Stats AI

Real-World Deployment of ML in Production Systems

Author: Hussain Abbas, MSc

Why Deploy ML Models?

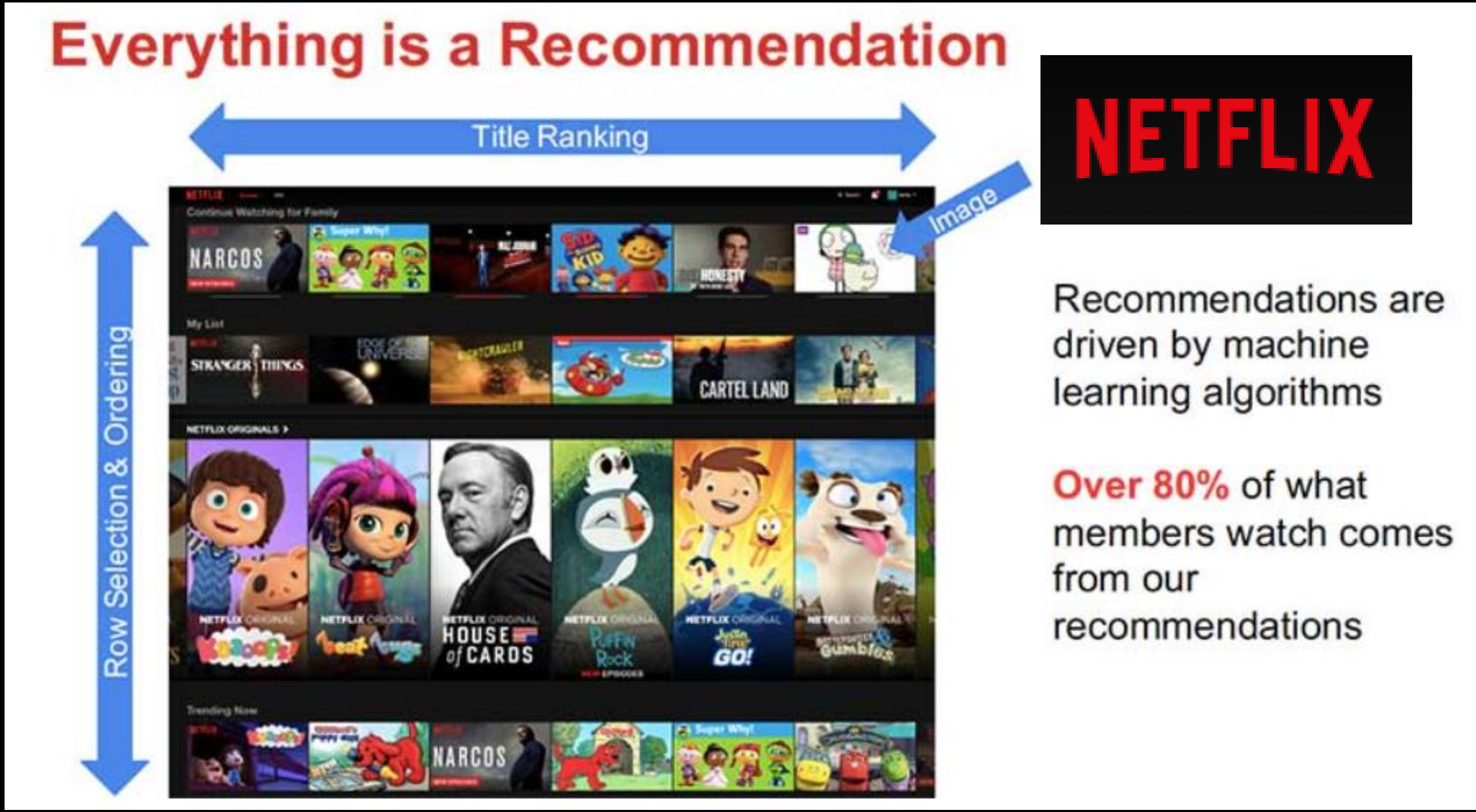
- ML model Deployment enables firms to create new services!

- The interaction between the user and the ML model <u>is the service</u>
  - Recommender Systems: Netflix Recommendation Algorithm
  - Financial Services: Algorithmic Portfolio Selection
  - Rideshare: Uber arrival time prediction

- Deployed ML models enable firms to:
  - Acquire new customers
  - Retain existing customers
  - Grow revenues
  - Cut costs
  - Gain market share
  - Gain competitive advantage
  - Stay relevant

Stats AI

Recommender Systems

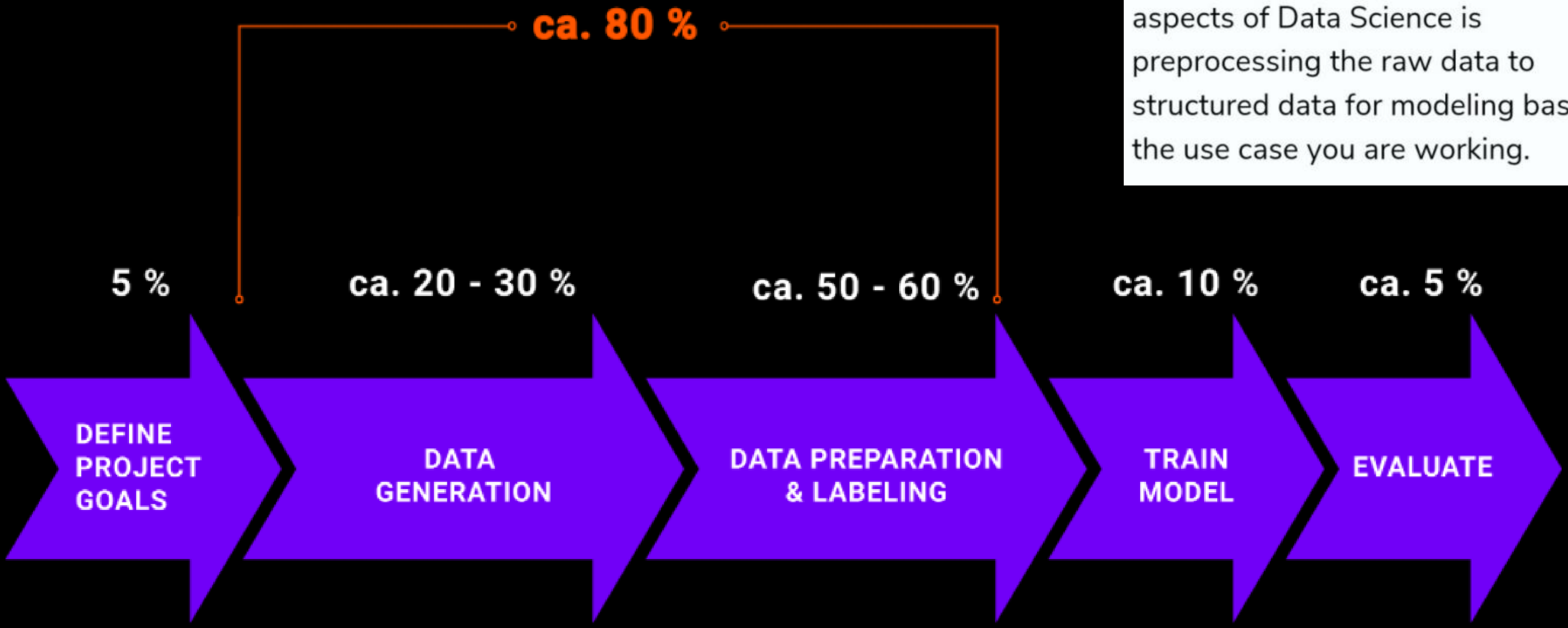


Stats AI

TYPICAL MACHINE LEARNING PROJECT

Phases and estimated time in percent

One of the important and difficult aspects of Data Science is preprocessing the raw data to structured data for modeling based on the use case you are working.



DATA GENERATION

- Aquire data (search, make or buy)
- Data generation
- Data Augmentation

DATA PREPARATION

- Store / load data
- Organize data
- Correct, normalize ..
- Label & annotate data

TRAIN & EVALUATE MODEL

- Choose model
- Train model
- Evaluate model
- Deploy model

You are here!

Stats AI

Model Deployment

- Once an ML model is developed, it needs to be deployed as a REST API so users can interact with it

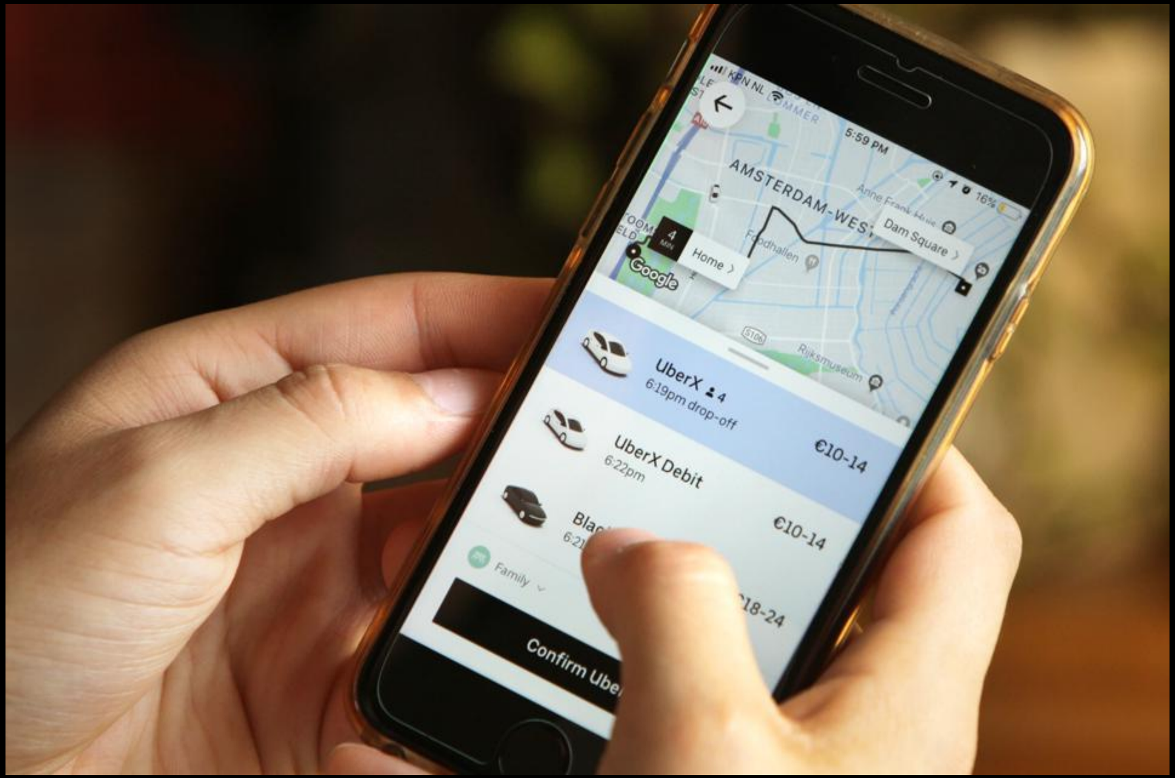
- How a ML REST API works under the hood:
  - The trained ML model is pickled
  - JSON payload delivers feature data via a POST request
  - Various transformations are applied to the JSON payload before it enters the model
  - The model takes in the input and an output prediction is generated
  - The API returns the prediction from the model in JSON
  - The JSON output is presented to the user either directly or indirectly
    - Direct: Streamlit, Gradio, GUI, etc.
    - Indirect: API output is an input into an Enterprise application

- The two types of deployment
  - Batch
    - Take in a set of inputs at a point in time, and generate predictions
  - Real-time
    - Generate predictions on the fly as requested

Stats AI

Model Deployment

- Where does the JSON payload that triggers the POST Request come from?
  - A database
  - User interaction
    - Manual user entry
    - User Location
  - Streaming data
  - All the above combined!



MySQL

Streamlit

Spark Streaming

gradio

Stats AI

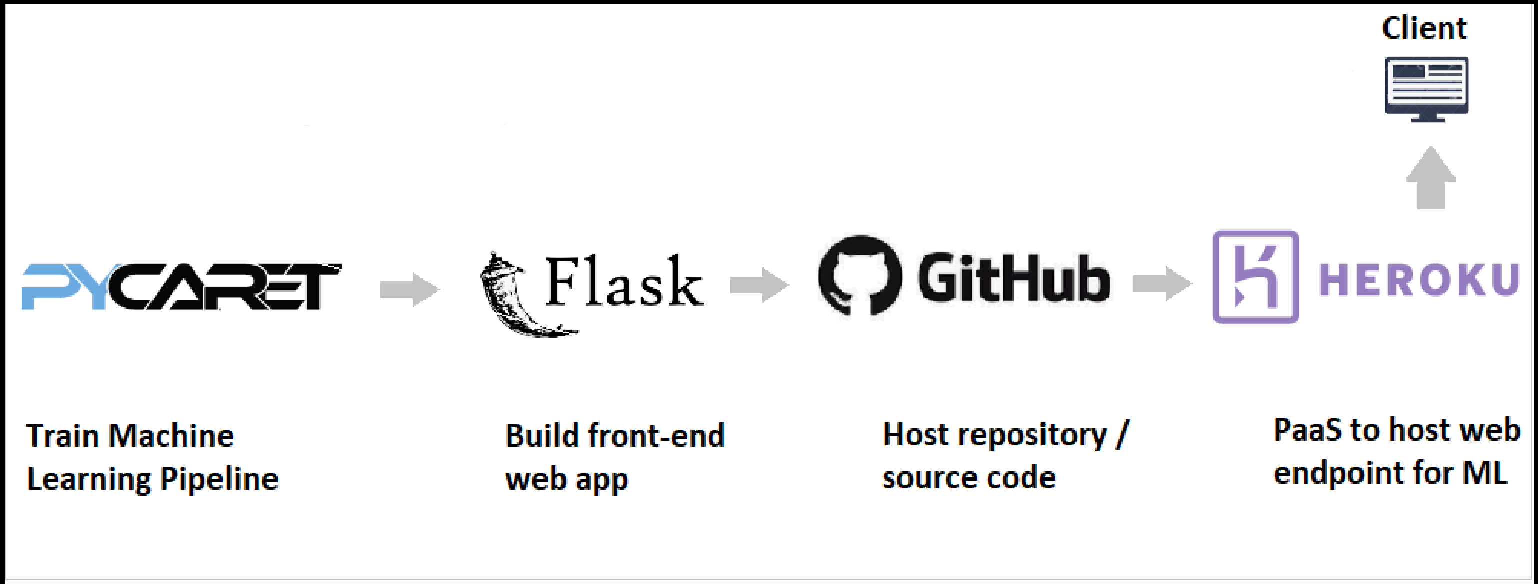
Model Deployment

- Generally, external customer facing services require a level of engineering effort magnitudes larger than internal services
- As internal services become more complex, they start to approximate external services
- Thus, what is needed are streamlined deployment systems that enable rapid iteration that can scale with complexity and demand … hence the Cloud

Deployment in the Cloud

Stats AI

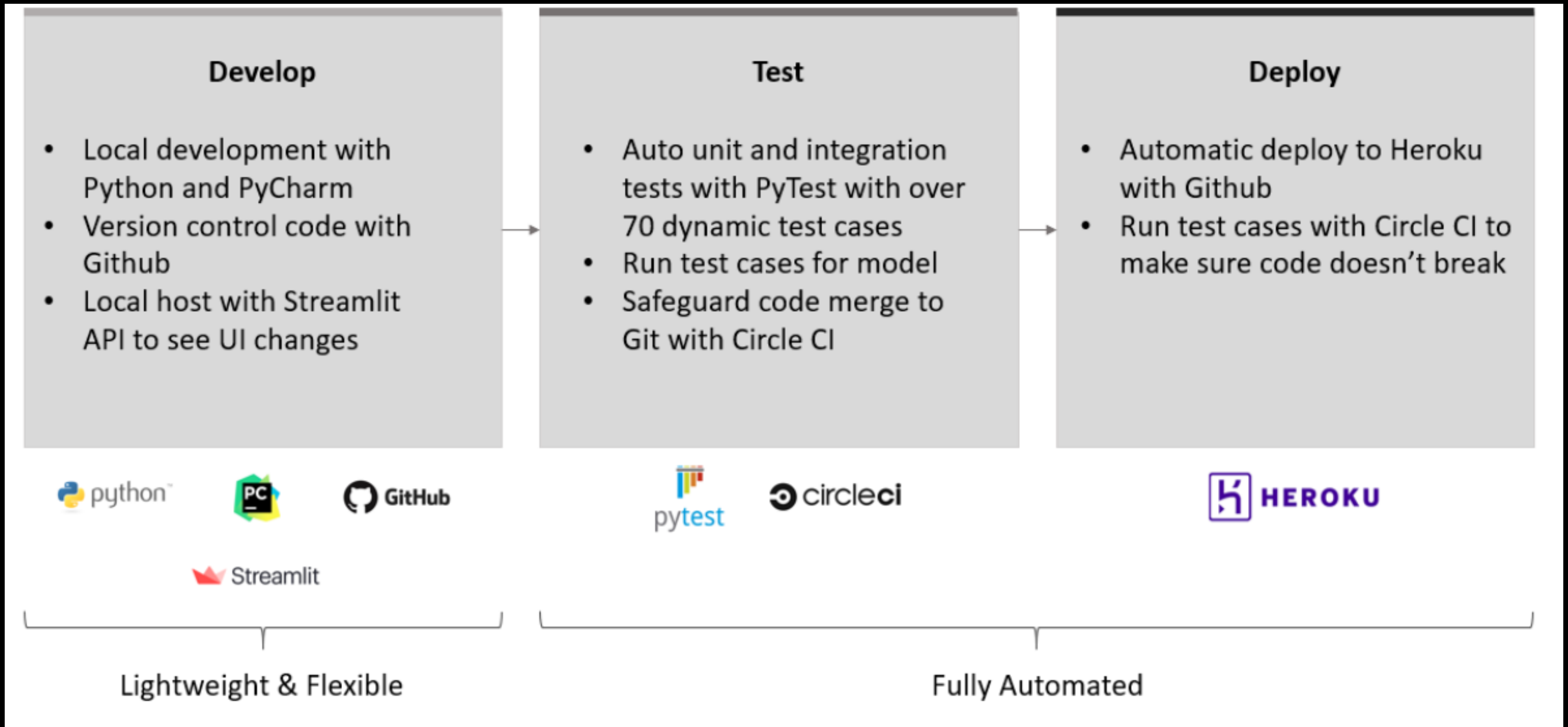
Example Workflow: Level I



https://towardsdatascience.com/build-and-deploy-your-first-machine-learning-web-app-e020db344a99

- Pycaret: Trains ML model and pickles saved model
- HTML: Website the user interacts with (in this case a simple web form)
- Flask: Backend web framework which powers the REST API and HTML front-end
- GitHub: Where all the code necessary to create and deploy the model lives
- Heroku: PAAS (Platform as a Service) used to quickly develop and host apps

Example Workflow: Level II



https://www.kdnuggets.com/2020/02/machine-learning-challenge-build-deploy-app-streamlit-devops.html

- Unit testing: ensures that an individual module behaves as expected
- Integration testing: ensure that a collection of modules interoperate as expected
- Circle CI: gate check code merge on GitHub as well as deployment of ML models to production

Stats AI

Docker & Kubernetes

- Container-based applications can be moved easily from on-prem systems to cloud environments or from developers’ laptops to servers

- Dockerfile
  - a text file with the instructions to build a Docker image
  - specifies the OS, the programming languages, environmental variables, file locations, network ports, and other components the application needs

- Docker Image:
  - is a portable file containing the actual files. Thus, it is usually a large file
  - can be stacked. i.e., you can take an image someone else created and build upon it

- Docker Container: an instance of the docker image, i.e., the program that you want to run

- Container Registry: DockerHub, Azure Container Registry Amazon ECR, Google Container Registry
  - Where we upload Docker images we create and download docker images created by others

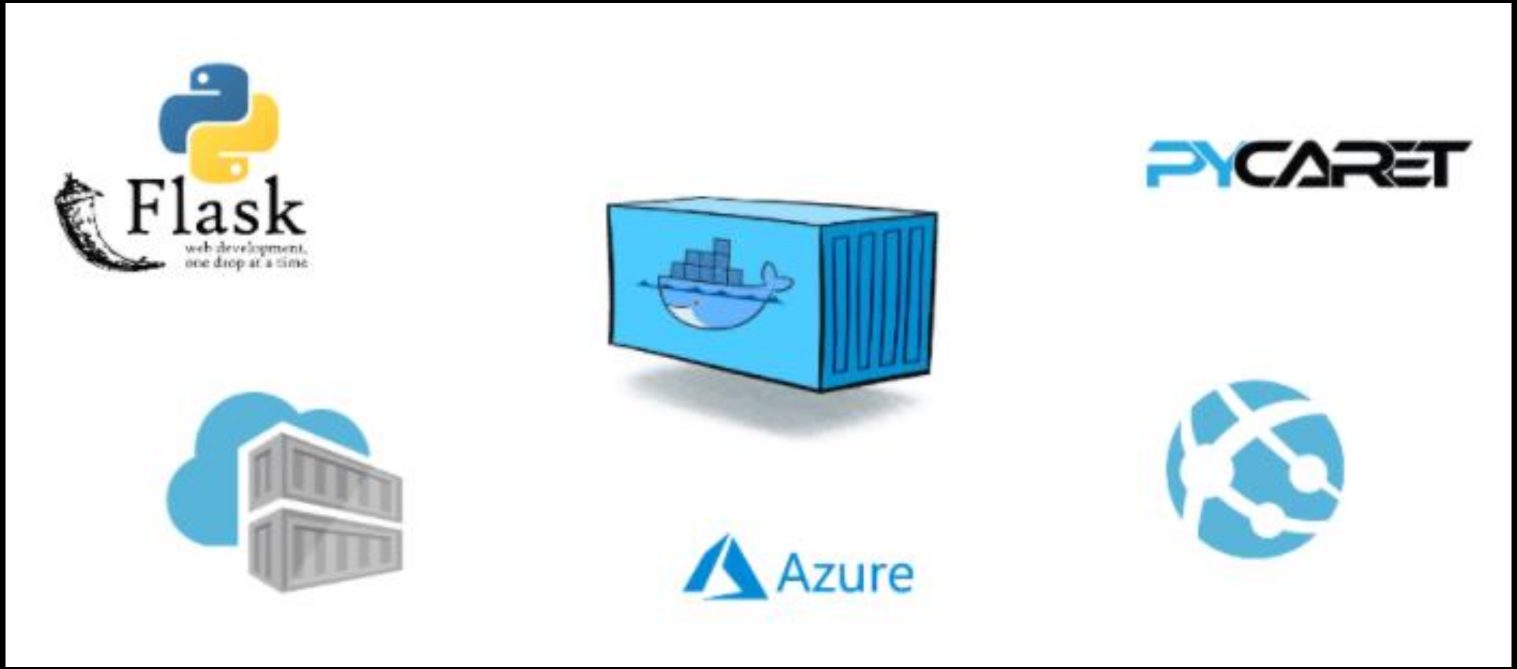
- Kubernetes: Enables container orchestration, i.e, the ability for containers to talk to each other

- We can create multiple containers from a single image on a single host OS
- Kubernetes is how we enable those containers to talk to each other across multiple host OS
- Thus, scaling up an application simply becomes spinning up more machines in a Kubernetes cluster

Stats AI

Example Workflow: Level III

Containerize and Deploy Machine Learning Pipeline as an Azure Web App



https://towardsdatascience.com/deploy-machine-learning-pipeline-on-cloud-using-docker-container-bec64458dc01

- Docker Image: a file consisting of all the dependencies needed to run our application
- Amazon Container Registry (ACR): Where we register the Docker Image we created
- Azure Web Apps (PAAS): Runs the Docker Container that we instantiated from the Docker Image

Stats AI

Example Workflow: Level III

Containerize and Deploy Machine Learning Pipeline on Google Kubernetes Engine

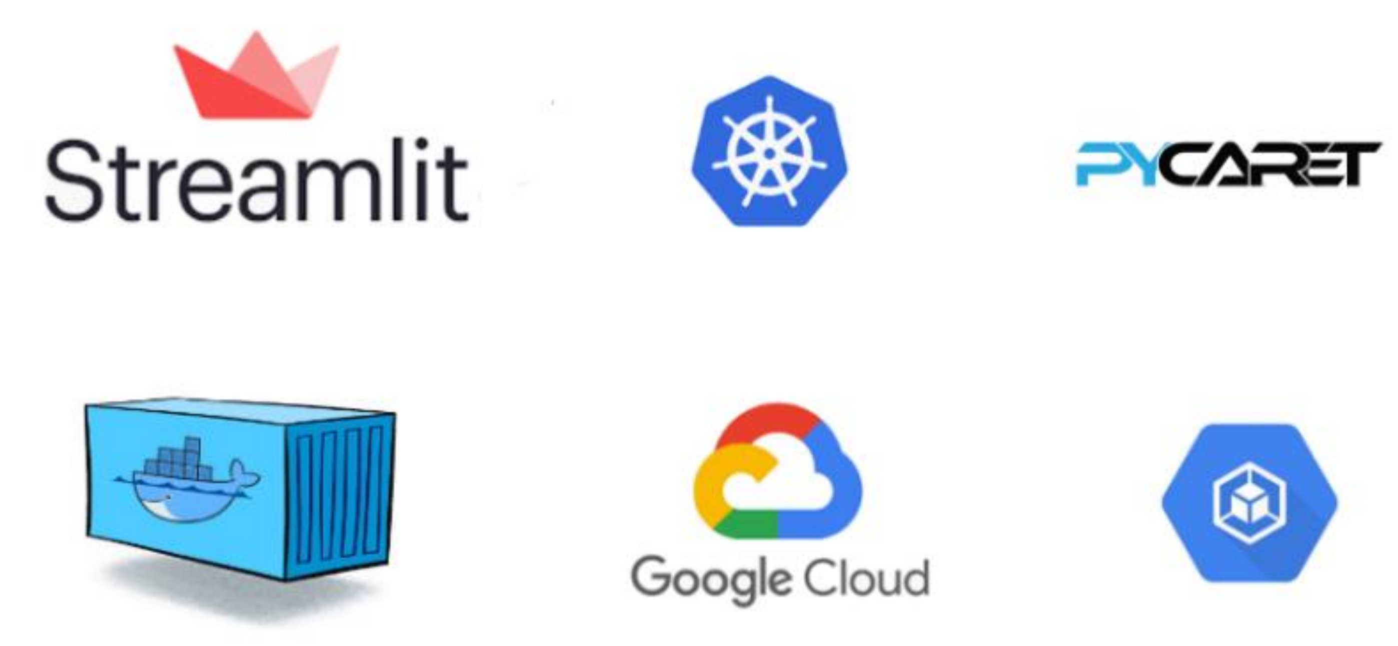
https://towardsdatascience.com/deploy-machine-learning-model-on-google-kubernetes-engine-94daac85108b

- Docker Image: a file consisting of all the dependencies needed to run our application
- Google Container Registry (ACR): Where we register the Docker Image we created
- Google Kubernetes Engine: Runs the Docker Container across a cluster which can autoscale

Stats AI

Example Workflow: Level III

Containerize and Deploy a Streamlit app on Google Kubernetes Engine



https://towardsdatascience.com/deploy-machine-learning-app-built-using-streamlit-and-pycaret-on-google-kubernetes-engine-fd7e393d99cb

- Note: Before we used Flask + HTML to deploy our GUI. Here instead Streamlit is used to do the same thing.
- Docker Image: a file consisting of all the dependencies needed to run our application
- Google Container Registry (ACR): Where we register the Docker Image we created
- Google Kubernetes Engine: Runs the Docker Container across a cluster which can autoscale

Example Workflow: Level III

Containerize and Deploy a Streamlit app on AWS Fargate

https://towardsdatascience.com/deploy-pycaret-and-streamlit-app-using-aws-fargate-serverless-infrastructure-8b7d7c0584c2

- AWS Fargate/Google Cloud Run are serverless PAAS (benefits of development w/o resource management)
- Docker Image: a file consisting of all the dependencies needed to run our application
- AWS Elastic Container Registry (EACR): Where we register the Docker Image we created
- AWS Fargate: Auto manages the resources on ECS or EKS to scale the Docker Container

The Rise of Feature Stores

Stats AI

Feature Stores

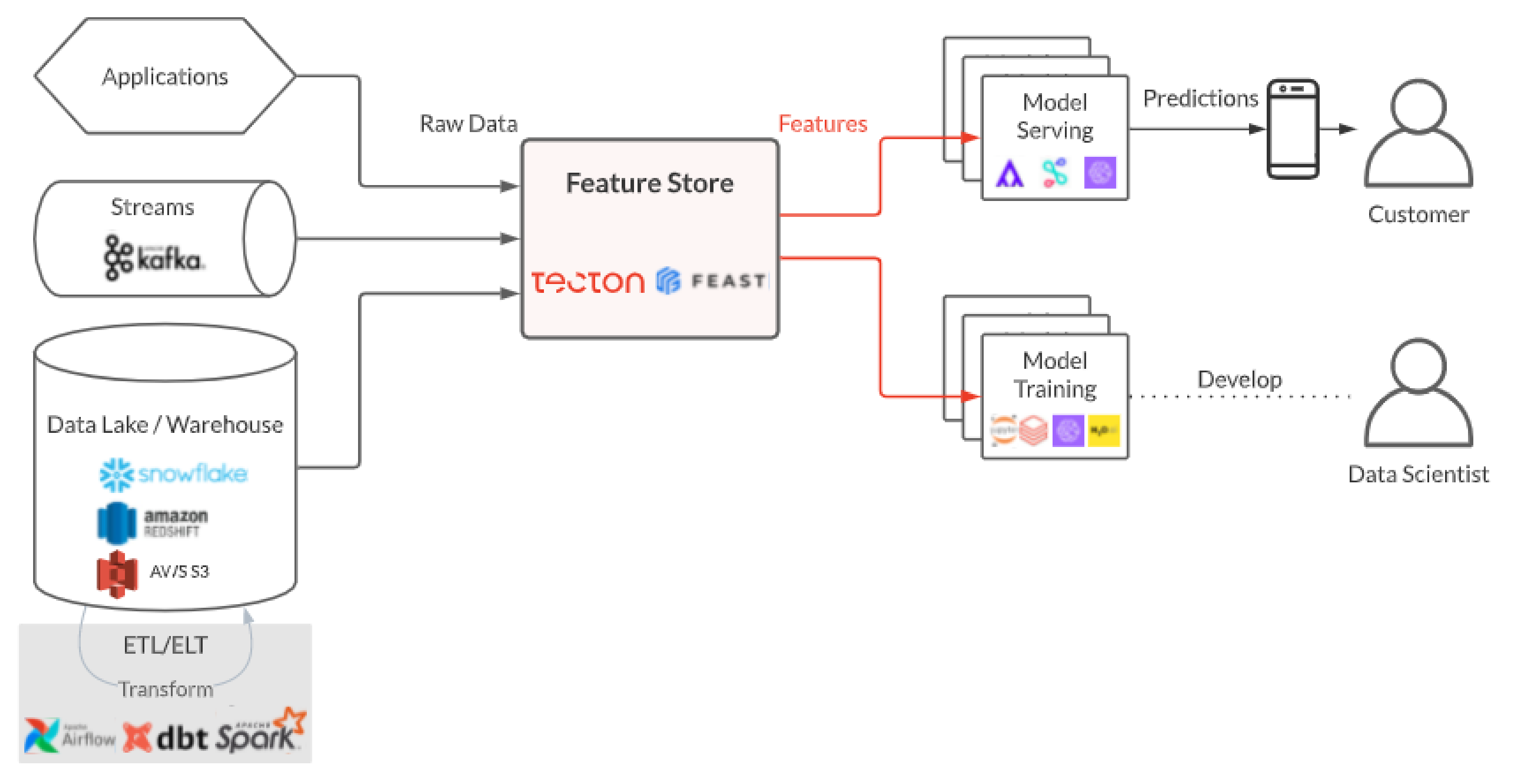
- Are basically databases for features

- By tightly coupling feature generation to data generating process, the Feature Store eliminates the possibility of finding features found in R&D that aren't possible to implement in production

- Eliminates Data Leakage:
  - Data is uploaded nightly.
  - The model developed in R&D finds that intraday data is best.
  - The problem is that that intraday data only exists after the fact.
  - Thus, only the prior night's data would be available for the production system
  - Thus, the feature store enables the data scientist developing the R&D system to "know" that intraday data would not be available at the time of inference

- Bottom line: The Feature stores value prop is its ability to serve as a single source of truth for development of both ML models and inference on fresh input values

Stats AI


Applications
Raw Data
Streams
kafka
Data Lake / Warehouse
snowflake
amazon REDSHIFT
AWS S3
ETL/ELT
Transform
Airflow
dbt
Spark
Feature Store
tecton
FEAST
Features
Model Serving
Predictions
Customer
Model Training
Develop
Data Scientist

Stats AI

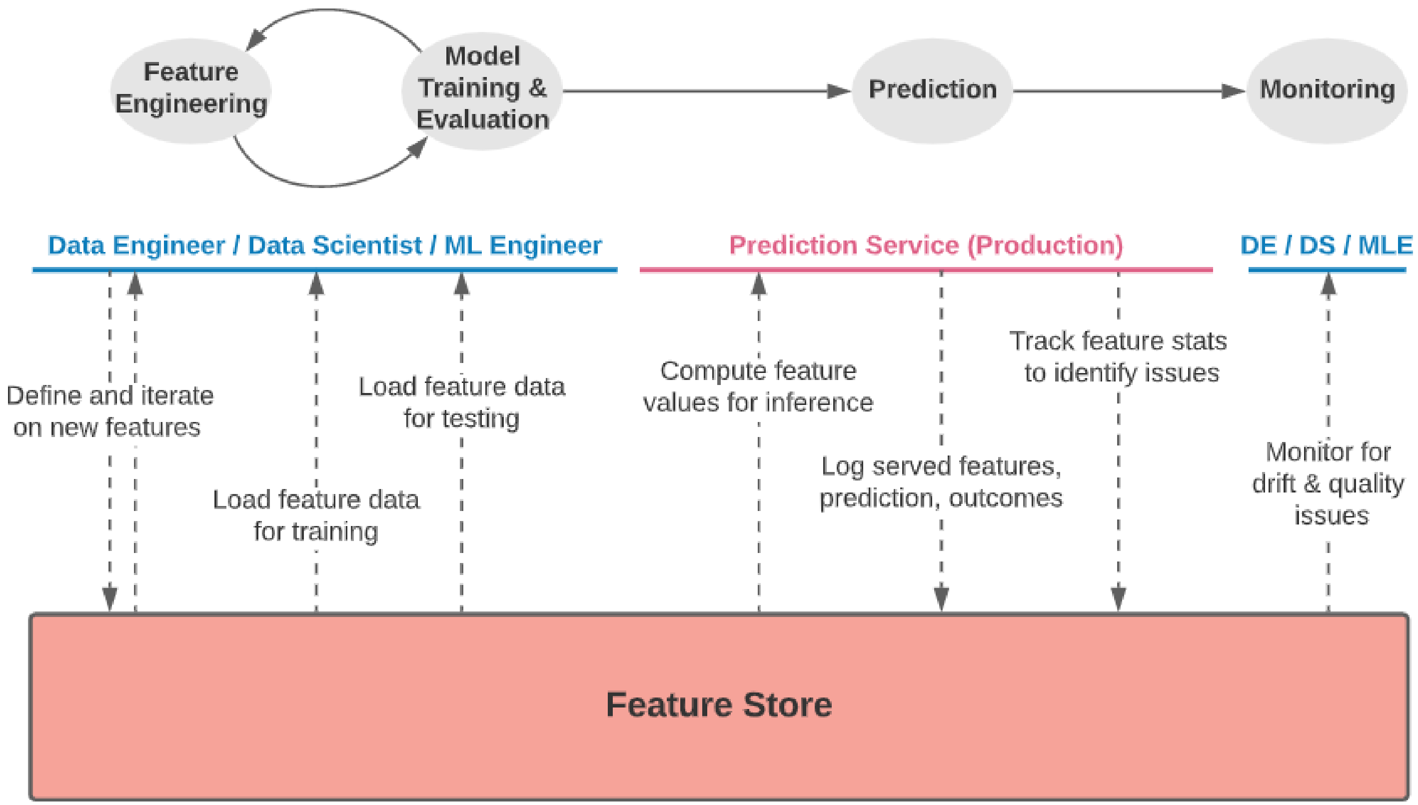
Feature Stores

- Introduce economies of scale:
    - Features discovered for one model are often useful in other models
    - We can group features by the tasks they are used in
        - regression tasks, classification tasks, anomaly detection tasks
    - When a newly created feature is registered in a feature store, it becomes available for immediate reuse by every other model across the organization
    - This reduces duplication of data engineering efforts and allows new ML projects to bootstrap with a library of curated production-ready features

Stats AI

A feature store is an ML-specific data system that:

- Runs data pipelines that **transform** raw data into feature values
- **Stores** and manages the feature data itself, and
- **Serves** feature data consistently for training and inference purposes



Stats AI