VERENA WEBER

Example Projects

The **paper icon** in the Outcome box means a paper was published about the project, click on the icon to **read the paper for more details**.

Distillation techniques for model stability

VERENA WEBER

PROBLEM

- Customer experience must not degrade after model update, i.e. ensure all functionalities still work
- Thus prediction changes between new and old model were analyzed
- The process is manual and time-consuming and does not scale

SOLUTION

- Bias the new model towards predictions of the old on training samples where the old model gave correct predictions (reduce negative flips)
- this reduces amount of prediction changes to be analyzed
- Done through change in the loss function

TOOLS

python™ PyTorch

Huggingface

OUTCOME

- reduced prediction changes by up to 55 %
- approach to be rolled out in production
- estimated to save 25 % of time in manual analysis

Model robustness and adversarial attacks

VERENA WEBER

PROBLEM

- Deep Learning models are very sensitive to changes in the input
- Even small changes, like adding a word like please, can change the classification
- This is not desirable customer experience

SOLUTION

- To make the model more robust towards small changes in the input, we trained a T5 model to generate so called adversarial attacks, i.e. data that leads the target model towards a wrong prediction
- We then used that generated adversarial data as training data for the production model

TOOLS

python™ PyTorch

Huggingface

OUTCOME

- 70 % error rate reduction on adversarial test data
- reduction of customer perceived defect in online A/B test, rolled out in production

Purchasable item recognition

VERENA WEBER

PROBLEM

- For a multi-functional voice assistant with shopping functionality, a purchasable item is an item that can be bought on the connected e-commerce platform. This is crucial for fulfillment of the request.
- E.g.: iPhone = purchasable ✅, Lamborghini ❌= non-purchasable

SOLUTION

- We designed a classifier run on top of the existing system to detect whether an item is purchasable (signal goes back into system)
- Semantic search is used to match the item in the request with data from the product catalog
- A twin network decides if item in request is similar enough to data from product catalog

TOOLS

python™ PyTorch

Huggingface

OUTCOME

- Classifier is able to correct the mistakes of existing system, increase in accuracy of 80 %
- Simplified approach rolled out in production led to reduction of online defects

Semi-supervised Learning

VERENA WEBER

PROBLEM

- Data annotation by humans is slow and costly.
- Usually just a random sample of data is annotated, regardless of whether the model is able to correctly interpret each instance in the sample (assuming we work with a system that is regularly updated with new data)

SOLUTION

- To speed up the annotation process, we show the annotator the model interpretation and let them verify or if needed correct it.
- As such, data is only annotated if the sample is incorrectly interpreted by the model
- Correct predictions are directly ingested as training data

TOOLS

python™ PyTorch

Huggingface

OUTCOME

- Annotation volume is reduced by 97 % and cost by 60 %
- Model performance increases due to reduced annotation inconsistencies

Tool for release of model artifacts

VERENA WEBER

PROBLEM

- The release of model artifacts was a very manual and therefore time-consuming process that took a scientist several days
- Repetitive tasks with high potential for automation, not a good use of a scientist's time

SOLUTION

- Delivered a unit-tested software package written in Python that automates the release process
- Designed and gave training for users of the new package on how to use the software

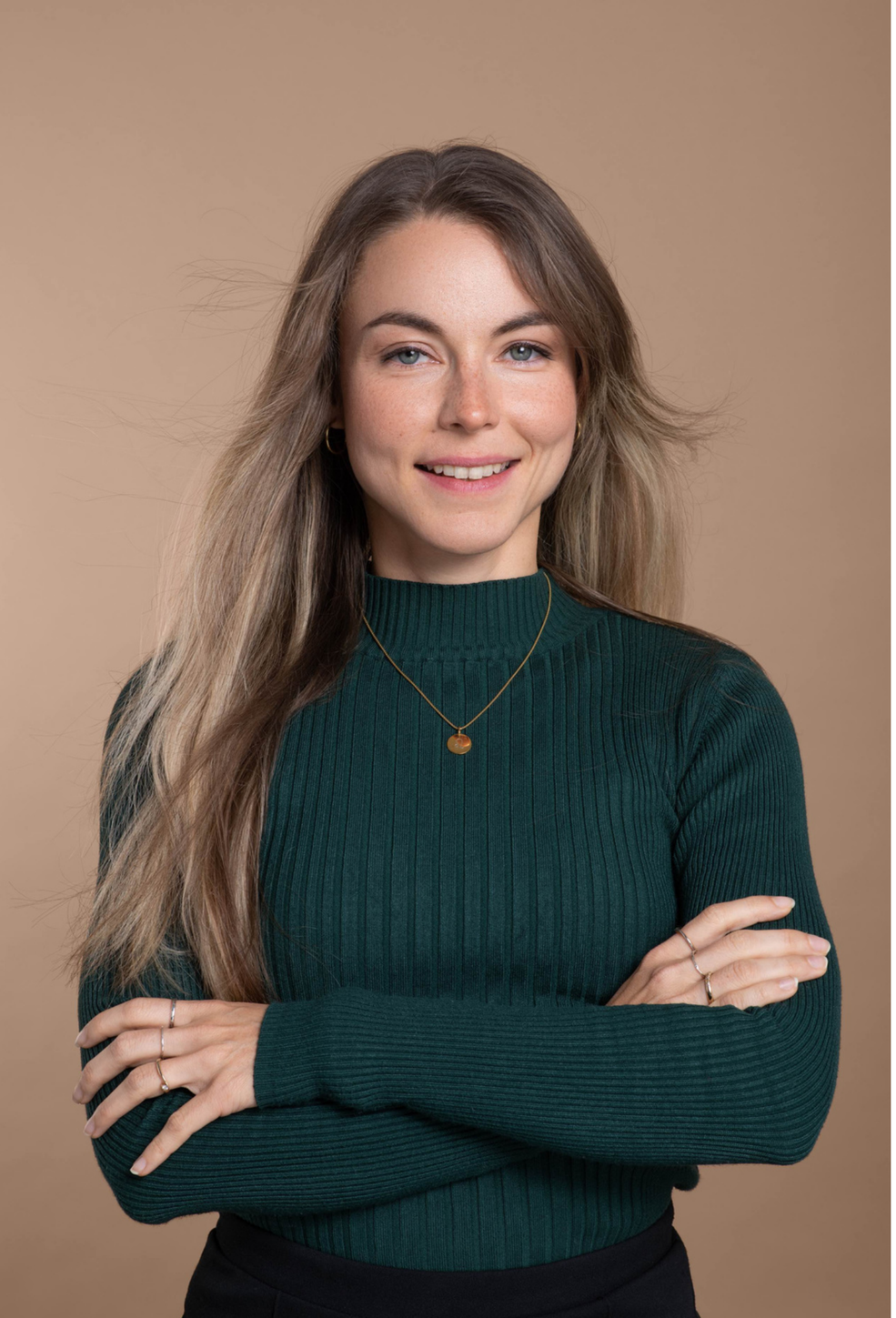
TOOLS

python™

pytest

OUTCOME

- Manual work reduced from days to hours
- Due to reduced complexity, release could be transferred from scientist to role with less technical expertise



VERENA WEBER

Let’s talk.

GET IN TOUCH

+49 176 62336998

verena@verenaweber.de