From Prototyping to Deployment at Scale with R and sparklyr

Kevin Kuo

June 2018

Menu today

- The deployment problem
- ML pipelines
- Model deployment and demo

The deployment problem

Deployment

Or, putting ML models *"into production"*.

Deployment

Or, putting ML models *"into production"*.

Basically, make it so that someone else can use your model (i.e. make some predictions with it).

Deployment - latency dimension

Batch

- Event-based/time-based
- E.g. nightly portfolio risk calculations, and
- Email campaigns
- It's OK to take a while

"Real-time"

- On demand
- E.g. instant loan approvals, and
- Fraud detection on credit card swipes
- Gotta be (relatively) fast, seconds to less than a second

Deployment - why is it hard?

Challenge 1/n: Putting ML models into production involves different expertise

Deployment - why is it hard?

Challenge 1/n: Putting ML models into production involves different expertise

...so it involves more people

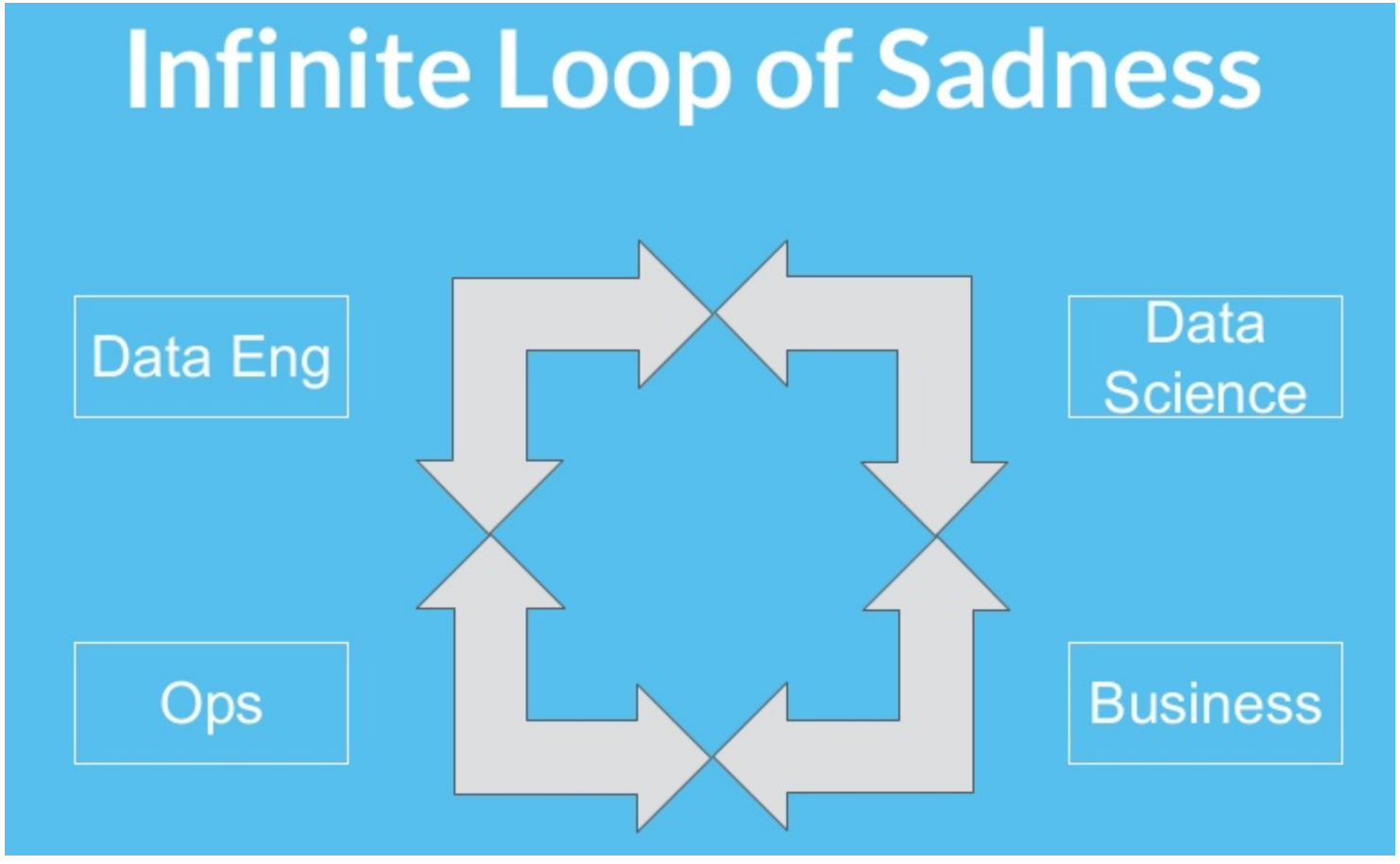
Deployment - why is it hard?

Challenge 1/n: Putting ML models into production involves different expertise

...so it involves more people

...mo ppl mo problems

Deployment - mo ppl mo problems



Credit: https://youtu.be/-K9SjrWpeys @josh_wills

Deployment - why is it hard?

Challenge 2/n: Rapidly changing landscape in deployment options

Deployment - why is it hard?

Challenge 2/n: Rapidly changing landscape in deployment options

What do?

- Spark ML persistence?
- dbml-local?
- PMML?
- PFA/Aardpfark?
- MLeap?
- ONNX?
- Roll our own thing?
- Re-implement the model in C++, because performance?
- Throw it into a container and do orchestration cuz it's cool?

Deployment - why is it hard?

Challenge 3/n: Too many ML frameworks and no standardization

Deployment - why is it hard?

Challenge 3/n: Too many ML frameworks and no standardization

We're focusing on Spark in this session, but we'll acknowledge other technologies we need to deal with

Deployment - diversity of ML frameworks

Spark ML, xgboost, random CRAN packages, scikit-learn, H2O, ...

Deployment - one of many scenarios

Data scientist: Hey this random forest loan decision model is ready to go!

Engineer: OK, we need to recode it in C#/Java, see you in 6 months!

Deployment - one of many scenarios

Data scientist: Hey this random forest loan decision model is ready to go!

Engineer: OK, we need to recode it in C#/Java, see you in 6 months!

Data scientist: Oh no that's too long, what about I give you this GLM with just a few parameters?

Engineer: 2 months.

Deployment - one of many scenarios

Data scientist: Hey this random forest loan decision model is ready to go!

Engineer: OK, we need to recode it in C#/Java, see you in 6 months!

Data scientist: Oh no that's too long, what about I give you this GLM with just a few parameters?

Engineer: 2 months.

Data scientist: 🤔

Deployment - challenges for the R user

On *average*, R users tend to...

- Be math/stats types
- Have little CS/software engineering training

So it's slightly tougher for them to collaborate with the folks doing model implementation.

Deployment - challenges for the R user

On *average*, R users tend to...

- Be math/stats types
- Have little CS/software engineering training

So it's slightly tougher for them to collaborate with the folks doing model implementation.

However,

- Data scientists (regardless of background) are becoming more comfortable moving up and down the stack
- There has been active development of technology to faciliate the data science-engineering handoff

Deployment - technology

Technology won't solve your people/process/culture issues, but it can *make collaboration easier*!

Deployment - technology

Technology won't solve your people/process/culture issues, but it can *make collaboration easier*!

Next up: we'll provide a quick review of Spark ML pipelines, and offer a couple ways of "deploying" them using the **sparklyr** ecosystem.

Spark ML pipelines

ML pipelines

Pipelines are basically…

> A structure in which you can throw in data transformers and ML models.

ML pipelines

Pipelines are basically...

> A structure in which you can throw in data transformers and ML models.

Keep in mind that when you deploy a model, you also need to deploy the feature engineering steps in order to feed the right inputs to the model! (E.g. converting a numeric `age` variable into an age range bucket that the model requires.)

ML pipelines

Pipelines are basically...

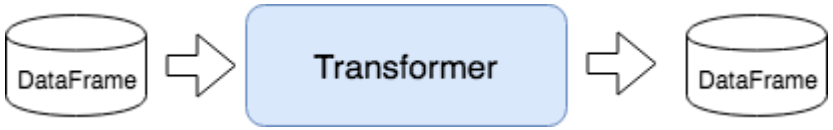
> A structure in which you can throw in data transformers and ML models.

Keep in mind that when you deploy a model, you also need to deploy the feature engineering steps in order to feed the right inputs to the model! (E.g. converting a numeric `age` variable into an age range bucket that the model requires.)

Now let's go through a (very quick) overview of pipeline concepts.

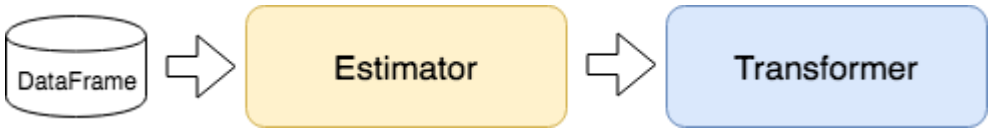
ML pipelines

- A `Transformer` takes a data frame, via `ml_transform()`, and returns a transformed data frame.



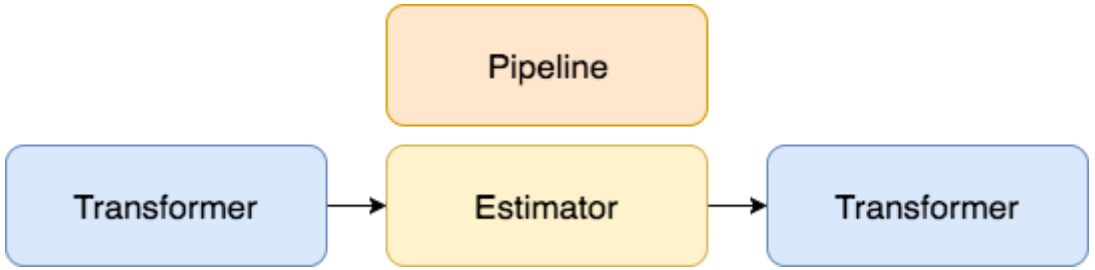
ML pipelines

- A `Transformer` takes a data frame, via `ml_transform()`, and returns a transformed data frame.
- An `Estimator` take a data frame, via `ml_fit()`, and returns a `Transformer`.



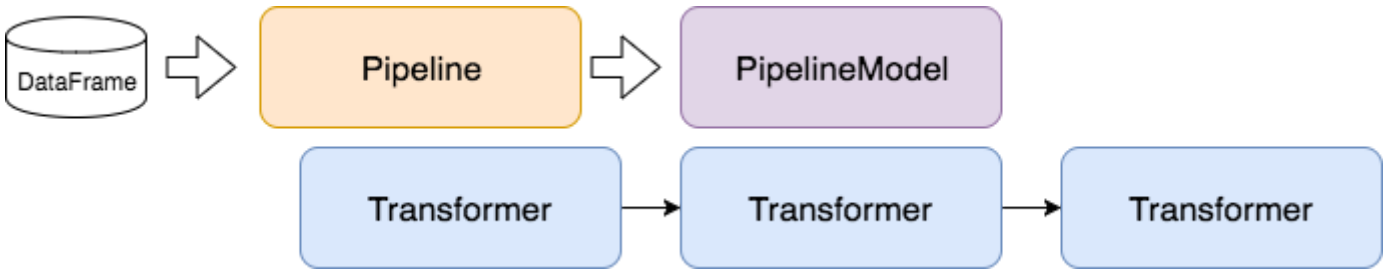
ML pipelines

- A `Transformer` takes a data frame, via `ml_transform()`, and returns a transformed data frame.
- An `Estimator` take a data frame, via `ml_fit()`, and returns a `Transformer`.
- A `Pipeline` consists of a sequence of stages—`PipelineStages`—that act on some data in order.
  - A `PipelineStage` can be either a `Transformer` or an `Estimator`.

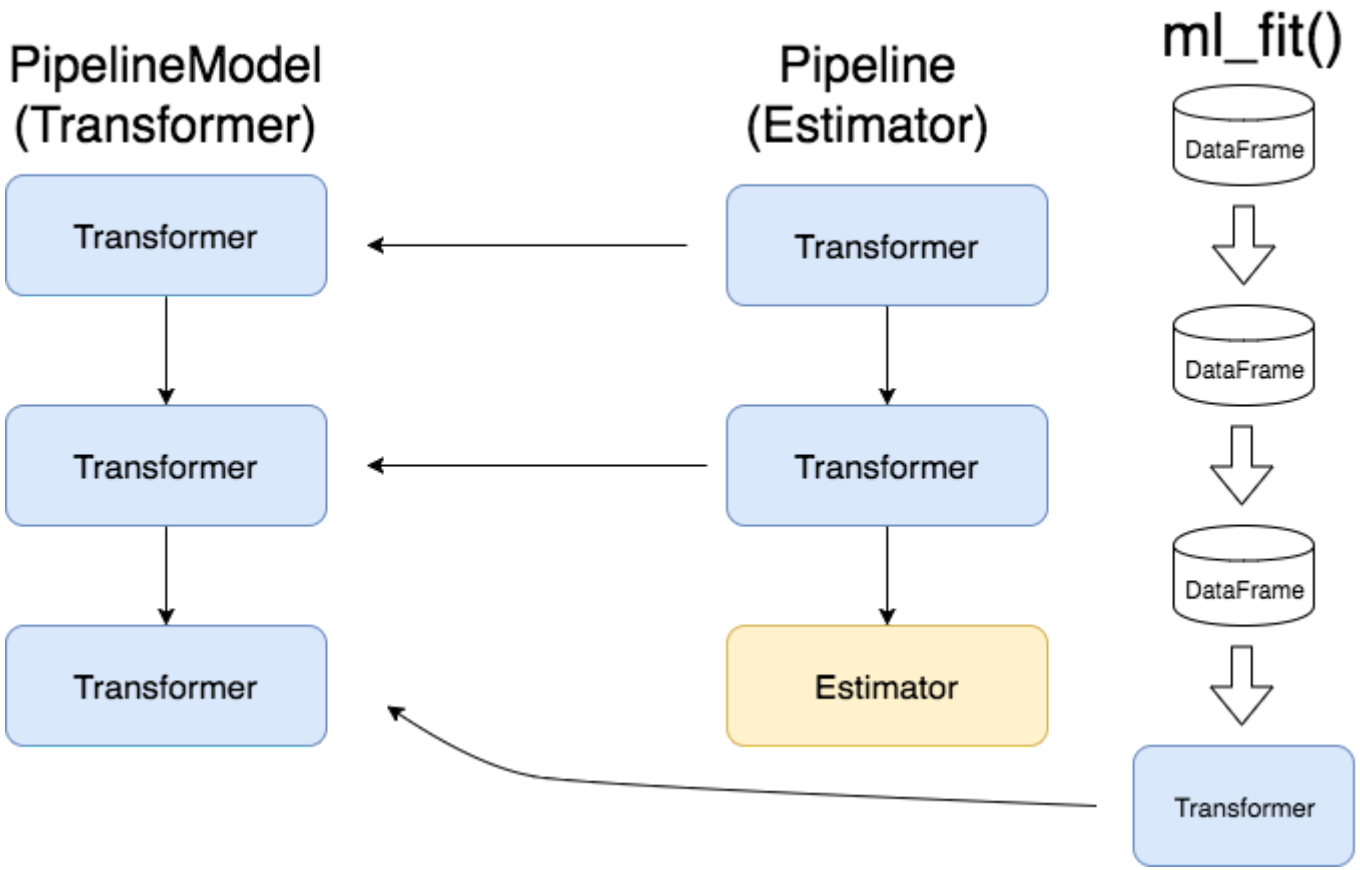


ML pipelines

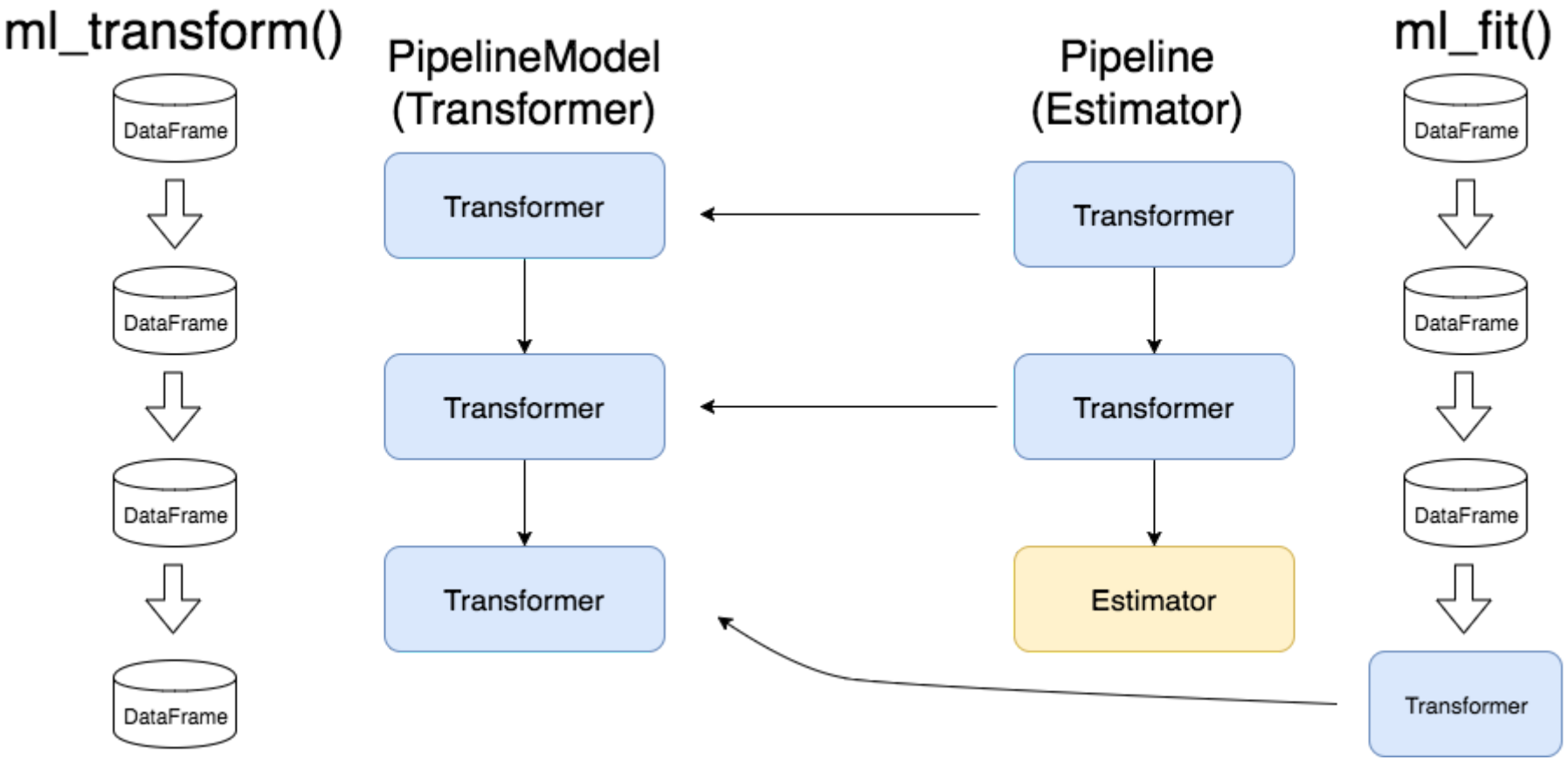
- A `Pipeline` consists of a sequence of stages—`PipelineStages`—that act on some data in order.
  - A `PipelineStage` can be either a `Transformer` or an `Estimator`.
- A `Pipeline` is always an `Estimator`, and its fitted form is called `PipelineModel` which is a `Transformer`.



ML pipelines



ML pipelines



Serving the model

Now, the trick is to persist this `PipelineModel` so we can use it to serve predictions later on.

Serving the model

Now, the trick is to persist this `PipelineModel` so we can use it to serve predictions later on.

We'll demo a couple ways today

- Native Spark ML persistence support
- MLeap (via the **mleap** R package)

Demo

Model deployment paths

Spark ML Persistence

- Appropriate for batch jobs, scoring lots of records at once
- Requires Spark session

MLeap

- Better for real-time prediction of a small number of records
- Doesn't require Spark session, portable to apps/devices that support JVM

Towards a better deployment story

Data scientist: Hey this random forest loan decision model is ready to go! Here is the `.zip` bundle, and here is the documentation you need to use it.

Engineer: Awesome! We won't need to write a bazillion if-else statements to recreate the model!

Towards a better deployment story

Data scientist: Hey this random forest loan decision model is ready to go! Here is the `.zip` bundle, and here is the documentation you need to use it.

Engineer: Awesome! We won't need to write a bazillion if-else statements to recreate the model!

When the model needs updating...

Towards a better deployment story

Data scientist: Hey this random forest loan decision model is ready to go! Here is the `.zip` bundle, and here is the documentation you need to use it.

Engineer: Awesome! We won't need to write a bazillion if-else statements to recreate the model!

When the model needs updating...

Data scientist: We decided to use a GBM instead for better accuracy, here's the updated bundle.

Engineer: Fantabulous! All we need to do is update the model directory!

Wrap up

Slides and code will be available at https://kevinykuo.com.

Inspirations/other talks to check out

- "Productionizing Spark ML pipelines with the portable format for analytics" https://youtu.be/h-B0VCkoRkE @MLnick
- "How to Productionize Your Machine Learning Models Using Apache Spark MLlib 2.x" https://youtu.be/r740xbIpb54 Richard Garris
- "MLeap and Combust ML" https://youtu.be/MGZDF6E41r4 Hollin Wilkins and Mikhail Semeniuk