

# Deploying Machine Learning in R with Amazon SageMaker

@MichaelFrasco

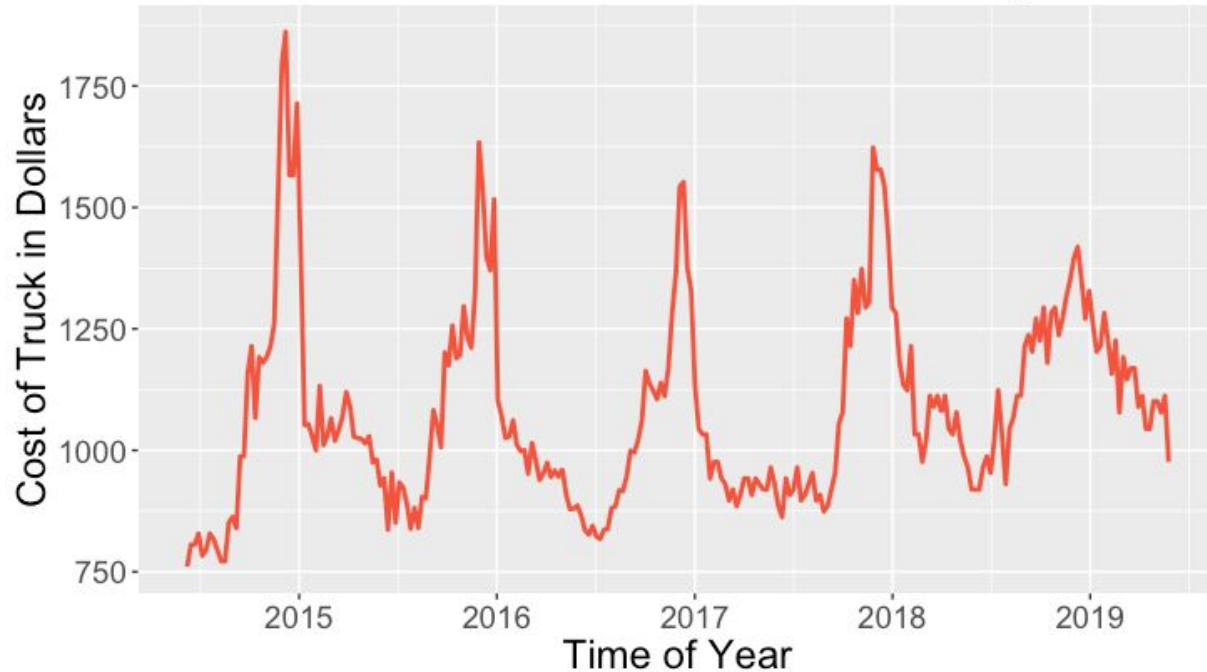
# Where I work

- Founded in 2015
- A two-sided marketplace for shippers and carriers
- Shippers
  - Companies that need to move freight across the country
- Carriers
  - Truck drivers that can move the freight

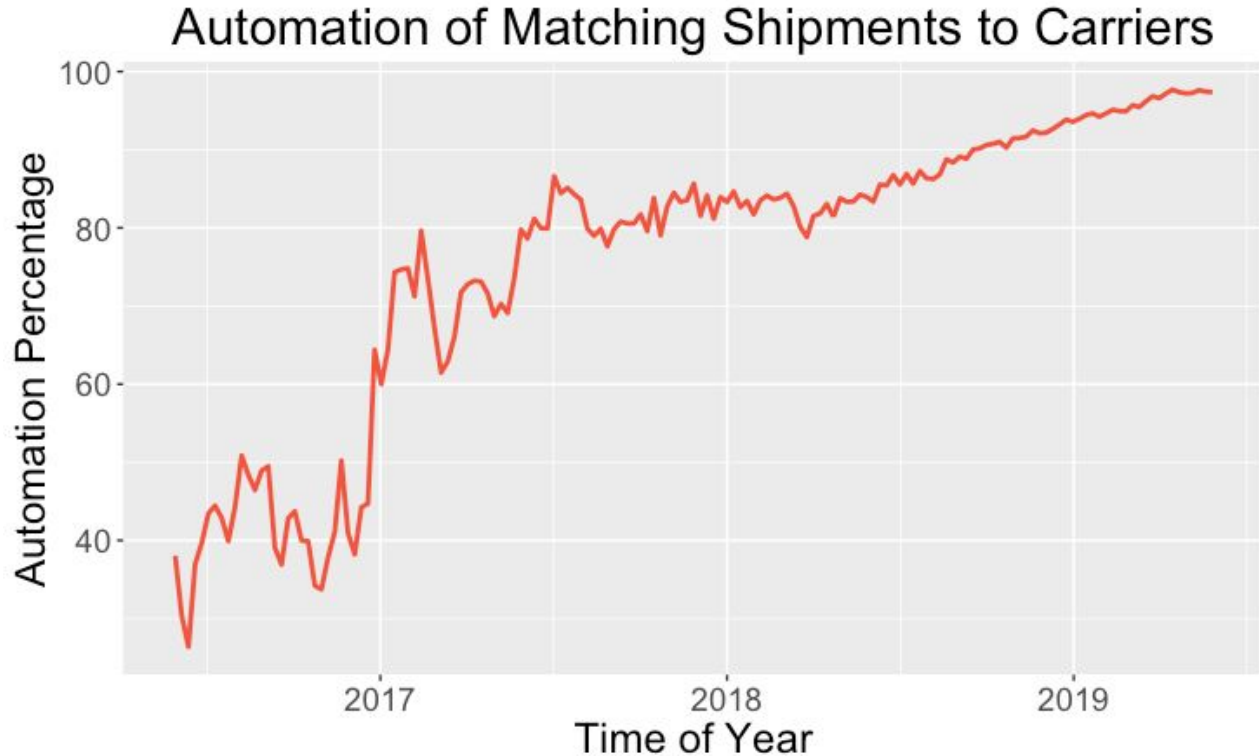


# Machine Learning is Critical to Our Success

Truck Costs From Seattle to Los Angeles



# Machine Learning is Critical to Our Success

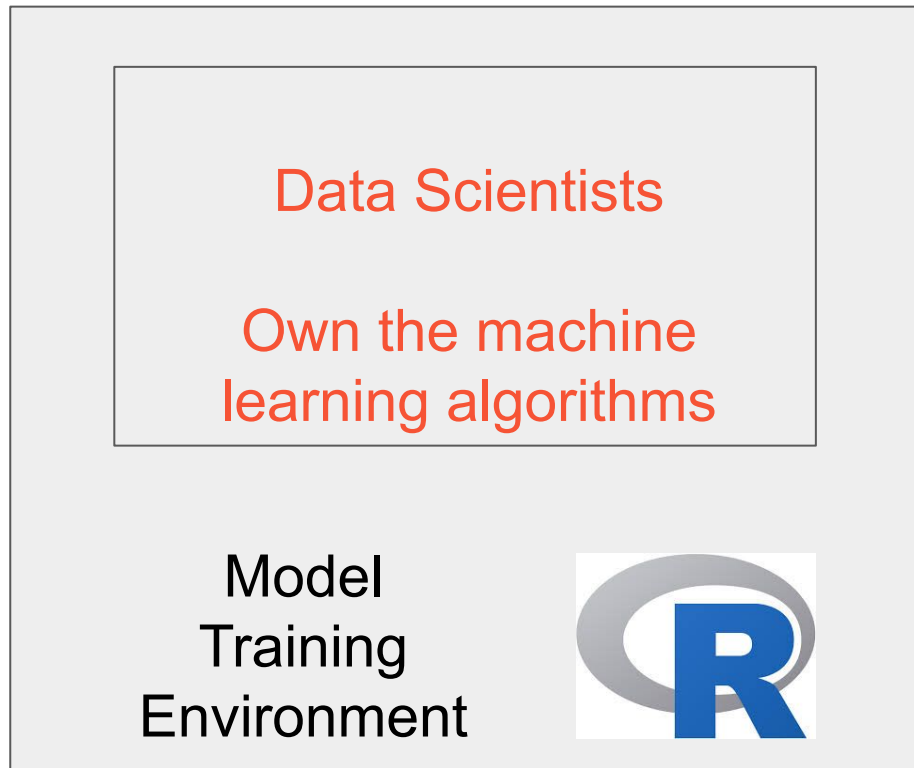


# Machine Learning Landscape

Data Scientists

Own the machine  
learning algorithms

# Machine Learning Landscape



# Machine Learning Landscape

Software Engineers

Own the uptime of our  
business systems

# Machine Learning Landscape





# Machine Learning Landscape

Data Scientists

Own the machine learning algorithms

Model  
Training  
Environment



Software Engineers

Own the uptime of our business systems

Convoy  
Production  
Environment



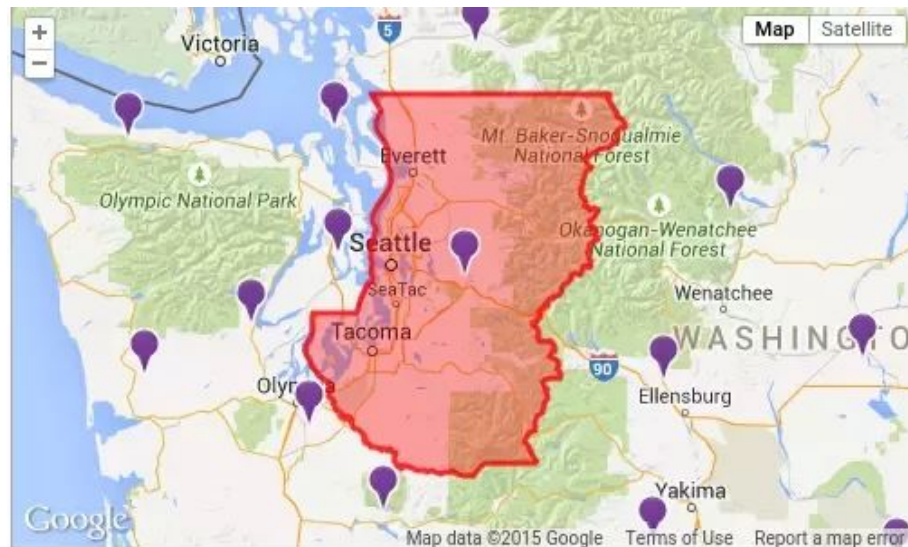
# Training-Serving Skew



**Training-serving skew is a difference between a model's behavior during training and during serving.**

# Examples of training-serving skew

- Distance to the nearest metro
- Different definitions of the standard deviation
  - MLE vs unbiased



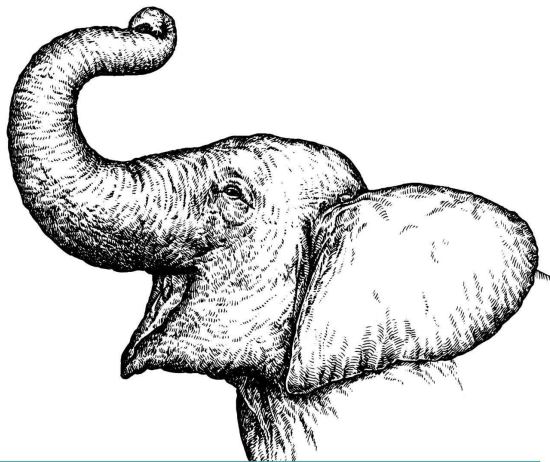
# Why does training-serving skew happen



- Iterating quickly and constantly
- Communication issues between data scientist and software engineer
- Different programming languages

# How do we prevent training-serving skew

*The answer to every programming question ever conceived*



It Depends

*The Definitive Guide*

Model Training  
Environment



Upload trained  
model

Model Serving  
Environment



Model Training  
Environment



Upload trained  
model and  
code

Model Serving  
Environment



Request prediction



Respond with result



Convoy  
Production  
Environment





# Communication Between Languages

- REST APIs
  - Established technology for software engineers
- plumber
  - Widely used R package
  - Transforms your functions into an API



# plumber requires a one line change

Tells the internet  
how to find this function

```
## @post /predict
#' This function inputs a new observation
#' and outputs a prediction from my model
function(request) {
  # put any code you want in here
  result <- my_ml_model(request)
  result
}
```



# But it is actually much more complicated

- Security!
- Reliability!
- Monitoring!
- Scaling!
- Unknowns!



**Should we build it ourselves?**

**Should we buy it from someone else?**

# Decided to buy from AWS SageMaker



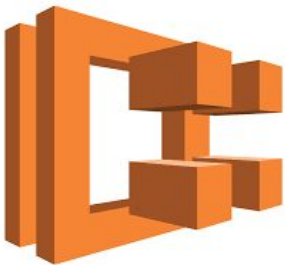


docker





Upload to  
→



**Amazon ECS**

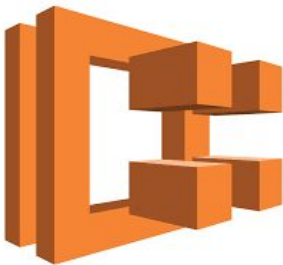


Upload to  
→

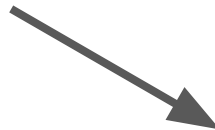




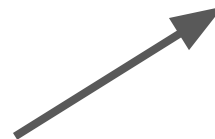
Upload to



Amazon ECS



Upload to



Amazon SageMaker



# We have seen terrific benefits

- Increased ownership
  - Less reliance on software engineers

# We have seen terrific benefits

- Increased ownership
  - Less reliance on software engineers
- Faster iterations between model improvements

# We have seen terrific benefits

- Increased ownership
  - Less reliance on software engineers
- Faster iterations between model improvements
- Reliable and trustworthy