Machine Learning Engineering With Python

Why I Wrote A Book When I Could Have Relaxed In The Evenings …

Andy McMahon
Speaking at Glasgow Caledonian University
10/3/2022

Some (Welcome) Stressors in Life …

Some (Welcome) Stressors in Life …

Having your first child (September 2020)...

Some (Welcome) Stressors in Life …

Having your first child (September 2020)...

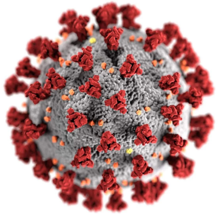
…Changing job (Jan 2021)

Some (Welcome) Stressors in Life …

Having your first child (September 2020)...

NatWest Group

…Changing job (Jan 2021)



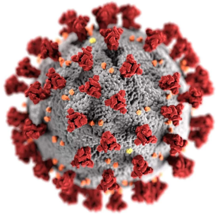
Living through a pandemic that shut down the world …. (all of above)

Some (Welcome) Stressors in Life …

Having your first child (September 2020)...

NatWest Group

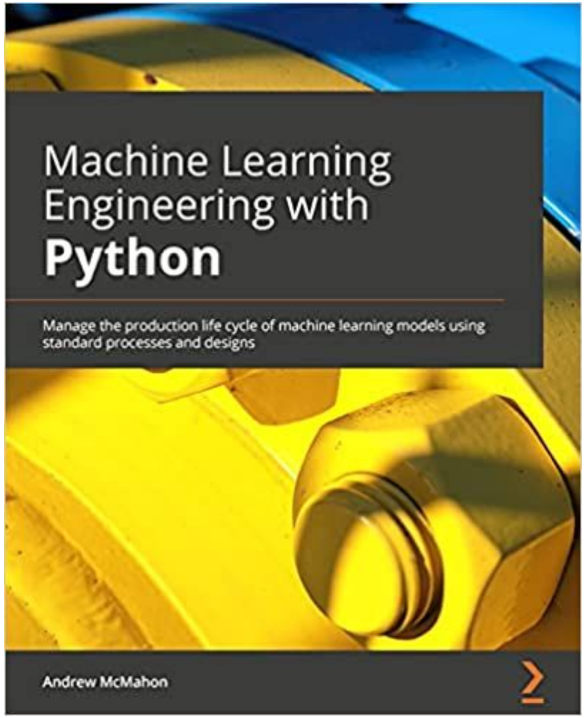
…Changing job (Jan 2021)



Living through a pandemic that shut down the world …. (all of above)

Writing a book to a tight deadline!!!! (Jan-Aug 2021)

The Book



Customer reviews
4.7 out of 5

Customer reviews
4.6 out of 5

Vishrut
Very concise read
Reviewed in the United States on January 17, 2022

Amazon Customer
A Practical Guide For Anyone Who Wants to Know How to Deploy ML models in Production
Reviewed in the United Kingdom on 6 December 2021

Amazon Customer
Covers important topics in machine learning engineering
Reviewed in the United States on January 14, 2022

Sheetal Kadam
Great book as guide for implementing ML
Reviewed in the United States on December 26, 2021

Malik K.
Love the organized breakdown of the varying data-roles and how they relate to each other
Reviewed in the United States on January 15, 2022

St. Michael
A Great MLE Book!!!
Reviewed in the United States on December 5, 2021

Vinoth
Excellent guide to starting machine learning
Reviewed in the United States on February 28, 2022

Discount Code: 25andy

Why?

ML is Big Business

ML is Big Business

STAMFORD, Conn., November 22, 2021

Gartner Forecasts Worldwide Artificial Intelligence Software Market to Reach $62 Billion in 2022

Market Growth Will Accelerate as Organizations Progress Their AI Maturity

ML is Big Business

The state of AI in 2020

November 17, 2020 | Survey

66% of businesses gained higher revenue due to their AI systems (in 2020)

Sizing the prize

PwC's Global Artificial Intelligence Study: Exploiting the AI Revolution

What's the real value of AI for your business and how can you capitalise?

$15.7tr

Potential contribution to the global economy by 2030 from AI

+26%

Up to 26% boost in GDP for local economies from AI by 2030

STAMFORD, Conn., November 22, 2021

Gartner Forecasts Worldwide Artificial Intelligence Software Market to Reach $62 Billion in 2022

Market Growth Will Accelerate as Organizations Progress Their AI Maturity

*The AI applications modelled will also create **18.4 – 38.2 million net jobs globally** (broadly equivalent to the number of people currently employed in the UK).*

How AI can enable a Sustainable Future

Microsoft in association with pwc

But There Is A Problem …

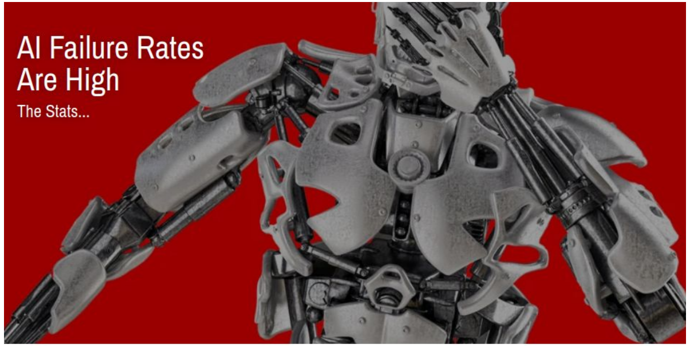
… A Big Problem

Why Big Data Science & Data Analytics Projects Fail

POSTED FEBRUARY 13, 2021 NICK HOTZ

Sponsored

Why do 87% of data science projects never make it into production?



A Survey of AI Project Failure Rates

When you think how much time, effort and money has gone into introducing artificial intelligence, how many of these projects have failed? Back in late 2017, Gartner analyst Nick Heudecker estimated

OPINION

Why 90 percent of all machine learning models never make it into production

Companies are lacking leadership support, effective communication between teams, and accessible data

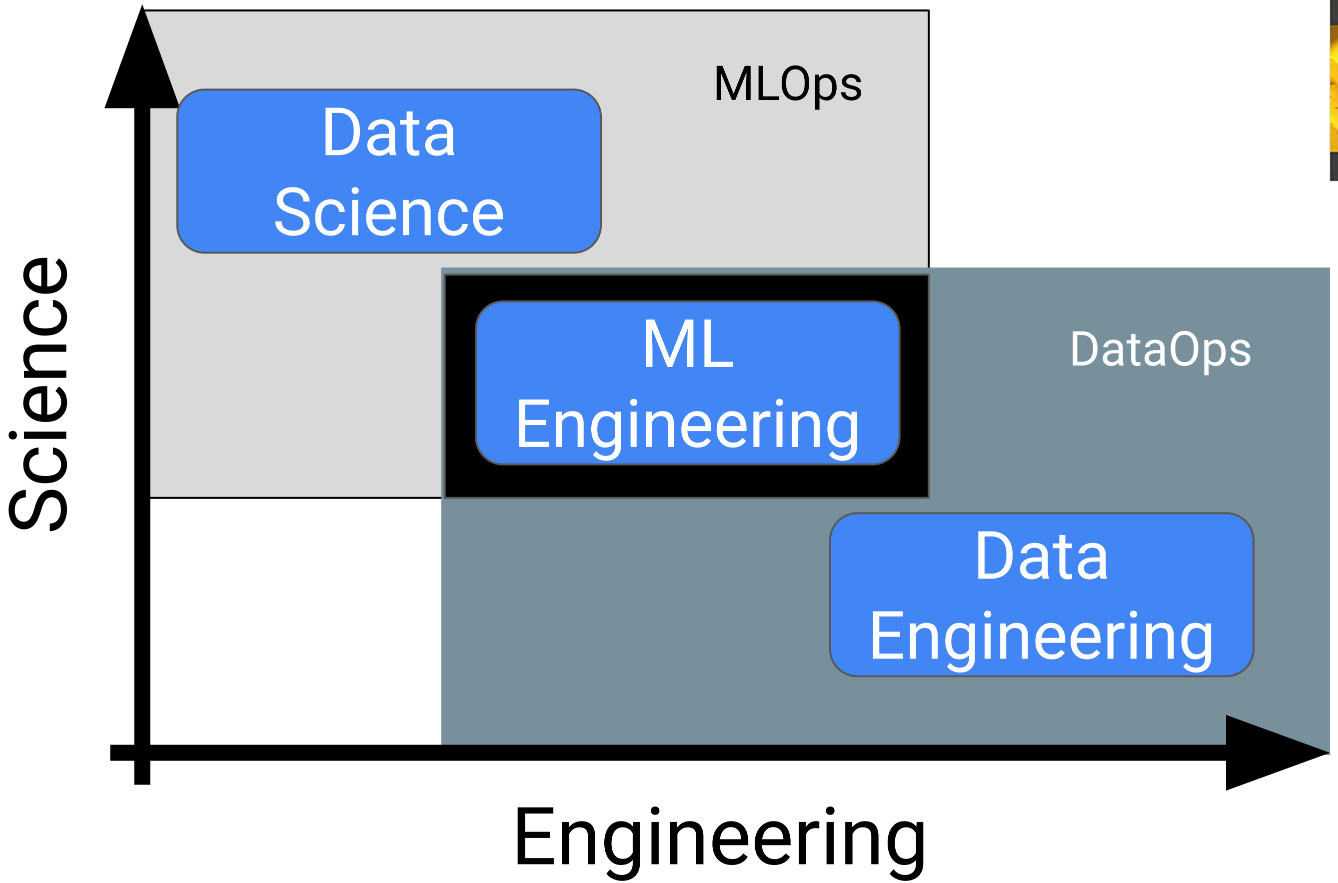
Ari Joury Nov 8, 2020 · 8 min read ★

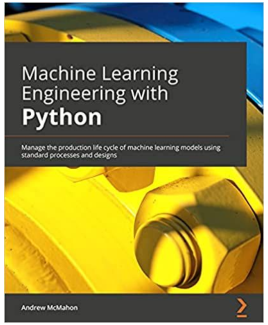
So … What Do We Do?

“You do not rise to the level of your goals.
You fall to the level of your systems.”

James Clear.

Systems → All The Ops



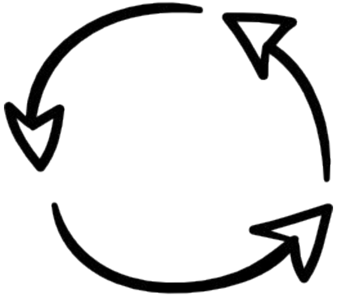


Chapter 1, Page 7

But What is ML Ops and What is ML Engineering?

MLOps and MLEng

ML Eng: Getting your data science models into production by applying enough software engineering.



MLOps: The end-to-end lifecycle management of models and solutions with ML in them.

Turning The Tide

Deloitte. Services Industries Insights Careers

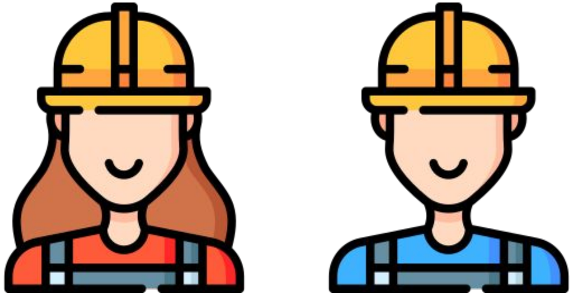
Machine Learning Operations - is not just about Artificial Intelligence it is about changing the way you do business

"Organizations that document and enforce MLOps processes are **twice as likely to achieve their AI goals.** They're also **nearly twice as likely to report being extremely prepared for risks associated with AI."** *Tech Trends 2021, Deloitte*

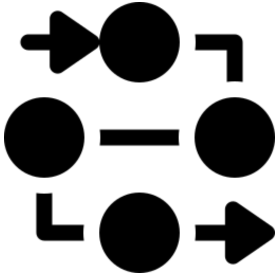
"Those that make significant investments in change management are **1.6 times more likely to report that AI initiatives exceed expectations** and **over 1.5 times more likely to achieve their desired goals."** *State of AI - Fourth Edition, Deloitte, Oct 2021*

As Leaders - How Do We Build This Capability?

The 4 P's



People



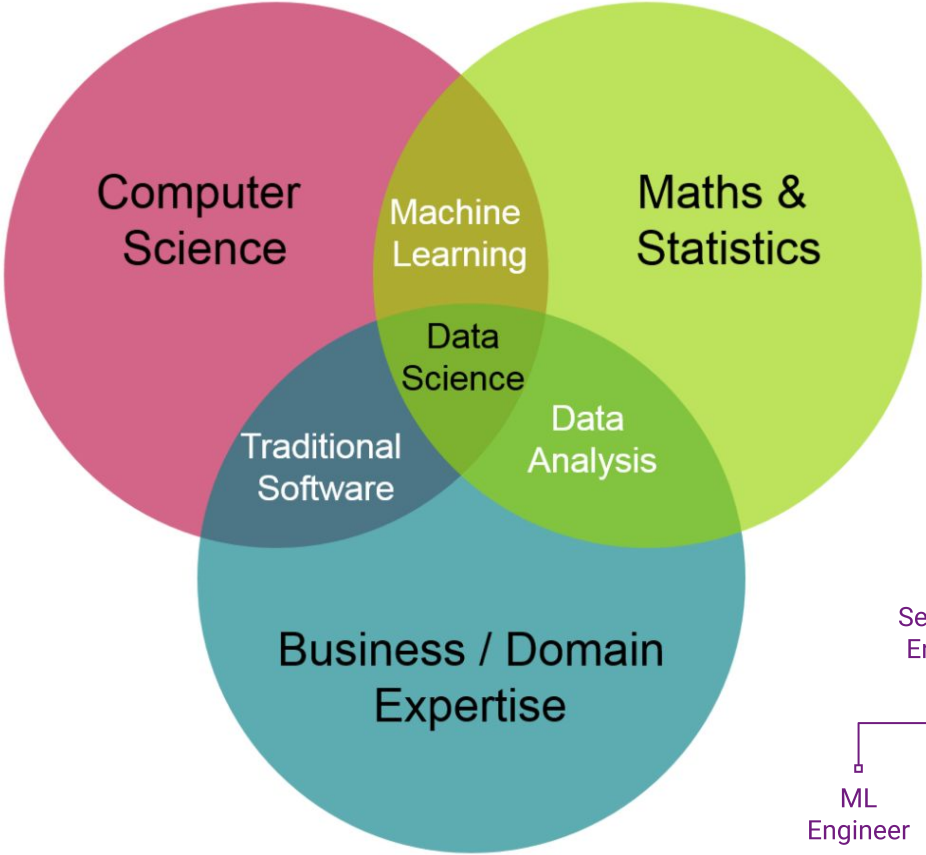
Process

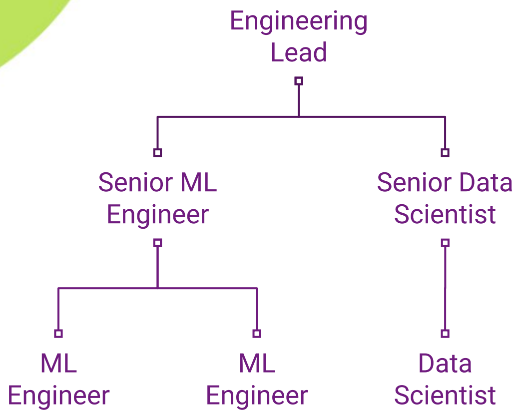
Product

Pattern

People: Blended Teams > Unicorns

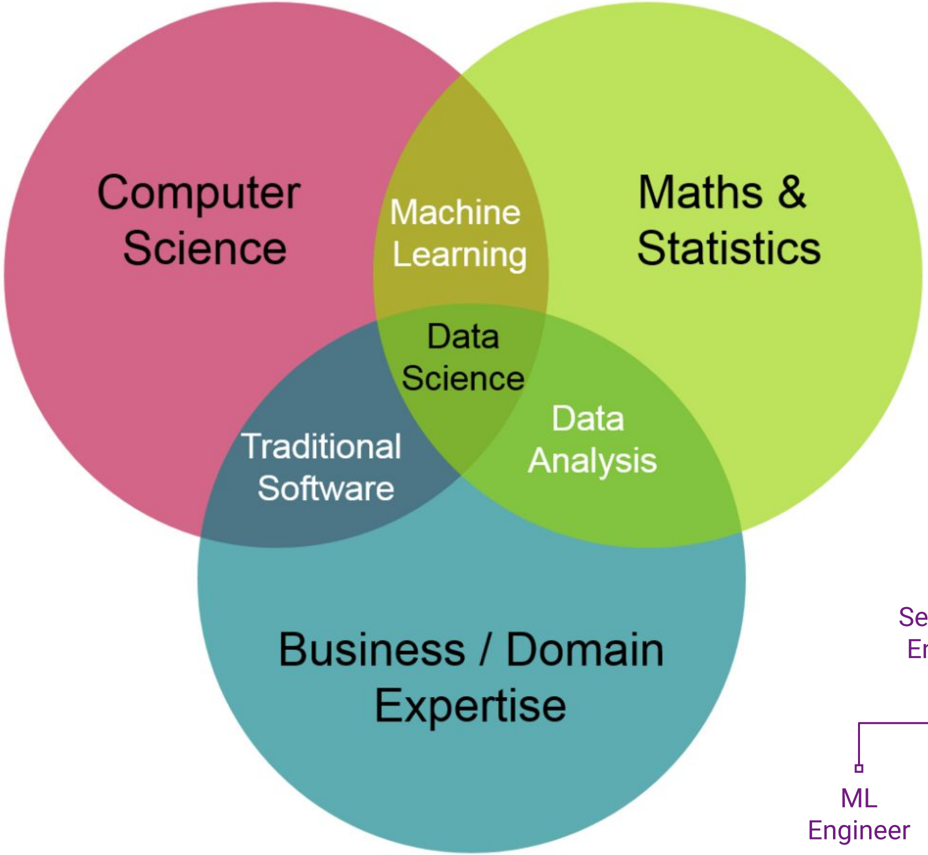
vs

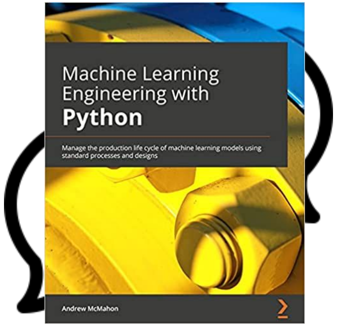




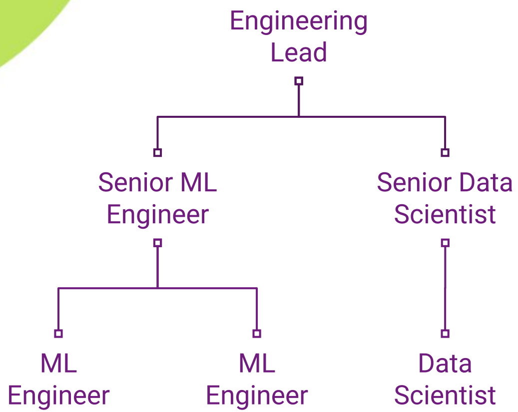
People: Blended Teams > Unicorns

vs





Chapter 1, Page 8

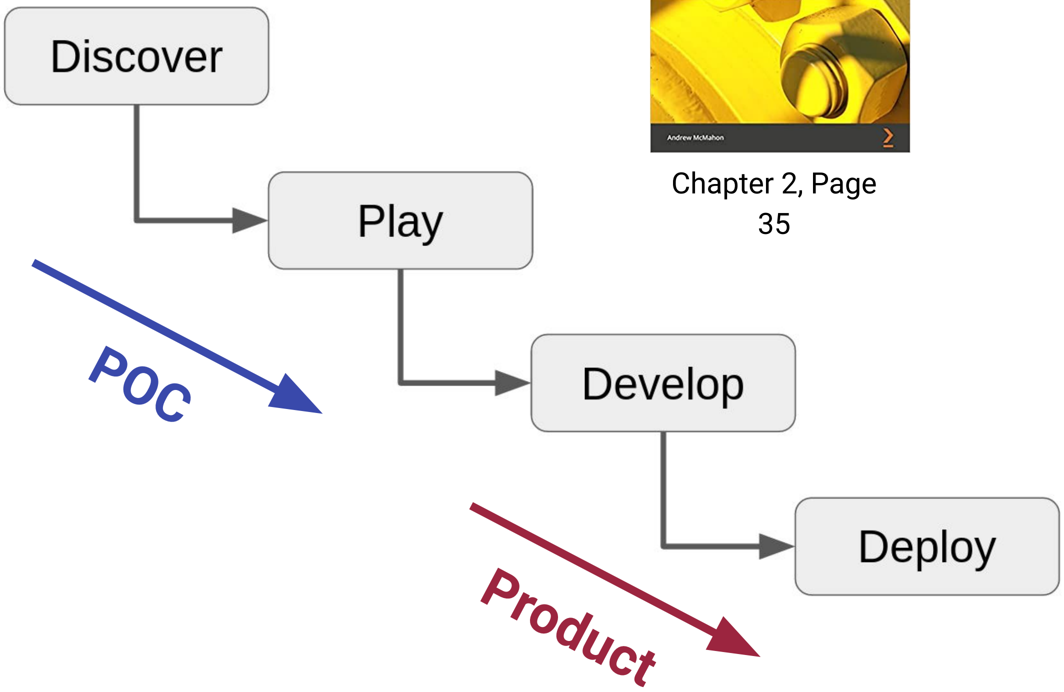


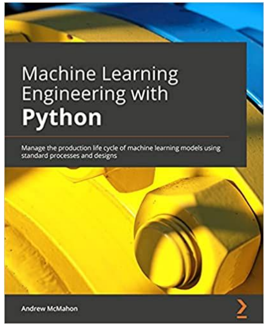
Process: Guide Don't Prescribe

- Important to provide high level steps to conceptualise progress

- These steps should help capture what's **<u>important</u>** in a project

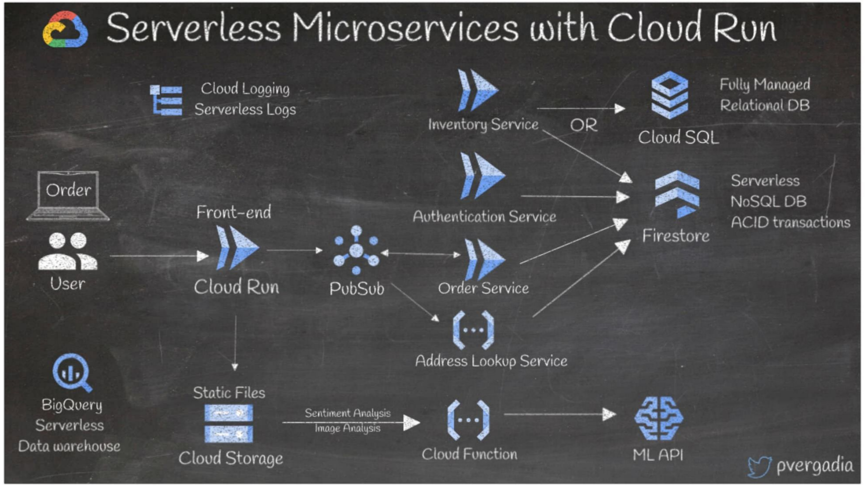
- We should still leave room for flexibility, whilst also encouraging consistency

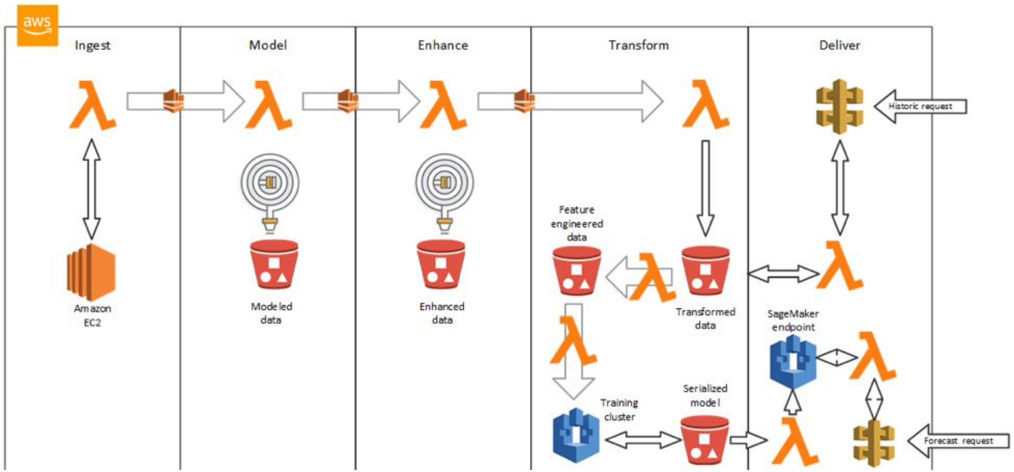




Chapter 2, Page 35

Patterns: Reuse, Repeat, Recycle

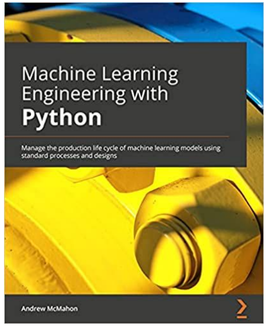




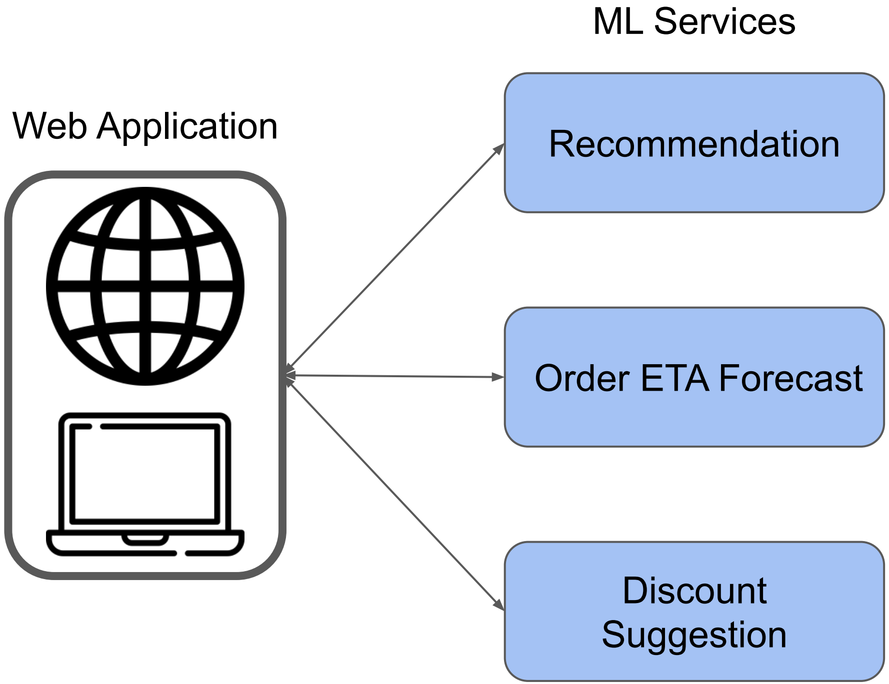
13 sample architectures to kickstart your Google Cloud journey

Machine Learning Lens
AWS Well - Architected Framework

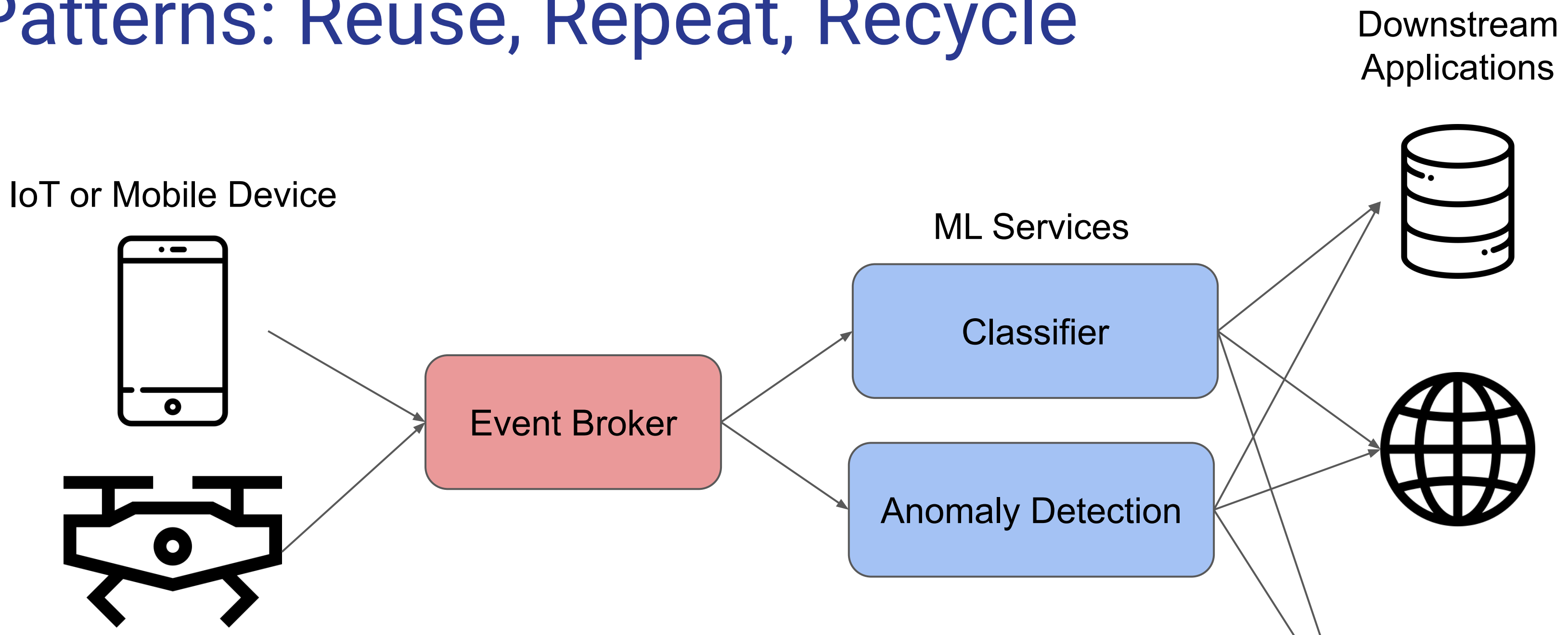
Patterns: Reuse, Repeat, Recycle

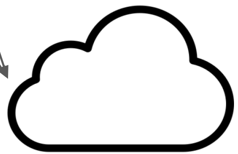


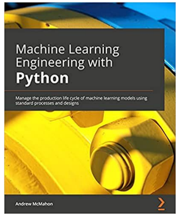
Chapter 5, Page 146



Patterns: Reuse, Repeat, Recycle

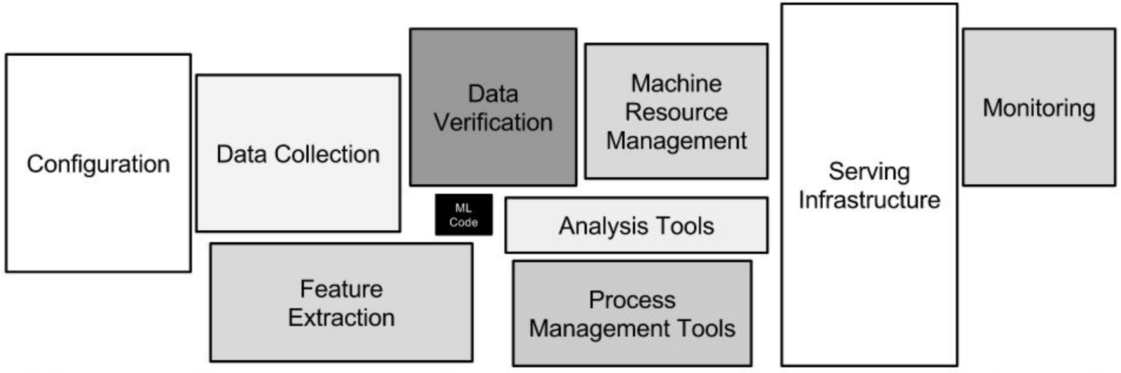






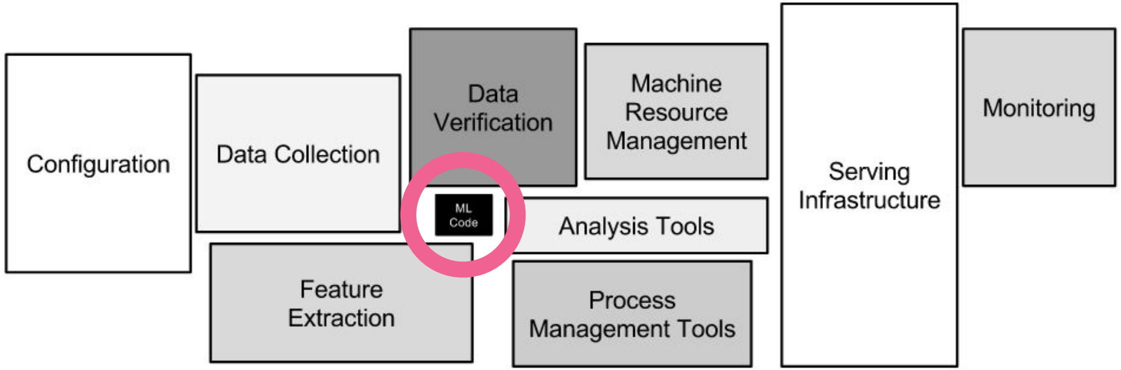
Chapter 5, Page 147

Products != Models != Analyses



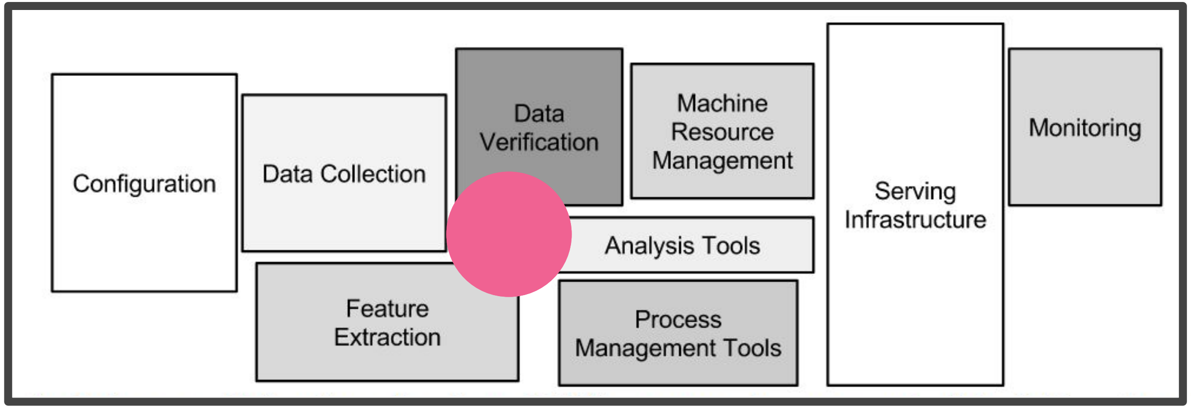
[Hidden Technical Debt in Machine Learning Systems, Google]

Products != Models != Analyses



[Hidden Technical Debt in Machine Learning Systems, Google]

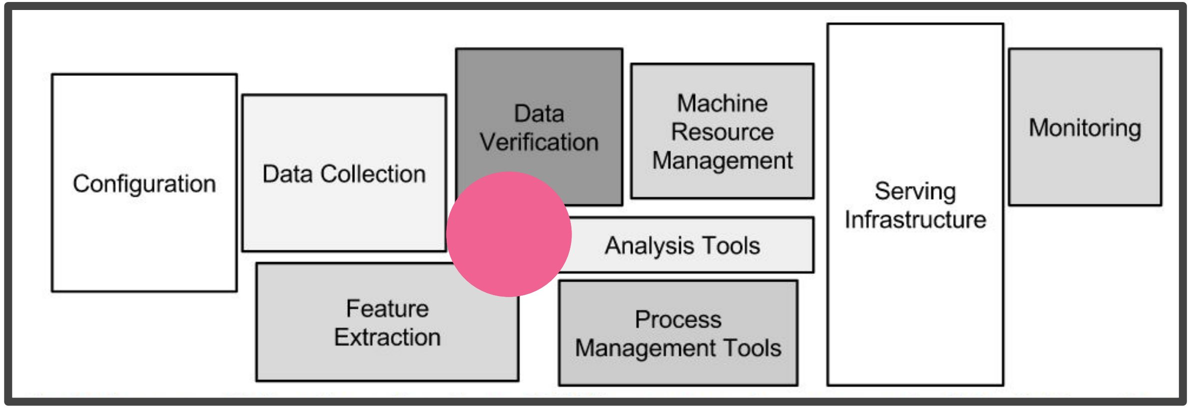
Products != Models != Analyses



MLEng/MLOps

[Hidden Technical Debt in Machine Learning Systems, Google]

Products != Models != Analyses



MLEng/MLOps

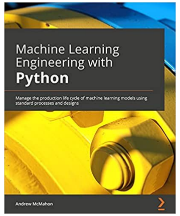
Hidden Technical Debt in Machine Learning Systems, Google

Products work: Test, test, test …

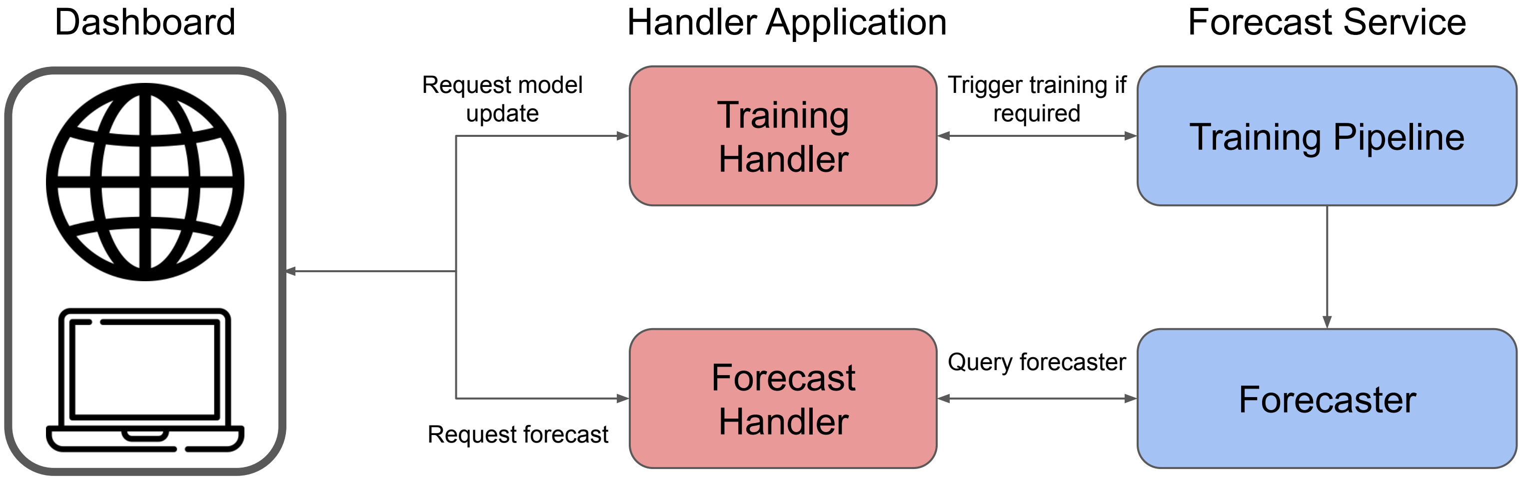
Products get delivered: Don't reinvent the wheel …

Products come in a variety of shapes: Build for flexibility ...

Products != Models != Analyses



Chapter 7, Page 221



Wrapping Up

What Was I Trying To Say?

- We all know that the opportunities provided by machine learning and data products are **enormous.**

- Those opportunities are **not being realized**.

- The winners of the ML age will be the people/teams/organisations that can successfully go from idea to production.

- This is the domain of ML engineering and MLOps.

- It’s clear what we need to be able to do - we just need to be brave enough to go after it within our teams/organisations/economy!

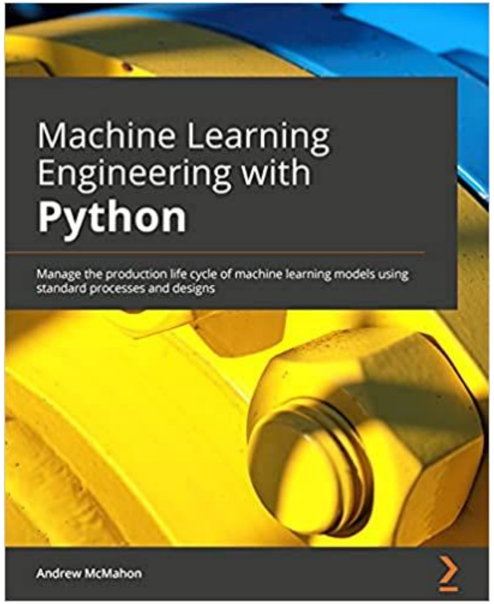
Thank you!

Some places you can find me (say hello!)

@electricweegie

AndyMc629

Andy McMahon

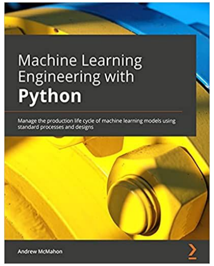


electricweegie.com

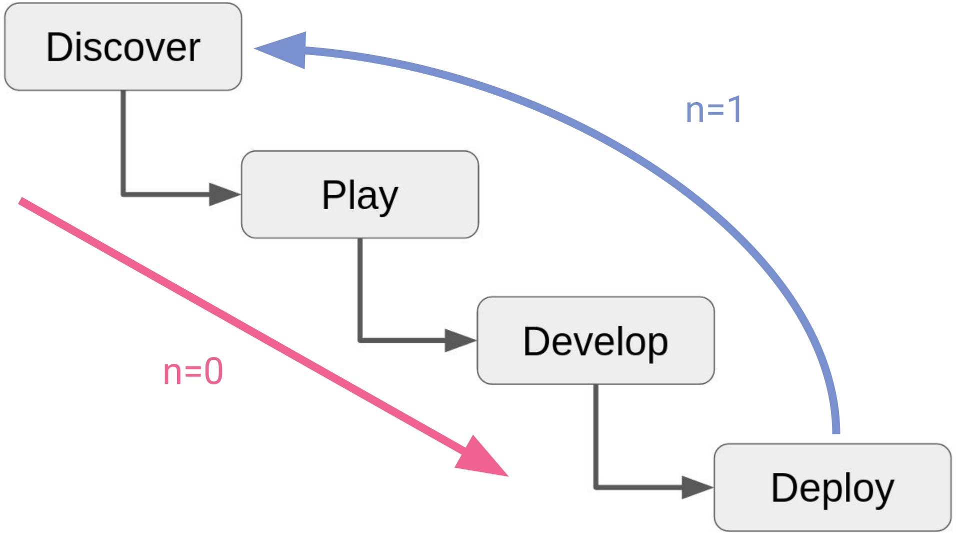
https://shows.acast.com/ai-right

Appendix

MLOps as Induction



1. Get your first model into production (n=0). **This is how you learn ML Engineering.**
2. Get the (n+1)th model into production given that the nth model is in production. **This is how you learn MLOps.**



Some Preamble

- My talk title is slightly tongue-in-cheek
- Objective: Let go of the fear of getting started
- Some phrases that can capture my sentiment:
  - Have a bias for action/execution is key
  - Bootstrap your capability
- In a nutshell - attempt what you are **not ready to do** and you will learn faster!

Challenge: Analysis Paralysis

mlflow

dataiku

neptune.ai

ALGORITHMIA

fiddler

Kedro

Kubeflow

truera

ALIBI

comet

DataRobot

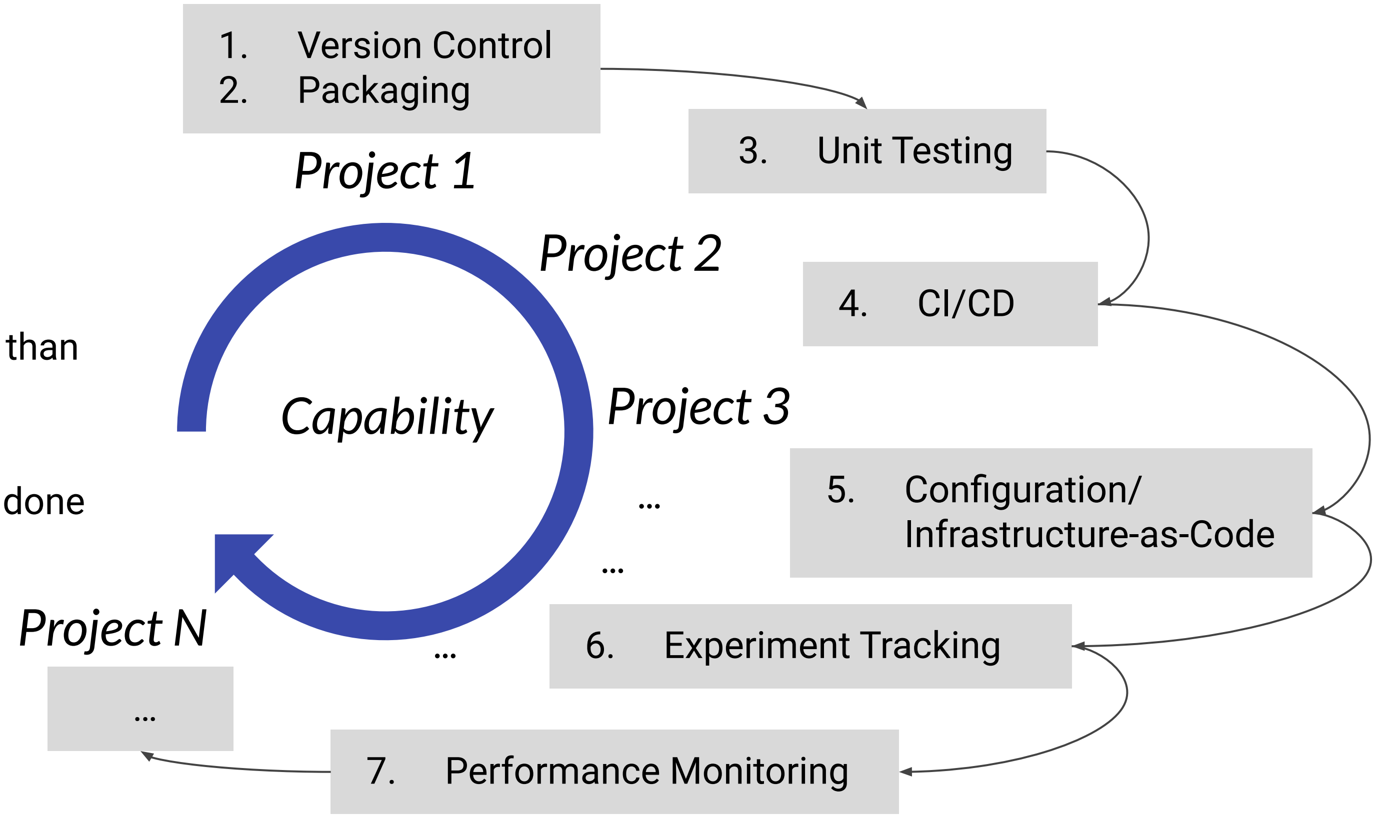
TensorBoard

Arthur

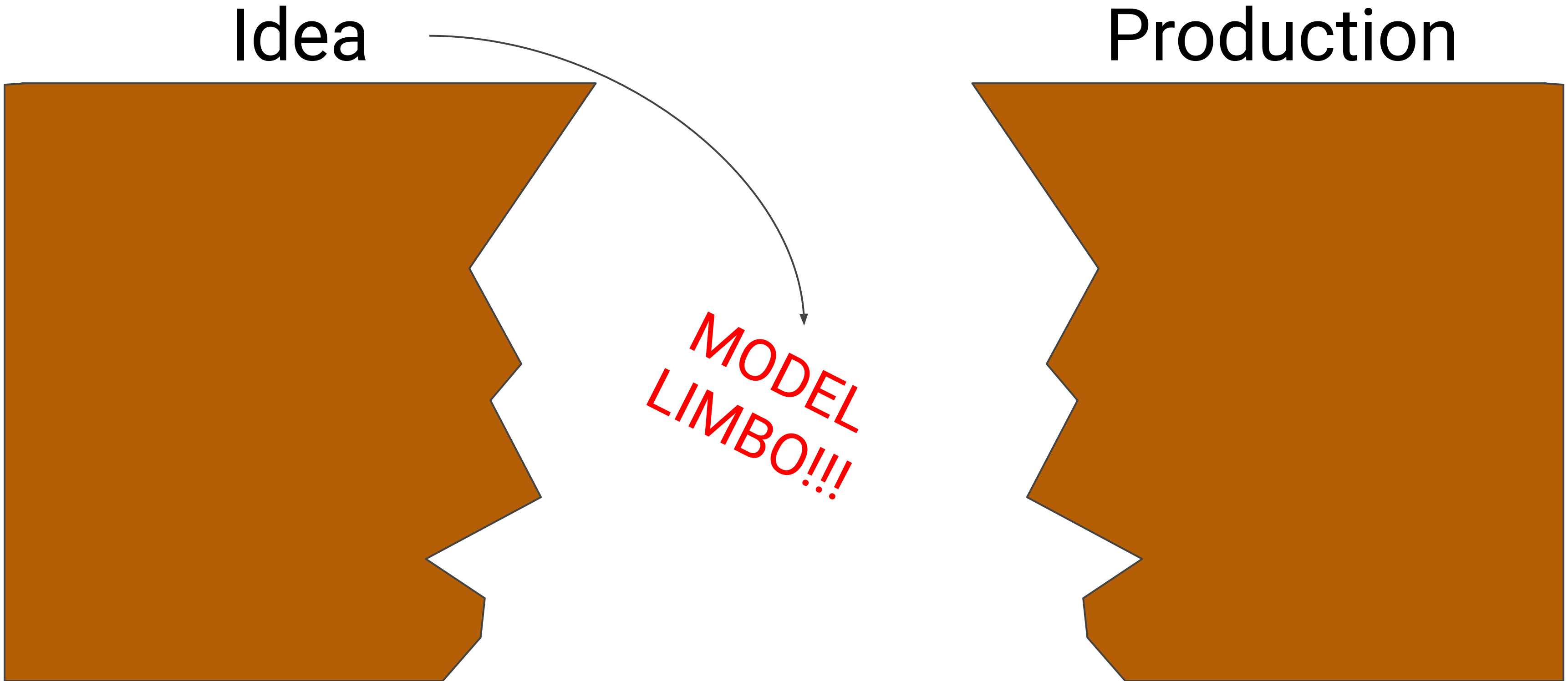
So What Do We Do?

Bootstrapping

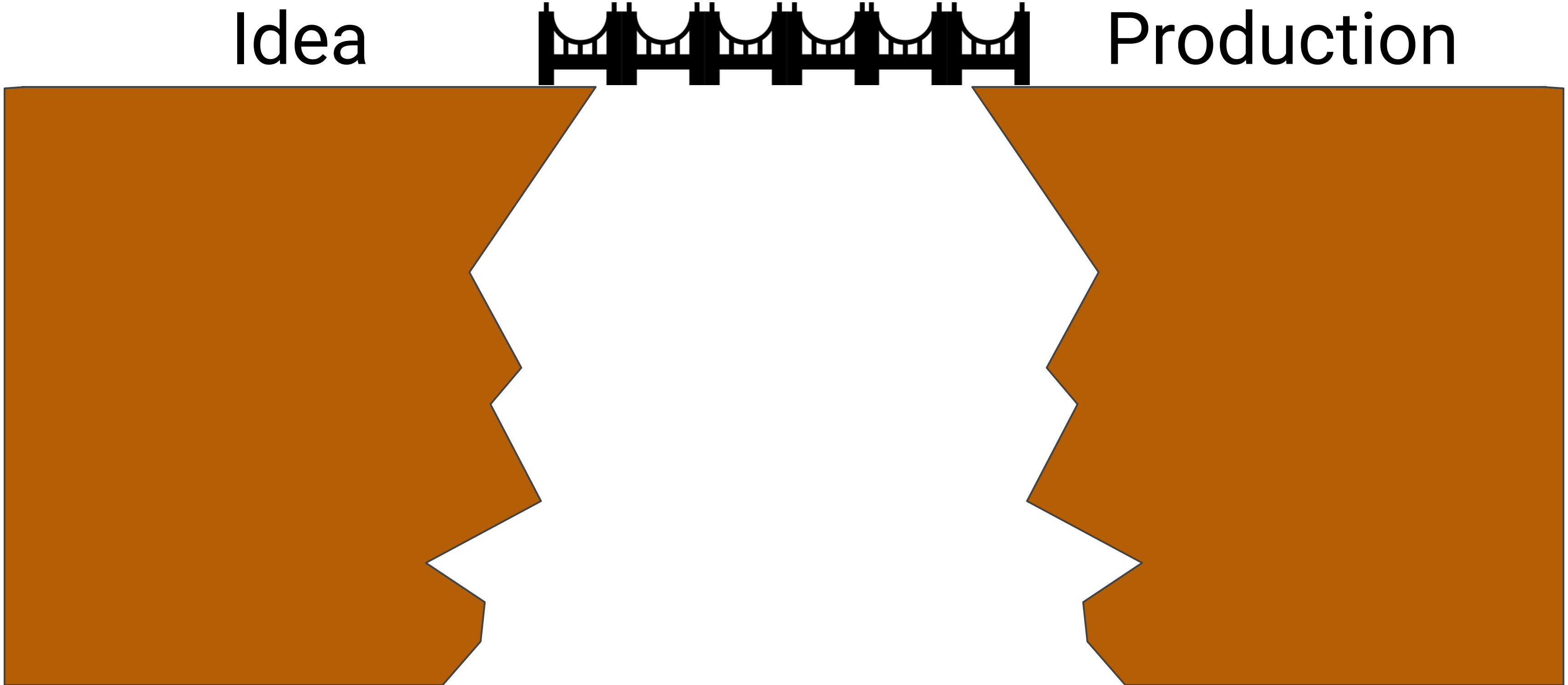
- Start small then ratchet up
- Working is better than perfect
- You will never be done



The Chasm



The Chasm



Getting The (n=0)th Model Into Production

ML Eng 101 - Production

What?

- An isolated environment with strict controls, defined risk profile, service level agreements ….
- The place where your ‘real’ inference happens
- Where request/response traffic goes

How?

- Forget about notebooks* (sorry everyone, but it’s true)
- We package, we lint, we test, we build, we version control
- We follow a clear route-to-live with appropriate promotion mechanisms

Done.

**We can fight about this one later …*

ML Eng 102 - The Code

Prep for Prophet

```
df.rename(columns= {'Datetime': 'ds', 'AEP_MW': 'y'}, inplace=True)
```

```
df['ds']=df['ds'].astype('datetime64[ns]')
```

```
df.dtypes
```

```
#Initialize Split Class, we'll split our data 5 times for cv
ts_splits = TimeSeriesSplit(n_splits=5)
```

Train and Forecast

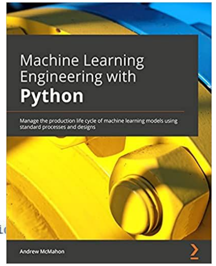
```
tmp = time_split_train_test(df.sort_values('ds', ascending=True).iloc[-1000:], ts_splits)
```

```
tmp.head()
```

Plot

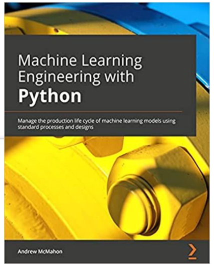
```
nrow = 5; ncol = 1;
fig, axs = plt.subplots(nrows=nrow, ncols=ncol, figsize=(20,30))
fig.subplots_adjust(hspace=0.4, wspace=0.4)
for i, ax in enumerate(fig.axes):
    split_rmse = tmp[(tmp['split']==i) & (tmp['train']==False)]['rmse'].iloc[0]

    ax.set_title('Split '+str(i)+' - RMSE: '+"{:.2f}".format(split_rmse))

    tmp[(tmp['split']==i) & (tmp['train']==True)].plot(x='ds', y='y', ax=ax, color='blue', marker='o')
    tmp[(tmp['split']==i) & (tmp['train']==False)].plot(x='ds', y='y', ax=ax, color='red', marker='o')
    tmp[(tmp['split']==i) & (tmp['train']==False)].plot(x='ds', y='yhat', ax=ax, color='orange', marker='^')
```

- OOP or Functional
- Separation of Concerns
- Keep it Simple Stupid
- Unit-test friendly



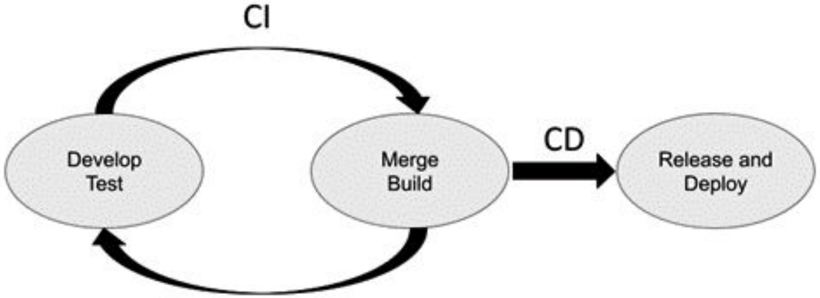

```
class Trainer(object):

    def __init__(self, forecast_session):
        self.forecast_session = forecast_session
        self.df_pred_metadata = self.get_df_pred_metadata()
        self.latest_predictor = self.get_latest_predictor()

    def get_df_pred_metadata(self):
        predictor_metadata = self.forecast_session.forecast.list_predictors()['Predi
        df_pred_metadata = pd.DataFrame.from_records(predictor_metadata)
        return df_pred_metadata

    def get_latest_predictor(self):
        latest_predictor = self.df_pred_metadata.sort_values(by='CreationTime', ascending=False).loc[0].to_dict()
        return latest_predictor

    def latest_predictor_in_tolerance(self, tolerance_days=2):
        train_time_elapsed_days = (
                datetime.datetime.now() - self.latest_predictor['CreationTime'].replace(tzinfo=None)
        ).days
        if train_time_elapsed_days < tolerance_days:
            return True
        else:
            return False

    def train_new_predictor(self):
        PREDICTOR_NAME = PREDICTOR_BASE_NAME + datetime.datetime.now().strftime(format='%Y_%m_%d_%H_%M')
        train_response = self.forecast_session.forecast.create_predictor(PredictorName=PREDICTOR_NAME,
                                                                         AlgorithmArn=ALGORITHM_ARN,
                                                                         ForecastHorizon=7,
                                                                         PerformAutoML=False,
                                                                         PerformHPO=False,
                                                                         InputDataConfig={
                                                                             "DatasetGroupArn": DATASET_GROUP_ARN},
                                                                         FeaturizationConfig={
                                                                             "ForecastFrequency": DATASET_FREQUENCY}
                                                                         )
        return train_response

    def create_latest_forecast(self):
        FORECAST_NAME = FORECAST_BASE_NAME + datetime.datetime.now().strftime(format='%Y_%m_%d_%H_%M')
        create_forecast_response = self.forecast_session.forecast.create_forecast(
            ForecastName=FORECAST_NAME,
            PredictorArn=self.latest_predictor['PredictorArn'])
        return create_forecast_response
```

ML Eng 103 - The Deployment



CI/CD Is Your Friend

- Github Actions, Jenkins, AWS CodePipeline/CodeBuild …
- Automate, automate, automate!
- Checks and balances like minimum test coverage or even data quality check passes

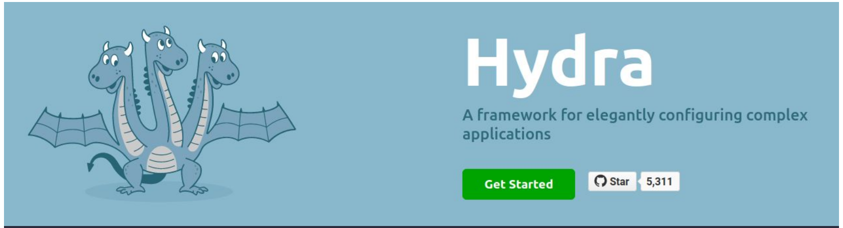



```yaml
# This is a basic workflow to help you get started with Actions

name: Upload DAGS to S3

# Controls when the action will run.
on:
  # Triggers the workflow on push or pull request events but only for the main branch
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

  # Allows you to run this workflow manually from the Actions tab
  workflow_dispatch:

jobs:
  deploy:
    name: Upload DAGS to Amazon S3
    runs-on: ubuntu-latest

    steps:
    - name: Checkout
      uses: actions/checkout@v2

    - name: Configure AWS credentials from account
      uses: aws-actions/configure-aws-credentials@v1
      with:
        aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
        aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
        aws-region: us-east-1

    - name: Copy files to bucket with the AWS CLI
      run: |
        aws s3 cp ./dags s3://github-actions-ci-cd-tests --recursive --include "*.py"
```

ML Eng 104 - The Deployment (Configuration)

Configuration Is Your Other Friend

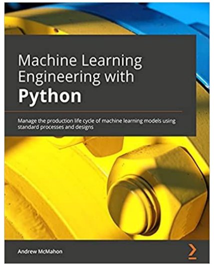
- I love YAML!
- Separate out what is instance specific and what is generic application logic/code/modelling
- Reduce complexity of deployments needed for configuration changes!



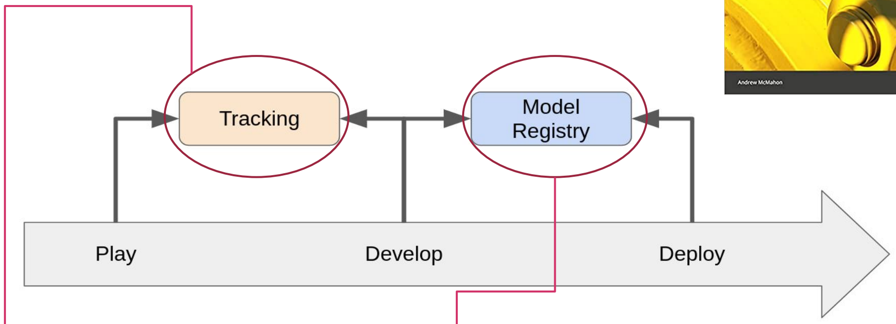

```python
@dataclass
class MySQLConfig:
    host: str = "localhost"
    port: int = 3306

@dataclass
class UserInterface:
    title: str = "My app"
    width: int = 1024
    height: int = 768

@dataclass
class MyConfig:
    db: MySQLConfig = MySQLConfig()
    ui: UserInterface = UserInterface()

cs = ConfigStore.instance()
cs.store(name="config", node=MyConfig)

@hydra.main(config_path=None, config_name="config")
def my_app(cfg: MyConfig) -> None:
    print(f"Title={cfg.ui.title}, size={cfg.ui.width}x{cfg.ui.height} pixels")

if __name__ == "__main__":
    my_app()
```

Getting The (n+1)th Model into Production

MLOps 101 - Model Management

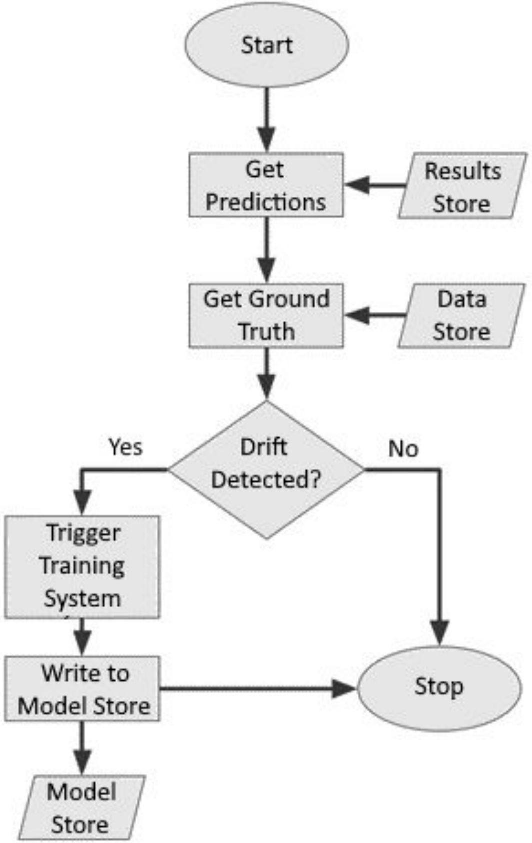



```python
with mlflow.start_run(run_name="YOUR_RUN_NAME") as run:
    params = {
        'tol': 1e-2,
        'solver': 'sag'
    }
    # Fit a ridge classifier after performing standard scaling
    std_scale_clf = make_pipeline(StandardScaler(), RidgeClassifier(**params))
    std_scale_clf.fit(X_train, y_train)
    y_pred_std_scale = std_scale_clf.predict(X_test)

    mlflow.log_metrics(
        {
            'accuracy': metrics.accuracy_score(y_test, y_pred_std_scale),
            'precision': metrics.precision_score(y_test, y_pred_std_scale, average='macro'),
            'f1': metrics.f1_score(y_test, y_pred_std_scale, average='macro'),
            'recall': metrics.recall_score(y_test, y_pred_std_scale, average='macro')
        }
    )

    mlflow.log_params(params)

    # Log the sklearn model and register as version 1
    mlflow.sklearn.log_model(
        sk_model=std_scale_clf,
        artifact_path="sklearn-model",
        registered_model_name="sk-learn-std-scale-clf"
    )
```

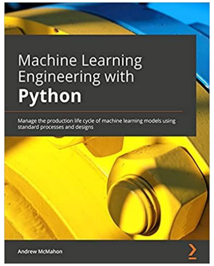



```python
# Transition the model stage to 'Staging'
client = MlflowClient()
client.transition_model_version_stage(
    name="sk-learn-std-scale-clf",
    version=1,
    stage="Staging"
)
# Transition the model stage to 'Production'
client = MlflowClient()
client.transition_model_version_stage(
    name="sk-learn-std-scale-clf",
    version=1,
    stage="Production"
)
```

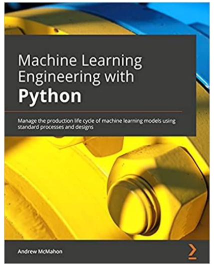
MLOps 102 - Performance Monitoring

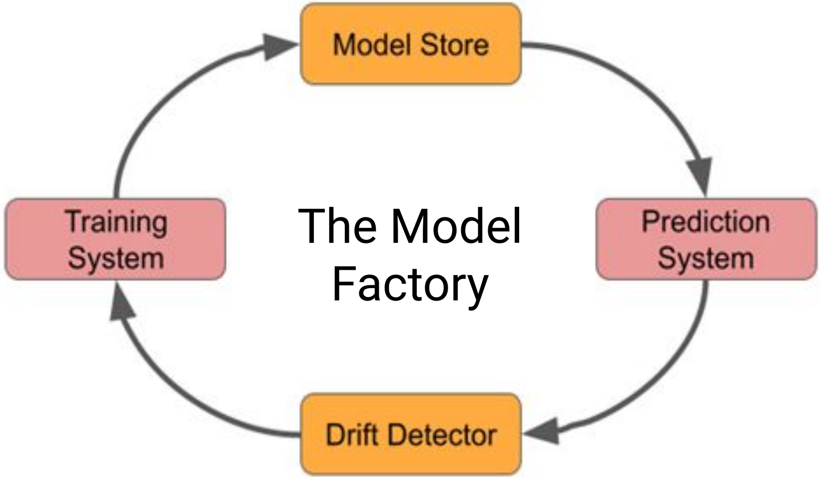



```python
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split

import alibi
from alibi_detect.cd import ChiSquareDrift, TabularDrift
from alibi_detect.utils.saving import save_detector, load_detector

# Grab the data
wine_data = load_wine()
feature_names = wine_data.feature_names
X, y = wine_data.data, wine_data.target

# Make a 50/50 reference/test split
X_ref, X_test, y_ref, y_test = train_test_split(X, y,
                                                test_size=0.50,
                                                random_state=42)

# Initialise the detector
cd = TabularDrift(p_val=.05, X_ref=X_ref)

# Check for drift
preds = cd.predict(X_test)
labels = ['No', 'Yes']
print('Drift: {}'.format(labels[preds['data']['is_drift']]))
```



MLOps 103 - Bringing it Together





The Train-Persist Process