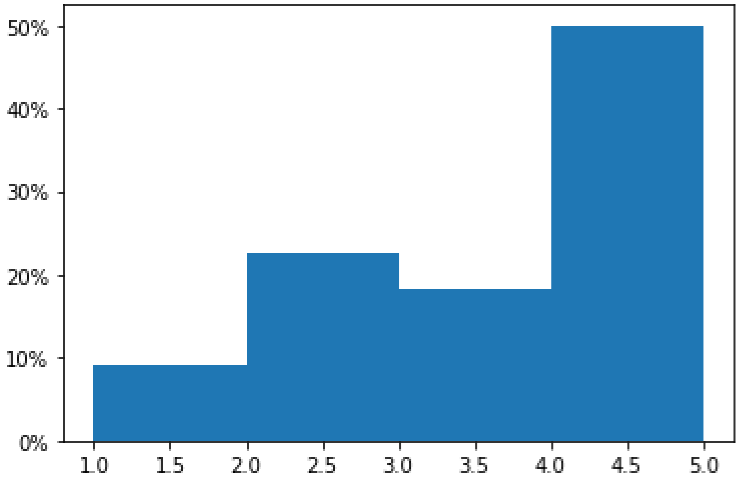
Problems related to MLOps in general **1**
Path to model deployment from a jupyter notebook is really long. 1
Sharing of model results is not easy 2
Model breaks for latency reasons 2
Infrastructure cost for Machine Learning is too high 2
Building monitoring and alerting of ML systems is really hard. 3
A/B infrastructure or tooling support is missing. 3
Maintaining retraining pipelines is really hard. 4
Debugging when things go wrong is very time consuming. 4
Problem Related to Jupyter Notebooks **5**
Notebooks don’t encourage production quality code. 5
Non linear execution of experimental notebook is error prone 6
Collaboration in notebooks is not easy. 6
Version control and code review of notebooks is not easy. 7
Notebooks present a security flaw 7
Notebooks don’t have basic IDE productivity features 8

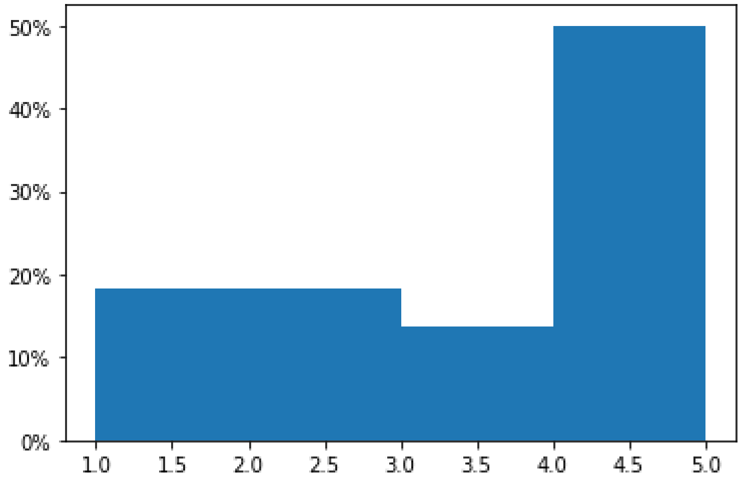
Problems related to MLOps in general

Path to model deployment from a jupyter notebook is really long.



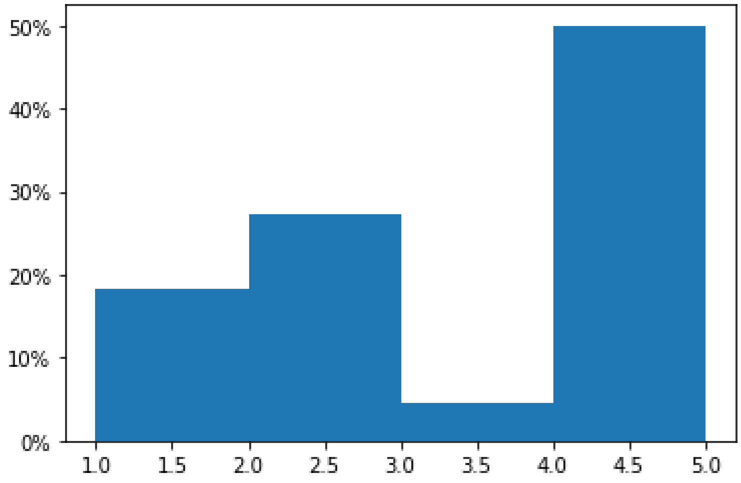
Sharing of model results is not easy

Sharing of model results with non-tech users and getting feedback is not easy.



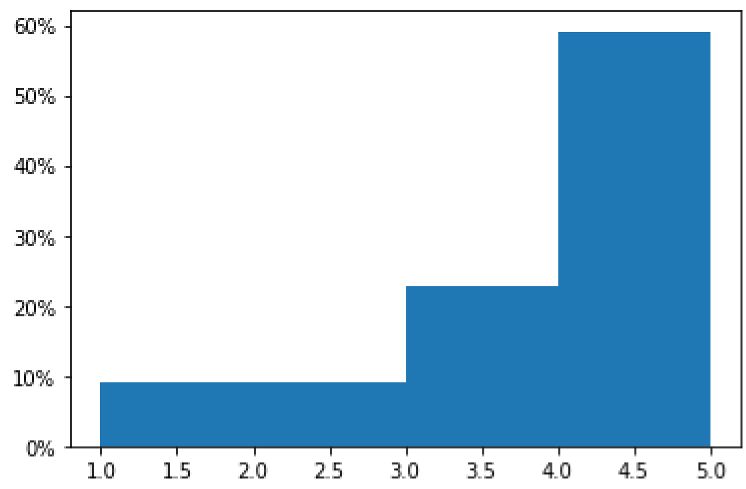
Model breaks for latency reasons

Even if a model is optimised for accuracy, it often breaks because of latency reasons when you deploy in production.

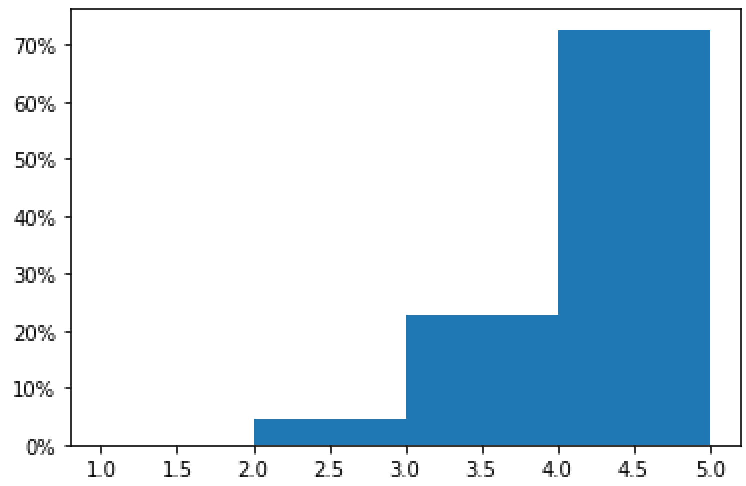


Infrastructure cost for Machine Learning is too high

Infrastructure cost for Machine Learning is too high and hard to justify the business use case. If data scientists can get an estimate of the model training, experimentation & maintenance cost- that will be important.

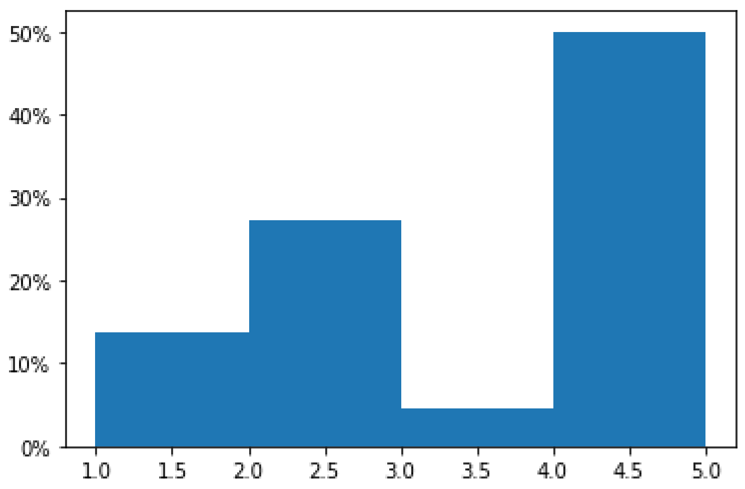


Building monitoring and alerting of ML systems is really hard.



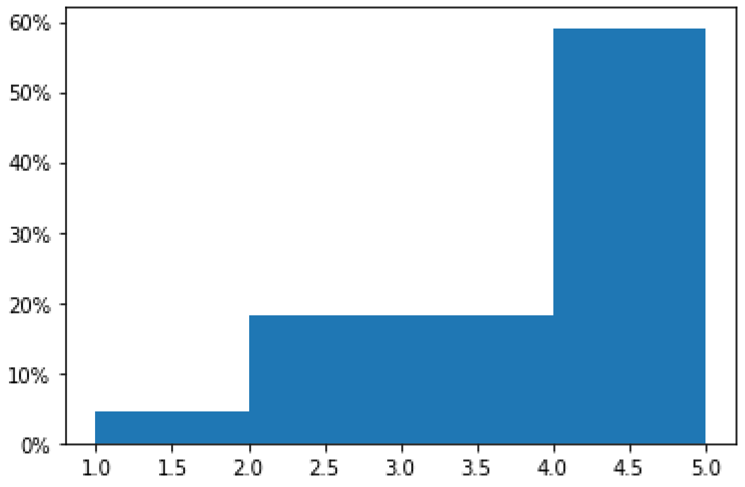
A/B infrastructure or tooling support is missing.

I am not able to run A/B tests due to lack of infrastructure or tooling support.



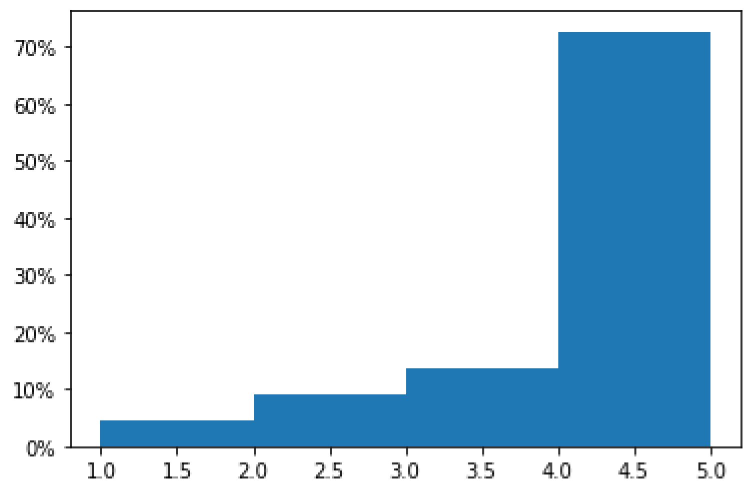
Maintaining retraining pipelines is really hard.

I am able to take my models to production but maintaining retraining pipelines is really hard.



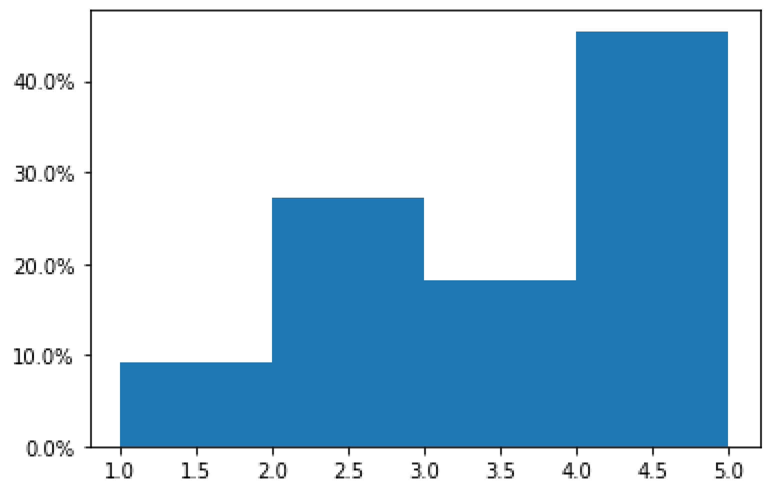
Debugging when things go wrong is very time consuming.

I am able to take my models to production but debugging when things go wrong is very time consuming.



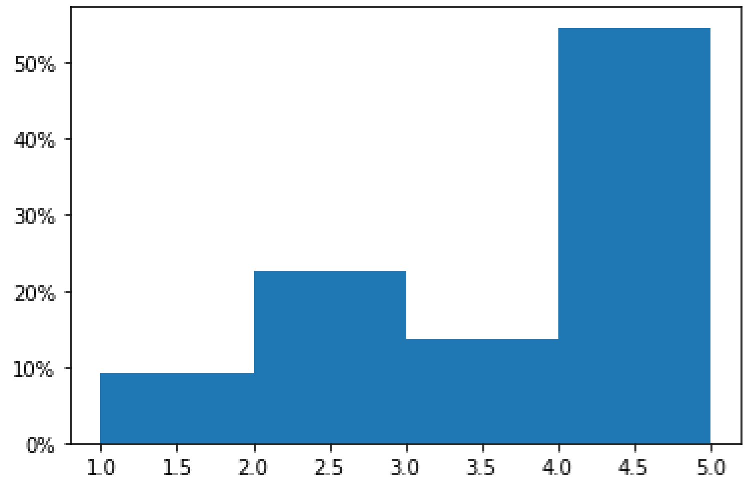
Problem Related to Jupyter Notebooks

Notebooks don’t encourage production quality code.

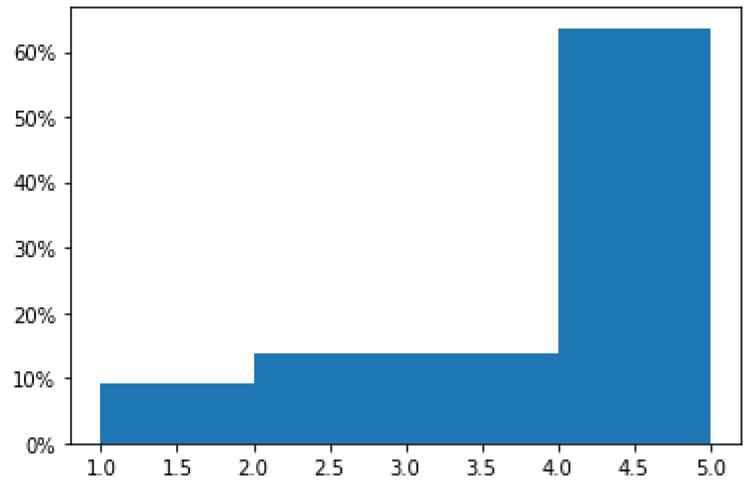


Non linear execution of experimental notebook is error prone

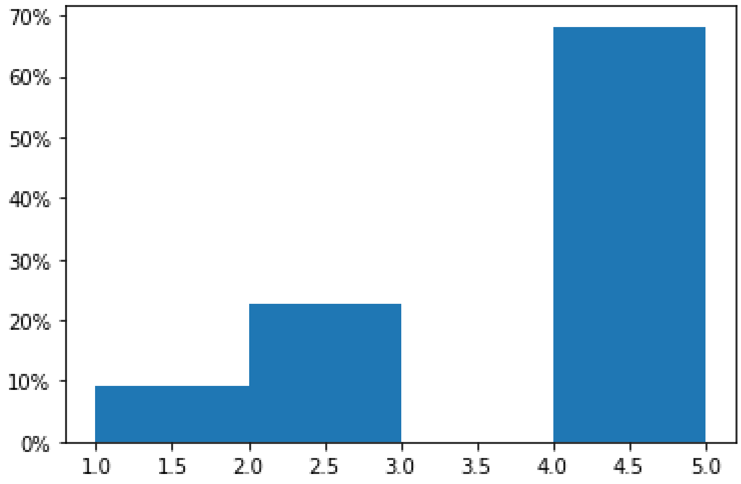
Non linear execution of experimental notebook code is very error prone and hard to maintain. Especially during collaboration, if one person executes notebook cells in a certain order and shares the notebook- the other cannot always reproduce the same.



Collaboration in notebooks is not easy.

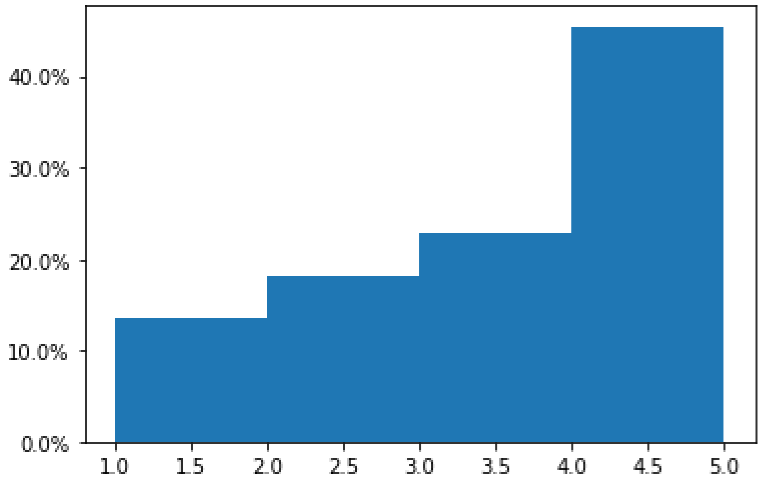


Version control and code review of notebooks is not easy.



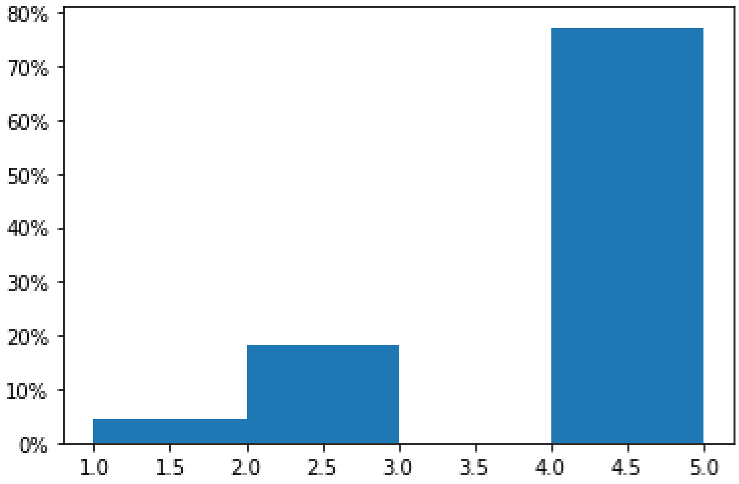
Notebooks present a security flaw

Notebooks present a security flaw because environment variables, database credentials, passwords etc. are usually part of the single notebook file.



Notebooks don’t have basic IDE productivity features

Notebooks don’t have basic IDE productivity features like high quality auto completes, code search across entire repo, linter & error suggestions.


80%
70%
60%
50%
40%
30%
20%
10%
0%
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0