# CHAPTER 1

**INTRODUCTION**

In the data science world, machine learning (ML) is becoming a core approach for solving complex real-world problems, transforming industries, and delivering value in respective domains. Therefore, there are many data science teams in the scientific and machine learning communities trying to improve the overall business value using descriptive and predictive models. As a consequence, ML data scientists and ML operations engineer teams study how to apply DevOps principles to their ML systems under study.

DevOps is a set of practices and tools based on software and systems engineering. Software engineering can be defined as a discipline dedicated to developing tools and techniques which allow creation and use of sophisticated software systems. Data science is less about program development and more about analyzing and getting insights from the data. Agile, on the other hand, refers to an iterative approach which focuses on collaboration, customer feedback, and small and rapid releases. DevOps and Agile are two pillars to support in achieving business strategy and overlap the traditional operational and developmental teams to create an environment that is continually improving operations through a cross-functional team of developers and operators The DevOps strategic objective is to investigate methods for improving service quality and features in a manner that satisfies their customer needs ML data scientists and ML operations engineering teams have introduced manual steps for the delivery of ML pipeline model. This method may likely produce unexpected results due to the dependency on data collection, preparation and preprocessing, model training, validation, and testing. Moreover, this method led to the conclusion that no apparent advantage exists in utilizing our manual approach for ML projects. Also, in terms of quality results, this ML manual pipeline method produces high operational costs and delays, which directly or indirectly affect the revenue or quality reputation of the business.

## Overview

The entire MlOps model will be integrated with the important two devops principals the continuous Integration and continuous delivery practices. Practicing MLOps means that we advocate for automation and monitoring at all steps of ML system construction, including integration, testing, releasing, deployment, and infrastructure management.

The functionality of CI is not limited only for testing, and validating code and components, but also testing and validating data, data schemas, and models. CD is no longer about a single package or service, but an ML pipeline that should automatically deploy another ML service. The MLOps is communication between data scientists and the operations or production team. [It’s deeply collaborative in nature](https://insidebigdata.com/2018/04/30/next-generation-devops-ml-ops/), designed to eliminate waste, automate as much as possible, and produce richer, more consistent insights with machine learning. ML can be a game changer for a business, but without [some form of systemization](https://github.com/EthicalML/awesome-machine-learning-operations), it can devolve into a science. In nature which helps the organization to build and deploy model very quickly which brings business interest back to the forefront of your ML operations. Data scientists work through the lens of organizational interest with clear direction and measurable benchmarks overall which reduce cost and save time by creating more efficient workflows, Leveraging Data analytics for decision making, and improving customer experience Automating model development and deployment with MLOps means faster go-to-market times and lower operational costs. It helps managers and developers be more agile and strategic in their decisions. one of the most promising ways to improve the accuracy, fairness, and robustness of an ML model is often to improve the dataset, via means such as data cleaning, integration, and label acquisition. As MLOps aims to understand, measure, and improve the quality of ML models, it is not surprising to see that data quality is playing a prominent and central role in MLOps but also helps to reduce the human activity and completely automate the process some of the major applications of MLOps are in autonomous cars and various opencv projects. In finance sector for fraud detection and analysis of fraud transactions and automate the detection process faster and provide accurate results with intime when any fraud occurs.

## Organisation of Report

The report is organized into 5 chapters, starting with the chapter 1 provides a brief introduction to the proposed system and an overview of the proposed system. Chapter 2 mainly discusses about the literature survey which includes details about the papers and journals referred. It also discusses about the drawbacks and limitations in the existing system and advantages of using proposed system. Chapter 3 provides the working principle of the proposed system, It provides information about the different components used and their working in the system to yield the required. Chapter 4 provides merits and demerits of the proposed system. Chapter 5 provides information about the different applications of the proposed system.

# CHAPTER 2

**LITERATURE SURVEY**

## Related works

* + 1. **Description**

The entire MlOps model will be integrated with the important two devops principals the continuous Integration and continuous delivery practices. Practicing MLOps means that we advocate for automation and monitoring at all steps of ML system construction, including integration, testing, releasing, deployment, and infrastructure management.

The functionality of CI is not limited only for testing, and validating code and components, but also testing and validating data, data schemas, and models. CD is no longer about a single package or service, but an ML pipeline that should automatically deploy another ML service. The MLOps is communication between data scientists and the operations or production team. [It’s deeply collaborative in nature](https://insidebigdata.com/2018/04/30/next-generation-devops-ml-ops/), designed to eliminate waste, automate as much as possible, and produce richer, more consistent insights with machine learning. ML can be a game changer for a business, but without [some form of systemization](https://github.com/EthicalML/awesome-machine-learning-operations), it can devolve into a science. In nature which helps the organization to build and deploy model very quickly which brings business interest back to the forefront of your ML operations. Data scientists work through the lens of organizational interest with clear direction and measurable benchmarks overall which reduce cost and save time by creating more efficient workflows, Leveraging Data analytics for decision making, and improving customer experience Automating model development and deployment with MLOps means faster go-to-market times and lower operational costs. It helps managers and developers be more agile and strategic in their decisions. one of the most promising ways to improve the accuracy, fairness, and robustness of an ML model is often to improve the dataset, via means such as data cleaning, integration, and label acquisition. As MLOps aims to understand, measure, and improve the quality of ML models, it is not surprising to see that data quality is playing a prominent and central role in MLOps but also helps to reduce the human activity and completely automate the process some of the major applications of MLOps are in autonomous cars and various opencv projects. In finance sector for fraud detection and analysis of fraud transactions and automate the detection process faster and provide accurate results with intime when any fraud occurs.

## Drawbacks

* + - * One of the main obstacles for machine learning being deployed in production is the level of disruption that may occur with inclusion of ML in frontend technologies.
      * Operational mode, e.g., how new indicators are distributed, prioritized, and scan targets selected in a scalable manner or what a reliable strategy is to re-schedule scans to derive meaningful data on trends.
      * One of the major technical challenge is regarding the lack of coordination and improper handoffs between data scientists and operation teams which can lead to delays and errors.

# CHAPTER 3

**PRINCIPLE/WORKING**

## Machine Learning Lifecycle Methodologies

There are various methodology and methods for the development of machine learning lifecycle different methodologies may work better in different scenarios and data types. which stands for Cross-Industry Standard Process for Data Mining is the most commonly used approach by data mining experts and it was introduced in 1996 by Daimler Chrysler which stands for Sample, Explore, Modify, Model and Assess is a popular project methodology.

* + 1. **CRISP-DM Methodology**

CRISP-DM Methodology has typically got six iterative phases:

• Business Understanding Phase: In this phase, determine business objectives, assess the situation, establish data mining goals, and produce the project plan.

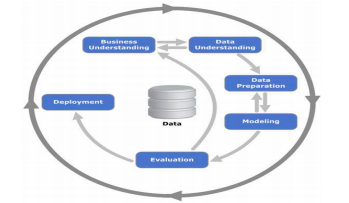
• Data Understanding Phase: In this phase, the initial data is available for the exploratory analysis, and evaluation of the data quality.

• Data Preparation Phase: In this phase, the preparation of data is a multistage process that comprises several individual steps. These steps are feature extraction, data cleaning, data reduction, data selection, and transformation.

• Modeling Phase: In this phase, the machine learning model is selected for the specific problem.

• Evaluation Phase: In this phase, the results can be processed by the selection of the ML model. Also, a review may be performed to check if the business understanding is achieved.

• Deployment Phase: In this phase, the steps are plan deployment, plan monitoring.



**Figure** **3.1.1: CRISP-DM Methodology**

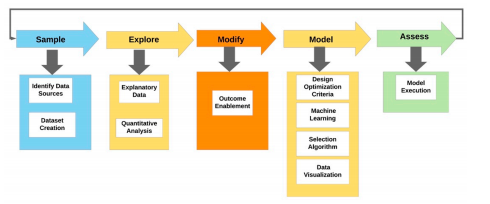
**3.1.2 SEMMA Methodology**

SEMMA methodology is defined by the SAS Institute as the process of sampling, exploring, modifying, modeling, and assessing a large amount of data to uncover previously unknown patterns which can be utilized as a business advantage. The main differences between the CRISP-DM methodology are the steps are not iterative and focus only on the procedures instead of results. some of the SEMMA steps are:

• Sample step: In this step, sample data is limited to collection and analysis of the data contained in form.

• Explore step: Understand the data exploring the outliers, patterns, and relationships.

• Modify step: Modify the data by selecting, transforming and deriving the required feature to enable reaching an outcome



**Figure 1.2 SEMMA Methodology**

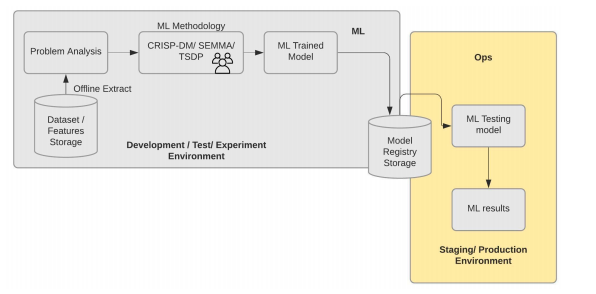
## Working

* + 1. **Machine Learning Manual Pipeline**

Machine learning (ML) manual pipeline process is a convenient and frequently used tool in the study of use cases. The process is script-driven, and every step is manual, including data analysis, data preparation, model training, and data validation. Manual execution of each step and manual transformation from one step to the next is necessary. This process is driven by known analytical methodologies such as CRISP-DM

The purpose of this process is to present a generic approach that is MLOps manual pipeline. The process separates teams, data scientists who create the model, and ops engineers who serve the model as a testing service. Continuous integration (CI) is missing in this process due to few implementations changes. Testing coding is typically a part of the execution of notebooks or scripts.

The scripts and notebooks that implement the steps of the experiment are controlled sources and produce artefacts such as trained models, evaluation metrics, and visualizations. Continuous delivery (CD) is not considered in this process due to not frequent model version deployment. Also, the MLOps manual process does not track or log the model results or actions, which are required in order to detect the model performance degradation The MLOps pipeline process is composed by two teams: ML scientists responsible for the setup of the development environment and the ops engineers responsible for the production environment.



**Figure 3.2.1.1: A Machine Learning Manual Pipeline Process.**

## Proposed Machine Learning Automate Pipeline with CI/CD

The Framework of MLOps brings together various principles of Continuous Integration and Continuous Delivery in a coherent and practical way for ML pipelines.

Process automation of retraining models in production using new data is introduced by two components from the DevOps framework such as continuous integration (CI) and continuous Delivery For a rapid and reliable update of the pipelines in production, we need a robust automated CI/CD system. The automated CI/CD system allows the data scientists to rapidly explore new ideas around feature engineering, model architecture, and hyperparameters. They can implement the process of building a model, checking in the code to a repository, building deployment package such as scripts and packages, and then deploying the new pipeline component.

The main components required for the development and implementation of MLOps pipeline automation with CI/CD routines are as follows

Business problem analysis: In any ML model, firstly, we define the business problem and establish the business understanding and the success criteria for the problem.

Dataset features and storage: In this step, a dataset with features/attributes is presented for the specific business problem.

ML analytical methodology: This step is essential for the selection of the ML methodology as presented in the previous section. During the analysis, the use of TSDP methodology has proven satisfactory in practice of MLOps pipeline.

Pipeline continuous integration (CI): In this stage, we build source code and run various ML trained models. The outputs of this stage are pipeline components (packages and artefacts) to be deployed in the staging/production environment of the continuous delivery (CD).

The machine learning code is just a small portion of real machine learning system; however, to the best of our knowledge, this complexity of the system requires more time and resources. An important component of the machine learning infrastructure is the configuration and data elaboration. The analysis of the data was performed with the help of cleaning, wrangling, and feature extraction. There is no doubt that the training process for the machine learning models may take many efforts; however, we need to invest also in the production process of the model in a way that allows us to retrain and tune the hyperparameters and to use the new data in order to improve the model.

A variety of software tools have been used for the deployment of machine learning models such as Jenkins, Git, Docker, Helm, and Kubernetes.

Version control is the process of tracking each version of the code or data in a way that makes it easy to track the history of changes and return to previous versions at any time, and also to share code with the collaborators. This also applies to the dataset used during each incremental and iterative cycle, and we have to keep track of that data in each increment in order to reproduce the same results in the future if required. In this step, any changes to the source code of the model or changes to the version of the dataset used will be pushed (uploaded) to the repository (code or data) to trigger the continuous delivery build.

Pull changes: In this stage of continuous integration automation, once the build model is triggered, the latest code tests were performed and analyzed by the code repository and then the associated data version is pulled from the data repository storage.

Run unit tests: One of the primary features of this step is the ability to provide robust code that cannot break easily upon changes like package updates, bug fixes, and new features.

Build/run machine learning model code: The design of the machine learning code depends upon the language type used, whether the language is interpreted or compiled. Moreover, this step includes the training and retraining of the ML models. The goal behind the training and retraining is either introducing new data or adjusting the hyperparameters.

Testing and validating code: Machine learning models help improve the decision-making process; therefore, it is significantly essential to have sufficiently accurate models; otherwise, these might have a negative impact rather than positive. An essential part of the machine learning delivery pipeline is to understand how well a model works. In order to validate this ML model, a subset of training data as a test set is selected. This test data is not used to build the ML model, but once the ML model is built, it is used as a test data to evaluate how well the model performs. For validating the code, there are a number of different strategies that can be used to ML models.

• Recreate: Terminate the version A and then roll out the version B.

• Ramped: Version B is rolled out in an increment way to replace version A.

• Blue/Green: Version B is released beside version A, once B is validated and confirmed its functionality traffic is switched to the new version B.

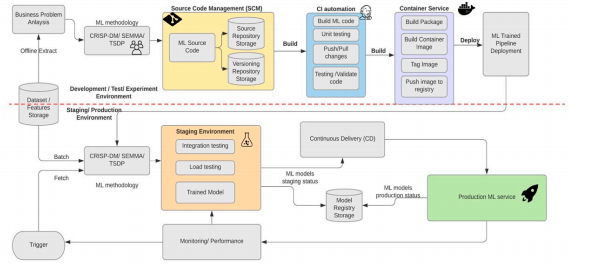
• Canary: Make a subset of the traffic to go to version B while old version A is still serving then proceed to roll out to full users.

Staging environment: In software engineering, it is very common and standard practice to deploy the ML trained model first to a staging environment. Software architecture is no longer limited to the deployed ML pipeline running at the customer infrastructure, but also includes the system being deployed for development and staging as well. The environment term is often used to describe the infrastructure required to run a service or an application (app servers, databases, caches) In this environment, the trained model is used as a modeling tool for the prediction of the testing ML models. The output of this stage is a testing ML model which it is pushed to the model registry storage.

Model registry storage: In this step, the ML models under staging status and ML models under production status are uploaded in a storage location. In addition, the ML engineers (ops) check the ML model objects into a model repository or upload them to a model registry.

Automated triggering: This step is automatically executed on schedule or in response to a trigger in the production environment. The output of this stage is a ML testing model which it is pushed to the staging environment.

The performance monitoring with which ML model operates is of the utmost importance for data scientists. Performance monitoring consists of a wide range of tasks, involves understanding the level of resource utilization, measuring throughput and operating efficiently, and knowing what services are available to users.



**Figure 3.2.2.1 ML pipeline automation for CI/CD**

# CHAPTER 4

**MERITS AND DEMERITS**

## Merits

The MLOps approach is very autonomous compared to conventional approach and very helpful because of the following reasons:

* + - Rapid innovation through robust machine learning lifecycle management.
    - Creation of reproducible workflows and models.
    - Effective management of entire machine learning engineering
    - Easy Deployment of high precision models in any location.
    - Machine Learning resource management system and control

## Demerits

* + - Machine Learning being deployed in production is the level of disruption that may occur with the inclusion of ML into forms/ front end applications.
    - Business is about the evaluation consideration of model risks when actualizing a machine learning model.
    - Major technical challenge is regarding the lack of coordination and improper handoffs between the data scientists and operation teams which can lead to delays and errors.

# CHAPTER 5

**APPLICATIONS**

* + - Business automation which helps us to deliver the product at faster phase.
    - Protection for data and networks.
    - Prevention of unauthorized users.

# CONCLUSION

In this paper, we extensively discussed ML automate pipeline with CI/CD principles. The method is based on DevOps practices, which are responsible for the integration and delivery of ML trained and tested models. We presented different ML methodologies and the TSDP methodology is more appropriate for this study. We discovered that the manual ML method produces high operational costs and delays in business organizations. The proposed automated ML method improves many areas such as time to market, integration across business units, and breakdown departmental silos; it also increases code and deployment quality, productivity, and visibility. However, it is not straightforward; many companies struggle and stuck at the start of the journey while others abort the implementation halfway through the process due to challenges like resistance to change, isolated teams (silos), lack of skillsets, etc. It is critical to understand that DevOps does not stand alone, but it relies on the adoption and integration of multiple frameworks and methodologies like ITSM, Agile, and Lean, The machine learning models lifecycle is different and more complex than traditional software development; it requires extensive work with data extraction, preparation and verification, infrastructure configuration, provisioning, post-production monitoring, and enhancements. Therefore, Agile principles/values and DevOps practices and tools are highly recommended to provide continuous delivery and co-creation of value to customers, increase the model quality, minimize waste, highlight the importance of supporting a rapid feedback loop, accommodate early changes, as well as explore the hidden technical debt that leads to an enormous increase in operational costs of real-world machine learning systems Leveraging our method, future work may be focused on specific aspects of the model for complex systems and develop a useful software to help testing ML models.

# GLOSSARY

**Devops**

DevOps is a set of practices that combines software development and IT operations. It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.

**Machine Learning**

Machine learning is the study of computer algorithms that improve automatically through experience and by the use of data it is seen as a part of artificial intelligence.

**GIT**

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

# ACRONYMS

**CI** Continuous Integration

**CD** ContinuousDelivery

**MLOps** Machine Learning Operations

# BIBLIOGRAPHY

1. Moreira, M. The Agile Enterprise: Building and Running Agile Organizations, 1st ed.; Apress: Berkeley, CA, USA, 2017.
2. Farroha, B.S.; Farroha, D.L. A framework for managing mission needs, compliance, and trust in the DevOps environment. In Proceedings of the IEEE Military Communications Conference, Baltimore, MD, USA, 6–8 October 2014; pp. 288–293.
3. Fitzgerald, B.; Stol, K.J. Continuous Software Engineering: A Roadmap and Agenda. J. Syst. Softw. 2017, 123, 176–189.
4. Bosch, J. Continuous Software Engineering: An Introduction. In Continuous Software Engineering; Springer: Berlin/Heidelberg, Germany, 2014; pp. 3–13.
5. Leppanen, M.; Makinen, S.; Pagels, M.; Eloranta, V.P.; Itkonen, J.; Mantyla, M.V.; Mannisto, T. The Highways and Country Roads to Continuous Deployment..
6. Weber, I.; Nepal, S.; Zhu, L. Developing Dependable and Secure Cloud Applications. IEEE Internet Comput. 2016, 20, 74–79.
7. Jenkins. Build Great Things at any Scale.
8. Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; Crespo, J.F.; Dennison, D. Hidden technical debt in Machine Learning Systems. In Advances in Neural Information Processing Systems; Curran Associates: New York, NY, USA, 2015; Volume 28, pp. 2503–2511.
9. Kontsevoi, B.; Soroka, E.; Terekov, S. TETRA as a set of techniques and tools for calculating technical debt principal and interest. In Proceedings of the IEEE/ACM International Conference on Technical Debt, TechDebt, Montreal, QC, Canada, 26–27 May 2019; pp. 64–65.
10. Virmani, M. Understanding DevOPs and bridging the gap from continuous integration to continuous delivery. In Proceedings of the 5th International Conference on Innovative Computing Technology INTECH, Pontevedra, Spain, 22–25 May 2015; pp. 78–82..
11. Lwakatare, L.E.; Kuvaja, P.; Oivo, M. Relationship of DevOps to Agile, Lean and Continuous Deployment. In Product-Focused Software Process Improvement, Proceedings of the 17th International Conference, PROFES 2016, Trondheim, Norway, 22–24 November 2016; Springer: Cham, Switzerland, 2016; Volume 10027, pp. 399–415.
12. Humble, J.; Farley, D. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation; Pearson Education, Inc.: Boston, MA, USA, 2011.
13. Virmani, M. Understanding DevOPs and bridging the gap from continuous integration to continuous delivery. In Proceedings of the 5th International Conference on Innovative Computing Technology INTECH, Pontevedra, Spain, 22–25 May 2015; pp. 78–82.