International Journal of Machine Learning and Cybernetics (IJMLC)
Volume 1, Issue 1, Jan-Dec 2023, pp. 25-34, Article ID: IJMLC_01_01_003
Available online at https://iaeme.com/Home/issue/IJMLC?Volume=1&Issue=1
Journal ID: 60X6-Y132 DOI: https://doi.org/10.17605/OSF.IO/UV532

LLMOps: EVALUATING AND FINE TUNING LLM MODELS FOR GENERATIVE AI

Amreth Chandrasehar
Informatica, CA, USA

ABSTRACT

As companies adopt Generative AI using LLM models, many pre-trained and fine-tuned models needs to be evaluated for its accuracy. LLMOps is a derivative of MLOps but specialized on model training and finetuning LLMs. Model evaluation process can take a lot of time and cost a fortune for companies, with LLMOps using model CI pipelines, frameworks and automation, the drawbacks are addressed and help organizations evaluate models quickly and, in a cost, optimized manner. This paper discusses how the pipelines, frameworks, Observability metrics collected during training can be used to optimally evaluate LLM models.

Keywords: Generative AI, LLM, MLOps, LLM Ops, Cost Optimization, Observability, Model CI

Cite this Article: Amreth Chandrasehar, LLMOps: Evaluating and Fine Tuning LLM Models for Generative AI, International Journal of Machine Learning and Cybernetics (IJMLC), 1(1), 2023, pp. 25-34
https://iaeme.com/Home/issue/IJMLC?Volume=1&Issue=1

1. INTRODUCTION TO LLMOPS

Machine Learning Operations (MLOps) is the discipline of operationalizing machine learning (ML) systems. It encompasses the practices and tools that ensure ML systems are developed, deployed, and managed effectively in production. MLOps is essential for organizations that want to derive business value from ML. Without MLOps, ML projects are often delayed, over budget, and fail to meet expectations.

LLMOps stands for Large Language Model Operations. It is a specific type of MLOps that is focused on the operationalization of large language models (LLMs). LLMOps encompasses the practices and tools that ensure LLMs are developed, deployed, and managed effectively in production. LLMOps involves steps of Data preparation, Model training, Model deployment, Model monitoring and Model maintenance. Each of them is discussed in below sections.

LLMOps is a complex discipline, but it is essential for organizations that want to derive business value from LLMs. By following the LLMOps process and using the right tools, organizations can ensure that their LLMs are developed, deployed, and managed effectively in production. Key principles are Planning and design, development and training, evaluation and deployment, monitoring and maintenance

Organizations use LLMs one of the 3 ways - Prompt a general-purpose LLM, Finetune a general-purpose LLM or Train a Custom LLM. This paper will focus on the last two usage types.

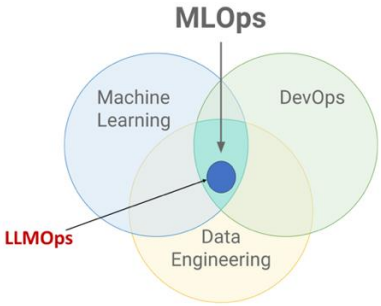

Figure 1: LLMOps [Wikipedia]

Training LLMs might differ from the traditional MLOps approach, LLMOps focus more on fine-tuning models, model monitoring and data quality. Below are some of the differences compared to MLOps [9]:

- Computational Resources: Training and fine-tuning large language models typically involves performing orders of magnitude more calculations on large data sets. To speed this process up, specialized hardware like GPUs is used for much faster data-parallel operations. For the traditional ML Models CPU resources might be sufficient.
- Transfer Learning: Unlike many traditional ML models that are created or trained from scratch, many large language models start from a foundation model and are fine-tuned with new data to improve performance in a more specific domain. Fine-tuning allows state-of-the-art performance for specific applications using less data and fewer compute resources.
- Human Feedback: One of the major improvements in training large language models has come through reinforcement learning from human feedback (RLHF). More generally, since LLM tasks are often very open-ended, human feedback from your application's end users is often critical for evaluating LLM performance. Integrating this feedback loop within LLMOps pipelines can often increase the performance of your trained large language model.
- Hyperparameter Tuning: In classical ML, hyperparameter tuning often centers around improving accuracy or other metrics. For LLMs, tuning also becomes important for reducing the cost and computational power requirements of training and inference. For example, tweaking batch sizes and learning rates can dramatically change the speed and cost of training. Thus, both classical ML and LLMs benefit from tracking and optimizing the tuning process, but with different emphases.
- Performance Metrics: Traditional ML models have very clearly defined performance metrics, such as accuracy, AUC, F1 score, etc. These metrics are straightforward to calculate. When it comes to evaluating LLMs, however, a whole different set of standard metrics and scoring apply — such as bilingual evaluation understudy (BLEU) and Recall-Oriented Understudy for Gisting Evaluation (ROGUE) that require some extra considering when implementing.

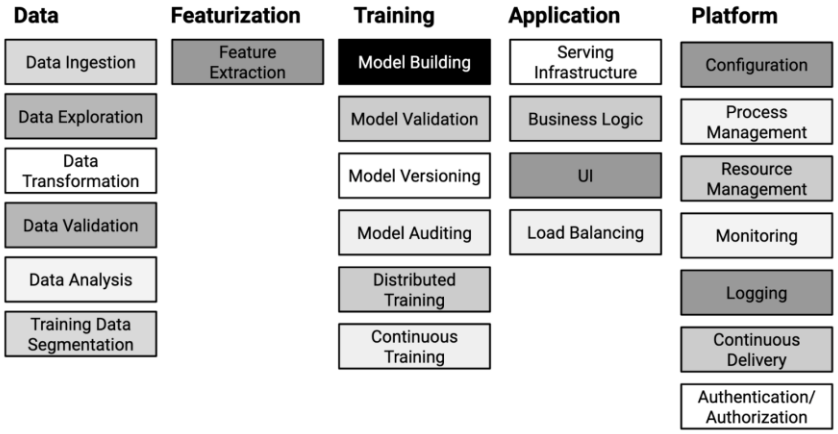

Figure 2: Building blocks of MLOps and LLM Ops [9]

2. MODEL CI

MLOps is an ML engineering culture and practice that aims at unifying ML system development (Dev) and ML system operation (Ops). Practicing MLOps means that you advocate for automation and monitoring at all steps of ML system construction, including integration, testing, releasing, deployment and infrastructure management.

Data scientists can implement and train an ML model with predictive performance on an offline holdout dataset, given relevant training data for their use case. However, the real challenge isn't building an ML model, the challenge is building an integrated ML system and to continuously operate it in production.

ML and other software systems are similar in continuous integration of source control, unit testing, integration testing, and continuous delivery of the software module or the package. However, in ML, there are a few notable differences:

- CI is no longer only about testing and validating code and components, but also testing and validating data, data schemas, and models.
- CD is no longer about a single software package or a service, but a system (an ML training pipeline) that should automatically deploy another service (model prediction service).
- CT is a new property, unique to ML systems, that's concerned with automatically retraining and serving the models.

Many organizations have ML engineers and data scientists, but the execution of below steps is fully manual. Each step is manually executed which can cause infrequent release cycles, slowing down features released to customers. This manual process does not have CI or CD or monitoring capabilities. Manual model process is shown below:

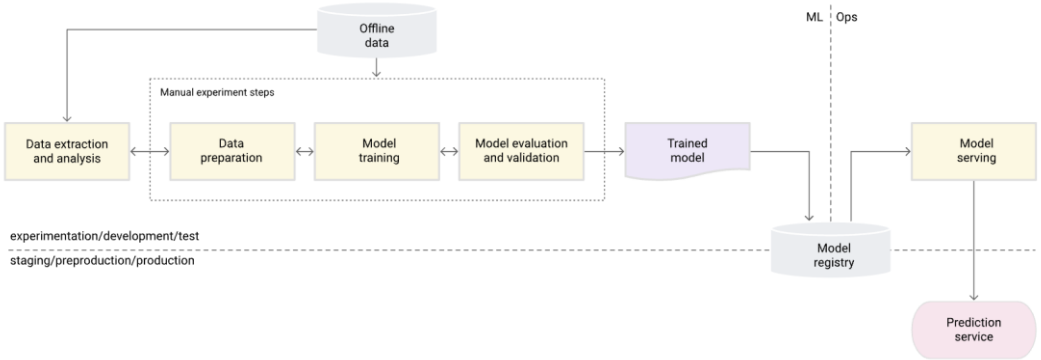

Figure 3: Manual Model process

Fully automated Model CI and CD pipelines are very important for the model development lifecycle. It helps organizations to push new features in models faster to production, monitor the models for accuracy, abuse and serve the model in production.

Below are some of the best practices to be followed:

a. Modularized code for components and pipelines: To construct ML pipelines, components need to be reusable, composable, and potentially shareable across ML pipelines. Therefore, while the EDA code can still live in notebooks, the source code for components must be modularized. In addition, components should ideally be containerized to do the following: [1]

    i. Decouple the execution environment from the custom code runtime.

    ii. Make code reproducible between development and production environments.

    iii. Isolate each component in the pipeline. Components can have their own version of the runtime environment and have different languages and libraries.

b. Use version control: Required to track changes to the model code, data, and configuration. It will help reproduce the results and models can be rolled back to a previous version if necessary.

c. Deploy models in stages: Deploy models in stages, starting with a small number of users and gradually increasing the number of users over time, instead of directly deploying in production. It will help to mitigate the risk of problems if the model does not perform as expected.

d. Monitoring models: Monitor the models to ensure it is performing as expected. Collecting metrics, logs and traces on the model's performance, accuracy and abuse is important.

e. Retrain models: Over time, the data the models are trained on may change. Ot can cause the models to be less accurate. To address, retraining models on new data on a regular basis is required.

Google's "MLOps: Continuous delivery and automation pipelines in machine learning" documentation provides and End to End pipeline of MLOps model training. It covers from data discovery to data preparation, model evaluation to uploading the artifact to an artifact store and then serving the model. In case of LLMs, if a SaaS LLM service is provided, manifest files are stored in repository.

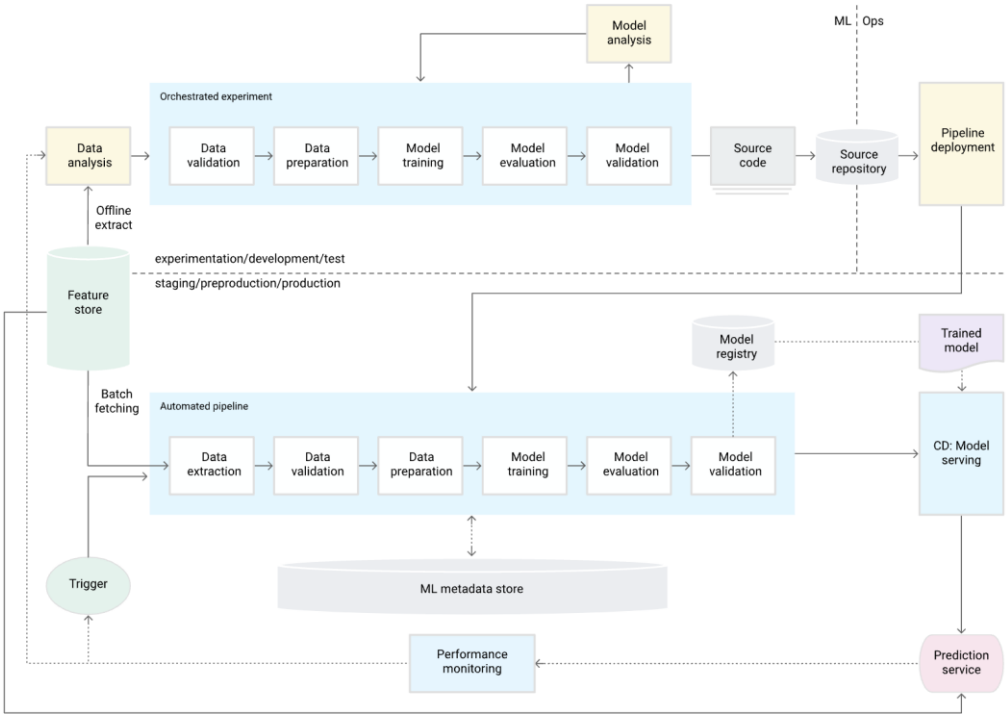

Figure 4: Automated Model CI and CD (Serving/Inference) [1]

The above flow diagram provides how a Model CI and CD should be created. There are various components that are required to build a successful LLMOps pipeline. The table below provides the explanations to each requirement in detail:

Sno	Requirement	Explanation
1	Model Registry	Centralized repository for storing and managing trained machine learning models. It enables to easily store, version, discover, and deploy models
2	Artifact Store	An artifact store is a place where machine learning models and their associated artifacts such as data, configuration files, and trained models can be stored and versioned.
3	Orchestration Engine	An ML orchestration engine is a tool that helps data scientists and engineers manage and automate the process of building, training, and deploying machine learning models.
4	Monitoring Store	System that collects, stores, and analyzes data from various sources in real-time. It is used to monitor and track the performance of models and their training process.
5	Governance	Policies, processes, and technologies used to ensure the responsible and ethical use of ML including Data Governance, Model Privacy, Compliance, Access Controls, Auditing
6	Feature Store	Feature Store is a centralized repository of features (or attributes) used in ML models. It provides a way to manage, share, and reuse features across different ML pipelines and models.
7	Model Deployment	Model deployment refers to the process of taking a trained machine learning model and making it available for use by applications and end-users.
8	Pipeline	A pipeline is a series of automated steps that process and transform data, and then train and deploy machine learning models.
9	Model Serving	Model serving refers to the process of making a trained ML model available for use in production environments.
10	ML Training	Process of preparing data, running experiments, training models using various algorithms and data to make predictions.
11	Notebook	Popular way for data scientists and machine learning engineers to explore data, experiment with models, and develop and test code.
12	Auto Scaling	Ability of a platform to automatically adjust the amount of computing resources allocated to a machine learning task based on the workload.

3. MODEL MONITORING AND EVALUATION

Model monitoring is the practice of collecting and analyzing data about machine learning models in order to understand how they are performing and to identify any potential problems. Model monitoring is essential to ensure the machine learning models are performing as expected in production. By monitoring the models, organizations can identify and fix problems early on, before they cause outages or other disruptions to production systems.

The performance of ML models starts degrading over time. It can be due to data inconsistencies, skews, and drifts, making deployed models inaccurate and irrelevant. Appropriate ML monitoring helps identify precisely when the model performance started diminishing. Model monitoring helps in Early detection of problems, Improved understanding of model performance, Proactive remediation and Improved decision-making, foster users' trust in ML systems. Model monitoring can be enabled using below ways:

- Amazon Sagemaker – Native models in Sagemaker can use Estimator SDK to collect metrics. The main drawback is we cannot use the SDK for models not native to Sagemaker. Alternatively, monitoring frameworks can be used to collect metrics
- Langchain + Langsmith – Frameworks to be enabled in CI pipeline and models to collect model metrics, including LLMs
- Grafana + Prometheus to monitor model metrics or ML platform metrics.

- Logs generated by models and ML platforms can be ingested into any monitoring solution to build analytical capabilities.

There are more tools in the market that can help with model monitoring, many commercial offerings have now come to offer the capabilities. More analysis needs to be done with these tools.

In general, below are the metrics that are required to evaluate the model:

- Model health, Performance Monitoring
- Drift Detection
- Quality Checks
- Expandability (understanding why the model makes the predictions that it does)
- Custom Alerts
- Ground Truth Updates
- NLP and Computer Vision Monitoring

LLM Models are benchmarked with below 5 metrics:

- ARC (AI2 Reasoning Challenge
- HellaSwag
- MML (Multilingual and Multimodal Understanding Evaluation)
- TruthfulQA
- Carbon footprint

These metrics helps to choose the right model to fit your company specific use cases. Other than these metrics, license terms need to be reviewed carefully to ensure we are compliant with terms and conditions of model usage.

Benchmark:	ARC	HellaSwag	MML	TruthfulQA
Measure of?	Model's ability to answer questions correctly hence It focuses on measuring the model's comprehension and reasoning capabilities. This averages scores of LLM on a number of different tasks.	Model's ability to predict plausible completions for a given prompt while avoiding completion choices that are unrealistic or nonsensical. It aims to evaluate the model's common-sense reasoning and contextual understanding.	Performance of models in multilingual and multimodal tasks. The MML score measures the model's ability to summarize text concisely and accurately and ability to perform a variety of tasks	Capability to provide reliable and factual information while avoiding incorrect or misleading responses. The TruthfulQA score measures the model's ability to avoid generating false or misleading information.
How its done?	AI2 Reasoning Challenge (25-shot) - a set of grade-school science questions.	HellaSwag (10-shot) - a test of commonsense inference, which is easy for humans (~95%) but challenging for SOTA models.	MML (5-shot) - a test to measure a text model's multitask accuracy, coverings 57 tasks including elementary mathematics, US history, computer science, law, and more.	TruthfulQA (0-shot) - a test to measure a model's propensity to reproduce falsehoods commonly found online.
Task type	Question answering	Creativity	Versatility	Trustworthiness
Example	Given the text "The cat sat on the mat.", what is the color of the cat?	Write a poem about a cat sitting on a mat.	Given the text "The cat sat on the mat.", summarize the text in 100 words.	Is the statement "The cat is a dog" true?
Strengths	Can be used to compare LLMs on a variety of tasks	Can be used to assess LLM's creativity	Can be used to assess LLM's versatility	Can be used to assess LLM's trustworthiness
Weaknesses	Computationally expensive	Ideal for creativity tasks	Subpar performance in creativity-oriented tasks	Subpar performance on other tasks

Figure 5: Automated Model CI and CD (Serving/Inference) [7]

Evidently has open-sourced Python library to evaluate, test, and monitor ML models from validation to production. It works with tabular, text data and embeddings. In below screenshot, we can see data drift, model performance and target drift metrics to monitor the LLM Models.

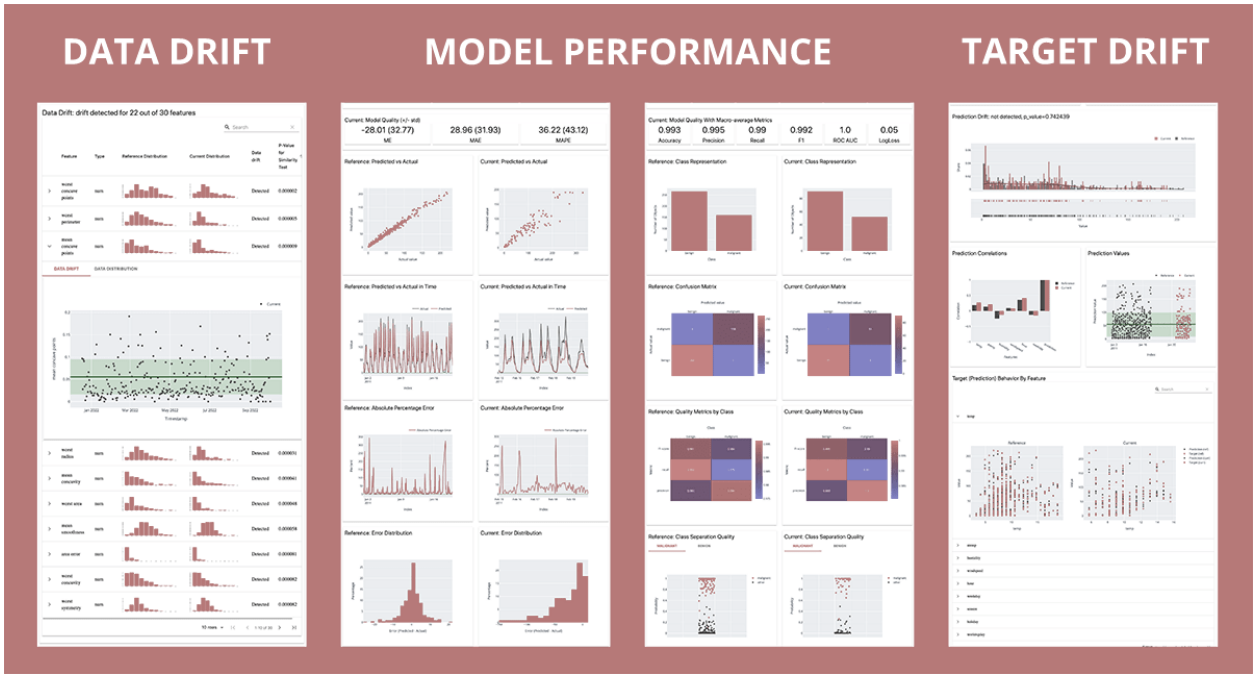

Figure 7: Reports calculate various data and ML metrics and render rich visualizations. [8]

4. ABUSE DETECTION AND IMPACT DUE TO BIAS IN AI

Abuse detection in LLM models is the process of identifying and mitigating the risk of abuse in LLMs. LLMOps should have abuse and bias detection in the pipeline to identify and alert the developers to fix the datasets. AI has the potential to deliver tremendous value to businesses and the overall economy but is proving to be problematic when it comes to accurately representing entire populations. Concern around AI bias has risen nearly as quickly as AI adoption itself. Companies that have experienced bias in their algorithms have lost revenue, customers, and employees as a result. It is critically important that AI is trusted and explainable to ensure AI models are fair and unbiased.

There are several techniques that can be used to detect abuse in LLM models:

- Text classification is a technique that classifies text into different categories, such as spam, hate speech, or CSAM and can be operationalized using machine learning algorithms.
- Topic modeling is a technique that identifies the topics that are discussed in a text and can be used to identify potential abuse, such as spam or hate speech, that is disguised as a different topic.
- Sentiment analysis is a technique that identifies the sentiment of a text, such as positive, negative, or neutral and can be used to identify potential abuse, such as hate speech.
- Named entity recognition is a technique that identifies named entities in a text, such as people, organizations, or places and can be used to identify potential abuse, like CSAM.
- Malware detection is a technique that identifies malware in a text and can be used to identify potential abuse, like phishing attacks, comparing the text to a database of known malware signatures.

Abuse in Large Language models can occur at many forms, below are some of the abuse categories:

- Hate speech: It is a form of communication that attacks a person, or a group based on race, religion, ethnicity, nationality, sex, disability, sexual orientation, or gender identity. It can be used to incite violence or discrimination and can create a hostile environment for people who are targeted.
- Discrimination: An unfair or prejudicial treatment of a person, or a group based on race, religion, ethnicity, nationality, sex, disability, sexual orientation, or gender identity. It can take many forms, such as denying someone a job or housing, or treating them differently in a social setting.
- Harassment: An unwanted or unwelcome behavior that is threatening, intimidating, or offensive. It can take many forms, such as verbal abuse, physical abuse, or cyberbullying.

- Spam: An unsolicited or unwanted electronic messages, such as emails or text messages. Spam can be used to spread malware, phishing attacks and other forms of abuse.
- CSAM: CSAM stands for child sexual abuse material. Any material that depicts or describes the sexual abuse of a child. It is illegal and can have a devastating impact on children.

Due to bias in LLMs, it impacts business in a very negative way. Bias can cause ethical issues, reputational damage, lost opportunities, lack of trust from users, regulatory and compliance problems to business. Below are some statistics from DataRobot report from "State of AI bias" survey:

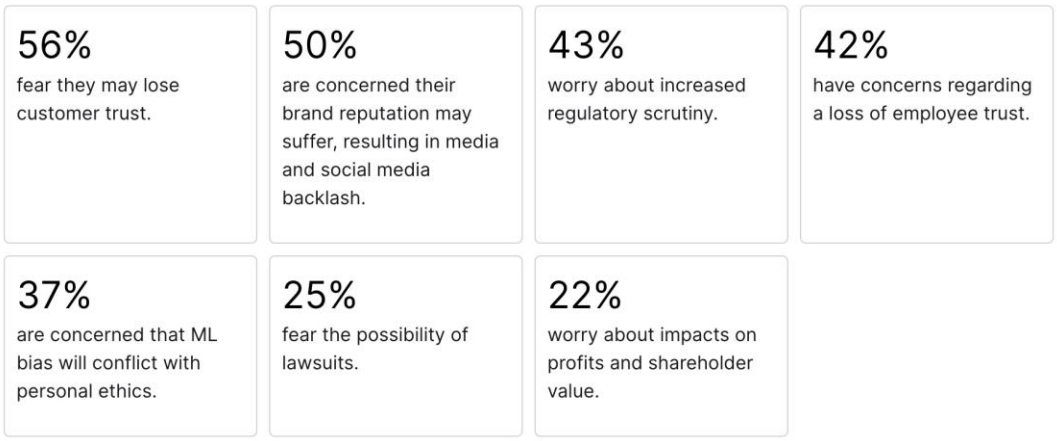

Figure 8: Automated Model CI and CD (Serving/Inference) [6]

5. RESULTS

Using MLOps and LLOps pipelines can significantly reduce the time taken to train and test the models for use in Production. The pipelines have helped to reduce the amount of time taken for testing by 75%. The ML practitioners surveyed who have deployed at least one ML model, half have already adopted DataOps and MLOps to some degree and 97 percent of those said they have achieved significant improvements by doing so [11]

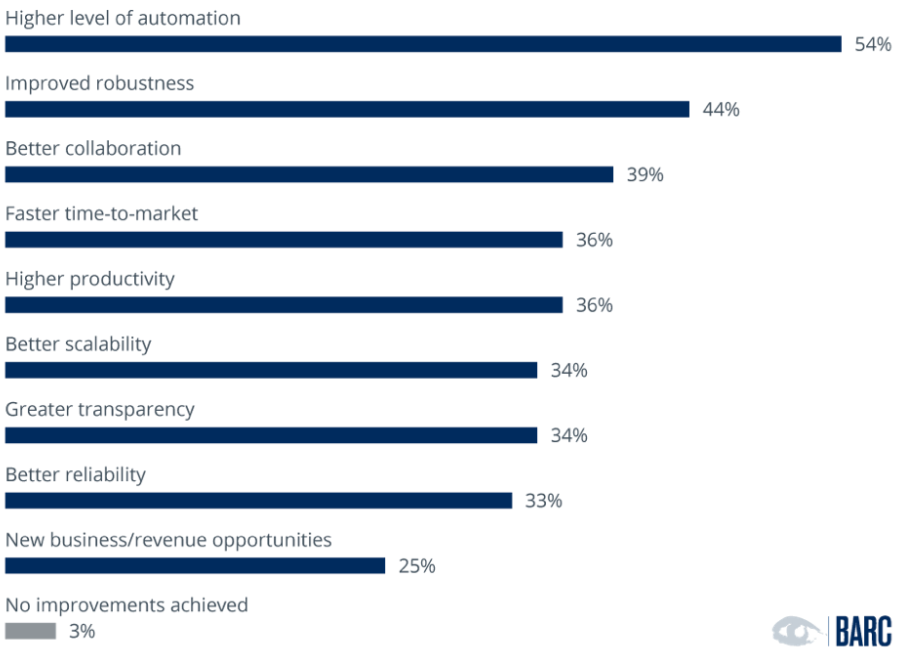

Figure 9: Significant improvements achieved with the introduction of MLOps [11]

6. CONCLUSION

In conclusion, The MLOps, LLMOps lifecycle offers a complete workflow to connect experimentation with the production environment. As part of this pipeline, short cycles of development can bring new features into production. Work packages for data engineers, data scientist and machine learning engineers are clearly defined. Early bias and abuse detection will help companies to build trust with customers and ensure no loss of revenue. Key to successful production usage of LLM models are right configuration of LLMOps pipelines. The results support significant improvements achieved with the introduction of MLOps in various organizations.

REFERENCE

[1] https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

[2] https://github.com/EleutherAI/lm-evaluation-harness

[3] https://en.wikipedia.org/wiki/MLOps

[4] Prasun Mishra, LLMOps: The Next Frontier of MLOps for Generative AI

[5] https://www.datarobot.com/wp-content/uploads/2022/01/DataRobot-Report-State-of-AI-Bias_V5.pdf

[6] https://www.fiddler.ai/ml-model-monitoring/model-monitoring-framework

[7] Prasun Mishra, Unlocking the Power of Open LLMs and Generative AI: A 10-Step Guide to Finding Your Perfect Language Model

[8] https://github.com/evidentlyai/evidently

[9] https://www.databricks.com/glossary/mlops

[10] https://github.com/sbueringer/kubecon-slides/blob/master/slides/2019-kubecon-eu/Tutorial%20Introduction%20to%20Kubeflow%20Pipelines%20-%20Michelle%20Casbon%2C%20Dan%20Sanche%2C%20Dan%20Anghel%2C%20%26%20Michal%20Zylinski%2C%20Google%20-%20kccnceu19_casbon_v1.pdf

[11] https://barc-research.com/press-release-dataops-mlops-ml-challenges/

Citation: Amreth Chandrasehar, LLMOps: Evaluating and Fine Tuning LLM Models for Generative AI, International Journal of Machine Learning and Cybernetics (IJMLC), 1(1), 2023, pp. 25-34

https://doi.org/10.17605/OSF.IO/UV532

Article Link:
https://iaeme.com/MasterAdmin/Journal_uploads/IJMLC/VOLUME_1_ISSUE_1/IJMLC_01_01_003.pdf

Abstract:
https://iaeme.com/Home/article_id/IJMLC_01_01_003

editor@iaeme.com