Supervised Learning vs. Reinforcement Learning

Learning Portfolio 8

universität innsbruck

Background

In Learning Portfolio 5, I have used 2022 transaction data from my bank account to categorize expenses and create expense reports.

The AI was trained using huggingface transformers in a supervised learning approach.

However, as my spending habits changed when I moved from Frankfurt to Innsbruck, I had a problem with data drift.

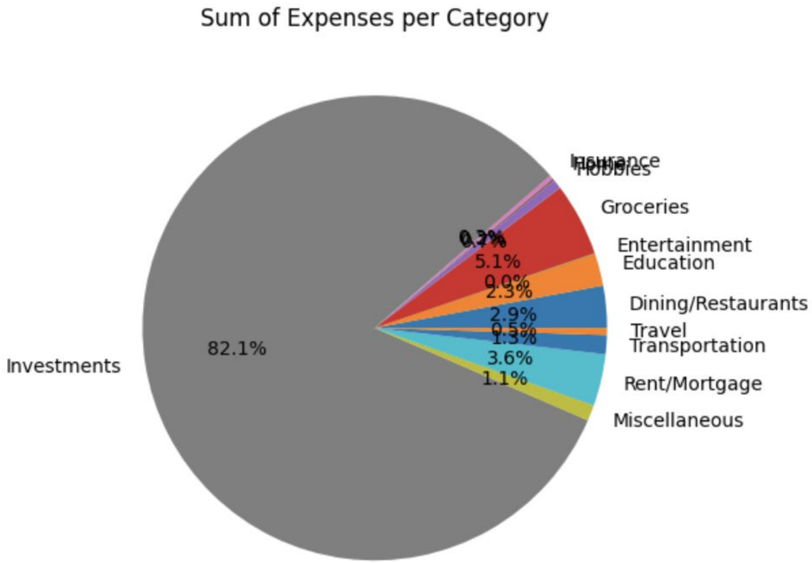
In this presentation, I try to address this issue.

Reminder: The problem

Training data was from 2022 when I lived in Frankfurt, Germany.

I created an expense report for April 2023 and there were more predictions errors.

Data drift: My transactions have changed, because I lived in Innsbruck and was on vacation in Italy



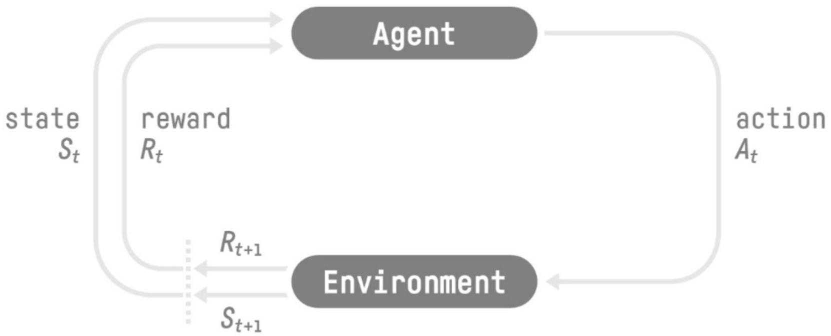
The hope: Reinforcement learning

My motivation for learning reinforcement learning was to solve the issue. So how could that look like in Reinforcement learning?

States: Each variable position of a feature vector of text

Actions: Make a prediction, read next feature

Rewards/Punishments: Correct prediction/Wrong prediction



The hope: Reinforcement learning

Limitations:

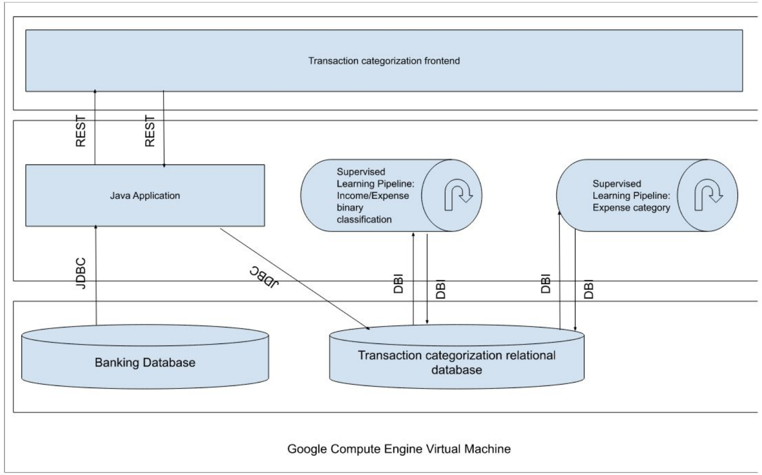
- Q-Learning is not scalable
- Action “Next Feature”: Why would the algorithm ever not read the next word?
- Very few research papers about the issue
- Tedious coding because of limited library support

	Reinforcement Learning	Supervised Learning
Objective	Learn optimal actions in an environment through trial and error.	Learn a mapping function between input and output data.
Feedback	Receives feedback in the form of rewards or penalties.	Receives labeled training data for each input.
Training data	Interacts with an environment, no explicit input-output pairs.	Uses labeled input-output pairs for training.
Exploration	Emphasizes exploration to discover optimal actions.	No explicit exploration; relies on labeled data.
Output	Actions or policies (sequences of actions) for decision-making.	Predicted labels or continuous values for inputs.
Feedback timing	Feedback may be delayed and received over multiple steps.	Immediate feedback is available for each training example.
Dependency on labels	No explicit dependency on labeled data.	Depends on labeled data for training and evaluation.
Generalization	Learns from experiences to generalize to unseen states.	Learns patterns to generalize to unseen input instances.
Application	Commonly used in robotics, game playing, and control systems.	Widely used in classification, regression, and prediction tasks.
Examples	AlphaGo, self-driving cars, game-playing agents.	Image classification, speech recognition, sentiment analysis.

universität innsbruck

The alternative: Iteratively train supervised learning model

1. Train a supervised expense categorization pipeline using a basic dataset.
2. Push the model and associated weights to a repository.
3. Query new data at the end of each month from my banking provider.
4. Predict expense categories using the trained expense categorization pipeline.

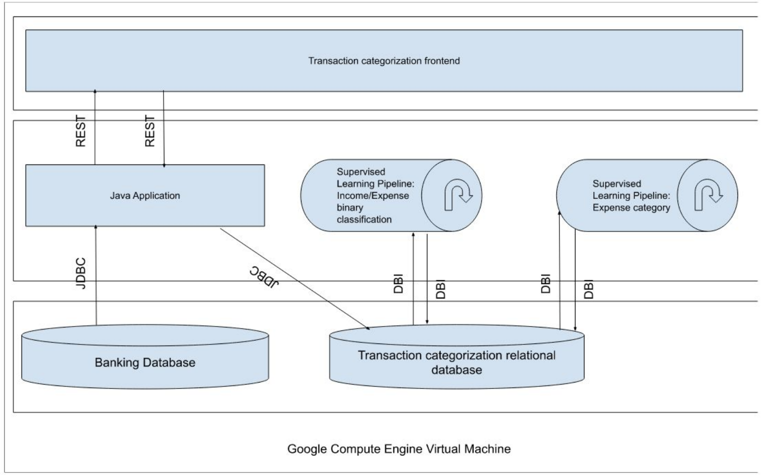


The alternative: Iteratively train supervised learning model

5. Display the predictions as a report in a webapplication.

6. Display the single predictions in a webapplication. Allow for recategorization.

7. Collect changed predictions every regularly and adapt the model weights by training on the new data.



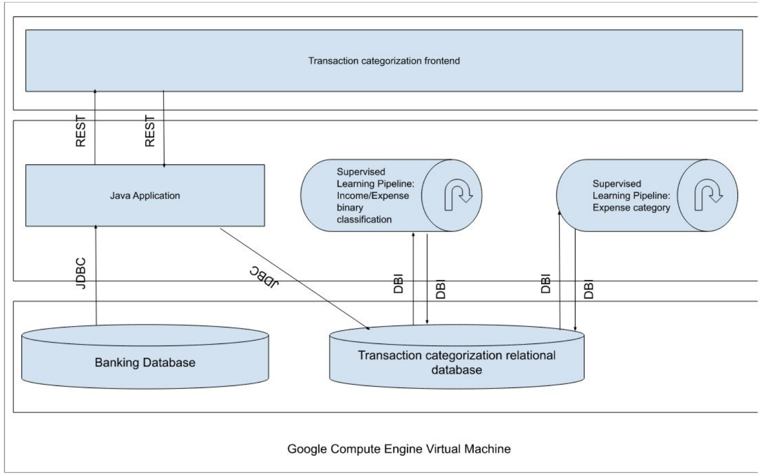
The alternative: Iteratively train supervised learning model

While it would be a cool challenge, it would be a lot of work too. Particular challenges would be to provide a secure frontend as sensitive data would be exposed to the web.

Also, querying the banking API will be challenging given the server-side token validations they have in place.

Finally, getting everything running on a cloud VM is a bit of work too.

As this is final's week, the portfolio won't go beyond the architecture sketch.



Kontakt

Fabian Leuk
12215478
fabian.leuk@student.uibk.ac.at