MACHINE LEARNING ENGINEERING DONE RIGHT

Designing and Building Complex Intelligent Systems and Workflows with **Python**

[CONF42]

NUWORKS

Joshua “ARVS” Lat

- **Chief Technology Officer** of NuWorks Interactive Labs
- **AWS Machine Learning Hero**
- Author of a Machine Learning Engineering Book - **Amazon SageMaker Cookbook** (2021)]

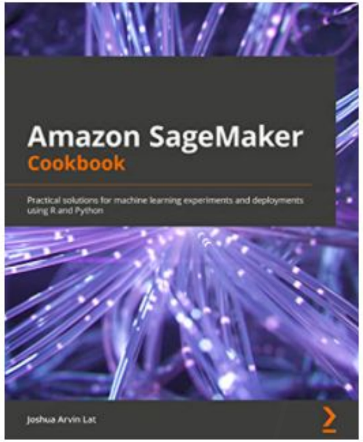
amazon

Kindle Store

All Today's Deals Buy Again Joshua's Amazon.com Browsing History Customer Service Registry Sell Gift Cards

Buy a Kindle Kindle eBooks Kindle Unlimited Prime Reading Best Sellers & More Kindle Book Deals Kindle Singles Newsstand Manage content and devices Advanced Search

‹ Back to results



Amazon SageMaker Cookbook: Practical solutions for machine learning experiments and deployments using R and Python Kindle Edition

by Joshua Arvin Lat (Author) | Format: Kindle Edition

› See all formats and editions

Effectively manage end-to-end machine learning developments and workflows using SageMaker Data Wrangler, SageMaker Debugger, SageMaker Experiments, and more

Key Features

- Perform machine learning experiments with built-in Amazon SageMaker algorithms using R and

‹ Read more

Print length	Language	Publisher	Publication date	Page Flip
724 pages	English	Packt Publishing	July 9, 2021	Not Enabled

Follow the Author

Joshua Arvin Lat

+ Follow

NUWORKS

MACHINE LEARNING ENGINEERING DONE RIGHT

- UNDERSTANDING THE NEEDS OF THE BUSINESS AND THE CUSTOMERS
- MAKING THE MOST OUT OF ML FRAMEWORKS AND ML PLATFORMS
- KNOWING WHEN TO WRITE PRODUCTION-LEVEL PYTHON CODE
- WORKING WITH AUTOMATED ML BIAS DETECTION AND ML EXPLAINABILITY CAPABILITIES
- ENFORCING PRACTICAL PYTHON CODING GUIDELINES
- REAPING THE BENEFITS OF CLOUD COMPUTING FOR AUTOMATED HYPERPARAMETER OPTIMIZATION JOBS
- USING PYTHON DESIGN PATTERNS AND METAPROGRAMMING TECHNIQUES
- OPTIMIZING COSTS BY USING TRANSIENT ML INSTANCES FOR TRAINING MODELS
- UTILIZING CONTINUOUS INTEGRATION AND DEPLOYMENT PIPELINES
- SECURING MACHINE LEARNING ENVIRONMENTS

UNDERSTANDING THE NEEDS OF THE BUSINESS AND THE CUSTOMERS

MACHINE LEARNING PREDICTION ENDPOINT

[FLASK]

VS

MACHINE LEARNING EXPERIMENT

[JUPYTER NOTEBOOK]

ENFORCING PRACTICAL PYTHON CODING GUIDELINES

20-LINE RULE

PEP-8

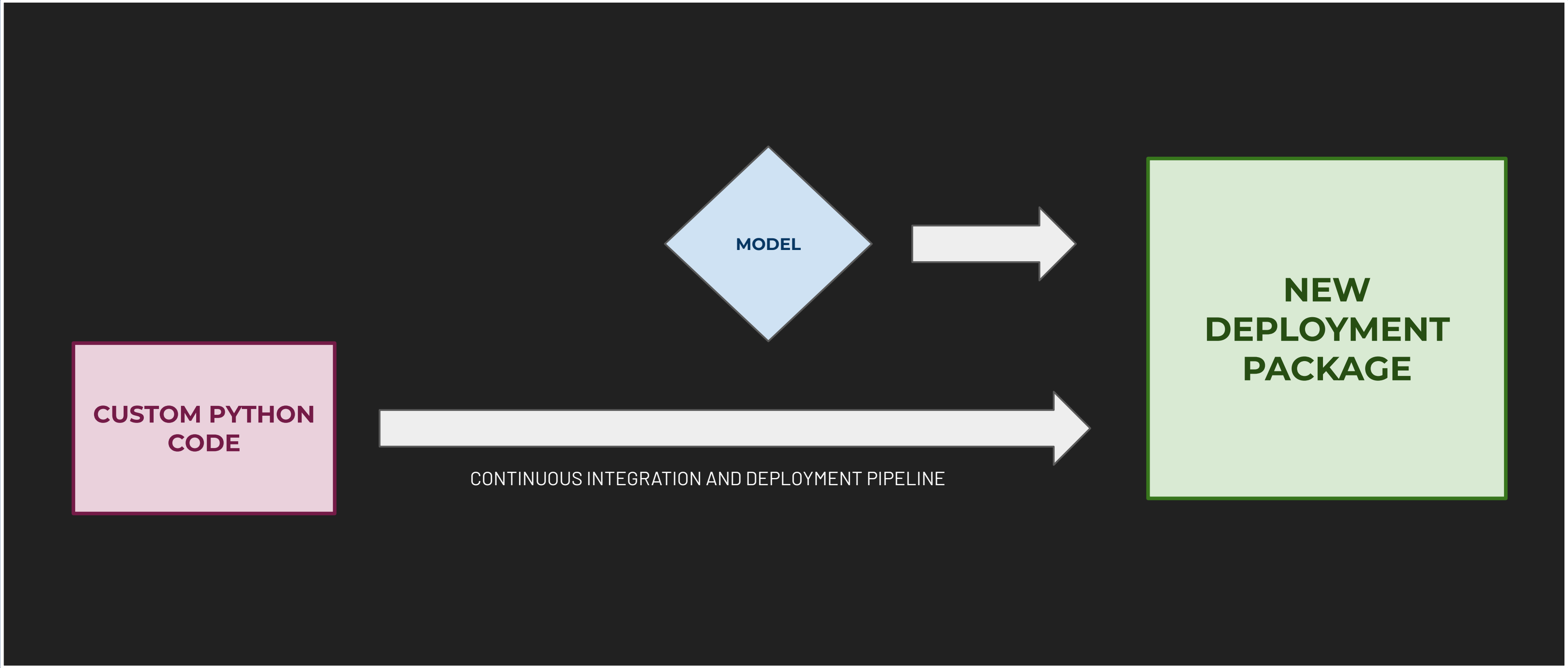
AVOIDANCE OF TRY-CATCH BLOCKS

WRITE TESTABLE PYTHON CODE

WRITE YOUR OWN CONVENIENCE LIBRARY!

(when it makes sense)

UTILIZING CONTINUOUS INTEGRATION AND DEPLOYMENT PIPELINES



MACHINE LEARNING ENGINEERING DONE RIGHT

UNDERSTANDING THE NEEDS OF THE BUSINESS AND THE CUSTOMERS

KNOWING WHEN TO WRITE PRODUCTION-LEVEL PYTHON CODE

ENFORCING PRACTICAL PYTHON CODING GUIDELINES

USING PYTHON DESIGN PATTERNS AND METAPROGRAMMING TECHNIQUES

UTILIZING CONTINUOUS INTEGRATION AND DEPLOYMENT PIPELINES

MAKING THE MOST OUT OF ML FRAMEWORKS AND ML PLATFORMS

WORKING WITH AUTOMATED ML BIAS DETECTION AND ML EXPLAINABILITY CAPABILITIES

REAPING THE BENEFITS OF CLOUD COMPUTING FOR AUTOMATED HYPERPARAMETER OPTIMIZATION JOBS

OPTIMIZING COSTS BY USING TRANSIENT ML INSTANCES FOR TRAINING MODELS

SECURING MACHINE LEARNING ENVIRONMENTS

BUILD EVERYTHING FROM SCRATCH

VS


Data Collection
Data Preparation and Cleaning
Data Visualization and Analysis
Feature Engineering
Model Training and Parameter Tuning
Model Evaluation
Model Deployment

MAKING THE MOST OUT OF ML FRAMEWORKS AND ML PLATFORMS

MODEL DEPLOYED IN AN EC2 INSTANCE

MODEL DEPLOYED IN A CONTAINER IN AN EC2 INSTANCE

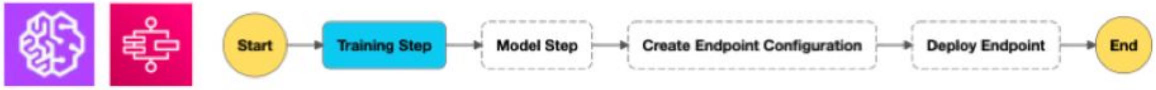
BUILT-IN ALGORITHM + SAGEMAKER ENDPOINT

CUSTOM CONTAINER + SAGEMAKER ENDPOINT

MAKING THE MOST OUT OF ML FRAMEWORKS AND ML PLATFORMS

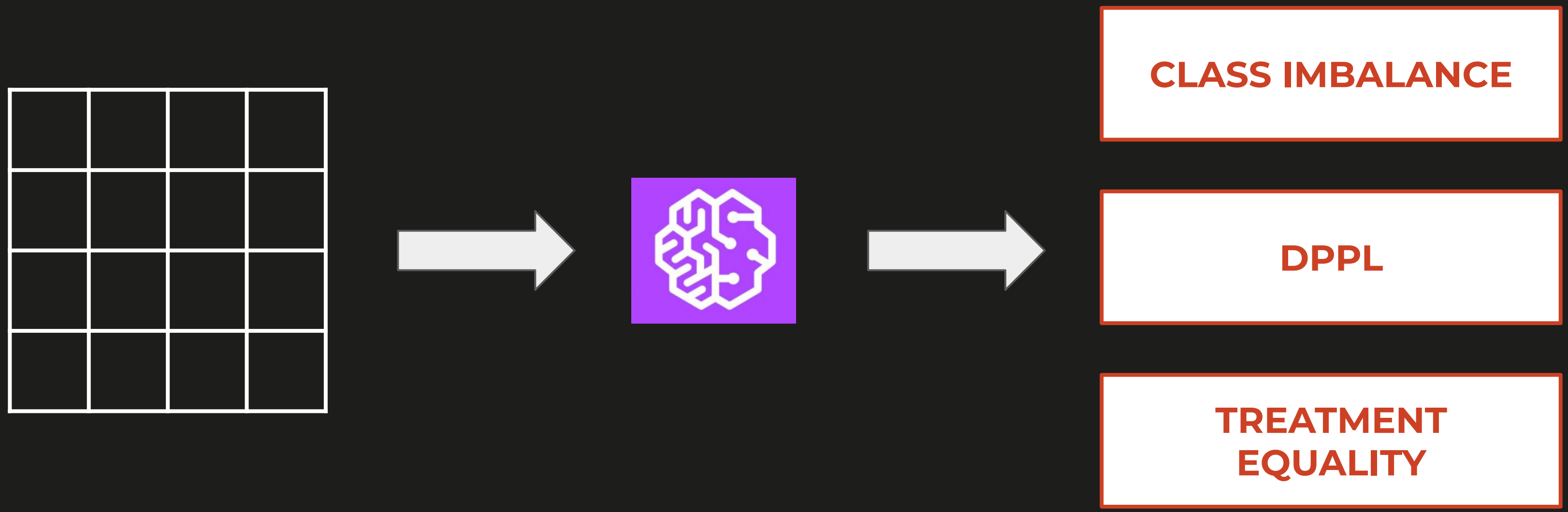
- MODEL DEPLOYED INSIDE A LAMBDA FUNCTION
- LAMBDA TRIGGERING A SAGEMAKER ENDPOINT
- API GATEWAY MAPPING TEMPLATES + SAGEMAKER
- MODEL DEPLOYED IN FARGATE

- SAGEMAKER MULTI-MODEL ENDPOINT
- SAGEMAKER MULTI-CONTAINER ENDPOINT
- SAGEMAKER A/B TESTING SETUP USING PRODUCTION VARIANTS
- MODEL DEPLOYED INSIDE A LAMBDA FUNCTION + CONTAINER



ID	Type	Step	Resource	Elapsed Time (ms)	Timestamp
▶ 1	ExecutionStarted		-	0.0	Sep 10, 2020 05:39:24.116 PM
▶ 2	TaskStateEntered	Training Step	-	32.0	Sep 10, 2020 05:39:24.148 PM
▶ 3	TaskScheduled	Training Step	Sagemaker training job	32.0	Sep 10, 2020 05:39:24.148 PM
▶ 4	TaskStarted	Training Step	Sagemaker training job	67.0	Sep 10, 2020 05:39:24.183 PM
▶ 5	TaskSubmitted	Training Step	Sagemaker training job	288.0	Sep 10, 2020 05:39:24.404 PM

WORKING WITH AUTOMATED ML BIAS DETECTION AND ML EXPLAINABILITY CAPABILITIES



```
from sagemaker import clarify


processor = clarify.SageMakerClarifyProcessor(
    role=role,
    instance_count=1,
    instance_type='ml.m5.large',
    sagemaker_session=session)


data_config = clarify.DataConfig(
    s3_data_input_path=s3_training_data_path,
    s3_output_path=s3_output_path,
    label='label',
    headers=training_data.columns.to_list(),
    dataset_type='text/csv')
```

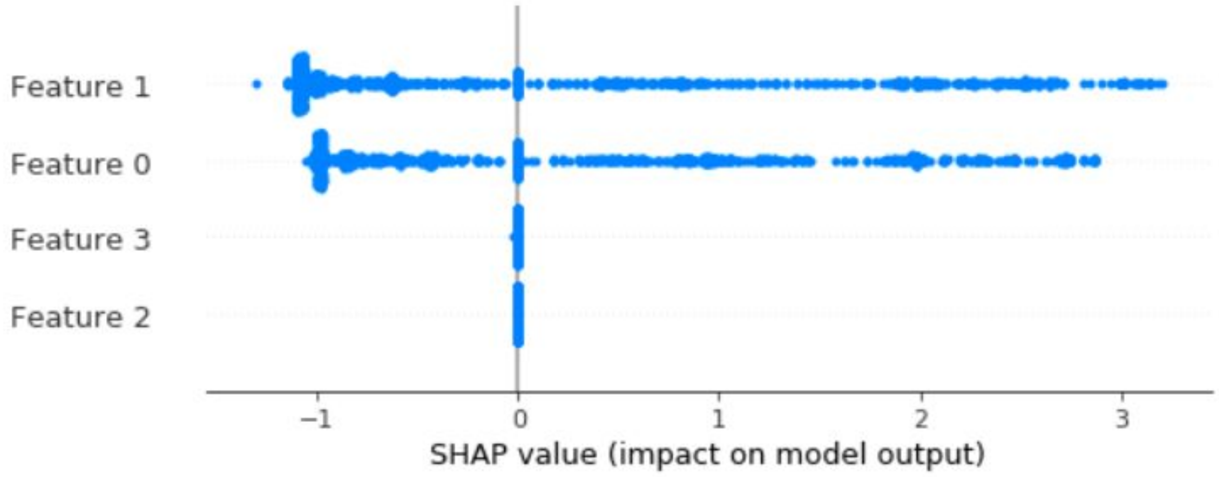
```
bias_config = clarify.BiasConfig(
    label_values_or_threshold=[1],
    facet_name='a',
    facet_values_or_threshold=[5])


processor.run_pre_training_bias(
    data_config=data_config,
    data_bias_config=bias_config,
    methods=['CI'])



processor.latest_job.outputs[0].destination
```

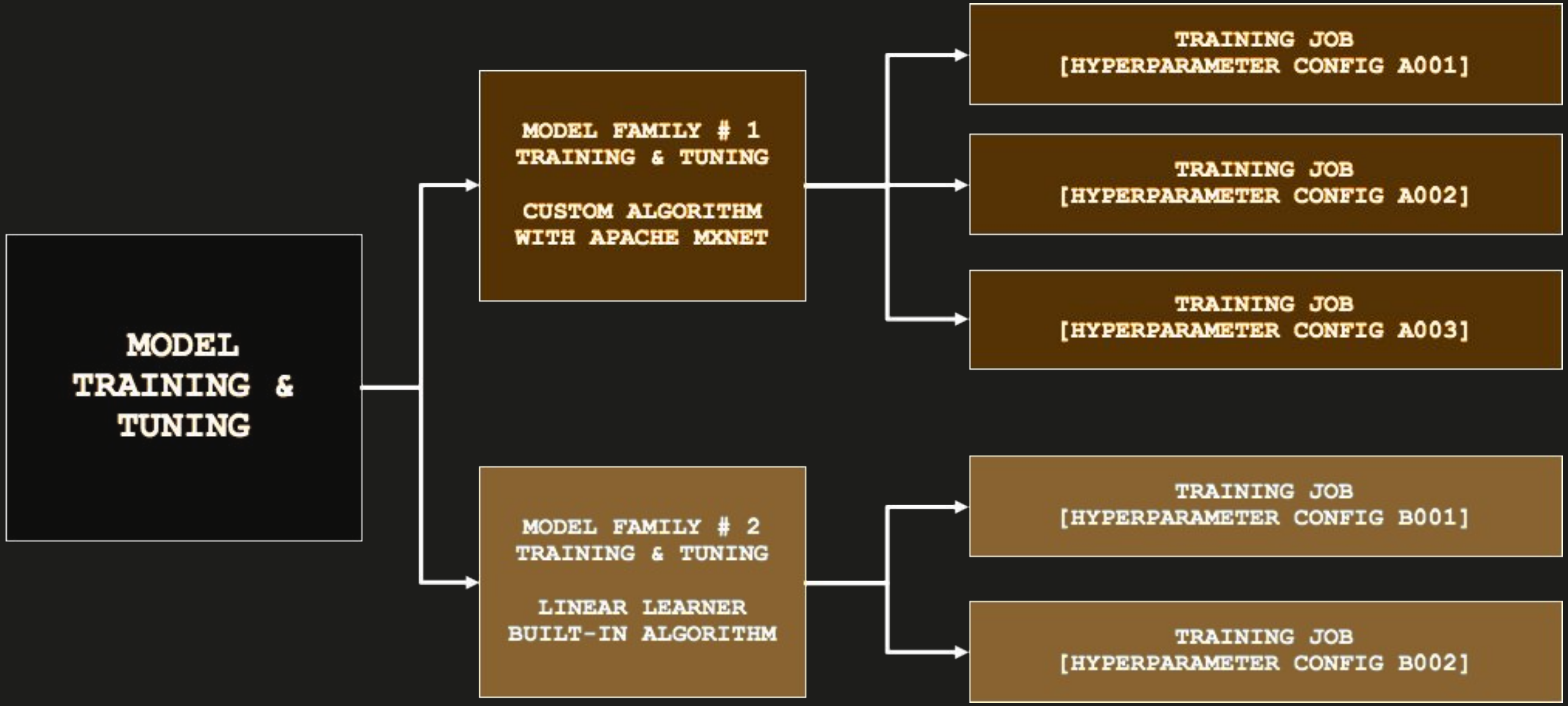
```json
{
    "version": "1.0",
    "pre_training_bias_metrics": {
        "label": "label",
        "facets": {
            "a": [
                {
                    "value_or_threshold": "(5.0, 13.99152988349206]",
                    "metrics": [
                        {
                            "name": "CI",
                            "description": "Class Imbalance (CI)",
                            "value": 0.9573333333333334
                        }
                    ]
                }
            ]
        },
        "label_value_or_threshold": "1"
    }
}
```


Feature 1
Feature 0
Feature 3
Feature 2
−1
0
1
2
3
SHAP value (impact on model output)

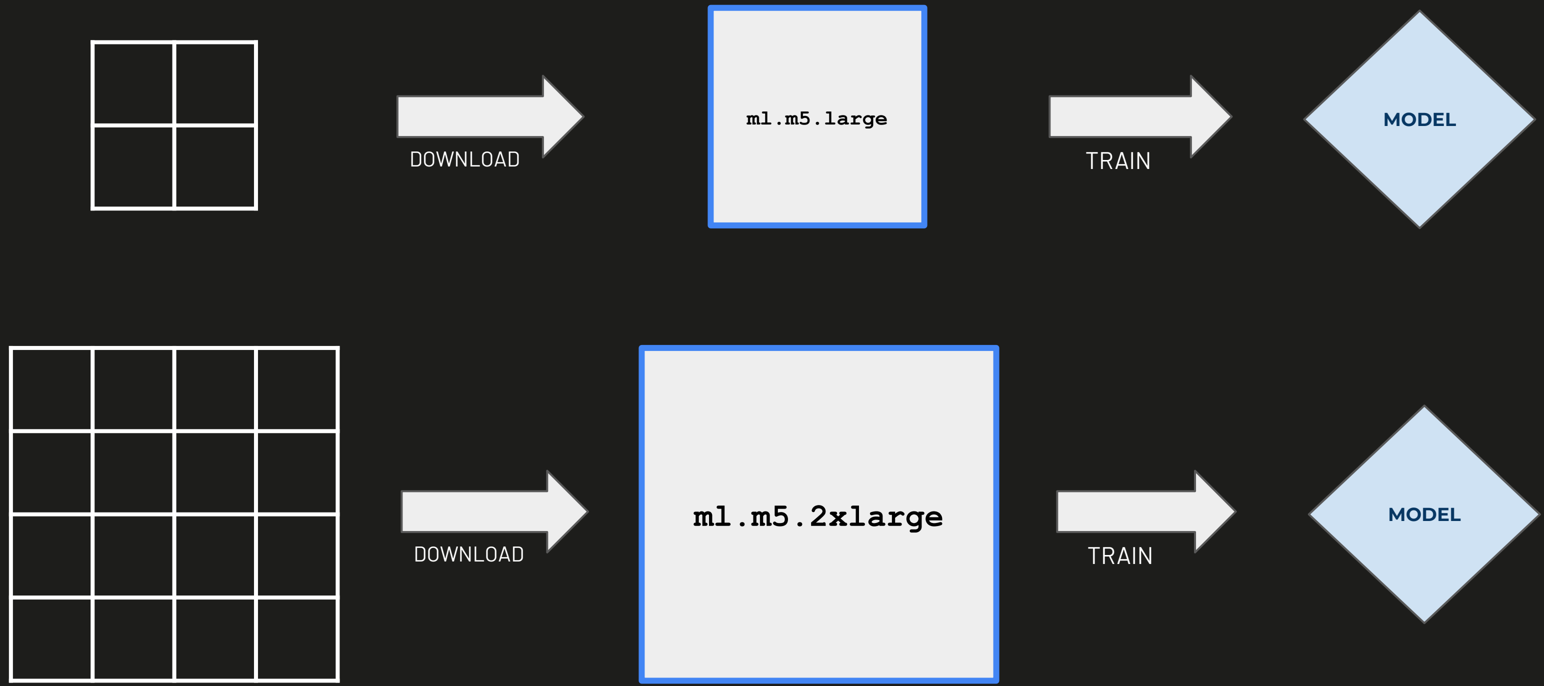
```
{
    "version": "1.0",
    "explanations": {
        "kernel_shap": {
            "label0": {
                "global_shap_values": {
                    "a": 0.1173995901699019,
                    "b": 0.37360024663733005,
                    "c": 0.01740283967164966,
                    "d": 0.015364162067494701
                },
                "expected_value": 0.34422817826271057
            }
        }
    }
}
```


MODEL
TRAINING &
TUNING
TRAINING JOB
[HYPERPARAMETER CONFIG 001]
TRAINING JOB
[HYPERPARAMETER CONFIG 002]
TRAINING JOB
[HYPERPARAMETER CONFIG 003]
TRAINING JOB
[HYPERPARAMETER CONFIG 004]
TRAINING JOB
[HYPERPARAMETER CONFIG 005]

REAPING THE BENEFITS OF CLOUD COMPUTING FOR AUTOMATED HYPERPARAMETER OPTIMIZATION JOBS



OPTIMIZING COSTS BY USING TRANSIENT ML INSTANCES FOR TRAINING MODELS



PRINCIPLE OF LEAST PRIVILEGE

Joblib

Navigation

Persistence

Use case

`joblib.dump()` and `joblib.load()` provide a replacement for pickle to work efficiently on arbitrary Python objects containing large data, in particular large numpy arrays.

Warning:

`joblib.dump()` and `joblib.load()` are based on the Python pickle serialization model, which means that arbitrary Python code can be executed when loading a serialized object with `joblib.load()`.

`joblib.load()` should therefore never be used to load objects from an untrusted source or otherwise you will introduce a security vulnerability in your program.

Note:

As of Python 3.8 and numpy 1.16, pickle protocol 5 introduced in PEP 574 supports efficient serialization and de-serialization for large data buffers natively using the standard library:

```
pickle.dump(large_object, fileobj, protocol=5)
```

MACHINE LEARNING ENGINEERING DONE RIGHT

- UNDERSTANDING THE NEEDS OF THE BUSINESS AND THE CUSTOMERS
- MAKING THE MOST OUT OF ML FRAMEWORKS AND ML PLATFORMS
- KNOWING WHEN TO WRITE PRODUCTION-LEVEL PYTHON CODE
- WORKING WITH AUTOMATED ML BIAS DETECTION AND ML EXPLAINABILITY CAPABILITIES
- ENFORCING PRACTICAL PYTHON CODING GUIDELINES
- REAPING THE BENEFITS OF CLOUD COMPUTING FOR AUTOMATED HYPERPARAMETER OPTIMIZATION JOBS
- USING PYTHON DESIGN PATTERNS AND METAPROGRAMMING TECHNIQUES
- OPTIMIZING COSTS BY USING TRANSIENT ML INSTANCES FOR TRAINING MODELS
- UTILIZING CONTINUOUS INTEGRATION AND DEPLOYMENT PIPELINES
- SECURING MACHINE LEARNING ENVIRONMENTS

Thanks!

FEEL FREE TO REACH OUT AND CONNECT:

EMAIL: joshua.arvin.lat@gmail.com
LINKEDIN: https://www.linkedin.com/in/joshualat/