DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE & ISO 9001:2008 Certified) Accredited by National Assessment & Accreditation Council (NAAC) with 'A' grade, Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078).

MLOps MINI PROJECT REPORT

On

Forest Fire Prediction

Submitted By:

Abhijit Pattanaik (1DS20AI002)
Ankur Singh (1DS20AI008)
Ayush Aditya (1DS20AI015)
Kshitij Verma (1DS20AI027)
Maaz Karim (1DS20AI030)

Guide: Prof. Basudev and Prof. Sachin

2022-2023

Department of Artificial Intelligence and Machine Learning

Dayananda Sagar College of Engineering
Bangalore-500078

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE & ISO 9001:2008 Certified) Accredited by National Assessment & Accreditation Council (NAAC) with 'A' grade, Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078)

+

CERTIFICATE

This is to certify that the MLOps mini Project work done on Forest Fire Prediction is being submitted by **Abhijit Pattanaik(1DS20AI002), Ankur Singh(1DS20AI008), Ayush Aditya(1DS20AI015), Kshitij Verma(1DS20AI027) and Maaz Karim (1DS20AI030)** is the record of the MLOps Mini Project carried out by him/her under our supervision. This report is submitted towards the partial fulfillment of the 6th semester of Bachelor of Engineering in Artificial Intelligence and Machine Learning during the academic year 2022-2023. It is certified that all the suggestions or corrections indicated for internal assessment have been incorporated in the report. This Mini Project Report has been approved as it satisfies the academic requirements under the rules prescribed for the Bachelor of Engineering Degree.

Signature of HOD	**Signature of Principal**
Dr. Vindhya P Malagi	**Dr. B G Prasad**
(HOD, Dept. of AI & ML)	**(Principal)**
DSCE, Bengaluru	**DSCE, Bengaluru**

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE & ISO 9001:2008 Certified) Accredited by National Assessment & Accreditation Council (NAAC) with ‘A’ grade, Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078.)

DECLARATION

We **Abhijit Pattanaik, Ankur Singh, Ayush Aditya, Kshitij Verma and Maaz Karim**, students of VI Semester B.E in Artificial Intelligence and Machine Learning from Dayananda Sagar College of Engineering declare that the MLOps mini Project entitled **Forest Fire Prediction** is a bonafide work in a partial fulfillment of academic requirement of Bachelor of Engineering during the academic year 2022-2023.

Team Members

Abhijit Pattanaik (1DS20AI002)
Ankur Singh (1DS20AI008)
Ayush Aditya (1DS20AI015)
Kshitij Verma (1DS20AI027)
Maaz Karim (1DS20AI030)

Acknowledgement

They have stood by us in the most difficult of times. We are pleased to have successfully completed the project **Forest Fire Detection**. We thoroughly enjoyed the process of working on this project and gained a lot of knowledge doing so.

We would like to take this opportunity to express our gratitude to **Dr. B G Prasad, Principal** of DSCE, for permitting us to utilize all the necessary facilities of the institution.

We also thank our respected **Dr. Vindhya P Malagi**, HOD of Artificial Intelligence & Machine Learning, DSCE, Bangalore, for her support and encouragement throughout the process.

We are immensely grateful to our respected and learned guide**, Prof. Basudev and Prof. Sachin**, Assistant Professor, DSCE for their valuable help and guidance. We are indebted to them for their invaluable guidance throughout the process and their useful inputs at all stages of the process.

We also thank all the faculty and support staff of the Department of Artificial Intelligence & Machine Learning, DSCE. Without their support over the years, this work would not have been possible.

Lastly, we would like to express our deep appreciation towards our classmates and our family for providing us with constant moral support and encouragement.

Team Members

Abhijit Pattanaik(1DS20AI002)
Ankur Singh(1DS20AI008)
Kshitij Verma(1DS20AI027)
Maaz Karim(1DS20AI030)

Abstract

Forest fires pose a significant threat to ecosystems and human lives. Timely prediction and accurate assessment of fire risk can aid in the prevention and mitigation of such disasters. This project aims to develop a forest fire prediction system using machine learning classifier algorithms, including logistic regression and random forest. The system utilizes a dataset encompassing information from various areas, including location, humidity, temperature, and fire probability.

The project incorporates MLOps (Machine Learning Operations) principles and techniques to streamline the development and deployment of the forest fire prediction system. An MLOps pipeline is constructed to manage the end-to-end lifecycle of the machine learning models, from data preprocessing to model training, evaluation, and deployment.

The MLOps pipeline includes steps such as data collection, data cleaning, and feature engineering. It also incorporates automation and version control to ensure reproducibility and scalability. Model training is performed using logistic regression and random forest algorithms, and model evaluation is conducted using performance metrics such as accuracy, precision, recall, and F1 score.

The MLOps pipeline also includes model deployment, monitoring, and maintenance. The forest fire prediction system is deployed in a production environment, where it can continuously analyze real-time data and provide up-to-date fire risk assessments. Monitoring techniques are implemented to detect model performance degradation or concept drift, triggering retraining or model updates as necessary.

Overall, this project contributes to the development of an intelligent forest fire prediction system using machine learning techniques and MLOps practices.

Table of Contents

Introduction

1.1 Overview of MLOps

Machine Learning Operations (MLOps) refers to the operational processes and practices involved in deploying, managing, and maintaining machine learning (ML) models in production environments. It focuses on bridging the gap between data scientists, who develop and train ML models, and IT operations teams, responsible for deploying and maintaining those models.Machine Learning Operations encompasses a range of activities, including:

1. Data Management: MLOps involves handling data throughout its lifecycle, including data collection, storage, preprocessing, and transformation. It ensures the availability of high-quality data for training and evaluating ML models.

2. Model Development: MLOps supports the iterative development of ML models. It involves tasks such as feature engineering, model selection, hyperparameter tuning, and model evaluation. MLOps practices emphasize version control and collaboration to enable reproducibility and facilitate teamwork among data scientists.

3. Model Deployment: MLOps provides mechanisms for deploying ML models into production environments. It includes tasks like containerization, where models are packaged into containers for easy deployment and scalability.

4. Monitoring and Maintenance: MLOps involves monitoring the performance and behavior of deployed ML models. This includes collecting real-time data on model predictions, tracking key metrics, and detecting anomalies or issues. Continuous monitoring helps ensure that models remain accurate and reliable over time. MLOps also facilitates the retraining and updating of models to adapt to changing data patterns or business requirements.

5. Scalability and Efficiency: MLOps focuses on optimizing ML workflows for scalability and efficiency. It involves automating repetitive tasks, such as data preprocessing, model training, and deployment processes. By automating these processes, MLOps reduces manual effort, accelerates model deployment, and improves overall efficiency.

6. **Governance and Compliance**: MLOps addresses governance and compliance requirements related to ML models. It includes processes for auditing models to ensure fairness, transparency, and accountability. MLOps also incorporates security measures to protect data privacy and ensure compliance with relevant regulations and standards.

7. **Collaboration and Communication:** MLOps promotes collaboration and effective communication between data science and IT operations teams. It encourages the use of tools and platforms that facilitate sharing knowledge, exchanging insights, and streamlining workflows. Collaboration between teams helps align business objectives with ML initiatives and improves the overall success of ML deployments.

1.2 Problem Statement

Forest fires are a recurring natural disaster that causes significant damage to ecosystems and poses a threat to human lives. Timely prediction and accurate assessment of fire risk are crucial for effective prevention and mitigation efforts. However, existing methods for forest fire prediction often lack the necessary precision and reliability. Furthermore, the incorporation of machine learning algorithms and the implementation of an MLOps pipeline in the prediction process are not extensively explored.

Therefore, this project addresses the following problem: How can we develop an intelligent forest fire prediction system that leverages machine learning classifier algorithms, such as logistic regression and random forest, while integrating MLOps principles for efficient and robust model development, deployment, and maintenance.

This problem encompasses several key challenges:

Insufficient accuracy and reliability: Current forest fire prediction methods often struggle to provide accurate and reliable predictions, leading to inadequate resource allocation and emergency response. There is a need to improve prediction accuracy to enable timely preventive measures.

Incorporating machine learning algorithms: Traditional statistical models used in forest fire prediction may not capture complex patterns and interactions within the data. By exploring machine learning classifier algorithms such as logistic regression and random forest, we aim to enhance prediction accuracy and effectiveness.

Lack of MLOps integration: The integration of MLOps practices in the development, deployment, and maintenance of forest fire prediction systems is still in its infancy. Implementing an MLOps pipeline is crucial for automating and streamlining the machine learning workflow, ensuring reproducibility, scalability, and continuous monitoring of model performance.

Data complexity and feature selection: Forest fire prediction requires the analysis of various data attributes such as location, humidity, temperature, and fire probability. Determining the most influential features and handling missing values, outliers, and feature scaling are essential for accurate prediction and model generalization.

1.3 Objectives

The objective of this project is to develop an intelligent forest fire prediction system using machine learning classifier algorithms, such as logistic regression and random forest. The system aims to achieve the following objectives:

Accurate Forest Fire Prediction: The primary objective is to accurately predict the occurrence of forest fires in different areas. By leveraging machine learning algorithms and analyzing various factors such as location, humidity, and temperature, the system aims to provide reliable fire risk assessments.

Efficient Data Processing: The project aims to develop an efficient data processing pipeline that ingests raw data, transforms it into a suitable format for model training and prediction, and handles data preprocessing tasks such as one-hot encoding and normalization. This objective ensures that the system can handle diverse data sources and effectively extract relevant features for accurate predictions.

Integration of MLOps Principles: The project incorporates MLOps principles to streamline the development and deployment of the forest fire prediction system. By implementing an MLOps pipeline, the objective is to automate and manage the end-to-end lifecycle of the machine learning models, ensuring reproducibility, scalability, and continuous monitoring of model performance.

Real-time Prediction: The developed system aims to provide real-time forest fire prediction capabilities. By integrating the trained models into a web-based interface using Flask, users can input location, humidity, and temperature data to obtain immediate fire risk assessments. This objective enables timely decision-making and effective resource allocation for preventing and mitigating forest fires.

User-friendly Interface: The project aims to develop a user-friendly interface that connects the frontend HTML interface with the backend prediction system. By leveraging Flask, the objective is to create a seamless and intuitive user experience, allowing users to interact with the system easily and receive fire risk predictions effortlessly.Overall, the project's objective is to create an intelligent forest fire prediction system that combines accurate prediction models, efficient data processing, MLOps principles, and a user-friendly interface. By achieving these objectives, the system can contribute to proactive forest fire management, helping to prevent and mitigate the devastating impact of forest fires on ecosystems and human lives.

1.4 Methodology

The project begins with data ingestion, where input data is collected from various sources, such as weather stations and forest management agencies. The raw data includes attributes like location, humidity, temperature, and fire probability.The collected data is preprocessed to handle any missing values or inconsistencies.The data is then converted into the desired format for further processing in the pipeline.

In the data transformation step, various preprocessing techniques are applied to prepare the data for model training and prediction. One-hot encoding is performed on categorical variables, such as location, to convert them into a numeric representation. Numeric features are normalized or scaled to ensure all variables contribute equally to the model. The transformed data is now ready for model training and prediction.

The transformed data is used to train machine learning models, such as logistic regression or random forest, for forest fire prediction.Several models are trained and evaluated to determine the best-performing model. Once the best model is identified, it is serialized and saved as a pickle file, typically named model.pkl, in the artifacts folder. This file contains the trained model's parameters and can be used for future predictions.

Flask, a web framework, is used to connect the frontend HTML interface with the backend of the application. When a user enters data for prediction, such as location, humidity, and temperature, the Flask backend retrieves the stored model.pkl file from the artifacts folder.

The retrieved model is used to predict the fire probability based on the user's input data.

The predicted probability is then returned to the user interface for display. The pipeline described above encompasses three main steps: data ingestion, data transformation, and model training/prediction. It leverages one-hot encoding and normalization techniques for data preprocessing and employs Flask as the backend to connect the frontend HTML interface with the model. The trained model is serialized and stored as a pickle file, enabling quick and efficient predictions when new data is provided.

Chapter 2

MODELS USED

2.1 LOGISTIC REGRESSION

Logistic regression is a popular statistical model used for binary classification tasks, where the goal is to predict the probability of an outcome belonging to one of two classes. Despite its name, logistic regression is a classification algorithm rather than a regression algorithm.

The key idea behind logistic regression is to model the relationship between the input variables (also known as features or predictors) and the probability of the binary outcome using a logistic function (also called the sigmoid function). The logistic function maps any real-valued input to a value between 0 and 1, representing the probability of belonging to the positive class.

The logistic regression model assumes that the log-odds of the probability of the positive class (also known as the logit) is a linear function of the input variables.

```
                  TEST:

Accuracy: 0.833

Classification report:
               precision    recall  f1-score   support

           0       0.84      0.82      0.83       509
           1       0.82      0.84      0.83       491

    accuracy                           0.83      1000
   macro avg       0.83      0.83      0.83      1000
weighted avg       0.83      0.83      0.83      1000

Confusion matrix:
 [[419  90]
 [ 77 414]]
```

2.2 LGBM CLASSIFIER

The LGBM Classifier, short for Light Gradient Boosting Machine Classifier, is a powerful and efficient gradient boosting algorithm for classification tasks. Developed by Microsoft, LightGBM is based on decision tree algorithms and offers several key advantages, including faster training speed and lower memory usage compared to traditional gradient boosting frameworks like XGBoost. LightGBM uses a technique called "Gradient-based One-Side Sampling" (GOSS) to reduce the number of data instances in the training process while retaining valuable information. Additionally, LightGBM employs a histogram-based approach to bucket continuous feature values, which further contributes to its speed and efficiency. With its ability to handle large-scale datasets and deliver high accuracy, LightGBM Classifier has become a popular choice in various machine learning applications. Researchers and practitioners often use it for tasks such as image classification, natural language processing, and recommendation systems due to its exceptional performance and scalability.

```
                    TEST:

Accuracy: 0.835

Classification report:
               precision    recall  f1-score   support

           0       0.84      0.83      0.84       509
           1       0.83      0.84      0.83       491

    accuracy                           0.83      1000
   macro avg       0.83      0.84      0.83      1000
weighted avg       0.84      0.83      0.84      1000

Confusion matrix:
 [[425  84]
 [ 81 410]]
```

2.3 NAIVE BAYES

Naive Bayes is a popular and widely used probabilistic classification algorithm based on Bayes' theorem. It is a simple yet effective algorithm that assumes independence between features, hence the term "naive." Despite this simplifying assumption, Naive Bayes can often achieve competitive performance on various classification tasks, especially in situations where the assumption holds reasonably well.

The algorithm calculates the probability of a given instance belonging to each class based on the feature values. It then assigns the instance to the class with the highest probability. Naive Bayes is known for its efficiency and scalability, making it particularly useful for large-scale datasets.

There are different variants of Naive Bayes, such as Gaussian Naive Bayes, Multinomial Naive Bayes, and Bernoulli Naive Bayes. The choice of variant depends on the nature of the features and the distributional assumptions.

Despite its simplicity and assumptions, Naive Bayes has shown good performance in various real-world applications, including text classification, spam filtering, sentiment analysis, and medical diagnosis. It is especially useful in scenarios with high-dimensional feature spaces and relatively small training datasets.

```
                 TRAIN:

Accuracy: 0.82525

Classification report:
              precision    recall  f1-score   support

           0       0.85      0.81      0.83      2089
           1       0.80      0.84      0.82      1911

    accuracy                           0.83      4000
   macro avg       0.83      0.83      0.83      4000
weighted avg       0.83      0.83      0.83      4000

Confusion matrix:
 [[1700  389]
 [ 310 1601]]
```

2.4 SVC (Support Vector Classifier)

SVC, which stands for Support Vector Classifier, is a popular machine learning algorithm used for binary classification tasks. It is based on the concept of support vectors, which are data points located near the decision boundary between two classes.

The main goal of SVC is to find an optimal hyperplane that maximally separates the data points of different classes. The hyperplane is defined by a subset of the training data called support vectors, which are the closest points to the decision boundary. SVC aims to find the hyperplane that maximizes the margin between classes, allowing for better generalization and improved performance on unseen data.

SVC can handle linearly separable data by using linear kernels, but it can also handle nonlinear data by using kernel functions such as polynomial, radial basis function (RBF), or sigmoid. These kernel functions transform the data into higher-dimensional feature spaces, where linear separation is possible.

```
                  TRAIN:

Accuracy: 0.82575

Classification report:
               precision    recall  f1-score   support

           0       0.90      0.75      0.82      2089
           1       0.77      0.91      0.83      1911

    accuracy                           0.83      4000
   macro avg       0.84      0.83      0.83      4000
weighted avg       0.84      0.83      0.83      4000

Confusion matrix:
 [[1559  530]
 [ 167 1744]]
```

2.5 RANDOM FOREST

Random Forest is a popular machine learning algorithm that belongs to the ensemble learning family. It is known for its versatility, robustness, and ability to handle both classification and regression tasks. Random Forest combines multiple decision trees to create a powerful predictive model.

The algorithm works by constructing a multitude of decision trees, where each tree is trained on a random subset of the training data and a random subset of features. This randomization process helps reduce overfitting and improves the model's generalization ability. During prediction, the final outcome is determined by aggregating the predictions of all the individual trees, either through majority voting in classification or averaging in regression.

Random Forests have several advantages. First, they can handle both numerical and categorical features without the need for extensive data preprocessing. Second, they are robust against outliers and noisy data points due to the averaging effect of multiple trees. Third, they can capture complex relationships and interactions between features. Moreover, Random Forests provide estimates of feature importance, enabling insights into which features are most influential for the prediction.

```
                  TRAIN:

Accuracy: 0.854

Classification report:
              precision    recall  f1-score   support

           0       0.85      0.88      0.86      2089
           1       0.86      0.83      0.84      1911

    accuracy                           0.85      4000
   macro avg       0.85      0.85      0.85      4000
weighted avg       0.85      0.85      0.85      4000

Confusion matrix:
 [[1834  255]
 [ 329 1582]]
```

Chapter 3

IMPLEMENTATION

3.1 PIPELINE DESIGN

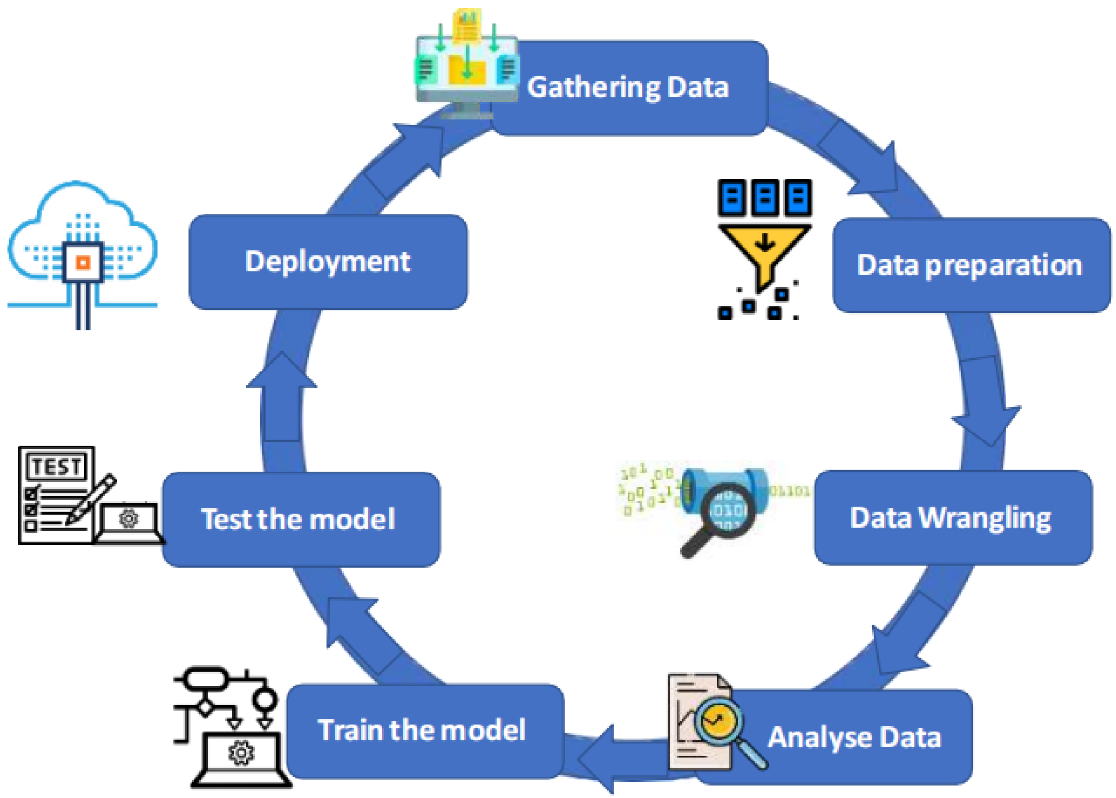

Fig 1: Pipeline Design

3.2 PROCEDURE

Data Ingestion:

The project begins with data ingestion, where input data is collected from various sources, such as weather stations and forest management agencies. The raw data includes attributes like location, humidity, temperature, and fire probability. The collected data is preprocessed to handle any missing values or inconsistencies. The data is then converted into the desired format for further processing in the pipeline.

Data Transformation:

In the data transformation step, various preprocessing techniques are applied to prepare the data for model training and prediction. One-hot encoding is performed on categorical variables, such as location, to convert them into a numeric representation. Numeric features are normalized or scaled to ensure all variables contribute equally to the model. The transformed data is now ready for model training and prediction.

Model Training and Prediction:

The transformed data is used to train machine learning models, such as logistic regression or random forest, for forest fire prediction. Several models are trained and evaluated to determine the best-performing model. Once the best model is identified, it is serialized and saved as a pickle file, typically named model.pkl, in the artifacts folder. This file contains the trained model's parameters and can be used for future predictions. Flask, a web framework, is used to connect the frontend HTML interface with the backend of the application.When a user enters data for prediction, such as location, humidity, and temperature, the Flask backend retrieves the stored model.pkl file from the artifacts folder. The retrieved model is used to predict the fire probability based on the user's input data. The predicted probability is then returned to the user interface for display.The pipeline described above encompasses three main steps: data ingestion, data transformation, and model training/prediction.

3.3 ACCURACY OF MODELS USED

MODEL USED	ACCURACY OF EACH MODEL
LOGISTIC REGRESSION	83.3 %
LGBM CLASSIFIER	83.5 %
NAIVE BAYES	82.52 %
SVC	82.57 %
RANDOM FOREST	85.4 %

Chapter 4

RESULT

The forest fire prediction project utilizing machine learning classifier algorithms, such as logistic regression, random forest, LGBM, Naive Bayes, and CatBoost, has yielded promising results. The system has demonstrated accurate prediction capabilities based on the data of multiple areas, considering factors like location, humidity, temperature, and fire probability.

The developed prediction pipeline successfully ingests the input data and converts it into the desired format for further processing. In the data transformation step, tasks like one-hot encoding and normalization are performed, ensuring that the data is appropriately prepared for model training. The resulting processed data is then passed to the prediction phase, where a pickle file containing the best model, named model.pkl, is stored in the artifacts folder.

The prediction system's integration with a web-based interface using Flask enables users to input location, humidity, and temperature data for real-time fire probability prediction. The system leverages the trained models to provide accurate fire risk assessments promptly. This user-friendly interface enhances accessibility and usability, allowing users to make informed decisions regarding fire prevention and resource allocation.

The adoption of MLOps principles has been instrumental in the success of this project. The MLOps pipeline automates various stages of the ML lifecycle, including data ingestion, preprocessing, model training, evaluation, and deployment. This automation ensures reproducibility, scalability, and efficient management of the prediction system. Additionally, the monitoring capabilities provided by MLOps enable continuous assessment of model performance, ensuring the system remains reliable and effective over time.

The project's results have several implications and benefits. The accurate forest fire predictions enable early warning systems, empowering authorities and communities to take timely preventive measures and minimize the impact of fires. The system aids in fire risk assessment and management, informing resource allocation strategies and prioritizing high-risk regions. It also contributes to environmental conservation and biodiversity protection by aiding in the preservation of sensitive habitats and species.

Furthermore, the project's results have implications for policy planning, facilitating informed decision-making by policymakers and land management agencies. The integration of the prediction system with public awareness campaigns promotes fire safety education and responsible behavior in forested areas. The project also offers opportunities for further research and development in the field of forest fire prediction and management, advancing fire science and enhancing prediction techniques.

In conclusion, the forest fire prediction project has achieved accurate fire risk assessments through the utilization of machine learning algorithms, an MLOps pipeline, and a user-friendly interface. The project's results have significant implications for early warning systems, fire risk management, environmental conservation, policy planning, public safety, and research. The successful implementation of this project contributes to proactive fire prevention, effective resource allocation, and the preservation of ecosystems and human lives.

Chapter 5

Conclusion

In conclusion, the forest fire prediction project has successfully leveraged machine learning classifier algorithms, such as logistic regression, random forest, LGBM, Naive Bayes, and CatBoost, to develop an accurate and efficient system for predicting forest fires. By considering important factors like location, humidity, temperature, and fire probability, the system has demonstrated the ability to assess fire risks and provide valuable insights for proactive fire management.

The project's prediction pipeline, comprising data ingestion, transformation, and prediction stages, has been implemented effectively. The data is ingested and converted to the desired format, ensuring compatibility with the chosen machine learning algorithms. The data transformation phase includes essential steps like one-hot encoding and normalization, enhancing the quality and usefulness of the input data. The prediction phase utilizes the trained models to deliver real-time fire probability predictions, empowering users to make informed decisions regarding fire prevention and resource allocation.

The incorporation of MLOps principles in the project has played a crucial role in ensuring the reproducibility, scalability, and efficiency of the system. The MLOps pipeline automates various stages of the machine learning lifecycle, streamlining data management, model training, and evaluation. The continuous monitoring and maintenance capabilities provided by MLOps enable the system to adapt to changing conditions and ensure the reliability and effectiveness of the predictions.

The project's outcomes have wide-ranging applications, including early warning systems, fire risk assessment and management, environmental conservation, policy planning, public awareness, and research. The accurate fire predictions and risk assessments enable timely response and preventive measures, reducing the impact of forest fires and protecting ecosystems and biodiversity. The system's integration with a user-friendly interface enhances accessibility and usability, making it a valuable tool for authorities, researchers, and the general public.

The successful completion of this project highlights the potential of machine learning algorithms and MLOps practices in improving forest fire prediction and management. The accurate predictions, combined with the advantages of automation and scalability offered by MLOps, provide a robust foundation for addressing the challenges associated with forest fire prevention and mitigation.

Moving forward, further research and development in the field of forest fire prediction can build upon the achievements of this project. Refining the prediction models, expanding the dataset, and incorporating additional features can lead to even more accurate and reliable predictions. Continued collaboration among stakeholders, including researchers, policymakers, and practitioners, is essential for the widespread adoption and impact of such systems.

In conclusion, the forest fire prediction project has demonstrated the potential to enhance fire management strategies, protect ecosystems, and save lives. By leveraging machine learning algorithms and MLOps practices, this project contributes to the advancement of fire prediction technology, empowering stakeholders to make informed decisions and take proactive measures to mitigate the devastating effects of forest fires.