Painless machine learning in production

H. Chase Stevens
Principal Data Science Engineer, teikametrics
Boston, MA
chase@chasestevens.com
@hchasestevens

Europython 2020
“Painless machine learning in production”
“Painless machine learning in production”
“Painless machine learning in production”

“Painless machine learning in production”

“Painless machine learning in production”
“Painless machine learning in production”

“Painless machine learning in production”

“Painless machine learning in production”
“Painless machine learning in production”

“Painless machine learning in production”

“Painless machine learning in production”
“Painless machine learning in production”

“Painless machine learning in production”

“Painless machine learning in production”
Lessons from industry regarding pain reduction and data scientist empowerment in the productionization of machine learning models
Contents

- Motivation
- Developer experience
- Our stack
- Lessons learned
Motivation

I. Ops is intrinsic to ML
Motivation

I. Ops is intrinsic to ML
II. MLOps is unsustainable
Motivation

I. Ops is intrinsic to ML
II. MLOps is unsustainable

Data scientists need to productionize their own models
Motivation

I. Ops is intrinsic to ML
II. MLOps is unsustainable

∴

Data scientists need to productionize their own models

III. Data scientists want to do data science
Motivation

I. Ops is intrinsic to ML
II. MLOps is unsustainable

∴

Data scientists need to productionize their own models

III. Data scientists want to do data science

∴

We need tooling and services to minimize “ops” overhead
I. Ops is intrinsic to ML

![Diagram showing the cycle of Deploy, Preprocess, Orchestration, Evaluate, and Train.]
I. Ops is intrinsic to ML
I. Ops is intrinsic to ML
I. Ops is intrinsic to ML

Sanders, H., & Saxe, J. (2017). Garbage in, garbage out: how purportedly great ML models can be screwed up by bad data.
I. Ops is intrinsic to ML
II. MLOps is unsustainable (in 1970)
II. MLOps is unsustainable (in 1970)

“You couldn't even delete a mistake”

“I had to wait hours for my programs to turn around”

“Only a select few programmers were allowed in the computer lab.”

“One of our finals was to design, code, punch, debug a solution - we got 4 days to do it which means finding typos, logic errors, and design errors and eliminating them all with only 4 re-runs”

“I submitted my program to the punch card crew, and got it back several days later with a rather strong note”
II. MLOps is unsustainable (in 1970)

“You couldn’t even delete a mistake”

“I had to wait hours for my programs to turn around”

“Only a select few programmers were allowed in the computer lab.”

“One of our finals was to design, code, punch, debug a solution - we got 4 days to do it which means finding typos, logic errors, and design errors and eliminating them all with only 4 re-runs”

“I submitted my program to the punch card crew, and got it back several days later with a rather strong note”
II. MLOps is unsustainable (in 2000)

Code → QA → Release (?)
II. MLOps is unsustainable (in 2000)

Code → QA → Release (?)
II. MLOps is unsustainable (in 2000)

Code → QA → Release (?)
II. MLOps is unsustainable (in 2000)

Code → QA → *Release (?)*

Diagram showing the flow from 'amazon web services' to 'heroku'.
II. MLOps is unsustainable (in 2000)

Code → QA → Release (?)
II. MLOps is unsustainable (in 2000)

Code → QA → **Release (?)**

The Rise Of DevOps: Why Enterprise Is Moving to DevOps

Published On: August 2, 2017 by **Thomas Johnston**

To stay competitive in 2017 and beyond, enterprise organizations are embracing DevOps methodologies and new technologies to accelerate
II. MLOps is unsustainable (today)

“Here’s the model”
II. MLOps is unsustainable (today)

“Here’s the model”

“This data isn’t available yet”
II. MLOps is unsustainable (today)

“Here’s the model”

“This data isn’t available yet”

“Try this instead”
II. MLOps is unsustainable (today)

“Here’s the model”

“Try this instead”

“This data isn’t available yet”

“Wrong version of numpy”
II. MLOps is unsustainable (today)

“Here’s the model”

“Try this instead”

“That should be corrected”

“This data isn’t available yet”

“Wrong version of numpy”
II. MLOps is unsustainable (today)

“Here’s the model”

“Try this instead”

“That should be corrected”

“This data isn’t available yet”

“Wrong version of numpy”

“This null value isn’t handled”
II. MLOps is unsustainable (today)

“Here’s the model”

“Try this instead”

“That should be corrected”

“Try again?”

“This data isn’t available yet”

“Wrong version of numpy”

“This null value isn’t handled”
II. MLOps is unsustainable (today)

“Here’s the model”
“Try this instead”
“That should be corrected”
“Try again?”

“This data isn’t available yet”
“Wrong version of numpy”
“This null value isn’t handled”
“The graphs aren’t displaying”
II. MLOps is unsustainable (today)

“Here’s the model”
“Try this instead”
“That should be corrected”
“Try again?”
“OK, delete that part”

“This data isn’t available yet”
“Wrong version of numpy”
“This null value isn’t handled”
“The graphs aren’t displaying”
II. MLOps is unsustainable (today)

“Here’s the model”

“Try this instead”

“That should be corrected”

“Try again?”

“OK, delete that part”

“This data isn’t available yet”

“Wrong version of numpy”

“This null value isn’t handled”

“The graphs aren’t displaying”

“This takes too long in prod”
II. MLOps is unsustainable (today)

“Here’s the model”

“Try this instead”

“That should be corrected”

“Try again?”

“OK, delete that part”

“... Ready to try version two?”

“This data isn’t available yet”

“Wrong version of numpy”

“This null value isn’t handled”

“The graphs aren’t displaying”

“This takes too long in prod”
Developer experience

$ cookiecutter git@github.com:teikametrics/sagemaker-framework.git

github_username [my-github-username]: hchasestevens
project_name [my-sagemaker-model]: europython-example-model
project_slug [europython_example_model]:
model_name [europython-example-model]:
description [An ML model living on the SageMaker platform.]: An example model for Europython 2020.

Select model_validation_metric:
1 - sklearn.metrics.mean_squared_error
2 - sklearn.metrics.r2_score
3 - sklearn.metrics.accuracy_score
4 - sklearn.metrics.log_loss
5 - sklearn.metrics.f1_score
6 - sagemaker_framework.utils.metrics.mean_absolute_percentage_error
Choose from 1, 2, 3, 4, 5, 6 (1, 2, 3, 4, 5, 6) [1]: 1

Select promotion_criterion:
1 - sagemaker_framework.utils.promotion.maximize
2 - sagemaker_framework.utils.promotion.minimize
3 - sagemaker_framework.utils.promotion.maximize_with_tol
4 - sagemaker_framework.utils.promotion.minimize_with_tol
5 - sagemaker_framework.utils.promotion.manual
6 - sagemaker_framework.utils.promotion.always_promote
Choose from 1, 2, 3, 4, 5, 6 (1, 2, 3, 4, 5, 6) [1]: 6

preprocessing_cpus [1]: 8
test_proportion [0.2]: 0.1
training_cpus [1]:
training_memory_in_gb [4]:
max_training_runtime_in_minutes [30]: 60
min_serving_instances [1]:
max_serving_instances [10]: 1
preprocessing_memory_in_gb [4]:
training_memory_in_gb [4]: 4
Developer experience

$ cookiecutter git@github.com:teikametrics/sagemaker-framework.git
github_username [my-github-username]: hchasestevens
project_name [my-sagemaker-model]: europython-example-model
project_slug [europython_example_model]:
model_name [europython-example-model]:
description [An ML model living on the SageMaker platform.]: An example model for Europython 2020.

Select model_validation_metric:
1 - sklearn.metrics.mean_squared_error
2 - sklearn.metrics.r2_score
3 - sklearn.metrics.accuracy_score
4 - sklearn.metrics.log_loss
5 - sklearn.metrics.f1_score
6 - sagemaker_framework.utils.metrics.mean_absolute_percentage_error
Choose from 1, 2, 3, 4, 5, 6 (1, 2, 3, 4, 5, 6) [1]: 1

Select promotion_criterion:
1 - sagemaker_framework.utils.promotion.maximize
2 - sagemaker_framework.utils.promotion.minimize
3 - sagemaker_framework.utils.promotion.maximize_with_tol
4 - sagemaker_framework.utils.promotion.minimize_with_tol
5 - sagemaker_framework.utils.promotion.manual
6 - sagemaker_framework.utils.promotion.always_promote
Choose from 1, 2, 3, 4, 5, 6 (1, 2, 3, 4, 5, 6) [1]: 6

preprocessing_cpus [1]: 1
preprocessing_memory_in_gb [4]: 8

Select test_proportion [0.2]: 0.1

training_cpus [1]:
training_memory_in_gb [4]:
training_volume_size_in_gb [2]:
max_training_runtime_in_minutes [30]: 60

min_serving_instances [1]:
max_serving_instances [10]: 1

serving_cpus [1]:
serving_memory_in_gb [4]:

test_proportion [0.2]: 0.1

$ tree -a europython-example-model/
europython-example-model/
	.bellybutton.yml
	bin
		build-docker-image
		deploy.sh
	.circlici
		config.yml
	
docker-compose.yml

dockerfile

europython_example_model
	config.py
		__init__.py
		model.py

github
	CODEOWNERS
	PULL_REQUEST_TEMPLATE.md

gitignore

README.md

requirements.txt

sagemaker-config.yml

setup.py

tests
	
test_config.py

test_model.py

test-model.txt

5 directories, 19 files
Developer experience

$ cookiecutter git@github.com:teikametrics/sagemaker-framework.git

github_username [my-github-username]: hchasestevens
project_name [my-sagemaker-model]: europython-example-model
project_slug [europython_example_model]:
model_name [europython-example-model]:
description [An ML model living on the SageMaker platform.]: An example model for Europython 2020.

Select model_validation_metric:
1 - sklearn.metrics.mean_squared_error
2 - sklearn.metrics.r2_score
3 - sklearn.metrics.accuracy_score
4 - sklearn.metrics.log_loss
5 - sklearn.metrics.f1_score
6 - sagemaker_framework.utils.metrics.mean_absolute_percentage_error
Choose from 1, 2, 3, 4, 5, 6 (1, 2, 3, 4, 5, 6) [1]: 1

Select promotion_criterion:
1 - sagemaker_framework.utils.promotion.maximize
2 - sagemaker_framework.utils.promotion.minimize
3 - sagemaker_framework.utils.promotion.maximize_with_tol
4 - sagemaker_framework.utils.promotion.minimize_with_tol
5 - sagemaker_framework.utils.promotion.manual
6 - sagemaker_framework.utils.promotion.always_promote
Choose from 1, 2, 3, 4, 5, 6 (1, 2, 3, 4, 5, 6) [1]: 6

preprocessing_cpus [1]: 1
preprocessing_memory_in_gb [4]: 8
test_proportion [0.2]: 0.1
training_cpus [1]:
training_memory_in_gb [4]:
max_training_runtime_in_minutes [30]: 60
min_serving_instances [1]:

$ tree -a europython-example-model/

europython-example-model/
    .bellybutton.yml
    bin
    ├── build-docker-image
    │    deploy.sh
    ├── .circleci
    │    config.yml
    │    docker-compose.yml
    │    Dockerfile
    │    europython_example_model
    │    ├── config.py
    │    │    __init__.py
    │    │    model.py
    │    ├── .github
    │    │    CODEOWNERS
    │    │    PULL_REQUEST_TEMPLATE.md
    │    │    .gitignore
    │    │    README.md
    │    │    requirements.txt
    │    │    sagemaker-config.yml
    │    │    setup.py
    │    │    tests
    │    │    ├── test_config.py
    │    │    │    test_model.py
    │    │    │    test-model.txt
    │    │    └── test-model.txt
    └── 5 directories, 19 files
def preprocess_data(seed=None) -> PreprocessingResult:
    '''Preprocess data for training.'''
    fetch_adgroup_performances_query = 
    SELECT
        ad_group_id,
        SUM(lkr.conversions_7d_attr) AS conversions,
        SUM(lkr.sales_7d_attr) AS sales
    FROM main.transforms.latest_keyword_reports lkr
    WHERE lkr.conversions_7d_attr > 0
        AND lkr.sales_7d_attr > 0
        AND lkr.keyword_report_local_date >= current_date() - 30
    GROUP BY ad_group_id
    
    return PreprocessingResult(
        training={
            'performances.msgpack': adgroup_performances[
                ~adgroup_performances.test
            ].apply(pd.to_numeric).to_msgpack(),
        },
        validation={},
        testing=test_cases
    )
def preprocess_data(seed=None) -> PreprocessingResult:
    """Preprocess data for training."""
    fetch_adgroup_performances_query = ""
    SELECT
        ad_group_id,
        SUM(lkr.conversions_7d_attr) AS conversions,
        SUM(lkr.sales_7d_attr) AS sales
    FROM main.transforms.latest_keyword_reports lkr
    WHERE lkr.conversions_7d_attr > 0
    AND lkr.sales_7d_attr > 0
    AND lkr.keyword_report_local_date >= current_date() - 30
    GROUP BY ad_group_id
    ""
    return PreprocessingResult(
        training={
            'performances.msgpack': adgroup_performances.
            apply(pd.to_numeric).to_msgpack(),
        }.items(),
        validation={},
        testing=test_cases
    )

def train_model(training_path: Path, validation_path: Path) -> Artifacts:
    training_dfs = load_zipped_data(
        training_path,
        fnames=MSGPACK_FNAMES,
        deserializer=pd.read_msgpack
    )
    all_adgroup_prices = training_dfs['prices.msgpack']
    performances = training_dfs['performances.msgpack']
    results = {
        marketplace_id: train_marketplace_model(
            marketplace_id=marketplace_id,
            market_adgroup_prices=market_df,
            performances=performances,
        ).asdict()
        for marketplace_id, market_df in all_adgroup_prices.groupby('marketplace_id')
    }
    return Artifacts({MODEL_FNAME: json.dumps(results).encode('utf-8')})
def preprocess_data(seed=None) -> PreprocessingResult:
    """Preprocess data for training."""
    fetch_adgroup_performances_query = ""
    SELECT
        ad_group_id,
        SUM(lkr.conversions_7d_attr) AS conversions,
        SUM(lkr.sales_7d_attr) AS sales
    FROM main.transforms.latest_keyword_reports lkr
    WHERE lkr.conversions_7d_attr > 0
    AND lkr.sales_7d_attr > 0
    AND lkr.keyword_report_local_date >= current_date() - 30
    GROUP BY ad_group_id
    ""
    return PreprocessingResult(
        training=
        {'performances.msgpack': adgroup_performances[  
            ~adgroup_performances.test
        ].apply(pd.to_numeric).to_msgpack()},
        validation=(),
        testing=test_cases
    )

def train_model(training_path: Path, validation_path: Path) -> Artifacts:
    training_dfs = load_zipped_data(
        training_path,
        fnames=MSGPACK_FNAMES,
        deserializer=pd.read_msgpack
    )
    all_adgroup_prices = training_dfs['prices.msgpack']
    performances = training_dfs['performances.msgpack']
    results = {
        marketplace_id: train_marketplace_model(
            marketplace_id=marketplace_id,
            market_adgroup_prices=market_df,
            performances=performances,
        ).asdict()
        for marketplace_id, market_df in all_adgroup_prices.groupby('marketplace_id')
    }
    return Artifacts({MODEL_FNAME: json.dumps(results).encode('utf-8')})

def load_model(path: Path) -> Model:
    with (path / MODEL_FNAME).open('r', encoding='utf-8') as f:
        parameters = {k: Parameters(**v) for k, v in json.load(f).items()}
    def model(configuration, instances) -> List[Optional[float]]:
        return [
            estimate_sales_per_conversion(...)  
            for price, conversions, sales in instances
        ]
    return model
Developer experience

request_schema: !jsonschema {
   type: 'object',
   properties: {
      configuration: {
         type: 'object',
         properties: {
            marketplaceId: {type: 'string'}
         }
      },
      instances: {
         type: 'array',
         items: {
            type: 'array',
            items: [
               {type: 'number', description: "Price", exclusiveMinimum: 0},
               {type: 'number', description: "Conversions", exclusiveMinimum: 0},
               {type: 'number', description: "Sales", exclusiveMinimum: 0}
            ],
         }
      },
      requesterId: {type: 'string'}
   },
   required: ['instances', 'configuration', 'requesterId'],
}

response_schema: !jsonschema {
   type: 'array',
   items: {type: 'number'},
   description: "Estimated sales per conversion, in order corresponding to request order"
}
Developer experience

- Test suite
- Linting (pylint, mypy, bellybutton)
- Dockerization
- CI/CD
- Airflow DAG generation
- Training orchestration
- Automated model evaluation and promotion
- Gradual rollout
Developer experience

- Test suite
- Linting (pylint, mypy, bellybutton)
- Dockerization
- CI/CD
- Airflow DAG generation
- Training orchestration
- Automated model evaluation and promotion
- Gradual rollout
- Automated rollback
- Monitoring
- Alerting
- Diagnostics
- Autoscaling
- Schema validation
- Data capture
- Healthchecks
- Cost monitoring
Developer experience

- Test suite
- Linting (pylint, mypy, bellybutton)
- Dockerization
- CI/CD
- Airflow DAG generation
- Training orchestration
- Automated model evaluation and promotion
- Gradual rollout

- Automated rollback
- Monitoring
- Alerting
- Diagnostics
- Autoscaling
- Schema validation
- Data capture
- Healthchecks
- Cost monitoring

III. Data scientists want to do data science
Developer experience
Our stack
## Our stack

<table>
<thead>
<tr>
<th>Technology</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS SageMaker</td>
<td>Model training, hosting; provenance info</td>
</tr>
<tr>
<td>Airflow (Astronomer.io)</td>
<td>Model lifecycle orchestration</td>
</tr>
<tr>
<td>Docker</td>
<td>Model packaging</td>
</tr>
<tr>
<td>Cookiecutter</td>
<td>Model repo templating</td>
</tr>
<tr>
<td>Jsonschema</td>
<td>Schema definition; PBT</td>
</tr>
<tr>
<td>Flask, gunicorn</td>
<td>Model server</td>
</tr>
<tr>
<td>DBT</td>
<td>Scalable data processing (in-warehouse)</td>
</tr>
<tr>
<td>Slack</td>
<td>Notifications, diagnostics</td>
</tr>
<tr>
<td>Pylint, mypy, bellybutton</td>
<td>Linting</td>
</tr>
<tr>
<td>Pytest, hypothesis, hypothesis-jsonschema</td>
<td>Test suite</td>
</tr>
</tbody>
</table>
Our stack
Lessons learned
Lessons learned

Model ready for promotion.
New model mean_absolute_percentage_error: 0.125717809901924
Production mean_absolute_percentage_error: 0.12795042672835955

Promoting to endpoint.
Updating existing endpoint configuration.
Lessons learned

"Best Practices" (whatever that means):

Lessons learned

"Best Practices" (whatever that means): "Worst Practices" (whatever that means):

## Lessons learned

<table>
<thead>
<tr>
<th>Instance type</th>
<th>vCPU</th>
<th>GPU</th>
<th>Mem (GiB)</th>
<th>GPU Mem (GiB)</th>
<th>Network Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standard – Current Generation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ml.t2.medium</td>
<td>2</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>Low to Moderate</td>
</tr>
<tr>
<td>ml.t2.large</td>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
<td>Low to Moderate</td>
</tr>
<tr>
<td>ml.t2.xlarge</td>
<td>4</td>
<td>16</td>
<td></td>
<td></td>
<td>Moderate</td>
</tr>
<tr>
<td>ml.t2.2xlarge</td>
<td>8</td>
<td>32</td>
<td></td>
<td></td>
<td>Moderate</td>
</tr>
<tr>
<td>ml.t3.medium</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td>Low to Moderate</td>
</tr>
<tr>
<td>ml.t3.large</td>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
<td>Low to Moderate</td>
</tr>
</tbody>
</table>
Lessons learned

<table>
<thead>
<tr>
<th>Column name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lesson 1</td>
<td>Text</td>
<td>Learned Principles</td>
</tr>
<tr>
<td>Lesson 2</td>
<td>Text</td>
<td>Another Principle</td>
</tr>
<tr>
<td>Lesson 3</td>
<td>Text</td>
<td>A Third Principle</td>
</tr>
<tr>
<td>Lesson 4</td>
<td>Text</td>
<td>Fourth Lesson</td>
</tr>
<tr>
<td>Lesson 5</td>
<td>Text</td>
<td>Lesson 5 Principles</td>
</tr>
<tr>
<td>Lesson 6</td>
<td>Text</td>
<td>Lesson Six Principles</td>
</tr>
<tr>
<td>Lesson 7</td>
<td>Text</td>
<td>Lesson Seven Principles</td>
</tr>
<tr>
<td>Lesson 8</td>
<td>Text</td>
<td>Lesson Eight Principles</td>
</tr>
<tr>
<td>Lesson 9</td>
<td>Text</td>
<td>Lesson Nine Principles</td>
</tr>
<tr>
<td>Lesson 10</td>
<td>Text</td>
<td>Lesson Ten Principles</td>
</tr>
<tr>
<td>Lesson 11</td>
<td>Text</td>
<td>Lesson Eleven Principles</td>
</tr>
<tr>
<td>Lesson 12</td>
<td>Text</td>
<td>Lesson Twelve Principles</td>
</tr>
</tbody>
</table>

Note: The value column shows the nature of the lessons learned.
Lessons learned
Lessons learned
Lessons learned

- **instances**: 1
- **instance_count**: 1
- **cpu**: <0.25 vCPUs
- **memory**: <0.5 GB
- **volume_size**: <2 GB
Lessons learned

Airflow:

- Hosting our own stack
- Deployment interruptions
- Not all contributions created equal
Questions?

H. Chase Stevens
Principal Data Science Engineer, teikametrics
Boston, MA
chase@chasestevens.com
@hchasestevens

https://www.teikametrics.com/company.html#careers