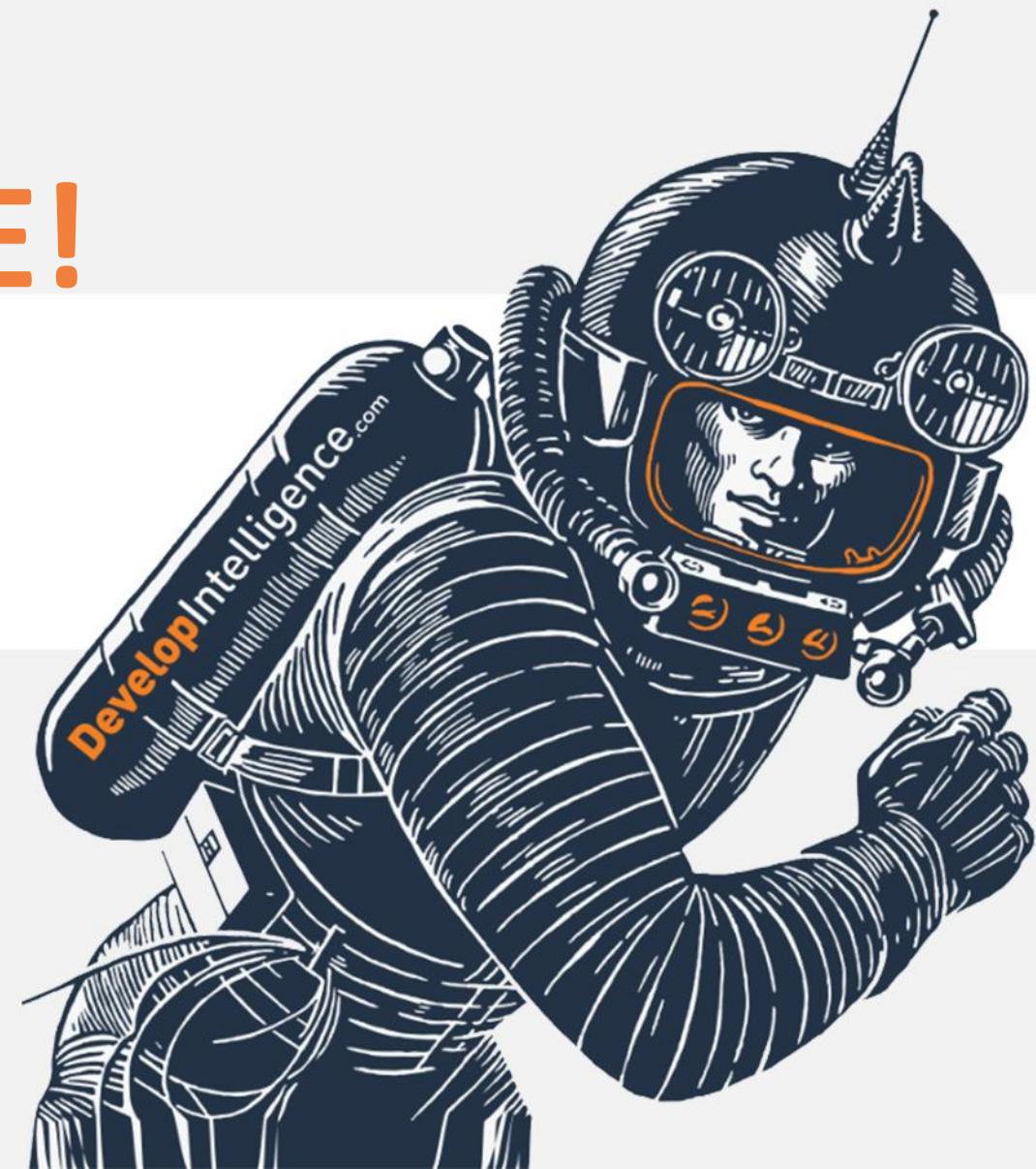


# Introduction to Sagemaker and ML Pipelines

## WELCOME!



Axel Sirota  
MCT and AI Consultant





# Join Us in Making Learning Technology Easier



## Our mission...

Over 16 years ago, we embarked on a journey to improve the world by making learning technology easy and accessible to everyone.

...impacts everyone daily.

And it's working. Today, we're known for delivering customized tech learning programs that drive innovation and transform organizations.

In fact, when you talk on the phone, watch a movie, connect with friends on social media, drive a car, fly on a plane, shop online, and order a latte with your mobile app, you are experiencing the impact of our solutions.



Over The Past Few Decades, We've Provided

Over  
**62,300,000**  
expert-led learning hours

In 2019 Alone, We Provided

Training to over  
**13,500** engineers

Programs in  
**30** countries

Over **120**  
active trainers, with  
an average of over  
two decades of  
experience each.

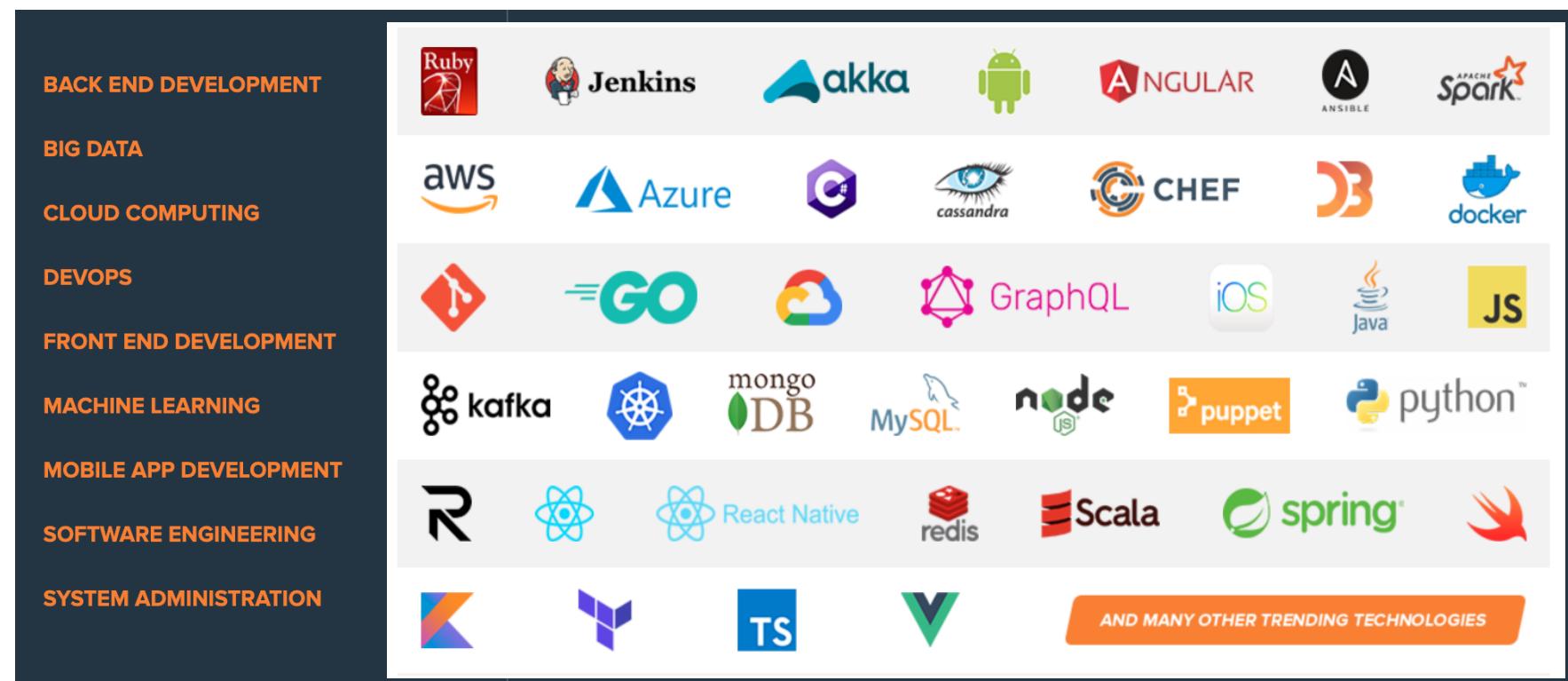


# Upskilling and Reskilling Offerings



Intimately customized learning experiences just for your teams.

	Workshop	2-3 day upskilling experiences
	Fast Track	5-day reskilling experiences
	Learning Spike	1-day technology overviews
	Target Topics	90-minute instructor-led micro-learnings
	Hack-a-thon	Learn and build an MVP in 2-3 days





# World Class Practitioners



250 best selling books authored



9+ years of training experience



Over 62 million practitioner led training hours



EXPERT PRACTITIONERS

SEASONED CONSULTANTS

ENGAGING INSTRUCTORS

150 speaking engagements at industry conferences



Over 17 years of industry experience per instructor

125 certifications in leading technologies



95% instructor satisfaction





# Note About Virtual Trainings



What we want



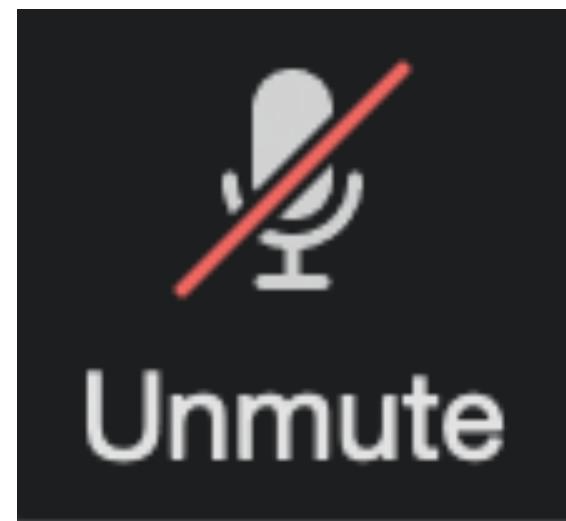
...what we've got



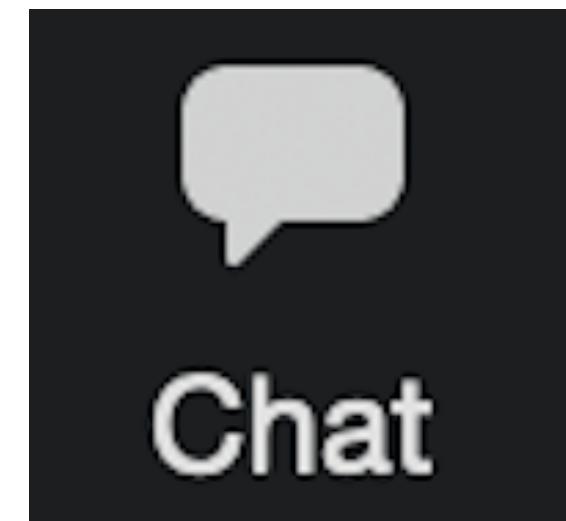
# Virtual Training Expectations for You



Arrive on time / return on time



Mute unless speaking



Use chat or ask  
questions verbally



# Virtual Training Expectations for Me



I pledge to:

- Make this as interesting and interactive as possible
- Ask questions in order to stimulate discussion
- Use whatever resources I have at hand to explain the material
- Try my best to manage verbal responses so that everyone who wants to speak can do so
- Use an on-screen timer for breaks so you know when to be back



# Prerequisites



- Some ML experience is helpful
- Python3 (some)



# What this course is about



This course **is about:**

- Learning how to use Sagemaker build, train and deploy models
- Learning how to apply MLOps practices to Sagemaker with Sagemaker Pipelines
- Learning how to scale models with Sagemaker

This course **is not about:**

- ML theory and how to tune jobs



# Objectives

At the end of this course you will be able to:

- Understand how SageMaker boots your productivity
- Devise a plan for an ML project from start to finish
- Develop pipelines and training jobs within SageMaker
- Deploy Realtime and Batch endpoints for consumption
- Scale and monitor model depending on production needs



# Structure of the Course / Course Takeaways



- We will be using a Github repository for labs throughout the training
- The labs will use AWS. We will assign users to each of you on the first demo
- Download the code and instructions for labs here: <https://github.com/axel-sirota/sagemaker-course>
- The slides will be uploaded to the repo



# Agenda

## Introduction

- Introductions
- Expectations
- Why SageMaker?

## Deploying

- Training and deploying
- Exposing a REST api

## Workshop

## SageMaker Fundamentals

- Setting up
- Data Analysis
- Building builtin and BYO models

## MLOps

- What is MLOps?
- Model drift, versioning and Projects
- Model monitor



# Agenda

## Introduction

- Introductions
- Expectations
- Why SageMaker?

## Deploying

- Training and deploying
- Exposing a REST api

## Workshop

## SageMaker Fundamentals

- Setting up
- Data Analysis
- Building builtin and BYO models

## MLOps

- What is MLOps?
- Model drift, versioning and Projects
- Model monitor

# Introduction



Who are we?





# Introductions



- One by one introduce yourself:
  - A. Name
  - B. Role
  - C. Previous experience with ML or SageMaker
  - D. What do you want to learn from this class?
  - E. Do you want to share a fun fact of yourself?



# Who am I?



- Microsoft Certified Trainer
- Author, Instructor, and Editor at [Pluralsight](#),  
[Develop Intelligence](#), and [O'Reilly Media](#)
- AI and Cloud Consultant



QR to my Pluralsight  
courses



QR to my O'Reilly  
trainings

# Introduction

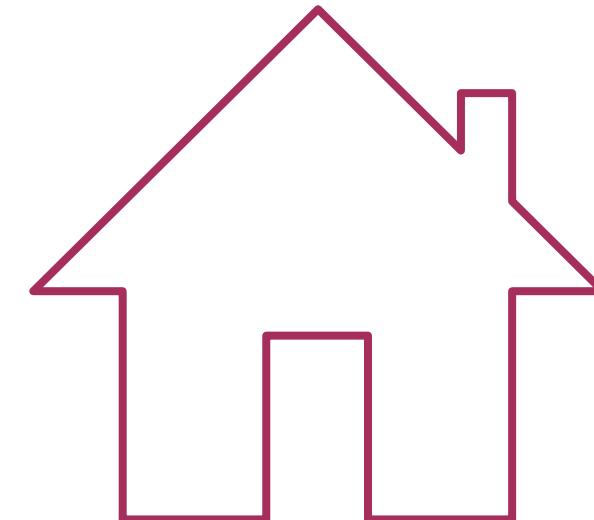


Why SageMaker?





# Case Study: Price Prediction Model



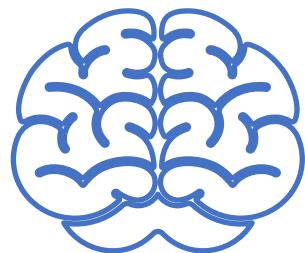
**2 bedroom, 2 bath, hardwood floors**



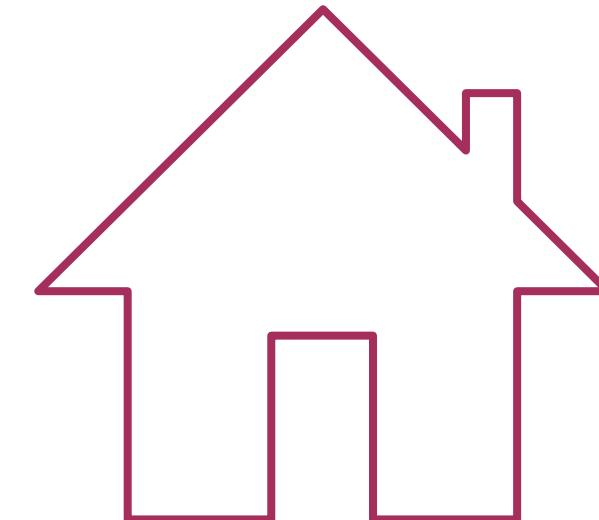
Price



# Case Study: Price Prediction Model



Model



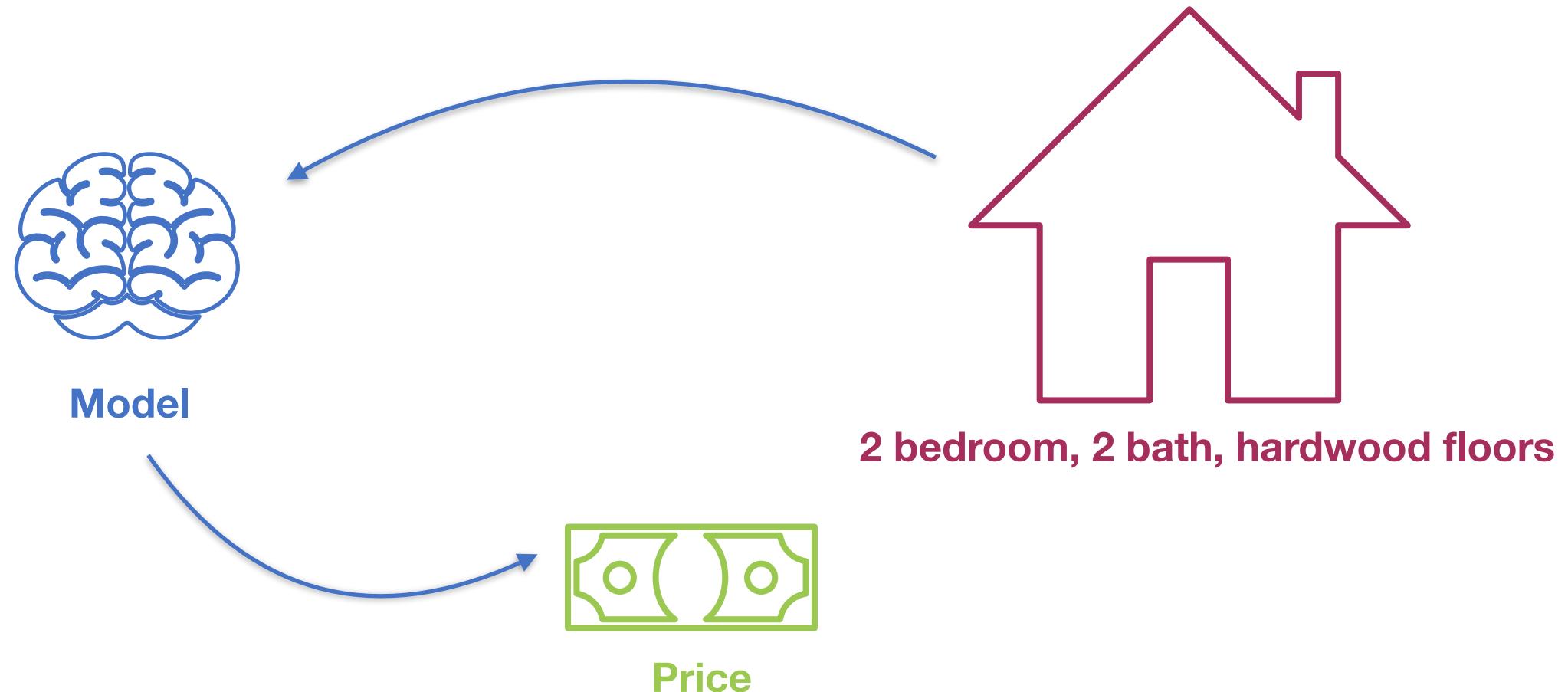
**2 bedroom, 2 bath, hardwood floors**



Price

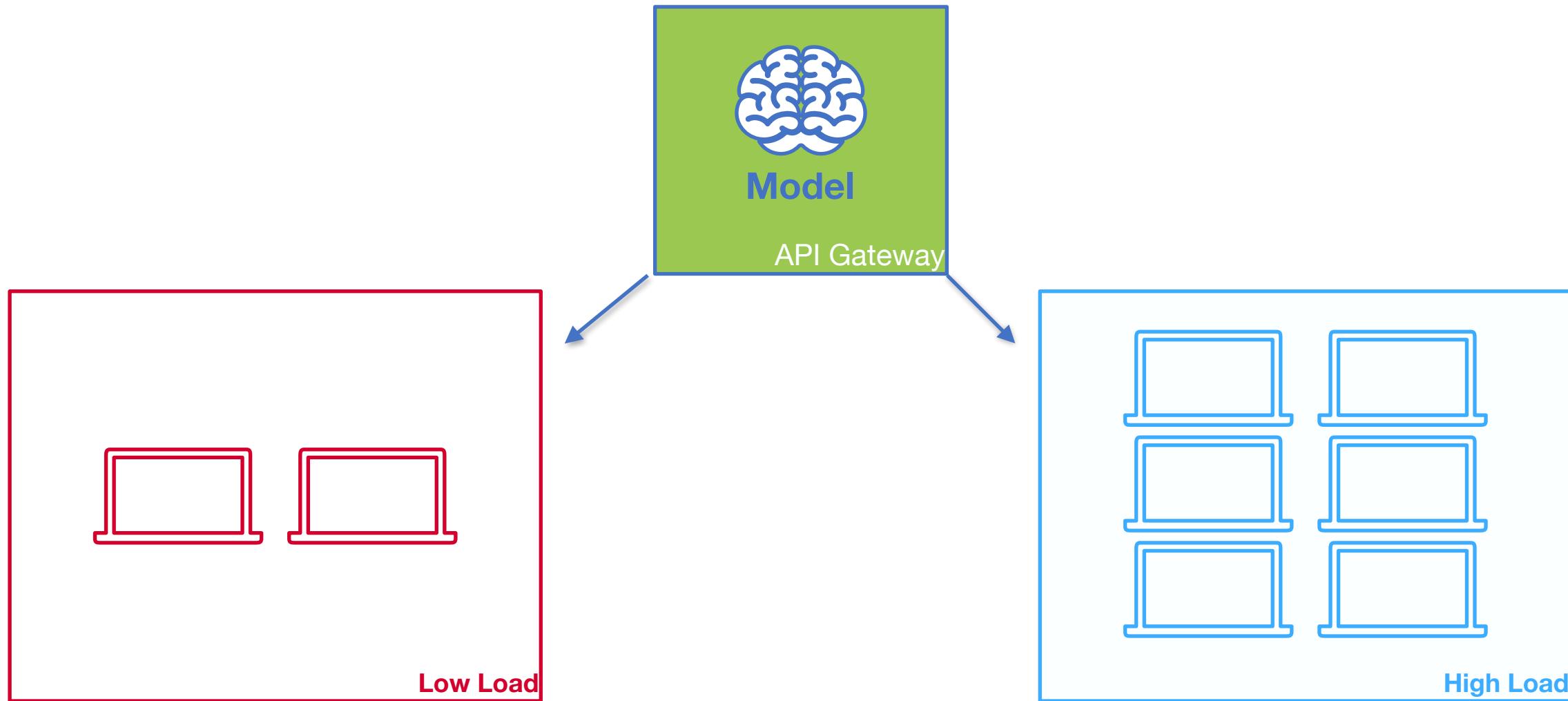


# Case Study: Price Prediction Model



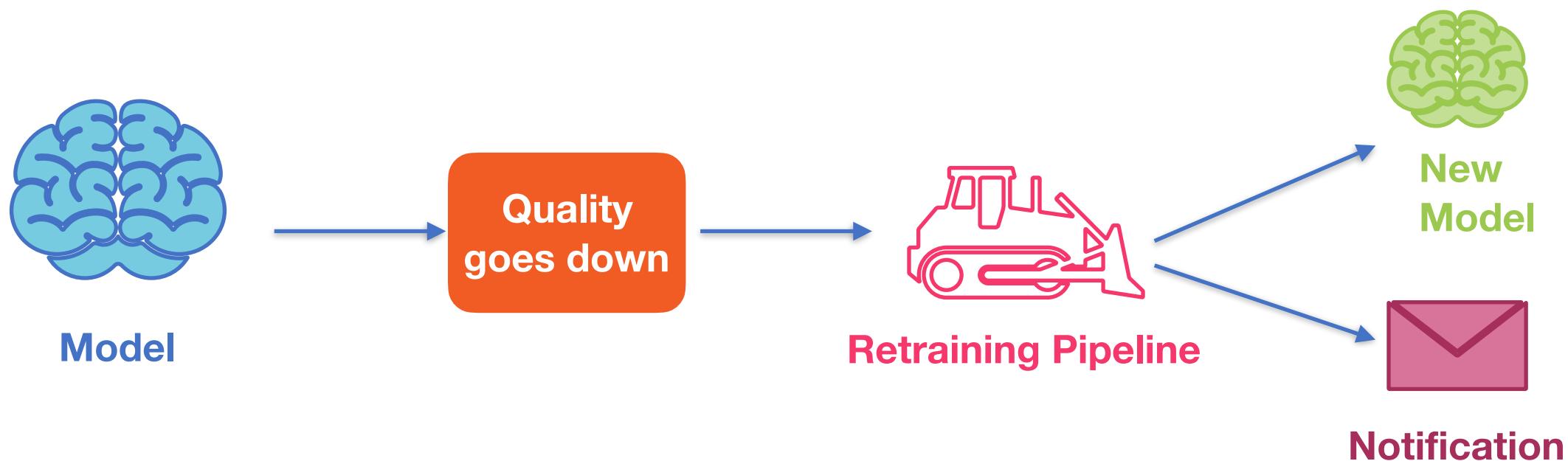


# Case Study: Price Prediction Model



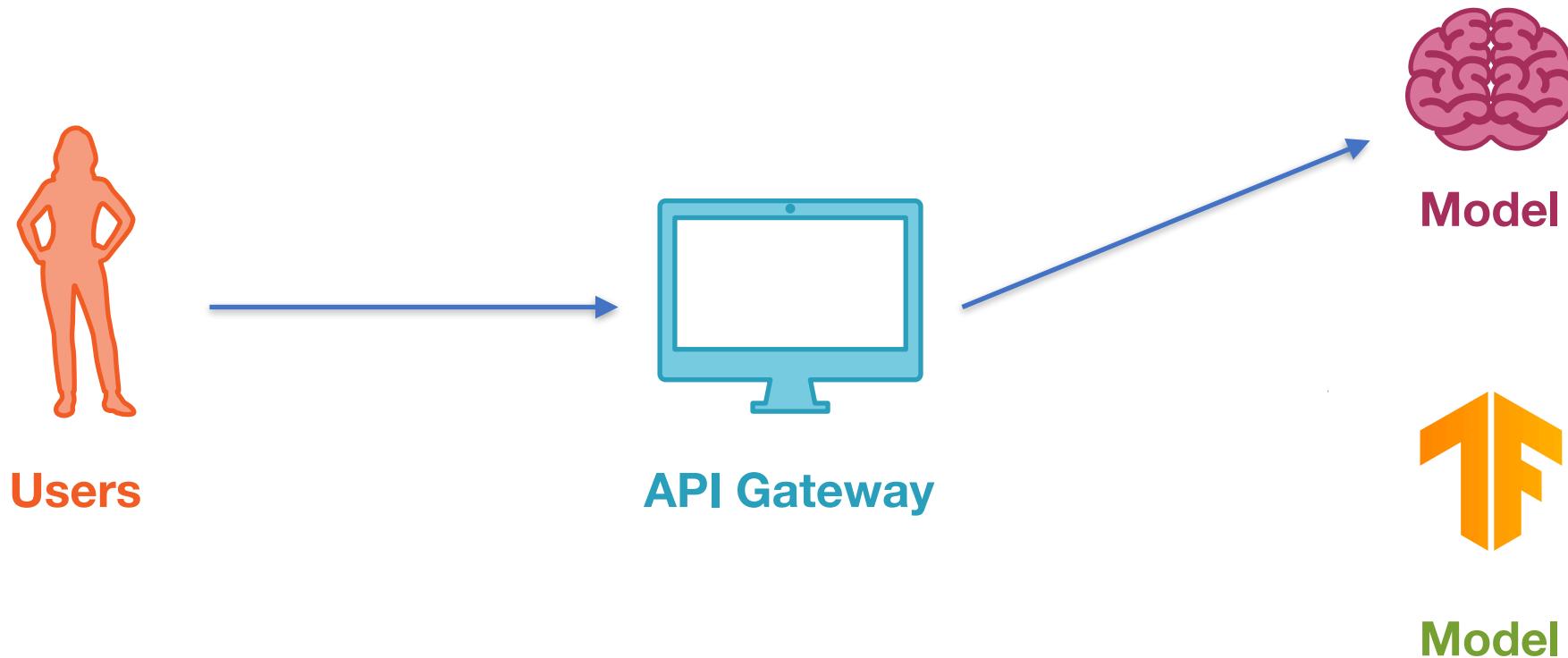


# Case Study: Price Prediction Model



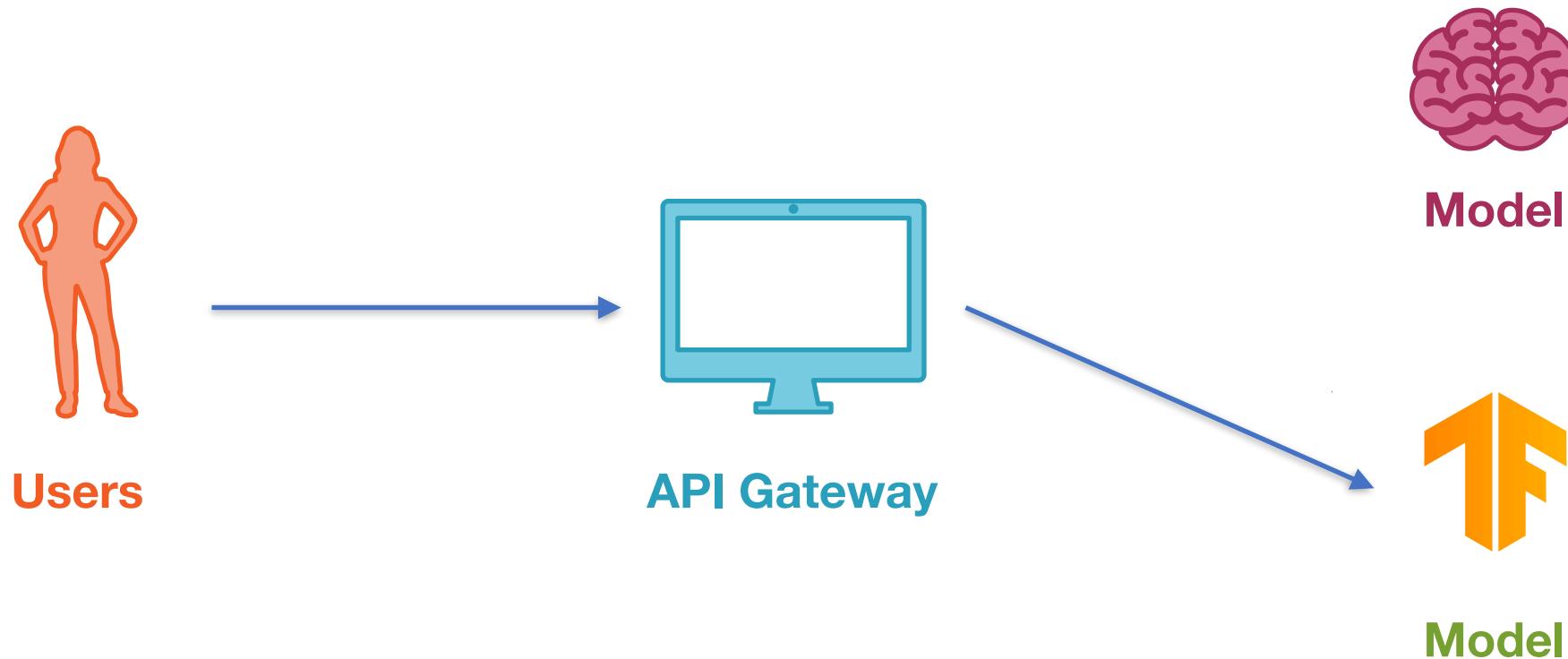


# Case Study: Price Prediction Model





# Case Study: Price Prediction Model





# Case Study: Price Prediction Model



## What would we need to do?

### Training and Deploying

Setting up the AMI/ Install stuff

Standardising the training framework and how to  
dispatch jobs

Configuring CloudWatch

Creating the CloudFormation template

### Retraining and Scalability

Setting up the ASG

Configuring the API Gateway and Route 53  
Creating a reporting and monitoring system

Creating the orchestrator system for retraining

All of these bring slowness, not happy customers and errors



# Case Study: Price Prediction Model



**AWS SageMaker come to solve  
this!**

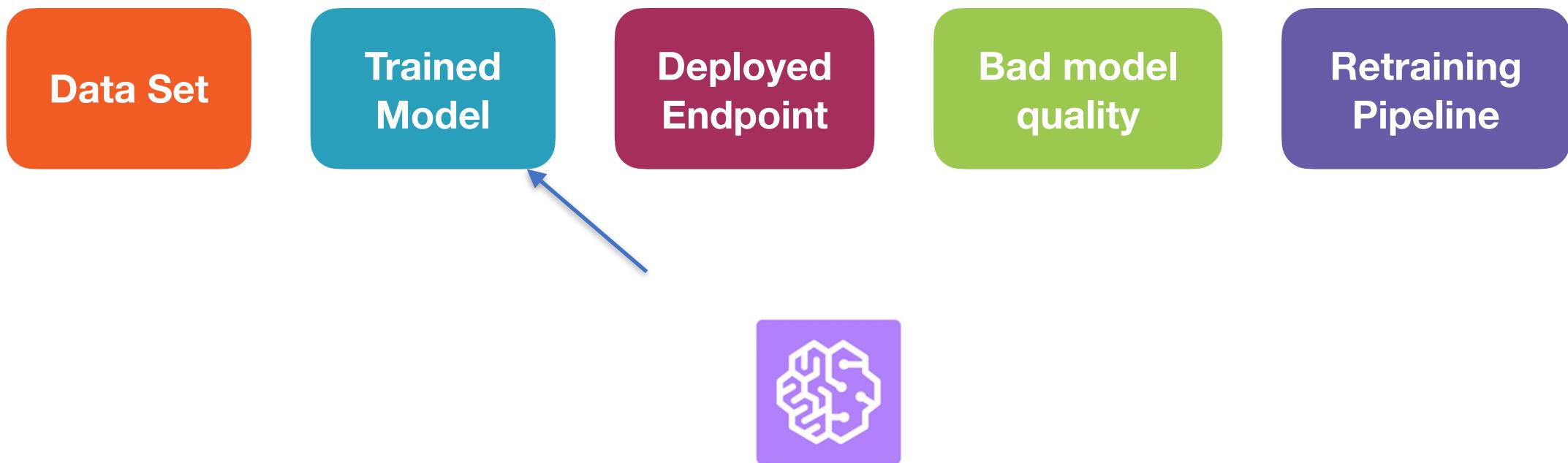


# Case Study: Price Prediction Model





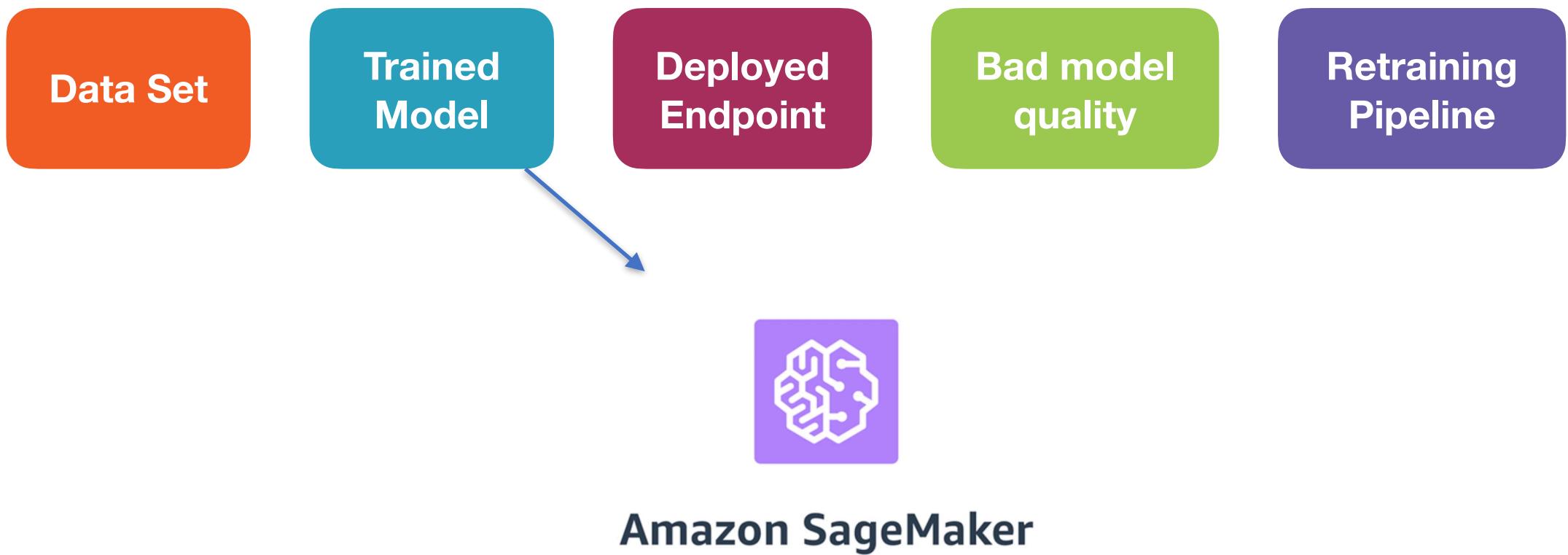
# Case Study: Price Prediction Model



Amazon SageMaker

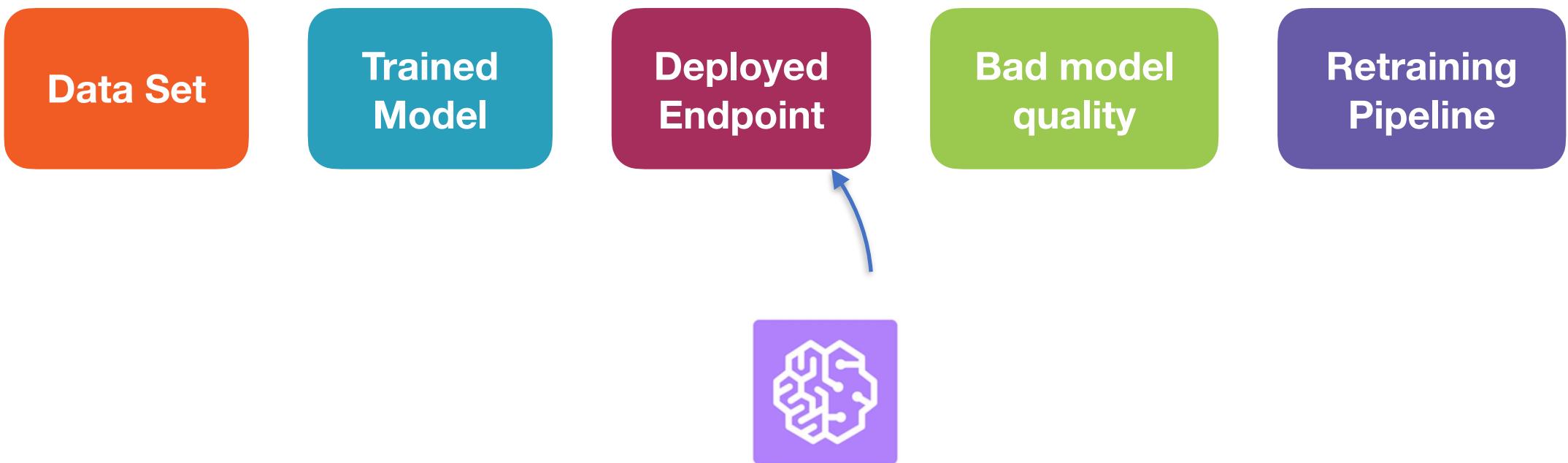


# Case Study: Price Prediction Model





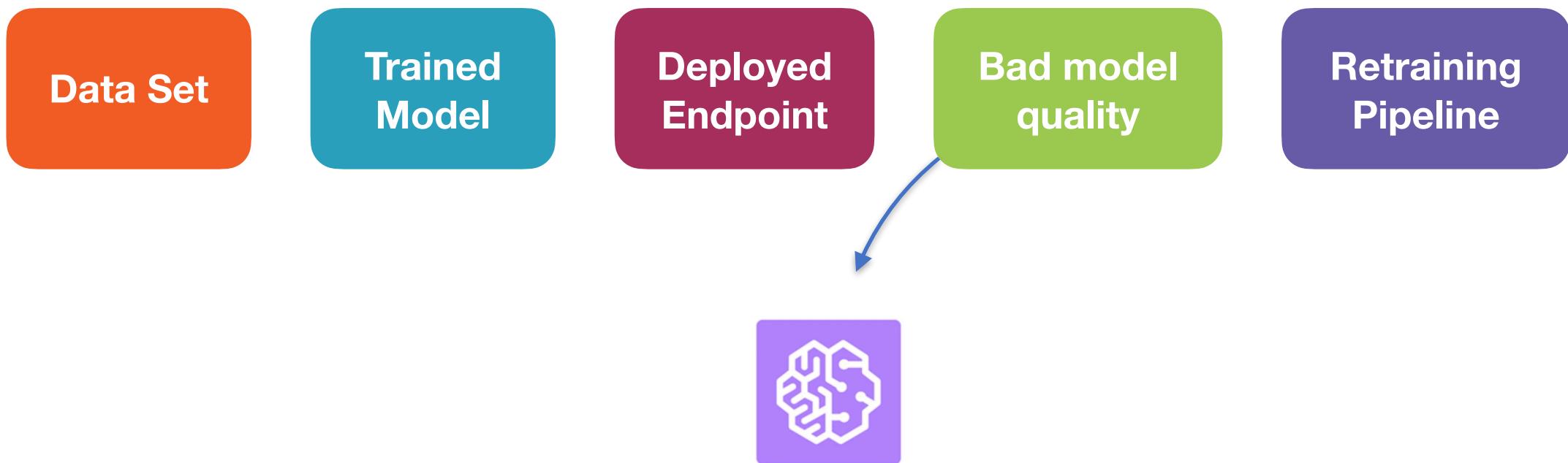
# Case Study: Price Prediction Model



Amazon SageMaker



# Case Study: Price Prediction Model



Amazon SageMaker



# Case Study: Price Prediction Model



Data Set

Trained  
Model

Deployed  
Endpoint

Bad model  
quality

Retraining  
Pipeline



Amazon SageMaker



# What is SageMaker?



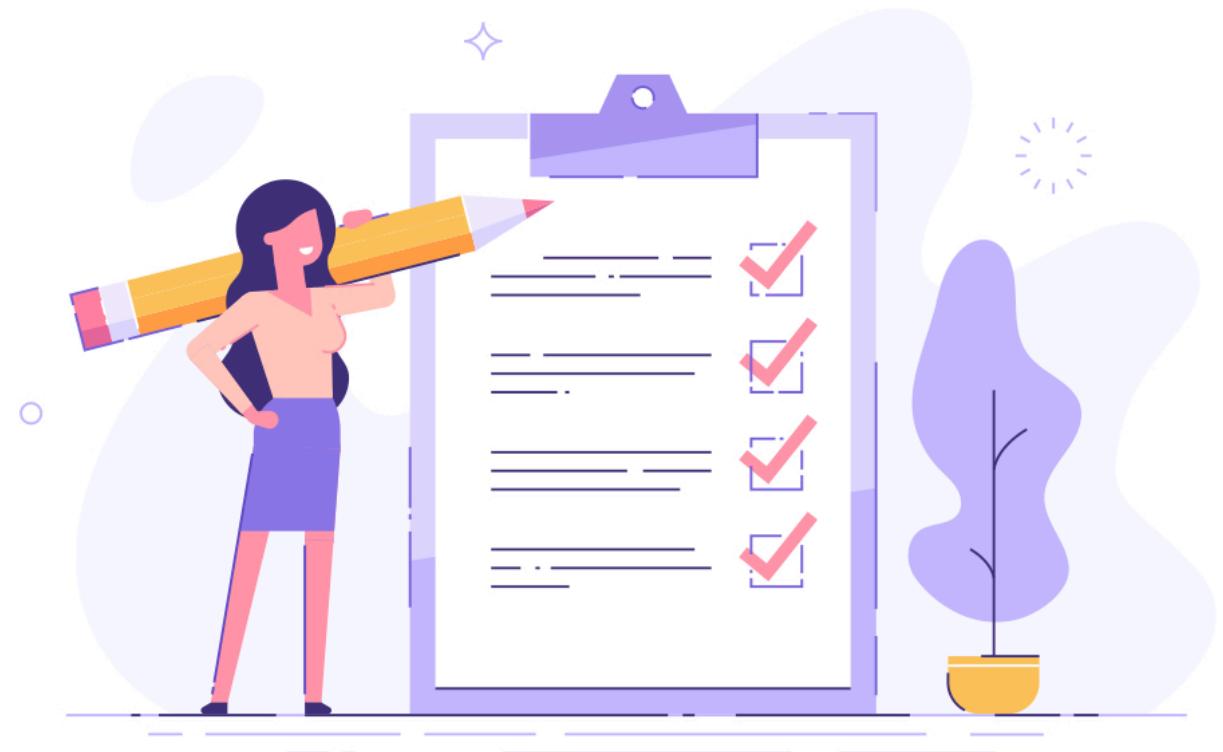
- One stop solution for all your Data Science needs
- For almost every need you have something at your display
  1. Create models -> Notebooks
  2. Deploy models -> Endpoints
  3. Monitor Models -> Model Monitor
  4. Retraining Models -> SageMaker Pipelines
  5. MLOps -> SageMaker projects
- It boosts productivity



# Exercise Time

Map in a diagram where the following live in an ML process

- Training
- Validation
- Model Serving/ Deployment
- Feature Engineering
- Model monitoring
- Model versioning



**Tip: Don't worry if you don't know it! This is more of a diagnostic exercise**



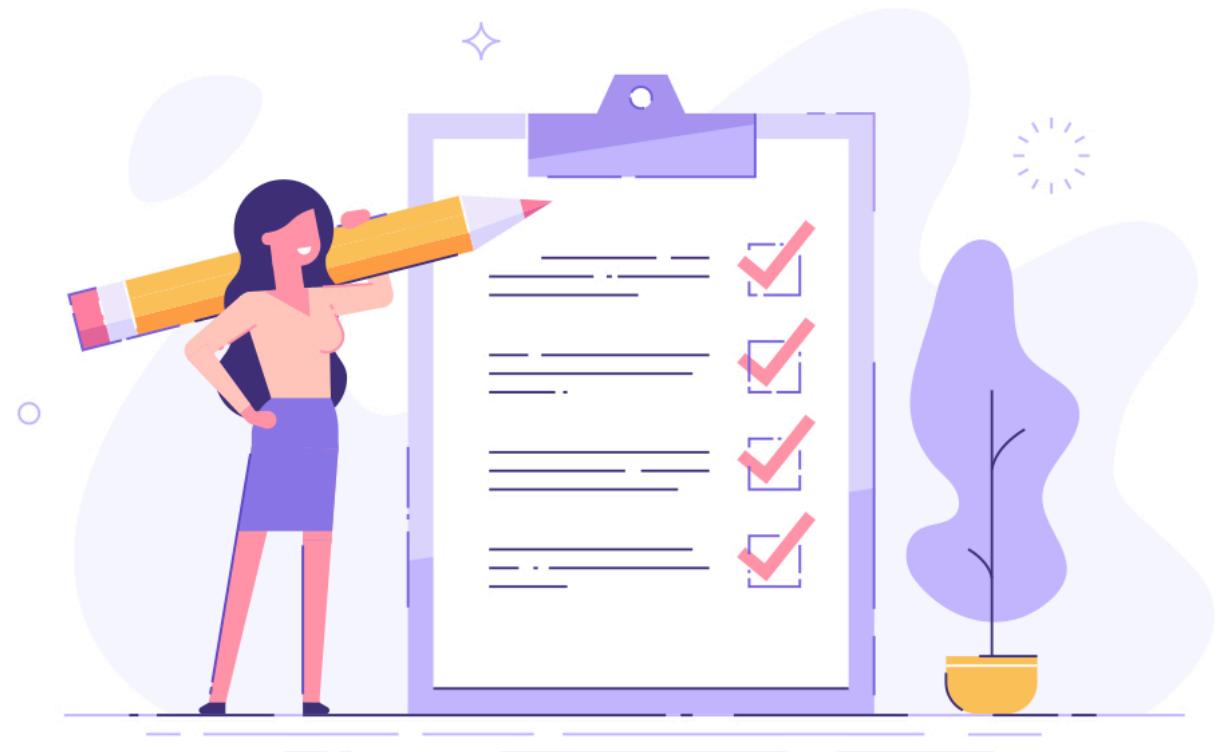
# Summary: What do we gain with SageMaker?



- Higher quality processes: Because we have a standard process to create them
- Faster delivery: Because all the setup is already pre-done, we only care of the business problem
- Lower costs: Because we don't need to over provision
- More Flexibility: To deploy whatever version and whatever number of versions of every model.



# Pulse Check



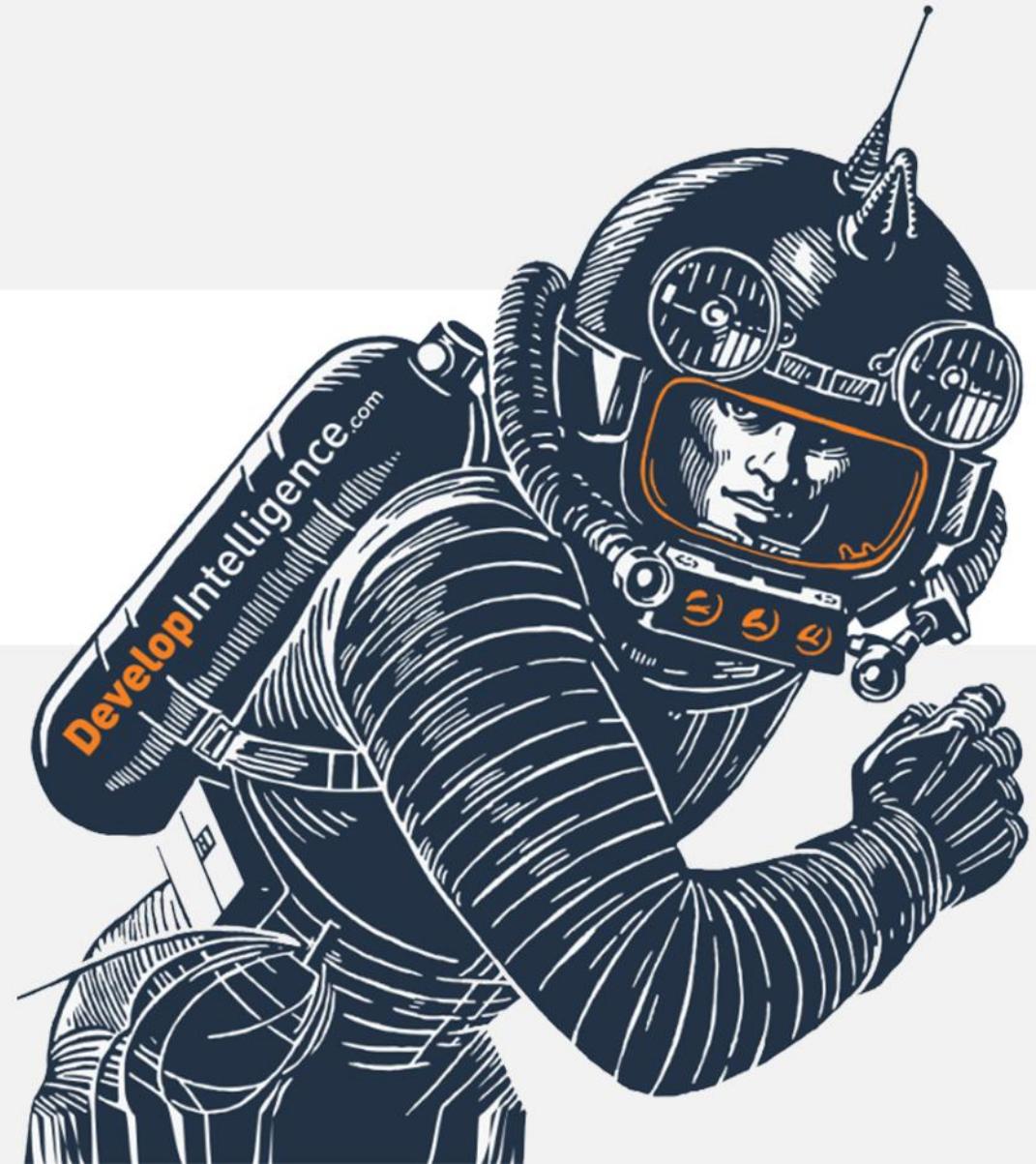
# Break: 15 minutes



# SageMaker Fundamentals



Setting Up





# Agenda

## Introduction

- Introductions
- Expectations
- Why SageMaker?

## Deploying

- Training and deploying
- Exposing a REST api
- Model monitor

## Workshop

## SageMaker Fundamentals

- Setting up
- Data Analysis
- Building builtin and BYO models

## MLOps

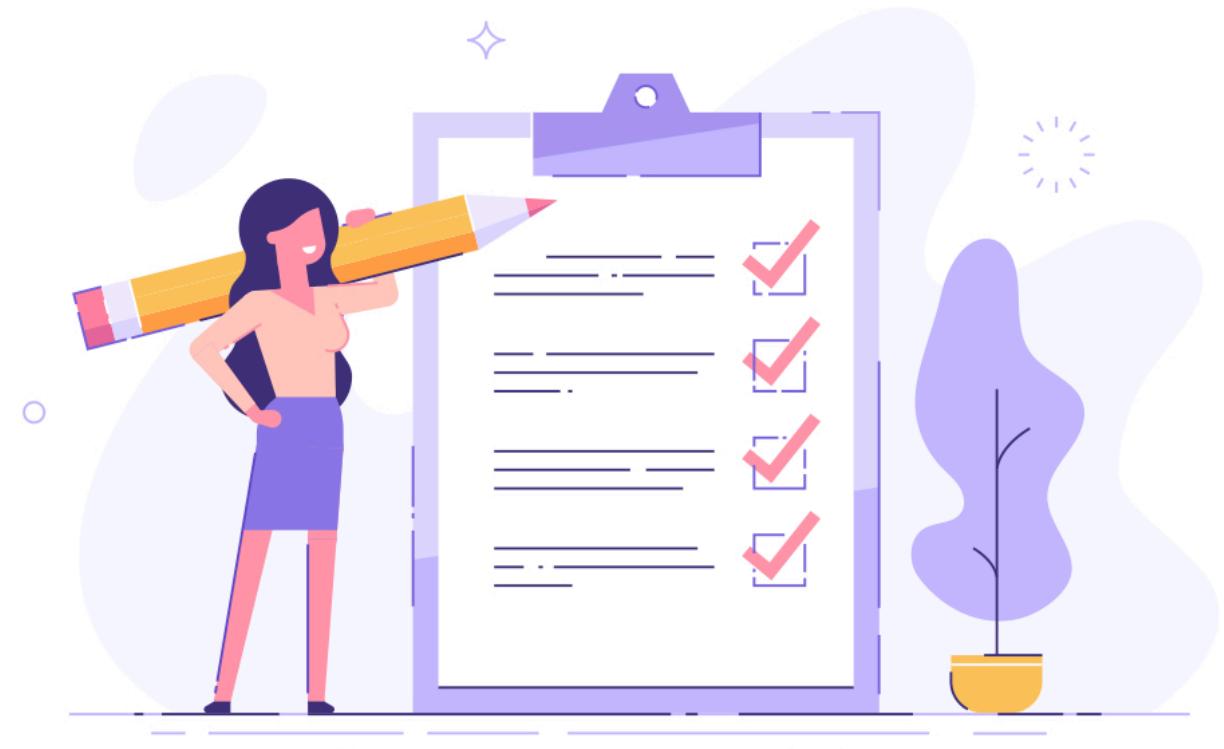
- What is MLOps?
- Model drift, versioning and Projects



# LAB: Setting Up

- Create an S3 bucket for our data
- Clone the project to work on
- Create a Notebook Instance

Allocated time: 30 minutes  
(depending on troubleshooting!)





# Regression as Case Study

Prediction

Supervised  
Learning

Unsupervised  
Learning



# Regression as Case Study



**Regression**

Predict a continuous variable

**Supervised  
Learning**

**Classification**

Predict a categorical variable

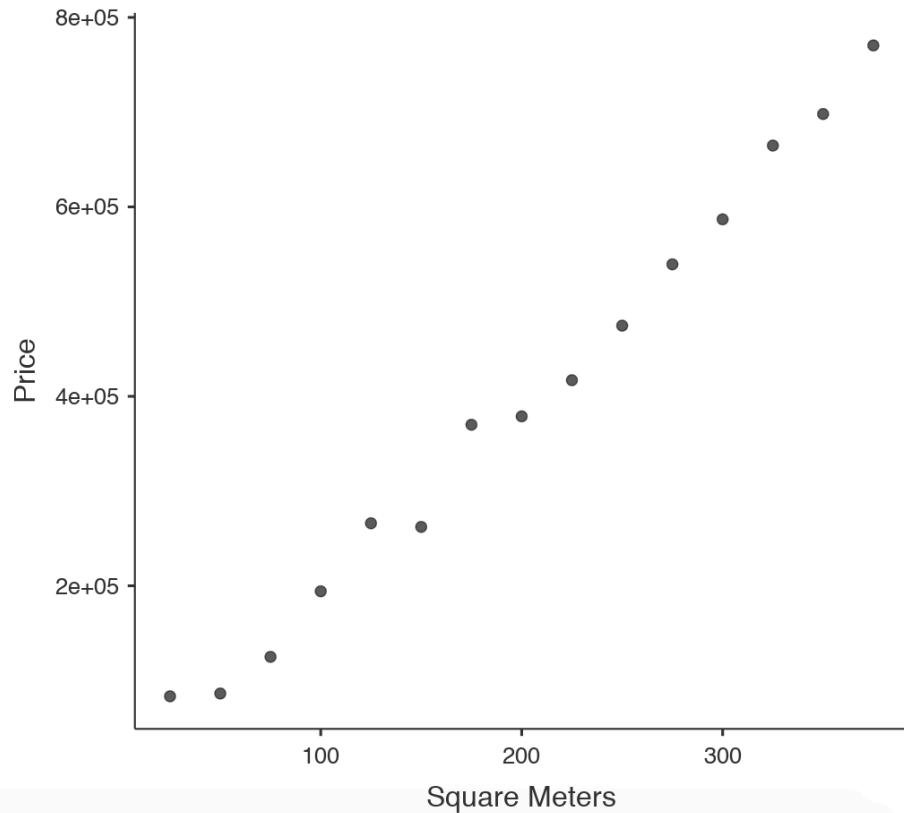
**Forecasting**

Predict a variable in a time series



# Regression

Scatterplot

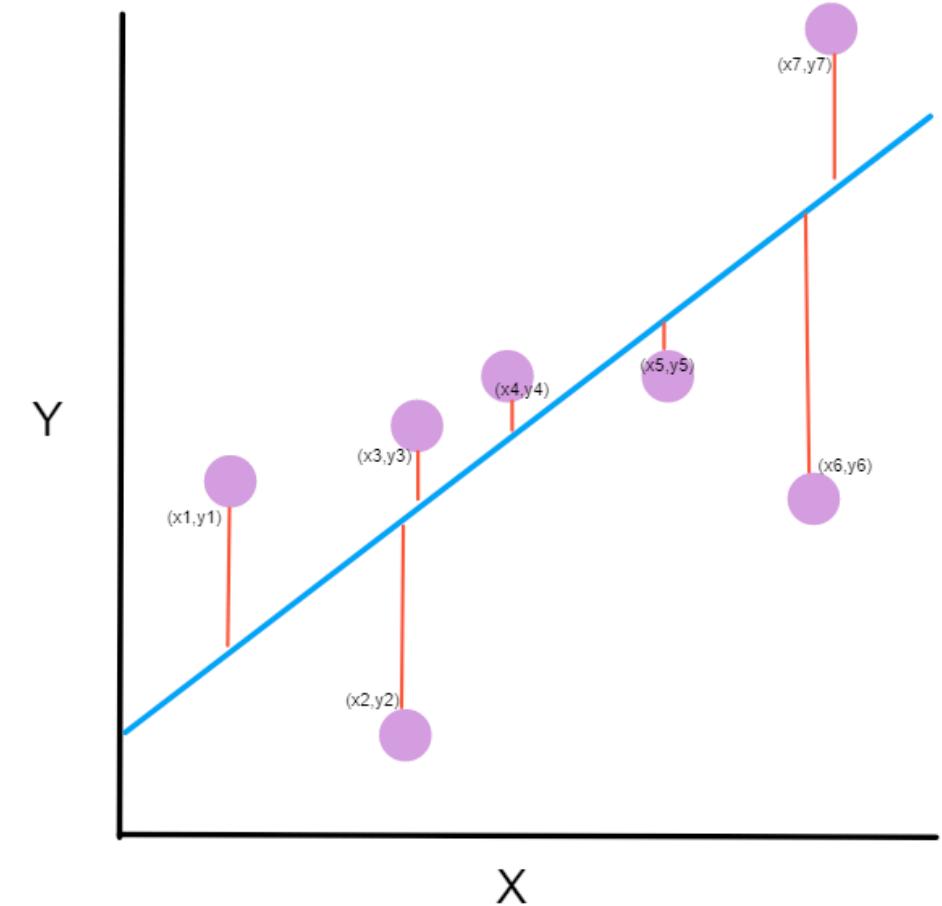


What would be the estimated price for a house of 212 square meters?



# Regression

- We parametrise the line
- The “training” consists of trying multiple lines until getting to the best one
- How? Using an optimiser that tries to minimise “the sum” of the red lines





The Boston Dataset tracks house prices in Boston area along with the following features:

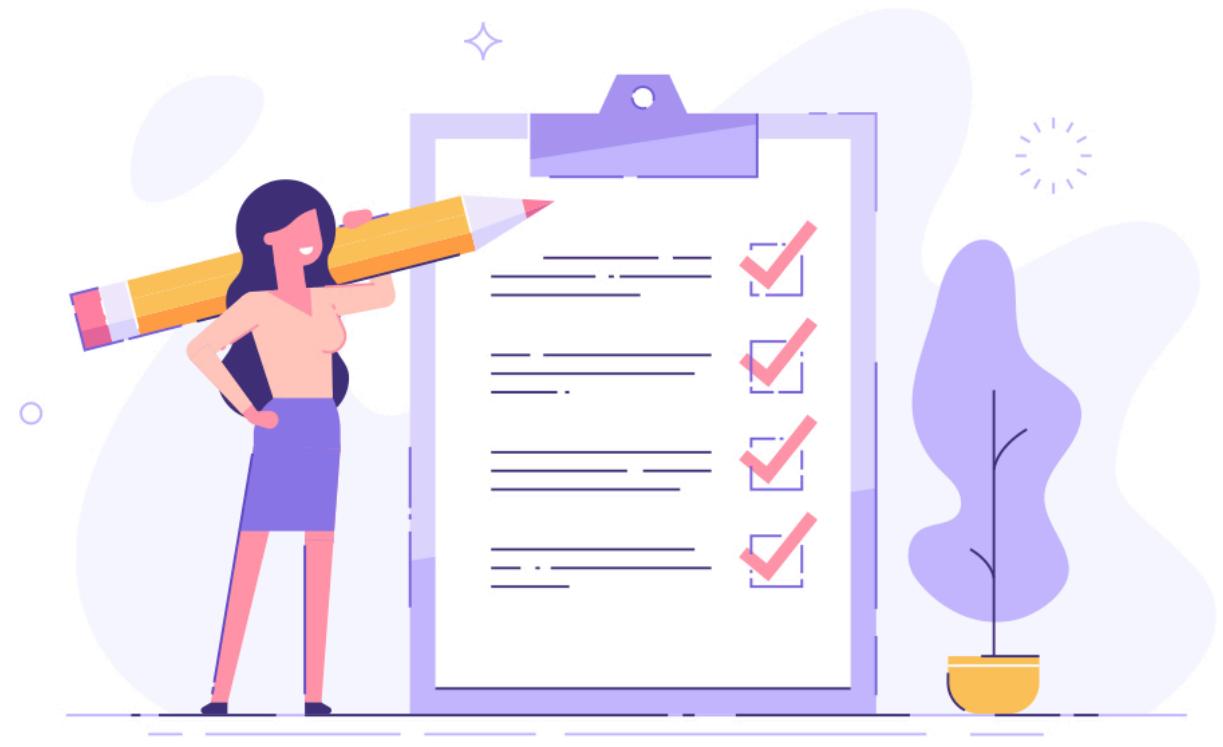
- CRIM - per capita crime rate by town
- ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS - proportion of non-retail business acres per town.
- CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
- NOX - nitric oxides concentration (parts per 10 million)
- RM - average number of rooms per dwelling
- AGE - proportion of owner-occupied units built prior to 1940
- DIS - weighted distances to five Boston employment centres
- RAD - index of accessibility to radial highways
- TAX - full-value property-tax rate per \\$10,000
- PTRATIO - pupil-teacher ratio by town
- B -  $1000(Bk - 0.63)^2$  where Bk is the proportion of blacks by town
- LSTAT - \% lower status of the population



# LAB: Loading and Processing data

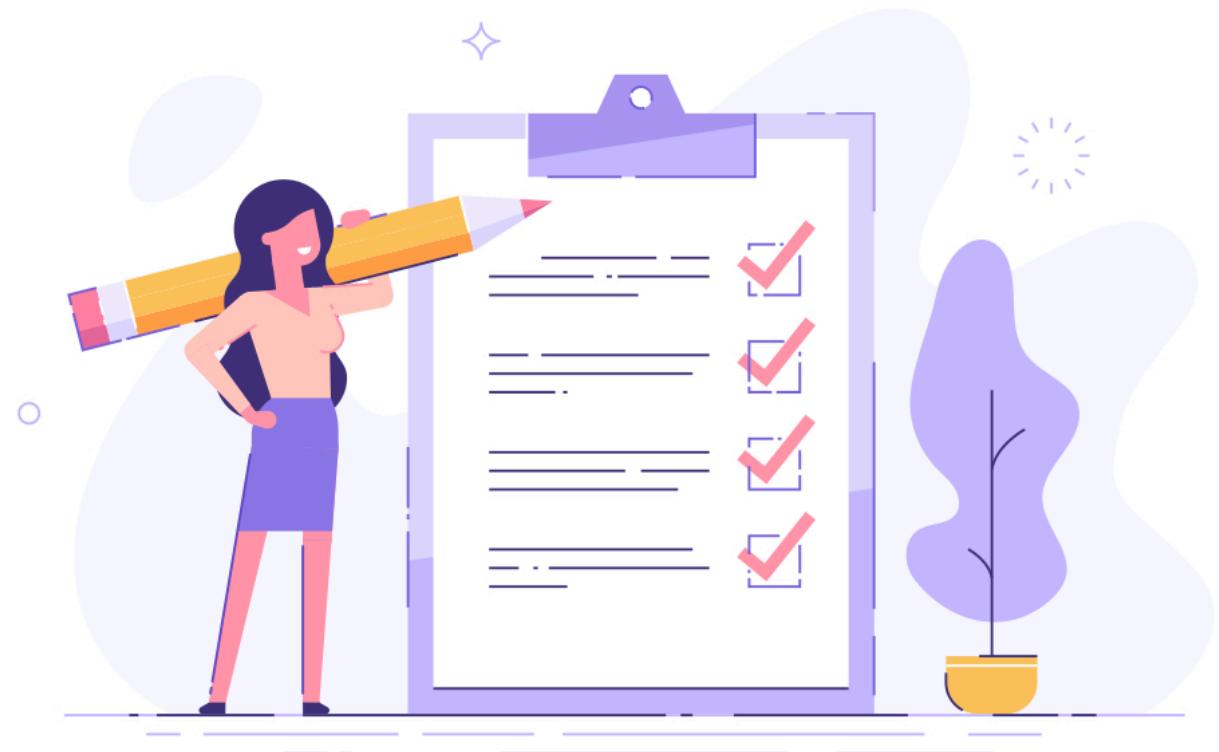
- Load the Boston Dataset
- Preprocess it
- Store it for future notebooks

Allocated time: 30 minutes





# Pulse Check





# SageMaker for Building models



## Use built-in algorithms

- Several, battle tested, algorithms for your use
- Optimised for production
- Easier to dispatch training and deploy

## Use your own algorithms

- Any algorithms you want (you bring your own Docker image)
- Any language or framework
- But you can use Tensorflow or MXNet images if you prefer as middle ground



# Built-in algorithms



**Regression**

**Linear Learner and XGBoost**

**Recommendations**

**Factorization Machines**

**Identify groups**

**K-Means**

**Dimensionality reduction**

**PCA**

**Image Classification**

**Image classification algorithm**

**Machine Translation**

**Sequence to sequence algorithm**

**Topic modelling**

**LDA**

**Neural Topic modelling**

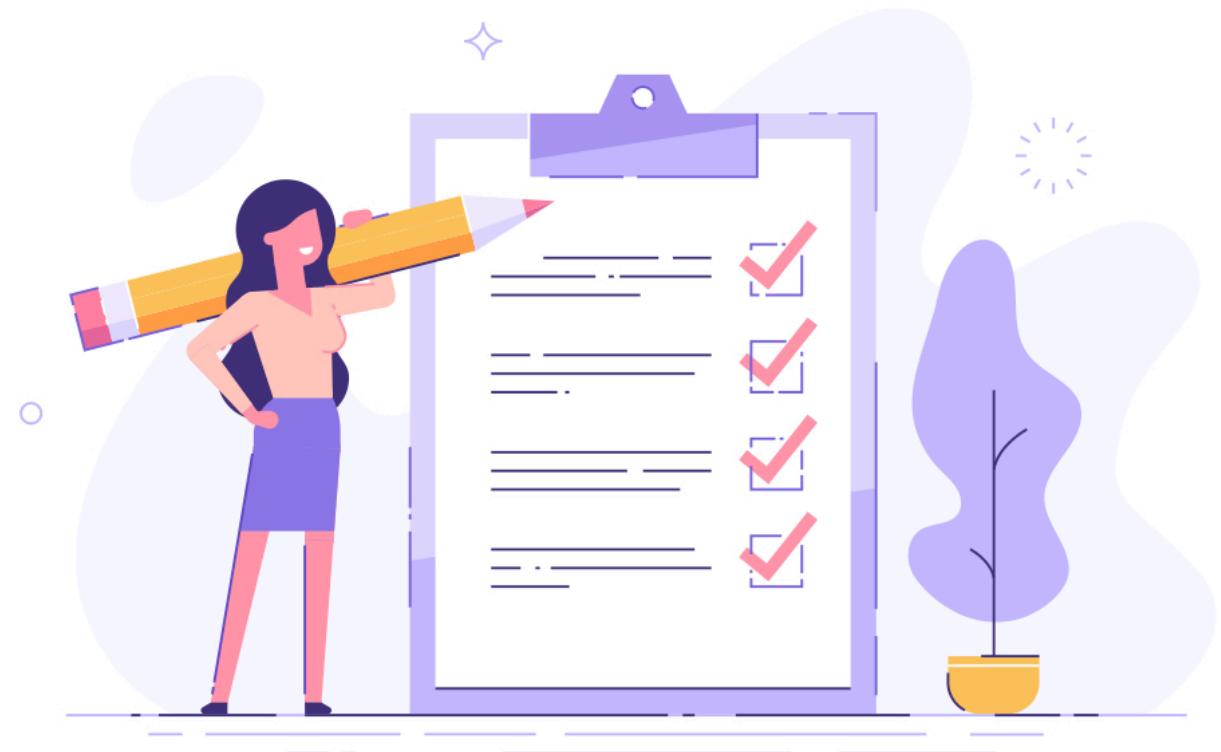
**Neural Topic Modelling algorithm**



# LAB: Train a Linear Learning

- Build and Train a linear learner

Allocated time: 15 minutes





Some tricks to keep in mind:

- Sagemaker reads from S3 the data
- It may be in text file or RecordIO for performance
- In Built-in algorithms the first column is the target
- Metrics and logs are in CloudWatch

# Break: 15 minutes





- There are many choices when it comes to hyperparameter choices
  - A. Learning rate
  - B. Batch size
  - C. Do we regularize?
- How do we choose? -> Hyperparameter Optimization
- We run all and just choose, right?

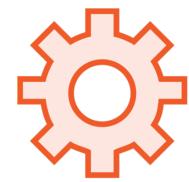


**Overfitting**

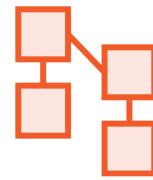
**The loss depends on  
the parameters, so we  
chase our tail**



Hyperparam 1



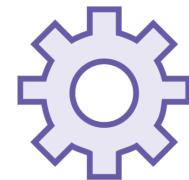
Trained Model 1



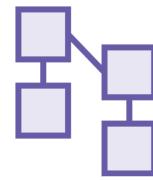
Metrics on Model 1



Hyperparam 2



Trained Model 2



Metrics on Model 2



Hyperparam 3



Trained Model 3



Metrics on Model 3

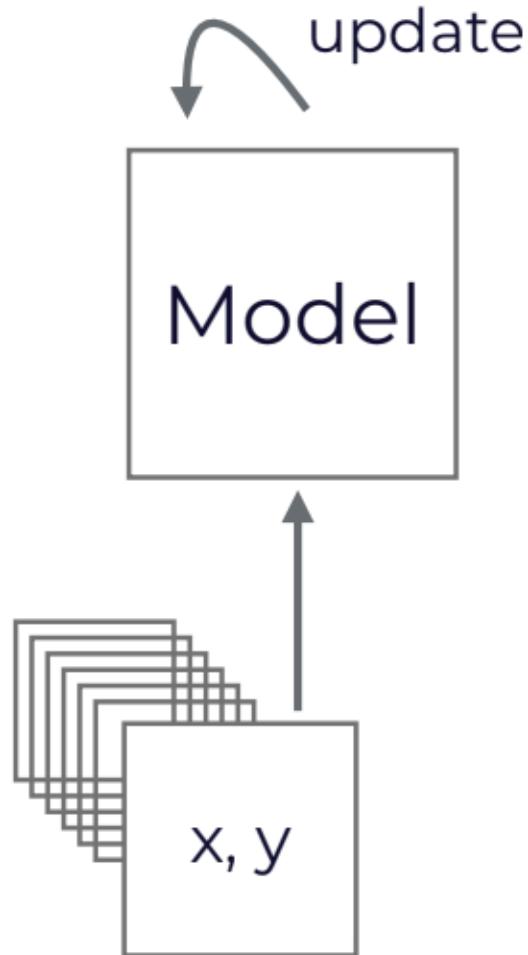


COMPARE





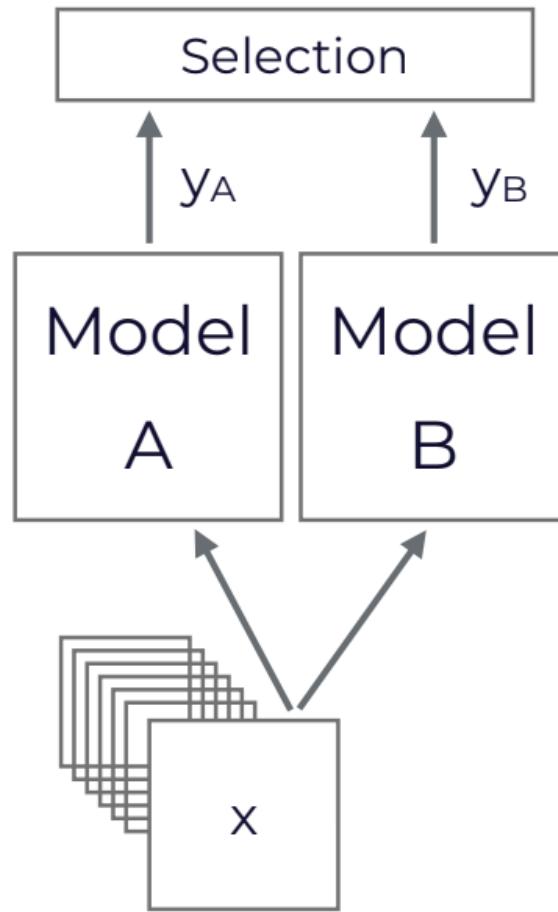
# Training set



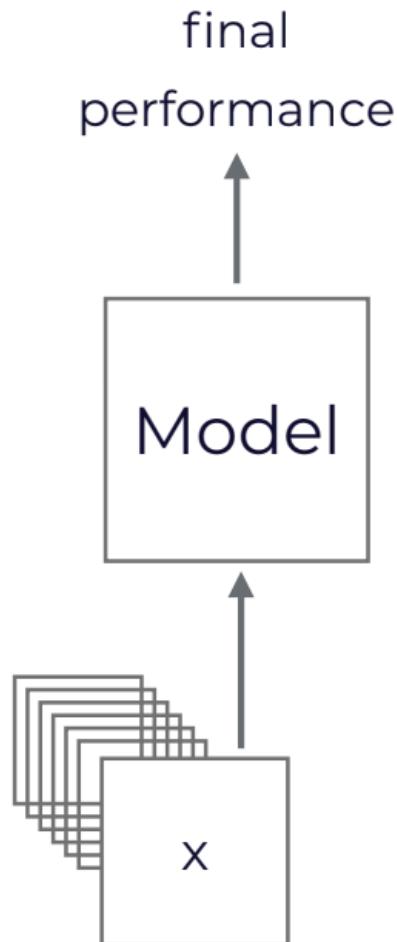
- Therefore, we have a training set to train the model.
- We will use the same set on every training run
- This is the train channel in SageMaker



# Validation set



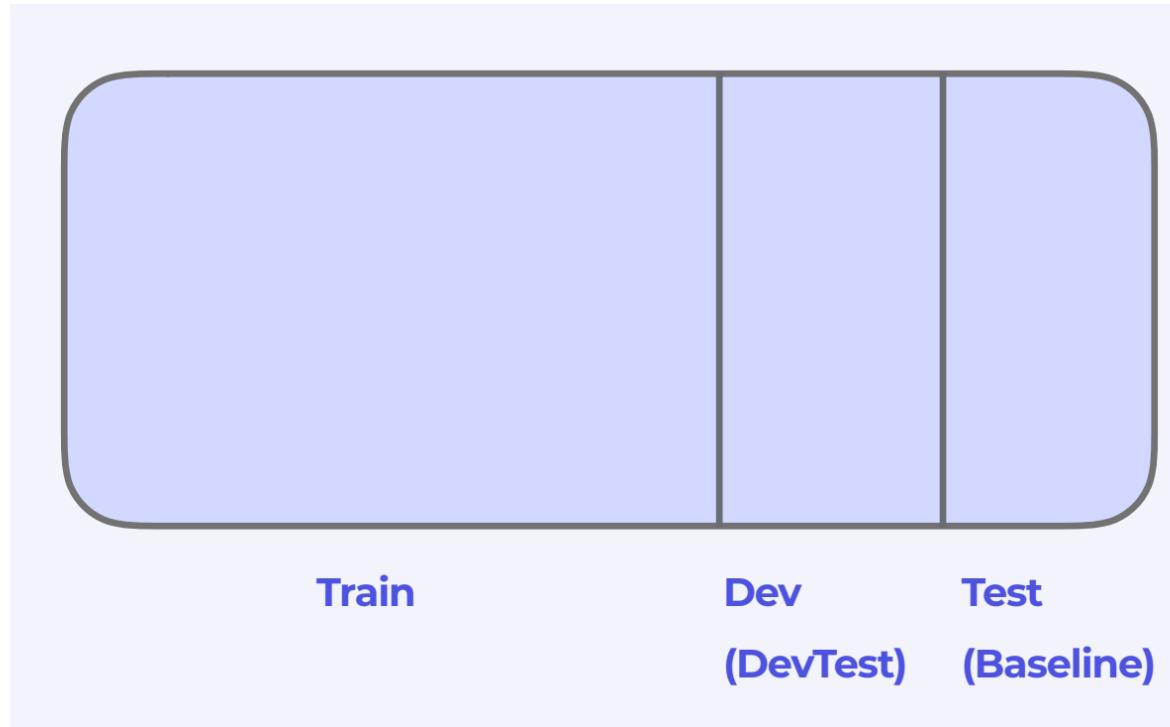
- Then, we have a validation set to accurately calculate the loss
- This way, we can choose the best model
- This is done by SageMaker if we provide the validation channel



- Finally, we have a test to assess model performance
- Why would we need another dataset?



# How much is enough?



**Normal datasets → training set around**

**70% of the data**

**Deep learning → training set as high as**

**97% of the data**



## Using built-in algorithms

- You don't need to specify metrics
- You just choose the one you like
- And set the hyper parameters you prefer

## Use your own algorithms

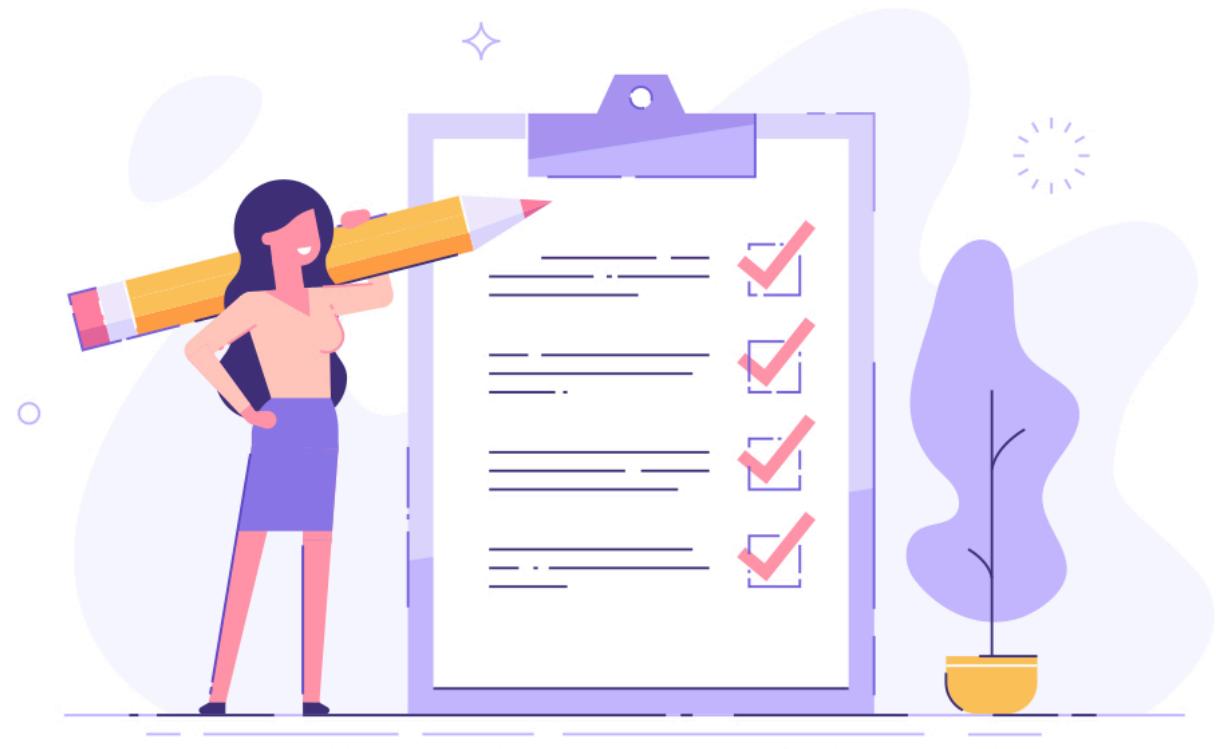
- It has to emit at least one metric on stderr or stdout
- It can have up to 20 metrics
- You define metrics with a regular expression



# LAB: Configuring HPO

- Configure HPO

Allocated time: 20 minutes



# Deploying and Testing



Deploying and REST APIs





# Agenda

## Introduction

- Introductions
- Expectations
- Why SageMaker?

## Deploying

- Training and deploying
- Exposing a REST api
- Model monitor

## Workshop

## SageMaker Fundamentals

- Setting up
- Data Analysis
- Building builtin and TF models

## MLOps

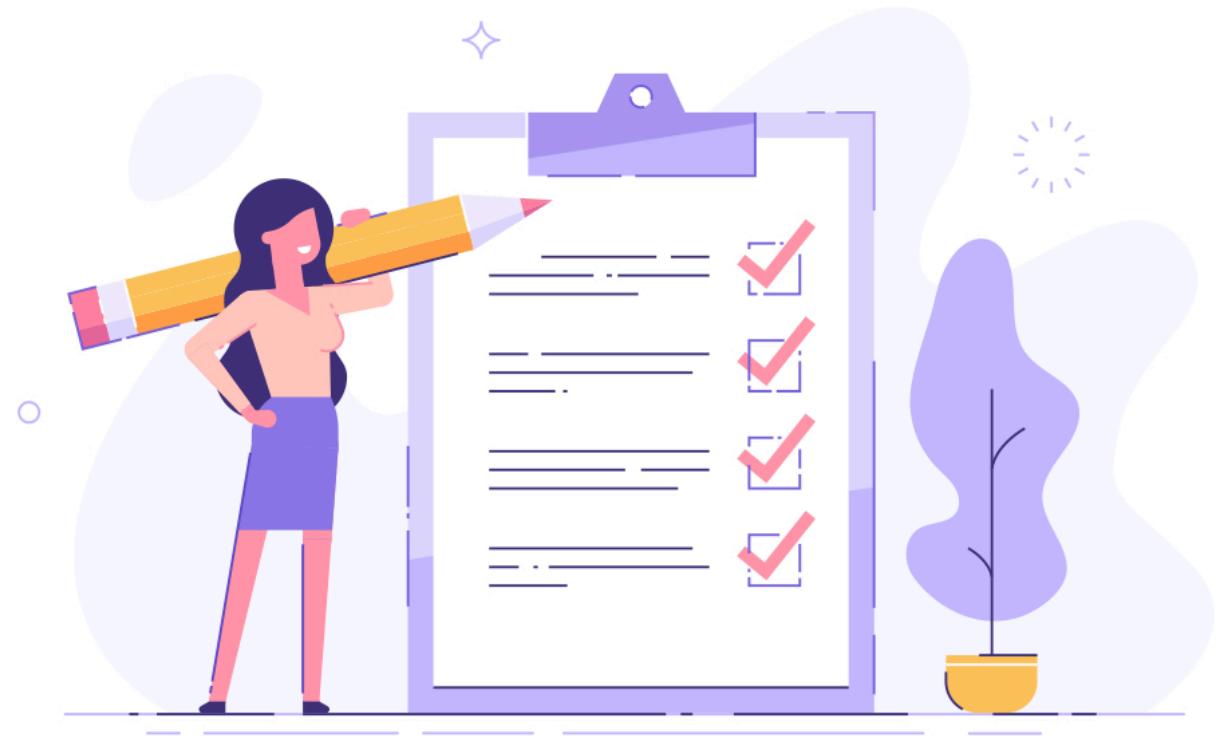
- What is MLOps?
- Model drift, versioning and Projects



# LAB: Deploy and validate

- Deploy to a real time endpoint
- Calculate RMSE

Allocated time: 20 minutes



# Break: 15 minutes





# Endpoints



## Real Time Endpoints

- 24x7 Live
- If multiple instances, they are deployed different AZ
- Created for production workloads

## Batch Endpoints

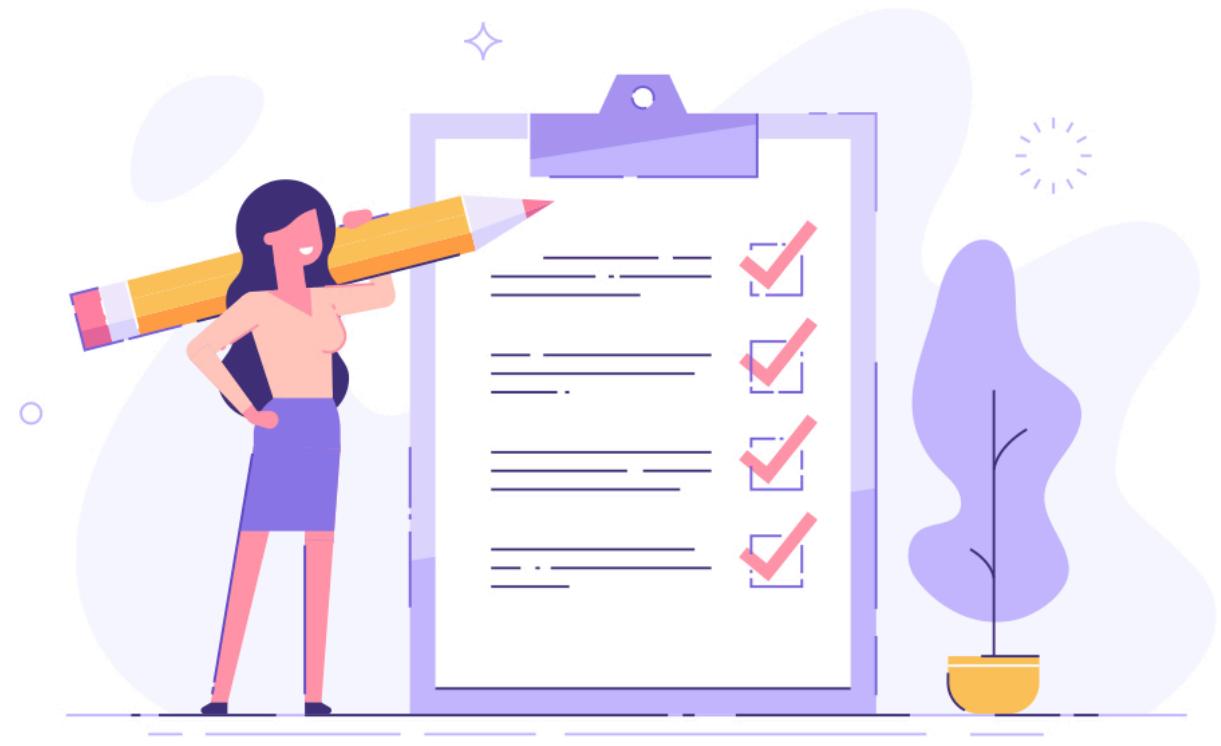
- You need a prediction for a period of time and then it can shut down
- It needs a moment to start up
- Amazingly popular for batch jobs like AI augmentation on Data Pipelines
- May run on Spot instances with less risk



# LAB: AB Testing

- Deploy multiple models in an endpoint

Allocated time: 20 minutes





Some tricks to keep in mind:

- HPO tunes the parameters we set and tries different values in a Bayesian way
- It is optimal to find the best model easily
- Deploying a model to an endpoint can be a single line
- And calling the endpoint a method in AWS
- But we can also call it from the API with HTTP

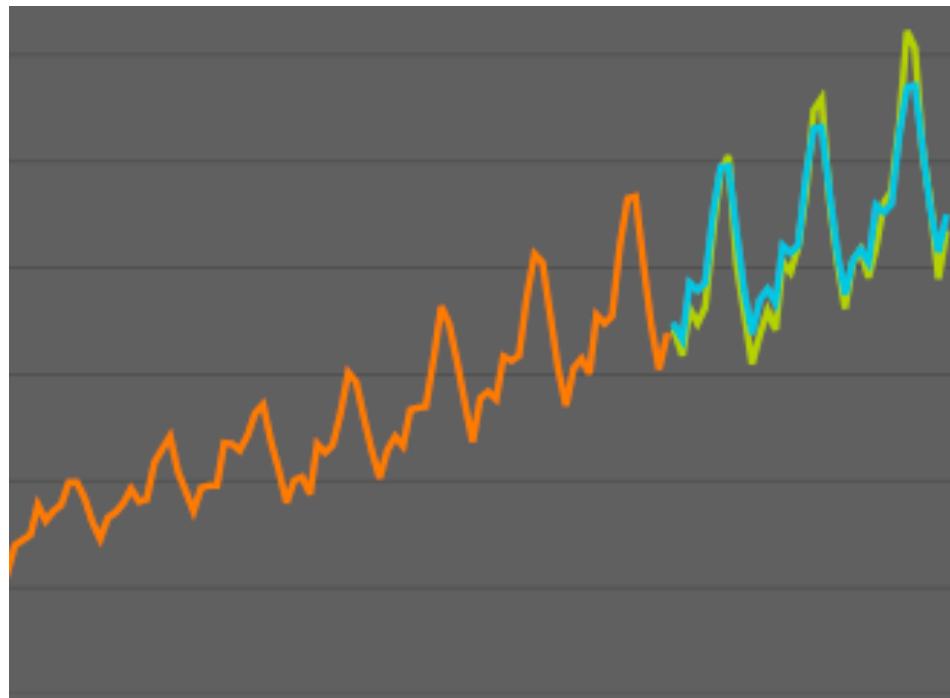


# Summary

- SageMaker enables us to preprocess-build-train-deploy models
- It offers notebooks to do all the work, which are shareable
- It has training and HPO jobs for training.
- It offers Endpoints for deploying



# Forecasting



- When we have time series data there is a complicated component: time
- We cannot go back in time, so regression most times won't work
- We usually split the data into 3 additive trends:
  1. Trend: Moving averages of the data
  2. Seasonality: Trying to model with a fixed width and amplitude (usually a Fourier series)
  3. Residues: What we cannot cover



# Forecasting



PROPHET

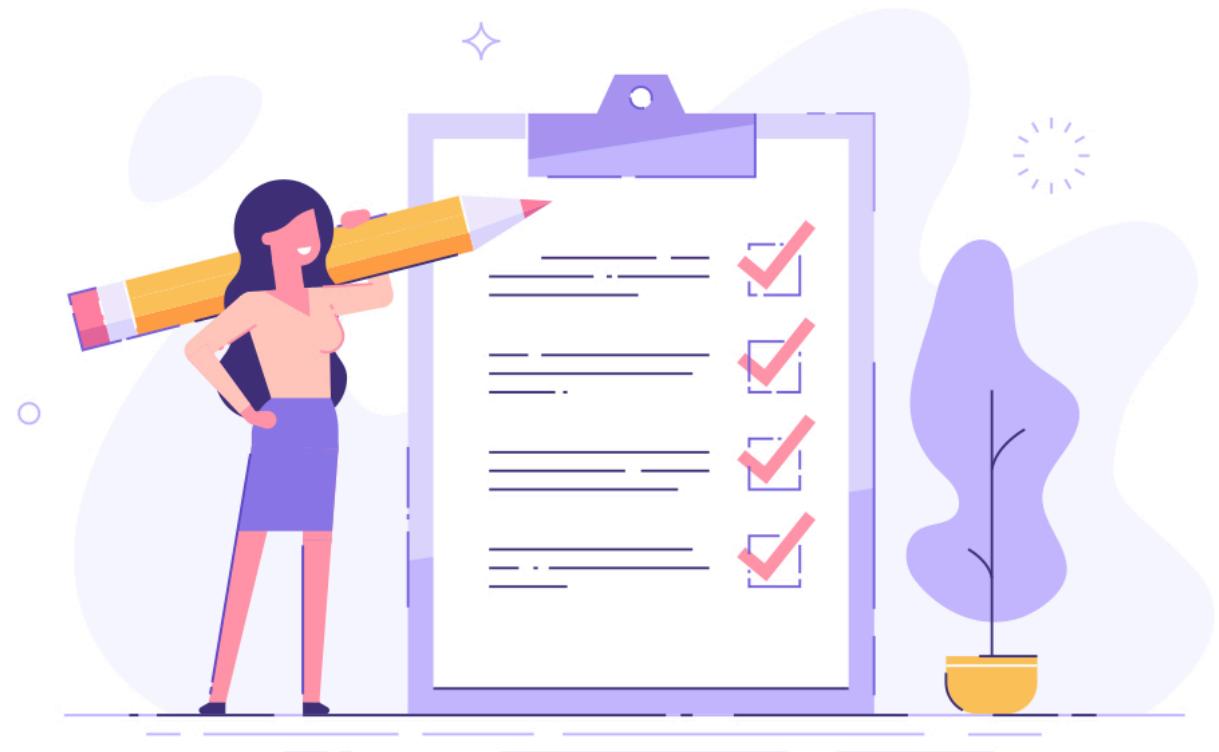
- Facebook created a model called Prophet
- It disentangles the data and offers a very simple API
- Before it one usually relied on ARIMA and advanced statistical tools



# Mini-LAB: Time Series Data

- Investigate seasonality!

Allocated time: 10 minutes

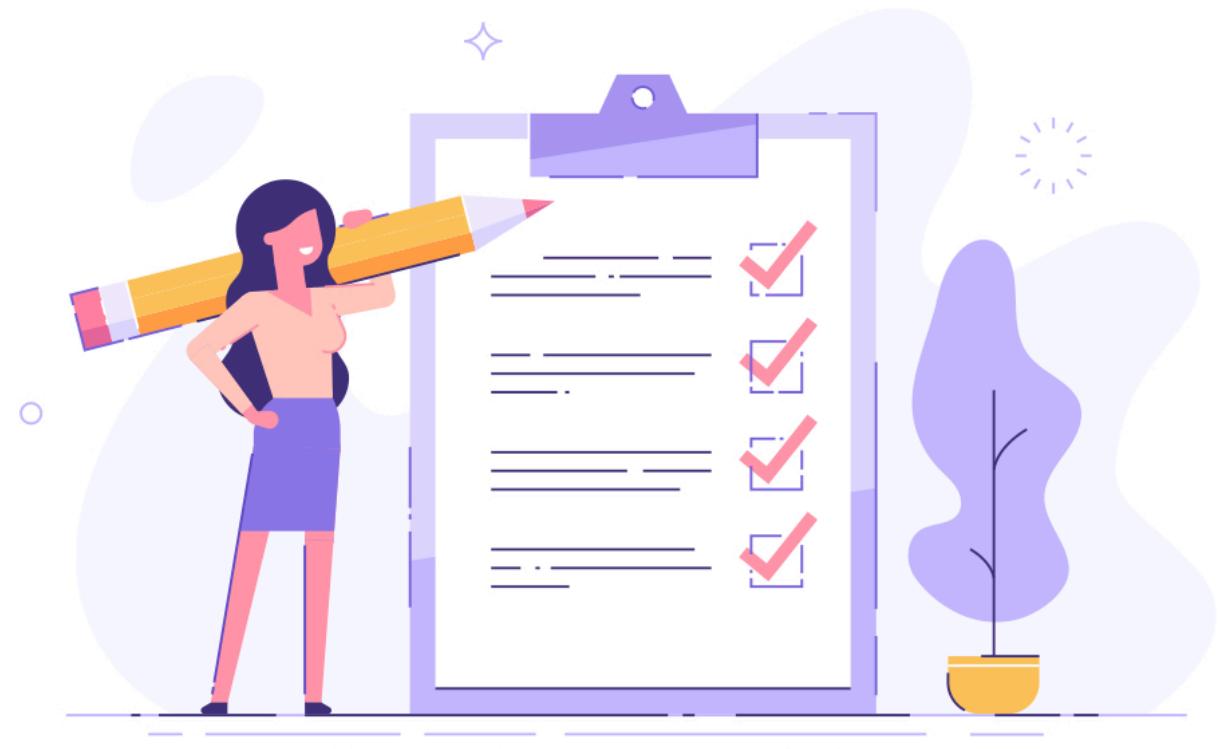




# LAB: Build and Train a forecast algorithm

- Build and Train a BYO algorithm
- I recommend playing around in a notebook with the model, unpicking it and plotting it!

Allocated time: 30 minutes

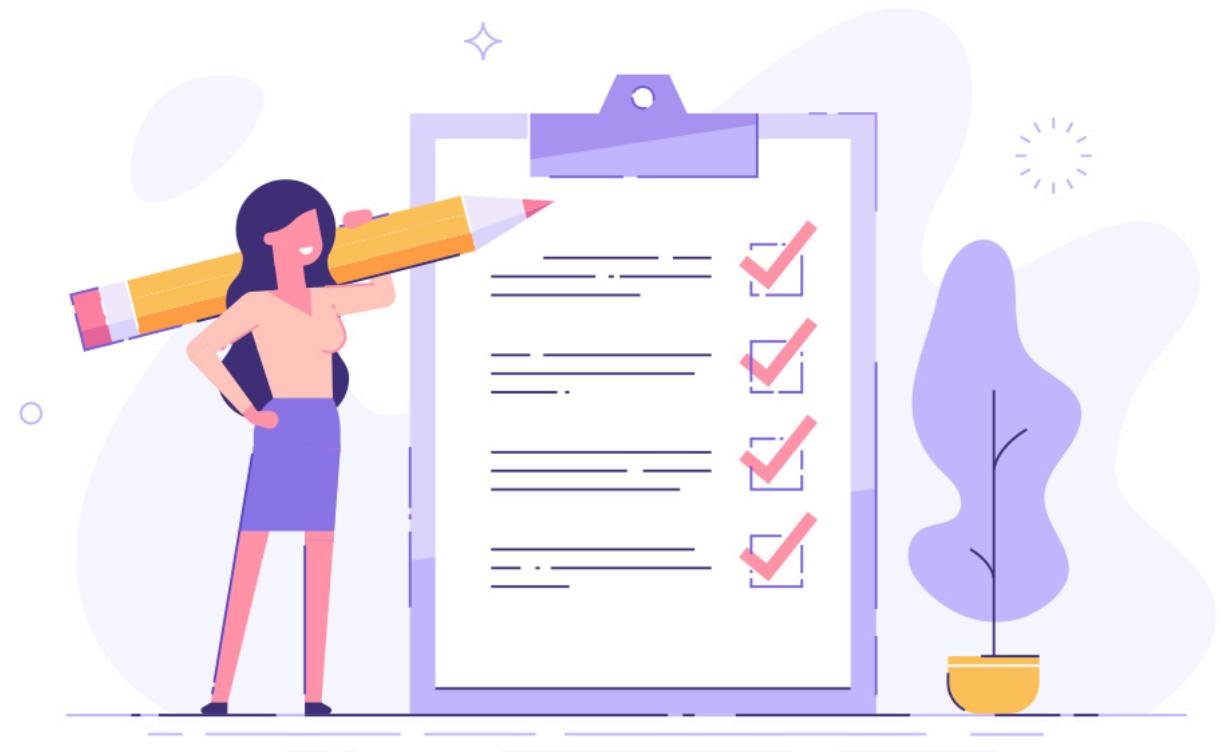




# Workshop

- Choose one of the models we did (or try another one!)
- Perform HPO
- Deploy an endpoint
- Create a lambda that calls it
- Expose a REST Api with API

Gateway to expose the model





# What we try to do



# End of Day 1





# Agenda

## Introduction

- Introductions
- Expectations
- Why SageMaker?

## Deploying

- Training and deploying
- Exposing a REST api
- Model monitor

## Workshop

## SageMaker Fundamentals

- Setting up
- Data Analysis
- Building builtin and TF models

## MLOps

- What is MLOps?
- Model drift, versioning and Projects



# What problems arise on ML workflows?



Hard to create  
workflows

Hard to  
troubleshoot

Hard to keep  
track of models

Hard to manage  
models

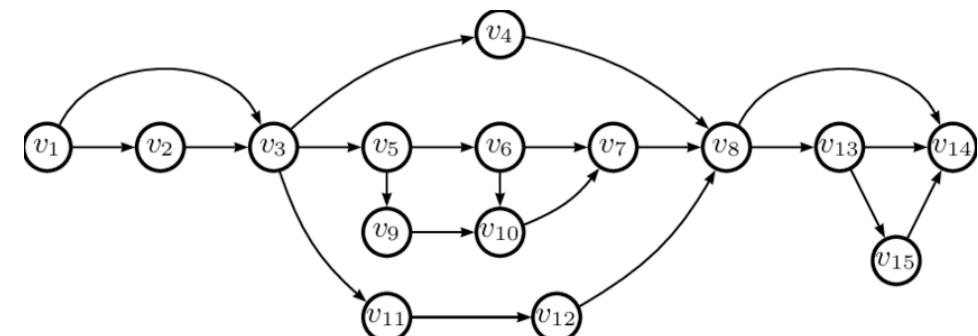
Hard to scale

SageMaker Pipelines and Projects solve this



# What is a pipeline?

- It is a DAG of operations to run
- Sagemaker integrated it in a fully managed service
- It is easier to promote models and later deploy, while keeping track





# What is a pipeline?

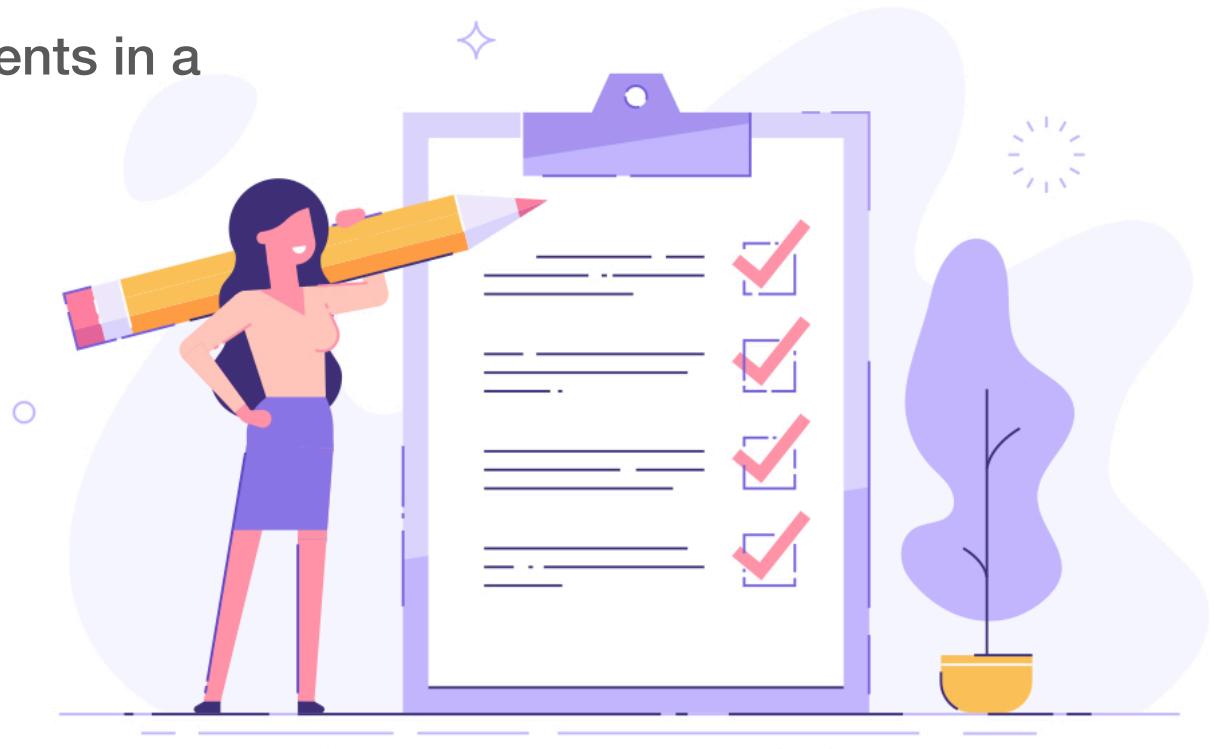
- Each box is a step
- And one can evaluate conditions
- It can be triggered by code events or data events





# Discussion

- Let's discuss, what would be the ideal workflow?
- To help the discussion let's map these elements in a process:
  - Retraining on new data
  - Model approval
  - Model promotion
  - Retraining on code changes
  - Tracking of model metrics
  - Scaling and AB Testing

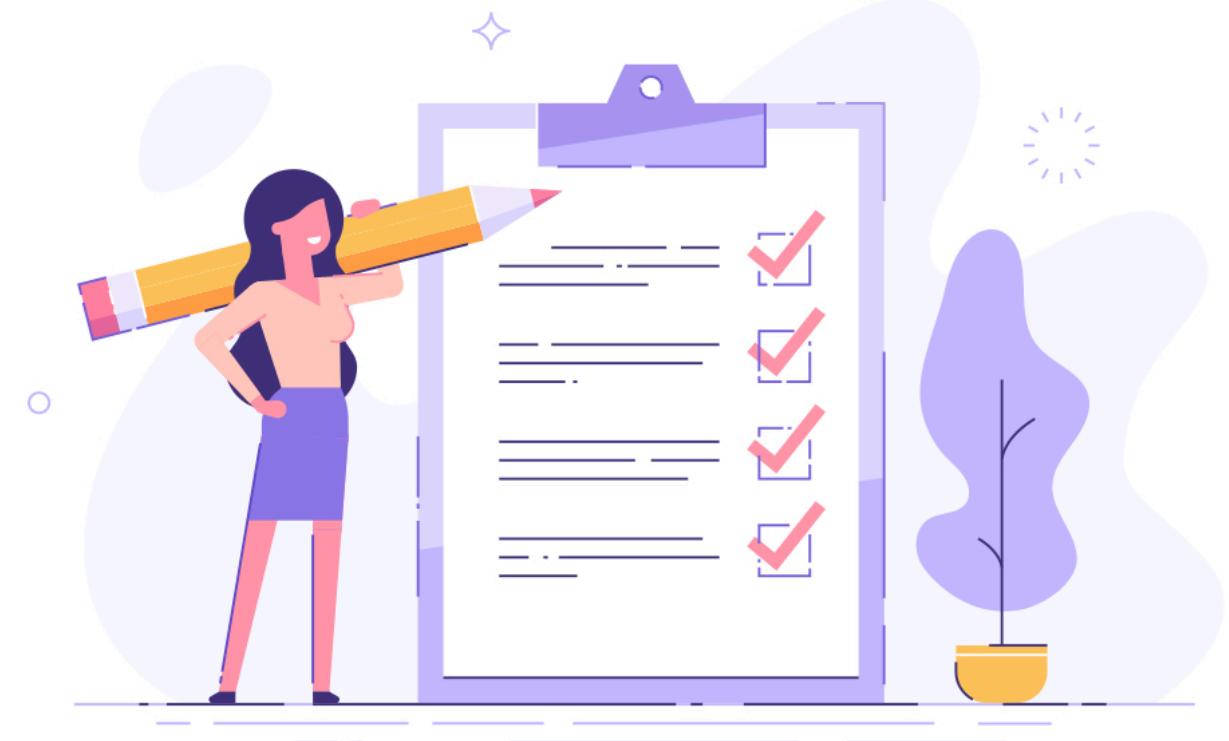




# Workshop: Create a pipeline



- Let's create a pipeline to have a TF model that predicts house prices
- Let's do it together!



Supposedly since 4 months ago we can trigger the pipeline from EventBridge, completing the loop. Haven't tried it yet!

# Please Fill in Survey



**THANK YOU**

