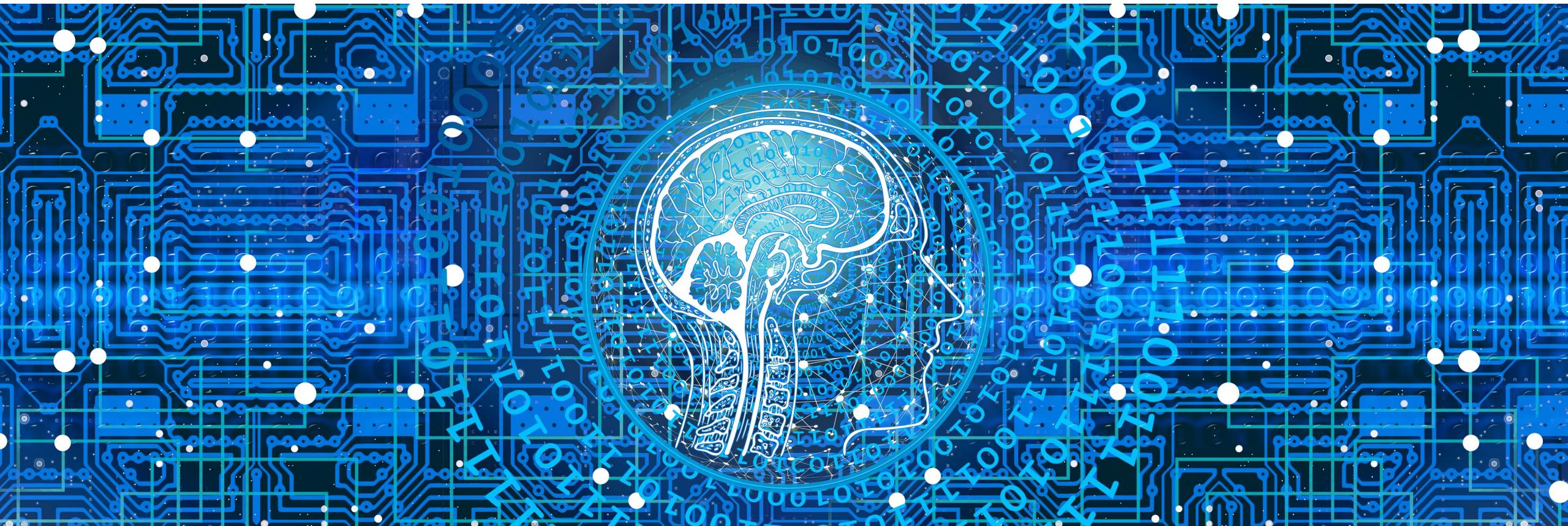


How does Machine Learning Meet the DevOps Technical Practices?



Mohammed Sayagh
msayagh.github.io

The goal of my Presentation is to Discuss my Research Experience and the Scientific Literature that use Machine Learning to Improve the DevOps Process

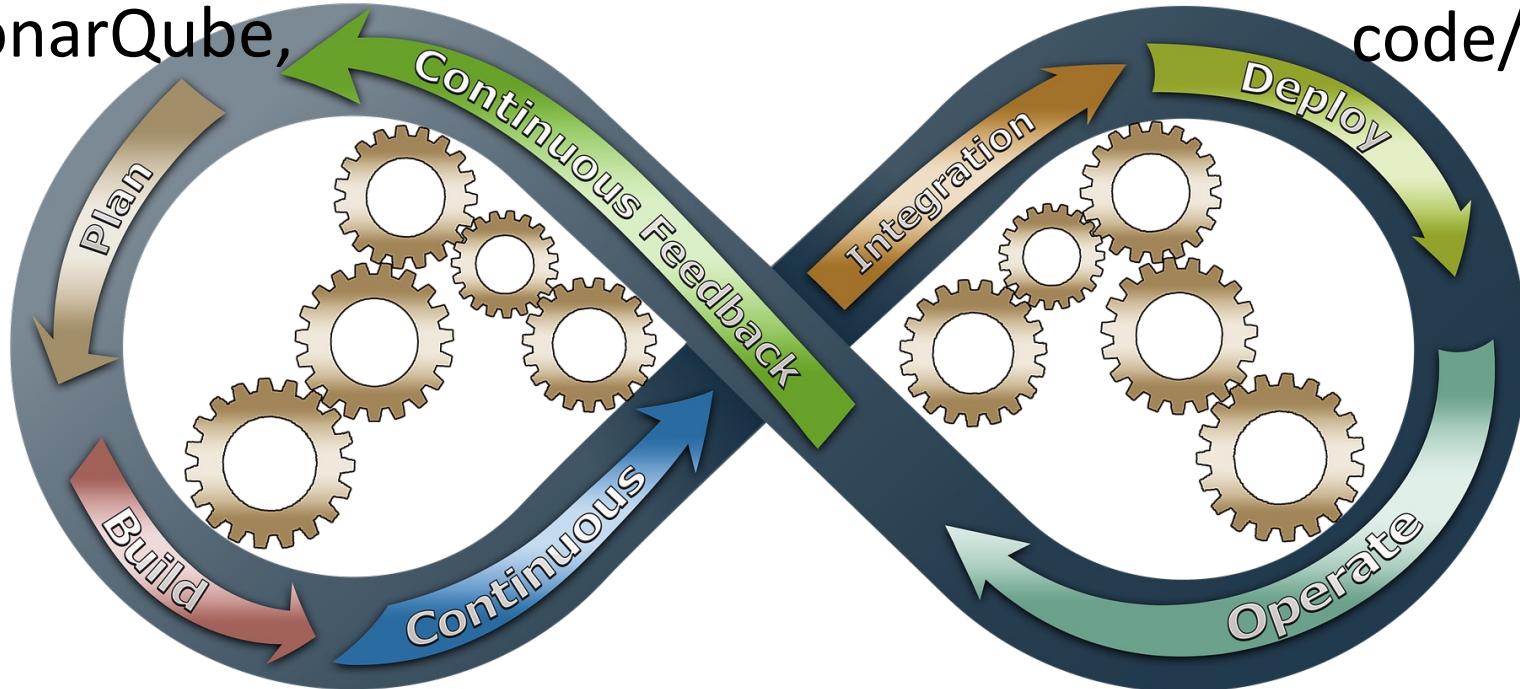
Machine learning models even simple algorithms can fix problems, while taking into consideration the context in which a model is applied should be taken into consideration



DevOps Practices and Tools Help Improve Quality and Agility

Plugins and tools for the quality – SonarQube, Snyk

CI/CD pipelines



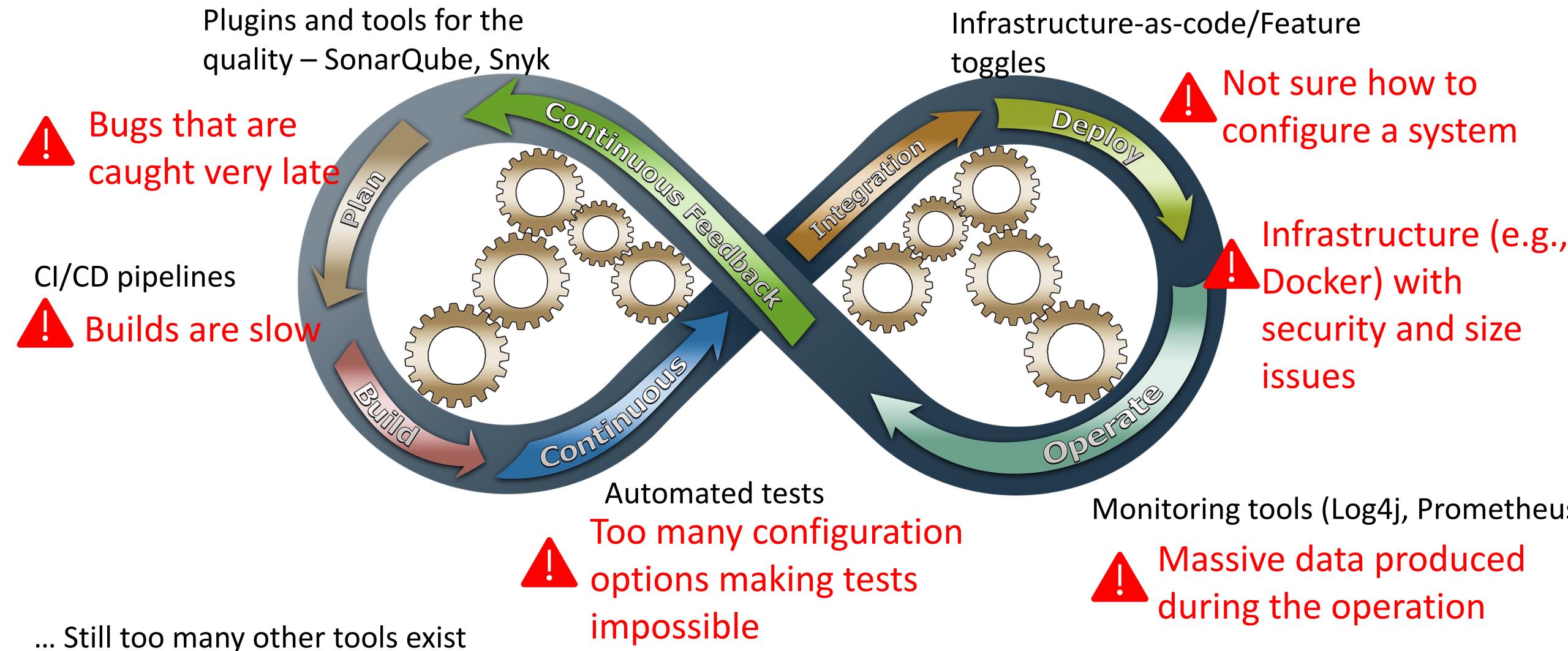
Automated tests

Infrastructure-as-code/Feature toggles

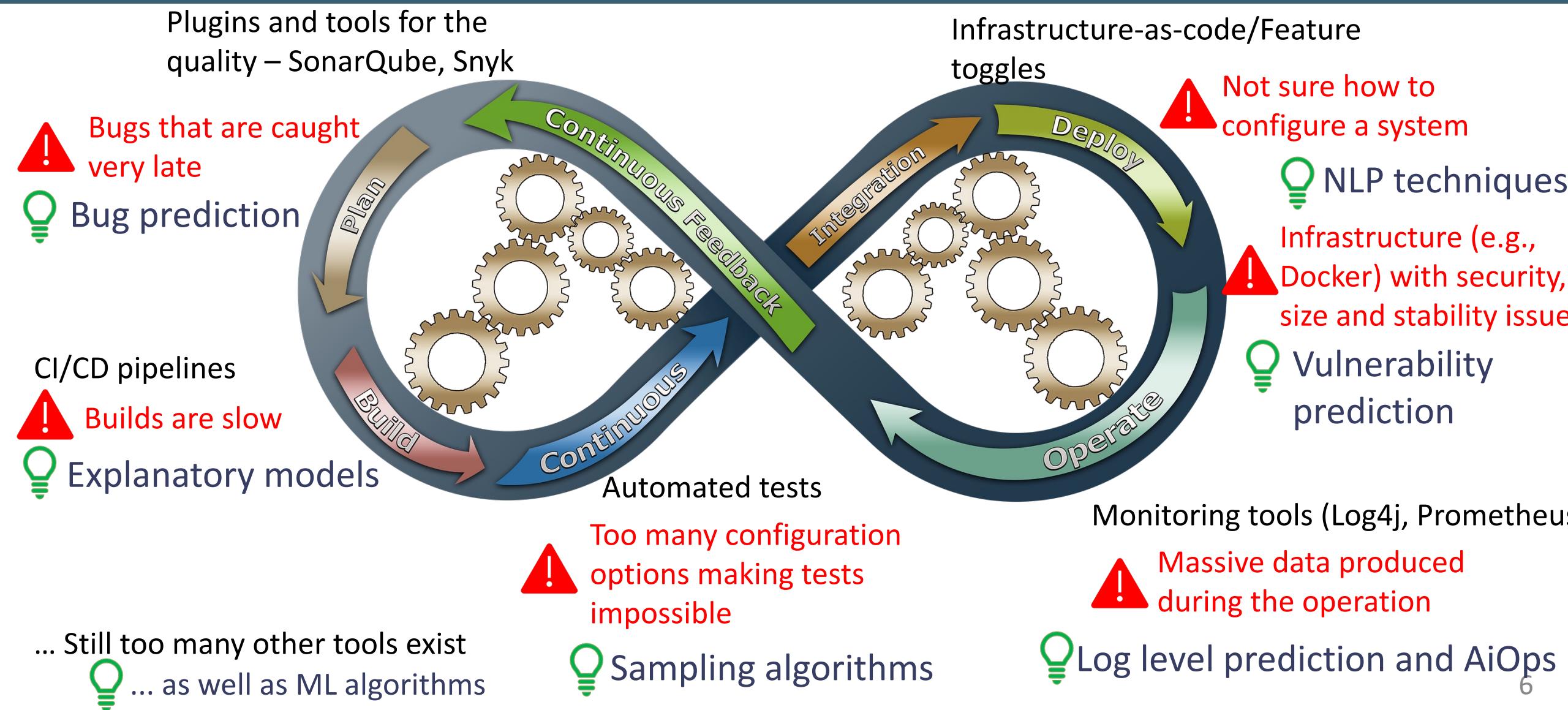
Monitoring tools (Log4j, Prometheus)

... Still too many other tools exist

Too many Problems still Exist

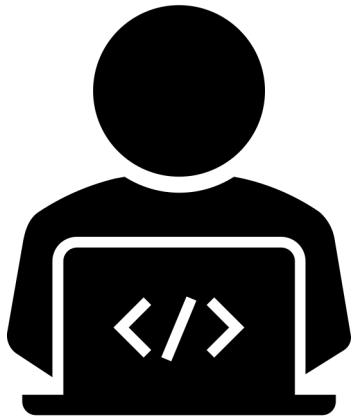


... for which the Scientific Literature Evaluated different Statistical and Machine Learning Models



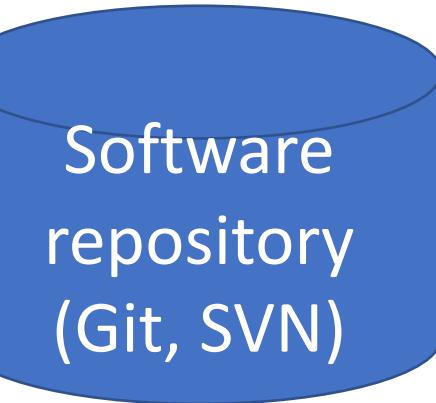
For example, Bug Prediction Models can Help Reduce Bugs before Pushing Code Changes

Developers

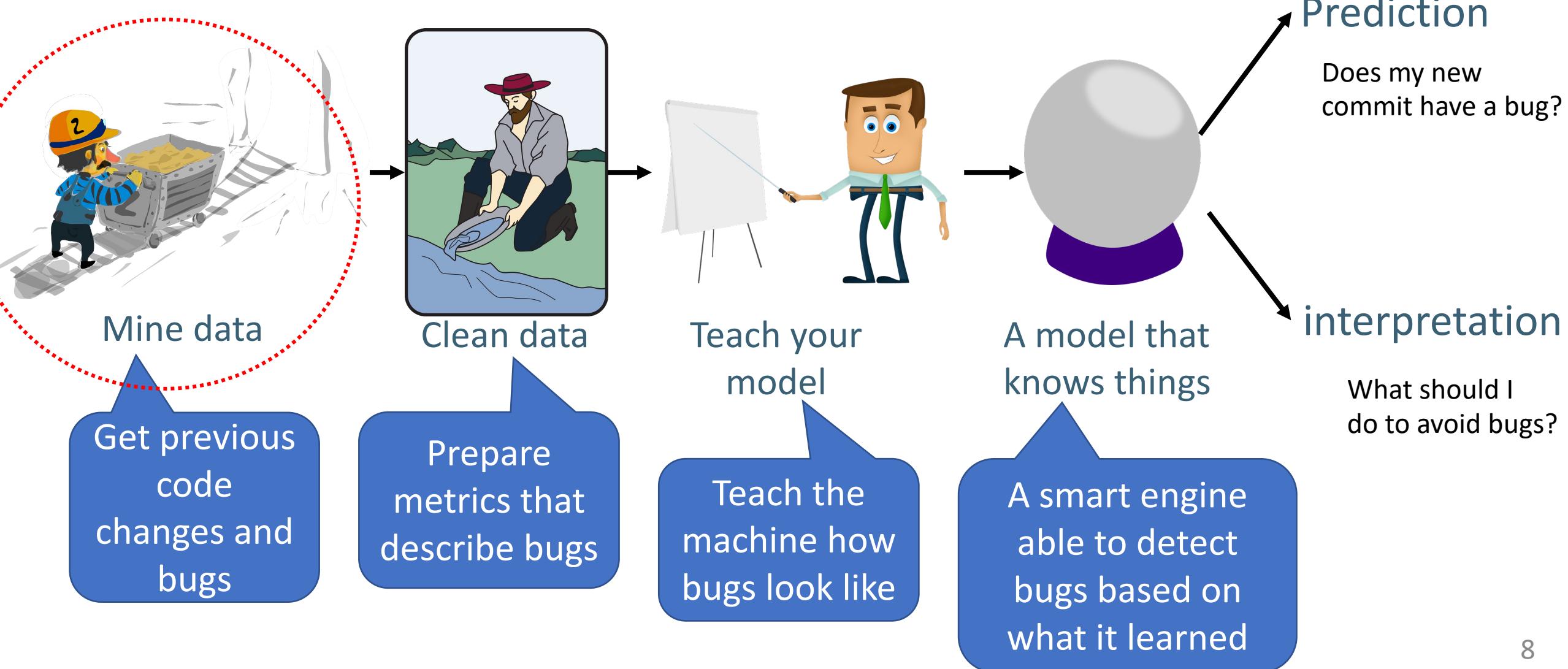


Git push

Your code change has a chance of 80% to be buggy



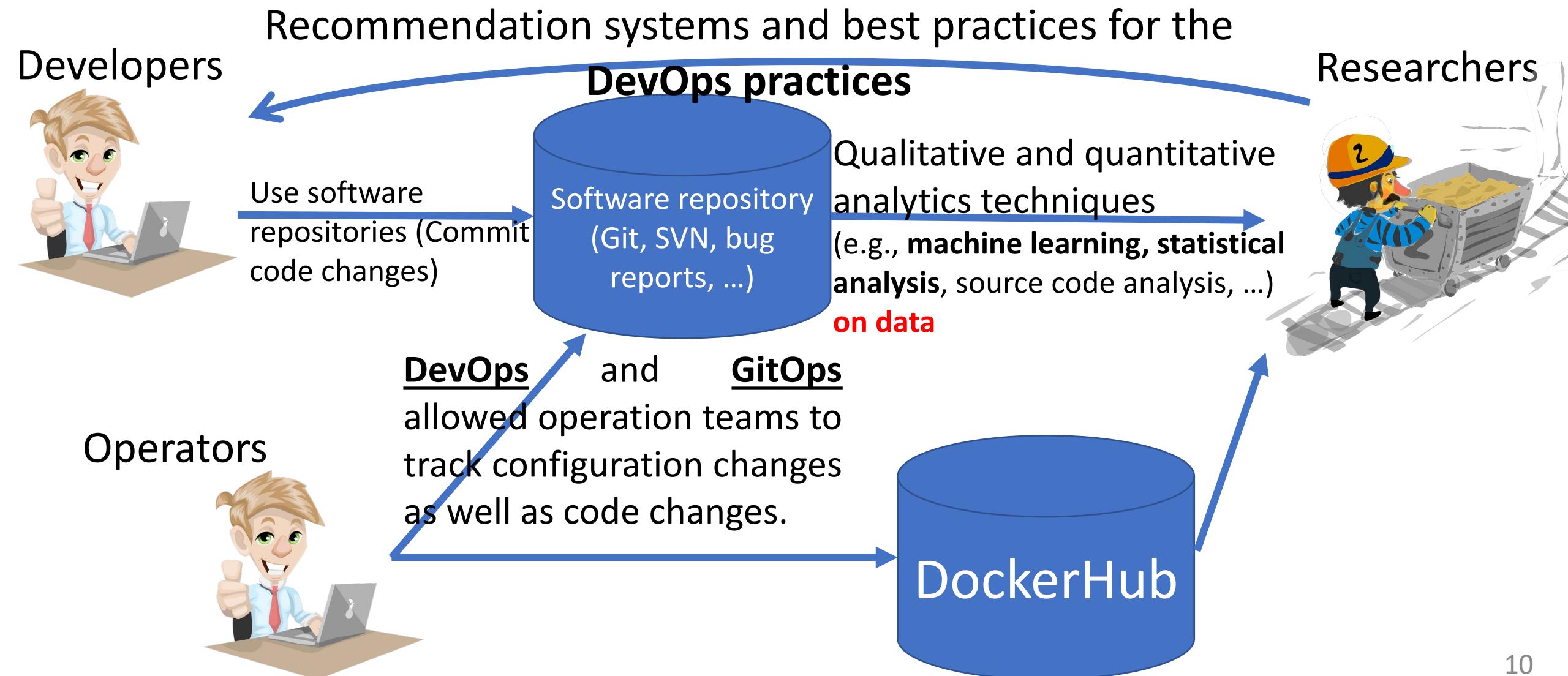
Machine Learning in a Nutshell



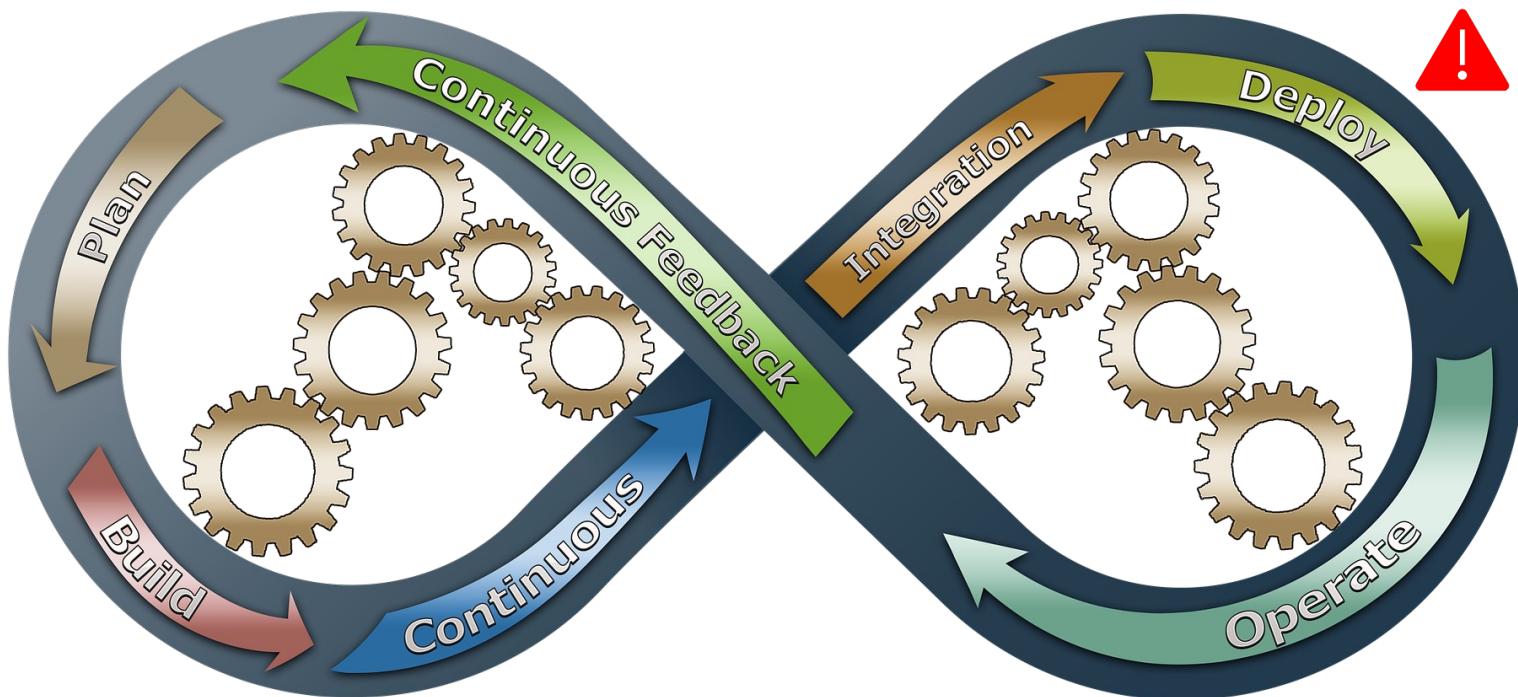
Okay! Where can we Get Data from?



These Machine Learning Solutions are Based on Data coming from Software and Operation Repositories



Leveraging NLP to Help Configuring a Software System



⚠️ Not sure how to
configure a system

💡 NLP techniques

⚠️ Massive data produced
during the operation

💡 Log level prediction

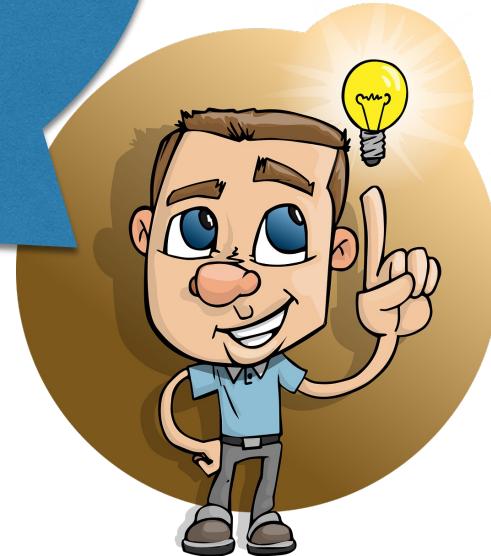
Configuration is a Mechanism to Customize the Behaviour of a Software System

- A set of configuration options stored in a file
- Increase the memory `memory_size_limit = 64 MB`
- Enable or disable the upload of files through the option `file_upload_enabled = On/Off`
- Very long textual files (e.g., conf.ini)



Configuring a software system is challenging and time-consuming

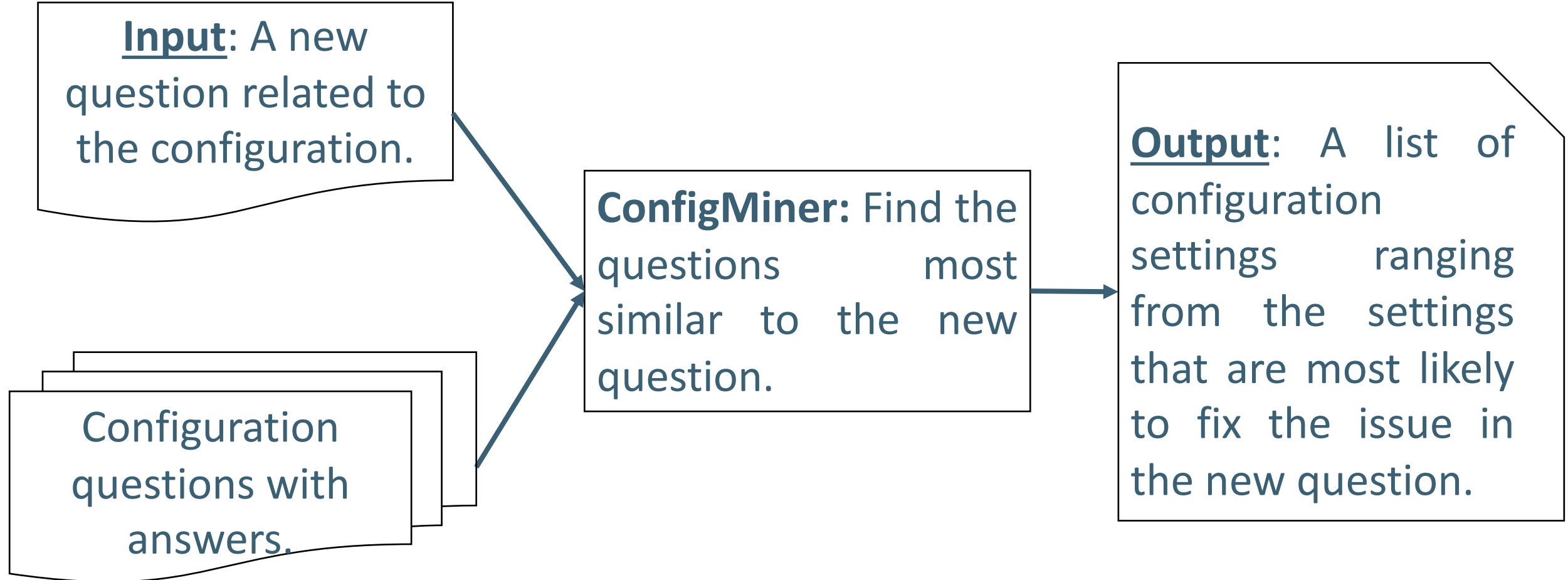
“nothing is more frustrating than having to figure out how to put an environment specific configuration on release day.”



Users ask about their Configuration in Online Platforms

- Ask in forums/internal platforms different types of questions
 - How to do something
 - How to troubleshoot an error
 - Things are not working
 - ...
- Will receive an answer but often after **a long time**

We Built a Dataset of Questions/Answers and use textual similarity to identify the correct configuration parameters



The Idea of the Framework using NLP Methods

New Question 1: Not able to upload images in my web application

Question 1: How to disable javascript in my browser
Option: javascript.enabled

Question 2: Uploading images ends up with a system crash
Option: memory_limit

Question 3: How to disable drop-down menu
Option: full-screen-api.warning.timeout

Similarity :

Rank	Similarity
1. New question vs Question 2	70%
2. New question vs Question 1	40%
3. New question vs Question 3	20%

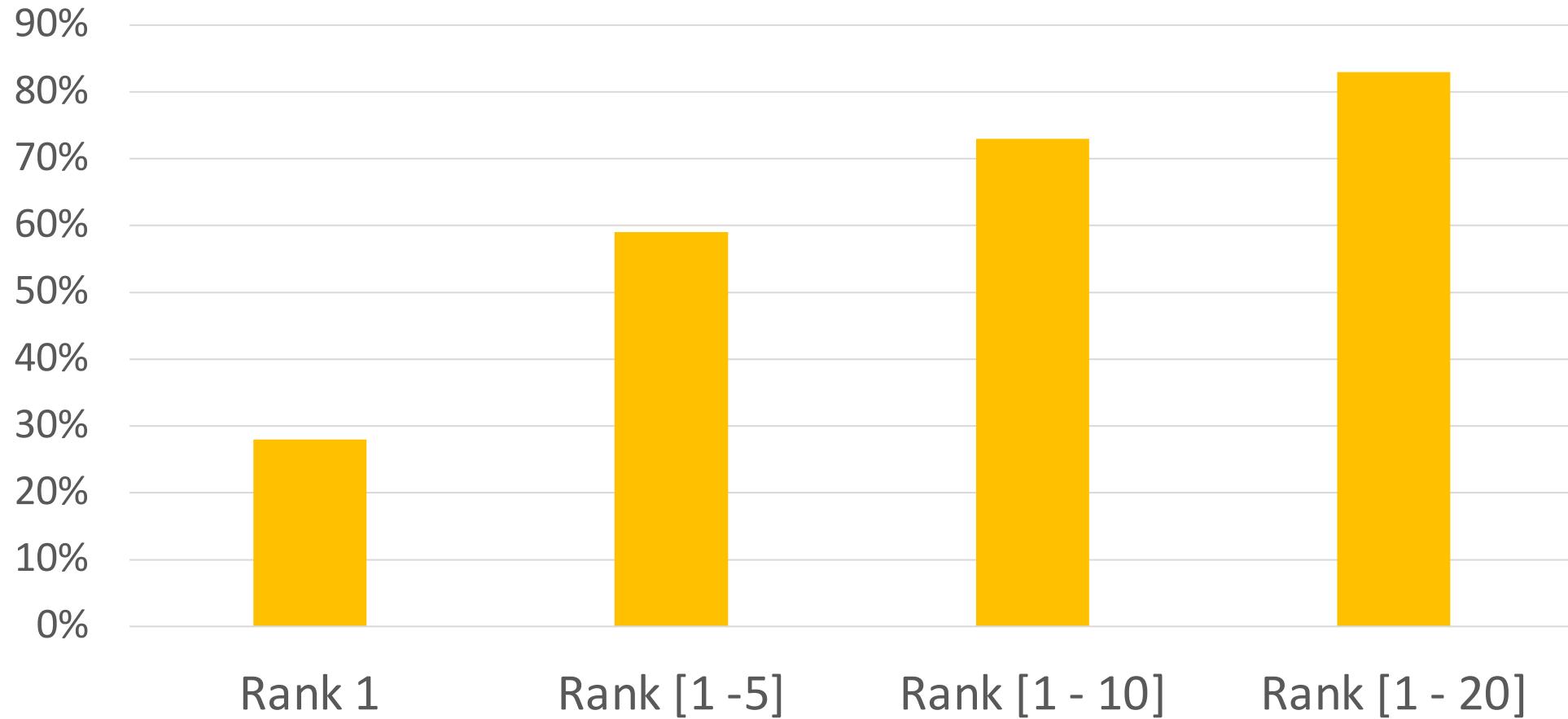
Results :

Rank	Option
1.	memory_limit
2.	java.script.enabled
3.	api.warning.timeout
...	
20.	

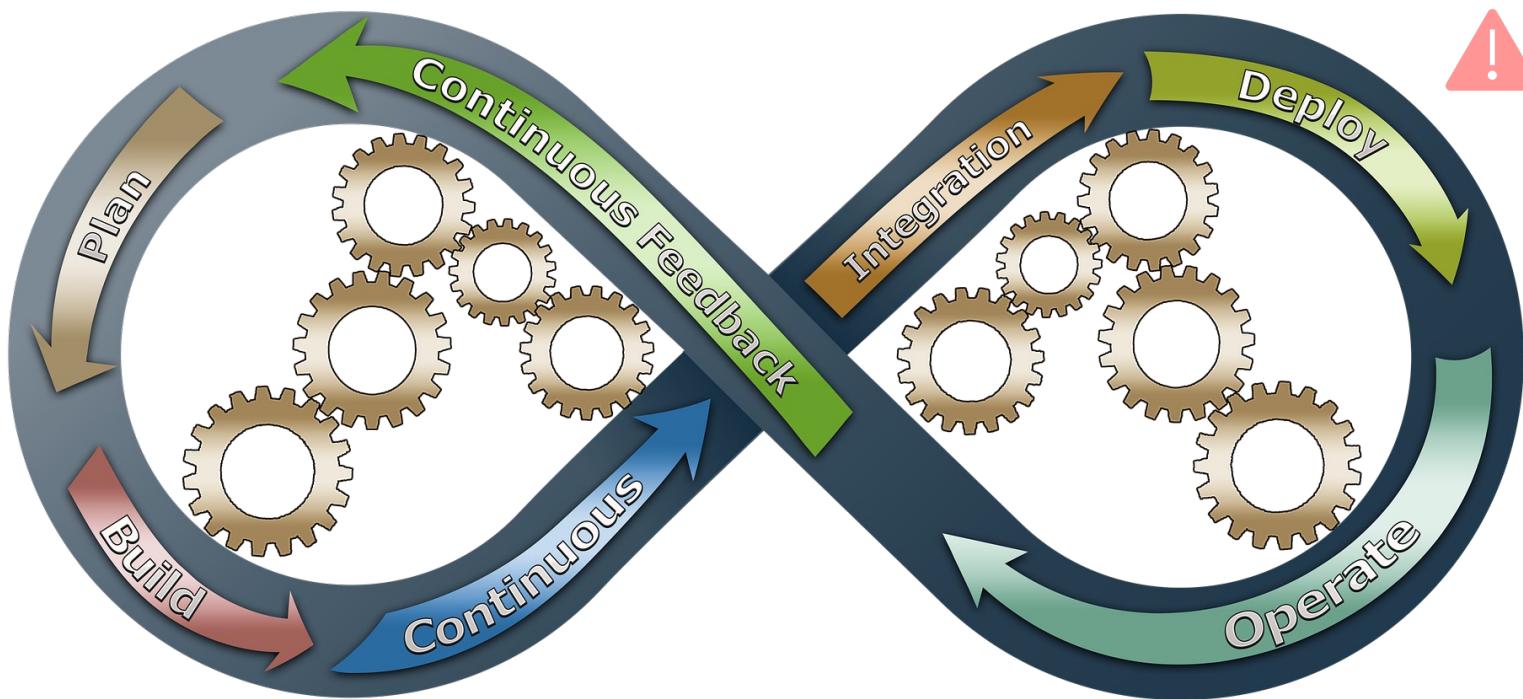
TF-IDF

doc2vec

ConfigMiner is able to Identify the Appropriate Option at a Median Rank of 4



How to Use Machine Learning for Developing better Log Levels



⚠ Not sure how to
configure a system

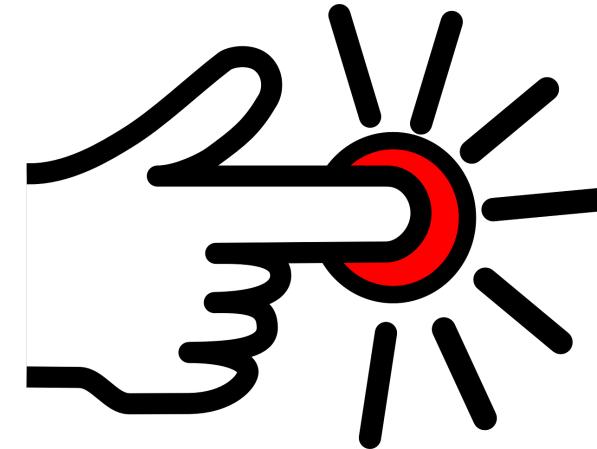
💡 NLP techniques

⚠ Massive data produced
during the operation

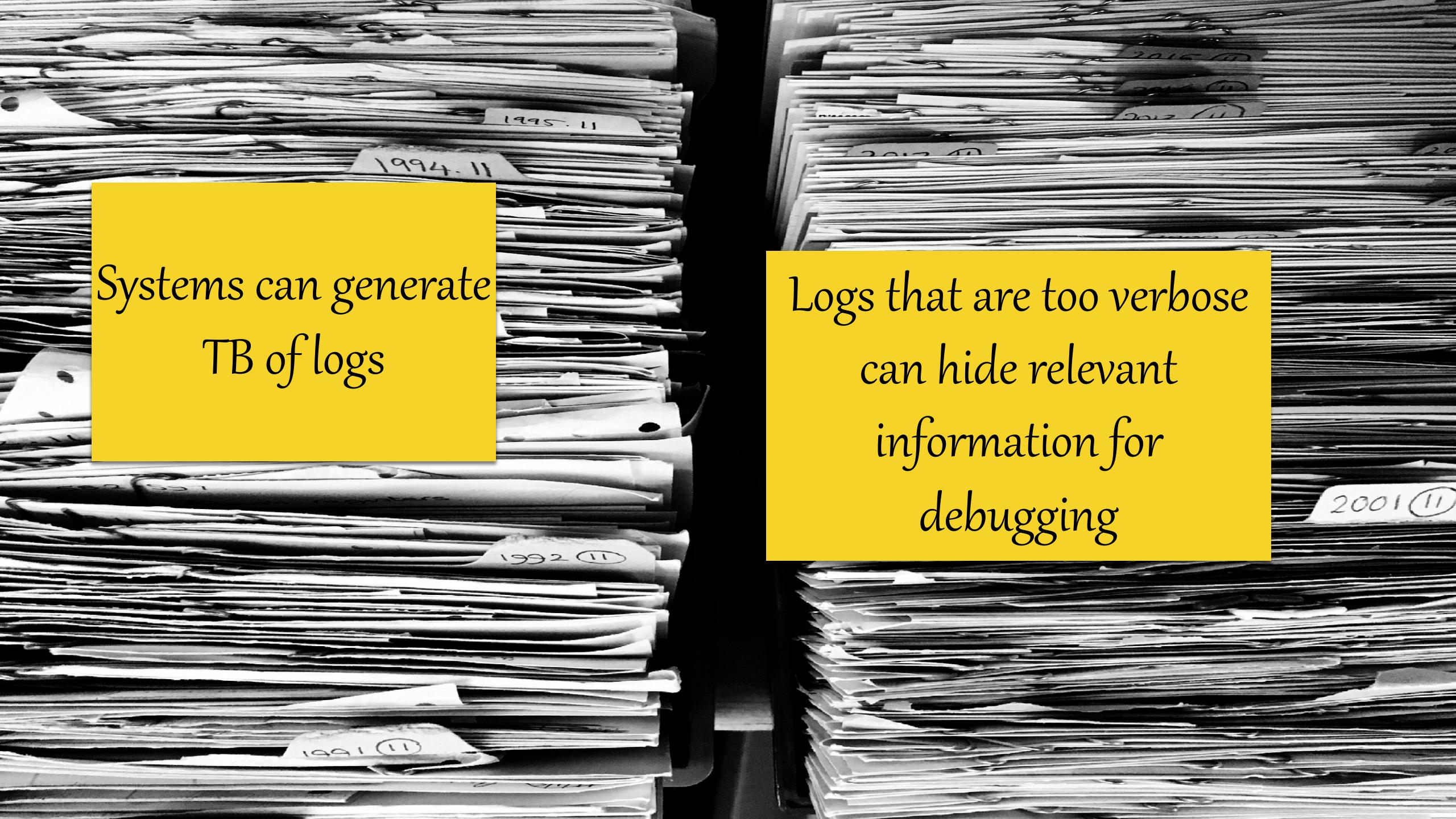
💡 Log level prediction

Developers insert logging statements to trace the execution of a software system

```
private void myFunction(String parameter){  
  
    logger.info("Entering myfunction");  
  
    if (bar() == false) {  
        logger.trace("Bar returned good results");  
    } else {  
        logger.error("Error in bar's results");  
    }  
  
    logger.info("Leaving myfunction");  
}
```



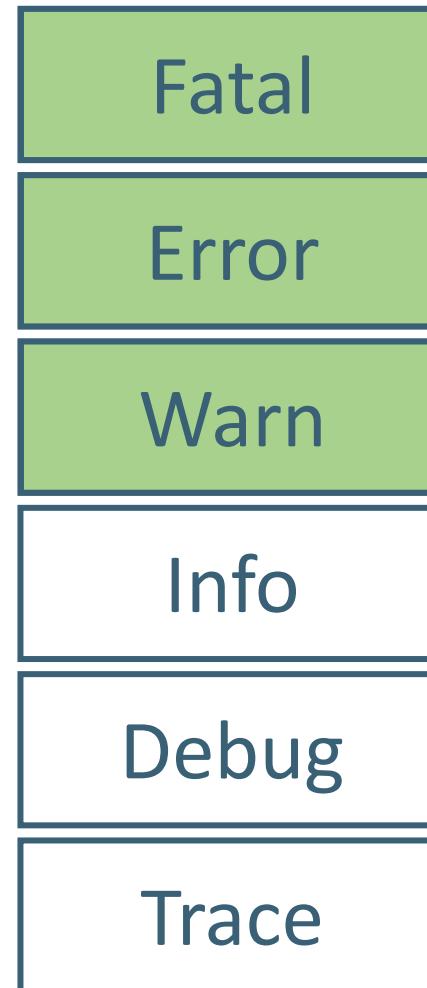
```
2022-10-24 18:50:45 INFO HelloExample:16 - Entering myfunction  
2022-10-24 18:50:45 TRACE HelloExample:19 - Bar returned good results  
2022-10-24 18:50:45 INFO HelloExample:24 - Leaving myfunction
```



Systems can generate
TB of logs

Logs that are too verbose
can hide relevant
information for
debugging

One Way is to Control the Verbosity of Logs



Less
verbose

More
Verbose

Fix the verbosity at Warn level

Deciding which log messages to be under which log level is not straightforward

Prediction Models Exist for Predicting Log Levels for a new Log Statement



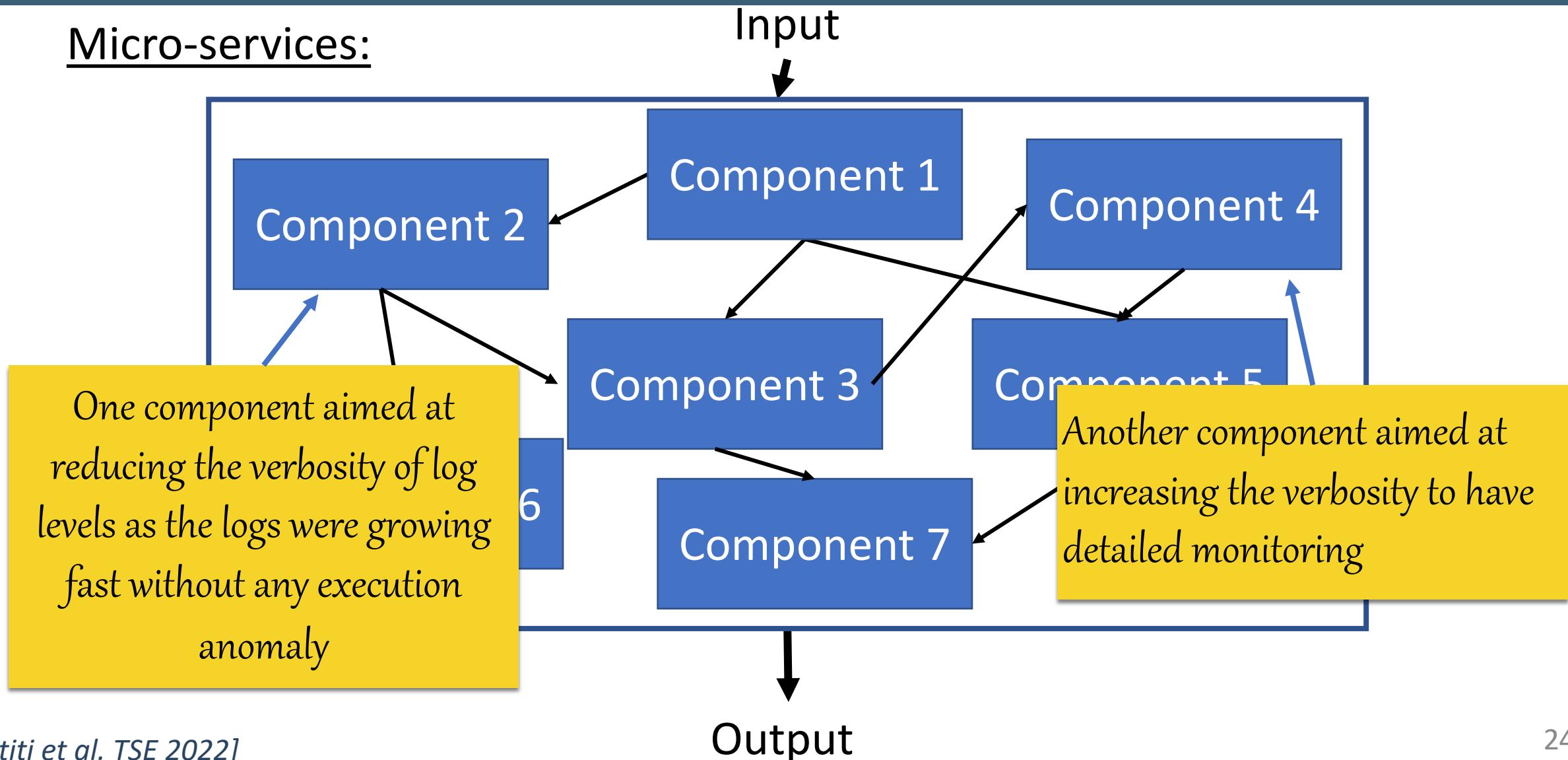
Predict log level

```
private void myFunction(String parameter){  
  
    logger.info("Entering myfunction");  
  
    if (bar() == false) {  
        logger.trace("Bar returned good results")  
    } else {  
        logger.error("Error in bar's results");  
    }  
  
    logger.info("Leaving myfunction");  
}
```

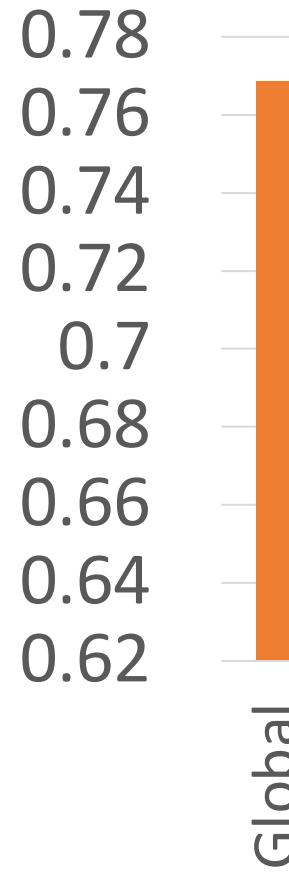
Good performances:
AUC above 75%

Other models exist on
the literature to predict
other aspects of logging

Should we have a Model for the whole Micro-service system or one Model for each Micro-service?

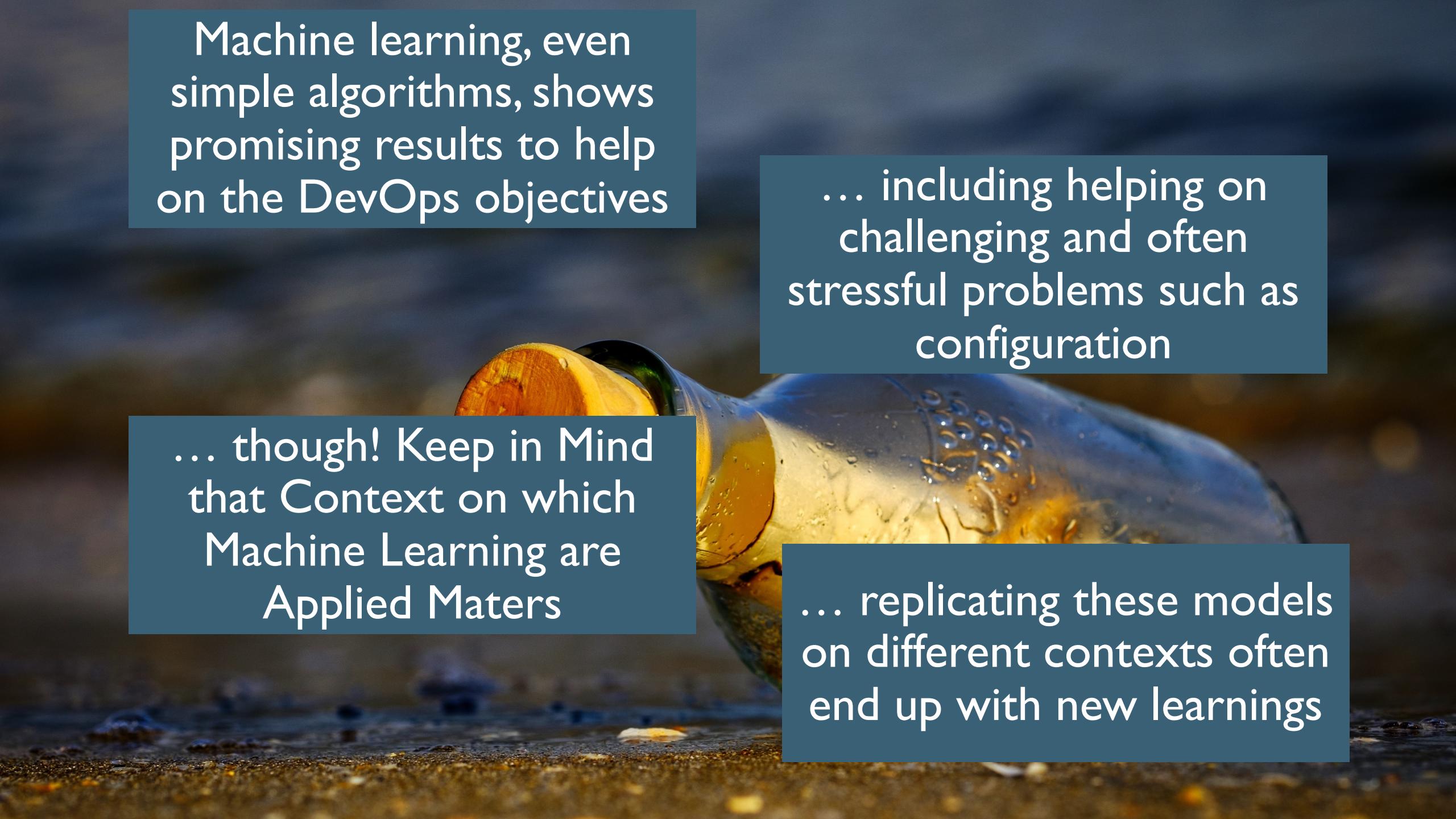


The performance of the global model on each component is lower than the performance of the same model when tested on the dataset that is obtained from all the components



References

- D'Ambros, Marco, Michele Lanza, and Romain Robbes. "An extensive comparison of bug prediction approaches." *2010 7th IEEE working conference on mining software repositories (MSR 2010)*. IEEE, 2010.
- Kamei, Yasutaka, et al. "A large-scale empirical study of just-in-time quality assurance." *IEEE Transactions on Software Engineering* 39.6 (2012): 757-773.
- Ghaleb, Taher Ahmed, Daniel Alencar Da Costa, and Ying Zou. "An empirical study of the long duration of continuous integration builds." *Empirical Software Engineering* 24.4 (2019): 2102-2139.
- McIntosh, Shane, et al. "An empirical study of build maintenance effort." *2011 33rd International Conference on Software Engineering (ICSE)*. IEEE, 2011.
- Medeiros, Flávio, et al. "A comparison of 10 sampling algorithms for configurable systems." *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*. IEEE, 2016.
- Sayagh, Mohammed, et al. "Software configuration engineering in practice interviews, survey, and systematic literature review." *IEEE Transactions on Software Engineering* 46.6 (2018): 646-673.
- Ibrahim, Md Hasan, Mohammed Sayagh, and Ahmed E. Hassan. "Too many images on DockerHub! How different are images for the same system?" *Empirical Software Engineering* 25.5 (2020): 4250-4281.
- Ouatiti, Youssef Esseddiq, et al. "An empirical study on log level prediction for multi-component systems." *IEEE Transactions on Software Engineering* 01 (2022): 1-1.
- Dang, Yingnong, Qingwei Lin, and Peng Huang. "AIOps: real-world challenges and research innovations." *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. IEEE, 2019.
- Lyu, Yingzhe, et al. "An empirical study of the impact of data splitting decisions on the performance of AIOps solutions." *ACM Transactions on Software Engineering and Methodology (TOSEM)* 30.4 (2021): 1-38.
- Yao, Kundi, et al. "Improving state-of-the-art compression techniques for log management tools." *IEEE Transactions on Software Engineering* (2021).
- Li, Heng, Weiyi Shang, and Ahmed E. Hassan. "Which log level should developers choose for a new logging statement?." *Empirical Software Engineering* 22.4 (2017): 1684-1716.
- Zerouali, Ahmed, et al. "On the relation between outdated docker containers, severity vulnerabilities, and bugs." *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2019.
- Zerouali, Ahmed, et al. "On the impact of outdated and vulnerable javascript packages in docker images." *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2019.
- Liu, Jinyang, et al. "Logzip: extracting hidden structures via iterative clustering for log compression." *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2019.
- Logothetis, Dionysios, et al. "In-situ {MapReduce} for Log Processing." *2011 USENIX Annual Technical Conference (USENIX ATC 11)*. 2011.



Machine learning, even simple algorithms, shows promising results to help on the DevOps objectives

... including helping on challenging and often stressful problems such as configuration

... though! Keep in Mind that Context on which Machine Learning are Applied Matters

... replicating these models on different contexts often end up with new learnings

