistance
Wrong Model Choice
Lack of Validation
Underfitting
Hyperparameter Tuning
Overlooking Model Explainability
Scalability Challanges
Lack of Conitnuous Monitoring
Non-compliance with Regulations
Ethical Missteps

Figure 1. Diagram representing the process of selecting prevalent and suitable data science pitfalls to demonstrate in the study materials. Not all pitfalls can be practically exemplified using the small-scale car domain, especially in the limited time frame of a practice session.

Underfitting was discarded due to being one of the least entertaining to demonstrate, as it simply results in bad driving models. Mistakes in hyperparameter tuning were discarded due to excessive computation time needed. Model explainability was discarded due to its complexity when working with neural networks (was considered too advanced for this course). Scalability challenges were deemed challenging and time-consuming to demonstrate in practice. Lack of continuous monitoring was discarded as it assumes deploying the solutions over a longer period of time. Non-compliance with regulations and ethical principles was discarded as these hardly apply in the context of small-scale cars. Additionally, two more problems were excluded from the list of problems that made it into the study materials. After experimenting to train a random forest regression algorithm, the problem of selecting and choosing wrong model types was deemed too time-consuming for this thesis and would not have resulted in interesting practical demonstrations. Lack of validation was excluded because it turned out to be difficult to vividly demonstrate this exact problem using toy cars. However, notice that a course using the practical materials created in this work can still cover all these left-out topics in theoretical materials and lectures.

The table below (Table 1) presents the final selection of issues.

Table 1. Data science problems addressed in the study materials, along with descriptions of how each problem is demonstrated and the corresponding lesson where it is addressed.

Problem	Description of how a problem is demonstrated in the study materials	Lesson
Insufficient data	The poor performance of a model trained on approximately 2500 frames of data (insufficient) is compared to the performance of a model trained on approximately 25 000 frames of data (sufficient).	Lesson 1: Data is the Most Important
Poor quality data	A model trained on data instances that have the steering angle labels manually corrupted is deployed on the track and performs poorly.	Lesson 2: Garbage in, Garbage out
Biased data	A model trained on data collected driving in clockwise direction deployed on the track and fails to drive in counter-clockwise direction.	Lesson 3: Generalization
Data drift and model drift	A model trained on data collected in daylight is deployed on the track in the dark or vice versa. Models perform poorly in novel light conditions.	Lesson 3: Generalization
Ineffective feature selection and overfitting	A model trained on data collected with irrelevant objects in the background is deployed on the track and performs poorly as soon as the background objects are removed or altered.	Lesson 4: Overfitting
Ignoring business metrics	A model is evaluated using open-loop[6] testing and closed-loop[7] testing. Open-loop testing does not actually directly evaluate the task we want the model to perform. (in here "business objective" is safe driving and "business metrics" are driving quality metrics).	Lesson 5: Evaluation
Integration issues and deployment complexity	A model trained on input images that are tripled in resolution is deployed on the track. The computations are too slow on our limited hardware and fail to achieve the objective of safe driving.	Lesson 5: Evaluation
User resistance	Students are asked if they would rather drive the toy cars themselves or let the self-driving model drive to complete 10 clean laps.	Lesson 5: Evaluation

---

[6] Open-loop testing evaluates the model's performance by comparing its output to the pre-recorded behaviour of the human expert driver without giving the model any control over the physical car [23].

[7] Closed-loop testing evaluates the model's performance in a realistic or simulated driving scenario by allowing the self-driving network to control the physical car [23].

Some problems were similar and vividly demonstrable in a single practical demonstration. Therefore, some demonstrations showcase many problems at once. The distribution of problems across the lessons is justified in the next chapter, where the learning objectives are defined.

2.2 Creating Study Materials Using the Backward Design Method

The backward design (BD) method [9] was used to develop study materials in this thesis.

Building upon the structure of the first stage of BD, the overarching objectives set for students were to comprehend key issues prevalent in the domain of data science, recognize these issues, pay attention to them, and avoid repeating them in the future. The study materials were created so that the students would be capable of explaining why the demonstrated shortcomings in the material arise. Upon completing the study materials, students should be able to address the following questions:

- Why a data science project might fail?
- What are some limitations of artificial intelligence (AI)?
- How can AI's limitations affect a data science project?
- What is important in evaluating and deploying a machine learning model?

To achieve the study objective of introducing common data science pitfalls, the study material was divided into 5 parts (lessons), with each lesson assigned a specific sub-objective aimed at helping students reach the general goals of the material:

1. The first lesson focuses on the importance of the data quantity in machine learning. The objective of this lesson is to teach how crucial it is to have a sufficient amount of input data in machine learning tasks.
2. The main aim of the second lesson is to explain the importance of data quality for machine learning tasks.
3. The objective of the third lesson is to make students understand the generalization problems commonly experienced when using machine learning models. By the end of the third lesson, students should be able to recognize, explain, and address generalization problems in machine learning.
4. The fourth lesson’s main goal is to make students recognise, understand and address the problem of overfitting in machine learning models.

5. The fifth lesson focuses on evaluating and deploying machine learning models. The main goals of this final lesson are to make students understand the importance of model evaluation, help them comprehend why it is crucial to critically think about the choice of the evaluation technique, and introduce students to some problems that might be encountered during the deployment of a machine learning model.

According to the task of the second stage in BD, assignments were created. These assignments were created in a form of reflective and analytical tasks that require short text answers. They were paired with practical tasks directly related to the study objectives, designed to help students comprehend the topics better. Each assignment was designed to be answered based on the observations from the practical tasks and additional theoretical information provided in the study material. Assignments were included with specific questions to answer as part of the answer. Digital Jupyter Notebook[8] worksheet was chosen as a suitable medium for solving and submitting tasks. Each worksheet was included with specific instructions on how to submit the assignments as a Jupyter Notebook file, with all the cells activated and visible output saved together with answer cells containing the required answers. Additionally, a guide for evaluating the assignments (the “key”) was created (see Appendix 2).

To achieve the objectives of the third stage of the BD model, self-driving toy cars were chosen to teach students about common pitfalls in data science, facilitated by Jupyter Notebook worksheets. These toy cars enable the practical demonstration of data science problems. The use of Jupyter Notebook files allows students to access the Donkey Car platform in Google Colab without the need to install the Donkey Car software on their personal computers. This streamlines the process of using the important features of the Donkey Car software for training new models, as the supervisor of the study materials does not have to address various problems that might arise during the installation of Donkey Car on different systems.

In addition to the practical exercises involving self-driving toy cars, texts briefly describing the discussed data science problems, code snippets, illustrative images and videos to help demonstrate these problems were included. The study materials were designed with the assumption that the target audience possesses basic programming knowledge. This implies that they should already be familiar with the Python programming language and using

---

[8] https://jupyter.org/

command-line tools, as these technologies are frequently utilised throughout the study materials. Additionally, the materials were created with the premise that students should have the opportunity to seek guidance while completing practical tasks. Answering the assignment questions on the worksheet is done without immediate assistance from the instructor.

In addition to practice session type of materials containing time-consuming tasks such as collecting new training data and training new self-driving neural networks, a shorter, less time-consuming study material was created. This shorter study material omits the time-consuming tasks of data collection and optimising machine learning models. Instead, pretrained self-driving models are provided, which can be used to demonstrate the failures of different approaches on the track. Each task of testing a pretrained model on the track is accompanied by an additional demonstrative video, showcasing the expected performance of the pretrained model on the track. These short course materials are intended for use in one-day or few-days workshops or continuous education courses. While these materials aim to achieve the same study outcomes, retention of knowledge might be lower due to less time spent on the tasks.

2.3 Data Gathering and the Track Setup

To create practical tasks involving self-driving toy cars, the tasks had to be solved beforehand. This ensured that the desired effects would appear when deploying the models. Additionally, these trained models were provided in some tasks of the study materials as downloadable pretrained models, so that students would not have to spend too long gathering new data and training a new model. Additionally, the models were used to film demonstration videos, showcasing the desired outcomes and serving as a fallback for unexpected situations that might arise while solving the practical tasks.

In this thesis, training data was collected by manually driving the toy car around the track (see Figure 3). Whenever the throttle value applied by the human driver was larger than 0, data was gathered automatically by the Donkey Car software at a rate of 20 frames per second. Constant throttle mode was used to drive the car, with its value adjusted so that the lap time would be approximately 15 seconds.

The recorded images had a resolution of 160x120 pixels. Recorded data was saved into folders called tubs. If necessary (training data contained crashes), the data tubs were edited using a built-in functionality of Donkey Car software called *tubclean* available via a web server (see Figure 2). This tool allowed to conveniently mark frames of data containing crashes or bad driving as “deleted”. Frames marked as “deleted” inside the tub were not actually deleted, just not used in other Donkey Car operations such as training the neural network, creating movies, etc.

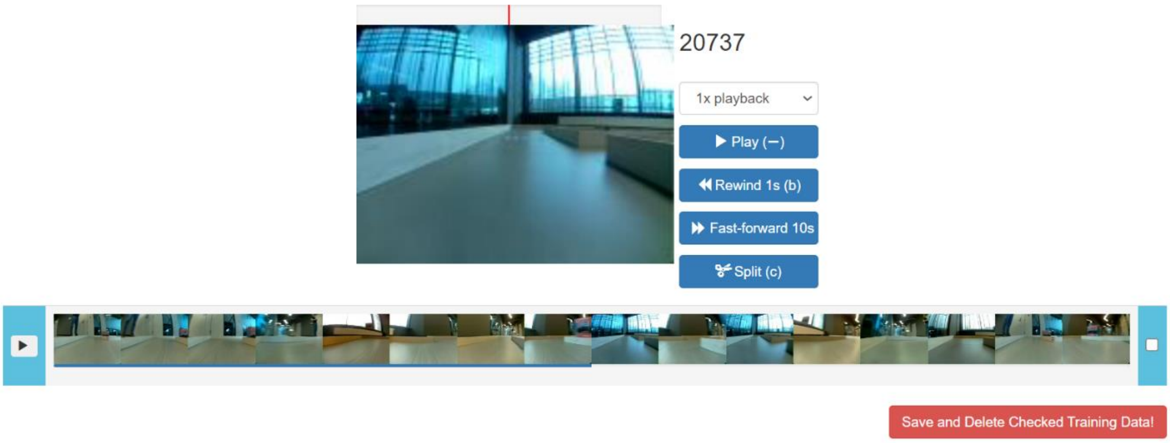


Figure 2. Web interface of a tool for cleaning data provided by the Donkey Car software.

The track used for collecting training data, testing the performance of the self-driving models, and filming demonstrative videos for the study materials is depicted in Figure 3. The track was constructed of wooden boards that outline the driving path. The surface of the track was free of any toys or other objects that might interfere with the driving path. While gathering data, three Donkey Car S1 boxes (original boxes in which the toy cars used in these study materials were shipped), coloured in red and blue, were placed around the track with the intent of allowing the models to overfit to the locations of these boxes.

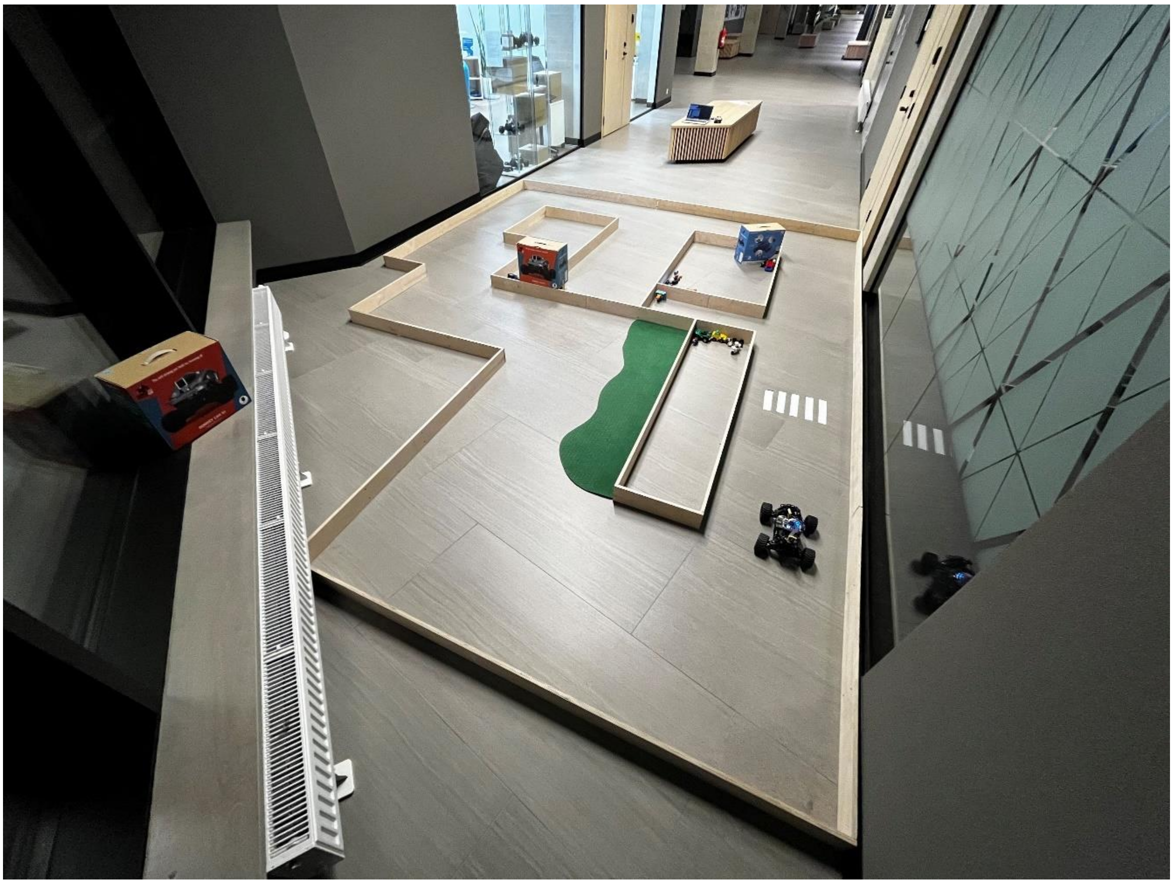


Figure 3. Track used for driving the toy cars.

The track resided next to a large window looking outside and two classrooms with big windows looking out onto the hallway. Lighting conditions at the track were influenced by the weather and lighting conditions outside, as well as by the ceiling lights in the hallway and lights in the classrooms next to the track.

All of the data was gathered by driving in a clockwise direction around the track. Due to the layout of the track, this results in a biased dataset that contains more right turns than left turns, as displayed in Figure 4.

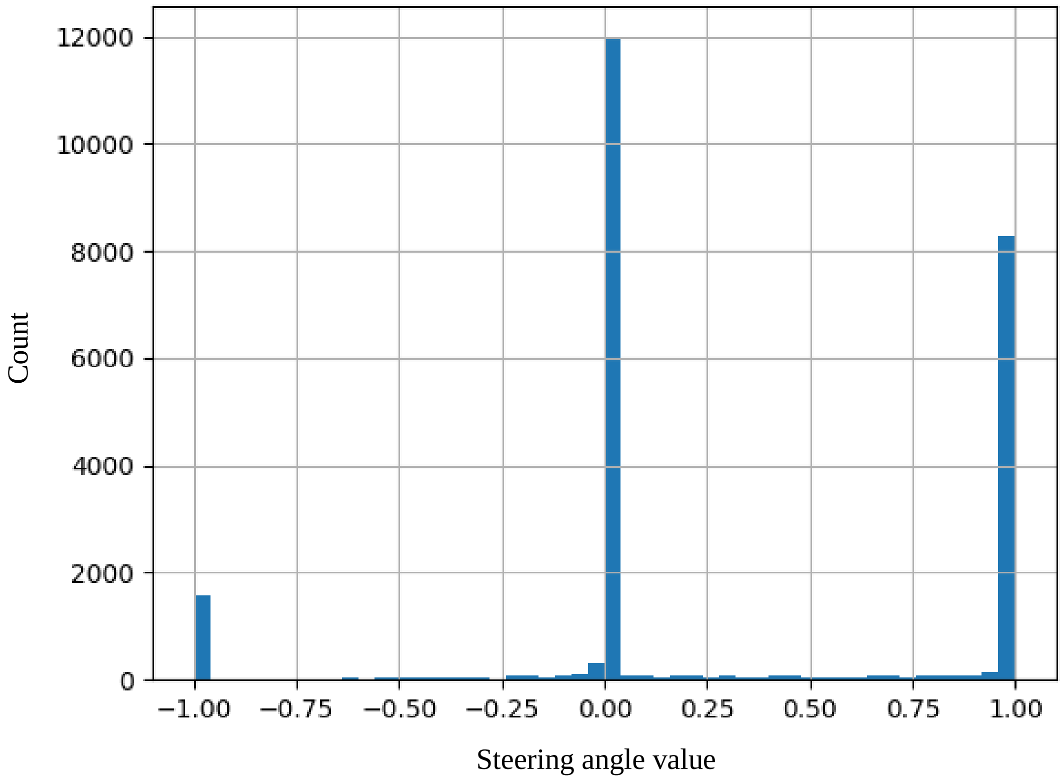


Figure 4. Distribution of steering angles in a dataset comprising 25 000 frames of clockwise driving. As displayed on the histogram, right turns (steering angle values higher than 0) are more represented in the dataset.

For certain tasks in the created study materials, pretrained models were provided for testing on the track. Since each pretrained model was intended to demonstrate a specific data science problem or solution on the track, it was crucial that the model performed as expected when tested by students. To help ensure this, the data was collected in three different lighting conditions: cloudy weather conditions during the day, sunny weather conditions during the day, and dark lighting conditions during the night. Approximately 25 000 frames of training data was gathered in each condition.

In addition to training the models, data was also gathered for open-loop validation purposes. For this, completely separate validation datasets consisting of approximately 6000 frames each were collected in the same three lighting conditions: cloudy weather during the day, sunny weather during the day, and dark lighting conditions during the night. Collected data is published with the study materials on Figshare (see Appendix 2). Detailed descriptions of all the gathered datasets can be found in Appendix 3.

2.4 Model Training

Various different self-driving models had to be trained and tested on the track to validate the practical tasks, film demonstration videos, and provide compliant pretrained models for the study materials. A neural network architecture predefined in the Donkey Car software was predominantly used, but modifications were made to achieve task-specific results for some self-driving models. The code changes made to the Donkey Car software for training the pretrained models are stored in a GitHub repository [24]. Training was conducted using the Donkey Car software on a personal computer or in the Google Colab environment, as the toy car's Raspberry Pi 4B lacks the computational power to train the models sufficiently.

Donkey Car software provides users with several built-in neural network architectures. In the study materials, all models are trained using a predefined neural network architecture called Keras Linear, with modifications applied as needed. Keras Linear can process incoming camera images and generate corresponding throttle values and steering angles. The neural network consists of an input layer, 5 convolutional layers, followed by two fully connected layers, and ending with a fully connected output layer. The network uses convolutional neural networks (CNNs) for image processing and fully connected layers for regression. The term "Linear" in the model's name describes the nature of the final network layer, which comprises two neurons responsible for outputting two continuous values: one for steering and one for throttle. No activation function is applied, and this final layer simply performs a linear combination (weighted sum) of its inputs. During training, the neural network adjusts its parameters iteratively to minimise the difference between its predictions and the ground truth throttle values and steering angles. According to the Donkey Car documentation [10], this model architecture may struggle to learn to predict throttle values. However, in this thesis, a constant throttle value was used when testing the trained self-driving models. Students are also guided to do so in the study materials. Despite not utilising the model's ability to predict throttle values, the model was trained to do so anyway due to using the default setup.

To train self-driving models with increased robustness and reduce the risk of overfitting to irrelevant features visible above the walls of the track, the default neural network architecture was modified by adding a Cropping2D[9] layer from the Keras library in front of the CNN layers. This added layer was adjusted to cut the input image by 50 pixels from the top before passing it to the CNN layers. Additionally, an alternative method of cropping the input image was tested. The Donkey Car software provides a transformation function, which, according to the Donkey Car documentation [10], should mask the input image by desired dimensions. However, when testing this solution, it did not seem to give the desired results.

To demonstrate a self-driving model that would be too computationally demanding for the Raspberry Pi 4B onboard the toy car, a layer to increase the size of the input image was added before passing it to the CNN layers. To resize the input image from 160x120 pixels to 480x360 pixels, the Resizing[10] layer predefined in the Keras library was used. By increasing the image resolution threefold, the closed-loop testing results on the track demonstrated the desired outcome where the Raspberry Pi was unable to process input images quickly enough.

To train a model on "garbage" data (incorrectly labelled data), the default unmodified Keras Linear neural network architecture was used. Modifications were made to the input training data by inserting a custom function call into the Donkey Car source code right before starting the training process. This function modifies the data read from the data tub (does not overwrite the data tub itself), simulating common mistakes that may occur during data entry, pre-processing, or data collection. Following modifications were made to the training data before training the neural network:

- For 25% of the training data, the steering angle was multiplied by -1 to simulate scenarios where a minus sign has been accidentally inserted or omitted during data entry.
- For 25% of the training instances, the original steering angle was replaced by 0 to simulate situations where missing values in the dataset would be replaced by zeros.
- For 25% of the training instances, the original steering angle's decimal place was shifted up by one spot (the original value was multiplied by 10) which would simulate scenarios where decimal points have been misplaced, resulting in steering angle values being ten times larger than intended.

---

[9] https://keras.io/api/layers/reshaping_layers/cropping2d/

[10] https://keras.io/api/layers/preprocessing_layers/image_preprocessing/resizing/

- The remaining 25% of the training data had the original steering angle replaced by the steering angle attached to the previous frame, which simulates scenarios where copying data between columns has led to shifted columns in the dataset.

To enhance the robustness of all pretrained models, image augmentation was employed during training [25]. Image augmentation is utilised to introduce more variability into the training data. Since the models are particularly sensitive to various lighting conditions, brightness augmentation was applied. This technique involves increasing and decreasing the brightness of random training images in the dataset, with the aim of creating greater diversity in the training data and producing a model that is better able to generalise to different lighting conditions.

2.5 Model Testing

To verify that the training procedures and the trained models are suitable for the study materials, they were evaluated using closed-loop testing. This approach was used because, as stated by Codevilla et al. [26], it gives a much better understanding of the model's ability to drive compared to the open-loop testing. Although open-loop testing was conducted as part of the model training pipeline, and it is also a topic covered in the study materials, by calculating the mean absolute error (MAE) across all the steering angle predictions made on the validation dataset, it was not used for evaluating the models.

Each trained model was evaluated on a single 10 lap test run while trying to maintain a speed of approximately 15 seconds per lap as during the data collection phase. Each model's interventions and infractions were counted and used to calculate a score (see formula (1)) that provides insight into the frequency of mistakes made by the car during a single lap, where a lower score indicates a better-performing model. An intervention occurs when the car makes a severe driving mistake, necessitating human intervention to place it back on track. Conversely, an infraction refers to a driving mistake where the car hits the track wall but can continue driving without human intervention.

$$SCORE = \frac{INFRACTIONS \times 0.5 + INTERVENTIONS}{10} \tag{1}$$

Models trained on data collected in various different lighting conditions were tested in three different lighting settings: cloudy daylight, sunny daylight, and dark lighting conditions. Rest of the models were only tested in the lighting conditions that they were trained in.

Testing results can be observed in the Appendix 4. These tests validated that the models expected to fail due to machine learning mistakes indeed perform weaker and the designed tasks put in evidence the desired pitfalls.

3 Results

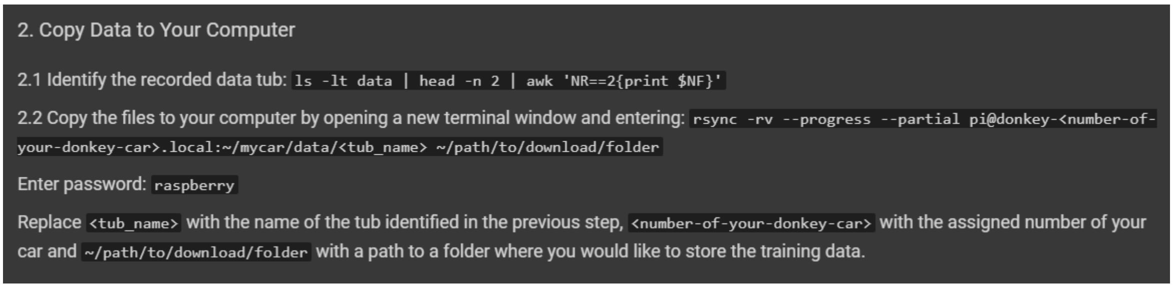
This chapter showcases the created study materials with qualitative and quantitative descriptions. Additionally, feedback of the experts on the study materials is presented.

3.1 Study Materials

This section describes the created study materials' components and volume.

3.1.1 Qualitative Description

As a main result, 5 practical Jupyter Notebook worksheets for use in the Google Colab environment were created (see Appendix 2). These consist of code cells and longer text passages that explain both the broader theoretical background and the importance of the tasks at hand, as well as instructions for completing the tasks (Figure 5) and submitting the assignments (Figure 6). Text cells are also used for collecting students' answers (Figure 7).



2. Copy Data to Your Computer

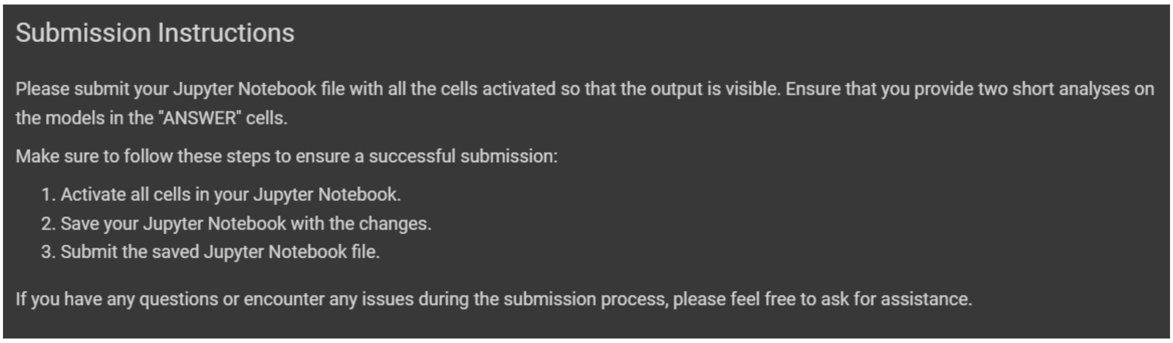
2.1 Identify the recorded data tub: `ls -lt data | head -n 2 | awk 'NR==2{print $NF}'`

2.2 Copy the files to your computer by opening a new terminal window and entering: `rsync -rv --progress --partial pi@donkey-<number-of-your-donkey-car>.local:~/mycar/data/<tub_name> ~/path/to/download/folder`

Enter password: `raspberry`

Replace `<tub_name>` with the name of the tub identified in the previous step, `<number-of-your-donkey-car>` with the assigned number of your car and `~/path/to/download/folder` with a path to a folder where you would like to store the training data.

Figure 5. Instructions for completing a subtask. Here, the student is instructed to copy gathered data from the Raspberry Pi on the Donkey Car to their personal computer as part of a larger task.



Submission Instructions

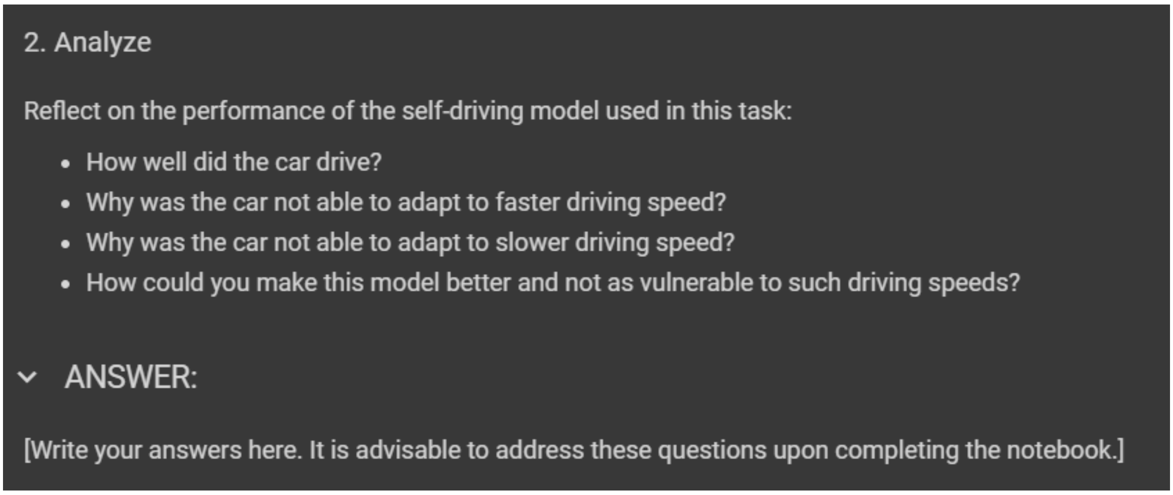
Please submit your Jupyter Notebook file with all the cells activated so that the output is visible. Ensure that you provide two short analyses on the models in the "ANSWER" cells.

Make sure to follow these steps to ensure a successful submission:

1. Activate all cells in your Jupyter Notebook.
2. Save your Jupyter Notebook with the changes.
3. Submit the saved Jupyter Notebook file.

If you have any questions or encounter any issues during the submission process, please feel free to ask for assistance.

Figure 6. Instructions for submitting the assignments at the end of each lesson.



2. Analyze

Reflect on the performance of the self-driving model used in this task:

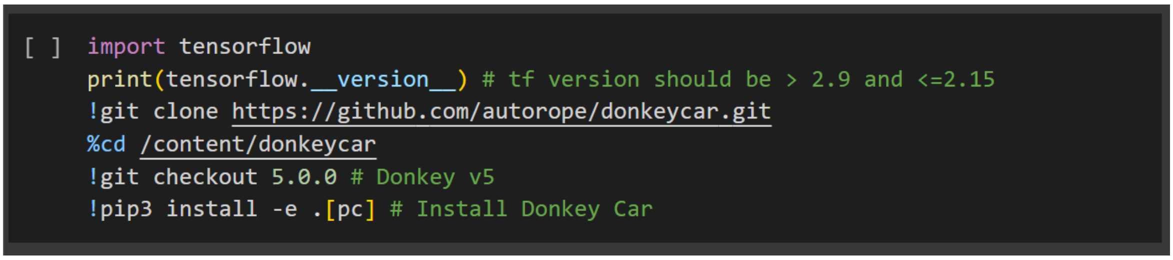
- How well did the car drive?
- Why was the car not able to adapt to faster driving speed?
- Why was the car not able to adapt to slower driving speed?
- How could you make this model better and not as vulnerable to such driving speeds?

ANSWER:

[Write your answers here. It is advisable to address these questions upon completing the notebook.]

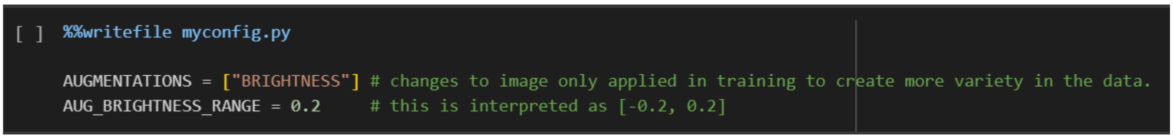
Figure 7. Example of a text cell used for collecting students' answers to the analysis assignments.

The code cells are divided into those that require activating and those that do not necessarily have to be activated. Cells that have to be activated are divided into cells that install and import the necessary software (Figure 8), cells that do not require any modifications but are essential for completing the task (Figure 9), and cells where the learner must provide the correct values for specified variables for completing the task (Figure 10).



```
[ ]  import tensorflow
     print(tensorflow.__version__) # tf version should be > 2.9 and <=2.15
     !git clone https://github.com/autorope/donkeycar.git
     %cd /content/donkeycar
     !git checkout 5.0.0 # Donkey v5
     !pip3 install -e .[pc] # Install Donkey Car
```

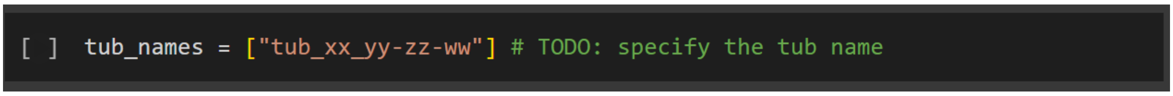
Figure 8. Code cell for importing Tensorflow and installing Donkey Car software.



```
[ ]  %%writefile myconfig.py

     AUGMENTATIONS = ["BRIGHTNESS"] # changes to image only applied in training to create more variety in the data.
     AUG_BRIGHTNESS_RANGE = 0.2     # this is interpreted as [-0.2, 0.2]
```

Figure 9. Code cell adding brightness augmentation to training data by writing specified lines to the *myconfig.py* file when activated.



```
[ ]  tub_names = ["tub_xx_yy-zz-ww"] # TODO: specify the tub name
```

Figure 10. Code cell that has to be modified by specifying the correct training data folder name.

Students do not have to activate code cells that are hidden and where the output is already visible. The purpose of such cells is to display a YouTube video (Figure 11), visualise images on a plot (Figure 12), or display a multiple-choice question (Figure 13).

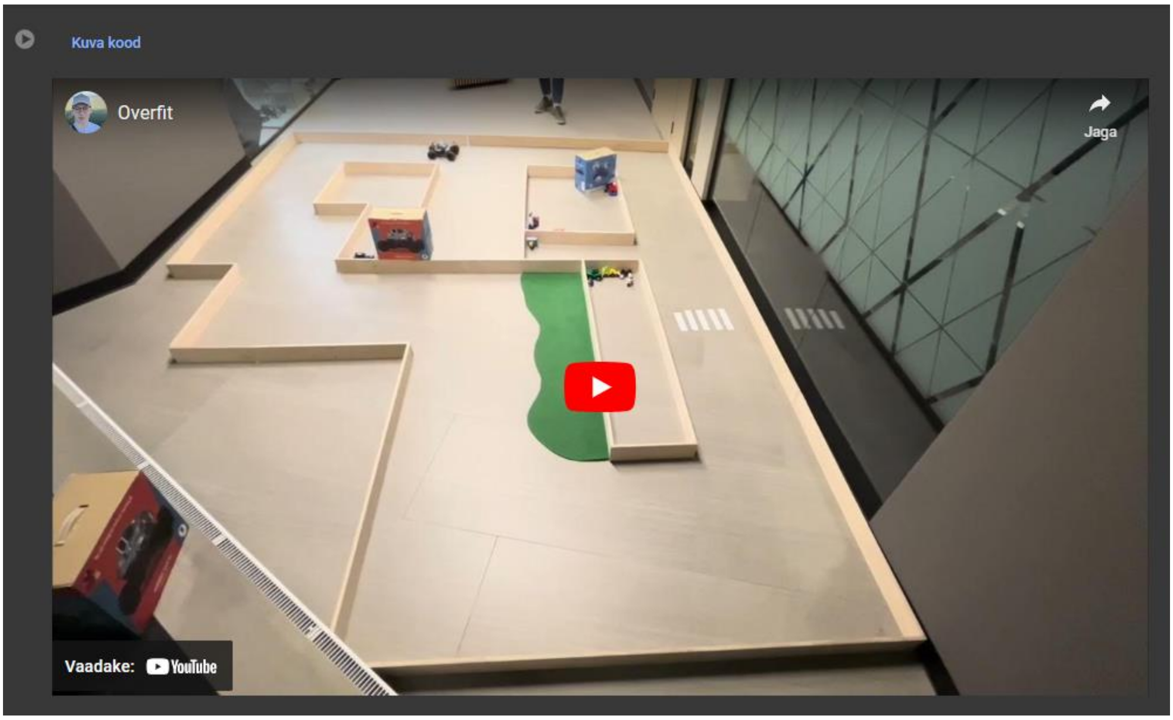


Figure 11. Hidden code cell with visible output saved displaying a YouTube video demonstrating the effect of overfitting.

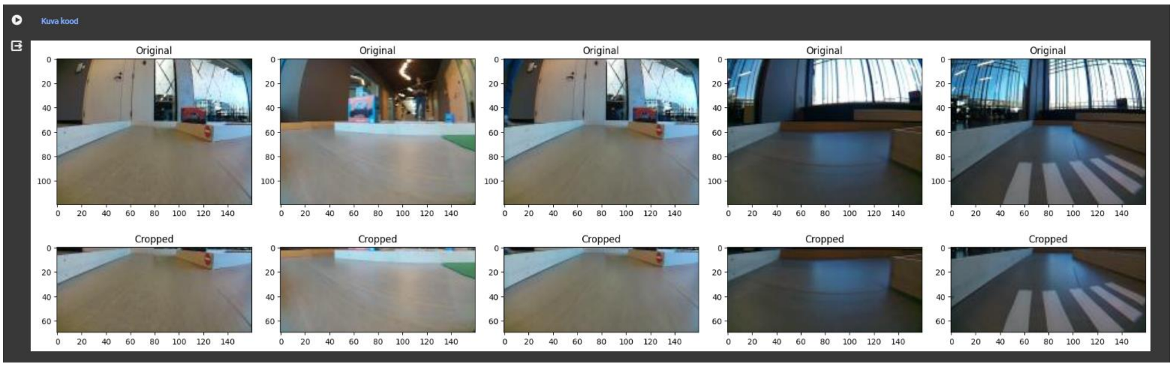
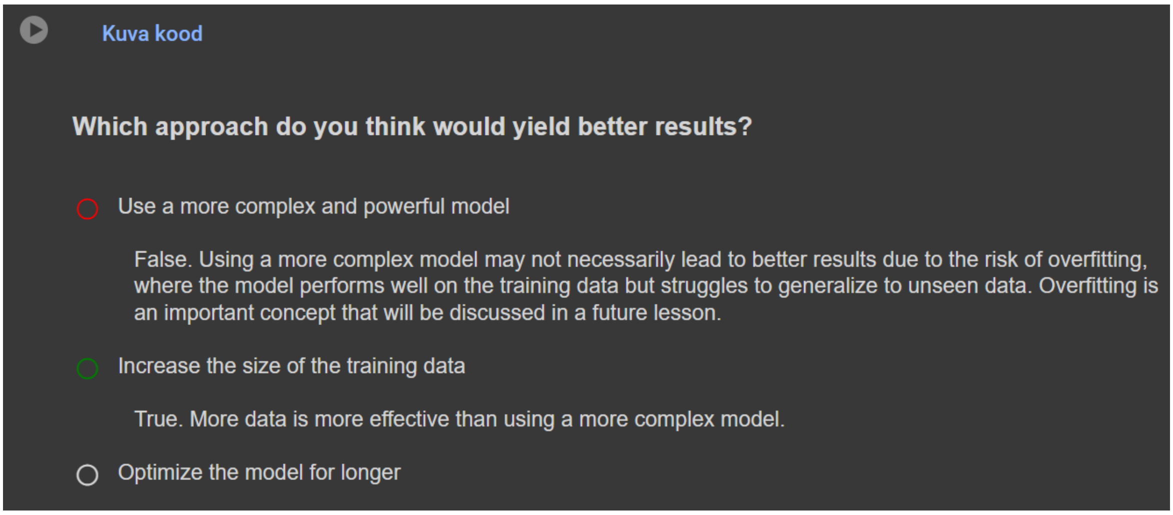


Figure 12. Hidden code cell with saved visible output displaying a plot that visualises the effect of cropping 50 pixels off the top of input images.



Which approach do you think would yield better results?

○ Use a more complex and powerful model

False. Using a more complex model may not necessarily lead to better results due to the risk of overfitting, where the model performs well on the training data but struggles to generalize to unseen data. Overfitting is an important concept that will be discussed in a future lesson.

○ Increase the size of the training data

True. More data is more effective than using a more complex model.

○ Optimize the model for longer

Figure 13. Hidden code cell with saved visible output displaying a multiple-choice question. This question is not an assignment and serves as a way to give students instant feedback. Clicking on each option reveals if the answer is correct, together with an explanation.

Practical tasks include training the self-driving models in Google Colab. To help speed up the process, the study materials include instructions on how to utilise GPU in Google Colab environment for training. This is necessary because the training process can be time-consuming, mainly depending on the size of the training data and neural network architecture. Since the training process is very time-consuming, the training of different models has been spread between the lessons as evenly as possible considering the tasks and study objectives of each lesson. Due to some tasks requiring data gathered in drastically different lighting conditions, some practical tasks are provided with appropriate pretrained self-driving models or pre-gathered datasets.

In addition to 5 separate lessons with practical and time-consuming tasks, a shorter alternative version of the study materials was created where the practical tasks were modified so that they do not require any data gathering or model training. Instead, students are supplied with pretrained models for each practical task that they can test out on the track. Practical model demonstrations on the track are paired with demonstrative videos of the desired model behaviour on the track. The concepts addressed in the shorter version of the study materials match with the more time-consuming version.

Since all the models in the study materials are trained using Donkey Car version 5.0, the toy car's Raspberry Pi must also be able to run a Donkey Car version >= 5.0. Together with the study materials, an SD card image file is provided, containing a compatible combination of Rasbian[11] and Donkey Car software, which can be easily installed on other Donkey Car S1

---

[11] https://www.raspbian.org/

toy cars. The provided image has been tested to be compatible with the models trained using Donkey Car version 5.0 either in Google Colab or a local environment.

3.1.2 Quantitative Description

Quantitative description of created worksheets is provided in Table 2.

Table 2. Quantitative description of the worksheets. In parentheses is the number of cells that must be activated.

Lesson	Number of practical tasks	Number of assignments	Number of code cells	Number of pages of text in text cells (500 words/page)
1: Data is the Most Important	2	2	12 (11)	7.0
2: Garbage in, Garbage out	1	1	8 (8)	3.4
3: Generalization	4	4	13 (12)	6.6
4: Overfitting	2	2	21 (18)	3.1
5: Evaluation	2	1	7 (7)	4.9
All lessons combined	11	10	61 (56)	25.0
Shortened course	12	9	18(0)	23.4

The lessons vary in size but efforts were made to keep the them as similar in size as possible while maintaining a logical structure in the study materials. The most time-consuming parts of the materials are the practical tasks. Notably, not all practical tasks require the same amount of time, as some do not involve collecting new data and training a new model, which can be very time-consuming. The exact time consumption of each lesson has not been measured on students.

3.2 Expert Feedback

To get some valuable feedback about the created study materials, two data science experts working in the University of Tartu were asked to evaluate the created study materials. Expert 1 is a junior researcher in the field of machine learning. Expert 2 is an associate professor in the field of autonomous driving. They were provided with both (the shorter and more time-consuming) versions of the study materials together with the answers to the assignments in a separate file. The experts did not perform any practical tasks and only read the materials and watched the videos provided in the materials. They were asked to assess various aspects about the structure, content and tasks of the study materials on a scale of 1 to 10. The exact evaluated aspects and the grades given by each expert can be seen in Appendix 5. Additionally, both experts provided qualitative feedback for the materials.

Feedback on the structure of the study materials was positive. The experts found that the learning objectives were clearly stated, the materials were sufficient to achieve the stated learning objectives, the progression and structure of the study materials were logical, and the text was generally clear and understandable. Experts pointed out mistyped words and gave suggestions to improve the clarity and wording of some sentences. The mistakes regarding spelling and wording have been corrected according to the feedback.

The content of the study materials was also evaluated positively. The choice of topics was deemed sufficient, and no factual mistakes regarding the covered topics were identified. According to the experts, the study materials felt complete but could include more and wider theoretical background information about the covered topics. This supports the claim that the created materials do not serve as an independent and comprehensive course but need to be supported by additional materials that are still in need of development. The demonstration videos filmed for the study materials were found helpful and useful for understanding the topics better.

The instructions for completing the tasks were considered thorough and easily followable for intended learners, but some instructions could have been clearer. The instructions have been improved based on the provided feedback, as comprehensibility of the tasks is crucial for the study materials. The assessments in the study materials were judged to align well with the learning objectives. Assessments were considered effective for evaluating the students' retention of new knowledge and providing meaningful feedback to the learners.

Overall, the received feedback was largely positive. No crucial mistakes or shortcomings were identified. The stated learning objectives were deemed relevant and sufficient. Experts' feedback suggests that chosen learning methods are effective to help students achieve these objectives.

4 Discussion

This chapter describes the strengths and limitations of the created study materials, as well as possible ways to further develop the created materials.

The objective of this thesis was to develop study materials that address common failure points in data science by utilising self-driving toy cars providing students with a practical and engaging way to experience these pitfalls. This objective was achieved by applying the backward design method in designing the materials. The Donkey Car platform was utilised to offer students practical tasks to retain the knowledge better through practical experiences. The learning process was facilitated by Jupyter Notebook format in Google Colab, which relieves the students from the need to install additional software on their personal computers to complete practical tasks. This not only saves time for students but also spares the supervisor from troubleshooting installation problems across different computer environments. All the practical tasks were solved and tested while developing the materials to assure that exercises have an outcome that is clearly connected to the learning objectives. Furthermore, the tasks were tested in various lighting conditions to assure that practical tasks are solvable in different conditions.

In addition to five separate worksheets containing time-consuming practical tasks, a shorter version of the study materials was created. The alternative shorter version of the study materials provides a quicker way of achieving the learning objectives. The material is supported by 14 videos that demonstrate desired performance of certain self-driving models. These videos are especially useful if testing the models on the track is not possible. This shorter version of the materials can also serve as a summary for students that have already completed the more time-consuming version of the study materials.

The study materials contain assignments connected to the learning objectives that need to be completed by the students. An evaluation guide containing expected correct answers that should ease the work of evaluating the assessments was also created as part of the thesis.

Although the study materials were created on a specific toy car driving track located in the University of Tartu Delta Centre, it is possible to adapt the study materials to be used on any track layout. Of note, performing practical tasks on a track with different layout would mean that the pre-recorded data and pretrained models would not give desired results.

The created study materials have received positive feedback about its structure, content, and tasks from two experts in the field of data science. Experts considered the study materials effective in achieving defined learning objectives. Since the defined learning objectives were quite broad and not limited to self-driving toy cars, we assume the learners can generalize retained knowledge to other fields of data science beyond just training self-driving neural networks through behavioural cloning.

4.1 Limitations and Future Work

The created study materials have limitations, some regarding the use of the study materials and some regarding future work.

Since the open-source Donkey Car software relies on various other programming libraries, it is important to ensure its compatibility with the environments installed on the Raspberry Pi and Google Colab. In this work, Donkey Car version 5.0 was used on the Pi as well as in Google Colab. However, future updates to Google Colab might necessitate updating the Donkey Car version used in that environment. Such updates could also require updating the Donkey Car version (and possibly the virtual environment) installed on the Pi to drive the trained self-driving models on the track.

Another potential challenge in using the study materials involves training the models in Google Colab. While it is convenient to avoid the struggles of installing the Donkey Car software on students' local computers, training the desired models might be slow in Google Colab, especially when the user has reached the limit on using the T4 GPU provided for free in the environment. The study materials do not specifically provide tailored instructions on completing tasks in the host computer's local environment in case students wish to complete time-consuming training tasks locally. However, the completion steps are analogous to those described in the study materials and comprehensive installation instructions are provided in the Donkey Car documentation.

While efforts were made to train self-driving models capable of performing adequately under various lighting conditions, there may still be instances where the pretrained models do not drive as expected on the track. Possible reasons for these occurrences include sharp shadows on the track's surface in the mornings (due to sunlight angle), ceiling lights being turned on in surrounding classrooms, or other unexpected lighting conditions. Another reason for unexpected performance when deploying a pretrained model might be hardware differences between the toy cars (e.g. different camera angle). Should such situations arise, the shorter version of the study materials, which involves tasks using pretrained models, includes videos demonstrating the expected performance of the model. In the above cases, collecting a new dataset and training a new model should bypass these data drift type of problems, only pre-trained models and pre-collected datasets are affected.

Regarding the two alternative study materials created in this thesis, it is essential to note that it is recommended to use the more time-consuming version consisting of five separate worksheets with data gathering and model training tasks. The purpose of the shorter study material is to provide a quicker way of achieving the study objectives. However, due to having fewer practical tasks, the student's retention of new knowledge might not be as effective compared to the version containing more time-consuming practical tasks. The shorter version can serve as a useful summary of the tasks and data science pitfalls addressed in the five separate practical worksheets.

Concerning the evaluation guide for marking the analytical assignments, it is important to state that since some questions are very open-ended, it is also up to the evaluator to determine whether the answer meets the criteria or not. This may also depend on expected level of proficiency of the students.

The created study materials are not a comprehensive course on common problems in data science. Some issues did not make it into the materials and need to be covered with additional lectures and theoretical materials. The study materials serve as practical materials for a potential 3 ECTS course that needs developing. Additionally, to further improve the study materials, it is essential to test the study materials and collect feedback from students to gain better insight about the time-consumption, clarity and effectiveness of the study materials.

List of References

[1] Ng A. Machine Learning Yearning. 2018. https://wordpress.deeplearning.ai/wp-content/uploads/2022/03/andrew-ng-machine-learning-yearning.pdf (18.04.2024)

[2] Mitchell T. M. Machine Learning. New York: McGraw-Hill. 1997.

[3] Domingos P. A Few Useful Things to Know about Machine Learning. *Communications of the ACM*, 2012, 55(10), pp. 78-87. https://dl.acm.org/doi/10.1145/2347736.2347755 (18.04.2024)

[4] Chakarov A., Nori A., Rajamani S., Sen S., Vijaykeerthy D. Debugging Machine Learning Tasks. 2016. https://arxiv.org/abs/1603.07292 (18.04.2024)

[5] OpenAI (2022). ChatGPT 3.5 (05.05.2024 version): https://chat.openai.com.

[6] Kolb D. A. Experiential Learning: Experience as the source of Learning and Development Second Edition. FT press, 2014.

[7] Hao Q., Barnes B., Jing M. Quantifying the effects of active learning environments: separating physical learning classrooms from pedagogical approaches. *Learning Environments Research*, 2020, 24(1), pp. 109-122. https://qhao.info/downloads/LER-2020.pdf (04.12.2023)

[8] Li D., Auerbach P., Okhrin O. Towards Autonomous Driving with Small-Scale Cars. 2024. https://arxiv.org/abs/2404.06229 (18.04.2024)

[9] Wiggins G., McTighe J. Understanding by design. Alexandria: Ascd. 2005.

[10] Donkey Car. https://docs.donkeycar.com/ (13.04.2024)

[11] Sammut C. Behavioral Cloning. In: Sammut C., Webb G. I. (eds). *Encyclopedia of Machine Learning*. Boston, MA: Springer, 2011, pp. 93-97. https://doi.org/10.1007/978-0-387-30164-8_69 (05.05.2024)

[12] Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge, Massachusetts: The MIT Press. 2016.

[13] Roosild K. Driving Speed as a Hidden Factor Behind Distribution Shift. University of Tartu Institute of Computer Science MSc thesis. 2022. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=75358&language=en (11.04.2024)

[14] Uduste I. Effect of Delays/Lag and Fighting it in Self-driving Neural Networks. University of Tartu Institute of Computer Science MSc thesis. 2022. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=74970&language=en (11.04.2024)

[15] Camara M. G. Using Adversarial Defense Methods to Improve the Performance of Deep-Neural-Network-Controlled Automatic Driving Systems. University of Tartu Institute of Computer Science MSc thesis. 2022. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=74730&year=2022 (11.04.2024)

[16] Tampuu A., Roosild K., Uduste I. The Effects of Speed and Delays on Test-Time Performance of End-to-End Self-Driving. *Sensors*, 2024, 24(6), p. 1963. https://doi.org/10.3390/s24061963 (05.05.2024)

[17] Wirth R., Hipp J. CRISP-DM: Towards a Standard Process Model for Data. *In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, 2000, 1, pp. 29-39. https://www.cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf (05.05.2024)

[18] OpenAI (2023). ChatGPT 4 (22.11.2023 version): https://chat.openai.com.

[19] Abdumalikov R., Acikalin A., Rubino G. self-driving-donkey-car. 2022. https://github.com/rabdumalikov/self-driving-donkey-car (11.04.2024)

[20] Bertin P. Autonomous vehicle project. 2022. https://medium.com/@paulux.bertin/autonomous-vehicle-project-1f3dae08529c (11.04.2024)

[21] Camara M. Self-driving competition Delta-X. 2022. https://github.com/mikecamara/self-driving-competition-delta-x (11.04.2024)

[22] Yu F. Techniques to Tackle Overfitting and Achieve Robustness for Donkey Car Neural Network Self-Driving Agent. 2019. https://flyyufelix.github.io/2019/06/04/pixmoving-hackathon.html (11.04.2024)

[23] Tampuu A., Matiisen T., Semikin M., Fishman D. Muhammad N. A Survey of End-to-End Driving: Architectures and Training Methods. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, 33(4), pp. 1364-1384. https://arxiv.org/abs/2003.06404 (06.05.2024)

[24] Kreegipuu A. Donkey Car Course Code. 2024. https://github.com/ArturKreegipuu/donkey-car-course/tree/main (05.05.2024)

[25] Shorten C., Khoshgoftaar T. M. A survey on image data augmentation for deep learning. *Journal of big data*, 2019, 6(1), pp. 1-48. https://link.springer.com/article/10.1186/s40537-019-0197-0?code=a6ae644c-3bfc-43d9-b292-82d77d5890d5 (12.05.2024)

[26] Codevilla F., Lopez A. M., Koltun V., Dosovitskiy A. On Offline Evaluation of Vision-based Driving Models. *In Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 236-251. https://arxiv.org/abs/1809.04843 (05.05.2024)

Appendix

I. List of Data Science Pitfalls Generated by ChatGPT 4

The full response generated by the OpenAI's ChatGPT 4 (November 22, 2023) to the following prompt: "You are an experienced machine learning engineer with many deployed solutions. Could you list the common failure points of machine learning solutions, where things can and often go wrong for teams. List them in the order of development cycle of the solution, first listing failures and challenges related to data and finally challenges related to deployment."

Certainly! As a machine learning engineer, I can outline the common failure points in the lifecycle of a machine learning solution. These failure points, listed in the order they typically arise from data acquisition to deployment, are critical to recognize and address for successful implementation:

1. **Data Collection and Quality Issues:**
    - **Insufficient Data:** Lack of enough data to train the model effectively.
    - **Biased Data:** The data may not be representative of the real-world scenario it's meant to model, leading to biased outcomes.
    - **Poor Quality Data:** Issues like missing values, noise, and errors in the data can severely affect model performance.
    - **Data Drift:** Changes in data over time that the model was not trained to handle.
2. **Data Processing and Feature Engineering:**
    - **Ineffective Feature Selection:** Choosing irrelevant or redundant features can decrease model performance.
    - **Overfitting:** Creating a model too complex for the data, making it perform well on training data but poorly on unseen data.
    - **Underfitting:** Conversely, a model too simple might not capture the underlying patterns in the data.
3. **Model Selection and Training:**
    - **Wrong Model Choice:** Selecting an inappropriate model for the task at hand.
    - **Hyperparameter Tuning:** Inadequate tuning can lead to suboptimal performance.
    - **Lack of Validation:** Not using proper validation techniques can lead to misleading assessments of a model's performance.
4. **Model Evaluation:**
    - **Ignoring Business Metrics:** Focusing solely on model metrics like accuracy, without considering how it impacts business goals.

- **Overlooking Model Explainability:** Failing to provide insights into how the model makes decisions, which is crucial for user trust and regulatory compliance.

5. **Integration and Scalability:**
   - **Integration Issues:** Difficulties in integrating the model with existing systems and workflows.
   - **Scalability Challenges:** The model may not scale well with increased data or usage, affecting performance and reliability.
6. **Monitoring and Maintenance:**
   - **Model Drift:** The model's performance may degrade over time as the real-world scenario it was trained to represent changes.
   - **Lack of Continuous Monitoring:** Failing to continually monitor the model's performance can lead to unnoticed deterioration.
7. **Deployment and User Adoption:**
   - **Deployment Complexity:** Challenges in deploying the model into a production environment.
   - **User Resistance:** End users might be reluctant to trust or adopt the model, especially if its decisions are not transparent or understandable.
8. **Regulatory and Ethical Considerations:**
   - **Non-compliance with Regulations:** Failing to adhere to data privacy and other regulatory requirements.
   - **Ethical Missteps:** Not considering or addressing the ethical implications of the model's use and its decisions.

Addressing these points requires a combination of technical skill, thorough planning, and an understanding of the broader context in which the model operates. Regular review and adaptation are key to navigating these challenges successfully.

II. Created Study Materials

Links to the created Jupyter Notebook files in Google Drive:

Lesson 1: Data is the Most Important:

https://colab.research.google.com/drive/1MDQPtZuRmSC-fWNefgqBzKTtk4r9aGfR?usp=drive_link

Lesson 2: Garbage in, Garbage out:

https://colab.research.google.com/drive/1H_quEPa9QZmwt-NRCMwaLegst9tuajjC?usp=drive_link

Lesson 3: Generalization:

https://colab.research.google.com/drive/1NjvMyczDuak0fbGzual4nCIojcCm13C5?usp=drive_link

Lesson 4: Overfitting:

https://colab.research.google.com/drive/1P0ttwcueRawhWCv2Mb6dZxOC_BxFhsI7?usp=drive_link

Lesson 5: Evaluation:

https://colab.research.google.com/drive/1LgW54886JxuXerK5GSj9EQSOvWpudYbS?usp=drive_link

Short version of the study material:

https://colab.research.google.com/drive/1oM1YbZnhPKxvh8QvehKbWjP_MwoxGeHh?usp=drive_link

Evaluation guide for marking the answers:

https://docs.google.com/document/d/16Oimkm1fIJUavNDxNSfR9SzIfhwAcDNWg6cCj3dUmc8/edit?usp=drive_link

Study materials are also published on Figshare:

https://figshare.com/articles/online_resource/Donkey_car_course_zip/25752162

III. Gathered Data Description

Table 3. Sizes of datasets collected in cloudy daylight conditions, in sunny daylight conditions, and in the dark lighting conditions for training and evaluation purposes.

Purpose	Conditions	Number of frames
training	sunny	24 023
training	cloudy	24 402
training	dark	23 243
open-loop evaluation	sunny	6725
open-loop evaluation	cloudy	6956
open-loop evaluation	dark	5994

IV. Model Testing Results

Testing results for self-driving models trained in the sunny conditions during the day are shown in the Table 4, testing results for self-driving models trained in the cloudy conditions during the day are shown in the Table 5, testing results for self-driving models trained during the night are shown in the Table 6, and testing results for models trained on combined dataset of data collected in cloudy conditions during the day, sunny conditions during the day and dark conditions during the night are shown in the Table 7.

Table 4. Self-driving models trained on data collected in sunny daylight conditions testing results in sunny daylight conditions.

Model	Frames	Infractions	Interventions	Score	MAE
default	2410	3	13	1.3	0.204
default	24 023	2	0	0.1	0.130
garbage	24 023	1	39	3.9	0.458
cropped	24 023	0	1	0.1	0.116
high-res	24 023	1	32	3.2	0.166

Table 5. Self-driving models trained on data collected in cloudy daylight conditions testing results in cloudy daylight conditions.

Model	Frames	Infractions	Interventions	Score	MAE
default	2587	2	8	0.9	0.189
default	24 402	1	0	0.05	0.137
garbage	24 402	2	35	3.6	0.700
cropped	24 402	6	0	0.3	0.154
high-res	24 402	0	34	3.4	0.157

Table 6. Self-driving models trained on data collected in dark lighting conditions testing results in dark lighting conditions.

Model	Frames	Infractions	Interventions	Score	MAE
default	2418	2	8	0.9	0.860
default	23 243	1	0	0.05	0.144
garbage	23 243	2	35	3.6	0.725
cropped	23 243	2	0	0.1	0.142
high-res	23 243	0	34	3.4	0.149

Table 7. Self-driving models trained on data collected in cloudy daylight conditions, in sunny daylight conditions, and in the dark lighting conditions testing results in different lighting conditions.

Model	Frames	Infractions	Interventions	Score	MAE	Conditions
default	71668	1	2	0.25	0.113	sunny
default	71668	0	1	0.1	0.124	cloudy
default	71668	1	4	0.45	0.134	dark
cropped	71668	1	2	0.25	0.110	sunny
cropped	71668	0	3	0.3	0.128	cloudy
cropped	71668	2	0	0.1	0.137	dark

V. Expert Feedback

Structure and logic

*** Clarity of study objectives: Evaluate if the main objectives of the course materials are clear and easily understandable.**

(1) It is impossible to understand the goals of this material.

(10) The goals are very clearly stated.

Expert 1: 9

Expert 2: 10

*** Achieving study objectives: Are the materials sufficient to achieve the stated study objectives?**

(1) Materials do not help achieve the stated goals.

(10) Materials are perfectly suitable to achieve the stated goals.

Expert 1: 9

Expert 2: 10

*** Structure logic: Evaluate the logical progression and structure of the study materials.**

(1) The materials are ordered randomly without apparent logic.

(10) There is clear path and progression in the ordering of tasks and topics.

Expert 1: 10

Expert 2: 10

*** Wording and understandability: Assess the clarity and understandability of the wording and structure of the text.**

(1) It is very difficult to understand what is said, instructions are unclear.

(10) The background information, definitions, and task instructions are clear and logical.

Expert 1: 9

Expert 2: 8

Content

*** Choice of content: Evaluate if the covered topics are well-chosen, relevant, and sufficient.**

(1) The topics are not relevant for data science and self-driving.

(10) All relevant topics are covered.

Expert 1: 10

Expert 2: 9

*** Accuracy of content: Evaluate the accuracy and correctness of the information presented in the study material, ensuring it aligns with established principles and practices in the field. Assess whether the content is factually correct and reflects current standards in the field.**

(1) The materials are full of factual errors, instructions go against good practices in data science.

(10) The materials are factually sound and present good practices.

Expert 1: 10

Expert 2: 10

*** Relevance of tasks: Evaluate the relevance and appropriateness of the tasks used to demonstrate important concepts addressed in the study material.**

(1) The tasks are irrelevant and do not support the written text.

(10) The amount of tasks and their content is perfectly tuned.

Expert 1: 10

Expert 2: 10

*** Completeness: Determine if the study material feels comprehensive and provides a sense of completeness. Were any important elements missing that should have been included?**

(1) The material omitted very important topics and felt incomplete.

(10) The materials covered all the most important topics and felt complete.

Expert 1: 8

Expert 2: 10

*** Quality and importance of the videos: Assess the importance and quality of the videos. Do they provide valuable insights or demonstrations that enhance the study material?**

(1) The videos were uninformative waste of time.

(10) The videos supported understanding and were of high quality.

Expert 1: 10

Expert 2: 10

Tasks

*** Completeness of instructions: Assess whether the instructions are clear and detailed enough for learners to understand and complete the tasks successfully.**

(1) Throughout the materials there were not enough instructions or they were unclear.

(10) Instructions for all tasks were high quality.

Expert 1: 8

Expert 2: 9

*** Effectiveness of assessments: Assess whether the assessments align with the learning objectives and provide meaningful feedback to learners. Do the assessments accurately measure learner comprehension and skill acquisition?**

(1) Assessments measure the acquisition of random and unimportant facts and do not help evaluate nor provide feedback.

(10) Assessments evaluate the retention of relevant pieces of knowledge, allow adequate feedback to learners.

Expert 1: 10

Expert 2: 10

VI. License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Artur Kreegipuu**,

1. grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

Creation of Materials to Teach Data Science via Self-Driving,

supervised by Ardi Tampuu.

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Artur Kreegipuu

14.05.2024