To Test or Not To Test

<In Machine Learning Projects>

¿Por qué esta

Testing: 9 [- details]

Notes:

* El tema de tests en codigo de experimentos y código re machine learning requiere mas discusión. Hay algo de test sobre el backend de la API, pero es mínimo.

Testing: 8 [- details]

nit tests para el backend.

s, aunque estamos mucho mejor que el período anterior.

para los experimentos.

Falta:

odas las partes del código que se pueden testear (API, Backend).

Testing: 5 [- details]

Notes:

Se hace test unitario en las API, pero no en el resto del código.

Por la naturaleza del código (Machine Learning) los tests en predictores son muy difícil de hacerlo.

Falta la práctica de cuando se encuentra un error o bug, evidenciar con un test que se ha corregido.

~~**Testing:** 8 [- details]~~

Disabled aspect

Notes:

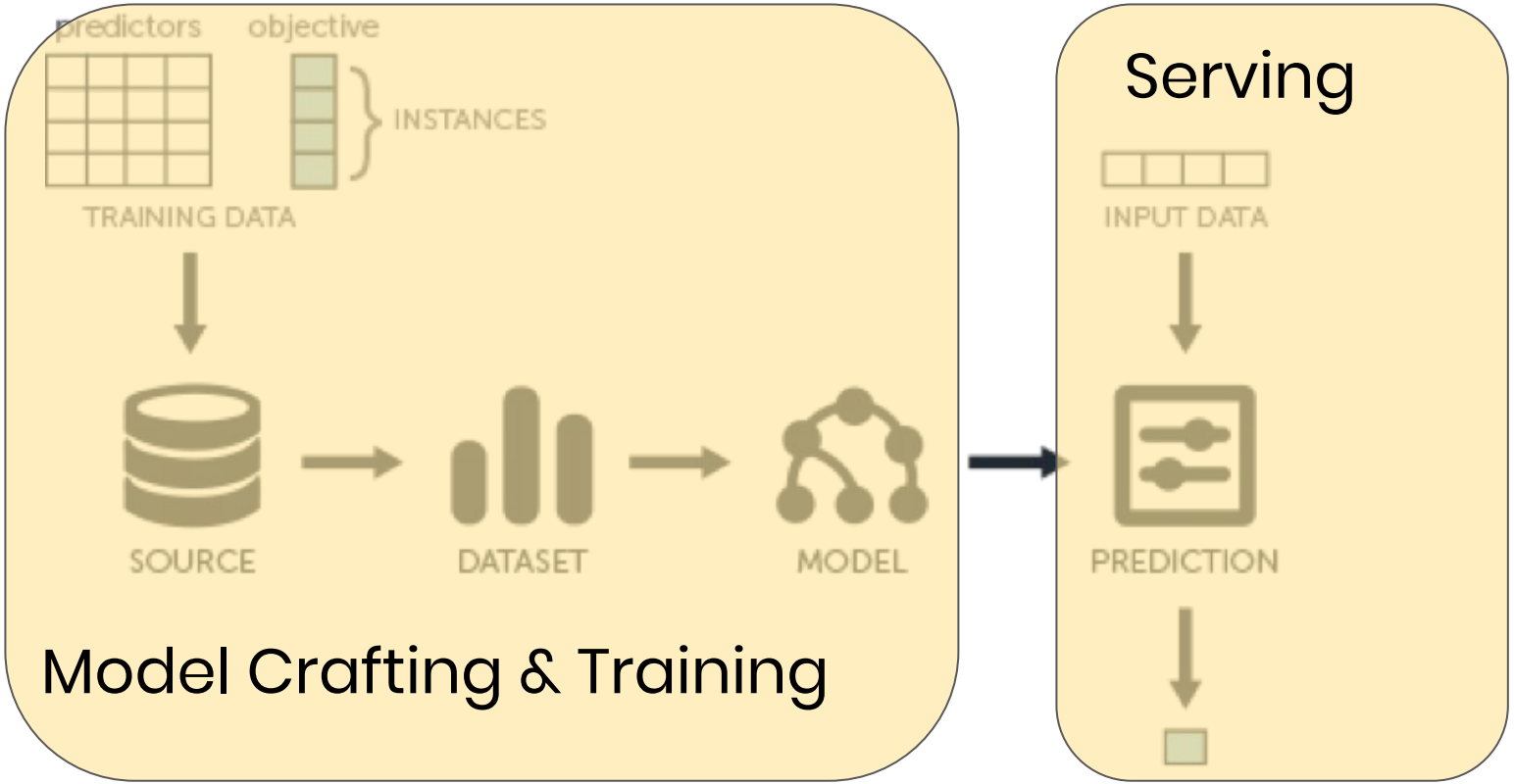
No hay tests, pero hay muy poco código productivo, así que no es grave. La garantía de la corrección de las soluciones está más ligada a la revisión de código.

Testing: 1 [- details]

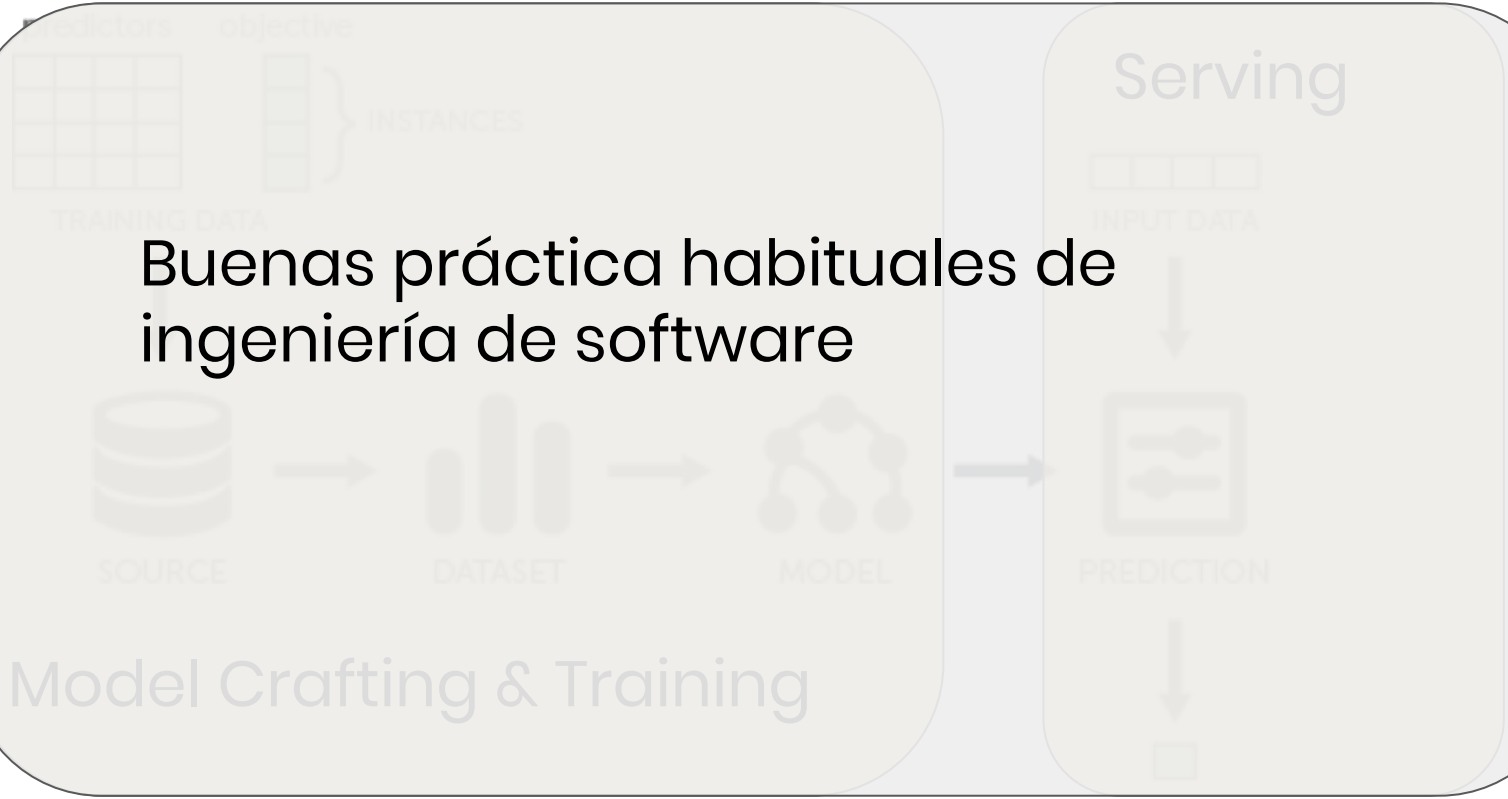
Notes:

no existen unit test, tampoco sabria como aplicarlo a los experimentos.

> Se asume


predictors
objective
INSTANCES
TRAINING DATA
SOURCE
DATASET
MODEL
Model Crafting & Training
Serving
INPUT DATA
PREDICTION

> Se asume



Buenas práctica habituales de ingeniería de software

> 1º Invitación: Criterios

Fases o estadíos

1. **Research Only projects / Discovery:**
   the team is working towards validating feasibility, or discovering opportunities
2. **Production in the RoadMap**
   the team is working focused on reaching production in the short-mid range
3. **Already in production**
   the team is maintaining, upgrading and monitoring an initiative that's already in production

¿Para qué?

- Expandir y ampliar la mirada sobre el tema
- Sync con líderes en la industria
- Proponer algunos criterios

> 2° Invitación: Ampliar la mirada

- Código
- Infraestructura
- Datos
- Métricas
- Sesgos
- Monitoreo

> Sync con Industria

- Features & Data
- Model Development
- Infrastructure
- Monitoring

Paper: What's your ML Test Score? A rubric for ML production systems
https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45742.pdf

> Sync con Industria

- Features & Data
- Model Development
- Infrastructure
- Monitoring

1. Usual testing practices for features code
2. Feature Distribution Expectations
3. Relationship w target or pairwise
4. Worth paying cost per feature
5. Easy & Stable Feature deprecation
6. Privacy across Pipeline
7. Calendar time for create & add new feature to prod

Paper: What's your ML Test Score? A rubric for ML production systems
https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45742.pdf

> Sync con Industria

- Features & Data
- Model Development
- Infrastructure
- Monitoring

1. Code Review Model Specifications
2. Relationship between proxy & actual metrics
3. Explore Hyperparameters
4. Effect of Model Staleness
5. Compare with Simpler Baseline
6. Quality on relevant data Slices
7. Presence of implicit Bias

Paper: What's your ML Test Score? A rubric for ML production systems
https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45742.pdf

> Sync con Industria

- Features & Data
- Model Development
- Infrastructure
- Monitoring

Muchas prácticas… ¿no?

Paper: What's your ML Test Score? A rubric for ML production systems
https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45742.pdf

> Sync con Industria

- Features & Data
- Model Development
- Infrastructure
- Monitoring

1. Reproducibility of training
2. Unit test model specification code
3. Integration test full pipeline
4. QA before serving
5. Incremental training
6. Server vs Model sync & Canary process
7. Roll back to previous version

Paper: What's your ML Test Score? A rubric for ML production systems
https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45742.pdf

> Sync con Industria

- Features & Data
- Model Development
- Infrastructure
- Monitoring

1. Upstream instability in features (in train & serving)
2. Data invariants hold (t & s)
3. Features compute the same in t & s
4. State of model staleness
5. NaN appearing in data in t & s
6. Bare metal peaks or leaks
7. Quality regressions on served predictions

Paper: What's your ML Test Score? A rubric for ML production systems
https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45742.pdf

> Sync con Industria

- Features & Data
- Model Development
- Infrastructure
- Monitoring

Muchas prácticas, en serio muchas

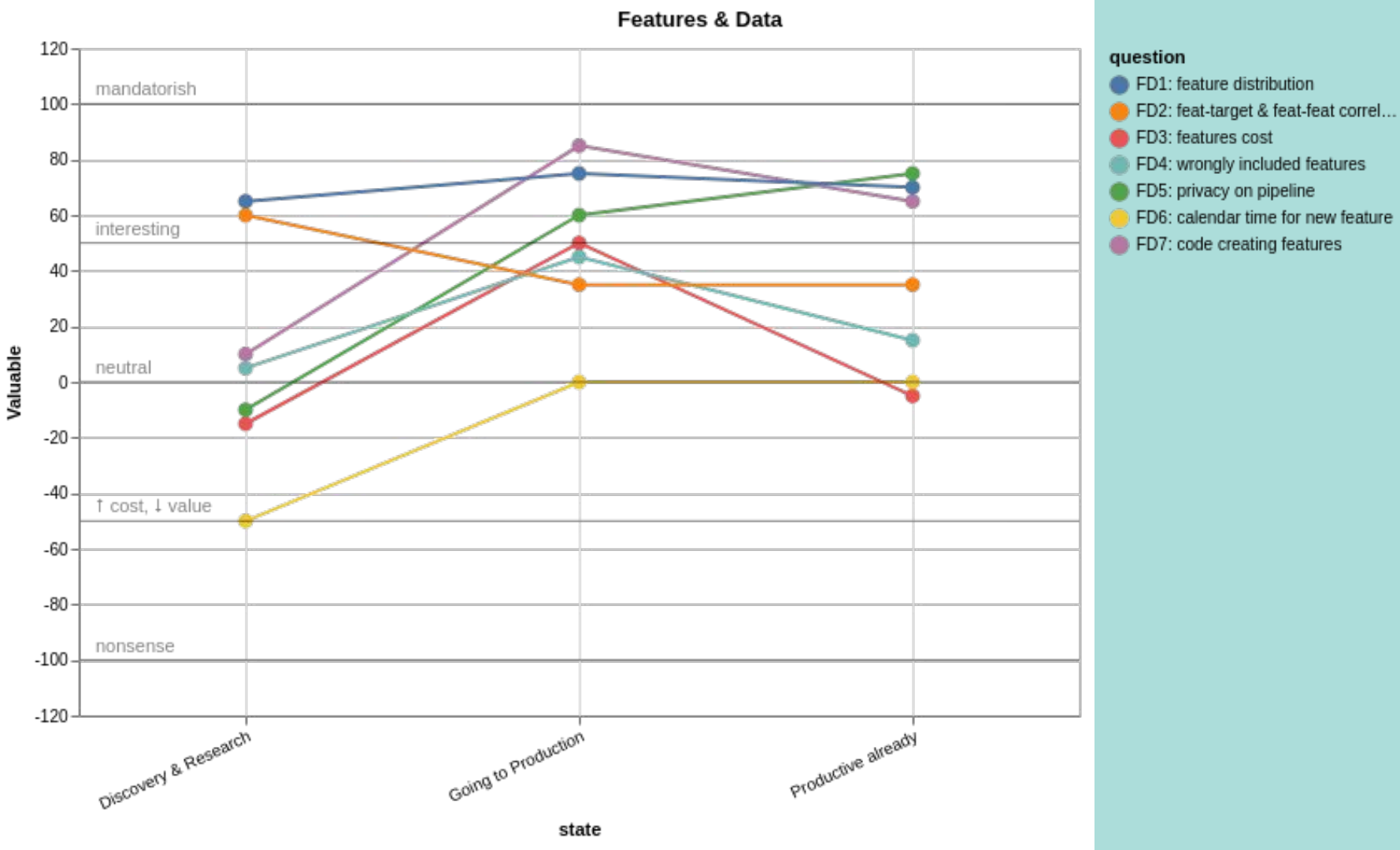
Paper: What's your ML Test Score? A rubric for ML production systems
https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45742.pdf

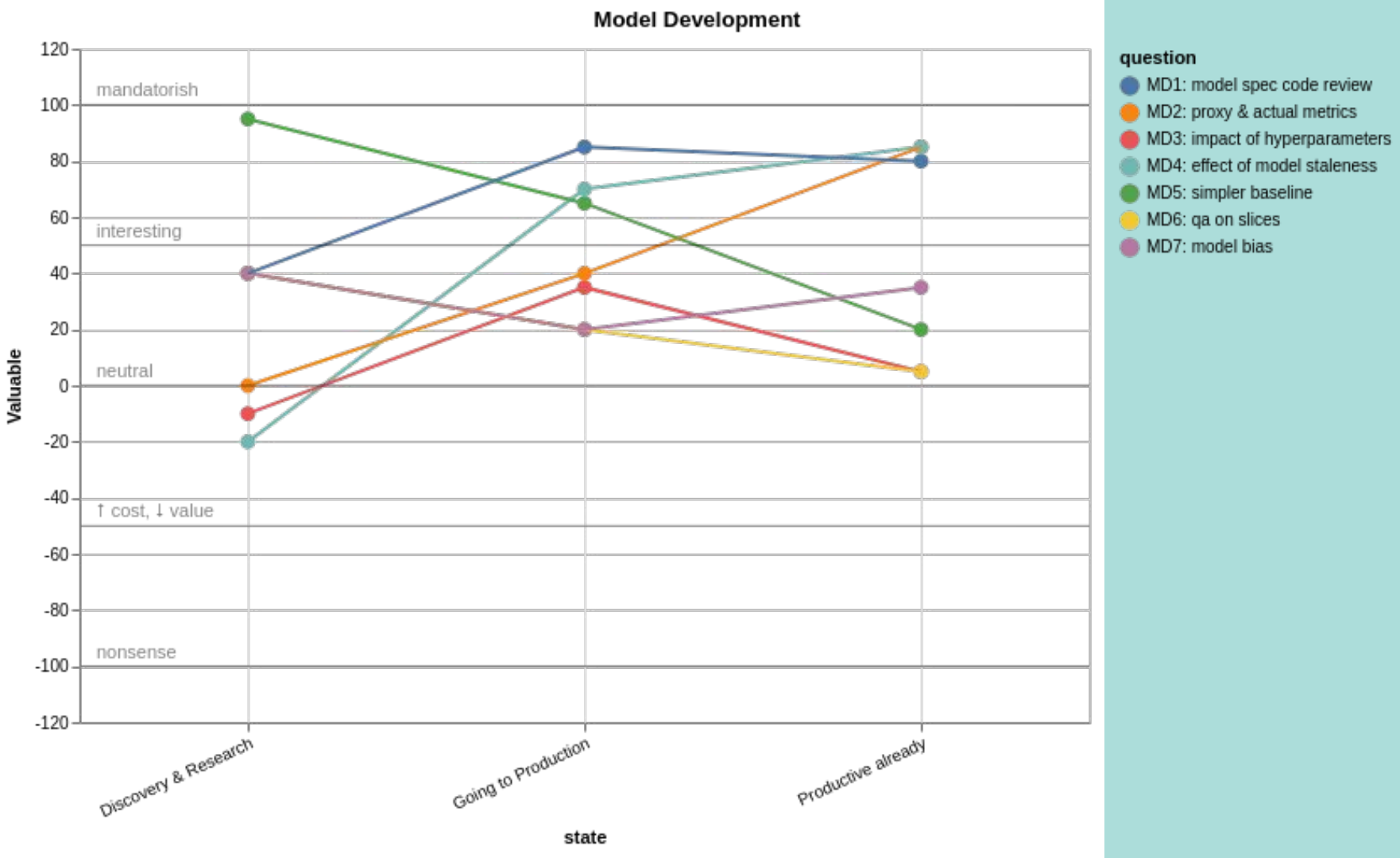
> Criterios (link)

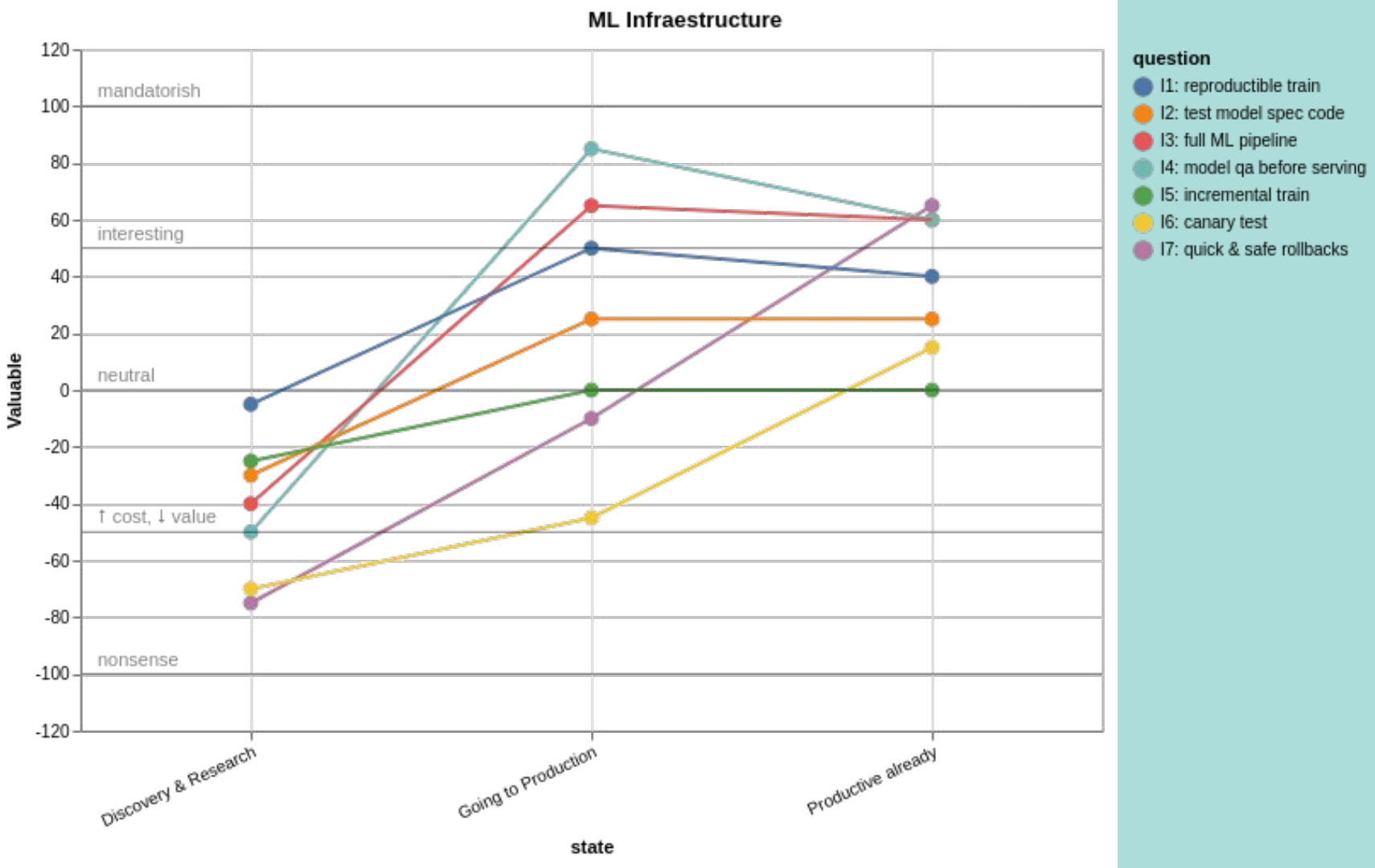
Cruzando dimensiones...

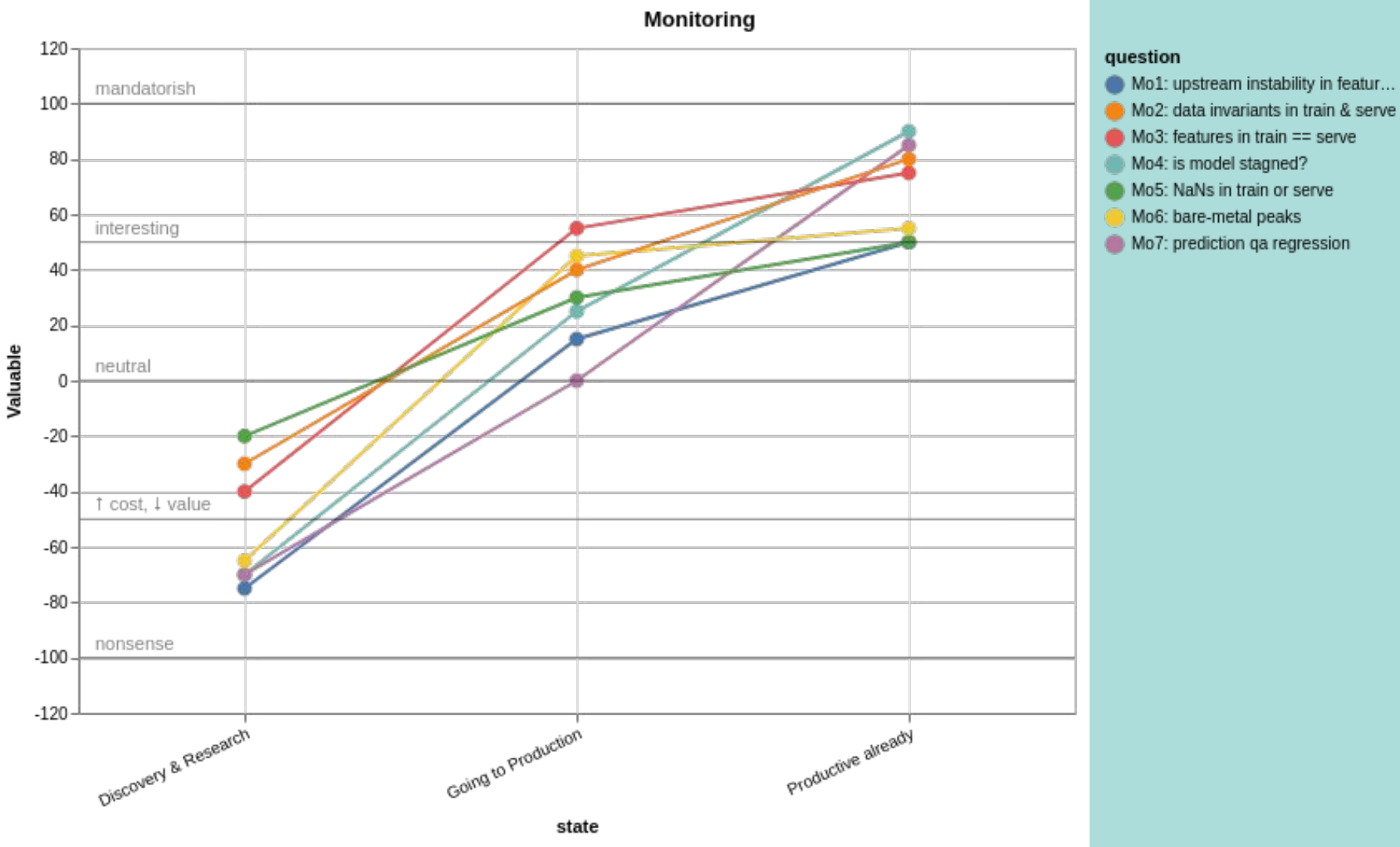
Test that the distribution of each feature match your expectations *

	Does not make sense	Poor value / Too exp...	Interesting / Optional	Mandatory
Research Only project...	○	○	○	○
Production in the road...	○	○	○	○
Already in production	○	○	○	○

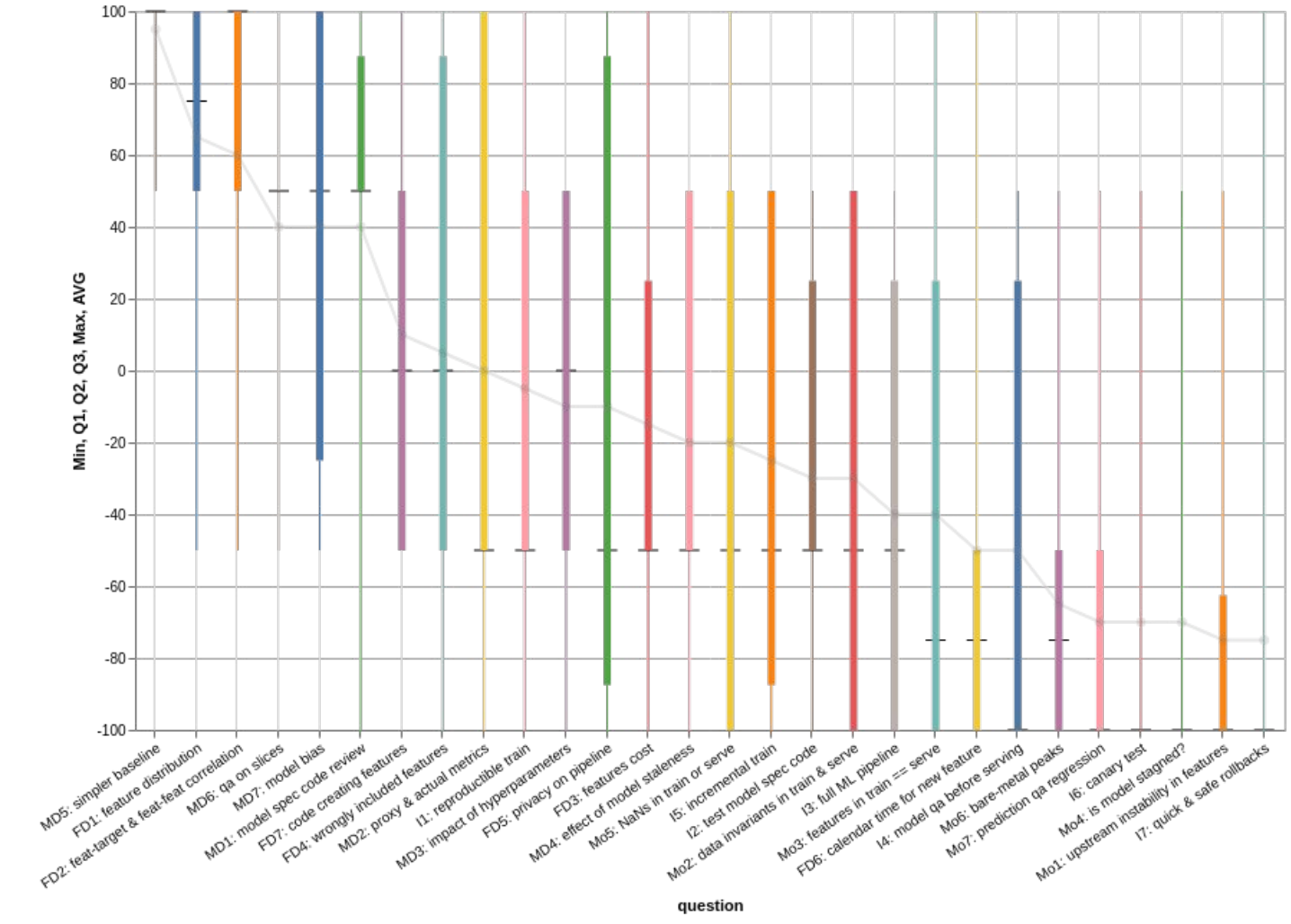

Features & Data
120
100
80
60
40
20
0
-20
-40
-60
-80
-100
-120
mandatorish
interesting
neutral
↑ cost, ↓ value
nonsense
Valuable
Discovery & Research
Going to Production
Productive already
state
question
FD1: feature distribution
FD2: feat-target & feat-feat correl…
FD3: features cost
FD4: wrongly included features
FD5: privacy on pipeline
FD6: calendar time for new feature
FD7: code creating features


Model Development
120
100
80
60
40
20
0
-20
-40
-60
-80
-100
-120
mandatorish
interesting
neutral
↑ cost, ↓ value
nonsense
Valuable
Discovery & Research
Going to Production
Productive already
state
question
MD1: model spec code review
MD2: proxy & actual metrics
MD3: impact of hyperparameters
MD4: effect of model staleness
MD5: simpler baseline
MD6: qa on slices
MD7: model bias


ML Infraestructure
Valuable
state
mandatorish
interesting
neutral
↑ cost, ↓ value
nonsense
120
100
80
60
40
20
0
-20
-40
-60
-80
-100
-120
Discovery & Research
Going to Production
Productive already
question
I1: reproductible train
I2: test model spec code
I3: full ML pipeline
I4: model qa before serving
I5: incremental train
I6: canary test
I7: quick & safe rollbacks

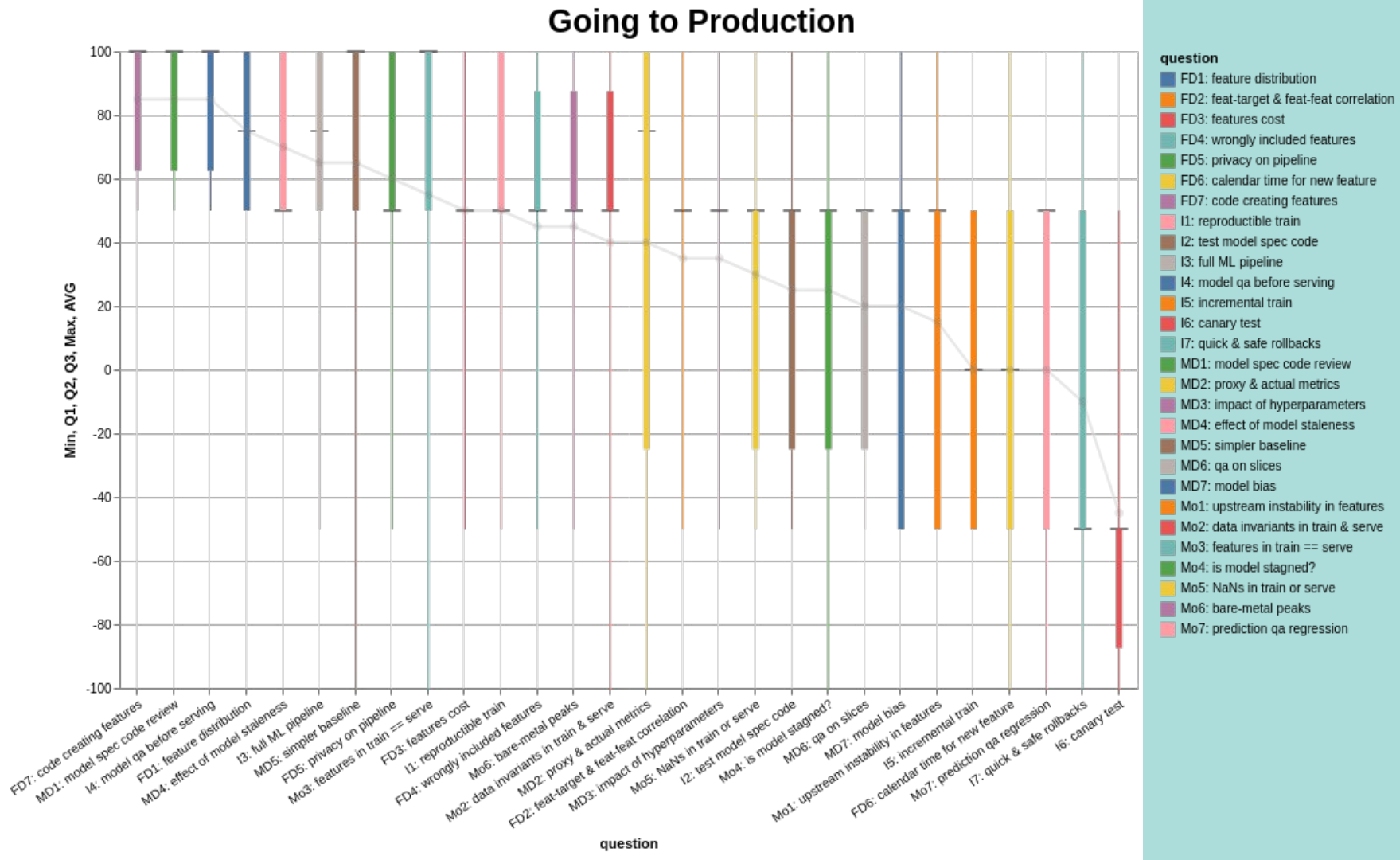

Monitoring
Valuable
state
mandatorish
interesting
neutral
↑ cost, ↓ value
nonsense
120
100
80
60
40
20
0
-20
-40
-60
-80
-100
-120
Discovery & Research
Going to Production
Productive already
question
Mo1: upstream instability in featur…
Mo2: data invariants in train & serve
Mo3: features in train == serve
Mo4: is model stagned?
Mo5: NaNs in train or serve
Mo6: bare-metal peaks
Mo7: prediction qa regression

Discovery & Research

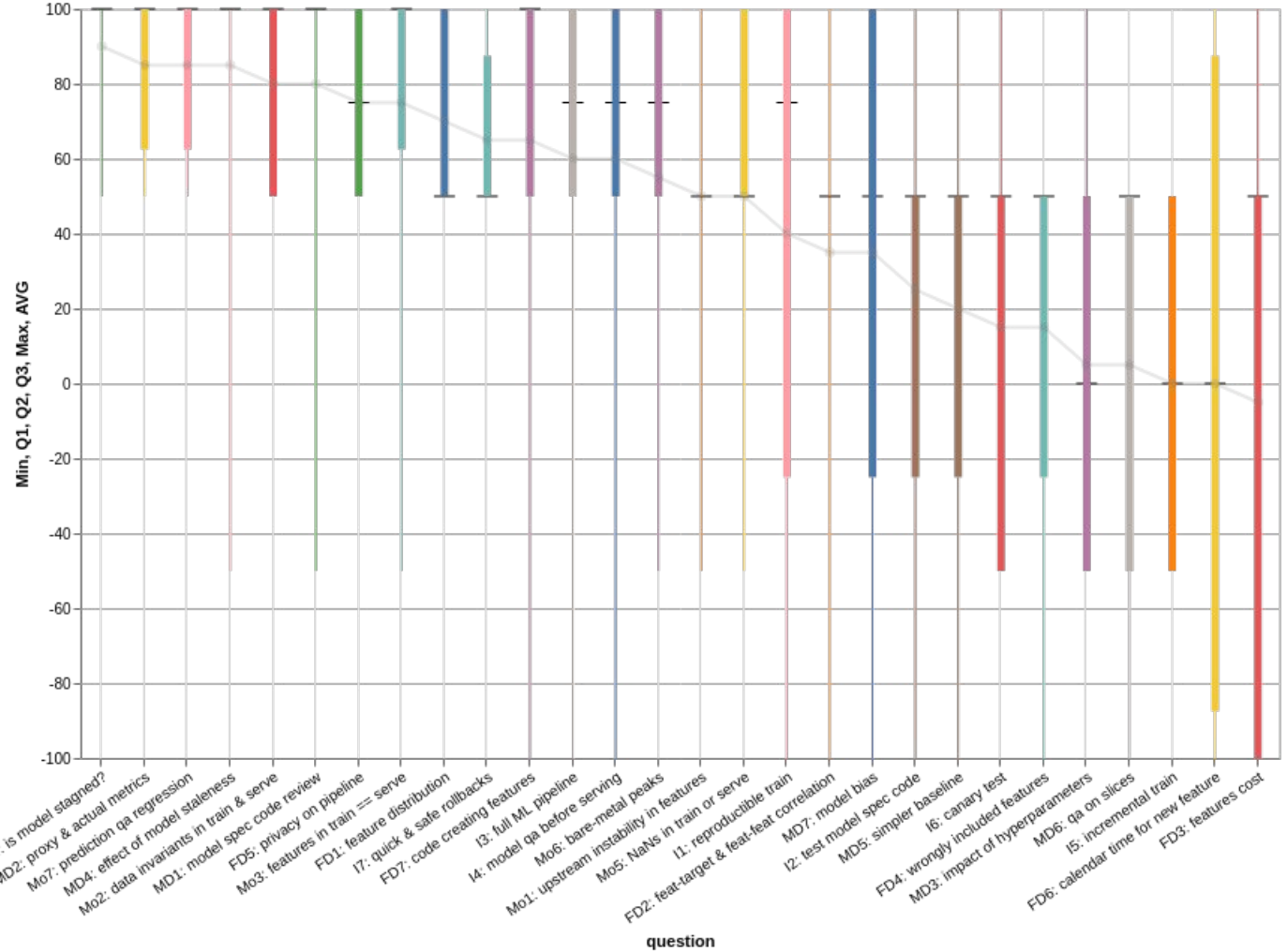


question

- FD1: feature distribution
- FD2: feat-target & feat-feat correlation
- FD3: features cost
- FD4: wrongly included features
- FD5: privacy on pipeline
- FD6: calendar time for new feature
- FD7: code creating features
- I1: reproductible train
- I2: test model spec code
- I3: full ML pipeline
- I4: model qa before serving
- I5: incremental train
- I6: canary test
- I7: quick & safe rollbacks
- MD1: model spec code review
- MD2: proxy & actual metrics
- MD3: impact of hyperparameters
- MD4: effect of model staleness
- MD5: simpler baseline
- MD6: qa on slices
- MD7: model bias
- Mo1: upstream instability in features
- Mo2: data invariants in train & serve
- Mo3: features in train == serve
- Mo4: is model stagned?
- Mo5: NaNs in train or serve
- Mo6: bare-metal peaks
- Mo7: prediction qa regression


Going to Production
Min, Q1, Q2, Q3, Max, AVG
100
80
60
40
20
0
-20
-40
-60
-80
-100
FD7: code creating features
MD1: model spec code review
I4: model qa before serving
FD1: feature distribution
MD4: effect of model staleness
I3: full ML pipeline
MD5: simpler baseline
FD5: privacy on pipeline
Mo3: features in train == serve
FD3: features cost
I1: reproductible train
FD4: wrongly included features
Mo6: bare-metal peaks
Mo2: data invariants in train & serve
MD2: proxy & actual metrics
FD2: feat-target & feat-feat correlation
MD3: impact of hyperparameters
Mo5: NaNs in train or serve
I2: test model spec code
Mo4: is model stagned?
MD6: qa on slices
MD7: model bias
Mo1: upstream instability in features
I5: incremental train
FD6: calendar time for new feature
Mo7: prediction qa regression
I7: quick & safe rollbacks
I6: canary test
question
question
FD1: feature distribution
FD2: feat-target & feat-feat correlation
FD3: features cost
FD4: wrongly included features
FD5: privacy on pipeline
FD6: calendar time for new feature
FD7: code creating features
I1: reproductible train
I2: test model spec code
I3: full ML pipeline
I4: model qa before serving
I5: incremental train
I6: canary test
I7: quick & safe rollbacks
MD1: model spec code review
MD2: proxy & actual metrics
MD3: impact of hyperparameters
MD4: effect of model staleness
MD5: simpler baseline
MD6: qa on slices
MD7: model bias
Mo1: upstream instability in features
Mo2: data invariants in train & serve
Mo3: features in train == serve
Mo4: is model stagned?
Mo5: NaNs in train or serve
Mo6: bare-metal peaks
Mo7: prediction qa regression

Productive already



question

- FD1: feature distribution
- FD2: feat-target & feat-feat correlation
- FD3: features cost
- FD4: wrongly included features
- FD5: privacy on pipeline
- FD6: calendar time for new feature
- FD7: code creating features
- I1: reproductible train
- I2: test model spec code
- I3: full ML pipeline
- I4: model qa before serving
- I5: incremental train
- I6: canary test
- I7: quick & safe rollbacks
- MD1: model spec code review
- MD2: proxy & actual metrics
- MD3: impact of hyperparameters
- MD4: effect of model staleness
- MD5: simpler baseline
- MD6: qa on slices
- MD7: model bias
- Mo1: upstream instability in features
- Mo2: data invariants in train & serve
- Mo3: features in train == serve
- Mo4: is model stagned?
- Mo5: NaNs in train or serve
- Mo6: bare-metal peaks
- Mo7: prediction qa regression

Gracias por la atención...

Preguntas?

Muchas Gracias