We’re hitting R a million times a day so we made a talk about it

rstudio::conf

January 29th, 2020

Heather Nolis – Machine Learning Engineer, T-Mobile – @heatherklus
Jacqueline Nolis – Principal Data Scientist, Nolis, LLC – @skyetetra
Don’t hurt R
OUR TEAM

AI @ T-Mobile

Goal  Use machine learning to improve the customer experience

Scope  customer care messaging
Our R models are hit over one million times a day (really!)

Hey! I’m travelling to Europe and need my phone to work.

I see your phone is locked. I’ll submit an unlock request now.

Our PRODUCT: EXPERT ASSIST

CUSTOMER

T-MOBILE EXPERT

Conversation topic
unlock

Account information
Status: Locked
Phone: Apple iPhone X - Space Gray - 64GB
IMEI: 352041097588970

Relevant C2 articles
- T-Mobile Device Unlock app
- Mobile Device Unlock requests
- Mobile device unlock troubleshooting
- Mobile Device Unlock escalations
- Mobile Device Unlock eligibility: Postpaid
How we do it

Keras
Create deep learning models with {keras}

plumber
Create REST APIs to host models

docker
Use Docker containers to serve APIs

kubernetes
Deploy Docker containers to Kubernetes

rstudio::conf 2019 talk: bit.ly/push-straight-to-prod
THREE TALES FROM BEYOND THE RELEASE:
LESSONS LEARNED FROM PRODUCTION
THE TALE OF TOO MANY TESTS
How can you ensure the model will work before deploying?

Our technical lead asked us a fair question
Run a unit testing script to test API endpoints as part of our deployment pipeline
How do you test the model itself?

If “I want to unlock my phone” is not at least topic UNLOCK with 0.95 probability then return ERROR

But:
- 0.95 (or any threshold) too arbitrary
- What if test fails but the model is overall more accurate?
LESSONS LEARNED

▪ Unit testing = bowling lane buffers
▪ Data scientists make the strikes
▪ Never just turn them off 😞
THE CURIOUSLY SLOW RESPONSE
Sometimes when we hit your topic model API the response is too slow. Can you find out why?
Saw the issue in our production logs

- You want a record of everything
- R packages like \{log4r\} integrate with production systems
LOAD TESTING
(seeing how your code behaves under high volumes)
Load testing with `{loadtest}`

NOW OPEN SOURCE!

Package available at: `github.com/tmobile/loadtest`
Two problems on two occasions

**GARBAGE COLLECTION**

Sometimes R needs to stop and clean up a bit

**CONNECTION TIME OUT**

If `httr` tries to connect to a service that’s down it’ll wait a bit

Both times root cause was found with `{profvis}`
R is single threaded

Single threaded = code can only do one thing at a time
You can work around single threading

- User parallelizing tools like \{future\} and RestRserve
- Making your code run better
- Deploying copies of your API and load balance
LESSONS LEARNED

▪ Load test your code
▪ Look at more than the just average time
▪ Consider how single threading affects code
THE STAKEHOLDERS WHO CRIED MODEL
STAKEHOLDERS

The agent tools are running slowly, is it the model?
The model classifies this strangely, is it broken?
There are no topics showing, is it the model?

When things go wrong people first look to the things they don’t understand
We had many Shiny demos but not enough

Great for selling work

But some may want to go deeper

Our customer detector Shiny demo: bit.ly/smalldatabigvalue
We created an explanatory tool

When Things Go Wrong People Blame the Model But It’s Never The Model

“I wish every product owner had this!” – Joel Werdell, Principal Product
Resolve issues by keeping track of things

**CONFLUENCE FOR COMMON ISSUES**

Reference Doc for handling SMPD AI msg-agent-publisher:
No C2 articles found --> splunk Alert

Go into Live Engage and check if Expert Assist is working:
Step 1: Go to Splunk query while checking the logs copy the conversation ID and open live engage and go into all connections

**DEFINITION OF DONE FOR REPEATED TASKS**

1. lintr run / all probs fixed on model-related code
2. Model can be run independently (single R file)
3. Sign-off from engineering on model input / output / purpose
4. Code review
5. Algorithm review
6. Standardized headers in R file that describe input, output, data, algorithm, description
7. Code in Bitbucket
8. All acceptance criteria are met
9. Model validation documented
10. Model validation review meeting held
LESSONS LEARNED

- Problems are unlikely to be the model
- Give people a tool to understand your model
- Keep track of issues
If we could go back in time would we use R again?

(Absolutely)
Putting R in production isn’t a stretch

If you are sitting in this conference, you can put an R model into production.

Production is for everyone! Irene Steves & Yogez Hertz: bit.ly/riskibops
Hardest part isn’t the language
it’s the small community

- When our code breaks it’s not because of the language
- Boatloads of blog posts on Python in production
- You, audience member, can help grow this space!
We are here to help!

- Blog posts
- Docker containers
- Tooling
- Examples
- And more!

http://putrinprod.com

We would love help to grow this!
Thank you!

http://putrinprod.com

Heather Nolis - @heatherklus  Jacqueline Nolis - @skyetetra

Machine learning BoF 5:30-7 Franciscan C