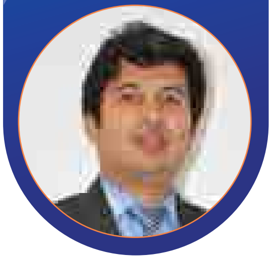
infogain
Engineering Business Outcomes
+
ABSOLUTDATA

BrainWave

DATA SCIENCE DIGEST

8TH EDITION

Q3 2022



Vipin Tripathi
Manager, Data Science & Analytics at Absolutdata

Next-Gen Machine Learning with MLOps

Despite the buzz around Machine Learning (ML), many organizations haven't gotten fully onboard with ML yet – or received the full benefits from experimenting with and deploying ML. One reason is that the ML development and deployment process can be slow and complicated. As a result, few decision-makers in this space feel their ML programs are sophisticated; according to Deloitte, 28% of Machine Learning projects fail due to problems in expertise, data, and development environments. Many more projects never even make it to the development stage.[1]

Machine Learning Operations, or MLOps, aims to fix this problem by improving the management and delivery of ML projects. It's a multifaceted approach that includes cross-team communication, simplified model deployment, automated model monitoring, production life cycle management, and enhanced practices around model deployment and governance. The MLOps umbrella includes:

- Data sourcing, labelling, and versioning.
- Model development, training, validation, and evaluation.
- Model versioning and deployment.
- Prediction monitoring and ongoing maintenance (including re-training).

Why MLOps?

AI and by extension ML are supposed to deliver insights into various business problems. They're meant to help us unlock hidden sources of revenue (and cut hidden cash drains), save time, work more efficiently, and make smarter decisions. But these goals are hard to accomplish without a meaningful framework.

MLOps borrows from DevOps and emphasizes automation and cross-team communication to speed development times while reducing the associated costs. It can become that needed framework that guides the widespread adoption and success of business ML. And, while we often focus on the large-scale applications of enterprise AI, MLOps is flexible and scalable; it can be used in large and small cases. Indeed, you can tailor MLOps to suit your company's needs and goals, implementing the practices that work best for your team.

Conclusion

MLOps can improve the quality of ML models, but it also shortens delivery times and provides for more productive results. More importantly, it makes ML more accessible to business uses. Using this framework of practices, organizations can create scalable solutions and align models with business needs (and regulatory requirements). Companies that can harness the power of MLOps to streamline their ML delivery and production process are likely to be the ones who thrive as we enter what Deloitte calls “the Golden Age of AI”.

References

1. Deloitte Insights: Tech Trends 2021

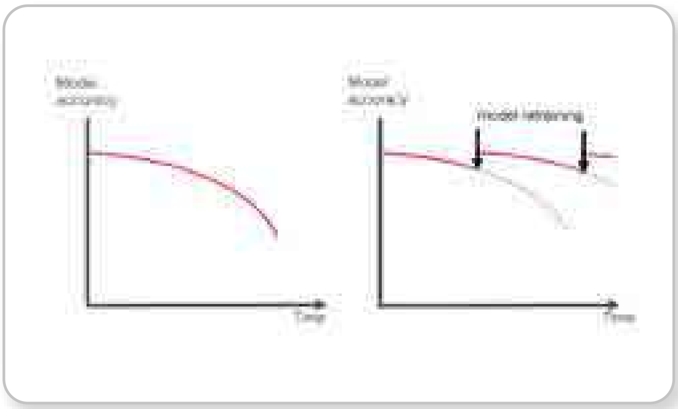
2. VentureBeat: MLOps is about to take off in the enterprise

3. Making the most of MLOps

4. Neptune AI: MLOps: What It Is, Why It Matters, and How to Implement It

Contents

Drift is the deterioration of a model's performance as a result of changes in real-world factors. The performance of any model is as good as the data it is trained on, but the world is dynamic, and data is constantly changing. This results in a gradual degradation of a model's predictive power when compared to its performance during the training period; it means the model may need to be retrained.



Source: Machine Learning Monitoring, Part 5: Why You Should Care About Data and Concept Drift

Types of Drift

Model drift can be further classified into:

1. Concept Drift

Concept drift occurs when there is a change in the relationship between input variables and the target variable. For example, the annual seasonal weather shift prompts consumers to buy warm coats in colder months, but the demand decreases once temperatures rise in spring and then increases again in the winter.

2. Prediction Drift

Prediction drift is a significant change in the distribution of the predictions (label or value), indicating a change in the underlying data. Having ground truth/labels is not required for prediction drift.

Detecting Drift

Drift detection can be done using either of the following two techniques:

1. **Statistical measures:** This approach uses statistical metrics, as they are easier to implement and comprehend. These are also easier to validate and have found a wide variety of uses in several industries.

.

Let's take a look at some of the statistical measures that can be used to detect drift:

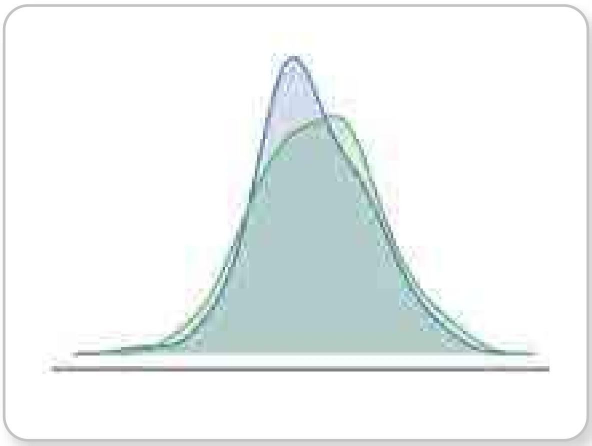
a. **The Kolmogorov-Smirnov** (KS) test compares two different populations based on their cumulative distribution. In this case, the two populations to be compared are training data and post-training data.

   In the context of drift, the KS test has a null hypothesis that the distributions from both datasets are identical. If this hypothesis is rejected, we can safely conclude that the model has drifted.

b. **The Population Stability Index (PSI)** assesses population changes over time. It detects model drift by monitoring population characteristics according to the following formula:

$$PSI = \sum ((Population\ A - Population\ B) * \ln(\frac{Population\ A}{Population\ B}))$$

- If PSI is less than 0.1, there's no substantial change to the population and the existing model is still relevant.
- If PSI is greater than 0.1 but less than 0.2, there's been some change and the model should be adjusted accordingly.
- If PSI is greater than 0.2, there has been a major population shift and the model should be retrained



Source: Population Stability Index

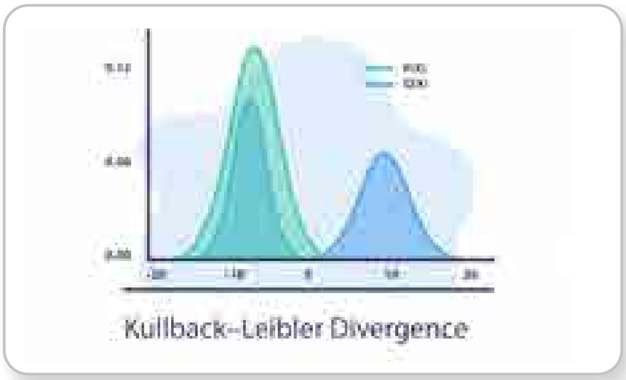
c. **The Kullback–Leibler Divergence** gauges statistical distance; it's also known as relative entropy. It calculates the difference between two probability distributions.

   If A is the distribution of the old data and B is the distribution of the new data, then:

$$KL(A||B) = -\sum_{x} P(x) * \log((\frac{A(x)}{B(x)}))$$

In the above formula, || represents the divergence.

- if P(x) is high and Q(x) is low, the divergence between the data distributions is high.
- If P(x) is low and Q(x) is high, the divergence is more moderate but still significant.
- If P(x) and Q(x) are somewhat alike, the divergence between the distributions is low.



Source: 8 Concept Drift Detection Methods

2. Measuring the Accuracy of the Model

This approach entails training a classification model, which will identify any similarities between data points in separate sets. If the model finds few differences, there's little drift. But if it does identify changes between data points, then drift is probable.

MLOps and ML Drift

Due to drift, there is gradual decay in the performance of the model over an interval of time. This can be resolved by periodically retraining the model on a fresh batch of data. Retraining models in shorter intervals can be rather tricky.

MLOps simplifies this process by automating model retraining. This automatic retraining can be run as a scheduled process using pre-existing data pipelines.

Optimal training frequency depends upon several external factors. However, constant monitoring at regular intervals is the best way to determine model drift and training frequency.

References

Aporia.com: Concept Drift in Machine Learning 101

Aporia.com: Why Monitor Prediction Drifts?

Towards Data Science: How to Detect Model Drift in MLOps Monitoring

Arize.com: Model Drift: Guide to Understanding Drift in AI

Authored by
Abhishek Bisht,
Analyst at Absolutdata

Coding Best Practices in MLOps

Coder’s Cauldron

“…developing and deploying ML systems is relatively fast and cheap but maintaining them over time is difficult and expensive.”

– D. Sculley, Hidden Technical Debt in Machine Learning Systems, NIPS 2015

All professional data scientists know the truth of this quote. You may have encountered this in solving a problem in your machine learning system’s many processes. Solving any number of solutions together usually means screwing up; meanwhile, problems tend to increase in complexity as the system gets older. Even worse, you can waste time and resources and cause production issues.

It can be easy to make a model fulfil its business objectives and deploy it, but operating it in production can be quite challenging. A change in input data or dependent variables may lead to poor performance and production quality. Microsoft itself has stated that data drift is one of the major reasons model accuracy degrades over time.

Also, models need to be retrained on every observation. Determining model performance metrics and continuously monitoring its performance – a process is called continuous training and continuous monitoring – is key to ML models’ long-term success. So let’s consider some best practices for consistent delivery in machine learning systems.

Open Communication

Machine learning teams need to communicate if they want to meet business objectives in the face of changing resources, data patterns, and expectations. In an ML project, teams include experts in data engineering, DevOps, data visualization, AI, software engineering, business domain knowledge, etc. All these experts must work together for the success of the project. And clear communication between these professionals as well as with the business teams and stakeholders is essential if the model is to perform properly.

Cost-Benefit Analysis

A major part of MLOps is the costs of the machine learning lifecycle. Introducing MLOps in short-term projects might be expensive; on the other hand, it is more beneficial for long-term projects. Therefore, understanding how MLOps can be helpful is critical.

Manual efforts to manage ML models can be reduced by automation (which is covered later in this article). This provides engineers more flexibility and more time for productive tasks.

When evaluating the desirability of MLOps. systematic process should be considered, including goals, budgets, ML activities, and team capabilities.

Clear Naming Conventions

Naming conventions make sure that everyone is on the same page. In general, a naming convention is a descriptive way to label variables, functions, and other elements within code or its documentation. The goal is for everyone to be able to easily read and understand the element by its name – what it does, represents, or contains.

A naming convention also helps to promote consistency within the teams. By avoiding naming conflicts, it's easier to transfer projects to another team or a new team member. Therefore, it's a good idea to introduce this standard and get contribution from the entire team, ensuring that everyone moves in the same direction together.

Choose Your MLOps Toolkit Strategically

There are many MLOps tools available. When choosing which tools you will use, think about the long term. Consider your business objective and the tasks you'll need to do. Think about your budget constraints and the ML team's knowledge and skill. Examine the data you'll use and your options for data versioning, parameter tuning, production monitoring, etc. Once you get the complete picture, you'll be able to select MLOps tools that will adequately support your team's efforts throughout the different lifecycle stages.

Experiment Tracking

Developing machine learning models is a highly repetitive process. Unlike the traditional software development process, multiple experiments on model training can be performed in parallel in ML before finalizing the production model.

Experimentation during ML model developments can take the form of several different scenarios. One way to track multiple experiments is to use different branches (such as Git), each one dedicated to a separate experiment. And depending on the performance metrics, the best ML model is selected across various trained models.

Validating Models Across Market Segments

ML models are vulnerable to poor data quality. The importance of a model reduces over time; then the model needs to be retained. To do this, a training pipeline is required.

Once a model is trained, it is evaluated with a testing dataset (which may or may not be a subset of the training dataset). The main purpose of using the testing dataset is to check the generalization ability of the trained model.

Monitoring

Changes in data can cause models' performance to deteriorate over time; we need to ensure that our systems are monitoring and responding to this degradation.

Therefore, we need to monitor the performance of our models online, tracking summary statistics of data and sending notifications when values deviate from expectations. We could also potentially start a new iteration in the ML process. This online monitoring serves as a signal for a new experiment iteration and the re-training of the model on new data.

Automation

In many MLOps environments, most of the machine learning tasks are done by people. This includes data pre-processing, feature engineering, splitting data into small pieces for training and testing purposes, training models, etc.

Many data scientists waste time by creating chances of error and resolving them, which can be used for research and exploration by doing these procedures manually.

Continuous retraining to forestall model drift is often the entry point of MLOps automation. This can lead to automating data ingestion pipelines, model validation and testing, and more.

Conclusion

MLOps can help machine learning teams manage the complex maintenance and operation of models. This requires clear communication, setting up naming conventions, using automation to manage time-consuming routine tasks, and more. This can save time and encourage a company to develop more ML models.

References

For references and new developments in deep reinforcement learning please check out the links below:

1. Continuous Delivery for Machine Learning
2. Best Practices for MLOps and the Machine Learning Lifecycle
3. MLOps Principles

Authored by
Anmol Singh,
Consultant at Absolutdata

Model Card for MLOps

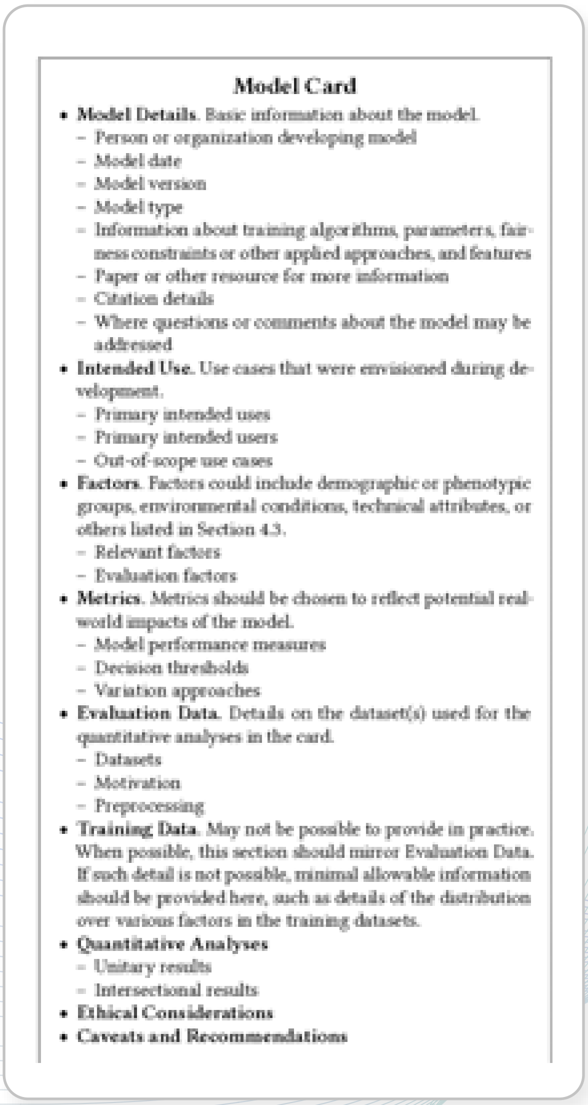
Vivid Visualization

Model cards are designed to give a clear, comprehensive picture of a Machine Learning model. A model card describes what the model does, who its target audience is, and how it is kept up. It also offers information about the model's architecture and the training data that were used in its creation. A model card not only includes raw performance information but also contextualizes the model's constraints and risk reduction options.

Model cards are documents that come with trained machine learning models and offer benchmarked evaluation in a range of circumstances. The context in which models are intended to be used, specifics of the performance evaluation processes, and other pertinent information are also disclosed on model cards.

Model Cards Contain:

- Model Details – Answers basic inquiries about the model version, types, and other specific details.
    1. Person or organization developing model: What person or organization developed the model? Stakeholders can use this to deduce information about potential conflicts of interest and model development.
    2. Model Date – When was the model deployed?
    3. Model Version – Which version is deployed? How does it differ from previous model versions?
    4. Model Type – This includes basic models like ANN, logistics, etc. without any hyper-parameter tuning.
- Intended Use - What the model should and should not be used for and the purpose for creating it.
    1. Primary uses - Whether the model was created for general or specific tasks.
- Factors - A overview of model performance across a range of pertinent criteria (such as groups, instrumentation, and environment) should ideally be provided.
    1. Relevant factors - The most important variables that will affect how well a model performs.
    2. Evaluation factors – Which factors are being reported. Checks whether factors are the same (and if they are, why).



Model Card

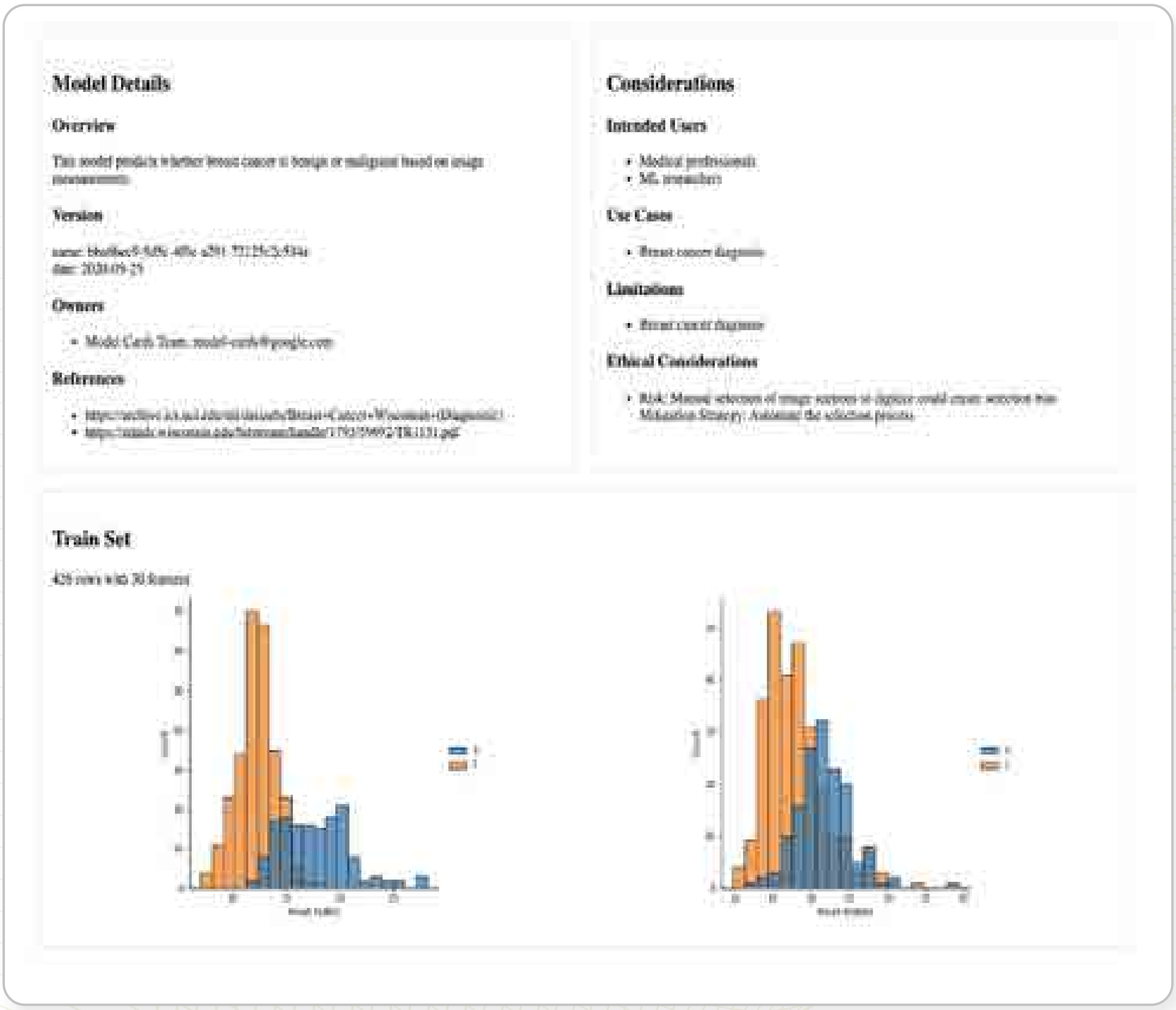
- **Model Details.** Basic information about the model.
    - Person or organization developing model
    - Model date
    - Model version
    - Model type
    - Information about training algorithms, parameters, fairness constraints or other applied approaches, and features
    - Paper or other resource for more information
    - Citation details
    - Where questions or comments about the model may be addressed
- **Intended Use.** Use cases that were envisioned during development.
    - Primary intended uses
    - Primary intended users
    - Out-of-scope use cases
- **Factors.** Factors could include demographic or phenotypic groups, environmental conditions, technical attributes, or others listed in Section 4.3.
    - Relevant factors
    - Evaluation factors
- **Metrics.** Metrics should be chosen to reflect potential real-world impacts of the model.
    - Model performance measures
    - Decision thresholds
    - Variation approaches
- **Evaluation Data.** Details on the dataset(s) used for the quantitative analyses in the card.
    - Datasets
    - Motivation
    - Preprocessing
- **Training Data.** May not be possible to provide in practice. When possible, this section should mirror Evaluation Data. If such detail is not possible, minimal allowable information should be provided here, such as details of the distribution over various factors in the training datasets.
- **Quantitative Analyses**
    - Unitary results
    - Intersectional results
- **Ethical Considerations**
- **Caveats and Recommendations**

Source: Semantic Scholar

- **Metrics -** Reflects the model's potential real-world effects. These primarily consist of model performance measures, decision thresholds, and the approach to uncertainty and variability.
- **Evaluation Data -** Documents that reveal the dataset's origin and contents must be publicly accessible to all. These may be new datasets or older ones that are provided along with the model card evaluations to allow for additional benchmarking (i.e., what datasets are used, the reason for choosing specific datasets, if there's any need to pre-process the data).
- **Training Data** – Generally, it contains as much information about the training data as the evaluation data. Nevertheless, it is quite possible this level of in-depth knowledge about the training data is not available.
- **Quantitative Analysis** – Displays the matric variation.
- **Ethical Consideration -** Presents stakeholders with examples of the ethical factors that were considered when developing the model. Major components include:

1. Data
2. Human life
3. Mitigation
4. Risk and harm

Diagram chart



Model Details

Overview

This model predicts whether breast cancer is benign or malignant based on image measurements.

Version

name: [illegible]
date: 2020-09-25

Owners

- Model Cards Team, model-cards@google.com

References

- https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
- https://minds.wisconsin.edu/bitstream/handle/1793/59692/TR1131.pdf

Considerations

Intended Users

- Medical professionals
- ML researchers

Use Cases

- Breast cancer diagnosis

Limitations

- Breast cancer diagnosis

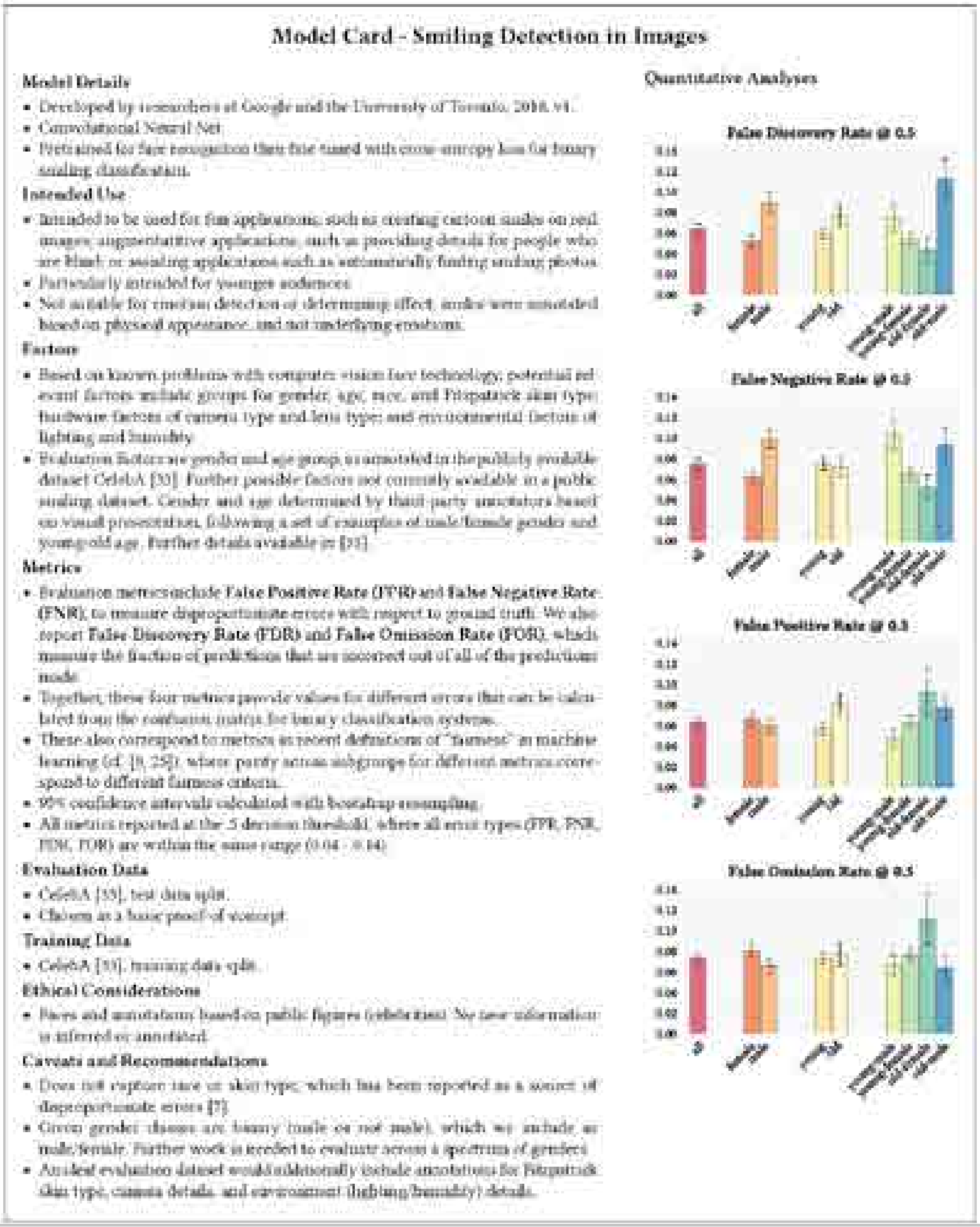
Ethical Considerations

- Risk: Manual selection of image sections to digitize could create selection bias
  Mitigation Strategy: Automate the selection process

Train Set

426 rows with 30 features

Source: Google Cloud



Model Card - Smiling Detection in Images

Model Details

- Developed by researchers at Google and the University of Toronto, 2018, v1.
- Convolutional Neural Net.
- Pretrained for face recognition then fine-tuned with cross-entropy loss for binary smiling classification.

Intended Use

- Intended to be used for fun applications, such as creating cartoon smiles on real images; augmentative applications, such as providing details for people who are blind; or assisting applications such as automatically finding smiling photos.
- Particularly intended for younger audiences.
- Not suitable for emotion detection or determining affect; smiles were annotated based on physical appearance, and not underlying emotions.

Factors

- Based on known problems with computer vision face technology, potential relevant factors include groups for gender, age, race, and Fitzpatrick skin type; hardware factors of camera type and lens type; and environmental factors of lighting and humidity.
- Evaluation factors are gender and age group, as annotated in the publicly available dataset CelebA [33]. Further possible factors not currently available in a public smiling dataset. Gender and age determined by third-party annotators based on visual presentation, following a set of examples of male/female gender and young/old age. Further details available in [33].

Metrics

- Evaluation metrics include **False Positive Rate (FPR)** and **False Negative Rate (FNR)**, to measure disproportionate errors with respect to ground truth. We also report **False Discovery Rate (FDR)** and **False Omission Rate (FOR)**, which measure the fraction of predictions that are incorrect out of all of the predictions made.
- Together, these four metrics provide values for different errors that can be calculated from the confusion matrix for binary classification systems.
- These also correspond to metrics in recent definitions of "fairness" in machine learning (cf. [9, 25]), where parity across subgroups for different metrics correspond to different fairness criteria.
- 95% confidence intervals calculated with bootstrap resampling.
- All metrics reported at the .5 decision threshold, where all error types (FPR, FNR, FDR, FOR) are within the same range (0.04 - 0.14).

Evaluation Data

- CelebA [33], test data split.
- Chosen as a basic proof-of-concept.

Training Data

- CelebA [33], training data split.

Ethical Considerations

- Faces and annotations based on public figures (celebrities). No new information is inferred or annotated.

Caveats and Recommendations

- Does not capture race or skin type, which has been reported as a source of disproportionate errors [7].
- Given gender classes are binary (male or not male), which we include as male/female. Further work is needed to evaluate across a spectrum of genders.
- An ideal evaluation dataset would additionally include annotations for Fitzpatrick skin type, camera details, and environment (lighting/humidity) details.

Quantitative Analyses

False Discovery Rate @ 0.5
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
False Negative Rate @ 0.5
False Positive Rate @ 0.5
False Omission Rate @ 0.5

Source : Semantic Scholar

References

- Margaret Mitchell et al.: Model Cards for Model Reporting
- Google Cloud: About Model Cards

Authored by
Tejashvi Anand,
Consultant at Absolutdata

Best Practices in Data Science

Thriving Traction

Given how much of our world is fueled by data, we may understandably wonder what data science's future will hold. While it's difficult to predict with any certainty what the future's defining innovations will be, Machine Learning appears to be of the utmost importance. Data scientists are looking for new ways to leverage Machine Learning to create more advanced, self-aware AI.

In a world that is more linked than ever, experts forecast that AI will be able to comprehend and communicate with humans, autonomous vehicles, and automated public transportation systems with ease. Data science will make this new world conceivable. The capacity for sales and marketing teams to thoroughly understand their customer is one of the most talked-about advantages of data science. An organization may design the finest customer experiences by using this information. The healthcare, financial, transportation, and defense industries are anticipated to undergo a revolution as a result.

Emerging Data Science Best Practices

1. Choosing a data versioning tool

- **Data modality:** Based upon the preview of the data.
- **Practicality:** Whether the tool can be implemented in the project workflow.
- **Comparison of the data sets:** Data needs to be in a format that can be easily compared to the data sets' flow.
- **Infra integration:** Whether the tool can work within with the Infrastructure or data modeling workflow.
- Examples of data version control tools:

  a. Neptune

  b. Pachyderm

  c. LakeFS

  d. DVC

  e. Dolt

  f. Delta Lake

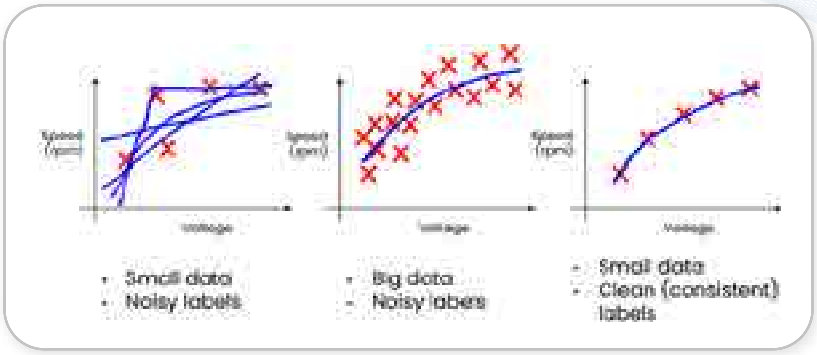
  g. Git LFS

2. Making data quality systematic

The most important factor is data consistency. We must utilize methods that enhance the data quality and enable models to perform well. We must also maintain the code while continually improving the data.

- Request sample labeling by two independent labelers.
- Compare labelers' consistency to identify areas of disagreement.

- When labelers disagree about a class, the labeling guidelines should be changed until everyone agrees.
- Consistency in labeling and small data is paramount.



3. From Big Data to Good Data

MLOps' most important task is to ensure constant high-quality data over the whole lifespan of the project. Good data has:

- A consistent definition.
- Clear labels.
- Good coverage of main cases.
- Coverage for concept and data drift.
- Timely feedback.
- The appropriate size.
- AI system = Code + Data.

4. Iterative evaluations

While the traditional software development process may end with functionality deployment (without taking into consideration update cycles), this is not the case with data-centric ML initiatives. The ML model in production will use data that it has never seen before; this data will certainly be different from the algorithm's training data.

As a result, the quality of the model should be evaluated throughout the process. It should not be seen as a last step. A requirement for online learning is the timely input from production systems, which, for instance, enables the identification and response to distributional data drifts.

5. Data augmentation

Different techniques for boosting the quantity of sample data points include creating new images by flipping, rotating, zooming, or cropping existing ones; similar methods exist in language processing. Data augmentation is used in data-centric Machine Learning to enhance the quantity of pertinent data points, such as the number of faulty production components

Businesses must change and expand along with AI technology, which is never static. Data quality is a key component of the move from model-centric to data-centric AI. A new kind of Artificial Intelligence technology called data-centric AI (DCAI) is devoted to comprehending, exploiting, and reaching conclusions from data. Prior to data-centric AI, AI relied mostly on rules and heuristics. While they might be helpful in some circumstances, they frequently produce less-than-ideal outcomes or even mistakes when used with new data sets.

By adding Machine Learning and Big Data analytics tools, data-centric AI enables systems to learn from data rather than depending on algorithms. They can thus make wiser choices and deliver more precise outcomes. Additionally, this method has the potential to be far more scalable than conventional AI. As data sets get bigger and more complicated, data-centric AI will become more significant.

References

- Neptune AI: Best Data Versioning Tools
- Dataquest: Evolution of Data Science Growth
- IntechOpen: Best Practices in Accelerating the Data Science Process
- DataCentricAI.org: What Is Data Centric AI?
- Brown, Sara: Data Centric AI in Demand

Authored by

Piyusha Rani,

Senior Program Engineer at Absolutdata

Busting MLOps Myths

Folk-Wisdom's Fallacy

With its ever-increasing traction in ML-backed services, MLOps has become a necessity. And this is with a good reason, as AI projects face possibly never being deployed and consequently never generating actual business values. Hence, various practices are being implemented at the development and operations level to scale ML (Machine Learning) projects.

The pace at which technologies and concepts are emerging is seldom so worrisome that we tend to often misunderstand the crux along with the applications of MLOps in general. So let's debunk some of the common myths around MLOps!

Myth 1: Development readiness and production readiness are the same.

Reality: MLOps provides a bridge between model development and model deployment.

Most of the developments around ML are done within an environment that's visible and accessible to the developers. As important as the performance of the models is, it is equally important to develop the project to maximize portability. Unfortunately, when the projects are taken to production, a massive amount of work often still remains. Subsequently, the models end up in a queue and deployment is compromised.

Deploying ML projects into production is a critical task, since data scientists start by experimenting with data in their native environments. Up till the production stage, many artifacts are generated that also need to be put into production systems. Additionally, these artifacts should be kept up to date for real-time predictions and scoring; hence, a to-and-fro connection is formed. Thus, MLOps is more about facilitating and streamlining both lab and production environments.

Myth 2: MLOps is all about models and metrics, not pipeline

Reality: Modelling is just a subset of a list of work that has to be done.

There are multiple crucial parts beyond model development – setting up environment, collecting the data, configuration, extracting features, data auditing and verification, resource management, infrastructure serving, and governance and monitoring.

All of these processes come together to form a production environment and get the models operationalized. Similarly, during monitoring, model accuracy comes second to pipeline and service health, including data and model drift detection.

Models and metrics play a vital role in research and development and are nice to have in monitoring. But looking from a deployment lens, various factors supersede (such as service response times, SLAs, throughput, etc.) that are essential for stable functioning.

Myth 3: Stack Overflow can fix all errors

Reality: Fixing production model errors require prior planning and fallbacks.

We all agree that almost all errors or bugs have already been discussed on Stack Overflow. But have you ever wondered why no one asks about production-level bugs or errors? That is because production bugs are very subjective; you won't be getting plain solutions anywhere. Such problems can arise due to various reasons – model drift, data mismatch, infrastructure fallacies, etc. This is why laying down a plan during deployment and development to make robust models and pipeline robust. This can include:

- Keeping a backup or a baseline model ready to swap.
- Having default values that can override / fill in while using rules.
- Including audit trails, event logs, etc. as much as possible to find issues within pipelines.
- Any maintenance / upgrades should not interrupt downstream services.

Myth 4: You don't need MLOps if you have AI governance

Reality: MLOps is distinct and can help support governance objectives.

While it is fairly correct to say that MLOps and AI governance are related, there are certain differences. The primary focus of AI governance is to regulate compliances and manage risks associated with machine learning. MLOps is mostly concerned about the uptime of services within the production systems, ensuring that models are delivering the level of performance and desired high-quality results.

The following intersections help distinguish what their relative objectives of AI governance vs. MLOps:

Access Control – Limiting access only trained operators vs. minimizing downtime to regulate compliance.

Audit Trails – Using trails to demonstrate compliance vs. using them for troubleshooting and process improvement.

Failover Plans – Using plans to remedy actual breakages vs. using plans to keep the system operational.

References

1. VentureBeat: 7 MLOps myths debunked
2. DataTechVibe: MLOps myths that are hampering your productivity
3. Towards Data Science: What is MLOps and why should we care?

Authored by

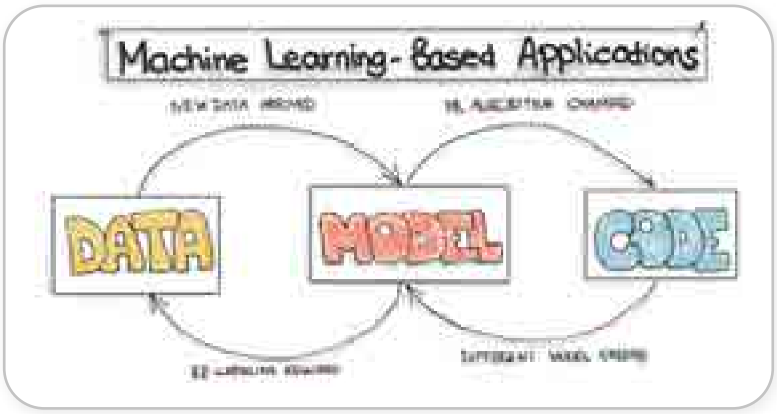
Avesh Kumar Verma,

Analyst at Absolutdata

MLOps Planogram

Experience Extended

There's a primary difference between machine learning and deep learning applications and other types of software applications: changes can come from multiple sources in ML. Traditional software changes only come from its code; ML application changes can come from the code, the model, or even the data. Thus, using purely Agile methods isn't ideal for ML.



Source: MLOps: Motivation

This is where MLOps come in. It incorporates ML and data as key players in the ecosystem, rather than just as variants of traditional software components. Let's look at a planogram of an ML project for details.

Levels of Enablement

This planogram includes a mix of generic and peculiar fragments where MLOps is integrated to:

- Unify the development cycle.
- Automate artifact gathering and testing.
- Enable continuous integration, training, etc.

Below is the high-level mapping of fragments and MLOps as an ingredient.

Project Fragment	MLOps ingredient
Infrastructure & Configuration	Infra-as-Code (IoC)
Data Engineering Operations	Continuous Integration
Machine Learning Operations	Continuous Integration, Training, and Testing
Deployment & Monitoring	CX

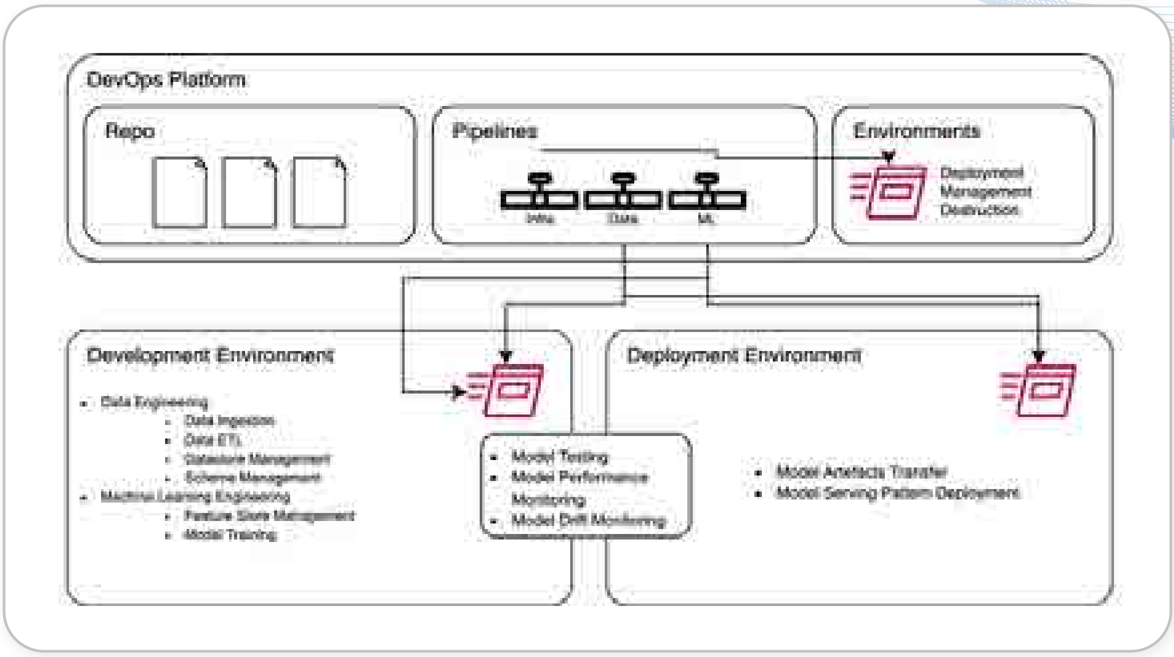
On-Ground Operations

It is always good to map theoretical logic to on-ground operations. Below we can see what on-ground operations are being performed inside a theoretical MLOps ingredient:

- **MLOps theoretical ingredient**
  - Project fragment
    - On ground operation
- **Infrastructure**
    - Deployment
    - Management
    - Destruction
- **Continuous Integration**
  - Data Engineering
    - Data Ingestion
    - Data ETL
    - Datastore Management
    - Schema Management
  - Machine Learning Engineering
    - Feature Store Management
    - Model Training
- **Continuous Testing**
    - Model Testing
    - Model Performance Monitoring
    - Model Drift Monitoring
- **Continuous Deployment**
    - Model Artefacts Transfer
    - Model Serving Pattern Deployment

Physical Lens

A next-generation text stack is a must for speedy MLOps implementation. For the planogram, we have a live-time implementation of the following tech stack:



Source: Absolutdata

Conclusion

1. To test, deploy, manage, and monitor ML models in actual production, we must build best practices that account for machine learning and AI in software products and services.
2. Implementing MLOps helps us prevent technical debt in Machine Learning applications.
3. Data, models, and other ML assets must be given the same status as other "traditional" elements in the software development lifecycle.
4. Machine Learning models should be included in a unified release process.

References

1. ml-ops.org

Authored by

Sumit Tyagi,

Data Scientist at Absolutdata

XOps: All the Ops Under One Umbrella

Food for Thought Experiment

Ops stands for Operationalization. It aims to align IT with business priorities and streamline the process from product development to end deliverables. It also attempts to reduce the time in whole operational processes in the software industry.

Industry has already been introduced to the facility of Ops. With the introduction of the Cloud, Big Data analytics has become far more complicated and dynamic. DevOps, MLOps, ModelOps and DataOps are adding great value there. But XOps is uniting all the Ops under one umbrella.

According to Gartner Top 10 Data and Analytics Trends of 2021, "the goal of XOps (data, machine learning, model, platform) is to achieve efficiencies and economies of scale using DevOps best practices — and to ensure reliability, reusability and repeatability while reducing the duplication of technology and processes and enabling automation". [1]

Why XOps Is Taking Off

The industry is going through speedy digital and AI transformations. A variety of use cases are emerging in the field of software and Machine Learning. The backbone of these digital solutions is data. To handle Big Data, complex engineering processes are being employed to ensure efficiency and speed.

But enterprises are facing scalability and operational challenges. Systematically productionizing the pipeline has become a major problem – one for which XOps has been introduced as reliable and scalable solution.

DataOps: Industrializing Data and Analytics

As one DataOps practitioner from a Fortune 50 company says, "DataOps consists of a stream of steps required to deliver value to the customer. We automate those steps where possible, minimize waste and redundancy, and foster a culture of continuous improvement." [2]

DataOps stems from DevOps and Agile practices. It emphasizes the use of short development sprints and self-organizing teams with business involvement. It uses version control systems and code repositories to include parallel development and increase efficiency.

Data Engineers: Leading the XOps March Forward

XOps largely deals with setting up infrastructure for data ingestion and production. Data engineers are the main players in this arena. According to Inside Big Data, "data engineering was the fastest-growing job of 2019, increasing by 50% year-over-year. Fast forward to today and data engineering opportunities are continuing to outpace data scientist roles. Data engineering is at the frontier of the data revolution." [3]

The growth of and need for XOps is fundamentally restructuring the role of data engineers. Creating stable and fast Cloud infrastructures is one of their main focuses.

References

1. Gartner Top 10 Data and Analytics Trends for 2021
2. Eckerson Group: Data Ops: Industrializing Data and Analytics
3. Inside Big Data: XOPs: The Rise of Smarter Tech Operations.

Authored by

Rohan Garg,

Consultant at Absolutdata

Courses:

1. MLOps (Machine Learning Operations) Fundamentals

- Identify and use core technologies required to support effective MLOps.
- Adopt the best CI/CD practices in the context of ML systems.
- Configure and provision Google Cloud architectures for reliable and effective MLOps environments.
- Implement reliable and repeatable training and inference workflows.

Link : https://www.coursera.org/learn/mlops-fundamentals#about

2. Machine Learning Engineering for Production (MLOps) Specialization [Deeplearning.AI]: series of 4 courses

What you will learn

- Design an ML production system end-to-end: project scoping, data needs, modeling strategies, and deployment requirements
- Establish a model baseline, address concept drift, and prototype how to develop, deploy, and continuously improve a productionized ML application.
- Build data pipelines by gathering, cleaning, and validating datasets. Establish data lifecycle by using data lineage and provenance metadata tools.
- Apply best practices and progressive delivery techniques to maintain and monitor a continuously operating production system.

4 courses:

- Introduction to Machine Learning in Production
- Machine Learning Data Lifecycle in Production
- Machine Learning Modeling Pipelines in Production
- Deploying Machine Learning Models in Production

Time: 4 months (5 hours/week)

Level: Advanced

Link: ML ops Specialization

3. Become a Machine Learning Engineer for Microsoft Azure: Udacity

In this program, students will enhance their skills by building and deploying sophisticated machine learning solutions using popular open source tools and frameworks, and gain practical experience running complex machine learning tasks using the built-in Azure labs

Estimated time: 3 Months

Link : Become a Machine Learning Engineer

Research Papers

- A Data Quality-Driven View of MLOps
- Ease.ML LifecycleManagement system for ML
- MLOps - Definitions, Tools and Challenges

Repositories:

Awesome Mlops : An awesome list of references for MLOps - Machine Learning Operations

Authored by

Ankit Tyagi,

Consultant at Absolutdata

infogain
Engineering Business Outcomes
+
ABSOLUTDATA

Thank You

For reading this edition of BrainWave from the NAVIK & Data Science Team. This digest focuses on some technical angles of analytics and data science. BrainWave is published about 4 times a year. If you haven't already, please subscribe so you receive future editions.

Subscribe