MLOPS & KUBEFLOW

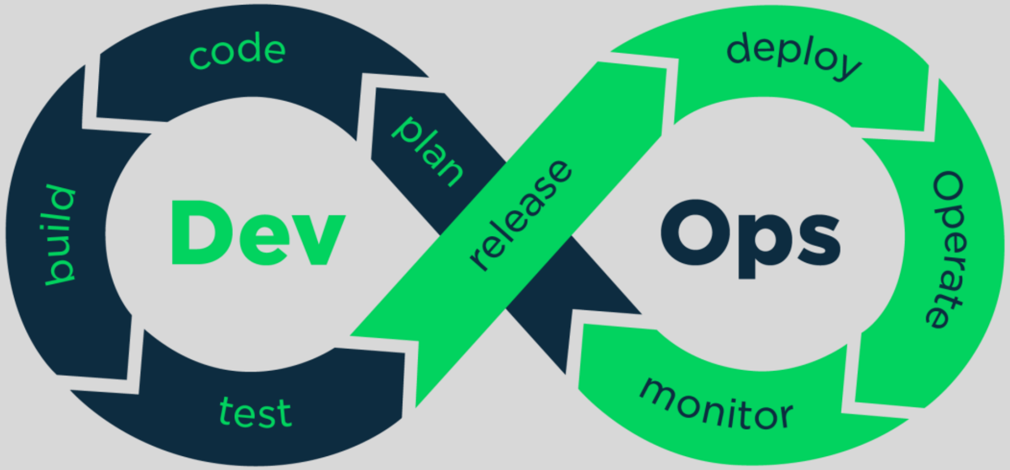
Grant Stevens | University of Bristol

WHAT IS MLOPS?

WHAT IS ~~MLOPS?~~ DEVOPS?

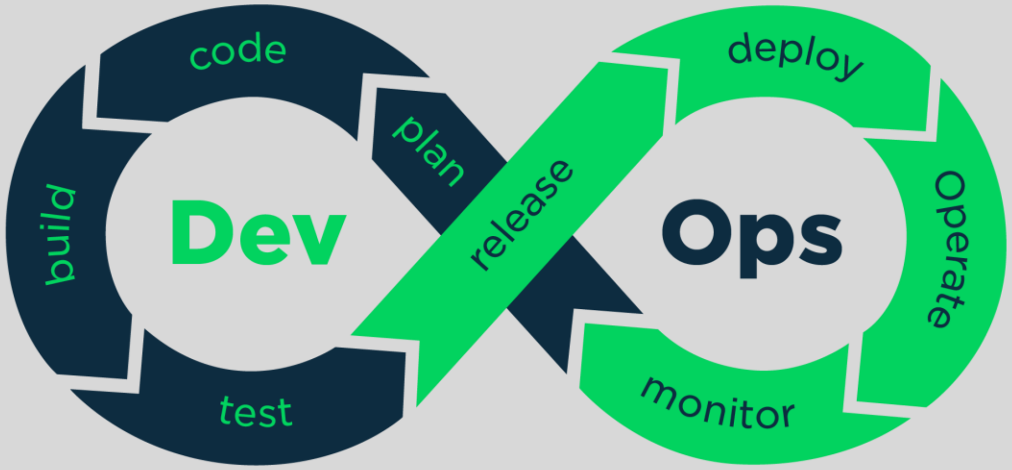
What is DevOps?

DevOps is a set of practices that works to automate and integrate the processes between software development and IT teams, so they can build, test, and release software faster and more reliably.



Increasing Scale

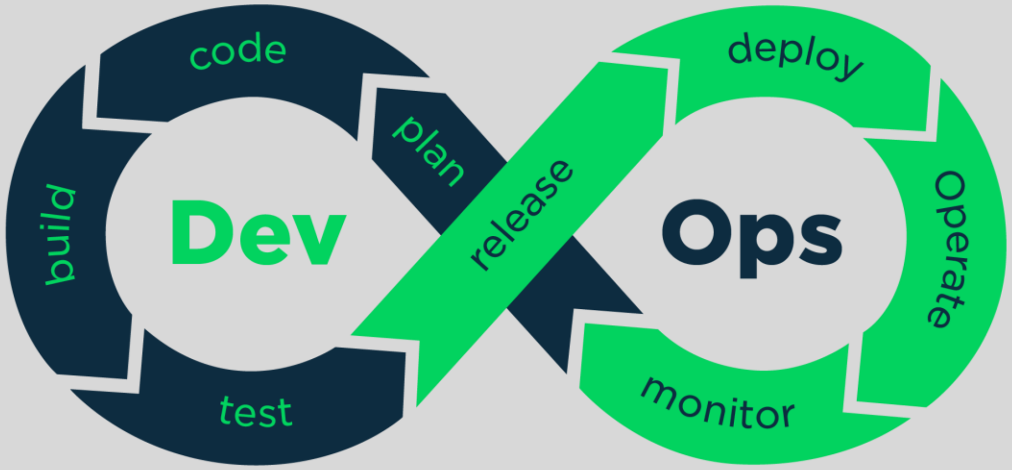
- No Comments
- No Version Control
- Minimal/Local Planning
- Skip Compatibility Checks



Increasing Scale

- No Comments
- No Version Control
- Minimal/Local Planning
- Skip Compatibility Checks

- Communication
- Shared Codebase
- Task Assignment
- Compatibility

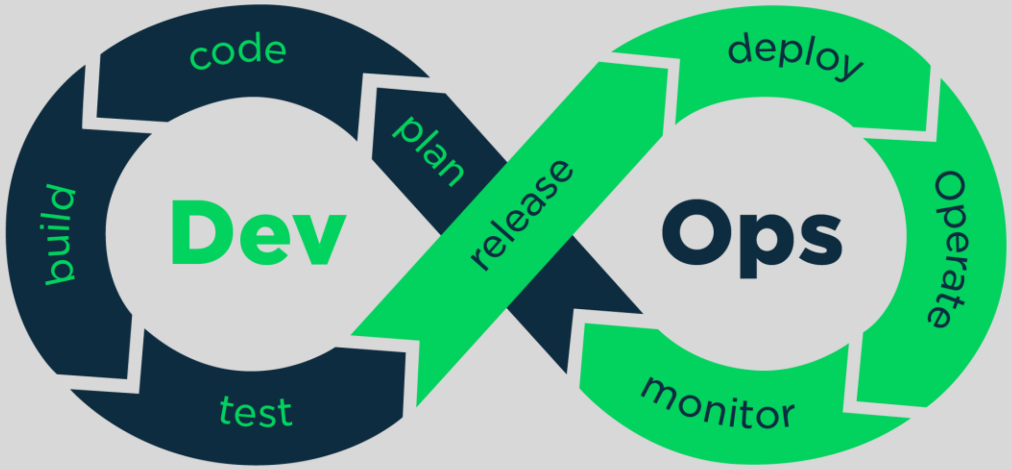


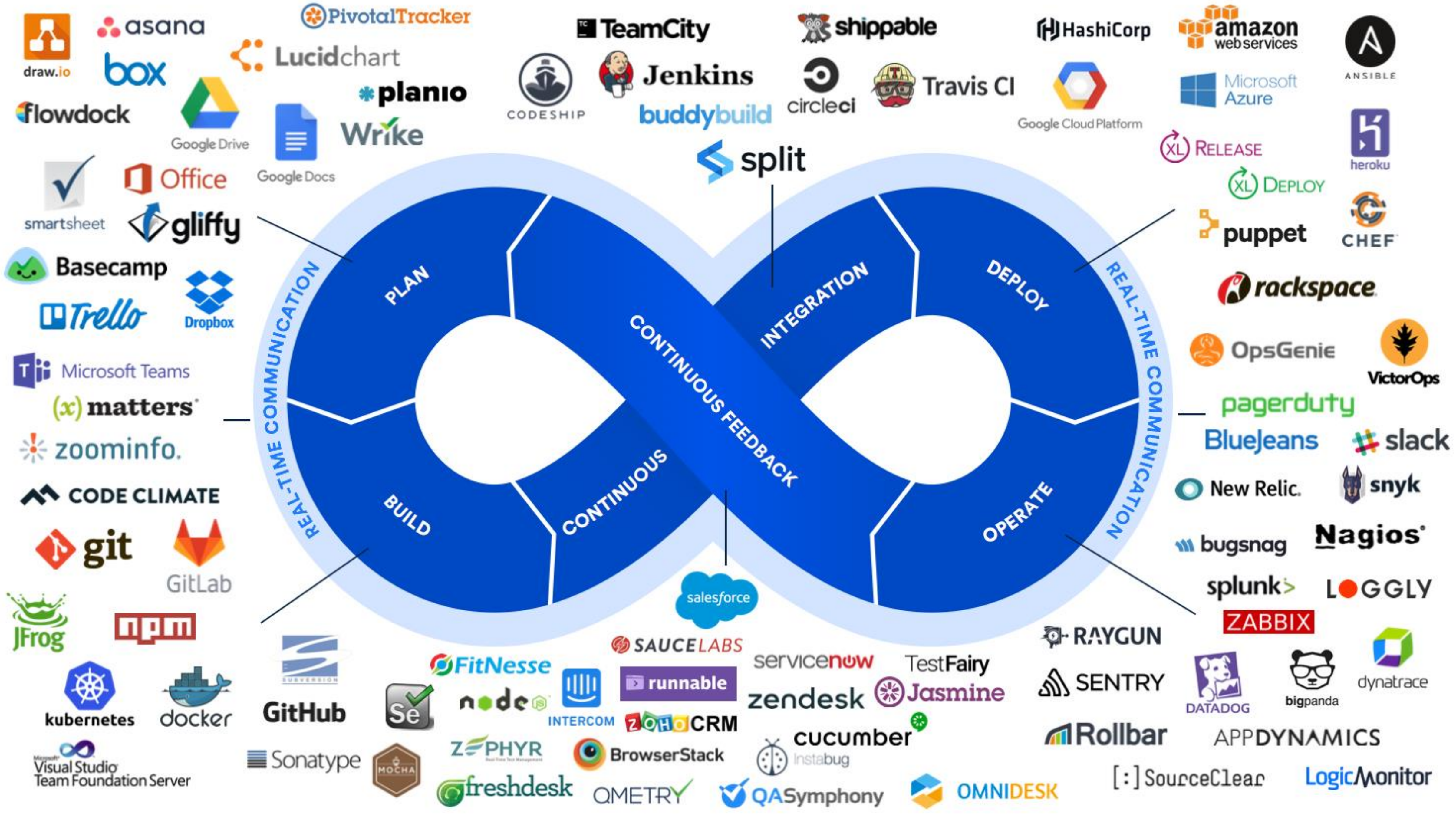
Increasing Scale

- No Comments
- No Version Control
- Minimal/Local Planning
- Skip Compatibility Checks

- Communication
- Shared Codebase
- Task Assignment
- Compatibility

How do you develop good software at industry scale with potential hundreds of developers?




PLAN
BUILD
CONTINUOUS
INTEGRATION
DEPLOY
OPERATE
CONTINUOUS FEEDBACK
REAL-TIME COMMUNICATION
REAL-TIME COMMUNICATION
draw.io
asana
box
PivotalTracker
Lucidchart
planio
flowdock
Google Drive
Google Docs
Wrike
smartsheet
Office
gliffy
Basecamp
Trello
Dropbox
Microsoft Teams
(x) matters
zoominfo.
CODE CLIMATE
git
GitLab
JFrog
npm
kubernetes
docker
Visual Studio
Team Foundation Server
TeamCity
CODESHIP
Jenkins
buddybuild
shippable
circleci
Travis CI
split
HashiCorp
amazon
webservices
Google Cloud Platform
Microsoft Azure
ANSIBLE
heroku
XL RELEASE
XL DEPLOY
puppet
CHEF
rackspace
OpsGenie
VictorOps
pagerduty
BlueJeans
slack
New Relic.
snyk
bugsnag
Nagios
splunk
LOGGLY
ZABBIX
salesforce
GitHub
Sonatype
MOCHA
FitNesse
node
INTERCOM
SAUCELABS
runnable
ZOHO CRM
ZEPHYR
freshdesk
BrowserStack
QMETRY
servicenow
zendesk
TestFairy
Jasmine
cucumber
Instabug
QASymphony
OMNIDESK
RAYGUN
SENTRY
Rollbar
DATADOG
bigpanda
dynatrace
APPDYNAMICS
[:] SourceClear
LogicMonitor

Why Start Here?

- The key questions that the ML community are now asking were asked by the software development industry a few decades ago.
- The result of many years of creating, optimising and testing solutions has led to what we now call DevOps.
- It is important to see how these challenges were overcome and why these solutions that work well for software development, are not sufficient for the development and deployment of ML applications.

Increasing Scale

Software Development

- No Comments
- No Version Control
- Minimal/Local Planning
- Skip Compatibility Checks

- Communication
- Shared Codebase
- Task Assignment
- Compatibility

How do you develop and deploy good software at industry scale with potentially hundreds of developers?

Machine Learning

Increasing Scale

Software Development

- No Comments
- No Version Control
- Minimal/Local Planning
- Skip Compatibility Checks

- Communication
- Shared Codebase
- Task Assignment
- Compatibility

How do you develop and deploy good software at industry scale with potentially hundreds of developers?

Machine Learning

Much of the work you have done/will do over the next few years will be here.

Increasing Scale

Software Development

- No Comments
- No Version Control
- Minimal/Local Planning
- Skip Compatibility Checks

- Communication
- Shared Codebase
- Task Assignment
- Compatibility

How do you develop and deploy good software at industry scale with potentially hundreds of developers?

Machine Learning

Much of the work you have done/will do over the next few years will be here.

The Group Project will give you an introductory experience doing ML in a team.

Increasing Scale

Software Development

- No Comments
- No Version Control
- Minimal/Local Planning
- Skip Compatibility Checks

- Communication
- Shared Codebase
- Task Assignment
- Compatibility

How do you develop and deploy good software at industry scale with potentially hundreds of developers?

Machine Learning

Much of the work you have done/will do over the next few years will be here.

The Group Project will give you an introductory experience doing ML in a team.

How do you develop and deploy accurate and reliable ML models at industry scale with potentially hundreds of developers?

Is MLOps just more DevOps?

Is MLOps just more DevOps?

Software = compile(code, environment)

Is MLOps just more DevOps?

Software = compile(code, environment)

Is MLOps just more DevOps?

Code → Test → Deploy → Monitor → Code

Software = compile(code, environment)

whereas

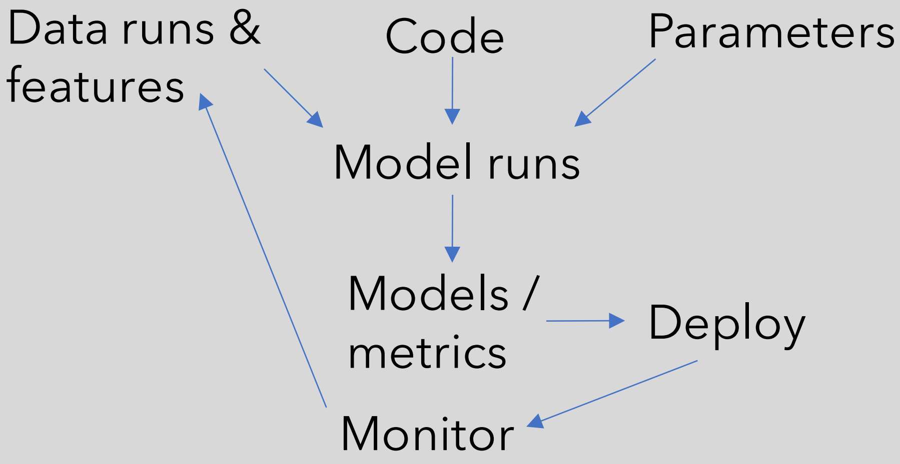
Model = train(data, params, code, environment)

Is MLOps just more DevOps?

Software = compile(code, environment)

whereas

Model = train(data, params, code, environment)

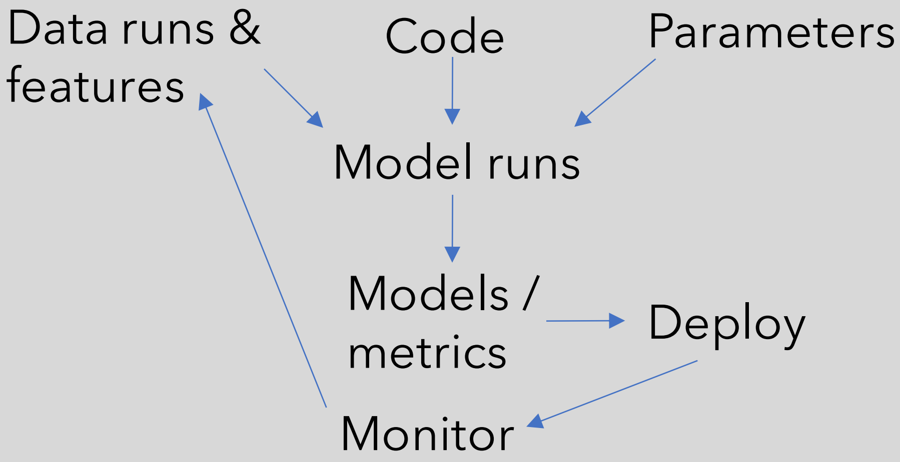


Is MLOps just more DevOps?

Software = compile(code, environment)

whereas

Model = train(data, params, code, environment)



Train is:

- Sometimes non-deterministic
- Usually more expensive than compile (limiting the number of changes)
- Sometimes distributed across multiple machines (adds complexity)

Potential Issues

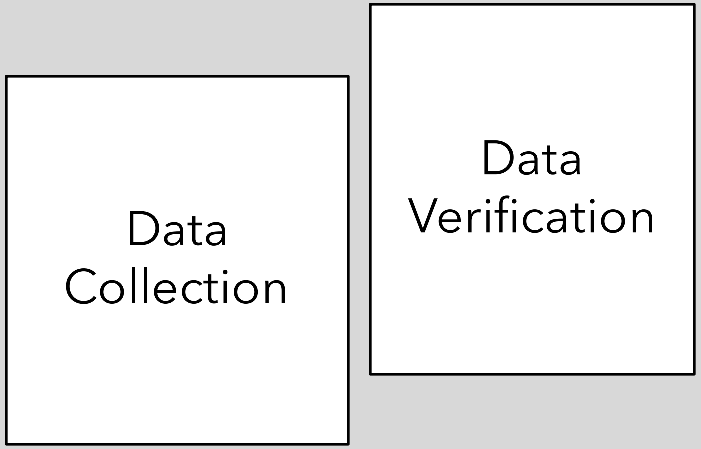
- If you have hundreds of models being trained at once, how do you choose which one to use in production?
- How do you know that multiple people aren't testing the same hyperparameters?
- If you only have capacity for training 10 models at one time, who should get priority?
- How do you share models?
- Are there the equivalent of unit tests for ML models?
- How do you effectively bridge the gap between Data Scientists, who likely won't know much of the software design, and the Software Developers who likely won't know much of the ML side of things?

And many more…

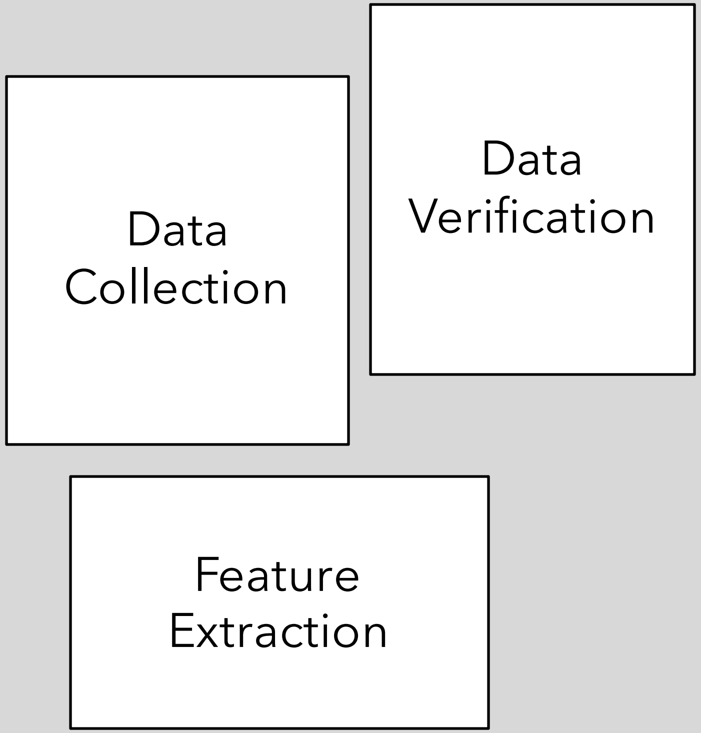
What are the key parts of infrastructure required for ML systems?

Data Collection

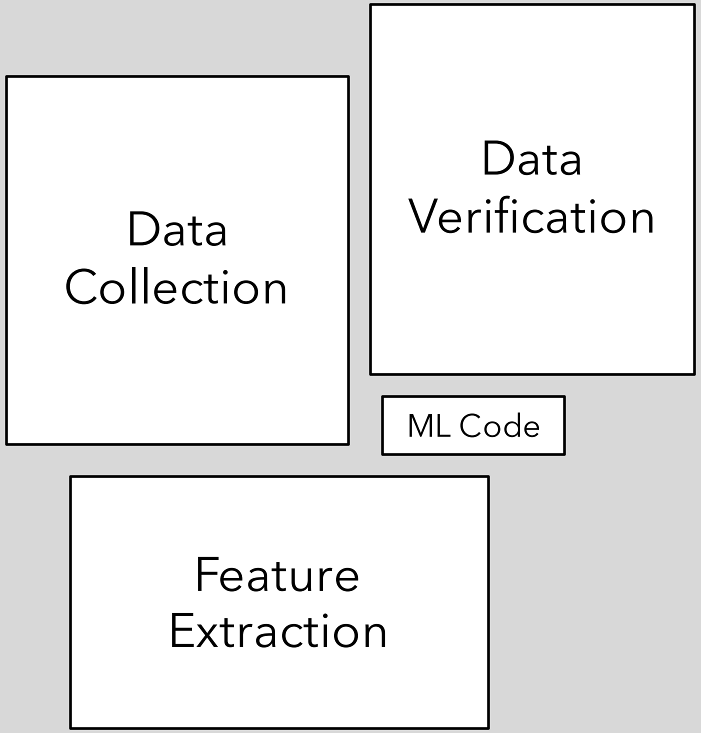
Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems*. 2015.



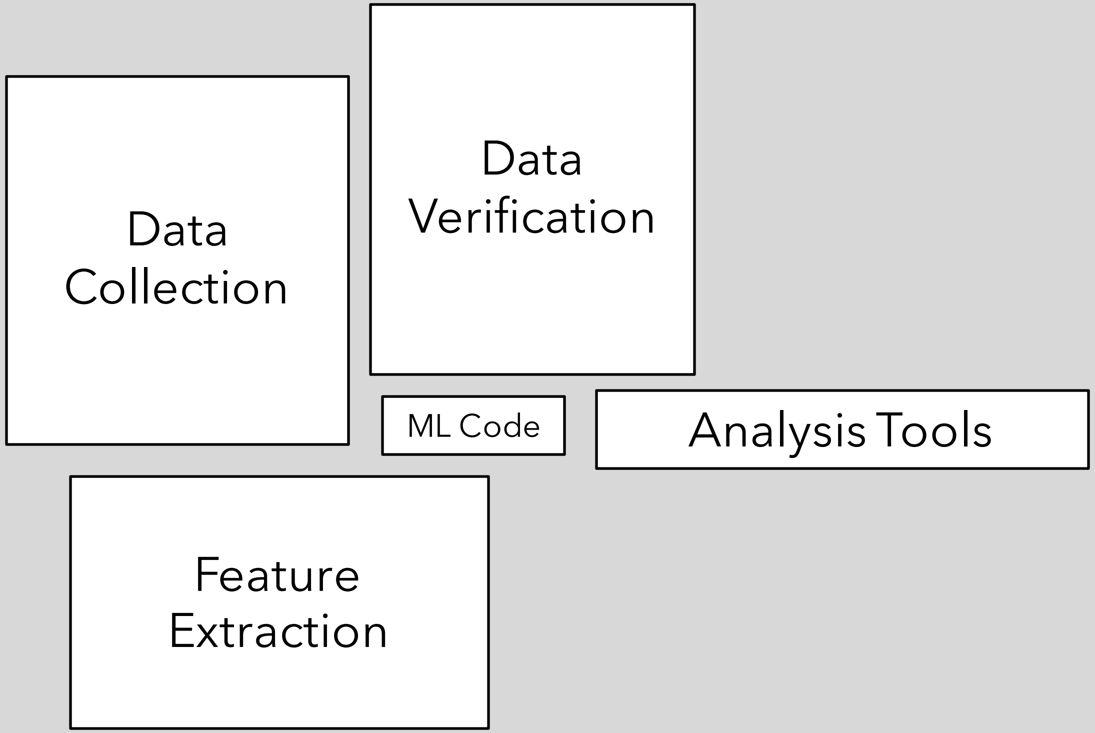
Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems*. 2015.



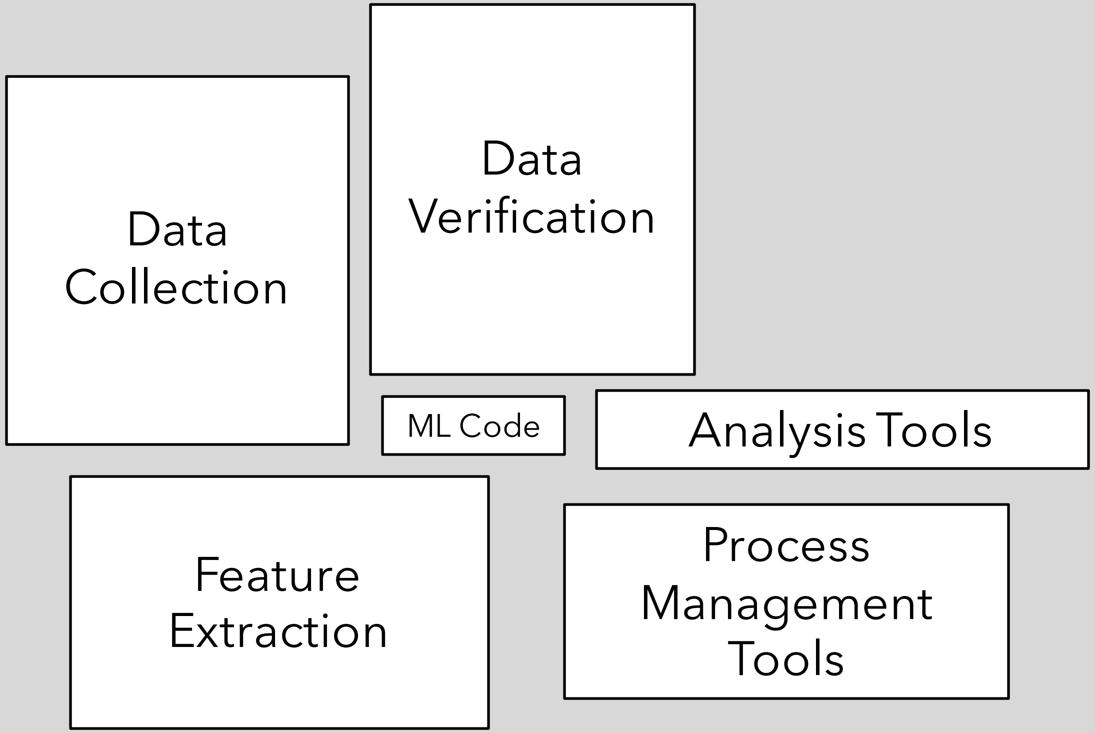
Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems*. 2015.



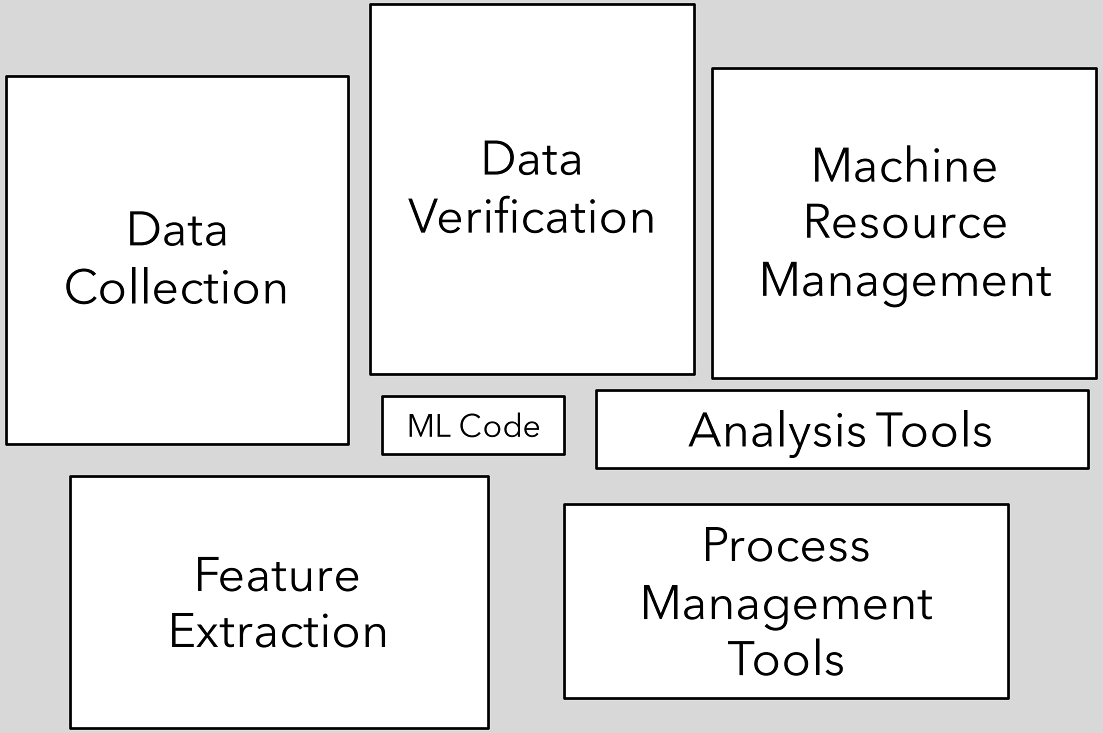
Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems*. 2015.



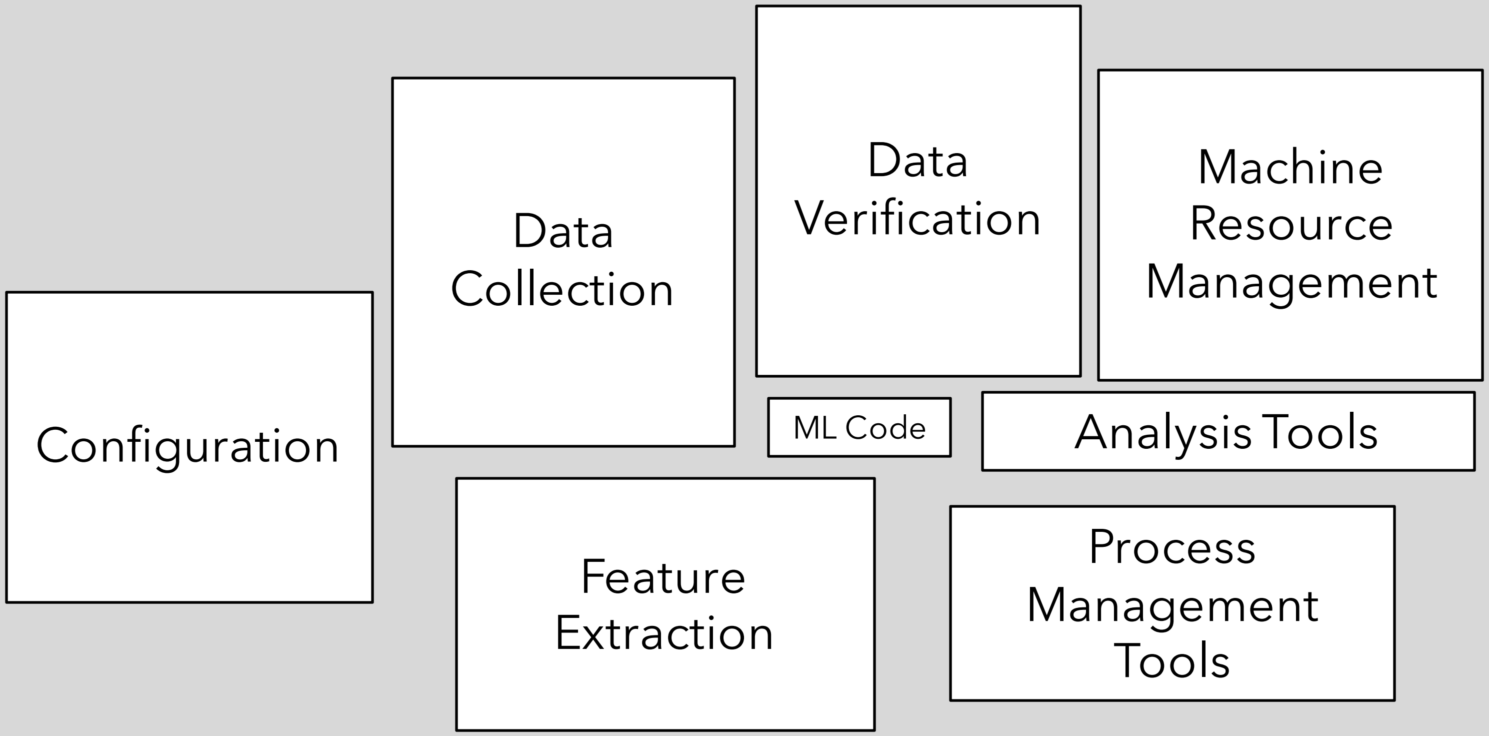
Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems*. 2015.



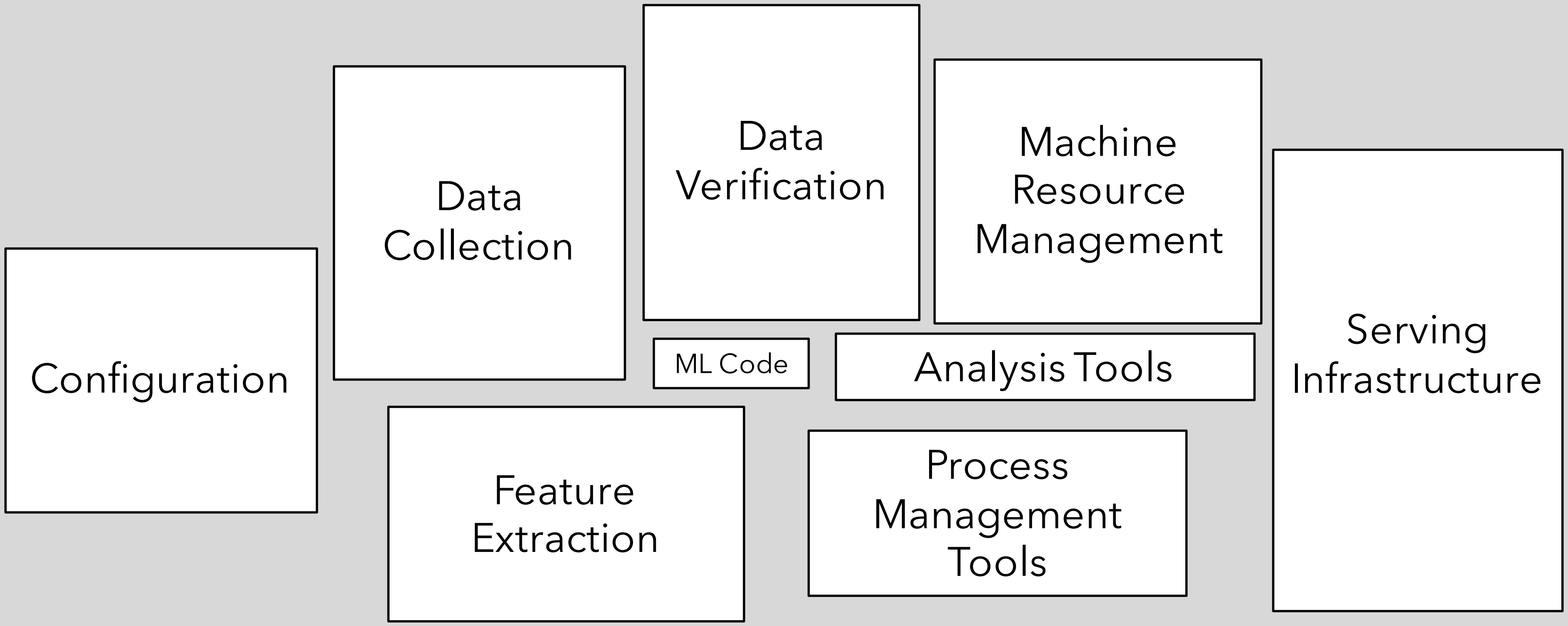
Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems*. 2015.



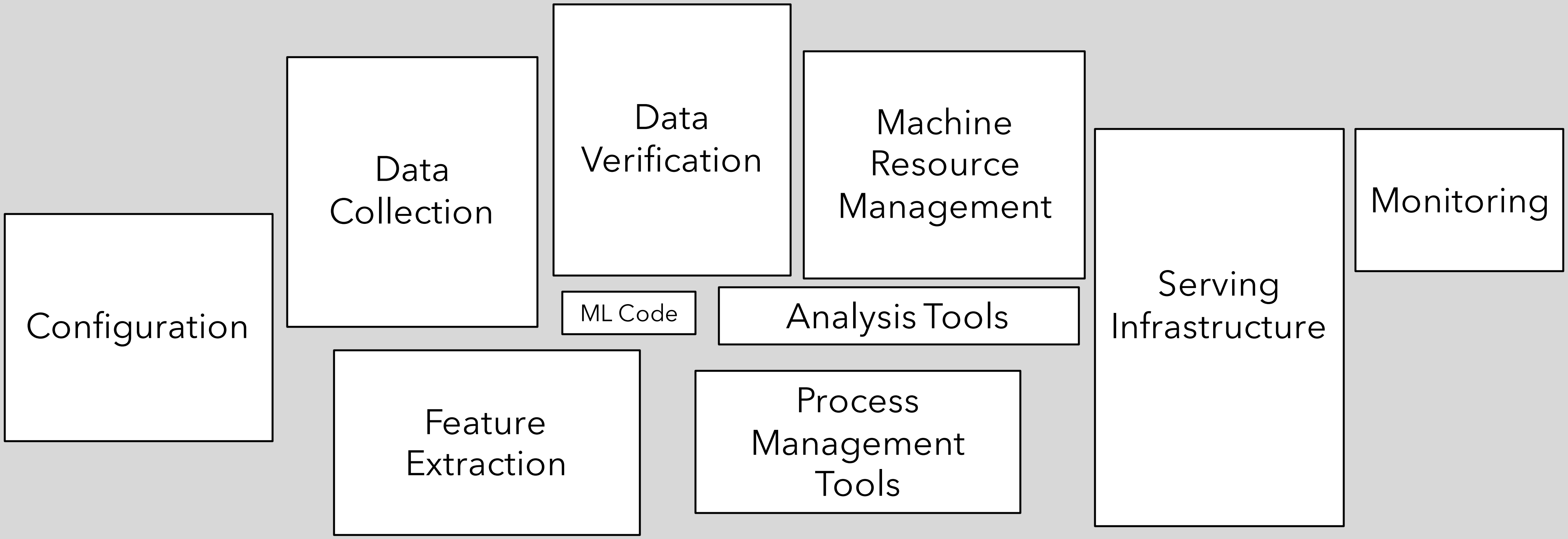
Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems*. 2015.



Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems*. 2015.



Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems*. 2015.



Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems*. 2015.

Kubeflow

Kubeflow is an open source project that contains a curated set of compatible tools and frameworks specific for ML.

Its built on top of Kubernetes, allowing it to run consistently across different environments.

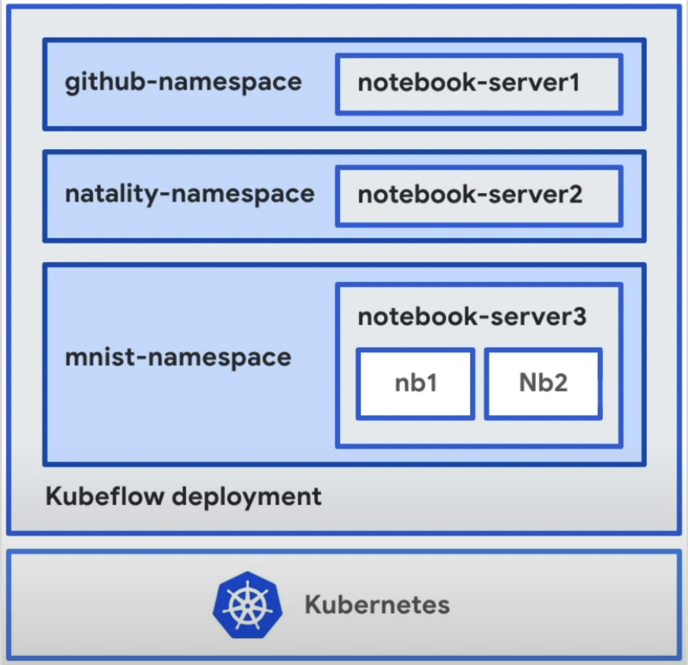
Kubeflow is built around 3 principles:
- Composability (you choose what works for you)
- Portability (Run any part of your workflow wherever you are running Kubeflow)
- Scalability (Your project can access more resources when needed and release them when not)

Jupyter Notebooks

Kubeflow has Jupyter Notebooks built into the system.

Using the notebooks on Kubeflow (rather than locally) allows for the following benefits:

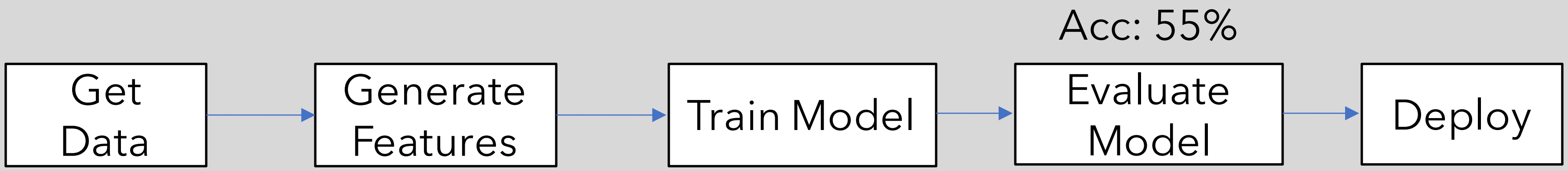
- Integration with the other Kubeflow components
- Access control and authentication (both for developers and users)
- Automated resource allocation



Pipelines

Typical machine learning workflows involve multiple steps, which can become complicated to keep track of when they are arranged in multiple scripts or notebooks.

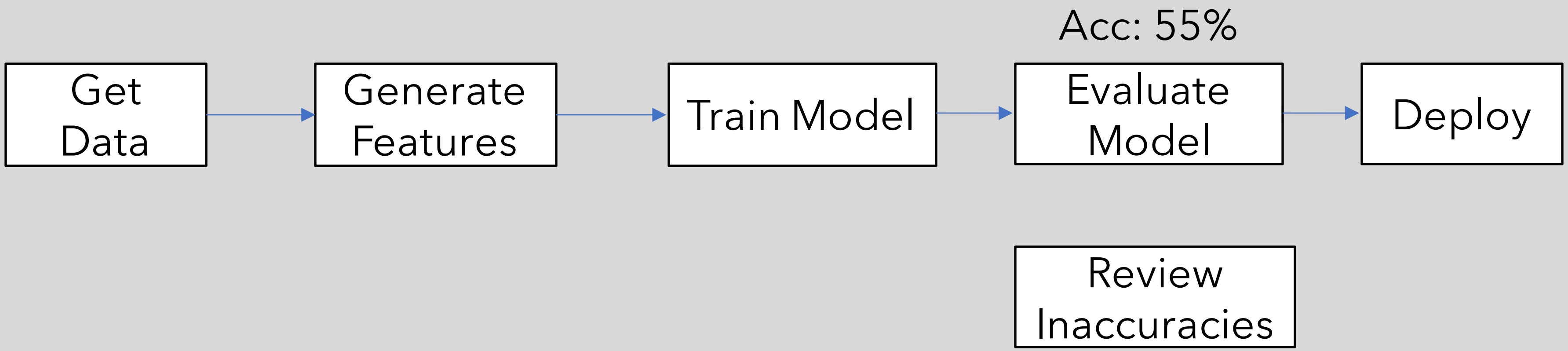
Kubeflow Pipelines allows developers to codify their ML workflows so that they are easily composable, shareable and reproducible.



Pipelines

Typical machine learning workflows involve multiple steps, which can become complicated to keep track of when they are arranged in multiple scripts or notebooks.

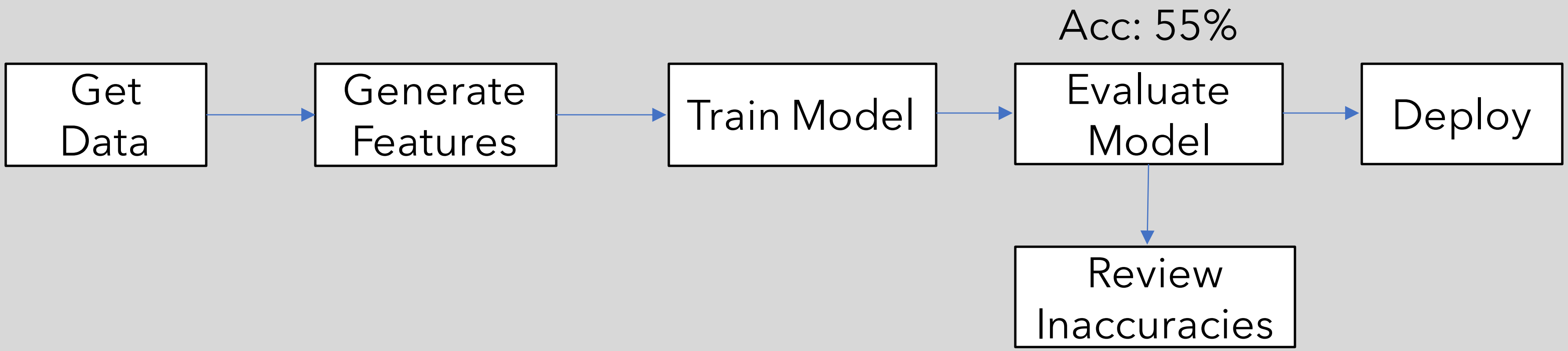
Kubeflow Pipelines allows developers to codify their ML workflows so that they are easily composable, shareable and reproducible.



Pipelines

Typical machine learning workflows involve multiple steps, which can become complicated to keep track of when they are arranged in multiple scripts or notebooks.

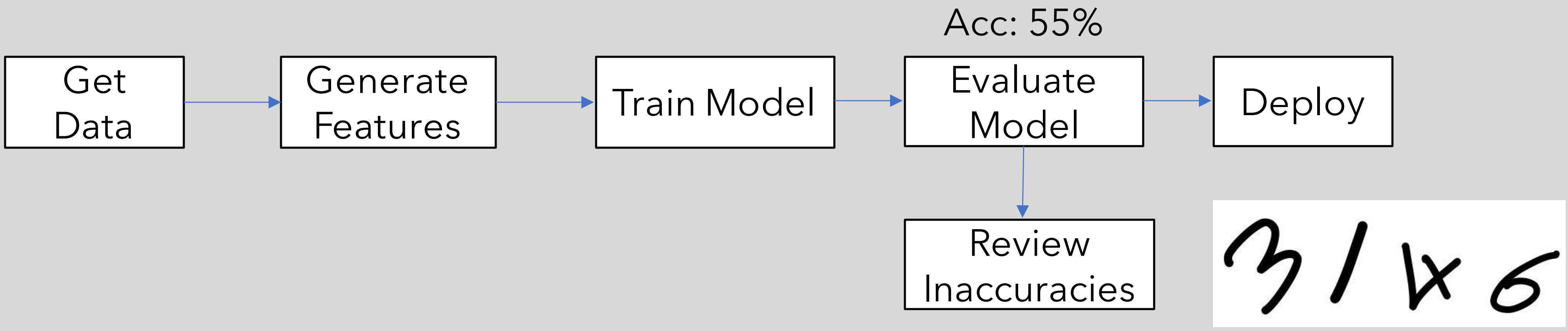
Kubeflow Pipelines allows developers to codify their ML workflows so that they are easily composable, shareable and reproducible.



Pipelines

Typical machine learning workflows involve multiple steps, which can become complicated to keep track of when they are arranged in multiple scripts or notebooks.

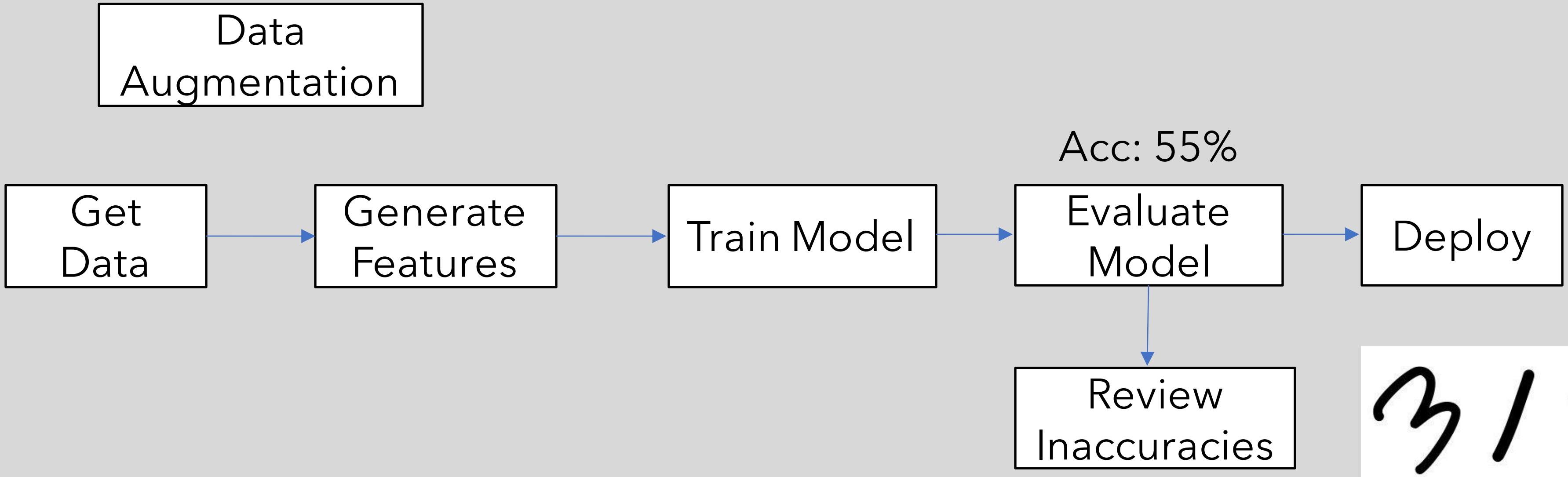
Kubeflow Pipelines allows developers to codify their ML workflows so that they are easily composable, shareable and reproducible.

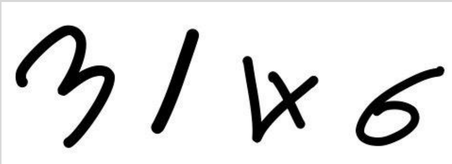


Pipelines

Typical machine learning workflows involve multiple steps, which can become complicated to keep track of when they are arranged in multiple scripts or notebooks.

Kubeflow Pipelines allows developers to codify their ML workflows so that they are easily composable, shareable and reproducible.

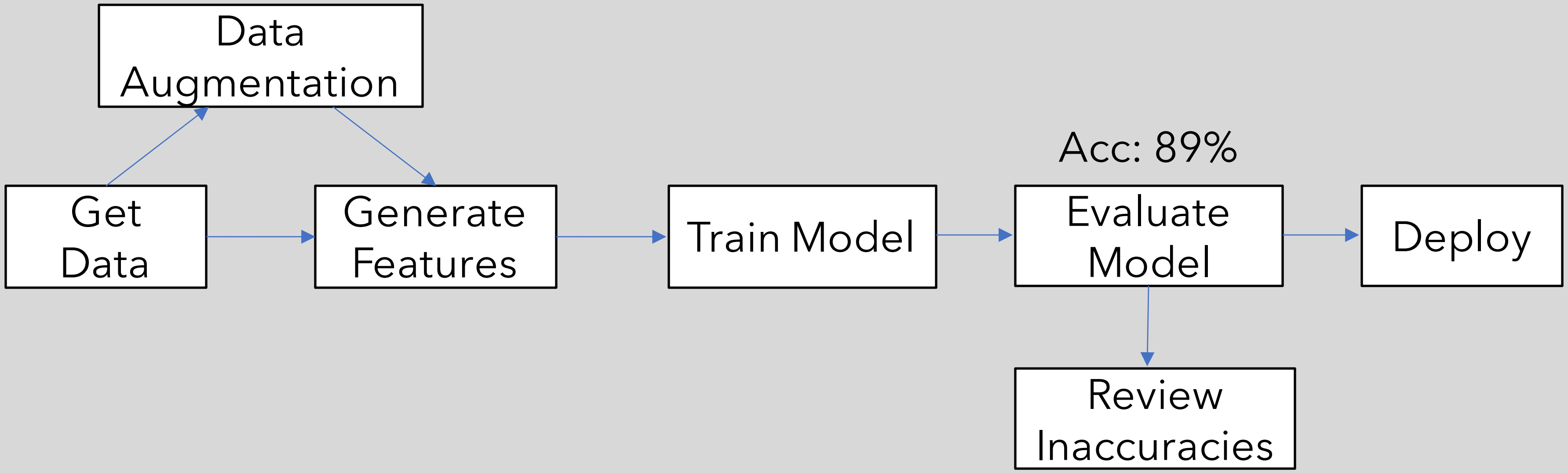




Pipelines

Typical machine learning workflows involve multiple steps, which can become complicated to keep track of when they are arranged in multiple scripts or notebooks.

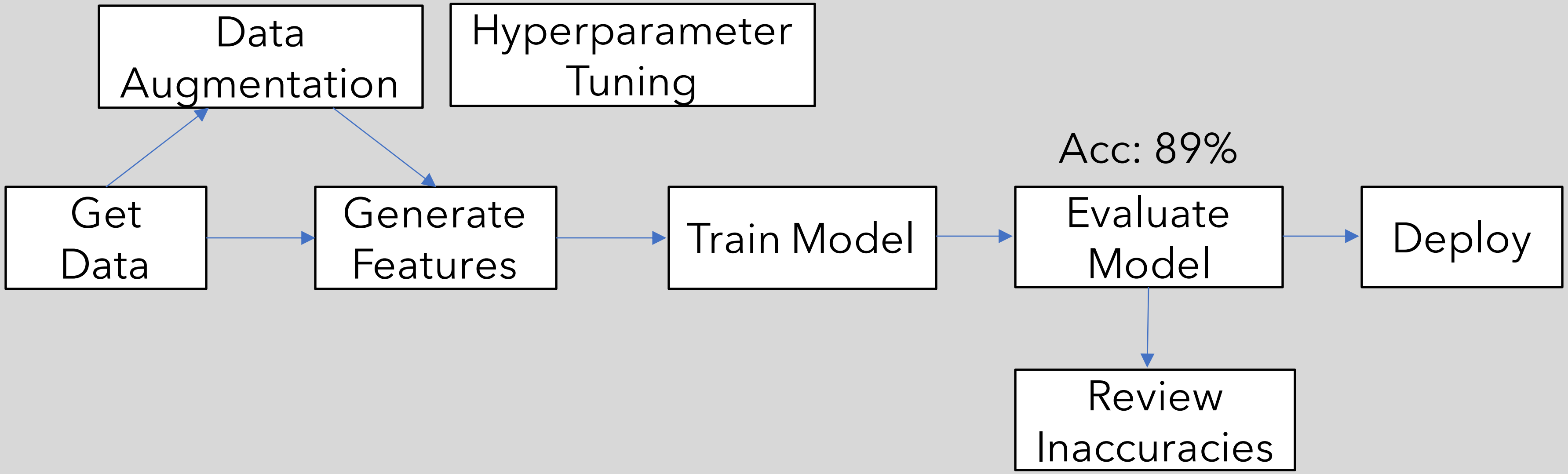
Kubeflow Pipelines allows developers to codify their ML workflows so that they are easily composable, shareable and reproducible.



Pipelines

Typical machine learning workflows involve multiple steps, which can become complicated to keep track of when they are arranged in multiple scripts or notebooks.

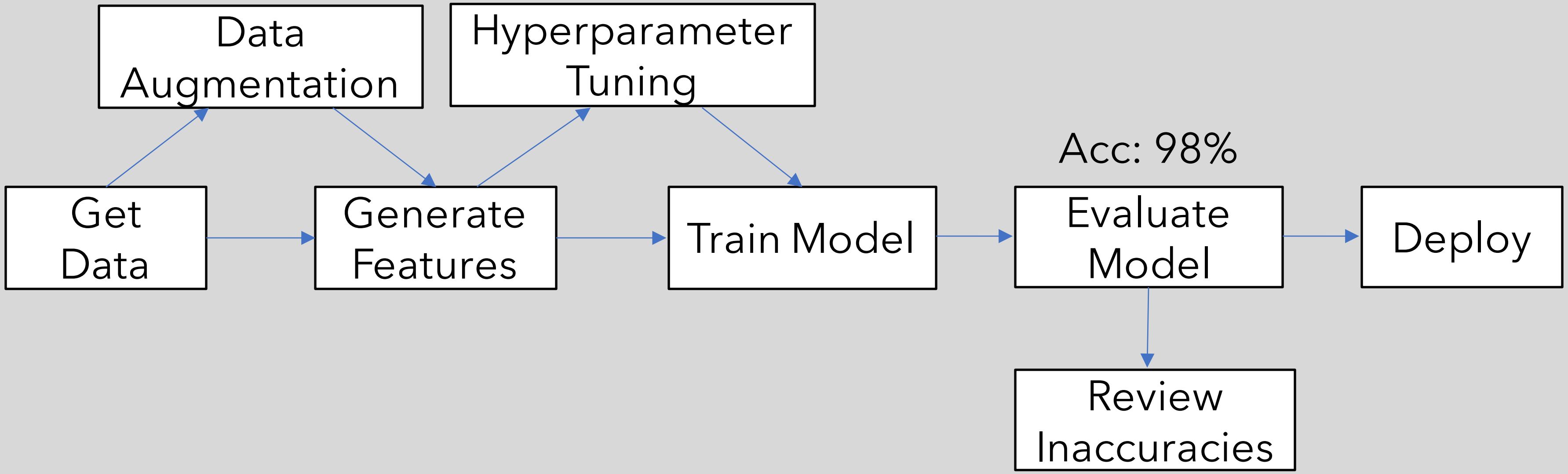
Kubeflow Pipelines allows developers to codify their ML workflows so that they are easily composable, shareable and reproducible.



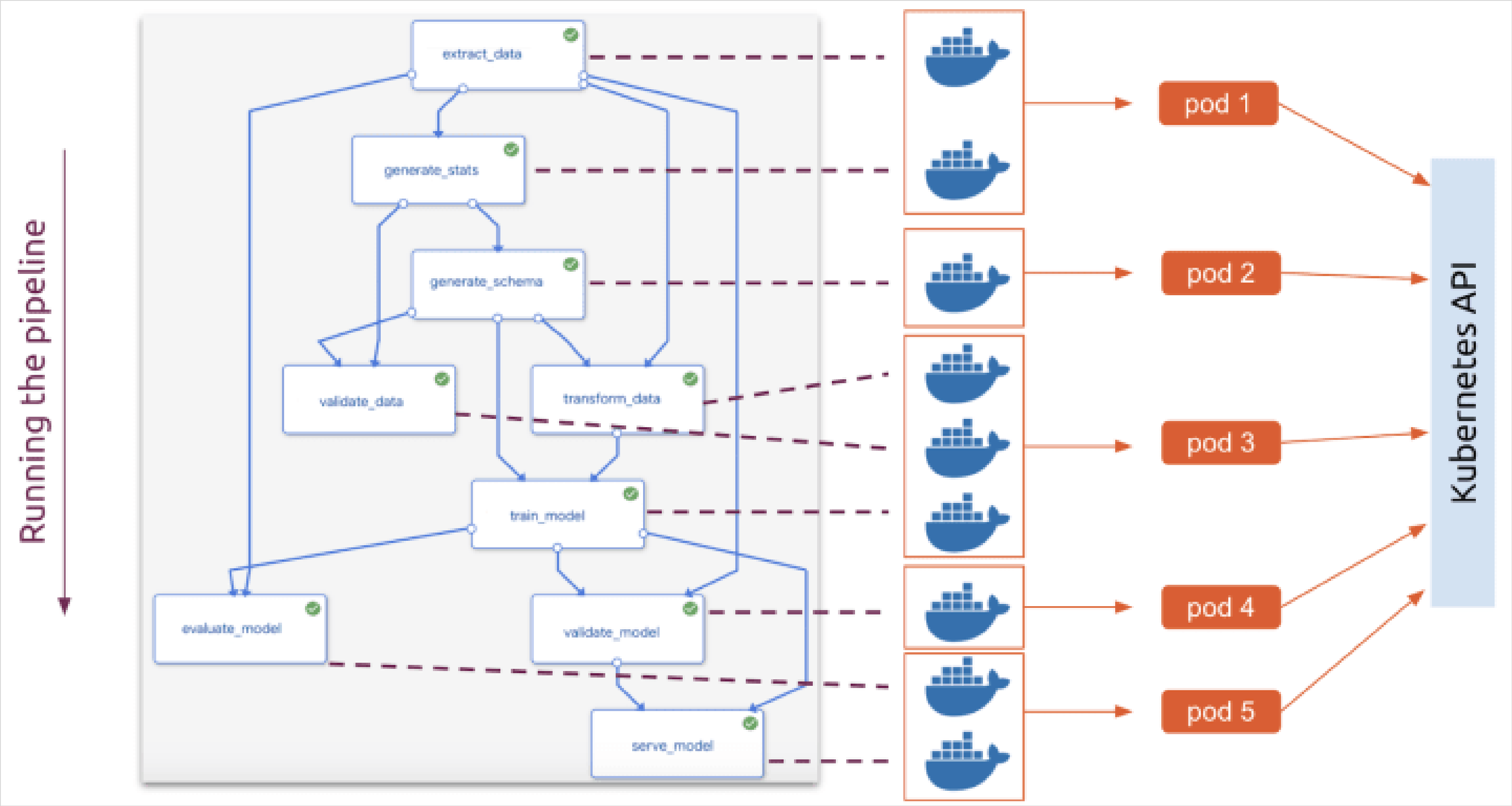
Pipelines

Typical machine learning workflows involve multiple steps, which can become complicated to keep track of when they are arranged in multiple scripts or notebooks.

Kubeflow Pipelines allows developers to codify their ML workflows so that they are easily composable, shareable and reproducible.

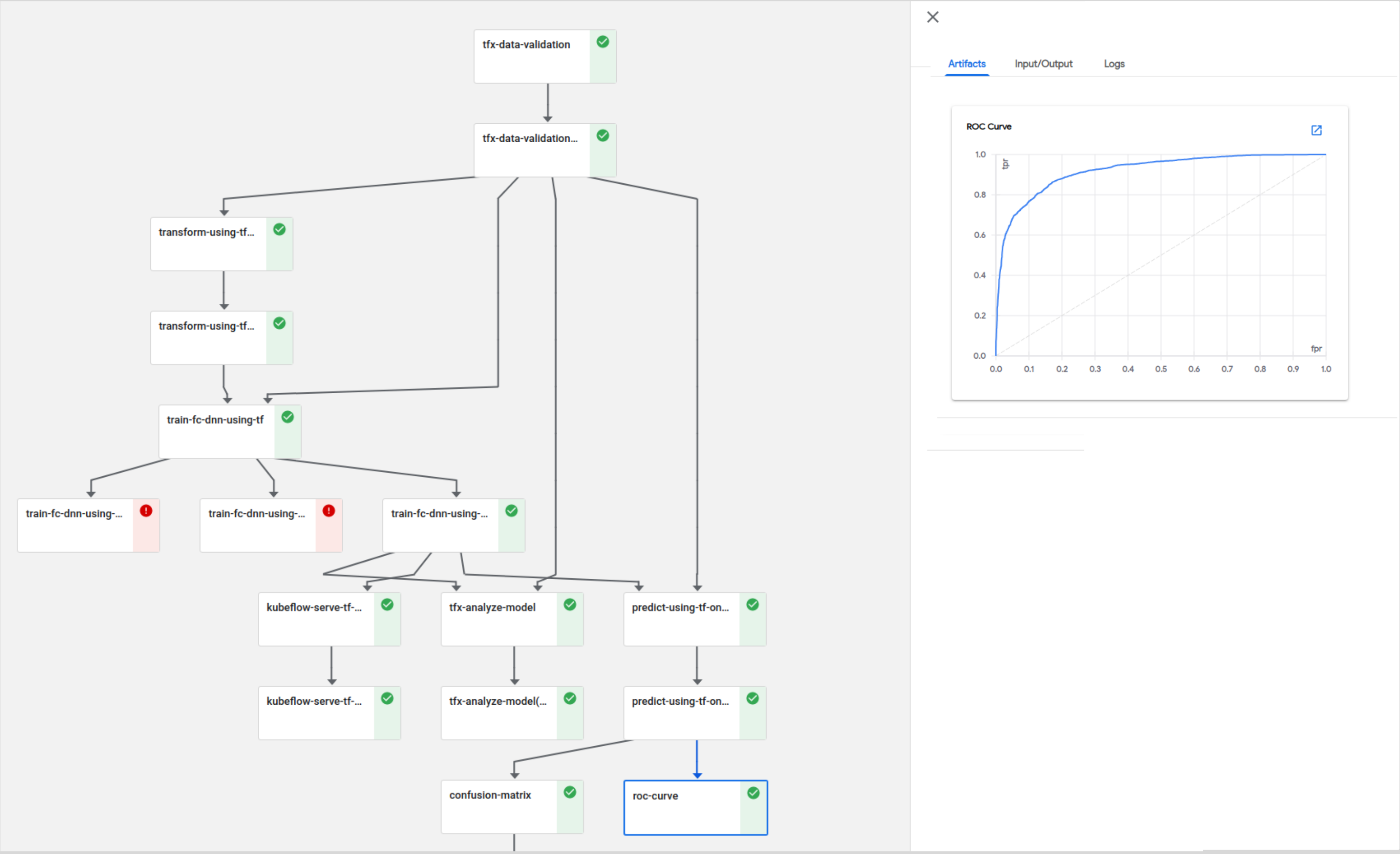


Pipelines



Running the pipeline
extract_data
generate_stats
generate_schema
validate_data
transform_data
train_model
evaluate_model
validate_model
serve_model
pod 1
pod 2
pod 3
pod 4
pod 5
Kubernetes API

Pipelines



tfx-data-validation
tfx-data-validation...
transform-using-tf...
transform-using-tf...
train-fc-dnn-using-tf
train-fc-dnn-using-...
train-fc-dnn-using-...
train-fc-dnn-using-...
kubeflow-serve-tf-...
tfx-analyze-model
predict-using-tf-on...
kubeflow-serve-tf-...
tfx-analyze-model(...
predict-using-tf-on...
confusion-matrix
roc-curve
Artifacts
Input/Output
Logs
ROC Curve
tpr
fpr
1.0
0.8
0.6
0.4
0.2
0.0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

Katib

Katib is Kubeflow's built in hyperparameter tuner.

Given that hyperparameter tuning locally is done through a big for loop, using Kubeflow can make the whole process more optimised.

```
spec:
  parallelTrialCount: 3
  maxTrialCount: 12
  maxFailedTrialCount: 3
  objective:
    type: maximize
    goal: 0.99
    objectiveMetricName: accuracy_1
  algorithm:
    algorithmName: random
```

How many at one time

Total number of tests

Stop after this many fails

Metric to improve on and auto-stop criteria

Algorithm for choosing next value to test (grid search, Bayesian optimisation, …)

Katib

Katib is Kubeflow's built in hyperparameter tuner.

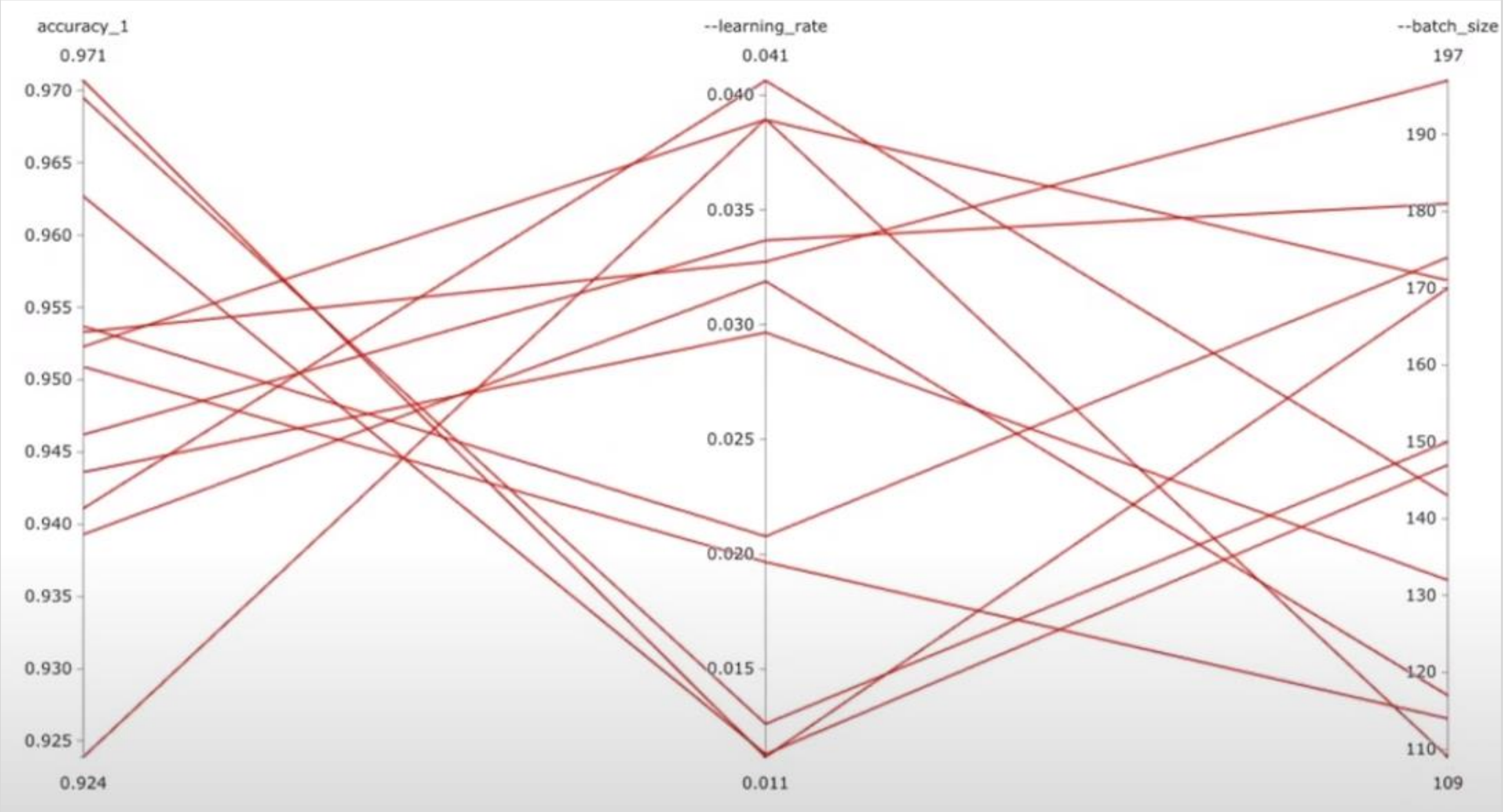
Given that hyperparameter tuning locally is done through a big for loop, using Kubeflow can make the whole process more optimised.

Choose hyperparameter and range to search

```yaml
parameters:
    - name: --learning_rate
      parameterType: double
      feasibleSpace:
        min: "0.01"
        max: "0.05"

    - name: --batch_size
      parameterType: int
      feasibleSpace:
        min: "100"
        max: "200"
```

Katib


accuracy_1
0.971
0.970
0.965
0.960
0.955
0.950
0.945
0.940
0.935
0.930
0.925
0.924
--learning_rate
0.041
0.040
0.035
0.030
0.025
0.020
0.015
0.011
--batch_size
197
190
180
170
160
150
140
130
120
110
109

Extras for your Pipelines

- There are a number of external tools that can be added to your Kubeflow pipeline to improve your ML models.

Method	Models	Explanations	Classification	Regression	Tabular	Text	Images
ALE	BB	global	✔	✔	✔		
Anchors	BB	local	✔		✔	✔	✔
CEM	BB* TF/Keras	local	✔		✔		✔
Counterfactuals	BB* TF/Keras	local	✔		✔		✔
Prototype Counterfactuals	BB* TF/Keras	local	✔		✔		✔
Integrated Gradients	TF/Keras	local	✔	✔	✔	✔	✔
Kernel SHAP	BB	local global	✔	✔	✔		
Tree SHAP	WB	local global	✔	✔	✔		

Model Confidence

These algorithms provide **instance-specific** scores measuring the model confidence for making a particular prediction.

Method	Models	Classification	Regression	Tabular	Text	Images	Categorical Features	Train set required
Trust Scores	BB	✔		✔	✔ (1)	✔(2)		Yes
Linearity Measure	BB	✔	✔	✔		✔		Optional

https://github.com/seldonio/alibi

Extras for your Pipelines

Method	Models	Explanations	Classification	Regression	Tabular	Text	Images
Anchors	BB	local	✔		✔	✔	✔

(a) Original image (b) Anchor for "beagle" (c) Images where Inception predicts $P(\text{beagle}) > 90\%$

https://github.com/seldonio/alibi

Time to open up Kubeflow