WHAT'S CD4ML? CONTINUOUS DELIVERY WITH MACHINE LEARNING EXPLAINED



Applying data science to your continuous delivery (CD) model is key to gain real world outcomes.

In a world where CI/CD is king, the growing demands for continuous rollouts of software need to incorporate machine learning and data-driven principles. Enterprises need to move beyond DevOps principles which are supported by data science…

Instead, we are looking at Machine Learning Operations (MLOps).

Continuous integration and continuous delivery need to be supported by machine learning models in a new type of life cycle. By integrating creating an environment that combines CI/CD and machine learning—known as CD4ML—every step that your business takes is powered by data. When you know where you want to go, data builds the path to get there. When you need to understand where you are, data informs you how to strengthen your position.

But it's not an easy process. Integrating data, data scientists, and data model training comes with plenty of challenges, mainly relating to cost and the undefined amount of time it can take the get a working data model. Can ML models improve your continuous delivery business needs? Let's find out.

How continuous delivery works with machine learning

Continuous delivery with machine learning requires a different business workflow. Although we are all used to DevOps processes, the practices are at risk of becoming inefficient if data cannot be properly integrated. A DevOps team with a data engineer who consults a model does not make a

CD4ML team.

What is CD4ML?

Continuous Development for Machine Learning is a new software engineering approach. The principles and practices of CD4ML are based on machine learning and data models to inform traditional software development processes, training data to make the deployment pipeline both more agile and more user focused.

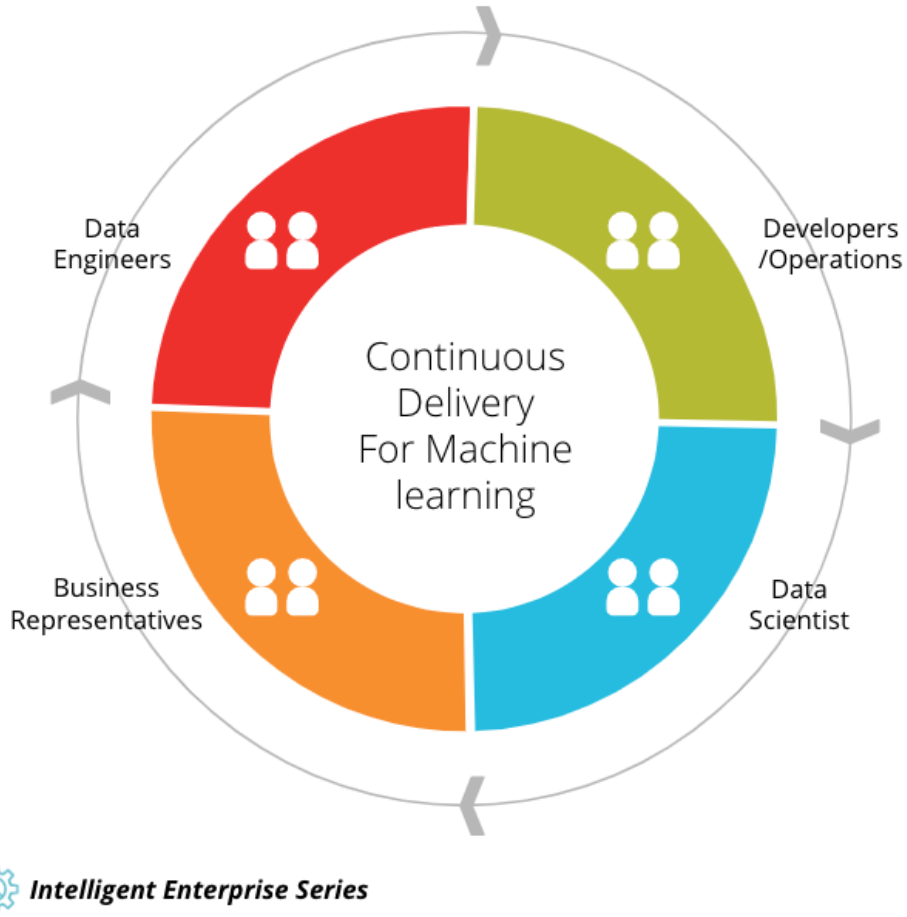
Continuous development and continuous integration feeds off data taken from the teams responsible for data collection and management. New data becomes the central concern for machine learning engineers and software development professionals alike.

Structure of CD4ML

CD4ML needs four key stages in order to work:

1. **Data scientists** collect, analyze, and present hypotheses to the development team.
2. The **DevOps team** integrates the findings into existing applications and services.
3. **Data engineers** continuous analyze and update the data in order to optimize the data model that is being used.
4. **Business representatives** present outcomes to the data scientists who then collect and explore new data to help fit the desired software development needs.

This process does not happen as a linear pipeline. This is a constant, cyclical process that needs business representatives to feed outcomes back to the scientists exploring the data. This completes the cycle and allows for continuous development and continuous deployment with the desired and necessary features.



Each stage of the CD4ML cycle refers to data and business outcomes to acquire real world outcomes. (Source: ThoughtWorks)

Comparing CD4ML vs DevOps

Within a machine learning informed production environment, the software development process is driven by data sets and ML engineers. Data engineering informs data models in production, allowing for development cycles to constantly refer to data scientist-driven information.

Pipelines in an MLOps team work by constantly feeding new information to the next stage. This allows for model development, model monitoring, informed development, and recalibrating goals and desired outcomes that underpin the ML system.

Is CD4ML a better approach than DevOps?

DevOps is widely adopted, but now is the time to move your production teams into the next paradigm. When implemented correctly, embedding data analysis and implementation into your software development workflow allows for:

- Development goals centered on user needs
- A cross-functional team
- Better version control

Of course, it is not all positive. In fact, CD4ML is generally prohibitively expensive for SMBs due to hiring costs and the general upkeep of employing an entire data team to integrate with your DevOps

pipeline. Is it worth it? If you're a small-sized operation, you may think not.

CD4ML benefits & drawbacks

Benefits of CD4ML	Drawbacks of CD4ML
User data is analyzed and implemented thanks to data scientists building models that explain user intentions and wants.	**The cost of a data science team** seriously limits the number of businesses that are able to hire enough data engineers and scientists to implement CD4ML
A cross-functional team brings more skillsets to the development process, refining and improving the end product with many expert perspectives.	**Initial development is no faster** as the data team can only work when there is actual data to use
Better version control is achieved through incrementally adding small pieces of code to existing software based on data-driven insights.	**Building effective data models is a slow process**, meaning that a business who wants to implement CD4ML will not see results until the data team has effectively trained the model
Shorter development cycles are measured in days due to the data science team feeding smaller-sized packages into the development process, speeding up production and allowing for better version control	

How does CD4ML improve business?

Depending on the size of your business, Continuous Delivery for Machine Learning allows you to integrate and use specific data to improve your software development process. This allows for better software rollouts as the continuous delivery and deployment of software is defined by the data that you collect from your users.

Implementing CD4ML in your business's workflow creates a better defined approach to updates and improvements. Data around your users needs to be analyzed and carry considered before it can properly be used for improving development. That's why you need both data scientists and data engineers:

- **Data scientists** clean and prepare your data before the developers can implement changes.
- **Data engineers** keep that data up to date and relevant to your business needs.

Although DevOps approaches are incredibly popular throughout the software world, changing to an MLOps approach will allow your business to take the next step. Your organization builds data into the development process, speeding up production and pin-pointing exact solutions to business problems as they are raised.

If you can afford to expand your process to include expansive data analysis teams, why wouldn't you integrate it into your DevOps pipeline?

Related reading

- BMC DevOps Blog
- BMC Machine Learning & Big Data Blog
- How To Use Jupyter Notebooks with Apache Spark
- Data Architecture Explained: Components, Standards & Changing Architectures
- Continuous Operations Explained