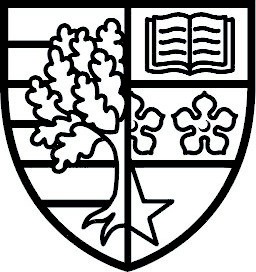
MLOps-Driven Music Recommendation System

Usman Jamshaid Khan

Honours Dissertation

*Supervised by* Dr Abrar Ullah

Deliverable 1: Final Year Dissertation



Heriot-Watt University

School of Mathematical and Computer Sciences

BSc (Hons.) Computer Science (Software Engineering)

September 2024

The copyright in this dissertation is owned by the author. Any quotation from the dissertation or use of any of the information contained in it must be ac- knowledged as the source of the quotation or information.

DECLARATION

I, Usman Jamshaid Khan, confirm that this work submitted for assessment is my own and is ex- pressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: Usman

Date: 14/10/24

i

ABSTRACT

Recently, machine learning (ML) has emerged as a crucial tool across many sectors, many businesses still struggle to effectively implement and maintain ML model in production. MLOps has gained recognition as a critical solution to the problems presented in the ML sector. The aim of this project is to demonstrate how Machine Learning Operations (MLOps) can improve the management and deployment of machine learning models for quick and efficient production. Like DevOps, MLOps uses development, testing, and monitoring to enhance ML production. MLOps comprises procedures for automation, continuous integration and continuous development (CI/CD) to guarantee that ML models can be swiftly and reliably deployed to production. MLOps uses continuous development and continuous monitoring to optimize ML production by tackling problems like slow deployment cycles and inadequate model management.

With the help of MLOps procedures, this project will develop a music recommendation system that makes song recommendations based on face expression detection. The recommendation system will make use of machine learning techniques like Convolutional Neural Networks (CNNs), which are essentially capable of identifying human emotions and recommending music in line with those feelings. By integrating MLOps pipelines, the system will achieve continuous development, automated deployment, and continuous monitoring. This project demonstrates how MLOps will improve the effectiveness and flexibility of face emotion-based recommendation systems.

i

Table of Contents

### [Declaration](#_bookmark0) i

[Abstract](#_bookmark1) iii

[Acknowledgements](#_bookmark2) v

[Table of Contents](#_bookmark3) vii

[List of Figures](#_bookmark4) ix

[List of Tables](#_bookmark5) xi

1. [Introduction](#_bookmark6) 1
   1. [Motivation](#_bookmark7) 1
   2. [Aim and Objectives](#_bookmark8) 2
   3. [Contributions](#_bookmark9) 2
   4. [Organization](#_bookmark10) 2
2. [Background](#_bookmark11) 3
   1. Machine Learning Development Challenges 3
   2. DevOps 4
   3. MLOps 5
   4. MLOps Practice 6
   5. MLOps Use Case 7

2.5.1 Recommendation System 7

2.5.1 Music Recommendation System using Facial Recognition 8

2.5.1 Music Recommendation System using MLOps 8

* 1. Related Work 9
  2. [Summary](#_bookmark14) 9

1. [Method](#_bookmark15)ology 10
   1. Theoretical Basis 10
   2. MLOps Pipelines 10
   3. Data Collection 10
   4. Model for Face Emotion Recognition 11
   5. Continuous Monitoring 11
   6. [Summary](#_bookmark26) 11
2. Requirements Analysis 12
   1. Functional Requirements 12
   2. Non-Functional Requirements 12
3. Evaluation 13
   1. Automation Efficiency 13
   2. Model Versioning 13
   3. Pipeline Efficiency 13
   4. Monitoring 13
   5. [Summary](#_bookmark36) 13
4. Project Management 10
5. PLES and Risk Analysis 10

[References](#_bookmark42) 15

A [Appendix: PLES](#_bookmark46) 18

B [Appendix: Project Management](#_bookmark47)  19

C [Appendix: Risk Analysis](#_bookmark49) 20

List of Figures

1. The Machine Learning Life Cycle 3
2. DevOps Environment Cycle 4
3. [MLOps involves DevOps, ML, and Data Engineering.](#_bookmark48) 5
4. The MLOps Workflow 6
5. MLOps Pipeline 11
6. D1 Gantt Chart 19
7. D2 Gantt Chart 19

ix

List of Tables

1. [Functional Requirements](#_bookmark35)…………………………………………………………………………13
2. Non-[Functional Requirements](#_bookmark35)…………………………………………………………………..13
3. [Risk Analysis](#_bookmark35)……………………………………………………………………………………….20

# INTRODUCTION

Machine learning has greatly improved many systems and one of them is the recommendation systems which can be related to music, e-commerce, movies, etc. These systems are advanced, and the deployment and maintenance of these systems remains challenging because of the complexity of maintaining the scalability and efficiency of models. This is where Machine Learning Operations (MLOps) emerges, which is a Development Operations (DevOps) concept modified for the management and optimization of machine learning processes. In order to ensure that machine learning models operate dependably and efficiently in production settings, MLOps optimizes the whole machine learning lifecycle, from data preprocessing and model development to deploying, and monitoring. This dissertation’s main goal is to create a music recommendation system that uses face recognition to determine the user’s moods and make appropriate song suggestions. The project intends to guarantee effective deployment and maintenance of the face emotion recognition-based music recommendation system by implementing MLOps techniques. Using the music recommendation system as a use case, this research will investigate how MLOps concepts may enable the continuous delivery, oversight, and scalability of recommendation system that uses machine learning.

## Motivation

Machine learning models have shown a lot of progress recently for resolving challenging issues in a variety of industries. Nevertheless, regardless of its potential, ML model implementation and maintenance continue to be extremely difficult. Delivering ML products that are scalable and reliable might be challenging due to difficulties caused by separate techniques from data scientists and ml engineers. By incorporating DevOps ideas into the ML process, MLOps fills the gap by facilitating continuous integration, continuous delivery, training, and monitoring and ensures that models are effectively maintained. In this dissertation, MLOps will be implemented to show how it improves the efficacy of a face emotion recognition-based music recommendation system. As models are developed on face emotions to suggest music, they should constantly adjust to new datasets and enhance the suggestions. By integrating MLOps the system may be designed, implemented, and monitored in an automatic and flexible way.

## Aim and Objectives

In this dissertation, the main aim is to investigate and apply MLOps approaches to create a machine learning pipeline for a music recommendation system that is reliable and scalable. The project will involve investigation into existing MLOps practices and their influence on machine learning processes and will be succeeded by the hands-on application of these methods. From implementation to deployment, and continuous procedures, the project will show how MLOps can greatly improve the operation of machine learning models workflow.

The main objectives are:

* Conduct research on MLOps techniques and typical ML model deployment challenges to comprehend current problems and guide the creation of a reliable and flexible machine learning pipeline.
* Develop ML pipelines with MLOps for effective data collection, training of data, and delivery.
* Implement CI/CD pipeline to automate evaluation, verification, delivery and continuous monitoring to identify model drift and automate retraining.
* Develop a face emotion-based music recommendation system using MLOps practices.
* Evaluate the MLOps pipelines for automation, and efficiency

## Contributions

The contributions of this project are:

* + 1. Adapting MLOps methods to meet the demands of machine learning systems, such as recommendation systems.
    2. Evaluation of how MLOps procedures affect the scalability and reliability of the system.
    3. Implementation of music recommendation system using facial recognition with MLOps procedures.

## Organization

The structure of this dissertation is as follows: Section 1 introduces the topic, outlines the motivation and emphasizes the key contributions. The background and related work are presented in Section 2, which reviews the current state of the art research in MLOps. The methodology is described in Section 3, which explains the system design and implementation techniques. In Section 4 the requirements analysis is described outlining the functional and non-functional requirements which are essential to the development. The evaluation is covered in Section 5 which describes the techniques used to measure the MLOps pipeline’s efficiency. Section 6 covers project management which includes the planning and scheduling of the project. Professional, Legal, Ethical, and Social (PLES) factors are addressed in Section 7, along with risk analysis to identify and handle possible issues.

# BACKGROUND

This background discusses recent studies done on the importance of Machine Learning Operations or (MLOps), focusing on why MLOps is a crucial framework in the field of machine learning and its relevance in music recommendation system using facial recognition. The emphasis is on comprehending how MLOps can improve Machine learning systems, recognizing the difficulties, and investigating relevant research in the field.

## Machine Learning Production Challenges

The field of machine learning enables machines to independently learn from data and previous events in order to recognize patterns in data, categorize data, and forecast outcomes with little assistance from humans (Mitchell & Jordan, 2015). Previously, the machine learning models were typically few in number, making them easier to oversee, or there was lesser organizational interest in comprehending these models and their relationships but with the emergence of decision automation, where choices are made more frequently without human input, machine learning models have become increasingly vital and the management of risk related to these models have become more significant at elevated levels withing the organization (Treveil et al., 2021).

Symeonidis et al. (2022) highlights that it has always been difficult to integrate machine leaning models into production, with data scientists, machine learning engineers and other groups encountering various obstacles during the development of production ready models, which results in very small percentage of ML projects even getting to the stage of production and this is why several tools have emerged to enhance processes such as creating models, processing data, training, all aimed to address those challenges and advance in this field. Figure 1 illustrates the workflow of machine learning life cycle.

A diagram of a process

Description automatically generated

Figure : The Machine Learning Life Cycle (Testi et al., 2022)

According to Treveil et al. (2021), three key factors make managing machine learning life cycles challenging. First there are many dependencies since business demands and data are ever-changing, necessitating ongoing alignment between the model’s performance and its initial objectives. Second, because business, data science, and IT teams employ distinct technologies and expertise, communication gaps develop between them. Finally, juggling several responsibilities strain data scientists, who are frequently not software engineers, and this is particularly true when the number of models increases, and they are expected to manage models that they did not initially develop.

## Development and Deployment methodologies

The implementation of machine learning systems into production has brought various challenges, and to tackle these challenges it’s essential to comprehend how traditional methodologies have progressed to enhance development. While there are several project management approaches available, choosing the best development and deployment technique is essential to guarantee a project’s effectiveness, efficiency, and reliability. Numerous approaches have been put out over time to meet the changing requirements and existing complexities in software development.

## Waterfall

In 1970, Winston Royce unveiled the Waterfall model, one of the first models of software development procedure. There are no overlaps among phases in this ordered, continuous technique, where every step must be finished before the next starts (Sinha & Das, 2021). Figure 2 shows the six different phases which are requirements, design, implementation, verification, deployment and maintenance. Since it necessitates extensive documenting prior to moving on the design phase, Sinha and Das (2021) claim that this methodology works best for projects whose needs are clear and consistent from the start. The design phase involves elements which describe the data movement and hardware requirements (Senarath, 2021). Following testing by the developer to guarantee accuracy, the implementation phase entails code that is based on design documentation (Sinha & Das, 2021). According to Senarath (2021), a specialized team then evaluates the program in the verification phase to make sure it complies with the original specifications and if it does not, then changes are performed. The production problems are resolved, and the program is optimized in the maintenance phase after the project is deployed. The waterfall model works well for projects with precise and steady needs, but this approach can be rigid when adjustments are required.

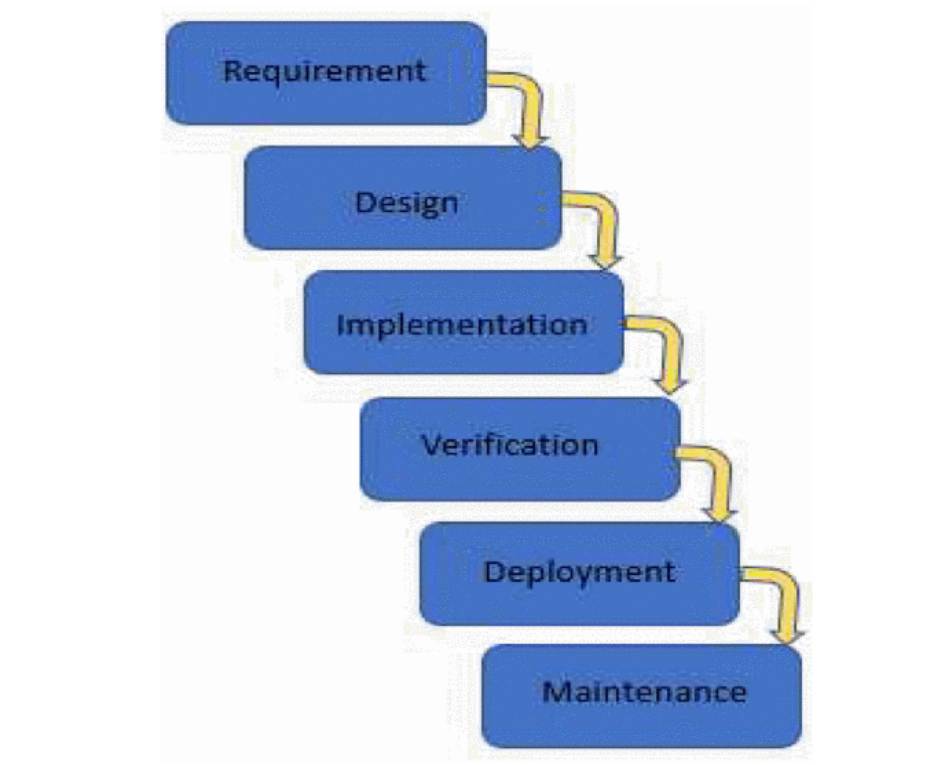


Figure Waterfall model (Sinha & Das, 2021)

## Agile

The Agile methodology is a popular framework for software development that is based on ideals and concepts that prioritize adaptability, teamwork, and ongoing progress (Salza, Musmarra and Ferruci, 2018). Agile encourages a gradual and dynamic technique in contrast to the conventional, Waterfall method, enabling teams to react swiftly to shifting needs (Sinha & Das, 2021). The main concept is to focus on continuous input and modification during a project’s lifespan rather than doing a lot of prior preparation (Salza, Musmarra and Ferruci, 2018). As seen Figure 3, the agile lifecycle comprises of adaptable steps such as requirements, design, development, testing, and deployment and these phases can frequently take place concurrently, speeding up delivery.

A blue circle with arrows

Description automatically generated

Figure Agile lifecycle (Salza, Musmarra and Ferrucci, 2018)

## DevOps

Prior to the development of Machine Learning Operations principles, businesses faced considerable difficulties in implementing solutions developed with the newest machine learning technology because of the substantial resources needed, which is why MLOps was so important (Alla & Adari, 2020). To understand MLOps we need to comprehend the definition of Development Operations (DevOps). DevOps is a collection of procedures that blends the work procedures of the operations groups with developers of software to provide a shared set of procedures that acts as a cross for both positions (Alla & Adari, 2020). A graph depicting the DevOps workflow is shown in Figure 2. It’s a change in structure where, rather than dispersed groups handling tasks independently, collaborative teams focus on ongoing operational feature deployments (Ebert et al., 2016).

With the implementation of DevOps, software development cycles are accelerated, guaranteeing continued software supply, and lowering overall expenses through reduced service costs-achieved through improved process efficiency in maintaining applications (Alla & Adari, 2020). DevOps tools like continuous integration (CI), continuous delivery (CD), automated testing, and monitoring enable those advantages. By using a continuous integration approach, software development companies aim to incorporate code created by developers and make frequent improvements to it (Deza & Gift, 2021). Continuous delivery is a practice of developing and testing code constantly without human involvement. Among the crucial techniques the most practiced are continuous integration and continuous delivery. Automated testing guarantees that modifications to the code do not create any new bugs or errors (Liu et al., 2023).

Monitoring allows teams to observe and evaluate the system's performance, enabling them to identify any problems. According to Kreuzberger et al. (2023), implementing DevOps techniques may enhance software quality and drastically cut time-to-market by encouraging an increasingly reactive and agile development process.

Diagram of software development process

Description automatically generated

Figure : An illustration of a DevOps environments cycle (Alla & Adari, 2020).

## MLOps

Since DevOps has been successfully implemented, businesses are searching for continuous methods for creating machine learning systems (John, Olsson and Bosch, 2021). To handle the difficulties of implementing and maintaining ML models, MLOps applies that concept of DevOps and the procedures of Data Engineering to the ML lifecycle as shown in Fig 3. Model creation, deployment, monitoring and maintenance are all included in MLOps which aims to optimize ML processes and guarantee that models are flexible, repeatable and maintained (Singla, 2023). A fundamental element of MLOps is the implementation of continuous integration and continuous deployment for machine learning models, which streamlines the training, validation and deployment processes for these models, and it allows models to be enhanced progressively without the need for human involvement (Kreuzberger et al., 2023). MLOps brings forth a new approach, alongside continuous integration and continuous deployment, known as continuous training (CT), which focuses on the automatic retraining of models when necessary. Continuous monitoring is also essential as it involves the ongoing evaluation of model performance to identify and resolve issues like model drift or poor data quality (Hymel et al., 2022). Version control is also crucial, which involves continuously developing different stages of machine learning models in order to keep track and revert to earlier versions as necessary. MLOps automates the process of machine learning models, extending beyond traditional software processes and it also combines the workflows of data scientists and Data engineers with those of the developers to enable the continuous delivery of high-quality machine learning models (Alla & Adari, 2020). Figure 4 shows a typical MLOps workflow.

A diagram of machine learning

Description automatically generated

Figure : MLOps involves DevOps, ML, and Data Engineering (Hewage and Meedeniya,2022)

A diagram of a model development

Description automatically generated

Figure : The MLOps Workflow (Testi et al., 2022)

## MLOps Practice

MLOps offers advantages, but it also encounters various difficulties. Technical problems like versioning models might be difficult because the models must be reproducible. Since models can act differently in development and production contexts, it can be difficult to ensure reliable performance across a variety of scenarios (Singla, 2023). This is in accordance with the difficulties mentioned by Ruf et al. (2021), and (Ghantous and Gill, 2017), which emphasizes how crucial it is to choose the appropriate tools for MLOps pipelines in order to guarantee quality and conformity with DevOps procedures. Implementing MLOps is further complicated by the intricacy of the deployment of machine learning systems numerous clouds settings (Banerjee et al., 2020). Implementing effective version control system and containerization methods can make the models reproducible and the deployment consistent (Singla, 2023).

The Organization difficulties include promoting teamwork among cross-disciplinary groups, like data scientists, ML engineers, and operations personnel, who might prioritize different objectives and operate with varying processes and diverse technologies and frameworks are frequently used by organizations, which can cause integrity issues and inefficiency (Singla, 2023). Granlund et al. (2021), discusses the difficulties in implementing hybrid models in various businesses and offers strategies to make deploying easier in these situations. The creation of specialized MLOps teams that prioritize interaction and information exchange can promote cross-functional cooperation. Organizational silos may be reduced, and processes streamlined by incorporating MLOps into current DevOps procedures.

Because implementing MLOps methods requires a change in processes and thinking, cultural difficulties include a reluctance to change, and the lack of skills is also a problem because teams must train and develop in order to use MLOps procedures effectively (Singla, 2023). Moreover, it is essential for everyone involved to have a common grasp of ML concepts to ensure effective implementation and training and skill development initiatives can help close gaps in skill while fostering a culture of constant progress and flexibility. (Singla, 2023). Furthermore, as suggested (Hewage and Meedeniya, 2022), the accessibility of technologies that are open source might improve interoperability and facilitate incorporation ain many environments. Overall, with implementing ML-specific automated testing and continuous integration and continuous deployment the dependability of ML systems will be increased, and the standards and compatibility are enhanced by open-source tools, which make integration into many ecosystems easier (Singla, 2023).

## MLOps Use Case

The necessity for real-time adaptation to users changing preferences, which traditional models cannot provide, makes recommendation systems an ideal application for MLOps.

## 2.5.1 Recommendation System

Chen & Chen (2005) explain that every time new data is produced, traditional recommendation systems must be retrained, which reduces efficiency and slows the improvements. They emphasize that MLOps automates the deploying and retraining of models, allowing for continuous procedures and quick response to changes. This feature is essential for recommendation systems in customized domains like music, where rapid response of appropriate recommendations improves user engagement. Given the vast number of products accessible online there is a demand for systems that help users find their desired items and recommendation system is a key service that assists users in overcoming the challenges of excessive information.

A recommendation system is a software application that provides strategies to propose recommendations to a user based on the likelihood of their preference. Collaborative filtering, content-based filtering, and hybrid systems are three of the prevalent methods utilized in recommendation systems. Collaborative filtering is based on the actions of user’s similar tastes to generate recommendations. Content-based filtering focuses on the characteristics of items to suggest similar options. In collaborative filtering, if a user prefers X, and Y is identical to X, eventually the user might also prefer Y and in content-based filtering, if a user X and a user Y are identical, then user X may prefer what user Y prefers (Fessahaye et al.) Hybrid systems merge both strategies to improve precision and resilience. Strong machine learning models that can manage massive data sets, update instantly and adjust to shifting consumer suggestions are necessary for recommendation systems to function well. The application of cutting-edge ML methods, such as deep learning and reinforcement learning has significantly enhanced the effectiveness of recommendation system making them more precise and effective (Wang et al., 2023).

## Music Recommendation using Facial Recognition

The Music Recommendation system that uses facial recognition can provide incredibly customized experiences by recognizing user emotions and pairing music appropriately. This system examines the user's facial expressions via a camera and employs machine learning algorithms to understand the emotions being shown and recommend music based on the users’ mood (Pradhan et al., 2022). According to Mishra et al. (2024), the system assesses the user’s mood to provide a playlist that corresponds with the way they are feeling. For example, it may select lively songs from a playlist to change a depressed mood or offer melodies to enhance positive thoughts when the user’s mood is good. Emotion detection using facial recognition can be achieved by deep learning techniques. Deep Leaning, utilizing artificial neural networks that feature numerous hidden layers to replicate the human brains cerebral cortex has greatly progressed the field of computer vision, especially via convolutional neural network (CNN), a robust algorithm, effectively handles millions of parameters by employing filters to analyze 2D images and generate data (Chauhan et al., 2018). A CNN model for face detection analyzes face images to identify features, such as facial landmarks, and works with a dataset containing emotions and this dataset goes through preprocessing steps that include reshaping, resizing and converting to arrays and the CNN then interprets the pixel information to determine the user’s emotions (Shakti et al., 2023). When the emotion is detected, a song will be played that matches the emotion.

## Music Recommendation System using MLOps

The implementation of a music recommendation system using MLOps practices is crucial as it will manage the life cycle of a model. MLOps automates machine learning operations and improves dependability to expedite the development and deployment of a music recommendation system. According to Susila (2022), MLOps creates a systematic pipeline for model creation, deployment and monitoring by including continuous integration and continuous delivery (CI/CD) into the systems model lifecycle, drawing inspiration from DevOps. The report further explains that the systems speed and precision in suggesting music according to identified emotions are improved by this method, which guarantees that every update of model, trained on face recognition data and emotion signals, is effectively evaluated and implemented. In order to maintain an efficient and flexible recommendation system that keeps up with changing user information MLOps automates the models process, which makes it more scalable and permits immediate changes, and facilities the continuous monitoring. All things considered, MLOps will offer crucial protocols for setting up and managing music recommendation systems that are always updated effortlessly.

## Related Work

There have been numerous studies where MLOps have been implemented in systems to enhance development and deployment. Integrating DevOps techniques to improve automation, expedite machine learning operations, and shorten production duration has been the subject of several research. Though these methods have effectively tackled a number of issues, they still highlight shortcomings in areas.

## Effective Use of MLOps In Music Recommendation System

In the study of Susila (2022), MLOps was used to implement the music recommendation system in order to avoid the problem of manually training data every time there’s update in data, which is time consuming and inefficient. They ensured that model changes were easily incorporated by automating the retraining process with MLOps. Furthermore, they mentioned that the stages are separated into Continuous Integration and Continuous Delivery. Continuous integration involved model parameters being regularly assessed to get higher precision and the model was only permitted to move to subsequent phase after it satisfied the necessity accuracy requirements. The verified model was then deployed to systems during the continuous delivery stage so that it can be accessible. Their approach enabled automation for quick response to modifications and better retraining but since it was mostly restricted to fixed dataset, it might not perform well in varied or uncertain situations.

## MLOps application in Electricity Forecasting market

Automating the delivery operations of machine learning models has become crucial, particularly in applications the run continuously such as forecasting, where the use of MLOps procedures is important and in recent research by Subramanya, Sierla and Vyatkin, (2022), they integrate DevOps ideas into the machine learning lifecycle to give a combined strategy to MLOps. For the Finnish power market, the researchers created flexible MLOps pipelines designed for time-series forecasting and tools like Jenkins, Docker, and MLflow were used for continuous procedures. An effective method for handling data input, training, and delivery is shown by the usage of pipelines and features stores. The research’s concentration is on predetermined time limits and its adaptability to a dynamic, user focused systems where continuous changes are essential. Furthermore, although the study emphasizes pipeline architecture and applies the methodology requirement like forecasting electricity, it still depends on human changes for data which can cause difficulties in deployment environments. In addition, the study focuses on time-series for controlled data but ignores other issues with user preferences, and uncontrolled data which can be found in a recommendation system

## Automating ML models Using MLOps

Utilizing tools like DVC, MLflow and GitHub Actions to manage continuous integration and continuous deployment procedures, a recent study has shown the efficacy of employing MLOps principles to automate machine learning pipelines (N Sirisha et al., 2024). These methods have proven effective in decreasing manual interventions and accelerating deployments, especially for predicting jobs like estimating the wines attributes. However, the research mostly concentrates on models of sequential processing instead of interactive systems and even though many tasks have been automated, problems like data drift continue to call for manual data evaluation and retraining. This raises a problem when growing to systems like recommendation systems where data changes quickly.

## MLOps-Enabled Security Strategies for Operational Technologies

As businesses work to safeguard sensitive data and preserve model dependability in production settings, the incorporation of MLOps standards for the safe deployment of machine learning models have grown in importance and recent research examines the best procedures for securing storage of models, data encryption and monitoring (Ahmad et al., 2024). The authors stress the importance of employing strong encryption techniques, such as AES and SSL protocols, to secure data while it is being sent and stored. In order to guarantee reproducibility and flexibility, the research further investigates model safety by utilizing version control, infrastructure as code and containerization with Docker and Kubernetes. While the paper highlights the significance of safety protocols on conventional machine learning deployments it mainly concentrates on protecting static data instead of instantaneously dynamic machine learning systems. It further highlights how manual procedures may become barriers when growing to the unpredictable circumstances. Additionally, the study skips over the difficulties of incorporating these safety measures into user intensive systems where quick model changes are crucial.

## Summary

The related work examines many developments in the field of MLOps, emphasizing how the deployment of machine learning models has been enhanced by the incorporation of MLOps procedures. Using continuous integration and delivery pipelines, research has shown how well MLOps may improve flexibility and efficiency of productions. Nevertheless, regardless of these achievements, current approaches frequently concentrate on certain systems situations, exposing gaps in places like automating pipelines, retraining, and data management. In order to broaden MLOps concepts to more flexible and user focused systems which call for adaptable pipelines, it is important that these gaps be filled.

# METHODOLOGY

This methodology section will cover the integration of machine learning operations (MLOps) methods to expedite the development, deployment, testing and monitoring of machine learning models, aligning with the objectives of this study. The Methodology includes the application of MLOps to a face emotion-based music recommendation system. The main aim is to demonstrate how well MLOps manages the whole lifespan of machine learning models.

The methodology is organized as follows: Initially the design of MLOps pipelines is discussed, with an emphasis on technologies that make model procedures easier, such as MLflow and Jenkins. Next, there will be discussion on how these pipelines will work with the implementation of face emotion-based music recommendation system and finally the Model monitoring will be covered. Python will be the main programming language utilized for the duration of this project because of its extensive library of machine learning tools, which makes it the perfect choice.

## MLOps Framework Overview

MLOps, a collection of procedures designed to streamline the construction and operation of machine learning systems, is the theoretical basis of this study. Through the automation of processes like deployment, versioning, testing, and monitoring, MLOps enhances the scalability, reproducibility, and effectiveness of machine learning workflows. The goal of this research is to create MLOps pipelines that smoothly flow into the model development procedure.

## MLOps Pipelines

Developing and deploying MLOps pipelines to automate essential ML workflow tasks is the basis of this study. Figure 7 shows the stages that are part of the MLOps process. The system will make use of:

* Docker for Containerization to guarantee constant deployment.
* Jenkins to automate CI/CD procedures to guarantee that models are validated, verified and then deployed automatically with little human intervention.
* MLflow for tracking performance, model versions and experiments results.

Collectively, these technologies will automate the steps of model development such as model training, evaluation, and delivery, allowing for ongoing upgrades. Automating the deployment procedure will be made possible via CI/CD pipelines and if modifications are uploaded, these pipelines will react immediately. In order to ensure that the system stays updated and continues to produce correct results when new data becomes available, Jenkins will handle model retraining.

## Data Collection

The CK+ (Extended Cohn-Kanade) dataset which contains labeled facial expressions for range of emotions, will be used in this study. The data will be preprocessed, which includes operations like picture resizing and normalization, to guarantee that the data is clear and to manage these preprocessing effectively, Python packages like OpenCV will be utilized.

## Model for Face Emotion Recognition

When the data is ready, a convolutional Neural Network (CNN) will be trained using the CK+ dataset for detecting facial emotions. PyTorch will be used for training the model and to ensure good management and reproducibility, MLflow will be used for tracking and versioning. This will allow us to work with different versions of models and provide continuous model enhancement.

## Continuous Integration and Monitoring

Jenkins will be used by the system to automate the deployment workflow. It comprises continuous integration, which tests and validates model modification, and continuous delivery, which automatically deploys verifies models to production. The model’s accuracy will be monitored using MLflow, which will record and display key metrics like model accuracy and response time as well as measure the way the model reacts to incoming data and allow assessment of multiple models. This method guarantees that models stay updated, and any notable modifications may be promptly found and fixed.

## Summary

The methodology shows how MLOps effectively deploy and manage machine learning models and project intends to provide a reliable and flexible pipeline for face-emotion based music recommendation system utilizing Docker for containerization, MLflow for tracking, and Jenkins for CI/CD.

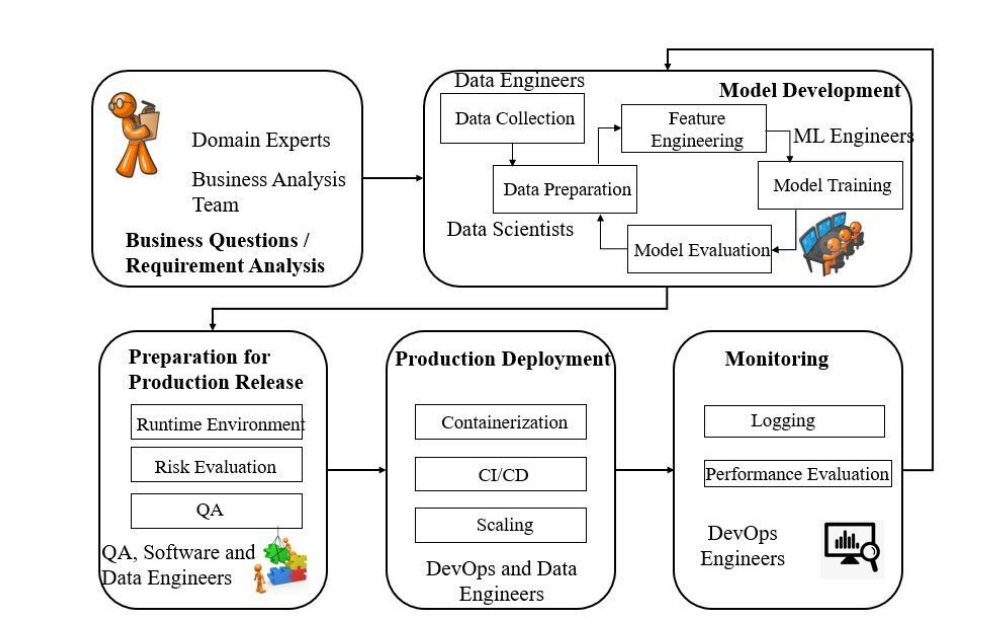


Figure MLOps workflow (Hewage and Meedeniya, 2022)

# REQUIREMENTS ANALYSIS

This part describes the functional requirements (FR) and non-functional requirements (NFR) with MoSCoW analysis to rank the requirements. To determine the project goals and demands and to comprehend what the system will achieve, requirements analysis is crucial. We will also include MoSCoW analysis which is used to classify the requirements into categories like Must Have, Should Have, Could Have, and Won’t Have.

## Functional Requirements

Functional Requirements specify the precise actions and features it must have in order to achieve its goals, and they outline the functions and responses of the system to various inputs.

Table 1: Functional Requirements

|  |  |  |
| --- | --- | --- |
| ID | Requirement | MoSCoW |
| FR1 | The system must automate model training using datasets | Must Have |
| FR2 | The system must automatically test the models or changes in models | Must Have |
| FR3 | The system must automate deployment of models | Must Have |
| FR4 | The system must be able to retrain when new data is available | Must Have |
| FR5 | The system should automate validation of model performance metrics | Must Have |
| FR6 | The system should provide containerized deployment | Should Have |
| FR7 | The system should automatically trigger model retraining if there’s drop in accuracy or the amount of error increases | Could Have |
| FR8 | The system should manage versions of model for tracking | Should have |
| FR9 | The system should continuously monitor for model drift | Could have |
| FR10 | The system should provide output on pipeline executions times | Should have |

## Non-Functional Requirements

Non-Functional Requirements define the quality, limitations, and performance of a system and instead of defining what system should do, non-functional requirements specify how it should function.

Table 2: Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| ID | Requirement | MoSCoW |
| NFR1 | The system should execute pipeline within an hour. | Must have |
| NFR2 | The system should provide monitoring output for performance data | Could have |
| NFR3 | The system should automate procedures to reduce human intervention | Should have |
| NFR4 | The system should support integration of new models, datasets, and pipelines. | Could Have |
| NFR5 | The system should optimize resource usage to minimize functional costs | Won’t Have |
| NFR6 | The system should automatically restore from errors without any manual input. | Should Have |
| NFR7 | The system should be able to scale up to accommodate growing data volumes | Could Have |
| NFR8 | The system will not have graphical user interface for music recommendation system | Won’t have |
| NFR9 | The system should guarantee reproducible model training and deployment | Must Have |

# EVALUATION

The MLOps pipeline evaluation technique is centered on evaluating the machine learning workflows overall performance and efficiency. This also covers continuous model training, deployment, versioning and monitoring. The objective is to guarantee the pipelines smooth operation, effective, and continuous deployment.

## Automation Efficiency

With little human involvement, the MLOPs pipeline should automate the crucial steps of model training, deployment, and testing. The time required to continuously initiate and carry out each procedure, including training and deploying models when fresh data becomes available, will be measured in order to assess the efficiency of automation. Important metrics to assess are deployment speed, duration of testing, training, and pipeline response time to new data.

## Model Versioning

For machine learning models to be reproducible, model versioning is essential. The system’s capacity to monitor various model versions and datasets will be evaluated. A good version control can quickly roll back to earlier model version and the metric for the accuracy of version control includes the success rate formula.

## Pipeline Efficiency

The whole machine learning cycle, from data intake to continuous deployment, is integrated into MLOps pipeline and the efficiency of the pipelines will be evaluated. Total pipeline execution time and success rate for automation are the important metrics to be assessed.

## Monitoring

Continuous monitoring guarantees that the model operates at its best when produced. This evaluation will assess how well the system monitors performance. Metrics include the duration of identifying model problems and time taken to retrain in order to fix them.

## Summary

With an emphasis on automation, versioning, pipeline efficiency, and monitoring, the evaluation approach assesses the essential components of MLOps pipeline. By assessing these components, we can be assured that the system can scale according to increasing needs and preserve excellent performance.

# Project Management

For the projects’ activities and timelines to be efficiently organized, tracked and managed, a project management plan is necessary. It contains deadlines, detailed assignments, and significant events that are necessary to guarantee the project’s success. We will use a Gantt chart to demonstrate this, which gives a schedule for project completion and track progress by specifying deadlines and task durations. Refer to Appendix B for the Gantt charts which includes the project timeline and deliverables.

# PLES and Risk Analysis

The analysis of professional, legal, ethical, and social issues related to this project is provided in Appendix A. Risk Analysis is an important component of the project which makes sure that potential problems are identified and mitigated. A detailed risk analysis is provided in Appendix C.

# REFERENCES

Alla, S. and Adari, S.K. (2020). What Is MLOps? *Beginning MLOps with MLFlow*, [online] pp.79–124. doi:https://doi.org/10.1007/978-1-4842-6549-9\_3.

Deza, A. and Gift, N. (2021) *Practical Mlops*. sebastopol, california: O’Reilly Media, Inc.

Symeonidis, G., Nerantzis, E., Kazakis, A. and Papakostas, G.A. (2022). MLOps - Definitions, Tools and Challenges. *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*. [online] doi:https://doi.org/10.1109/ccwc54503.2022.9720902.

Fessahaye, F., Perez, L., Zhan, T., Zhang, R., Fossier, C., Markarian, R., Chiu, C., Zhan, J., Gewali, L. and Oh, P. (2019). *T-RECSYS: A Novel Music Recommendation System Using Deep Learning*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICCE.2019.8662028.

Chen, H.-C. and Chen, A.L.P. (2005). A Music Recommendation System Based on Music and User Grouping. *Journal of Intelligent Information Systems*, [online] 24(2-3), pp.113–132. doi:https://doi.org/10.1007/s10844-005-0319-3.

Jordan, M.I. and Mitchell, T.M. (2020). Machine learning: Trends, perspectives, and prospects. *Science*, [online] 349(6245), pp.255–260. doi:https://doi.org/10.1126/science.aaa8415 (Accessed: 23 October 2024).

Treveil, M. *et al.* (2021) *Introducing Mlops: How to scale machine learning in the Enterprise*. Sebastopol, CA, california: O’Reilly Media, Inc.

Ebert, C., Gallardo, G., Hernantes, J. and Serrano, N. (2016). DevOps. *IEEE Software*, [online] 33(3), pp.94–100. doi:https://doi.org/10.1109/ms.2016.68 (Accessed: 24 October 2024).

John, M.M., Olsson, H.H. and Bosch, J. (2021). *Towards MLOps: A Framework and Maturity Model*. [online] IEEE Xplore. doi:https://doi.org/10.1109/SEAA53835.2021.00050.

Singla, A. (2023). Machine Learning Operations (MLOps): Challenges and Strategies. *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, [online] 2(3), pp.333–340. doi:https://doi.org/10.60087/jklst.vol2.n3.p340 Accessed 28 Oct. 2024.

Liu, Z., Xu, W., Zhang, W. and Jiang, Q. (2023). An emotion-based personalized music recommendation framework for emotion improvement. *Information Processing & Management*, [online] 60(3), p.103256. doi:https://doi.org/10.1016/j.ipm.2022.103256 Accessed 28 Oct. 2024.

Wang, D., Zhang, X., Yin, Y., Yu, D., Xu, G. and Deng, S. (2023). Multi-View Enhanced Graph Attention Network for Session-Based Music Recommendation. *ACM Transactions on Information Systems*, [online] 42(1), pp.1–30. doi:https://doi.org/10.1145/3592853 (Accessed 28 Oct. 2024).

Kreuzberger, D., Kühl, N. and Hirschl, S. (2023). Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *IEEE Access*, [online] 11, pp.1–1. doi:https://doi.org/10.1109/access.2023.3262138.

Granlund, T., Kopponen, A., Stirbu, V., Myllyaho, L. and Mikkonen, T. (2021). *MLOps Challenges in Multi-Organization Setup: Experiences from Two Real-World Cases*. [online] IEEE Xplore. doi:https://doi.org/10.1109/WAIN52551.2021.00019.

Hymel, S., Banbury, C., Situnayake, D., Elium, A., Ward, C., Kelcey, M., Baaijens, M., Majchrzycki, M., Plunkett, J., Tischler, D., Grande, A., Moreau, L., Maslov, D., Beavis, A., Jongboom, J. and Reddi, V.J. (2022). Edge Impulse: An MLOps Platform for Tiny Machine Learning. *arXiv:2212.03332 [cs]*. [online] Available at: https://arxiv.org/abs/2212.03332 [Accessed 4 Nov. 2024].

Hewage, N. and Meedeniya, D. (2022). MACHINE LEARNING OPERATIONS: A SURVEY ON MLOPS TOOL SUPPORT \*. *ArXiv*. [online] doi:https://doi.org/10.48550/arXiv.2202.10169.

Pradhan, A., Rathod, V., Jamdar, S. and Dangche, K. (2024). *Web Protection by Bitdefender*. [online] Ijeast.com. Available at: https://www.ijeast.com/papers/68-70 [Accessed 4 Nov. 2024].

Mishra, N., Gupta, R. and Raj, A. (2024). Music Recommendation System by Analyzing Facial Emotions Using Deep Neural Network. *SSRN Electronic Journal*. [online] doi:https://doi.org/10.2139/ssrn.4834085.

Chauhan, R., Ghanshala, K.K. and Joshi, R.C. (2018). *Convolutional Neural Network (CNN) for Image Detection and Recognition*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICSCCC.2018.8703316.

Susila, P. (2022). *Effective Use Of Mlops In Music Recommendation System MSc In Cloud Computing*. [online] *NCI*. Available at: https://norma.ncirl.ie/6476/1/pradeepmanoharansusila.pdf [Accessed 4 Nov. 2024].

Ruf, P., Madan, M., Reich, C. and Ould-Abdeslam, D. (2021). Demystifying MLOps and Presenting a Recipe for the Selection of Open-Source Tools. *Applied Sciences*, [online] 11(19), p.8861. doi:https://doi.org/10.3390/app11198861.

Ghantous, G. and Gill, A. (2017). *Association for Information Systems AIS Electronic Library (AISeL) DevOps: Concepts, Practices, Tools, Benefits and Challenges*. [online] PACIS. Available at: https://opus.lib.uts.edu.au/bitstream/10453/130066/1/DevOps-%20Concepts%20Practices%20Tools%20Benefits%20and%20Challenges.pdf [Accessed 4 Nov. 2024].

Banerjee, A., Chen, C.-C., Hung, C.-C., Huang, X., Wang, Y. and Chevesaran, R. (2020). *Challenges and Experiences with MLOps for Performance Diagnostics in Hybrid-Cloud Enterprise Software Deployments Challenges and Experiences with MLOps for Performance Diagnostics in Hybrid-Cloud Enterprise Software Deployments*. [online] USENIX. Available at: https://www.usenix.org/system/files/opml20\_paper\_banerjee.pdf [Accessed 4 Nov. 2024].

Shakti Sri, S., Sathya, S. and Pandiarajan, T. (2023). An Improved Music Recommendation System for Facial Recognition and Mood Detection. *ITM Web of Conferences*, [online] 56, p.01004. doi:https://doi.org/10.1051/itmconf/20235601004.

Subramanya, R., Sierla, S. and Vyatkin, V. (2022). From DevOps to MLOps: Overview and Application to Electricity Market Forecasting. *Applied Sciences*, [online] 12(19), p.9851. doi:https://doi.org/10.3390/app12199851.

Testi, M., Ballabio, M., Frontoni, E., Iannello, G., Moccia, S., Soda, P. and Vessio, G. (2022). MLOps: a Taxonomy and a Methodology. *IEEE Access*, [online] 10, pp.63606–63618. doi:https://doi.org/10.1109/access.2022.3181730.

Senarath, U. (2021). Waterfall Methodology, Prototyping and Agile Development. *Waterfall Methodology, Prototyping and Agile Development*. [online] doi:https://doi.org/10.13140/RG.2.2.17918.72001.

Sinha, A. and Das, P. (2021). *Agile Methodology Vs. Traditional Waterfall SDLC: A case study on Quality Assurance process in Software Industry*. [online] IEEE Xplore. doi:https://doi.org/10.1109/IEMENTech53263.2021.9614779.

N Sirisha, Kiran, A., Arshad, M. and Mounika M (2024). Automating ML Models Using MLOPS. [online] 3, pp.1–5. doi:https://doi.org/10.1109/assic60049.2024.10507923.

Ahmad, T., Adnan, M., Rafi, S., Muhammad Azeem Akbar and Anwar, A. (2024). MLOps-Enabled Security Strategies for Next-Generation Operational Technologies. *Association for Computing Machinery*. [online] doi:https://doi.org/10.1145/3661167.3661283.

# APPENDIX: PLES

Professional Issues:

There aren’t any major professional, legal, ethical, or social issues for this project. The project’s aim excludes any actions covered by specific codes of conduct and issues covered by certain computer laws, such as those related to data protection. Proper citations are mentioned for all ideas referenced from other sources to ensure academic integrity.

Legal Issues:

The projects datasets, CK+ and DEAM, are open-source, publicly accessible, and comprise anonymous data which guarantees that there are no privacy issues.

Open-source tools such as PyTorch, Jenkins, MLflow, and others facilitate the development of machine learning models.

Ethical Issues:

There are no ethical concerns that need to be addressed because the research does not use human subjects or gather private information. In the CK+ dataset explicit consent was taken from participants when data was being collected. As this project relies on ethically cleared and publicly available datasets no sensitive information was gathered.

Social Issues:

The main objective of this research is system performance and not commercial deployment. To avoid misuse, access to the system is restricted to supervisors and other authorized individuals.

# APPENDIX: PROJECT MANAGEMENT

A screenshot of a computer

Description automatically generated

Figure D1 Gantt Chart

A screenshot of a chat

Description automatically generated

Figure D2 Gantt Chart

# C APPENDIX: RISK ANALYSIS

Table 3 outlines the potential risks associated with the project, along with plans for mitigating such risks. The impact column shows how serious the risk will affect, and likelihood column tells the probability of the risk occurring. Both columns represent the corresponding degrees of effect and likelihood using L, M, and H which means low, medium and high respectively.

Table 3: Risk Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk ID | Risk | Impact | Likelihood | Mitigation Plan |
| R1 | Poor data quality in datasets | H | M | Use alternate dataset or verify data |
| R2 | Pipeline failure | H | L | Use version control to rollback |
| R3 | Project Delay | M | L | Follow the timetable and review work |
| R4 | Deadline not met | H | L | Contact supervisor and discuss about progress in meetings |
| R5 | Pipeline tools issues | H | L | Use alternate tools or perform testing |

The explanation of the impact and likelihood of each risk are provided below:

R1- Poor data quality in datasets:

* Impact (H): Poor data quality may result in erroneous model prediction, which would impair the systems functionality.
* Likelihood(M): Although there is moderate possibility of running into incorrect records, the CK+ and Deam datasets are dependable.

R2- Pipeline failure:

* Impact (H): If a CI/CD pipeline fails it can stop the process, which can cause delays.
* Likelihood(L): Version Control can lower the chances of this failure, but disruptions can occur.

R3- Project Delay:

* Impact (M): Although failed objectives might result in delays, they can usually be controlled with careful planning.
* Likelihood(L): The delays are less likely when the project plan is followed.

R4- Deadline not met:

* Impact (H): There may be serious repercussion if important due dates are missed, particularly if they are submissions or a demo.
* Likelihood(L): Observing the progress can reduce the chance.

R5- Pipeline tools issues:

* Impact (H): If the tools that are used fails or cause errors it can cause development delays.
* Likelihood(L): The tools used are very reliable and have good documentations. There can still be implementation issues, but the probability is very low.